3D Vision

Viktor Larsson, Marc Pollefeys

Spring 2019
# Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 18</td>
<td>Introduction</td>
</tr>
<tr>
<td>Feb 25</td>
<td>Geometry, Camera Model, Calibration</td>
</tr>
<tr>
<td>Mar 4</td>
<td>Features, Tracking / Matching</td>
</tr>
<tr>
<td>Mar 11</td>
<td><strong>Project Proposals by Students</strong></td>
</tr>
<tr>
<td>Mar 18</td>
<td><strong>Structure from Motion (SfM) + papers</strong></td>
</tr>
<tr>
<td>Mar 25</td>
<td>Dense Correspondence (stereo / optical flow) + papers</td>
</tr>
<tr>
<td>Apr 1</td>
<td>Bundle Adjustment &amp; SLAM + papers</td>
</tr>
<tr>
<td>Apr 8</td>
<td><strong>Student Midterm Presentations</strong></td>
</tr>
<tr>
<td>Apr 15</td>
<td>Multi-View Stereo &amp; Volumetric Modeling + papers</td>
</tr>
<tr>
<td>Apr 22</td>
<td>Easter break</td>
</tr>
<tr>
<td>Apr 29</td>
<td>3D Modeling with Depth Sensors + papers</td>
</tr>
<tr>
<td>May 6</td>
<td>3D Scene Understanding + papers</td>
</tr>
<tr>
<td>May 13</td>
<td>4D Video &amp; Dynamic Scenes + papers</td>
</tr>
<tr>
<td>May 20</td>
<td>papers</td>
</tr>
<tr>
<td>May 27</td>
<td><strong>Student Project Demo Day = Final Presentations</strong></td>
</tr>
</tbody>
</table>
3D Vision – Class 4

Structure from Motion

Chapter 7 in Szeliski’s Book
Chapter 9 in Hartley & Zisserman (online)
Tutorial chapters 3.2 and 4
Structure from Motion (SfM)

Rome dataset

74,394 images

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-Motion revisited, CVPR 2016
Sequential / Incremental SfM

1. Initialize Motion
2. Initialize Structure
3. Extend Motion
4. Extend Structure
Sequential / Incremental SfM

1. Initialize Motion
   - Two view reconstruction
     - Epipolar geometry
     - Fundamental matrix $F$
   - Essential matrix $E$
   - Computing $F$ and $E$

2. Initialize Structure

3. Extend Motion

4. Extend Structure
Epipolar Geometry

\[
R \quad t
\]

\[
R
\]
The Fundamental Matrix $F$

- Algebraic representation of epipolar geometry:
  - Projective mapping of points to lines:
    $$ x \mapsto l' \quad l' = Fx $$
  - $F$ has rank 2 since projection
- Correspondence condition:
  - Points $x, x'$ form correspondence $x \leftrightarrow x'$ if
    $$ x'^T l' = x'^T Fx = 0 $$
The Fundamental Matrix $\mathbf{F}$

- Geometric derivation:
Properties of $F$

$F$ is the unique 3x3 rank 2 matrix that satisfies $x'\,^T\!Fx = 0$ for all $x \leftrightarrow x'$

(i) **Transpose:** if $F$ is fundamental matrix for $(P,P')$, then $F^T$ is fundamental matrix for $(P',P)$

(ii) **Epipolar lines:** $l' = Fx$ & $l = F^T\!x'$

(iii) **Epipoles:** on all epipolar lines, thus $e'^T\!Fx = 0$, $\forall x$ $\Rightarrow e'^T\!F = 0$, similarly $Fe = 0$

(iv) $F$ has 7 d.o.f., i.e. 3x3 -1(homogeneous) -1(rank 2)

(v) $F$ is a correlation, projective mapping from a point $x$ to a line $l' = Fx$ (not a proper correlation, i.e. not invertible)
The Essential Matrix $E$

- Calibrated case: $P_1 = K_1[\mathbf{I}|\mathbf{0}]$, $P_2 = K_2[R|t]$
Properties of $\mathbf{E}$

$\mathbf{E}$ is an essential matrix iff two of its singular values are equal, third is 0

• Relationship to $\mathbf{F}$:

• Inherits $\mathbf{F}$’s properties (see previous slide)
Computation of $F$ & $E$

- Linear (8-point) ($F$ & $E$)
- Minimal (7-point) ($F$ & $E$)
- Calibrated (5-point) (only $E$)
Linear Solution (8-point)

