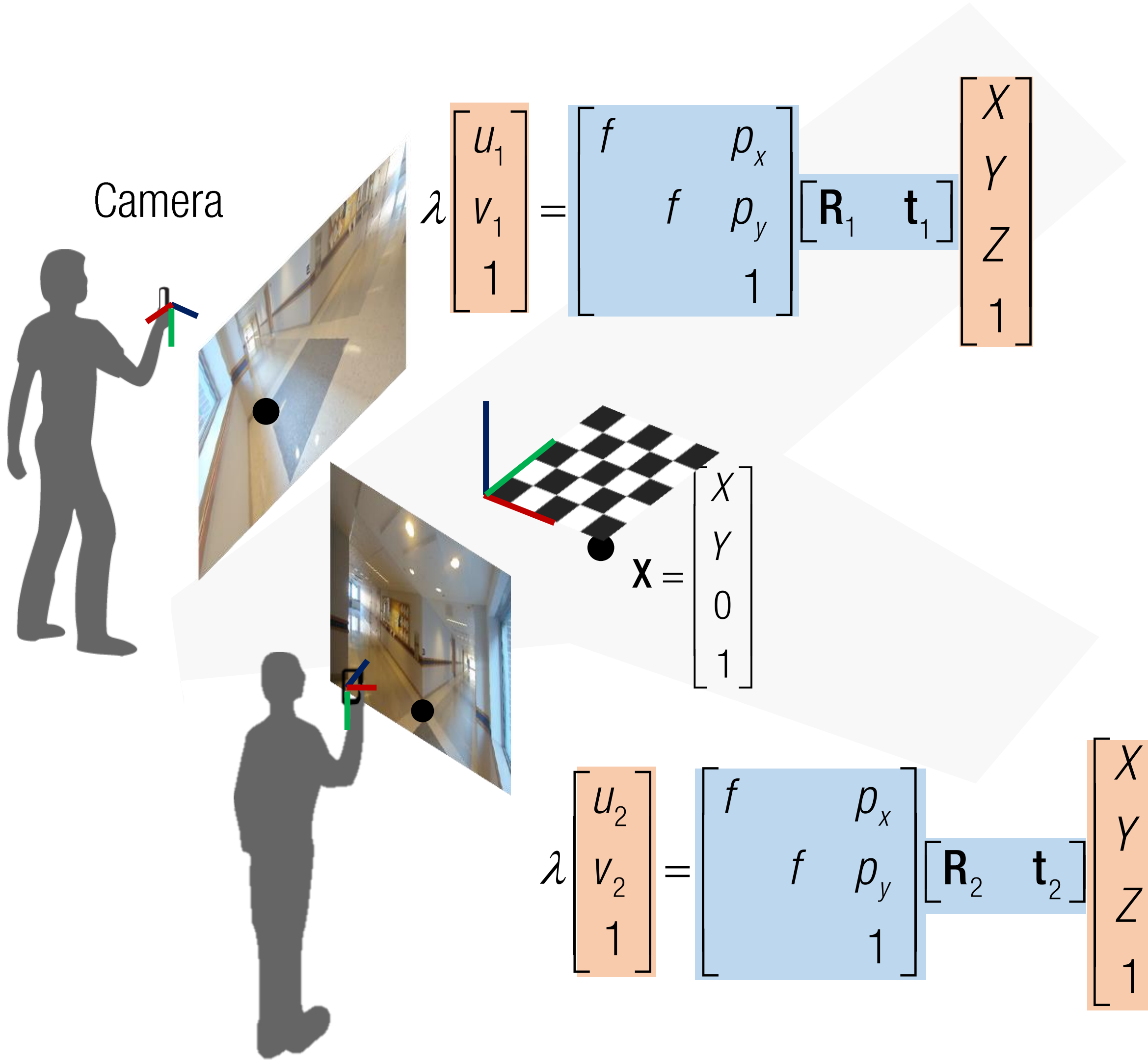
Where am I via Homography?

Announcement

- HW #2 grading will be done by Friday
- Paper selection by next Tuesday (Feb 28)

Recall: Camera Calibration from Multiple Imags



$$\lambda \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 & \mathbf{t}_1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$\lambda \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_2 & \mathbf{t}_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

# of unknowns: 3 (**K**) + 6n (**R** and **t**)

n: the number of images

# of equations: 2nm (**X**)

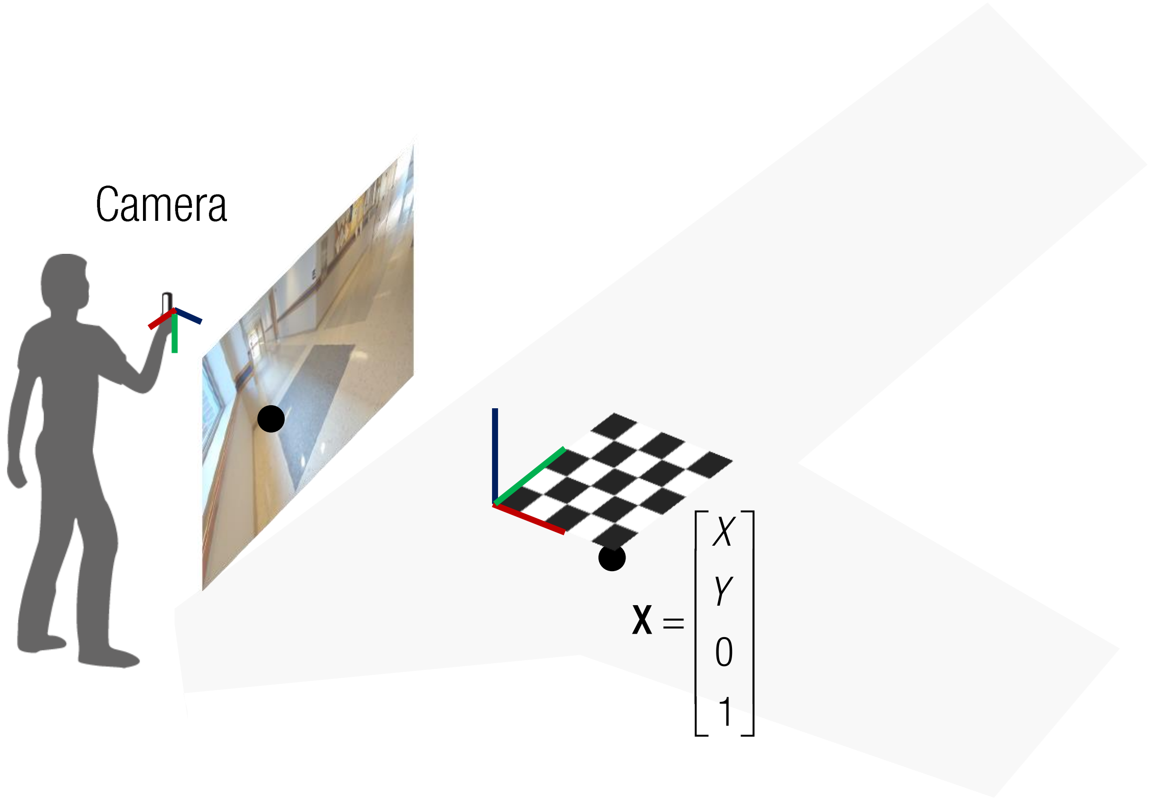
m: the number of known 3D points

We can solve for **K**, **R**, **t** if $3 + 6n < 2\ nm$

: Knowns

: Unknowns

Homography Mapping



$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

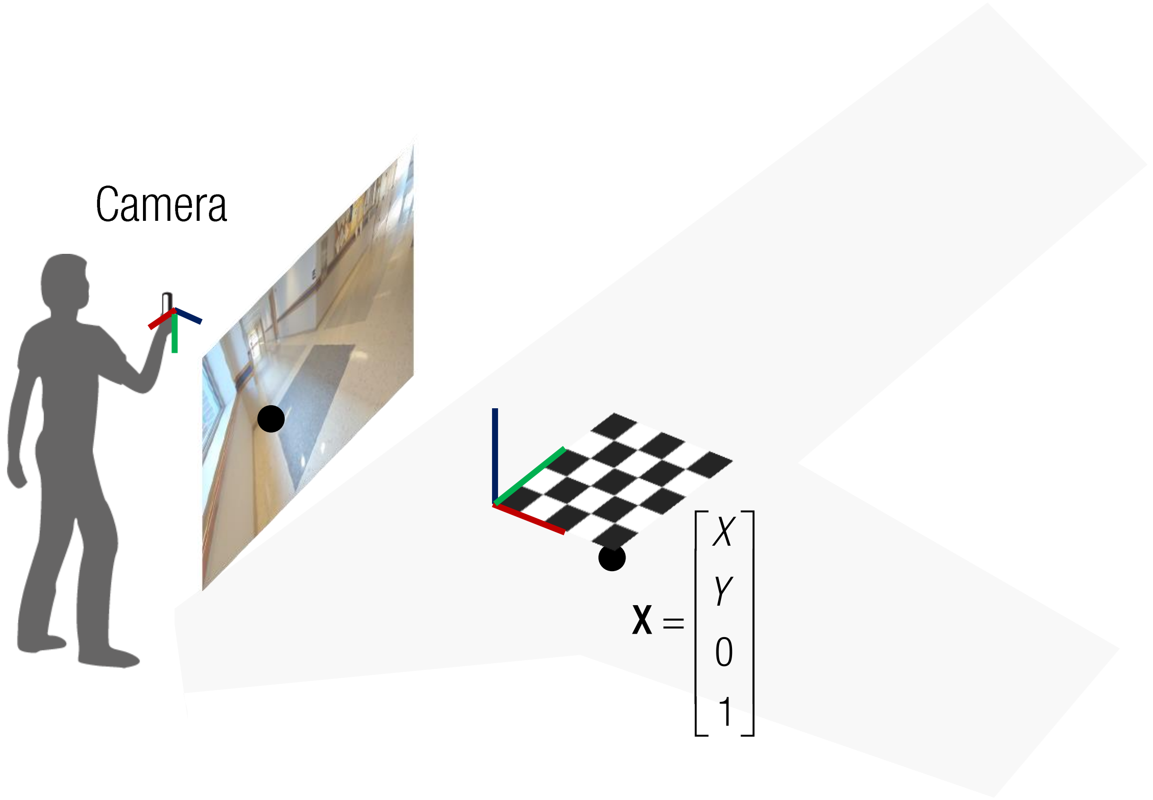
Points in 2D plane are mapped to an image with homography:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

: Knowns (orange: u, v, 1 and X, Y, 0, 1)

: Unknowns (blue: the intrinsic matrix and $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, $\mathbf{t}$)

Homography Mapping



$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

Points in 2D plane are mapped to an image with homography:

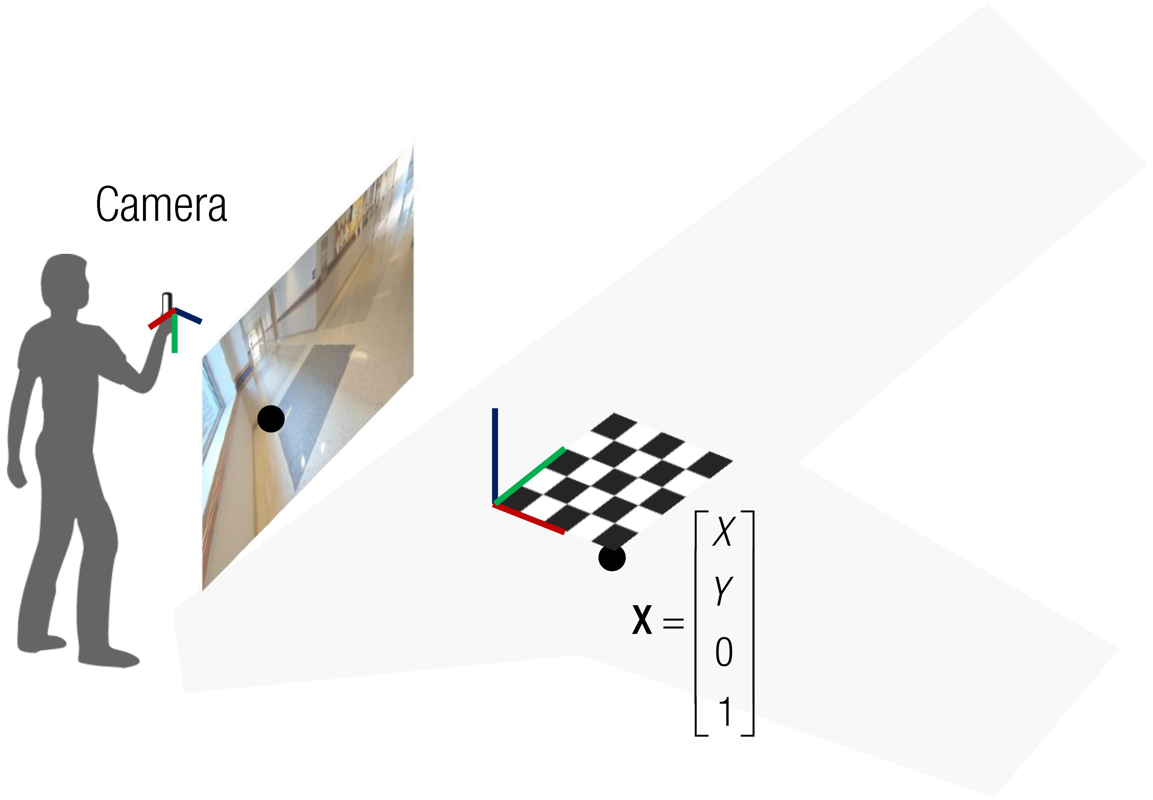
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}}_{\mathbf{Q}} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \underbrace{\mathbf{H}}_{3\text{x}3} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

: Knowns

: Unknowns

Homography Mapping



Points in 2D plane are mapped to an image with homography:

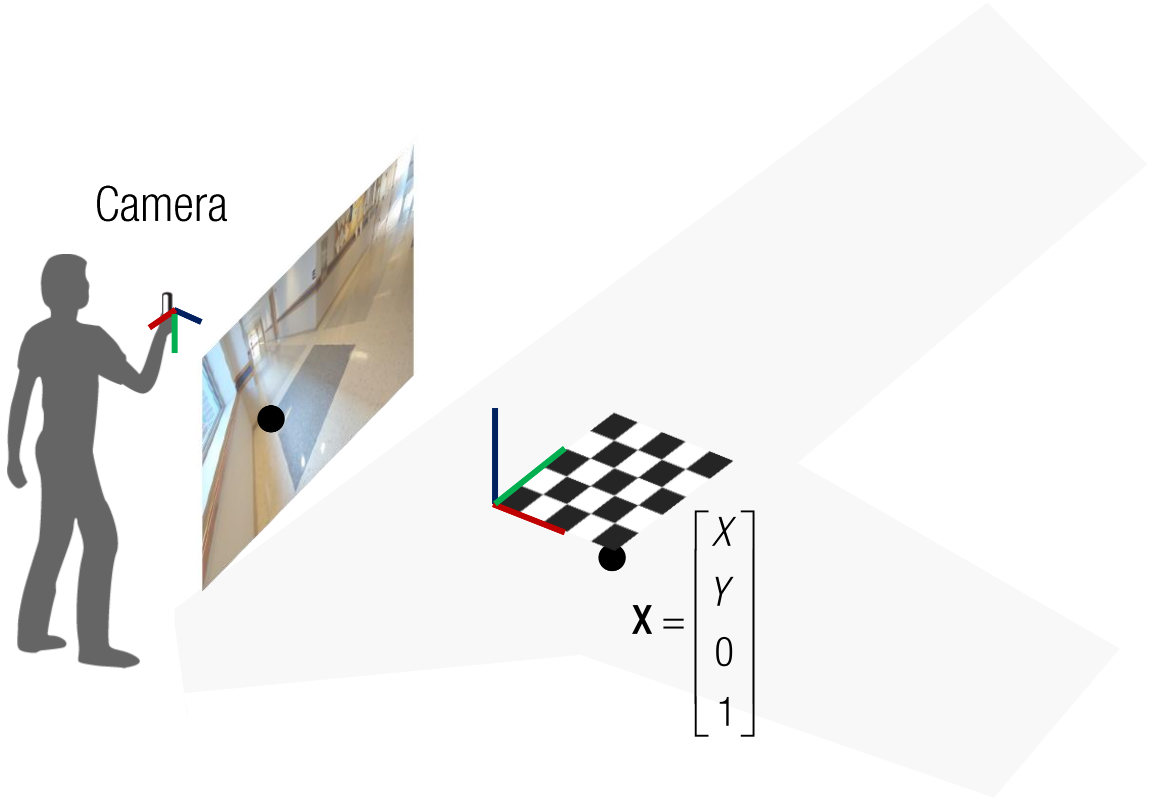
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}}_{\mathbf{Q}} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \underbrace{\mathbf{H}}_{3\text{x}3} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

: Knowns (orange)

: Unknowns (blue)

Homography Mapping



$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

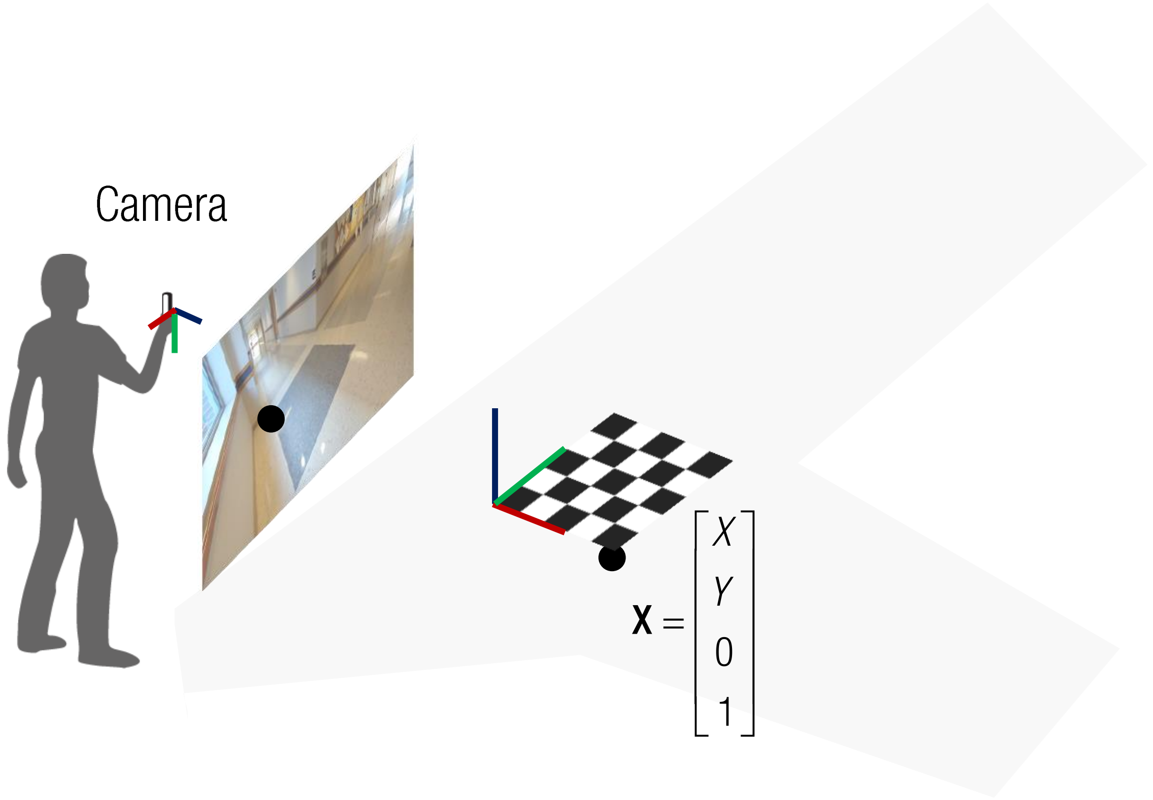
Points in 2D plane are mapped to an image with homography:

$$\mathbf{K}^{-1}\mathbf{H} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$$

: Knowns

: Unknowns

Homography Mapping



$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

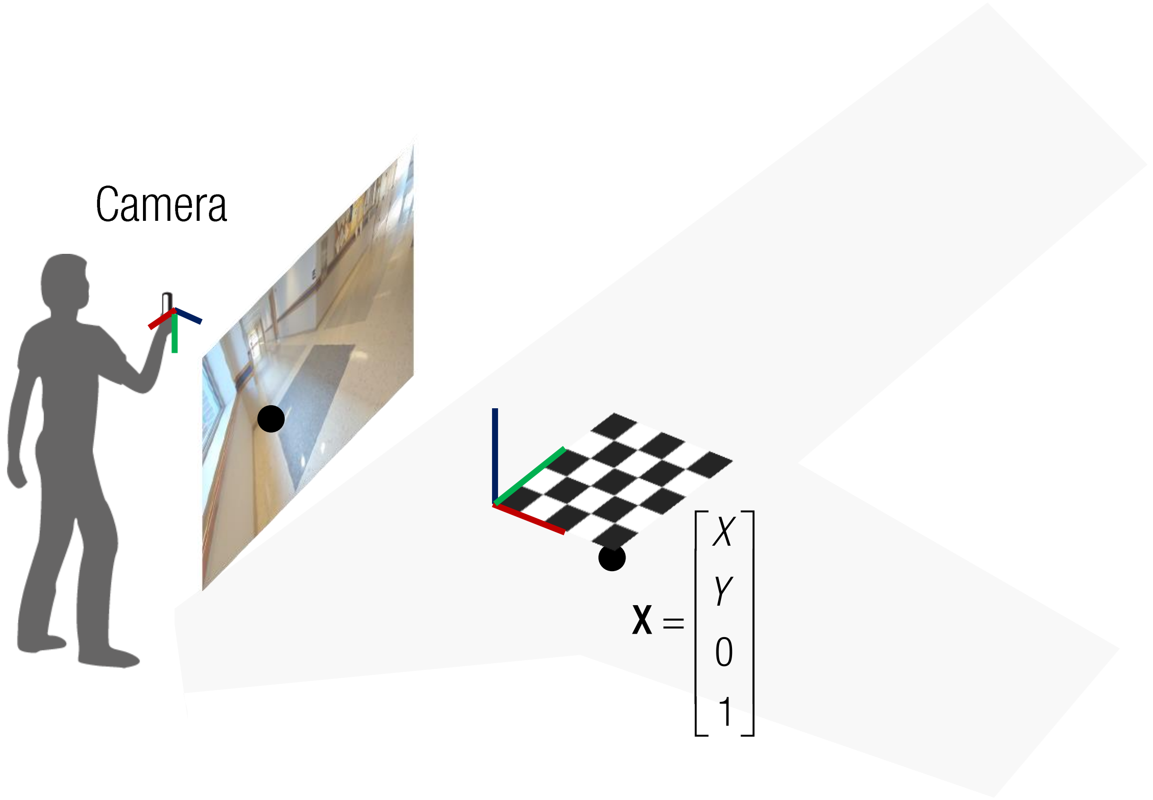
Points in 2D plane are mapped to an image with homography:

$$\mathbf{K}^{-1}\mathbf{H} = \mathbf{K}^{-1}\begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$$

$$\rightarrow \mathbf{r}_1 = \frac{\mathbf{K}^{-1}\mathbf{h}_1}{\left\|\mathbf{K}^{-1}\mathbf{h}_1\right\|}, \quad \mathbf{r}_2 = \frac{\mathbf{K}^{-1}\mathbf{h}_2}{\left\|\mathbf{K}^{-1}\mathbf{h}_1\right\|}, \quad \mathbf{t} = \frac{\mathbf{K}^{-1}\mathbf{h}_3}{\left\|\mathbf{K}^{-1}\mathbf{h}_1\right\|}$$

Common denominator

Homography Mapping



$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

Points in 2D plane are mapped to an image with homography:

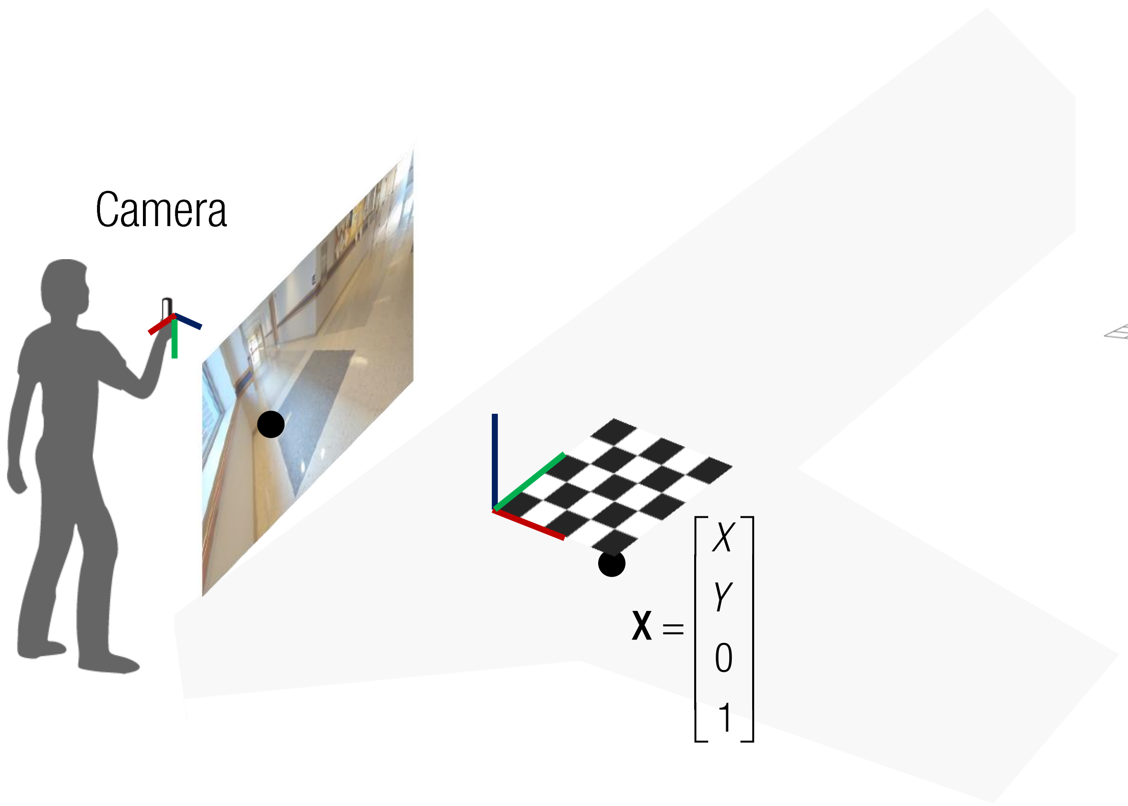
$$\mathbf{K}^{-1}\mathbf{H} = \mathbf{K}^{-1}\begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$$

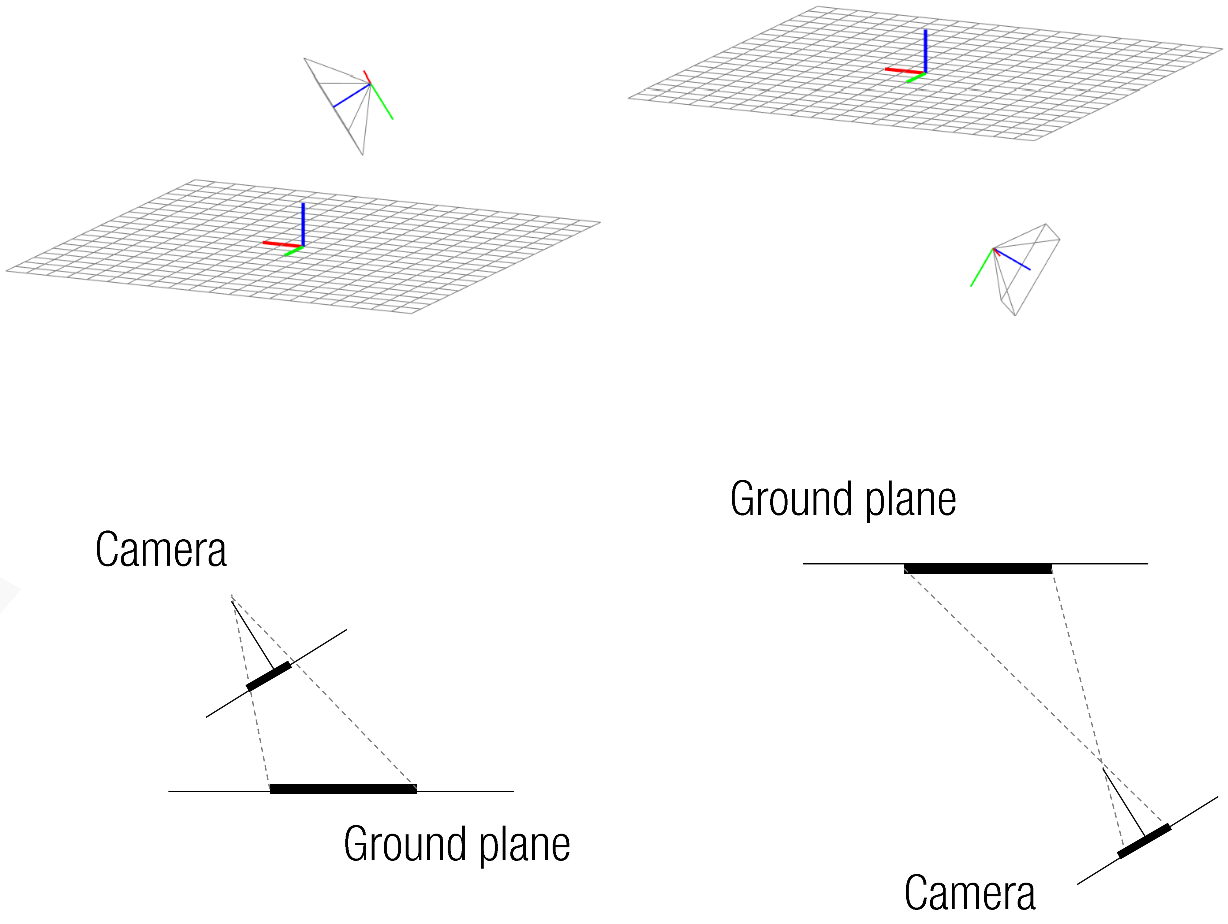
$$\rightarrow \mathbf{r}_1 = \frac{\mathbf{K}^{-1}\mathbf{h}_1}{\left\|\mathbf{K}^{-1}\mathbf{h}_1\right\|}, \quad \mathbf{r}_2 = \frac{\mathbf{K}^{-1}\mathbf{h}_2}{\left\|\mathbf{K}^{-1}\mathbf{h}_1\right\|}, \quad \mathbf{t} = \frac{\mathbf{K}^{-1}\mathbf{h}_3}{\left\|\mathbf{K}^{-1}\mathbf{h}_1\right\|}$$

Common denominator

$$\rightarrow \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

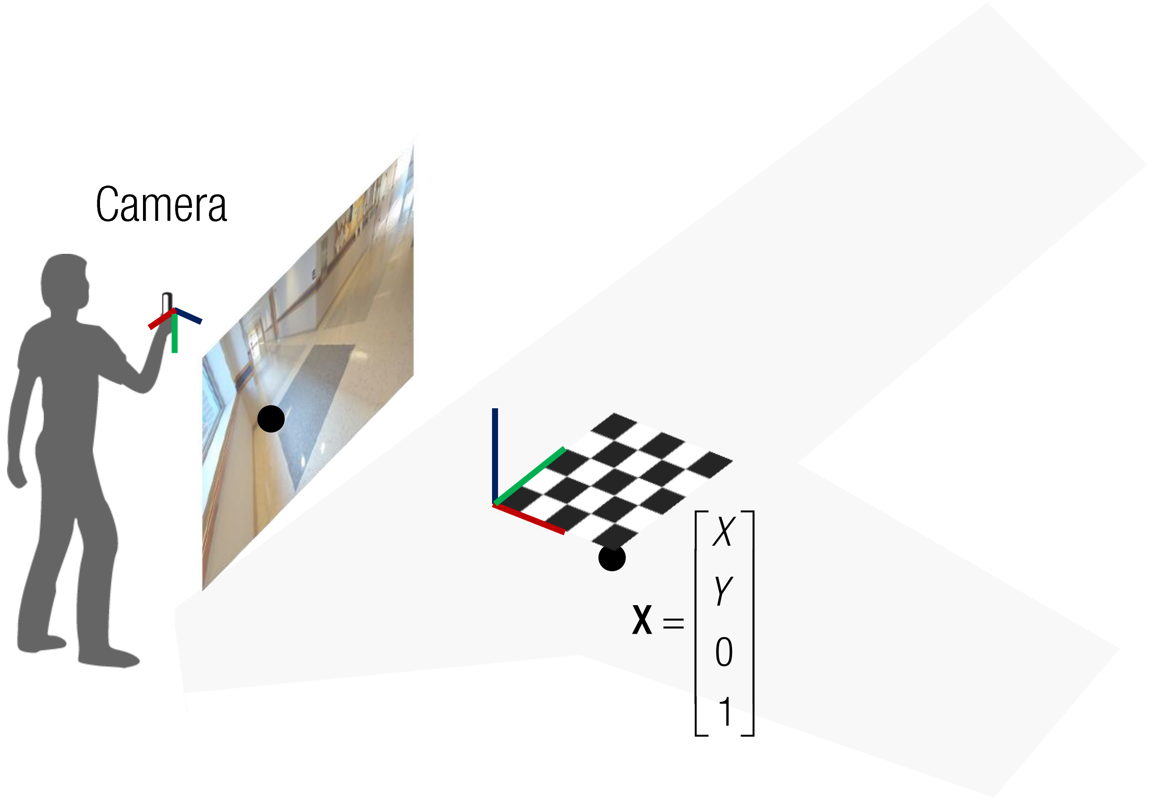
Geometric Ambiguity





Two configurations

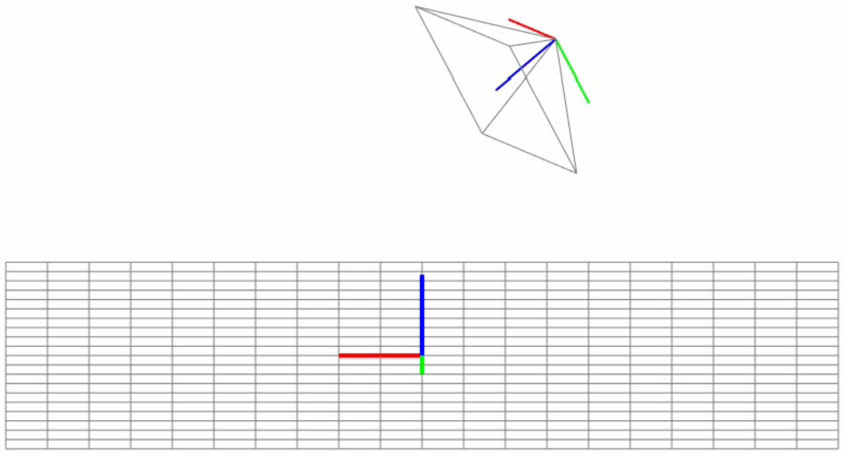
Homography Mapping



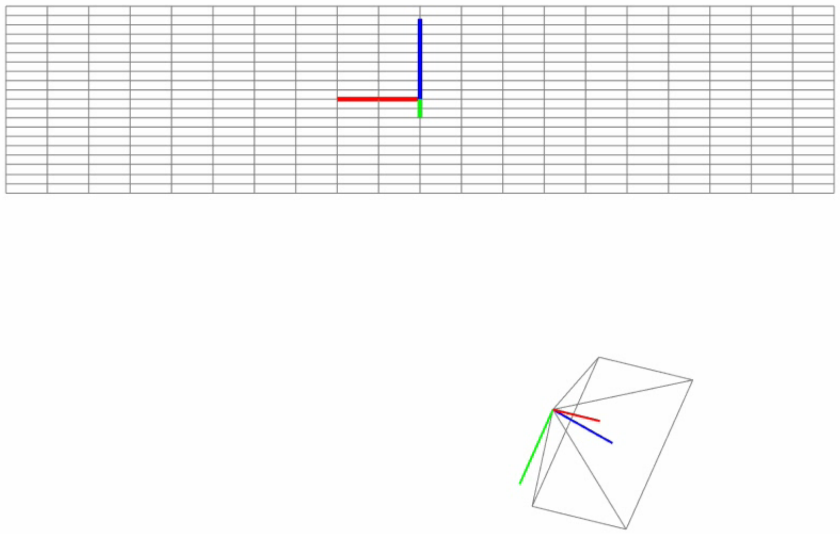
ComputeCameraFromHomography.m

```
function ComputeCameraFromHomography

f = 1300;
K = [f 0 size(im,2)/2;
     0 f size(im,1)/2;
     0 0 1];

m11 = [2145;2120;1];m12 = [2566;1191;1];
m13 = [1804;935;1];m14 = [1050;1320;1];

u = [m11(1:2)';m12(1:2)';m13(1:2)'; m14(1:2)'];
X = [0 0;1 0;1 1;0 1];
X = [X ones(4,1)]; % homogeneous coordinate

H = ComputeHomography(u, X)

denom = norm(inv(K)*H(:,1));
r1 = inv(K)*H(:,1)/denom;
r2 = inv(K)*H(:,2)/denom;
t = inv(K)*H(:,3)/denom;

r3 = Vec2Skew(r1)*r2;
```

Homography Mapping

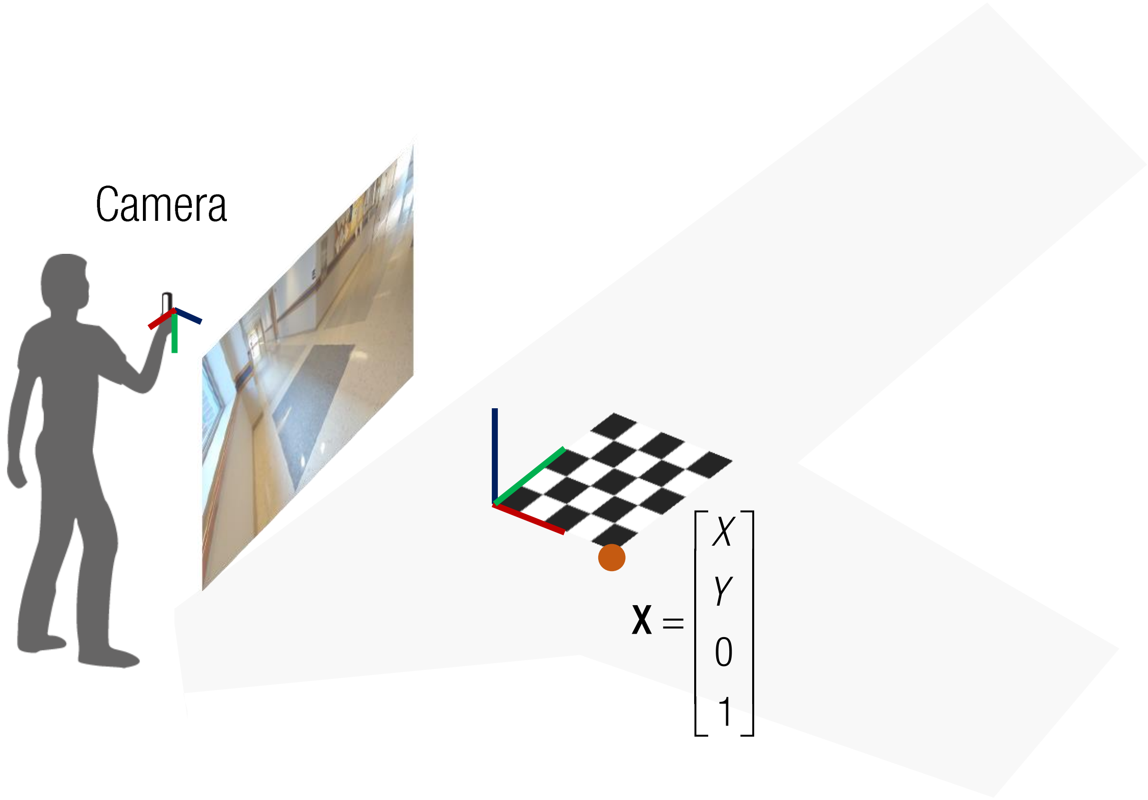


H

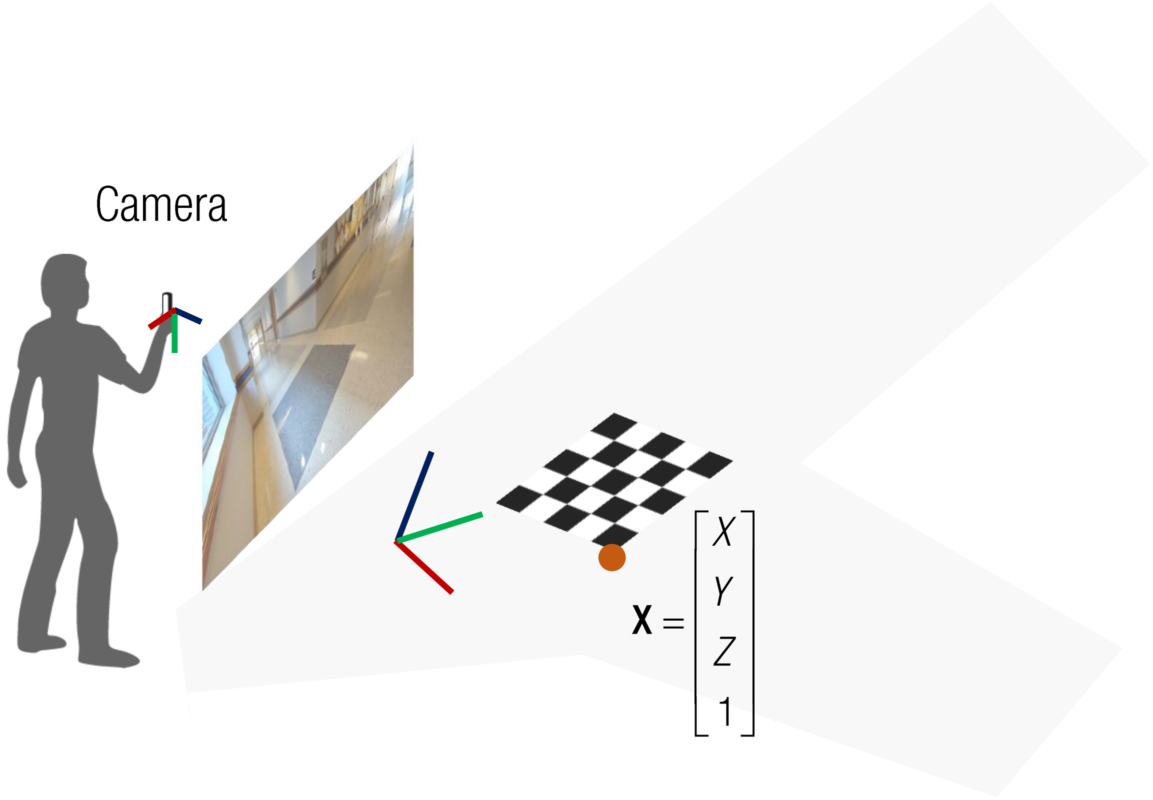


-H

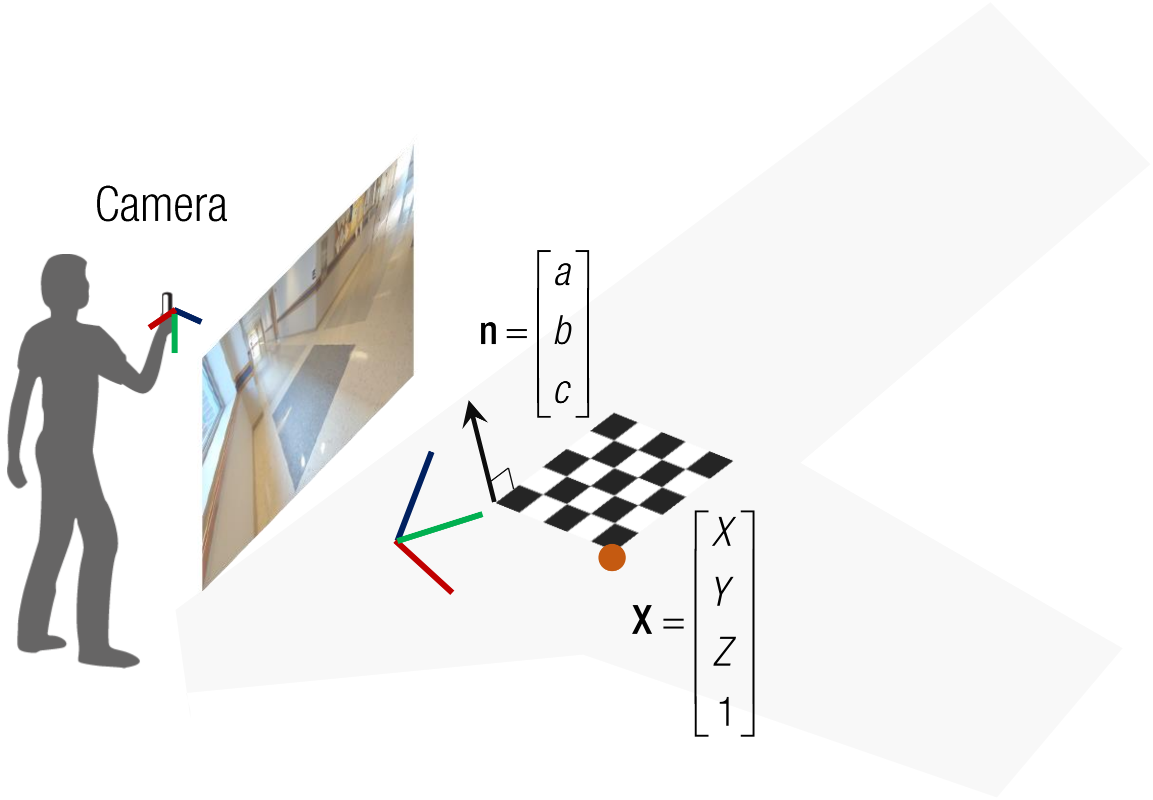
Plane Representation



Plane Representation



Plane Representation



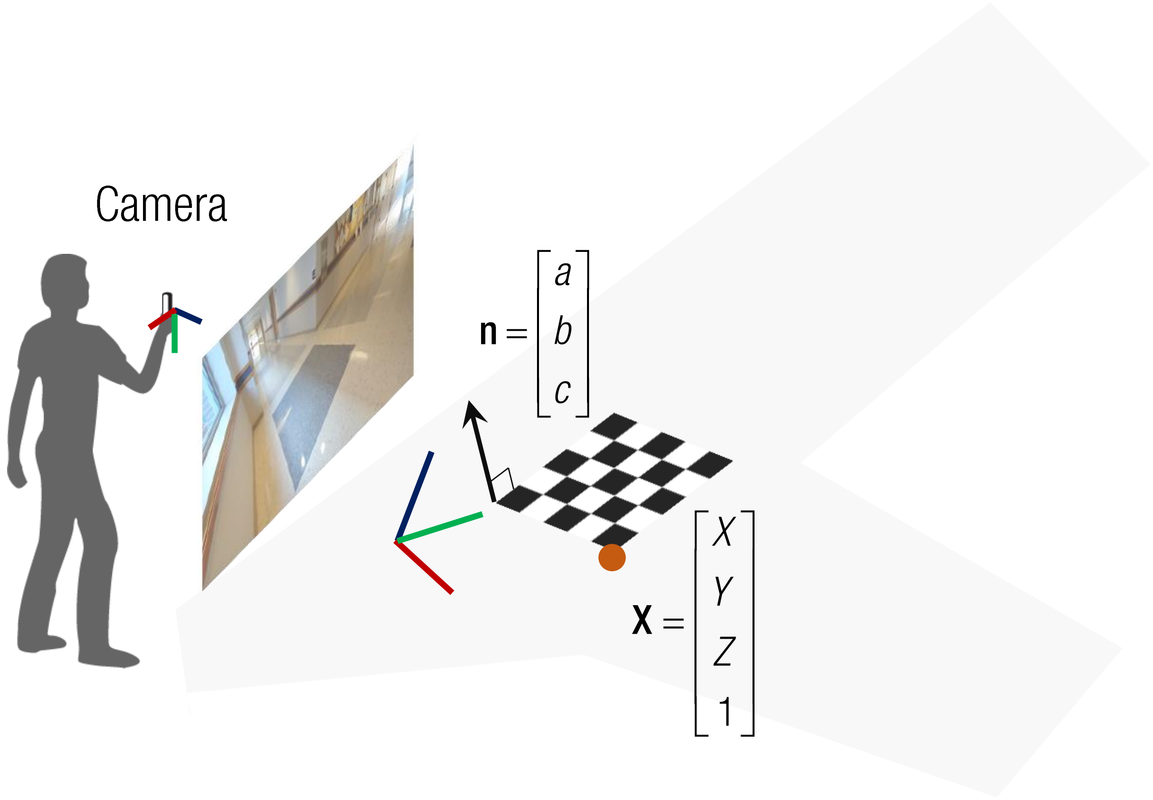
Plane equation:

$$aX + bY + cZ + d = 0$$

Surface normal:

$$\mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Plane Representation



Plane equation:

$$aX + bY + cZ + d = 0$$

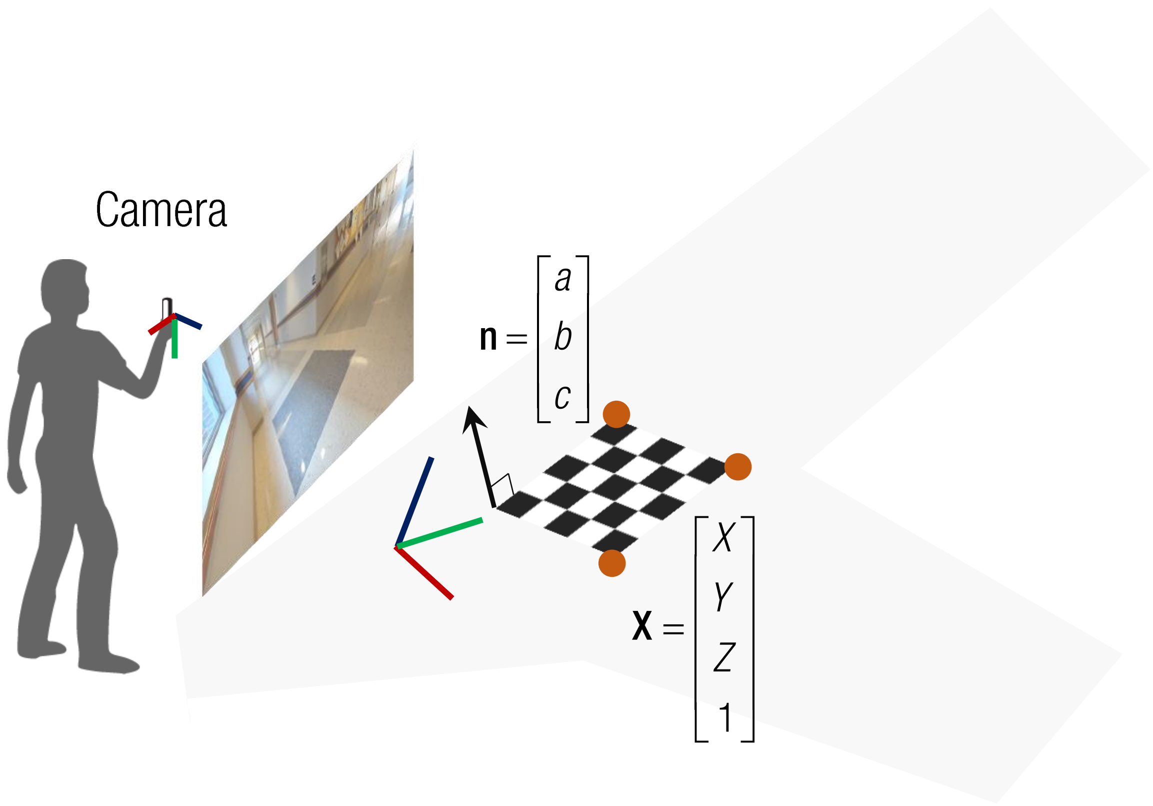
Surface normal:

$$\mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

How many points to define a plane?

$$aX + bY + cZ + d = 0$$

Plane Representation



Plane equation:

$$aX + bY + cZ + d = 0$$

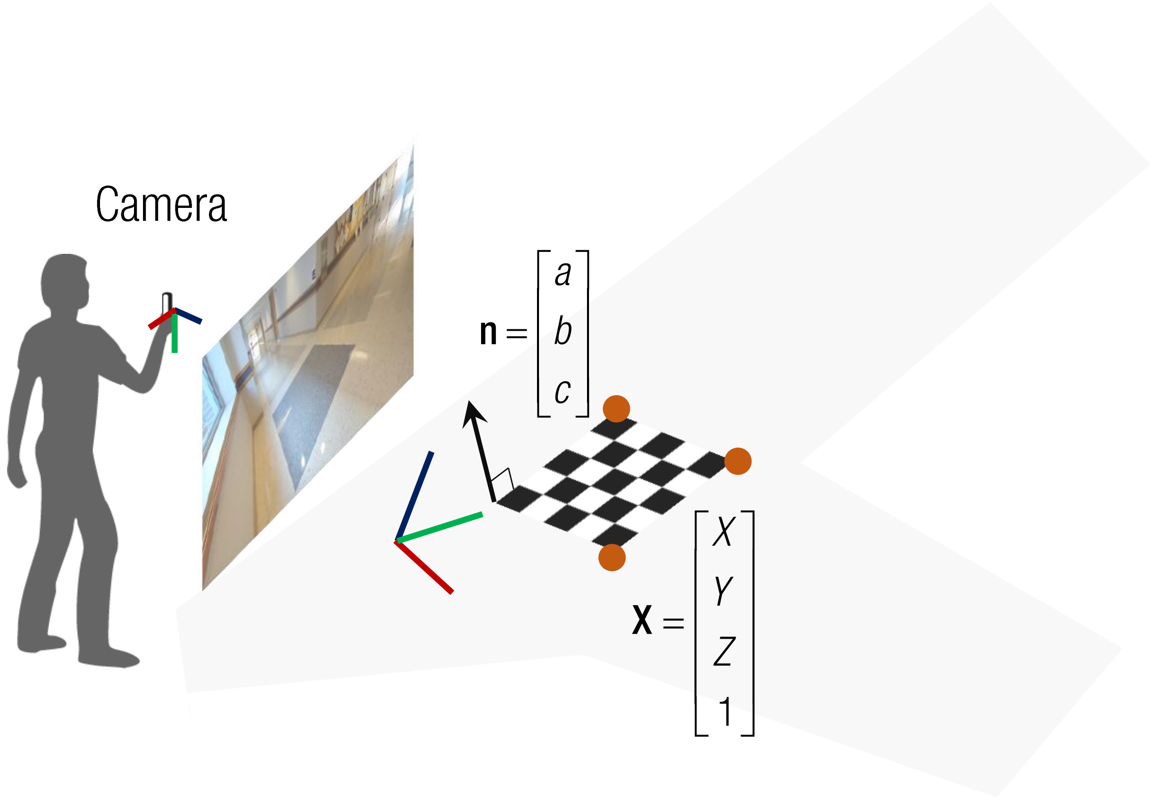
Surface normal:

$$\mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

How many points to define a plane?

$$aX_1 + bY_1 + cZ_1 + d = 0$$
$$aX_2 + bY_2 + cZ_2 + d = 0$$
$$aX_3 + bY_3 + cZ_3 + d = 0$$

Plane Representation



Plane equation:

$$aX + bY + cZ + d = 0$$

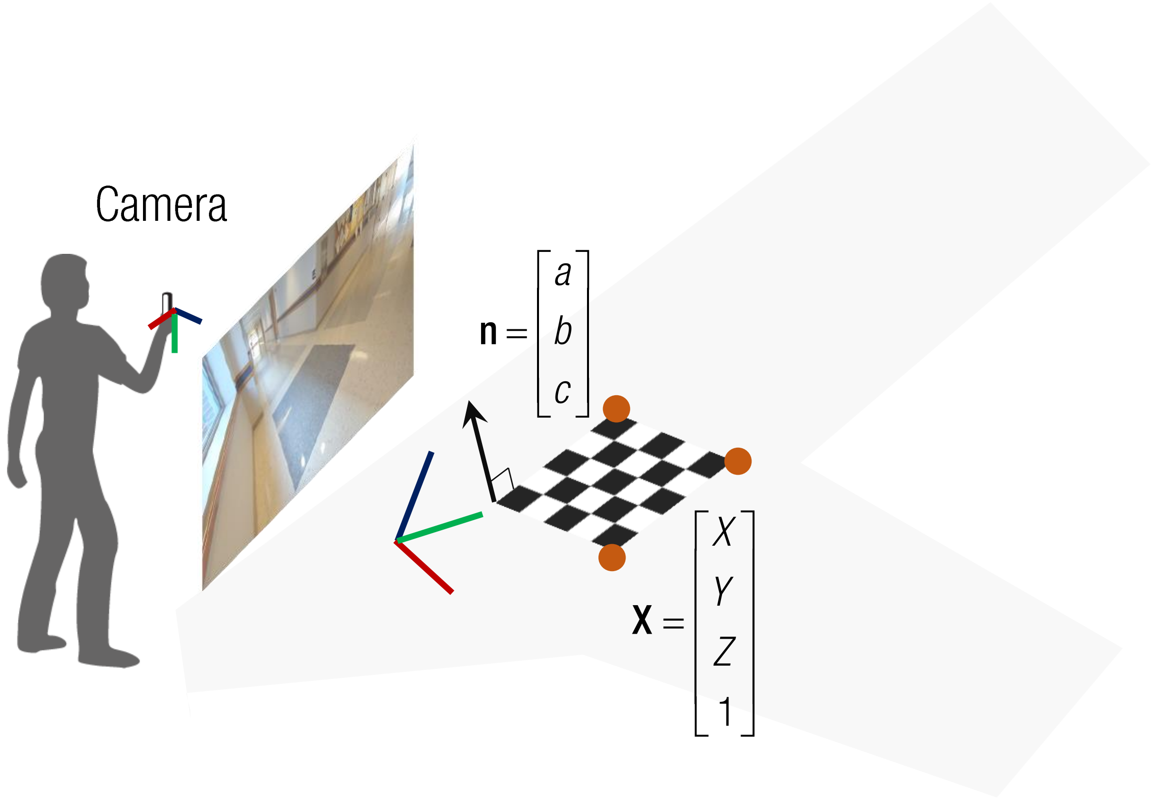
Surface normal:

$$\mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

How many points to define a plane?

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ X_3 & Y_3 & Z_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Plane Representation



Plane equation:

$$aX + bY + cZ + d = 0$$

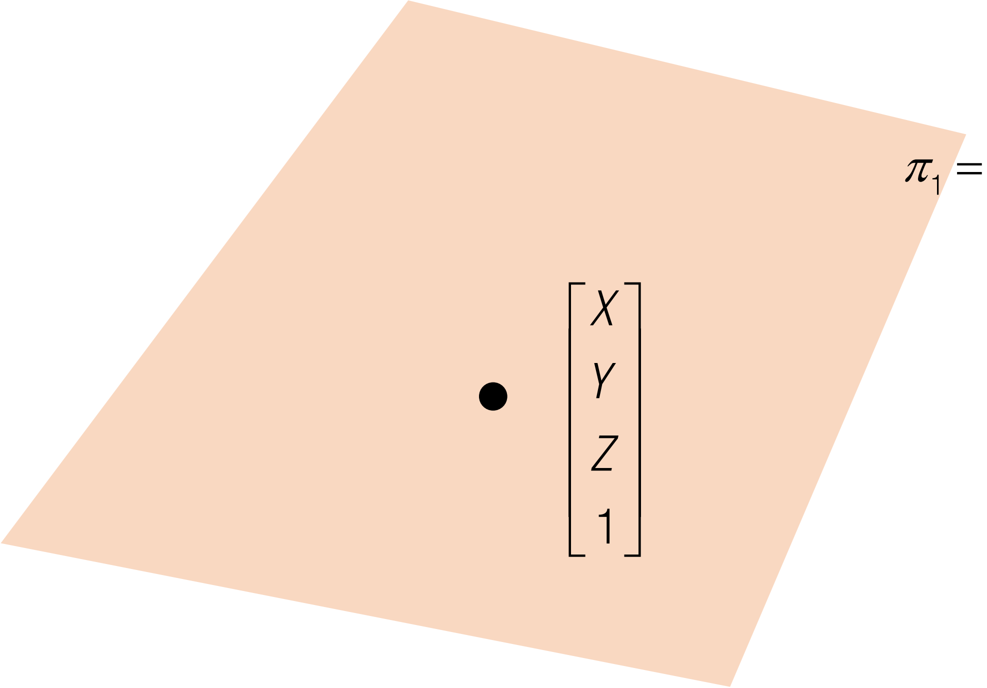
Surface normal:

$$\mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

How many points to define a plane?

$$\underbrace{\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ X_3 & Y_3 & Z_3 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}}$$

Plane-Point Duality



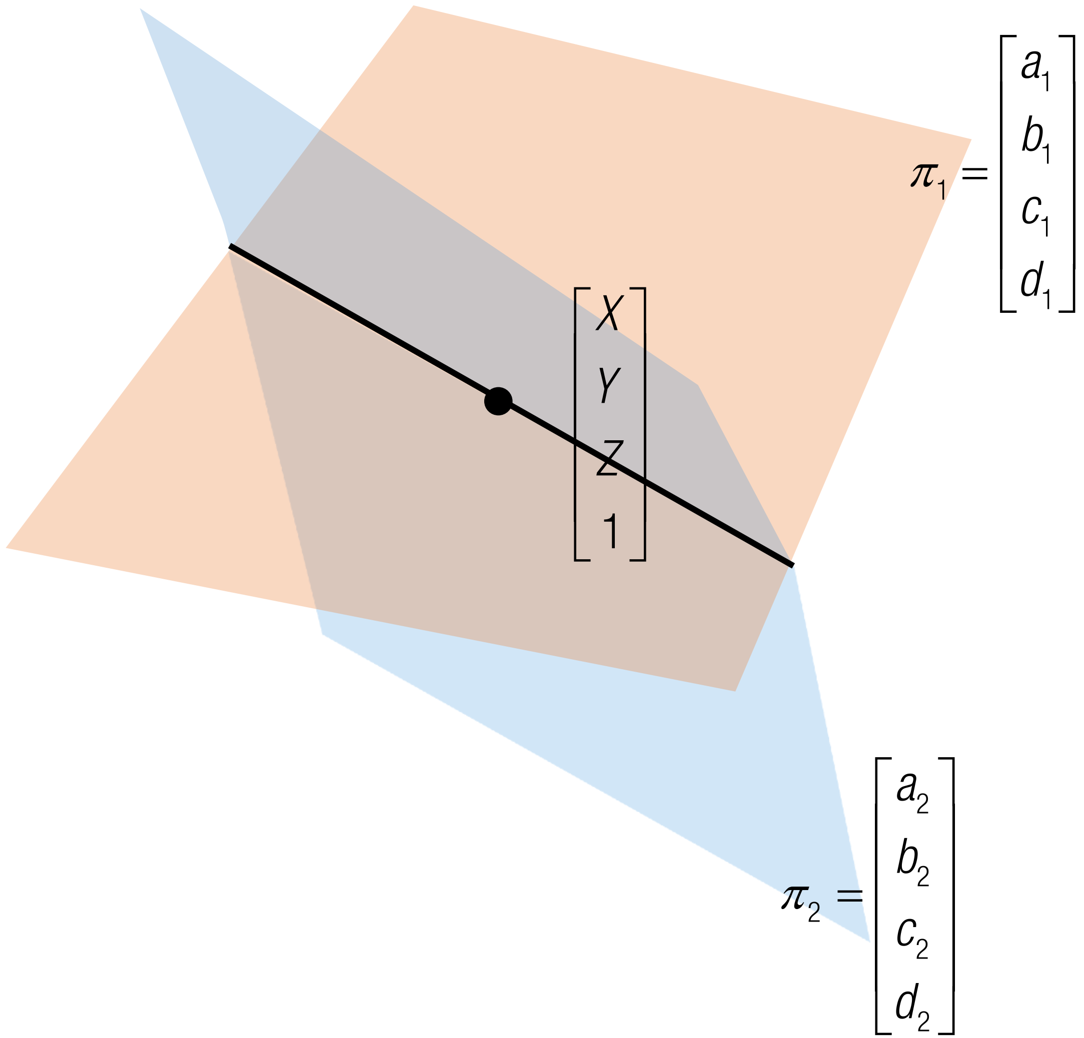
$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\pi_1 = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix}$$

Plane equation:

$$a_1X + b_1Y + c_1Z + d_1 = 0$$

Plane-Point Duality

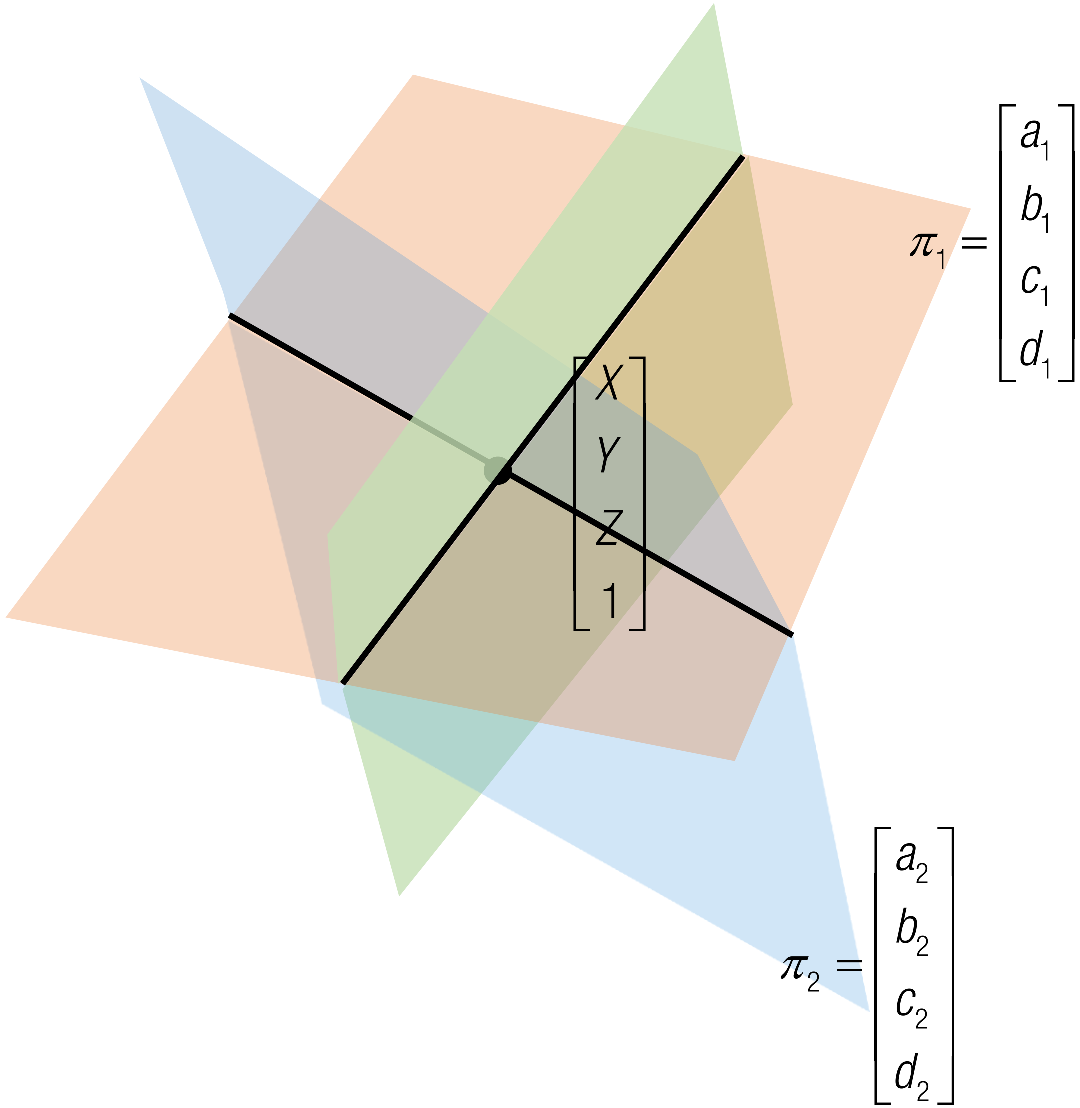


Plane equation:

$$a_1 X + b_1 Y + c_1 Z + d_1 = 0$$

$$a_2 X + b_2 Y + c_2 Z + d_2 = 0$$

Plane-Point Duality



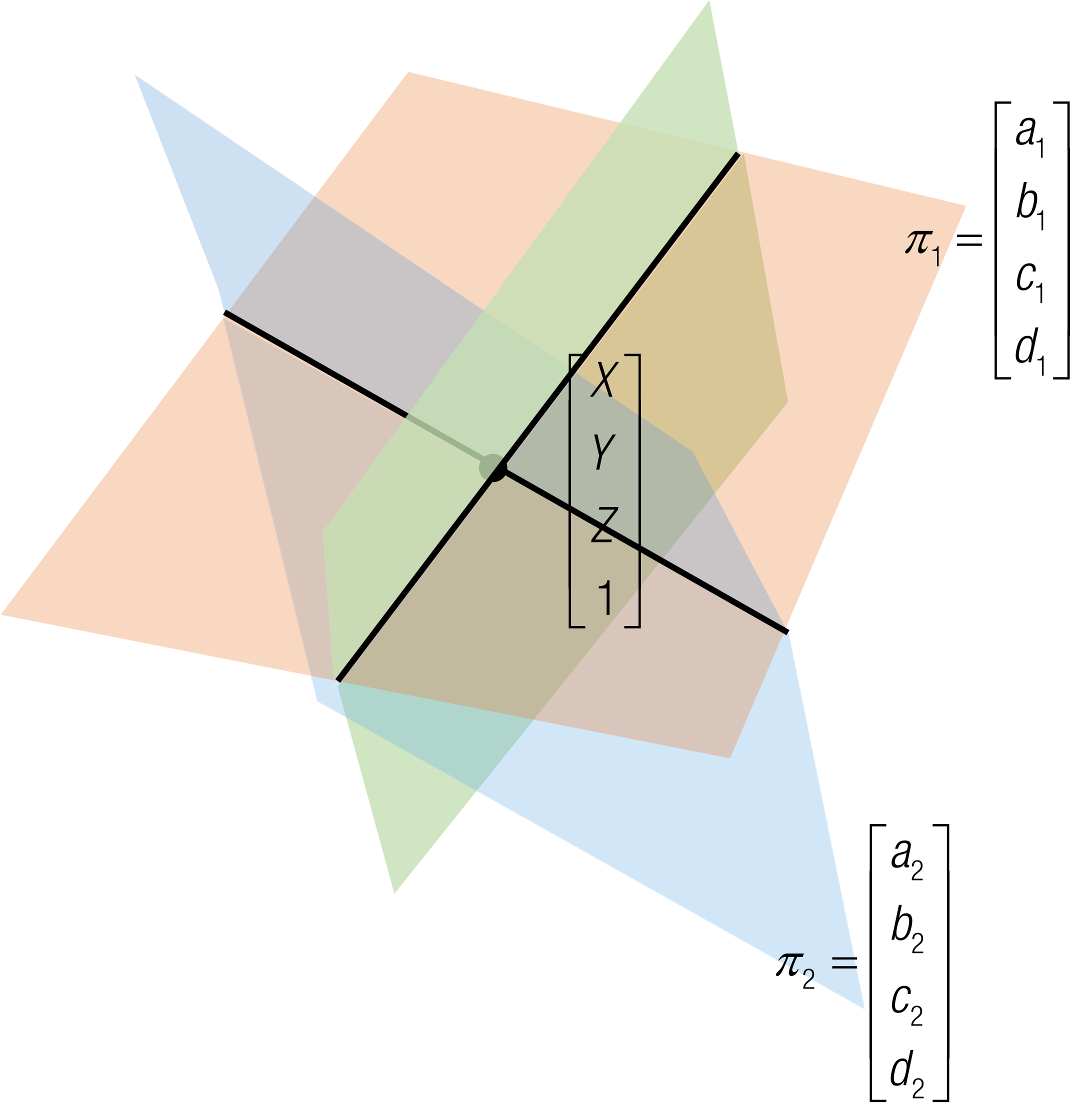
Plane equation:

$$a_1X + b_1Y + c_1Z + d_1 = 0$$

$$a_2X + b_2Y + c_2Z + d_2 = 0$$

$$a_3X + b_3Y + c_3Z + d_3 = 0$$

Plane-Point Duality



Plane equation:

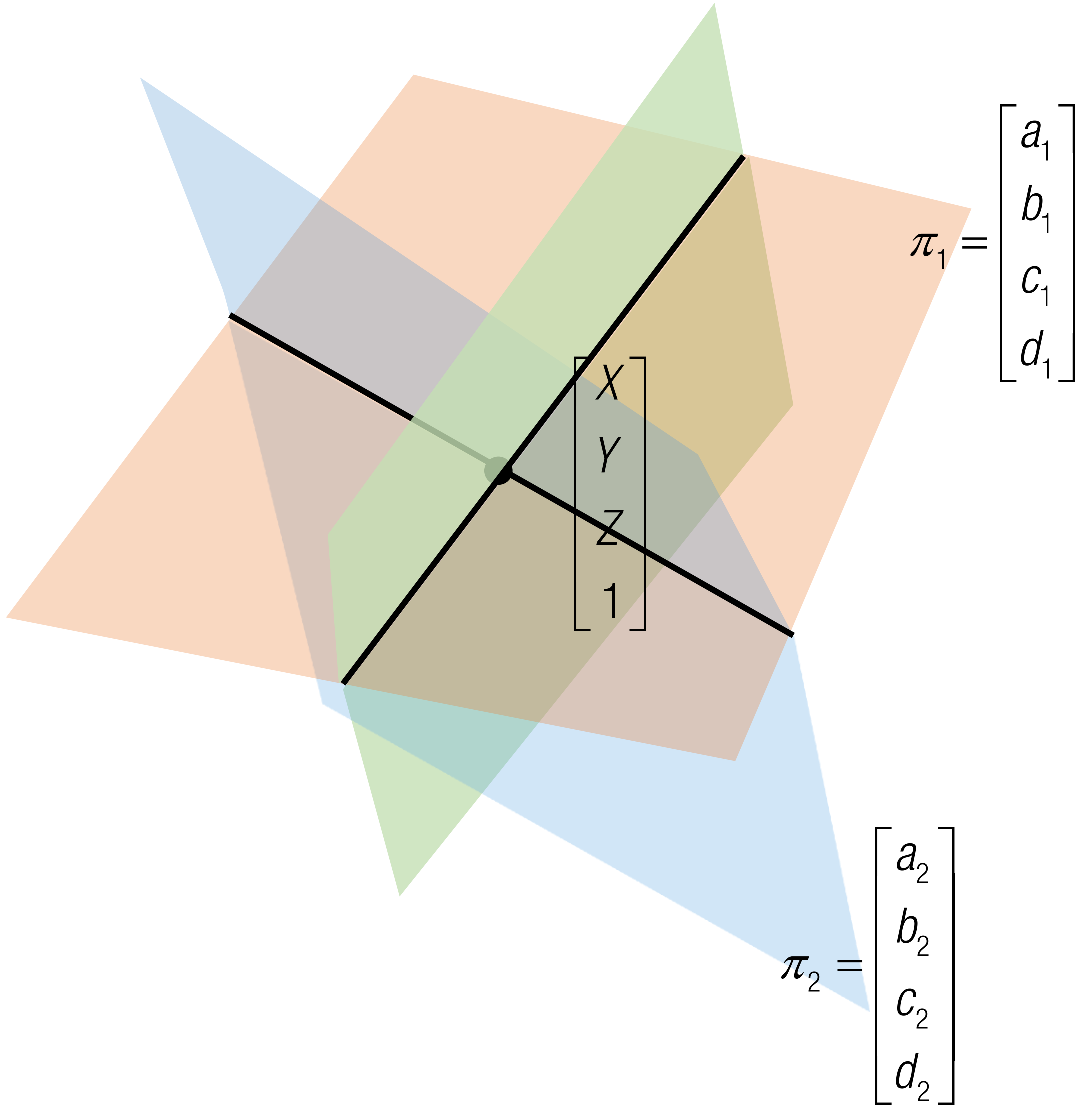
$$a_1X + b_1Y + c_1Z + d_1 = 0$$

$$a_2X + b_2Y + c_2Z + d_2 = 0$$

$$a_3X + b_3Y + c_3Z + d_3 = 0$$

$$\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Plane-Point Duality



Plane equation:

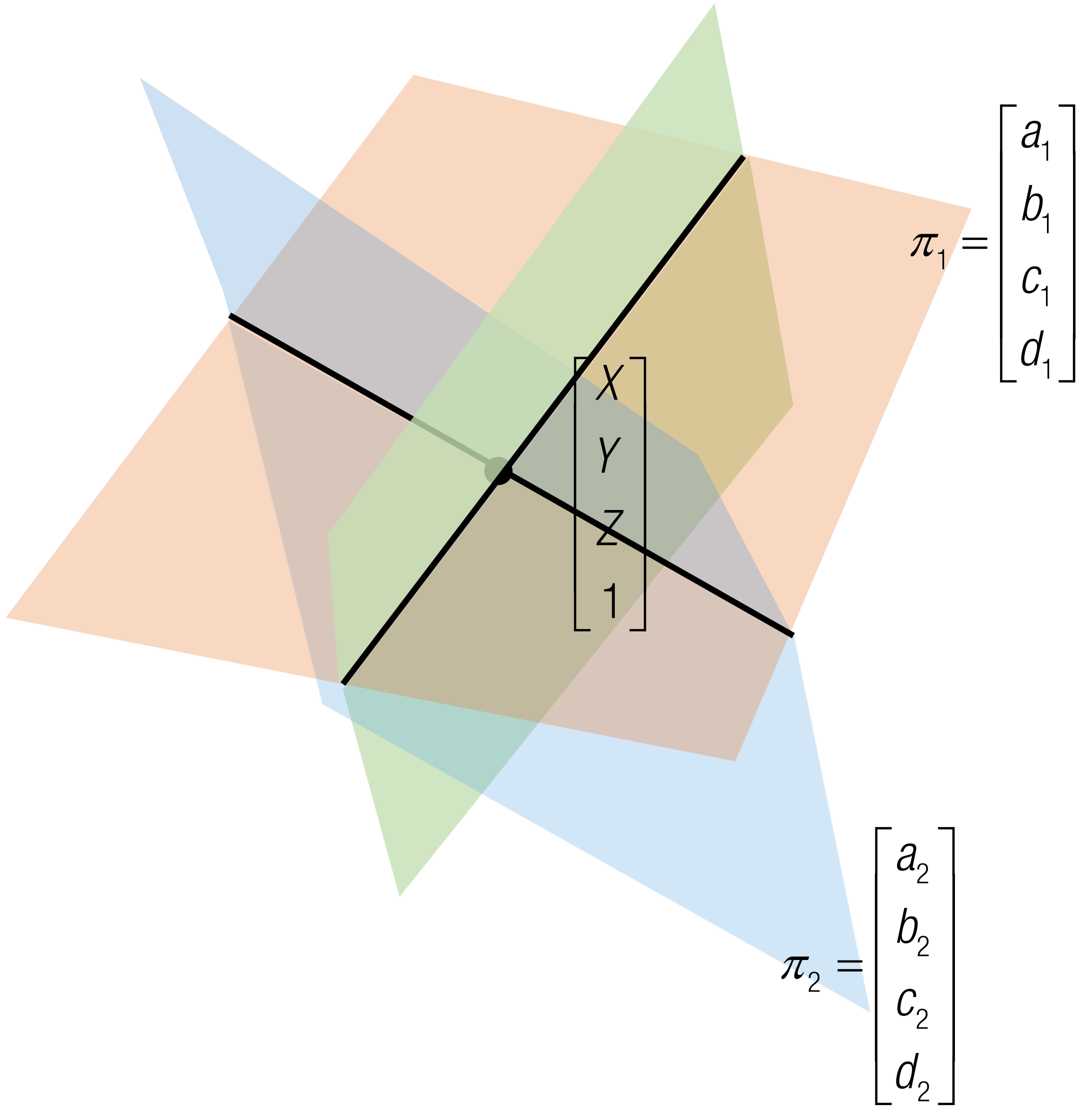
$$a_1X + b_1Y + c_1Z + d_1 = 0$$

$$a_2X + b_2Y + c_2Z + d_2 = 0$$

$$a_3X + b_3Y + c_3Z + d_3 = 0$$

$$\underbrace{\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}}$$

Plane-Point Duality

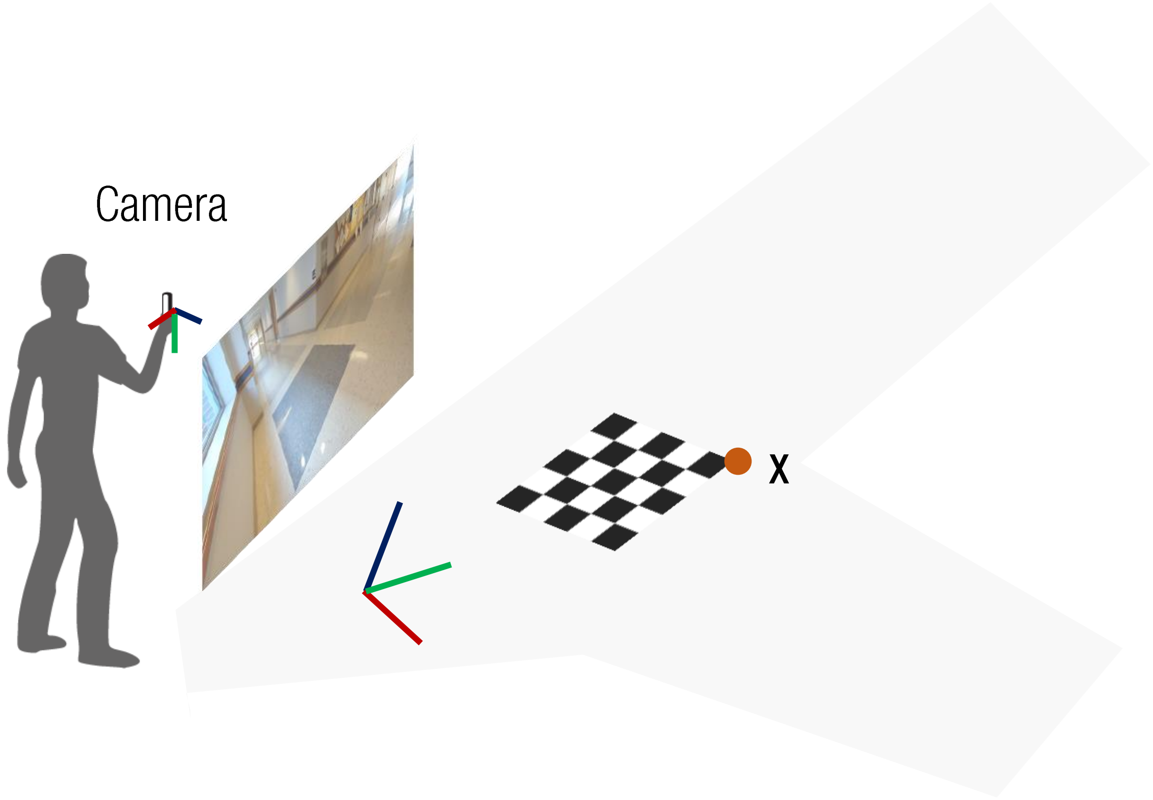


$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ X_3 & Y_3 & Z_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Given any formula, we can switch the meaning of point and plane to get another formula.

Plane Representation



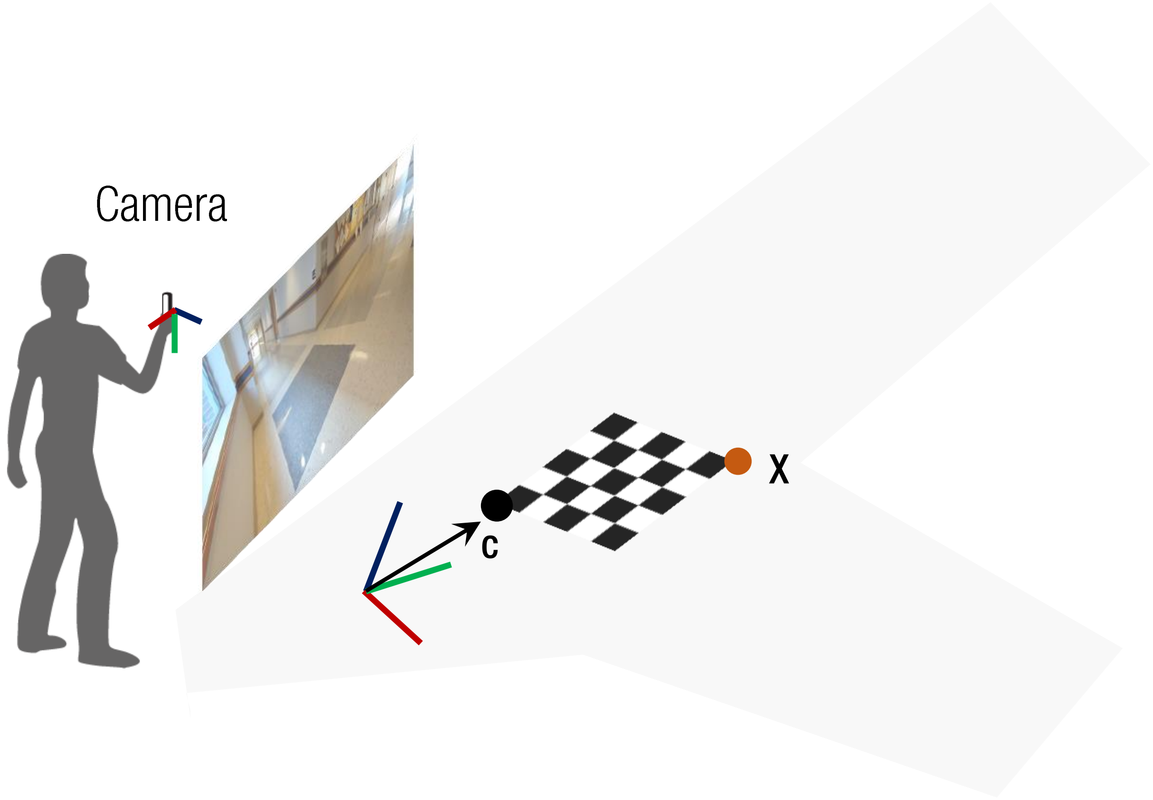
How to parametrize a point in the plane?

$$aX + bY + cZ + d = 0$$

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

3DOF

Plane Representation



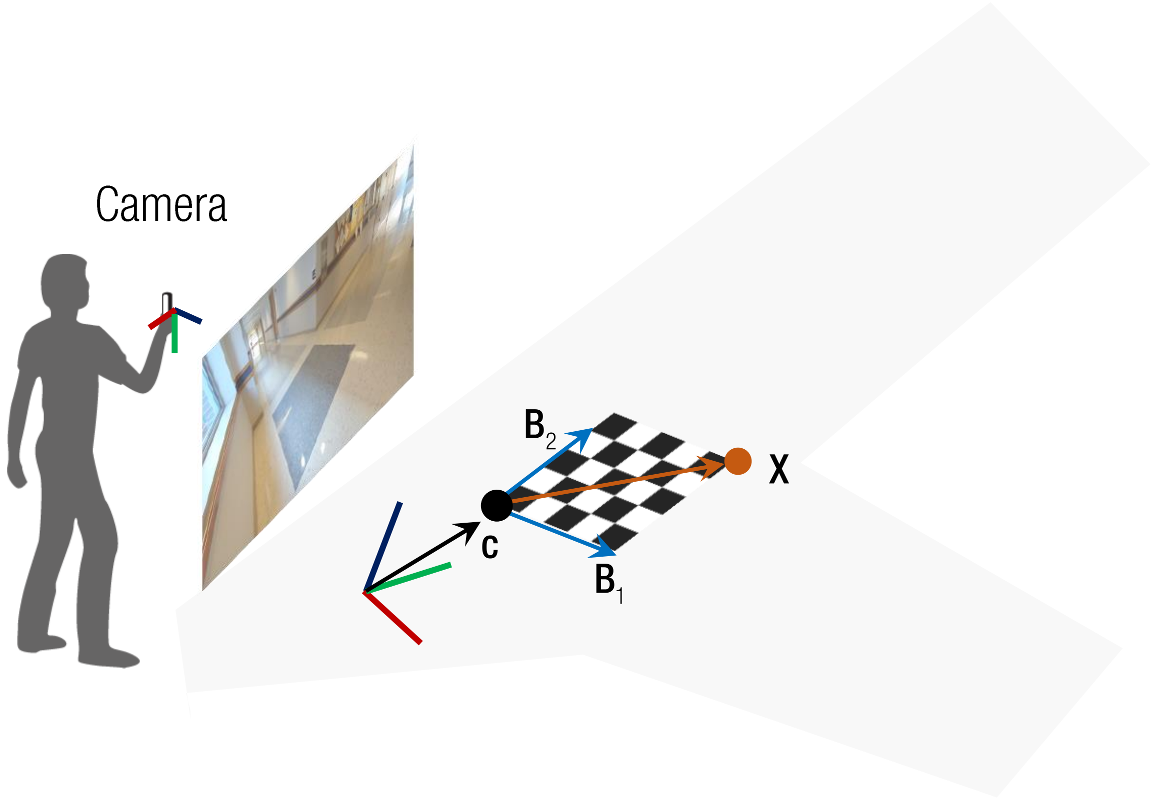
How to parametrize a point in the plane?

$$aX + bY + cZ + d = 0$$

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{c} +$$

3DOF

Plane Representation



How to parametrize a point in the plane?

