Robust fitting
Camera calibration

• Given a set of correspondences between 3D world points and image points:
  • \((\mathbf{Q}, \mathbf{q})\), \(\ldots\), \((\mathbf{Q}_n, \mathbf{q}_n)\)

• Set up an optimization problem:
  \[
  \min_{\mathbf{p}} \| A\mathbf{p} \|_2 \\
  \text{such that} \\
  \| \mathbf{p} \|_2 = 1
  \]

• Solve for \(P\)
Homography estimation

• Given a set of correspondences between 3D world points on a plane and image points:
  • \( (\vec{Q}_1, \vec{q}_1), \ldots, (\vec{Q}_n, \vec{q}_n) \)

• Set up an optimization problem:
  \[
  \min_h \|Ah\|_2 \\
  \text{such that} \\
  \|h\|_2 = 1
  \]

• Solve for \( H \)
Fundamental matrix/ essential matrix estimation

• Given a set of correspondences between image points in two images:
  • \((\overrightarrow{q_1^1}, \overrightarrow{q_1^2}), ..., (\overrightarrow{q_n^1}, \overrightarrow{q_n^2})\)

• Set up an optimization problem:

\[
\min_f \|Af\|_2 \\
\text{such that} \\
\|f\|_2 = 1 \\
\text{rank}(F) = 2
\]

• Solve for \(F\)
What happens when correspondences are wrong?

• Not just noisy
  • Noise (e.g., ~1 pixel) can be handled because we are solving a minimization problem, rather than exactly satisfy equations

• Wrong correspondences can be way off.
Impact of incorrect correspondences

Correct:

Incorrect:
Outliers
Fitting in general

- Fitting: find the parameters of a model that best fit the data
- Other examples:
  - least squares linear regression
Least squares: linear regression

\[ y = mx + b \]

\((y_i, x_i)\)
Linear regression

\[
\min_{m,b} \sum_{i} (y_i - (mx_i + b))^2
\]
Robustness

Problem: Fit a line to these datapoints

Outliers!

Least squares fit
How do we find the best line in the presence of outliers?

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

• Hypothesize-and-test
  • Try out many lines, keep the best one
  • Which lines?
  • How to measure which is best?
Choosing lines to hypothesize

• Randomly choose lines?
  • Not optimal: highly unlikely to run into correct line by chance

• Idea: randomly sample data points from the dataset and fit line to them

• How many data points should we sample?
  • Any point we pick might be an outlier 😞
  • Want to maximize the chance that all sampled points are inliers → get the true line
  • Idea: sample the *minimum number of points necessary to get a line*
Choosing lines to hypothesize
Choosing lines to hypothesize
Choosing lines to hypothesize
Choosing lines to hypothesize
Measuring goodness of a line

• 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
RANSAC (Random Sample Consensus)

Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

**Repeat** 1-3 until the best model is found with high confidence
RANSAC

Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence
RANSAC

Algorithm:
1. Sample (randomly) the number of points required to fit the model (#=2)
2. Solve for model parameters using samples
3. Score by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Line fitting example
Algorithm:
1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

**Repeat** 1-3 until the best model is found with high confidence
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 marriages are alike; every bad marriage is bad in its own way.”
  – Tolstoy via Alyosha Efros
Translations
RAn dom SA mple Consensus

Select one match at random, count inliers
Select another match at random, count *inliers*
RAndom SAmple Consensus

Output the translation with the highest number of inliers
Final step: least squares fit

Find average translation vector over all inliers
RANSAC - hyperparameters

• **Inlier threshold** related to the amount of noise we expect in inliers
  • Often model noise as Gaussian with 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?
How many rounds?

