Two View (Epipolar) Geometry

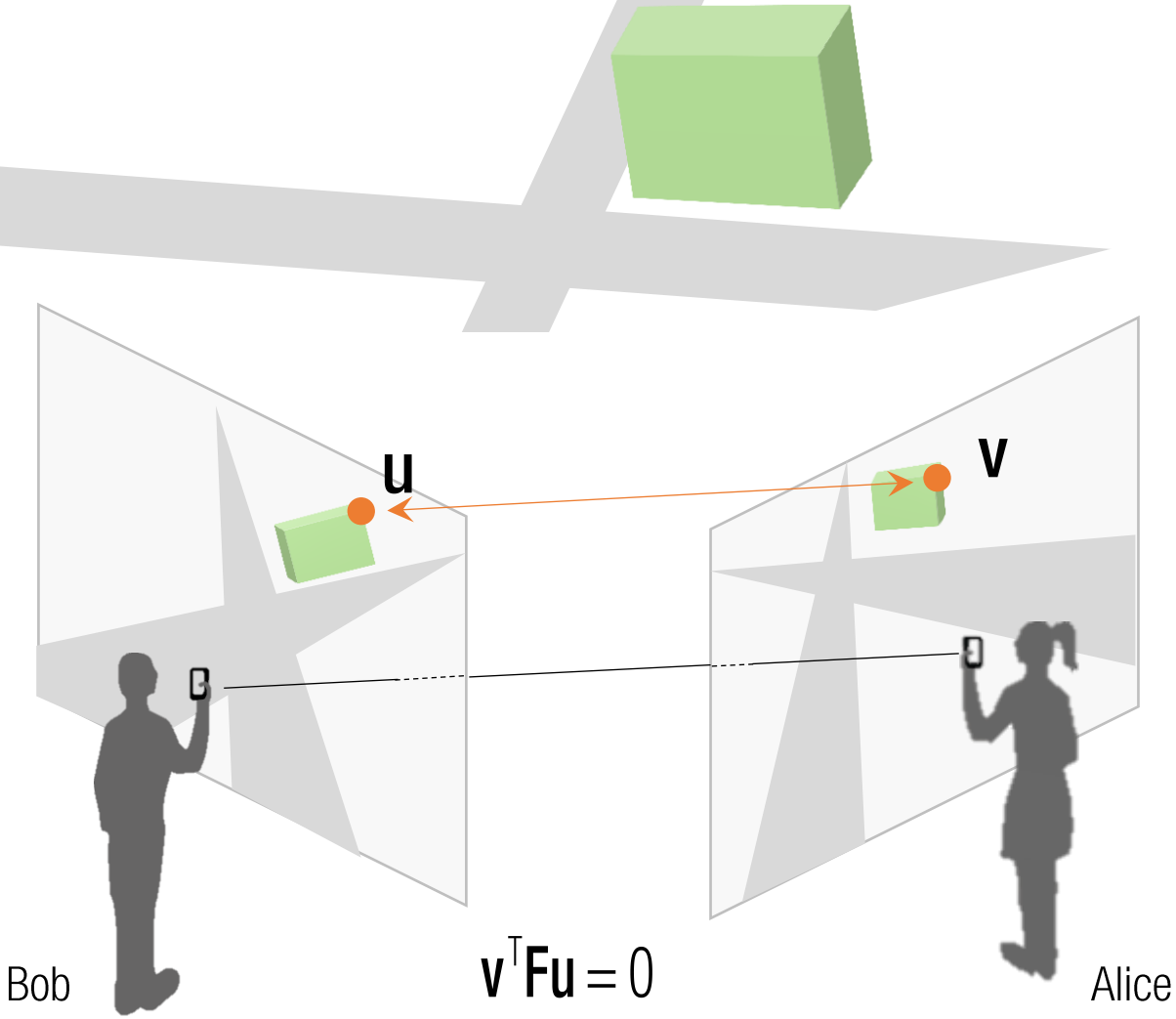
Announcement

- HW #4 is out (start early)

- HW #5 grading is done


Daniel Wedge “The Fundamental Matrix Song”

Fundamental Matrix Estimation

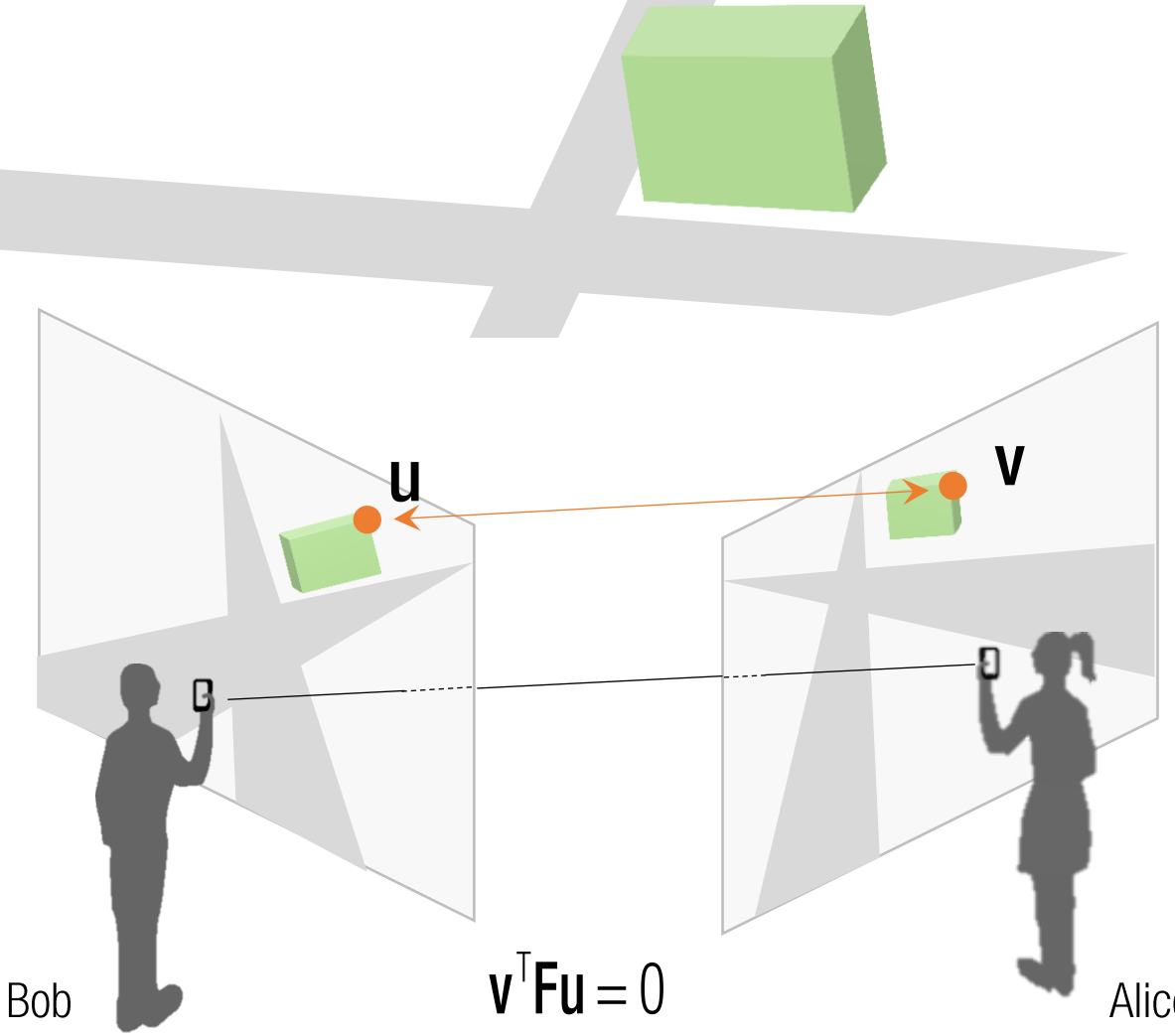


$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

Degree of freedom of fundamental matrix:
7 = 9 (3x3 matrix) – 1(scale) – 1 (rank 2)

We will estimate fundamental matrix with 8 parameter by ignoring rank constraint and then project onto rank 2 matrix:

Fundamental Matrix Estimation

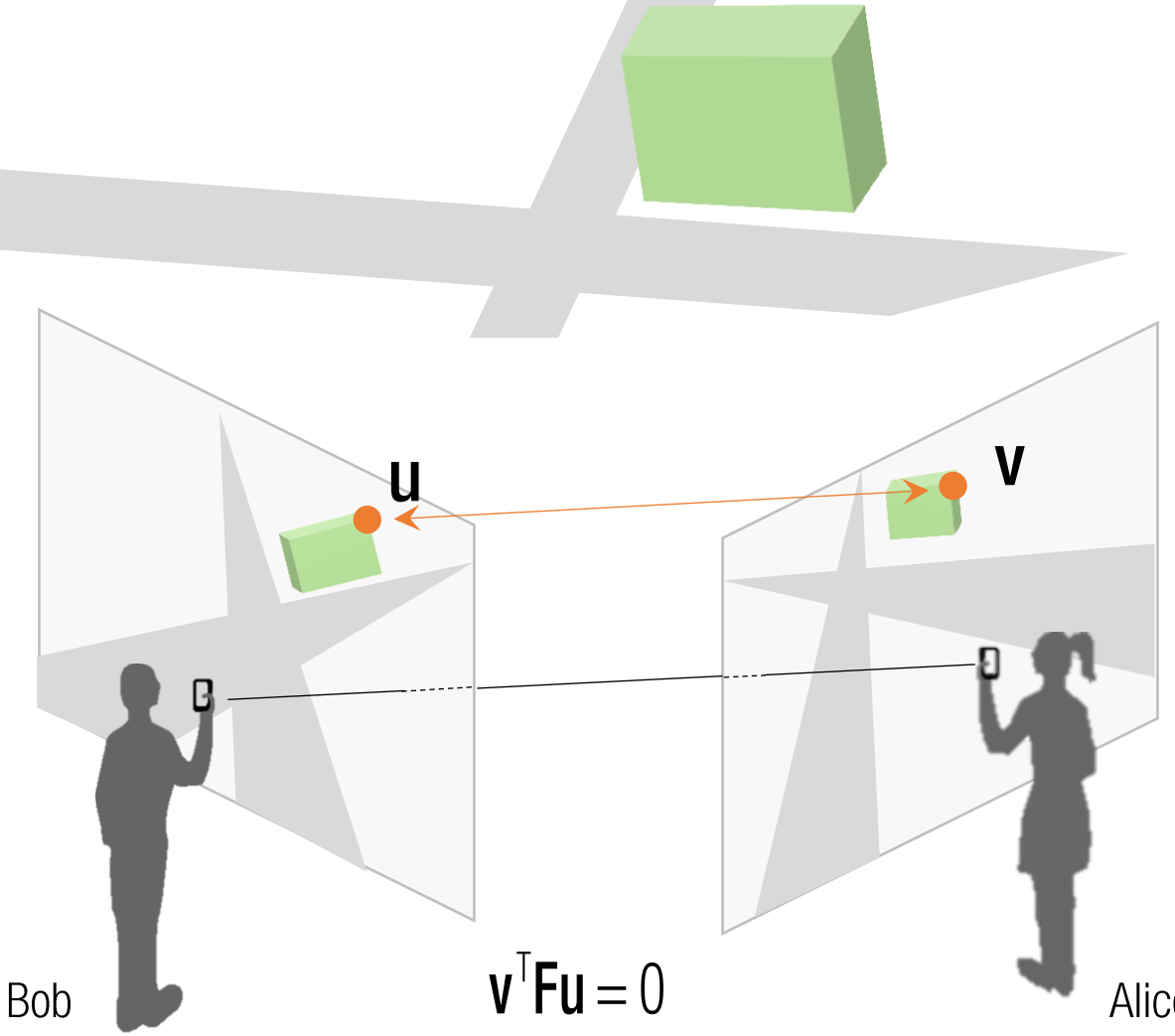


$$\mathbf{v}^T\mathbf{F}\mathbf{u} = \begin{bmatrix} v^x & v^y & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u^x \\ u^y \\ 1 \end{bmatrix}$$

$$= f_{11}u^xv^x + f_{12}u^yv^x + f_{13}v^x + f_{21}u^xv^y + f_{22}u^yv^y + f_{23}v^y + f_{31}u^x + f_{32}u^y + f_{33}$$

$$= 0 \qquad \text{Linear in } \mathbf{F}.$$

Fundamental Matrix Estimation



$$\mathbf{v}^T\mathbf{F}\mathbf{u} = \begin{bmatrix} v^x & v^y & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u^x \\ u^y \\ 1 \end{bmatrix}$$

$$= f_{11}u^xv^x + f_{12}u^yv^x + f_{13}v^x + f_{21}u^xv^y + f_{22}u^yv^y + f_{23}v^y + f_{31}u^x + f_{32}u^y + f_{33}$$

$$= 0$$

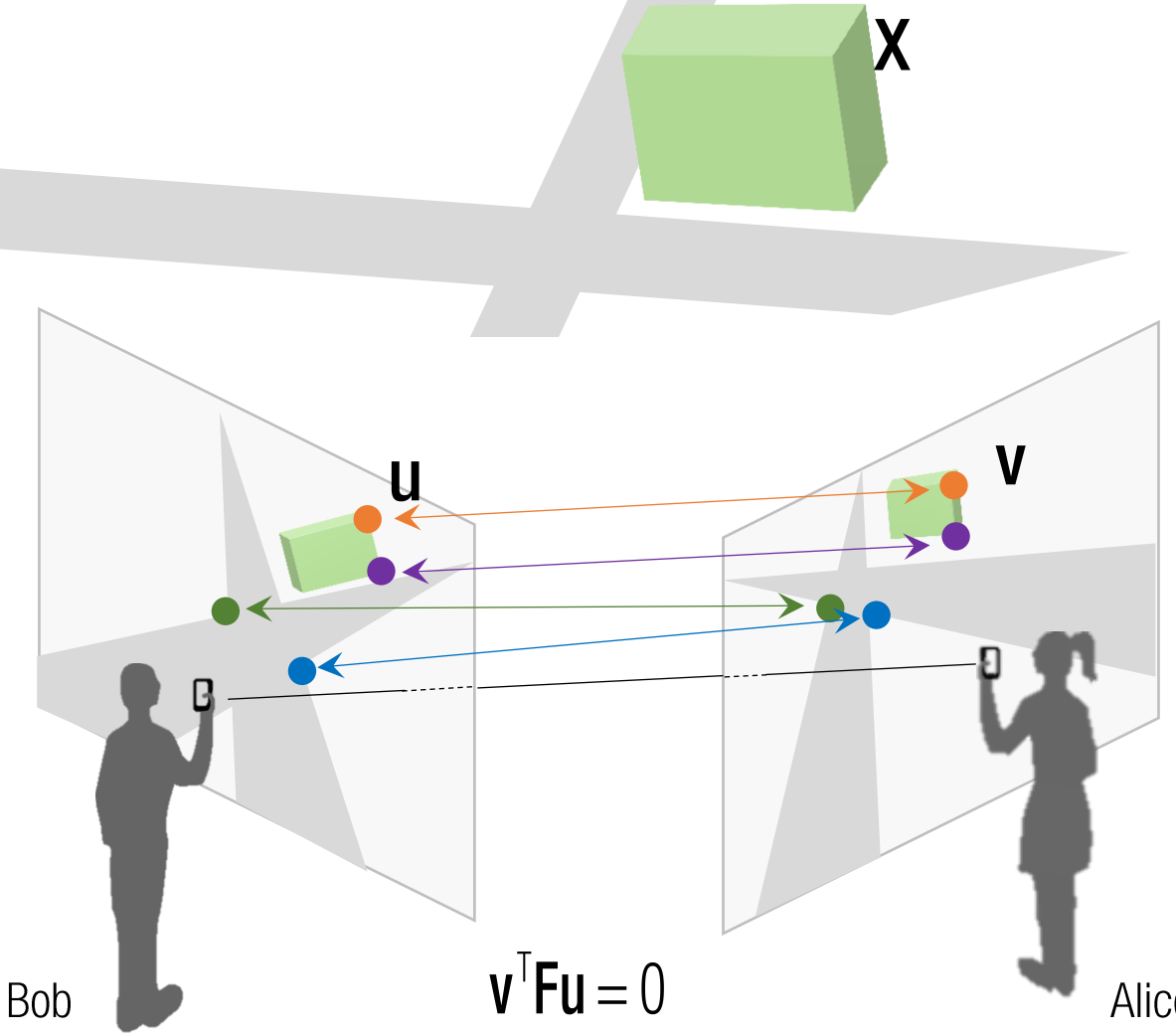
Linear in **F**.

$$\longrightarrow \begin{bmatrix} u^xv^x & u^yv^x & v^x & u^xv^y & u^yv^y & v^y & u^x & u^y & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

$\mathbf{v}^T\mathbf{F}\mathbf{u} = 0$

# of unknowns: 9
# of equations per correspondence: 1

Fundamental Matrix Estimation



$$\mathbf{v}^T\mathbf{F}\mathbf{u} = \begin{bmatrix} v^x & v^y & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u^x \\ u^y \\ 1 \end{bmatrix}$$

$$= f_{11}u^xv^x + f_{12}u^yv^x + f_{13}v^x + f_{21}u^xv^y + f_{22}u^yv^y + f_{23}v^y + f_{31}u^x + f_{32}u^y + f_{33}$$

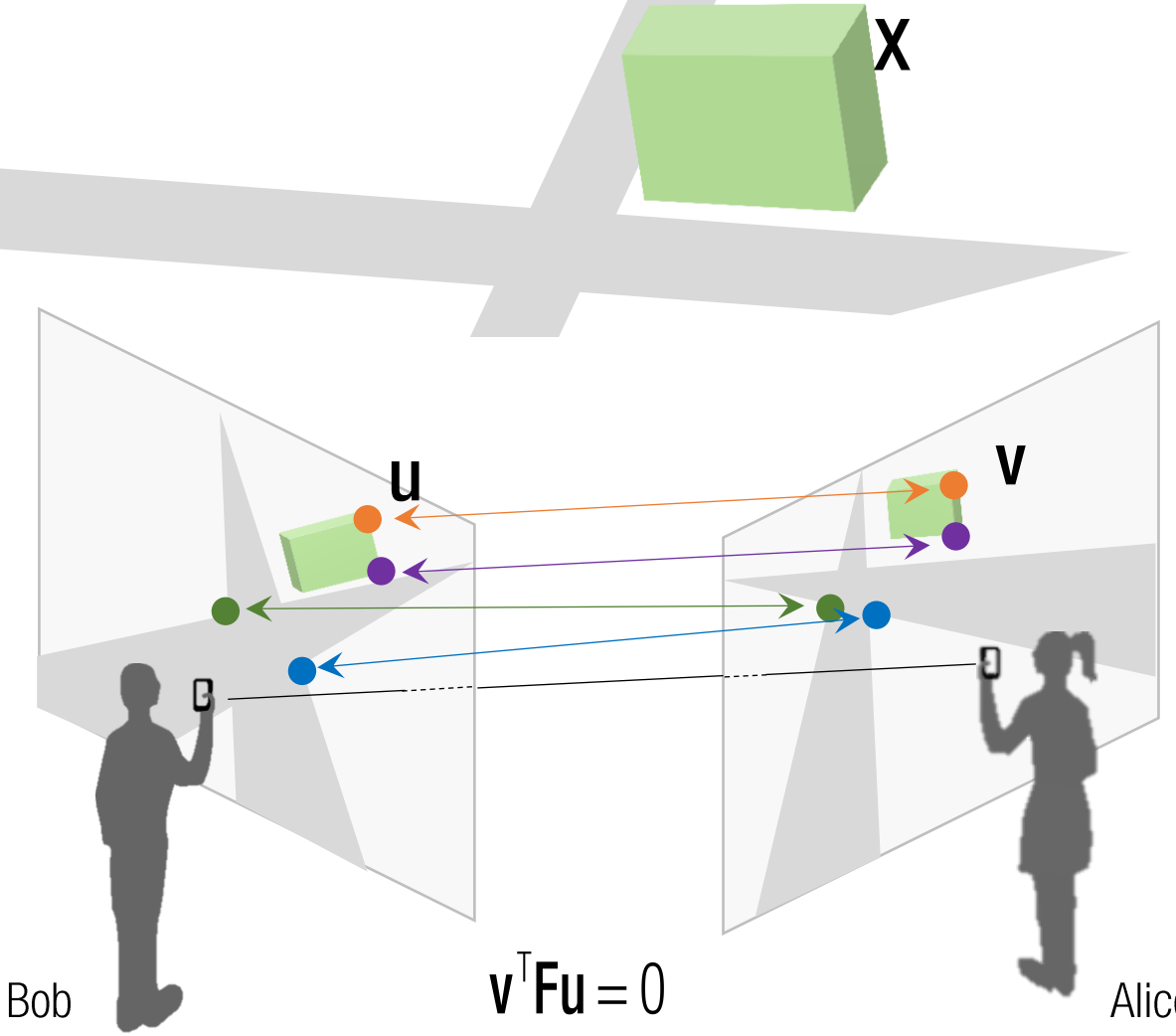
$$= 0$$

Linear in **F**.

$$\rightarrow \begin{bmatrix} u_1^xv_1^x & u_1^yv_1^x & v_1^x & u_1^xv_1^y & u_1^yv_1^y & v_1^y & u_1^x & u_1^y & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_m^xv_m^x & u_m^yv_m^x & v_m^x & u_m^xv_m^y & u_m^yv_m^y & v_m^y & u_m^x & u_m^y & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}_{m\times 1}$$

What is minimum m?

Fundamental Matrix Estimation



$$\mathbf{v}^T\mathbf{F}\mathbf{u} = \begin{bmatrix} v^x & v^y & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u^x \\ u^y \\ 1 \end{bmatrix}$$

$$= f_{11}u^xv^x + f_{12}u^yv^x + f_{13}v^x + f_{21}u^xv^y + f_{22}u^yv^y + f_{23}v^y + f_{31}u^x + f_{32}u^y + f_{33}$$

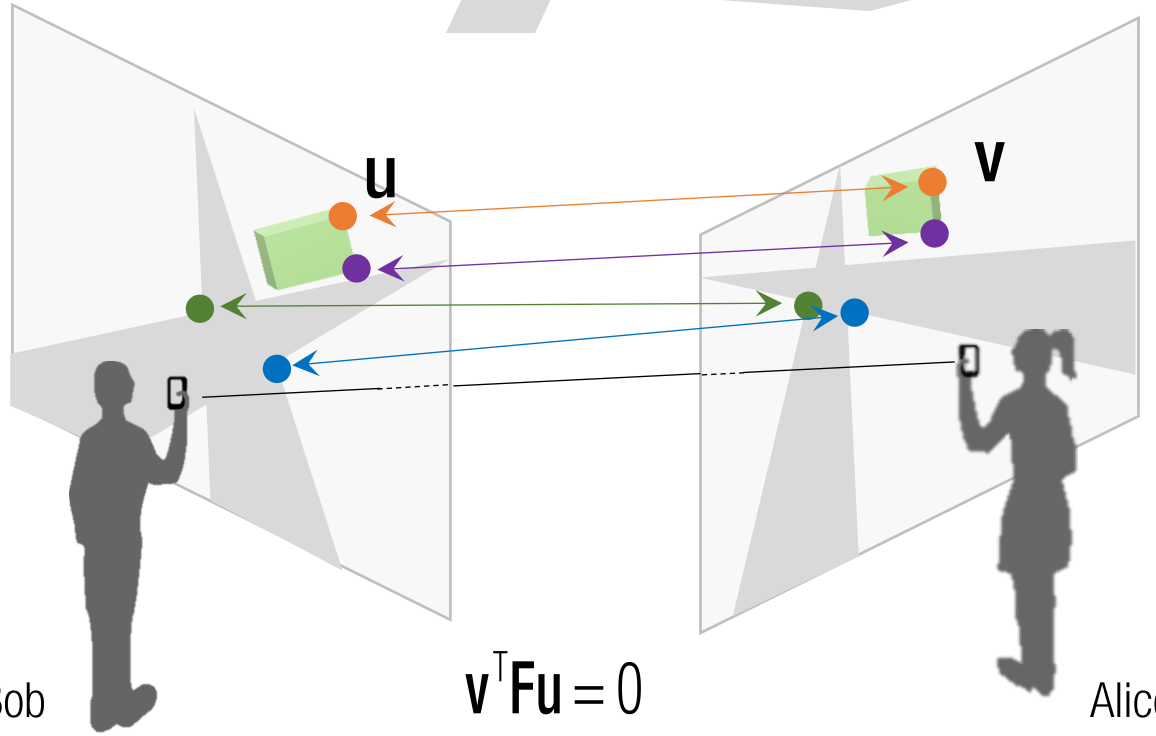
$$= 0$$

Linear in **F**.

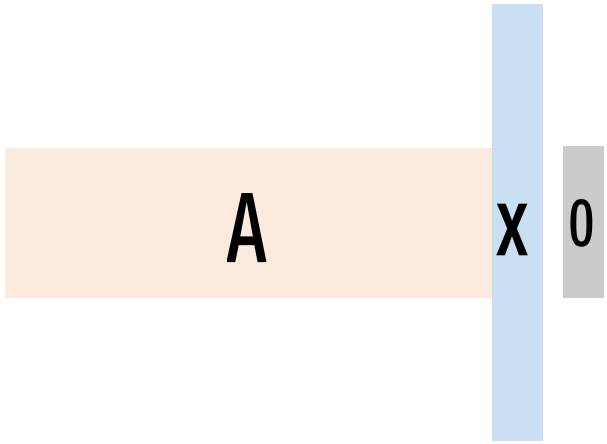
$$\rightarrow \begin{bmatrix} u_1^xv_1^x & u_1^yv_1^x & v_1^x & u_1^xv_1^y & u_1^yv_1^y & v_1^y & u_1^x & u_1^y & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_m^xv_m^x & u_m^yv_m^x & v_m^x & u_m^xv_m^y & u_m^yv_m^y & v_m^y & u_m^x & u_m^y & 1 \end{bmatrix} \mathbf{A} \quad \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ \vdots \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} \mathbf{x} = \mathbf{0}$$

What is minimum m?

Fundamental Matrix Estimation

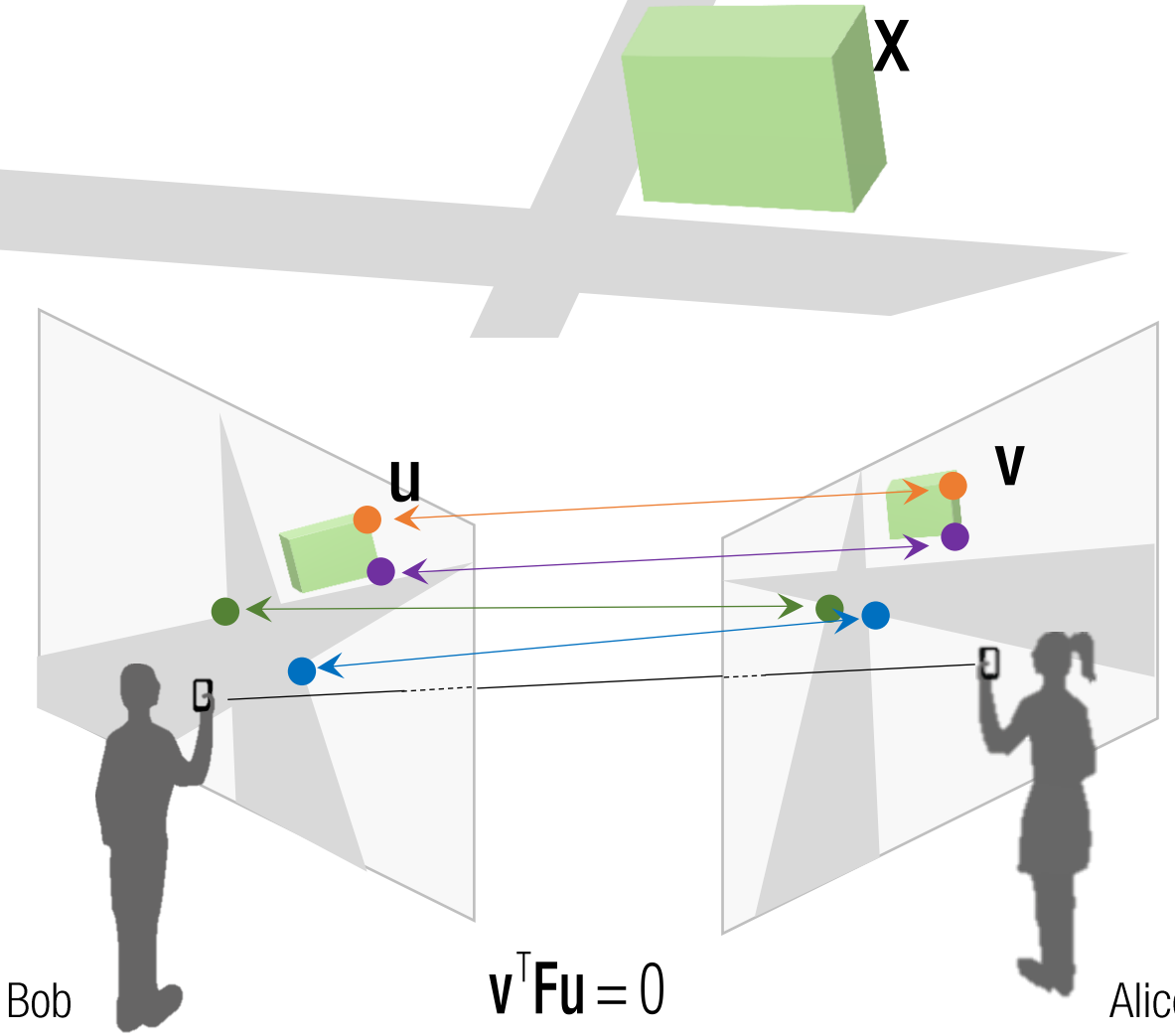


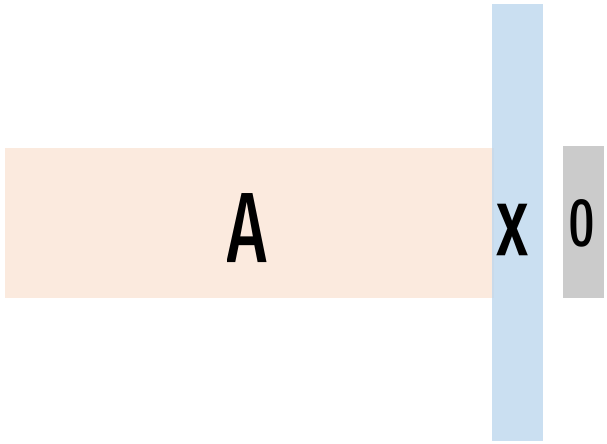
$v^T F u = 0$



The solution is not necessarily satisfy rank 2 constraint.

Fundamental Matrix Estimation

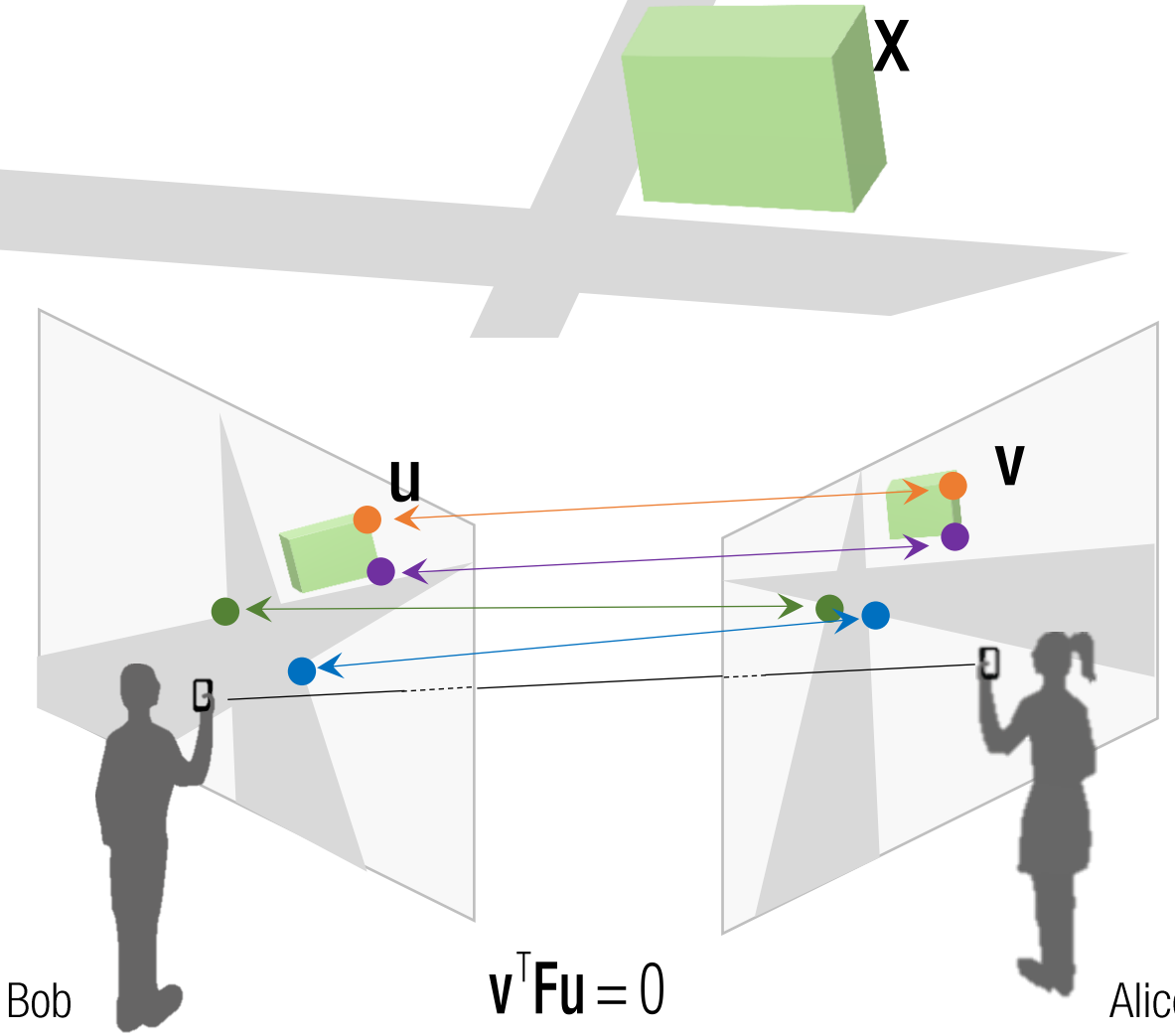


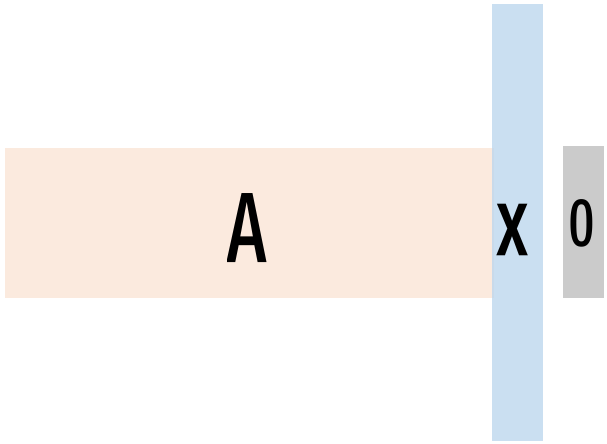


The solution is not necessarily satisfy rank 2 constraint.

$$\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} = U \; D \; V^T$$

Fundamental Matrix Estimation





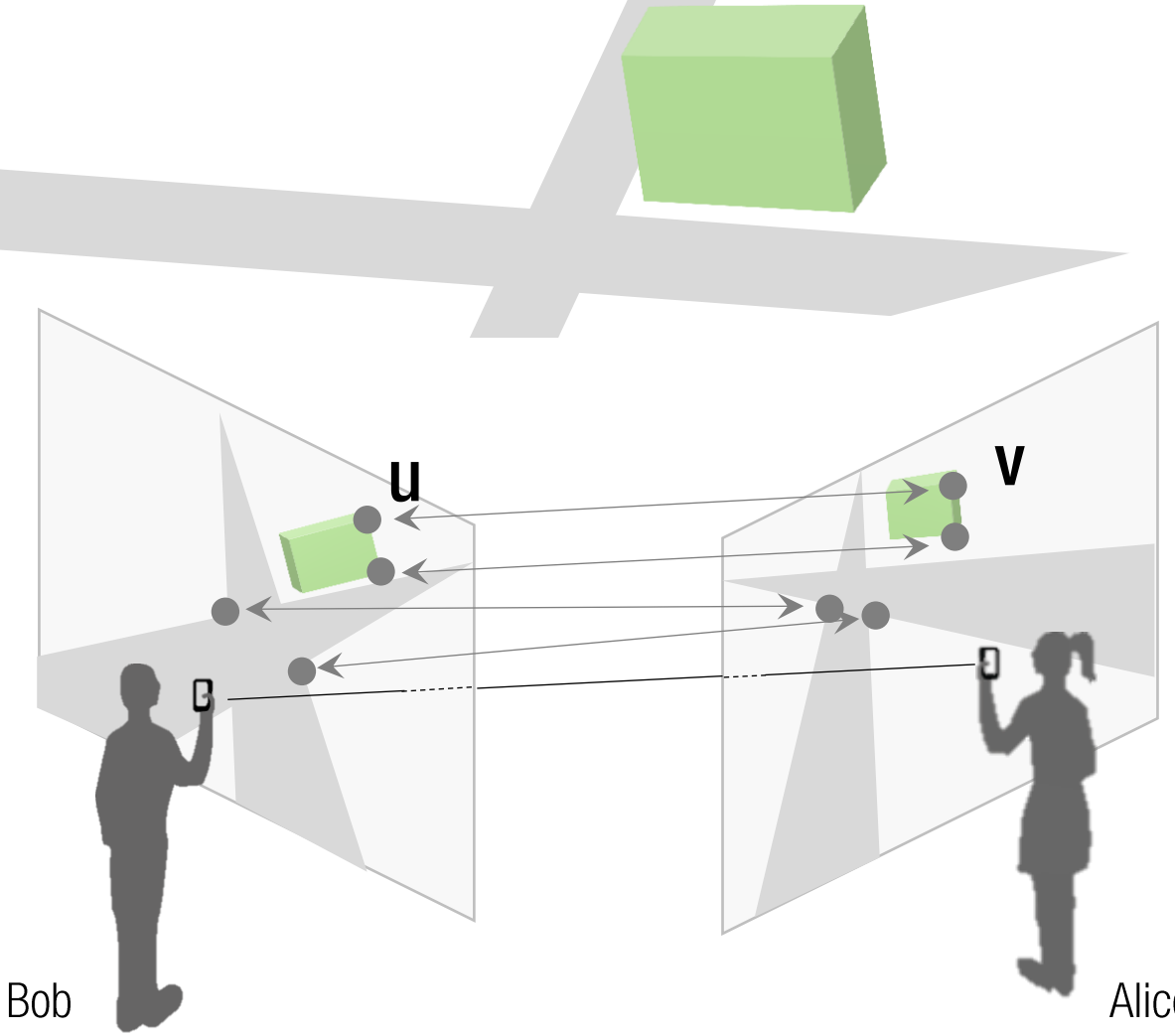
The solution is not necessarily satisfy rank 2 constraint.

