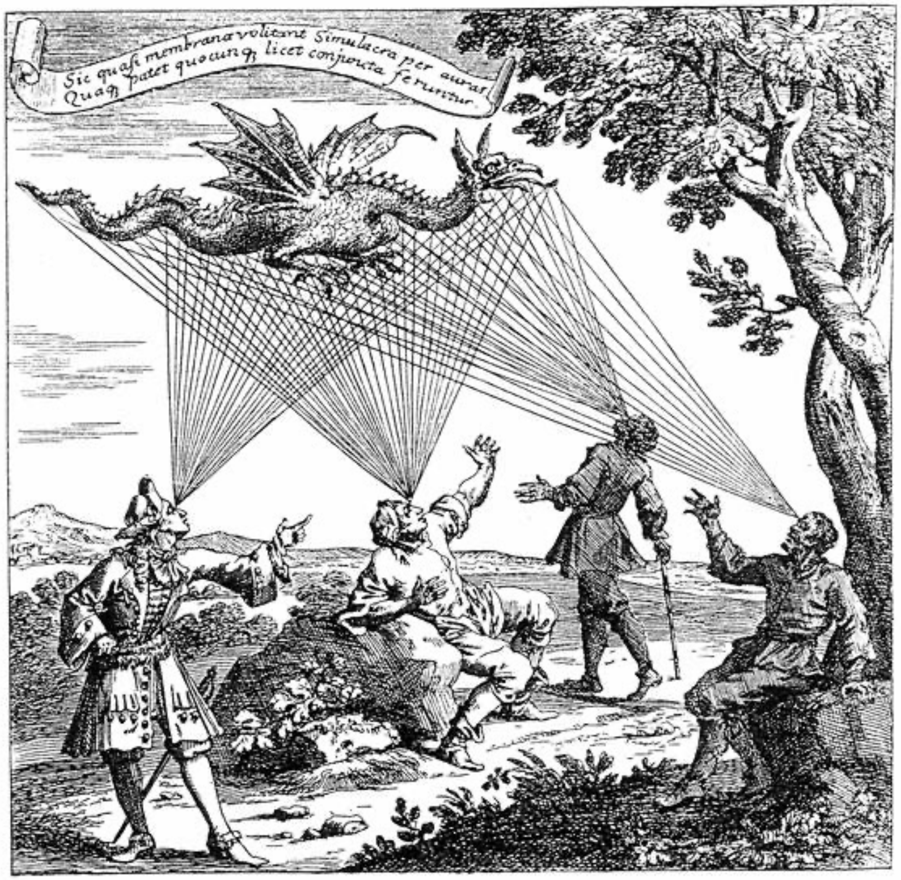
Structure from motion


Sic quaſi membrana volitant Simulacra per auras
Quaq; patet quocunq; licet conjuncta feruntur

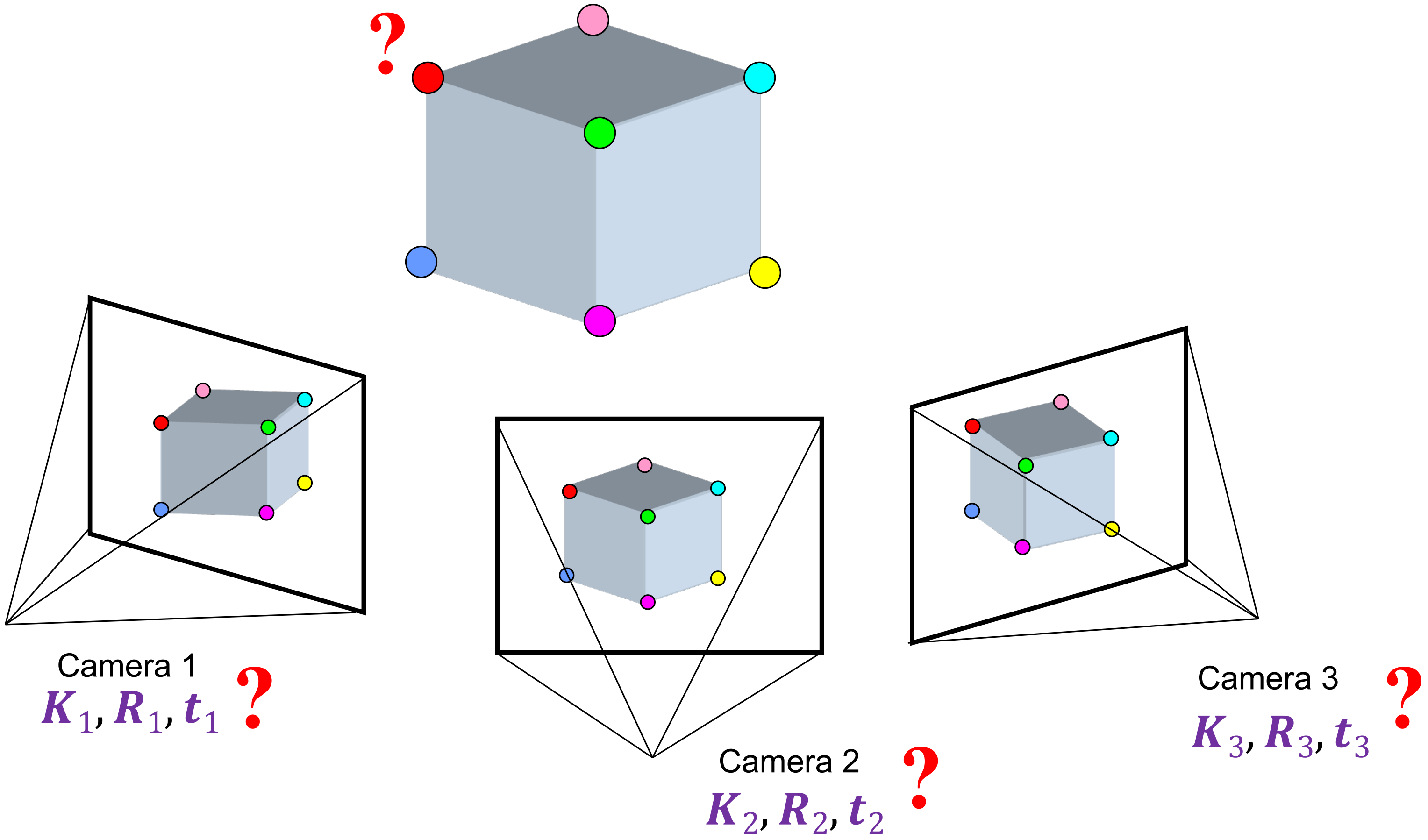
Драконъ, видимый подъ различными углами зрѣнія

По гравюрѣ на мѣди изъ „Oculus artificialis teledioptricus" Цана. 1702 года.

Outline: Structure from motion

- Problem definition and ambiguities
- Affine structure from motion
  - Factorization
- Projective structure from motion
  - Bundle adjustment
- Modern structure from motion pipeline

Structure from motion



?

Camera 1

$\boldsymbol{K}_1, \boldsymbol{R}_1, \boldsymbol{t}_1$?

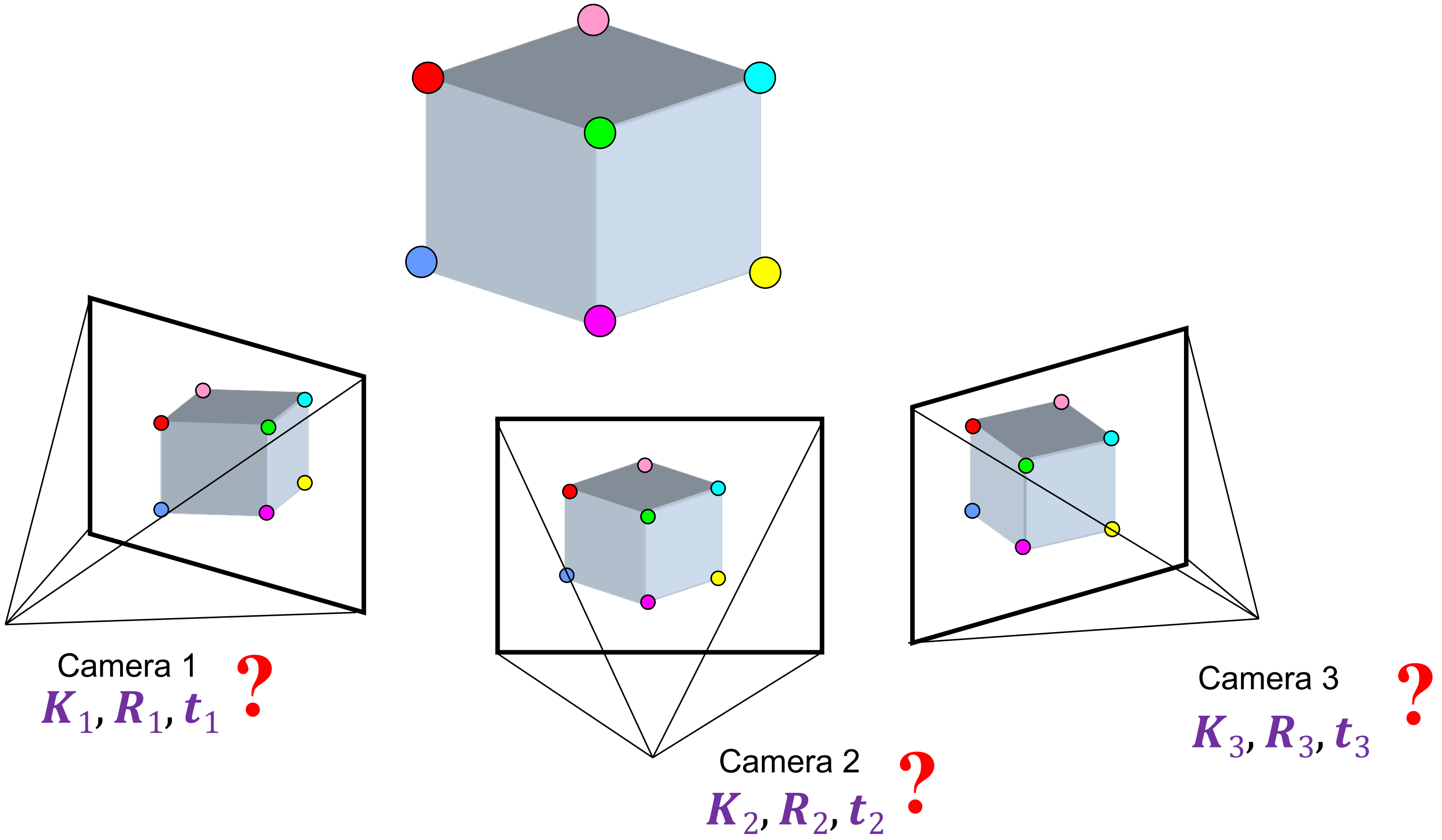
Camera 2

$\boldsymbol{K}_2, \boldsymbol{R}_2, \boldsymbol{t}_2$?

Camera 3

$\boldsymbol{K}_3, \boldsymbol{R}_3, \boldsymbol{t}_3$?

Recall: Calibration



Camera 1

$\boldsymbol{K}_1, \boldsymbol{R}_1, \boldsymbol{t}_1$?

Camera 2

$\boldsymbol{K}_2, \boldsymbol{R}_2, \boldsymbol{t}_2$?

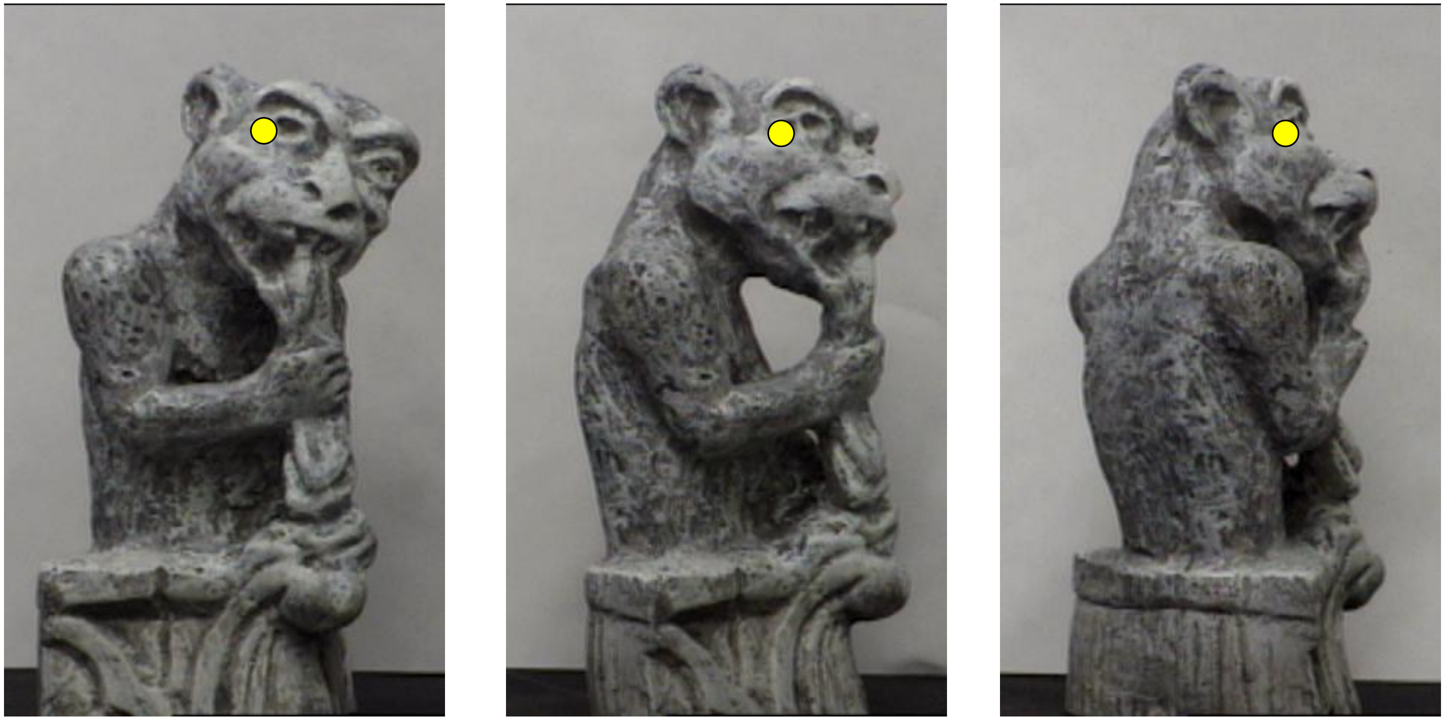
Camera 3

$\boldsymbol{K}_3, \boldsymbol{R}_3, \boldsymbol{t}_3$?

