CAP 5415 Computer Vision
Fall 2012

Dr. Mubarak Shah
Univ. of Central Florida
Example
An Application

- What is an object?
- How can we find it?
Edge Detection in Images

- At edges intensity or color changes
What is an Edge?

- Discontinuity of intensities in the image
- Edge models
  - Step
  - Roof
  - Ramp
  - Spike

Alper Yilmaz, Mubarak Shah Fall 2012 UCF
Detecting Discontinuities

- **Image derivatives**

\[
\frac{\partial f}{\partial x} = \lim_{\varepsilon \to 0} \left( \frac{f(x + \varepsilon) - f(x)}{\varepsilon} \right)
\]

\[
\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}) - f(x)}{\Delta x}
\]

- **Convolve image with derivative filters**

  - Backward difference
  - Forward difference
  - Central difference

Alper Yilmaz, Mubarak Shah Fall 2012 UCF
Derivative in Two-Dimensions

- **Definition**

\[
\frac{\partial f(x, y)}{\partial x} = \lim_{\varepsilon \to 0} \left( \frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon} \right)
\]

\[
\frac{\partial f(x, y)}{\partial y} = \lim_{\varepsilon \to 0} \left( \frac{f(x, y + \varepsilon) - f(x, y)}{\varepsilon} \right)
\]

- **Approximation**

\[
\frac{\partial f(x, y)}{\partial x} \approx \frac{f(x_{n+1}, y_m) - f(x_n, y_m)}{\Delta x}
\]

\[
\frac{\partial f(x, y)}{\partial y} \approx \frac{f(x_n, y_{m+1}) - f(x_n, y_m)}{\Delta x}
\]

- **Convolution kernels**

\[
f_x = \begin{bmatrix} 1 & -1 \end{bmatrix}
\]

\[
f_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}
\]

Alper Yilmaz, Mubarak Shah Fall 2012 UCF
Image Derivatives

Image $I$

\[ I_x = I \ast \begin{bmatrix} 1 & -1 \end{bmatrix} \]

\[ I_y = I \ast \begin{bmatrix} 1 \\ -1 \end{bmatrix} \]
Derivatives and Noise

- **Strongly affected by noise**
  - obvious reason: image noise results in pixels that look very different from their neighbors
- **The larger the noise is the stronger the response**

- **What is to be done?**
  - Neighboring pixels look alike
  - Pixel along an edge look alike
  - Image smoothing should help
    - Force pixels different to their neighbors (possibly noise) to look like neighbors

Alper Yilmaz, Mubarak Shah Fall 2012 UCF
Derivatives and Noise

Increasing noise

Zero mean additive gaussian noise

Alper Yilmaz, Mubarak Shah Fall 2012 UCF
Image Smoothing

- Expect pixels to “be like” their neighbors
  - Relatively few reflectance changes
- Generally expect noise to be independent from pixel to pixel
  - Smoothing suppresses noise
Gaussian Smoothing

- Scale of Gaussian $\sigma$
  - As $\sigma$ increases, more pixels are involved in average
  - As $\sigma$ increases, image is more blurred
  - As $\sigma$ increases, noise is more effectively suppressed

$g(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$
Gaussian Smoothing (Examples)

σ=0.05

σ=0.1

σ=0.2

no smoothing

σ=1 pixel

σ=2 pixels

Alper Yilmaz, Mubarak Shah Fall 2012 UCF
Edge Detectors

- Gradient operators
  - Prewit
  - Sobel
- Laplacian of Gaussian (Marr-Hildreth)
- Gradient of Gaussian (Canny)
Prewitt and Sobel Edge Detector

- Compute derivatives
  - In $x$ and $y$ directions
- Find gradient magnitude
- Threshold gradient magnitude
Prewitt Edge Detector

Alper Yilmaz, Mubarak Shah Fall 2012 UCF
Sobel Edge Detector

Image $I$

\[
\begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1
\end{bmatrix}
\]

\[\frac{d}{dx} I\]

\[\frac{d}{dy} I\]

$\sqrt{\left(\frac{d}{dx} I\right)^2 + \left(\frac{d}{dy} I\right)^2}$

Threshold \hspace{1cm} Edges

Alper Yilmaz, Mubarak Shah Fall 2012 UCF
Sobel Edge Detector

Alper Yilmaz, Mubarak Shah Fall 2012 UCF
Sobel Edge Detector

$$\Delta = \sqrt{\left(\frac{d}{dx}I\right)^2 + \left(\frac{d}{dy}I\right)^2}$$

$$\Delta \geq \text{Threshold} = 100$$

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Marr Hildreth Edge Detector

- Smooth image by Gaussian filter $\Rightarrow S$
- Apply Laplacian to $S$
  - Used in mechanics, electromagnetics, wave theory, quantum mechanics and Laplace equation
- Find zero crossings
  - Scan along each row, record an edge point at the location of zero-crossing.
  - Repeat above step along each column
Marr Hildreth Edge Detector

- **Gaussian smoothing**

\[
\tilde{S} = \tilde{g} \ast \tilde{I}
\]

\[
g = \frac{1}{\sqrt{2\pi \sigma}} e^{-\frac{x^2 + y^2}{2\sigma^2}}
\]

