CSE 152: Computer Vision

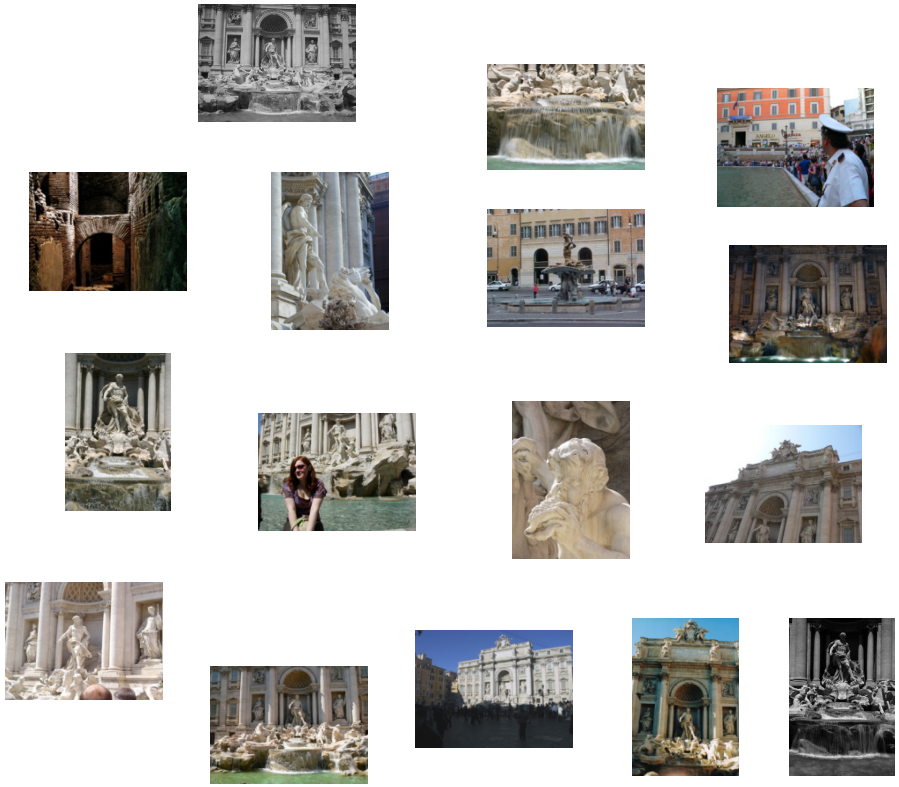
Hao Su

Lecture 13: Multi-View 3D Reconstruction

Some slides in this lecture are courtesy to Prof. Manmohan and Prof. Silvio Savarese

