Structure from Motion

Introduction to Computer Vision
CSE 152
Lecture 9

Announcements

• HW2 assigned
• Midterm moved to Monday 5/13 – next class after HW2 due date.
Under perspective projection, the mapping from a plane to a plane is given by a linear transformation of homogeneous coordinates (called a projective transformation or homography).

Question: Where are the epipoles in the rectified image?
How is warping done? Forward Method

- Input: Source image: $I$ and
  Rectification matrix $H$
- For each corner $c_s$ of Source image in
  homogenous coordinates, compute
  $c_t = Hc_s$
- Compute smallest and largest $x$ and $y$
  of $c_t$’s, determine bounding box on
  target image, create target image $T$
  with size of bounding box.
- For each pixel coordinate $p_s$
  (homogenous) in the Source image,
  compute location in the Target image
  as $p_t = Hp_s$. Copy $I(p_s)$ to $T(p_t)$

Problem with Forward Method

- There’s no guarantee that every pixel
  in Target Image will be written to.
- If Target Image is larger than Source
  or Target is highly stretched, there
  may be missing points that appear as
  speckles or lines.
How is warping done? Backward method

- Input: Source image: I and rectification matrix H
- For each corner \( c_s \) of Source image in homogenous coordinates, compute \( c_t = Hc_s \)
- Compute smallest and largest x and y of \( c_t \)'s, determine bounding box on target image, create target image T with size of bounding box.
- For each pixel coordinate \( p_t \) (homogenous) in the Target, compute location in the Source as \( p_s = H^{-1}p_t \)
  - If \( p_s \) is within source image, copy \( I(p_s) \) to \( T(p_t) \)

---

Estimate 3D structure from images
How many views and how many points are needed to solve this?

Consider $M$ images of $N$ points, how many unknowns

1. Affix world coordinate system to location of first camera frame: $(M-1)*6$ unknowns for cameras
2. 3-D Structure: $3*N$ unknowns for points
3. Can only recover structure and motion up to scale factor (one fewer unknown

Total number of unknowns: $(M-1)*6+3*N-1$

Total number of measurements: $2*M*N$

Solution is possible when more measurements than unknowns: $(M-1)*6+3*N-1 \leq 2*M*N$

Some values of $N$ and $M$ satisfying this:

- $M = 2, \ N = 5$
- $M = 3, \ N = 4$

Two view structure from motion
The vectors $\overrightarrow{OP} \cdot \overrightarrow{OO'}$ and $\overrightarrow{O'P'}$ are coplanar

$$\overrightarrow{OP} : [\overrightarrow{OO'} \times \overrightarrow{O'P'}] = 0$$

$$\pmb{1} \cdot \left[ \begin{array}{c} \pmb{t}_2 \times (\pmb{R} \cdot \pmb{p}'') \end{array} \right] = 0$$

$\pmb{1} \pmb{p}^T \pmb{E} \cdot \pmb{p}' = 0 \text{ with } \pmb{E} = \left[ \left( \pmb{t}_2 \right)_x \right] \pmb{R}$

**Essential Matrix**
(Longuet-Higgins, 1981)

The Eight-Point Algorithm (Longuet-Higgins, 1981)

Input: 8 corresponding points in two images

$\pmb{1} \pmb{p}_i = \pmb{K}_1^{-1} \pmb{q}_i \quad \pmb{2} \pmb{p}'_i = \pmb{K}_2^{-1} \pmb{q}'_i$

$\pmb{1} \pmb{p}^T \pmb{E} \cdot \pmb{p}' = 0 \text{ with } \pmb{E} = \left[ \left( \pmb{t}_2 \right)_x \right] \pmb{R}$

$\begin{bmatrix} \pmb{E}_{11} & \pmb{E}_{12} & \pmb{E}_{13} \\ \pmb{E}_{21} & \pmb{E}_{22} & \pmb{E}_{23} \\ \pmb{E}_{31} & \pmb{E}_{32} & \pmb{E}_{33} \end{bmatrix} \begin{bmatrix} \pmb{u}' \\ \pmb{v}' \\ 1 \end{bmatrix} = 0$

