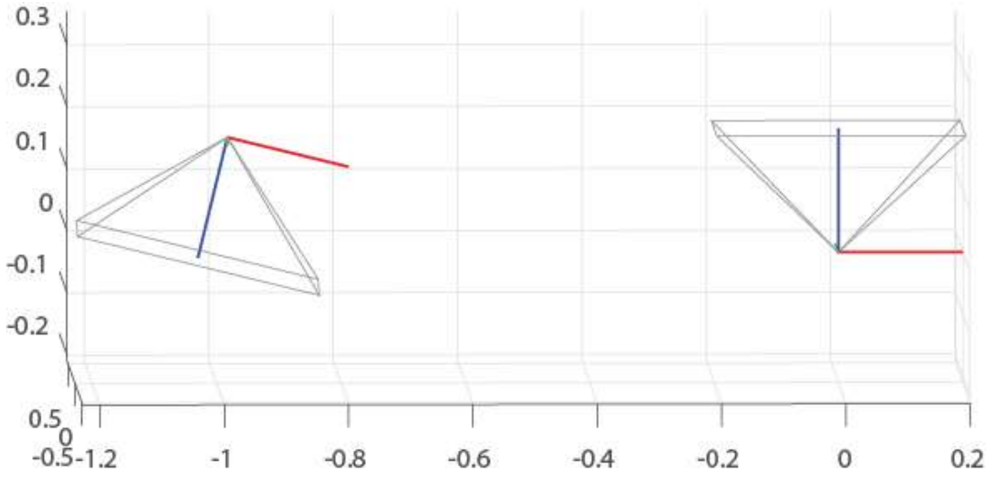
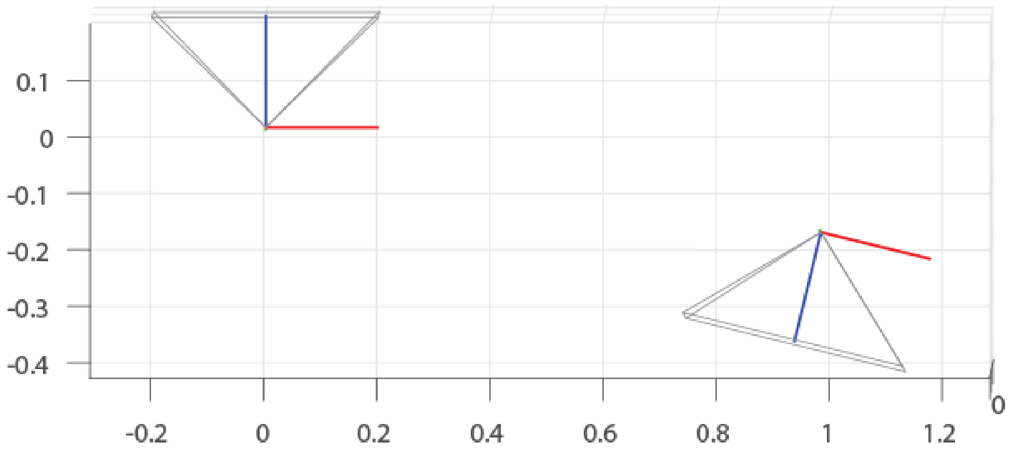
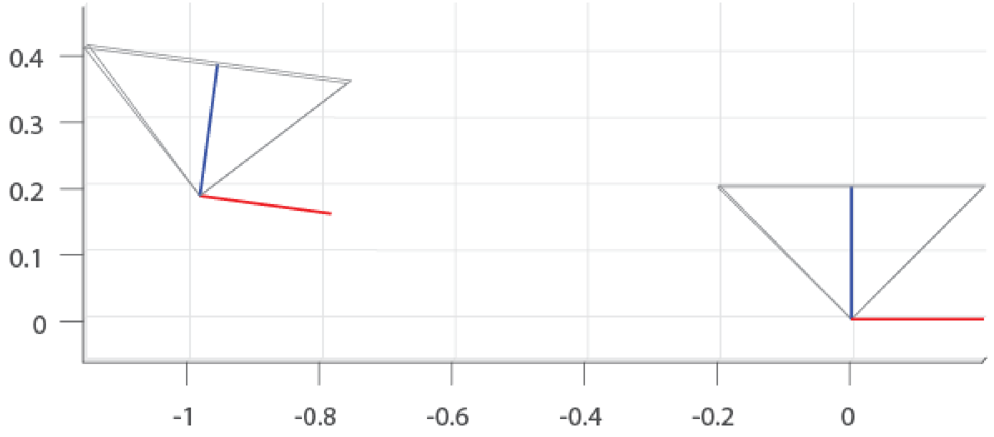
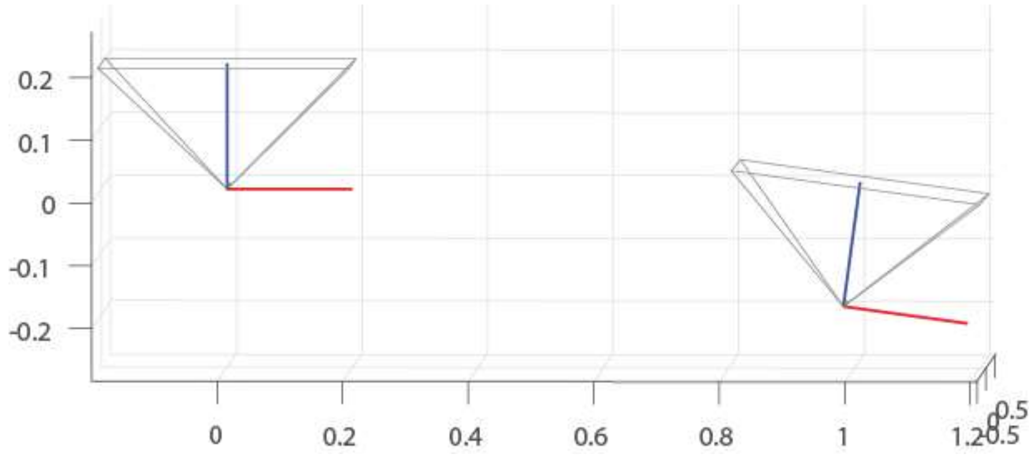
Triangulation

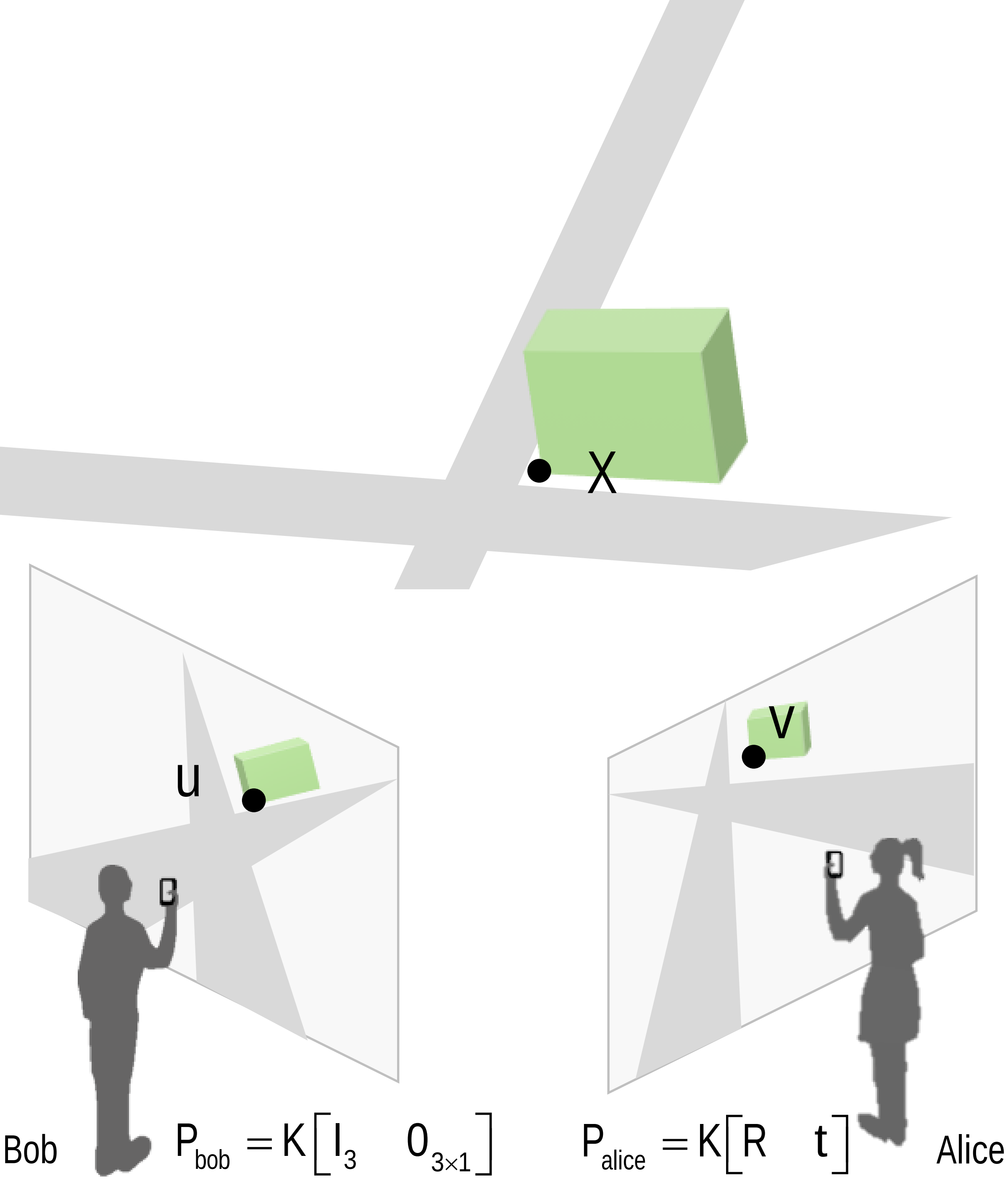
How to Disambiguate?





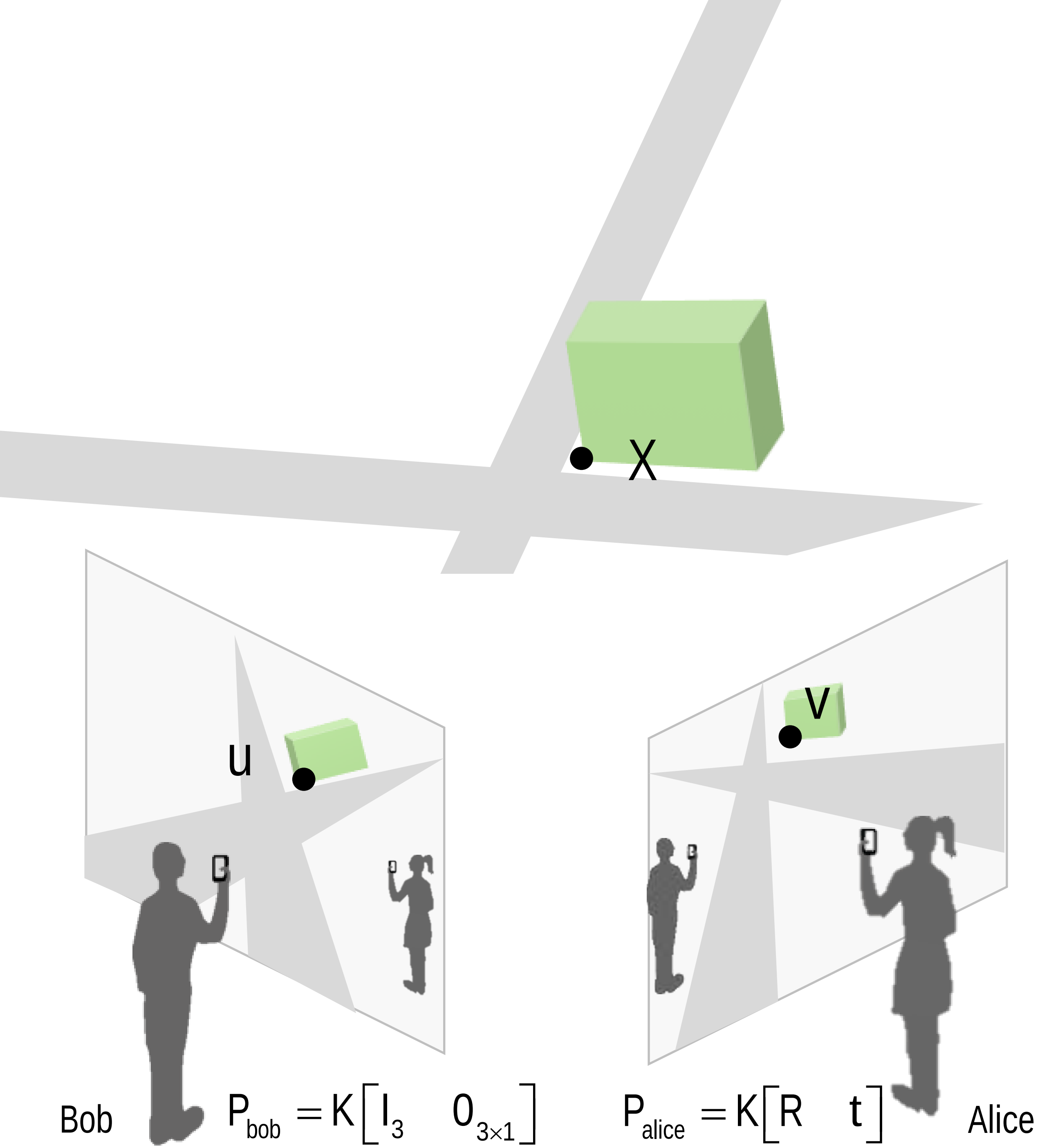






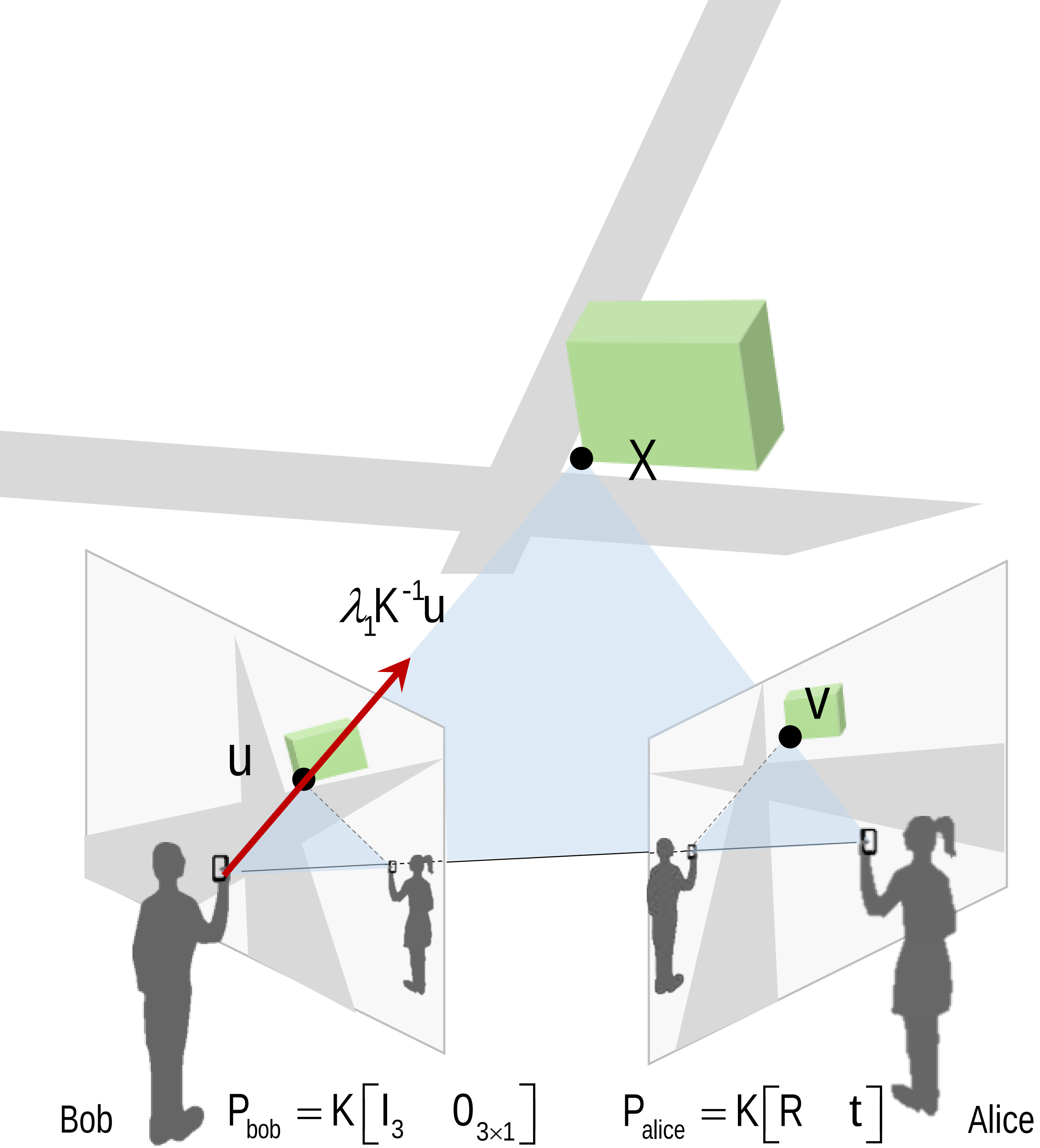
$$P_{bob} = K\begin{bmatrix} I_3 & 0_{3\times1} \end{bmatrix}$$

