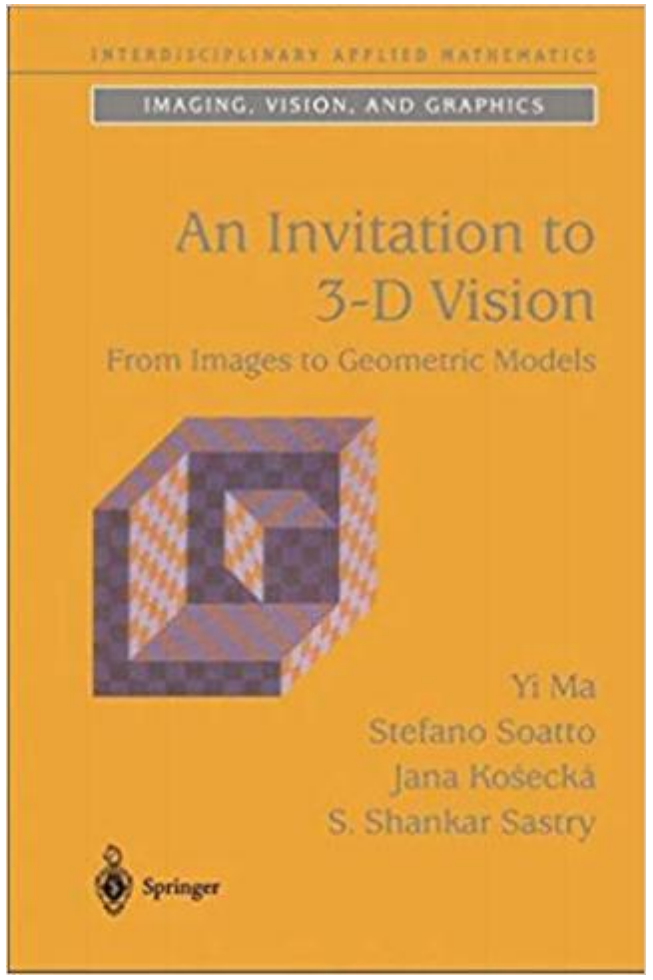
16.485: VNAV - Visual Navigation for Autonomous Vehicles

Luca Carlone

Lecture 14: 2-view Geometry

Today

- 2-view geometry
- RANSAC
- 3D-3D correspondences


INTERDISCIPLINARY APPLIED MATHEMATICS

IMAGING, VISION, AND GRAPHICS

An Invitation to 3-D Vision

From Images to Geometric Models

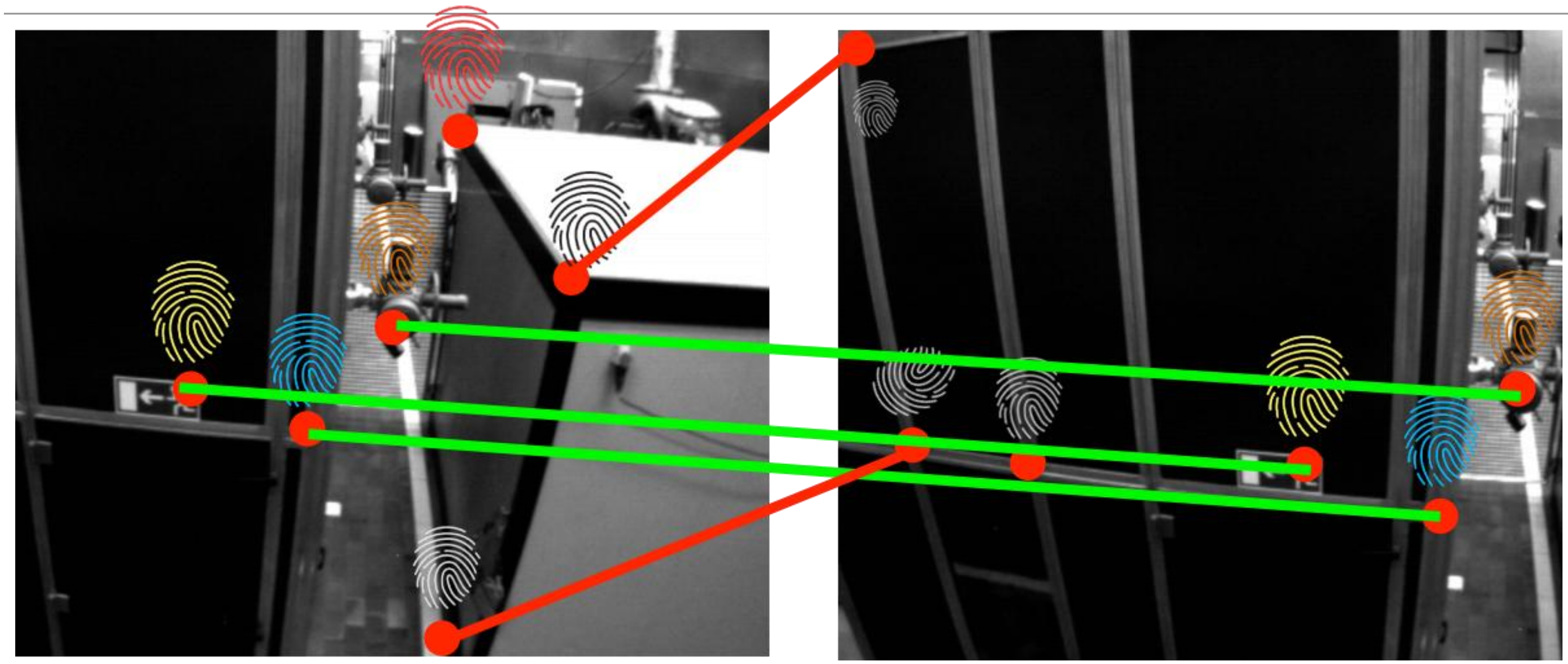
Yi Ma
Stefano Soatto
Jana Košecká
S. Shankar Sastry

Springer

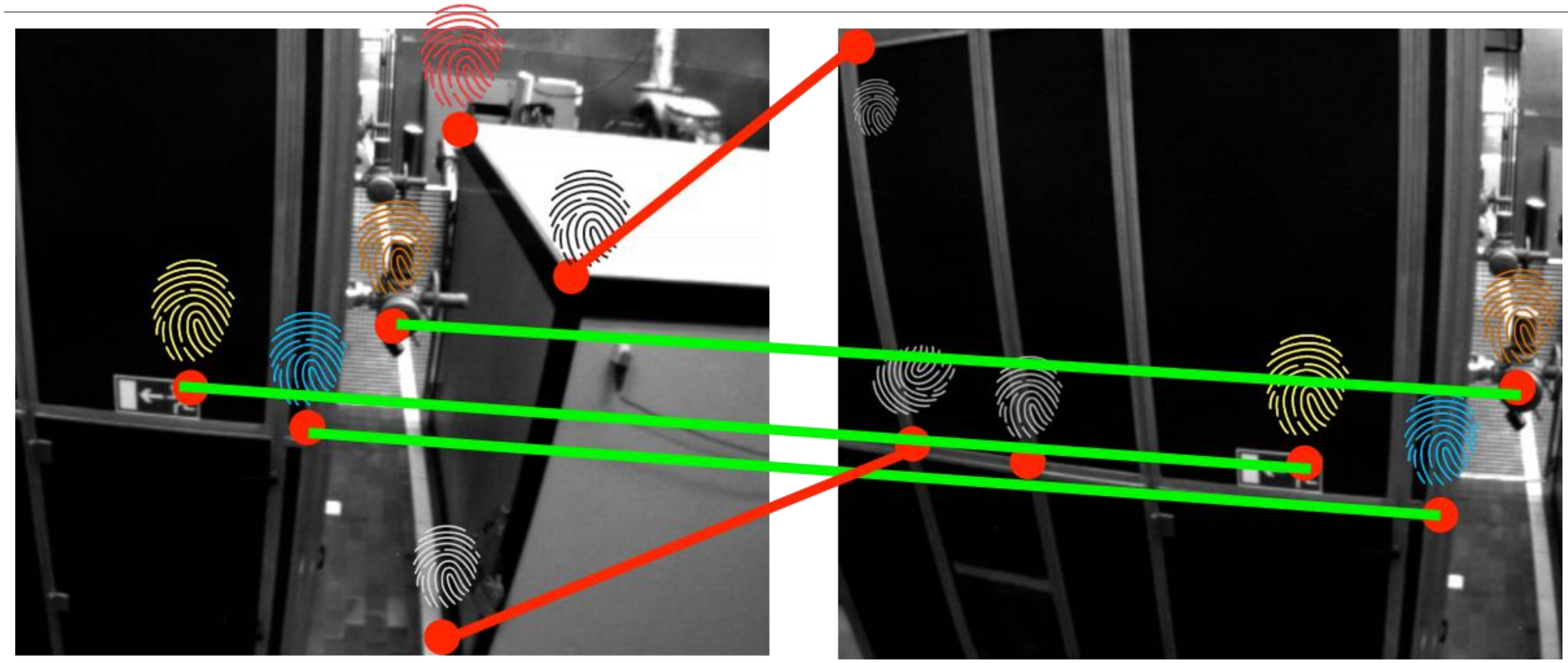
Chapter 5

Reconstruction from Two Calibrated Views

Recap: Point Correspondences



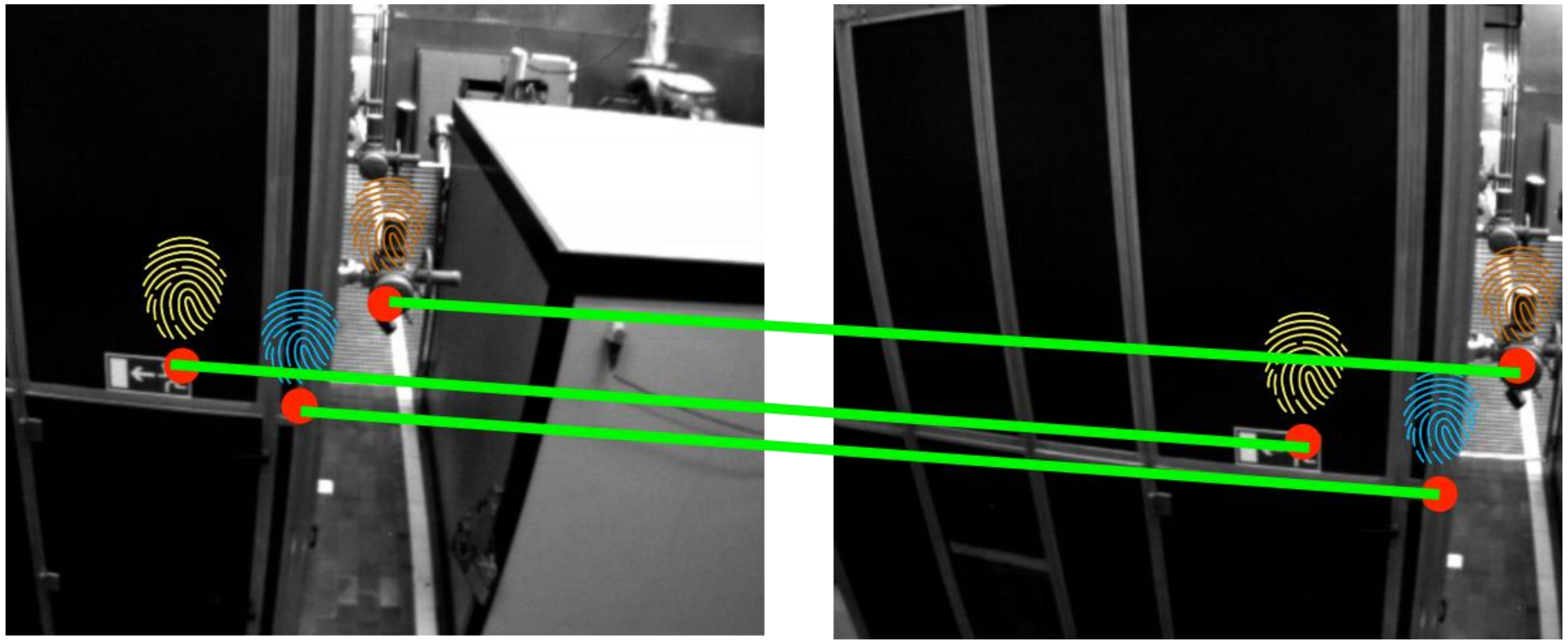
2-view Geometry



Question: can we estimate the motion of the camera between I_1 and I_2 using pixel correspondences?

如何根据图像I_1、I_2 间的像素匹配估计相机的运动?

2-view Geometry

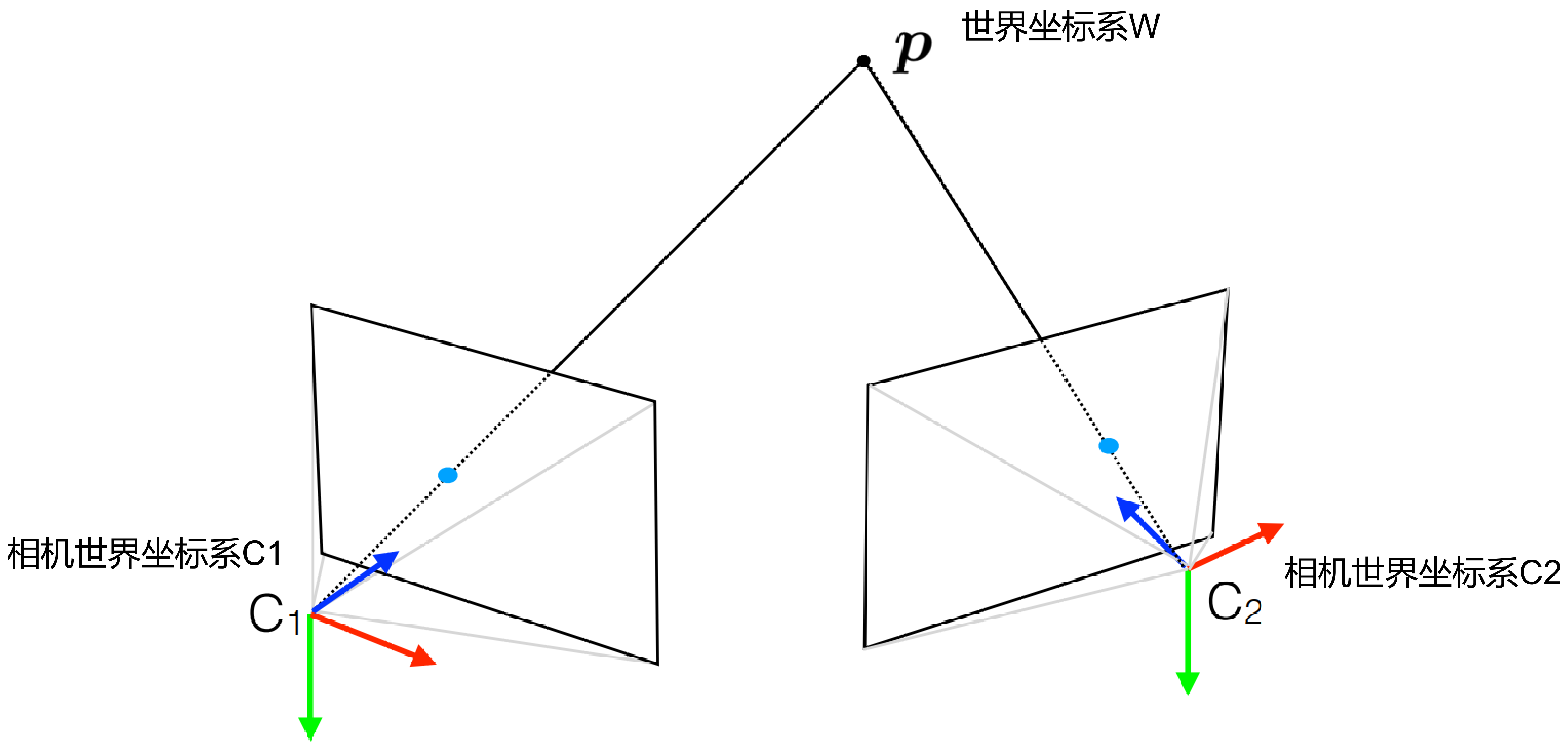


如何根据图像*I*₁、*I*₂ 间的像素匹配估计相机的运动?

该问题的3个假设:

1. 没有错误的匹配点 (no wrong correspondences (outliers))
2. 3D点没有移动 (3D point is not moving)
3. 相机内参已知 (camera calibration is known)

2-view Geometry



$$p_z^{\mathrm{C1}}\ \tilde{\boldsymbol{x}}_1 = \boldsymbol{K}_1\left[\boldsymbol{R}_{\mathrm{W}}^{\mathrm{C1}}\ \boldsymbol{t}_{\mathrm{W}}^{\mathrm{C1}}\right]\tilde{\boldsymbol{p}}^{\mathrm{W}}$$

$$p_z^{\mathrm{C2}}\ \tilde{\boldsymbol{x}}_2 = \boldsymbol{K}_2\left[\boldsymbol{R}_{\mathrm{W}}^{\mathrm{C2}}\ \boldsymbol{t}_{\mathrm{W}}^{\mathrm{C2}}\right]\tilde{\boldsymbol{p}}^{\mathrm{W}}$$

$$\boldsymbol{K}_1 = \begin{bmatrix} s_{x_1} f_1 & s_{\theta_1} f_1 & o_{x_1} \\ 0 & s_{y_1} f_1 & o_{y_1} \\ 0 & 0 & 1 \end{bmatrix}$$

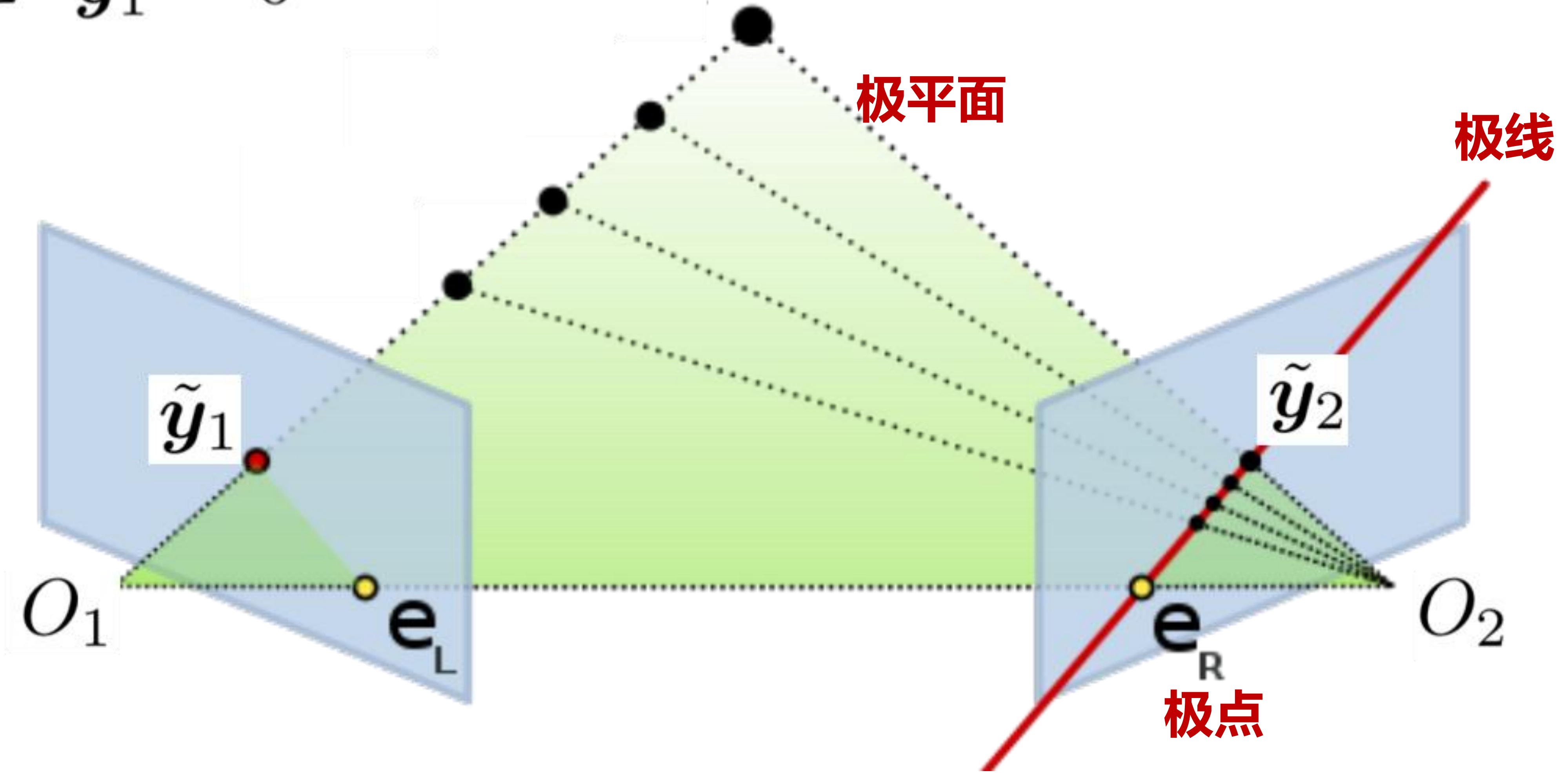
$$\boldsymbol{K}_2 = \begin{bmatrix} s_{x_2} f_2 & s_{\theta_2} f_2 & o_{x_2} \\ 0 & s_{y_2} f_2 & o_{y_2} \\ 0 & 0 & 1 \end{bmatrix}$$

回顾

Epipolar Geometry

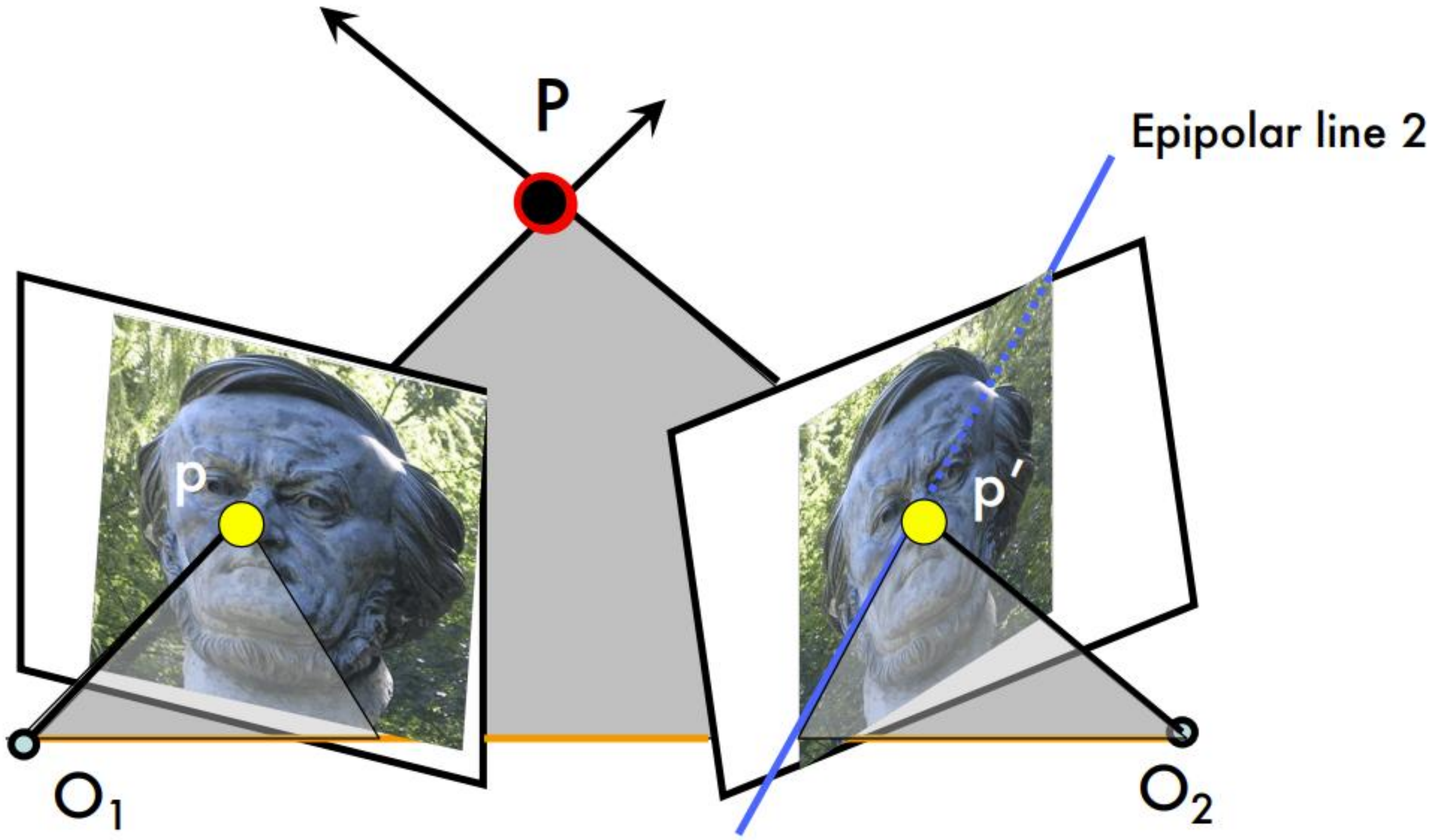
E 本质矩阵、F 基础矩阵

$$\tilde{y}_2^\top \ \boldsymbol{E} \ \tilde{y}_1 = 0$$



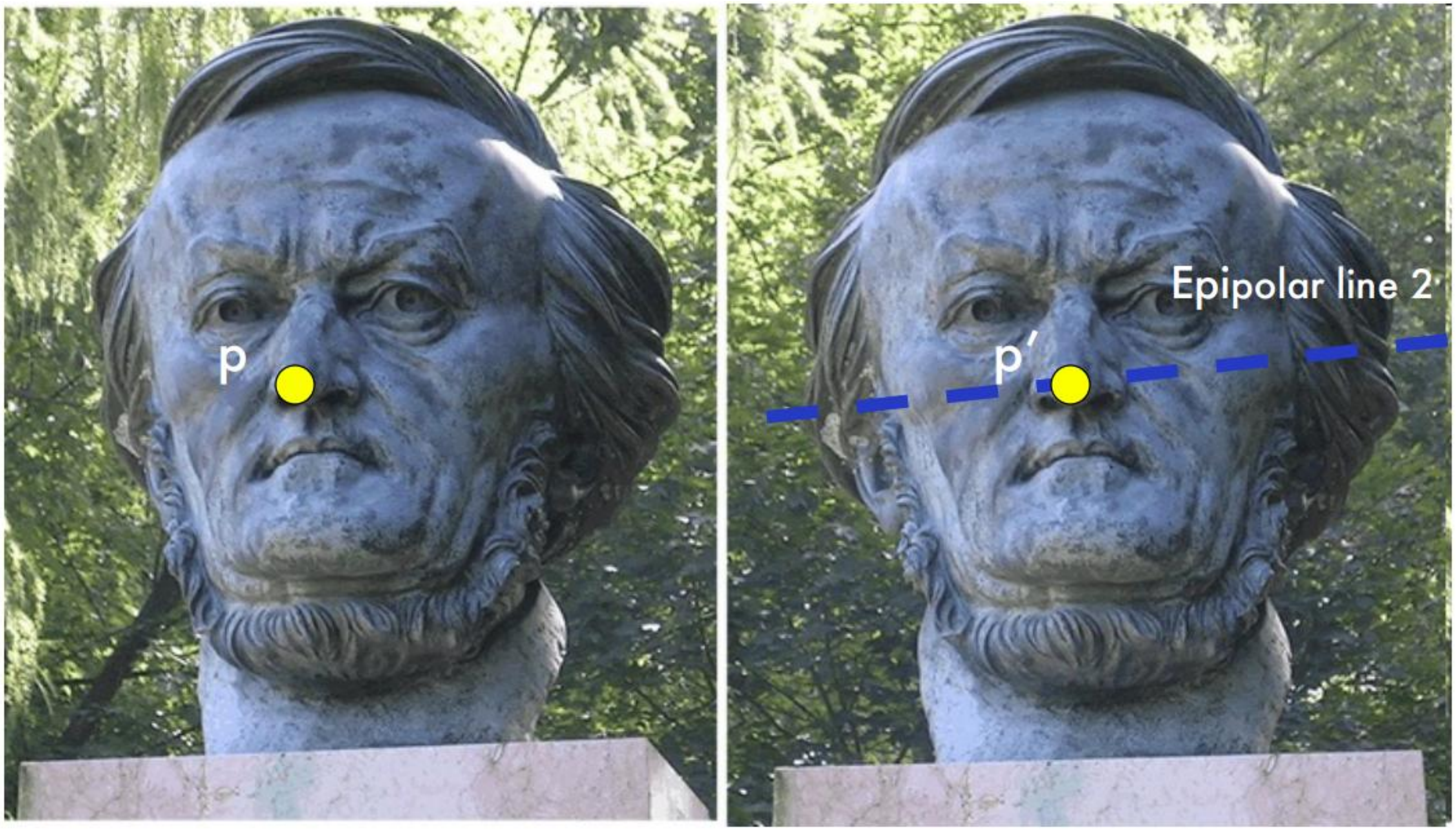
极平面	极线	极点
🟩 epipolar plane	/ epipolar line	e_L , e_R : epipoles