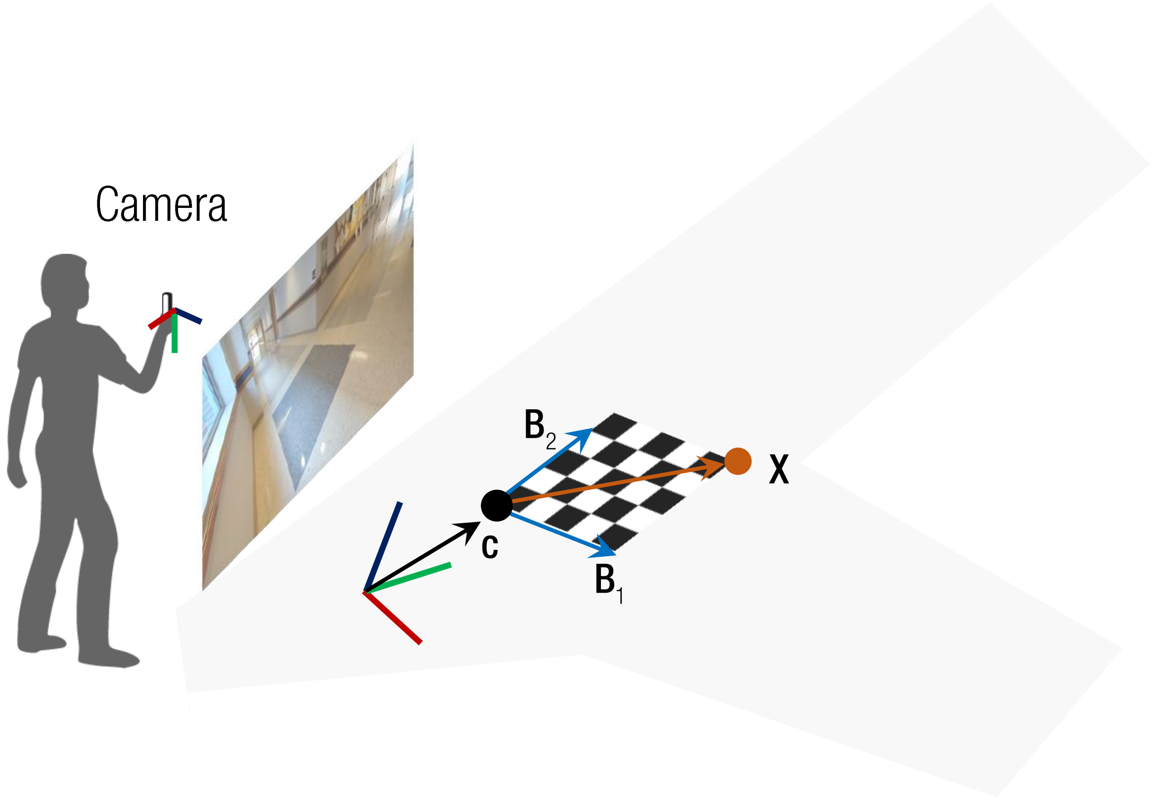
$$aX + bY + cZ + d = 0$$

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{c} + \mu_1 \mathbf{B}_1 + \mu_2 \mathbf{B}_2$$

3DOF

Basis

Plane Representation



How to parametrize a point in the plane?

$$aX + bY + cZ + d = 0$$

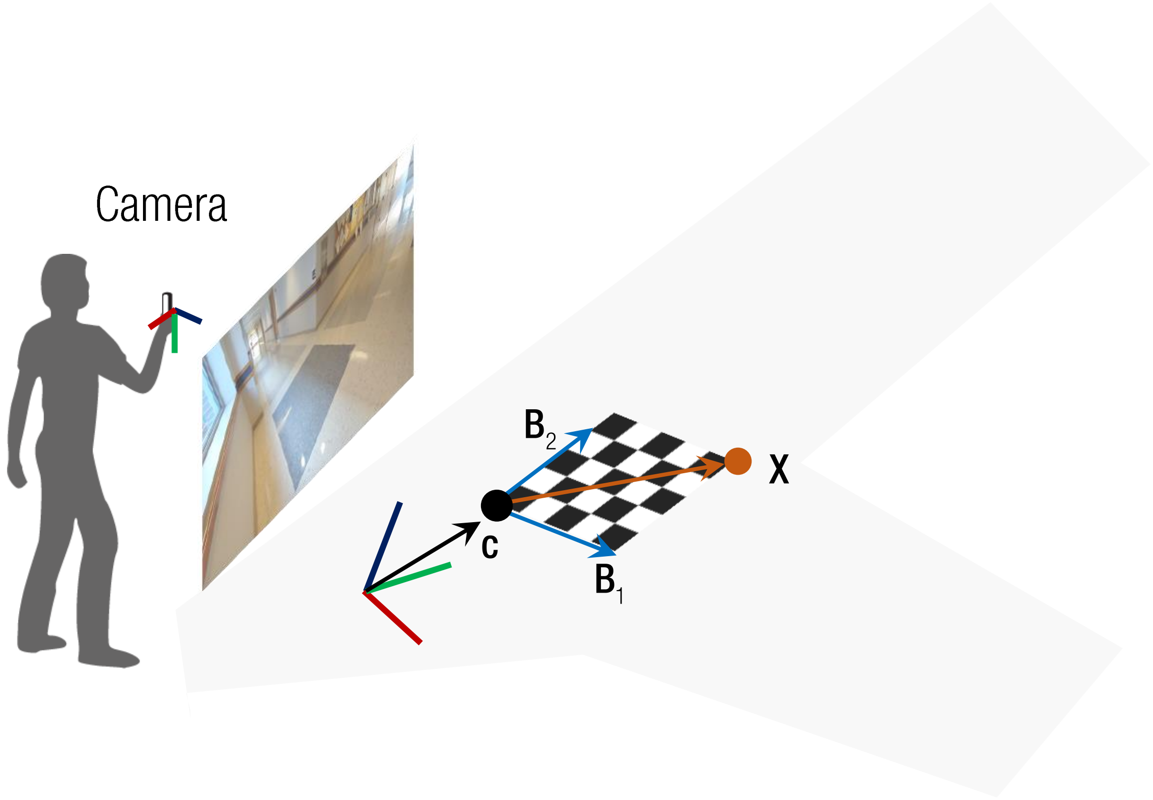
$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{c} + \mu_1 \mathbf{B}_1 + \mu_2 \mathbf{B}_2 = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 1 \end{bmatrix}$$

Basis

3DOF

2DOF

Plane Representation



How to parametrize a point in the plane?

$$aX + bY + cZ + d = 0$$

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{c} + \mu_1 \mathbf{B}_1 + \mu_2 \mathbf{B}_2 = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 1 \end{bmatrix}$$

3DOF | Basis | 2DOF

Plane projection:

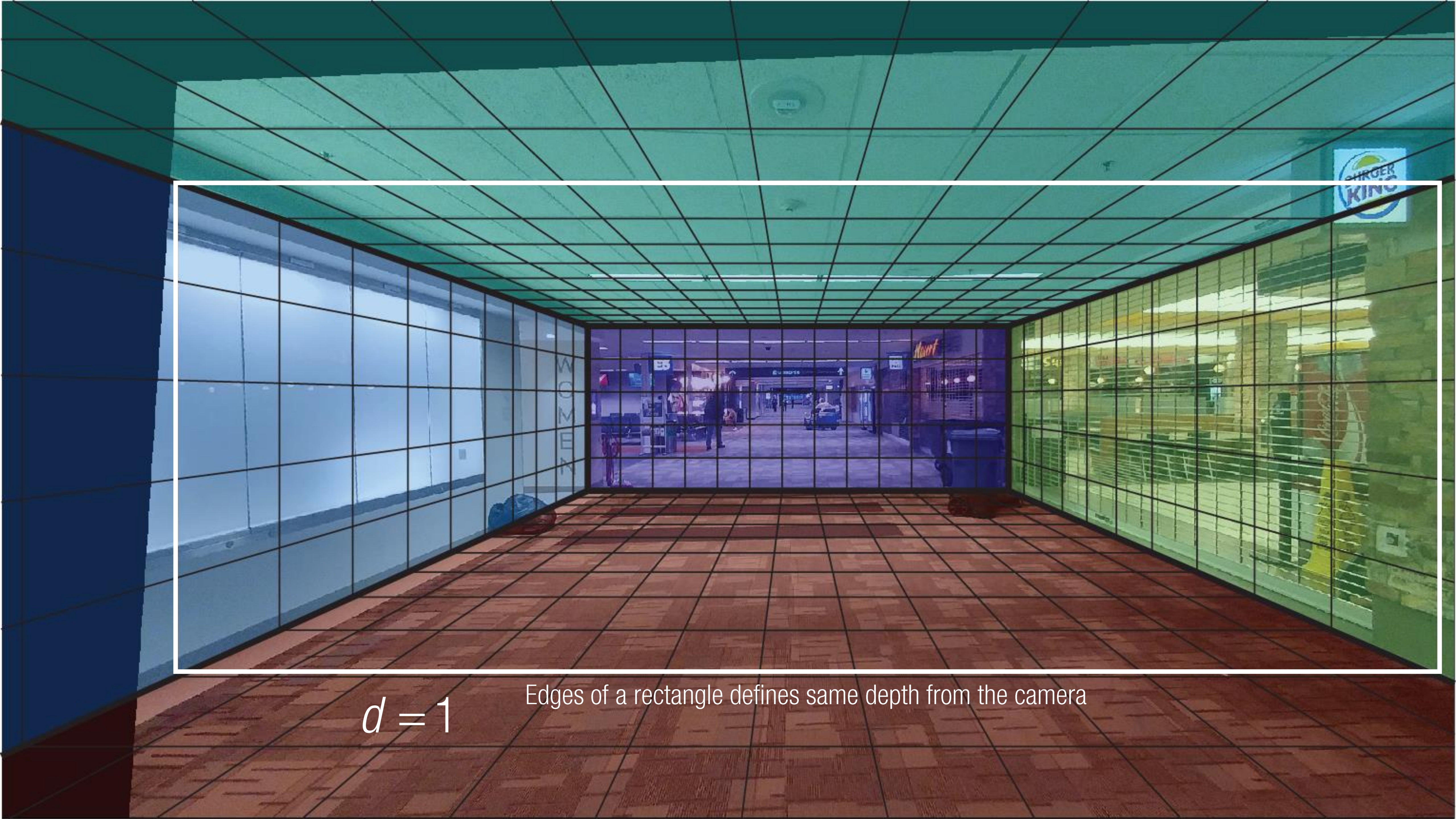
$$\lambda \mathbf{u} = \mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 1 \end{bmatrix} \\ 1 \end{bmatrix}$$

$$= \mathbf{K}\begin{bmatrix} \mathbf{R}\mathbf{B}_1 & \mathbf{R}\mathbf{B}_2 & \mathbf{R}\mathbf{c} + \mathbf{t} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 1 \end{bmatrix}$$


BURGER KING
WOMEN
E5

HW #3 Tour into your photo


BURGER KING
WOMEN



WOMEN
BURGER KING
$d = 1$
Edges of a rectangle defines same depth from the camera

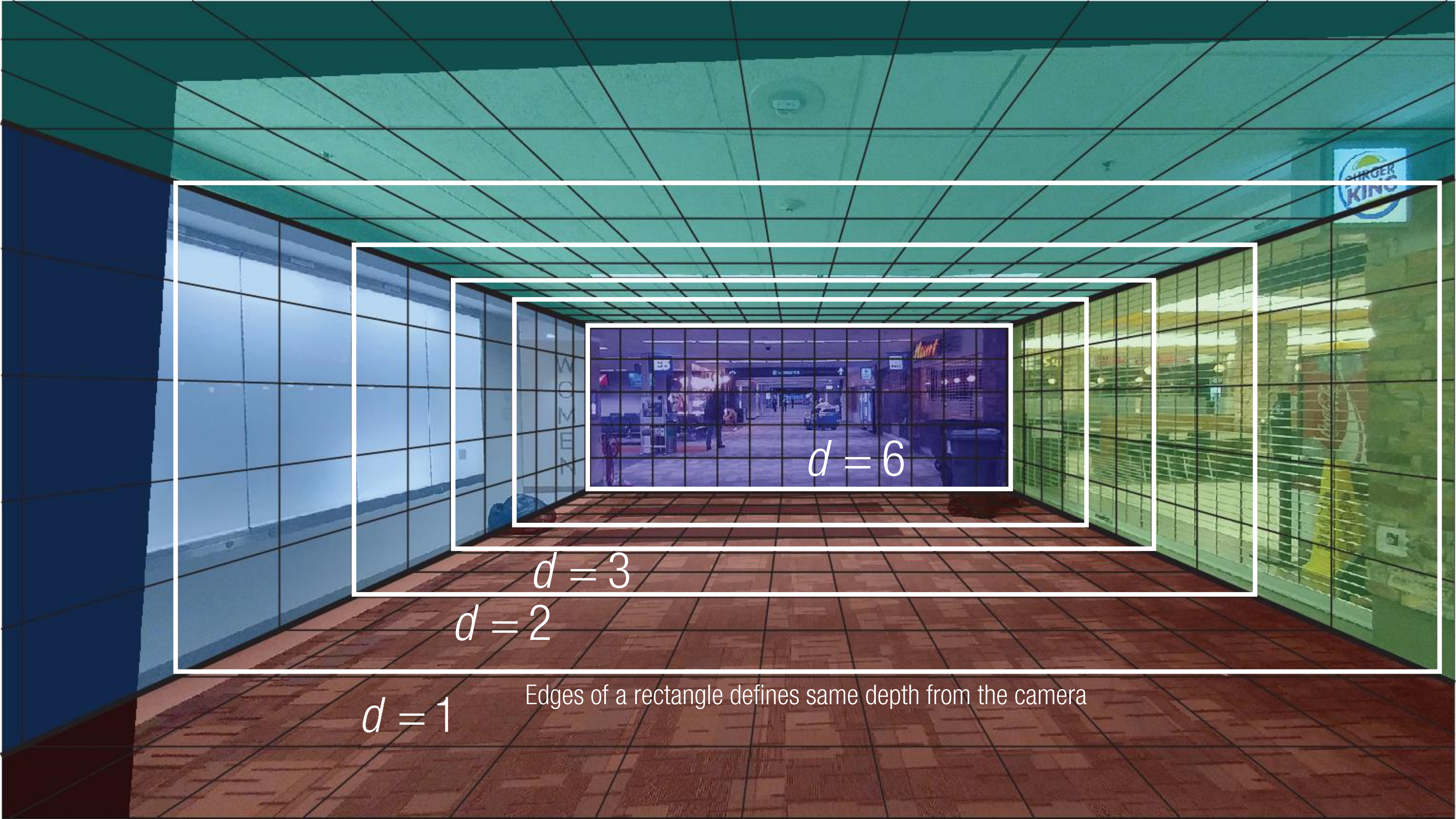
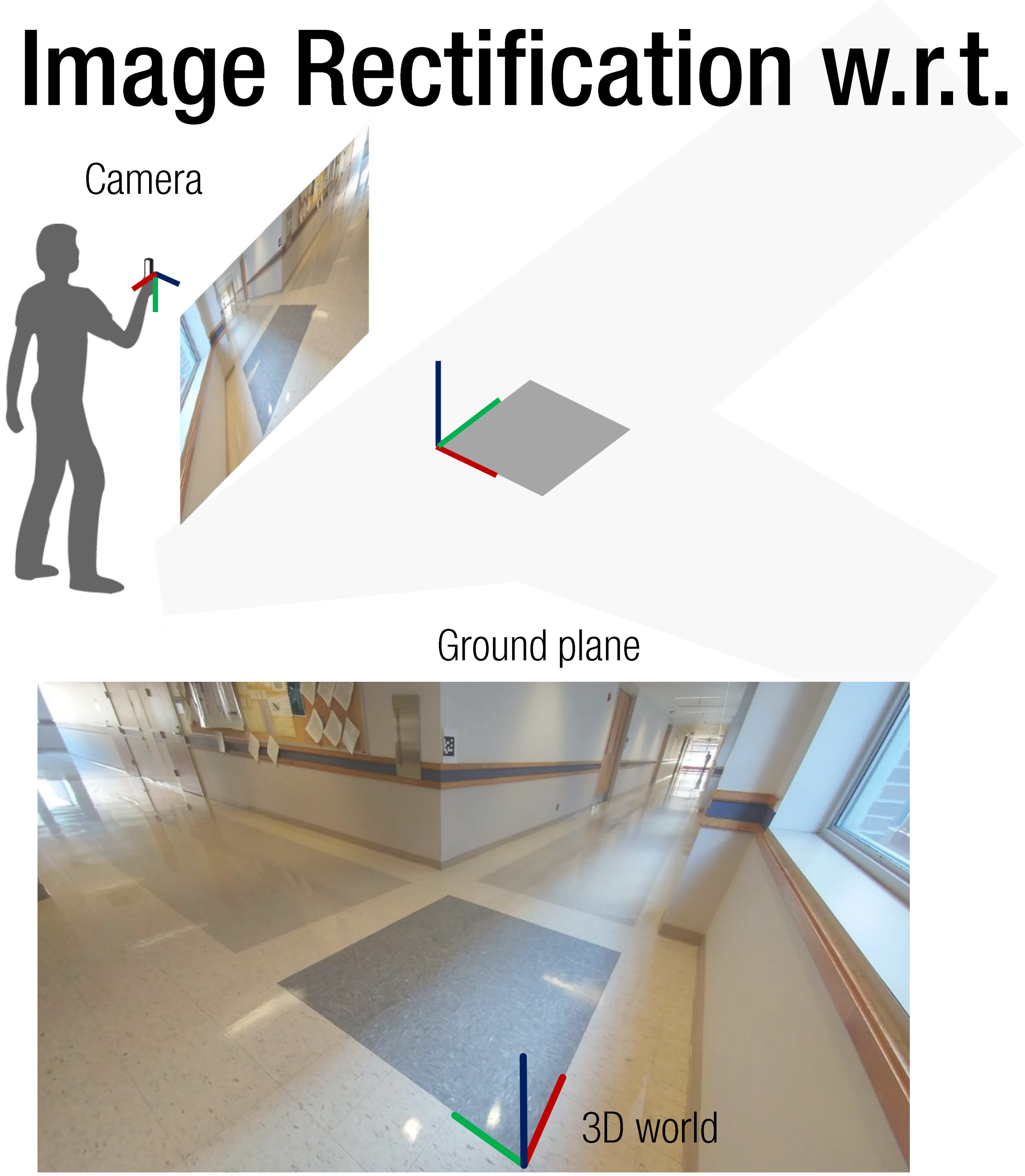

d = 6
d = 3
d = 2
Edges of a rectangle defines same depth from the camera
d = 1

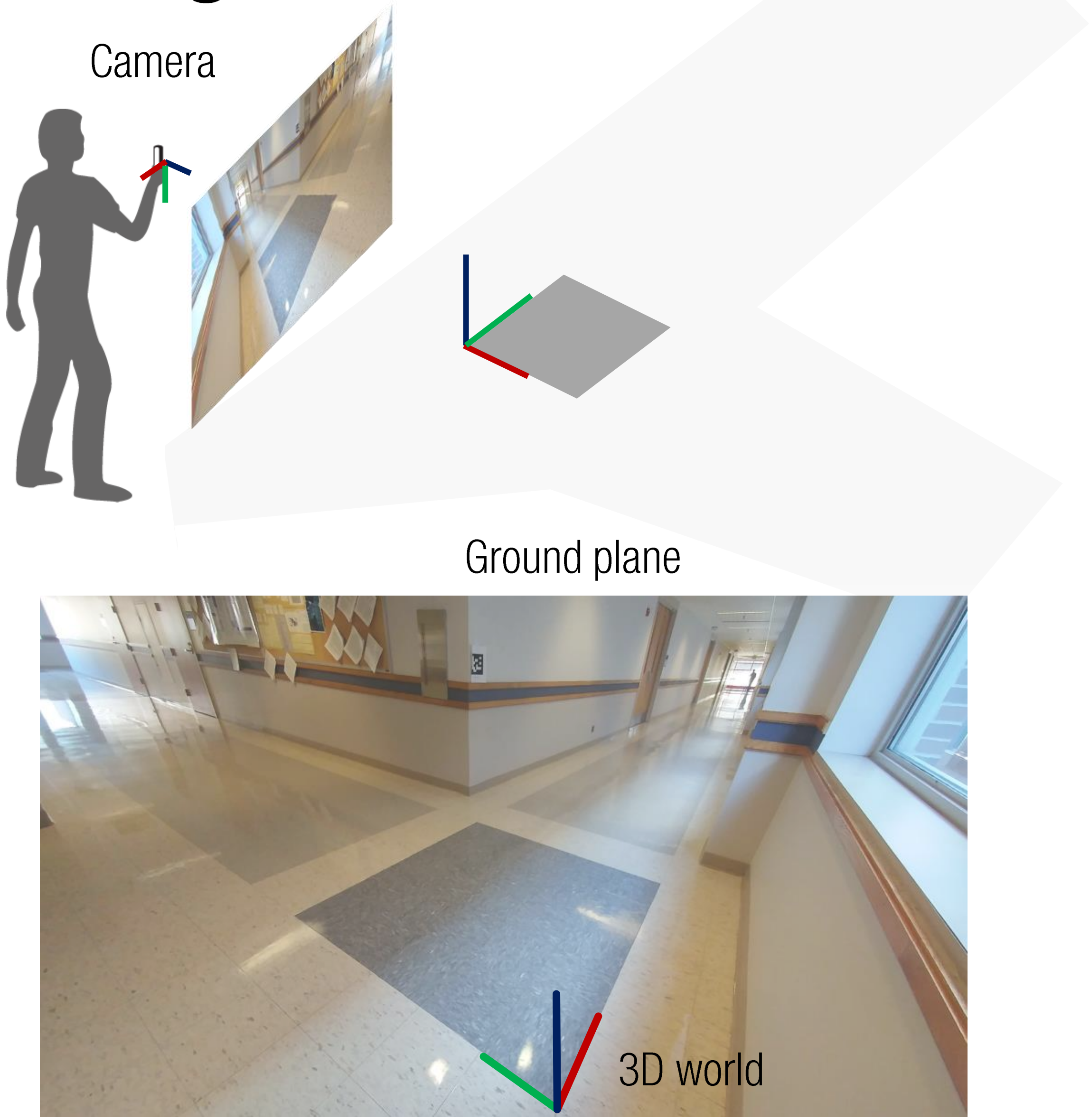
Image Rectification w.r.t. Ground Plane



How can I make my image upright?

→ Y axis of camera // Surface normal of ground plane

Image Rectification w.r.t. Ground Plane



How can I make my image upright?

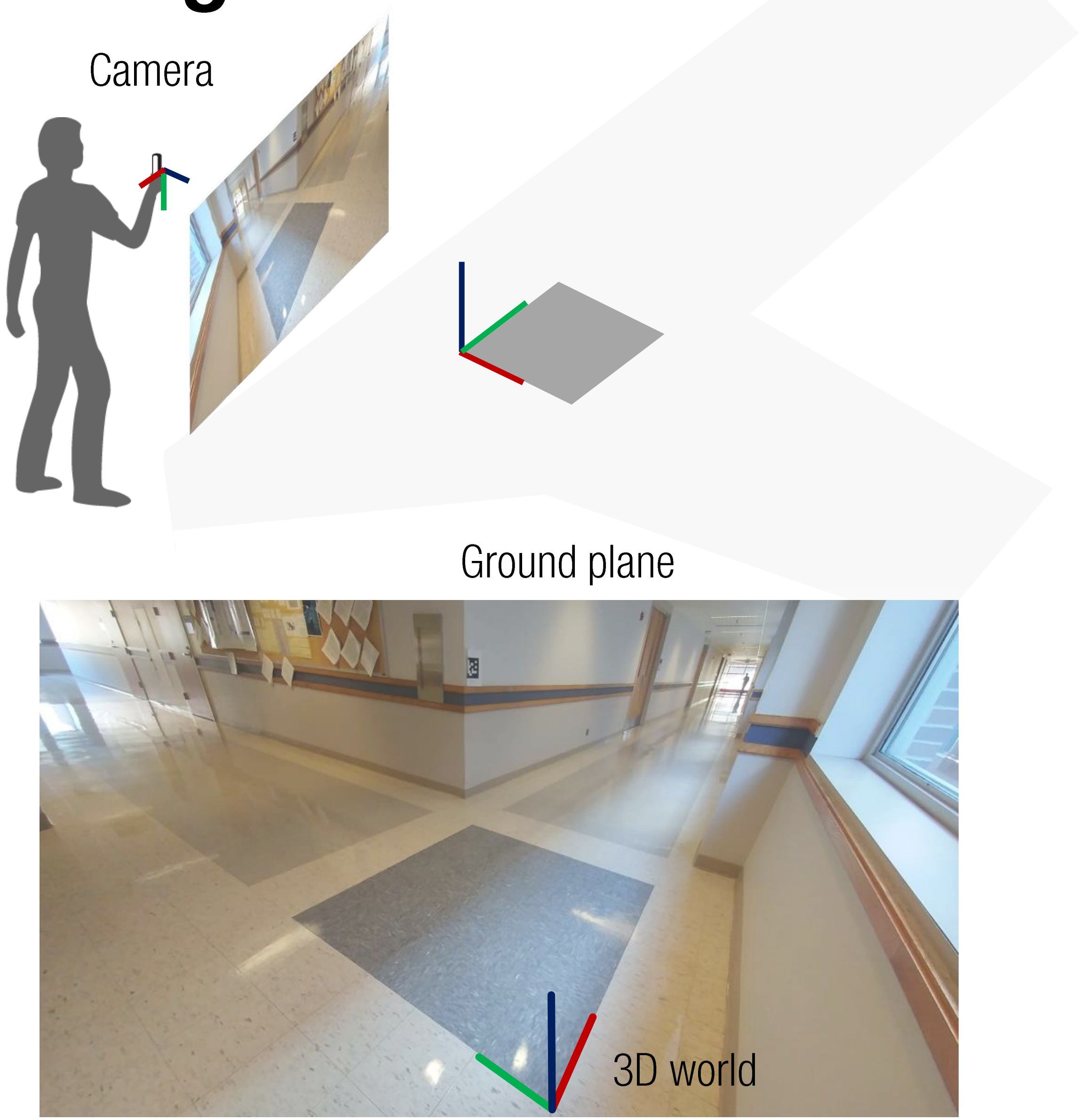
→ Y axis of camera // Surface normal of ground plane

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

Camera pose from homography

Image Rectification w.r.t. Ground Plane



How can I make my image upright?

→ Y axis of camera // Surface normal of ground plane

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

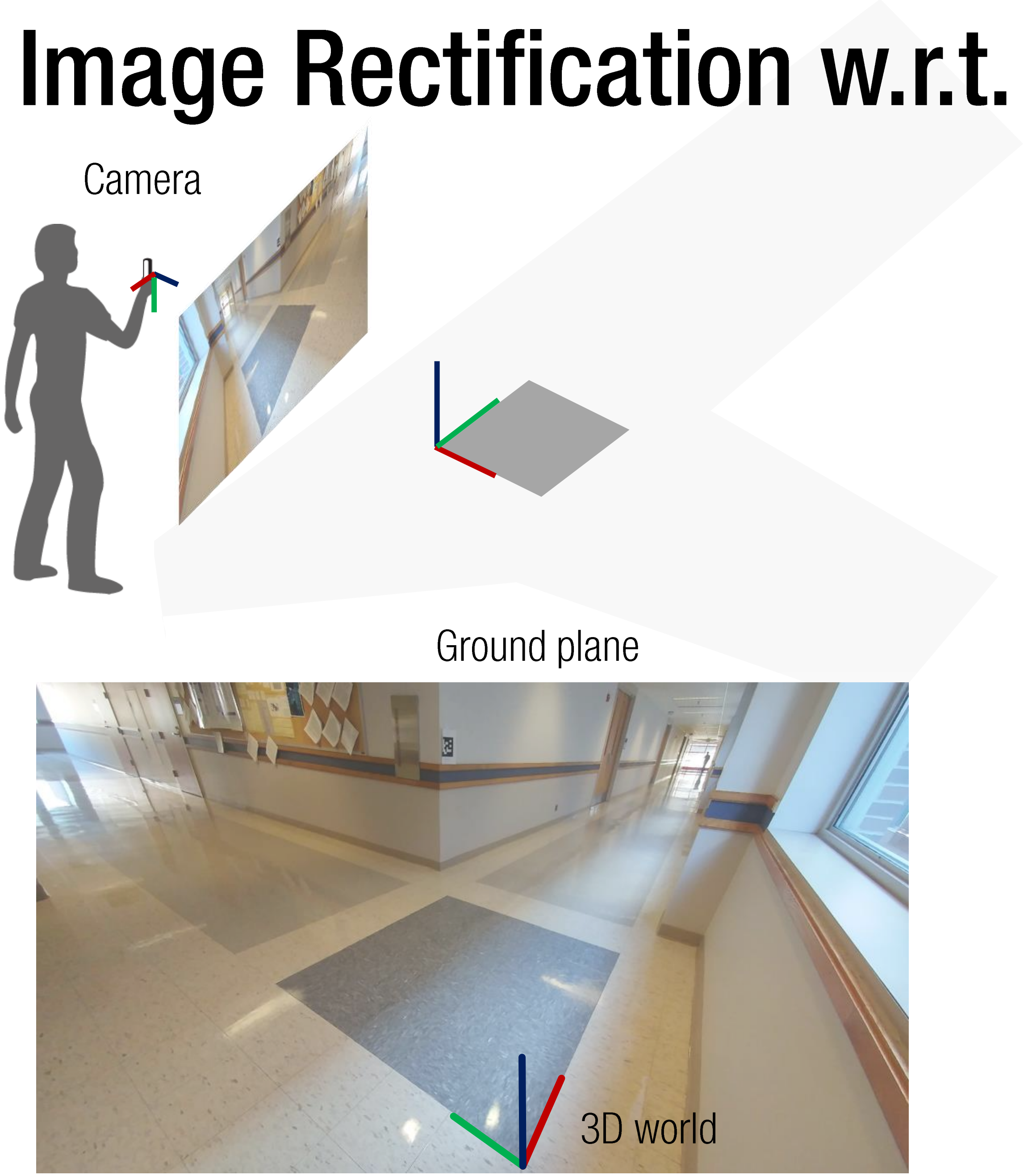
$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

Camera pose from homography

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Image rotation

Image Rectification w.r.t. Ground Plane



How can I make my image upright?

→ Y axis of camera // Surface normal of ground plane

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

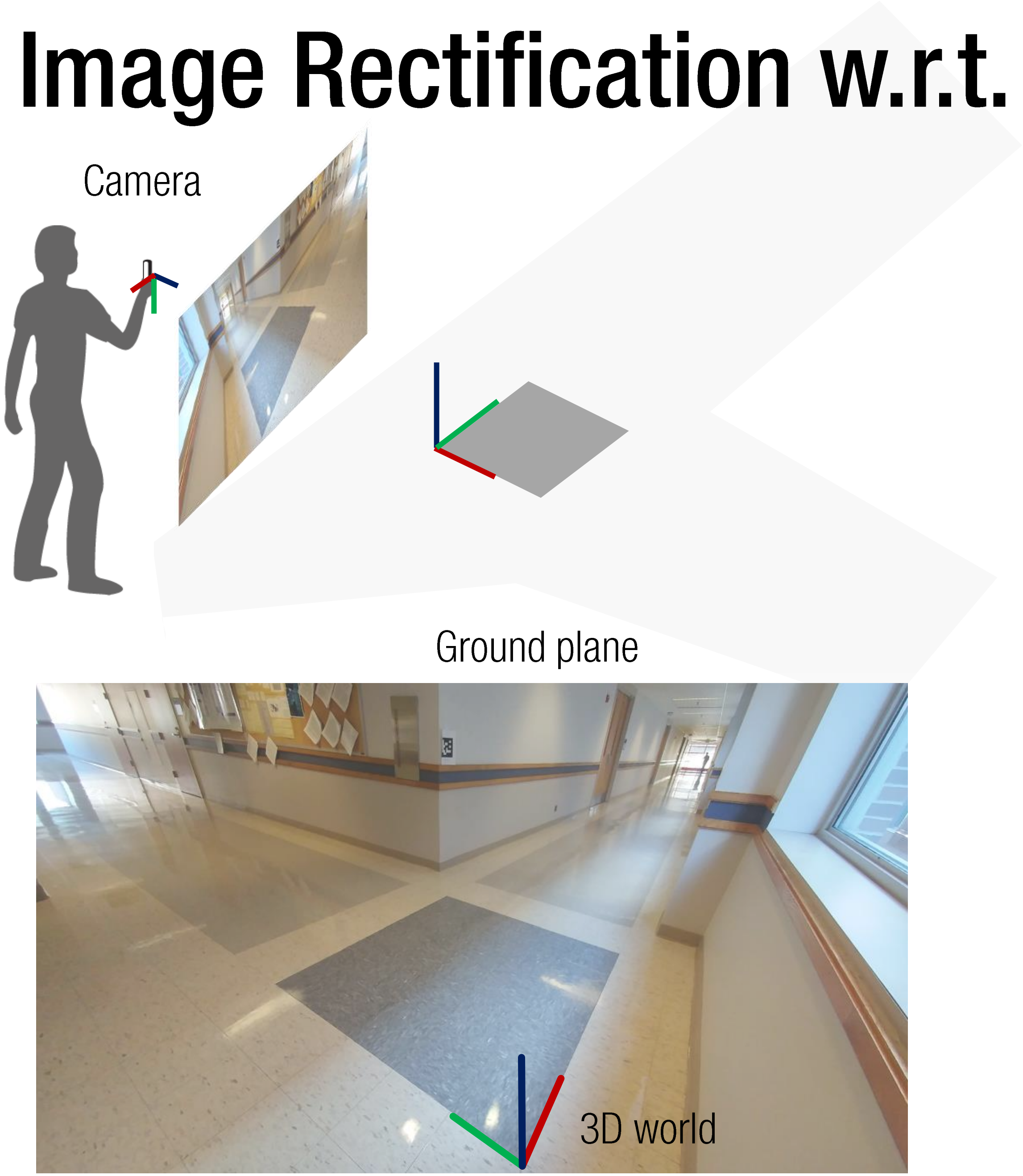
Camera pose from homography

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \longrightarrow \tilde{\mathbf{R}} = \begin{bmatrix} & \tilde{\mathbf{r}}_x & \\ 0 & 0 & -1 \\ & \tilde{\mathbf{r}}_z & \end{bmatrix}$$

Image rotation

Rectified rotation

Image Rectification w.r.t. Ground Plane



$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \longrightarrow \tilde{\mathbf{R}} = \begin{bmatrix} & \tilde{\mathbf{r}}_x & \\ 0 & 0 & -1 \\ & \tilde{\mathbf{r}}_z & \end{bmatrix}$$

Image rotation

Rectified rotation

$$\tilde{\mathbf{r}}_x = \frac{\mathbf{r}_x - \left(\mathbf{r}_x \cdot \tilde{\mathbf{r}}_y\right)\tilde{\mathbf{r}}_y}{\left\|\mathbf{r}_x - \left(\mathbf{r}_x \cdot \tilde{\mathbf{r}}_y\right)\tilde{\mathbf{r}}_y\right\|}$$

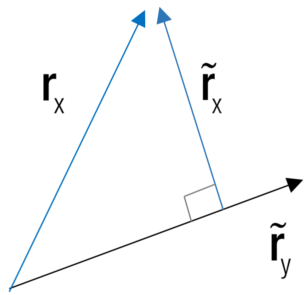
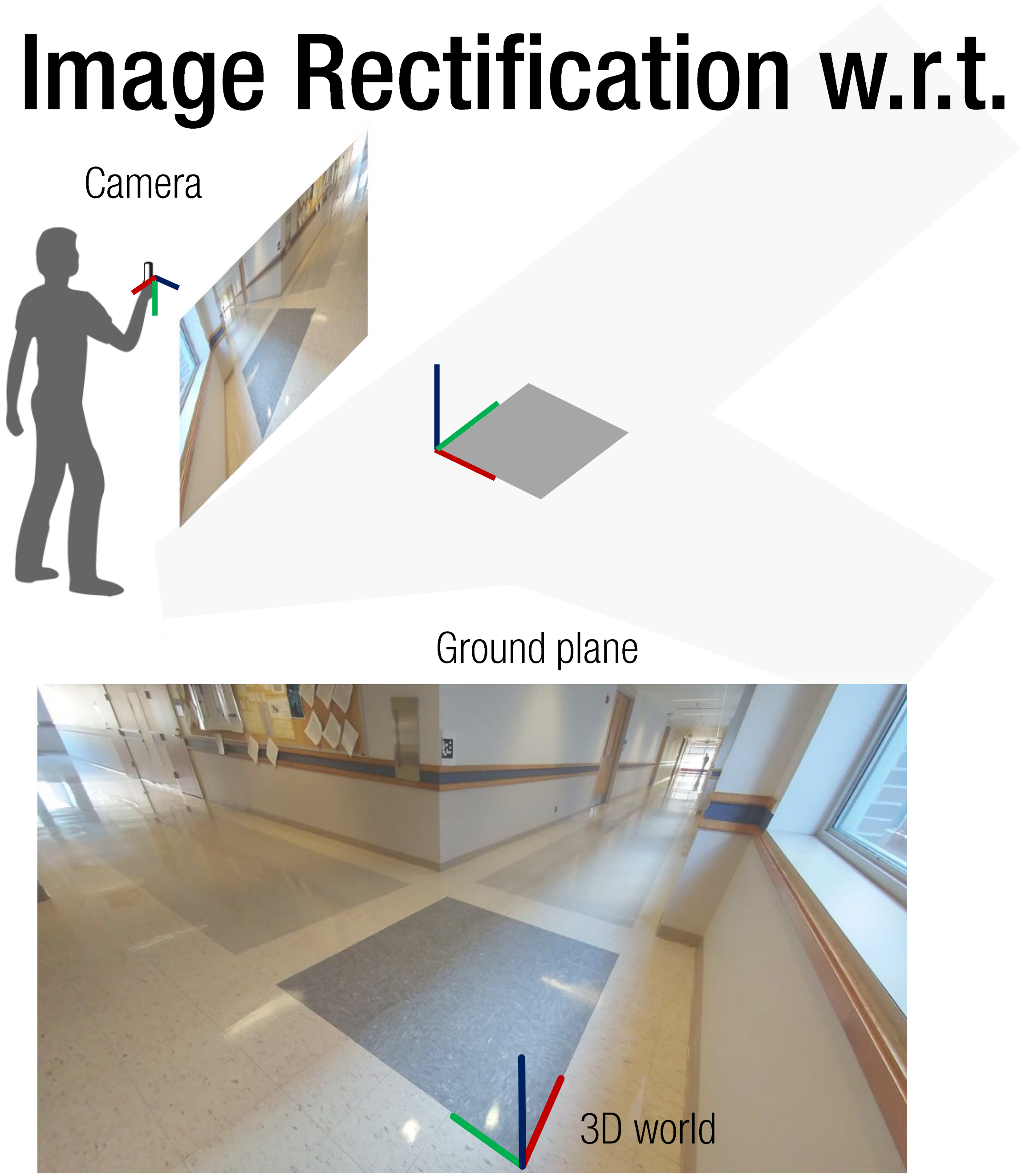


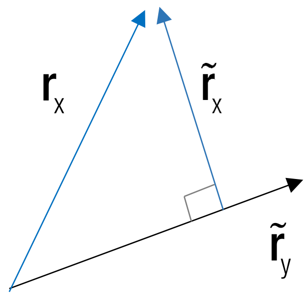
Image Rectification w.r.t. Ground Plane



$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \longrightarrow \tilde{\mathbf{R}} = \begin{bmatrix} & \tilde{\mathbf{r}}_x & \\ 0 & 0 & -1 \\ & \tilde{\mathbf{r}}_z & \end{bmatrix}$$

