CSE 152: Computer Vision

Hao Su

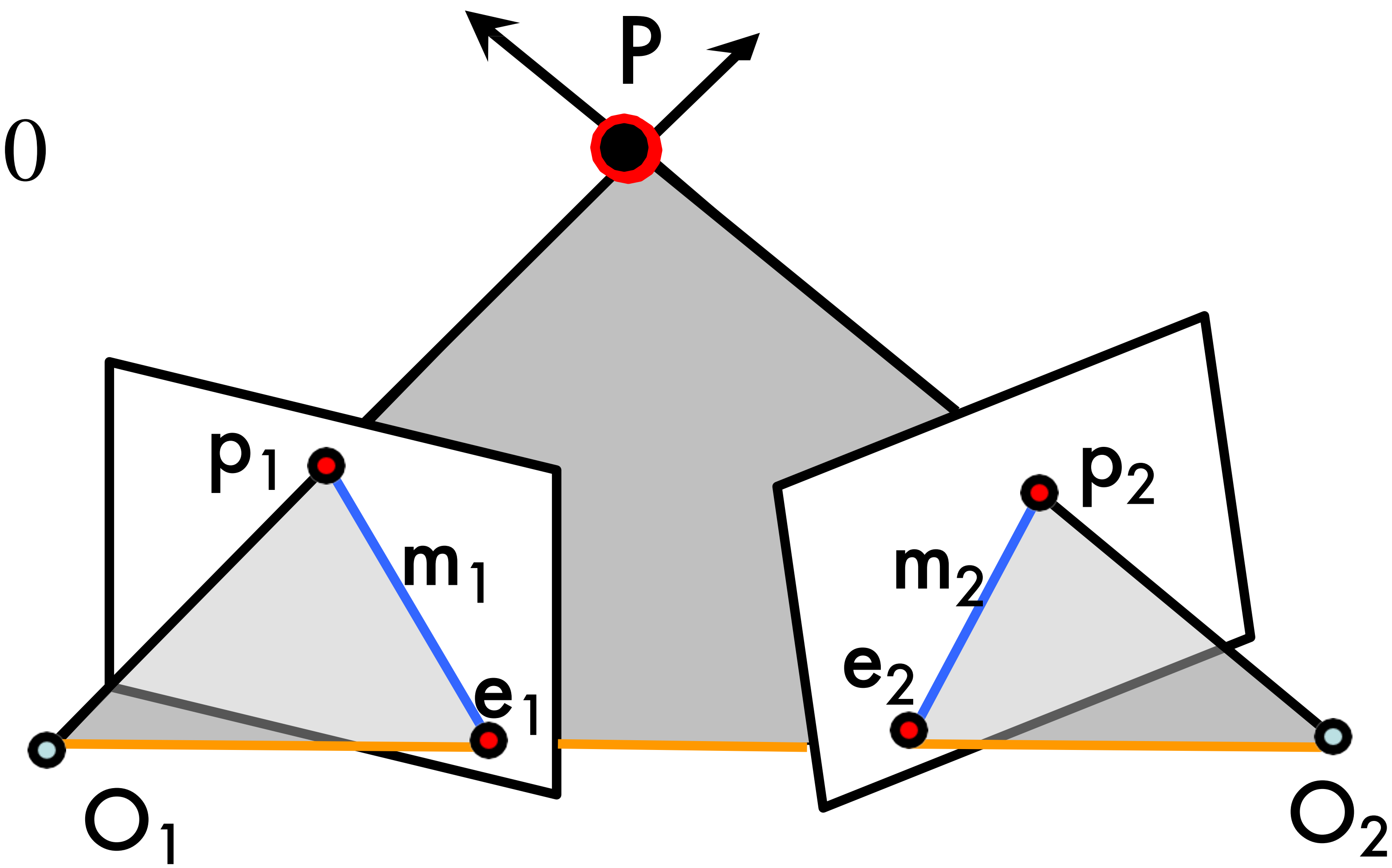
Lecture 16: Stereo Reconstruction

Agenda

- **Estimating F from Correspondences**
- RANSAC for F Estimation
- Multi-View 3D Reconstruction

Epipolar Constraint

$$p_1^T \cdot F p_2 = 0$$



- w_1 = F p_2 defines an equation $w_1^T p_1 = 0$, the epipolar line m_1 of p_2
- $w_2 = F^T p_1$ defines an equation $w_2^T p_2 = 0$, the epipolar line m_2 of p_1
- F is singular (rank two)
- $F e_2 = 0$ and $F^T e_1 = 0$

Estimating F

Suppose we have a pair of corresponding points:

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad p' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

[Eq. 13] $$\mathrm{p}^{\mathrm{T}}\,\mathrm{F}\,\mathrm{p}' = 0 \quad \Rightarrow$$

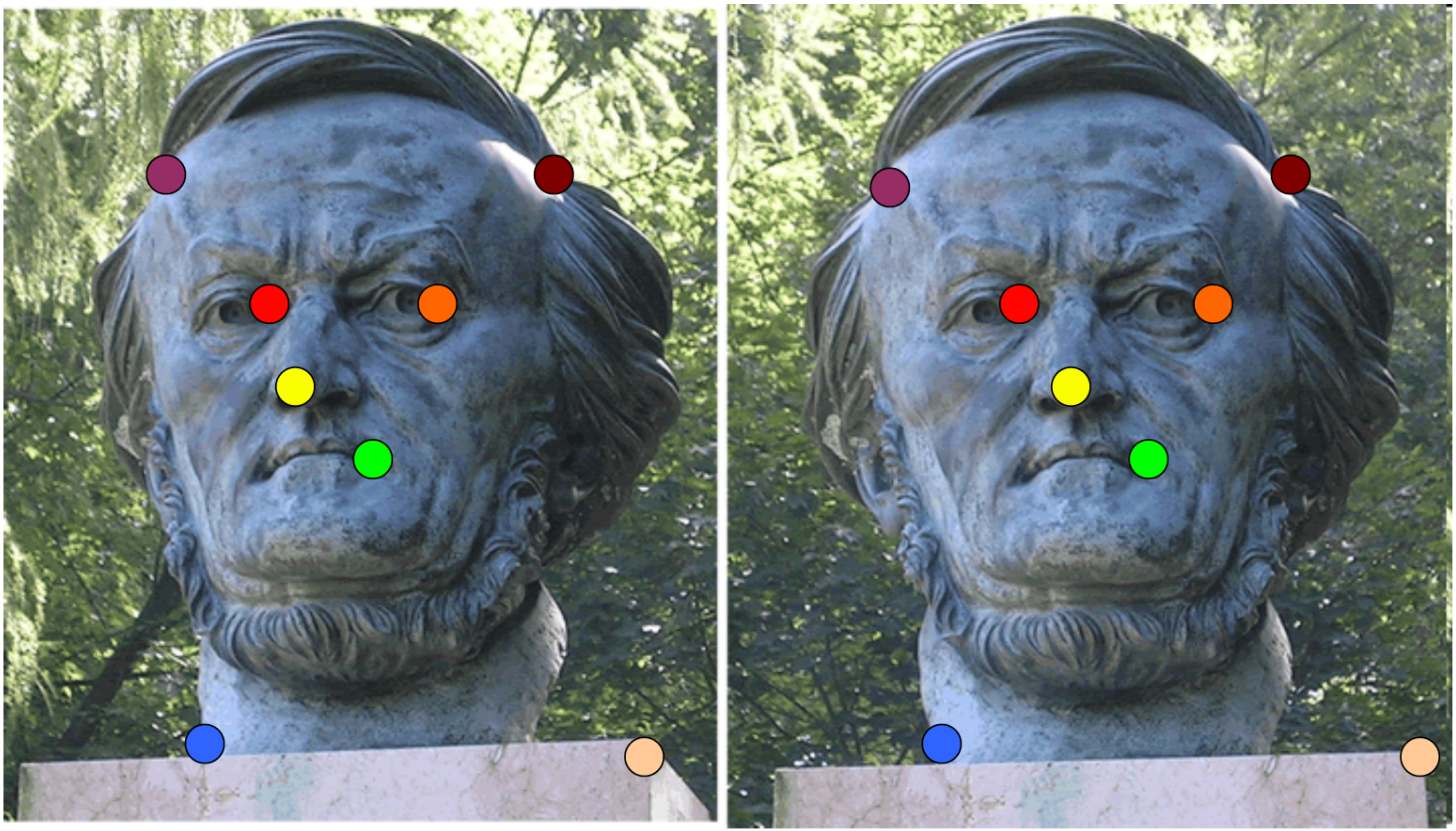
$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

$$\Rightarrow \quad (uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

[Eq. 14]