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

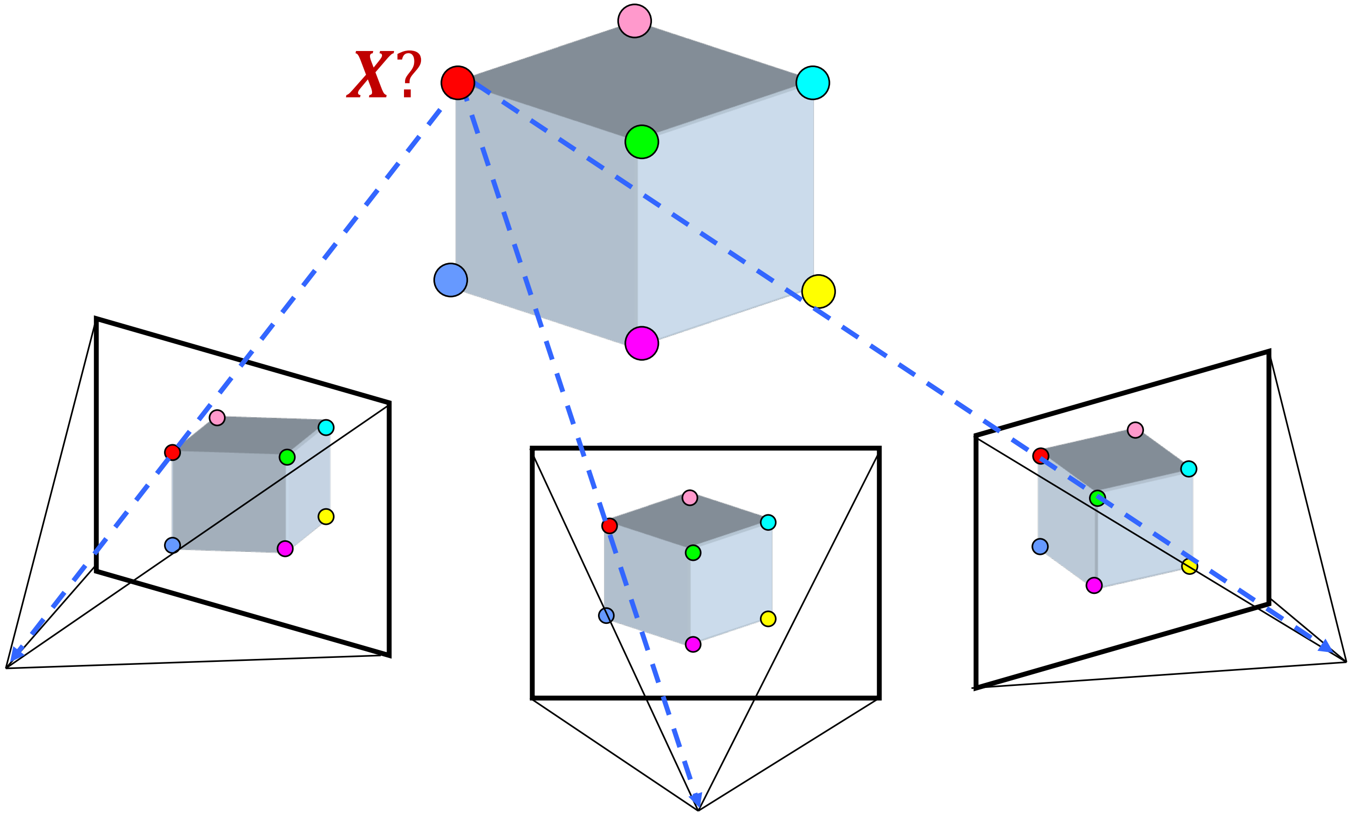
Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



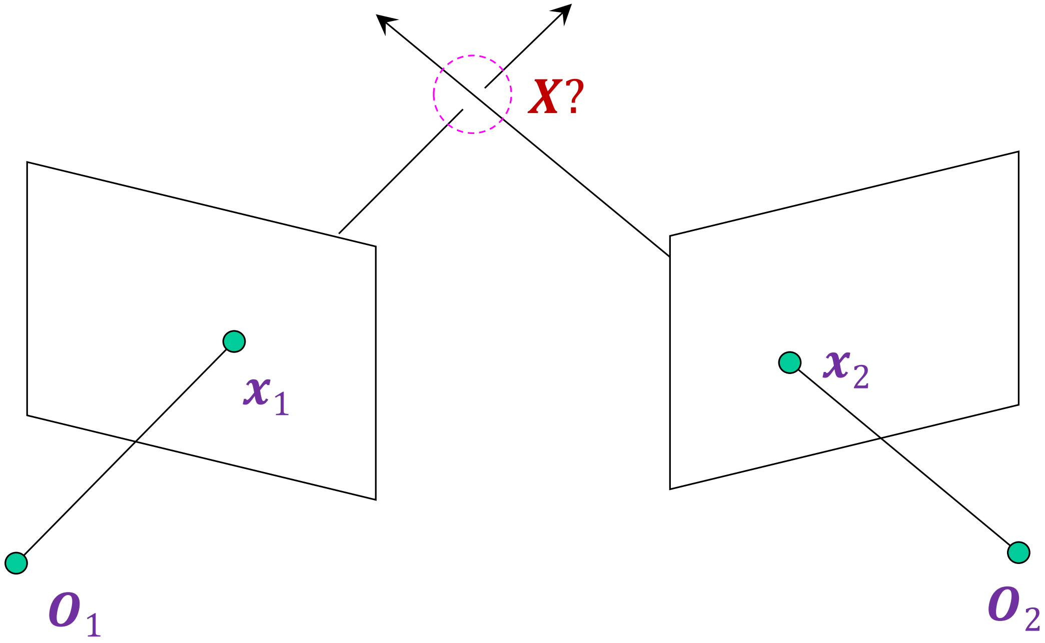
Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



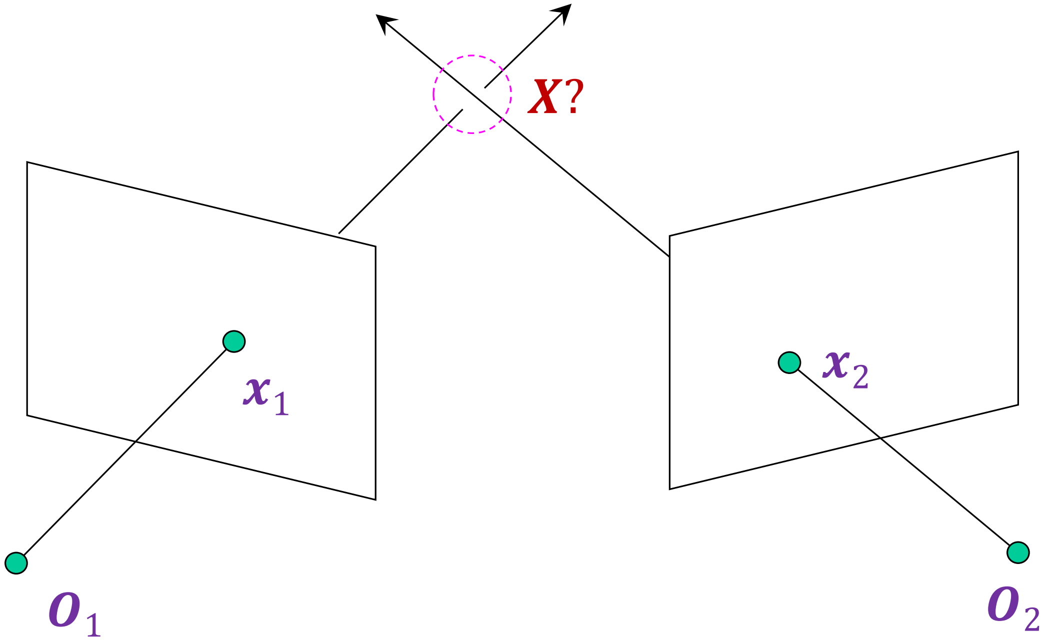
Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point

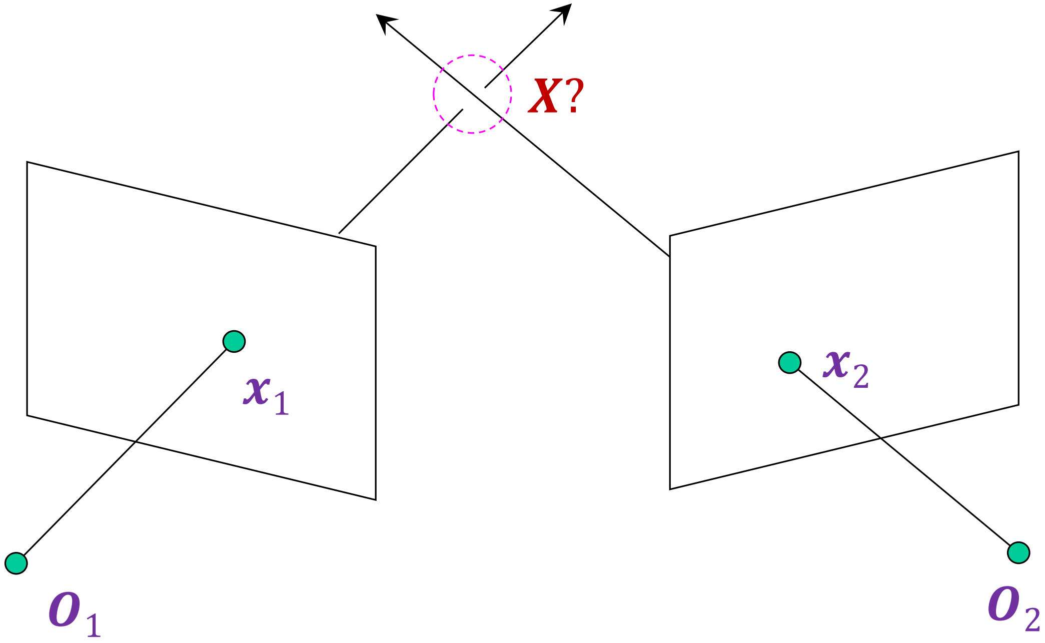


Triangulation

- We want to intersect the two visual rays corresponding to $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, but because of noise and numerical errors, they don't meet exactly



Ignored camera intrinsics cause cameras are known!



$$\mathbf{x}_1 \equiv [\mathcal{I}|\mathbf{0}]\,\mathbf{X}$$

$$\mathbf{x}_2 \equiv [\mathcal{R}|\mathbf{t}]\,\mathbf{X}$$

$$\mathbf{x}_1 \equiv [\mathcal{I}|\mathbf{0}]\,\mathbf{X} \qquad \mathbf{x}_2 \equiv [\mathcal{R}|\mathbf{t}]\,\mathbf{X}$$

$$X_1 = x_1 X_3$$

$\uparrow$

Affine coordinates of image point

$\downarrow$

$$X_2 = y_1 X_3$$

Remember camera is known,
And substitute

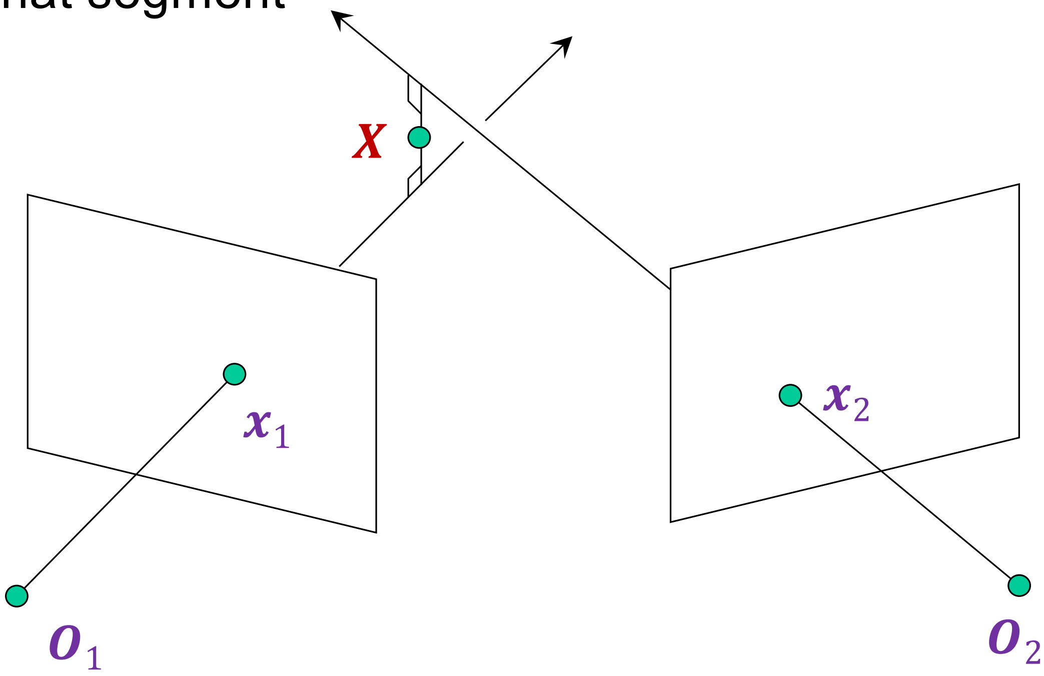
$$\mathbf{x}_1 \equiv [\mathcal{I}|\mathbf{0}]\,\mathbf{X} \qquad \mathbf{x}_2 \equiv [\mathcal{R}|\mathbf{t}]\,\mathbf{X}$$

$$x_2 = \frac{r_{11}x_1X_3 + r_{12}y_1X_3 + r_{13}X_3 + t_1}{r_{31}x_1X_3 + r_{32}y_1X_3 + r_{32}X_3 + t_3}$$

$$y_2 = \frac{r_{21}x_1X_3 + r_{22}y_1X_3 + r_{23}X_3 + t_2}{r_{31}x_1X_3 + r_{32}y_1X_3 + r_{32}X_3 + t_3}$$

Triangulation – Straightforward Approaches

- Above gives two possible points, average them
- Choose least squares X_3
- Find shortest segment connecting the two viewing rays and let ***X*** be the midpoint of that segment