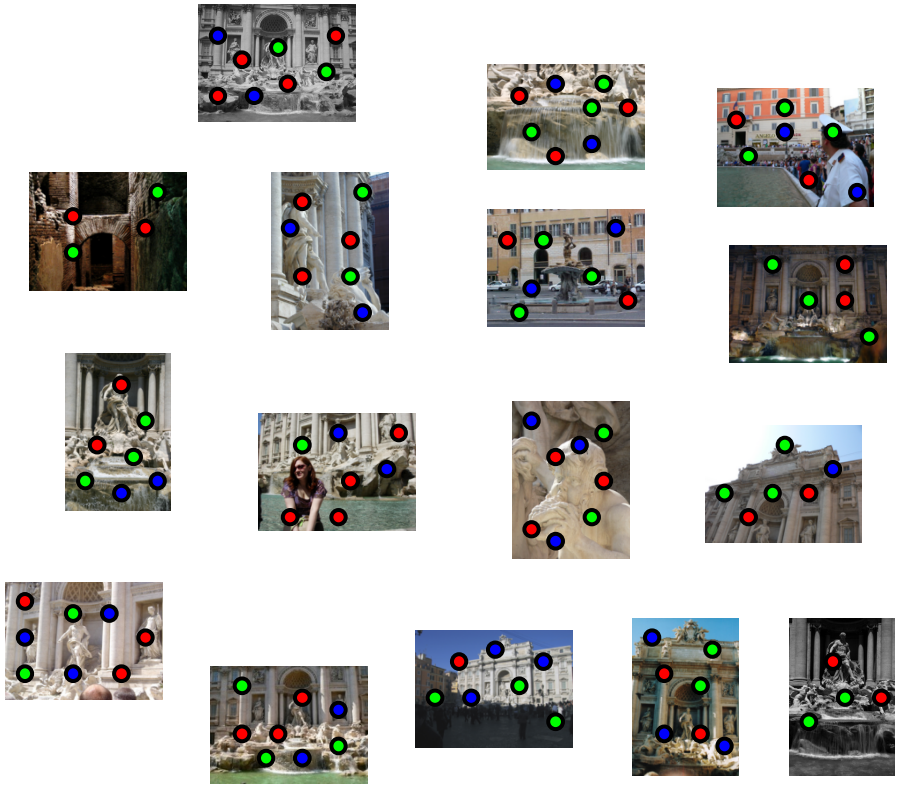
Structure from Motion

Several images observe a scene from different viewpoints



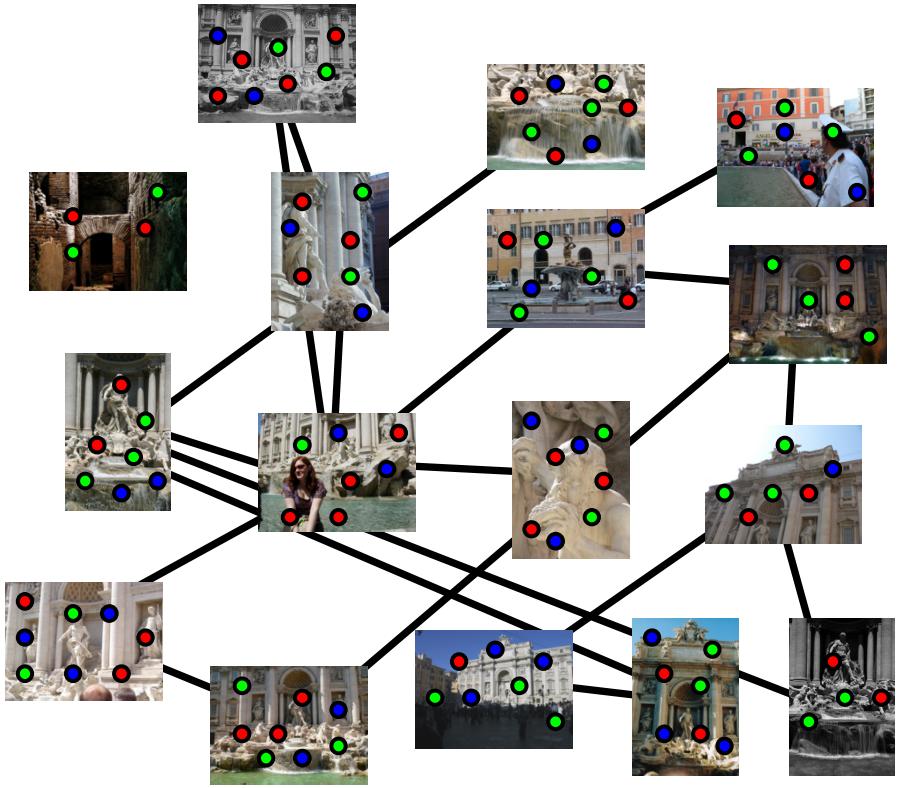
Feature detection

Detect features using, for example, SIFT [Lowe, IJCV 2004]