• If we have to choose $k$ samples each time
  • with an inlier ratio $p$
  • and we want the right answer with probability $P$

<table>
<thead>
<tr>
<th>$p$</th>
<th>95%</th>
<th>90%</th>
<th>80%</th>
<th>75%</th>
<th>70%</th>
<th>60%</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
To ensure that the random sampling has a good chance of finding a true set of inliers, a sufficient number of trials $S$ must be tried. Let $p$ be the probability that any given correspondence is valid and $P$ be the total probability of success after $S$ trials. The likelihood in one trial that all $k$ random samples are inliers is $p^k$. Therefore, the likelihood that $S$ such trials will all fail is

$$1 - P = (1 - p^k)^S$$

and the required minimum number of trials is

$$S = \frac{\log(1 - P)}{\log(1 - p^k)}.$$ (6.30)

<table>
<thead>
<tr>
<th>proportion of inliers $p$</th>
<th>0.95</th>
<th>0.90</th>
<th>0.80</th>
<th>0.75</th>
<th>0.70</th>
<th>0.60</th>
<th>0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</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></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></td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>17</td>
<td>34</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>12</td>
<td>17</td>
<td>26</td>
<td>57</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>7</td>
<td>16</td>
<td>24</td>
<td>37</td>
<td>97</td>
<td>293</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>8</td>
<td>20</td>
<td>33</td>
<td>54</td>
<td>163</td>
<td>588</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>9</td>
<td>26</td>
<td>44</td>
<td>78</td>
<td>272</td>
<td>1177</td>
</tr>
</tbody>
</table>

$P = 0.99$
How big is $k$?

- 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>$[I \mid t]_{2×3}$</td>
<td>2</td>
<td>orientation + ⋯</td>
<td></td>
</tr>
<tr>
<td>rigid (Euclidean)</td>
<td>$[R \mid t]_{2×3}$</td>
<td>3</td>
<td>lengths + ⋯</td>
<td></td>
</tr>
<tr>
<td>similarity</td>
<td>$[sR \mid t]_{2×3}$</td>
<td>4</td>
<td>angles + ⋯</td>
<td></td>
</tr>
<tr>
<td>affine</td>
<td>$[A]_{2×3}$</td>
<td>6</td>
<td>parallelism + ⋯</td>
<td></td>
</tr>
<tr>
<td>projective</td>
<td>$[\tilde{H}]_{3×3}$</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
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...
RANSAC - Setup

• Given
  • A dataset $D = \{p_1, p_2, ..., p_N\}$
    • Example 1: Line fitting: $\{(x_1, y_1), ..., (x_n, y_n)\}$
    • Example 2: Homography fitting: $\{(\vec{Q}_1, \vec{q}_1), (\vec{Q}_2, \vec{q}_2), ..., (\vec{Q}_N, \vec{q}_N)\}$
  • A set of parameters $\theta$ that need to be fitted
    • Line fitting: $\theta = (m, b)$
    • Homography estimation $\theta = H, ||h|| = 1$
  • A cost function $C(p, \theta)$
    • Line fitting: $C((x, y), (m, b)) = ||y - (mx + b)||^2$
    • Homography estimation $C((\vec{Q}, \vec{q}), H) = E(H)$ (Reprojection error)
  • A minimum number needed $k$
    • Line fitting: 2
    • Homography estimation: 4
RANSAC - Setup

• Given
  • A dataset $D = \{ p_1, p_2, ..., p_N \}$
  • A set of parameters $\theta$ that need to be fitted
  • A cost function $C(p, \theta)$
  • $k$
  • $\theta^* = \min_{\theta} \sum_i C(p_i, \theta)$?
  • Problem: outliers
RANSAC - Algorithm

• Given: \( D = \{ p_1, p_2, ..., p_N \}, C(\theta, p), k \)
• \( \theta_{best} \leftarrow \phi, D_{inlier} \leftarrow \phi \)
• For \( i = 1, ..., S \)
  • Sample \( k \) points
  • Minimize \( C \) for these \( k \) points to get \( \theta_{hyp} \)
  • Compute the set of inliers: \( D_{hyp} = \{ p \in D : C(\theta_{hyp}, p) < \delta \} \)
  • If size of \( D_{hyp} \) is more than size of \( D_{inlier} \)
    • \( \theta_{best} \leftarrow \theta_{hyp} \)
    • \( D_{inlier} \leftarrow D_{hyp} \)
• Minimize \( \theta \) over \( D_{inlier} \)
RANSAC: how many iterations do we need?

- $p =$ inlier fraction
- $k =$ minimum number of data points
- $S =$ iter
- $P = (1 - (1 - p^k)^S)$