Structure From Motion

EECS 442 – David Fouhey and Justin Johnson
Winter 2021, University of Michigan

https://web.eecs.umich.edu/~justincj/teaching/eecs442/WI2021/
Structure-from-Motion Revisited

Johannes L. Schönberger, Jan-Michael Frahm

CVPR 2016

Code available at: https://github.com/colmap/colmap
Structure from motion

Have: 2D points $p_{ij}$ seen in $m$ images
Assume: points generated from $n$ fixed 3D points $X_j$ and cameras $M_i$ or $p_{ij} \equiv M_iX_j$
Want: Cameras $M_i$, points $X_j$

(Remember)

$M_i \equiv K_i[R_i, t_i]$
$\lambda p_{ij} = M_iX_j, \lambda \neq 0$
Is SFM always uniquely solvable?

- Necker cube

Source: N. Snavely
Structure from motion ambiguities

Let’s first find one easy ambiguity

\[ p_{ij} \equiv M_i X_j \]

3x1 3x4 4x1
Structure from motion ambiguities

Let’s first find one easy ambiguity

\[ p_{ij} \equiv M_i X_j \]

Can pick any arbitrary scaling factor \( k \) and adjust the cameras and points

\[ p_{ij} \equiv M_i k^{-1} k X_j \]

(Can usually be fixed in practice: just need a number, obtainable from heights of known objects or an IMU)
Structure from motion ambiguity

Does this diagram change meaning if I use this coordinate system?

Versus this coordinate system?

Coordinate system irrelevant!
So global $R, t$ also ambiguous
Structure from motion ambiguities

Not just limited to scale. Given:

$$ p_{ij} \equiv M_i X_j $$

Can insert any global transform $H$

$$ p_{ij} \equiv M_i X_j = M_i H^{-1} H X_j $$

$H$ is a 3D homography / perspective transform / projective transform
Similarity/Affine/Perspective

Given:

Perspective

Lines

\[
\begin{bmatrix}
a & b & c \\
d & e & f \\
g & h & i \\
\end{bmatrix}
\]

Affine

+Parallelism

\[
\begin{bmatrix}
a & b & c \\
d & e & f \\
0 & 0 & 1 \\
\end{bmatrix}
\]

Similarity

+Angles

\[
\begin{bmatrix}
sR & t \\
0 & 1 \\
\end{bmatrix}
\]

3D: same idea, different dimensions

House image: A. Efros
Projective ambiguity

With no constraints on cameras matrices and scene, can only reconstruct up to a perspective ambiguity

\[ p_{ij} \equiv M_i X_j = M_i H^{-1} H X_j \]
Projective ambiguity

Slide credit: S. Lazebnik
Affine ambiguity

If we have constraints in the form of what lines are parallel, can reduce ambiguity to \textit{affine ambiguity}.

\[ p_{ij} \equiv M_i X_j = M_i H^{-1} H X_j \]

Slide credit: S. Lazebnik
Affine ambiguity

Slide credit: S. Lazebnik
Similarity ambiguity

If we have orthogonality constraints, get up to similarity transform. *Really the best we can do.* We get this if we have calibrated cameras.

\[ p_{ij} \equiv M_i X_j = M_i H^{-1} H X_j \]
Similarity ambiguity

Slide credit: S. Lazebnik
Affine structure from motion

We’ll do the math with affine / weak perspective cameras (math is much easier)
Recall: orthographic projection

Orthographic camera: things infinitely far away but you have an amazing camera

\[
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]
Field of view and focal length

wide-angle  standard  telephoto
Affine Camera

\[ M = \begin{bmatrix} A_{2D} & t_{2D} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_{3D} & t_{3D} \\ 0 & 1 \end{bmatrix} \]

3x3 Matrix 3x4 Ortho. 4x4 Matrix
Affine 2D Proj Affine 3D

Tedious math...

\[ M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]
Affine Camera

So what? Who cares?
Examine the projection

\[
\begin{bmatrix}
  u \\
v \\
1
\end{bmatrix} \equiv 
\begin{bmatrix}
a_{11} & a_{12} & a_{13} & b_1 \\
a_{21} & a_{22} & a_{23} & b_2 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]

Projection becomes linear mapping + translation and doesn’t involve homogeneous coordinates!

