Edge Detection

Lecture 8
26-09-02

- Filtering
  - Through application of mask
- Differentiation
  - Through application of mask
- Detection
  - Through some form of Thresholding
Filtering Masks

- Mean Filter
- Gaussian Filter

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

<table>
<thead>
<tr>
<th>Mean Filter</th>
<th>1/9 x</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1</td>
<td>5 21 35 21 5</td>
</tr>
<tr>
<td>1 1 1</td>
<td>21 94 155 94 21</td>
</tr>
<tr>
<td>1 1 1</td>
<td>35 155 255 155 35</td>
</tr>
<tr>
<td>1 1 1</td>
<td>21 94 155 94 21</td>
</tr>
<tr>
<td>1 1 1</td>
<td>5 21 35 21 5</td>
</tr>
</tbody>
</table>

Derivative Masks

<table>
<thead>
<tr>
<th>Derivative Masks</th>
<th>-1 -1 -1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 0 1</td>
<td>-1 -1 -1</td>
</tr>
<tr>
<td>-1 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>-1 0 1</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Derivative Masks</th>
<th>-1 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Properties of Masks

- Filtering Masks
  - All values are +ve
  - Sum to 1
  - Output on smooth region is unchanged
  - Blurs areas of high contrast
  - Larger mask -> more smoothing

- Derivative Masks
  - opposite signs
  - Sum to zero
  - Output on smooth region is zero
  - Gives high output in areas of high contrast
  - Larger mask -> more edges detected

Derivative Masks

- Prewit Operator

\[
\begin{array}{ccc}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{ccc}
-1 & -1 & -1 \\
0 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\]
- Sobel Operator

\[
\begin{array}{ccc}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{array}
\quad \begin{array}{ccc}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1 \\
\end{array}
\]

- Robert’s Operator

\[
\begin{array}{cc}
1 & 0 \\
0 & -1 \\
\end{array}
\quad \begin{array}{cc}
0 & 1 \\
-1 & 0 \\
\end{array}
\]
First Derivative of Gaussian

- Expression?
- Effect?
- Filtering + Derivative

Laplacian of Gaussian Operator

- Proposed by Marr/Hildreth
- Gaussian Filter for noise elimination
- Problem of finding maximum
  - Normalized magnitude peaks may be flat
- Finds zero crossings of Laplacian of image
- Does not require thresholding
- Motivated by human vision system
On board, Derivation of Laplacian of Gaussian expression

Zero Crossings in 2nd Derivative

- Zero Crossings
  - Four types of zero crossings:
    - {+, -}, {+, 0, -}, {-, +}, {-, 0, +}
  - Avoid weak zero crossings
    - Ignore ones with small slope
    - Slope of zero crossing \{a, -b\} = |a + b|
LOG - Algorithm

- Generate LOG Mask for given $\sigma$
- Apply Mask to image
- Detect Zero Crossings
  - Scan along each row, record an edge point at the location of zero crossings
  - Repeat along each column

Properties of Ideal Edge Detector?

- Good Detection:
  - Low probability of failing to mark edge points
  - Low probability of falsely marking edge points
- Good Localization
  - Edge points marked should be close to center of true edge
- One response to single edge
  - Two nearby operators should not respond to the same edge (thickness)
Properties of Ideal Edge Detector?

- Conflicting Requirements
- Good Detection -> larger masks
- Good Localization -> smaller masks

Canny’s Edge Detector

- Uses first derivative Gaussian masks
- Uses Non-Maxima Suppression
- Uses Hysteresis Thresholding
Non-Maxima Suppression

- Along the gradient direction, pick the 'best' point as edge pixel
- 'best' is maximum point in gradient direction

\[
M(x, y) = \begin{cases} 
M(x, y) & \text{if } M(x, y) > M(x', y') \text{ and } M(x, y) > M(x'', y'') \\
0 & \text{otherwise}
\end{cases}
\]

Where \(M(x', y')\) and \(M(x'', y'')\) are gradient magnitudes on both sides of edge at \((x, y)\) in the gradient magnitude direction
Non-Maxima Supression

Quantization of Gradient Direction

\[ \theta = \arctan \frac{f_y}{f_x} \]
Hysteresis Thresholding

- Two thresholds, \( T_H \) and \( T_L \)
- Apply non-maxima suppression to \( M \) (gradient magnitude)
- Scan image from left to right, top to bottom
- If \( M(x,y) \) is above \( T_H \) mark it as edge
- Recursively look at neighbors; if gradient magnitude is above \( T_L \) mark it as edge
**Algorithm Summary**

- Compute gradient of image $f(x, y)$ by convolving with first derivative of Gaussian in $x$ and $y$ directions

$$f_x(x, y) = f(x, y) \ast \left( \frac{-x}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}.$$  

$$f_y(x, y) = f(x, y) \ast \left( \frac{-y}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}.$$  

**Algorithm Summary**

- Compute gradient magnitude and direction at each pixel
- Perform non-maxima suppression
  - Find gradient direction at pixel
  - Quantize it in 8 directions
    \{0, 45, 90, 135, ... 315\}
  - Compare current value of $M$ with two neighbors in appropriate direction
  - If maximum, keep it, otherwise make it zero
Algorithm Summary

- Perform Hysteresis thresholding
  - Scan image from left to right, top to bottom
  - If $M(x,y)$ is above $T_H$ mark it as edge
  - Recursively look at neighbors; if gradient magnitude is above $T_L$ mark it as edge

Program 2

- Implementation of Canny’s Edge Detector
- Main Function
- Gradient Magnitude / Direction function
- Suppress_NonMaxima function
- Edge_Detect Function
- Follow_Edge (Recursive)
Program 2

- Due Next Thursday (3-10-02)
- Should save output at each stage
  - Write out generated mask in a text file
  - Show $f_x, f_y$ images
  - Show gradient magnitude image (scale appropriately)
  - Show quantized gradient direction (use 8 colors or 4 colors)
  - Show non-maxima suppressed $M$
  - Show final edge output
  - Report on choices of your parameters