Epipolar Geometry



回顾

Epipolar Geometry



极线约束公式

- **本质矩阵**
  - **对规范化摄像机拍摄的两个视点图像间的极几何关系进行代数化描述：**

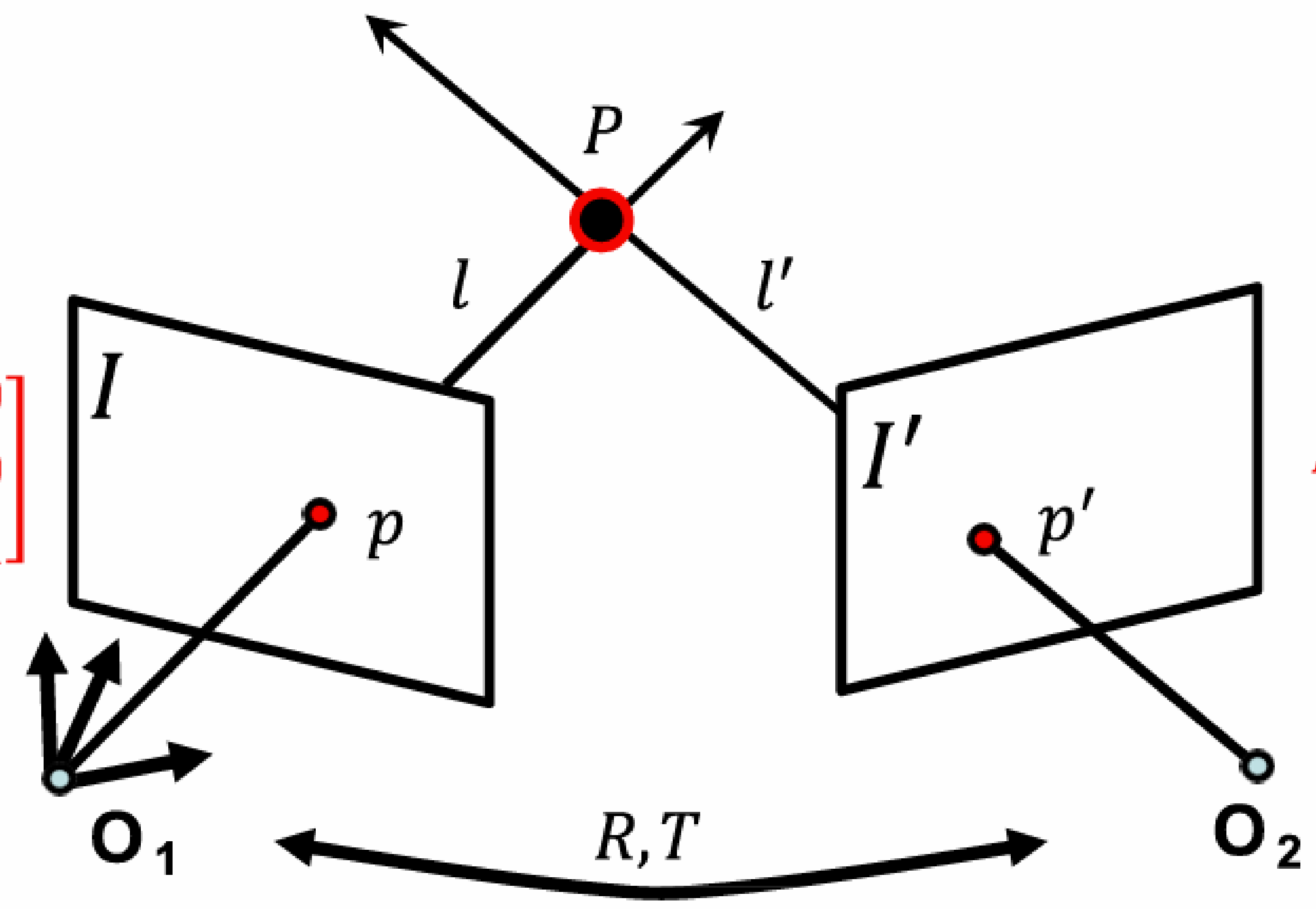
$$\mathbf{p}'^{\mathrm{T}} \boxed{\mathbf{E}} \mathbf{p} = 0$$

- **基础矩阵**

$$\mathbf{x}'^{\mathrm{T}} \boxed{\mathbf{K}'^{-\mathrm{T}} \mathbf{E} \mathbf{K}^{-1}} \mathbf{x} = 0$$

$$\mathbf{x}'^{\mathrm{T}} \boxed{\mathbf{F}} \mathbf{x} = 0$$

极线约束 – 本质矩阵



$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $K' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

图像 I 上的点 p 像素坐标为 (u, v) 图像 I'上的点 p' 像素坐标为 (u', v')

$K = K'$已知且为规范化相机 $\Rightarrow$ 三维点的欧式（非齐次）坐标 = 图像点的齐次坐标

空间点 p 在 O_1 坐标系下的非齐次坐标为 $(u, v, 1)$ 空间点 p' 在 O_2 坐标系下的非齐次坐标为 $(u', v', 1)$

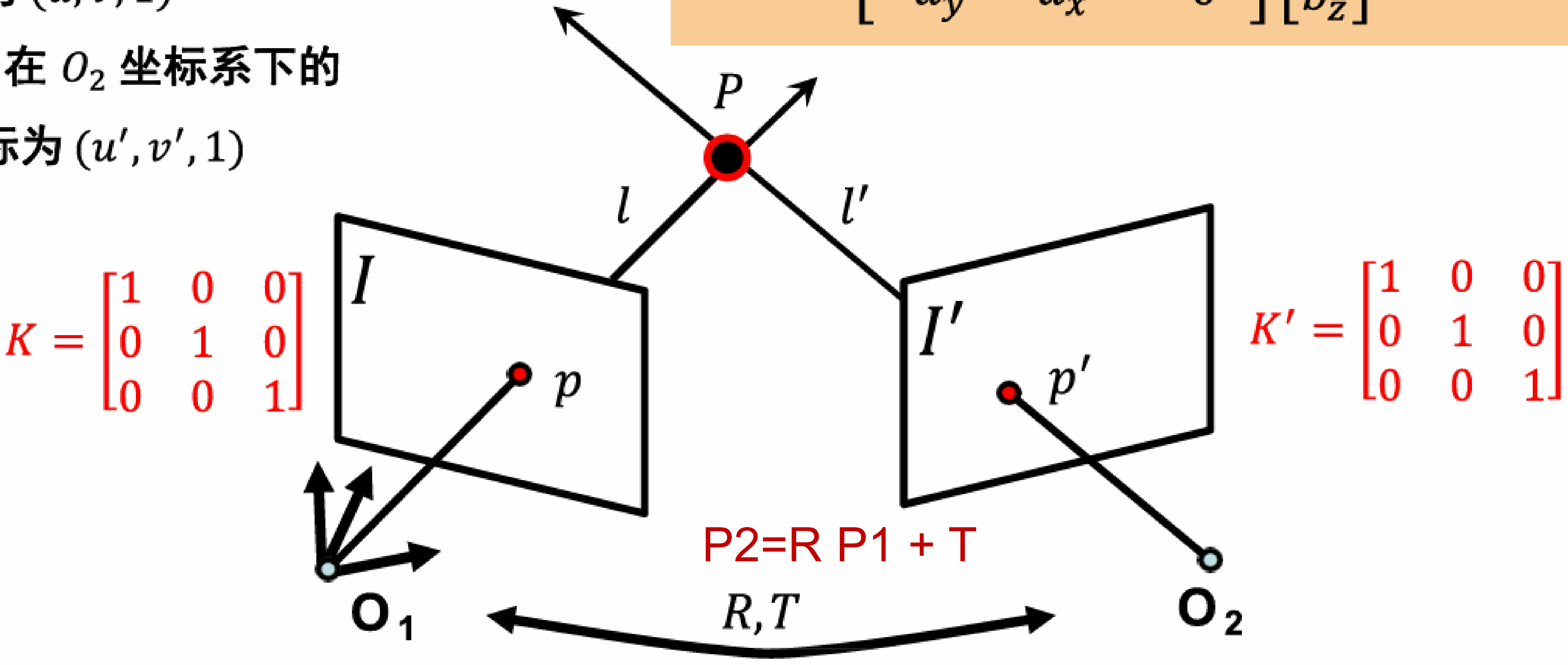
极线约束 – 本质矩阵

叉乘的矩阵表示 回顾

$$a \times b = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [a_\times]b$$

空间点 p 在 O_1 坐标系下的非齐次坐标为 $(u, v, 1)$

空间点 p' 在 O_2 坐标系下的非齐次坐标为 $(u', v', 1)$



P
l
l'
I
I'
p
p'
$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$K' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
P2=R P1 + T
O_1
O_2
R, T

p' 在O_1的坐标$= R^T p' - R^T T$

O_2在O_1的坐标$= -R^T T$

$$\Rightarrow \quad R^T T \times (R^T p' - R^T T) = R^T T \times R^T p'$$

垂直于极平面

$$\Downarrow$$

$$\boxed{p'^T [T \times R] p = 0} \quad \Leftarrow \quad [R^T T \times R^T p']^T \cdot p = 0$$

极线约束 – 本质矩阵

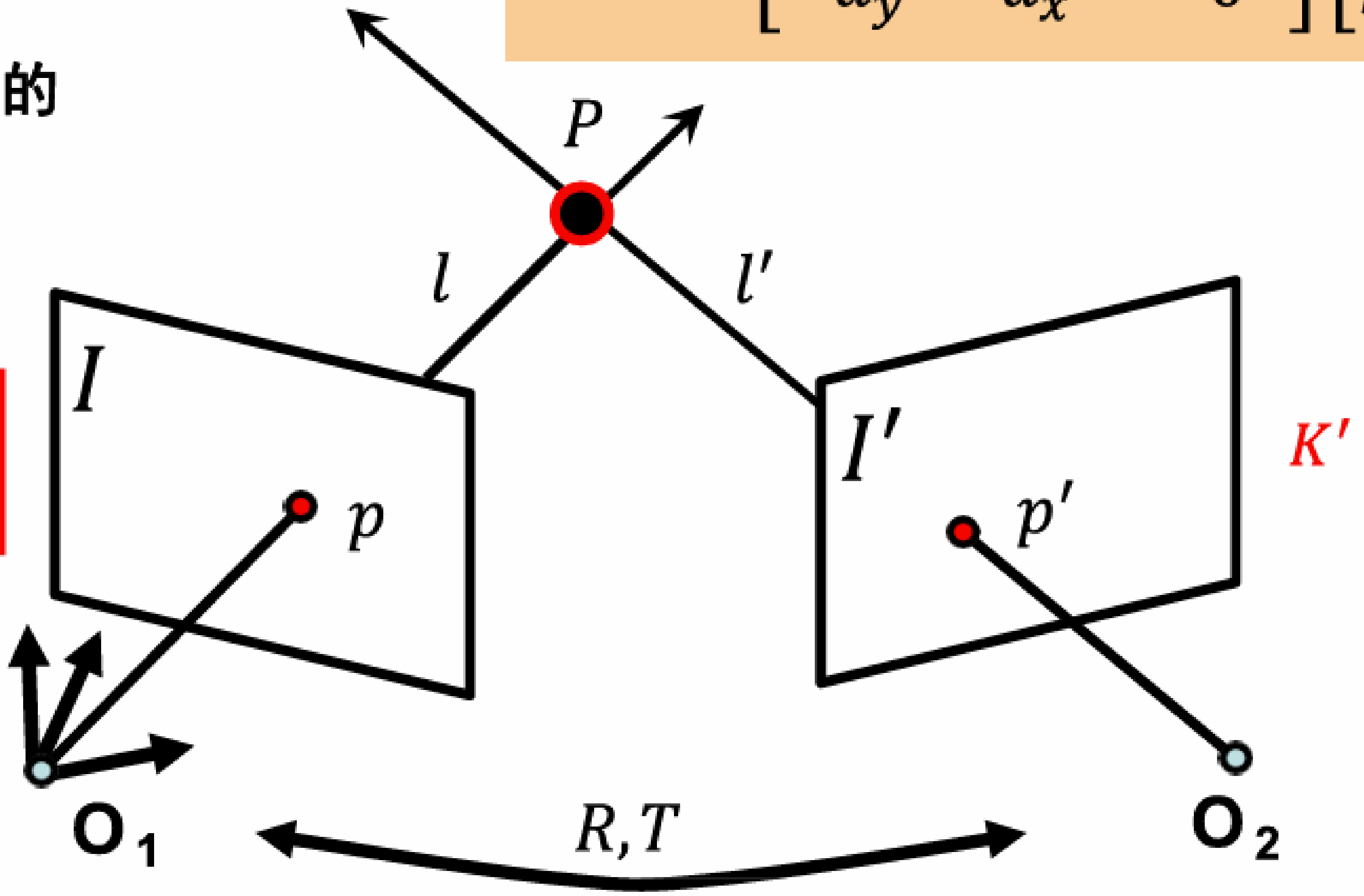
叉乘的矩阵表示 回顾

$$a \times b = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [a_\times]b$$

空间点 p 在 O_1 坐标系下的非齐次坐标为 $(u, v, 1)$

空间点 p' 在 O_2 坐标系下的非齐次坐标为 $(u', v', 1)$

$$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$K' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$p'^T[T \times R]p = p'^T[T_\times]Rp = 0$$

$$\boxed{p'^T E p = 0}$$

$$E = T \times R = [T_\times]R$$

E = 本质矩阵

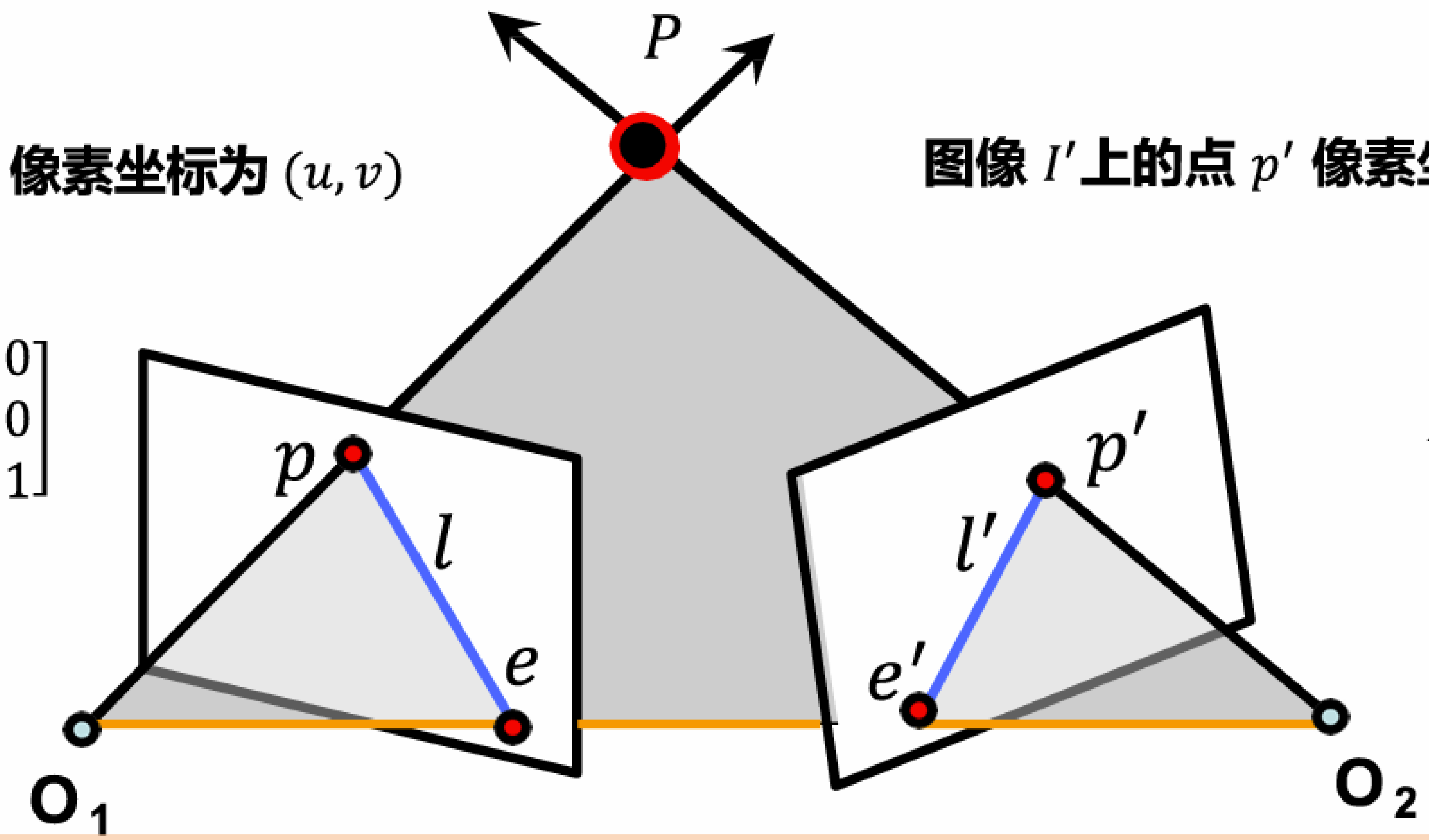
极线约束 – 本质矩阵

图像 I 上的点 p 像素坐标为 (u,v)

图像 I'上的点 p' 像素坐标为 (u',v')

$$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- p 对应的极线是 l'（$l' = Ep$）
- p'对应的极线是 l（$l = E^T p'$）
- $Ee = 0$ 与 $e'^T E = 0$
- E 是奇异的（秩2）
- E 5个自由度 （三个旋转+三个平移，$\det(E) = 0$去掉一个自由度）

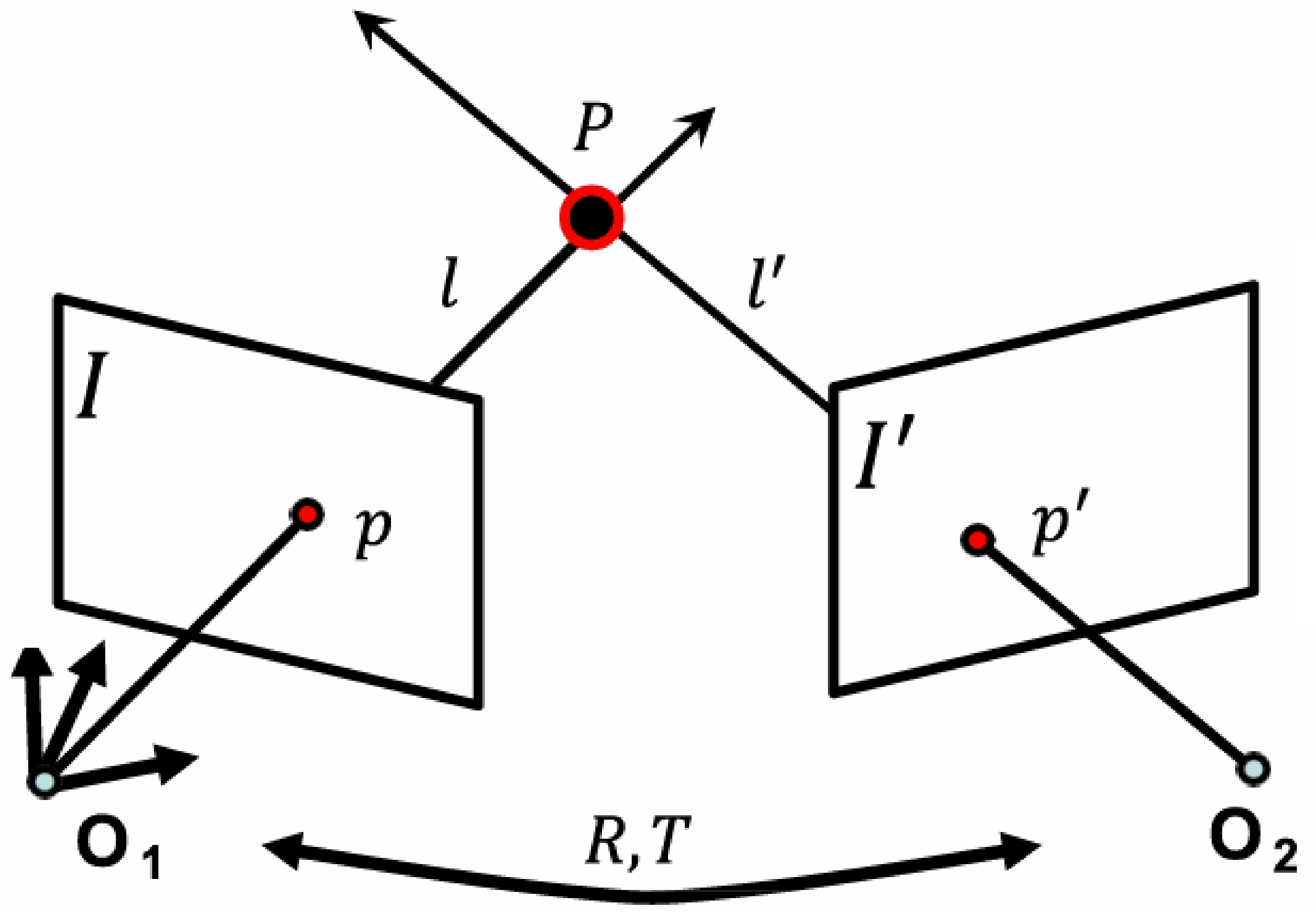
$$\boxed{p'^T E p = 0}$$

极线约束 – 基础矩阵

思路：变换到规范化摄像机！

$$p_c = K^{-1}p$$

$$p'_c = K'^{-1}p'$$



图像 I 上的点 p 像素坐标为 (u, v)

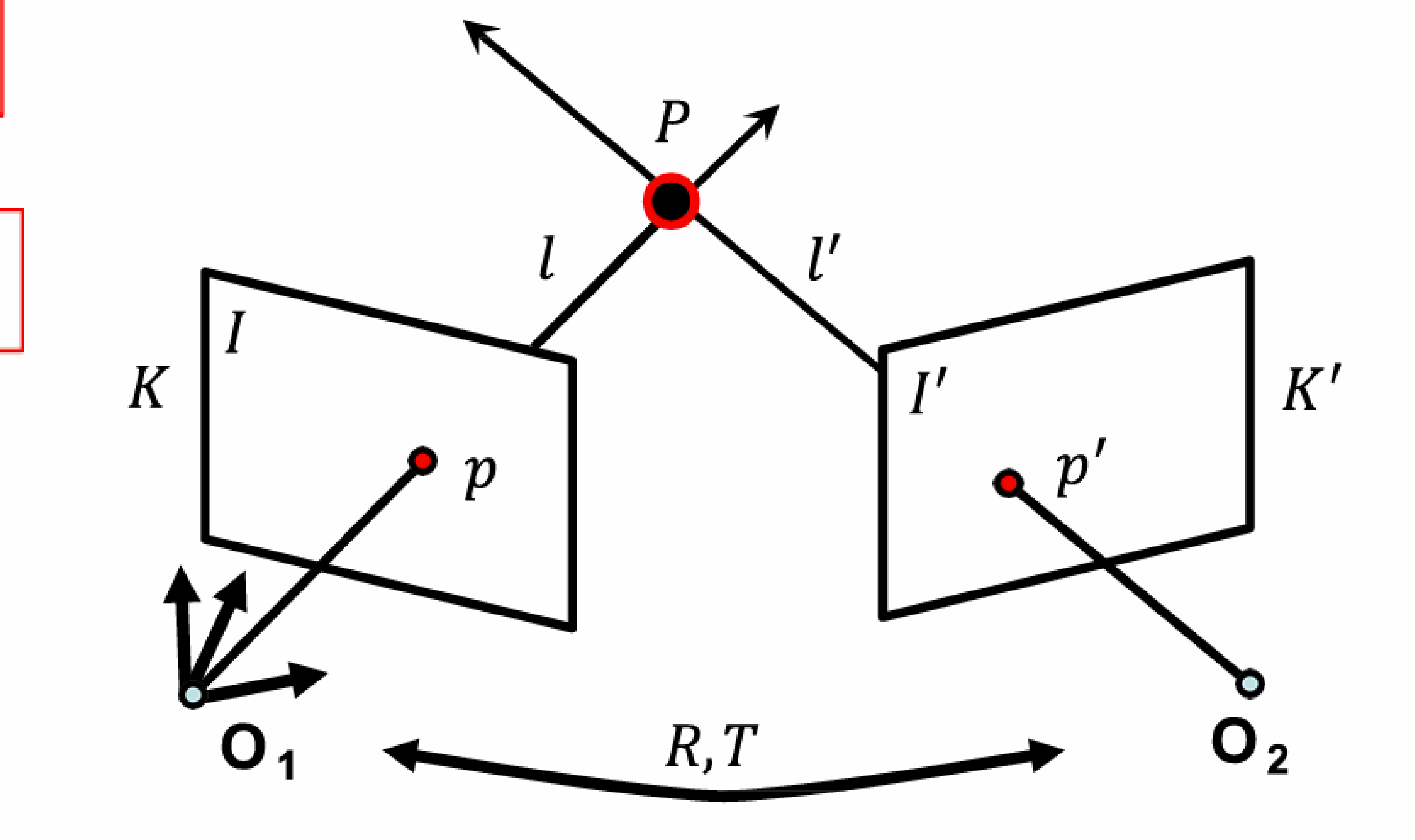
图像 I' 上的点 p' 像素坐标为 (u', v')