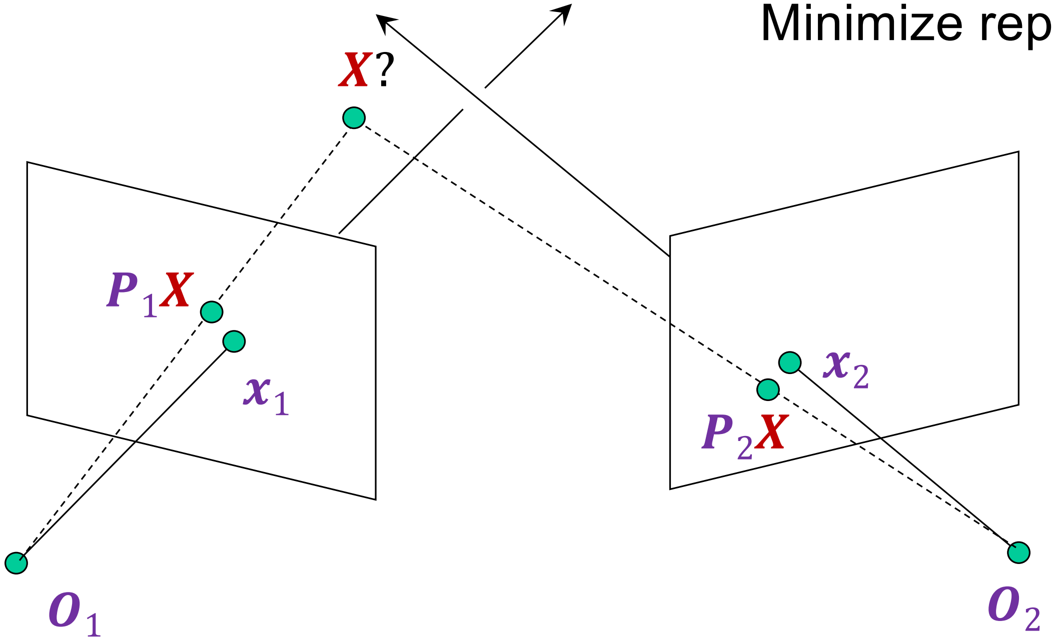


Triangulation: Nonlinear approach

- Find $\boldsymbol{X}$ that minimizes

$$\|\text{proj}(\boldsymbol{P}_1\boldsymbol{X}) - \boldsymbol{x}_1\|_2^2 + \|\text{proj}(\boldsymbol{P}_2\boldsymbol{X}) - \boldsymbol{x}_2\|_2^2$$

Minimize reprojection error

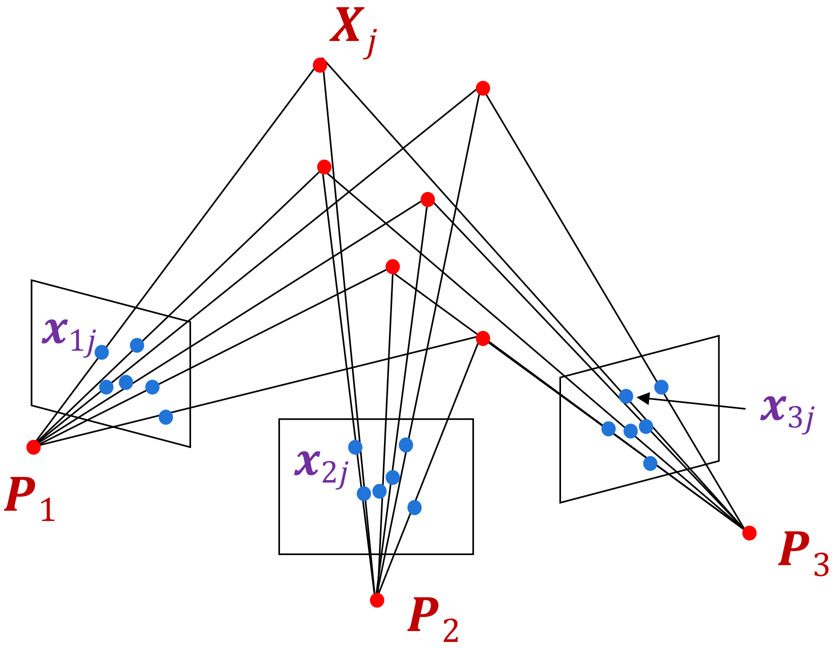


Structure from motion: Problem formulation

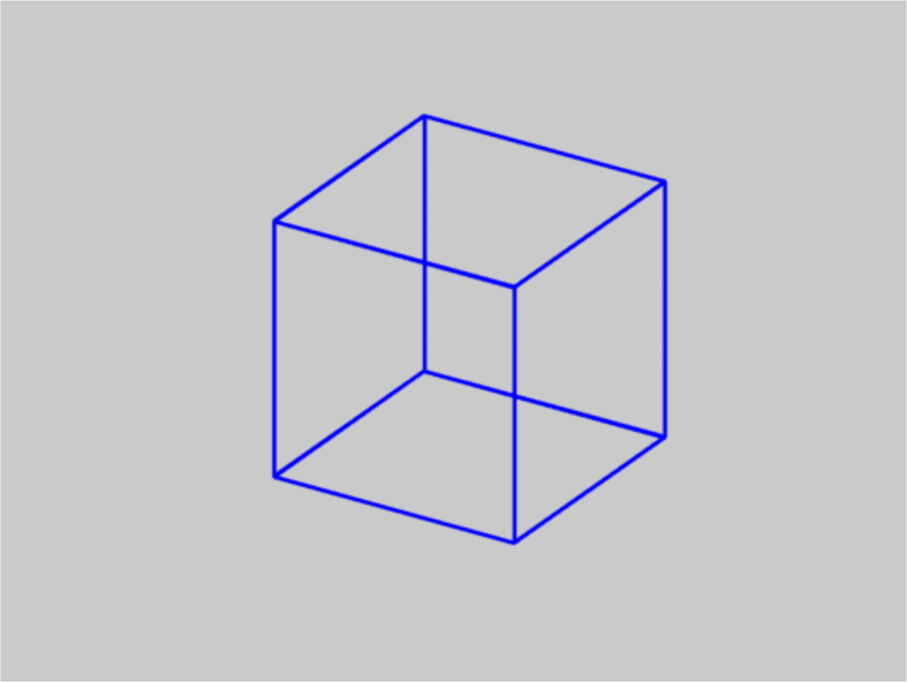
- Given: m images of n fixed 3D points such that (ignoring visibility)

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

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



Is SFM always uniquely solvable?

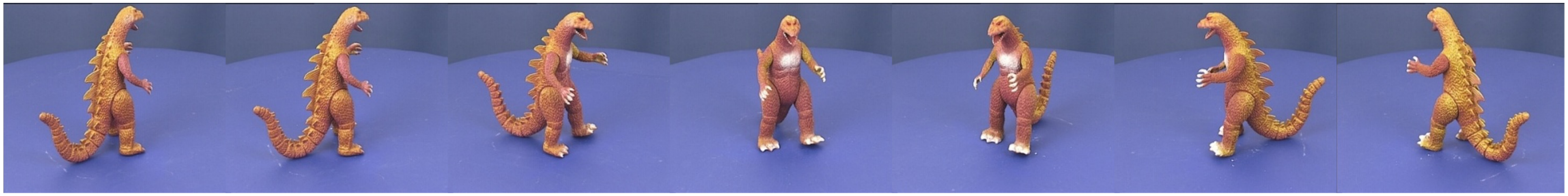


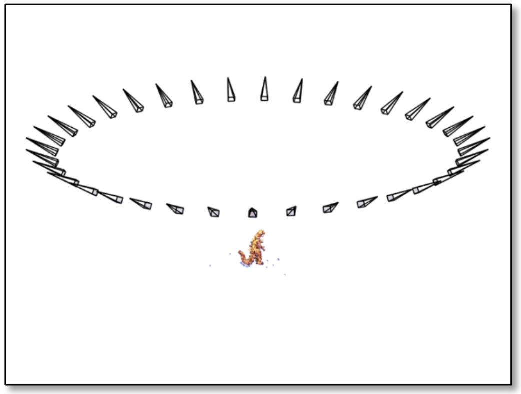
Necker cube

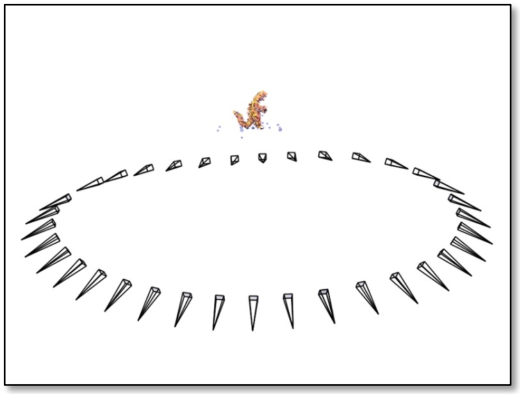
Source: N. Snavely

Is SFM always uniquely solvable?

- Could actually happen in affine structure from motion:







Source: N. Snavely

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:

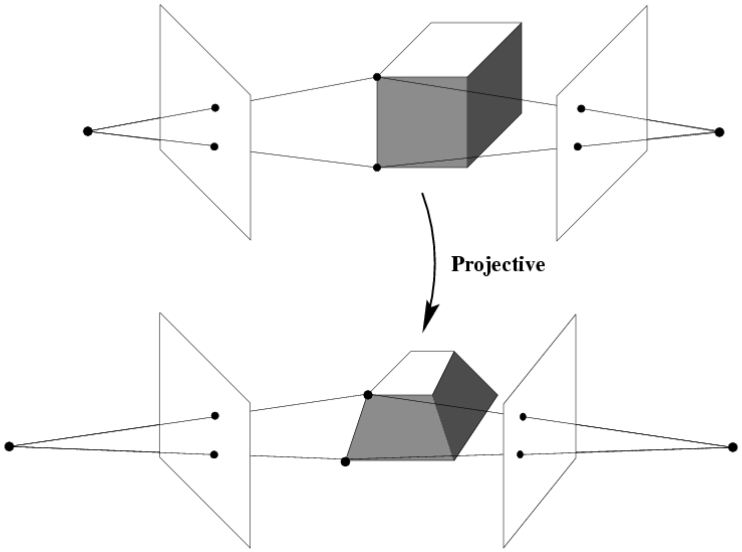
$$\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = (\boldsymbol{P}\boldsymbol{Q}^{-1})(\boldsymbol{Q}\boldsymbol{X})$$

Projective ambiguity

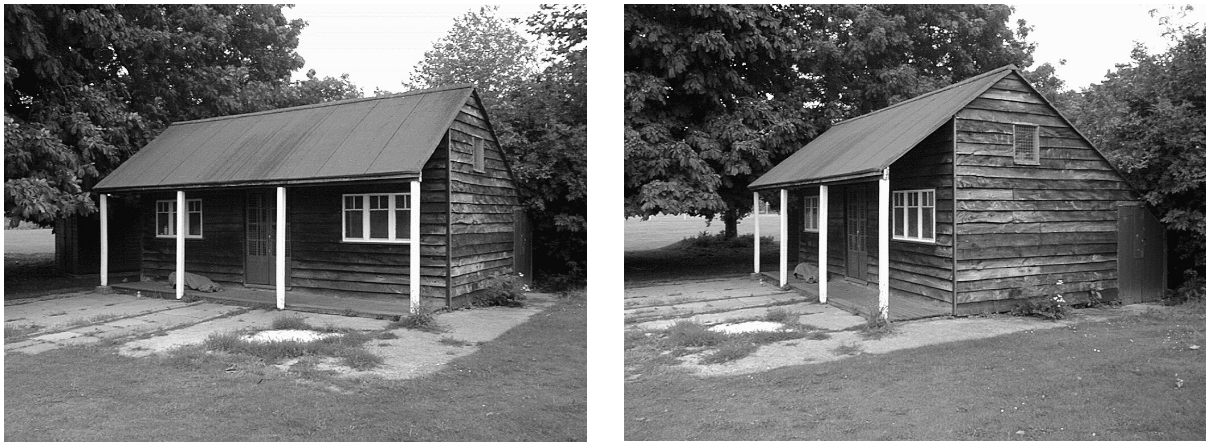
- With no constraints on the camera calibration matrices or on the scene, we can reconstruct up to a *projective* ambiguity:

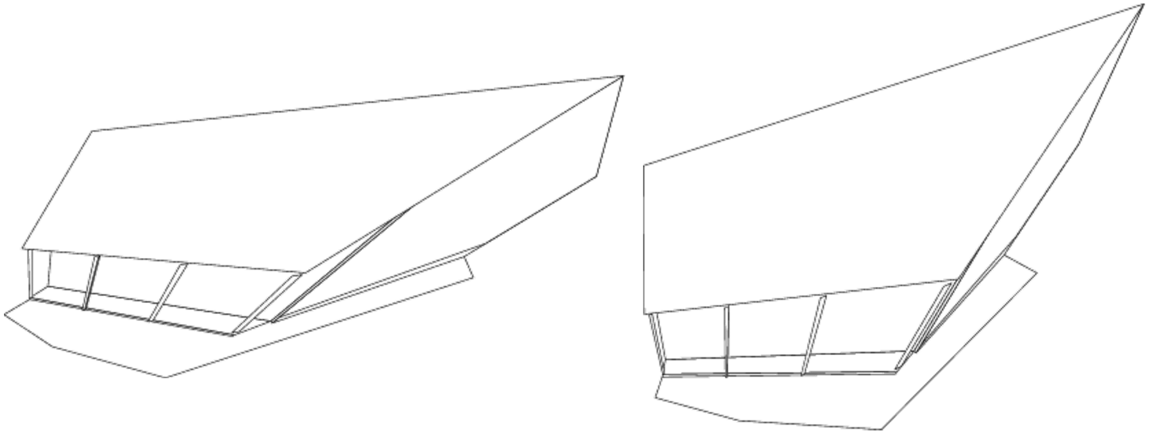
$$\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = (\boldsymbol{P}\boldsymbol{Q}^{-1})(\boldsymbol{Q}\boldsymbol{X})$$

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



Projective ambiguity



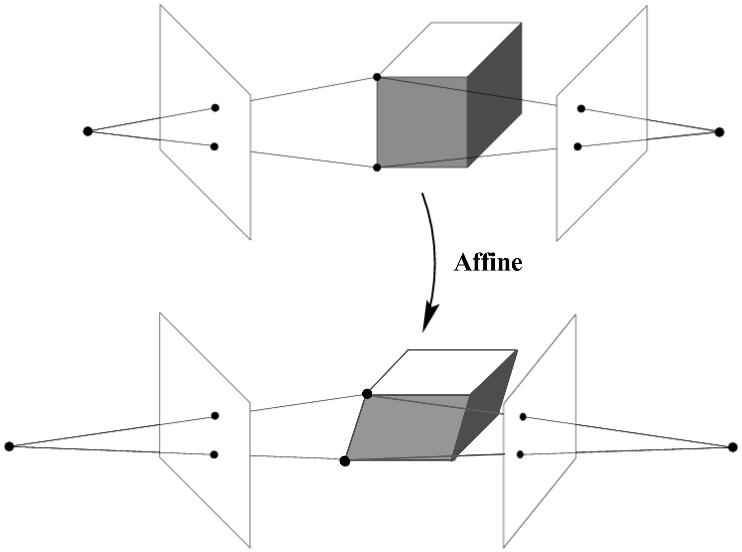


Affine ambiguity

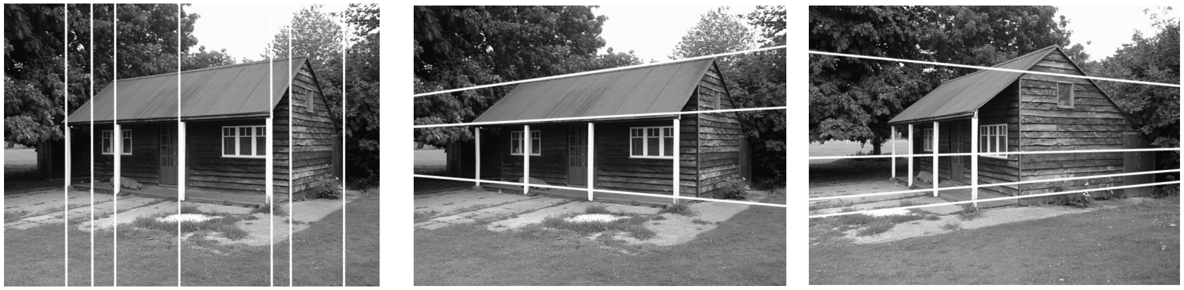
- If we impose parallelism constraints, we can get a reconstruction up to an *affine* ambiguity:

