Lecture 8
Fitting and Matching

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

Reading:
[HZ] Chapter 4 “Estimation – 2D projective transformation”
[FP] Chapter 10 “Grouping and model fitting”

Some slides of this lecture 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 transformations
- Fundamental matrices
- Shape models
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, matching and recognition are interconnected problems
Fitting

Critical issues:
- noisy data
- outliers
- missing data
- Intra-class variation
Critical issues: noisy data
Critical issues: outliers

What's H?
Critical issues: outliers

What’s H? 😞
Critical issues: missing data (occlusions)
Critical issues: noisy data (intra-class variability)
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: \((x_1, y_1), \ldots, (x_n, y_n)\)

- Model of the line:
  \[ y_i - mx_i - b = 0 \]  \[\text{[Eq. 1]}\]

- Parameters: \(m, b\)

- Find \((m, b)\) to minimize fitting error (residual):
  \[ E = \sum_{i=1}^{n} (y_i - mx_i - b)^2 \]  \[\text{[Eq. 2]}\]
Least squares methods
- fitting a line -

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

\[ E = \sum_{i=1}^{n} \left( y_i - \begin{bmatrix} x_i & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} \right)^2 \]
\[ = \left\| \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} \right\|^2 \]
\[ = \| Y - Xh \|^2 \]  \[ \text{[Eq. 3]} \]

\[ = (Y - Xh)^T (Y - Xh) = Y^T Y - 2(Xh)^T Y + (Xh)^T (Xh) \]  \[ \text{[Eq. 4]} \]

Find \( h = [m, b]^T \) that minimizes \( E \)
\[ \frac{dE}{dh} = -2X^T Y + 2X^T Xh = 0 \]  \[ \text{[Eq. 5]} \]

\[ X^T Xh = X^T Y \]  \[ \text{[Eq. 7]} \]

Normal equation
\[ h = \left( X^T X \right)^{-1} X^T Y \]  \[ \text{[Eq. 6]} \]
Least squares methods
- fitting a line -

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

\[
h = \left( X^T X \right)^{-1} X^T Y
\]

\[
h = \begin{bmatrix} m \\ b \end{bmatrix}
\]

[Eq. 6]

Issues?

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

- Distance between point \((x_i, y_i, 1)\) and line \((a, b, d)\)

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

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

\[
A h = 0
\]

Data  Model parameters
Least squares methods
- fitting a line -

\[ Ah = 0 \quad \text{A is rank deficient} \]

Minimize \( \| Ah \| \) \quad \text{subject to} \quad \| h \| = 1

\[ A = UDV^T \]

\( h = \text{last column of } V \)

See [HZ], sec. A5.3 - page 593
Least squares methods  
- fitting an homography -

\[
A h = 0 \quad \text{[Eq. 10]}
\]

See HZ  
- Sec 4.1 for details (DLT algorithm)  
- Sec 4.1.2 (or APPENDIX)
Least squares: Robustness to noise
Least squares: Robustness to noise

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

Instead of minimizing \[ E = \sum_{i=1}^{n} (ax_i + by_i - d)^2 \] \[ \text{[Eq. 8]} \]

We minimize \[ E = \sum \rho(u_i; \sigma) \] \[ \text{[Eq. 11]} \]

\[ u_i = ax_i + by_i - d \]

- \( u_i \) = error (residual) of \( i^{\text{th}} \) point w.r.t. model parameters \( h = (a,b,d) \)

- \( \rho = \) robust function of \( u_i \) with scale parameter \( \sigma \)

\( \rho(\sigma; u) = \frac{u^2}{\sigma^2 + u^2} \) \[ \text{[Eq. 12]} \]

**Robust function \( \rho \):**
- When \( u \) is large, \( \rho \) saturates to 1
- When \( u \) is small, \( \rho \) is a function of \( u^2 \)

In conclusion:
- Favors a configuration with small residuals
- Penalizes large residuals
Least squares: Robust estimators

Instead of minimizing $E = \sum_{i=1}^{n} (ax_i + by_i - d)^2$ [Eq. 8]

We minimize $E = \sum_i \rho(u_i ; \sigma)$ [Eq. 11] $u_i = ax_i + by_i - d$

- $u_i =$ error (residual) of $i^{th}$ point w.r.t. model parameters $h = (a,b,d)$
- $\rho =$ robust function of $u_i$ with scale parameter $\sigma$

\begin{align*}
\rho(u ; \sigma) &= \frac{u^2}{\sigma^2 + u^2} \\
\end{align*}

\begin{itemize}
  \item Small sigma $\rightarrow$ highly penalize large residuals
  \item Large sigma $\rightarrow$ mildly penalize large residual (like LSQR)
\end{itemize}
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
Least squares: Robust estimators

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

**CONCLUSION:** Robust estimator useful if prior info about the distribution of points is known
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: Noisy data points will not vote consistently for any single model (“few” outliers)

• Assumption 2: There are enough data points to agree on a good model (“few” missing data)
Example:
Line fitting

- Enough “good” data points supporting the line model in presence of noise
- “Few” outliers compared to the “good” data points – these few outliers won’t “consistently” vote for a line model
RANSAC

(RANdom SAmple Consensus) :

Fischler & Bolles in ‘81.