Let's take 8 corresponding points

Estimating F



Estimating F

$$\left(u_i u'_i, u_i v'_i, u_i, v_i u'_i, v_i v'_i, v_i, u'_i, v_i', 1\right)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0 \qquad \text{[Eq. 14]}$$

Estimating F

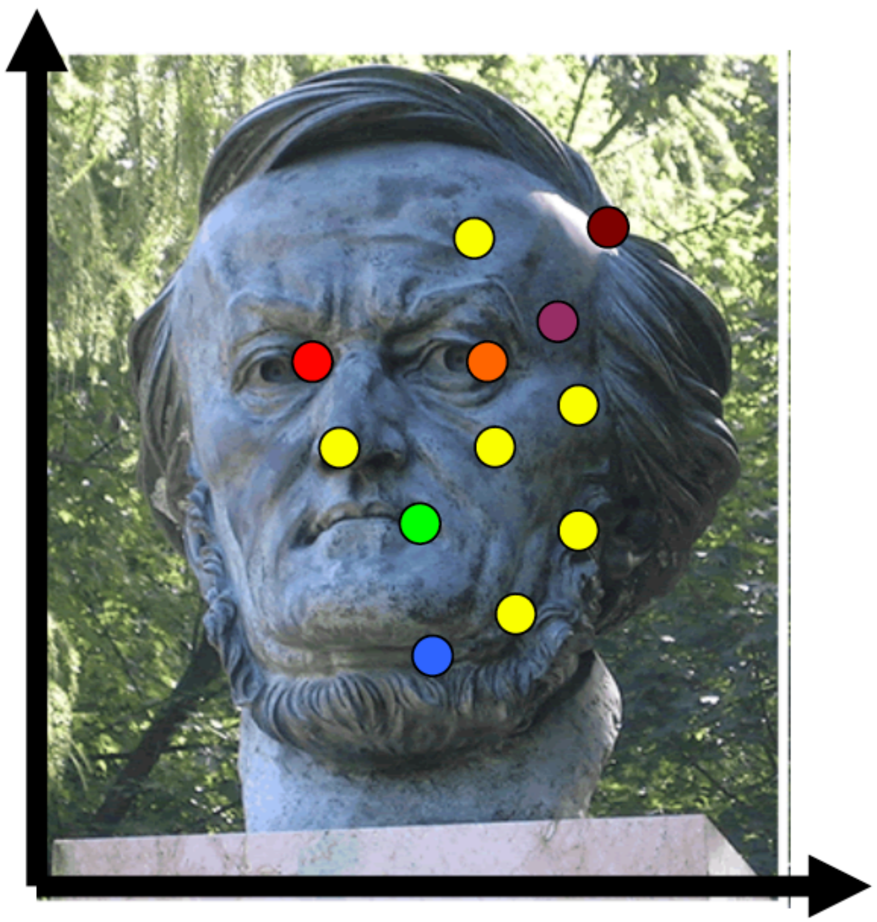
$$\mathbf{W}\begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 & 1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 & 1 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 & 1 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 & 1 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 & 1 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 & 1 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 & 1 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 & 1 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0 \quad \mathbf{f}$$

[Eqs. 15]

- Homogeneous system $\mathbf{W}\,\mathbf{f} = 0$
- Rank 8 ⟶ A non-zero solution exists (up to scale)
- If N>8 ⟶ Lsq. solution by SVD! ⟶ $\hat{\mathrm{F}}$

  $\|\mathbf{f}\| = 1$

Basic Flow of the 8-Point Algorithm



$$Wf = 0, \ \|f\| = 1$$

↓ Least-square

$$\underset{f}{\textbf{minimize}} \quad \|Wf\|^2$$
$$s.t. \quad \|f\| = 1$$

↓

$$\underset{f}{\textbf{minimize}} \quad f^T W^T W f$$
$$s.t. \quad f^T f = 1$$

↓

Do you remember how to solve the problem?
Hint: Check your HW1 (by the SVD of W)

$\hat{F}$ satisfies: $p^T \hat{F} p' = 0$

and estimated $\hat{F}$ may have full rank ($\det(\hat{F}) \neq 0$)

But remember: fundamental matrix is Rank2

$\hat{F}$ satisfies: $p^T \hat{F} p' = 0$

and estimated $\hat{F}$ may have full rank ($\det(\hat{F}) \neq 0$)

But remember: fundamental matrix is Rank2

Find F that minimizes $\|F - \hat{F}\|$ Frobenius norm (*)

Subject to rank(F)=2

SVD (again!) can be used to solve this problem

(*) Sq. root of the sum of squares of all entries

Find F that minimizes $\left\| \mathrm{F} - \hat{\mathrm{F}} \right\|$ Frobenius norm (*)