\[
\begin{bmatrix}
  u \\
v
\end{bmatrix} \equiv 
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23}
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} + 
\begin{bmatrix}
b_1 \\
b_2
\end{bmatrix}
\]

\textbf{b} is projection of origin. \textbf{Can anyone see why?}
Affine structure from motion

General structure from motion:
\[ p_{ij} \equiv M_i X_j \]

Assume M is affine camera:
\[ p_{ij} = A_i X_j + b_i \]

\( m \)n 2D points, \( m \) cameras, \( n \) 3D points up to arbitrary 3D affine (12 DOF)

Need:
\[ 2mn \geq 8m + 3n - 12 \]
\((m = 2): \ n \geq 4 \)
\((\text{for all } m!))\]
One simplifying trick

\[ p_{ij} = A_i X_j + b_i \]

Subtract off the average 2D point

\[ \hat{p}_{ij} = p_{ij} - \frac{1}{n} \sum_{k=1}^{n} p_{ik} = A_i X_j + b_i - \frac{1}{n} \sum_{k=1}^{n} A_i X_k + b_i \]

Gather terms involving \( A_i \), push out \( b_i \)

\[ \hat{p}_{ij} = A_i \left( X_j - \frac{1}{n} \sum_{k=1}^{n} X_k \right) + b_i - \frac{1}{n} \sum_{k=1}^{n} b_i \]

Set origin to mean of 3D points

\[ \hat{p}_{ij} = A_i X_j \]

Can do this entirely in terms of \( A! \)
Affine structure from motion

First, make data measurement matrix consisting of all the points stacked together

\[
\begin{bmatrix}
p_{11} & \cdots & p_{1n} \\
p_{m1} & \cdots & p_{mn}
\end{bmatrix}
\]

\[
\begin{bmatrix}
u_{11} & \cdots & \hat{v}_{1n} \\
\vdots & \ddots & \vdots
\end{bmatrix}
\]

\[
\begin{bmatrix}
u_{m1} & \cdots & \hat{v}_{mn}
\end{bmatrix}
\]


How big is this matrix?
Affine structure from motion

Then, write all the equations in one in terms of product of cameras and points.

\[
D = \begin{bmatrix}
\hat{p}_{11} & \cdots & \hat{p}_{1n} \\
\vdots & \ddots & \vdots \\
\hat{p}_{m1} & \cdots & \hat{p}_{mn}
\end{bmatrix} = \begin{bmatrix}
A_1 \\
\vdots \\
A_m
\end{bmatrix} \begin{bmatrix}
X_1 \\
\cdots \\
X_n
\end{bmatrix}
\]

What's the rank of D?

3!

Making Matrices Rank Deficient

Repeat of epipolar geometry class, but important enough to see twice. Given matrix \( M \):

\[
M \rightarrow U\Sigma V^T
\]

\( U_{m \times m}, V_{n \times n} \) rotation matrices

\( \Sigma_{m \times n} \) diagonal scaling matrix

\[
\Sigma = \begin{bmatrix}
\sigma_1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \sigma_m
\end{bmatrix}
\]

Keep only \( k \) biggest \( \sigma \); set others to 0

\[
\hat{M} \leftarrow U\hat{\Sigma}V^T
\]

Minimizes \( \| M - \hat{M} \|_F \) (sum of squares) subject to \( \text{rank}(\hat{M}) \leq k \)

See Eckart–Young–Mirsky theorem if you’re interested
Affine structure from motion

We’d like to take the measurements and convert them into \( M, S \)

\[
2m \quad D \quad=\quad M \times \quad S \quad 3
\]
Affine structure from motion

Do SVD (typically you don’t make full $U, \Sigma, V$)

\[ D_{2m \times n} = U_{n \times n} \Sigma_{n \times n} V_{n \times n}^T \]

Truncate to top 3 singular values

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

Remake of M. Hebert diagram
Affine structure from motion

Nearly there apart from this annoying $\Sigma_3$.

One solution (split $\Sigma_3$ in two):

$$ D = U_3 \times \Sigma_3 \times V_3^T $$

But remember that we can put $HH^{-1}$ in the middle.
Eliminating the affine ambiguity

Rows $a_i$ of $A_i$ give axes of camera. Can multiply each projection $A_i$ with $C$ to make $A_iC$ that satisfies:

$$a_1^T a_2 = 0$$
$$\|a_1\| = 1$$
$$\|a_2\| = 1$$

Gives 3 equations per camera, can set $A_iC$ to new camera, and $C^{-1}S$ to new points.