Set $\pmb{E}_{33}$ to 1
- Use 8 points $(\pmb{u}_i, \pmb{v}_i)$, $i=1..8$

$\begin{bmatrix} \pmb{u}_1 \pmb{u}_1' \pmb{u}_1'' \\ \pmb{u}_2 \pmb{u}_2' \pmb{u}_2'' \\ \pmb{u}_3 \pmb{u}_3' \pmb{u}_3'' \\ \pmb{u}_4 \pmb{u}_4' \pmb{u}_4'' \\ \pmb{u}_5 \pmb{u}_5' \pmb{u}_5'' \\ \pmb{u}_6 \pmb{u}_6' \pmb{u}_6'' \\ \pmb{u}_7 \pmb{u}_7' \pmb{u}_7'' \\ \pmb{u}_8 \pmb{u}_8' \pmb{u}_8'' \end{bmatrix} \begin{bmatrix} \pmb{E}_{11} \\ \pmb{E}_{12} \\ \pmb{E}_{13} \\ \pmb{E}_{21} \\ \pmb{E}_{22} \\ \pmb{E}_{23} \\ \pmb{E}_{31} \\ \pmb{E}_{32} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

- Solve for $\pmb{E}_{11}$ to $\pmb{E}_{32}$
These are elements of the Essential Matrix
- Then solve for $\pmb{R}$, $\pmb{t}$

$\pmb{E} = \left[ \left( \pmb{t}_2 \right)_x \right] \pmb{R}$
Sketch of Two View SFM Algorithm

Input: Two images
1. Detect feature points in each image
2. Using intrinsic parameters from calibration, compute
   \[ p_{ij} = K_j^{-1} q_{ij} \]
1. Find 8 matching feature points (easier said than done)
2. Compute the Essential Matrix \( E \) using 8-point Algorithm
3. Compute \( R \) and \( t \) (recall that \( E = [t_x]R \) where \( [t_x] \) is a skew symmetric matrix). \( t \) can only be recovered up to a scale factor
4. Perform stereo matching using recovered epipolar geometry expressed via \( E \)
5. Reconstruct 3-D positions of corresponding points using \( R, T \) and \( p_{ij} \)

N-view structure from motion
Feature detection

Several images observe a scene from different viewpoints

Detect features using, for example, SIFT [Lowe, IJCV 2004]
Feature matching

Match features between each pair of images

Structure from motion

minimize $g(R, T, X)$

non-linear least squares
Bundle adjustment

- Minimize sum of squared reprojection errors:

\[ g(X, R, T) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| P(x_i, R_j, t_j) - \left[ u_{i,j} \right] \right\|^2 \]

where \( q_{ij} = (u_{ij}, v_{ij}) \) are the image coordinates and \( P(x_i, R_j, t_j) \) is the projection of 3D point \( x_i \) for a camera located at \( t_j \) with orientation \( R_j \).

- Optimized with non-linear least squares
- Levenberg-Marquardt is a popular choice

- Practical challenges?
  - Initialization
  - Outliers

Inliers vs. Outliers

- As you saw in HW1, not every match was correct when using SIFT descriptors.
- Squared errors metrics like \( \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| P(x_i, R_j, t_j) - \left[ u_{i,j} v_{i,j} \right] \right\|^2 \) highly penalize mismatches because they get squared.
- Inliers: Given a model with some assumed distribution, inliers are data points that fit the model.
- Outliers are points that do not fit the model.
- Example: line fitting

Inliers:

Outliers:
Given \( n \) points \((x_i, y_i)\), estimate parameters of line 
\[ ax_i + by_i - d = 0 \]
subject to the constraint that 
\[ a^2 + b^2 = 1 \]
Note: \( ax_i + by_i - d \) is distance from \((x_i, y_i)\) to line.

Cost Function: 
Sum of squared distances between each point and the line

Problem: minimize 
\[ E(a, b, d) = \sum_{i=1}^{n} (ax_i + by_i - d)^2 \]

with respect to \((a, b, d)\).

1. Minimize \( E \) with respect to \( d \):
\[ \frac{\partial E}{\partial d} = 0 \Rightarrow d = \frac{1}{n} \sum_{i=1}^{n} ax_i + by_i = a\bar{x} + b\bar{y} \]
Where \((\bar{x}, \bar{y})\) is the mean of the data points
Line fitting cont.

2. Substitute $d$ back into $E$

$$E = \sum_{i=1}^{n} [a(x_i - \bar{x}) + b(y_i - \bar{y})]^2 = \| \mathbf{u} \mathbf{n} \|^2$$

where $\mathbf{n} = (a \; b)^T$.