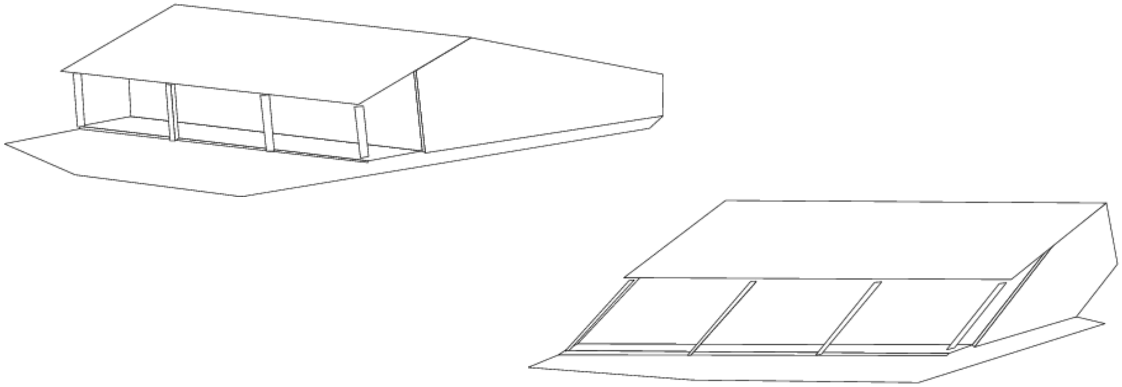
$$\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X} = \left(\boldsymbol{P}\boldsymbol{Q}_A^{-1}\right)\left(\boldsymbol{Q}_A\boldsymbol{X}\right)$$

$$\boldsymbol{Q}_A = \begin{bmatrix} \underbrace{\boldsymbol{A}}_{\substack{3\times3\\\text{full-rank}\\\text{matrix}}} & \underbrace{\boldsymbol{t}}_{\substack{3\times1\\\text{translation}\\\text{vector}}} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$



Affine ambiguity





Similarity ambiguity

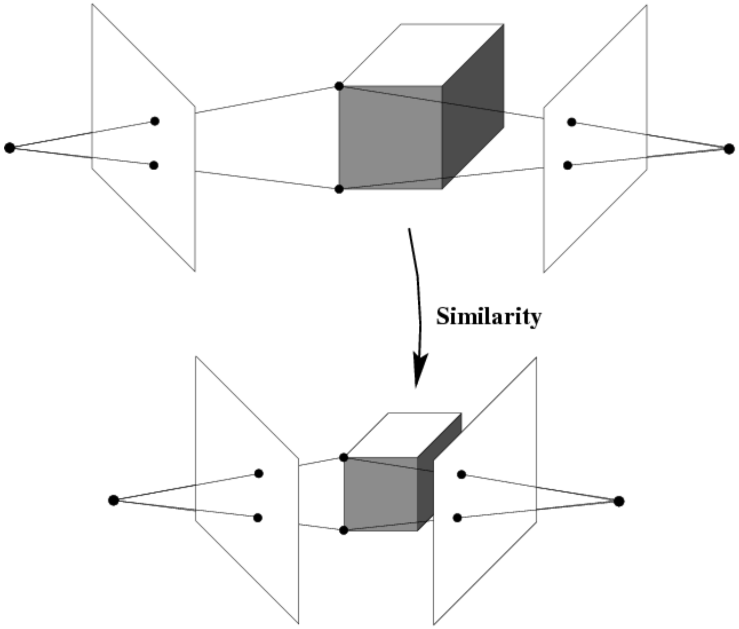
- A reconstruction that obeys orthogonality constraints on camera parameters and/or scene

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

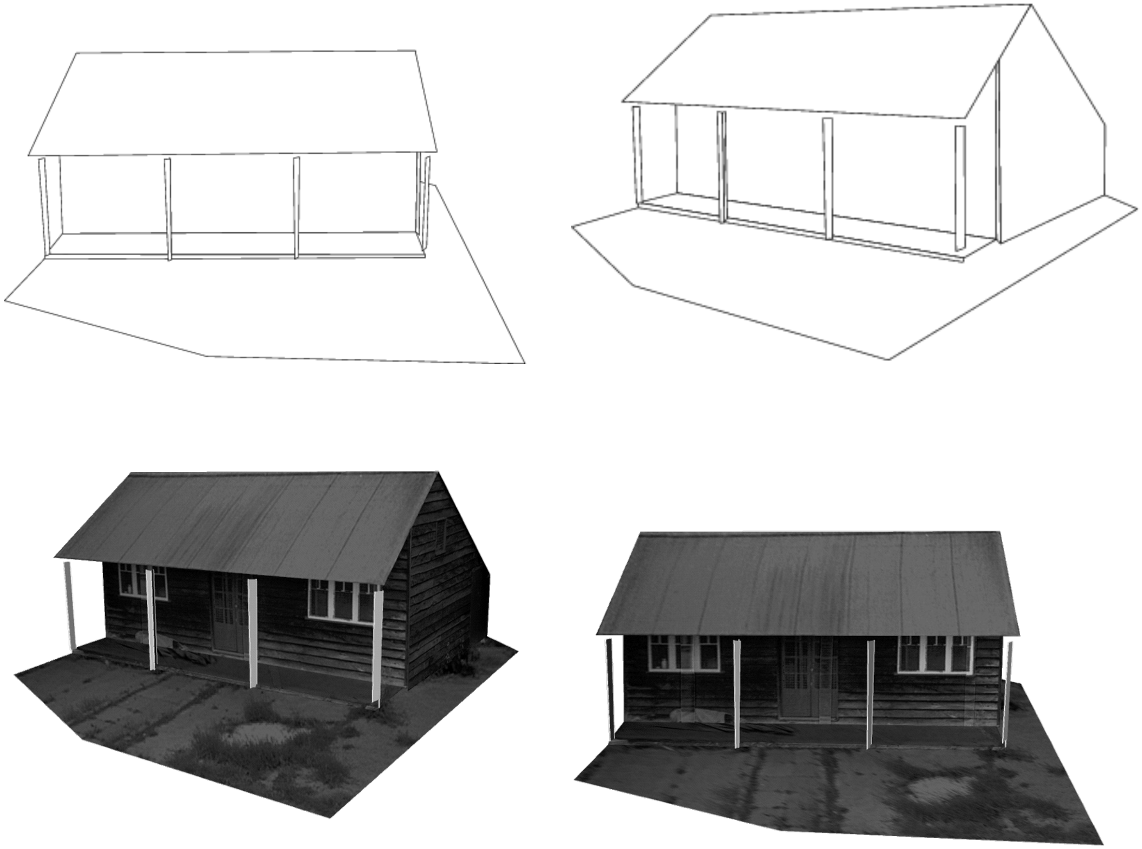
3×3 rotation matrix

3×1 translation vector

$$\boldsymbol{Q}_S = \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$



Similarity ambiguity

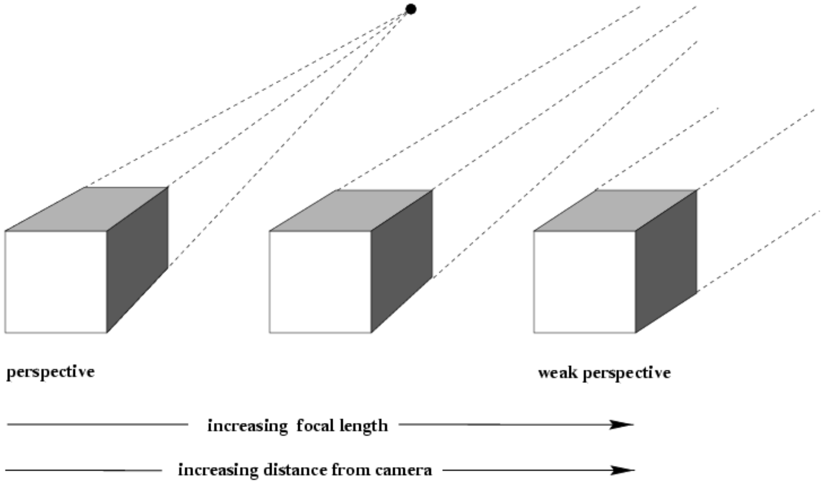


Outline: Structure from motion

- Problem definition and ambiguities
- Affine structure from motion
  - Factorization

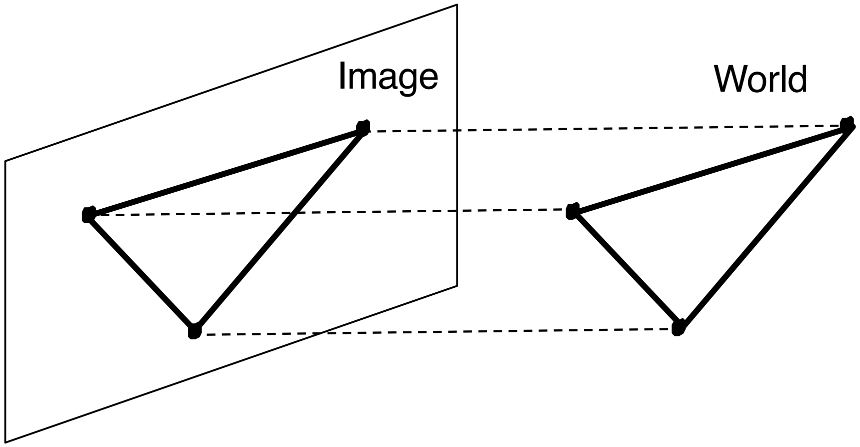
Affine structure from motion

- Let’s start with *affine* or *weak perspective* cameras



center at infinity

Recall: Orthographic projection



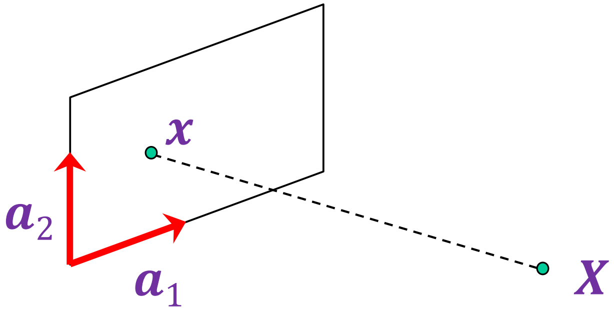
Just drop the z coordinate!

$$\begin{pmatrix}x\\y\\1\end{pmatrix} = \left[\begin{matrix}\qquad\qquad\qquad\qquad\end{matrix}\right]\begin{pmatrix}x\\y\\z\\1\end{pmatrix}$$

General affine projection

- A general affine projection is a 3D-to-2D linear mapping plus translation:

$$\boldsymbol{P} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$



$\boldsymbol{a}_1, \boldsymbol{a}_2$: rows of projection matrix

- In non-homogeneous coordinates:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \boldsymbol{AX} + \boldsymbol{t}$$

Projection of world origin

Affine structure from motion

- **Given**: m images of n fixed 3D points such that

$$\boldsymbol{x}_{ij} = \boldsymbol{A}_i \boldsymbol{X}_j + \boldsymbol{t}_i, \quad i = 1, \dots, m, \; j = 1, \dots, n$$

- **Problem**: use the mn correspondences $\boldsymbol{x}_{ij}$ to estimate m projection matrices $\boldsymbol{A}_i$ and translation vectors $\boldsymbol{t}_i$, and n points $\boldsymbol{X}_j$
- The reconstruction is defined up to an arbitrary *affine* transformation $\boldsymbol{Q}$ (12 degrees of freedom):

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ 0^T & 1 \end{bmatrix} \rightarrow \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ 0^T & 1 \end{bmatrix} \boldsymbol{Q}^{-1}, \qquad \begin{pmatrix} \boldsymbol{X}_j \\ 1 \end{pmatrix} \rightarrow \boldsymbol{Q} \begin{pmatrix} \boldsymbol{X}_j \\ 1 \end{pmatrix}$$

- How many knowns and unknowns for m images and n points?
  - $2mn$ knowns and $8m + 3n$ unknowns
  - To be able to solve this problem, we must have $2mn \geq 8m + 3n - 12$ (affine ambiguity takes away 12 dof)
  - E.g., for two views, we need four point correspondences

Affine structure from motion

- First, center the data by subtracting the centroid of the image points in each view:

$$
\begin{aligned}
\hat{\boldsymbol{x}}_{ij} &= \boldsymbol{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{x}_{ik} \\
&= \boldsymbol{A}_i \boldsymbol{X}_j + \boldsymbol{t}_i - \frac{1}{n}\sum_{k=1}^{n} (\boldsymbol{A}_i \boldsymbol{X}_k + \boldsymbol{t}_i) \\
&= \boldsymbol{A}_i \left(\boldsymbol{X}_j - \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{X}_k \right) \\
&= \boldsymbol{A}_i \hat{\boldsymbol{X}}_j
\end{aligned}
$$

Affine structure from motion

- After centering, each normalized 2D point $\hat{\boldsymbol{x}}_{ij}$ is related to the 3D point by

$$\hat{\boldsymbol{x}}_{ij} = \boldsymbol{A}_i \hat{\boldsymbol{X}}_j$$

- We can get rid of the need to center the 3D data (and the translation ambiguity) by defining the origin of the world coordinate system as the centroid of the 3D points

Affine structure from motion

- Let’s create a $2m \times n$ data (measurement) matrix:

$$\boldsymbol{D} = \begin{bmatrix} \hat{\boldsymbol{x}}_{11} & \hat{\boldsymbol{x}}_{12} & \cdots & \hat{\boldsymbol{x}}_{1n} \\ \hat{\boldsymbol{x}}_{21} & \hat{\boldsymbol{x}}_{22} & \cdots & \hat{\boldsymbol{x}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{x}}_{m1} & \hat{\boldsymbol{x}}_{m2} & \cdots & \hat{\boldsymbol{x}}_{mn} \end{bmatrix}$$

cameras $(2m)$

$$\hat{\boldsymbol{x}}_{ij} = \boldsymbol{A}_i \boldsymbol{X}_j$$

points (n)