$$\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} = U D V^T$$

$$\approx F_{\text{rank}2} = U \tilde{D} V^T$$

SVD cleanup

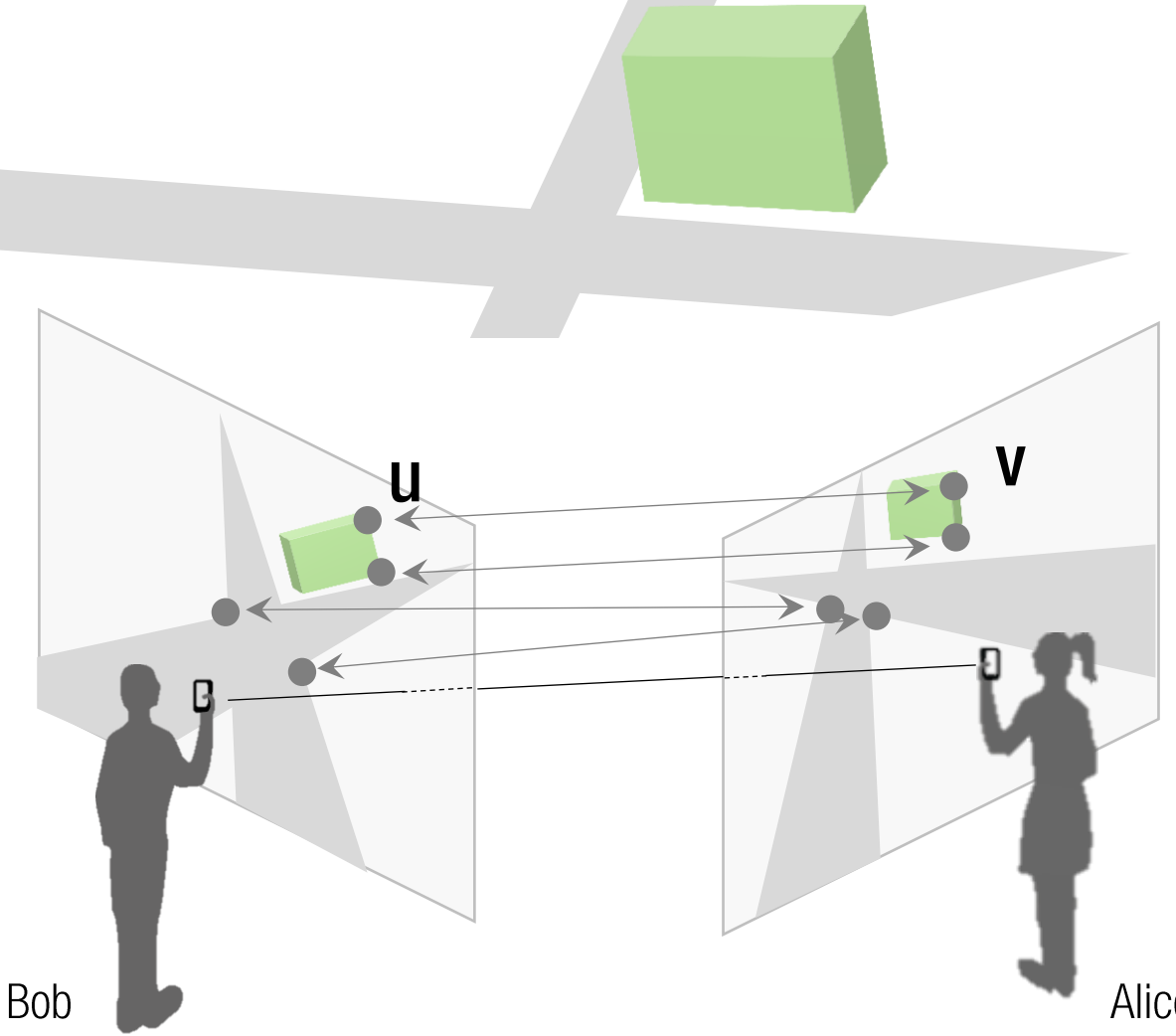
Essential Matrix



$$F = F(R,t)$$

$$= K^{-T}[t]_{\times} RK^{-1} = K^{-T}EK^{-1}$$

Essential Matrix



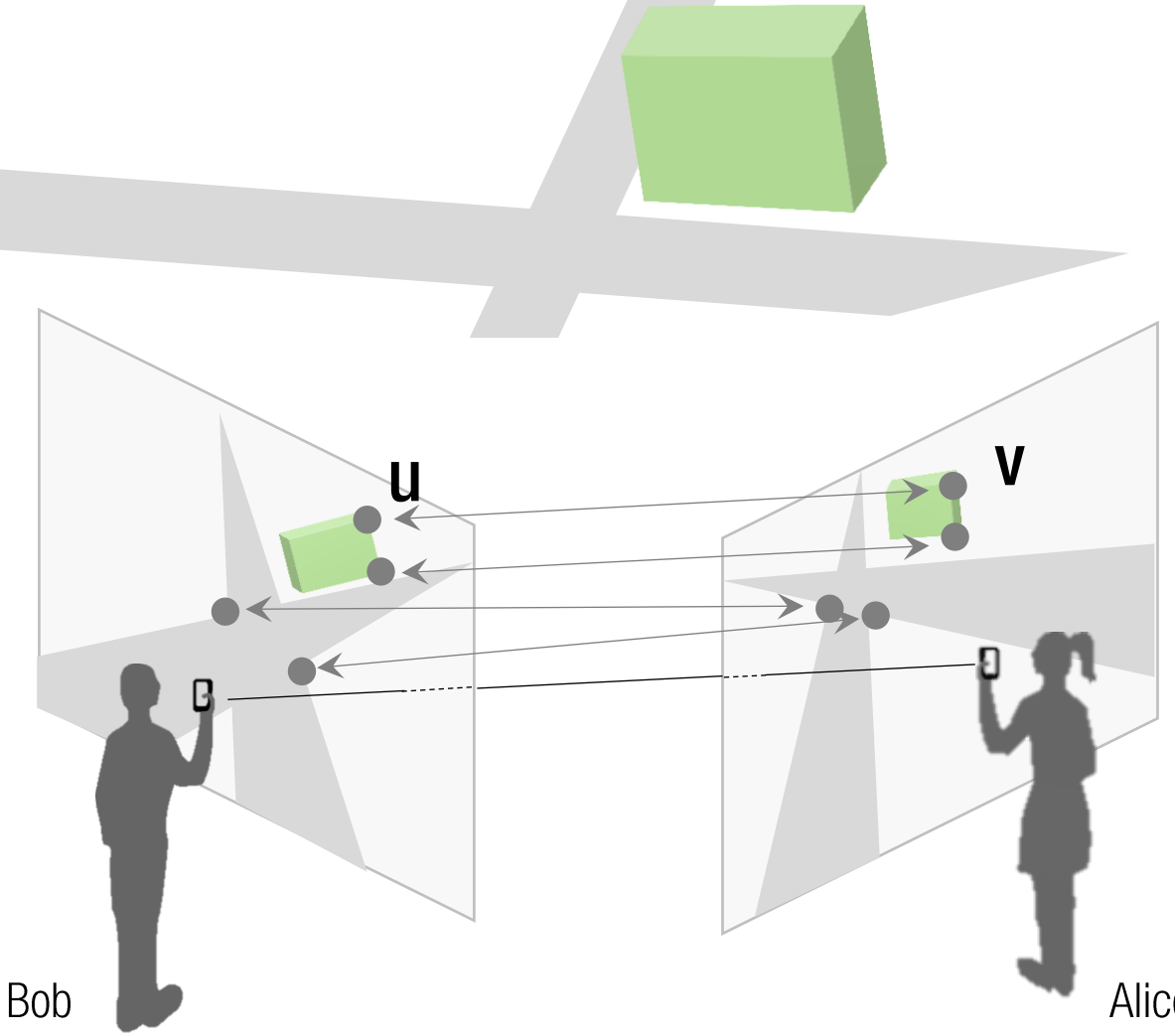
Essential Matrix:

$$\mathbf{F} = \mathbf{F}(\mathbf{R}, \mathbf{t})$$

$$= \mathbf{K}^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1}$$

$$\rightarrow \mathbf{E} = \underbrace{\mathbf{K}^{T} \mathbf{F} \mathbf{K}}_{\text{Calibrated fundamental matrix}} \quad \text{where} \quad \mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$$

Essential Matrix



Essential Matrix:

$$\mathbf{F} = \mathbf{F}(\mathbf{R}, \mathbf{t})$$

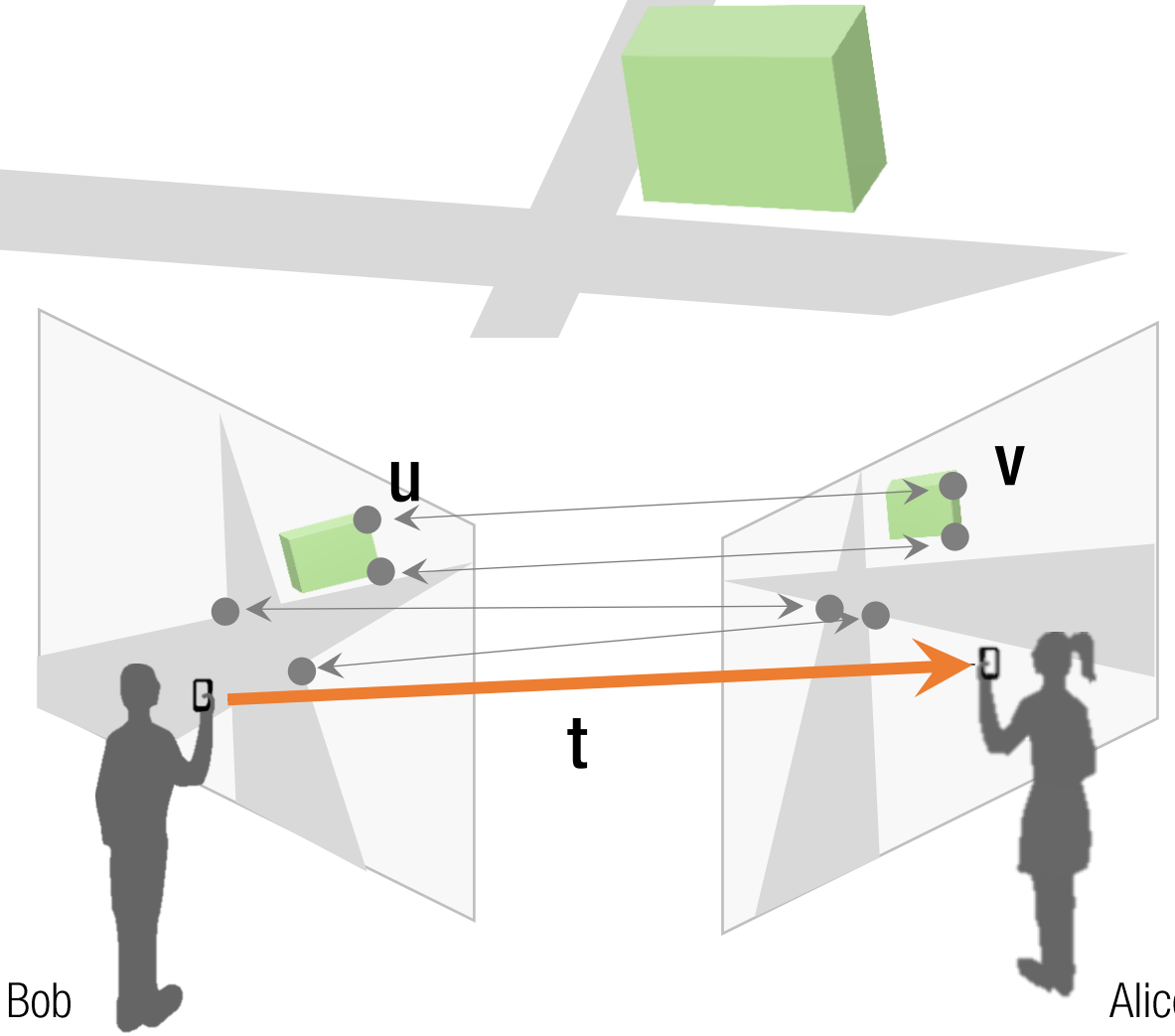
$$= \mathbf{K}^{-T}[\mathbf{t}]_{\times}\mathbf{R}\mathbf{K}^{-1} = \mathbf{K}^{-T}\mathbf{E}\mathbf{K}^{-1}$$

$$\rightarrow \mathbf{E} = \underbrace{\mathbf{K}^{T}\mathbf{F}\mathbf{K}}_{\text{Calibrated fundamental matrix}} \quad \text{where} \quad \mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R}$$

Property of essential matrix:

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

Camera Pose from Essential Matrix (Translation)



Essential Matrix:

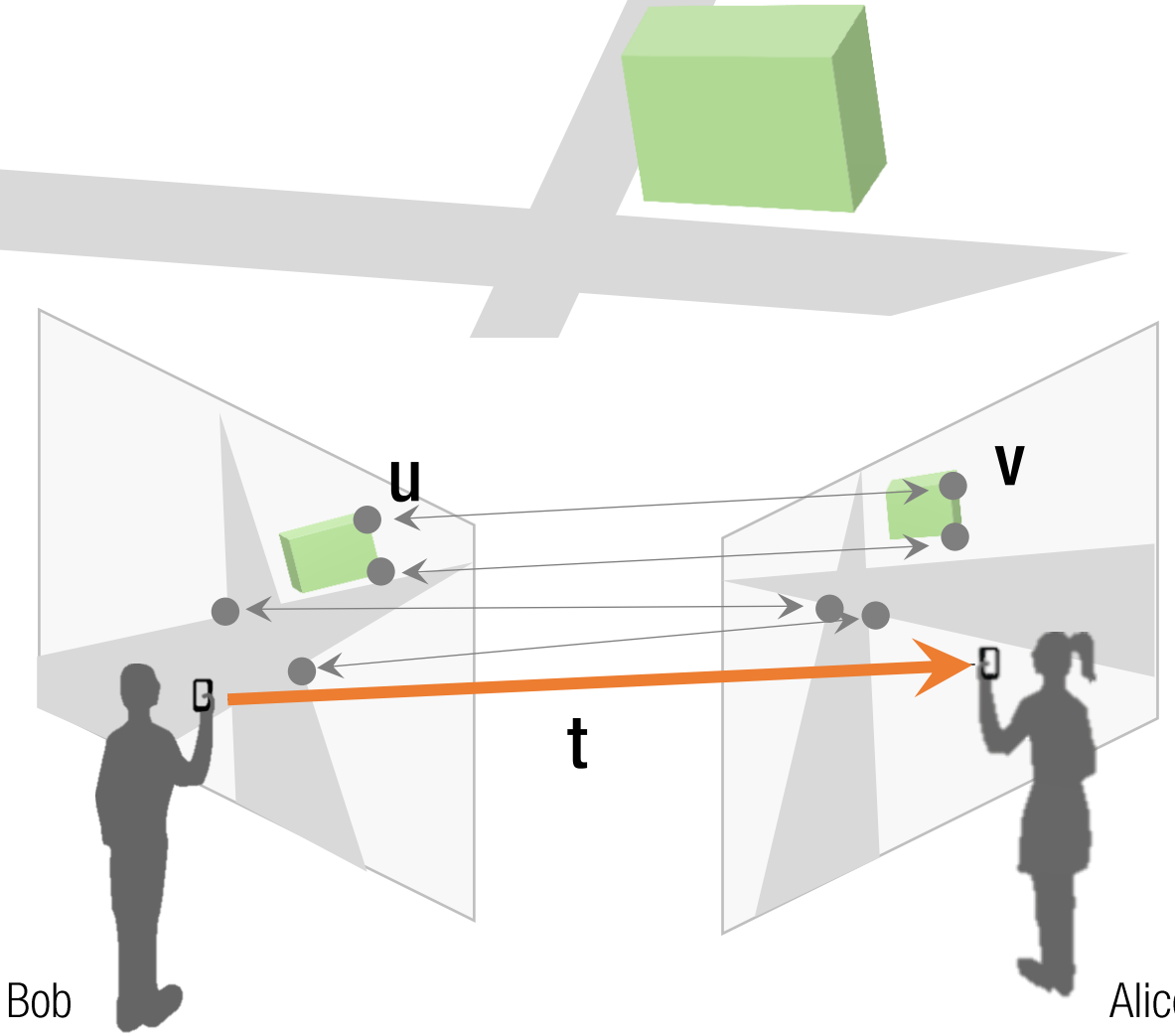
$$\mathbf{F} = \mathbf{F}(\mathbf{R}, \mathbf{t})$$

$$= \mathbf{K}^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1}$$

$$\rightarrow \mathbf{E} = \underbrace{\mathbf{K}^{T} \mathbf{F} \mathbf{K}}_{\text{Calibrated fundamental matrix}} \quad \text{where} \quad \mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$$

$$\mathbf{t} =$$

Camera Pose from Essential Matrix (Translation)



Essential Matrix:

$$\mathbf{F} = \mathbf{F}(\mathbf{R}, \mathbf{t})$$

$$= \mathbf{K}^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1}$$

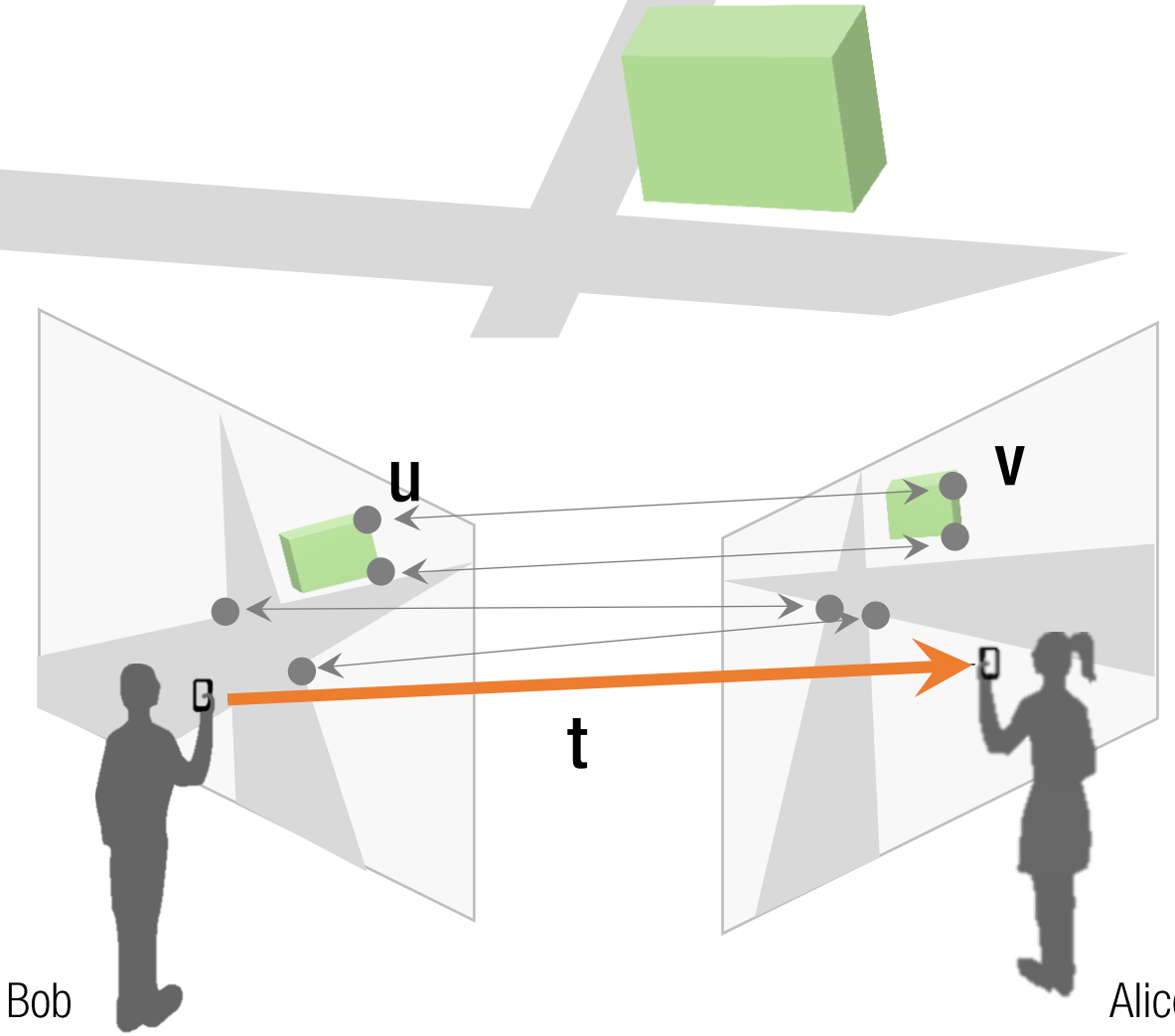
$$\rightarrow \mathbf{E} = \underline{\mathbf{K}^{T} \mathbf{F} \mathbf{K}} \qquad \text{where } \mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$$

Calibrated fundamental matrix

Left null space of E is translation vector, **t**:

$$\mathbf{t} =$$

Camera Pose from Essential Matrix (Translation)



Essential Matrix:

$$\mathbf{F} = \mathbf{F}(\mathbf{R}, \mathbf{t})$$

$$= \mathbf{K}^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1}$$

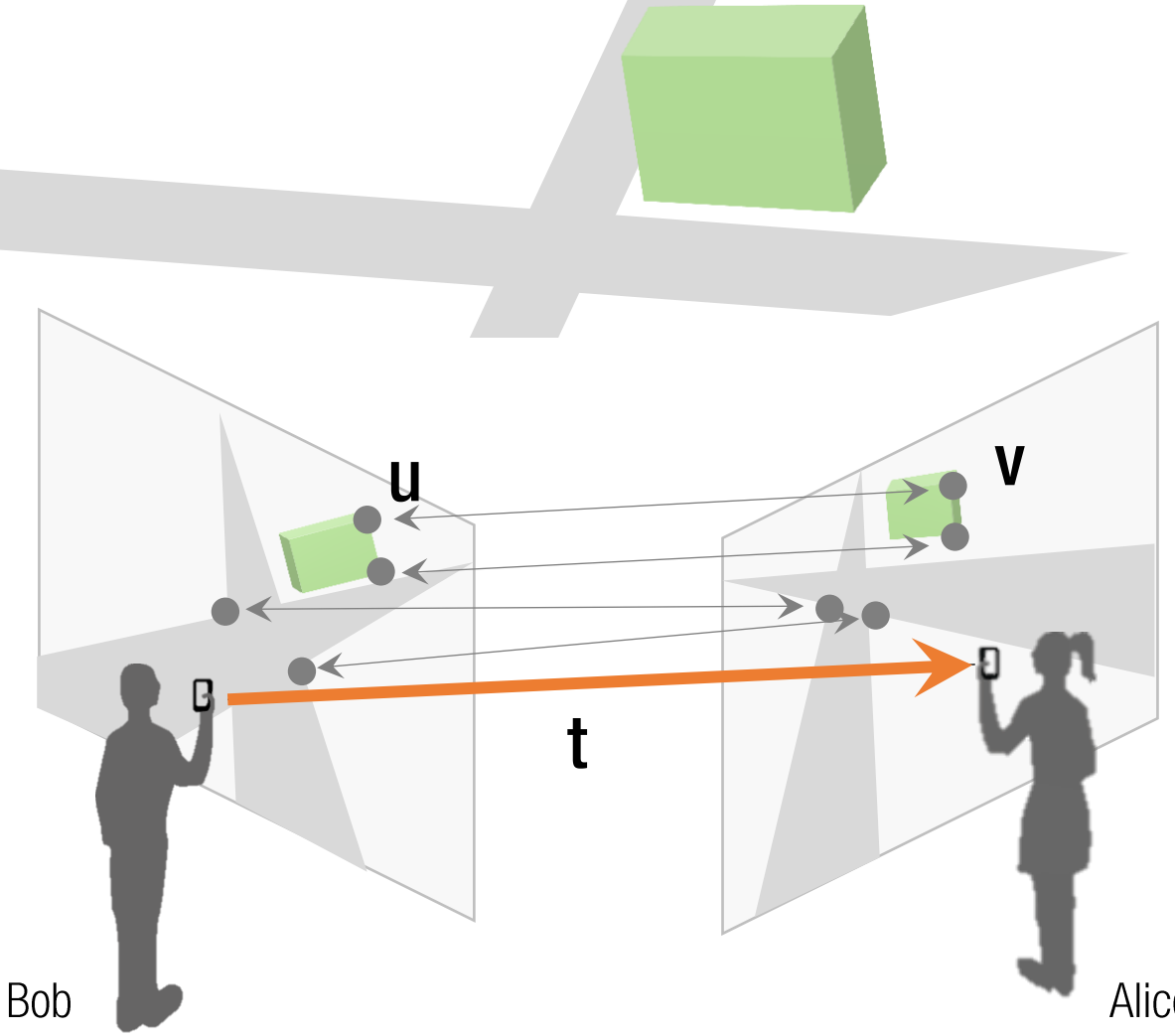
$$\rightarrow \mathbf{E} = \underline{\mathbf{K}^{T} \mathbf{F} \mathbf{K}} \qquad \text{where } \mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$$

Calibrated fundamental matrix

Left null space of E is translation vector, **t**:

$$\mathbf{t} = \pm \text{null}(\mathbf{E}^{T}) = \pm \text{null}(([\mathbf{t}]_{\times} \mathbf{R})^{T})$$

Camera Pose from Essential Matrix (Translation)



Essential Matrix:

$$\mathbf{F} = \mathbf{F}(\mathbf{R}, \mathbf{t})$$

$$= \mathbf{K}^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1}$$

$$\rightarrow \mathbf{E} = \underline{\mathbf{K}^{T} \mathbf{F} \mathbf{K}} \qquad \text{where } \mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R}$$

Calibrated fundamental matrix

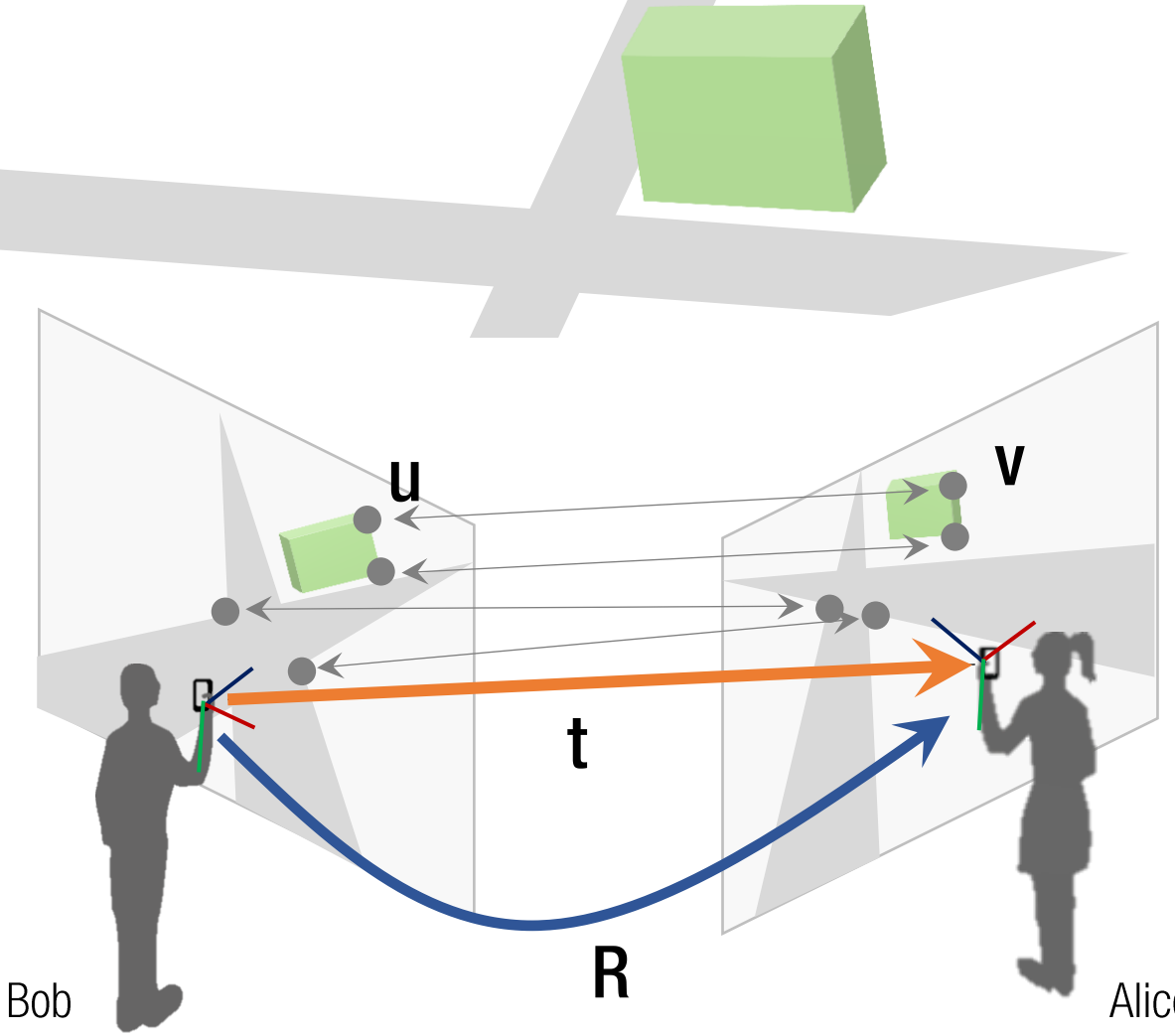
Left null space of E is translation vector, **t**:

$$\mathbf{t} = \pm \text{null}(\mathbf{E}^{T}) = \pm \text{null}(([\mathbf{t}]_{\times} \mathbf{R})^{T})$$

$$\because \mathbf{t}^{T} [\mathbf{t}]_{\times} \mathbf{R} = -([\mathbf{t}]_{\times} \mathbf{t})^{T} \mathbf{R} = -\underline{(\mathbf{t} \times \mathbf{t})}^{T} \mathbf{R} = 0$$

Self-cross product

Essential Matrix Decomposition



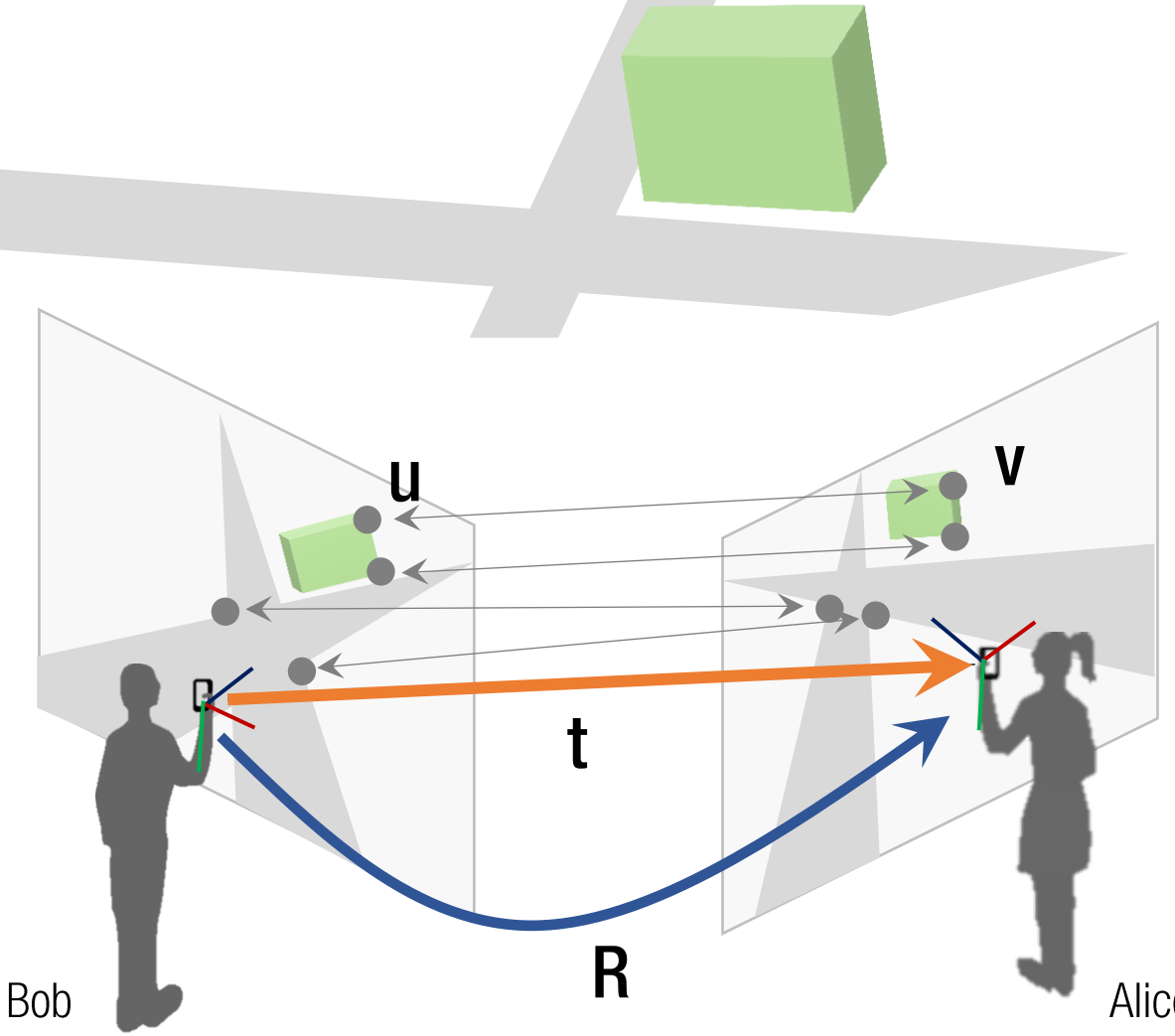
Left null space of E is translation vector, **t**:

$$\mathbf{t} = \text{null}(\mathbf{E}^T) = \text{null}((\lfloor \mathbf{t} \rfloor_\times \mathbf{R})^T)$$

$$\longrightarrow \quad \mathbf{t} = \mathbf{u}_3 \qquad \text{where} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}$$

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

Essential Matrix Decomposition



Left null space of E is translation vector, **t**:

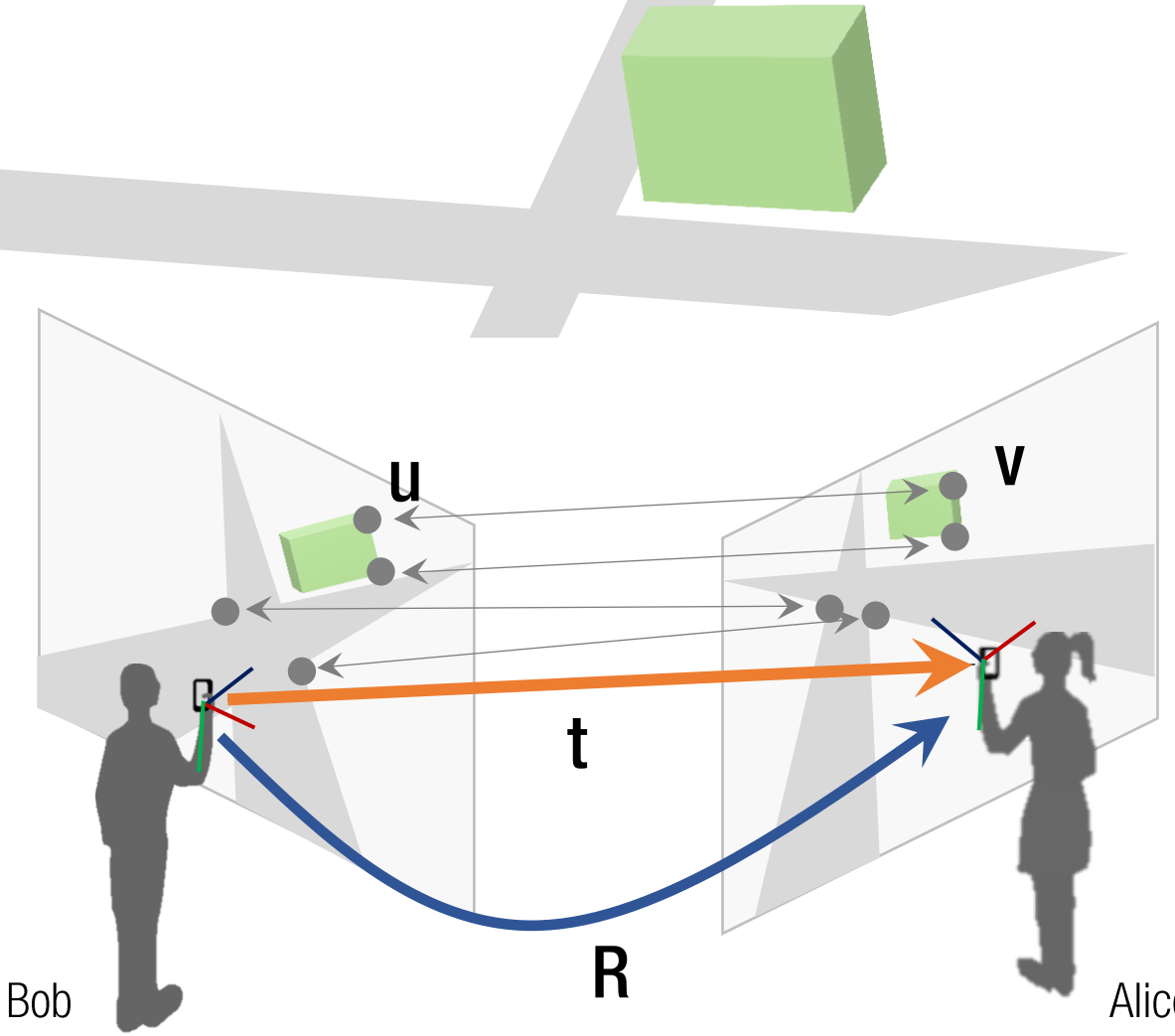
$$\mathbf{t} = \text{null}(\mathbf{E}^T) = \text{null}(([\mathbf{t}]_\times \mathbf{R})^T)$$

$$\longrightarrow \quad \mathbf{t} = \mathbf{u}_3 \qquad \text{where} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}$$

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

$$\longrightarrow \quad \mathbf{t} = \mathbf{u}_1 \times \mathbf{u}_2 \quad \text{(orthogonal matrix, } \mathbf{U}\text{)}$$

Essential Matrix Decomposition



Left null space of E is translation vector, **t**:

$$\mathbf{t} = \text{null}(\mathbf{E}^T) = \text{null}(([\mathbf{t}]_\times \mathbf{R})^T)$$

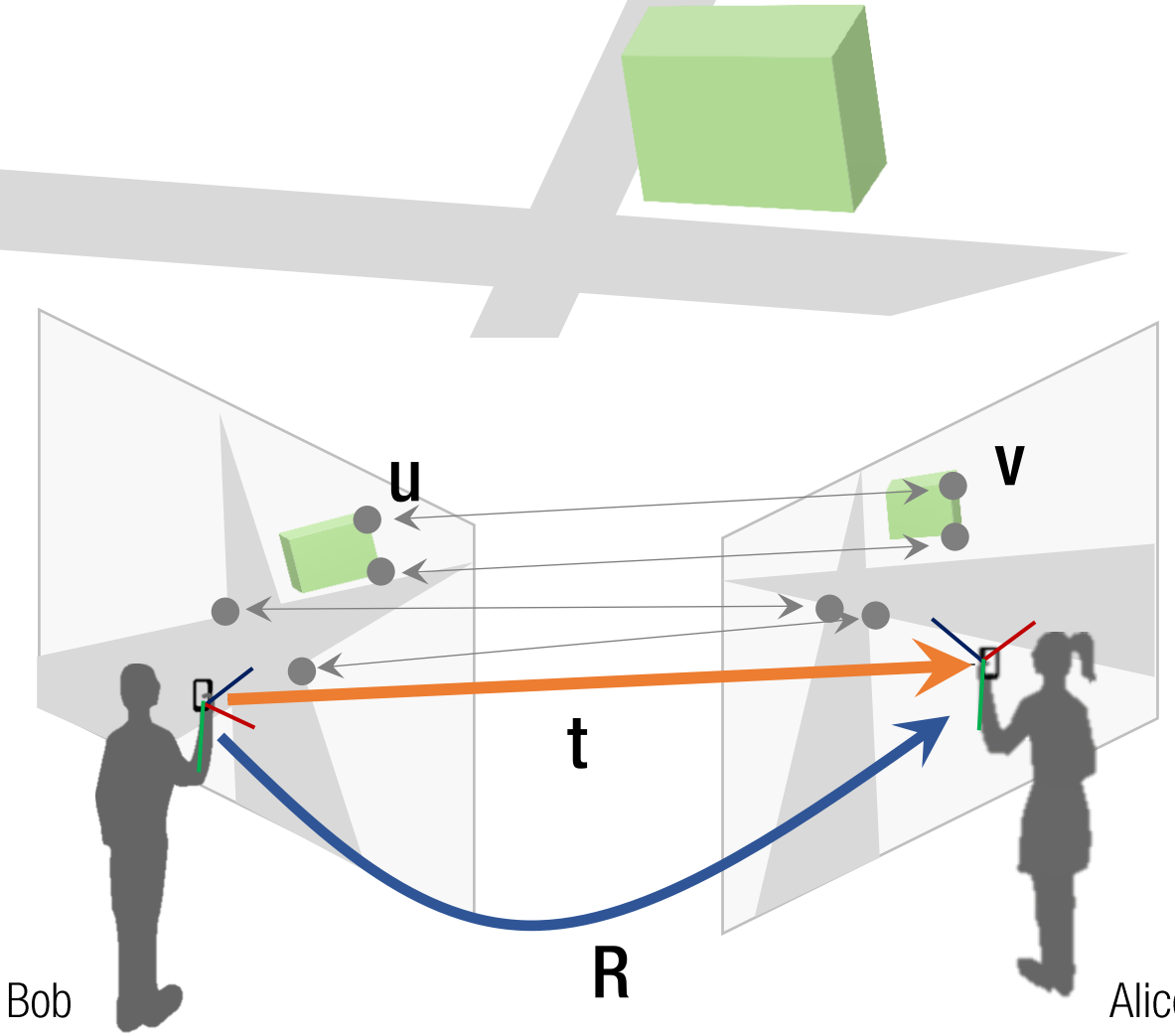
$$\longrightarrow \quad \mathbf{t} = \mathbf{u}_3 \qquad \text{where} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}$$

