Fitting methods

- Problem formulation
- Least square methods
- RANSAC
- Hough transforms
- Multi-model fitting
- Fitting helps matching!

Reading:  [HZ] Chapters: 4, 11
[F] Chapters: 16

Some slides of this lectures are courtesy of profs. S. Lazebnik & K. Grauman
Fitting

Goals:
- Choose a parametric model to fit a certain quantity from data
- Estimate model parameters
- Lines
- Curves
- Homographic transformation
- Fundamental matrix
- Shape model
Example: fitting lines
(for computing vanishing points)
Example: Estimating an homographic transformation
Example: Estimating F
Example: fitting a 2D shape template
Example: fitting a 3D object model
Fitting

Critical issues:
- noisy data
- outliers
- missing data
Critical issues: noisy data
Critical issues: noisy data (intra-class variability)
Critical issues: outliers
Critical issues: missing data (occlusions)
Fitting

Goal: Choose a parametric model to fit a certain quantity from data

Techniques:
- Least square methods
- RANSAC
- Hough transform
- EM (Expectation Maximization) [not covered]
Least squares methods
- fitting a line -

- Data: \((x_1, y_1), \ldots, (x_n, y_n)\)
- Line equation: \(y_i = mx_i + b\)
- Find \((m, b)\) to minimize

\[
E = \sum_{i=1}^{n} (y_i - mx_i - b)^2
\]
Least squares methods
- fitting a line -

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

\[
E = \sum_{i=1}^{n} \left( y_i - \begin{bmatrix} x_i \\ 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} \right)^2 = \left\| \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} \right\|^2 = \| Y - XB \|^2
\]

= \( Y - XB \)^T (Y - XB) = Y^T Y - 2(XB)^T Y + (XB)^T (XB)

Find \((m, b)\) that minimize \(E\)

\[
\frac{dE}{dB} = 2 X^T XB - 2 X^T Y = 0
\]

\[
X^T XB = X^T Y
\]

Normal equation

\[
B = \left( X^T X \right)^{-1} X^T Y
\]
Least squares methods
- fitting a line -

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

\[ B = (X^T X)^{-1} X^T Y \]

Limitations

- Fails completely for vertical lines
Least squares methods
- fitting a line -

- Distance between point $(x_n, y_n)$ and line $ax+by=d$

- Find $(a, b, d)$ to minimize the sum of squared perpendicular distances

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

\[ \begin{bmatrix} U \mid N \end{bmatrix} = 0 \]

- data  model parameters
Least squares methods
- fitting a line -

\[ A \mathbf{h} = 0 \]

Minimize \[ \| A \mathbf{h} \| \] subject to \[ \| \mathbf{h} \| = 1 \]

\[ A = UDV^T \]

\[ \mathbf{h} = \text{last column of} \ V \]
Least squares methods
- fitting an homography -

\[
\begin{bmatrix}
U \\ N
\end{bmatrix} = 0
\]

data \hspace{1cm} model parameters
Least squares: Robustness to noise
Least squares: Robustness to noise

Problem: squared error heavily penalizes outliers
Critical issues: outliers

CONCLUSION: Least square is not robust w.r.t. outliers
Least squares: Robust estimators

• General approach:
  
  $$\sum_i \rho(u_i(X_i, \theta); \sigma)$$
  
  where
  
  $$u_i(X_i, \theta) = \text{residual of } i^{\text{th}} \text{ point w.r.t. model parameters } \theta$$

  $$\rho = \text{robust function with scale parameter } \sigma$$

  $$u = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

• $$u_i(X_i, \theta) = \text{residual of } i^{\text{th}} \text{ point w.r.t. model parameters } \theta$$

The robust function $$\rho$$

• Favors a configuration with small residuals
• Penalizes large residuals
Least squares: Robust estimators