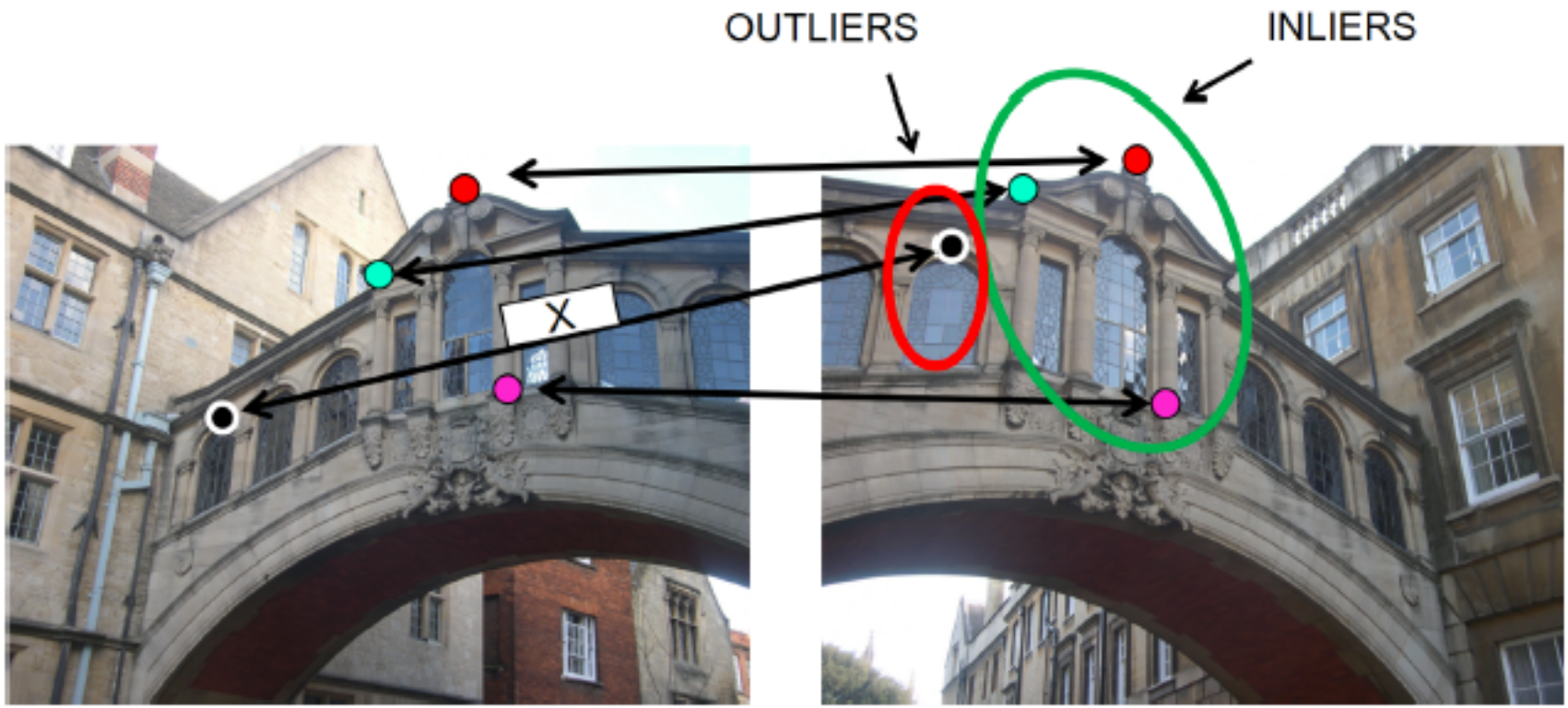


Feature matching

Match features between each pair of images

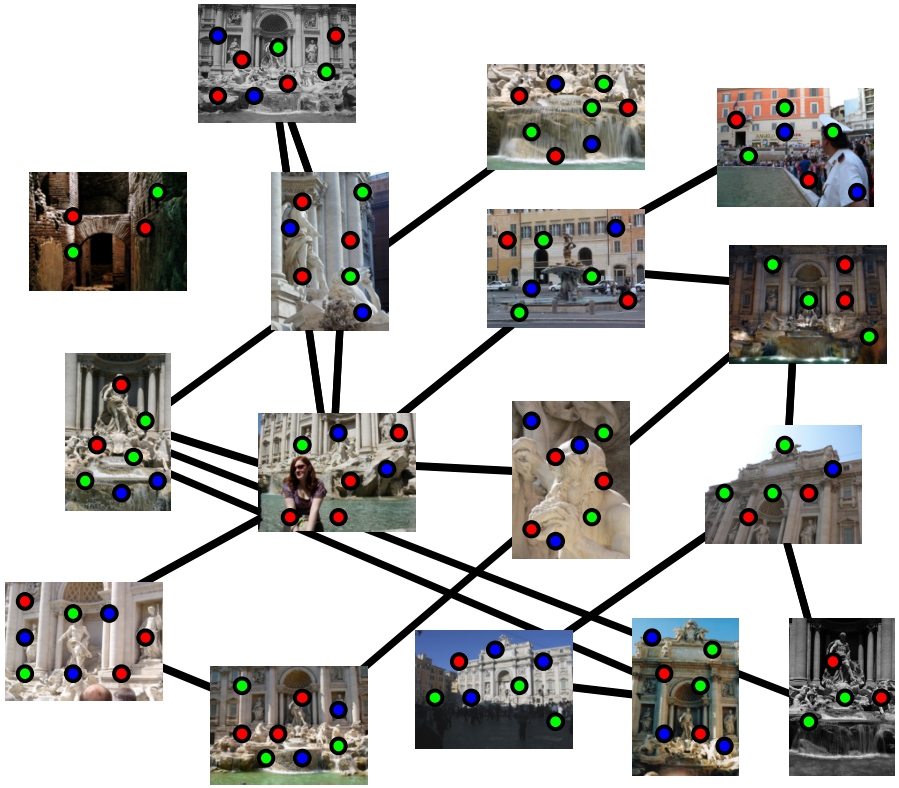


Feature matching

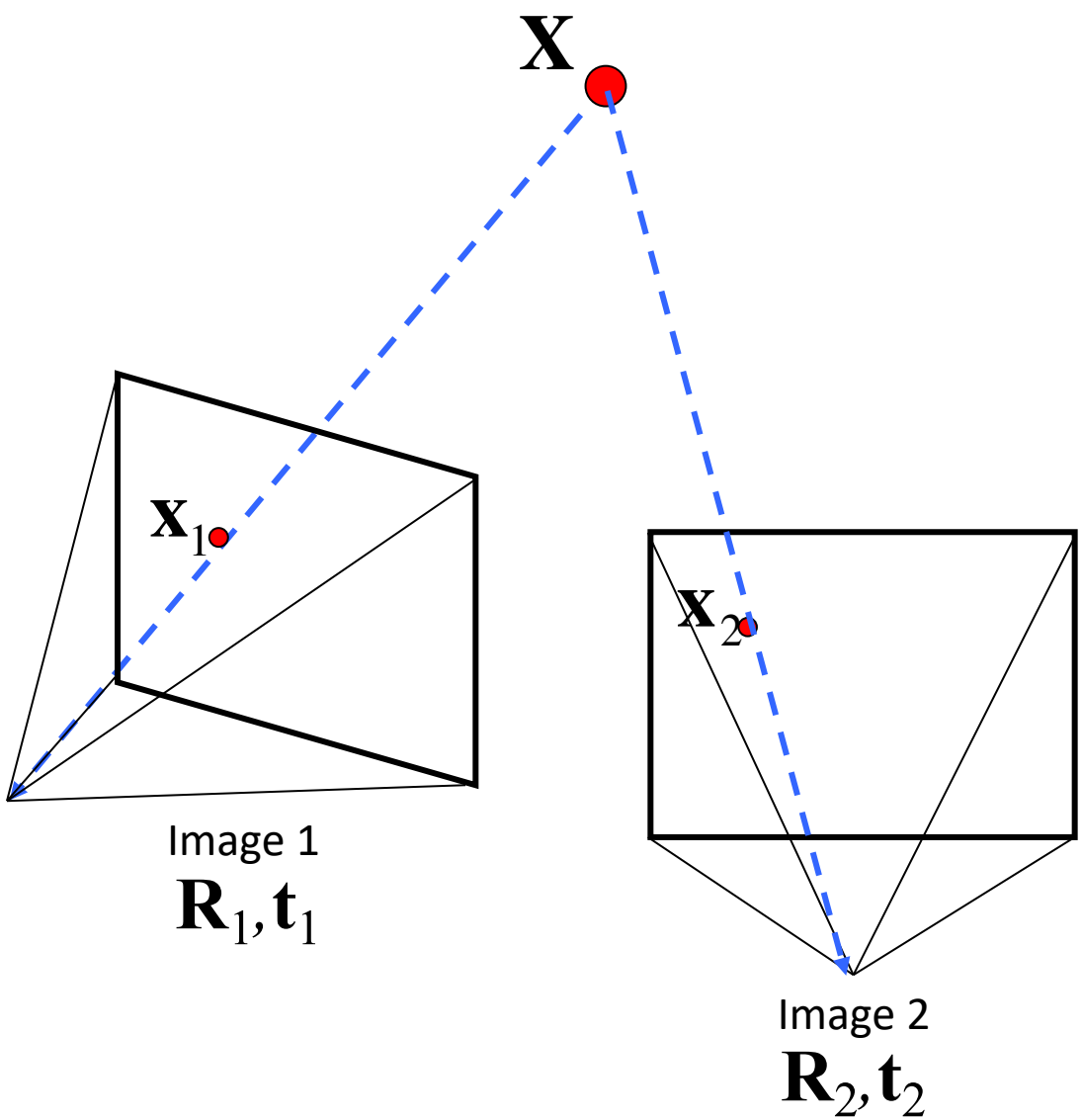


Feature matching

Refine matching using RANSAC to estimate fundamental matrix between each pair



Fundamental Matrix



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

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

8-point algorithm

Given *n* point correspondences, set up a system of equations:

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

