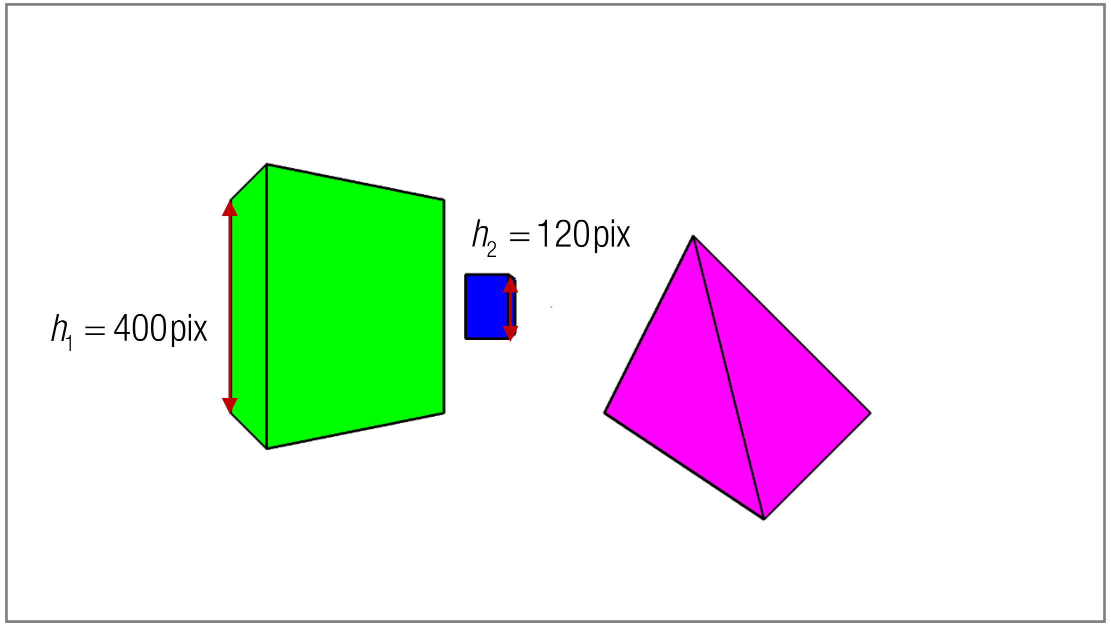
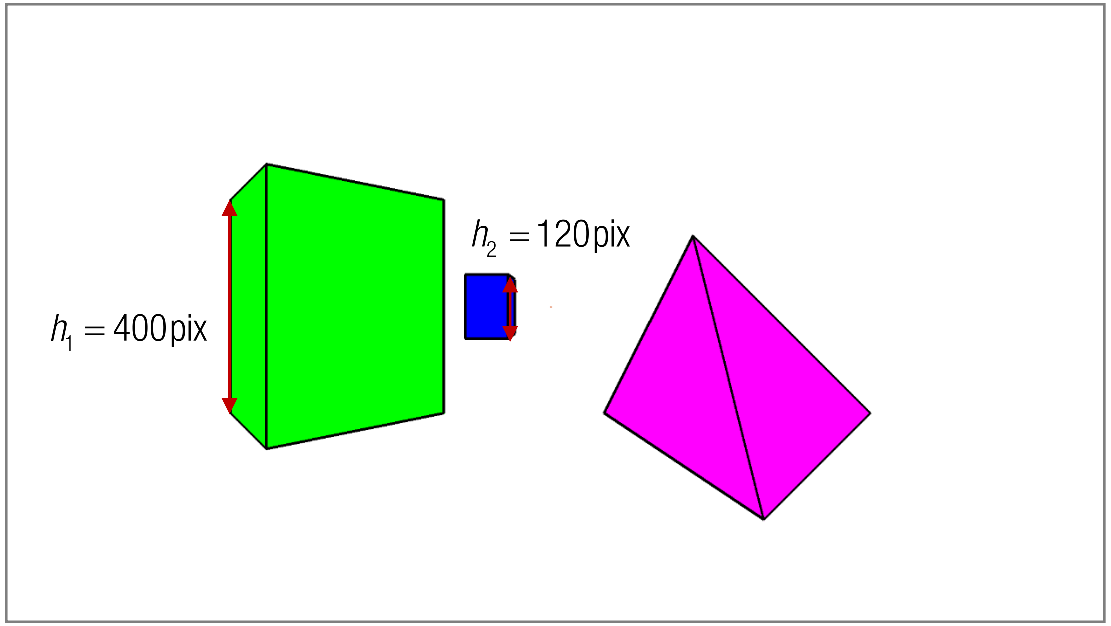
Camera Projection Matrix

Where am I with Dolly Zoom?



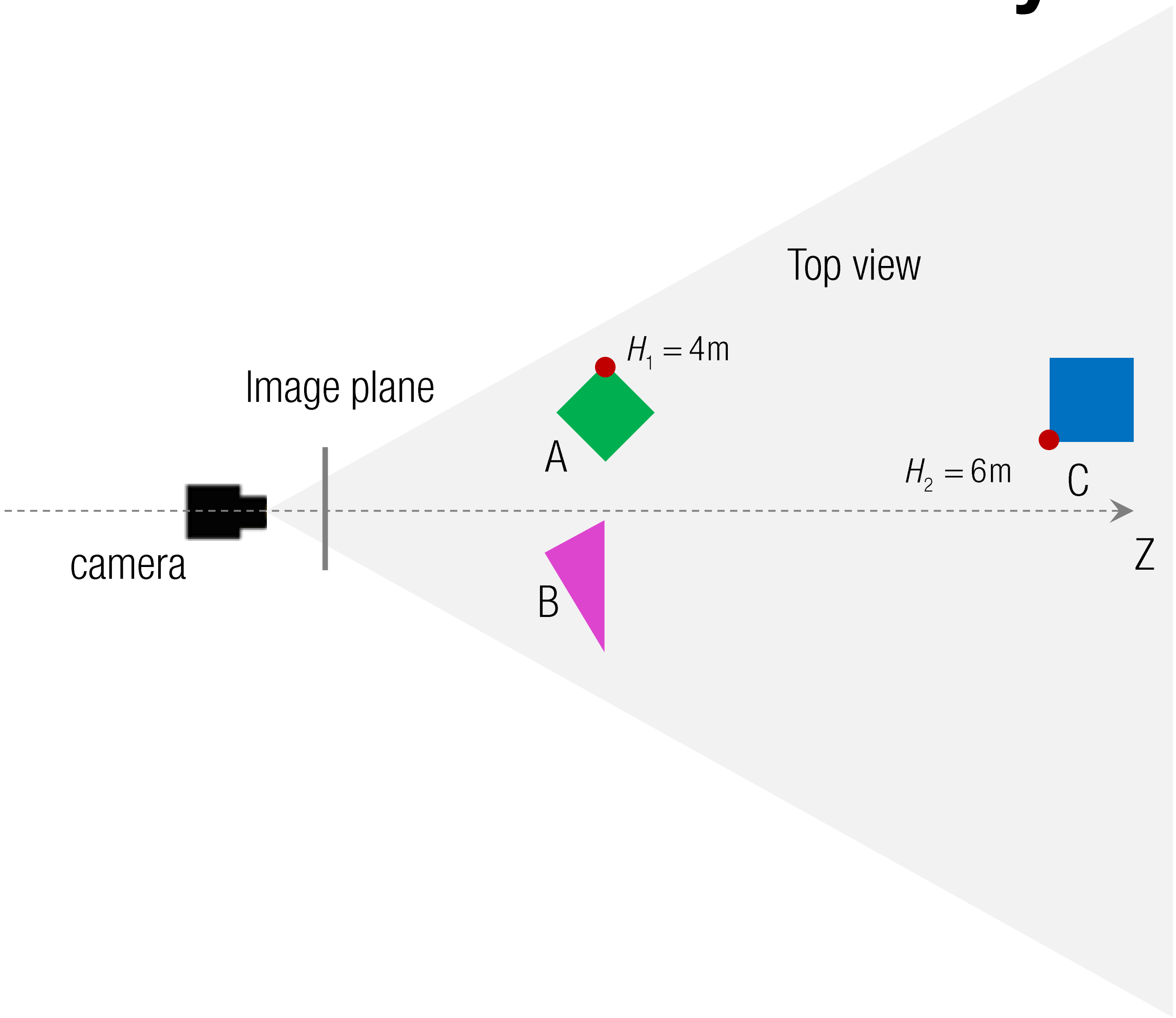
Where am I with Dolly Zoom?

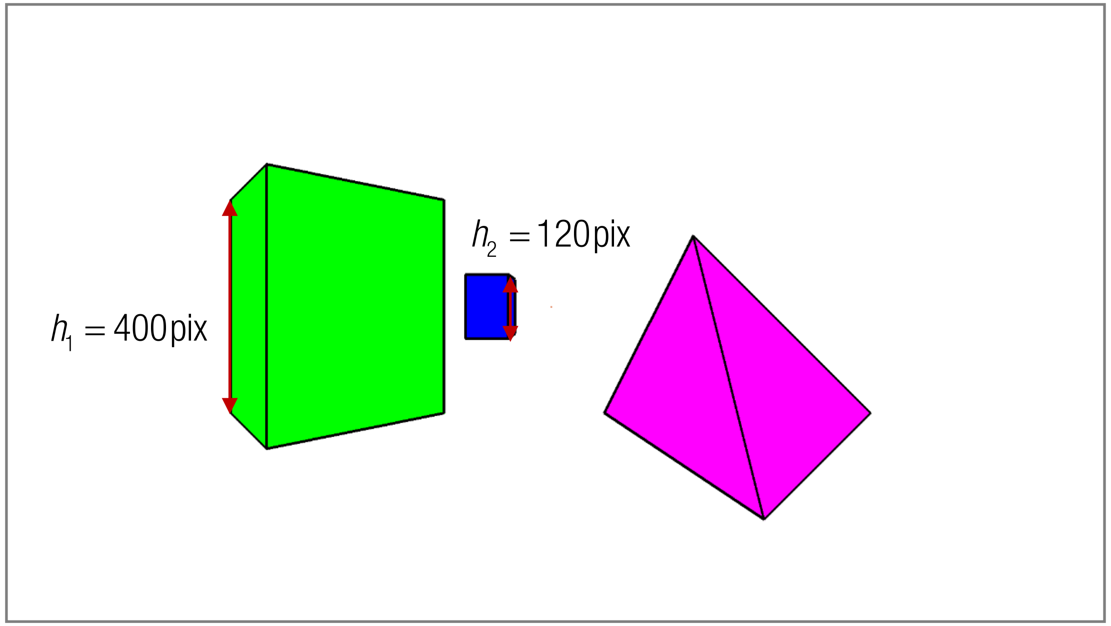


How far I need to step back with zoom factor x2?

How will h_2 change?

Where am I with Dolly Zoom?

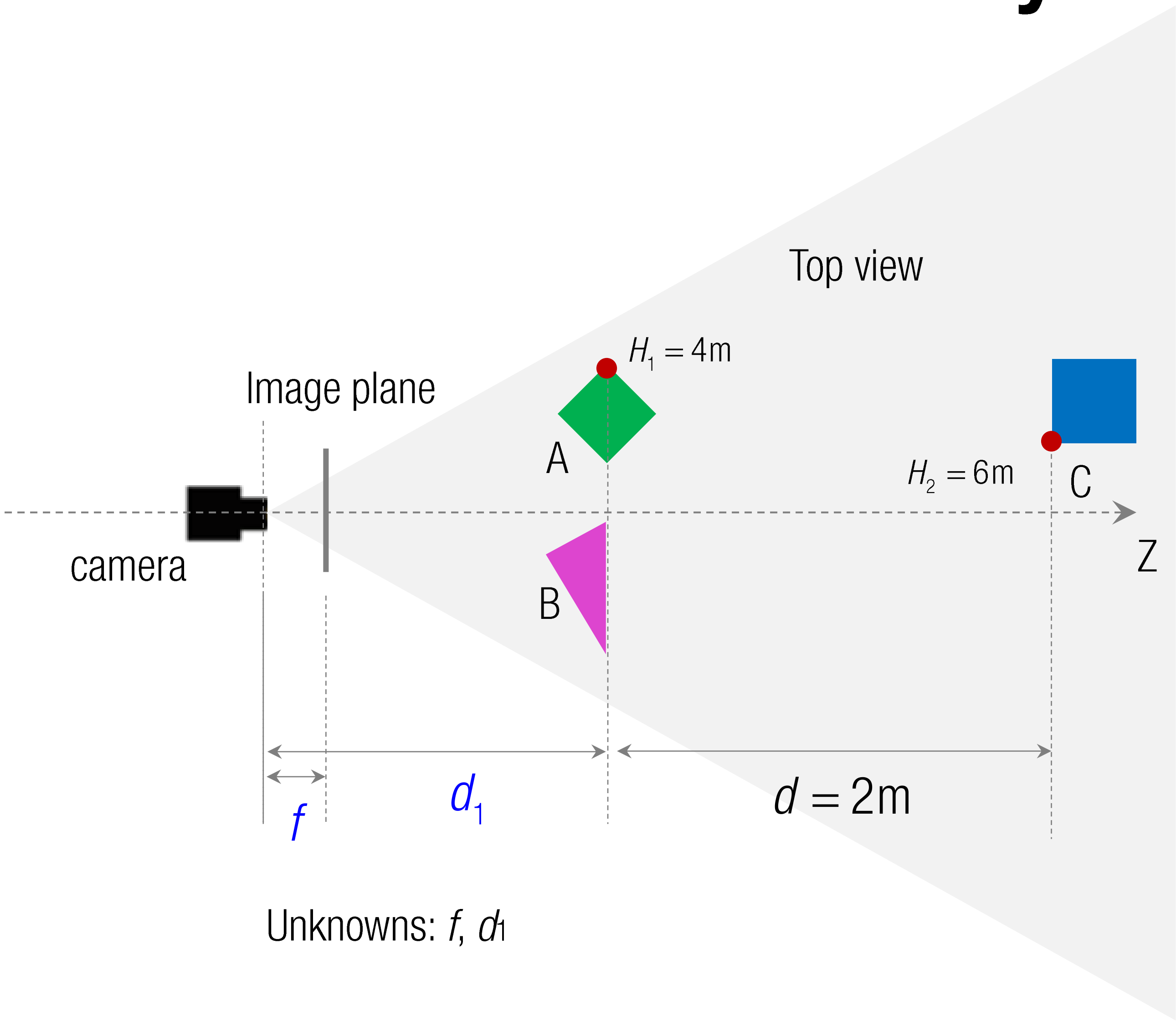




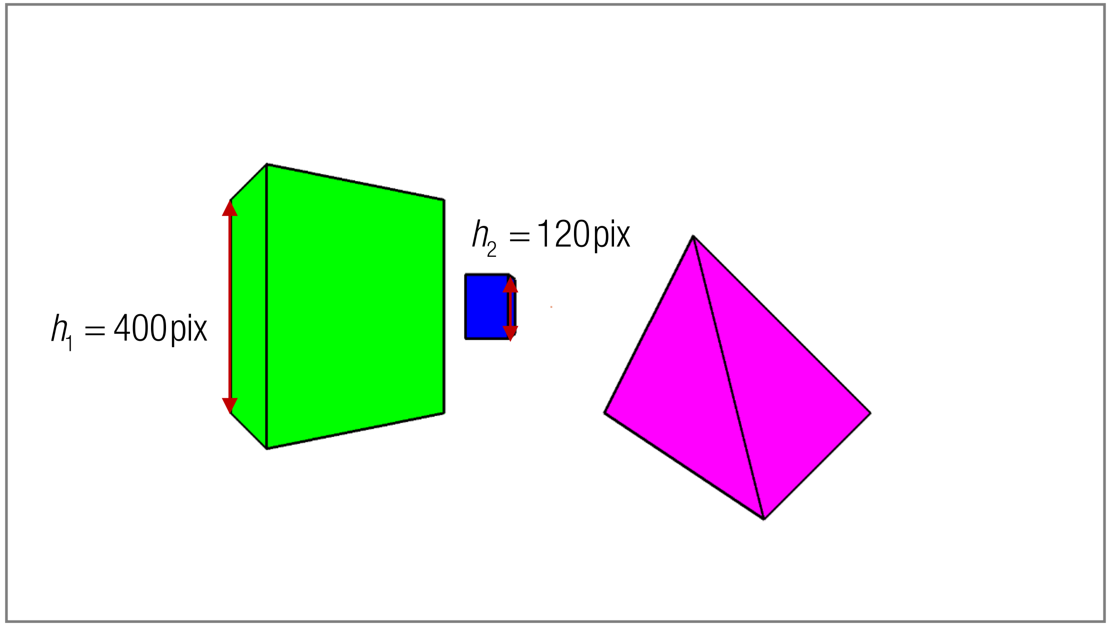
How far I need to step back with zoom factor x2?

How will h_2 change?

Where am I with Dolly Zoom?



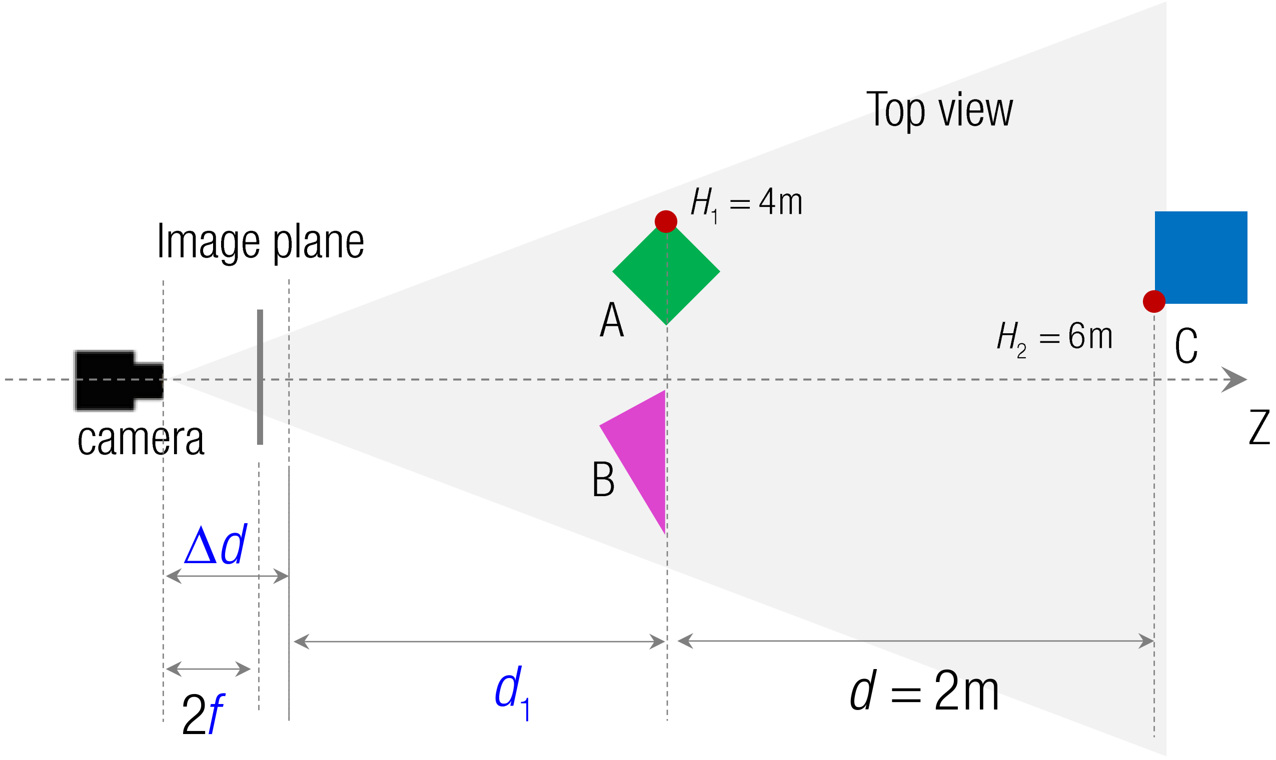
Unknowns: f, d_1

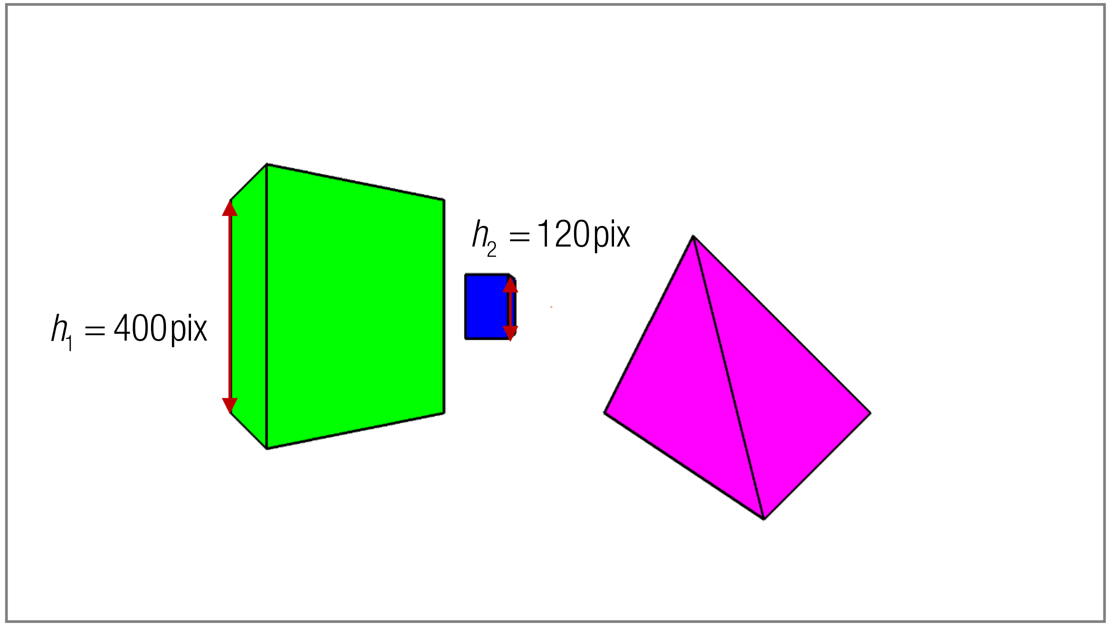


How far I need to step back with zoom factor x2?

How will h_2 change?

Where am I with Dolly Zoom?





Unknowns: f, d_1, Δd

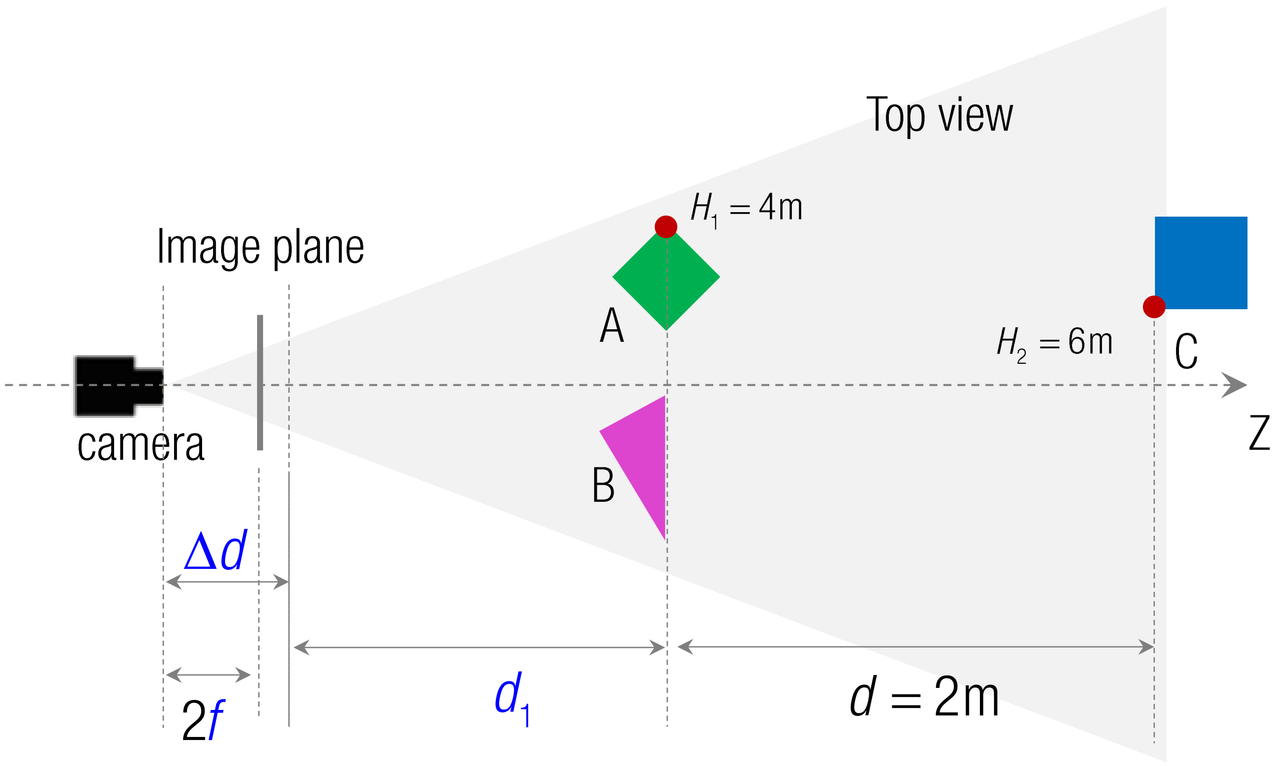
How far I need to step back with zoom factor x2?

How will h_2 change?

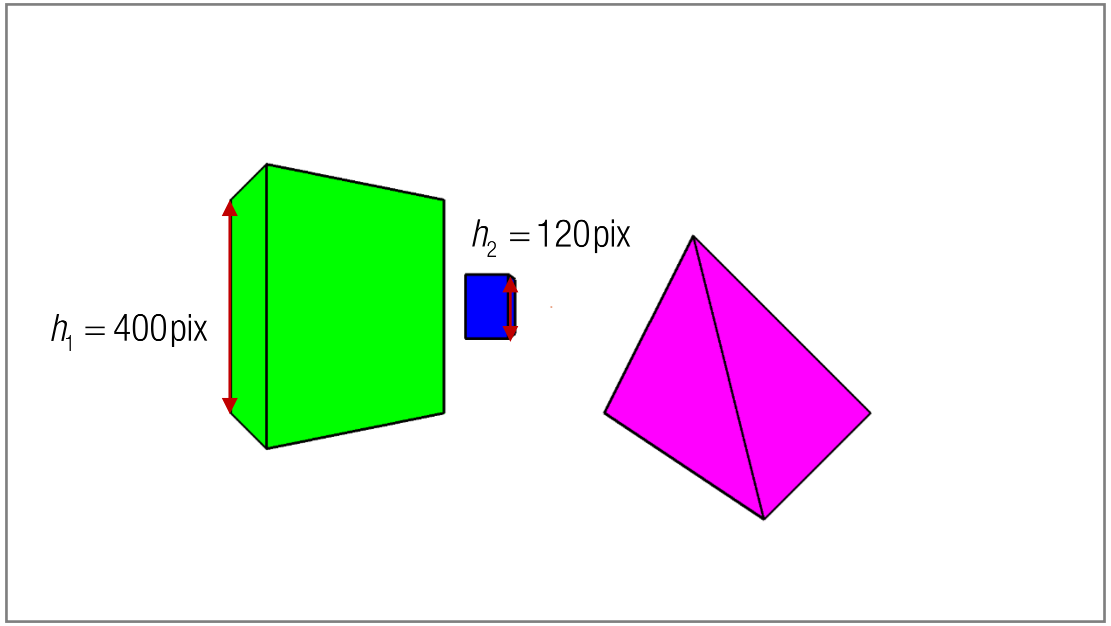
Where am I with Dolly Zoom?

Equations:

$$h_1 = f\frac{H_1}{d_1}$$



Unknowns: f, d_1, Δd



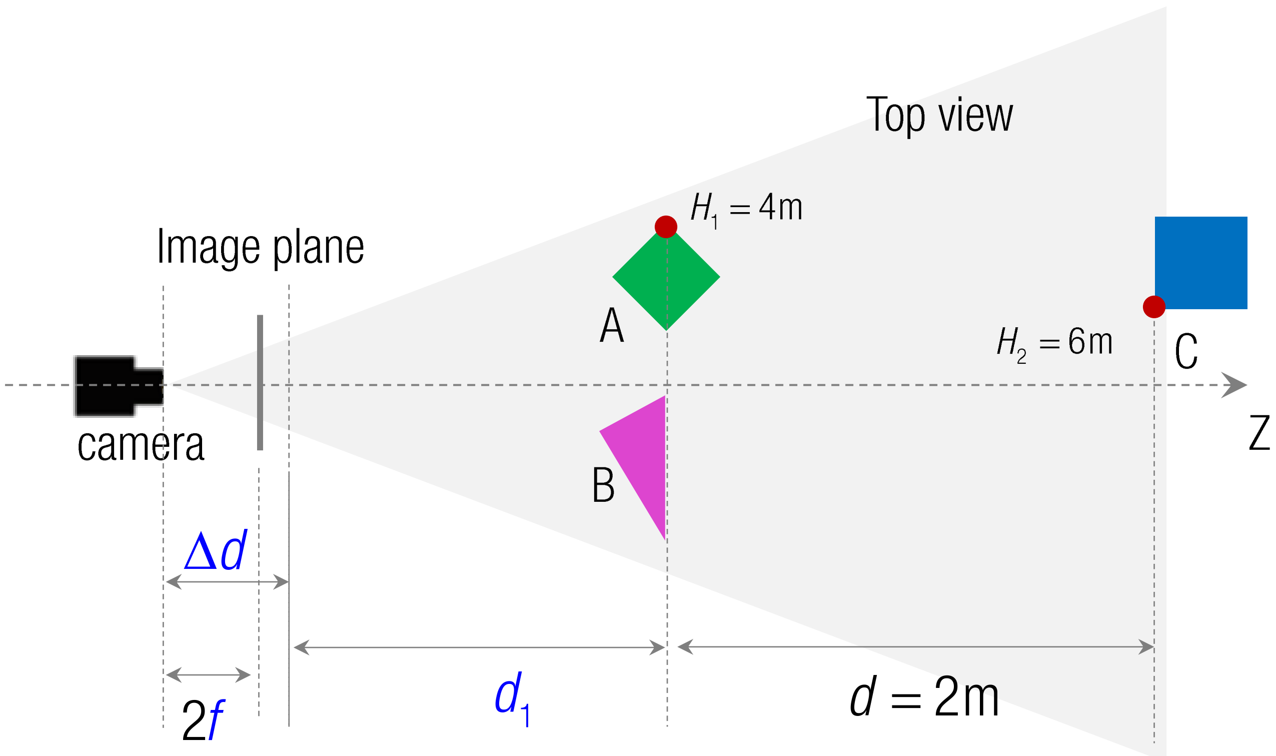
How far I need to step back with zoom factor x2?

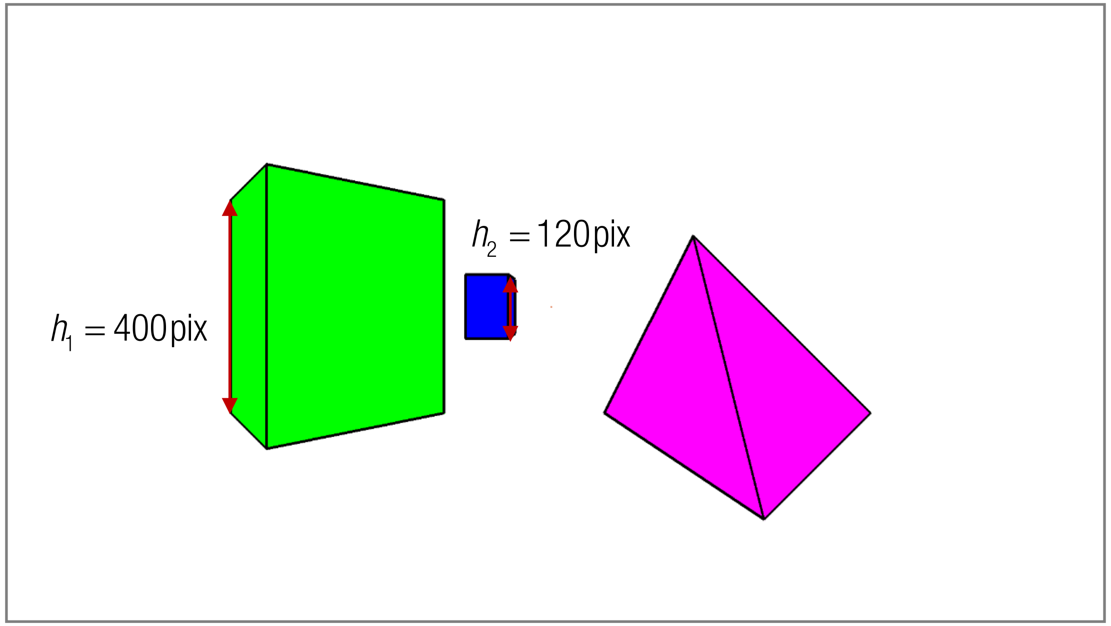
How will h_2 change?

Where am I with Dolly Zoom?

Equations:

$$h_1 = f\frac{H_1}{d_1} \qquad h_1 = 2f\frac{H_1}{\Delta d + d_1}$$





Unknowns: f, d_1, Δd

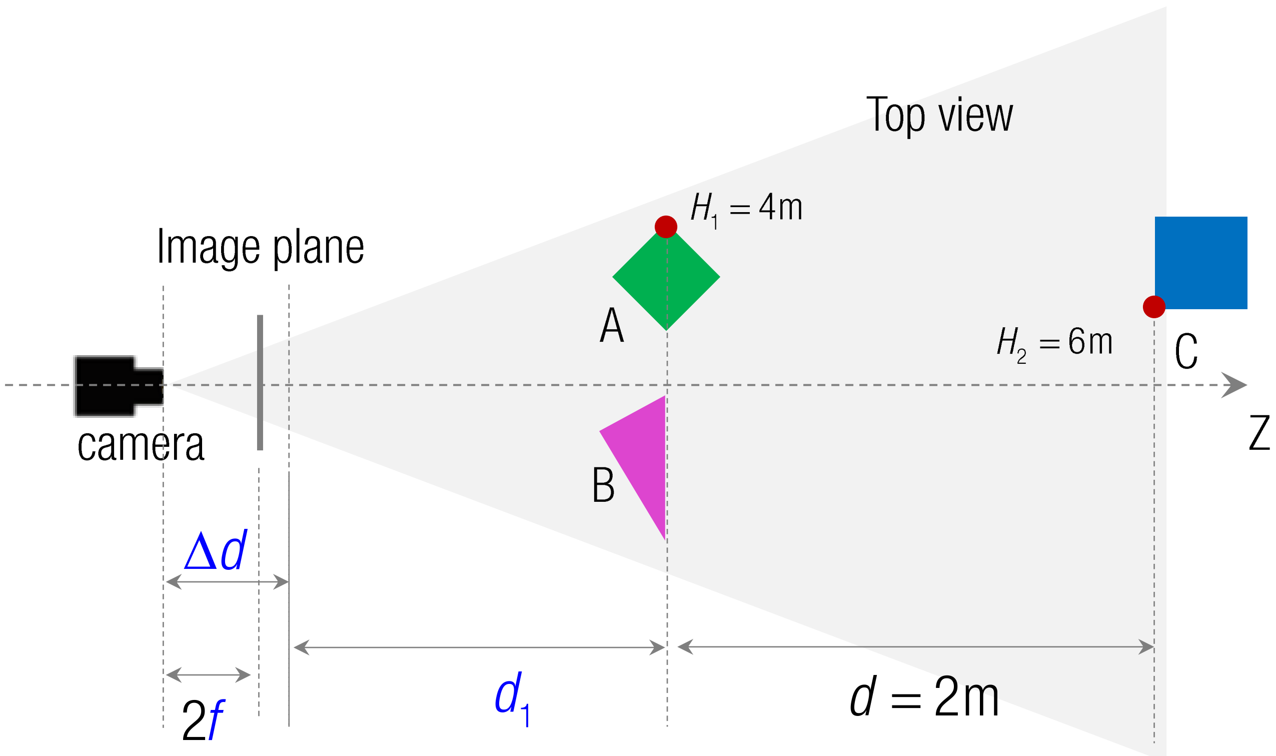
How far I need to step back with zoom factor x2?

How will h_2 change?

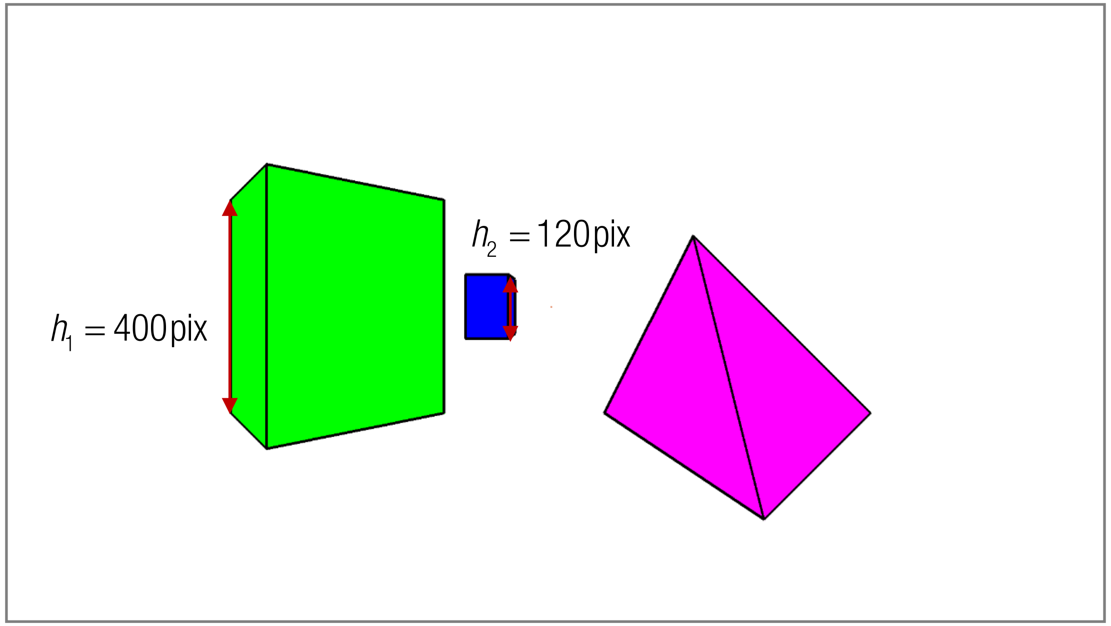
Where am I with Dolly Zoom?

Equations:

$$h_1 = f\frac{H_1}{d_1} \qquad h_1 = 2f\frac{H_1}{\Delta d + d_1} \longrightarrow \Delta d = d_1$$



Unknowns: f, d_1, Δd



How far I need to step back with zoom factor x2?

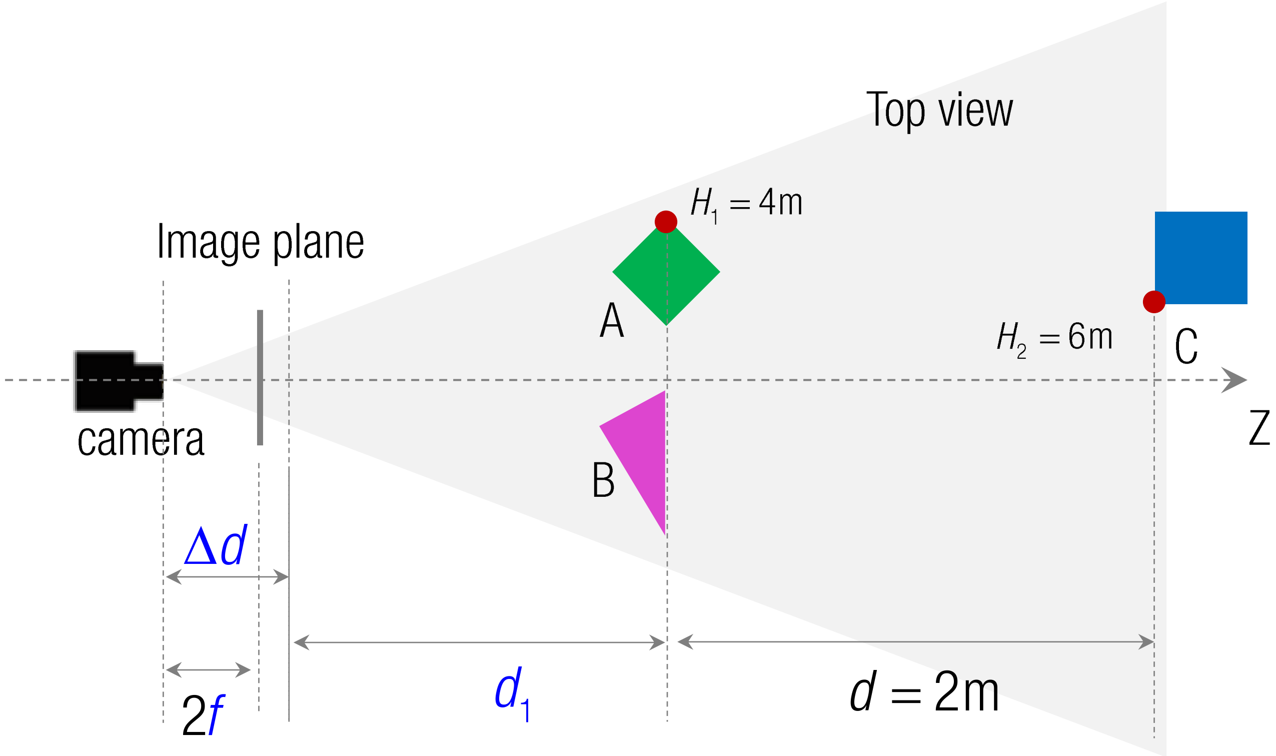
How will h_2 change?

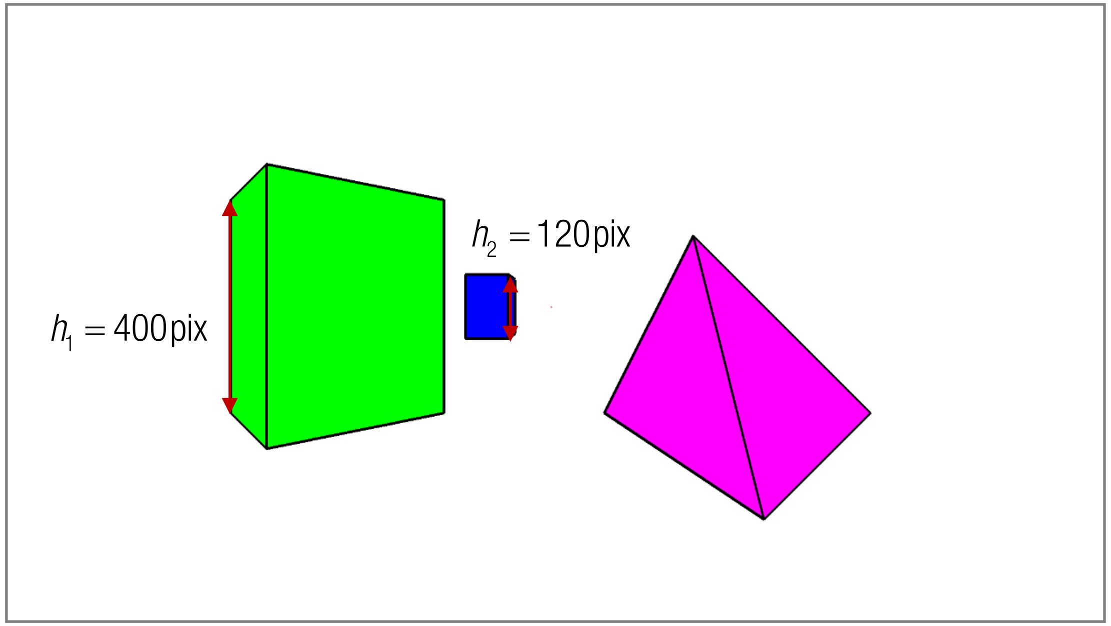
Where am I with Dolly Zoom?

Equations:

$$h_1 = f\frac{H_1}{d_1} \qquad h_1 = 2f\frac{H_1}{\Delta d + d_1} \longrightarrow \Delta d = d_1$$

$$h_2 = f\frac{H_2}{d_1 + d}$$





Unknowns: f, d_1, Δd

How far I need to step back with zoom factor x2?

How will h_2 change?

Where am I with Dolly Zoom?

Equations:

$$h_1 = f\frac{H_1}{d_1}$$

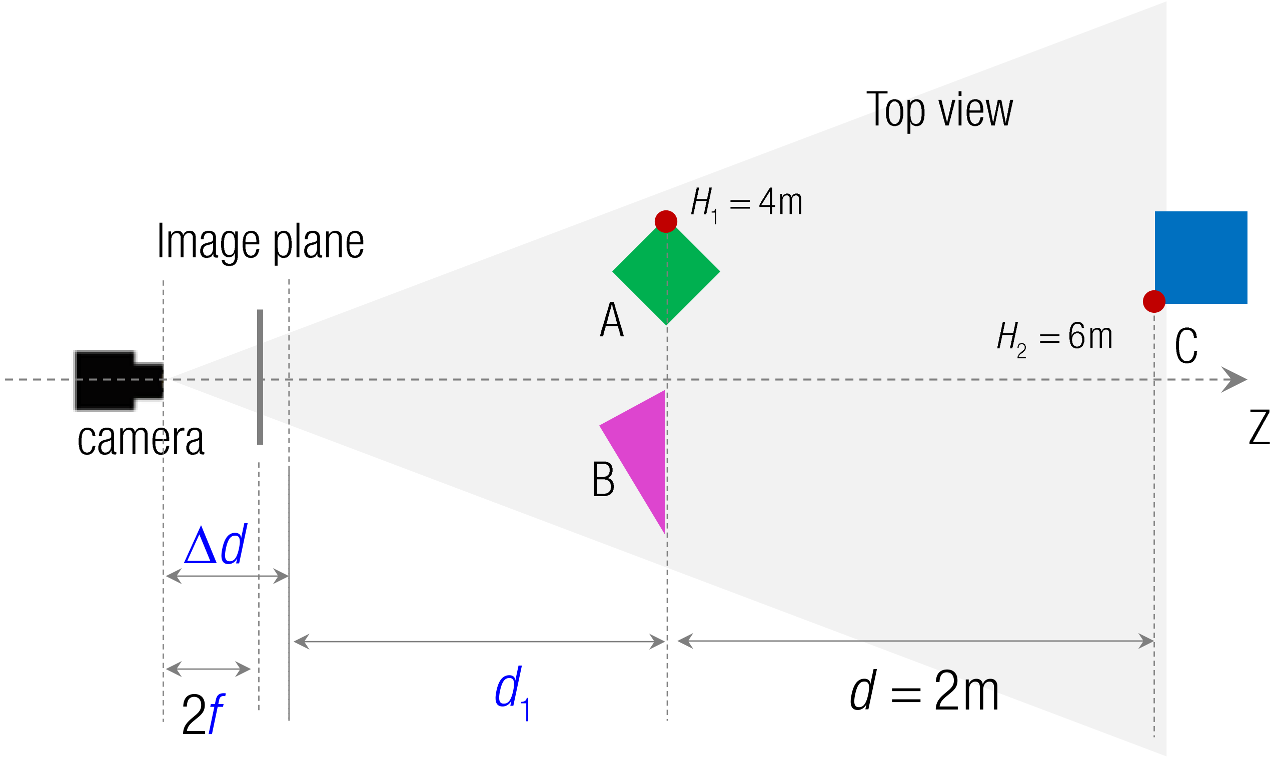
$$h_2 = f\frac{H_2}{d_1 + d}$$

$$\downarrow$$

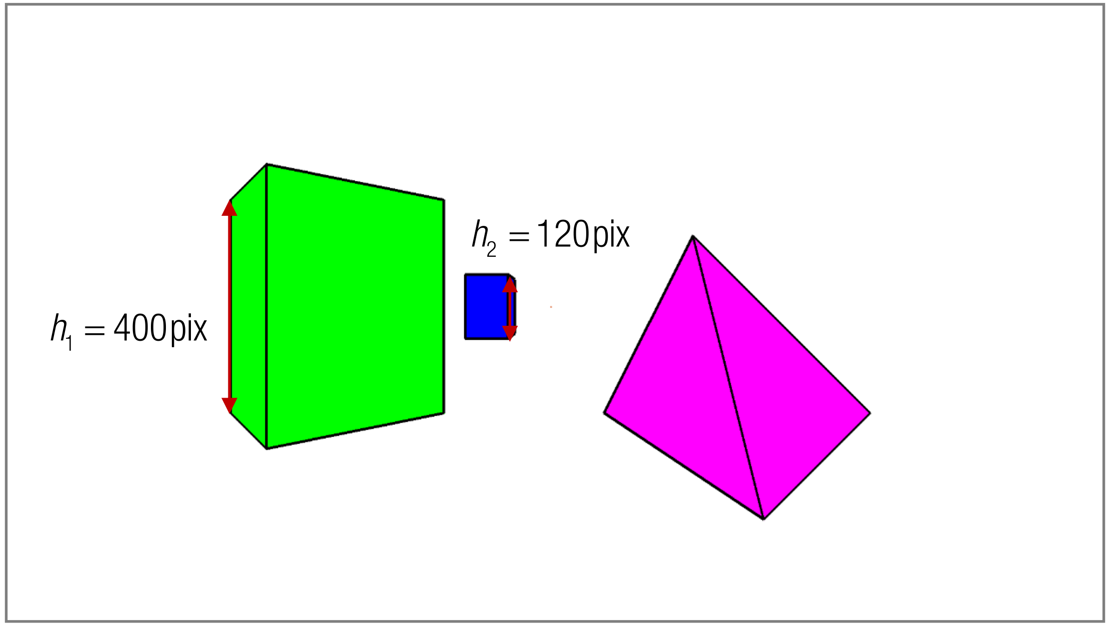
$$d_1 = \frac{1}{1 - \frac{h_2 H_1}{h_1 H_2}} d = 2.5\text{m}$$

$$h_1 = 2f\frac{H_1}{\Delta d + d_1} \longrightarrow \Delta d = d_1$$

$$\Delta d = 2.5\text{m}$$



Unknowns: f, d_1, Δd



$h_2 = 120\text{pix} \longrightarrow h_2' = 200\text{pix}$

Where am I with Dolly Zoom?