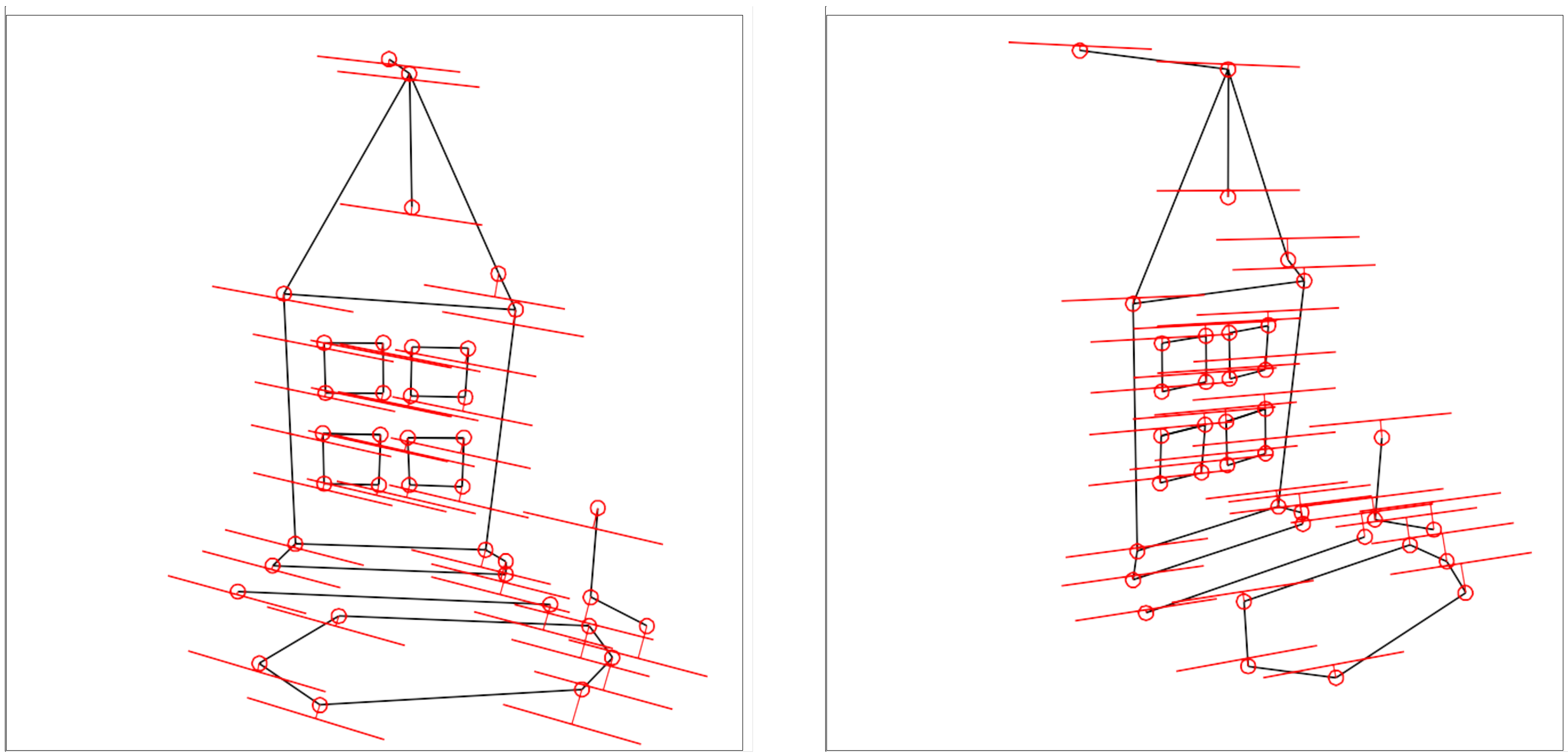
Subject to det(F)=0

$$F = U \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

Where:

$$U \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix} V^T = SVD(\hat{F})$$

[HZ] pag 281, chapter 11, "Computation of F"



Mean errors:

10.0pixel

9.1pixel

Agenda

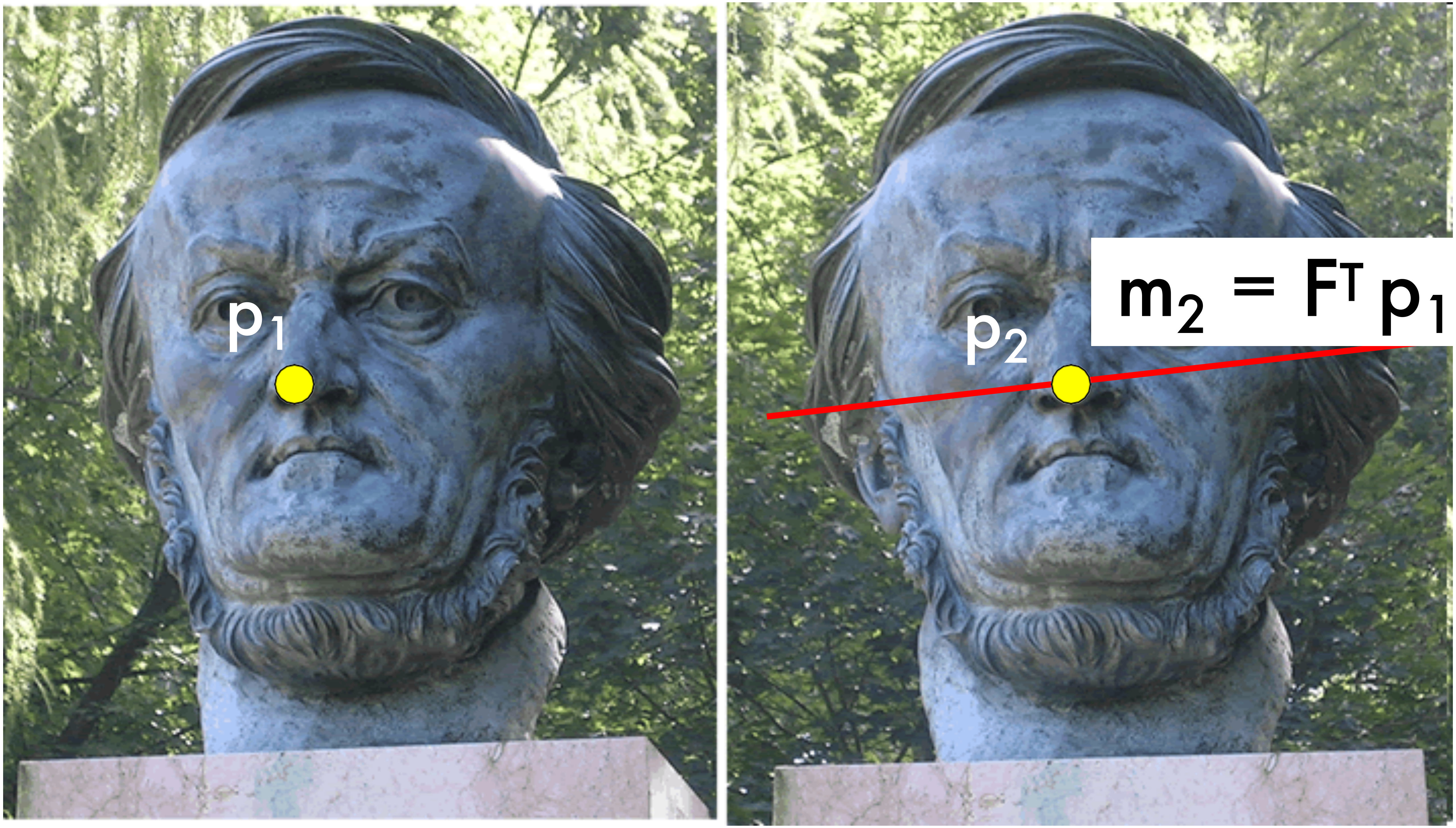
- Estimating F from Correspondences
- **RANSAC for F Estimation**
- Multi-View 3D Reconstruction

