Structure from Motion (SfM)

Jiayuan Gu

gujy1@shanghaitech.edu.cn

Many slides modified from Prof. Svetlana Lazebnik (CS543)

Outline

- Problem Formulation
- Projective structure from motion
- SfM pipeline

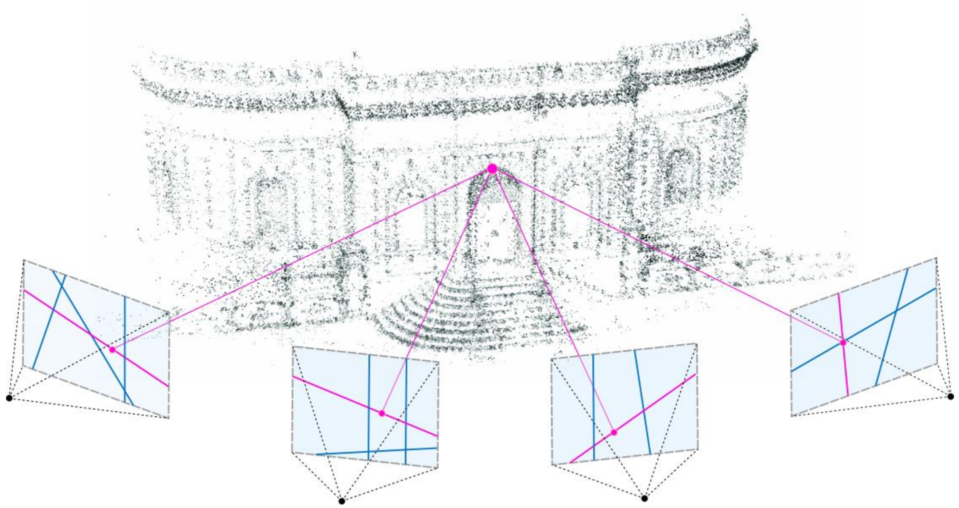
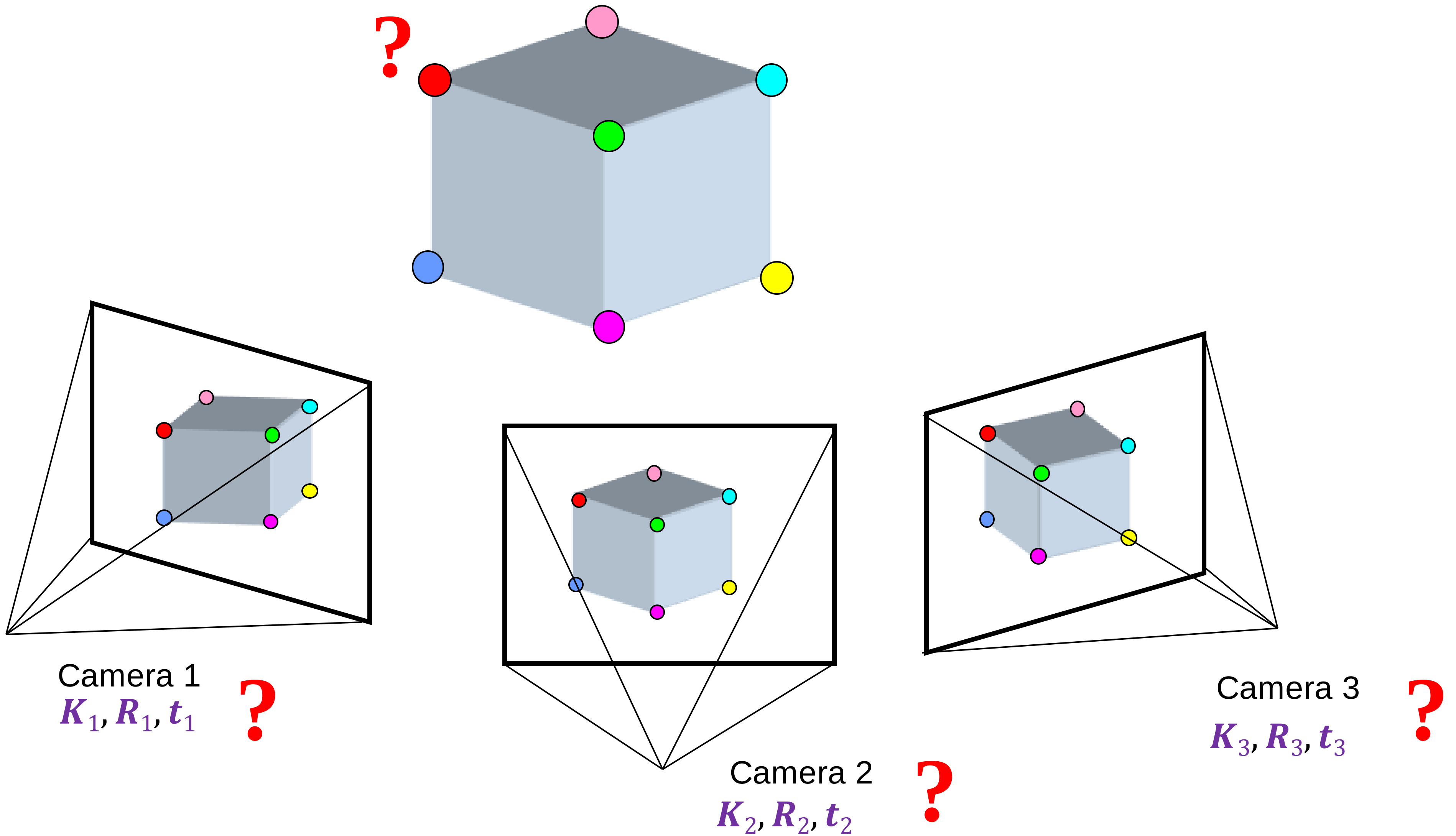
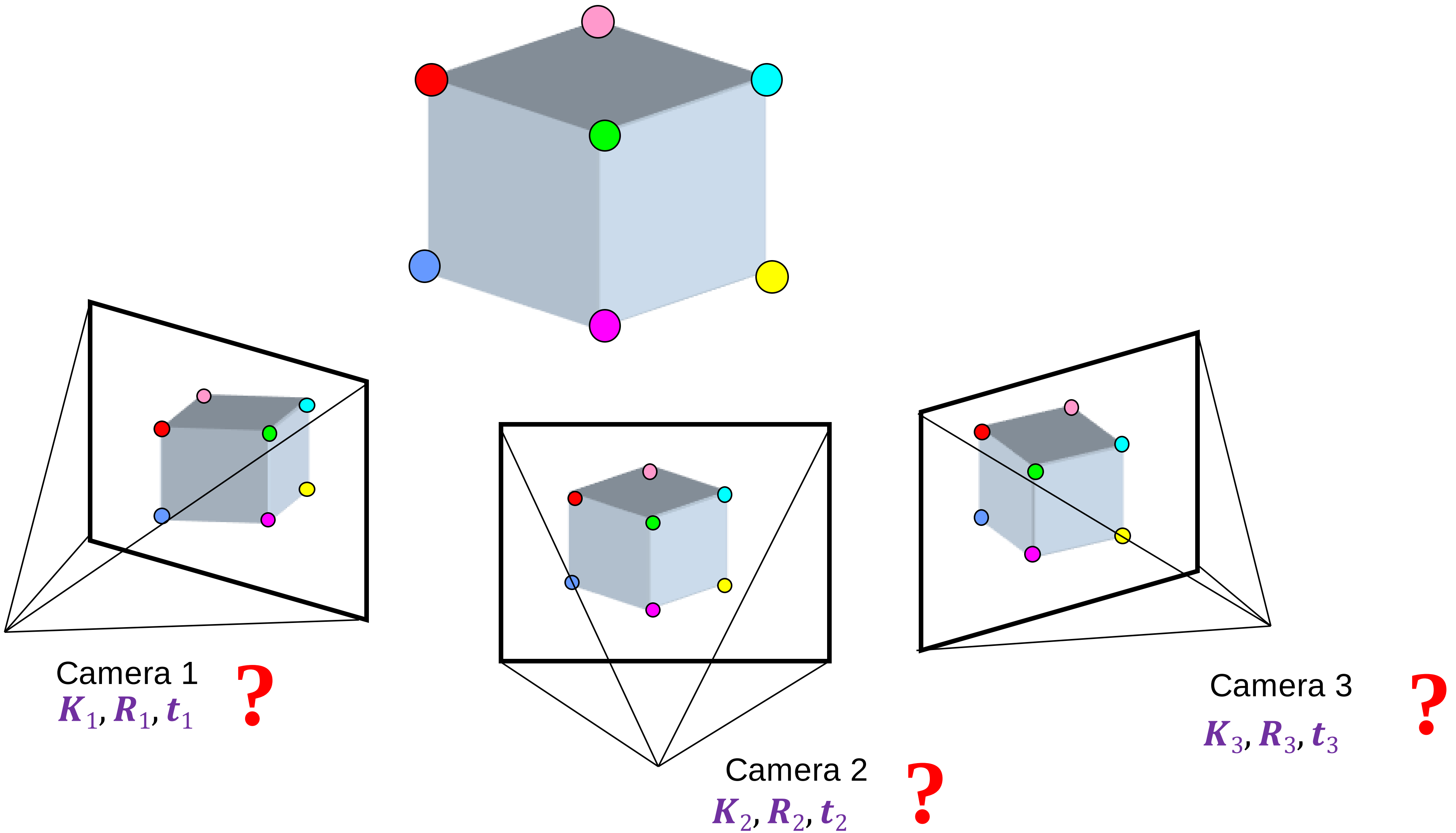