$$P_{alice} = K\begin{bmatrix} R & t \end{bmatrix}$$



$$P_{bob} = K\begin{bmatrix} I_3 & 0_{3\times1} \end{bmatrix}$$

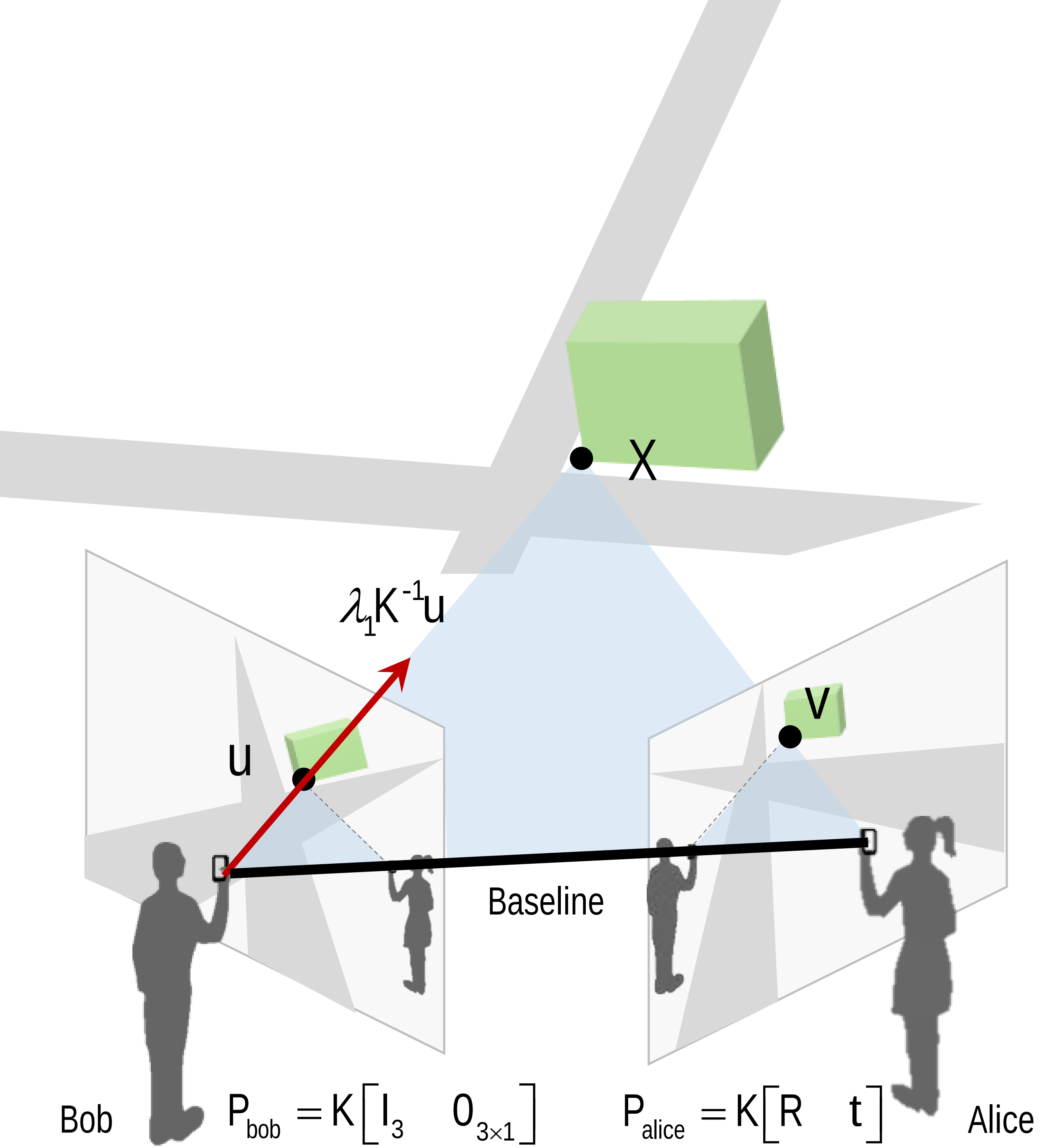
$$P_{alice} = K\begin{bmatrix} R & t \end{bmatrix}$$



$$P_{bob} = K\begin{bmatrix} I_3 & 0_{3\times1} \end{bmatrix}$$

$$P_{alice} = K\begin{bmatrix} R & t \end{bmatrix}$$

→ $\lambda_1 K^{-1}u$



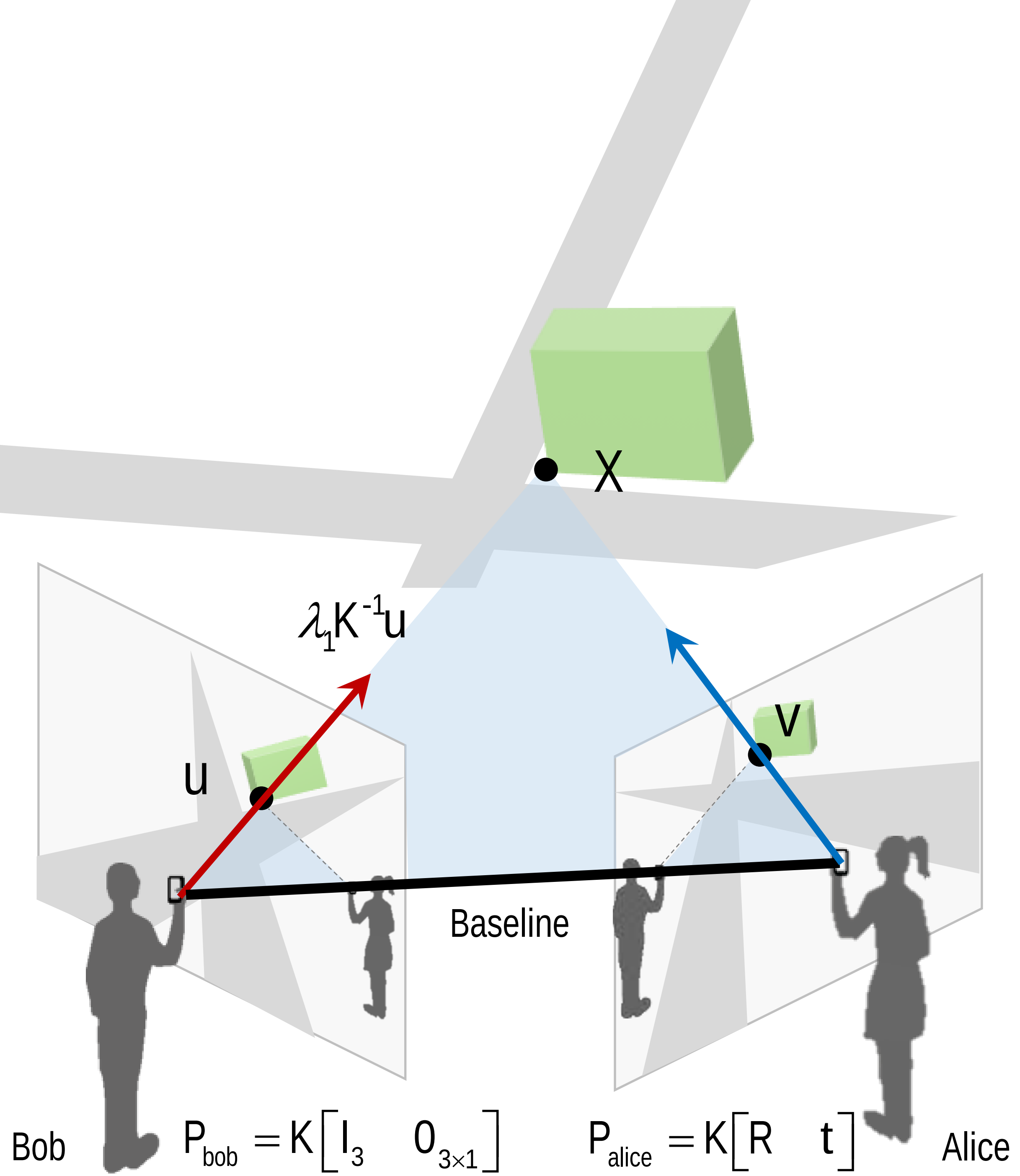
$$P_{bob} = K\begin{bmatrix} I_3 & 0_{3\times1} \end{bmatrix}$$

$$P_{alice} = K\begin{bmatrix} R & t \end{bmatrix}$$

→ $\lambda_1 K^{-1}u$

— $\underline{-R^T t}$

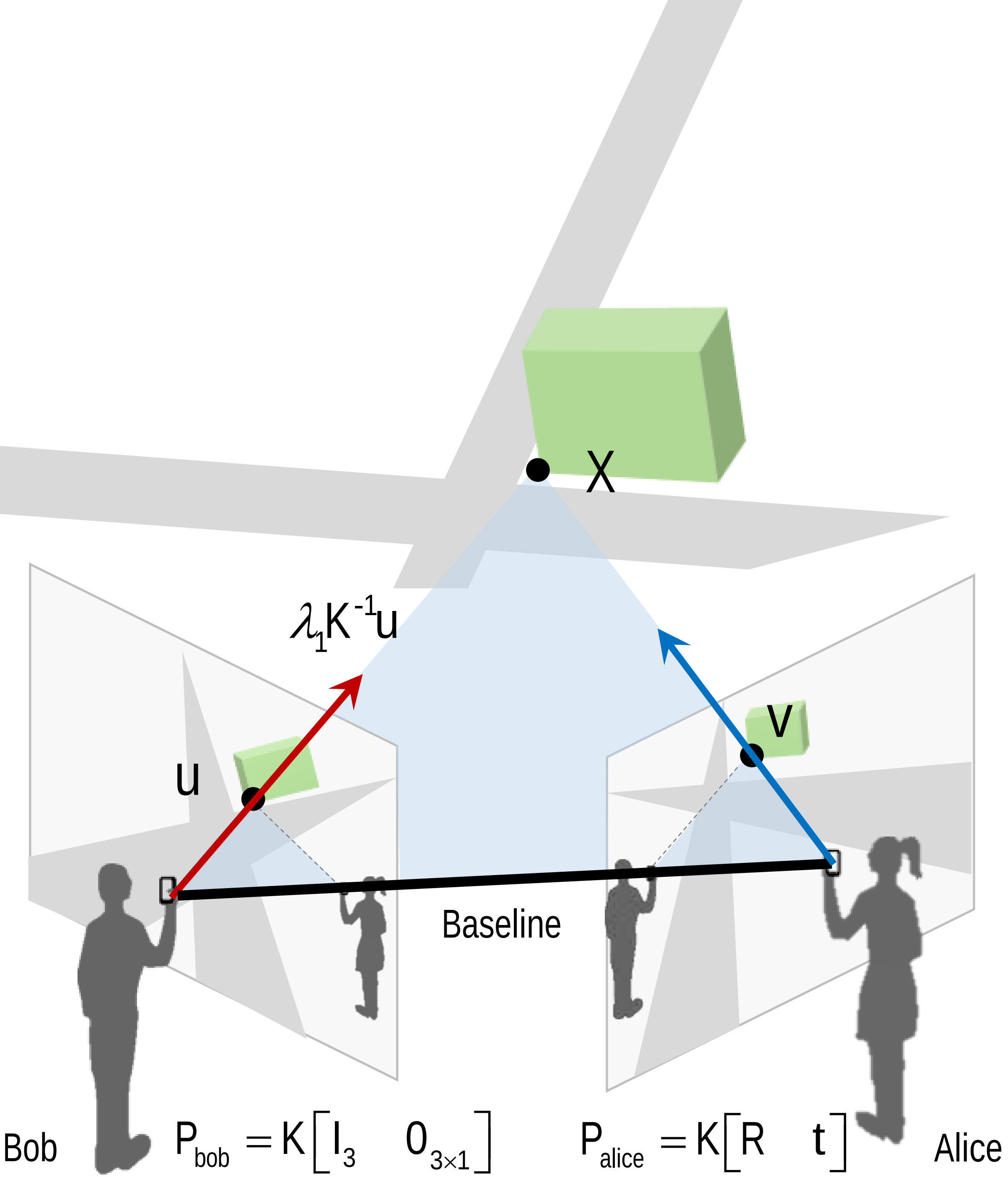
Alice's camer location



→ $\lambda_1 K^{-1}u$

— $\underbrace{-R^T t}_{\text{Alice's camer location}}$

→ $\underbrace{\lambda_2 R^T K^{-1} v}_{\text{Direction}} \underbrace{-R^T t}_{\text{Alice's camer location}}$



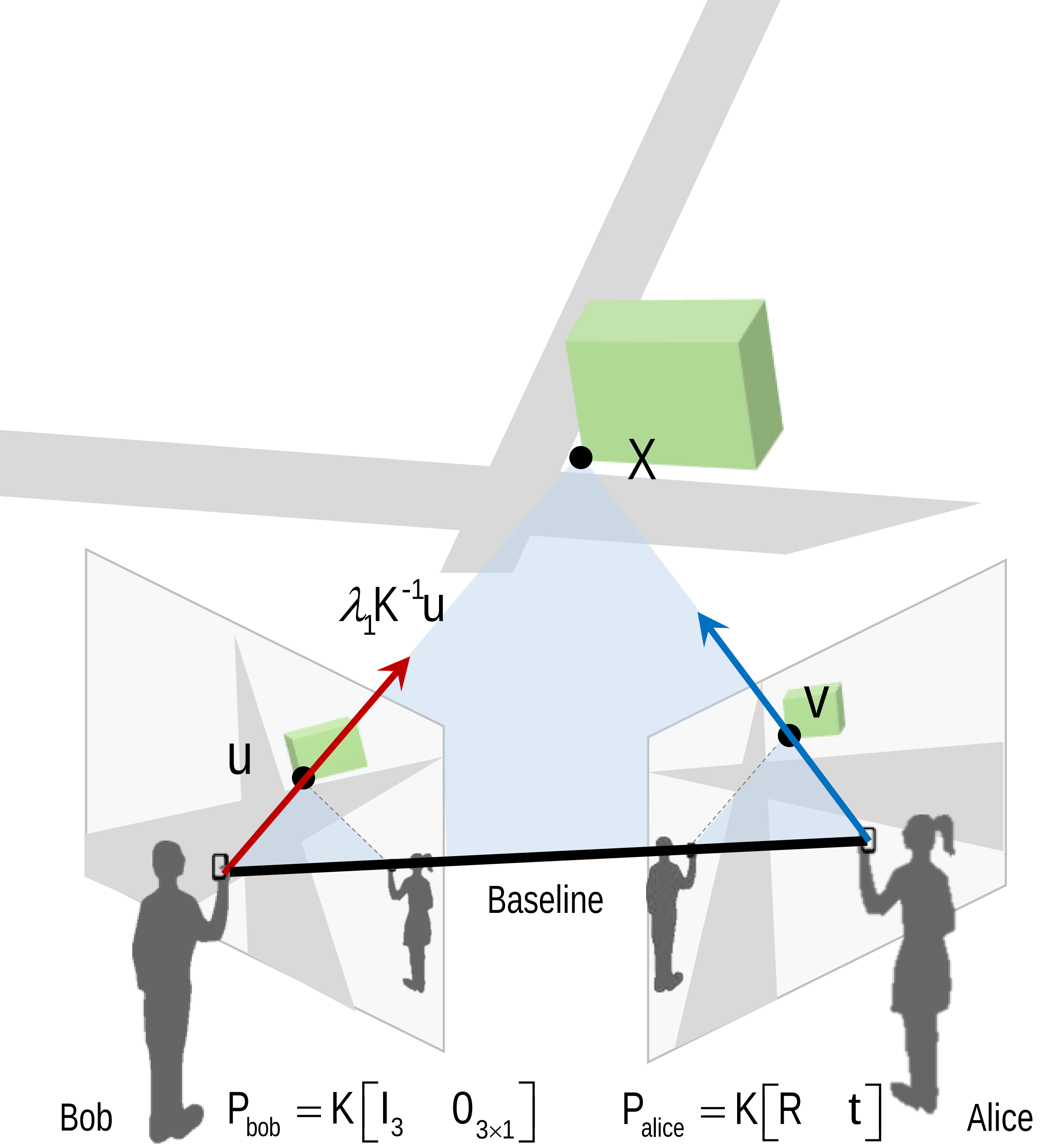
→ $\lambda_1 K^{-1}u$

— $\underbrace{-R^T t}_{\text{Alice's camer location}}$

→ $\underbrace{\lambda_2 R^T K^{-1}v}_{\text{Direction}} \underbrace{- R^T t}_{\text{Alice's camer location}}$

$$\underbrace{X = \lambda_1 K^{-1}u = \lambda_2 R^T K^{-1}v - R^T t}_{\text{3D point}}$$

# of unknowns: 2
# of equations: 3



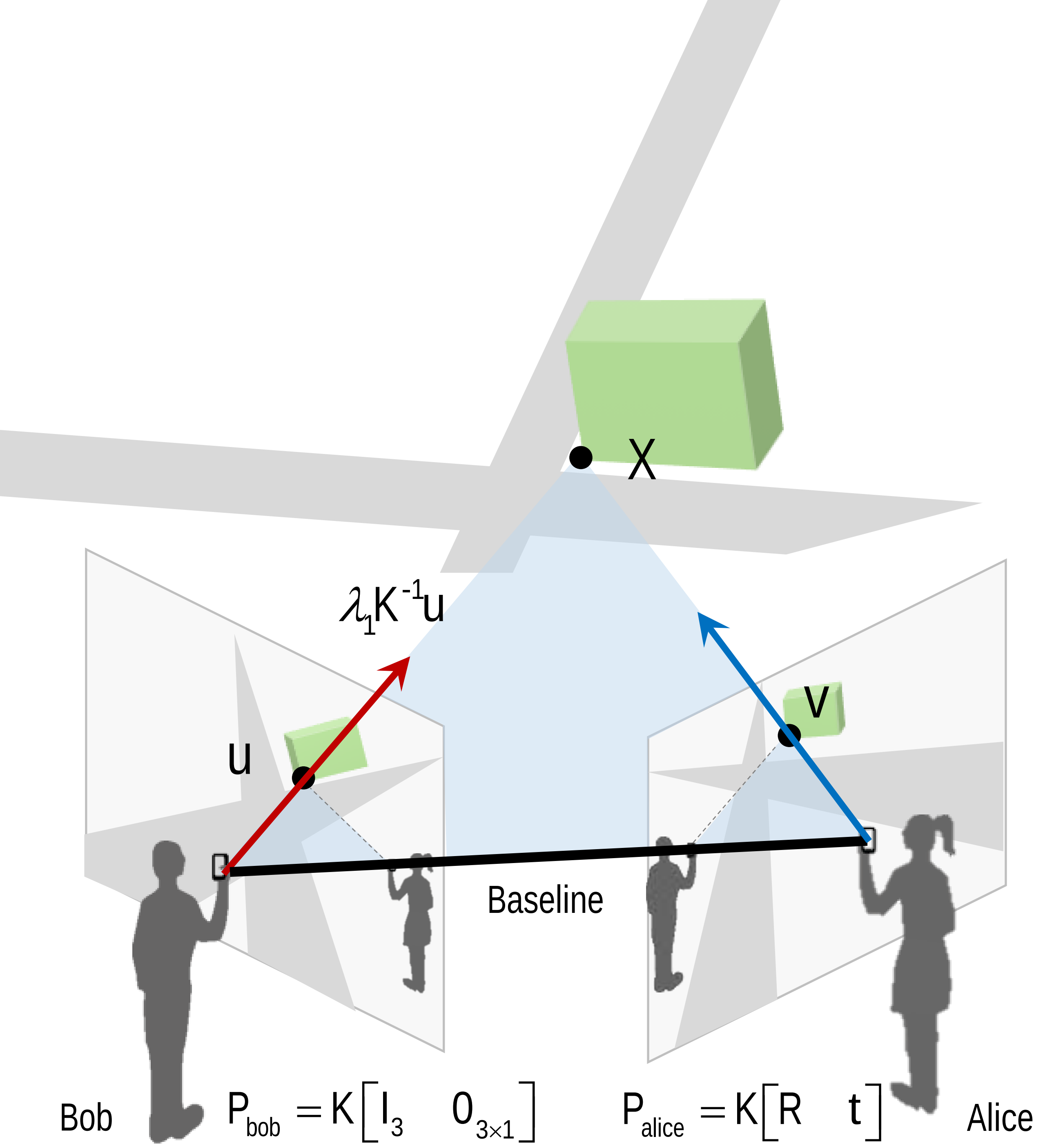
→ $\lambda_1 K^{-1}u$

— $\underbrace{-R^T t}_{\text{Alice's camer location}}$

→ $\underbrace{\lambda_2 R^T K^{-1} v}_{\text{Direction}} \underbrace{-R^T t}_{\text{Alice's camer location}}$

$$\underbrace{X = \lambda_1 K^{-1}u = \lambda_2 R^T K^{-1} v - R^T t}_{\text{3D point}}$$

$$\rightarrow \lambda_1 K^{-1}u - \lambda_2 R^T K^{-1} v = -R^T t$$



$$P_{bob} = K\begin{bmatrix} I_3 & 0_{3\times1} \end{bmatrix} \qquad P_{alice} = K\begin{bmatrix} R & t \end{bmatrix}$$