- Basic epipolar equation: \( x^T F x = 0 \)
- Expand:
  \[
  x' x f_{11} + x' y f_{12} + x' f_{13} + y' x f_{21} + y' y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0
  \]
- Separate known and unknown variables:
  \[
  \begin{bmatrix}
  x' x, x' y, x', y' x, y', y', x, y, 1
  \end{bmatrix}
  \begin{bmatrix}
  f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}
  \end{bmatrix}^T = 0
  \]
  (data) (unknowns)
- Write as linear equation:
  \[
  \begin{bmatrix}
  x_1' x_1 & x_1' y_1 & x_1' & y_1' x_1 & y_1' y_1 & y_1' & x_1 & y_1 & 1
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
  x_n' x_n & x_n' y_n & x_n' & y_n' x_n & y_n' y_n & y_n' & x_n & y_n & 1
  \end{bmatrix}
  \begin{bmatrix}
  f
  \end{bmatrix} = 0
  \]
- 8 unknowns (up to scale): Use 8 points
Normalized 8-point Algorithm

\[
\begin{bmatrix}
    x_1 x_1' & y_1 x_1' & x_1' & x_1 y_1' & y_1 y_1' & y_1' & x_1 & y_1 & 1 \\
    x_2 x_2' & y_2 x_2' & x_2' & x_2 y_2' & y_2 y_2' & y_2' & x_2 & y_2 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    x_n x_n' & y_n x_n' & x_n' & x_n y_n' & y_n y_n' & y_n' & x_n & y_n & 1
\end{bmatrix}
\begin{bmatrix}
    f_{11} \\
    f_{12} \\
    f_{13} \\
    f_{21} \\
    f_{22} \\
    f_{23} \\
    f_{31} \\
    f_{32} \\
    f_{33}
\end{bmatrix} = 0
\]

\(~10000 \sim 10000 \sim 100 \sim 10000 \sim 10000 \sim 100 \sim 100 \sim 100 \sim 100 \sim 100 \sim 100 \sim 100 \sim 100 \sim 100 \sim 100 \sim 100 \sim 100 \sim 100 ~1\)

\(\text{Orders of magnitude difference between column of data matrix} \rightarrow \text{least-squares yields poor results}\)

- Normalize point coordinates prior to computing \(F\)
- Same as for the normalized DLT algorithm for homography estimation (see lecture 2)
The Singularity Constraint
The Singularity Constraint

\[ e'^T F = 0 \quad F e = 0 \quad \det F = 0 \quad \text{rank } F = 2 \]

- SVD from linearly computed F matrix (rank 3):
  \[
  F = U \begin{bmatrix}
  \sigma_1 \\
  \sigma_2 \\
  \sigma_3
  \end{bmatrix} V^T = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T + U_3 \sigma_3 V_3^T
  \]

- Compute closest rank-2 approximation: \( \min \|F - F'\|_F \)
  \[
  F' = U \begin{bmatrix}
  \sigma_1 \\
  \sigma_2 \\
  0
  \end{bmatrix} V^T = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T
  \]
The Singularity Constraint
Minimal Case: 7 Point Correspondences

• Setup linear system from 7 correspondences:

\[
\begin{bmatrix}
  x'_1 x_1 & x'_1 y_1 & x'_1 y'_1 x_1 & y'_1 y_1 & y'_1 x_1 & y_1 & 1 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  x'_7 x_7 & x'_7 y_7 & x'_7 y'_7 x_7 & y'_7 y_7 & y'_7 x_7 & y_7 & 1
\end{bmatrix} \mathbf{f} = 0
\]

• Resulting solution has 2D solution space

\[
A = U_{7 \times 7} \text{diag}(\sigma_1, \ldots, \sigma_7, 0, 0) V_{9 \times 9}^T \Rightarrow A[V_8 V_9] = 0_{9 \times 2}
\]

• \( F \) is linear combination of \( V_8 \) and \( V_9 \):

\[
x_i^T (F_1 + \lambda F_2)x_i = 0, \forall i = 1 \ldots 7
\]

• … but \( F_1 + \lambda F_2 \) not automatically rank 2
Minimal Case: 7 Point Correspondences

- Enforce rank-2 constraint from determinant:

\[
\det(F_1 + \lambda F_2) = a_3 \lambda^3 + a_2 \lambda^2 + a_1 \lambda + a_0 = 0
\]

- Cubic equation in \( \lambda \)
- Either 1 or 3 solutions
Calibrated Case: 5-point Solver