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Affine structure from motion

- Let’s create a $2m \times n$ data (measurement) matrix:

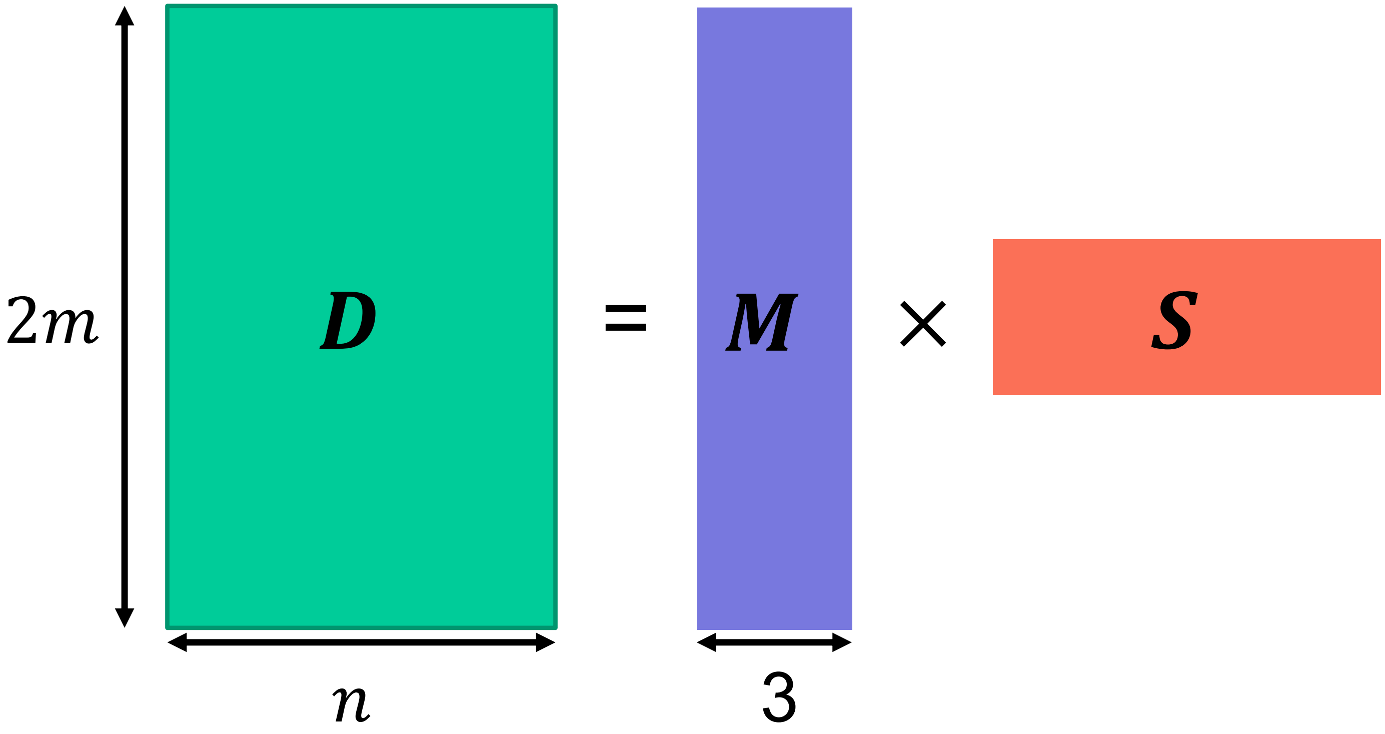
$$\boldsymbol{D} = \begin{bmatrix} \hat{\boldsymbol{x}}_{11} & \hat{\boldsymbol{x}}_{12} & \cdots & \hat{\boldsymbol{x}}_{1n} \\ \hat{\boldsymbol{x}}_{21} & \hat{\boldsymbol{x}}_{22} & \cdots & \hat{\boldsymbol{x}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{x}}_{m1} & \hat{\boldsymbol{x}}_{m2} & \cdots & \hat{\boldsymbol{x}}_{mn} \end{bmatrix} = \underset{\substack{\boldsymbol{M} \\ \text{cameras} \\ (2m \times 3)}}{\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_m \end{bmatrix}} \underset{\substack{\boldsymbol{S} \\ \text{points } (3 \times n)}}{[\boldsymbol{X}_1 \quad \boldsymbol{X}_2 \quad \cdots \quad \boldsymbol{X}_n]}$$

- What must be the rank of the measurement matrix $\boldsymbol{D} = \boldsymbol{M}\boldsymbol{S}$?

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

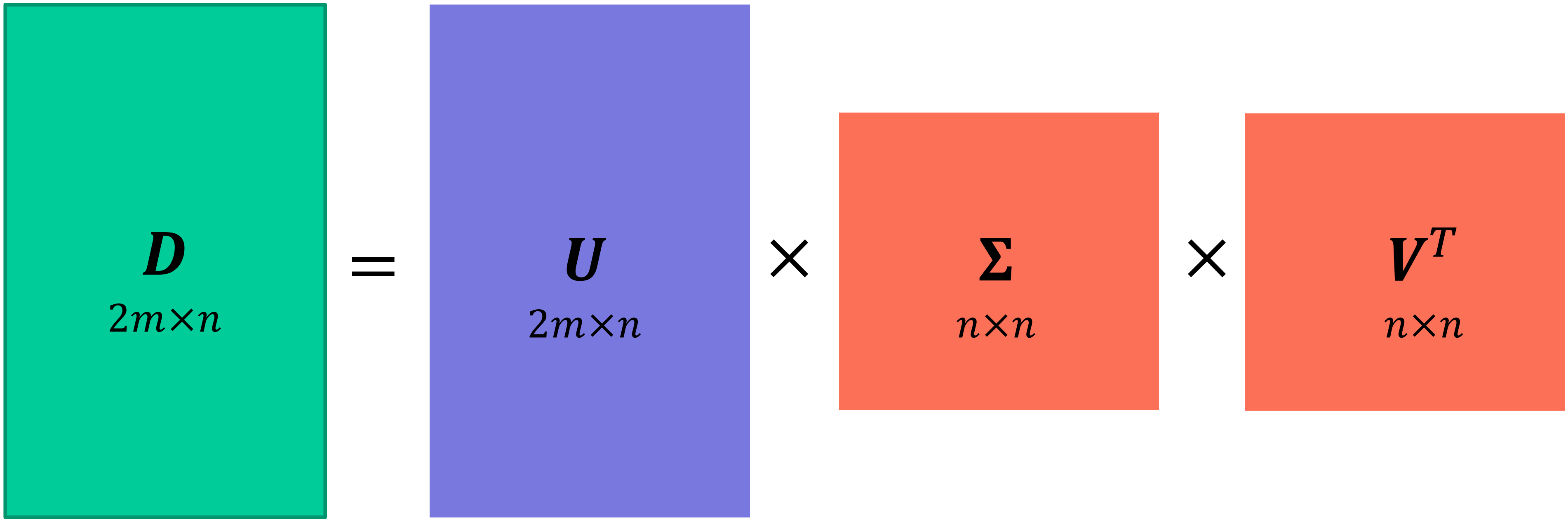
Factorizing the measurement matrix

- We want:



Factorizing the measurement matrix

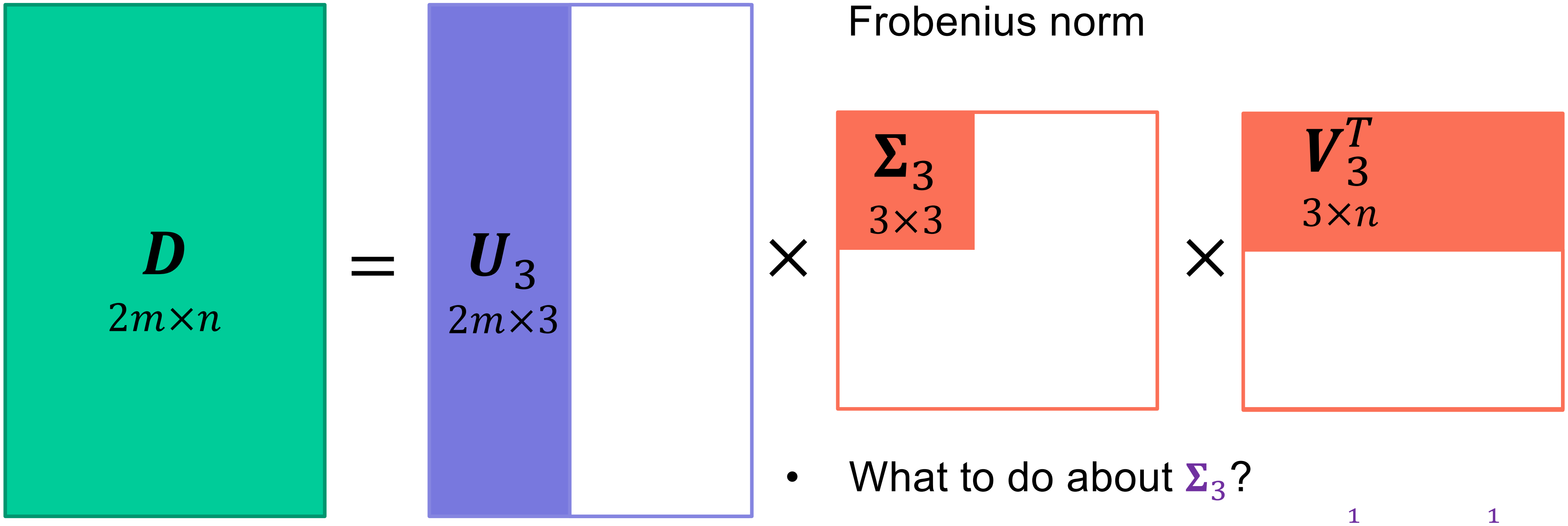
- Perform SVD of $\boldsymbol{D}$:



$$\underset{2m\times n}{\boldsymbol{D}} = \underset{2m\times n}{\boldsymbol{U}} \times \underset{n\times n}{\boldsymbol{\Sigma}} \times \underset{n\times n}{\boldsymbol{V}^{T}}$$

Factorizing the measurement matrix

- Keep top 3 singular values:

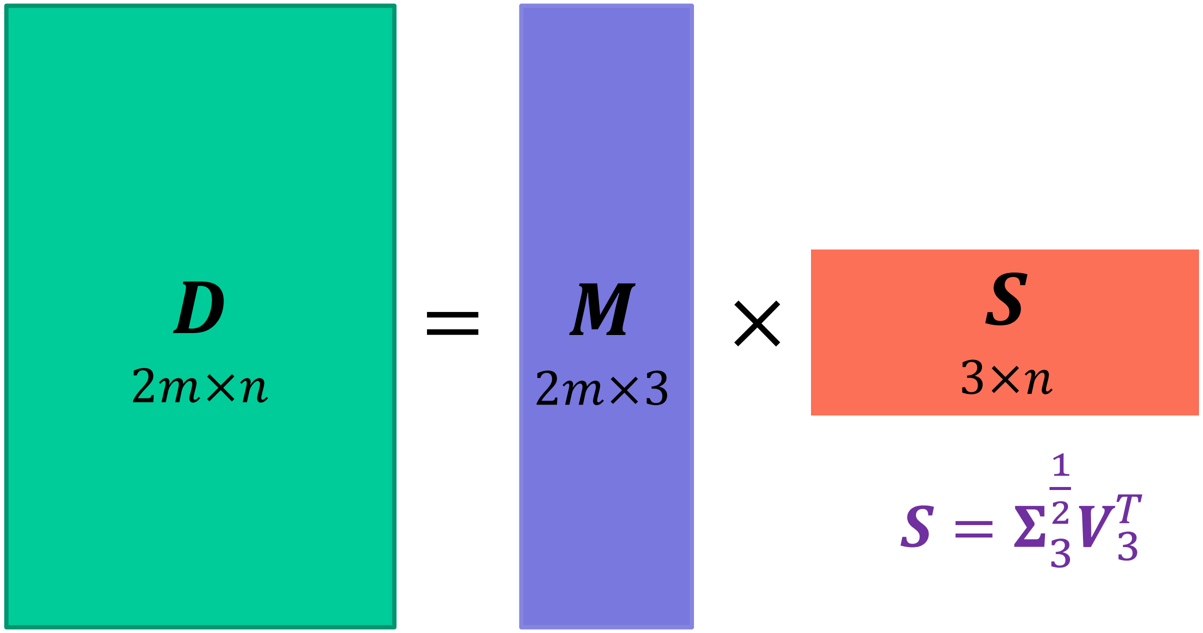


- This is the closest approximation of $\boldsymbol{D}$ with a rank-3 matrix in terms of Frobenius norm

- What to do about $\boldsymbol{\Sigma}_3$?
  - One solution: $\boldsymbol{M} = \boldsymbol{U}_3\boldsymbol{\Sigma}_3^{\frac{1}{2}}, \boldsymbol{S} = \boldsymbol{\Sigma}_3^{\frac{1}{2}}\boldsymbol{V}_3^T$

Factorizing the measurement matrix

- One possible solution:



$$\underset{2m\times n}{\boldsymbol{D}} = \underset{2m\times 3}{\boldsymbol{M}} \times \underset{3\times n}{\boldsymbol{S}}$$

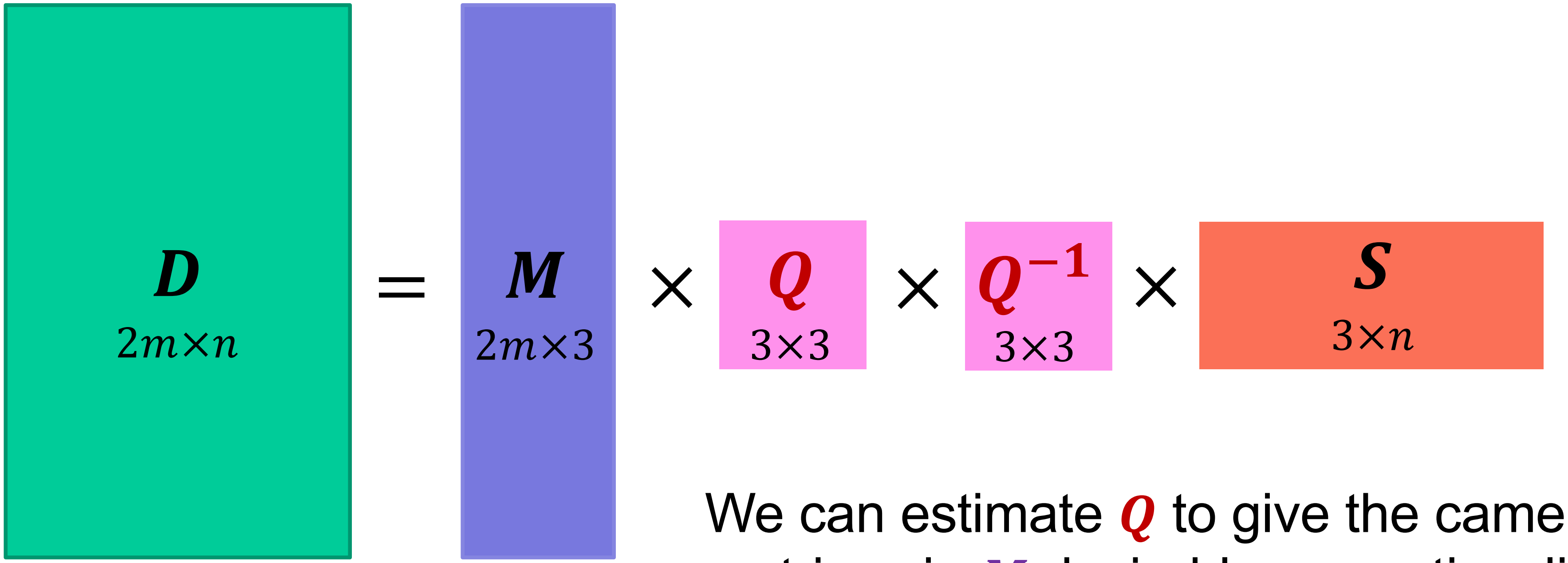
$$\boldsymbol{S} = \boldsymbol{\Sigma}_3^{\frac{1}{2}} \boldsymbol{V}_3^T$$

$$\boldsymbol{M} = \boldsymbol{U}_3 \boldsymbol{\Sigma}_3^{\frac{1}{2}}$$

- Are there other solutions?