$K = K'$已知且为规范化相机

极线约束 – 基础矩阵

$$\boxed{p_c = K^{-1}p}$$

$$\boxed{p'_c = K'^{-1}p'}$$

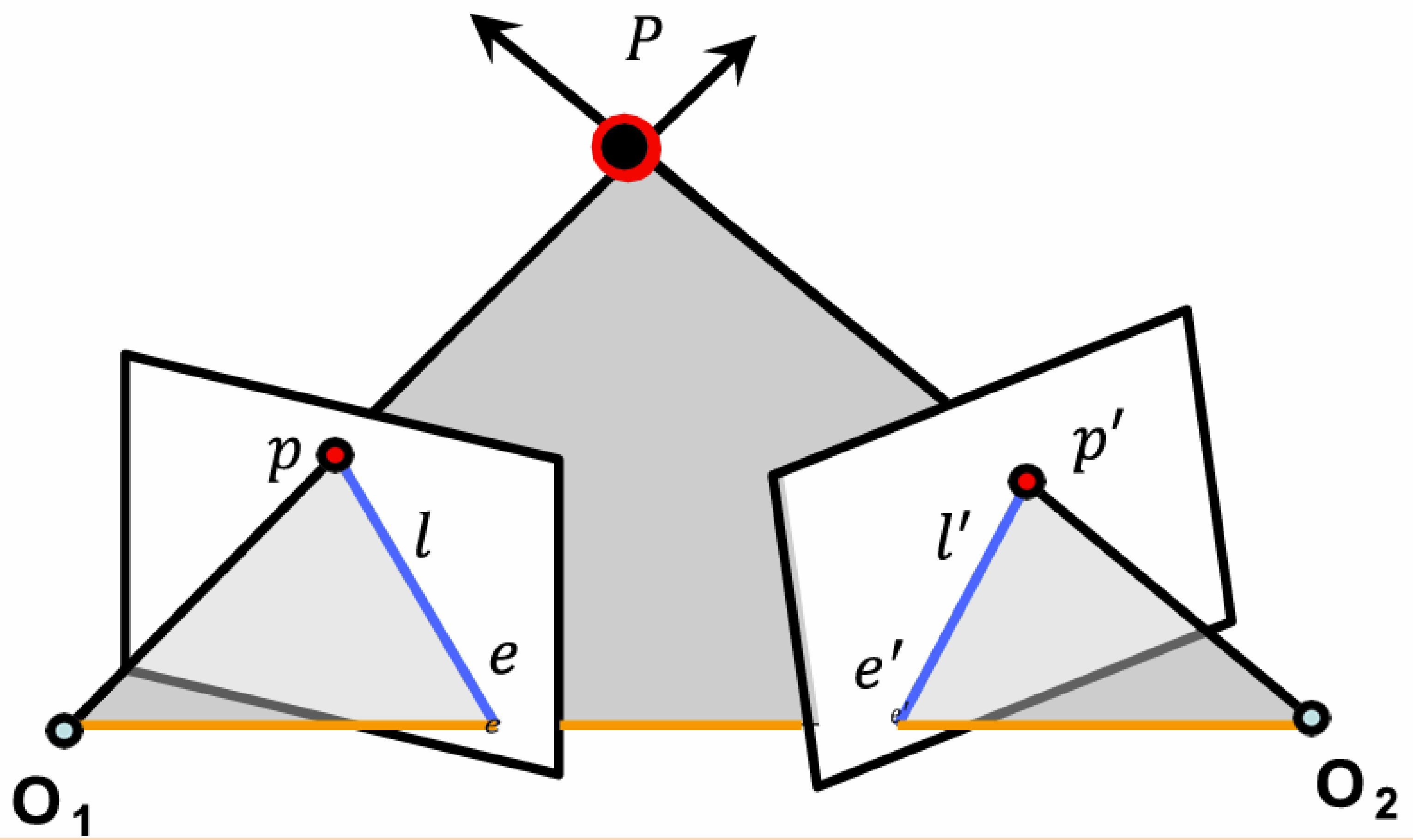


$$p_c'^T E p_c = p_c'^T [T_\times] R p_c = \left(K'^{-1}p'\right)^T \cdot [T_\times] R K^{-1} p = p'^T \boxed{K'^{-T}[T_\times]RK^{-1}}\, p = 0$$

$$\Downarrow$$

$$\boxed{p'^T F p = 0}$$

极线约束 – 基础矩阵



- p 对应的极线是 l'（$l' = Fp$）
- p'对应的极线是 l（$l = F^T p'$）
- $Fe = 0$ 与 $e'^T F = 0$
- F 是奇异的(秩为2)
- F 7个自由度（尺度无法确定，$\det(F) = 0$）

$$\boxed{p'^T F p = 0}$$

$$F = K'^{-T}[T_\times]RK^{-1}$$

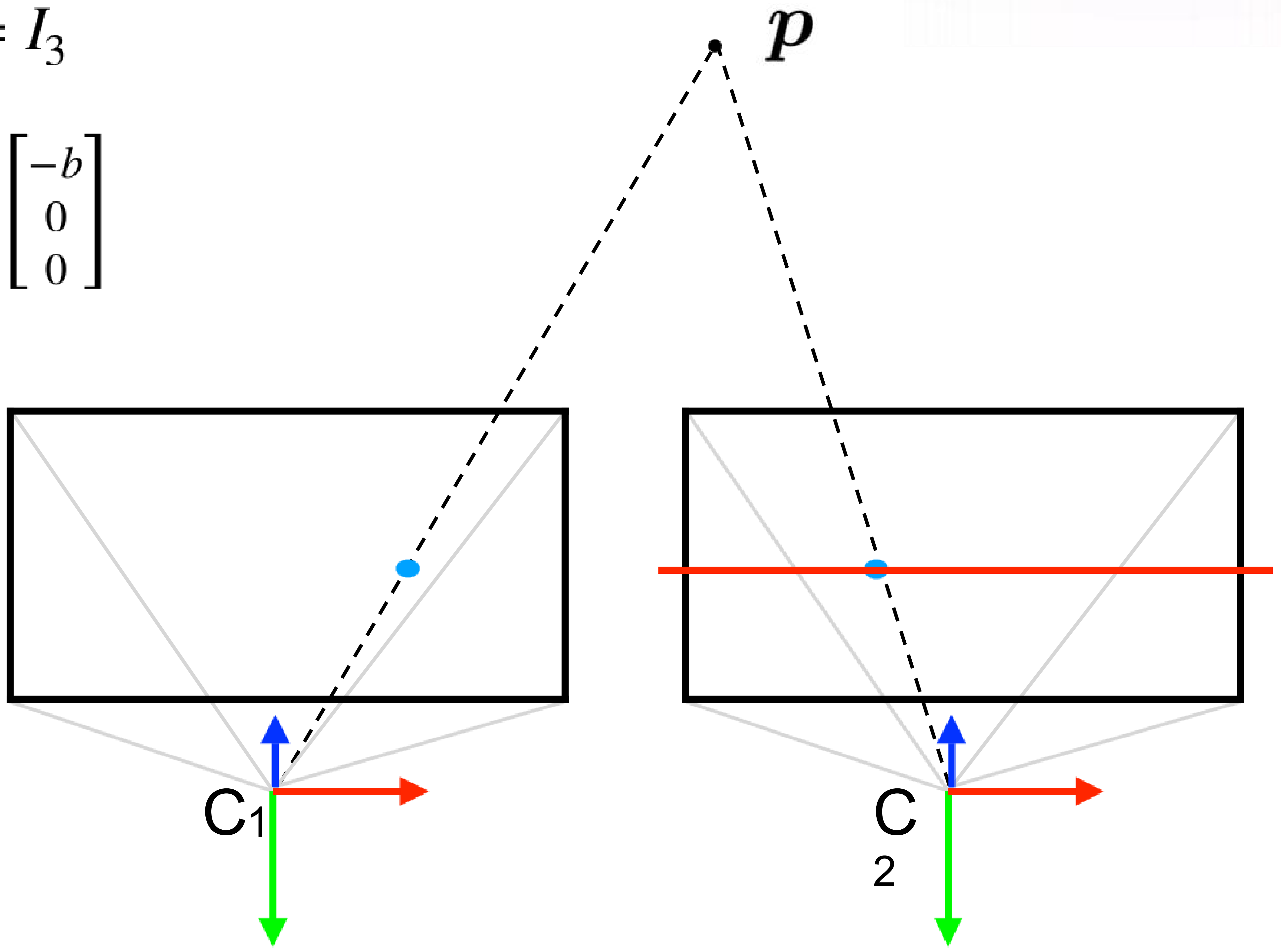
如何求解基础矩阵呢?

回顾

特例: Stereo Camera

$$R_{C_1}^{C_2} = I_3$$

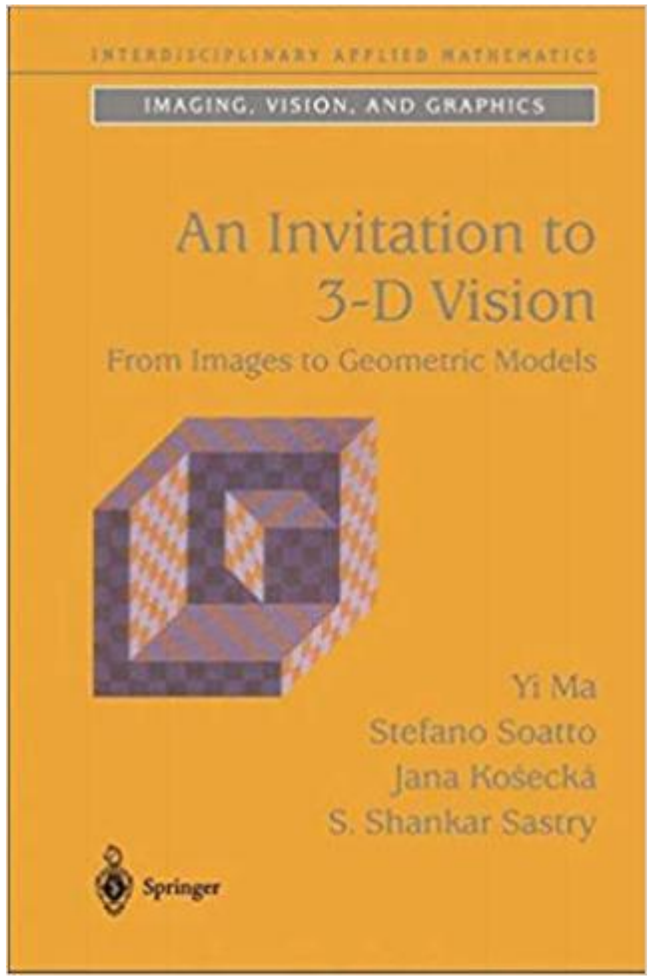
$$t_{C_1}^{C_2} = \begin{bmatrix} -b \\ 0 \\ 0 \end{bmatrix}$$



also: easy to triangulate points given geometry

Today

- 2-view geometry
- RANSAC
- 3D-3D correspondences


INTERDISCIPLINARY APPLIED MATHEMATICS

IMAGING, VISION, AND GRAPHICS

An Invitation to 3-D Vision

From Images to Geometric Models

Yi Ma

Stefano Soatto

Jana Košecká

S. Shankar Sastry

Springer

Chapter 5

Reconstruction from Two Calibrated Views

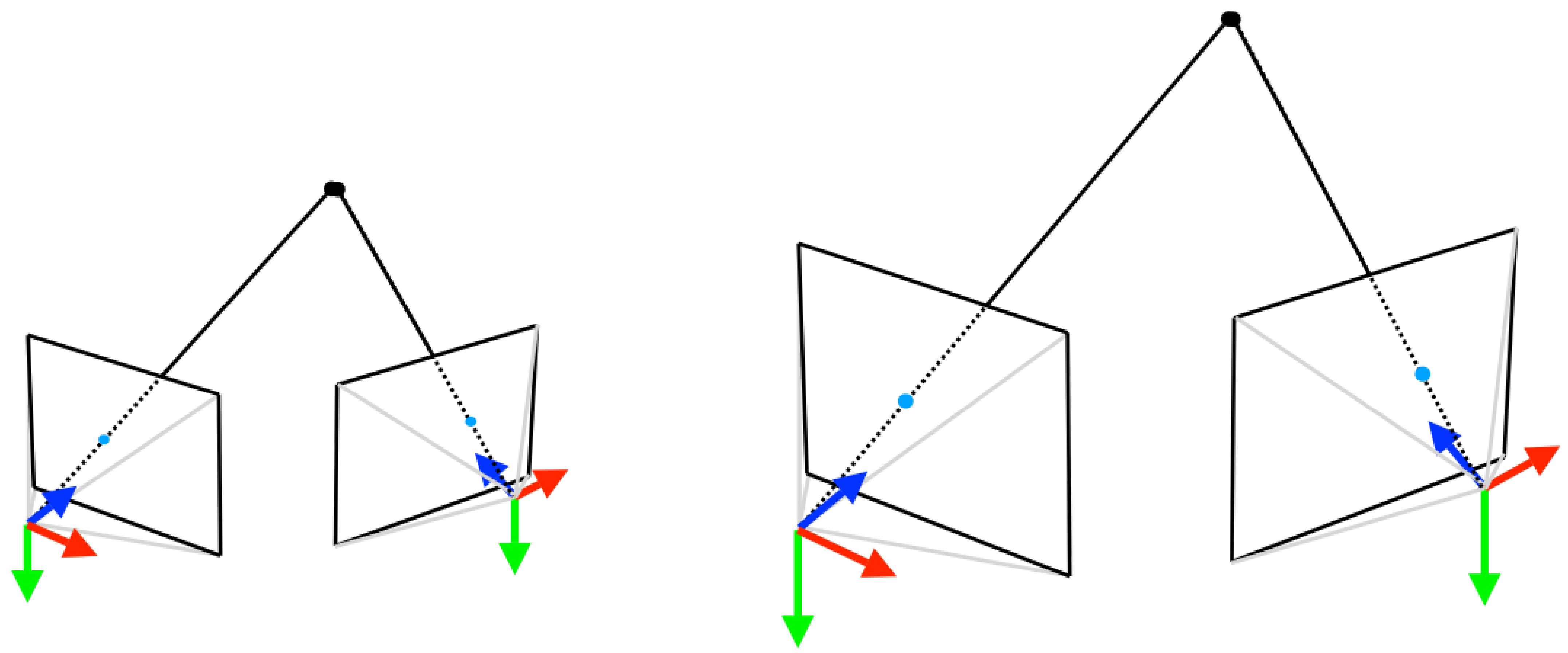
Estimating Poses from Correspondences

（1）已知2D图像到2D图像之间的关系，求解图像帧间位姿

Given *N* calibrated pixel correspondences:

$$(\tilde{\boldsymbol{y}}_{1,k}, \tilde{\boldsymbol{y}}_{2,k}) \text{ for } k = 1, \ldots, N$$

compute the relative pose between the cameras



Can we estimate the scale of the translation (baseline)?

Estimating Poses from Correspondences

（1）已知2D图像到2D图像之间的关系，求解图像帧间位姿

Given *N* calibrated pixel correspondences:

$$(\tilde{\boldsymbol{y}}_{1,k}, \tilde{\boldsymbol{y}}_{2,k}) \text{ for } k = 1, \ldots, N$$

1. leverage the epipolar constraints to estimate the essential matrix ***E***

$$\tilde{\boldsymbol{y}}_{2,k}^{\top} \boldsymbol{E} \, \tilde{\boldsymbol{y}}_{1,k} = 0$$

2. Retrieve the rotation and translation (up to scale) from the ***E***

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

Retrieving Pose from Essential Matrix

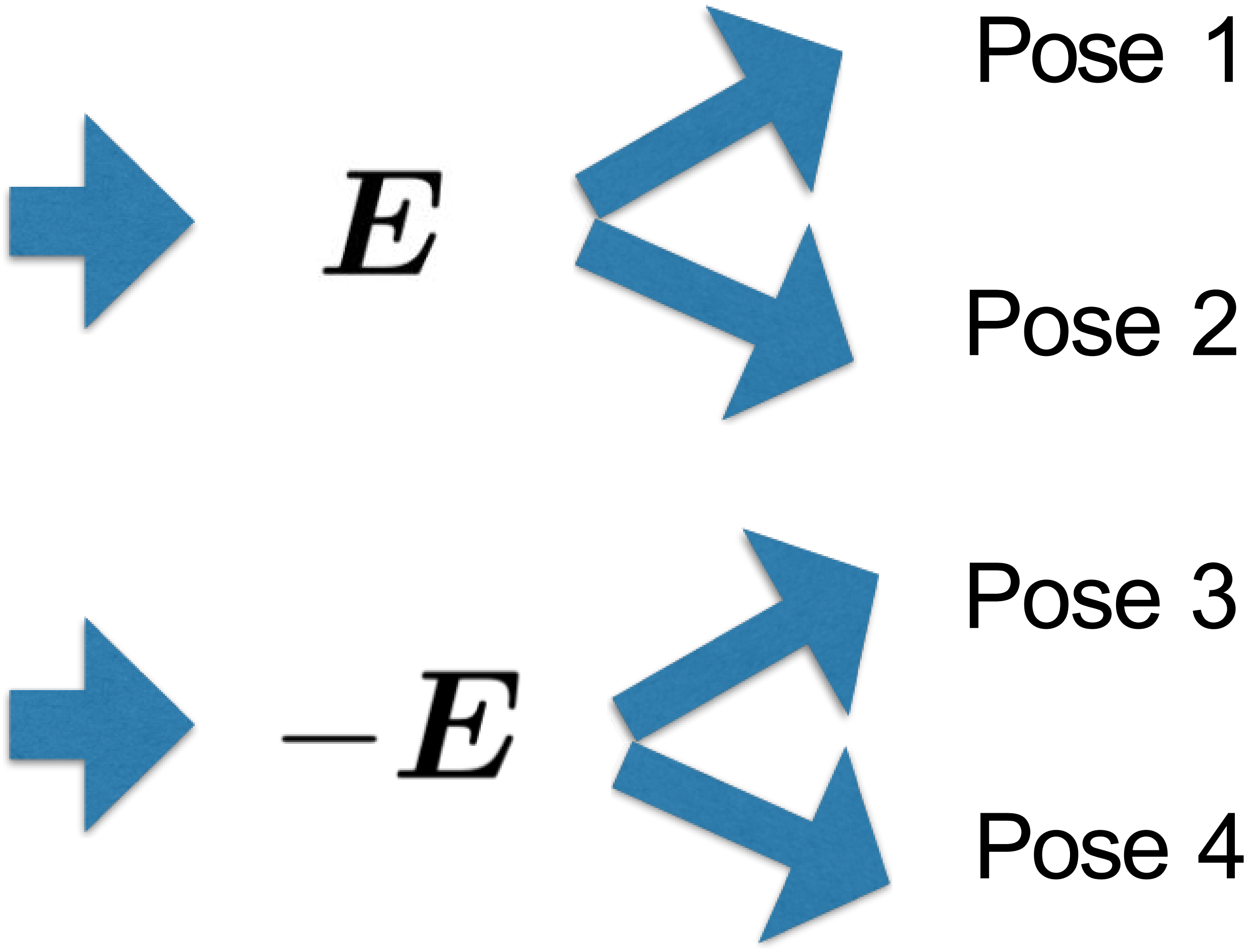
(1) 已知2D图像到2D图像之间的关系，求解图像帧间位姿

Theorem 1 (Pose recovery from essential matrix, Thm 5.7 in [1]). *There exist exactly two relative poses* $(\boldsymbol{R}, \boldsymbol{t})$ *with* $\boldsymbol{R} \in \mathrm{SO}(3)$ *and* $\boldsymbol{t} \in \mathbb{R}^3$ *corresponding to a nonzero essential matrix* $\boldsymbol{E}$ *(i.e., such that* $\boldsymbol{E} = [\mathbf{t}]_\times \boldsymbol{R}$*):*

$$\boldsymbol{t}_1 = \boldsymbol{U}\boldsymbol{R}_z(+\pi/2)\boldsymbol{\Sigma}\boldsymbol{U}^\mathsf{T} \qquad \boldsymbol{R}_1 = \boldsymbol{U}\boldsymbol{R}_z(+\pi/2)\boldsymbol{V}^\mathsf{T} \tag{13.19}$$

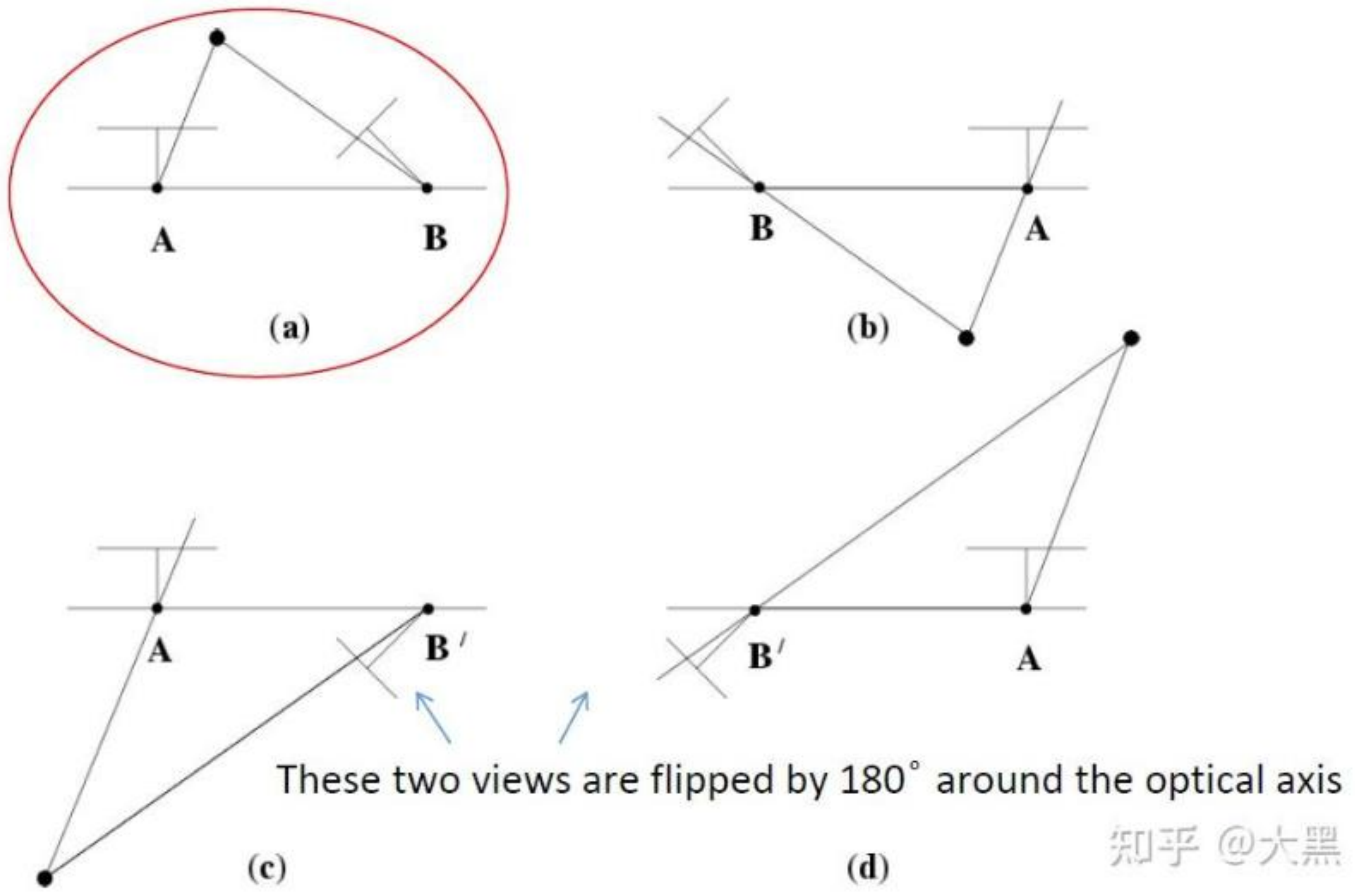
$$\boldsymbol{t}_2 = \boldsymbol{U}\boldsymbol{R}_z(-\pi/2)\boldsymbol{\Sigma}\boldsymbol{U}^\mathsf{T} \qquad \boldsymbol{R}_2 = \boldsymbol{U}\boldsymbol{R}_z(-\pi/2)\boldsymbol{V}^\mathsf{T} \tag{13.20}$$

where $\boldsymbol{E} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\mathsf{T}$ *is the singular value decomposition of the matrix* $\boldsymbol{E}$*, and* $\boldsymbol{R}_z(+\pi/2)$ *is an elementary rotation around the z-axis of an angle* $\pi/2$.

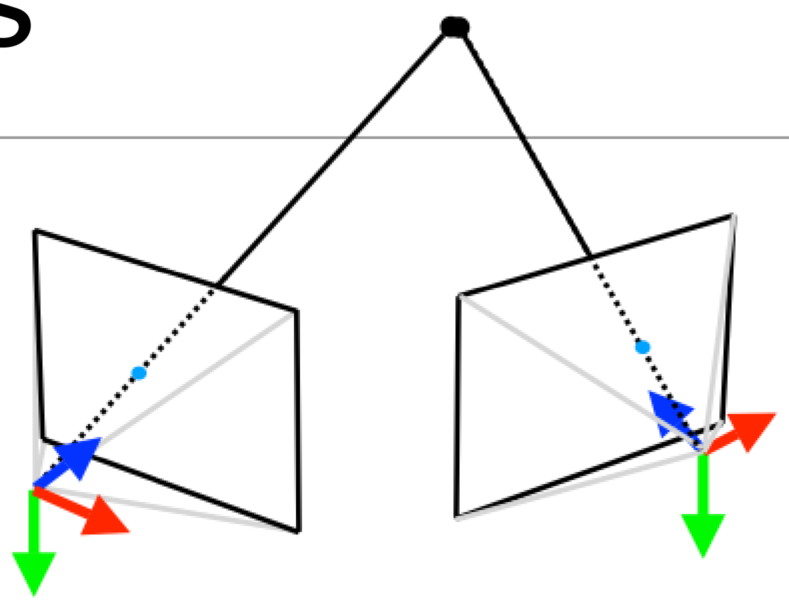


Cheirality constraints

Points must be in front of the cameras!



8-point method: Limitations

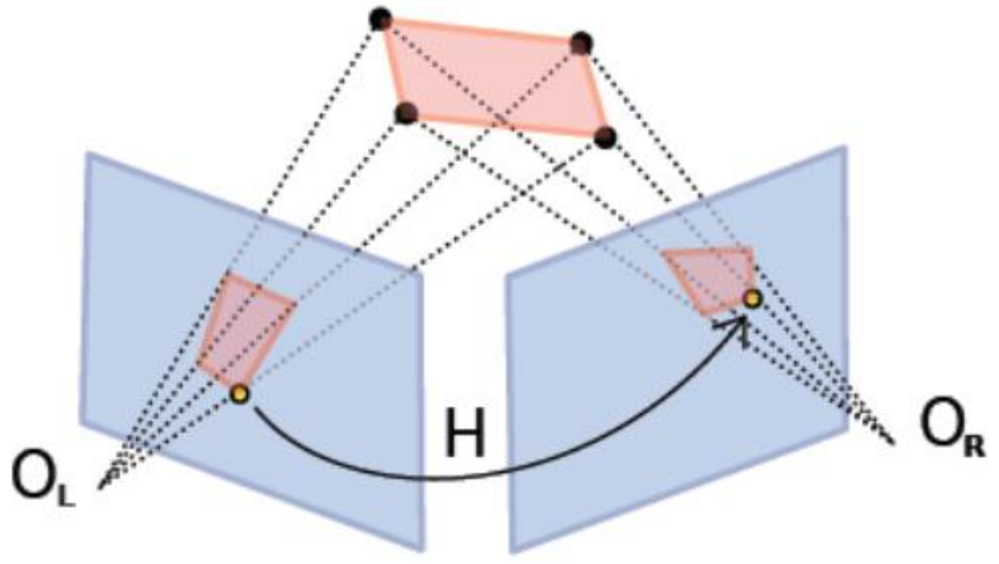


$$E = [t]_\times R$$

Number of correspondences: do we really need 8 points?