→ $\lambda_1 K^{-1} u$

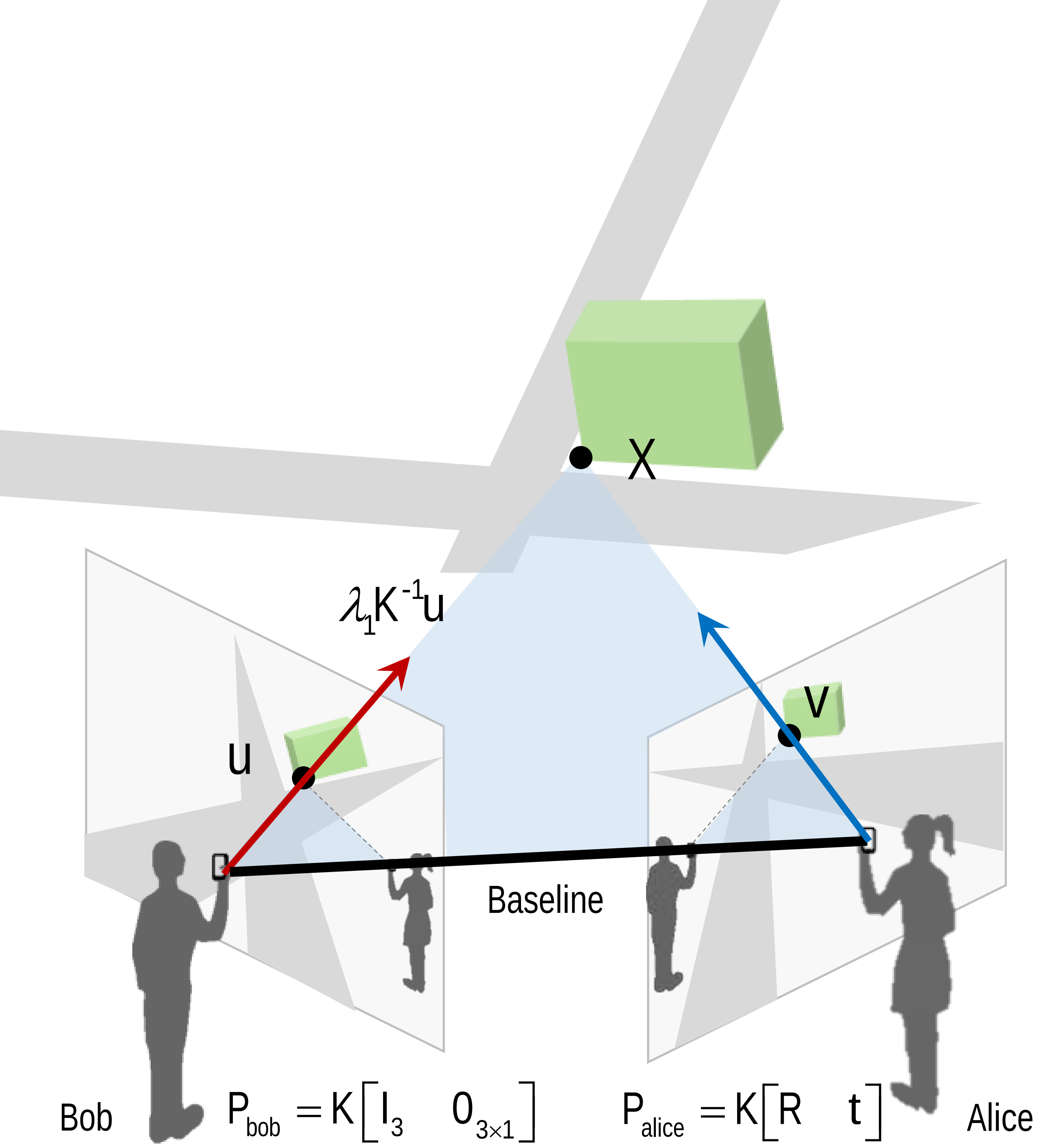
— $\underbrace{-R^T t}_{\text{Alice's camer location}}$

→ $\underbrace{\lambda_2 R^T K^{-1} v}_{\text{Direction}} \underbrace{- R^T t}_{\text{Alice's camer location}}$

$$\underbrace{X = \lambda_1 K^{-1} u = \lambda_2 R^T K^{-1} v - R^T t}_{\text{3D point}}$$

$$\rightarrow \lambda_1 K^{-1} u - \lambda_2 R^T K^{-1} v = -R^T t$$

$$\rightarrow \begin{bmatrix} K^{-1} u & -R^T K^{-1} v \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = -R^T t$$



→ $\lambda_1 K^{-1}u$

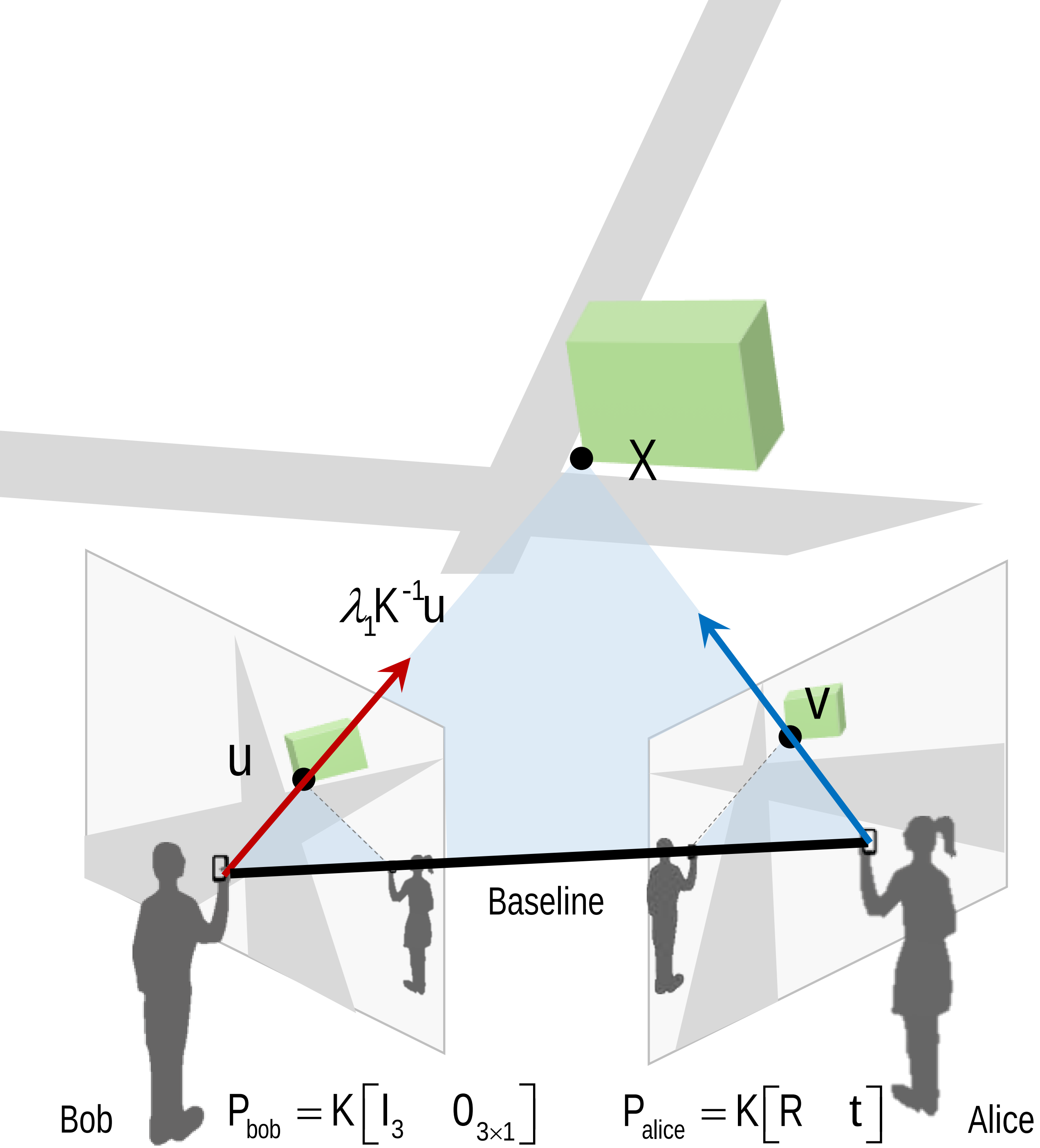
— $\underbrace{-R^Tt}_{\text{Alice's camer location}}$

→ $\underbrace{\lambda_2 R^T K^{-1}v}_{\text{Direction}}\ \underbrace{-R^Tt}_{\text{Alice's camer location}}$

$$\underbrace{X = \lambda_1 K^{-1}u = \lambda_2 R^T K^{-1}v - R^Tt}_{\text{3D point}}$$

$$\rightarrow \lambda_1 K^{-1}u - \lambda_2 R^T K^{-1}v = -R^Tt$$

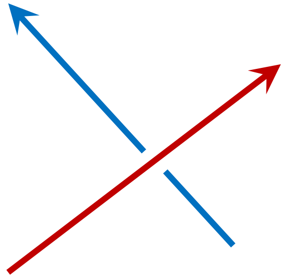
$$\rightarrow \underbrace{\overset{A}{\begin{bmatrix} K^{-1}u & -R^T K^{-1}v\end{bmatrix}}}_{3x2}\overset{x}{\begin{bmatrix}\lambda_1 \\ \lambda_2\end{bmatrix}} = \overset{b}{-R^Tt}$$



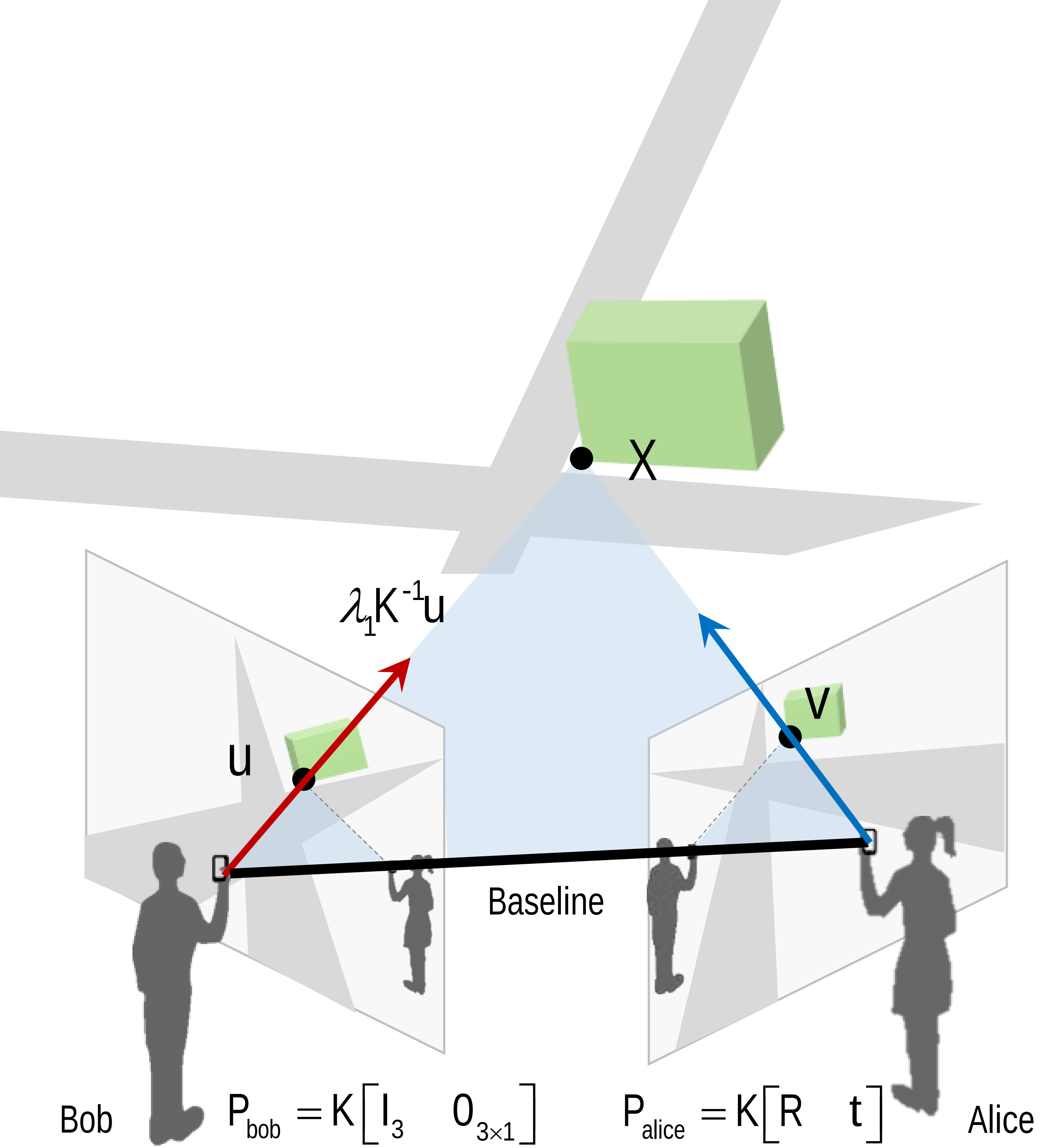
$$P_{bob} = K\begin{bmatrix} I_3 & 0_{3\times1}\end{bmatrix}$$

$$P_{alice} = K\begin{bmatrix} R & t\end{bmatrix}$$

What if two does not meet at a point?

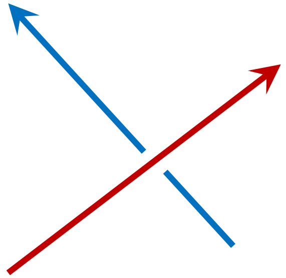


$$\rightarrow \underset{3\text{x}2}{\overset{A}{\begin{bmatrix} K^{-1}u & -R^{T}K^{-1}v\end{bmatrix}}} \overset{x}{\begin{bmatrix}\lambda_1\\ \lambda_2\end{bmatrix}} = \overset{b}{-R^{T}t}$$



$$P_{bob} = K\begin{bmatrix} I_3 & 0_{3\times1}\end{bmatrix} \qquad P_{alice} = K\begin{bmatrix} R & t\end{bmatrix}$$

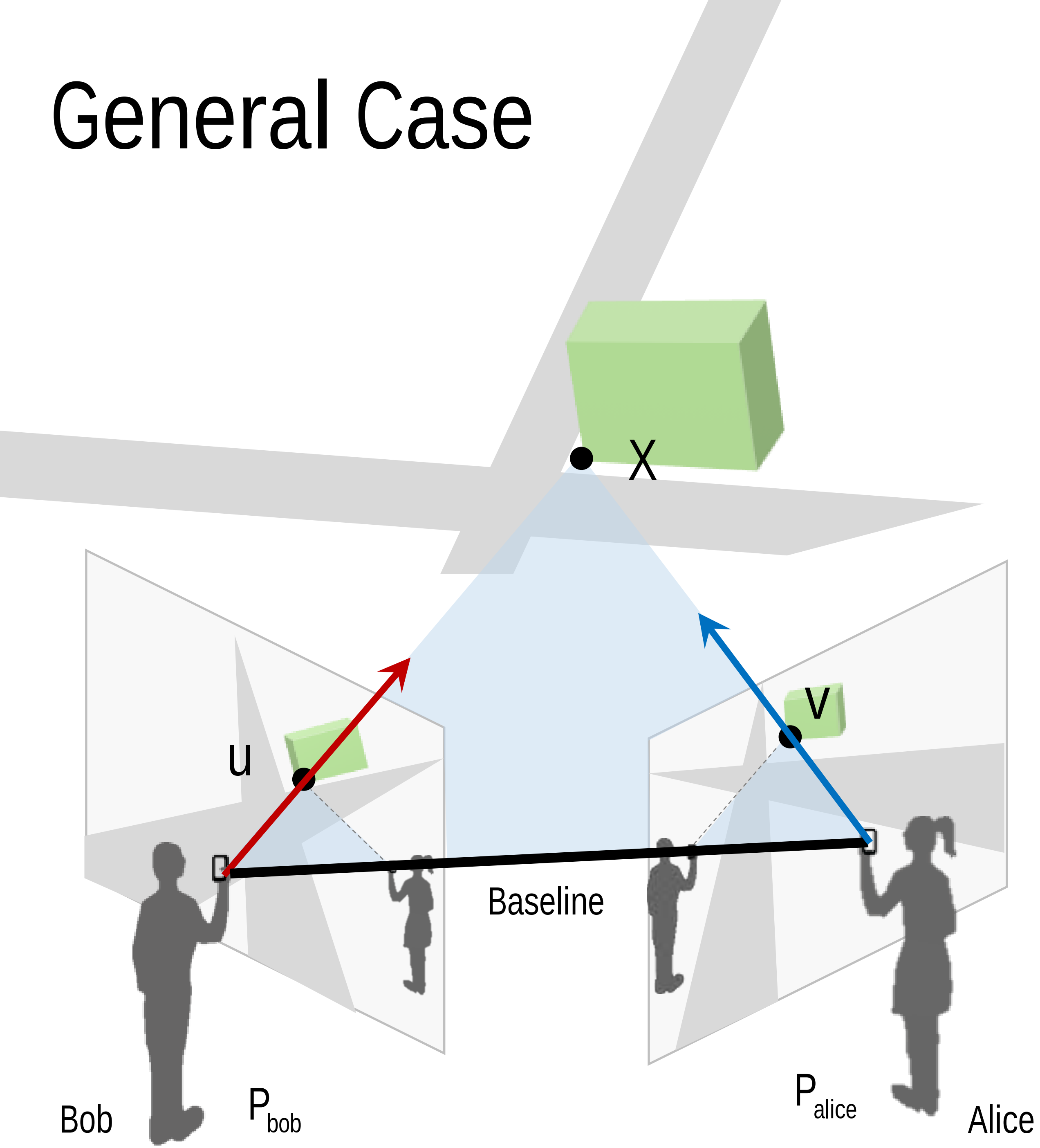
What if two does not meet at a point?



Least square solution finds *somewhere* in the middle.

