Model Fitting
Introduction to Computer Vision
CSE 152
Lecture 11

What to do with edges?
• Segment linked edge chains into curve features (e.g., line segments).
• Group unlinked or unrelated edges into lines (or curves in general).
• Accurately fitting parametric curves (e.g., lines) to grouped edge points.

Finding lines in an image
\[ y = mx + b \]
image space \[ \rightarrow \]
Hough space \[ \rightarrow \]
\[ (m, b) \]
Connection between image \((x, y)\) and Hough \((m, b)\) spaces
• A line in the image corresponds to a point in Hough space

Hough Transform
[ Patented 1962 ]

Finding lines in an image
\[ b = -x_0m + y_0 \]
\( (x_0, y_0) \)
image space \[ \rightarrow \]
Hough space \[ \rightarrow \]
\[ (m, b) \]
Connection between image \((x, y)\) and Hough \((m, b)\) spaces
• A line in the image corresponds to a point in Hough space
• What does a point \((x_0, y_0)\) in the image space map to?
The equation of any line passing through \((x_0, y_0)\) has form
\[ y = mx + b \]
This is a line in Hough space

Announcements
• Graded homework 1 was returned last week
• Homework 2 is due May 7, 11:59 PM
  – Extended three days
• Reading:
  – Chapter 10: Grouping and Model Fitting
Hough Transform Algorithm

• Typically use a different parameterization
  
  \[ d = x \cos \theta + y \sin \theta \]
  
  – \( d \) is the perpendicular distance from the line to the origin
  – \( \theta \) is the angle this perpendicular makes with the x axis

• Basic Hough transform algorithm
  1. Initialize \( H[d, \theta] = 0 \); \( H \) is called accumulator array
  2. for each edge point \( I[x,y] \) in the image
     for \( \theta = 0 \) to 180
        \( H[d, \theta] += 1 \)
  3. Find the value(s) of \( (d, \theta) \) where \( H[d, \theta] \) is the global maximum
  4. The detected line in the image is given by \( d = x \cos \theta + y \sin \theta \)
  
  • What’s the running time (measured in # votes)?

Hough Transform: 20 colinear points

Hough Transform: Random points

Hough Transform: “Noisy line”

Extension: Oriented Edges

procedure Hough((x, y, \theta)):  
  1. Clear the accumulator array.
  2. For each detected edge at location \( (x, y) \) and orientation \( \theta = \tan^{-1} m \) compute the value of
     \[ d = x m_x + y m_y \]
     and increment the accumulator corresponding to \((\theta, d)\).
  3. Find the peaks in the accumulator corresponding to lines.
     Optionally re-fit the lines to the constituent edges.

Algorithm 4.2 Outline of a Hough transform algorithm based on oriented edge segments.

Hough Transform for Curves

Generalized Hough Transform

• The Hough transform can be generalized to detect any curve that can be expressed in parametric form:

  \[ y = f(x, a_1, a_2, \ldots a_p) \]
  \[ \text{or} \]
  \[ g(x, y, a_1, a_2, \ldots a_p) = 0 \]
  
  – \( a_1, a_2, \ldots a_p \) are the parameters
  – The parameter space is \( p \)-dimensional
  – The accumulating array is large
Example: Finding circles

Equation for circle is

\[(x - x_c)^2 + (y - y_c)^2 = r^2\]

where the parameters are the circle’s center \((x_c, y_c)\) and radius \(r\).

Three dimensional generalized Hough space.

Given an edge point \((x, y)\),

1. Loop over all values of \((x_c, y_c)\),
2. Compute \(r\)
3. Increment \(H(x_c, y_c, r)\)

Transmission Electron Microscopy (TEM) Image of Keyhole Limpet Hemocyanin (KLH) with detected particles

Three-dimensional maps of KLH at a resolution of 23.5 Å reconstructed using particles extracted either manually or automatically as described in the text. (a), (b) The side- and top- view of a 3D map reconstructed from a set of 1042 manually selected particle images.

Processing in Stage 1 for KLH

- Canny edge detection.
- A sequence of ordered Hough transforms (HTs) is applied in order from the computationally simplest one to the most complex one.
- Edges covered by the detected shapes are removed immediately from edge images following the application of the last HT.

Picking KLH Particles in Stage 1

Zhu et al., IEEE Transactions on Medical Imaging, 2003

Line Fitting
Line Fitting

Given \( n \) points \((x_i, y_i)\), estimate parameters of line

\[ ax_i + by_i - d = 0 \]

subject to the constraint that

\[ a^2 + b^2 = 1 \]

Note: \( ax_i + by_i - d \) is distance from \((x_i, y_i)\) to line.

Cost Function:

Sum of squared distances between each point and the line

with respect to \((a, b, d)\).

