Structure from motion
Outline

• Representative SfM pipeline
  • Incremental SfM
  • Bundle adjustment
• Ambiguities in SfM
• Special Case: Affine structure from motion
  • Factorization
• SfM in practice
Structure from motion

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

\[ R_1, t_1 \]
\[ R_2, t_2 \]
\[ R_3, t_3 \]
Representative SFM pipeline

http://phototour.cs.washington.edu/

Slide from L. Lazebnik.
Feature detection

Detect SIFT features

Source: N. Snavely
Feature detection

Detect SIFT features

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

Slide from L. Lazebnik.
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
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 \( Q \):
\[
X \rightarrow QX, \ P \rightarrow PQ^{-1}
\]

• We can solve for structure and motion when
\[
2mn \geq 11m + 3n - 15
\]

• For two cameras, at least 7 points are needed

Slide from L. Lazebnik.
Projective SFM: Two-camera case

- Compute fundamental matrix \( F \) between the two views
- First camera matrix: \([I|0]\)
- Second camera matrix: \([A \ |b]\)
- where, \( b \) is the epipole (i.e. \( F^Tb = 0 \)) and \( A = [b \times]F \)

9.5.3 Canonical cameras given \( F \)

We have shown that \( F \) determines the camera pair up to a projective transformation of 3-space. We will now derive a specific formula for a pair of cameras with canonical form given \( F \). We will make use of the following characterization of the fundamental matrix \( F \) corresponding to a pair of camera matrices:

**Result 9.12.** A non-zero matrix \( F \) is the fundamental matrix corresponding to a pair of camera matrices \( P \) and \( P' \) if and only if \( P'^TFP \) is skew-symmetric.

**Proof.** The condition that \( P'^TFP \) is skew-symmetric is equivalent to \( X^TP'^TFPX = 0 \) for all \( X \). Setting \( x' = P'X \) and \( x = PX \), this is equivalent to \( x'^T F x = 0 \), which is the defining equation for the fundamental matrix.
Projective SFM: Two-camera case

- Compute fundamental matrix $F$ between the two views
- First camera matrix: $[I|0]$
- Second camera matrix: $[A \mid b]$
- where, $b$ is the epipole (i.e. $F^Tb = 0$) and $A = [b\times]F$

Result 9.13. Let $F$ be a fundamental matrix and $S$ any skew-symmetric matrix. Define the pair of camera matrices

$$P = [I \mid 0] \quad \text{and} \quad P' = [SF \mid e'],$$

where $e'$ is the epipole such that $e'^TF = 0$, and assume that $P'$ so defined is a valid camera matrix (has rank 3). Then $F$ is the fundamental matrix corresponding to the pair $(P, P')$.

Result 9.14. The camera matrices corresponding to a fundamental matrix $F$ may be chosen as $P = [I \mid 0]$ and $P' = [(e')_\times F \mid e']$. 

Slide from L. Lazebnik, Hartley and Zisserman
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

Slide from L. Lazebnik.
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

Slide from L. Lazebnik.
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

Slide from L. Lazebnik.
Bundle adjustment

• Non-linear method for refining structure and motion
• Minimize reprojection error

\[ \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \left\| x_{ij} - \frac{1}{\lambda_{ij}} P_i X_j \right\|^2 \]

visibility flag: is point j visible in view i?
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 ($\mathbf{R}$ and $\mathbf{t}$) 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

Slide from L. Lazebnik.
Photo Tourism
Exploring photo collections in 3D

Noah Snavely    Steven M. Seitz    Richard Szeliski
University of Washington    Microsoft Research

SIGGRAPH 2006

See also: http://grail.cs.washington.edu/projects/rome/
Outline

- Representative SfM pipeline
  - Incremental SfM
  - Bundle adjustment
- Ambiguities in SfM
- Special Case: Affine structure from motion
  - Factorization
- SfM in practice
Which way is the cube rotating?

- A. Left to right
- B. Right to left

Necker cube

Source: N. Snavely
Is SFM always uniquely solvable?

Necker cube

Source: N. Snavely
Is SFM always uniquely solvable?

