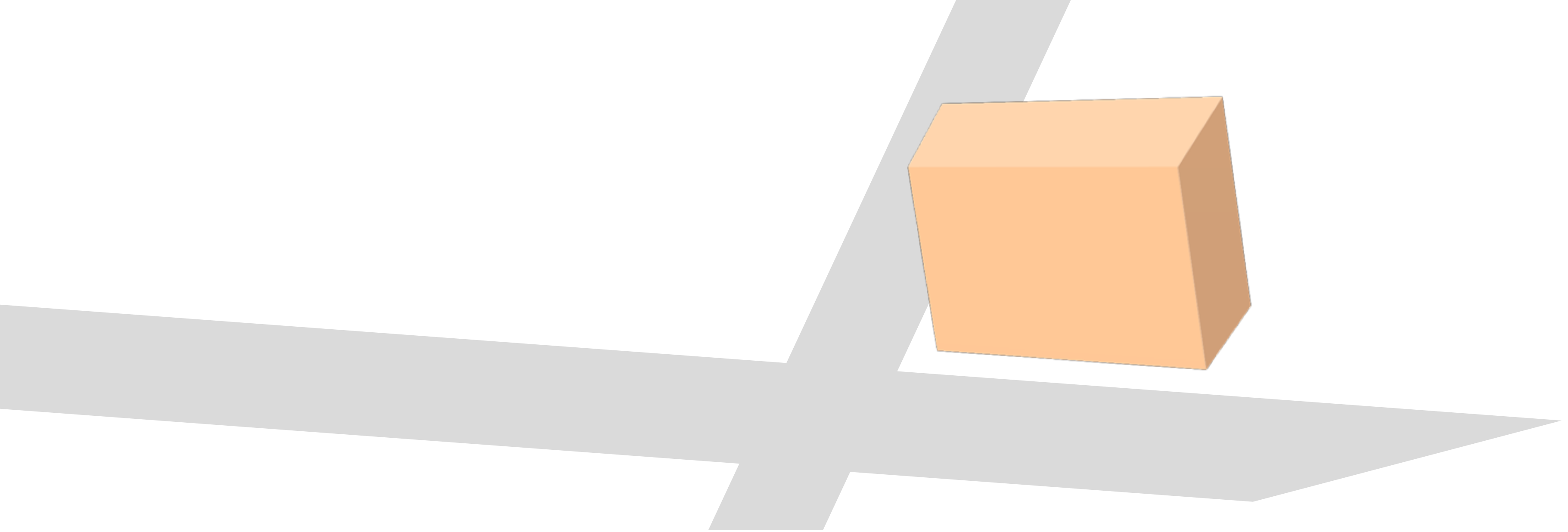
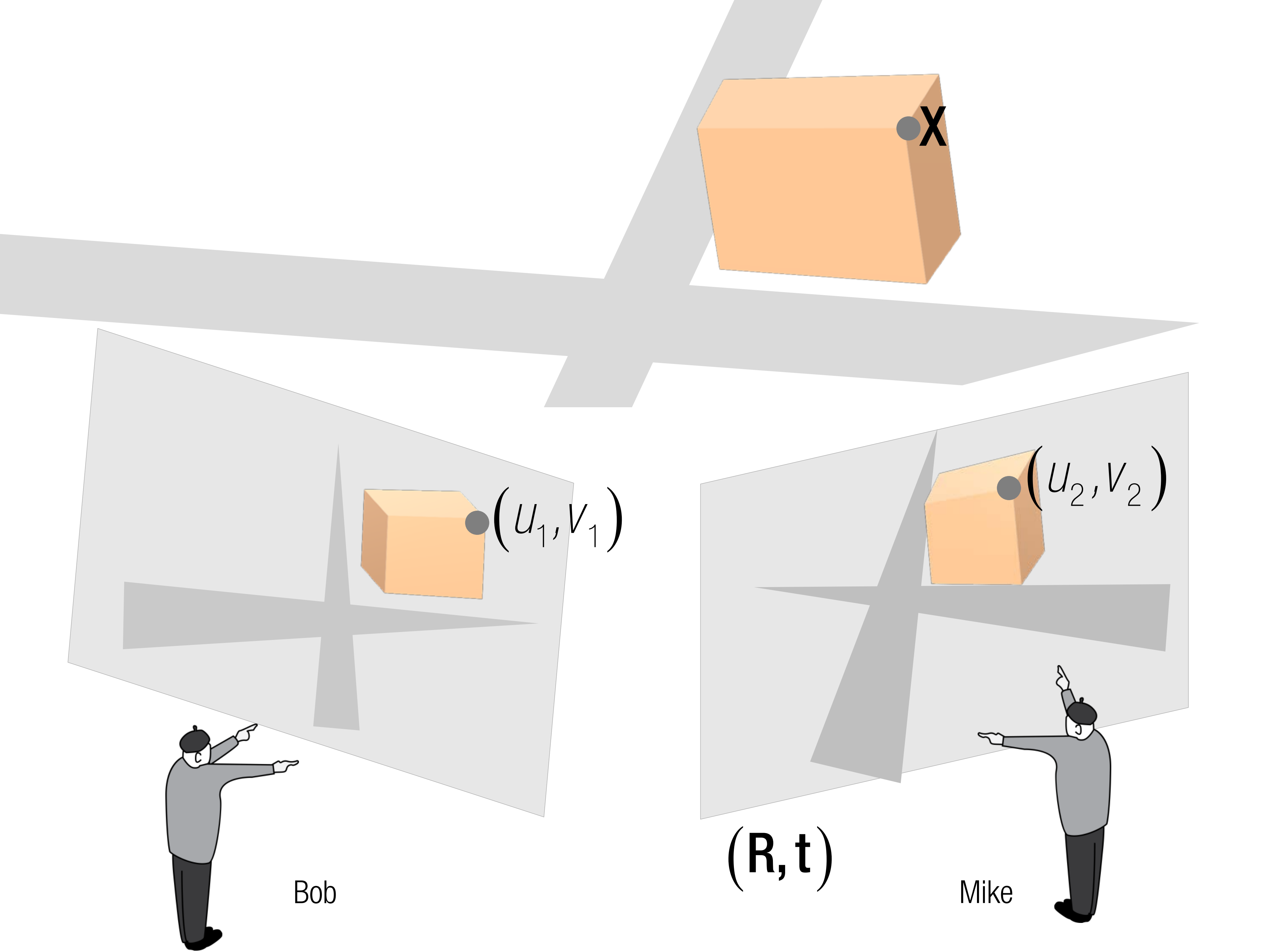
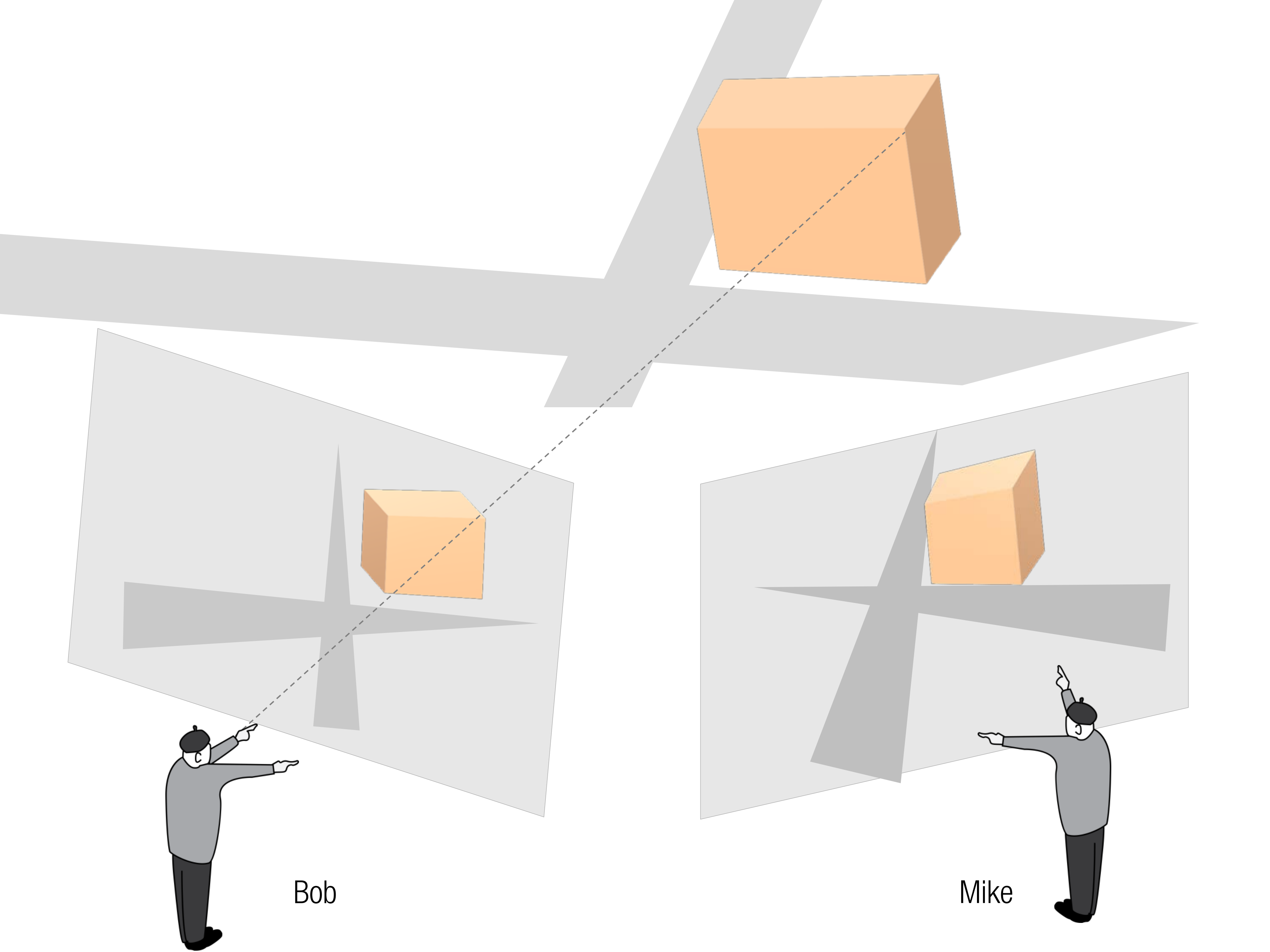
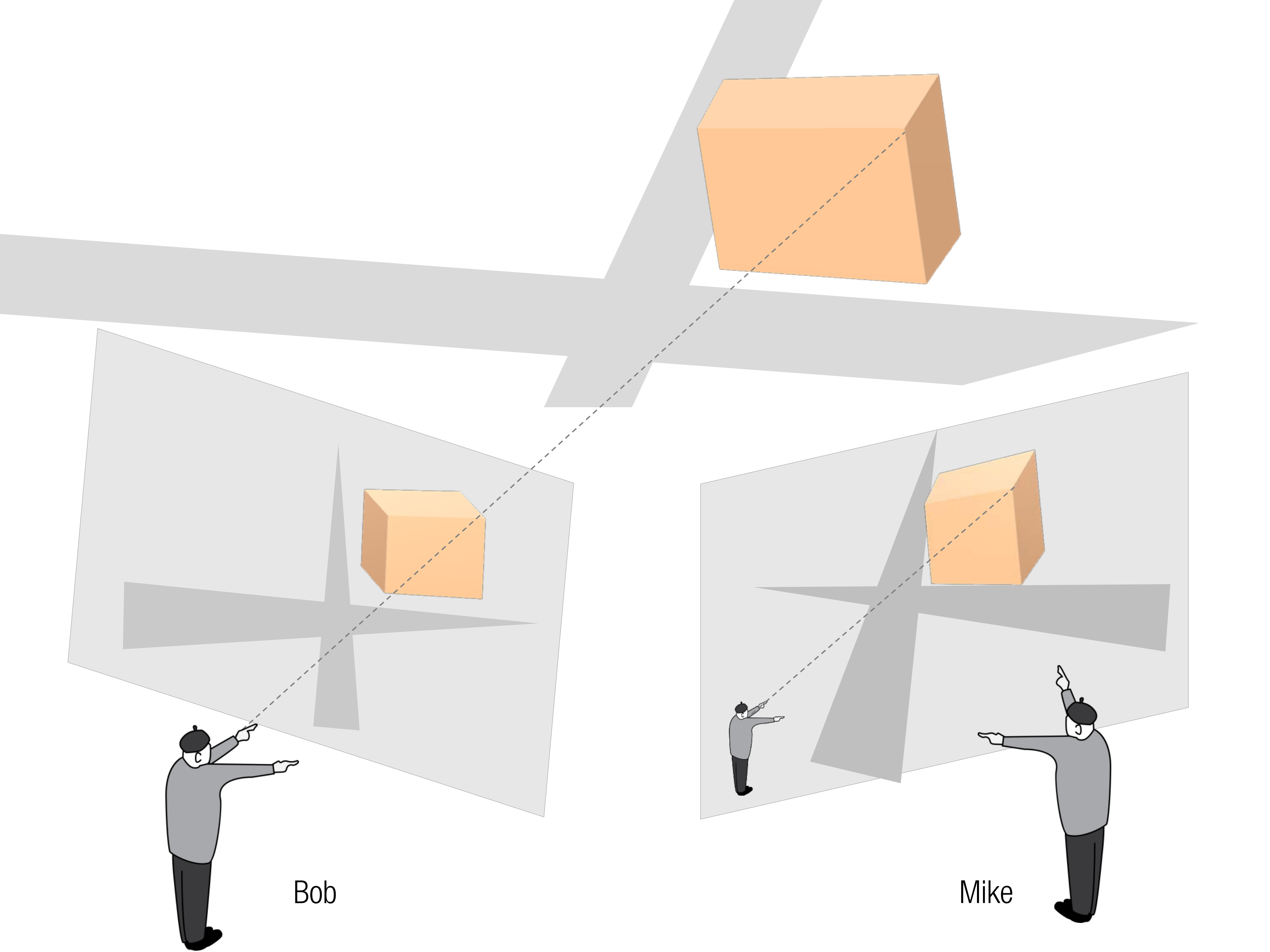
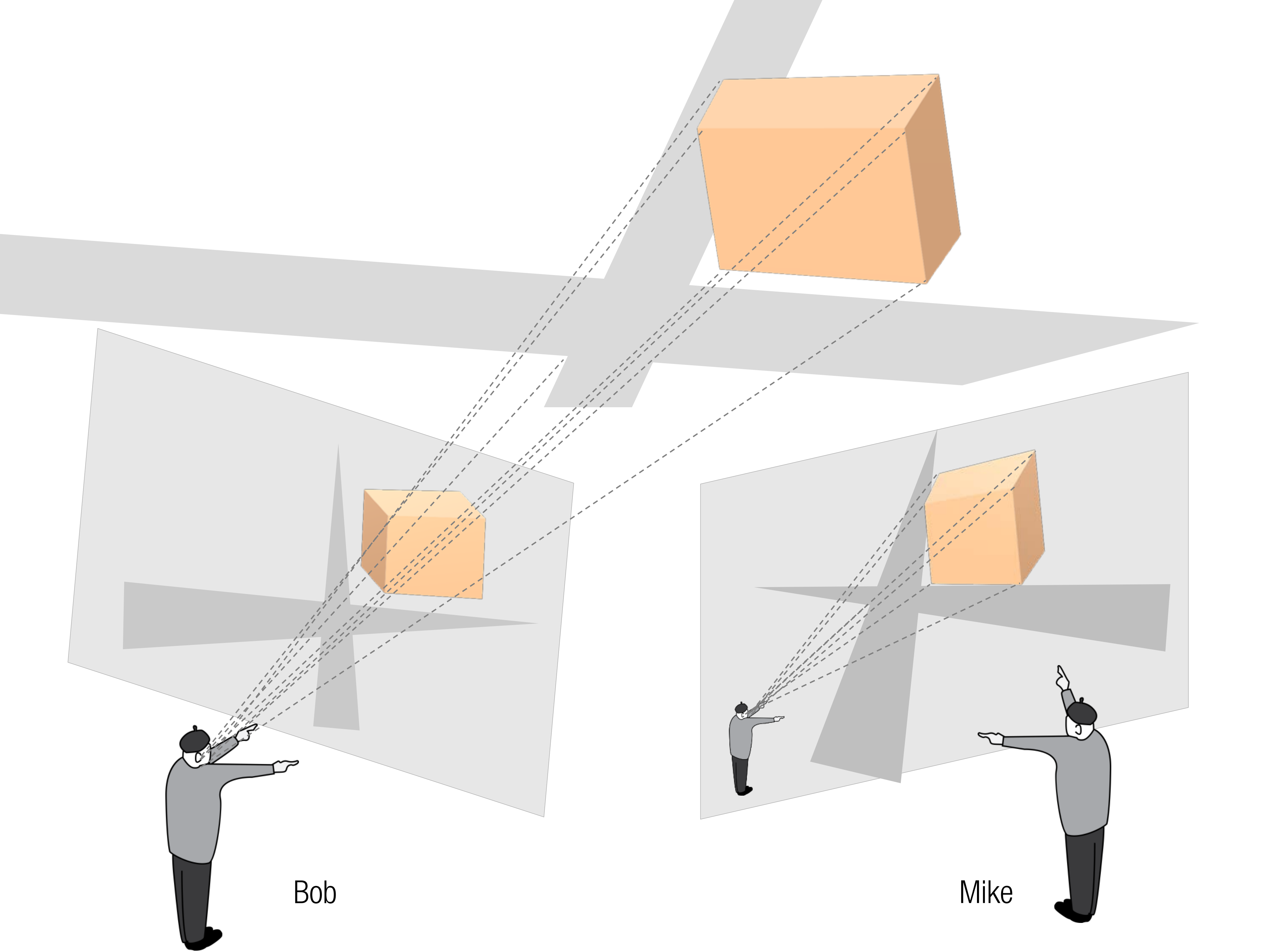
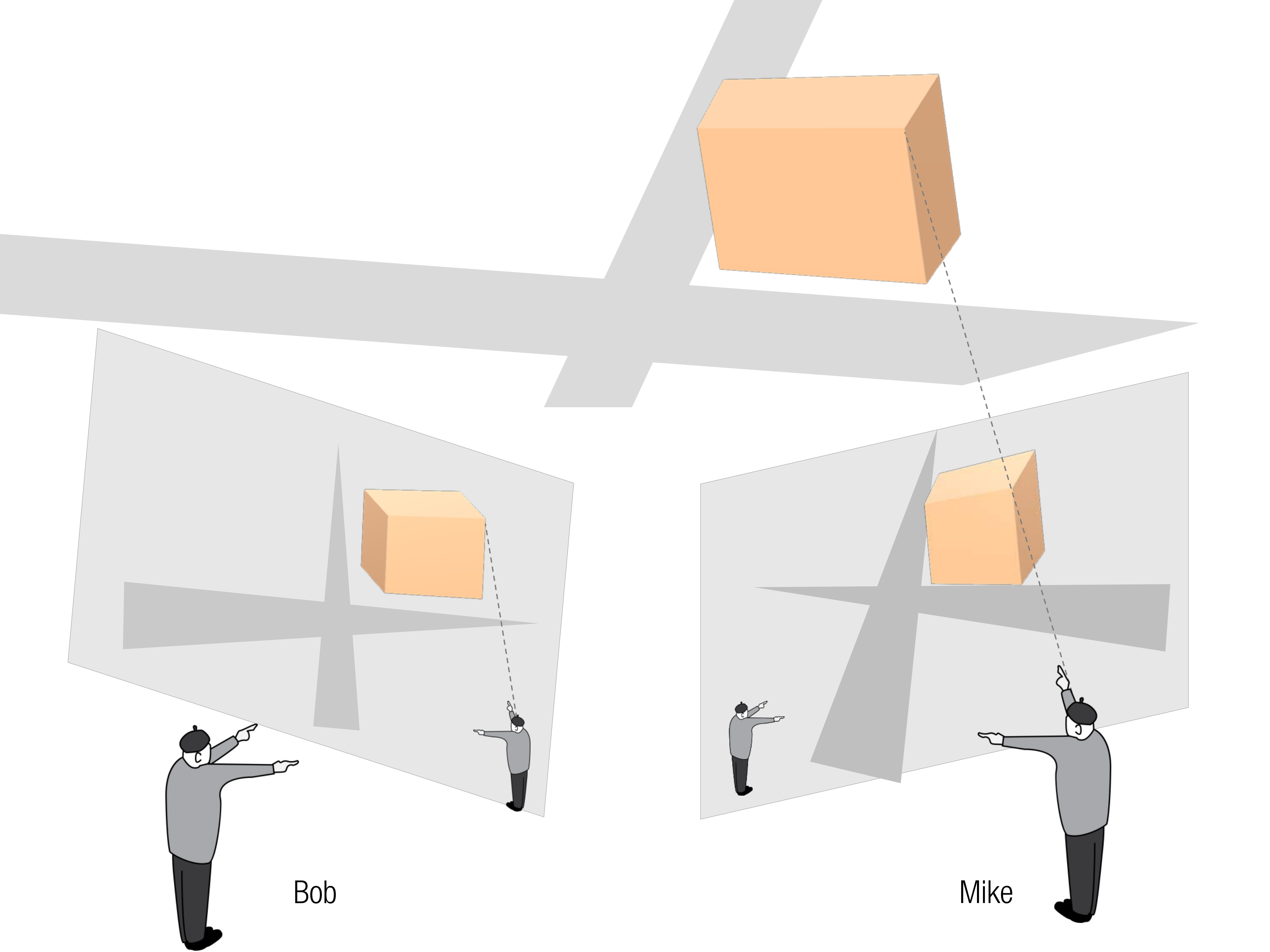


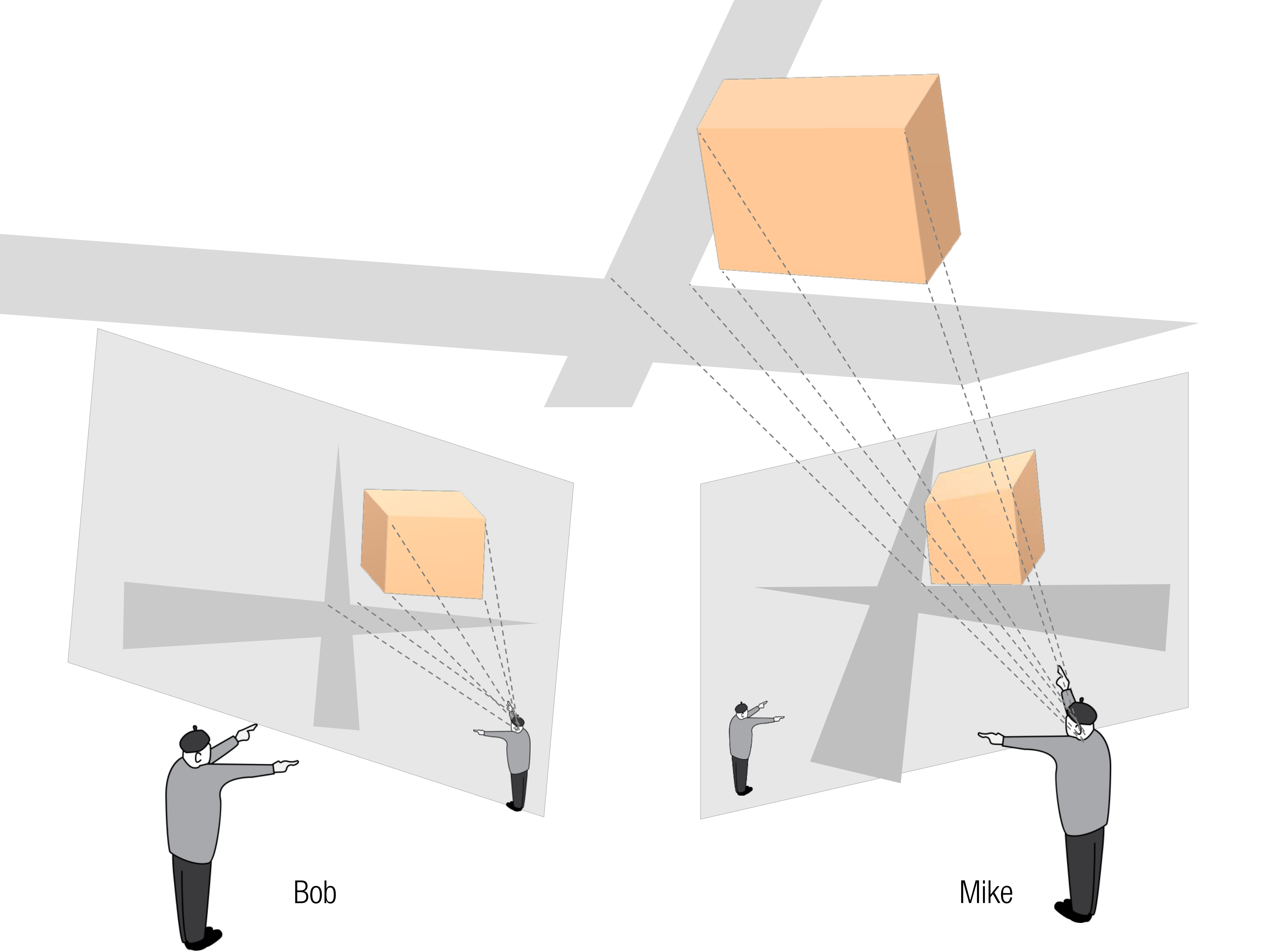

X
(u_1, v_1)
(u_2, v_2)
$(\mathbf{R}, \mathbf{t})$
Bob
Mike

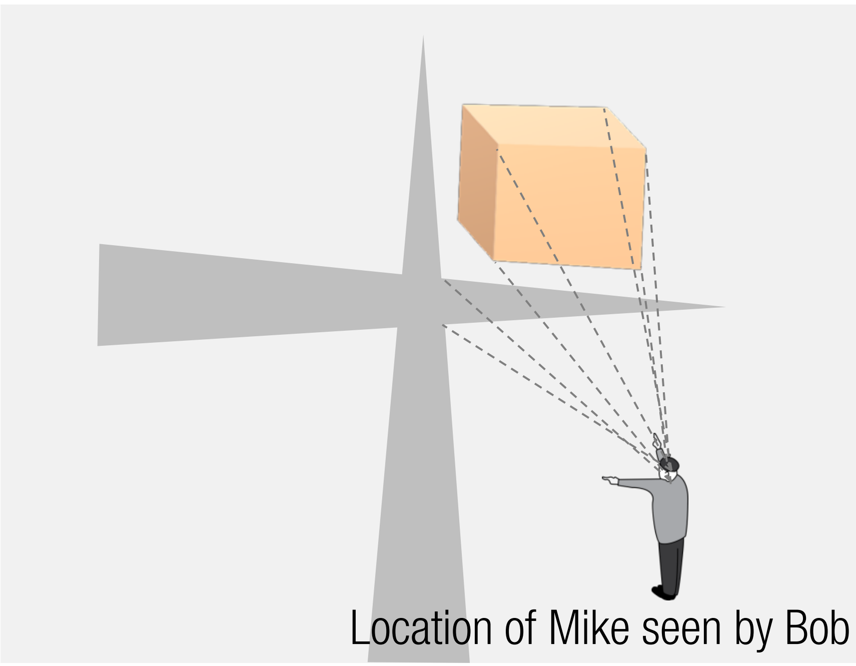

Bob
Mike


Bob
Mike

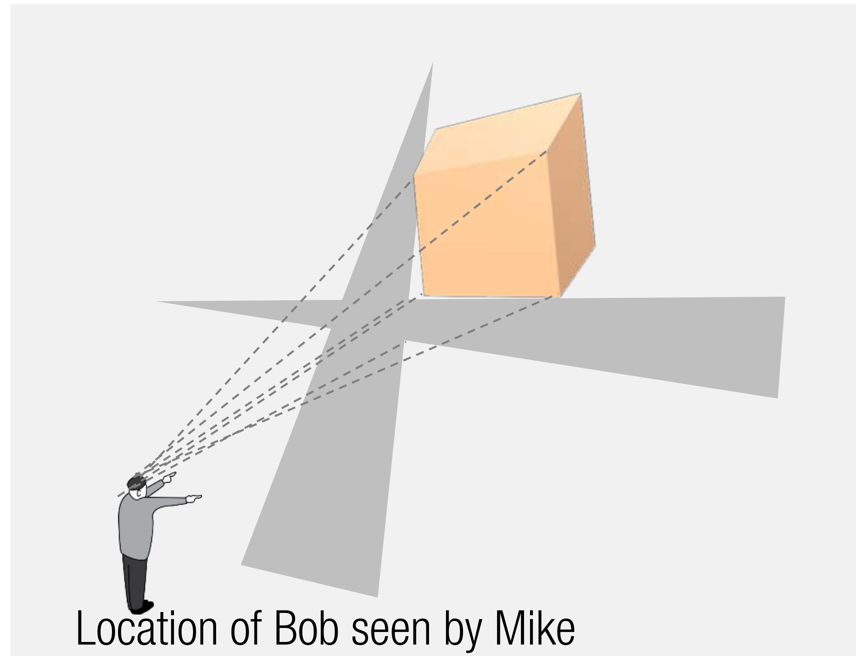

Bob
Mike


Bob
Mike

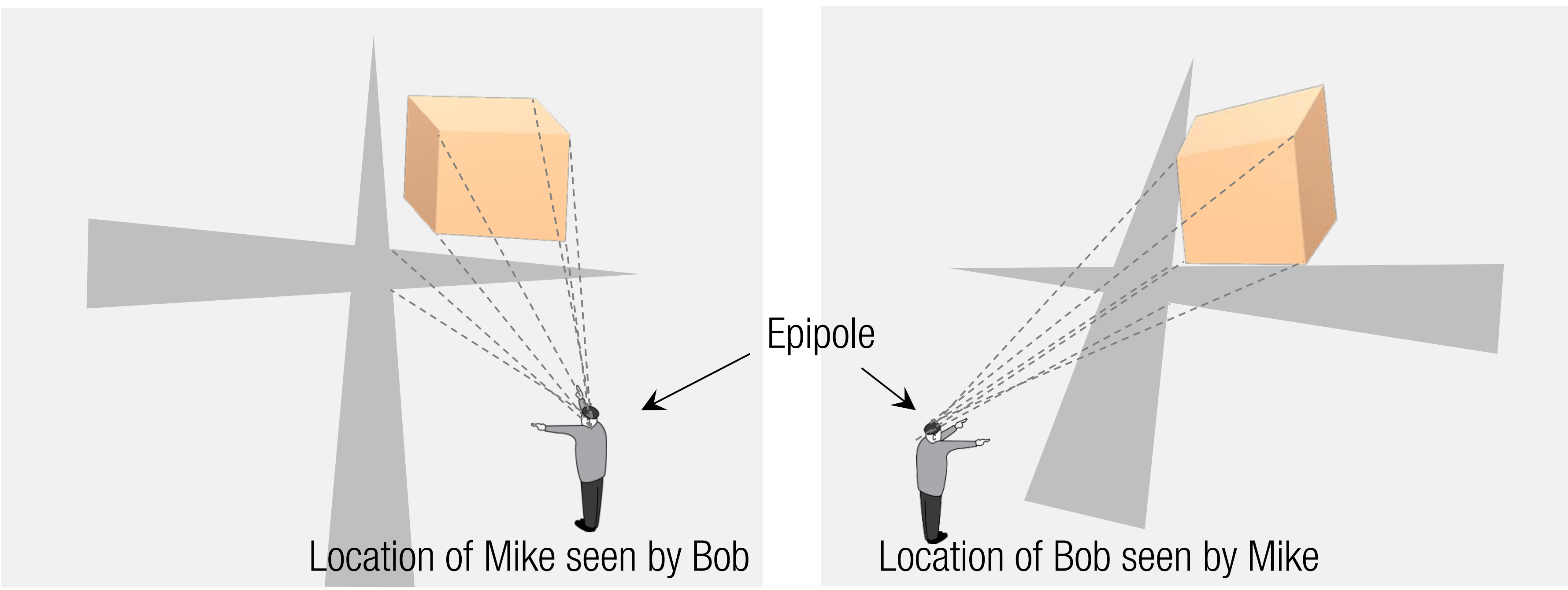

Bob
Mike



Bob’s view

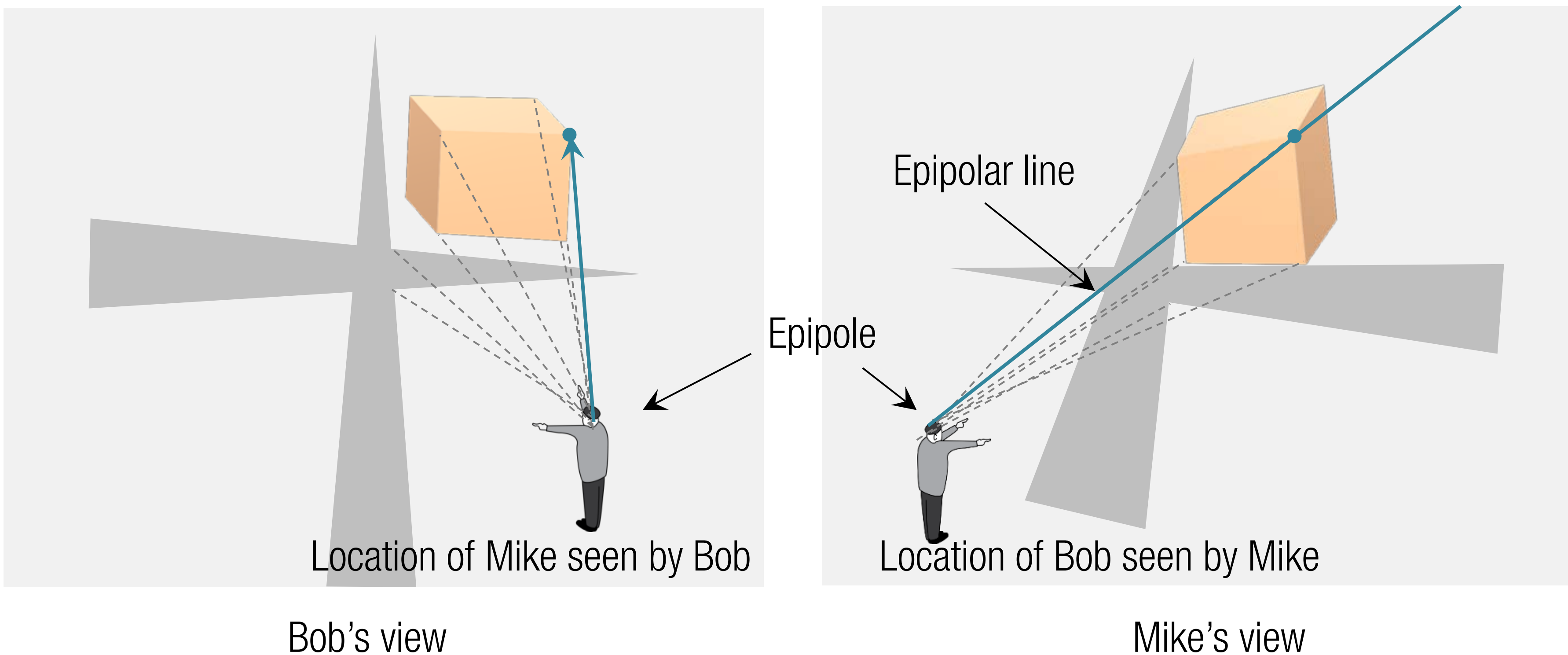


Mike’s view



Bob's view

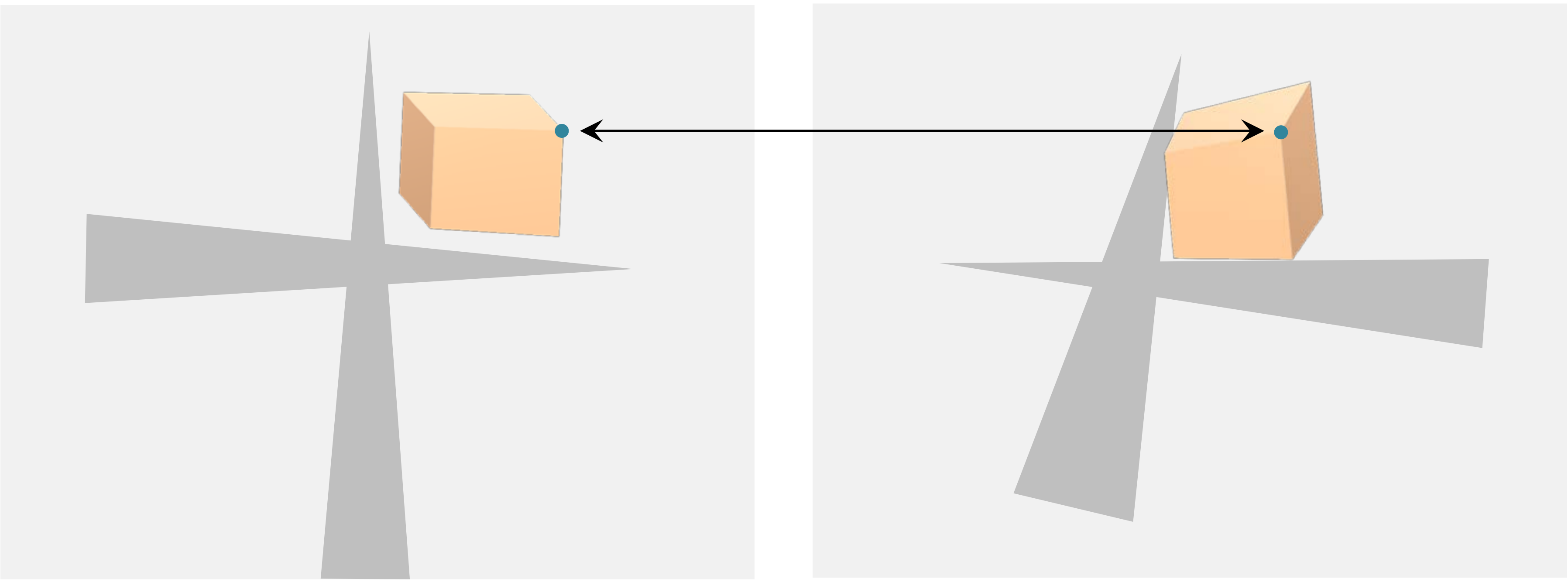
Mike's view



Observation:

Given a point in Bob's view, there exists a conjugate line passing the corresponding point in Mike's view.

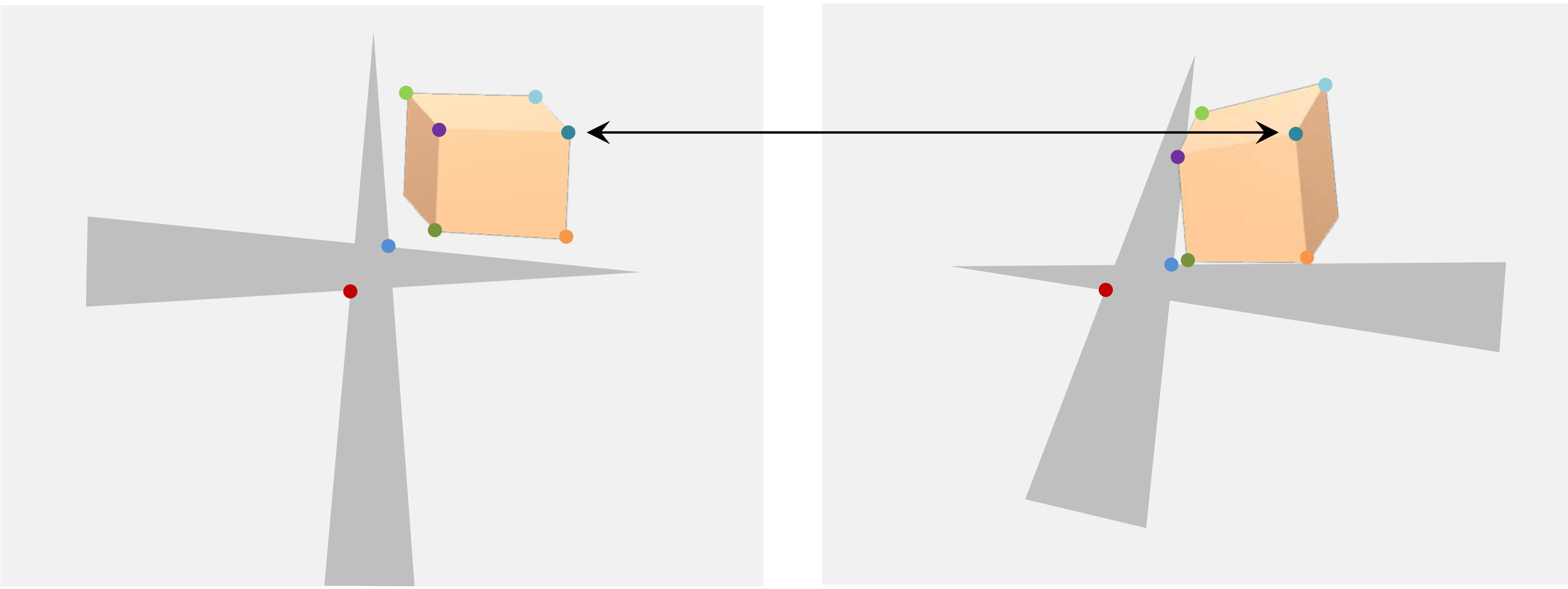
Point correspondence



Bob's view

Mike's view

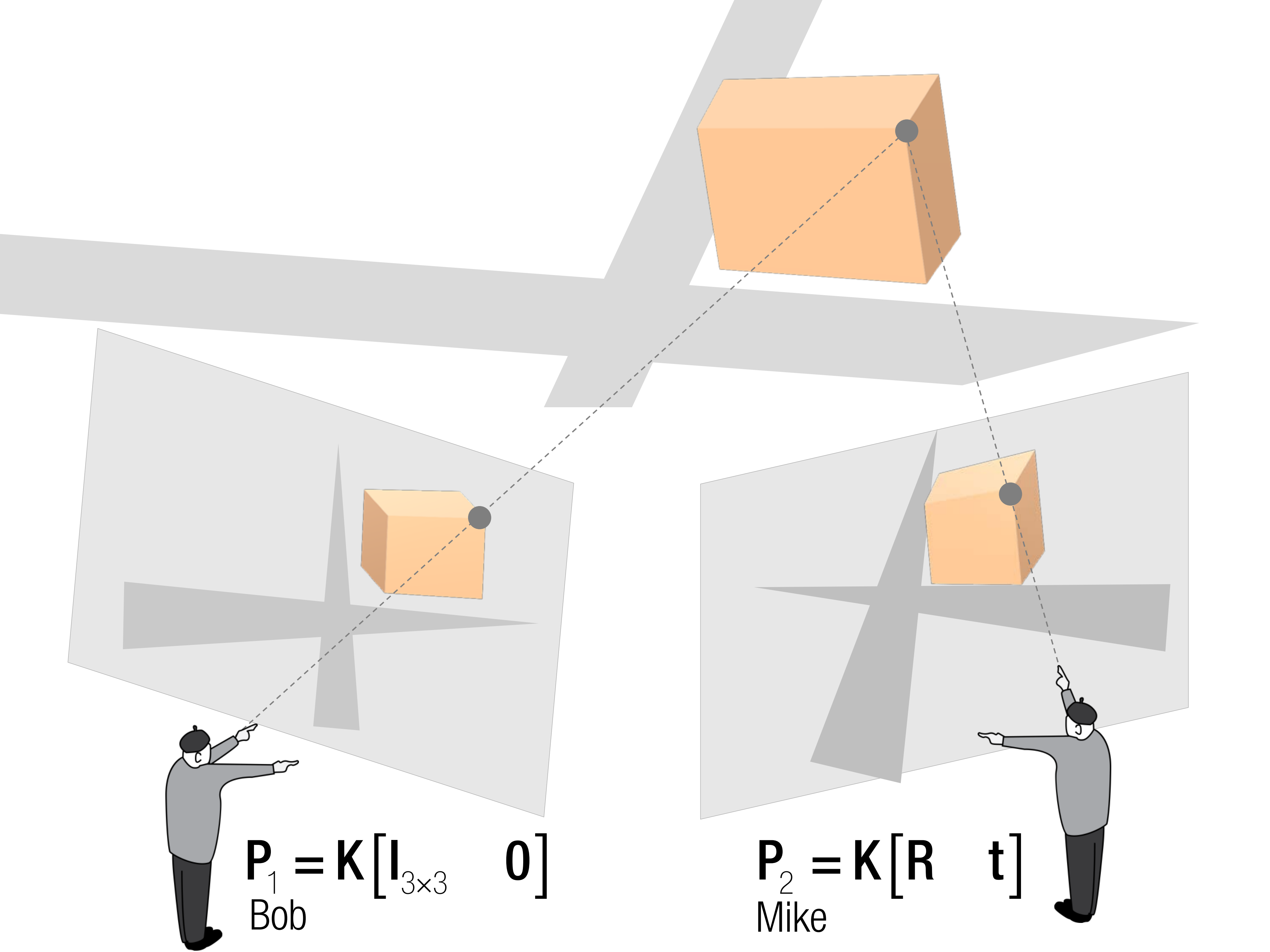
Point correspondence



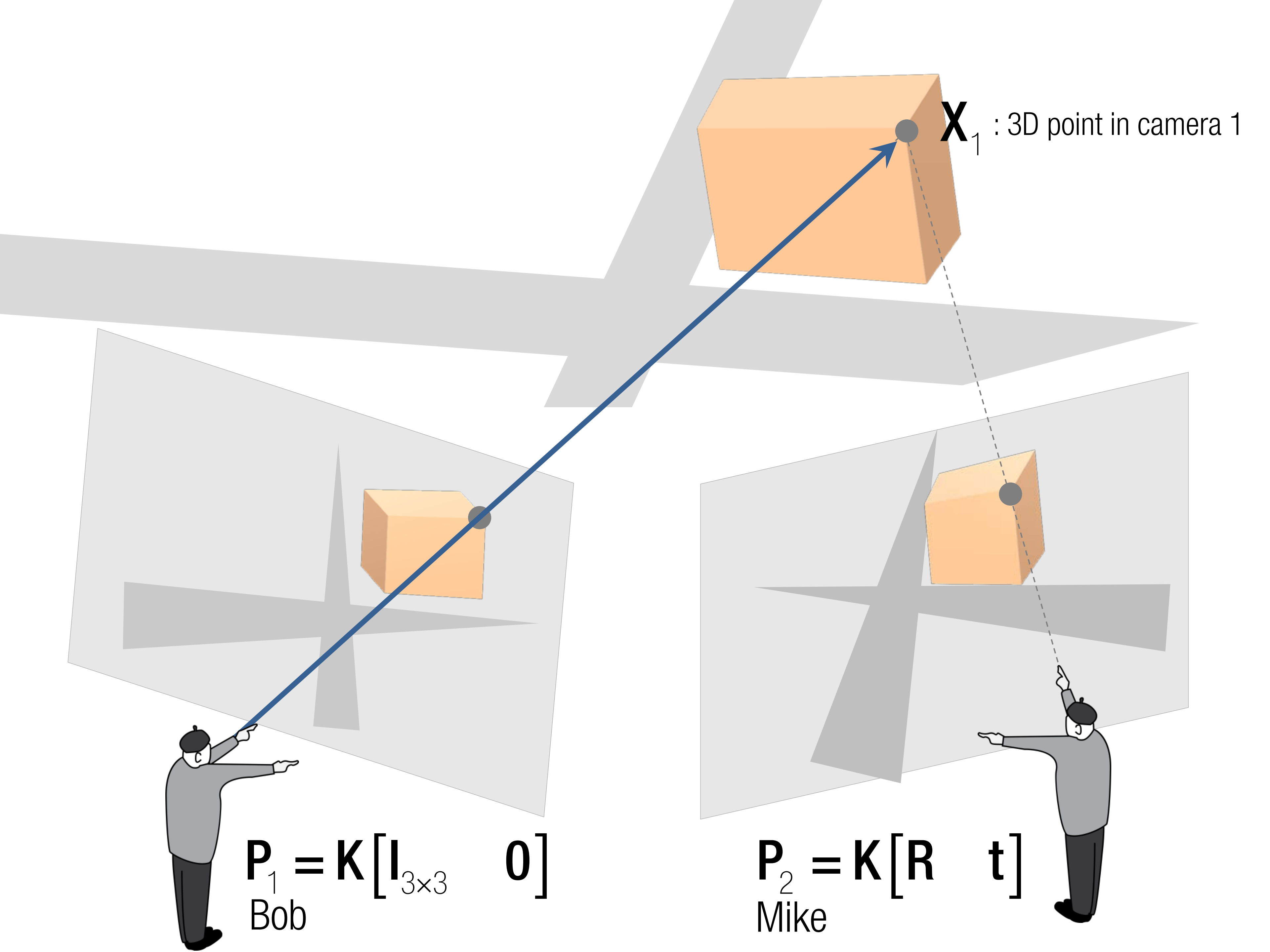
Bob's view

Mike's view

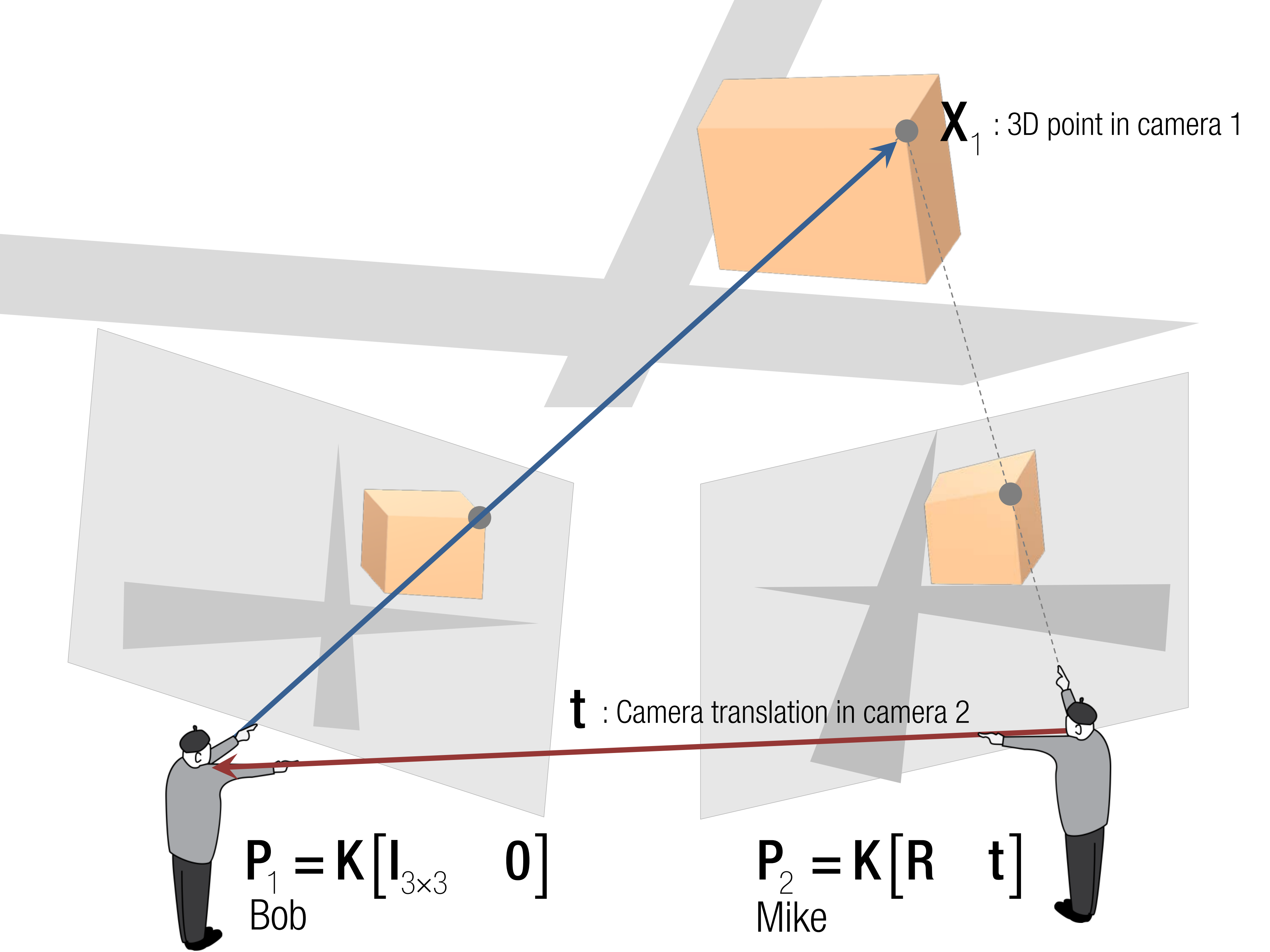
8 correspondences



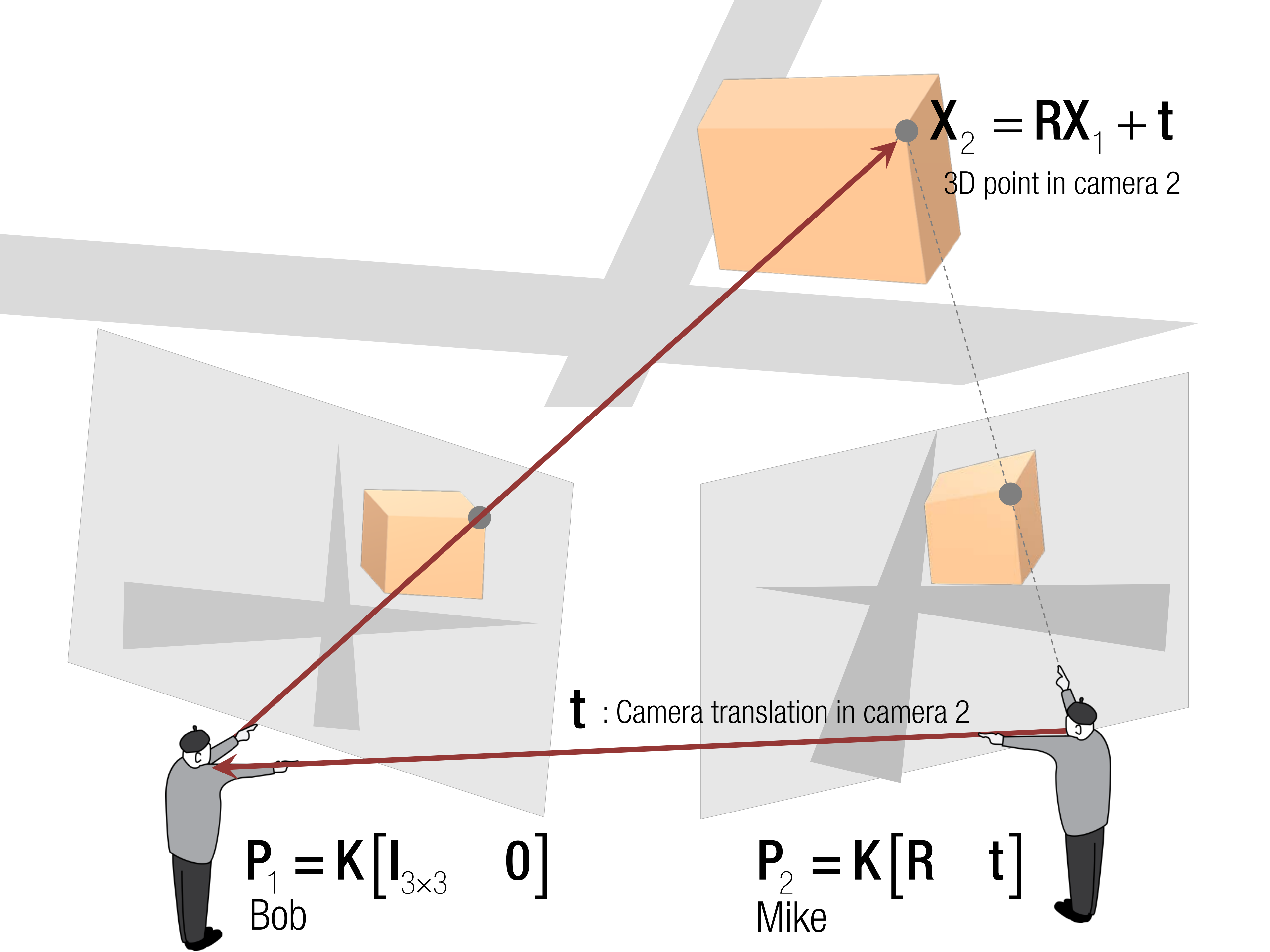
$P_1 = K[I_{3\times3} \quad 0]$
Bob
$P_2 = K[R \quad t]$
Mike



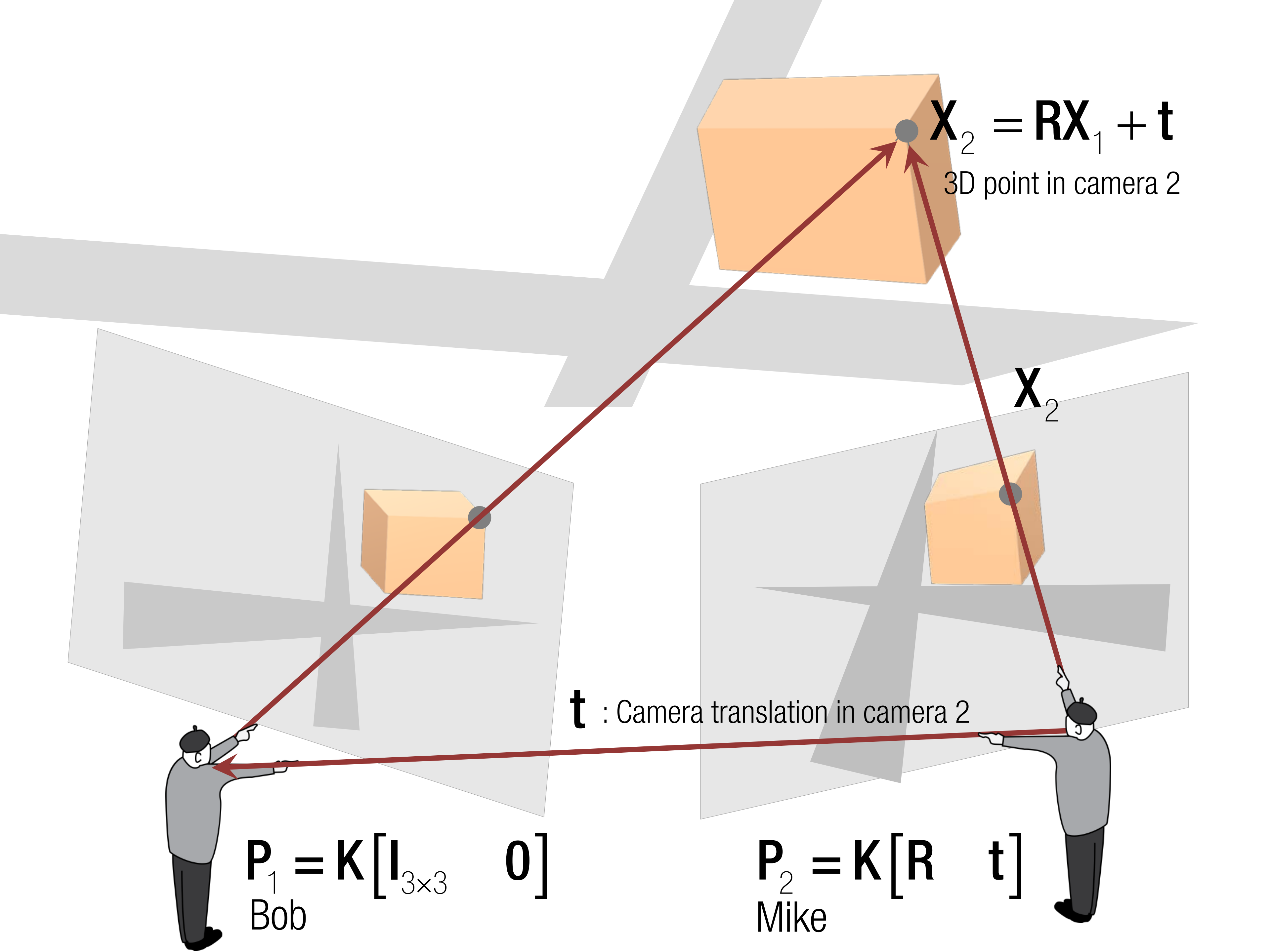
$\mathbf{X}_1$: 3D point in camera 1
$\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}\end{bmatrix}$
Bob
$\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$
Mike

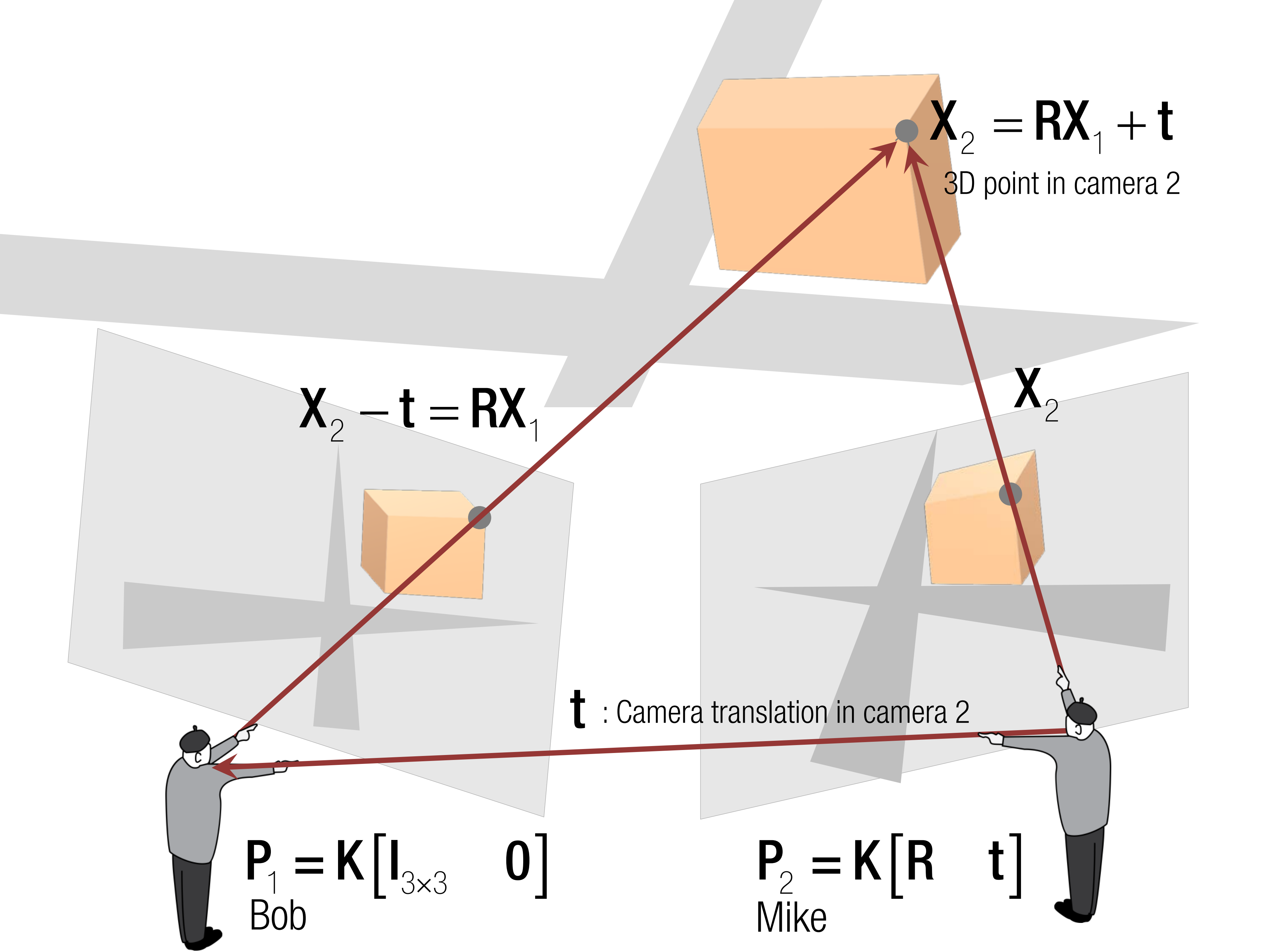


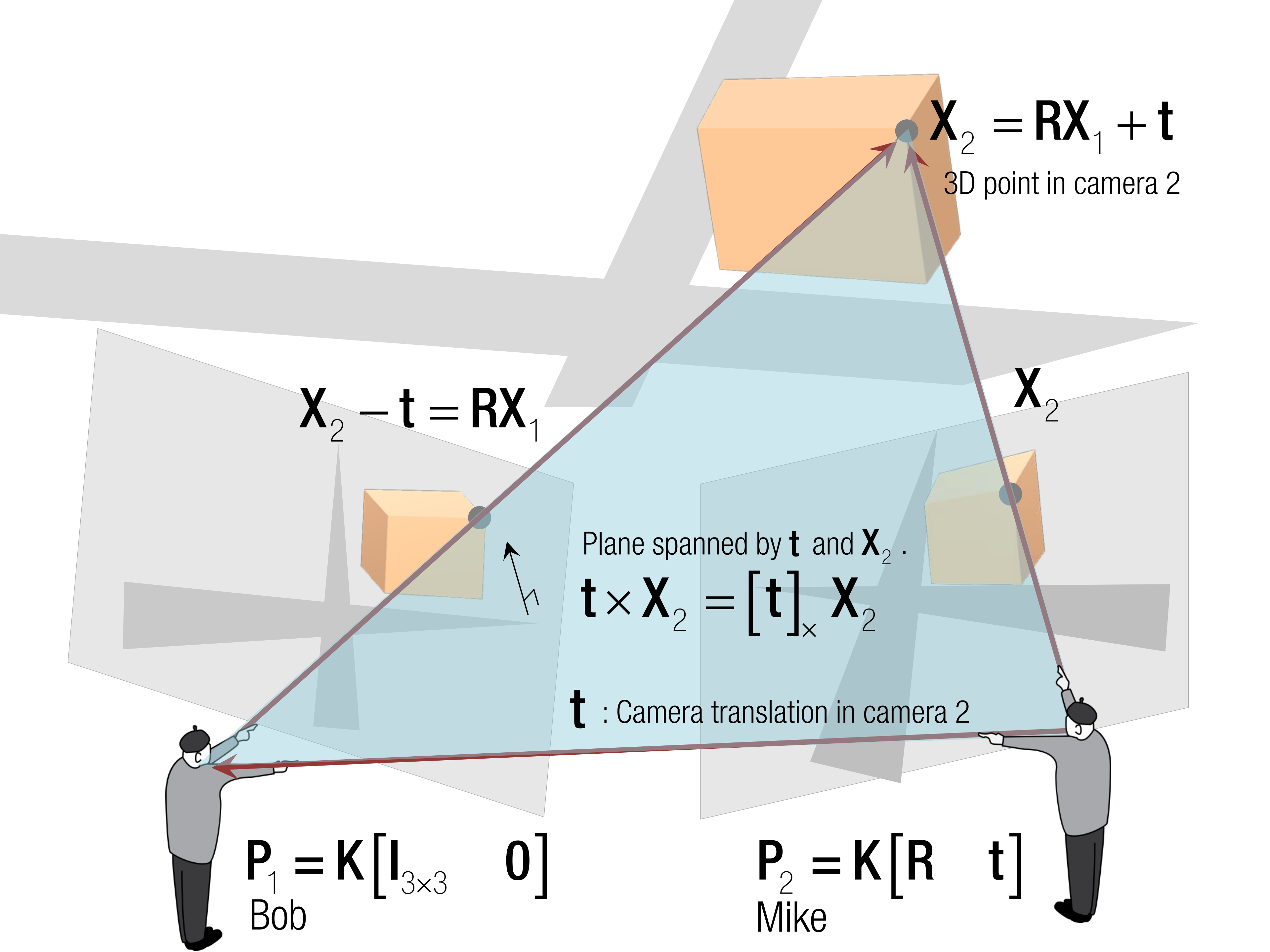
$\mathbf{X}_1$: 3D point in camera 1
$\mathbf{t}$: Camera translation in camera 2
$\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}\end{bmatrix}$
Bob
$\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$
Mike

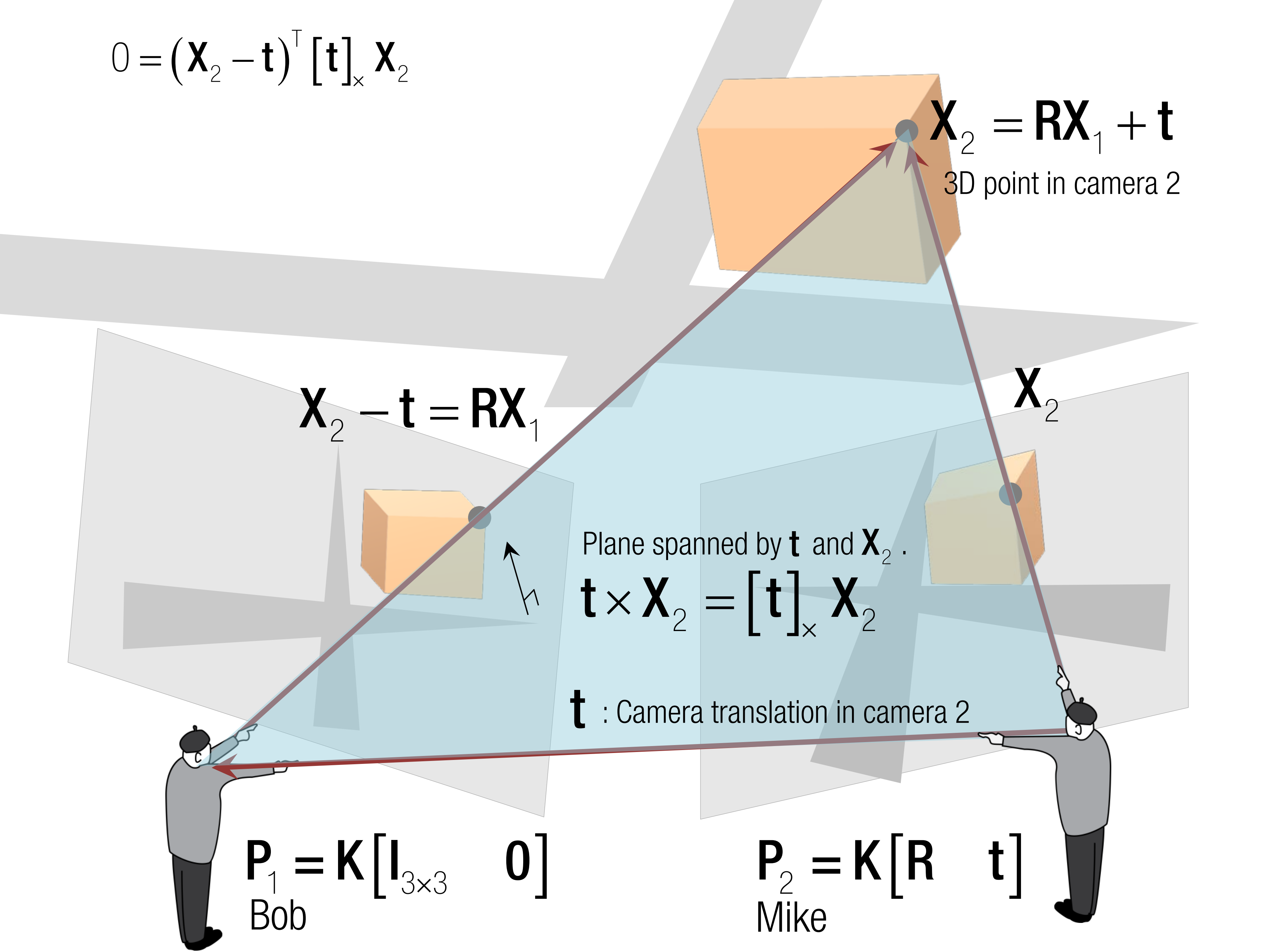


$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \mathbf{t}$
3D point in camera 2
$\mathbf{t}$: Camera translation in camera 2
$\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}\end{bmatrix}$
Bob
$\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$
Mike

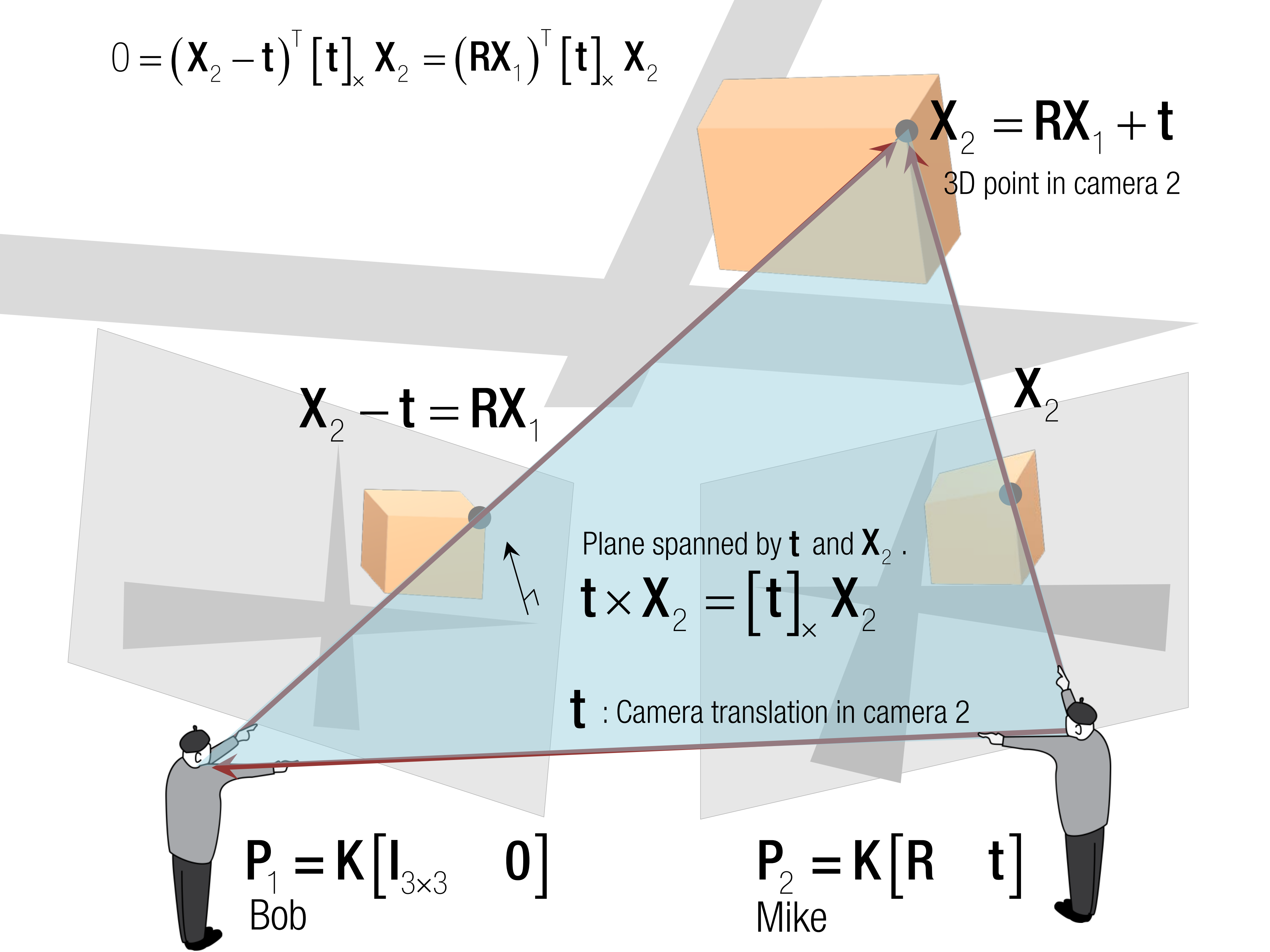

$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \mathbf{t}$
3D point in camera 2
$\mathbf{X}_2$
$\mathbf{t}$: Camera translation in camera 2
$\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}\end{bmatrix}$
Bob
$\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$
Mike

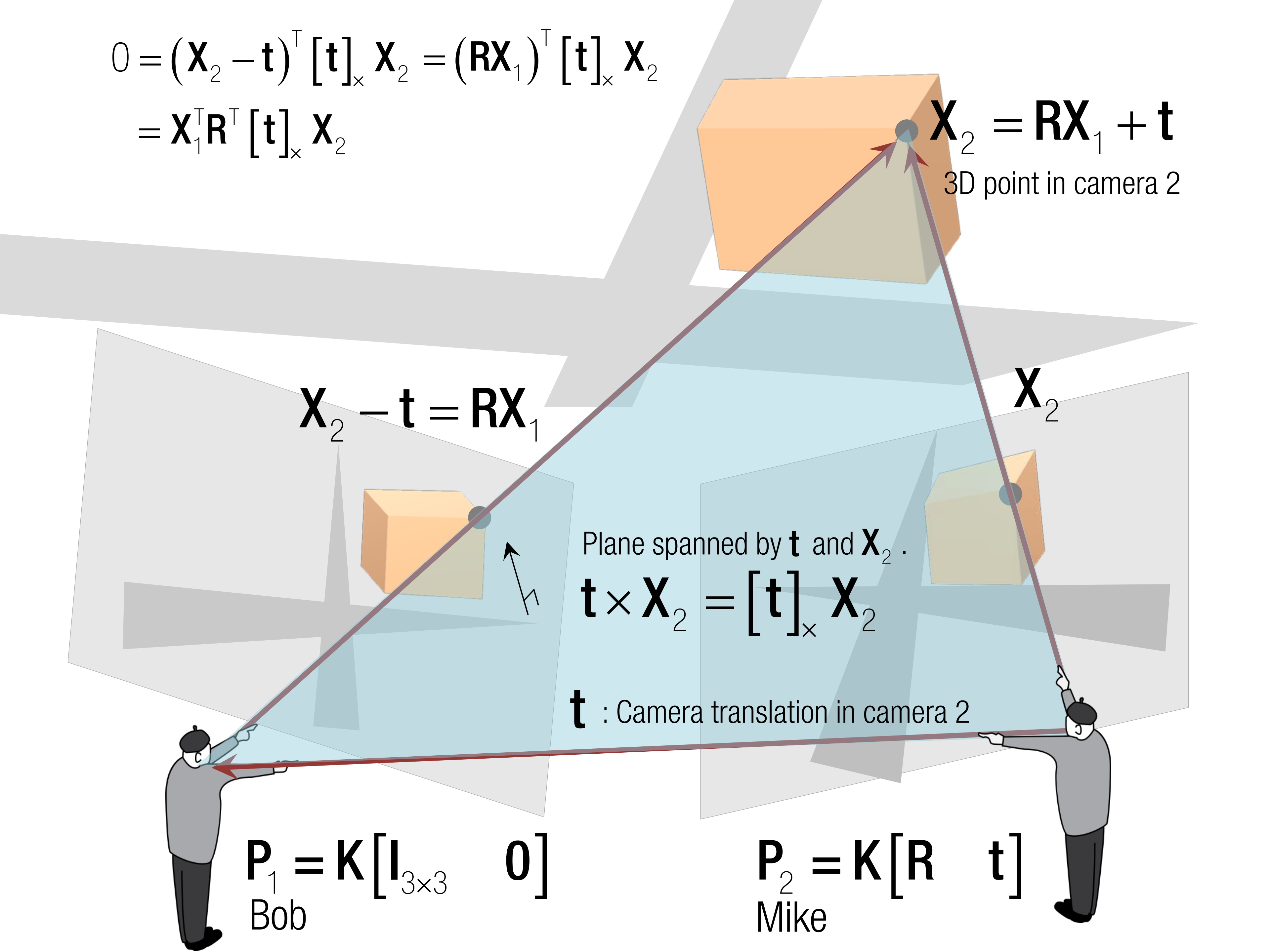

$X_2 = RX_1 + t$
3D point in camera 2
$X_2 - t = RX_1$
X_2
t : Camera translation in camera 2
$P_1 = K[I_{3\times3} \quad 0]$
Bob
$P_2 = K[R \quad t]$
Mike


$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \mathbf{t}$
3D point in camera 2
$\mathbf{X}_2 - \mathbf{t} = \mathbf{R}\mathbf{X}_1$
$\mathbf{X}_2$
Plane spanned by $\mathbf{t}$ and $\mathbf{X}_2$.
$\mathbf{t} \times \mathbf{X}_2 = [\mathbf{t}]_\times \mathbf{X}_2$
$\mathbf{t}$: Camera translation in camera 2
$\mathbf{P}_1 = \mathbf{K}[\mathbf{I}_{3\times3} \quad \mathbf{0}]$
Bob
$\mathbf{P}_2 = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$
Mike



$0 = (\mathbf{X}_2 - \mathbf{t})^{\top} [\mathbf{t}]_{\times} \mathbf{X}_2$
$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \mathbf{t}$
3D point in camera 2
$\mathbf{X}_2 - \mathbf{t} = \mathbf{R}\mathbf{X}_1$
$\mathbf{X}_2$
Plane spanned by $\mathbf{t}$ and $\mathbf{X}_2$.
$\mathbf{t} \times \mathbf{X}_2 = [\mathbf{t}]_{\times} \mathbf{X}_2$
$\mathbf{t}$: Camera translation in camera 2
$\mathbf{P}_1 = \mathbf{K}[\mathbf{I}_{3\times3} \quad \mathbf{0}]$
Bob
$\mathbf{P}_2 = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$
Mike


$0 = (\mathbf{X}_2 - \mathbf{t})^{\top} [\mathbf{t}]_{\times} \mathbf{X}_2 = (\mathbf{R}\mathbf{X}_1)^{\top} [\mathbf{t}]_{\times} \mathbf{X}_2$
$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \mathbf{t}$
3D point in camera 2
$\mathbf{X}_2 - \mathbf{t} = \mathbf{R}\mathbf{X}_1$
$\mathbf{X}_2$
Plane spanned by $\mathbf{t}$ and $\mathbf{X}_2$.
$\mathbf{t} \times \mathbf{X}_2 = [\mathbf{t}]_{\times} \mathbf{X}_2$
$\mathbf{t}$: Camera translation in camera 2
$\mathbf{P}_1 = \mathbf{K}[\mathbf{I}_{3\times3} \quad \mathbf{0}]$
Bob
$\mathbf{P}_2 = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$
Mike


$0 = (\mathbf{X}_2 - \mathbf{t})^\top [\mathbf{t}]_\times \mathbf{X}_2 = (\mathbf{R}\mathbf{X}_1)^\top [\mathbf{t}]_\times \mathbf{X}_2$
$= \mathbf{X}_1^\top \mathbf{R}^\top [\mathbf{t}]_\times \mathbf{X}_2$
$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \mathbf{t}$
3D point in camera 2
$\mathbf{X}_2 - \mathbf{t} = \mathbf{R}\mathbf{X}_1$
$\mathbf{X}_2$
Plane spanned by $\mathbf{t}$ and $\mathbf{X}_2$.
$\mathbf{t} \times \mathbf{X}_2 = [\mathbf{t}]_\times \mathbf{X}_2$
$\mathbf{t}$: Camera translation in camera 2
$\mathbf{P}_1 = \mathbf{K}[\mathbf{I}_{3\times3} \quad \mathbf{0}]$
Bob
$\mathbf{P}_2 = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$
Mike

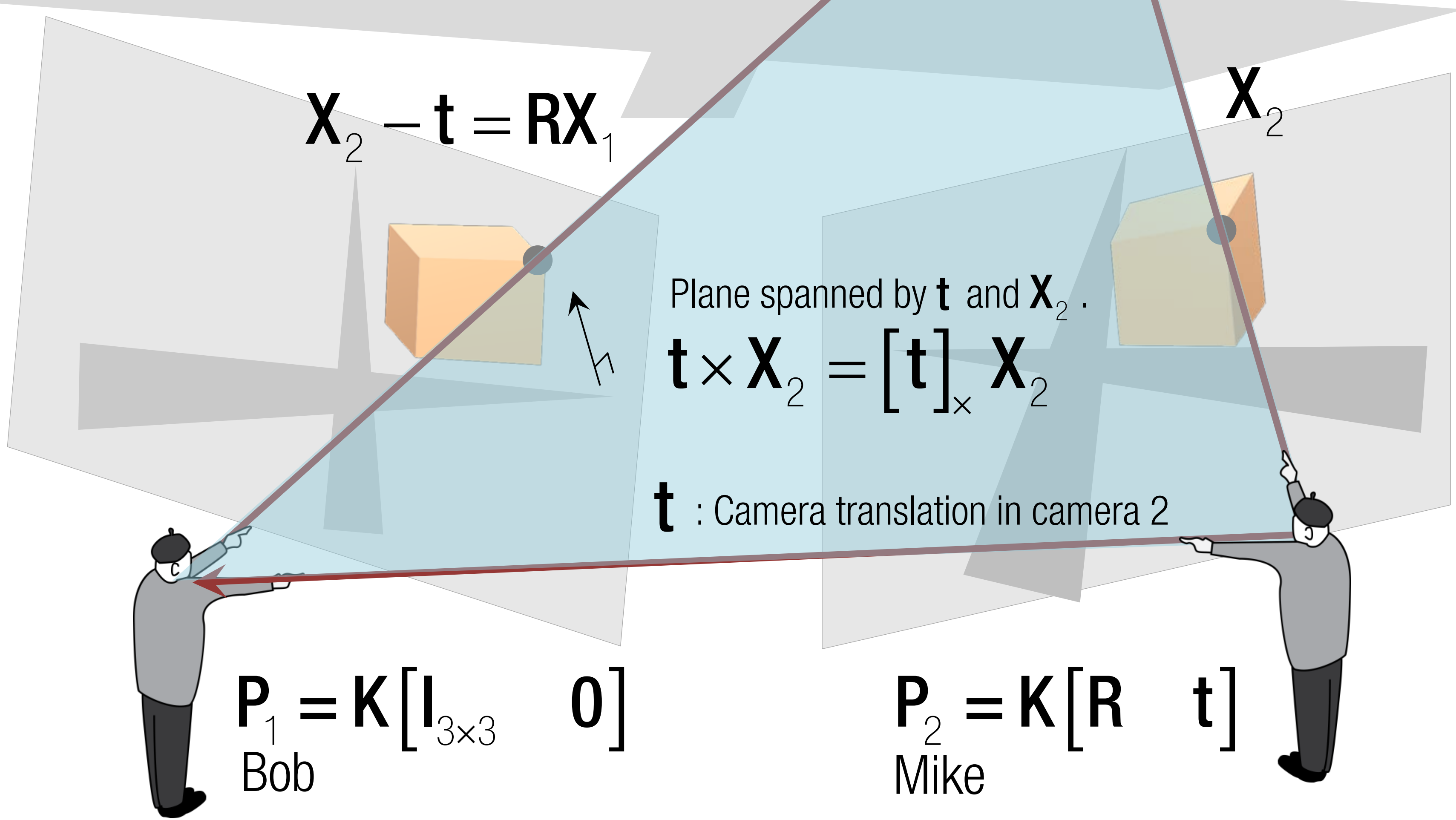
$$0 = (\mathbf{X}_2 - \mathbf{t})^{\mathsf{T}} [\mathbf{t}]_\times \mathbf{X}_2 = (\mathbf{R}\mathbf{X}_1)^{\mathsf{T}} [\mathbf{t}]_\times \mathbf{X}_2$$

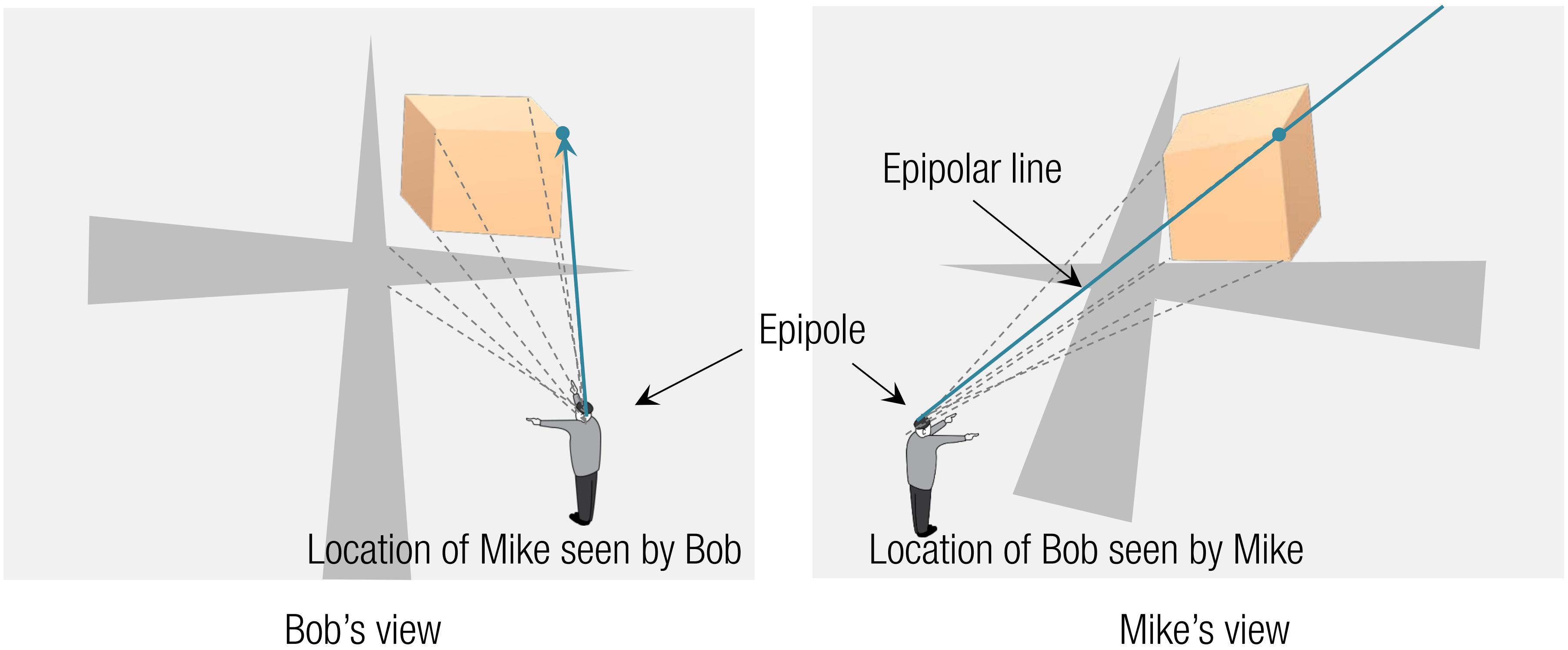
$$= \mathbf{X}_1^{\mathsf{T}} \mathbf{R}^{\mathsf{T}} [\mathbf{t}]_\times \mathbf{X}_2$$

$$= -\mathbf{X}_2^{\mathsf{T}} [\mathbf{t}]_\times \mathbf{R}\mathbf{X}_1 = -\mathbf{X}_2^{\mathsf{T}} \mathbf{E} \mathbf{X}_1$$

Essential matrix

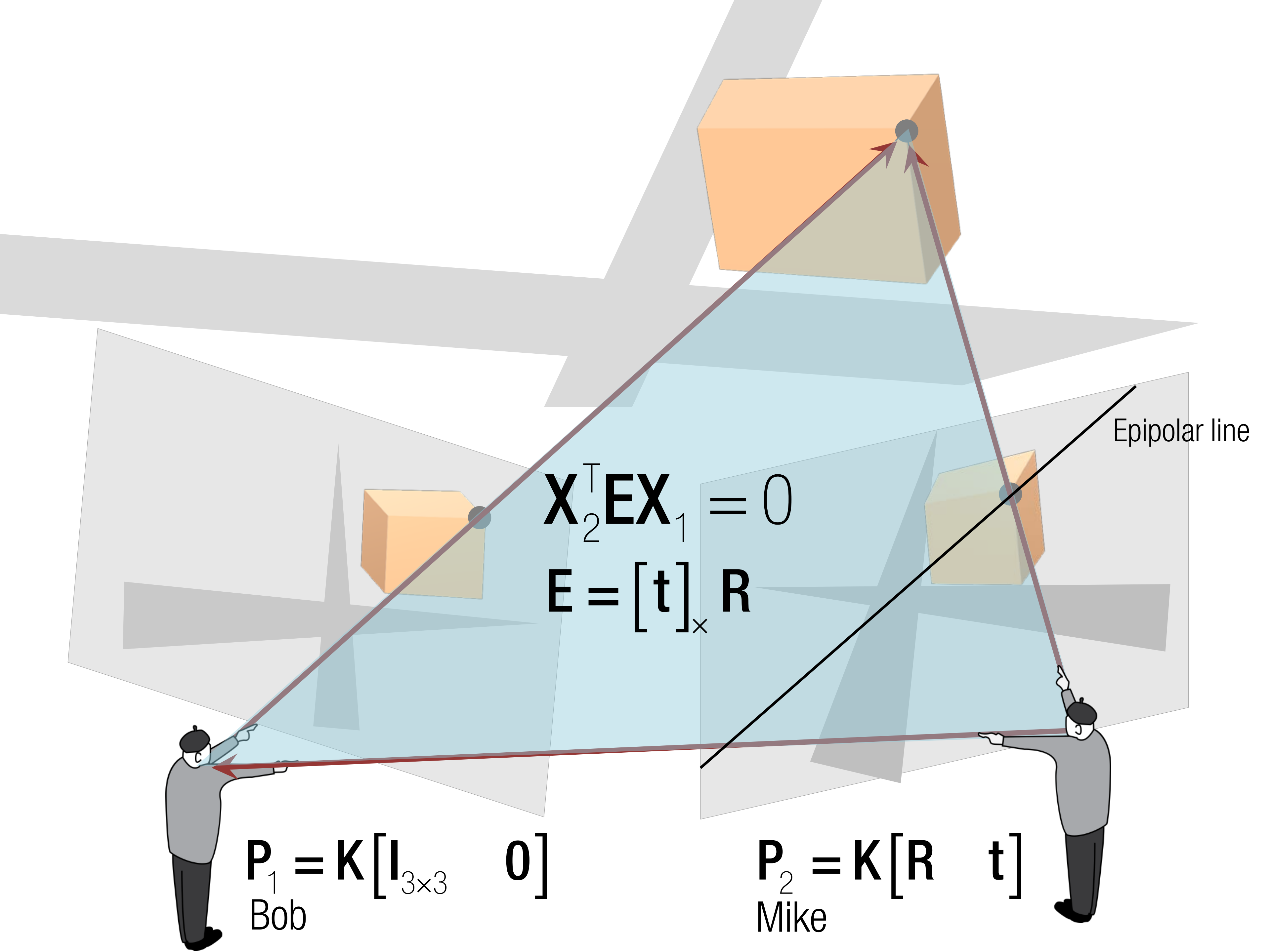
$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$$



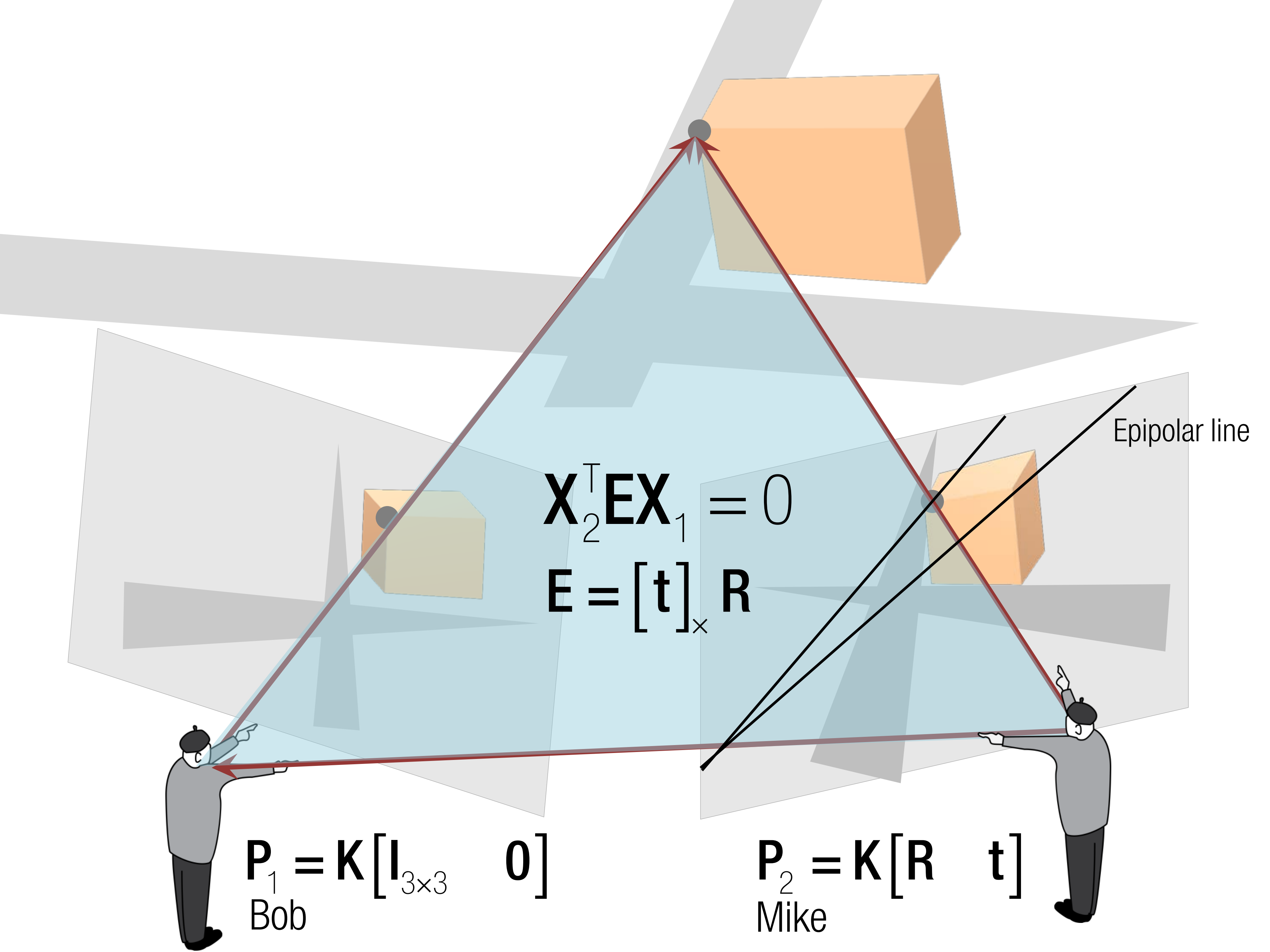


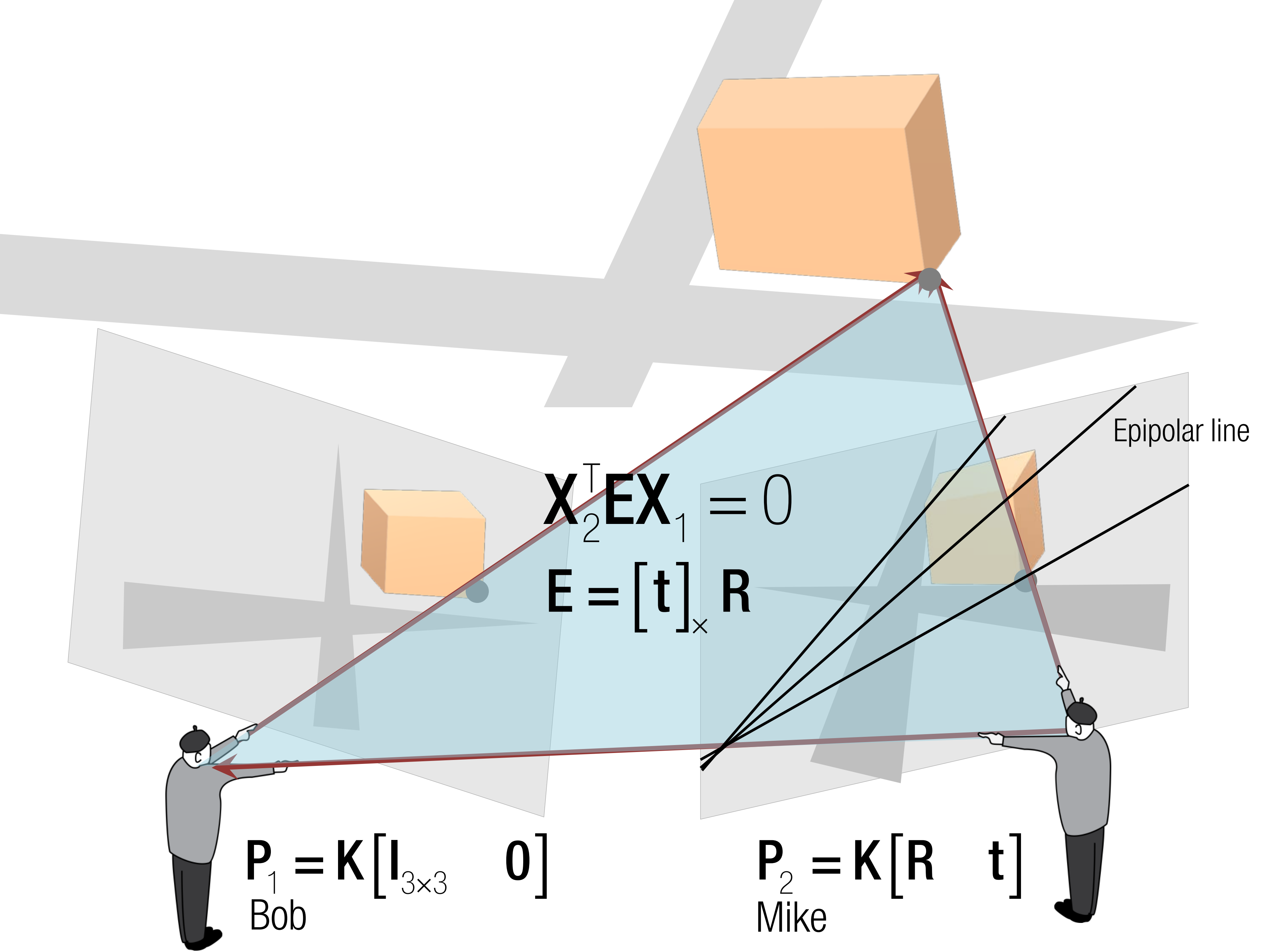
Observation:

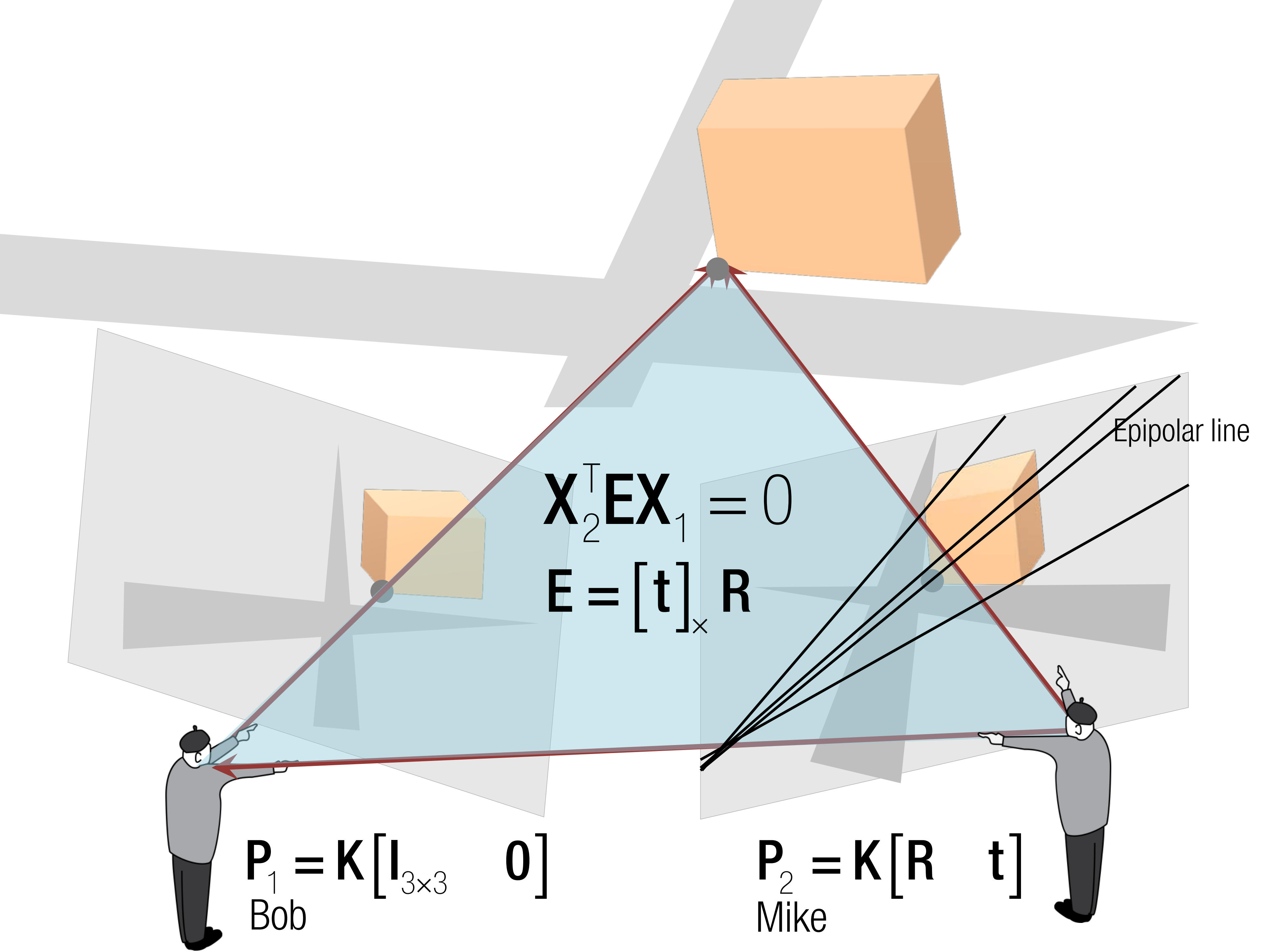
Given a point in Bob’s view, there exists a conjugate line passing the corresponding point in Mike’s view.

