Face Recognition Using a Line Edge Map


IEEE Pattern Analysis and Machine Intelligence 2002

Slides by David Anthony Torres
Computer Science and Engineering — University of California at San Diego
Interest Points Vs. Edge Maps

- Interest point detectors are popular
  - SIFT, Harris/Forstner
- What about edge information?
  - Can carry distinguishing info too.
  - Interest points don’t capture this info
Line Edge Map

• Humans recognize line drawings well.
  ➢ Maybe computer algorithms can too.

• Benefits of using edge information:
  ➢ Advantages of template matching and geometrical feature matching:
    ▪ Partially illumination-invariant
    ▪ Low memory requirement
    ▪ Recognition performance of template matching
Line Edge Map

• Takács (1998) used edge maps for face recognition.
  ➢ Apply edge-detector to get a binary input image $I$
  ➢ $I$ is a set of edge points.
  ➢ Use Hausdorff distance to measure the similarity between two sets of points $I_1$ and $I_2$. 
Hausdorff Distance

\[ h(I_1, I_2) = \frac{1}{|I_1|} \sum_{i \in I_1} \min_{j \in I_2} \| i - j \| \]

- \( i \) and \( j \) are edge pixel positions \((x,y)\).
- For each pixel \( i \) in \( I_1 \),
  - Find the closest corresponding pixel \( j \) in \( I_2 \)
  - Take the average of all these distances \(||i-j||\).
- Calculated without explicitly pairing the sets of points.
- Achieved a 92% accuracy in their experiments.
Line Edge Map

• Takács Edge Map doesn’t consider local structure.
• Authors introduce the Line Edge Map (LEM)
• Groups edge pixels into line segments.
  ➢ Apply polygonal line fitting to a thinned edge map
Line Edge Map

- LEM is a series of line segments.
  - LEM records only the endpoints of lines.
  - Further reduces storage requirements.
Line-Segment Hausdorff Distance (LHD)

• Need a new distance measure between sets of line segments.
• Expect it to be better because it uses line-orientation.
• First we’ll see an initial model…
• Add to the model to make it more robust
  ➢ Encourage one-one mapping of lines
  ➢ Encourage mapping of “similar” lines.
Line-Segment Hausdorff Distance

• Given two LEMs $S = (s_1, s_2, \ldots s_p)$ and $T = (t_1, t_2, \ldots t_q)$
• The LHD is built on the vector $d(s_i, t_j)$
  ➢ $d()$ represents the distance between two line segments

\[
\vec{d}(m_i^l, t_j^l) = \begin{bmatrix}
  d_\theta(m_i^l, t_j^l) \\
  d_{//}(m_i^l, t_j^l) \\
  d_\perp(m_i^l, t_j^l)
\end{bmatrix}
\]
Line-Segment Hausdorff Distance

\[
\bar{d}(m_i^l, t_j^l) = \begin{bmatrix}
    d_\theta(m_i^l, t_j^l) \\
    d_{\parallel}(m_i^l, t_j^l) \\
    d_{\perp}(m_i^l, t_j^l)
\end{bmatrix}
\]

\[
d_\theta \left( m_i^l, t_j^l \right)
\]

(a) (b)

smallest intersecting angle
Line-Segment Hausdorff Distance

\[ d_\theta \left( m_i^l, t_j^l \right) = f \left( \theta \left( m_i^l, t_j^l \right) \right) \]

- \( f() \) is a penalty function: \( f(\theta) = \theta^2 / W \)
  - Higher penalty on large deviation
- \( W \) is determined in training.
Line-Segment Hausdorff Distance

\[ d_{\parallel}(m_i^l, t_j^l) = \min(l_{\parallel 1}, l_{\parallel 2}) \]

\[ l_{\perp} \]

\[ l_{\parallel 1} \]

\[ l_{\parallel 2} \]
Line-Segment Hausdorff Distance

\[ d_\perp (m_i^l, t_j^l) = l_\perp \]

- In general lines will not be parallel
- So rotate the shortest line
Line-Segment Hausdorff Distance

• Finally,

\[ d(m_i^l, t_j^l) = \sqrt{d_0^2(m_i^l, t_j^l) + d_{//}^2(m_i^l, t_j^l) + d_{\bot}^2(m_i^l, t_j^l)} \]

• Primary line-segment Hausdorff Distance (LHD)

\[ H(I, J) = \max(h(I, J), h(J, I)) \]

where

\[ h(I, J) = \frac{1}{\sum_{i \in I} \left\| i \right\|} \sum_{i \in I} \left\| i \right\| \cdot \min_{j \in J} d(i, j) \]
Some Problems...

