Outline

• Extracting useful information from Images
  – Edge Models & Edge Detection
Edges

- **Goal:** Identify sudden changes (discontinuities) in an image
  - Most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels
  - Marks the border of an object
Origins of Edges

- Edges are caused by a variety of factors

  - surface normal discontinuity
  - depth discontinuity
  - surface color discontinuity
  - illumination discontinuity
Close-up Edges
Close-up Edges
Characterizing Edges

• An edge is a place of rapid change in the image intensity function

Slide Credit: James Hays
Gradient Operations (Gx, Gy)

\[ G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \]
Revisit: Image Gradient

- The gradient of an image: \( \nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \)

\[
\nabla f = \left[ \frac{\partial f}{\partial x}, 0 \right]
\]

\[
\nabla f = \left[ 0, \frac{\partial f}{\partial y} \right]
\]

The gradient points in the direction of most rapid increase in intensity.

The gradient direction is given by \( \theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right) \).

- how does this relate to the direction of the edge?

The edge strength is given by the gradient magnitude:

\[
\| \nabla f \| = \sqrt{ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 }
\]

Source: Steve Seitz
Effects of Noise on Gradient Computation

Increasing noise

Zero mean additive Gaussian noise
Effects of Noise

Where is the edge (solution of derivative)?
Solution: Smoothing

- Smoothing removes noise, but *blurs* edge.
Edge Detection

- Requires **difference** filters
- What do we expect from these filters? What should be the main properties of such filters?
Design Criteria for Edge Detection Problems

- **Good Detection**: minimize prob. of FP (detecting spurious edges) and FN (missing real edges)
- **Good Localization**: must be as close as possible to the true edges
- **Single Response**: must return one point only for each true edge point
Basic Comparisons of Edge Operators

Gradient:
\[
\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]
\]

Roberts (2 x 2):

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Sobel (3 x 3):

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Sobel (5 x 5):

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>-2</td>
<td>-3</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>-3</td>
<td>-5</td>
<td>0</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>-2</td>
<td>-3</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Good Localization
Noise Sensitive
Poor Detection

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Good Detection

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-2</td>
<td>-3</td>
<td>-5</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
</tbody>
</table>

Poor Localization
Less Noise Sensitive
Example: Laplacian of Gaussian (LoG) and Canny Edge Detector

Marr and Hildreth Filtering, 1980. (LoG)
• Smooth Image with Gaussian Filter
• Applying the Laplacian for a Gaussian-filtered image can be done in one step of convolution.
• Find zero-crossings
• Find slope of zero-crossings
• Apply threshold to slope and mark edges

J. Canny. 1986. (Canny)
• Smooth Image with Gaussian filter
• Compute Derivative of filtered image
• Find Magnitude and Orientation of gradient
• Apply Non-max suppression
• Apply Thresholding (Hysteresis)
Example: Laplacian of Gaussian (LoG) and Canny Edge Detector

*Marr and Hildreth Filtering, 1980. (LoG)*
- Smooth Image with Gaussian Filter
- Applying the Laplacian for a Gaussian-filtered image can be done in one step of convolution.
- Find zero-crossings
- Find slope of zero-crossings
- Apply threshold to slope and mark edges

*J. Canny. 1986. (Canny)*
- Smooth Image with Gaussian filter
- Compute Derivative of filtered image
- Find Magnitude and Orientation of gradient
- Apply Non-max suppression
- Apply Thresholding (Hysteresis)
Laplacian Operator

- Edges can also be identified with zero-crossings of second-order derivatives.

\[ \Delta I = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \]
Laplacian Operator

Gaussian

Derivative of Gaussian in $x$ direction (gradient)

Derivative of Gaussian in $y$ direction (gradient)

Laplacian of Gaussian
Laplacian of ....?