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

$$\longrightarrow \quad \mathbf{t} = \mathbf{u}_1 \times \mathbf{u}_2 \qquad \text{(orthogonal matrix, } \mathbf{U}\text{)}$$

$$[\mathbf{t}]_\times = [\mathbf{u}_1 \times \mathbf{u}_2]_\times = \mathbf{u}_2\mathbf{u}_1^T - \mathbf{u}_1\mathbf{u}_2^T$$

Essential Matrix Decomposition



Left null space of E is translation vector, **t**:

$$\mathbf{t} = \text{null}(\mathbf{E}^T) = \text{null}(([\mathbf{t}]_\times \mathbf{R})^T)$$

$$\longrightarrow \quad \mathbf{t} = \mathbf{u}_3 \qquad \text{where} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}$$

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

$$\longrightarrow \quad \mathbf{t} = \mathbf{u}_1 \times \mathbf{u}_2 \qquad \text{(orthogonal matrix, } \mathbf{U}\text{)}$$

$$[\mathbf{t}]_\times = [\mathbf{u}_1 \times \mathbf{u}_2]_\times = \mathbf{u}_2\mathbf{u}_1^T - \mathbf{u}_1\mathbf{u}_2^T$$

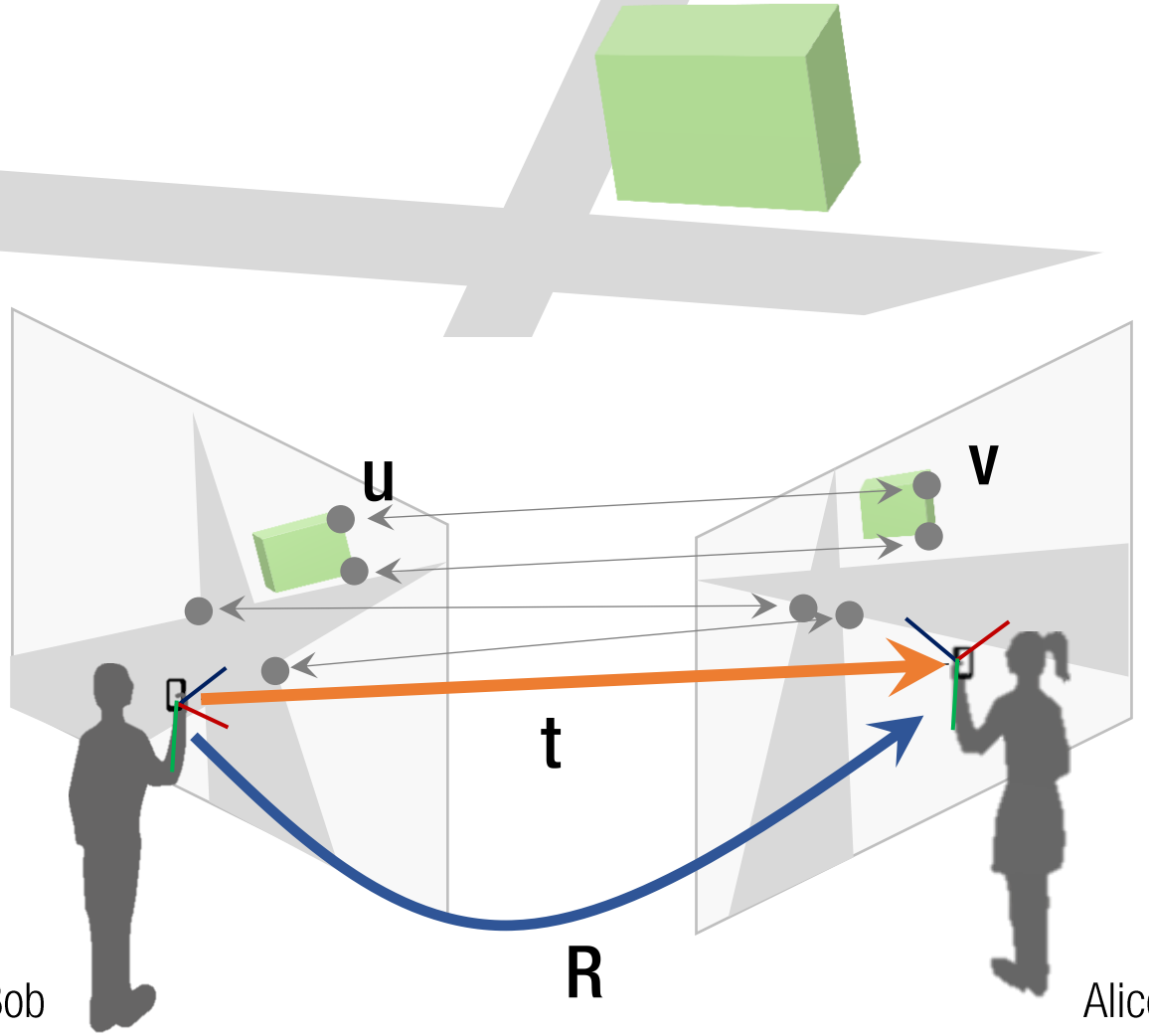
$$= \mathbf{U}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{U}^T$$

Prove!

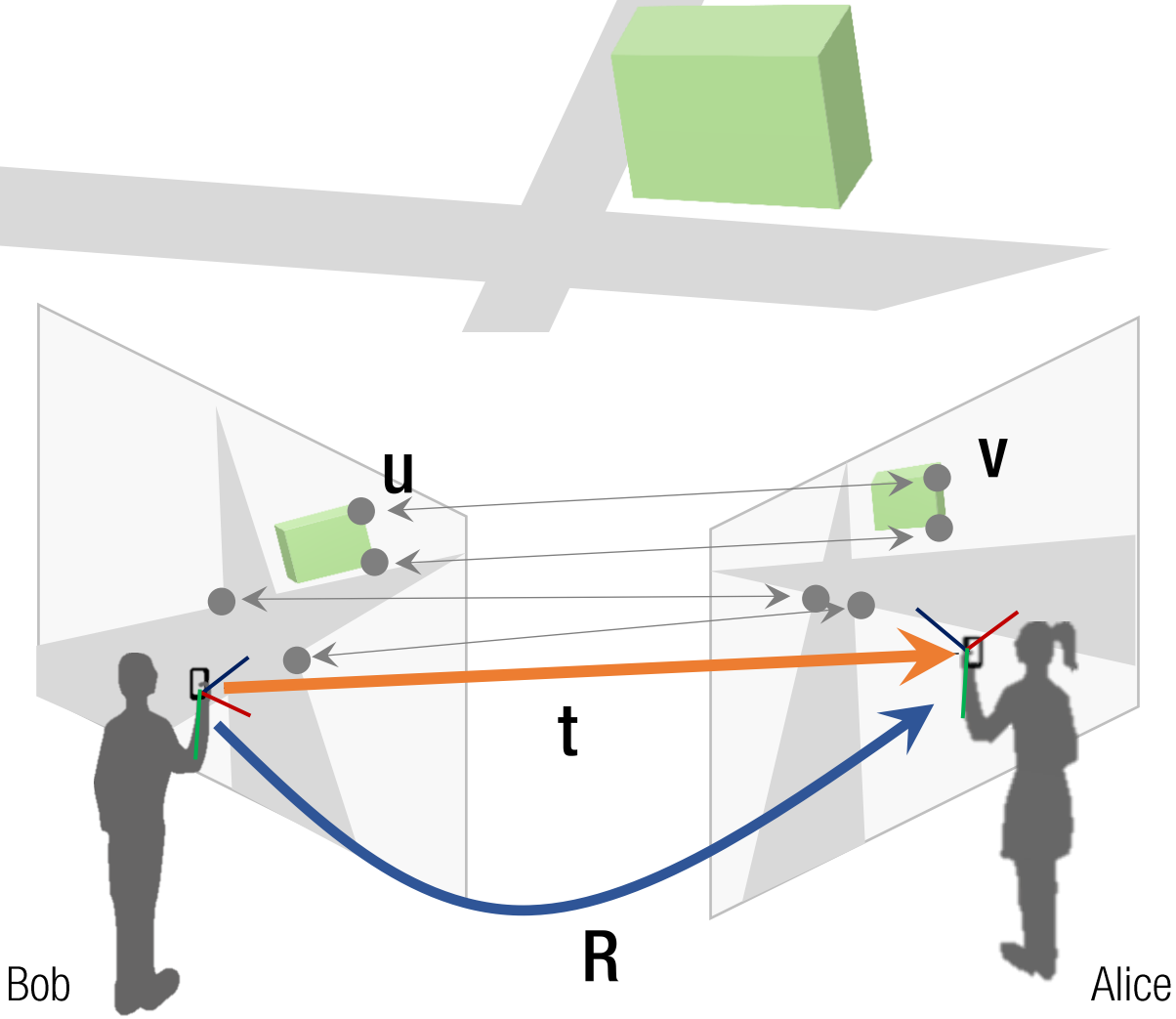
Essential Matrix Decomposition

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

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



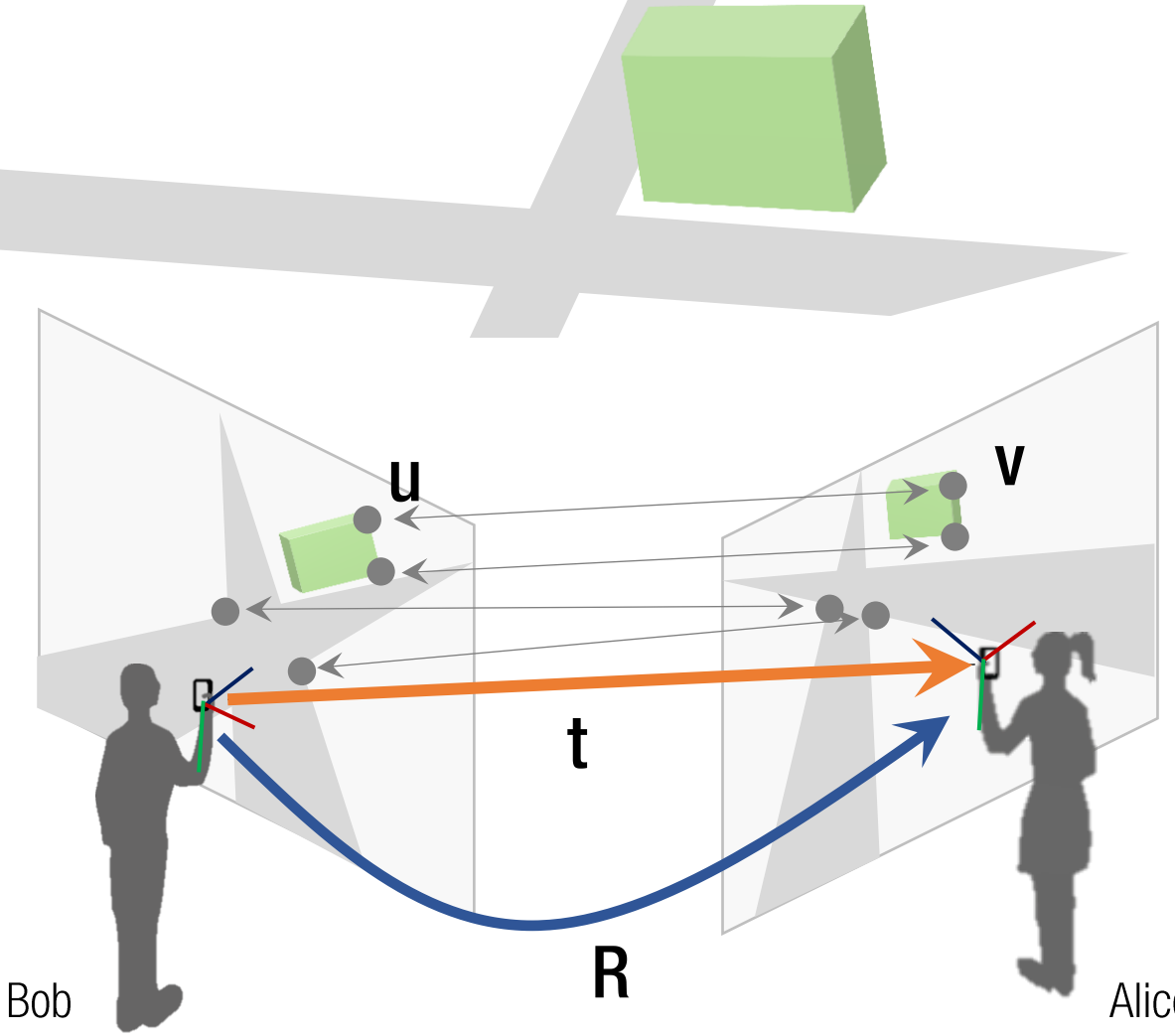
Essential Matrix Decomposition



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

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

Essential Matrix Decomposition



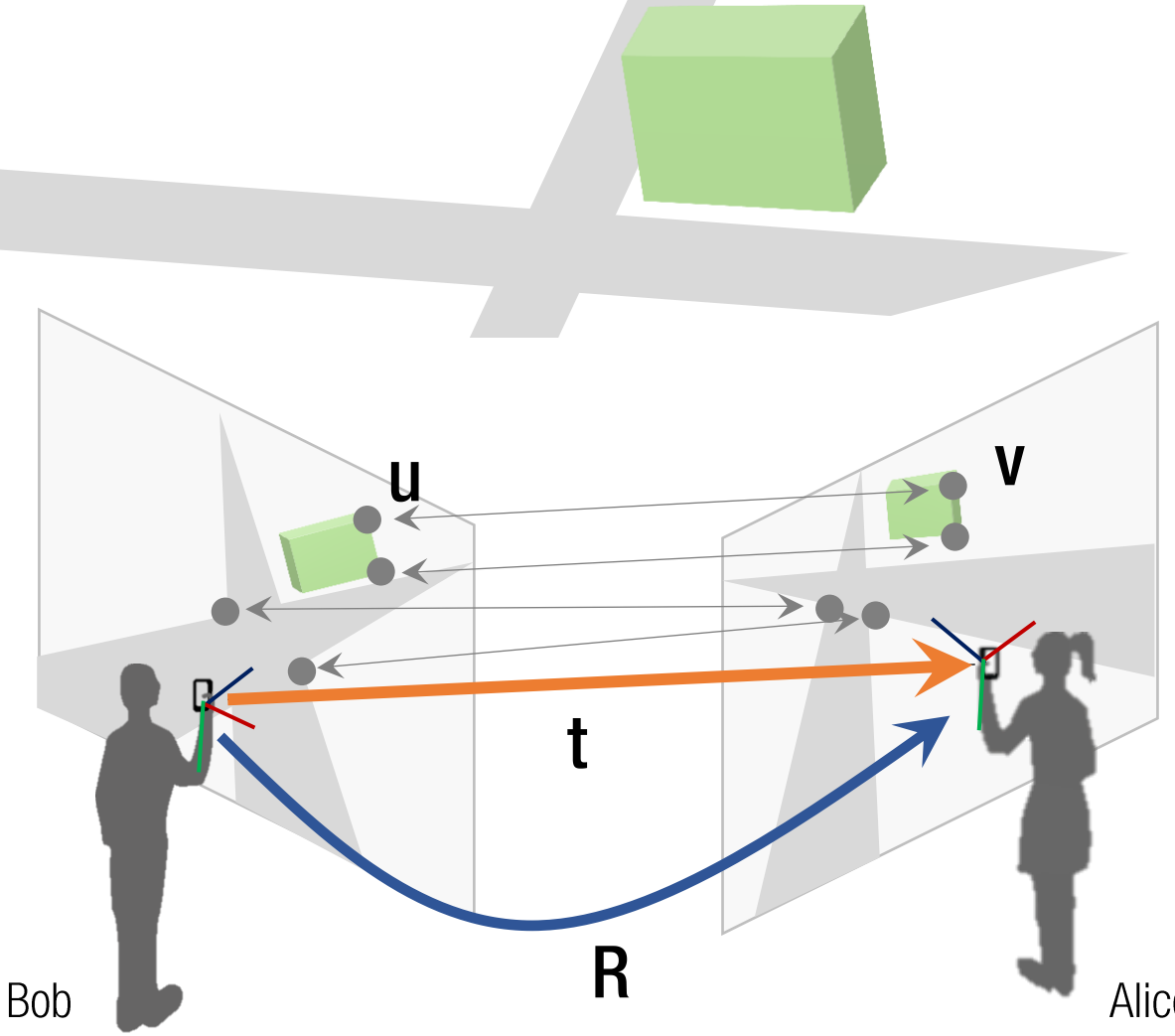
$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R} = \mathbf{U} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{U}^\mathsf{T} \mathbf{R} = \mathbf{U} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \mathbf{V}^\mathsf{T}$$

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

Define $\mathbf{R} = \underline{\mathbf{UWV}^\mathsf{T}}$

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

Essential Matrix Decomposition



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

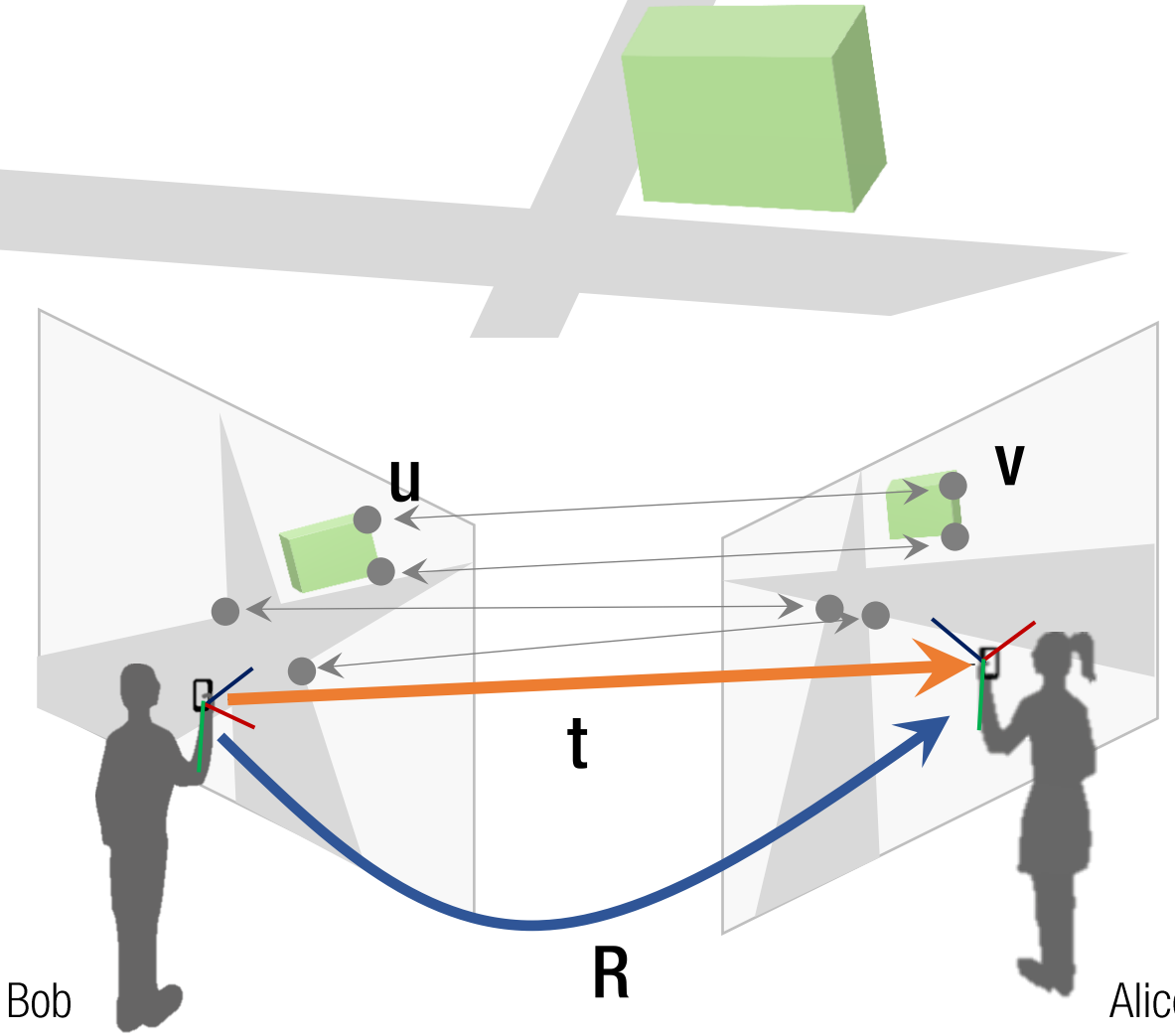
where $\mathbf{R} \in SO(3)$

Define $\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^\mathsf{T}$

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

$$\rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{W}$$

Essential Matrix Decomposition



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

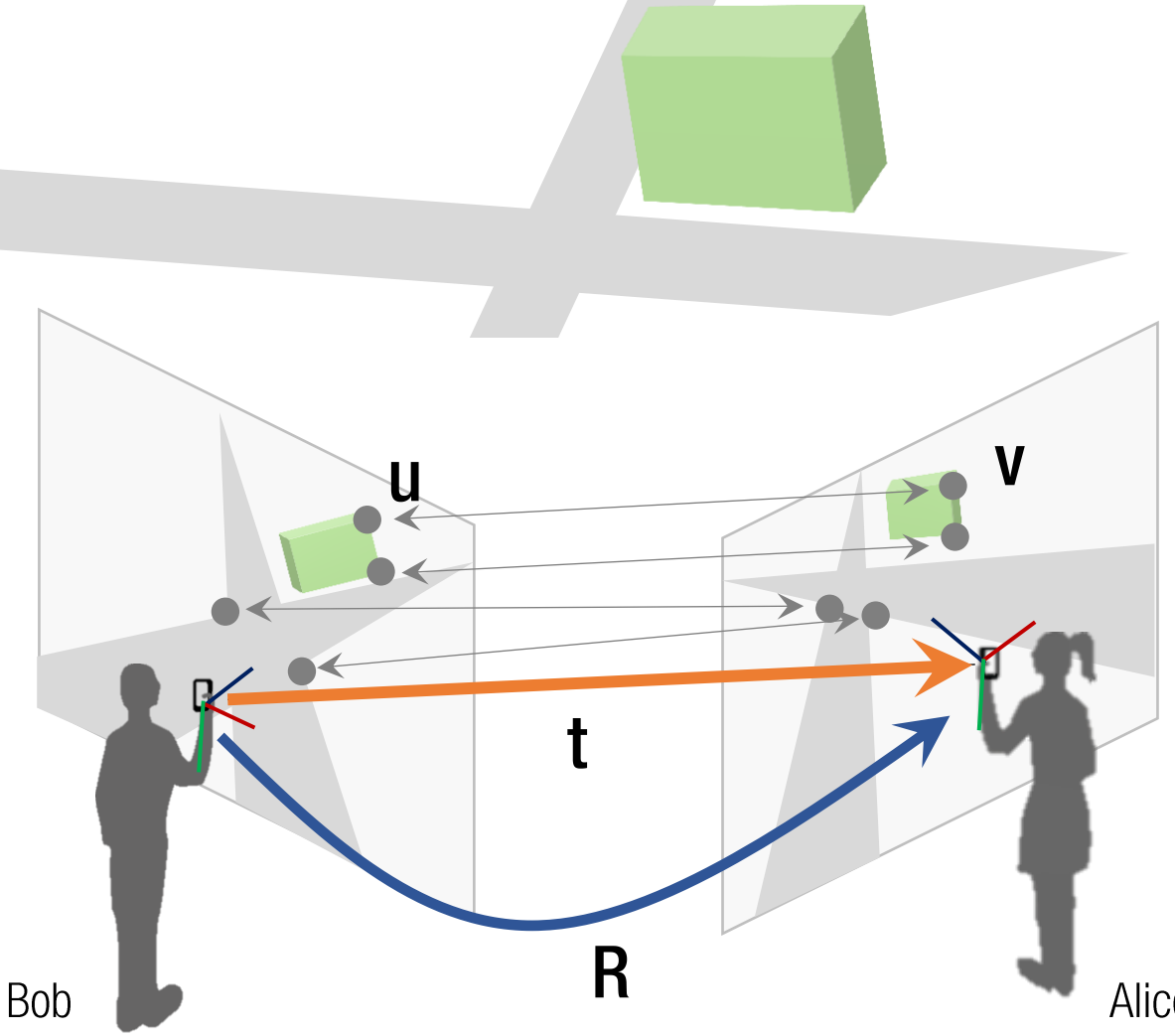
where $\mathbf{R} \in SO(3)$

Define $\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^\mathsf{T}$

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

$$\rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{W} \qquad \mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ or } \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Camera Pose from Essential Matrix (Rotation)

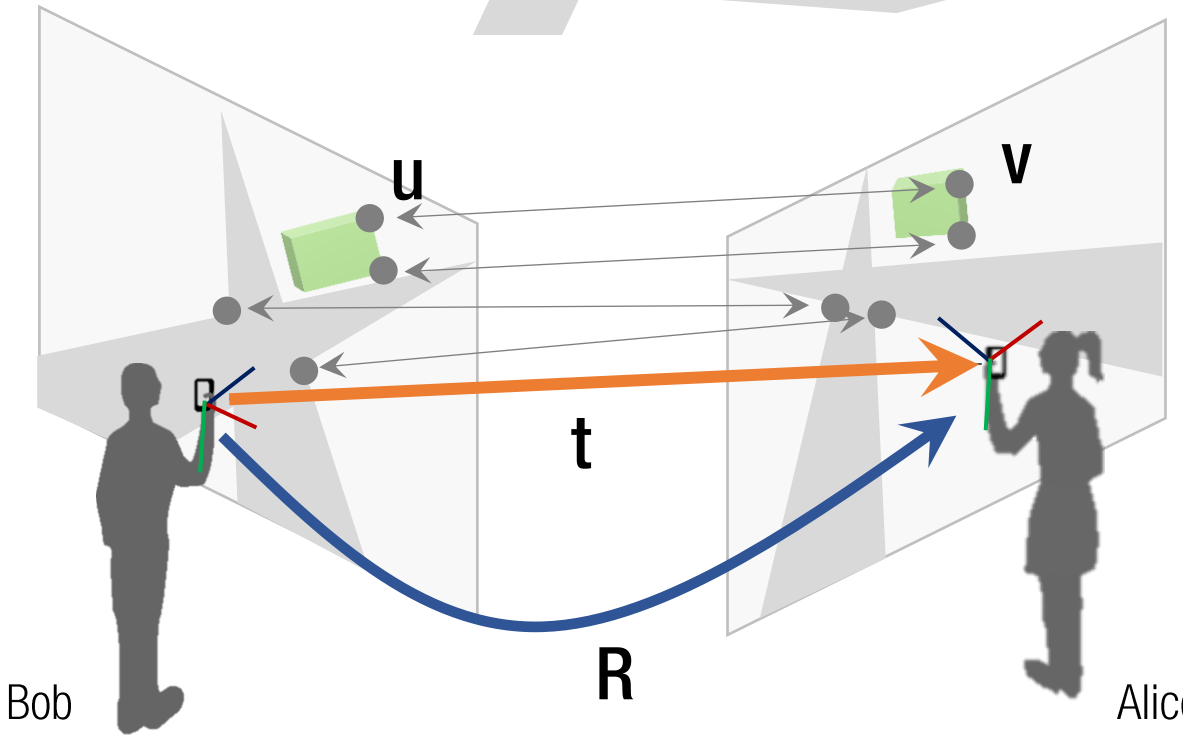


$$E = [t]_\times R = U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T R = U \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} V^T$$

where $R \in SO(3)$

$$R = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T \text{, or } U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T$$

Where Am I?



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

$$\mathbf{t} = \pm\mathbf{u}_3$$

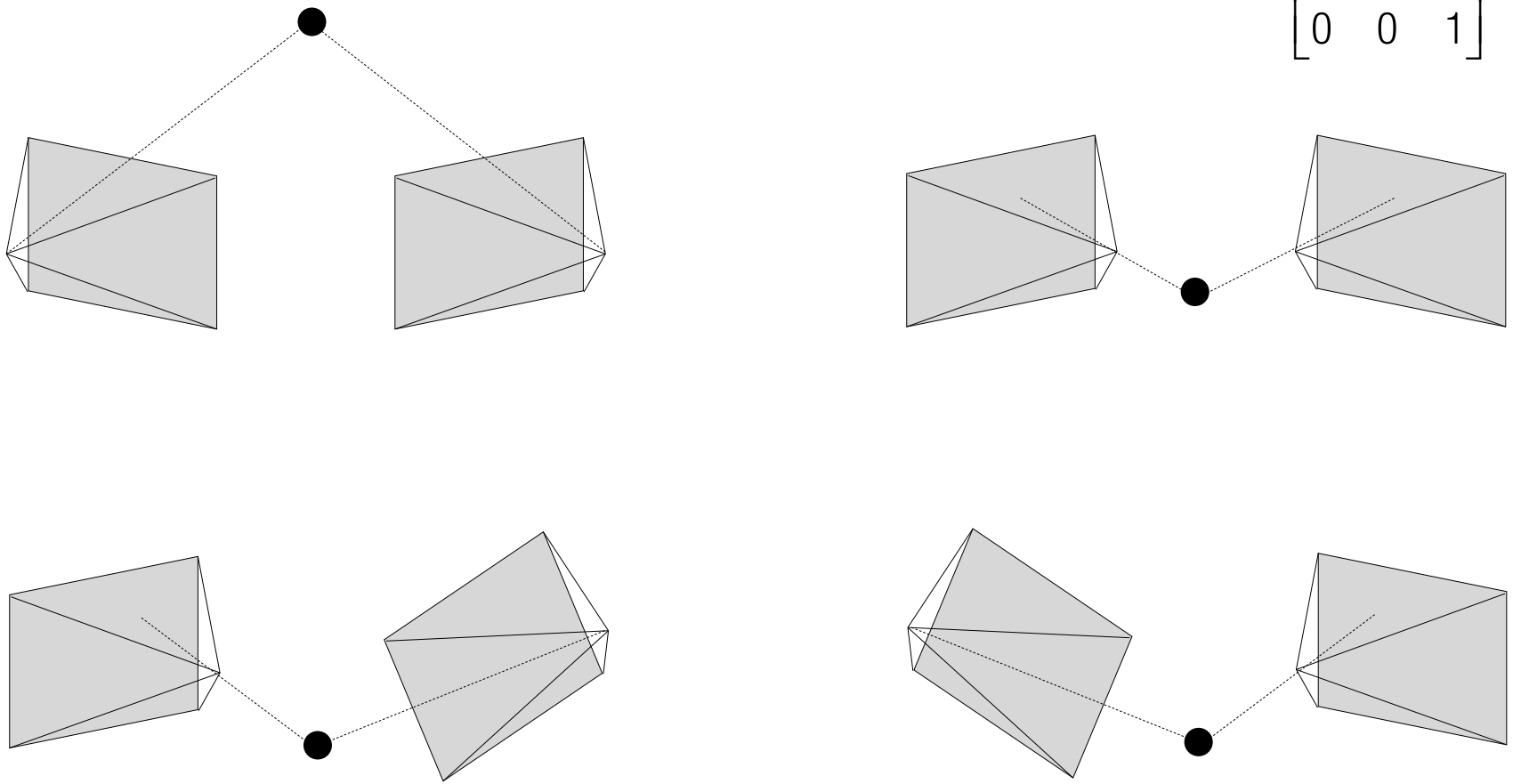
$$\mathbf{R} = \mathbf{U}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{V}^{\mathsf{T}}, \text{ or } \mathbf{U}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{V}^{\mathsf{T}}$$

⟶ Four configurations

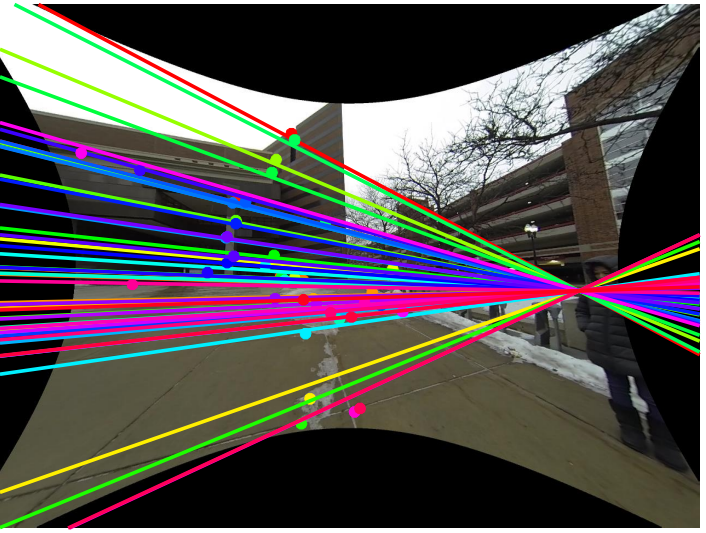
Four Configurations

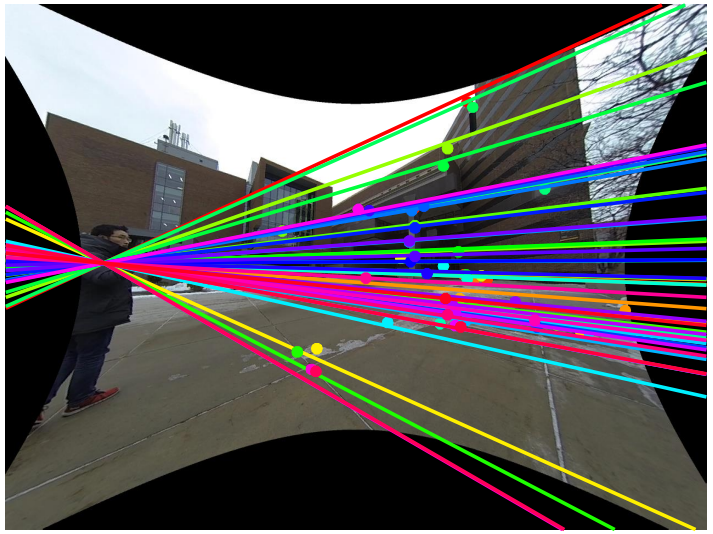
$$\mathbf{t} = \pm \mathbf{u}_3$$

$$\mathbf{R} = \mathbf{U} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{V}^T \text{ , or } \mathbf{U} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{V}^T$$



Camera Pose Estimation





$$E = K^T F K$$

```
function E = ComputeEssentialMatrix(F, K)

E = K' * F * K;

[u d v] = svd(E);

d(1,1) = 1;
d(2,2) = 1;
d(3,3) = 0;

E = u * d * v';
```

SVD cleanup

D =

```
1.0468         0         0
     0    0.9975         0
     0         0    0.0000
```

Before cleanup

D =

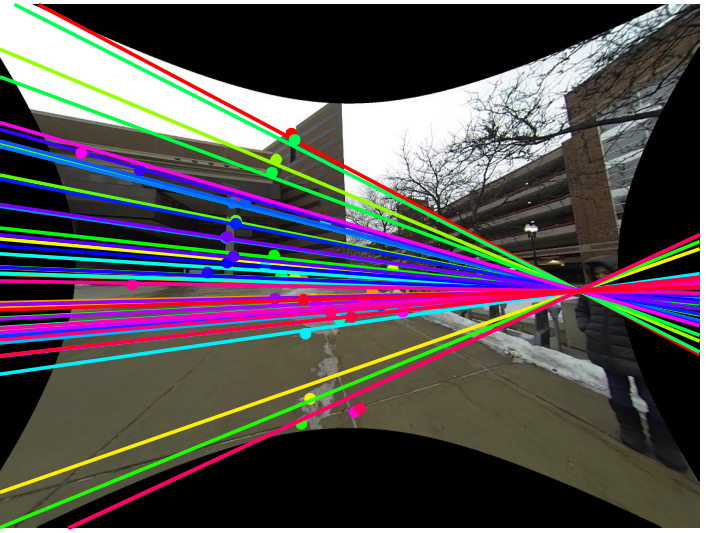
```
1.0000         0         0
     0    1.0000         0
     0         0    0.0000
```

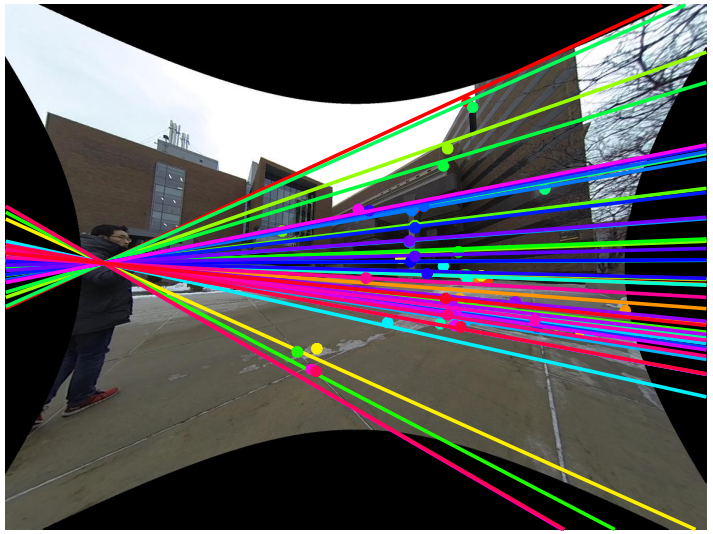
After cleanup

Camera Pose Estimation

$$\mathbf{t} = \pm \mathbf{u}_3$$

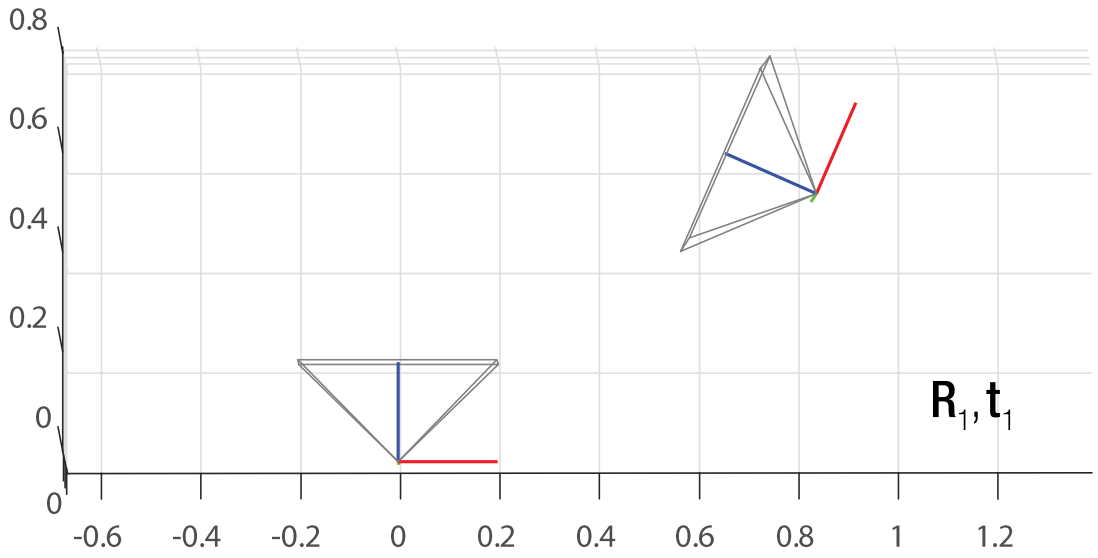
$$\mathbf{R} = \mathbf{U} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{V}^T \text{ , or } \mathbf{U} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{V}^T$$