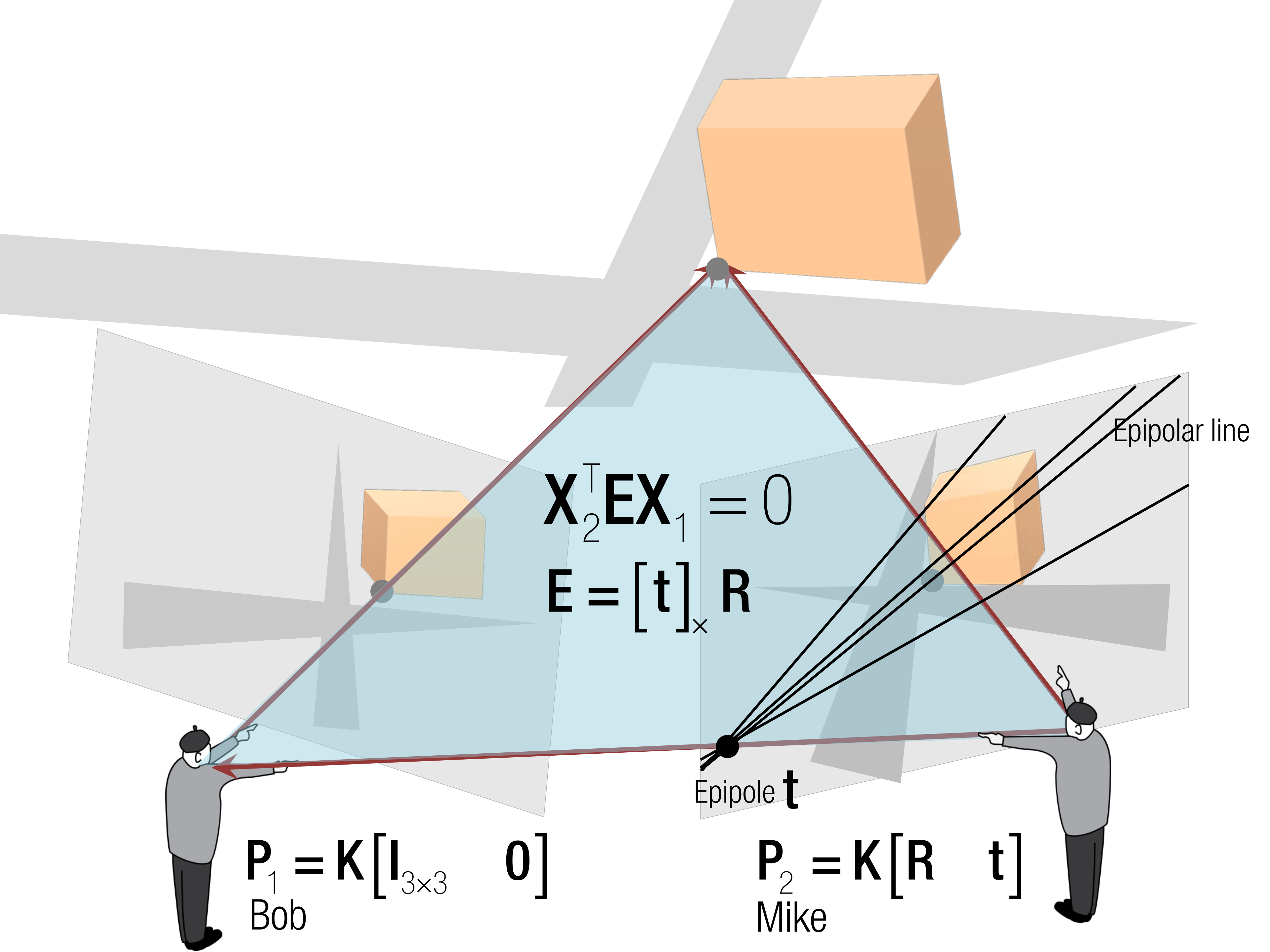


Epipolar line
$X_2^\top E X_1 = 0$
$E = [t]_\times R$
$P_1 = K[I_{3\times3} \quad 0]$
Bob
$P_2 = K[R \quad t]$
Mike


Epipolar line
$X_2^{\top} E X_1 = 0$
$E = [t]_{\times} R$
$P_1 = K[I_{3\times3} \quad 0]$
Bob
$P_2 = K[R \quad t]$
Mike

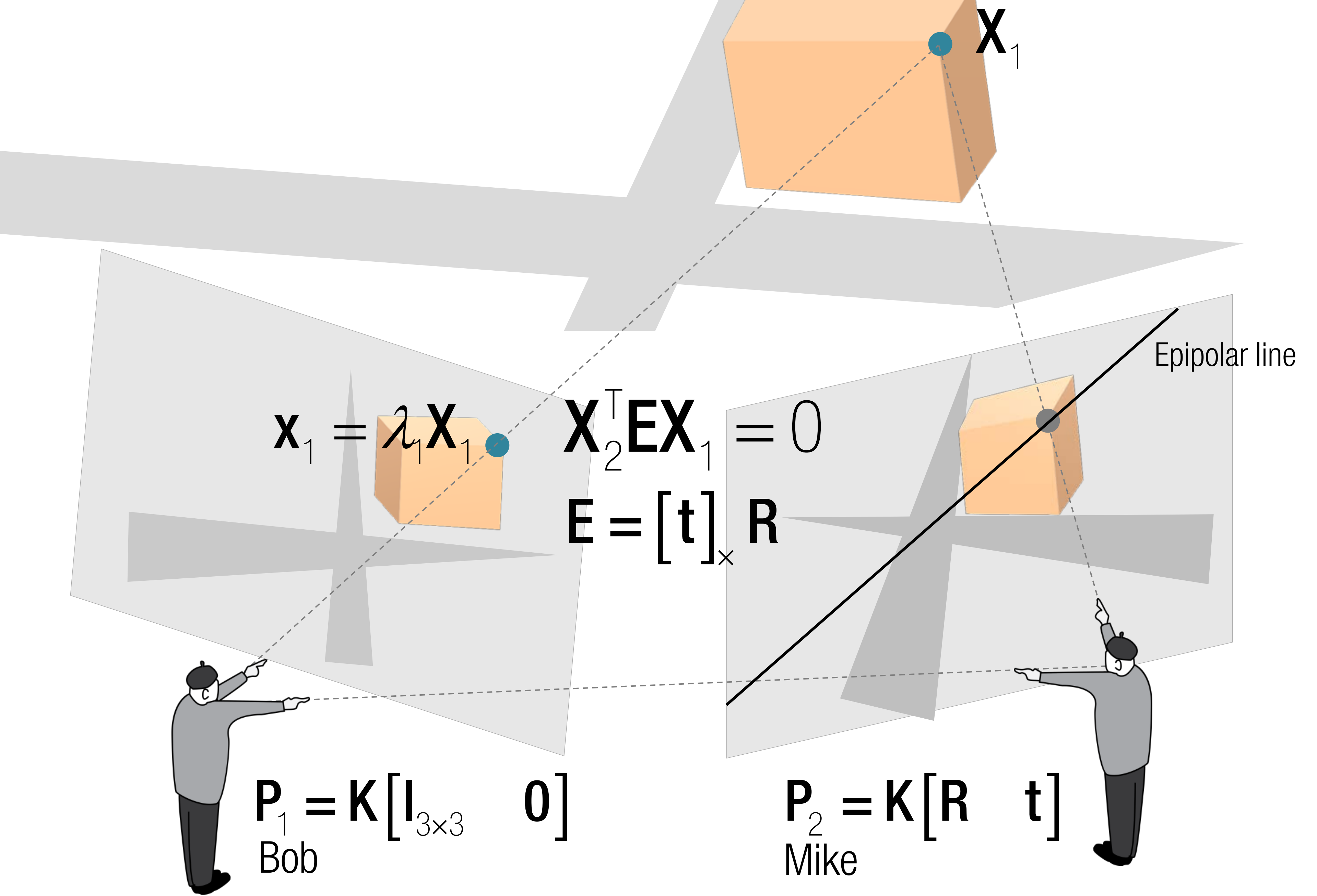

Epipolar line
$X_2^{T} E X_1 = 0$
$E = [t]_{\times} R$
$P_1 = K[I_{3\times 3} \quad 0]$
Bob
$P_2 = K[R \quad t]$
Mike


Epipolar line
$X_2^{\top} E X_1 = 0$
$E = [t]_{\times} R$
$P_1 = K[I_{3\times3} \quad 0]$
Bob
$P_2 = K[R \quad t]$
Mike

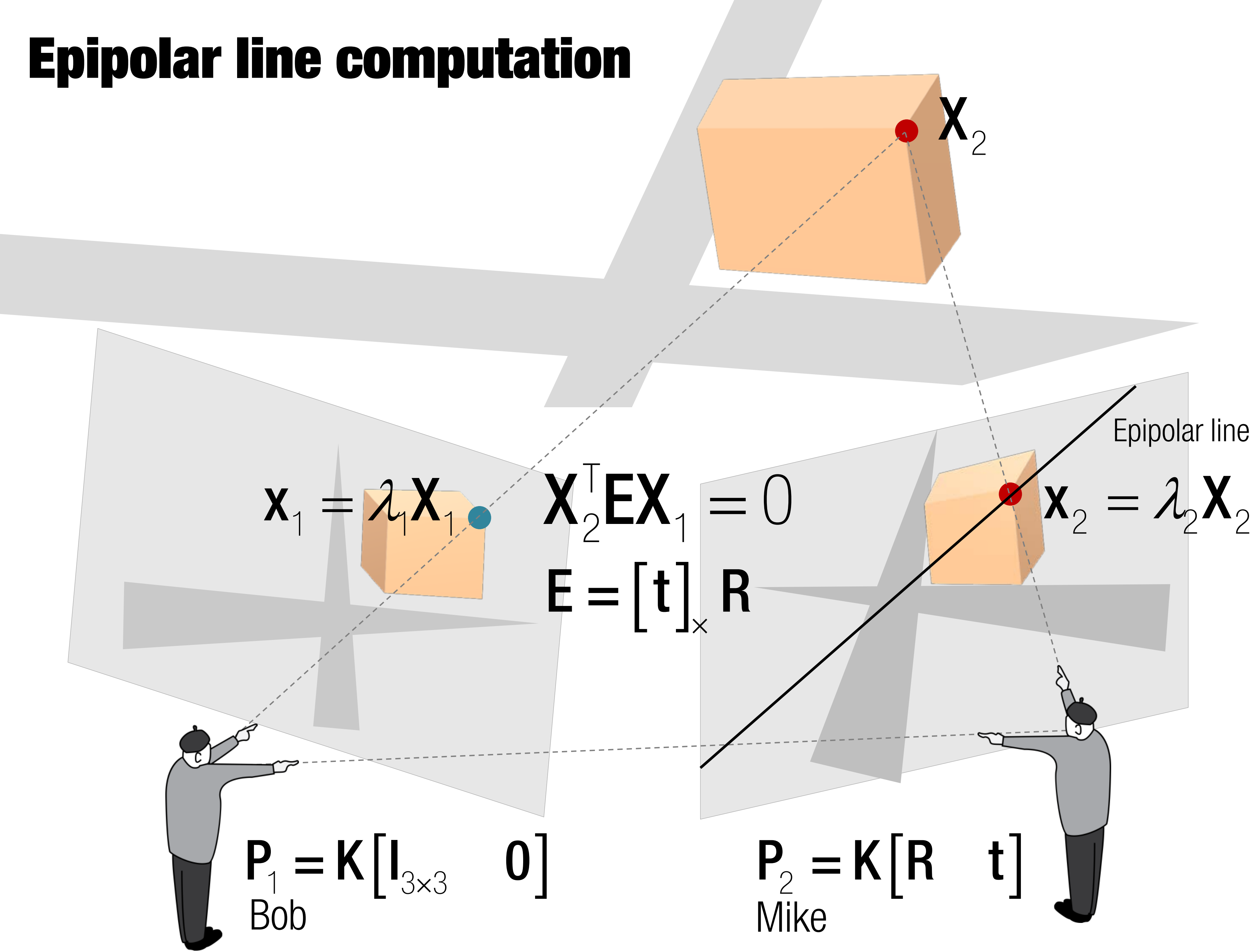


$X_2^T E X_1 = 0$
$E = [t]_\times R$
Epipolar line
Epipole t
$P_1 = K[I_{3\times3} \quad 0]$
Bob
$P_2 = K[R \quad t]$
Mike

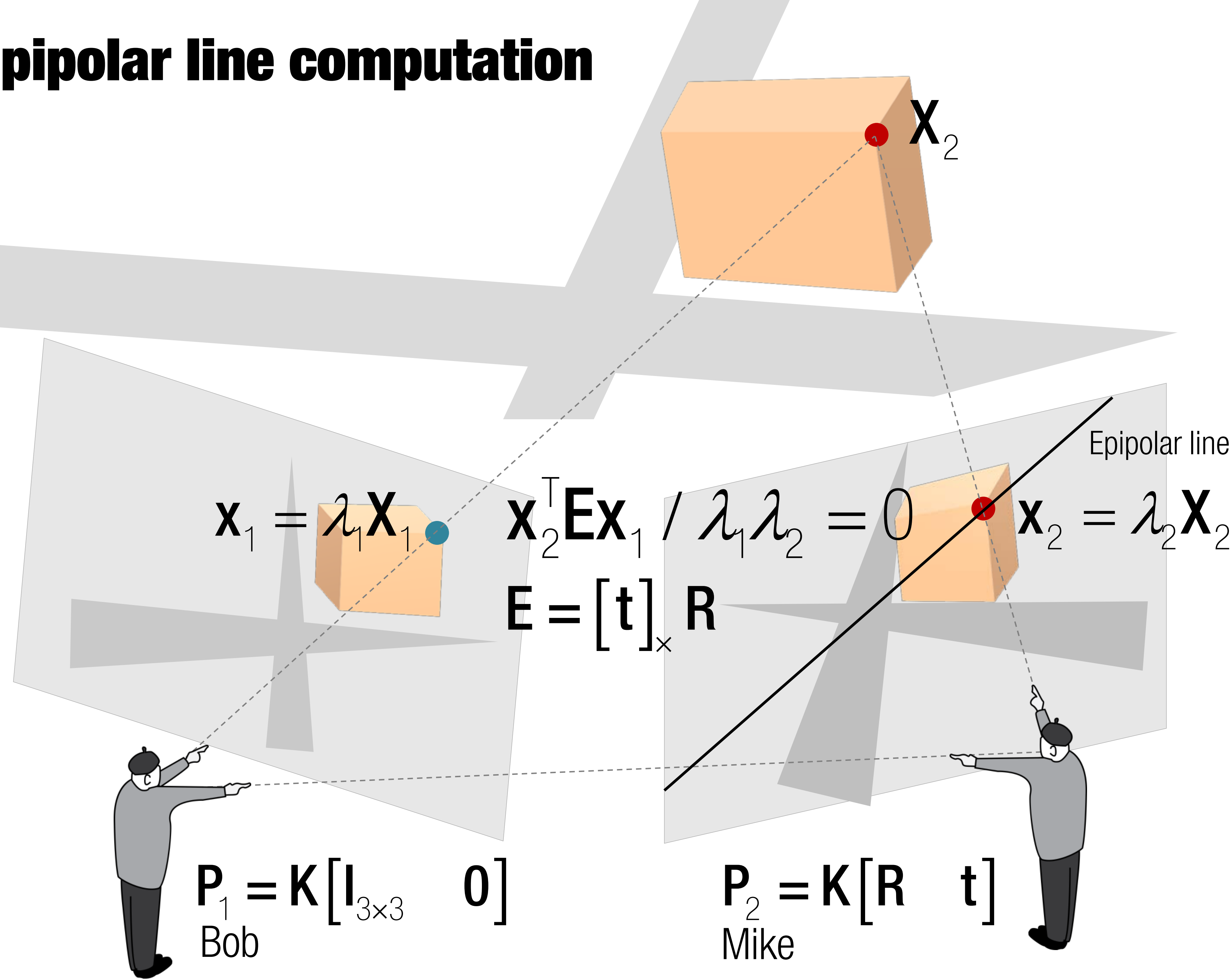
Epipolar line computation



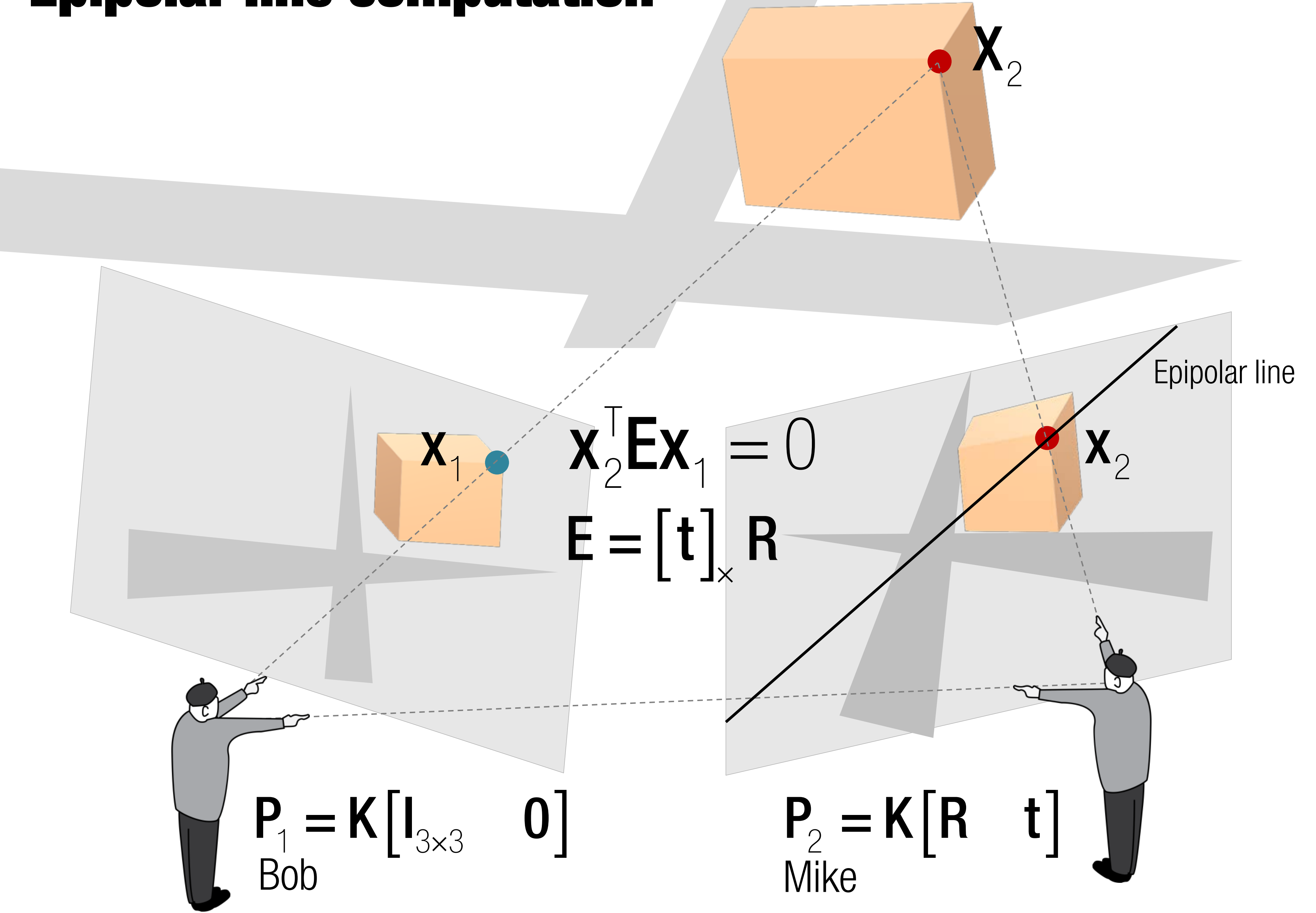
Epipolar line computation



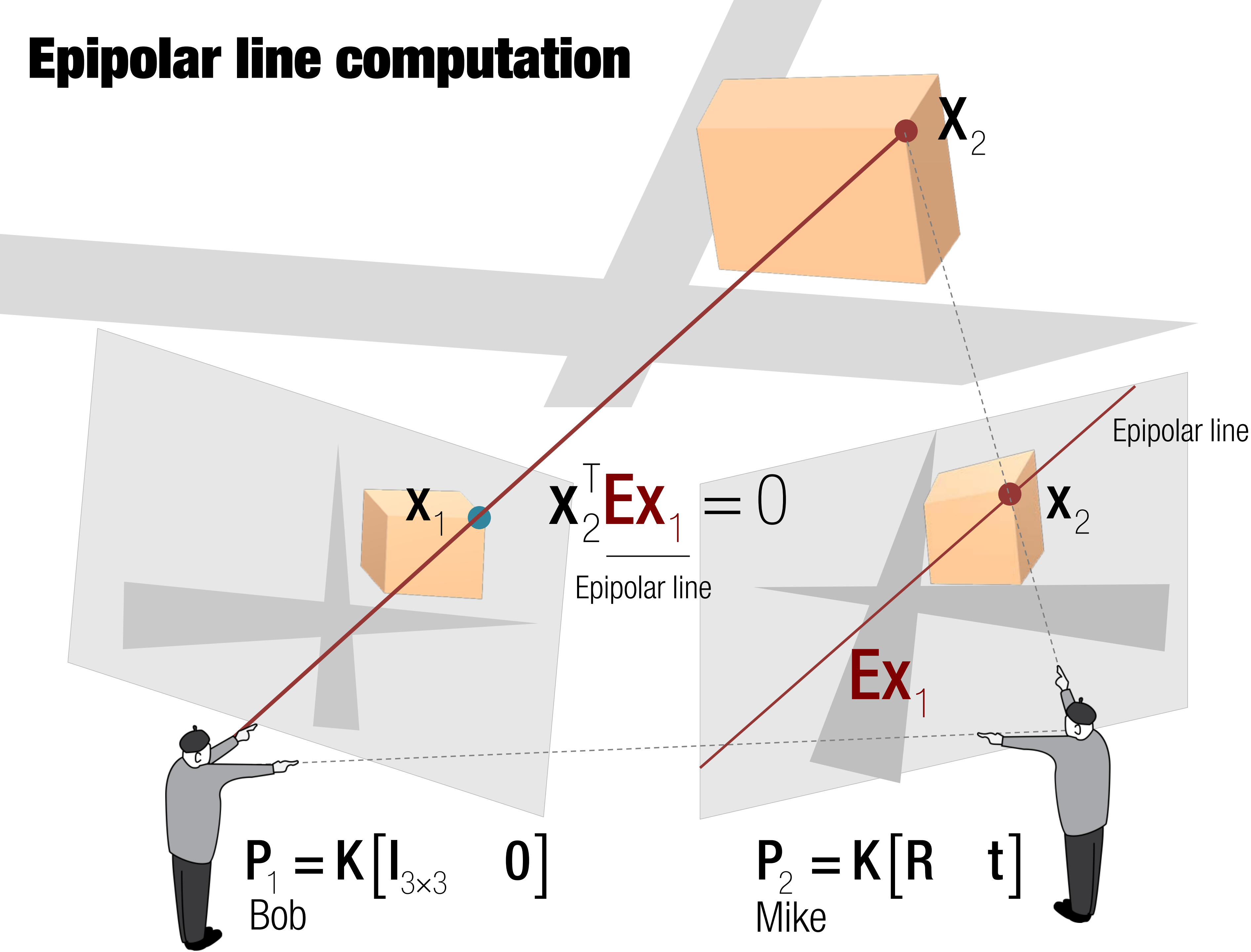
Epipolar line computation



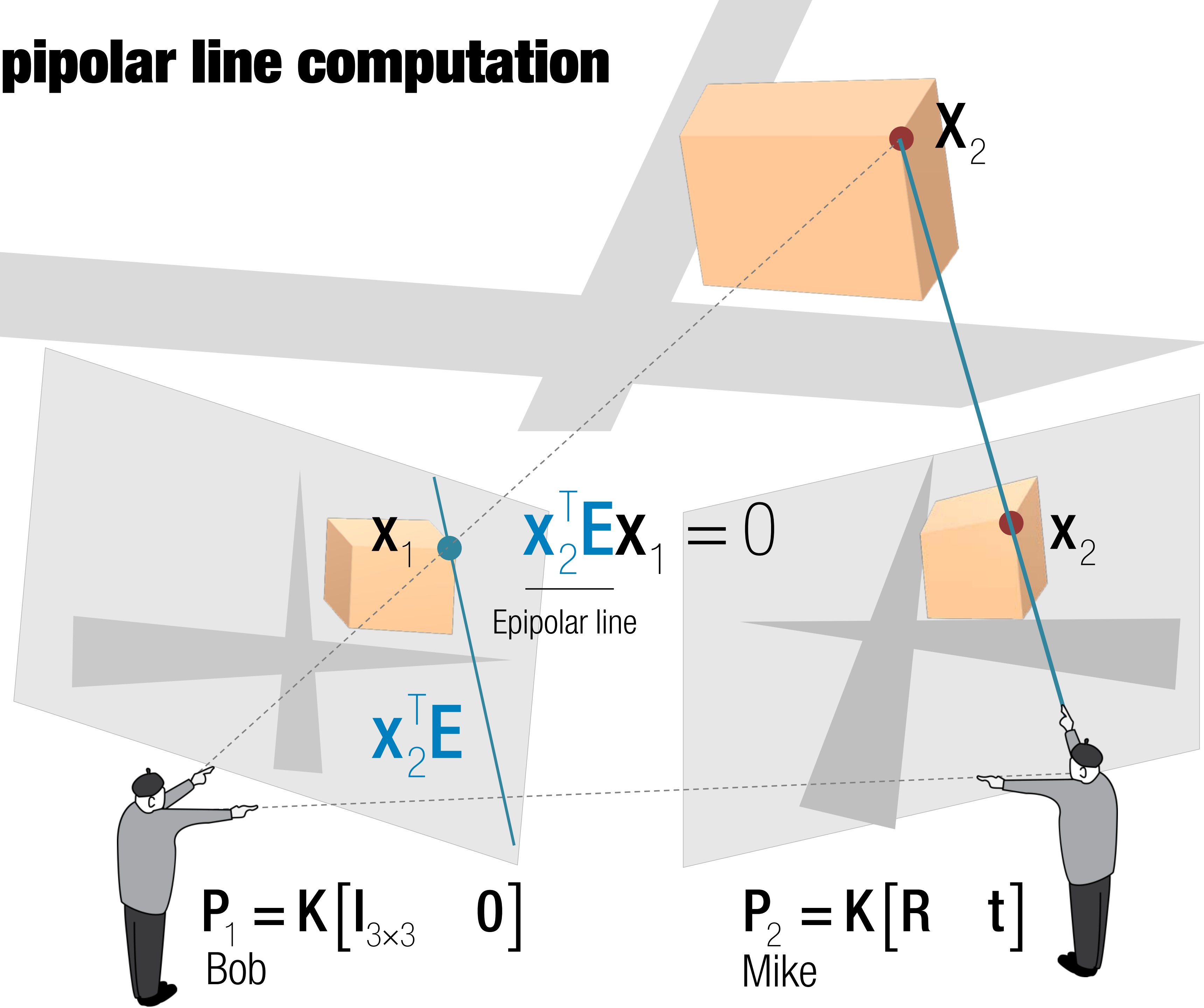
Epipolar line computation



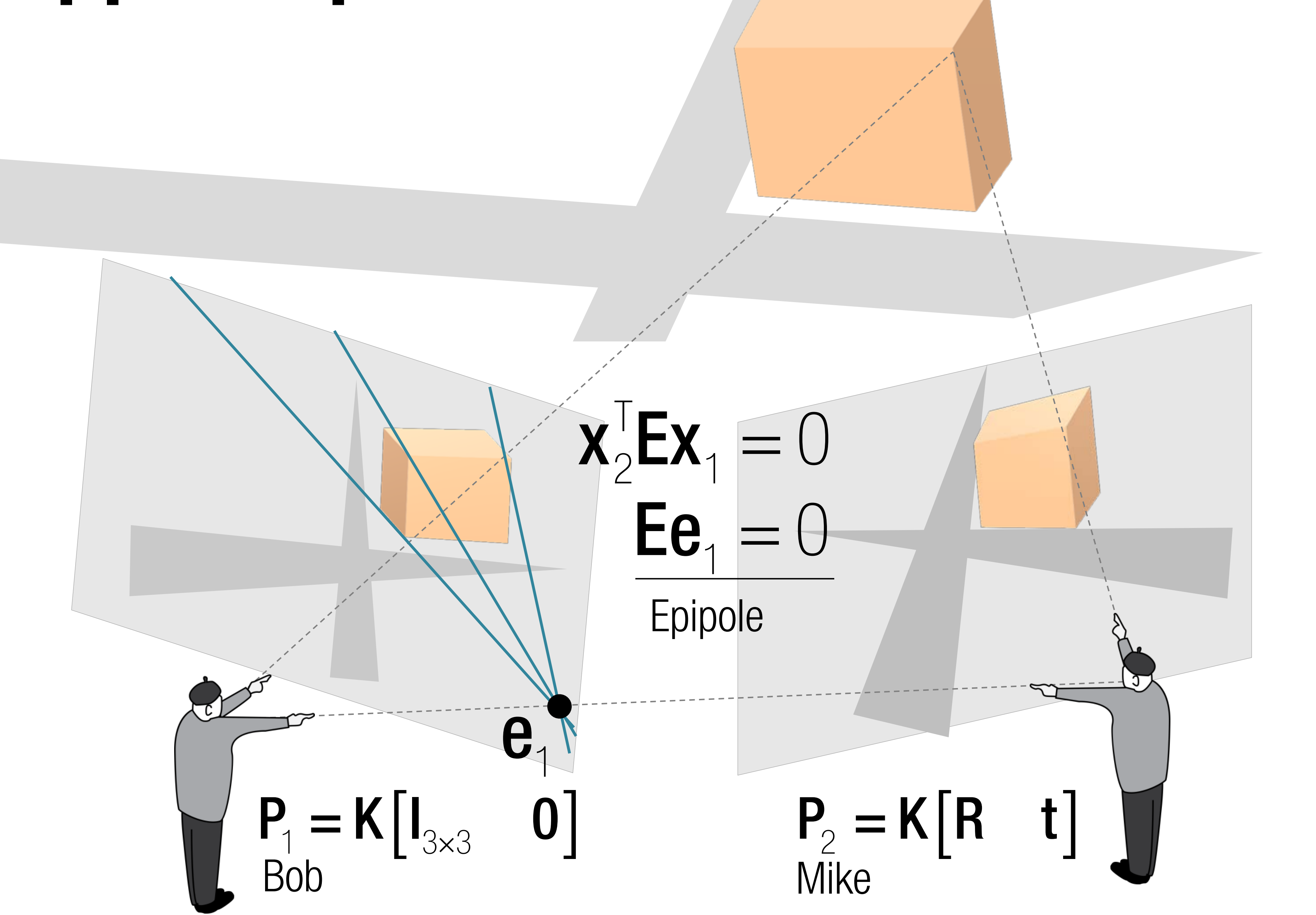
Epipolar line computation



Epipolar line computation



Epipole computation



$$\mathbf{x}_2^{T}\mathbf{E}\mathbf{x}_1 = 0$$

$$\underline{\mathbf{E}\mathbf{e}_1 = 0}$$

Epipole

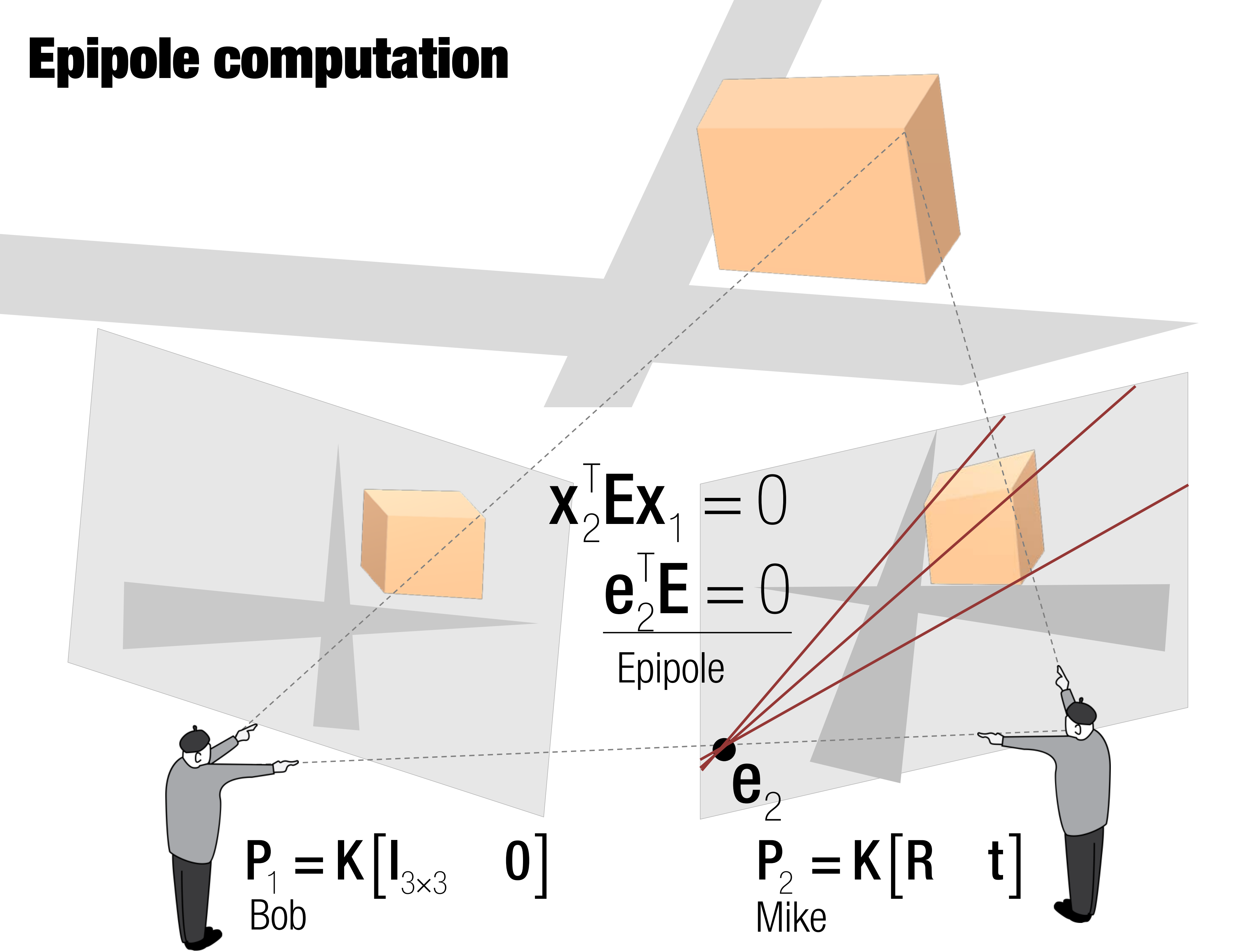
$$\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times 3} & \mathbf{0}\end{bmatrix}$$

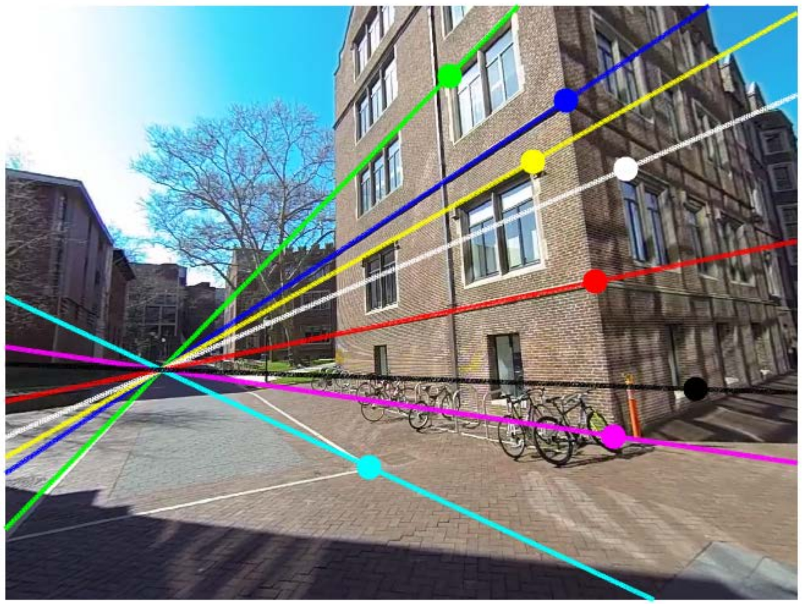
Bob

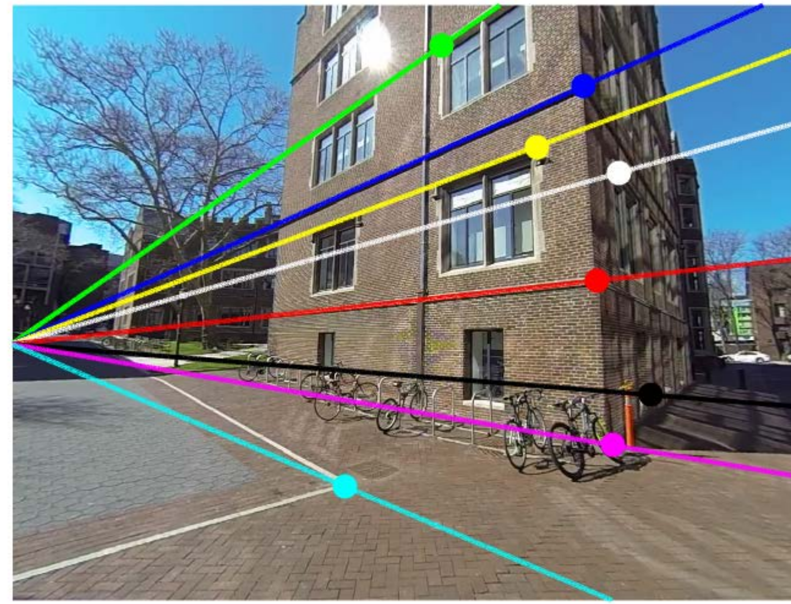
$$\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$$

Mike

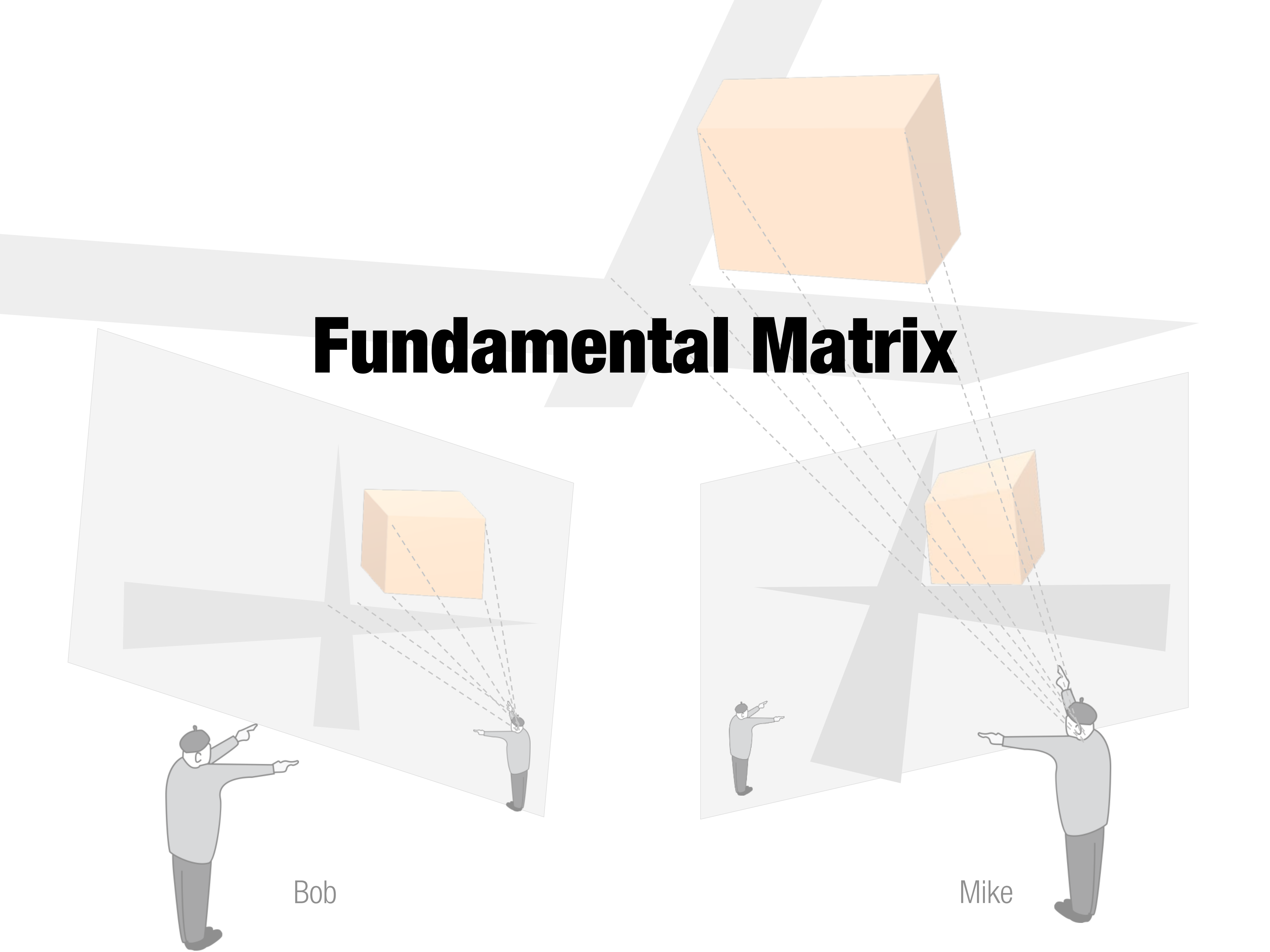
Epipole computation

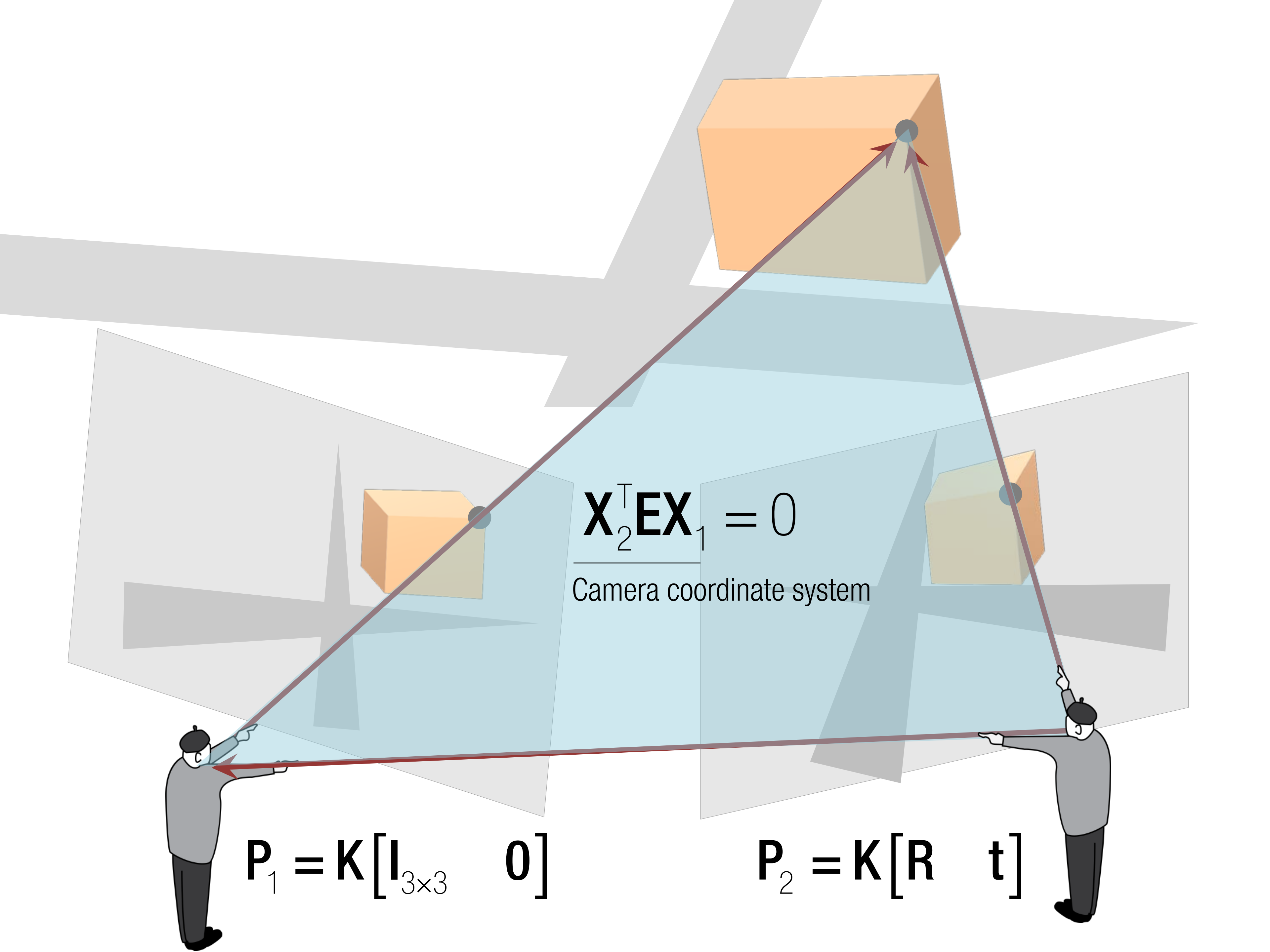


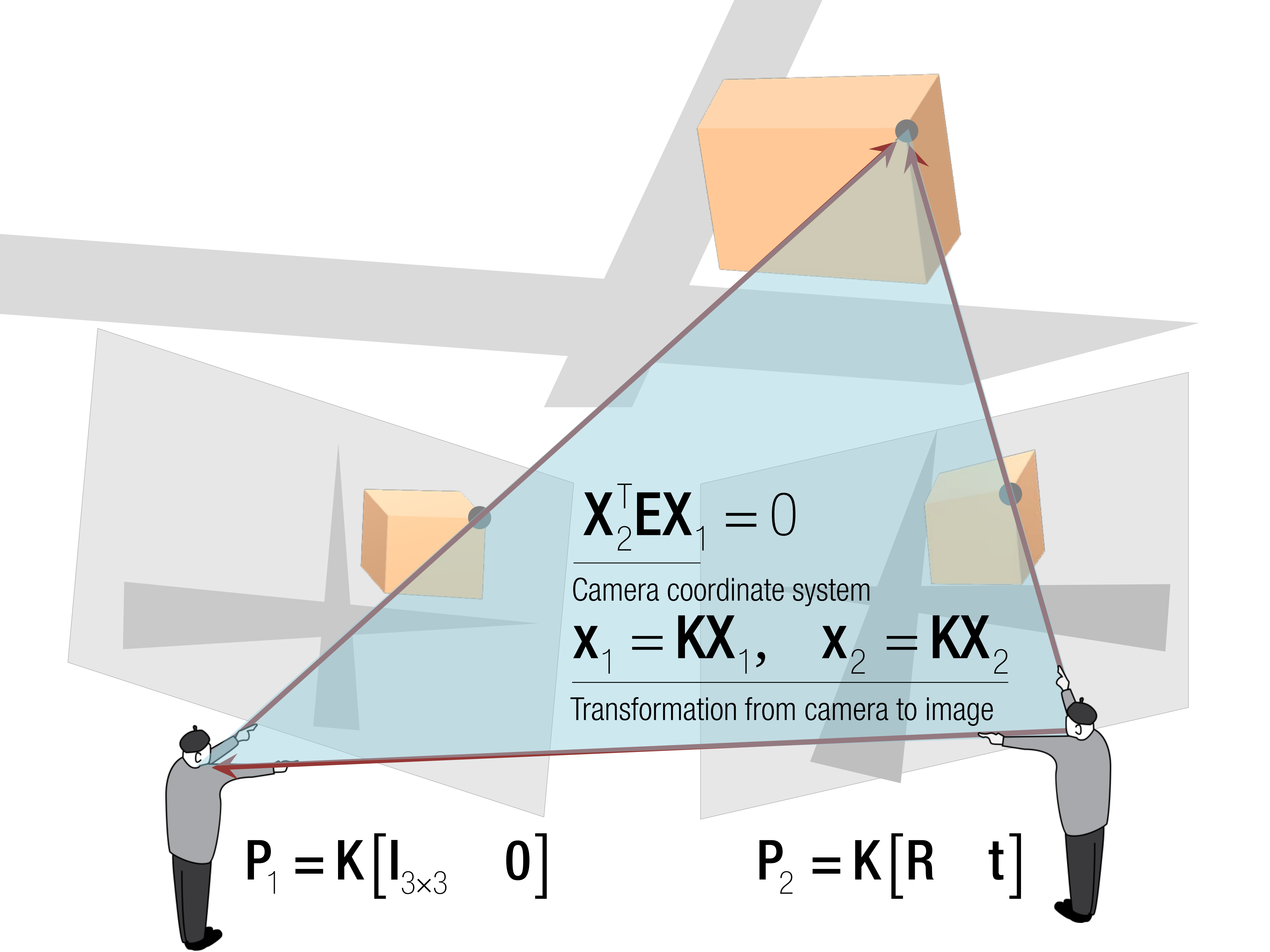


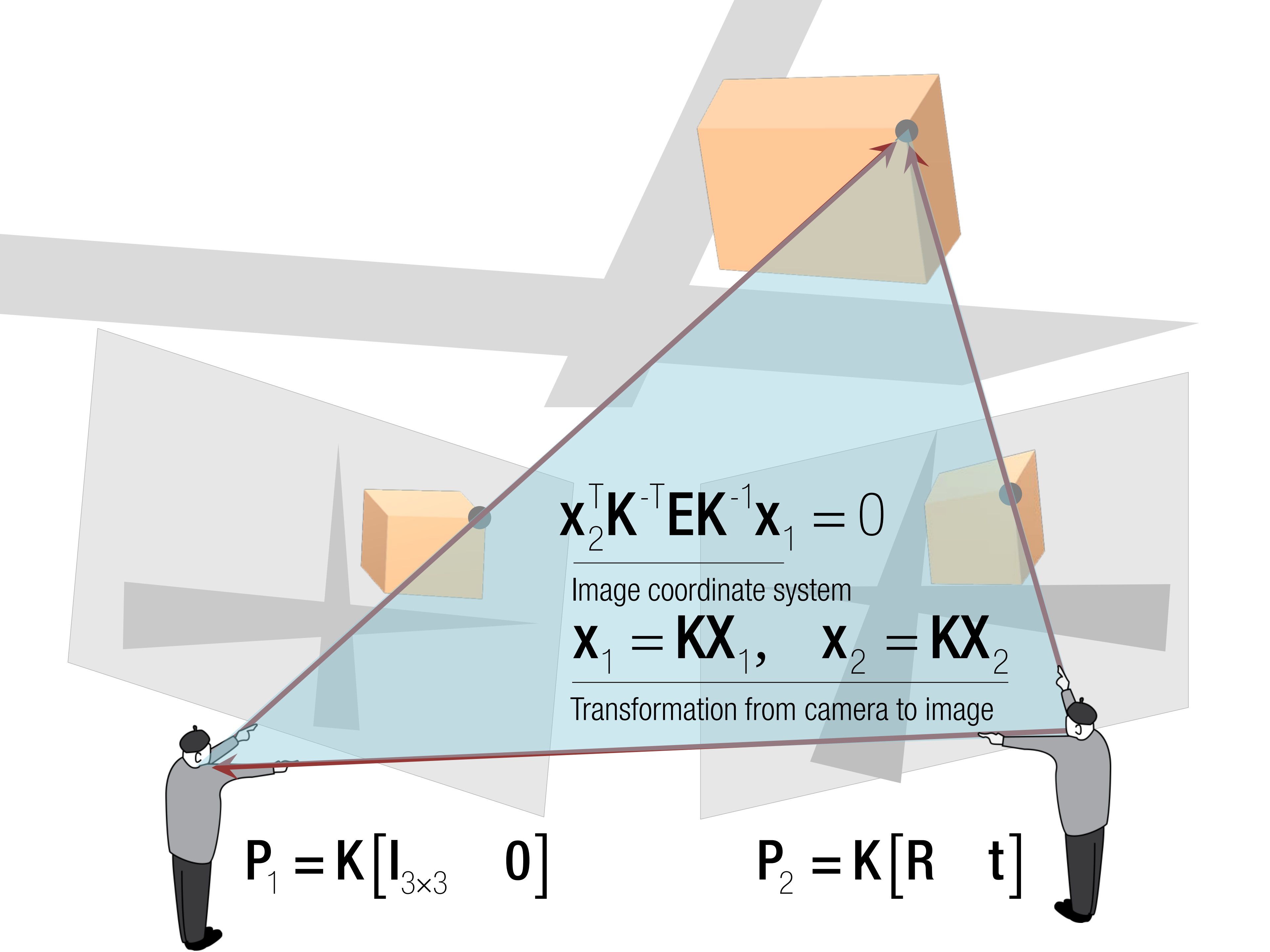


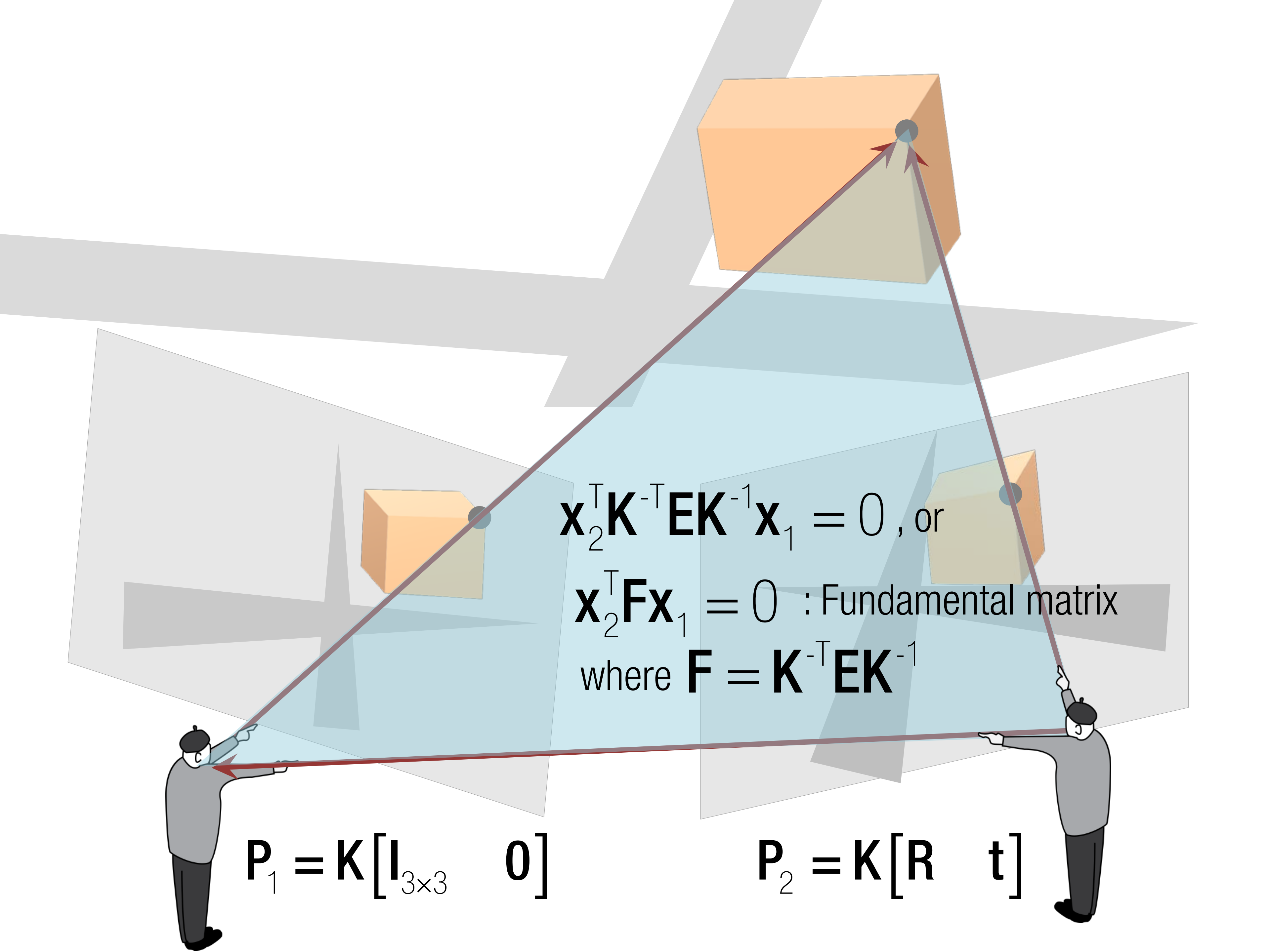
Fundamental Matrix

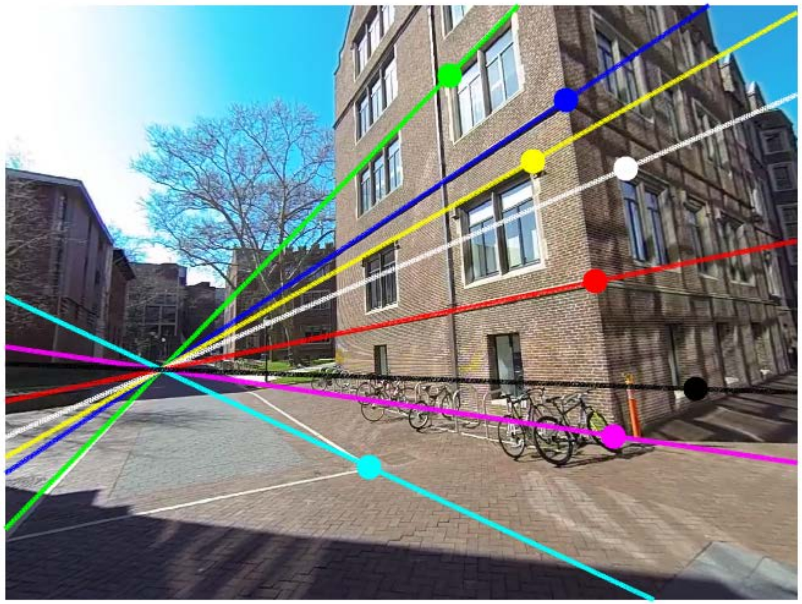


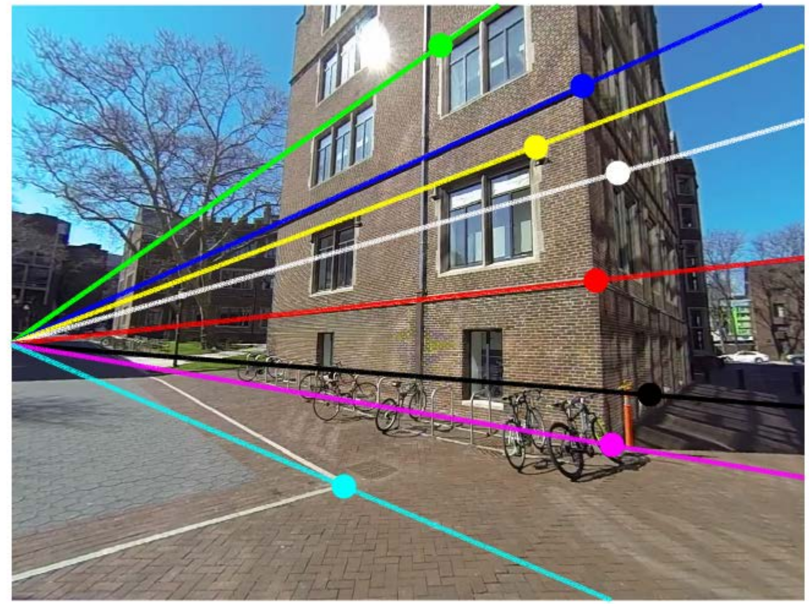

$X_2^T E X_1 = 0$
Camera coordinate system
$P_1 = K[I_{3\times3} \quad 0]$
$P_2 = K[R \quad t]$


$\mathbf{X}_2^{\mathsf{T}}\mathbf{E}\mathbf{X}_1 = 0$
Camera coordinate system
$\mathbf{x}_1 = \mathbf{K}\mathbf{X}_1, \quad \mathbf{x}_2 = \mathbf{K}\mathbf{X}_2$
Transformation from camera to image
$\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}\end{bmatrix}$
$\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

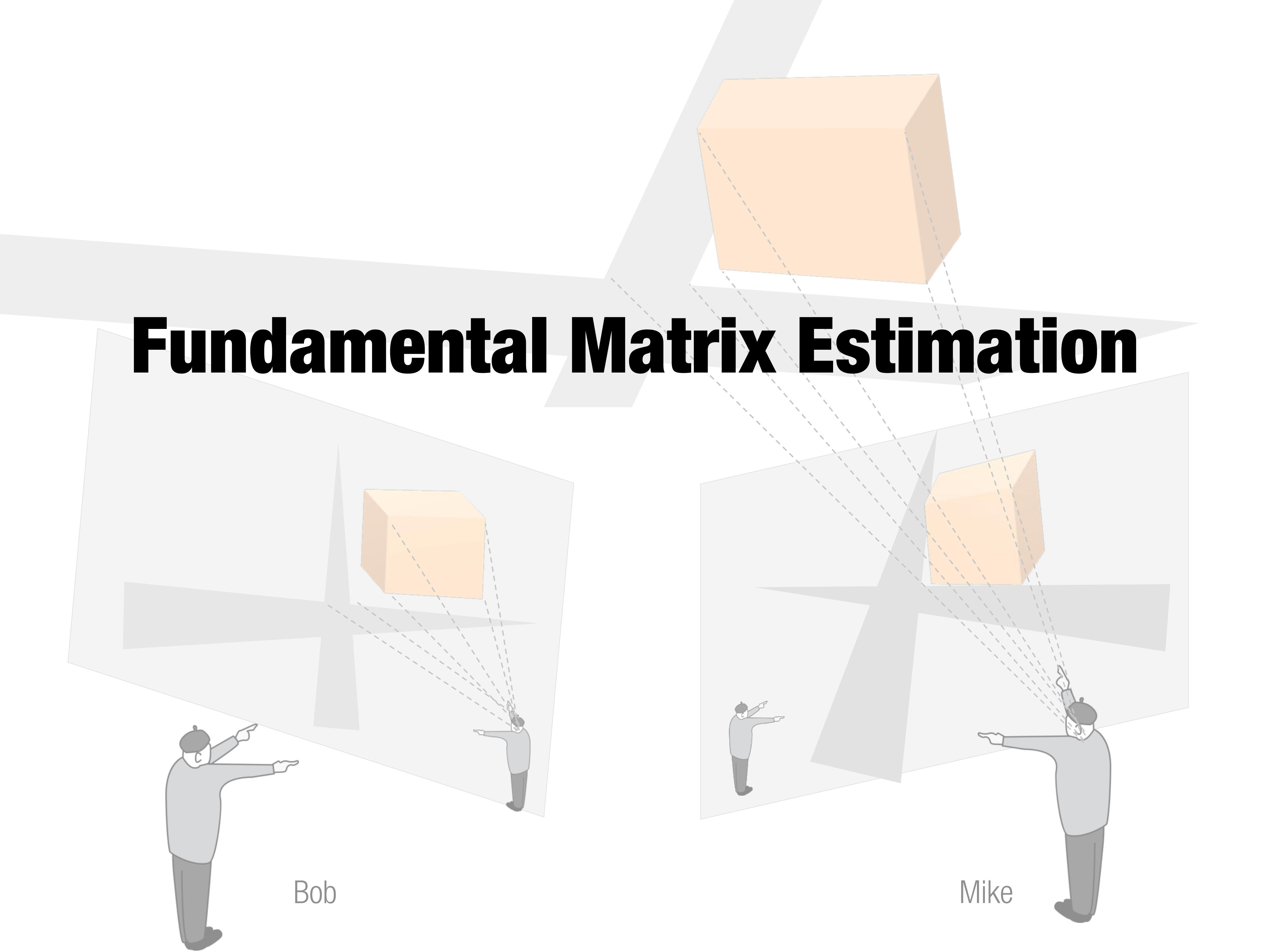

$x_2^T K^{-T} E K^{-1} x_1 = 0$
Image coordinate system
$x_1 = KX_1, \quad x_2 = KX_2$
Transformation from camera to image
$P_1 = K[I_{3\times3} \quad 0]$
$P_2 = K[R \quad t]$


$\mathbf{x}_2^{\mathsf{T}}\mathbf{K}^{-\mathsf{T}}\mathbf{E}\mathbf{K}^{-1}\mathbf{x}_1 = 0$, or
$\mathbf{x}_2^{\mathsf{T}}\mathbf{F}\mathbf{x}_1 = 0$: Fundamental matrix
where $\mathbf{F} = \mathbf{K}^{-\mathsf{T}}\mathbf{E}\mathbf{K}^{-1}$
$\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times 3} & \mathbf{0}\end{bmatrix}$
$\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$





Fundamental Matrix Estimation



Fundamental Matrix

$$\mathbf{x}_2^{\top}\mathbf{F}\mathbf{x}_1 = 0$$

$$\mathbf{F} \in \mathbb{R}^{3\times 3}$$

$$\mathrm{rank}(\mathbf{F}) = 2$$

Matrix dimensions

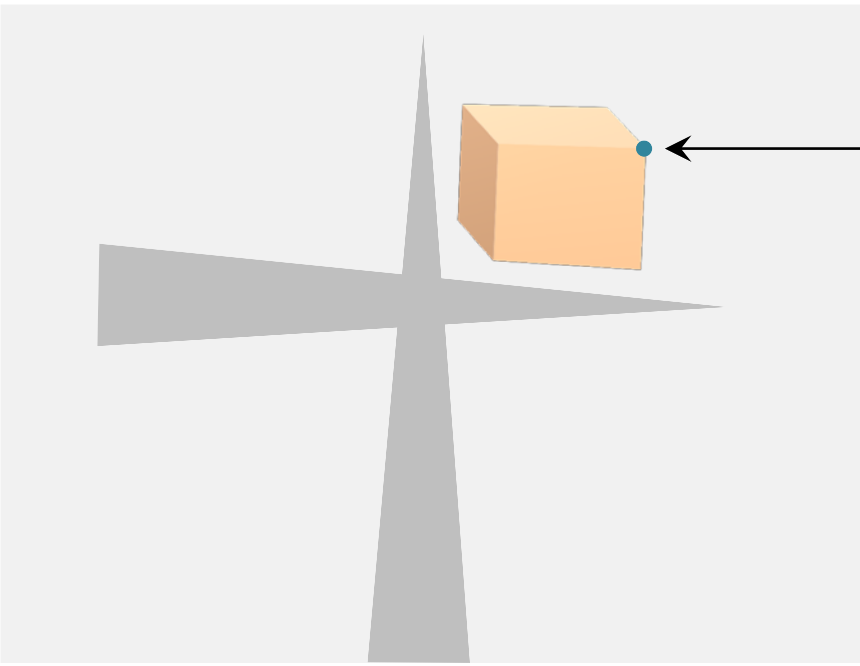
Degree of freedoms: $\underbrace{\overbrace{3\text{x}3}^{\text{Matrix dimensions}}-\underbrace{1}_{\text{scale factor}}}{} = 8$

\# of unknowns: 8

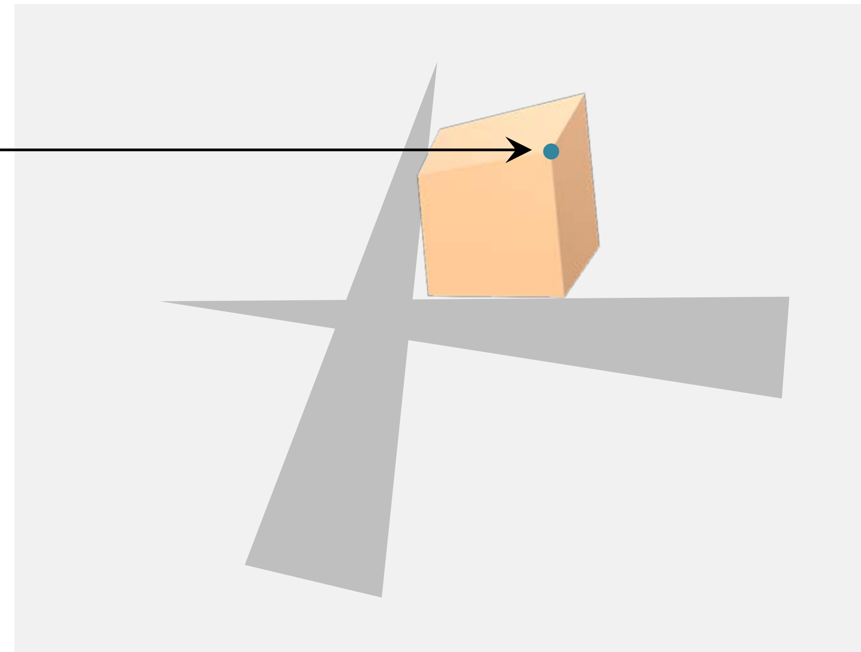
\# of required equations: 8

$$\left.\begin{array}{c} \mathbf{x}_{2,1}^{\top}\mathbf{F}\mathbf{x}_{1,1} = 0 \\ \vdots \\ \mathbf{x}_{2,8}^{\top}\mathbf{F}\mathbf{x}_{1,8} = 0 \end{array}\right|\ \text{8 correspondences}$$

Point correspondence

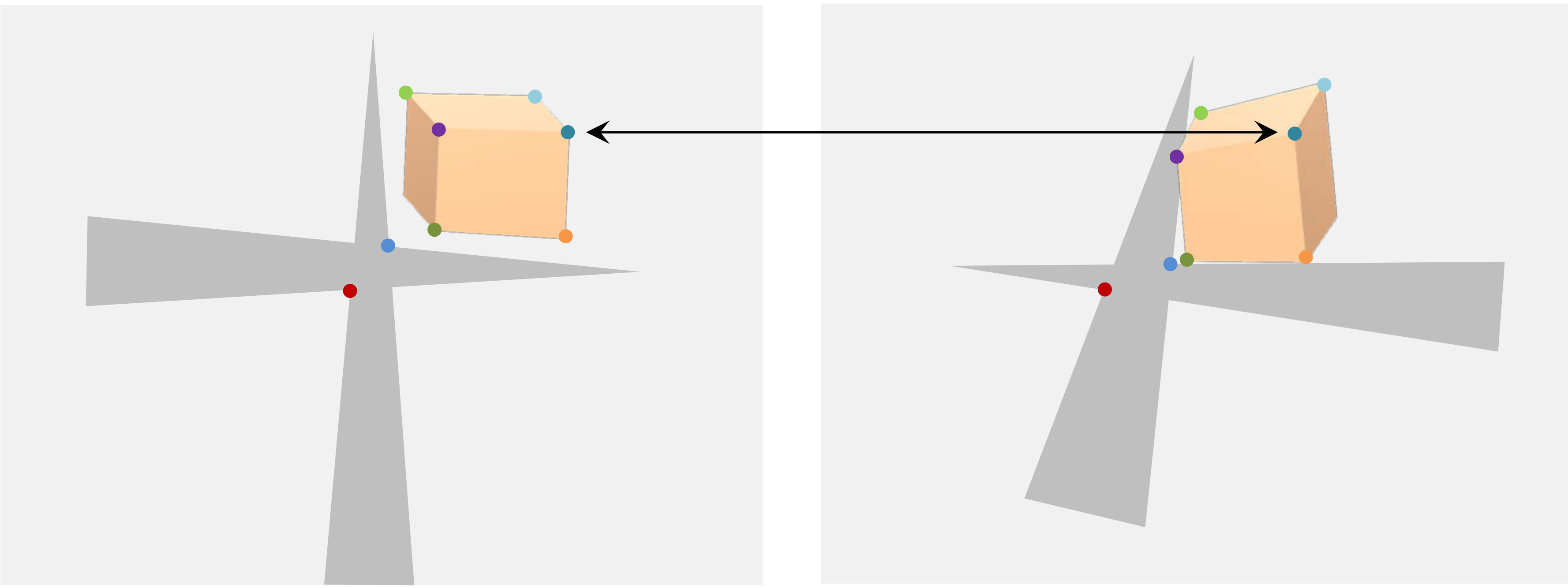


Bob's view



Mike's view

Point correspondence



Bob's view Mike's view

8 correspondences

$$\mathbf{x}_2^{\mathsf{T}}\mathbf{F}\mathbf{x}_1 = 0$$

$$\begin{bmatrix} u_i^2 & v_i^2 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_i^1 \\ v_i^1 \\ 1 \end{bmatrix} = 0$$

Linear equation in F:

$$u_i^1 u_i^2 f_{11} + u_i^1 v_i^2 f_{21} + u_i^1 f_{31} + v_i^1 u_i^2 f_{12} + v_i^1 v_i^2 f_{22} + v_i^1 f_{32} + u_i^2 f_{13} + v_i^2 f_{23} + f_{33} = 0$$

$$\mathbf{x}_2^{\top}\mathbf{F}\mathbf{x}_1 = 0$$

$$\begin{bmatrix} u_i^2 & v_i^2 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_i^1 \\ v_i^1 \\ 1 \end{bmatrix} = 0$$

Linear equation in F:

$$\begin{bmatrix} u_1^1 u_1^2 & u_1^1 v_1^2 & u_1^1 & v_1^1 u_1^2 & v_1^1 v_1^2 & v_1^1 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_8^1 u_8^2 & u_8^1 v_8^2 & u_8^1 & v_8^1 u_8^2 & v_8^1 v_8^2 & v_8^1 & u_8^2 & v_8^2 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

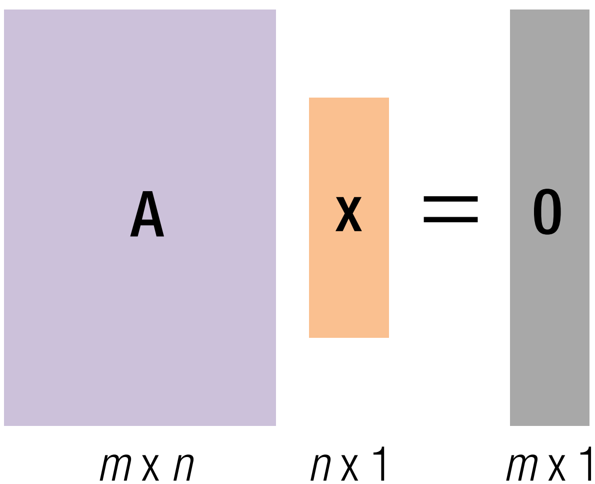
$$\mathbf{x}_2^{\top}\mathbf{F}\mathbf{x}_1 = 0$$

$$\begin{bmatrix} u_i^2 & v_i^2 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_i^1 \\ v_i^1 \\ 1 \end{bmatrix} = 0$$

Linear equation in F:

$$\underbrace{\begin{bmatrix} u_1^1u_1^2 & u_1^1v_1^2 & u_1^1 & v_1^1u_1^2 & v_1^1v_1^2 & v_1^1 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_8^1u_8^2 & u_8^1v_8^2 & u_8^1 & v_8^1u_8^2 & v_8^1v_8^2 & v_8^1 & u_8^2 & v_8^2 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}}_{\mathbf{X}} = \mathbf{0}$$

Linear Homogeneous Equations



Linear least square solve produces a trivial solution:

$$\mathbf{x} = \left(\mathbf{A}^{\mathsf{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{b} \rightarrow \mathbf{x} = \mathbf{0}$$

An additional constraint on $\mathbf{X}$ to avoid the trivial solution: $\left\|\mathbf{x}\right\| = 1$

1) $\textbf{rank(A)} = r < n-1$: infinite number of solutions

$$\mathbf{x} = \lambda_{r+1}\mathbf{V}_{r+1} + \cdots + \lambda_n \mathbf{V}_n \quad \text{where} \quad \sum_{i=r+1}^{n} \lambda_i^2 = 1$$

2) $\textbf{rank(A)} = n-1$: one exact solution

$$\mathbf{x} = \mathbf{V}_n$$

3) $n < m$: no exact solution in general (needs least squares)

$$\boxed{\min_{\mathbf{x}} \left\|\mathbf{A}\mathbf{x}\right\|^2 \text{ subject to } \left\|\mathbf{x}\right\| = 1 \rightarrow \mathbf{x} = \mathbf{V}_n}$$

8 Point Algorithm

- Construct 8x9 matrix **A**.

$$\begin{bmatrix} u_1^1 u_1^2 & u_1^1 v_1^2 & u_1^1 & v_1^1 u_1^2 & v_1^1 v_1^2 & v_1^1 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_8^1 u_8^2 & u_8^1 v_8^2 & u_8^1 & v_8^1 u_8^2 & v_8^1 v_8^2 & v_8^1 & u_8^2 & v_8^2 & 1 \end{bmatrix}$$

A

8 Point Algorithm

- Construct 8x9 matrix $\mathbf{A}$.
- Solving linear homogeneous equations via SVD:

  $\mathbf{x} = \mathbf{V}_{:,9}$ where $\mathbf{A} = \mathbf{UDV}^{\top}$

  $\mathbf{F} = \mathrm{reshape}(\mathbf{x}, 3, 3)$: constructing matrix from vector.

8 Point Algorithm

- Construct 8x9 matrix **A**.
- Solving linear homogeneous equations via SVD:

  $\mathbf{x} = \mathbf{V}_{:,8}$ where $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$

  $\mathbf{F} = \mathrm{reshape}(\mathbf{x}, 3, 3)$: constructing matrix from vector.
- Applying rank constraint, i.e., $\mathrm{rank}(\mathbf{F}) = 2$.

$\mathbf{F}_{\mathrm{rank2}} = \mathbf{U}\tilde{\mathbf{D}}\mathbf{V}^{\mathrm{T}}$ where $\tilde{\mathbf{D}}$ $:\mathbf{D}$ with the last element zero.

$\mathbf{F}_{\mathrm{rank2}} = \mathbf{U} \; \tilde{\mathbf{D}} \; \mathbf{V}^{\mathrm{T}}$ $\qquad$ $\mathbf{F} = \mathbf{U} \; \mathbf{D} \; \mathbf{V}^{\mathrm{T}}$

SVD cleanup

1.2 Match Outlier Rejection via RANSAC

Goal Given N correspondences between two images ($N \geq 8$), $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$, implement the following function that estimates inlier correspondences using fundamental matrix based RANSAC:

```
[y1 y2 idx] = GetInliersRANSAC(x1, x2)
```

(INPUT) `x1` and `x2`: $N \times 2$ matrices whose row represents a correspondence.

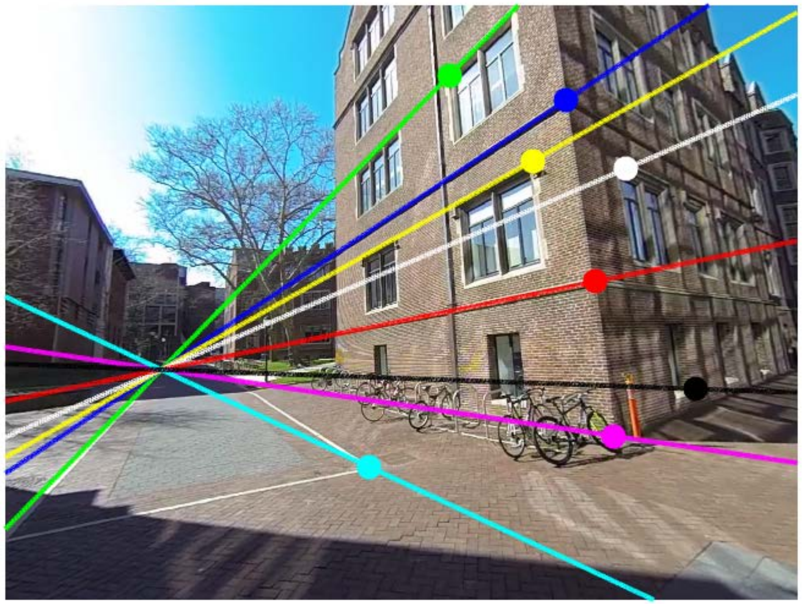
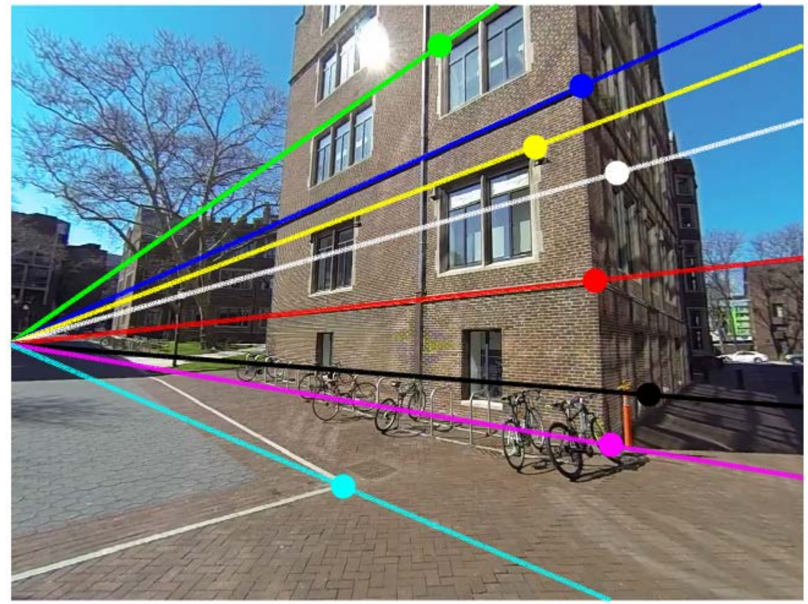
(OUTPUT) `y1` and `y2`: $N_i \times 2$ matrices whose row represents an inlier correspondence where N_i is the number of inliers.

(OUTPUT) `idx`: $N \times 1$ vector that indicates ID of inlier `y1`.

A pseudo code the RANSAC is shown in Algorithm 2.

Algorithm 2 GetInliersRANSAC

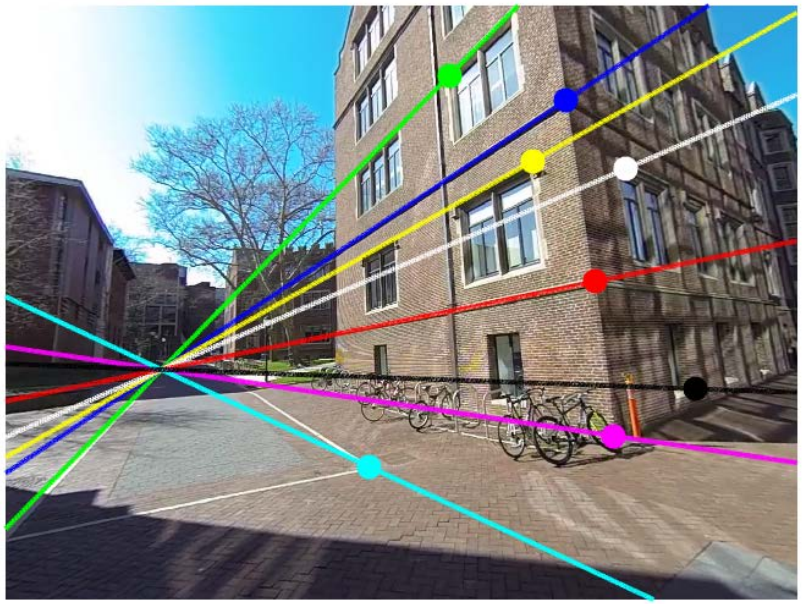
$n \leftarrow 0$
for $i = 1 : M$ **do**
    Choose 8 correspondences, $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$, randomly
    `F = EstimateFundamentalMatrix(`$\hat{\mathbf{x}}_1$`, `$\hat{\mathbf{x}}_2$`)`
    $\mathcal{S} \leftarrow \emptyset$
    for $j = 1 : N$ **do**
        if $|\mathbf{x}_{2j}^{\mathsf{T}} \mathbf{F} \mathbf{x}_{1j}| < \epsilon$ **then**
            $\mathcal{S} \leftarrow \mathcal{S} \cup \{j\}$
        end if
    end for
    if $n < |\mathcal{S}|$ **then**
        $n \leftarrow |\mathcal{S}|$
        $\mathcal{S}_{in} \leftarrow \mathcal{S}$
    end if
end for

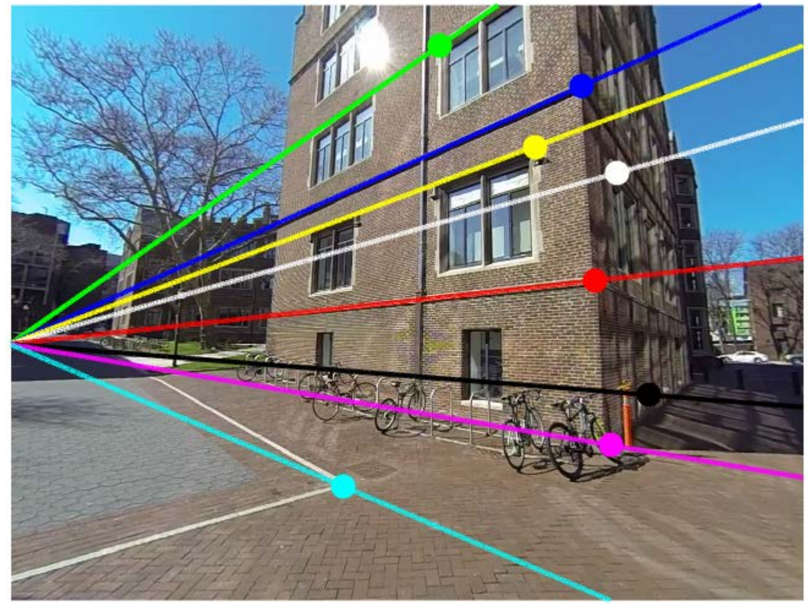



F =

1.0e+003 *

0.0000	0.0001	-0.0463
-0.0001	0.0000	0.0181
0.0519	-0.0043	-9.9997





```
>> rank(F)
ans =
    3
>> [u,d,v] = svd(F);
>> d(3,3) = 0;
```

```
>> F = u * d * v'          : SVD cleanup
F =
  1.0e+003 *
    0.0000    0.0001   -0.0463
   -0.0001    0.0000    0.0181
    0.0519   -0.0043   -9.9997
>> rank(F)
ans =
    2
```

x1 =

950 450

L2 =

-0.1024 -0.9947 547.0942

$$\mathbf{L}_2 = \mathbf{F}\mathbf{x}_1$$

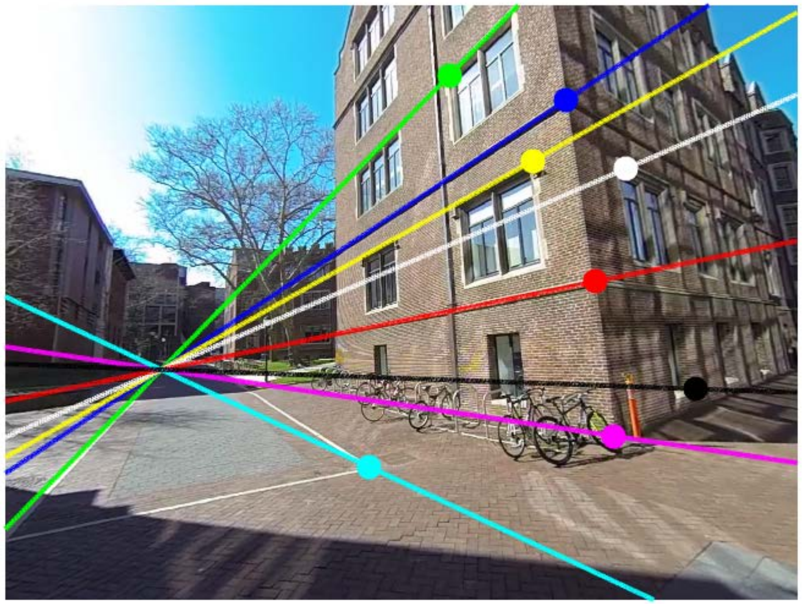
L1 =

0.5489 0.8359 -627.0515

x2 =

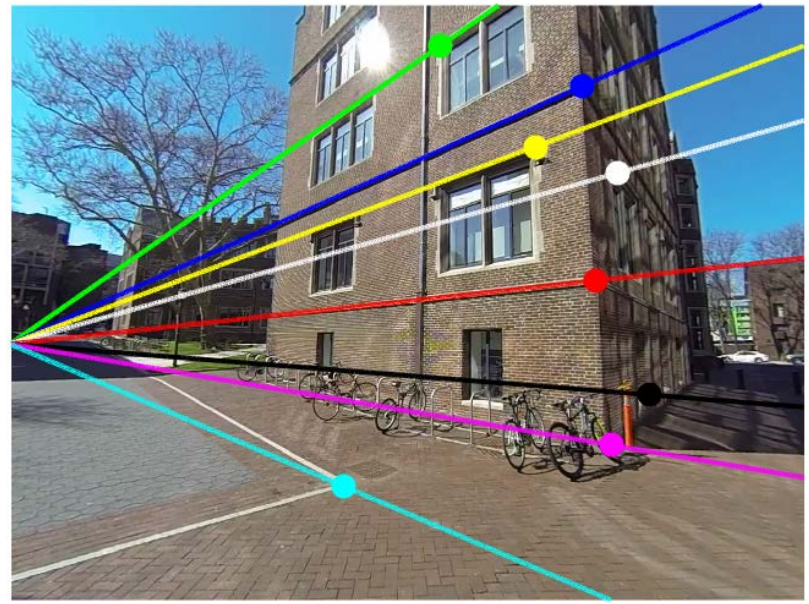
920 130

$$\mathbf{L}_1 = \mathbf{F}^\top \mathbf{x}_2$$



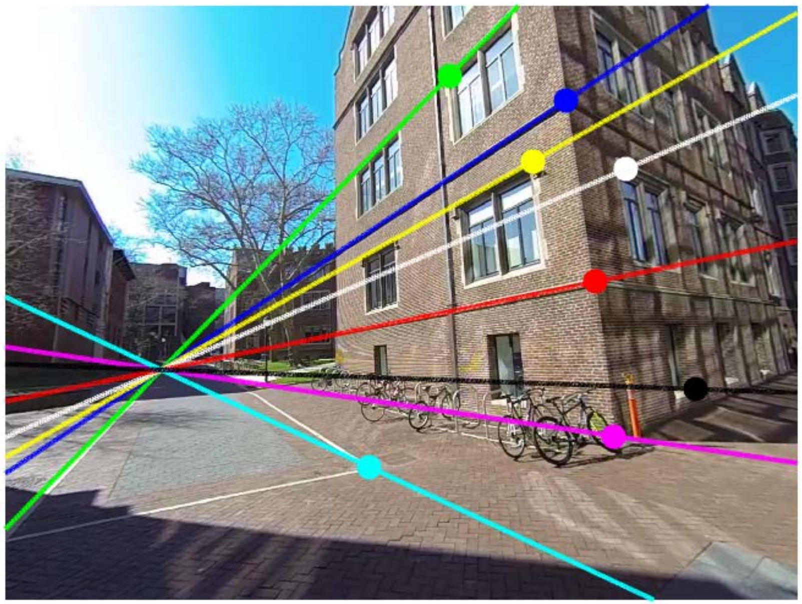
```
[u,d] = eigs(F'*F);
u =
   -0.0052    0.9258   -0.3780
    0.0004   -0.3780   -0.9258
    1.0000    0.0050   -0.0016
d =
   1.0000         0              0
        0    6.4719e-10          0
        0         0    -7.6511e-22
```

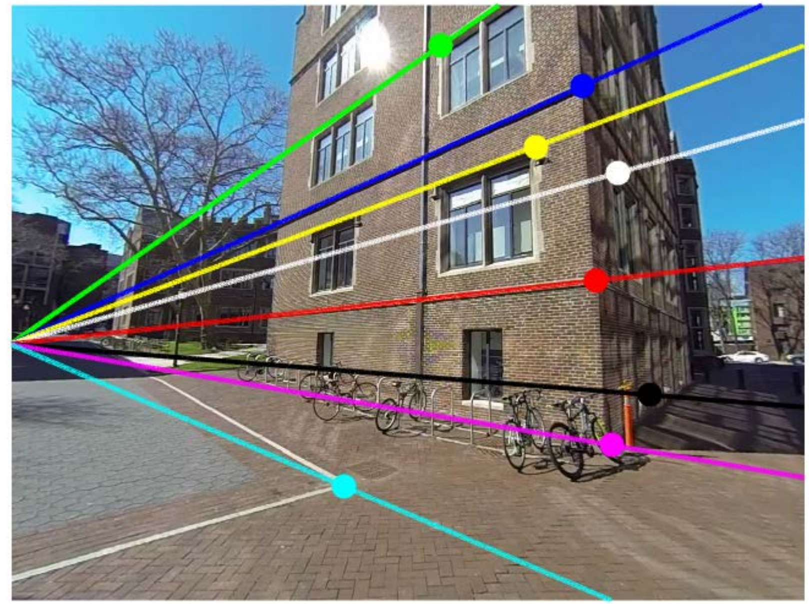
uu = u(:, 3) = [-0.3780, -0.9258, -0.0016]



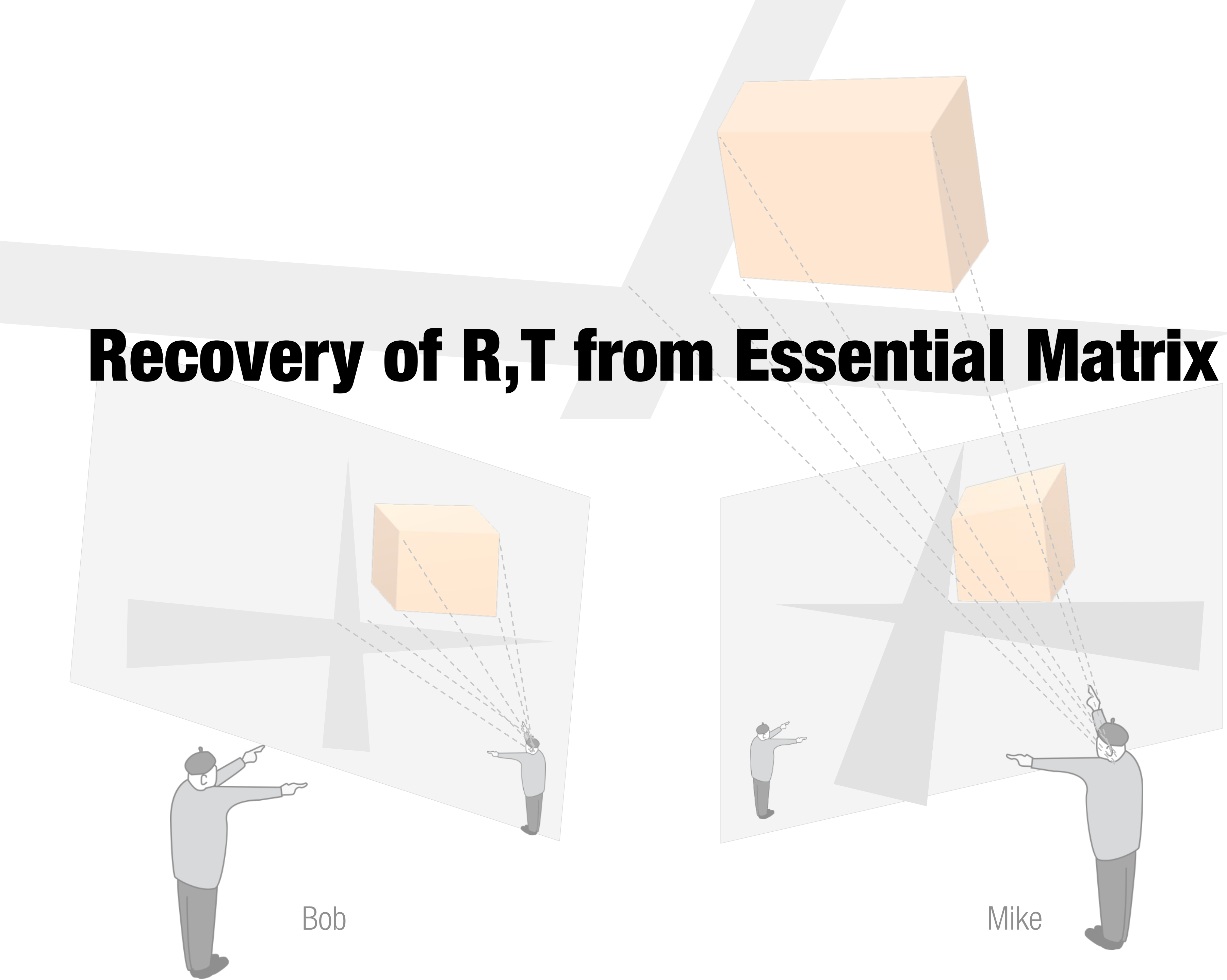
```
[u,d] = eigs(F*F');
u =
    0.0046    1.0000    0.0029
   -0.0018    0.0029   -1.0000
    1.0000   -0.0046   -0.0018
d =
   1.0000          0              0
        0    6.4719e-10          0
        0          0    -5.6583e-21
```

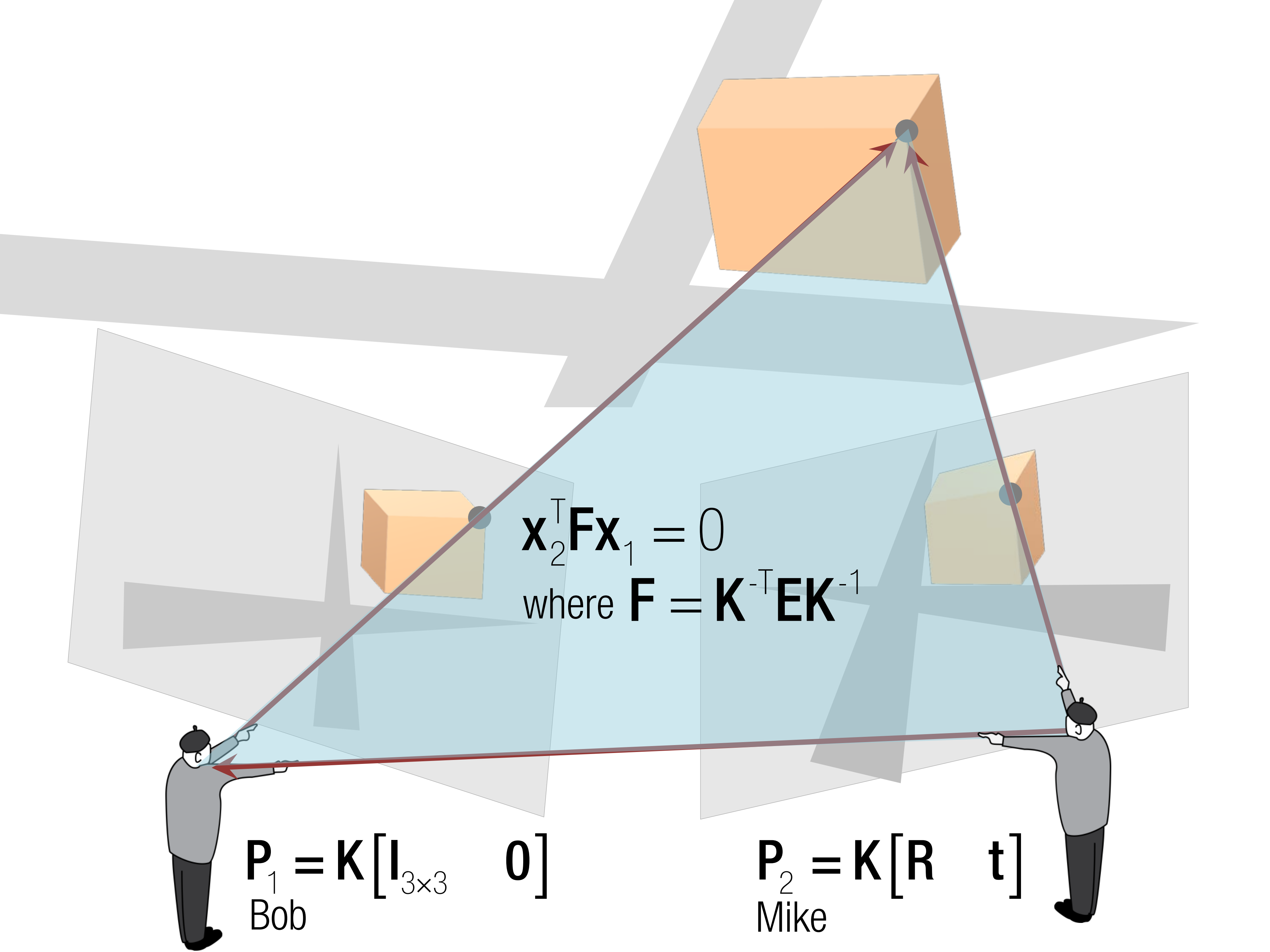
uu = u(:, 3) = [0.0029, -1.0000, -0.0018]

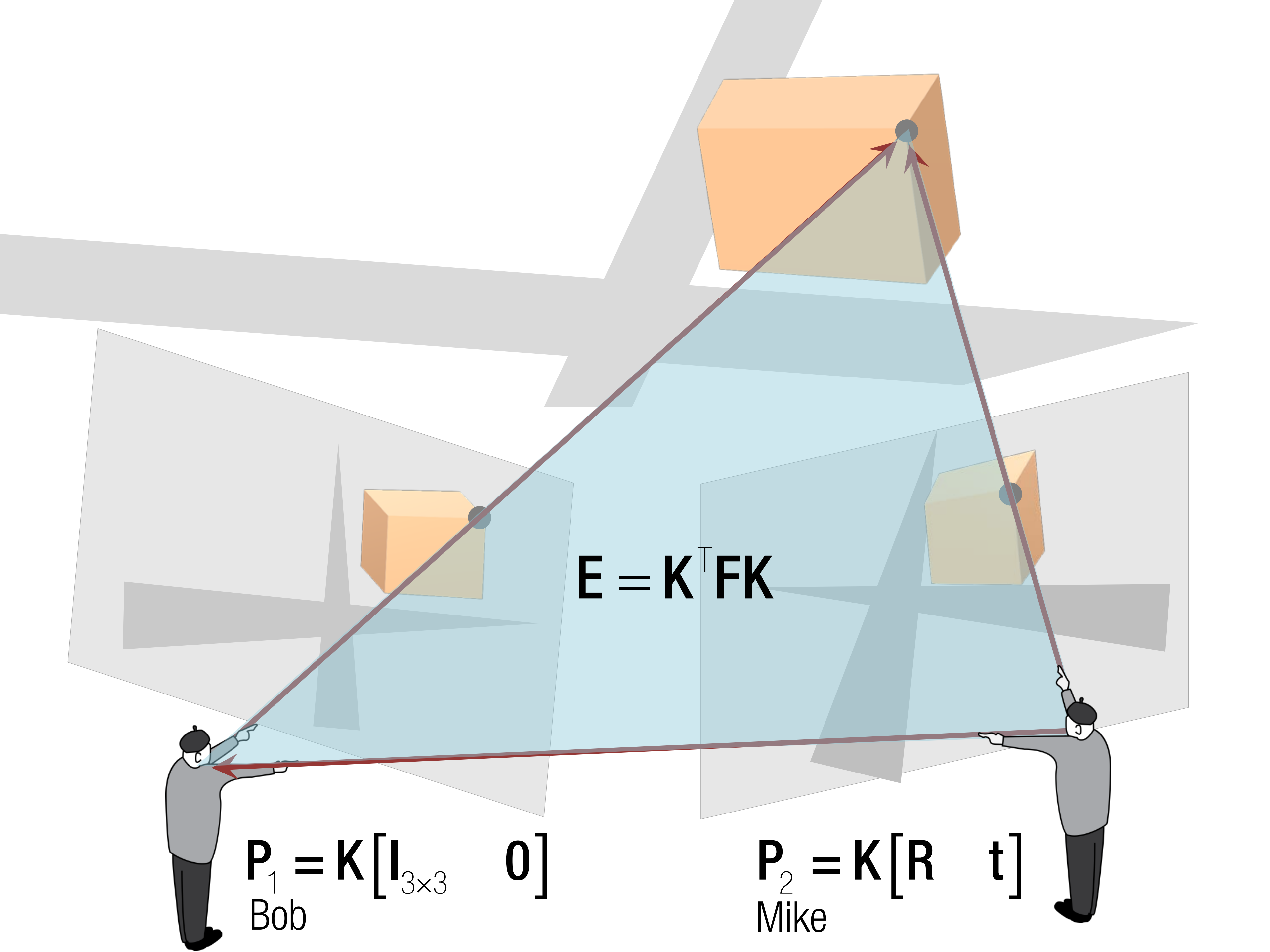




Recovery of R,T from Essential Matrix

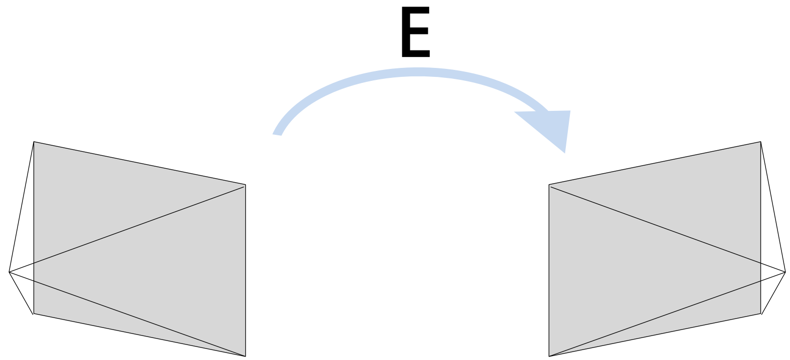



$\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0$
where $\mathbf{F} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1}$
$\mathbf{P}_1 = \mathbf{K}[\mathbf{I}_{3\times3} \quad \mathbf{0}]$
Bob
$\mathbf{P}_2 = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$
Mike


$\mathbf{E} = \mathbf{K}^{\top}\mathbf{F}\mathbf{K}$
$\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times 3} & \mathbf{0}\end{bmatrix}$
Bob
$\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$
Mike

$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$ How to decompose the essential matrix to rotation and translation?

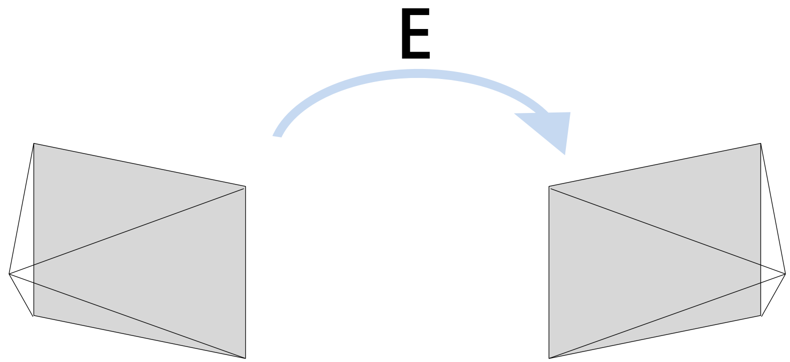
Essential matrix (rank 2)



$$\mathbf{P}_1 = \left[\mathbf{I}_{3\times3} \mid \mathbf{0}_3 \right] \qquad \mathbf{P}_2 = [\mathbf{R} \mid \mathbf{t}]$$

$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$ How to decompose the essential matrix to rotation and translation?

Essential matrix (rank 2)

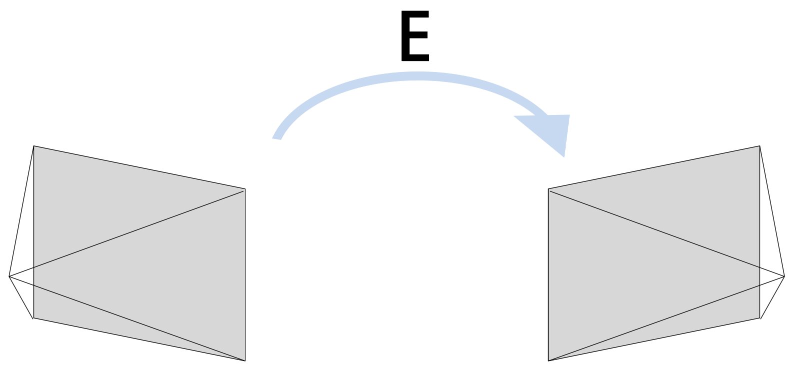


$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I}_{3\times 3} & | & \mathbf{0}_3 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix}$$

$\mathbf{t}$: Epipole in image 2 because $\mathbf{P}_2 \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \mathbf{t}$

$$\underline{\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}}$$ How to decompose the essential matrix to rotation and translation?

Essential matrix (rank 2)

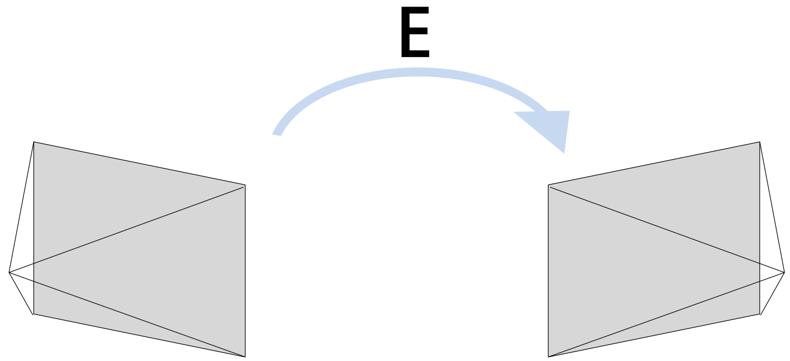


$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I}_{3\times3} & | & \mathbf{0}_3 \end{bmatrix} \qquad \mathbf{P}_2 = [\mathbf{R} \mid \mathbf{t}]$$

$\mathbf{t}$: Epipole in image 2 because $\mathbf{P}_2 \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = [\mathbf{R} \mid \mathbf{t}] \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \mathbf{t}$

$\mathbf{t}^\top \mathbf{E} = \mathbf{0}$: Left nullspace of the essential matrix is the epipole in image 2.

$$\underbrace{\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}}_{\text{Essential matrix (rank 2)}}$$ How to decompose the essential matrix to rotation and translation?



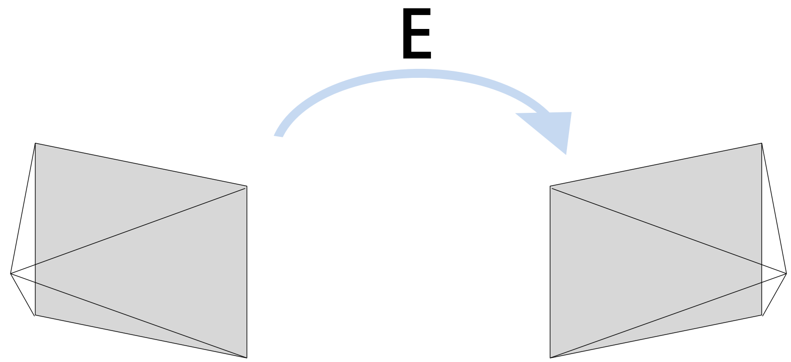
$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I}_{3\times3} & | & \mathbf{0}_3 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix}$$

$\mathbf{t}$: Epipole in image 2 because $\mathbf{P}_2 \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \mathbf{t}$

$\mathbf{t}^\mathsf{T}\mathbf{E} = \mathbf{0}$: Left nullspace of the essential matrix is the epipole in image 2.

$\longrightarrow$ $\mathbf{t} = \mathbf{u}_3$, or $-\mathbf{u}_3$ where $\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}$ and $\underbrace{\mathbf{E} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^\mathsf{T}}_{\text{Singular value decomposition (SVD)}}$

$\underbrace{\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R}}_{\text{Essential matrix}}$ How to decompose the essential matrix to rotation and translation?

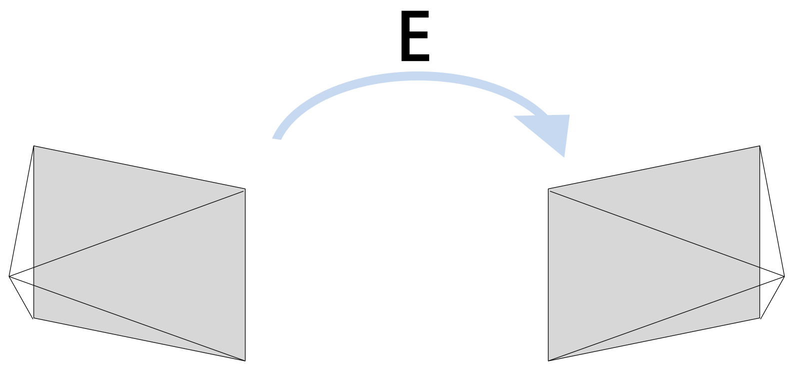


$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I}_{3\times 3} & | & \mathbf{0}_3 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix}$$

$$\mathbf{E} = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{V}^{\mathsf{T}}$$

$$\underbrace{\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}}_{\text{Essential matrix}}$$

How to decompose the essential matrix to rotation and translation?

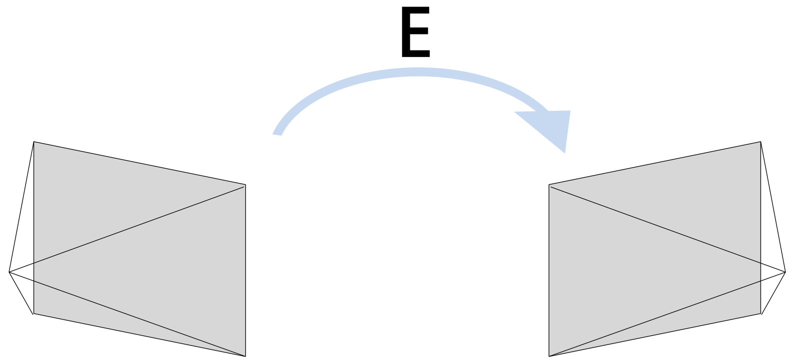


$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I}_{3\times 3} & | & \mathbf{0}_3 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix}$$

$$\mathbf{E} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^\mathsf{T} = [\mathbf{t}]_\times \mathbf{R}$$

$$\underline{\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}}$$
Essential matrix

How to decompose the essential matrix to rotation and translation?



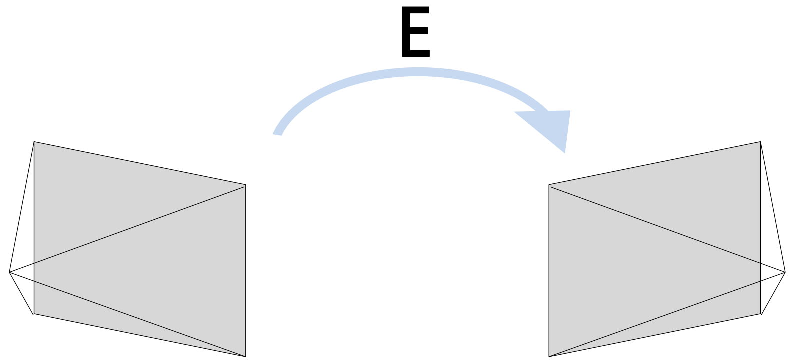
$$\mathbf{P}_1 = \left[\mathbf{I}_{3\times3} \mid \mathbf{0}_3\right] \qquad \mathbf{P}_2 = [\mathbf{R} \mid \mathbf{t}]$$

$$\mathbf{E} = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{V}^\mathsf{T} = [\mathbf{t}]_\times \mathbf{R} = \left(\mathbf{U}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{U}^\mathsf{T}\right)\left(\mathbf{U}\mathbf{Y}\mathbf{V}^\mathsf{T}\right)$$

where $\mathbf{U} = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{t}\end{bmatrix}$

$$\underbrace{\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}}_{\text{Essential matrix}}$$

How to decompose the essential matrix to rotation and translation?