Equations:

$$h_1 = f\frac{H_1}{d_1}$$

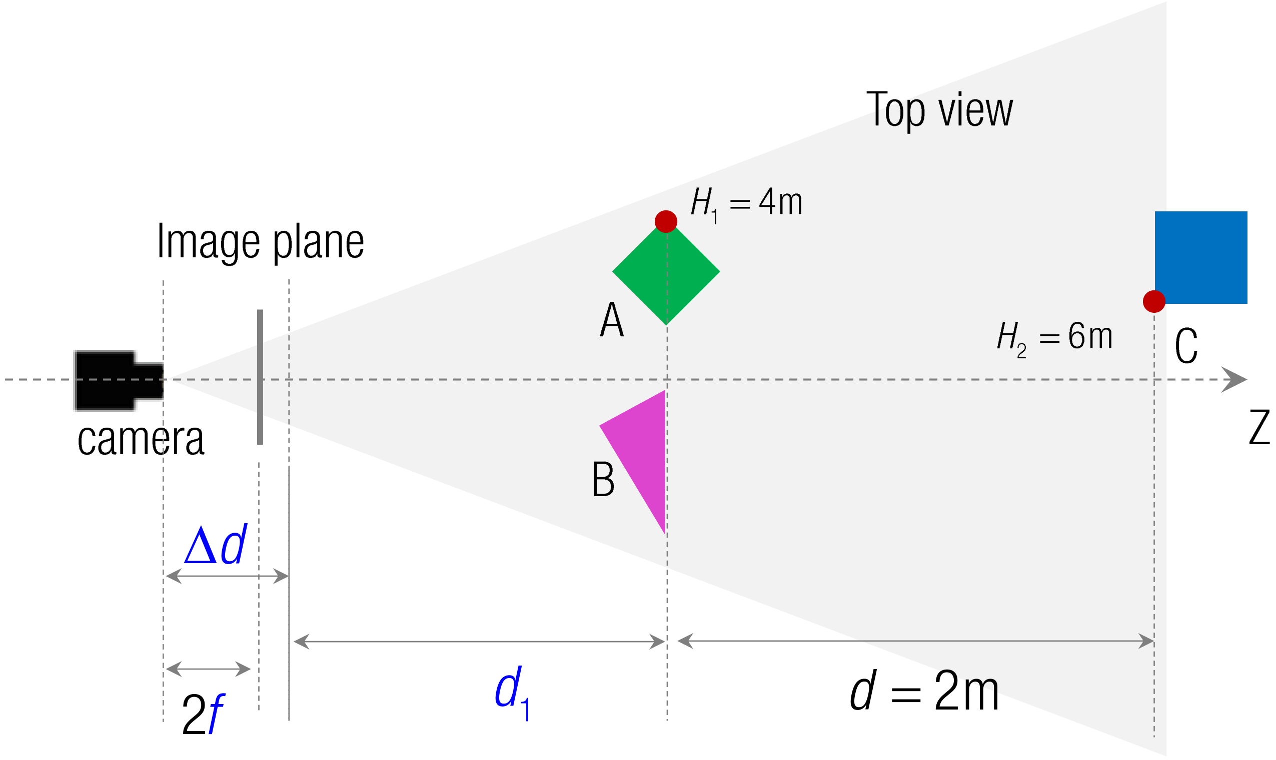
$$h_2 = f\frac{H_2}{d_1 + d}$$

$$\downarrow$$

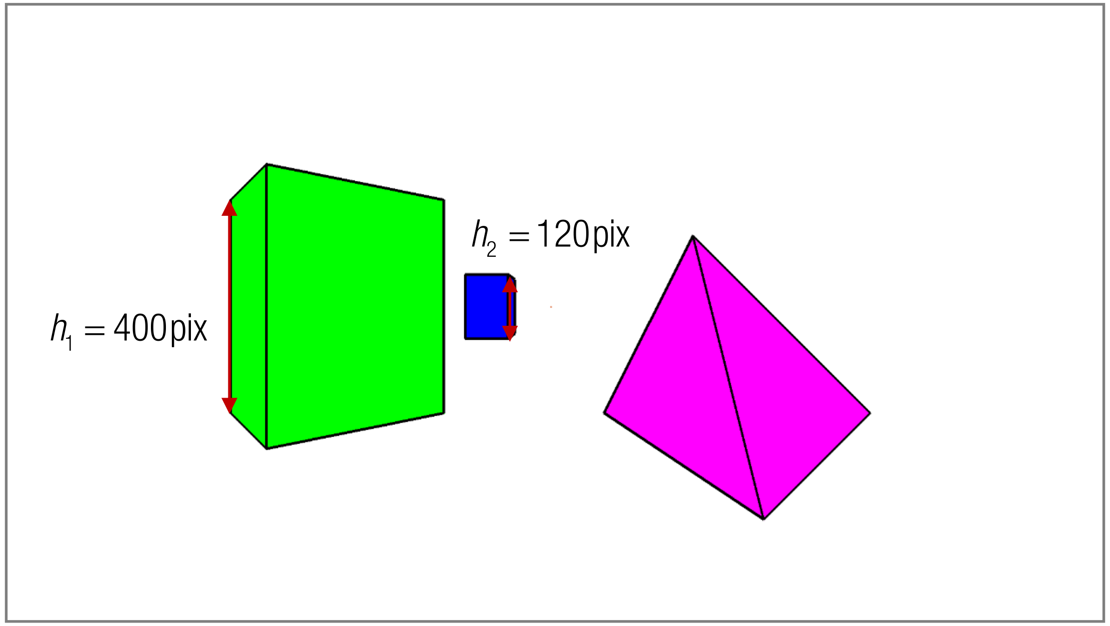
$$d_1 = \frac{1}{1-\frac{h_2 H_1}{h_1 H_2}} d = 2.5\text{m}$$

$$h_1 = 2f\frac{H_1}{\Delta d + d_1} \longrightarrow \Delta d = d_1$$

$$\Delta d = 2.5\text{m} \qquad f = 250\text{pix}$$



Unknowns: f, d_1, Δd



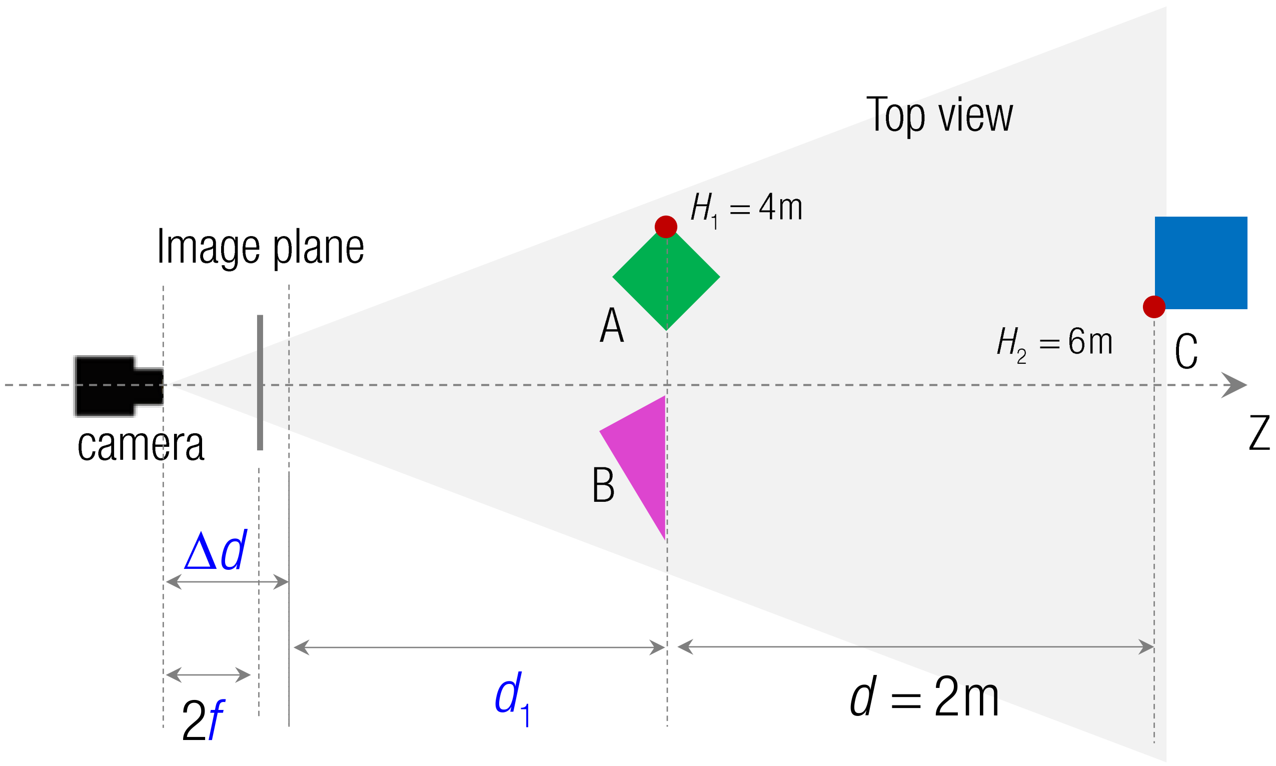
$$h_2 = 120\text{pix} \longrightarrow h_2' = 200\text{pix}$$

Where am I with Dolly Zoom?

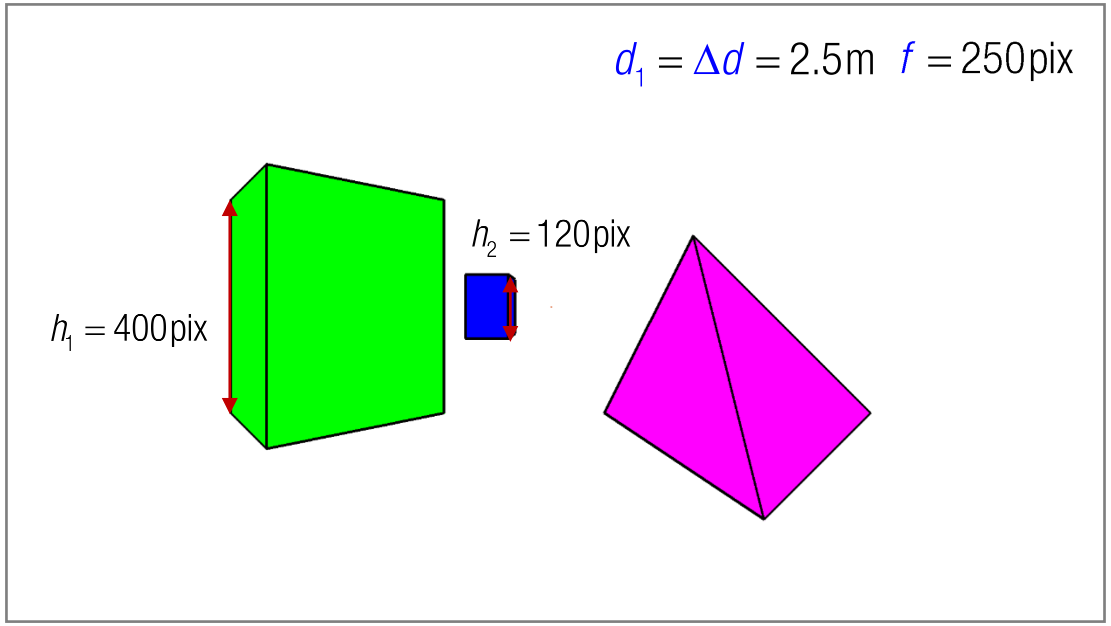
Equations:

$$h_1 = f\frac{H_1}{d_1} \qquad h_1 = 2f\frac{H_1}{\Delta d + d_1}$$

$$h_2 = f\frac{H_2}{d_1 + d} \qquad h_2' = 2f\frac{H_2}{\Delta d + d_1 + d}$$



Unknowns: f, d_1, Δd



How far I need to step back with zoom factor x2?

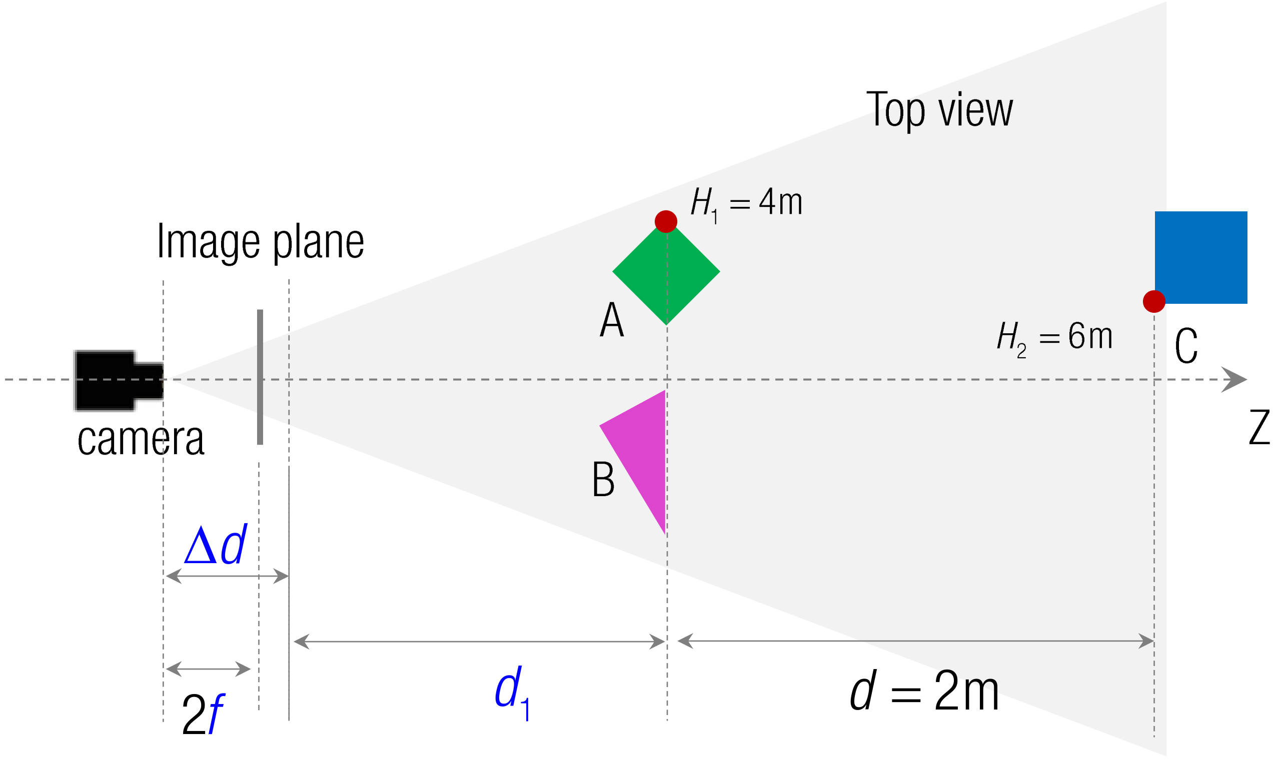
How will h_2 change?

Where am I with Dolly Zoom?

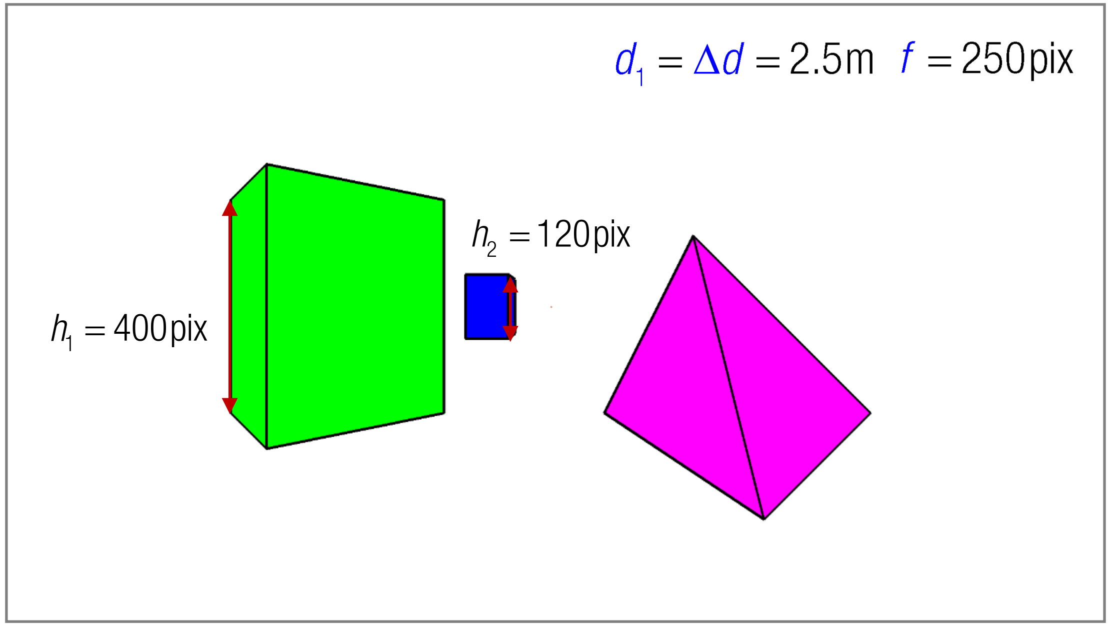
Equations:

$$h_1 = f\frac{H_1}{d_1} \qquad h_1 = 2f\frac{H_1}{\Delta d + d_1}$$

$$h_2 = f\frac{H_2}{d_1 + d} \qquad h_2' = 2f\frac{H_2}{\Delta d + d_1 + d} = 429\text{pix}$$



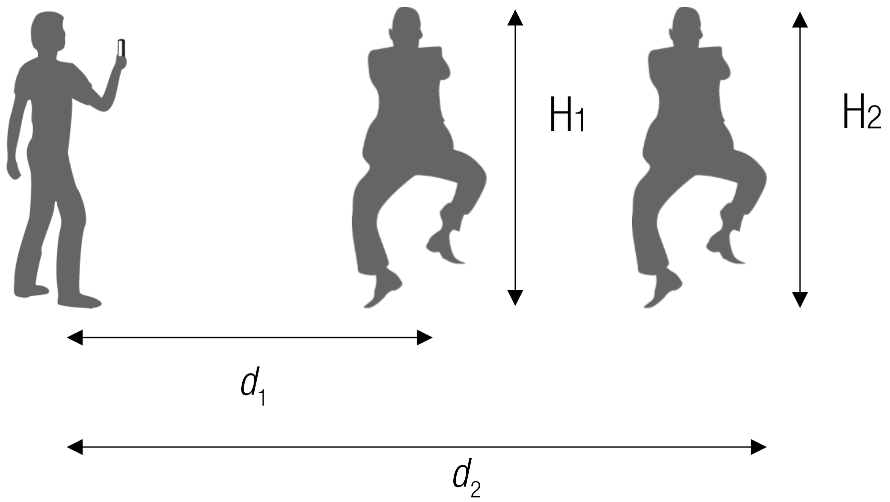
Unknowns: f, d_1, Δd

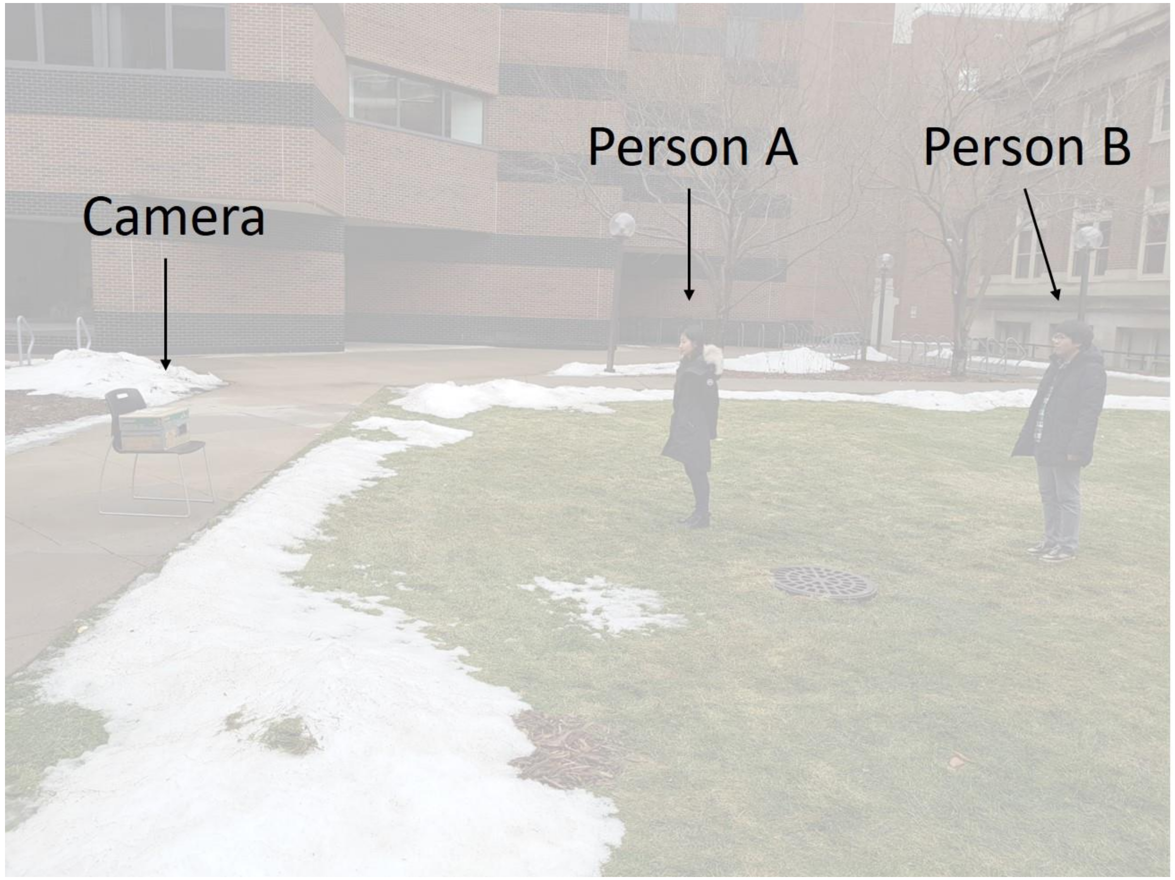


How far I need to step back with zoom factor x2?

How will h_2 change?

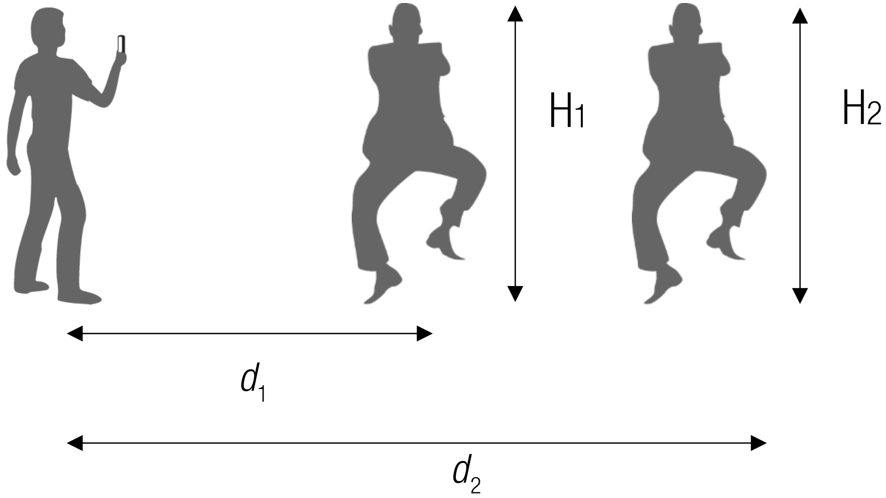
HW #1 Camera Obscura and Dolly Zoom



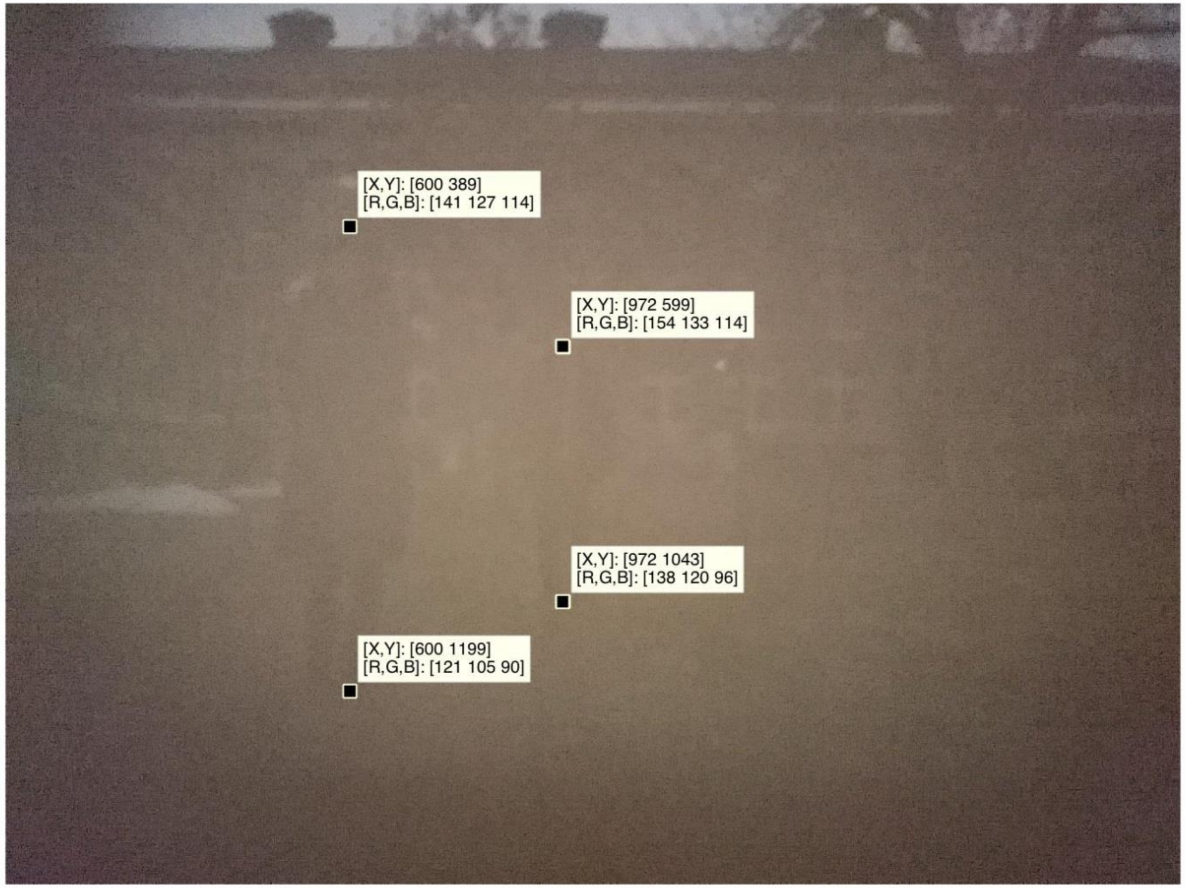


1) Take a photo of two persons

HW #1 Camera Obscura and Dolly Zoom

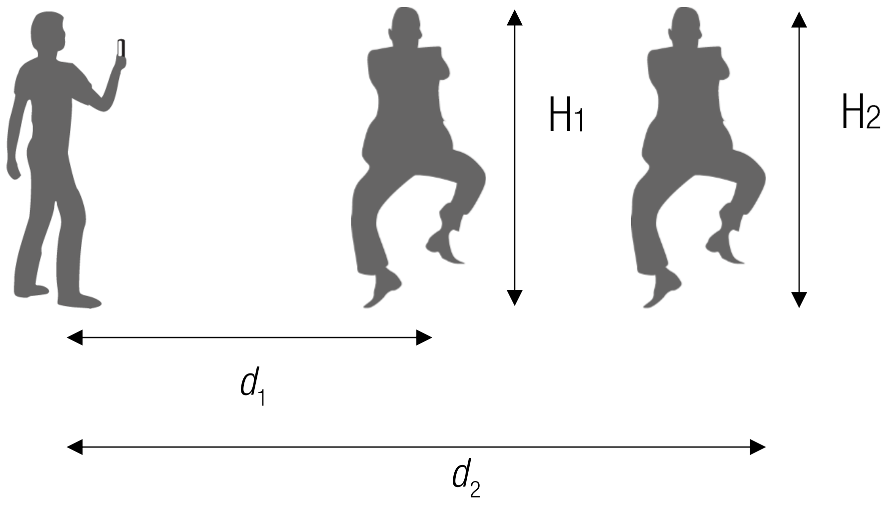


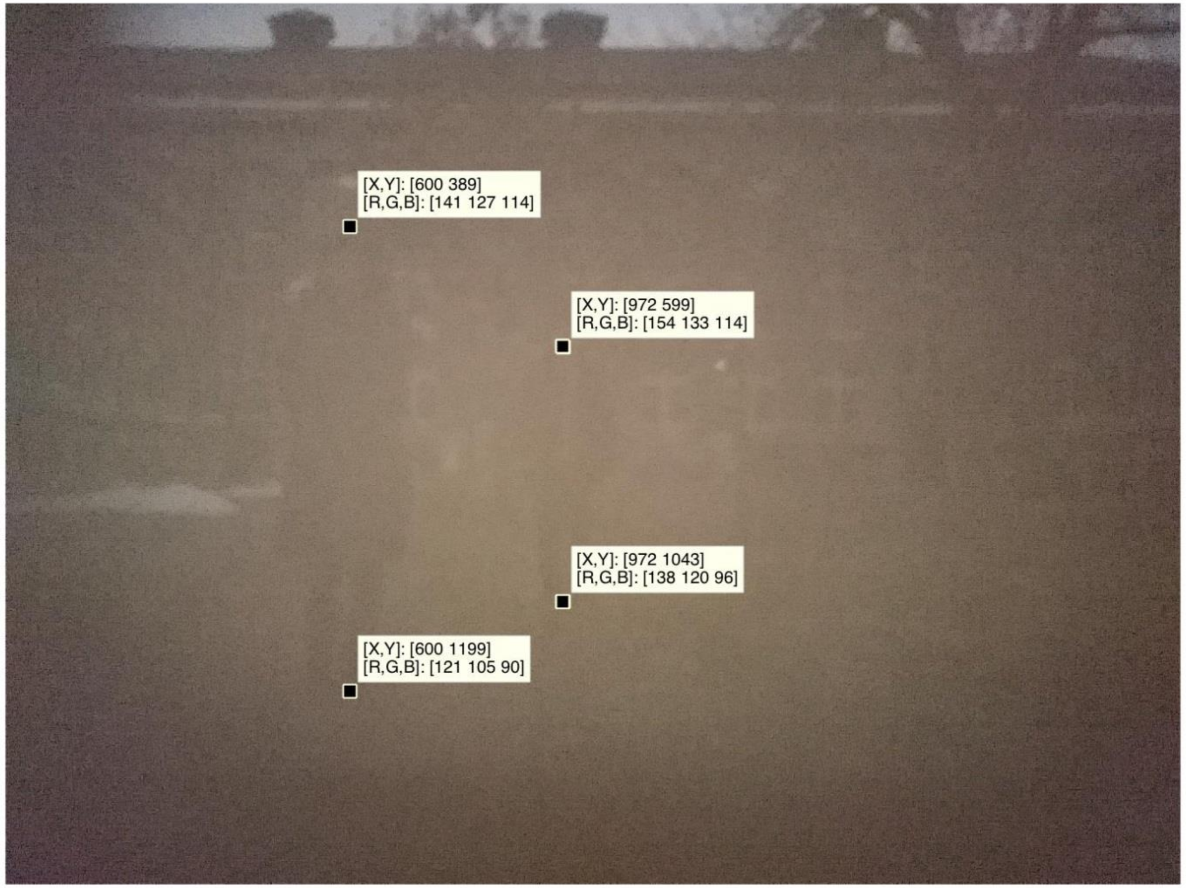
1) Take a photo of two persons



$f = 2017.43\text{pix} \quad H_1 = 1.6\text{m}$

HW #1 Camera Obscura and Dolly Zoom

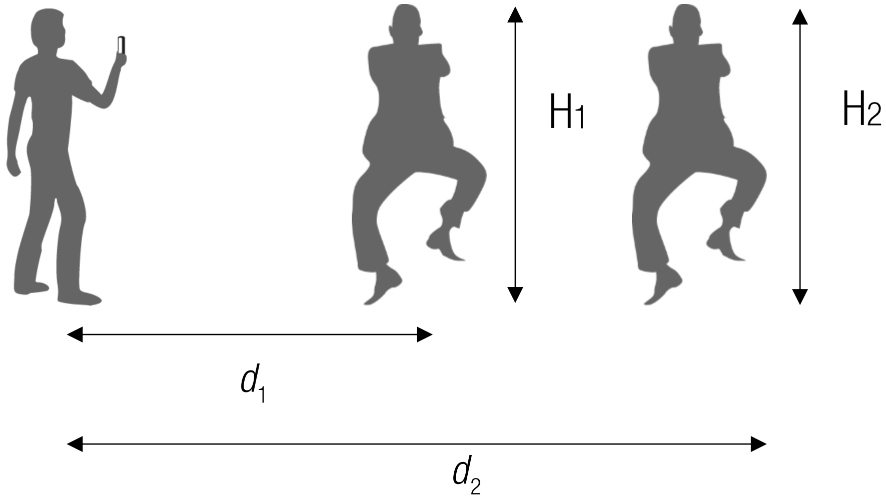


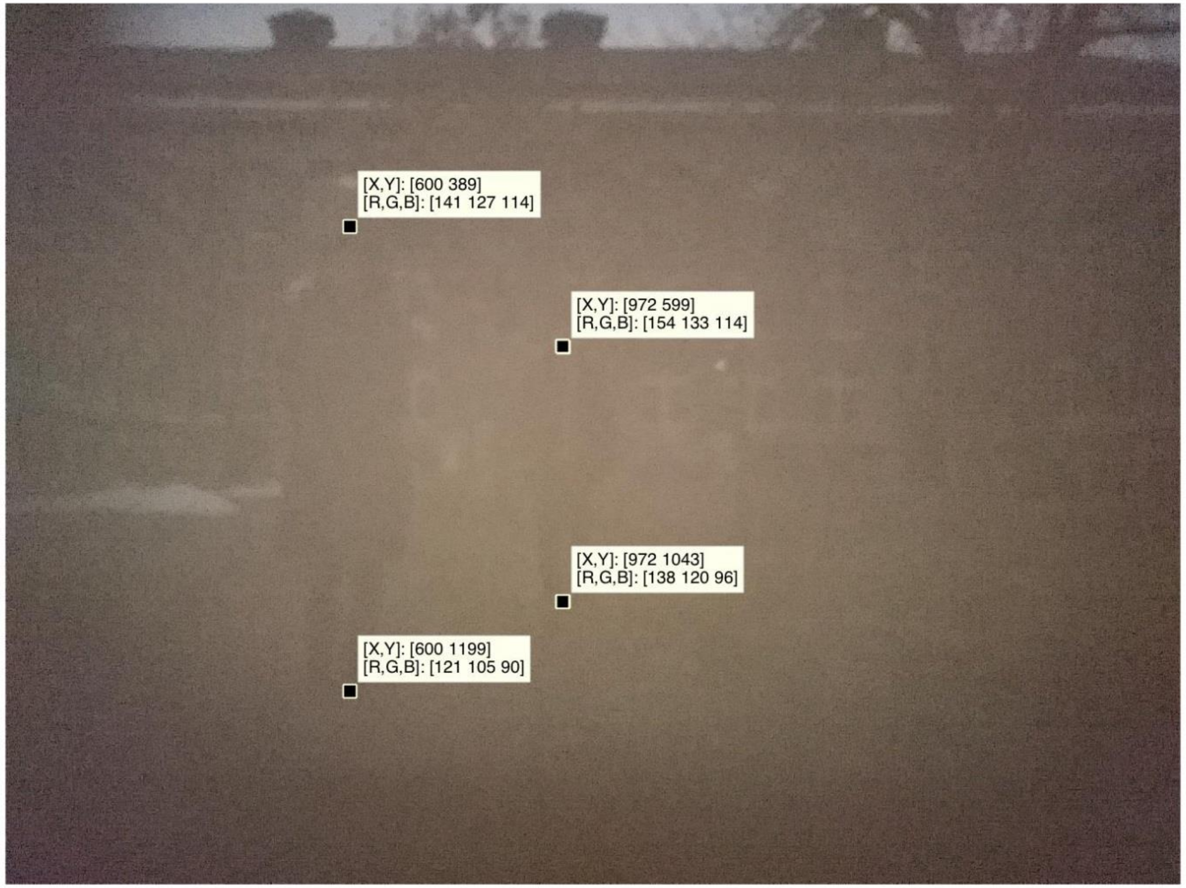


1) Take a photo of two persons

$f = 2017.43\text{pix}$ $H_1 = 1.6\text{m}$ $d_1 = 3.99\text{m}$

HW #1 Camera Obscura and Dolly Zoom

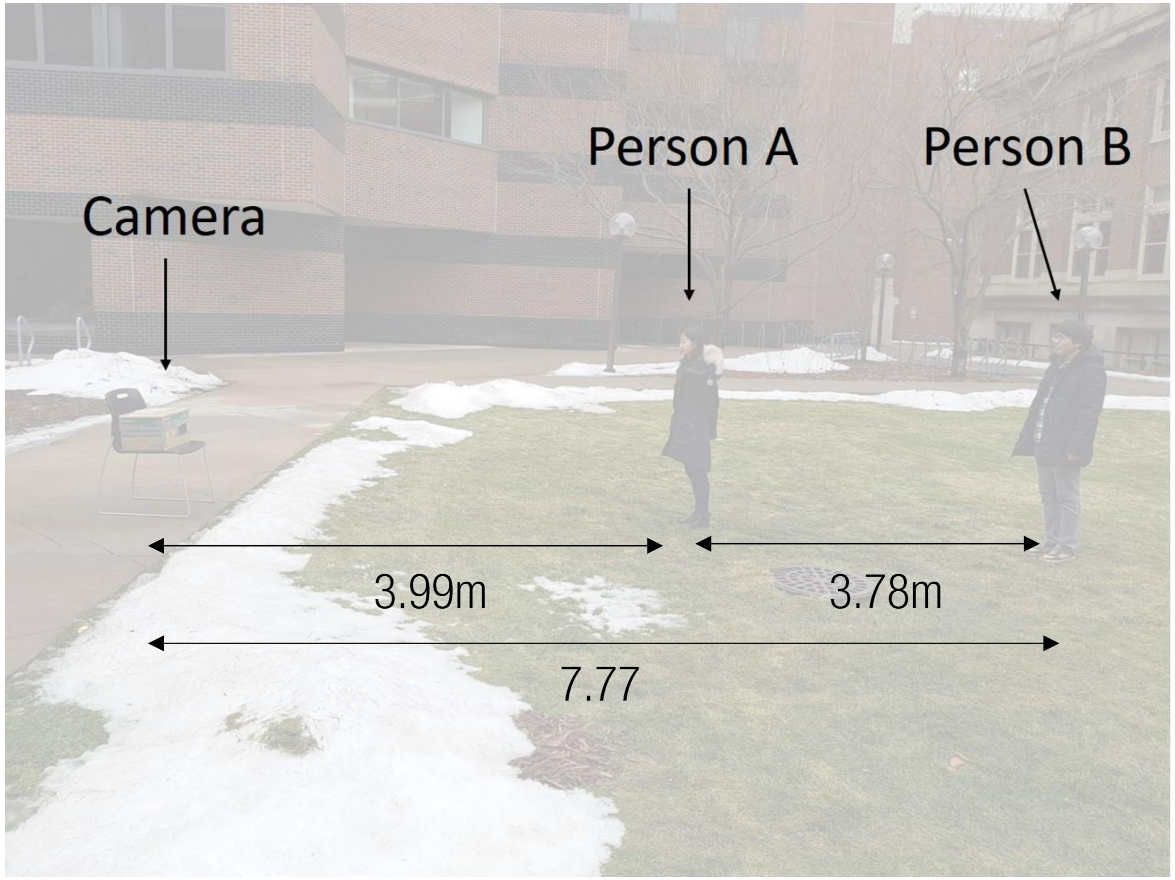


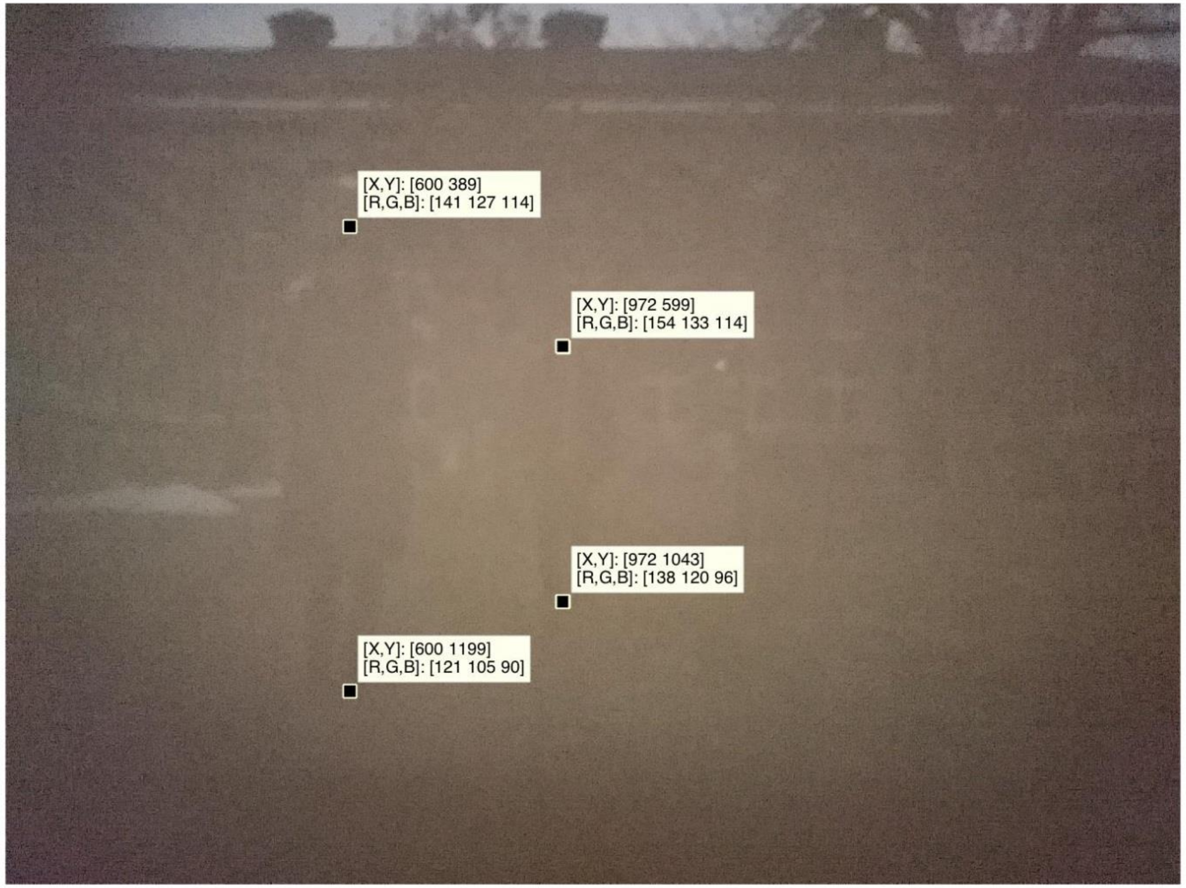


1) Take a photo of two persons

$f = 2017.43\text{pix} \quad H_2 = 1.71\text{m}$

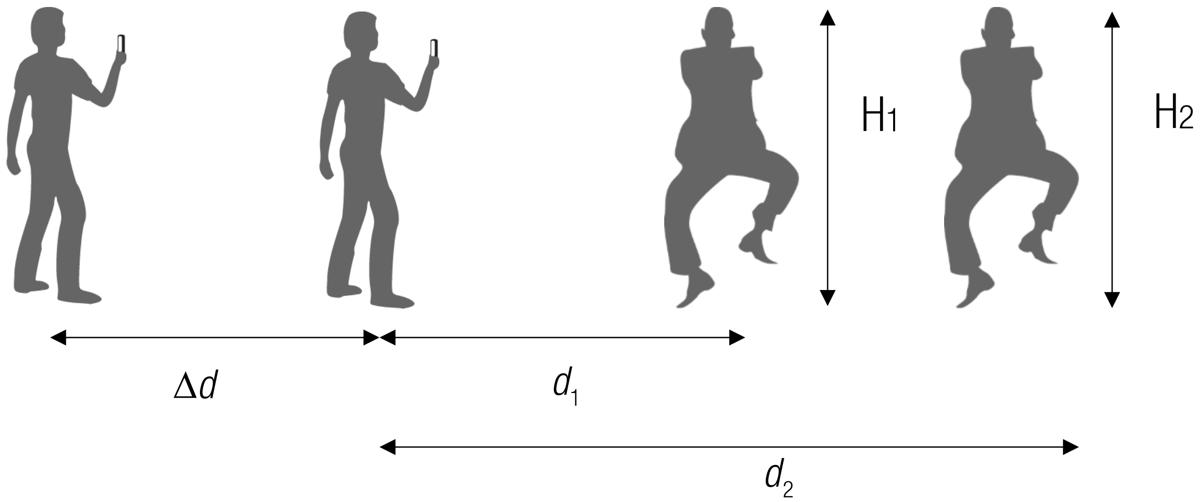
HW #1 Camera Obscura and Dolly Zoom



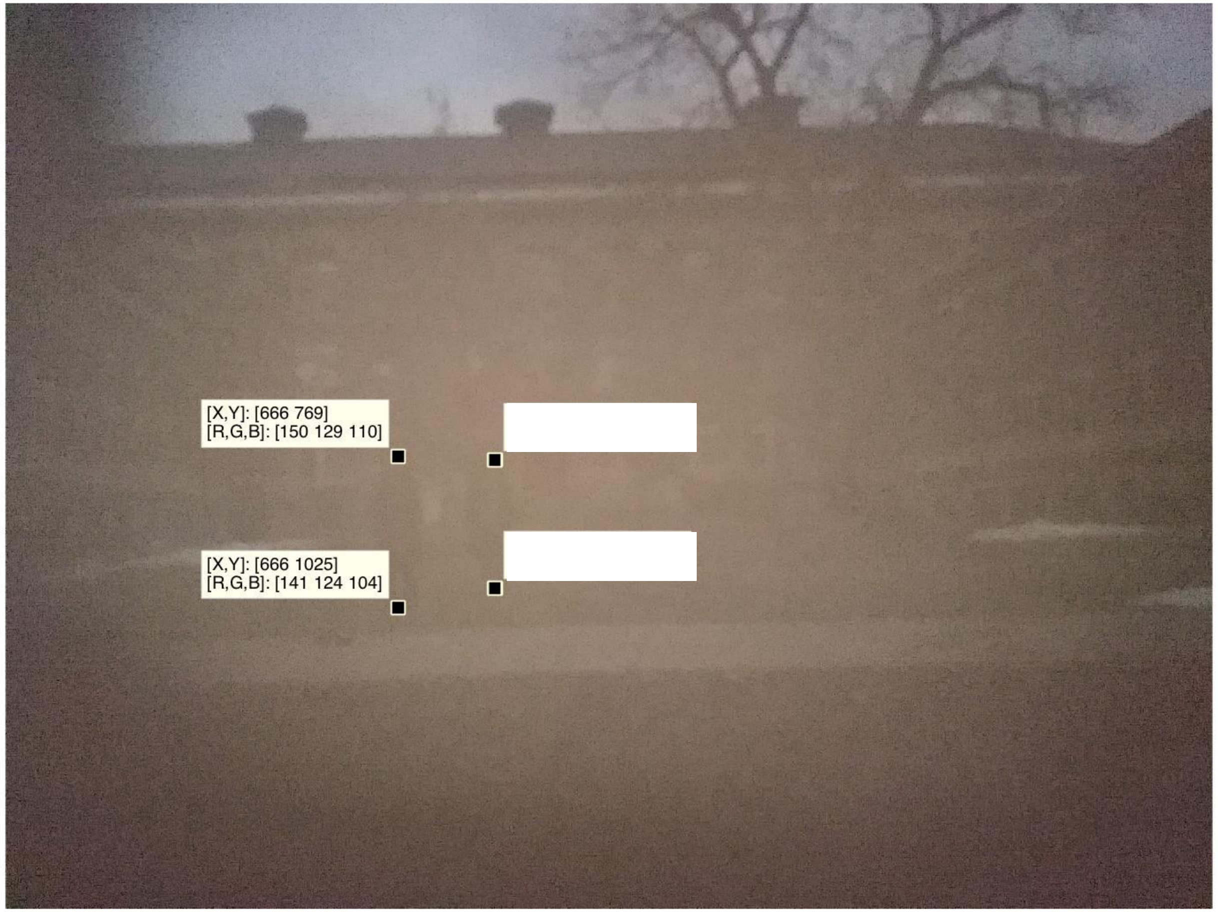


$f = 2017.43\text{pix}$ $H_2 = 1.71\text{m}$ $d_2 = 7.77\text{m}$

HW #1 Camera Obscura and Dolly Zoom

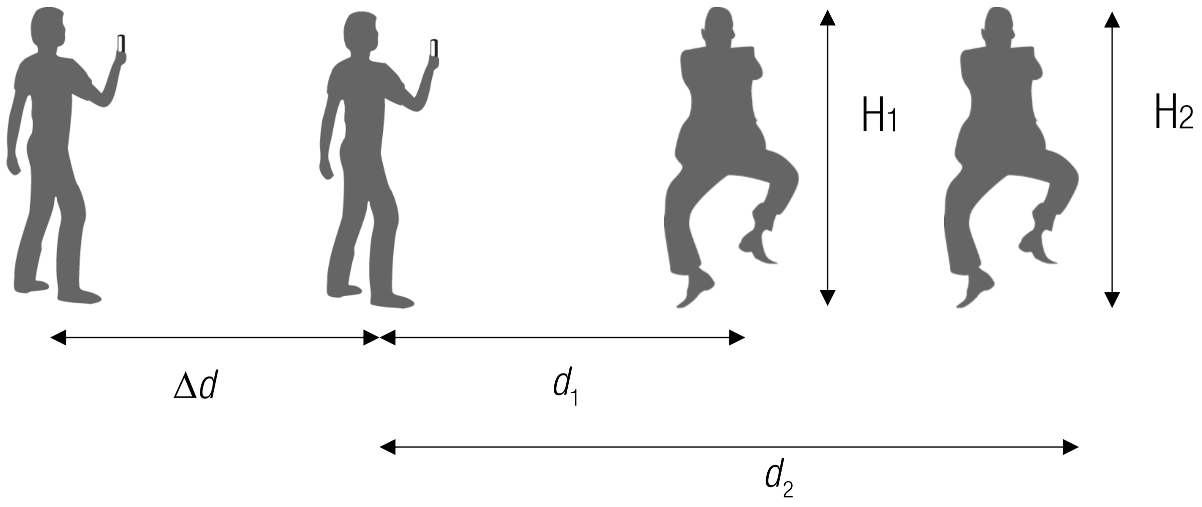


1) Take a photo of two persons
2) Take another photo of them after moving back

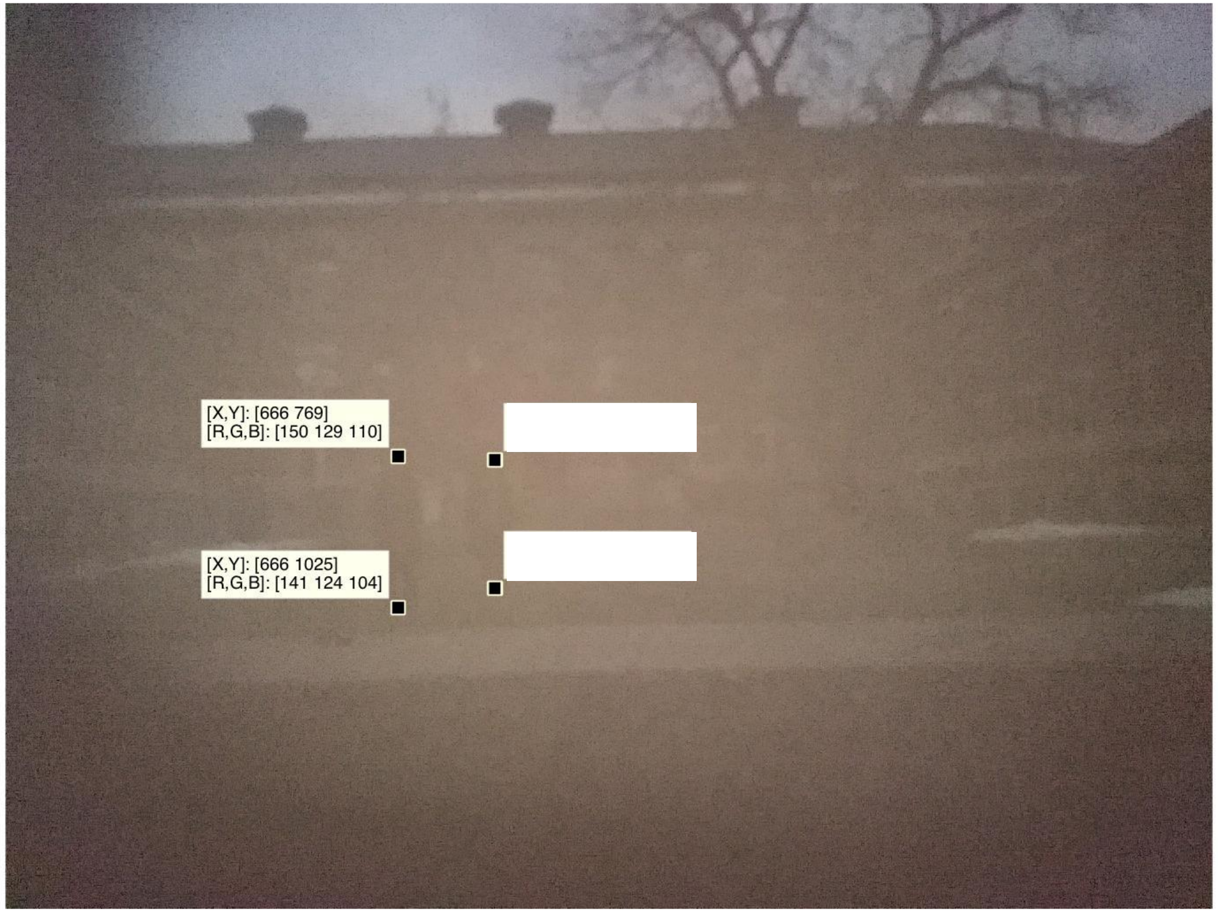


$f = 2017.43\text{pix} \quad H_1 = 1.6\text{m}$

HW #1 Camera Obscura and Dolly Zoom



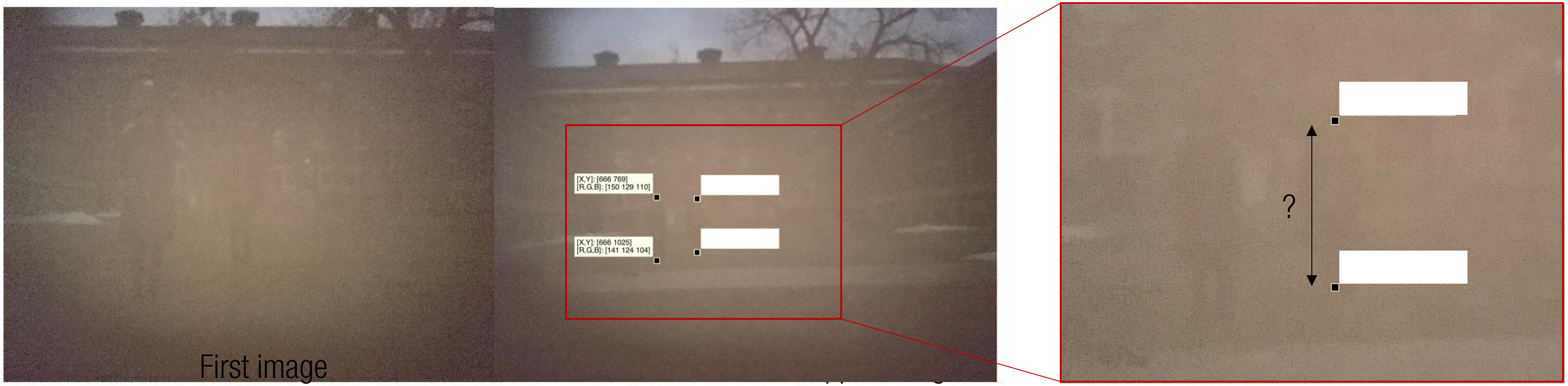
1) Take a photo of two persons
2) Take another photo of them after moving back



$f = 2017.43\text{pix}$ $\quad H_1 = 1.6\text{m}$ $\quad \Delta d + d_1 = 12.6\text{m}$