也可以是5个以上点，但是8个点效果更好！

Scene structures: there are certain configurations of 3D points that make the algorithm fail



匹配3D点共面 或者 纯旋转，得到单应矩阵！

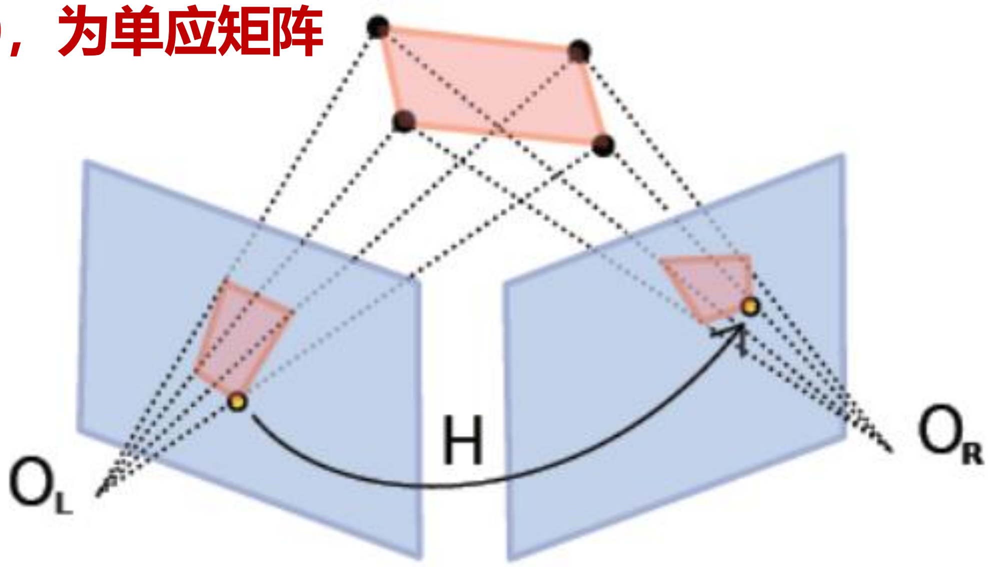
Parallax: what if $\boldsymbol{t} = \mathbf{0}$?

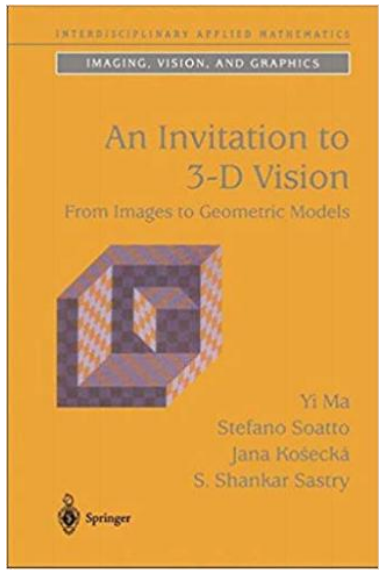
Other Matrices in 2-view Geometry

Homography matrix **H**

$$\lambda_2 \boldsymbol{x}_2 = H \lambda_1 \boldsymbol{x}_1$$

匹配点共面 或者 t=0，为单应矩阵

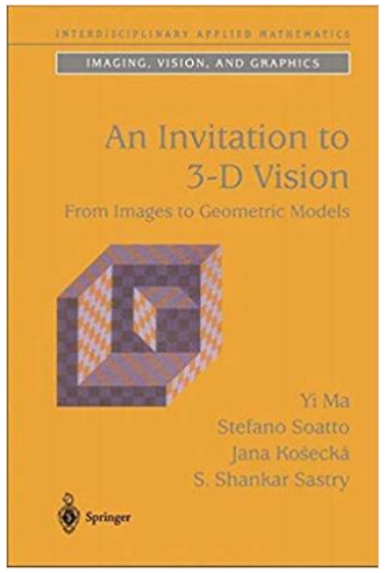



INTERDISCIPLINARY APPLIED MATHEMATICS
IMAGING, VISION, AND GRAPHICS
An Invitation to 3-D Vision
From Images to Geometric Models
Yi Ma
Stefano Soatto
Jana Košecká
S. Shankar Sastry
Springer

Section 5.3

Fundamental matrix **F**

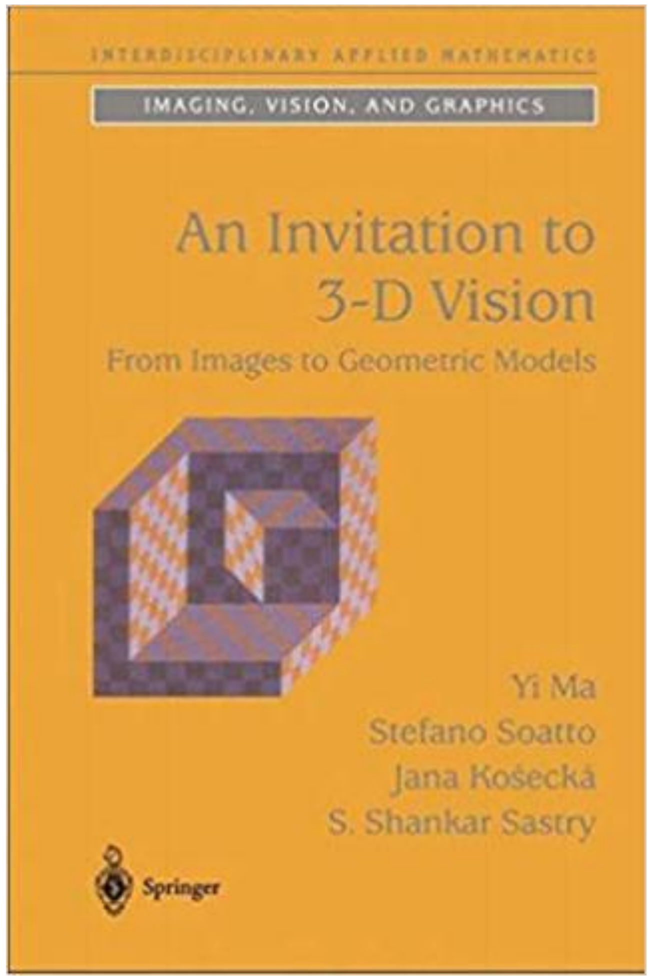
$$\boldsymbol{F} = \boldsymbol{K}_2^{-\top} \; [\boldsymbol{t}]_\times \boldsymbol{R} \; \boldsymbol{K}_1^{-1}$$


INTERDISCIPLINARY APPLIED MATHEMATICS
IMAGING, VISION, AND GRAPHICS
An Invitation to 3-D Vision
From Images to Geometric Models
Yi Ma
Stefano Soatto
Jana Košecká
S. Shankar Sastry
Springer

Chapter 6

Today

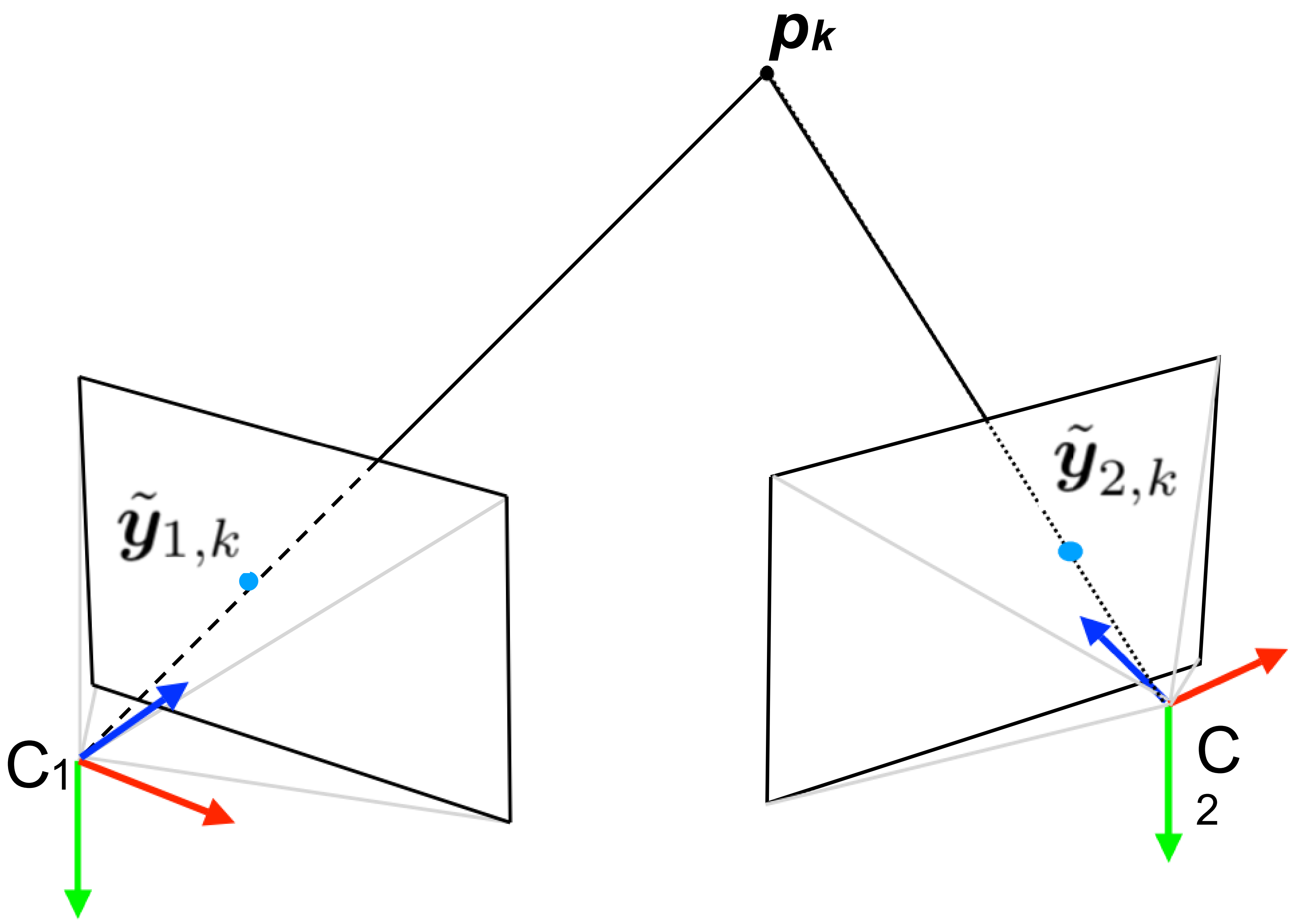
- 2-view geometry
- RANSAC
- 3D-3D correspondences


INTERDISCIPLINARY APPLIED MATHEMATICS
IMAGING, VISION, AND GRAPHICS
An Invitation to 3-D Vision
From Images to Geometric Models
Yi Ma
Stefano Soatto
Jana Košecká
S. Shankar Sastry
Springer

Chapter 5

Reconstruction from Two Calibrated Views

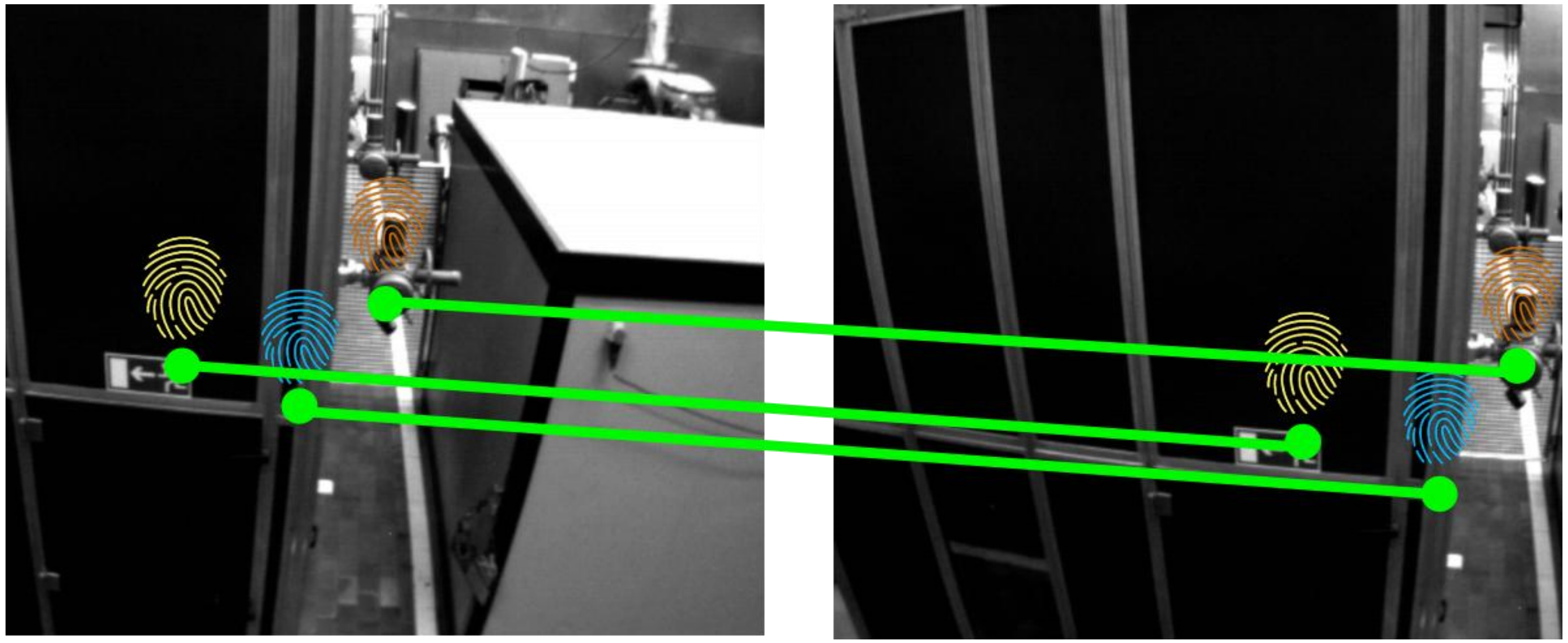
2-view Geometry



$$\tilde{y}_{2,k}^{\top} \, E \, \tilde{y}_{1,k} = 0$$

Essential matrix encodes relative pose (up to scale) between C_1 and C_2

2-view Geometry



Last week's assumptions:

- no wrong correspondences (outliers)
- 3D point is not moving
- camera calibration is known

Estimating Poses from Correspondences

Given *N* calibrated pixel correspondences:

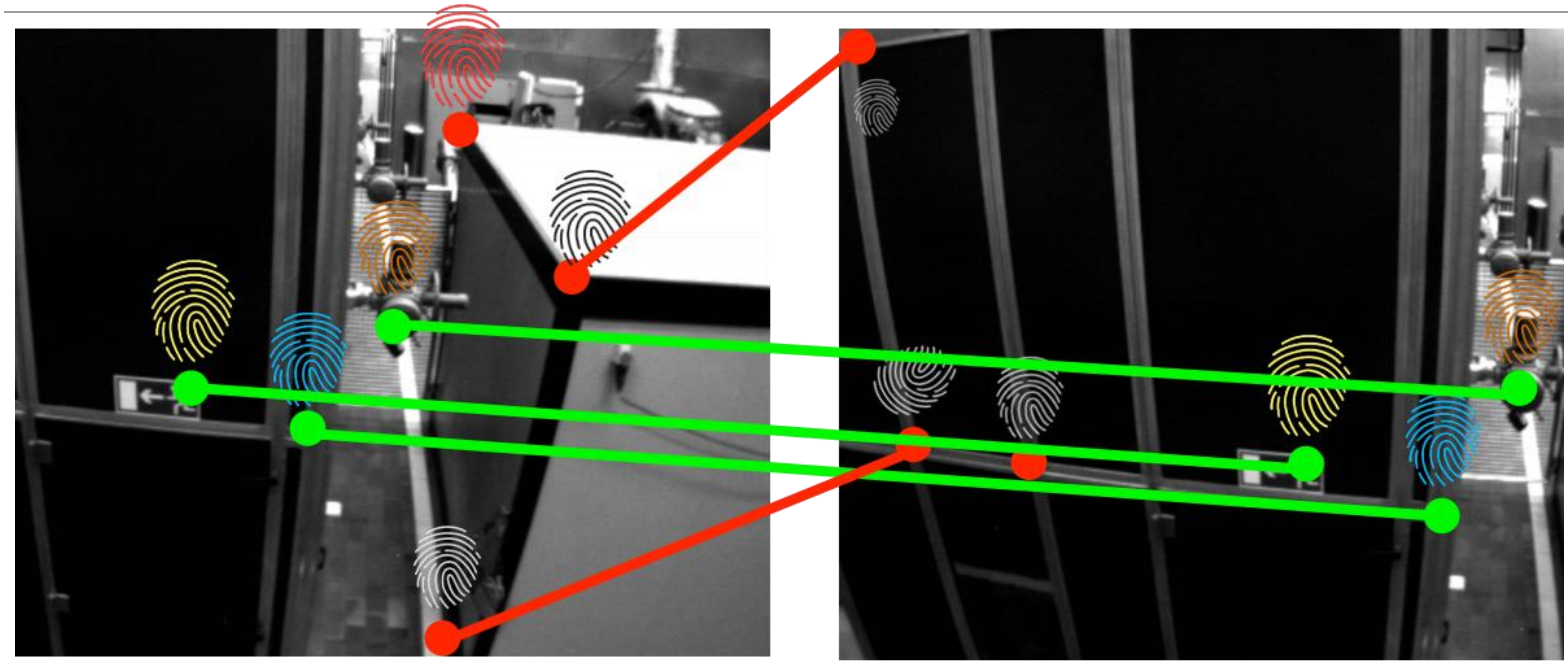
$$(\tilde{\boldsymbol{y}}_{1,k}, \tilde{\boldsymbol{y}}_{2,k}) \text{ for } k = 1, \ldots, N$$

1. leverage the epipolar constraints to estimate the essential matrix ***E*** $\tilde{\boldsymbol{y}}_{2,k}^{\mathsf{T}} \, \boldsymbol{E} \, \tilde{\boldsymbol{y}}_{1,k} = 0$

For 8 points: $\boldsymbol{A}\boldsymbol{e} = 0$ N>8 points: $\arg\min_{\|\boldsymbol{e}\|=1} \|\boldsymbol{A}\boldsymbol{e}\|^2$

2. Retrieve the rotation and translation (up to scale) from the ***E*** $\boldsymbol{E} = [\boldsymbol{t}]_{\times}\boldsymbol{R}$

2-view Geometry



In practice:

- Many wrong correspondences (outliers)
- Some 3D points might be moving

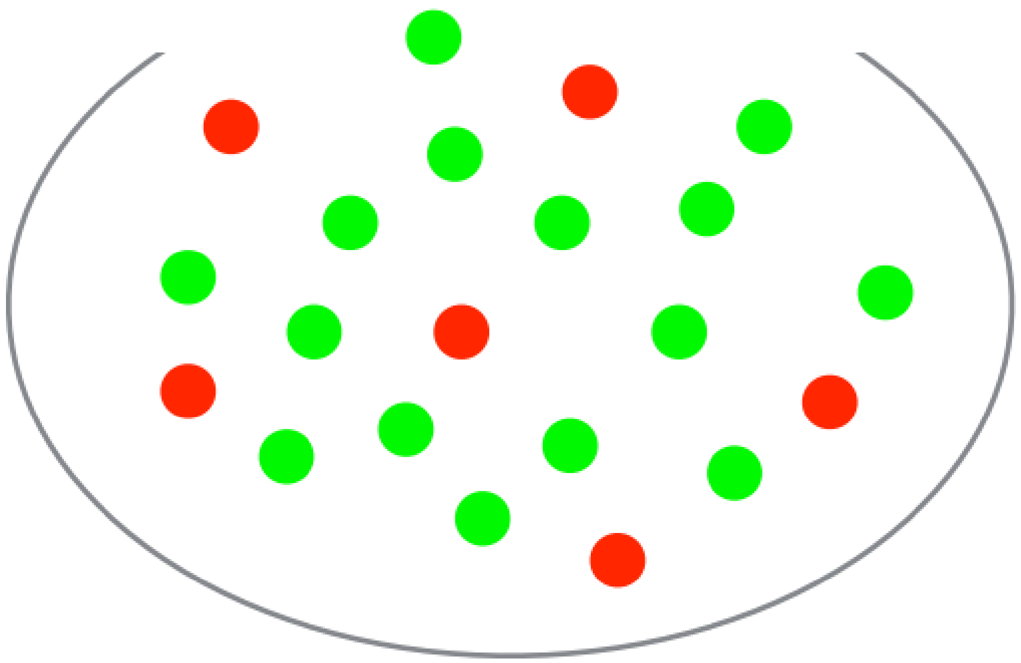
存在错误的匹配点，或者存在动态物体的匹配点；均为外点!

RANSAC

随机采样一致性 策略

RANdom **SA**mple **C**onsensus

Problem: estimate model *P* from N data points, possibly corrupted with outliers.



Assume: we have an algorithm to estimate *P* from *n* data points (n << N)

Basic idea:

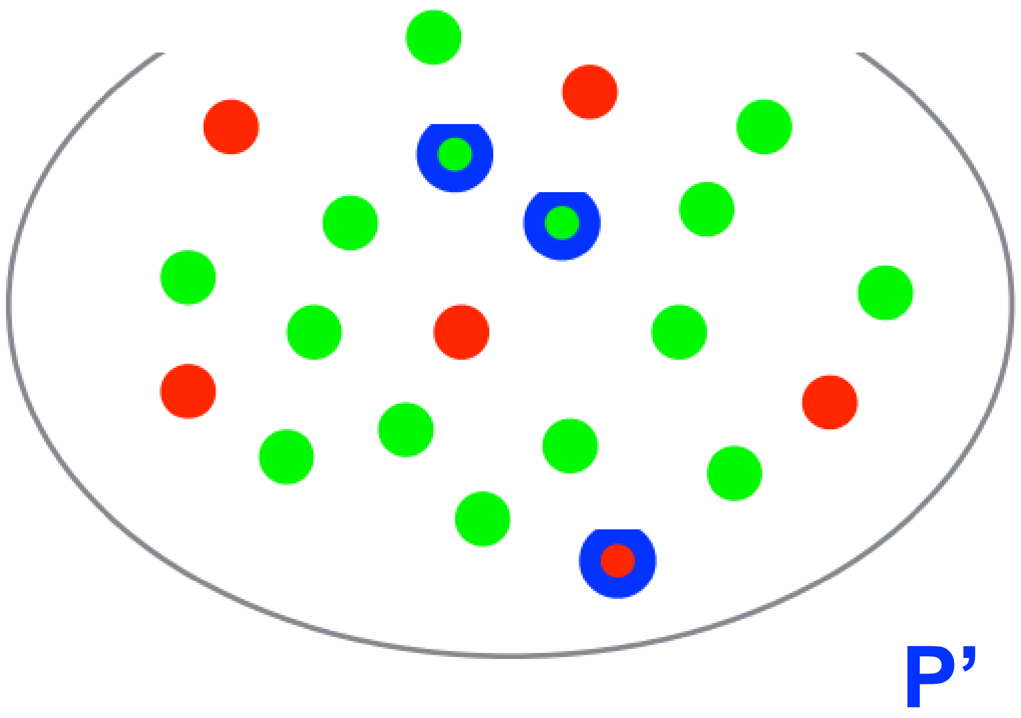
1. sample *n* points
2. compute an estimate *P'* of *P*
3. count how many other points agree with *P'*
4. repeat until you get a *P'* that agrees with many points

RANSAC 随机采样一致性 策略

RANdom **SA**mple **C**onsensus

Problem: estimate model *P* from N data points, possibly corrupted with outliers.

Assume: we have an algorithm to estimate *P* from *n* data points (n << N)



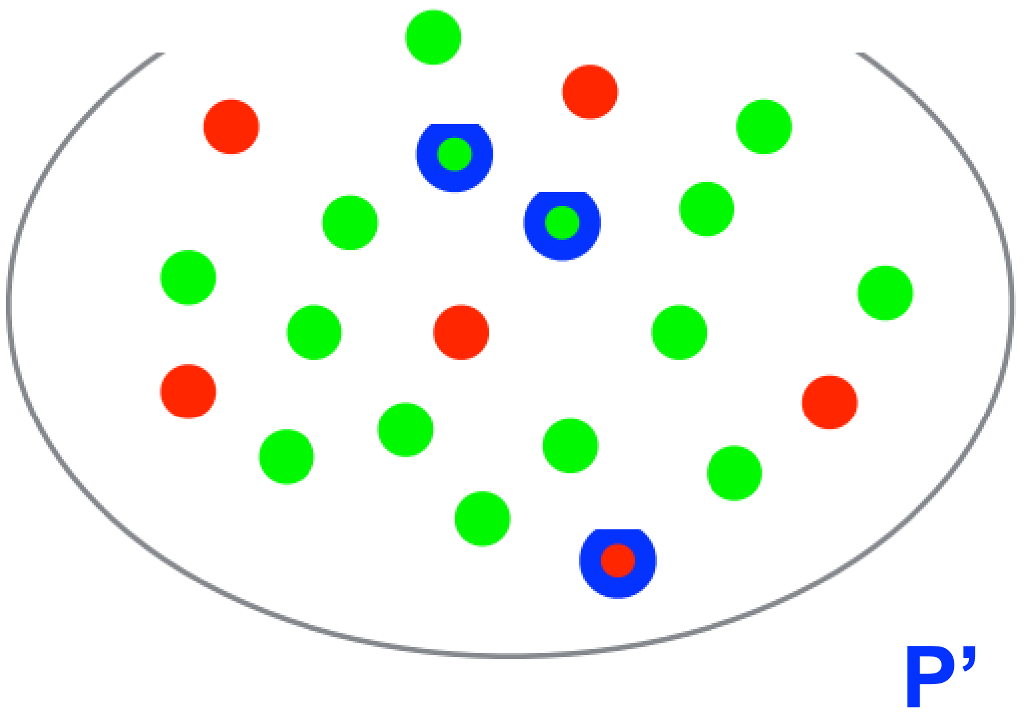
Basic idea:

1. sample *n* points
2. compute an estimate *P’* of *P*
3. count how many other points agree with *P’*
4. repeat until you get a *P’* that agrees with many points

RANSAC 随机采样一致性 策略

RANdom **SA**mple **C**onsensus

Problem: estimate model *P* from N data points, possibly corrupted with outliers.



Assume: we have an algorithm to estimate *P* from *n* data points (n << N)

Basic idea:

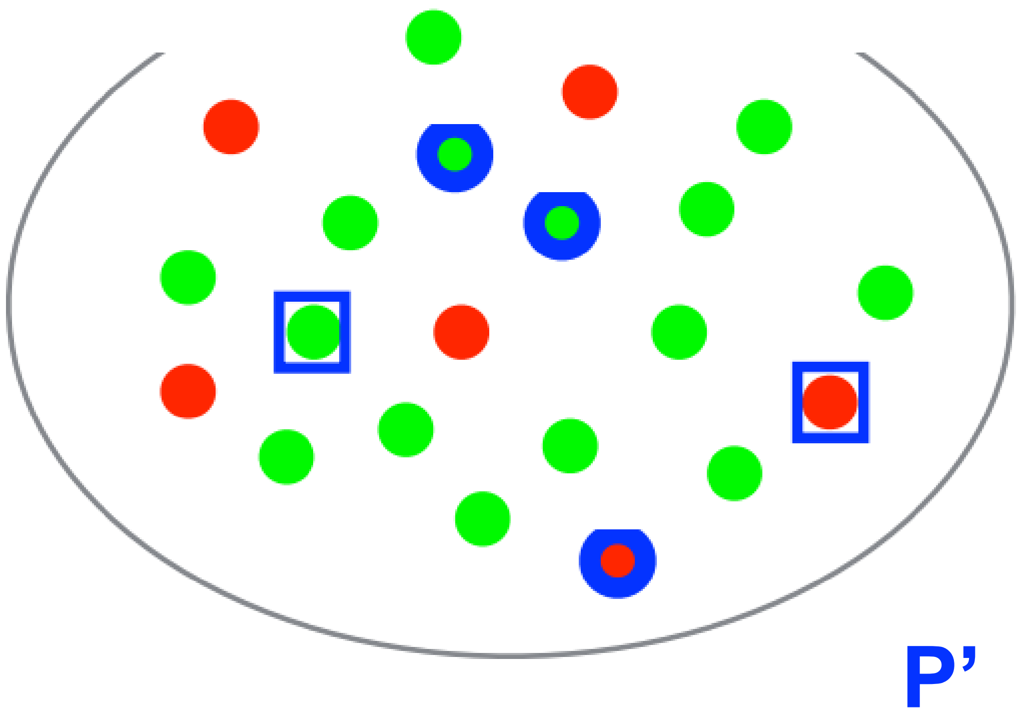
1. sample *n* points
2. compute an estimate *P’* of *P*
3. count how many other points agree with *P’*
4. repeat until you get a *P’* that agrees with many points

RANSAC

RANdom **SA**mple **C**onsensus

Problem: estimate model *P* from N data points, possibly corrupted with outliers.