$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I}_{3\times 3} & | & \mathbf{0}_3 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix}$$

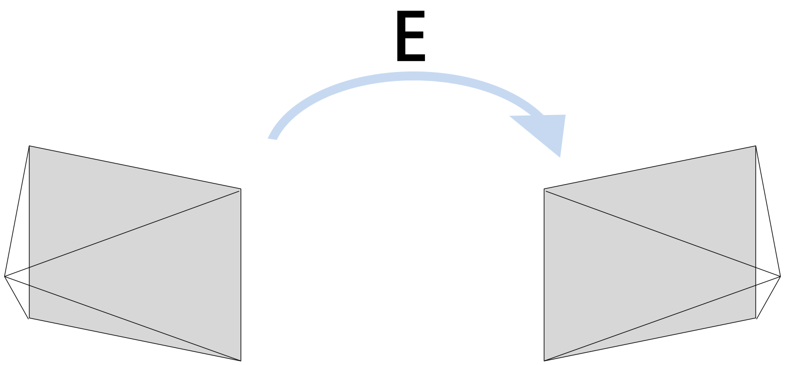
$$\mathbf{E} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^\mathsf{T} = [\mathbf{t}]_\times \mathbf{R} = \left(\mathbf{U} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{U}^\mathsf{T} \right) \left(\mathbf{U}\mathbf{Y}\mathbf{V}^\mathsf{T} \right)$$

How do we set $\mathbf{Y}$?

where $\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{t} \end{bmatrix}$

$$\underset{\text{Essential matrix}}{\underline{\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}}}$$

How to decompose the essential matrix to rotation and translation?



$$\mathbf{P}_1 = \left[\mathbf{I}_{3\times3} \mid \mathbf{0}_3\right] \qquad \mathbf{P}_2 = [\mathbf{R} \mid \mathbf{t}]$$

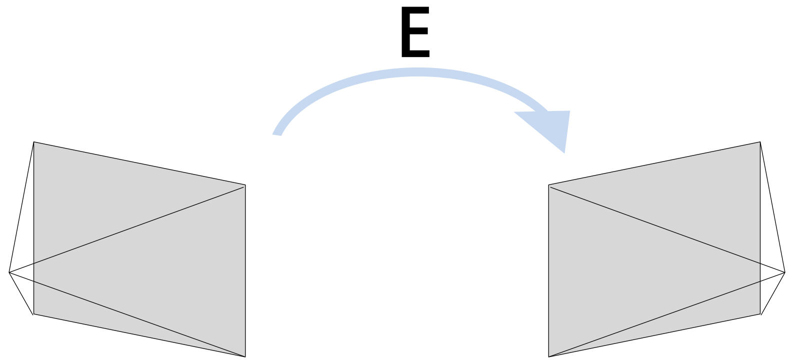
$$\mathbf{E} = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{V}^\mathsf{T} = [\mathbf{t}]_\times \mathbf{R} = \mathbf{U}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{Y}\mathbf{V}^\mathsf{T}$$

How do we set $\mathbf{Y}$?

$$\therefore \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{Y}$$

$$\underbrace{\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}}_{\text{Essential matrix}}$$

How to decompose the essential matrix to rotation and translation?



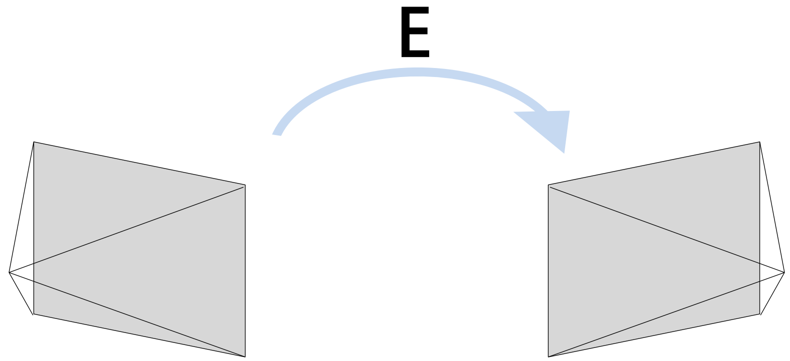
$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I}_{3\times3} & | & \mathbf{0}_3 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix}$$

$$\mathbf{E} = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{V}^\mathsf{T} = [\mathbf{t}]_\times \mathbf{R} = \mathbf{U}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{Y}\mathbf{V}^\mathsf{T}$$

How do we set $\mathbf{Y}$?

$$\therefore \mathbf{Y} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ or } \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^\mathsf{T}$$

$$\underbrace{\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}}_{\text{Essential matrix}}$$ How to decompose the essential matrix to rotation and translation?



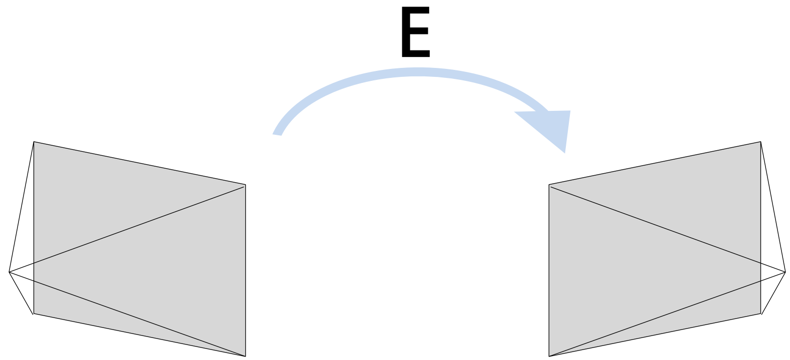
$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I}_{3\times3} & | & \mathbf{0}_3 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix}$$

$$\mathbf{t} = \mathbf{u}_3 \quad \text{, or } -\mathbf{u}_3 \qquad \mathbf{R} = \mathbf{U}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{V}^{\mathsf{T}}, \text{ or } \mathbf{U}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}}\mathbf{V}^{\mathsf{T}}$$

where $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}}$ $\qquad \mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{t} \end{bmatrix}$

$$\underbrace{\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}}_{\text{Essential matrix}}$$

How to decompose the essential matrix to rotation and translation?



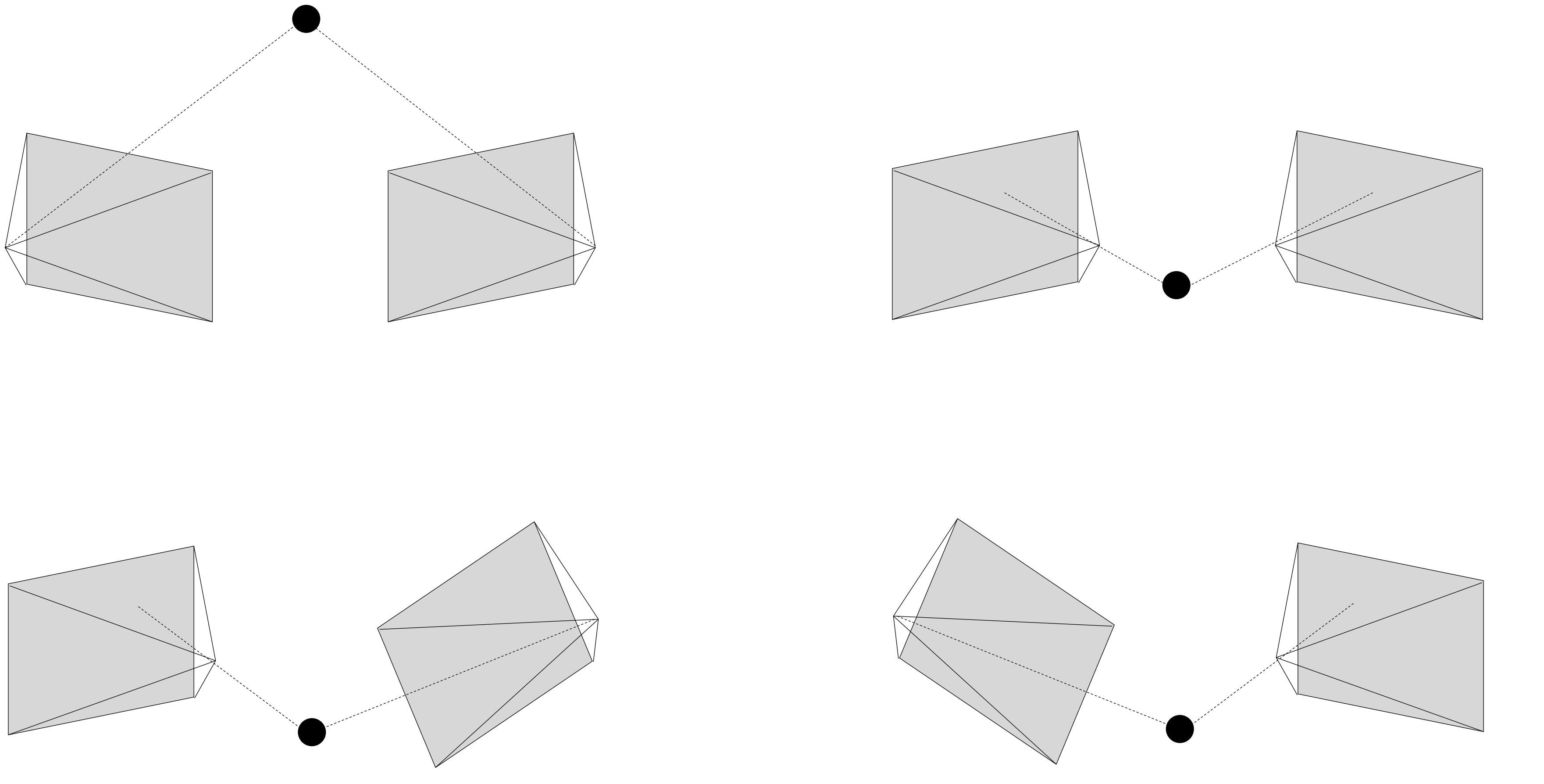
$$\mathbf{P}_1 = \left[\mathbf{I}_{3\times 3} \mid \mathbf{0}_3 \right] \qquad \mathbf{P}_2 = [\mathbf{R} \mid \mathbf{t}]$$

Four configurations:

$$\mathbf{P}_2 = \left[\mathbf{U}\mathbf{Y}\mathbf{V}^\top \mid \mathbf{u}_3 \right], \text{ or } \left[\mathbf{U}\mathbf{Y}^\top\mathbf{V}^\top \mid \mathbf{u}_3 \right], \text{ or } \left[\mathbf{U}\mathbf{Y}\mathbf{V}^\top \mid -\mathbf{u}_3 \right], \text{ or } \left[\mathbf{U}\mathbf{Y}^\top\mathbf{V}^\top \mid -\mathbf{u}_3 \right]$$

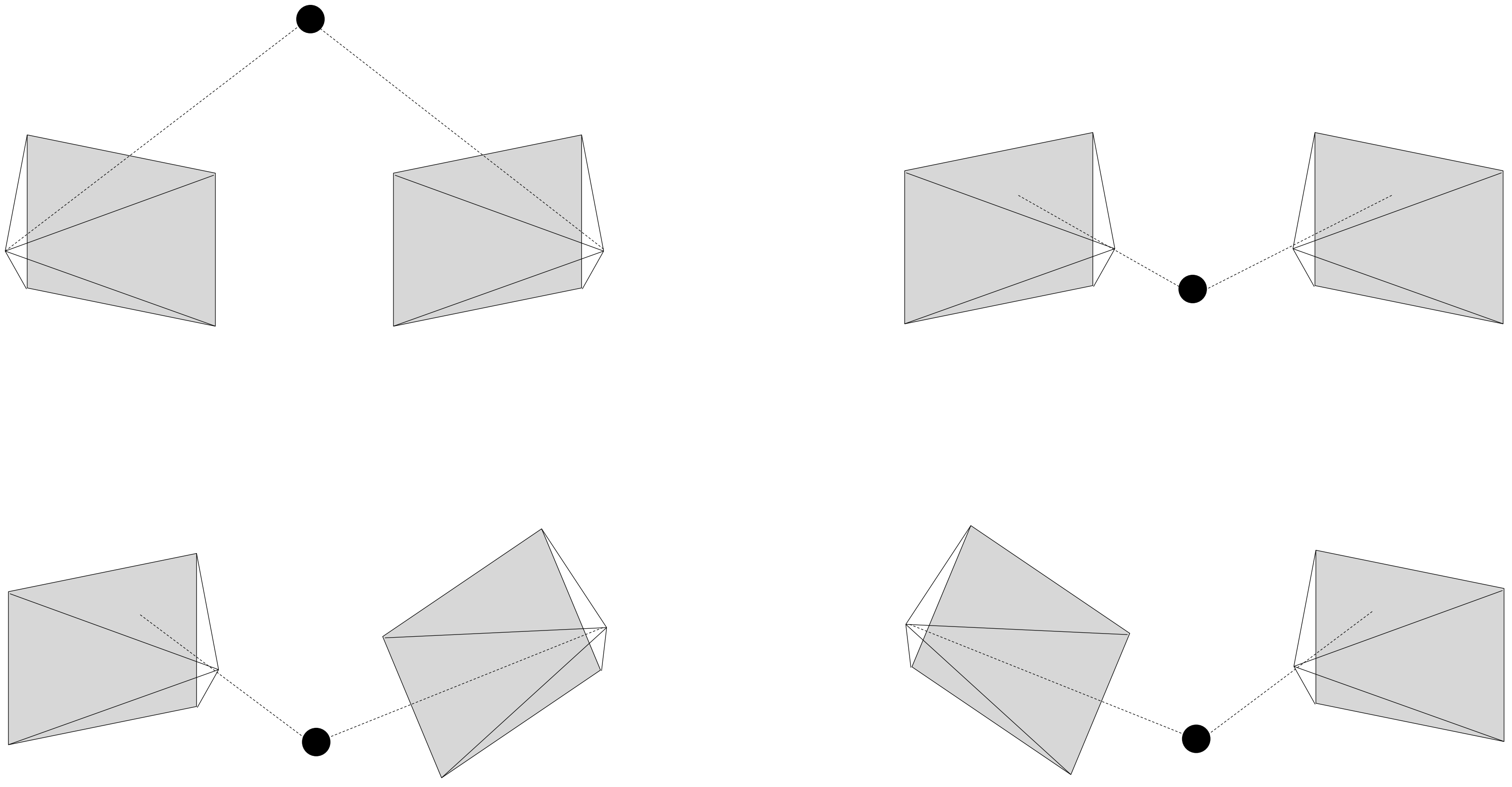
$$\underbrace{\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}}_{\text{Essential matrix}}$$ How to decompose the essential matrix to rotation and translation?

Four configurations:



$$\underbrace{\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}}_{\text{Essential matrix}}$$ How to decompose the essential matrix to rotation and translation?

Four configurations: can be resolved by point triangulation.



2.2 Camera Pose Extraction

Goal Given $\mathbf{E}$, enumerate four camera pose configurations, $(\mathbf{C}_1, \mathbf{R}_1)$, $(\mathbf{C}_2, \mathbf{R}_2)$, $(\mathbf{C}_3, \mathbf{R}_3)$, and $(\mathbf{C}_4, \mathbf{R}_4)$ where $\mathbf{C} \in \mathbb{R}^3$ is the camera center and $\mathbf{R} \in SO(3)$ is the rotation matrix, i.e., $\mathbf{P} = \mathbf{KR}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}$:

```
[Cset Rset] = ExtractCameraPose(E)
```

> (INPUT) `E`: essential matrix
>
> (OUTPUT) `Cset` and `Rset`: four configurations of camera centers and rotations, i.e., `Cset{i}`$=\mathbf{C}_i$ and `Rset{i}`$=\mathbf{R}_i$.

There are four camera pose configurations given an essential matrix. Let $\mathbf{E} = \mathbf{UDV}^\mathsf{T}$ and $\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. The four configurations are enumerated below:

1. $\mathbf{C}_1 = \mathbf{U}(:, 3)$ and $\mathbf{R}_1 = \mathbf{UWV}^\mathsf{T}$
2. $\mathbf{C}_2 = -\mathbf{U}(:, 3)$ and $\mathbf{R}_2 = \mathbf{UWV}^\mathsf{T}$
3. $\mathbf{C}_3 = \mathbf{U}(:, 3)$ and $\mathbf{R}_3 = \mathbf{UW}^\mathsf{T}\mathbf{V}^\mathsf{T}$
4. $\mathbf{C}_4 = -\mathbf{U}(:, 3)$ and $\mathbf{R}_4 = \mathbf{UW}^\mathsf{T}\mathbf{V}^\mathsf{T}$.

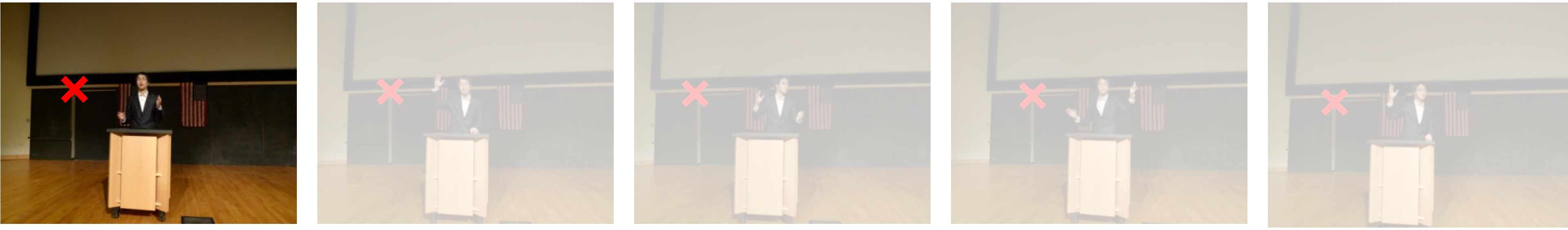
Note that the determinant of a rotation matrix is one. If $\det(\mathbf{R}) = -1$, the camera pose must be corrected, i.e., $\mathbf{C} \leftarrow -\mathbf{C}$ and $\mathbf{R} \leftarrow -\mathbf{R}$.

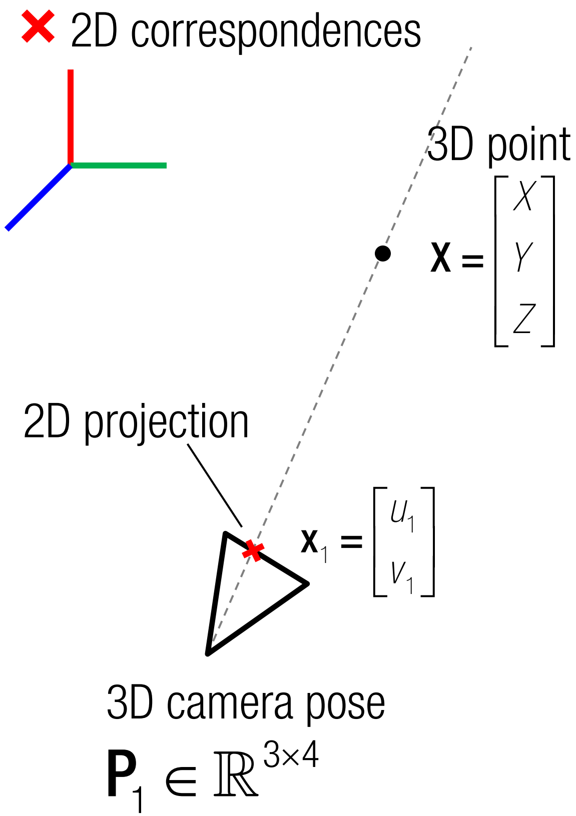
Point Triangulation

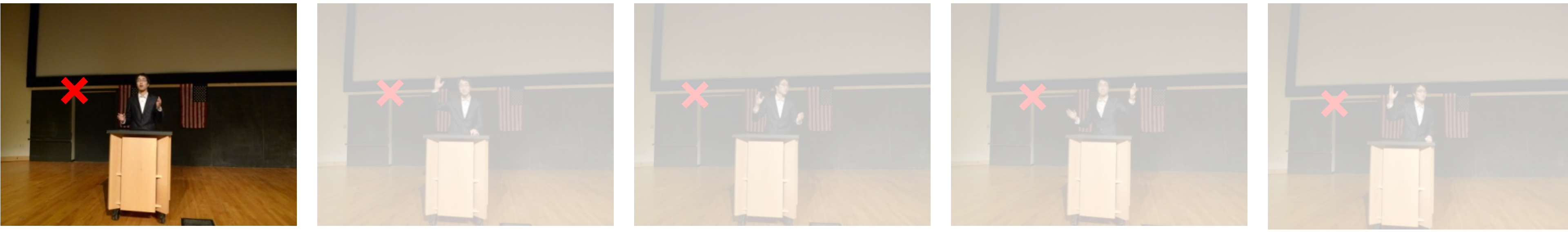
✕ 2D correspondences

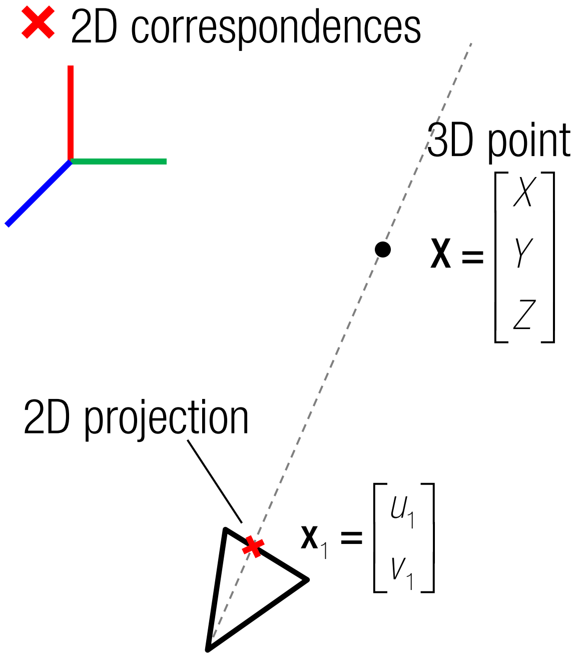
Point Triangulation





Point Triangulation



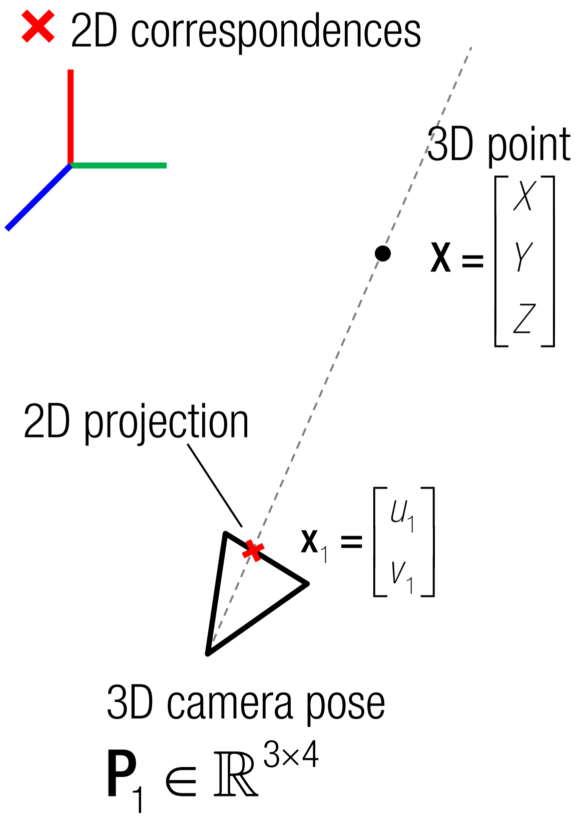


3D camera pose

$\mathbf{P}_1 \in \mathbb{R}^{3\times 4}$

$$\lambda \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

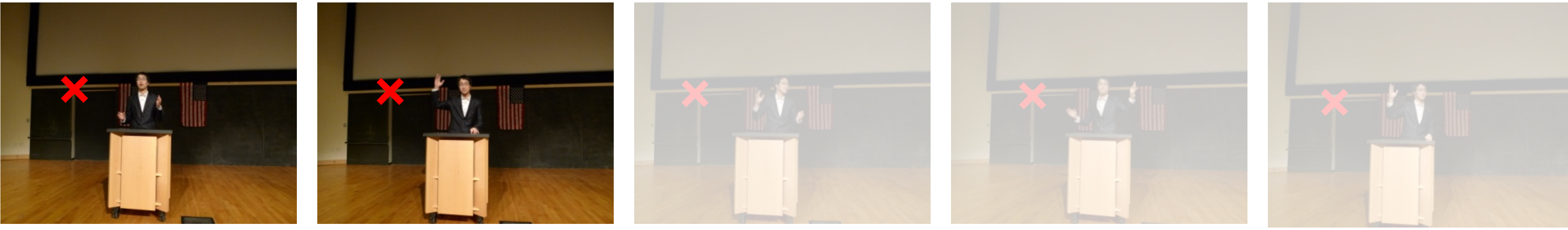
Point Triangulation

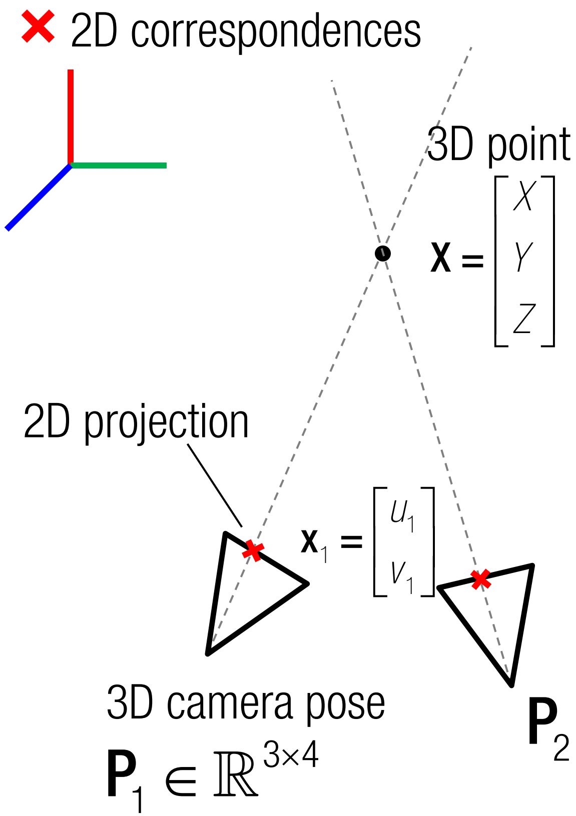


$$\lambda \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

Cross product between two parallel vectors equals to zero.

Point Triangulation

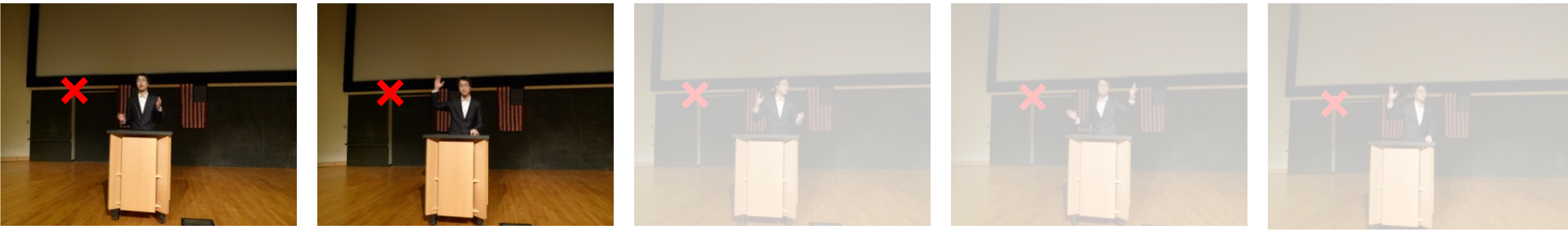


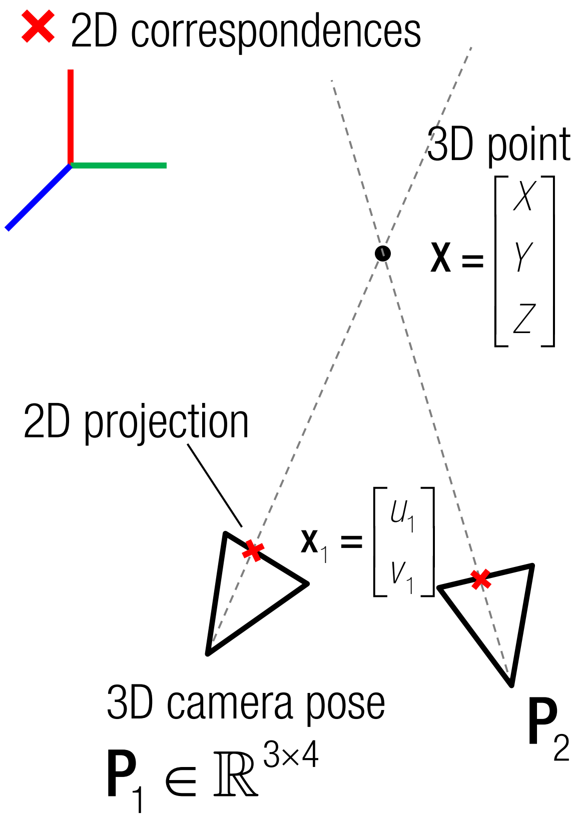


$$\lambda \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times \mathbf{P}_2 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

Point Triangulation



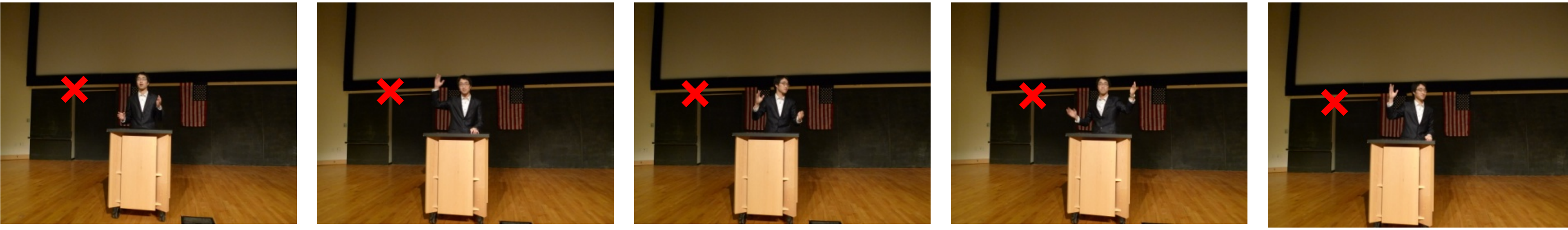


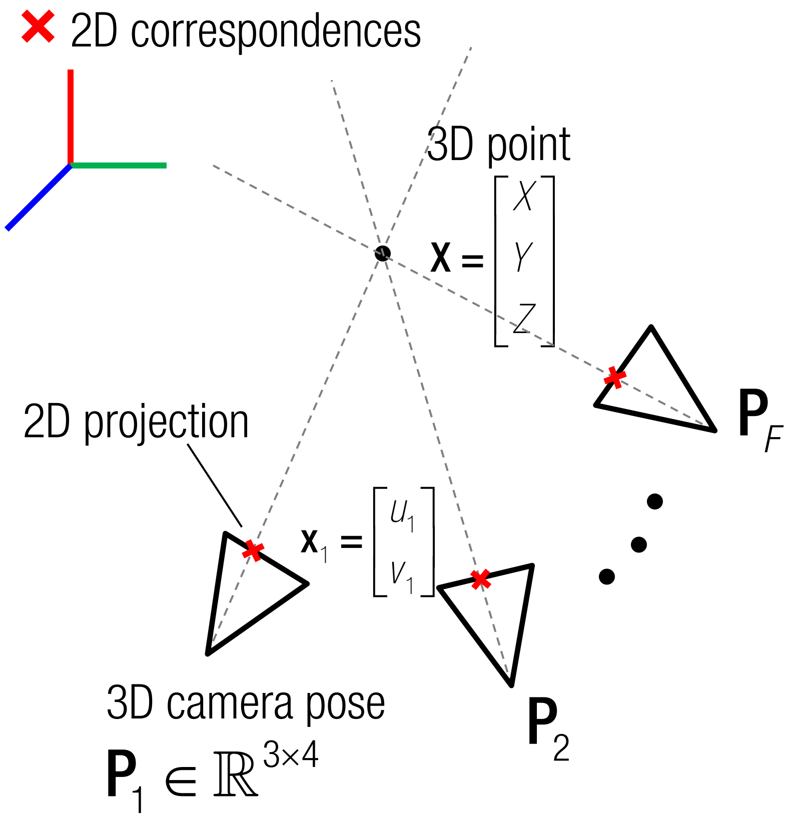
$$\lambda \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times \mathbf{P}_2 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \\ \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times \mathbf{P}_2 \\ \\ \\ \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

Point Triangulation

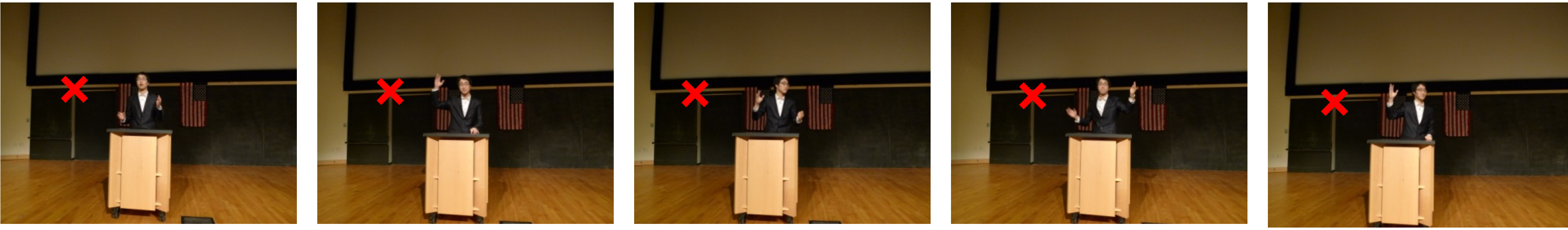


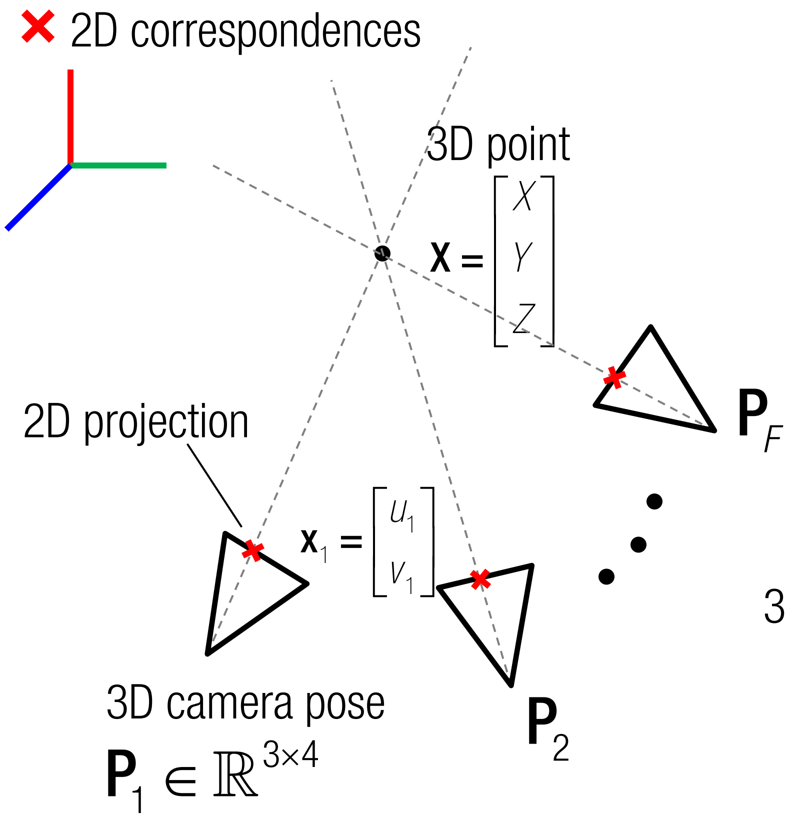


$$\lambda \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \longrightarrow \begin{aligned} \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} &= \mathbf{0} \\ \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times \mathbf{P}_2 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} &= \mathbf{0} \end{aligned}$$

$$\begin{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \\ \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times \mathbf{P}_2 \\ \vdots \\ \begin{bmatrix} \mathbf{x}_F \\ 1 \end{bmatrix}_\times \mathbf{P}_F \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

Point Triangulation





$$\lambda \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix} = \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times \mathbf{P}_2 \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

$$3F\left\{ \overbrace{\begin{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \\ \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times \mathbf{P}_2 \\ \vdots \\ \begin{bmatrix} \mathbf{x}_F \\ 1 \end{bmatrix}_\times \mathbf{P}_F \end{bmatrix}}^{4} \right. \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

$$\text{rank}\left(\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}_\times \mathbf{P} \right) = 2$$

Least squares if $F \geq 2$

Point Triangulation

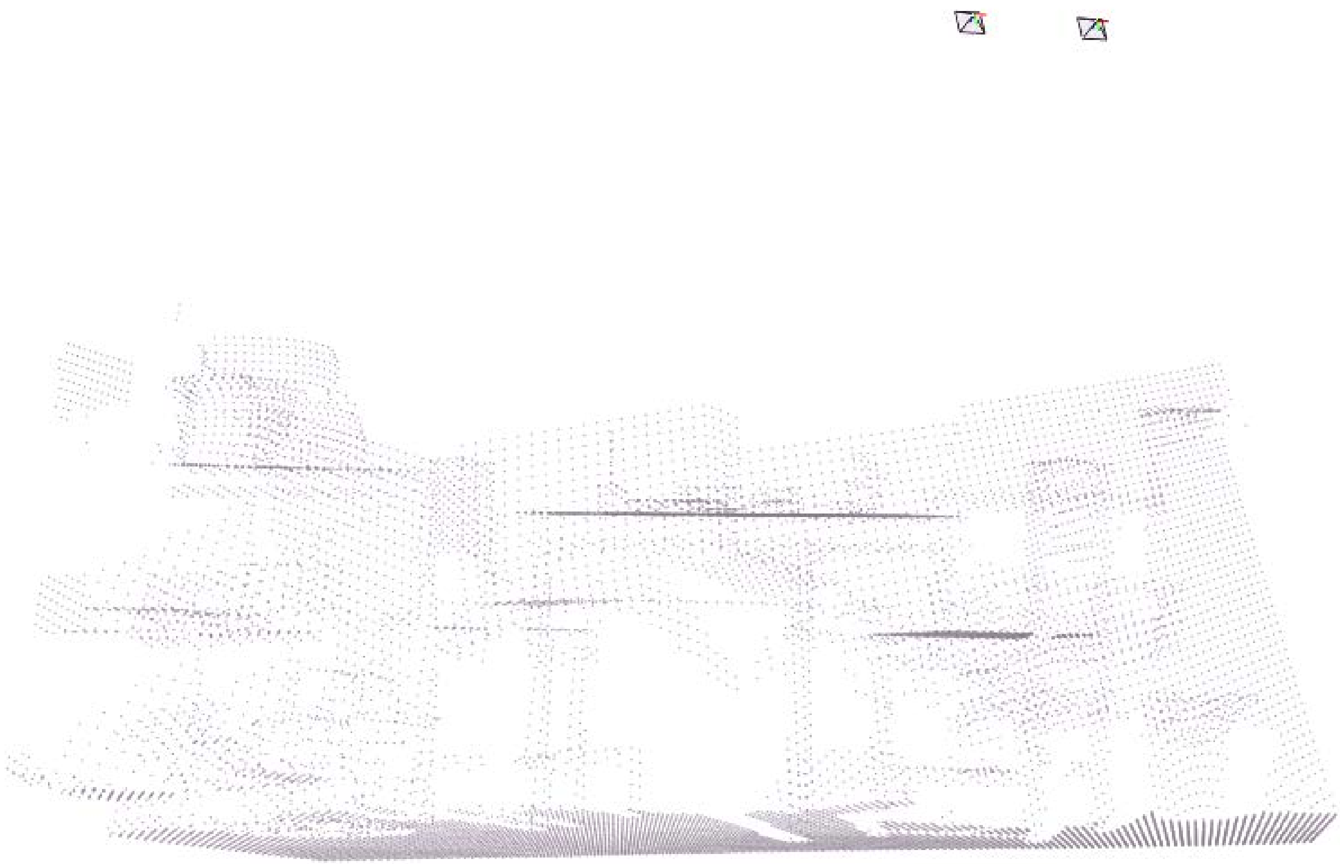
$$\mathbf{P}_1 = \mathbf{K}_1\begin{bmatrix}\mathbf{I}_{3\times 3} & \mathbf{0}_3\end{bmatrix}$$

$$\mathbf{P}_2 = \mathbf{K}_2\begin{bmatrix}\mathbf{I}_{3\times 3} & -\mathbf{C}\end{bmatrix} \quad \mathbf{C} = \begin{bmatrix}1\\0\\0\end{bmatrix}$$

```
% Intrinsic parameter
K1 = [2329.558 0 1141.452; 0 2329.558 927.052; 0 0 1];
K2 = [2329.558 0 1241.731; 0 2329.558 927.052; 0 0 1];

% Camera matrices
P1 = K1 * [eye(3) zeros(3,1)];
C = [1;0;0];
P2 = K2 * [eye(3) -C];

% Correspondences
x1 = [1382;986;1];
x2 = [1144;986;1];
skew1 = Vec2Skew(x1);
skew2 = Vec2Skew(x2);

% Solve
A = [skew1*P1; skew2*P2];
[u,d,v] = svd(A);
X = v(:,end)/v(end,end);

function skew = Vec2Skew(v)
skew  = [0 -v(3) v(2); v(3) 0 -v(1); -v(2) v(1) 0];
```

```
X =
    0.7111
    0.1743
    6.8865
    1.0000
```

Point Triangulation



3.1 Linear Triangulation

Goal Given two camera poses, $(\mathbf{C}_1, \mathbf{R}_1)$ and $(\mathbf{C}_2, \mathbf{R}_2)$, and correspondences $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$, triangulate 3D points using linear least squares:

```
X = LinearTriangulation(K, C1, R1, C2, R2, x1, x2)
```

(INPUT) `C1` and `R1`: the first camera pose

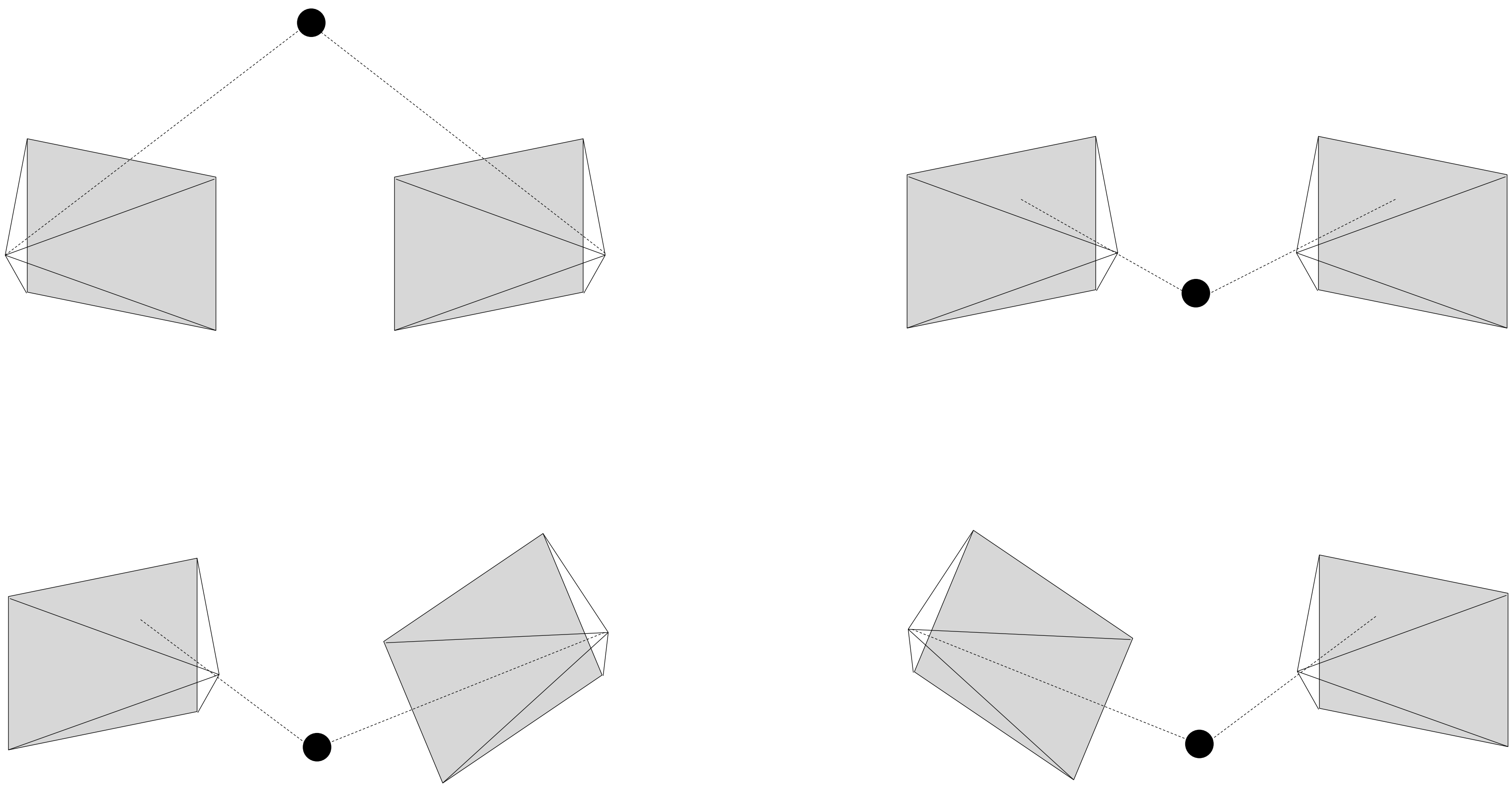
(INPUT) `C2` and `R2`: the second camera pose

(INPUT) `x1` and `x2`: two $N \times 2$ matrices whose row represents correspondence between the first and second images where N is the number of correspondences.

(OUTPUT) `X`: $N \times 3$ matrix whose row represents 3D triangulated point.

Camera pose disambiguation via point triangulation

Four configurations:



3.2 Camera Pose Disambiguation

Goal Given four camera pose configuration and their triangulated points, find the unique camera pose by checking the *cheirality* condition—the reconstructed points must be in front of the cameras:

```
[C R X0] = DisambiguateCameraPose(Cset, Rset, Xset)
```

(INPUT) `Cset` and `Rset`: four configurations of camera centers and rotations

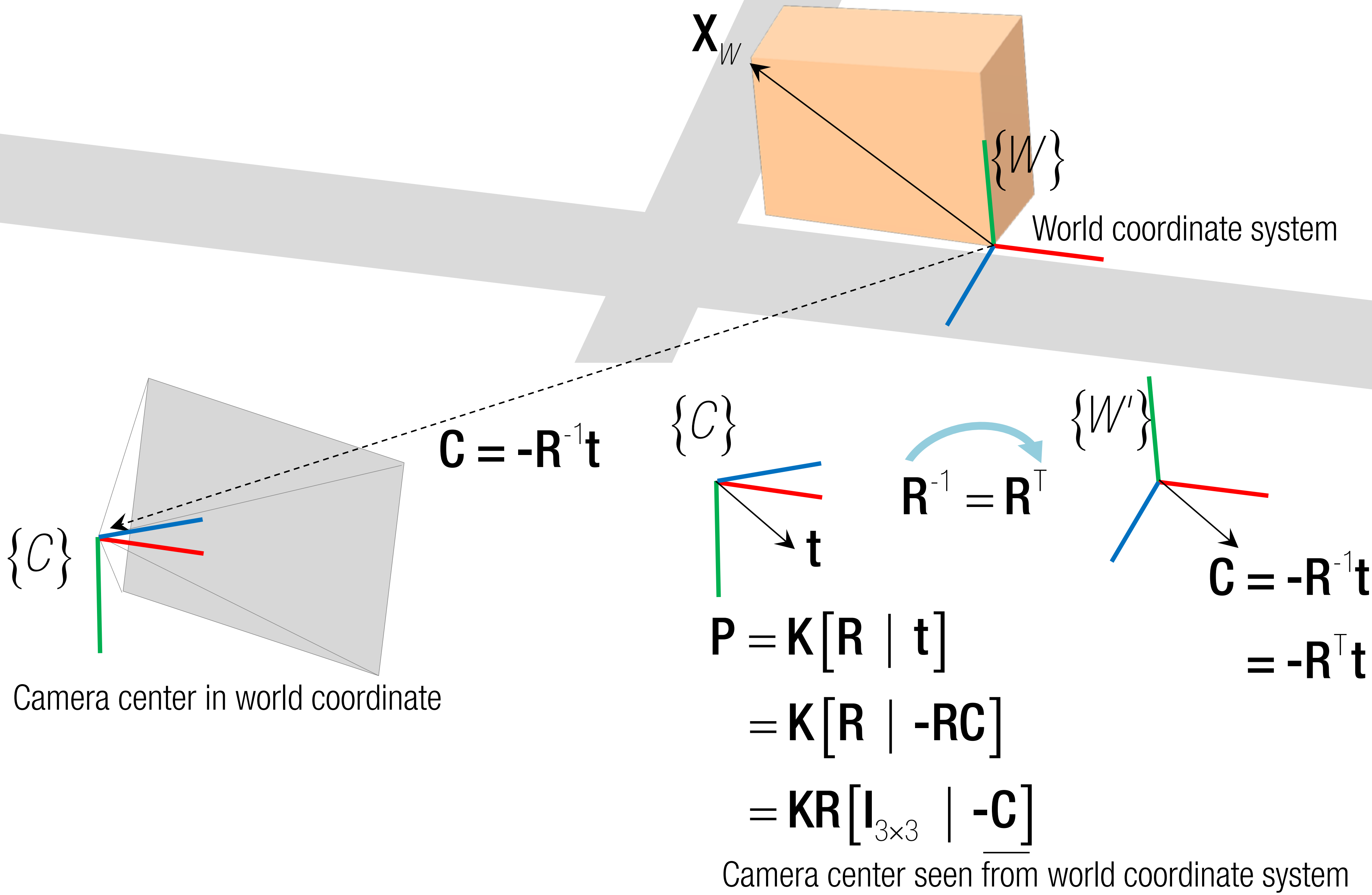
(INPUT) `Xset`: four sets of triangulated points from four camera pose configurations

(OUTPUT) `C` and `R`: the correct camera pose

(OUTPUT) `X0`: the 3D triangulated points from the correct camera pose

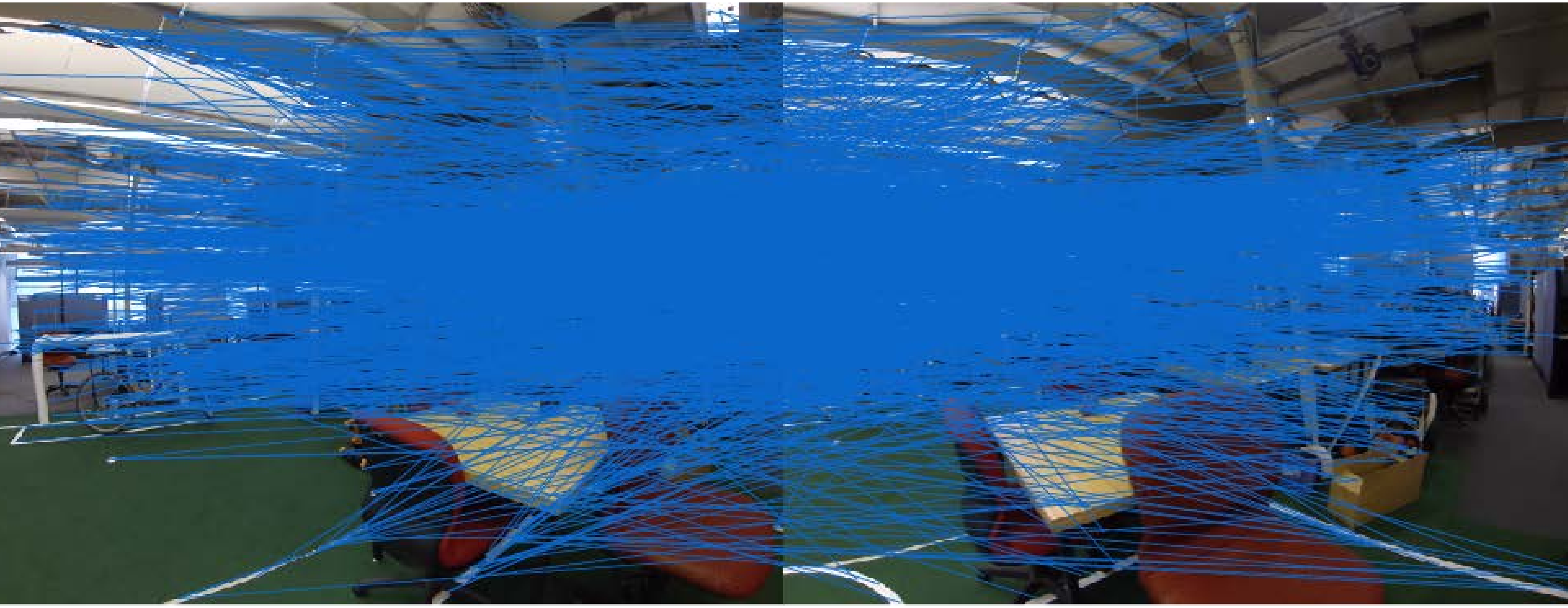
The sign of the Z element in the camera coordinate system indicates the location of the 3D point with respect to the camera, i.e., a 3D point $\mathbf{X}$ is in front of a camera if $(\mathbf{C}, \mathbf{R})$ if $\mathbf{r}_3(\mathbf{X} - \mathbf{C}) > 0$ where $\mathbf{r}_3$ is the third row of $\mathbf{R}$. Not all triangulated points satisfy this condition due to the presence of correspondence noise. The best camera configuration, $(\mathbf{C}, \mathbf{R}, \mathbf{X})$ is the one that produces the maximum number of points satisfying the cheirality condition.

Third person (world) perspective



Feature Matching using Fundamental Matrix

Feature Matching using Fundamental Matrix



Nearest neighbor search between two images

Feature Matching using Fundamental Matrix

8 (bad) points to compute fundamental matrix

$$F = \begin{matrix} 0.0000 & 0.0000 & -0.0159 \\ 0.0000 & -0.0000 & -0.0001 \\ 0.0102 & -0.0004 & 0.9998 \end{matrix}$$

Feature Matching using Fundamental Matrix

8 (bad) points to compute fundamental matrix

F =

0.0000	0.0000	-0.0159
0.0000	-0.0000	-0.0001
0.0102	-0.0004	0.9998

# of inliers: 65

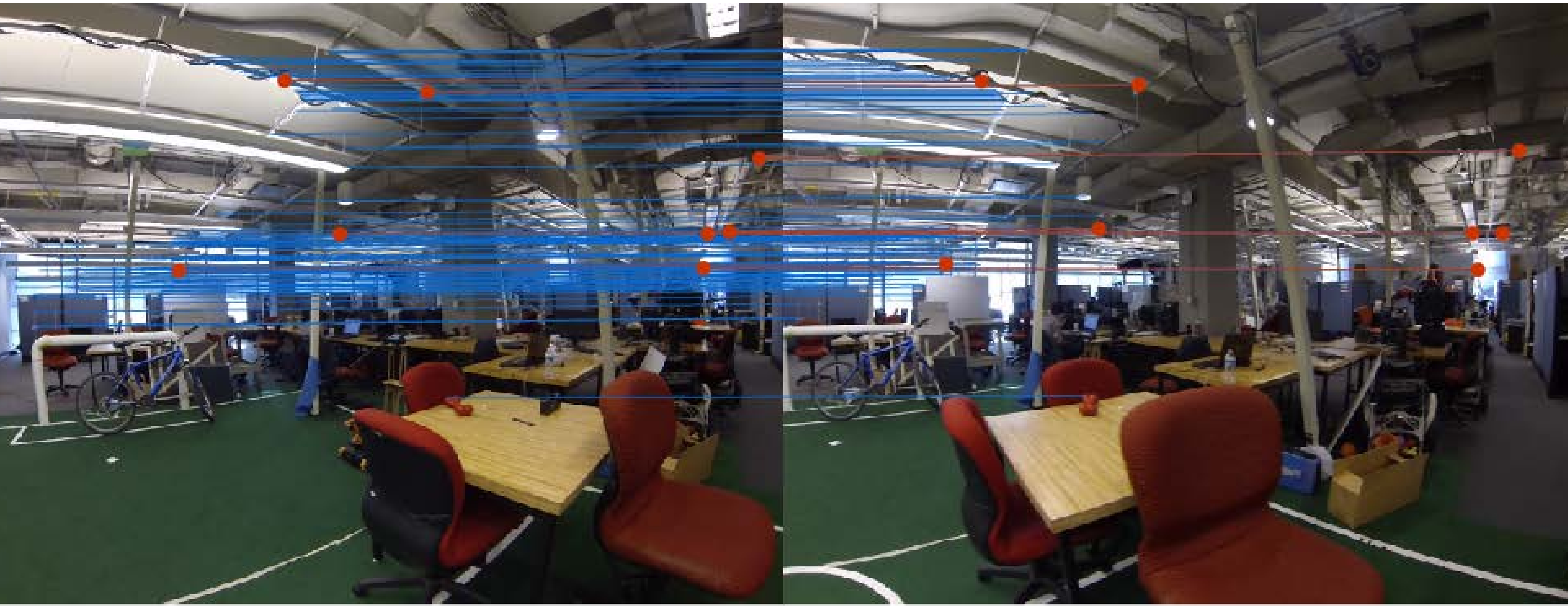
Feature Matching using Fundamental Matrix

8 (good) points to compute fundamental matrix

F =

$$\begin{matrix} 0.0000 & -0.0000 & 0.0017 \\ 0.0000 & -0.0000 & -0.0169 \\ -0.0033 & 0.0148 & 0.9997 \end{matrix}$$

Feature Matching using Fundamental Matrix



8 (good) points to compute fundamental matrix

F =

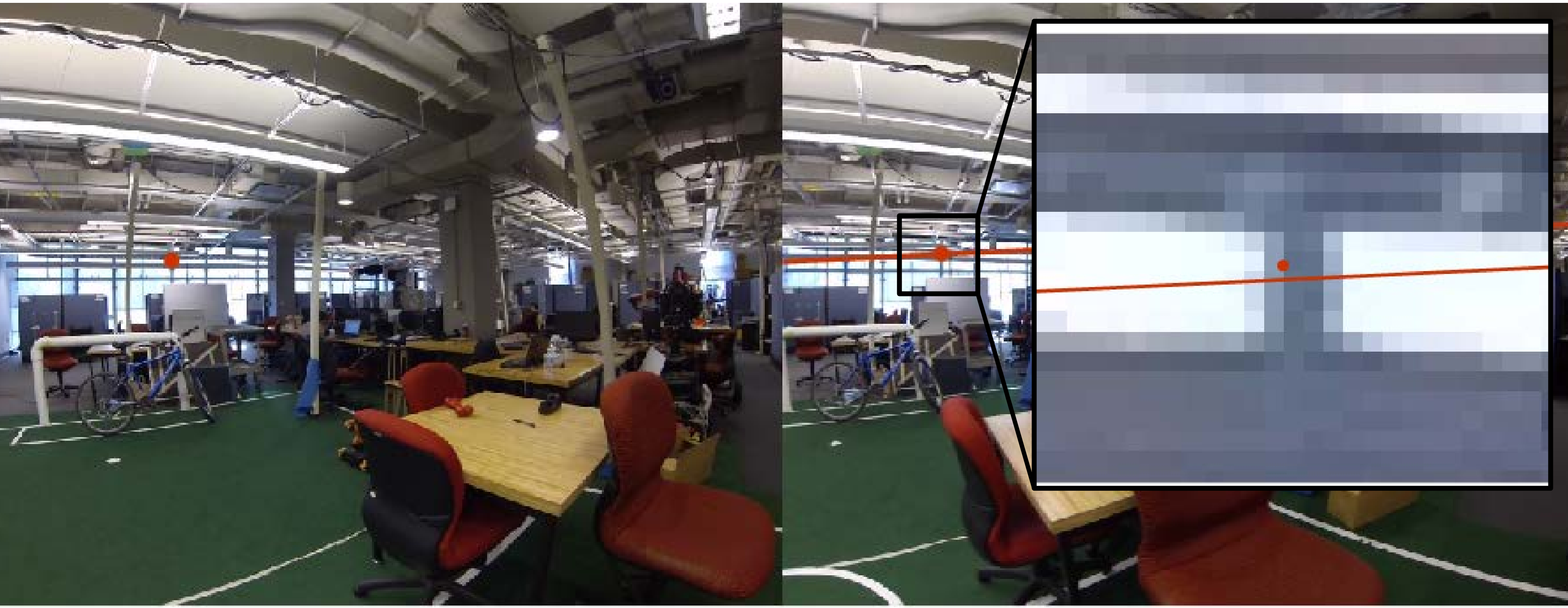
0.0000	-0.0000	0.0017
0.0000	-0.0000	-0.0169
-0.0033	0.0148	0.9997

# of inliers: 118

Feature Matching using Fundamental Matrix

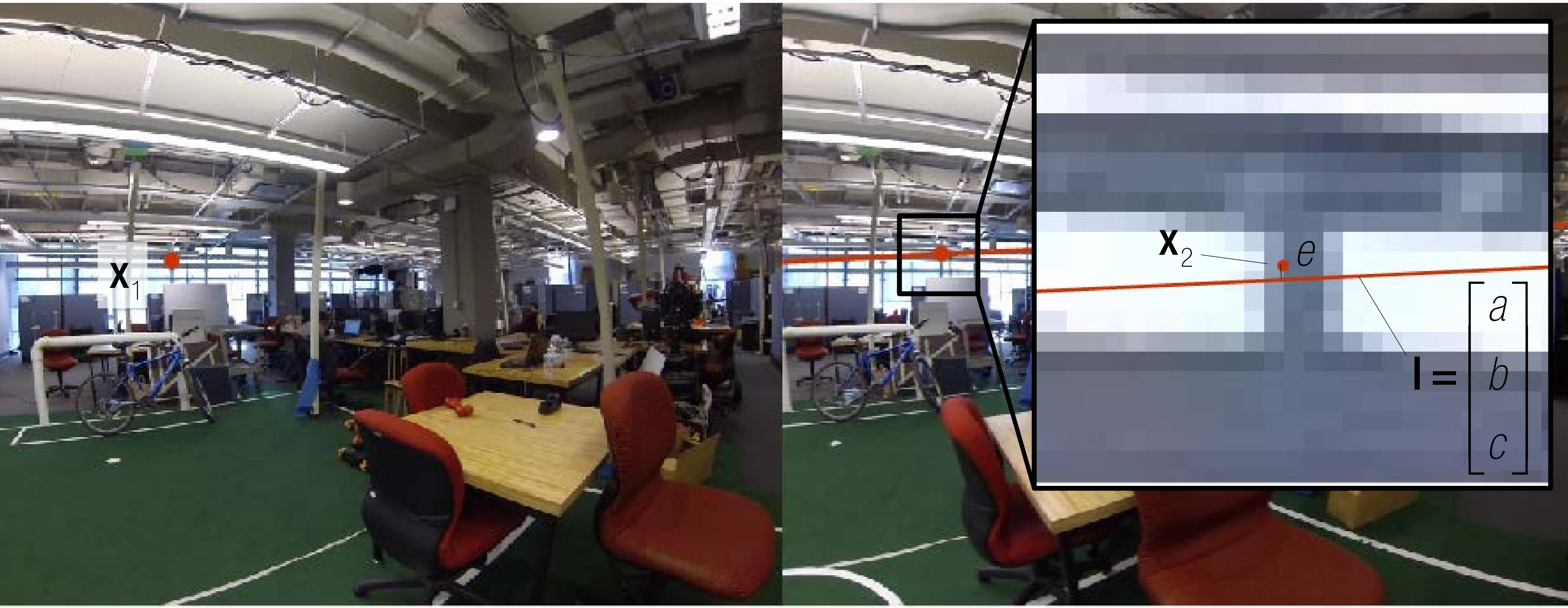
Epipolar line

Feature Matching using Fundamental Matrix



Epipolar line

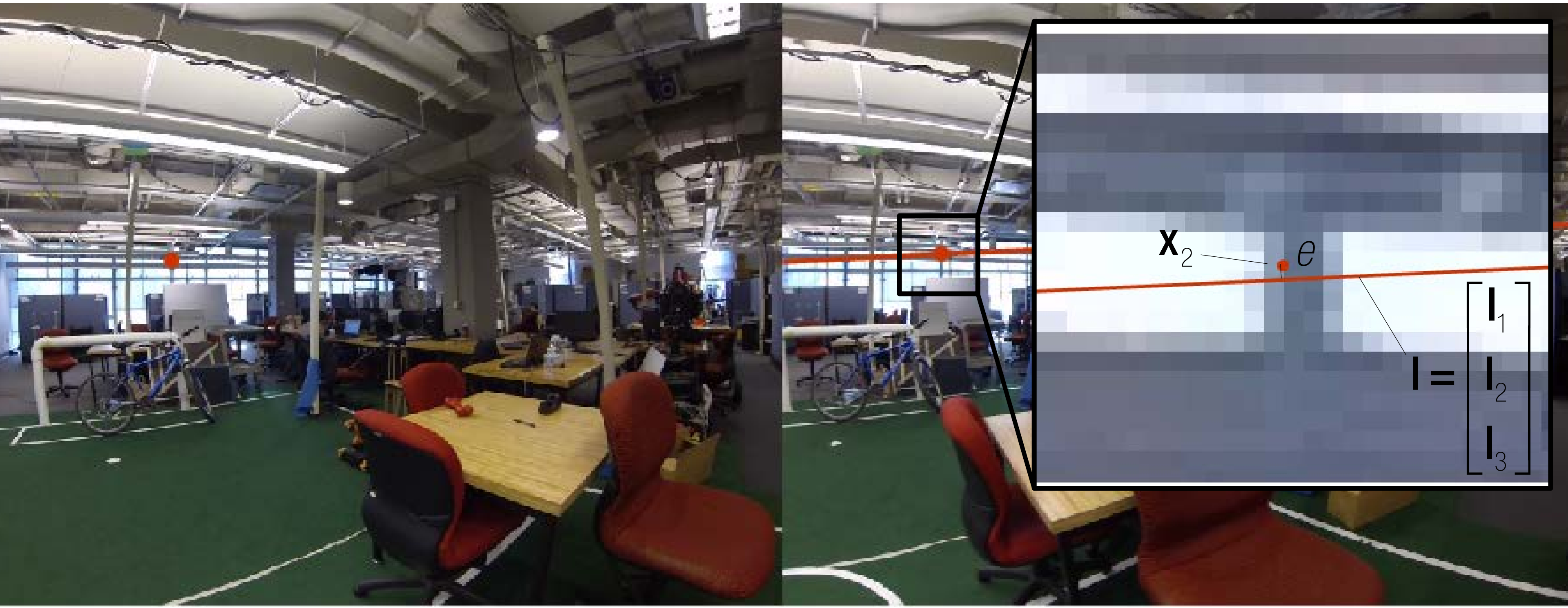
Feature Matching using Fundamental Matrix



Epipolar line

$$e = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}}$$

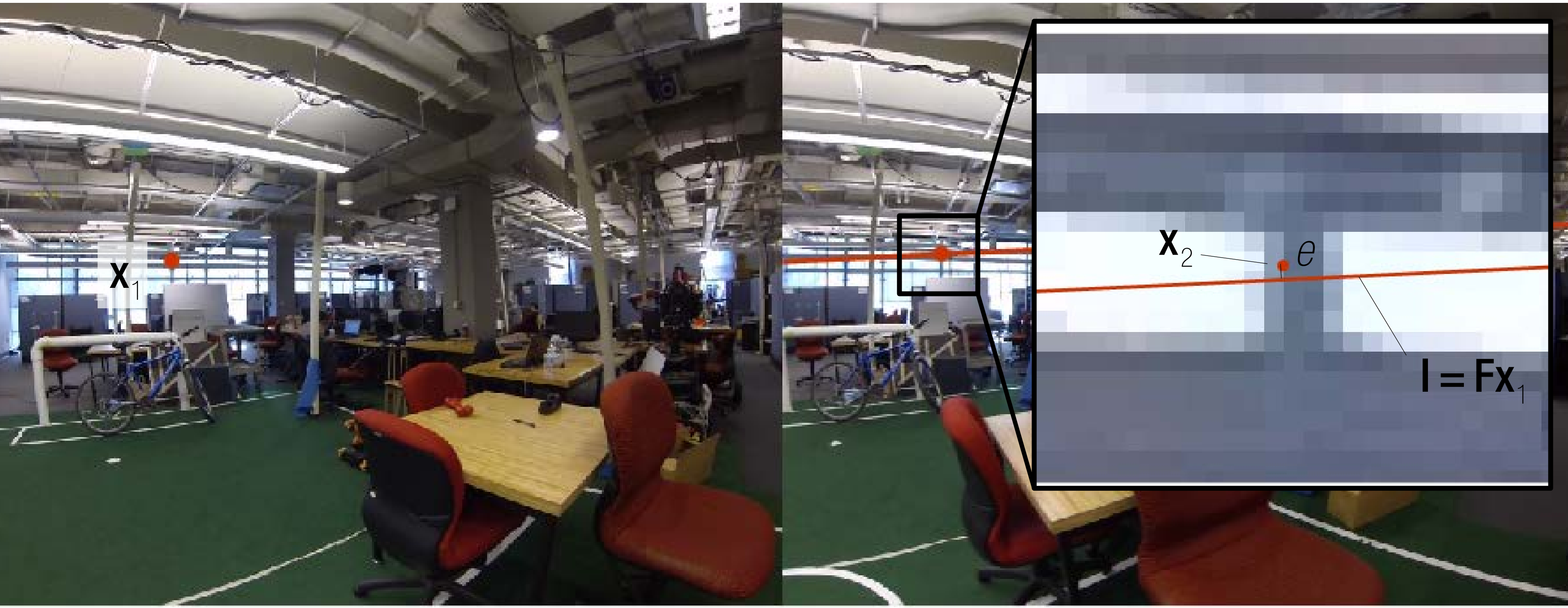
Feature Matching using Fundamental Matrix



Epipolar line

$$e = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}} = \frac{|\mathbf{x}_2^{\mathrm{T}}\mathbf{l}|}{\sqrt{\mathbf{l}_1^2 + \mathbf{l}_2^2}}$$

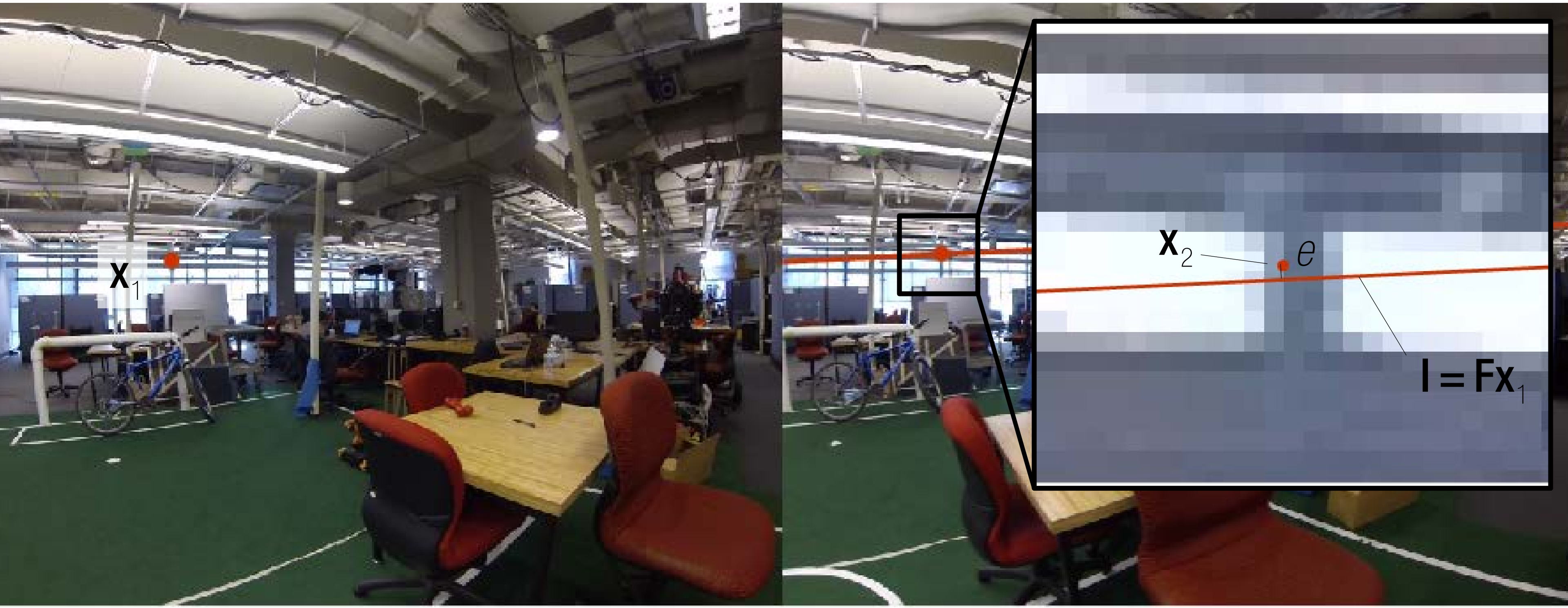
Feature Matching using Fundamental Matrix



Epipolar line

$$e = \frac{\left|ax + by + c\right|}{\sqrt{a^2 + b^2}} = \frac{\left|\mathbf{x}_2^{\mathsf{T}}\mathbf{l}\right|}{\sqrt{\mathbf{l}_1^2 + \mathbf{l}_2^2}} = \frac{\left|\mathbf{x}_2^{\mathsf{T}}\mathbf{F}\mathbf{x}_1\right|}{\sqrt{\left(\mathbf{F}_1\mathbf{x}_1\right)^2 + \left(\mathbf{F}_2\mathbf{x}_1\right)^2}}$$

Feature Matching using Fundamental Matrix

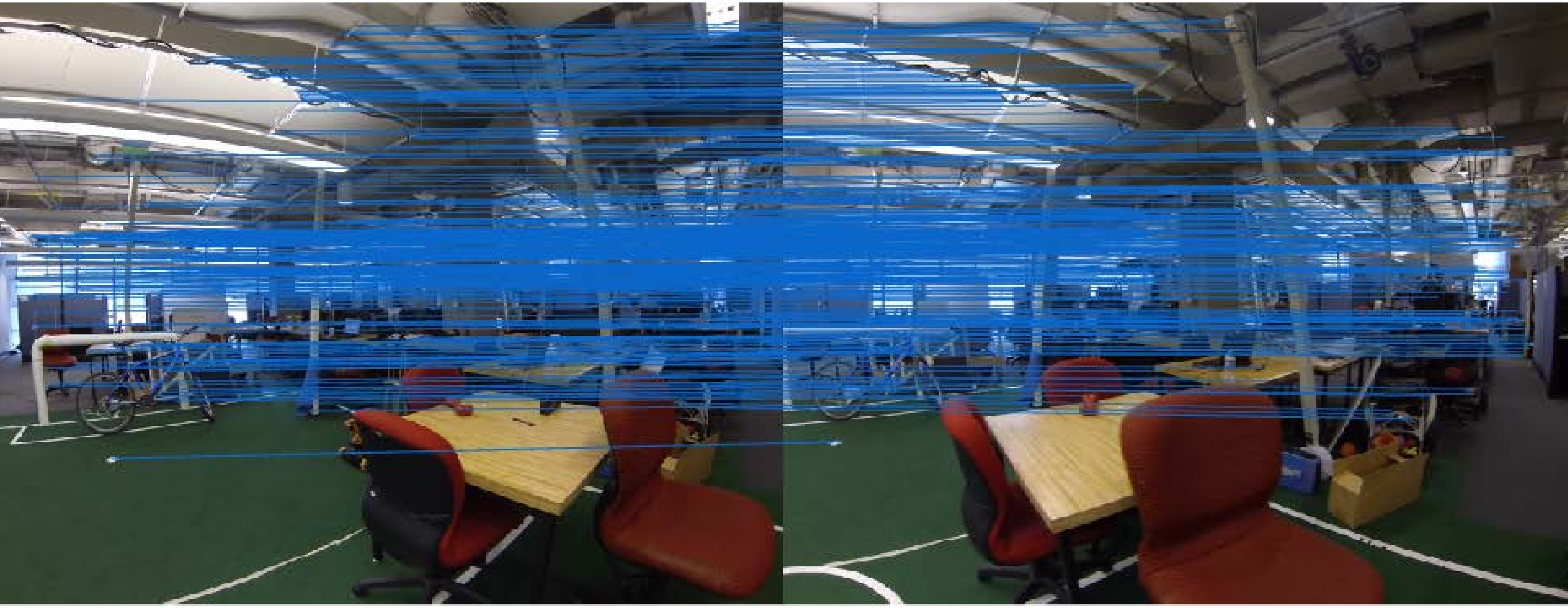


Epipolar line

$$e = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}} = \frac{|\mathbf{x}_2^{\mathsf{T}}\mathbf{l}|}{\sqrt{\mathbf{l}_1^2 + \mathbf{l}_2^2}} = \frac{|\mathbf{x}_2^{\mathsf{T}}\mathbf{F}\mathbf{x}_1|}{\sqrt{(\mathbf{F}_1\mathbf{x}_1)^2 + (\mathbf{F}_2\mathbf{x}_1)^2}}$$

Epipolar error: 0.7089 pixel error

Feature Matching using Fundamental Matrix

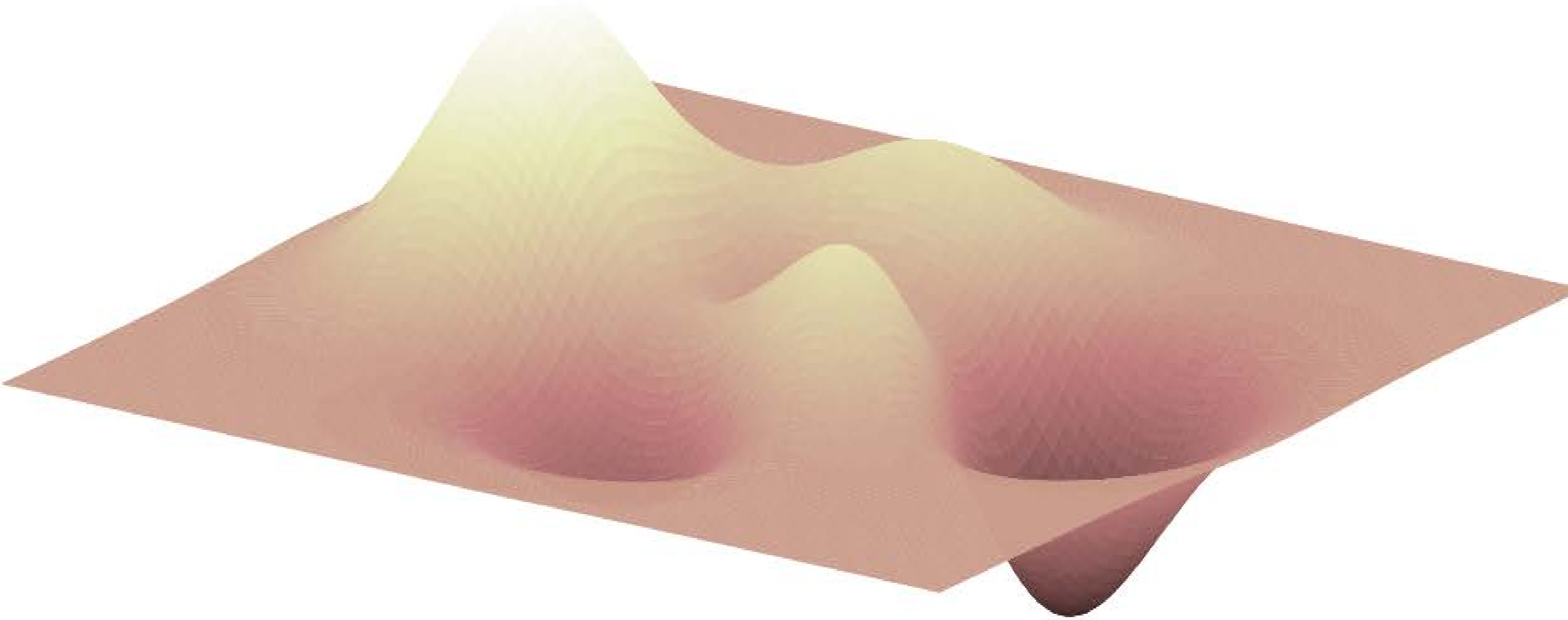


Final inliers using RANSAC

$$F = \begin{matrix} 0.0000 & 0.0000 & -0.0031 \\ -0.0000 & -0.0000 & 0.0283 \\ 0.0017 & -0.0294 & 1.0000 \end{matrix}$$

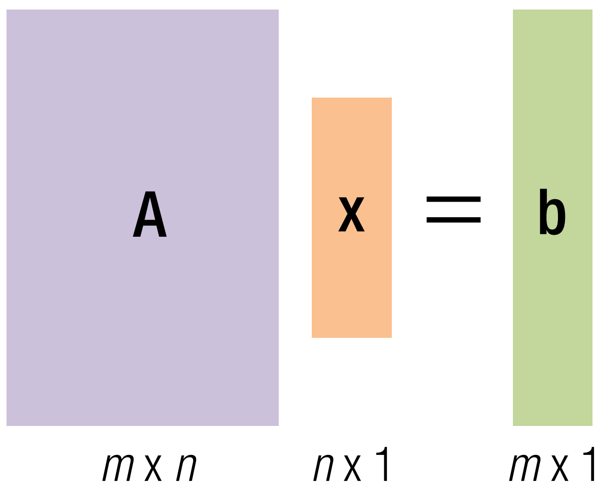
# of inliers: 443

Nonlinear Least Squares



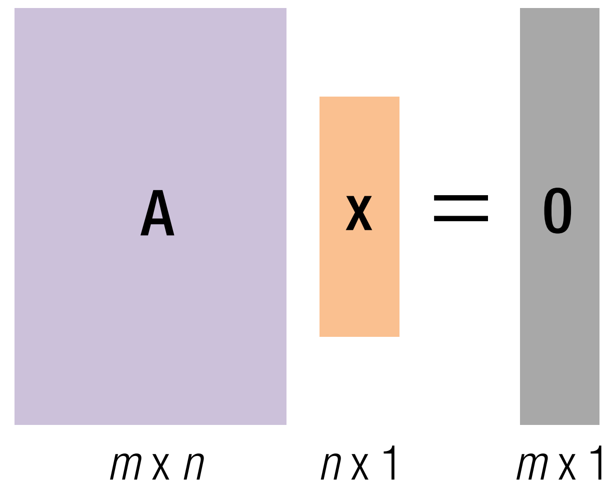
So far,

Type I



$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|^2$$

Type II



$$\min_{\mathbf{x}} \|\mathbf{Ax}\|^2 \text{ subject to } \|\mathbf{x}\| = 1$$

Why Linear Least Squares?

$$\min_{x} \left\| Ax - b \right\|^2 = \min_{x} \left(Ax - b \right)^T \left(Ax - b \right)$$

Why Linear Least Squares?

$$\min_{\mathbf{x}} \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|^2 = \min_{\mathbf{x}} \left(\mathbf{A}\mathbf{x} - \mathbf{b} \right)^{\top} \left(\mathbf{A}\mathbf{x} - \mathbf{b} \right)$$

$$= \min_{\mathbf{x}} \mathbf{x}^{\top} \mathbf{A}^{\top} \mathbf{A}\mathbf{x} - 2\mathbf{b}^{\top} \mathbf{A}\mathbf{x} - \mathbf{b}^{\top} \mathbf{b}$$

Why Linear Least Squares?

$$
\begin{aligned}
\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 &= \min_{\mathbf{x}} (\mathbf{A}\mathbf{x}-\mathbf{b})^\top (\mathbf{A}\mathbf{x}-\mathbf{b}) \\
&= \min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} - 2\mathbf{b}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{b} \\
&= \min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} - 2\mathbf{b}^\top \mathbf{A}\mathbf{x}
\end{aligned}
$$

Why Linear Least Squares?

$$\min_x \|Ax - b\|^2 = \min_x (Ax - b)^\top (Ax - b)$$

$$= \min_x x^\top A^\top Ax - 2b^\top Ax - b^\top b$$

$$= \min_x x^\top A^\top Ax - 2b^\top Ax$$

$$= \min_x x^\top Qx + cx \qquad \text{where } Q = A^\top A, \quad c = -2b^\top A$$

Why Linear Least Squares?

$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|^2 = \min_{\mathbf{x}} (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b})$$

$$= \min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{b}^\top \mathbf{Ax} - \mathbf{b}^\top \mathbf{b}$$

$$= \min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{b}^\top \mathbf{Ax}$$

$$= \min_{\mathbf{x}} \underbrace{\mathbf{x}^\top \mathbf{Qx} + \mathbf{cx}}_{\text{Quadratic equation in } \mathbf{x}.} \qquad \text{where } \mathbf{Q} = \mathbf{A}^\top \mathbf{A}, \quad \mathbf{c} = -2\mathbf{b}^\top \mathbf{A}$$

Q is positive definite (all eigen values of **Q** are positive). $\longrightarrow$ There exists the global solution **x**.

Why Linear Least Squares?

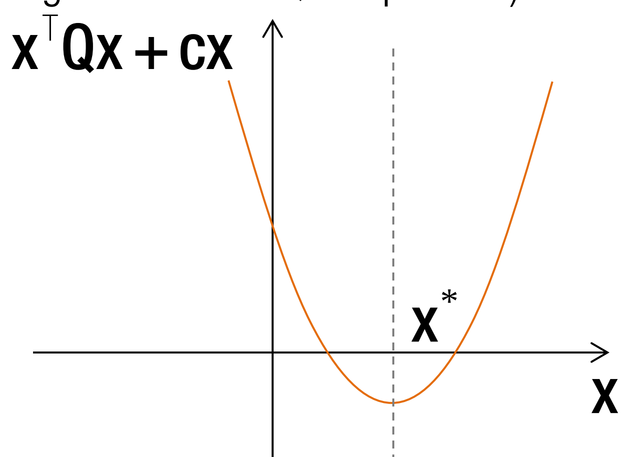
$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|^2 = \min_{\mathbf{x}} (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b})$$

$$= \min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{b}^\top \mathbf{Ax} - \mathbf{b}^\top \mathbf{b}$$

$$= \min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{b}^\top \mathbf{Ax}$$

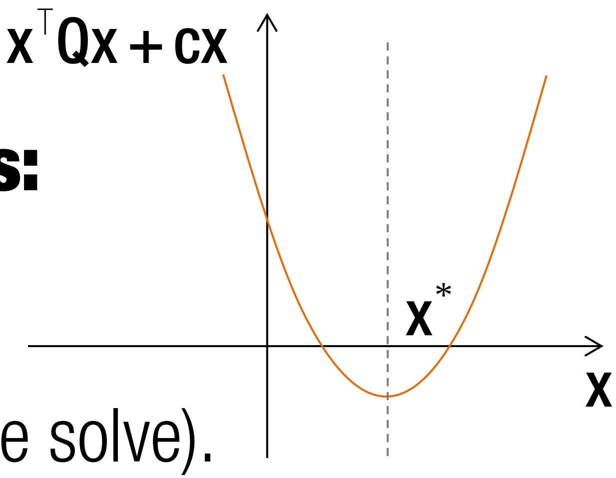
$$= \min_{\mathbf{x}} \underbrace{\mathbf{x}^\top \mathbf{Qx} + \mathbf{cx}}_{\text{Quadratic equation in } \mathbf{x}.} \qquad \text{where } \mathbf{Q} = \mathbf{A}^\top \mathbf{A}, \quad \mathbf{c} = -2\mathbf{b}^\top \mathbf{A}$$

Q is positive definite (all eigen values of **Q** are positive). $\longrightarrow$ There exists the global solution **x**.



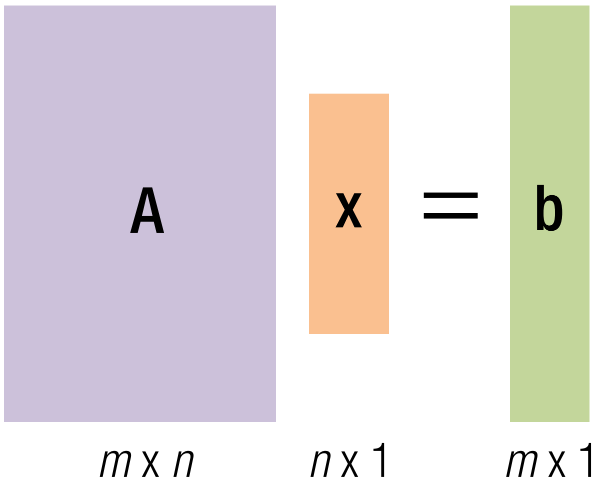
$\mathbf{x}^*$: global solution

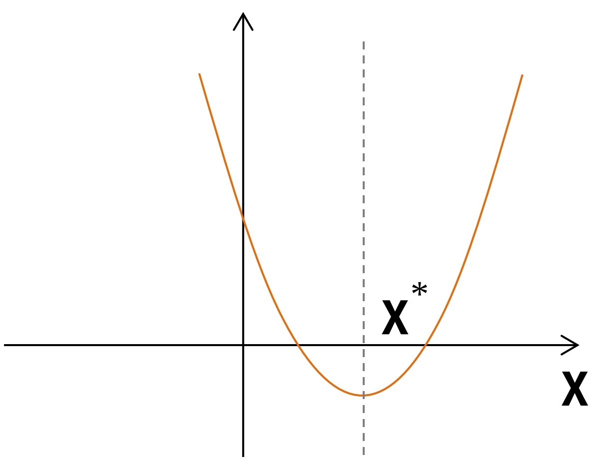
The properties of *linear* least squares:



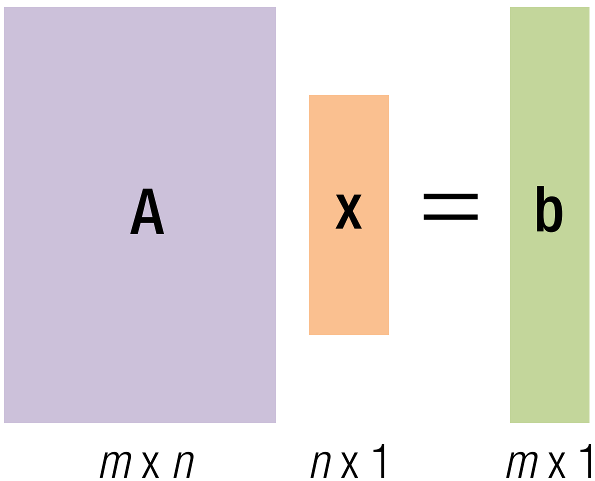
- Has the global/unique solution.
- Has the closed form solution (non-iterative solve).
- Is solved efficiently (SVD).
- Requires no extra parameters such an initialization.

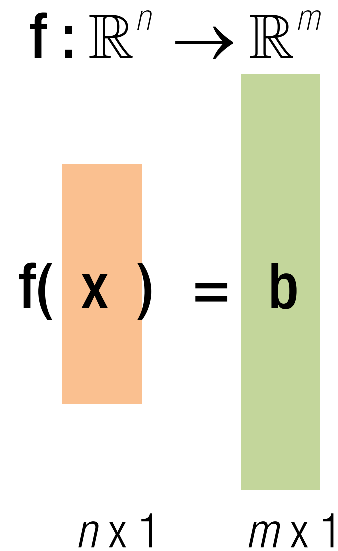
Life isn't that easy.

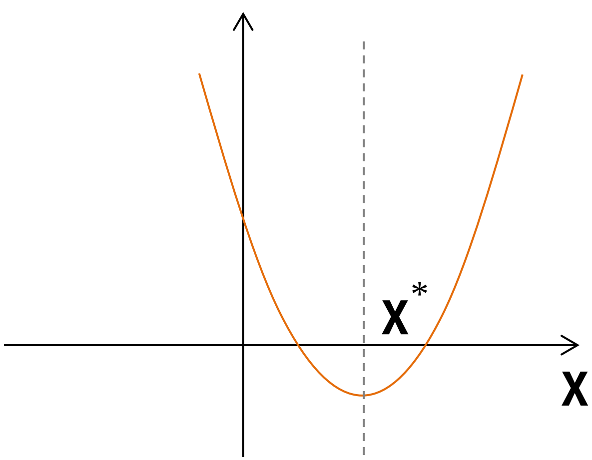




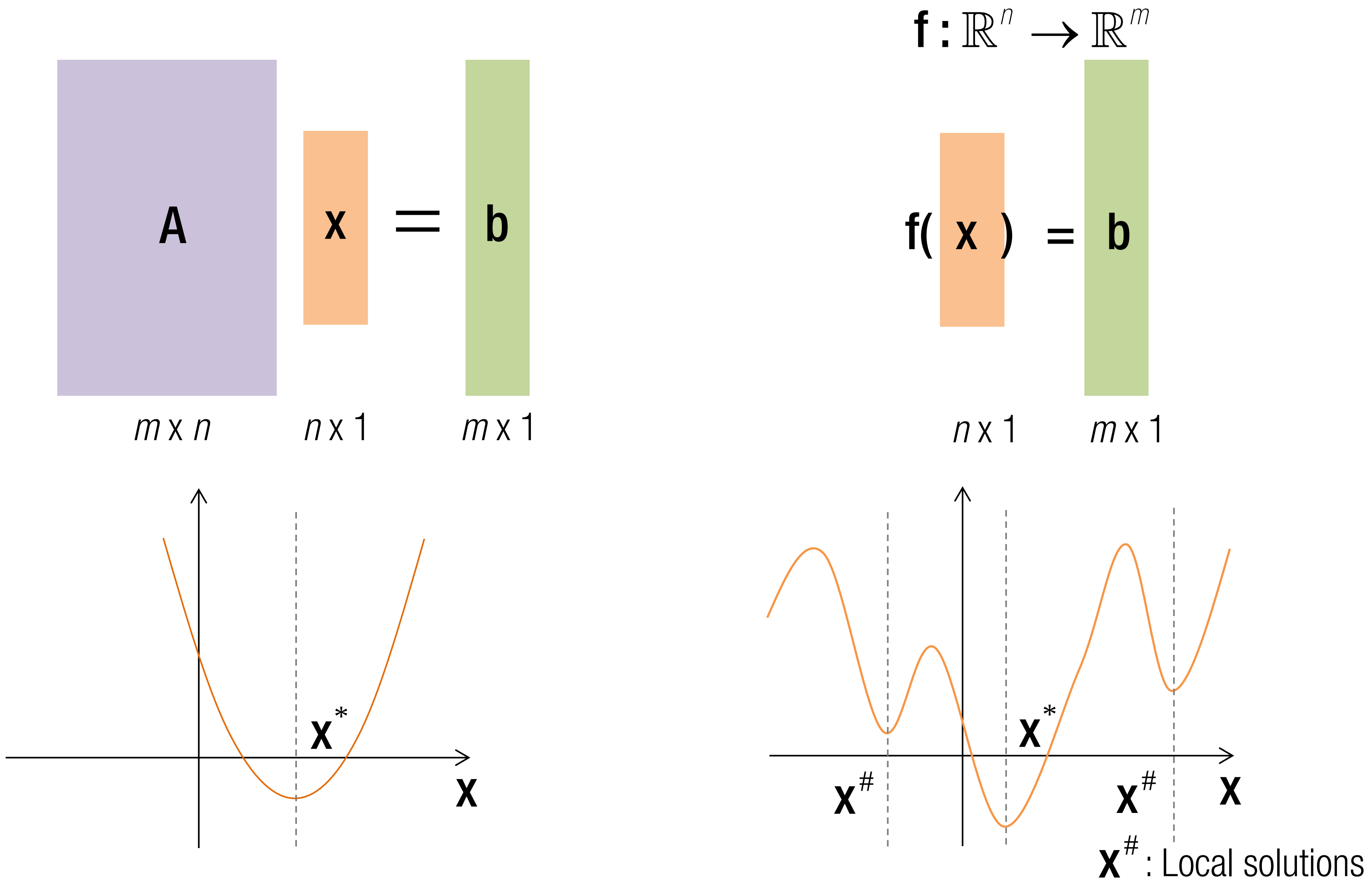
Life isn't that easy.



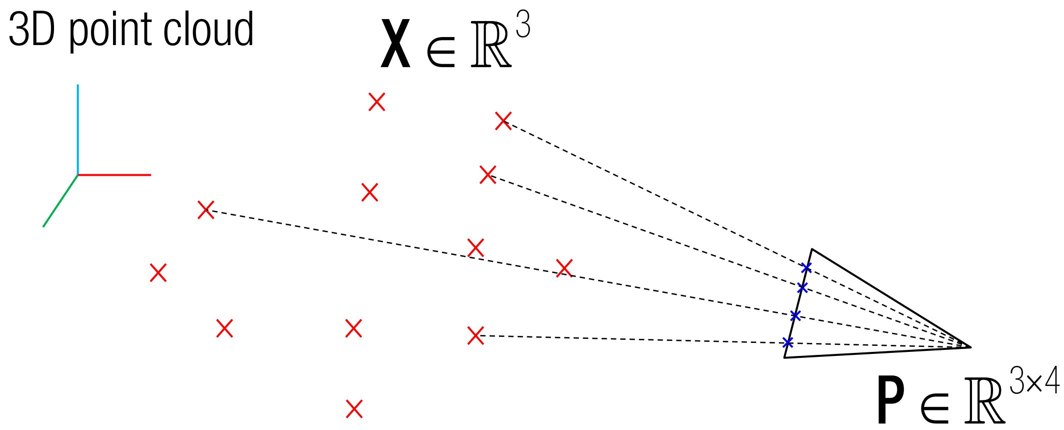


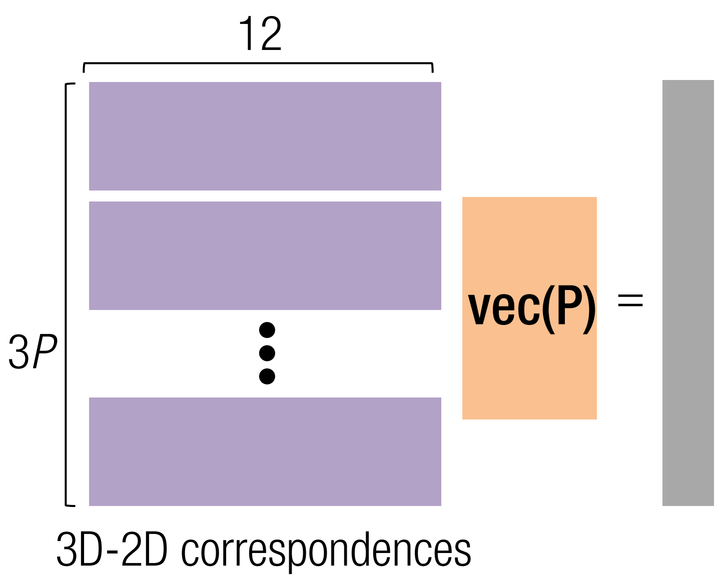


Life isn't that easy.



Example I: Perspective-n-Point

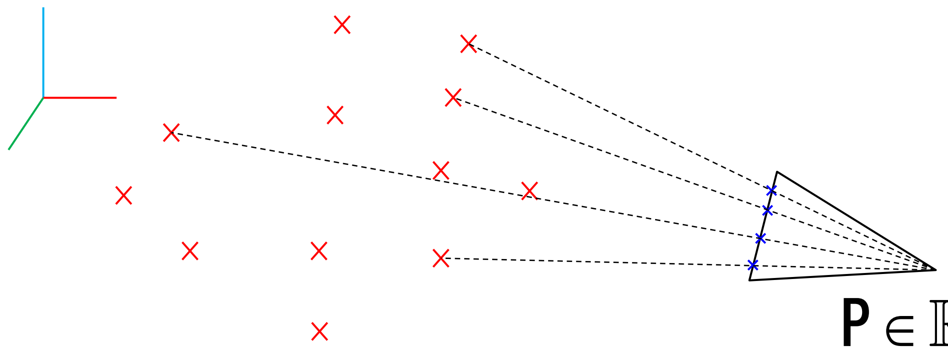




Example I: Perspective-n-Point

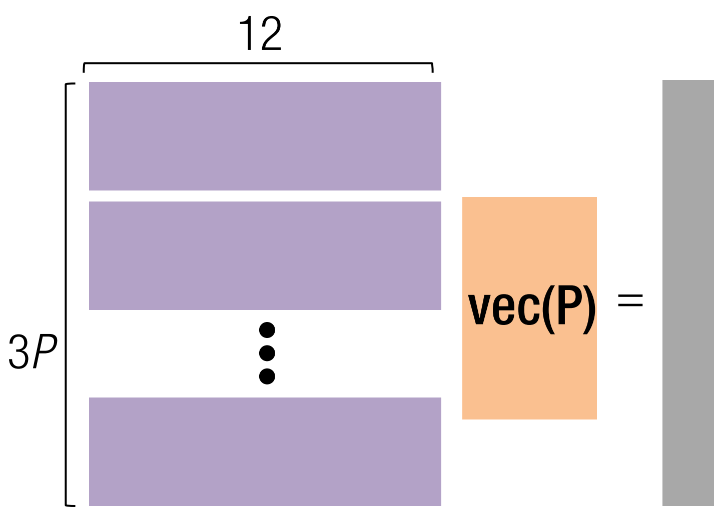
3D point cloud

$\mathbf{X} \in \mathbb{R}^3$



$$\mathbf{P} \in \mathbb{R}^{3\times4} \longrightarrow \mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix} \in \mathbb{R}^{3\times4}$$

where $\mathbf{R} \in \mathbf{SO}(3)$

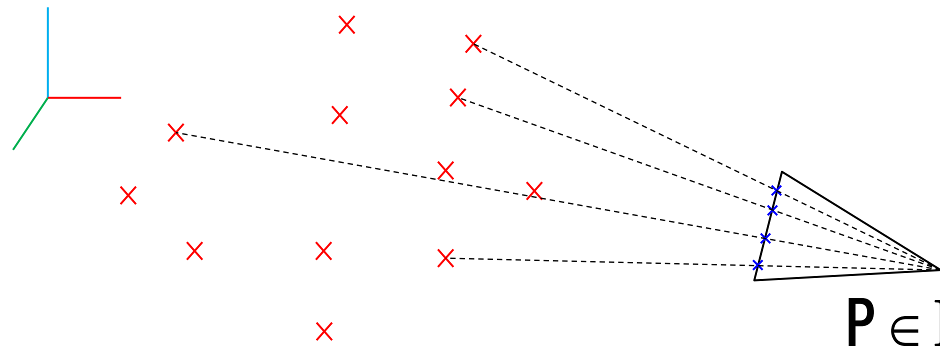


3D-2D correspondences

Example I: Perspective-n-Point

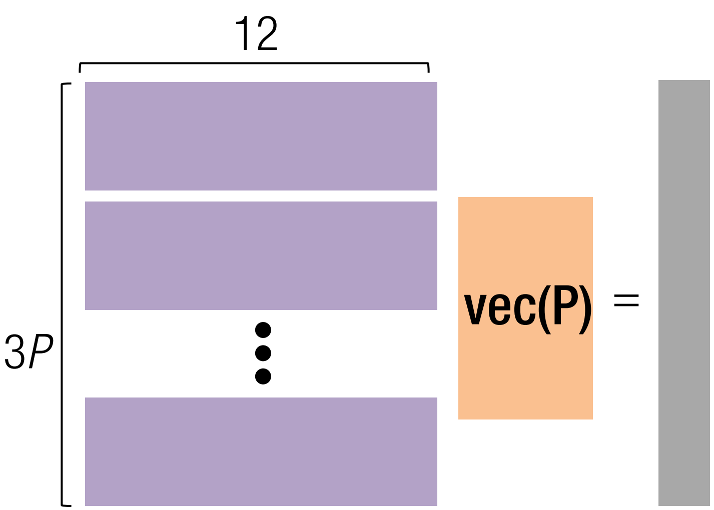
3D point cloud

$\mathbf{X} \in \mathbb{R}^3$



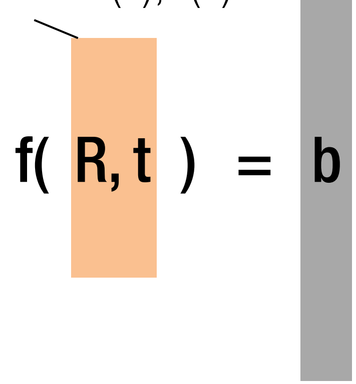
$\mathbf{P} \in \mathbb{R}^{3\times 4} \longrightarrow \mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix} \in \mathbb{R}^{3\times 4}$

where $\mathbf{R} \in \mathbf{SO}(3)$



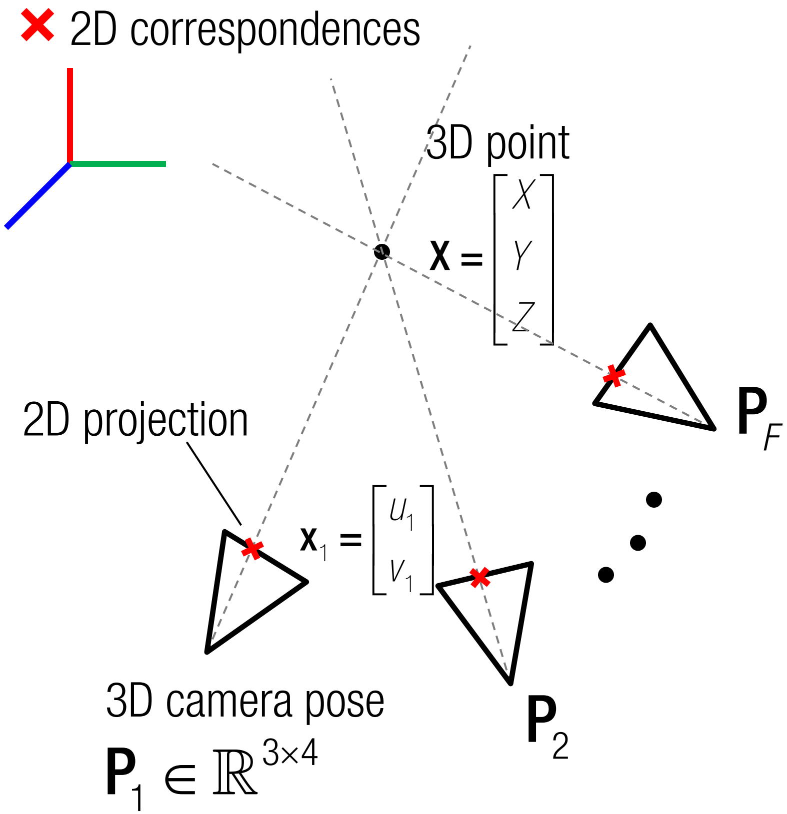
3D-2D correspondences

6 parameters: **R**(3), **t**(3)



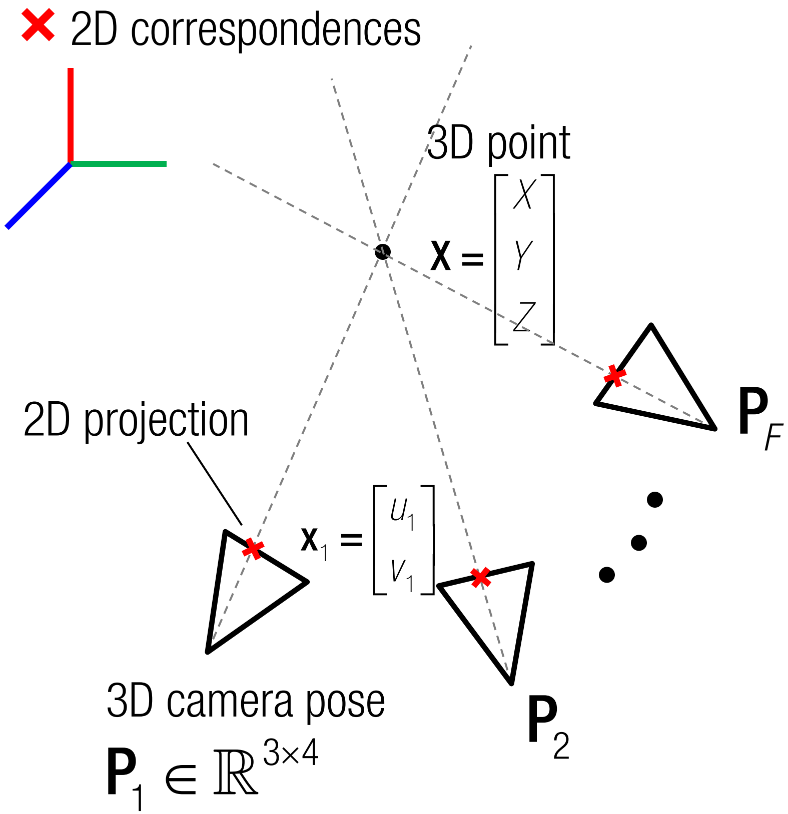
Nonlinear least squares

Example II: Triangulation



$$\begin{array}{cc} & \overbrace{\hphantom{\begin{bmatrix}\mathbf{x}_1\\1\end{bmatrix}_\times \mathbf{P}_1}}^{4} \\ 3F & \begin{bmatrix} \begin{bmatrix}\mathbf{x}_1\\1\end{bmatrix}_\times \mathbf{P}_1 \\ \begin{bmatrix}\mathbf{x}_2\\1\end{bmatrix}_\times \mathbf{P}_2 \\ \vdots \\ \begin{bmatrix}\mathbf{x}_F\\1\end{bmatrix}_\times \mathbf{P}_F \end{bmatrix} \end{array} \begin{bmatrix}\mathbf{X}\\1\end{bmatrix} = \mathbf{0}$$

Example II: Triangulation

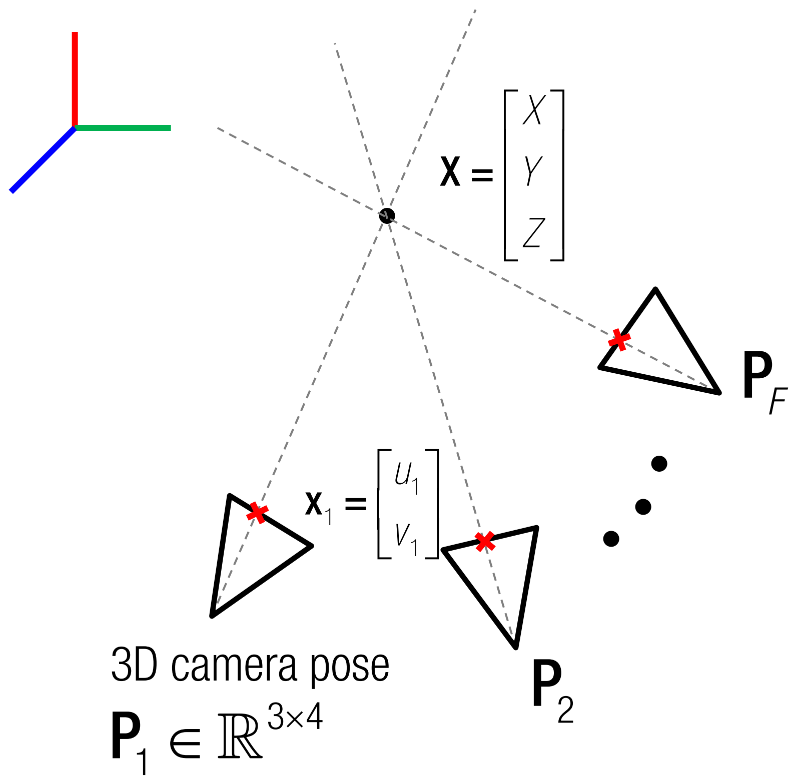


$$3F \left[\overbrace{\begin{matrix} \begin{bmatrix}\mathbf{x}_1\\1\end{bmatrix}_\times \mathbf{P}_1 \\ \begin{bmatrix}\mathbf{x}_2\\1\end{bmatrix}_\times \mathbf{P}_2 \\ \vdots \\ \begin{bmatrix}\mathbf{x}_F\\1\end{bmatrix}_\times \mathbf{P}_F \end{matrix}}^{4} \right] \begin{bmatrix}\mathbf{X}\\1\end{bmatrix} = \mathbf{0}$$

$$\min_{\mathbf{x}} \left\| \mathbf{Ax} - \mathbf{b} \right\|^2$$

Minimizes an algebraic error, i.e., there is no geometrical meaning.