• Say $T$ is an input LEM, $M$ is its matching model LEM, and $N$ is some other non-matching model.

• Due to segmentation problems it could be the case that

$$H(T,M) >> H(T,N)$$

• Keeping track of matched line-pairs could help.
Neighborhoods

• Positional neighborhood $N_p$
• Angular neighborhood $N_a$
• Heuristic: lines that fall within the neighborhood are probably matches.
Neighborhoods

- If \( \geq 1 \) line falls into the neighborhoods we call the original line segment \( I \), a high confidence line.

\[ \Theta \text{-neighborhood} \]

Matching Line-segment in \( J \)

Line Segment in \( I \) is a High Confidence Line
High Confidence Ratio

- $N_{hc}$ is the num. of high confidence lines in a LEM.
- $N_{total}$ is the total num. of lines in a LEM.

$$R = \frac{N_{hc}}{N_{total}}$$
New Hausdorff Distance

\[ H'(T, M) = \sqrt{H^2(T, M) + (W_n D_n)^2} \]

- \( W_n \) is a weight.
- \( D_n \) is the average number of lines (across input and model) that are not confidently-matched, i.e.

\[ D_n = 1 - \frac{R_M + R_T}{2} = \frac{(1 - R_M) + (1 - R_T)}{2} \]

\( R_T \) and \( R_M \) are the high confidence ratios for input and model respectively.
Summary

• Start with to LEM’s

• Calculate Hausdorff Distance

\[ H(I, J) = \max(h(I, J), h(J, I)) \]

\[ h(I, J) = \frac{1}{\sum_{i \in I} \| i \|} \sum_{i \in I} \| i \| \cdot \min_{j \in J} d(i, j) \]
Summary

- \[ d(m_i^l, t_j^l) = \sqrt{d_\theta^2(m_i^l, t_j^l) + d_{/\parallel}^2(m_i^l, t_j^l) + d_{/\perp}^2(m_i^l, t_j^l)} \]

- \[ \vec{d}(m_i^l, t_j^l) = \begin{bmatrix} d_\theta(m_i^l, t_j^l) \\ d_{/\parallel}(m_i^l, t_j^l) \\ d_{/\perp}(m_i^l, t_j^l) \end{bmatrix} \]
Summary

• Finally we take into account the effect of neighborhoods

\( H'(T, M) = \sqrt{H^2(T, M) + (W_n D_n)^2} \)

• \( D_n = 1 - \frac{R_M + R_T}{2} = \frac{(1 - R_M) + (1 - R_T)}{2} \)
Free Parameters

- We have four free parameters to fix
  - \((W, W_n, N_p, N_a)\)
    - \(\theta^2/W = f(\theta) = d_\theta\)
    - \(H'(T, M) = \sqrt{H^2(T, M) + (W_nD_n)^2}\)
    - Neighborhoods \(N_p, N_a\)

- Use simulated annealing to estimate
  - With probability \(p = e^{-\frac{\Delta Err}{t}}\)
Results
Face Recognition under Controlled Conditions

Bern Database

AR Database
## Face Recognition under Controlled Conditions

<table>
<thead>
<tr>
<th>Method</th>
<th>Bern database</th>
<th>AR database</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EM</td>
<td>Eigenface</td>
</tr>
<tr>
<td>Recognition rate</td>
<td>96.7%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Face Recognition under Controlled Conditions

<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEM</td>
<td>96.43%</td>
</tr>
<tr>
<td>Eigenface (20-eigenvectors)</td>
<td>55.36%</td>
</tr>
<tr>
<td>Eigenface (60-eigenvectors)</td>
<td>71.43%</td>
</tr>
<tr>
<td>Eigenface (112-eigenvectors)</td>
<td>78.57%</td>
</tr>
</tbody>
</table>
Face Recognition under Controlled Conditions

![Recognition rate after 2 weeks]