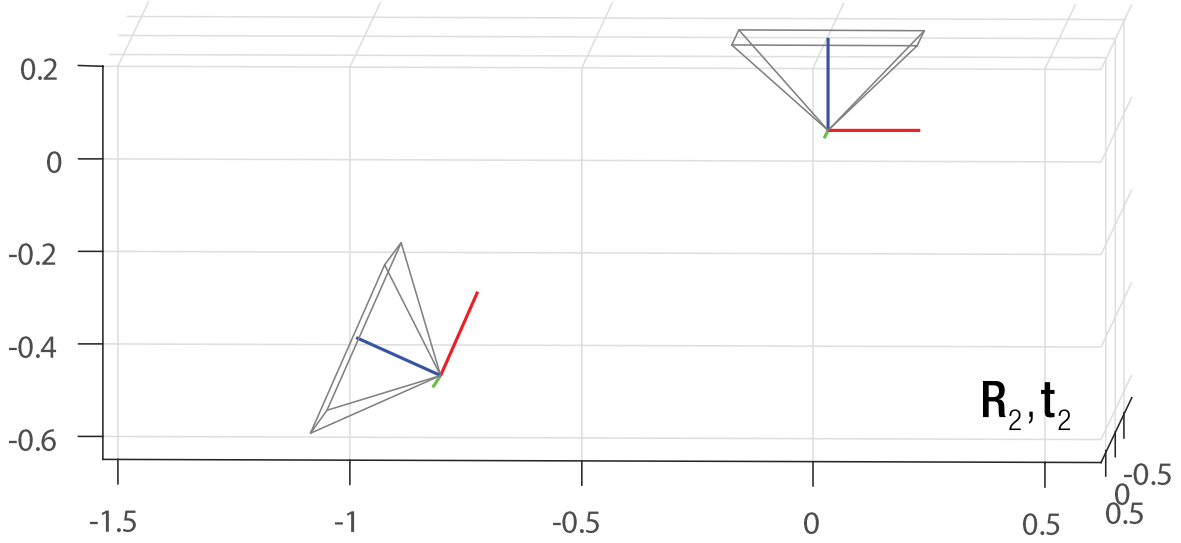


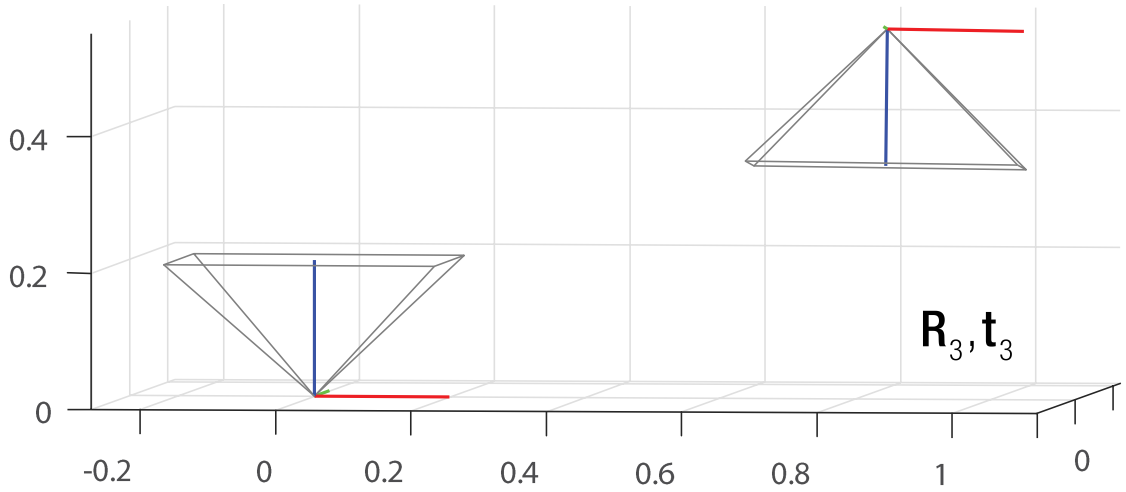


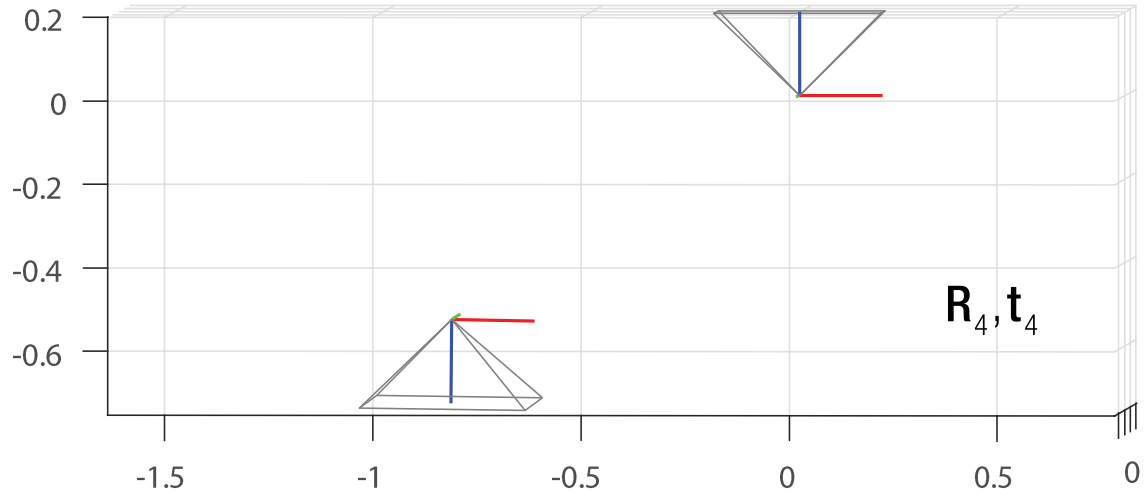
```
function [R1 t1, R2, t2, R3, t3, R4, t4] = …
CameraPoseFromEssentialMatrix(E)

[U D V] = svd(E);

W = [0 -1 0;
    1 0 0;
    0 0 1];

t1 = U(:,3);
R1 = U * W * V';
if det(R1) < 0
  t1 = -t1;  R1 = -R1;
end

…
```

$\det(\mathbf{R}) = 1$

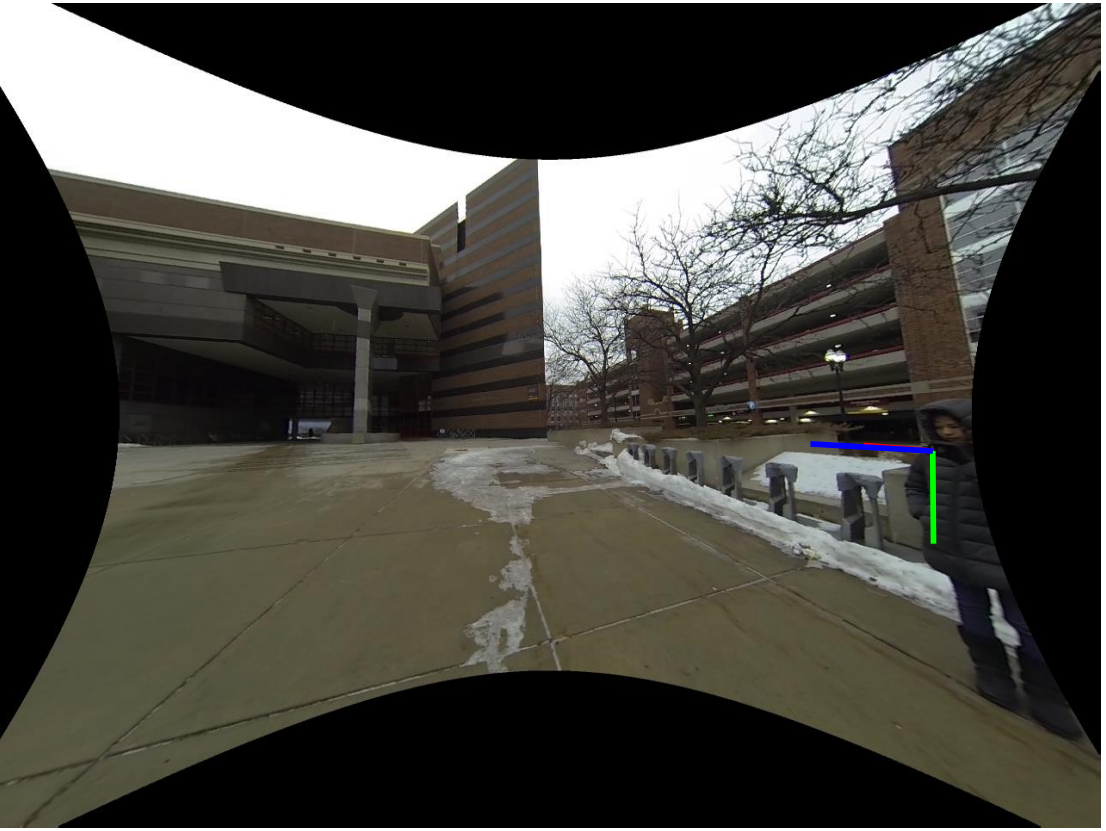

0.8
0.6
0.4
0.2
0
0
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
R_1, t_1

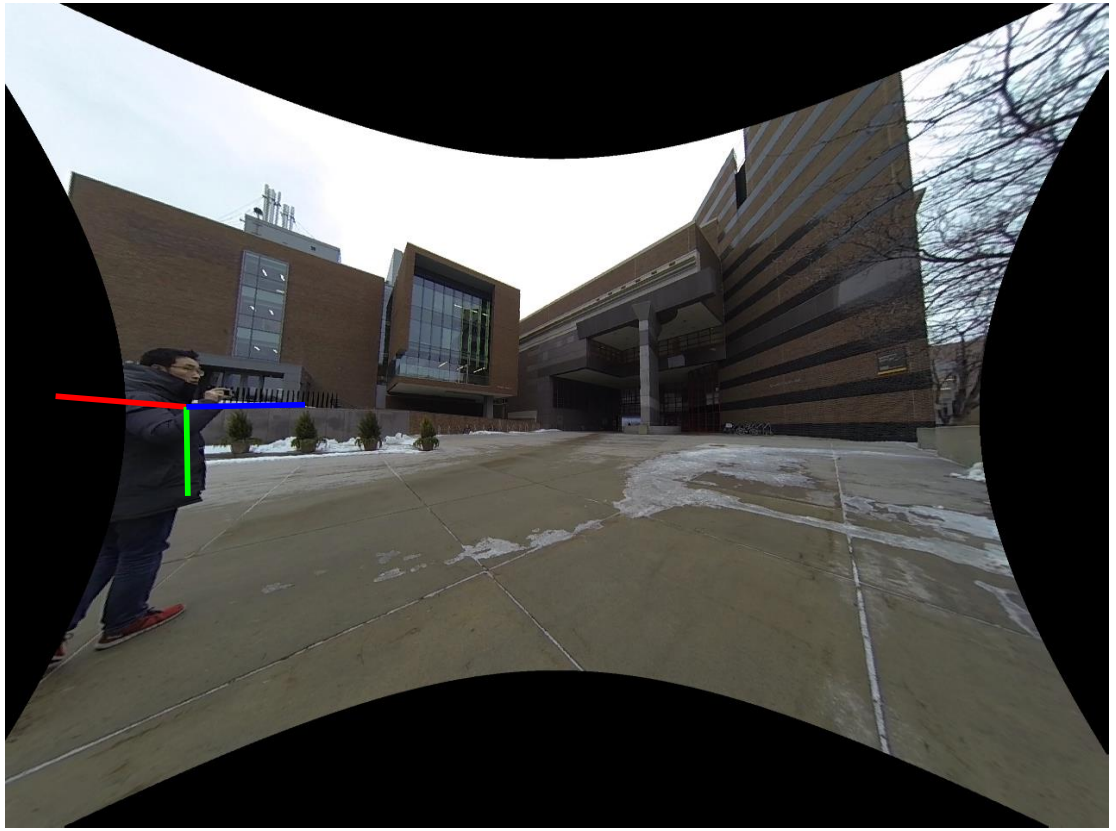

0.2
0
-0.2
-0.4
-0.6
-1.5
-1
-0.5
0
0.5
-0.5
0
0.5
R_2, t_2


0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1
0
R_3, t_3


0.2
0
-0.2
-0.4
-0.6
-1.5
-1
-0.5
0
0.5
0
R_4, t_4

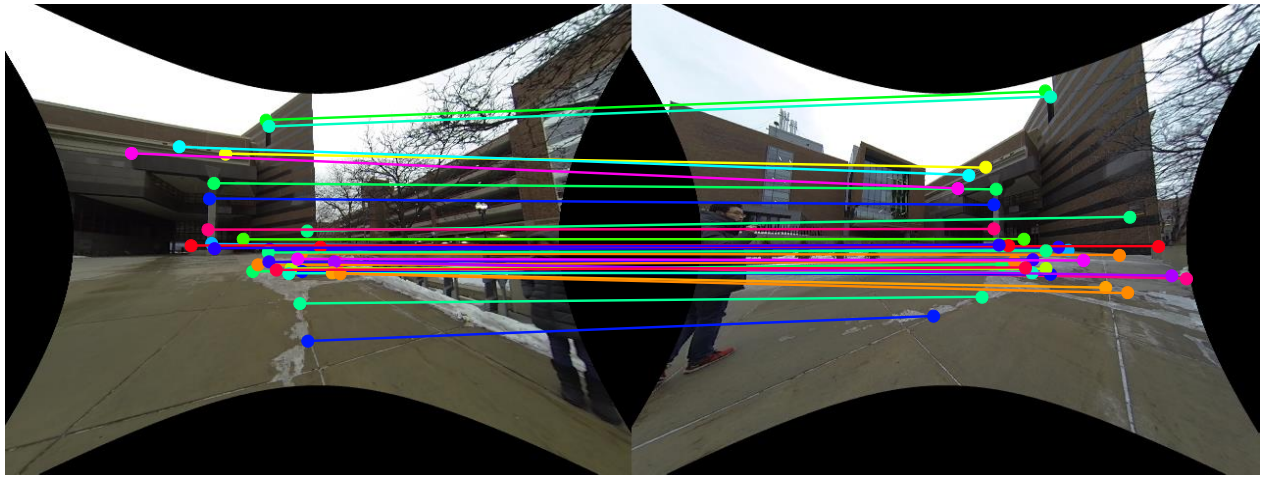
Camera Image Projection





Feature Matching

How Many Correspondences?



$$\begin{bmatrix} u_1^x v_1^x & u_1^y v_1^x & v_1^x & u_1^x v_1^y & u_1^y v_1^y & v_1^y & u_1^x & u_1^y & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_m^x v_m^x & u_m^y v_m^x & v_m^x & u_m^x v_m^y & u_m^y v_m^y & v_m^y & u_m^x & u_m^y & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ \vdots \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} \mathbf{x} = \mathbf{0}$$

What is minimum m?


u
v

Local Scale Invariant Feature Transform (SIFT)

SIFT automatically finds the optimal scale of feature point and its orientation.

Desired properties:

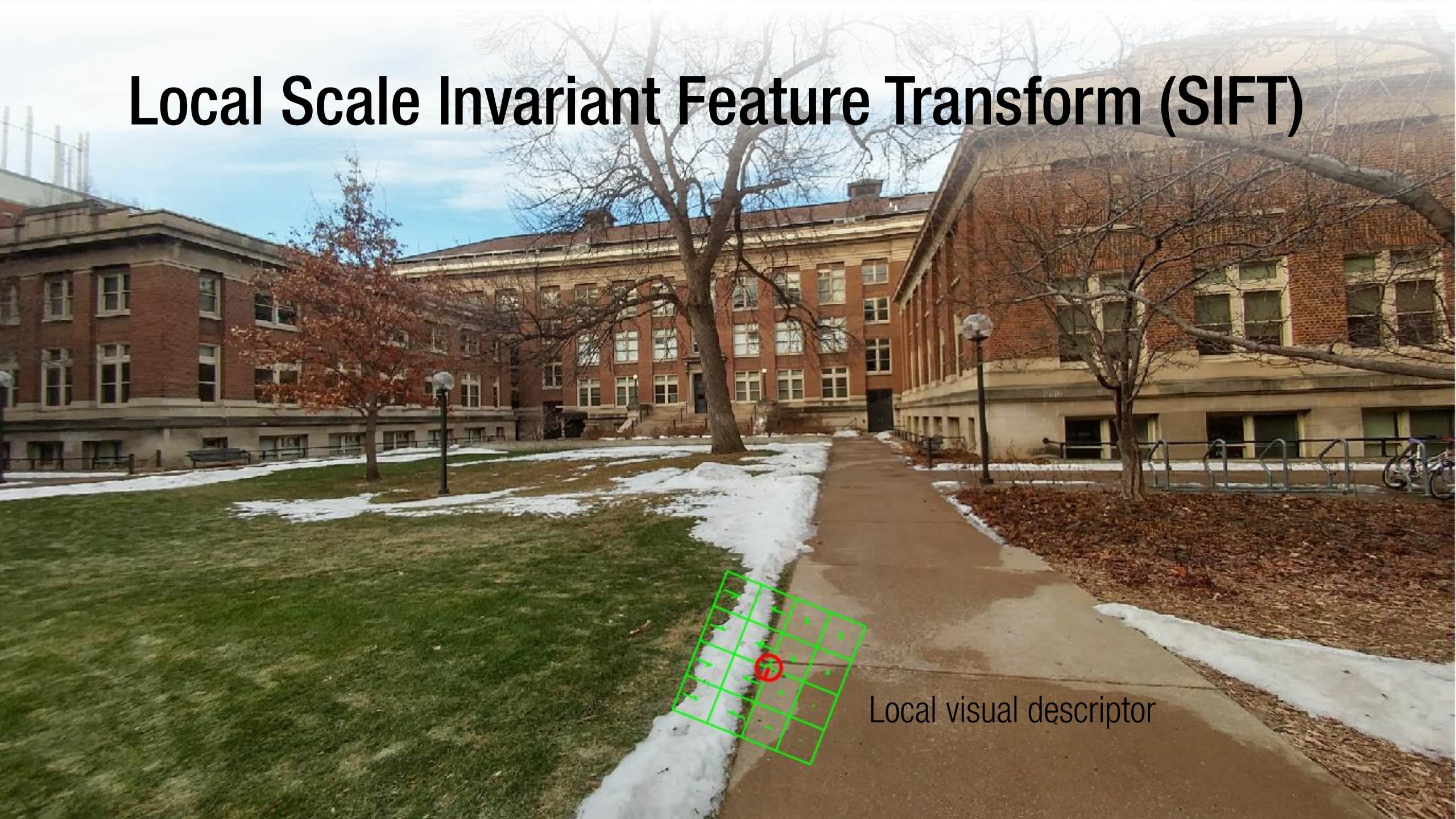
- Repeatability: the same point is repeatedly detected.
- Discriminativity: the point is unique.
- Orientation aware

More detail from CSCI 5561

Local Scale Invariant Feature Transform (SIFT)

Local visual descriptor

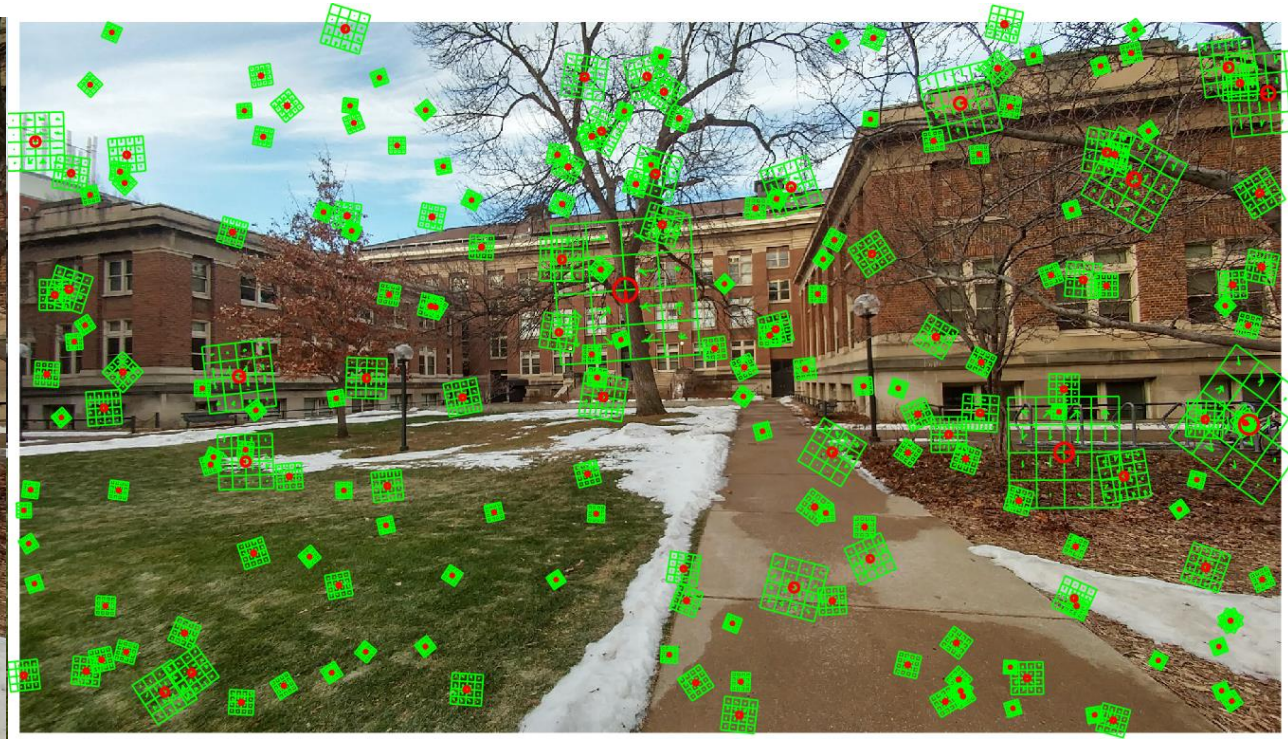
Local Scale Invariant Feature Transform (SIFT)



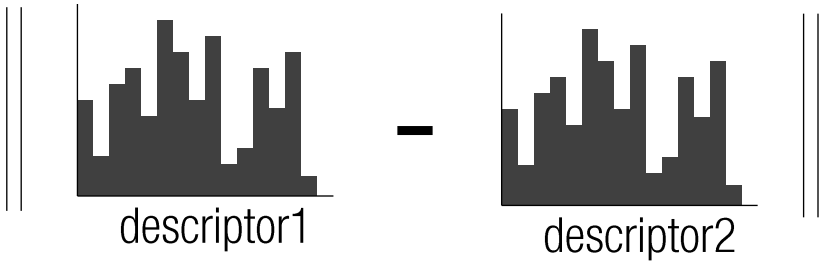
Local Scale Invariant Feature Transform (SIFT)

$$\left\| \text{descriptor1} - \text{descriptor2} \right\| = 0$$

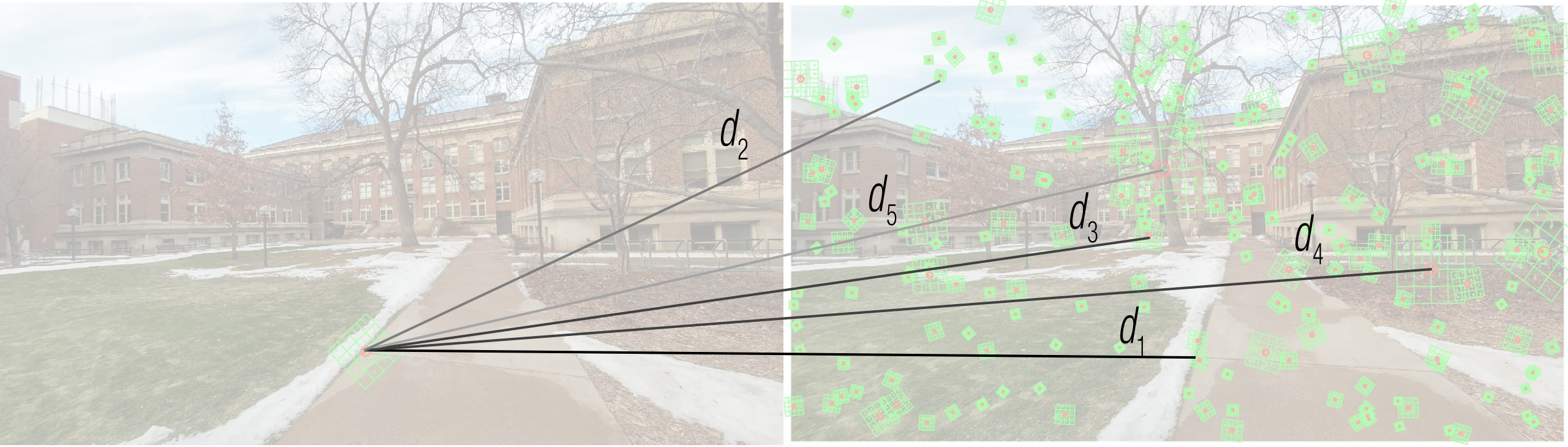
Local Scale Invariant Feature Transform (SIFT)



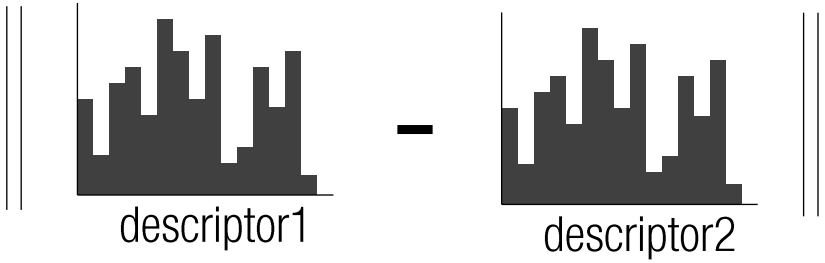
Feature match candidates



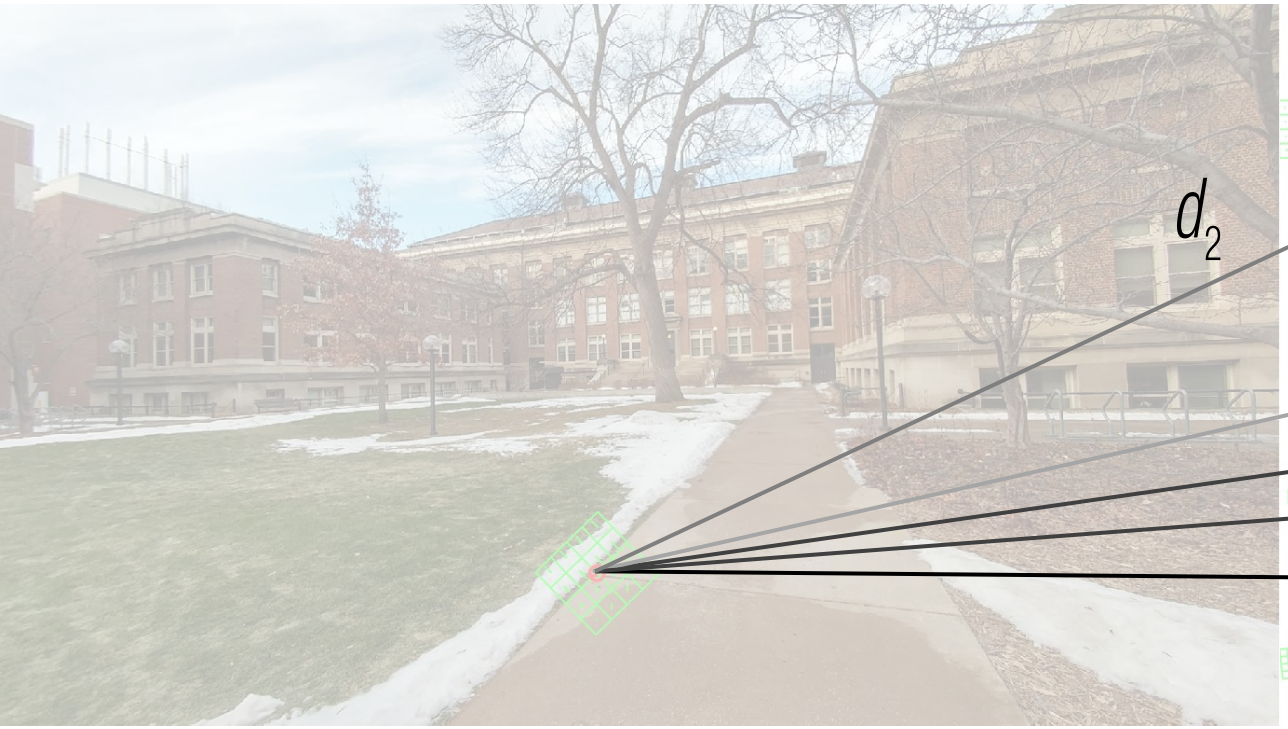
Nearest Neighbor Search

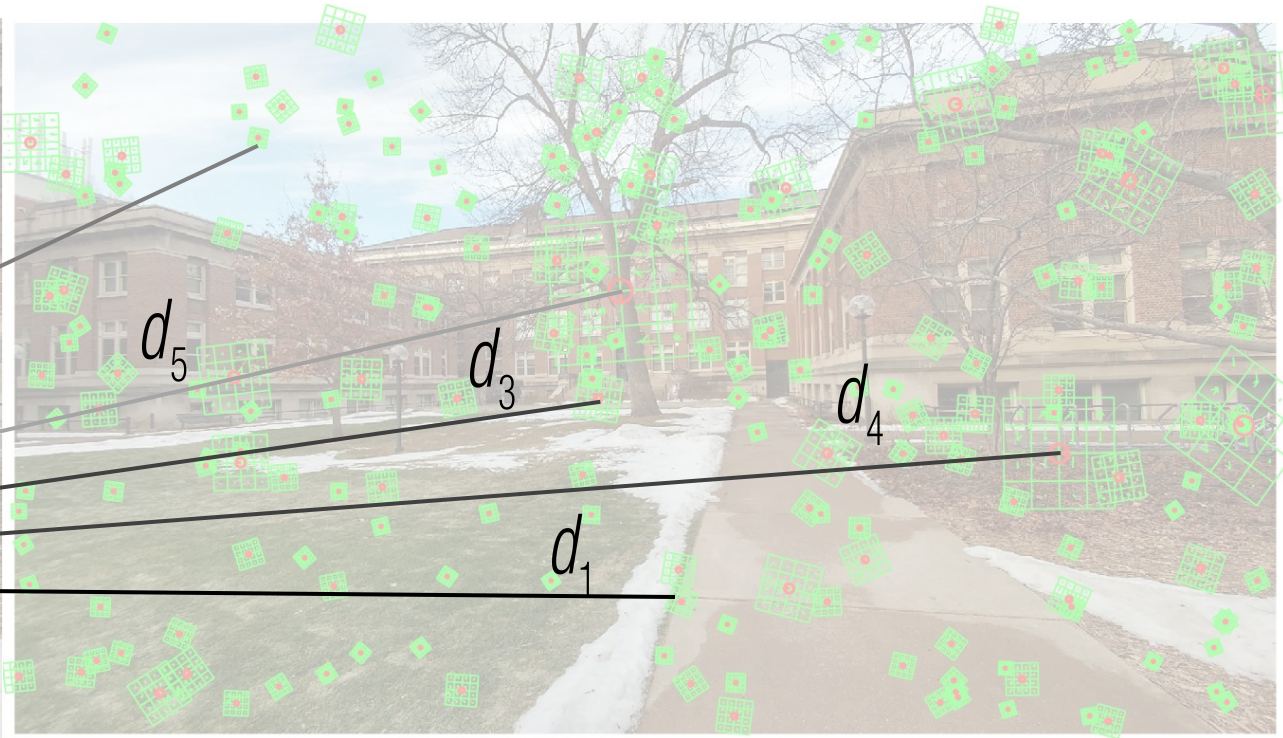


Feature match candidates



Nearest Neighbor Search

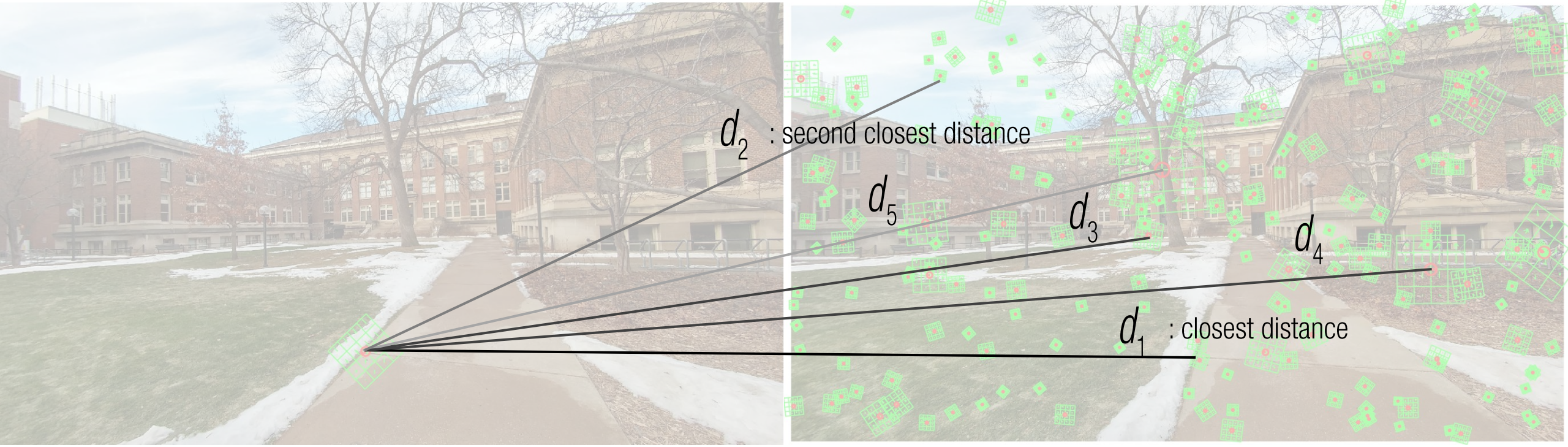




Feature match candidates

Discriminativity: how is the feature point unique?