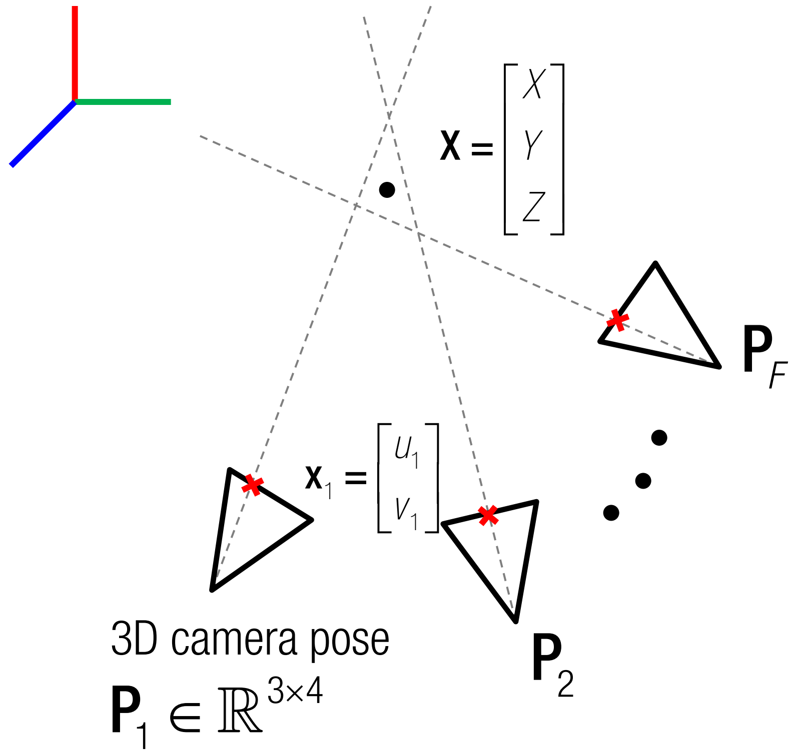
Example II: Triangulation



No noise in **x** and **P**.

✕ 2D correspondences

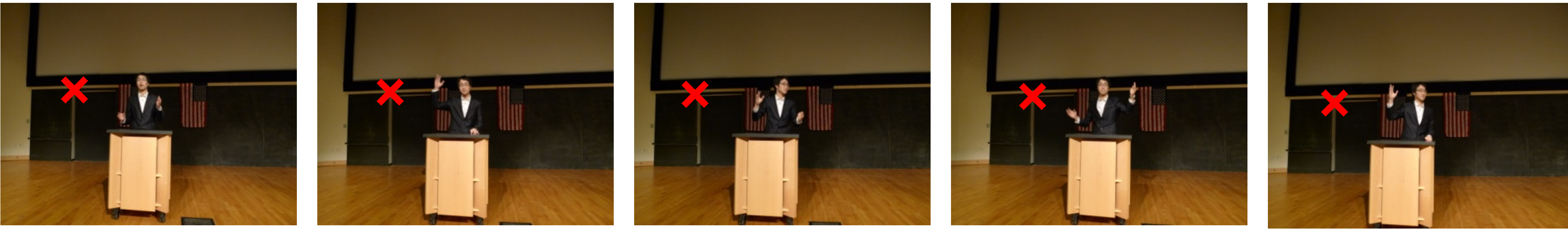
Example II: Triangulation



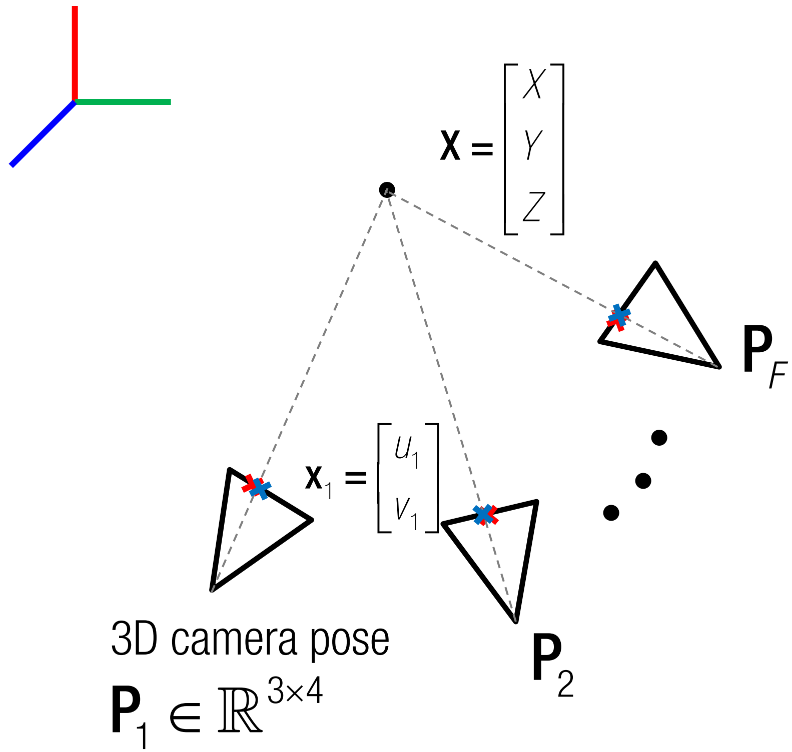
Noise in **x** and **P**.

→Rays do not meet at a 3D point.

✕ 2D correspondences



Example II: Triangulation

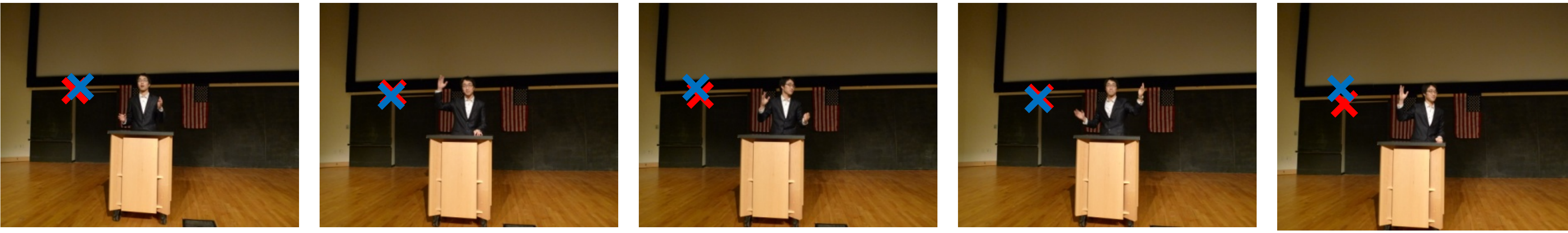


Noise in **x** and **P**.

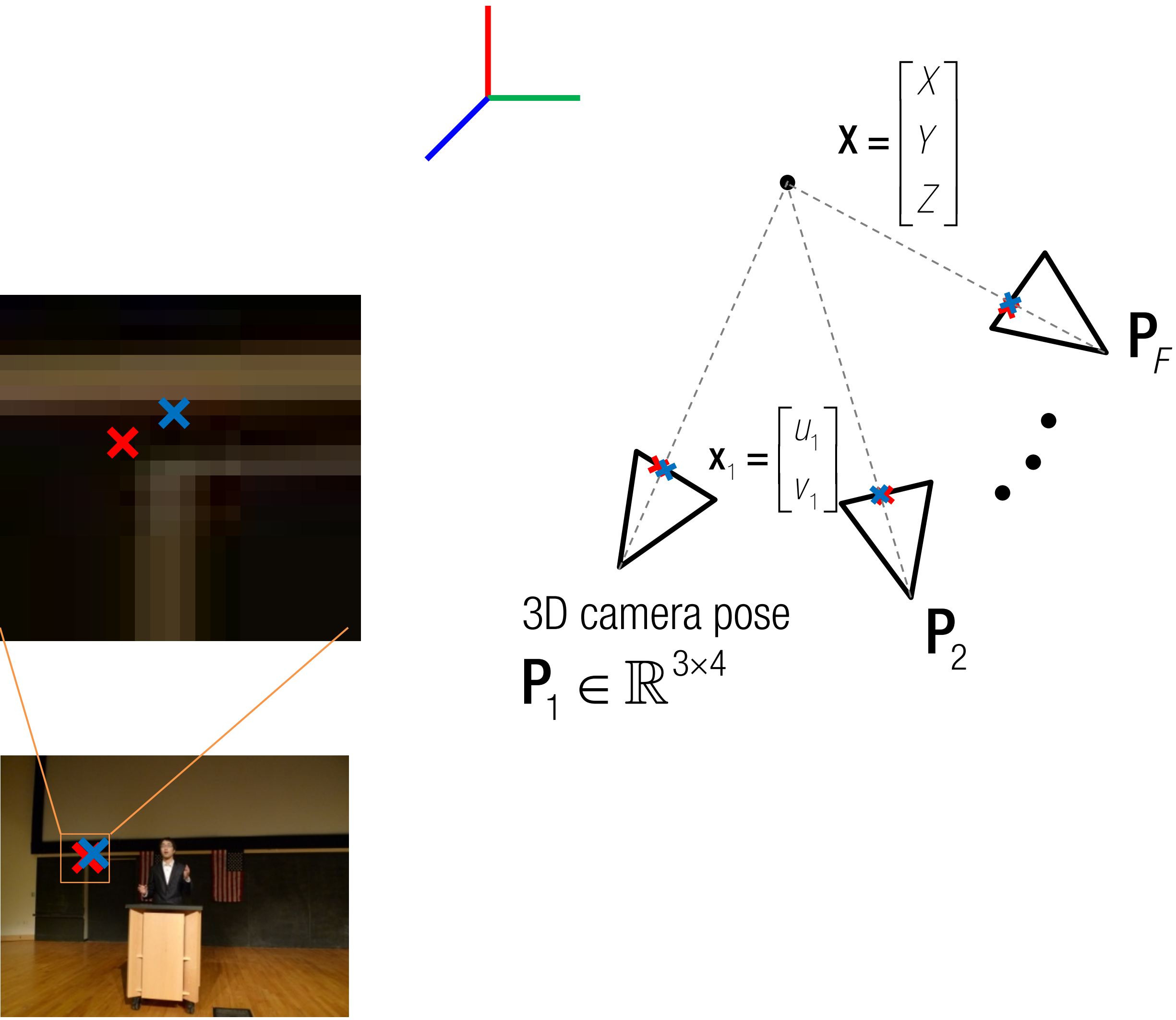
→Rays do not meet at a 3D point.

2D correspondences

Reprojection



Reprojection Error (Geometric Error)



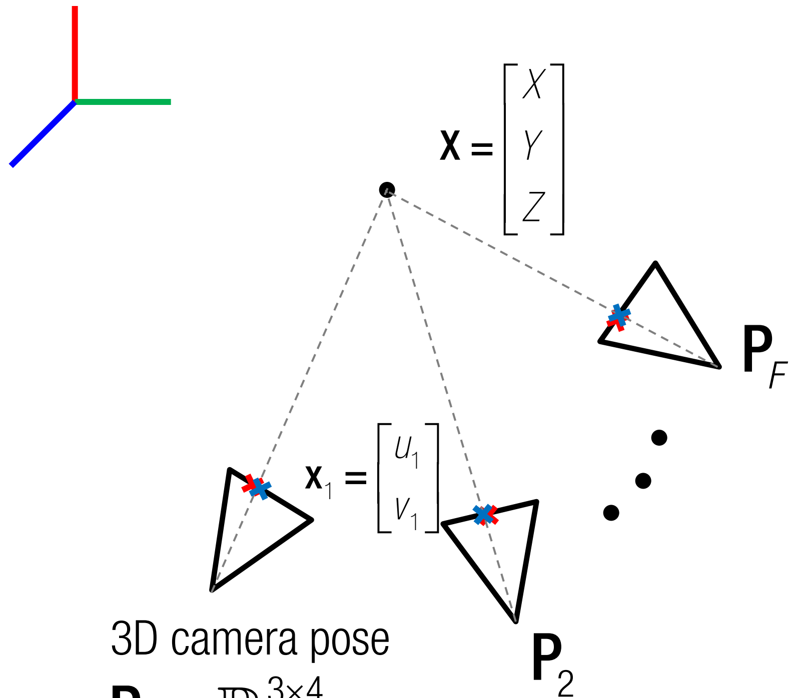
Noise in **x** and **P**.

→Rays do not meet at a 3D point.

✕ 2D correspondences

✕ Reprojection

Reprojection Error (Geometric Error)



3D camera pose

$\mathbf{P}_1 \in \mathbb{R}^{3\times 4}$

Noise in **x** and **P**.

→Rays do not meet at a 3D point.

❌ 2D correspondences

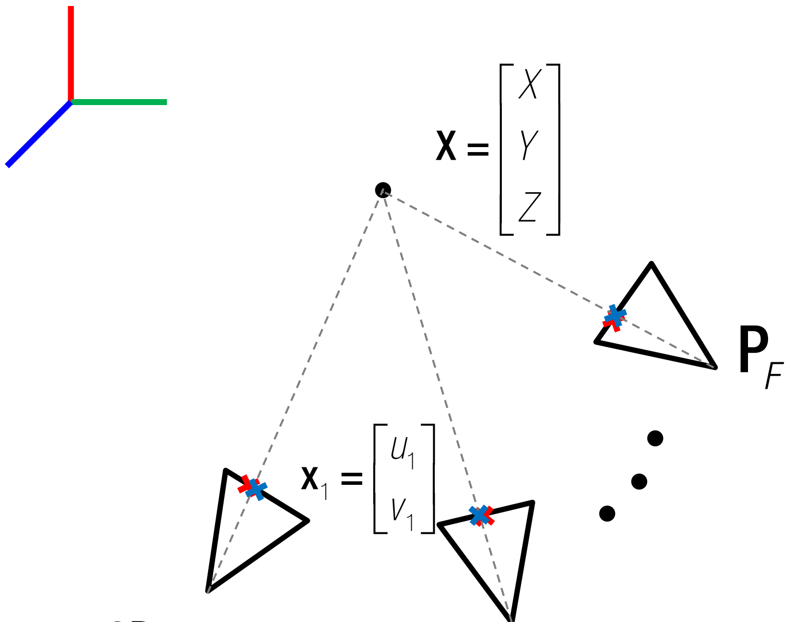
✖ Reprojection

❌ (u, v)

✖ $(u_{\text{repro}}, v_{\text{repro}}) = \left(\dfrac{\mathbf{P}^1\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}}, \dfrac{\mathbf{P}^2\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}} \right)$

where $\mathbf{P} = \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{bmatrix}$ and $\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$

Reprojection Error (Geometric Error)



3D camera pose

$\mathbf{P}_1 \in \mathbb{R}^{3\times 4}$

$$E_{\text{repro}} = \left(u - u_{\text{repro}}\right)^2 + \left(v - v_{\text{repro}}\right)^2$$

$$= \left(u - \frac{\mathbf{P}^1\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}}\right)^2 + \left(v - \frac{\mathbf{P}^2\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}}\right)^2$$

$$\min_{\mathbf{X}} \sum_{i=1}^{F} \left(u_i - \frac{\mathbf{P}_i^1\tilde{\mathbf{X}}}{\mathbf{P}_i^3\tilde{\mathbf{X}}}\right)^2 + \left(v_i - \frac{\mathbf{P}_i^2\tilde{\mathbf{X}}}{\mathbf{P}_i^3\tilde{\mathbf{X}}}\right)^2$$

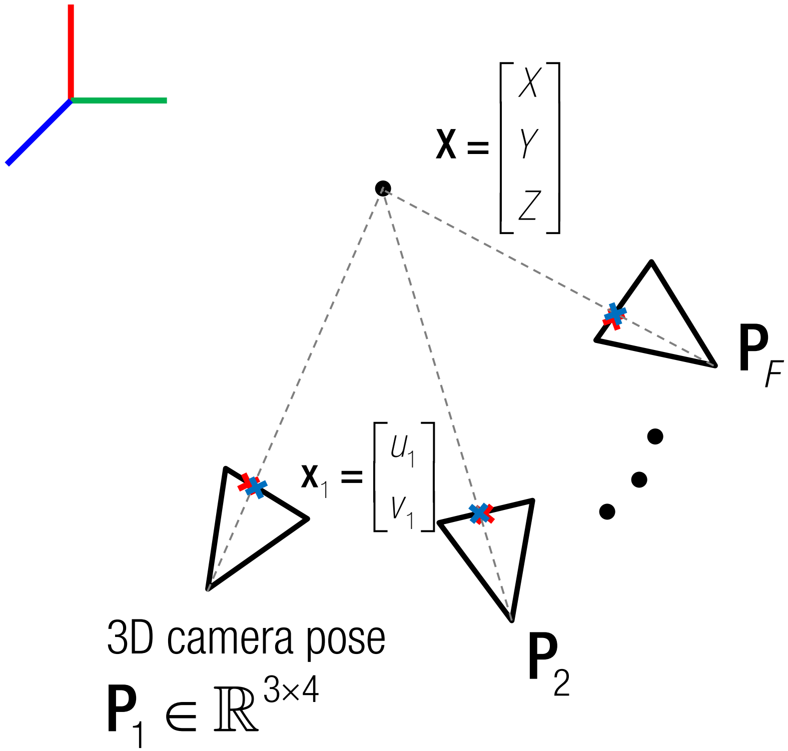
Reprojection error minimization

✖ (u, v)

✖ $\left(u_{\text{repro}}, v_{\text{repro}}\right) = \left(\frac{\mathbf{P}^1\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}}, \frac{\mathbf{P}^2\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}}\right)$

where $\mathbf{P} = \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{bmatrix}$ and $\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$

Reprojection Error (Geometric Error)



$$E_{\text{repro}} = \left(u - u_{\text{repro}}\right)^2 + \left(v - v_{\text{repro}}\right)^2$$

$$= \left(u - \frac{\mathbf{P}^1\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}}\right)^2 + \left(v - \frac{\mathbf{P}^2\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}}\right)^2$$

$$\mathbf{f}\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\right) = \begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_F \\ v_F \end{bmatrix}$$

Nonlinear least squares

✖ (u, v)

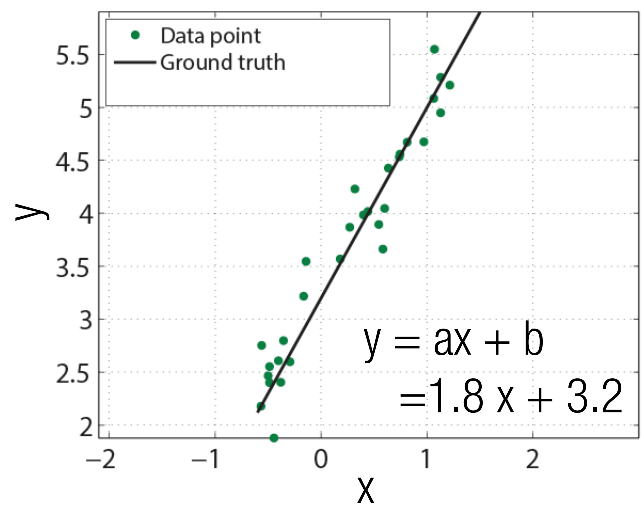
✖ $(u_{\text{repro}}, v_{\text{repro}}) = \left(\frac{\mathbf{P}^1\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}}, \frac{\mathbf{P}^2\tilde{\mathbf{X}}}{\mathbf{P}^3\tilde{\mathbf{X}}}\right)$

where $\mathbf{P} = \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{bmatrix}$ and $\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$

How to Solve? (Linear)

Linear least squares:

$$\min_{x} \|Ax - b\|^2$$



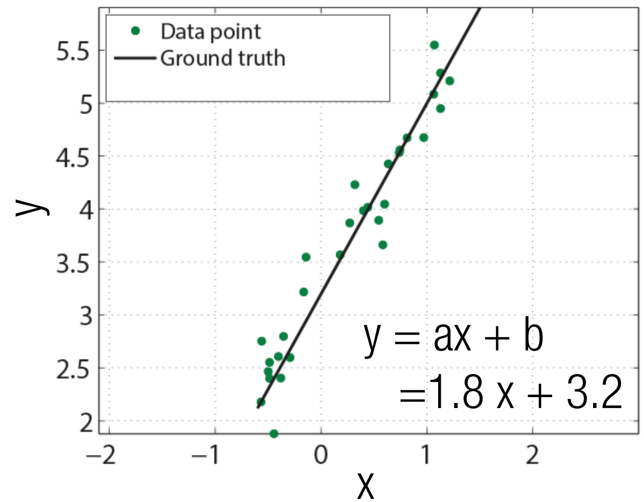
How to Solve? (Linear)

Linear least squares:

$$\min_{x} \|Ax - b\|^2 = \min_{x} (Ax - b)^\top (Ax - b)$$

$$= \min_{x} x^\top A^\top Ax - 2b^\top Ax - b^\top b$$

$$= \min_{x} x^\top A^\top Ax - 2b^\top Ax$$



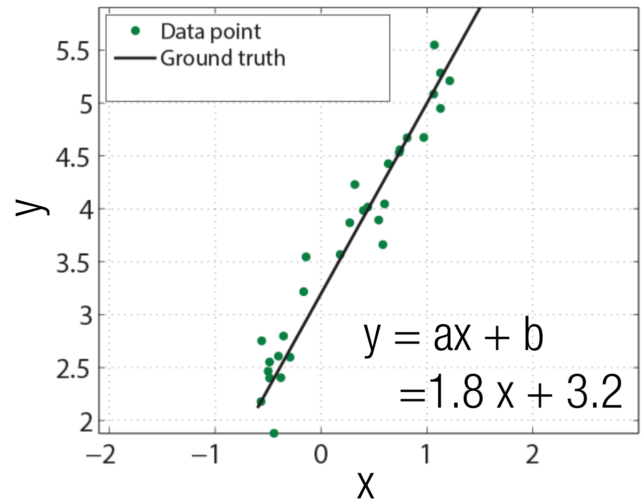
How to Solve? (Linear)

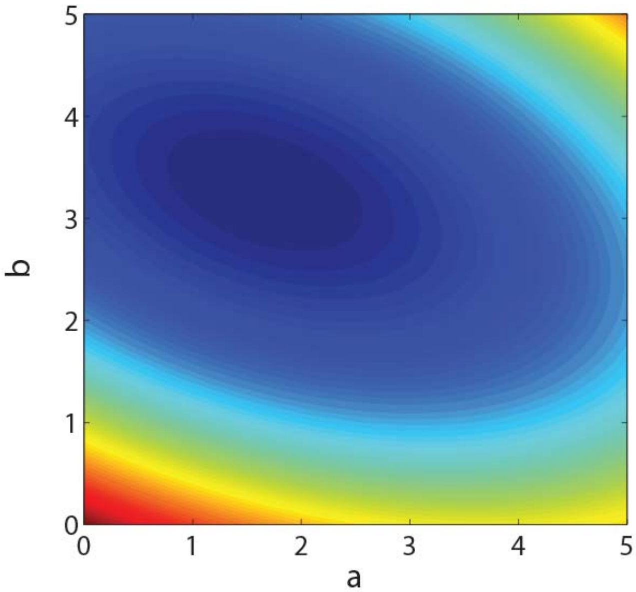
Linear least squares:

$$\min_{x} \|Ax - b\|^2 = \min_{x} (Ax - b)^{\top} (Ax - b)$$
$$= \min_{x} x^{\top} A^{\top} A x - 2b^{\top} A x - b^{\top} b$$
$$= \min_{x} x^{\top} A^{\top} A x - 2b^{\top} A x$$

Error:

$$E = x^{\top} A^{\top} A x - 2b^{\top} A x$$





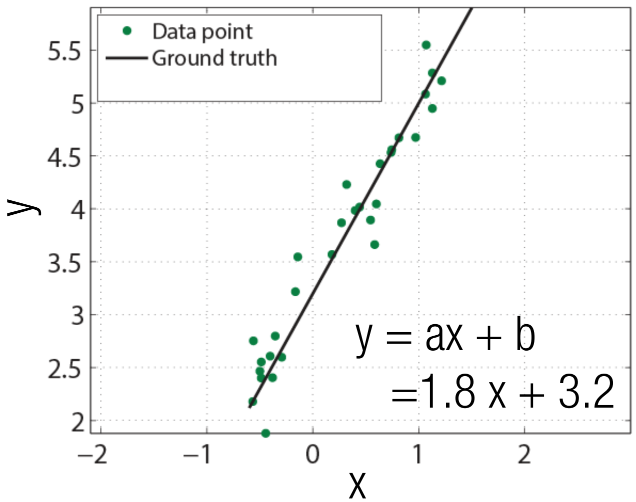
How to Solve? (Linear)

Linear least squares:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 = \min_{\mathbf{x}} (\mathbf{A}\mathbf{x}-\mathbf{b})^\top (\mathbf{A}\mathbf{x}-\mathbf{b})$$

$$= \min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} - 2\mathbf{b}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{b}$$

$$= \min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} - 2\mathbf{b}^\top \mathbf{A}\mathbf{x}$$

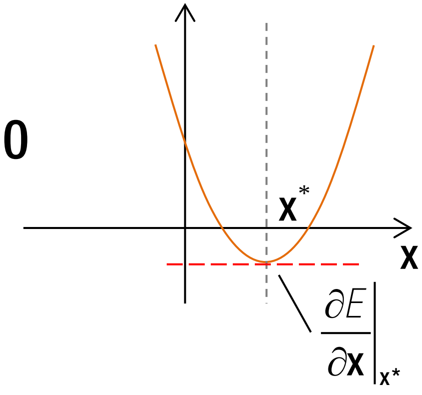


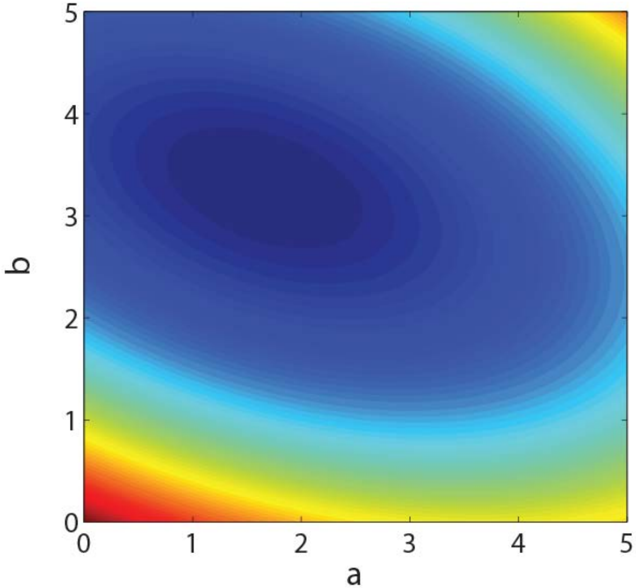
Error:

$$E = \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} - 2\mathbf{b}^\top \mathbf{A}\mathbf{x}$$

Condition for the solution:

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\mathbf{A}^\top \mathbf{A}\mathbf{x} - 2\mathbf{A}^\top \mathbf{b} = 0$$

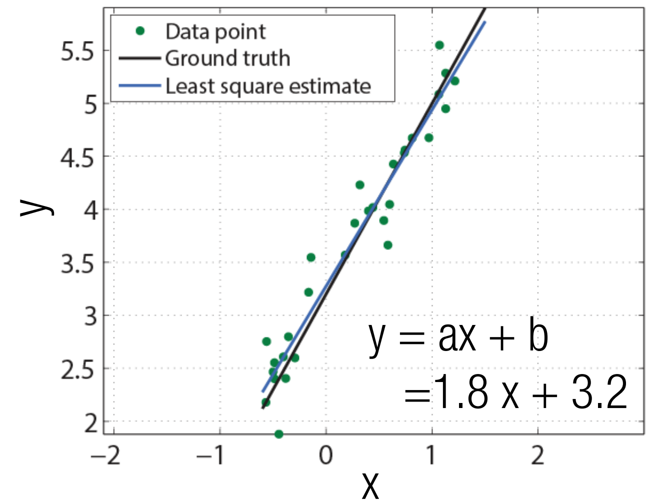




How to Solve? (Linear)

Linear least squares:

$$\min_{\mathbf{x}} \|\mathbf{Ax}-\mathbf{b}\|^2 = \min_{\mathbf{x}} (\mathbf{Ax}-\mathbf{b})^T(\mathbf{Ax}-\mathbf{b})$$
$$= \min_{\mathbf{x}} \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - 2\mathbf{b}^T\mathbf{Ax} - \mathbf{b}^T\mathbf{b}$$
$$= \min_{\mathbf{x}} \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - 2\mathbf{b}^T\mathbf{Ax}$$

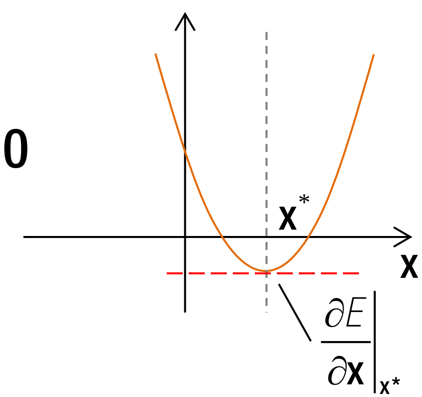


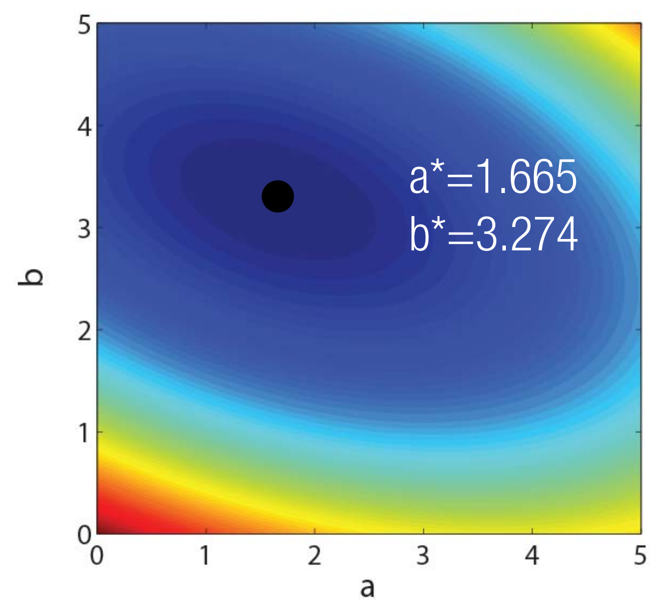
Error:

$$E = \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - 2\mathbf{b}^T\mathbf{Ax}$$

Condition for the solution:

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\mathbf{A}^T\mathbf{Ax} - 2\mathbf{A}^T\mathbf{b} = 0$$





Solution:

$$\mathbf{x}^* = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$$

How to Solve? (Nonlinear)

Nonlinear least squares:

$$\min_x \left\| f(x) - b \right\|^2 = \min_x \left(f(x) - b \right)^\top \left(f(x) - b \right)$$

$$= \min_x f(x)^\top f(x) - 2b^\top f(x) - b^\top b$$

$$= \min_x f(x)^\top f(x) - 2b^\top f(x)$$

How to Solve? (Nonlinear)

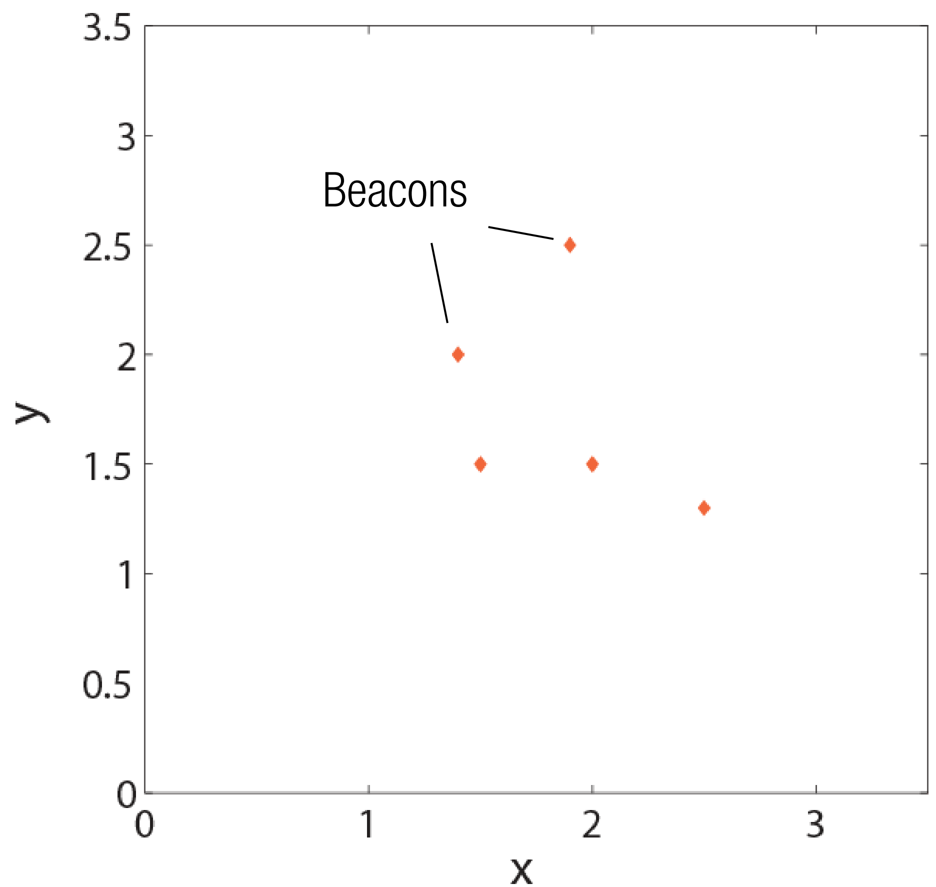
Nonlinear least squares:

$$\min_x \|f(x) - b\|^2 = \min_x (f(x) - b)^T (f(x) - b)$$

$$= \min_x f(x)^T f(x) - 2b^T f(x) - b^T b$$

$$= \min_x f(x)^T f(x) - 2b^T f(x)$$

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

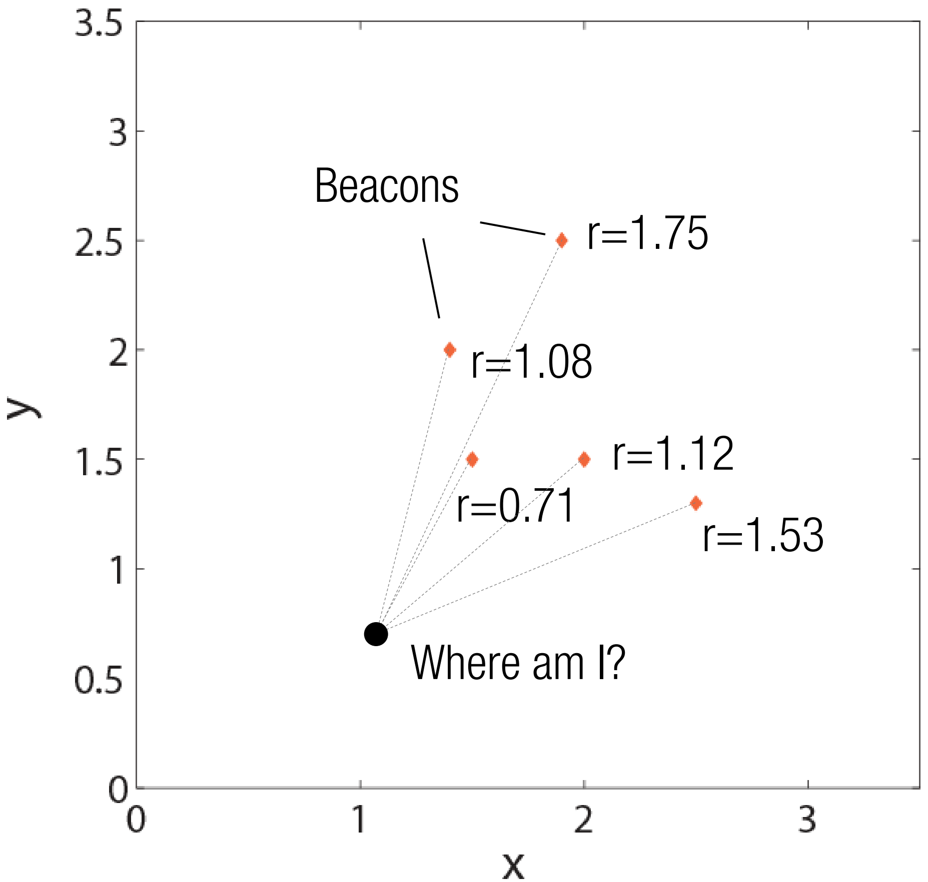
Nonlinear least squares:

$$\min_x \|f(x) - b\|^2 = \min_x \left(f(x) - b\right)^T \left(f(x) - b\right)$$

$$= \min_x f(x)^T f(x) - 2b^T f(x) - b^T b$$

$$= \min_x f(x)^T f(x) - 2b^T f(x)$$

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

Nonlinear least squares:

$$\min_{\mathbf{x}} \|\mathbf{f(x)} - \mathbf{b}\|^2 = \min_{\mathbf{x}} \left(\mathbf{f(x)} - \mathbf{b}\right)^\top \left(\mathbf{f(x)} - \mathbf{b}\right)$$

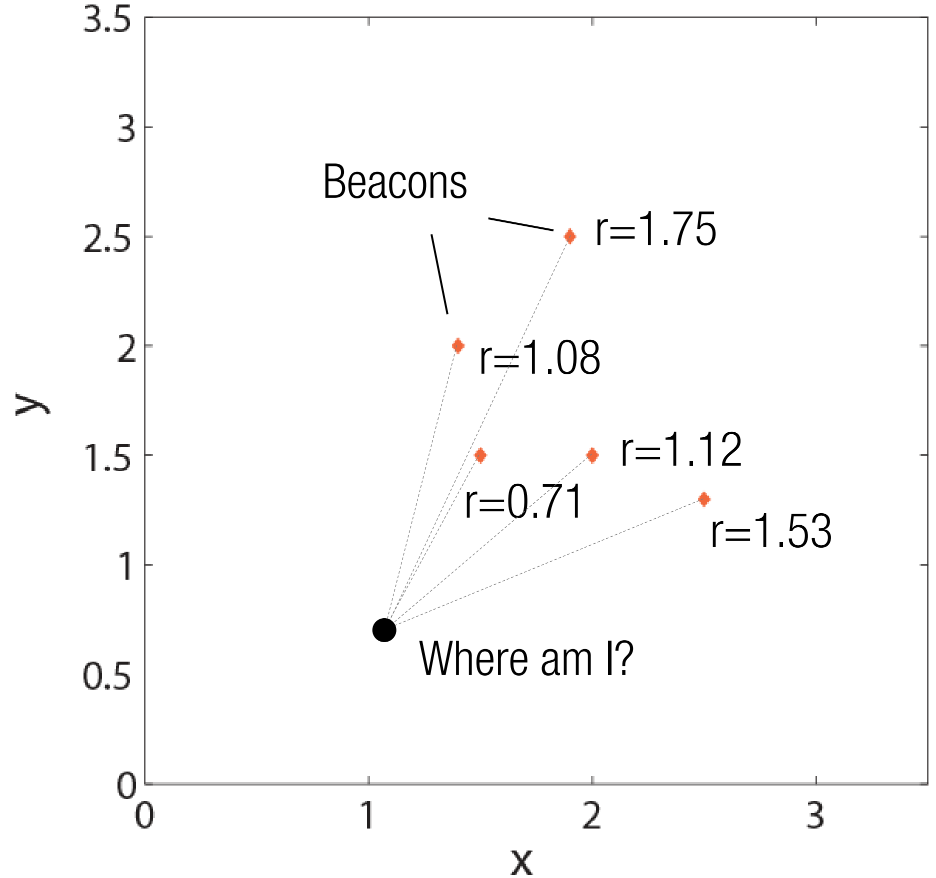
$$= \min_{\mathbf{x}} \mathbf{f(x)}^\top \mathbf{f(x)} - 2\mathbf{b}^\top \mathbf{f(x)} - \mathbf{b}^\top \mathbf{b}$$

$$= \min_{\mathbf{x}} \mathbf{f(x)}^\top \mathbf{f(x)} - 2\mathbf{b}^\top \mathbf{f(x)}$$

Error:

$$E = \mathbf{f(x)}^\top \mathbf{f(x)} - 2\mathbf{b}^\top \mathbf{f(x)}$$

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

Nonlinear least squares:

$$\min_{\mathbf{x}} \left\| \mathbf{f}(\mathbf{x}) - \mathbf{b} \right\|^2 = \min_{\mathbf{x}} \left(\mathbf{f}(\mathbf{x}) - \mathbf{b} \right)^{\mathsf{T}} \left(\mathbf{f}(\mathbf{x}) - \mathbf{b} \right)$$

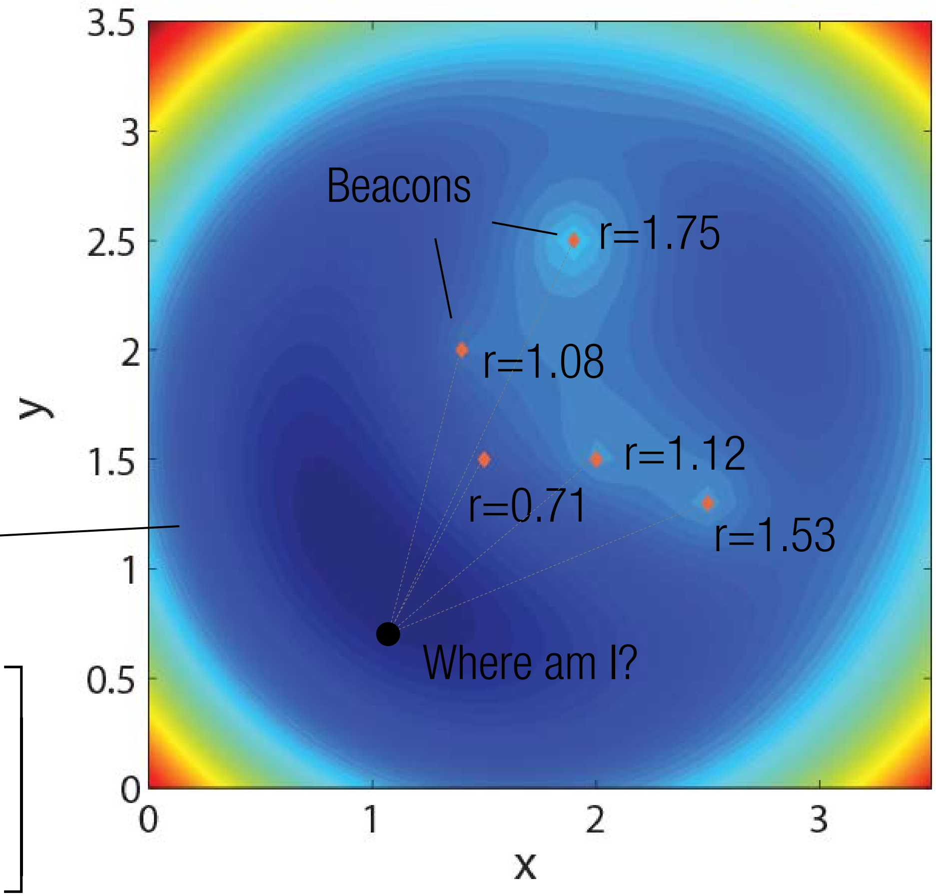
$$= \min_{\mathbf{x}} \mathbf{f}(\mathbf{x})^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\mathbf{b}^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - \mathbf{b}^{\mathsf{T}} \mathbf{b}$$

$$= \min_{\mathbf{x}} \mathbf{f}(\mathbf{x})^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\mathbf{b}^{\mathsf{T}} \mathbf{f}(\mathbf{x})$$

Error:

$$E = \mathbf{f}(\mathbf{x})^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\mathbf{b}^{\mathsf{T}} \mathbf{f}(\mathbf{x})$$

Example:
Localization using range data from beacons



$$E = \mathbf{S}^{\mathsf{T}} \mathbf{S}$$

where $$\mathbf{S} = \begin{bmatrix} \sqrt{(u_1 - x)^2 + (v_1 - y)^2} \\ \vdots \\ \sqrt{(u_5 - x)^2 + (v_5 - y)^2} \end{bmatrix} - \begin{bmatrix} r_1 \\ \vdots \\ r_5 \end{bmatrix}$$

How to Solve? (Nonlinear)

Nonlinear least squares:

$$\min_{\mathbf{x}} \|\mathbf{f}(\mathbf{x}) - \mathbf{b}\|^2 = \min_{\mathbf{x}} (\mathbf{f}(\mathbf{x}) - \mathbf{b})^{\mathsf{T}} (\mathbf{f}(\mathbf{x}) - \mathbf{b})$$

$$= \min_{\mathbf{x}} \mathbf{f}(\mathbf{x})^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\mathbf{b}^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - \mathbf{b}^{\mathsf{T}} \mathbf{b}$$

$$= \min_{\mathbf{x}} \mathbf{f}(\mathbf{x})^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\mathbf{b}^{\mathsf{T}} \mathbf{f}(\mathbf{x})$$

Error:

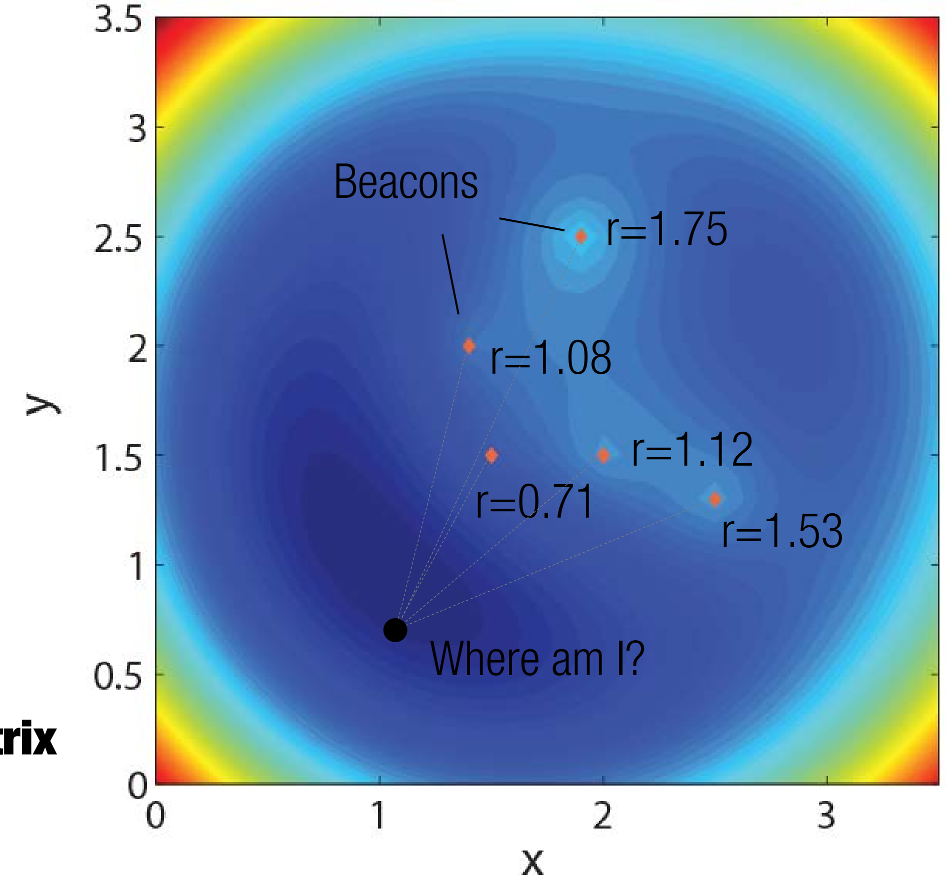
$$E = \mathbf{f}(\mathbf{x})^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\mathbf{b}^{\mathsf{T}} \mathbf{f}(\mathbf{x})$$

Condition for the solution:

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{b} = 0$$

where $\dfrac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_n} \\ \vdots & & \vdots \\ \frac{\partial \mathbf{f}_m}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_m}{\partial \mathbf{x}_n} \end{bmatrix}$ **: Jacobian matrix**

Example:
Localization using range data from beacons

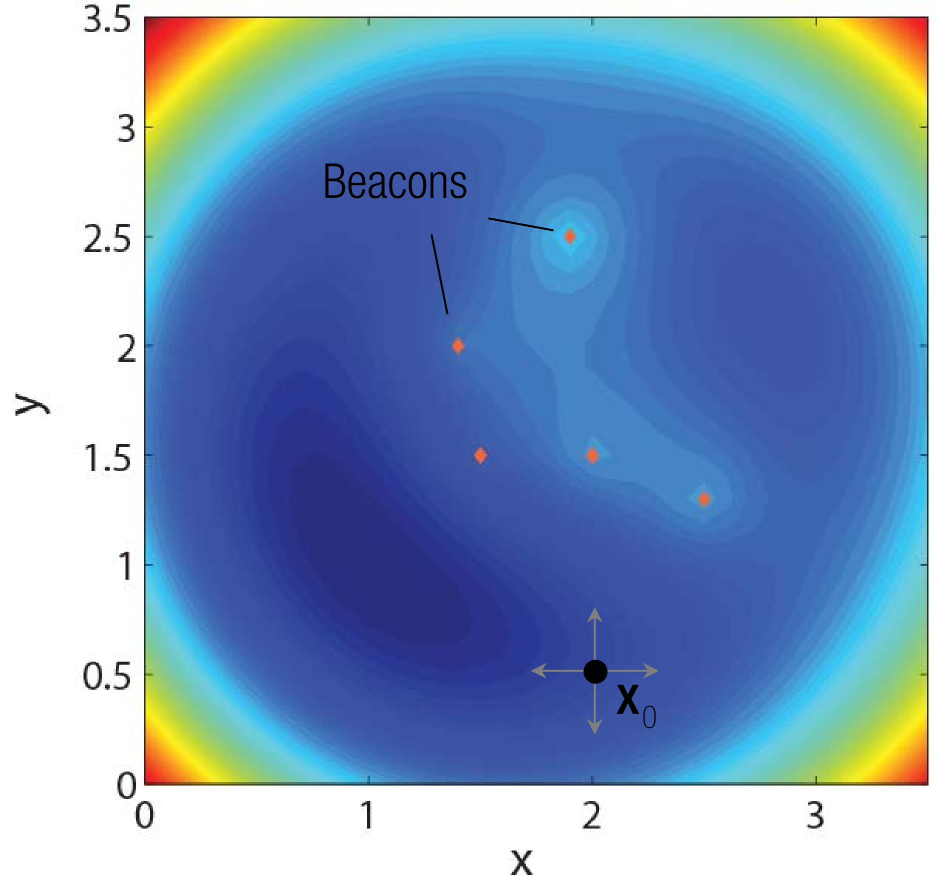


How to Solve? (Nonlinear)

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathrm{T}} \mathbf{f}(\mathbf{x}) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathrm{T}} \mathbf{b} = \mathbf{0} \qquad (1)$$

Objective:
Given $\mathbf{x}_0$, move $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}_0)$.

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{b} = \mathbf{0} \qquad (1)$$

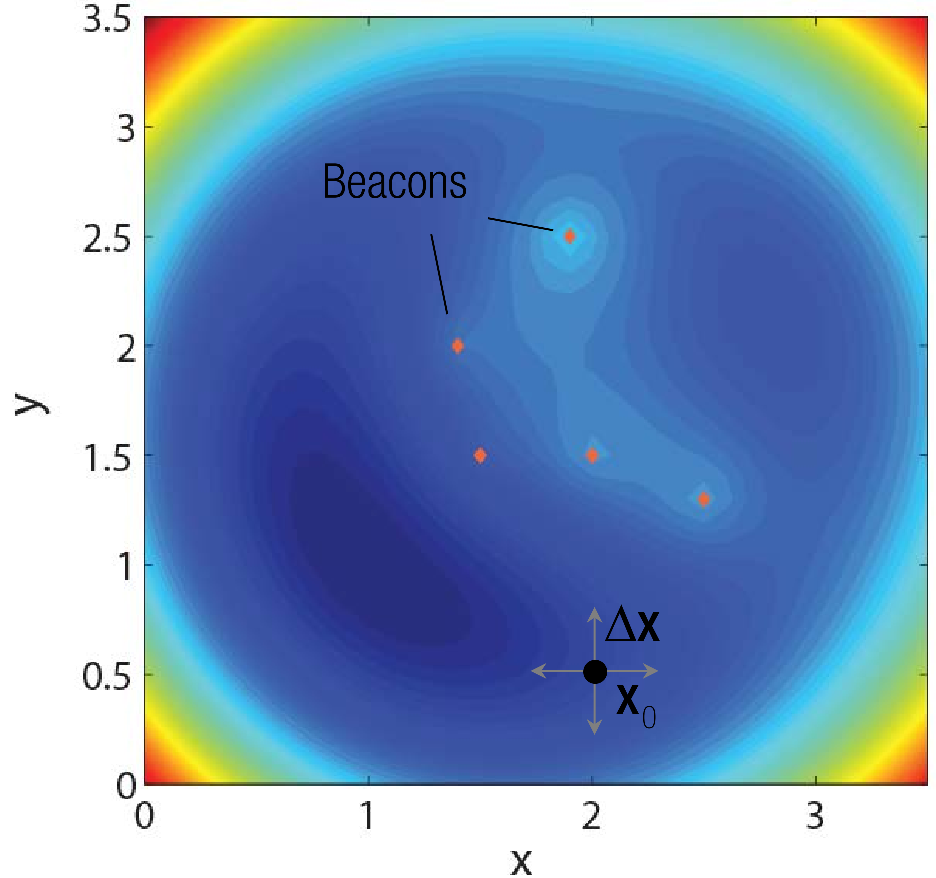
Objective:

Given $\mathbf{x}_0$, move $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}_0)$.

Taylor expansion around $\mathbf{x}$:

$$\mathbf{f}(\mathbf{x}+\Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\Delta\mathbf{x}$$

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{b} = \mathbf{0} \qquad (1)$$

Objective:

Given $\mathbf{x}_0$, move $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}_0)$.

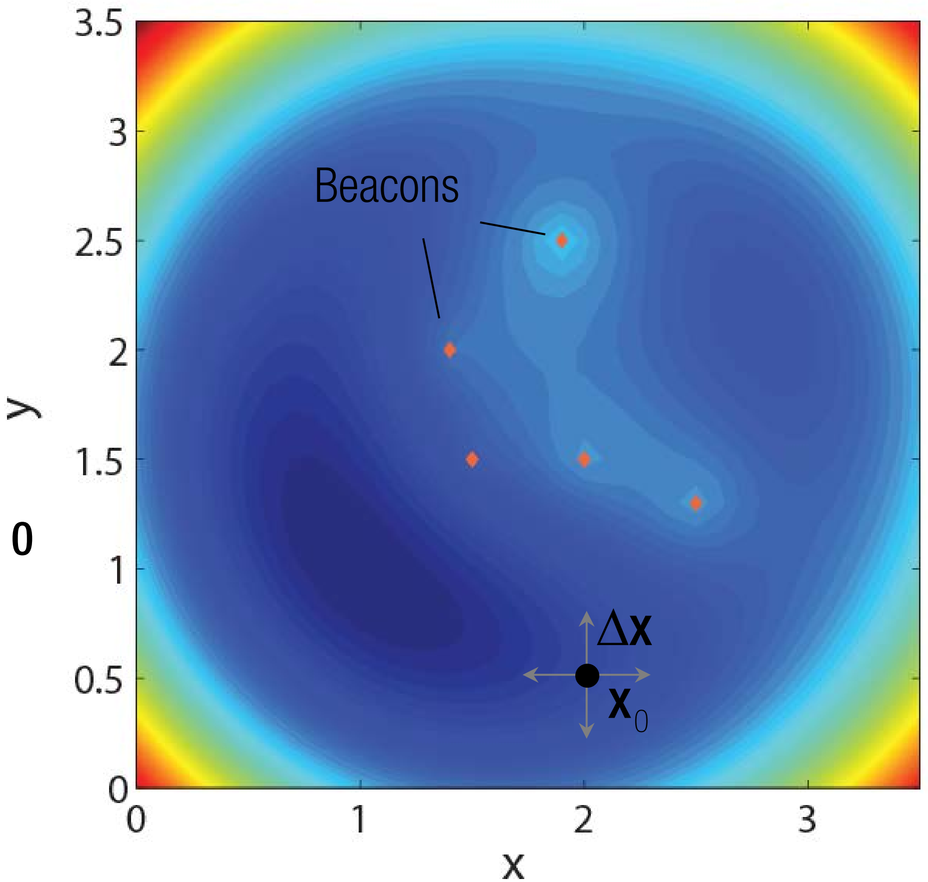
Taylor expansion around $\mathbf{x}$:

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \Delta\mathbf{x}$$

Plugging into Equation (1):

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \left(\mathbf{f}(\mathbf{x}) + \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \Delta\mathbf{x} \right) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{b} = \mathbf{0}$$

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\top} \mathbf{f}(\mathbf{x}) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\top} \mathbf{b} = \mathbf{0} \qquad (1)$$

Objective:

Given $\mathbf{x}_0$, move $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}_0)$.

Taylor expansion around $\mathbf{x}$:

$$\mathbf{f}(\mathbf{x}+\Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\Delta\mathbf{x}$$

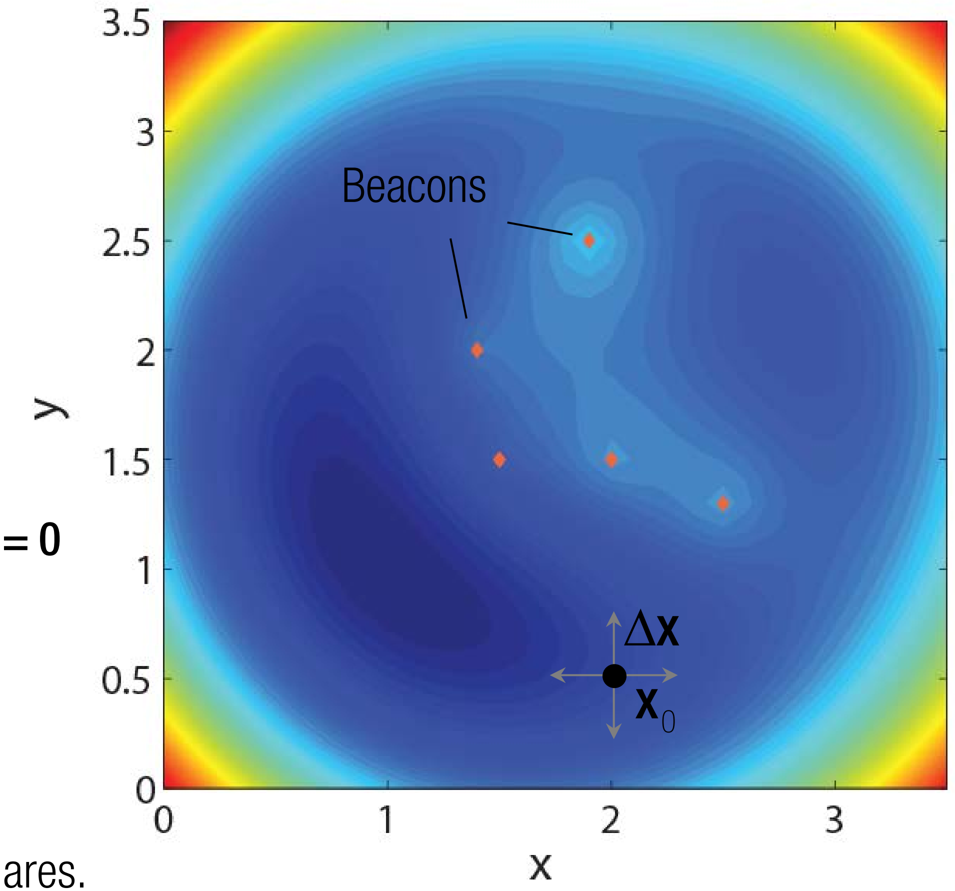
Plugging into Equation (1):

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\top}\left(\mathbf{f}(\mathbf{x}) + \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\Delta\mathbf{x}\right) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\top} \mathbf{b} = \mathbf{0}$$

$$\longrightarrow \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\top}\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\Delta\mathbf{x} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\top}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

Note that $\mathbf{A}^{\top}\mathbf{A}\mathbf{x} = \mathbf{A}^{\top}\mathbf{b}$ for linear least squares.

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}}\mathbf{f}(\mathbf{x}) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}}\mathbf{b} = \mathbf{0} \qquad (1)$$

Objective:
Given $\mathbf{x}_0$, move $\mathbf{x}$ such that $\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}_0)$.

Taylor expansion around $\mathbf{x}$:

$$\mathbf{f}(\mathbf{x}+\Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\Delta\mathbf{x}$$

Plugging into Equation (1):

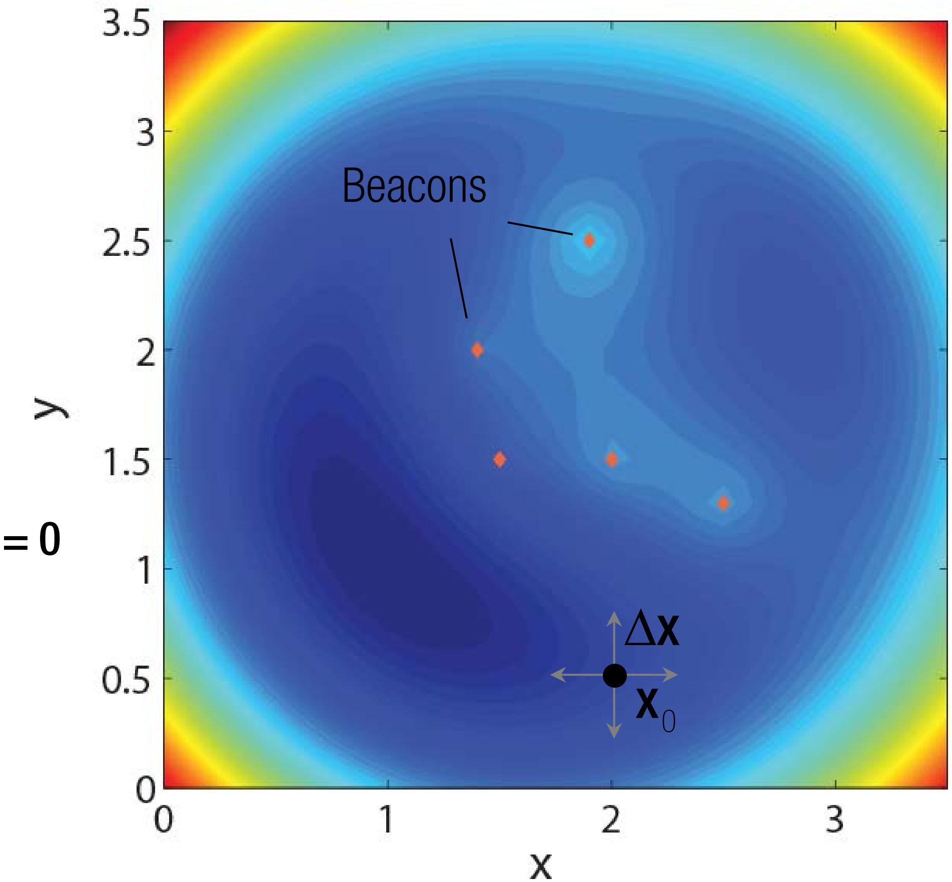
$$\left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}}\left(\mathbf{f}(\mathbf{x}) + \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\Delta\mathbf{x}\right) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}}\mathbf{b} = \mathbf{0}$$

$$\longrightarrow \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}}\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\Delta\mathbf{x} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}}$$

where $\Delta\mathbf{x} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}}\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{-1}\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$

Example:
Localization using range data from beacons

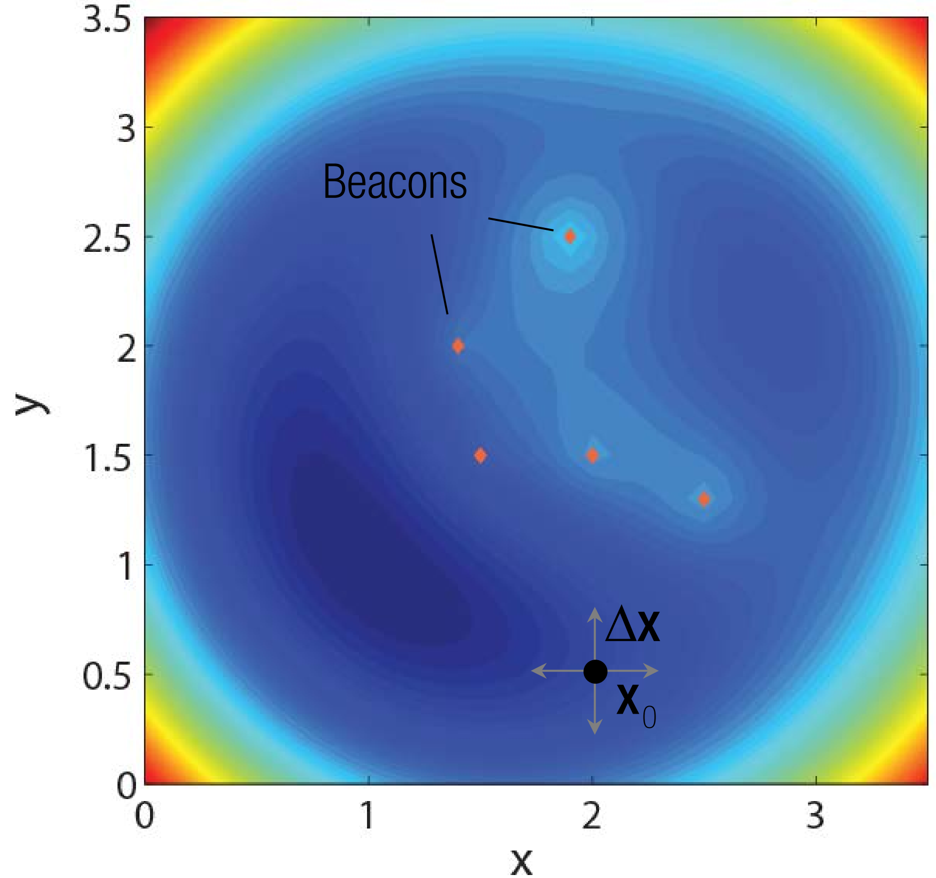


How to Solve? (Nonlinear)

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}}$$

$$\text{where } \Delta\mathbf{x} = \left(\frac{\partial\mathbf{f}(\mathbf{x})}{\partial\mathbf{x}}^{\mathrm{T}}\frac{\partial\mathbf{f}(\mathbf{x})}{\partial\mathbf{x}}\right)^{-1}\frac{\partial\mathbf{f}(\mathbf{x})}{\partial\mathbf{x}}^{\mathrm{T}}\left(\mathbf{b}-\mathbf{f}(\mathbf{x})\right)$$

Example:
Localization using range data from beacons



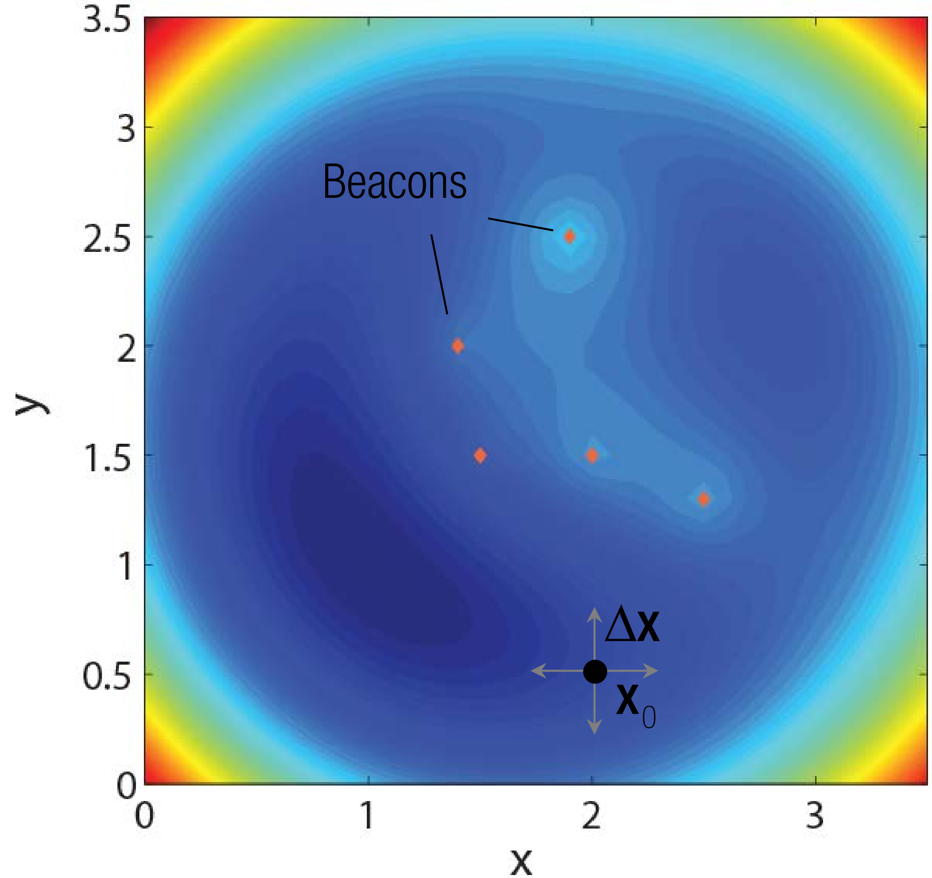
How to Solve? (Nonlinear)

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}}$$

where $\Delta\mathbf{x} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathrm{T}} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathrm{T}} \left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$

$$\mathbf{f}(x, y) = \begin{bmatrix} \sqrt{(u_1 - x)^2 + (v_1 - y)^2} \\ \vdots \\ \sqrt{(u_5 - x)^2 + (v_5 - y)^2} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} r_1 \\ \vdots \\ r_5 \end{bmatrix}$$

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

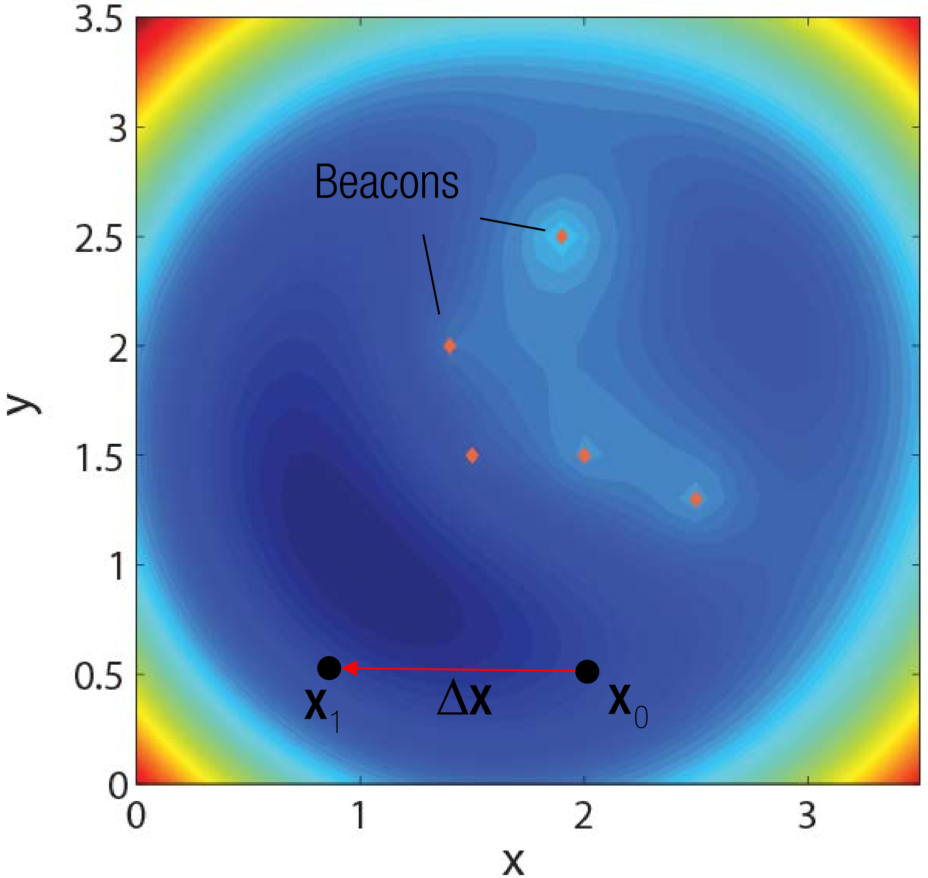
$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}}$$

where $\Delta\mathbf{x} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathrm{T}} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathrm{T}} \left(\mathbf{b} - \mathbf{f}(\mathbf{x}) \right)$

$$\mathbf{f}(x, y) = \begin{bmatrix} \sqrt{(u_1 - x)^2 + (v_1 - y)^2} \\ \vdots \\ \sqrt{(u_5 - x)^2 + (v_5 - y)^2} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} r_1 \\ \vdots \\ r_5 \end{bmatrix}$$

$$\underbrace{\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{u_1 - x}{\sqrt{(u_1 - x)^2 + (v_1 - y)^2}} & \frac{v_1 - y}{\sqrt{(u_1 - x)^2 + (v_1 - y)^2}} \\ \\ \frac{u_5 - x}{\sqrt{(u_5 - x)^2 + (v_5 - y)^2}} & \frac{v_5 - y}{\sqrt{(u_5 - x)^2 + (v_5 - y)^2}} \end{bmatrix}}_{\text{Jacobian matrix}}$$

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

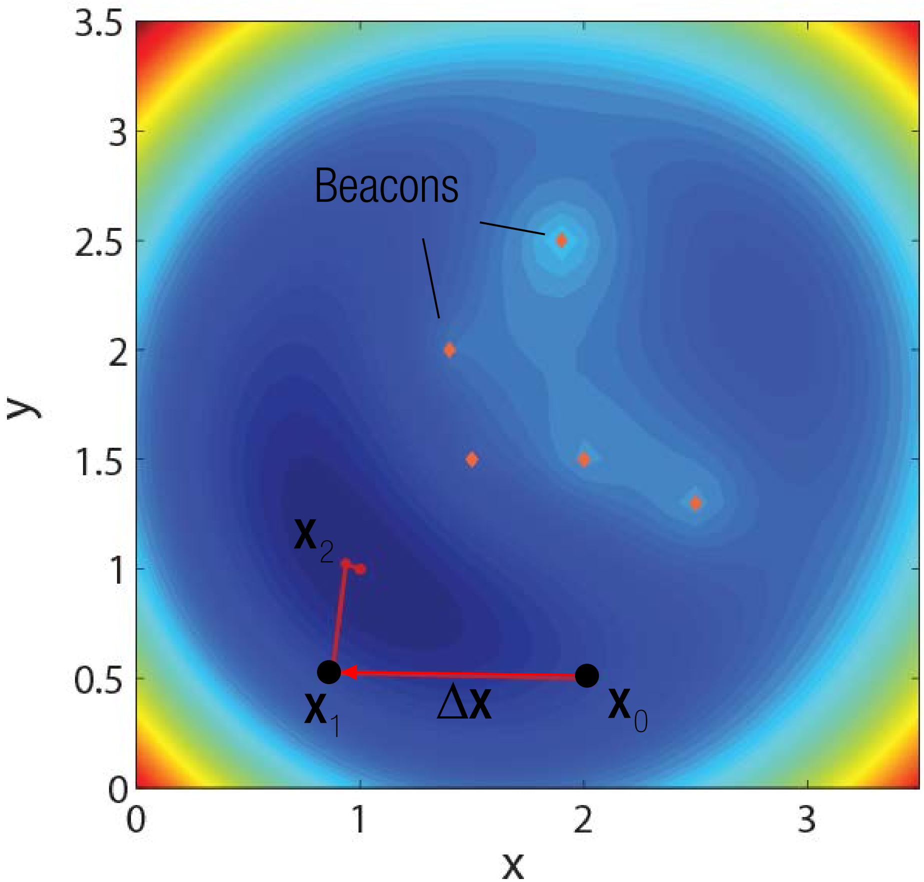
$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}}$$

$$\text{where } \Delta\mathbf{x} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^T \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^T (\mathbf{b} - \mathbf{f}(\mathbf{x}))$$

$$\mathbf{f}(x,y) = \begin{bmatrix} \sqrt{(u_1 - x)^2 + (v_1 - y)^2} \\ \vdots \\ \sqrt{(u_5 - x)^2 + (v_5 - y)^2} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} r_1 \\ \vdots \\ r_5 \end{bmatrix}$$

$$\underbrace{\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{u_1 - x}{\sqrt{(u_1 - x)^2 + (v_1 - y)^2}} & \frac{v_1 - y}{\sqrt{(u_1 - x)^2 + (v_1 - y)^2}} \\ \frac{u_5 - x}{\sqrt{(u_5 - x)^2 + (v_5 - y)^2}} & \frac{v_5 - y}{\sqrt{(u_5 - x)^2 + (v_5 - y)^2}} \end{bmatrix}}_{\text{Jacobian matrix}}$$

Example:
Localization using range data from beacons



How to Solve? (Nonlinear)

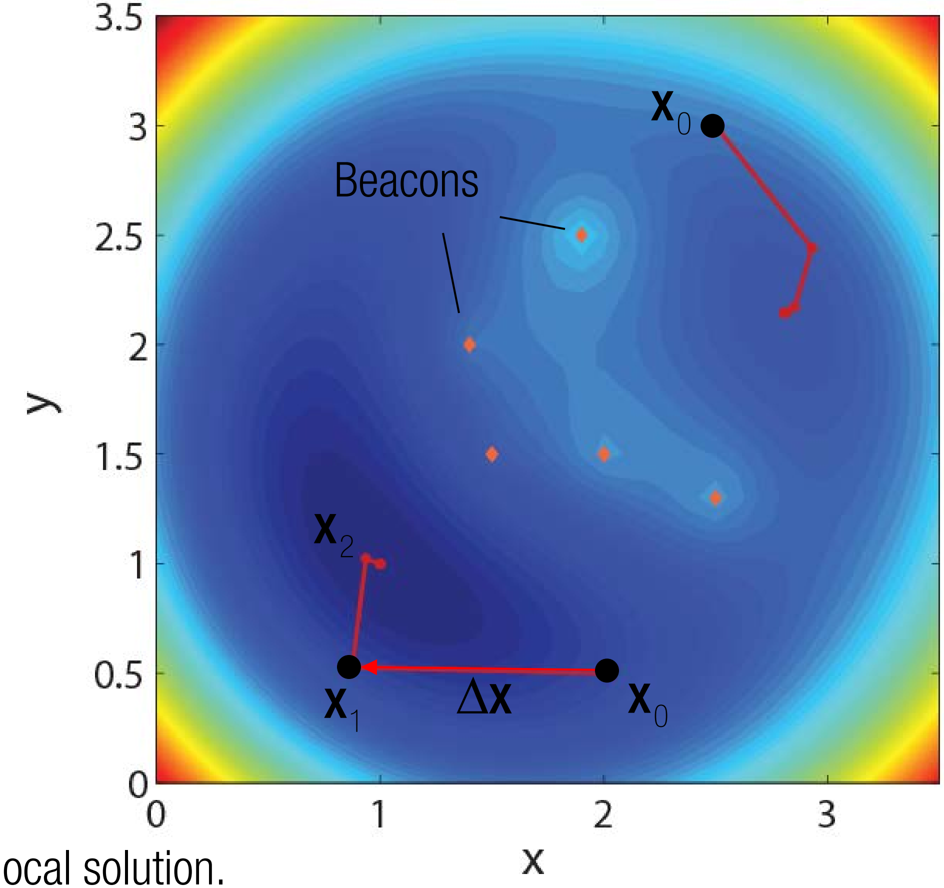
$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}}$$

where $\Delta\mathbf{x} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \left(\mathbf{b} - \mathbf{f}(\mathbf{x}) \right)$

$$\mathbf{f}(x, y) = \begin{bmatrix} \sqrt{(u_1 - x)^2 + (v_1 - y)^2} \\ \vdots \\ \sqrt{(u_5 - x)^2 + (v_5 - y)^2} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} r_1 \\ \vdots \\ r_5 \end{bmatrix}$$

$$\underbrace{\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{u_1 - x}{\sqrt{(u_1 - x)^2 + (v_1 - y)^2}} & \frac{v_1 - y}{\sqrt{(u_1 - x)^2 + (v_1 - y)^2}} \\ \frac{u_5 - x}{\sqrt{(u_5 - x)^2 + (v_5 - y)^2}} & \frac{v_5 - y}{\sqrt{(u_5 - x)^2 + (v_5 - y)^2}} \end{bmatrix}}_{\text{Jacobian matrix}}$$

Example:
Localization using range data from beacons



A different initialization converges to a different local solution.

How to Solve? (Nonlinear)

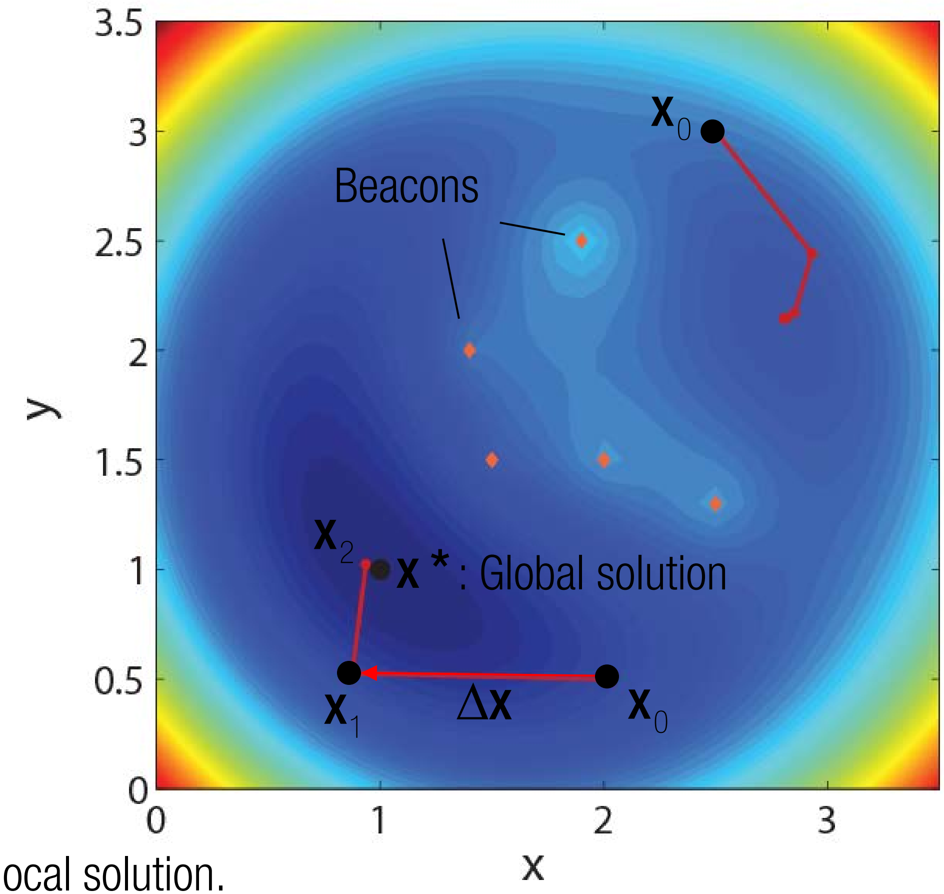
$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}}$$

where $\Delta\mathbf{x} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{T} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{T} \left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$

$$\mathbf{f}(x,y) = \begin{bmatrix} \sqrt{(u_1 - x)^2 + (v_1 - y)^2} \\ \vdots \\ \sqrt{(u_5 - x)^2 + (v_5 - y)^2} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} r_1 \\ \vdots \\ r_5 \end{bmatrix}$$

$$\underbrace{\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{u_1 - x}{\sqrt{(u_1 - x)^2 + (v_1 - y)^2}} & \frac{v_1 - y}{\sqrt{(u_1 - x)^2 + (v_1 - y)^2}} \\ \\ \frac{u_5 - x}{\sqrt{(u_5 - x)^2 + (v_5 - y)^2}} & \frac{v_5 - y}{\sqrt{(u_5 - x)^2 + (v_5 - y)^2}} \end{bmatrix}}_{\text{Jacobian matrix}}$$

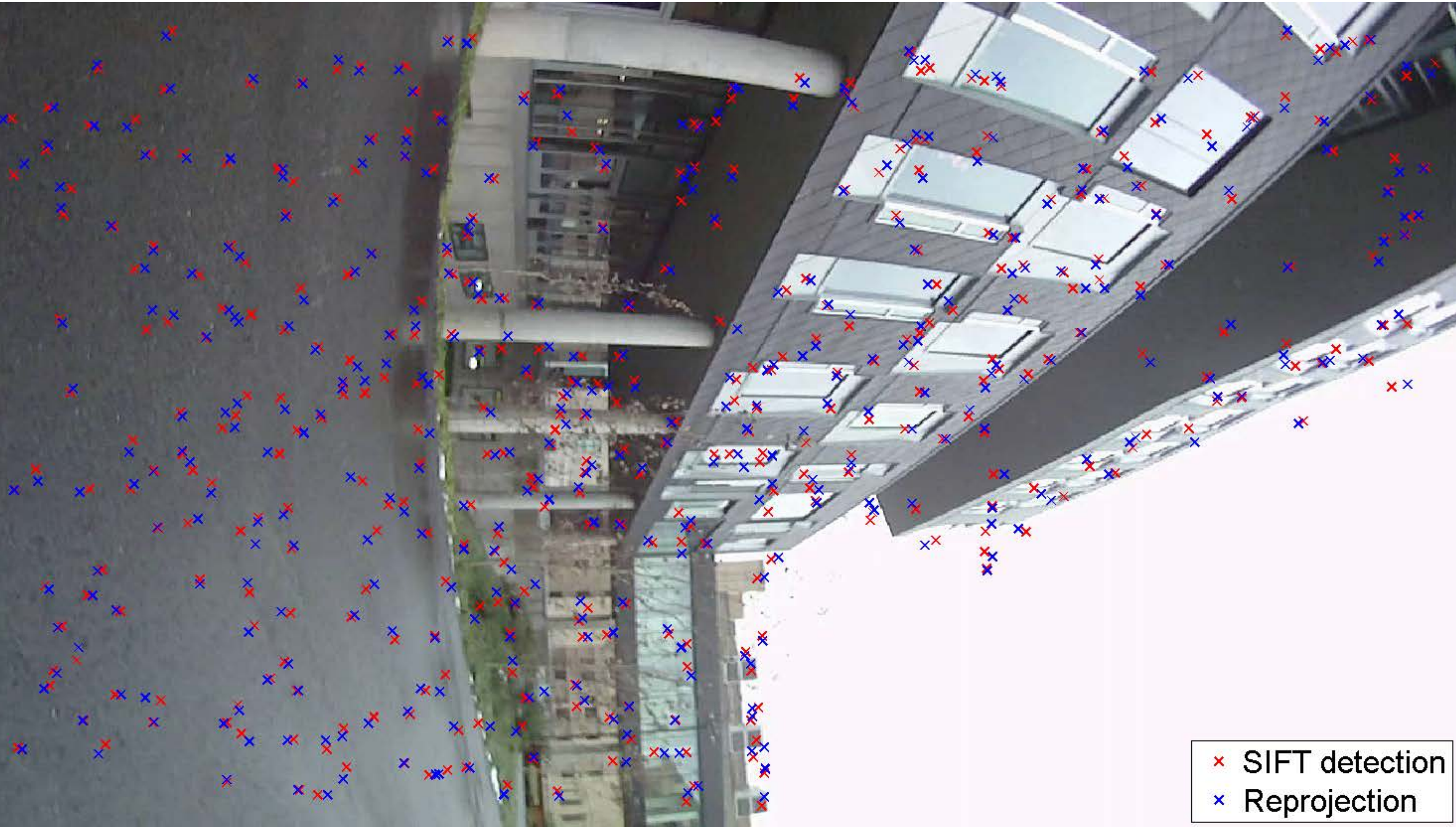
Example:
Localization using range data from beacons



A different initialization converges to a different local solution.

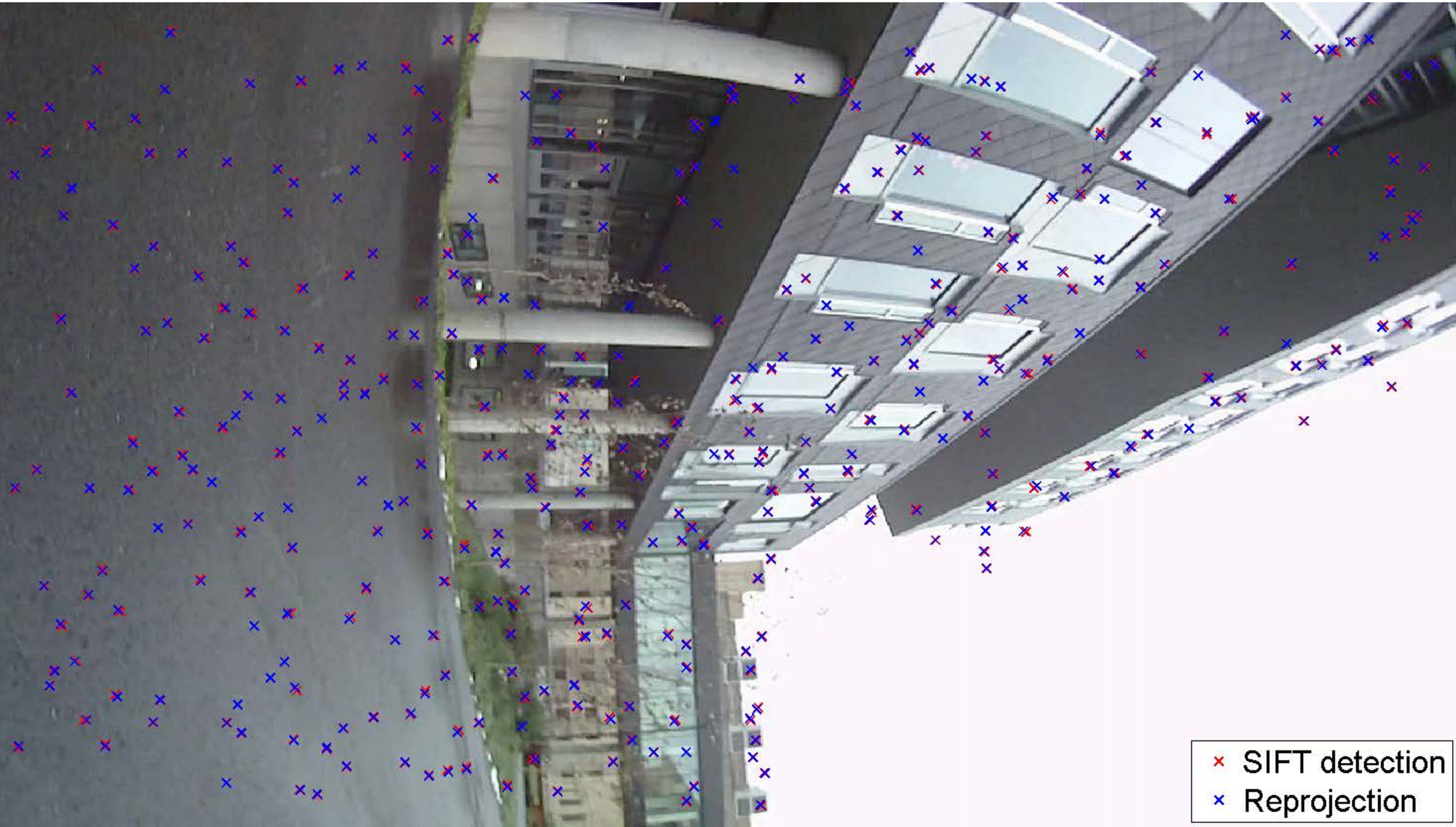
Geometric Refinement

Before Bundle Adjustment

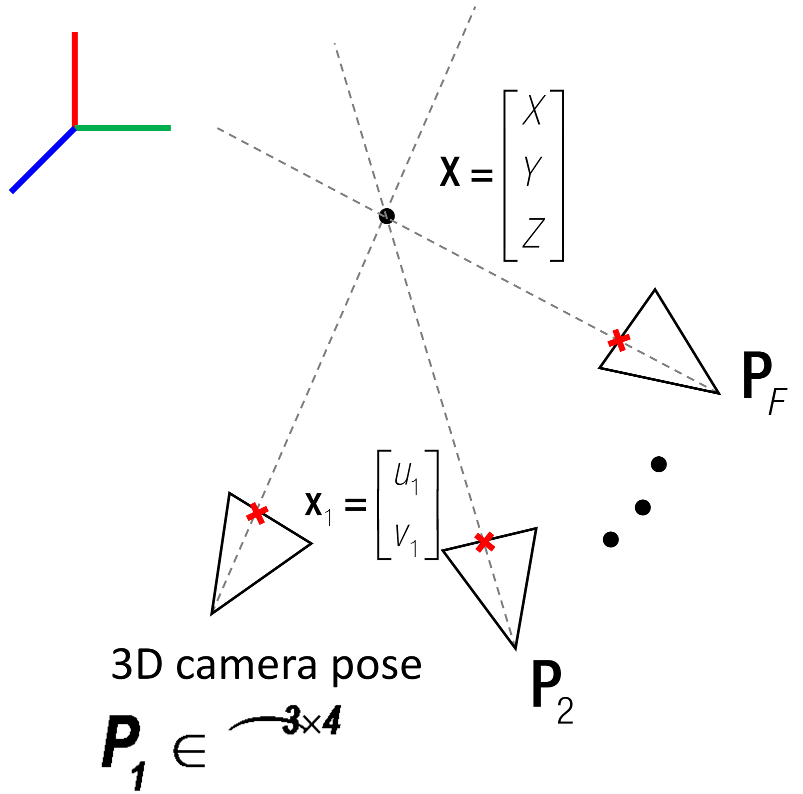


Geometric Refinement

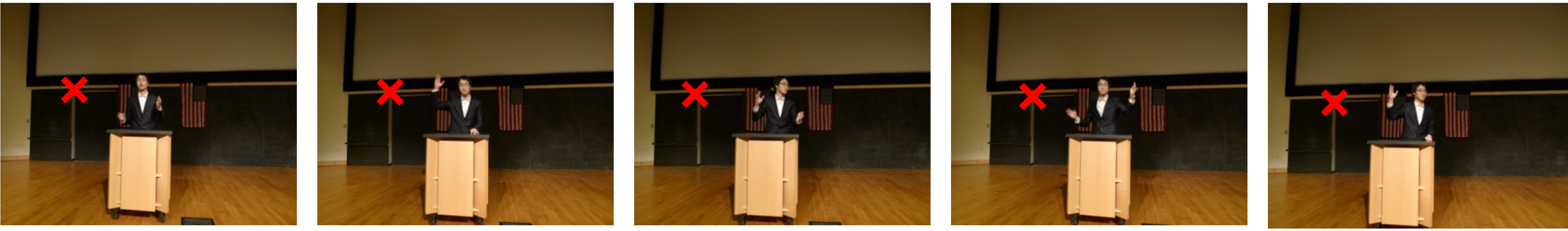
After Bundle Adjustment



Optical Flow: 2D point correspondences

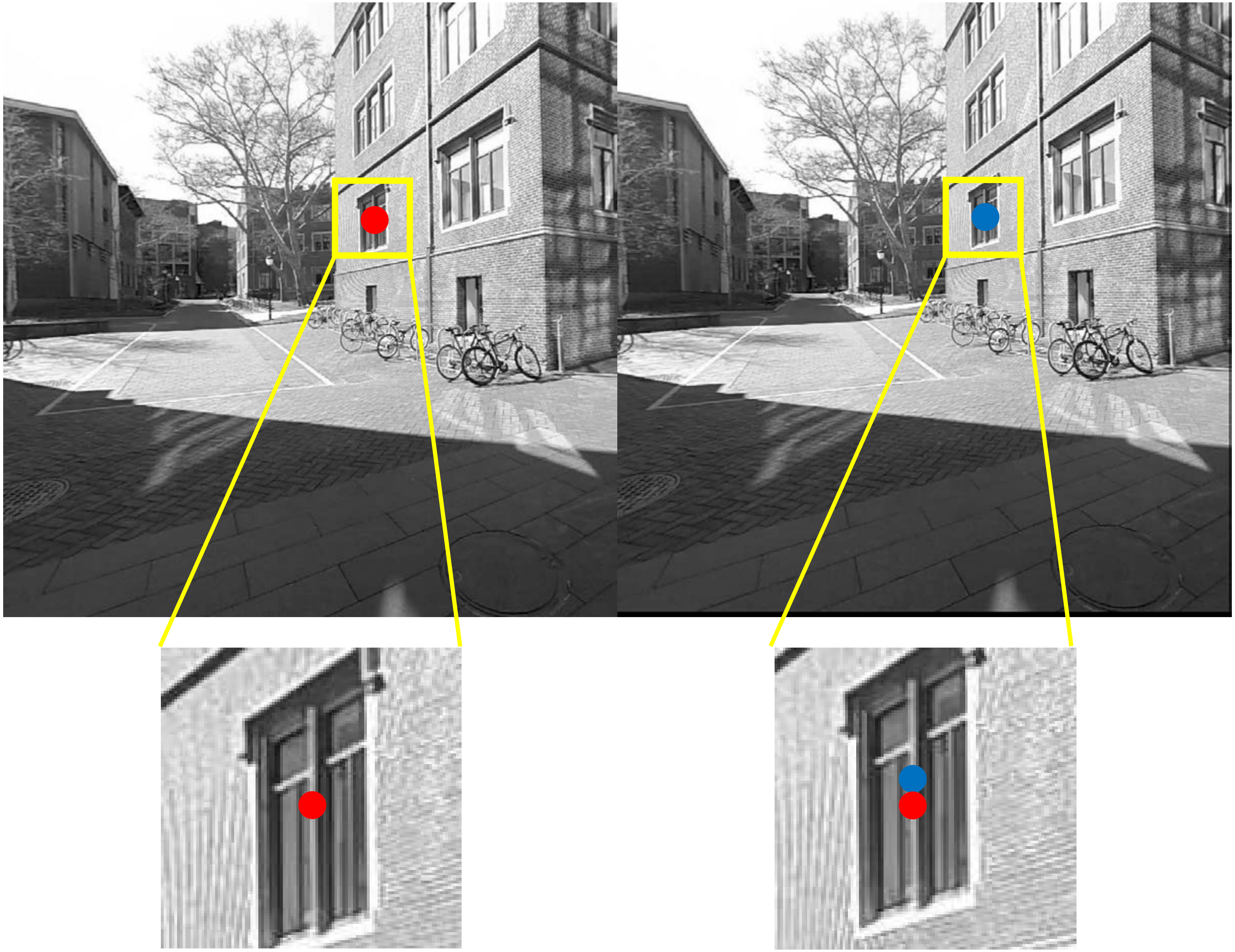


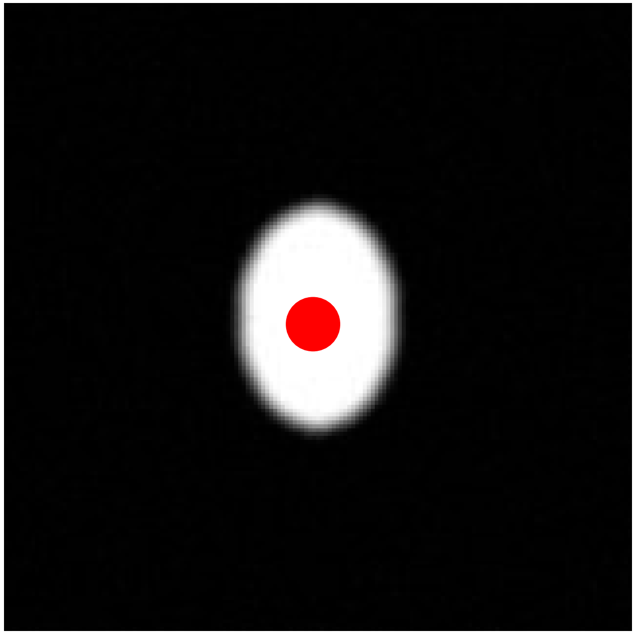
✕ 2D correspondences

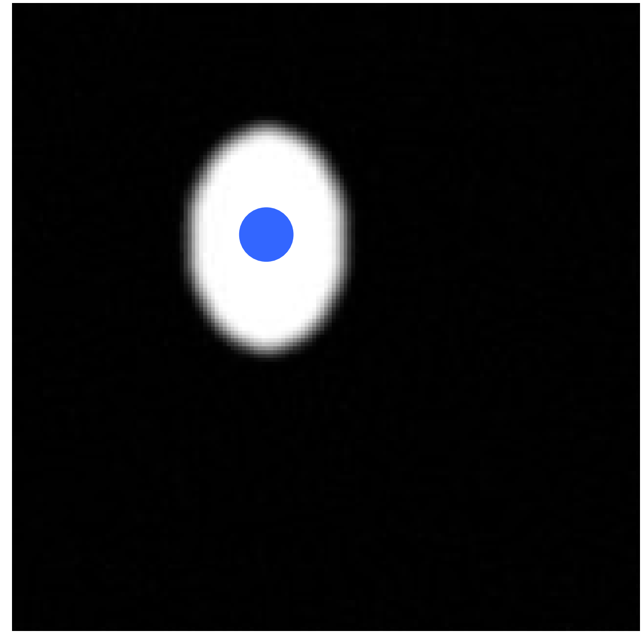


Optical Flow: 2D point correspondences

Optical Flow: 2D point correspondences

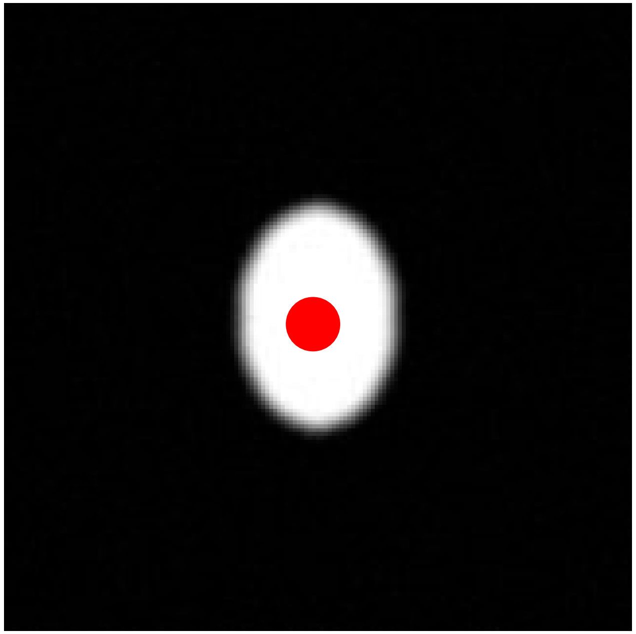




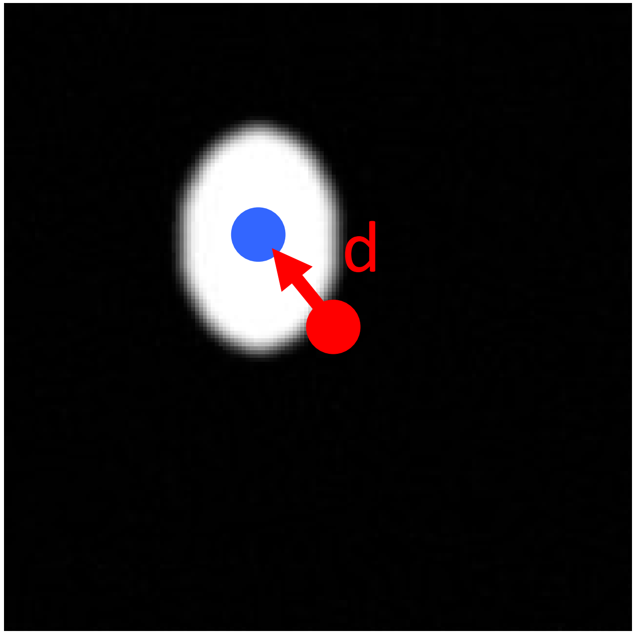


$\mathbf{I}(\mathbf{x})$ $\quad$ $\mathbf{J}(\mathbf{x})$

$t = 0$ $\quad$ $t = 1$



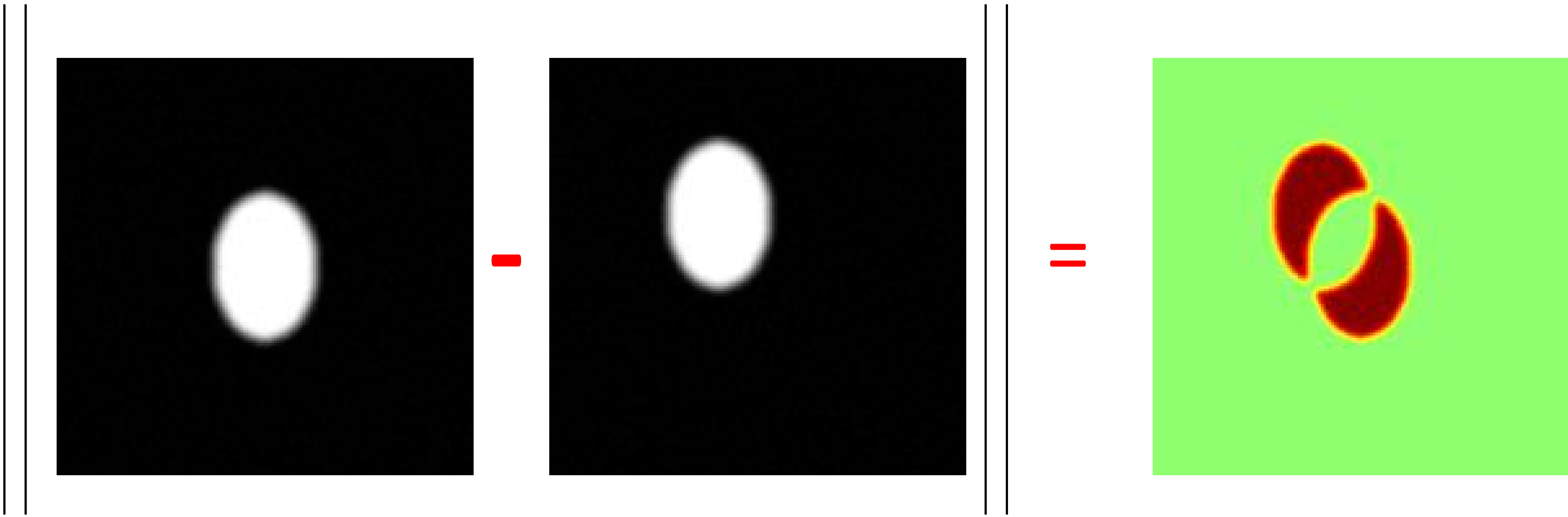
$\mathbf{I}(\mathbf{x})$



$\mathbf{J}(\mathbf{x})$

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x} + \mathbf{d})$$

$$\min_{\mathbf{d}} E(\mathbf{d}) = \|\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})\|^2$$



When d=0

E(d=0)

Three steps for solving this problem

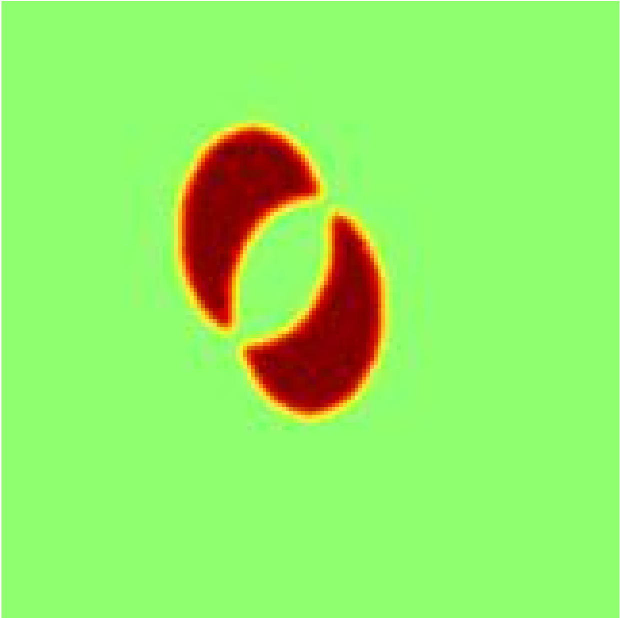
$$1\text{: Solve for } \left.\frac{\delta E}{\delta \mathbf{d}}\right|_{\mathbf{d}^*} = 0$$

$$2\text{: Taylor expansion on } \mathbf{J}(\mathbf{x}+\mathbf{d})$$

$$3\text{: Solve for d, warp image, iterate}$$

Step 1: Solve for $\left.\dfrac{\delta E}{\delta \mathbf{d}}\right|_{d^*} = 0$

$$E(\mathbf{d}) = ||\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})||^{\mathbf{2}}$$

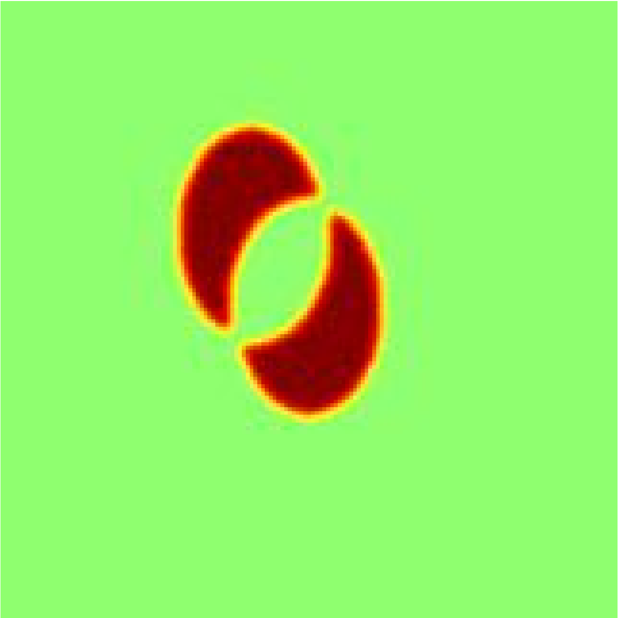


$$\mathbf{E}(\mathbf{d}^* = \mathbf{0})$$

Step 1: Solve for $\left.\frac{\delta E}{\delta \mathbf{d}}\right|_{d^*} = 0$

$$E(\mathbf{d}) = ||\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})||^{\mathbf{2}}$$

$$\left.\frac{\delta E}{\delta \mathbf{d}}\right|_{d^*} = 2\frac{\delta \mathbf{J}(\mathbf{x}+\mathbf{d})}{\delta \mathbf{d}}^T (\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})) = 0$$