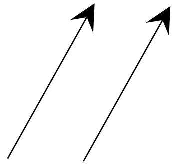
$$\rightarrow \underset{3\mathrm{x}2}{\overset{A}{\begin{bmatrix} K^{-1}u & -R^{T}K^{-1}v \end{bmatrix}}} \overset{x}{\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}} = \overset{b}{-R^{T}t}$$

General Case



General camera pose

$$\lambda_1 \begin{bmatrix} u \\ 1 \end{bmatrix} = P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix}$$

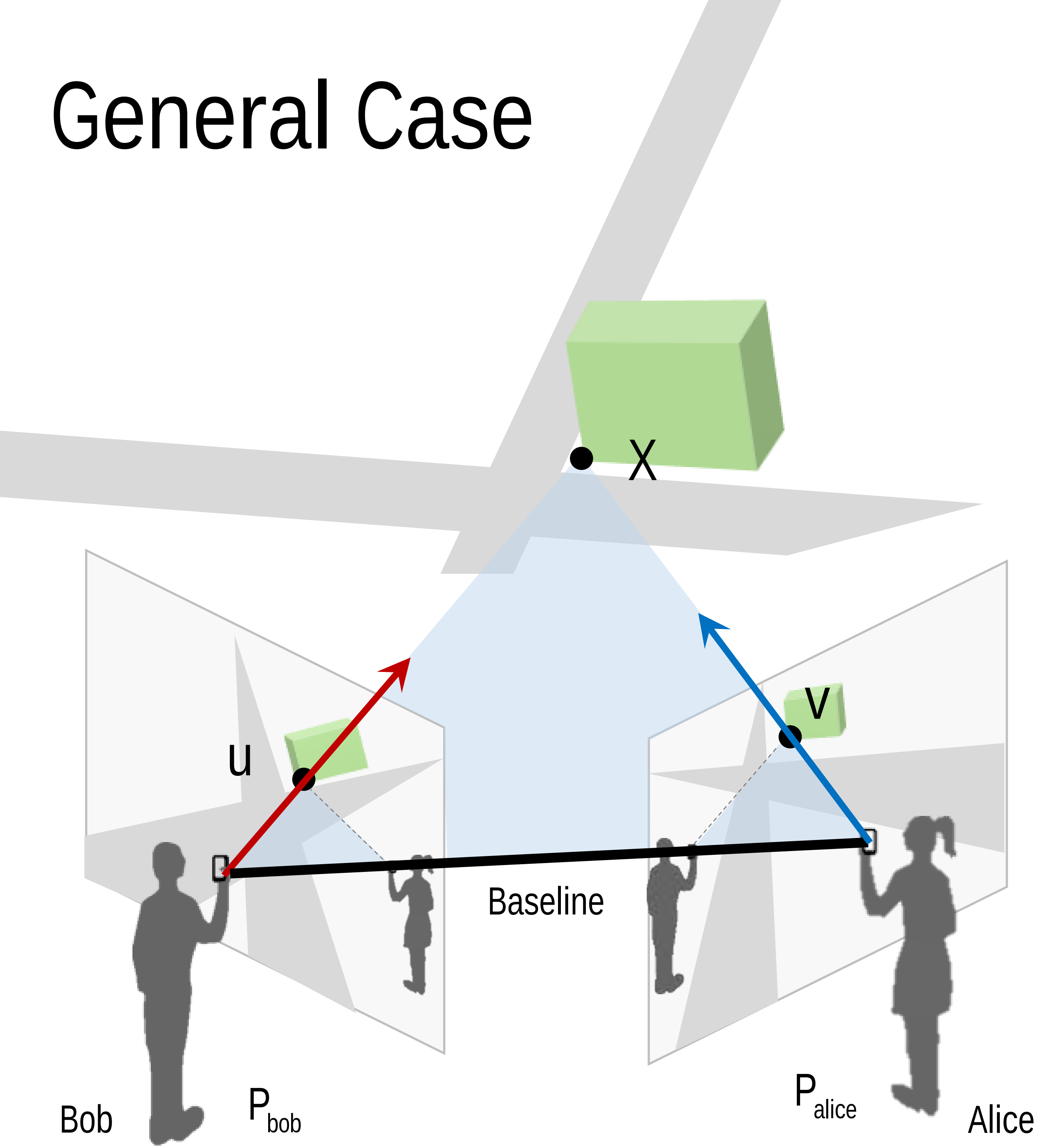


Two 3D vectors are parallel.

$$\rightarrow \begin{bmatrix} u \\ 1 \end{bmatrix} \times P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

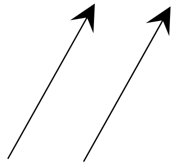
$$\rightarrow \underbrace{\begin{bmatrix} u \\ 1 \end{bmatrix}_\times}_{\text{Skew-symmetric matrix}} P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

General Case



General camera pose

$$\lambda_1 \begin{bmatrix} u \\ 1 \end{bmatrix} = P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix}$$



Two 3D vectors are parallel.

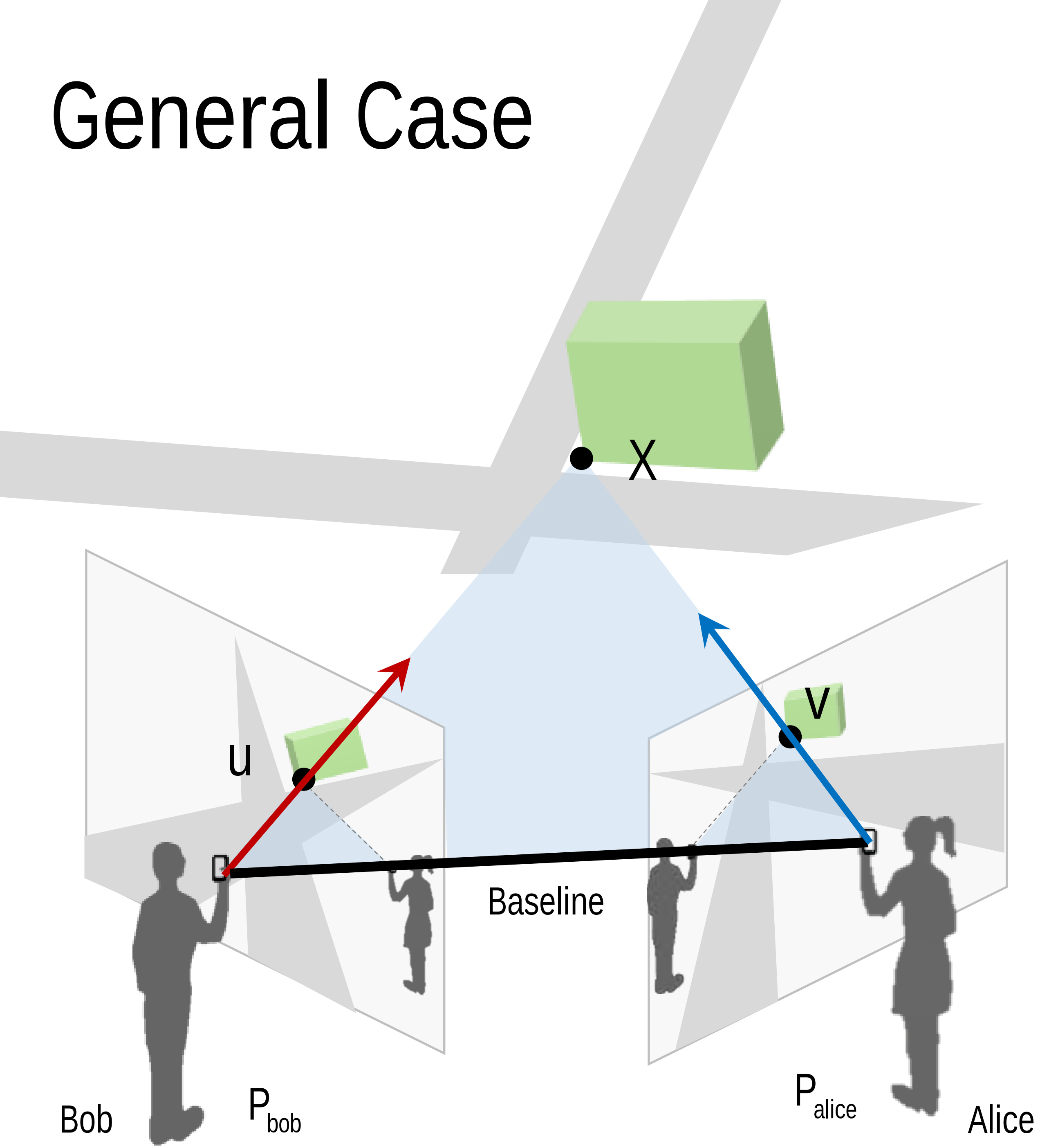
$$\longrightarrow \begin{bmatrix} u \\ 1 \end{bmatrix} \times P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

$$\longrightarrow \underbrace{\begin{bmatrix} u \\ 1 \end{bmatrix}_{\times}}_{\text{Skew-symmetric matrix}} P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

🟧 : Knowns

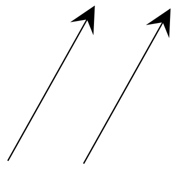
🟦 : Unknowns

General Case



General camera pose

$$\lambda_1 \begin{bmatrix} u \\ 1 \end{bmatrix} = P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix}$$



Two 3D vectors are parallel.

$$\longrightarrow \begin{bmatrix} u \\ 1 \end{bmatrix} \times P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

$$\longrightarrow \begin{bmatrix} u \\ 1 \end{bmatrix}_\times P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

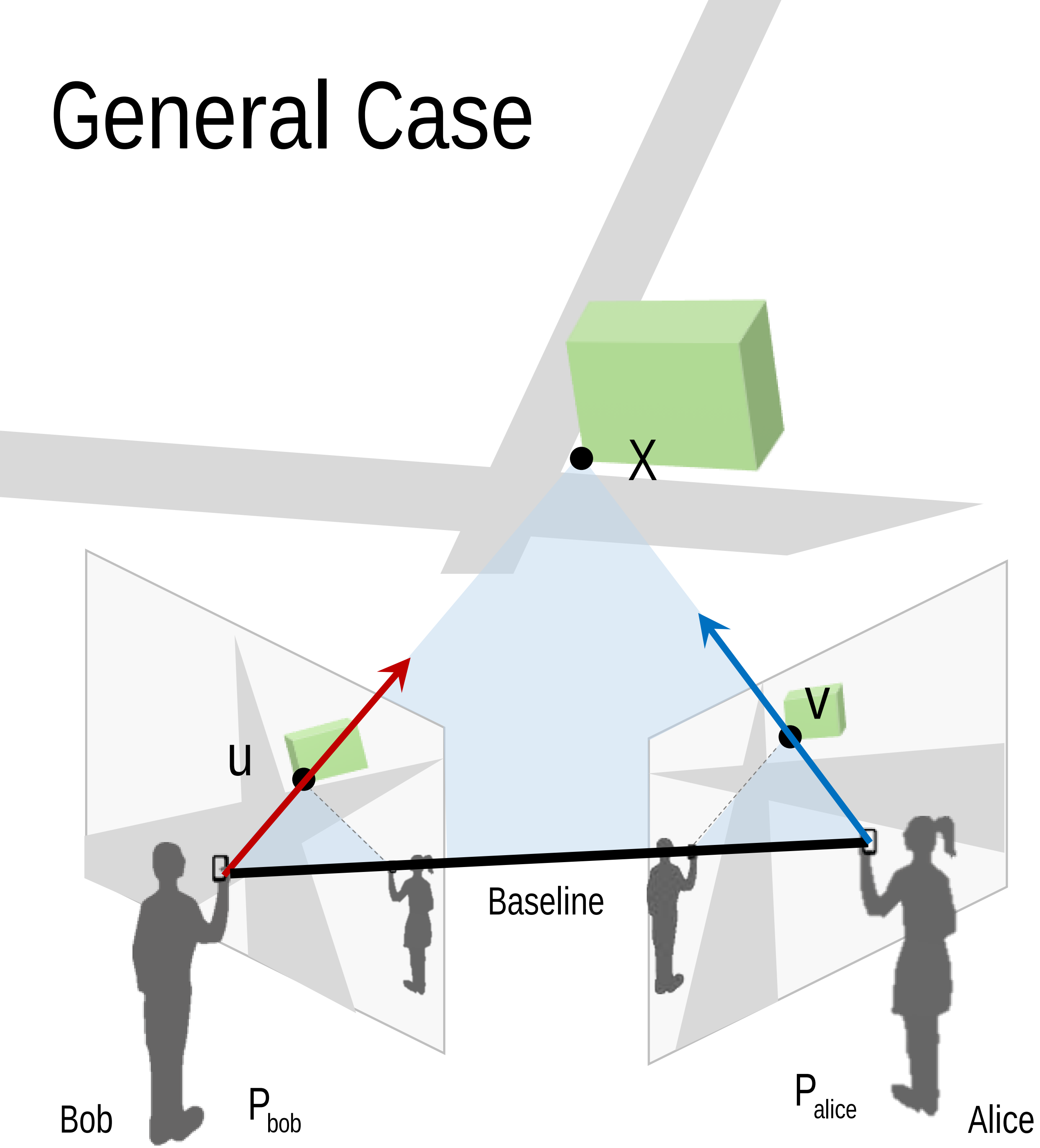
3x4

🟧 : Knowns

🟦 : Unknowns

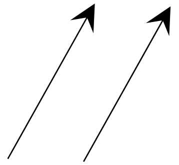
Can we solve for X? (single view reconstruction)
Why not?

General Case



General camera pose

$$\lambda_1 \begin{bmatrix} u \\ 1 \end{bmatrix} = P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix}$$



Two 3D vectors are parallel.

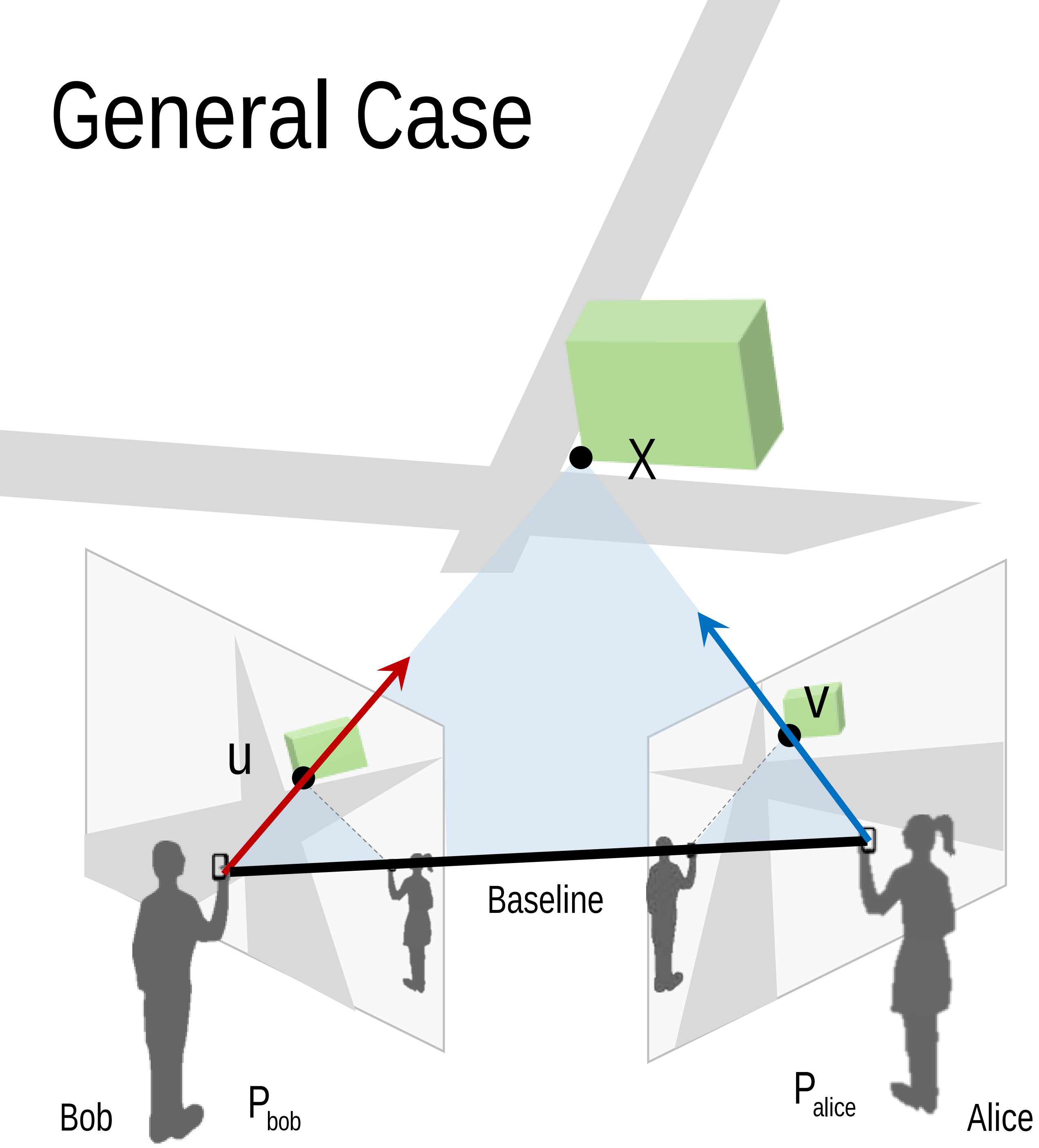
$$\longrightarrow \begin{bmatrix} u \\ 1 \end{bmatrix} \times P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

$$\longrightarrow \underbrace{\begin{bmatrix} u \\ 1 \end{bmatrix}_{\times} P_{bob}}_{2x4} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

🟧 : Knowns

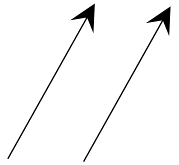
🟦 : Unknowns

General Case



General camera pose

$$\lambda_1 \begin{bmatrix} u \\ 1 \end{bmatrix} = P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix}$$



Two 3D vectors are parallel.