Image rotation

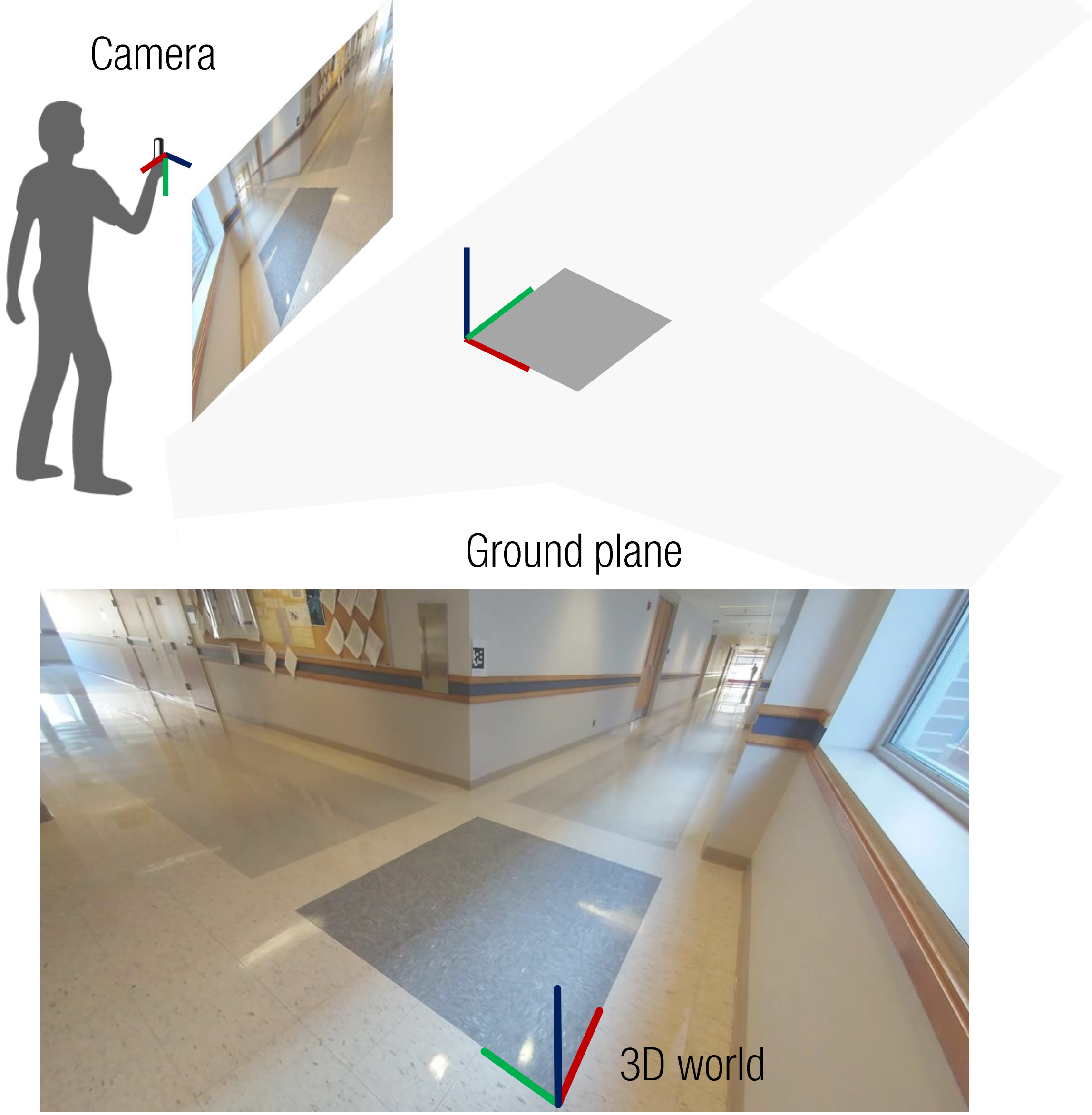
Rectified rotation



$$\tilde{\mathbf{r}}_x = \frac{\mathbf{r}_x - (\mathbf{r}_x \cdot \tilde{\mathbf{r}}_y)\tilde{\mathbf{r}}_y}{\left\| \mathbf{r}_x - (\mathbf{r}_x \cdot \tilde{\mathbf{r}}_y)\tilde{\mathbf{r}}_y \right\|}$$

$$\tilde{\mathbf{r}}_z = \tilde{\mathbf{r}}_x \times \tilde{\mathbf{r}}_y$$

Image Rectification w.r.t. Ground Plane

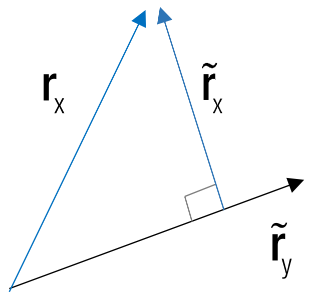


$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \longrightarrow \tilde{\mathbf{R}} = \begin{bmatrix} & \tilde{\mathbf{r}}_x & \\ 0 & 0 & -1 \\ & \tilde{\mathbf{r}}_z & \end{bmatrix}$$

Image rotation

Rectified rotation

$$\tilde{\mathbf{r}}_x = \frac{\mathbf{r}_x - \left(\mathbf{r}_x \cdot \tilde{\mathbf{r}}_y\right)\tilde{\mathbf{r}}_y}{\left\| \mathbf{r}_x - \left(\mathbf{r}_x \cdot \tilde{\mathbf{r}}_y\right)\tilde{\mathbf{r}}_y \right\|}$$



$$\tilde{\mathbf{r}}_z = \tilde{\mathbf{r}}_x \times \tilde{\mathbf{r}}_y$$

$$\lambda \tilde{\mathbf{R}}^{\mathrm{T}} \mathbf{K}^{-1} \tilde{\mathbf{u}} = \mathbf{R}^{\mathrm{T}} \mathbf{K}^{-1} \mathbf{u} \longrightarrow \lambda \tilde{\mathbf{u}} = \mathbf{K} \tilde{\mathbf{R}} \mathbf{R}^{\mathrm{T}} \mathbf{K}^{-1} \mathbf{u}$$

Image Rectification w.r.t. Ground Plane

Image Rectification w.r.t. Ground Plane

```
im = imread('undistorted.png');
f = 1300;
K = [f 0 size(im,2)/2;
     0 f size(im,1)/2;
     0 0 1];
m11 = [2145;2120;1];m12 = [2566;1191;1];m13 = [1804;935;1];m14 = [1050;1320;1];

u = [m11(1:2)';m12(1:2)';m13(1:2)'; m14(1:2)'];
X = [0 0;1 0;1 1;0 1];
X = [X ones(4,1)]; % homogeneous coordinate

H = ComputeHomography(u, X);

denom = norm(inv(K)*H(:,1));
r1 = inv(K)*H(:,1)/denom;  r2 = inv(K)*H(:,2)/denom;  t = inv(K)*H(:,3)/denom;
r3 = Vec2Skew(r1)*r2;
R = [r1 r2 r3];

r2_n = [0 0 -1];
r1_n = (R(1,:) - (R(1,:)*r2_n')*r2_n);
r3_n = Vec2Skew(r1_n)*r2_n';

R_n = [r1_n; r2_n; r3_n'];
H_new = K * R_n * inv(R) * inv(K);

im_warped = ImageWarping(im, H_new);
```

RectificationFromHomography.m

Image Rectification w.r.t. Ground Plane

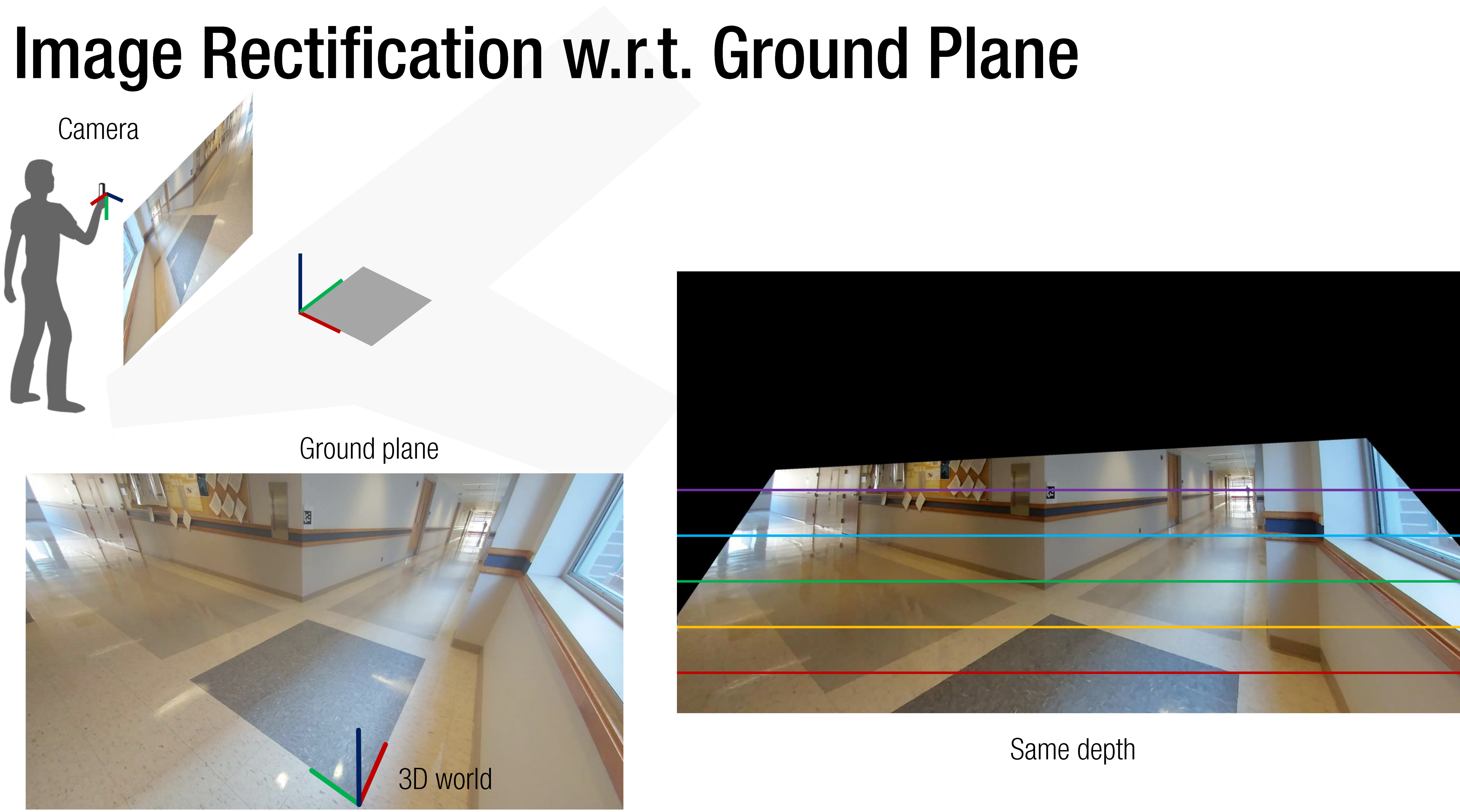
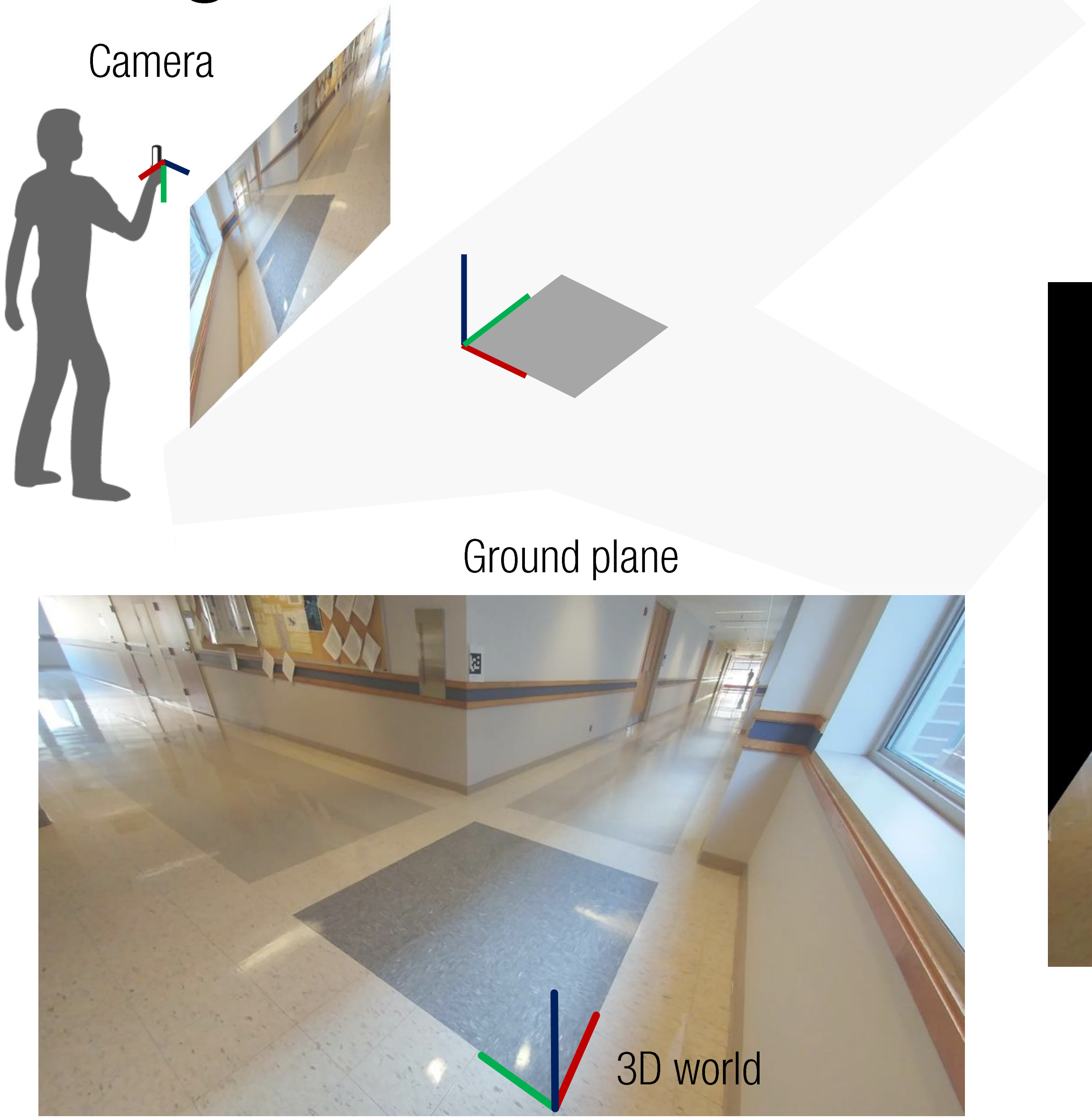
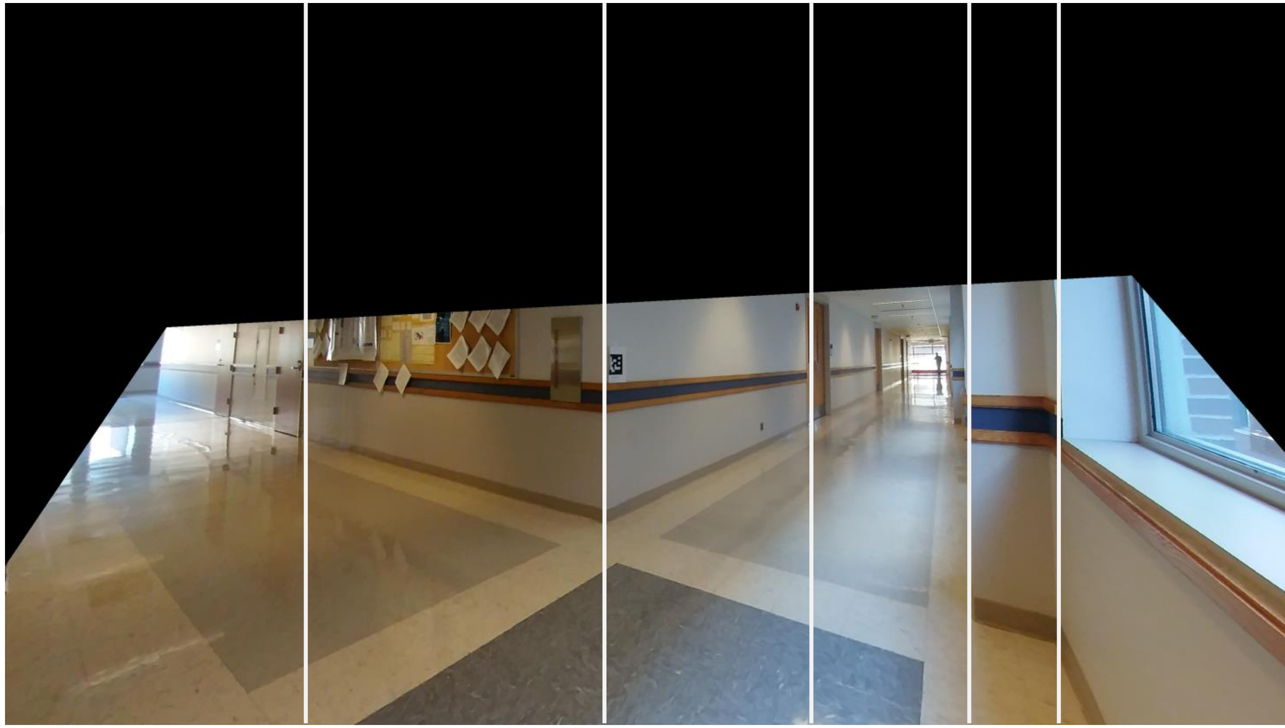


Image Rectification w.r.t. Ground Plane

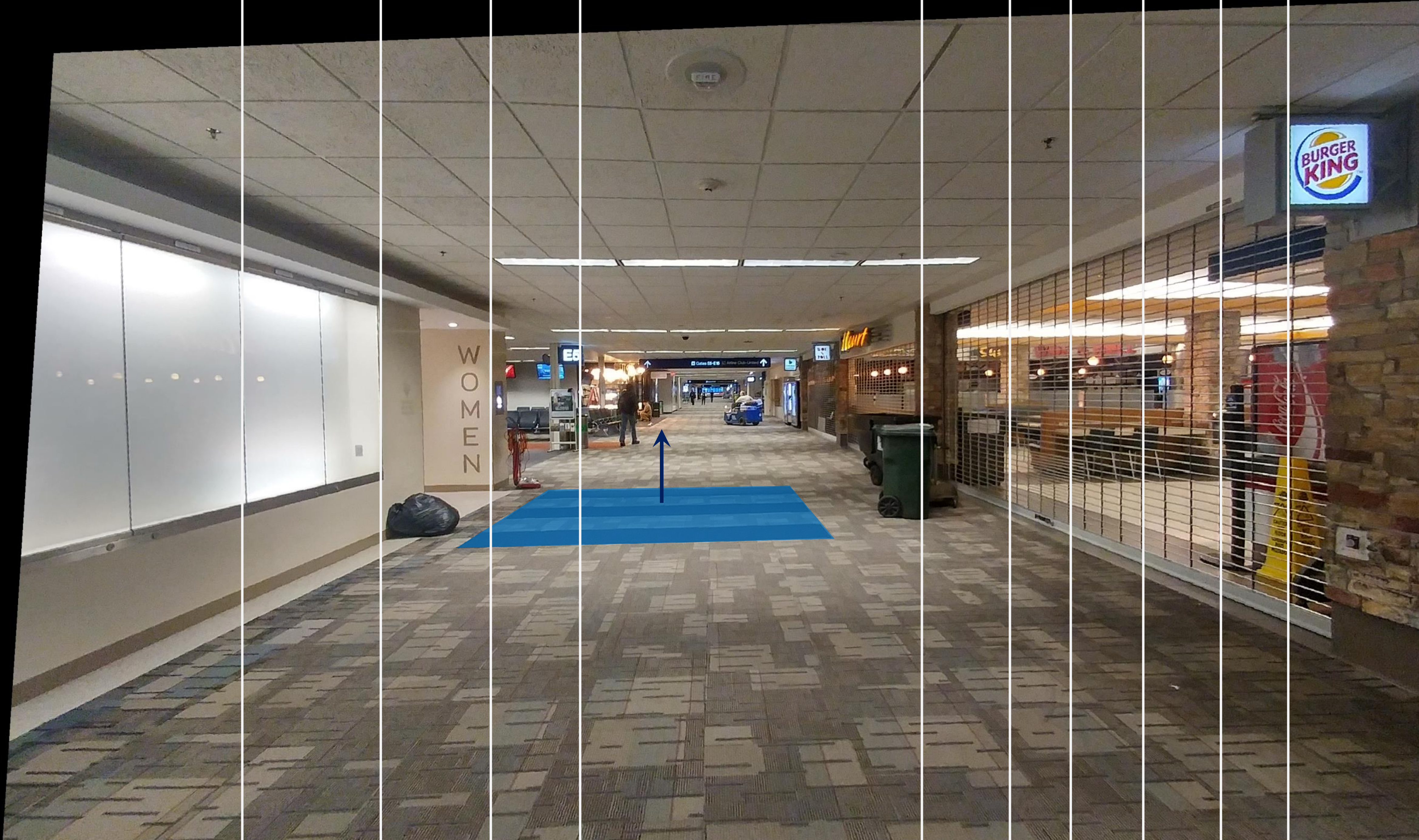





BURGER KING
WOMEN
E5
Gates E6-E16

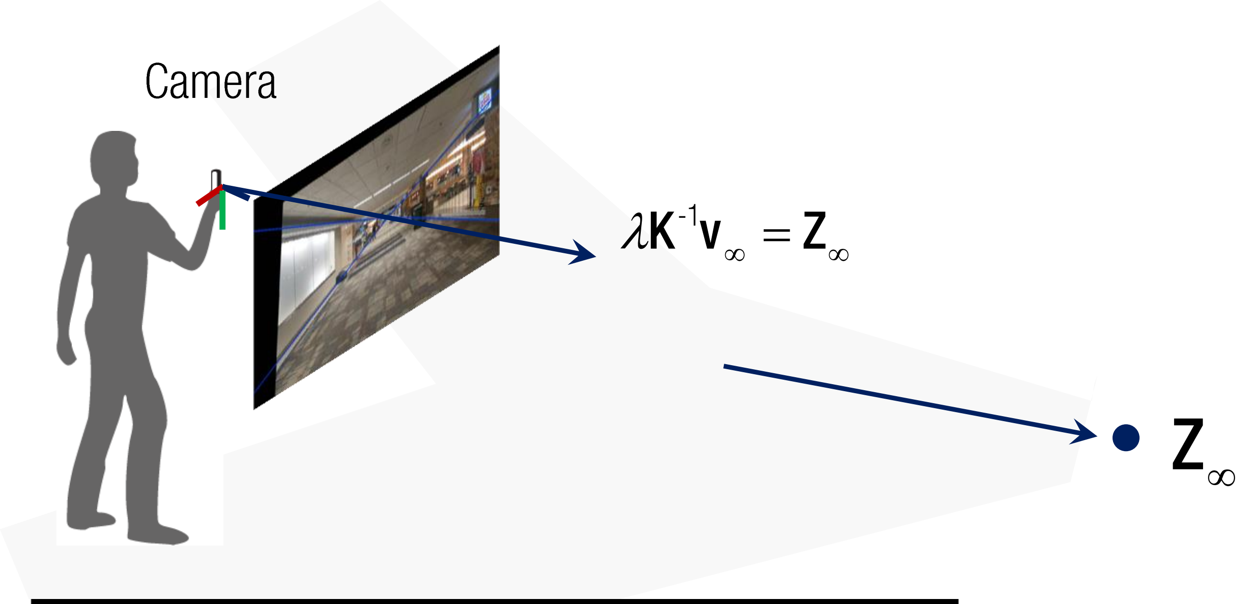

BURGER KING
WOMEN
E5

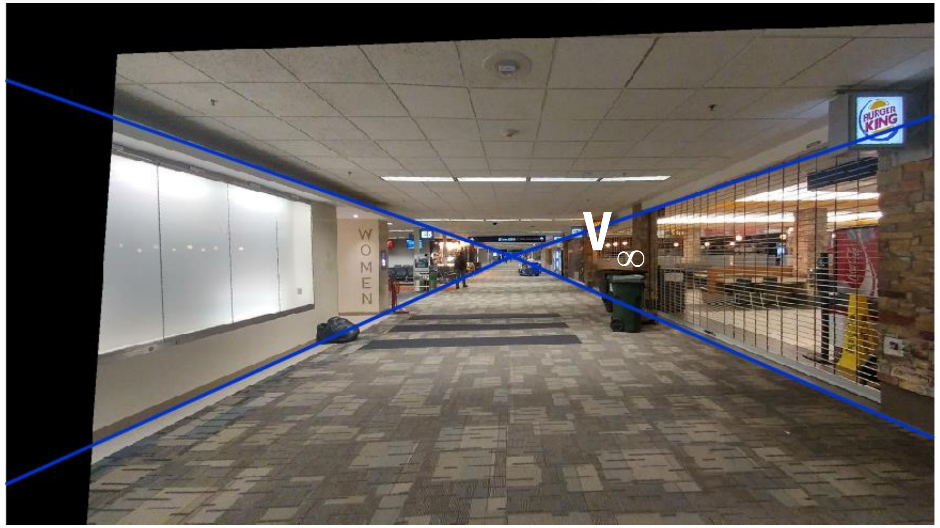

BURGER KING
WOMEN
E5


WOMEN
BURGER KING


Vanishing point
WOMEN
BURGER KING

Vanishing Point



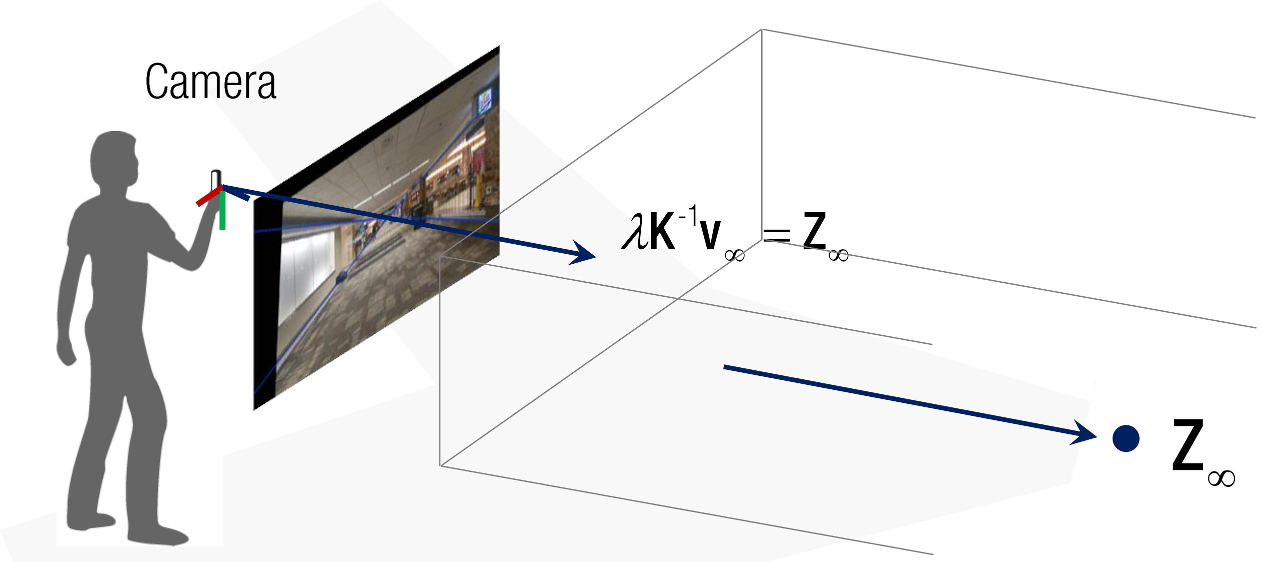


Vanishing point projection:

$$\lambda \mathbf{v}_{\infty} = \mathbf{K}\mathbf{Z}_{\infty}$$

$$\longrightarrow \quad \lambda \mathbf{K}^{-1}\mathbf{v}_{\infty} = \mathbf{Z}_{\infty}$$

Box Representation

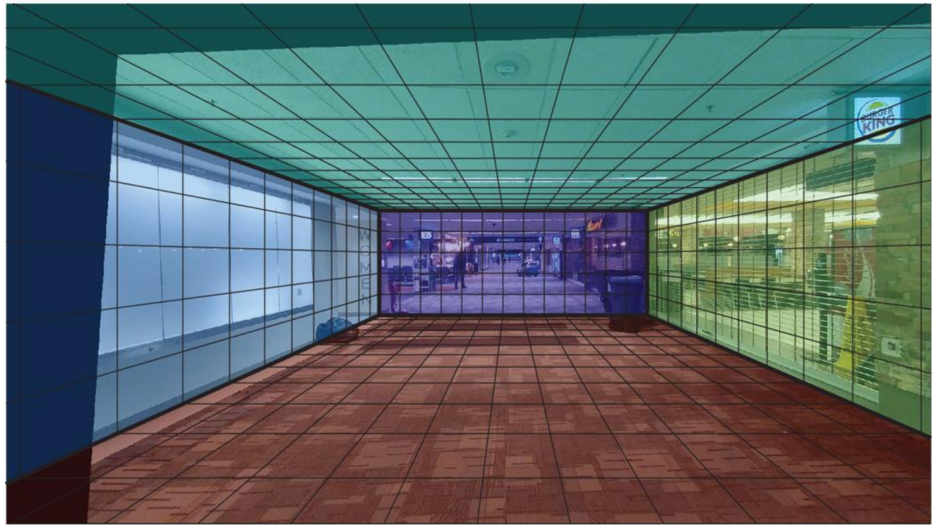


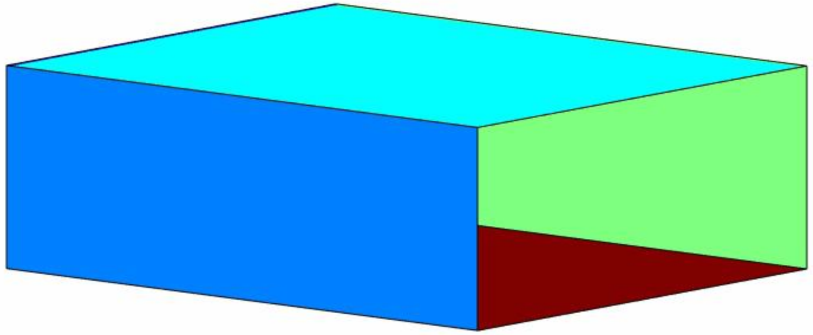
Vanishing point projection:

$$\lambda \mathbf{v}_\infty = \mathbf{K}\mathbf{Z}_\infty$$

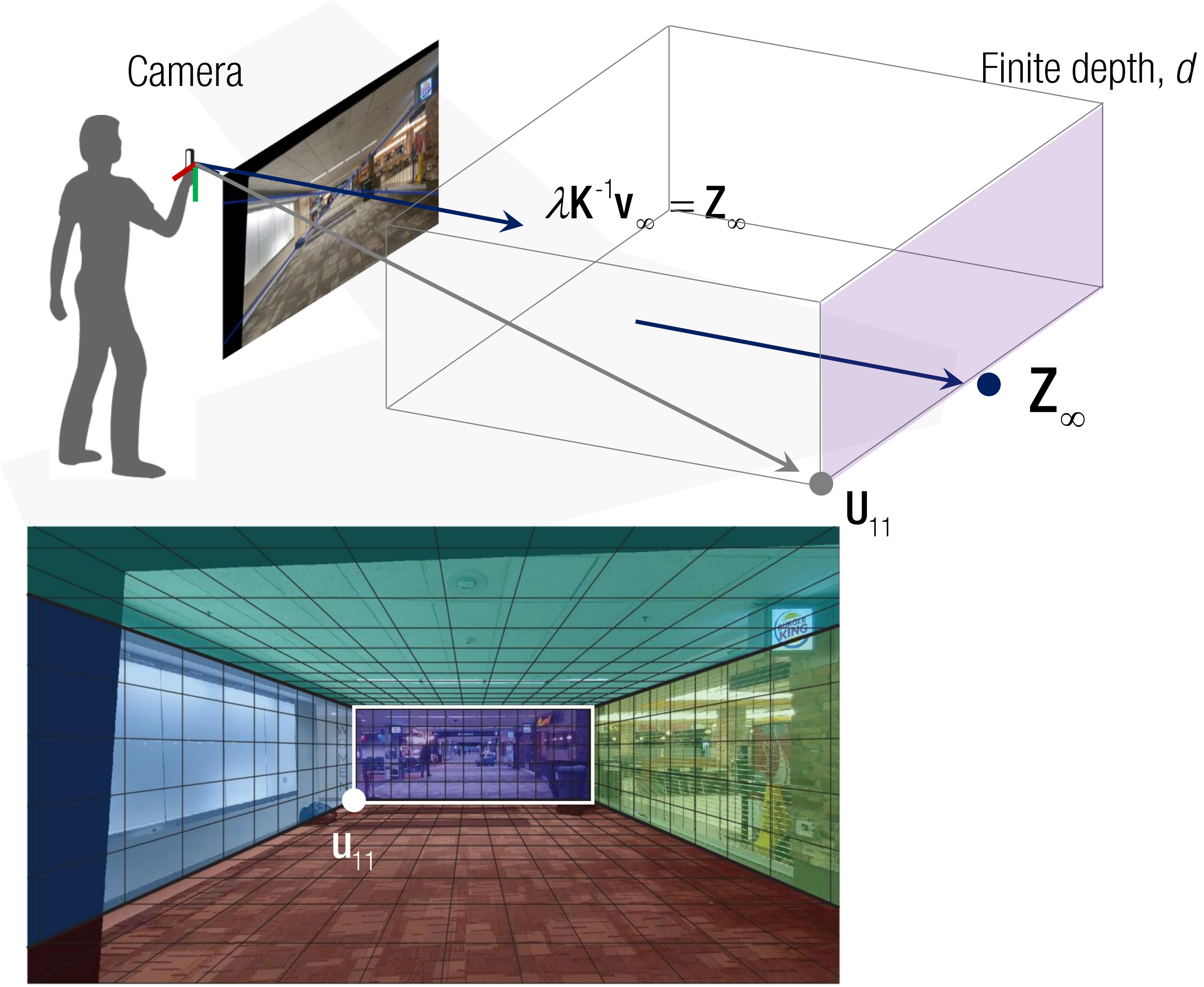
$$\longrightarrow \quad \lambda \mathbf{K}^{-1}\mathbf{v}_\infty = \mathbf{Z}_\infty$$

Define the direction of the box





Box Representation



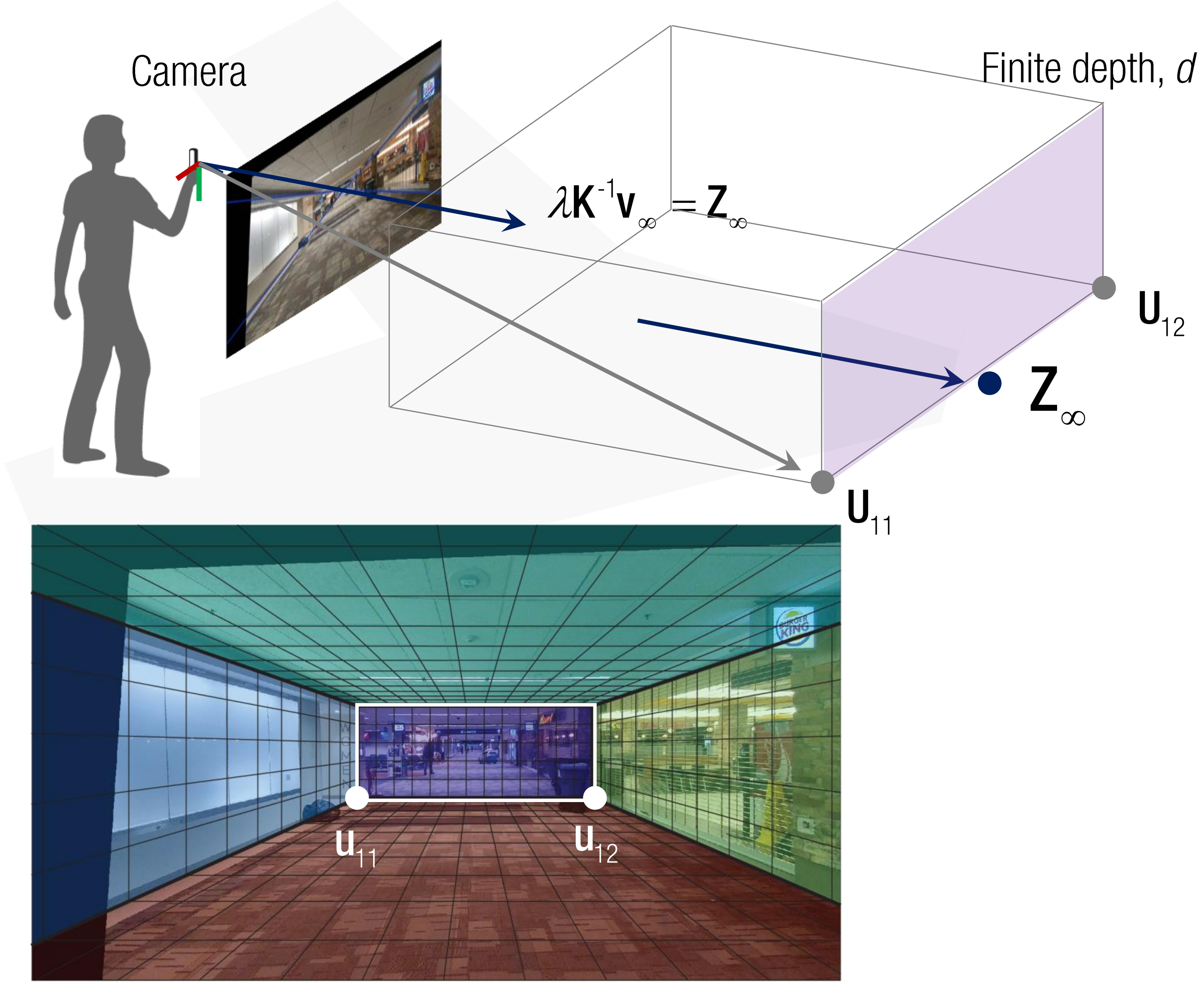
Vanishing point projection:

$$\lambda \mathbf{v}_\infty = \mathbf{K}\mathbf{Z}_\infty$$

$$\longrightarrow \lambda \mathbf{K}^{-1}\mathbf{v}_\infty = \mathbf{Z}_\infty$$

$$\mathbf{U}_{11} = d\mathbf{K}^{-1}\mathbf{u}_{11}$$

Box Representation



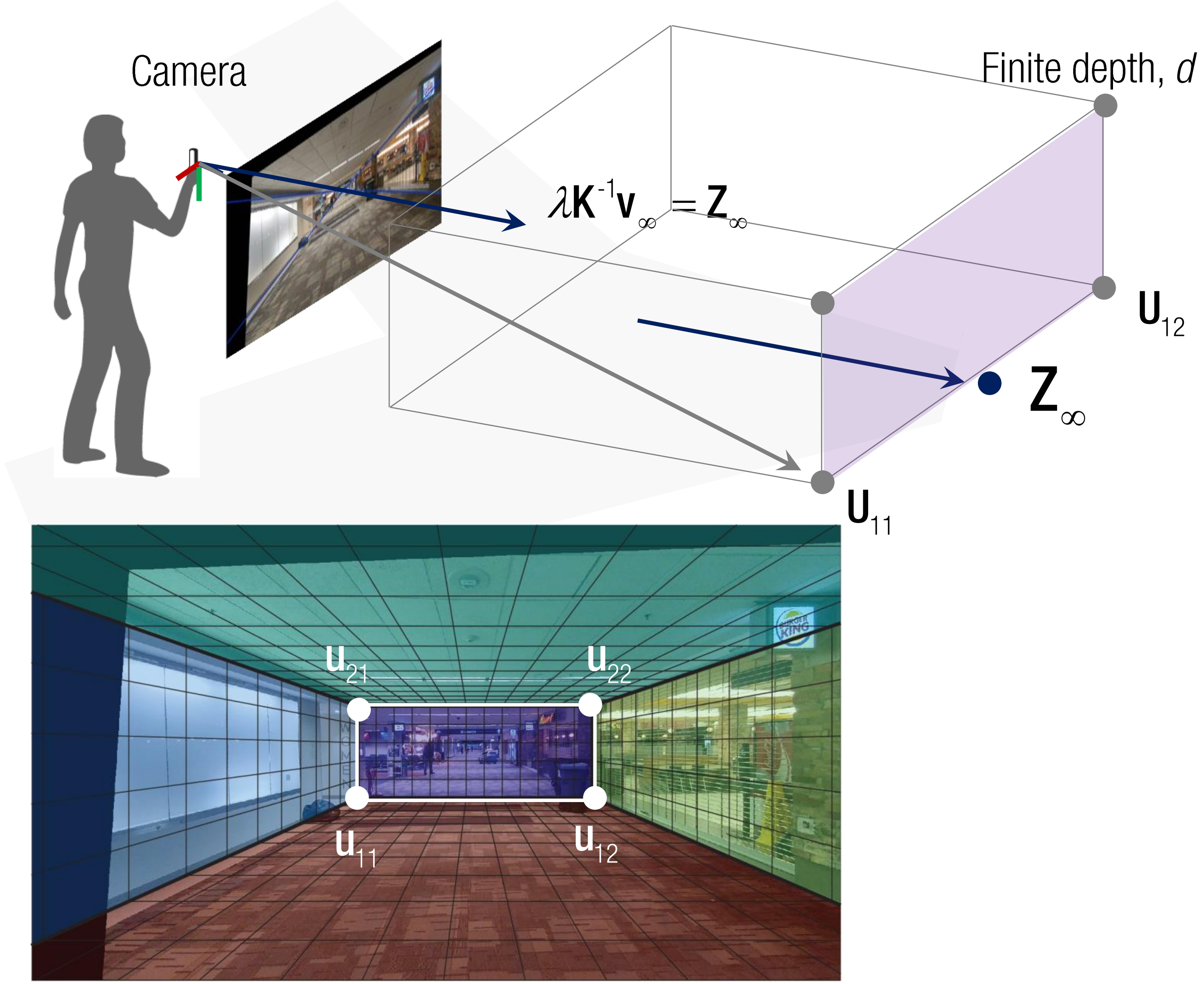
Vanishing point projection:

$$\lambda \mathbf{v}_{\infty} = \mathbf{K}\mathbf{Z}_{\infty}$$

$$\longrightarrow \quad \lambda \mathbf{K}^{-1}\mathbf{v}_{\infty} = \mathbf{Z}_{\infty}$$

$\mathbf{U}_{11} = d\mathbf{K}^{-1}\mathbf{u}_{11}$ $\qquad$ $\mathbf{U}_{12} = d\mathbf{K}^{-1}\mathbf{u}_{12}$: Same x coord.

Box Representation



Vanishing point projection:

$$\lambda \mathbf{v}_\infty = \mathbf{K}\mathbf{Z}_\infty$$

$$\longrightarrow \quad \lambda \mathbf{K}^{-1}\mathbf{v}_\infty = \mathbf{Z}_\infty$$

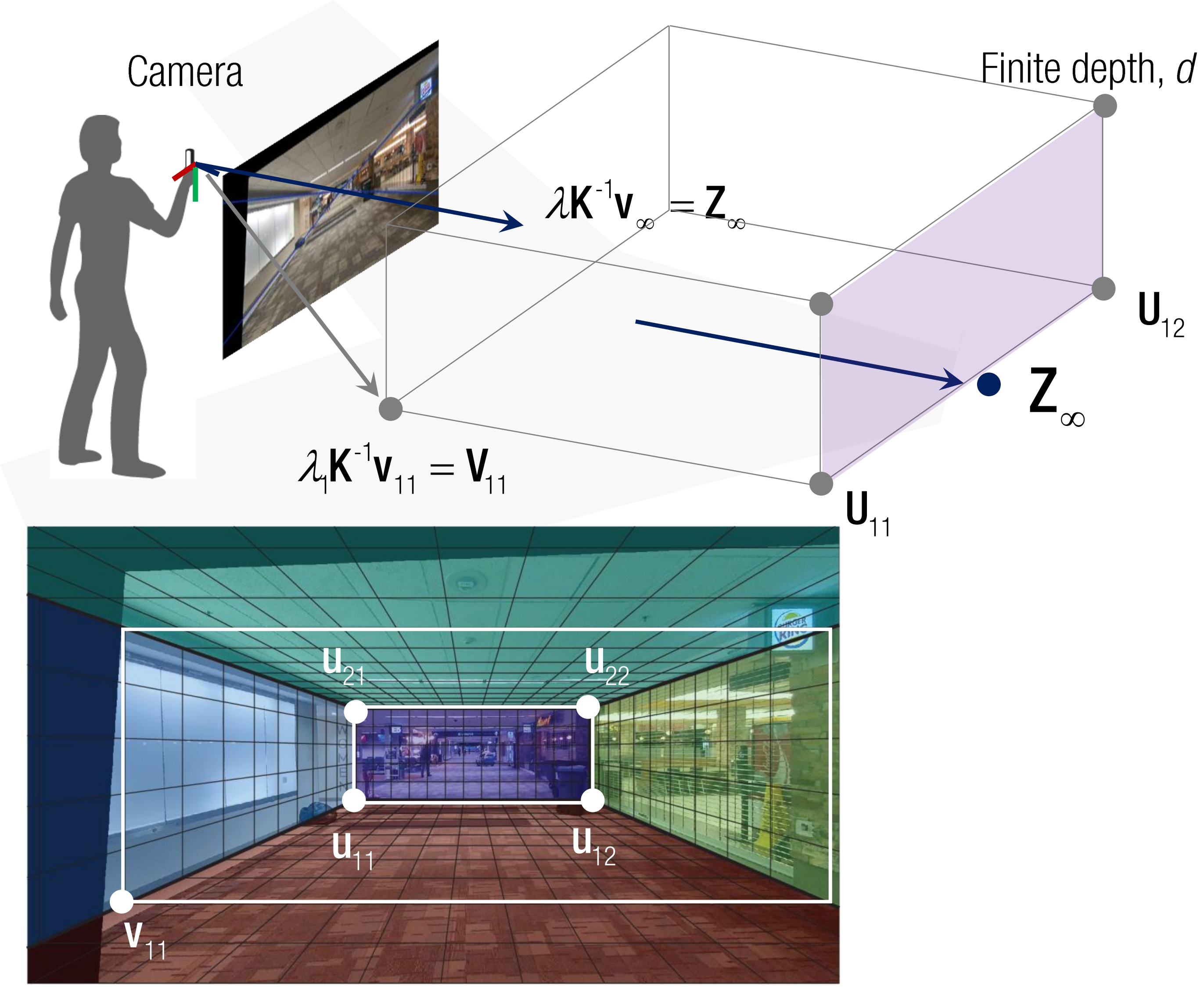
$$\mathbf{U}_{11} = d\mathbf{K}^{-1}\mathbf{u}_{11} \qquad \mathbf{U}_{12} = d\mathbf{K}^{-1}\mathbf{u}_{12}$$

: Same x coord.

$$\mathbf{U}_{21} = d\mathbf{K}^{-1}\mathbf{u}_{21} \qquad \mathbf{U}_{22} = d\mathbf{K}^{-1}\mathbf{u}_{22}$$

Same y coord.