- **Find Laplacian**

\[
\Delta^2 S = \frac{\partial^2}{\partial x^2} S + \frac{\partial^2}{\partial y^2} S
\]

- \(\nabla\) is used for gradient (derivative)
- \(\Delta\) is used for Laplacian

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Marr Hildreth Edge Detector

- Deriving the Laplacian of Gaussian (LoG)

\[
\Delta^2 S = \Delta^2 (g \ast I) = (\Delta^2 g) \ast I
\]

\[
\Delta^2 g = -\frac{1}{\sqrt{2\pi}\sigma^3}\left(2 - \frac{x^2 + y^2}{\sigma^2}\right)e^{-\frac{x^2 + y^2}{2\sigma^2}}
\]

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Gaussian

$$g(x) = e^{-\frac{x^2}{2\sigma^2}}$$

<table>
<thead>
<tr>
<th>$x$</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g(x)$</td>
<td>.011</td>
<td>.13</td>
<td>.6</td>
<td>1</td>
<td>.6</td>
<td>.13</td>
<td>.011</td>
</tr>
</tbody>
</table>
2-D Gaussian

\[ g(x, y) = e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>11</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>4</td>
<td>11</td>
<td>20</td>
<td>30</td>
<td>34</td>
<td>30</td>
<td>20</td>
<td>11</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>11</td>
<td>26</td>
<td>50</td>
<td>73</td>
<td>82</td>
<td>73</td>
<td>50</td>
<td>26</td>
<td>11</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>20</td>
<td>50</td>
<td>93</td>
<td>136</td>
<td>154</td>
<td>136</td>
<td>93</td>
<td>50</td>
<td>20</td>
<td>6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>30</td>
<td>73</td>
<td>136</td>
<td>198</td>
<td>225</td>
<td>198</td>
<td>136</td>
<td>73</td>
<td>30</td>
<td>9</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>34</td>
<td>82</td>
<td>154</td>
<td>225</td>
<td>255</td>
<td>225</td>
<td>154</td>
<td>82</td>
<td>34</td>
<td>11</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>30</td>
<td>73</td>
<td>136</td>
<td>198</td>
<td>225</td>
<td>198</td>
<td>136</td>
<td>73</td>
<td>30</td>
<td>9</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>20</td>
<td>50</td>
<td>93</td>
<td>136</td>
<td>154</td>
<td>136</td>
<td>93</td>
<td>50</td>
<td>20</td>
<td>6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>11</td>
<td>26</td>
<td>50</td>
<td>73</td>
<td>82</td>
<td>73</td>
<td>50</td>
<td>26</td>
<td>11</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>4</td>
<td>11</td>
<td>20</td>
<td>30</td>
<td>34</td>
<td>30</td>
<td>20</td>
<td>11</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>11</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
2-D Gaussian
LoG Filter

\[ \Delta^2 G_\sigma = -\frac{1}{\sqrt{2\pi \sigma^3}} \left( 2 - \frac{x^2 + y^2}{\sigma^2} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

\begin{array}{cccccccc}
0.0008 & 0.0066 & 0.0215 & 0.031 & 0.0215 & 0.0066 & 0.0008 \\
0.0066 & 0.0438 & 0.0982 & 0.108 & 0.0982 & 0.0438 & 0.0066 \\
0.0215 & 0.0982 & 0 & -0.242 & 0 & 0.0982 & 0.0215 \\
0.031 & 0.108 & -0.242 & -0.7979 & -0.242 & 0.108 & 0.031 \\
0.0215 & 0.0982 & 0 & -0.242 & 0 & 0.0982 & 0.0215 \\
0.0066 & 0.0438 & 0.0982 & 0.108 & 0.0982 & 0.0438 & 0.0066 \\
0.0008 & 0.0066 & 0.0215 & 0.031 & 0.0215 & 0.0066 & 0.0008 \\
\end{array}
Finding Zero Crossings

- Four cases of zero-crossings:
  - {+, -}
  - {+, 0, -}
  - {-, +}
  - {-, 0, +}
- Slope of zero-crossing {a, -b} is |a+b|.
- To mark an edge
  - compute slope of zero-crossing
  - Apply a threshold to slope

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
On the Separability of LoG

- Similar to separability of Gaussian filter
  - Two-dimensional Gaussian can be separated into 2 one-dimensional Gaussians
    \[ h(x, y) = I(x, y) \ast g(x, y) \]
    \[ h(x, y) = (I(x, y) \ast g_1(x)) \ast g_2(y) \]
    \[ g(x) = e^{-\frac{x^2}{2\sigma^2}} \]
    \[ g_1 = g(x) = [0.011 \quad 0.13 \quad 0.6 \quad 1.0 \quad 0.13 \quad 0.011] \]
    \[ g_2 = g(y) = [0.011 \quad 0.13 \quad 0.6 \quad 1.0 \quad 0.13 \quad 0.011] \]

\[ n^2 \text{ multiplications} \]
\[ 2n \text{ multiplications} \]