<table>
<thead>
<tr>
<th>Method</th>
<th>Top 1</th>
<th>Top 3</th>
<th>Top 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>edge map (MHD)</td>
<td>88.39</td>
<td>95.54</td>
<td>98.21</td>
</tr>
<tr>
<td>LEM (pLHD)</td>
<td>93.75</td>
<td>97.32</td>
<td>100</td>
</tr>
<tr>
<td>LEM (LHD)</td>
<td>96.43</td>
<td>99.11</td>
<td>100</td>
</tr>
</tbody>
</table>

w/o neighborhood heuristic
Sensitivity to Size Variation

**TABLE 3**
Recognition Results with Size Variations

<table>
<thead>
<tr>
<th>Method</th>
<th>Top 1</th>
<th>Top 5</th>
<th>Top 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge map</td>
<td>43.3%</td>
<td>56.0%</td>
<td>64.7%</td>
</tr>
<tr>
<td>Eigenface (112-eigenvectors)</td>
<td>44.9%</td>
<td>68.8%</td>
<td>75.9%</td>
</tr>
<tr>
<td>LEM (pLHD)</td>
<td>53.8%</td>
<td>67.6%</td>
<td>71.9%</td>
</tr>
<tr>
<td>LEM (LHD)</td>
<td>66.5%</td>
<td>75.9%</td>
<td>79.7%</td>
</tr>
</tbody>
</table>

- Used the AR data base.
- Applied a random scaling factor of ±10%
Recognition Under Varying Lighting
<table>
<thead>
<tr>
<th>Testing faces</th>
<th>Eigenface</th>
<th>Edge map</th>
<th>LEM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20-eigenvectors</td>
<td>6.25%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60-eigenvectors</td>
<td>9.82%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors</td>
<td>9.82%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors w/o 1(^{st}) 3</td>
<td>26.79%</td>
<td></td>
</tr>
<tr>
<td>Left light on</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20-eigenvectors</td>
<td>4.46%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60-eigenvectors</td>
<td>7.14%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors</td>
<td>7.14%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors w/o 1(^{st}) 3</td>
<td>49.11%</td>
<td></td>
</tr>
<tr>
<td>Right light on</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20-eigenvectors</td>
<td>1.79%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60-eigenvectors</td>
<td>2.68%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors</td>
<td>2.68%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors w/o 1(^{st}) 3</td>
<td>64.29%</td>
<td></td>
</tr>
<tr>
<td>Both lights on</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recognition Under Facial Expression Changes
<table>
<thead>
<tr>
<th>Testing faces</th>
<th>Eigenface</th>
<th>EM</th>
<th>LEM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Smiling expression</strong></td>
<td>20-eigenvectors</td>
<td>87.85%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60-eigenvectors</td>
<td>94.64%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors</td>
<td>93.97%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors w/o 1st 3</td>
<td>82.04%</td>
<td></td>
</tr>
<tr>
<td><strong>Angry expression</strong></td>
<td>20-eigenvectors</td>
<td>78.57%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60-eigenvectors</td>
<td>84.82%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors</td>
<td>87.50%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors w/o 1st 3</td>
<td>73.21%</td>
<td></td>
</tr>
<tr>
<td><strong>Screaming expression</strong></td>
<td>20-eigenvectors</td>
<td>34.82%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60-eigenvectors</td>
<td>41.96%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors</td>
<td>45.54%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>112-eigenvectors w/o 1st 3</td>
<td>32.14%</td>
<td></td>
</tr>
</tbody>
</table>
View Based Identification — “Leave One Out” Experiment.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge map</td>
<td>26.06%</td>
</tr>
<tr>
<td>Eigenface*</td>
<td>24.4%</td>
</tr>
<tr>
<td>Correlation*</td>
<td>23.9%</td>
</tr>
<tr>
<td>Linear Subspace*</td>
<td>21.6%</td>
</tr>
<tr>
<td>Eigenface w/o 1(^{st}) 3*</td>
<td>15.3%</td>
</tr>
<tr>
<td>LEM</td>
<td>14.55%</td>
</tr>
<tr>
<td>Fisherface*</td>
<td>7.3%</td>
</tr>
</tbody>
</table>
## Recognition Under Varying Pose

**TABLE 7**  
Face Recognition Results under Pose Different Variations

<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Edge map</td>
</tr>
<tr>
<td>Looks left/right</td>
<td>50.00%</td>
</tr>
<tr>
<td>Looks up</td>
<td>65.00%</td>
</tr>
<tr>
<td>Looks down</td>
<td>67.67%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>58.17%</td>
</tr>
</tbody>
</table>
Additional Material...
Matching Time for LEM