Box Representation



Vanishing point projection:

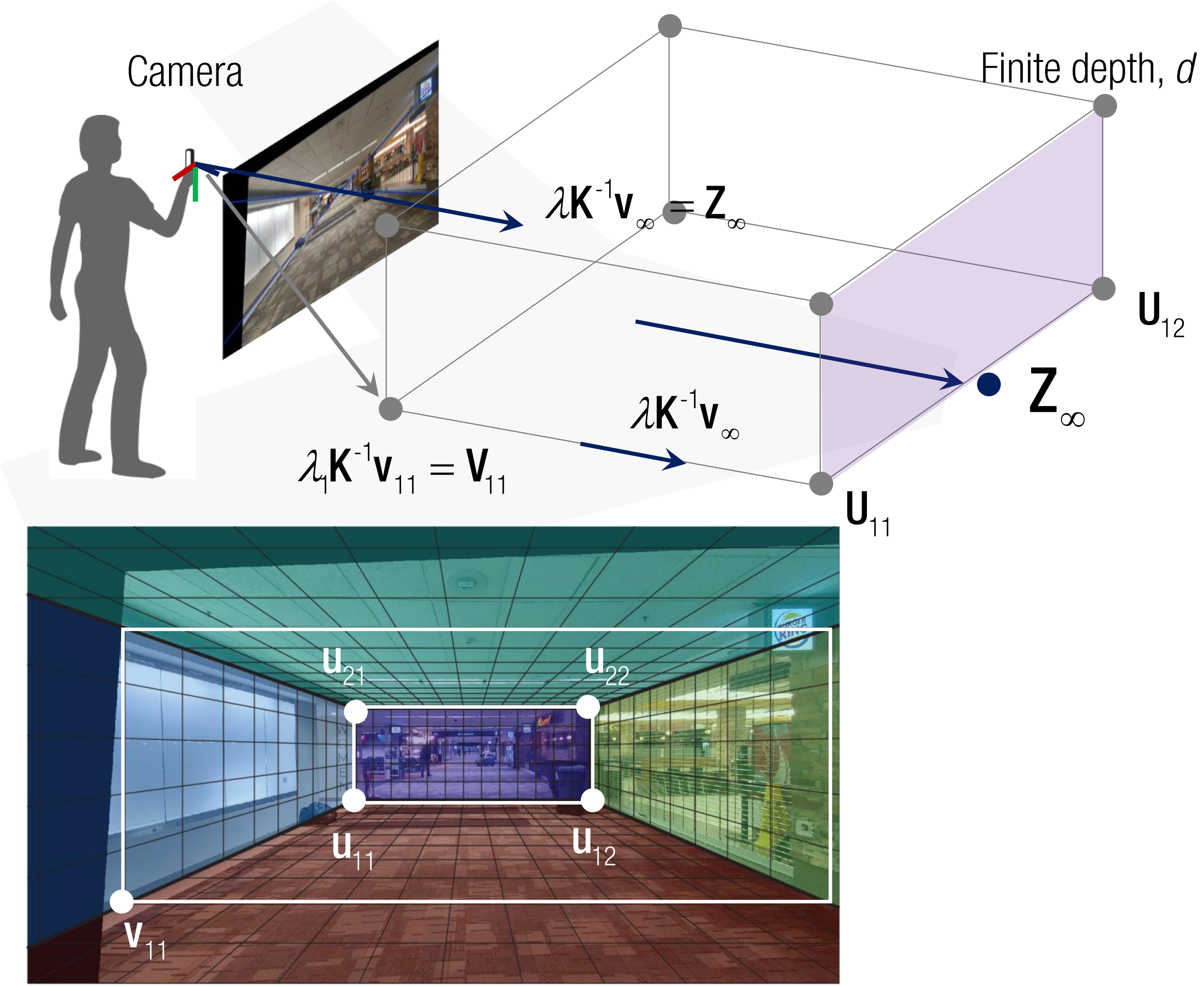
$$\lambda \mathbf{v}_\infty = \mathbf{K}\mathbf{Z}_\infty$$

$$\longrightarrow \quad \lambda \mathbf{K}^{-1}\mathbf{v}_\infty = \mathbf{Z}_\infty$$

Depth of frontal surface?

$$\underline{\lambda_1} \mathbf{K}^{-1}\mathbf{v}_{11} = \mathbf{V}_{11}$$

Box Representation



Vanishing point projection:

$$\lambda \mathbf{v}_\infty = \mathbf{K}\mathbf{Z}_\infty$$

$$\longrightarrow \quad \lambda \mathbf{K}^{-1}\mathbf{v}_\infty = \mathbf{Z}_\infty$$

Depth of frontal surface?

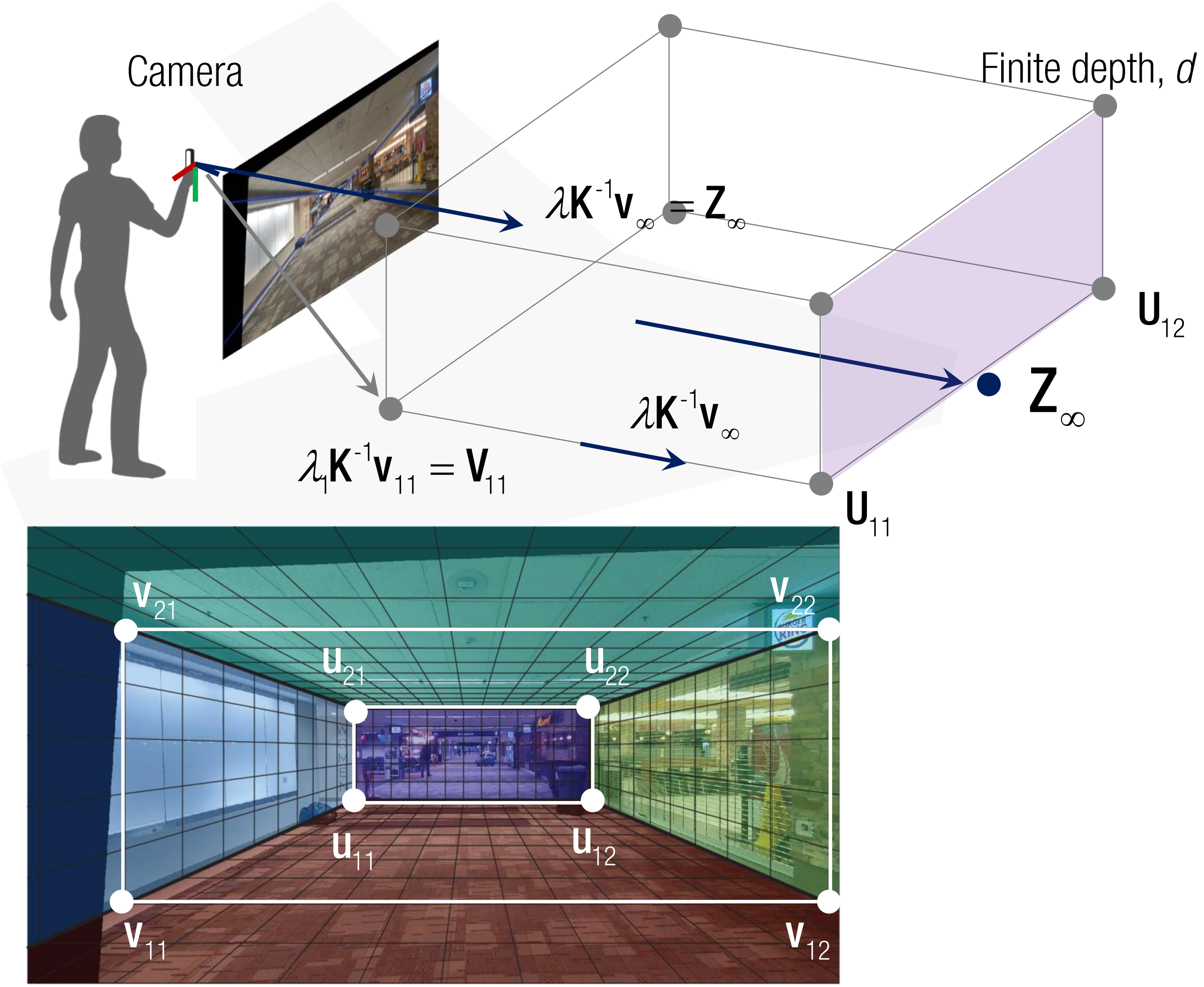
$$\underline{\lambda_1}\mathbf{K}^{-1}\mathbf{v}_{11} = \mathbf{V}_{11}$$

Line between $\mathbf{U}_{11}$ and $\mathbf{V}_{11}$ is parallel to the vanishing point direction.

$$\lambda_1 \mathbf{K}^{-1}\mathbf{v}_{11} + \lambda \mathbf{K}^{-1}\mathbf{v}_\infty = \mathbf{U}_{11} = d\mathbf{K}^{-1}\mathbf{u}_{11}$$

HW: express λ_1 using d.

Box Representation



Vanishing point projection:

$$\lambda \mathbf{v}_\infty = \mathbf{K}\mathbf{Z}_\infty$$

$$\longrightarrow \quad \lambda \mathbf{K}^{-1}\mathbf{v}_\infty = \mathbf{Z}_\infty$$

Depth of frontal surface?

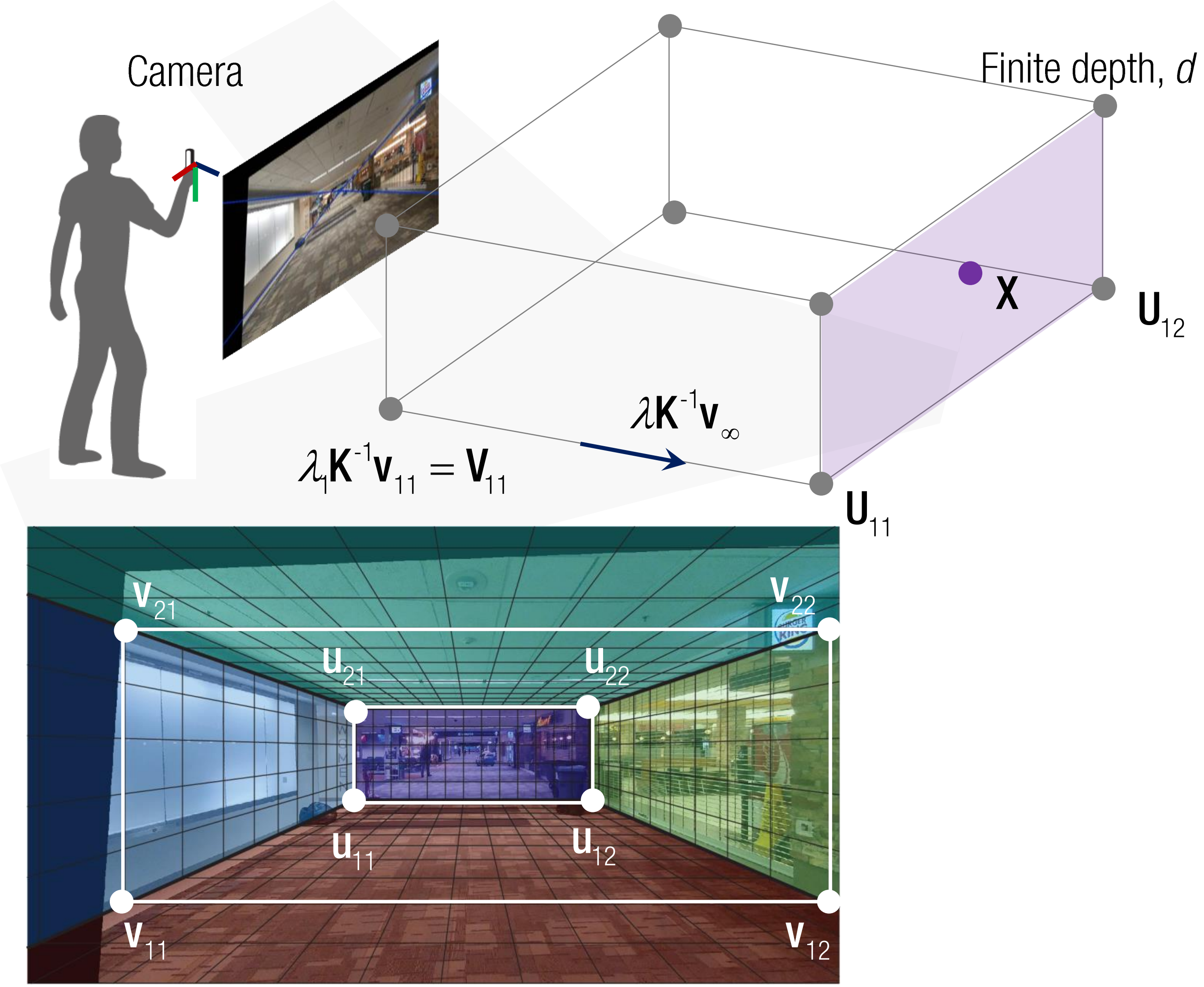
$$\underline{\lambda_1}\mathbf{K}^{-1}\mathbf{v}_{11} = \mathbf{V}_{11}$$

Line between $\mathbf{U}_{11}$ and $\mathbf{V}_{11}$ is parallel to the vanishing point direction.

$$\lambda_1 \mathbf{K}^{-1}\mathbf{v}_{11} + \lambda \mathbf{K}^{-1}\mathbf{v}_\infty = \mathbf{U}_{11} = d\mathbf{K}^{-1}\mathbf{u}_{11}$$

HW: express λ_1 using d.

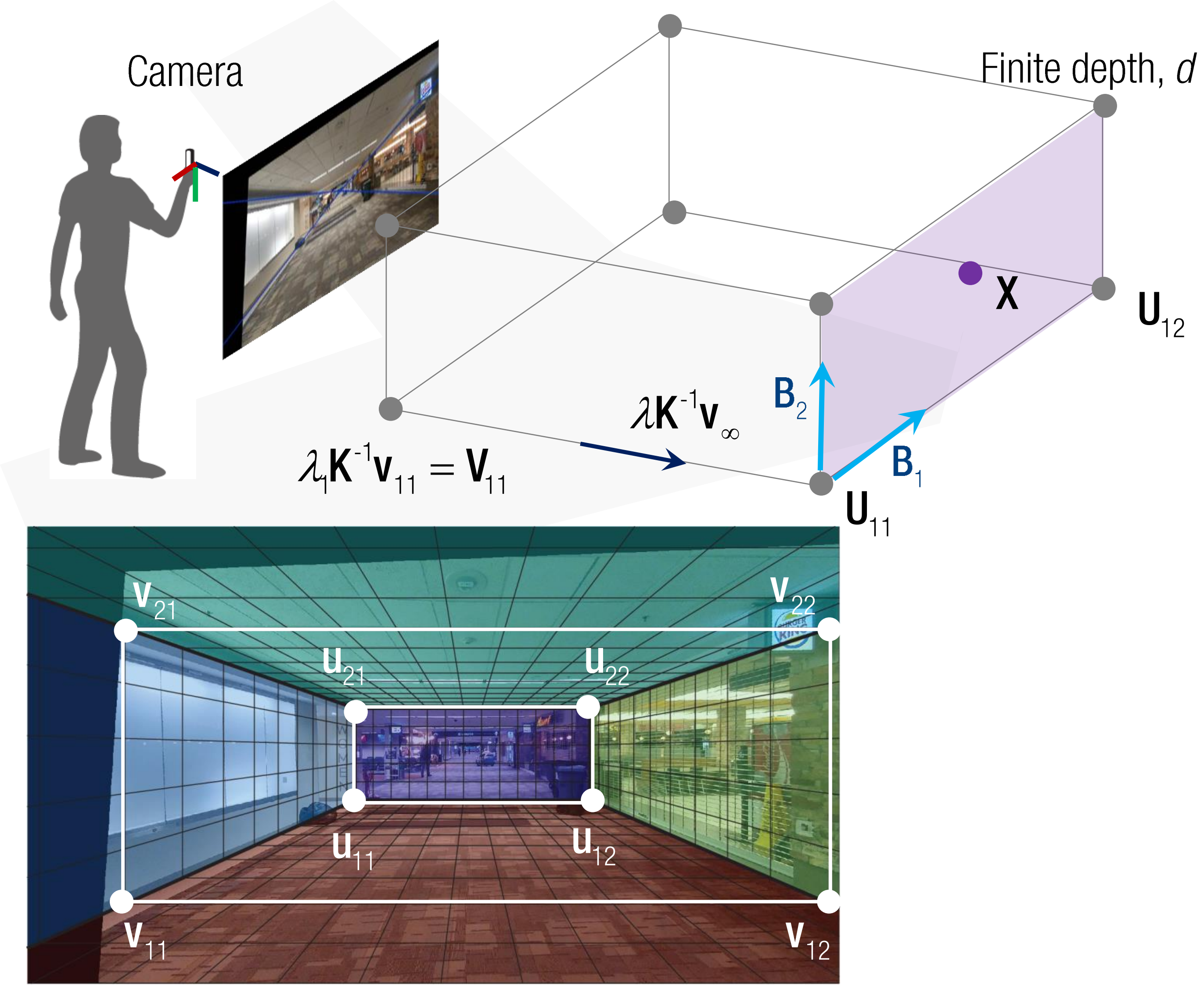
Box Representation



Point in a plane:

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Box Representation



Point in a plane:

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{U}_{11} + \mu_1 \mathbf{B}_1 + \mu_2 \mathbf{B}_2$$

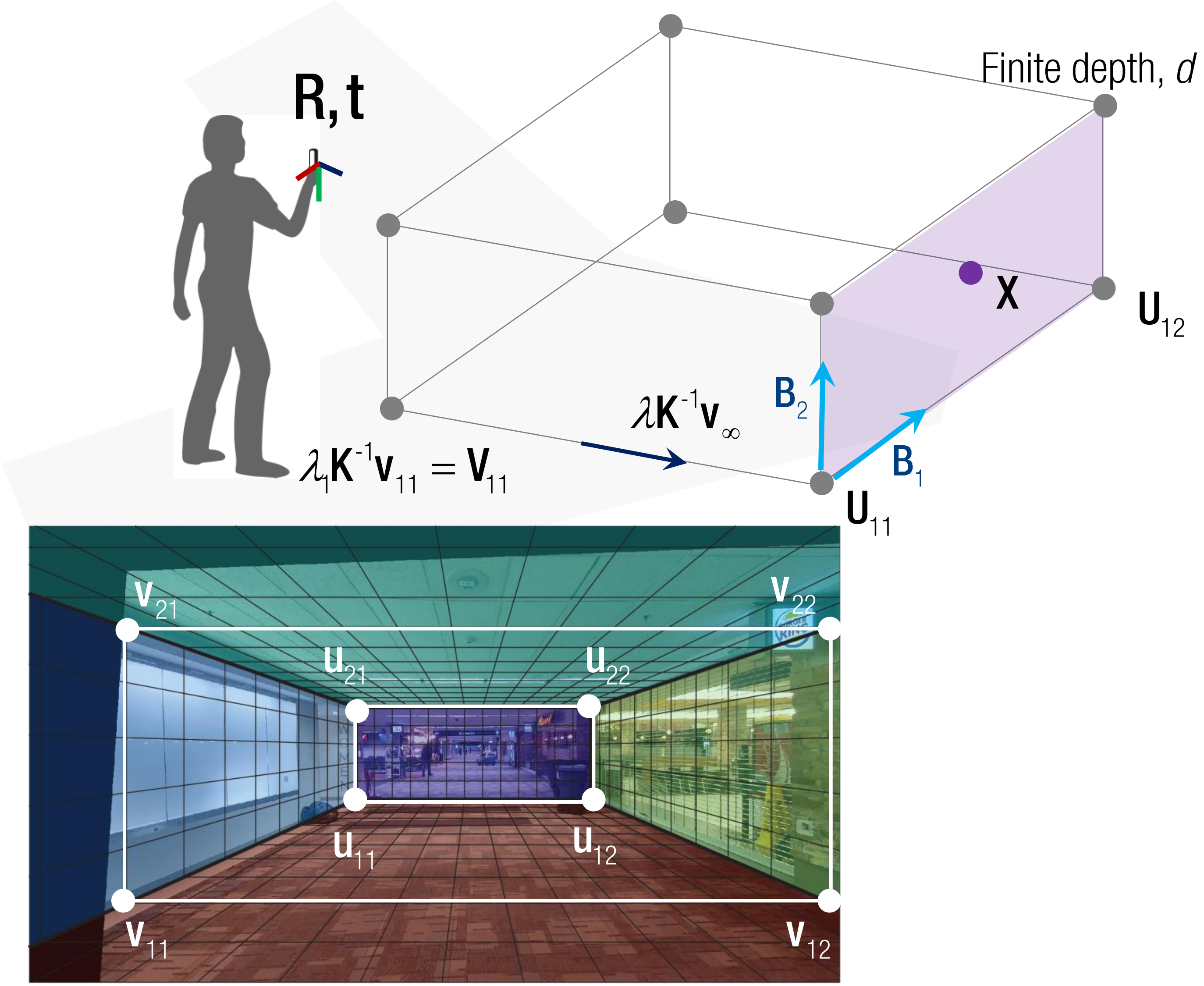
$$= \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 1 \end{bmatrix}$$

2DOF

Homography mapping from 3D plane to image:

$$\lambda \mathbf{u} = \mathbf{K} \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 1 \end{bmatrix}$$

Box Representation

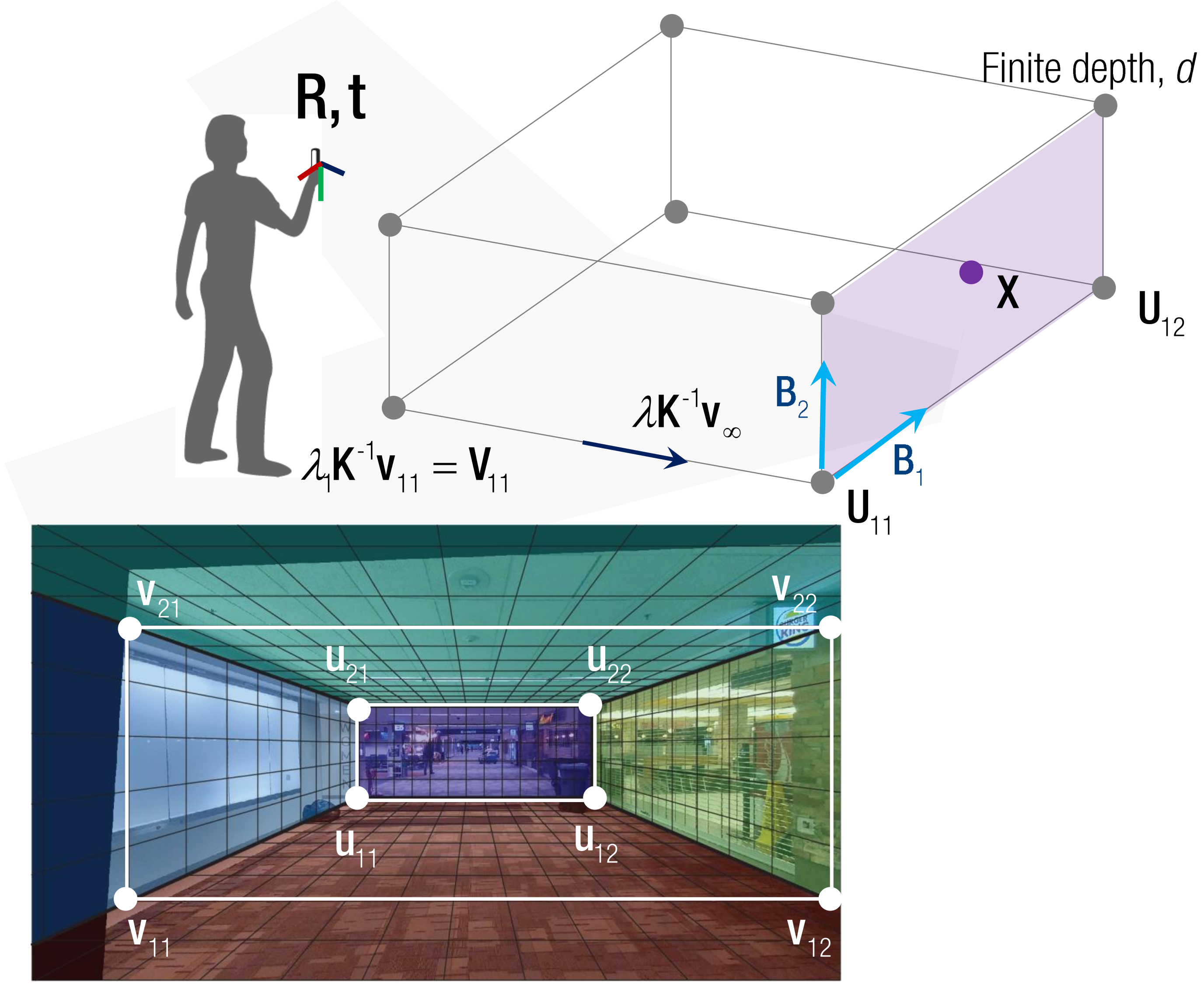


Homography mapping from 3D plane to image:

$$\lambda \mathbf{u} = \mathbf{K}\begin{bmatrix}\mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c}\end{bmatrix}\begin{bmatrix}\mu_1 \\ \mu_2 \\ 1\end{bmatrix} = \mathbf{H}\begin{bmatrix}\mu_1 \\ \mu_2 \\ 1\end{bmatrix}$$

Texture Mapping

Homography



Homography mapping from 3D plane to image:

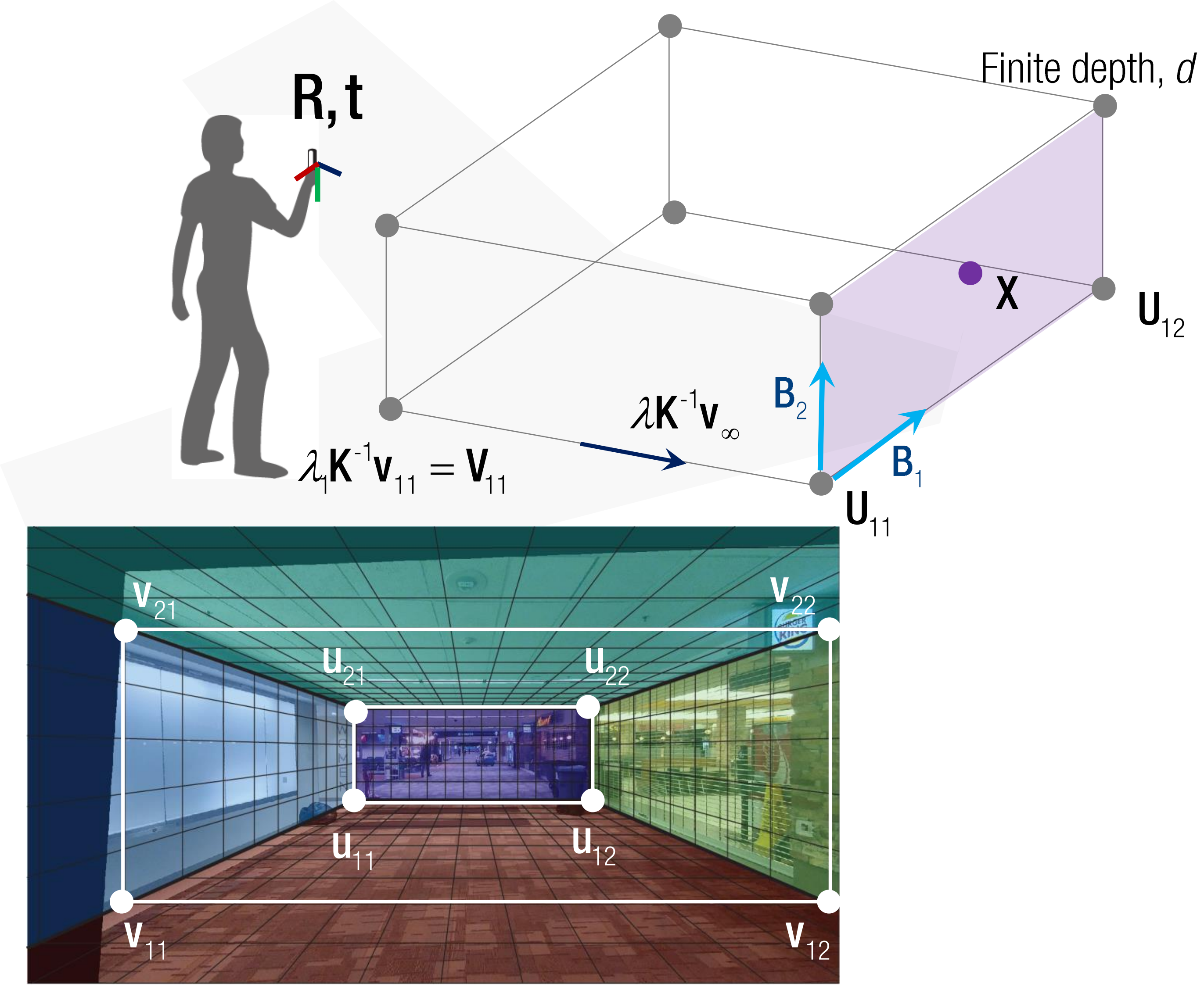
$$\lambda \mathbf{u} = \mathbf{K}\begin{bmatrix}\mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c}\end{bmatrix}\begin{bmatrix}\mu_1 \\ \mu_2 \\ 1\end{bmatrix} = \mathbf{H}\begin{bmatrix}\mu_1 \\ \mu_2 \\ 1\end{bmatrix}$$

Homography mapping from 3D plane to target image:

$$\lambda \tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix}X \\ Y \\ Z \\ 1\end{bmatrix} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix}\begin{bmatrix}\mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c}\end{bmatrix}\begin{bmatrix}\mu_1 \\ \mu_2 \\ 1\end{bmatrix} \\ 1\end{bmatrix}$$

$$= \mathbf{K}\begin{bmatrix}\mathbf{R}\mathbf{B}_1 & \mathbf{R}\mathbf{B}_2 & \mathbf{R}\mathbf{c}+\mathbf{t}\end{bmatrix}\begin{bmatrix}\mu_1 \\ \mu_2 \\ 1\end{bmatrix} = \tilde{\mathbf{H}}\begin{bmatrix}\mu_1 \\ \mu_2 \\ 1\end{bmatrix}$$

Homography



Homography mapping from 3D plane to image:

$$\lambda \mathbf{u} = \mathbf{K}\begin{bmatrix}\mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c}\end{bmatrix}\begin{bmatrix}\mu_1\\ \mu_2\\ 1\end{bmatrix} = \mathbf{H}\begin{bmatrix}\mu_1\\ \mu_2\\ 1\end{bmatrix}$$

Homography mapping from 3D plane to target image:

$$\lambda \tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix}X\\ Y\\ Z\\ 1\end{bmatrix} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix}\begin{bmatrix}\mathbf{B}_1 & \mathbf{B}_2 & \mathbf{c}\end{bmatrix}\begin{bmatrix}\mu_1\\ \mu_2\\ 1\end{bmatrix}\\ 1\end{bmatrix}$$

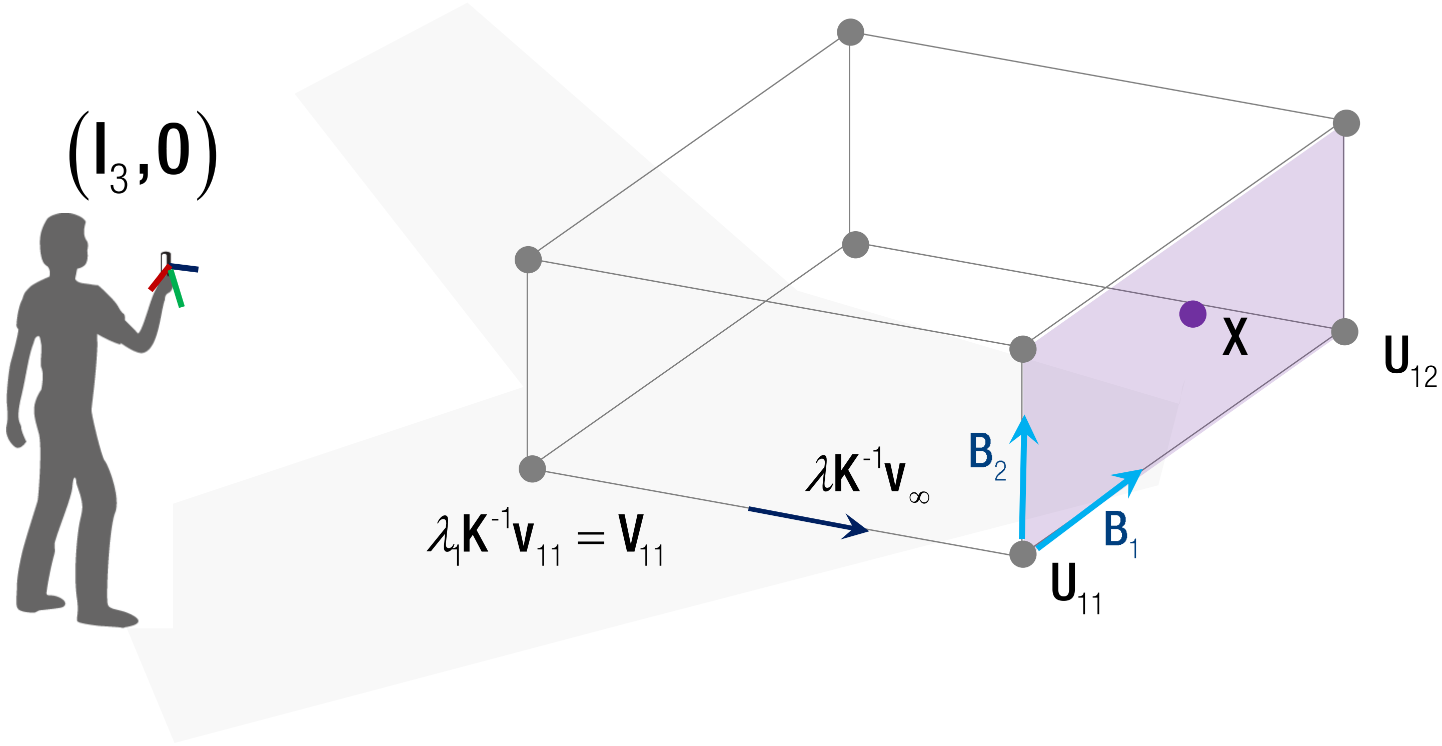
$$= \mathbf{K}\begin{bmatrix}\mathbf{R}\mathbf{B}_1 & \mathbf{R}\mathbf{B}_2 & \mathbf{R}\mathbf{c}+\mathbf{t}\end{bmatrix}\begin{bmatrix}\mu_1\\ \mu_2\\ 1\end{bmatrix} = \tilde{\mathbf{H}}\begin{bmatrix}\mu_1\\ \mu_2\\ 1\end{bmatrix}$$

$$\rightarrow \lambda \tilde{\mathbf{H}}^{-1}\tilde{\mathbf{u}} = \begin{bmatrix}\mu_1\\ \mu_2\\ 1\end{bmatrix} = \lambda_1 \mathbf{H}^{-1}\mathbf{u} \qquad \rightarrow \lambda \tilde{\mathbf{u}} = \tilde{\mathbf{H}}\mathbf{H}^{-1}\mathbf{u}$$