$\Delta d + d_2 = 16.38\text{m}$

HW #1 Camera Obscura and Dolly Zoom



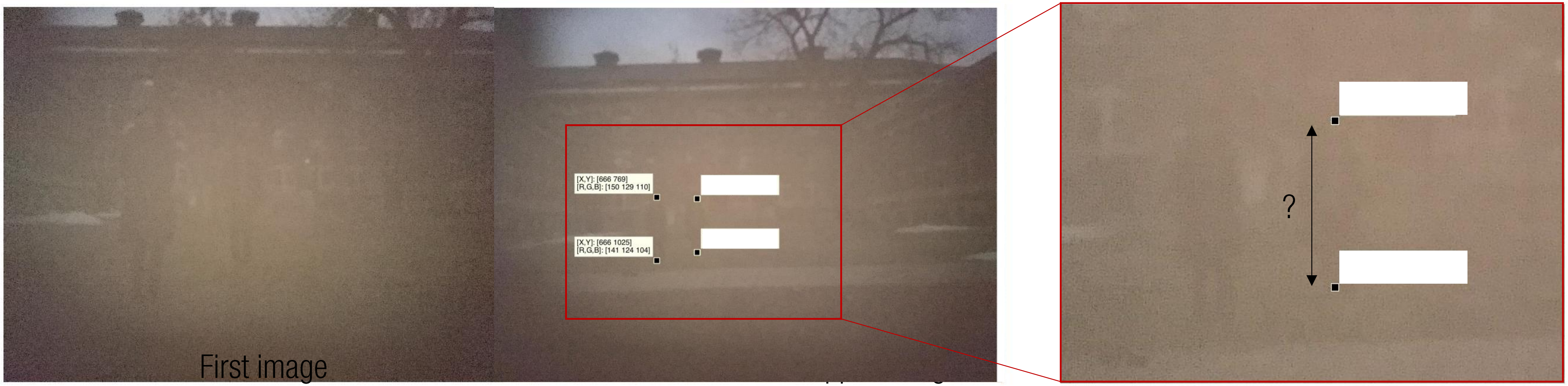
1) Take a photo of two persons
2) Take another photo of them after moving back
3) Scale up and crop the second image such that h_1 remains the same.
4) Predict h_2

$$f_2 = 2017.43\frac{810}{256} = 6383.27\,\text{pix}$$

$$\Delta d + d_2 = 16.38\text{m}$$

$$H_2 = 1.71\text{m}$$

HW #1 Camera Obscura and Dolly Zoom



1) Take a photo of two persons
2) Take another photo of them after moving back
3) Scale up and crop the second image such that h_1 remains the same.
4) Predict h_2

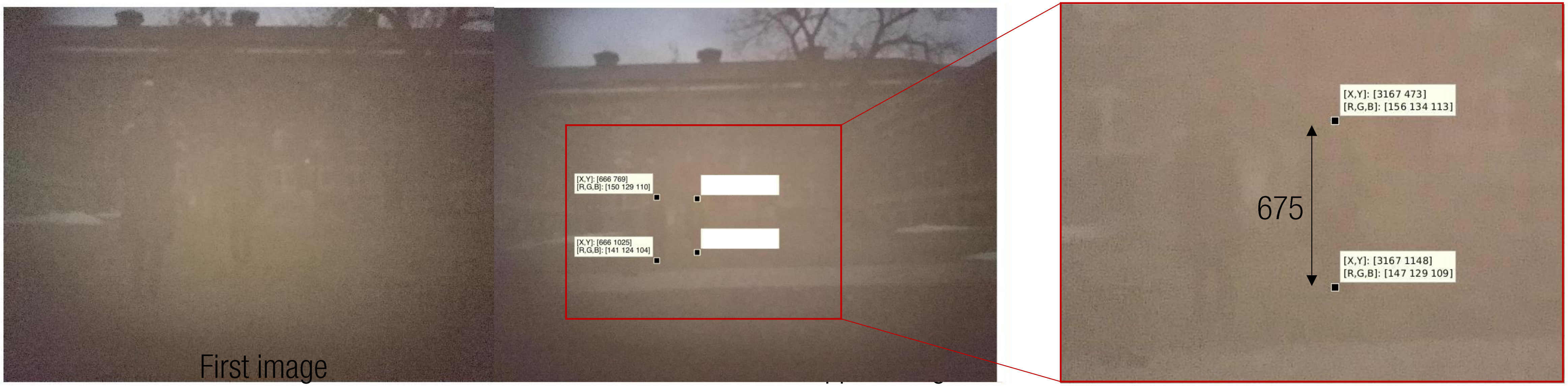
$$f_2 = 2017.43\frac{810}{256} = 6383.27\text{pix}$$

$$\Delta d + d_2 = 16.38\text{m}$$

$$H_2 = 1.71\text{m}$$

$$h_2 = 666.3\text{pix}$$

HW #1 Camera Obscura and Dolly Zoom



1) Take a photo of two persons
2) Take another photo of them after moving back
3) Scale up and crop the second image such that h_1 remains the same.
4) Predict h_2

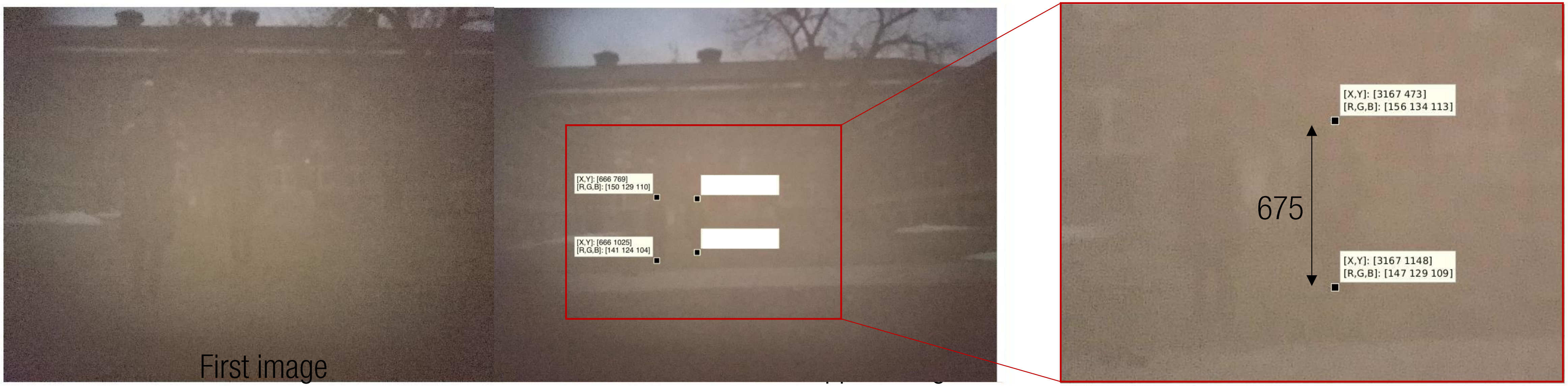
$$f_2 = 2017.43\frac{810}{256} = 6383.27\,\text{pix}$$

$$\Delta d + d_2 = 16.38\text{m}$$

$$H_2 = 1.71\text{m}$$

$$h_2 = 666.3\,\text{pix}$$

HW #1 Camera Obscura and Dolly Zoom



1) Take a photo of two persons
2) Take another photo of them after moving back
3) Scale up and crop the second image such that h_1 remains the same.
4) Predict h_2

$$f_2 = 2017.43\frac{810}{256} = 6383.27\,\text{pix}$$

$$\Delta d + d_2 = 16.38\text{m}$$

$$H_2 = 1.71\text{m}$$

$$h_2 = 666.3\,\text{pix}$$

Camera Projection Matrix

Raw First-person Footage

Lens configuration (internal parameter)

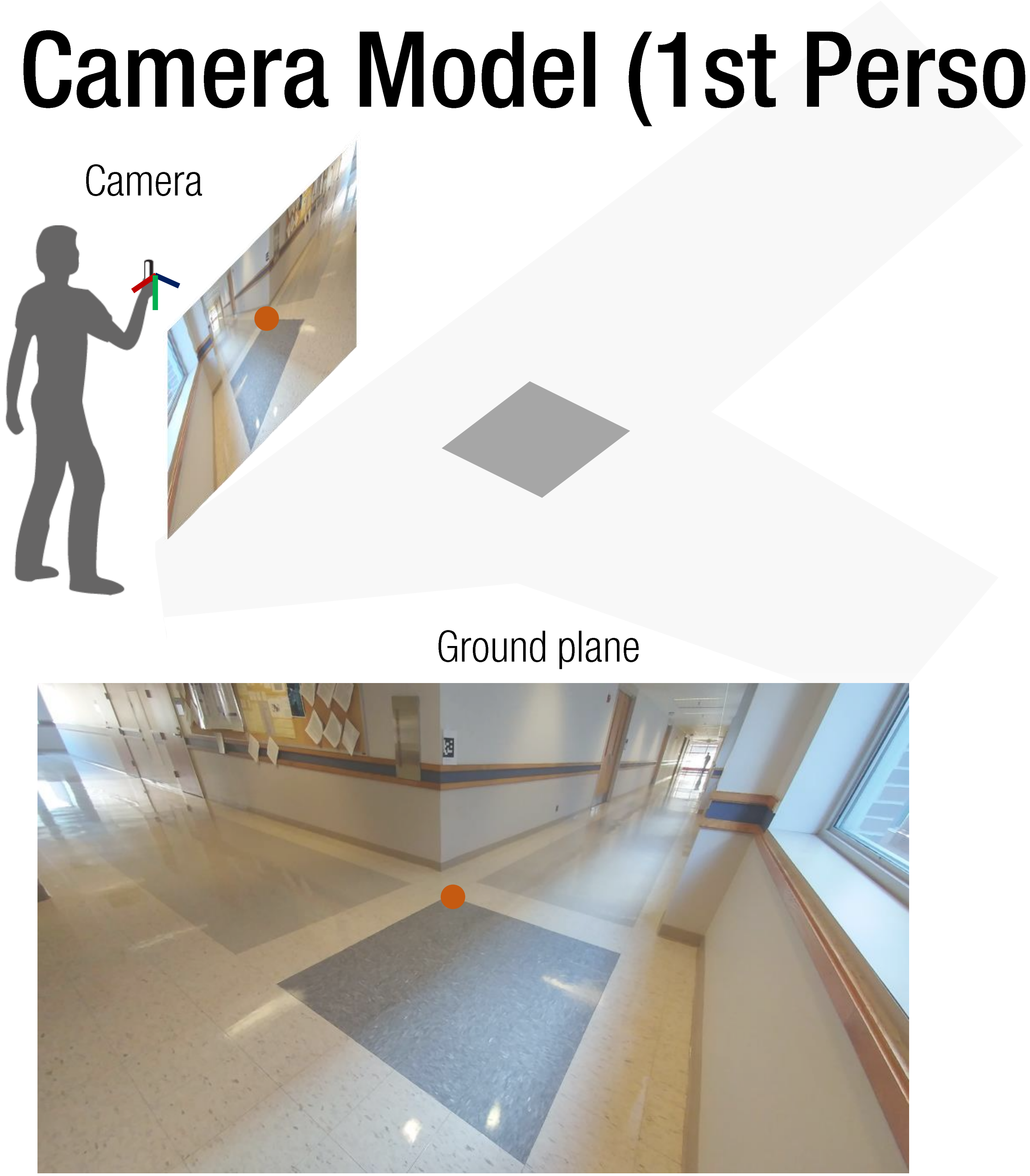
$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = L\left(\mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \right)$$

Spatial relationship between sensor and pinhole
(internal parameter)

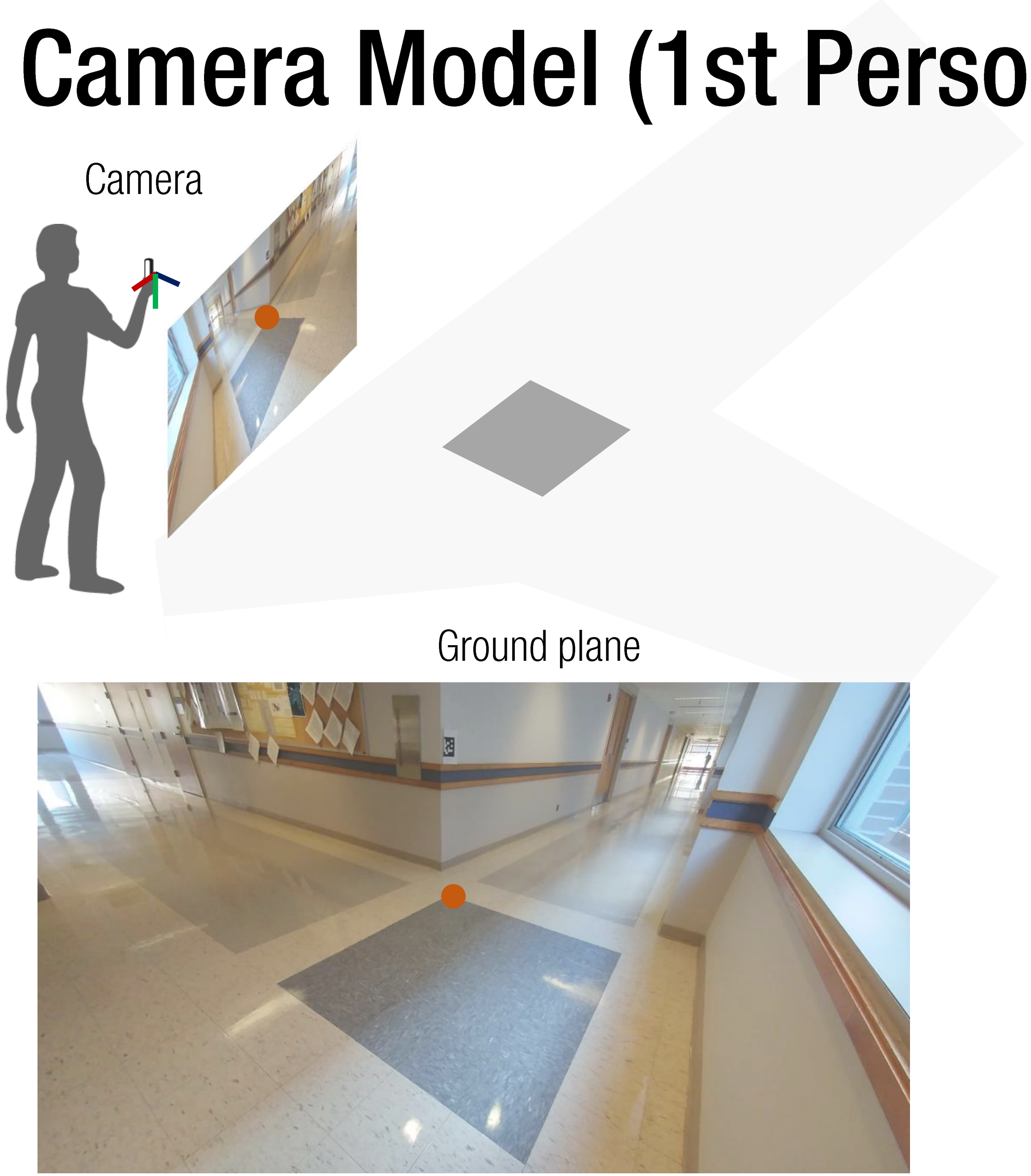
Camera body configuration
(extrinsic parameter)

Camera Model

Camera Model (1st Person Perspective)



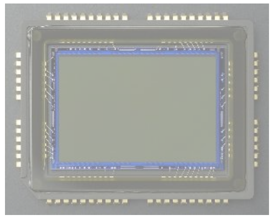
Camera Model (1st Person Perspective)



Recall camera projection matrix:

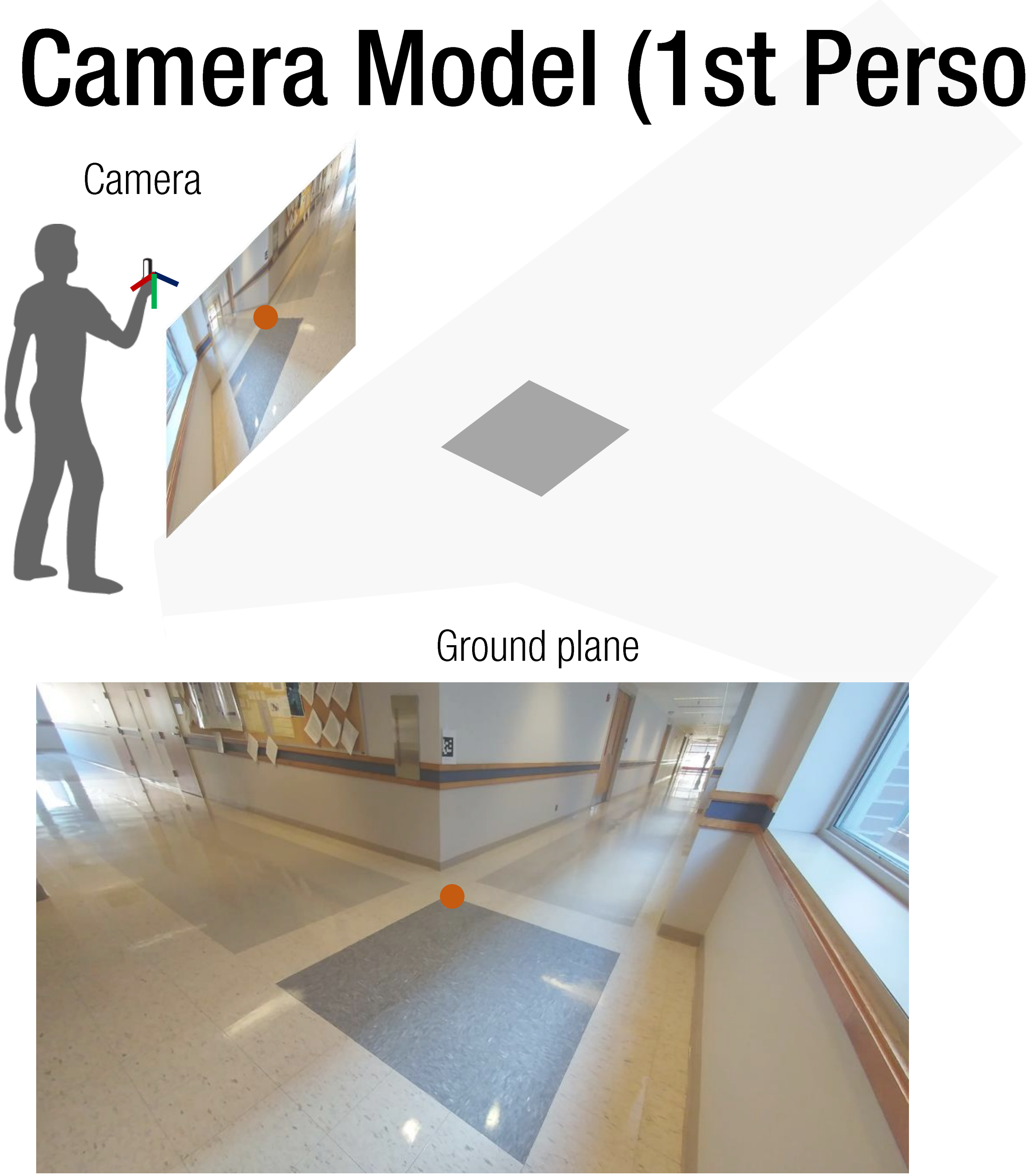
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underline{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

+



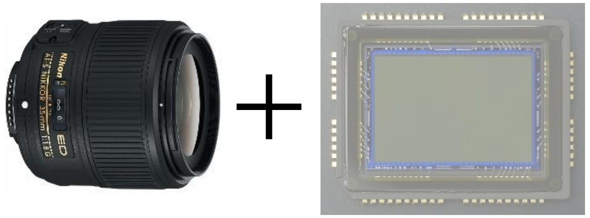
Camera intrinsic parameter
: metric space to pixel space

Camera Model (1st Person Perspective)



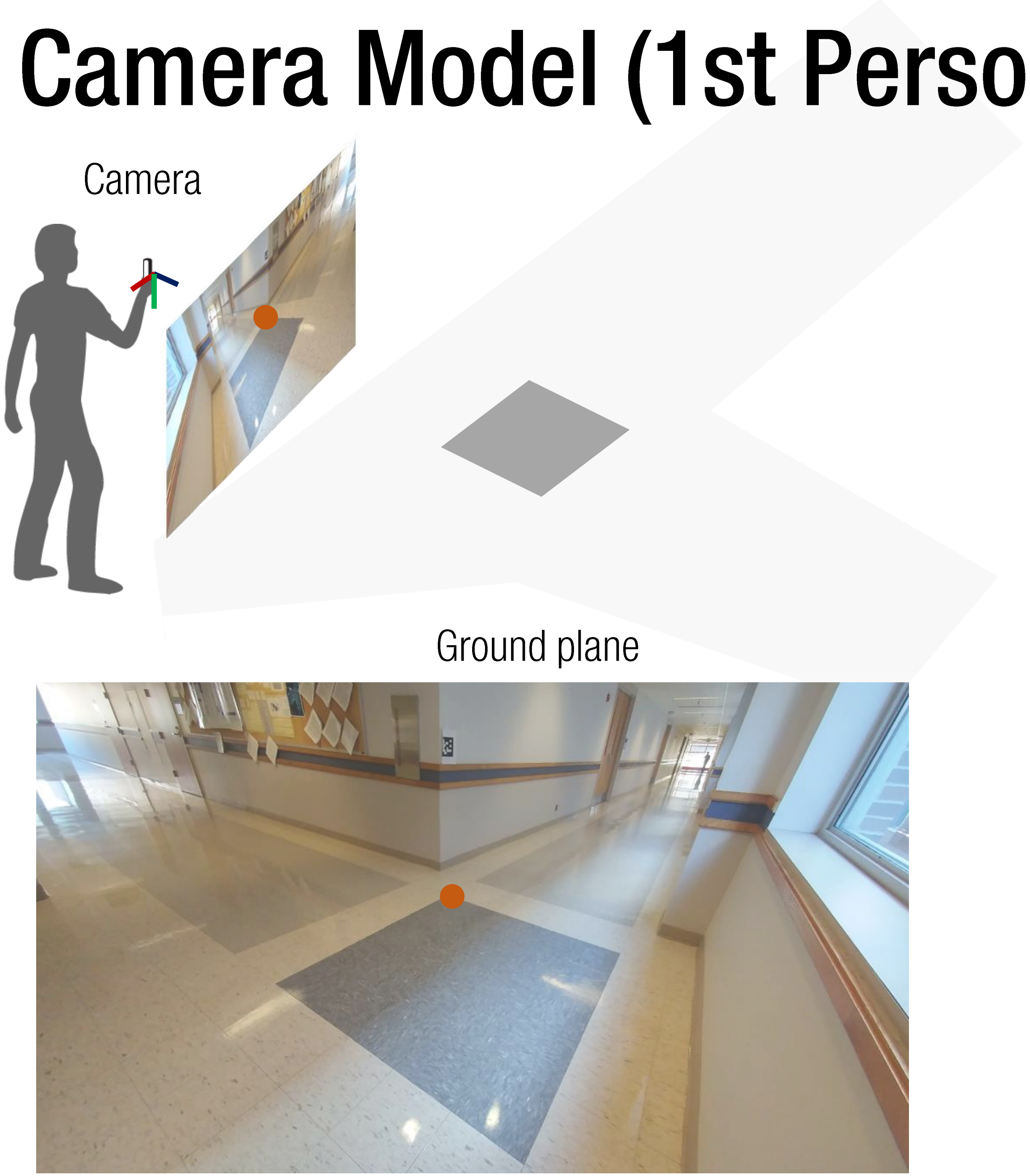
Recall camera projection matrix:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$



Camera intrinsic parameter
: metric space to pixel space

Camera Model (1st Person Perspective)

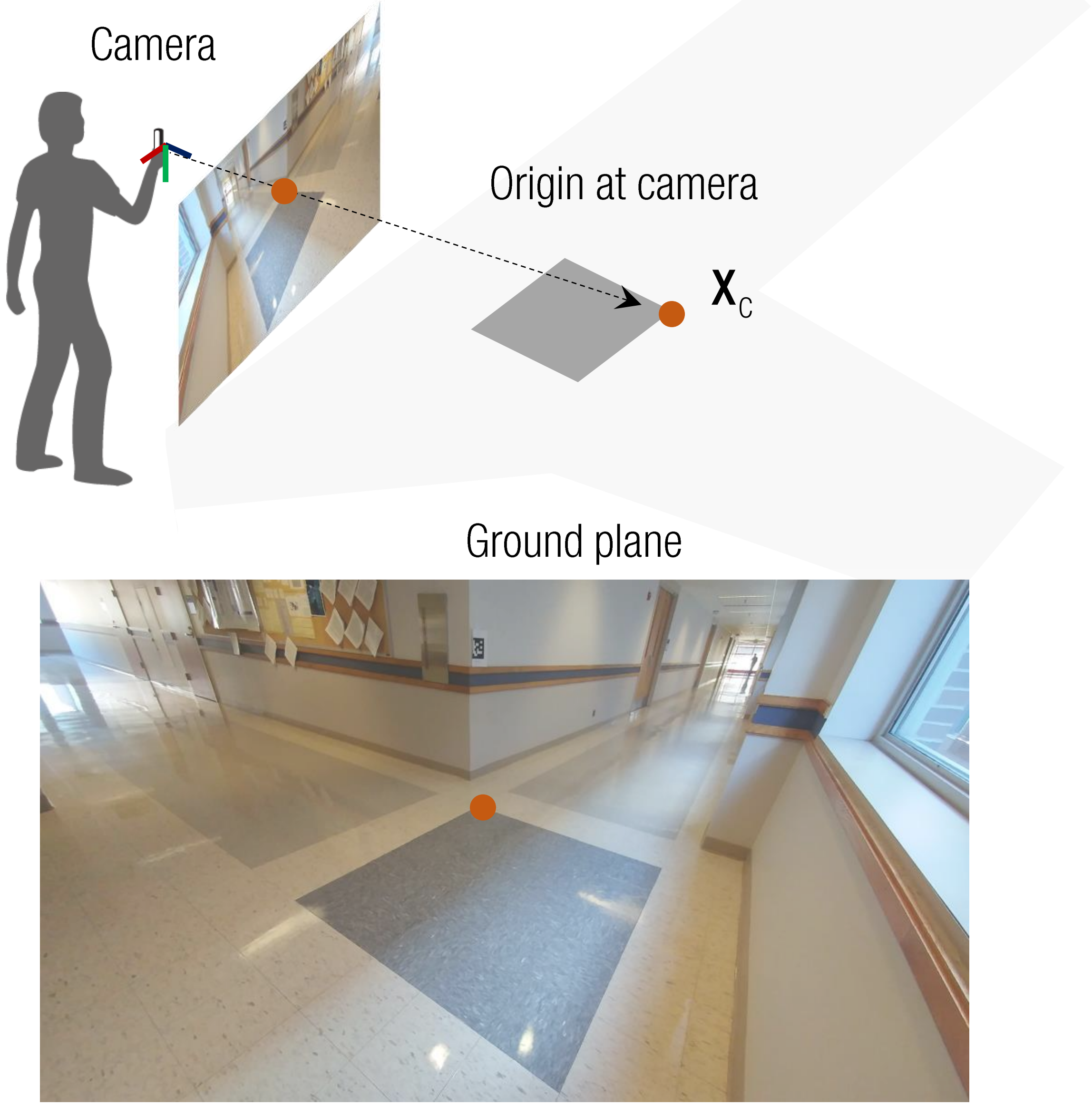


Recall camera projection matrix:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

2D image (pix) 3D world (metric)

Camera Model (1st Person Perspective)



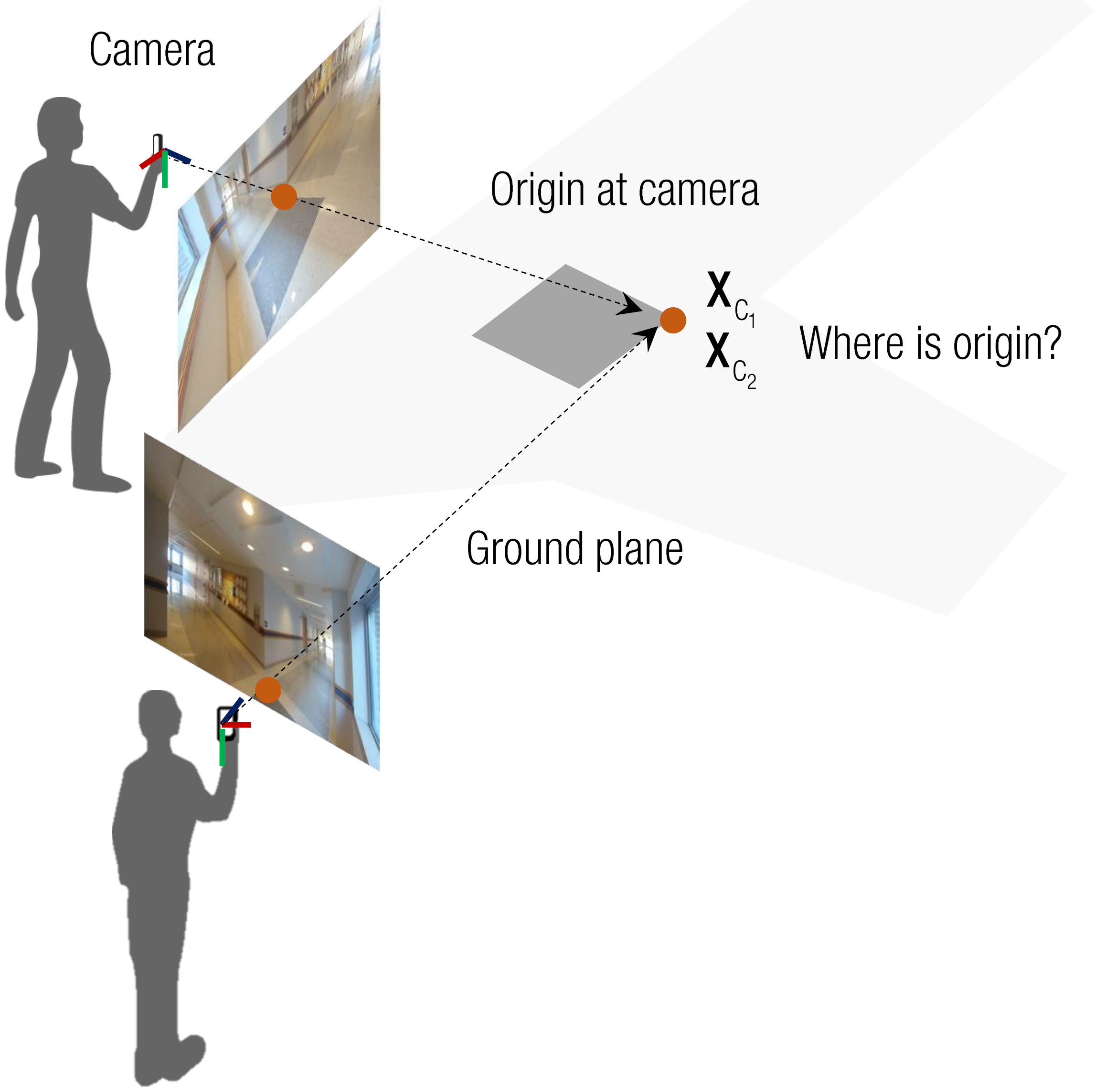
Recall camera projection matrix:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

2D image (pix) 3D world (metric)

$$\longrightarrow \lambda \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{X}_C$$

Camera Model (3rd Person Perspective)



Recall camera projection matrix:

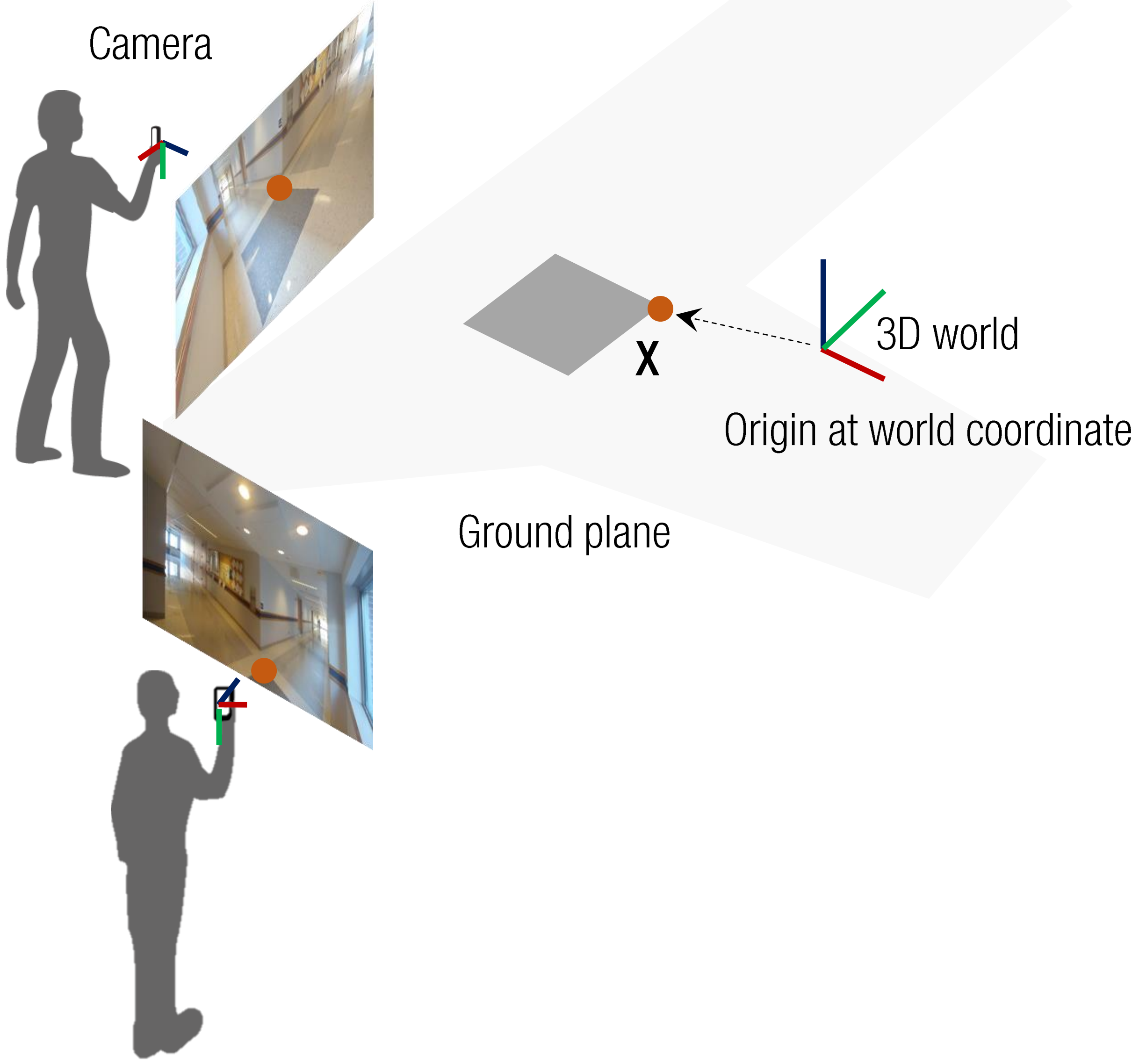
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$

2D image (pix) 3D world (metric)

$$\rightarrow \lambda \mathbf{K}^{-1} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \mathbf{X}_{C_1}$$

$$\lambda \mathbf{K}^{-1} \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \mathbf{X}_{C_2}$$

Camera Model (3rd Person Perspective)



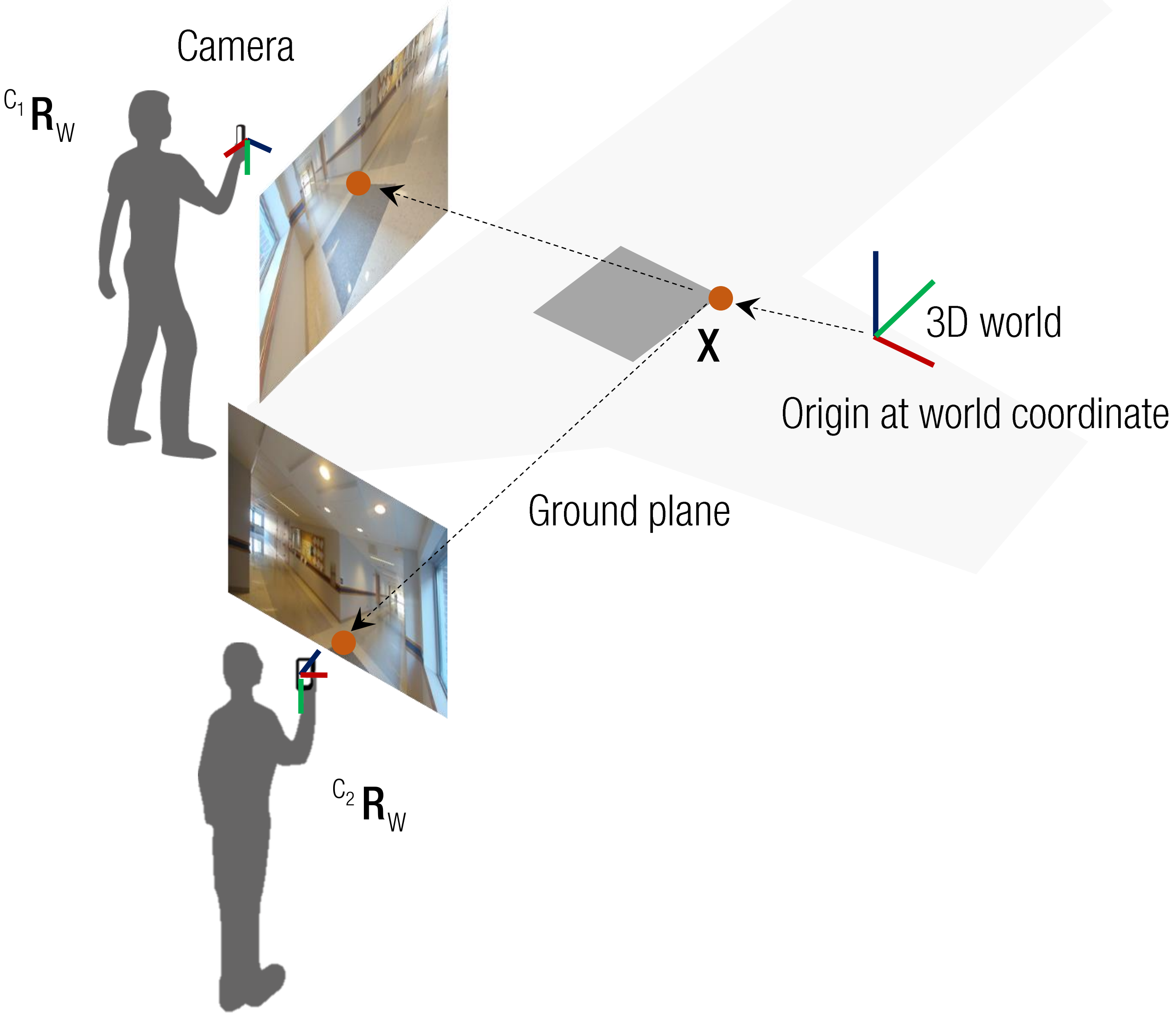
Recall camera projection matrix:

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

2D image (pix)

3D world (metric)

Coordinate Transform (Rotation)



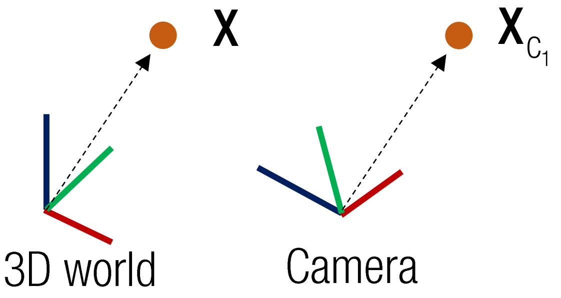
Recall camera projection matrix:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{\mathbf{X}}$$

2D image (pix) 3D world (metric)

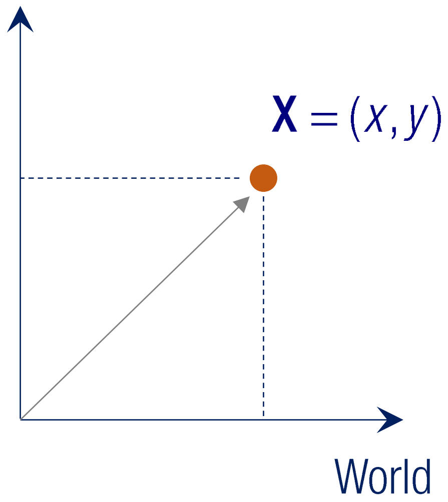
Coordinate transformation from world to camera:

$$\mathbf{X}_{C_1} = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X}$$



Coordinate Transform (Rotation)

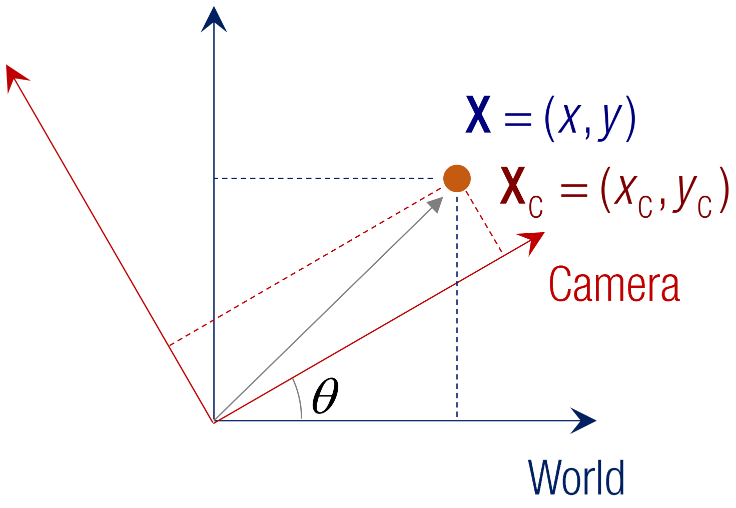
2D coordinate transform:



$$\begin{bmatrix} x \\ y \end{bmatrix}$$

Coordinate Transform (Rotation)

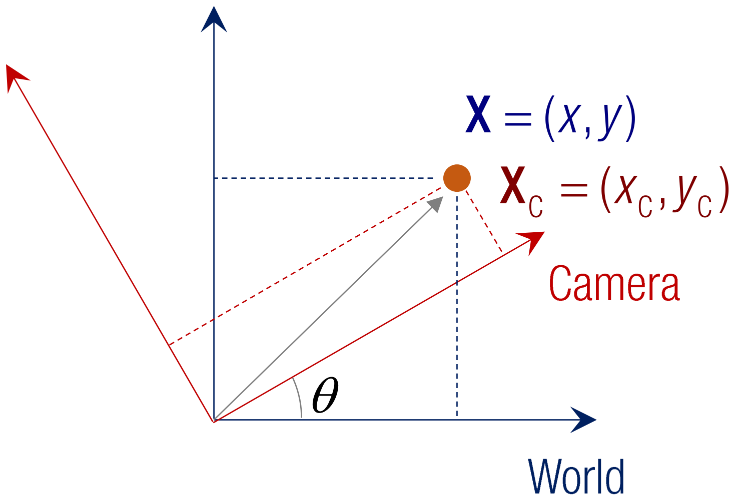
2D coordinate transform:



$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \quad ? \quad \begin{bmatrix} x \\ y \end{bmatrix}$$

Coordinate Transform (Rotation)

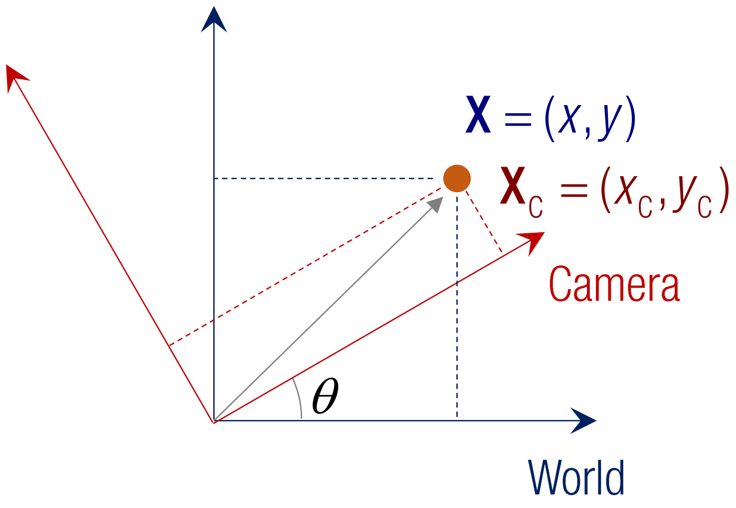
2D coordinate transform:



$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Coordinate Transform (Rotation)

2D coordinate transform:

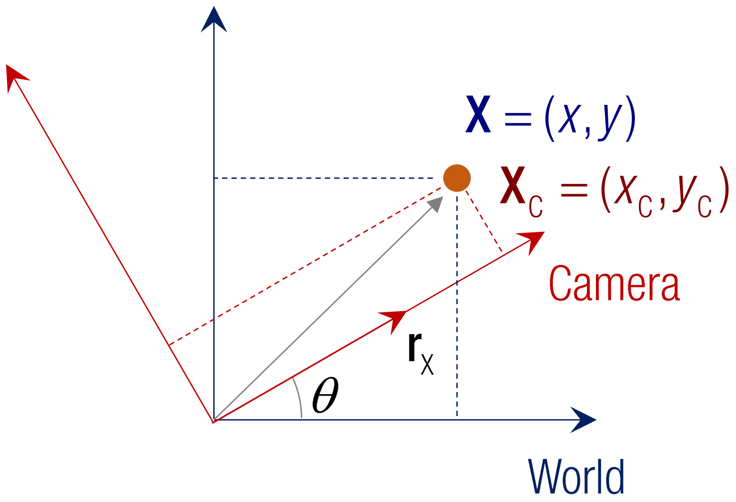


$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\det\left(\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\right) = \cos^2\theta + \sin^2\theta = 1$$

Coordinate Transform (Rotation)

2D coordinate transform:

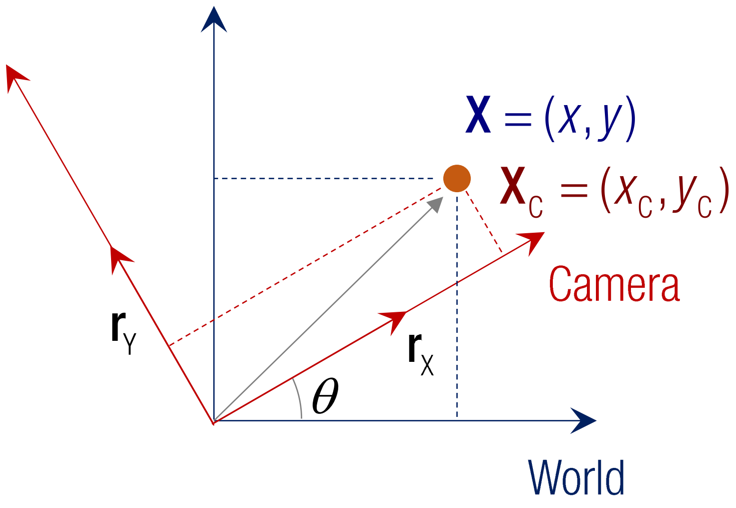


$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$\mathbf{r}_x$: x axis of camera seen from the world

Coordinate Transform (Rotation)

2D coordinate transform:



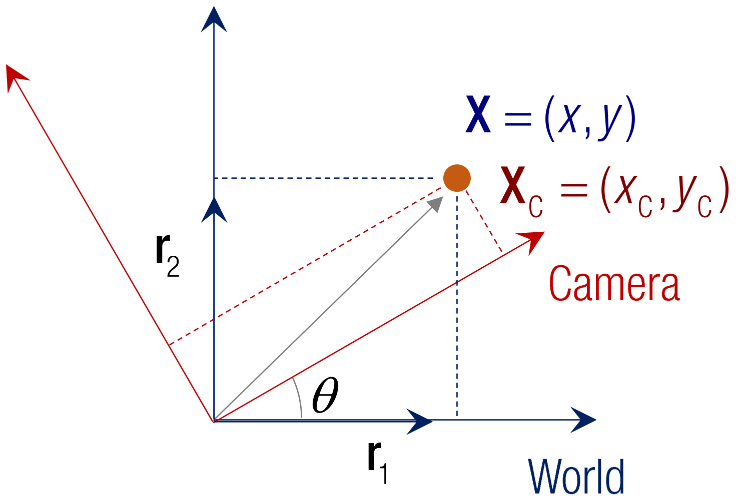
$$\begin{bmatrix} x_C \\ y_C \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad \begin{matrix} \mathbf{r}_X \\ \mathbf{r}_Y \end{matrix}$$

$\mathbf{r}_X$: x axis of camera seen from the world

$\mathbf{r}_Y$: y axis of camera seen from the world

Coordinate Transform (Rotation)

2D coordinate transform:

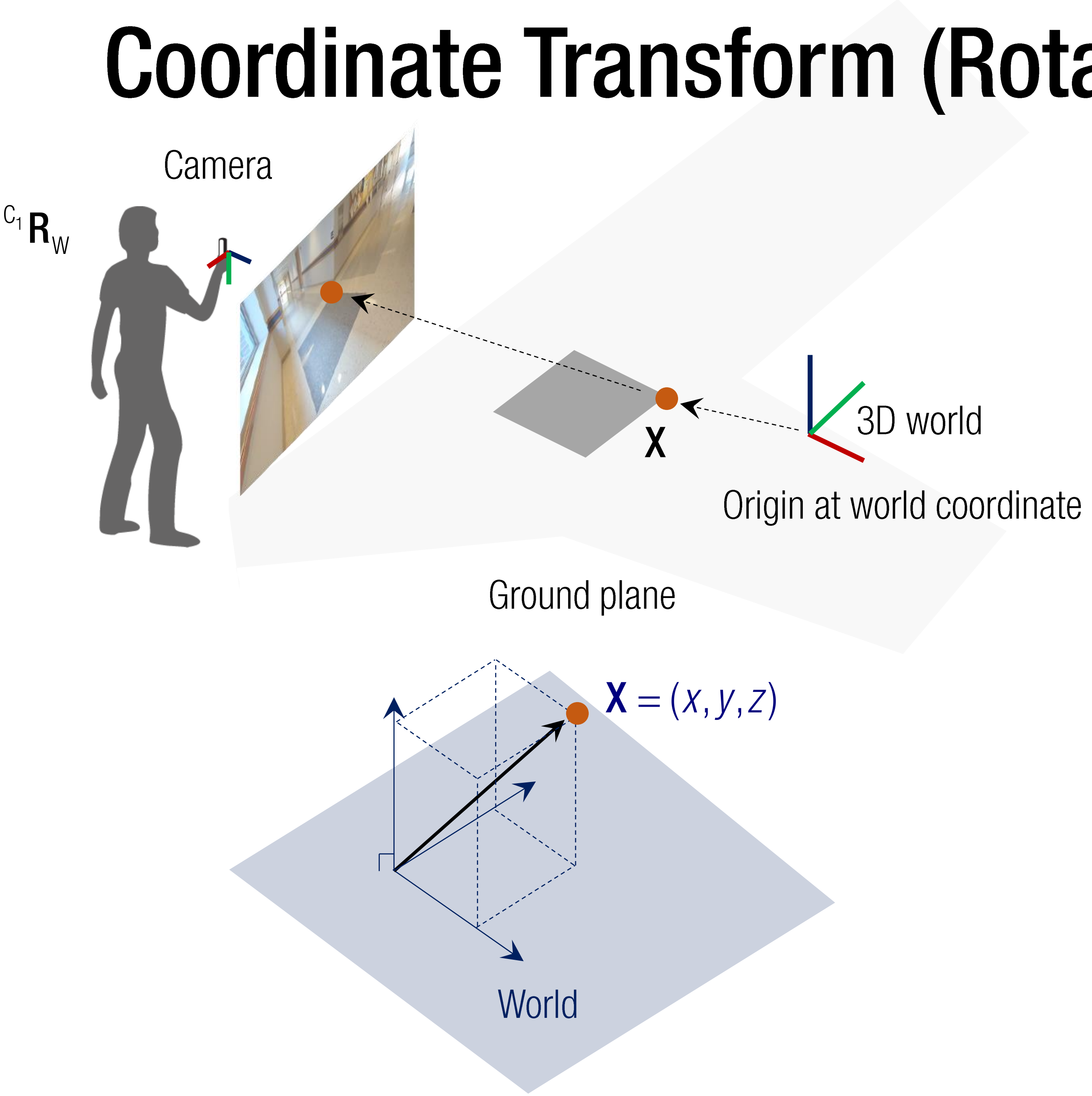


$$\begin{bmatrix} x_C \\ y_C \end{bmatrix} = \begin{bmatrix} \underbrace{\begin{matrix}\cos\theta \\ -\sin\theta\end{matrix}}_{\mathbf{r}_1} & \underbrace{\begin{matrix}\sin\theta \\ \cos\theta\end{matrix}}_{\mathbf{r}_2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$\mathbf{r}_1$: x axis of world seen from the camera

$\mathbf{r}_2$: y axis of world seen from the camera

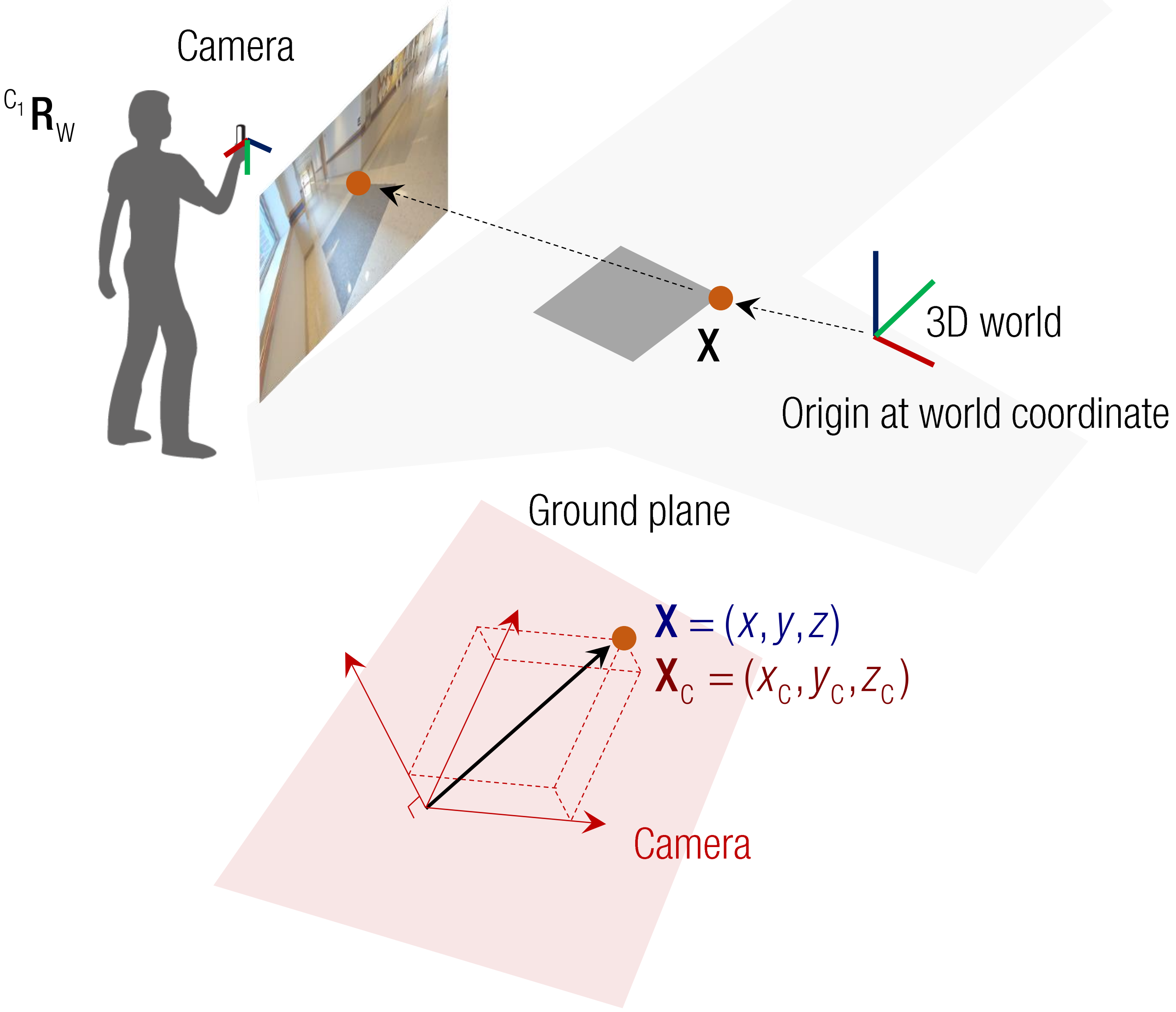
Coordinate Transform (Rotation)