Nearest Neighbor Search w/ Ratio Test

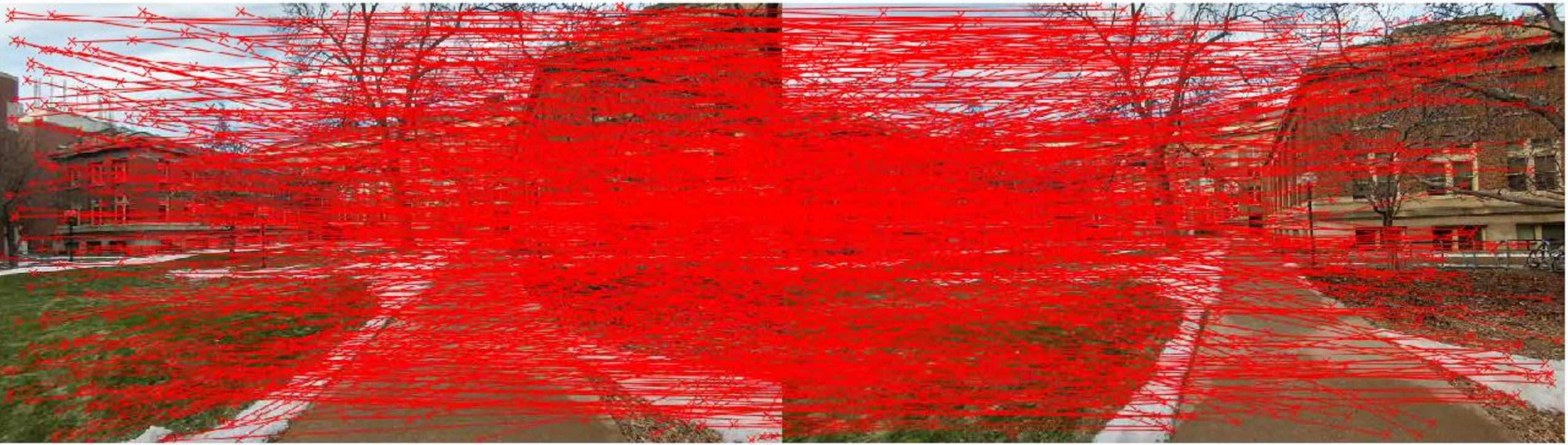


Feature match candidates

Discriminativity: how is the feature point unique?

$$\frac{d_1}{d_2} < 0.7$$

Nearest Neighbor Search w/o Ratio Test

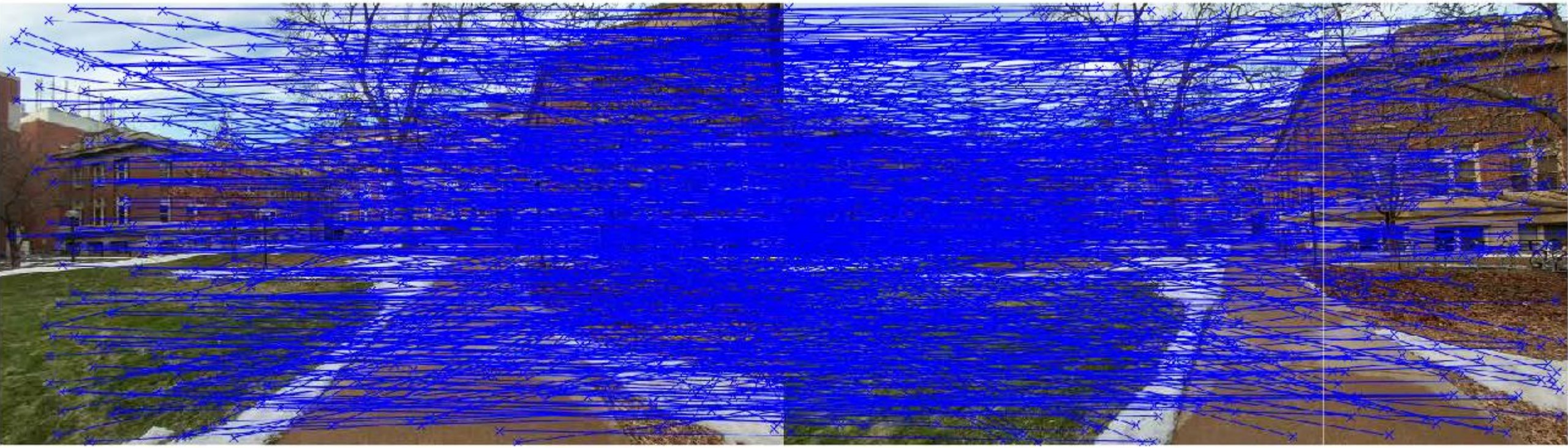


Left image→right image

Nearest Neighbor Search w/ Ratio Test

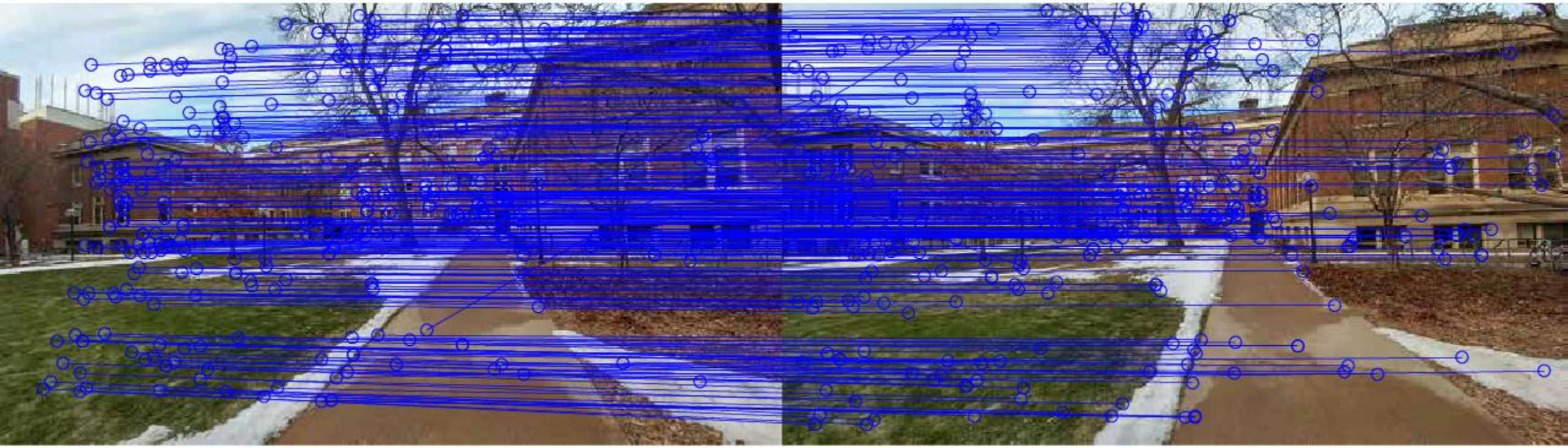
Left image→right image

Nearest Neighbor Search w/o Ratio Test



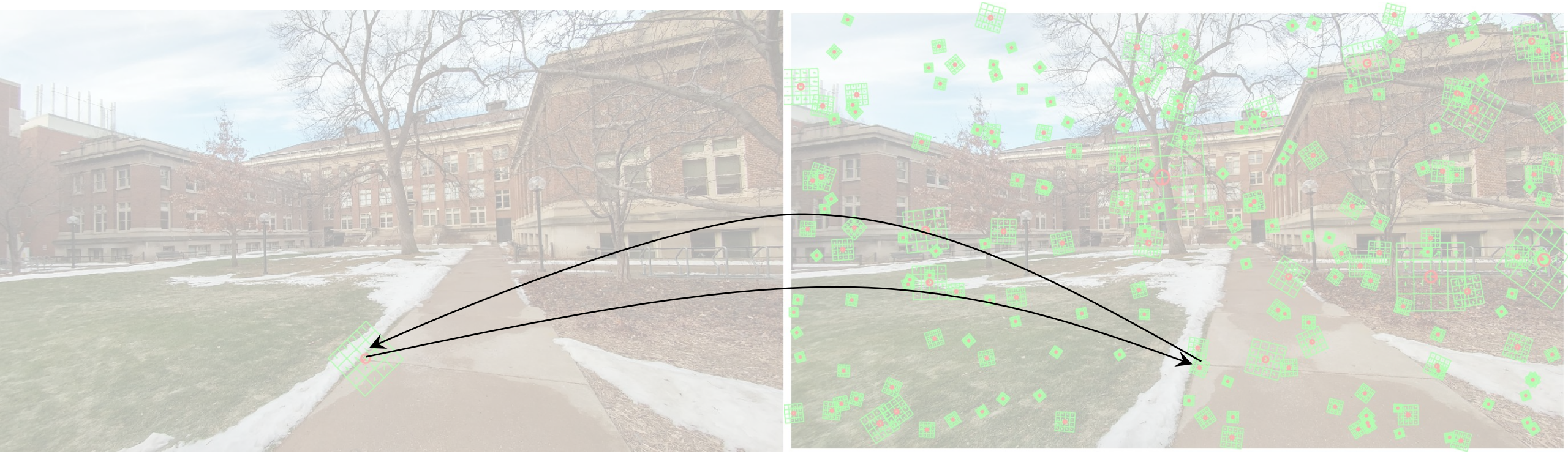
Left image←right image

Nearest Neighbor Search w/ Ratio Test



Left image←right image

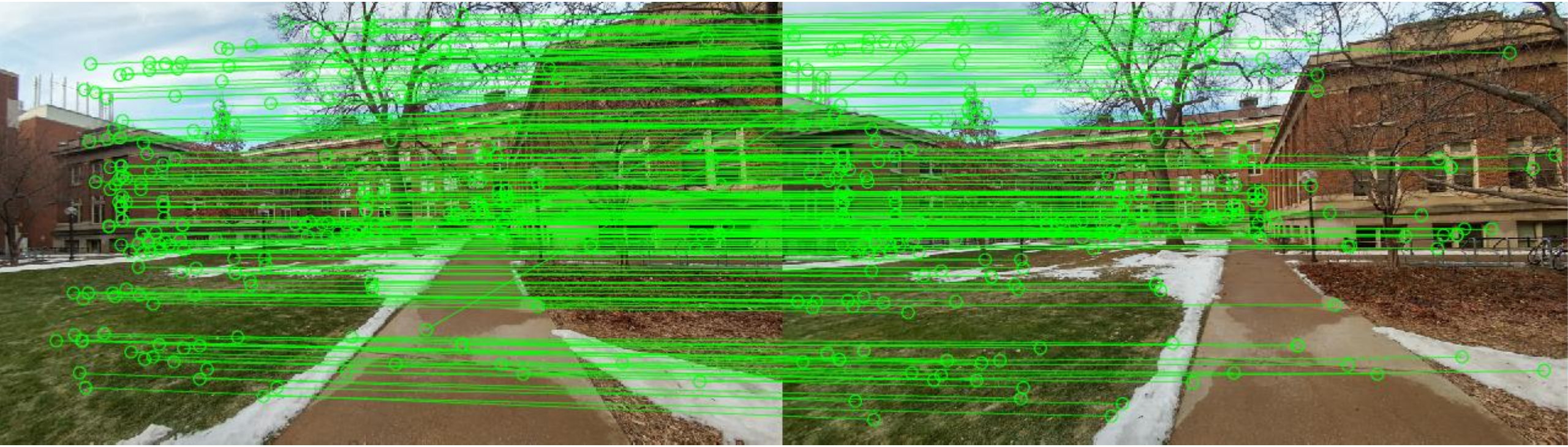
Bi-directional Consistency Check



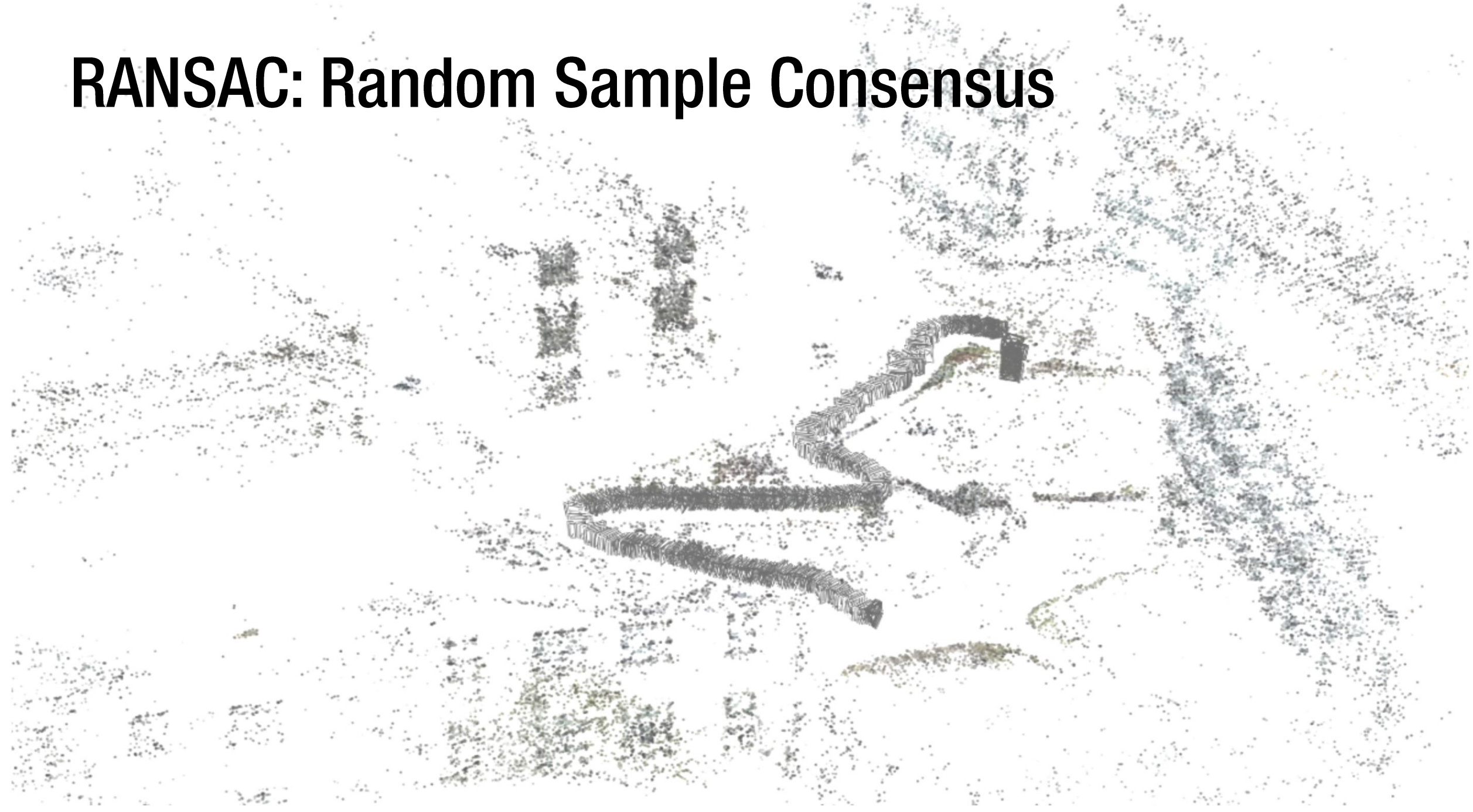
Feature match candidates

Consistency: would a feature match correspond to each other?

Bi-directional Consistency Check

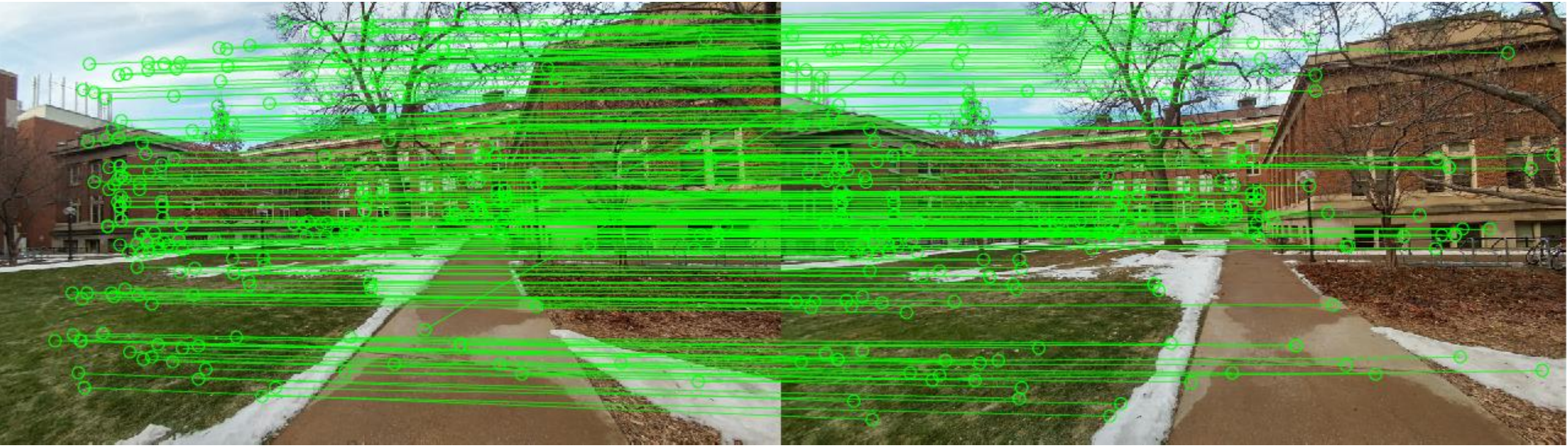


RANSAC: Random Sample Consensus



Fundamental Matrix Computation: Linear Least Squares

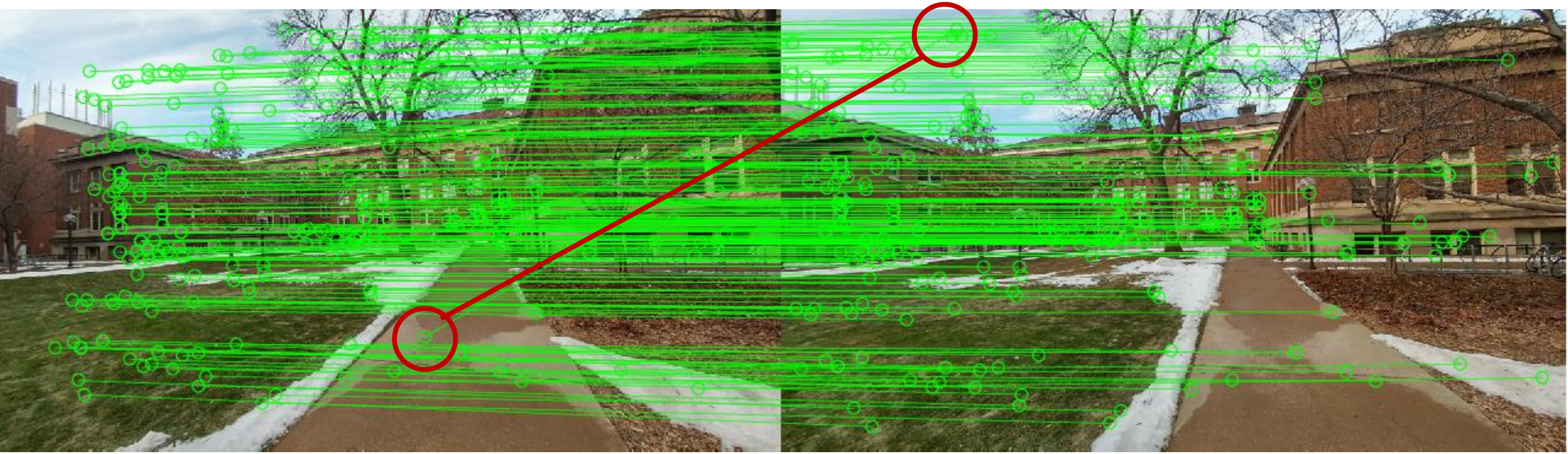
$$\begin{bmatrix} u_1^x v_1^x & u_1^y v_1^x & v_1^x & u_1^x v_1^y & u_1^y v_1^y & v_1^y & u_1^x & u_1^y & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_m^x v_m^x & u_m^y v_m^x & v_m^x & u_m^x v_m^y & u_m^y v_m^y & v_m^y & u_m^x & u_m^y & 1 \end{bmatrix} \mathbf{A} \quad \mathbf{X} \quad \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$



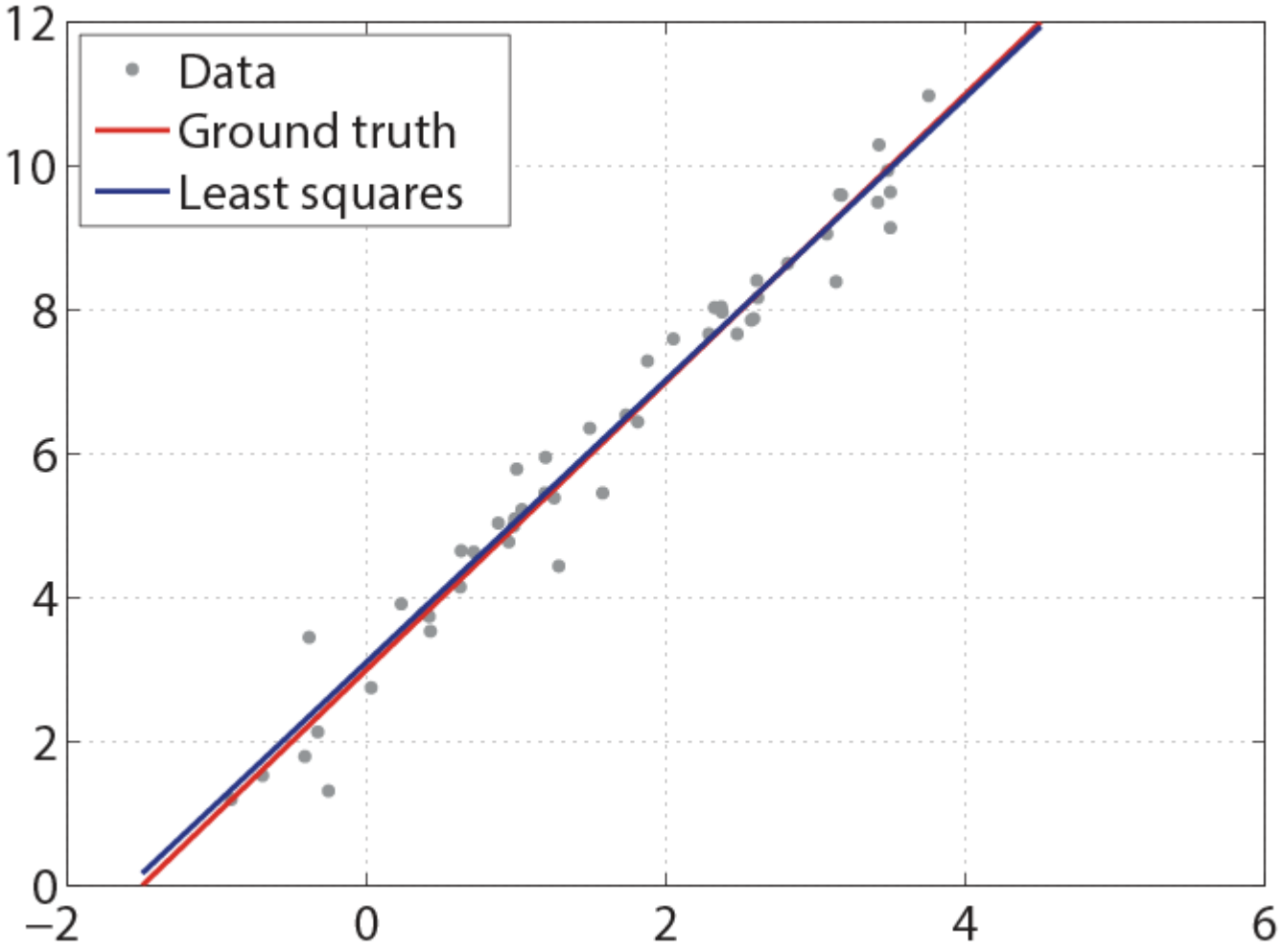
Fundamental Matrix Computation: Linear Least Squares

$$\begin{bmatrix} u_1^x v_1^x & u_1^y v_1^x & v_1^x & u_1^x v_1^y & u_1^y v_1^y & v_1^y & u_1^x & u_1^y & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_m^x v_m^x & u_m^y v_m^x & v_m^x & u_m^x v_m^y & u_m^y v_m^y & v_m^y & u_m^x & u_m^y & 1 \end{bmatrix} \mathbf{A} \quad \mathbf{x} \quad \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Outlier?



Recall: Line Fitting (Ax=b)



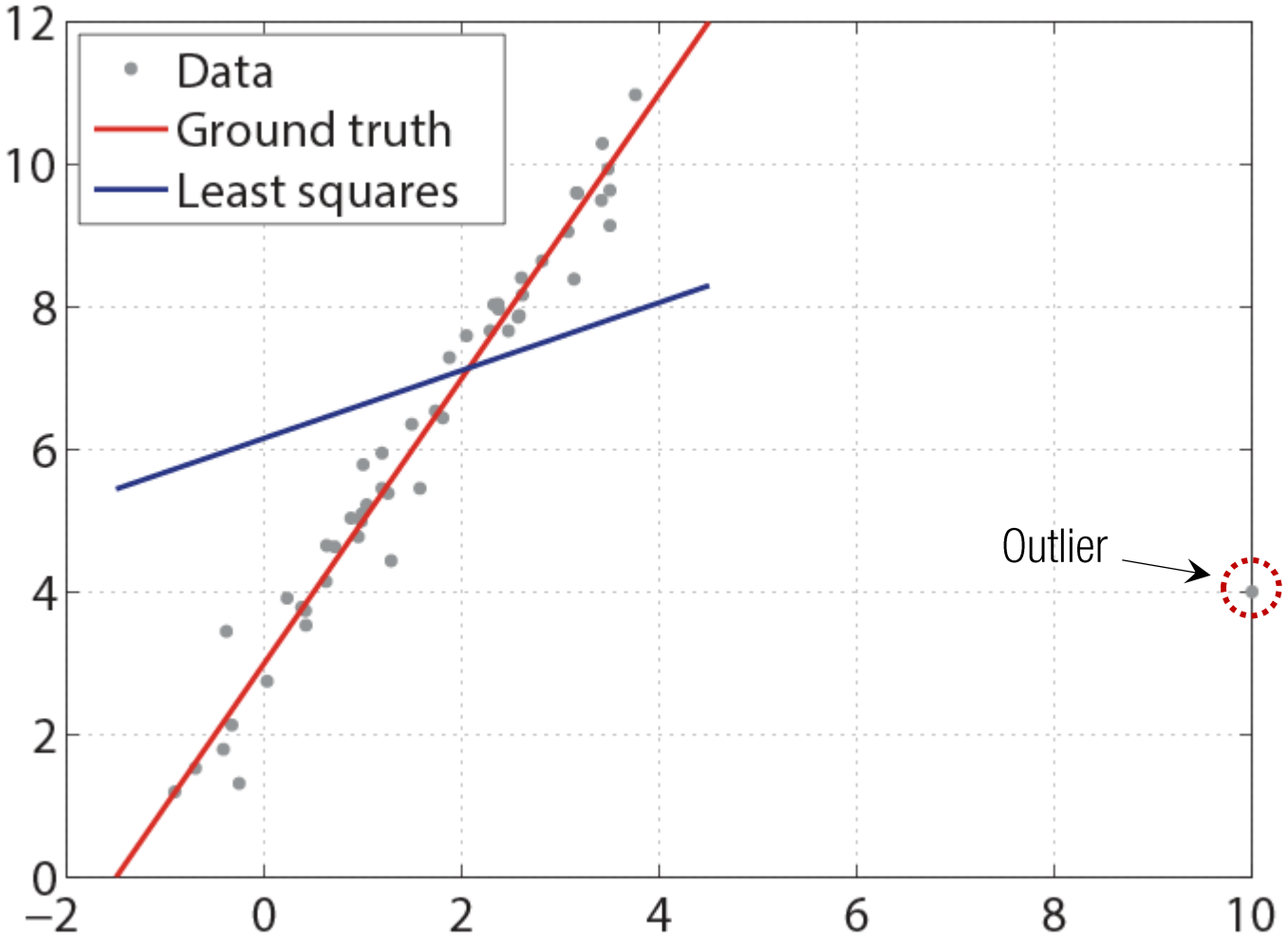
$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

$$\mathbf{A} \quad \mathbf{x} \approx \mathbf{b}$$

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$$

$$\mathbf{x} = \left[\mathbf{A}^T \mathbf{A} \right]^{-1} \mathbf{A}^T \mathbf{b}$$

Outlier

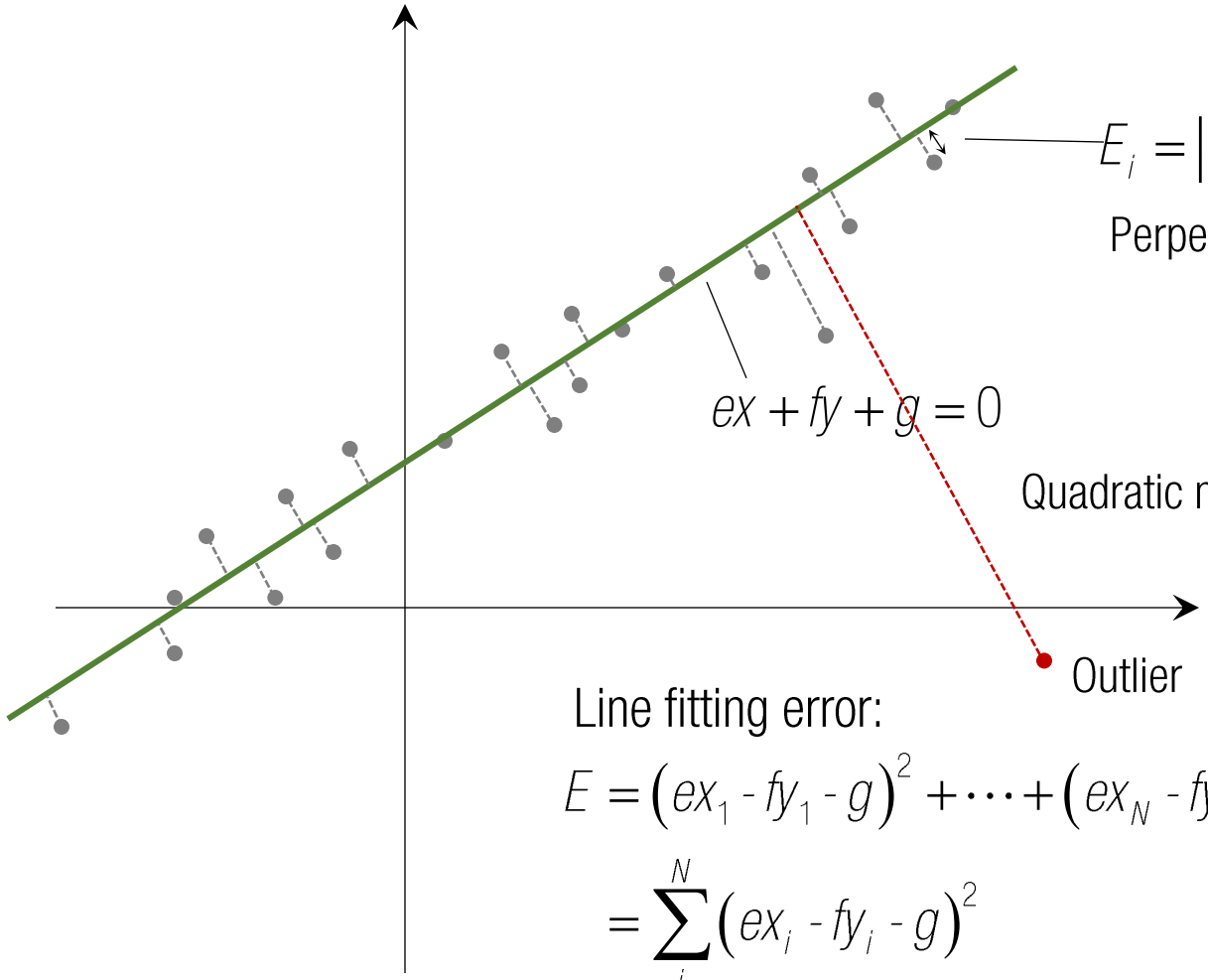


$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

$$\mathbf{A} \quad \mathbf{x} \approx \mathbf{b}$$

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$$

$$\mathbf{x} = \left[\mathbf{A}^T \mathbf{A} \right]^{-1} \mathbf{A}^T \mathbf{b}$$



$E_i = |ex_i + fy_i + g|$

Perpendicular distance

$ex + fy + g = 0$

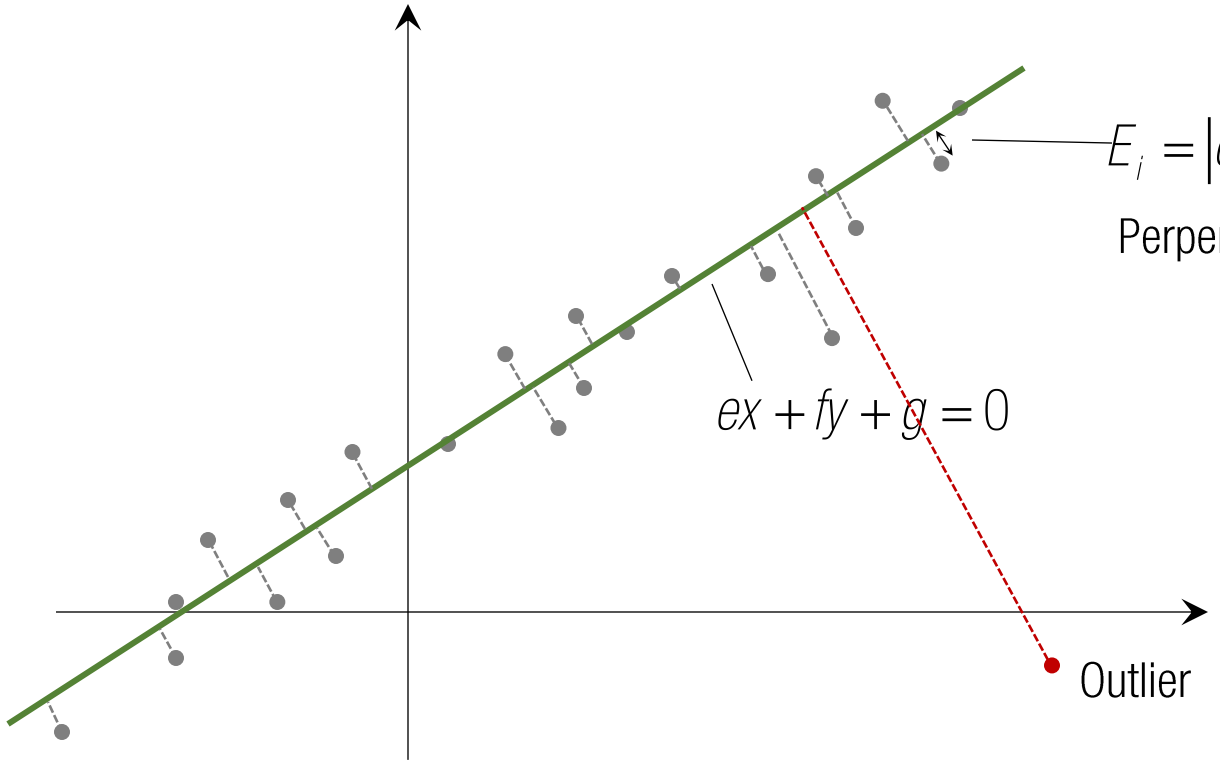
Quadratic magnification of error of outliers

Outlier

Line fitting error:

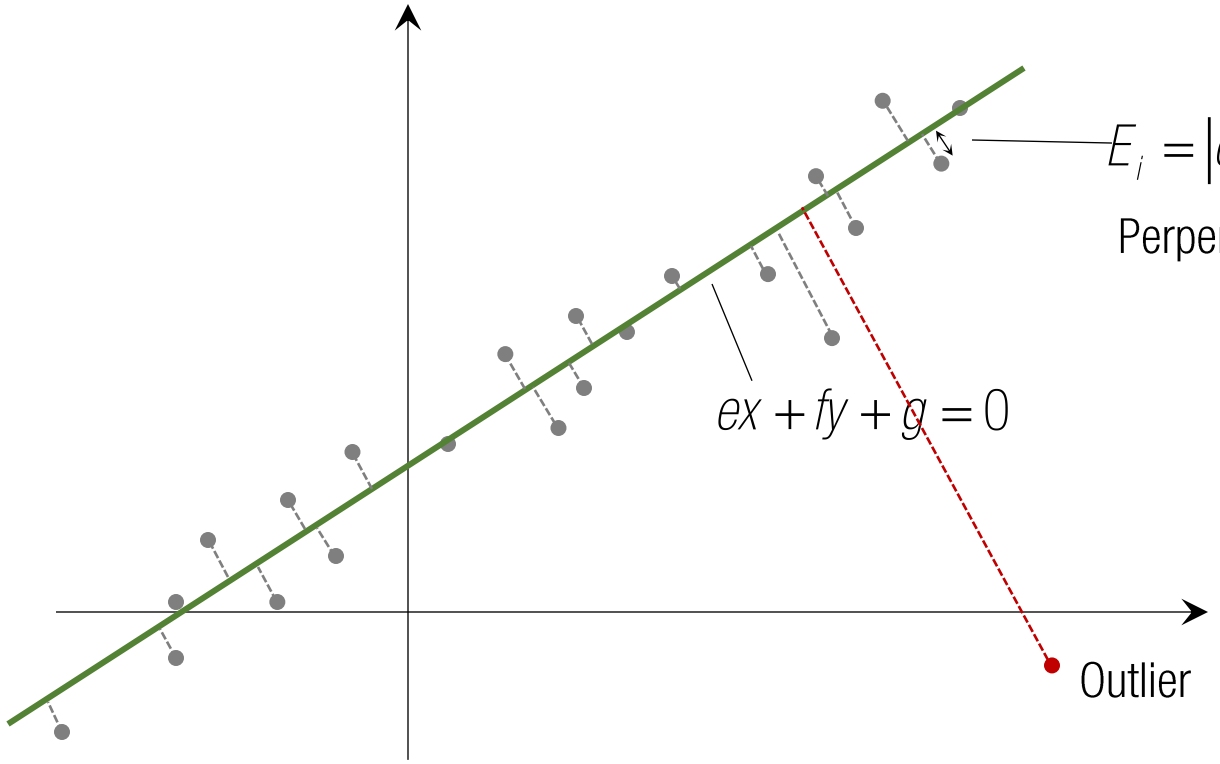
$$E = (ex_1 - fy_1 - g)^2 + \cdots + (ex_N - fy_N - g)^2$$

$$= \sum_i^N (ex_i - fy_i - g)^2$$



Outlier rejection strategy:

To find the best line that explanes the maximum number of points.



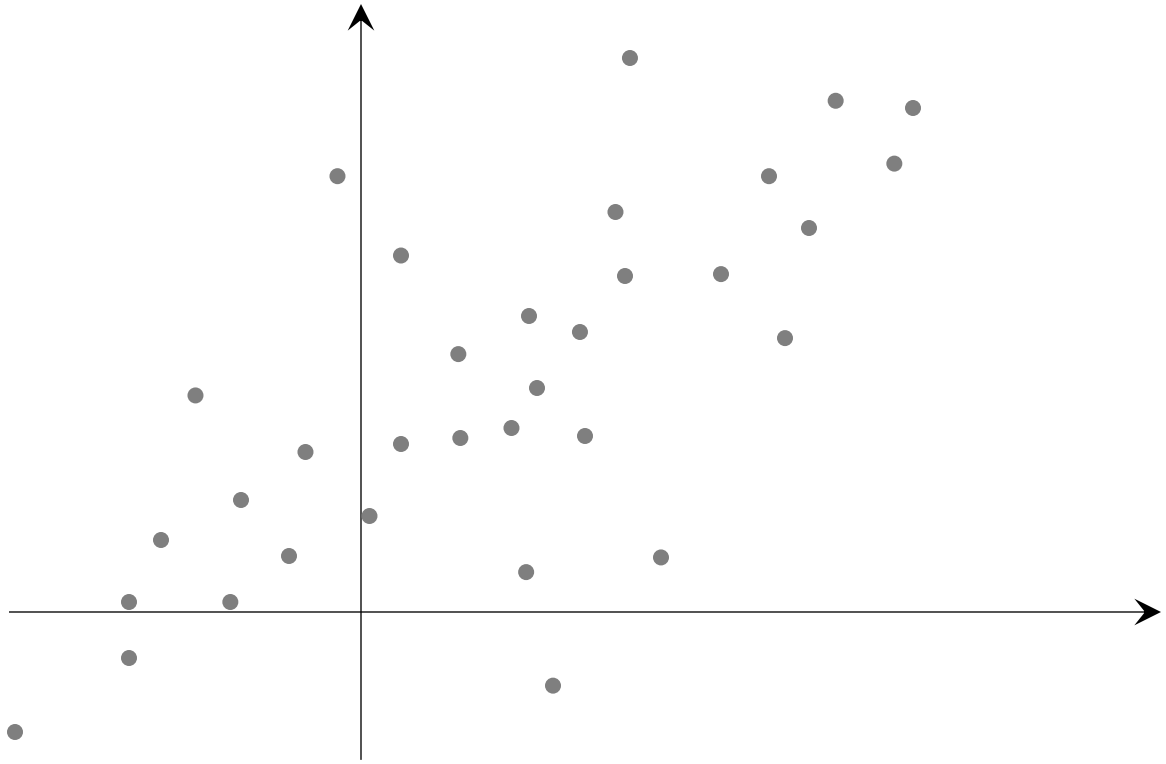
Outlier rejection strategy:

To find the best line that explanes the maximum number of points.