- In reality, instead of solving $\mathbf{Af} = 0$, we seek $\mathbf{f}$ to minimize $\|\mathbf{Af}\|$.

8-point algorithm – Problem?

- **F** should have rank 2
- To enforce that **F** is of rank 2, **F** is replaced by **F'** subject to the rank constraint.

- This is achieved by SVD. Let $\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, where

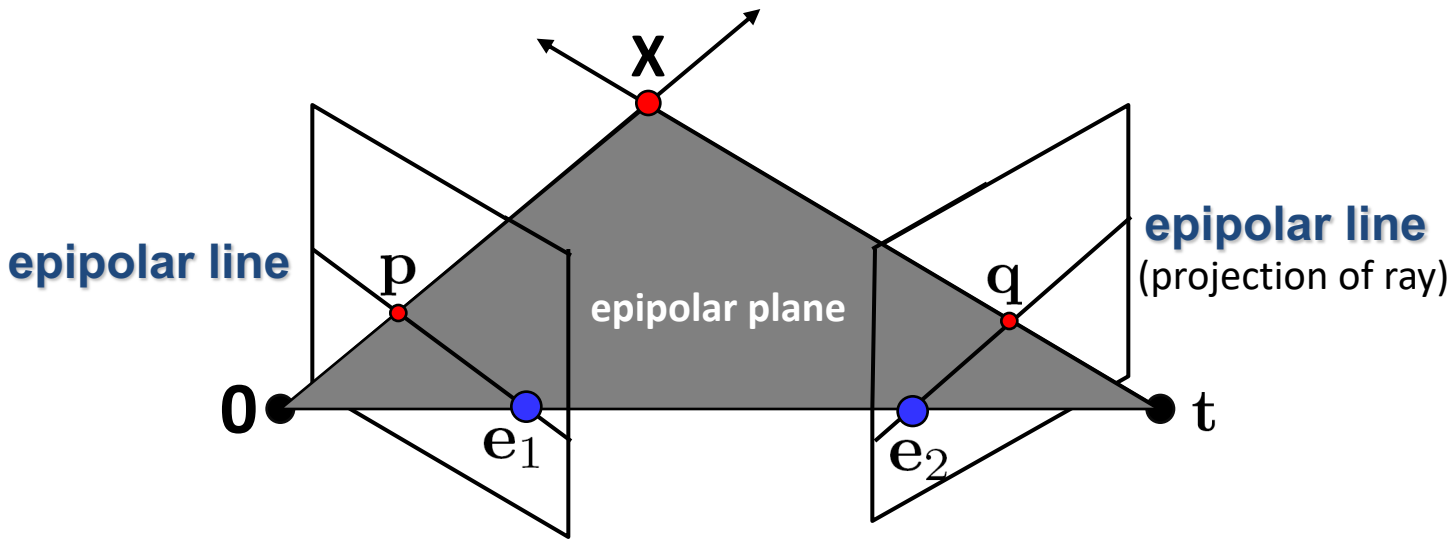
$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \text{ Let } \quad \Sigma' = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