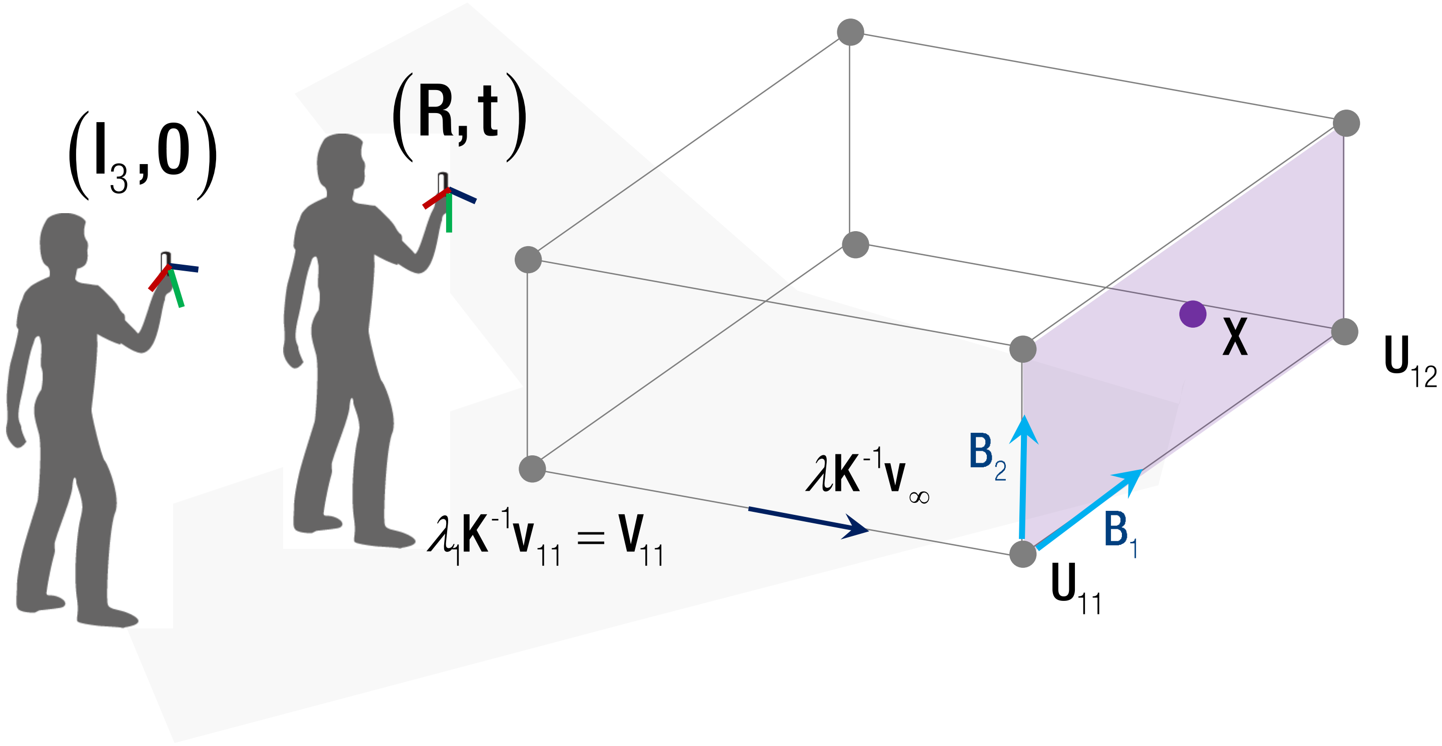
HW #3 Tour into your photo

Spatial Rotation

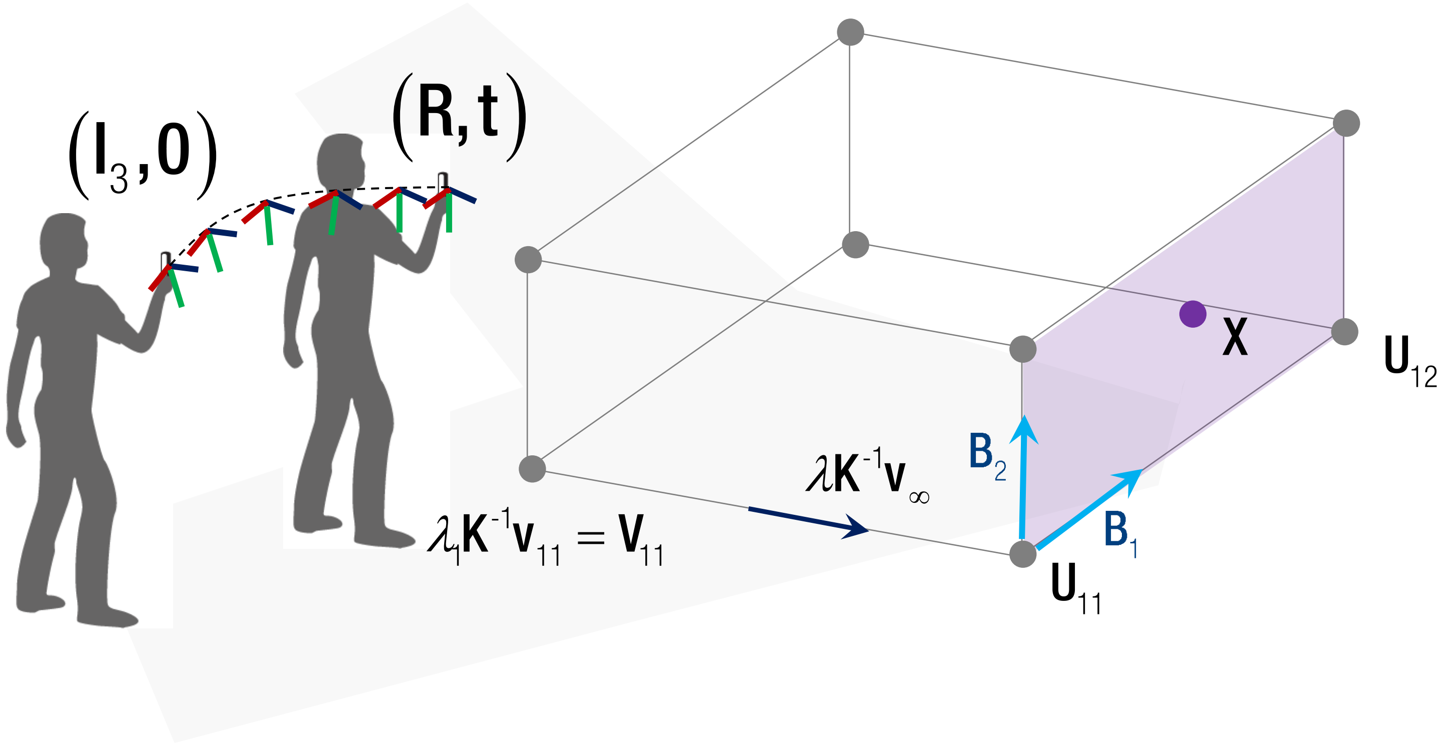
Interpolation of Transformation



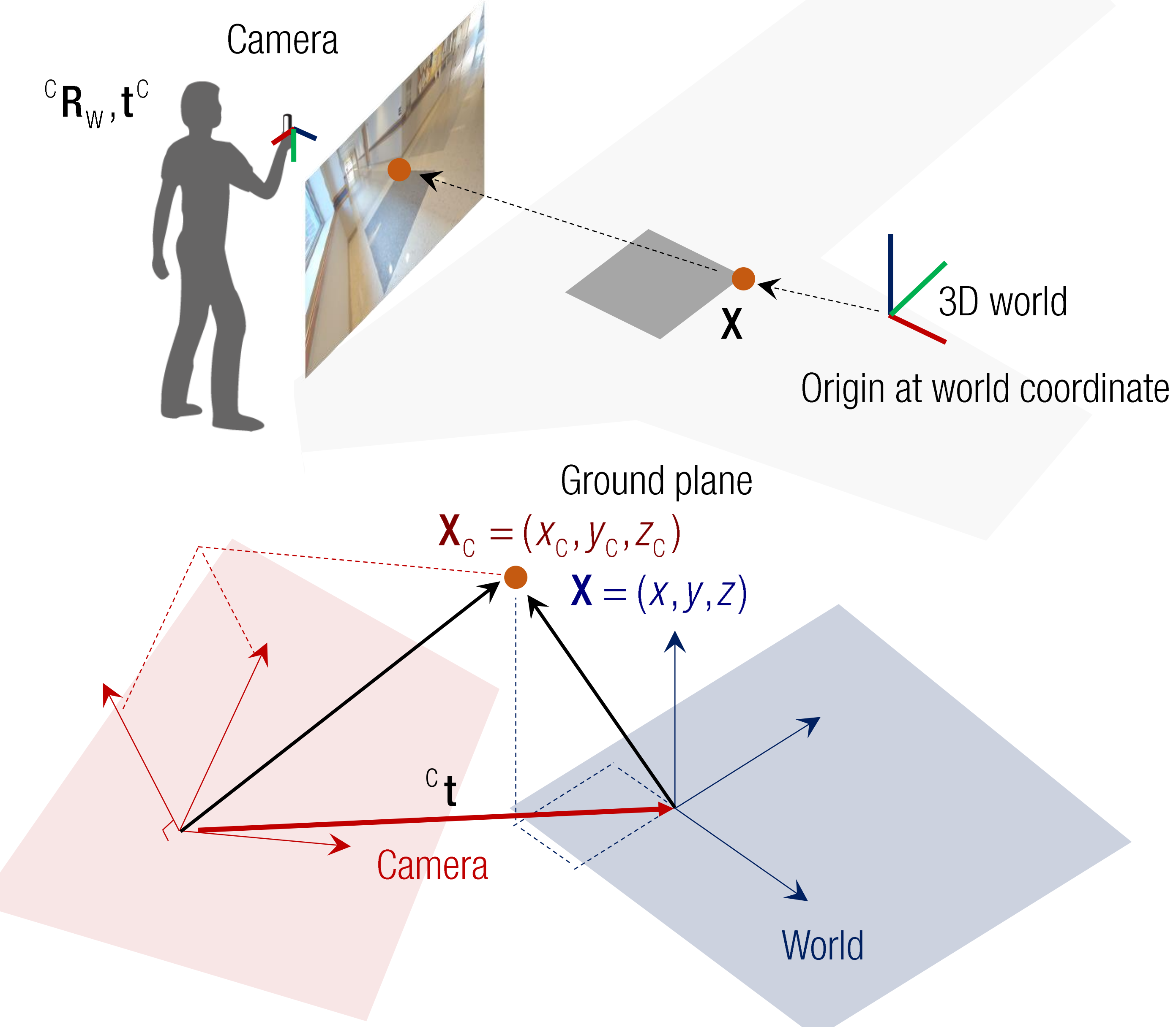
Interpolation of Transformation



Interpolation of Transformation



Recall: Rotate and then, Translate

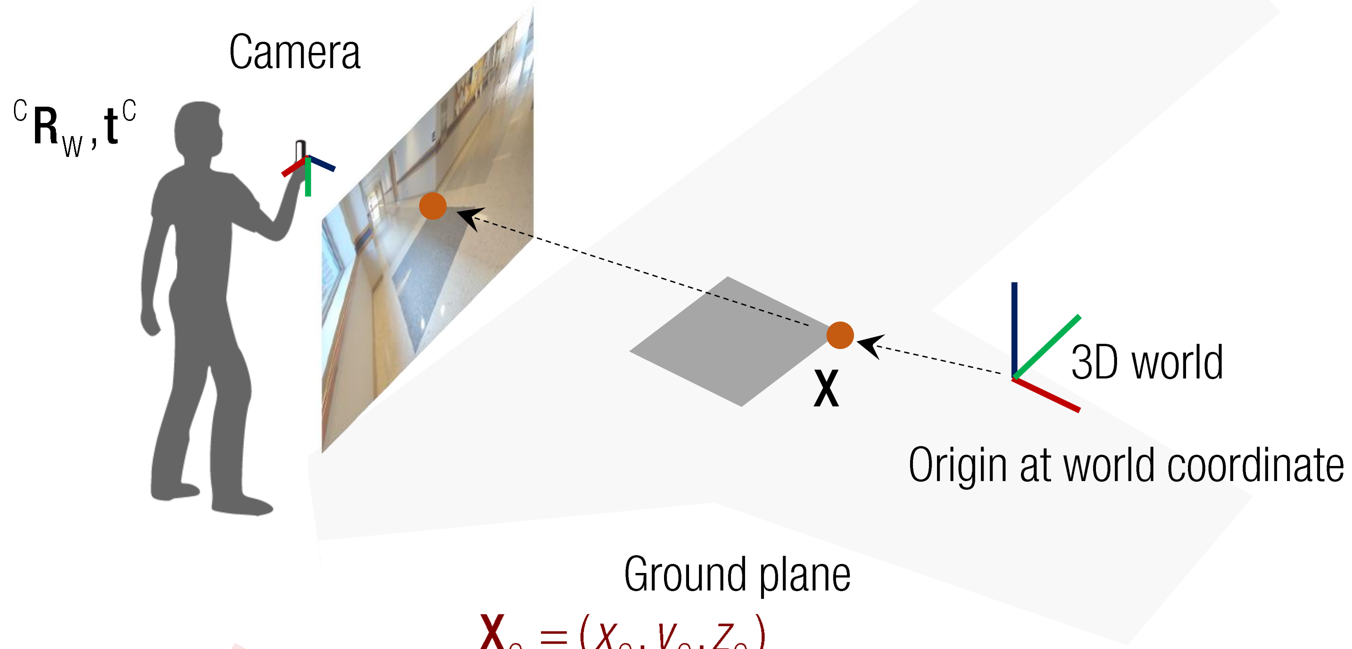


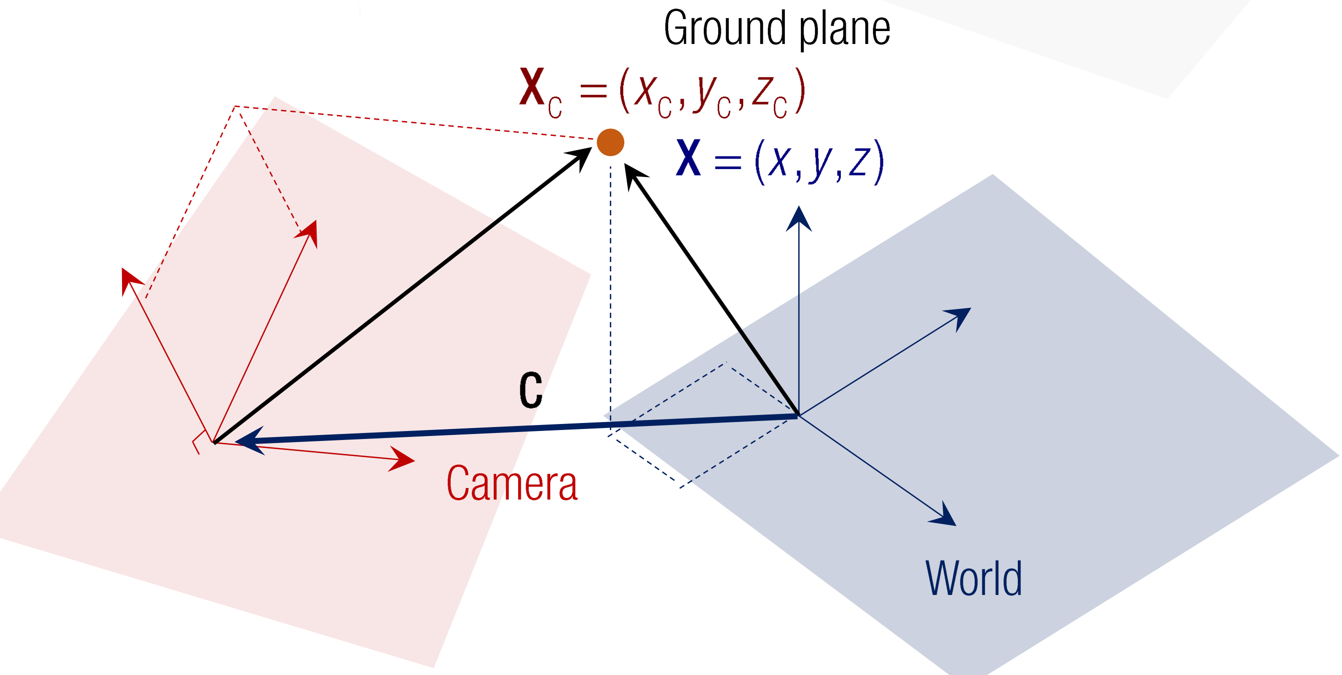
$$\mathbf{X}_C = {}^{C}\mathbf{R}_W \mathbf{X} + {}^{C}\mathbf{t} = \begin{bmatrix} r_{x1} & r_{x2} & r_{x3} & t_x \\ r_{y1} & r_{y2} & r_{y3} & t_y \\ r_{z1} & r_{z2} & r_{z3} & t_z \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

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

Rotate and then, translate.

Recall: Translate and the, Rotate





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

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

Rotate and then, translate.

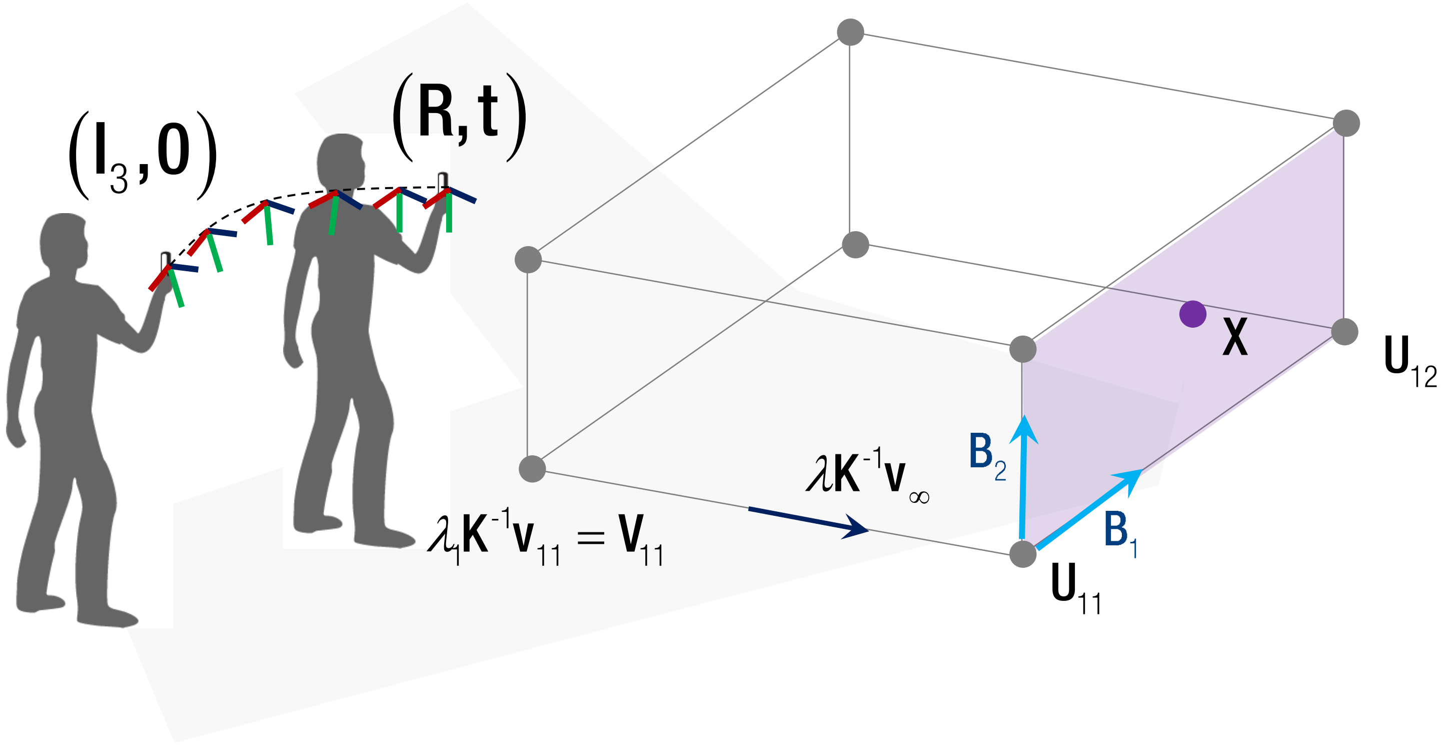
---

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

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

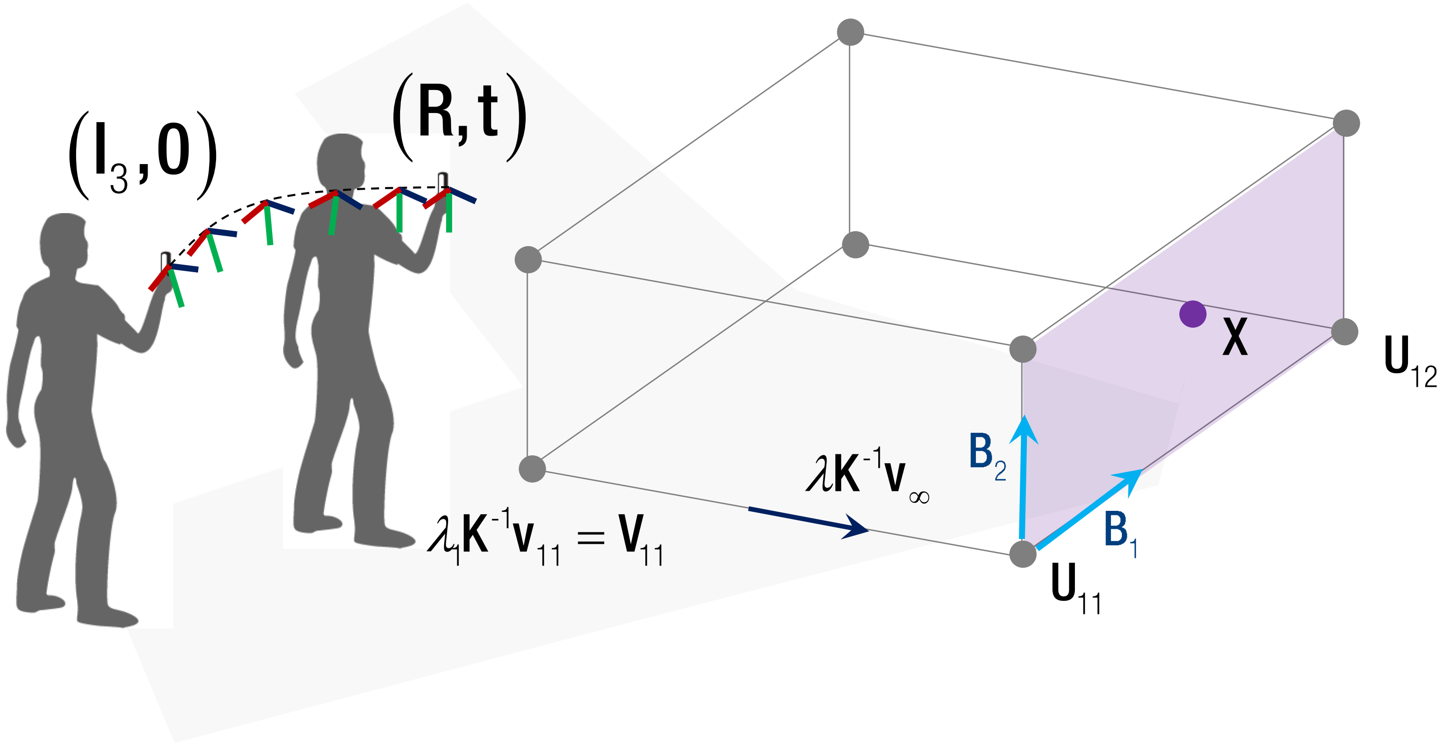
where $\mathbf{C}$ is translation from world to camera seen from world.

Interpolation of Translation



$$\lambda \tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{C} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Interpolation of Translation



$$\lambda \tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} = \mathbf{KR}\begin{bmatrix}\mathbf{I}_3 & -\mathbf{C}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

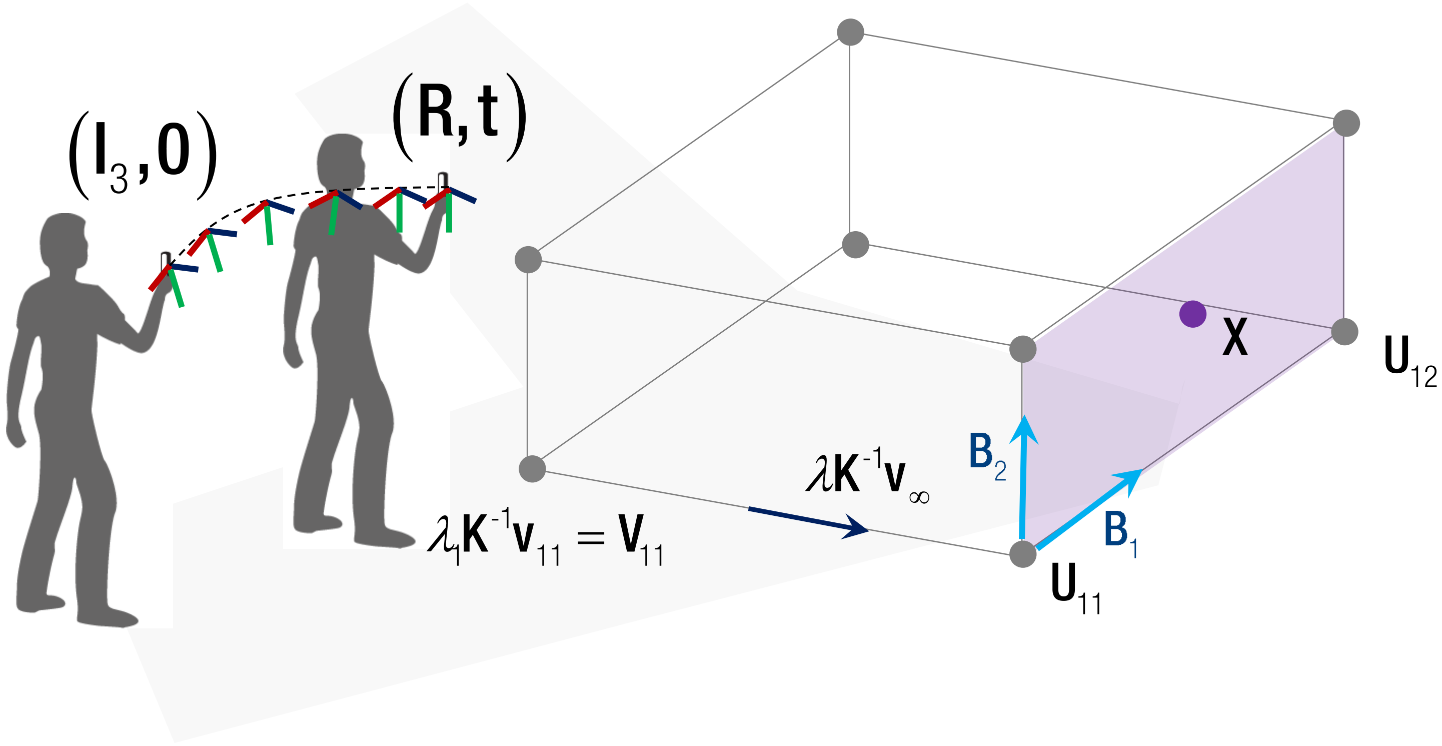
Rot. ➔ Trans. Trans. ➔ Rot.

Translation is independent on rotation.

How to interpolate translation?

$$\mathbf{C}_1 = \begin{bmatrix}C_1^x\\C_1^y\\C_1^z\end{bmatrix} \rightarrow \mathbf{C}_2 = \begin{bmatrix}C_2^x\\C_2^y\\C_2^z\end{bmatrix}$$

Interpolation of Translation



$$\lambda \tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\begin{bmatrix}X \\ Y \\ Z \\ 1\end{bmatrix} = \mathbf{KR}\begin{bmatrix}\mathbf{I}_3 & -\mathbf{C}\end{bmatrix}\begin{bmatrix}X \\ Y \\ Z \\ 1\end{bmatrix}$$

Rot. ➔ Trans. Trans. ➔ Rot.

Translation is independent on rotation.

How to interpolate translation?

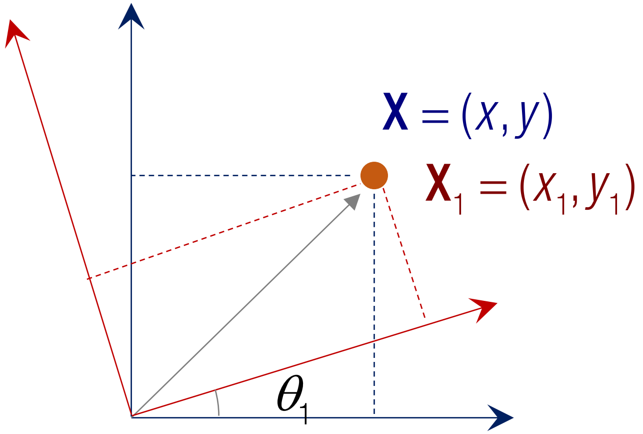
$$\mathbf{C}_1 = \begin{bmatrix}C_1^x \\ C_1^y \\ C_1^z\end{bmatrix} \rightarrow \mathbf{C}_2 = \begin{bmatrix}C_2^x \\ C_2^y \\ C_2^z\end{bmatrix}$$

Interpolated camera center:

$$\mathbf{C}_i = w\mathbf{C}_1 + (1-w)\mathbf{C}_2 \qquad w \in [0,1]$$

Interpolation of Rotation

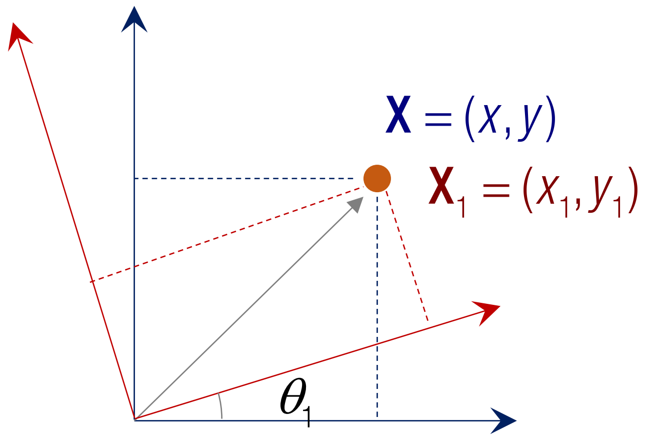
2D coordinate transform:



$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \qquad \begin{bmatrix} x \\ y \end{bmatrix}$$

Interpolation of Rotation

2D coordinate transform:

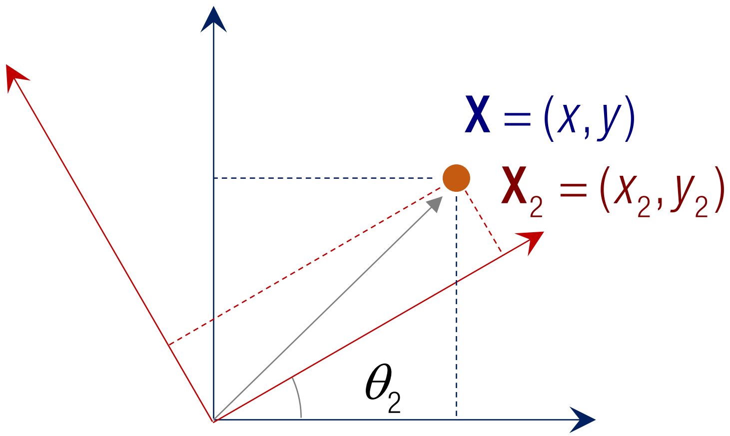


$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\det\left(\begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \right) = \cos^2\theta_1 + \sin^2\theta_1 = 1$$

Interpolation of Rotation

2D coordinate transform:



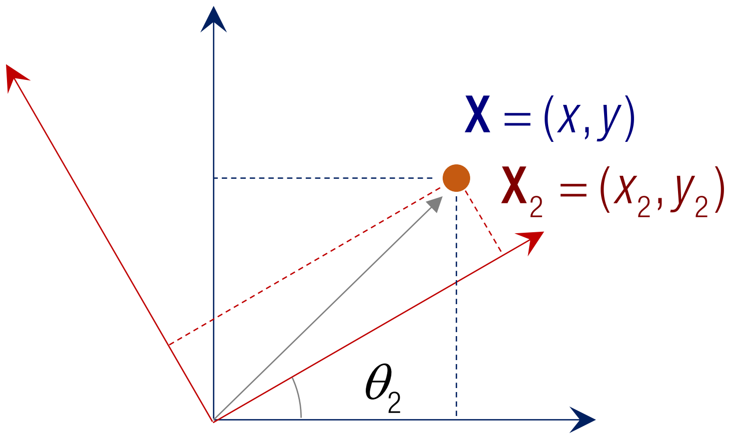
$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 \\ -\sin\theta_2 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Interpolation of Rotation

2D coordinate transform:



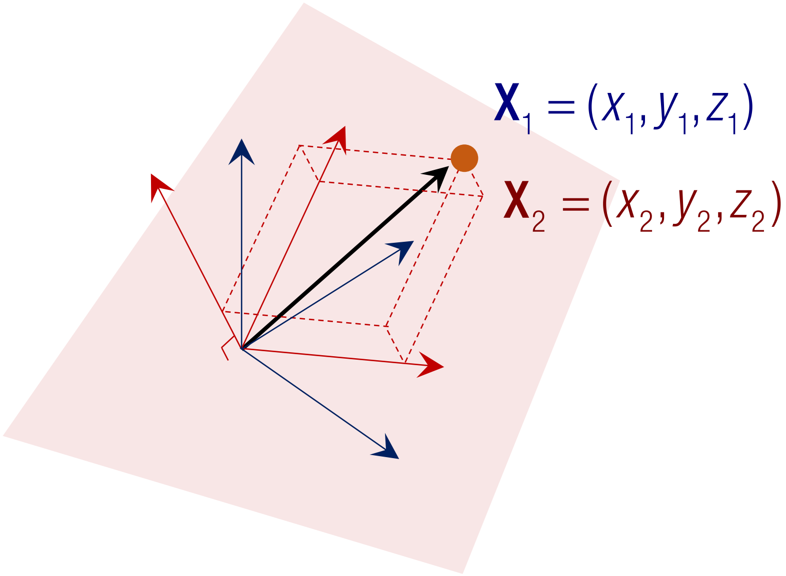
$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\theta = w\theta_1 + (1-w)\theta_2$$

$$w \in [0,1]$$

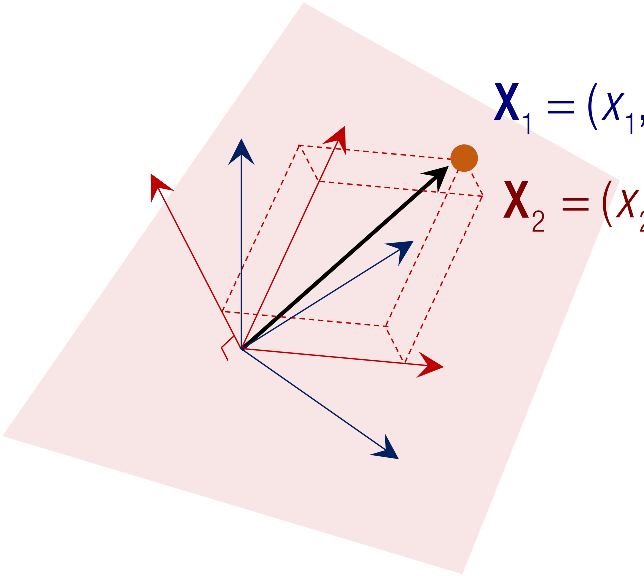
$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 \\ -\sin\theta_2 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Interpolation of Rotation in 3D



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

Interpolation of Rotation in 3D



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

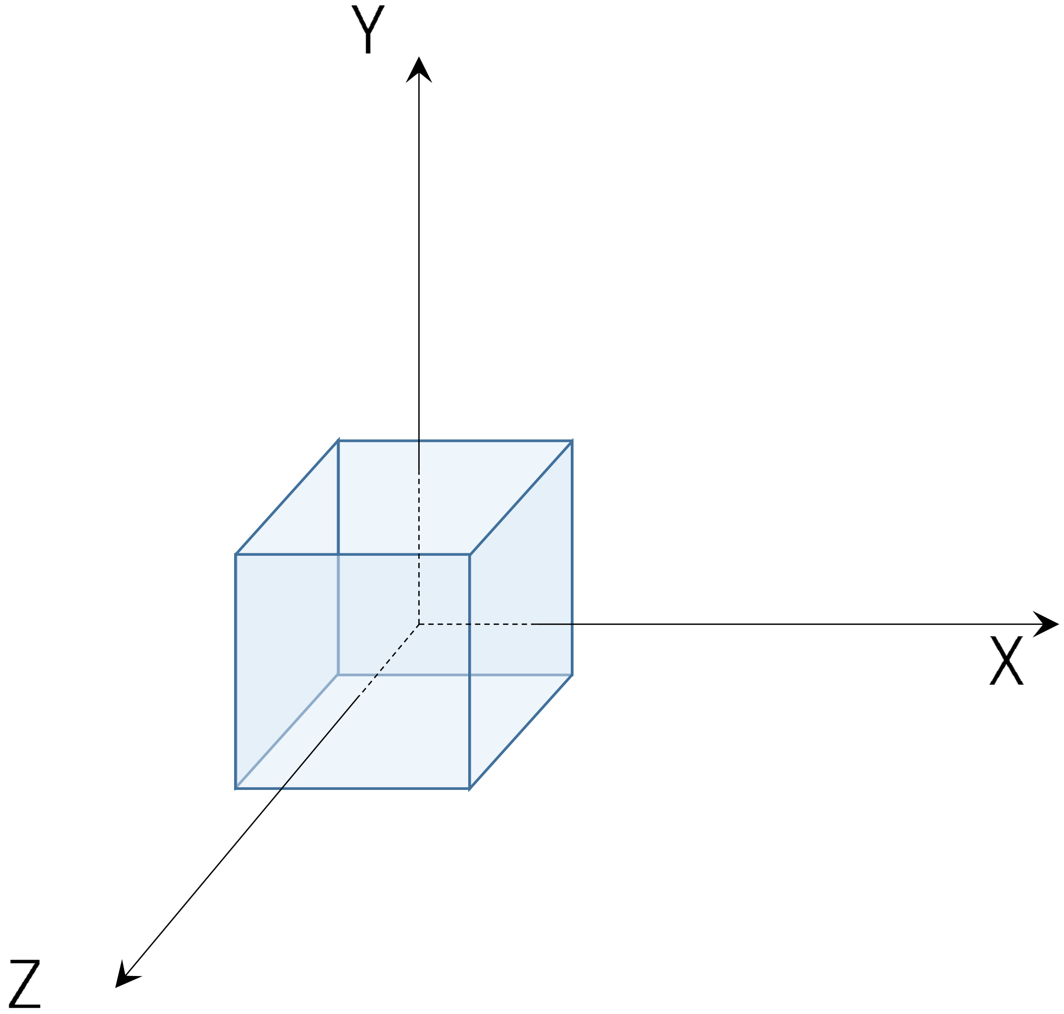
How to interpolate between two coordinates?

$$\mathbf{R}_1 \rightarrow \mathbf{R}_2$$

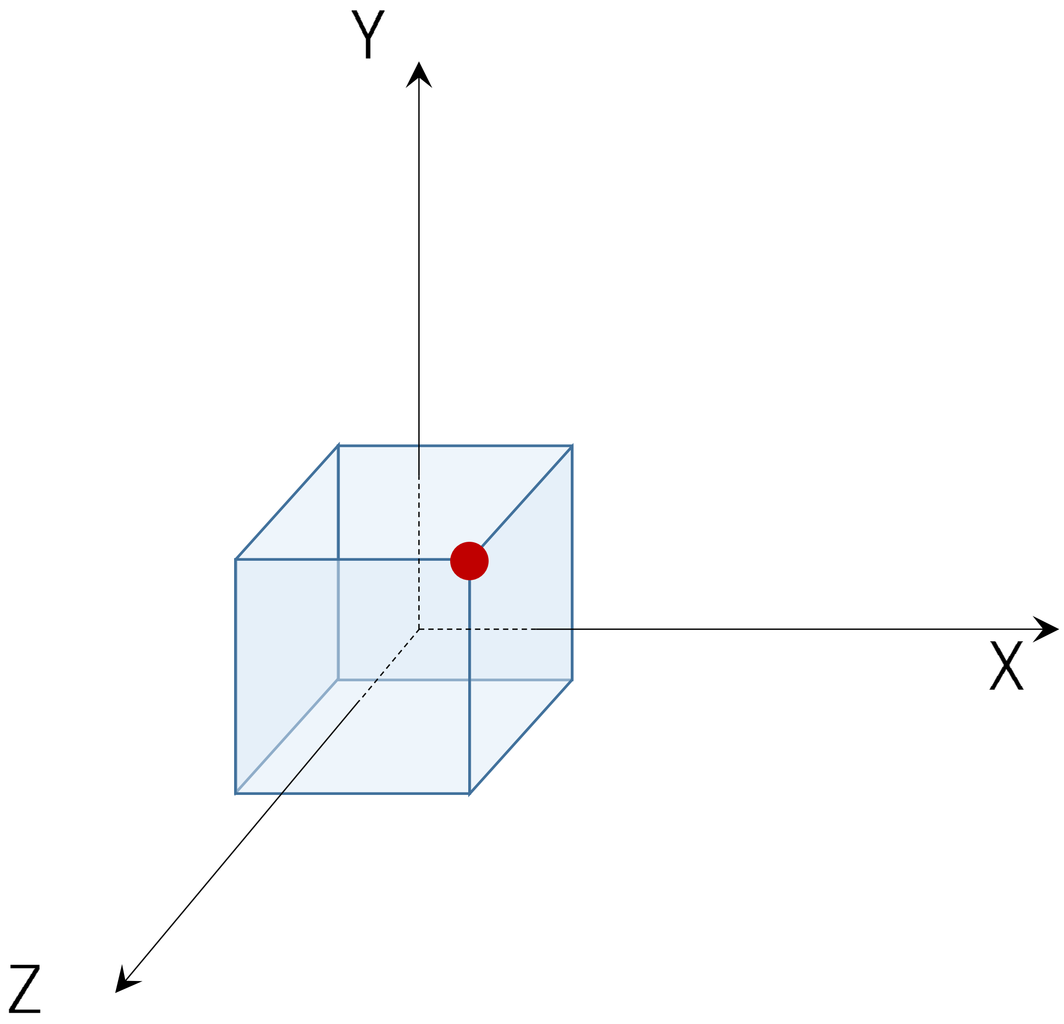
dof: 3

of parameters: 9

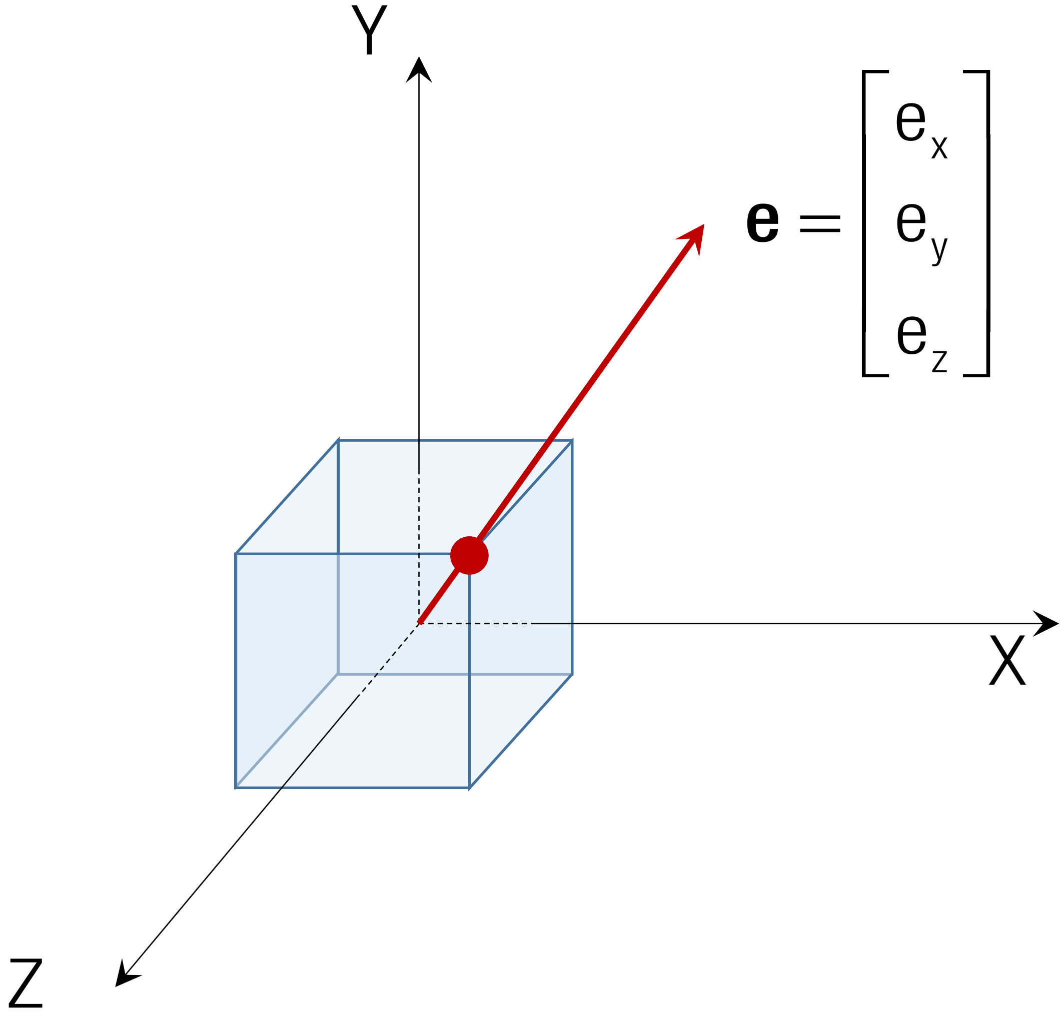
Axis Angle Representation



Axis Angle Representation

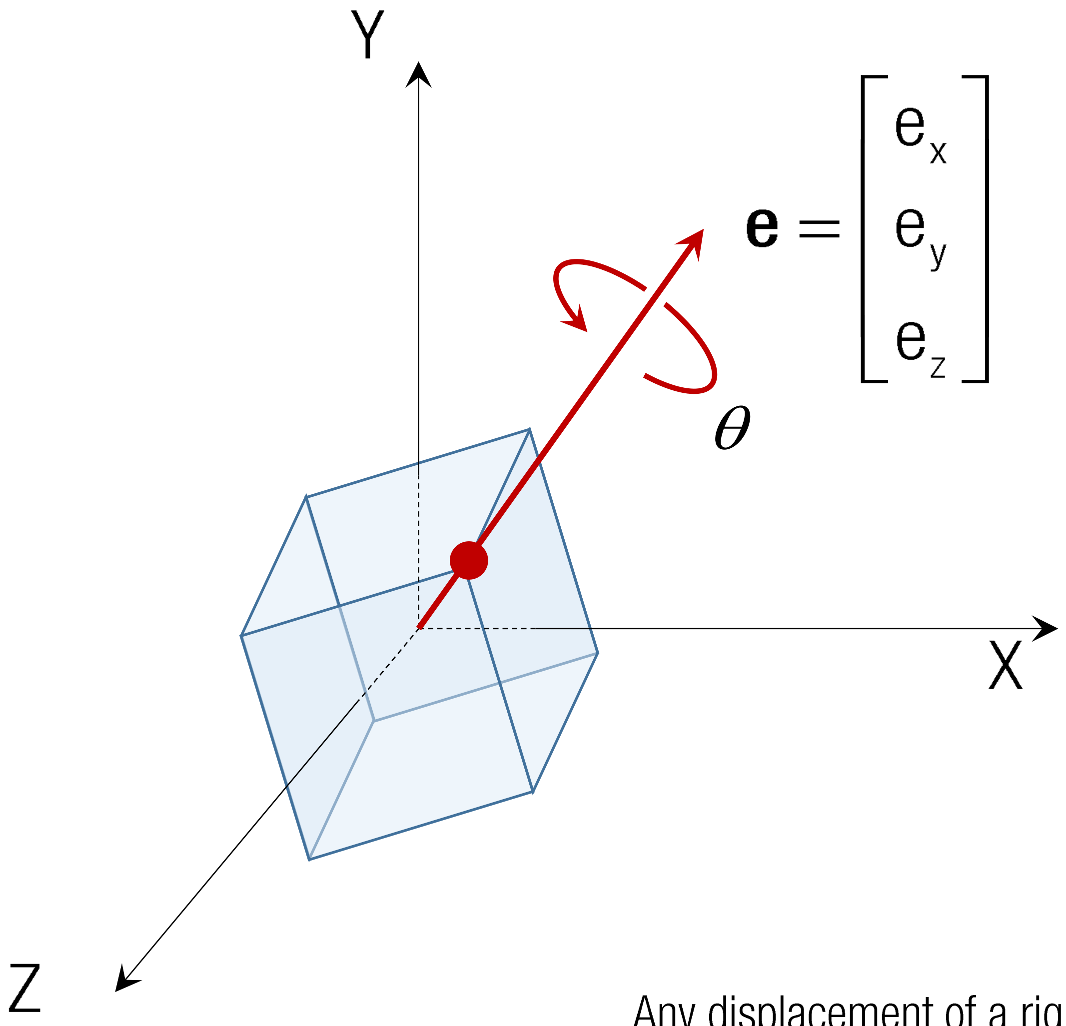


Axis Angle Representation



$$\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}$$

Axis Angle Representation

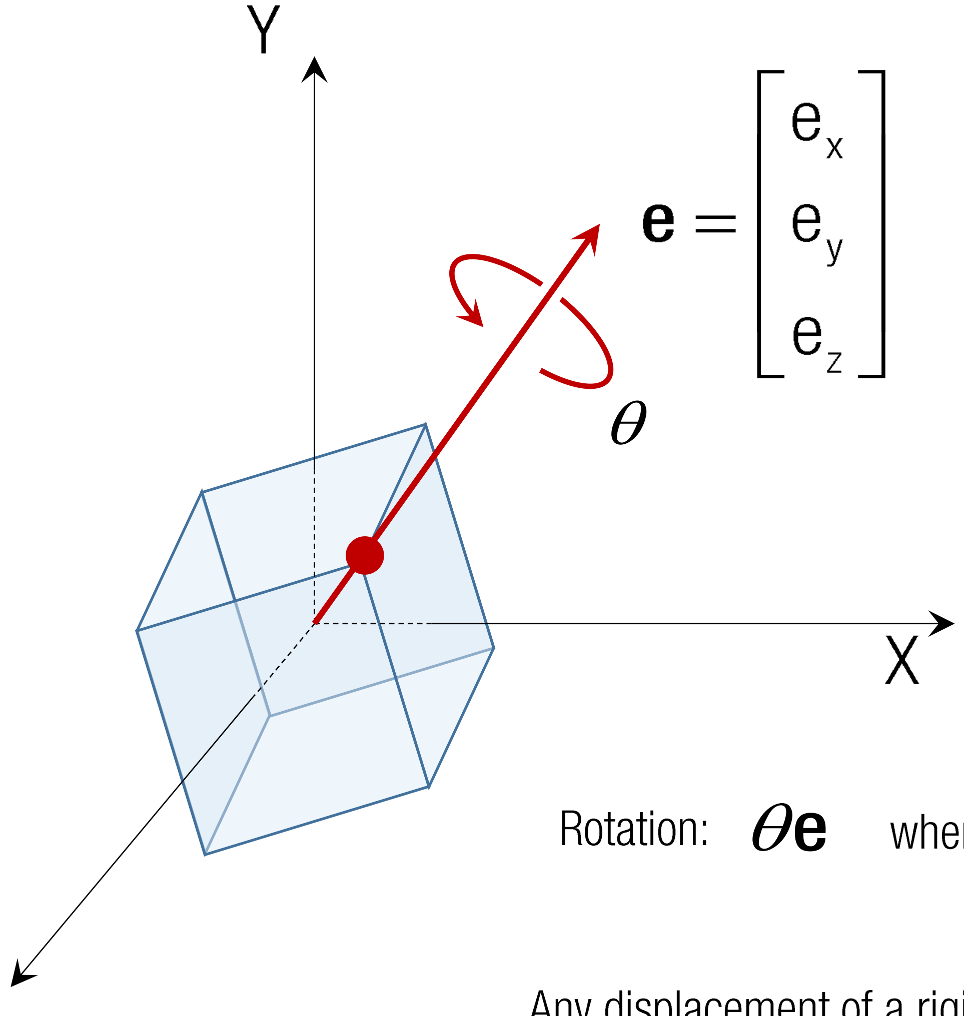


$$\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}$$

Euler's theorem

Any displacement of a rigid body such that a point on the rigid body remains fixed, is equivalent to a single rotation about some axis that runs through the fixed point.