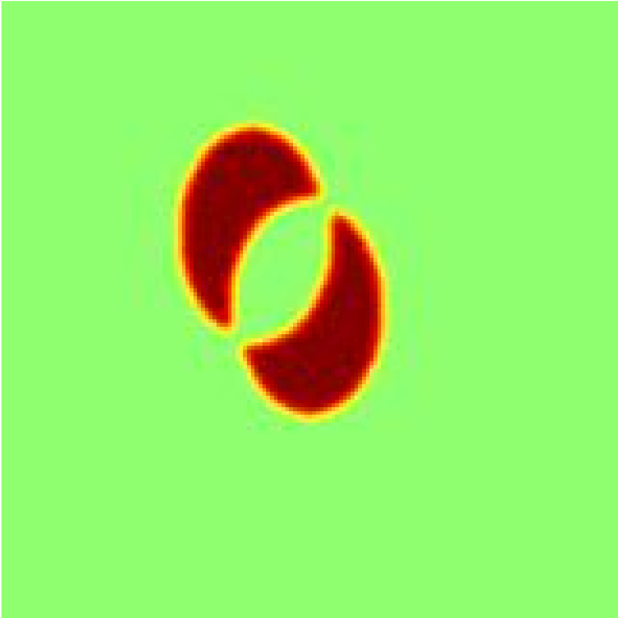
$$\mathbf{E}(\mathbf{d}^* = \mathbf{0})$$

Step 1: Solve for $\left.\frac{\delta E}{\delta \mathbf{d}}\right|_{d*} = 0$

$$E(\mathbf{d}) = ||\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})||^{\mathbf{2}}$$

$$\left.\frac{\delta E}{\delta \mathbf{d}}\right|_{d*} = 2\frac{\delta \mathbf{J}(\mathbf{x}+\mathbf{d})}{\delta \mathbf{d}}^{T}(\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})) = 0$$

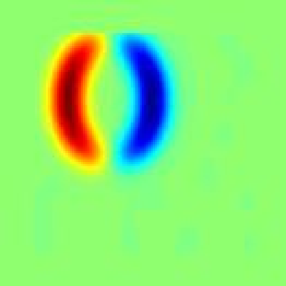
$$\left.\frac{\delta E}{\delta \mathbf{d}}\right|_{d*} = 2\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T}(\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})) = 0$$



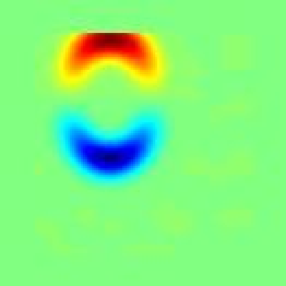
$$\mathbf{E}(\mathbf{d}^{*} = \mathbf{0})$$

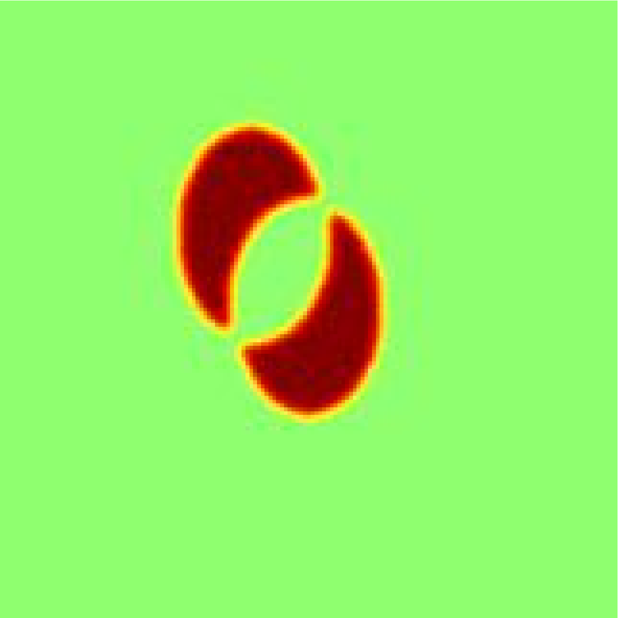
$$\left.\frac{\delta E}{\delta \mathbf{d}}\right|_{d^*} = 2\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T (\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})) = 0$$

$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta x} =$$



$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta y} =$$

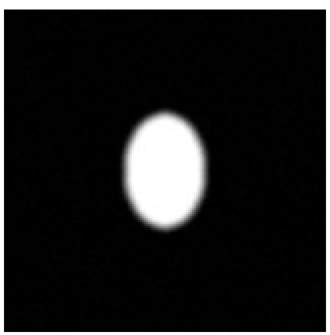




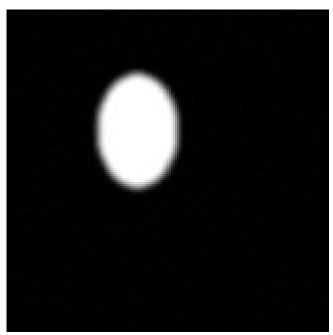
$$\mathbf{E}(\mathbf{d}^* = \mathbf{0})$$

Step 2: Taylor expansion $\mathbf{J}(\mathbf{x}+\mathbf{d})$

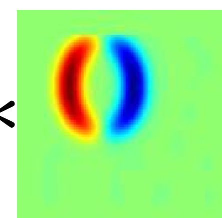
$$\mathbf{J}(\mathbf{x}+\mathbf{d}) = \mathbf{J}(\mathbf{x}) + \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}\mathbf{d}$$



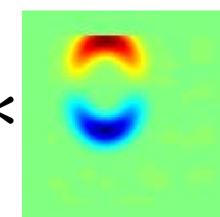
=



$+\mathbf{d_x}*$



$+\mathbf{d_y}*$



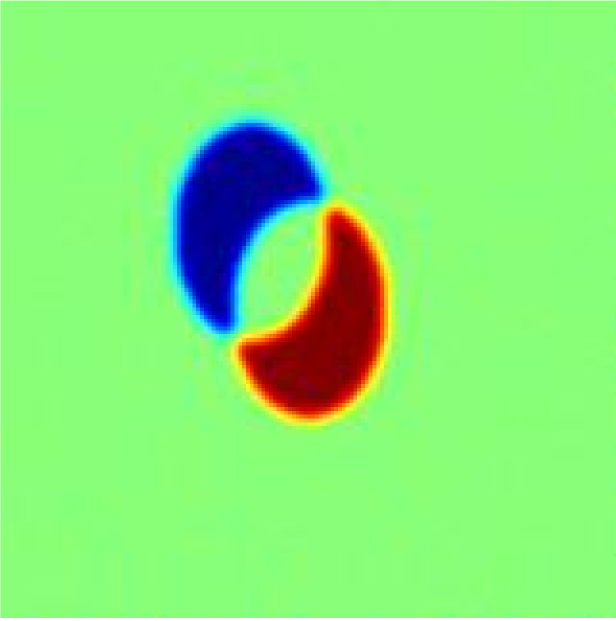
Putting all together

$$\left.\frac{\delta E}{\delta \mathbf{d}}\right|_{d*} = 2\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T (\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})) = 0$$

$$\mathbf{J}(\mathbf{x}+\mathbf{d}) = \mathbf{J}(\mathbf{x}) + \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}\mathbf{d}$$

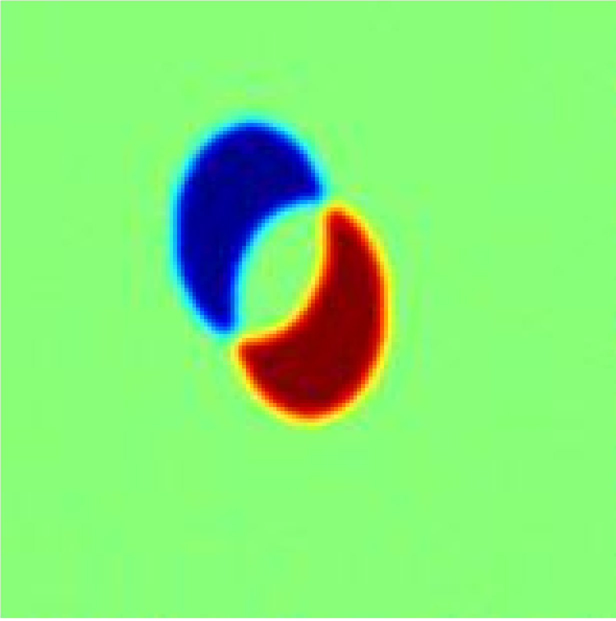
$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}} \mathbf{d} = \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$

$$\boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}}^T \boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}} \mathbf{d} = \boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}}^T (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$



$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}} \mathbf{d} = \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$

2D unknowns flow vector per pixel
2 equations



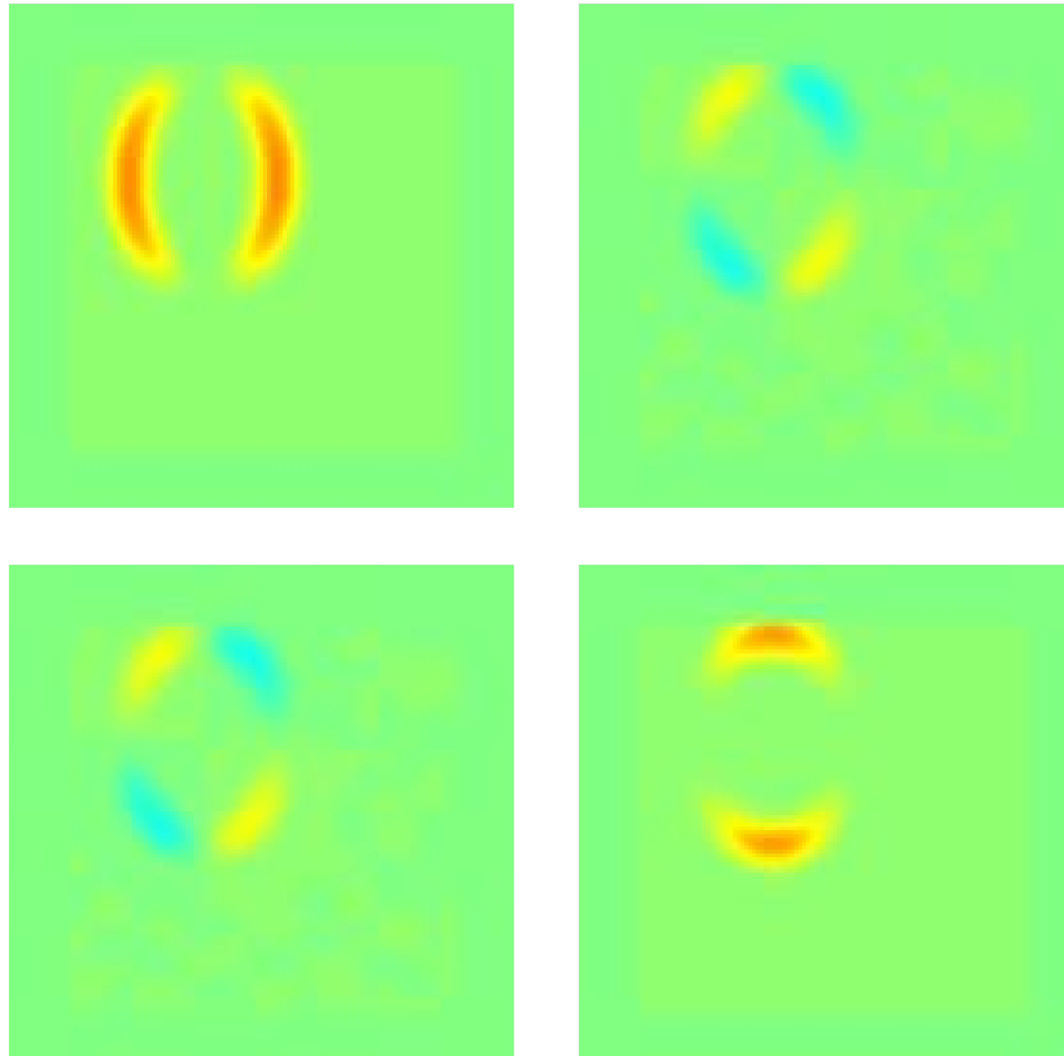
$$\boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}} \mathbf{d} = \boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}}^{T} (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$

Also known as second moment matrix

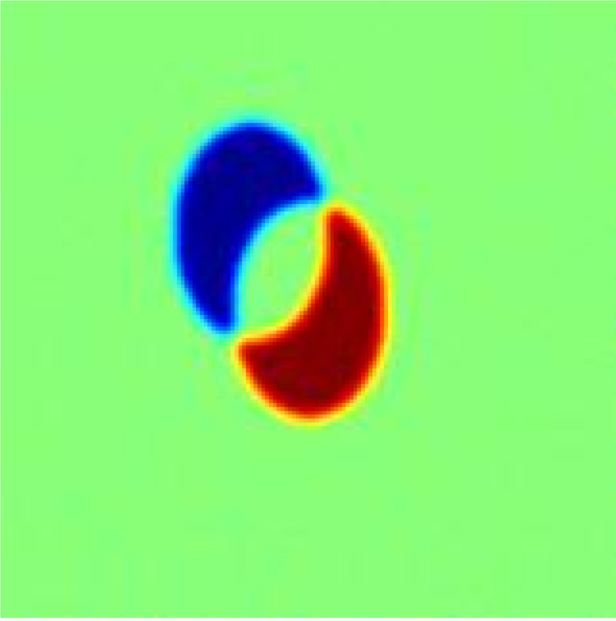
$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}$$

$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta x}^{2} \qquad \frac{\delta \mathbf{J}(\mathbf{x})}{\delta x}\frac{\delta \mathbf{J}(\mathbf{x})}{\delta y}$$

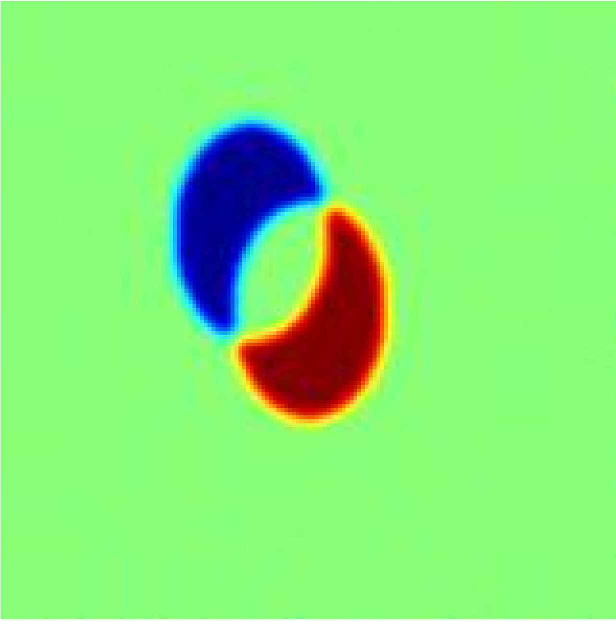
$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta x}\frac{\delta \mathbf{J}(\mathbf{x})}{\delta y} \qquad \frac{\delta \mathbf{J}(\mathbf{x})}{\delta y}^{2}$$

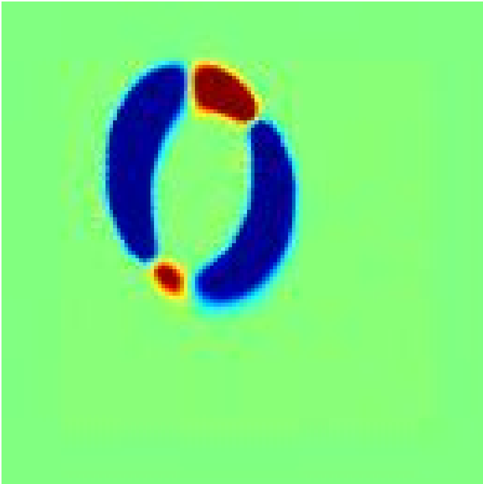


$$\boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}} \mathbf{d} = \boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}}^T (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$

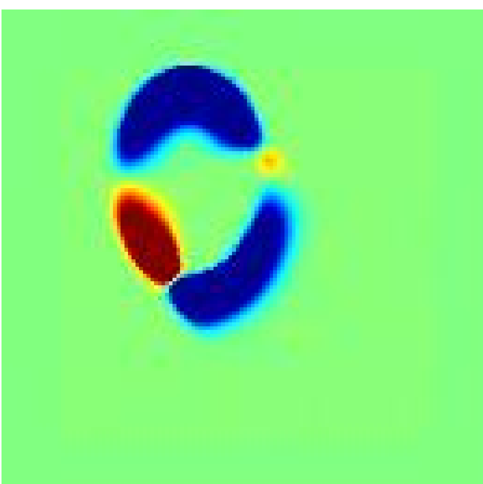


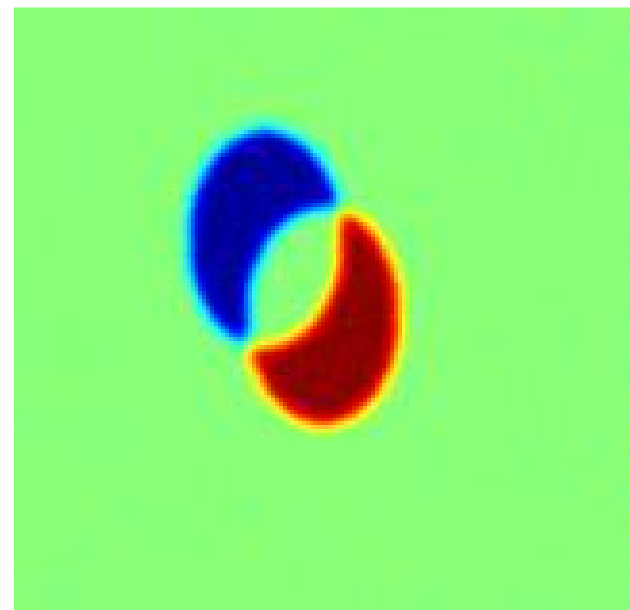
$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}} \mathbf{d} = \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$



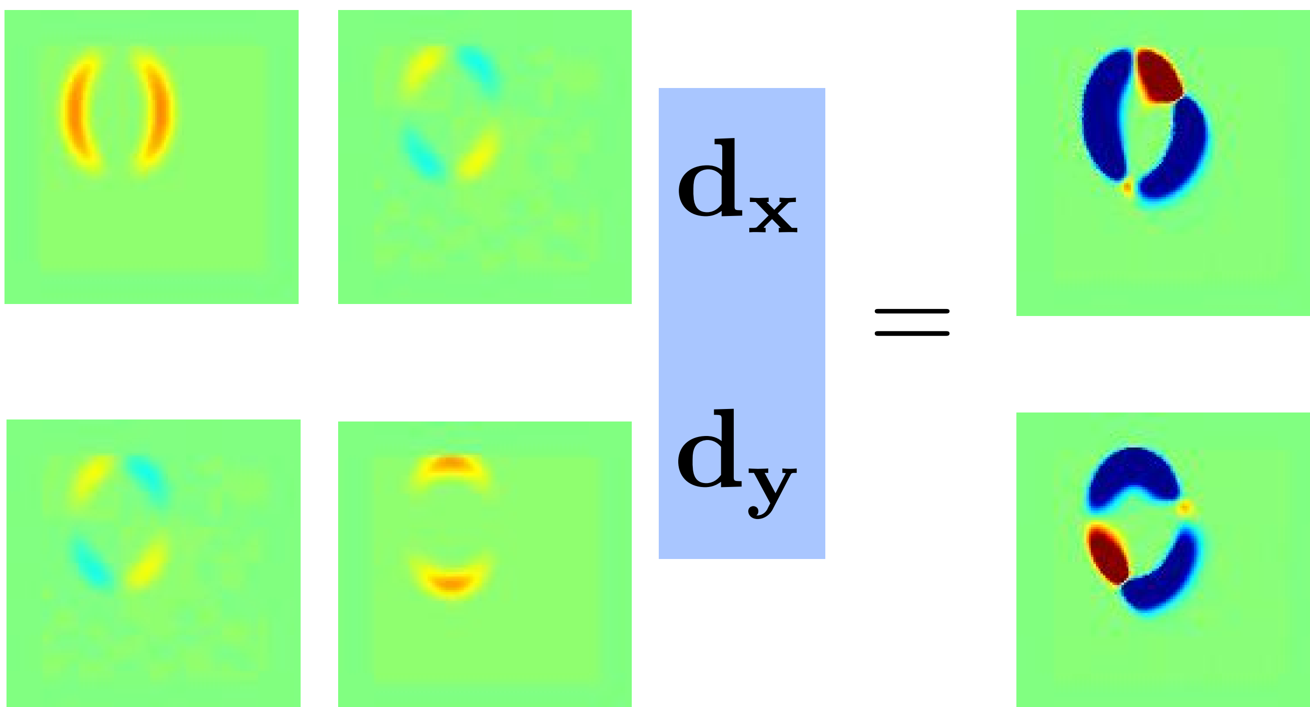


$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} \left(\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x})\right)$$

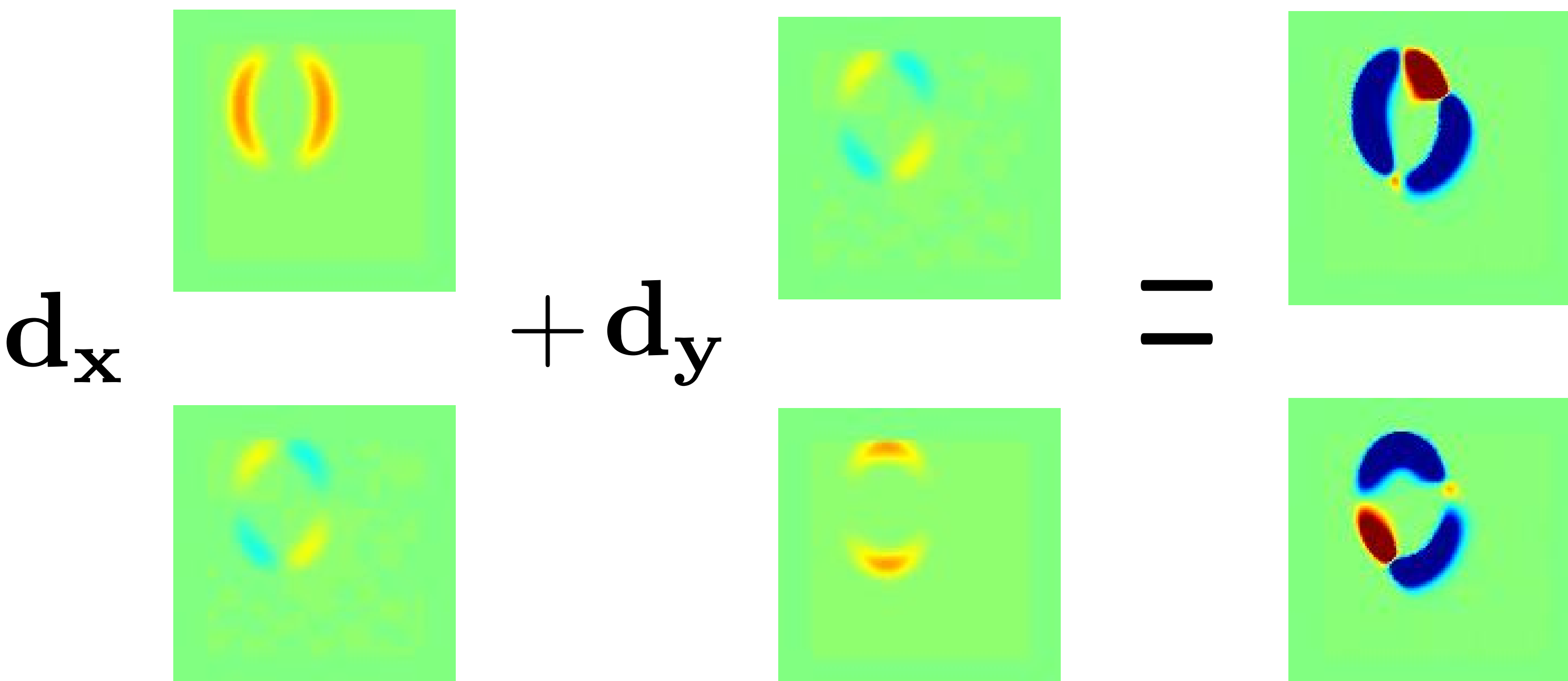




$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}} \mathbf{d} = \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$



$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}} \mathbf{d} = \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$

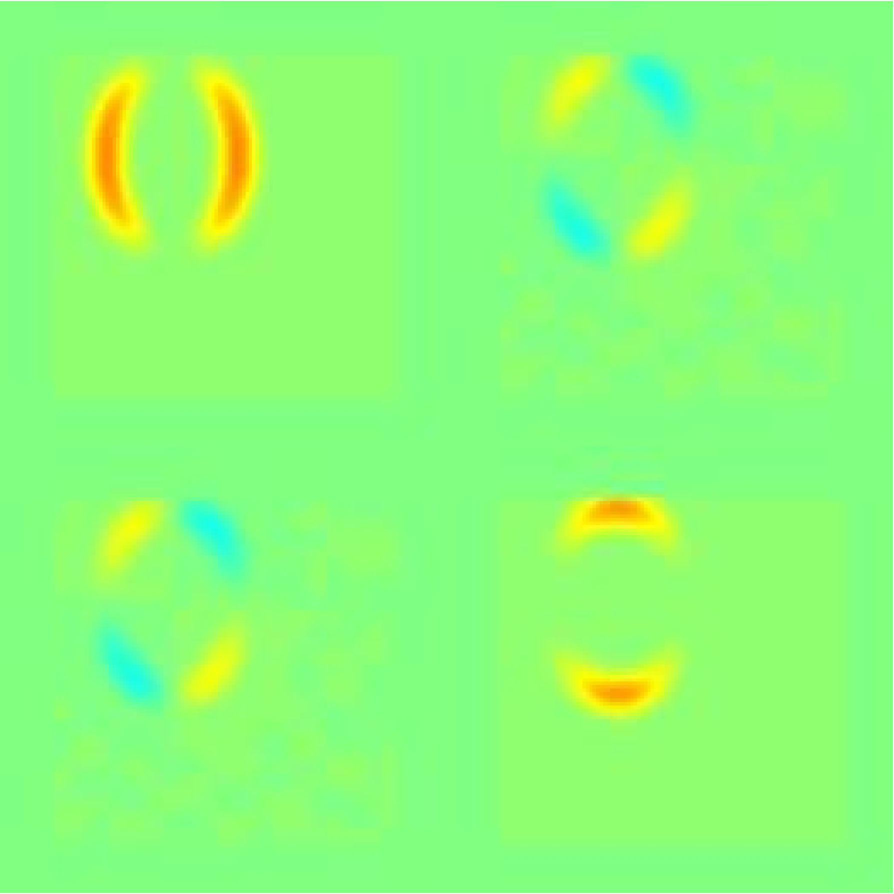


$$\mathbf{min}_{\mathrm{d}} E(\mathbf{d}) = \sum_{\mathbf{x}} ||\mathbf{J}(\mathbf{x} + \mathbf{d}) - \mathbf{I}(\mathbf{x})||^{\mathbf{2}}$$

$$\mathbf{min}_{\mathbf{d}} E(\mathbf{d}) = \sum_{\mathbf{x}} \|\mathbf{J}(\mathbf{x}+\mathbf{d}) - \mathbf{I}(\mathbf{x})\|^{\mathbf{2}}$$

$$\sum_{\mathbf{x}} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}} \mathbf{d} = \sum_{\mathbf{x}} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^T (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$

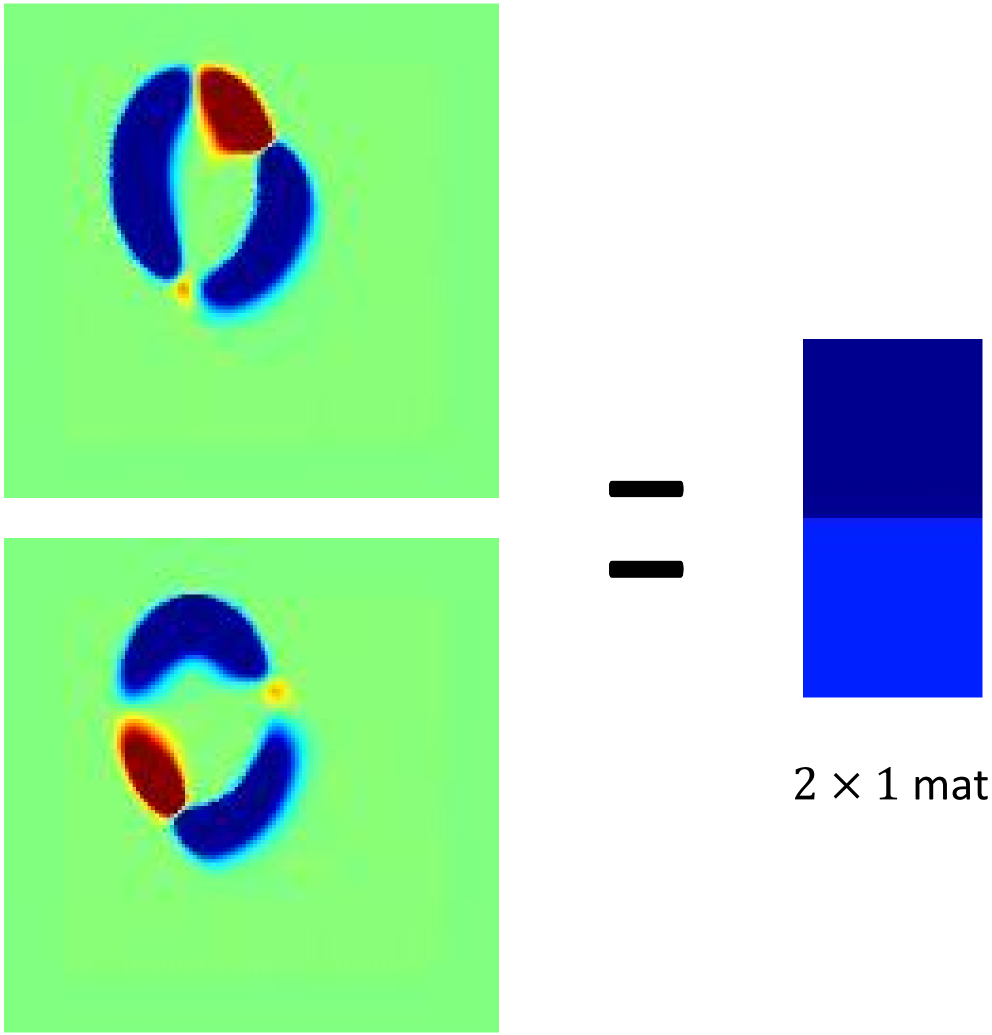
Summing over pixels



=

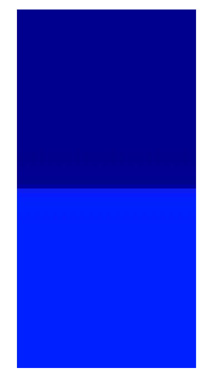
2 × 2 matrix

Summing over pixels

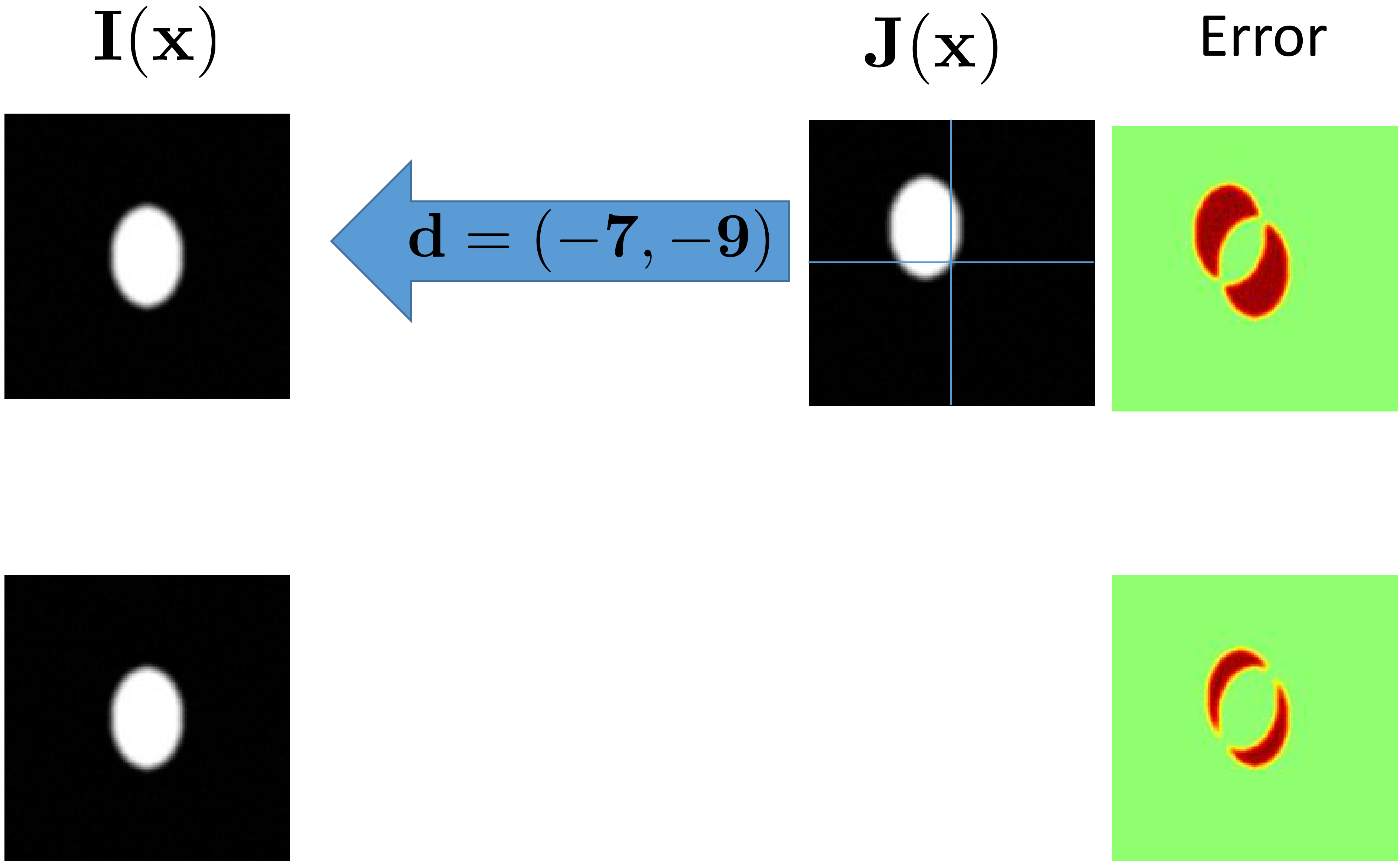


2×1 matrix

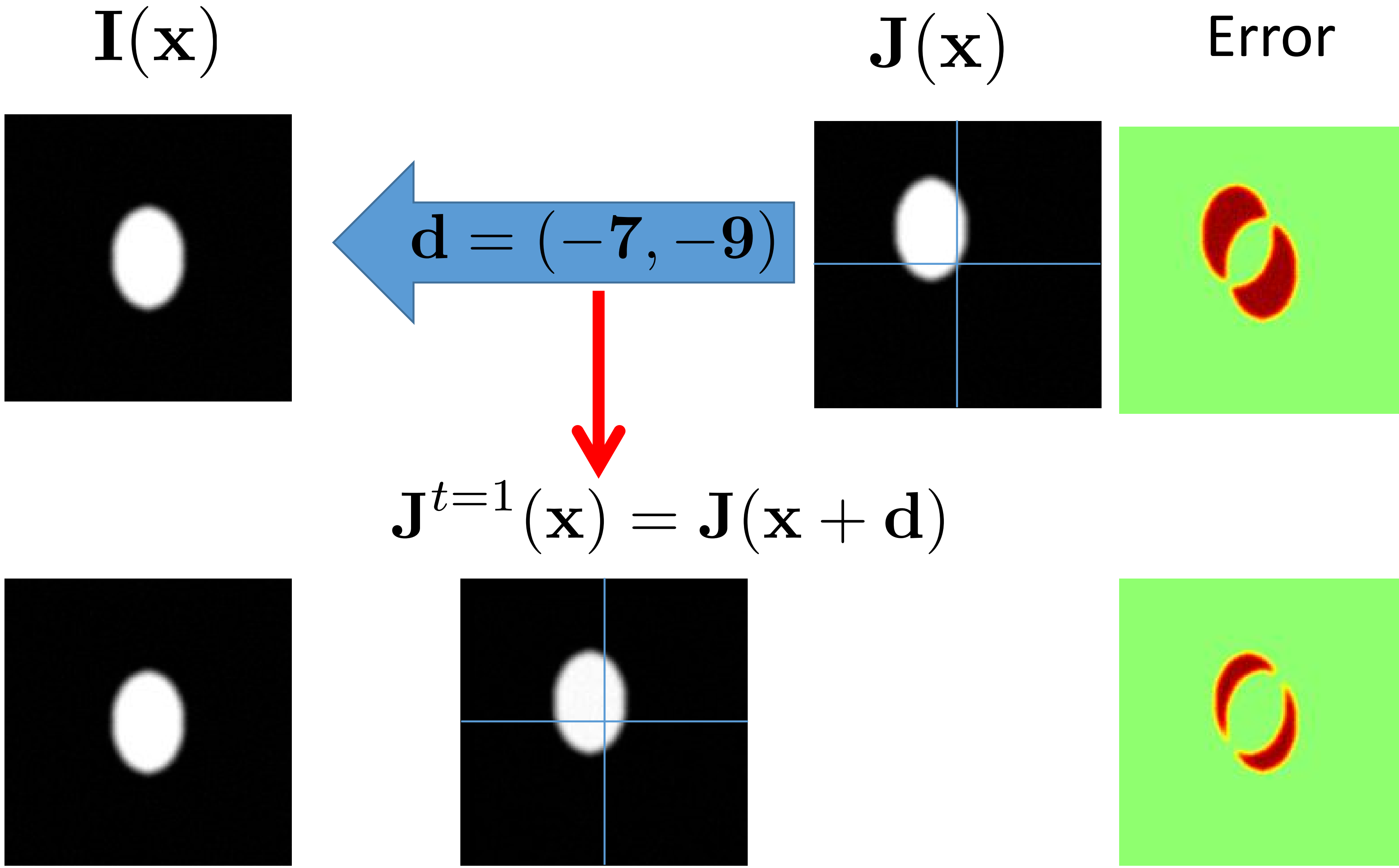
$$\sum_{\mathbf{x}} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}} \mathbf{d} = \sum_{\mathbf{x}} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} \left(\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x})\right)$$

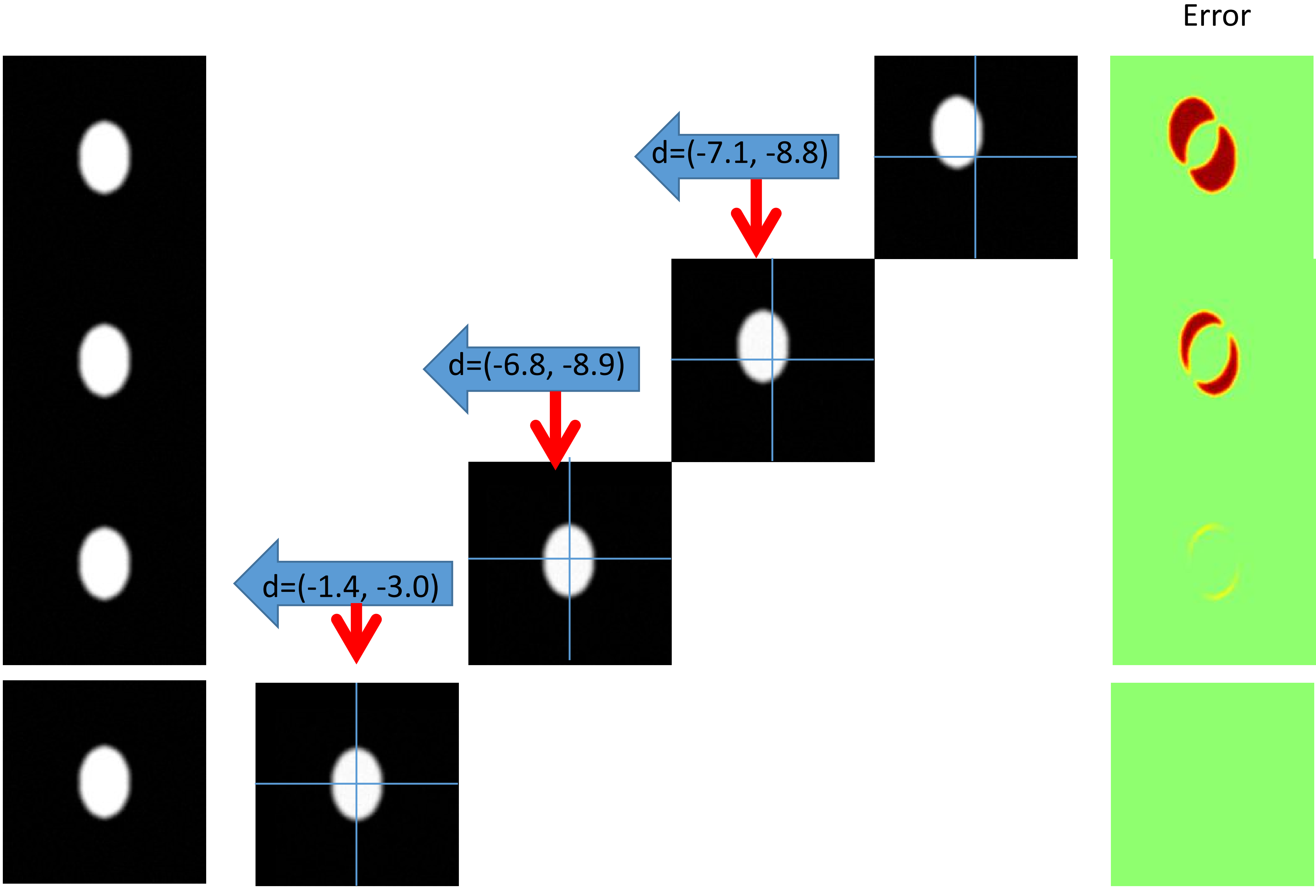
$\mathbf{d}_{\mathbf{x}}$ $+\,\mathbf{d}_{\mathbf{y}}$ $=$ 

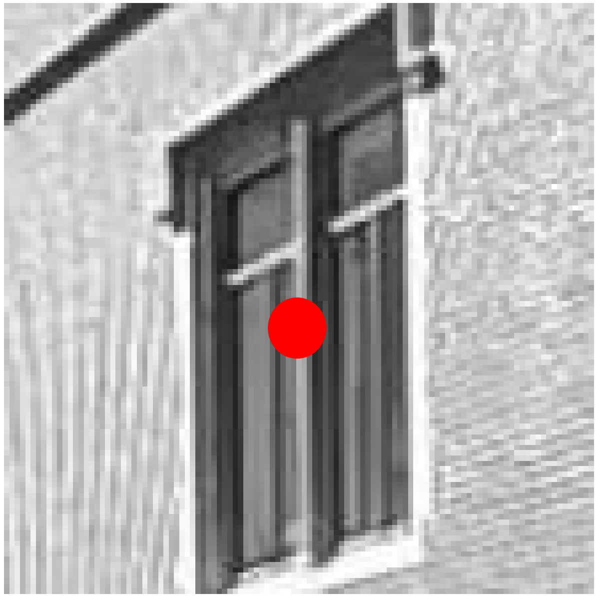
3: Solve for d, warp image, iterate



3: Solve for d, warp image, iterate

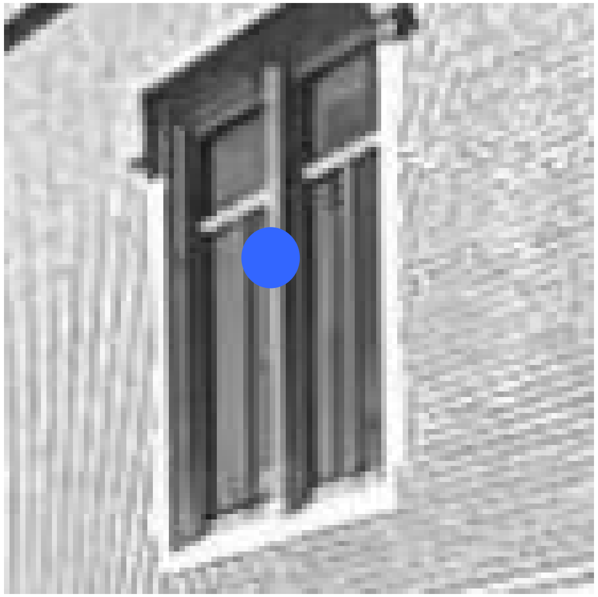



Error
d=(-7.1, -8.8)
d=(-6.8, -8.9)
d=(-1.4, -3.0)



$\mathbf{I}(\mathbf{x})$

$t = 0$



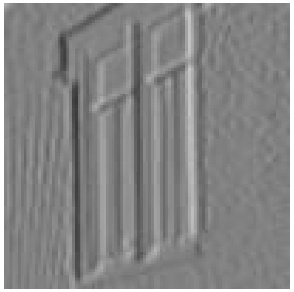
$\mathbf{J}(\mathbf{x})$

$t = 1$

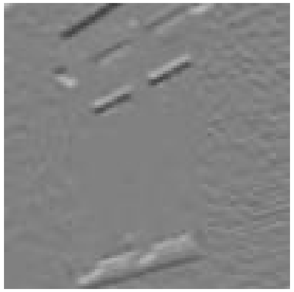
$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}} \mathbf{d} = \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} \boxed{(\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))}$$

$$\boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}}^T \boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}} \mathbf{d} = \boxed{\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}}^T (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$

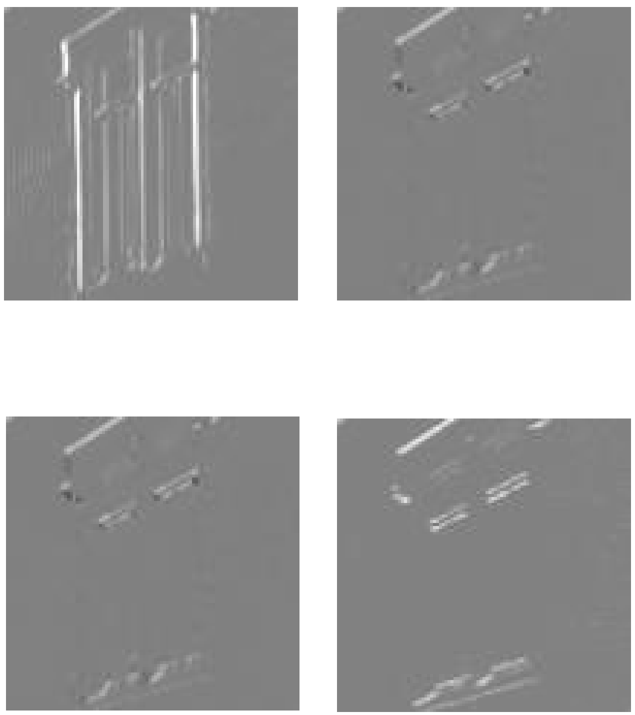
$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta x} =$$



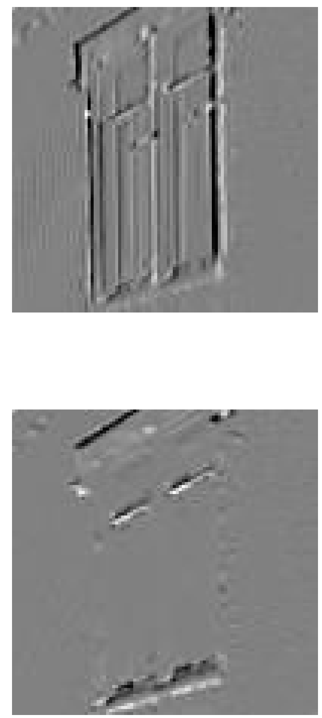
$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta y} =$$



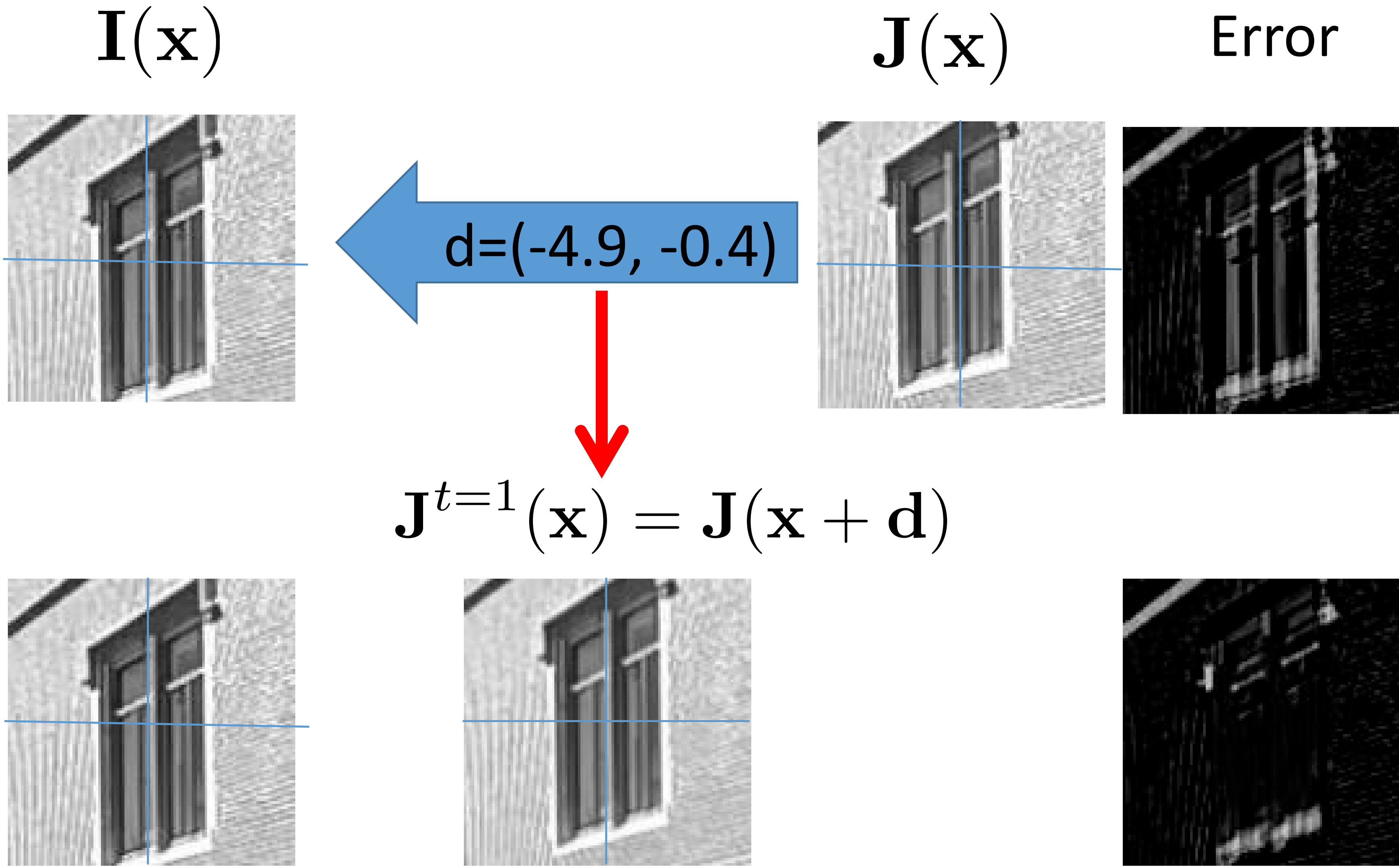
$$\frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}} \mathbf{d} = \frac{\delta \mathbf{J}(\mathbf{x})}{\delta \mathbf{x}}^{T} (\mathbf{I}(\mathbf{x}) - \mathbf{J}(\mathbf{x}))$$

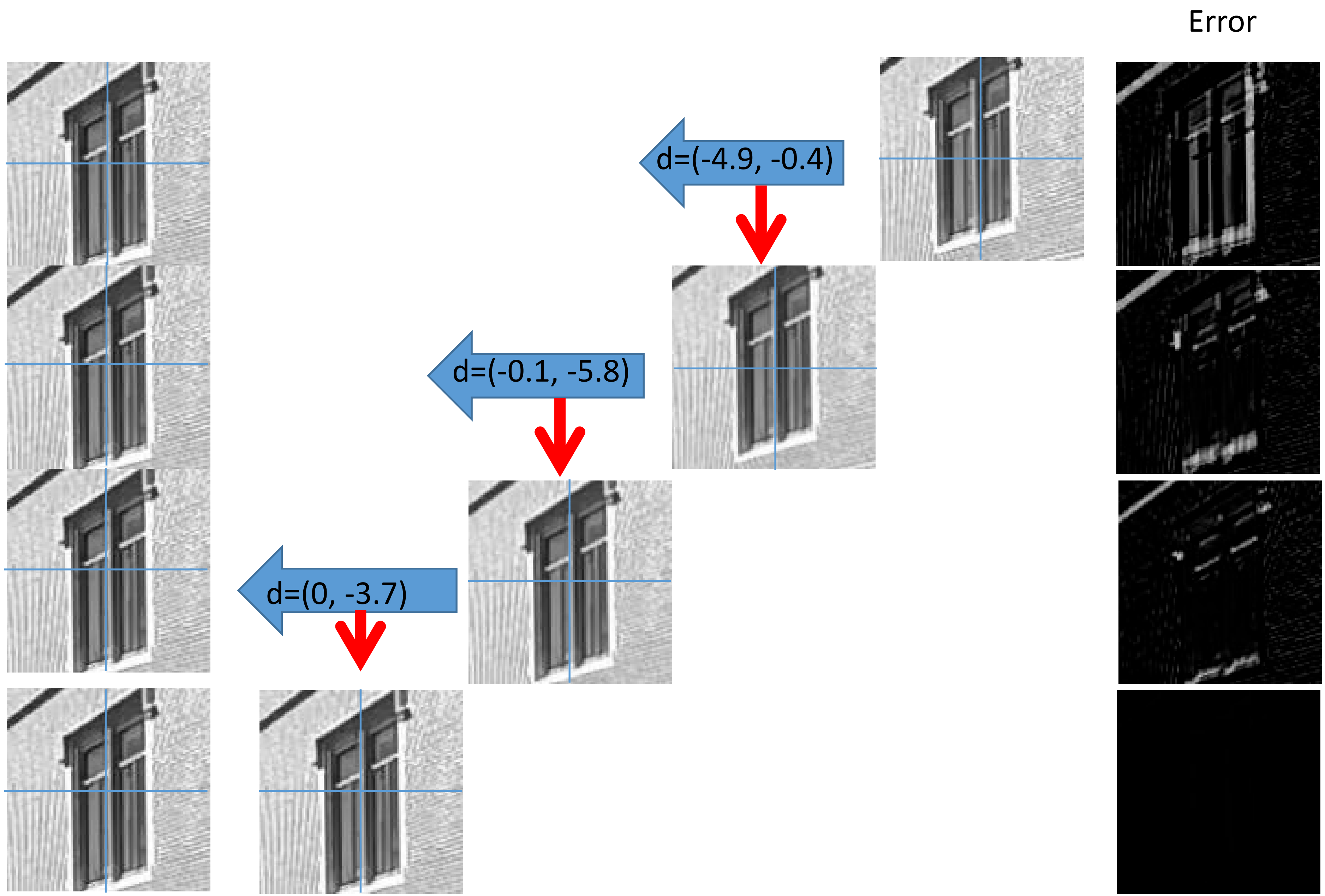


$$\begin{bmatrix} \mathbf{d_x} \\ \mathbf{d_y} \end{bmatrix} =$$



3: Solve for d, warp image, iterate




Error
d=(-4.9, -0.4)
d=(-0.1, -5.8)
d=(0, -3.7)

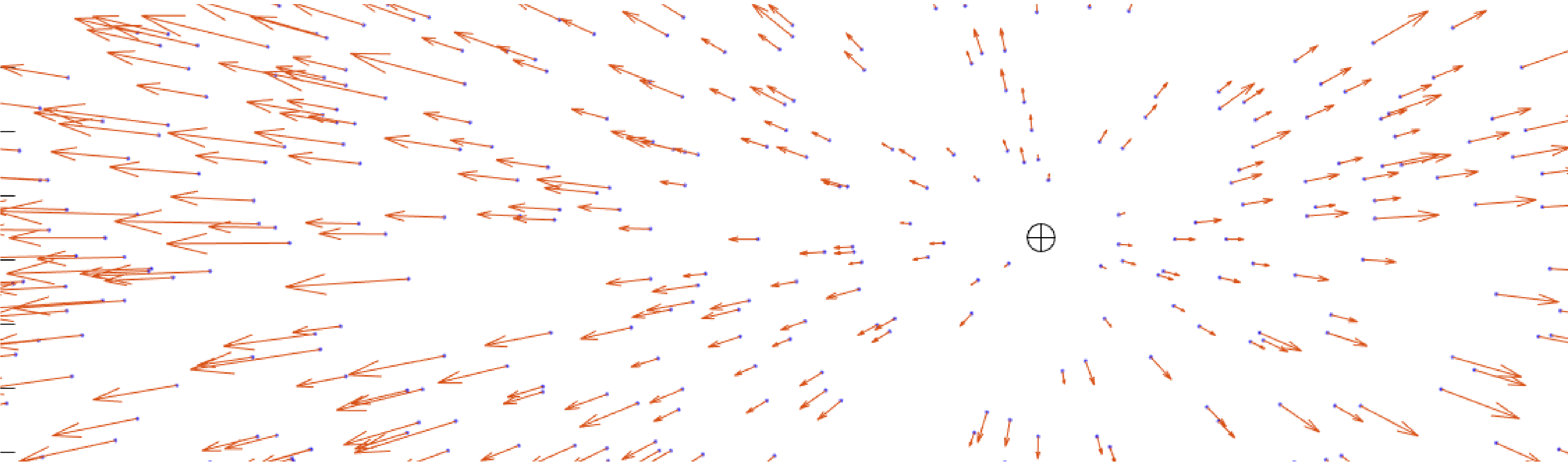
Perception:
3D Velocities from Optical Flow

Kostas Daniilidis

Which direction is this quadrotor moving?

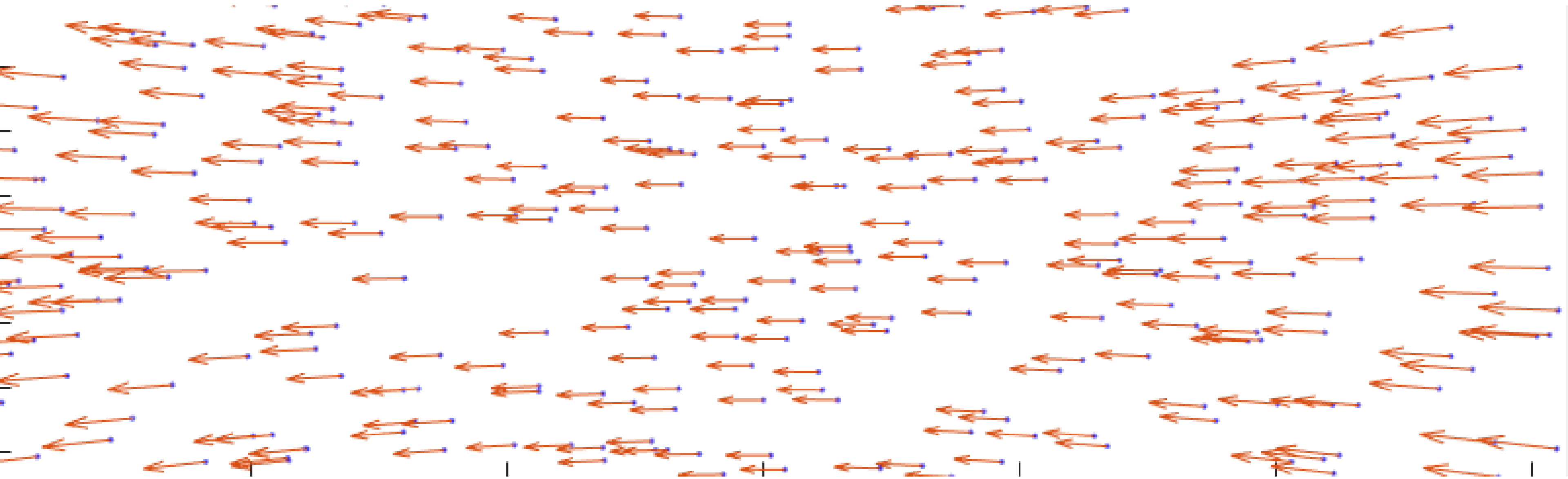
Which direction is this car moving?

If the camera translates only



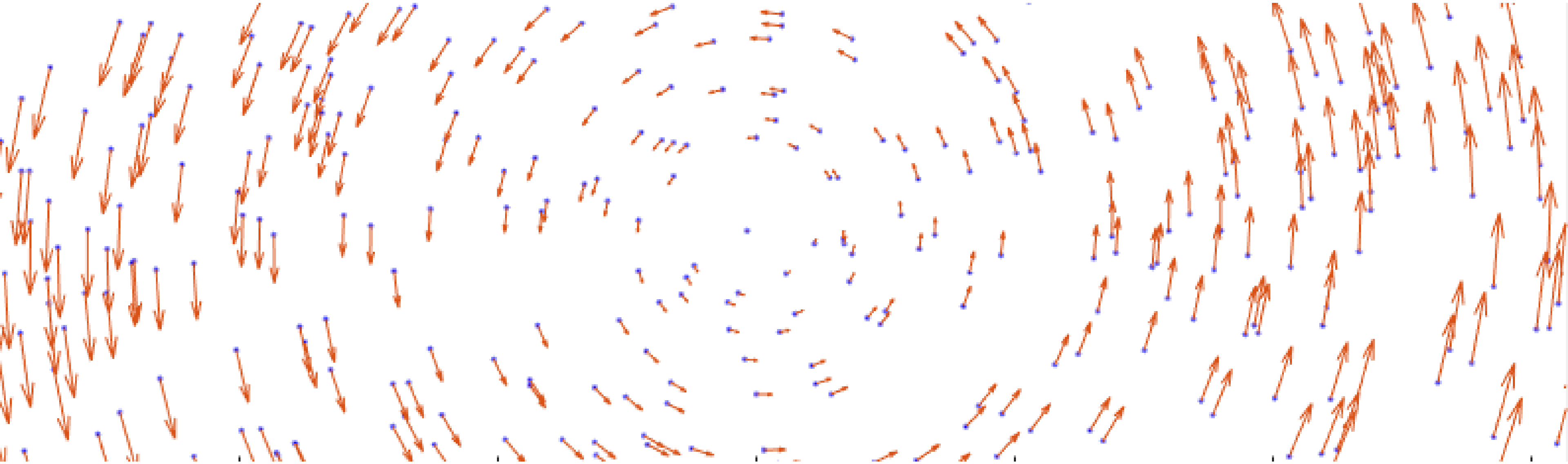
… the vector field is radially expanding
from a point called the Focus of Expansion and we
can quite easily infer our direction of motion.

If the camera rotates only ..



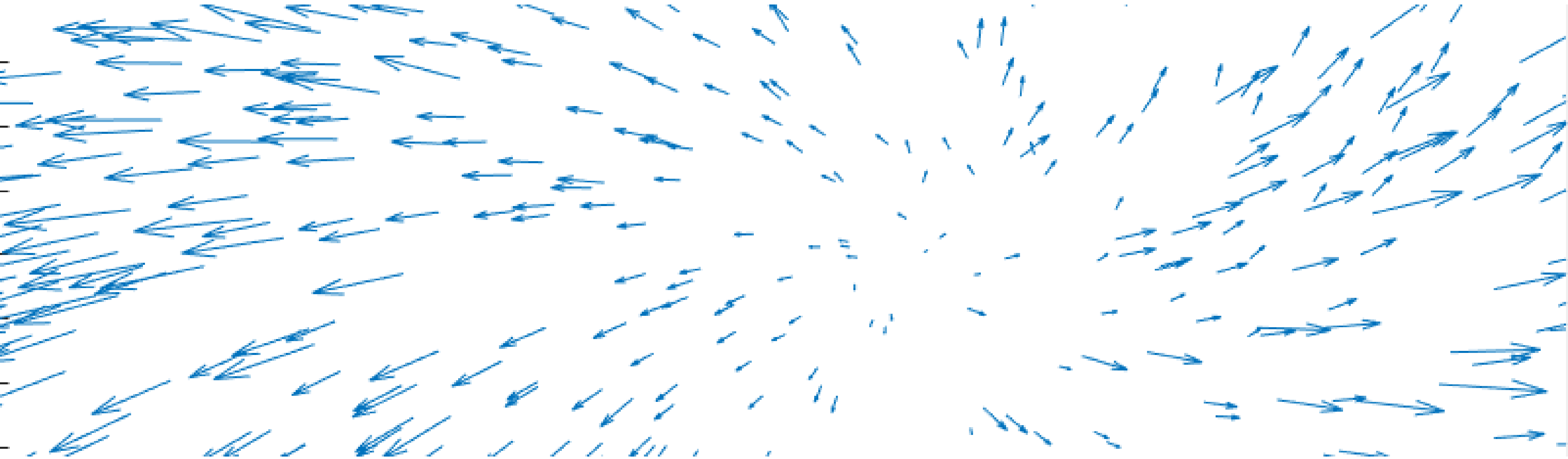
.... for example around the vertical axis the pattern is almost horizontal and vectors are longer left and right.

If the camera rotates only around optical axis



.... for example around the optical axis
we obtain a curling vector field.

If we combine translation and rotation..



…. It is hard to tell whereto we are moving!

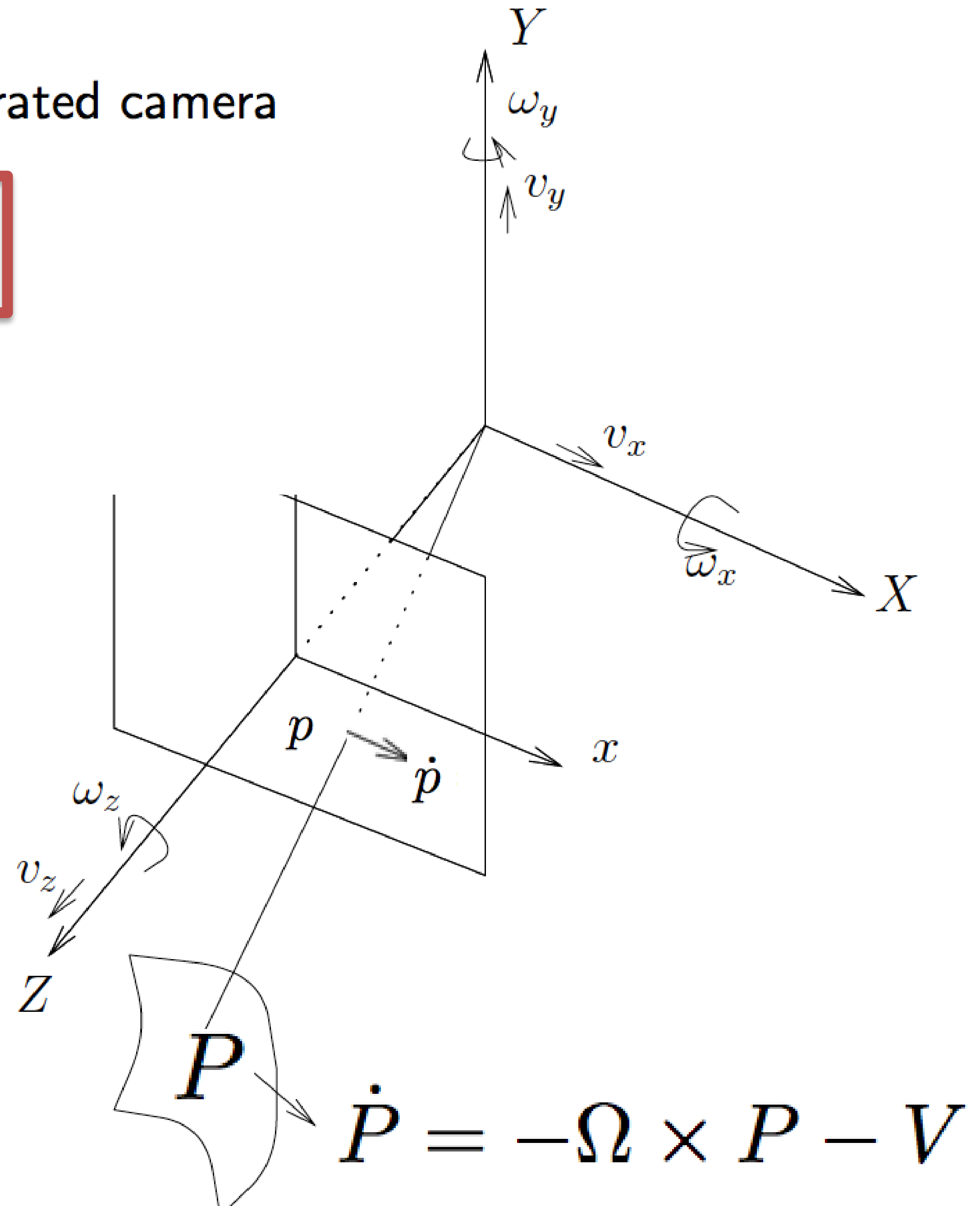
Projection equations for calibrated camera

$x = \frac{X}{Z}$, $y = \frac{Y}{Z}$

or in vector notation $\boxed{p = \frac{1}{Z}P}$

Differentiating w.r.t. time yields:

$$\dot{p} = \frac{\dot{P}}{Z} - \frac{\dot{Z}}{Z}p$$



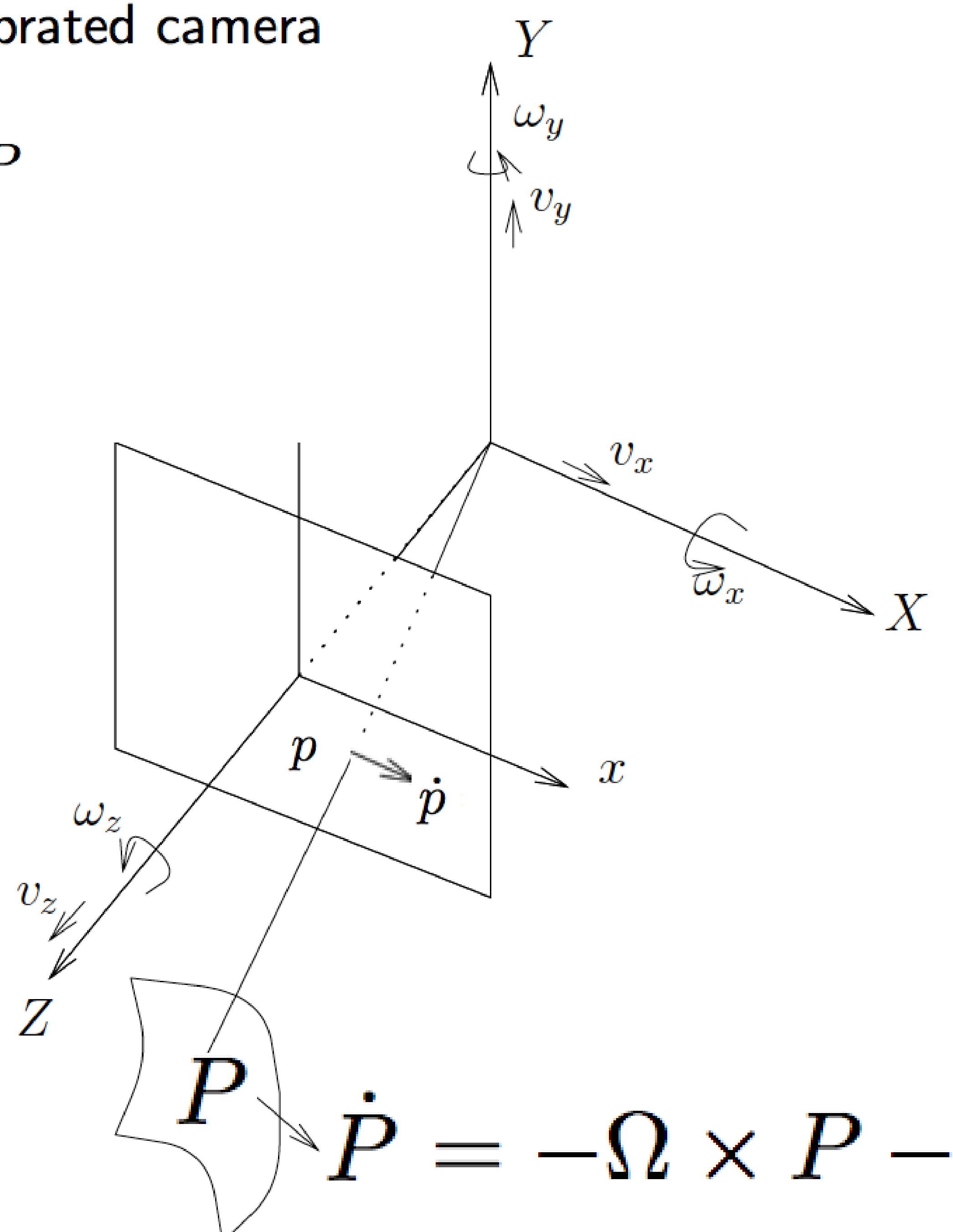
Projection equations for calibrated camera

$x = \frac{X}{Z}$, $y = \frac{Y}{Z}$

or in vector notation $p = \frac{1}{Z}P$

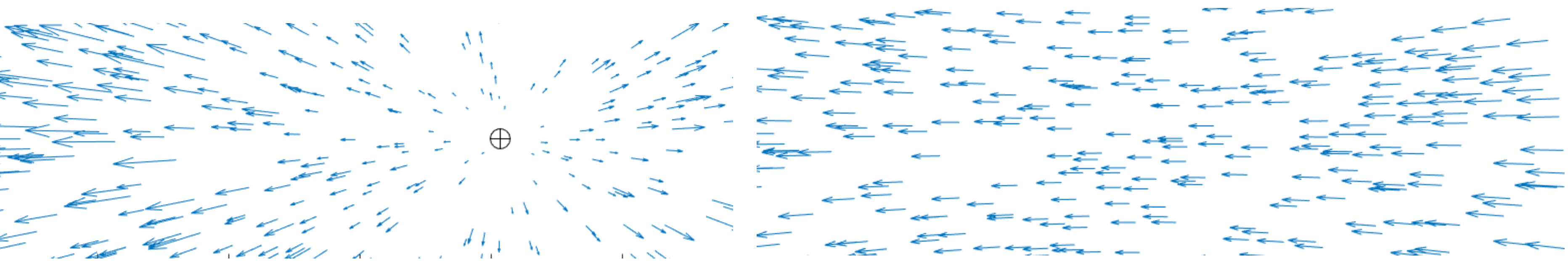
Differentiating w.r.t. time yields:

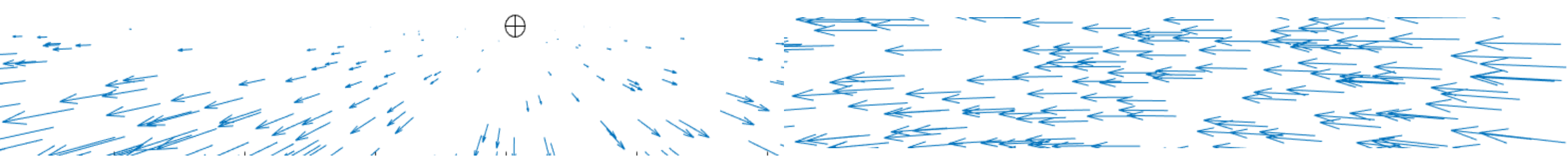
$$\boxed{\dot{p} = \frac{\dot{P}}{Z} - \frac{\dot{Z}}{Z}p}$$



$$\dot{P} = -\Omega \times P -$$

$$\dot{p} = \underbrace{\frac{1}{Z}\begin{bmatrix} xV_z - V_x \\ yV_z - V_y \end{bmatrix}}_{\text{translational flow}} + \underbrace{\begin{bmatrix} xy & -(1+x^2) & y \\ (1+y)^2 & -xy & -x \end{bmatrix}\Omega}_{\text{rotational flow independent of depth}}$$





$$\dot{p} = \underbrace{\frac{1}{Z}\begin{bmatrix} xV_z - V_x \\ yV_z - V_y \end{bmatrix}}_{\text{translational flow}} + \underbrace{\begin{bmatrix} xy & -(1+x^2) & y \\ (1+y)^2 & -xy & -x \end{bmatrix}\Omega}_{\text{rotational flow independent of depth}}$$

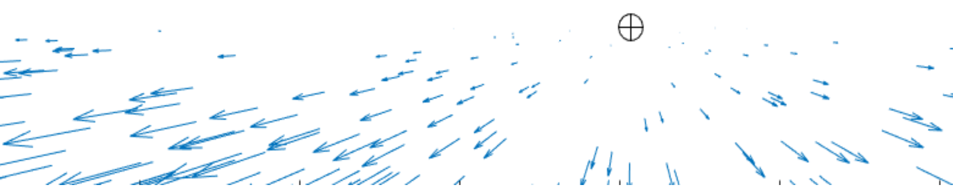
If Z is known, $\dot{p}$ is linear in V and Ω.

Having at least 3 optical flow vectors not on collinear points and corresponding depths we can solve for the 3D velocities from 6 equations.

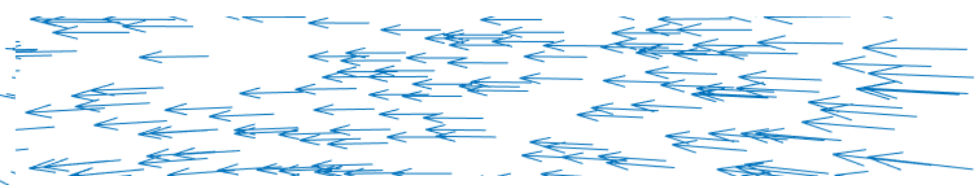
If the field is purely rotational then we have no information about depth.

$$\dot{p} = \underbrace{\begin{bmatrix} xy & -(1+x^2) & y \\ (1+y)^2 & -xy & -x \end{bmatrix} \Omega}_{\text{rotational flow independent of depth}}$$

Translation does not move points at infinity.

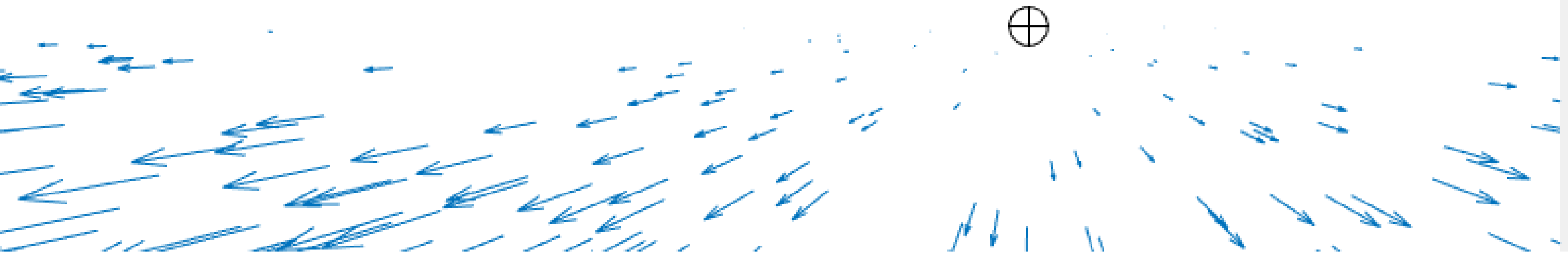


If we look at the horizontal plane points at infinity still rotate.



Translational Flow:

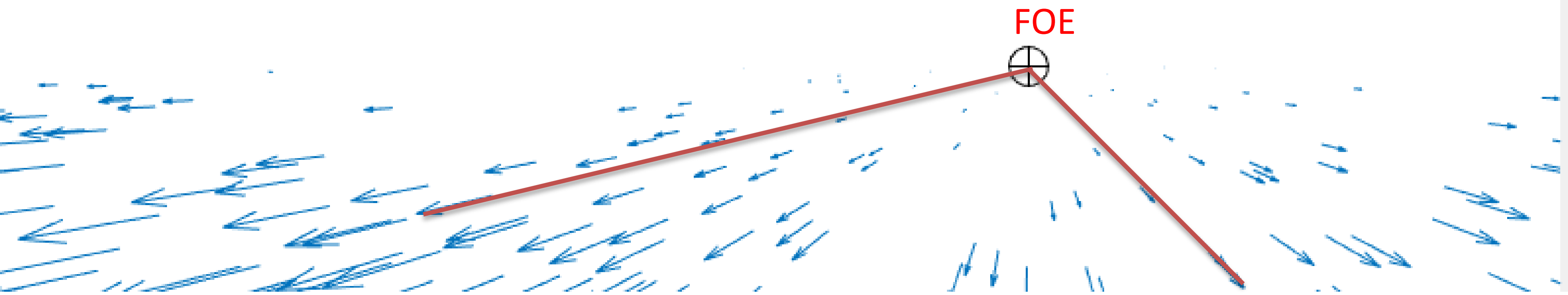
$$\dot{p}_{\text{trans}} = \frac{V_z}{Z}\begin{bmatrix} x - \frac{V_x}{V_z} \\ y - \frac{V_y}{V_z} \end{bmatrix}$$



By intersecting the lines spanned by $\dot{p}_{\text{trans}}$, we can obtain the Focus of Expansion (FOE) also called Epipole

$$FOE = (V_x/V_z, V_y/V_z)$$

FOE can also be at infinity if $V_z = 0$.



Translational Flow:

$$\dot{p}_{\text{trans}} = \frac{V_z}{Z} \begin{bmatrix} x - \frac{V_x}{V_z} \\ y - \frac{V_y}{V_z} \end{bmatrix}$$

The time to collision (which birds and insects estimate) is

$$\frac{Z}{V_z}$$

$$\frac{V_z}{Z} = \frac{||\dot{p}_{\text{trans}}||}{||p - F\vec{O}E||}$$

Points at the same radial distance from FOE have flow vector lengths proportional to inverse depth (or inverse time to collision).

From

$$\dot{p}_{trans}^T (p \times V) = 0$$

we obtain the following coplanarity condition

$$V^T (p \times \dot{p}_{trans}) = 0$$

which says that image point, flow, and linear velocity lie on the same plane.

We can obtain V from two points

$$V \sim (p_1 \times \dot{p_1}) \times (p_2 \times \dot{p_2})$$

From

$$\dot{p}_{trans}^T (p \times V) = 0$$

we obtain the following coplanarity condition

$$V^T (p \times \dot{p}_{trans}) = 0$$

which says that image point, flow, and linear velocity lie on the same plane.

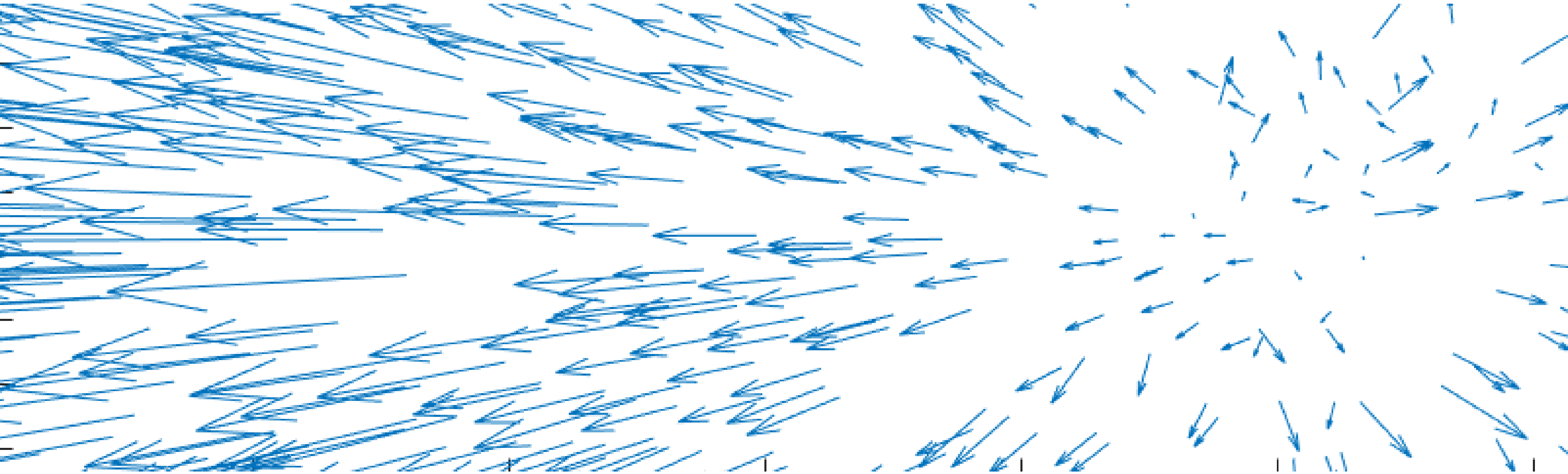
We can obtain V from two points

$$V \sim (p_1 \times \dot{p_1}) \times (p_2 \times \dot{p_2})$$

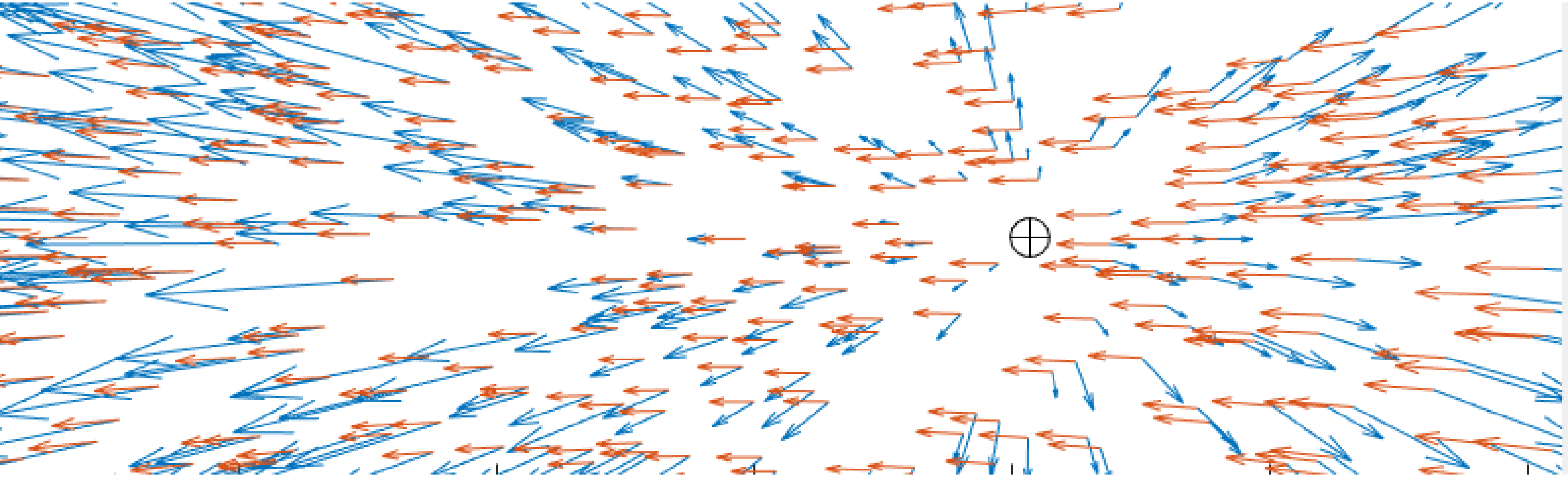
and from n points we obtain a homogeneous system

$$\underbrace{\begin{pmatrix} (p_1 \times \dot{p}_1)^T \\ p_2 \times \dot{p}_2)^T \\ \dots \\ p_n \times \dot{p}_n)^T \end{pmatrix}}_{A} V = 0 \qquad (1)$$

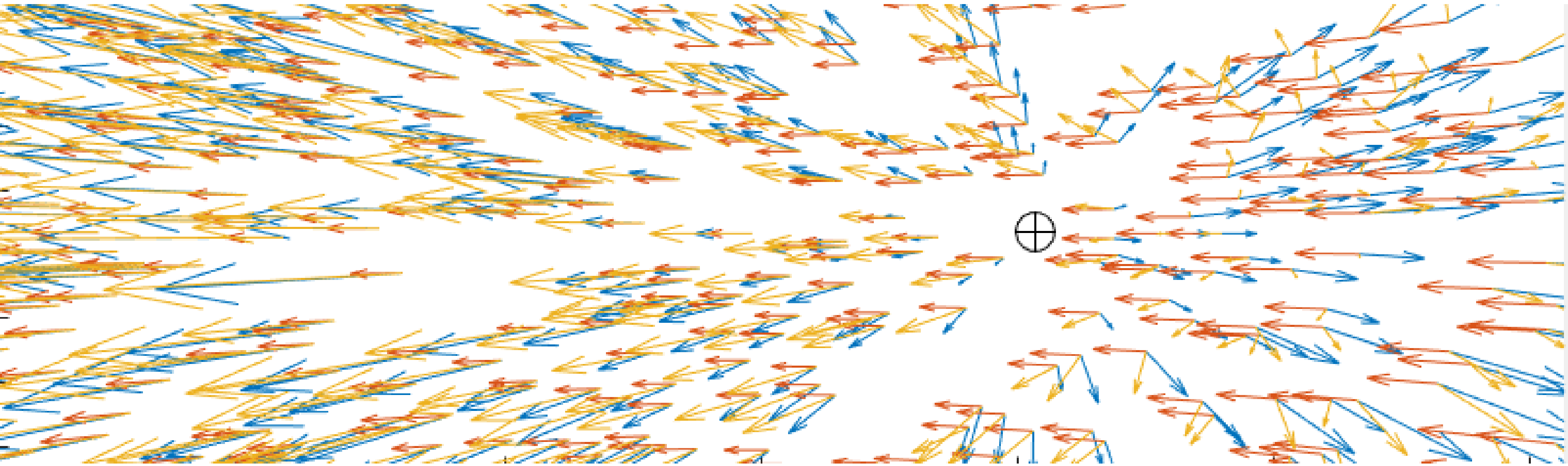
Then V is the nullspace of A which can be obtained from SVD.



.. But how can we split a mixed optical flow field?



.. As addition of two vector fields?



Both V and Ω unknown

Recall that

$$\dot{p} = \frac{1}{Z}F(x,y)V + G(x,y)\Omega$$

This is can be written linearly in inverse depths and Ω:

$$\dot{p} = \left[F(x,y)V \;\; G(x,y)\right] \begin{bmatrix} \frac{1}{Z} \\ \Omega \end{bmatrix}$$

For n points we can write out a system of equations:

$$\begin{pmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dots \\ \dot{p}_n \end{pmatrix} = \Phi(V) \begin{pmatrix} \frac{1}{Z_1} \\ \frac{1}{Z_1} \\ \dots \\ \frac{1}{Z_N} \\ \Omega \end{pmatrix}$$

The Φ matrix is a 2N by (N+3) matrix and is a function of V

$$\dot{d} = \Phi(V) \begin{pmatrix} \frac{1}{Z_1} \\ \frac{1}{Z_1} \\ \dots \\ \frac{1}{Z_N} \\ \Omega \end{pmatrix}$$

The Φ matrix is a 2N by (N+3) matrix and is a function of V

$$\dot{d} = \Phi(V) \begin{pmatrix} \frac{1}{Z_1} \\ \frac{1}{Z_1} \\ \dots \\ \frac{1}{Z_N} \\ \Omega \end{pmatrix}$$

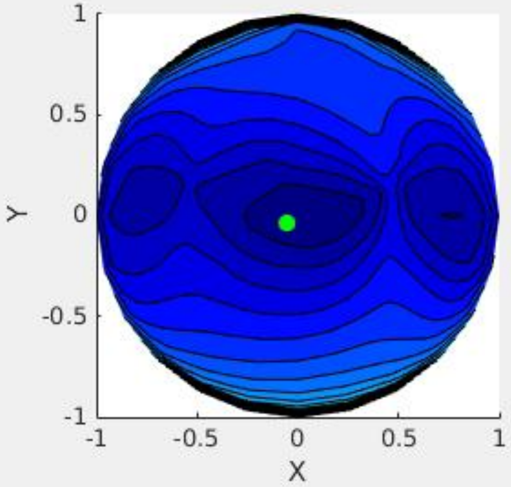
If we solve for the unknown vector of inverse depths and Ω we obtain

$$\Phi^{+}(V)\dot{d}$$

which we can insert back in the objective function.

A search on the sphere yields then V:

$$\arg\min_{V \in S^2} \|\dot{d} - \Phi(V)\Phi(V)^{+}\dot{d}\|^2$$

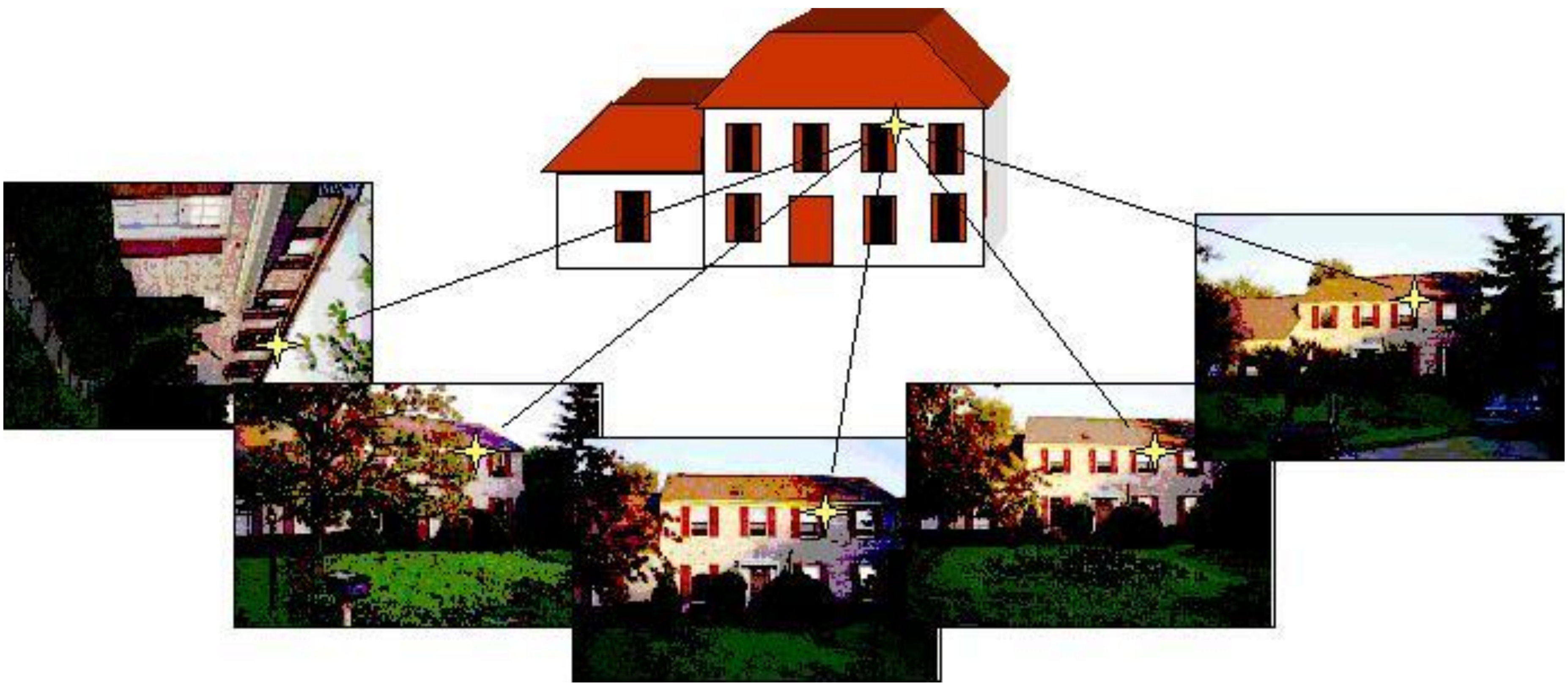


Error function on the sphere of all translation directions (foci of expansion)

Perception: 3D Motion and Structure from Multiple Views or Bundle Adjustment

Kostas Daniilidis

Extract camera poses and structure from multiple views of the same scene

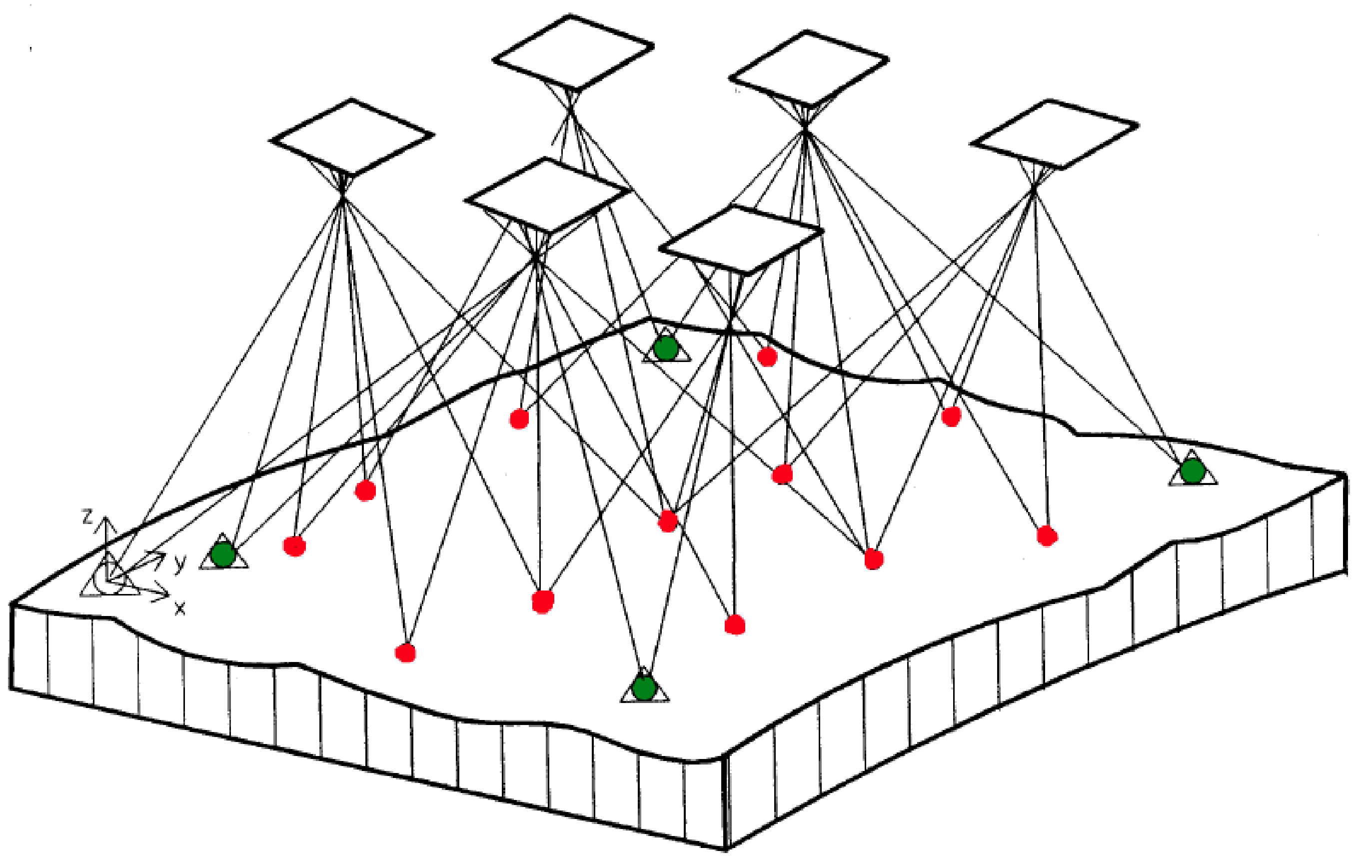


.. and an example closer to us

3D reconstruction

Urbanscape project 2006

„Bündelblockausgleichung“ is an old problem

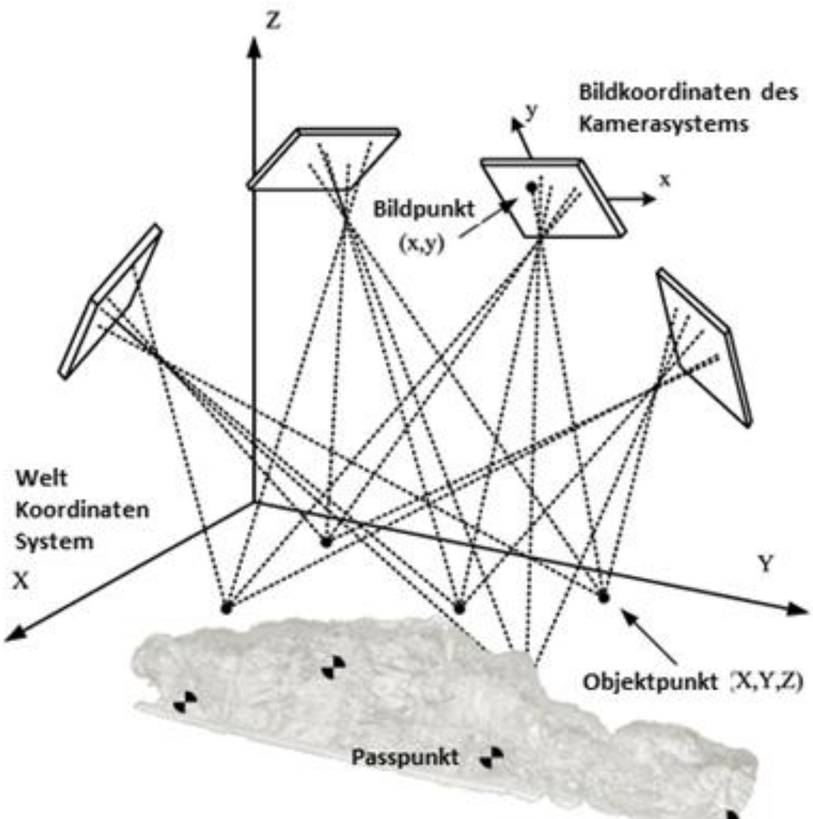


Some times as combination with PnP (resection) if ground control points (green) are known

Figure from photogeo.de

3D model from multiple views

3D-Geofotogrammetrische Aufnahme



Ergänzt nach: ajisaka.entopos.co.id

Ergebnis:
Entzerrtes und skalierbares **3D-Modell**

Given calibrated point projections of $p = 1 \ldots N$ points in $f = 1 \ldots F$ frames $\boxed{(x_p^f, y_p^f)}$

Find the 3D rigid transformation $\boxed{R^f, T^f}$ and the 3D points $\mathbf{X}_p = (X_p, Y_p, Z_p)$ that best satisfy the projection equations

$$\boxed{\begin{aligned} x_p^f &= \frac{R_{11}^f X_p + R_{12}^f Y_p + R_{13}^f Z_p + T_x}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \\ y_p^f &= \frac{R_{21}^f X_p + R_{22}^f Y_p + R_{23}^f Z_p + T_y}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \end{aligned}}$$

Reference frame ambiguity hence we fix the first frame to be the world frame:

$$\boxed{R_1 = I \qquad \text{and} \qquad T_1 = 0}$$

Even with fixing the first frame, a global scale factor is still present. If we multiply all 3D points and T with the same scale measurements do not change.

Hence we have $6(F-1)+3N-1$ independent unknowns

and $2NF$ equations:

$$\begin{aligned} x_p^f &= \frac{R_{11}^f X_p + R_{12}^f Y_p + R_{13}^f Z_p + T_x}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \\ y_p^f &= \frac{R_{21}^f X_p + R_{22}^f Y_p + R_{23}^f Z_p + T_y}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \end{aligned}$$

Reference frame ambiguity hence we fix the first frame to be the world frame:

$$R_1 = I \qquad \text{and} \qquad T_1 = 0$$

Even with fixing the first frame, a global scale factor is still present. If we multiply all 3D points and T with the same scale measurements do not change.

Hence we have $\boxed{6(F-1)+3N-1}$ independent unknowns

and $2NF$ equations:

$$\begin{aligned} x_p^f &= \frac{R_{11}^f X_p + R_{12}^f Y_p + R_{13}^f Z_p + T_x}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \\ y_p^f &= \frac{R_{21}^f X_p + R_{22}^f Y_p + R_{23}^f Z_p + T_y}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \end{aligned}$$

Reference frame ambiguity hence we fix the first frame to be the world frame:

$$R_1 = I \qquad \text{and} \qquad T_1 = 0$$

Even with fixing the first frame, a global scale factor is still present. If we multiply all 3D points and T with the same scale measurements do not change.

Hence we have $6(F-1)+3N-1$ independent unknowns

and $\boxed{2NF \text{ equations:}}$

$$\begin{aligned} x_p^f &= \frac{R_{11}^f X_p + R_{12}^f Y_p + R_{13}^f Z_p + T_x}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \\ y_p^f &= \frac{R_{21}^f X_p + R_{22}^f Y_p + R_{23}^f Z_p + T_y}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \end{aligned}$$

If equations are independent (not always) then

$$\boxed{2NF \geq 6F + 3N - 7}$$

For two frames, it was already known that $N \geq 5$.

For three frames, $N \geq 4$.

Bundle Adjustment is the solution of this problem as nonlinear least-squares:

$$\boxed{\arg\min_{R^f, T^f, X_p} \epsilon^T C^{-1} \epsilon}$$

minimized with respect to all $6(F-1)$ motions and $3N-1$ structure unknowns, where ϵ is the error vector

$$\epsilon^T = \left(\dots \quad x_p^f - \frac{R_{11}^f X_p + R_{12}^f Y_p + R_{13}^f Z_p + T_x}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \quad y_p^f - \frac{R_{21}^f X_p + R_{22}^f Y_p + R_{23}^f Z_p + T_y}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \quad \dots \right.$$

and C is its error covariance. We will continue with the assumption that $C = I$.

Bundle Adjustment is the solution of this problem as nonlinear least-squares:

$$\arg\min_{R^f, T^f, X_p} \epsilon^T C^{-1} \epsilon$$

minimized with respect to all $6(F-1)$ motions and $3N-1$ structure unknowns, where ϵ is the error vector

$$\epsilon^T = \Big(\ldots \quad \boxed{x_p^f - \frac{R_{11}^f X_p + R_{12}^f Y_p + R_{13}^f Z_p + T_x}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z} \quad y_p^f - \frac{R_{21}^f X_p + R_{22}^f Y_p + R_{23}^f Z_p + T_y}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z}} \quad \ldots$$

and C is its error covariance. We will continue with the assumption that $C = I$.

Basics of nonlinear minimization

Call the objective function $\Phi(u) = \epsilon(u)^T \epsilon(u)$.

Given a starting value for the vector of unknowns u we iterate with steps Δu by locally fitting a quadratic function to $\Phi(u)$:

$$\boxed{\Phi(u + \Delta u) = \Phi(u) + \Delta u^T \nabla \Phi(u) + \frac{1}{2} \Delta u^T H(u) \Delta u}$$

where $\nabla \Phi$ is the gradient and H is the Hessian of Φ.

The minimum of this quadratic is at Δu satisfying

$$H \delta u = -\nabla \Phi(u)$$

Basics of nonlinear minimization

Call the objective function $\Phi(u) = \epsilon(u)^T \epsilon(u)$.

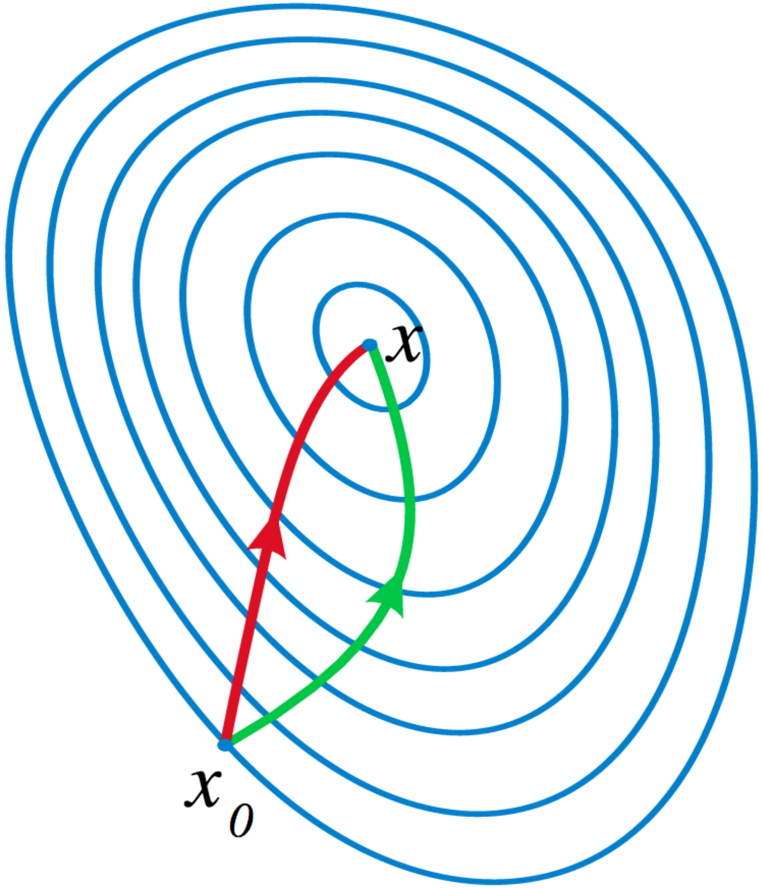
Given a starting value for the vector of unknowns u we iterate with steps Δu by locally fitting a quadratic function to $\Phi(u)$:

$$\Phi(u + \Delta u) = \Phi(u) + \Delta u^T \nabla \Phi(u) + \frac{1}{2} \Delta u^T H(u) \Delta u$$

where $\nabla \Phi$ is the gradient and H is the Hessian of Φ.

The minimum of this quadratic is at Δu satisfying

$$\boxed{H \delta u = -\nabla \Phi(u)}$$



The minimum of this quadratic is at Δu satisfying

$$\boxed{H\delta u = -\nabla\Phi(u)}$$

Vs the green gradient descent iteration.

If $\Phi(u) = \epsilon(u)^T \epsilon(u)$ then

$$\boxed{\nabla\Phi = 2\sum_i \epsilon_i(u)\nabla\epsilon_i(u)^T = J(u)^T\epsilon}$$

where the Jacobian J consists of elements

$$J_{ij} = \frac{\partial \epsilon_i}{\partial u_j}$$

and the Hessian reads

$$H = 2\sum_i \left(\nabla\epsilon_i(u)\nabla\epsilon_i(u)^T + \epsilon_i(u)\frac{\partial^2\epsilon_i}{\partial u^2}\right) = 2\left(J(u)^T J(u) + \sum_i \epsilon_i(u)\frac{\partial^2\epsilon_i}{\partial u^2}\right)$$

by omitting quadratic terms inside the Hessian.