- Necker reversal
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:

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:

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

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.
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:

$$x = PX = (PQ^{-1})(QX)$$
# Types of ambiguity

<table>
<thead>
<tr>
<th>Type</th>
<th>Transformation Matrix</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projective</td>
<td>$\begin{bmatrix} A &amp; t \ v^T &amp; 1 \end{bmatrix}$</td>
<td>Preserves intersection and tangency</td>
</tr>
<tr>
<td>15dof</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Affine</td>
<td>$\begin{bmatrix} A &amp; t \ 0^T &amp; 1 \end{bmatrix}$</td>
<td>Preserves parallellism, volume ratios</td>
</tr>
<tr>
<td>12dof</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Similarity</td>
<td>$\begin{bmatrix} sR &amp; t \ 0^T &amp; 1 \end{bmatrix}$</td>
<td>Preserves angles, ratios of length</td>
</tr>
<tr>
<td>7dof</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Euclidean</td>
<td>$\begin{bmatrix} R &amp; t \ 0^T &amp; 1 \end{bmatrix}$</td>
<td>Preserves angles, lengths</td>
</tr>
<tr>
<td>6dof</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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

• With no constraints on the camera calibration matrix or on the scene, we can reconstruct up to a *projective* ambiguity

\[
X = PX = \left( PQ_P^{-1} \right) (Q_P X)
\]

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

Slide from L. Lazebnik.
Projective ambiguity

Slide from L. Lazebnik.
Affine ambiguity

- If we impose parallelism constraints, we can get a reconstruction up to an 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

Slide from L. Lazebnik.
Similarity ambiguity

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

\[ X = PX = \left( PQ_s^{-1} \right)(Q_sX) \]

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

Slide from L. Lazebnik.
Similarity ambiguity

Slide from L. Lazebnik.
Outline

- Representative SfM pipeline
  - Incremental SfM
  - Bundle adjustment
- Ambiguities in SfM
- Special Case: Affine structure from motion
  - Factorization
- SfM in practice
Special Case: Affine structure from motion

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

Projection along the z direction

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

Slide from L. Lazebnik.
Affine cameras

Orthographic Projection

Parallel Projection

Slide from L. Lazebnik.
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:

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

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

\[
X = \begin{pmatrix}
x \\
y \\
z \\
\end{pmatrix}
= \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
0 & 0 & 0 \\
\end{bmatrix}
\begin{pmatrix}
X \\
Y \\
Z \\
\end{pmatrix}
+ \begin{pmatrix}
b_1 \\
b_2 \\
0 \\
\end{pmatrix}
= AX + b
\]
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

---

Slide from L. Lazebnik.
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:

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

- cameras (2m)

- points (n)

Affine structure from motion

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

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

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:

\[
\mathbf{D} = \mathbf{U} \times \mathbf{W} \times \mathbf{V}_3^T
\]

\[
\mathbf{D} = \mathbf{U}_3 \times \mathbf{W}_3 \times \mathbf{V}_3^T
\]

Source: M. Hebert
Factorizing the measurement matrix

• Singular value decomposition of D:

\[ \begin{align*}
2m & \quad D = \quad U \\
3 & \quad W \times V^T = \quad U_3 \\
3 & \quad W_3 \times V_3^T
\end{align*} \]

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:

\[ D = U_3 \times W_3 \times V_3^T \]

Source: M. Hebert
Factorizing the measurement matrix

- Obtaining a factorization from SVD:

\[ D = U_3 W_3 \times V_3^T \]

Possible decomposition:

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

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

Source: M. Hebert
Affine ambiguity

- The decomposition is not unique. We get the same $D$ by using any $3\times3$ 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:

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

- 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

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:

   ![Matrix Diagram]

   - 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).

Slide from L. Lazebnik.
Dealing with missing data

- 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 (triangulation)
3. Solve for a new camera that sees at least three known 3D points (calibration)

Outline

• Representative SfM pipeline
  • Incremental SfM
  • Bundle adjustment
• Ambiguities in SfM
• Special Case: Affine structure from motion
  • Factorization
• SfM in practice
The devil is in the details

- Handling degenerate configurations (e.g., homographies)
- Eliminating outliers
- Dealing with repetitions and symmetries
Repetitive structures

The devil is in the details

• Handling degenerate configurations (e.g., homographies)
• Eliminating outliers
• Dealing with repetitions and symmetries
• Handling multiple connected components
• Closing loops
• Making the whole thing efficient!
  • See, e.g., Towards Linear-Time Incremental Structure from Motion

Slide from L. Lazebnik.
SFM software

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

• Representative SfM pipeline
  • Incremental SfM
  • Bundle adjustment
• Ambiguities in SfM
• Special Case: Affine structure from motion
  • Factorization
• SfM in practice