then $\mathbf{F}' = \mathbf{U}\Sigma'\mathbf{V}^{\mathrm{T}}$ is the solution.

RANSAC

1. Randomly choose *s* samples (correspondences)
   - For fundamental matrix, what is the size of s?
2. Fit the model (fundamental matrix) to those samples
   - How would you fit the model to s correspondences?
3. Count the number of inliers among all other correspondences
   - How can you determine which points are inliers?
4. Repeat *N* times
   - How do you choose *N*?
5. Choose the model with the largest set of inliers

Essential matrix



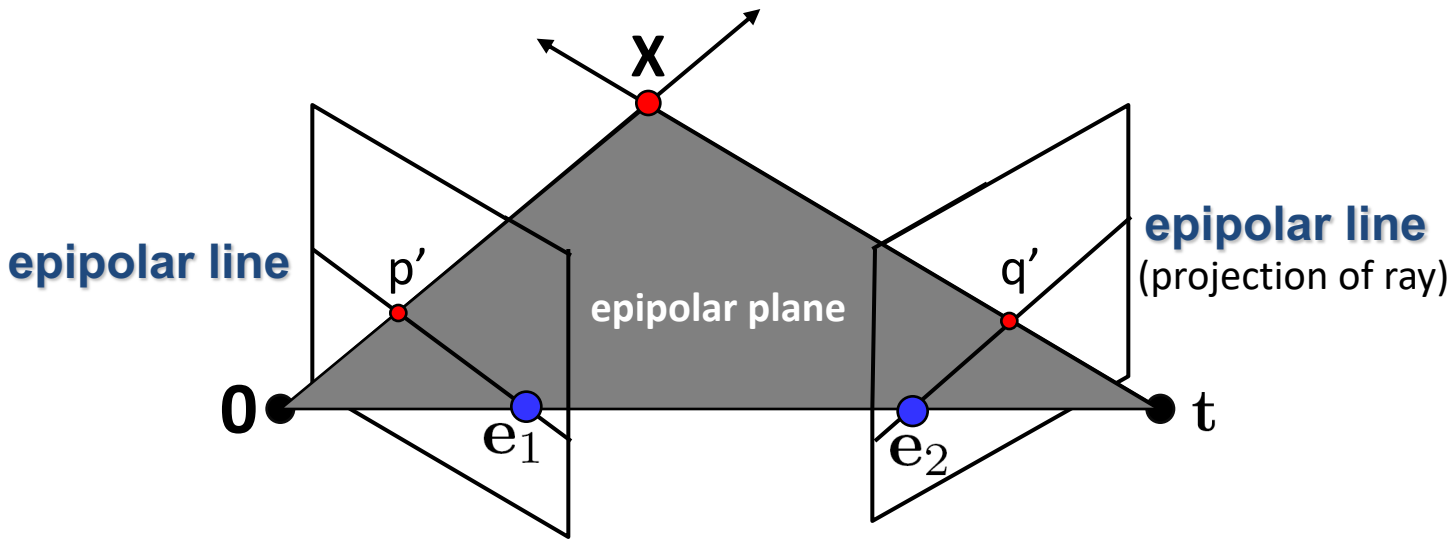
- We have: $\mathbf{q}^\top [\mathbf{t}]_\times \mathbf{R}\mathbf{p} = 0$
- Define:

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

- Then, we have:

$$\boxed{\mathbf{q}^\top \mathbf{E}\mathbf{p} = 0}$$

Fundamental matrix



- Essential matrix constraint in pixel space: $(\mathbf{K}_2^{-1}\mathbf{q}')^\top \mathbf{E}(\mathbf{K}_1^{-1}\mathbf{p}') = 0$.
- Rearranging: $$\mathbf{q}'^\top(\mathbf{K}_2^{-\top}\mathbf{E}\mathbf{K}_1^{-1})\mathbf{p}' = 0$$
- Define: $\mathbf{F} = \mathbf{K}_2^{-\top}\mathbf{E}\mathbf{K}_1^{-1}$
- Then, we have: $$\boxed{\mathbf{q}'^\top \mathbf{F}\mathbf{p}' = 0}$$