3. Minimize $E = |\mathbf{u} \mathbf{n}|^2 = \mathbf{u}^T \mathbf{S} \mathbf{u}$ with respect to $a, b$ subject to the constraint $\mathbf{n}^T \mathbf{n} = 1$. Note that $\mathbf{S}$ is given by

$$\mathbf{S} = \mathbf{u}^T \mathbf{u} = \begin{pmatrix} \sum_{i=1}^{n} x_i^2 - n \bar{x}^2 & \sum_{i=1}^{n} x_i y_i - n \bar{x} \bar{y} \\ \sum_{i=1}^{n} x_i y_i - n \bar{x} \bar{y} & \sum_{i=1}^{n} y_i^2 - n \bar{y}^2 \end{pmatrix}$$

And it’s a real, symmetric, positive definite

Line Fitting – Finished

4. This is a constrained optimization problem in $\mathbf{n}$. Solve with Lagrange multiplier

$$L(\mathbf{n}) = \mathbf{n}^T \mathbf{S} \mathbf{n} - \lambda (\mathbf{n}^T \mathbf{n} - 1)$$

Take partial derivative (gradient) w.r.t. $\mathbf{n}$ and set to 0.

$$\nabla L = 2 \mathbf{S} \mathbf{n} - 2 \lambda \mathbf{n} = 0$$

or

$$\mathbf{S} \mathbf{n} = \lambda \mathbf{n}$$

$\mathbf{n} = (a, b)$ is an Eigenvector of the symmetric matrix $\mathbf{S}$ (the one corresponding to the smallest Eigenvalue).

5. $d$ is computed from Step 1.
Motivation

• Estimating models in the presence of outliers
  – Lines
  – Transformations for mossaicing
  – Essential matrix
  – And other models

• Typically: keypoints in two images
Simpler Example

• Fitting a straight line (model has line parameters)

RANSAC Idea applied to line fitting

Problem: Given s points and threshold $\tau$, determine best fit line in presence of outliers
Repeat N times
  – Select two points at random
  – Determine line equation from the two points
  – Count number of points that are within distance $\tau$ from the line. This is called the “support” of the line and it’s the number of inliers
  – Line with the greatest support wins
Why will this work?

Iter 1
# of inliers: 2
# of outliers: 5

Iter 2
# of inliers: 6
# of outliers: 1

Why will this work?
RANSAC More Generally

- What do we need to apply RANSAC
  1. A parameterized model
  2. A way to estimate the model parameters from s data points \( \{x_1, \ldots, x_s\} \)
  3. Given the parameters of the model, a way to estimate the distance from a data point \( x_i \) to the model

RANSAC More Generally

**Objective**
Robust fit of model to data set \( S \) which contains outliers

**Algorithm**
REPEAT
  (i) Randomly select a sample of \( s \) data points from \( S \)
  (ii) Instantiate the model from this sample.
  (iii) Determine the set of data points \( S_i \) which are within a distance threshold \( t \) of the model. The set \( S_i \) is the **consensus set** of samples and defines the inliers of \( S \).
  (iv) \( S_{\text{largest}} = S_i \) if \( S_i \) is larger than \( S_{\text{largest}} \)
UNTIL (The size of \( S_i \) is greater than some threshold \( T \)) OR (There have been \( N \) trials)
The model is re-estimated using all the points in \( S_{\text{largest}} \)
How many samples?  
(What is N?)

Choose $N$ (number of samples) so that, with probability $p$, at least one random sample is free from outliers. e.g. $p=0.99$

$\left(1 - (1 - e)^s\right)^N = 1 - p$

$N = \log(1 - p) / \log(1 - (1 - e)^s)$

$e$: proportion of outliers
$s$: Number of points needed for the model

<table>
<thead>
<tr>
<th>proportion of outliers $e$</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>25%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>19</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>17</td>
<td>34</td>
<td>72</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>17</td>
<td>26</td>
<td>57</td>
<td>146</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>7</td>
<td>16</td>
<td>24</td>
<td>37</td>
<td>97</td>
<td>293</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>8</td>
<td>20</td>
<td>33</td>
<td>54</td>
<td>163</td>
<td>588</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>9</td>
<td>26</td>
<td>44</td>
<td>78</td>
<td>272</td>
<td>1177</td>
</tr>
</tbody>
</table>

Acceptable consensus set?

- Typically, terminate when inlier ratio reaches expected ratio of inliers

$$T = (1 - e)N$$

- Where
  - $N$ : Number of Samples
  - $E$ : proportion of outliers
  - $T$ : Size of consensus set when to stop
Using RANSAC to estimate the Essential Matrix

• What is the model?
  Essential Matrix (8 parameters)

• How many “points” are needed, and where do they come from?
  8 points in each image or 8 matched pairs (usually use this)

• What distance do we use to compute the consensus set?
  $L^2$ distance of points to epipolar line

• How often do outliers occur
  Usually not known.

Feature points extracted by a corner detector
Matched points by RANSAC

Putative matches of the feature points in both images are computed by using a correlation measure for points in one image with features in the other image. Only features within a small window are considered to limit computation time. Mutually best matches are retained. RANSAC is used to robustly determine F from these putative matches.

Epipolar Geometry from Matched Points