Axis Angle Representation



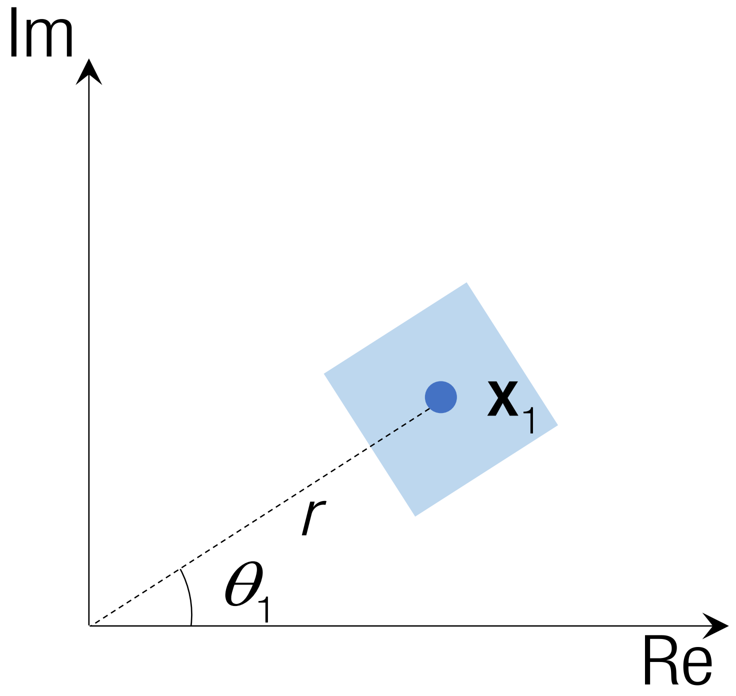
$$\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}$$

Rotation: $\theta\mathbf{e}$ where $\|\mathbf{e}\| = 1$

Euler's theorem

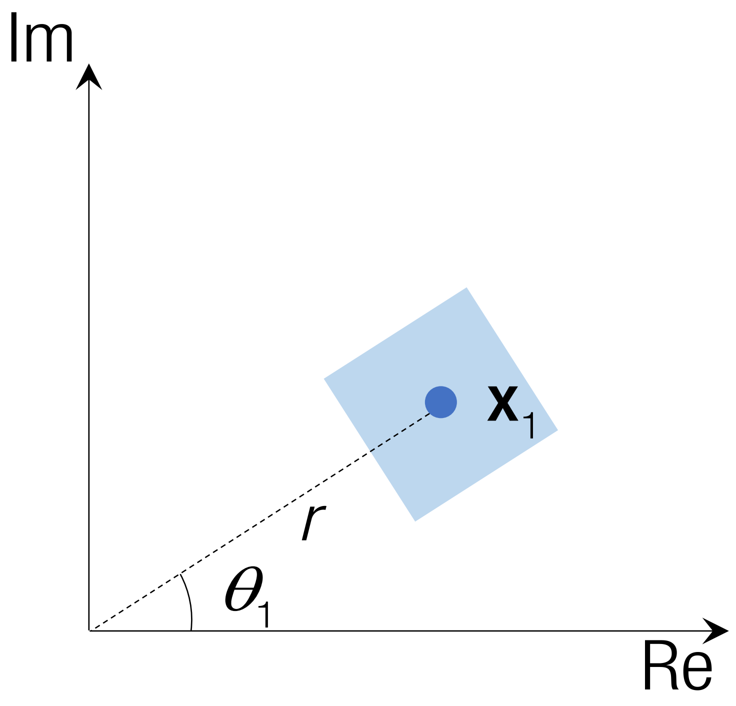
Any displacement of a rigid body such that a point on the rigid body remains fixed, is equivalent to a single rotation about some axis that runs through the fixed point.

2D Exponential Map (Euler's Formula)



$$\mathbf{x}_1 =$$

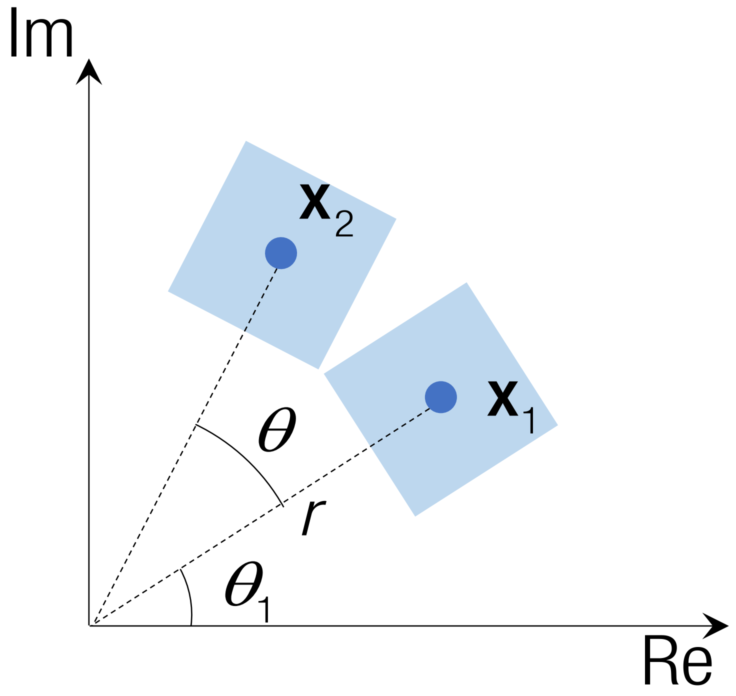
2D Exponential Map (Euler's Formula)



$$\mathbf{x}_1 = r\exp(i\theta_1) = r\left(\cos\theta_1 + i\sin\theta_1\right)$$

Ref) $$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

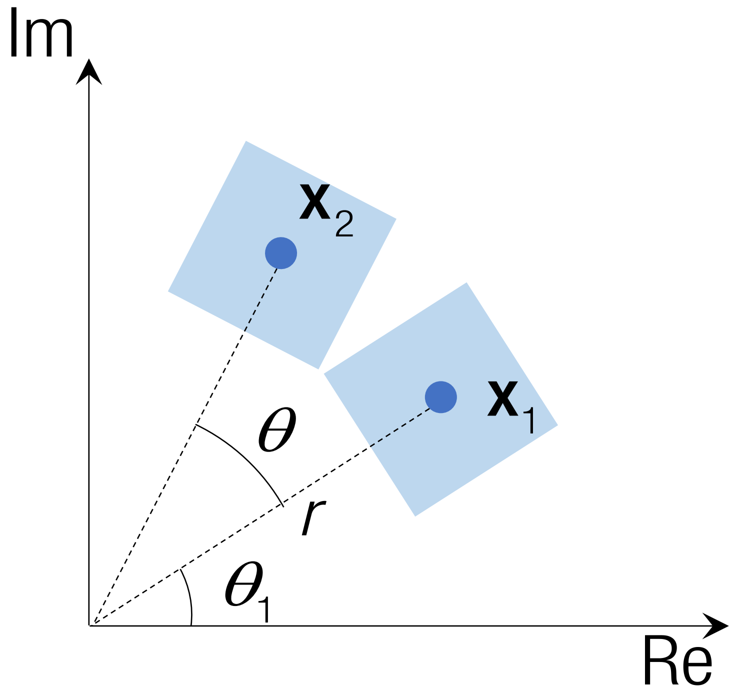
2D Exponential Map (Euler's Formula)



$$\mathbf{x}_1 = r\exp(i\theta_1) = r\left(\cos\theta_1 + i\sin\theta_1\right)$$

$\mathbf{x}_2$

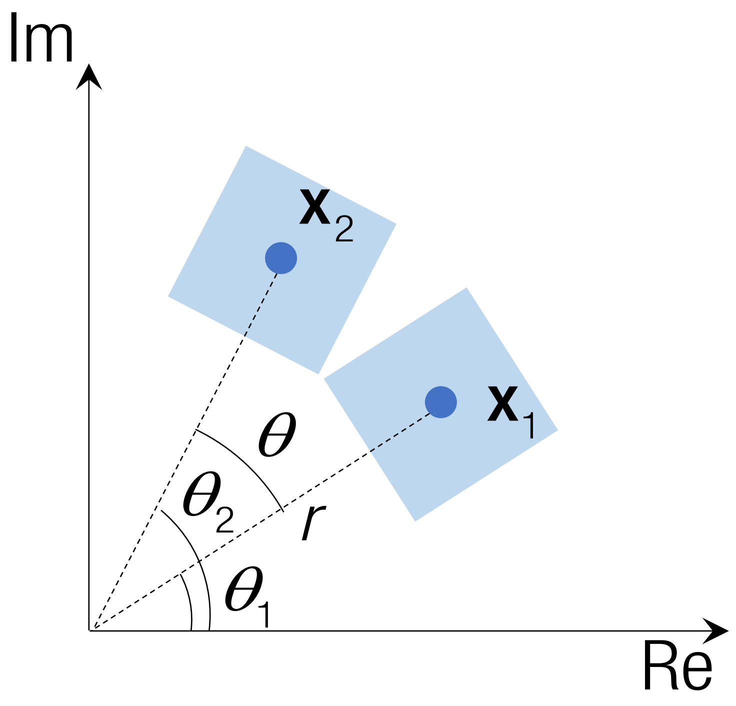
2D Exponential Map (Euler's Formula)



$$\mathbf{x}_1 = r\exp(i\theta_1) = r\left(\cos\theta_1 + i\sin\theta_1\right)$$

$$\begin{aligned}\mathbf{x}_2 &= \exp(i\theta)\mathbf{x}_1 = r\left(\cos\theta + i\sin\theta\right)\left(\cos\theta_1 + i\sin\theta_1\right)\\ &= r\left(\cos\theta\cos\theta_1 - \sin\theta\sin\theta_1 + i\left(\cos\theta\sin\theta_1 + \sin\theta\cos\theta_1\right)\right)\\ &= r\left(\cos(\theta+\theta_1) + i\sin(\theta+\theta_1)\right)\end{aligned}$$

2D Exponential Map (Euler's Formula)

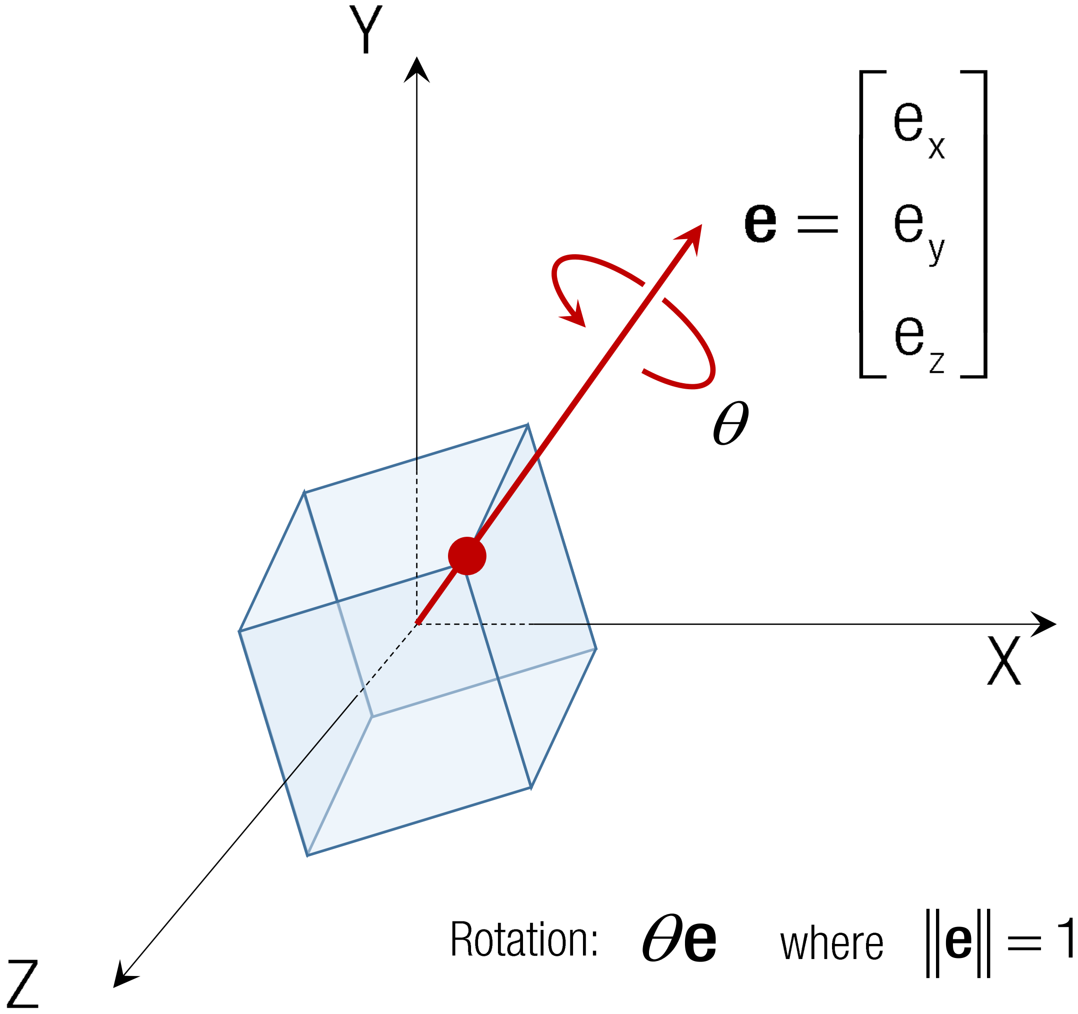


$$\mathbf{x}_1 = r\exp(i\theta_1) = r\left(\cos\theta_1 + i\sin\theta_1\right)$$

$$\begin{aligned}
\mathbf{x}_2 &= \exp(i\theta)\mathbf{x}_1 = r\left(\cos\theta + i\sin\theta\right)\left(\cos\theta_1 + i\sin\theta_1\right) \\
&= r\left(\cos\theta\cos\theta_1 - \sin\theta\sin\theta_1 + i\left(\cos\theta\sin\theta_1 + \sin\theta\cos\theta_1\right)\right) \\
&= r\left(\cos(\theta+\theta_1) + i\sin(\theta+\theta_1)\right) \\
&= r\left(\cos\theta_2 + i\sin\theta_2\right)
\end{aligned}$$

$$\theta_2 = \theta_1 + \theta$$

3D Exponential Map: Quaternion

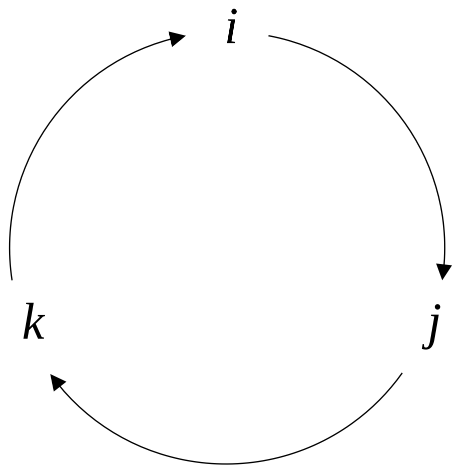


$$\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}$$

Rotation: $\theta\mathbf{e}$ where $\|\mathbf{e}\| = 1$

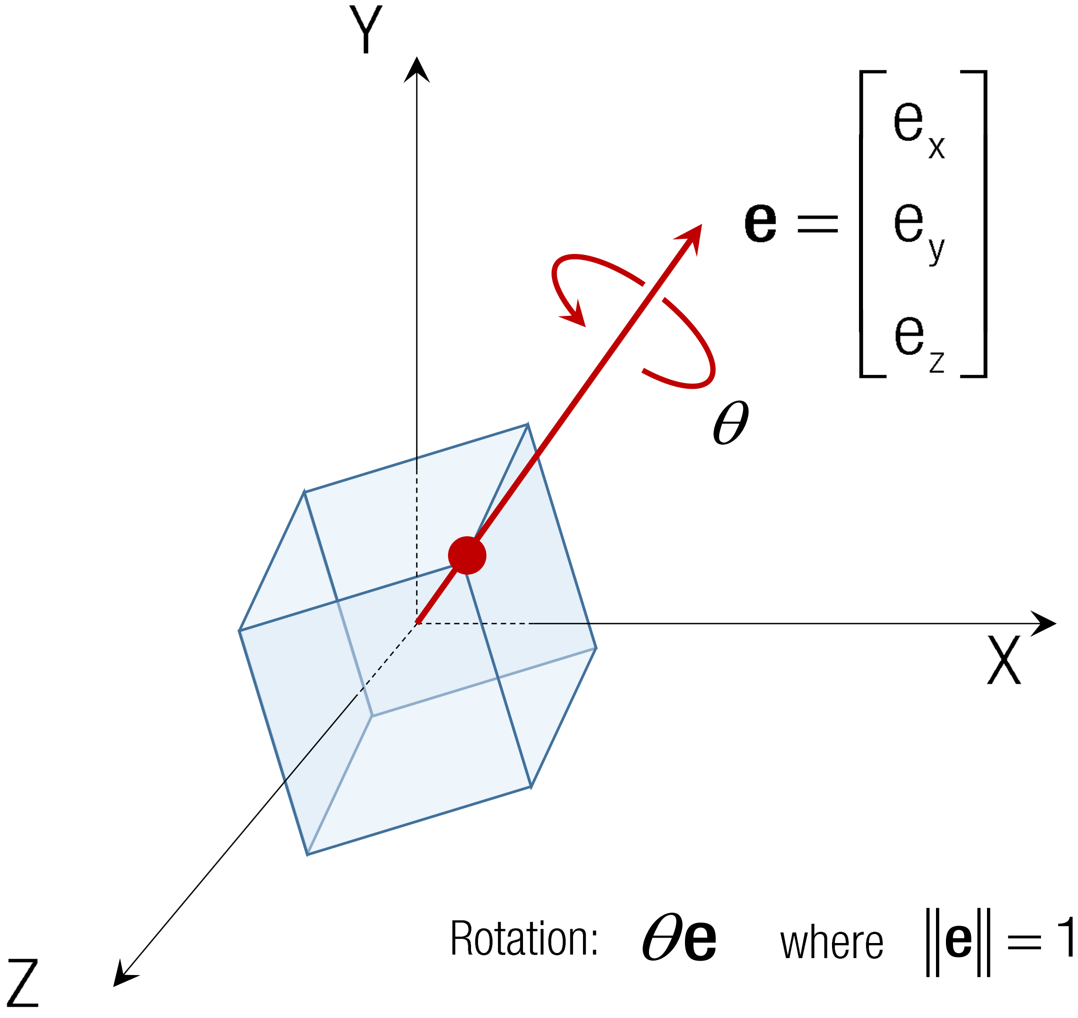
$$\exp\left(\frac{\theta}{2}\mathbf{e}\right) = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\left(i\,e_x + j\,e_y + k\,e_z\right)$$

$$i^2 = j^2 = k^2 = ijk = -1$$



$$ij = k$$
$$jk = i$$
$$ki = j$$

Exercise

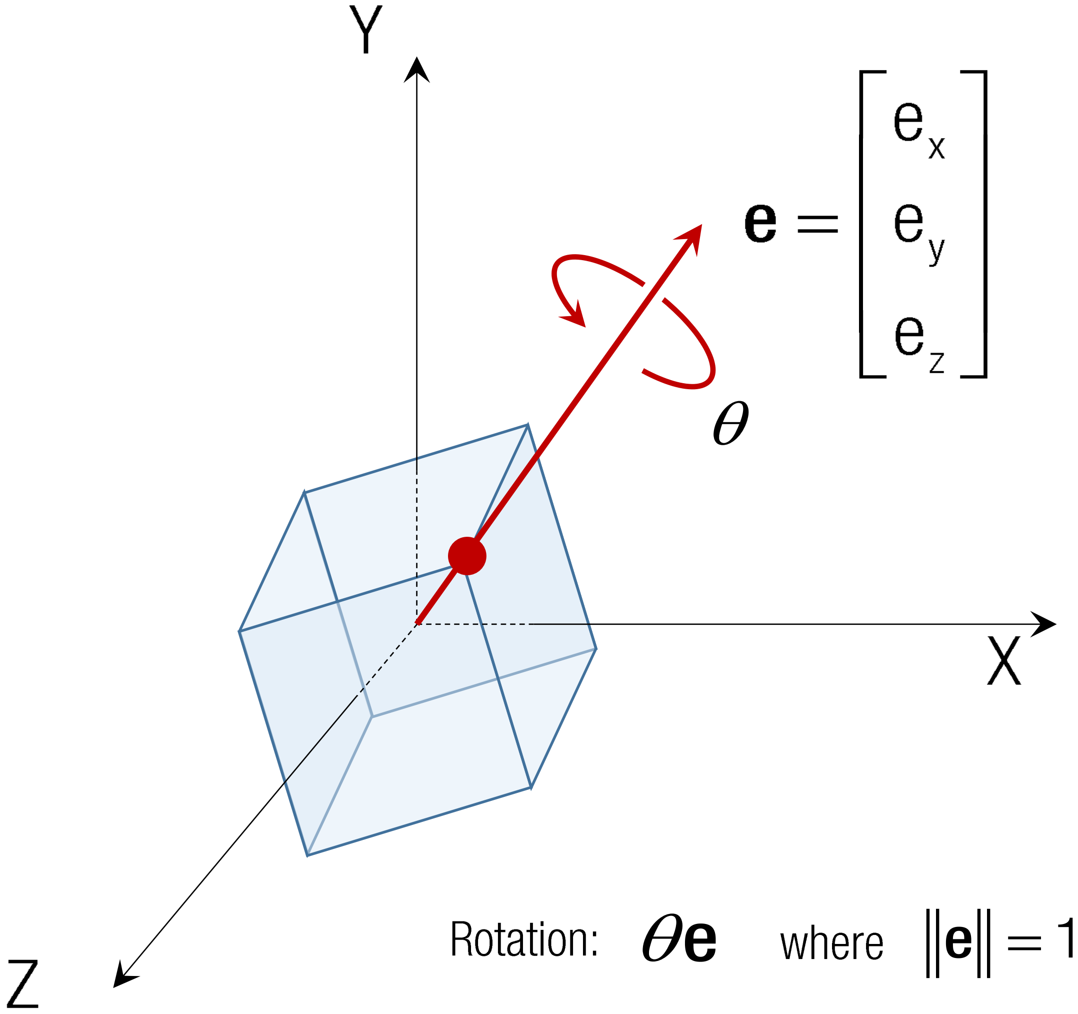


Rotation: $\theta\mathbf{e}$ where $\|\mathbf{e}\| = 1$

$$\exp\left(\frac{\theta}{2}\mathbf{e}\right) = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\left(i\, e_x + j\, e_y + k\, e_z\right)$$

Find a quaternion **q** such that it describes a rotation of 60 degrees about the axis **a**=[3, 4, 0].

Exercise



Rotation: $\theta\mathbf{e}$ where $\|\mathbf{e}\| = 1$

$$\exp\left(\frac{\theta}{2}\mathbf{e}\right) = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\left(i\,e_x + j\,e_y + k\,e_z\right)$$

Find a quaternion **q** such that it describes a rotation of 60 degrees about the axis **a**=[3, 4, 0].

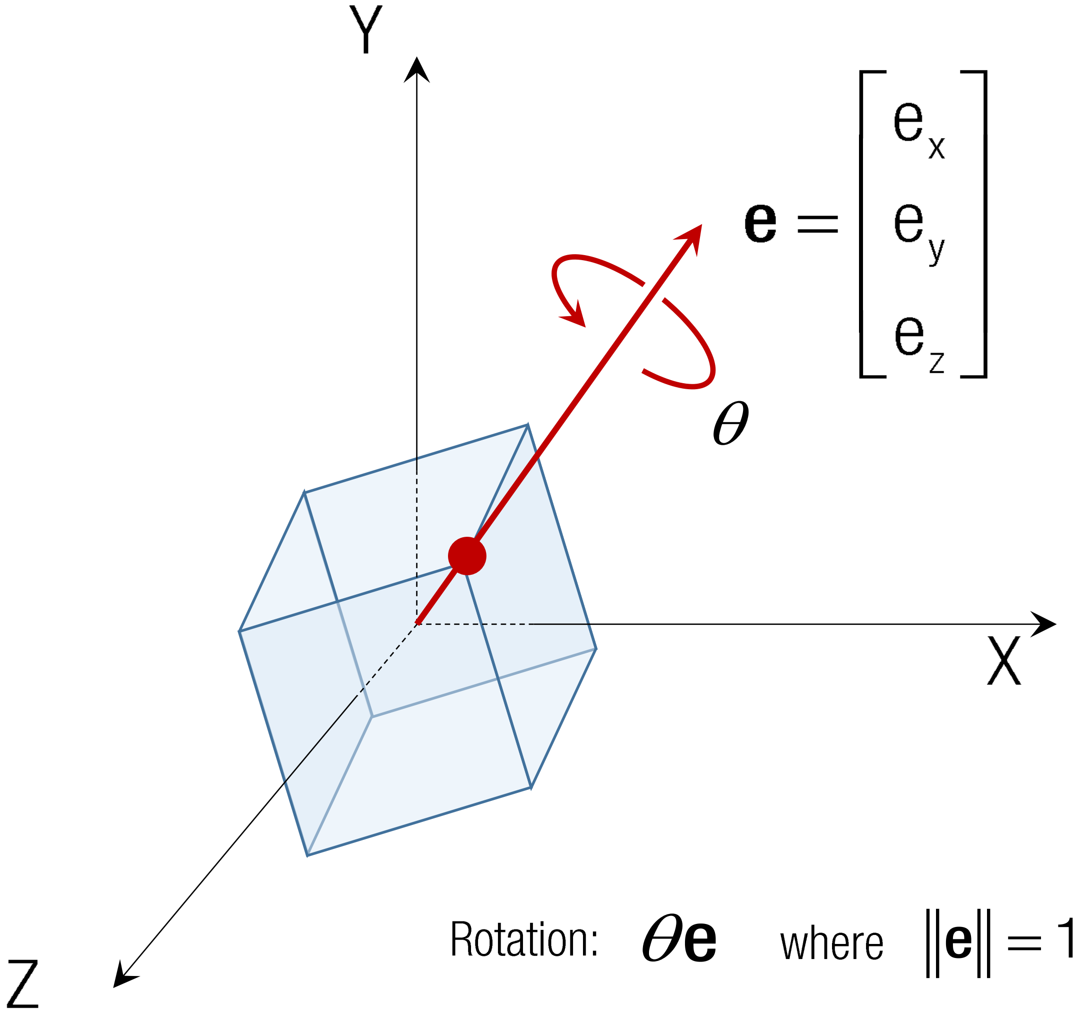
$$\mathbf{e} = \mathbf{a} / \|\mathbf{a}\| = \underbrace{i\frac{3}{5} + j\frac{4}{5} + k\frac{0}{5}}_{\text{Unit vector}}$$

$$\mathbf{q} = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\left(i\frac{3}{5} + j\frac{4}{5} + k\frac{0}{5}\right) \qquad \theta = \frac{\pi}{3}$$

$$= \cos\frac{\pi}{3\cdot 2} + \sin\frac{\pi}{3\cdot 2}\left(i\frac{3}{5} + j\frac{4}{5} + k\frac{0}{5}\right)$$

$$= \frac{\sqrt{3}}{2} + \frac{1}{2}\left(i\frac{3}{5} + j\frac{4}{5} + k\frac{0}{5}\right)$$

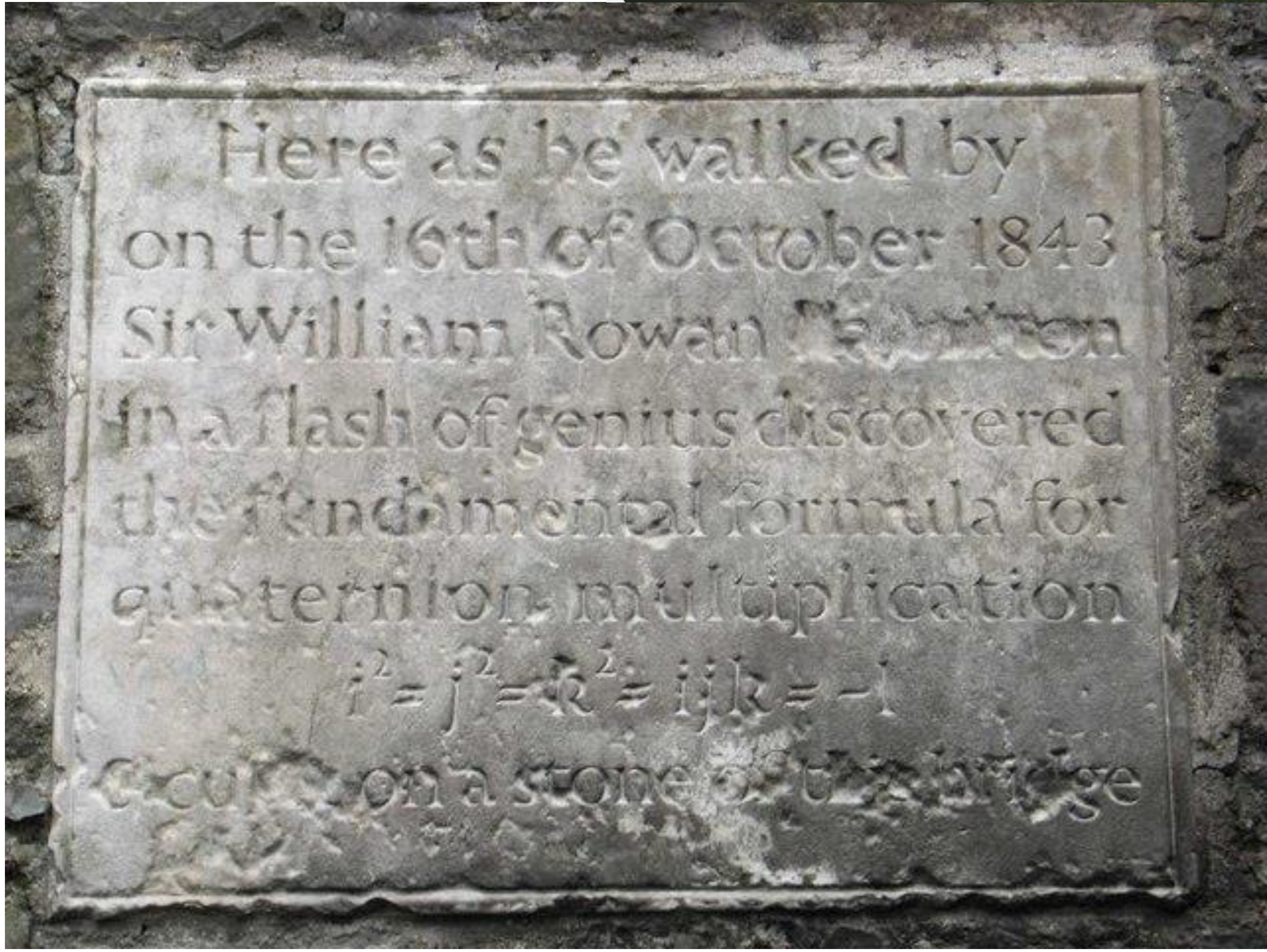
3D Exponential Map: Quaternion



$$\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}$$

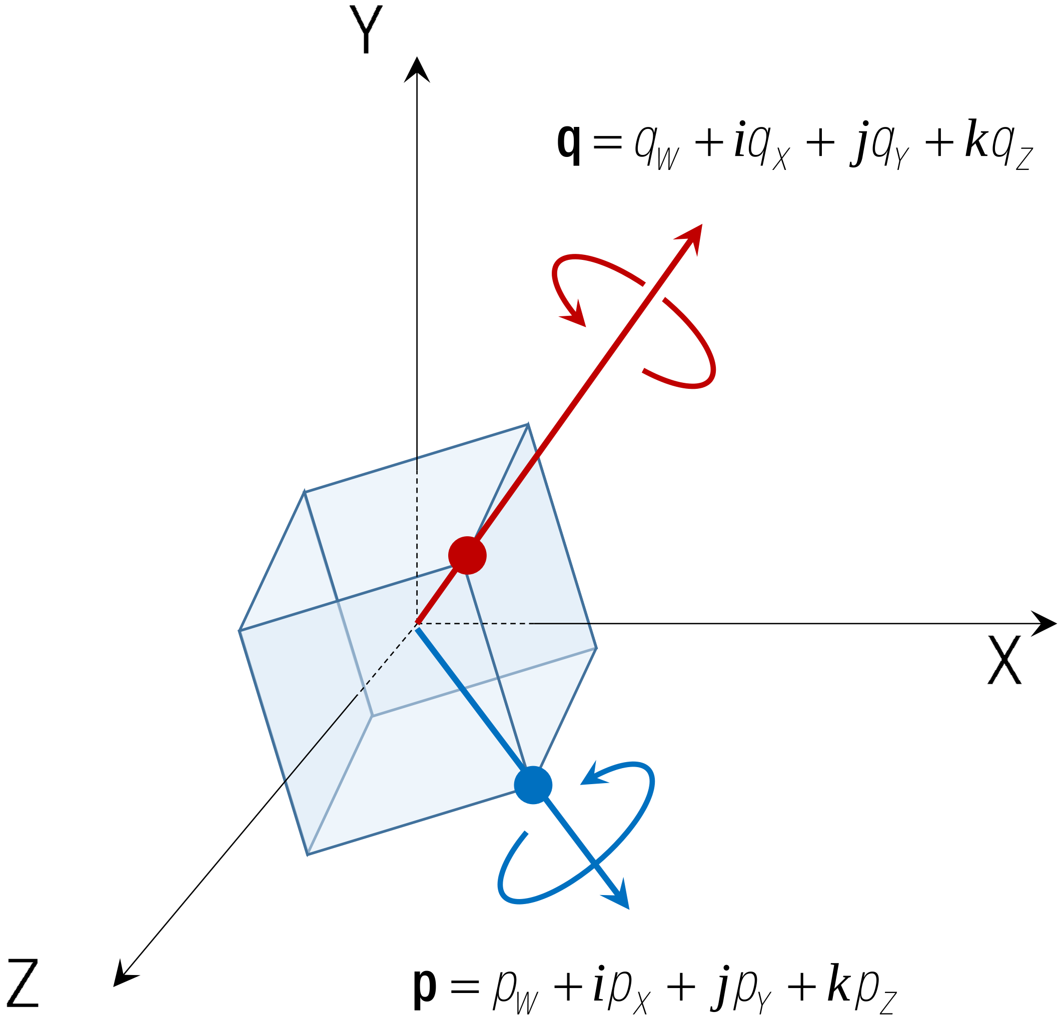
Rotation: $\theta\mathbf{e}$ where $\|\mathbf{e}\| = 1$


Broom Bridge, Dublin, Ireland



Quaternion Product

$$\mathbf{q} = q_W + iq_X + jq_Y + kq_Z$$



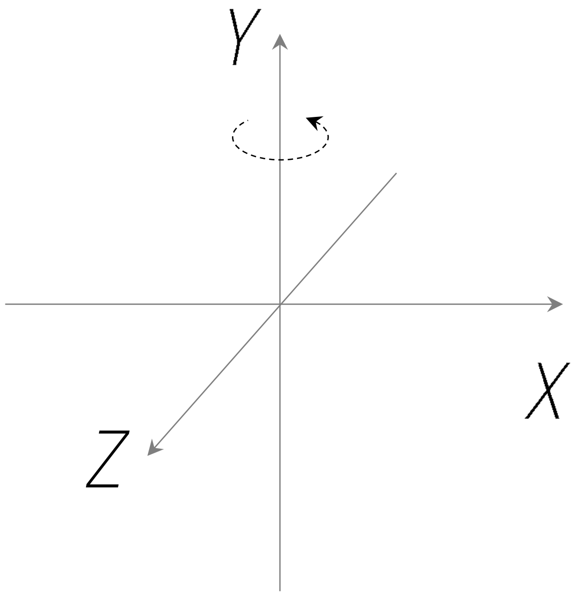
$$\mathbf{p} = p_W + ip_X + jp_Y + kp_Z$$

Rotate **q** and then, **p**:

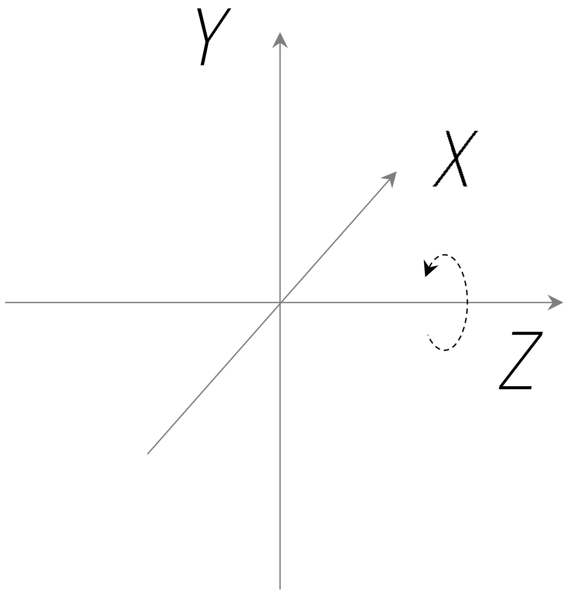
$$\begin{aligned}
\mathbf{qp} &= \left(q_W + iq_X + jq_Y + kq_Z\right)\left(p_W + ip_X + jp_Y + kp_Z\right) \\
&= \left(q_W p_W - q_X p_X - q_Y p_Y - q_Z p_Z\right) + i\left(q_W p_X + q_X p_W + q_Y p_Z - q_Z p_Y\right) \\
&\quad + j\left(q_W p_Y - q_X p_Z + q_Y p_W + q_Z p_X\right) + k\left(q_W p_Z + q_X p_Y - q_Y p_X + q_Z p_W\right) \\
&= \left(q_W p_W - \hat{\mathbf{q}}\cdot\hat{\mathbf{p}}\right) + \left(q_W \hat{\mathbf{p}} + p_W \hat{\mathbf{q}} + \hat{\mathbf{q}}\times\hat{\mathbf{p}}\right)
\end{aligned}$$

where $\hat{\mathbf{q}} = iq_X + jq_Y + kq_Z$

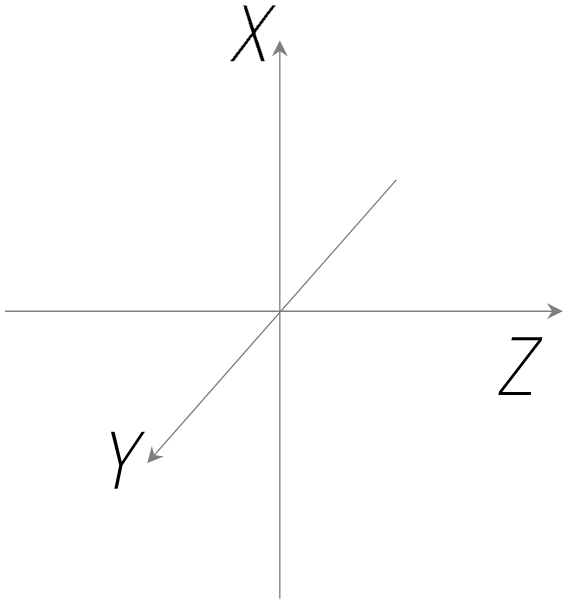
Quaternion Product Example



Rotating 90 degrees about *Y* axis.