Assume: we have an algorithm to estimate *P* from *n* data points (n << N)



Basic idea:

1. sample *n* points
2. compute an estimate *P'* of *P*
3. count how many other points agree with *P'*
4. repeat until you get a *P'* that agrees with many points

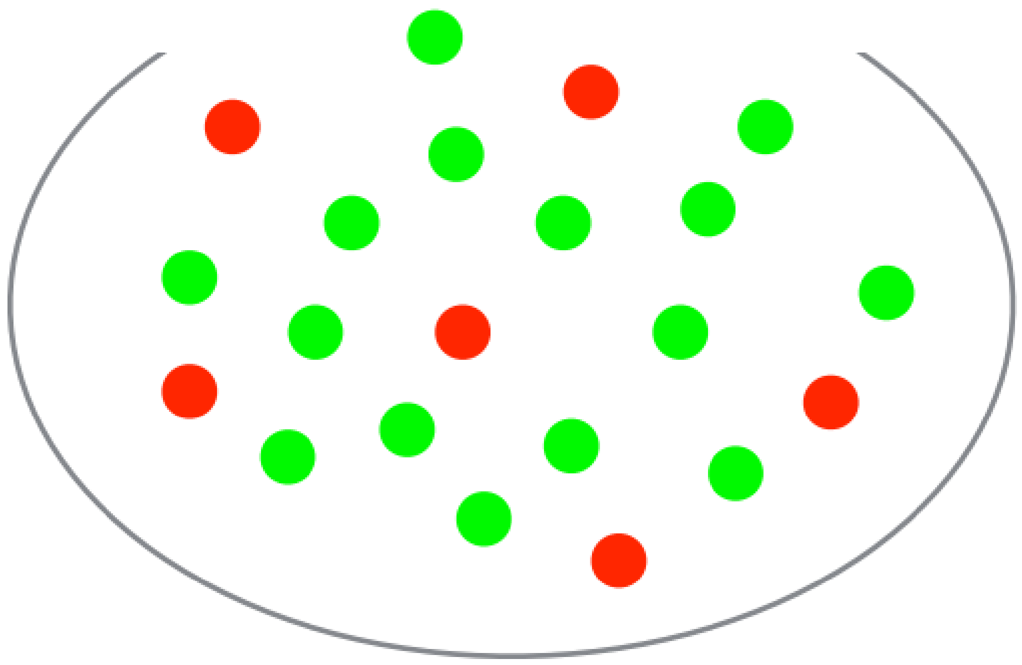
Consensus Set

RANSAC

RANdom **SA**mple **C**onsensus

Problem: estimate model *P* from N data points, possibly corrupted with outliers.

Assume: we have an algorithm to estimate *P* from *n* data points (n << N)



Basic idea:

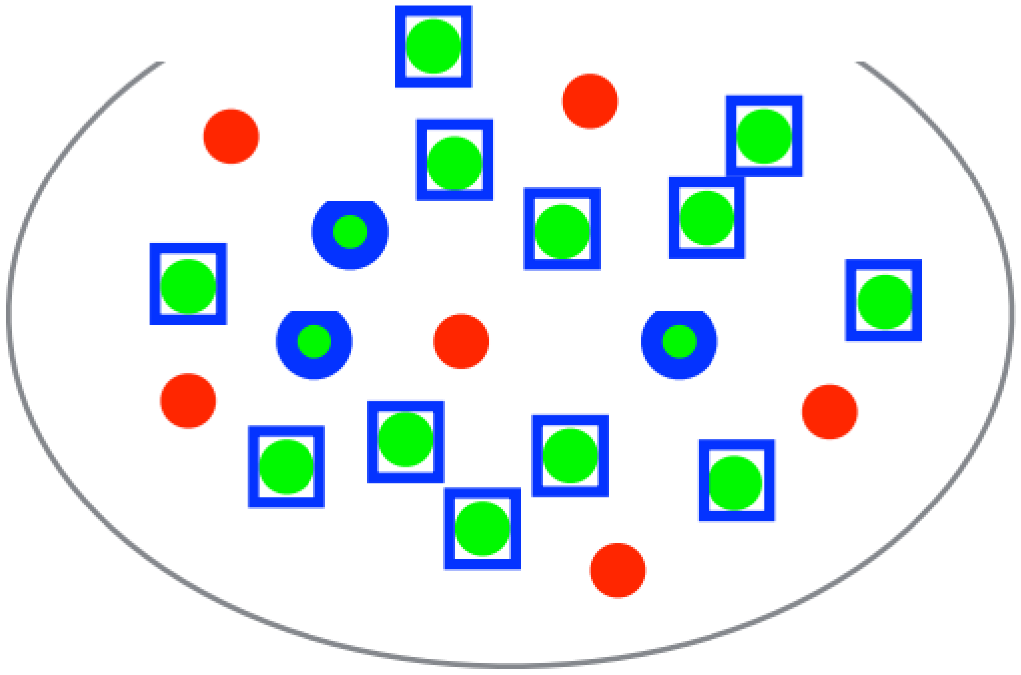
1. sample *n* points
2. compute an estimate *P’* of *P*
3. count how many other points agree with *P’*
4. repeat until you get a *P’* that agrees with many points

RANSAC

RANdom **SA**mple **C**onsensus

Problem: estimate model *P* from N data points, possibly corrupted with outliers.

Assume: we have an algorithm to estimate *P* from *n* data points (n << N)



P'

Consensus Set

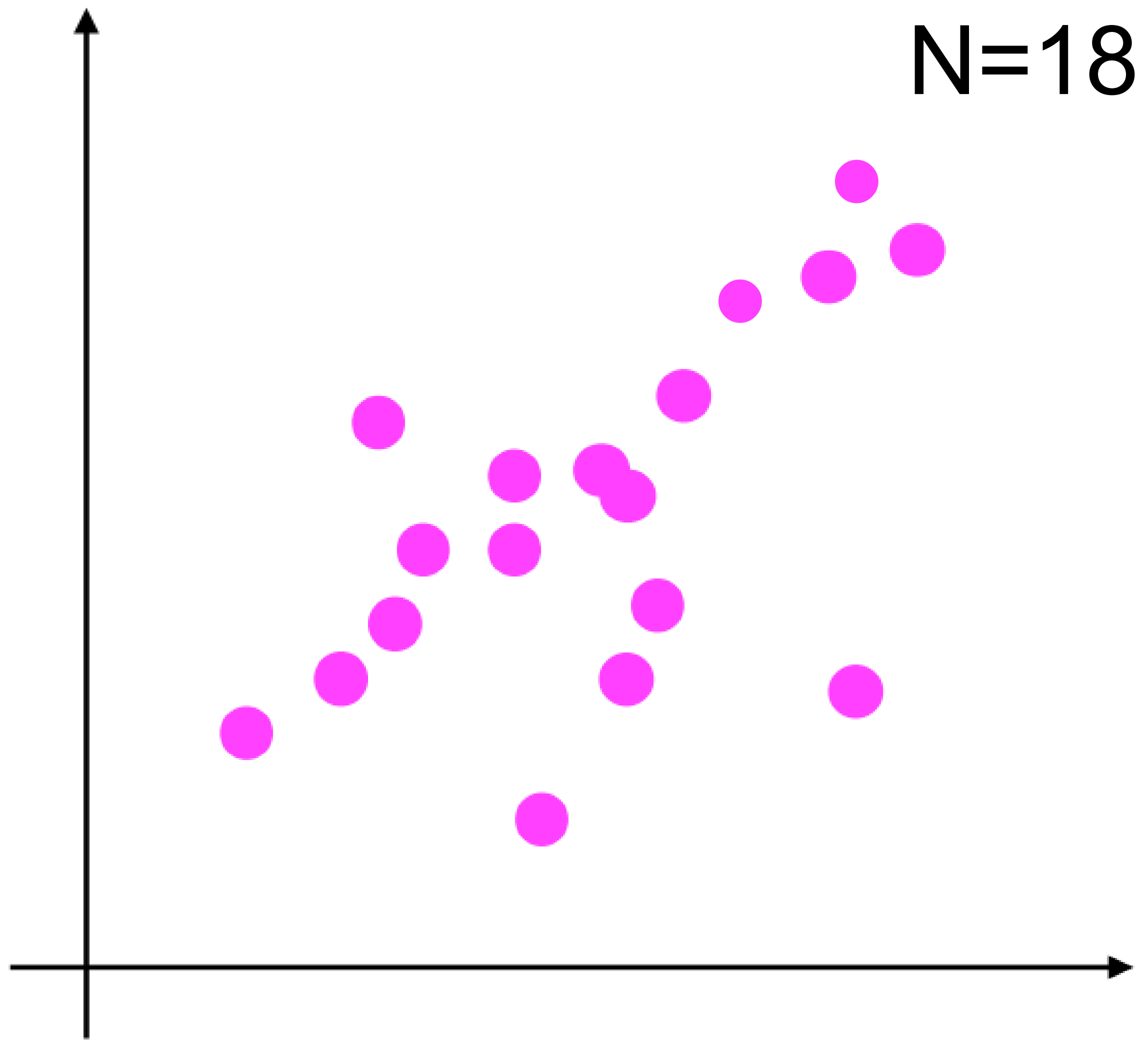
Basic idea:

1. sample *n* points
2. compute an estimate *P'* of *P*
3. count how many other points agree with *P'*
4. repeat until you get a *P'* that agrees with many points

Example: Linear Regression

Fit a line through
N 2D points, possibly
corrupted with outliers.

Note: we have an algorithm
to estimate a line from n=2 points



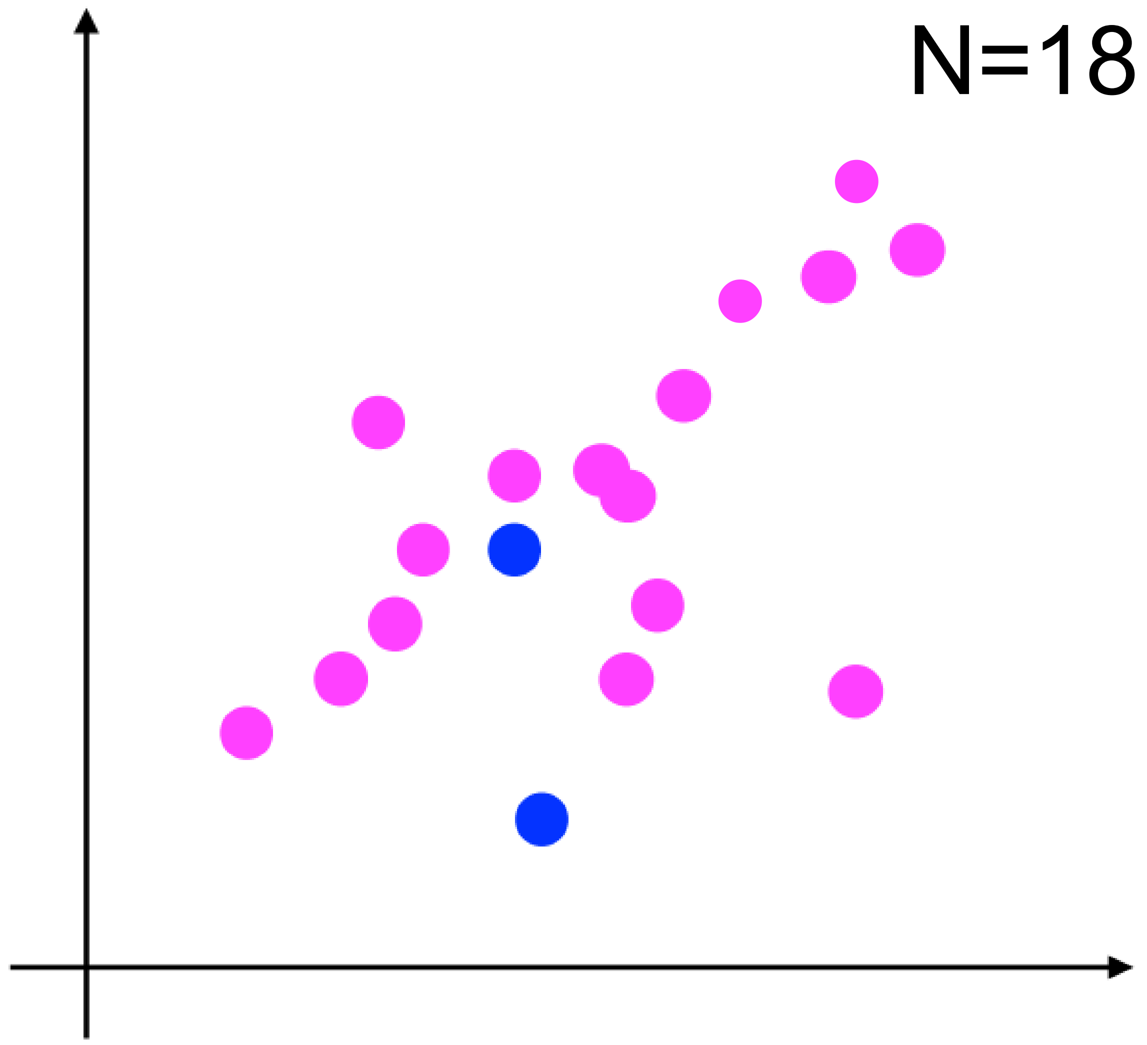
RANSAC:

1. sample 2 points
2. compute a line estimate *P’* of *P*
3. count how many points are within a **tolerance** from *P’*
4. repeat until you get a *P’* that agrees with many points

Example: Linear Regression

Fit a line through
N 2D points, possibly
corrupted with outliers.

Note: we have an algorithm
to estimate a line from n=2 points



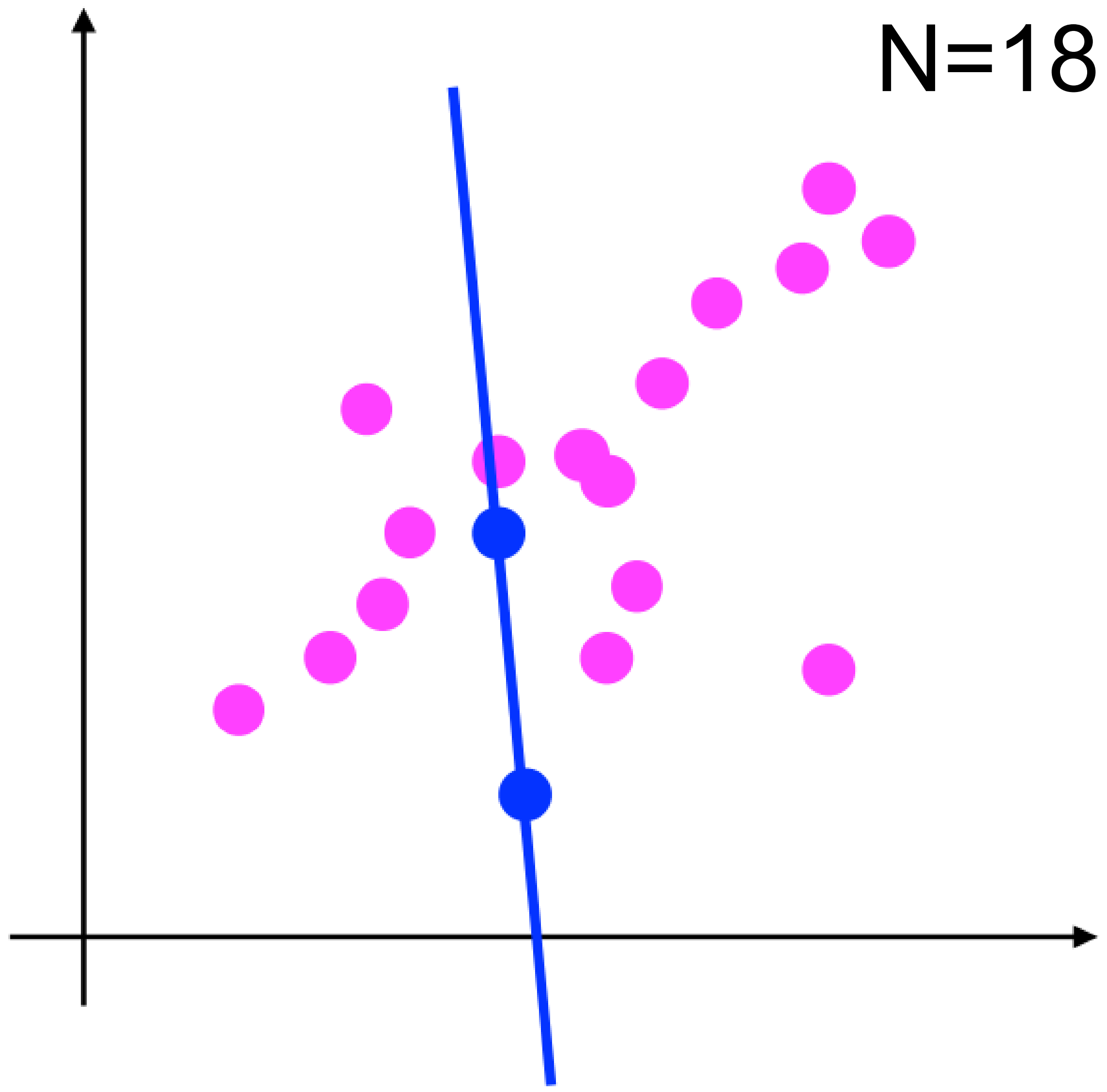
RANSAC:

1. sample 2 points
2. compute a line estimate *P'* of *P*
3. count how many points are within a **tolerance** from *P'*
4. repeat until you get a *P'* that agrees with many points

Example: Linear Regression

Fit a line through
N 2D points, possibly
corrupted with outliers.

Note: we have an algorithm
to estimate a line from n=2 points



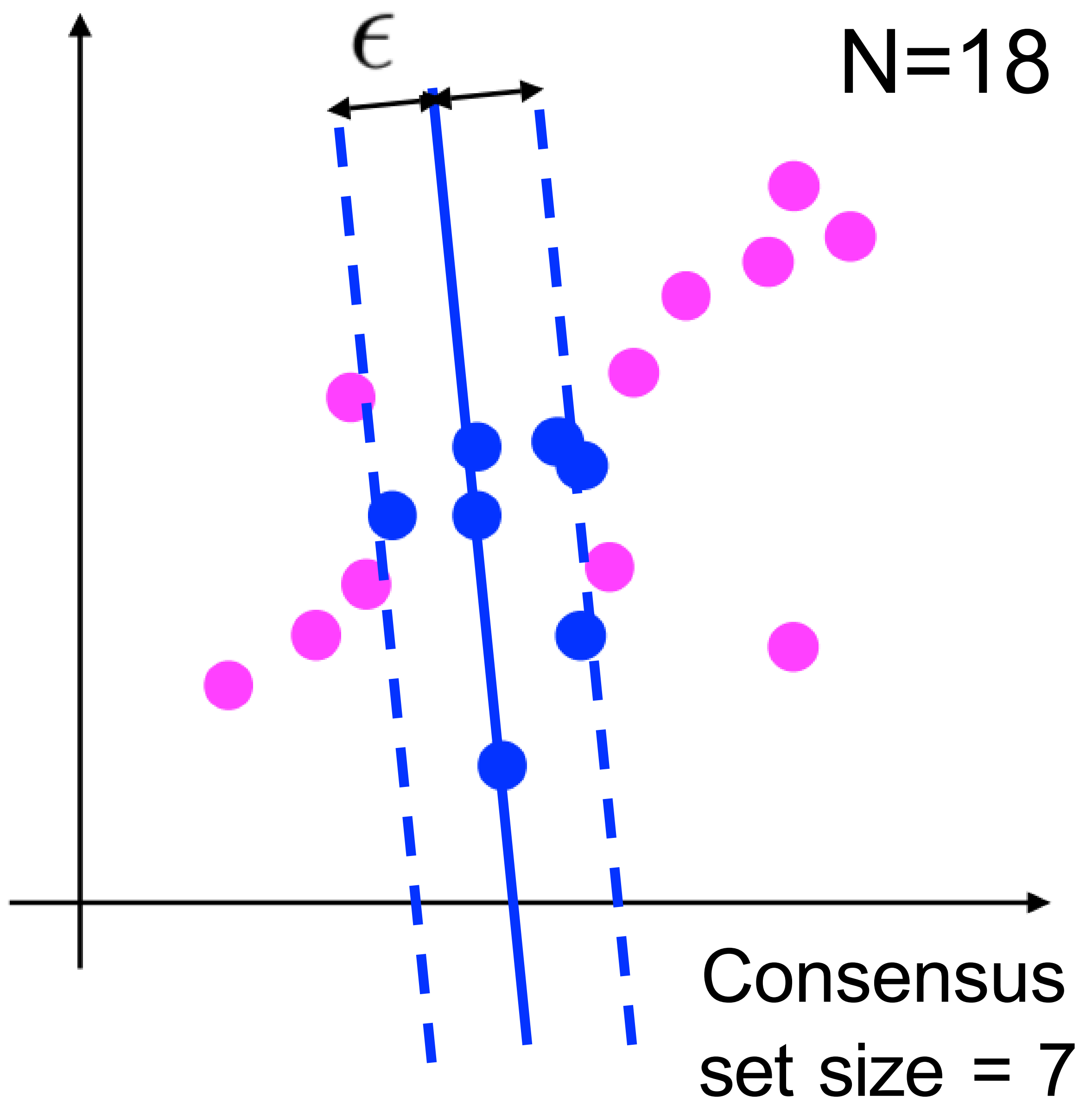
RANSAC:

1. sample 2 points
2. compute a line estimate *P'* of *P*
3. count how many points are within a **tolerance** from *P'*
4. repeat until you get a *P'* that agrees with many points

Example: Linear Regression

Fit a line through
N 2D points, possibly
corrupted with outliers.

Note: we have an algorithm
to estimate a line from n=2 points



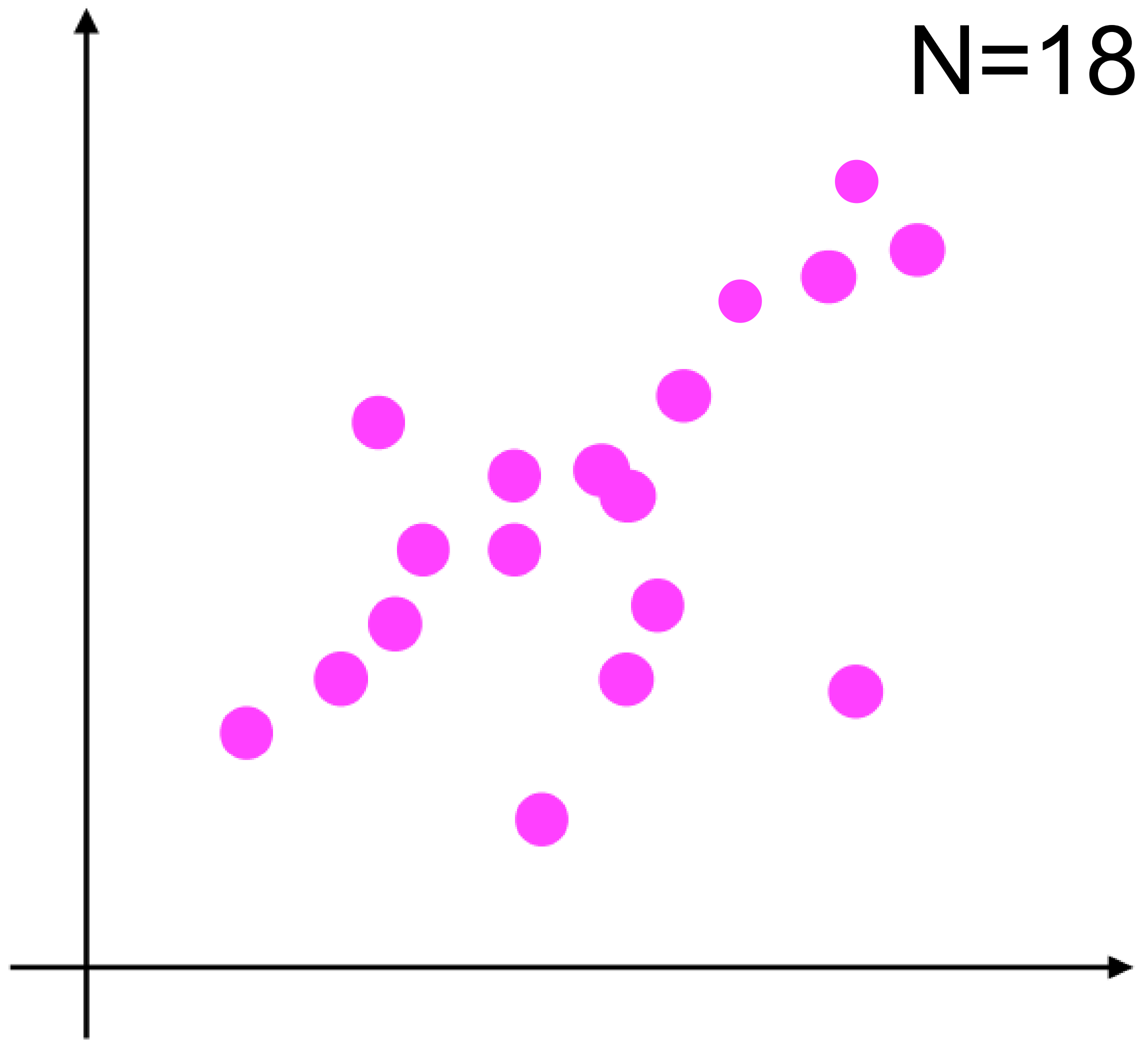
RANSAC:

1. sample 2 points
2. compute a line estimate *P'* of *P*
3. count how many points are within a **tolerance** from *P'*
4. repeat until you get a *P'* that agrees with many points

Example: Linear Regression

Fit a line through
N 2D points, possibly
corrupted with outliers.