Possible Ways to Obtain Correspondences

- Human annotation
  - expensive
  - not real-time

- Automatic correspondence computation

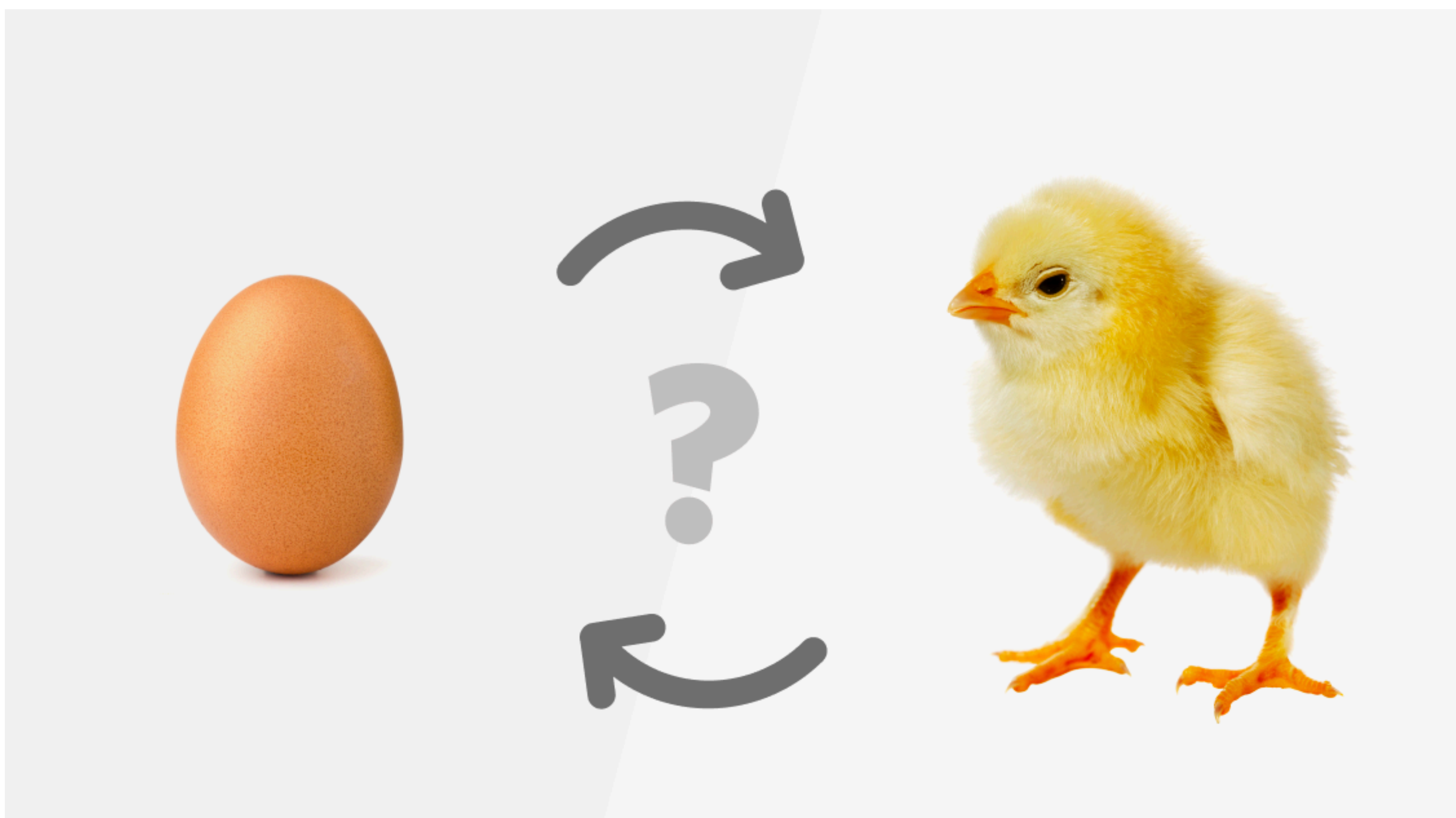
When F is known:



- Suppose F is known
- No additional information about the scene and camera is given
- Given a point on left image, we can compute the corresponding epipolar line in the second image

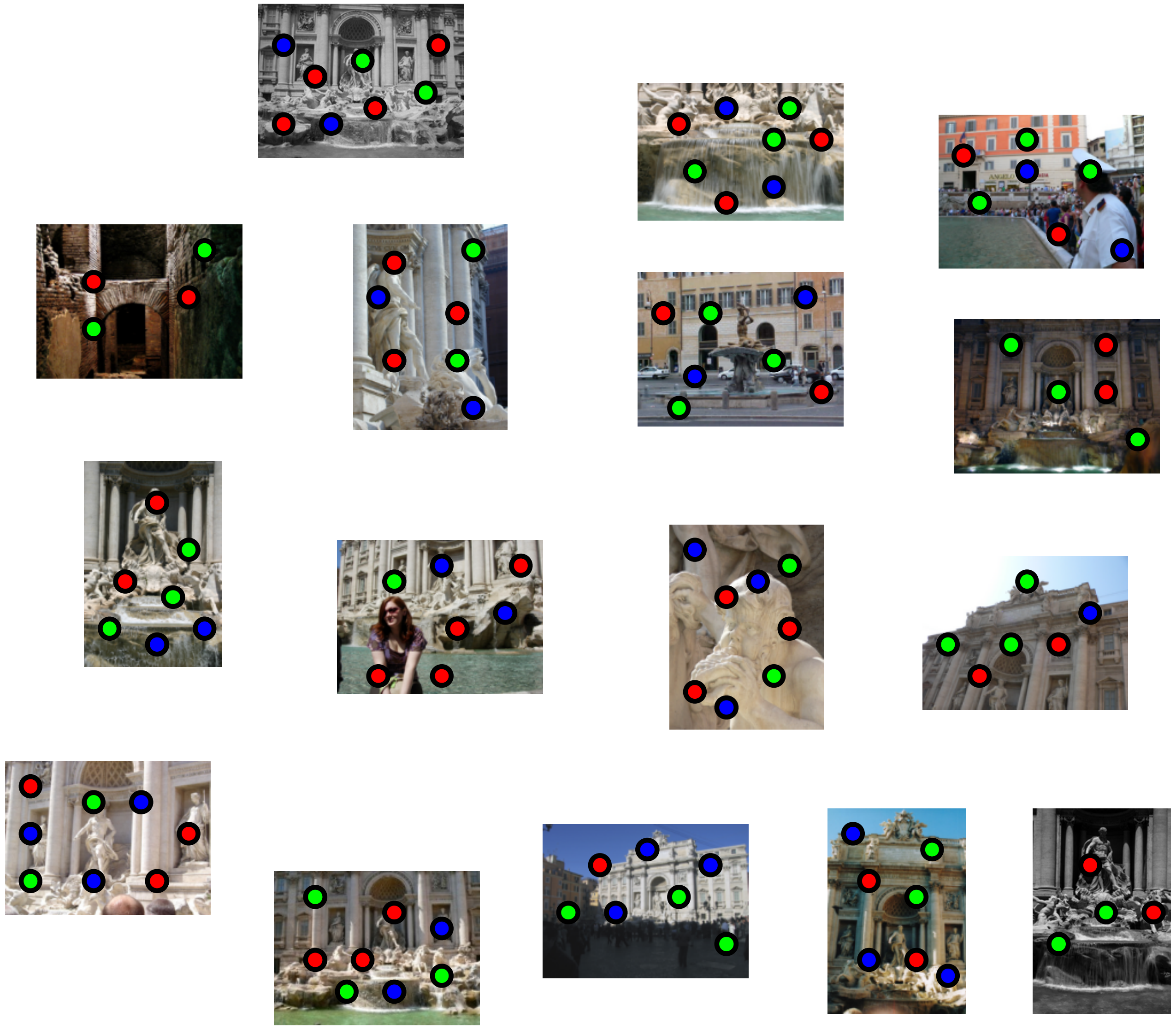
When F is unknown:

- ***Simultaneous*** Correspondence and F Estimation
  - With F, it is easier to compute correspondence
  - With correspondence, we can estimate F
- A Chicken-or-Egg problem