- LEM takes longer than eigenface
  - Time $O(Nn) > O(Nm)$
    - $N$ is # of faces
    - $n$ is avg. # LEM-features
    - $m$ is # eigenvectors
- Authors propose a face pre-filtering scheme
  - Idea: filter out faces before performing matching.
Face Prefiltering

• Quantize an LEM into:

\[ \vec{S} = \begin{bmatrix} \Gamma \\ \Theta \end{bmatrix} \]

• Where \( \Gamma \) is the sum of line segment lengths

• \( \Theta = \frac{1}{\sum_i l_i} \sum_i \phi_i l_i \)

where \( \phi \) is the angle if the angle is <90 degrees.
Face Pre-filtering

\[ \Delta \mathbf{S} \sim N_2 \left( \bar{\mu}, \bar{\Sigma} \right), \]

where

\[
\bar{\mu} = \begin{bmatrix} \mu_1 \\ \mu_\theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \bar{\Sigma} = \begin{bmatrix} \sigma^2_1 & \sigma_{1\theta} \\ \sigma_{\theta 1} & \sigma^2_\theta \end{bmatrix} = \begin{bmatrix} \sigma^2_1 & \sigma_1 \sigma_\theta \rho \\ \sigma_\theta \sigma_1 \rho & \sigma^2_\theta \end{bmatrix},
\]

and the correlation coefficient

\[
\rho = \frac{\sigma_{1\theta}}{\sigma_1 \sigma_\theta}.
\]
Face Pre-filtering

Then, the density function of the error vector can be represented as

\[
f(\Delta \vec{S}) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} \left( \Delta \vec{S} - \vec{\mu} \right)^T \Sigma^{-1} \left( \Delta \vec{S} - \vec{\mu} \right) \right\}, \Delta \vec{S} \in \mathbb{R}^2.
\]
Face Pre-filtering

Since $|\tilde{\Sigma}| = \sigma_l^2 \sigma_\theta^2 (1 - \rho^2)$, the inverse of $\tilde{\Sigma}$ exists if and only if $|\rho| < 1$. Straightforward calculation shows that

$$
\tilde{\Sigma}^{-1} = \frac{1}{\sigma_l^2 \sigma_\theta^2 (1 - \rho^2)} \begin{bmatrix}
\sigma_l^2 & -\sigma_l \sigma_\theta \rho \\
-\sigma_\theta \sigma_l \rho & \sigma_\theta^2
\end{bmatrix}.
$$

(15)
Face Pre-filtering

Thus, the density function of $\Delta \bar{S}$ becomes

$$f(\Delta \bar{S}) = \frac{1}{2\pi \sigma_l \sigma_\theta \sqrt{1 - \rho^2}} \exp \left\{ - \frac{1}{2(1 - \rho^2)} \left[ \left( \frac{\Delta \Gamma - \mu_l}{\sigma_l} \right)^2 \right. \right.$$ 

$$- 2\rho \left( \frac{\Delta \Gamma - \mu_l}{\sigma_l} \right) \left( \frac{\Delta \Theta - \mu_\theta}{\sigma_\theta} \right) + \left( \frac{\Delta \Theta - \mu_\theta}{\sigma_\theta} \right)^2 \left. \right] \right\}$$
Face Pre-filtering

The constant density contours for a bivariate normal are a series of ellipses with different values of \( d \) as shown in the following equation:

\[
\left( \Delta \mathbf{S} - \bar{\mu} \right)^T \Sigma^{-1} \left( \Delta \mathbf{S} - \bar{\mu} \right) = d^2
\]

or

\[
\left( \frac{\Delta \Gamma - \mu_l}{\sigma_l} \right)^2 - 2\rho \left( \frac{\Delta \Gamma - \mu_l}{\sigma_l} \right) \left( \frac{\Delta \Theta - \mu_\theta}{\sigma_\theta} \right) + \left( \frac{\Delta \Theta - \mu_\theta}{\sigma_\theta} \right)^2 = d^2 (1 - \rho^2).
\]
Face Pre-filtering

The probability that $\Delta S$ falls in the elliptic region $\Omega$ of parameter $d$ is given by

$$F(d) = \Pr \left( \Delta S \in \Omega \right) = \int \int_{\Omega} f \left( \Delta S \right) d(\Delta \Gamma) d(\Delta \Theta)$$

$$= \int \int_{\Omega} \frac{1}{2\pi \sigma_l \sigma_\theta \sqrt{1 - \rho^2}} \exp \left\{ - \frac{1}{2(1 - \rho^2)} \left[ \left( \frac{\Delta \Gamma - \mu_l}{\sigma_l} \right)^2 
- 2\rho \left( \frac{\Delta \Gamma - \mu_l}{\sigma_l} \right) \left( \frac{\Delta \Theta - \mu_\theta}{\sigma_\theta} \right)
+ \left( \frac{\Delta \Theta - \mu_\theta}{\sigma_\theta} \right)^2 \right] \right\} d(\Delta \Gamma) d(\Delta \Theta).$$
Face Pre-filtering