Figure from blog

Structure from motion

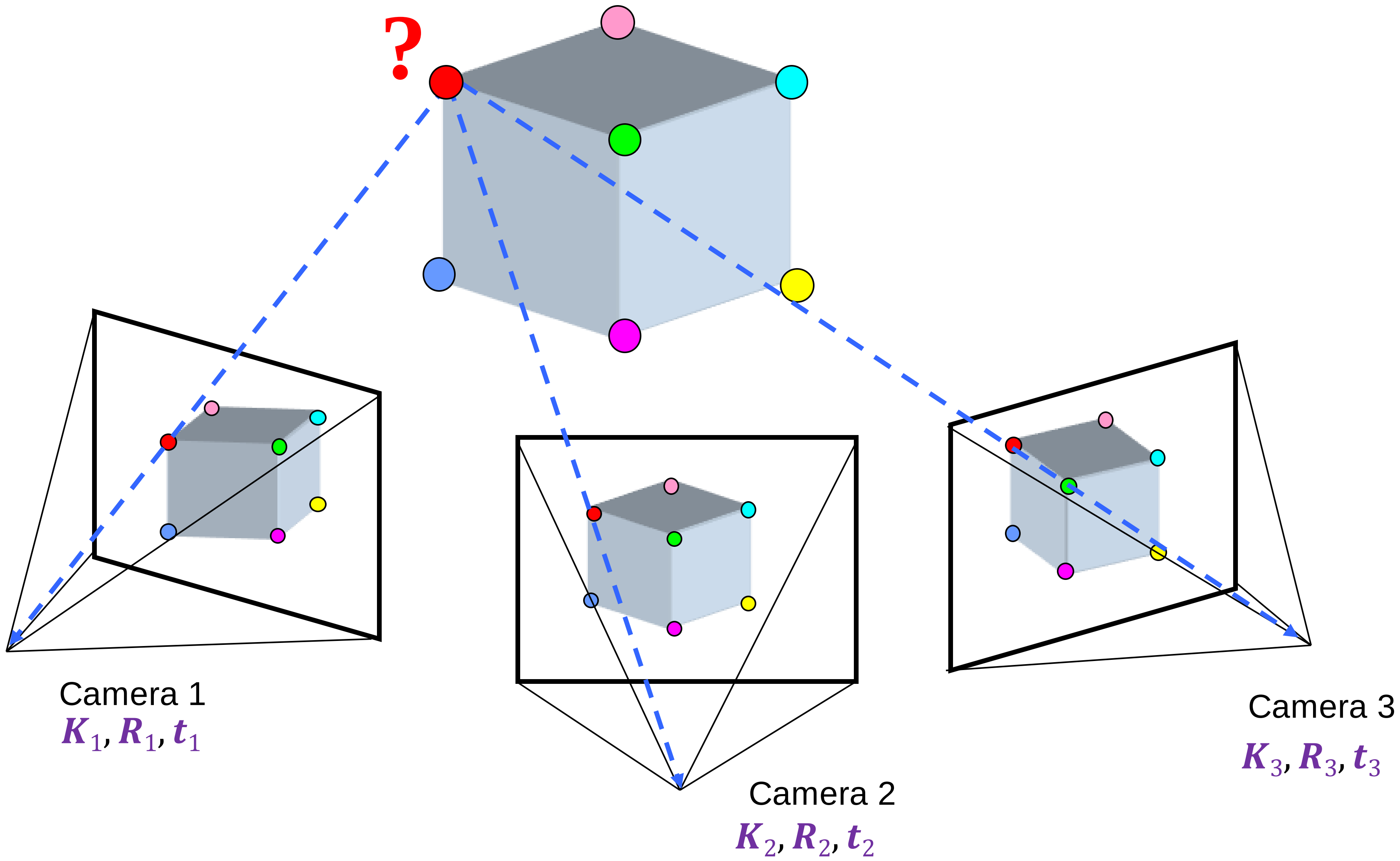


Recall: Calibration



Given a set of *known* 3D points seen by a camera, compute the camera parameters

Recall: Triangulation



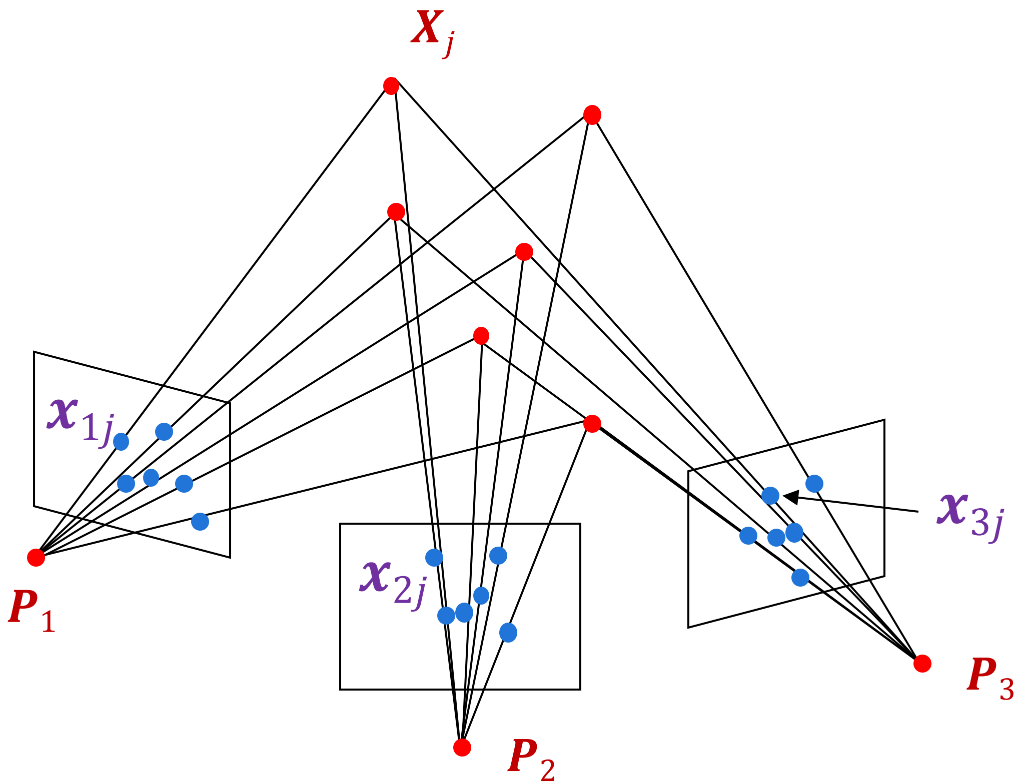
Given *known cameras* and projections of the same 3D point in two or more images, compute the 3D coordinates of that point