Factorizing the measurement matrix

- Other possible solutions:



$$\underset{2m\times n}{\boldsymbol{D}} = \underset{2m\times 3}{\boldsymbol{M}} \times \underset{3\times 3}{\boldsymbol{Q}} \times \underset{3\times 3}{\boldsymbol{Q}^{-1}} \times \underset{3\times n}{\boldsymbol{S}}$$

We can estimate $\boldsymbol{Q}$ to give the camera matrices in $\boldsymbol{M}$ desirable properties, like orthographic projection

Eliminating the affine ambiguity

- So far, we have obtained one solution:

$$\boldsymbol{D} = \underset{2m\times 3}{\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_m \end{bmatrix}} \underset{3\times n}{[\boldsymbol{X}_1 \quad \boldsymbol{X}_2 \quad \cdots \quad \boldsymbol{X}_n]}$$

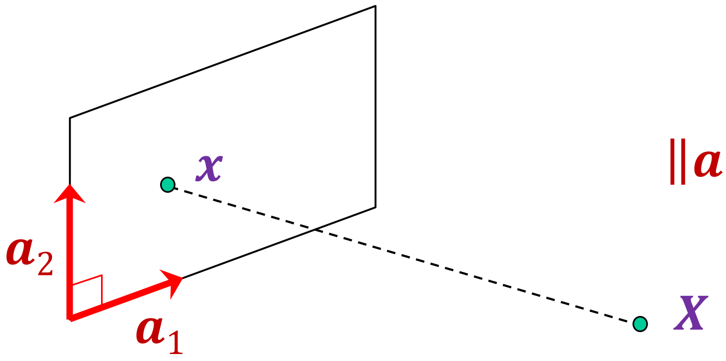
- We want:

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{A}_1\boldsymbol{Q} \\ \boldsymbol{A}_2\boldsymbol{Q} \\ \vdots \\ \boldsymbol{A}_m\boldsymbol{Q} \end{bmatrix} [\boldsymbol{Q}^{-1}\boldsymbol{X}_1 \quad \boldsymbol{Q}^{-1}\boldsymbol{X}_2 \quad \cdots \quad \boldsymbol{Q}^{-1}\boldsymbol{X}_n]$$

such that each camera matrix $\boldsymbol{A}_i\boldsymbol{Q}$ represents orthographic projection, i.e., has orthonormal axes (rows)

Eliminating the affine ambiguity

- Let $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ be the rows of a $2{\times}3$ orthographic projection matrix. Then



$$\boldsymbol{a}_1 \cdot \boldsymbol{a}_2 = 0$$
$$\|\boldsymbol{a}_1\|^2 = \|\boldsymbol{a}_2\|^2 = 1$$

- This translates into $3m$ constraints on the 9 entries of $\boldsymbol{Q}$:

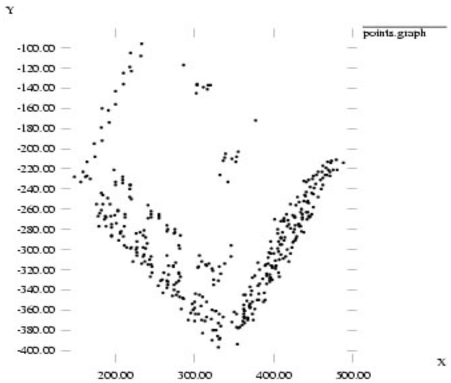
$$(\boldsymbol{A}_i\boldsymbol{Q})(\boldsymbol{A}_i\boldsymbol{Q})^T = \boldsymbol{A}_i(\boldsymbol{Q}\boldsymbol{Q}^T)\boldsymbol{A}_i^T = \mathbf{I}_{2\times 2}, \qquad i = 1, \dots, m$$

  - Are the constraints linear?
  - First, solve for $\boldsymbol{L} = \boldsymbol{Q}\boldsymbol{Q}^T$
  - Recover $\boldsymbol{Q}$ from $\boldsymbol{L}$ by Cholesky decomposition
  - Update $\boldsymbol{M}$ to $\boldsymbol{M}\boldsymbol{Q}$, $\boldsymbol{S}$ to $\boldsymbol{Q}^{-1}\boldsymbol{S}$

Reconstruction results

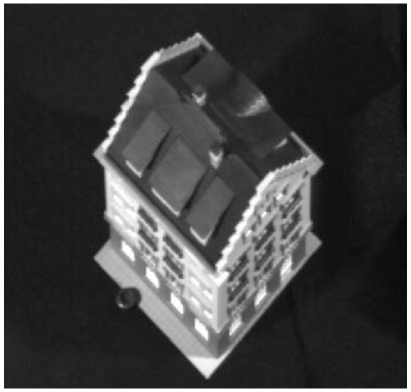
1

60



120

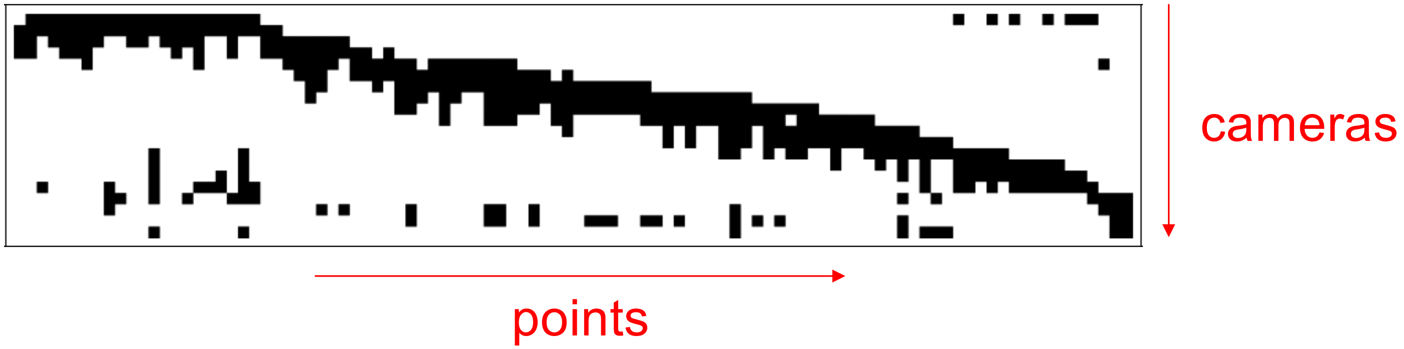
150



C. Tomasi and T. Kanade, Shape and motion from image streams under orthography: A factorization method, IJCV 1992

Dealing with missing data

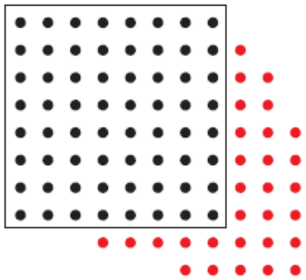
- So far, we have assumed that all points are visible in all views
- In reality, the measurement matrix typically looks something like this:



- Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results
  - Unfortunately, finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)

Dealing with missing data

- Incremental bilinear refinement:



Perform factorization on a dense sub-block

Solve for a new 3D point visible by at least two known cameras – triangulation

Solve for a new camera that sees at least three known 3D points – calibration

F. Rothganger et al. Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving Objects. PAMI 2007.

Outline: Structure from motion

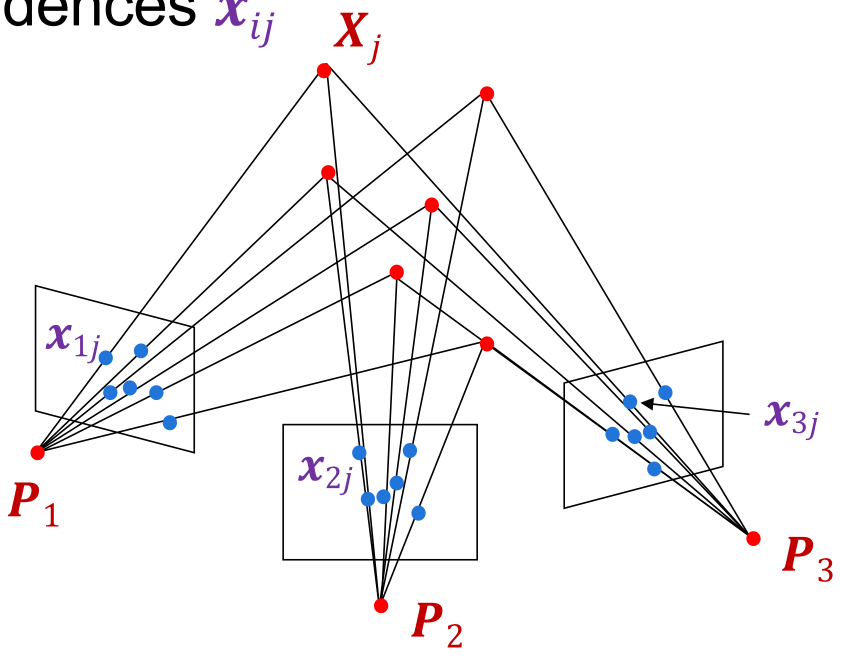
- Problem definition and ambiguities
- Affine structure from motion
  - Factorization
- Projective structure from motion

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 $\boldsymbol{x}_{ij}$



Projective structure from motion

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

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

- **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 $4{\times}4$ 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

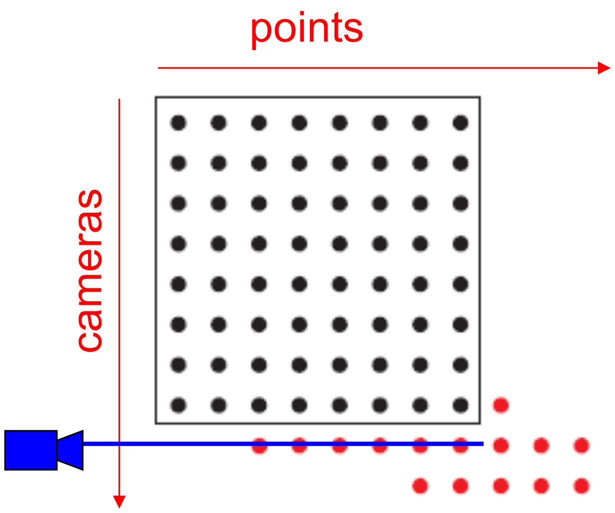
1. Estimate fundamental matrix $\boldsymbol{F}$ between the two views
2. Set first camera matrix to $[\boldsymbol{I} \mid \boldsymbol{0}]$
3. Then the second camera matrix is given by $[\boldsymbol{A} \mid \boldsymbol{t}]$ where $\boldsymbol{t}$ is the epipole ($\boldsymbol{F}^T\boldsymbol{t} = \boldsymbol{0}$)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

- In practice, SFM pipelines use guesses of intrinsic parameters and the five-point algorithm

F&P sec. 8.3.2

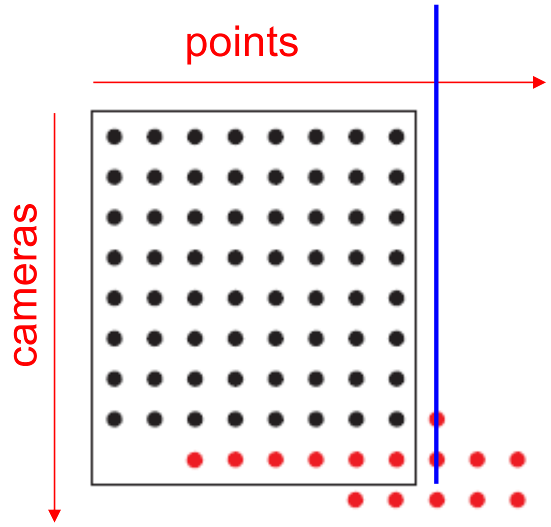
Incremental structure from motion

- 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



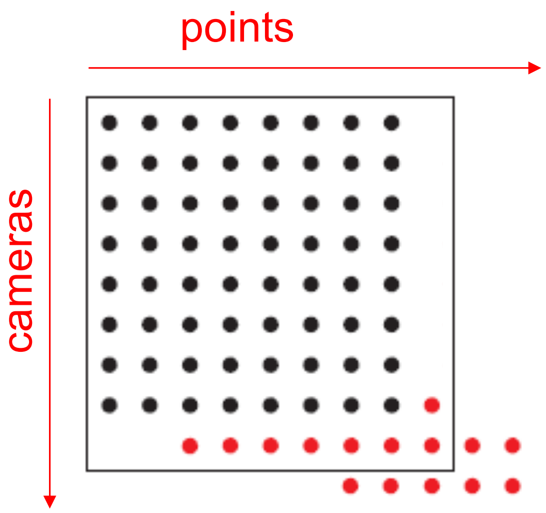
Incremental structure from motion

- 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

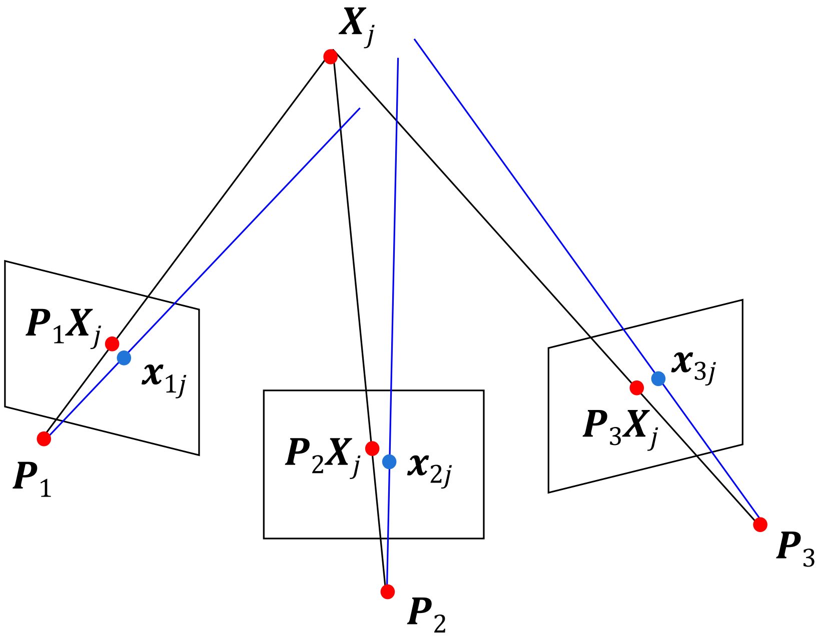


Bundle adjustment

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

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

visibility flag: is point j visible in view i?