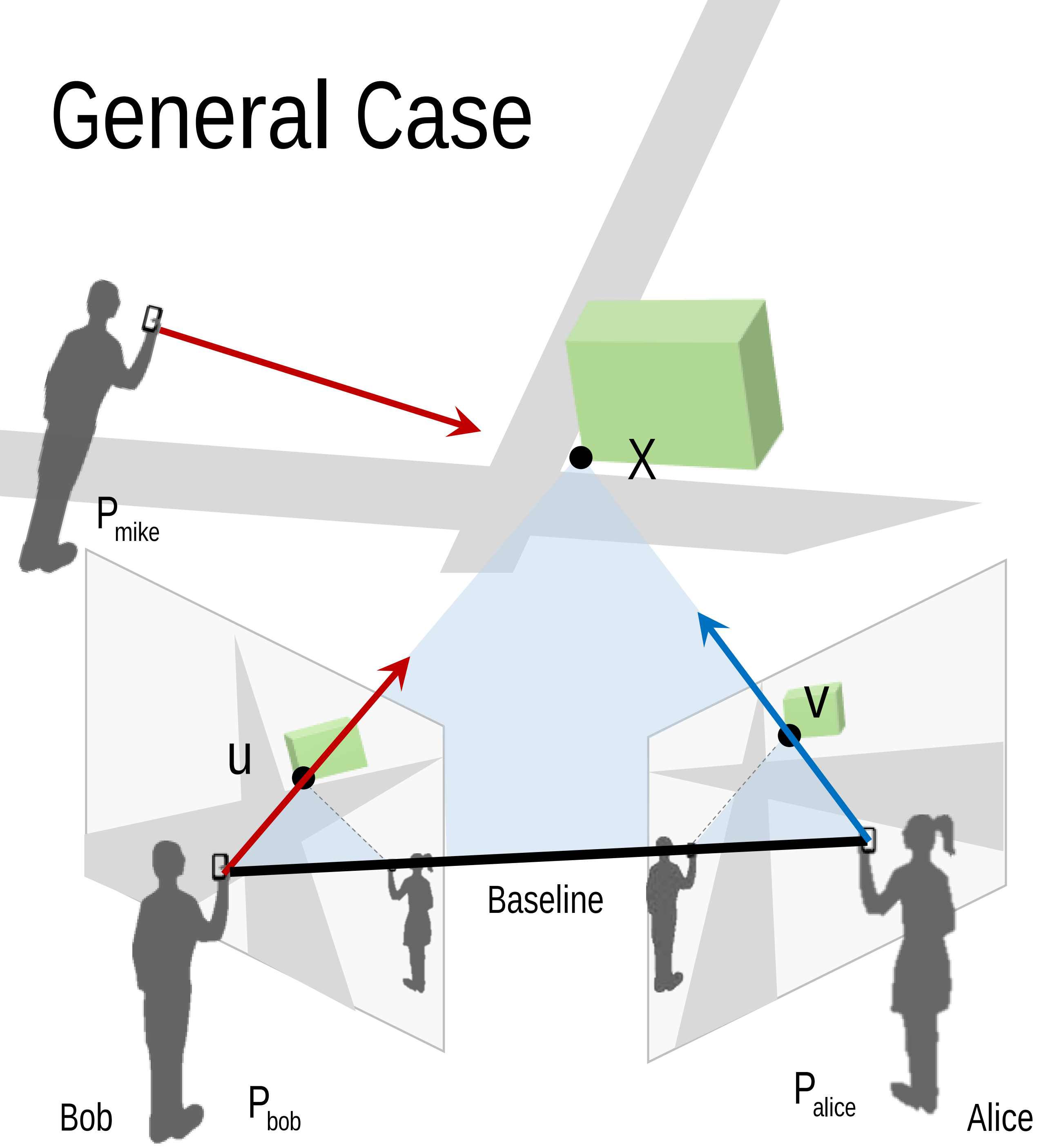
$$\longrightarrow \begin{bmatrix} u \\ 1 \end{bmatrix} \times P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

$$\longrightarrow \underbrace{\begin{matrix} \begin{bmatrix} u \\ 1 \end{bmatrix}_\times P_{bob} \\ \begin{bmatrix} v \\ 1 \end{bmatrix}_\times P_{alice} \end{matrix}}_{4x4} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

🟧 : Knowns

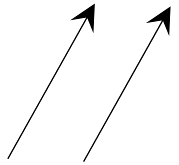
🟦 : Unknowns

General Case



General camera pose

$$\lambda_1 \begin{bmatrix} u \\ 1 \end{bmatrix} = P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix}$$



Two 3D vectors are parallel.

→ $$\begin{bmatrix} u \\ 1 \end{bmatrix} \times P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

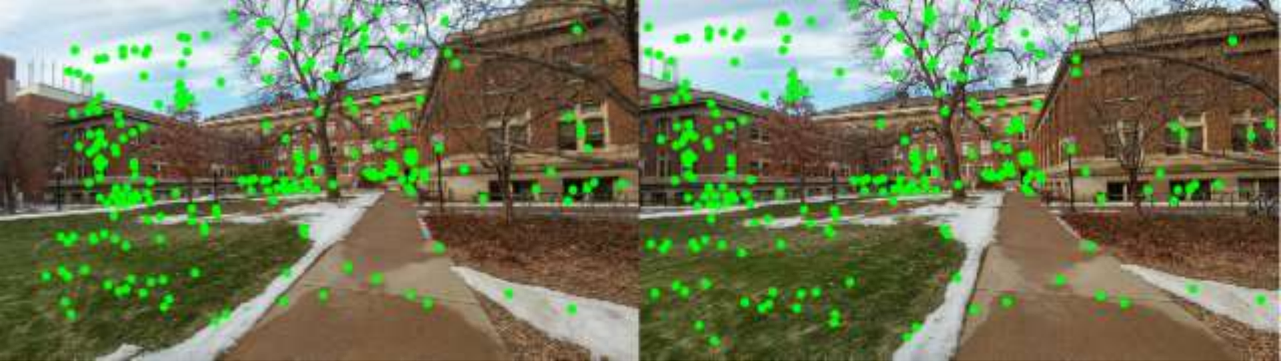
→ $$\begin{bmatrix} u \\ 1 \end{bmatrix}_\times P_{bob} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

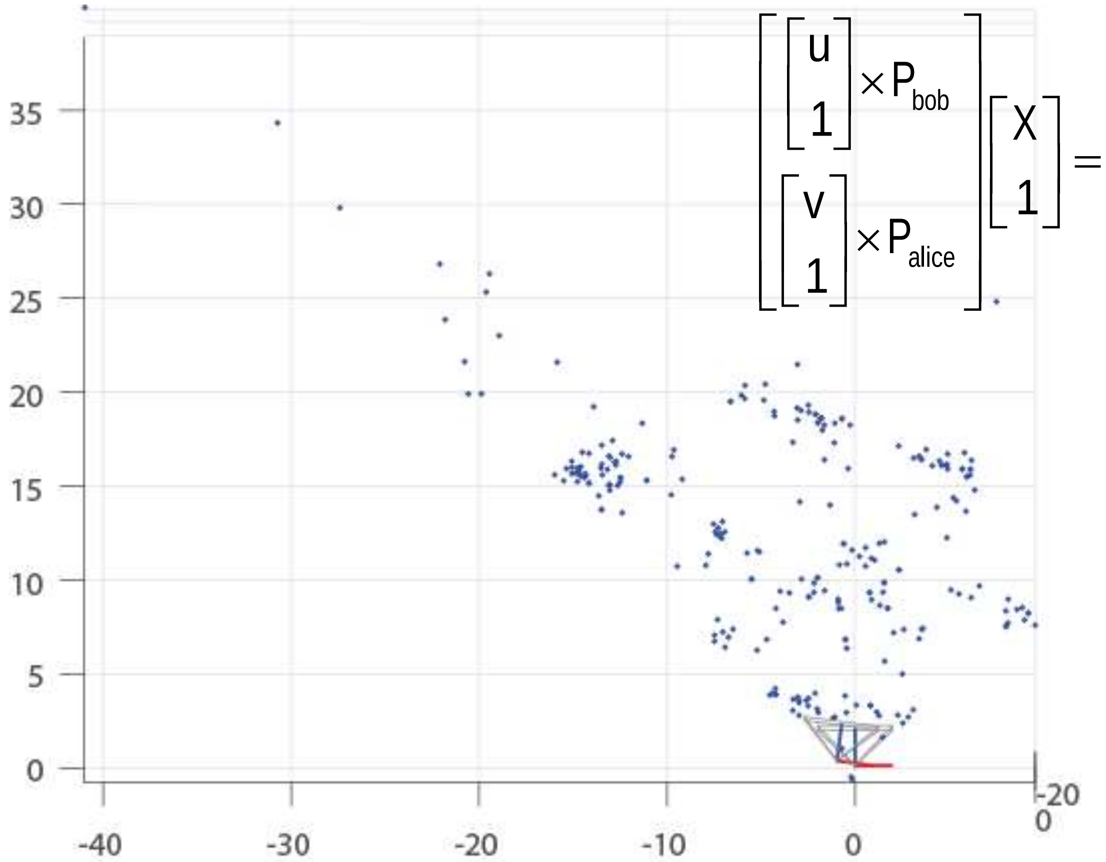
$$\begin{bmatrix} v \\ 1 \end{bmatrix}_\times P_{alice}$$

$$\begin{bmatrix} w \\ 1 \end{bmatrix}_\times P_{mike}$$

🟧 : Knowns

🟦 : Unknowns





$$\begin{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix}_{\times} P_{bob} \\ \begin{bmatrix} v \\ 1 \end{bmatrix}_{\times} P_{alice} \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix} = 0$$

Download Triangulation.m and triangulation.mat files

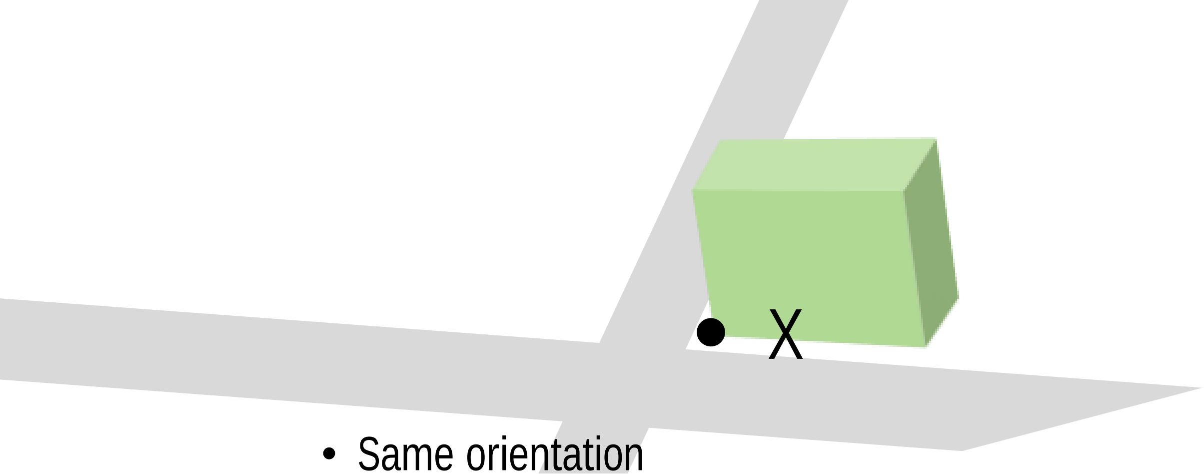
```
function Triangulation
%% Data loading
load('triangulation.mat');

% (C1, R1) and (C2, R2) are camera center and orientation of camera 1 and 2, respectively.
% u and v are Nx2 correspondences

%% Camera matrix build
K = [700/2 0 960/2;
     0 700/2 540/2;
     0 0 1];
% Build camera matrix 1 and 2
% P1
% P2
```

Fill out

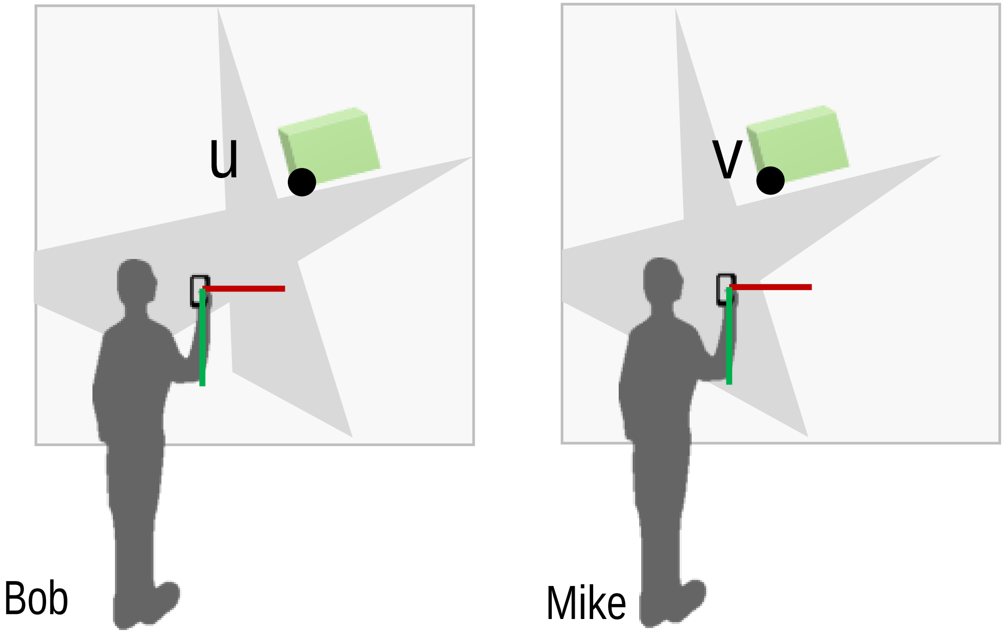
```
%% Triangulation
% Go to each correspondence and compute the 3D point X (3xN) matrix
for i = 1 : size(u,1)
    % Construct A matrix
    % Solve linear least squares to get 3D point
    % X(:,i) = point_3d;
end
```

Fill out

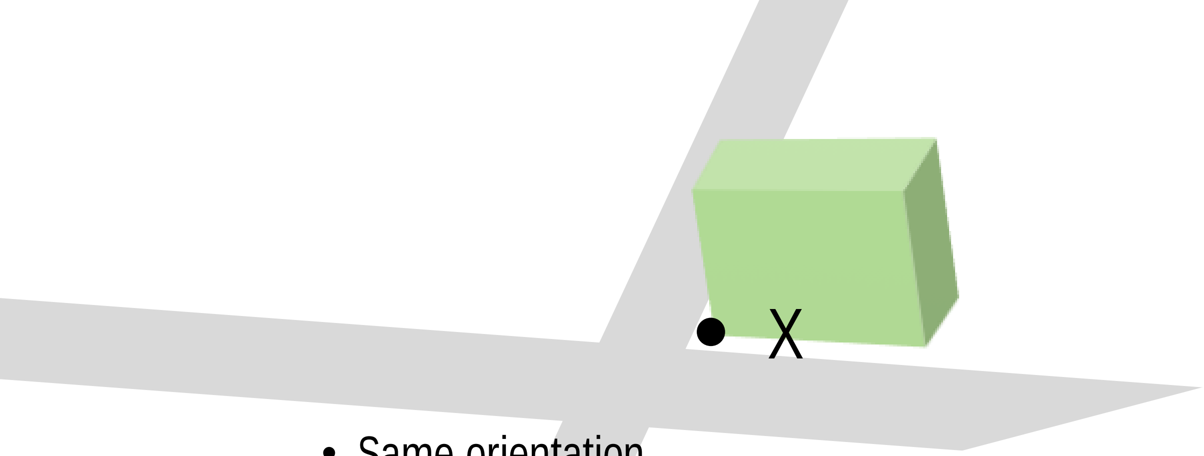
Special Case: Stereo



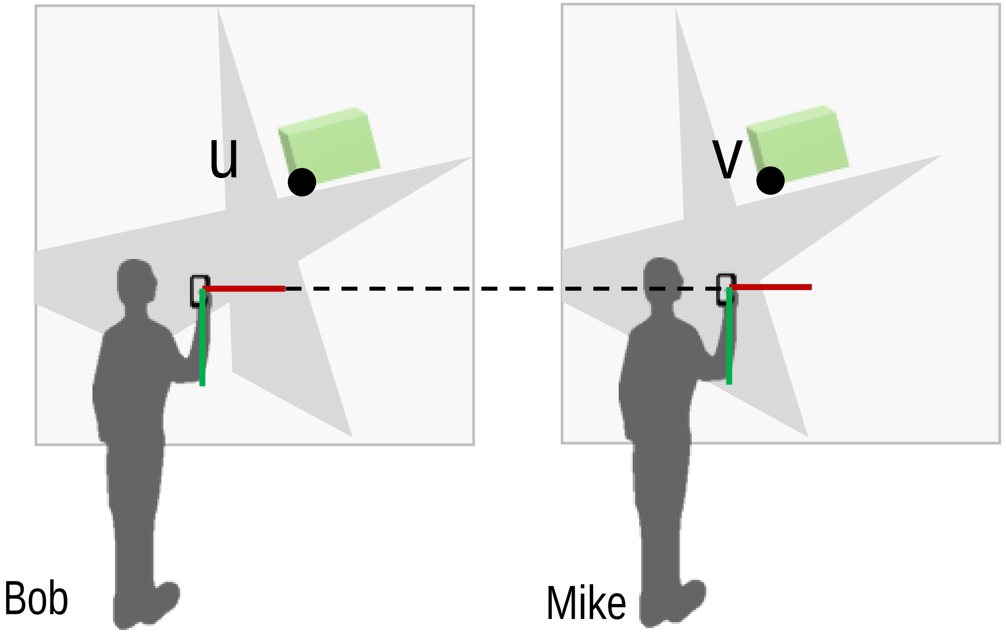
- Same orientation



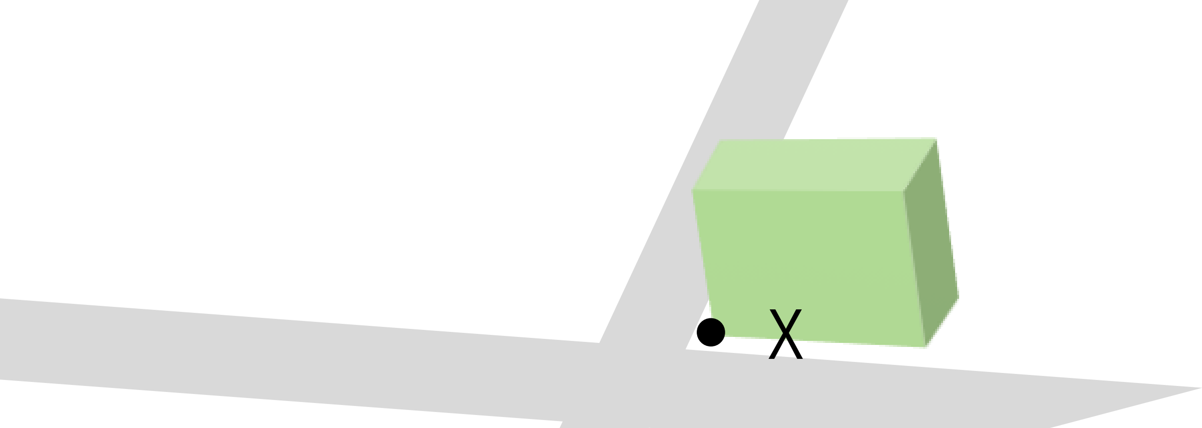
Special Case: Stereo



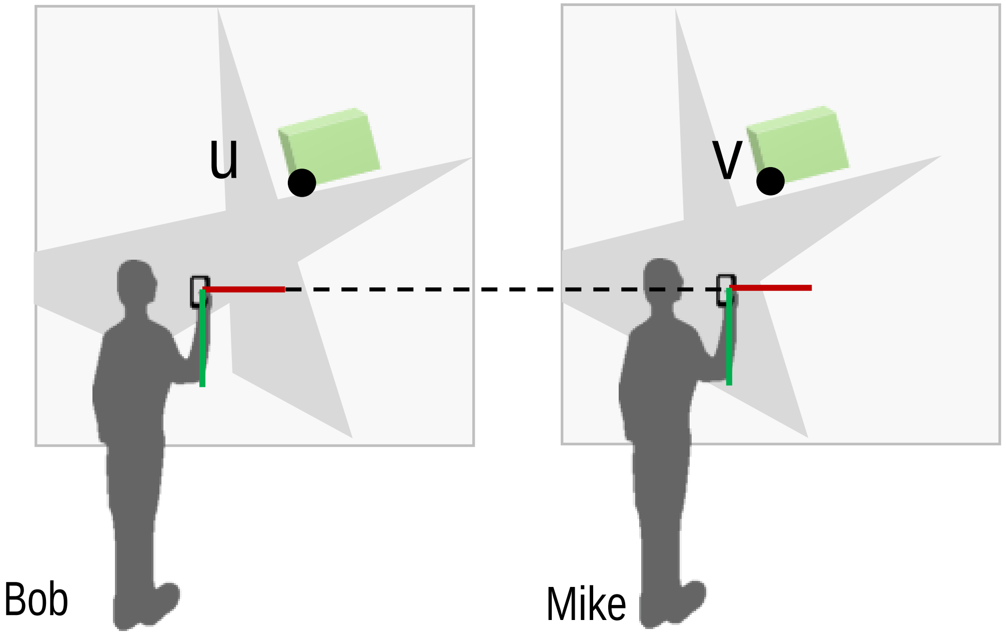
- Same orientation
- Alignment between X axis and baseline