If $\Phi(u) = \epsilon(u)^T \epsilon(u)$ then

$$\nabla\Phi = 2\sum_i \epsilon_i(u)\nabla\epsilon_i(u)^T = J(u)^T\epsilon$$

where the Jacobian J consists of elements

$$J_{ij} = \frac{\partial \epsilon_i}{\partial u_j}$$

and the Hessian reads

$$\boxed{H = 2\sum_i \left(\nabla\epsilon_i(u)\nabla\epsilon_i(u)^T + \epsilon_i(u)\frac{\partial^2\epsilon_i}{\partial u^2}\right) = 2\left(J(u)^T J(u) + \sum_i \epsilon_i(u)\frac{\partial^2\epsilon_i}{\partial u^2}\right)}$$

by omitting quadratic terms inside the Hessian.

If $\Phi(u) = \epsilon(u)^T \epsilon(u)$ then

$$\nabla\Phi = 2\sum_i \epsilon_i(u)\nabla\epsilon_i(u)^T = J(u)^T\epsilon$$

where the Jacobian J consists of elements

$$J_{ij} = \frac{\partial \epsilon_i}{\partial u_j}$$

and the Hessian reads

$$H = 2\sum_i \left(\nabla\epsilon_i(u)\nabla\epsilon_i(u)^T + \epsilon_i(u)\frac{\partial^2\epsilon_i}{\partial u^2}\right) = 2\left(J(u)^T J(u) + \boxed{\sum_i \epsilon_i(u)\frac{\partial^2\epsilon_i}{\partial u^2}}\right)$$

by omitting quadratic terms inside the Hessian.

This yields the Gauss-Newton Iteration

$$\boxed{\Delta u = -(J^T J)^{-1} J^T \epsilon}$$

involving the inversion of a $(6F + 3N - 7) \times (6F + 3N - 7)$ matrix.

Bundle adjustment is about the "art" of inverting efficiently $(J^T J)$.

Let us split the unknown vector $u(a, b)$ into $u = (a, b)$ (following SBA paper by Lourakis):

- $6F - 6$ motion unknowns a
- $3P - 1$ structure unknonws b

and we will explain this case better if we assume two motion unknowns a_1 and a_2 corresponding to 2 frames, and 3 unknown points b_1, b_2, b_3.

For keeping symmetry in writing we do not deal here with the global reference and the global scale ambiguity.

The Jacobian for 2 frames and 3 points has 6 pairs of rows (one pair for each image projection) and 15 columns/unknowns: columns/unknowns:

$$J = \frac{\partial \epsilon}{\partial(a,b)} = \left(\begin{array}{ccccc} A_1^1 & 0 & B_1^1 & 0 & 0 \\ 0 & A_1^2 & B_1^2 & 0 & 0 \\ A_2^1 & 0 & 0 & B_2^1 & 0 \\ 0 & A_2^2 & 0 & B_2^2 & 0 \\ A_3^1 & 0 & 0 & 0 & B_3^1 \\ 0 & A_3^2 & 0 & 0 & B_3^2 \\ \multicolumn{2}{c}{\underbrace{\hphantom{A_3^1 \quad A_3^2}}_{\text{motion}}} & \multicolumn{3}{c}{\underbrace{\hphantom{B_3^1 \quad B_3^2 \quad B_3^3}}_{\text{structure}}} \end{array}\right)$$

with A matrices being 2×6 and B matrices being 2×3 being Jacobians of the error ϵ_i^f of the projection of the i-th point in the f-th frame.

The Jacobian for 2 frames and 3 points has 6 pairs of rows (one pair for each image projection) and 15 columns/unknowns: columns/unknowns:

$$J = \frac{\partial \epsilon}{\partial (a, b)} = \left(\begin{array}{cc|ccc} A_1^1 & 0 & B_1^1 & 0 & 0 \\ 0 & A_1^2 & B_1^2 & 0 & 0 \\ A_2^1 & 0 & 0 & B_2^1 & 0 \\ 0 & A_2^2 & 0 & B_2^2 & 0 \\ A_3^1 & 0 & 0 & 0 & B_3^1 \\ 0 & A_3^2 & 0 & 0 & B_3^2 \\ \multicolumn{2}{c|}{\underbrace{\hphantom{A_1^1 \quad A_1^2}}_{\text{motion}}} & \multicolumn{3}{c}{\underbrace{\hphantom{B_1^1 \quad B_2^1 \quad B_3^1}}_{\text{structure}}} \end{array} \right)$$

with A matrices being 2×6 and B matrices being 2×3 being Jacobians of the error ϵ_i^f of the projection of the i-th point in the f-th frame.

We observe now a pattern emerging

$$J^T J = \begin{pmatrix} U^1 & 0 & W_1^1 & W_2^1 & W_3^1 \\ 0 & U^2 & W_1^2 & W_2^2 & W_3^3 \\ .. & .. & V_1 & 0 & 0 \\ .. & .. & 0 & V_2 & 0 \\ .. & .. & 0 & 0 & V_3 \end{pmatrix}$$

with the block diagonals for motion and structure separated.

We observe now a pattern emerging

$$
J^T J = \begin{pmatrix}
U^1 & 0 & W_1^1 & W_2^1 & W_3^1 \\
0 & U^2 & W_1^2 & W_2^2 & W_3^3 \\
.. & .. & V_1 & 0 & 0 \\
.. & .. & 0 & V_2 & 0 \\
.. & .. & 0 & 0 & V_3
\end{pmatrix}
$$

with the block diagonals for motion and structure separated.

Let us rewrite the basic iteration

$$\Delta u = -(J^T J)^{-1} J^T \epsilon$$

as

$$\boxed{\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} = \begin{pmatrix} \epsilon'_a \\ \epsilon'_b \end{pmatrix}}$$

and premultiply with

$$\begin{pmatrix} I & WV^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} = \begin{pmatrix} I & WV^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} \epsilon'_a \\ \epsilon'_b \end{pmatrix}$$

Let us rewrite the basic iteration

$$\Delta u = -(J^T J)^{-1} J^T \epsilon$$

as

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} = \begin{pmatrix} \epsilon'_a \\ \epsilon'_b \end{pmatrix}$$

and premultiply with

$$\boxed{\begin{pmatrix} I & WV^{-1} \\ 0 & I \end{pmatrix}} \begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} = \boxed{\begin{pmatrix} I & WV^{-1} \\ 0 & I \end{pmatrix}} \begin{pmatrix} \epsilon'_a \\ \epsilon'_b \end{pmatrix}$$

Let us rewrite the basic iteration

$$\Delta u = -(J^T J)^{-1} J^T \epsilon$$

as

$$\begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} = \begin{pmatrix} \epsilon'_a \\ \epsilon'_b \end{pmatrix}$$

and premultiply with

*

$$\begin{pmatrix} I & WV^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} U & W \\ W^T & V \end{pmatrix} \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} = \begin{pmatrix} I & WV^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} \epsilon'_a \\ \epsilon'_b \end{pmatrix}$$

=0

Motion parameters can be updated separately by inverting a $6F \times 6F$ matrix:

$$\boxed{(U - WV^{-1}W^T)\Delta a = \epsilon'_a - WV^{-1}\epsilon'_b}$$

Motion parameters can be updated separately by inverting a $6F \times 6F$ matrix:

$$(U - WV^{-1}W^T)\Delta a = \epsilon'_a - WV^{-1}\epsilon'_b$$

Each 3D point can be be updated separately by inverting a $3\times$ 3 matrix V:

$$\boxed{V\Delta b = \epsilon'_b - W^T\Delta a}$$

If a point i does not appear in frame f then matrices A_i^f and B_i^f are set to zero.

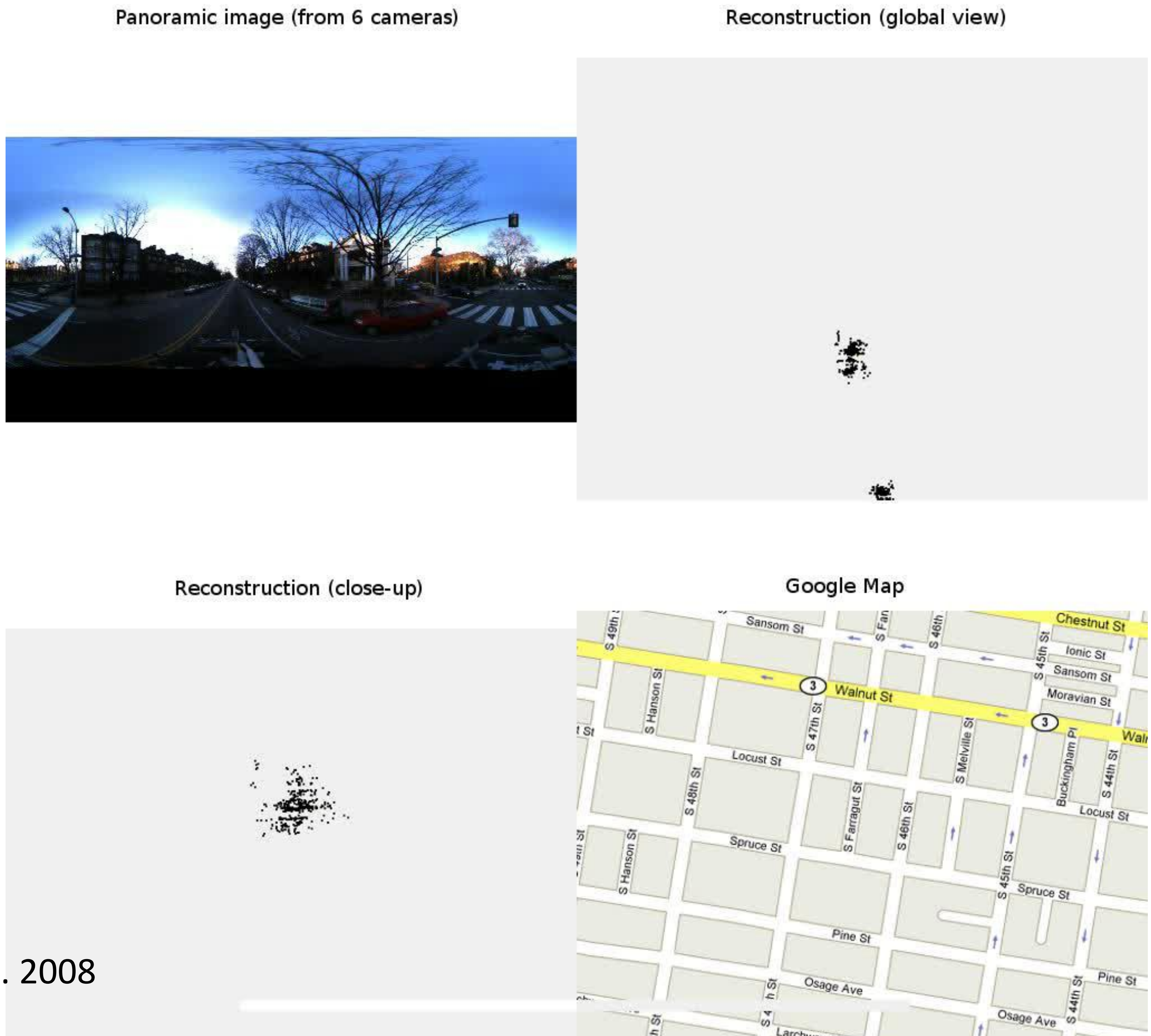
Bundler©   Structure from Motion for Unordered Image Collections

We will see how it will be used in Visual Odometry as well !

Perception: Visual Odometry

Kostas Daniilidis

Extract camera trajectory from video



UPenn,
Tardif et al. 2008

What is Odometry ?

- Measuring how far you go by counting wheel rotations or steps.
- Known as “path integration” in biological perception.
- More general, integration of velocity or acceleration measurements: inertial odometry.

What is Visual Odometry ?

The process of incrementally estimating your position and orientation with respect to an initial reference frames by tracking visual features.

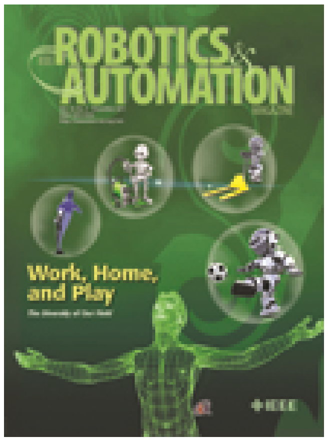
What is the difference to multiple views/bundle adjustment in the last lecture?

- Bundle adjustment can have large baselines, different cameras, sparse viewpoints
- Visual odometry is based on video and needs to be incremental
- Video allows a motion model

What is the difference to visual SLAM (Simultaneous Localization and Mapping) ?

- Used interchangeably but visual SLAM produces also map of features while visual odometry focuses on the camera trajectory.

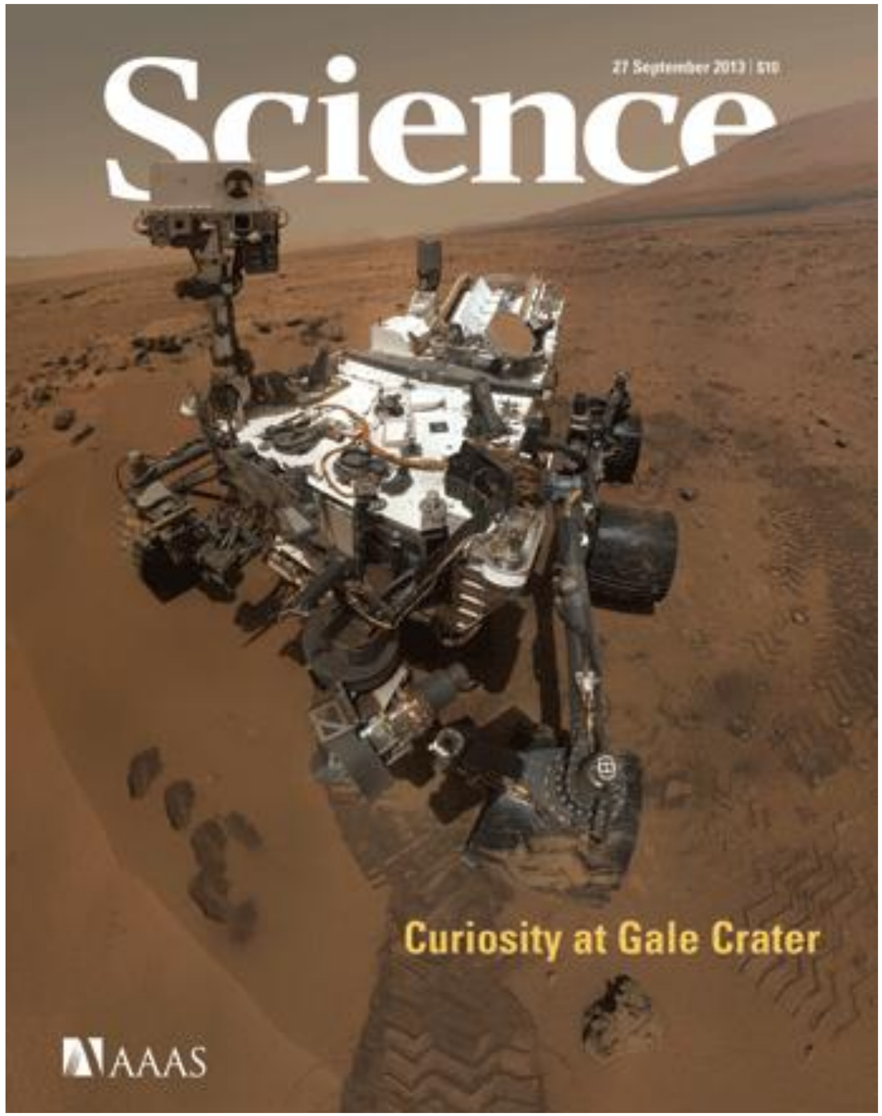
New field, not in textbooks, but good reference tutorial



- Scaramuzza, D., Fraundorfer, F., Visual Odometry: Part I - The First 30 Years and Fundamentals, IEEE Robotics and Automation Magazine, Volume 18, issue 4, 2011.
- Fraundorfer, F., Scaramuzza, D., Visual Odometry: Part II - Matching, Robustness, and Applications, IEEE Robotics and Automation Magazine, Volume 19, issue 1, 2012.

The Future of Real-Time SLAM: 18th December 2015 (ICCV Workshop)

Visual odometry on the MARS



Dyson 360 (Andrew Davison)

Multiple views setting:

Given calibrated point projections of $p = 1 \dots N$ points in $f = 1 \dots F$ frames (x_p^f, y_p^f)

Find the 3D rigid transformation R^f, T^f and the 3D points $\mathbf{X}_p = (X_p, Y_p, Z_p)$ that best satisfy the projection equations

Multiple views setting:

Given calibrated point projections of $p = 1 \dots N$ points in $f = 1 \dots F$ frames (x_p^f, y_p^f)

Find the 3D rigid transformation R^f, T^f and the 3D points $\mathbf{X}_p = (X_p, Y_p, Z_p)$ that best satisfy the projection equations

Visual Odometry:

Given **an estimate** R_k, T_k of the current camera pose as well as the 3D points $\mathbf{X}_p = (X_p, Y_p, Z_p)$ and correspondences to calibrated point projections in frame $(k+1)$ (x_p^{k+1}, y_p^{k+1})

Update to the pose R_{k+1}, T_{k+1}

Monocular visual odometry will leave an unknown global scale.

Update step for rotation:

- Find correspondences from view k to view $k+1$ using RANSAC and 5-point algorithm.
- Solve for epipolar geometry between two views k and $k+1$ using all inliers
- Use the rotation estimate ${}^{k}R_{k+1}$ to update the rotational pose

$$R_{k+1} = R_k \; {}^{k}R_{k+1}$$

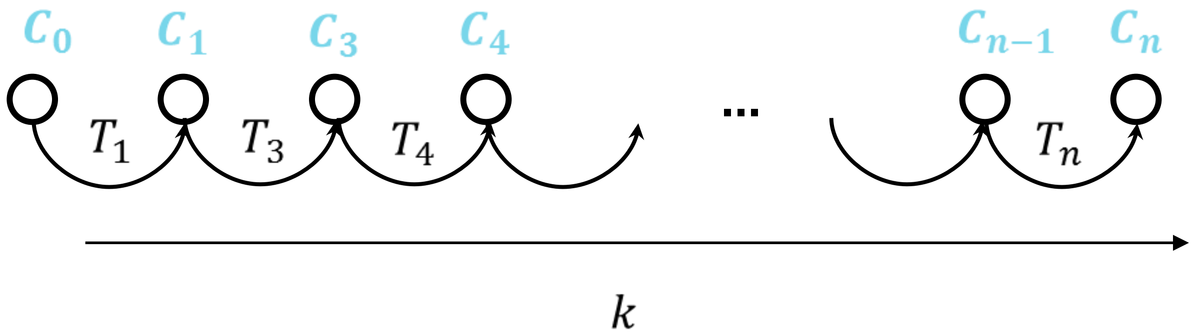
- If we use this for translation, we do not know the scale of ${}^{k}T_{k+1}$:

$$T_{k+1} = T_k + R_k \; {}^{k}T_{k+1}$$

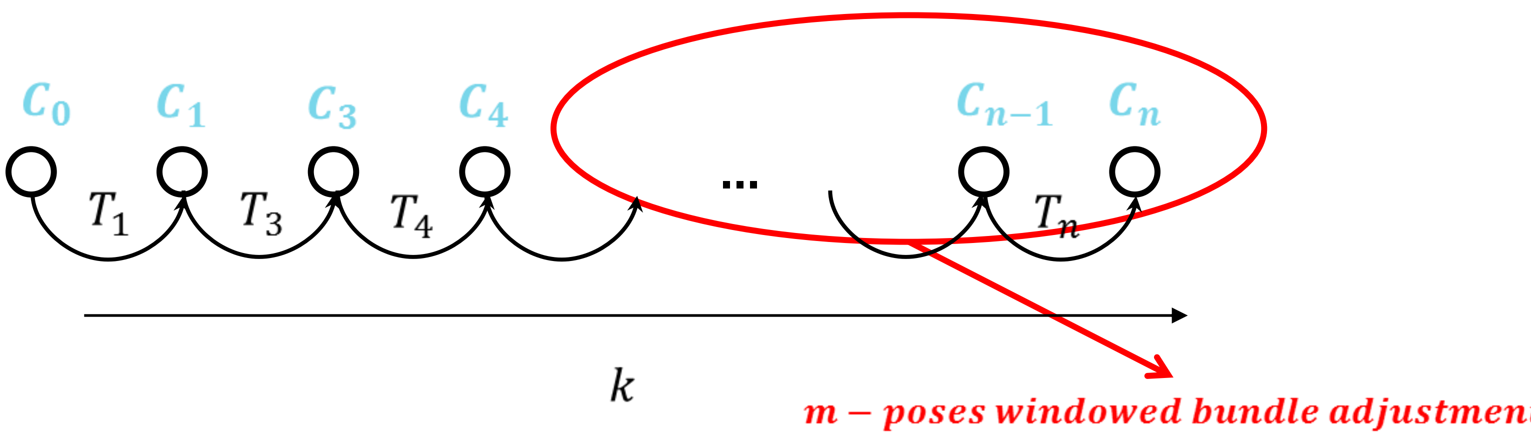
Update step for translation and structure:

- Use the estimated 3D points $\mathbf{X}_p$ and their projection correspondences (x_p^{k+1}, y_p^{k+1}) to update translational pose R_{k+1}, T_{k+1} using 2D-3D pose algorithms (usually only the translation is updated).
- Update the estimates of the 3D points $\mathbf{X}_p$

Main cycle of visual odometry



To minimize drift we run bundle adjustment over a window

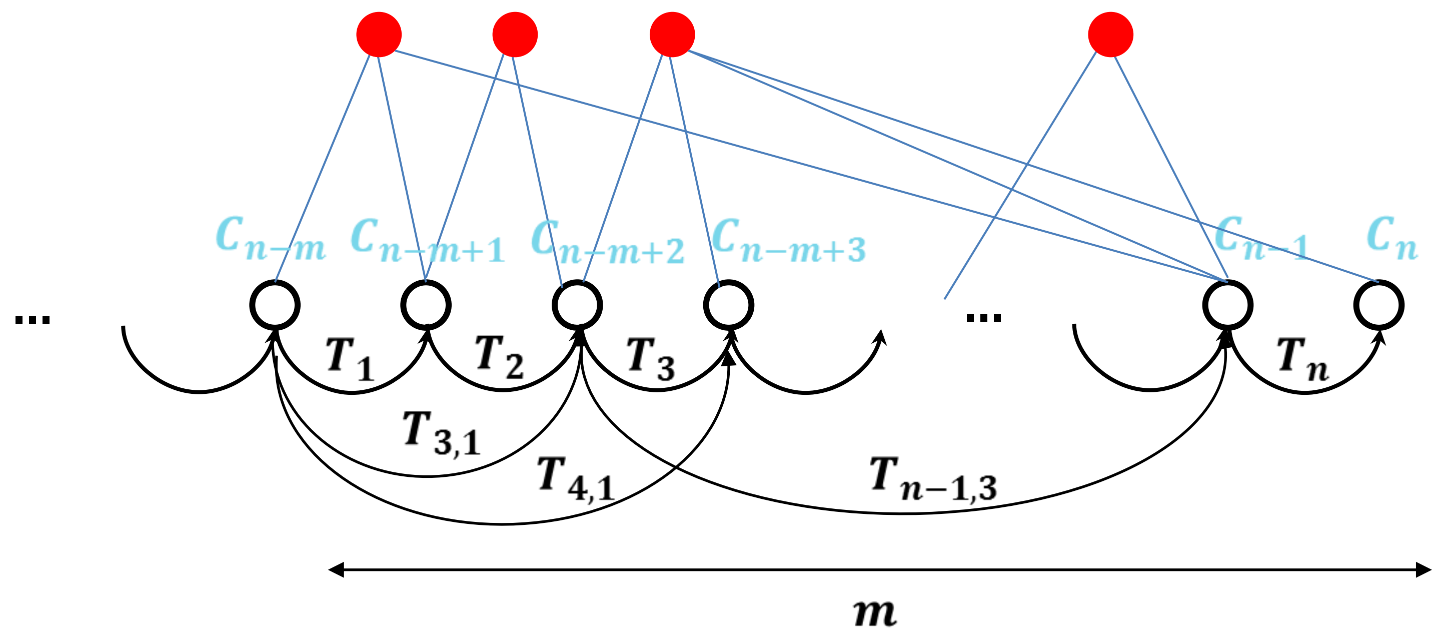


Use the Bundle Adjustment equations?

$$\arg\min_{X^i, C_k} \sum_{i,k} \| p_k^i - g(X^i, C_k) \|^2$$

$$x_p^f = \frac{R_{11}^f X_p + R_{12}^f Y_p + R_{13}^f Z_p + T_x}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z}$$

$$y_p^f = \frac{R_{21}^f X_p + R_{22}^f Y_p + R_{23}^f Z_p + T_y}{R_{31}^f X_p + R_{32}^f Y_p + R_{33}^f Z_p + T_z}$$



State vector consists of all points which remain fixed in the global frame (some approaches use the projections and inverse depths as state) as well as the poses and velocities .

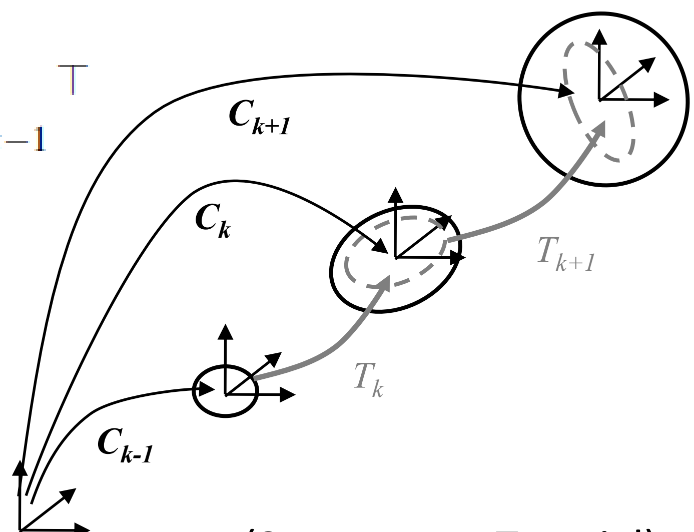
$$
\begin{aligned}
\mathbf{X}_p^{k+1} &= \mathbf{X}_p^k \\
R^{k+1} &= e^{\hat{\omega}^k} R^k \\
\omega^{k+1} &= \omega^k \\
T^{k+1} &= T^k + R^k v^k \\
v^{k+1} &= v^k
\end{aligned}
$$

Error Propagation

State Covariance is an estimate of its uncertainty. If uncertainty is Gaussian it can be visualized as an ellipsoid.

Its update depends on the previous uncertainty Σ_{k-1}, the measurement uncertainty $\Sigma_{k,k-1}$, and the Jacobian with respect to state J.

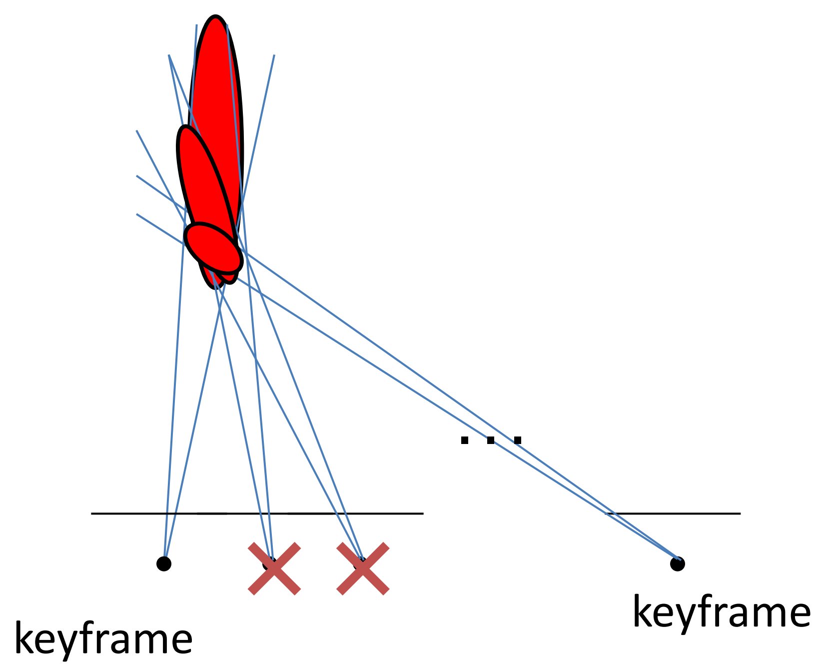
$$\begin{aligned}\Sigma_k &= J \begin{bmatrix} \Sigma_{k-1} & 0 \\ 0 & \Sigma_{k,k-1} \end{bmatrix} J^\top \\ &= J_{\vec{C}_{k-1}} \Sigma_{k-1} J_{\vec{C}_{k-1}}{}^\top + J_{\vec{T}_{k,k-1}} \Sigma_{k,k-1} J_{\vec{T}_{k,k-1}}{}^\top\end{aligned}$$



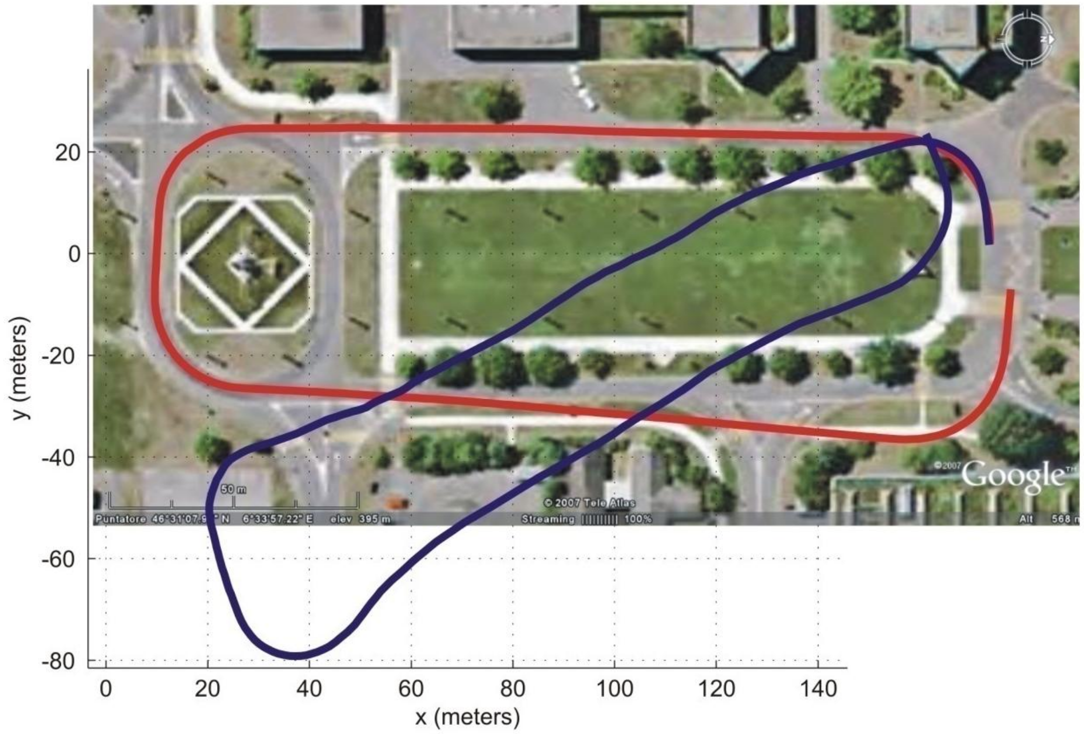
(Scaramuzza Tutorial)

Triangulation and Keyframe Selection

- Pose (translation) update depends on triangulated points whose error depends on baseline and distance.
- Wait until error in 3D triangulation decreases and then update pose: keyframe



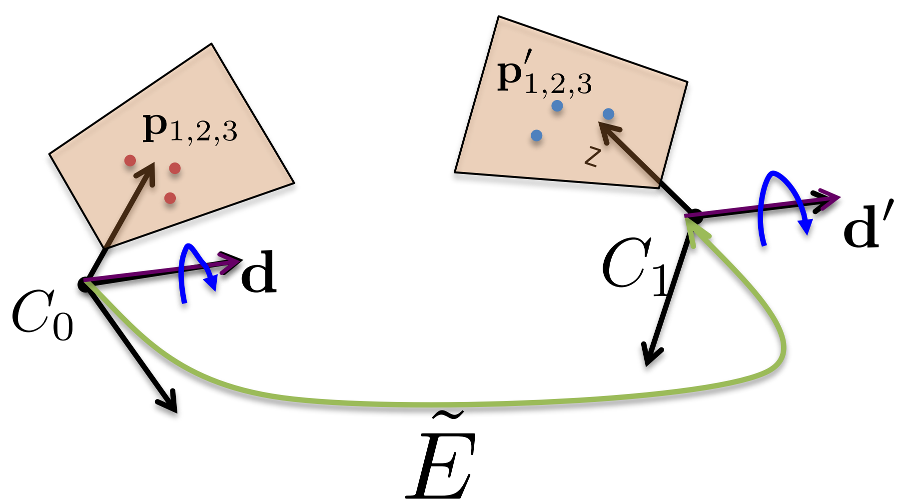
Outliers in VO: before and after



- While keyframe selection reduces drift, a large factor is the good inlier selection in point correspondences (Nister 2004, 5-point algorithm)

The 3-pt algorithm

- Gravity or a point at infinity constrains 2 DOF of the rotation, 3 DOF remaining ("yaw" and two for translation)
- First align each directional vector with the y-axis
  - Only rotation around the y-axis remains
- New 3-DOF epipolar constraint is $\mathbf{p}_i'^{\top} \tilde{E} \mathbf{p}_i = 0$



Formulating the 3pt Problem

- Parameterize the essential matrix

$$\tilde{E} = \hat{\tilde{\mathbf{t}}}(I + \sin\theta\hat{\mathbf{e}}_2 + (1 - \cos\theta)\hat{\mathbf{e}}_e^2)$$

- 4 unknowns $\tilde{\mathbf{t}} = [x, y, 1]^\top$, $\sin\theta$ and $\cos\theta$
- To make a polynomial system, let

$$c = \cos\theta \qquad s = \sin\theta$$

- Add the trigonometric constraint

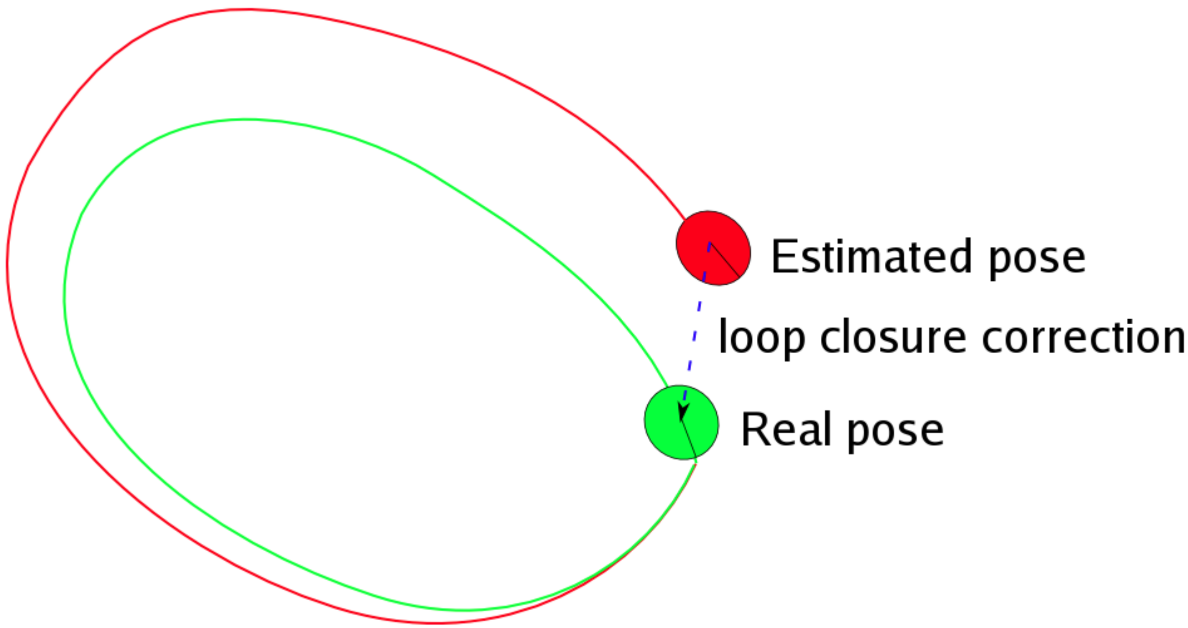
$$s^2 + c^2 - 1 = 0$$

- Result: 4 polynomial equations in 4 unknowns

Naroditsky, et. al, *Two Efficient Solutions for Visual Odometry Using Directional Correspondence*, PAMI 2012

Visual loop closing

Angelis et al. 2008



TWO STEPS:

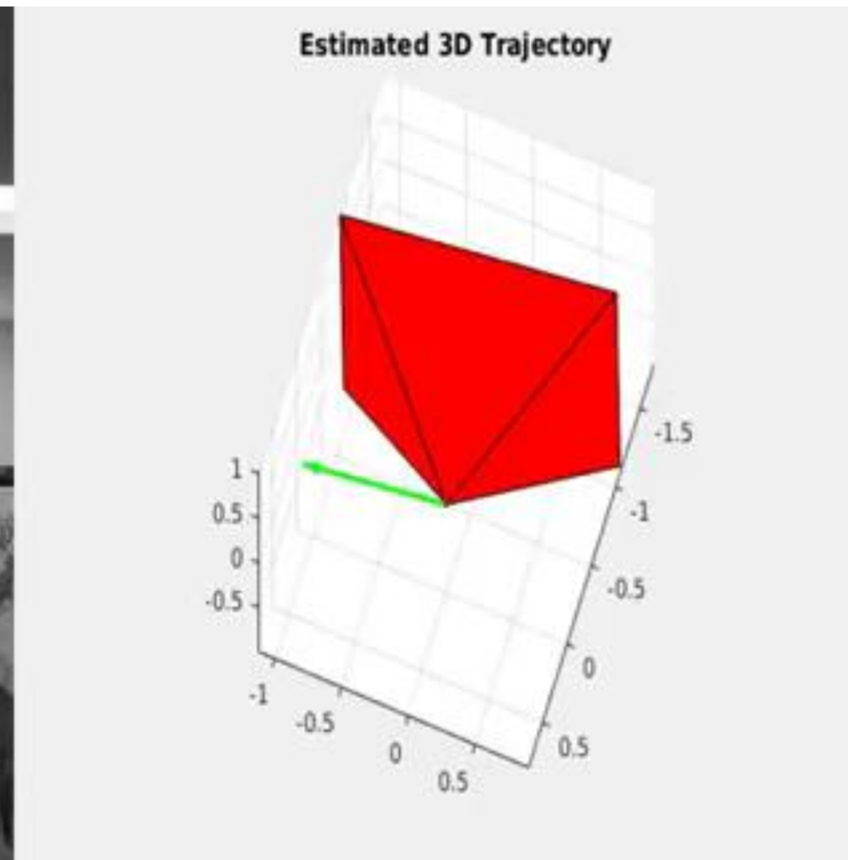
- Search for the closest VISITED image using feature retrieval (vocabulary trees)
- Geometric consistency with epipolar constraint.

Summary of Visual Odometry Tools

- Bundle Adjustment over a window
- Keyframe Selection
- RANSAC for 5-points or reduced minimal problem with 3 points.
- Visual Closing to produce unique trajectories when places are revisited

Integration with IMU (Inertial Measurement Unit)

- Acceleration measurements make the unknown monocular scale observable!
- State vector is augmented with the unknown bias in the acceleration and angular velocity IMU measurements.

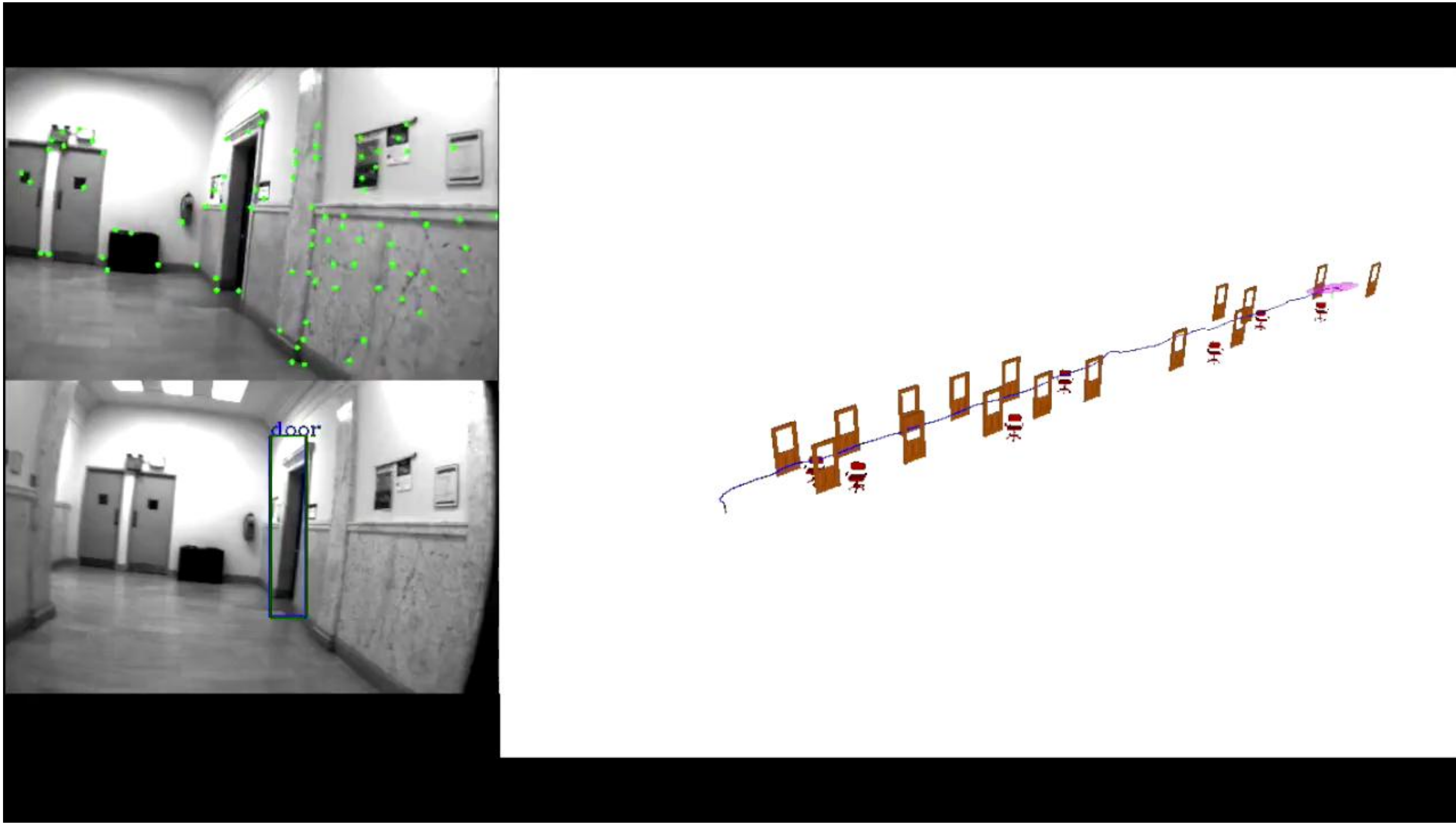


Libviso (Geiger at al. 2012)

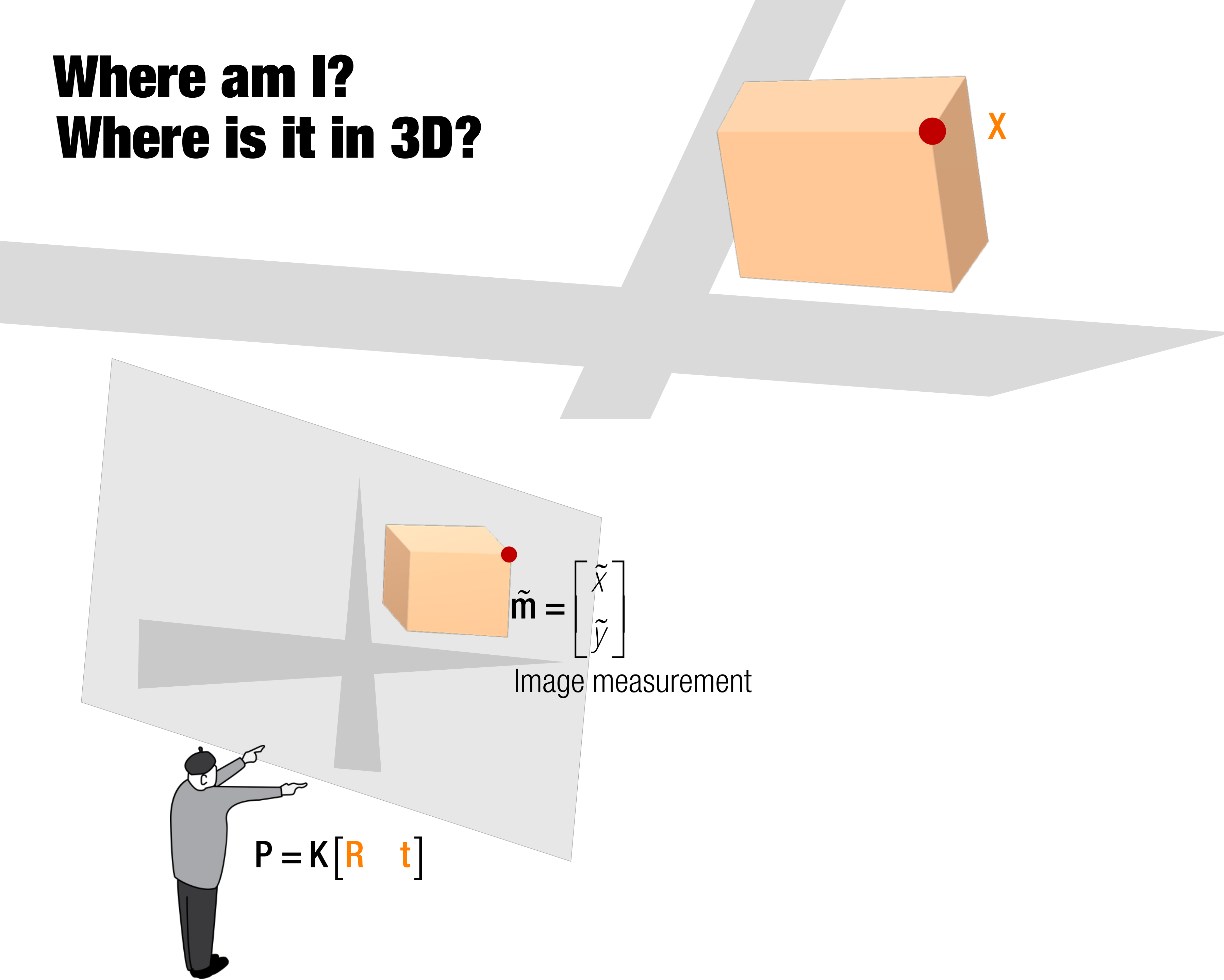
Semi Direct Visual Odometry (Forster et al. 2014)

Google Project Tango

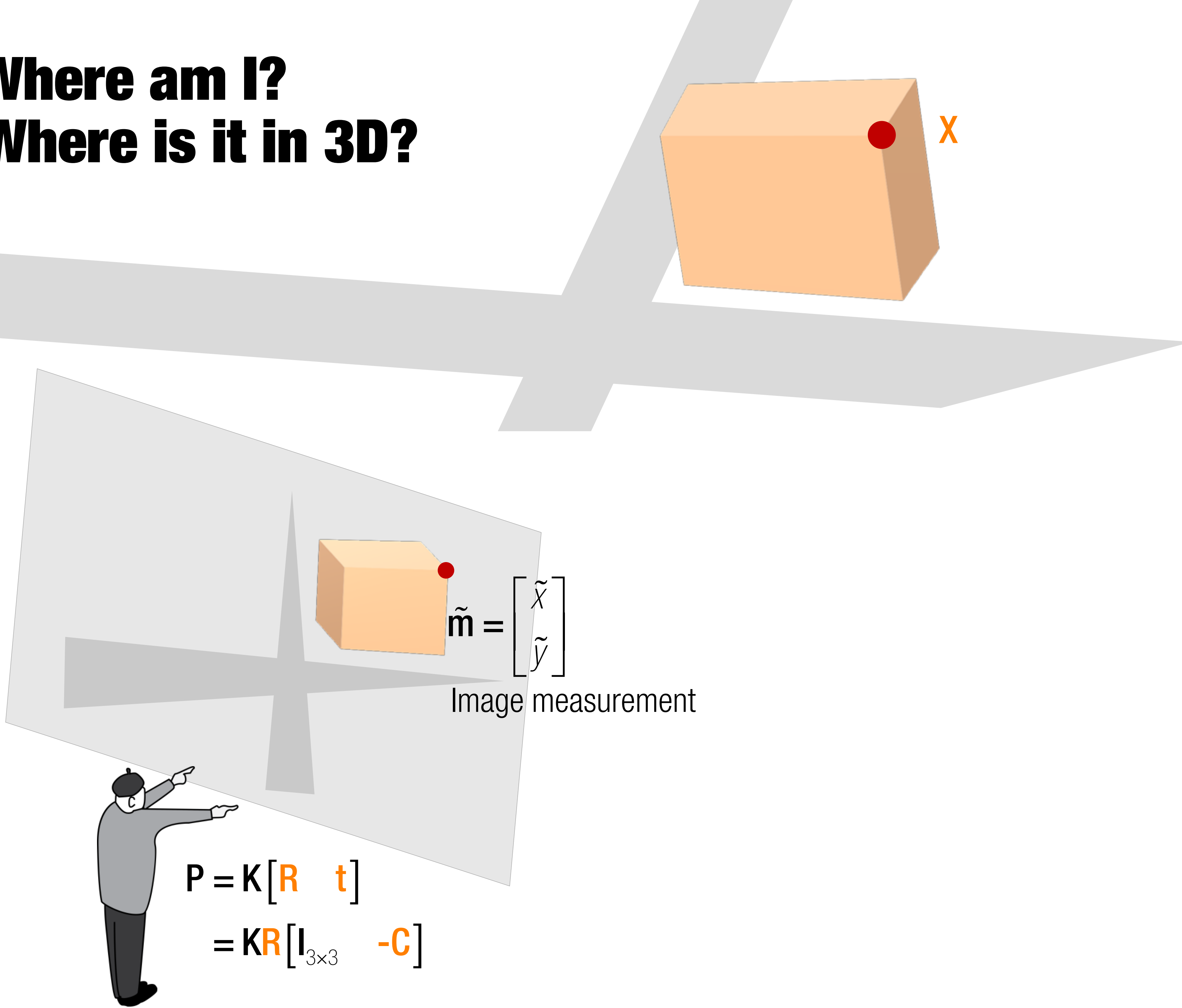
The Future of visual SLAM: semantic visual inertial navigation (Bowman et al. 2016)



Where am I?
Where is it in 3D?



Where am I?
Where is it in 3D?

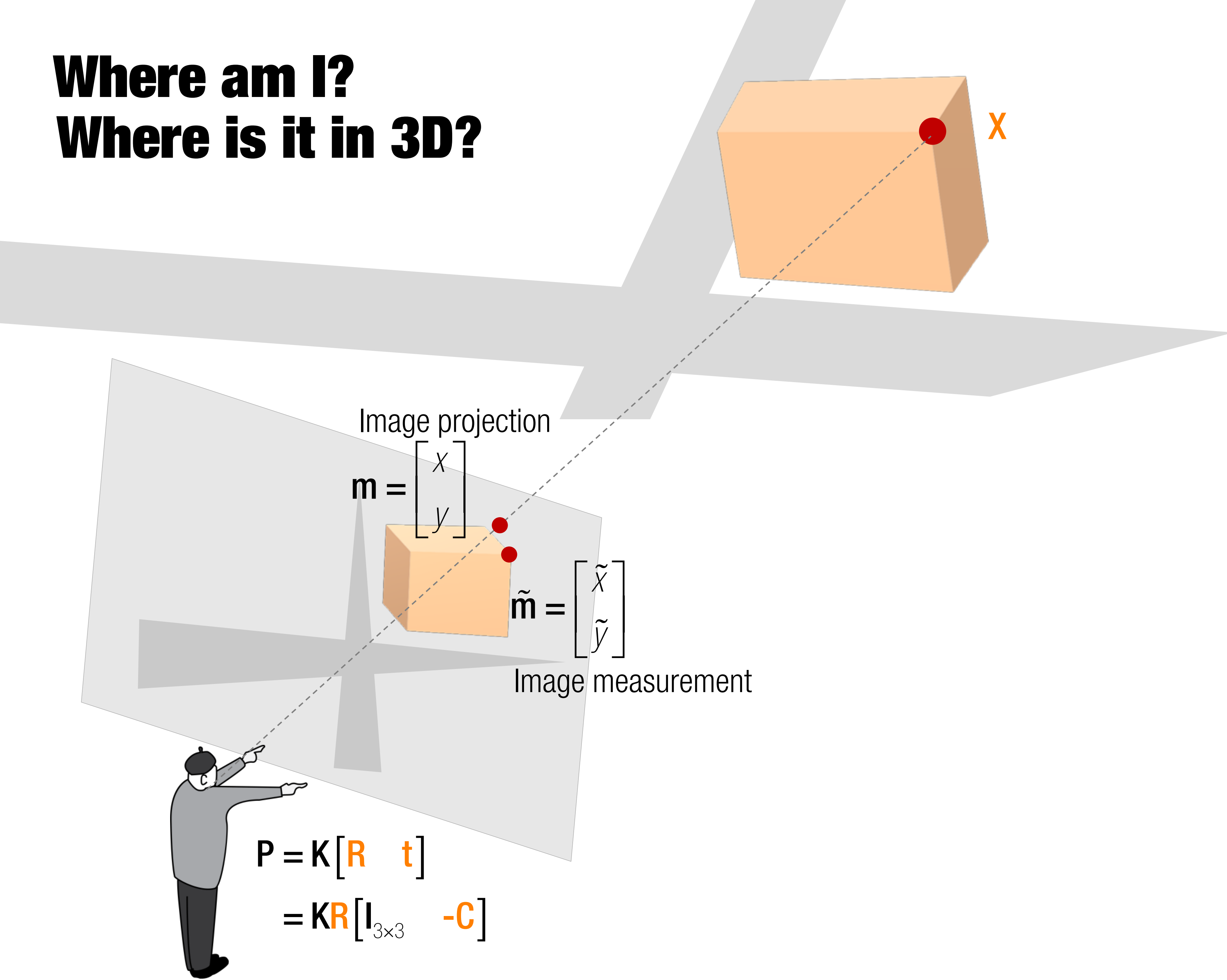


$$\tilde{\mathbf{m}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}$$

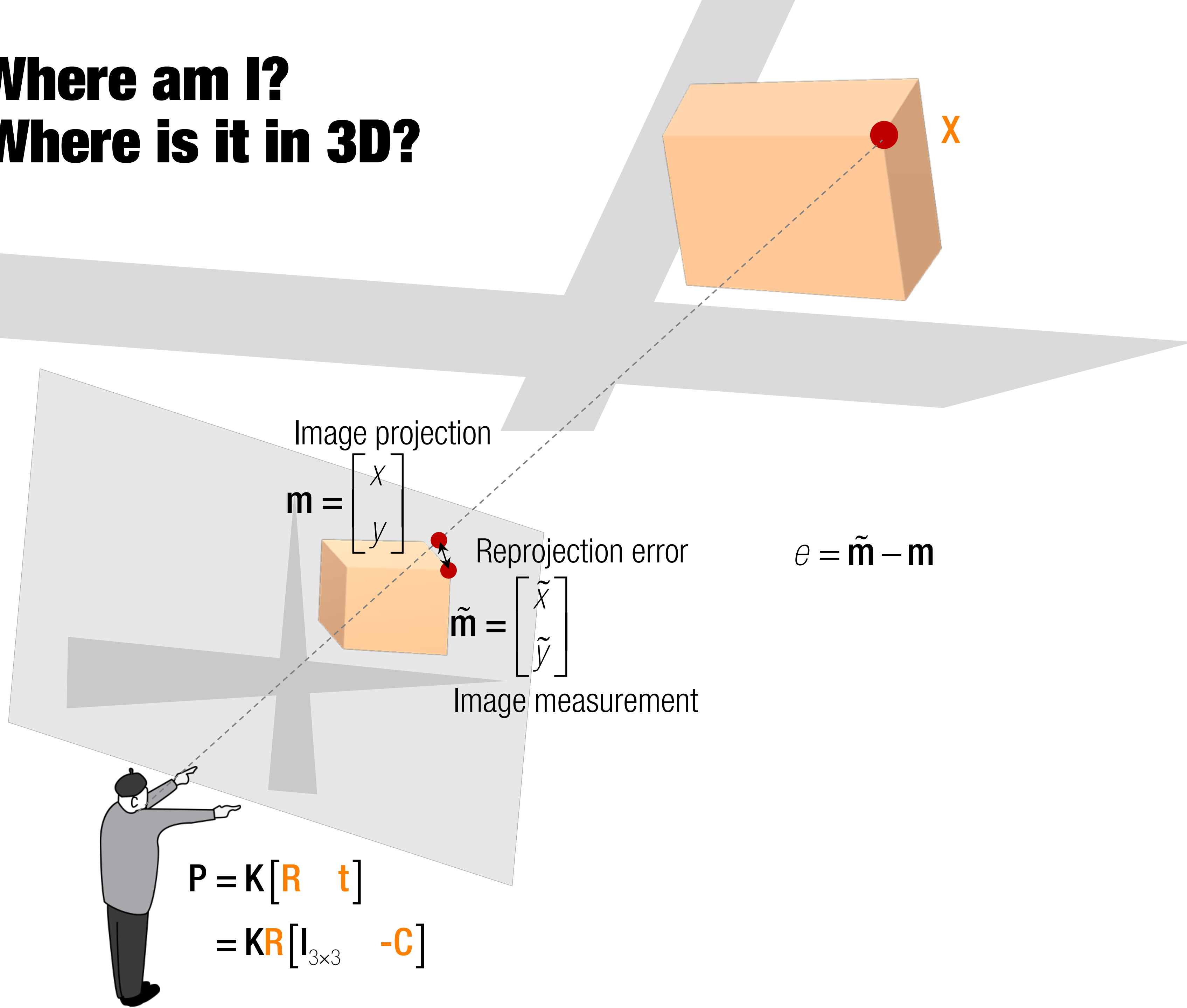
$$\mathbf{P} = \mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$$

$$= \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}$$

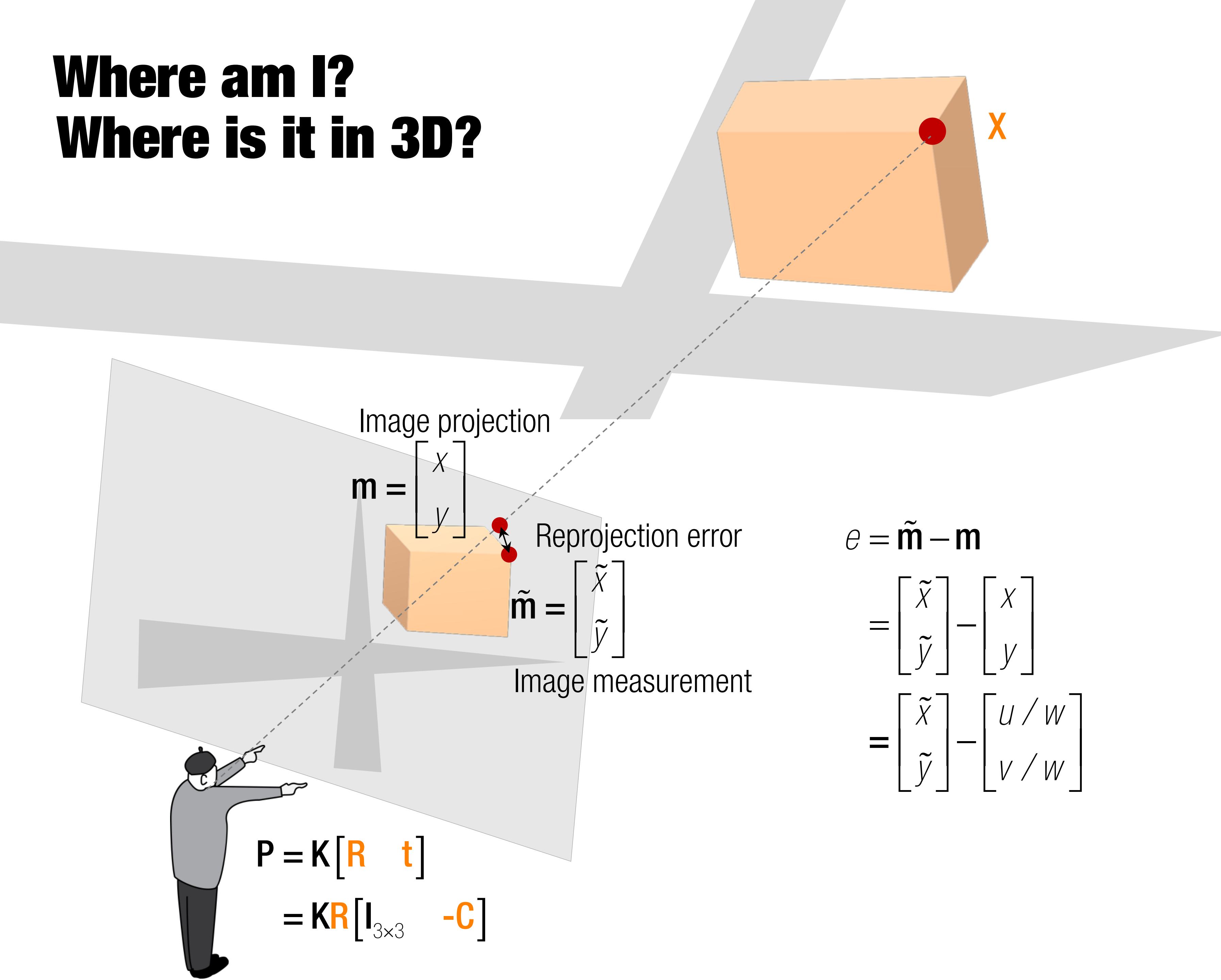
Where am I?
Where is it in 3D?



Where am I?
Where is it in 3D?



Where am I?
Where is it in 3D?



$$\mathbf{m} = \begin{bmatrix} x \\ y \end{bmatrix}$$

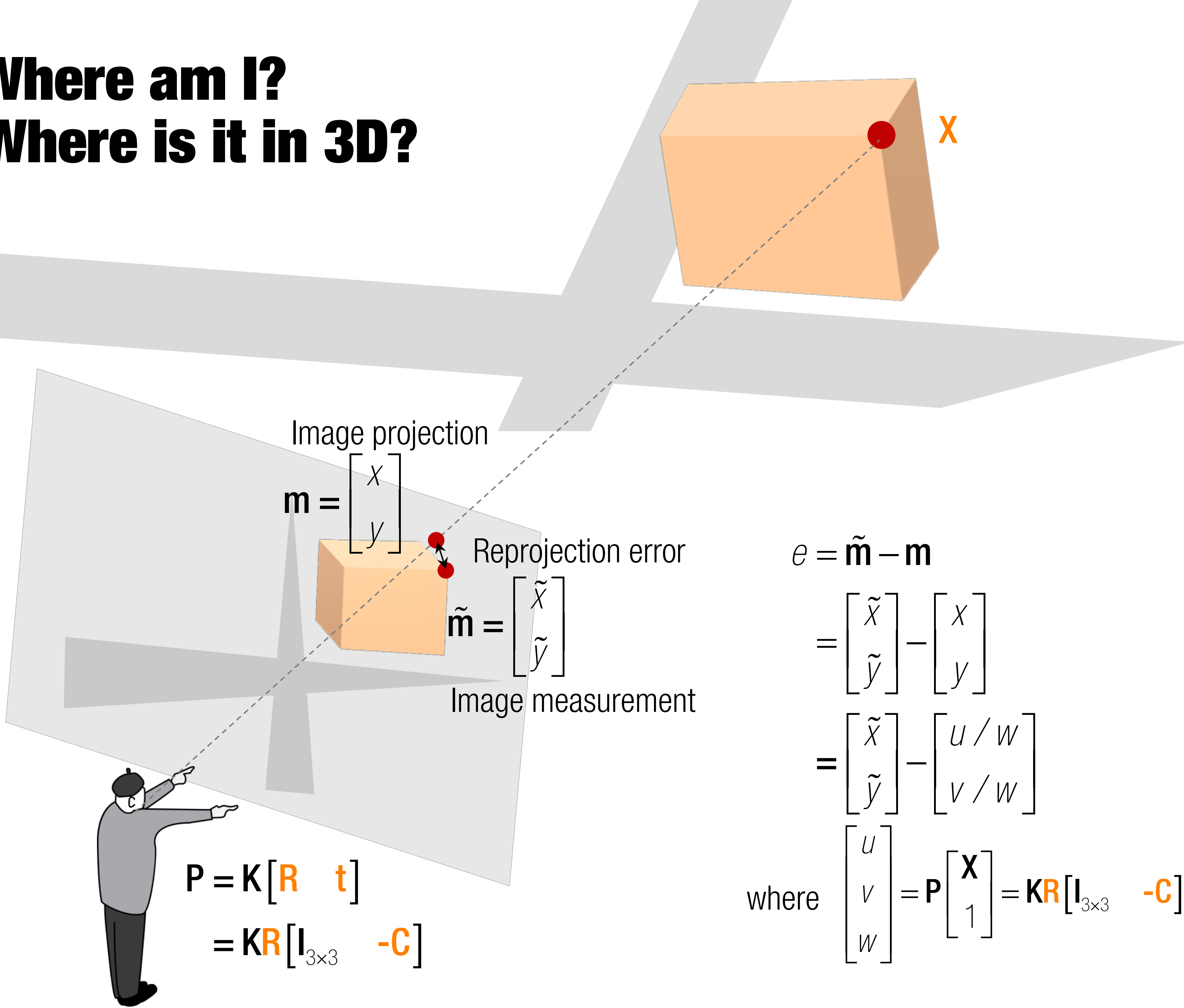
$$\tilde{\mathbf{m}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}$$

$$\mathbf{P} = \mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$$

$$= \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}$$

$$\begin{aligned} e &= \tilde{\mathbf{m}} - \mathbf{m} \\ &= \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \\ &= \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u / w \\ v / w \end{bmatrix} \end{aligned}$$

Where am I?
Where is it in 3D?



$$\mathbf{m} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\tilde{\mathbf{m}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}$$

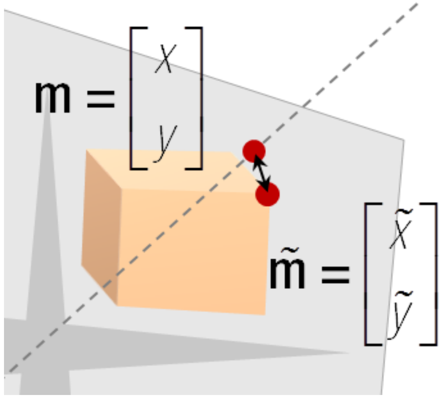
$$\mathbf{P} = \mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$$
$$= \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{-C} \end{bmatrix}$$

$$\begin{aligned} e &= \tilde{\mathbf{m}} - \mathbf{m} \\ &= \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \\ &= \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \end{aligned}$$

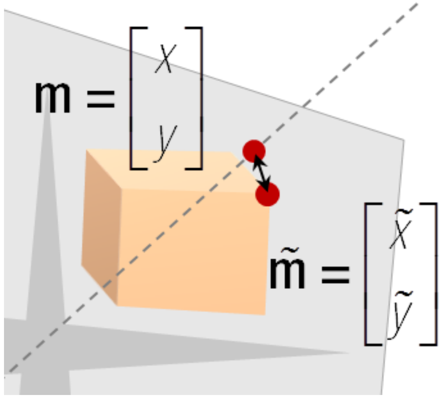
where $\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{-C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$

Reprojection error

$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$



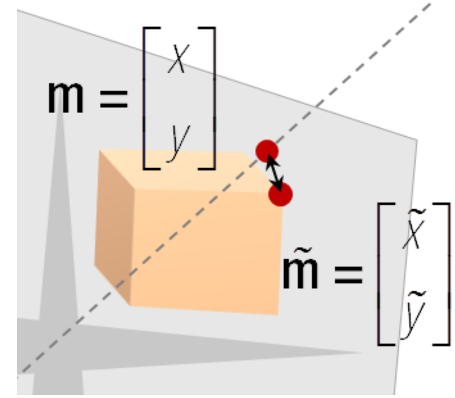
Reprojection error



$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\underset{\mathbf{R},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R},\mathbf{C},\mathbf{X}) / w(\mathbf{R},\mathbf{C},\mathbf{X}) \\ v(\mathbf{R},\mathbf{C},\mathbf{X}) / w(\mathbf{R},\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2$$

Reprojection error



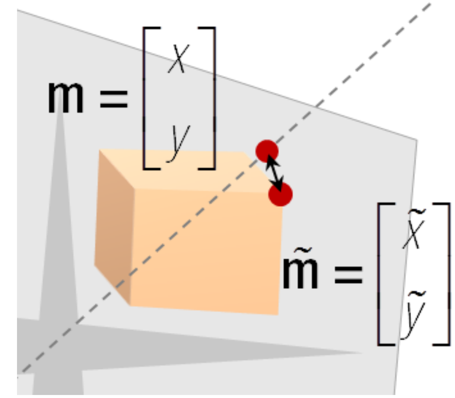
$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\underset{\mathbf{R},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R},\mathbf{C},\mathbf{X}) / w(\mathbf{R},\mathbf{C},\mathbf{X}) \\ v(\mathbf{R},\mathbf{C},\mathbf{X}) / w(\mathbf{R},\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2$$

$$= \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2$$

: Quaternion parameterization

Reprojection error



$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\left[\mathbf{I}_{3\times 3} \quad -\mathbf{C}\right]\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\underset{\mathbf{R},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R},\mathbf{C},\mathbf{X})/w(\mathbf{R},\mathbf{C},\mathbf{X}) \\ v(\mathbf{R},\mathbf{C},\mathbf{X})/w(\mathbf{R},\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2$$

$$= \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \underbrace{\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}}_{\mathbf{b}} - \underbrace{\begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})/w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})/w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix}}_{\mathbf{f}(\mathbf{R},\mathbf{C},\mathbf{X})} \right\|^2$$

: Quaternion parameterization

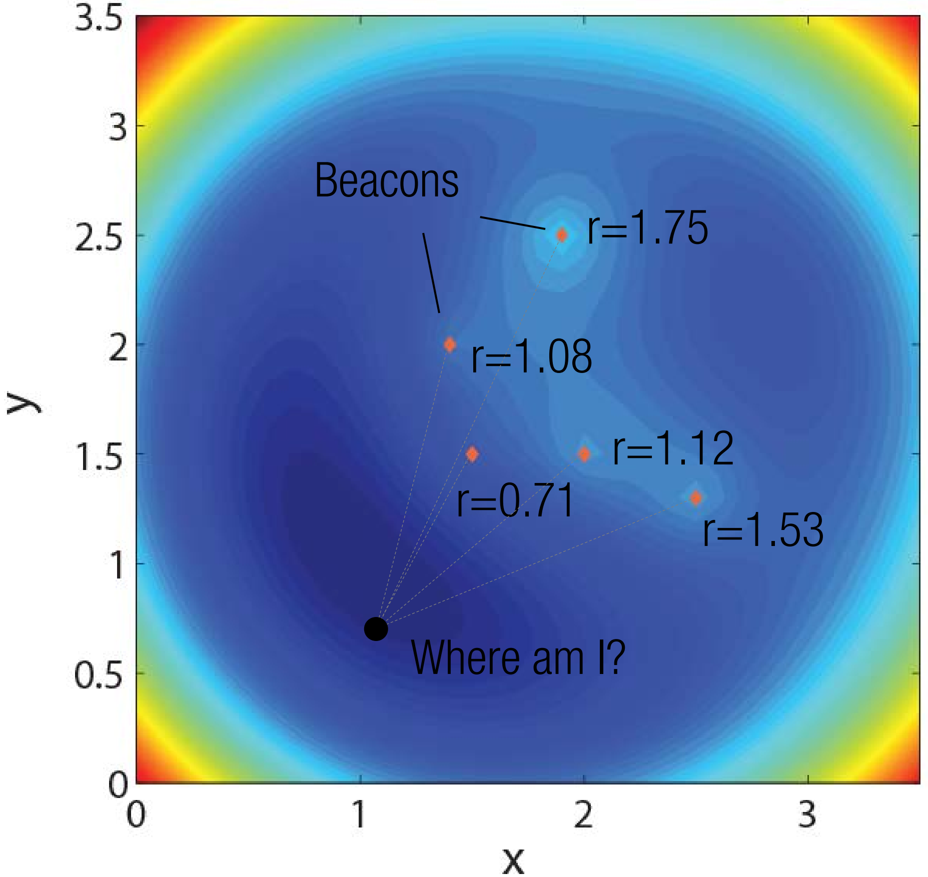
Recall

Nonlinear least squares:

$$\underset{\mathbf{x}}{\text{minimize}} \left\| \mathbf{f}(\mathbf{x}) - \mathbf{b} \right\|^2 = \underset{\mathbf{x}}{\text{minimize}}\ \mathbf{f}(\mathbf{x})^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\mathbf{b}^{\mathsf{T}} \mathbf{f}(\mathbf{x})$$

$$\longrightarrow \left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{f}(\mathbf{x}) - 2\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{b} = 0$$

Example:
Localization using range data from beacons



Recall

Nonlinear least squares:

$$\underset{\mathbf{x}}{\text{minimize}} \left\| \mathbf{f(x)} - \mathbf{b} \right\|^2 = \underset{\mathbf{x}}{\text{minimize}}\ \mathbf{f(x)}^{\mathsf{T}}\mathbf{f(x)} - 2\mathbf{b}^{\mathsf{T}}\mathbf{f(x)}$$

$$\longrightarrow \left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f(x)}}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{f(x)} - 2\frac{\partial \mathbf{f(x)}}{\partial \mathbf{x}}^{\mathsf{T}} \mathbf{b} = \mathbf{0}$$

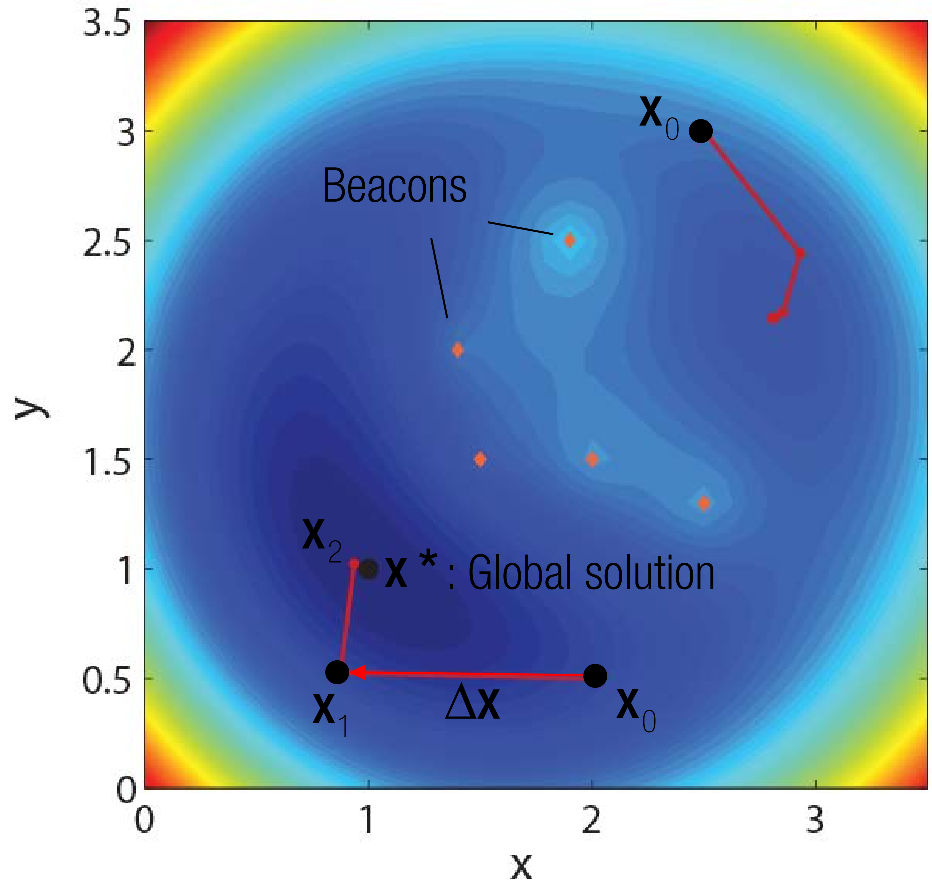
Taylor expansion around x:

$$\mathbf{f}(\mathbf{x}+\Delta\mathbf{x}) \approx \mathbf{f(x)} + \frac{\partial \mathbf{f(x)}}{\partial \mathbf{x}}\Delta\mathbf{x}$$

where $\boxed{\Delta\mathbf{x} = \left(\mathbf{J}^{\mathsf{T}}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathsf{T}}\left(\mathbf{b}-\mathbf{f(x)}\right)}$

Normal equation

Example:
Localization using range data from beacons



Recall

Nonlinear least squares:

$$\underset{\mathbf{x}}{\text{minimize}} \left\| \mathbf{f(x)} - \mathbf{b} \right\|^2 = \underset{\mathbf{x}}{\text{minimize}}\ \mathbf{f(x)}^T \mathbf{f(x)} - 2\mathbf{b}^T \mathbf{f(x)}$$

$$\longrightarrow \left.\frac{\partial E}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} = 2\frac{\partial \mathbf{f(x)}}{\partial \mathbf{x}}^T \mathbf{f(x)} - 2\frac{\partial \mathbf{f(x)}}{\partial \mathbf{x}}^T \mathbf{b} = \mathbf{0}$$

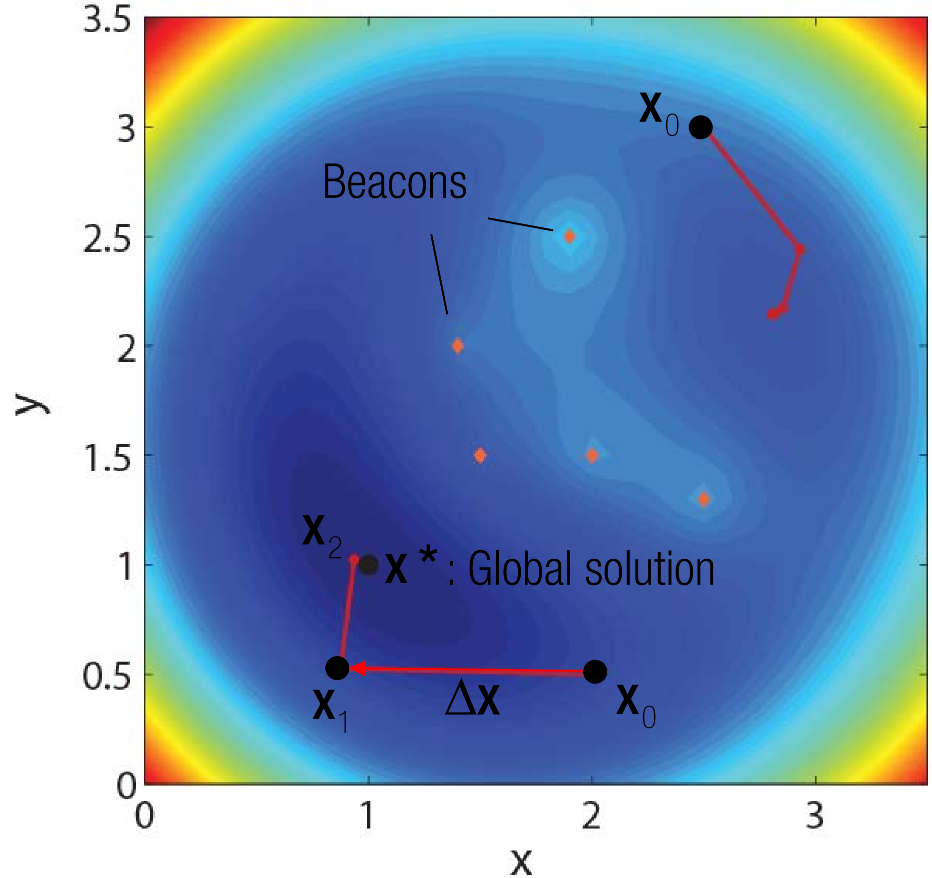
Taylor expansion around x:

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f(x)} + \frac{\partial \mathbf{f(x)}}{\partial \mathbf{x}} \Delta\mathbf{x}$$

where $\underbrace{\boxed{\Delta\mathbf{x} = \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\left(\mathbf{b} - \mathbf{f(x)}\right)}}_{\text{Normal equation}}$

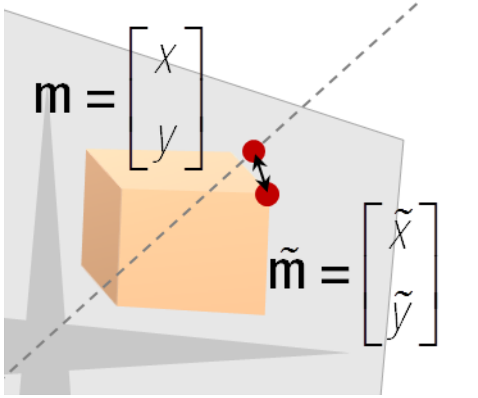
$$\mathbf{J} = \frac{\partial \mathbf{f(x)}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_n} \\ \vdots & & \vdots \\ \frac{\partial \mathbf{f}_m}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_m}{\partial \mathbf{x}_n} \end{bmatrix} \text{ : Jacobian matrix}$$

Example:
Localization using range data from beacons



Reprojection error

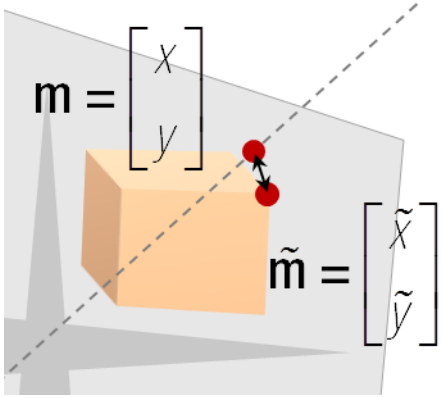
$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$



$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix}$$

Reprojection error

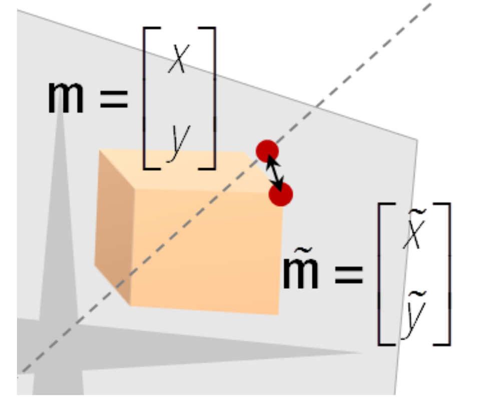


$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})/w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})/w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{X} - \mathbf{C} \end{bmatrix}$$

Reprojection error



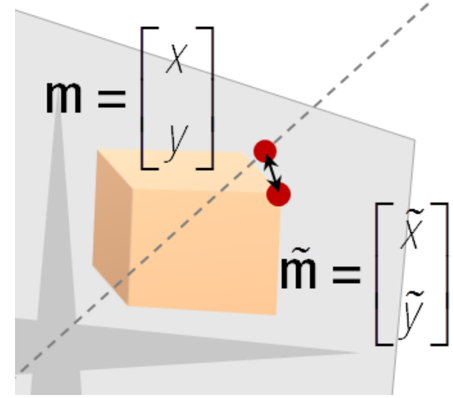
$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix} \qquad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{X} - \mathbf{C} \end{bmatrix}$$

$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} \mathbf{X} - \mathbf{C} \end{bmatrix}$$

Reprojection error



$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

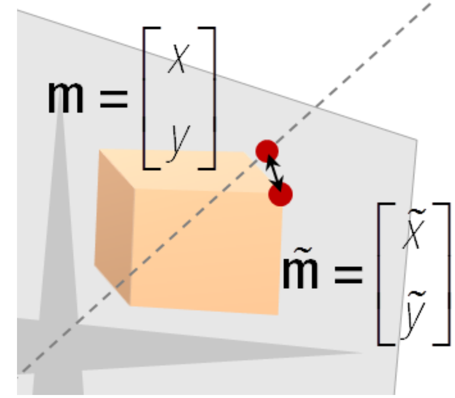
$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{X} - \mathbf{C} \end{bmatrix}$$

$$= \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} \mathbf{X} - \mathbf{C} \end{bmatrix}$$

$$= \begin{bmatrix} fr_{11} + p_x r_{31} & fr_{12} + p_x r_{32} & fr_{13} + p_x r_{33} \\ fr_{21} + p_y r_{31} & fr_{22} + p_y r_{32} & fr_{23} + p_y r_{33} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix} \mathbf{X} - \mathbf{C} \end{bmatrix}$$

Reprojection error



$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

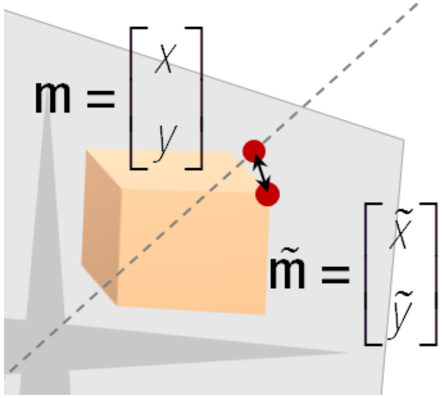
$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})/w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})/w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix}$$

$$\text{where} \quad \begin{aligned} u &= \begin{bmatrix} fr_{11}+p_x r_{31} & fr_{12}+p_x r_{32} & fr_{13}+p_x r_{33} \end{bmatrix}\begin{bmatrix}\mathbf{X}-\mathbf{C}\end{bmatrix} \\ v &= \begin{bmatrix} fr_{21}+p_y r_{31} & fr_{22}+p_y r_{32} & fr_{23}+p_y r_{33} \end{bmatrix}\begin{bmatrix}\mathbf{X}-\mathbf{C}\end{bmatrix} \\ w &= \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix}\begin{bmatrix}\mathbf{X}-\mathbf{C}\end{bmatrix} \end{aligned}$$

Reprojection error

$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

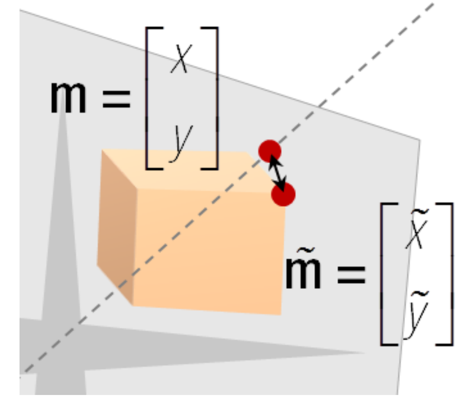


$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

J = ?

Reprojection error

$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u / w \\ v / w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\left[\mathbf{I}_{3\times3} \quad -\mathbf{C}\right] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

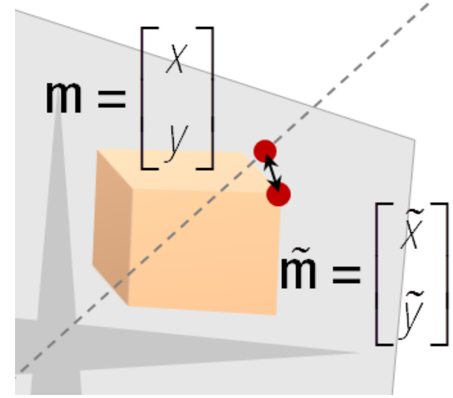


$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$\mathbf{J} =$

$\mathbf{q}$ $\qquad$ $\mathbf{C}$ $\qquad$ $\mathbf{X}$

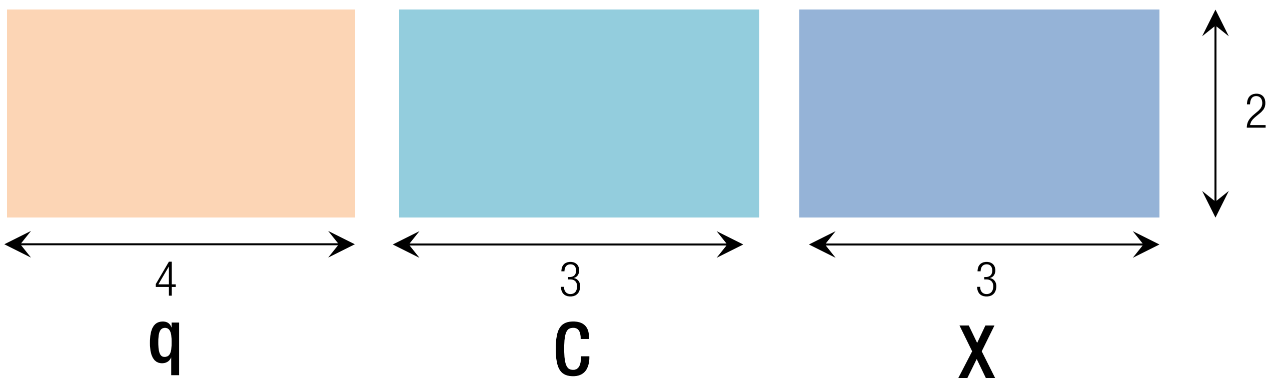
Reprojection error



$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

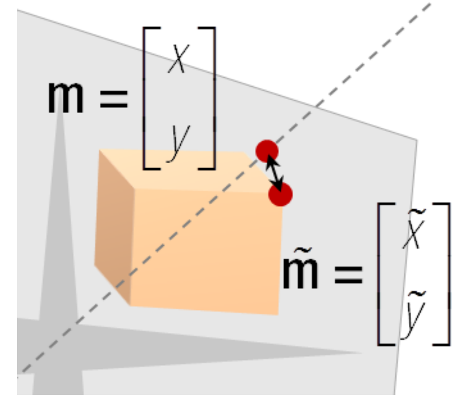
$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$\mathbf{J} =$



Reprojection error

$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

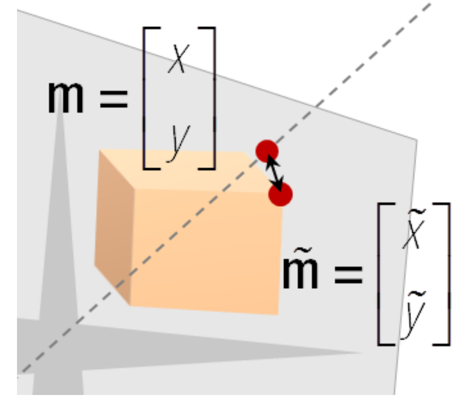


$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$$\mathbf{J} = \begin{array}{c} \begin{bmatrix} \underbrace{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{q}}}_{4} & \underbrace{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{3} & \underbrace{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{3} \end{bmatrix} \updownarrow 2 \\ \begin{matrix} \qquad \mathbf{q} \qquad & \qquad \mathbf{C} \qquad & \qquad \mathbf{X} \qquad \end{matrix} \end{array}$$

Reprojection error

$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$



$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{q}} & \dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}} & \dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}} \end{bmatrix}$$

$$= \begin{bmatrix} \underbrace{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} \underbrace{\dfrac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} & \underbrace{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} & \underbrace{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} \underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} \end{bmatrix}$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix}$$

where

$$u = \begin{bmatrix} fr_{11} + p_x r_{31} & fr_{12} + p_x r_{32} & fr_{13} + p_x r_{33} \end{bmatrix} \begin{bmatrix} \mathbf{X} - \mathbf{C} \end{bmatrix}$$

$$v = \begin{bmatrix} fr_{21} + p_y r_{31} & fr_{22} + p_y r_{32} & fr_{23} + p_y r_{33} \end{bmatrix} \begin{bmatrix} \mathbf{X} - \mathbf{C} \end{bmatrix}$$

$$w = \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} \mathbf{X} - \mathbf{C} \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} \underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} \end{bmatrix}$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix}$$

where

$$u = \begin{bmatrix} fr_{11} + p_x r_{31} & fr_{12} + p_x r_{32} & fr_{13} + p_x r_{33} \end{bmatrix} [\mathbf{X} - \mathbf{C}]$$

$$v = \begin{bmatrix} fr_{21} + p_y r_{31} & fr_{22} + p_y r_{32} & fr_{23} + p_y r_{33} \end{bmatrix} [\mathbf{X} - \mathbf{C}]$$

$$w = \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} [\mathbf{X} - \mathbf{C}]$$

$$\underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} = \begin{bmatrix} \dfrac{w\dfrac{\partial u}{\partial \mathbf{C}} - u\dfrac{\partial w}{\partial \mathbf{C}}}{w^2} \\[2ex] \dfrac{w\dfrac{\partial v}{\partial \mathbf{C}} - v\dfrac{\partial w}{\partial \mathbf{C}}}{w^2} \end{bmatrix}$$

where

$$\frac{\partial u}{\partial \mathbf{C}} = -\begin{bmatrix} fr_{11} + p_x r_{31} & fr_{12} + p_x r_{32} & fr_{13} + p_x r_{33} \end{bmatrix}$$

$$\frac{\partial v}{\partial \mathbf{C}} = -\begin{bmatrix} fr_{21} + p_y r_{31} & fr_{22} + p_y r_{32} & fr_{23} + p_y r_{33} \end{bmatrix}$$

$$\frac{\partial w}{\partial \mathbf{C}} = -\begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} \underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} \end{bmatrix}$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix}$$

where

$$u = \begin{bmatrix} fr_{11} + p_x r_{31} & fr_{12} + p_x r_{32} & fr_{13} + p_x r_{33} \end{bmatrix} [\mathbf{X} - \mathbf{C}]$$

$$v = \begin{bmatrix} fr_{21} + p_y r_{31} & fr_{22} + p_y r_{32} & fr_{23} + p_y r_{33} \end{bmatrix} [\mathbf{X} - \mathbf{C}]$$

$$w = \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} [\mathbf{X} - \mathbf{C}]$$

$$\underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} = \begin{bmatrix} \dfrac{w\dfrac{\partial u}{\partial \mathbf{X}} - u\dfrac{\partial w}{\partial \mathbf{X}}}{w^2} \\[2ex] \dfrac{w\dfrac{\partial v}{\partial \mathbf{X}} - v\dfrac{\partial w}{\partial \mathbf{X}}}{w^2} \end{bmatrix}$$

where

$$\frac{\partial u}{\partial \mathbf{X}} = \begin{bmatrix} fr_{11} + p_x r_{31} & fr_{12} + p_x r_{32} & fr_{13} + p_x r_{33} \end{bmatrix}$$

$$\frac{\partial v}{\partial \mathbf{X}} = \begin{bmatrix} fr_{21} + p_y r_{31} & fr_{22} + p_y r_{32} & fr_{23} + p_y r_{33} \end{bmatrix}$$

$$\frac{\partial w}{\partial \mathbf{X}} = \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} \underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} \end{bmatrix}$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix} \quad \text{where} \quad \begin{aligned} u &= \begin{bmatrix} fr_{11}+p_x r_{31} & fr_{12}+p_x r_{32} & fr_{13}+p_x r_{33} \end{bmatrix} [\mathbf{X}-\mathbf{C}] \\ v &= \begin{bmatrix} fr_{21}+p_y r_{31} & fr_{22}+p_y r_{32} & fr_{23}+p_y r_{33} \end{bmatrix} [\mathbf{X}-\mathbf{C}] \\ w &= \begin{bmatrix} r_{31} & r_{32} & r_{33} \end{bmatrix} [\mathbf{X}-\mathbf{C}] \end{aligned}$$

$$\underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} = \begin{bmatrix} \dfrac{w\frac{\partial u}{\partial \mathbf{R}} - u\frac{\partial w}{\partial \mathbf{R}}}{w^2} \\ \dfrac{w\frac{\partial v}{\partial \mathbf{R}} - v\frac{\partial w}{\partial \mathbf{R}}}{w^2} \end{bmatrix} \quad \text{where} \quad \begin{aligned} \frac{\partial u}{\partial \mathbf{R}} &= \begin{bmatrix} f(X_1 - C_1) & \mathbf{0}_{1\times 3} & p_x(X_3 - C_3) \end{bmatrix} \\ \frac{\partial v}{\partial \mathbf{R}} &= \begin{bmatrix} \mathbf{0}_{1\times 3} & f(X_1 - C_1) & p_x(X_3 - C_3) \end{bmatrix} \\ \frac{\partial w}{\partial \mathbf{R}} &= \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & (X_3 - C_3) \end{bmatrix} \end{aligned}$$

$$\mathbf{J} = \begin{bmatrix} \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} \underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} & \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 1-2q_z^2-2q_y^2 & -2q_zq_w+2q_yq_x & 2q_yq_w+2q_zq_x \\ 2q_xq_y+2q_wq_z & 1-2q_z^2-2q_x^2 & 2q_zq_y-2q_xq_w \\ 2q_xq_z-2q_wq_y & 2q_yq_z+2q_wq_x & 1-2q_y^2-2q_x^2 \end{bmatrix}$$

where

$$\mathbf{q} = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^T$$

$$\mathbf{J} = \left[\underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} \underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} \quad \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} \quad \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} \right]$$

$$\mathbf{R} = \begin{bmatrix} 1-2q_z^2-2q_y^2 & -2q_zq_w+2q_yq_x & 2q_yq_w+2q_zq_x \\ 2q_xq_y+2q_wq_z & 1-2q_z^2-2q_x^2 & 2q_zq_y-2q_xq_w \\ 2q_xq_z-2q_wq_y & 2q_yq_z+2q_wq_x & 1-2q_y^2-2q_x^2 \end{bmatrix}$$

where

$$\mathbf{q} = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^T$$

$$\underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} = \begin{bmatrix} \dfrac{\partial r_{11}}{\partial \mathbf{q}} \\ \dfrac{\partial r_{12}}{\partial \mathbf{q}} \\ \vdots \\ \dfrac{\partial r_{33}}{\partial \mathbf{q}} \end{bmatrix}$$

where

$$\frac{\partial \mathbf{R}_{11}}{\partial \mathbf{q}} = \begin{bmatrix} 0 & -4q_y & -4q_z & 0 \end{bmatrix}$$

$$\frac{\partial \mathbf{R}_{12}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_y & 2q_x & -2q_w & -2q_z \end{bmatrix}$$

$$\frac{\partial \mathbf{R}_{13}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_z & 2q_w & 2q_x & 2q_y \end{bmatrix}$$

$$\frac{\partial \mathbf{R}_{21}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_y & 2q_x & 2q_w & 2q_z \end{bmatrix}$$

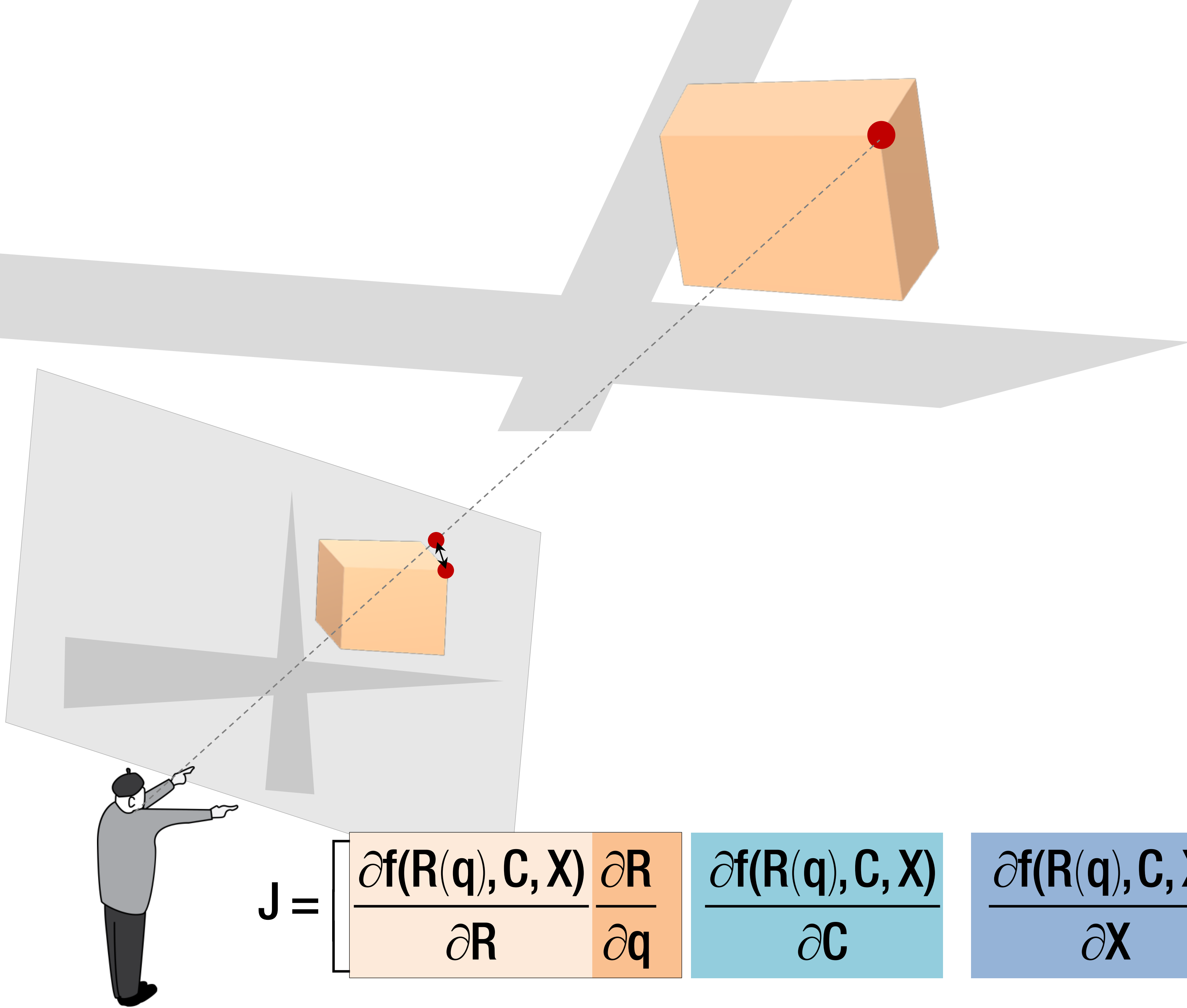
$$\frac{\partial \mathbf{R}_{22}}{\partial \mathbf{q}} = \begin{bmatrix} -4q_x & 0 & -4q_z & 0 \end{bmatrix}$$

$$\frac{\partial \mathbf{R}_{23}}{\partial \mathbf{q}} = \begin{bmatrix} -2q_w & 2q_z & 2q_y & 2q_x \end{bmatrix}$$

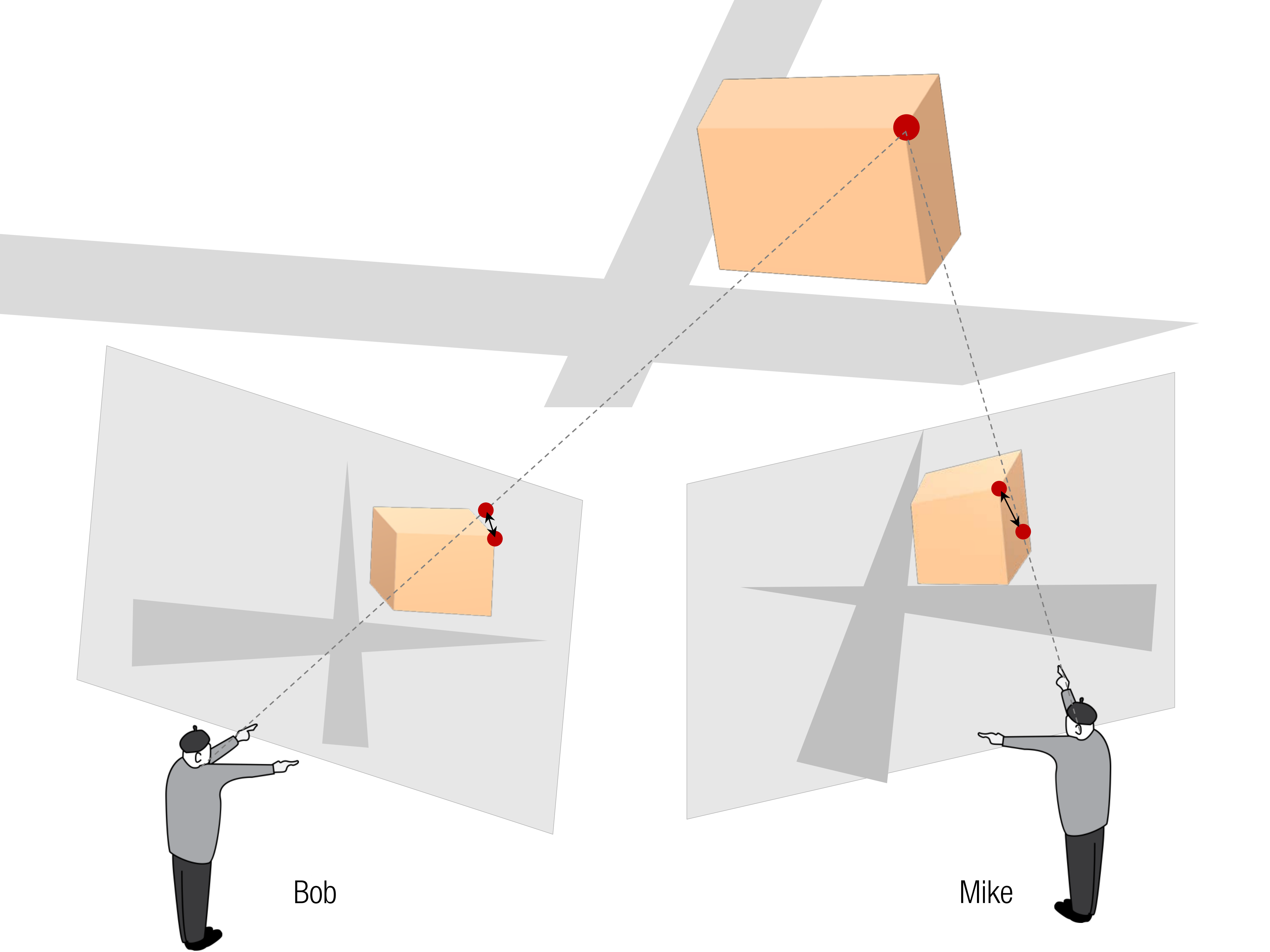
$$\frac{\partial \mathbf{R}_{31}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_z & -2q_w & 2q_x & -2q_y \end{bmatrix}$$

$$\frac{\partial \mathbf{R}_{33}}{\partial \mathbf{q}} = \begin{bmatrix} -4q_x & -4q_y & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \mathbf{R}_{32}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_w & 2q_z & 2q_y & 2q_x \end{bmatrix}$$

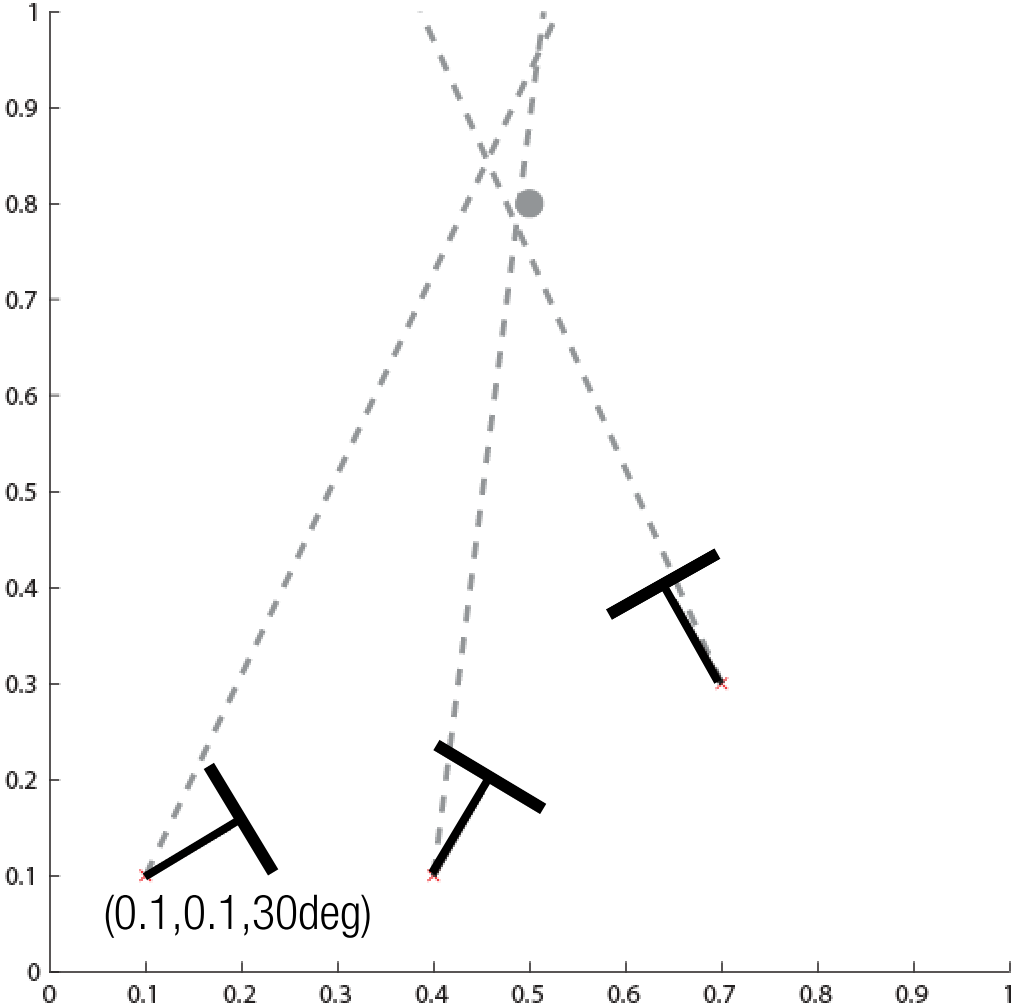


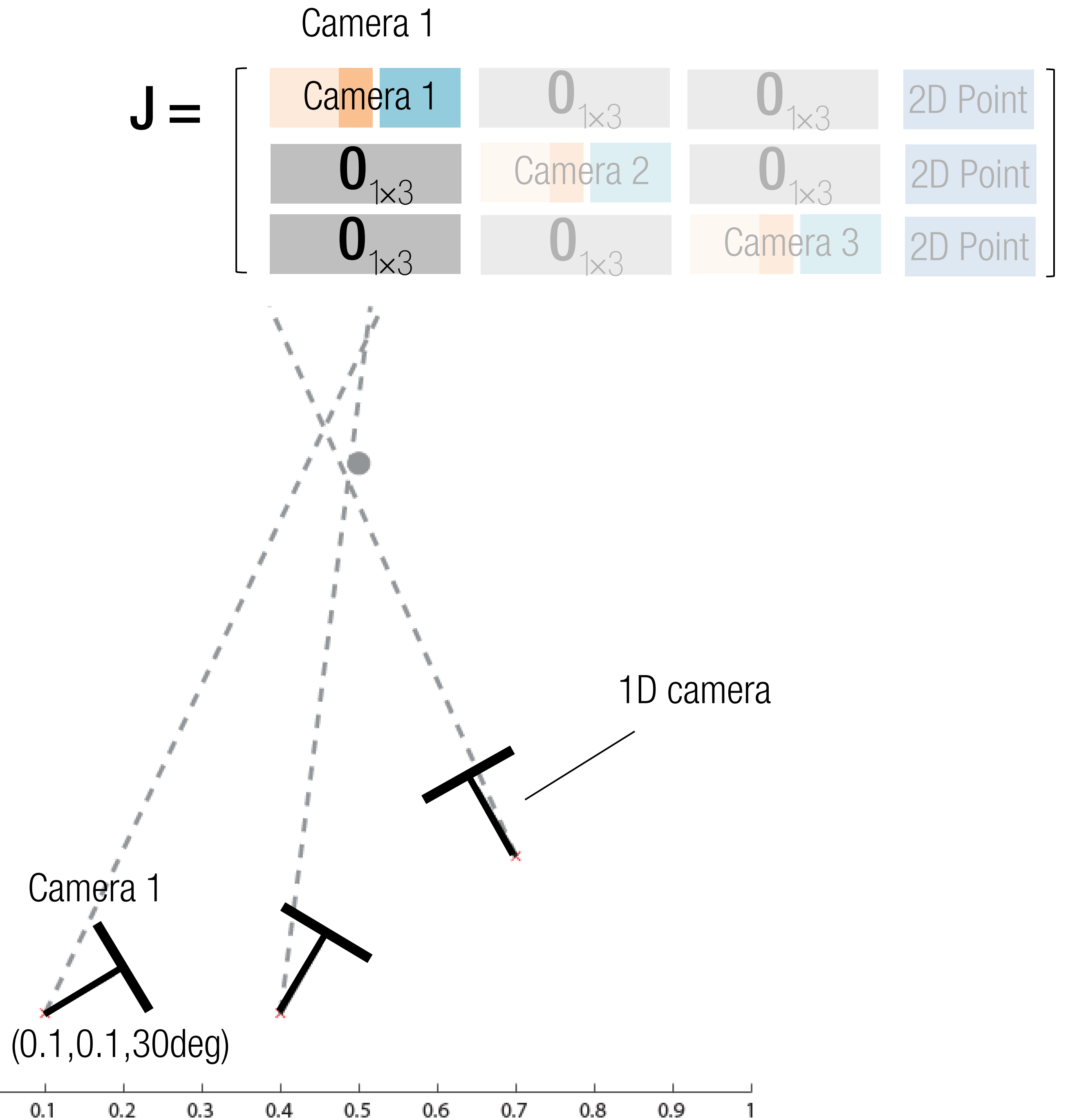
$$J = \begin{bmatrix} \dfrac{\partial f(R(q), C, X)}{\partial R} \dfrac{\partial R}{\partial q} & \dfrac{\partial f(R(q), C, X)}{\partial C} & \dfrac{\partial f(R(q), C, X)}{\partial X} \end{bmatrix}$$

