Structure from Motion

CSC 767
Structure from motion

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates.
Structure from motion

- Given: \( m \) images of \( n \) fixed 3D points

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

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

Necker cube

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 in the image remain exactly the same:

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

It is impossible to recover the absolute scale of the scene!
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 in the image remain exactly the same.

- More generally, if we transform the scene using a transformation $Q$ and apply the inverse transformation to the camera matrices, then the images do not change:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = (\mathbf{PQ}^{-1})(\mathbf{QX})$$
Types of ambiguity

- **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 parallelism, 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

- With no constraints on the camera calibration matrix or on the scene, we get a *projective* reconstruction.
- Need additional information to *upgrade* the reconstruction to affine, similarity, or Euclidean.
Projective ambiguity

\[
Q_p = \begin{bmatrix}
A & t \\
\mathbf{v}^T & v
\end{bmatrix}
\]

\[
x = PX = \left(PQ_P^{-1}\right)(Q_PX)
\]
Projective ambiguity
Affine ambiguity

\[ Q_A = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} \]

\[ x = PX = \left( PQ_A^{-1} \right) (Q_A x) \]
Affine ambiguity
Similarity ambiguity

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

\[ x = PX = (PQ_s^{-1})(Q_sX) \]
Similarity ambiguity
Structure from motion

• Let’s start with **affine** or **weak perspective** cameras (the math is easier)
Orthographic Projection

Projection along the z direction

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= 
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix} 
\Rightarrow (x, y)
\]
Affine cameras

Orthographic Projection

Parallel Projection
Affine cameras

• A general affine camera combines the effects of an affine transformation of the 3D space, orthographic projection, and an affine transformation of the image:

\[
P = [3 \times 3 \text{ affine}] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} [4 \times 4 \text{ affine}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix}
\]

• Affine projection is a linear mapping + translation in non-homogeneous coordinates

\[
x = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = AX + b
\]

Projection of world origin
Affine structure from motion

• Given: $m$ images of $n$ fixed 3D points:

$$x_{ij} = A_i X_j + b_i, \quad i = 1, \ldots , m, \quad j = 1, \ldots , n$$

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

• The reconstruction is defined up to an arbitrary affine transformation $Q$ (12 degrees of freedom):

$$\begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} Q^{-1}, \quad \begin{pmatrix} X \\ 1 \end{pmatrix} \rightarrow Q \begin{pmatrix} X \\ 1 \end{pmatrix}$$

• We have $2mn$ knowns and $8m + 3n$ unknowns (minus 12 dof for affine ambiguity)

• Thus, we must have $2mn \geq 8m + 3n - 12$

• For two views, we need four point correspondences
Affine structure from motion

• Centering: subtract the centroid of the image points in each view

\[
\hat{x}_{ij} = x_{ij} - \frac{1}{n} \sum_{k=1}^{n} x_{ik} = A_i X_j + b_i - \frac{1}{n} \sum_{k=1}^{n} (A_i X_k + b_i)
\]

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

• For simplicity, set the origin of the world coordinate system to the centroid of the 3D points

• After centering, each normalized 2D point is related to the 3D point \( X_j \) by

\[
\hat{x}_{ij} = A_i X_j
\]
Affine structure from motion

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

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

points ($n$)
cameras (2$m$)

Affine structure from motion

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

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

The measurement matrix $D = MS$ must have rank 3!

Factorizing the measurement matrix

\[ \text{Measurements} = \text{Motion} \times \text{Shape} \]

\[ D = MS \]

Source: M. Hebert
Factorizing the measurement matrix

- Singular value decomposition of $D$:

\[
\begin{align*}
D & = U \times W \times V^T \\
& = U_3 \times W_3 \times V_3^T
\end{align*}
\]

Source: M. Hebert
Factorizing the measurement matrix

- Singular value decomposition of $D$:

$$
\begin{align*}
D &= U \times W \times V^T \\
U_3 &= W_3 \times V_3^T
\end{align*}
$$

To reduce to rank 3, we just need to set all the singular values to 0 except for the first 3.
Factorizing the measurement matrix

- Obtaining a factorization from SVD:

\[ D = U_3 \times W_3 \times V_3^T \]
Factorizing the measurement matrix

- Obtaining a factorization from SVD:

\[ M = U_3 W_3^{1/2} \quad S = W_3^{1/2} V_3^T \]

This decomposition minimizes \(|D-MS|^2\)

Source: M. Hebert
Affine ambiguity

- The decomposition is not unique. We get the same $D$ by using any $3 \times 3$ matrix $C$ and applying the transformations $M \rightarrow MC, S \rightarrow C^{-1}S$.