Coordinate transformation from world to camera:

$$\mathbf{X}_C = \quad ? \quad \mathbf{X}$$

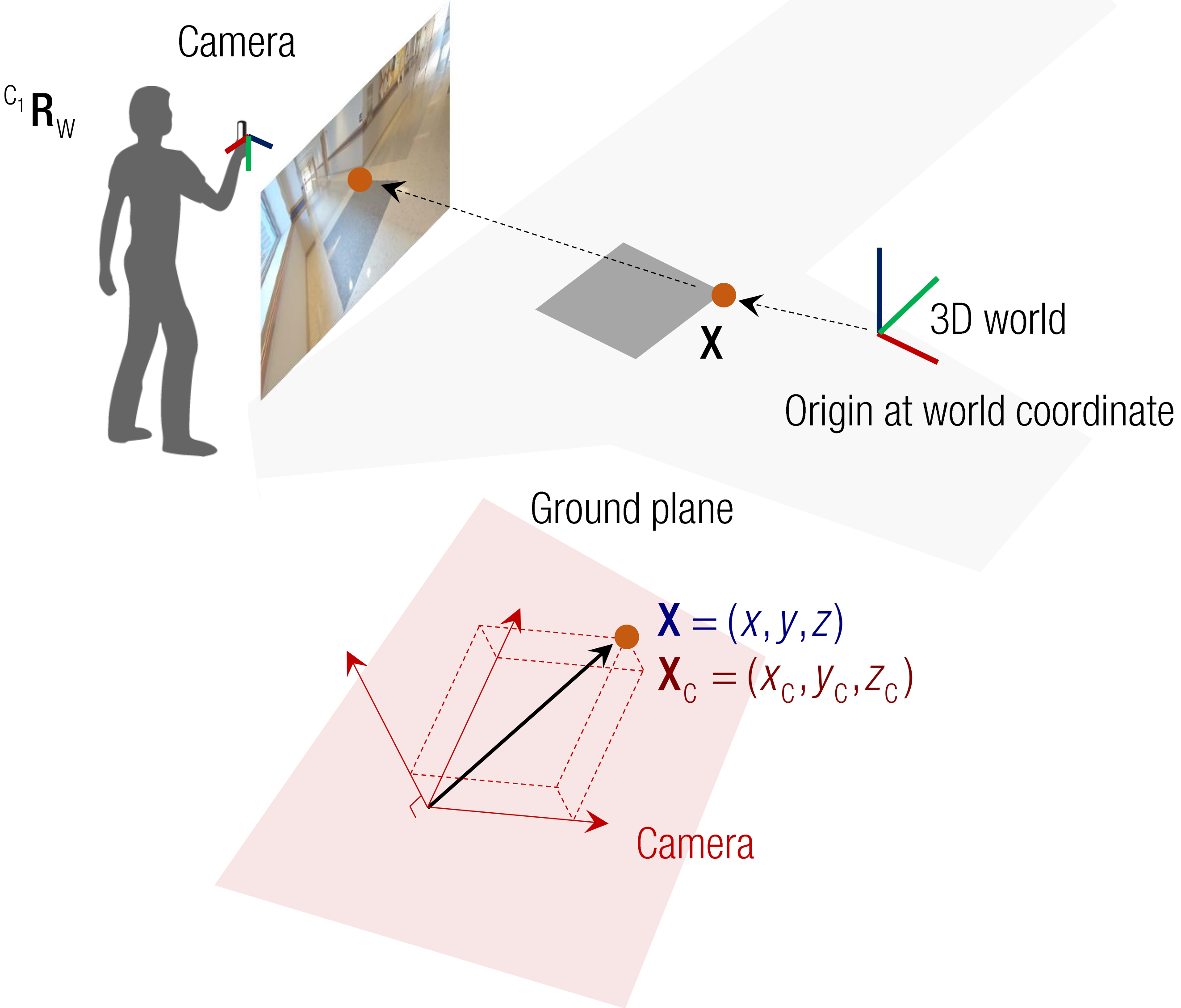
Coordinate Transform (Rotation)



Coordinate transformation from world to camera:

$$\mathbf{X}_C = \quad ? \quad \mathbf{X}$$

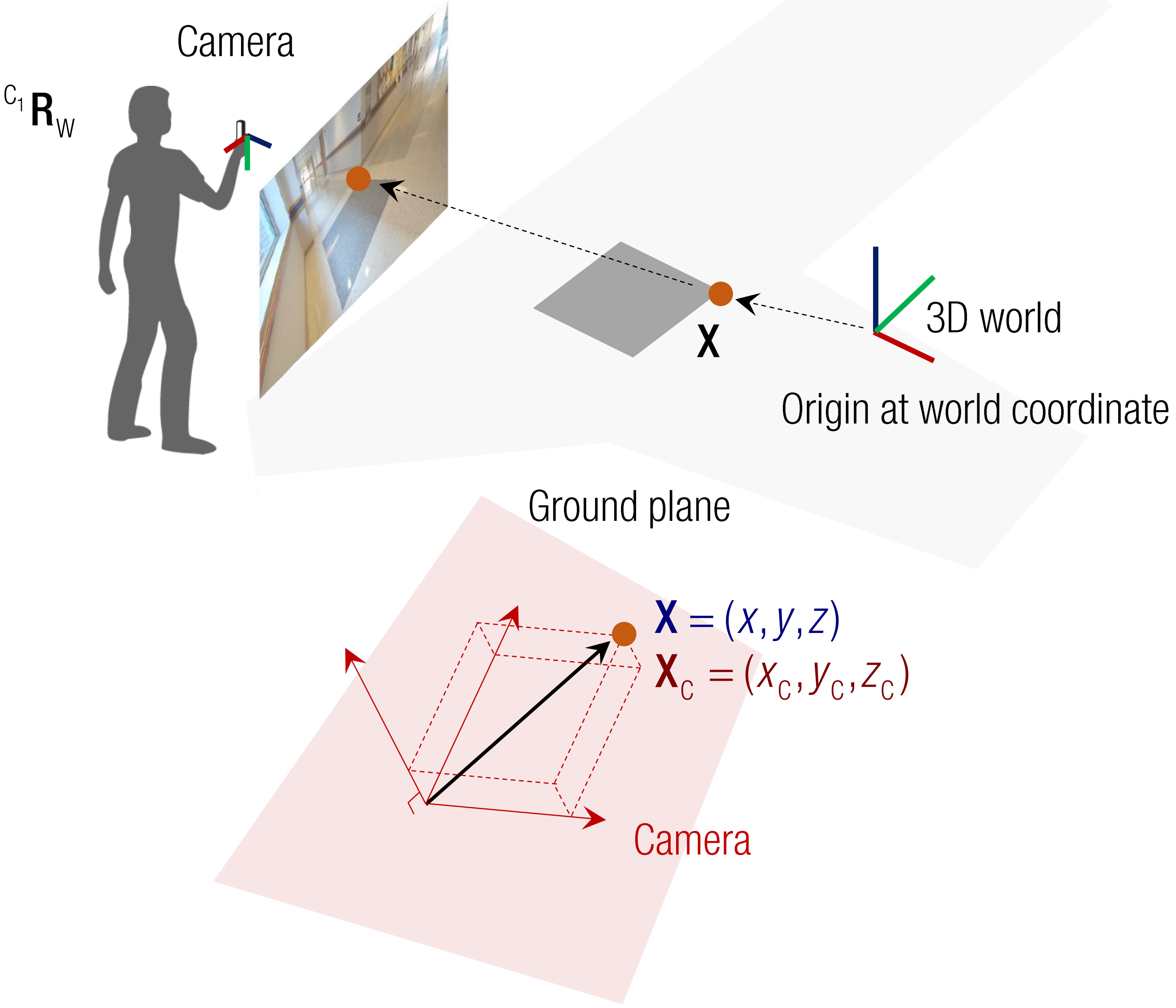
Coordinate Transform (Rotation)



Coordinate transformation from world to camera:

$$\mathbf{X}_C = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_W \mathbf{X}$$

Coordinate Transform (Rotation)

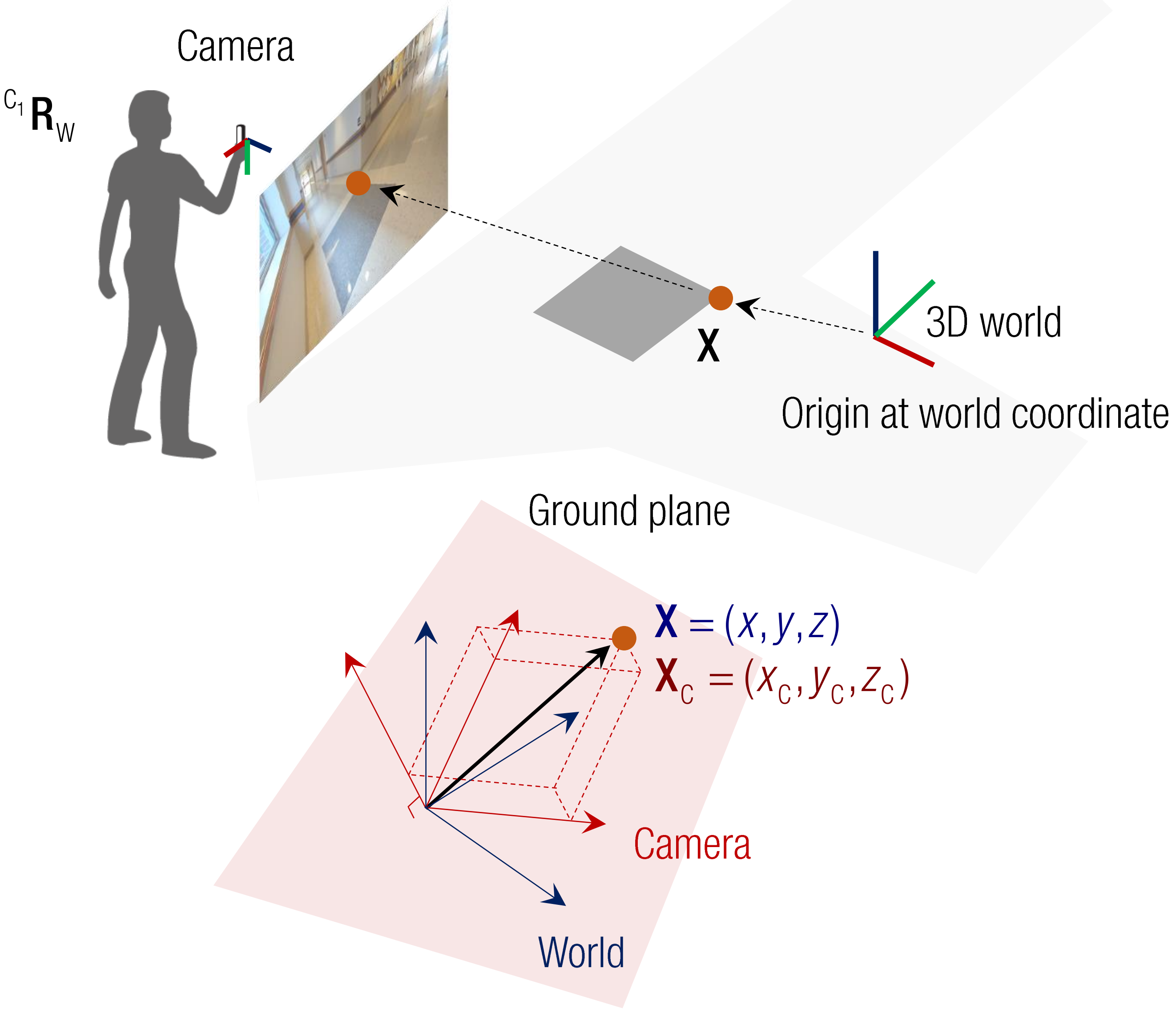


Coordinate transformation from world to camera:

$$\mathbf{X}_C = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_W \mathbf{X}$$

Degree of freedom?

Coordinate Transform (Rotation)



Coordinate transformation from world to camera:

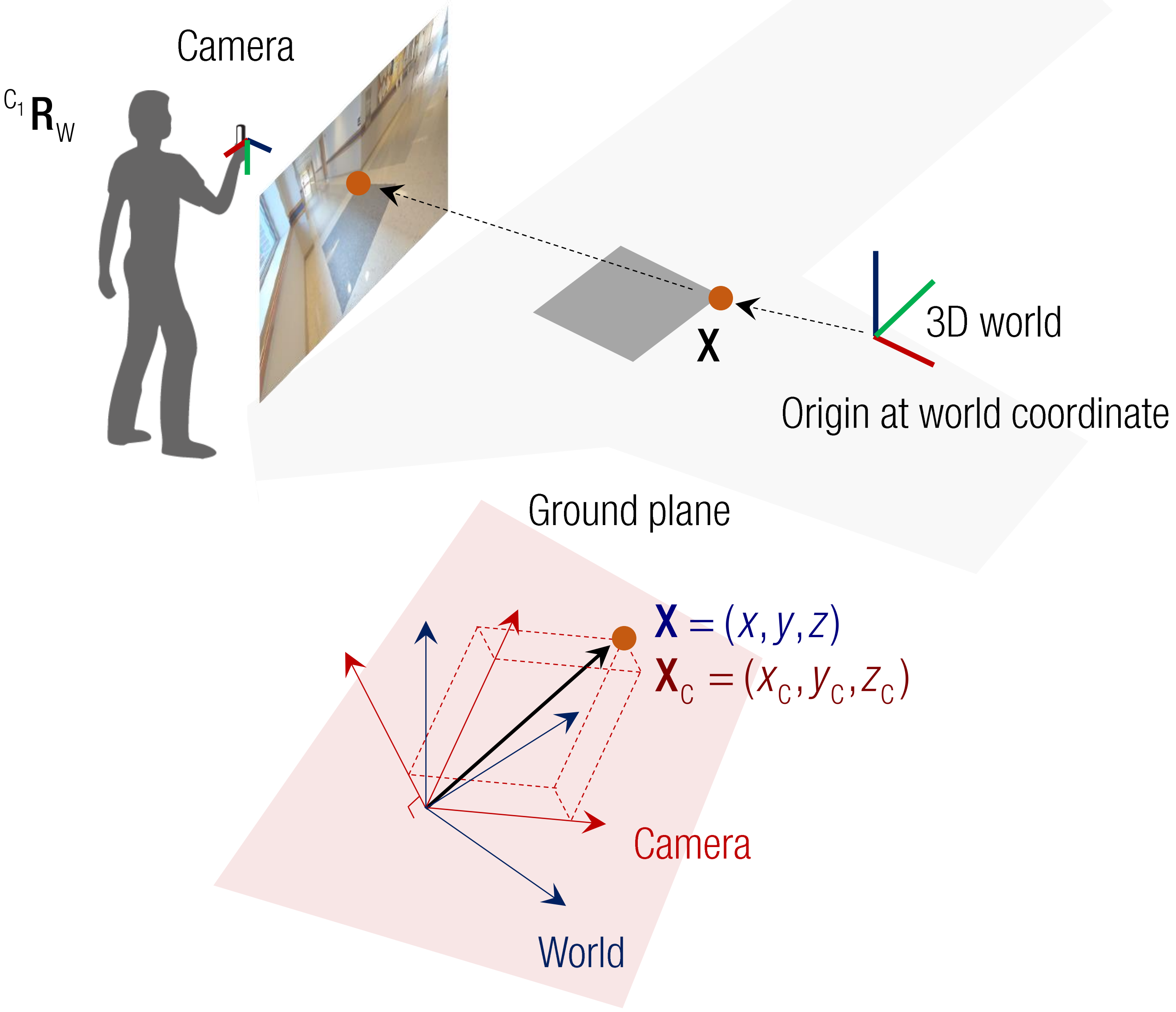
$$\mathbf{X}_C = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_W \mathbf{X}$$

Degree of freedom?

$${}^{C}\mathbf{R}_W \in SO(3)$$

- Orthogonal matrix $\longrightarrow \left({}^{C}\mathbf{R}_W\right)^T \left({}^{C}\mathbf{R}_W\right) = \mathbf{I}_3, \quad \det\left({}^{C}\mathbf{R}_W\right) = 1$

Coordinate Transform (Rotation)

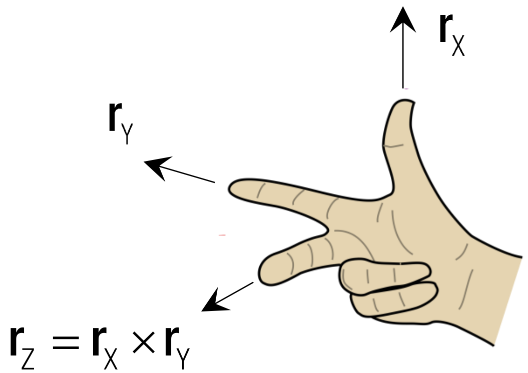


Coordinate transformation from world to camera:

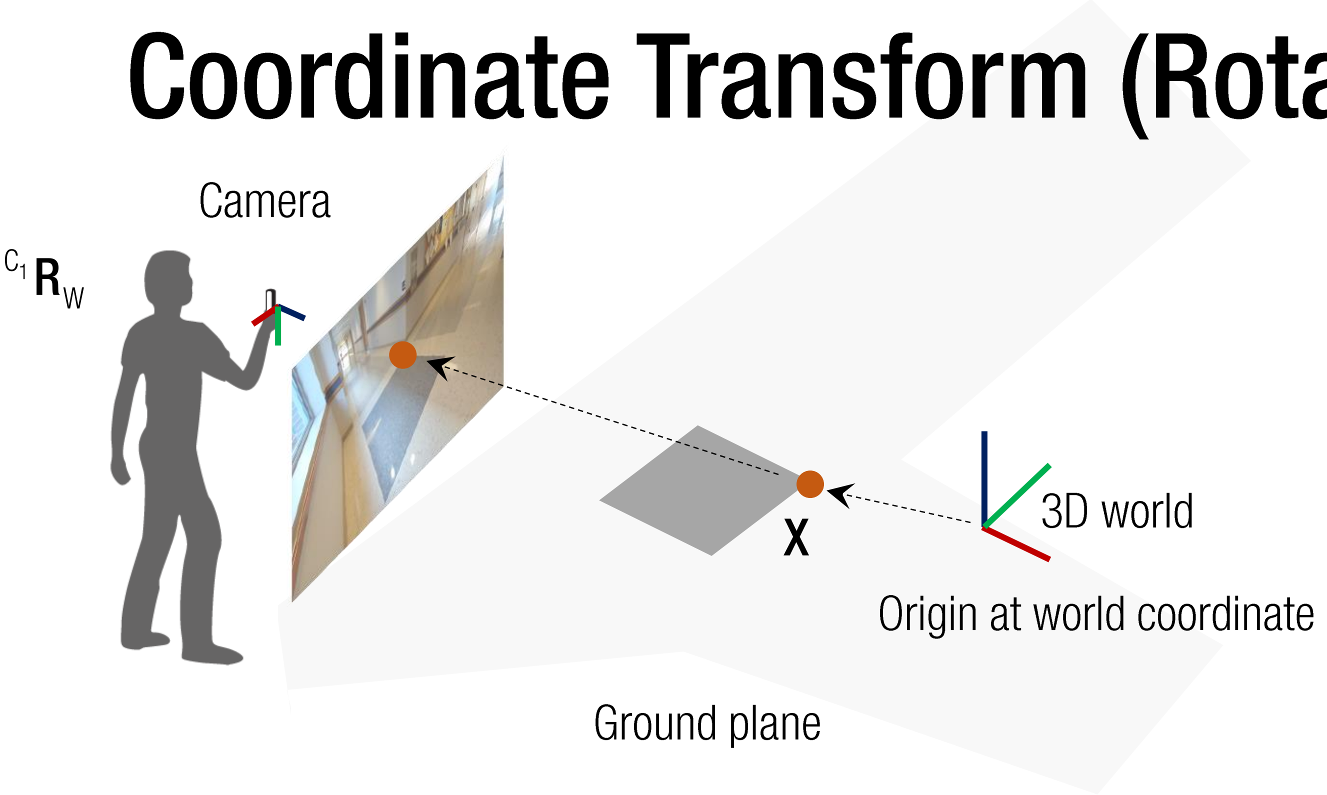
$$\mathbf{X}_C = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_W \mathbf{X}, \qquad \begin{bmatrix} \mathbf{r}_X \\ \mathbf{r}_Y \\ \mathbf{r}_Z \end{bmatrix}$$

$${}^{C}\mathbf{R}_W \in SO(3)$$

- Orthogonal matrix $\longrightarrow \left({}^{C}\mathbf{R}_W\right)^{T}\left({}^{C}\mathbf{R}_W\right) = \mathbf{I}_3, \quad \det\left({}^{C}\mathbf{R}_W\right) = 1$
- Right hand rule



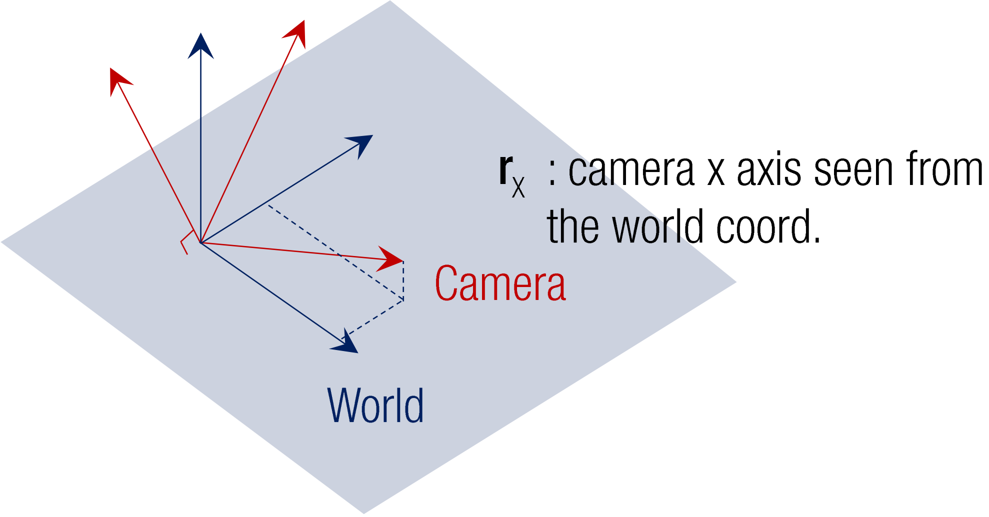
Coordinate Transform (Rotation)



Coordinate transformation from world to camera:

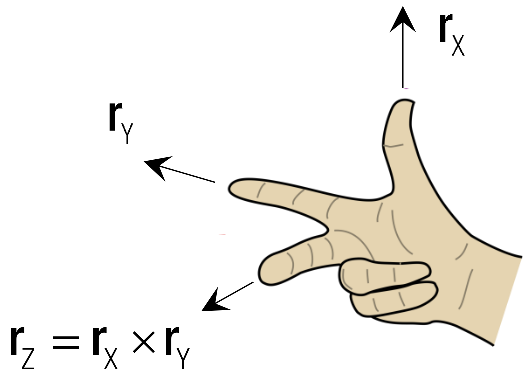
$$\mathbf{X}_C = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_W \mathbf{X}$$

$$\begin{bmatrix} \mathbf{r}_X \\ \mathbf{r}_Y \\ \mathbf{r}_Z \end{bmatrix}$$

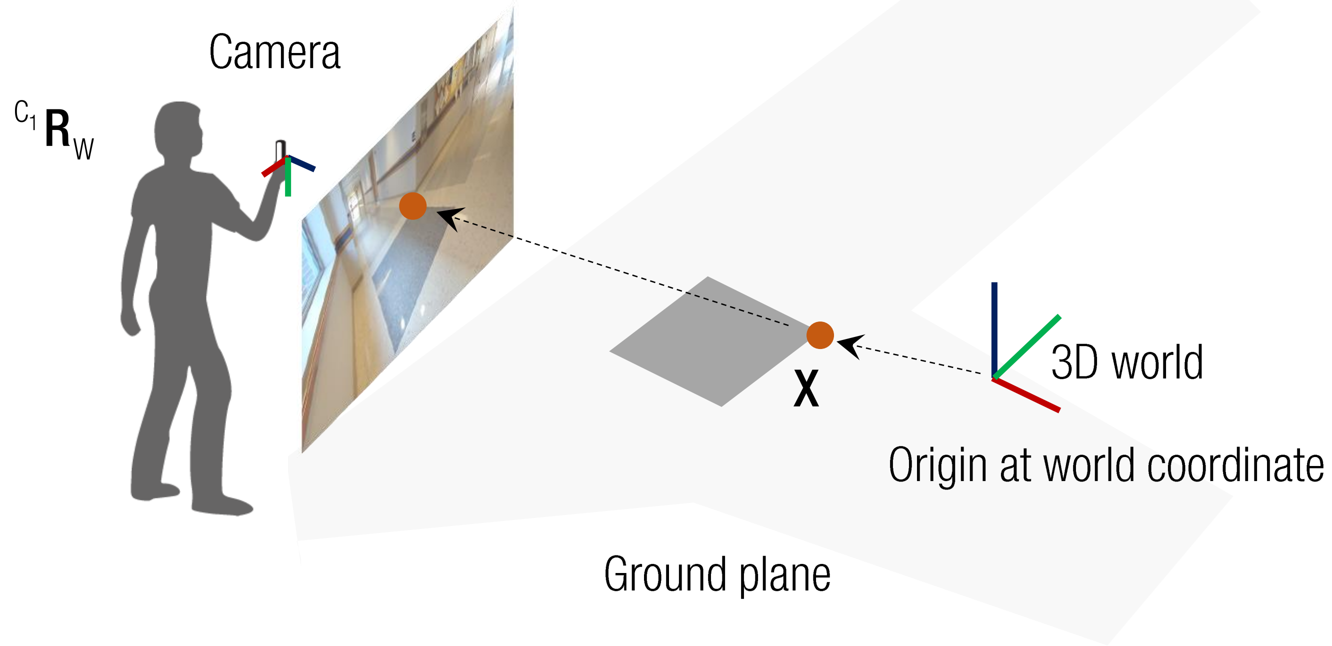


$${}^{C}\mathbf{R}_W \in SO(3)$$

- Orthogonal matrix $\longrightarrow \left({}^{C}\mathbf{R}_W\right)^{T}\left({}^{C}\mathbf{R}_W\right) = \mathbf{I}_3, \quad \det\left({}^{C}\mathbf{R}_W\right) = 1$
- Right hand rule



Coordinate Transform (Rotation)

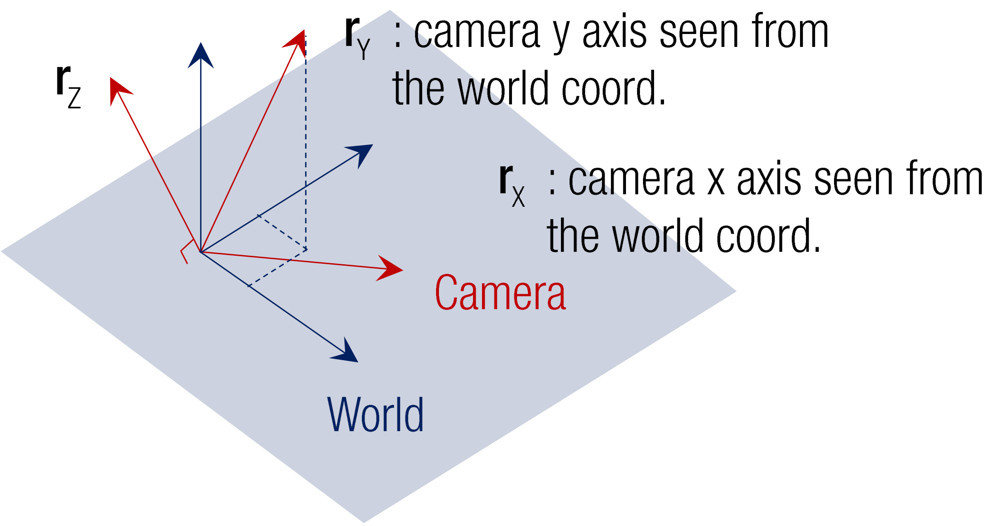


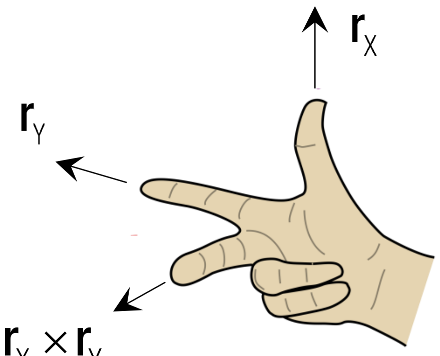
Coordinate transformation from world to camera:

$$\mathbf{X}_C = \begin{bmatrix} \mathbf{r}_X \\ \mathbf{r}_Y \\ \mathbf{r}_Z \end{bmatrix}\mathbf{X} = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix}\mathbf{X} = {}^{C}\mathbf{R}_W\mathbf{X}$$

$${}^{C}\mathbf{R}_W \in SO(3)$$

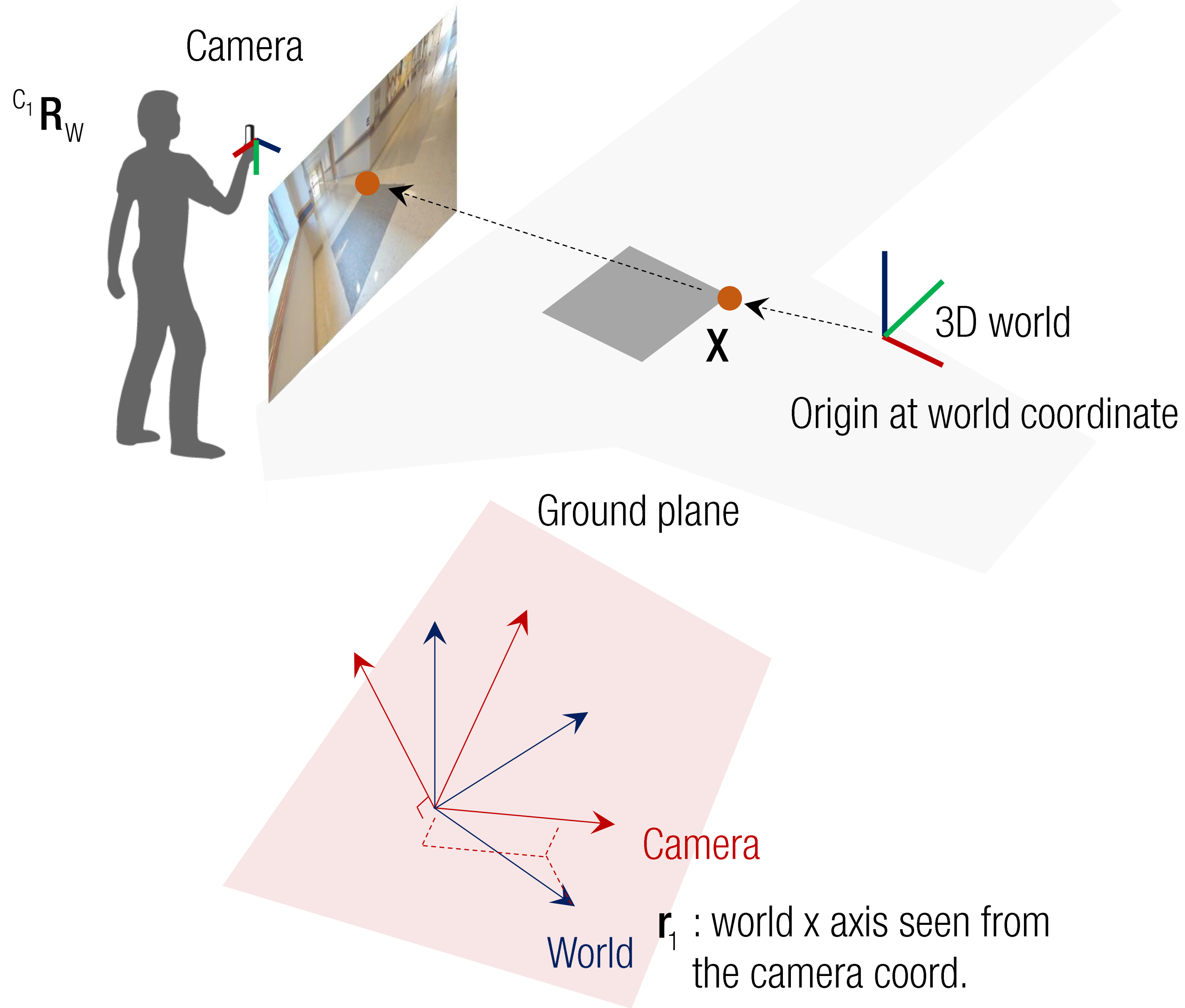
- Orthogonal matrix $\longrightarrow \left({}^{C}\mathbf{R}_W\right)^{T}\left({}^{C}\mathbf{R}_W\right) = \mathbf{I}_3, \quad \det\left({}^{C}\mathbf{R}_W\right) = 1$
- Right hand rule





$$\mathbf{r}_Z = \mathbf{r}_X \times \mathbf{r}_Y$$

Coordinate Transform (Rotation)

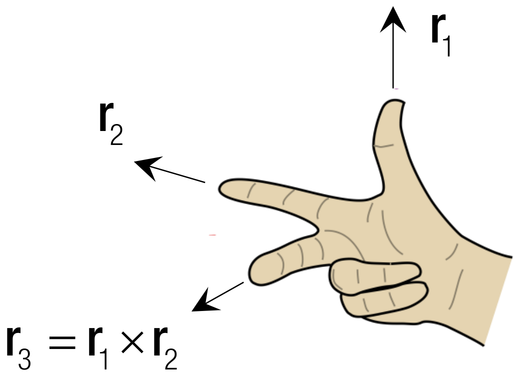


Coordinate transformation from world to camera:

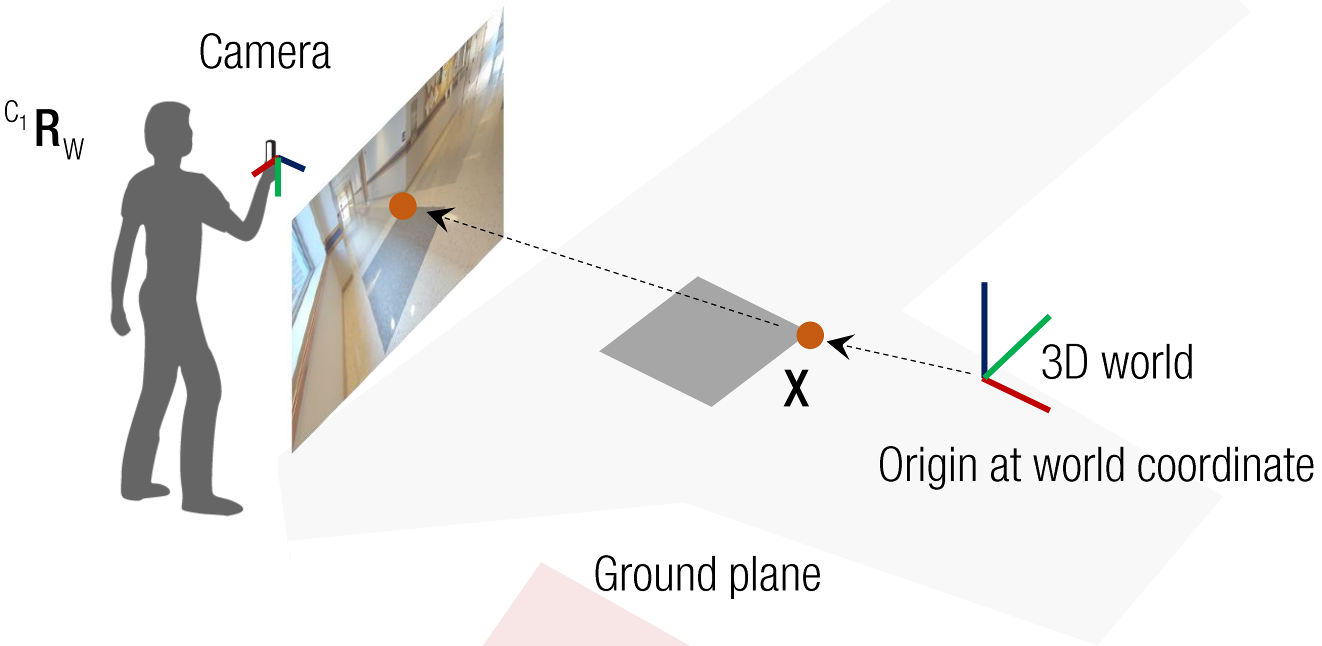
$$\mathbf{X}_C = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_W \mathbf{X}$$

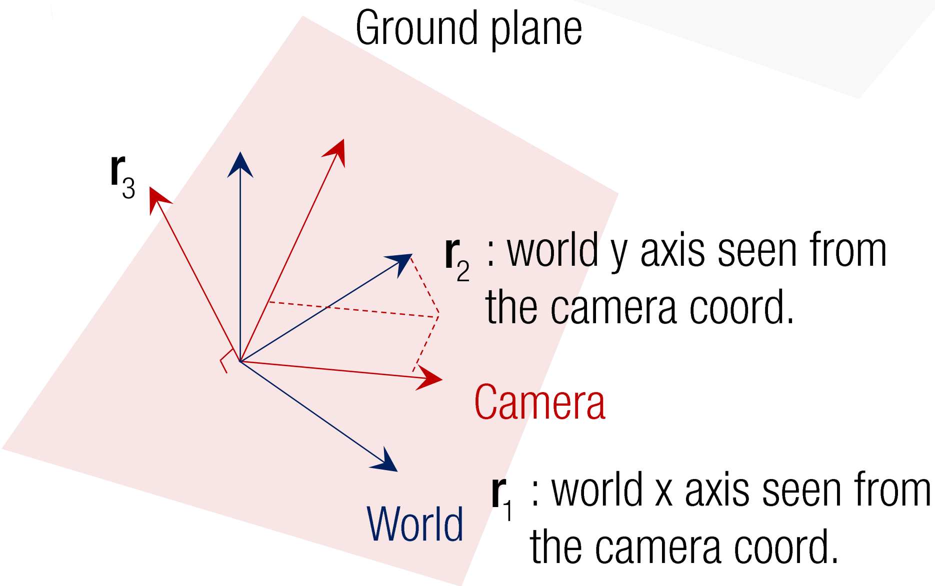
$${}^{C}\mathbf{R}_W \in SO(3)$$

- Orthogonal matrix $\longrightarrow \left({}^{C}\mathbf{R}_W\right)^T \left({}^{C}\mathbf{R}_W\right) = \mathbf{I}_3, \quad \det\left({}^{C}\mathbf{R}_W\right) = 1$
- Right hand rule



Coordinate Transform (Rotation)





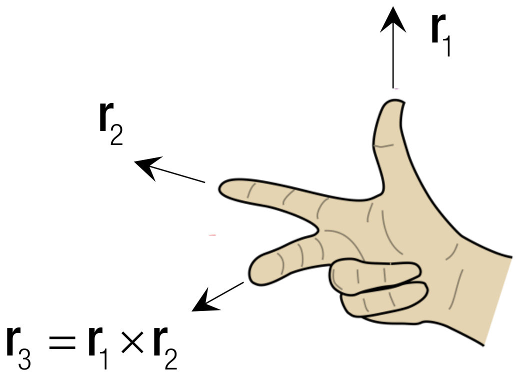
Coordinate transformation from world to camera:

$$\mathbf{X}_C = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_W \mathbf{X}$$

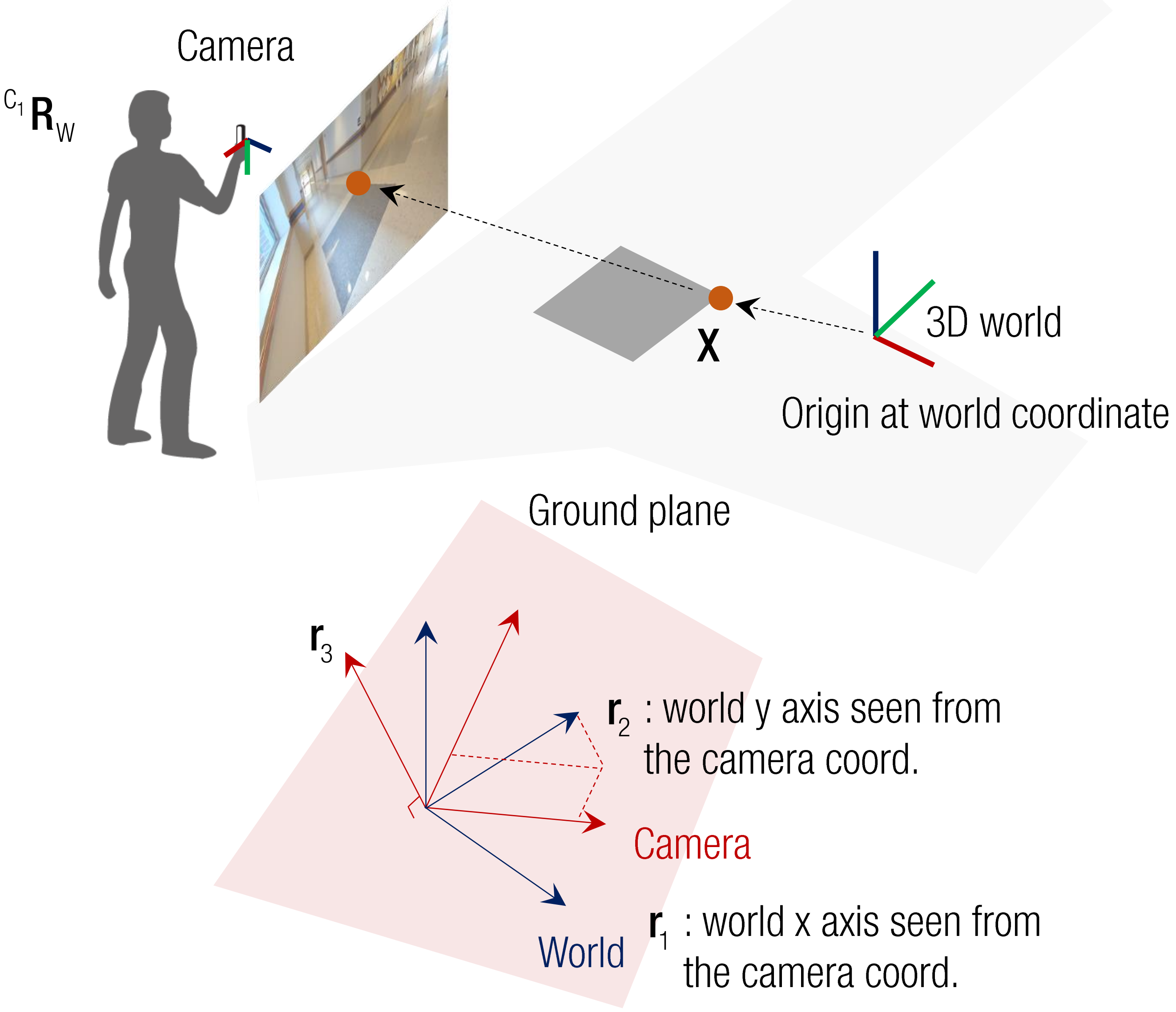
with the columns $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$.

$${}^{C}\mathbf{R}_W \in SO(3)$$

- Orthogonal matrix $\longrightarrow \left({}^{C}\mathbf{R}_W\right)^{T}\left({}^{C}\mathbf{R}_W\right) = \mathbf{I}_3, \quad \det\left({}^{C}\mathbf{R}_W\right) = 1$
- Right hand rule



Camera Projection (Pure Rotation)



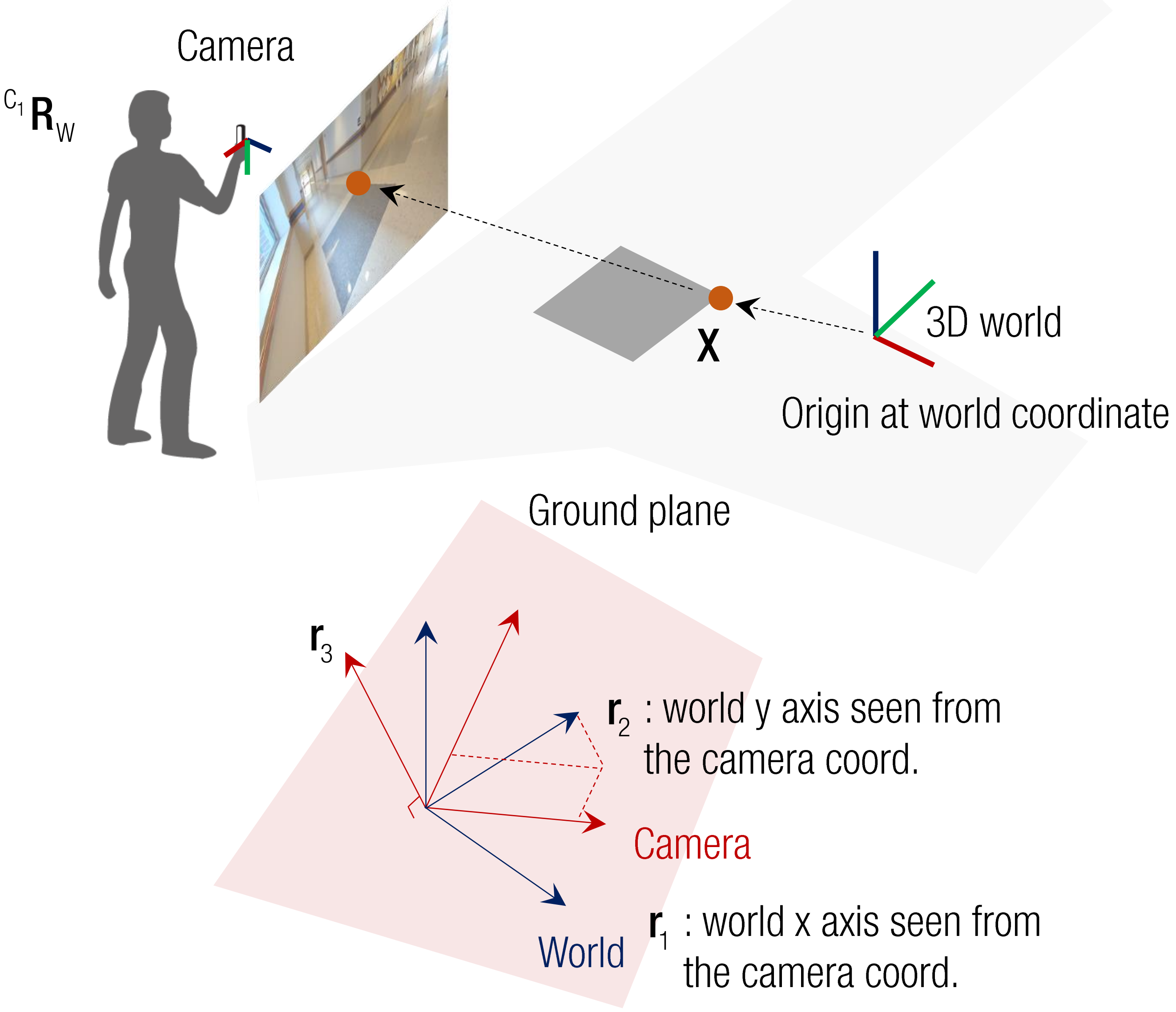
Coordinate transformation from world to camera:

$$\mathbf{X}_C = \begin{bmatrix} \underset{\mathbf{r}_1}{\begin{matrix} r_{x1} \\ r_{y1} \\ r_{z1} \end{matrix}} & \underset{\mathbf{r}_2}{\begin{matrix} r_{x2} \\ r_{y2} \\ r_{z2} \end{matrix}} & \underset{\mathbf{r}_3}{\begin{matrix} r_{x3} \\ r_{y3} \\ r_{z3} \end{matrix}} \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_W \mathbf{X}$$

Camera projection of world point:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}$$

Camera Projection (Pure Rotation)



Coordinate transformation from world to camera:

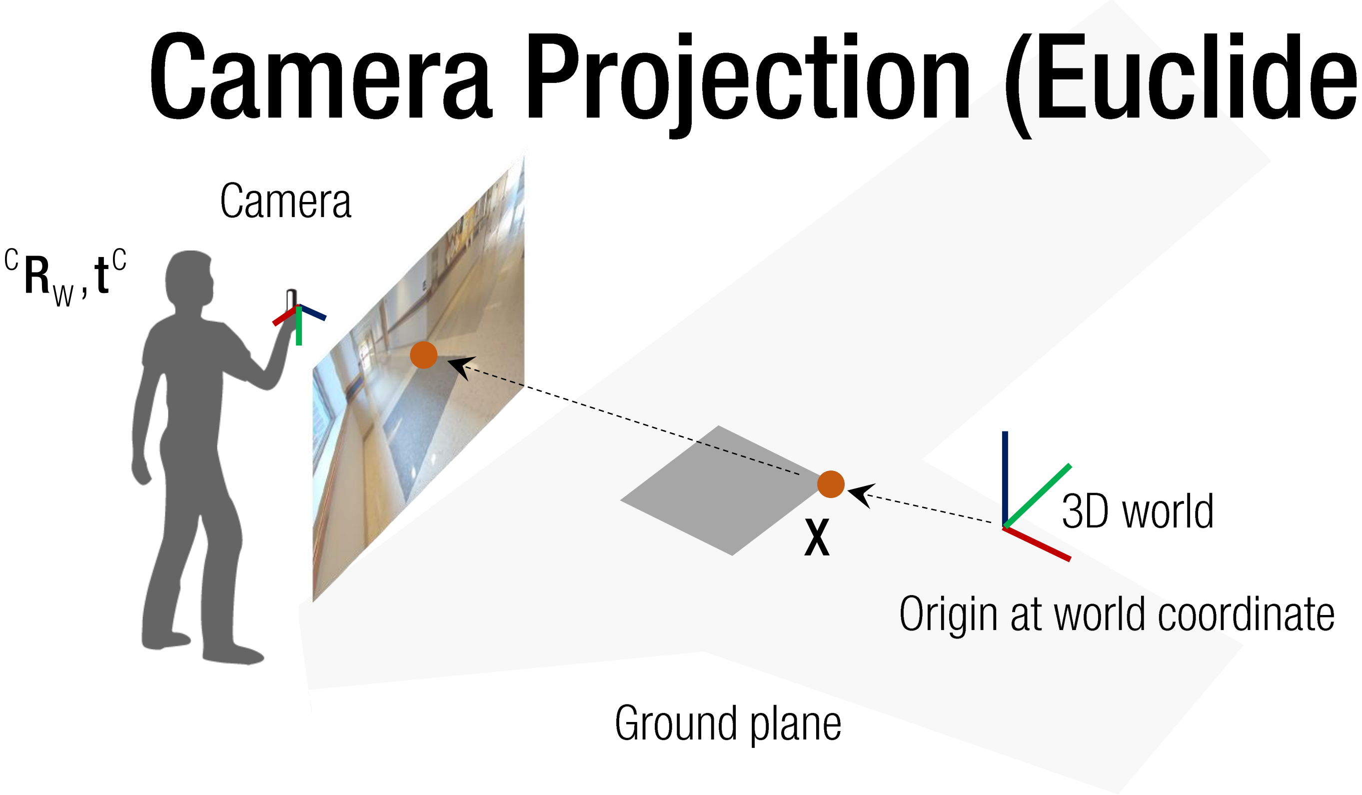
$$\mathbf{X}_C = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_W \mathbf{X}$$

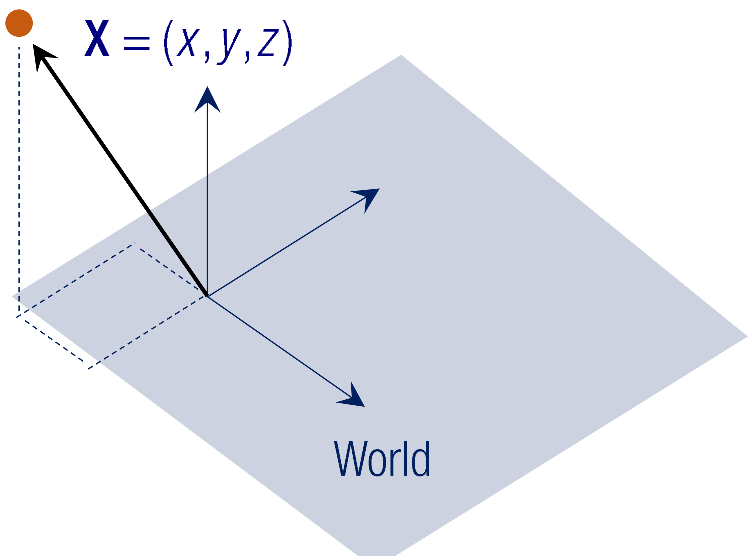
Camera projection of world point:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Camera Projection (Euclidean Transform)