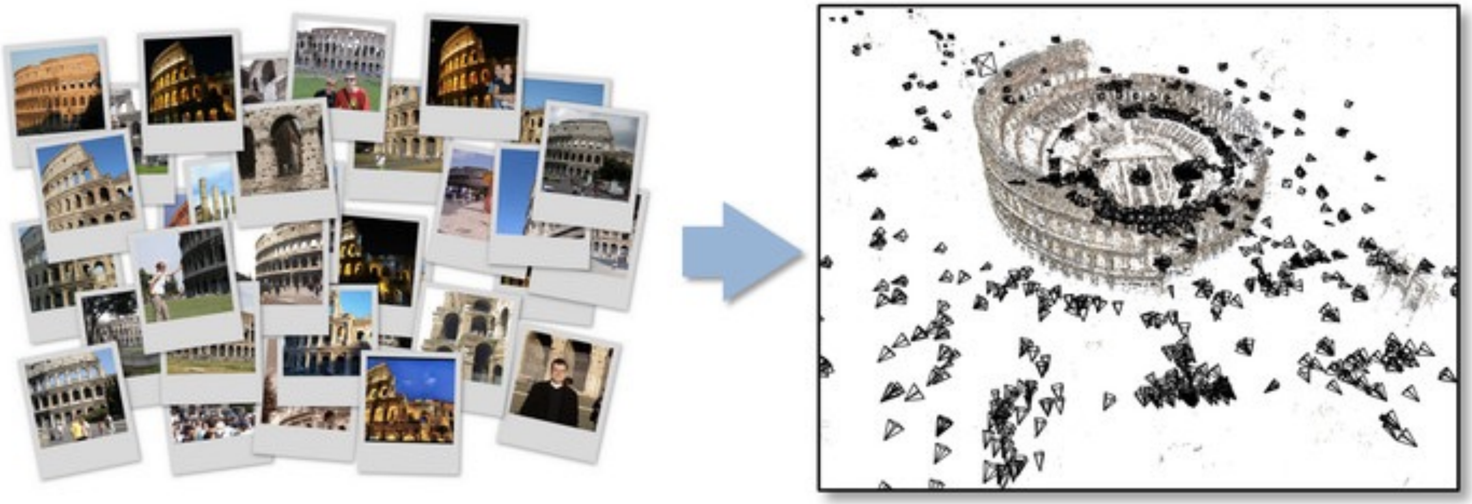


B. Triggs et al. Bundle adjustment – A modern synthesis. International Workshop on Vision Algorithms, 1999

Outline: Structure from motion

- Problem definition and ambiguities
- Affine structure from motion
  - Factorization
- Projective structure from motion
  - Incremental reconstruction, bundle adjustment
- Modern structure from motion pipeline

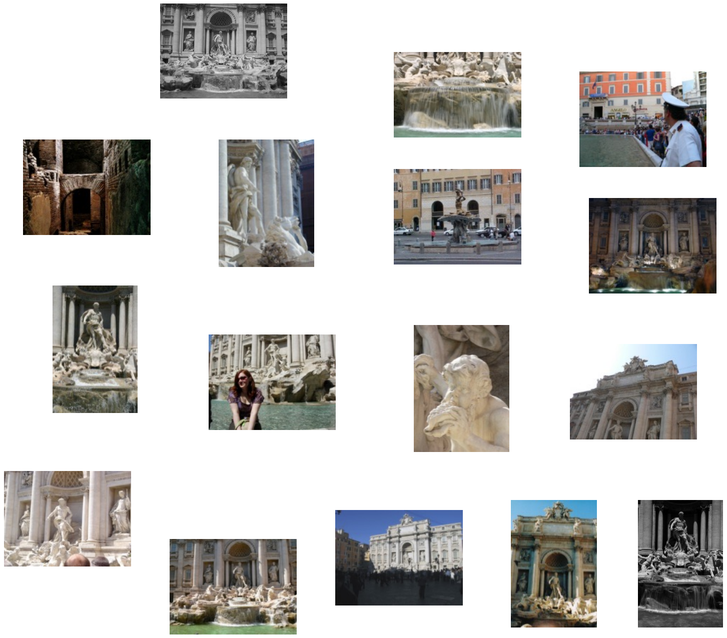
Representative SFM pipeline



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

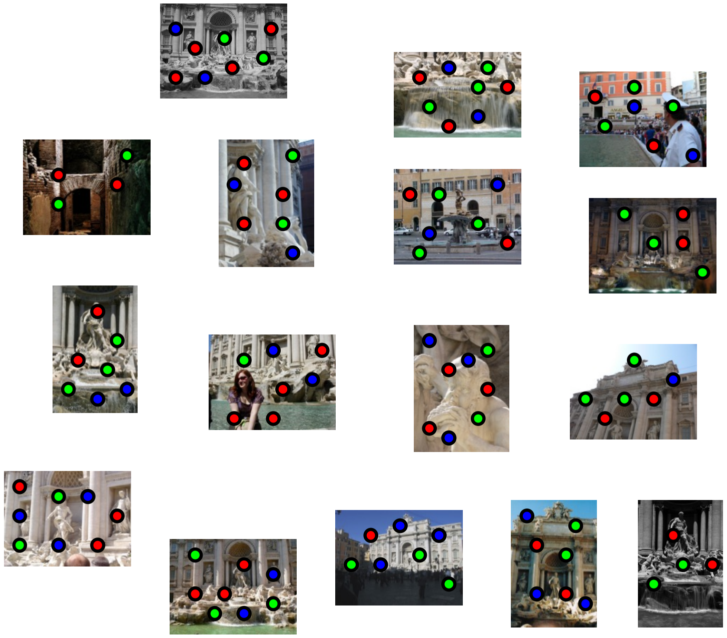


Source: N. Snavely

Feature detection

Detect SIFT features

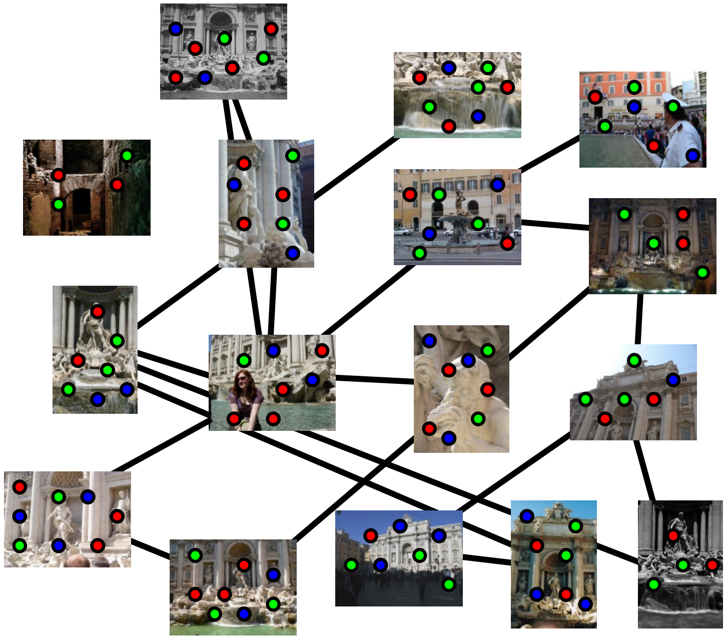
Other popular feature types: SURF, ORB, BRISK, …



Source: N. Snavely

Feature matching

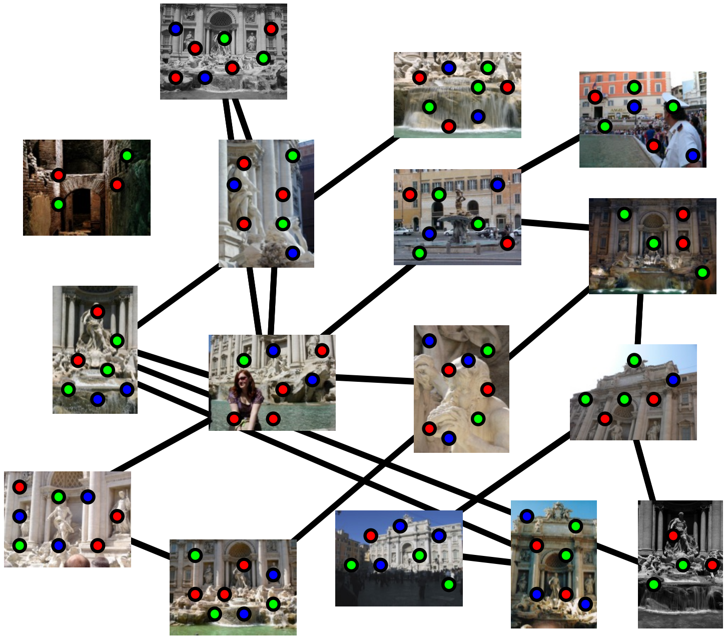
Match features between each pair of images



Source: N. Snavely

Feature matching

Use RANSAC to estimate fundamental matrix between each pair



Source: N. Snavely

Feature matching

Use RANSAC to estimate fundamental matrix between each pair

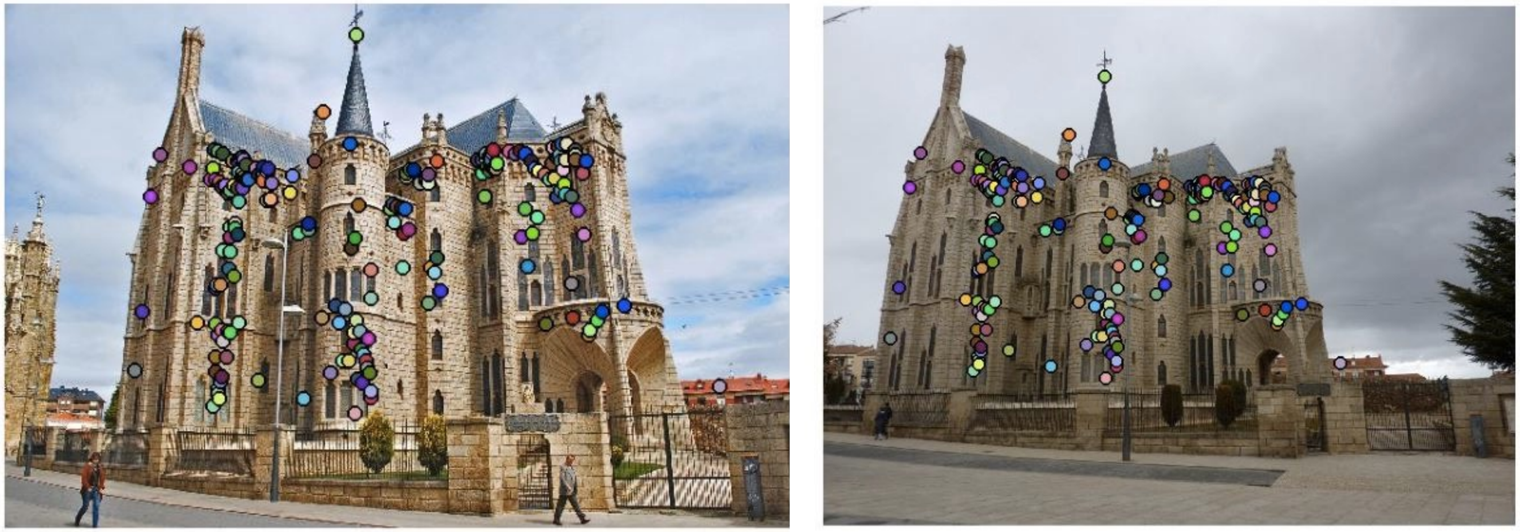
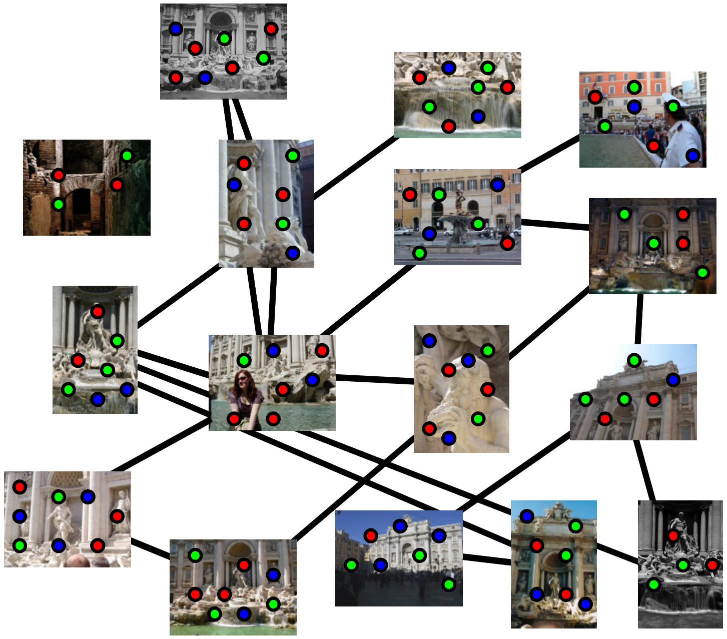