Note: we have an algorithm
to estimate a line from n=2 points



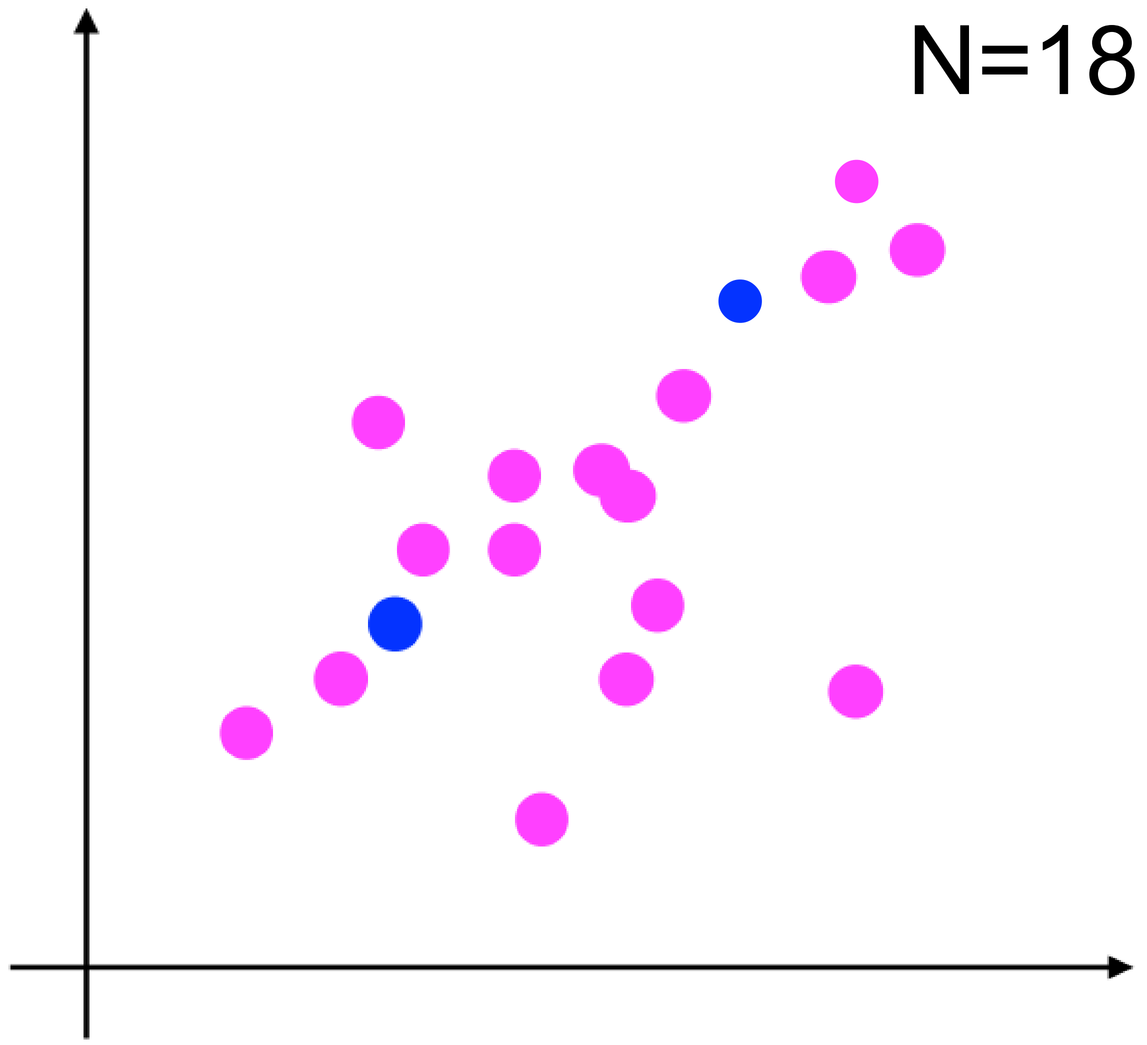
RANSAC:

1. sample 2 points
2. compute a line estimate *P'* of *P*
3. count how many points are within a **tolerance** from *P'*
4. repeat until you get a *P'* that agrees with many points

Example: Linear Regression

Fit a line through
N 2D points, possibly
corrupted with outliers.

Note: we have an algorithm
to estimate a line from n=2 points



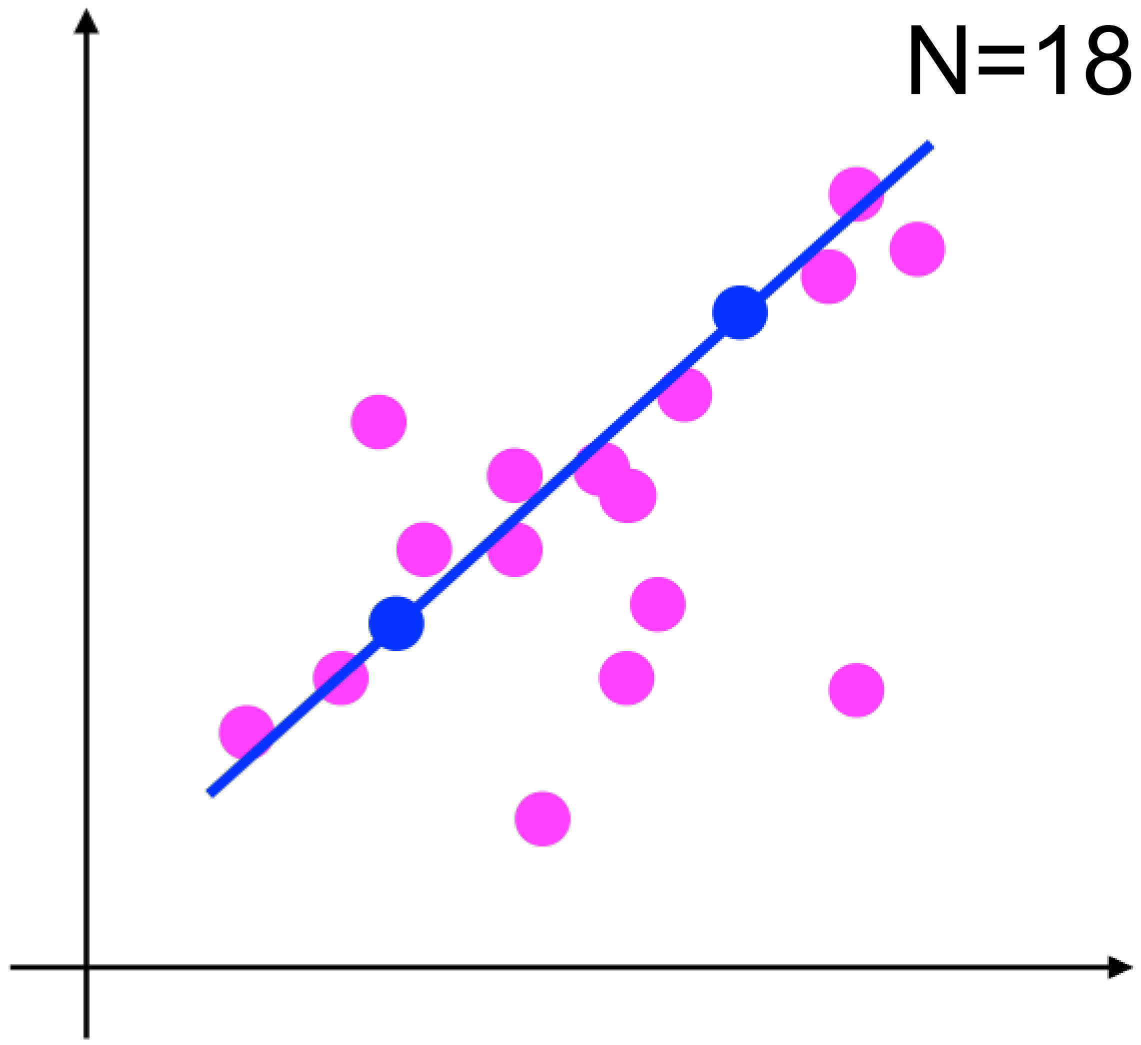
RANSAC:

1. sample 2 points
2. compute a line estimate *P’* of *P*
3. count how many points are within a **tolerance** from *P’*
4. repeat until you get a *P’* that agrees with many points

Example: Linear Regression

Fit a line through
N 2D points, possibly
corrupted with outliers.

Note: we have an algorithm
to estimate a line from n=2 points



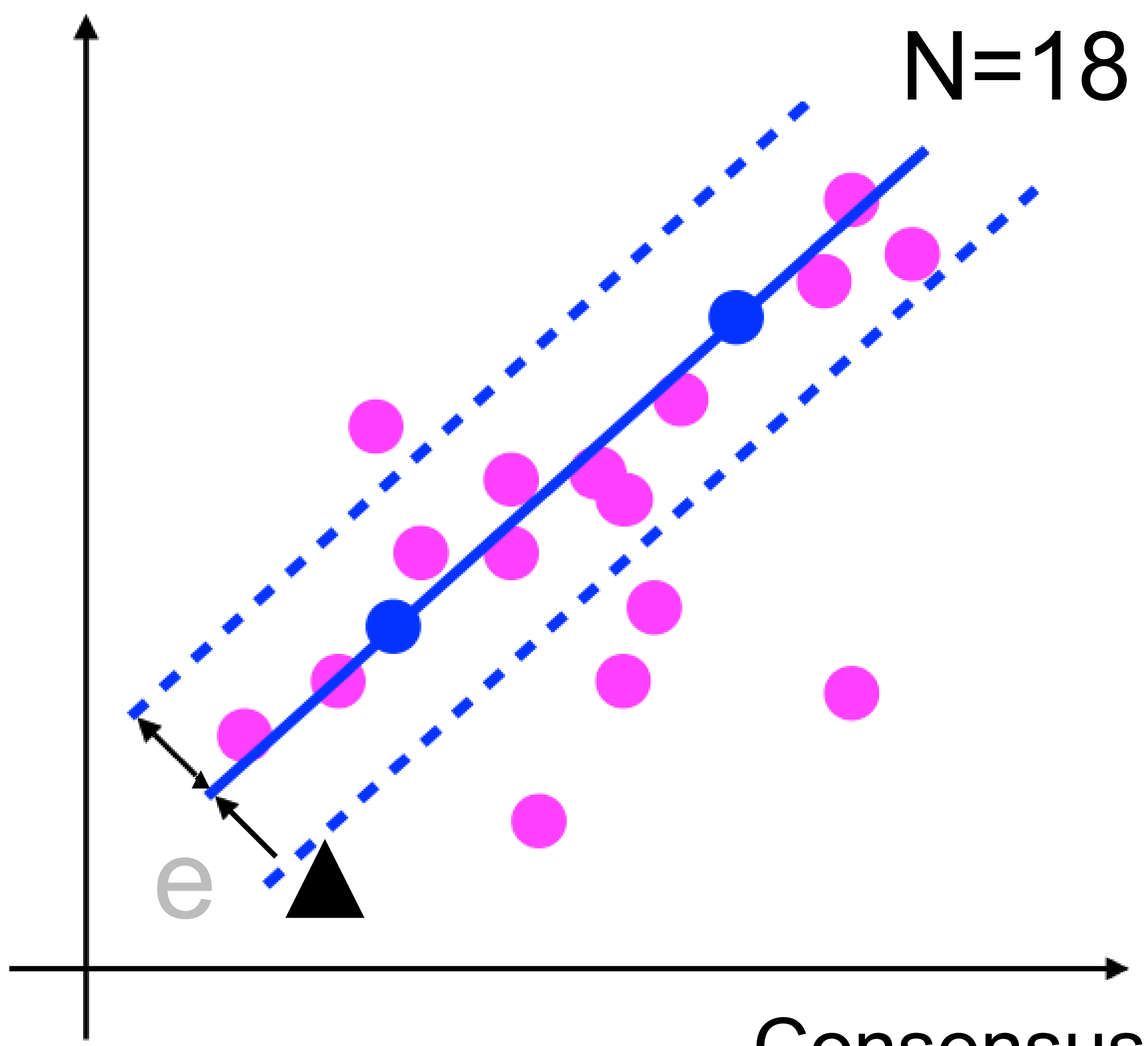
RANSAC:

1. sample 2 points
2. compute a line estimate *P'* of *P*
3. count how many points are within a **tolerance** from *P'*
4. repeat until you get a *P'* that agrees with many points

Example: Linear Regression

Fit a line through
N 2D points, possibly
corrupted with outliers.

Note: we have an algorithm
to estimate a line from n=2 points



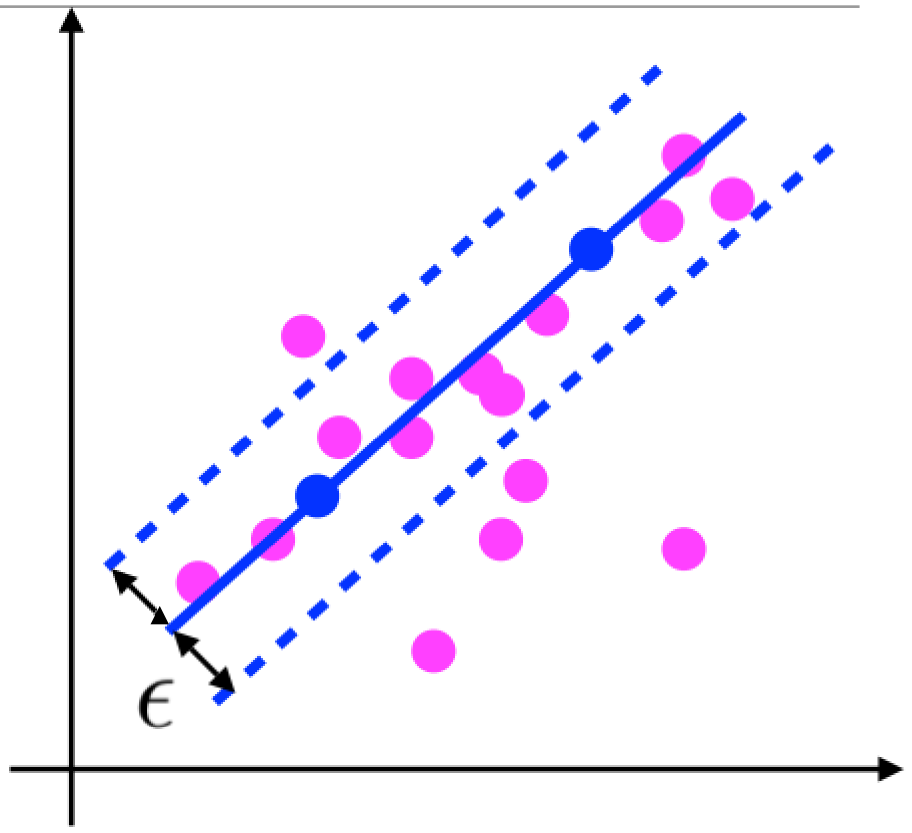
Consensus
set size = 13

RANSAC:

1. sample 2 points
2. compute a line estimate *P'* of *P*
3. count how many points are within a **tolerance** from *P'*
4. repeat until you get a *P'* that agrees with many points

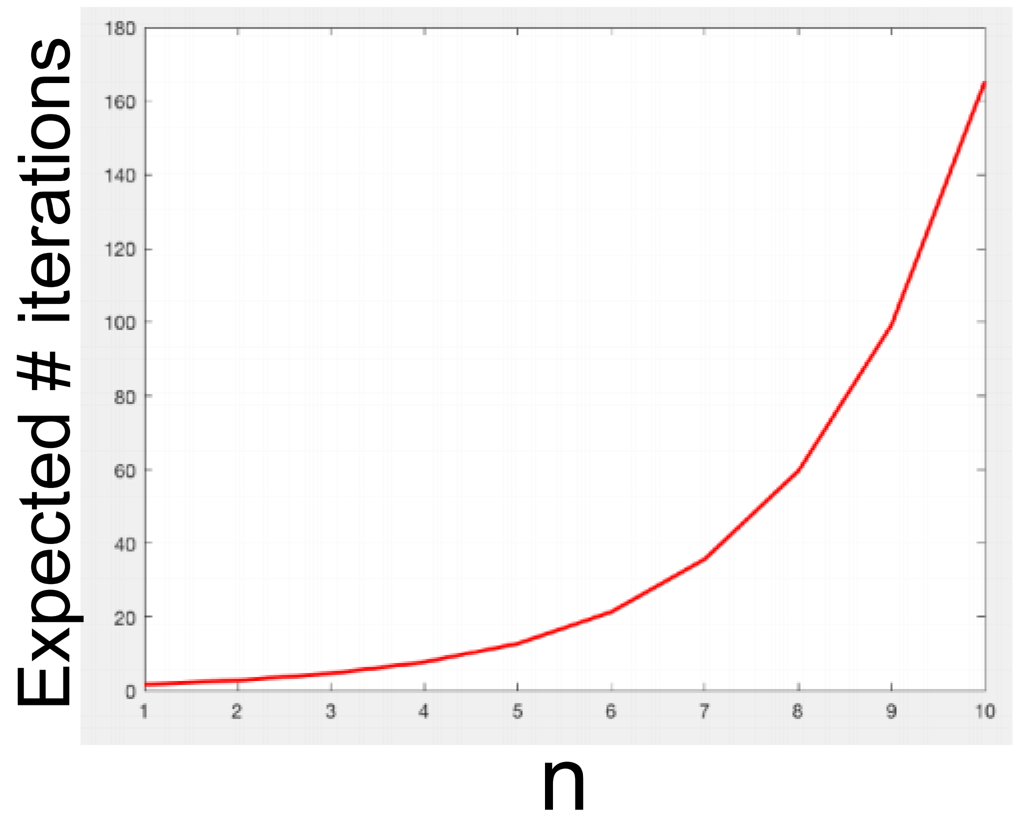
RANSAC: Parameter Tuning

1. **Error Tolerance** ϵ :
   depends on the noise

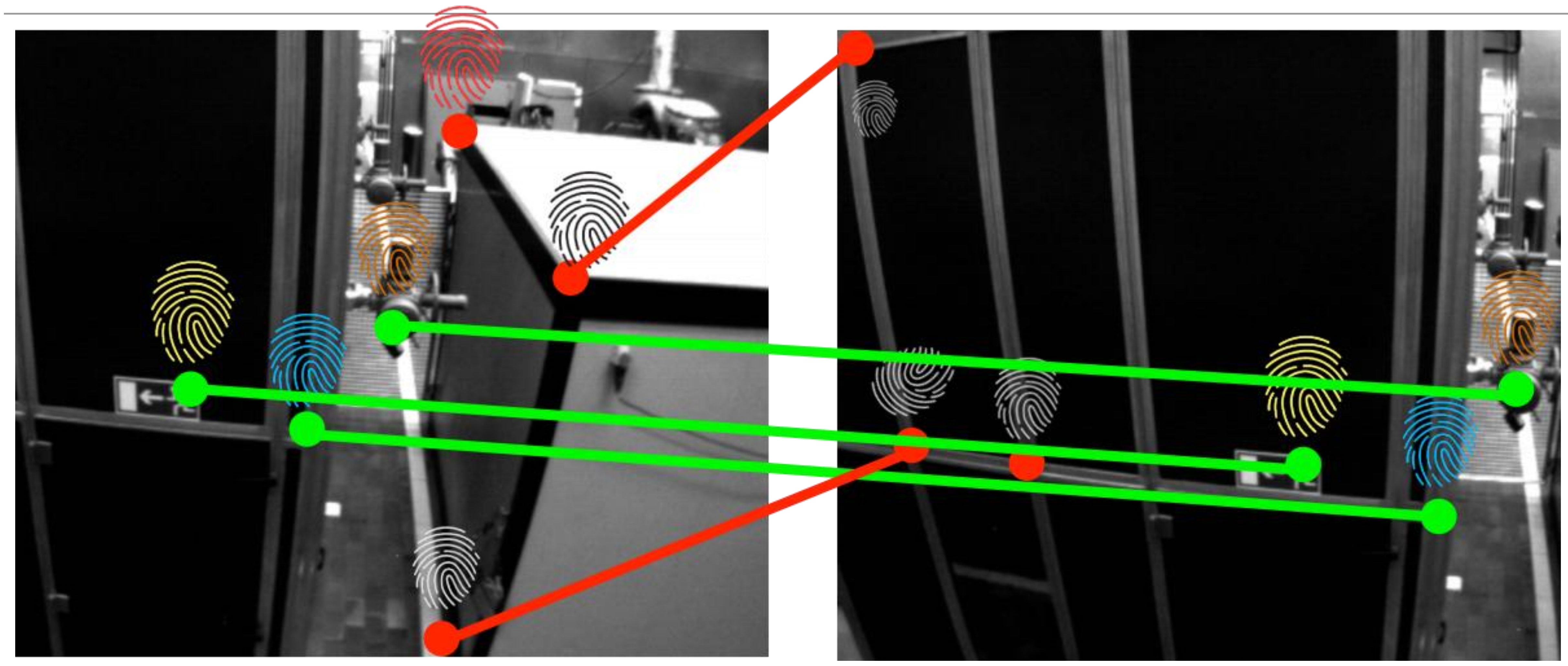


2. **Acceptable consensus set**:
   - from the paper: n+5
   - rule of thumb: >50% of points

3. **Maximum number of iterations**



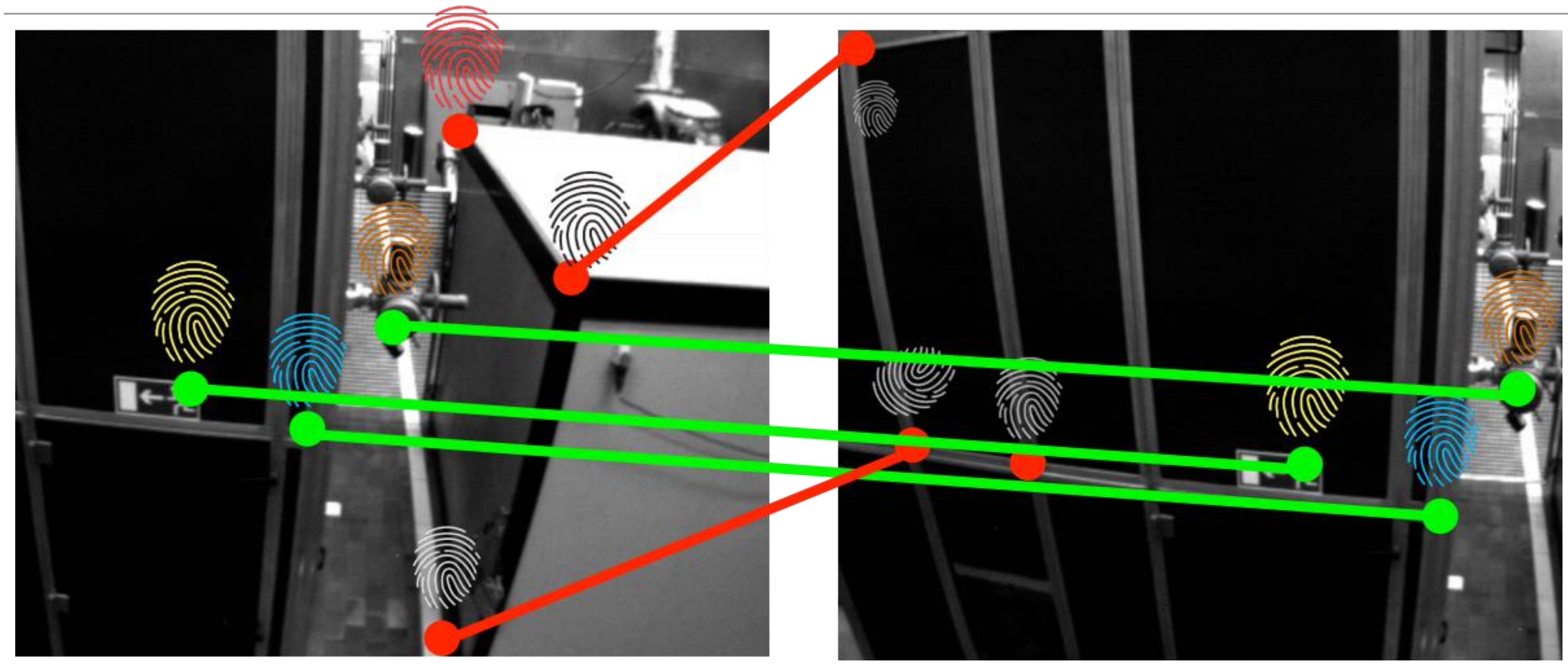
Example: RANSAC for Essential Matrix estimation



RANSAC:

1. sample *n* point correspondences
2. compute an estimate E' of the essential matrix E
3. count how many points are within a **tolerance** from *E'*
4. repeat until you get a *E'* that agrees with many points

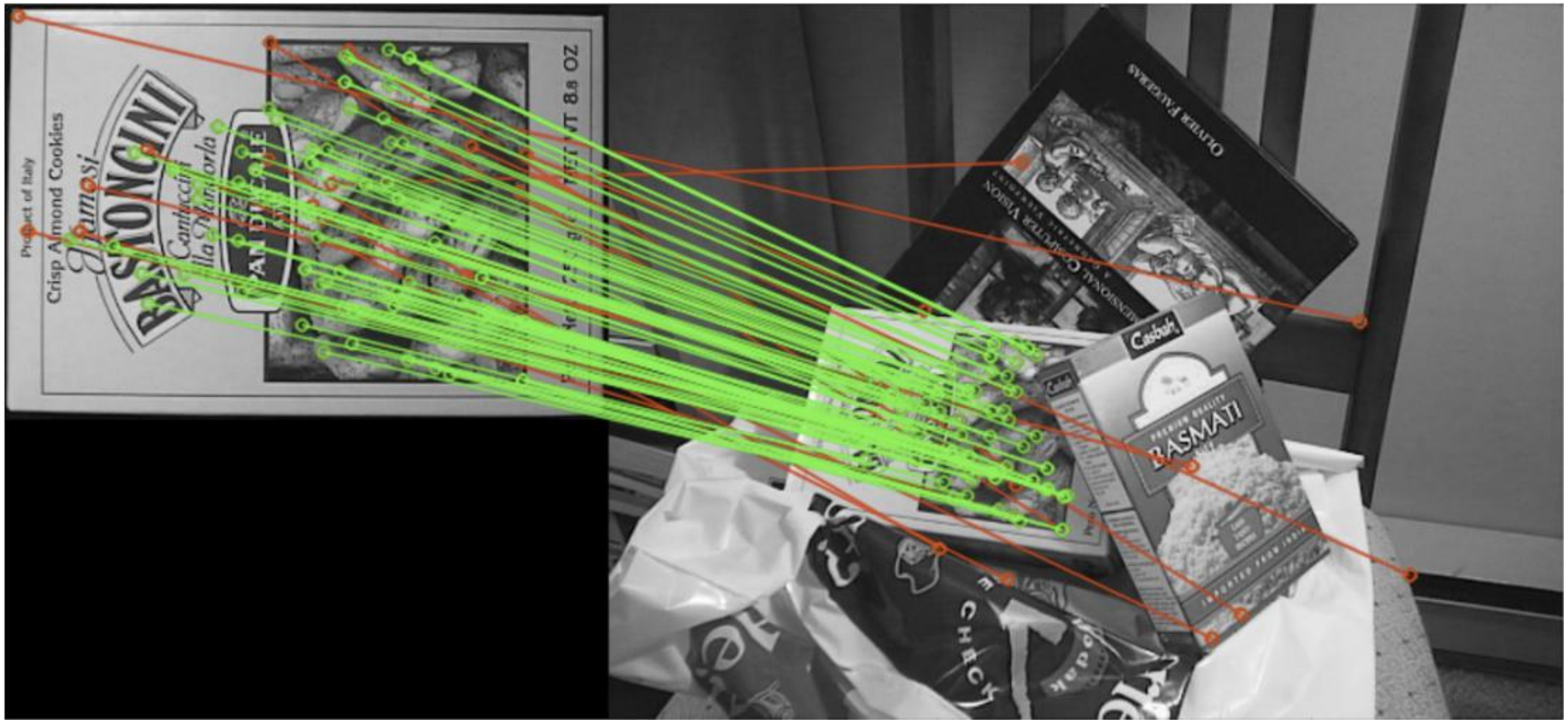
Example: RANSAC for Essential Matrix estimation