Motion from correspondences

- Use 8-point algorithm to estimate F
- Get E from F:

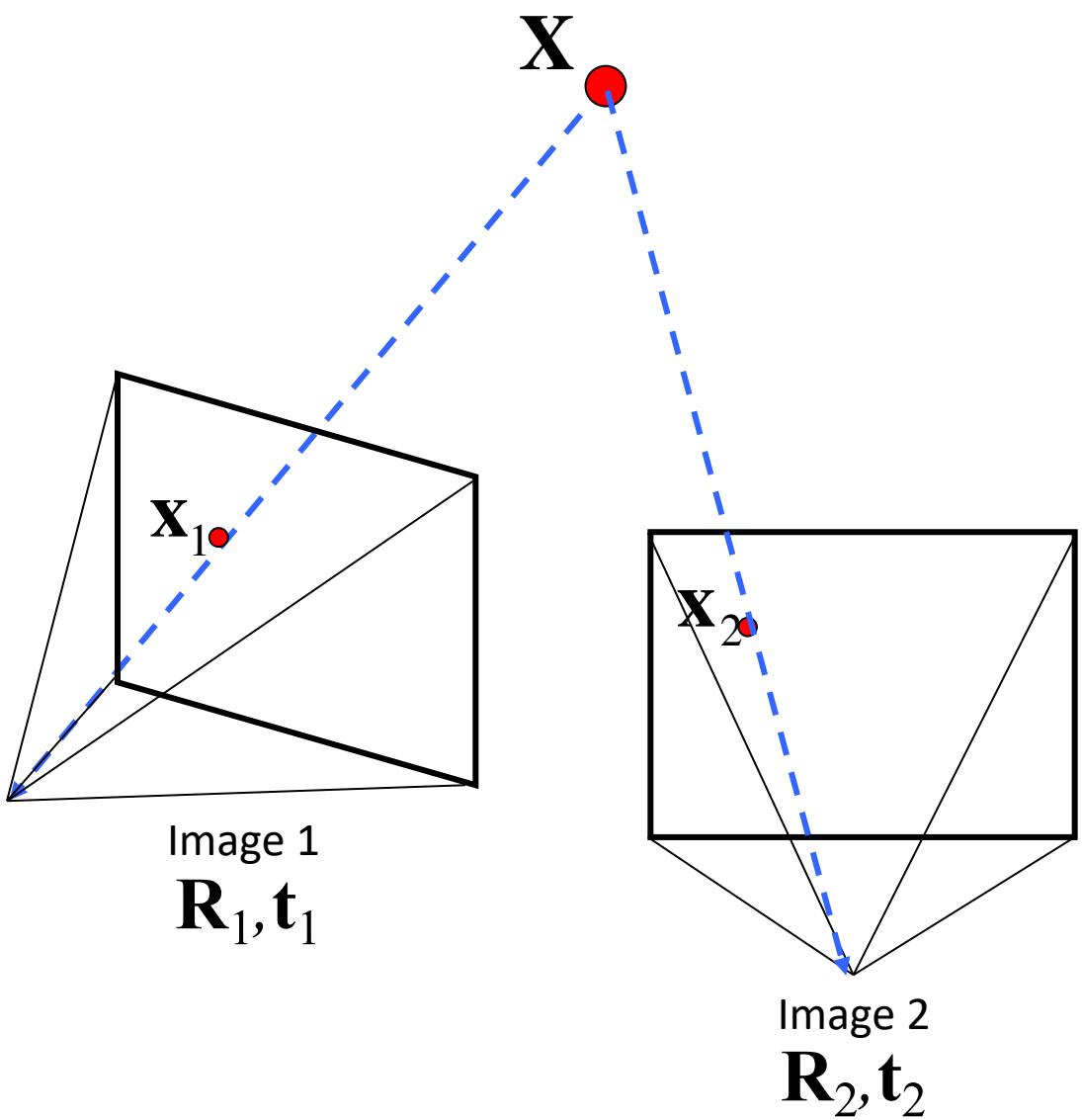
$$\mathbf{F} = \mathbf{K}_2^{-\top}\mathbf{E}\mathbf{K}_1^{-1}$$