Structure from Motion: Problem formulation

- Given: m images of n fixed 3D points such that (ignoring visibility)

$$\boldsymbol{x}_{ij} \cong \boldsymbol{P}_i \boldsymbol{X}_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

- Problem: estimate m projection matrices $\boldsymbol{P}_i$ and n 3D points $\boldsymbol{X}_j$ from the mn correspondences $\boldsymbol{x}_{ij}$



Structure from Motion Ambiguity

- If we scale the entire scene by some factor k and, at the same time, scale the camera matrices by the factor of $1/k$, the projections of the scene points remain exactly the same:

$$\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = \left(\frac{1}{k}\boldsymbol{P}\right)(k\boldsymbol{X})$$

- Without a reference measurement, it is impossible to recover the absolute scale of the scene!
- In general, if we transform the scene using a transformation $\boldsymbol{Q}$ and apply the inverse transformation to the camera matrices, then the image observations do not change:

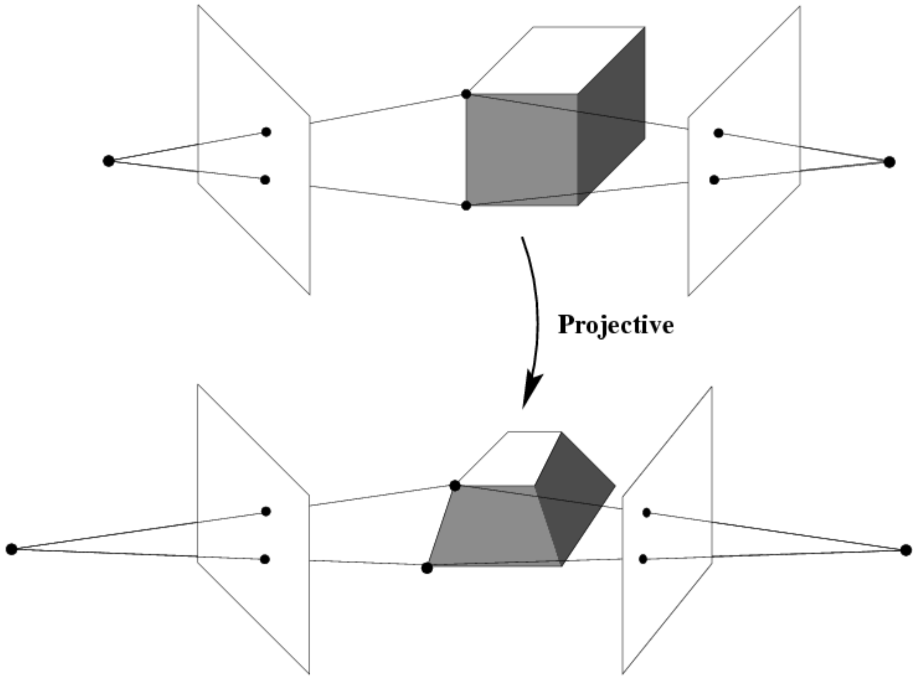
$$\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = (\boldsymbol{P}\boldsymbol{Q}^{-1})(\boldsymbol{Q}\boldsymbol{X})$$

Projective Ambiguity

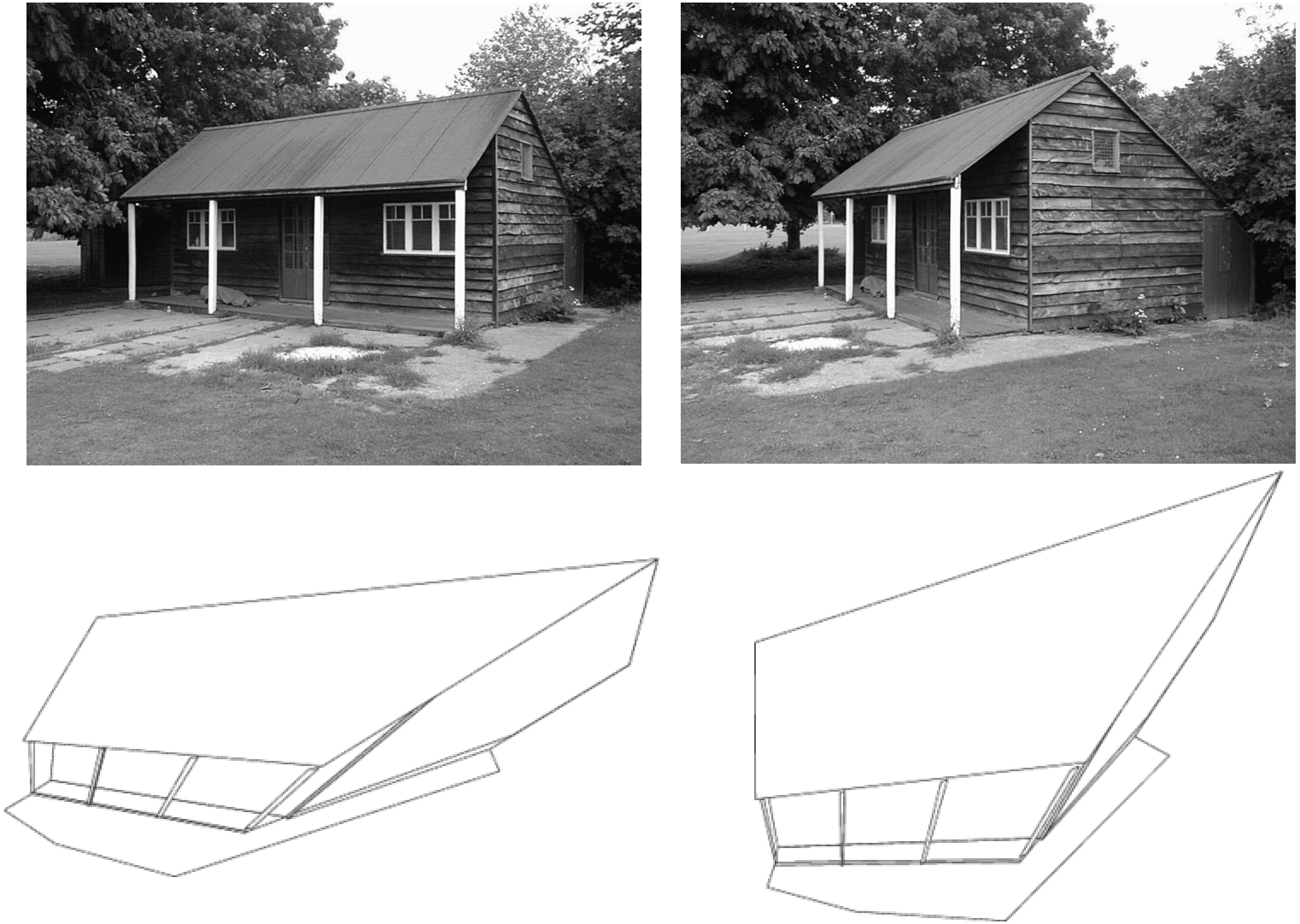
- With no constraints on the camera calibration matrices or on the scene, we can reconstruct up to a *projective* ambiguity:

$$\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = (\boldsymbol{P}\boldsymbol{Q}^{-1})(\boldsymbol{Q}\boldsymbol{X})$$

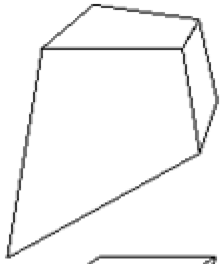
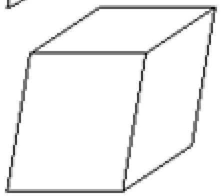
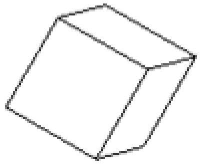
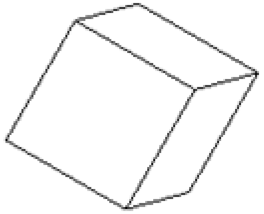
$\boldsymbol{Q}$ is a general full-rank 4×4 matrix