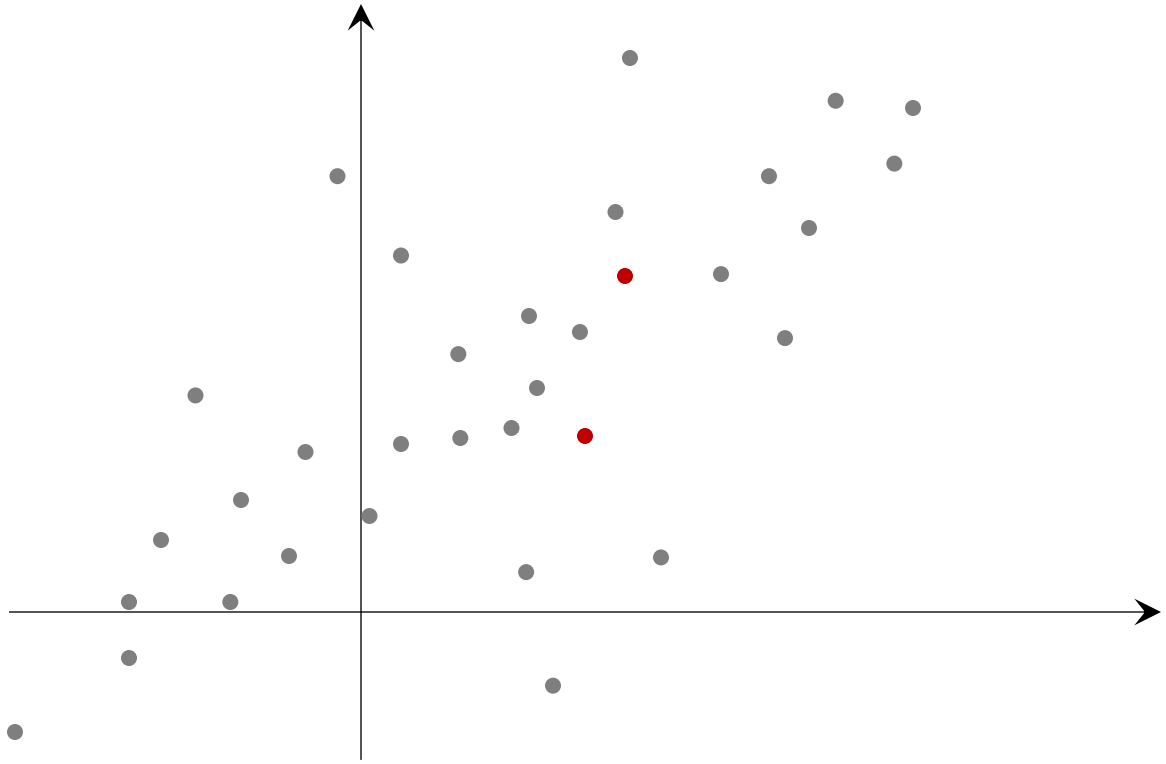
Assumptions:

1. Majority of good samples agree with the underlying model (good apples are same and simple.).

2. Bad samples does not consistently agree with a single model (all bad apples are different and complicated.).

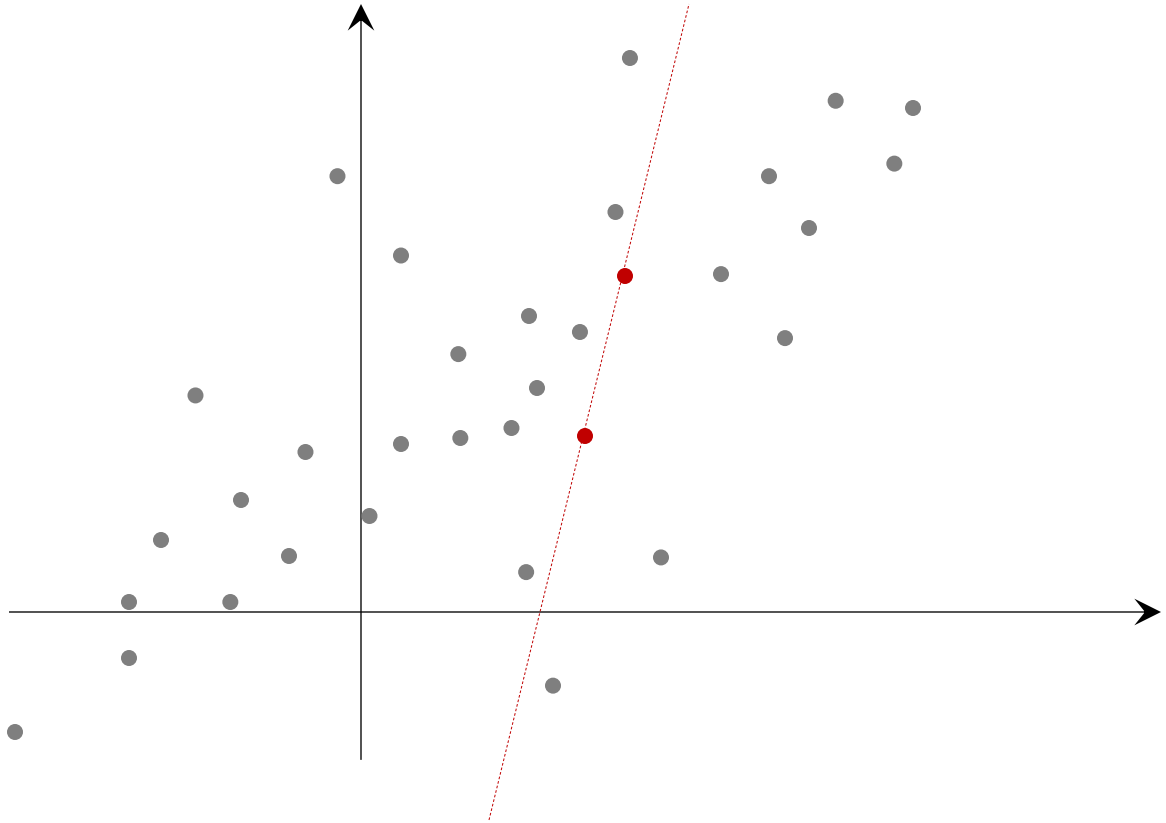


RANSAC: Random Sample Consensus



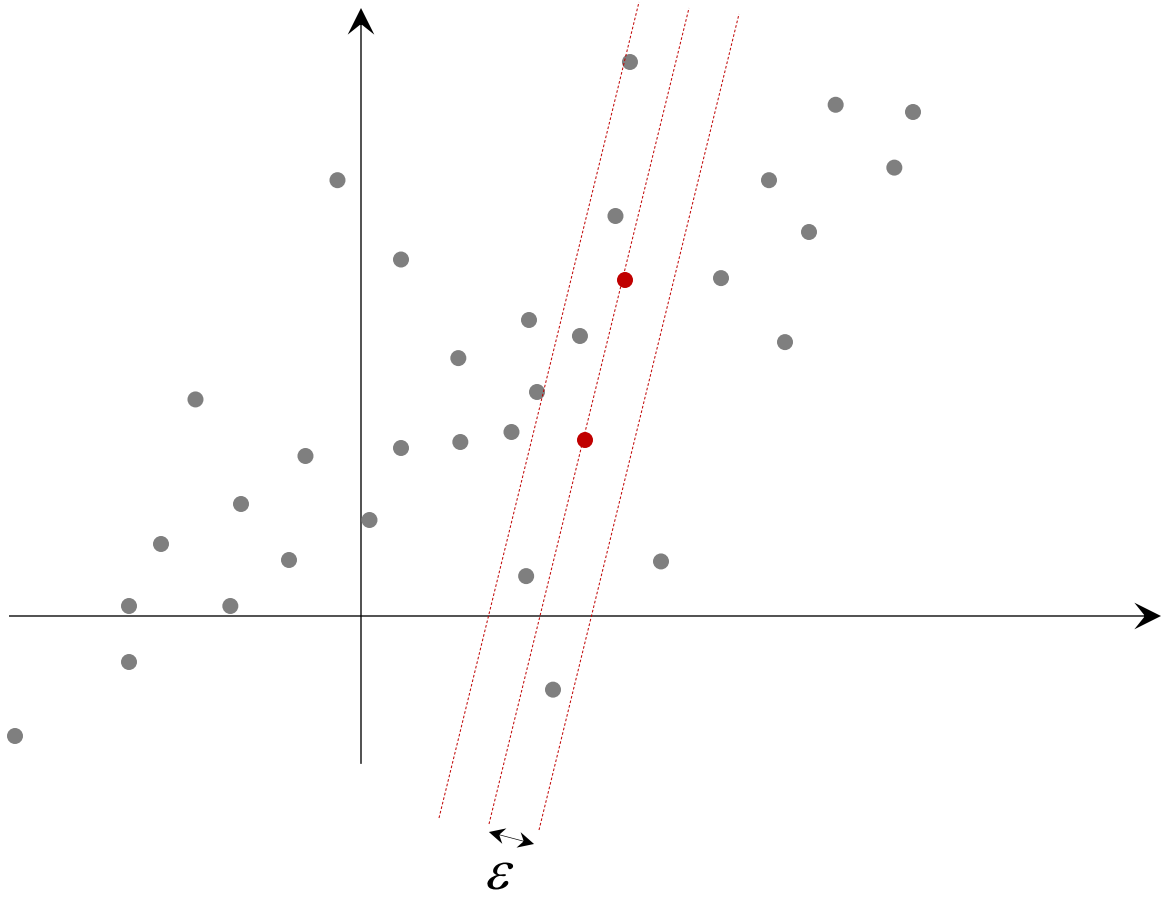
1. Random sampling

RANSAC: Random Sample Consensus



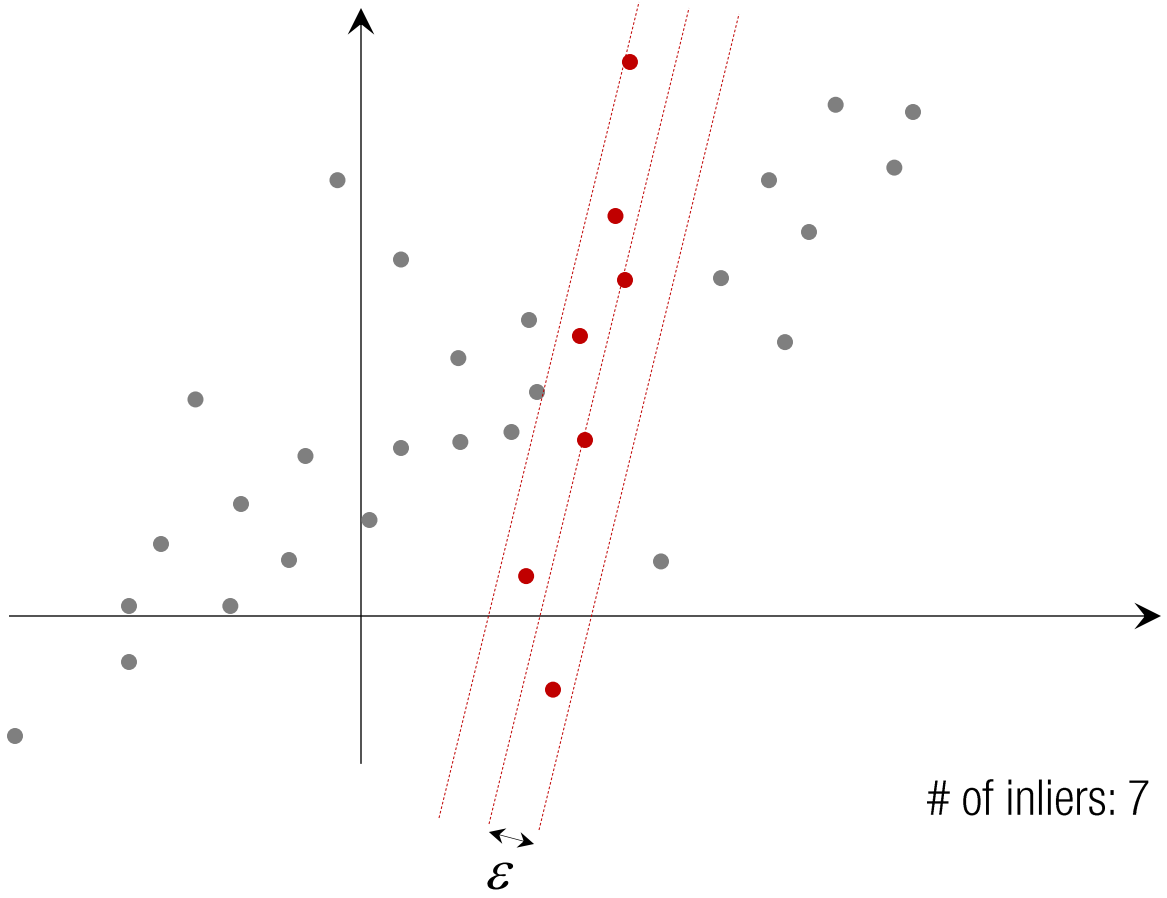
1. Random sampling
2. Model building

RANSAC: Random Sample Consensus



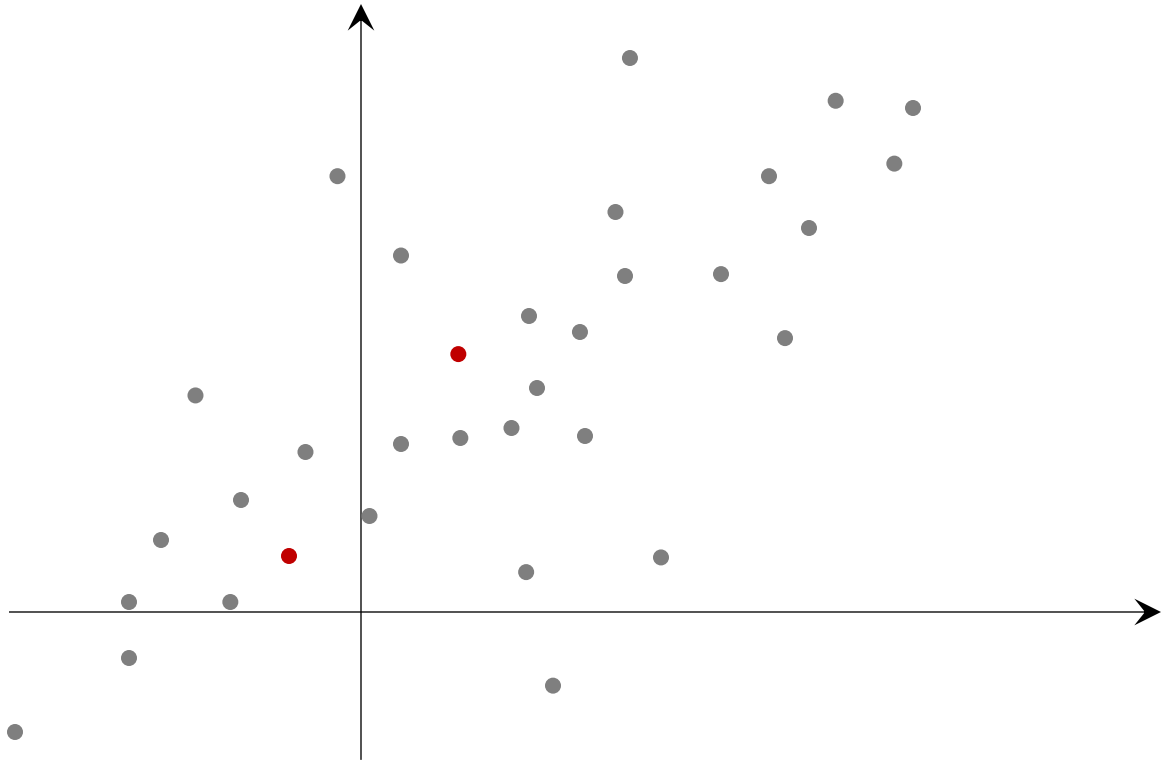
1. Random sampling
2. Model building
3. Thresholding

RANSAC: Random Sample Consensus



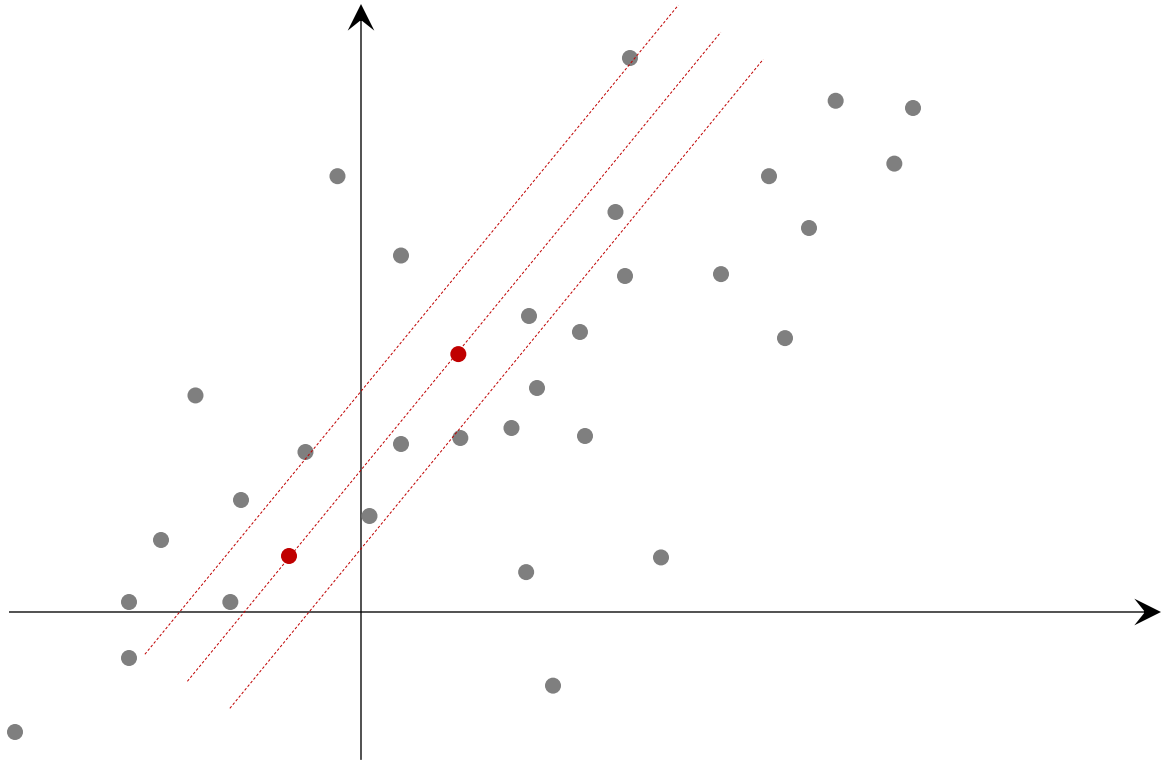
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



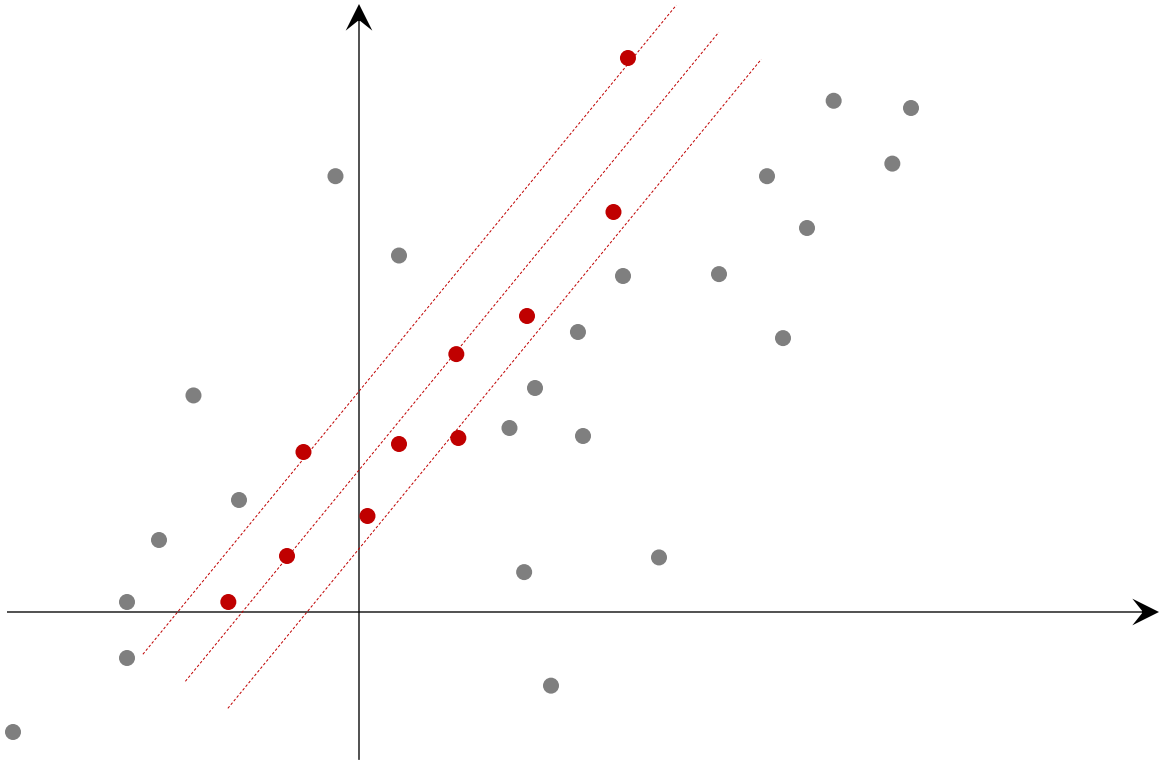
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



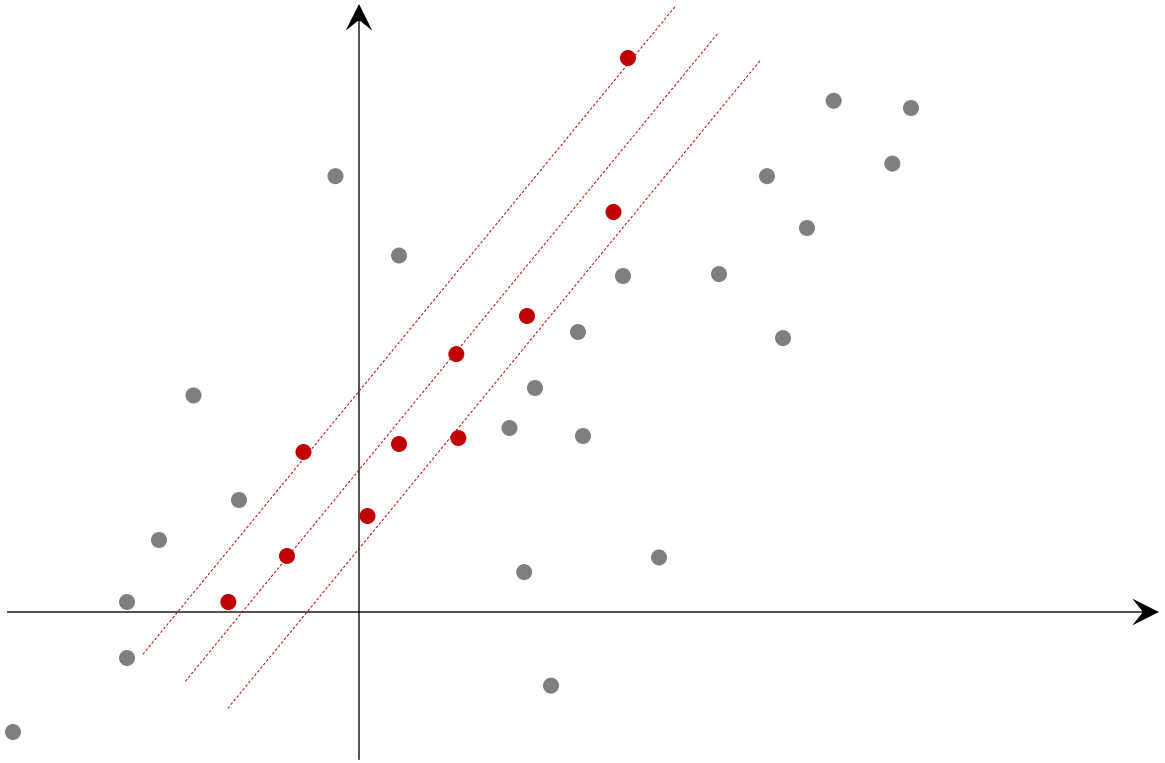
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

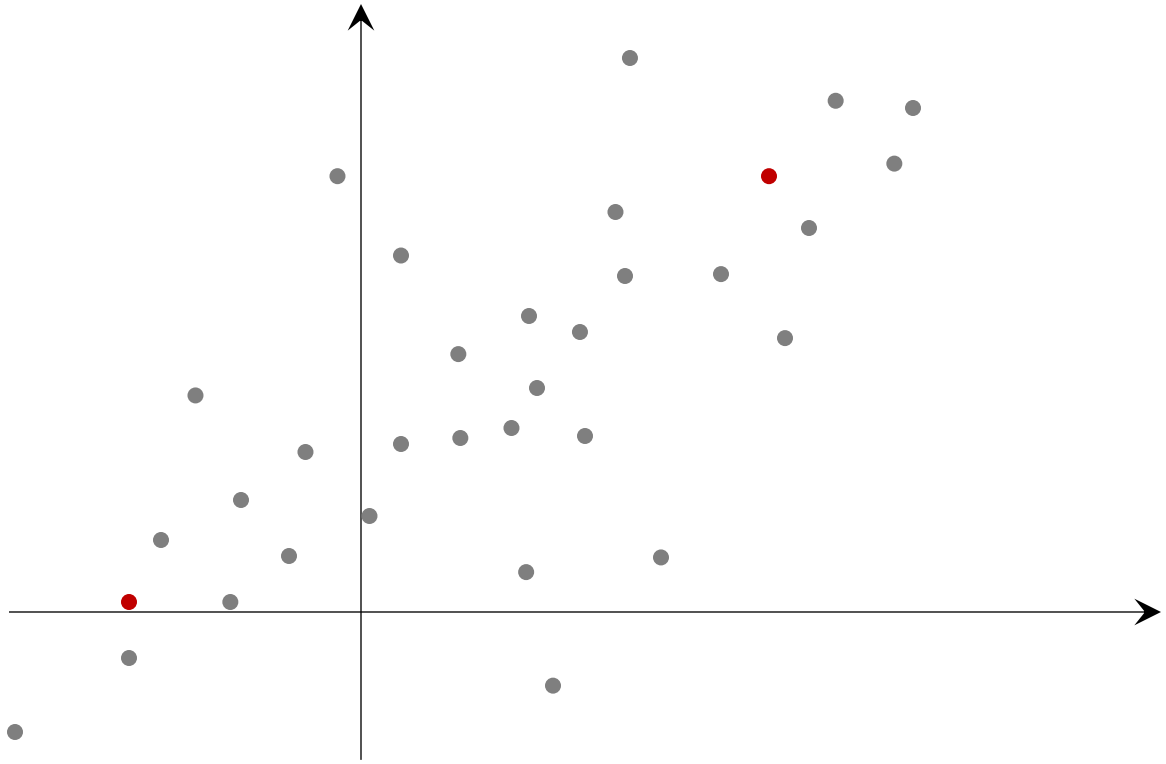
RANSAC: Random Sample Consensus



# of inliers: 10

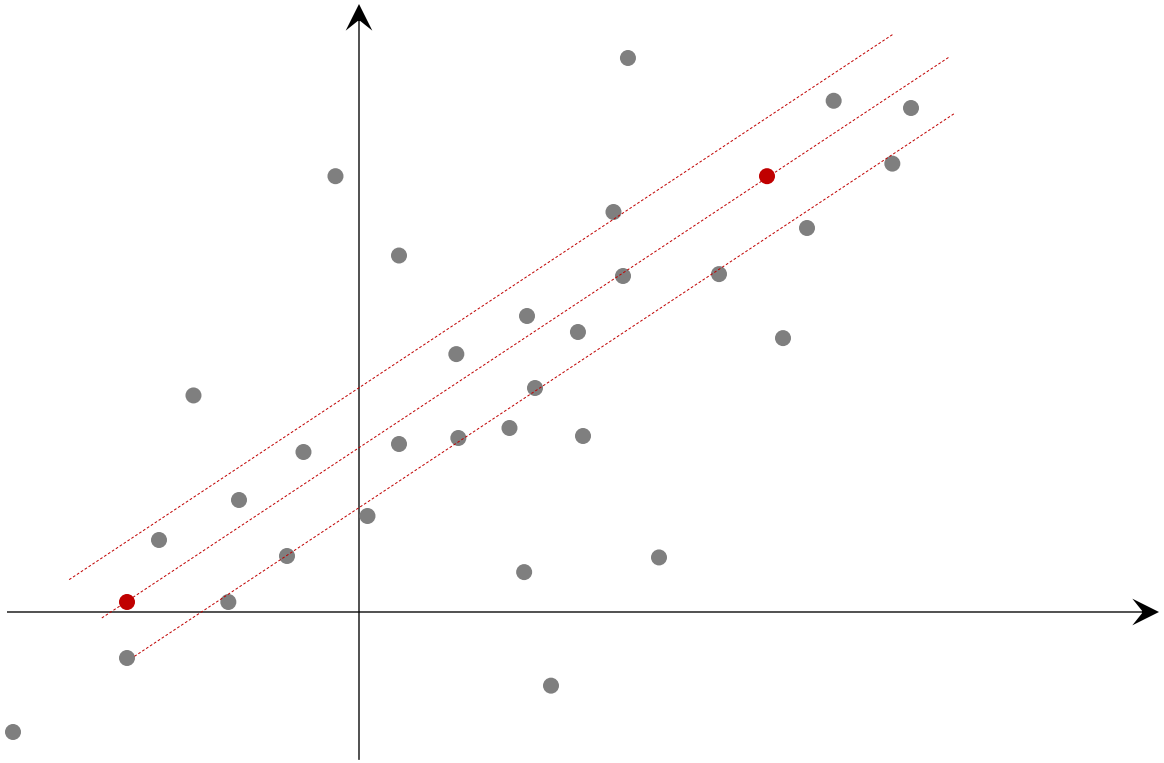
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



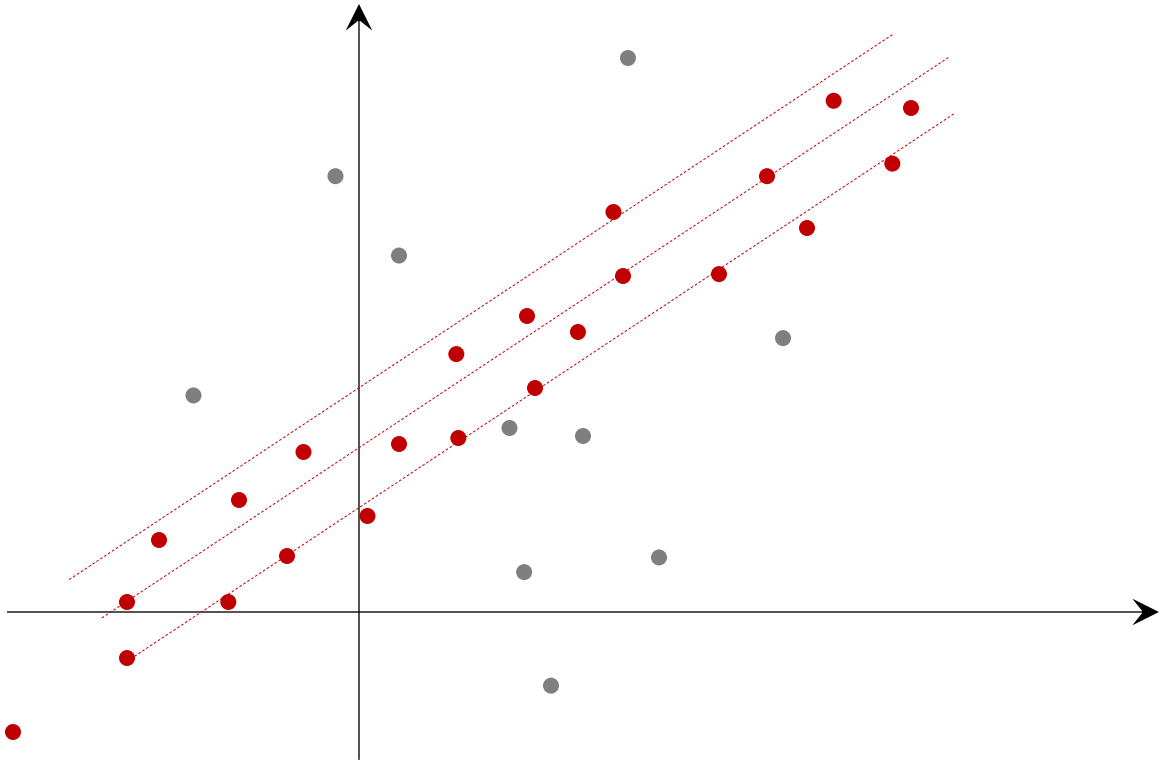
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



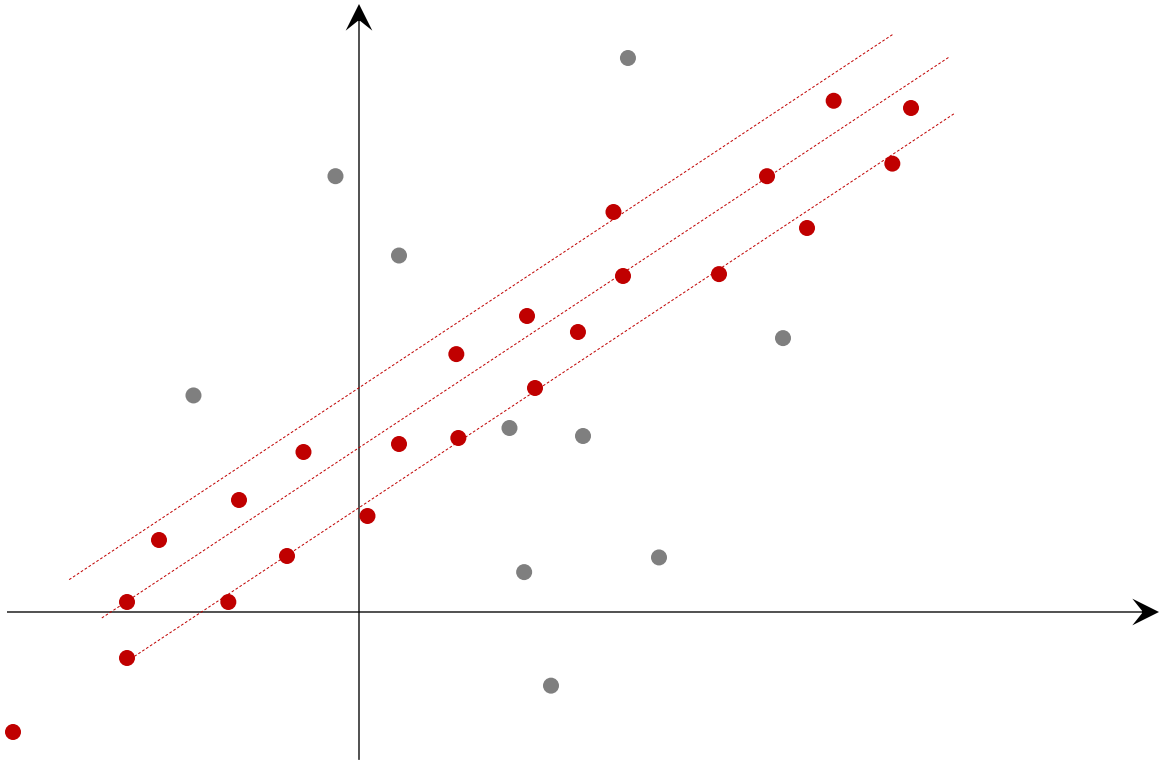
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

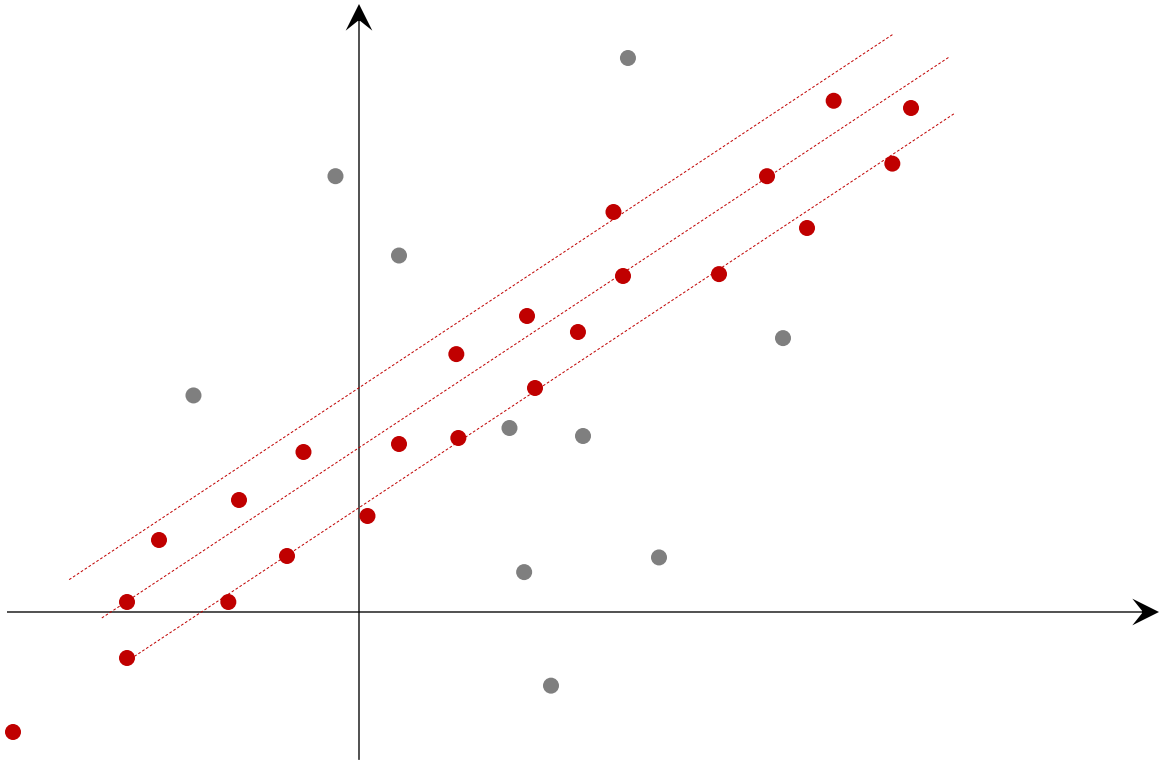
RANSAC: Random Sample Consensus



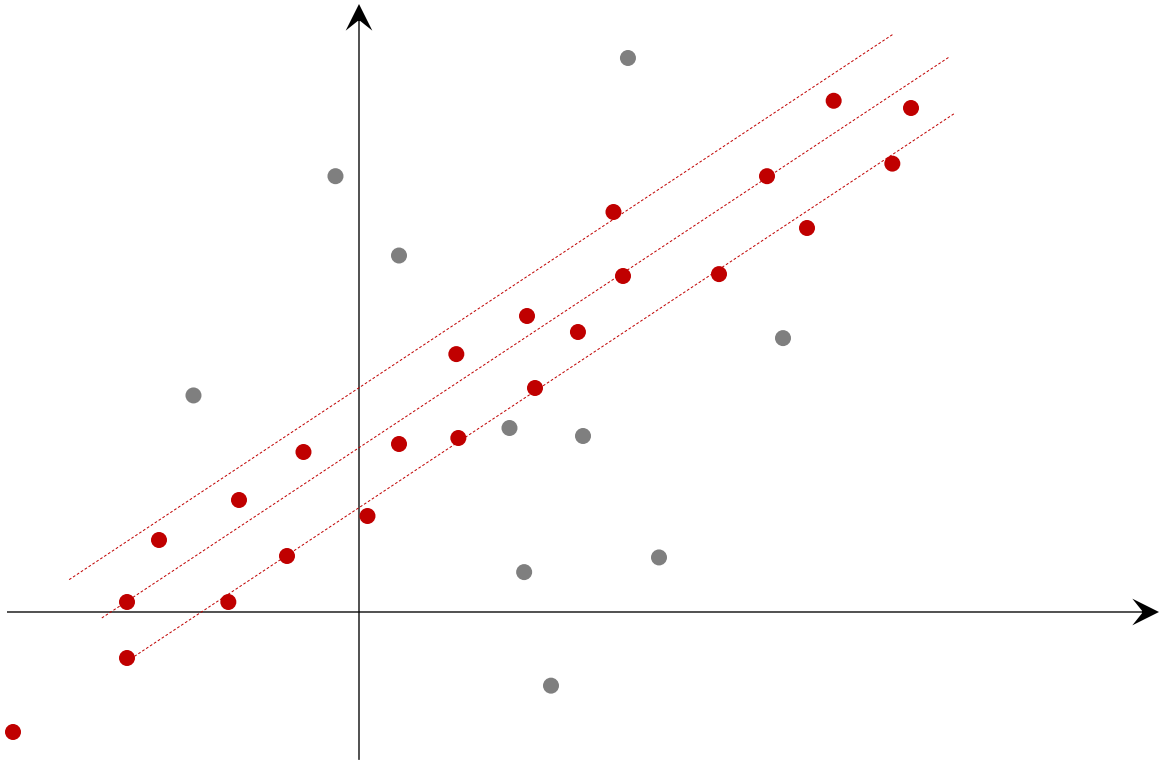
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

$$\frac{\text{\# of inliers: 23}}{\text{Maximum number of inliers}}$$

RANSAC: Random Sample Consensus



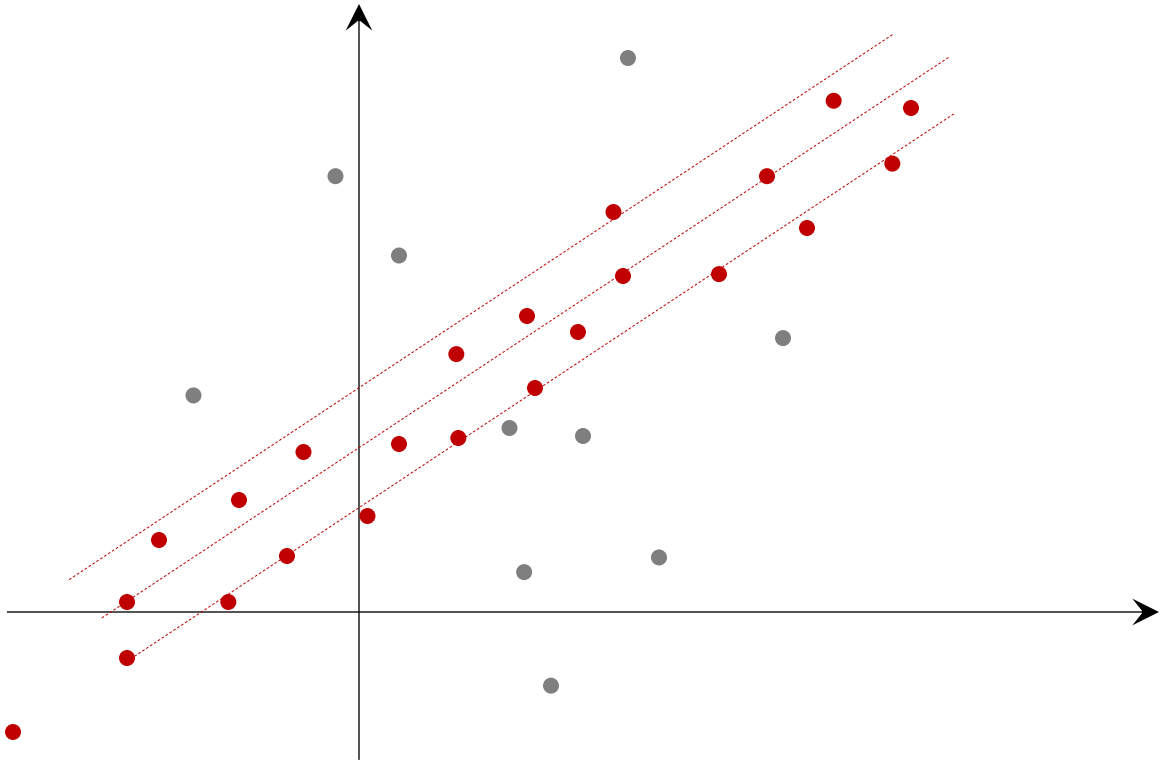
Required number of iterations with p success rate:



Required number of iterations with *p* success rate:

Probability of choosing an inlier:

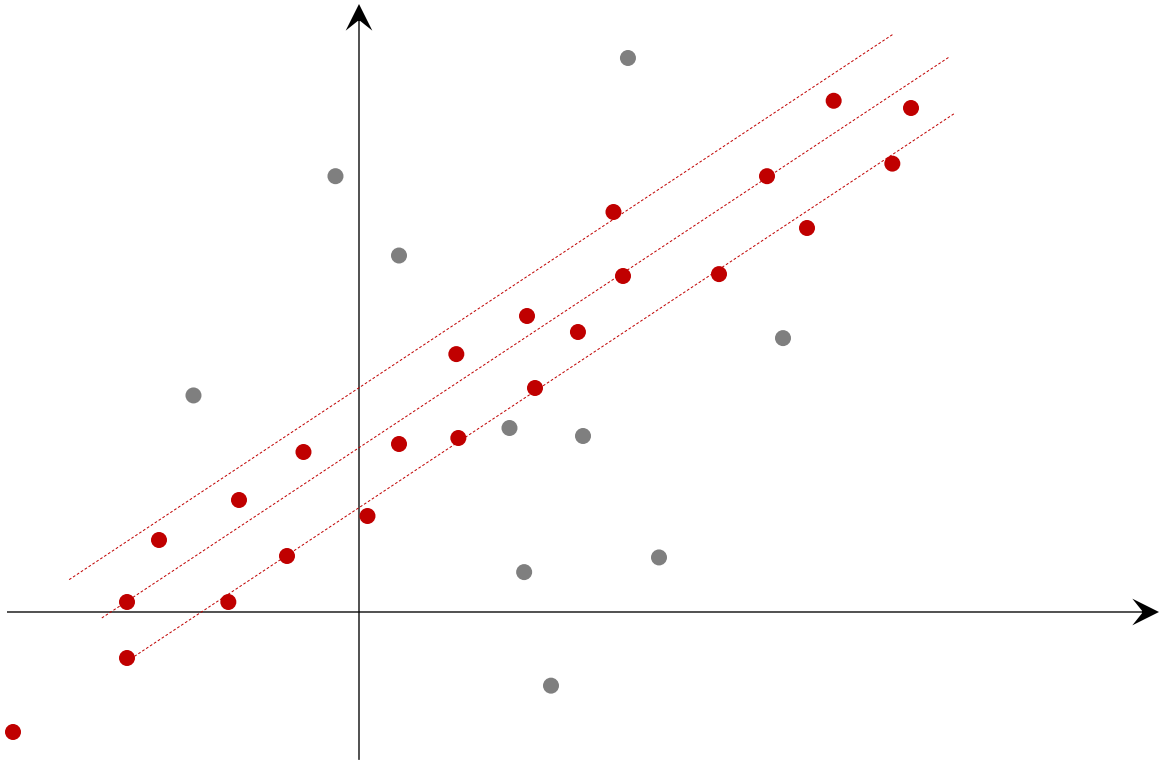
$$w = \frac{\#\,\text{of inliers}}{\#\,\text{of samples}}$$



Required number of iterations with *p* success rate:

Probability of choosing an inlier: $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$

Probability of building a correct model: w^n where n is the number of samples to build a model.

