many slides are from Svetlana Lazebnik @ UNC
Fitting

• Choose a parametric model (e.g., line, circle, ellipse, ...) to represent a set of features
Three questions

• What model represents this set of features best?
• Which of several model instances get which features?
• How many model instances are there?
Fitting: Issues

Line detection:
• Noise
• Extraneous data: clutter (outliers), multiple lines
• Missing data: occlusions
Fitting: Issues

• If we know which points belong to the line, how do we find the “optimal” line parameters?
  – Least squares

• What if there are outliers?
  – Robust fitting, RANSAC

• What if there are many lines?
  – Voting methods: RANSAC, Hough transform

• What if we’re not even sure it’s a line?
  – Model selection
Last time: Hough transform

• An accumulation buffer for the parameters is created (i.e., parametric space)
• Select image points
  – Vote for all possible parameters that contain this point
  – This vote is cast in the accumulation buffer
• Scan the acc buffer to find the parameters with the highest vote
  – Hypothesis that we have a “shape” with these parameters in the image
Example

Peak corresponding to this line using "normal" parameters
Homework 4
Detection of lane markers
Least squares line fitting

- line: \( y_i = a x_i + b \)
- Find \( a \) and \( b \) to minimize

\[
E = \sum_{i=1}^{n} (y_i - ax_i - b)^2
\]

\[
\frac{\partial E}{\partial a} = 0, \quad \frac{\partial E}{\partial b} = 0
\]

\[
\begin{bmatrix}
  n & \sum_{i=1}^{n} x_i \\
  \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
= \begin{bmatrix}
\sum_{i=1}^{n} y_i \\
\sum_{i=1}^{n} x_i y_i
\end{bmatrix}
\]

Normal equations: \( X^T XB = X^T Y \)
Total least squares

• Line: \( ax + by + c = 0 \)

• Perpendicular distance between a point \((u, v)\) and the line is \(|au + bv + c|\) if \(a^2 + b^2 = 1\)

• Minimize the sum of perpendicular distances

\[
E = \sum_{i=1}^{n} (ax_i + by_i + c)^2
\]

subject to \(a^2 + b^2 = 1\)
Total least squares

\[ \frac{\partial E}{\partial c} = 0 \implies c = -a\bar{x} - b\bar{y} \]

\[ \implies E = \sum_{i=1}^{n} (ax_i - a\bar{x} + by_i - b\bar{y})^2 + \lambda(a^2 + b^2 - 1) \]

\[ \frac{\partial E}{\partial a} = 0, \frac{\partial E}{\partial b} = 0 \implies \begin{bmatrix} x^2 - \bar{x}\cdot\bar{x} & \bar{xy} - \bar{x}\cdot\bar{y} \\ \bar{xy} - \bar{x}\cdot\bar{y} & y^2 - \bar{y}\cdot\bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \lambda \begin{bmatrix} a \\ b \end{bmatrix} \]

\[ X_B = \lambda B \]

Solution is the eigenvector of \( X \) with the smallest eigenvalue
Least squares for general curves

- We would like to minimize the sum of squared geometric distances between the data points and the curve $(x_i, y_i)$

\[ d((x_i, y_i), C) \]
Least squares for conics

- Equation of a general conic:
  \[ C(a, x) = a \cdot x = ax^2 + bxy + cy^2 + dx + ey + f = 0, \]
  \[ a = [a, b, c, d, e, f], \]
  \[ x = [x^2, xy, y^2, x, y, 1] \]
- Minimizing the geometric distance is non-linear even for a conic
- **Algebraic distance:** \( C(a, x) \)
- Algebraic distance minimization by linear least squares:

\[
\begin{bmatrix}
  x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\
  x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1
\end{bmatrix} \begin{bmatrix}
  a \\
  b \\
  c \\
  d \\
  e \\
  f
\end{bmatrix} = 0
\]
Robustness
Robustness
Fitting: Issues

• If we know which points belong to the line, how do we find the “optimal” line parameters?
  – Least squares

• What if there are outliers?
  – Robust fitting, RANSAC

• What if there are many lines?
  – Voting methods: RANSAC, Hough transform

• What if we’re not even sure it’s a line?
  – Model selection
Robust fitting

- General approach: minimize \( \sum_i \rho(r_i(x_i, \theta); \sigma) \)

\( r_i(x_i, \theta) \) – residual of ith point w.r.t. model parameters \( \theta \)

\( \rho \) – robust function with scale parameter \( \sigma \)

The robust function \( \rho \) behaves like squared distance for small values of the residual \( r \) but saturates for larger values of \( r \)

\[ \rho = \frac{r^2}{r^2 + \sigma^2} \]
Choosing the scale: Just right
Choosing the scale: Too small
RANSAC

• Robust fitting can deal with a few outliers - what if we have very many?

• **Random sample consensus (RANSAC):** Very general framework for model fitting in the presence of outliers

• Outline
  – Choose a small subset of points uniformly at random
  – Fit a model to that subset
  – Find all remaining points that are “close” to the model and reject the rest as outliers
  – Do this many times and choose the best model

RANSAC for line fitting

Repeat $N$ times:

- Draw $s$ points uniformly at random
- Fit line to these $s$ points
- Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than $t$)
- If there are $d$ or more inliers, accept the line and refit using all inliers
Choosing the parameters

• Initial number of points $s$
  – Typically minimum number needed to fit the model

• Distance threshold $t$
  – Choose $t$ so probability for inlier is $p$ (e.g. 0.95)
  – Zero-mean Gaussian noise with std. dev. $\sigma$: $t^2 = 3.84\sigma^2$

• Number of samples $N$
  – Choose $N$ so that, with probability $p$, at least one random sample is free from outliers (e.g. $p=0.99$) (outlier ratio: $e$)

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

\[
N = \log(1 - p) / \log\left(1-(1-e)^s\right)
\]

<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>
Adaptively determining the number of samples

• Outlier ratio $e$ is often unknown a priori, so pick worst case, e.g. 50%, and adapt if more inliers are found, e.g. 80% would yield $e=0.2$

• Adaptive procedure:
  – $N=\infty$, sample_count = 0
  – While $N>\text{sample\_count}$
    • Choose a sample and count the number of inliers
    • Set $e = 1 - (\text{number of inliers})/(\text{total number of points})$
    • Recompute $N$ from $e$:
      \[
      N = \frac{\log(1 - p)}{\log\left(1 - (1 - e)^s\right)}
      \]
    • Increment the sample_count by 1
RANSAC pros and cons

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

• **Cons**
  – Lots of parameters to tune
  – Can’t always get a good initialization of the model based on the minimum number of samples
  – Sometimes too many iterations are required
  – Can fail for extremely low inlier ratios
  – We can often do better than brute-force sampling