Projective Ambiguity



Types of Transformation

Projective 15dof	$\begin{bmatrix} A & t \\ v^{T} & v \end{bmatrix}$		Preserves intersection and tangency
Affine 12dof	$\begin{bmatrix} A & t \\ 0^{T} & 1 \end{bmatrix}$		Preserves parallellism, volume ratios
Similarity 7dof	$\begin{bmatrix} sR & t \\ 0^{T} & 1 \end{bmatrix}$		Preserves angles, ratios of length
Euclidean 6dof	$\begin{bmatrix} R & t \\ 0^{T} & 1 \end{bmatrix}$		Preserves angles, lengths

Projective Structure from Motion

- **Given**: m images of n fixed 3D points such that (ignoring visibility):
  - $\boldsymbol{x}_{ij} \cong \boldsymbol{P}_i \boldsymbol{X}_j$, $\quad i = 1, \dots, m, \quad j = 1, \dots, n$
- **Problem**: estimate m projection matrices $\boldsymbol{P}_i$ and n 3D points $\boldsymbol{X}_j$ from the mn correspondences $\boldsymbol{x}_{ij}$
- With no calibration info, cameras and points can only be recovered up to a 4×4 projective transformation $\boldsymbol{Q}$:
  - $\boldsymbol{X} \rightarrow \boldsymbol{QX}, \boldsymbol{P} \rightarrow \boldsymbol{PQ}^{-1}$
- We can solve for structure and motion when $2mn \geq 11m + 3n - 15$
- For two cameras, at least 7 points are needed

Projective SFM: Two-Camera Case

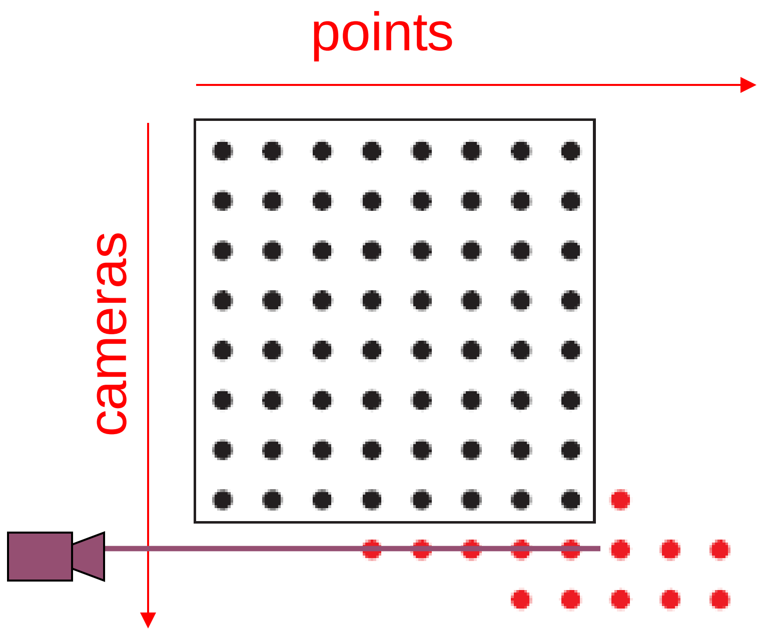
1. Estimate fundamental matrix $\boldsymbol{F}$ between the two views
2. Set first camera matrix to $[\boldsymbol{I} \mid \boldsymbol{0}]$
3. Then the second camera matrix is given by $[\boldsymbol{A} \mid \boldsymbol{t}]$ where $\boldsymbol{t}$ is the epipole ($\boldsymbol{F}^T\boldsymbol{t} = \boldsymbol{0}$) and $\boldsymbol{A} = -[\boldsymbol{t}_\times]\boldsymbol{F}$

- In practice, SFM pipelines use the guesses of intrinsic parameters and the five-point algorithm (for essential matrix)
  - Recall 7-point or 8-point algorithm for fundamental matrix

See *Multiple View Geometry* Sec 9.5

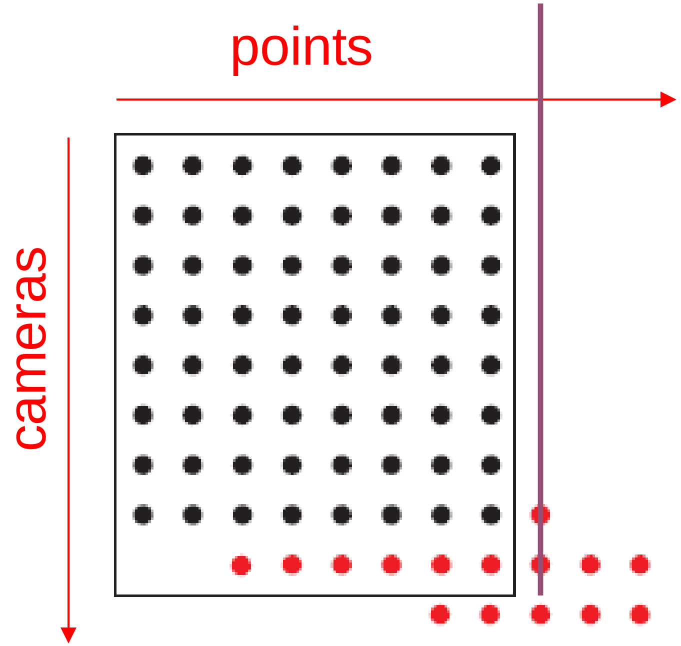
Incremental Structure from Motion

- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – calibration



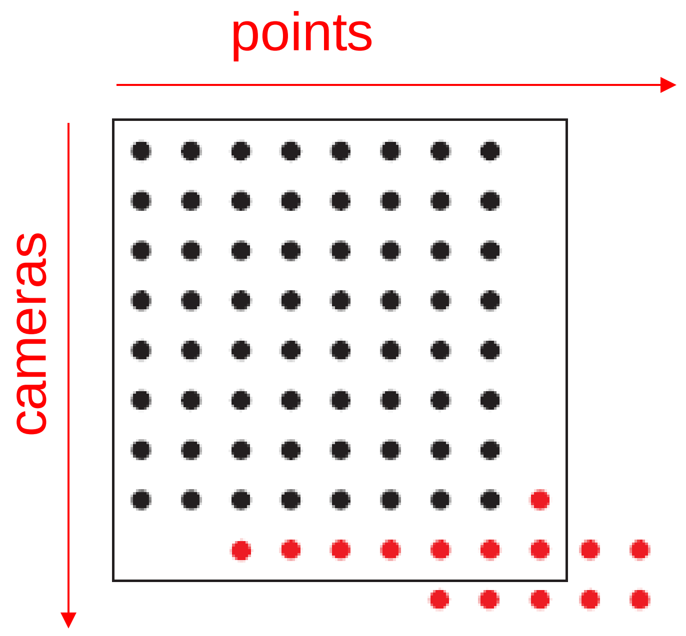
Incremental Structure from Motion

- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – calibration
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – triangulation



Incremental Structure from Motion

- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – calibration
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – triangulation
- Refine structure and motion: bundle adjustment