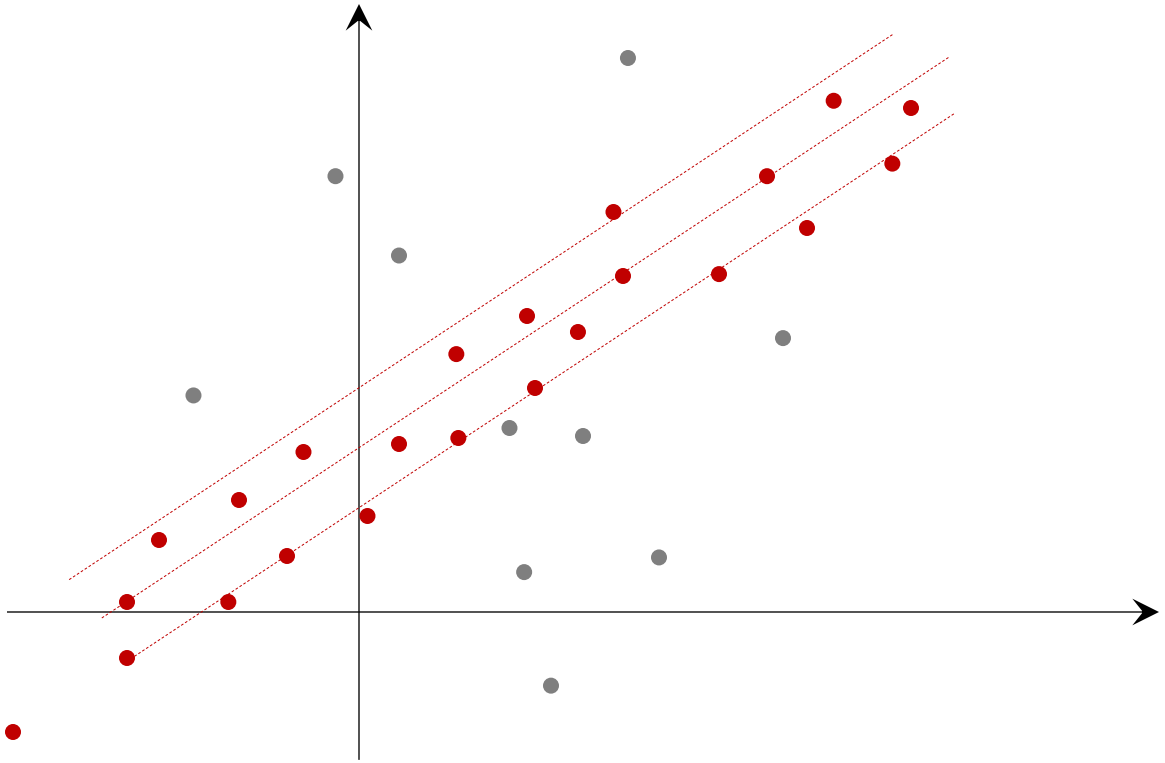


Required number of iterations with *p* success rate:

Probability of choosing an inlier: $$w = \frac{\#\,\text{of inliers}}{\#\,\text{of samples}}$$

Probability of building a correct model: w^n where n is the number of samples to build a model.

Probability of not building a correct model during *k* iterations: $\left(1 - w^n\right)^k$



Required number of iterations with *p* success rate:

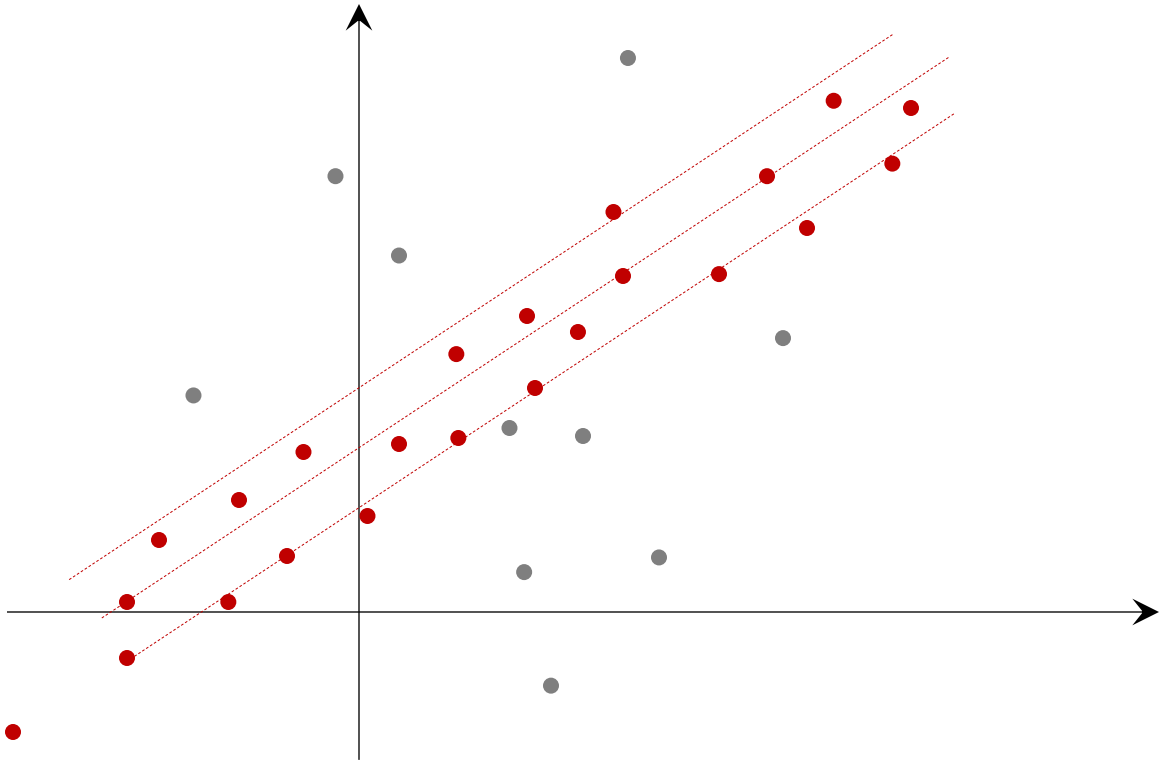
Probability of choosing an inlier: $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$

Probability of building a correct model: w^n where n is the number of samples to build a model.

Probability of not building a correct model during *k* iterations: $\left(1-w^n\right)^k$

$\left(1-w^n\right)^k = 1-p$ where *p* is desired RANSAC success rate.

$$k = \frac{\log\left(1-p\right)}{\log\left(1-w^n\right)}$$



Required number of iterations with *p* success rate:

$$k = \frac{\log(1-p)}{\log(1-w^n)} \qquad \text{where} \quad w = \frac{\#\,\text{of inliers}}{\#\,\text{of samples}}$$

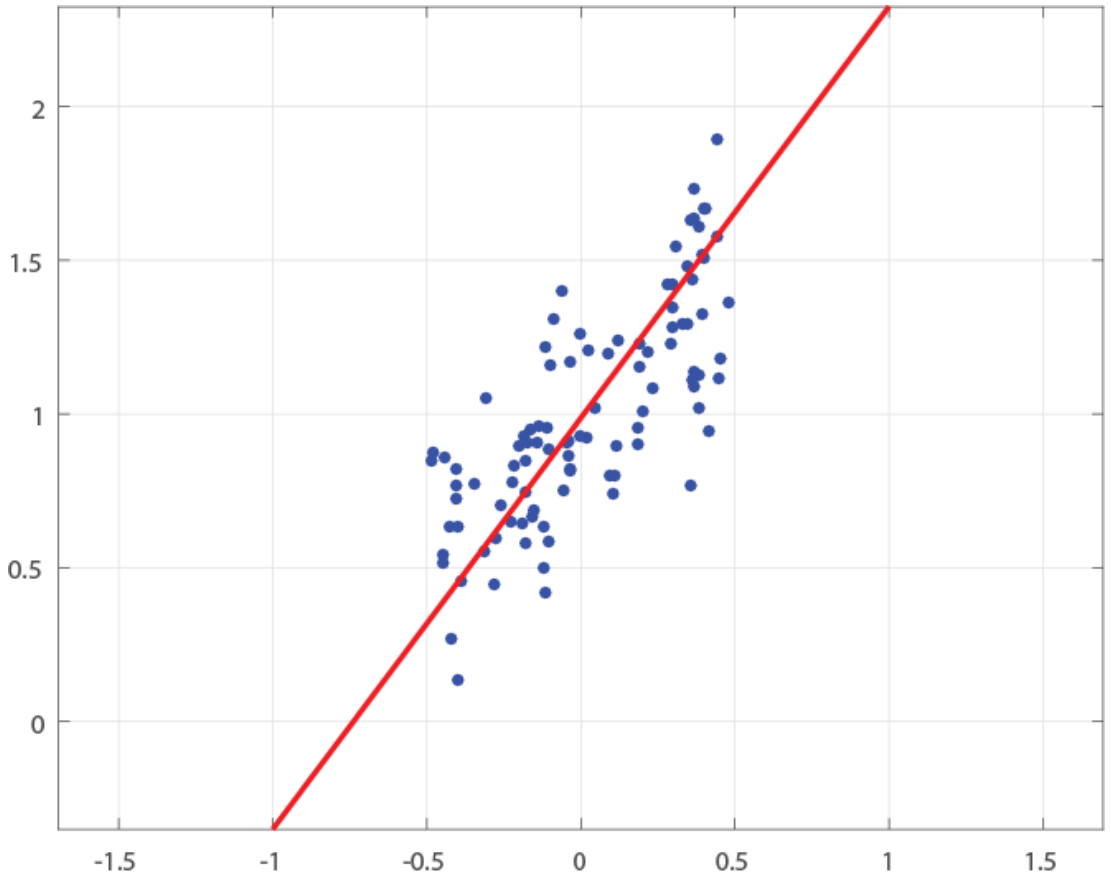
Probability of choosing an inlier: $w = \dfrac{\#\,\text{of inliers}}{\#\,\text{of samples}}$

Probability of building a correct model: w^n where n is the number of samples to build a model.

Probability of not building a correct model during *k* iterations: $\left(1-w^n\right)^k$

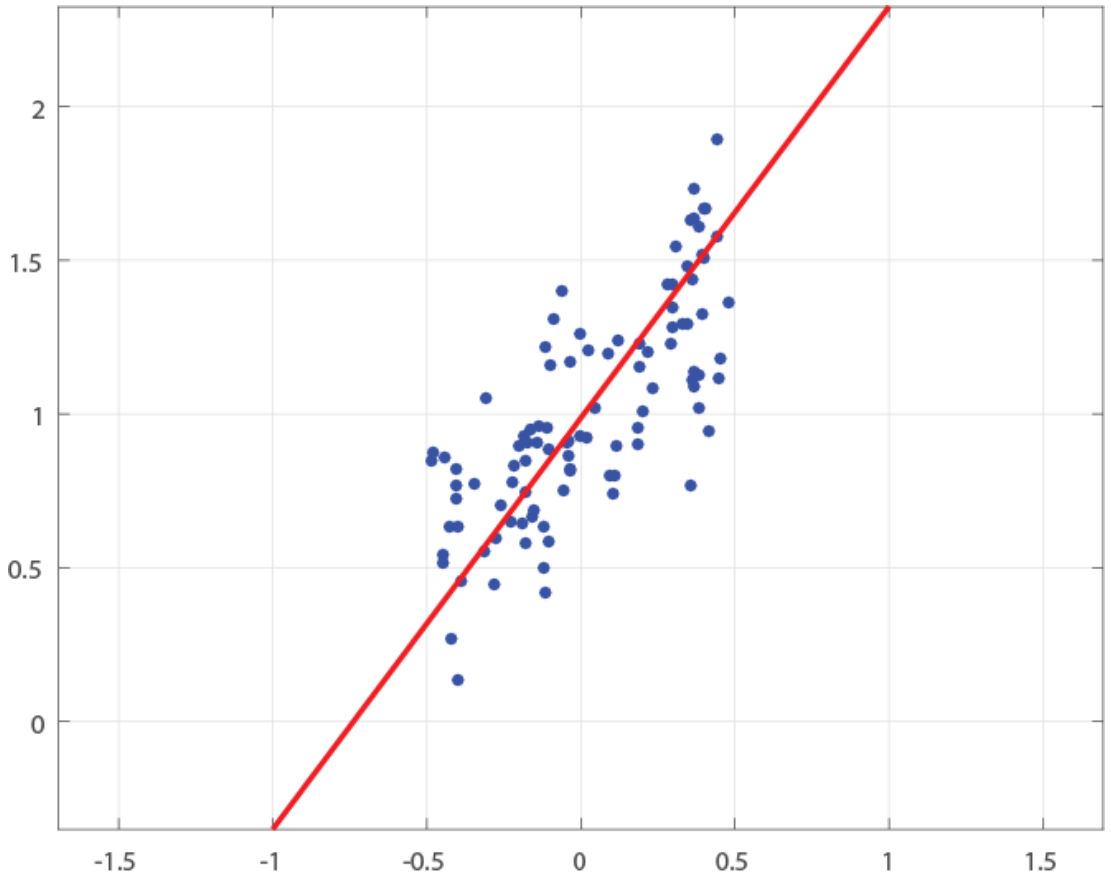
$\left(1-w^n\right)^k = 1-p$ where *p* is desired RANSAC success rate.

$$k = \frac{\log(1-p)}{\log(1-w^n)}$$




```
function RANSAC_line

a = 1;
b = 1;

nPoints = 100;
x = rand(nPoints,1)-0.5;
y = a*x + b + 0.2*randn(nPoints,1);
```

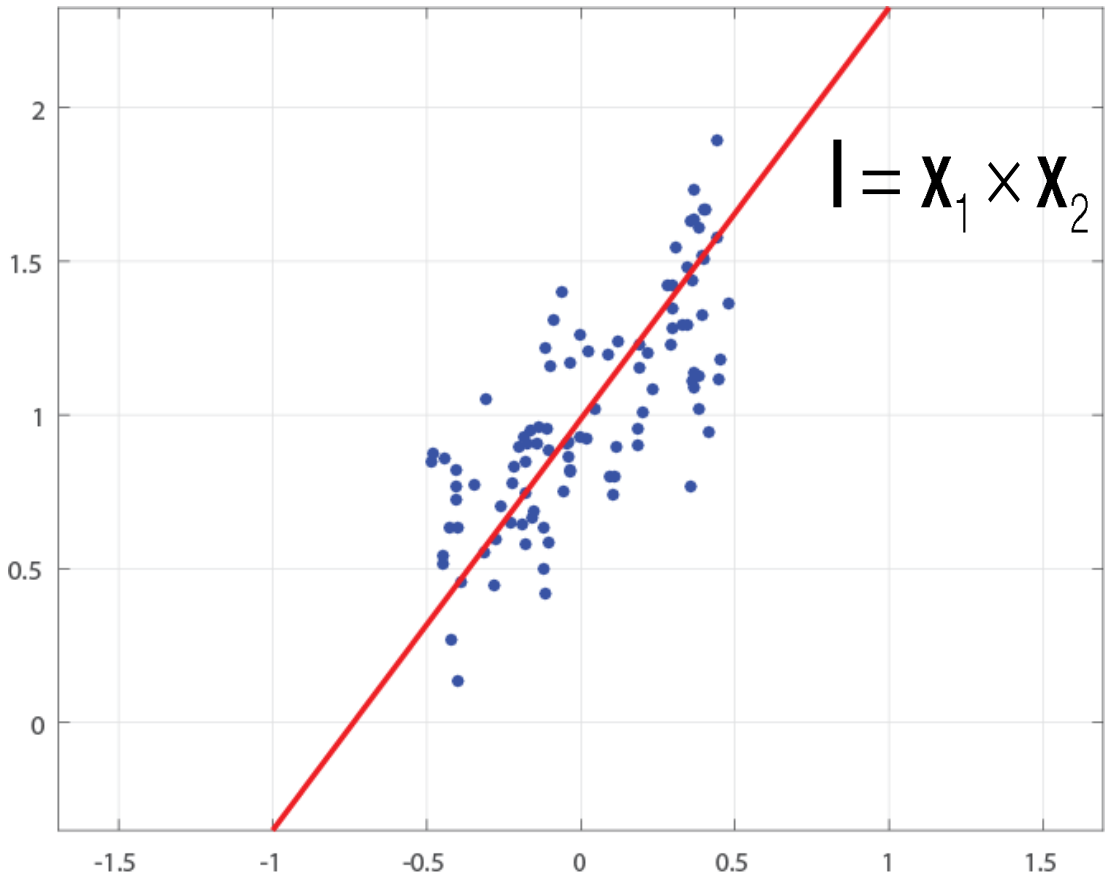
Generating samples




```
function RANSAC_line

a = 1;
b = 1;

nPoints = 100;
x = rand(nPoints,1)-0.5;
y = a*x + b + 0.2*randn(nPoints,1);

nRansacIter = 500;
threshold = 0.1;
max_nInliers = 0;
for i = 1 : nRansacIter
    r = randperm(nPoints);
    s1 = [x(r(1)); y(r(1)); 1];
    s2 = [x(r(2)); y(r(2)); 1];
```

Generating samples

Random sampling

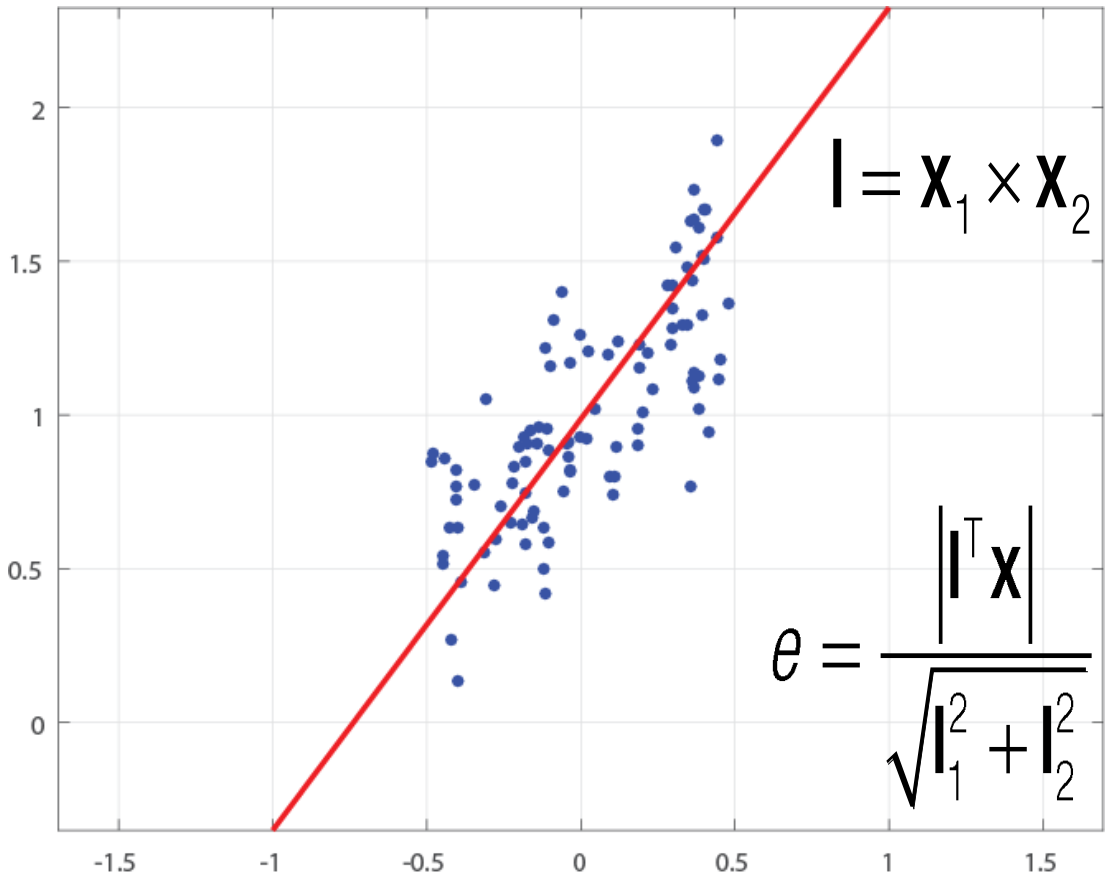



```
function RANSAC_line

a = 1;
b = 1;

nPoints = 100;
x = rand(nPoints,1)-0.5;
y = a*x + b + 0.2*randn(nPoints,1);

nRansacIter = 500;
threshold = 0.1;
max_nInliers = 0;
for i = 1 : nRansacIter
    r = randperm(nPoints);
    s1 = [x(r(1)); y(r(1)); 1];
    s2 = [x(r(2)); y(r(2)); 1];

    l = GetLineFromTwoPoints(s1, s2);
```

Generating samples

Random sampling

Model building



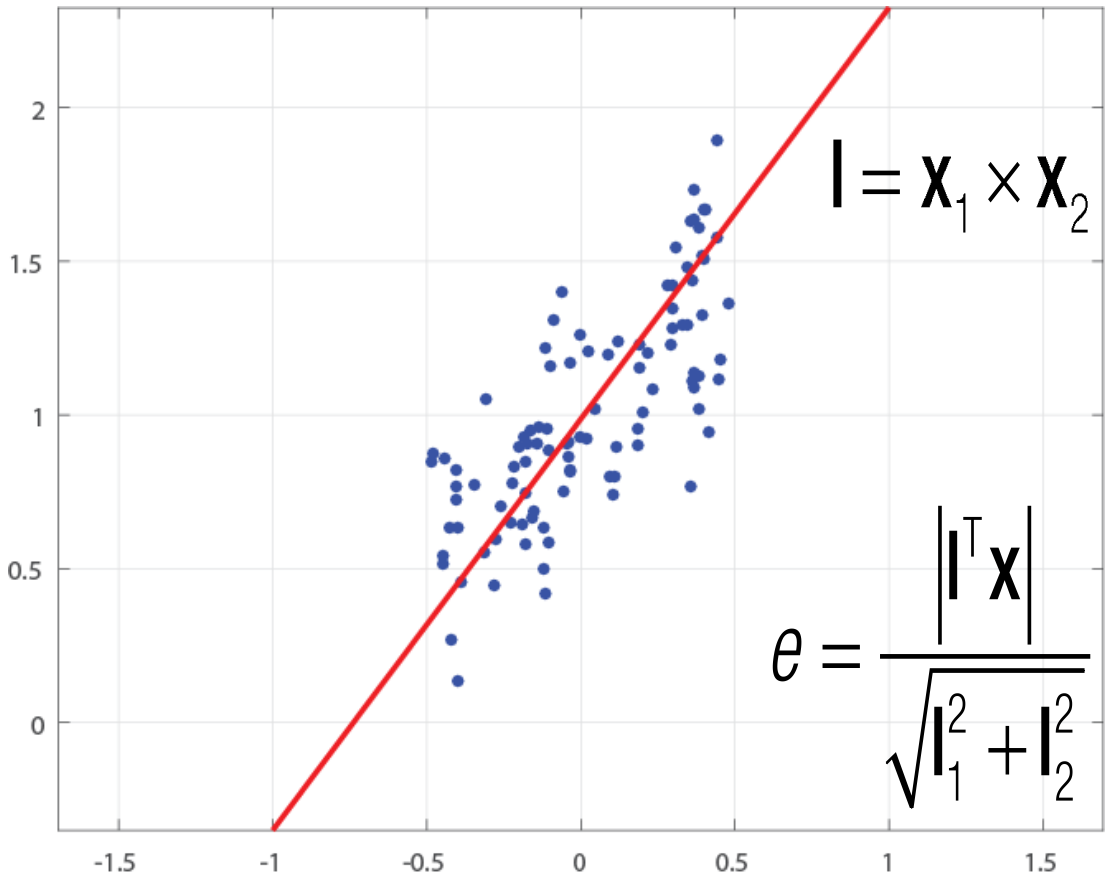

```
function RANSAC_line

a = 1;
b = 1;

nPoints = 100;
x = rand(nPoints,1)-0.5;
y = a*x + b + 0.2*randn(nPoints,1);

nRansacIter = 500;
threshold = 0.1;
max_nInliers = 0;
for i = 1 : nRansacIter
   r = randperm(nPoints);
   s1 = [x(r(1)); y(r(1)); 1];
   s2 = [x(r(2)); y(r(2)); 1];

   l = GetLineFromTwoPoints(s1, s2);

   nInliers = 0;
   for j = 1 : nPoints
      e = abs(l' * [x(j); y(j); 1])/norm(l(1:2));
      if (e < threshold)
         nInliers = nInliers + 1;
      end
   end
```

Generating samples

Random sampling

Model building

Thresholding
Inlier counting




```
function RANSAC_line

a = 1;
b = 1;

nPoints = 100;
x = rand(nPoints,1)-0.5;
y = a*x + b + 0.2*randn(nPoints,1);

nRansacIter = 500;
threshold = 0.1;
max_nInliers = 0;
for i = 1 : nRansacIter
   r = randperm(nPoints);
   s1 = [x(r(1)); y(r(1)); 1];
   s2 = [x(r(2)); y(r(2)); 1];

   l = GetLineFromTwoPoints(s1, s2);

   nInliers = 0;
   for j = 1 : nPoints
      e = abs(l' * [x(j); y(j); 1])/norm(l(1:2));
      if (e < threshold)
         nInliers = nInliers + 1;
      end
   end
    if (nInliers > max_nInliers)
      max_nInliers = nInliers;
      L = l;
   end
end
```

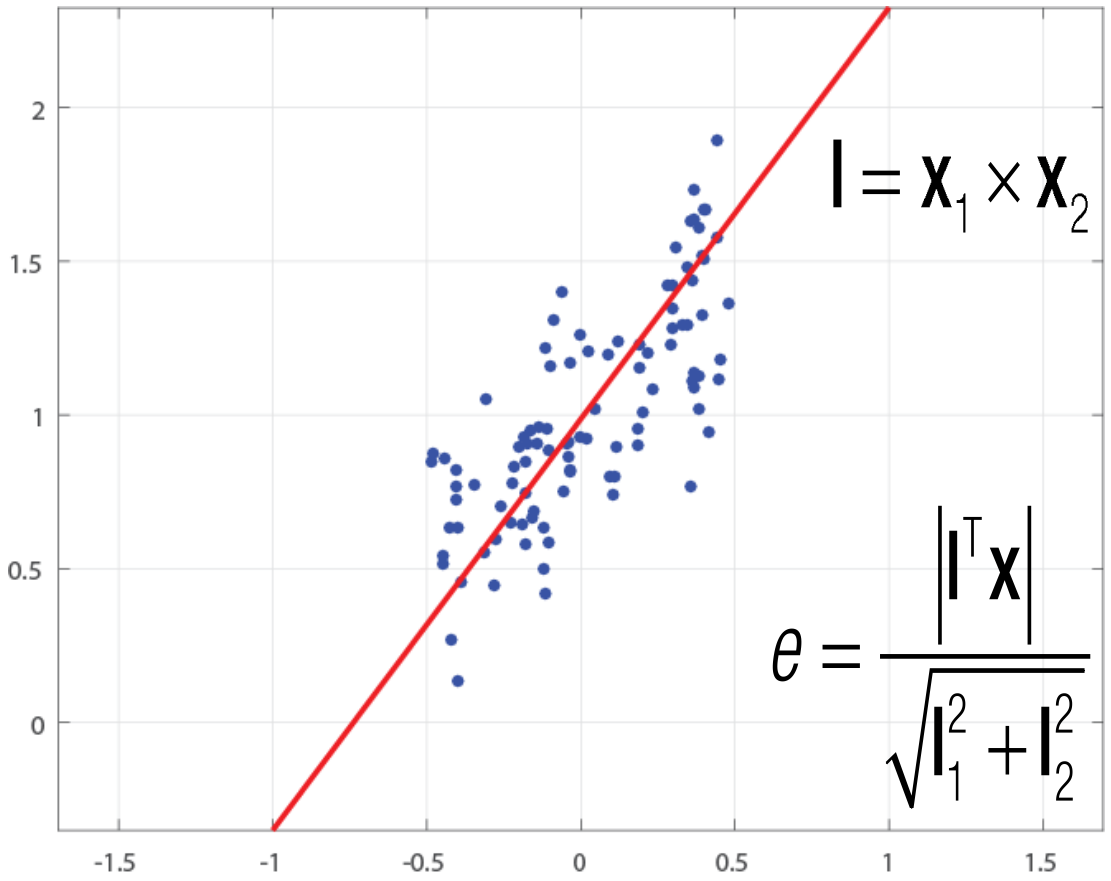
Generating samples

Random sampling

Model building

Thresholding
Inlier counting

Model updating




```
function RANSAC_line

a = 1;
b = 1;

nPoints = 100;
x = rand(nPoints,1)-0.5;
y = a*x + b + 0.2*randn(nPoints,1);

nRansacIter = 500;
threshold = 0.1;
max_nInliers = 0;
for i = 1 : nRansacIter
   r = randperm(nPoints);
   s1 = [x(r(1)); y(r(1)); 1];
   s2 = [x(r(2)); y(r(2)); 1];

   l = GetLineFromTwoPoints(s1, s2);

   nInliers = 0;
   for j = 1 : nPoints
      e = abs(l' * [x(j); y(j); 1])/norm(l(1:2));
      if (e < threshold)
         nInliers = nInliers + 1;
      end
   end
    if (nInliers > max_nInliers)
      max_nInliers = nInliers;
      L = l;
   end
end
```

Generating samples

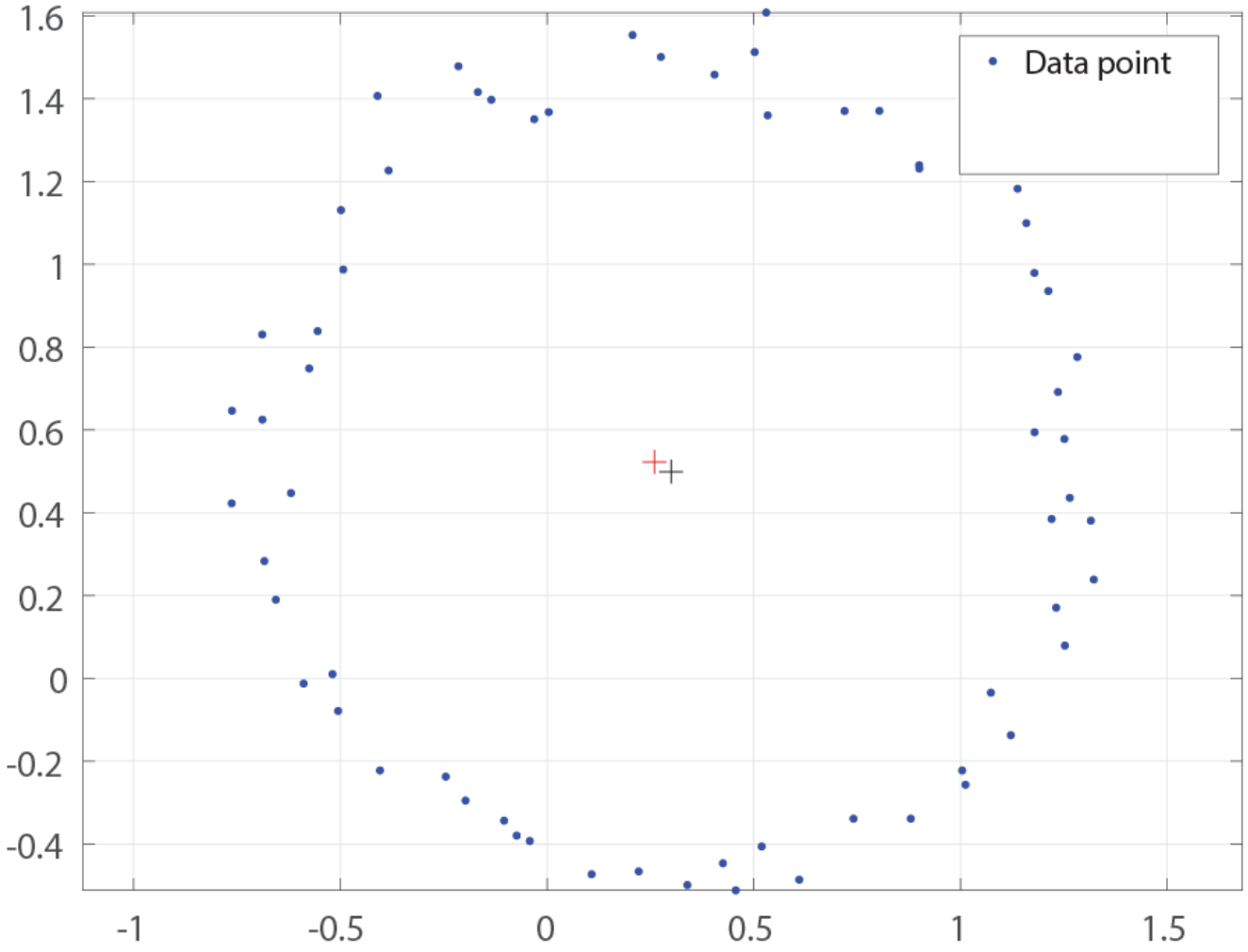
Random sampling

Model building

Thresholding
Inlier counting

Model updating

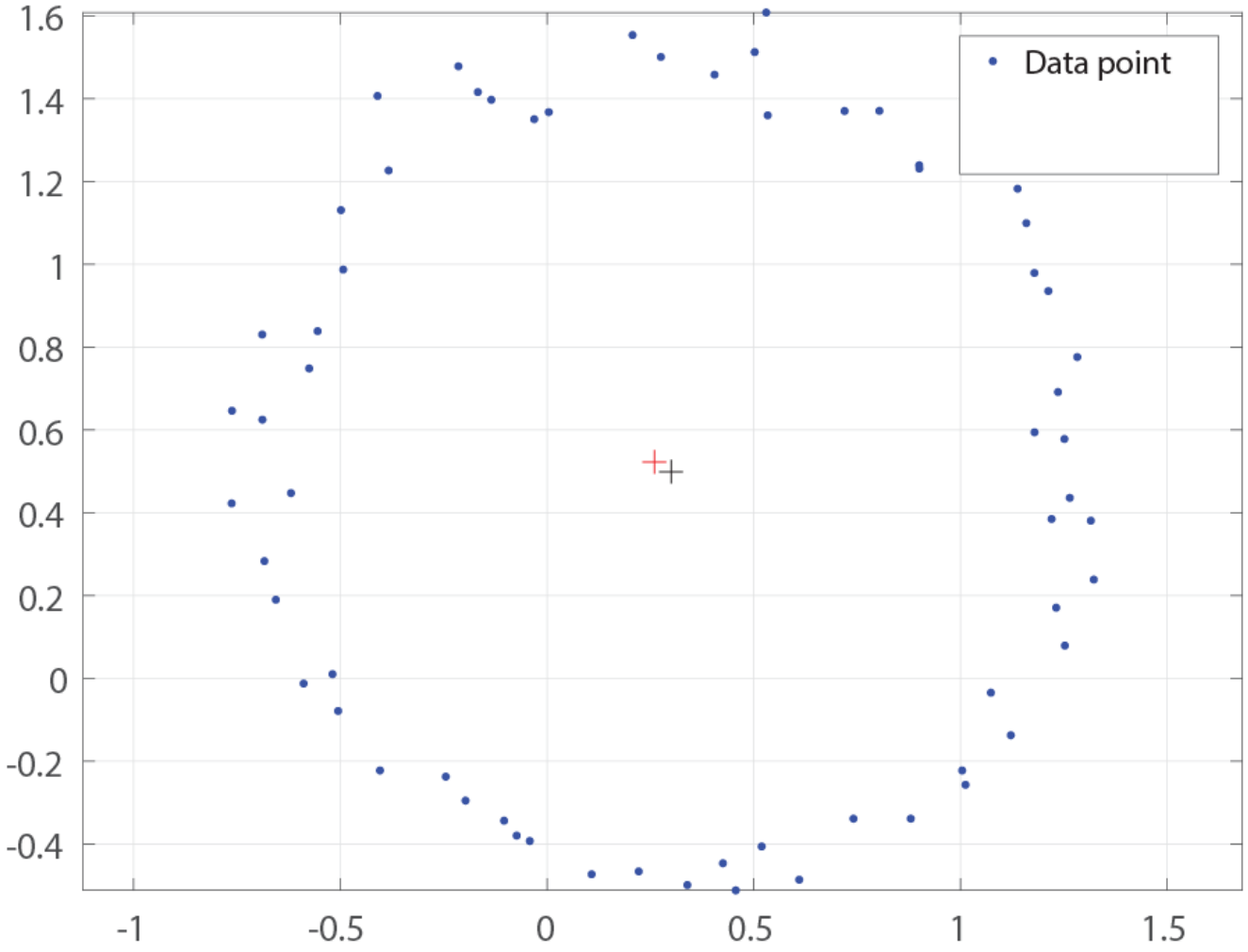
Recall: Circle Fitting (Ax=b)



$$(x_1 - c_x)^2 + (y_1 - c_y)^2 = r^2$$
$$\vdots$$
$$(x_n - c_x)^2 + (y_n - c_y)^2 = r^2$$