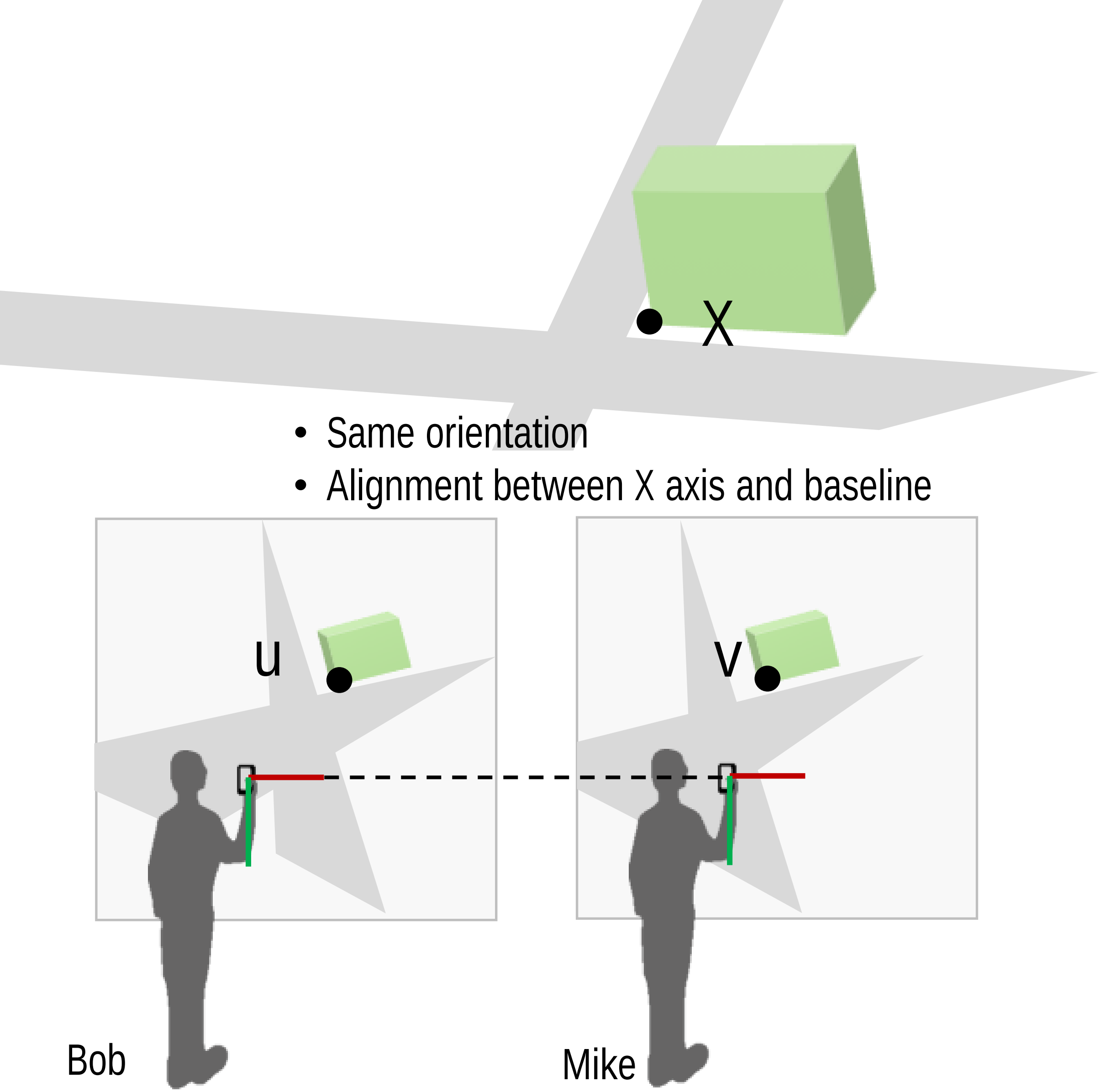
Special Case: Stereo



- Same orientation
- Alignment between X axis and baseline



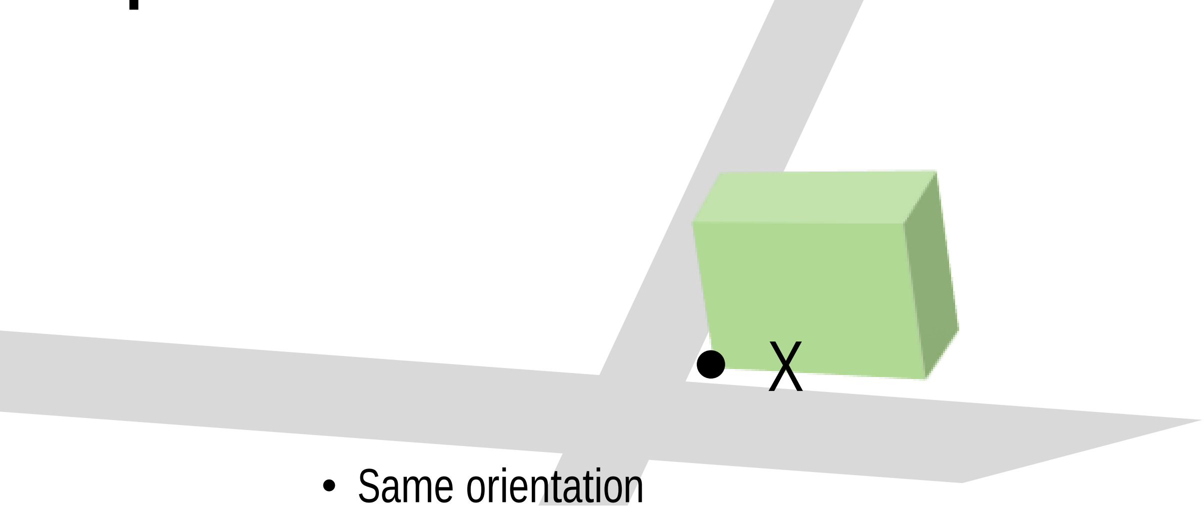
Special Case: Stereo



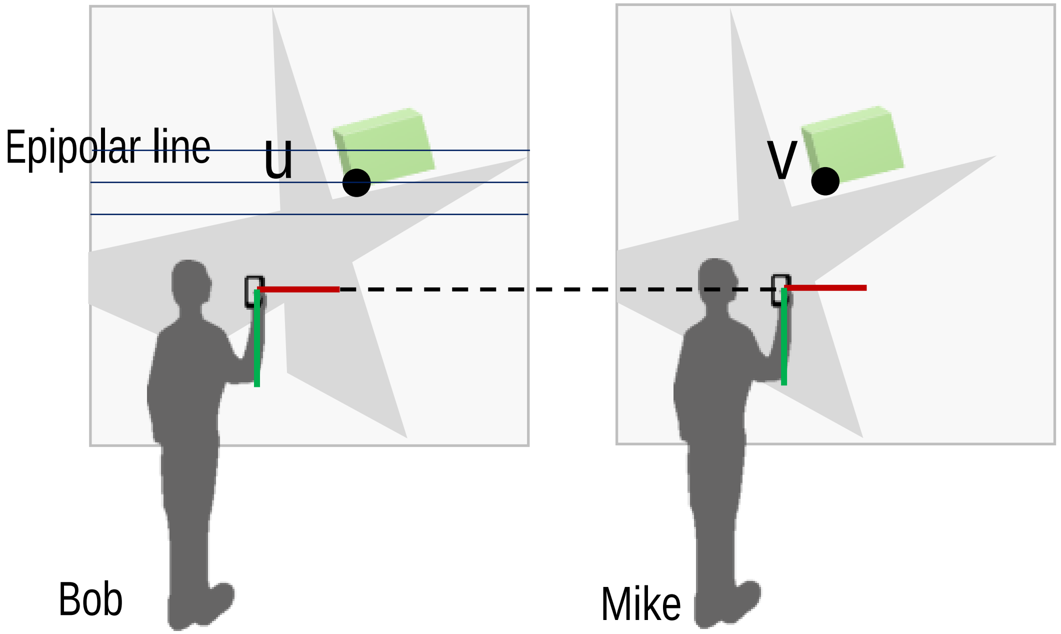
- Same orientation
- Alignment between X axis and baseline