D. Nister, An Efficient Solution to the Five-Point Relative Pose Problem, CVPR 2003

- **Linear equations from 5 points**
  \[
  \begin{bmatrix}
  x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\
  x'_2x_2 & x'_2y_2 & x'_2 & y'_2x_2 & y'_2y_2 & y'_2 & x_2 & y_2 & 1 \\
  x'_3x_3 & x'_3y_3 & x'_3 & y'_3x_3 & y'_3y_3 & y'_3 & x_3 & y_3 & 1 \\
  x'_4x_4 & x'_4y_4 & x'_4 & y'_4x_4 & y'_4y_4 & y'_4 & x_4 & y_4 & 1 \\
  x'_5x_5 & x'_5y_5 & x'_5 & y'_5x_5 & y'_5y_5 & y'_5 & x_5 & y_5 & 1 \\
  \end{bmatrix}
  \begin{bmatrix}
  E_{11} \\
  E_{12} \\
  E_{13} \\
  E_{21} \\
  E_{22} \\
  E_{23} \\
  E_{31} \\
  E_{32} \\
  E_{33} \\
  \end{bmatrix} = 0
  \]

- **4D linear solution space:**
  \[
  E = xX + yY + zZ + wW \quad \text{scale does not matter, choose } w = 1
  \]

- **Insert into non-linear constraints**
  \[
  \det E = 0 \\
  2EE^T E - tr(EE^T) E = 0.
  \]
  \[
  \begin{cases}
  10 \text{ cubic polynomials}
  \end{cases}
  \]
  (assumes normalized coordinates)
Calibrated Case: 5-point Solver

D. Nister, An Efficient Solution to the Five-Point Relative Pose Problem, CVPR 2003

- Perform Gauss-Jordan elimination on polynomials

$$[n]$$ represents polynomial of degree $$n$$ in $$z$$

$$\langle e \rangle - z \langle f \rangle$$
$$\langle g \rangle - z \langle h \rangle$$
$$\langle i \rangle - z \langle j \rangle$$

$$\langle n \rangle \equiv \text{det}(B)$$
Automatic Computation of $F$

Step 1. Extract features
Step 2. Compute a set of potential matches
Step 3. Robust estimation of $F$ via RANSAC
Step 4. Compute $F$ based on all inliers
Step 5. Look for additional matches
Step 6. Refine $F$ based on all correct matches
RANdom SAmple Consensus (RANSAC)


• Problem: Estimate $F$ in presence of wrong matches

• RANSAC algorithm:
  • Repeat:
    • Randomly select minimal sample (5 or 7 points)
    • Compute hypothesis for $F$ from minimal sample
    • Verify hypothesis: Count inliers
  • Until probability of finding better solution $< \eta$

<table>
<thead>
<tr>
<th>#inliers</th>
<th>90%</th>
<th>80%</th>
<th>70%</th>
<th>60%</th>
<th>50%</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>#samples (5)</td>
<td>5</td>
<td>12</td>
<td>25</td>
<td>57</td>
<td>145</td>
<td>14k</td>
</tr>
<tr>
<td>#samples (7)</td>
<td>7</td>
<td>20</td>
<td>54</td>
<td>162</td>
<td>587</td>
<td>359k</td>
</tr>
</tbody>
</table>

$\eta=0.01\%$
Finding More Matches

• Restricted search around epipolar line (e.g. ±1.5 pixels)
• Relax disparity restriction (along epipolar line)
Want to Know More?

http://danielwedge.com/fmatrix/
Sequential / Incremental SfM

1. Initialize Motion
   - Initialize motion from $F$ or $E$
   - Triangulate structure from motion

2. Initialize Structure

3. Extend Motion

4. Extend Structure
Initial Motion and Structure Estimation (Calibrated Case)

- Recap Essential matrix: $E = [t]_x R$
- Motion for two cameras: $[I|0], [R|t]$
- Essential Matrix decomposition: $E = U \Sigma V^T$
  
  $$
  \Sigma = \begin{pmatrix}
  s & 0 & 0 \\
  0 & s & 0 \\
  0 & 0 & 0
  \end{pmatrix}
  \quad
  W = \begin{pmatrix}
  0 & -1 & 0 \\
  1 & 0 & 0 \\
  0 & 0 & 1
  \end{pmatrix}
  \quad
  W^{-1} = W^T = \begin{pmatrix}
  0 & 1 & 0 \\
  -1 & 0 & 0 \\
  0 & 0 & 1
  \end{pmatrix}
  $$