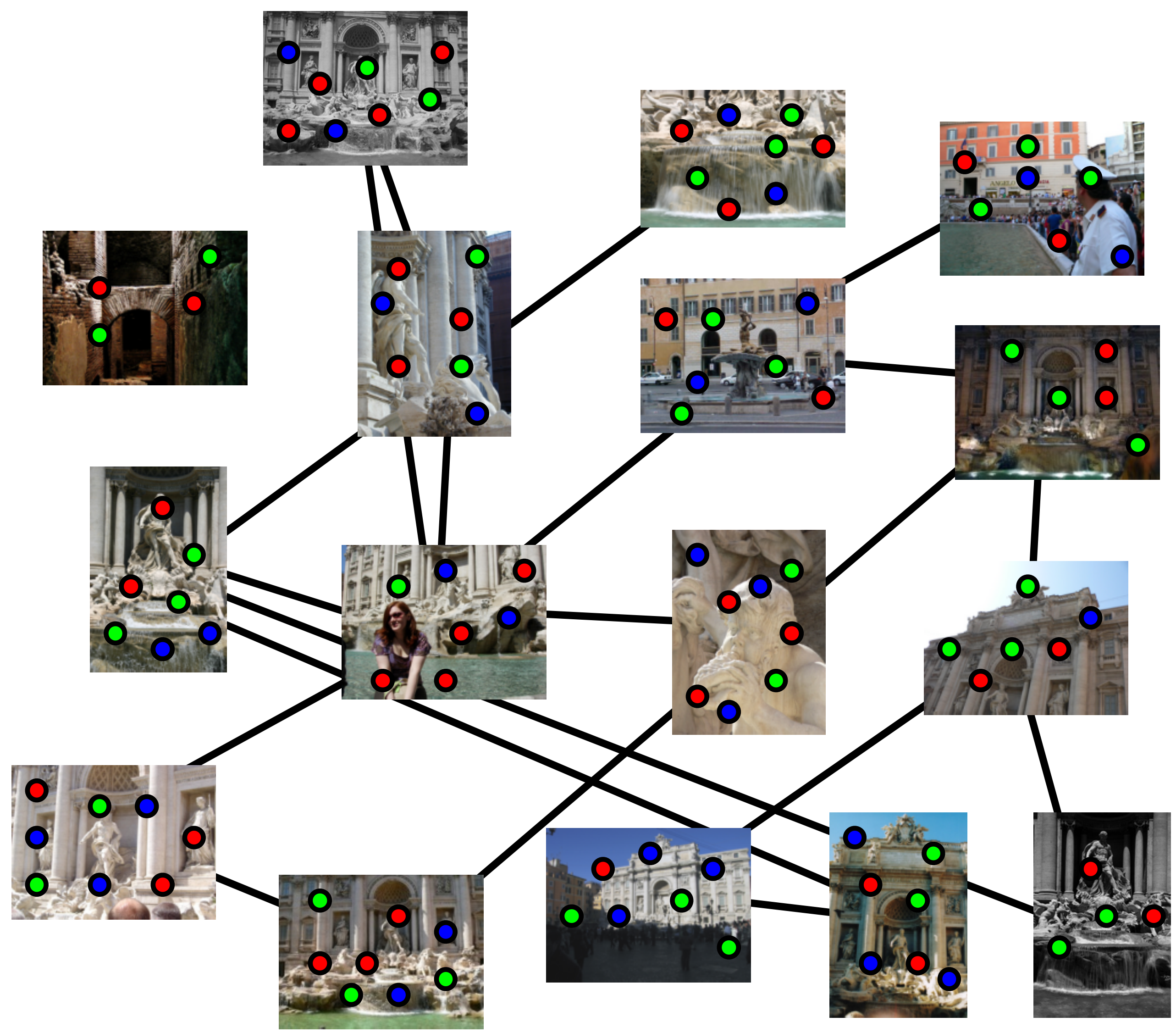
Basic Pipeline: Feature detection

Detect features using, for example, SIFT [Lowe, IJCV 2004] or learning-based keypoint detector