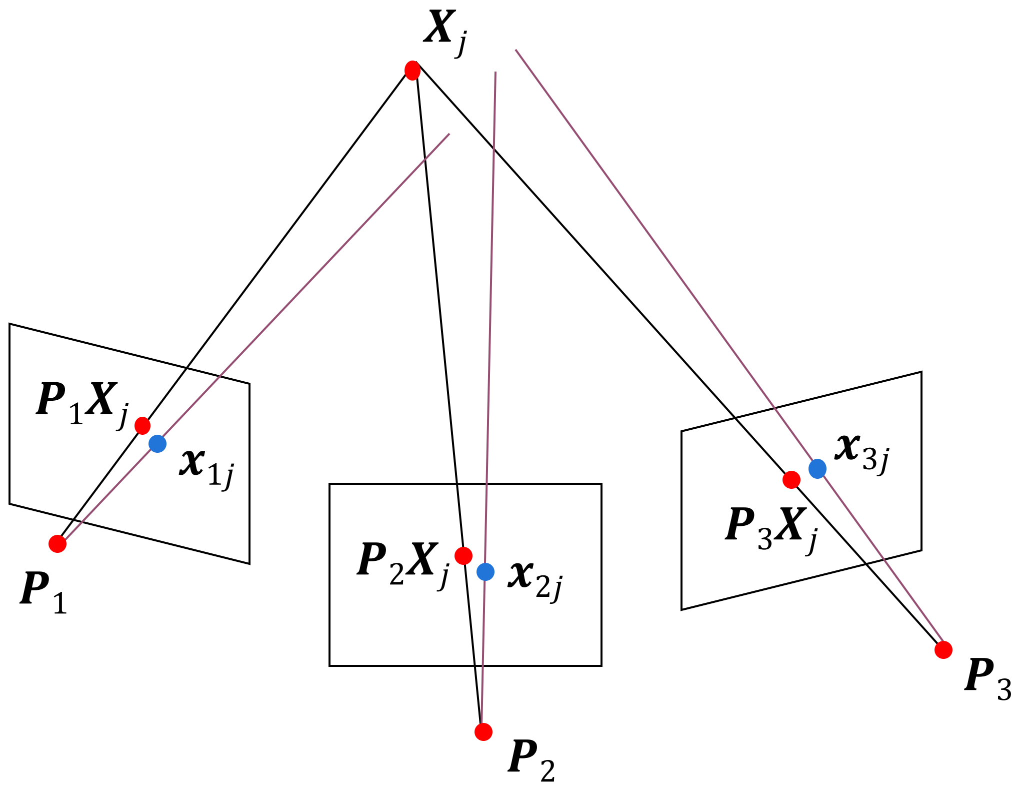


Bundle Adjustment

- Non-linear method for refining structure and motion
- Minimize reprojection error (with lots of bells and whistles):

$$\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}\, d\left(\boldsymbol{x}_{ij} - \text{proj}\left(\boldsymbol{P}_i\boldsymbol{X}_j\right)\right)^2$$

visibility flag: is point j visible in view i?



B. Triggs et al. Bundle adjustment – A modern synthesis. International Workshop on Vision Algorithms, 1999

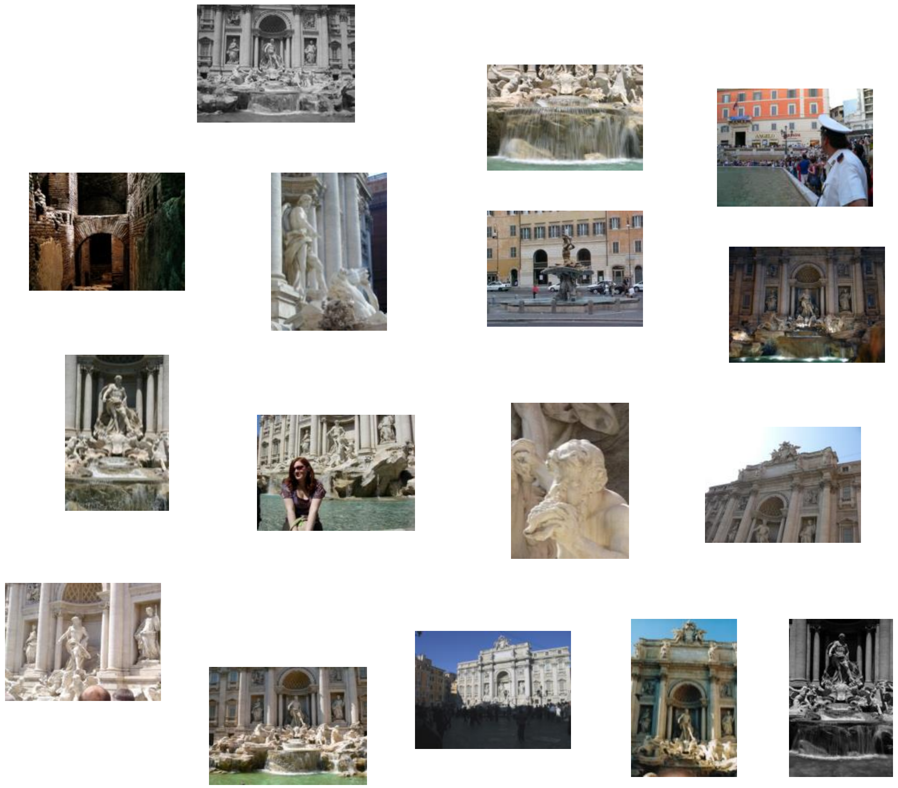
Outline

- Problem Formulation
- Projective structure from motion
- SfM pipeline

Representative SFM Pipeline

N. Snavely, S. Seitz, and R. Szeliski. [Photo tourism: Exploring photo collections in 3D](). SIGGRAPH 2006
http://phototour.cs.washington.edu/