- Recover $E$ and $t$ as
  - $t = u_3$ or $t = -u_3$
  - $R = UWV^T$ or $R = UW^TV^T$
- Four solutions, but only one meaningful

(see Hartley and Zisserman, Sec.9.6)
Using the Cheirality Constraint

(see Hartley and Zisserman, Sec. 9.6)
Given: Motion, correspondence

Estimate 3D point via triangulation
Triangulation

- Backprojection

\[ \lambda x = Px \]

\[ P_3Xx = P_1X \]

\[ P_3Xy = P_2X \]

\[
\begin{bmatrix}
\lambda x \\
\lambda y \\
\lambda
\end{bmatrix}
= 
\begin{bmatrix}
P_1 \\
P_2 \\
P_3
\end{bmatrix}
X
\]

\[
\begin{bmatrix}
P_3x - P_1 \\
P_3y - P_2
\end{bmatrix}
X = 0
\]

- Triangulation

\[
\begin{bmatrix}
P_3x - P_1 \\
P_3y - P_2 \\
P_3'x' - P_1' \\
P_3'y' - P_2'
\end{bmatrix}
X = 0
\]

- Maximum Likelihood Triangulation (geometric error)

\[
\arg \min_X \sum_i \left( x_i - \lambda^{-1} P_i X \right)^2
\]
Optimal 3D Point in Epipolar Plane

- Given an epipolar plane, find best 3D point for \((m_1, m_2)\)

- Select closest points \((m_1', m_2')\) on epipolar lines
- Obtain 3D point through exact triangulation
- Guarantees minimal reprojection error (given this epipolar plane)
Optimal Two-View Triangulation

- **Non-iterative method:** (Hartley and Sturm, CVIU´97)
  - Determine optimal epipolar plane for reconstruction
  - \[ D(m_1, l_1(\alpha))^2 + D(m_2, l_2(\alpha))^2 \]  (polynomial of degree 6)
  - Reconstruct optimal point from selected epipolar plane
  - Note: Only works for two views

\[ \begin{align*}
  &l_1(\alpha) \quad m_1 \\
  &m_2 \quad l_2(\alpha)
\end{align*} \]

1 DOF
Sequential / Incremental SfM

1. Initialize Motion
   - Find camera with matches to previous images
   - Matches define 2D-3D correspondences
   - Estimate camera pose wrt. 3D structure

2. Initialize Structure

3. Extend Motion

4. Extend Structure
Pose Estimation from 2D-3D Matches

\[ x_i = P_i X(x_1, \ldots, x_{i-1}) \]

Compute \( P_{i+1} \) using robust approach (6-point RANSAC)
Extend and refine reconstruction
Compute P with 6-point RANSAC

- Generate hypothesis using 6 points

\[
\begin{bmatrix}
0^\top & -w_i X_i^\top & y_i X_i^\top \\
w_i X_i^\top & 0^\top & -x_i X_i^\top
\end{bmatrix}
\begin{pmatrix}
P^1 \\
P^2 \\
P^3
\end{pmatrix} = 0
\]

(two equations per point)

- Planar scenes are degenerate!

(similar DLT algorithm as see in 2\textsuperscript{nd} lecture for homographies)
3-Point-Perspective Pose – P3P (Calibrated Case)

All techniques yield 4th order polynomial

Incremental SfM

- **Initialize:**
  - Compute pairwise epipolar geometry
  - Find pair to initialize structure and motion

- **Repeat:**
  - For each additional view
    - Determine pose from structure
    - Extend structure
    - Refine structure and motion (bundle adjustment, see lecture 7)
Global SfM

- **Initialize:**
  - Compute pairwise epipolar geometry

- **Compute:**
  - Estimate all orientations
  - Estimate all positions
  - Triangulate structure
  - Refine structure and motion (bundle adjustment)

- **Pros:** More efficient, more accurate
- **Con:** Less robust
SfM Software

- **Colmap** (Johannes Schönberger)
  - Incremental SfM, very efficient, nice GUI, open source

- **VisualSFM** (Changchang Wu)
  - Incremental SfM, very efficient, GUI, binaries

- **Bundler** (Noah Snavely)
  - Incremental SfM, open source

- **OpenMVG** (Pierre Moulon)
  - Incremental and Global SfM, open source

- **Theia** (Chris Sweeney)
  - Incremental and Global SfM, very efficient, open source
Next week:
Dense Correspondence / Stereo