Let

\[ u = \frac{\Delta \Gamma - \mu_l}{\sigma_l}, \quad v = \frac{\Delta \Theta - \mu_\theta}{\sigma_\theta}. \]  

(19)

The equation of constant density contour can be rewritten as

\[ u^2 - 2\rho uv + v^2 = d^2(1 - \rho^2). \]  

(20)
Face - Prefiltering

\[ F(d) = \int \int_{\Omega} \frac{1}{2\pi \sqrt{1 - \rho^2}} \exp \left\{ -\frac{1}{2(1 - \rho^2)} \left[ u^2 - 2\rho uv + v^2 \right] \right\} dvudv \]
Now for some hand-waving action...

- Rotate the Gaussian so that its axis aligned
- Perform a change of coordinates into a polar system

\[
F(d) = \int_{0}^{d} \int_{0}^{2\pi} \frac{1}{2\pi ab} \exp \left\{ -\frac{1}{2} r^2 \right\} |J| dr d\theta
\]

\[
= 1 - e^{-\frac{1}{2}d^2}.
\]

- \[d = \sqrt{-2 \ln[1 - F(d)]}\]
To summarize

• Given a probability \( F(d) \) we can obtain a constant density ellipse of the form:

\[
\left( \frac{\Delta \Gamma - \mu_l}{\sigma_l} \right)^2 - 2\rho \left( \frac{\Delta \Gamma - \mu_l}{\sigma_l} \right) \left( \frac{\Delta \Theta - \mu_\theta}{\sigma_\theta} \right) + \left( \frac{\Delta \Theta - \mu_\theta}{\sigma_\theta} \right)^2 = d^2 \left( 1 - \rho^2 \right)
\]

• where

\[
d = \sqrt{-2 \ln[1 - F(d)]}
\]
To summarize

- So if the error vector satisfies:

\[
\left( \frac{\Delta \Gamma}{\sigma_l} \right)^2 - 2\rho \left( \frac{\Delta \Gamma}{\sigma_l} \right) \left( \frac{\Delta \Theta}{\sigma_\theta} \right) + \left( \frac{\Delta \Theta}{\sigma_\theta} \right)^2 < d^2 (1 - \rho^2).
\]

- then the model is classified as a potential face.
Pre-Filtering Results

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( c_1 )</th>
<th>( c_\theta )</th>
<th>( \mu_1 )</th>
<th>( \mu_\theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>145.36</td>
<td>4.33</td>
<td>26.42</td>
<td>0.27</td>
</tr>
</tbody>
</table>

- Train to find parameter above.
- Small rho indicates vector components are nearly independent.
<table>
<thead>
<tr>
<th>$F(d)$</th>
<th>$d^2$</th>
<th>True acceptance rate</th>
<th>Filter out rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>90%</td>
<td>4.61</td>
<td>88.39%</td>
<td>50.31%</td>
</tr>
<tr>
<td>95%</td>
<td>5.99</td>
<td>92.86%</td>
<td>41.37%</td>
</tr>
<tr>
<td>99%</td>
<td>9.21</td>
<td>97.32%</td>
<td>26.58%</td>
</tr>
<tr>
<td>99.5%</td>
<td>10.60</td>
<td>99.11%</td>
<td>22.02%</td>
</tr>
<tr>
<td>99.7%</td>
<td>12.43</td>
<td>100%</td>
<td>17.06%</td>
</tr>
<tr>
<td>$F(d)$</td>
<td>$d^2$</td>
<td>True acceptance rate</td>
<td>Filter out rate</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>----------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>90%</td>
<td>4.61</td>
<td>96.67%</td>
<td>61.55%</td>
</tr>
<tr>
<td>95%</td>
<td>5.99</td>
<td>96.67%</td>
<td>53.91%</td>
</tr>
<tr>
<td>96%</td>
<td>6.44</td>
<td>100%</td>
<td>51.95%</td>
</tr>
</tbody>
</table>