• Since second derivative (zero-crossing) is very sensitive to noise, it is desirable to .... images?
Laplacian of **Gaussian** (LoG)

- Since second derivative (zero-crossing) is very sensitive to noise, it is desirable to **smooth** images
Laplacian of **Gaussian** (LoG)

- Since second derivative (zero-crossing) is very sensitive to noise, it is desirable to smooth images.

\[ G * I \]
Laplacian of **Gaussian** (LoG)

- Since second derivative (zero-crossing) is very sensitive to noise, it is desirable to smooth images

\[
\nabla^2 (G \ast I)
\]

\(G \ast I\):
- Gaussian Kernel
- Image
Laplacian of Gaussian (LoG)

- Since second derivative (zero-crossing) is very sensitive to noise, it is desirable to smooth images.
Laplacian of Gaussian (LoG)

- Only one convolution!
- Known also as “Mexican hat”

\[ I \ast \nabla^2 G \]
Laplacian of Gaussian (LoG)

• Only one convolution!
• Known also as “Mexican hat”

\[ \nabla^2 G \]
Laplacian of Gaussian (LoG)

- Only one convolution!
- Known also as “Mexican hat”

\[ = I \ast \nabla^2 G \]

\[ \nabla^2 G_\sigma(x, y) = \frac{1}{2\pi\sigma^4} \left( \frac{x^2 + y^2 - 2\sigma^2}{\sigma^2} \right) e^{-(x^2 + y^2)/2\sigma^2} \]

The laplacian operator
The laplacian operator (include diagonals)
Laplacian of **Gaussian** (LoG)

*(Credit: A.Zhang of Princeton)*
Example Filtering
Example Filtering

3x3 Laplacian
Example Filtering

3x3 LoG
Example Filtering

5x5 Laplacian

Credit: Dewald Esterhuizen
Example Filtering

5x5 LoG
Example Filtering

3x3 Sobel
Example Filtering

3x3 Prewitt
Question 1?

• What happens if we add Laplacian filtered image into original image? How does the resulting image look like?
Simple Image Enhancement

• Image gets sharpened!

\[ I + \nabla^2 I \]

Image \hspace{1cm} edges
Simple Image Enhancement

• Image gets sharpened!
Question 2?

- What happens if we subtract Gaussian filtered image from its unfiltered version? How does the resulting image look like?
Question 2

a. Radiography of the skull, b. low-pass filter with a Gaussian filter (std=15, 20 x 20), c. high-pass Filter obtained from subtracting b from a.
Another look at sharpening

\[ I = I + X - X \]
Another look at sharpening

\[ I = I + X - X \]

Let X is Smoothed image I \( \rightarrow \)

\[ X = I \ast G \]
Another look at sharpening

\[ I = I + X - X \]

Let \( X \) is Smoothed image \( I \) \( \rightarrow \)

\[ X = I \ast G \]

\[ I = I + I \ast G - I \ast G \]

Smoothed image
Another look at sharpening

\[ I = I + X - X \]

Let \( X \) is Smoothed image \( I \) \( \rightarrow \)

\[ X = I \ast G \]

\[ I = I + I \ast G - I \ast G \]

\[ I = I \ast G + I - I \ast G \]
Another look at sharpening

\[ I = I + X - X \]

Let \( X \) is Smoothed image \( I \) \( \rightarrow \)
\[ X = I \ast G \]

\[ I = I + I \ast G - I \ast G \]

Smoothed image

\[ I = I \ast G + I - I \ast G \]

Smoothed image  
Edge image
Another look at sharpening

\[ I = I \ast G + I - I \ast G \]

- Smoothed image
- Edge image

If you want to enhance the image, you need to increase the portion of the edge image.
Another look at sharpening

\[ I = I * G + I - I * G \]

- Smoothed image
- Edge image

If you want to enhance the image, you need to increase the portion of the edge image.

\[ I' = I * G + \alpha(I - I * G) \]

alpha is scalar, bigger than 1.
Another look at sharpening

\[ I = I \ast G + I - I \ast G \]

Smoothed image

Edge image

If you want to enhance the image, you need to increase the portion of the edge image.

\[ I' = I \ast G + \alpha(I - I \ast G) \]

alpha is scalar, bigger than 1.

This operation is called **UNSHARP MASKING!**
Unsharp Masking Example

Original Image  
Enhanced Image
Unsharp Masking Example 2

Original CT Data

Filtered CT Data
Unsharp Masking Example 3
Example Filtering ?
Example Filtering?

Unsharp masking
Unsharp Masking Example
Questions?