Good scale parameter $\sigma$

The effect of the outlier is eliminated
Least squares: Robust estimators

- Bad scale parameter $\sigma$ (too small!)
- Fits only locally
- Sensitive to initial condition
Least squares: Robust estimators

Bad scale parameter $\sigma$ (too large!)
Same as standard LSQ

• CONCLUSION: Robust estimator useful if prior info about the distribution of points is known

• Robust fitting is a nonlinear optimization problem (iterative solution)
• Least squares solution provides good initial condition
Fitting

Goal: Choose a parametric model to fit a certain quantity from data

Techniques:
• Least square methods
• RANSAC
• Hough transform
Basic philosophy
(voting scheme)

• Data elements are used to vote for one (or multiple) models

• Robust to outliers and missing data

• Assumption 1: Noise features will not vote consistently for any single model (“few” outliers)

• Assumption 2: there are enough features to agree on a good model (“few” missing data)
RANSAC

(RANDom SAMple Consensus):
Learning technique to estimate parameters of a model by random sampling of observed data

Fischler & Bolles in ‘81.

\[ \pi : I \rightarrow \{P, O\} \]

such that:

\[ f(P, \beta) < \delta \]

\[ \min_{\pi} |O| \]

\[ f(P, \beta) = \left\| \beta - (P^T P)^{-1} P^T \right\| \]

Model parameters
RANSAC

Sample set = set of points in 2D

Algorithm:
1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
Repeat 1-3 until model with the most inliers over all samples is found
Algorithm:

1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
Repeat 1-3 until model with the most inliers over all samples is found

Sample set = set of points in 2D
Algorithm:

1. Select random sample of minimum required size to fit model [?]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found
Algorithm:

1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found
RANSAC

(RANdom SAmple Consensus):
Fischler & Bolles in ‘81.

Algorithm:
1. Select random sample of minimum required size to fit model [?]
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
Repeat 1-3 until model with the most inliers over all samples is found
How many samples?

- **Number of samples** $N$
  - $p =$ probability at least one random sample is free from outliers (e.g. $p=0.99$)
  - $e =$ outlier ratio
  - $s =$ minimum number needed to fit the model

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

<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>
Estimating H by RANSAC

Algorithm:

1. Select a random sample of minimum required size
2. Compute a putative model from these
3. Compute the set of inliers to this model from whole sample space

Repeat 1-3 until model with the most inliers over all samples is found

Sample set = set of matches between 2 images

- H \rightarrow 8 \text{ DOF}
- Need 4 correspondences
Estimating $F$ by RANSAC

1. Select a random sample of minimum required size [?]
2. Compute a putative model from these
3. Compute the set of inliers to this model from whole sample space

Repeat 1-3 until model with the most inliers over all samples is found

Sample set = set of matches between 2 images

Algorithm:
1. Select a random sample of minimum required size [?]
2. Compute a putative model from these
3. Compute the set of inliers to this model from whole sample space

Repeat 1-3 until model with the most inliers over all samples is found

$F \rightarrow 7$ DOF
• Need 7 (8) correspondences
RANSAC - conclusions

Good:

• Simple and easily implementable
• Successful in different contexts

Bad:

• Many parameters to tune
• Trade-off accuracy-vs-time
• Cannot be used if ratio inliers/outliers is too small
Fitting

Goal: Choose a parametric model to fit a certain quantity from data

Techniques:
• Least square methods
• RANSAC
• Hough transform
Hough transform


Given a set of points, find the curve or line that explains the data points best
Hough transform


Given a set of points, find the curve or line that explains the data points best

\[ y = m \cdot x + n \]

Hough space

\[ y_1 = m \cdot x_1 + n \]
Hough transform


Issue: parameter space \([m,n]\) is unbounded...

Use a polar representation for the parameter space

\[
x \cos \theta + y \sin \theta = \rho
\]
Hough transform - experiments

features

votes
Hough transform - experiments

IDEA: introduce a grid and count intersection points in each cell.