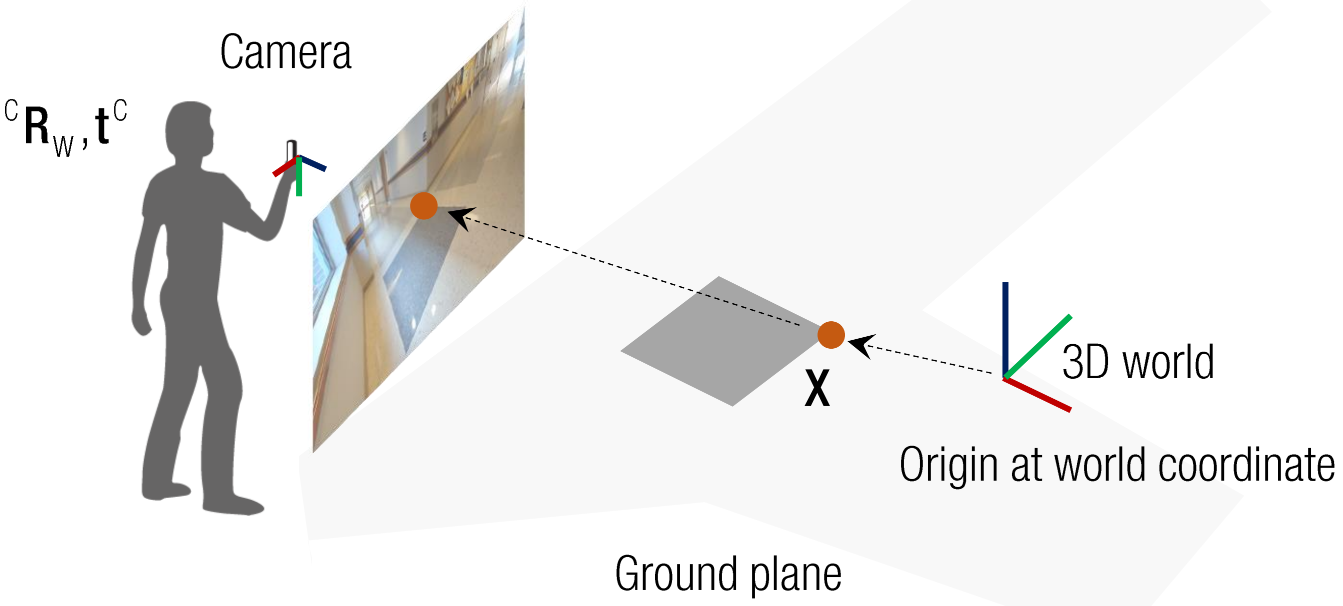


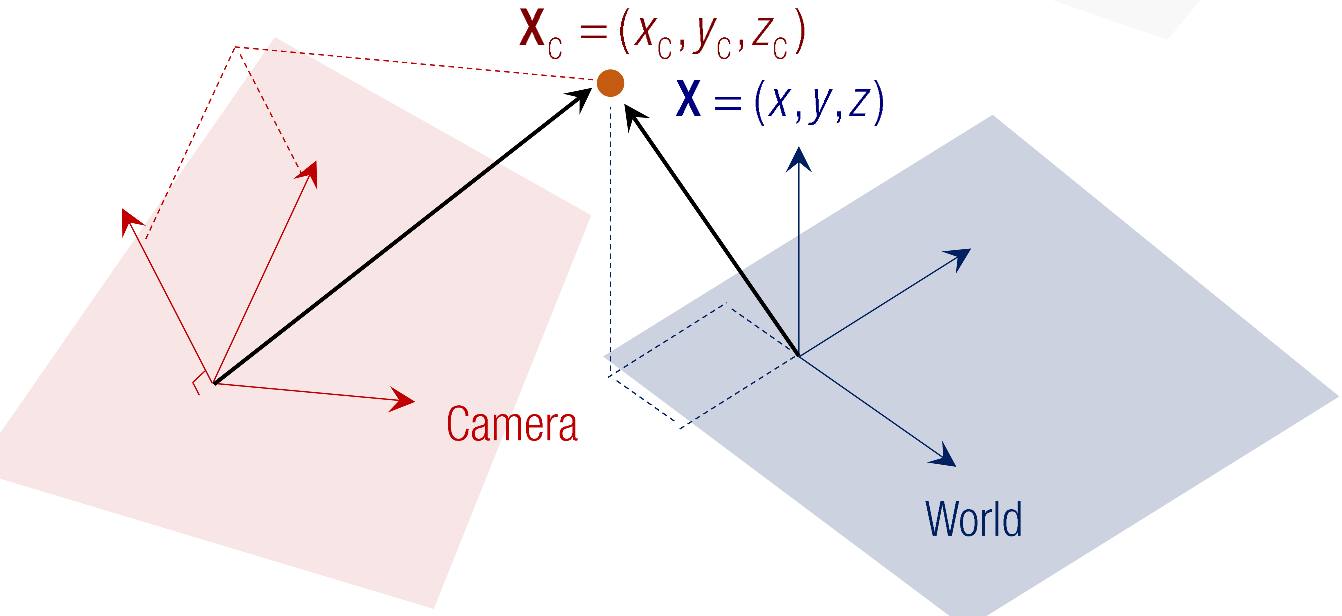


Coordinate transformation from world to camera:

$$\mathbf{X}_C = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \mathbf{X} = {}^{C}\mathbf{R}_{W}\mathbf{X}$$

Camera Projection (Euclidean Transform)

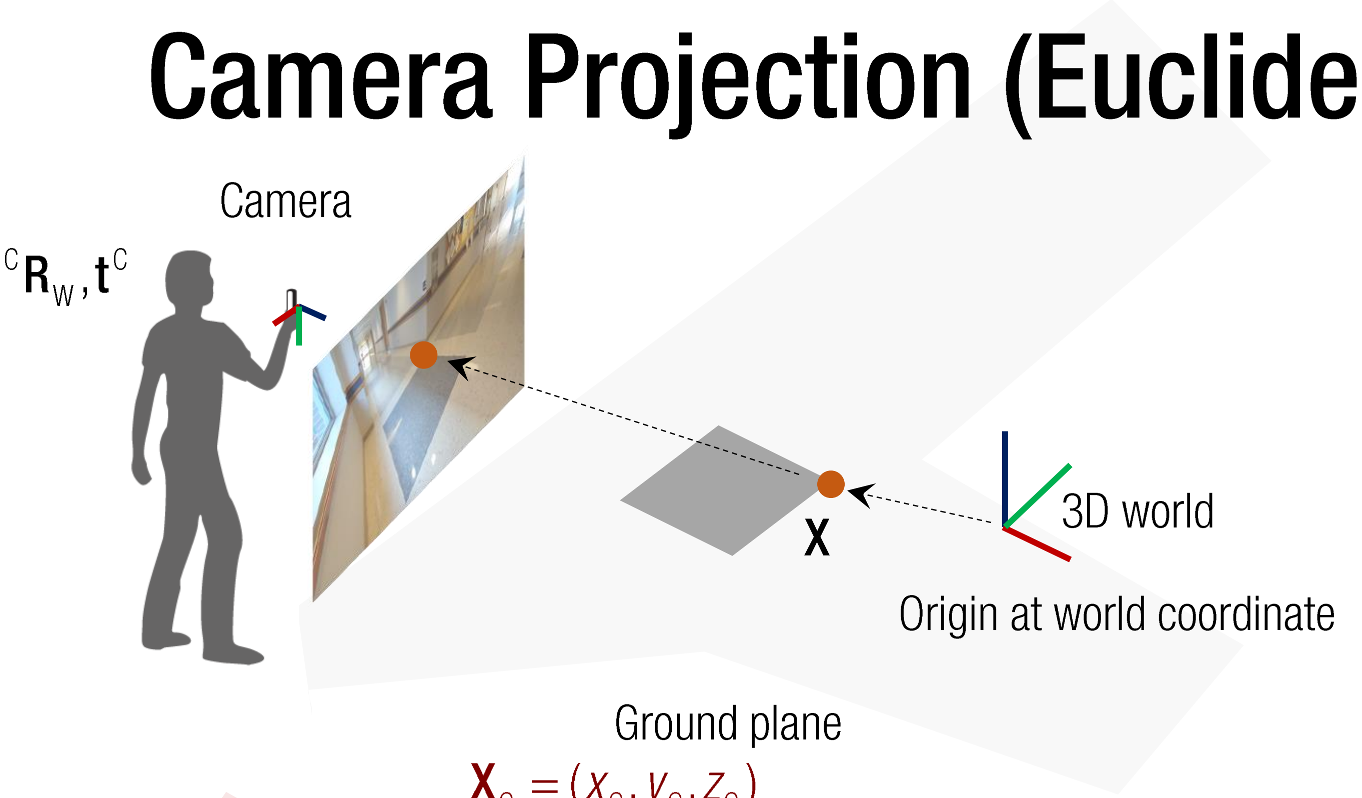


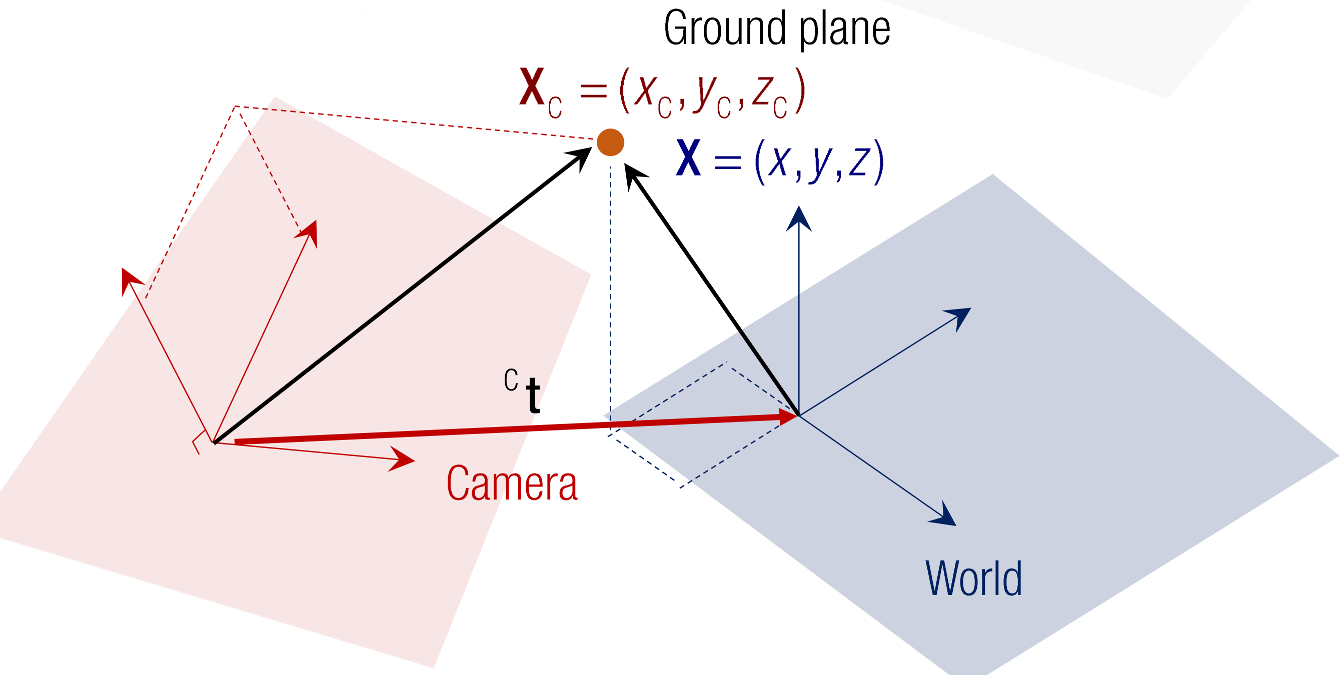


Coordinate transformation from world to camera:

$$\mathbf{X}_C = {}^C\mathbf{R}_W\mathbf{X} + {}^C\mathbf{t} = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} & t_x \\ r_{y1} & r_{y2} & r_{y3} & t_y \\ r_{z1} & r_{z2} & r_{z3} & t_z \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

Camera Projection (Euclidean Transform)



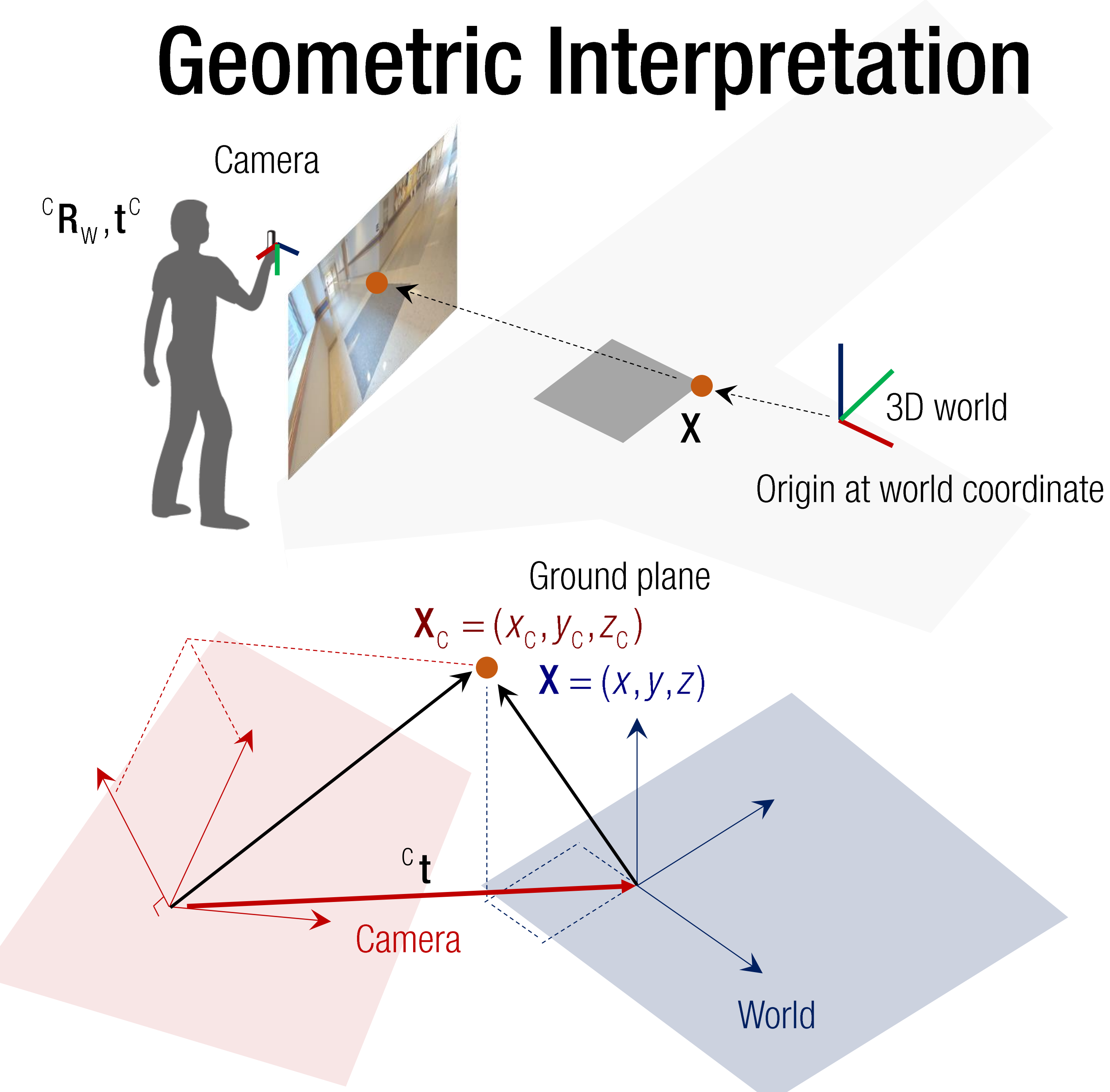


Coordinate transformation from world to camera:

$$\mathbf{X}_C = {}^{C}\mathbf{R}_W \mathbf{X} + {}^{C}\mathbf{t} = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} & t_x \\ r_{y1} & r_{y2} & r_{y3} & t_y \\ r_{z1} & r_{z2} & r_{z3} & t_z \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

where ${}^{C}\mathbf{t}$ is translation from world to camera seen from camera.

Geometric Interpretation



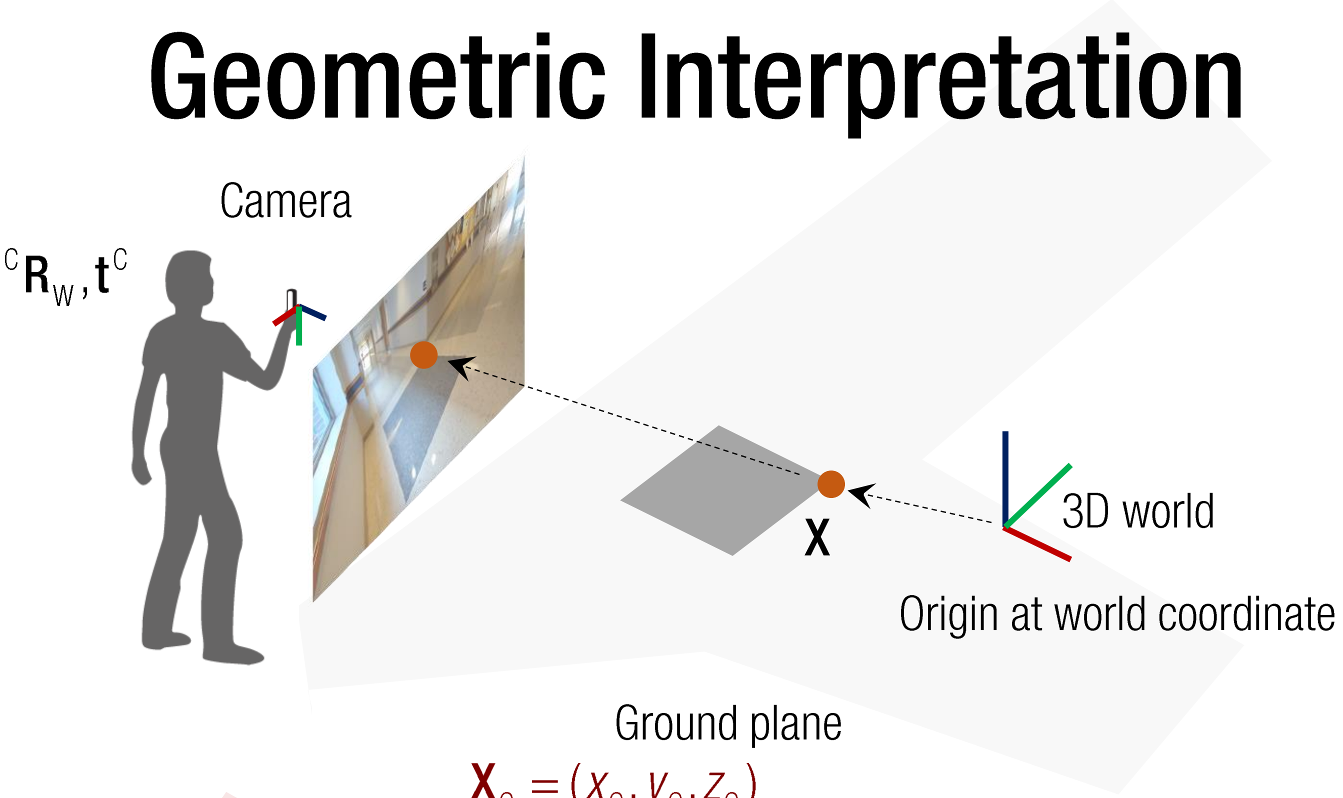
Coordinate transformation from world to camera:

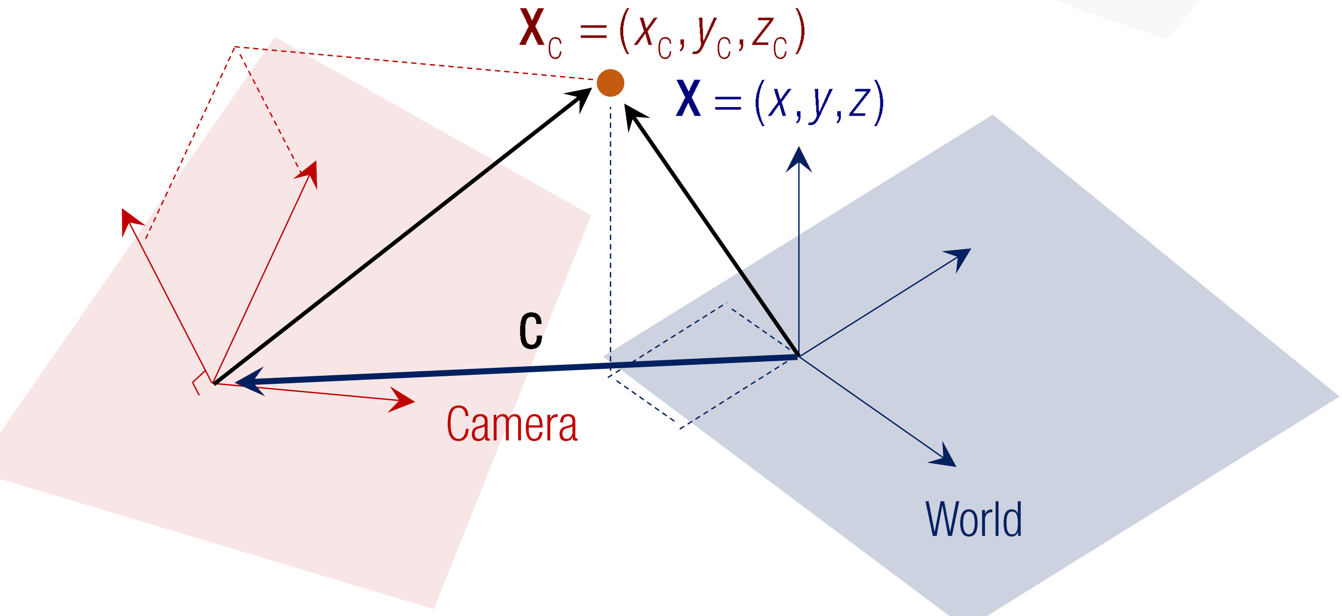
$$\mathbf{X}_C = {}^C\mathbf{R}_W \mathbf{X} + {}^C\mathbf{t} = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} & t_x \\ r_{y1} & r_{y2} & r_{y3} & t_y \\ r_{z1} & r_{z2} & r_{z3} & t_z \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

where ${}^C\mathbf{t}$ is translation from world to camera seen from camera.

Rotate and then, translate.

Geometric Interpretation





Coordinate transformation from world to camera:

$$\mathbf{X}_C = {}^{C}\mathbf{R}_{W}\mathbf{X} + {}^{C}\mathbf{t} = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} & t_x \\ r_{y1} & r_{y2} & r_{y3} & t_y \\ r_{z1} & r_{z2} & r_{z3} & t_z \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

where ${}^{C}\mathbf{t}$ is translation from world to camera seen from camera.

Rotate and then, translate.

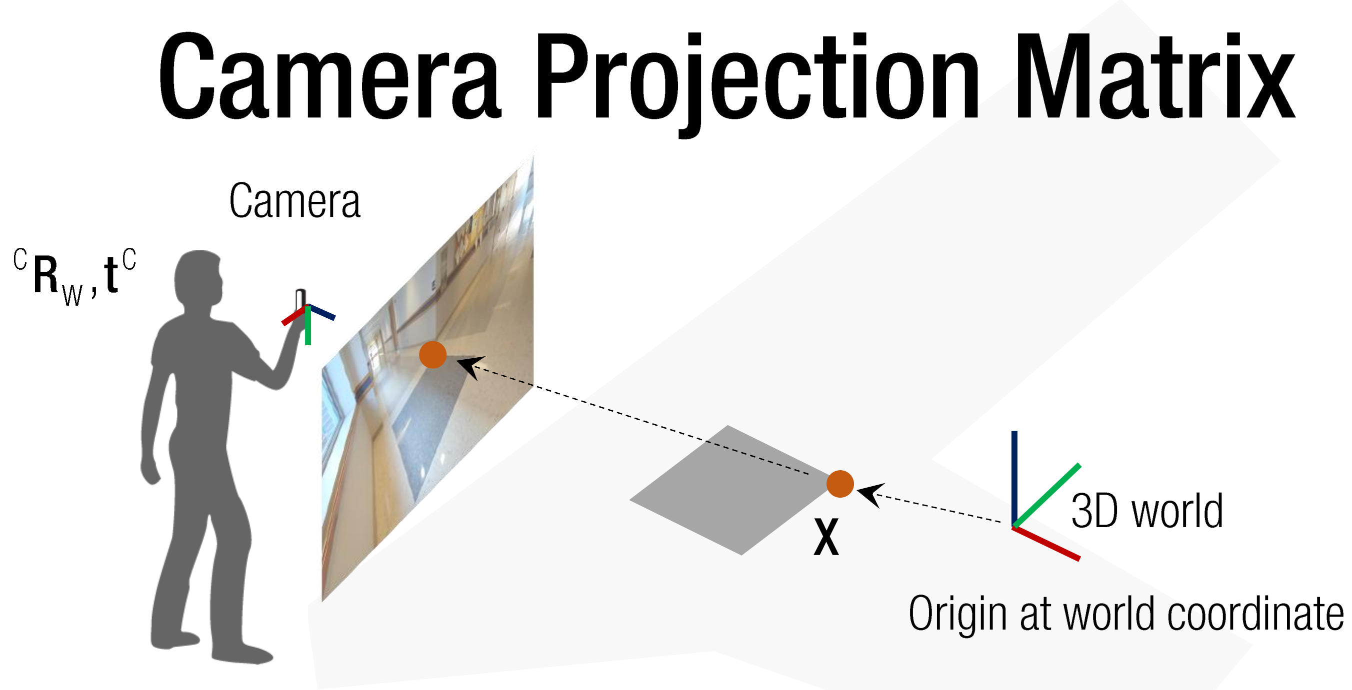
---

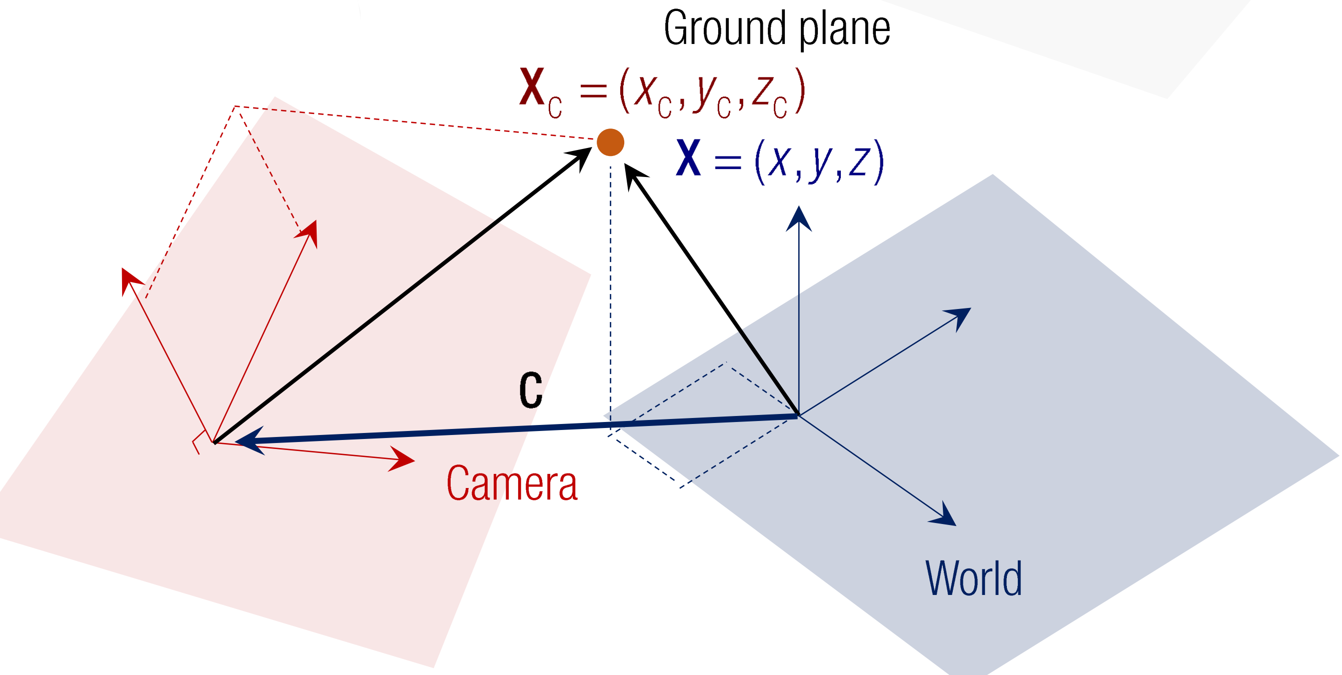
cf) **Translate and then, rotate.**

$$\mathbf{X}_C = {}^{C}\mathbf{R}_{W}(\mathbf{X} - \mathbf{C}) = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{bmatrix} \begin{bmatrix} 1 & & & -C_x \\ & 1 & & -C_y \\ & & 1 & -C_z \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

where $\mathbf{C}$ is translation from world to camera seen from world.

Camera Projection Matrix





Coordinate transformation from world to camera:

$$\mathbf{X}_C = {}^C\mathbf{R}_W\mathbf{X} + {}^C\mathbf{t} = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} & t_x \\ r_{y1} & r_{y2} & r_{y3} & t_y \\ r_{z1} & r_{z2} & r_{z3} & t_z \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

Camera projection of world point:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} \underbrace{\begin{matrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{matrix}}_{{}^C\mathbf{R}_W} & \underbrace{\begin{matrix} t_x \\ t_y \\ t_z \end{matrix}}_{{}^C\mathbf{t}} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Image Projection

$$\mathbf{C} = \begin{bmatrix} -r \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

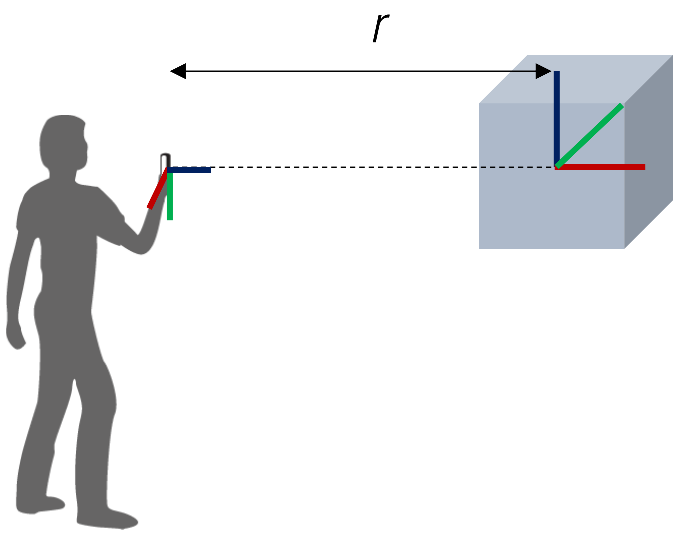
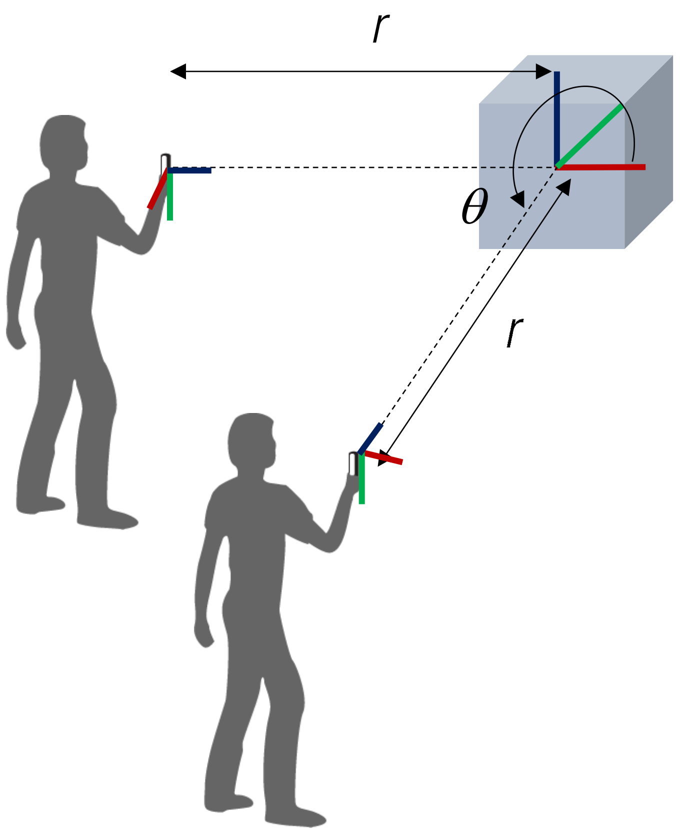
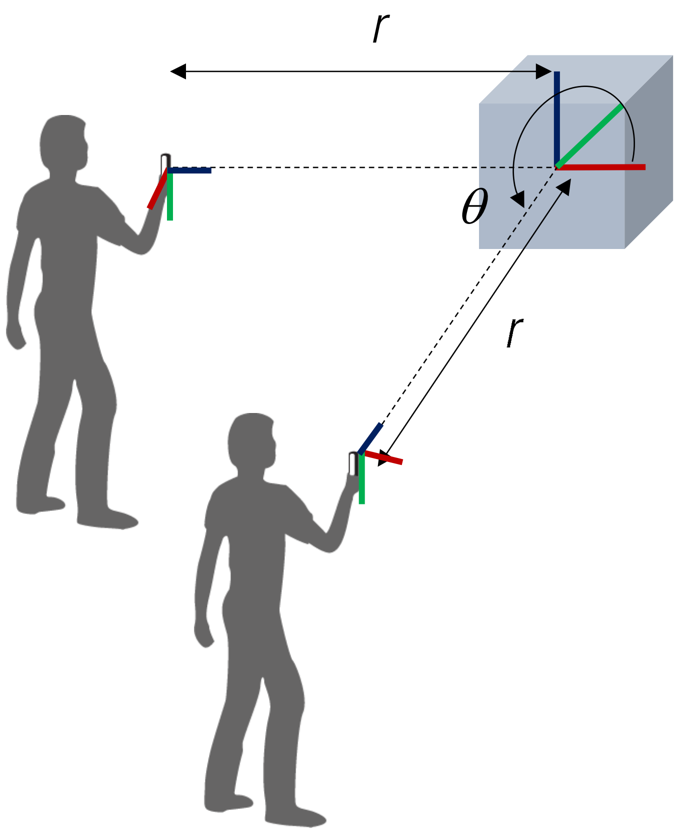


Image Projection



$$\mathbf{C} = \begin{bmatrix} -r \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

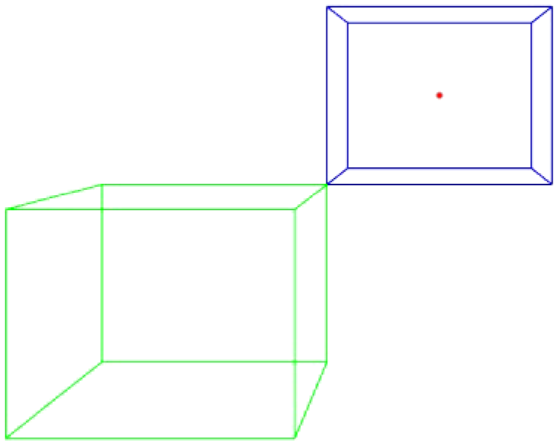
Image Projection



$$\mathbf{C} = \begin{bmatrix} -r \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

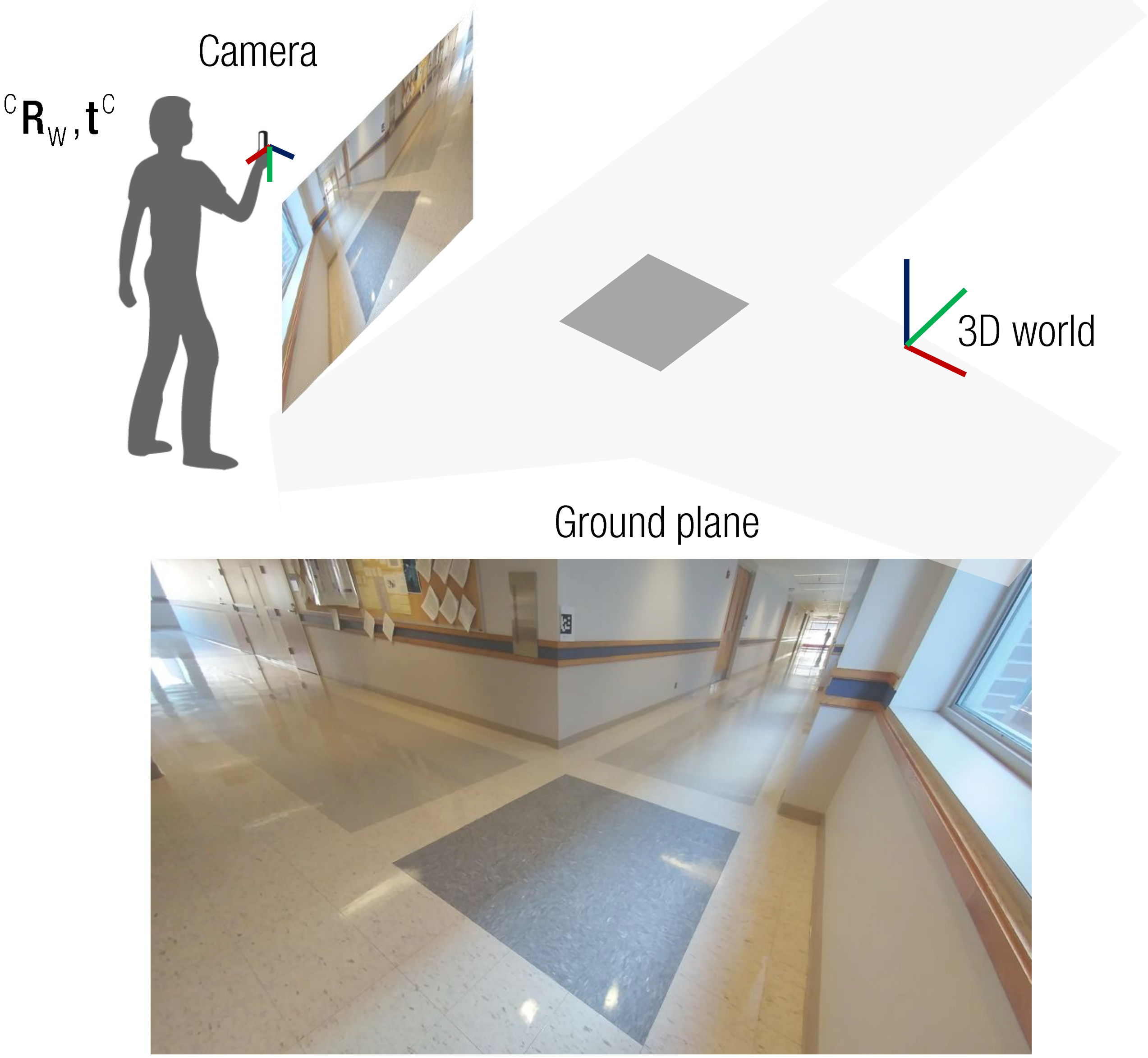
$$\mathbf{C} = \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 \\ 0 & 0 & -1 \\ -\cos\theta & -\sin\theta & 0 \end{bmatrix}$$

Image Projection



```
K = [200 0 100;
     0 200 100;
     0 0 1];

radius = 5;

theta = 0:0.02:2*pi;

for i = 1 : length(theta)
    camera_offset = [radius*cos(theta(i)); radius*sin(theta(i)); 0];
    camera_center = camera_offset + center_of_mass';

    rz = [-cos(theta(i)); -sin(theta(i)); 0];
    ry = [0 0 1]';
    rx = [-sin(theta(i)); cos(theta(i)); 0];
    R = [rx'; ry'; rz'];
    C = camera_center;
    P = K * R * [ eye(3) -C];

    proj = [];
    for j = 1 : size(sqaure_point,1)
        u = P * [sqaure_point(j,:)';1];
        proj(j,:) = u'/u(3);
    end
end
```

Geometric Interpretation



Camera projection of world point:

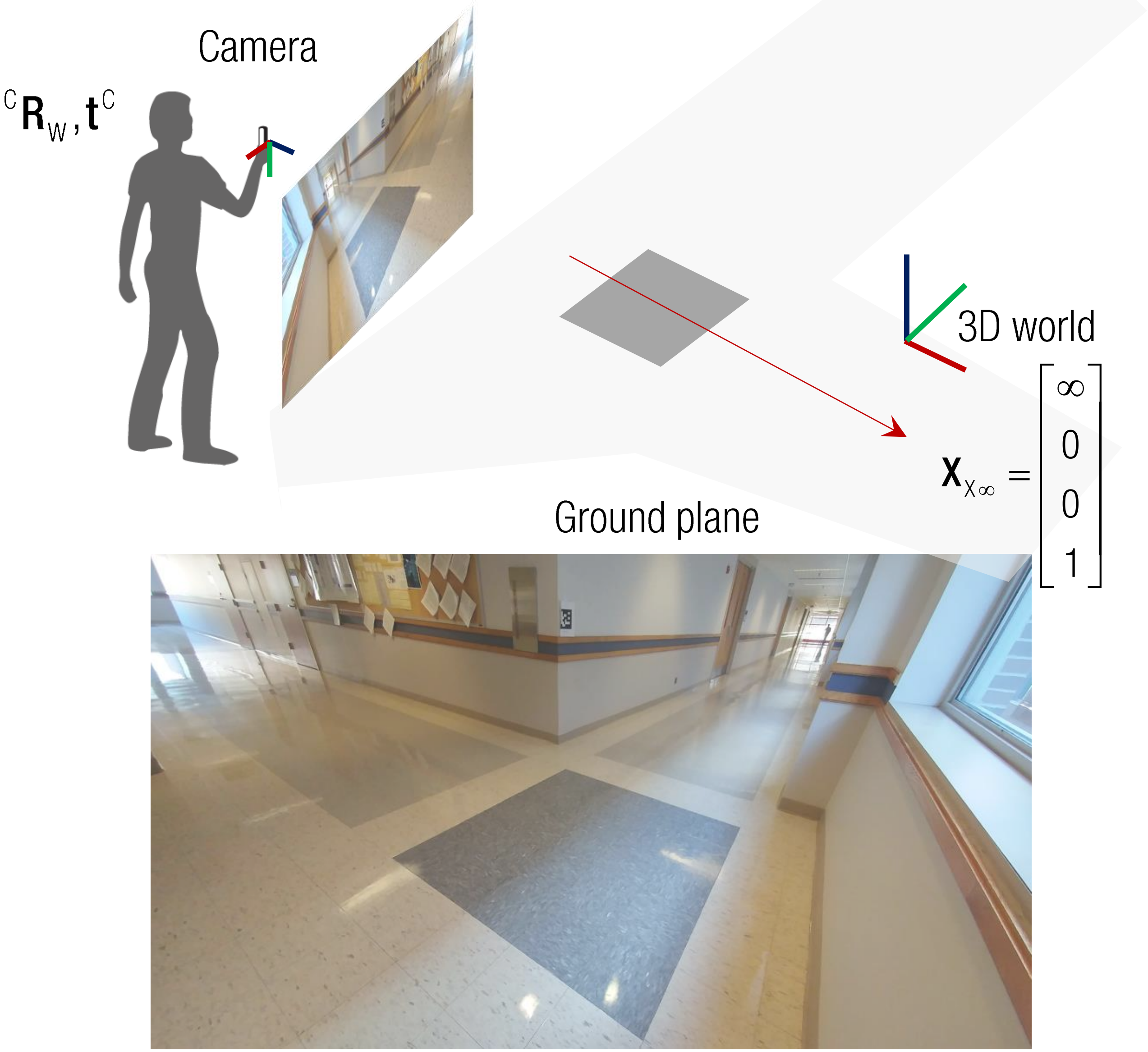
$$\lambda\begin{bmatrix}u\\ v\\ 1\end{bmatrix} = \underbrace{\begin{bmatrix}f & & p_x\\ & f & p_y\\ & & 1\end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} & t_x\\ r_{y1} & r_{y2} & r_{y3} & t_y\\ r_{z1} & r_{z2} & r_{z3} & t_z \end{bmatrix} \begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix} = \mathbf{P}\begin{bmatrix}\mathbf{X}\\ 1\end{bmatrix}$$

$$\mathbf{K}\begin{bmatrix} {}^{C}\mathbf{R}_{W} & {}^{C}\mathbf{t} \end{bmatrix}$$

$$\rightarrow \lambda\begin{bmatrix}u\\ v\\ 1\end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14}\\ p_{21} & p_{22} & p_{23} & p_{24}\\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix}$$

What does each number mean?

Geometric Interpretation



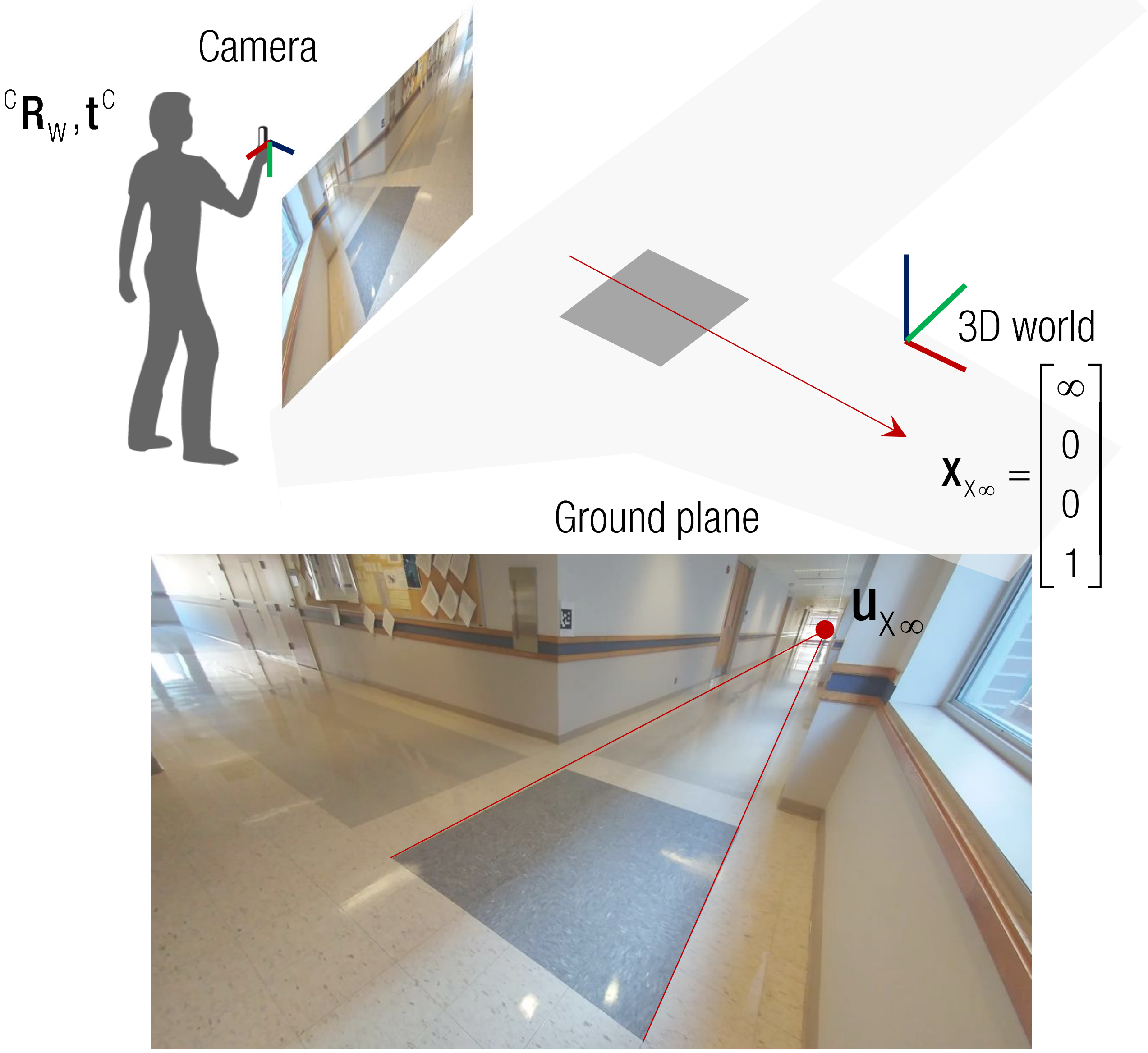
Camera projection of world point:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} \underbrace{\begin{matrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{matrix}}_{{}^{C}\mathbf{R}_{W}} & \underbrace{\begin{matrix} t_x \\ t_y \\ t_z \end{matrix}}_{{}^{C}\mathbf{t}} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\longrightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} \infty \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

What is point at infinity in world x direction?

Geometric Interpretation



Camera projection of world point:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} \underbrace{\begin{matrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{matrix}}_{{}^{C}\mathbf{R}_{W}} & \underbrace{\begin{matrix} t_x \\ t_y \\ t_z \end{matrix}}_{{}^{C}\mathbf{t}} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

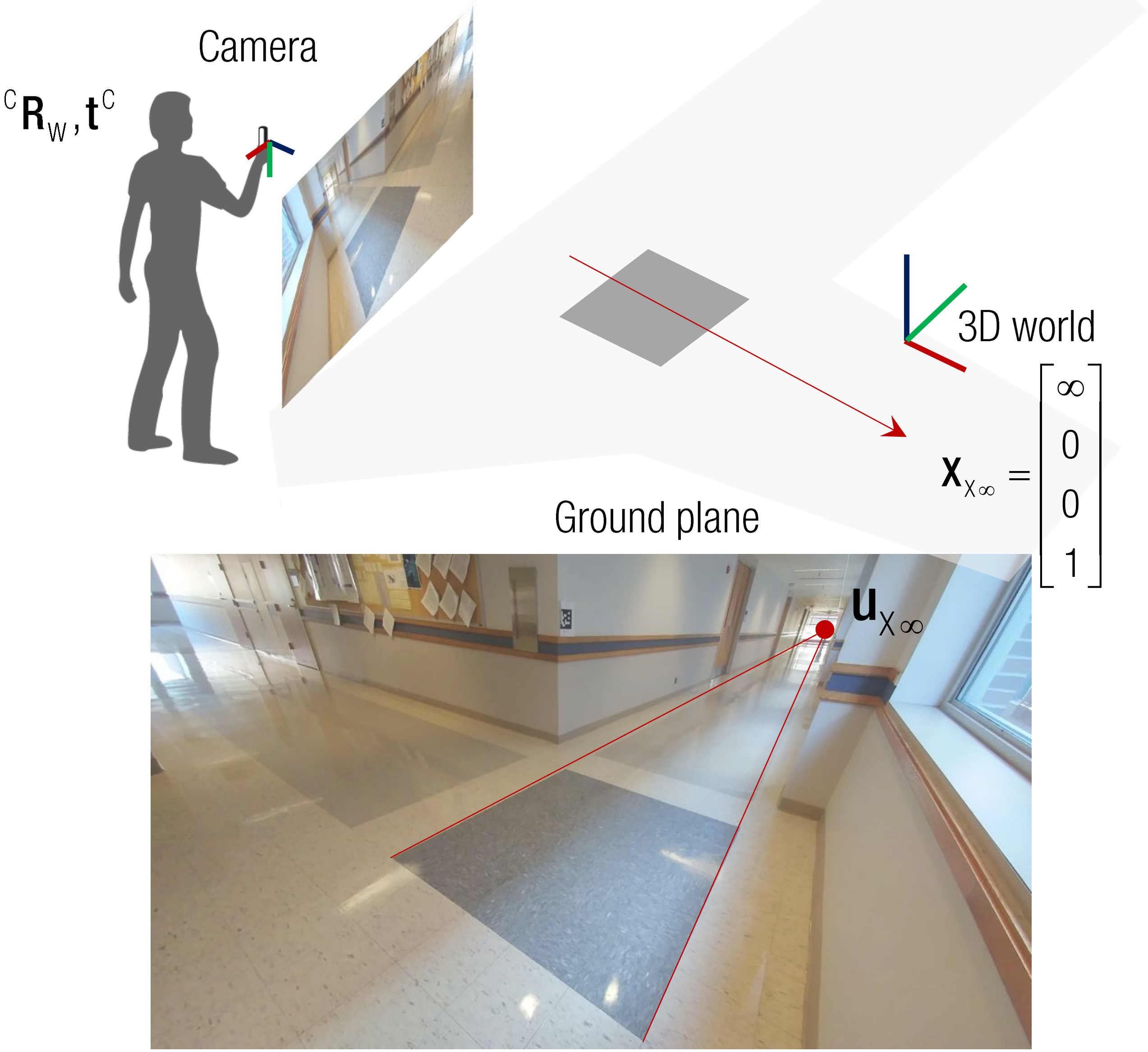
$$\longrightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} \infty \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

What is point at infinity in world x direction?

This point is at infinite but finite in image.