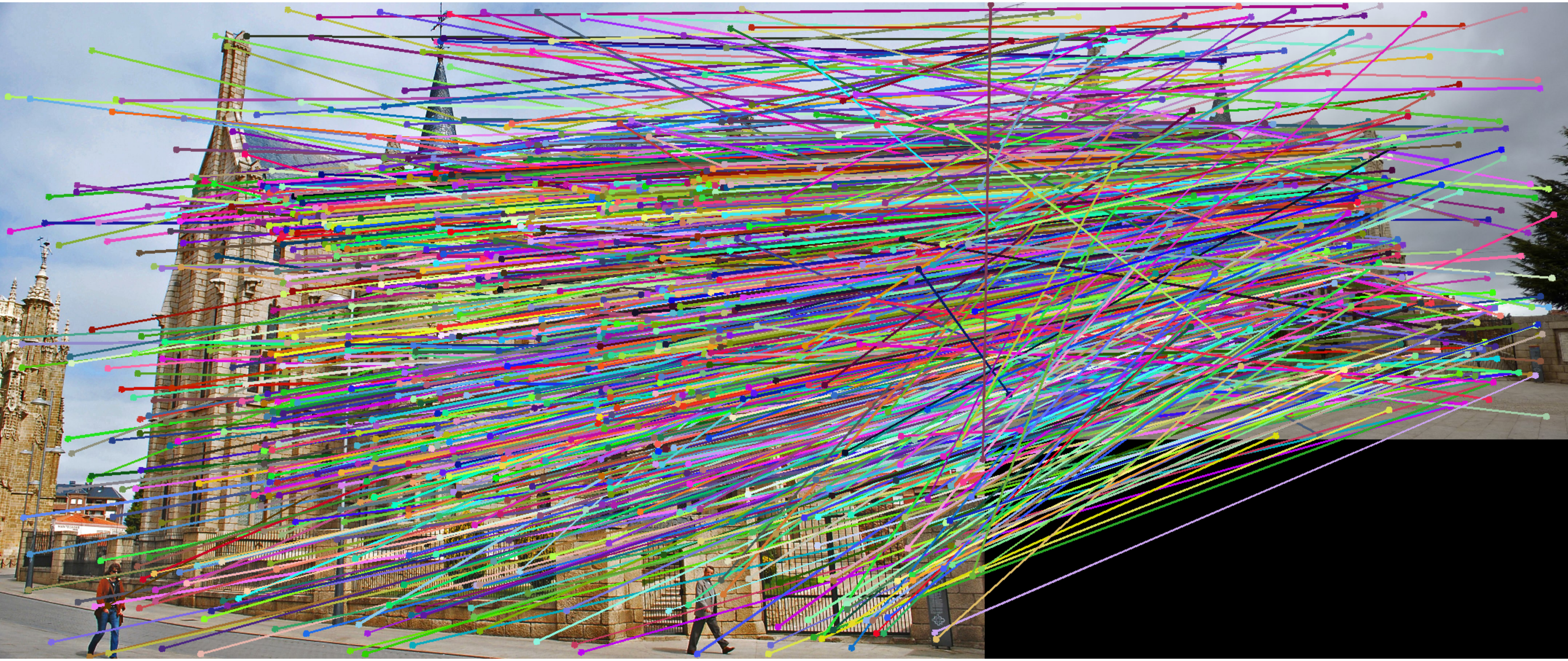


Basic Pipeline: Feature Matching

Match features between each pair of images

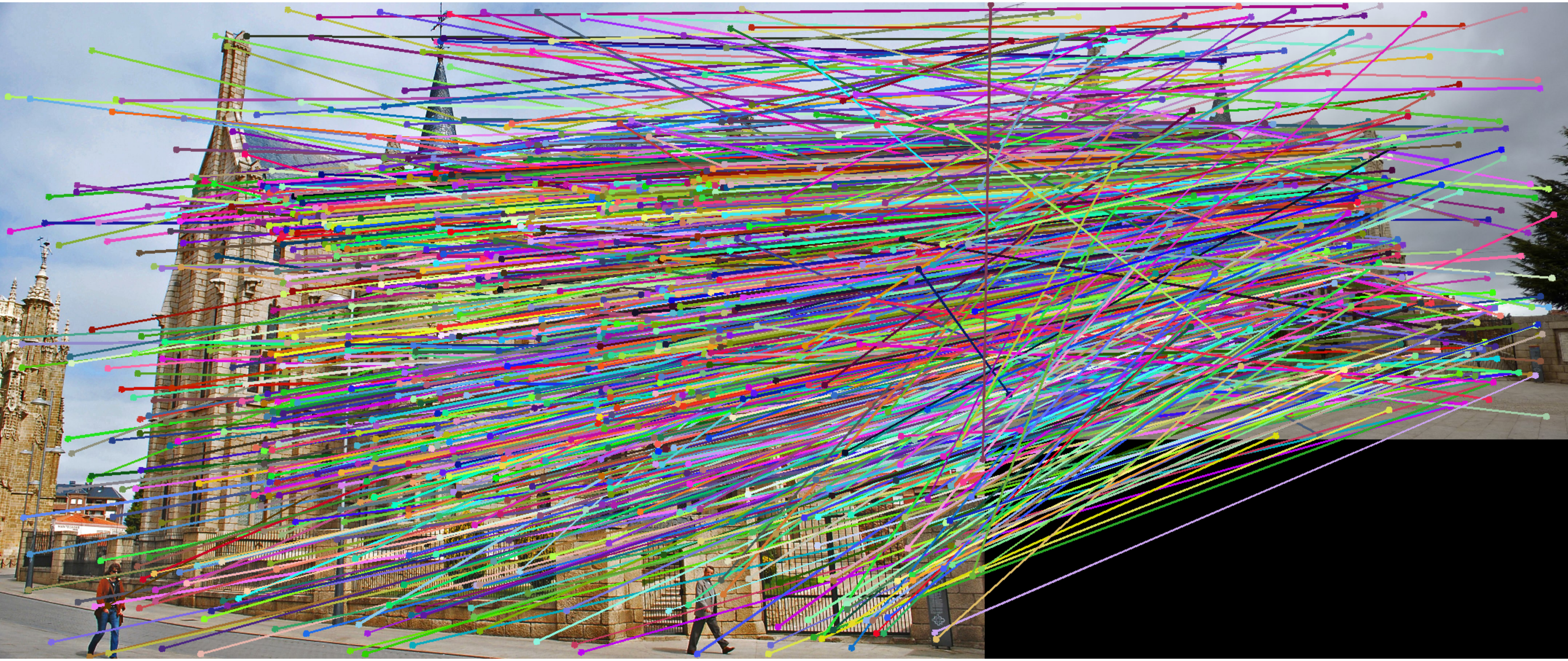


VLFeat's 800 most confident matches among 10,000+ local features.



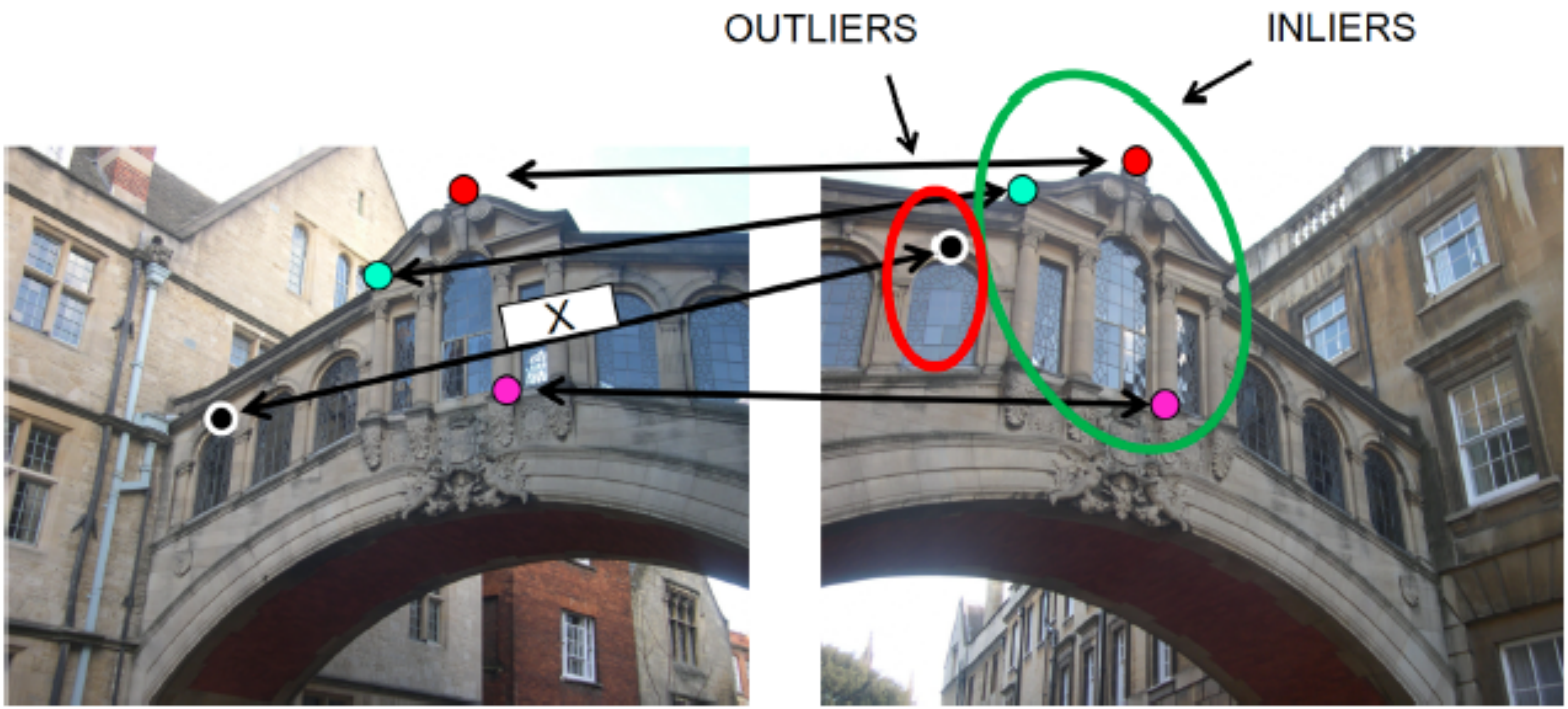
Basic Pipeline: Initial estimate of F (RANSAC)