Image source

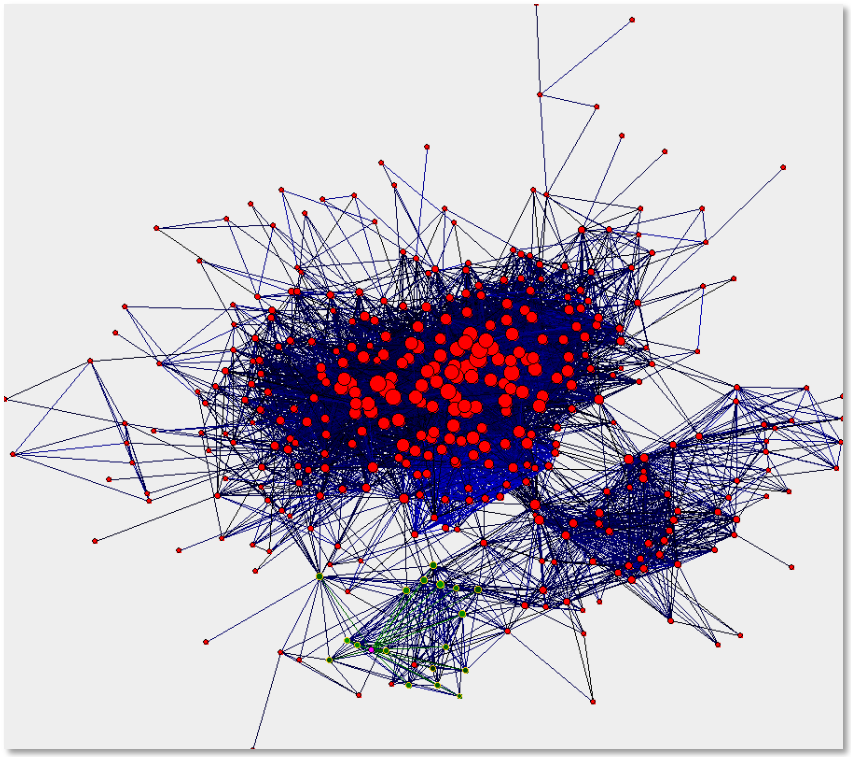
Feature matching

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

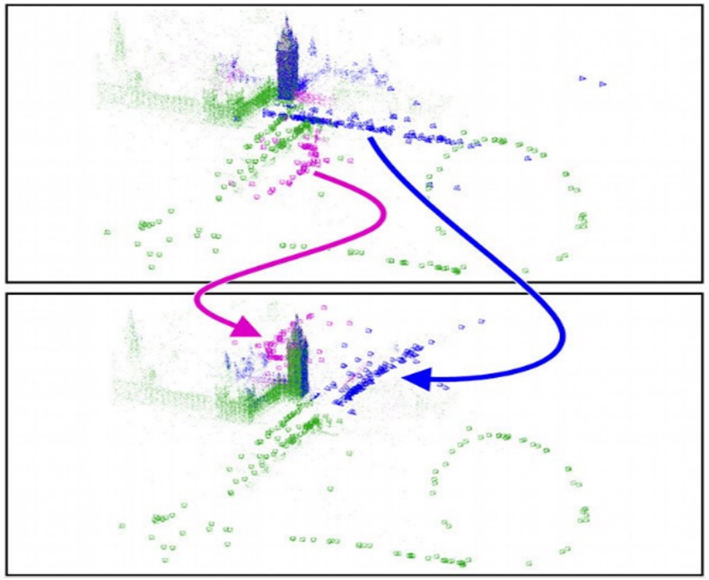
Incremental SFM

- 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

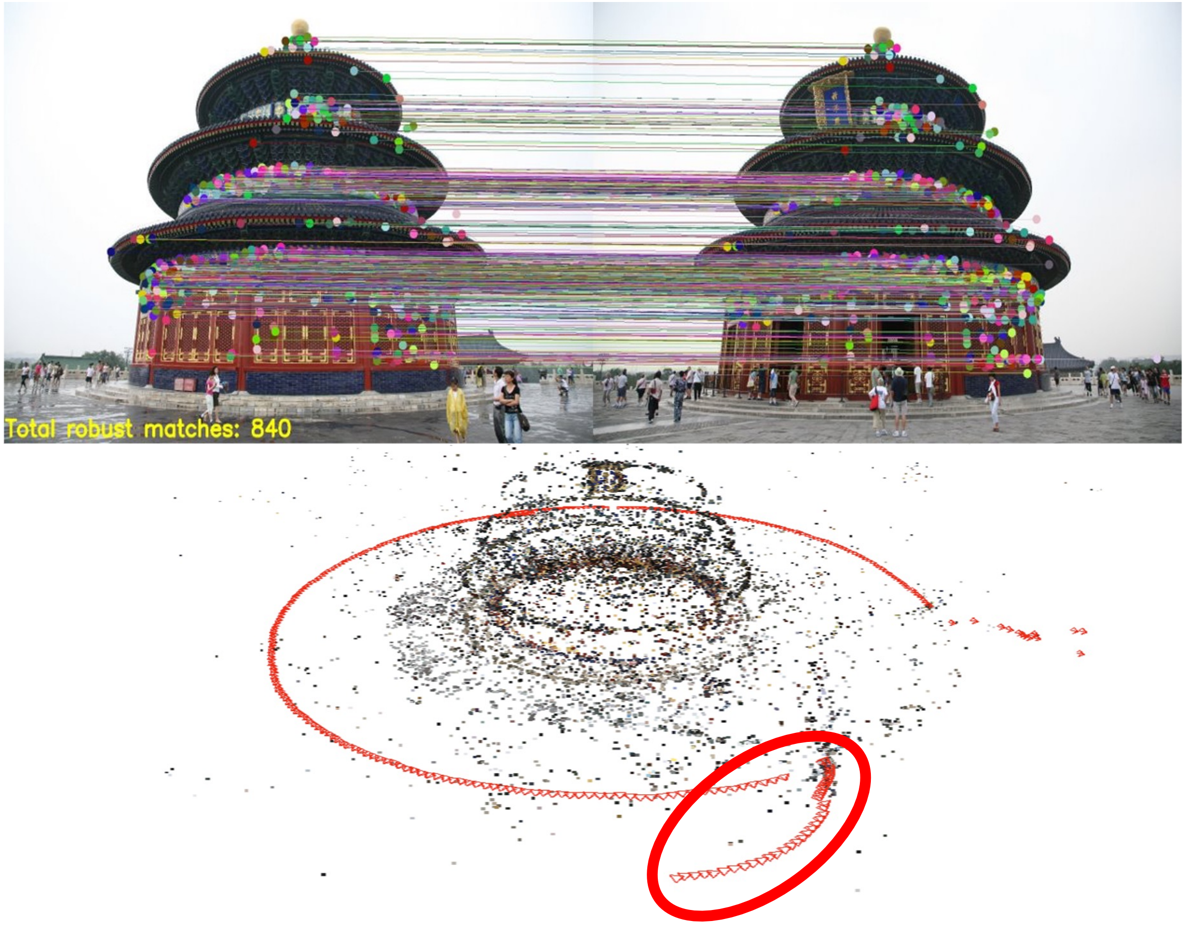
- Handling degenerate configurations (e.g., homographies)
- Filtering out incorrect matches
- Dealing with repetitions and symmetries

Repetitive structures cause catastrophic failures



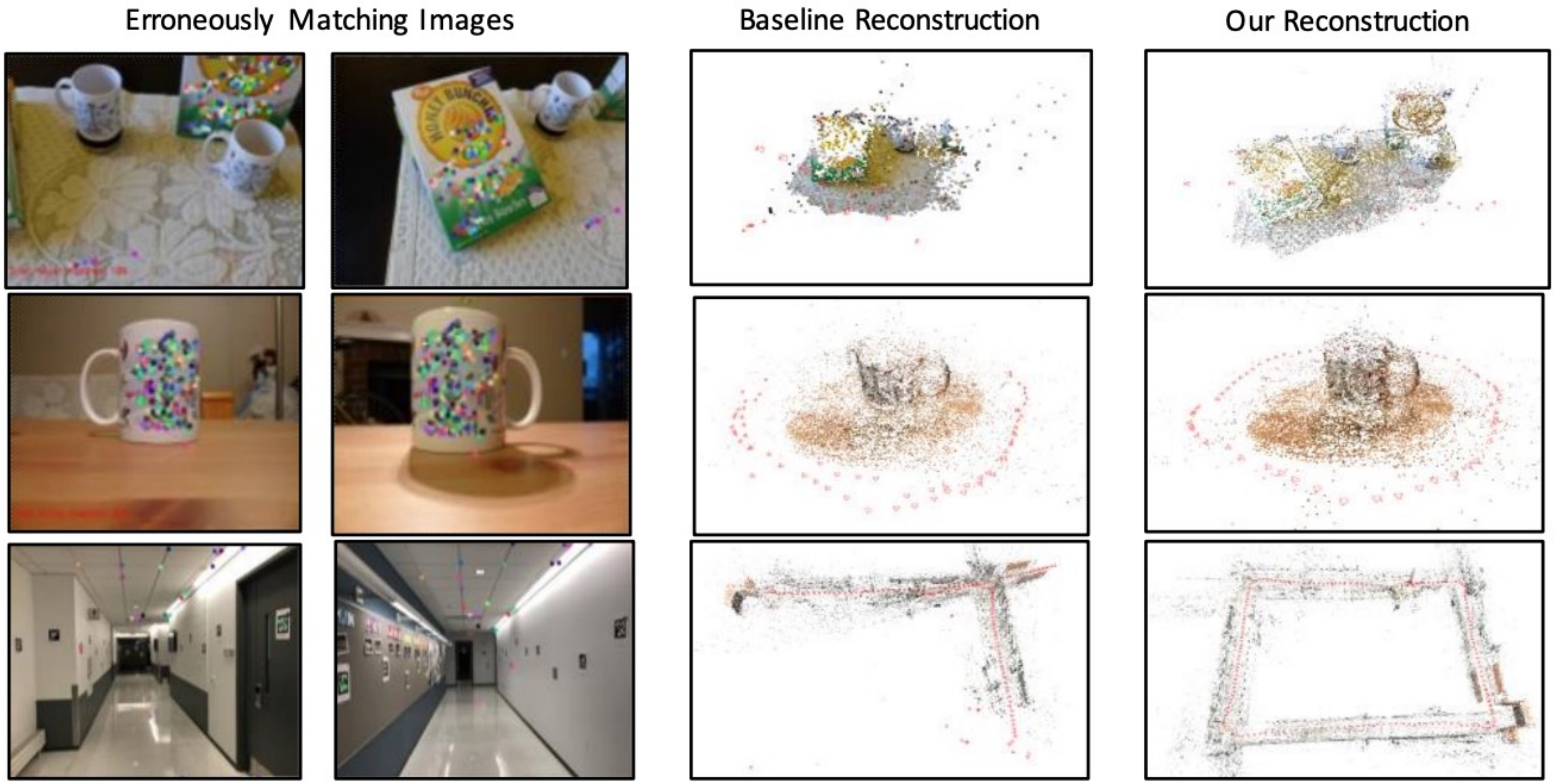
https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

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

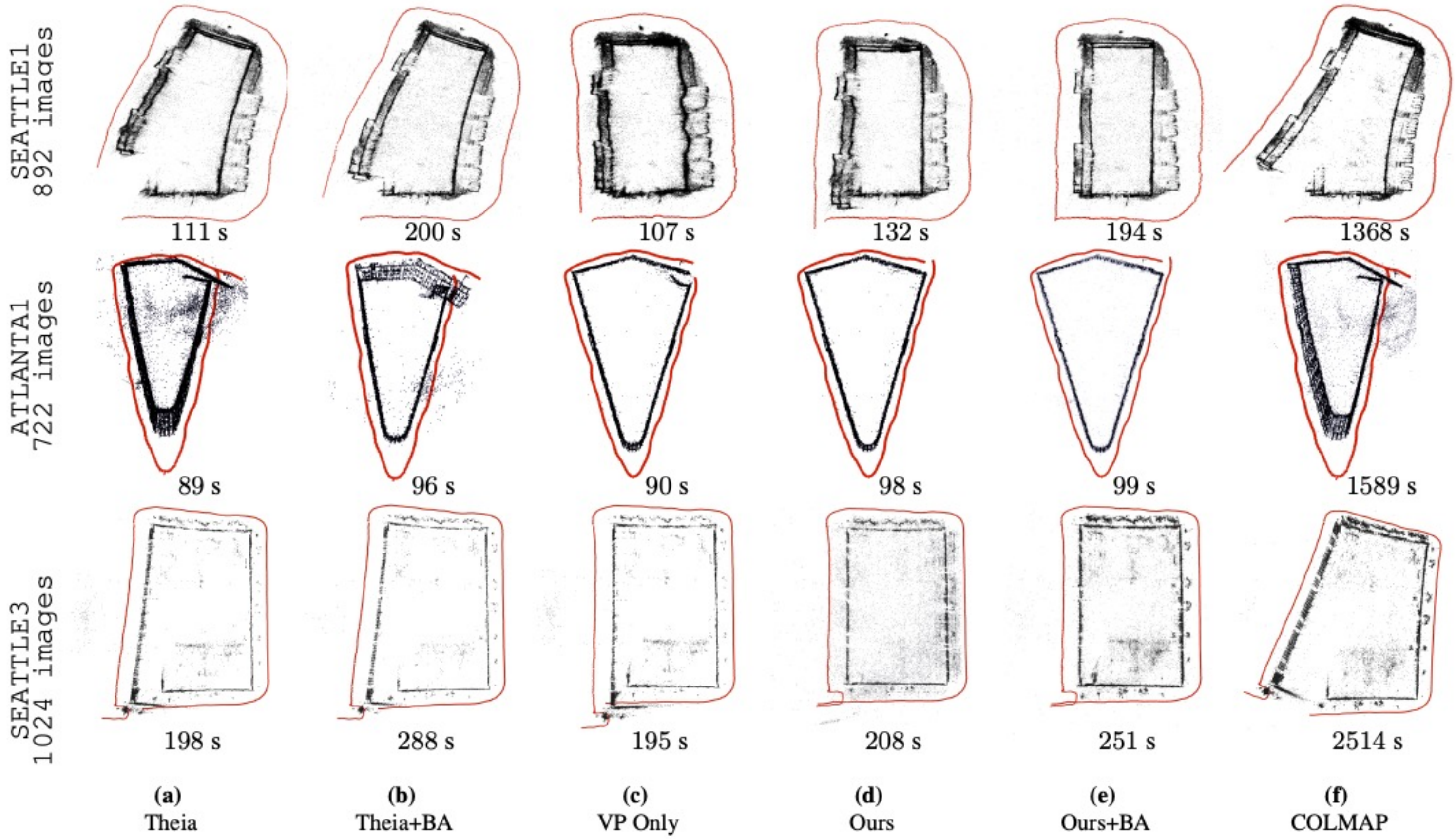
The devil is in the details

- 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

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