- That is because we have only an affine transformation and we have not enforced any Euclidean constraints (like forcing the image axes to be perpendicular, for example).

Source: M. Hebert
Eliminating the affine ambiguity

- Transform each projection matrix $A$ to another matrix $AC$ to get orthographic projection
  - Image axes are perpendicular and scale is 1

  This translates into $3m$ equations:

  $$ (A_i C)(A_i C)^T = A_i (CC^T) A_i = \text{Id}, \quad i = 1, \ldots, m $$

  - Solve for $L = CC^T$
  - Recover $C$ from $L$ by Cholesky decomposition: $L = CC^T$
  - Update $M$ and $S$: $M = MC$, $S = C^{-1}S$

Source: M. Hebert
Reconstruction results

Algorithm summary

• Given: \( m \) images and \( n \) features \( x_{ij} \)

• For each image \( i \), center the feature coordinates

• Construct a \( 2m \times n \) measurement matrix \( \mathbf{D} \):
  • Column \( j \) contains the projection of point \( j \) in all views
  • Row \( i \) contains one coordinate of the projections of all the \( n \) points in image \( i \)

• Factorize \( \mathbf{D} \):
  • Compute SVD: \( \mathbf{D} = \mathbf{U} \mathbf{W} \mathbf{V}^T \)
  • Create \( \mathbf{U}_3 \) by taking the first 3 columns of \( \mathbf{U} \)
  • Create \( \mathbf{V}_3 \) by taking the first 3 columns of \( \mathbf{V} \)
  • Create \( \mathbf{W}_3 \) by taking the upper left \( 3 \times 3 \) block of \( \mathbf{W} \)

• Create the motion and shape matrices:
  • \( \mathbf{M} = \mathbf{U}_3 \mathbf{W}_3^{\frac{1}{2}} \) and \( \mathbf{S} = \mathbf{W}_3^{\frac{1}{2}} \mathbf{V}_3^T \) (or \( \mathbf{M} = \mathbf{U}_3 \) and \( \mathbf{S} = \mathbf{W}_3 \mathbf{V}_3^T \))

• Eliminate affine ambiguity

Source: M. Hebert
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:
Dealing with missing data

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

- Incremental bilinear refinement

1. Perform factorization on a dense sub-block
2. Solve for a new 3D point visible by at least two known cameras (linear least squares)
3. Solve for a new camera that sees at least three known 3D points (linear least squares)

Projective structure from motion

- Given: $m$ images of $n$ fixed 3D points

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

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

• Given: $m$ images of $n$ fixed 3D points

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

• 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 4x4 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

• Compute fundamental matrix $F$ between the two views

• First camera matrix: $[I \mid 0]$

• Second camera matrix: $[A \mid b]$

• Then $b$ is the epipole ($F^Tb = 0$), $A = -[b \times]F$
Sequential 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*
Sequential 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*
Sequential 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
- Minimizing reprojection error

\[ E(P, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, P_i X_j)^2 \]
Modern SFM pipeline

Feature detection

Detect features using SIFT [Lowe, IJCV 2004]

Source: N. Snavely
Feature detection

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
Image connectivity graph

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

Source: N. Snavely
Review: Structure from motion

- Ambiguity
- Affine structure from motion
  - Factorization
- Dealing with missing data
  - Incremental structure from motion
- Projective structure from motion
  - Bundle adjustment
  - Modern structure from motion pipeline
Credit:
Slide set developed by S. Lazebnik, University of Illinois at Urbana-Champaign