Issue: Grid size needs to be adjusted...
Hough transform - experiments

Issue: spurious peaks due to uniform noise
Hough transform - conclusions

Good:

• All points are processed independently, so can cope with occlusion/outliers
• Some robustness to noise: noise points unlikely to contribute consistently to any single bin

Bad:

• Spurious peaks due to uniform noise
• Trade-off noise-grid size (hard to find sweet point)
Hough transform - experiments

Courtesy of TKK Automation Technology Laboratory
Generalized Hough transform
[more on forthcoming lectures]

D. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, Pattern Recognition 13(2), 1981

- Identify a shape model by measuring the location of its parts and shape centroid
- Measurements: orientation theta, location of p
- Each measurement casts a vote in the Hough space: $p + r(\theta)$
Generalized Hough transform

B. Leibe, A. Leonardis, and B. Schiele, *Combined Object Categorization and Segmentation with an Implicit Shape Model*, ECCV Workshop on Statistical Learning in Computer Vision 2004
EECS 442 – Computer vision

Fitting methods

- Problem formulation
- Least square methods
- RANSAC
- Hough transforms
- Multi-model fitting
- Fitting helps matching!

Reading: [HZ] Chapters: 4, 11
[FP] Chapters: 16

Some slides of this lectures are courtesy of profs. S. Lazebnik & K. Grauman
Fitting multiple models

- Incremental fitting
- E.M. (probabilistic fitting)
- Hough transform
Incremental line fitting

Scan data point sequentially (using locality constraints)

Perform following loop:

1. Select N point and fit line to N points
2. Compute residual $R_N$
3. Add a new point, re-fit line and re-compute $R_{N+1}$
4. Continue while line fitting residual is small enough,

- When residual exceeds a threshold, start fitting new model (line)
Hough transform

Same cons and pros as before…
Fitting helps matching!

Feature are matched (for instance, based on correlation)
Idea:
- Fitting an homography $H$ (by RANSAC) mapping features from images 1 to 2
- Bad matches will be labeled as outliers (hence rejected)!

Matches bases on appearance only
Red: good matches
Green: bad matches

Fitting helps matching!
Fitting helps matching!
Recognising Panoramas

Fitting helps matching!

Images courtesy of Brandon Lloyd
Next lecture:
Feature detectors part II
Least squares methods
- fitting a line -

\[ Ax = b \]

- More equations than unknowns

- Look for solution which minimizes \( ||Ax-b|| = (Ax-b)^T(Ax-b) \)

- Solve \( \frac{\partial (Ax - b)^T (Ax - b)}{\partial x_i} = 0 \)

- LS solution

\[ x = (A^T A)^{-1} A^T b \]
Least squares methods
- fitting a line -

Solving  \[ x = (A^t A)^{-1} A^t b \]

\[ A^+ = (A^t A)^{-1} A^t \quad = \text{pseudo-inverse of } A \]

\[ A = U \Sigma V^t \quad = \text{SVD decomposition of } A \]

\[ A^{-1} = V \Sigma^{-1} U \]

\[ A^+ = V \Sigma^+ U \]

with  \( \Sigma^+ \) equal to  \( \Sigma^{-1} \) for all nonzero singular values and zero otherwise
Least squares methods
- fitting an homography -

\[ h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' - x' = 0 \]
\[ h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' - y' = 0 \]

From \( n \geq 4 \) corresponding points:

\[ A \mathbf{h} = 0 \]

\[
\begin{pmatrix}
  x_1 & y_1 & 1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & x_1 & y_1 & 1 & -x_1x' & -y_1x' & -x'_1 \\
  x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x' & -y_2x' & -x'_2 \\
  0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y' & -y_2y' & -y'_2 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n & -x'_n \\
  0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n & -y'_n \\
\end{pmatrix}
\begin{pmatrix}
  h_{1,1} \\
  h_{1,2} \\
  \vdots \\
  h_{3,3}
\end{pmatrix} = 0
\]