Point at infinity

Geometric Interpretation



Camera projection of world point:

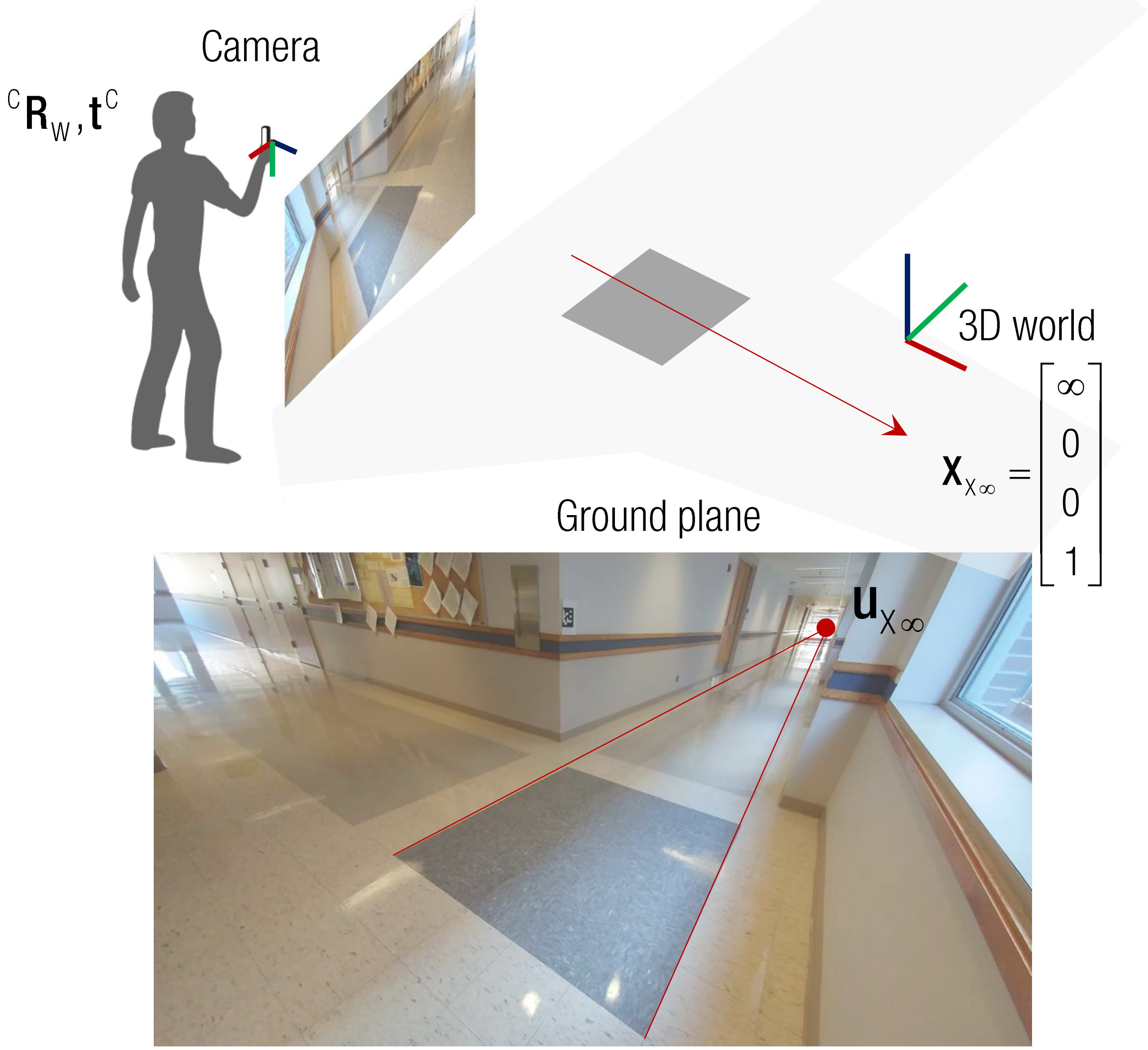
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} \underbrace{\begin{matrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{matrix}}_{^{C}\mathbf{R}_{W}} & \underbrace{\begin{matrix} t_x \\ t_y \\ t_z \end{matrix}}_{^{C}\mathbf{t}} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\longrightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} \infty \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

What is point at infinity in world x direction?

This point is at infinite but finite in image.

Geometric Interpretation



Camera projection of world point:

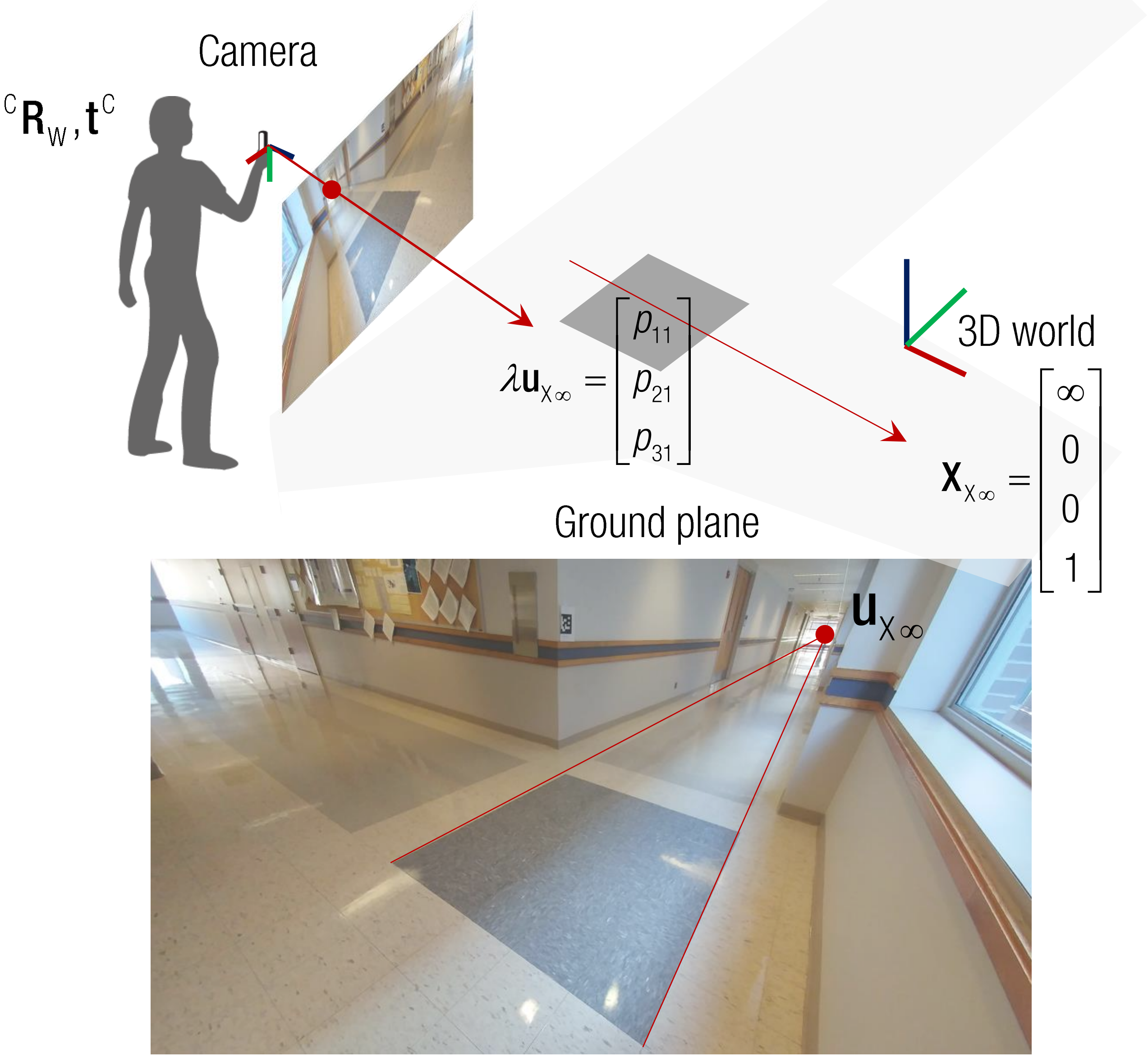
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \Bigg[\underbrace{\begin{matrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{matrix}}_{{}^{C}\mathbf{R}_{W}} \; \underbrace{\begin{matrix} t_x \\ t_y \\ t_z \end{matrix}}_{{}^{C}\mathbf{t}} \Bigg] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\longrightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} \infty \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\longrightarrow \quad u = \lim_{X \to \infty} \frac{p_{11}X + p_{14}}{p_{31}X + p_{34}} = \frac{p_{11}}{p_{31}}$$

$$v = \lim_{X \to \infty} \frac{p_{21}X + p_{24}}{p_{31}X + p_{34}} = \frac{p_{21}}{p_{31}}$$

Geometric Interpretation



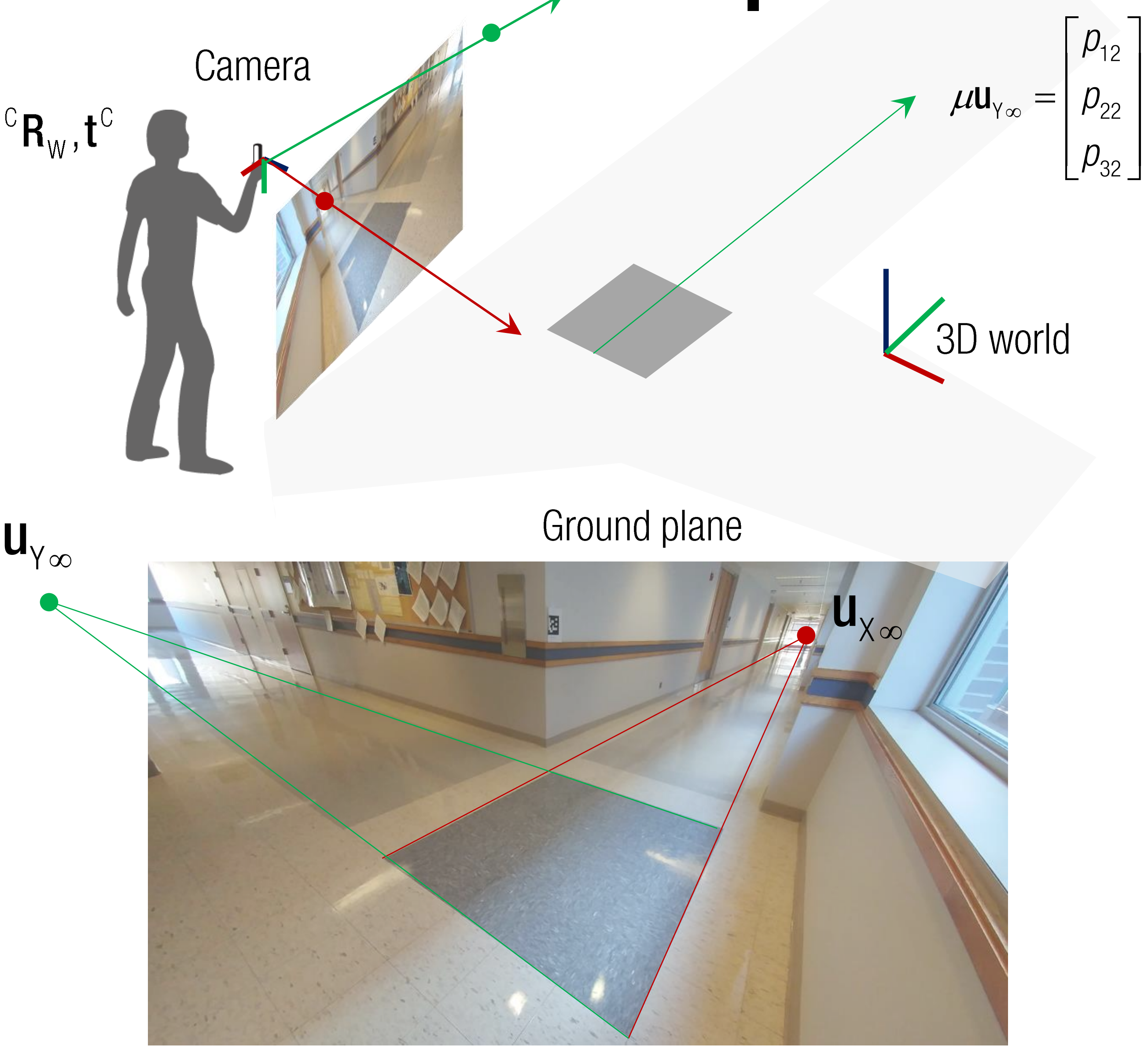
Camera projection of world point:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \left[\underbrace{\begin{matrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{matrix}}_{{}^{C}\mathbf{R}_{W}} \; \underbrace{\begin{matrix} t_x \\ t_y \\ t_z \end{matrix}}_{{}^{C}\mathbf{t}} \right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\longrightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} \infty \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\longrightarrow \begin{aligned} u &= \lim_{X \to \infty} \frac{p_{11}X + p_{14}}{p_{31}X + p_{34}} = \frac{p_{11}}{p_{31}} \\ v &= \lim_{X \to \infty} \frac{p_{21}X + p_{24}}{p_{31}X + p_{34}} = \frac{p_{21}}{p_{31}} \end{aligned} \longrightarrow \lambda \mathbf{u}_{X\infty} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \end{bmatrix}$$

Geometric Interpretation



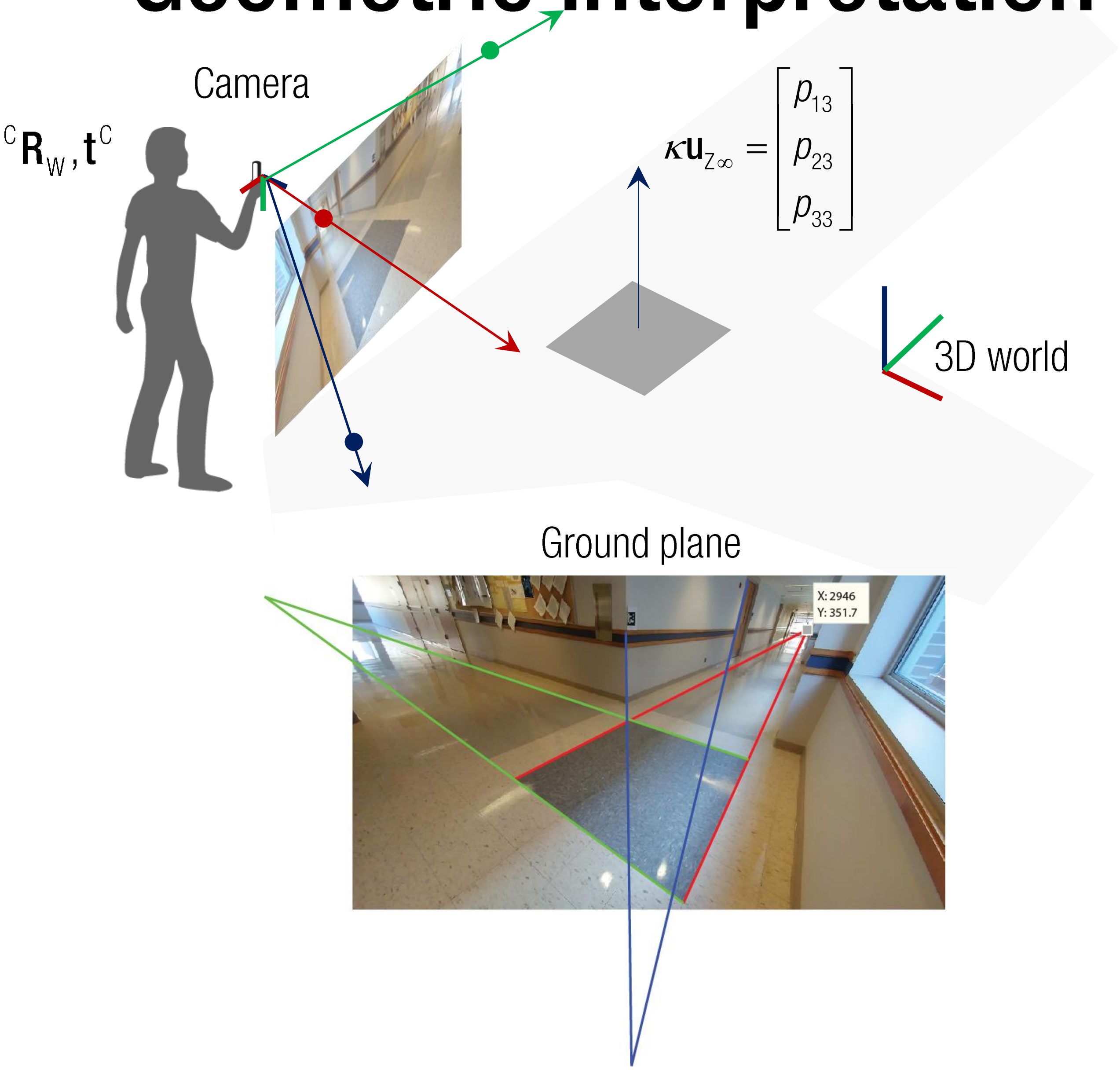
Camera projection of world point:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \Big[\underbrace{\begin{matrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{matrix}}_{{}^C\mathbf{R}_W} \; \underbrace{\begin{matrix} t_x \\ t_y \\ t_z \end{matrix}}_{{}^C\mathbf{t}} \Big] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\longrightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} 0 \\ \infty \\ 0 \\ 1 \end{bmatrix}$$

$$\longrightarrow \begin{aligned} u &= \lim_{X \to \infty} \frac{p_{12}Y + p_{14}}{p_{32}Y + p_{34}} = \frac{p_{12}}{p_{32}} \\ v &= \lim_{X \to \infty} \frac{p_{22}Y + p_{24}}{p_{32}Y + p_{34}} = \frac{p_{22}}{p_{32}} \end{aligned} \longrightarrow \mu\mathbf{u}_{Y\infty} = \mu \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{12} \\ p_{22} \\ p_{32} \end{bmatrix}$$

Geometric Interpretation



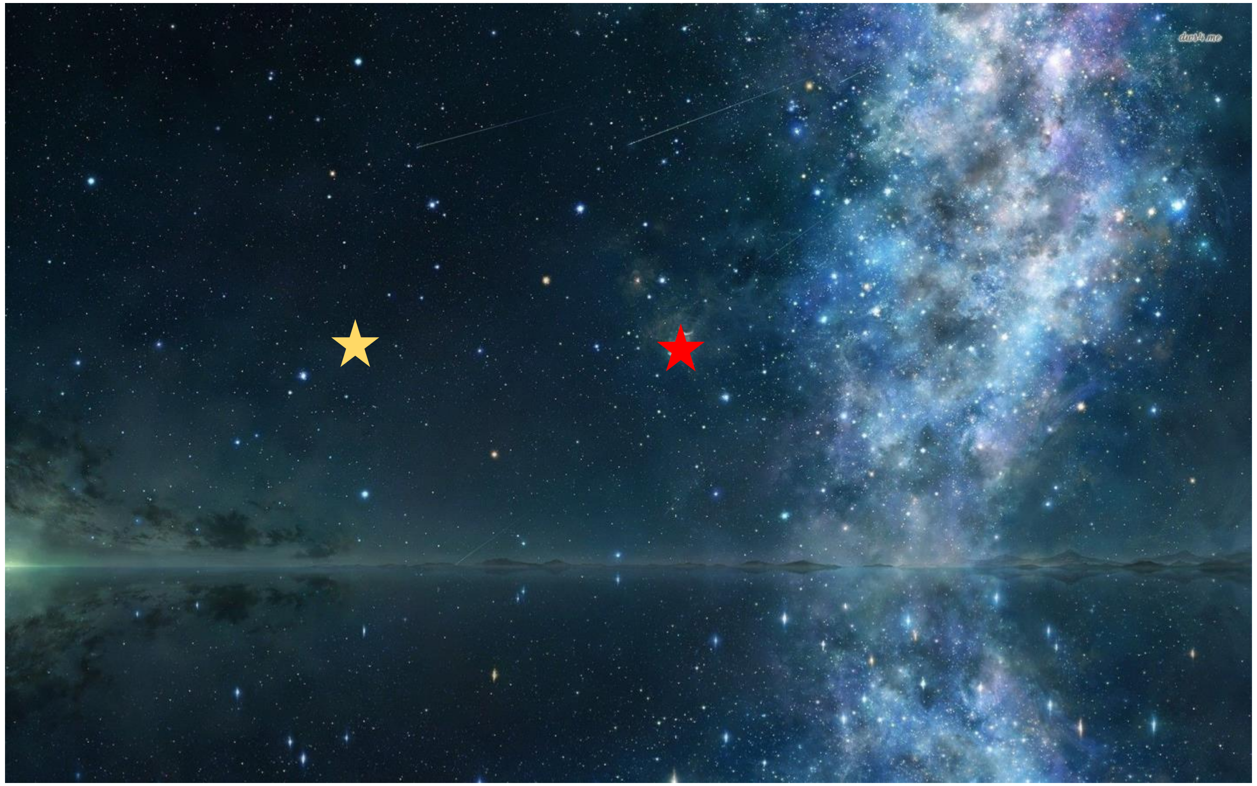
Camera projection of world point:

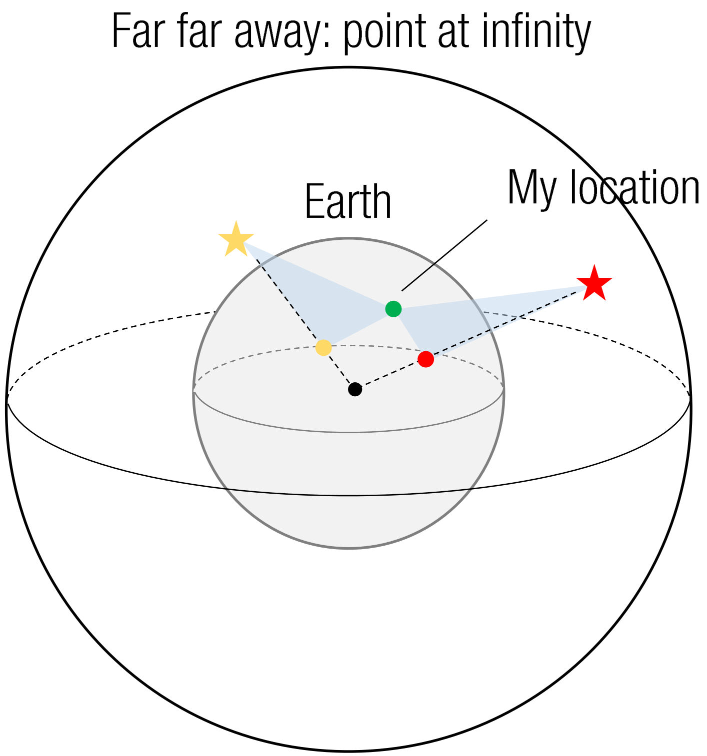
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} \underbrace{\begin{matrix} r_{x1} & r_{x2} & r_{x3} \\ r_{y1} & r_{y2} & r_{y3} \\ r_{z1} & r_{z2} & r_{z3} \end{matrix}}_{{}^{C}\mathbf{R}_{W}} & \underbrace{\begin{matrix} t_x \\ t_y \\ t_z \end{matrix}}_{{}^{C}\mathbf{t}} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\longrightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \infty \\ 1 \end{bmatrix}$$

$$\longrightarrow \begin{aligned} u &= \lim_{X \to \infty} \frac{p_{13}Z + p_{14}}{p_{33}Z + p_{34}} = \frac{p_{13}}{p_{33}} \\ v &= \lim_{X \to \infty} \frac{p_{23}Z + p_{24}}{p_{33}Z + p_{34}} = \frac{p_{23}}{p_{33}} \end{aligned} \longrightarrow \kappa \mathbf{u}_{Z\infty} = \kappa \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_{13} \\ p_{23} \\ p_{33} \end{bmatrix}$$

Celestial Navigation





Practice

$$f = f_{\mathrm{m}} \frac{W_{\mathrm{img}}}{W_{\mathrm{ccd}}} = 0.00153 \frac{3840}{0.0048} = 1224\mathrm{pix}$$

$$p_x = \frac{W_{\mathrm{img}}}{2} = \frac{3840}{2} = 1920\mathrm{pix} \qquad p_y = \frac{H_{\mathrm{img}}}{2} = \frac{2160}{2} = 1080\mathrm{pix}$$

$$\mathbf{C} = \begin{bmatrix} 0.0070 \\ 0.7520 \\ -0.2738 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} -0.8496 & 0.0498 & 0.5731 \\ -0.3216 & -0.8203 & -0.4067 \\ 0.4180 & -0.5299 & 0.6835 \end{bmatrix}$$

Practice

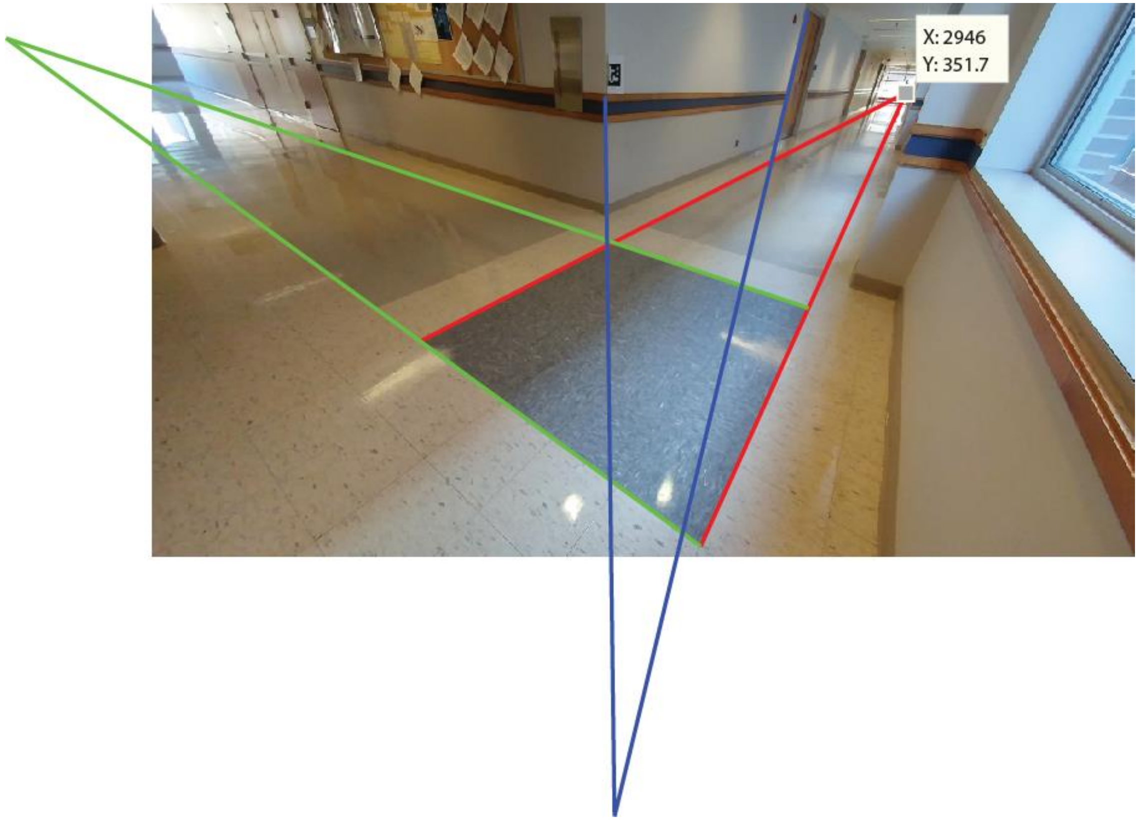
$$f = f_m \frac{W_{img}}{W_{ccd}} = 0.00153 \frac{3840}{0.0048} = 1224\text{pix}$$

$$p_x = \frac{W_{img}}{2} = \frac{3840}{2} = 1920\text{pix} \qquad p_y = \frac{H_{img}}{2} = \frac{2160}{2} = 1080\text{pix}$$

$$\mathbf{C} = \begin{bmatrix} 0.0070 \\ 0.7520 \\ -0.2738 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} -0.8496 & 0.0498 & 0.5731 \\ -0.3216 & -0.8203 & -0.4067 \\ 0.4180 & -0.5299 & 0.6835 \end{bmatrix}$$

$$\mathbf{P} = \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix} = \begin{bmatrix} -0.2374 & -0.9565 & 2.0138 & 1.2723 \\ 0.0578 & -1.5763 & 0.2404 & 1.2508 \\ 0.0004 & -0.0005 & 0.0007 & 0.0006 \end{bmatrix}$$

Practice



$$f = f_m \frac{W_{img}}{W_{ccd}} = 0.00153 \frac{3840}{0.0048} = 1224\text{pix}$$

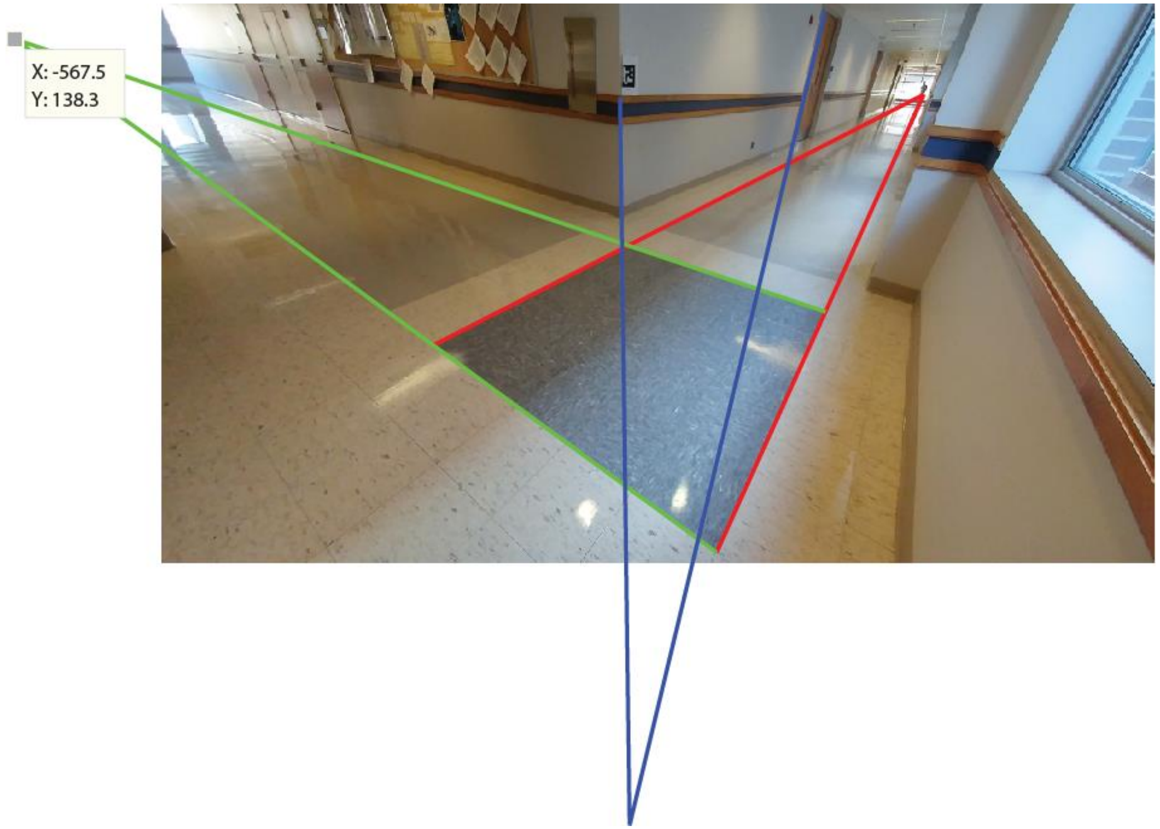
$$p_x = \frac{W_{img}}{2} = \frac{3840}{2} = 1920\text{pix} \qquad p_y = \frac{H_{img}}{2} = \frac{2160}{2} = 1080\text{pix}$$

$$\mathbf{C} = \begin{bmatrix} 0.0070 \\ 0.7520 \\ -0.2738 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} -0.8496 & 0.0498 & 0.5731 \\ -0.3216 & -0.8203 & -0.4067 \\ 0.4180 & -0.5299 & 0.6835 \end{bmatrix}$$

$$\mathbf{P} = \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix} = \begin{bmatrix} -0.2374 & -0.9565 & 2.0138 & 1.2723 \\ 0.0578 & -1.5763 & 0.2404 & 1.2508 \\ 0.0004 & -0.0005 & 0.0007 & 0.0006 \end{bmatrix}$$

```
R = [-0.8496   0.0498   0.5731
   -0.3216  -0.8203  -0.4067
    0.4180  -0.5299   0.6835];

C = [0.0070
    0.7520
   -0.2738];

P = K * R * [eye(3) -C]
u_x = P(1:2,1)/P(3,1)
u_y = P(1:2,2)/P(3,2)
u_z = P(1:2,3)/P(3,3)
```

Practice



$$f = f_m \frac{W_{img}}{W_{ccd}} = 0.00153 \frac{3840}{0.0048} = 1224\text{pix}$$

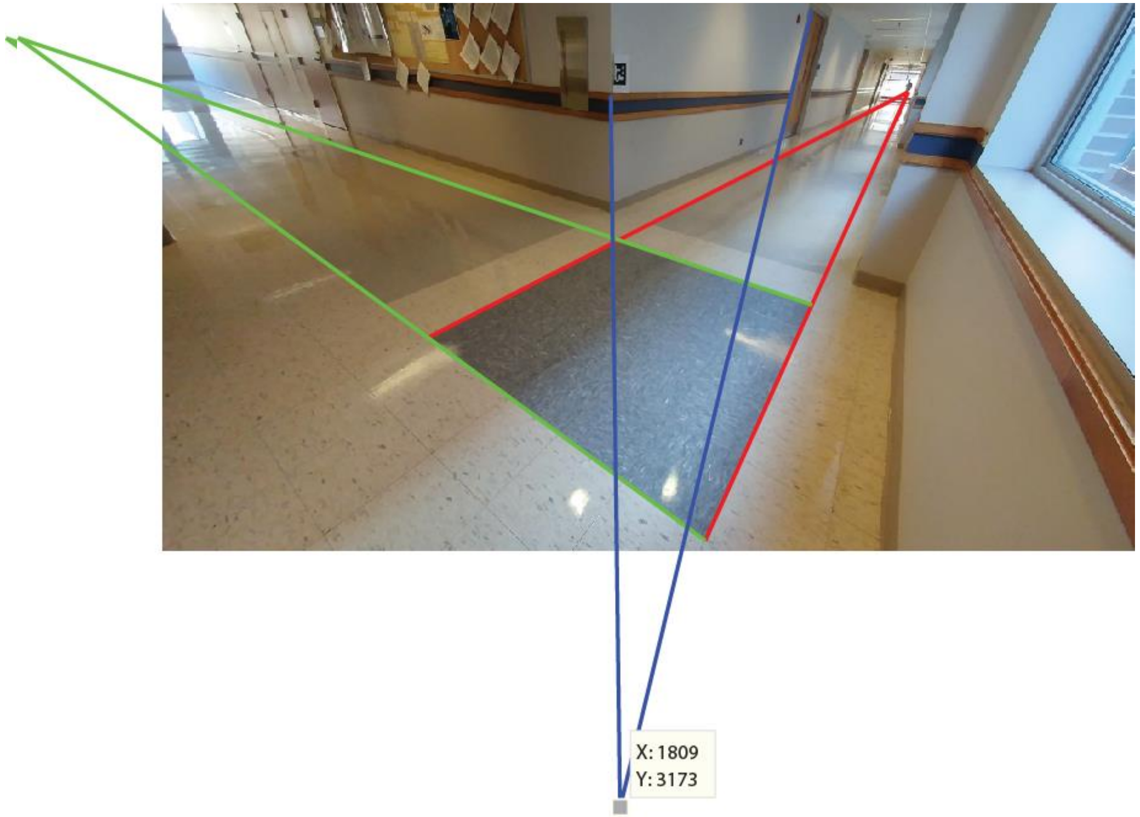
$$p_x = \frac{W_{img}}{2} = \frac{3840}{2} = 1920\text{pix} \qquad p_y = \frac{H_{img}}{2} = \frac{2160}{2} = 1080\text{pix}$$

$$\mathbf{C} = \begin{bmatrix} 0.0070 \\ 0.7520 \\ -0.2738 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} -0.8496 & 0.0498 & 0.5731 \\ -0.3216 & -0.8203 & -0.4067 \\ 0.4180 & -0.5299 & 0.6835 \end{bmatrix}$$

$$\mathbf{P} = \mathbf{KR}\begin{bmatrix}\mathbf{I}_3 & -\mathbf{C}\end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix}-0.2374 \\ 0.0578 \\ 0.0004\end{matrix}} & \begin{matrix}-0.9565 \\ -1.5763 \\ -0.0005\end{matrix} & \begin{matrix}2.0138 \\ 0.2404 \\ 0.0007\end{matrix} & \begin{matrix}1.2723 \\ 1.2508 \\ 0.0006\end{matrix} \end{bmatrix}$$

```
R = [-0.8496   0.0498   0.5731
  -0.3216  -0.8203  -0.4067
   0.4180  -0.5299   0.6835];

C = [0.0070
   0.7520
  -0.2738];

P = K * R * [eye(3) -C]
u_x = P(1:2,1)/P(3,1)
u_y = P(1:2,2)/P(3,2)
u_z = P(1:2,3)/P(3,3)
```

Practice



$$f = f_m \frac{W_{img}}{W_{ccd}} = 0.00153 \frac{3840}{0.0048} = 1224\text{pix}$$

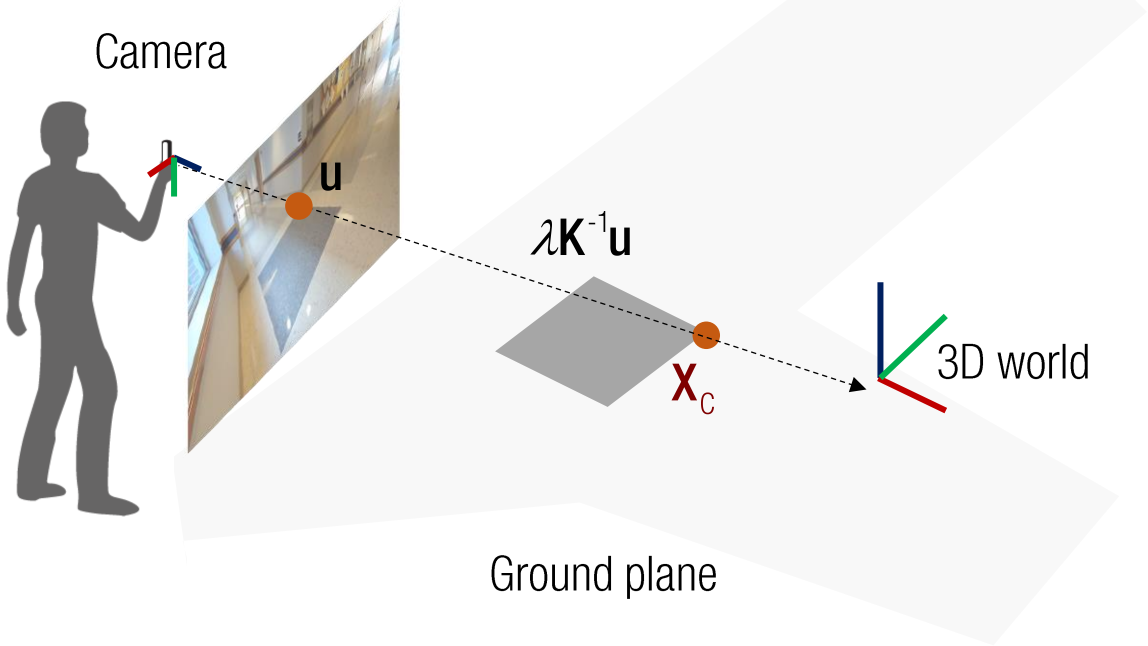
$$p_x = \frac{W_{img}}{2} = \frac{3840}{2} = 1920\text{pix} \qquad p_y = \frac{H_{img}}{2} = \frac{2160}{2} = 1080\text{pix}$$

$$\mathbf{C} = \begin{bmatrix} 0.0070 \\ 0.7520 \\ -0.2738 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} -0.8496 & 0.0498 & 0.5731 \\ -0.3216 & -0.8203 & -0.4067 \\ 0.4180 & -0.5299 & 0.6835 \end{bmatrix}$$

$$\mathbf{P} = \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix} = \begin{bmatrix} -0.2374 & -0.9565 & 2.0138 & 1.2723 \\ 0.0578 & -1.5763 & 0.2404 & 1.2508 \\ 0.0004 & -0.0005 & 0.0007 & 0.0006 \end{bmatrix}$$

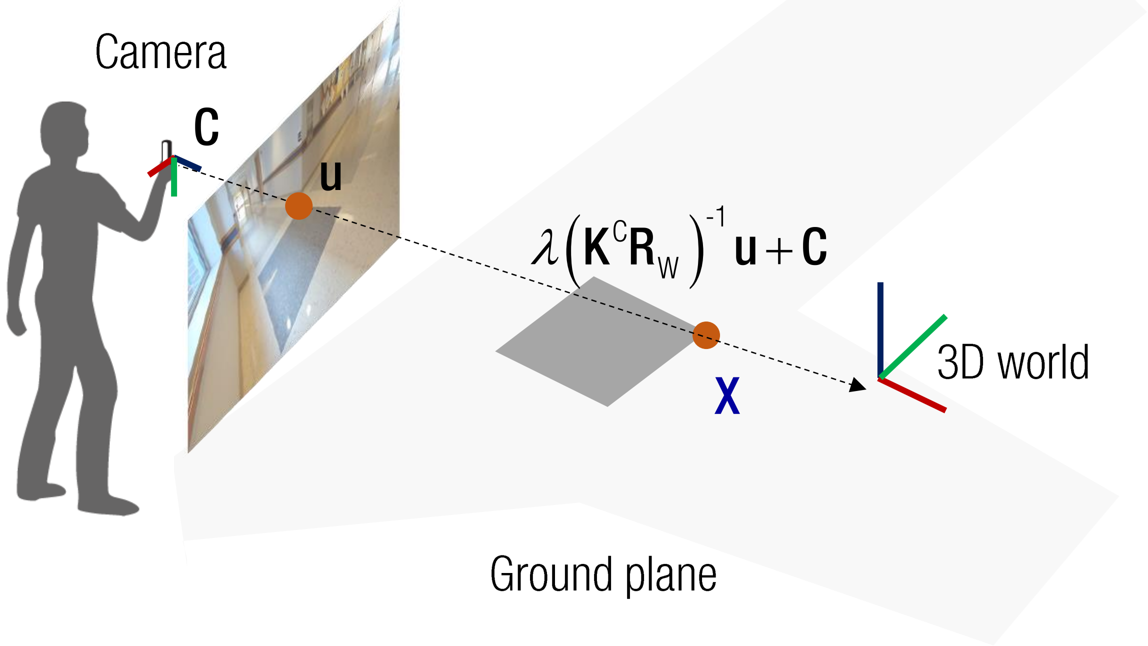
```
R = [-0.8496   0.0498   0.5731
  -0.3216  -0.8203  -0.4067
   0.4180  -0.5299   0.6835];

C = [0.0070
   0.7520
  -0.2738];

P = K * R * [eye(3) -C]
u_x = P(1:2,1)/P(3,1)
u_y = P(1:2,2)/P(3,2)
u_z = P(1:2,3)/P(3,3)
```

Inverse of Camera Projection Matrix



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{X}_C$$

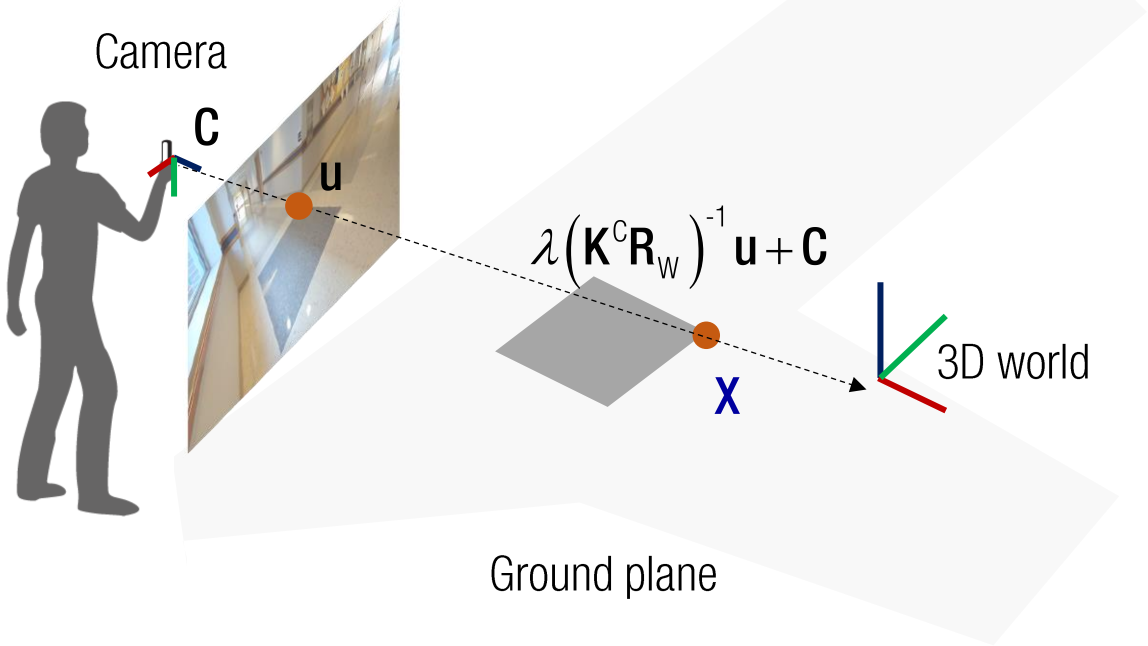
Inverse of Camera Projection Matrix



$$\lambda\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{X}_{C}$$

$$= \mathbf{K}\,{}^{C}\mathbf{R}_{W}\mathbf{X} + {}^{C}\mathbf{t} = \mathbf{K}\,{}^{C}\mathbf{R}_{W}(\mathbf{X} - \mathbf{C})$$

Inverse of Camera Projection Matrix

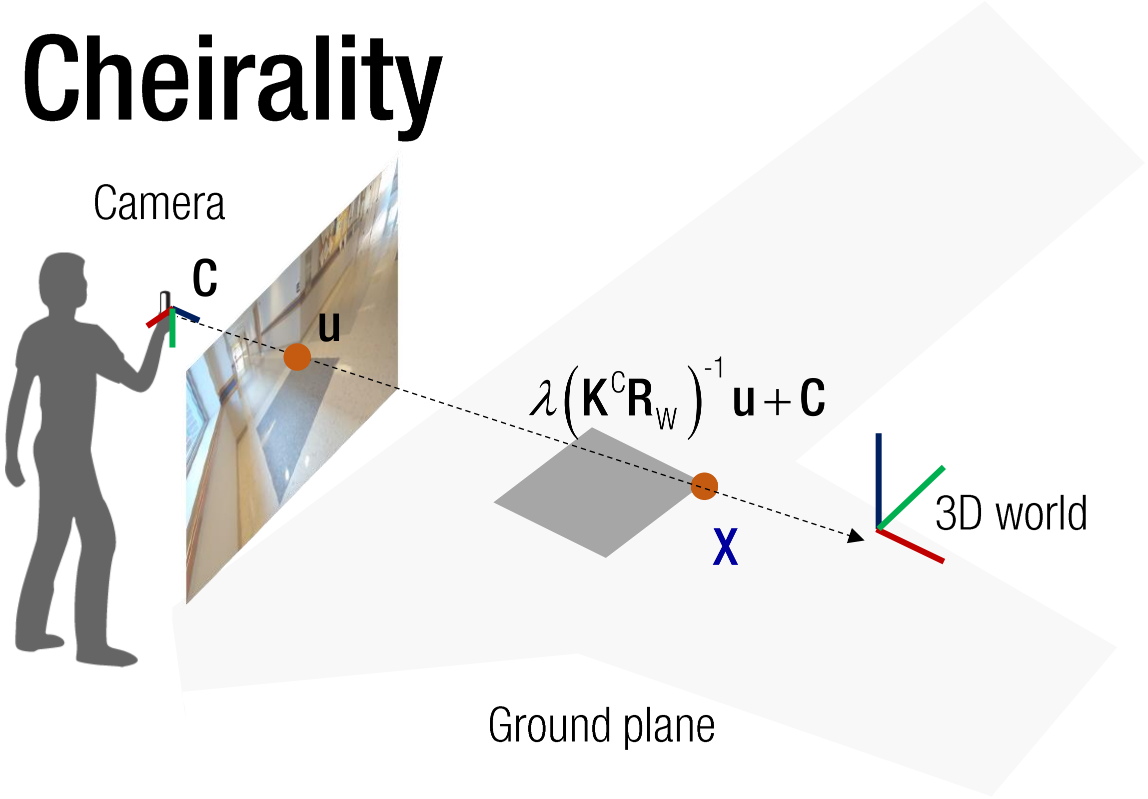


$$\lambda\begin{bmatrix}u\\v\\1\end{bmatrix}=\mathbf{K}\mathbf{X}_C$$

$$=\mathbf{K}^{C}\mathbf{R}_{W}\mathbf{X}+{}^{C}\mathbf{t}=\mathbf{K}^{C}\mathbf{R}_{W}(\mathbf{X}-\mathbf{C})$$

$$\longrightarrow\quad \mathbf{X}=\underbrace{\lambda\left(\mathbf{K}^{C}\mathbf{R}_{W}\right)^{-1}\begin{bmatrix}u\\v\\1\end{bmatrix}}_{\text{3D ray direction}}+\underbrace{\mathbf{C}}_{\text{3D ray origin}}$$

Cheirality



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{X}_{C}$$

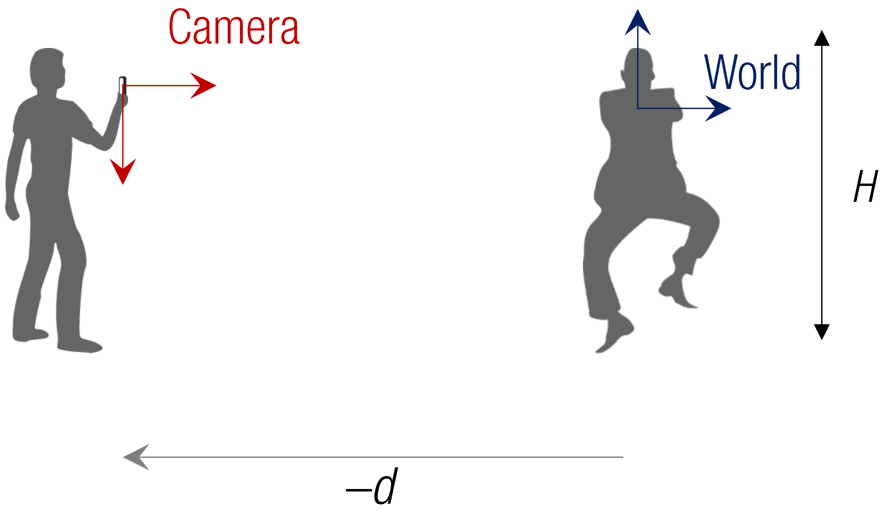
$$= \mathbf{K}\,{}^{C}\mathbf{R}_{W}\mathbf{X} + {}^{C}\mathbf{t} = \mathbf{K}\,{}^{C}\mathbf{R}_{W}(\mathbf{X}-\mathbf{C})$$

$$\longrightarrow \quad \mathbf{X} = \underbrace{\lambda\left(\mathbf{K}\,{}^{C}\mathbf{R}_{W}\right)^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\text{3D ray direction}} + \underbrace{\mathbf{C}}_{\text{3D ray origin}}$$

where $\lambda > 0$

Dolly Zoom Camera Matrix

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

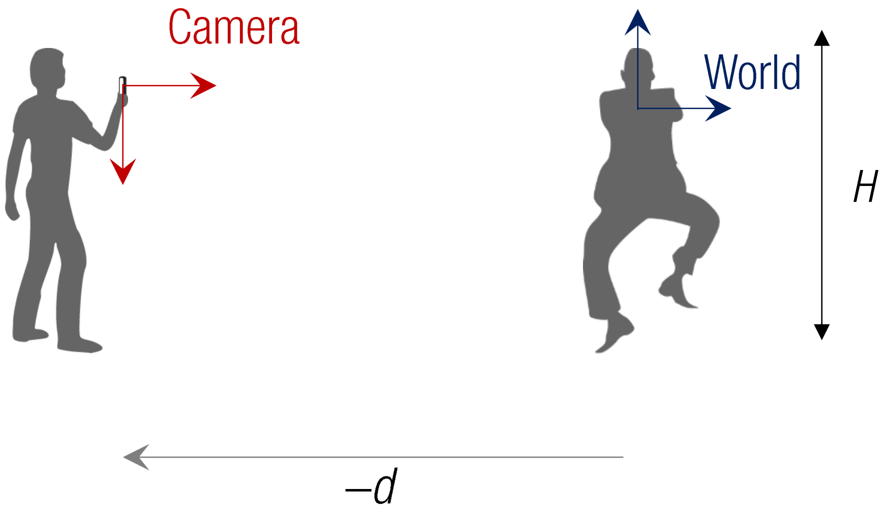


Dolly Zoom Camera Matrix

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KR}(\mathbf{X} - \mathbf{C}) \qquad \text{: translate and then, rotate}$$

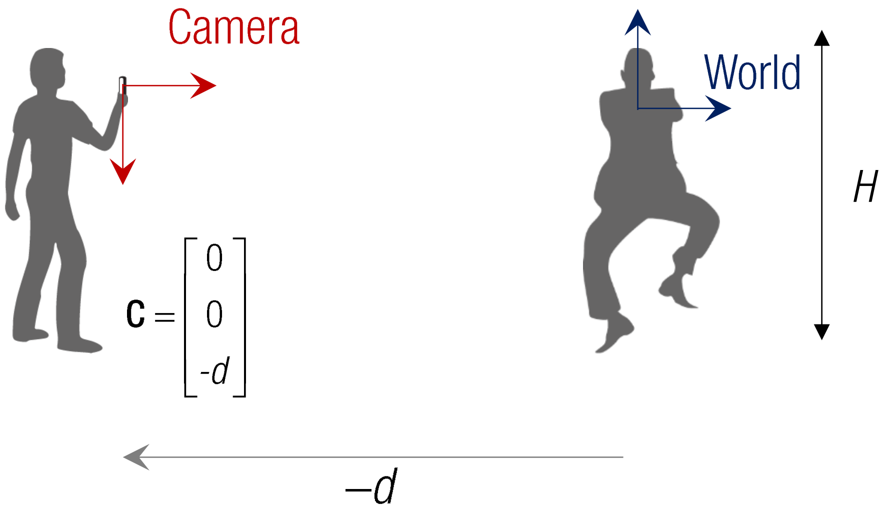
$$= \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

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



Dolly Zoom Camera Matrix

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



$$\mathbf{C} = \begin{bmatrix} 0 \\ 0 \\ -d \end{bmatrix}$$

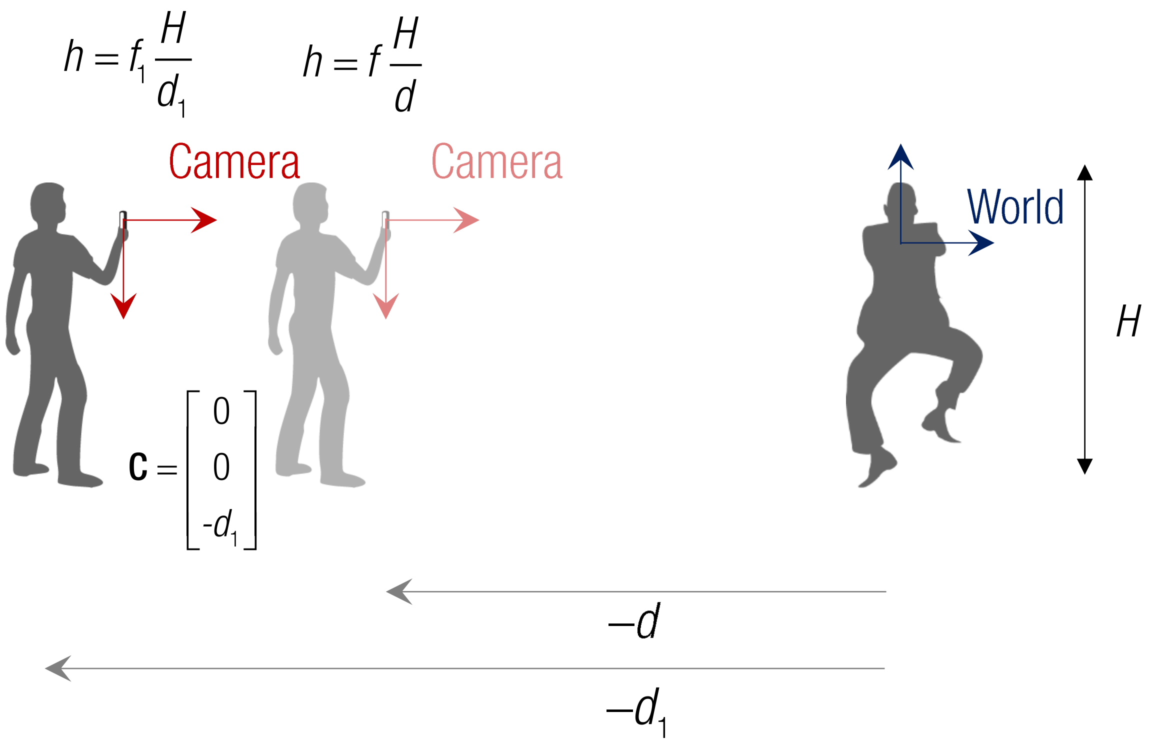
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KR}(\mathbf{X} - \mathbf{C})$$: translate and then, rotate

$$= \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}\underline{\begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}}\begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

Camera z axis (optical axis) is aligned with the world z coordinate.