1. Minimize \( E \) with respect to \( d \):

\[ \frac{\partial E}{\partial d} = 0 \Rightarrow \frac{1}{n} \sum_{i=1}^{n} ax_i + by_i = \bar{x} \bar{y} \]

where \((\bar{x}, \bar{y})\) is the mean of the data points

2. Substitute \( d \) back into \( E \)

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

3. Minimize \( E = |Un|^2 \Rightarrow U^TUn = n^T Sn \) with respect to \( a, b \)

subject to the constraint \( n^Tn = 1 \). Note that \( S \) is given by

\[ S = U^T U = \begin{bmatrix} \sum_{i=1}^{n} x_i^2 - n\bar{x}^2 & \sum_{i=1}^{n} x_iy_i - n\bar{x}\bar{y} \\ \sum_{i=1}^{n} x_iy_i - n\bar{x}\bar{y} & \sum_{i=1}^{n} y_i^2 - n\bar{y}^2 \end{bmatrix} \]

which is real, symmetric, and positive definite

4. This is a constrained optimization problem in \( n \). Solve with Lagrange multiplier

\[ L(n) = n^T Sn - \lambda(n^Tn - 1) \]

Take partial derivative (gradient) w.r.t. \( n \) and set to 0.

\[ \nabla L = 2Sn - 2\lambda n = 0 \]

or

\[ Sn = \lambda n \]

\( n = (a, b) \) is an Eigenvector of the symmetric matrix \( S \)

(the one corresponding to the smallest Eigenvalue).

\( d \) is computed from Step 1.

RANSAC

Slides shamelessly taken from
Frank Dellaert and Marc Pollefeys and
modified

Simpler Example

• Fitting a straight line

Discard Outliers

• No point with \( d > t \)
• RANSAC:
  – RANdom SAmple Consensus
  – Fischler & Bolles 1981
  – Copes with a large proportion of outliers
Main Idea

- Select 2 points at random
- Fit a line
- “Support” = number of inliers
- Line with most inliers wins

Why will this work?

Best Line has most support

- More support -> better fit

RANSAC

Objective

Robust fit of model to data set S which contains outliers

Algorithm

(i) Randomly select a sample of \( s \) data points from S and instantiate the model from this subset.
(ii) Determine the set of data points \( Si \) which are within a distance threshold \( t \) of the model. The set \( Si \) is the consensus set of samples and defines the inliers of S.
(iii) If the size of \( Si \) is greater than some threshold \( T \), re-estimate the model using all the points in \( Si \) and terminate
(iv) If the size of \( Si \) is less than \( T \), select a new subset and repeat the above.
(v) After \( N \) trials the largest consensus set \( Si \) is selected, and the model is re-estimated using all the points in the subset \( Si \).

Number of trials

Choose \( N \) (number of trials) so that, with probability \( p \), at least one random sample is free from outliers. e.g. \( p = 0.99 \)

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

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

e: proportion of outliers

<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>

Number of inliers threshold

- Typically, terminate when inlier ratio reaches expected ratio of inliers

\[ T = (1 - e)N \]
Distance threshold

Choose threshold \( t \) so probability for inlier is \( \alpha \) (e.g., 0.95)
- Often empirically
- Zero-mean Gaussian noise \( \sigma \) then \( d^2 \) follows \( \chi^2 \) distribution with \( m = \) codimension of model
  
  \[ (\text{codimension}=\text{dimension of space} – \text{dimension of subspace}) \]

<table>
<thead>
<tr>
<th>Codimension</th>
<th>Model</th>
<th>( t^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E, F, 2D line</td>
<td>3.84( \sigma^2 )</td>
</tr>
<tr>
<td>2</td>
<td>P</td>
<td>5.99( \sigma^2 )</td>
</tr>
</tbody>
</table>

Using RANSAC to estimate the Fundamental Matrix

- What is the model?
- What is the sample size and where do the samples come from?
- What distance do we use to compute the consensus set?
- How often do outliers occur?

Other models

- 2D motion models
- Typically: points in two images
- Candidates:
  - Translation
  - Euclidean
  - Similarity
  - Affine
  - Projective

Feature Detection and Matching

Input Images

Feature Detection and Matching

Detected Corners

Feature Detection and Matching

Simple Matching
Feature Detection and Matching

Simple Matching
Including Outlier Rejection

Mosaicing: Homography
Estimate with RANSAC

www.cs.cmu.edu/~dellaert/mosaicking

Next Lecture

• Motion
• Reading:
  – Section 10.6.1: Optical Flow and Motion
  – Section 10.6.2: Flow Models
  – Introductory Techniques for 3-D Computer Vision, Trucco and Verri
    • Chapter 8: Motion