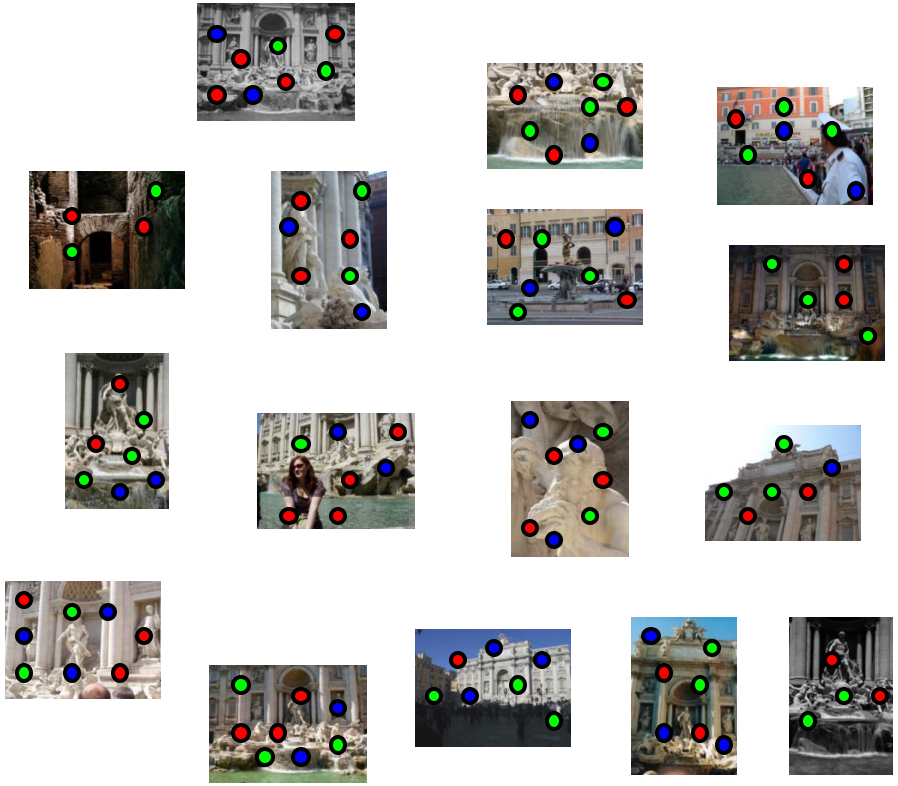
Feature Detection



Detect SIFT features

Source: N. Snavely

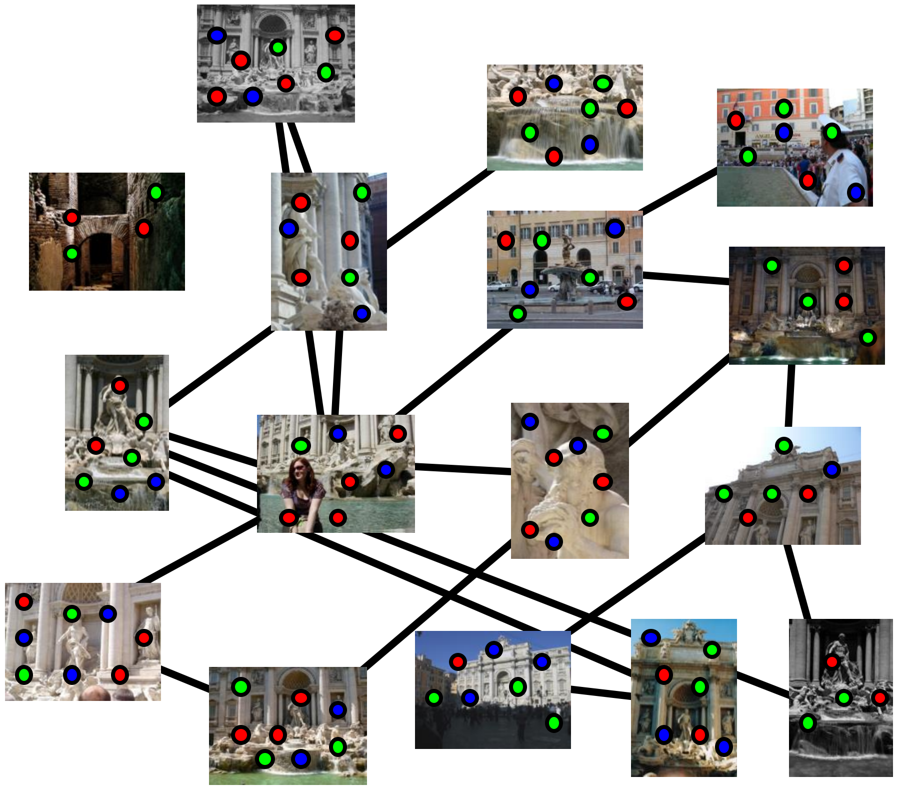
Feature Detection



Other popular feature types: SURF, ORB, BRISK, …

Source: N. Snavely

Feature Matching



Match features between each pair of images

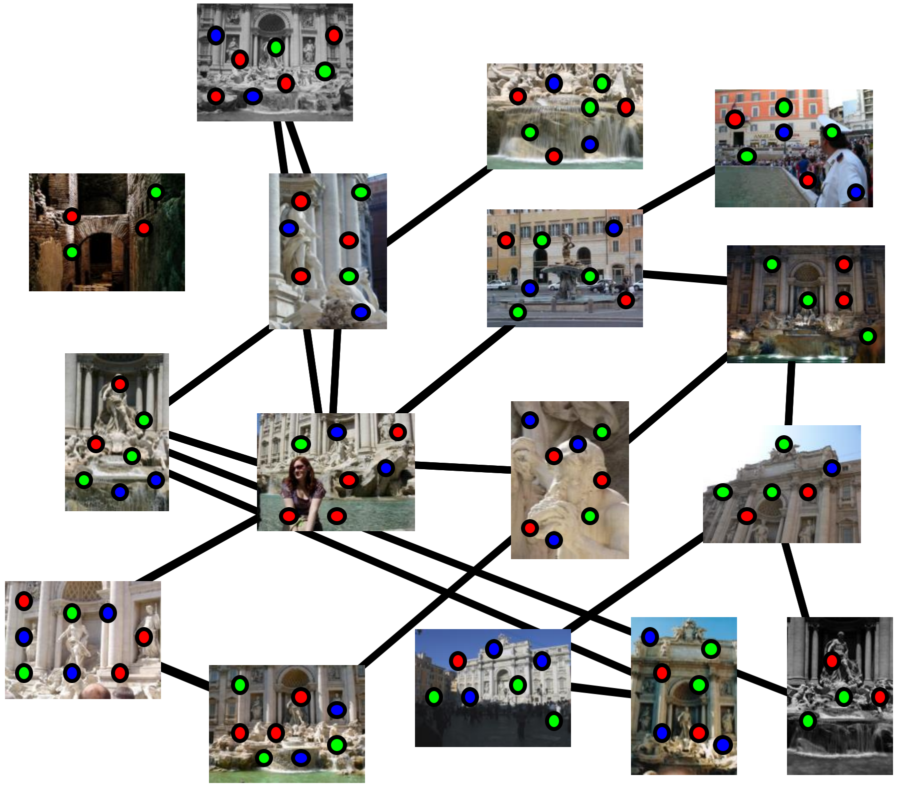
Source: N. Snavely

Feature Matching

Use RANSAC to estimate fundamental matrix between each pair

Image source

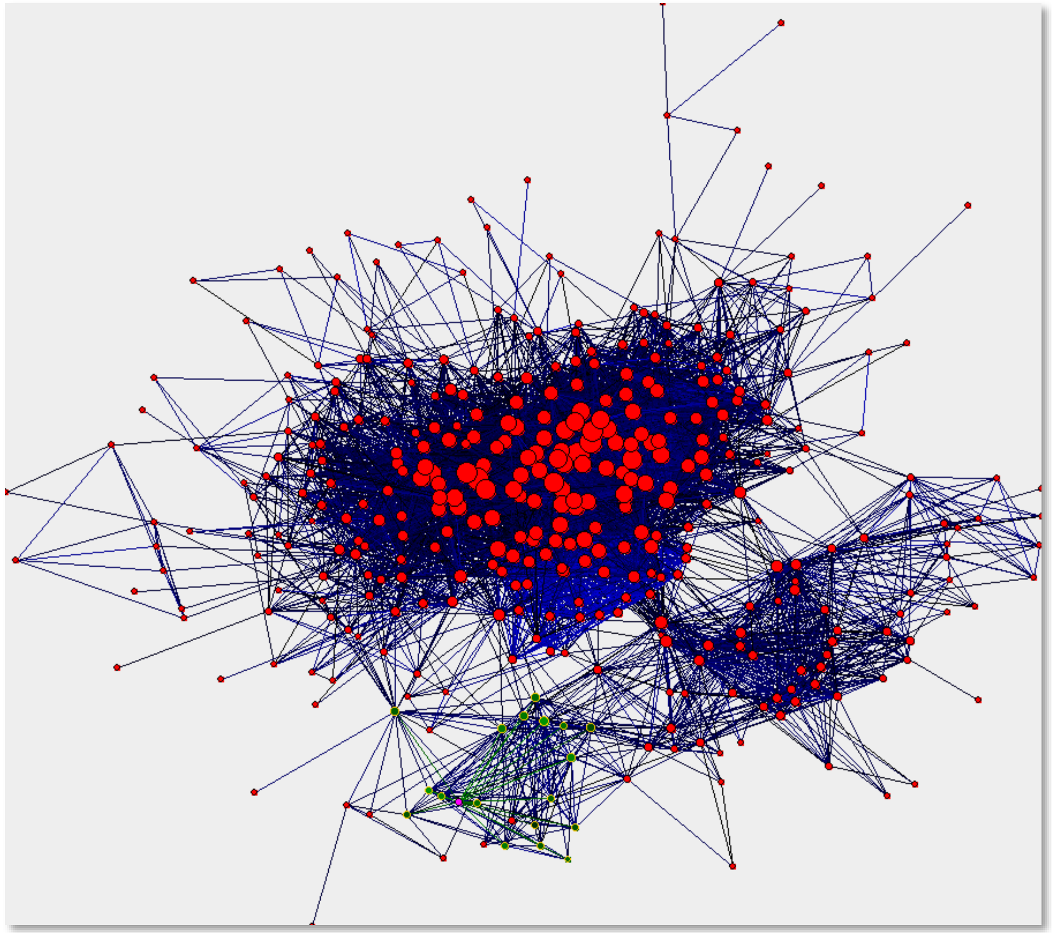
Feature Matching



Use RANSAC to estimate fundamental matrix between each pair

Source: N. Snavely

Image Connectivity Graph



(graph layout produced using the Graphviz toolkit: http://www.graphviz.org/)

Source: N. Snavely

Structure from Motion (SfM) Pipeline

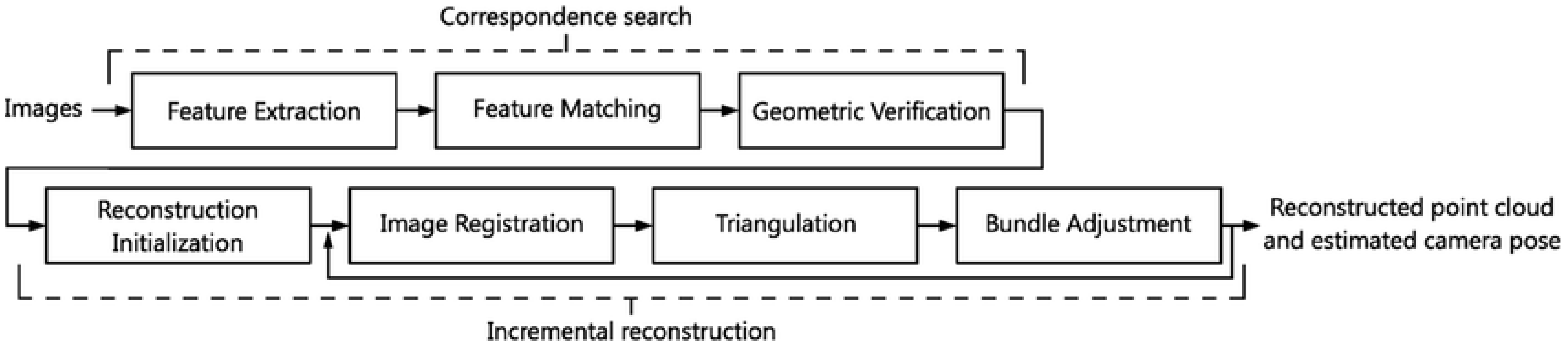


Figure from [paper]