RANSAC
- essentially selects the set of inliers
- provides **geometric verification** for the correspondences

Beyond Motion Estimation

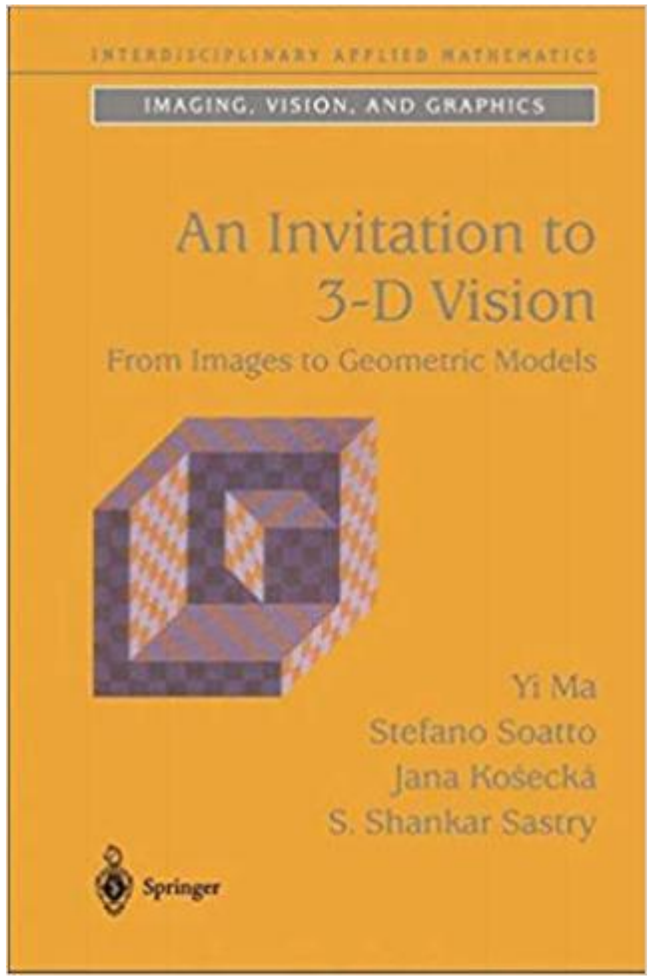
The tools we discussed (feature matching, essential matrix estimation, RANSAC) can be used also for **object detection and localization**



So far: pixel correspondences,
a.k.a., **2D-2D correspondences**

Today

- 2-view geometry
- RANSAC
- 3D-3D correspondences

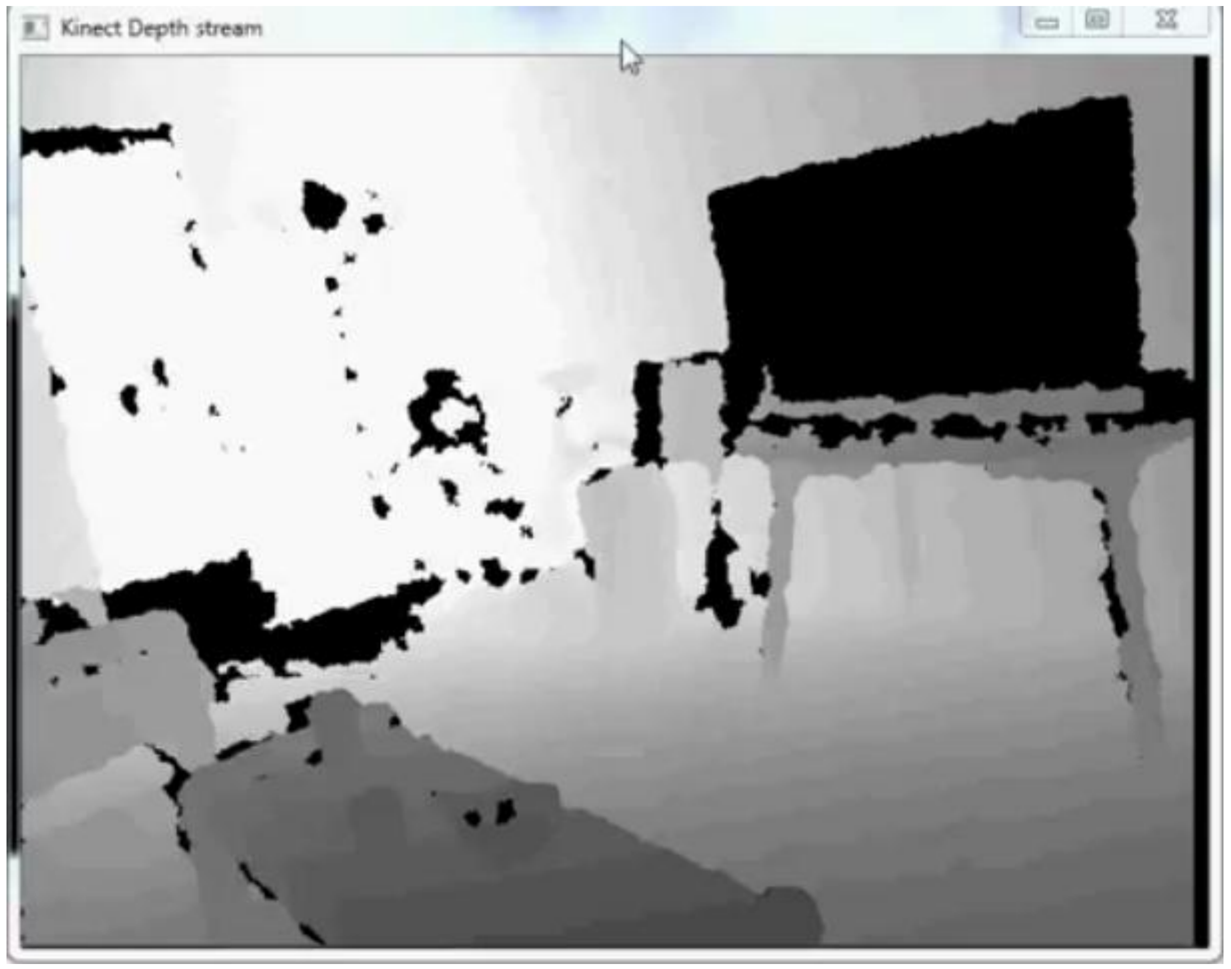

INTERDISCIPLINARY APPLIED MATHEMATICS
IMAGING, VISION, AND GRAPHICS
An Invitation to 3-D Vision
From Images to Geometric Models
Yi Ma
Stefano Soatto
Jana Košecká
S. Shankar Sastry
Springer

Chapter 5

Reconstruction from Two Calibrated Views

3D-3D Point Correspondences

Structured Light Cameras



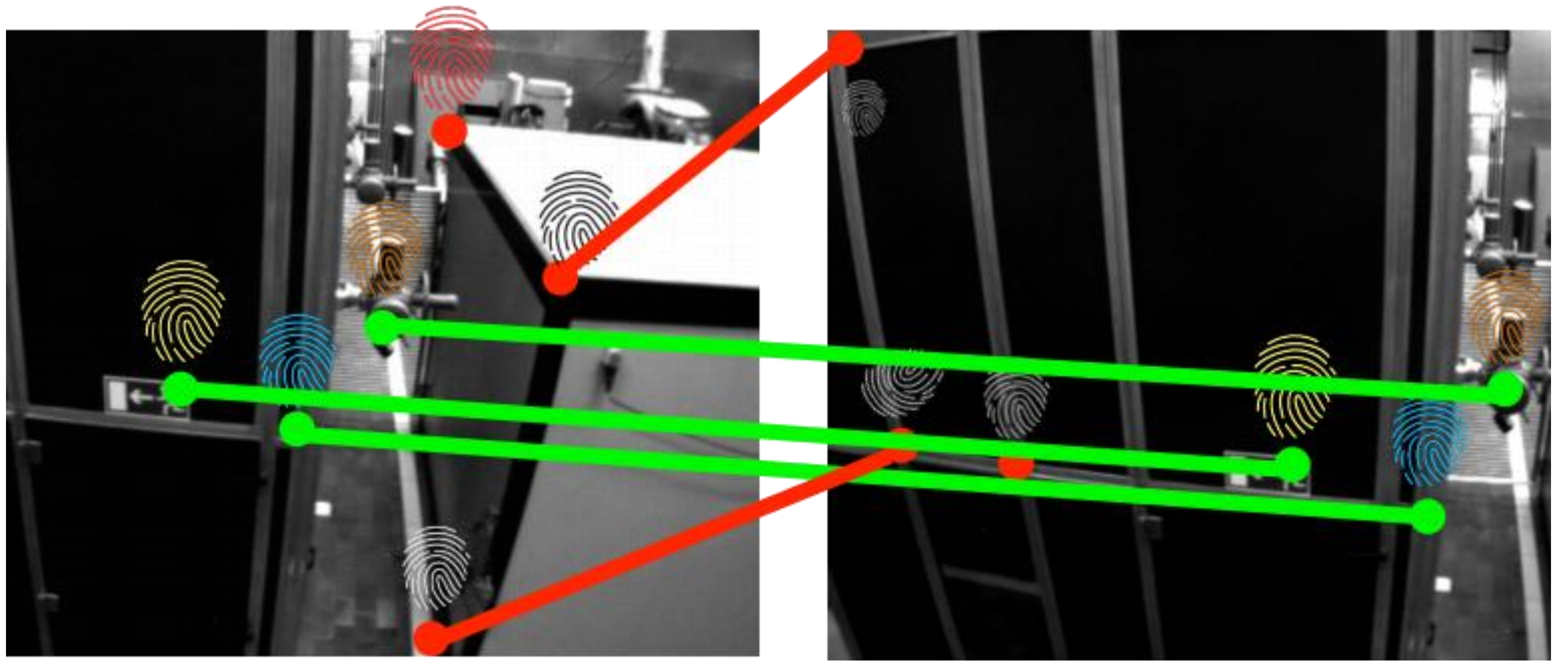
RGB-D cameras can measure depth (D) and image (RBG)

How can we use the depth information to estimate the relative pose between two RGB-D cameras observing the same scene?

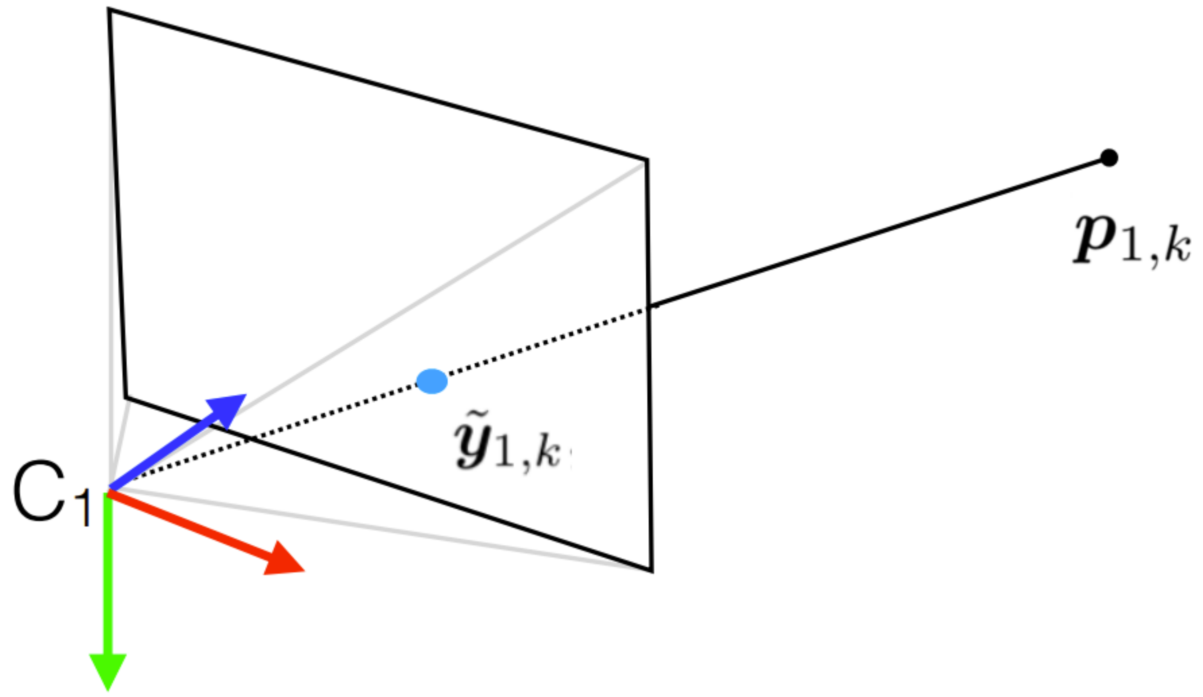
3D-3D Point Correspondences

1. We can use camera images to establish 2D-2D correspondences:

$(\tilde{\boldsymbol{y}}_{1,k}, \tilde{\boldsymbol{y}}_{2,k})$ for $k = 1, \ldots, N$

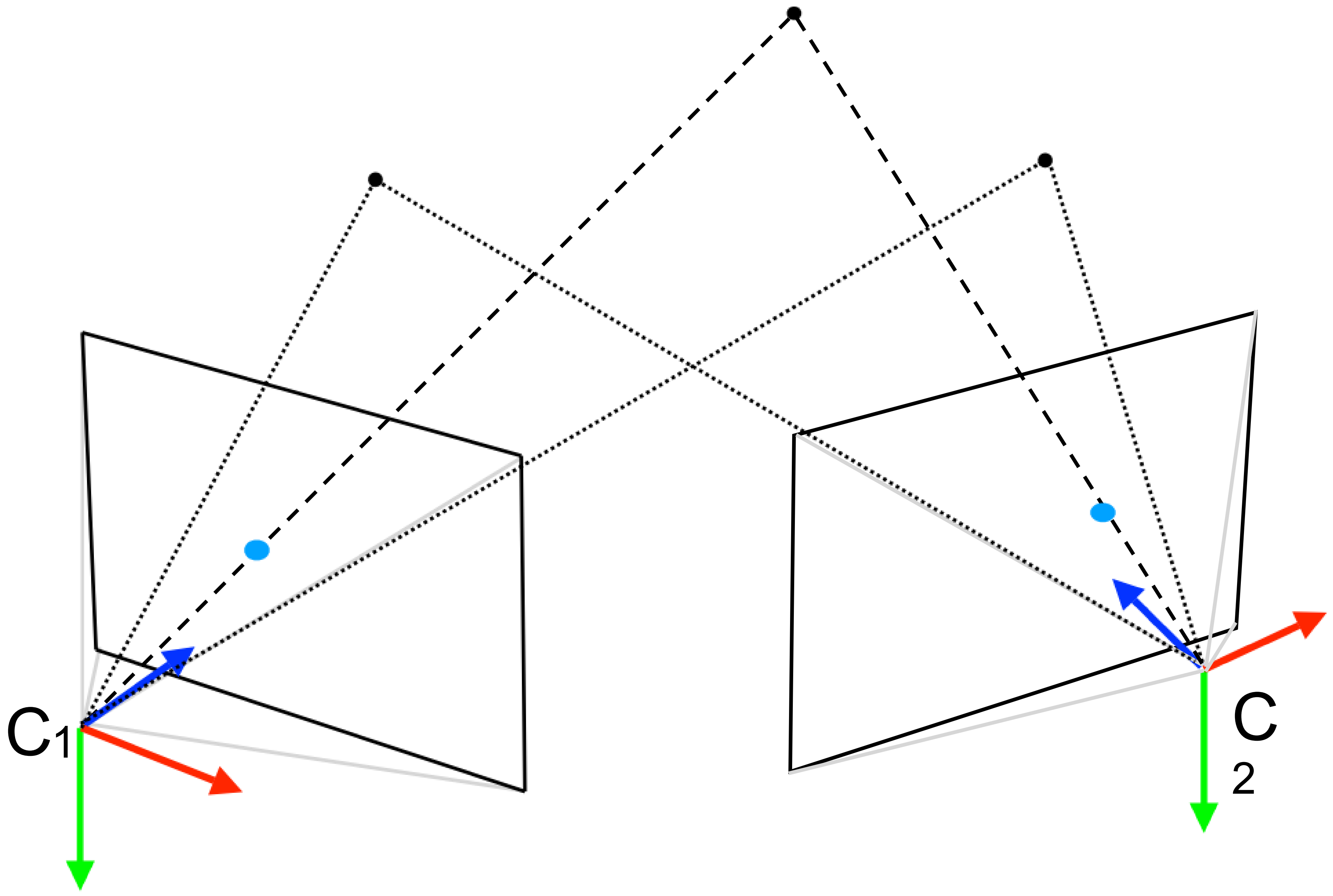


2. For each camera we can compute the set of 3D points corresponding to pixels



We obtain 3D-3D correspondences: $(\boldsymbol{p}_{1,k}, \boldsymbol{p}_{2,k}) \quad k = 1, \ldots, N$

2-view Geometry from 3D-3D Correspondences



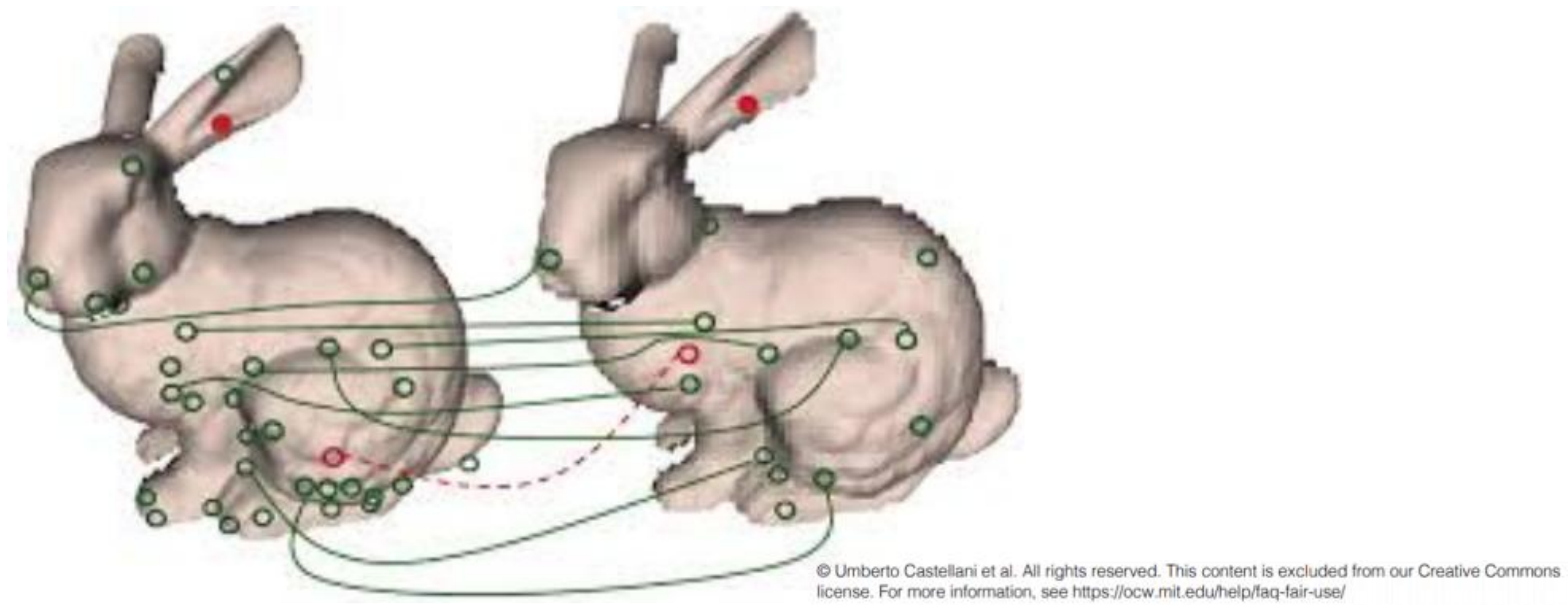
How to estimate the relative pose between the cameras from 3D-3D correspondences $(\boldsymbol{p}_{1,k}, \boldsymbol{p}_{2,k})$ **with** $k = 1, \ldots, N$ **?**

Few More Comments:

3 points are sufficient to compute the relative pose from 3D-3D correspondences

We can use the solver seen today as a 3-point minimal solver within a **RANSAC** method

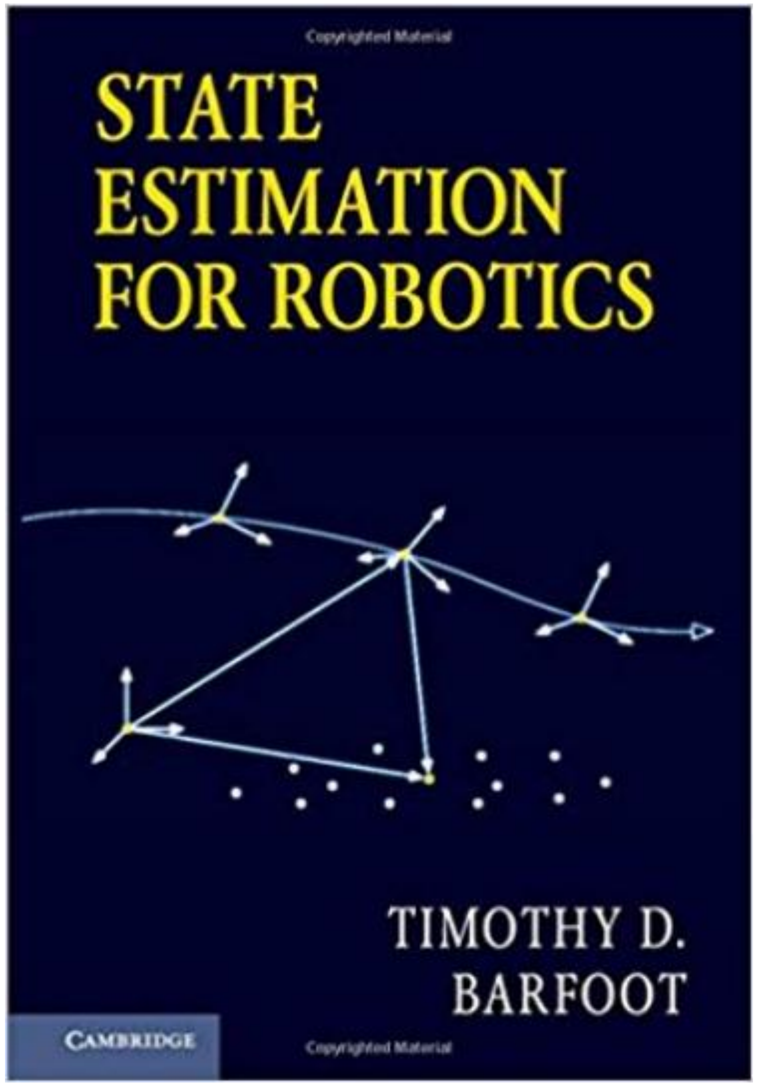
Also useful for 3D objects localization:



Other names: vector registration, point cloud alignment, ..

Today

- Optimization examples
- Estimation Basics

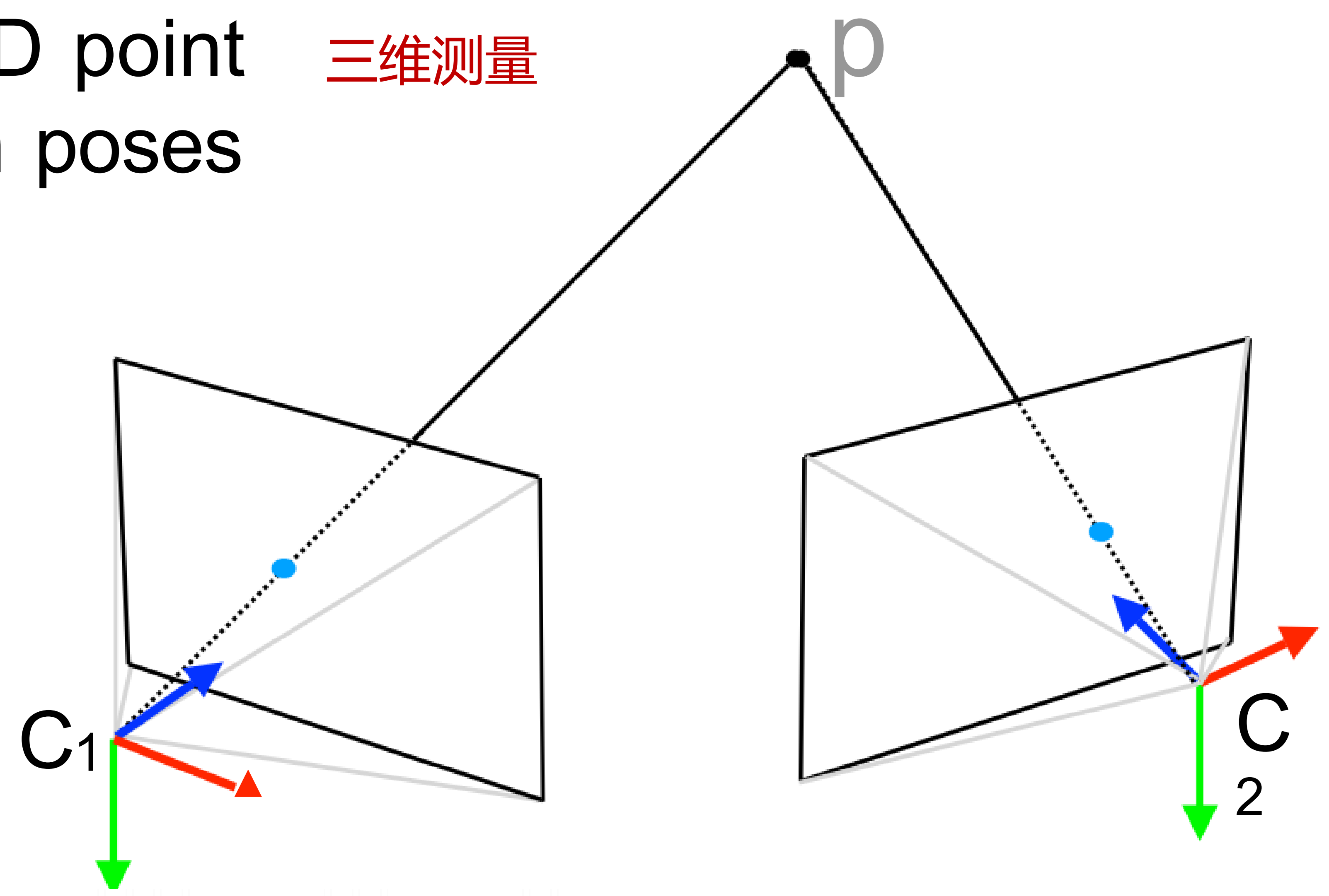


Part I: Estimation Machinery
(more than what we need)

Example **1a**: Triangulation (Structure Reconstruction)

Compute 3D point from known poses

三维测量



$$\lambda_2 \tilde{\boldsymbol{x}}_2 = \underbrace{\boldsymbol{K}_2 [\boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}_2} \; \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}_2}]}_{\boldsymbol{\Pi}_1} \tilde{\boldsymbol{p}}^{\mathrm{w}}$$

$$\lambda_2 \tilde{\boldsymbol{x}}_2 = \underbrace{\boldsymbol{K}_2 [\boldsymbol{R}_{\mathrm{w}}^{\mathrm{c}_2} \; \boldsymbol{t}_{\mathrm{w}}^{\mathrm{c}_2}]}_{\boldsymbol{\Pi}_2} \tilde{\boldsymbol{p}}^{\mathrm{w}}$$