Randomly choose some correspondences and estimate F



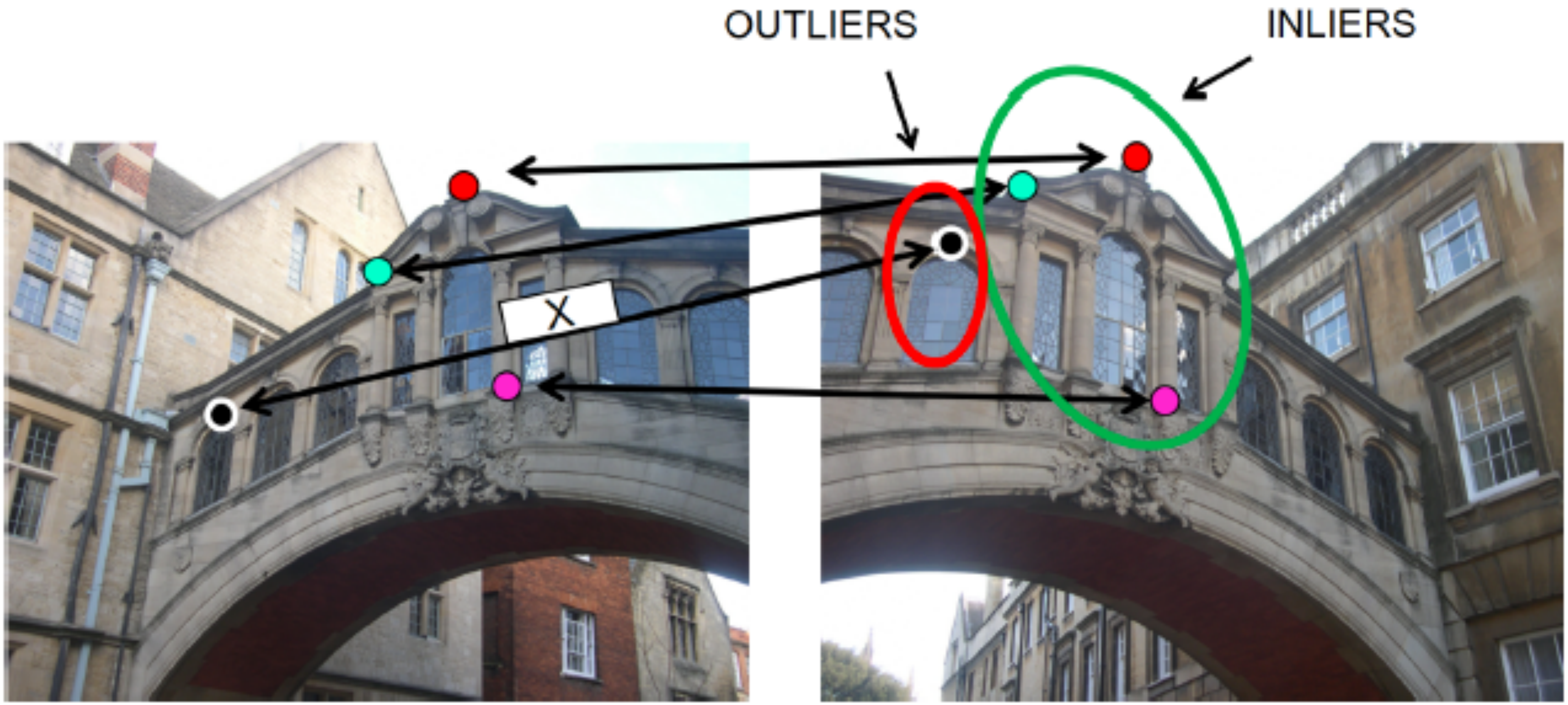
Basic Pipeline: Outlier removal (RANSAC)

Based on the estimated F, remove pairs with big errors



Basic Pipeline: Reestimate F (RANSAC)

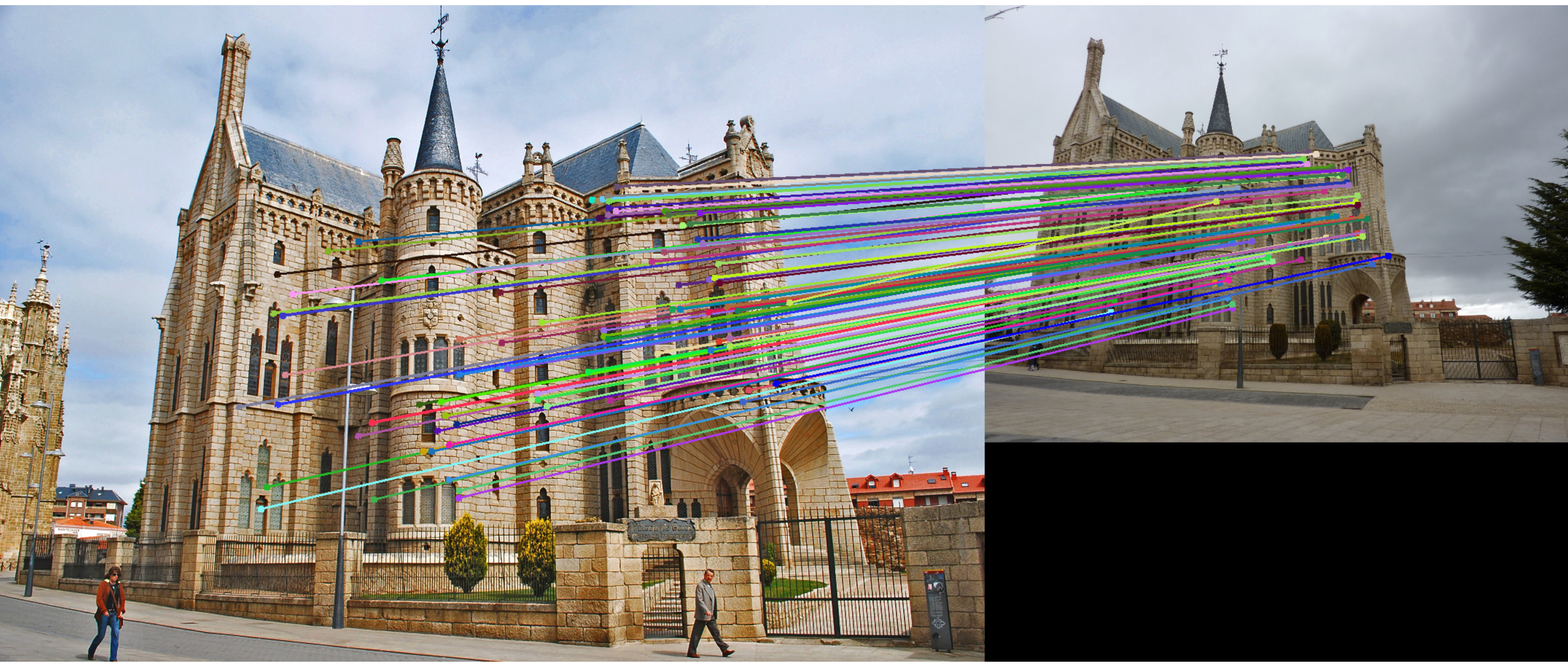
Reestimate F using inliners



Basic Pipeline: Repeat the above steps (RANSAC)

```
for i in range(n):
    randomly choose some pairs
    repeat for m times:
        based on the inliers, estimate F
        based on F, remove pairs with big errors
```

Keep only the matches at are "inliers" with respect to the "best" fundamental matrix