Top view

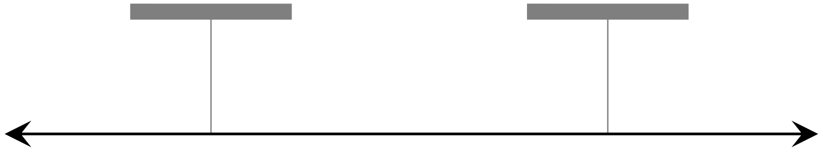
Special Case: Stereo



- Same orientation
- Alignment between X axis and baseline



Epipole?
Point at infinity

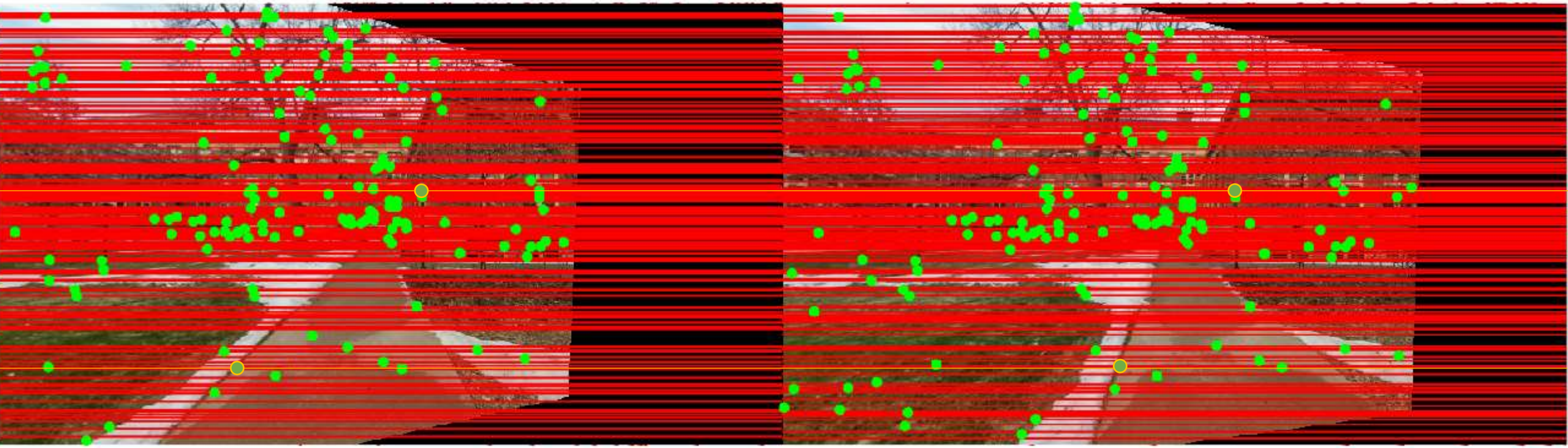


Top view

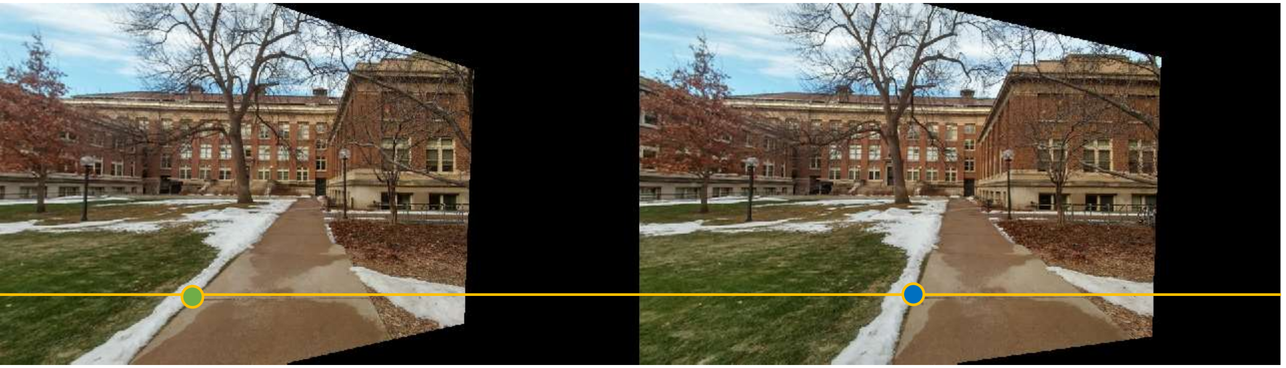
Special Case: Stereo

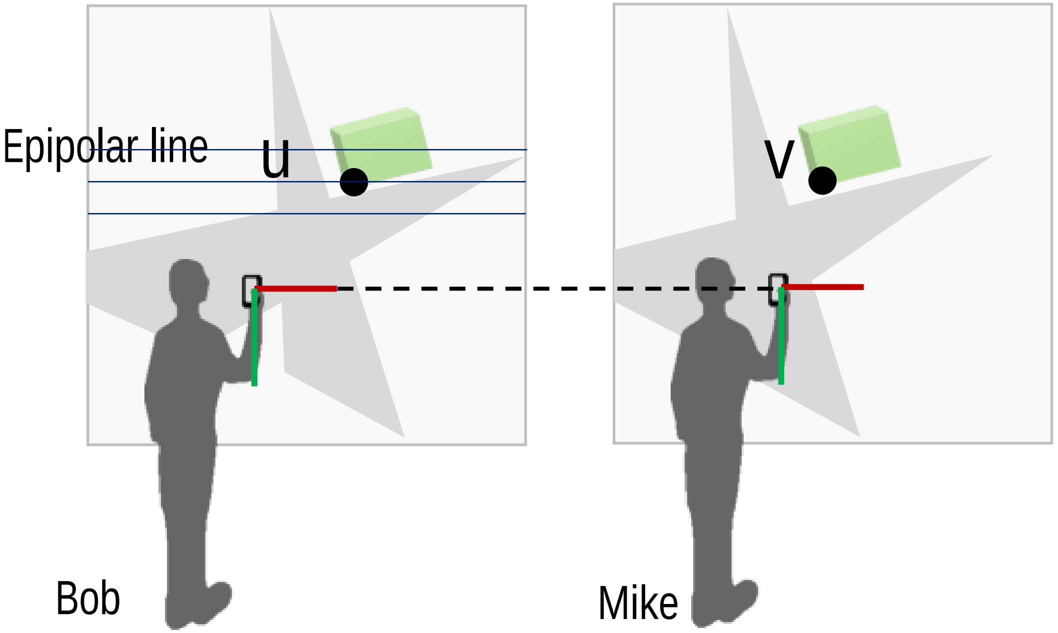
Special Case: Stereo

Special Case: Stereo

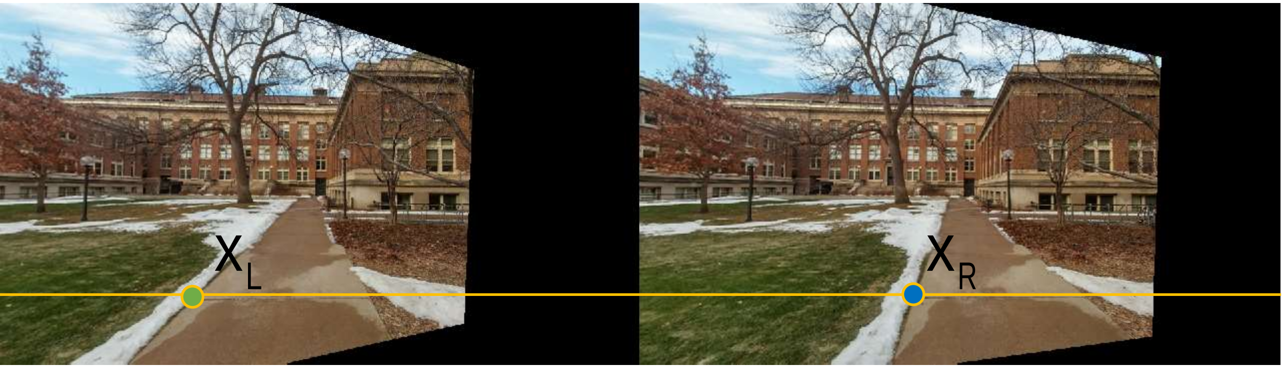


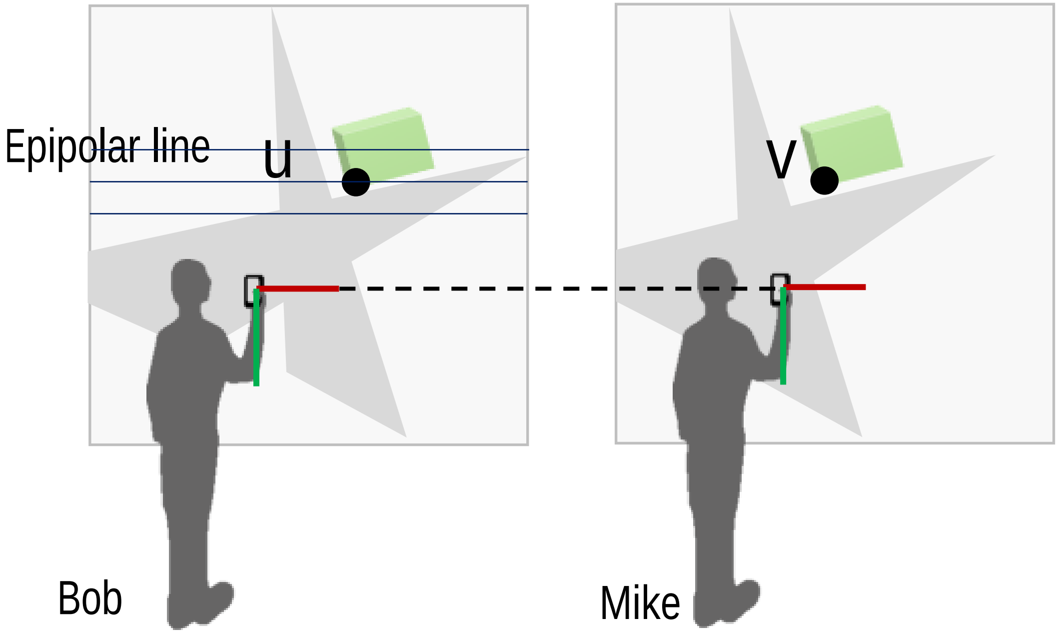
Special Case: Stereo

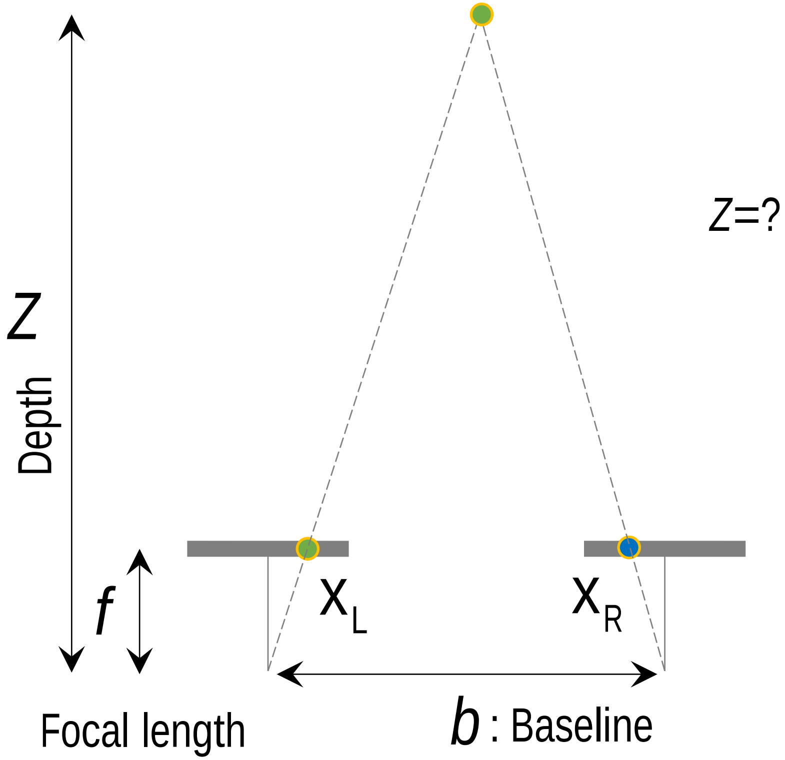




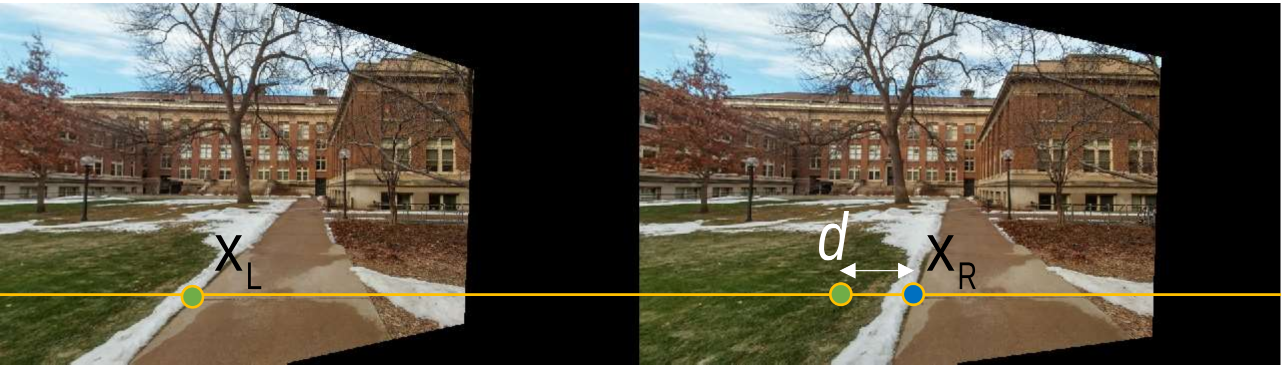
Special Case: Stereo

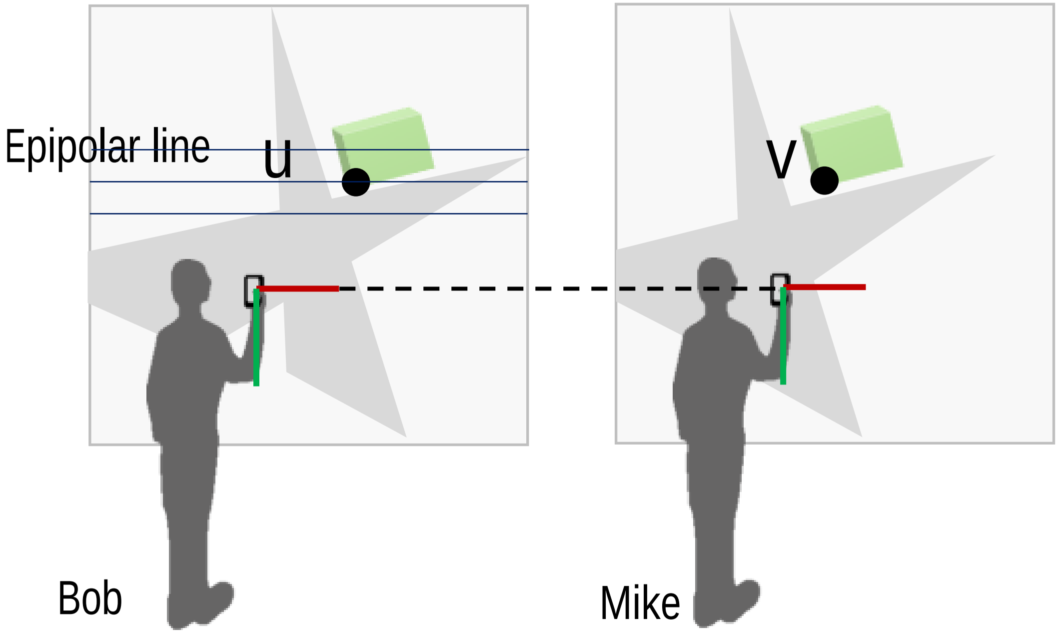


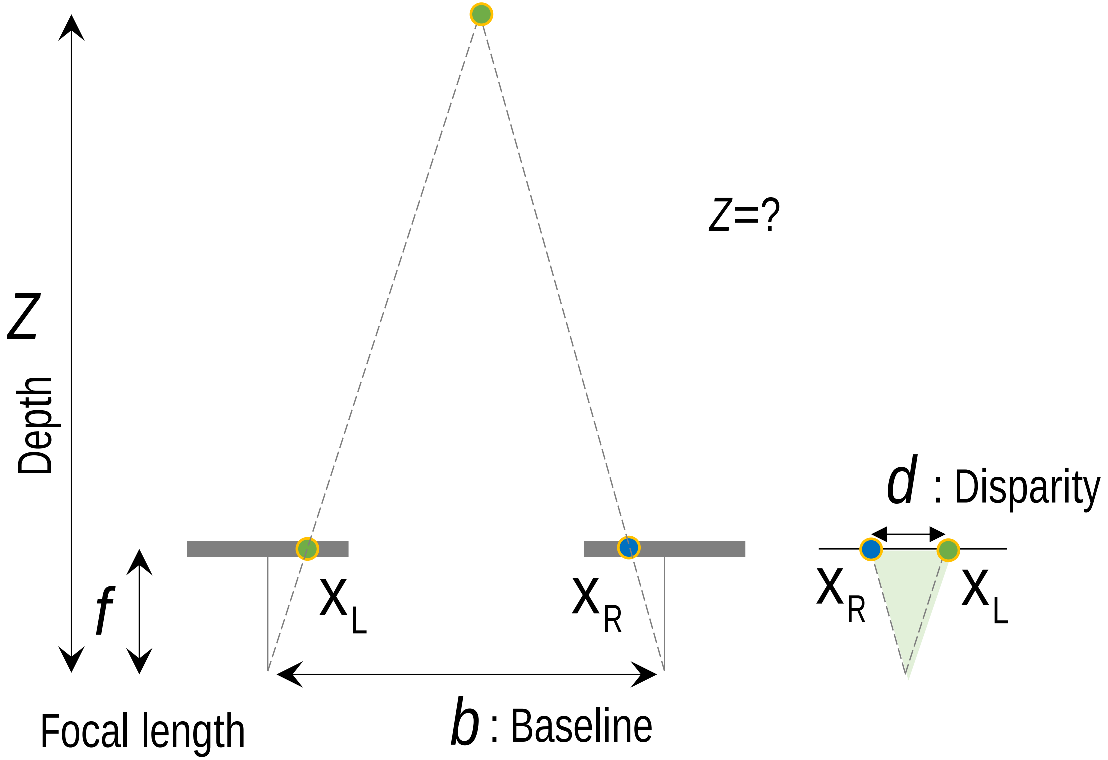




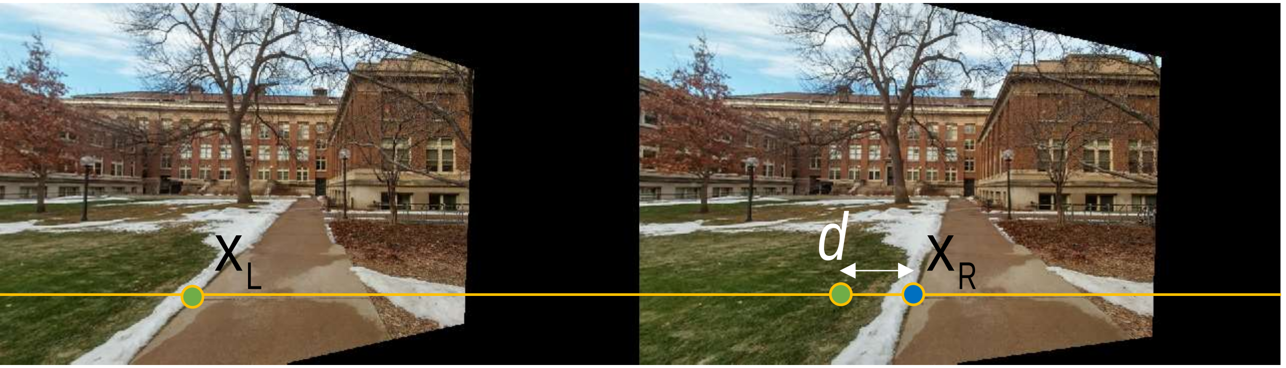
Special Case: Stereo

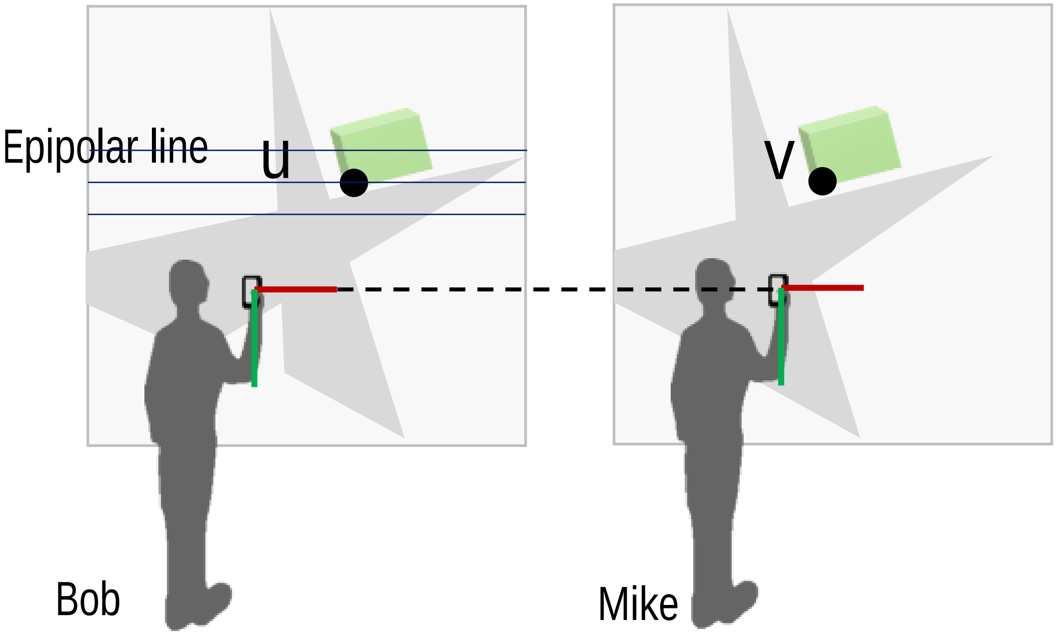


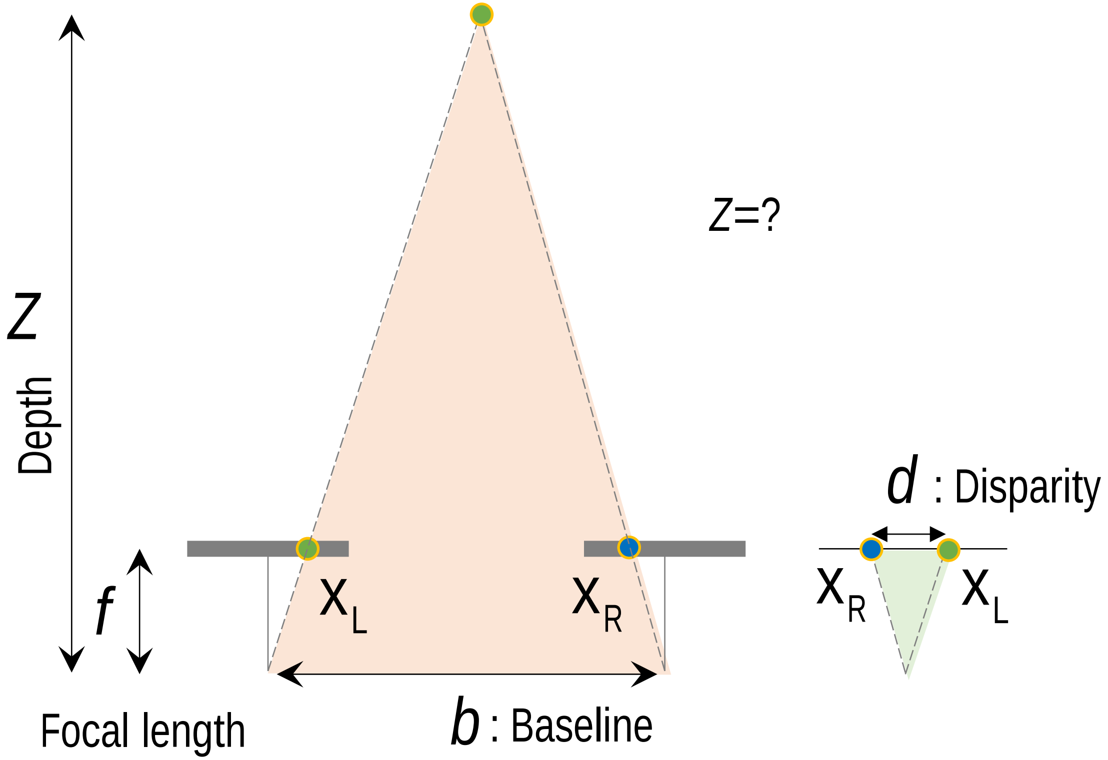




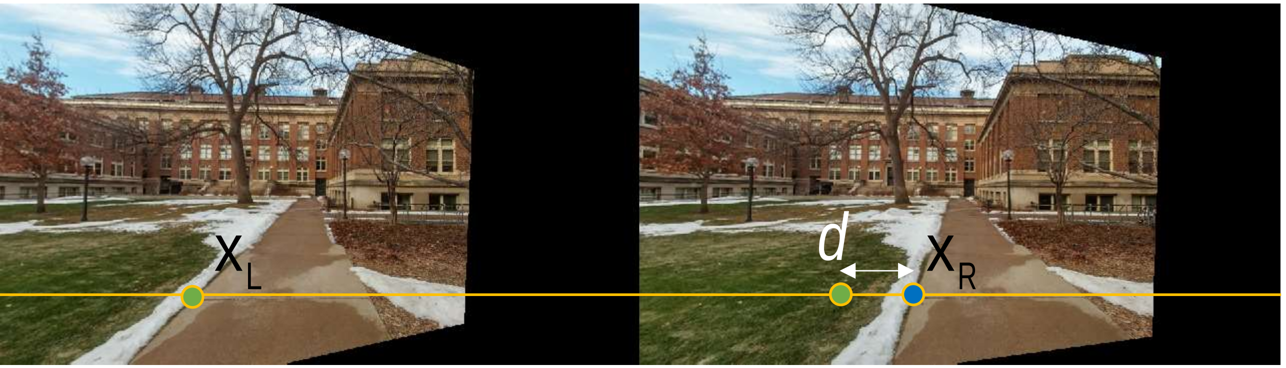
Special Case: Stereo

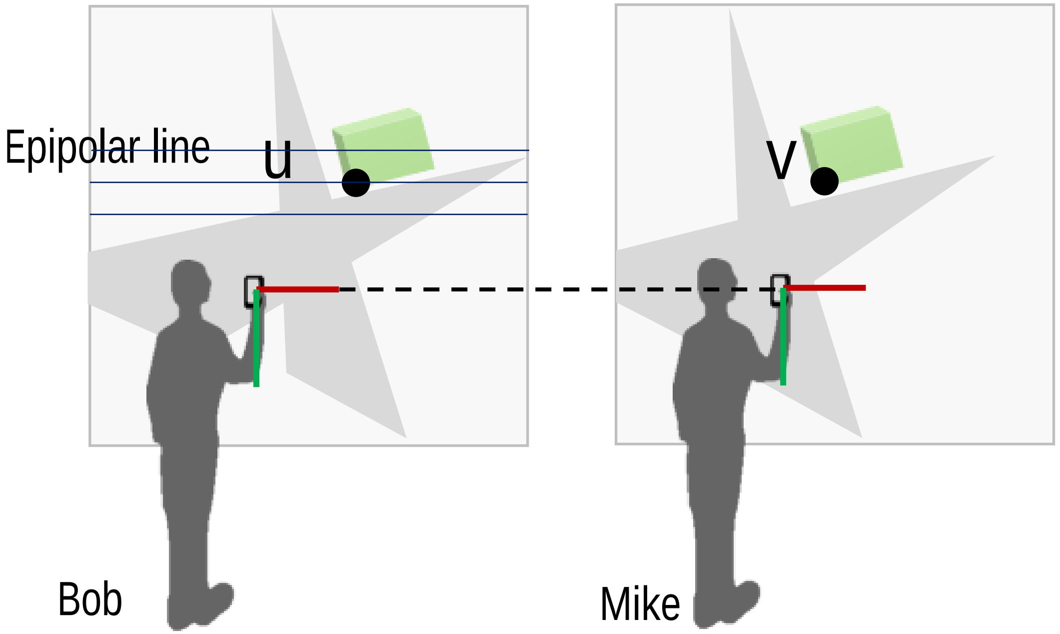


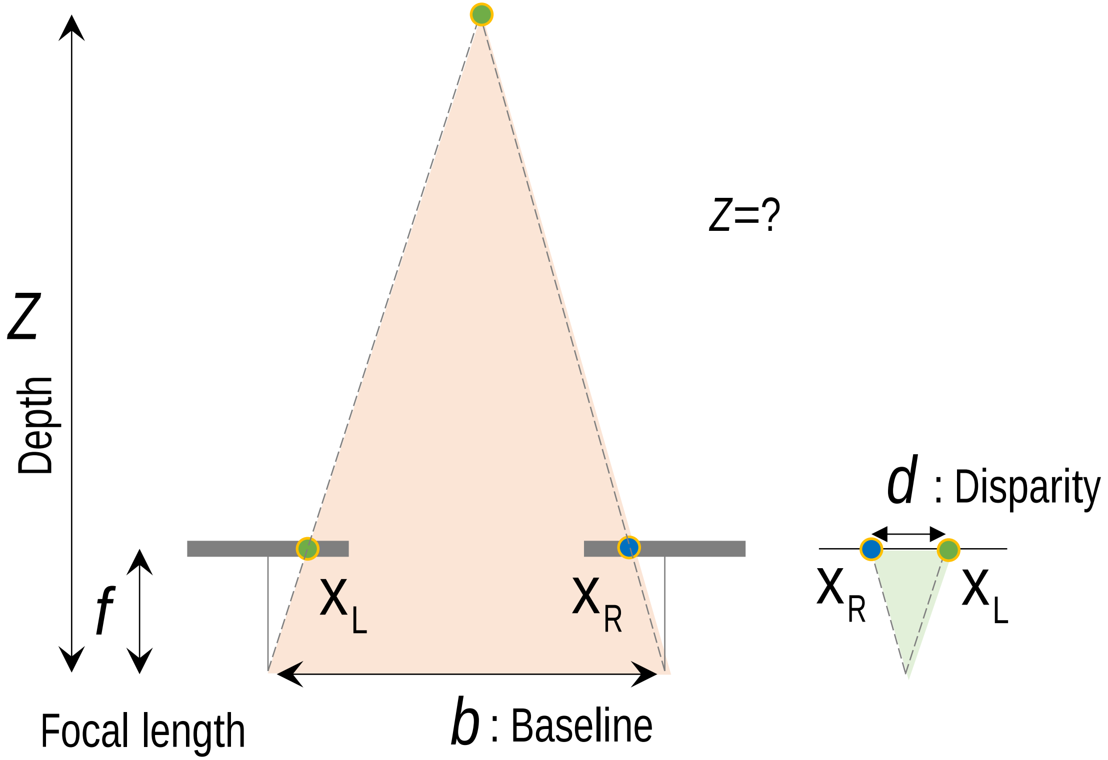


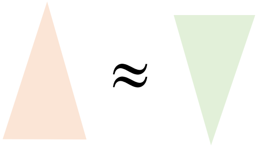


Special Case: Stereo







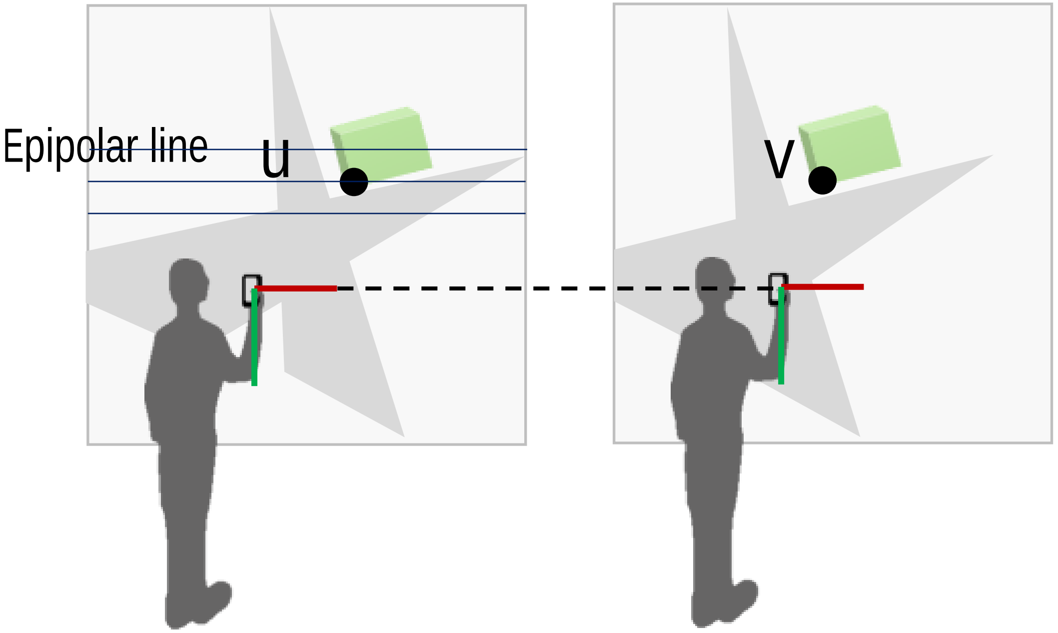


Two triangles are similar.

$$\frac{Z}{b}=\frac{f}{d} \rightarrow Z=b\frac{f}{d}$$

Stereo Rectification

Stereo Rectification



$$P_{\mathrm{bob}} = K\begin{bmatrix} I & 0 \end{bmatrix} \qquad P_{\mathrm{mike}} = KR\begin{bmatrix} I & -C \end{bmatrix}$$