$$\mathbf{E} = \mathbf{K}_2{}^{\top}\mathbf{F}\mathbf{K}_1$$

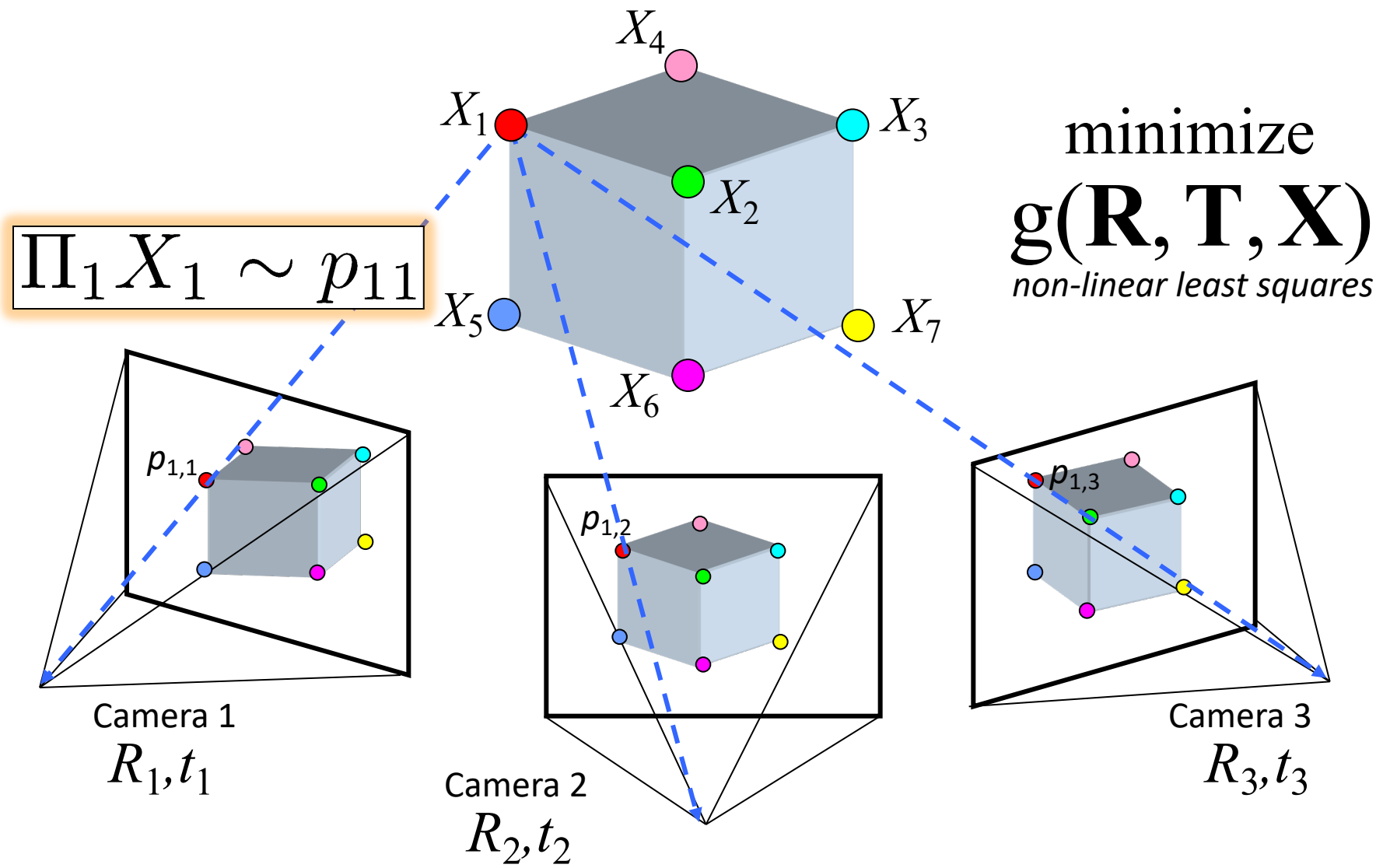
- Decompose E into skew-symmetric and rotation matrices:

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

Triangulation



Structure from motion



Bundle adjustment

- Minimize sum of squared reprojection errors:

$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| \mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j) - \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} \right\|^2$$