Agenda

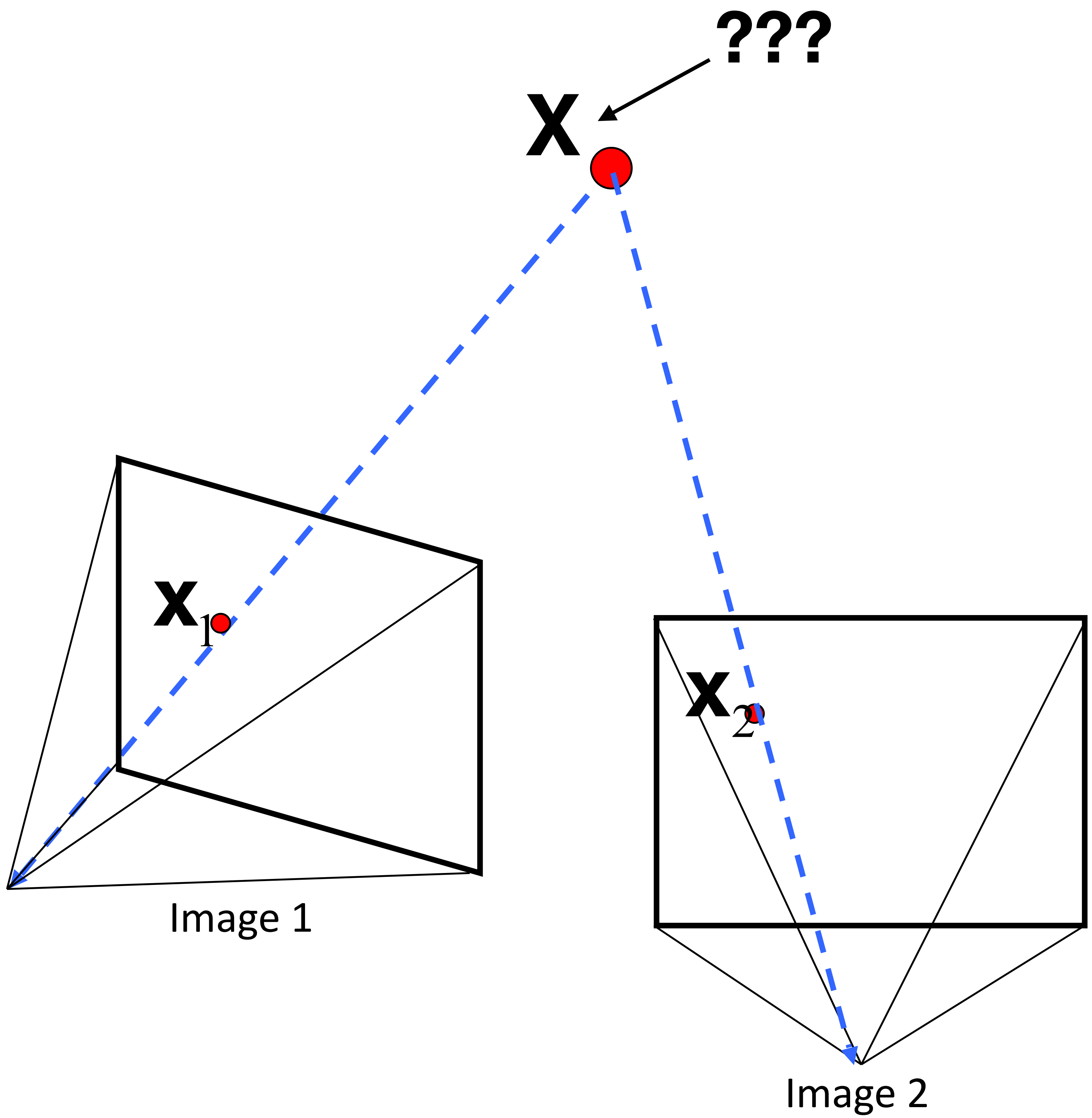
- Estimating F from Correspondences
- RANSAC for F Estimation
- **Multi-View 3D Reconstruction**

Problem setup

- Known:
  - Two views of the same scene
  - Corresponding points between views
  - Intrinsic camera matrices (K_1, K_2), i.e., camera calibration has been done
  - Fundamental matrix F

- Question: Point coordinates in 3D space

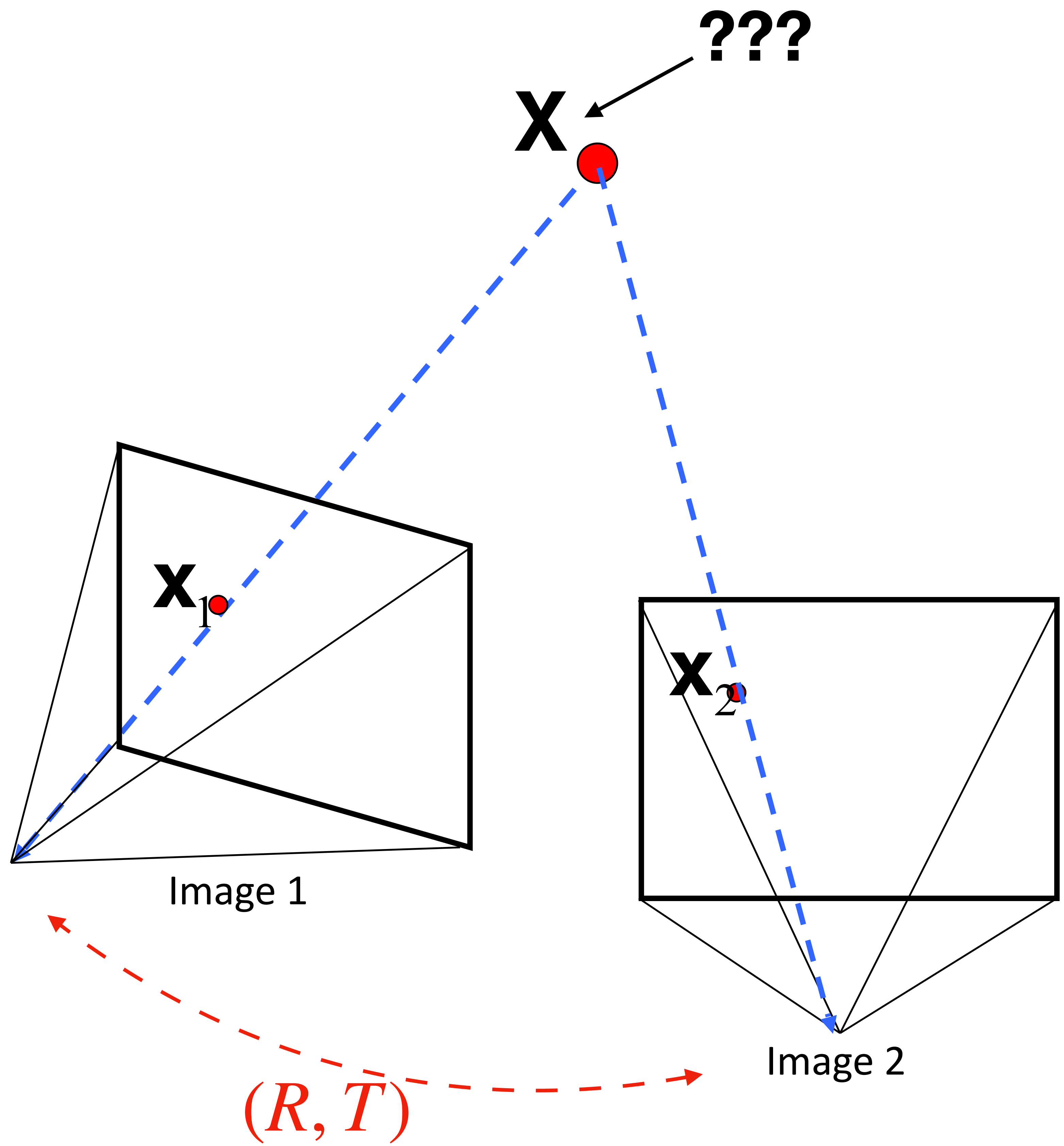
Problem Setup



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

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

Step I: Estimate (R,T) Between Cameras



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

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

Step I: Estimate (R,T) Between Cameras

- Get E from F:

$$F = K_1^{-T} E K_2^{-1}$$
$$E = K_1^{T} E K_2$$

- Decompose E into skew-symmetric and rotation matrices:

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

https://en.wikipedia.org/wiki/Essential_matrix

Step II: Reprojection Error Minimization

- With K_i, R, t, we can compute the projection matrices for both cameras:

$$P' = M\,P_w = K\begin{bmatrix} R & T \end{bmatrix} P_w$$

Internal parameters

External parameters

- The projective projection equation:

$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix} \qquad M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

in pixels

A non-linear transformation, denoted by $p_i = f(P_i)$

Step II: Reprojection Error Minimization

- Minimize sum of squared reproduction errors:

$$\underset{\{P_i\}}{\textbf{minimize}} \quad \sum_{i=1}^{n} \sum_{j=1}^{2} w_{ij} \left\| f(P_i; R_j, t_j) - \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} \right\|^2$$

w_{ij} → *indicator variable*: whether point *i* visible in image *j*

$f(P_i; R_j, t_j)$: *predicted* image location

$\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}$: *observed* image location

- Optimized with non-linear least squares
- LM algorithm (Levenberg-Marquardt) is a popular choice

Large systems built on these steps