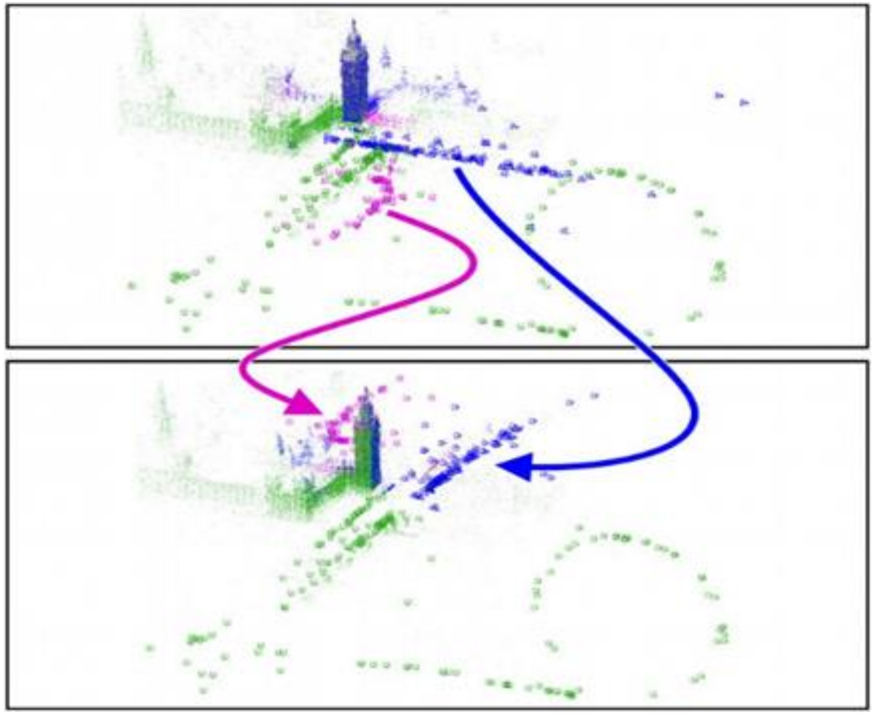
Incremental SFM

- Pick a pair of images with lots of inliers (and preferably, good EXIF data)
  - Initialize intrinsic parameters (focal length, principal point) from EXIF
  - Estimate extrinsic parameters ($\boldsymbol{R}$ and $\boldsymbol{t}$) using five-point algorithm
  - Use triangulation to initialize model points
- While remaining images exist
  - Find an image with many feature matches with images in the model
  - Run RANSAC on feature matches to register new image to model
  - Triangulate new points
  - Perform bundle adjustment to re-optimize everything
  - Optionally, align with GPS from EXIF data or ground control points

The devil is in the details

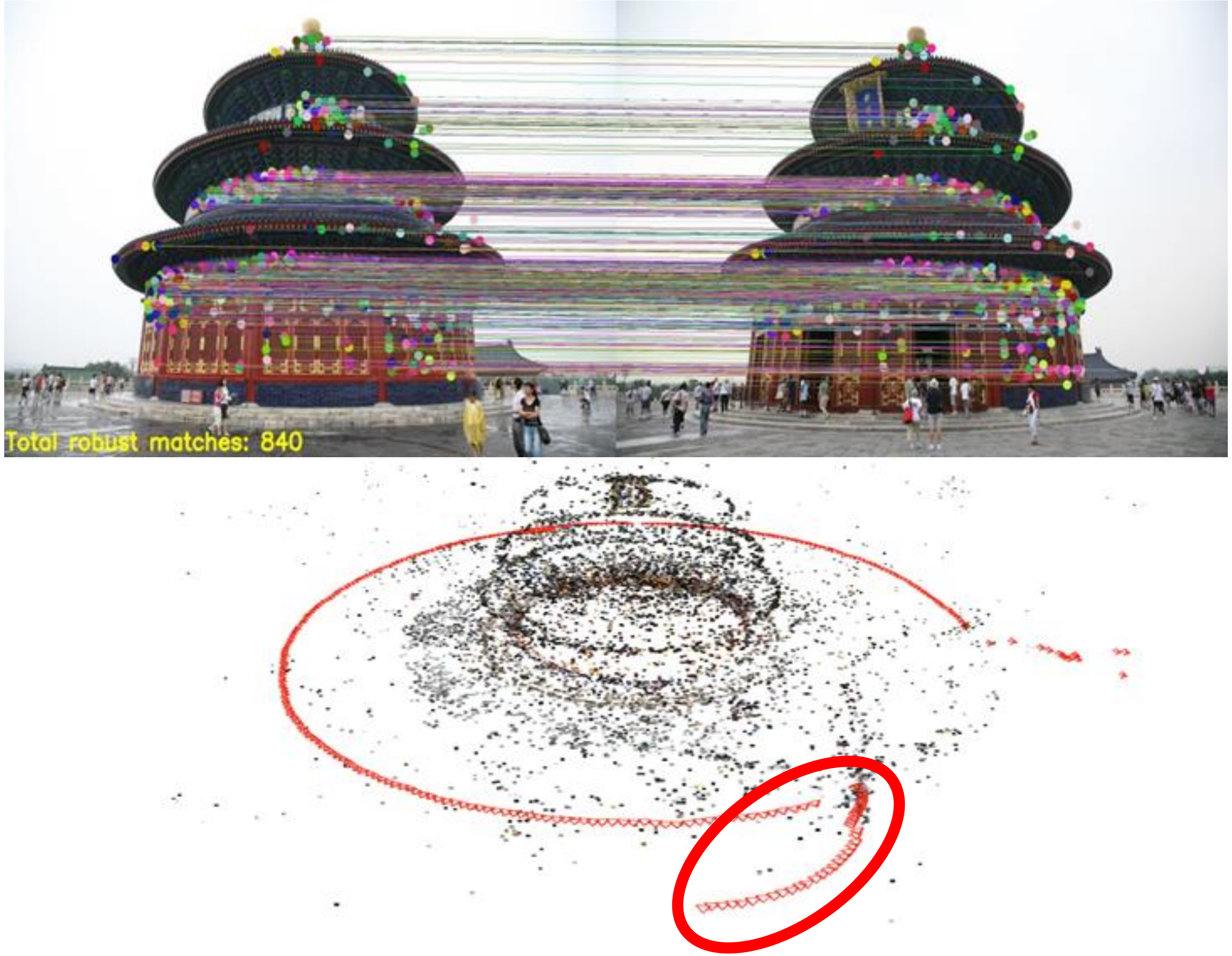
- Handling degenerate configurations (e.g., homographies)
- Filtering out incorrect matches
- Dealing with repetitions and symmetries