Dolly Zoom Camera Matrix



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KR}(\mathbf{X} - \mathbf{C})$$: translate and then, rotate

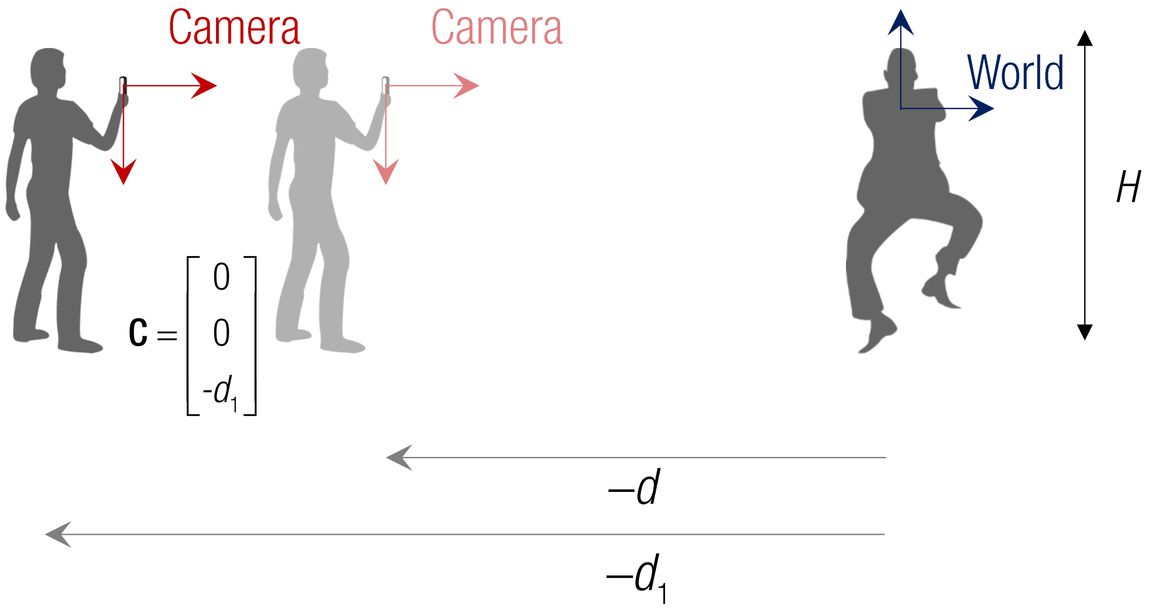
$$= \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \underline{\begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

Camera z axis (optical axis) is aligned with the world z coordinate.

Dolly Zoom Camera Matrix

$$h = f_1 \frac{H}{d_1} \qquad h = f \frac{H}{d}$$



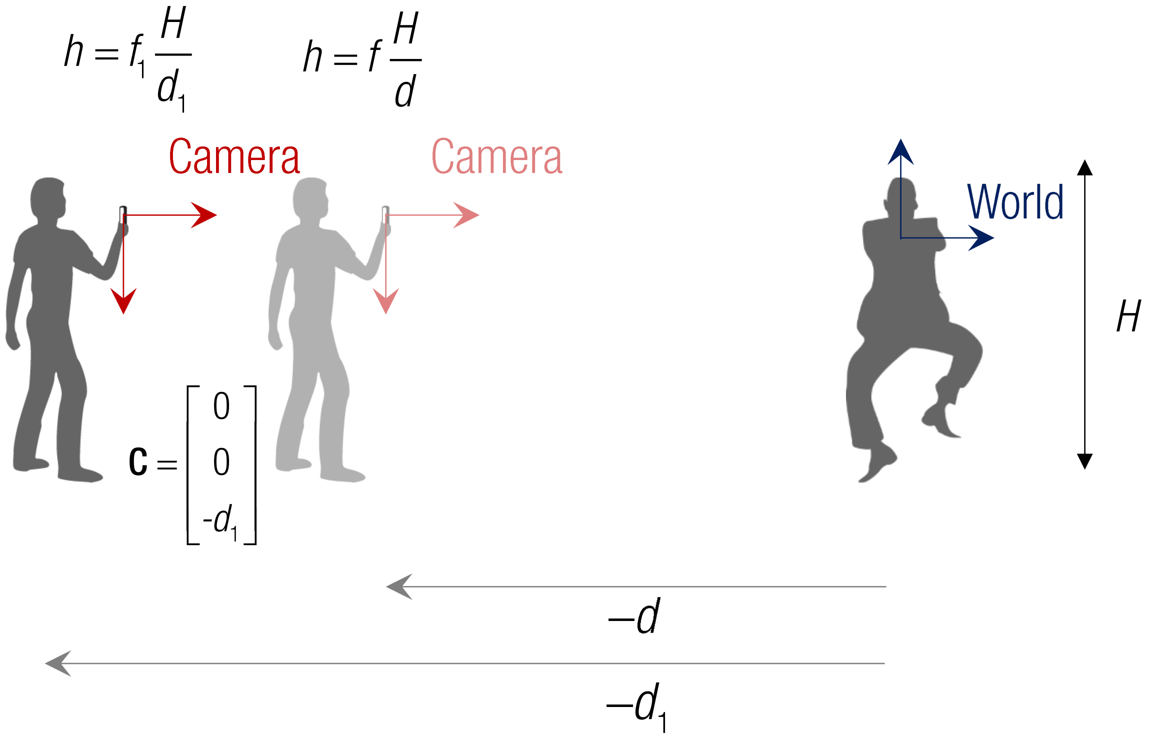
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KR}(\mathbf{X} - \mathbf{C})$$: translate and then, rotate

$$= \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

?

Dolly Zoom Camera Matrix



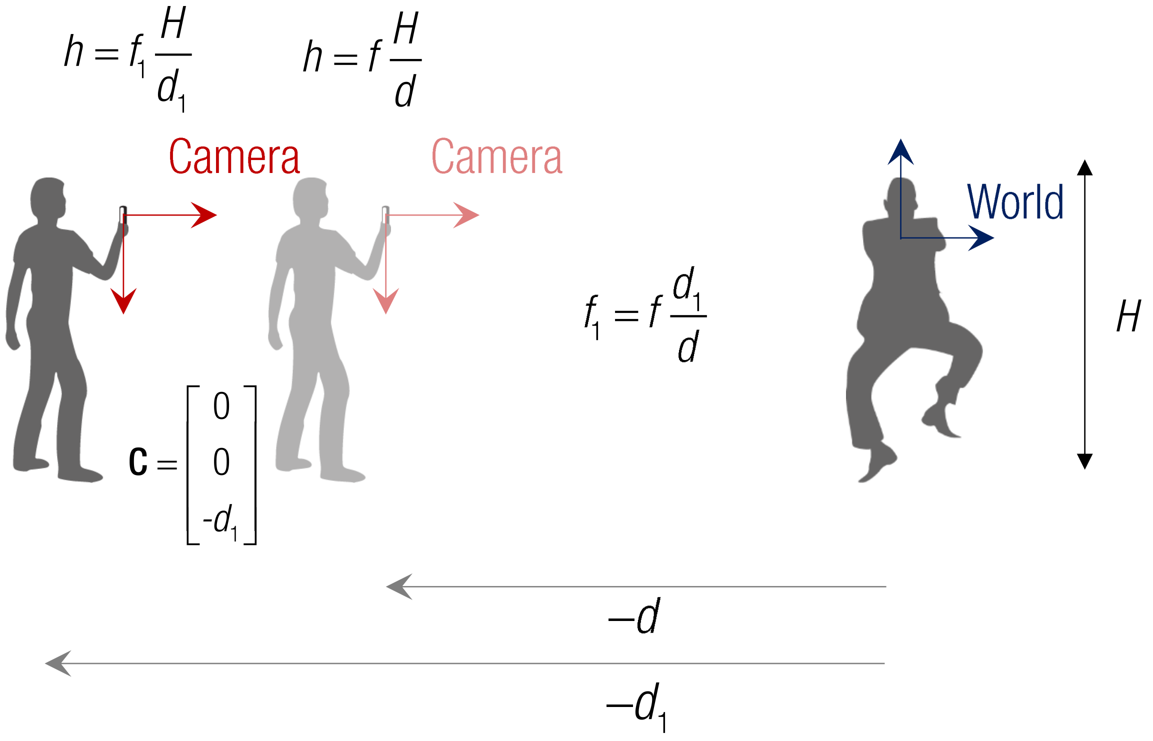
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KR}(\mathbf{X} - \mathbf{C})$$: translate and then, rotate

$$= \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f_1 & & p_x \\ & f_1 & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d_1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

Dolly Zoom Camera Matrix



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KR}(\mathbf{X} - \mathbf{C}) \qquad \text{: translate and then, rotate}$$

$$= \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

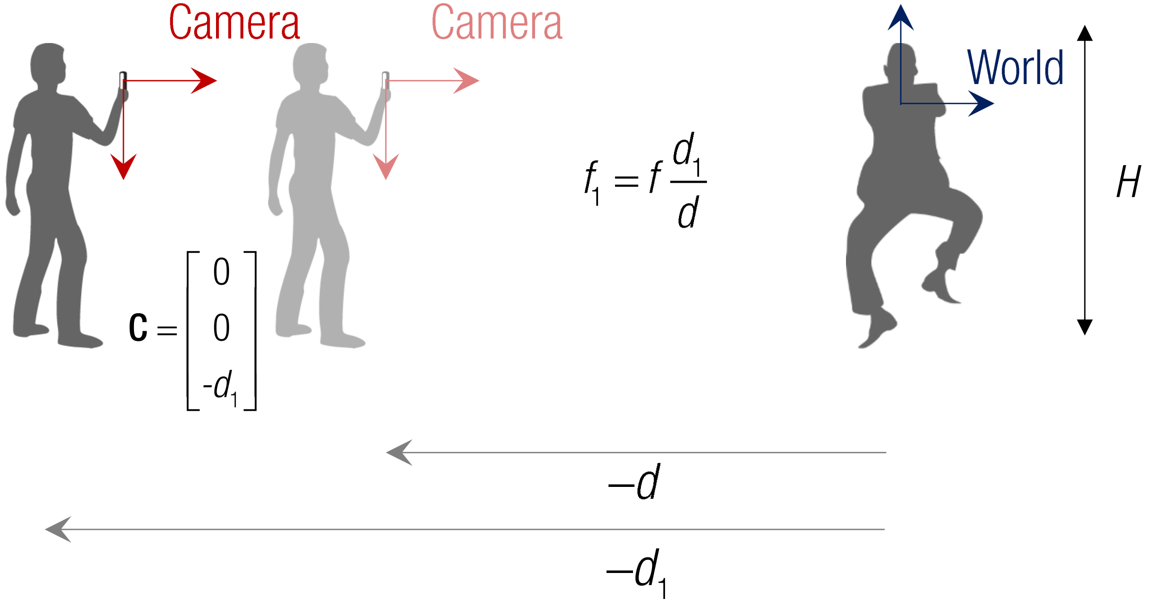
$$= \begin{bmatrix} f_1 & & p_x \\ & f_1 & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d_1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f \dfrac{d_1}{d} & & p_x \\ & f \dfrac{d_1}{d} & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d_1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

Dolly Zoom Camera Matrix

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f\frac{d_1}{d} & & p_x \\ & f\frac{d_1}{d} & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d_1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$h = f_1 \frac{H}{d_1} \qquad h = f\frac{H}{d}$$



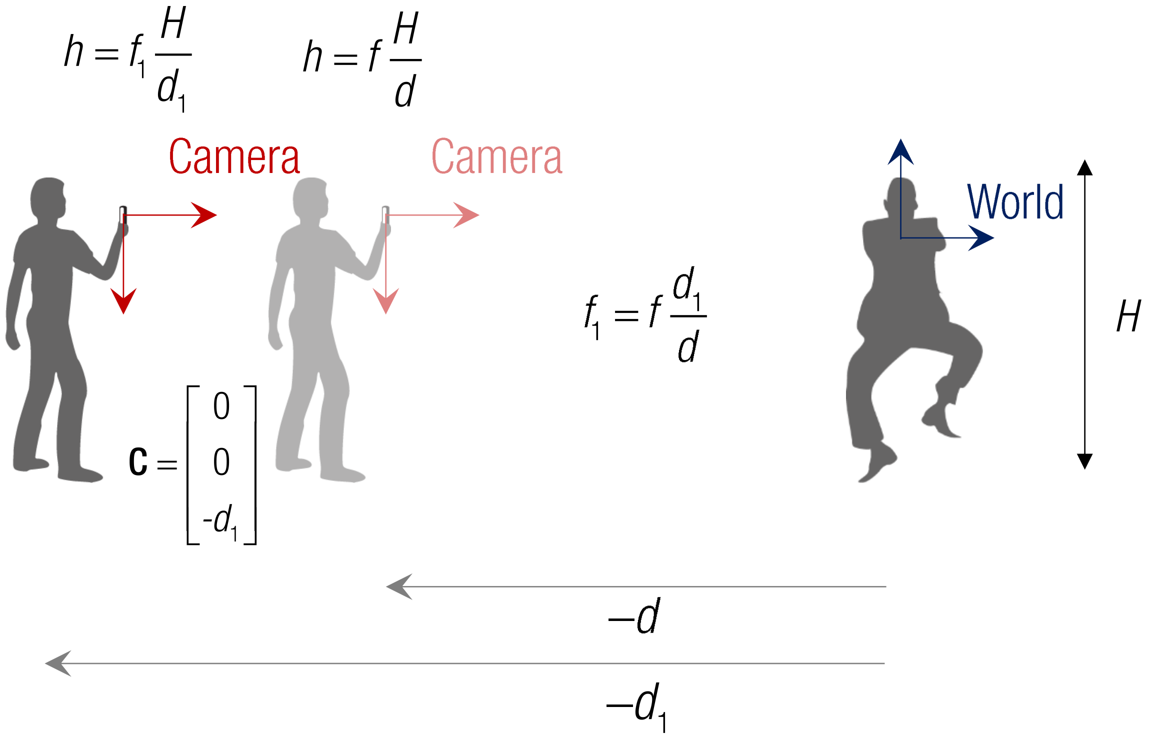
$$f_1 = f\frac{d_1}{d}$$

$$\mathbf{C} = \begin{bmatrix} 0 \\ 0 \\ -d_1 \end{bmatrix}$$

Dolly Zoom Camera Matrix

$$\lambda\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f\frac{d_1}{d} & & p_x\\ & f\frac{d_1}{d} & p_y\\ & & 1\end{bmatrix}\begin{bmatrix}r_{x1} & r_{x2} & 0\\ r_{y1} & r_{y2} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & & & 0\\ & 1 & & 0\\ & & 1 & d_1\end{bmatrix}\begin{bmatrix}\mathbf{X}\\1\end{bmatrix}$$

$$=\begin{bmatrix}f & & p_x\\ & f & p_y\\ & & 1\end{bmatrix}\begin{bmatrix}\frac{d_1}{d} & & \\ & \frac{d_1}{d} & \\ & & 1\end{bmatrix}\begin{bmatrix}r_{x1} & r_{x2} & 0\\ r_{y1} & r_{y2} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & & & 0\\ & 1 & & 0\\ & & 1 & d_1\end{bmatrix}\begin{bmatrix}\mathbf{X}\\1\end{bmatrix}$$



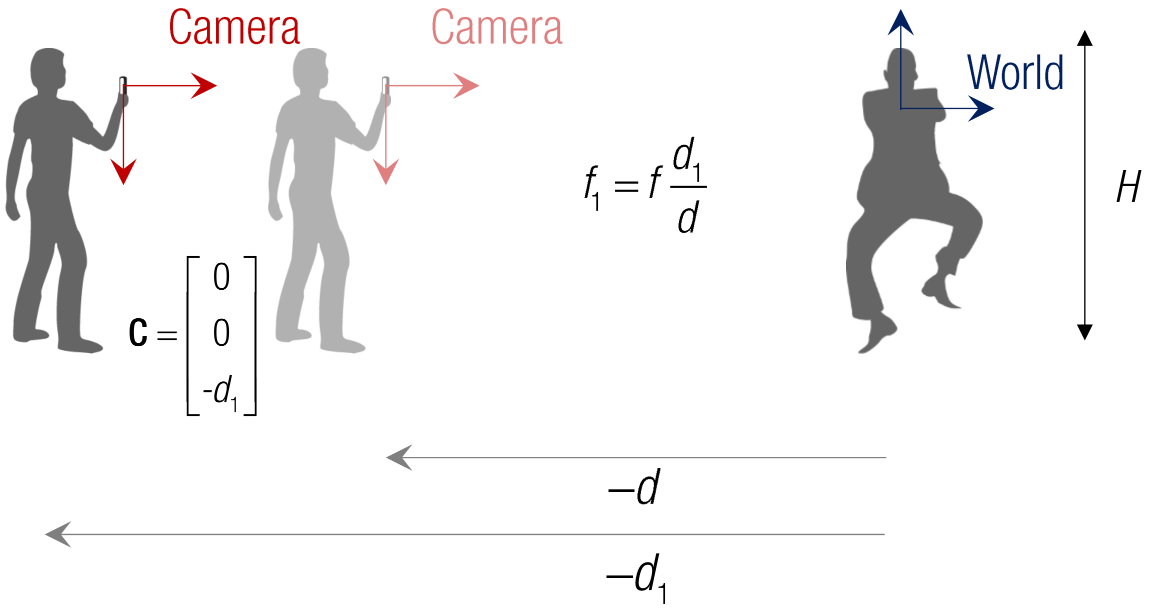
Dolly Zoom Camera Matrix

$$\lambda\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f\frac{d_1}{d} & & p_x\\ & f\frac{d_1}{d} & p_y\\ & & 1\end{bmatrix}\begin{bmatrix}r_{x1} & r_{x2} & 0\\ r_{y1} & r_{y2} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & & & 0\\ & 1 & & 0\\ & & 1 & d_1\end{bmatrix}\begin{bmatrix}\mathbf{X}\\1\end{bmatrix}$$

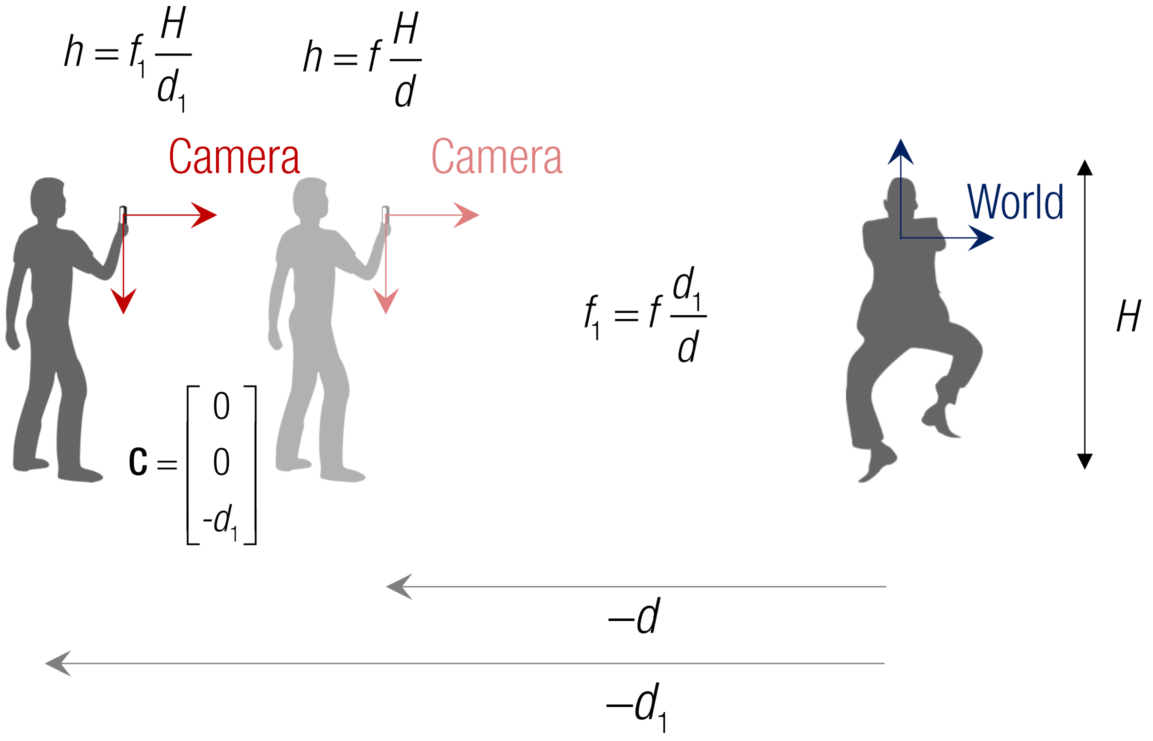
$$=\begin{bmatrix}f & & p_x\\ & f & p_y\\ & & 1\end{bmatrix}\begin{bmatrix}\frac{d_1}{d} & & \\ & \frac{d_1}{d} & \\ & & 1\end{bmatrix}\begin{bmatrix}r_{x1} & r_{x2} & 0\\ r_{y1} & r_{y2} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & & & 0\\ & 1 & & 0\\ & & 1 & d_1\end{bmatrix}\begin{bmatrix}\mathbf{X}\\1\end{bmatrix}$$

$$=\frac{d_1}{d}\begin{bmatrix}f & & p_x\\ & f & p_y\\ & & 1\end{bmatrix}\begin{bmatrix}r_{x1} & r_{x2} & 0\\ r_{y1} & r_{y2} & 0\\ 0 & 0 & \frac{d}{d_1}\end{bmatrix}\begin{bmatrix}1 & & & 0\\ & 1 & & 0\\ & & 1 & d_1\end{bmatrix}\begin{bmatrix}\mathbf{X}\\1\end{bmatrix}$$

$h=f_1\frac{H}{d_1}$ $\qquad$ $h=f\frac{H}{d}$



Dolly Zoom Camera Matrix

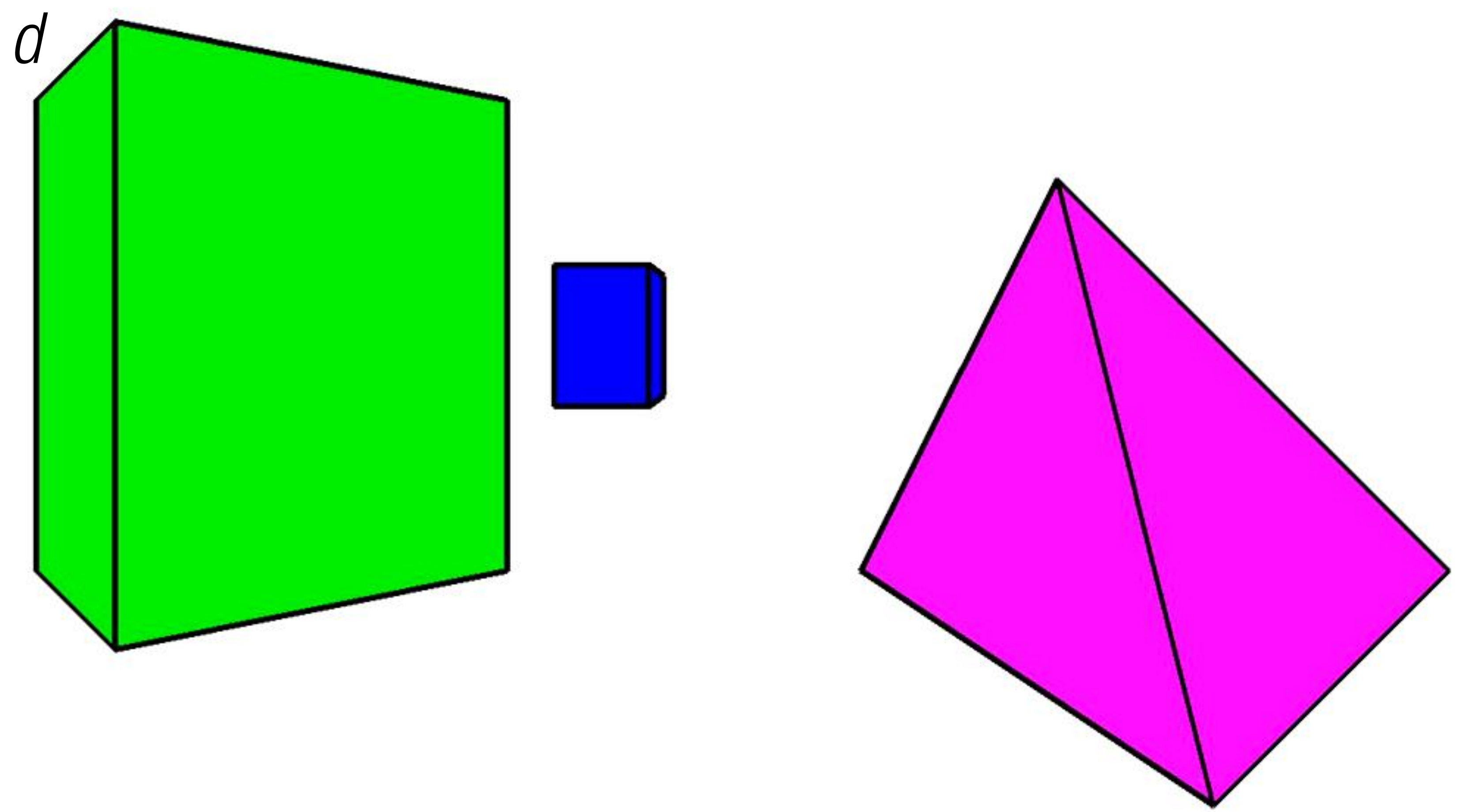


$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f\dfrac{d_1}{d} & & p_x \\ & f\dfrac{d_1}{d} & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d_1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \dfrac{d_1}{d} & & \\ & \dfrac{d_1}{d} & \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d_1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \frac{d_1}{d} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 \\ r_{y1} & r_{y2} & 0 \\ 0 & 0 & \dfrac{d}{d_1} \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & d_1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

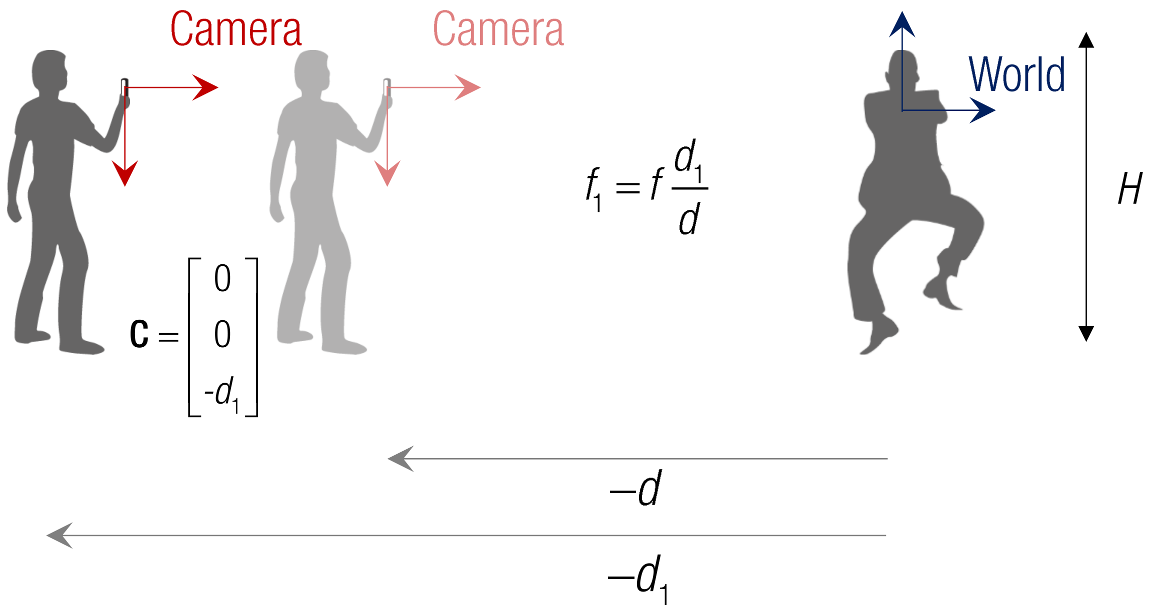
$$= \frac{d_1}{d} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & \dfrac{d}{d_1} & \boxed{d} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$


d

Weak perspectiveness

Strong perspectiveness

Affine Camera



Weak perspectiveness

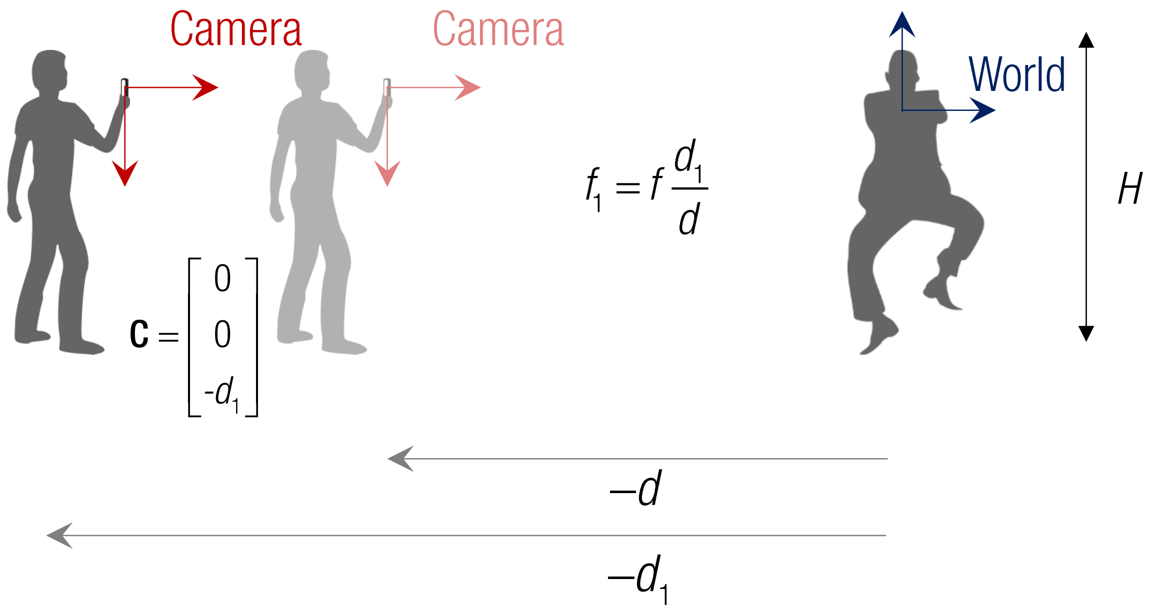
Strong perspectiveness

Dolly zoom camera:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & \frac{d}{d_1} & d \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

What happens if d_1 goes infinity?

Affine Camera



Weak perspectiveness

Strong perspectiveness

Dolly zoom camera:

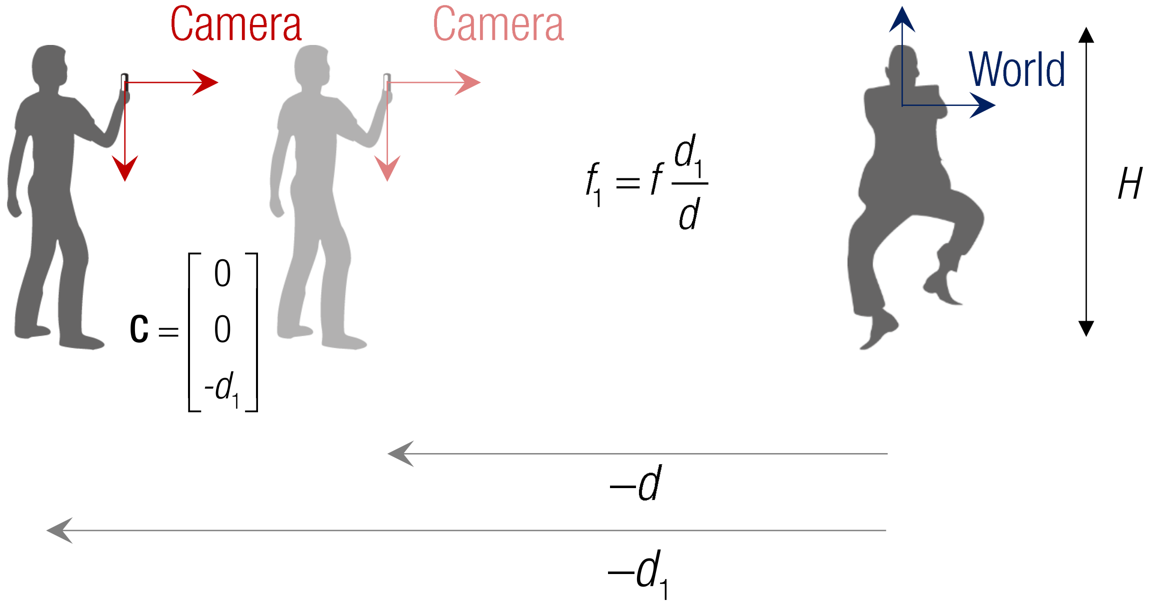
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & \dfrac{d}{d_1} & d \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

What happens if d_1 goes infinity?

$$\lim_{d_1 \to \infty} \mathbf{P} = \lim_{d_1 \to \infty} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & \dfrac{d}{d_1} & d \end{bmatrix}$$

$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & d \end{bmatrix}$$

Affine Camera



Dolly zoom camera:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & \dfrac{d}{d_1} & d \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

What happens if d_1 goes infinity?

$$\lim_{d_1 \to \infty} \mathbf{P} = \lim_{d_1 \to \infty} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & \dfrac{d}{d_1} & d \end{bmatrix}$$

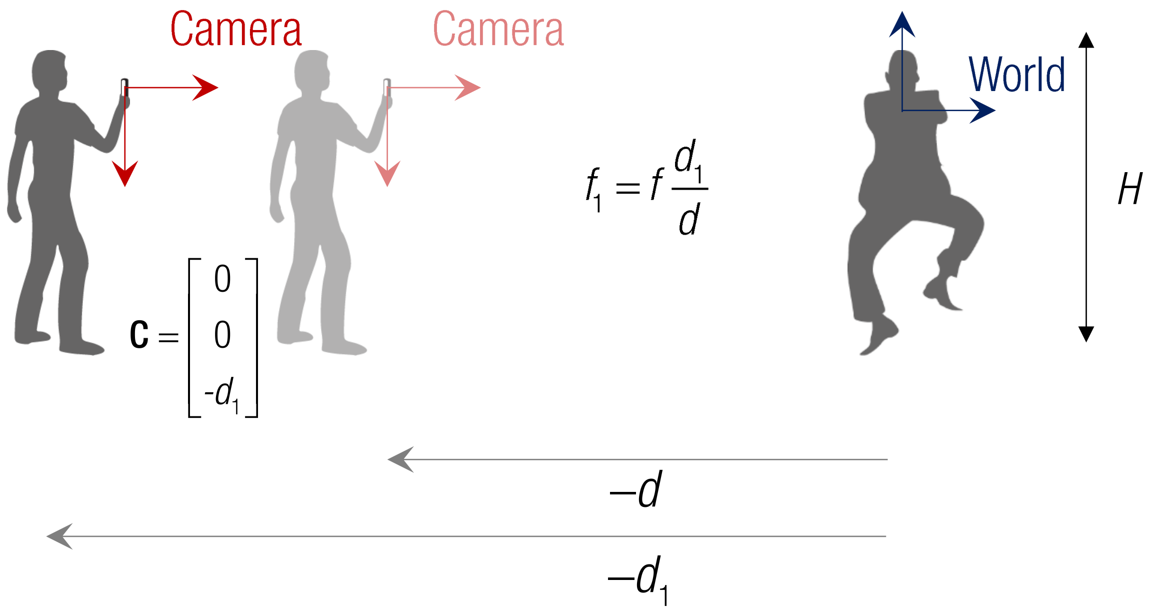
$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ \boxed{0 \quad 0 \quad 0} & & & d \end{bmatrix}$$

: affine camera $\quad d_1 >> 0$

Weak perspectiveness

Strong perspectiveness

Affine Camera



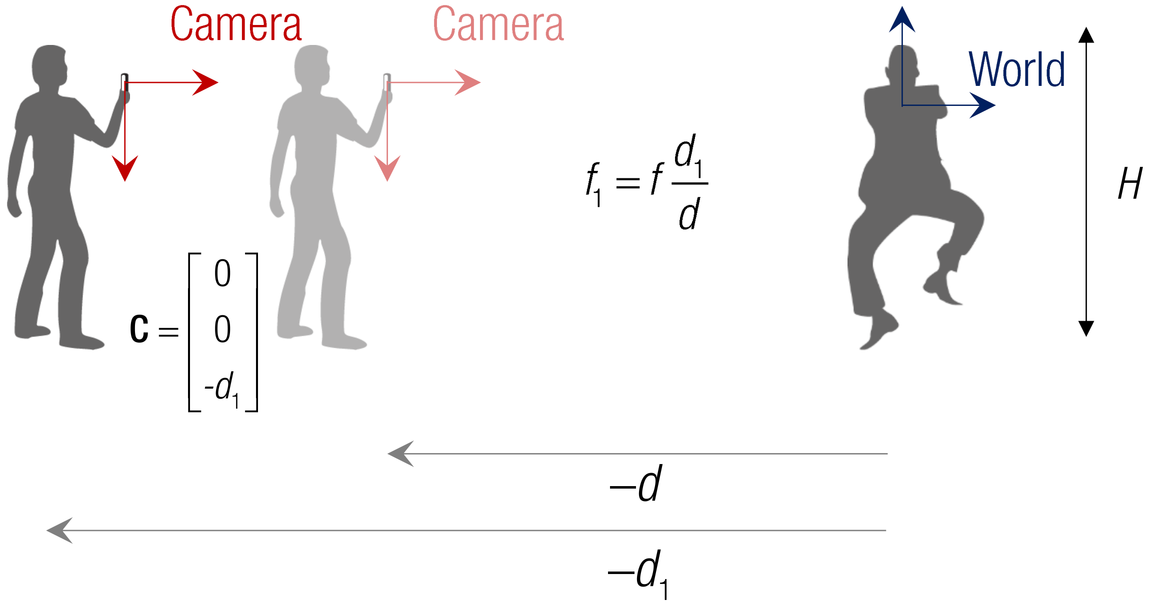
Affine camera:

$$\mathbf{P} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}\begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & d \end{bmatrix} = d\begin{bmatrix} f/d & & p_x \\ & f/d & p_y \\ & & 1 \end{bmatrix}\begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Weak perspectiveness

Strong perspectiveness

Affine Camera



Weak perspectiveness

Strong perspectiveness

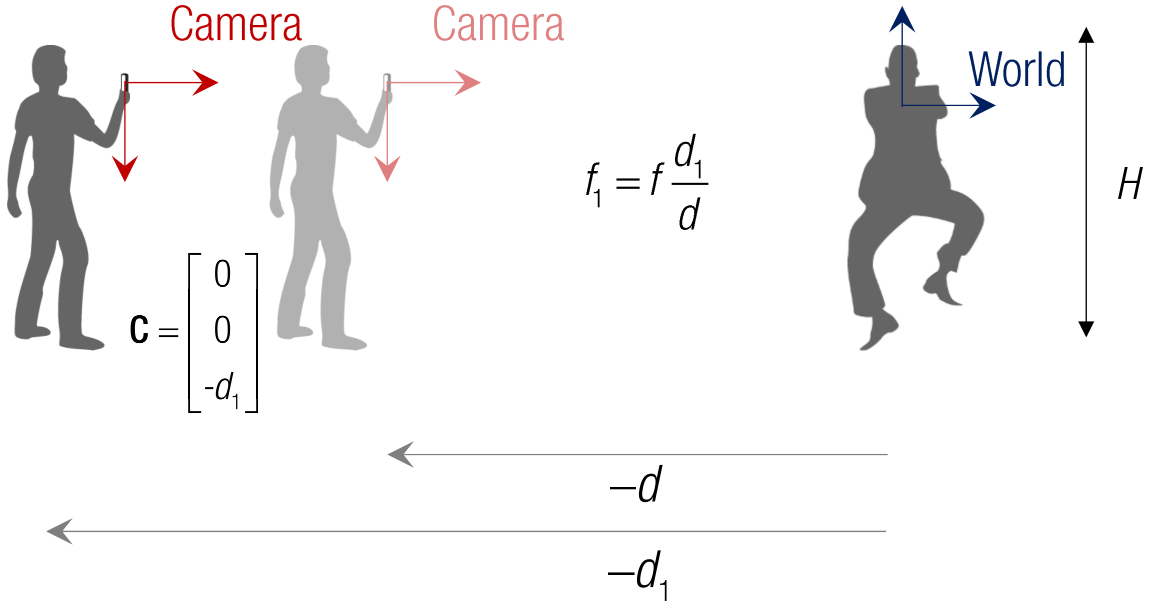
Affine camera:

$$\mathbf{P}_A = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}\begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & d \end{bmatrix} = d\begin{bmatrix} f/d & & p_x \\ & f/d & p_y \\ & & 1 \end{bmatrix}\begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

No scaler

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P}_A\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = d\begin{bmatrix} f/d & & p_x \\ & f/d & p_y \\ & & 1 \end{bmatrix}\begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

Affine Camera



Affine camera:

$$\mathbf{P}_{\mathrm{A}} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & d \end{bmatrix} = d \begin{bmatrix} f/d & & p_x \\ & f/d & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

No scaler

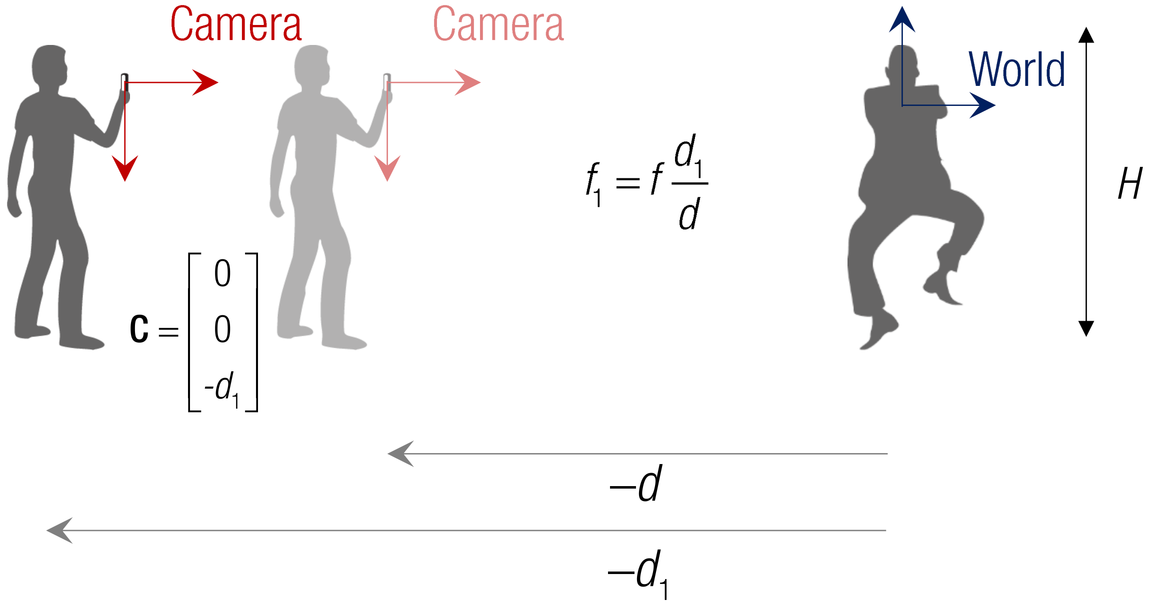
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P}_{\mathrm{A}} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = d \begin{bmatrix} f/d & & p_x \\ & f/d & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

Weak perspectiveness

Strong perspectiveness

Affine Camera



Weak perspectiveness

Strong perspectiveness

Affine camera:

$$\mathbf{P}_{\mathrm{A}} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & d \end{bmatrix} = d \begin{bmatrix} f/d & & p_x \\ & f/d & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

No scaler

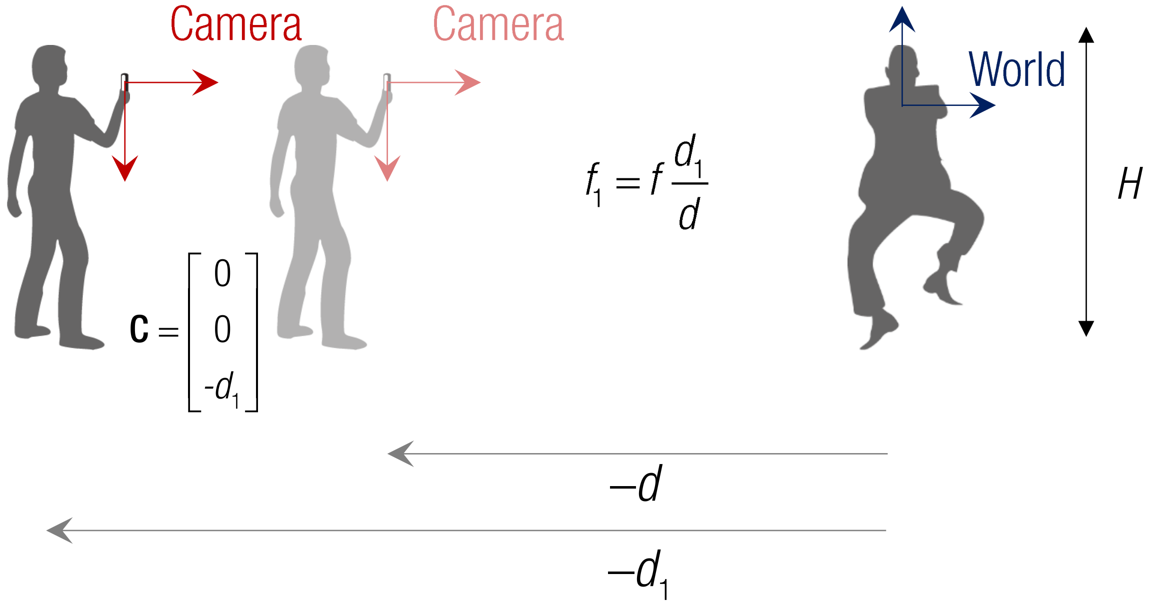
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P}_{\mathrm{A}} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = d \begin{bmatrix} f/d & & p_x \\ & f/d & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{23} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

No denominator

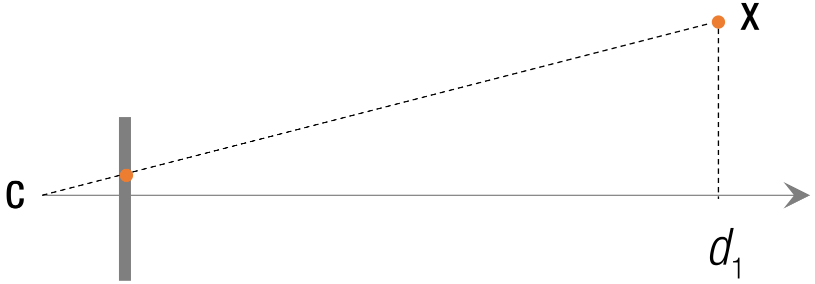
Validity of Affine Approximation



$$f_1 = f\frac{d_1}{d}$$

$$\mathbf{C} = \begin{bmatrix} 0 \\ 0 \\ -d_1 \end{bmatrix}$$

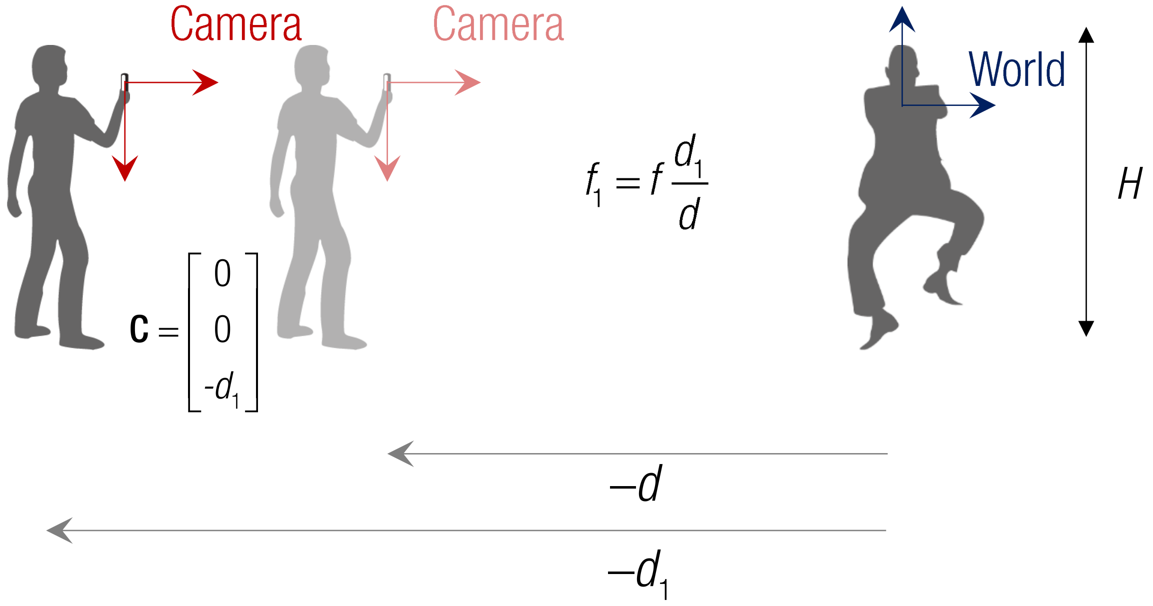
$$\mathbf{u} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$



Weak perspectiveness

Strong perspectiveness

Validity of Affine Approximation



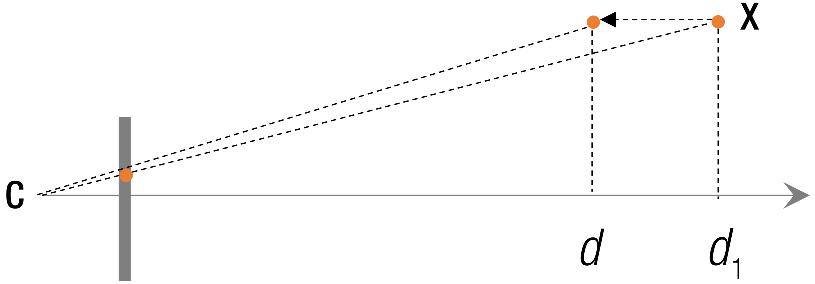
$$\mathbf{C} = \begin{bmatrix} 0 \\ 0 \\ -d_1 \end{bmatrix}$$

$$f_1 = f\frac{d_1}{d}$$

Weak perspectiveness

Strong perspectiveness

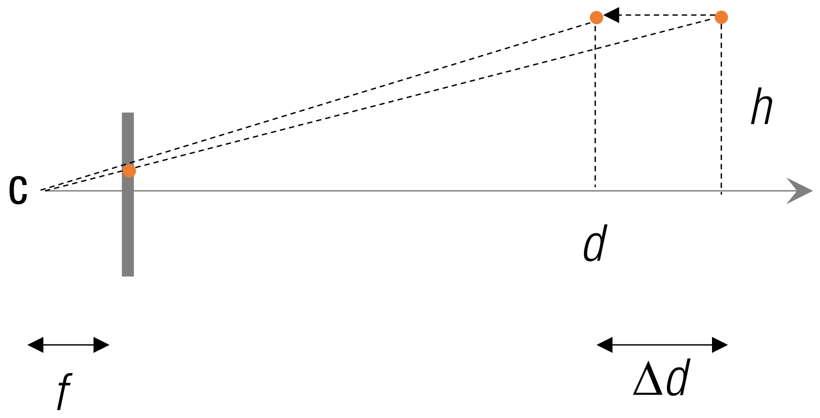
$$\mathbf{u} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$



$$\mathbf{u}_A = \mathbf{P}_A\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

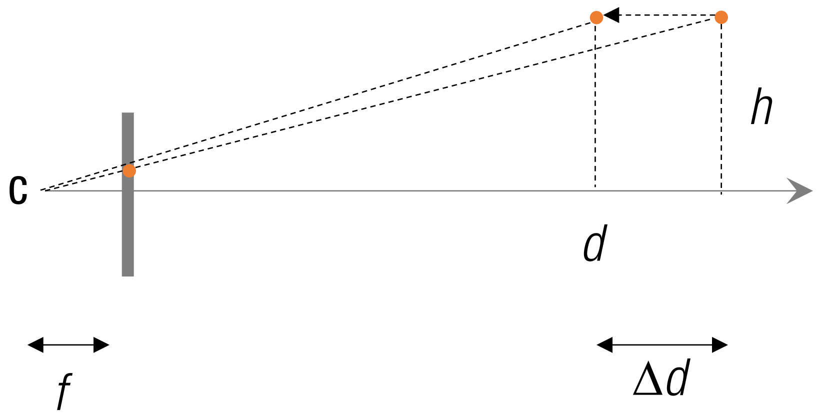
$$\mathbf{P}_A = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}\begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & d \end{bmatrix}$$

Approximation Error



Given image resolution (200x200) with 200 pixel focal length, how much approximatation error at d_1=10m (d=9m, object height = 1) will be?