In general, a recipe for eliminating ambiguities

Remake of M. Hebert diagram
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, assume we can see all points in all views.

In reality, measurement matrix typically looks like this:

Possible solution: find dense blocks, solve in block, fuse.

In general, finding these dense blocks is NP-complete.

Figure Credit: S. Lazebnik
But cameras aren’t affine!

Want: m cameras $M_i$, n 3D points $X_j$
Given: mn 2D points $p_{ij}$

$$p_{ij} \equiv M_i X_j = M_i H^{-1} H X_j$$
When is this Possible?

Want: \( m \) cameras \( M_i \), \( n \) 3D points \( X_j \)

Given: \( mn \) 2D points \( p_{ij} \)

\[
p_{ij} \equiv M_i X_j = M_i H^{-1} H X_j
\]

Need \( 2mn \geq 11m + 3n - 15 \)

\((m = 2)\): \( n \geq 7 \)

\((m = 3)\): \( n \geq 6 \) (doesn’t get better after)

\((m=1)\): \( n \leq 4 \)
Two Camera Case

For two cameras, we need 7 points. Hmm. What else (in theory) requires 7 points?

Compute fundamental matrix $F$ and epipole $b$ s.t. $F^T b = 0$. Then:

$$M_1 = [I, 0]$$
$$M_2 = [-[b_x]F, b]$$

Remember: this is up to a projective ambiguity!
Incremental SFM

Key idea: incrementally add cameras, points

Note: numbers of points aren’t to scale.
Incremental SFM

Key idea: incrementally add cameras, points

1. Initialize motion $M_i = [R_i, t_i]$ with fundamental matrix
Incremental SFM

Key idea: incrementally add cameras, points

1. Initialize motion $M_i = [R_i, t_i]$ with fundamental matrix
2. Initialize structure $X_j$ with triangulation

How could we add another camera?

Note: numbers of points aren’t to scale.
Incremental SFM

Key idea: incrementally add cameras, points

1. Solve for camera matrix using visible, known points using calibration

Note: numbers of points aren’t to scale.
Incremental SFM

Key idea: incrementally add cameras, points

1. Solve for camera matrix using visible, known points using calibration

Now we can see the fourth point in two cameras.

Note: numbers of points aren’t to scale.
Incremental SFM

Key idea: incrementally add cameras, points

1. Solve for camera matrix using visible, known points using calibration

2. Solve for 3D coordinates of newly visible points using triangulation

Note: numbers of points aren’t to scale.
Incremental SFM

Key idea: incrementally add cameras, points

Big problem: don’t ever jointly consider all the 3D points and camera.

Leads to final step, called bundle adjustment.

Note: numbers of points aren’t to scale.

Remake of S. Lazebnik material
Bundle Adjustment

Do non-linear minimization over cameras $M_i$, points $X_j$ to minimize distance between observed points $p_{ij}$ and projections $M_iX_j$ when they’re visible.

$$\arg \min_{M_i, X_j} w_{ij} \ d((M_iX_j, p_{ij})^2$$

Visibility flag

Figure Credit: S. Lazebnik
Devil is in the details

High-level idea: \[ \arg \min_{M_i, X_j} w_{ij} d(M_i X_j, p_{ij})^2 \]

In practice:
• Have to initialize reasonably well
• Should minimize over K,R,t directly
• Problem is very sparse: \( w_{ij} \) almost always zero
• Need to integrate uncertainty information
• Probably want to use a system written by experts
Representative SFM pipeline

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

Detect SIFT features
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
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
In practice

- 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 $\mathbf{t}$) 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

Source: N. Snavely
The devil is in the details

• Degenerate configurations (homographies)
• Eliminating outliers
• Repetition and symmetry
The devil is in the details

- Degenerate configurations (homographies)
- Eliminating outliers
- Repetition and symmetry
- Multiple connected components
Next Class
Particular Challenge

Given: two RGB images with unknown relationship
Want: single, coherent reconstruction

Linyi Jin  Shengyi Qian  Andrew Owens
Particular Challenge

Given: two RGB images with unknown relationship
Want: single, coherent reconstruction

Output: Planes + Relative Camera Pose