- Same orientation

$$R_{\mathrm{rect}} = \begin{bmatrix} r_x^T \\ r_y^T \\ r_z^T \end{bmatrix}$$

- Alignment between X axis and baseline

$$r_x = \frac{C}{\|C\|}$$

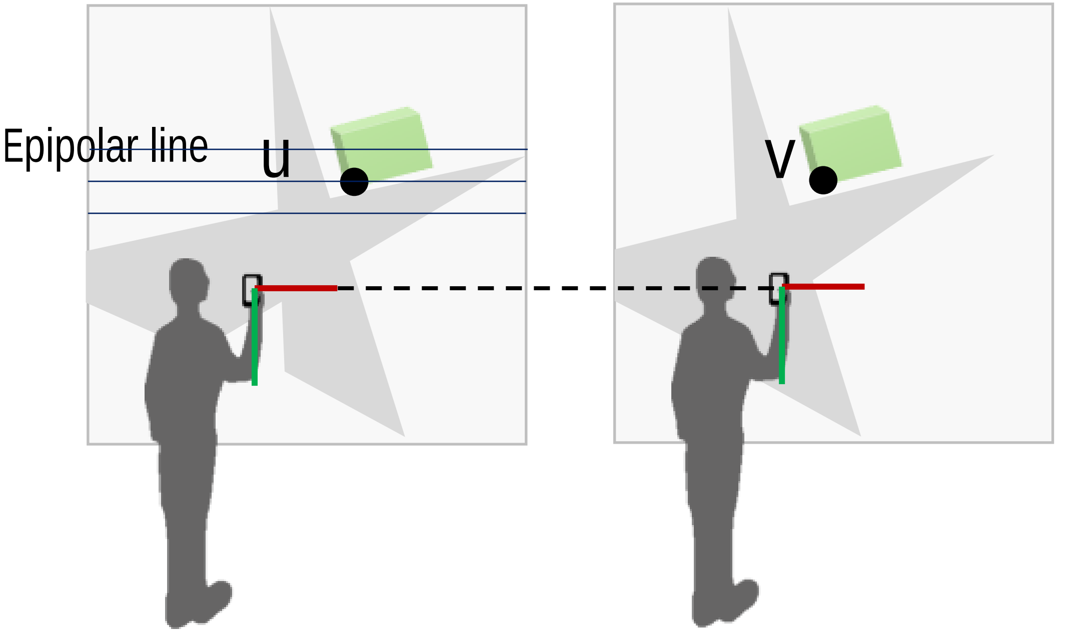
$$r_z = \frac{\tilde{r}_z - (\tilde{r}_z \cdot r_x) r_x}{\|\tilde{r}_z - (\tilde{r}_z \cdot r_x) r_x\|}$$

: Orthogonal projection

$$r_y = r_z \times r_x$$

where $\tilde{r}_z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

Stereo Rectification



$$P_{bob} = K\begin{bmatrix} I & 0 \end{bmatrix}$$

$$P_{mike} = KR\begin{bmatrix} I & -C \end{bmatrix}$$

Homography by pure rotation: R_{rect}

$$H_{bob} = KR_{rect}K^{-1}$$

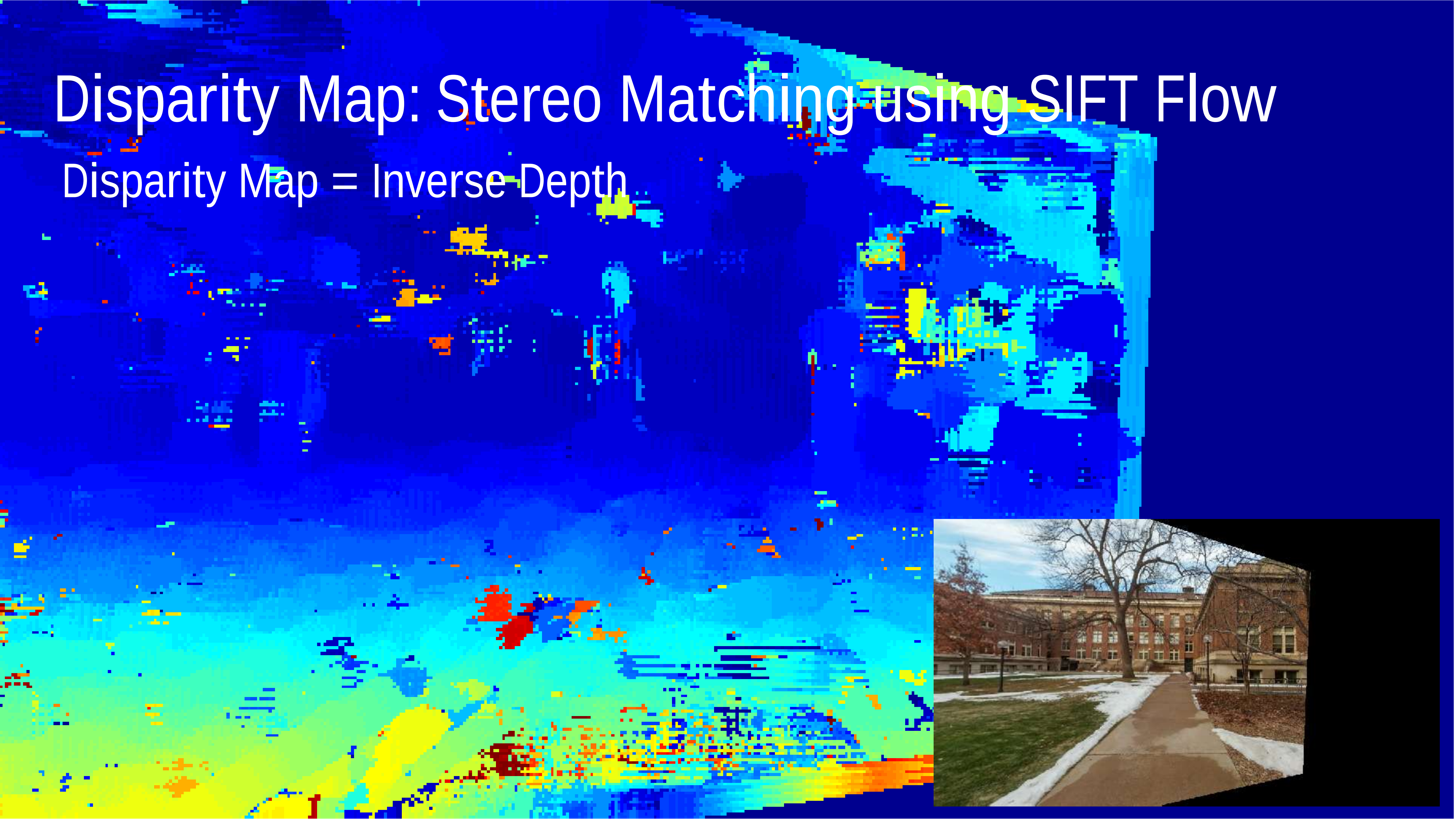
$$H_{mike} = KR_{rect}R^{T}K^{-1}$$

Dense Feature Matching using SIFT Flow

Find a minimum distance over the epipolar line

Disparity Map: Stereo Matching using SIFT Flow

Disparity Map = Inverse Depth

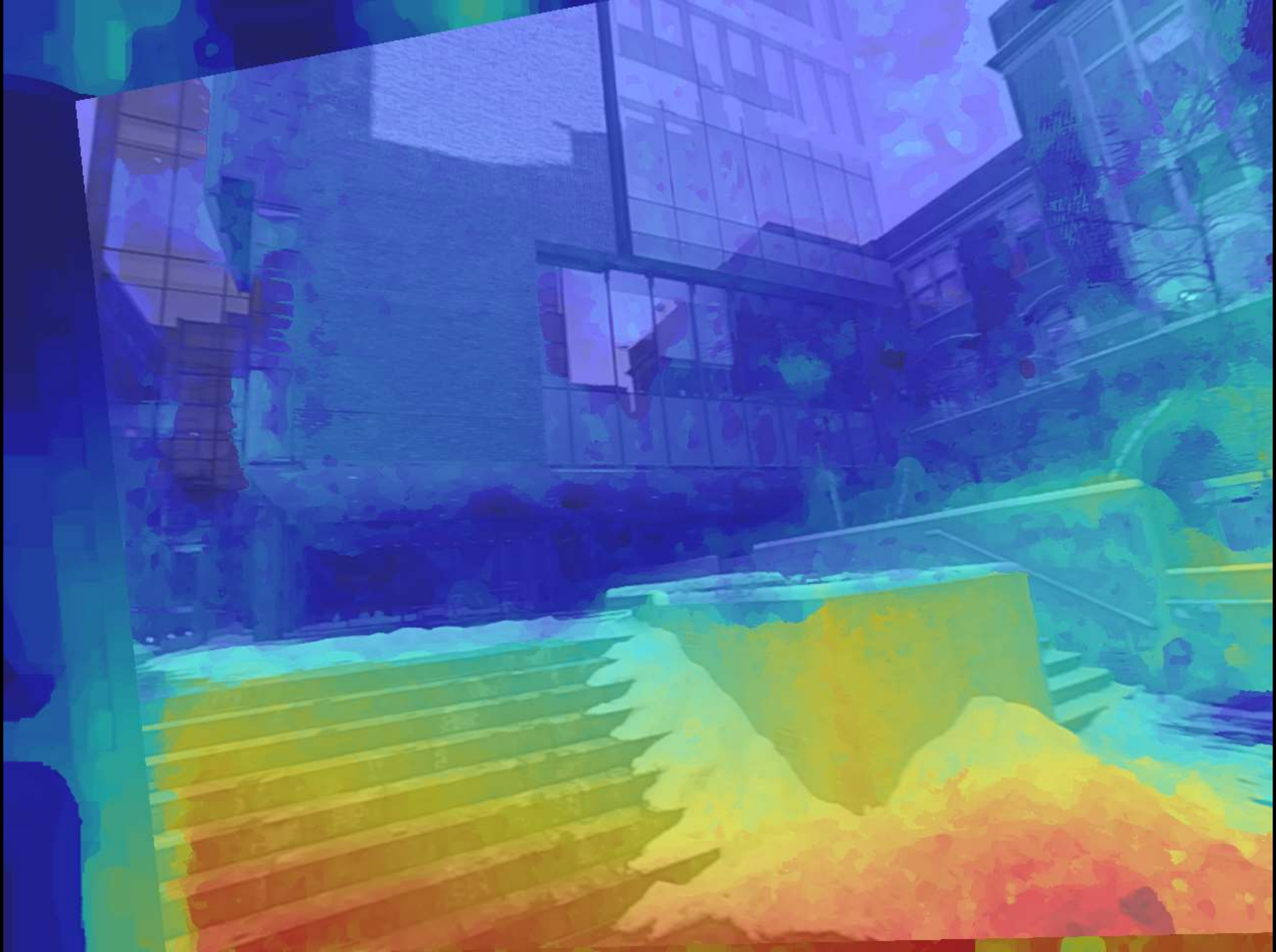


Disparity Map: Stereo Matching using SIFT Flow

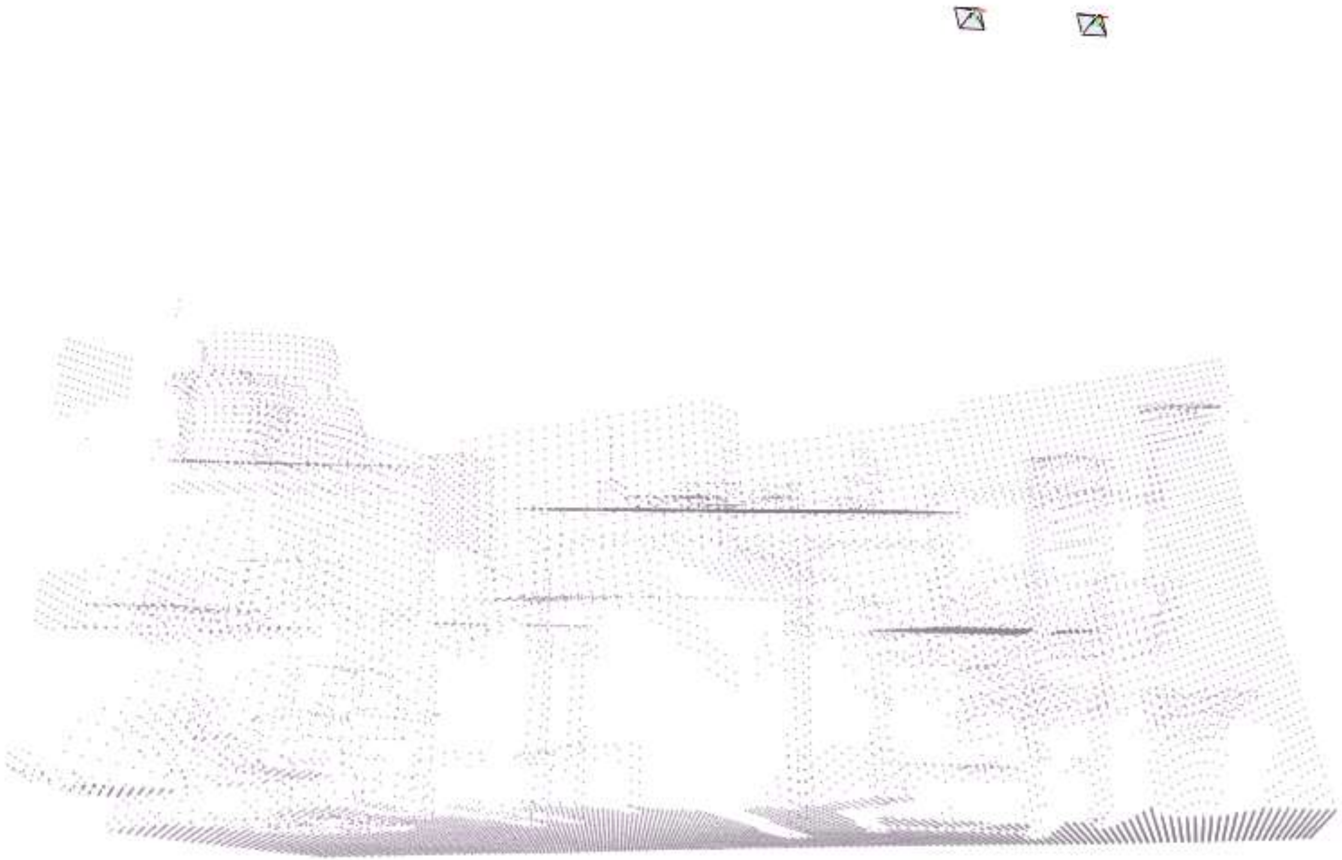
Disparity Map = Inverse Depth

Disparity Map: Stereo Matching using SIFT Flow

Disparity Map = Inverse Depth




maxell
VHS VIDEOCASSETTE
maxell
VHS VIDEOCASSETTE




EgoMotion Dataset (outdoor)

Dense Reconstruction using a Monocular Camera