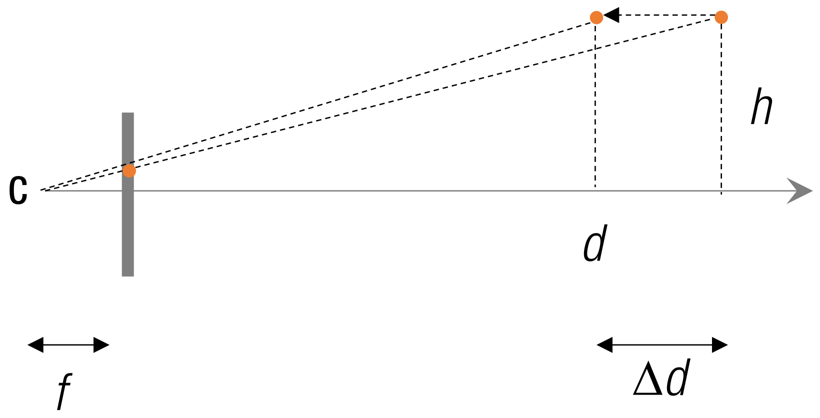
Approximation Error



Given image resolution (200x200) with 200 pixel focal length, how much approximatation error at d_1=10m (d=9m, object height = 1) will be?

$$e = f\frac{h}{d} - f\frac{h}{d+\Delta d}$$

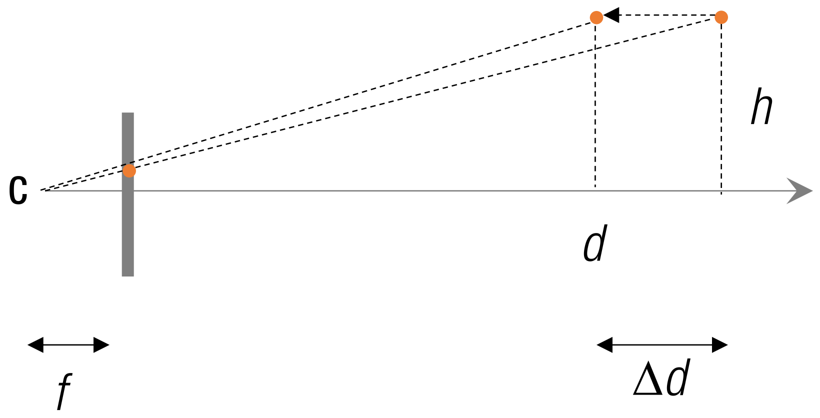
Approximation Error



Given image resolution (200x200) with 200 pixel focal length, how much approximatation error at d_1=10m (d=9m, object height = 1) will be?

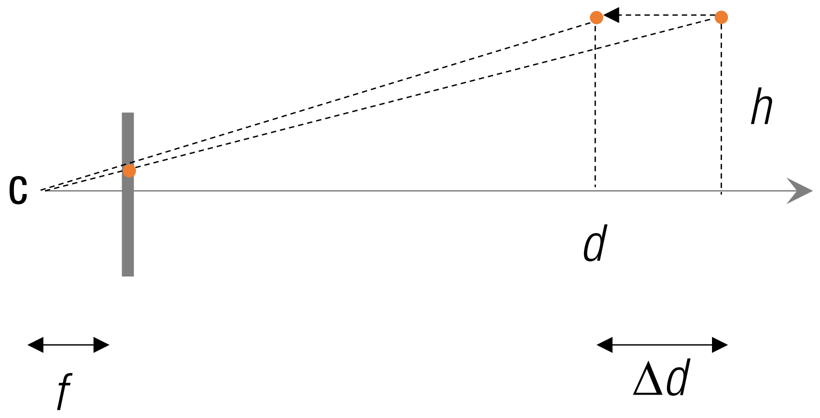
$$e = f\frac{h}{d} - f\frac{h}{d+\Delta d} = 200\frac{1}{9} - 200\frac{1}{10} = 2.22\text{pixel}$$

Approximation Error



Given image resolution (200x200) with 200 pixel focal length, when is the affine camera model valid ($e < 0.5$ pixel)?

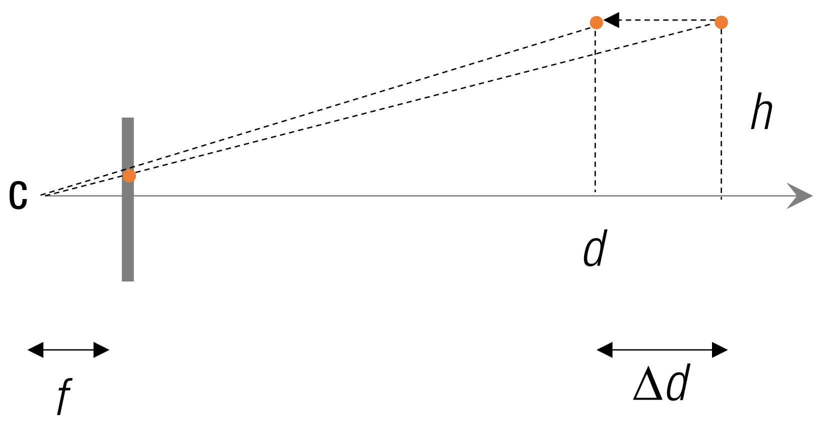
Approximation Error



Given image resolution (200x200) with 200 pixel focal length, when is the affine camera model valid ($e < 0.5$ pixel)?

$$e = f\frac{h}{d} - f\frac{h}{d+\Delta d} < 0.5\,\text{pixel}$$

Approximation Error

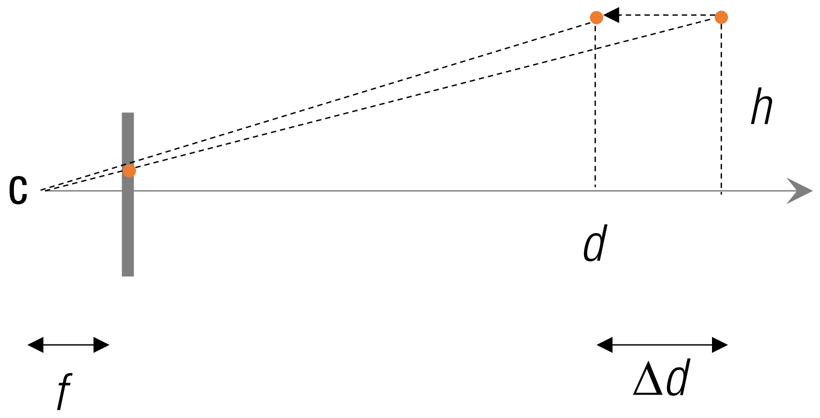


Given image resolution (200x200) with 200 pixel focal length, when is the affine camera model valid ($e < 0.5$ pixel)?

$$e = f\frac{h}{d} - f\frac{h}{d+\Delta d} < 0.5\,\text{pixel}$$

$$\longrightarrow ed^2 + e\Delta d d - fh\Delta d = 0.5d^2 + 0.5d - 200 = 0$$

Approximation Error



Given image resolution (200x200) with 200 pixel focal length, when is the affine camera model valid ($e < 0.5$ pixel)?

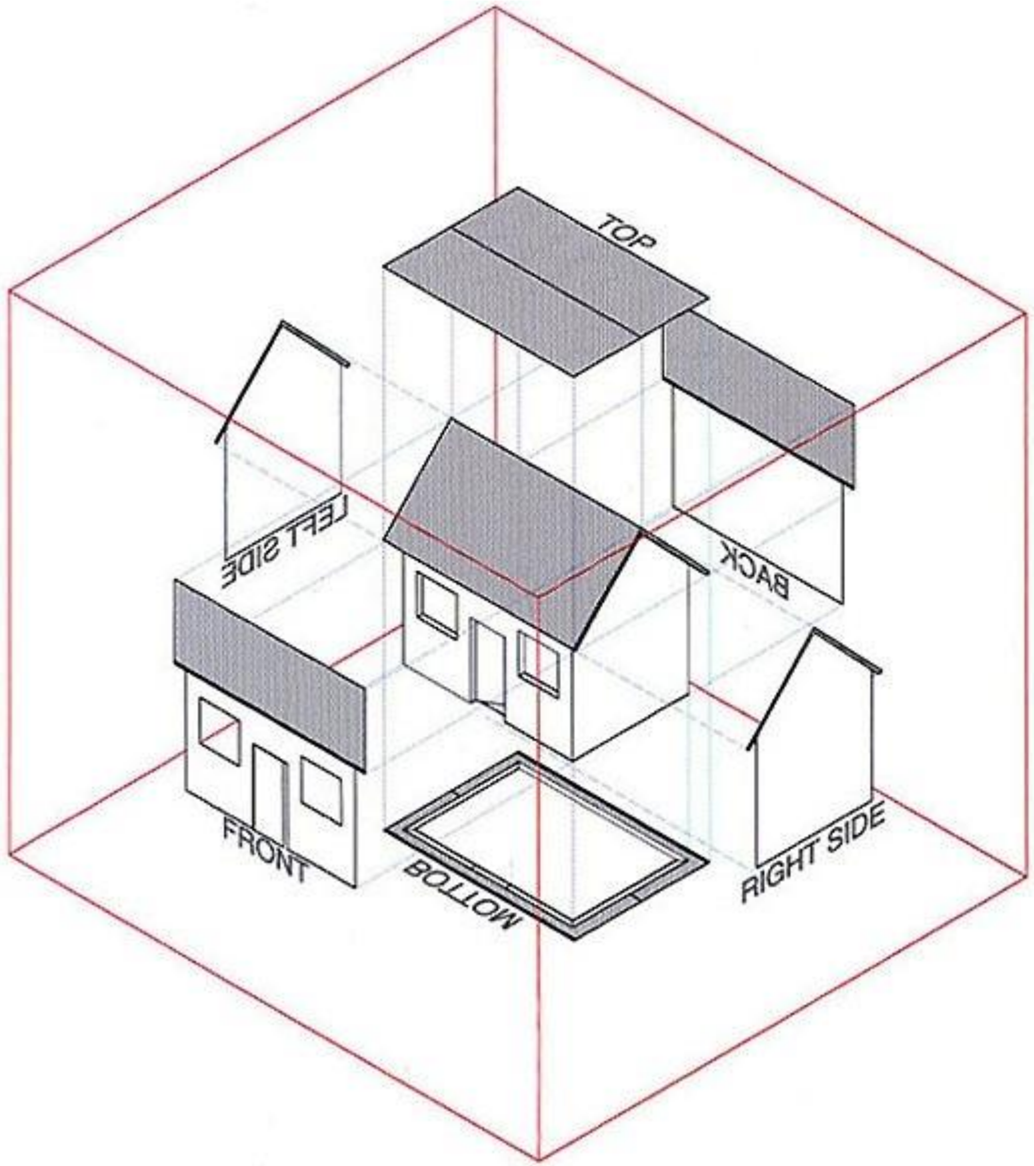
$$e = f\frac{h}{d} - f\frac{h}{d+\Delta d} < 0.5\text{pixel}$$

$$\longrightarrow ed^2 + e\Delta dd - fh\Delta d = 0.5d^2 + 0.5d - 200 = 0$$

$$\longrightarrow d^2 + d - 400 = 0$$

$$d = 19.5\text{m}$$

Orthographic Camera



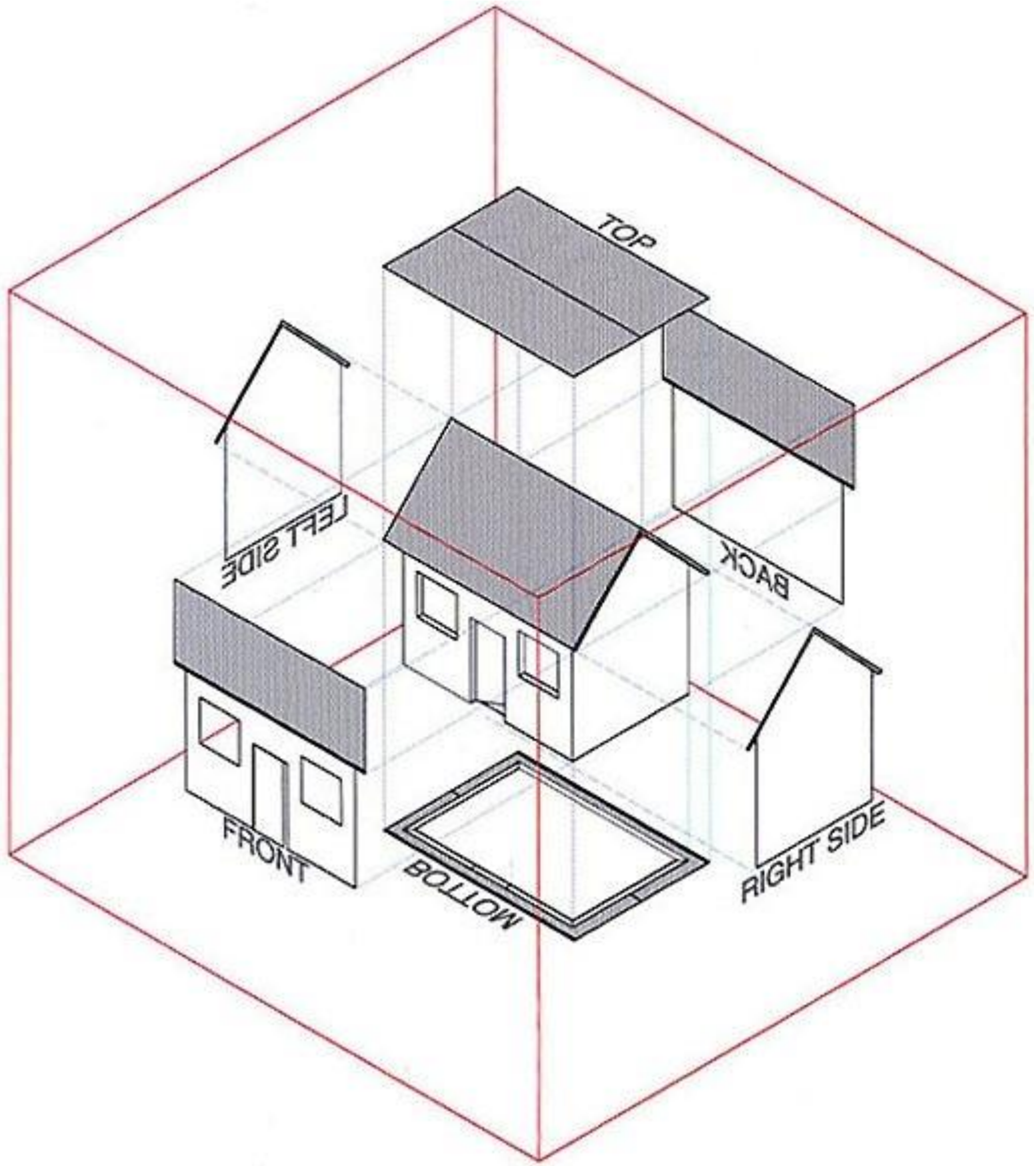
Affine camera:

$$\mathbf{P}_A = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & d \end{bmatrix} = d \begin{bmatrix} f/d & & p_x \\ & f/d & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Orthographic camera:

$$f = 1 \quad p_X = p_Y = 0$$

Orthographic Camera



Affine camera:

$$\mathbf{P}_A = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & d \end{bmatrix} = d \begin{bmatrix} f/d & & p_x \\ & f/d & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{x1} & r_{x2} & 0 & 0 \\ r_{y1} & r_{y2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Orthographic camera:

$$f = 1 \quad p_X = p_Y = 0$$

$$\mathbf{P}_0 = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} & 0 \\ r_{y1} & r_{y2} & r_{y3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Lens configuration (internal parameter)

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = L\left(\mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \right)$$

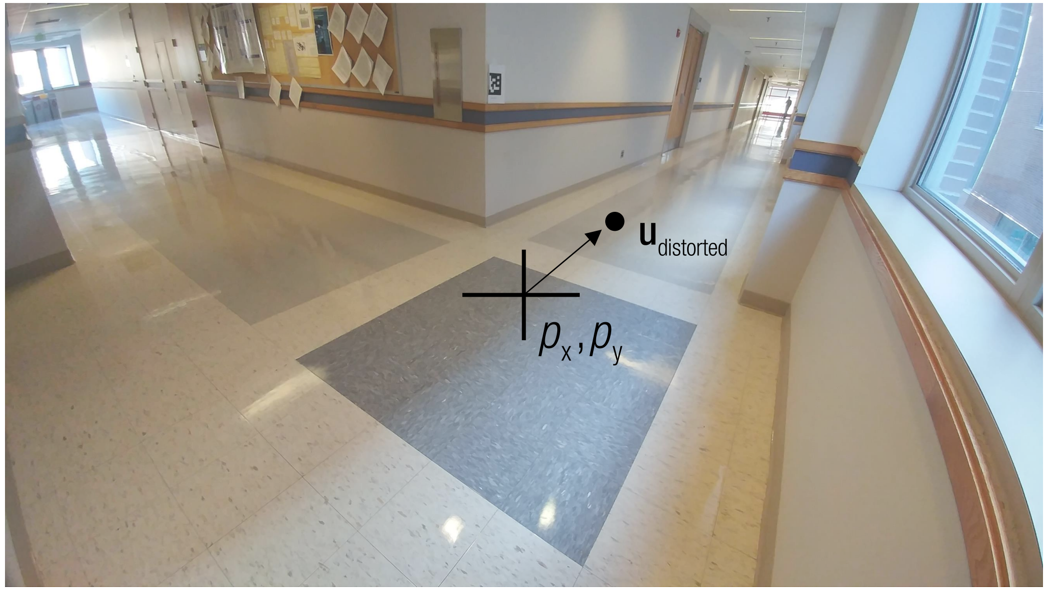
Spatial relationship between sensor and pinhole
(internal parameter)

Camera body configuration
(extrinsic parameter)

Lens Radial Distortion

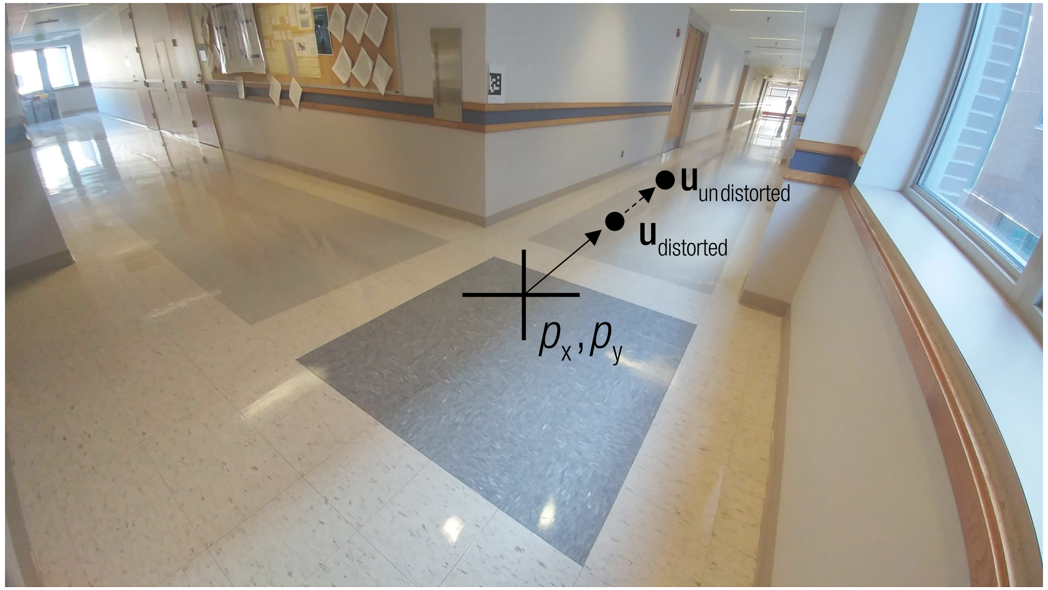
Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.



Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.



$$\bar{\mathbf{u}}_{\text{distorted}} = L(\rho)\bar{\mathbf{u}}_{\text{undistorted}}$$

where $\rho = \left\| \bar{\mathbf{u}}_{\text{distorted}} \right\|$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

Radial Distortion Model

$$\bar{\mathbf{u}}_{\text{distorted}} = L(\rho)\bar{\mathbf{u}}_{\text{undistorted}}$$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

Radial Distortion Model

$$\bar{\mathbf{u}}_{\text{distorted}} = L(\rho)\bar{\mathbf{u}}_{\text{undistorted}}$$

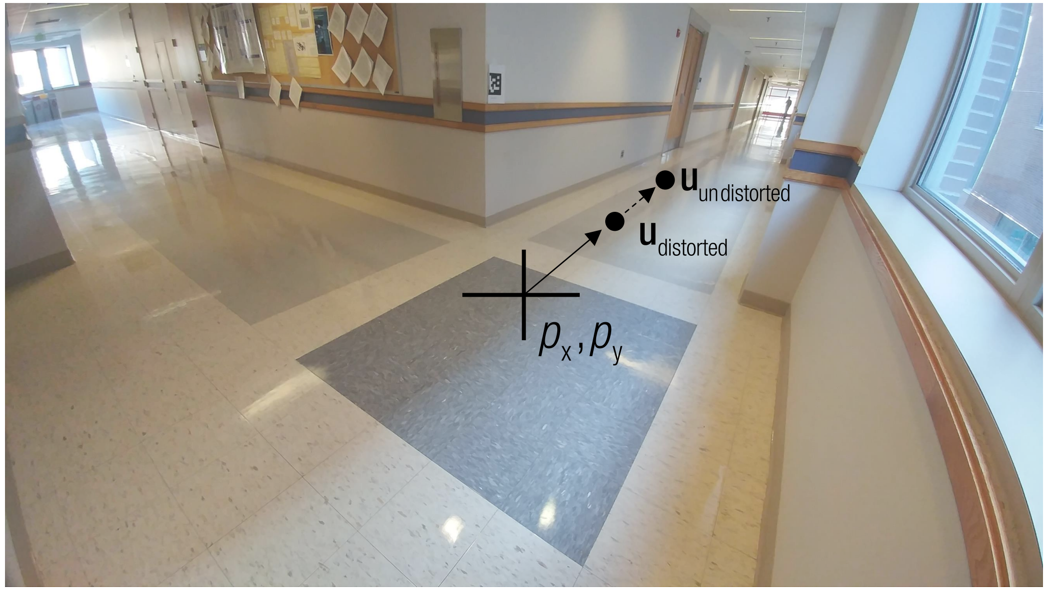
$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

$k_1 < 0$

$k_1 > 0$

Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.



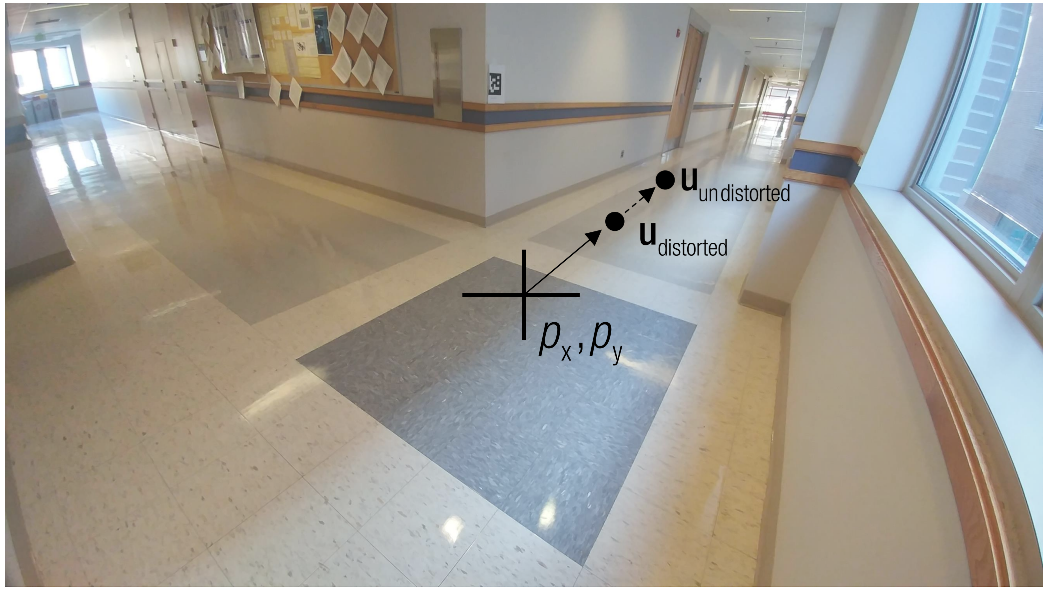
$$\bar{\mathbf{u}}_{\text{distorted}} = L(\rho)\bar{\mathbf{u}}_{\text{undistorted}}$$

where $\rho = \left\| \bar{\mathbf{u}}_{\text{distorted}} \right\|$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.



Normalized point:

$$\bar{\mathbf{u}}_{\text{distorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{distorted}}, \quad \bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

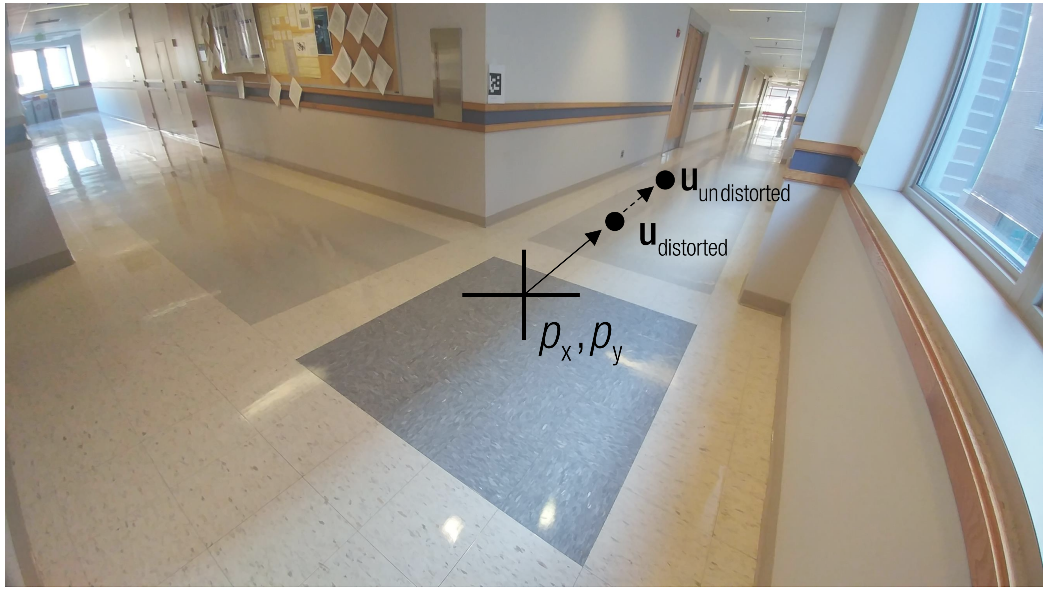
$$\bar{\mathbf{u}}_{\text{distorted}} = L(\rho)\bar{\mathbf{u}}_{\text{undistorted}}$$

where $\rho = \left\| \bar{\mathbf{u}}_{\text{distorted}} \right\|$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.



$$f = f_m \frac{W_{img}}{W_{ccd}} = 0.00153 \frac{3840}{0.0048} = 1224\,\text{pix}$$

$$p_x = \frac{W_{img}}{2} = \frac{3840}{2} = 1920\,\text{pix} \qquad p_y = \frac{H_{img}}{2} = \frac{2160}{2} = 1080\,\text{pix}$$

Normalized point:

$$\bar{\mathbf{u}}_{\text{distorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{distorted}}, \quad \bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

$$\bar{\mathbf{u}}_{\text{distorted}} = L(\rho)\bar{\mathbf{u}}_{\text{undistorted}}$$

where $\rho = \left\|\bar{\mathbf{u}}_{\text{distorted}}\right\|$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.

Distorted image

```
im = imread('image.jpg');
f = 1224;
k = -0.08;
px = size(im,2)/2;
py = size(im,1)/2;
```

$$f = f_{\mathrm{m}} \frac{W_{\mathrm{img}}}{W_{\mathrm{ccd}}} = 0.00153 \frac{3840}{0.0048} = 1224 \mathrm{pix}$$

$$p_{\mathrm{x}} = \frac{W_{\mathrm{img}}}{2} = \frac{3840}{2} = 1920 \mathrm{pix} \qquad p_{\mathrm{y}} = \frac{H_{\mathrm{img}}}{2} = \frac{2160}{2} = 1080 \mathrm{pix}$$

Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.

Distorted image

Undistorted image


```
im = imread('image.jpg');
f = 1224;
k = -0.08;
px = size(im,2)/2;
py = size(im,1)/2;

im_new = zeros(size(im));   % create a new image

for i = 1 : size(im,1)
   for j = 1 : size(im,2)
```

$$f = f_{\mathrm{m}} \frac{W_{\mathrm{img}}}{W_{\mathrm{ccd}}} = 0.00153 \frac{3840}{0.0048} = 1224\,\mathrm{pix}$$

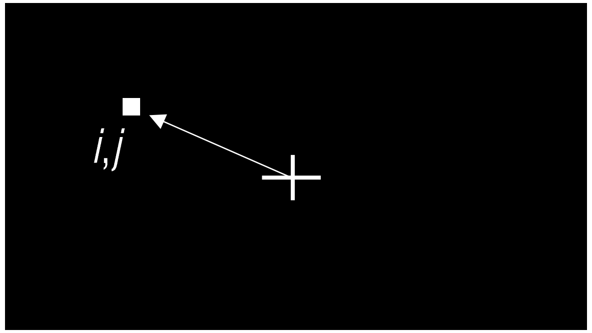
$$p_{\mathrm{x}} = \frac{W_{\mathrm{img}}}{2} = \frac{3840}{2} = 1920\,\mathrm{pix} \qquad p_{\mathrm{y}} = \frac{H_{\mathrm{img}}}{2} = \frac{2160}{2} = 1080\,\mathrm{pix}$$

Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.

Distorted image

Undistorted image




```
im = imread('image.jpg');
f = 1224;
k = -0.08;
px = size(im,2)/2;
py = size(im,1)/2;

im_new = zeros(size(im));   % create a new image

for i = 1 : size(im,1)
   for j = 1 : size(im,2)
      dx = ([j;i]-[px;py])/f;
      r = norm(dx);
```

$$\bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

$$\rho = \left\| \bar{\mathbf{u}}_{\text{distorted}} \right\|$$

$$f = f_{\text{m}} \frac{W_{\text{img}}}{W_{\text{ccd}}} = 0.00153 \frac{3840}{0.0048} = 1224\,\text{pix}$$

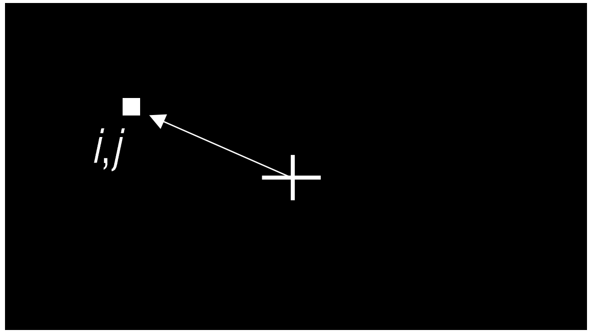
$$p_x = \frac{W_{\text{img}}}{2} = \frac{3840}{2} = 1920\,\text{pix} \qquad p_y = \frac{H_{\text{img}}}{2} = \frac{2160}{2} = 1080\,\text{pix}$$

Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.

Distorted image

Undistorted image




```
im = imread('image.jpg');
f = 1224;
k = -0.08;
px = size(im,2)/2;
py = size(im,1)/2;

im_new = zeros(size(im));   % create a new image

for i = 1 : size(im,1)
   for j = 1 : size(im,2)
      dx = ([j;i]-[px;py])/f;
      r = norm(dx);
      l = 1 + k*r*r;
      x = f*l*dx+[cx;cy];
```

$$\bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

$$\rho = \left\| \bar{\mathbf{u}}_{\text{distorted}} \right\|$$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

$$\mathbf{K}\bar{\mathbf{u}}_{\text{distorted}} = \mathbf{u}_{\text{distorted}}$$

$$f = f_{\text{m}} \frac{W_{\text{img}}}{W_{\text{ccd}}} = 0.00153 \frac{3840}{0.0048} = 1224 \text{pix}$$

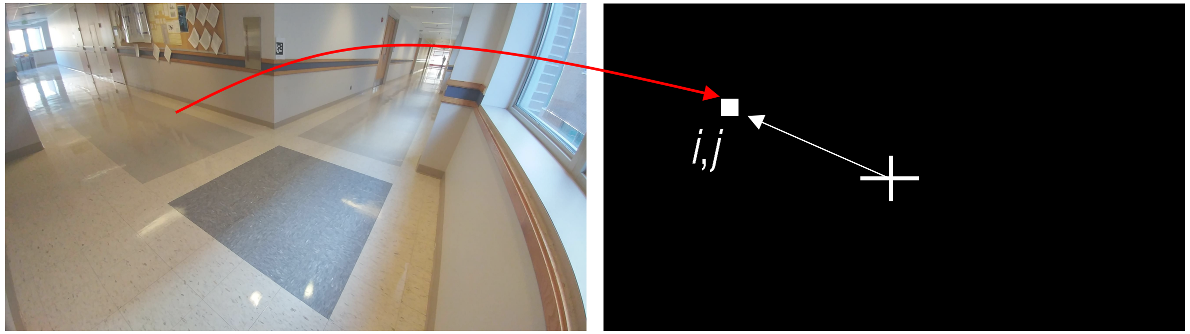
$$p_x = \frac{W_{\text{img}}}{2} = \frac{3840}{2} = 1920 \text{pix} \qquad p_y = \frac{H_{\text{img}}}{2} = \frac{2160}{2} = 1080 \text{pix}$$

Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.

Distorted image

Undistorted image




```
im = imread('image.jpg');
f = 1224;
k = -0.08;
px = size(im,2)/2;
py = size(im,1)/2;

im_new = zeros(size(im));   % create a new image

for i = 1 : size(im,1)
   for j = 1 : size(im,2)
      dx = ([j;i]-[px;py])/f;
      r = norm(dx);
      l = 1 + k*r*r;
      x = f*l*dx+[cx;cy];

      if floor(x(1))<=0 || floor(x(1))>size(im,2) || floor(x(2))<=0 || floor(x(2))>size(im,1)
         continue;
      end

      im_new(i,j,:) = im(floor(x(2)), floor(x(1)),:);
   end
end
```

$$\bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

$$\rho = \left\|\bar{\mathbf{u}}_{\text{distorted}}\right\|$$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

$$\mathbf{K}\bar{\mathbf{u}}_{\text{distorted}} = \mathbf{u}_{\text{distorted}}$$

$$f = f_{\text{m}}\frac{W_{\text{img}}}{W_{\text{ccd}}} = 0.00153\frac{3840}{0.0048} = 1224\text{pix}$$

$$p_x = \frac{W_{\text{img}}}{2} = \frac{3840}{2} = 1920\text{pix} \qquad p_y = \frac{H_{\text{img}}}{2} = \frac{2160}{2} = 1080\text{pix}$$

Radial Distortion Model

Assumption: Lens distortion is a function of distance from the principal point.

Distorted image

Undistorted image


```
im = imread('image.jpg');
f = 1224;
k = -0.08;
px = size(im,2)/2;
py = size(im,1)/2;

im_new = zeros(size(im));   % create a new image

for i = 1 : size(im,1)
   for j = 1 : size(im,2)
      dx = ([j;i]-[px;py])/f;
      r = norm(dx);
      l = 1 + k*r*r;
      x = f*l*dx+[cx;cy];

      if floor(x(1))<=0 || floor(x(1))>size(im,2) || floor(x(2))<=0 || floor(x(2))>size(im,1)
         continue;
      end

      im_new(i,j,:) = im(floor(x(2)), floor(x(1)),:);
   end
end
```

$$\bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

$$\rho = \left\|\bar{\mathbf{u}}_{\text{distorted}}\right\|$$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

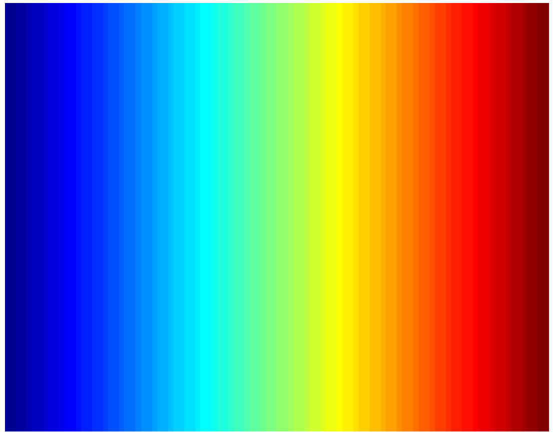
$$\mathbf{K}\bar{\mathbf{u}}_{\text{distorted}} = \mathbf{u}_{\text{distorted}}$$

$$f = f_{\text{m}}\frac{W_{\text{img}}}{W_{\text{ccd}}} = 0.00153\frac{3840}{0.0048} = 1224\text{pix}$$

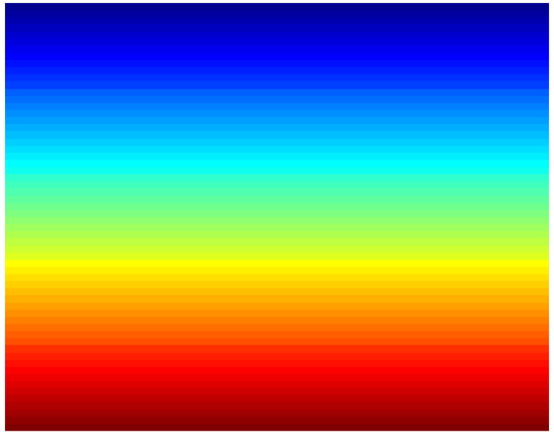
$$p_x = \frac{W_{\text{img}}}{2} = \frac{3840}{2} = 1920\text{pix} \qquad p_y = \frac{H_{\text{img}}}{2} = \frac{2160}{2} = 1080\text{pix}$$

Radial Distortion Model (MATLAB Efficient)

Assumption: Lens distortion is a function of distance from the principal point.



X



Y

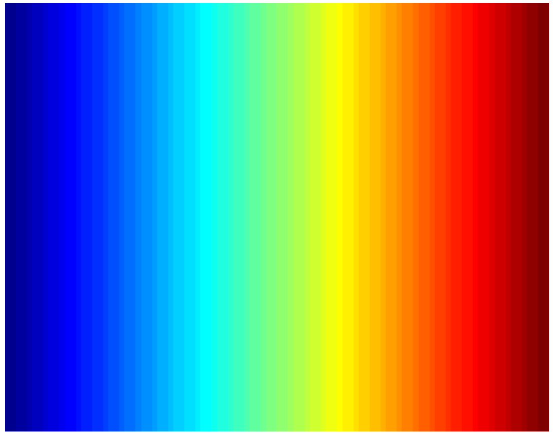
```
[X, Y] = meshgrid(1:(size(im,2)), 1:(size(im,1)));
h = size(X, 1); w = size(X,2);
X = X(:);
Y = Y(:);

pt = [X'; Y'];
;
```

$$f = f_{\mathrm{m}} \frac{W_{\mathrm{img}}}{W_{\mathrm{ccd}}} = 0.00153 \frac{3840}{0.0048} = 1224\,\mathrm{pix}$$

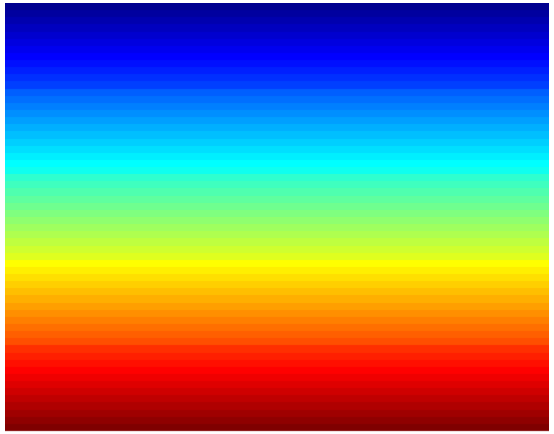
$$p_{\mathrm{x}} = \frac{W_{\mathrm{img}}}{2} = \frac{3840}{2} = 1920\,\mathrm{pix} \qquad p_{\mathrm{y}} = \frac{H_{\mathrm{img}}}{2} = \frac{2160}{2} = 1080\,\mathrm{pix}$$

Radial Distortion Model (MATLAB Efficient)

Assumption: Lens distortion is a function of distance from the principal point.



X



Y

```
[X, Y] = meshgrid(1:(size(im,2)), 1:(size(im,1)));
h = size(X, 1); w = size(X,2);
X = X(:);
Y = Y(:);

pt = [X'; Y'];
pt = bsxfun(@minus, pt, [px;py]);
pt = bsxfun(@rdivide, pt, [f;f]);
```

Elementwise batch operation

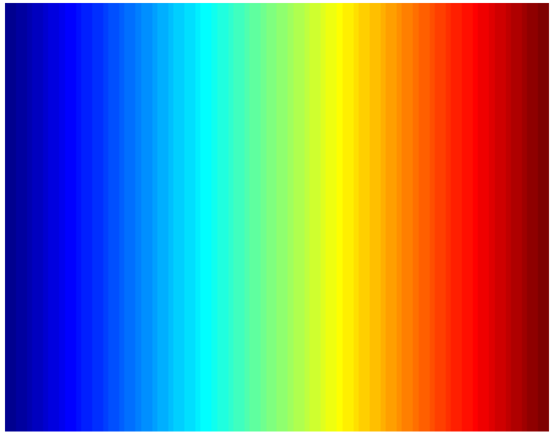
$$\bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

$$f = f_{\text{m}}\frac{W_{\text{img}}}{W_{\text{ccd}}} = 0.00153\frac{3840}{0.0048} = 1224\text{pix}$$

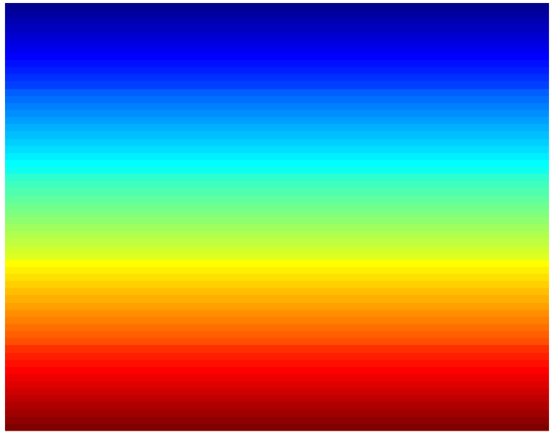
$$p_{\text{x}} = \frac{W_{\text{img}}}{2} = \frac{3840}{2} = 1920\text{pix} \qquad p_{\text{y}} = \frac{H_{\text{img}}}{2} = \frac{2160}{2} = 1080\text{pix}$$

Radial Distortion Model (MATLAB Efficient)

Assumption: Lens distortion is a function of distance from the principal point.



X



Y

```
[X, Y] = meshgrid(1:(size(im,2)), 1:(size(im,1)));
h = size(X, 1); w = size(X,2);
X = X(:);
Y = Y(:);

pt = [X'; Y'];
pt = bsxfun(@minus, pt, [px;py]);
pt = bsxfun(@rdivide, pt, [f;f]);
r_u = sqrt(sum(pt.^2, 1));
pt = bsxfun(@times, pt, 1 + k * r_u.^2);
pt = bsxfun(@times, pt, [f;f]);
pt = bsxfun(@plus, pt, [px;py]);
```

$$\bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

$$\rho = \left\|\bar{\mathbf{u}}_{\text{distorted}}\right\|$$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

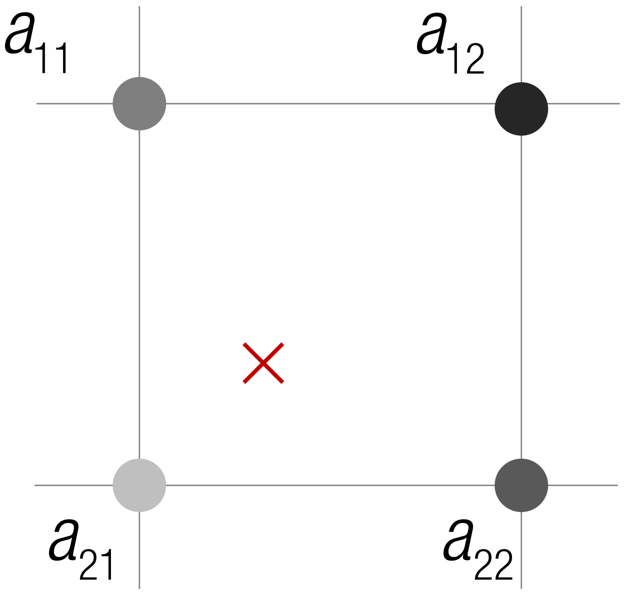
$$\mathbf{K}\bar{\mathbf{u}}_{\text{distorted}} = \mathbf{u}_{\text{distorted}}$$

$$f = f_{\text{m}}\frac{W_{\text{img}}}{W_{\text{ccd}}} = 0.00153\frac{3840}{0.0048} = 1224\text{pix}$$

$$p_{\text{x}} = \frac{W_{\text{img}}}{2} = \frac{3840}{2} = 1920\text{pix} \qquad p_{\text{y}} = \frac{H_{\text{img}}}{2} = \frac{2160}{2} = 1080\text{pix}$$

Radial Distortion Model (MATLAB Efficient)

Assumption: Lens distortion is a function of distance from the principal point.




```
[X, Y] = meshgrid(1:(size(im,2)), 1:(size(im,1)));
h = size(X, 1); w = size(X,2);
X = X(:);
Y = Y(:);

pt = [X'; Y'];
pt = bsxfun(@minus, pt, [px;py]);
pt = bsxfun(@rdivide, pt, [f;f]);
r_u = sqrt(sum(pt.^2, 1));
pt = bsxfun(@times, pt, 1 + k * r_u.^2);
pt = bsxfun(@times, pt, [f;f]);
pt = bsxfun(@plus, pt, [px;py]);

imUndistortion(:,:,1) = reshape(interp2(im(:,:,1), pt(1,:), pt(2,:)), [h, w]);
imUndistortion(:,:,2) = reshape(interp2(im(:,:,2), pt(1,:), pt(2,:)), [h, w]);
imUndistortion(:,:,3) = reshape(interp2(im(:,:,3), pt(1,:), pt(2,:)), [h, w]);
```

Bilinear interpolation

$$\bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

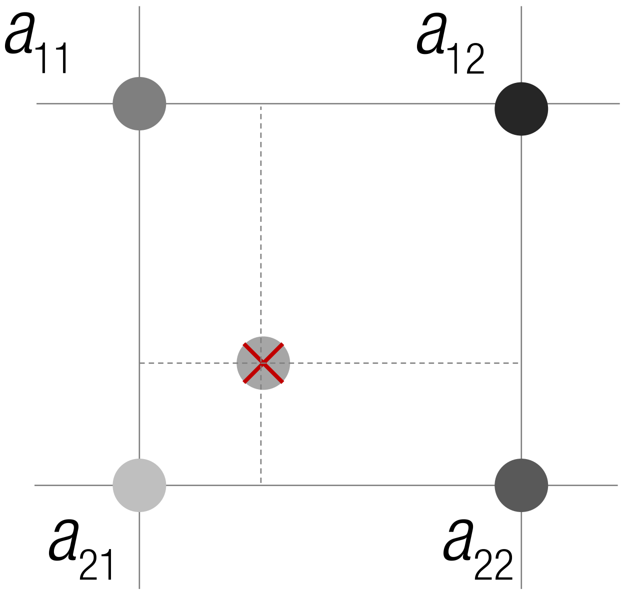
$$\rho = \left\|\bar{\mathbf{u}}_{\text{distorted}}\right\|$$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

$$\mathbf{K}\bar{\mathbf{u}}_{\text{distorted}} = \mathbf{u}_{\text{distorted}}$$

Radial Distortion Model (MATLAB Efficient)

Assumption: Lens distortion is a function of distance from the principal point.




```
[X, Y] = meshgrid(1:(size(im,2)), 1:(size(im,1)));
h = size(X, 1); w = size(X,2);
X = X(:);
Y = Y(:);

pt = [X'; Y'];
pt = bsxfun(@minus, pt, [px;py]);
pt = bsxfun(@rdivide, pt, [f;f]);
r_u = sqrt(sum(pt.^2, 1));
pt = bsxfun(@times, pt, 1 + k * r_u.^2);
pt = bsxfun(@times, pt, [f;f]);
pt = bsxfun(@plus, pt, [px;py]);

imUndistortion(:,:,1) = reshape(interp2(im(:,:,1), pt(1,:), pt(2,:)), [h, w]);
imUndistortion(:,:,2) = reshape(interp2(im(:,:,2), pt(1,:), pt(2,:)), [h, w]);
imUndistortion(:,:,3) = reshape(interp2(im(:,:,3), pt(1,:), pt(2,:)), [h, w]);
```

Bilinear interpolation

$$\bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

$$\rho = \left\|\bar{\mathbf{u}}_{\text{distorted}}\right\|$$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

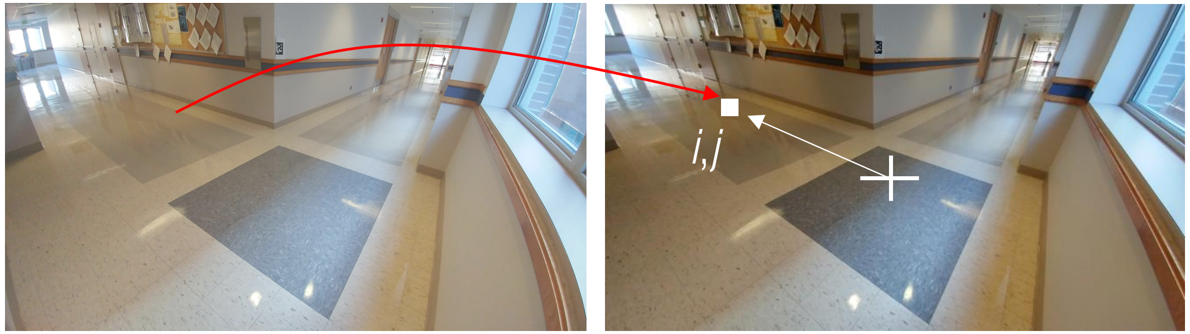
$$\mathbf{K}\bar{\mathbf{u}}_{\text{distorted}} = \mathbf{u}_{\text{distorted}}$$

Radial Distortion Model (MATLAB Efficient)

Assumption: Lens distortion is a function of distance from the principal point.

Distorted image

Undistorted image




```
[X, Y] = meshgrid(1:(size(im,2)), 1:(size(im,1)));
h = size(X, 1); w = size(X,2);
X = X(:);
Y = Y(:);

pt = [X'; Y'];
pt = bsxfun(@minus, pt, [px;py]);
pt = bsxfun(@rdivide, pt, [f;f]);
r_u = sqrt(sum(pt.^2, 1));
pt = bsxfun(@times, pt, 1 + k * r_u.^2);
pt = bsxfun(@times, pt, [f;f]);
pt = bsxfun(@plus, pt, [px;py]);

imUndistortion(:,:,1) = reshape(interp2(im(:,:,1), pt(1,:), pt(2,:)), [h, w]);
imUndistortion(:,:,2) = reshape(interp2(im(:,:,2), pt(1,:), pt(2,:)), [h, w]);
imUndistortion(:,:,3) = reshape(interp2(im(:,:,3), pt(1,:), pt(2,:)), [h, w]);
```

Bilinear interpolation

$$\bar{\mathbf{u}}_{\text{undistorted}} = \mathbf{K}^{-1}\mathbf{u}_{\text{undistorted}}$$

$$\rho = \left\|\bar{\mathbf{u}}_{\text{distorted}}\right\|$$

$$L(\rho) = 1 + k_1\rho^2 + k_2\rho^4 + \cdots$$

$$\mathbf{K}\bar{\mathbf{u}}_{\text{distorted}} = \mathbf{u}_{\text{distorted}}$$


Lens Radial Distortion Correction