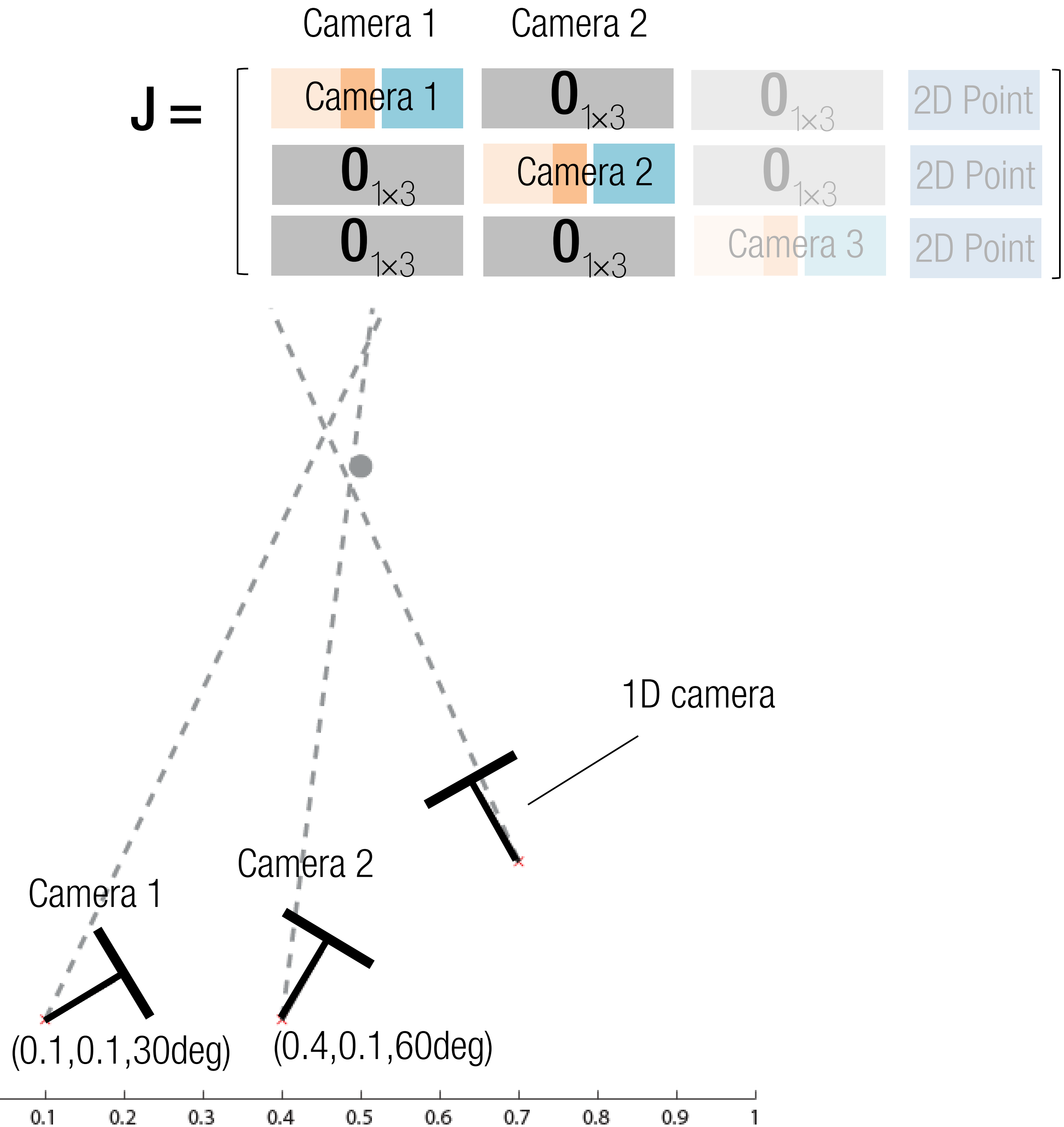

Bob
Mike

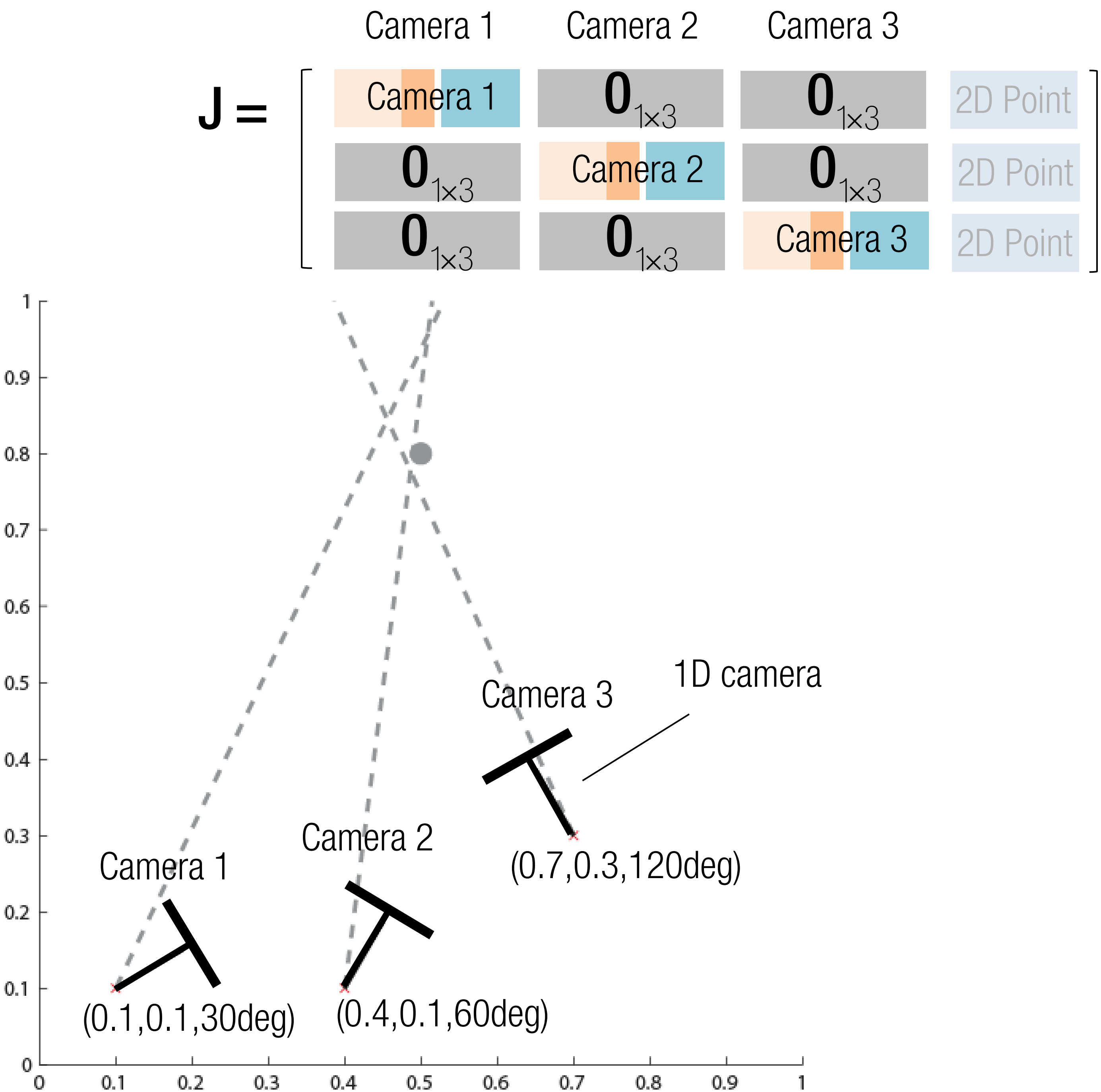
$$\mathbf{J} = \left[\underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}), \mathbf{C}, \mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} \underbrace{\frac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} \quad \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}), \mathbf{C}, \mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} \quad \underbrace{\frac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}), \mathbf{C}, \mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} \right]$$

$$\mathbf{J} = \begin{bmatrix} \text{Bob's Jacobian} & \mathbf{0}_{2\times 7} & \text{3D Point} \\ \mathbf{0}_{2\times 7} & \text{Mike's Jacobian} & \text{3D Point} \end{bmatrix}$$

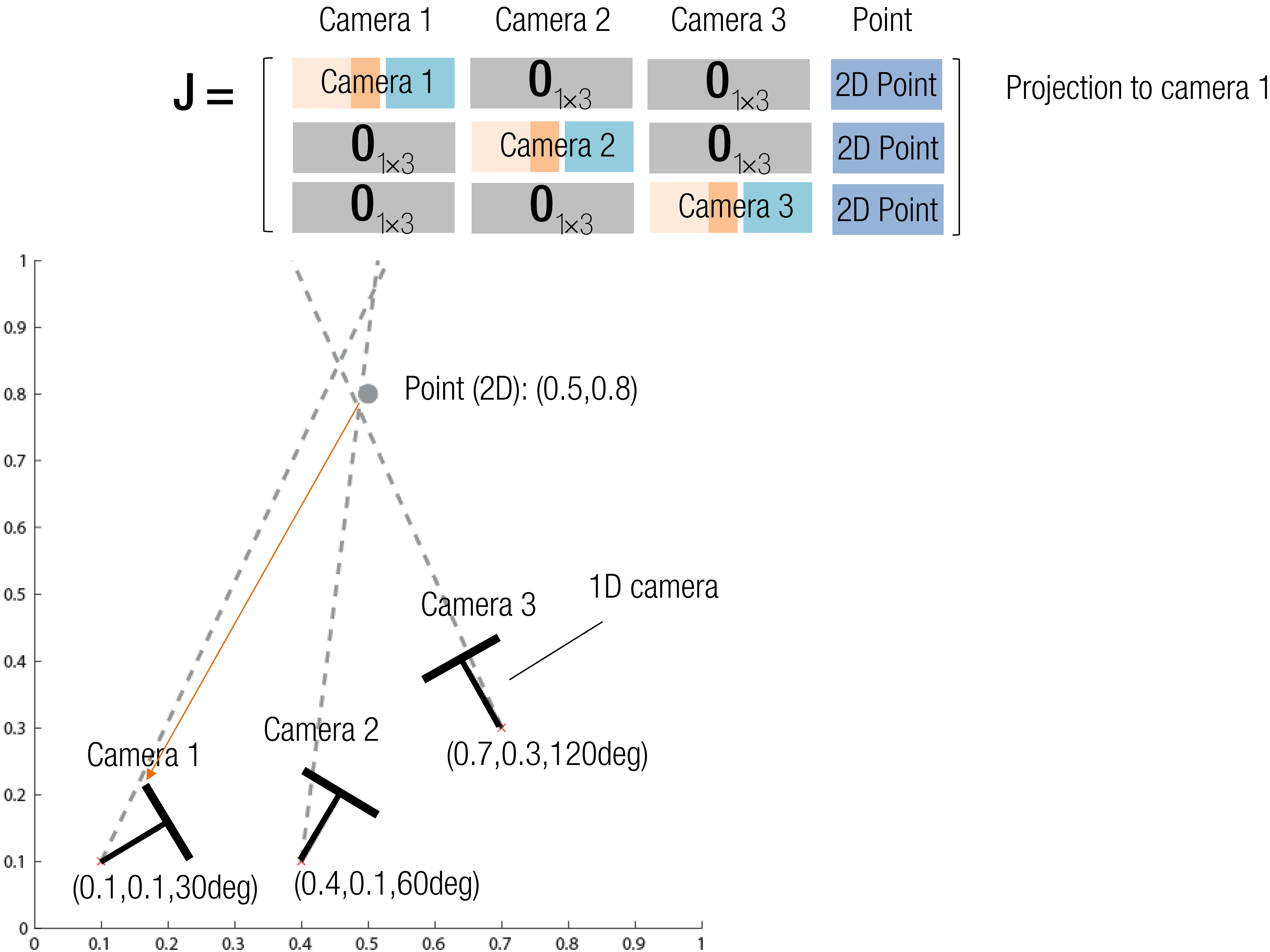

(0.1,0.1,30deg)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

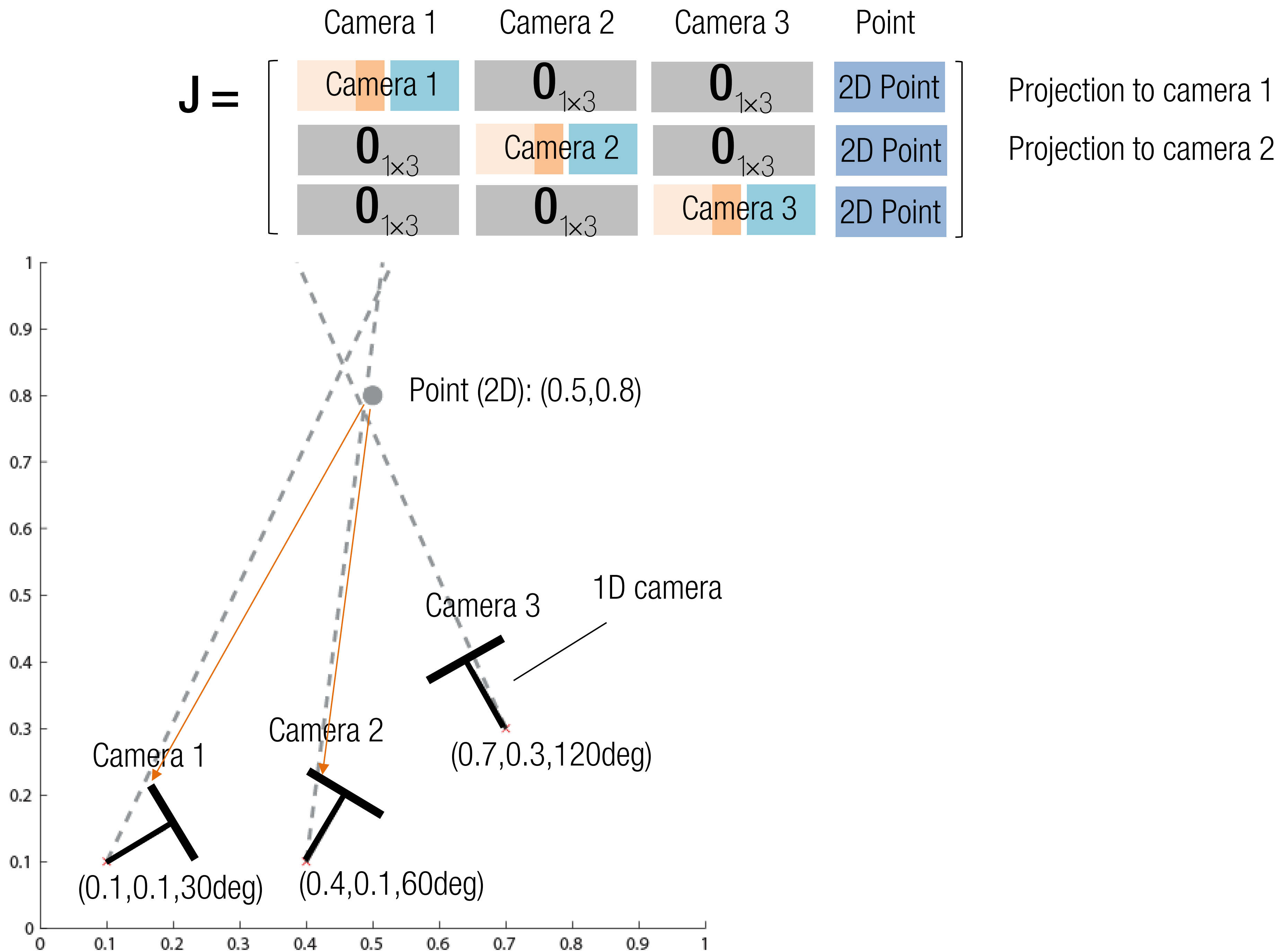

Camera 1
J =
Camera 1
0 1×3
0 1×3
2D Point
0 1×3
Camera 2
0 1×3
2D Point
0 1×3
0 1×3
Camera 3
2D Point
1D camera
Camera 1
(0.1,0.1,30deg)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

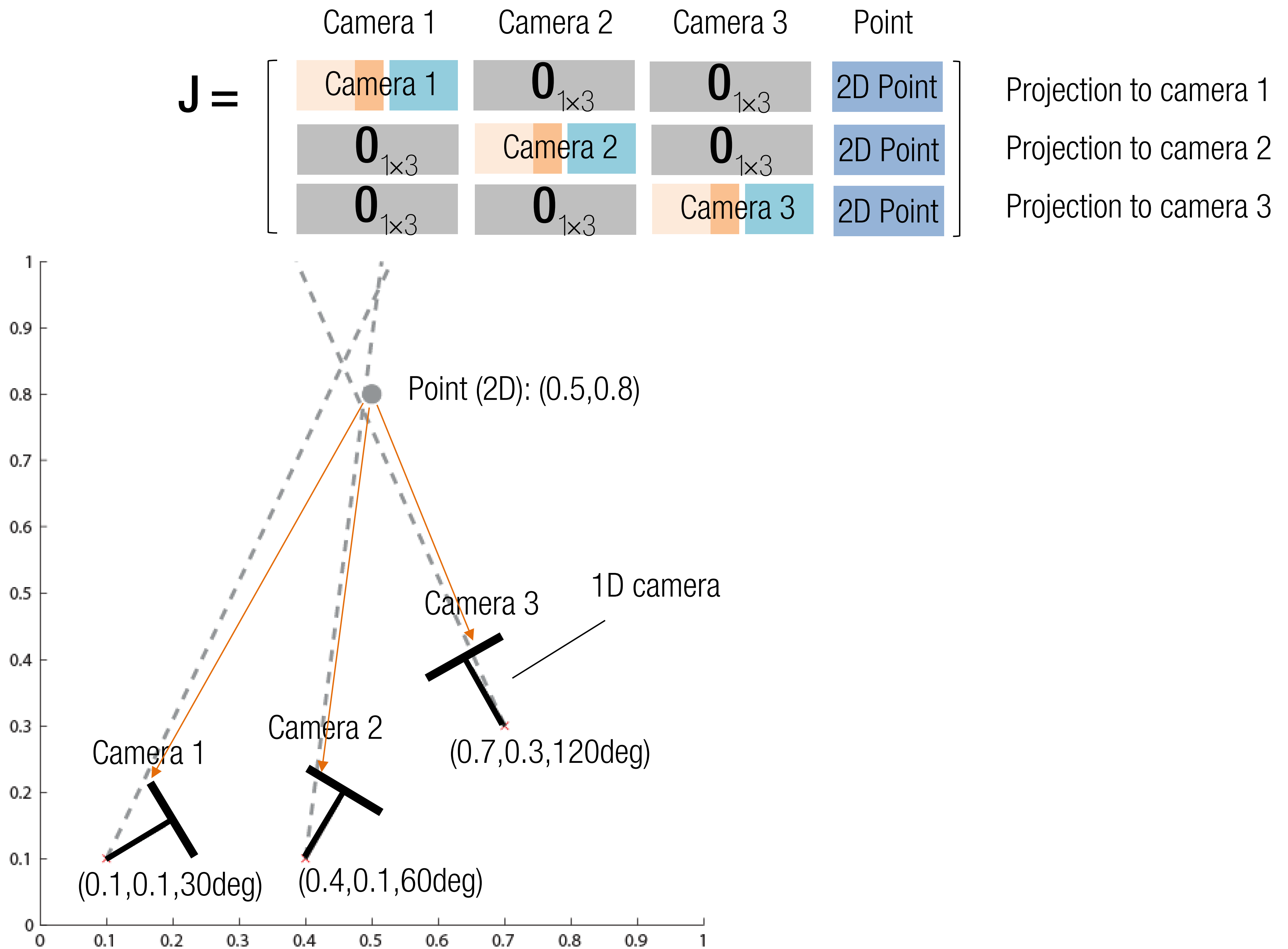

Camera 1
Camera 2
J =
Camera 1
$\mathbf{0}_{1\times3}$
$\mathbf{0}_{1\times3}$
2D Point
$\mathbf{0}_{1\times3}$
Camera 2
$\mathbf{0}_{1\times3}$
2D Point
$\mathbf{0}_{1\times3}$
$\mathbf{0}_{1\times3}$
Camera 3
2D Point
1D camera
Camera 1
Camera 2
(0.1,0.1,30deg)
(0.4,0.1,60deg)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

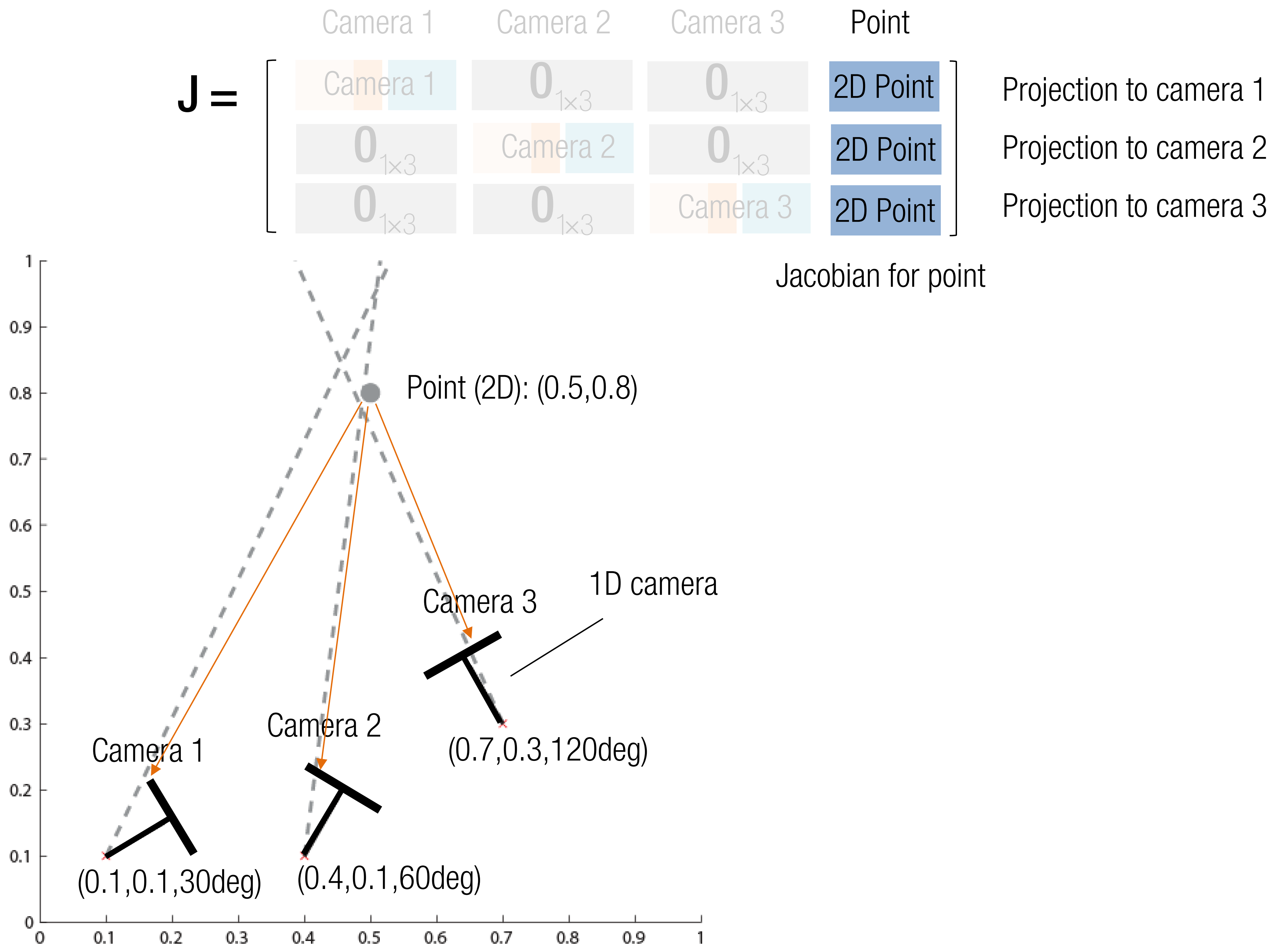


Camera 1
Camera 2
Camera 3
J =
Camera 1
$\mathbf{0}_{1\times3}$
$\mathbf{0}_{1\times3}$
2D Point
$\mathbf{0}_{1\times3}$
Camera 2
$\mathbf{0}_{1\times3}$
2D Point
$\mathbf{0}_{1\times3}$
$\mathbf{0}_{1\times3}$
Camera 3
2D Point
1D camera
Camera 3
(0.7,0.3,120deg)
Camera 2
Camera 1
(0.1,0.1,30deg)
(0.4,0.1,60deg)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Camera 1
Camera 2
Camera 3
Point
J =
Camera 1
0 1×3
0 1×3
2D Point
Projection to camera 1
0 1×3
Camera 2
0 1×3
2D Point
0 1×3
0 1×3
Camera 3
2D Point
Point (2D): (0.5,0.8)
1D camera
Camera 3
(0.7,0.3,120deg)
Camera 2
Camera 1
(0.1,0.1,30deg)
(0.4,0.1,60deg)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Camera 1
Camera 2
Camera 3
Point
J =
Camera 1
0 1×3
0 1×3
2D Point
Projection to camera 1
0 1×3
Camera 2
0 1×3
2D Point
Projection to camera 2
0 1×3
0 1×3
Camera 3
2D Point
Point (2D): (0.5,0.8)
1D camera
Camera 3
Camera 2
Camera 1
(0.7,0.3,120deg)
(0.1,0.1,30deg)
(0.4,0.1,60deg)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

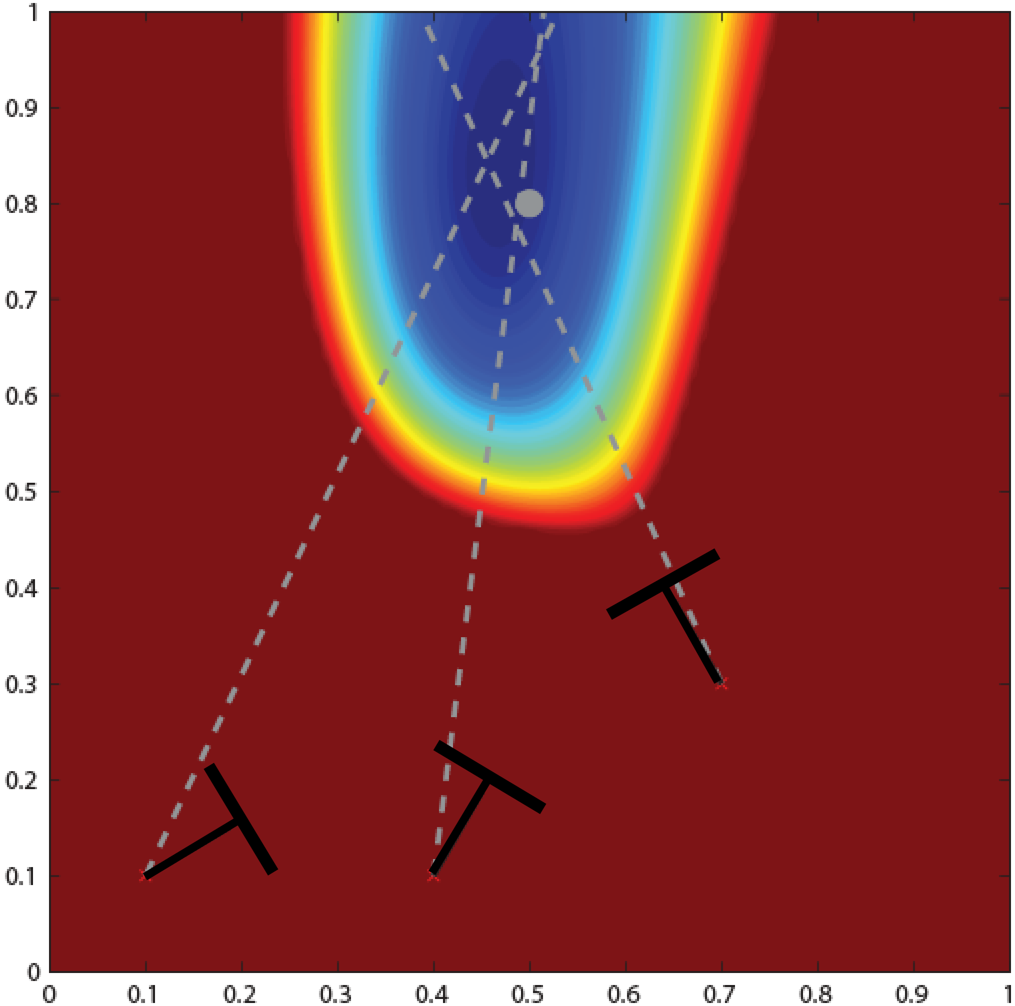

Camera 1
Camera 2
Camera 3
Point
J =
Camera 1
0 1×3
0 1×3
2D Point
Projection to camera 1
0 1×3
Camera 2
0 1×3
2D Point
Projection to camera 2
0 1×3
0 1×3
Camera 3
2D Point
Projection to camera 3
Point (2D): (0.5,0.8)
1D camera
Camera 3
(0.7,0.3,120deg)
Camera 2
Camera 1
(0.1,0.1,30deg)
(0.4,0.1,60deg)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Camera 1
Camera 2
Camera 3
Point
J =
Camera 1
$0_{1\times3}$
$0_{1\times3}$
2D Point
$0_{1\times3}$
Camera 2
$0_{1\times3}$
2D Point
$0_{1\times3}$
$0_{1\times3}$
Camera 3
2D Point
Projection to camera 1
Projection to camera 2
Projection to camera 3
Jacobian for point
Point (2D): (0.5,0.8)
1D camera
Camera 3
(0.7,0.3,120deg)
Camera 2
Camera 1
(0.1,0.1,30deg)
(0.4,0.1,60deg)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

$$
\mathbf{J} = \begin{array}{cccc|l}
\text{Camera 1} & \text{Camera 2} & \text{Camera 3} & \text{Point} & \\
\end{array}
$$

$$
\mathbf{J} = \left[\begin{array}{cccc}
\text{Camera 1} & 0_{1\times 3} & 0_{1\times 3} & \text{2D Point} \\
0_{1\times 3} & \text{Camera 2} & 0_{1\times 3} & \text{2D Point} \\
0_{1\times 3} & 0_{1\times 3} & \text{Camera 3} & \text{2D Point}
\end{array}\right]
\begin{array}{l}
\text{Projection to camera 1} \\
\text{Projection to camera 2} \\
\text{Projection to camera 3}
\end{array}
$$

Jacobian for point

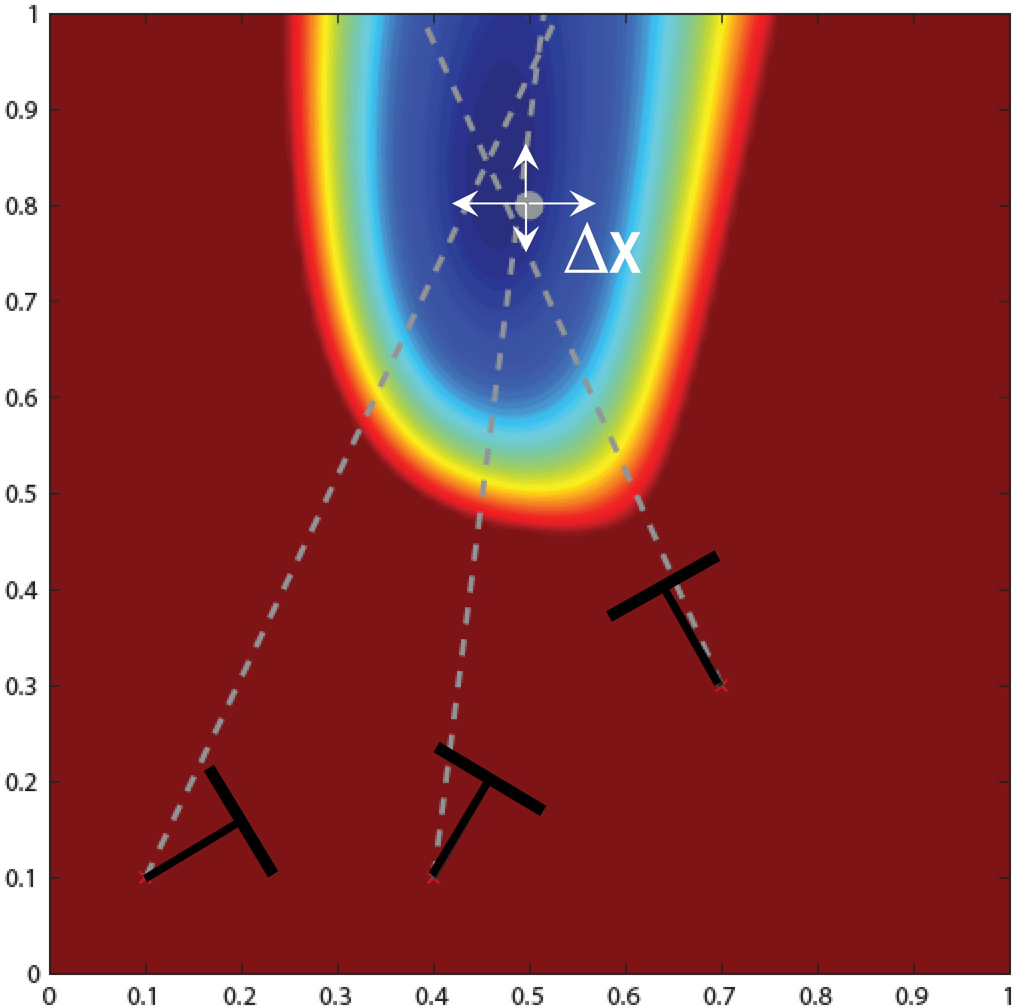


Cost:

$$
\sum_{i=1}^{3} \left\| \tilde{x} - f_i(\mathbf{X}) \right\|^2 = \sum_{i=1}^{3} \left\| \tilde{x} - u_i / w_i \right\|^2
$$

$$
\begin{array}{cccccc}
 & \text{Camera 1} & \text{Camera 2} & \text{Camera 3} & \text{Point} & \\
\mathbf{J} = & \left[\begin{array}{c} \text{Camera 1} \\ 0_{1\times3} \\ 0_{1\times3} \end{array}\right. & \begin{array}{c} 0_{1\times3} \\ \text{Camera 2} \\ 0_{1\times3} \end{array} & \begin{array}{c} 0_{1\times3} \\ 0_{1\times3} \\ \text{Camera 3} \end{array} & \left.\begin{array}{c} \text{2D Point} \\ \text{2D Point} \\ \text{2D Point} \end{array}\right] & \begin{array}{l} \text{Projection to camera 1} \\ \text{Projection to camera 2} \\ \text{Projection to camera 3} \end{array}
\end{array}
$$

Jacobian for point



Cost:

$$\sum_{i=1}^{3} \left\| \tilde{x} - f_i(\mathbf{X}) \right\|^2 = \sum_{i=1}^{3} \left\| \tilde{x} - u_i / w_i \right\|^2$$

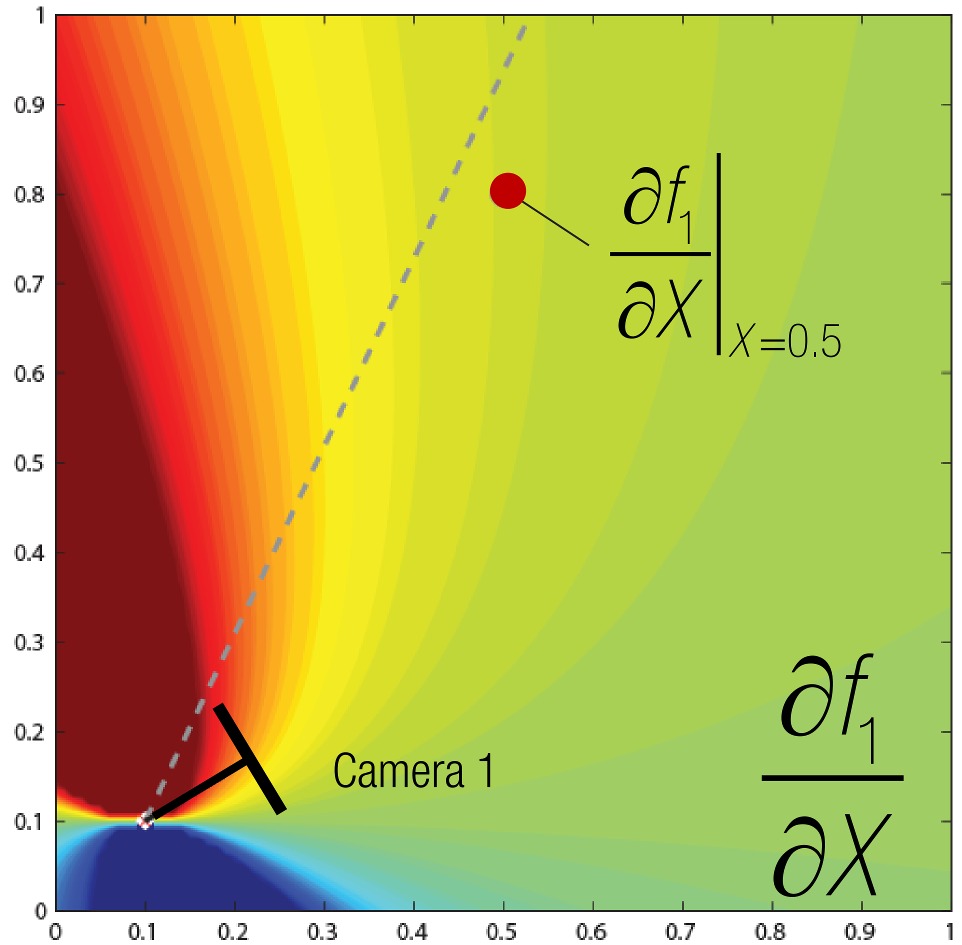
$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \Delta\mathbf{x}$$

$$\text{where } \mathbf{J}^{\mathsf{T}}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^{\mathsf{T}}$$

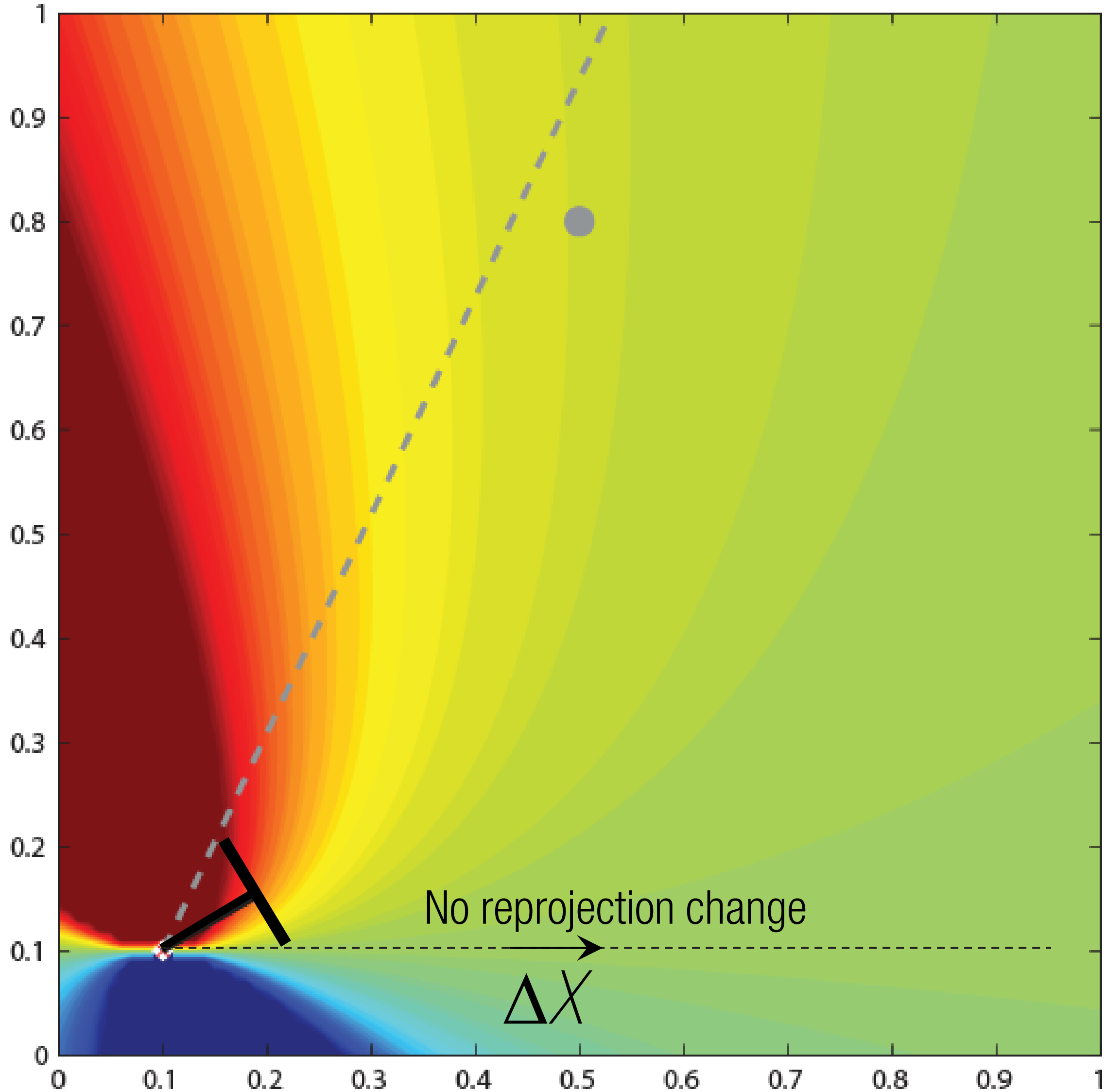
$$\mathbf{J}^{\mathrm{T}}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\mathrm{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right) \quad \text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{J} = \begin{bmatrix} \boxed{\dfrac{\partial f_1}{\partial X}} & \dfrac{\partial f_1}{\partial Y} \\ \dfrac{\partial f_2}{\partial X} & \dfrac{\partial f_2}{\partial Y} \\ \dfrac{\partial f_3}{\partial X} & \dfrac{\partial f_3}{\partial Y} \end{bmatrix}$$



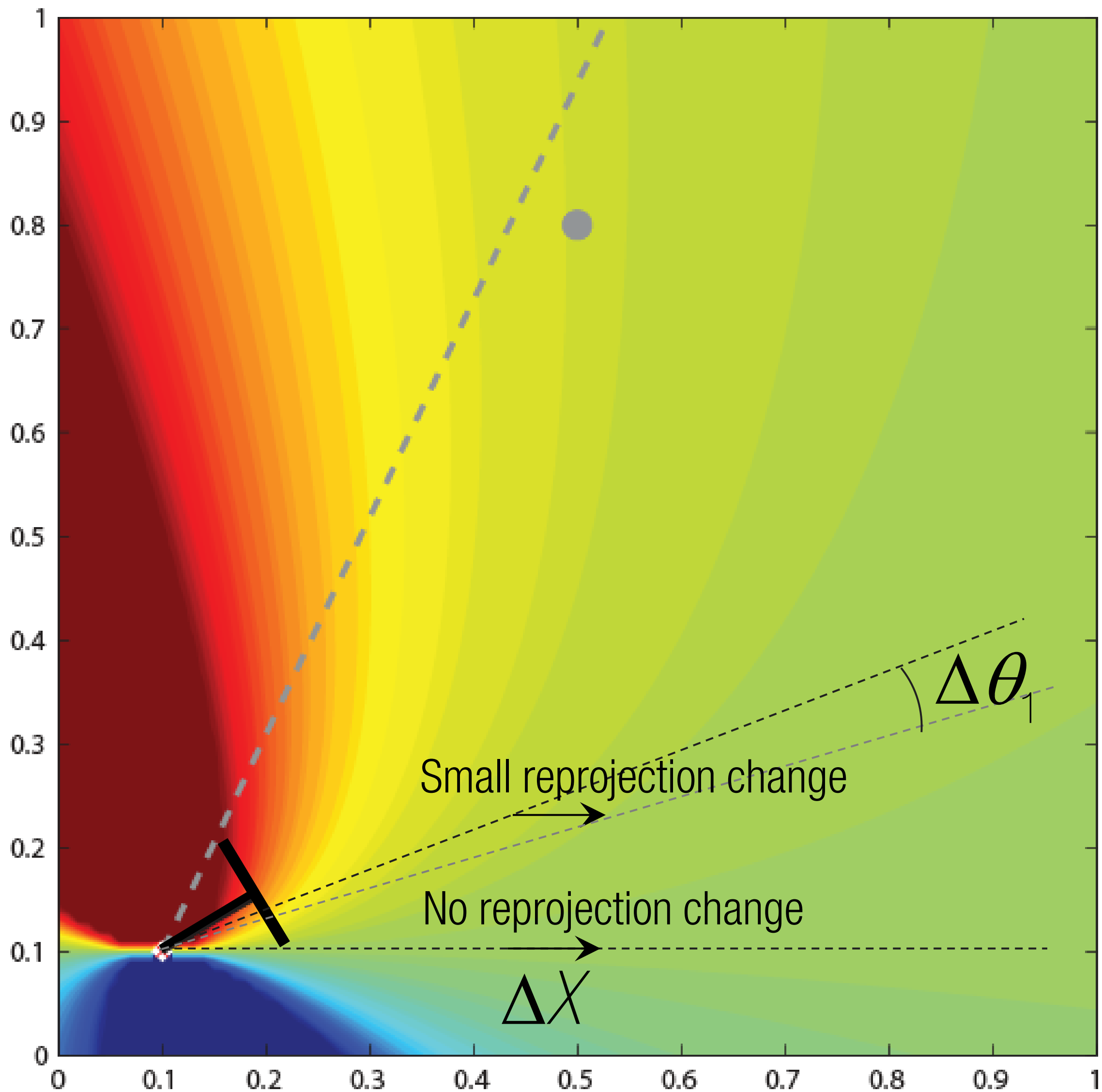
$$\mathbf{J}^{\mathsf{T}}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right) \quad \text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^{\mathsf{T}}$$

$$\frac{\partial f_1}{\partial X} =$$



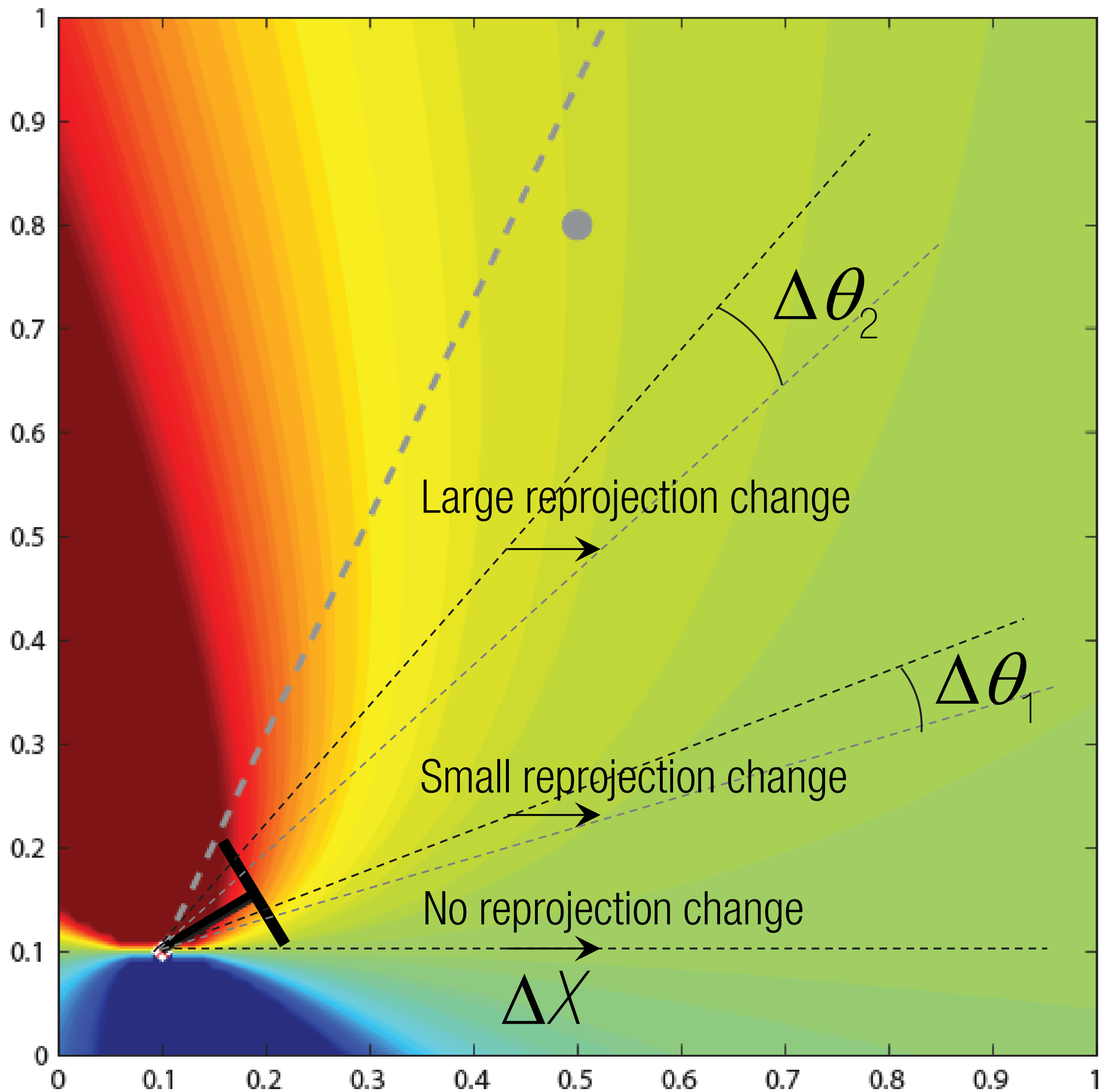
$$\mathbf{J}^{\mathsf{T}}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right) \quad \text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^{\mathsf{T}}$$

$$\frac{\partial f_1}{\partial X} =$$



$$\mathbf{J}^{\mathsf{T}}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right) \quad \text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^{\mathsf{T}}$$

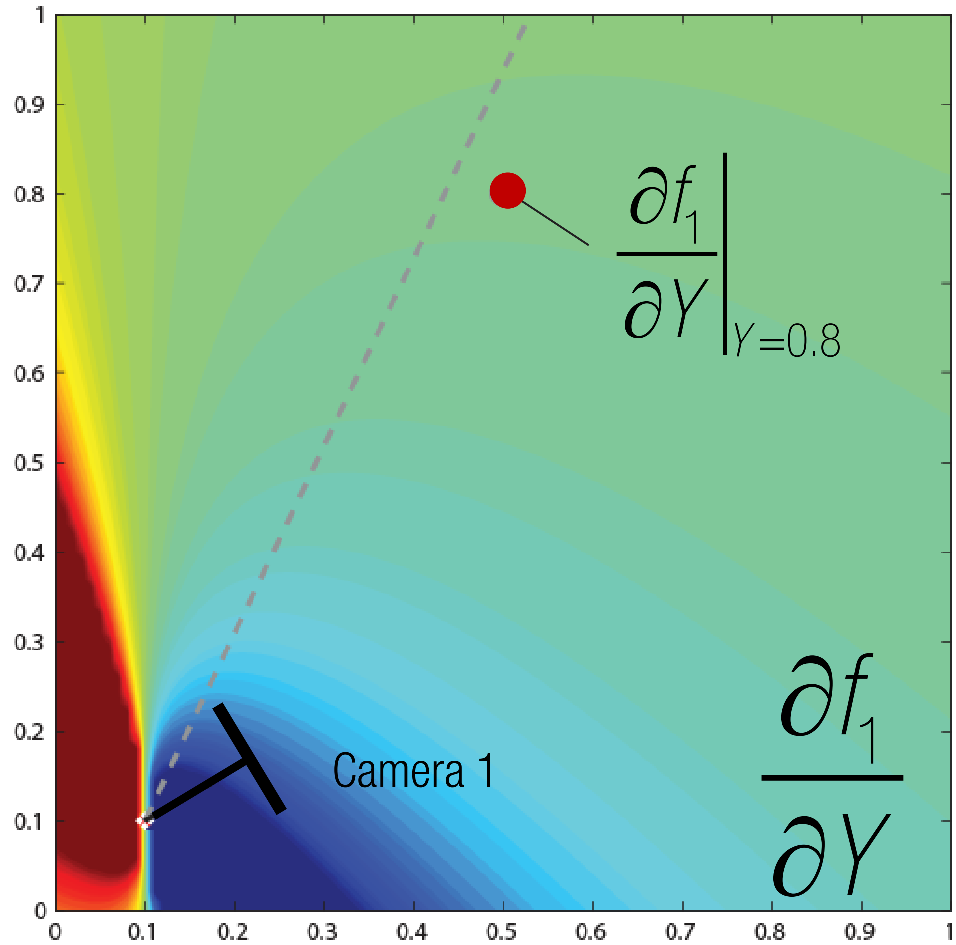
$$\frac{\partial f_1}{\partial X} =$$



$$\Delta\theta_2 > \Delta\theta_1$$

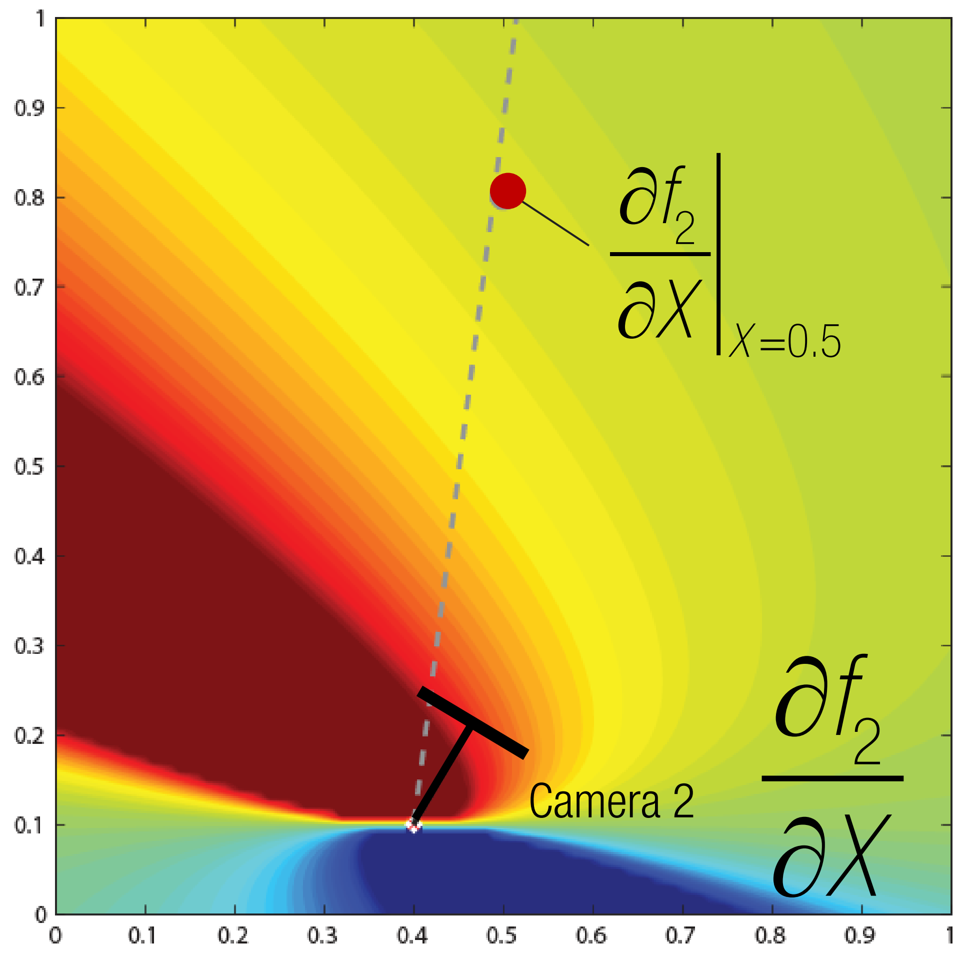
$$\mathbf{J}^{\mathrm{T}}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\mathrm{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right) \quad \text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial f_1}{\partial X} & \boxed{\dfrac{\partial f_1}{\partial Y}} \\ \dfrac{\partial f_2}{\partial X} & \dfrac{\partial f_2}{\partial Y} \\ \dfrac{\partial f_3}{\partial X} & \dfrac{\partial f_3}{\partial Y} \end{bmatrix}$$



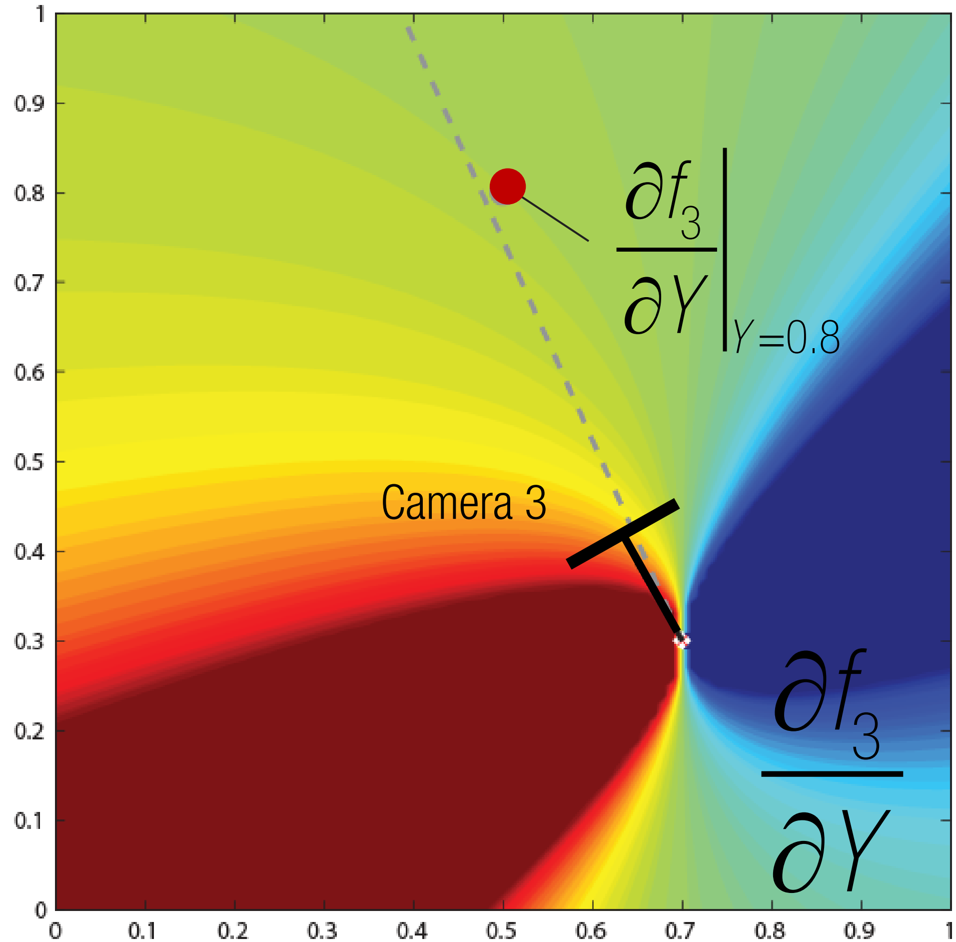
$$\mathbf{J}^{\mathsf{T}}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right) \quad \text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^{\mathsf{T}}$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial X} & \frac{\partial f_1}{\partial Y} \\ \boxed{\frac{\partial f_2}{\partial X}} & \frac{\partial f_2}{\partial Y} \\ \frac{\partial f_3}{\partial X} & \frac{\partial f_3}{\partial Y} \end{bmatrix}$$



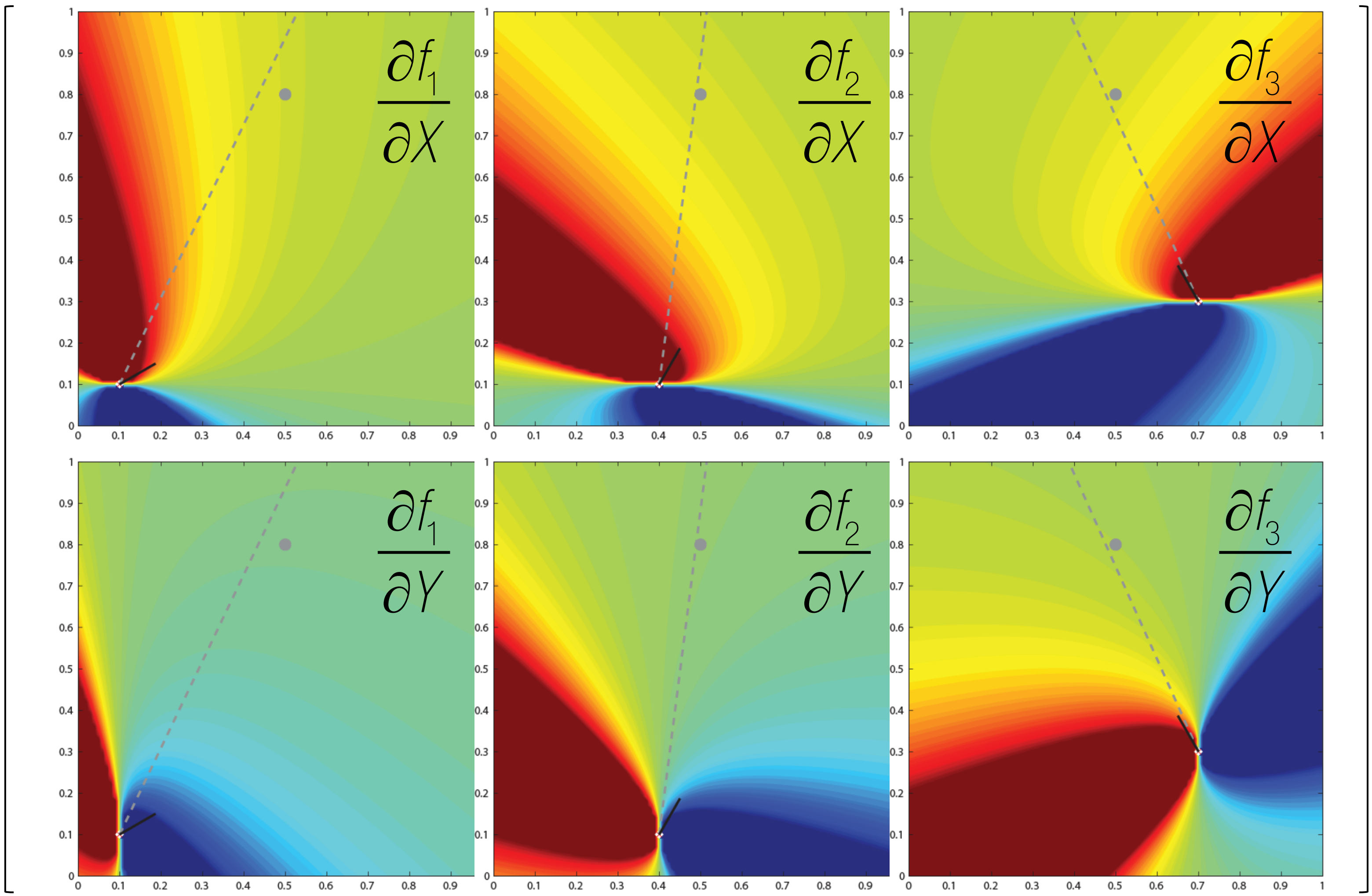
$$\mathbf{J}^{\mathrm{T}}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\mathrm{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right) \quad \text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial f_1}{\partial X} & \dfrac{\partial f_1}{\partial Y} \\ \dfrac{\partial f_2}{\partial X} & \dfrac{\partial f_2}{\partial Y} \\ \dfrac{\partial f_3}{\partial X} & \boxed{\dfrac{\partial f_3}{\partial Y}} \end{bmatrix}$$



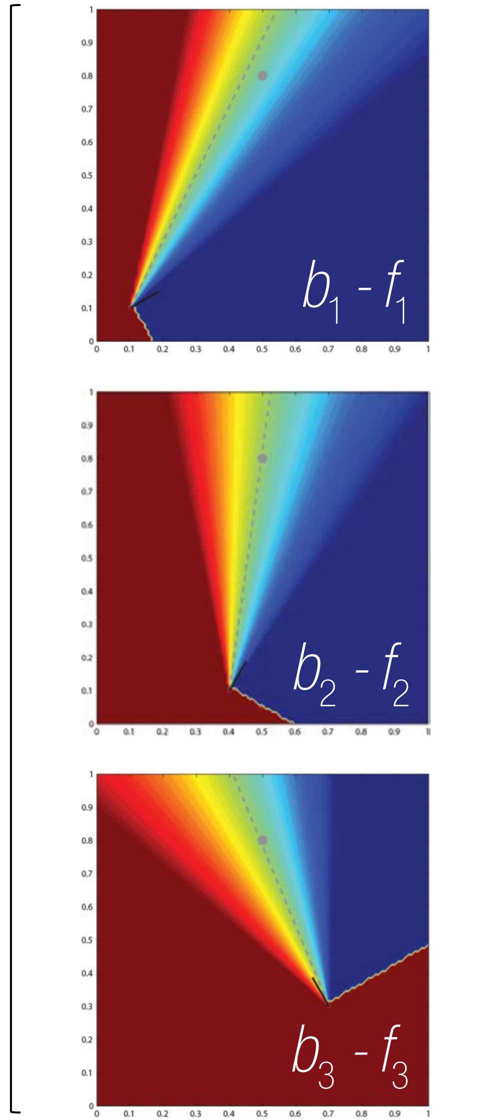
$$\mathbf{J}^{\mathsf{T}}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

$$\mathbf{J}^{\mathsf{T}} =$$



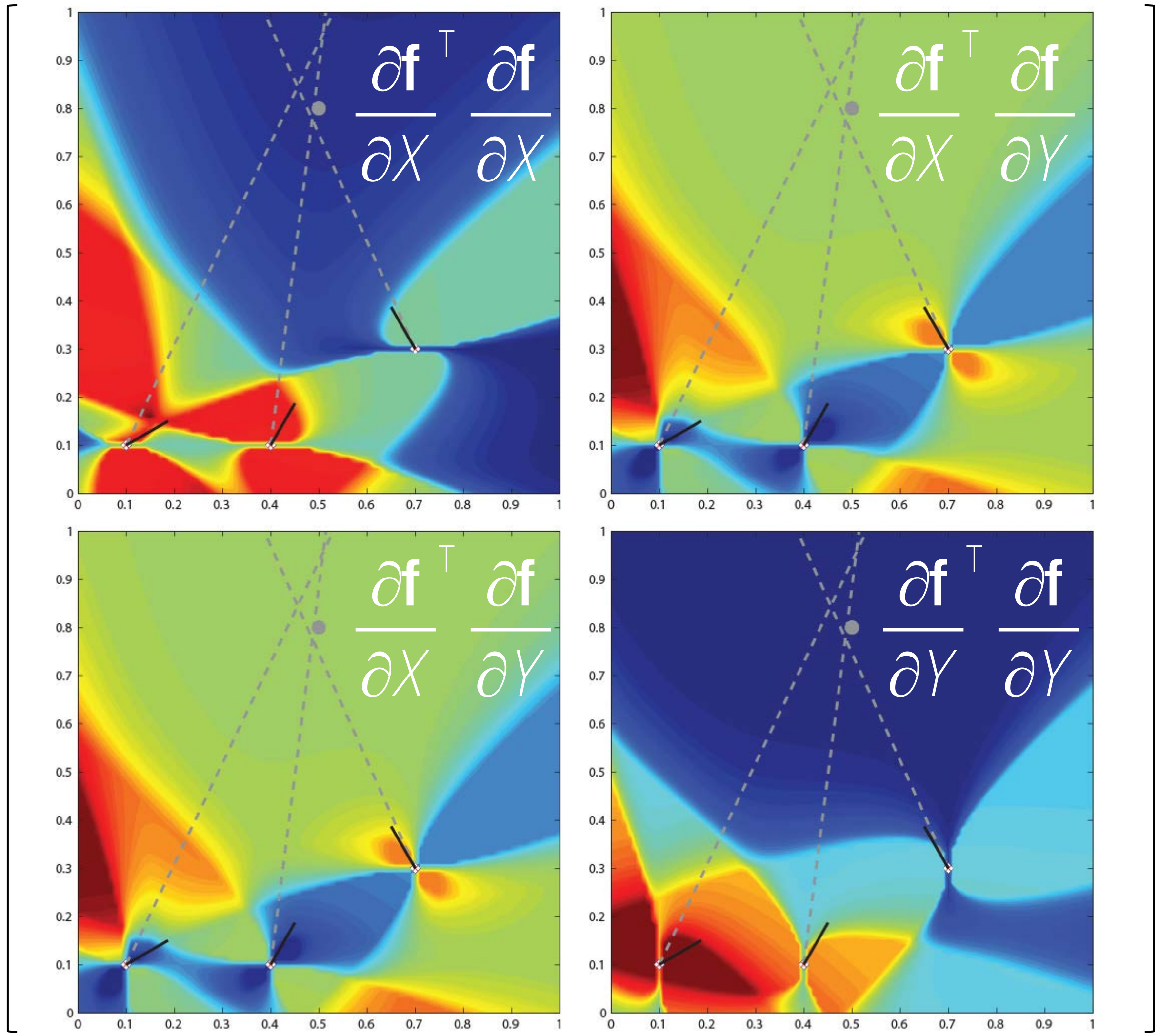
$$\mathbf{J}^\top \mathbf{J}\Delta \mathbf{x} = \mathbf{J}^\top \left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

$$\mathbf{b} - \mathbf{f}(\mathbf{x}) =$$

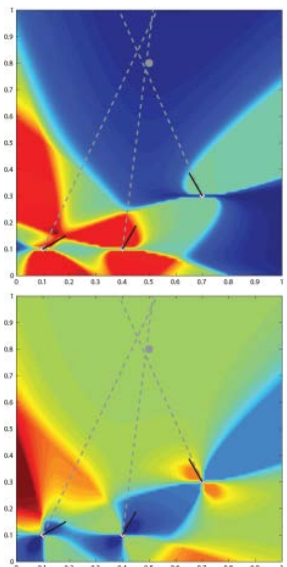


$$\mathbf{J}^\top \mathbf{J}\Delta \mathbf{x} = \mathbf{J}^\top \left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

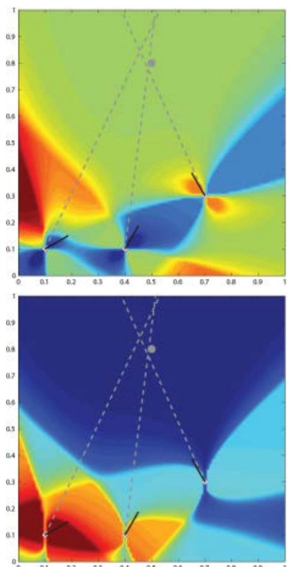
$$\mathbf{J}^\top \mathbf{J} =$$



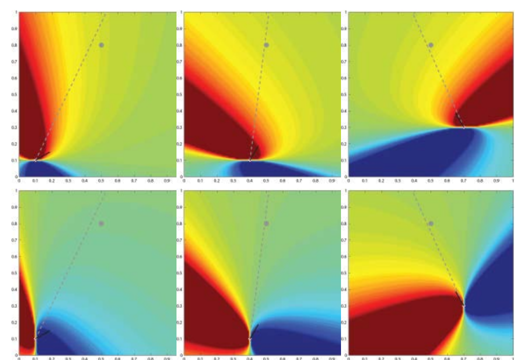
$$\mathbf{J}^\top \mathbf{J}\Delta\mathbf{x} = \mathbf{J}^\top\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

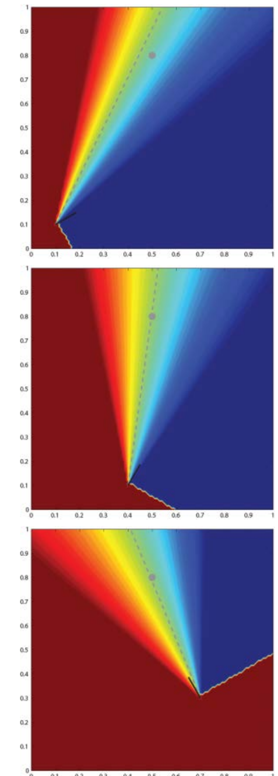


$\Delta X +$



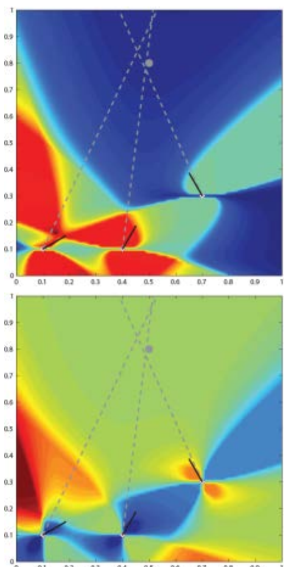
$\Delta Y =$



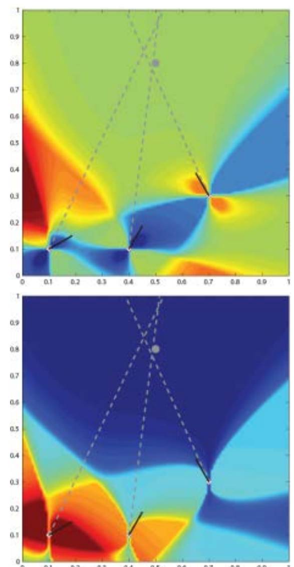


where $\Delta\mathbf{x} = \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix}$

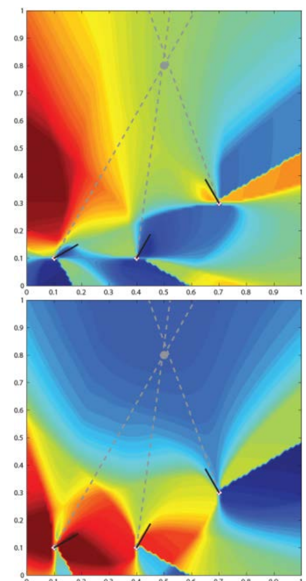
$$\mathbf{J}^\top \mathbf{J} \Delta \mathbf{x} = \mathbf{J}^\top \left(\mathbf{b} - \mathbf{f}(\mathbf{x}) \right)$$



$\Delta X +$

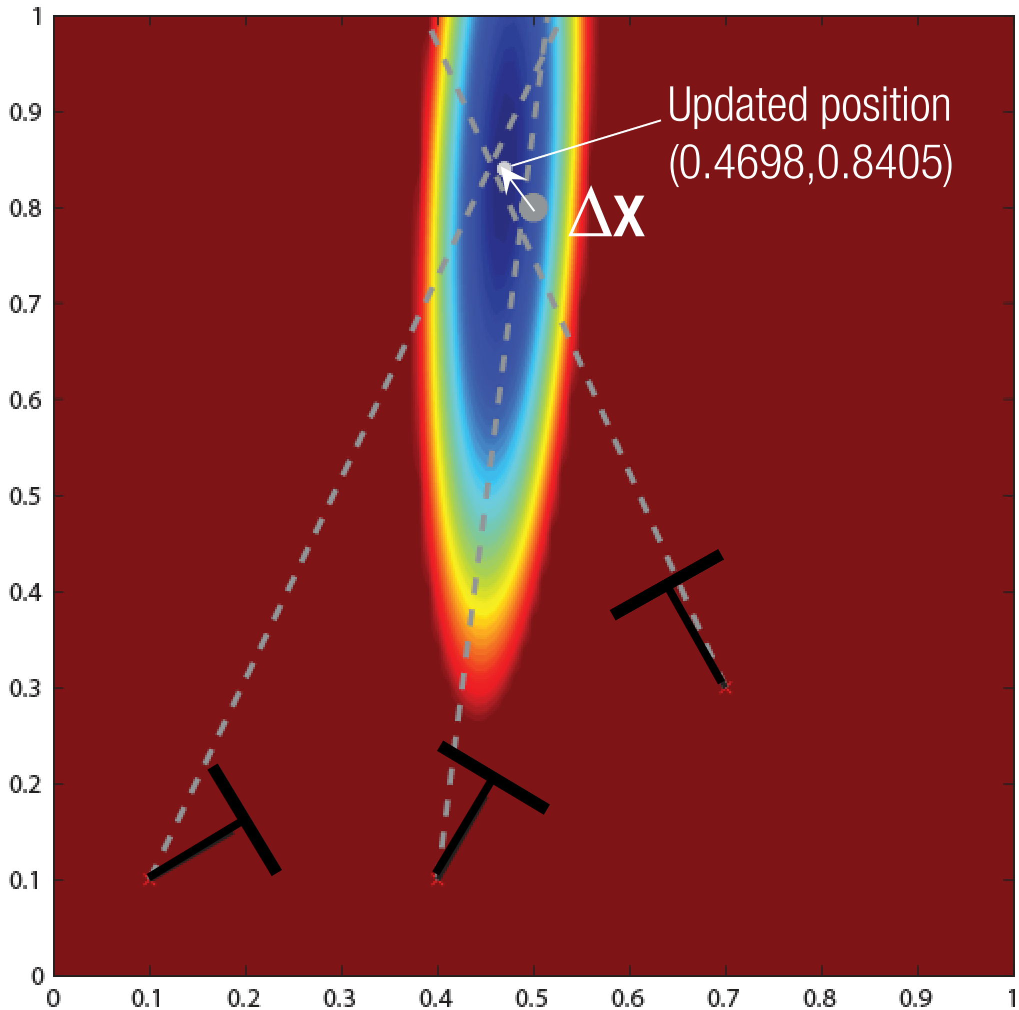


$\Delta Y =$



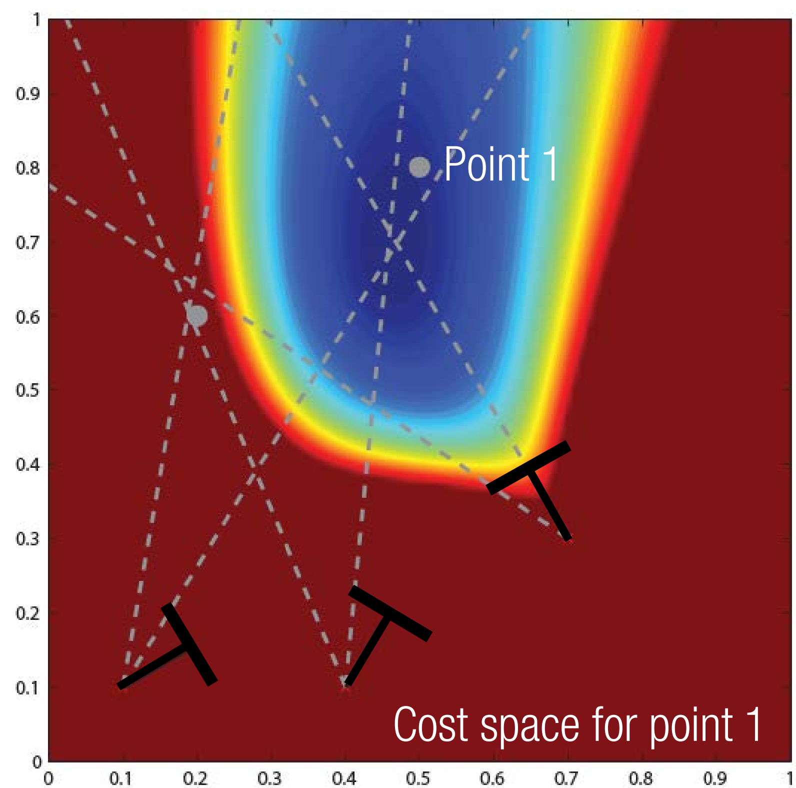
where $\Delta \mathbf{x} = \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix}$

$$\Delta \mathbf{x} = \left(\mathbf{J}^{\mathsf{T}}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$



$$\left\| \mathbf{J}^{\mathsf{T}}\mathbf{J}\Delta \mathbf{x} - \mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right) \right\|$$

$$
\begin{array}{cccccc}
 & \text{Camera 1} & \text{Camera 2} & \text{Camera 3} & \text{Point 1} & \text{Point 2} \\
\mathbf{J} = & \left[\begin{array}{c} \text{Camera 1} \\ \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} \\ \text{Camera 1} \\ \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} \end{array}\right. &
\begin{array}{c} \mathbf{0}_{1\times 3} \\ \text{Camera 2} \\ \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} \\ \text{Camera 2} \\ \mathbf{0}_{1\times 3} \end{array} &
\begin{array}{c} \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} \\ \text{Camera 3} \\ \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} \\ \text{Camera 3} \end{array} &
\begin{array}{c} \text{2D Point} \\ \text{2D Point} \\ \text{2D Point} \\ \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} \end{array} &
\left.\begin{array}{c} \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} \\ \text{2D Point} \\ \text{2D Point} \\ \text{2D Point} \end{array}\right]
\end{array}
\begin{array}{l} \text{Proj. point 1 to cam 1} \\ \text{Proj. point 1 to cam 2} \\ \text{Proj. point 1 to cam 3} \\ \\ \\ \\ \end{array}
$$

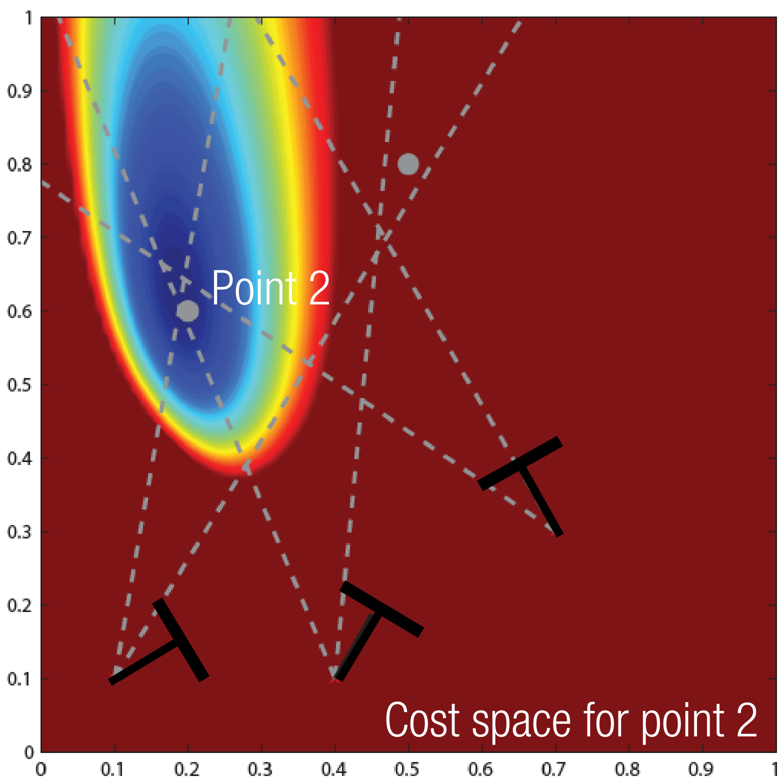


$$\sum_{i=1}^{3} \left\|\tilde{X}_{i1} - f_{i1}(\mathbf{X})\right\|^2$$

$$= \sum_{i=1}^{3} \left\|\tilde{X}_{i1} - u_{i1} / w_{i1}\right\|^2$$

$$\tilde{X}_{\underbrace{i}_{\text{Camera index}}\underbrace{1}_{\text{Point index}}}$$

$$
\mathbf{J} = \begin{array}{cl}
\begin{array}{ccccc} \text{Camera 1} & \text{Camera 2} & \text{Camera 3} & \text{Point 1} & \text{Point 2} \end{array} & \\
\left[\begin{array}{ccccc}
\text{Camera 1} & 0_{1\times3} & 0_{1\times3} & \text{2D Point} & 0_{1\times2} \\
0_{1\times3} & \text{Camera 2} & 0_{1\times3} & \text{2D Point} & 0_{1\times2} \\
0_{1\times3} & 0_{1\times3} & \text{Camera 3} & \text{2D Point} & 0_{1\times2} \\
\text{Camera 1} & 0_{1\times3} & 0_{1\times3} & 0_{1\times2} & \text{2D Point} \\
0_{1\times3} & \text{Camera 2} & 0_{1\times3} & 0_{1\times2} & \text{2D Point} \\
0_{1\times3} & 0_{1\times3} & \text{Camera 3} & 0_{1\times2} & \text{2D Point}
\end{array}\right] & \begin{array}{l} \\ \\ \\ \text{Proj. point 2 to cam 1} \\ \text{Proj. point 2 to cam 2} \\ \text{Proj. point 2 to cam 3} \end{array}
\end{array}
$$

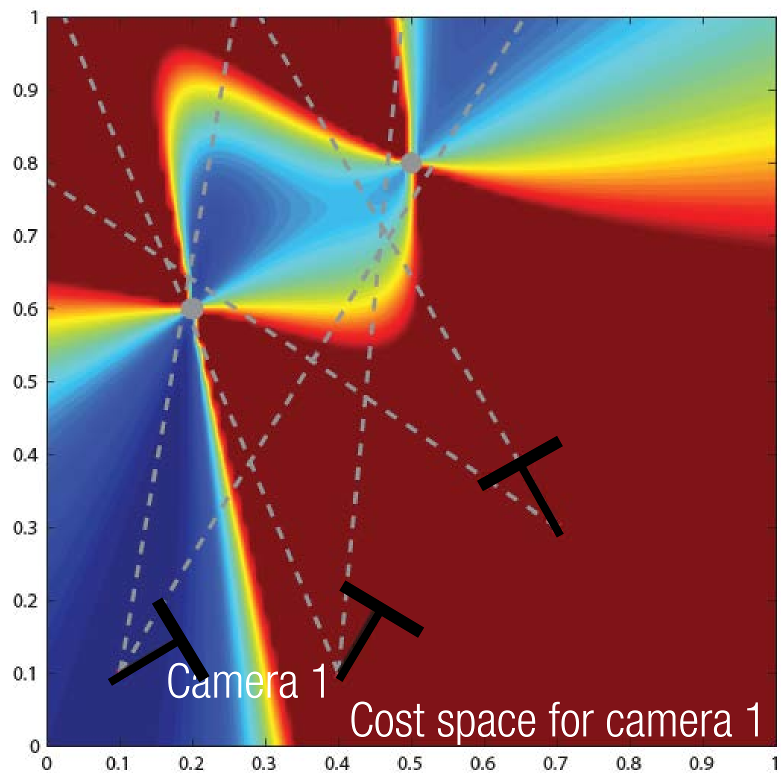


$$\sum_{i=1}^{3} \left\| \tilde{x}_{i2} - f_{i2}(\mathbf{X}) \right\|^2$$

$$= \sum_{i=1}^{3} \left\| \tilde{x}_{i2} - u_{i2} / w_{i2} \right\|^2$$

$$\tilde{x}_{\underbrace{i}_{\text{Camera index}}\underbrace{2}_{\text{Point index}}}$$

$$
\begin{array}{cccccccc}
 & & \text{Camera 1} & \text{Camera 2} & \text{Camera 3} & \text{Point 1} & \text{Point 2} & \\
\mathbf{J} = & \left[\right. & \text{Camera 1} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \text{2D Point} & \mathbf{0}_{1\times2} & \left. \right] \quad \text{Proj. point 1 to cam 1} \\
 & & \mathbf{0}_{1\times3} & \text{Camera 2} & \mathbf{0}_{1\times3} & \text{2D Point} & \mathbf{0}_{1\times2} & \\
 & & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \text{Camera 3} & \text{2D Point} & \mathbf{0}_{1\times2} & \\
 & & \text{Camera 1} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} & \text{2D Point} & \quad \text{Proj. point 2 to cam 1} \\
 & & \mathbf{0}_{1\times3} & \text{Camera 2} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} & \text{2D Point} & \\
 & & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \text{Camera 3} & \mathbf{0}_{1\times2} & \text{2D Point} &
\end{array}
$$

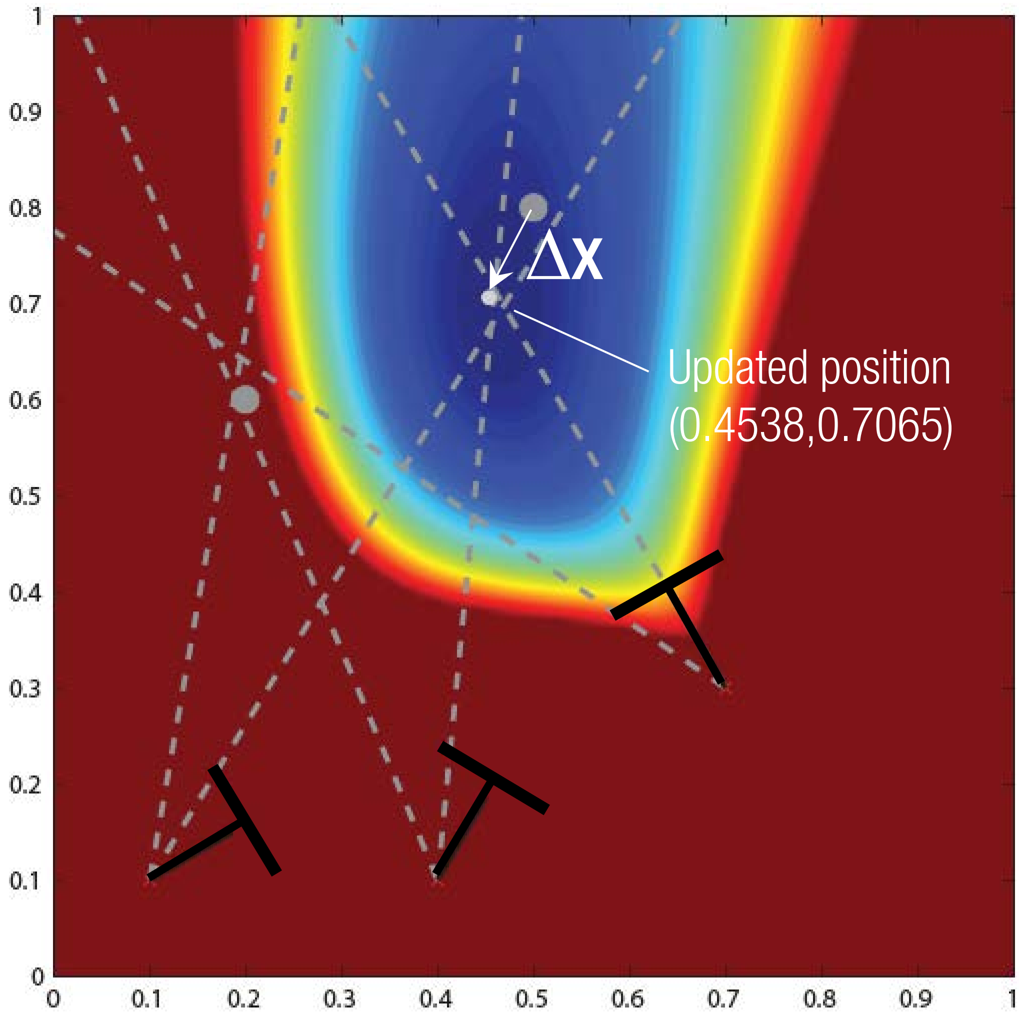


$$\sum_{j=1}^{2} \left\| \tilde{x}_{1j} - f_{1j}(\mathbf{X}) \right\|^2$$

$$= \sum_{j=1}^{2} \left\| \tilde{x}_{1j} - u_{1j} / w_{1j} \right\|^2$$

$$\tilde{X}_{\underbrace{1}_{\text{Camera index}}\underbrace{j}_{\text{Point index}}}$$

$$\Delta \mathbf{x} = \left(\mathbf{J}^{\mathrm{T}}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

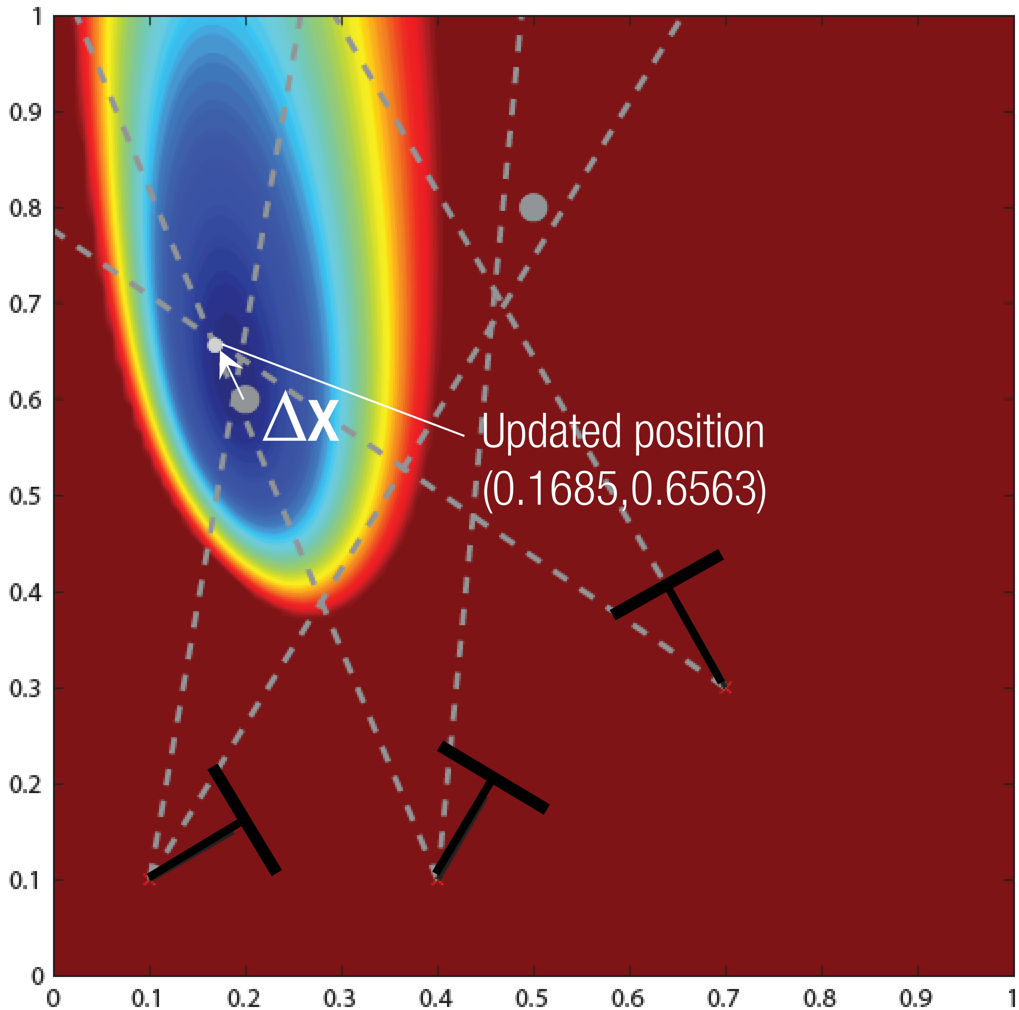


$$\sum_{i=1}^{3}\left\|\tilde{x}_{i1} - f_{i1}(\mathbf{X})\right\|^2$$

$$= \sum_{i=1}^{3}\left\|\tilde{x}_{i1} - u_{i1} / w_{i1}\right\|^2$$

$$\tilde{x}_{\underbrace{i}_{\text{Camera index}}\underbrace{1}_{\text{Point index}}}$$

$$\Delta \mathbf{x} = \left(\mathbf{J}^{\mathrm{T}}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

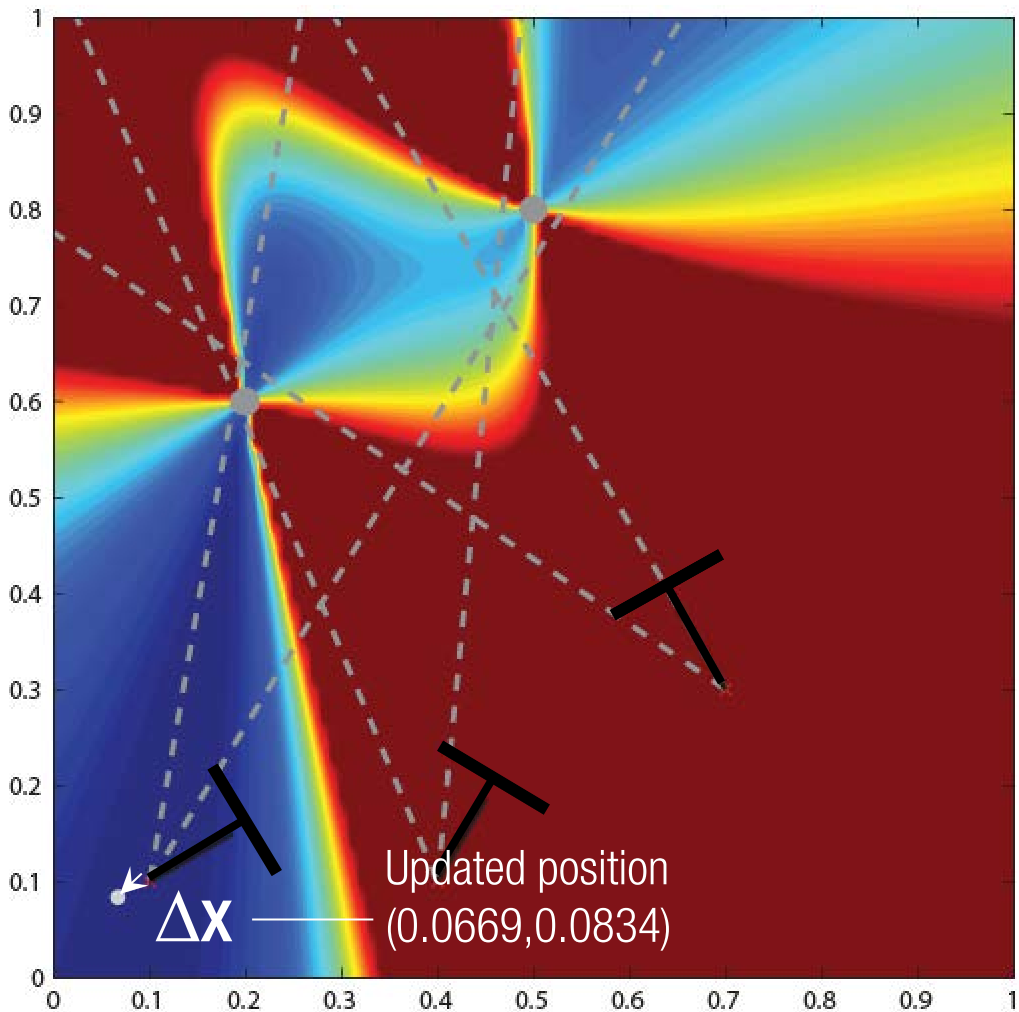


$$\sum_{i=1}^{3}\left\|\tilde{x}_{i2} - f_{i2}(\mathbf{X})\right\|^2$$

$$= \sum_{i=1}^{3}\left\|\tilde{x}_{i2} - u_{i2}/w_{i2}\right\|^2$$

$$\tilde{x}_{\underbrace{i}_{\text{Camera index}}\underbrace{2}_{\text{Point index}}}$$

$$\Delta \mathbf{x} = \left(\mathbf{J}^{\mathsf{T}}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

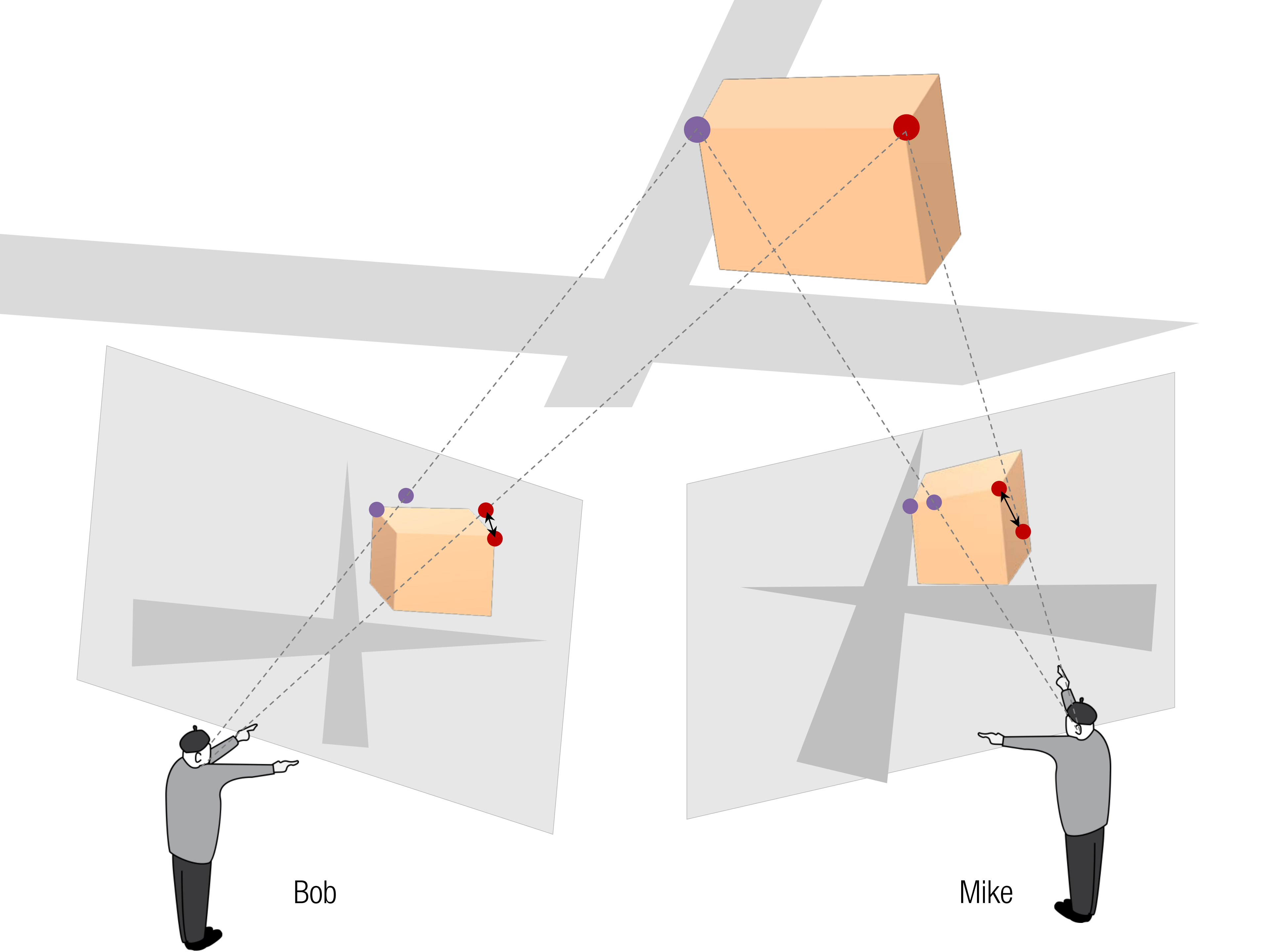


$$\sum_{j=1}^{2}\left\|\tilde{x}_{1j} - f_{1j}(\mathbf{X})\right\|^2$$

$$= \sum_{j=1}^{2}\left\|\tilde{x}_{1j} - u_{1j} / w_{1j}\right\|^2$$

$$\tilde{x}_{1j}$$

Camera index   Point index


Bob
Mike

$$\mathbf{J} = \begin{bmatrix} \underset{2\text{x}9}{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}} \underset{9\text{x}4}{\dfrac{\partial \mathbf{R}}{\partial \mathbf{q}}} & \underset{2\text{x}3}{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}} & \underset{2\text{x}3}{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}} \end{bmatrix}$$

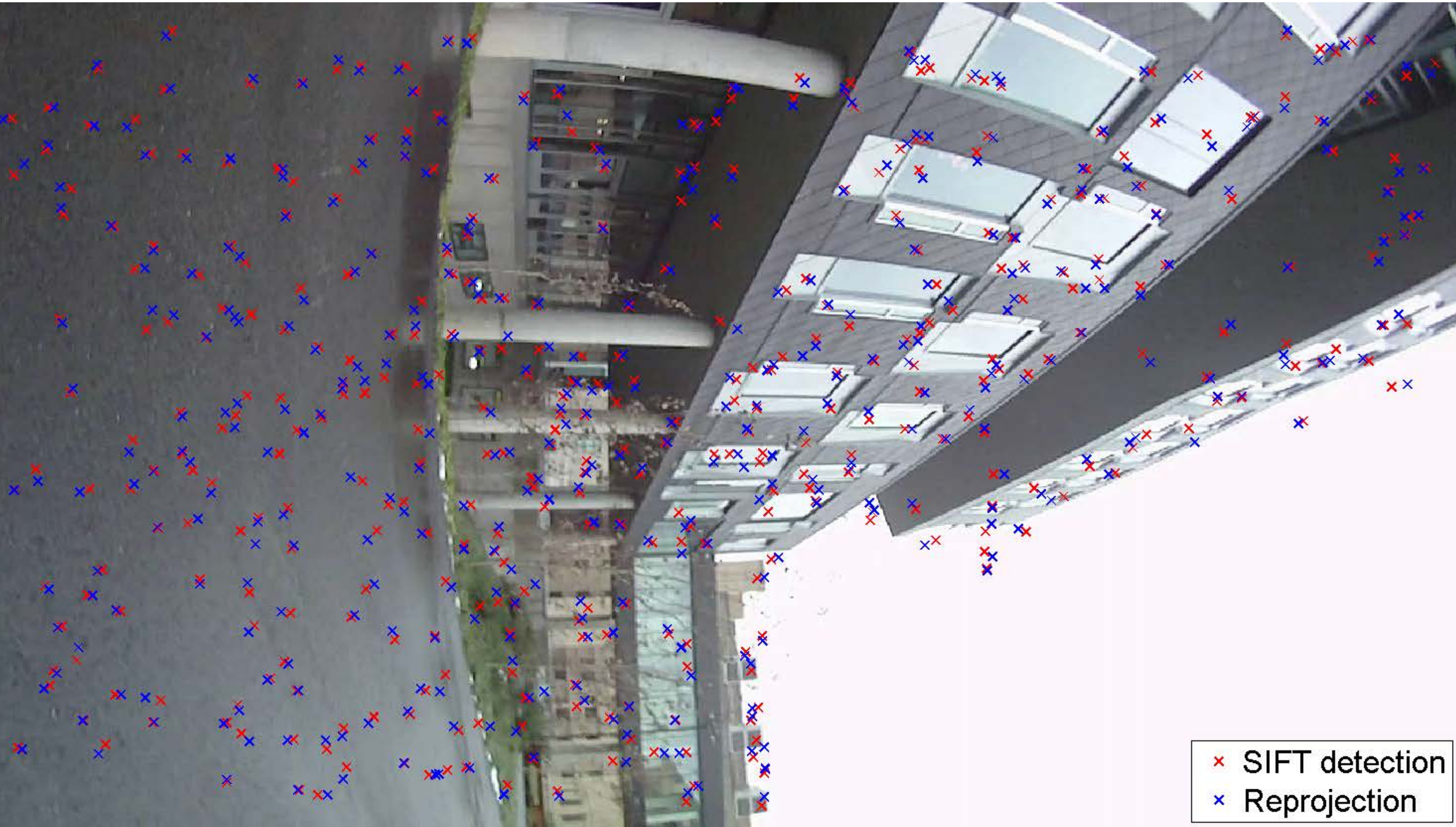
$$\mathbf{J} = \begin{bmatrix} \text{Bob's Jacobian} & \mathbf{0}_{2\times 7} & \text{3D Point} \\ \mathbf{0}_{2\times 7} & \text{Mike's Jacobian} & \text{3D Point} \end{bmatrix}$$

$$J = \left[\underbrace{\frac{\partial f(R(q), C, X)}{\partial R}}_{2\text{x}9}\underbrace{\frac{\partial R}{\partial q}}_{9\text{x}4} \quad \underbrace{\frac{\partial f(R(q), C, X)}{\partial C}}_{2\text{x}3} \quad \underbrace{\frac{\partial f(R(q), C, X)}{\partial X}}_{2\text{x}3}\right]$$

$$J = \begin{bmatrix} \text{Bob's Jacobian} & \mathbf{0}_{2\times 7} & \text{3D Point} & \mathbf{0}_{2\times 3} \\ \mathbf{0}_{2\times 7} & \text{Mike's Jacobian} & \text{3D Point} & \mathbf{0}_{2\times 3} \\ \text{Bob's Jacobian} & \mathbf{0}_{2\times 7} & \mathbf{0}_{2\times 3} & \text{3D Point} \\ \mathbf{0}_{2\times 7} & \text{Mike's Jacobian} & \mathbf{0}_{2\times 3} & \text{3D Point} \end{bmatrix}$$

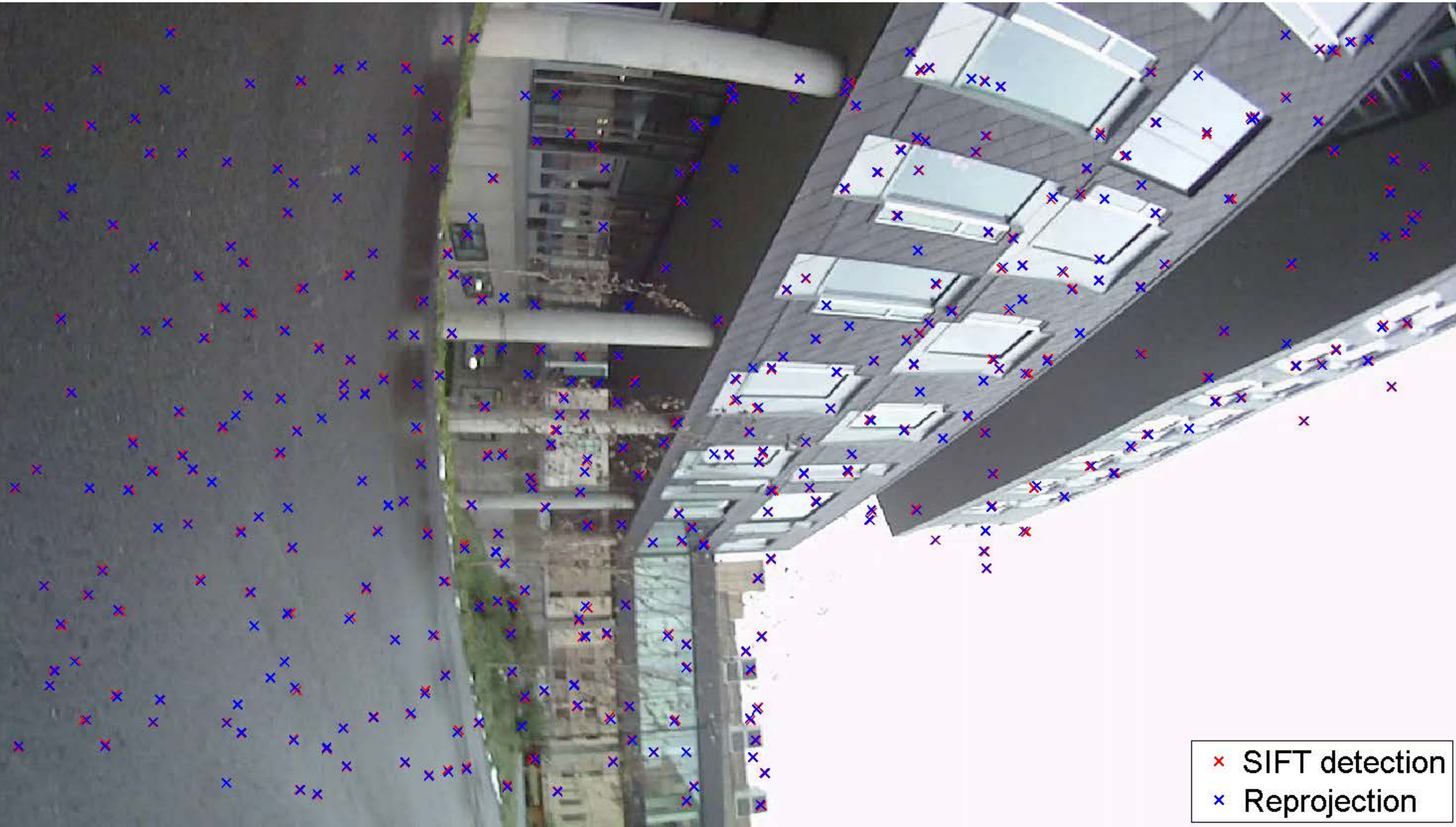
Geometric Refinement

Before Bundle Adjustment



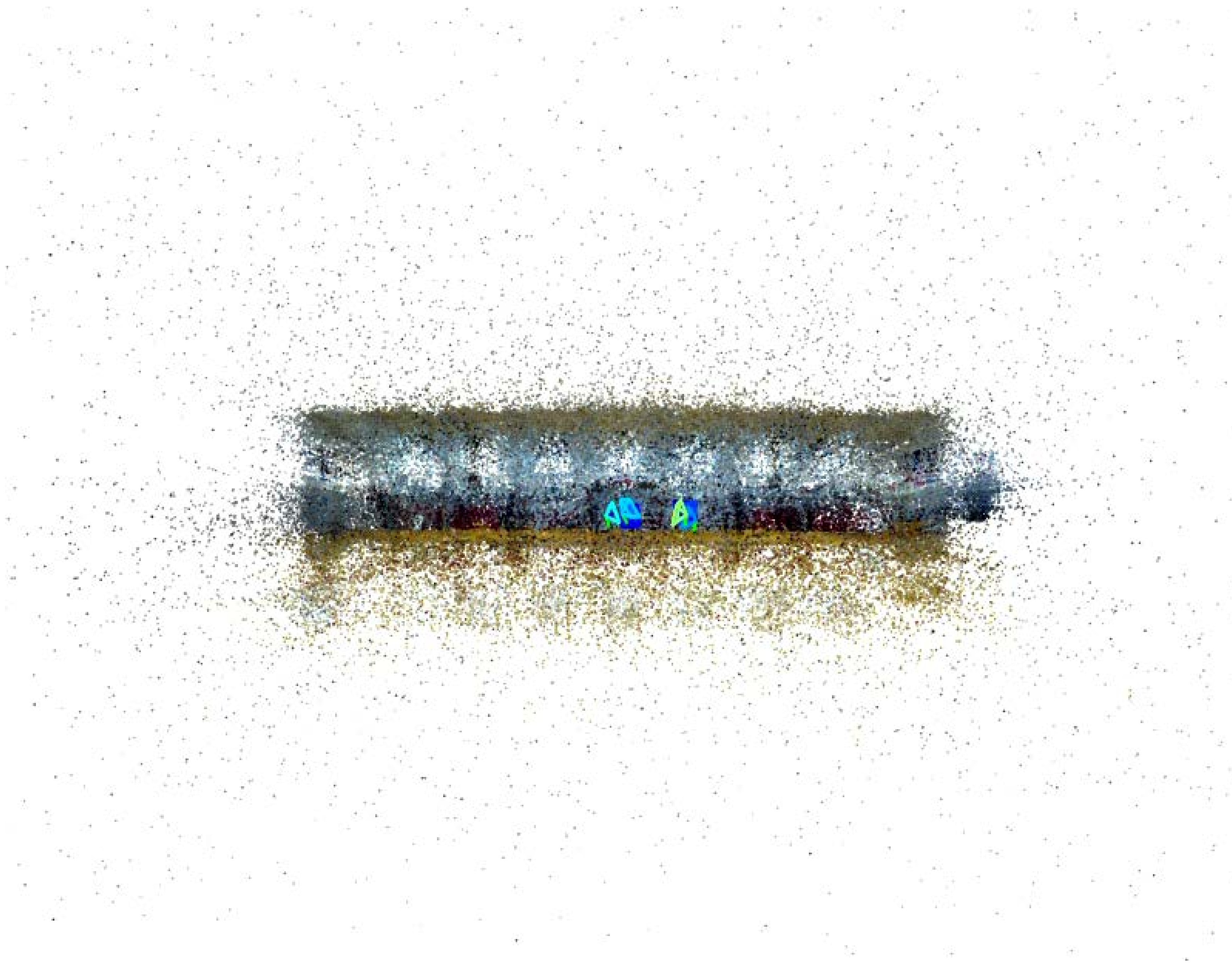
Geometric Refinement

After Bundle Adjustment

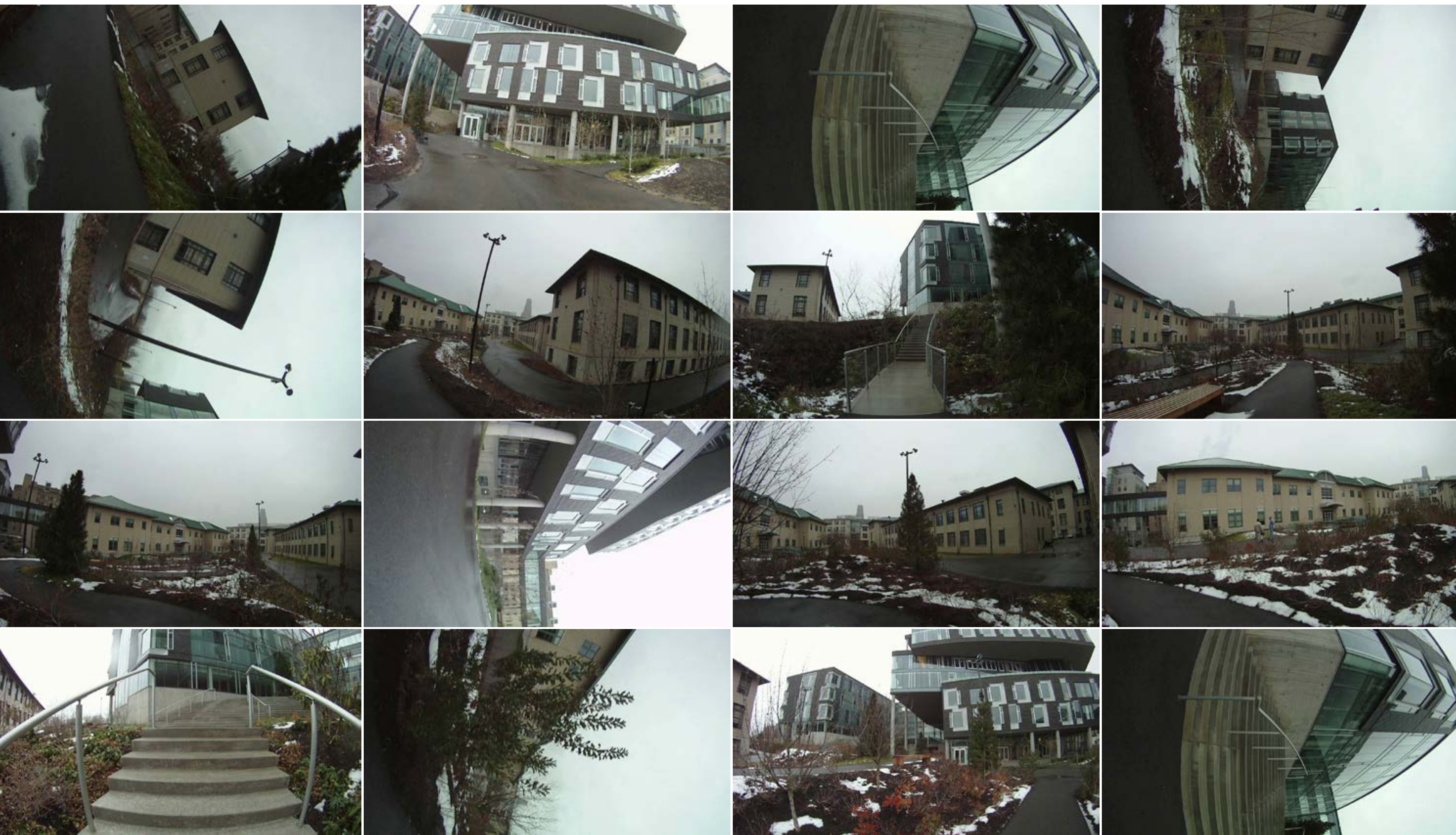


Structure from Motion Pipeline

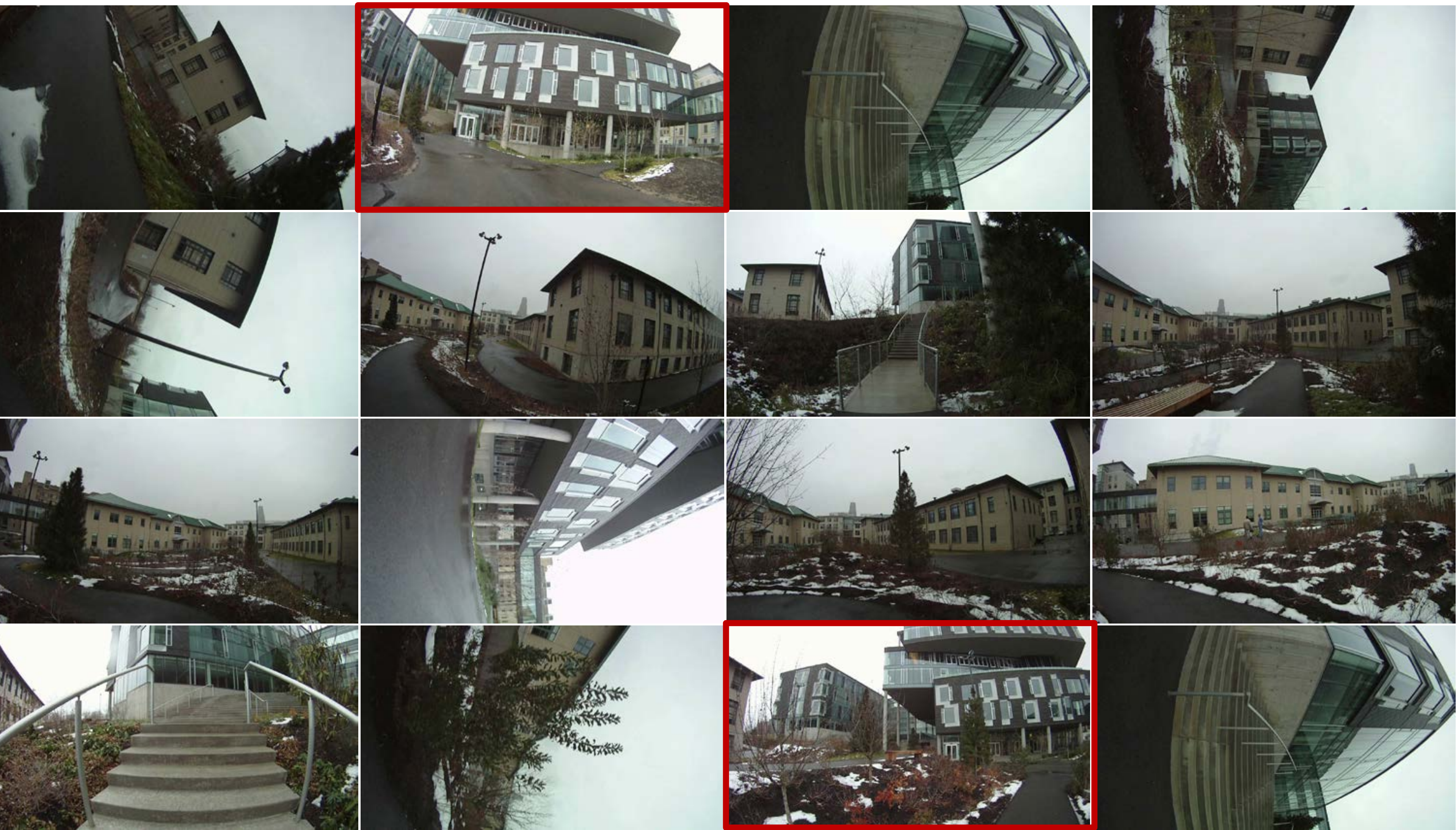
Where am I (camera)?
Where are they (points)?