Unkowns: c_x, c_y, r

Recall: Circle Fitting (Ax=b)

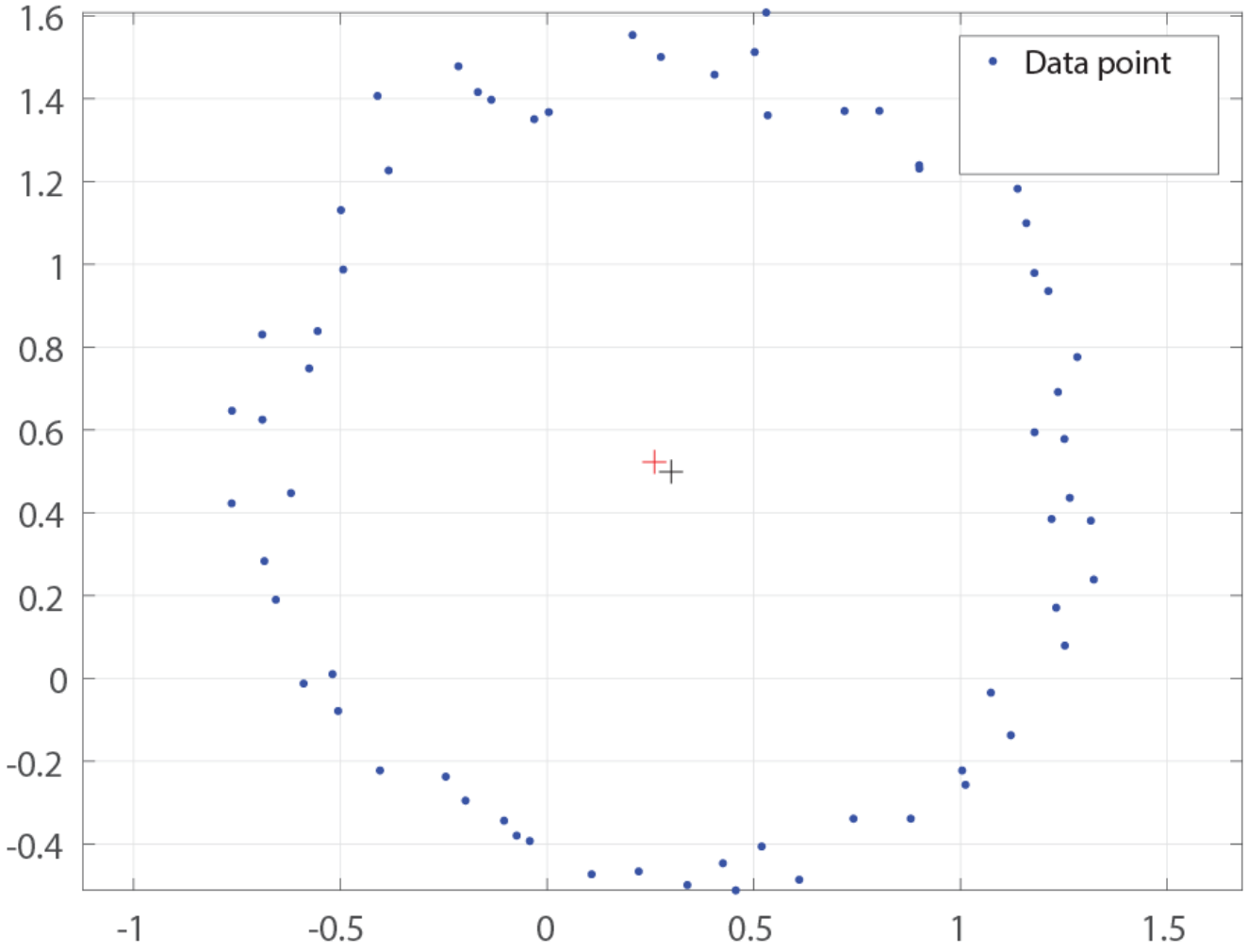


$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

Recall: Circle Fitting (Ax=b)



$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

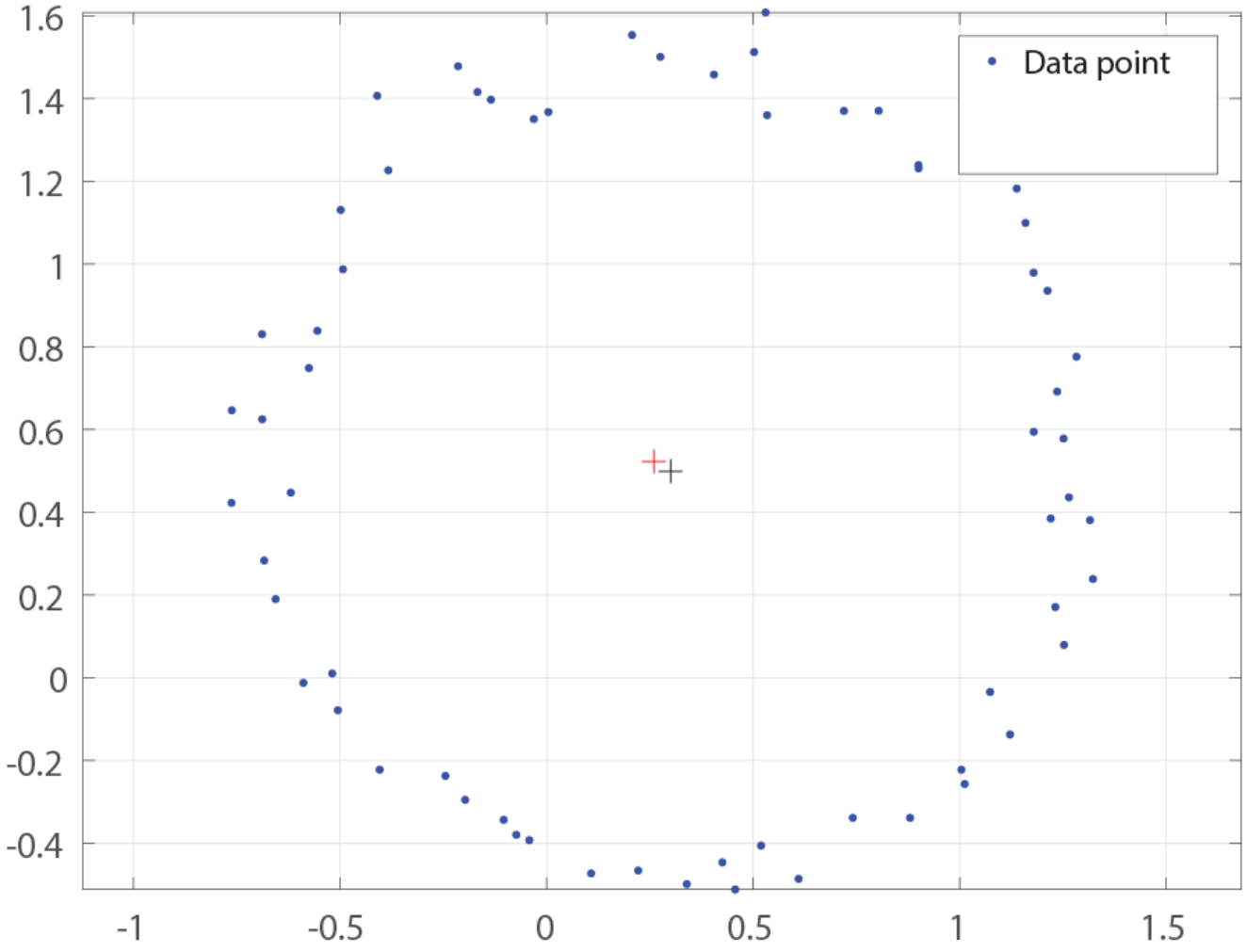
$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

$$\downarrow$$

$$x_i^2 - x_1^2 - 2c_x\left(x_i - x_1\right) + y_i^2 - y_1^2 - 2\left(y_i - y_1\right)c_y = 0$$

Recall: Circle Fitting (Ax=b)



$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

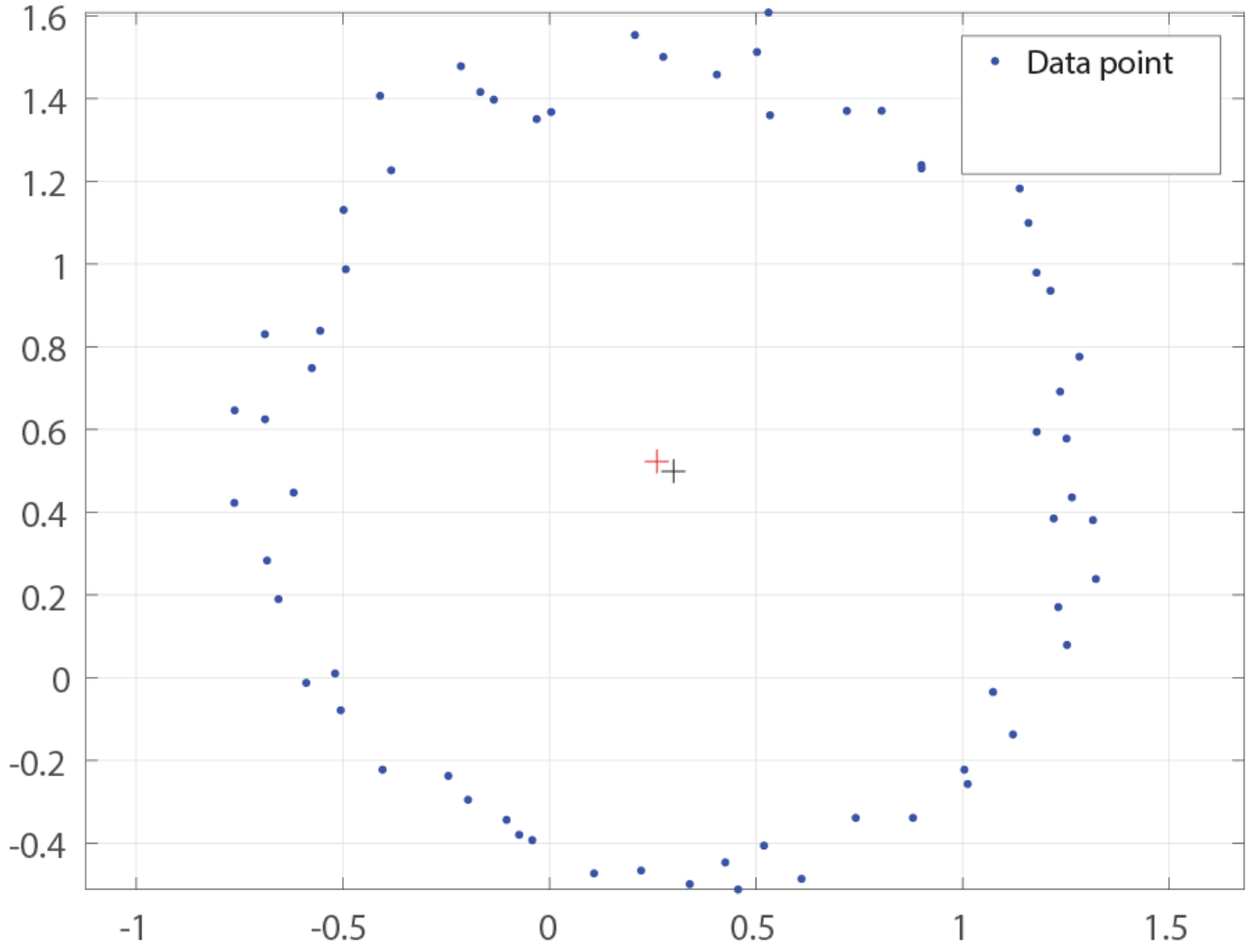
$$\downarrow$$

$$x_i^2 - x_1^2 - 2c_x\left(x_i - x_1\right) + y_i^2 - y_1^2 - 2\left(y_i - y_1\right)c_y = 0$$

$$\downarrow$$

$$\begin{bmatrix} 2\left(x_2 - x_1\right) & 2\left(y_2 - y_1\right) \\ \vdots & \vdots \\ 2\left(x_n - x_1\right) & 2\left(y_n - y_1\right) \end{bmatrix} \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix}$$

Circle Fitting (Ax=b)



$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

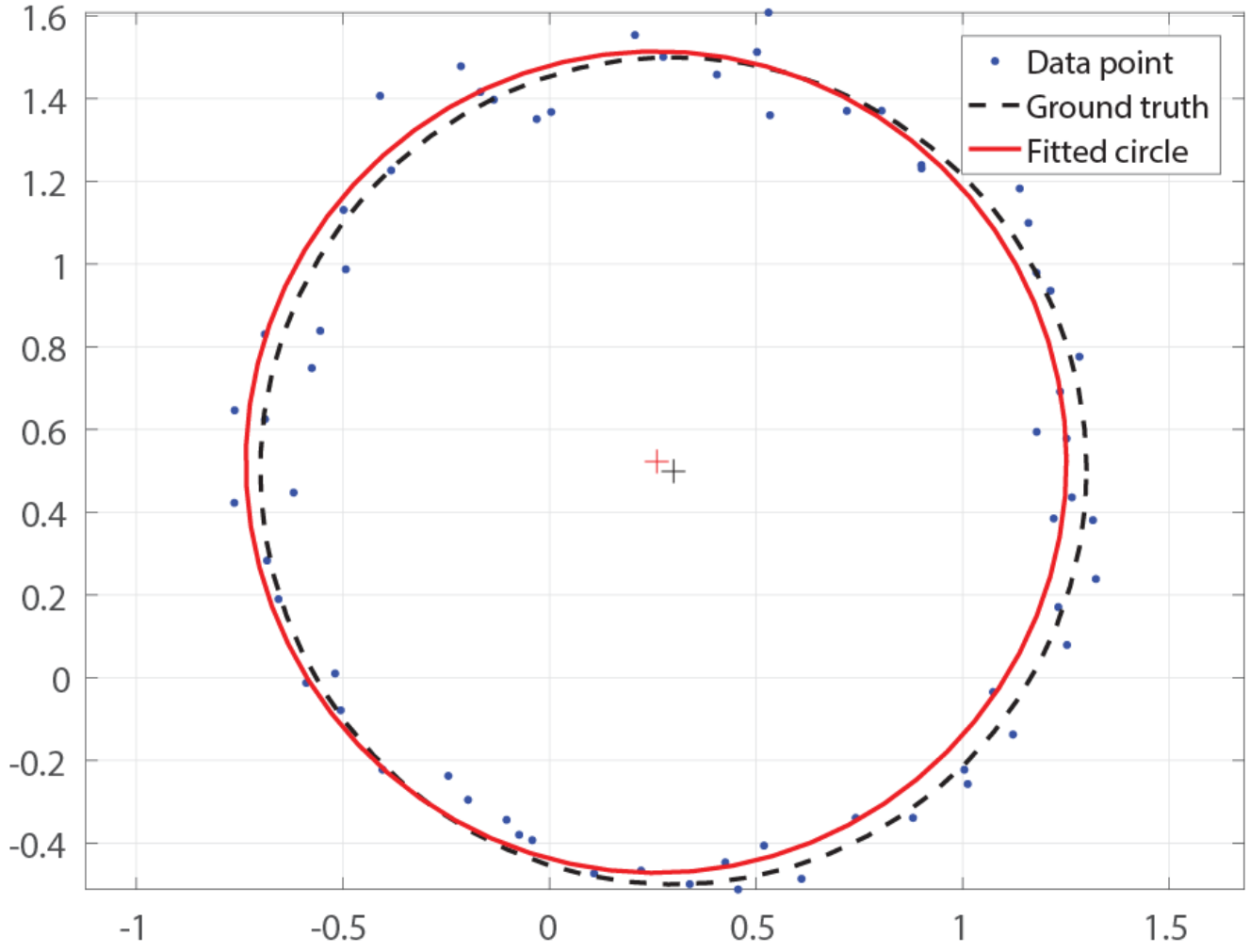
$$\downarrow$$

$$x_i^2 - x_1^2 - 2c_x\left(x_i - x_1\right) + y_i^2 - y_1^2 - 2\left(y_i - y_1\right)c_y = 0$$

$$\downarrow$$

$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ \vdots & \\ \mathbf{A} & \\ 2(x_n - x_1) & 2(y_n - y_1) \end{bmatrix} \begin{bmatrix} c_x \\ c_y \end{bmatrix} \mathbf{x} = \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ \mathbf{b} \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix}$$

Recall: Circle Fitting (Ax=b)



$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

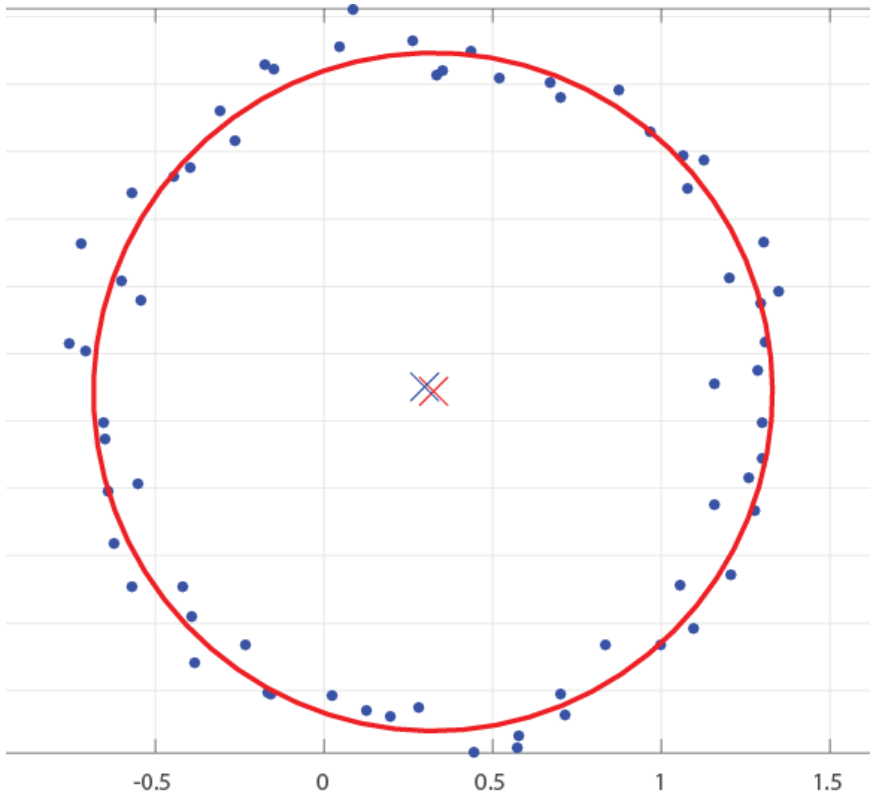
$$\downarrow$$

$$x_i^2 - x_1^2 - 2c_x\left(x_i - x_1\right) + y_i^2 - y_1^2 - 2\left(y_i - y_1\right)c_y = 0$$

$$\downarrow$$

$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ \vdots & \\ \mathbf{A} & \\ 2(x_n - x_1) & 2(y_n - y_1) \end{bmatrix} \begin{bmatrix} c_x \\ c_y \end{bmatrix} \mathbf{x} = \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ \mathbf{b} \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix}$$

Circle Fitting RANSAC



$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ \vdots & \vdots \\ 2(x_n - x_1) & 2(y_n - y_1) \end{bmatrix} \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix}$$

Download **RANSAC_circle.m**

```
function RANSAC_circle

cx = 0.3;
cy = 0.5;
r = 1;

theta = 0:0.1:2*pi+0.1;
theta = theta';
x = cos(theta) + cx + 0.05*randn(size(theta));
y = sin(theta) + cy + 0.05*randn(size(theta));

figure(1)
clf;
plot(x,y, 'b.');
axis equal
grid on

nRansacIter = 500;
threshold = 0.1;
max_nInliers = 0;
%% RANSAC circle
% U: Estimated center of circle
% R: Radius

%%
hold on
plot(R*cos(theta)+U(1), R*sin(theta)+U(2), 'r-');
hold on
plot(U(1), U(2), 'rx');
hold on
plot(cx, cy, 'bx');
```

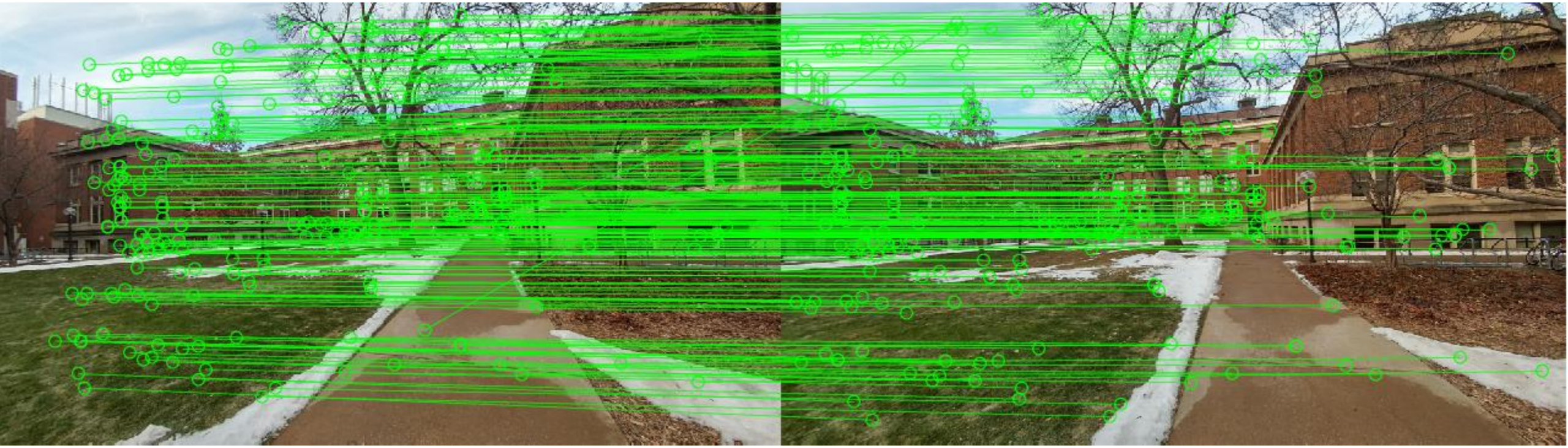
Fill out this part

HW #4 RANSAC Fundamental matrix

Algorithm 1 GetInliersRANSAC

$n \leftarrow 0$
for $i = 1 : M$ **do**
    Choose 8 correspondences, $\mathbf{u}_r$ and $\mathbf{v}_r$, randomly from $\mathbf{u}$ and $\mathbf{v}$.
    $\mathbf{F}_r$ = `ComputeFundamentalMatrix(`$\mathbf{u}_r$`,` $\mathbf{v}_r$`)`
    Compute the number of inliers, n_r, with respect to `F`.
    if $n_r > n$ **then**
        $n \leftarrow n_r$
        $\mathbf{F} = \mathbf{F}_r$
    end if
end for

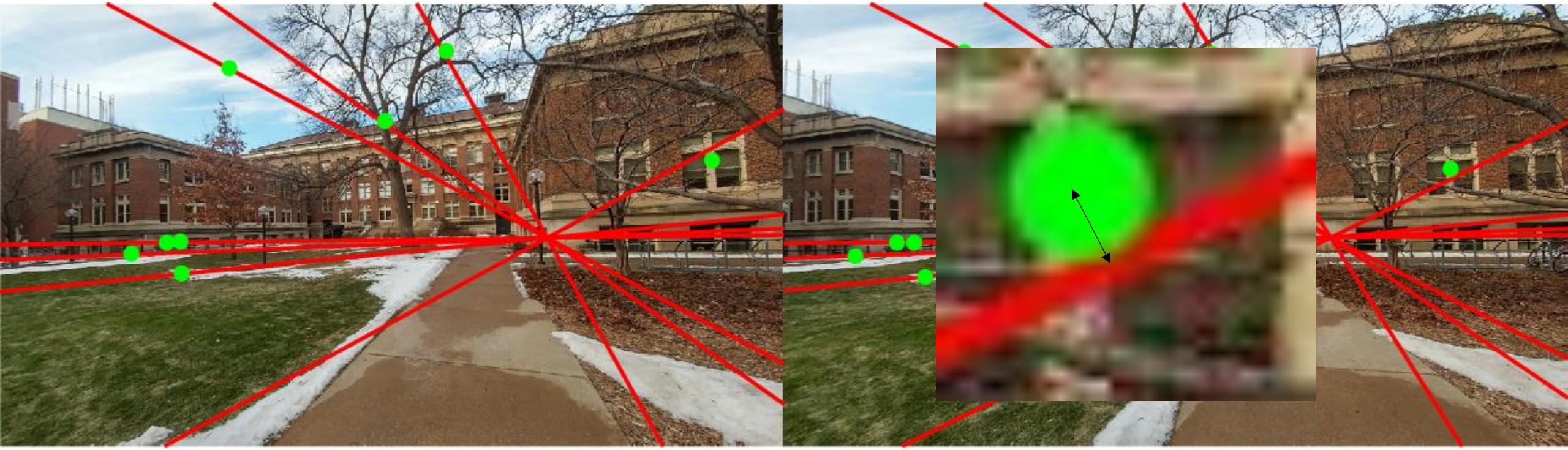
Fundamental Matrix Computation via RANSAC



Fundamental Matrix Computation via RANSAC

$$\begin{bmatrix} u_1^x v_1^x & u_1^y v_1^x & v_1^x & u_1^x v_1^y & u_1^y v_1^y & v_1^y & u_1^x & u_1^y & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_m^x v_m^x & u_m^y v_m^x & v_m^x & u_m^x v_m^y & u_m^y v_m^y & v_m^y & u_m^x & u_m^y & 1 \end{bmatrix} \mathbf{A} \quad \mathbf{x} \quad \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Fundamental Matrix Computation via RANSAC



Epipolar line: $\mathbf{l}_u = \mathbf{F}\mathbf{u}$

Distance: $$d = \frac{\left|au_x + bu_y + c\right|}{\sqrt{a^2 + b^2}} = \frac{\left|\mathbf{F}\mathbf{u}\right|}{\sqrt{(\mathbf{F}_{1,:}\mathbf{u})^2 + (\mathbf{F}_{2,:}\mathbf{u})^2}}$$

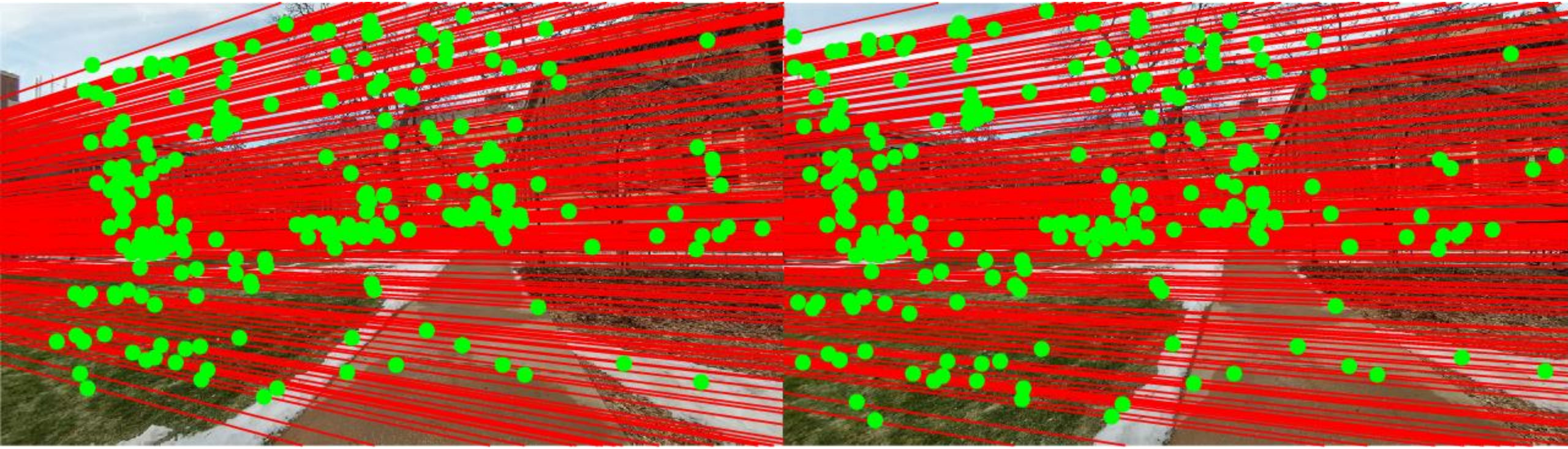
Fundamental Matrix Computation via RANSAC

# of inliers: 65 out of 260

Fundamental Matrix Computation via RANSAC

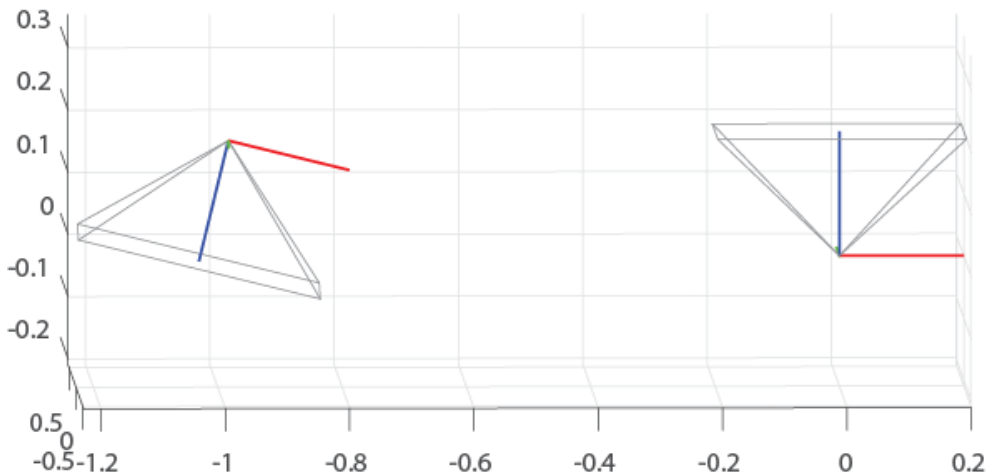
# of inliers: 65 out of 260

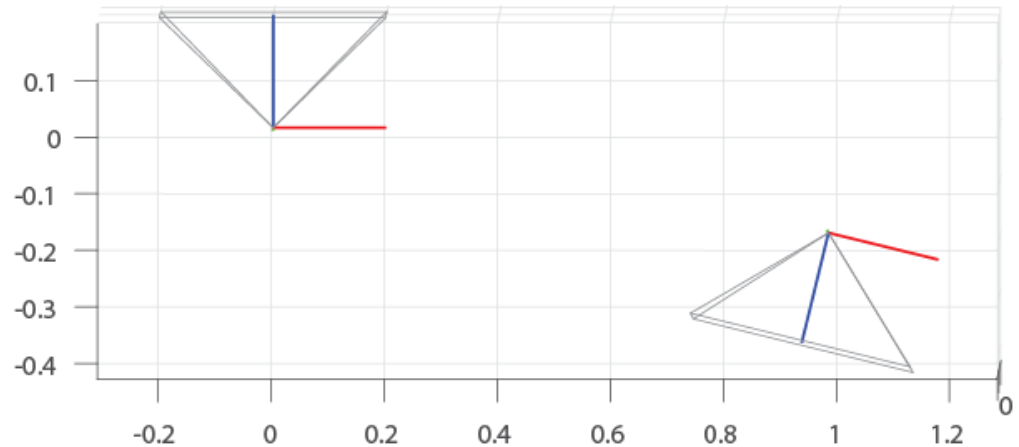
Fundamental Matrix Computation via RANSAC

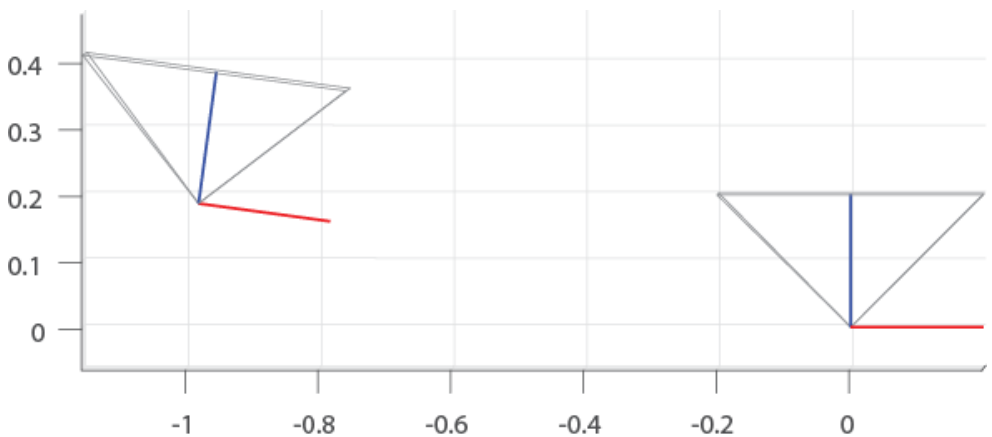


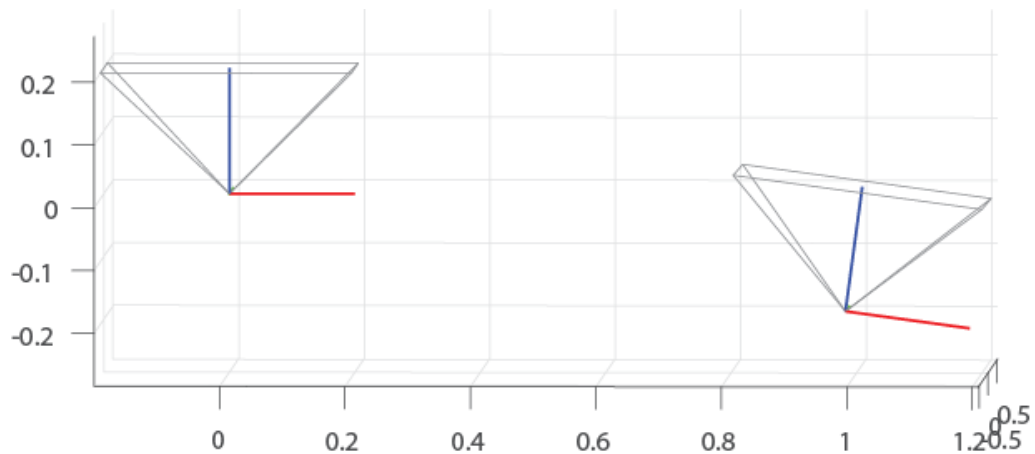
# of inliers: 186 out of 260

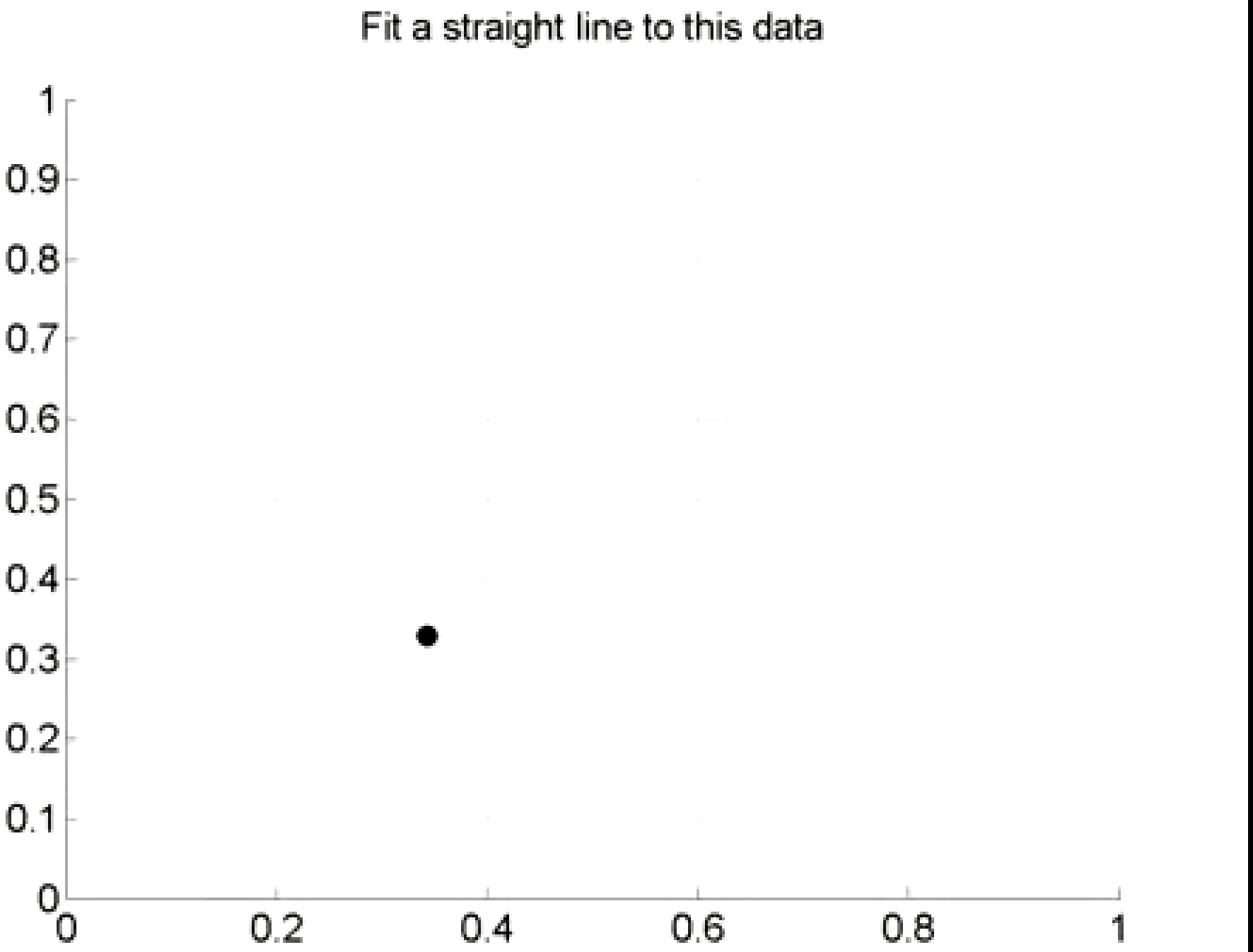
Four Camera Pose Config. From Essential Matrix











Daniel Wedge “The RANSAC Song”