CS5760: Computer Vision

RANSAC

Reading

• Szeliski: Chapter 6.1
Announcements

• Vote for Project 1 artifacts by Friday, 11:59pm
• Project 2: code due on Monday, March 2 at 11:59pm
  – Report due Wednesday, March 4 at 11:59pm
• Midterm
  – Plan to release in-class next Wednesday, March 4
  – Due at the beginning of class, Monday, March 9
Outliers
Robustness

• Let’s consider the problem of linear regression

Problem: Fit a line to these datapoints

Least squares fit

• How can we fix this?
We need a better cost function…

• Suggestions?
Idea

• Given a hypothesized line
• Count the number of points that “agree” with the line
  – “Agree” = within a small distance of the line
  – I.e., the inliers to that line
• For all possible lines, select the one with the largest number of inliers
Counting inliers
Counting inliers

Inliers: 3
Counting inliers

Inliers: 20
How do we find the best line?

• Unlike least-squares, no simple closed-form solution

• Hypothesize-and-test
  – Try out many lines, keep the best one
  – Which lines?
Translations
RAndom SAmple Consensus

Select one match at random, count inliers
RAndon SAmple Consensus

Select another match at random, count *inliers*
Output the translation with the highest number of inliers
RANSAC

• Idea:
  – All the inliers will agree with each other on the translation vector; the (hopefully small) number of outliers will (hopefully) disagree with each other
    • RANSAC only has guarantees if there are < 50% outliers
  – “All good matches are alike; every bad match is bad in its own way.”
    – Tolstoy via Alyosha Efros
RANSAC

- **Inlier threshold** related to the amount of noise we expect in inliers
  - Often model noise as Gaussian w/ some standard deviation (e.g. 3 pixels)
- **Number of rounds** related to the percentage of outliers we expect, and the probability of success we’d like to guarantee
  - Suppose there are 20% outliers, and we want to find the correct answer with 99% probability
  - How many rounds do we need?
RANSAC: Another view

Set threshold so that, e.g., 95% of the Gaussian lies inside that radius.
RANSAC

- Back to linear regression
- How do we generate a hypothesis?
RANSAC

• Back to linear regression
• How do we generate a hypothesis?
RANSAC

• General version:
  1. Randomly choose $s$ samples
     • Typically $s =$ minimum sample size that lets you fit a model
  2. Fit a model (e.g., line) to those samples
  3. Count the number of inliers that approximately fit the model
  4. Repeat $N$ times
  5. Choose the model that has the largest set of inliers
How many rounds?

• If we have to choose \( s \) samples each time
  – with an outlier ratio \( e \)
  – and we want the right answer with probability \( p \)

\[
N \geq \frac{\log(1 - p)}{\log(1 - (1 - e)^s)}
\]

<table>
<thead>
<tr>
<th>( s )</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>

\( p = 0.99 \)

Source: M. Pollefeys
How big is $s$?

- For alignment, depends on the motion model
  - Here, each sample is a correspondence (pair of matching points)

<table>
<thead>
<tr>
<th>Name</th>
<th>Matrix</th>
<th># D.O.F.</th>
<th>Preserves:</th>
<th>Icon</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>$[I \mid t]_{2\times3}$</td>
<td>2</td>
<td>orientation + ⋯</td>
<td></td>
</tr>
<tr>
<td>rigid (Euclidean)</td>
<td>$[R \mid t]_{2\times3}$</td>
<td>3</td>
<td>lengths + ⋯</td>
<td></td>
</tr>
<tr>
<td>similarity</td>
<td>$[sR \mid t]_{2\times3}$</td>
<td>4</td>
<td>angles + ⋯</td>
<td></td>
</tr>
<tr>
<td>affine</td>
<td>$[A]_{2\times3}$</td>
<td>6</td>
<td>parallelism + ⋯</td>
<td></td>
</tr>
<tr>
<td>projective</td>
<td>$[\tilde{H}]_{3\times3}$</td>
<td>8</td>
<td>straight lines</td>
<td></td>
</tr>
</tbody>
</table>
RANSAC pros and cons

• Pros
  – Simple and general
  – Applicable to many different problems
  – Often works well in practice

• Cons
  – Parameters to tune
  – Sometimes too many iterations are required
  – Can fail for extremely low inlier ratios
  – We can often do better than brute-force sampling
Final step: least squares fit

Find average translation vector over all inliers
RANSAC

• An example of a “voting”-based fitting scheme
• Each hypothesis gets voted on by each data point, best hypothesis wins

• There are many other types of voting schemes
  – E.g., Hough transforms…
Panoramas

• Now we know how to create panoramas!

• Given two images:
  – Step 1: Detect features
  – Step 2: Match features
  – Step 3: Compute a homography using RANSAC
  – Step 4: Combine the images together (somehow)

• What if we have more than two images?
Can we use homographies to create a 360 panorama?

• In order to figure this out, we need to learn what a camera is
360 panorama
Questions?