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

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

such that:

\[ r(I, h) < \delta, \quad \forall I \in I \]

\[ r(I, h) = \text{residual} = \sum_{i=1}^{n} (ax_i + by_i + d)^2 \]

[Eq. 12]
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

\[ P = \text{Sample set} = \text{set of points in 2D} \]
RANSAC

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

\( P = \text{Sample set} = \text{set of points in 2D} \)
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
RANSAC

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 \[\delta\]
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?

• Computationally unnecessary (and infeasible) to explore the entire sample space

• **N samples are sufficient**
  • **N** = number of samples required to ensure, with a probability **p**, that at least one random sample produces an inlier set that is free from “real” outliers

• Function of **s** and **e**:
  - **e** = outlier ratio
  - **s** = minimum number of data points needed to fit the model

• Usually, **p**=0.99
• Here a random sample is given by two green points
• The estimated inlier set is given by the green+blue points
• How many “real” outliers we have here? 2

\[ e = \text{outlier ratio} = \frac{6}{20} = 30\% \]
\[ s = 2 \]
Random sample is given by two green points
The estimated inlier set is given by the green+blue points
How many "real" outliers we have here?

\[ e = \text{outlier ratio} = \frac{6}{20} = 30\% \]
\[ s = 2 \]
N is the number of times we need to sample my data (and thus repeat the steps 1-3 in the previous slides) before I find the configuration above with probability $p$. Again this is function of $e$ and $s$ as well.

$e = \text{outlier ratio is } \frac{6}{20} = 30\%$

$s = 2$
How many samples?

• Number $N$ of samples required to ensure, with a probability $p$, that at least one random sample produces an inlier set that is free from “real” outliers for a given $s$ and $e$.

• E.g., $p=0.99$

$$N = \log(1 - p) / \log(1 - (1 - e)^s)$$  \[\text{Eq. 13}\]

<table>
<thead>
<tr>
<th>$e$</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>$s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>17</td>
<td>34</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>17</td>
<td>26</td>
<td>57</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>16</td>
<td>24</td>
<td>37</td>
<td>97</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>20</td>
<td>33</td>
<td>54</td>
<td>163</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>5</td>
<td>9</td>
<td>26</td>
<td>44</td>
<td>78</td>
<td>272</td>
</tr>
</tbody>
</table>

Note: this table assumes “negligible” measurement noise

e = outlier ratio
$s$ = minimum number needed to fit the model
Estimating $H$ by RANSAC

- $H \rightarrow 8$ DOF
- Need 4 correspondences

$P = \text{Sample set} = \text{set of matches between 2 images}$

**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
Estimating F 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

P = Sample set = set of matches between 2 images

- F \rightarrow 7 DOF
- Need 7 (8) correspondences

Outlier matches
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

Hough transform


Given a set of points, find the line parameterized by $m, n$ that explains the data points best: that is, $m = m'$ and $n = n'$
Hough transform


Given a set of points, find the line parameterized by $m,n$ that explains the data points best: that is, $m = m'$ and $n = n'$

Original space where the data points are

Hough space defined by the parameters of the model we want to fit (i.e., $m, n$)
Hough transform


Any Issue? The parameter space \([m,n]\) is unbounded...
Hough transform


**Any Issue?** The parameter space \([m,n]\) is unbounded...

**Use a polar representation for the parameter space**

\[
x \cos \theta + y \sin \theta = \rho
\]  

[Eq. 13]
Hough transform - experiments

Original space

Hough space
How to compute the intersection point? In presence of noise!

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

Bad:

• Spurious peaks due to uniform noise
• Trade-off noise-grid size (hard to find sweet point)
• Doesn’t handle well high dimensional models
Applications – lane detection

Courtesy of Minchae Lee
Applications – computing vanishing points
Generalized Hough transform
[more on forthcoming lectures]

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

• Parameterize a shape by measuring the location of its parts and shape centroid
• Given a set of measurements, cast a vote in the Hough (parameter) space

• Used in object recognition! (the implicit shape model)

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
Lecture 9
Fitting and Matching

- Problem formulation
- Least square methods
- RANSAC
- Hough transforms
- Multi-model fitting
- Fitting helps matching!
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...
Lecture 9

Fitting and Matching

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

Features 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 based on appearance only
Green: good matches
Red: bad matches

Fitting helps matching!
Fitting helps matching!
Next lecture:
Feature detectors and descriptors
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** \( x = (A^t A)^{-1} A^t b \)

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

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

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

\[
A^+ = V \sum^+ U^T
\]

with \( \sum^+ \) equal to \( \sum^{-1} \) 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:

\[
\begin{bmatrix}
    x_1 & y_1 & 1 & 0 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\
    0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 & -y'_1 \\
    x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 & -x'_2 \\
    0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 & -y'_2 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n & -x'_n \\
    0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n & -y'_n \\
\end{bmatrix}
\begin{bmatrix}
    h_{1,1} \\
    h_{1,2} \\
    \vdots \\
    h_{3,3}
\end{bmatrix} = 0
\]
Hough transform - experiments

Issue: spurious peaks due to uniform noise
Hough transform
Fitting helps matching!

Images courtesy of Brandon Lloyd