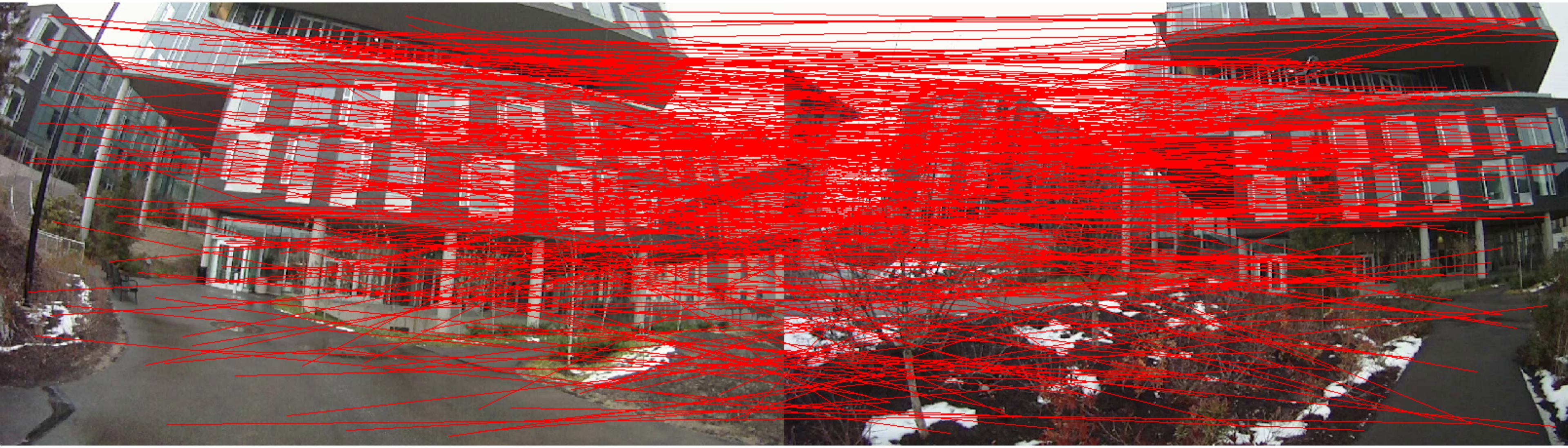
Input Images

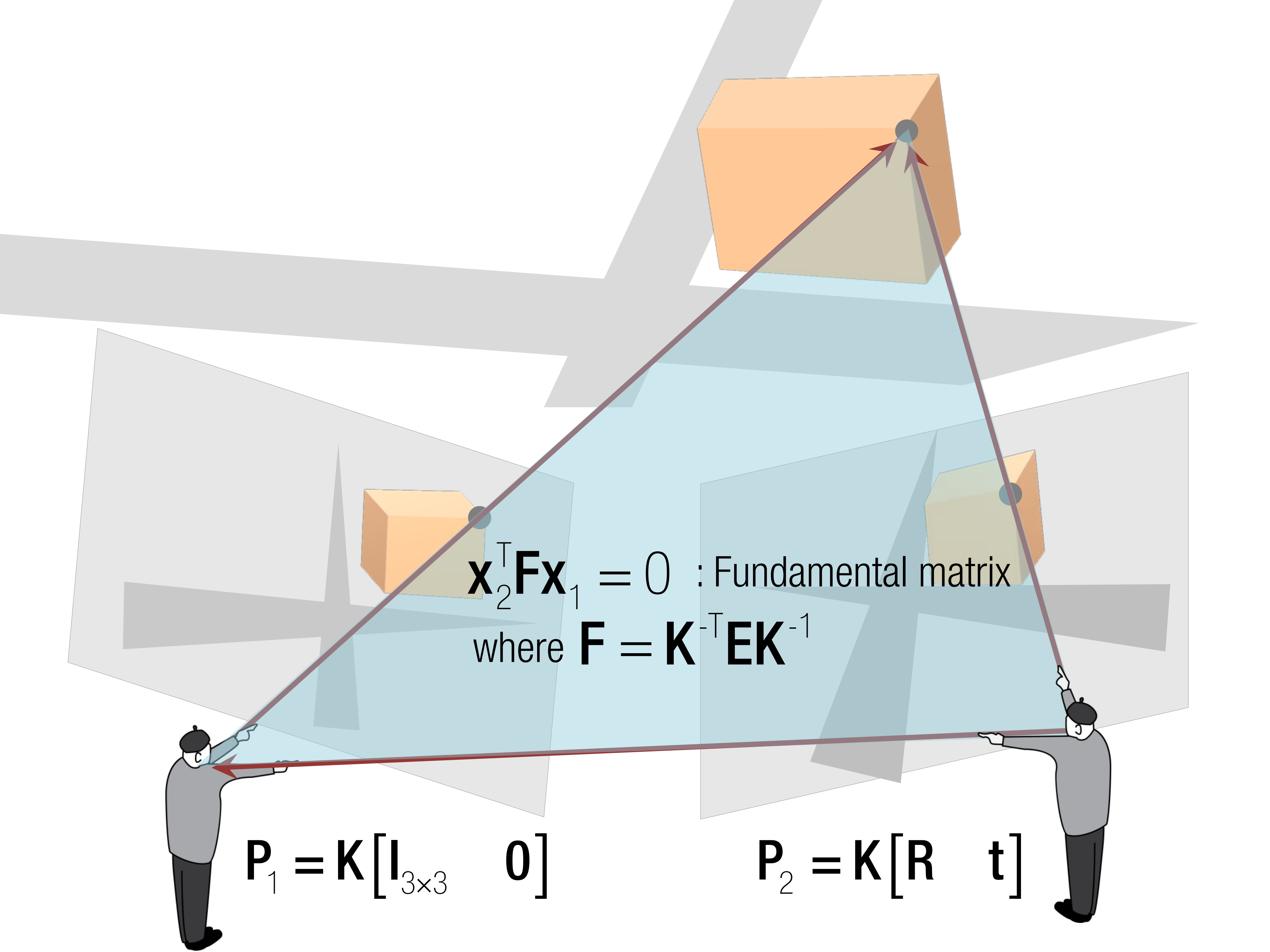


Initial Pair Images



1. Pairwise Image Feature Matching





$\mathbf{x}_2^{T}\mathbf{F}\mathbf{x}_1 = 0$: Fundamental matrix
where $\mathbf{F} = \mathbf{K}^{-T}\mathbf{E}\mathbf{K}^{-1}$
$\mathbf{P}_1 = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}\end{bmatrix}$
$\mathbf{P}_2 = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

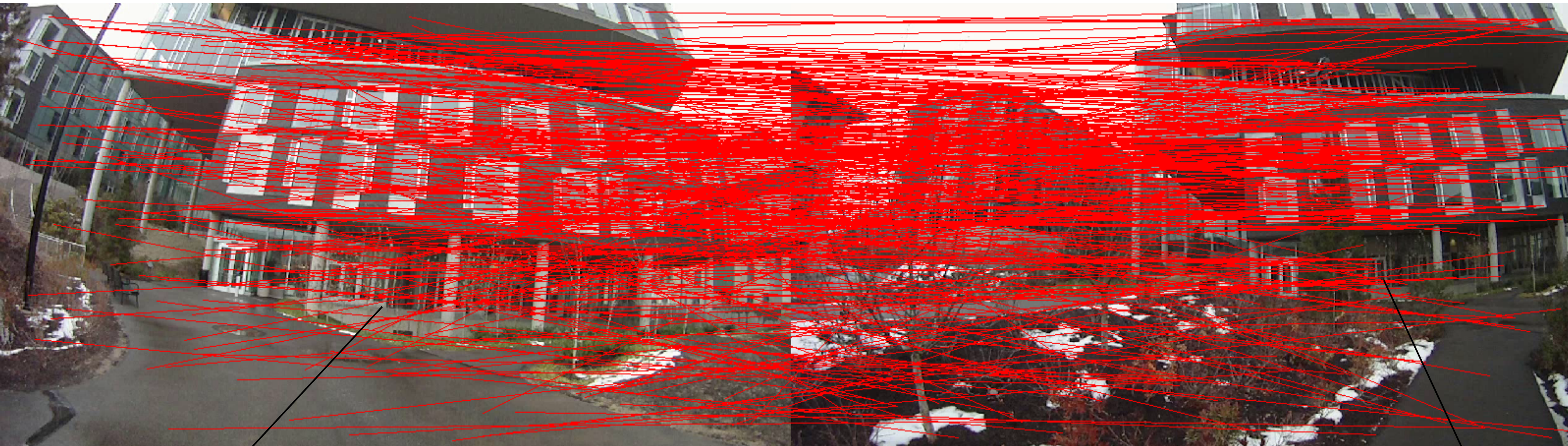
1. Pairwise Image Feature Matching

$\mathbf{x}_1$ $\mathbf{x}_2$

$$\mathbf{x}_2^{\top}\mathbf{F}\mathbf{x}_1 = 0$$

Epipolar constraint

1. Pairwise Image Feature Matching



$\mathbf{x}_1$ $\mathbf{x}_2$

$$\mathbf{x}_2^{\mathsf{T}}\mathbf{F}\mathbf{x}_1 = 0$$

Epipolar constraint

# of unknowns: 8
# of required equations: 8

1. Pairwise Image Feature Matching

$\mathbf{x}_1$ $\mathbf{x}_2$

$$\mathbf{x}_2^{\mathsf{T}}\mathbf{F}\mathbf{x}_1 = 0$$

Epipolar constraint

\# of unknowns: 8
\# of required equations: 8

$$\mathbf{x}_{2,1}^{\mathsf{T}}\mathbf{F}\mathbf{x}_{1,1} = 0$$

$$\vdots$$

$$\mathbf{x}_{2,8}^{\mathsf{T}}\mathbf{F}\mathbf{x}_{1,8} = 0$$

1. Pairwise Image Feature Matching

$\mathbf{x}_1$

$\mathbf{x}_2$

$$\mathbf{x}_2^{\mathsf{T}}\mathbf{F}\mathbf{x}_1 = 0$$

Epipolar constraint

$$\underbrace{\begin{bmatrix} u_1^1u_1^2 & u_1^1v_1^2 & u_1^1 & v_1^1u_1^2 & v_1^1v_1^2 & v_1^1 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_8^1u_8^2 & u_8^1v_8^2 & u_8^1 & v_8^1u_8^2 & v_8^1v_8^2 & v_8^1 & u_8^2 & v_8^2 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}}_{\mathbf{X}} = \mathbf{0}$$

8 Point Algorithm

- Construct 8x9 matrix $\mathbf{A}$.
- Solving linear homogeneous equations via SVD:

  $\mathbf{x} = \mathbf{V}_{:,8}$ where $\mathbf{A} = \mathbf{UDV}^{\mathrm{T}}$

  $\mathbf{F} = \mathrm{reshape}(\mathbf{x}, 3, 3)$: constructing matrix from vector.
- Applying rank constraint, i.e., $\mathrm{rank}(\mathbf{F}) = 2$.

$\mathbf{F}_{\mathrm{rank2}} = \mathbf{U}\tilde{\mathbf{D}}\mathbf{V}^{\mathrm{T}}$ where $\tilde{\mathbf{D}}$:$\mathbf{D}$ with the last element zero.

SVD cleanup

Epipolar Geometry

$\mathbf{x}_1$ $\mathbf{x}_2$

$$\mathbf{x}_2^\top \mathbf{F} \mathbf{x}_1 = 0$$

2. Outlier Rejection via RANSAC

$\mathbf{x}_1$

$\mathbf{x}_2$

$\mathbf{x}_2^{\top}\mathbf{F}\mathbf{x}_1 = 0$

Random sampling

Model building

Thresholding

Inlier counting

2. Outlier Rejection via RANSAC

$\mathbf{x}_1$ $\mathbf{x}_2$

$$\mathbf{x}_2^{\mathrm{T}}\mathbf{F}\mathbf{x}_1 = 0$$

Random sampling

Model building

Thresholding

Inlier counting

2. Outlier Rejection via RANSAC

$\mathbf{x}_1$ $\mathbf{x}_2$

$$\mathbf{x}_2^\top \mathbf{F} \mathbf{x}_1 = 0$$

Random sampling

Model building

Thresholding

Inlier counting

2. Outlier Rejection via RANSAC

$\mathbf{x}_1$

$\mathbf{x}_2$

$$\mathbf{x}_2^{\mathrm{T}}\mathbf{F}\mathbf{x}_1 = 0$$

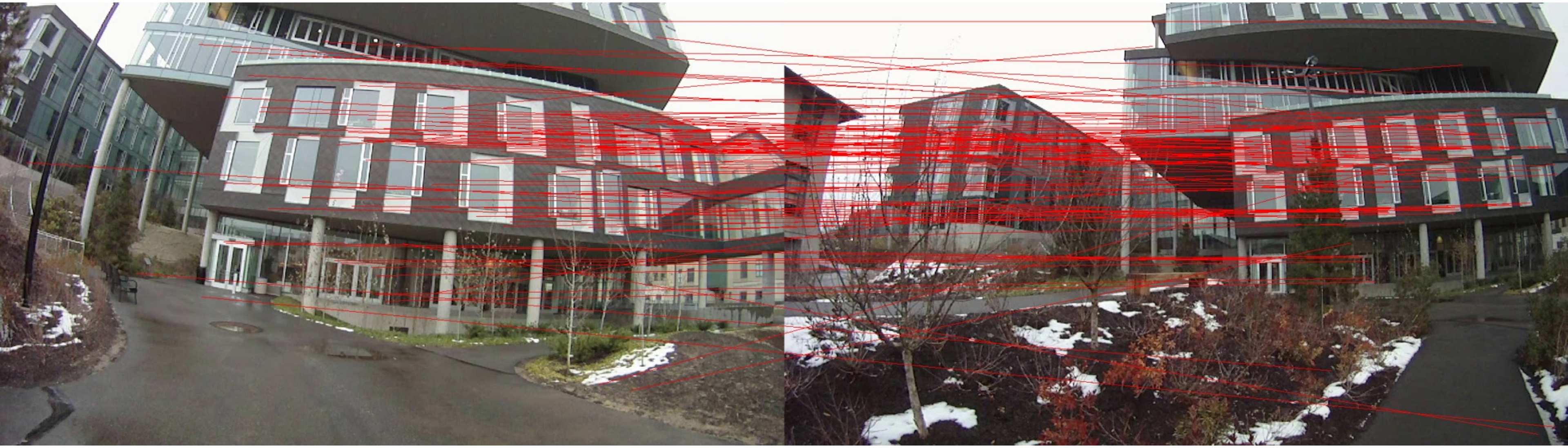
Random sampling

Model building

Thresholding

Inlier counting

2. Outlier Rejection via RANSAC



$$\mathbf{x}_2^{\mathsf{T}}\mathbf{F}\mathbf{x}_1 = 0$$

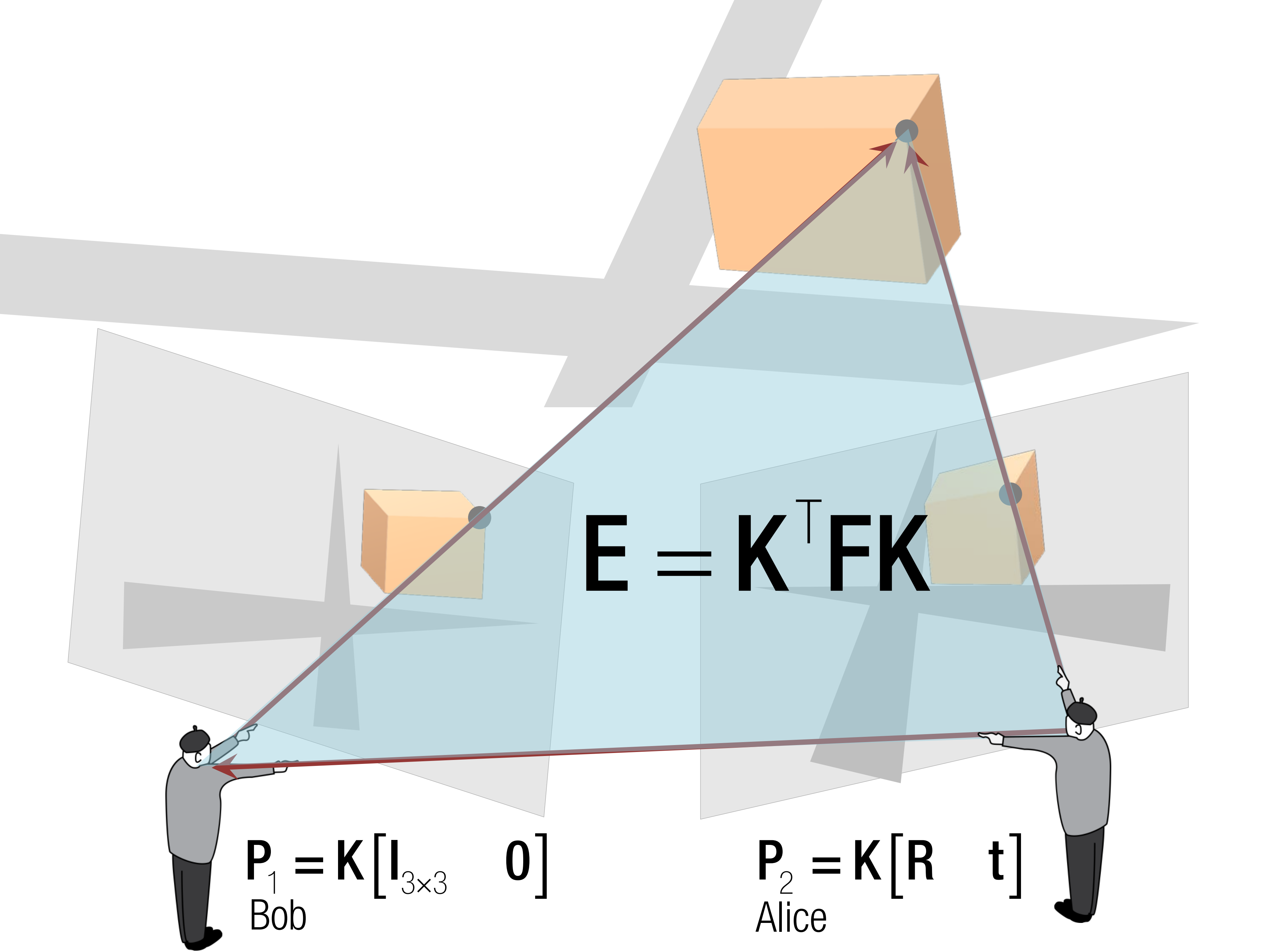
# of inliers: 253

Random sampling

Model building

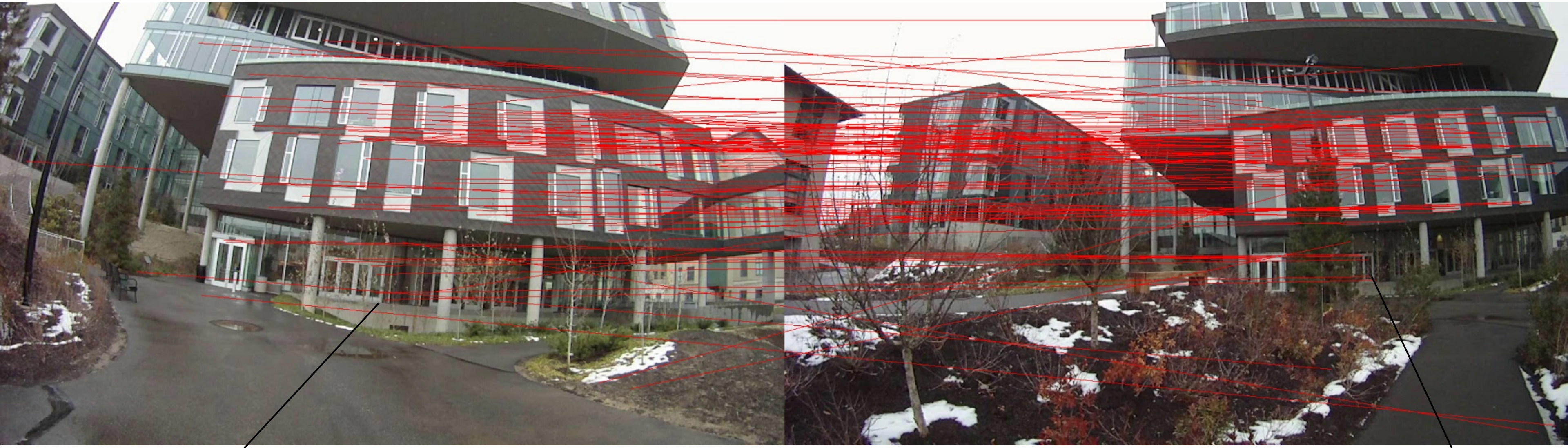
Thresholding

Inlier counting



$E = K^{\top}FK$
$P_1 = K[I_{3\times3} \quad 0]$
Bob
$P_2 = K[R \quad t]$
Alice

3. Essential Matrix Computation



$\mathbf{x}_1$

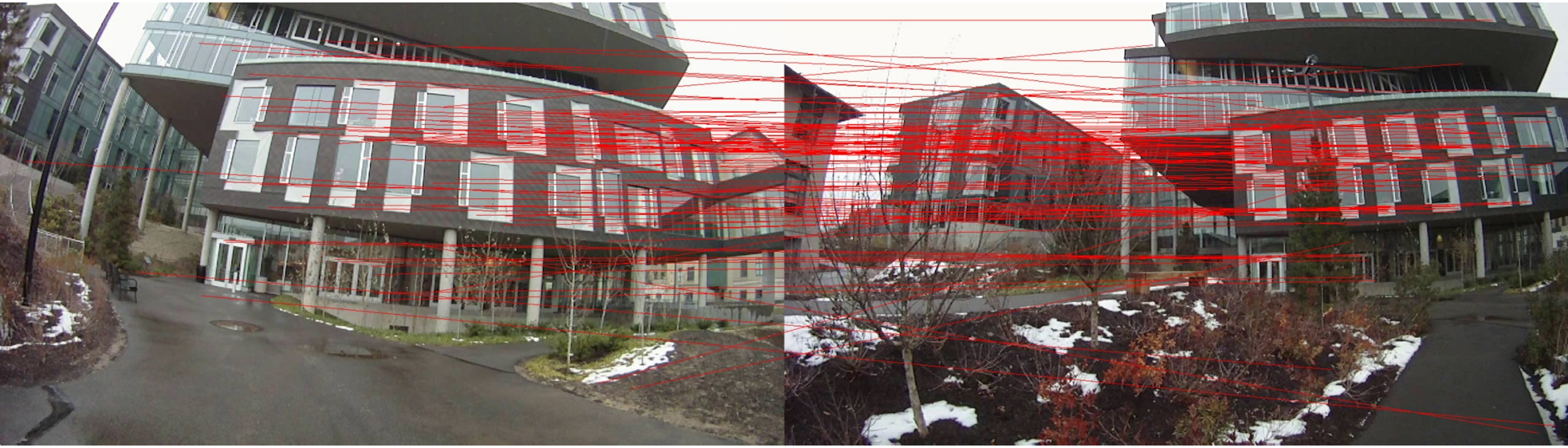
$\mathbf{x}_2$

$$\mathbf{x}_2^T \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1} \mathbf{x}_1 = 0$$

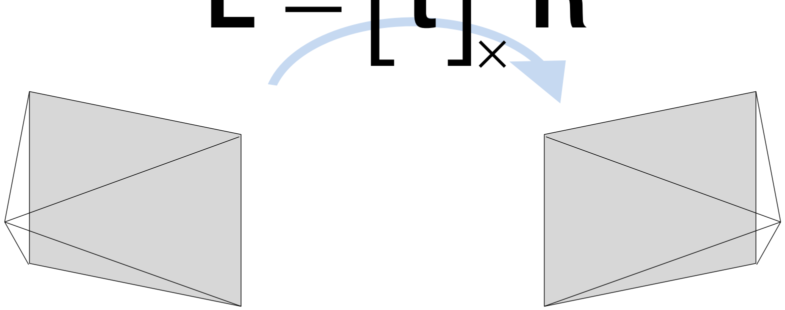
$$\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0$$

$$\longrightarrow \quad \mathbf{K}^T \mathbf{F} \mathbf{K} = \mathbf{E}$$

4. Relative Transform from Essential Matrix



$$E = [t]_\times R$$



$$P_1 = \left[I_{3\times3} \mid 0_3 \right] \qquad P_2 = \left[R \mid t \right]$$

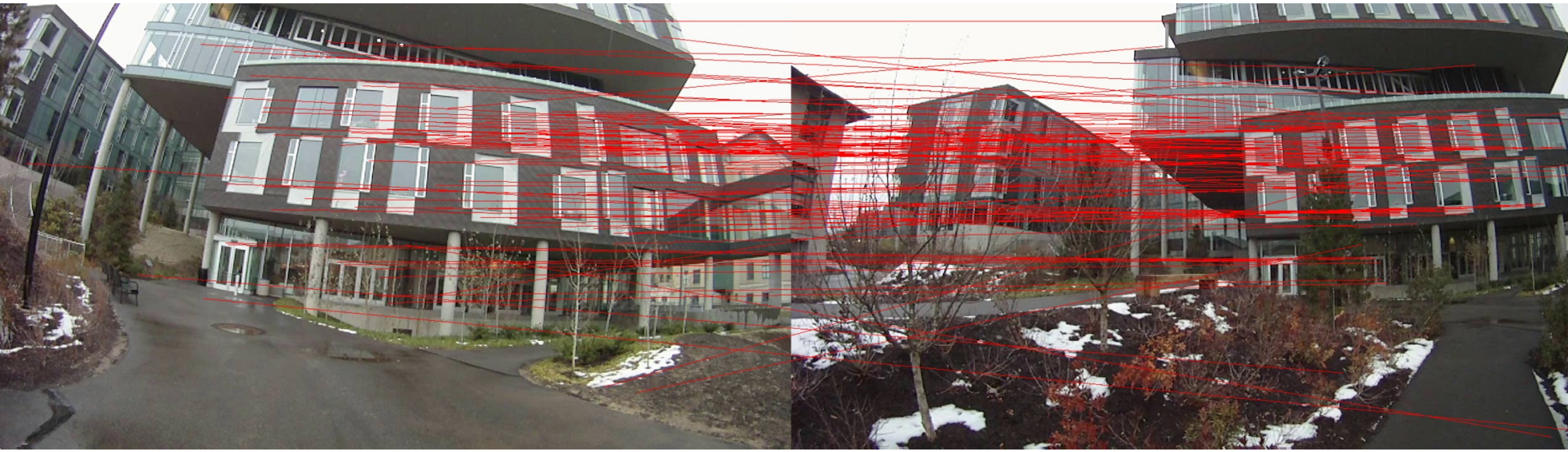
$$P_2 = \left[UYV^\top \mid u_3 \right]$$

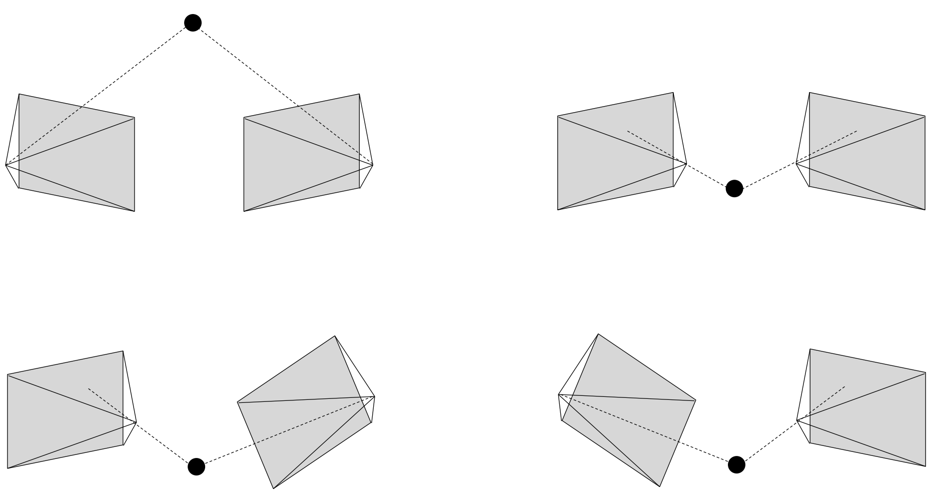
$$\left[UY^\top V^\top \mid u_3 \right]$$

$$\left[UYV^\top \mid -u_3 \right]$$

$$\left[UY^\top V^\top \mid -u_3 \right]$$

4. Relative Transform from Essential Matrix





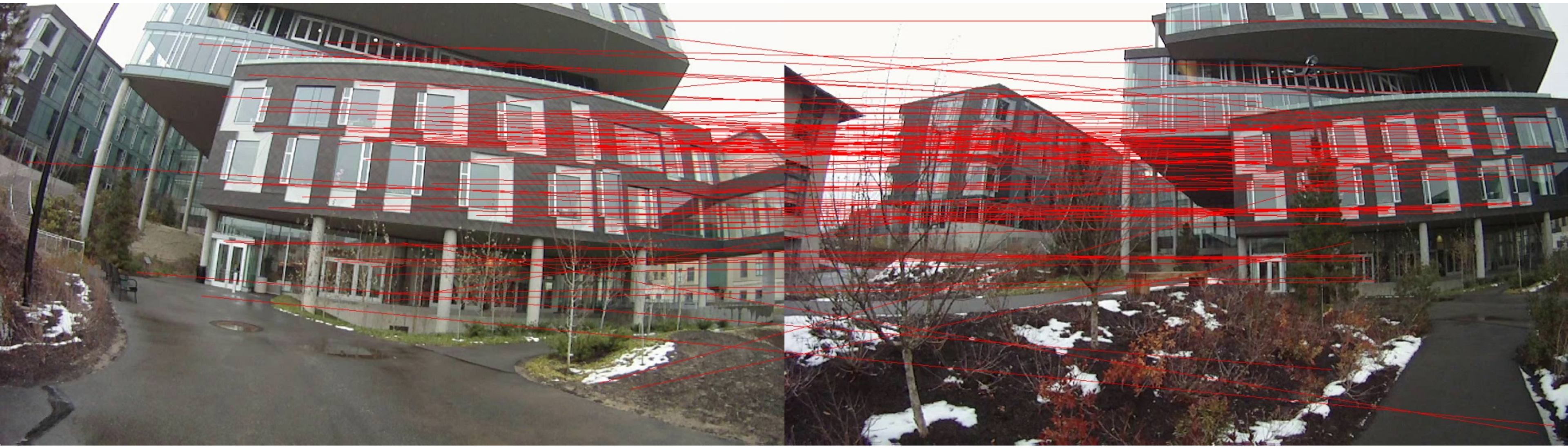
$$P_2 = \left[\, UYV^\top \;\middle|\; u_3 \,\right]$$

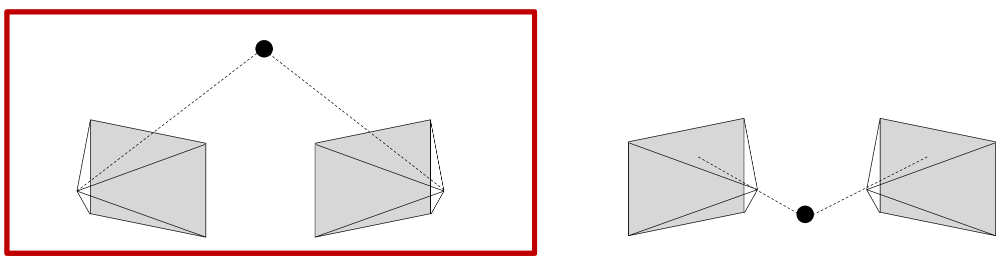
$$\left[\, UY^\top V^\top \;\middle|\; u_3 \,\right]$$

$$\left[\, UYV^\top \;\middle|\; -u_3 \,\right]$$

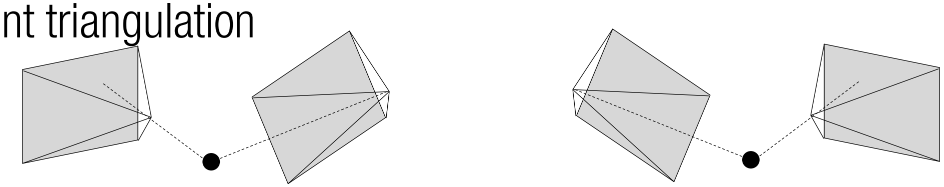
$$\left[\, UY^\top V^\top \;\middle|\; -u_3 \,\right]$$

4. Relative Transform from Essential Matrix





Correct configuration resolved via point triangulation



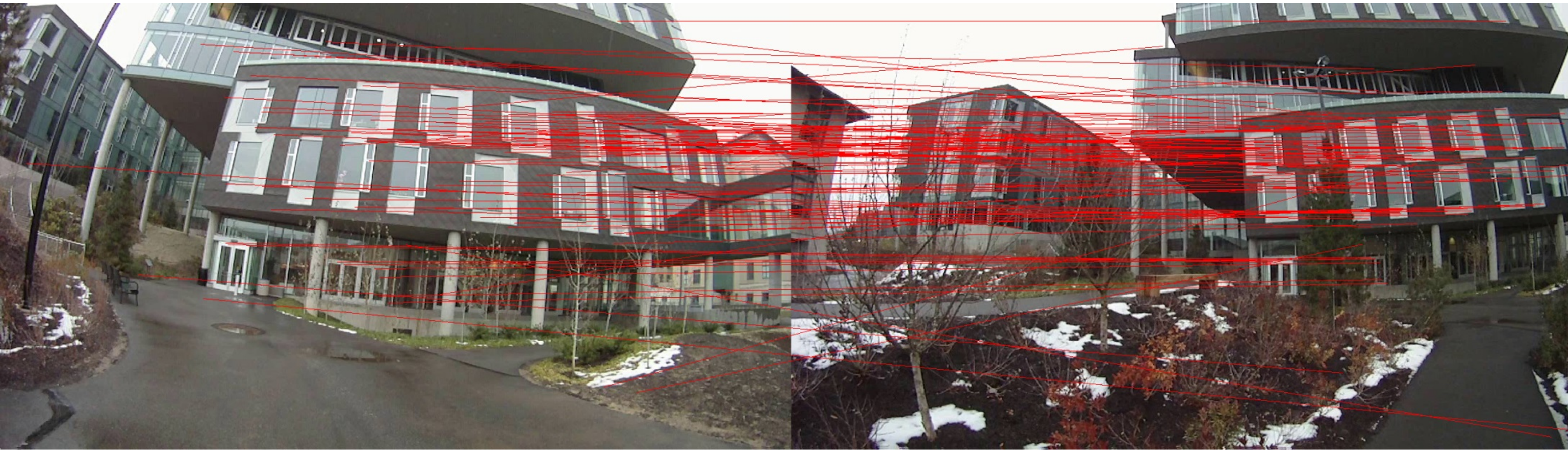
$$P_2 = \left[UYV^\top \mid u_3 \right]$$

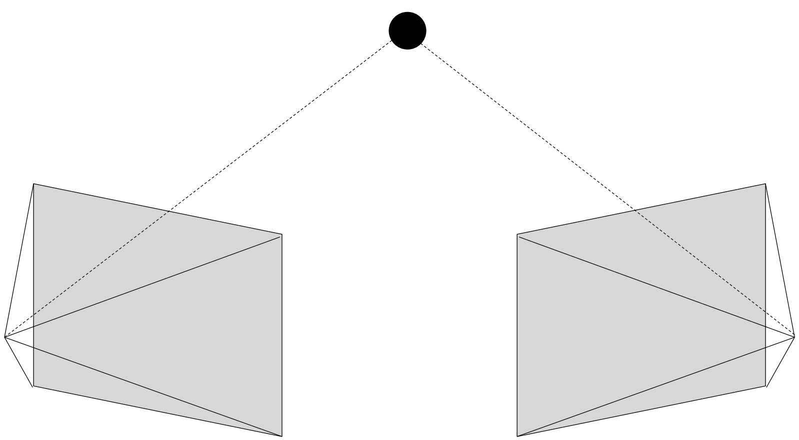
$$\left[UY^\top V^\top \mid u_3 \right]$$

$$\left[UYV^\top \mid -u_3 \right]$$

$$\left[UY^\top V^\top \mid -u_3 \right]$$

5. Point Triangulation

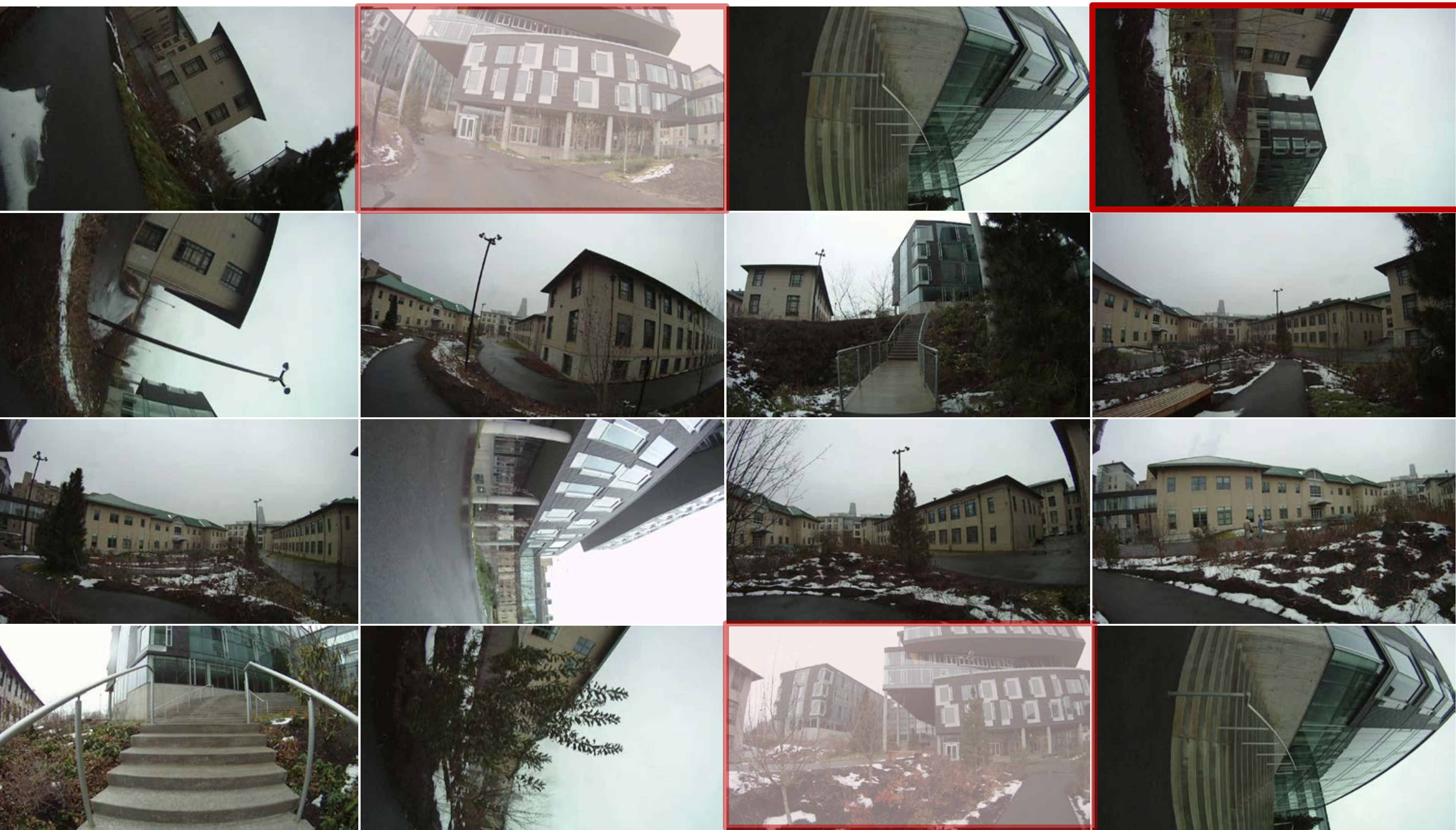




$$\begin{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times \mathbf{P}_1 \\ \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times \mathbf{P}_2 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

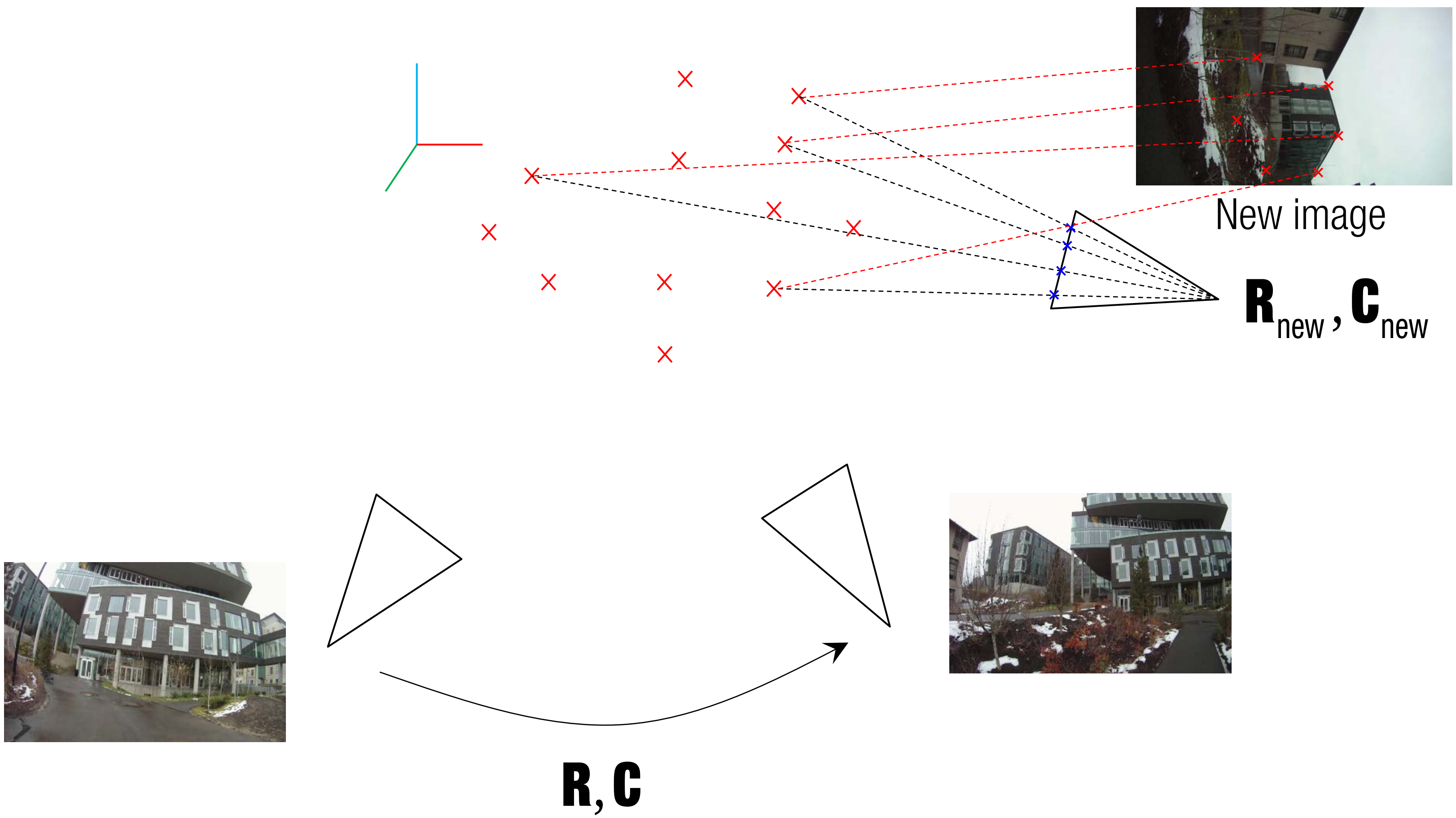
Initial Pair Images

Adding New Image



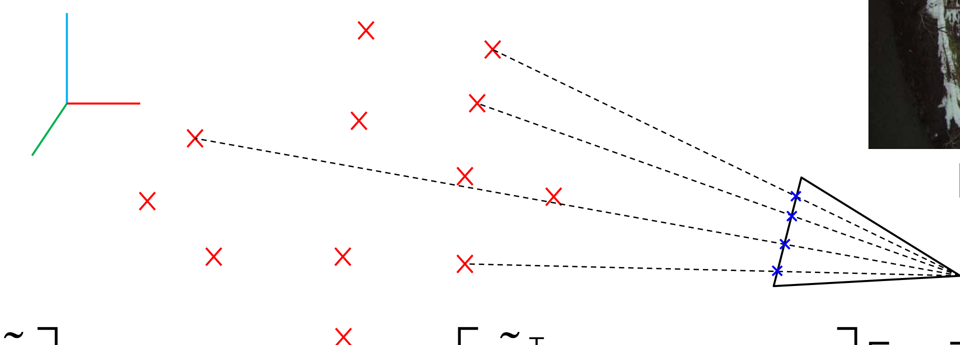
6. New Camera Registration

Perspective-n-point



6. New Camera Registration

Perspective-n-point



$\mathbf{R}_{new}, \mathbf{C}_{new}$

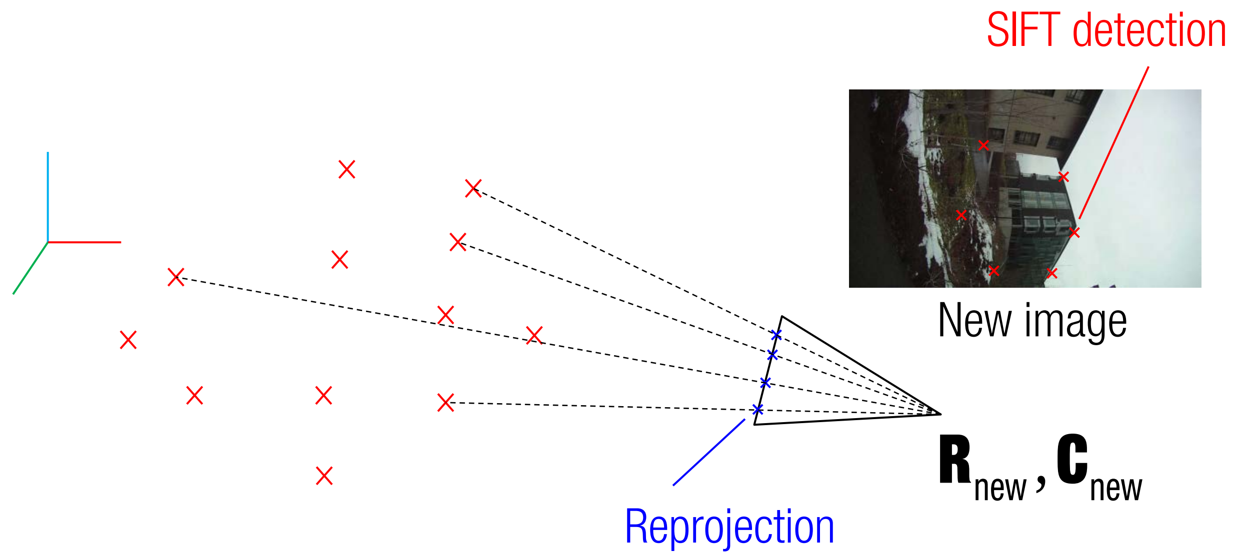
New image

$$\begin{bmatrix}u\\v\\1\end{bmatrix}_\times \begin{bmatrix}\mathbf{P}_1\\\mathbf{P}_2\\\mathbf{P}_3\end{bmatrix}\tilde{\mathbf{X}} = \begin{bmatrix}u\\v\\1\end{bmatrix}_\times \begin{bmatrix}\mathbf{P}_1\tilde{\mathbf{X}}\\\mathbf{P}_2\tilde{\mathbf{X}}\\\mathbf{P}_3\tilde{\mathbf{X}}\end{bmatrix} = \begin{bmatrix}0 & -1 & v\\1 & 0 & -u\\-v & u & 0\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} & \mathbf{0}_{1\times4}\\\mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4}\\\mathbf{0}_{1\times4} & \mathbf{0}_{1\times4} & \tilde{\mathbf{X}}^\mathsf{T}\end{bmatrix}\begin{bmatrix}\mathbf{P}_1^\mathsf{T}\\\mathbf{P}_2^\mathsf{T}\\\mathbf{P}_3^\mathsf{T}\end{bmatrix} =$$

$$\begin{bmatrix}\mathbf{0}_{1\times4} & -\tilde{\mathbf{X}}^\mathsf{T} & v\tilde{\mathbf{X}}^\mathsf{T}\\\tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4} & -u\tilde{\mathbf{X}}^\mathsf{T}\\-v\tilde{\mathbf{X}}^\mathsf{T} & u\tilde{\mathbf{X}}^\mathsf{T} & \mathbf{0}_{1\times4}\end{bmatrix}\begin{bmatrix}\mathbf{P}_1^\mathsf{T}\\\mathbf{P}_2^\mathsf{T}\\\mathbf{P}_3^\mathsf{T}\end{bmatrix} = \mathbf{0}$$

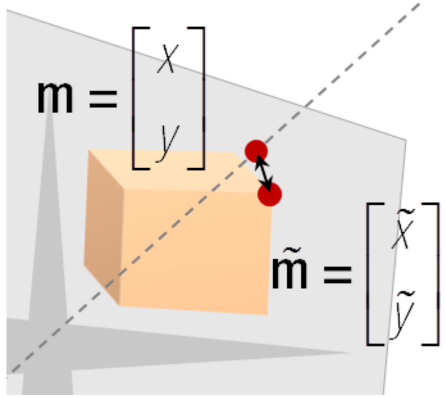
3x12 matrix

7. Bundle Adjustment



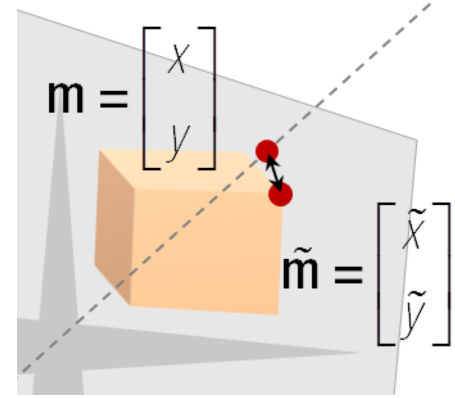
Reprojection error

$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$



Reprojection error

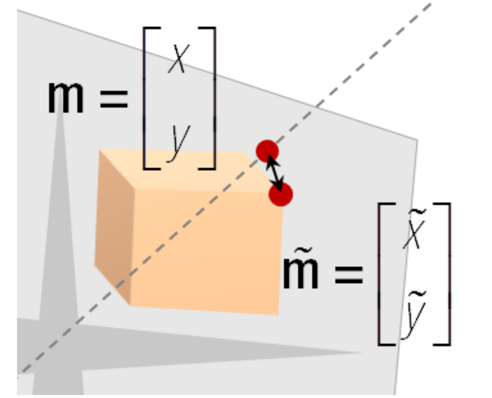
$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$



$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix}$$

Reprojection error



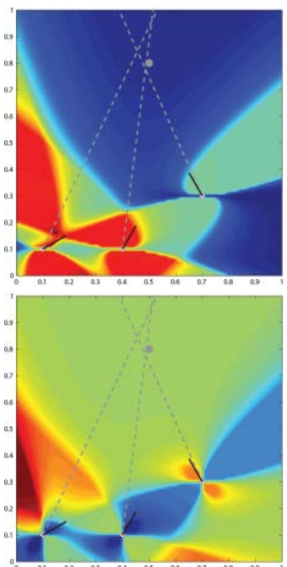
$$e = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

$$\underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \\ v(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) / w(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \end{bmatrix} \right\|^2 = \underset{\mathbf{q},\mathbf{C},\mathbf{X}}{\text{minimize}} \left\| \mathbf{b} - \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) \right\|^2$$

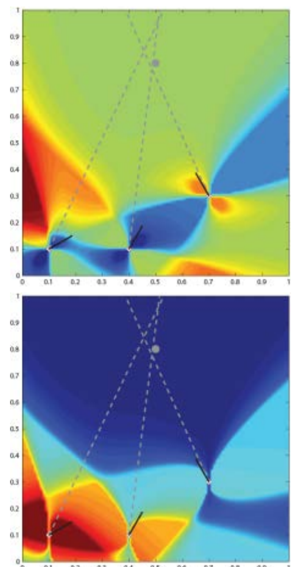
$$\mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X}) = \begin{bmatrix} u/w \\ u/w \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} \underbrace{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{R}}}_{2\text{x}9} \underbrace{\dfrac{\partial \mathbf{R}}{\partial \mathbf{q}}}_{9\text{x}4} & \underbrace{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{C}}}_{2\text{x}3} & \underbrace{\dfrac{\partial \mathbf{f}(\mathbf{R}(\mathbf{q}),\mathbf{C},\mathbf{X})}{\partial \mathbf{X}}}_{2\text{x}3} \end{bmatrix}$$

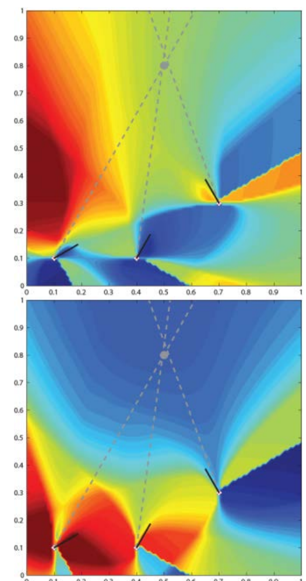
$$\mathbf{J}^{\top}\mathbf{J}\Delta\mathbf{x} = \mathbf{J}^{\top}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$



$\Delta X +$

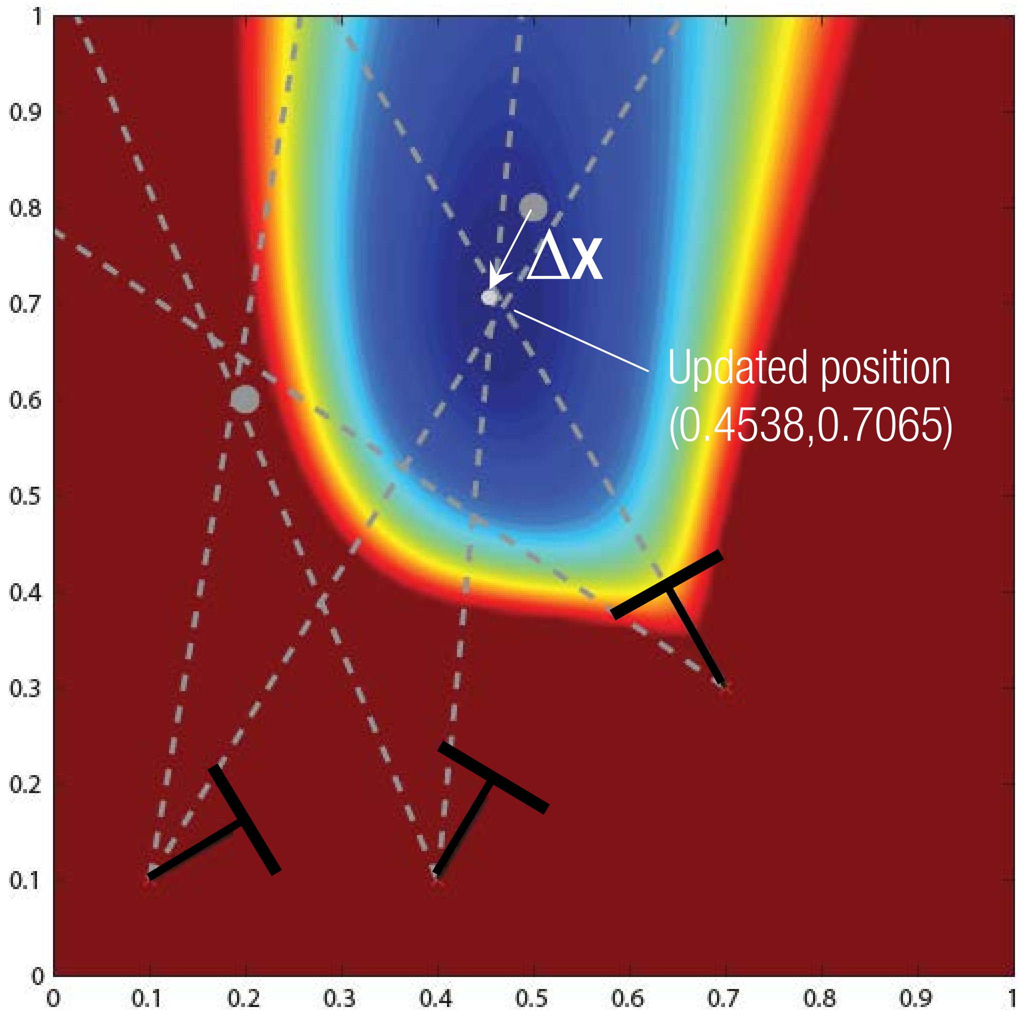


$\Delta Y =$



where $\Delta\mathbf{x} = \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix}$

$$\Delta \mathbf{x} = \left(\mathbf{J}^{\mathrm{T}}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

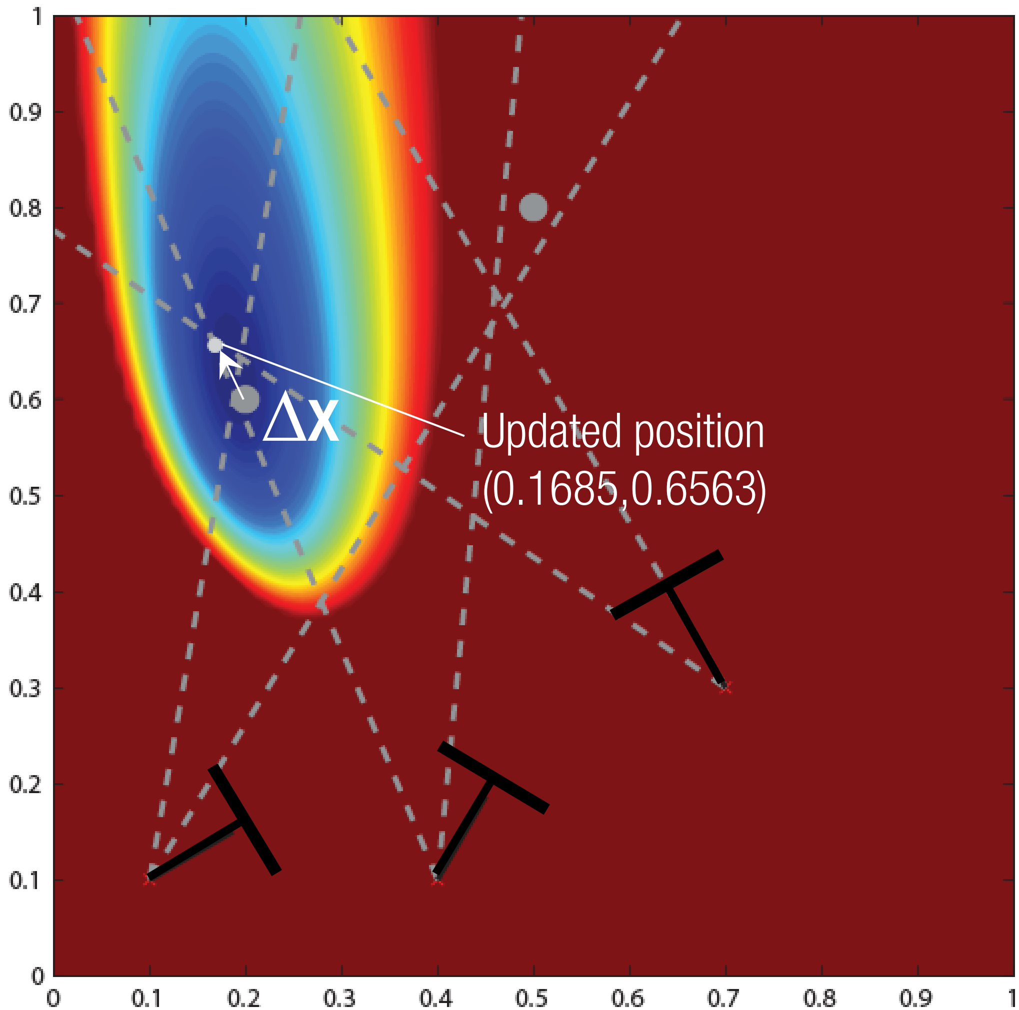


$$\sum_{i=1}^{3}\left\|\tilde{x}_{i1} - f_{i1}(\mathbf{X})\right\|^2$$

$$= \sum_{i=1}^{3}\left\|\tilde{x}_{i1} - u_{i1} / w_{i1}\right\|^2$$

$$\tilde{x}_{\underbrace{i}_{\text{Camera index}}\underbrace{1}_{\text{Point index}}}$$

$$\Delta \mathbf{x} = \left(\mathbf{J}^{\mathrm{T}}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

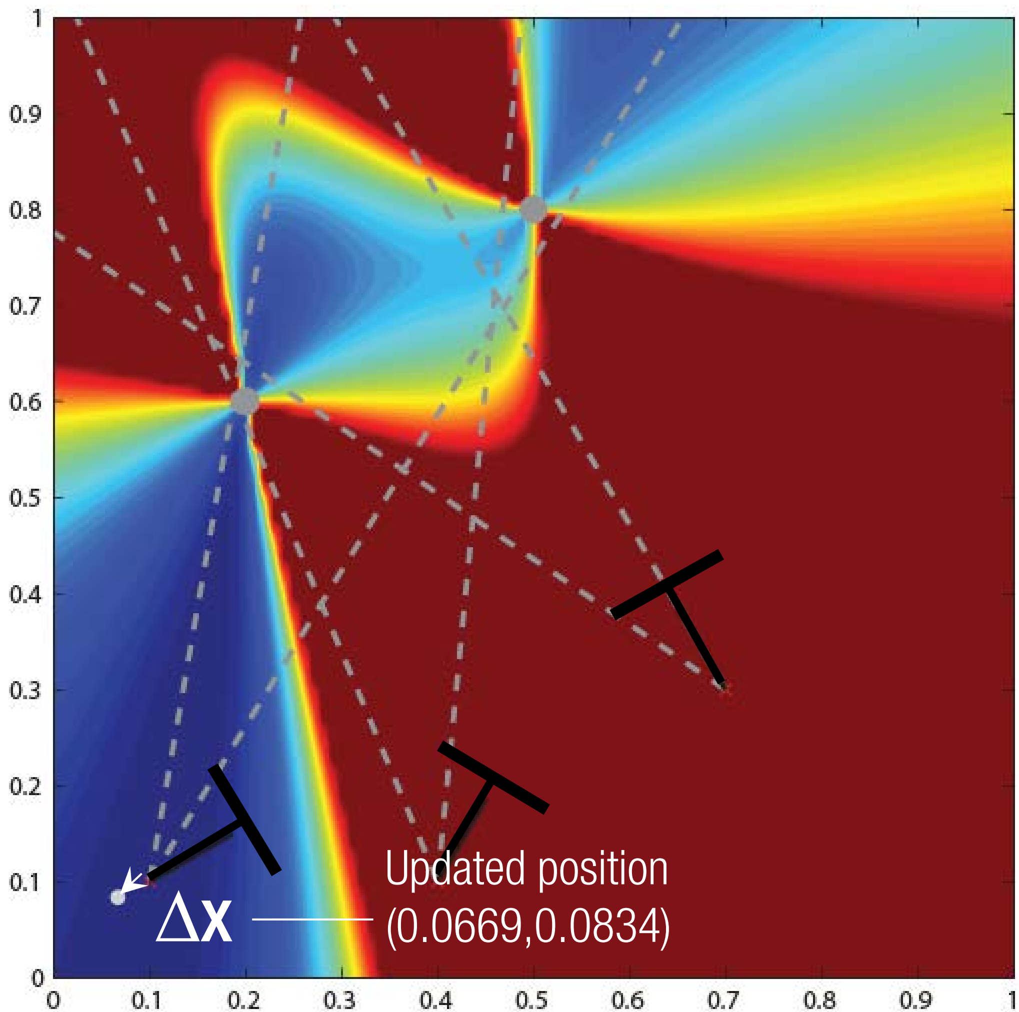


$$\sum_{i=1}^{3}\left\|\tilde{x}_{i2} - f_{i2}(\mathbf{X})\right\|^2$$

$$= \sum_{i=1}^{3}\left\|\tilde{x}_{i2} - u_{i2} / w_{i2}\right\|^2$$

$$\tilde{x}_{\underbrace{i}_{\text{Camera index}}\underbrace{2}_{\text{Point index}}}$$

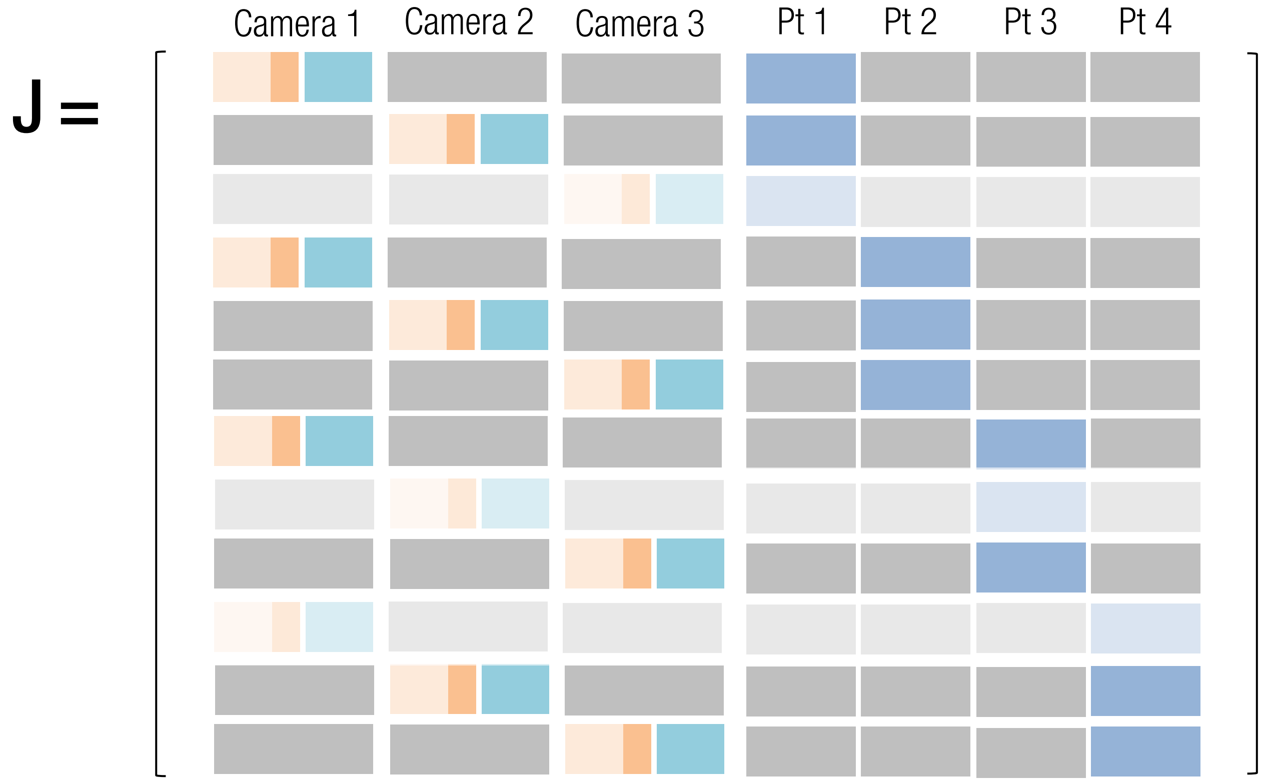
$$\Delta \mathbf{x} = \left(\mathbf{J}^{\mathsf{T}}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathsf{T}}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

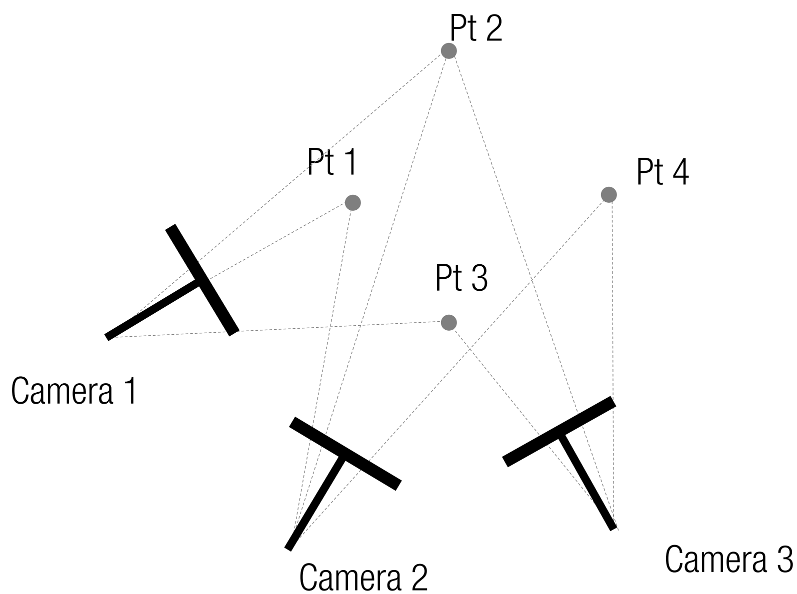


$$\sum_{j=1}^{2}\left\|\tilde{x}_{1j} - f_{1j}(\mathbf{X})\right\|^2$$

$$= \sum_{j=1}^{2}\left\|\tilde{x}_{1j} - u_{1j} / w_{1j}\right\|^2$$

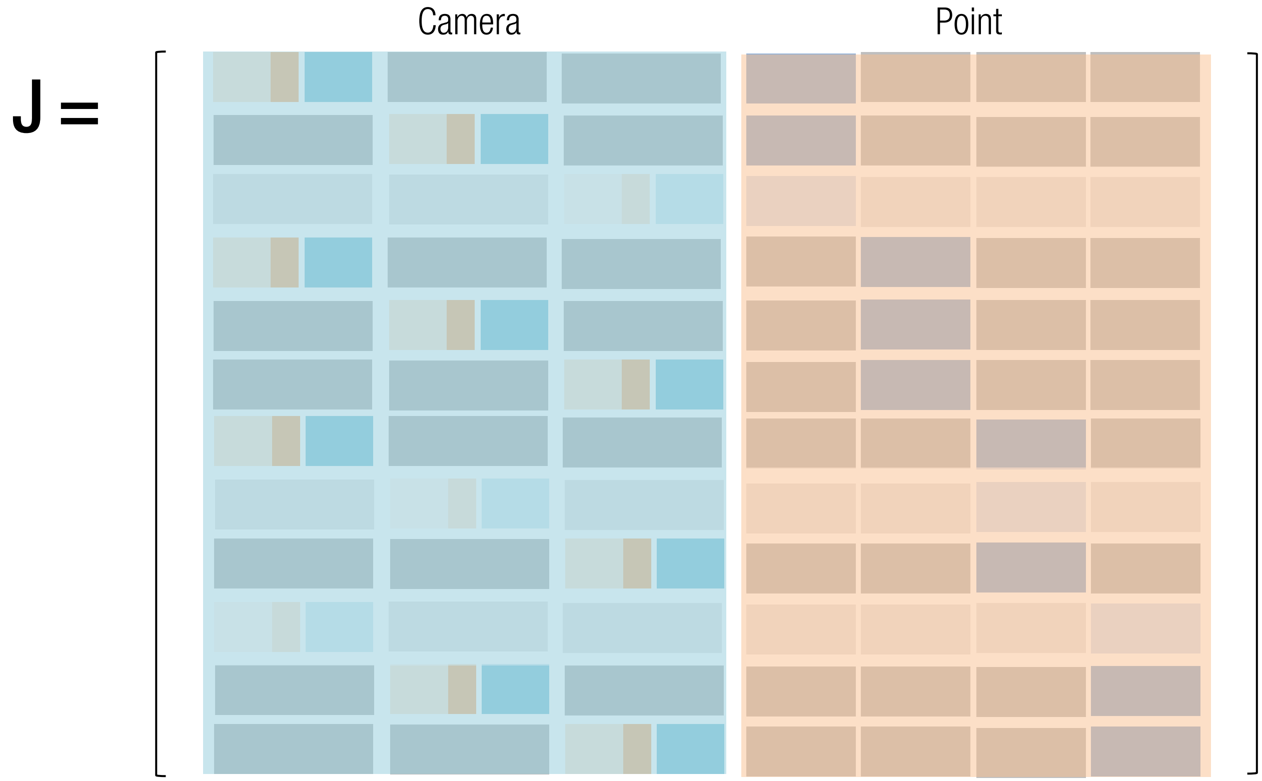
$$\underbrace{\tilde{x}_{\underbrace{1}_{\text{Camera index}}\underbrace{j}_{\text{Point index}}}}$$

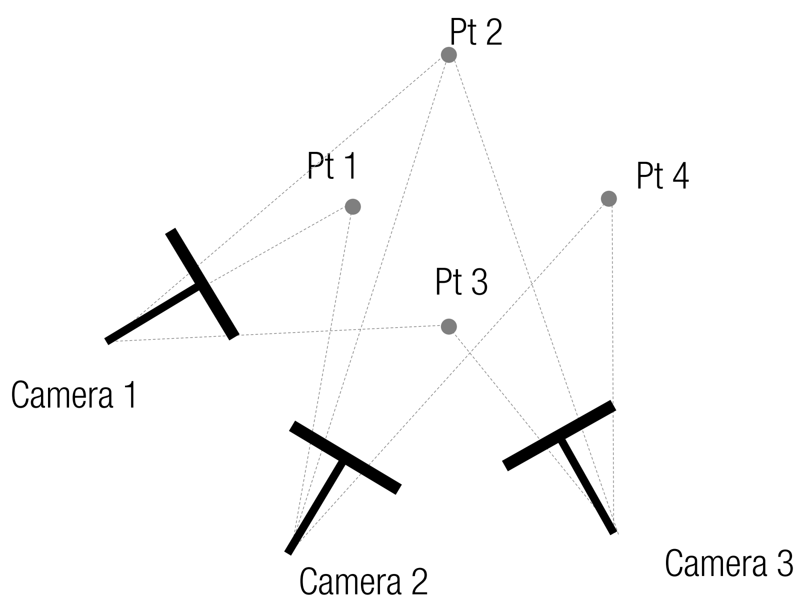




$$\Delta \mathbf{x} = \underline{\left(\mathbf{J}^{T}\mathbf{J}\right)^{-1}}\,\mathbf{J}^{T}\left(\mathbf{b}-\mathbf{f}(\mathbf{x})\right)$$

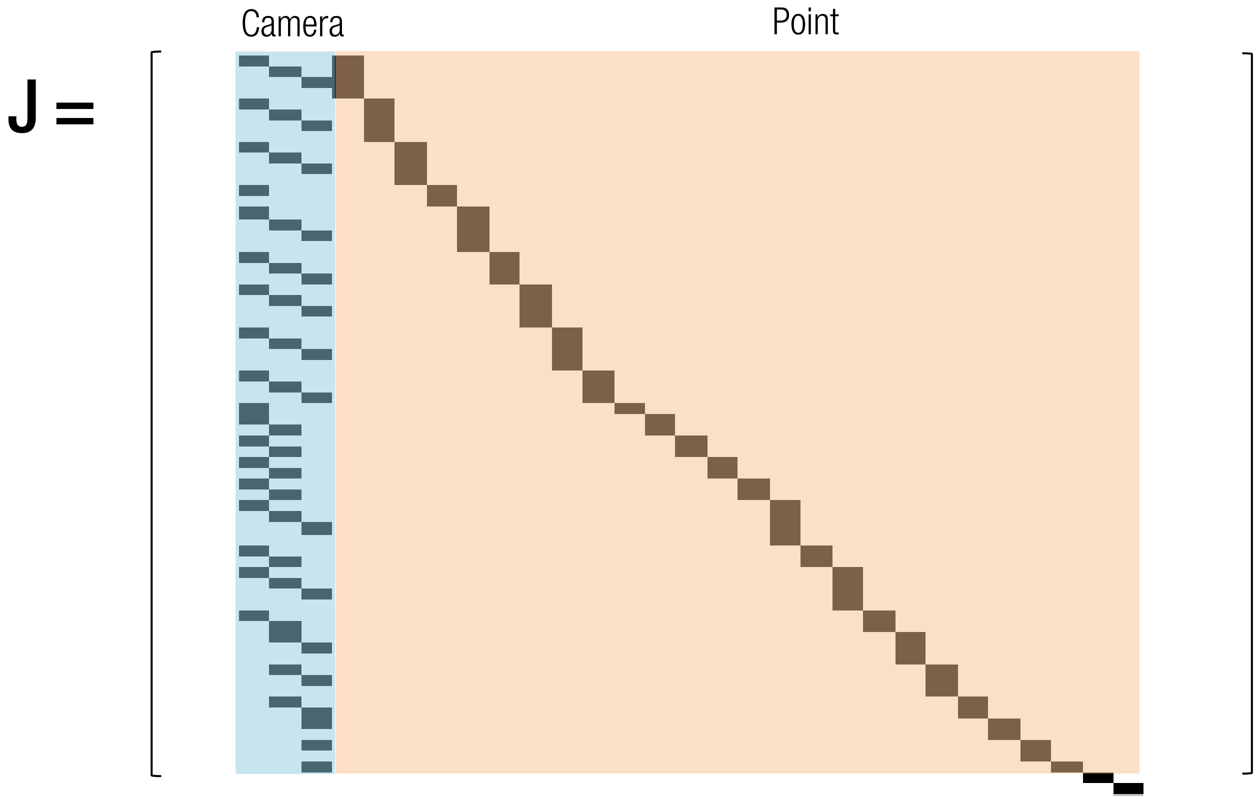
Main computational bottle neck

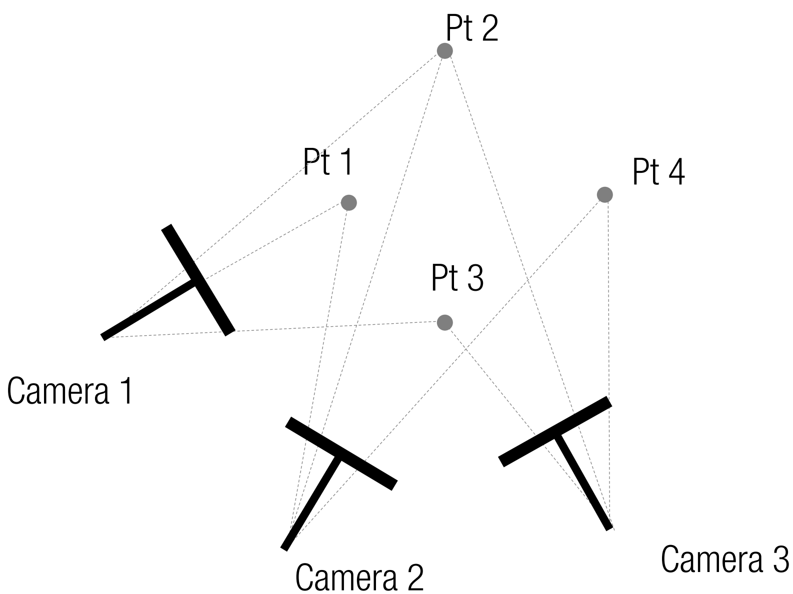




$$\Delta \mathbf{x} = \underline{\left(\mathbf{J}^{T}\mathbf{J}\right)^{-1}}\, \mathbf{J}^{T}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

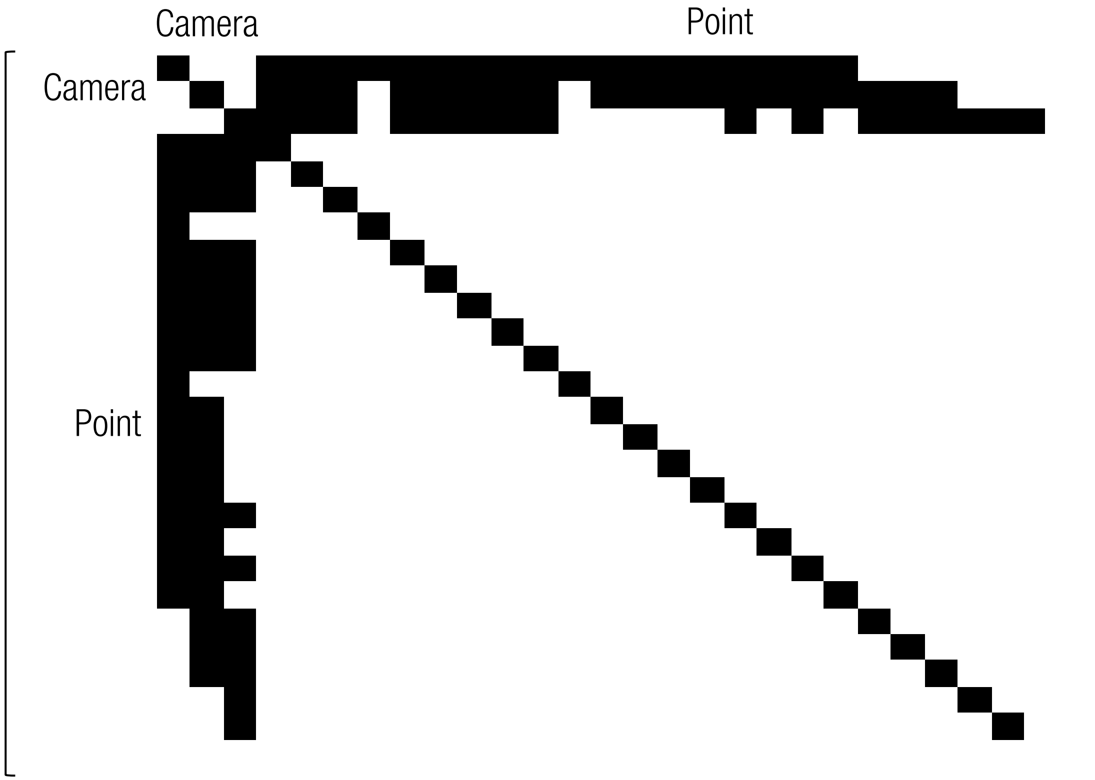
Main computational bottle neck

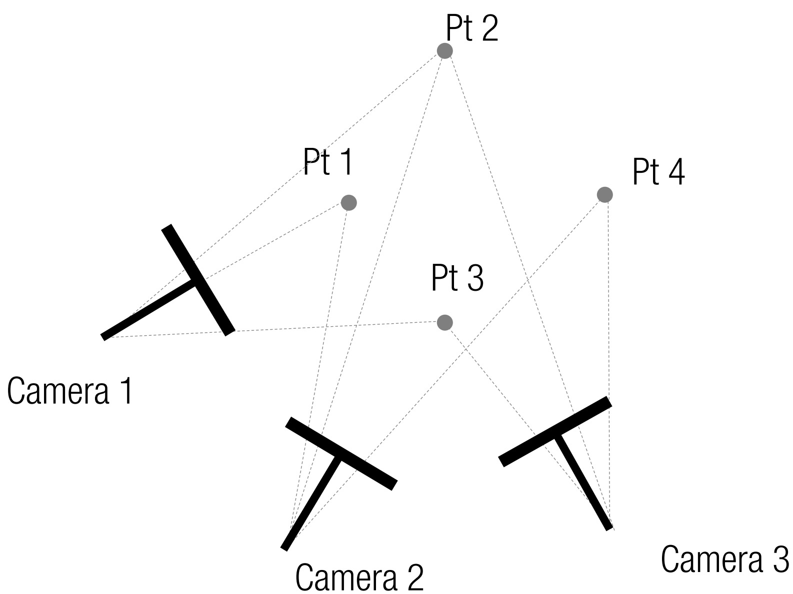




$$\Delta \mathbf{x} = \underbrace{\left(\mathbf{J}^{\mathrm{T}}\mathbf{J}\right)^{-1}}_{\text{Main computational bottle neck}} \mathbf{J}^{\mathrm{T}}\left(\mathbf{b}-\mathbf{f}(\mathbf{x})\right)$$

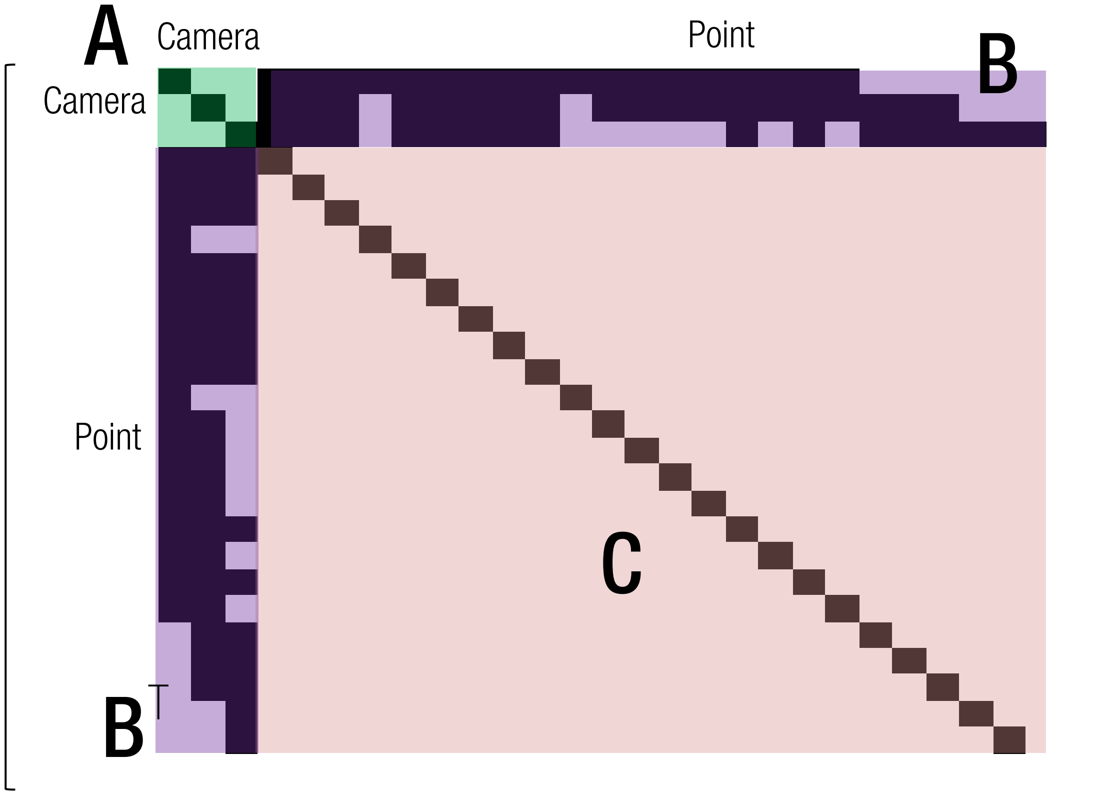
$$J^{T}J =$$





$$\Delta \mathbf{x} = \underbrace{\left(\mathbf{J}^{T}\mathbf{J}\right)^{-1}}_{\text{Main computational bottle neck}} \mathbf{J}^{T}\left(\mathbf{b} - \mathbf{f}(\mathbf{x})\right)$$

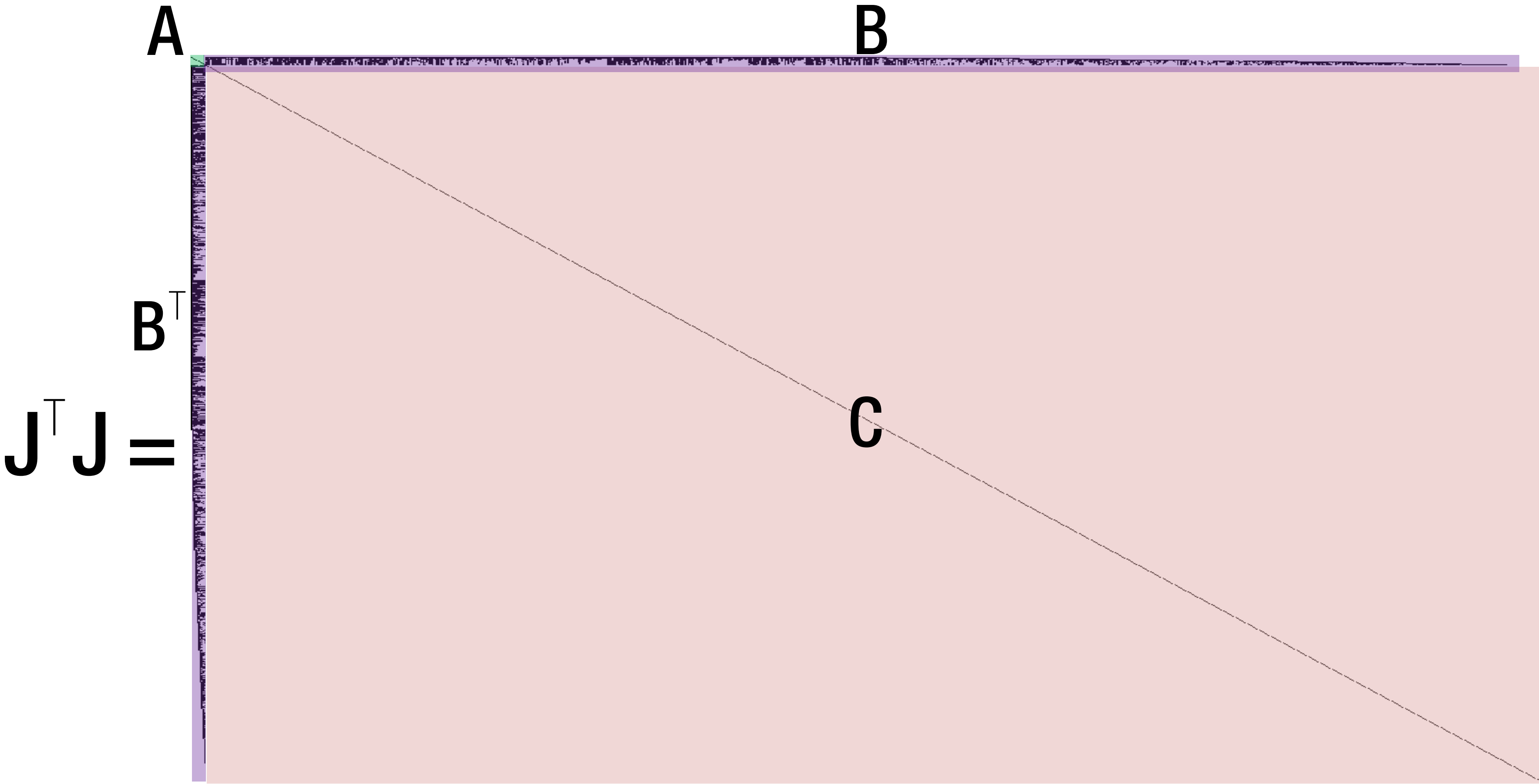
$$\mathbf{J}^T\mathbf{J} =$$



$$\Delta \mathbf{x} = \underbrace{\left(\mathbf{J}^T\mathbf{J}\right)^{-1}}_{\text{Main computational bottle neck}} \mathbf{J}^T\left(\mathbf{b}-\mathbf{f}(\mathbf{x})\right)$$

$$\left(\mathbf{A}-\mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T\right)\Delta\mathbf{x}_c = \mathbf{e}_c - \mathbf{B}\mathbf{C}^{-1}\mathbf{e}_p$$: Reduced system

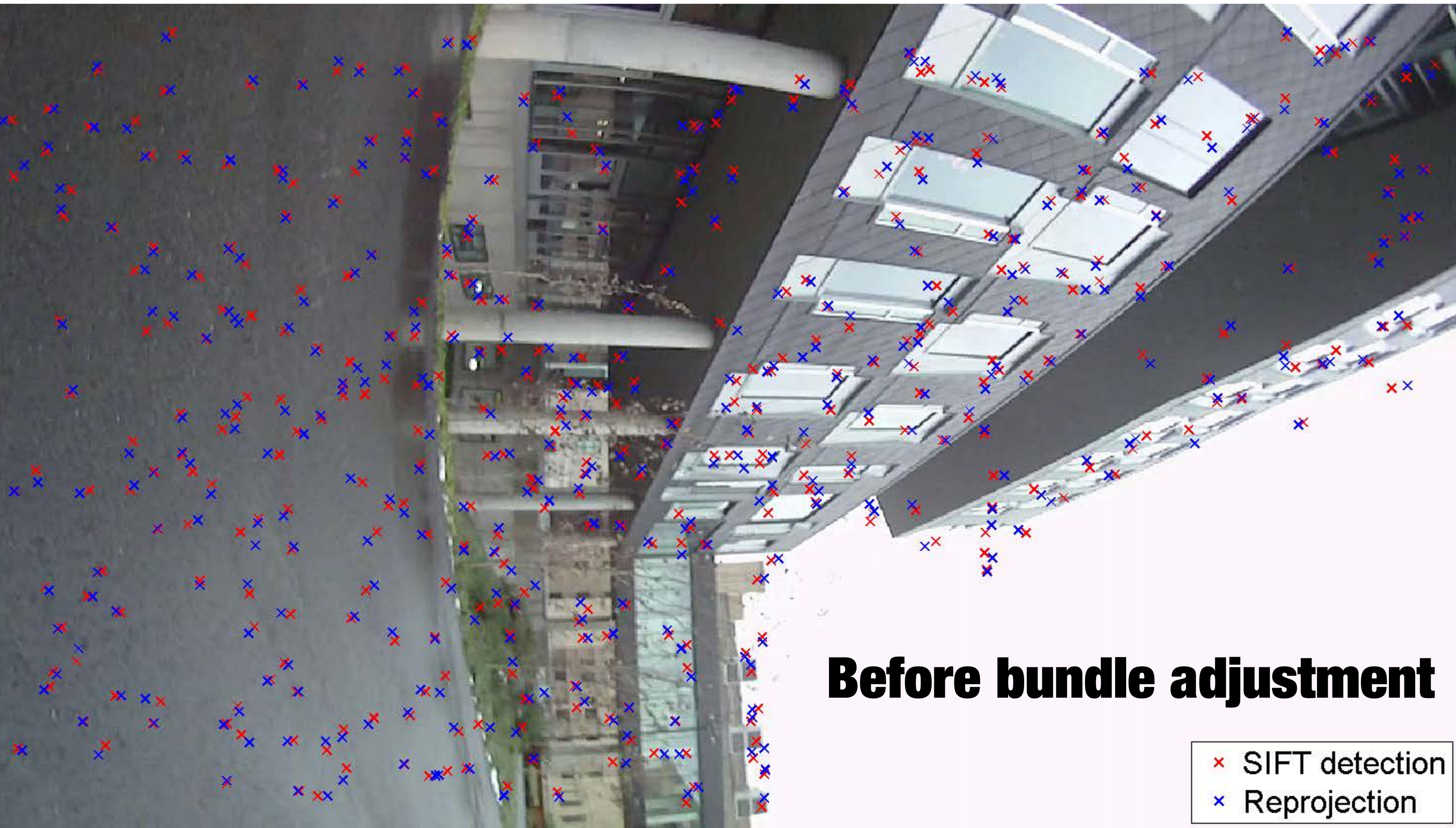
$$\mathbf{C}\Delta\mathbf{x}_p = \mathbf{e}_p - \mathbf{B}^T\Delta\mathbf{x}_c$$: Back substitution



$$\left(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^{T}\right)\Delta\mathbf{x}_{c} = \mathbf{e}_{c} - \mathbf{B}\mathbf{C}^{-1}\mathbf{e}_{p}$$: Reduced system

$$\mathbf{C}\Delta\mathbf{x}_{p} = \mathbf{e}_{p} - \mathbf{B}^{T}\Delta\mathbf{x}_{c}$$: Back substitution

7. Bundle Adjustment



7. Bundle Adjustment