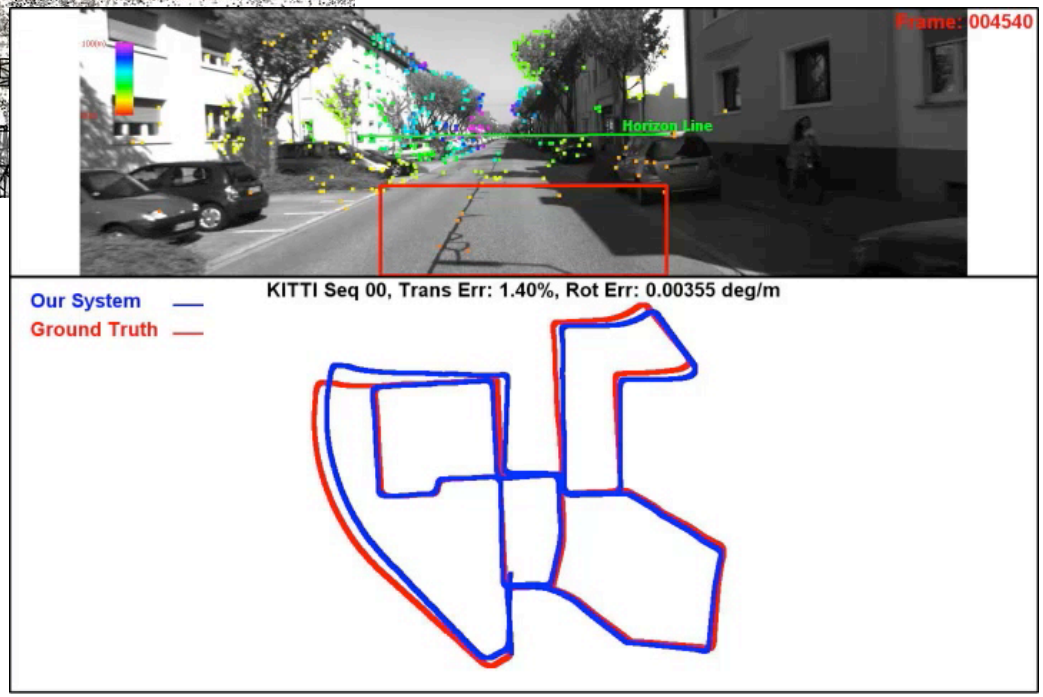
indicator variable: whether point *i* visible in image *j*

predicted image location

observed image location

  - Optimized with non-linear least squares
  - Levenberg-Marquardt is a popular choice
- Practical challenges?
  - Initialization
  - Outliers

Large systems built on these steps



Benchmark datasets

The KITTI Vision Benchmark Suite

A project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago

Karlsruhe Institute of Technology

home | setup | stereo | flow | scene flow | odometry | object | tracking | road | semantics | raw data | submit results | jobs

Andreas Geiger (MPI Tübingen) | Philip Lenz (KIT) | Christoph Stiller (KIT) | Raquel Urtasun (University of Toronto)

Visual Odometry / SLAM Evaluation 2012

The odometry benchmark consists of 22 stereo sequences, saved in loss less png format: We provide 11 sequences (00-10) with ground truth trajectories for training and 11 sequences (11-21) without ground truth for evaluation. For this benchmark you may provide results using monocular or stereo visual odometry, laser-based SLAM or algorithms that combine visual and LIDAR information. The only restriction we impose is that your method is fully automatic (e.g., no manual loop-closure tagging is allowed) and that the same parameter set is used for all sequences. A development kit provides details about the data format.

- Download odometry data set (grayscale, 22 GB)
- Download odometry data set (color, 65 GB)
- Download odometry data set (velodyne laser data, 80 GB)
- Download odometry data set (calibration files, 1 MB)
- Download odometry ground truth poses (4 MB)
- Download odometry development kit (1 MB)

Benchmark datasets

Additional information used by the methods

- Stereo: Method uses left and right (stereo) images
- Laser Points: Method uses point clouds from Velodyne laser scanner
- Loop Closure Detection: This method is a SLAM method that detects loop closures
- Additional training data: Use of additional data sources for training (see details)

	Method	Setting	Code	Translation	Rotation	Runtime	Environment	Compare
1	V-LOAM			**0.68 %**	**0.0015 [deg/m]**	0.1 s	2 cores @ 2.5 Ghz (C/C++)	
J. Zhang and S. Singh: Visual-lidar Odometry and Mapping: Low drift, Robust, and Fast. IEEE International Conference on Robotics and Automation(ICRA) 2015.								
2	LOAM			0.70 %	0.0017 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	
J. Zhang and S. Singh: LOAM: Lidar Odometry and Mapping in Real-time. Robotics: Science and Systems Conference (RSS) 2014.								
3	SOFT2			0.72 %	0.0016 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	
4	LG-SLAM			0.82 %	0.0020 [deg/m]	0.2 s	2 cores @ 2.5 Ghz (C/C++)	
5	GDVO			0.86 %	0.0031 [deg/m]	0.09 s	1 core @ >3.5 Ghz (C/C++)	
6	HypERROCC			0.88 %	0.0027 [deg/m]	0.25 s	2 cores @ 2.0 Ghz (C/C++)	
7	SOFT			0.88 %	0.0022 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	
I. Cvišić and I. Petrović: Stereo odometry based on careful feature selection and tracking. European Conference on Mobile Robots (ECMR) 2015.								
8	RotRocc			0.88 %	0.0025 [deg/m]	0.3 s	2 cores @ 2.0 Ghz (C/C++)	
M. Buczko and V. Willert: Flow-Decoupled Normalized Reprojection Error for Visual Odometry. 19th IEEE Intelligent Transportation Systems Conference (ITSC) 2016.								
9	EDVO			0.89 %	0.0030 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)	
10	svo2			0.94 %	0.0021 [deg/m]	0.2 s	1 core @ 2.5 Ghz (C/C++)	