Repetitive structures cause catastrophic failures



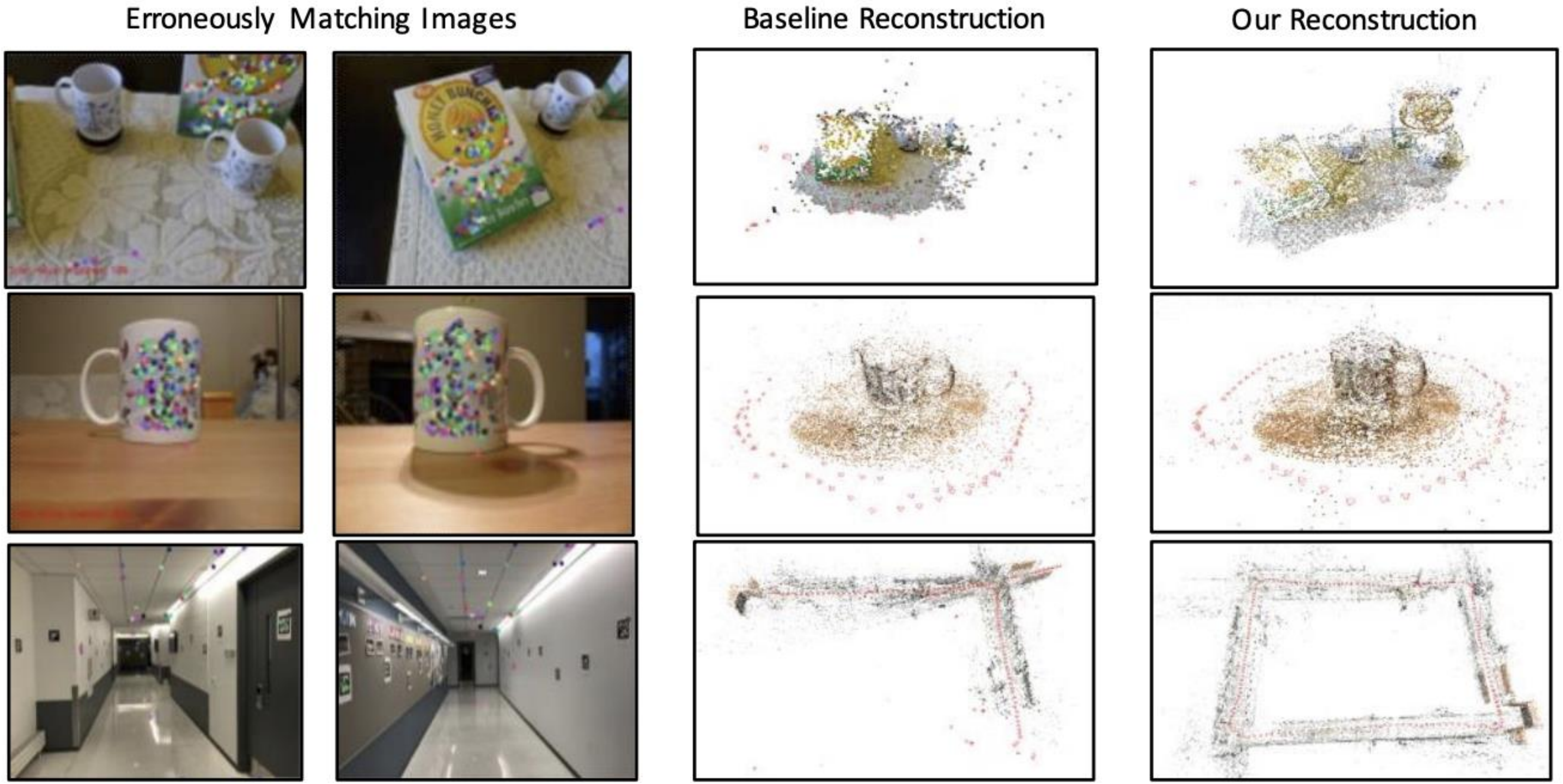
https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

Repetitive structures cause catastrophic failures



R. Kataria et al. [Improving Structure from Motion with Reliable Resectioning](). 3DV 2020

Repetitive structures cause catastrophic failures



R. Kataria et al. Improving Structure from Motion with Reliable Resectioning. 3DV 2020

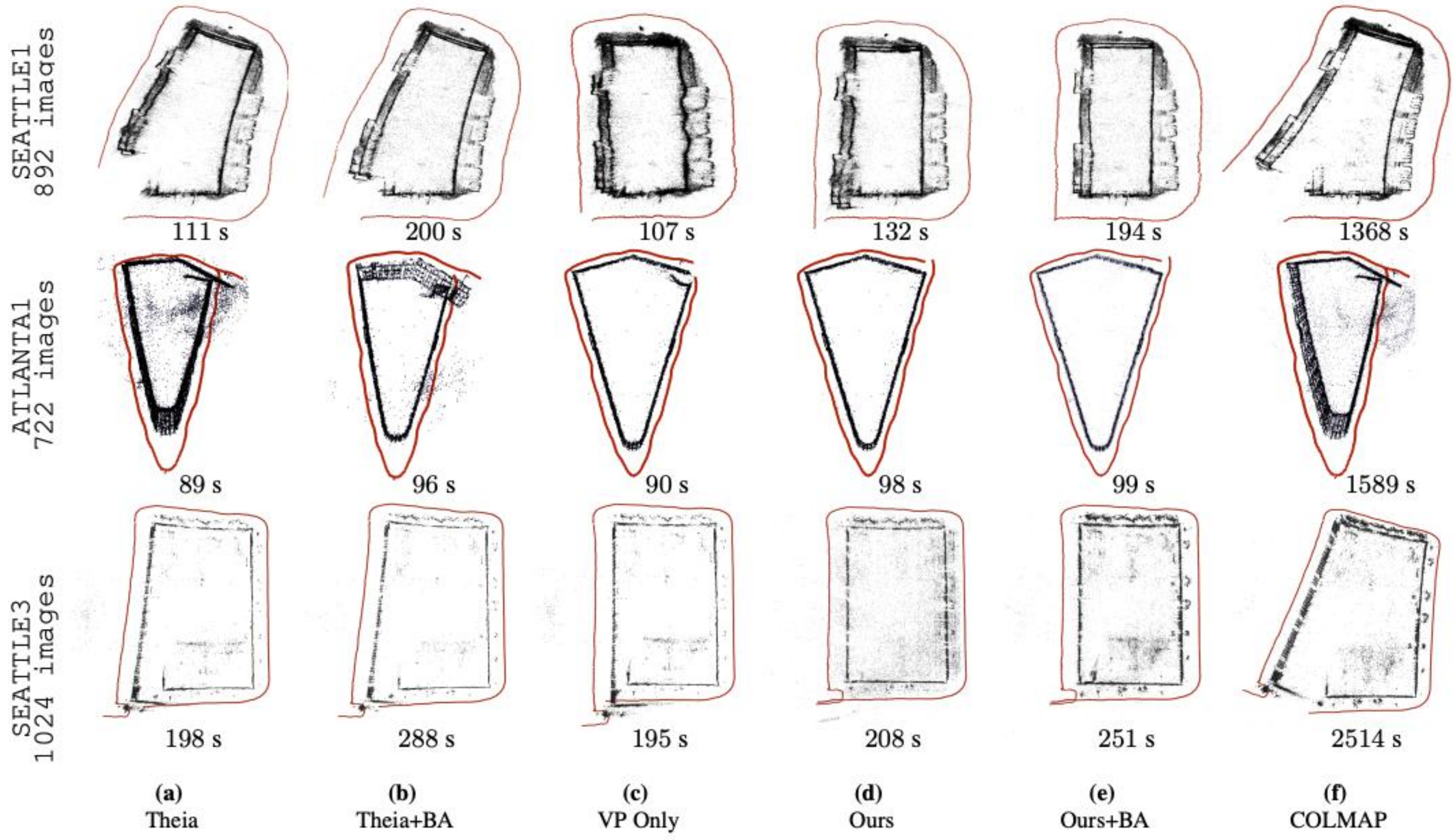
The devil is in the details

- Handling degenerate configurations (e.g., homographies)
- Filtering out incorrect matches
- Dealing with repetitions and symmetries
- Reducing error accumulation and closing loops

Reducing error accumulation and closing loops

A. Holynski et al. [Reducing Drift in Structure From Motion Using Extended Features](). arXiv 2020

Reducing error accumulation and closing loops



A. Holynski et al. [Reducing Drift in Structure From Motion Using Extended Features](). arXiv 2020

The devil is in the details

- Handling degenerate configurations (e.g., homographies)
- Filtering out incorrect matches
- Dealing with repetitions and symmetries
- Reducing error accumulation and closing loops
- Making the whole thing efficient!
  - See, e.g., Towards Linear-Time Incremental Structure from Motion

SfM Software

- COLMAP
- Bundler
- OpenSfM
- OpenMVG
- VisualSFM
- See also Wikipedia’s list of toolboxes