Rotating 90 degrees about *Z* axis.



$$\mathbf{q}_1 = \cos\frac{\pi/2}{2} + j\sin\frac{\pi/2}{2}$$

$$\mathbf{q}_2 = \cos\frac{\pi/2}{2} + k\sin\frac{\pi/2}{2}$$

$$\mathbf{q}_{12} = \mathbf{q}_1\mathbf{q}_2 \quad ?$$

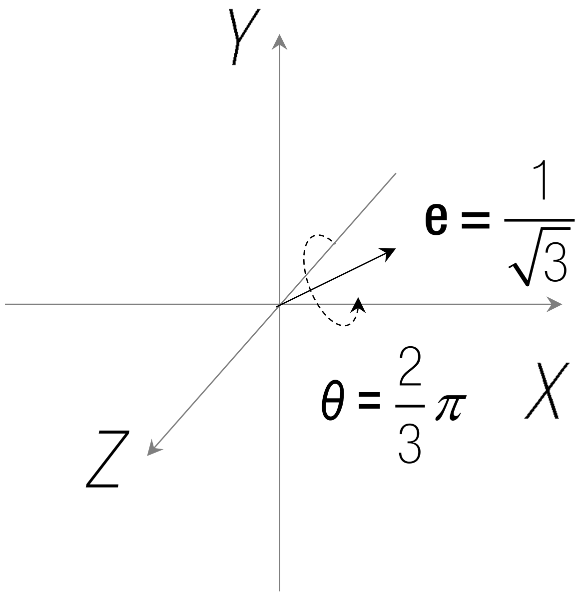
Quaternion Product Example

$$\mathbf{qp} = \left(q_W p_W - \hat{\mathbf{q}} \cdot \hat{\mathbf{p}}\right) + \left(q_W \hat{\mathbf{p}} + p_W \hat{\mathbf{q}} + \hat{\mathbf{q}} \times \hat{\mathbf{p}}\right)$$

$$\mathbf{q}_1 = \cos\frac{\pi/2}{2} + j\sin\frac{\pi/2}{2} \qquad \mathbf{q}_2 = \cos\frac{\pi/2}{2} + k\sin\frac{\pi/2}{2}$$

Quaternion Product Example

$$\mathbf{qp} = \left(q_W p_W - \hat{\mathbf{q}} \cdot \hat{\mathbf{p}}\right) + \left(q_W \hat{\mathbf{p}} + p_W \hat{\mathbf{q}} + \hat{\mathbf{q}} \times \hat{\mathbf{p}}\right)$$



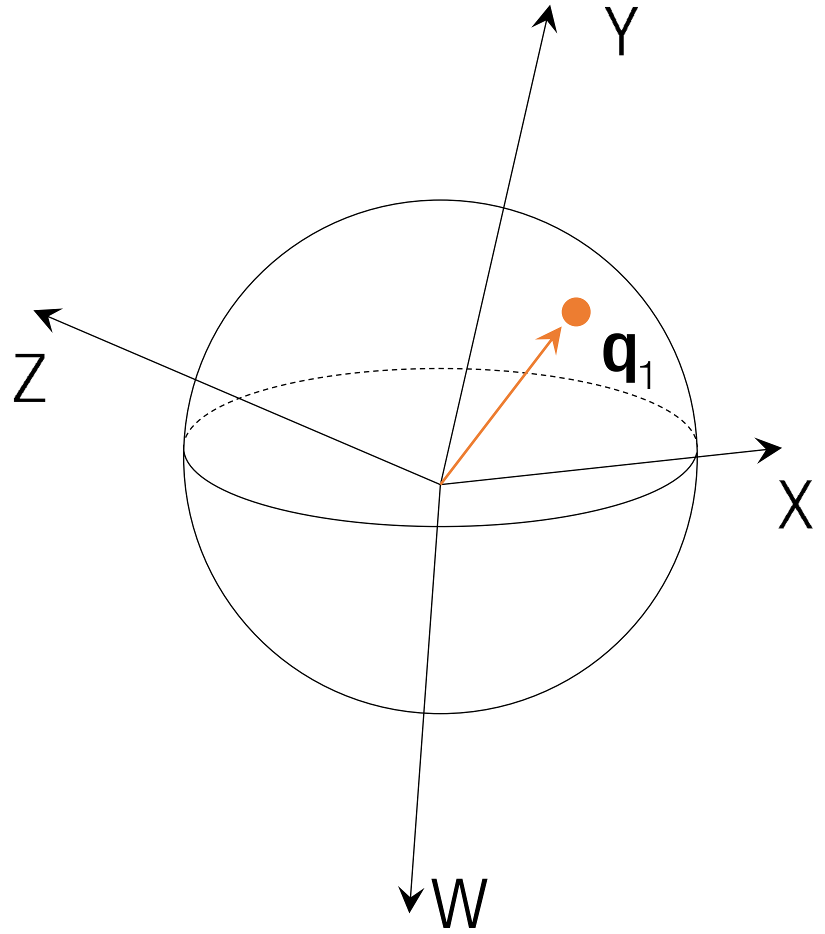
$$\mathbf{e} = \frac{1}{\sqrt{3}}\left(i + j + k\right)$$

$$\theta = \frac{2}{3}\pi$$

$$\mathbf{q}_1 = \cos\frac{\pi/2}{2} + j\sin\frac{\pi/2}{2} \qquad \mathbf{q}_2 = \cos\frac{\pi/2}{2} + k\sin\frac{\pi/2}{2}$$

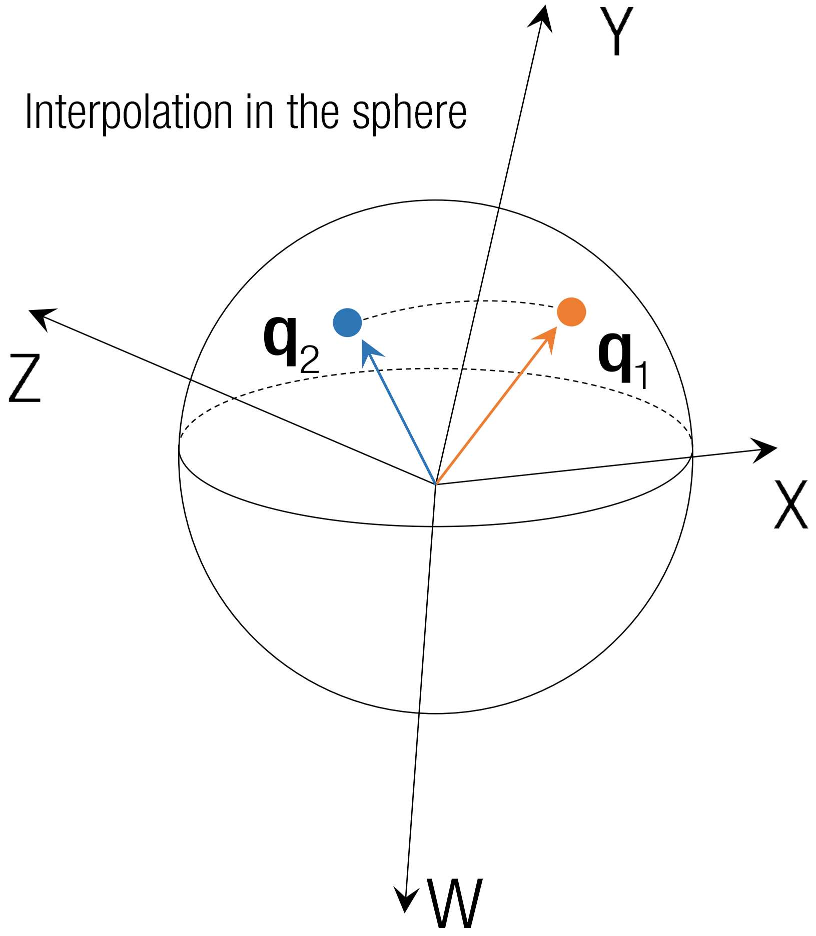
$$\begin{aligned}
\mathbf{q}_{12} &= \mathbf{q}_1\mathbf{q}_2 \\
&= \left(\frac{\sqrt{2}}{2} + \boldsymbol{i}\frac{\sqrt{2}}{2}\right)\left(\frac{\sqrt{2}}{2} + \boldsymbol{j}\frac{\sqrt{2}}{2}\right) \\
&= \frac{1}{2} + \frac{1}{2}\boldsymbol{i} + \frac{1}{2}\boldsymbol{j} + \frac{1}{2}\boldsymbol{k} \\
&= \frac{1}{2} + \frac{1}{2}\boldsymbol{i} + \frac{1}{2}\boldsymbol{j} + \frac{1}{2}\boldsymbol{k} \\
&= \frac{1}{2} + \frac{\sqrt{3}}{2}\left(\frac{1}{\sqrt{3}}\boldsymbol{i} + \frac{1}{\sqrt{3}}\boldsymbol{j} + \frac{1}{\sqrt{3}}\boldsymbol{k}\right)
\end{aligned}$$

Quaternion in 4D Sphere



$$\mathbf{q} = q_W + \boldsymbol{i} q_X + \boldsymbol{j} q_Y + \boldsymbol{k} q_Z = \begin{bmatrix} q_W \\ q_X \\ q_Y \\ q_Z \end{bmatrix} \qquad q_W^2 + q_X^2 + q_Y^2 + q_Z^2 = 1$$

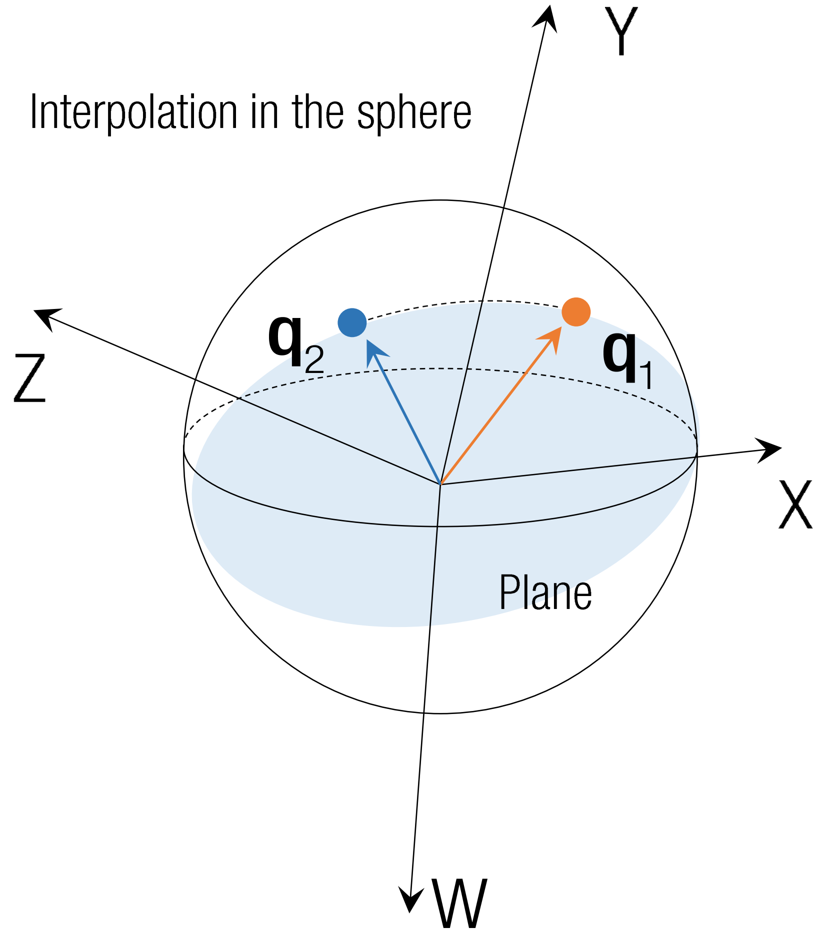
Quaternion in 4D Sphere



$$\mathbf{q} = q_W + \boldsymbol{i}q_X + \boldsymbol{j}q_Y + \boldsymbol{k}q_Z = \begin{bmatrix} q_W \\ q_X \\ q_Y \\ q_Z \end{bmatrix}$$

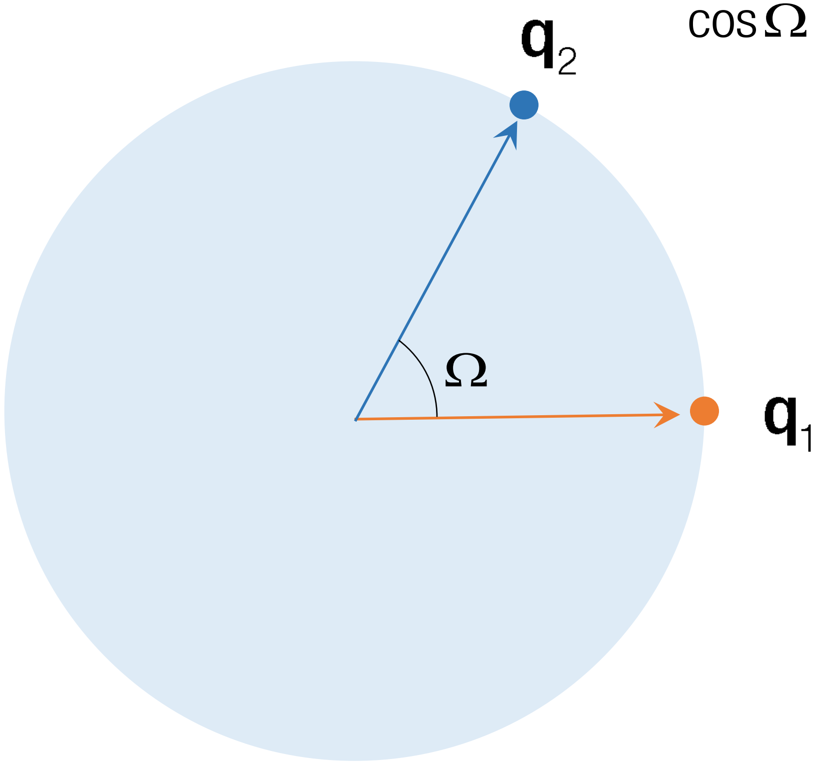
$$q_W^2 + q_X^2 + q_Y^2 + q_Z^2 = 1$$

Quaternion in 4D Sphere



$$\mathbf{q} = q_W + iq_X + jq_Y + kq_Z = \begin{bmatrix} q_W \\ q_X \\ q_Y \\ q_Z \end{bmatrix} \qquad q_W^2 + q_X^2 + q_Y^2 + q_Z^2 = 1$$

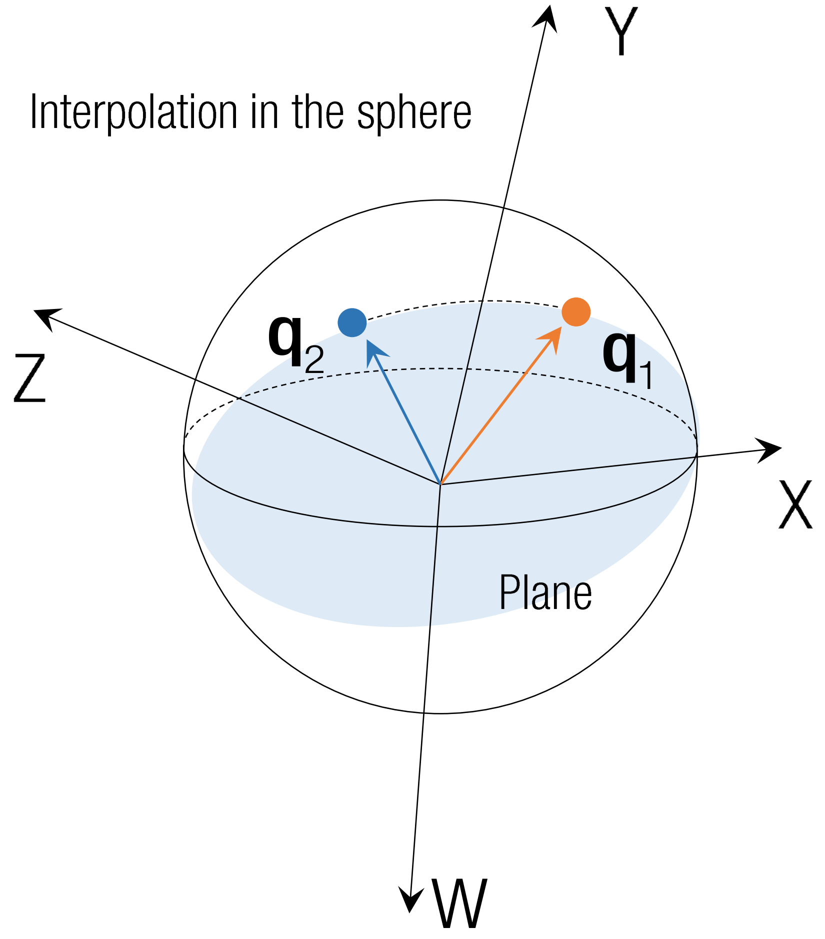
$$\cos\Omega = \mathbf{q}_1 \cdot \mathbf{q}_2$$



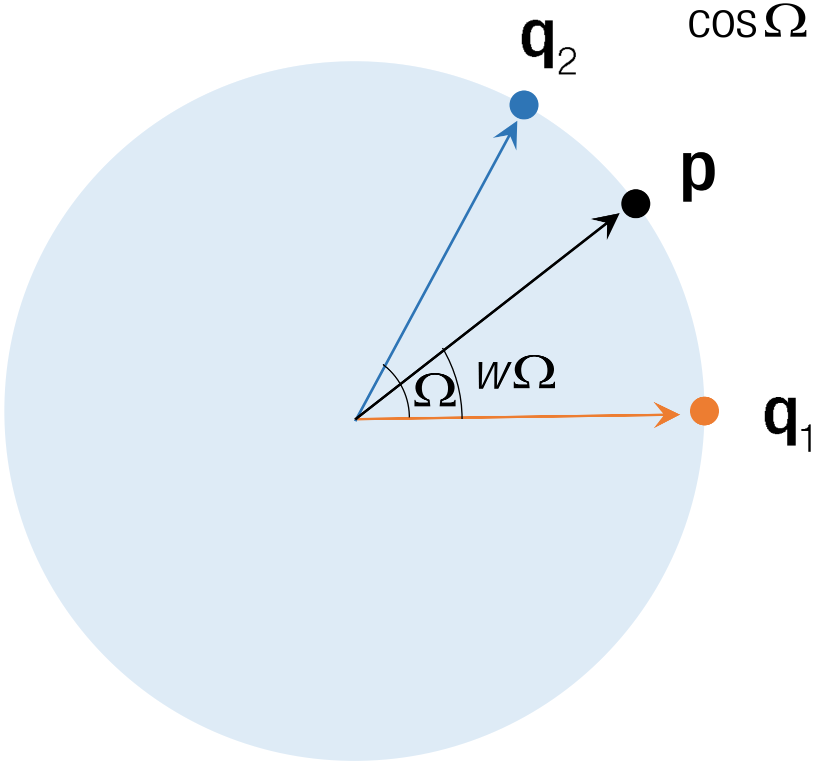
Quaternion Interpolation

$$\mathbf{q} = q_W + iq_X + jq_Y + kq_Z = \begin{bmatrix} q_W \\ q_X \\ q_Y \\ q_Z \end{bmatrix} \qquad q_W^2 + q_X^2 + q_Y^2 + q_Z^2 = 1$$

Interpolation in the sphere

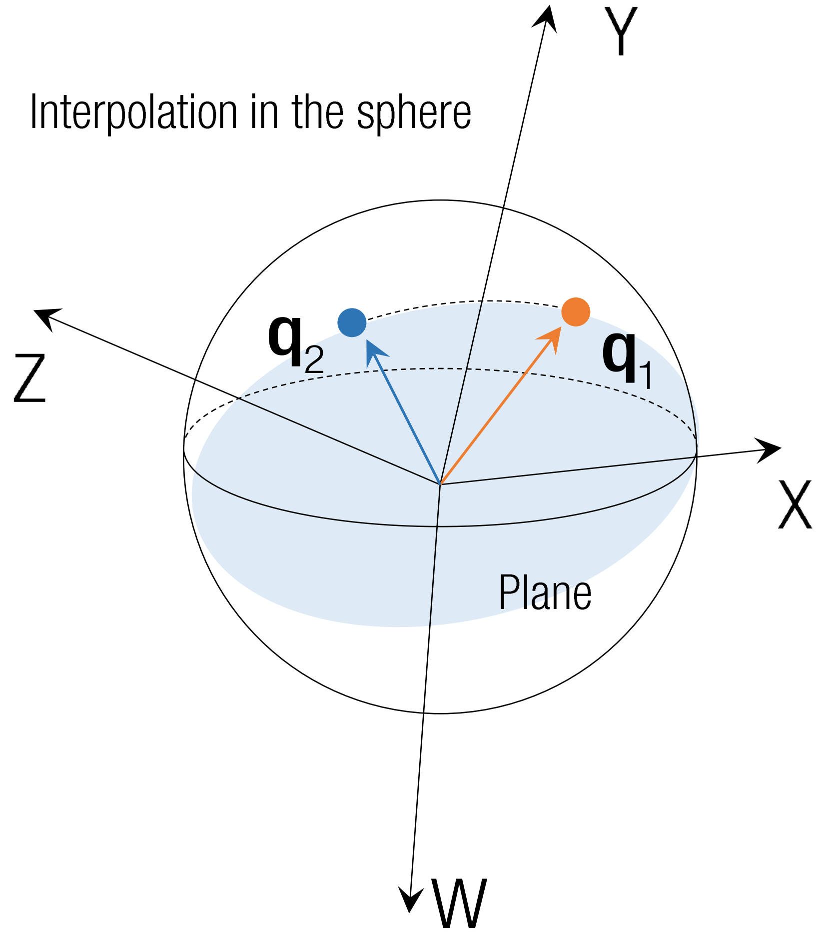


$$\cos\Omega = \mathbf{q}_1 \cdot \mathbf{q}_2$$



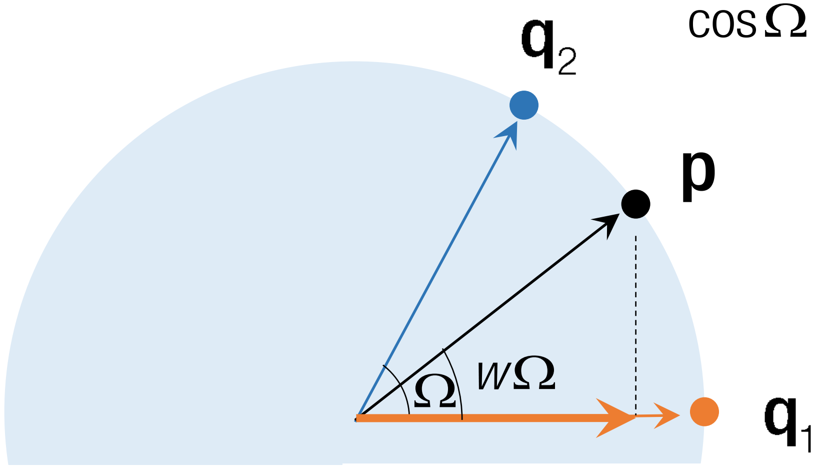
Quaternion Interpolation

Interpolation in the sphere



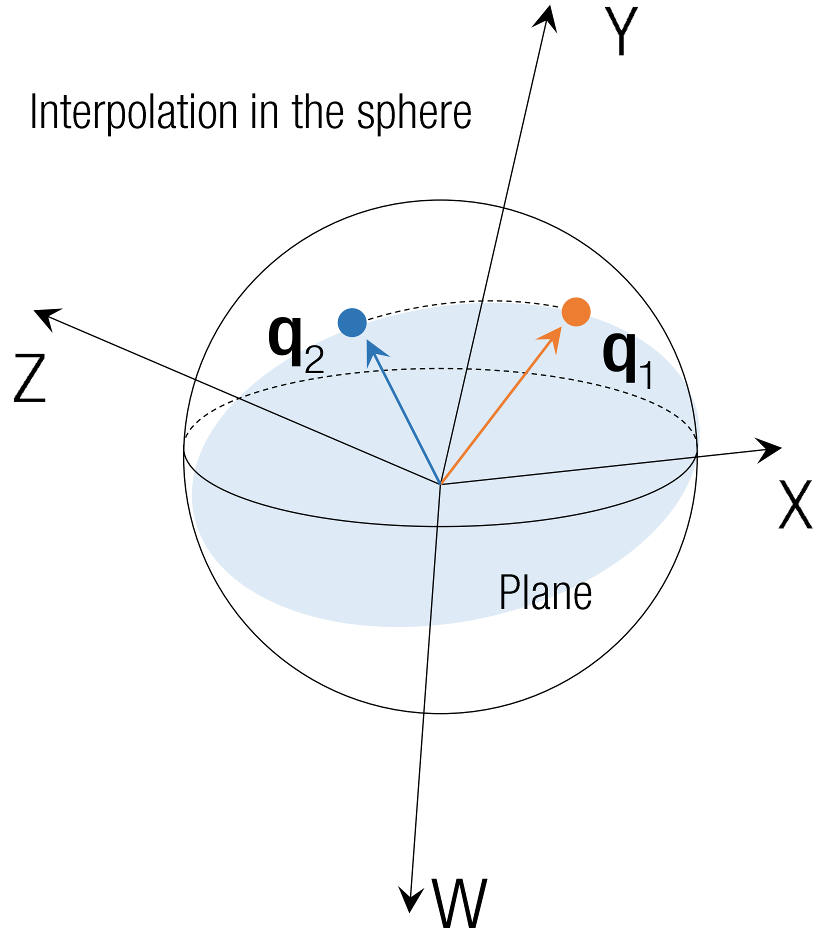
$$\mathbf{q} = q_W + iq_X + jq_Y + kq_Z = \begin{bmatrix} q_W \\ q_X \\ q_Y \\ q_Z \end{bmatrix} \qquad q_W^2 + q_X^2 + q_Y^2 + q_Z^2 = 1$$

$$\cos\Omega = \mathbf{q}_1 \cdot \mathbf{q}_2$$



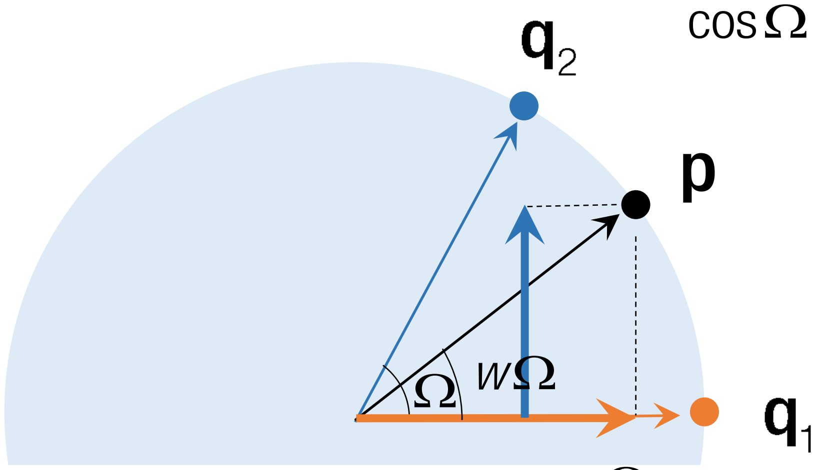
$$\mathbf{p} = \mathbf{q}_1 \cos w\Omega$$

Quaternion Interpolation



$$\mathbf{q} = q_W + iq_X + jq_Y + kq_Z = \begin{bmatrix} q_W \\ q_X \\ q_Y \\ q_Z \end{bmatrix} \qquad q_W^2 + q_X^2 + q_Y^2 + q_Z^2 = 1$$

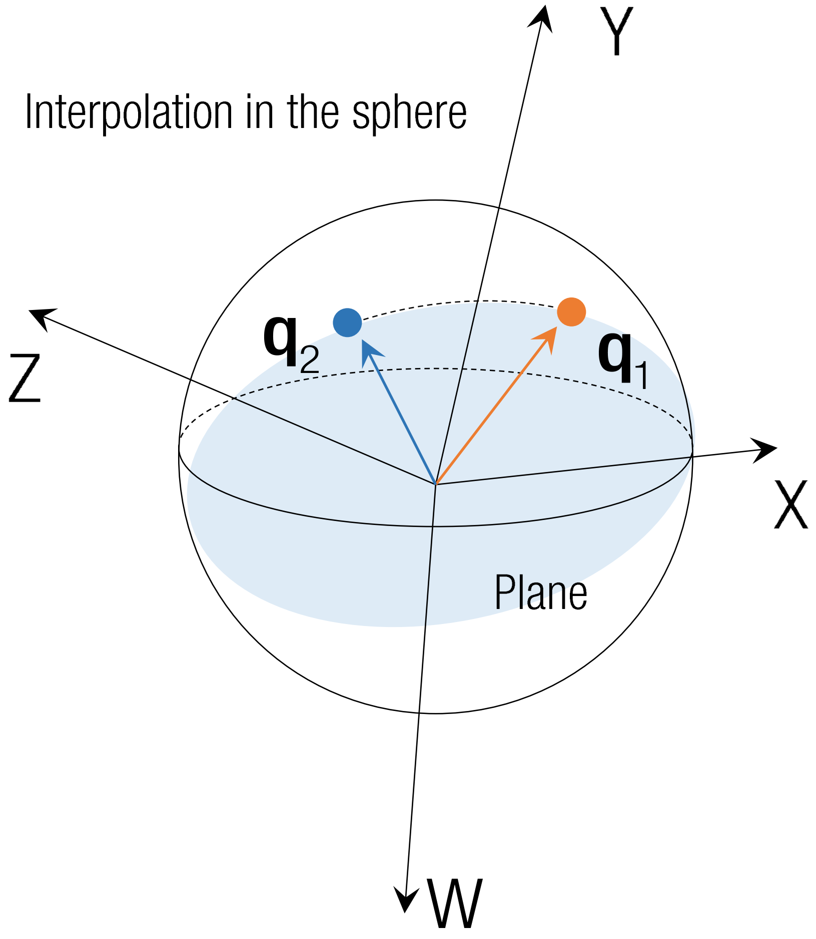
$$\cos\Omega = \mathbf{q}_1 \cdot \mathbf{q}_2$$



$$\mathbf{p} = \mathbf{q}_1 \cos w\Omega + \frac{\mathbf{q}_2 - \mathbf{q}_1 \cos\Omega}{\sin\Omega} \sin w\Omega$$

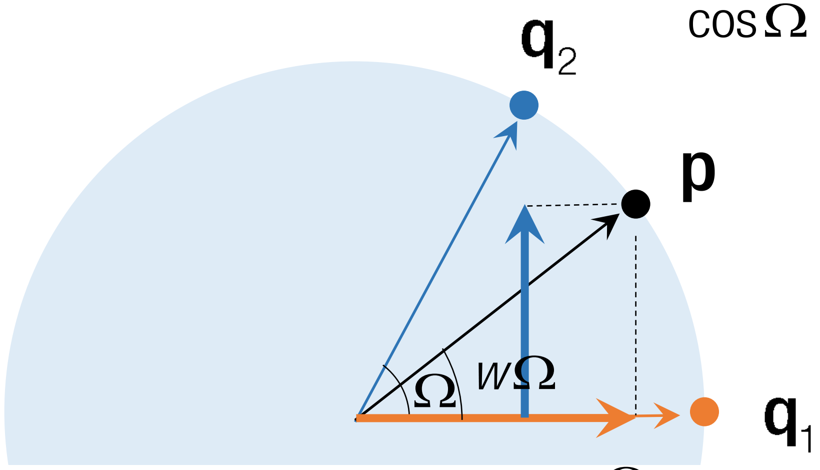
Quaternion Interpolation

Interpolation in the sphere



$$\mathbf{q} = q_W + iq_X + jq_Y + kq_Z = \begin{bmatrix} q_W \\ q_X \\ q_Y \\ q_Z \end{bmatrix} \qquad q_W^2 + q_X^2 + q_Y^2 + q_Z^2 = 1$$

$$\cos\Omega = \mathbf{q}_1 \cdot \mathbf{q}_2$$

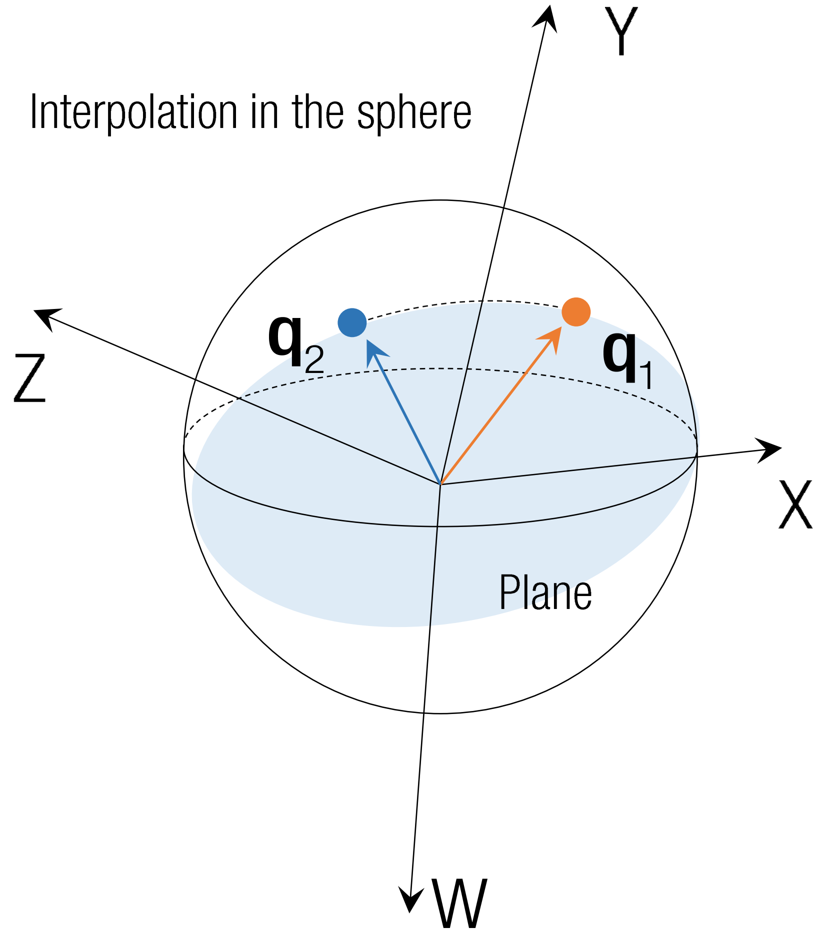


$$\mathbf{p} = \mathbf{q}_1 \cos w\Omega + \frac{\mathbf{q}_2 - \mathbf{q}_1 \cos\Omega}{\sin\Omega} \sin w\Omega$$

$$= \frac{\mathbf{q}_1\left(\sin\Omega \cos w\Omega - \cos\Omega \sin w\Omega\right) + \mathbf{q}_2 \sin w\Omega}{\sin\Omega}$$

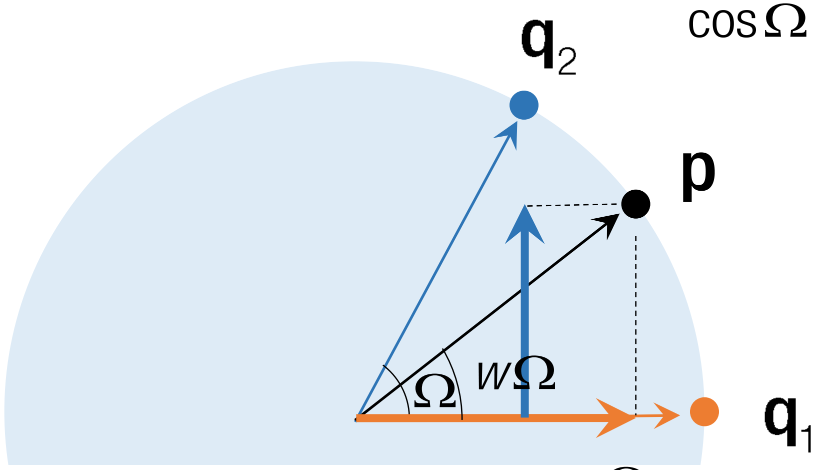
Quaternion Interpolation

Interpolation in the sphere



$$\mathbf{q} = q_W + iq_X + jq_Y + kq_Z = \begin{bmatrix} q_W \\ q_X \\ q_Y \\ q_Z \end{bmatrix} \qquad q_W^2 + q_X^2 + q_Y^2 + q_Z^2 = 1$$

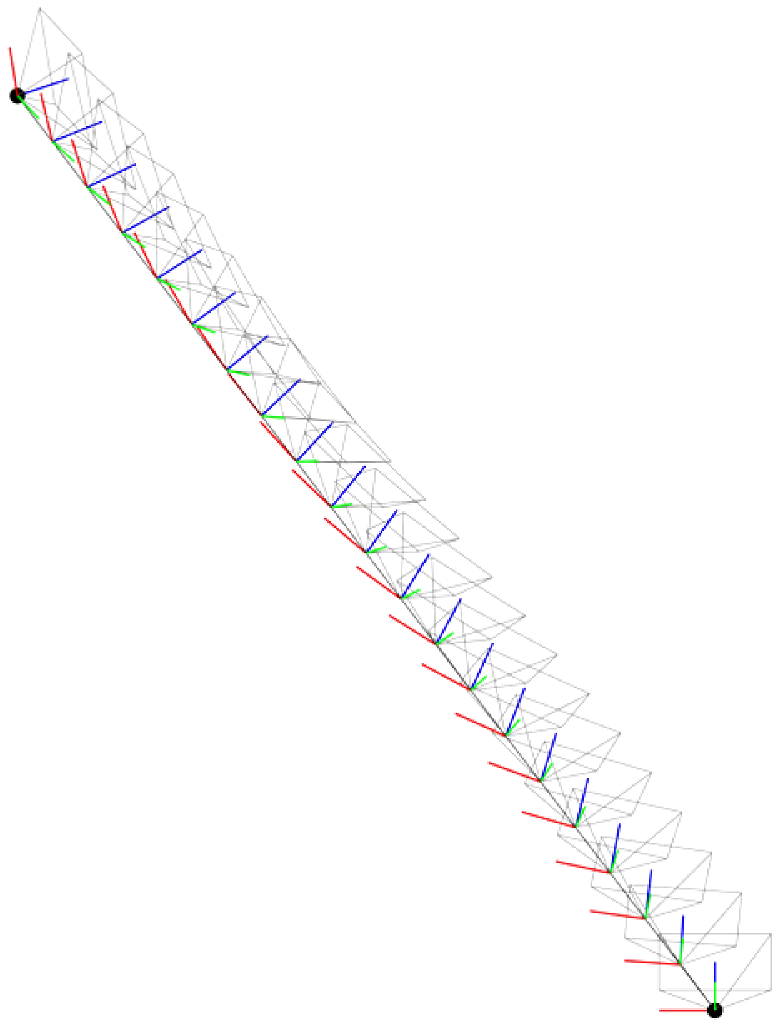
$$\cos\Omega = \mathbf{q}_1 \cdot \mathbf{q}_2$$

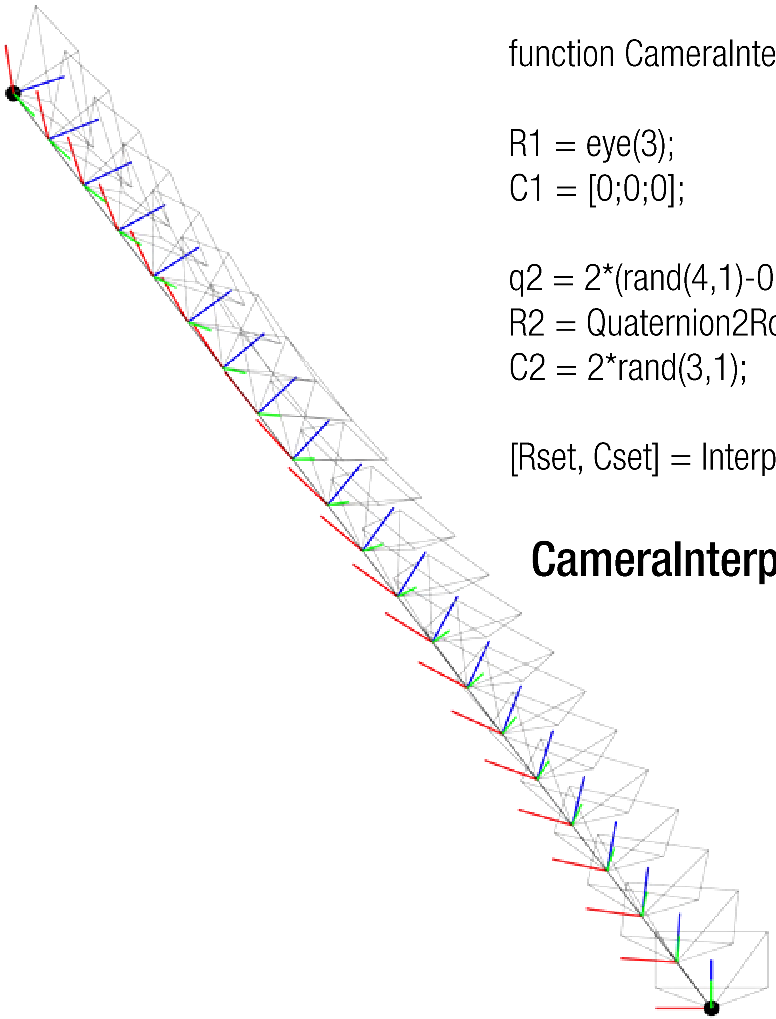


$$\mathbf{p} = \mathbf{q}_1 \cos w\Omega + \frac{\mathbf{q}_2 - \mathbf{q}_1 \cos\Omega}{\sin\Omega} \sin w\Omega$$

$$= \frac{\mathbf{q}_1 \left(\sin\Omega \cos w\Omega - \cos\Omega \sin w\Omega\right) + \mathbf{q}_2 \sin w\Omega}{\sin\Omega}$$

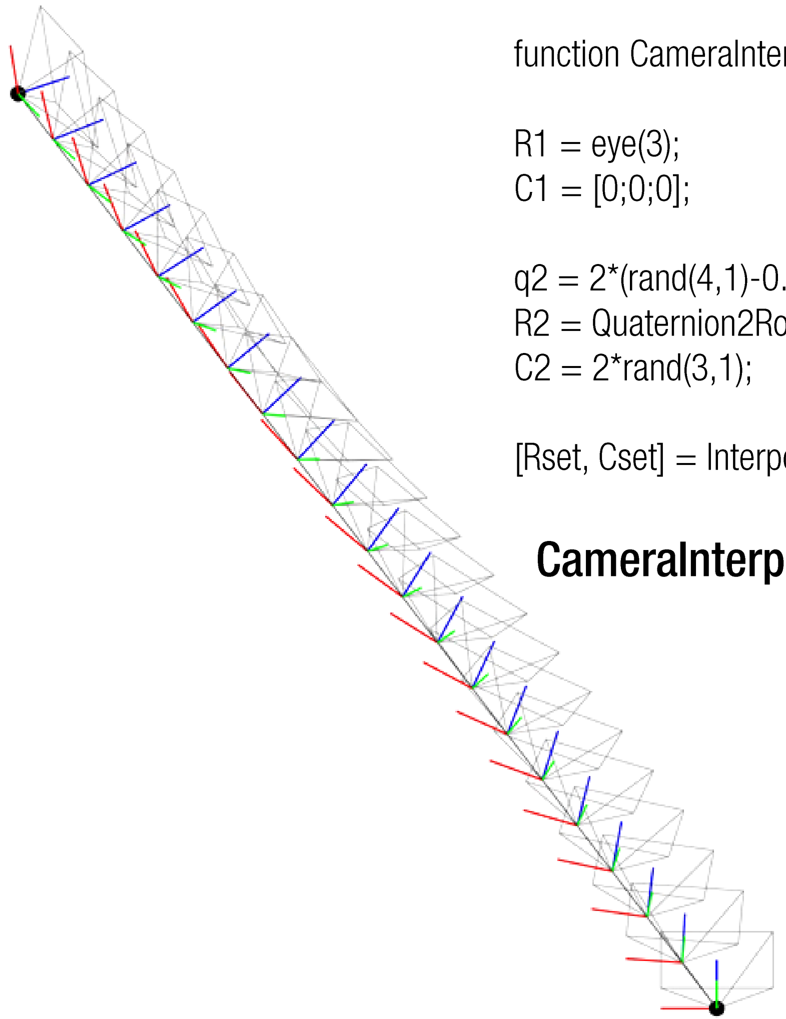
$$= \frac{\mathbf{q}_1 \sin(1-w)\Omega + \mathbf{q}_2 \sin w\Omega}{\sin\Omega}$$





```
function CameraInterpolation

R1 = eye(3);
C1 = [0;0;0];

q2 = 2*(rand(4,1)-0.5);
R2 = Quaternion2Rotation(q2);
C2 = 2*rand(3,1);

[Rset, Cset] = InterpolateCoordinate(R1, C1, R2, C2, 20);
```

CameraInterpolation.m



```
function CameraInterpolation

R1 = eye(3);
C1 = [0;0;0];

q2 = 2*(rand(4,1)-0.5);
R2 = Quaternion2Rotation(q2);
C2 = 2*rand(3,1);

[Rset, Cset] = InterpolateCoordinate(R1, C1, R2, C2, 20);
```

CameraInterpolation.m

```
function [Rset, Cset] = InterpolateCoordinate(R1, C1, R2, C2, n)

Cx = linspace(C1(1), C2(1), n+1);
Cy = linspace(C1(2), C2(2), n+1);
Cz = linspace(C1(3), C2(3), n+1);

Cset = [Cx; Cy; Cz];

w = 0 : 1/n : 1;
q1 = Rotation2Quaternion(R1);
q2 = Rotation2Quaternion(R2);

omega = acos(q1'*q2);
for i = 1 : length(w)
    q = sin(omega*(1-w(i)))/sin(omega) * q1 + sin(omega*w(i))/sin(omega) * q2;
    Rset{i} = Quaternion2Rotation(q);
end
```

InterpolateCoordinate.m

View Interpolation

Looking left

Looking right

View Interpolation (HW #3)