Structure from motion
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: $K_1, R_1, t_1$

Camera 2: $K_2, R_2, t_2$

Camera 3: $K_3, R_3, t_3$
Recall: Calibration

- Given a set of known 3D points seen by a camera, compute the camera parameters $K_1, R_1, t_1$, $K_2, R_2, t_2$, and $K_3, R_3, t_3$. 

*camera 1* 

- $K_1, R_1, t_1$ ?

*camera 2* 

- $K_2, R_2, t_2$ ?

*camera 3* 

- $K_3, R_3, t_3$ ?

• 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)

\[
x_{ij} \cong P_i X_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n
\]

- Problem: estimate \( m \) projection matrices \( P_i \) and \( n \) 3D points \( X_j \) from the \( mn \) correspondences \( x_{ij} \)
Is SFM always uniquely solvable?

Source: N. Snavely

Necker cube
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:

$$x \cong PX = \left(\frac{1}{k} P\right)(kX)$$

• 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 $Q$ and apply the inverse transformation to the camera matrices, then the image observations do not change:

$$x \cong PX = (PQ^{-1})(QX)$$
Projective ambiguity

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

\[ x \cong PX = (PQ^{-1})(QX) \]

\( Q \) 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:

\[ x \cong PX = \left( PQ_A^{-1}\right)(Q_AX) \]

\[ Q_A = \begin{bmatrix} A & t \\ 0_T & 1 \end{bmatrix} \]
Affine ambiguity
Similarity ambiguity

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

\[ x \cong PX = (PQ_S^{-1})(Q_SX) \]

\[ Q_S = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} \]

- 3x3 rotation matrix
- 3x1 translation vector
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
Recall: Orthographic projection

Just drop the z coordinate!

\[
\begin{pmatrix}
  x \\
  y \\
  1
\end{pmatrix} = \begin{bmatrix}
  \end{bmatrix}
\begin{pmatrix}
  x \\
  y \\
  z \\
  1
\end{pmatrix}
\]
General affine projection

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

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

- In non-homogeneous coordinates:

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

- \(a_1, a_2\): rows of projection matrix

Projection of world origin
Affine structure from motion

• **Given:** $m$ images of $n$ fixed 3D points such that 
  \[ x_{ij} = A_i X_j + t_i, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \]

• **Problem:** use the $mn$ correspondences $x_{ij}$ to estimate $m$ projection matrices $A_i$ and translation vectors $t_i$, and $n$ points $X_j$

• The reconstruction is defined up to an arbitrary affine transformation $Q$ (12 degrees of freedom):
  \[ \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} Q^{-1}, \quad \begin{bmatrix} X_j \\ 1 \end{bmatrix} \rightarrow Q \begin{bmatrix} X_j \\ 1 \end{bmatrix} \]

• 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:

\[
\hat{x}_{ij} = x_{ij} - \frac{1}{n} \sum_{k=1}^{n} x_{ik}
\]

\[
= A_i X_j + t_i - \frac{1}{n} \sum_{k=1}^{n} (A_i X_k + t_i)
\]

\[
= A_i \left( X_j - \frac{1}{n} \sum_{k=1}^{n} X_k \right)
\]

\[
= A_i \hat{X}_j
\]
Affine structure from motion

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

$$\hat{x}_{ij} = A_i \hat{X}_j$$

- 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:

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

Cameras ($2m$) -> Points ($n$)

$$\hat{x}_{ij} = A_iX_j$$

Affine structure from motion

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

$$D = \begin{bmatrix} \hat{x}_{11} & \hat{x}_{12} & \cdots & \hat{x}_{1n} \\ \hat{x}_{21} & \hat{x}_{22} & \cdots & \hat{x}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{m1} & \hat{x}_{m2} & \cdots & \hat{x}_{mn} \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_n \end{bmatrix}$$

- $M$ cameras ($2m \times 3$)
- $S$ points ($3 \times n$)

• What must be the rank of the measurement matrix $D = MS$?

Factorizing the measurement matrix

- We want:

\[ D = M \times S \]
Factorizing the measurement matrix

• Perform SVD of $\mathbf{D}$:

$$
\mathbf{D}_{2m \times n} = \mathbf{U}_{2m \times n} \times \mathbf{\Sigma}_{n \times n} \times \mathbf{V}^T_{n \times n}
$$
Factorizing the measurement matrix

- Keep top 3 singular values:

\[ D = U_3 \Sigma_3 V_3^T \]

- This is the closest approximation of \( D \) with a rank-3 matrix in terms of Frobenius norm.

- What to do about \( \Sigma_3 \)?

- One solution: \( M = U_3 \Sigma_3^{\frac{1}{2}} S = \Sigma_3^{\frac{1}{2}} V_3^T \)
Factorizing the measurement matrix

• One possible solution:

\[ D_{2m \times n} = M_{2m \times 3} \times S_{3 \times n} \]

\[ M = U_3 \Sigma_3^{1 \frac{1}{2}} \]

\[ S = \Sigma_3^{2} V_3^T \]

• Are there other solutions?
Factorizing the measurement matrix

- Other possible solutions:

\[ D_{2m \times n} = M_{2m \times 3} \times Q_{3 \times 3} \times Q^{-1}_{3 \times 3} \times S_{n \times n} \]

We can estimate \( Q \) to give the camera matrices in \( M \) desirable properties, like orthographic projection.
Eliminating the affine ambiguity

• So far, we have obtained one solution:

\[ D = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} [X_1 \quad X_2 \quad \cdots \quad X_n]_{3 \times m} \]

\[ 2n \times 3 \]

• We want:

\[ D = \begin{bmatrix} A_1 Q \\ A_2 Q \\ \vdots \\ A_m Q \end{bmatrix} [Q^{-1}X_1 \quad Q^{-1}X_2 \quad \cdots \quad Q^{-1}X_n] \]

such that each camera matrix \( A_i Q \) represents orthographic projection, i.e., has orthonormal axes (rows)
Eliminating the affine ambiguity

• Let $a_1$ and $a_2$ be the rows of a $2 \times 3$ orthographic projection matrix. Then

- This translates into $3m$ constraints on the 9 entries of $Q$:

$$ (A_i Q)(A_i Q)^T = A_i (QQ^T)A_i^T = I_{2 \times 2}, \quad i = 1, \ldots, m $$

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

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

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):

$$x_{ij} \cong P_i X_j, \ i = 1, \ldots, m, \ j = 1, \ldots, n$$

- **Problem**: estimate $m$ projection matrices $P_i$ and $n$ 3D points $X_j$ from the $mn$ correspondences $x_{ij}$.
Projective structure from motion

• **Given:** \( m \) images of \( n \) fixed 3D points such that (ignoring visibility):
  \[
  x_{ij} \simeq P_i X_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n
  \]

• **Problem:** estimate \( m \) projection matrices \( P_i \) and \( n \) 3D points \( X_j \) from the \( mn \) correspondences \( x_{ij} \)

• With no calibration info, cameras and points can only be recovered up to a \( 4 \times 4 \) projective transformation \( Q \):
  \[
  X \rightarrow QX, \quad P \rightarrow PQ^{-1}
  \]

• 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 $F$ between the two views
2. Set first camera matrix to $[I | 0]$
3. Then the second camera matrix is given by $[A | t]$ where $t$ is the epipole ($F^T t = 0$) and $A = -[t \times] F$

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

$$\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} d \left( x_{ij} - \text{proj}(P_i X_j) \right)^2$$

Visibility flag: is point $j$ visible in view $i$?

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

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
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 ($R$ and $t$) 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

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

Reducing error accumulation and closing loops

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