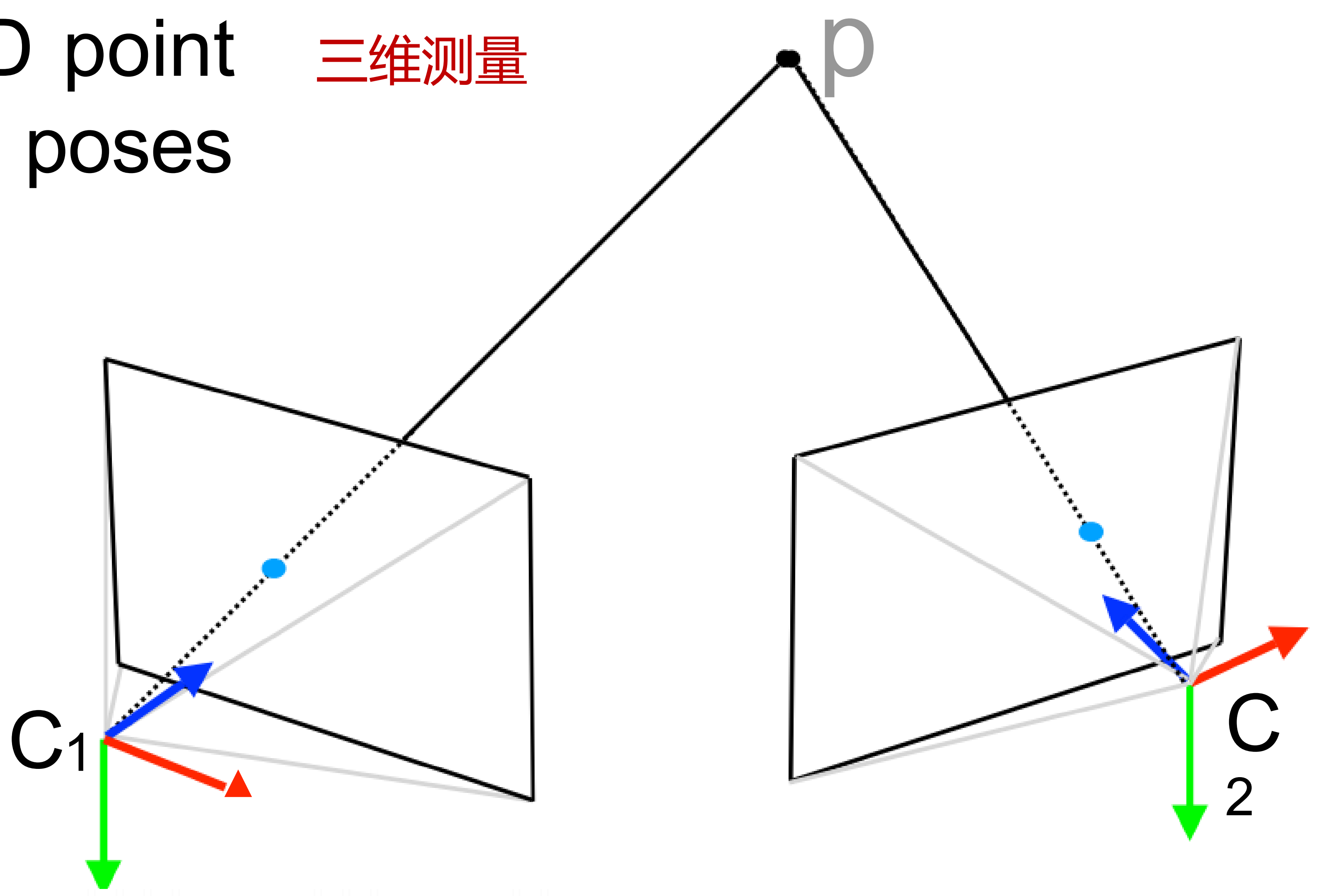
线性优化求解

Linear triangulation: $\min_{\|\tilde{\boldsymbol{p}}^{\mathrm{w}}\|=1} \|\boldsymbol{A}\tilde{\boldsymbol{p}}^{\mathrm{w}}\|^2$

Example **1b**: Triangulation (Structure Reconstruction)

Compute 3D point from known poses

三维测量



$$\lambda_2 \tilde{\boldsymbol{x}}_2 = \underbrace{\boldsymbol{K}_2 [\boldsymbol{R}^{c_2}_{w} \ \boldsymbol{t}^{c_2}_{w}]}_{\boldsymbol{\Pi}_1} \tilde{\boldsymbol{p}}^{w}$$

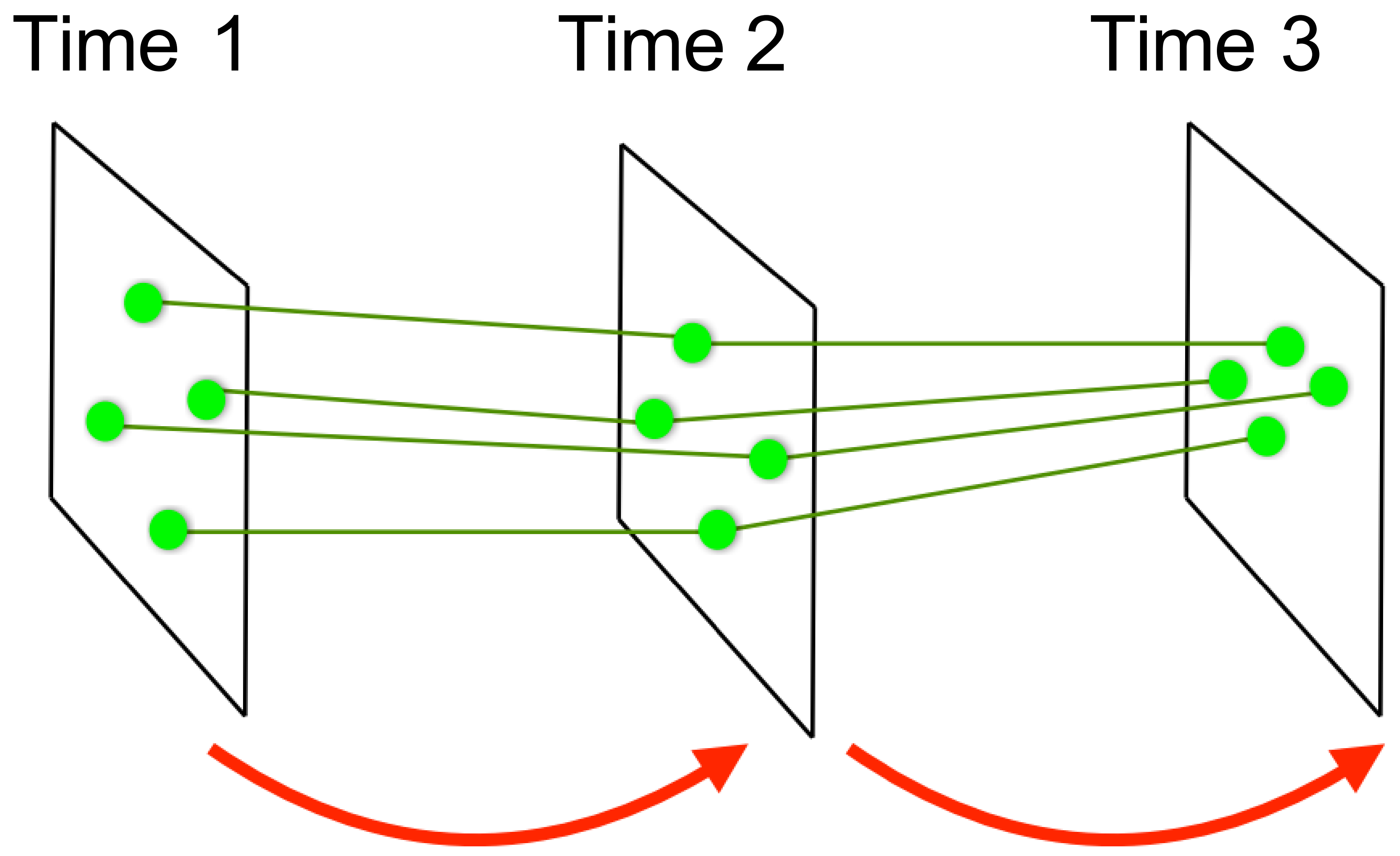
$$\lambda_2 \tilde{\boldsymbol{x}}_2 = \underbrace{\boldsymbol{K}_2 [\boldsymbol{R}^{c_2}_{w} \ \boldsymbol{t}^{c_2}_{w}]}_{\boldsymbol{\Pi}_2} \tilde{\boldsymbol{p}}^{w}$$

非线性优化求解

$$\min_{\boldsymbol{p}^{w}} \|\boldsymbol{x}_1 - \pi(\boldsymbol{R}^{w}_{c_1}, \boldsymbol{t}^{w}_{c_1}, \boldsymbol{p}^{w})\|^2 + \|\boldsymbol{x}_2 - \pi(\boldsymbol{R}^{w}_{c_2}, \boldsymbol{t}^{w}_{c_2}, \boldsymbol{p}^{w})\|^2$$

Example **2a**: Motion Estimation

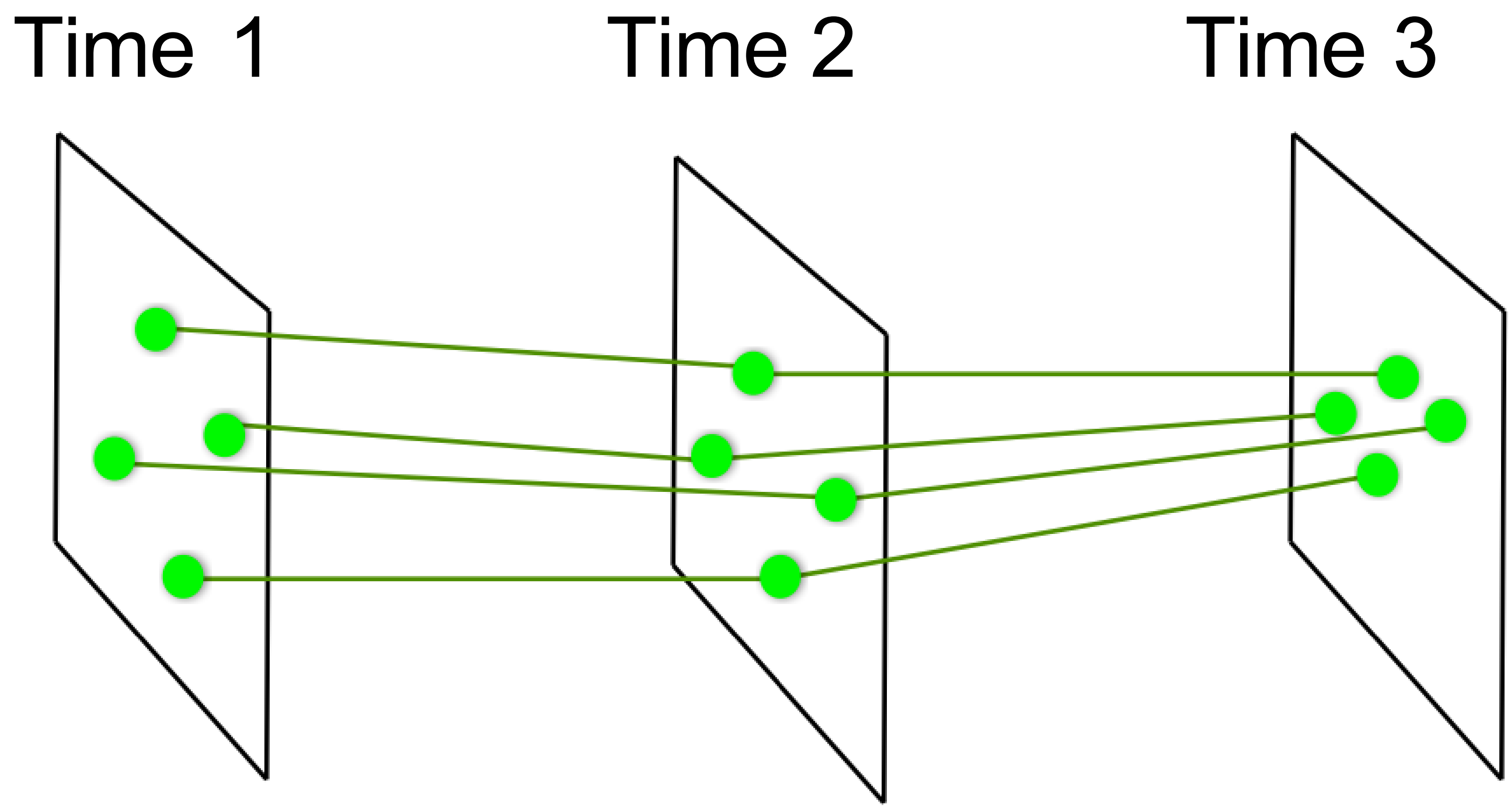
（1）已知2D图像到2D图像之间的关系，求解图像帧间位姿



$$\boldsymbol{E}_{12} = \underset{\boldsymbol{E}_{12} \in \mathcal{S}_E}{\arg\min} \sum_{k=1}^{N} |\tilde{\boldsymbol{y}}_{k,2}^{\mathsf{T}} \boldsymbol{E}_{12} \tilde{\boldsymbol{y}}_{k,1}|^2 \qquad \boldsymbol{E}_{23} = \underset{\boldsymbol{E}_{23} \in \mathcal{S}_E}{\arg\min} \sum_{k=1}^{N} |\tilde{\boldsymbol{y}}_{k,3}^{\mathsf{T}} \boldsymbol{E}_{23} \tilde{\boldsymbol{y}}_{k,2}|^2$$

Example **2b**: Motion Estimation

（2）已知2D图像&对应3D到2D图像之间的关系，求解图像帧间位姿



已知：n个点在世界坐标系下的坐标P_1、P_2、……、P_i、……、P_n，

对应像素坐标系下的坐标p_1、p_2、……、p_i、……、p_n，

相机的内参矩阵K。

求解：**相机坐标系$(O_cX_cY_cZ_c)$相对于世界坐标系$(O_wX_wY_wZ_w)$的位姿**，公式(1)中的$[R \quad t]$。

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = [R \quad t] \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

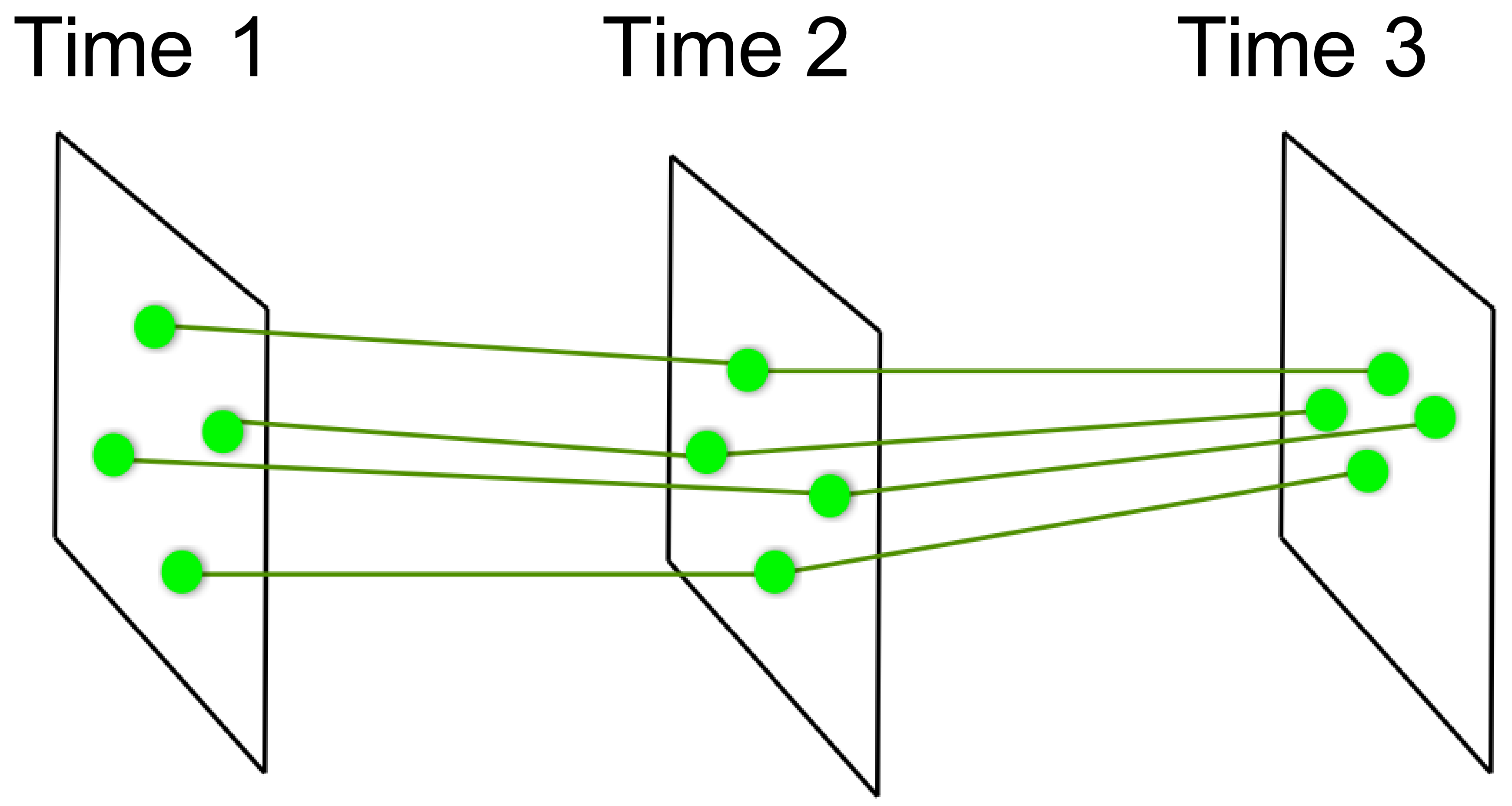
详解PnP - 知乎

PnP算法详解（超详细公式推导）-CSDN博客

一文详解PnP算法原理 - 知乎

Example 2c: Motion Estimation

(3）已知3D到2D图像之间的关系，求解图像帧间位姿

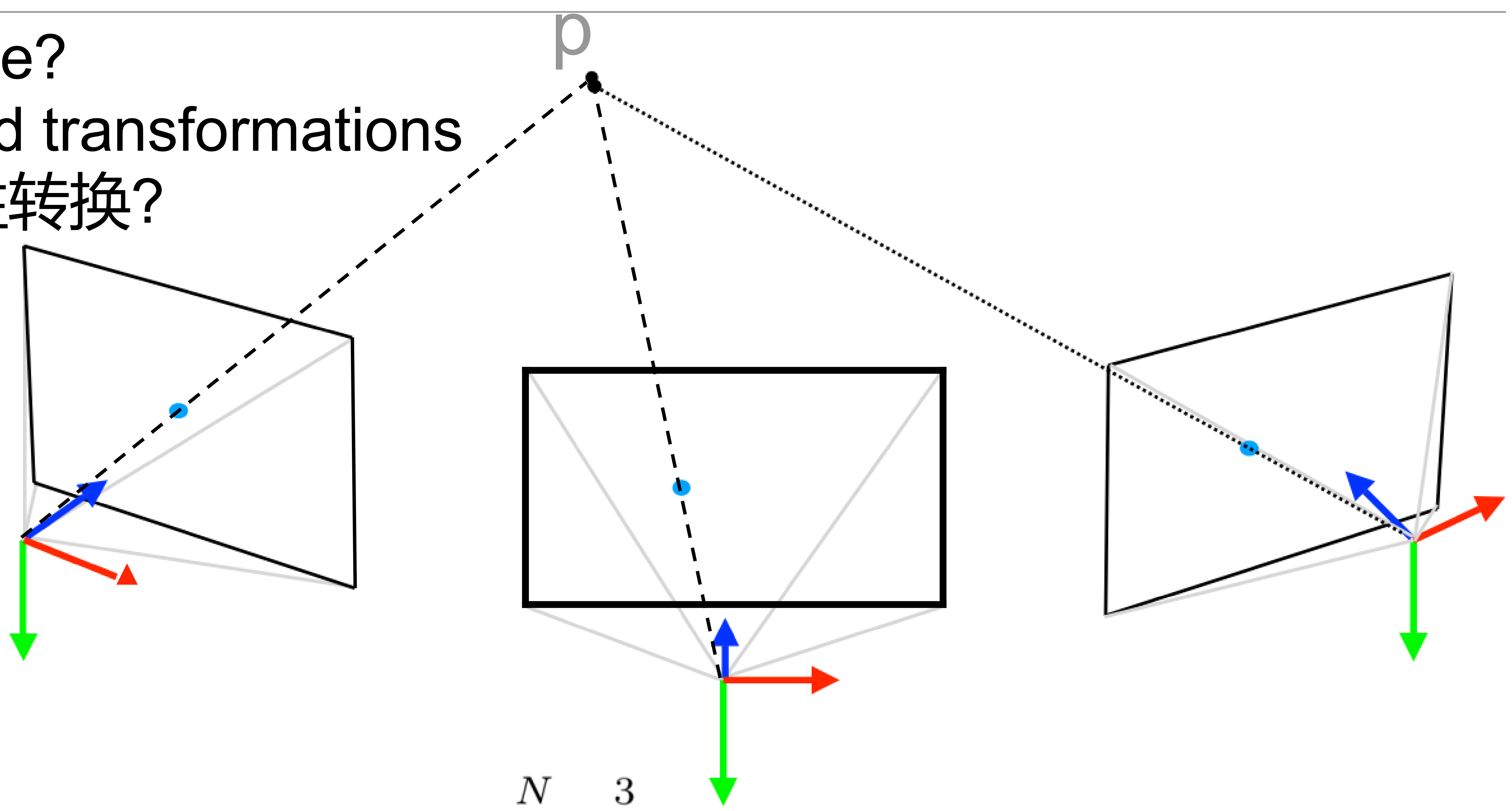


$$\min_{\substack{(\boldsymbol{R}^{\mathrm{w}}_{c_i},\boldsymbol{t}^{\mathrm{w}}_{c_i}),i=1,2,3\\ \boldsymbol{p}^{\mathrm{w}}_k,k=1,\ldots,N}} \sum_{k=1}^{N}\sum_{i=1}^{3} \|\boldsymbol{x}_{k,i} - \pi(\boldsymbol{R}^{\mathrm{w}}_{c_i}, \boldsymbol{t}^{\mathrm{w}}_{c_i}, \boldsymbol{p}^{\mathrm{w}}_k)\|^2$$

Generalizes to K cameras: **Bundle adjustment**

Example **2c**: Motion **and Structure** Estimation

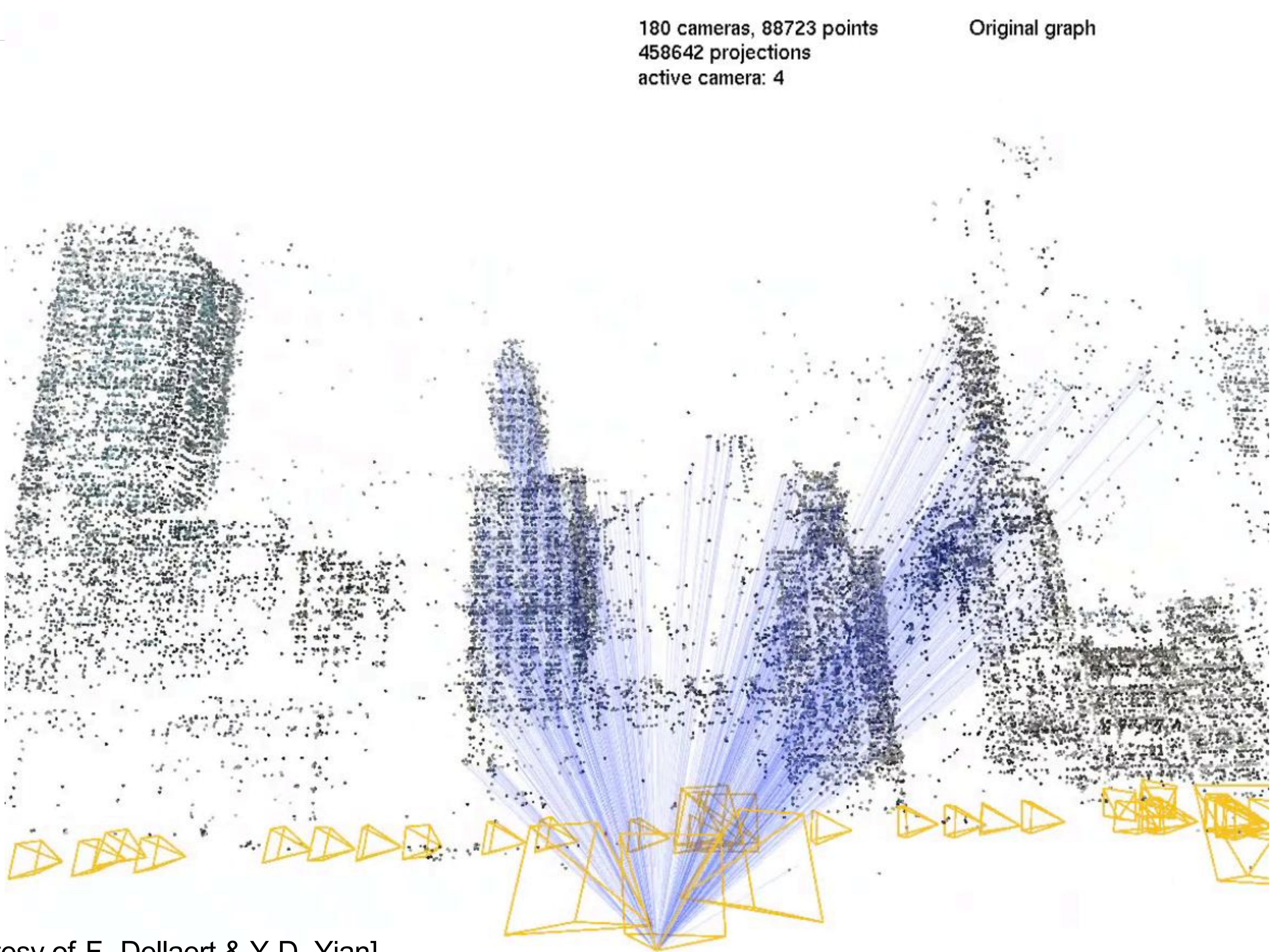
Scale?
Rigid transformations
刚性转换?



$$\min_{\substack{(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_i},\boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_i}),i=1,2,3\\ \boldsymbol{p}^{\mathrm{w}}_k,k=1,\ldots,N}} \sum_{k=1}^{N}\sum_{i=1}^{3} \|\boldsymbol{x}_{k,i} - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_i},\boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_i},\boldsymbol{p}^{\mathrm{w}}_k)\|^2$$

Generalizes to K cameras: **Bundle adjustment**

Structure from Motion



[courtesy of F. Dellaert & Y-D. Yian]

Optimization

Linear triangulation:

$$\min_{\|\tilde{\boldsymbol{p}}^{\mathrm{w}}\|=1} \|\boldsymbol{A}\tilde{\boldsymbol{p}}^{\mathrm{w}}\|^2$$

Nonlinear triangulation:

$$\min_{\boldsymbol{p}^{\mathrm{w}}} \|\boldsymbol{x}_1 - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_1}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_1}, \boldsymbol{p}^{\mathrm{w}})\|^2 + \\ + \|\boldsymbol{x}_2 - \pi(\boldsymbol{R}^{\mathrm{w}}_{\mathrm{c}_2}, \boldsymbol{t}^{\mathrm{w}}_{\mathrm{c}_2}, \boldsymbol{p}^{\mathrm{w}})\|^2$$

End