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
On the Separability of LoG

\[ \Delta^2 S = \Delta^2 (g * I) = (\Delta^2 g) * I = I * (\Delta^2 g) \]

Requires \( n^2 \) multiplications

\[ \Delta^2 S = (I * g_{xx}(x)) * g(y) + (I * g_{yy}(y)) * g(x) \]

Requires \( 4n \) multiplications

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Seperability

Gaussian Filtering

Image → $g(x)$ → $g(y)$ → $I \ast g$

Laplacian of Gaussian Filtering

Image → $g_{xx}(x)$ → $g(y)$ → $\Delta^2 S$

Image → $g_{yy}(y)$ → $g(x)$ → $\Delta^2 S$

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Example

$I$  $I * (\Delta^2 g)$  Zero crossings of $\Delta^2 S$

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Example

\[ \sigma = 1 \]

\[ \sigma = 3 \]

\[ \sigma = 6 \]
Algorithm

- Compute LoG
  - Use 2D filter
  - Use 4 1D filters
- Find zero-crossings from each row
- Find slope of zero-crossings
- Apply threshold to slope and mark edges

\[ \Delta^2 g(x, y) \]

\[ g(x), \; g_{xx}(x), \; g(y), \; g_{yy}(y) \]
Quality of an Edge

- Robust to noise
- Localization
- Too many or too less responses
Quality of an Edge

- True edge
- Poor robustness to noise
- Poor localization
- Too many responses

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector

- **Criterion 1: Good Detection:** The optimal detector must minimize the probability of false positives as well as false negatives.

- **Criterion 2: Good Localization:** The edges detected must be as close as possible to the true edges.

- **Single Response Constraint:** The detector must return one point only for each edge point.

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector Steps

1. Smooth image with Gaussian filter
2. Compute derivative of filtered image
3. Find magnitude and orientation of gradient
4. Apply “Non-maximum Suppression”
5. Apply “Hysteresis Threshold”
Canny Edge Detector
First Two Steps

- **Smoothing**
  \[ S = I \ast g(x, y) = g(x, y) \ast I \]

- **Derivative**
  \[ \nabla S = \nabla (g \ast I) = (\nabla g) \ast I \]
  \[ \nabla S = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \ast I = \begin{bmatrix} g_x \ast I \\ g_y \ast I \end{bmatrix} \]

\[ g(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

\[ \nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \]

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector
Derivative of Gaussian

\[ g(x, y) \]

\[ g_x(x, y) \]

\[ g_y(x, y) \]

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector
First Two Steps

\[ I \]

\[ S_x \]

\[ S_y \]

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector
Third Step

- Gradient magnitude and gradient direction

\[
(S_x, S_y) \quad \text{Gradient Vector magnitude} \quad \sqrt{S_x^2 + S_y^2}
\]

\[
\text{direction} = \theta = \tan^{-1} \left( \frac{S_y}{S_x} \right)
\]

image

gradient magnitude

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector
Fourth Step

- Non maximum suppression

We wish to mark points along the curve where the magnitude is largest. We can do this by looking for a maximum along a slice normal to the curve (non-maximum suppression). These points should form a curve. There are then two algorithmic issues: at which point is the maximum, and where is the next one?

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector
Non-Maximum Suppression

- Suppress the pixels in $|\nabla S|$ which are not local maximum

$$M(x, y) = \begin{cases} |\nabla S|(x, y) & \text{if } |\nabla S|(x, y) > |\Delta S|(x', y') \\
& \quad \text{and } |\Delta S|(x, y) > |\Delta S|(x'', y'') \\
0 & \text{otherwise} \end{cases}$$

$x'$ and $x''$ are the neighbors of $x$ along normal direction to an edge

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector
Non-Maximum Suppression

\[ |\Delta S| = \sqrt{S_x^2 + S_y^2} \]

For visualization
\[ M \geq \text{Threshold} = 25 \]

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector
Hysteresis Thresholding

- If the gradient at a pixel is
  - above “High”, declare it as an ‘edge pixel’
  - below “Low”, declare it as a “non-edge-pixel”
  - between “low” and “high”

- Consider its neighbors iteratively then declare it an “edge pixel” if it is connected to an ‘edge pixel’ directly or via pixels between “low” and “high”.

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector
Hysteresis Thresholding

- Connectedness

4 connected

8 connected

6 connected

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector
Hysteresis Thresholding

- Scan the image from left to right, top-bottom.
  - The gradient magnitude at a pixel is above a high threshold declare that as an edge point
  - Then recursively consider the neighbors of this pixel.
    - If the gradient magnitude is above the low threshold declare that as an edge pixel.

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Canny Edge Detector

Hysteresis Thresholding

$M \geq 25$

Hysteresis

$High = 35$

$Low = 15$

Alper Yilmaz, Mubarak Shah Fall 2012, UCF
Suggested Reading

- Chapter 4, Emanuele Trucco, Alessandro Verri, "Introductory Techniques for 3-D Computer Vision"
- Chapter 2, Mubarak Shah, “Fundamentals of Computer Vision”

Alper Yilmaz, Mubarak Shah Fall 2012, UCF