Image Gradients and Gradient Filtering

16-385 Computer Vision
What is an image edge?
Recall that an image is a 2D function

$f(x)$
How would you detect an edge?
What kinds of filter would you use?
The ‘Sobel’ filter

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

a derivative filter
(with some smoothing)

*Filter returns large response on vertical or horizontal lines?*
The ‘Sobel’ filter

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

a derivative filter
(with some smoothing)

Filter returns large response on vertical or horizontal lines?

Is the output always positive?
The ‘Sobel’ filter

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

a derivative filter
(with some smoothing)

Responds to horizontal lines

Output can be positive or negative
Output of which Sobel filter?

How do you visualize negative derivatives/gradients?
Derivative in X direction
Visualize with scaled absolute value
Derivative in Y direction
The ‘Sobel’ filter

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

*Where does this filter come?*
Do you remember this from high school?

\[ f'(x) = \lim_{h \to 0} \frac{f(x + h) - f(x)}{h} \]
Do you remember this from high school?

The derivative of a function \( f \) at a point \( x \) is defined by the limit

\[
f'(x) = \lim_{h \to 0} \frac{f(x + h) - f(x)}{h}
\]

Approximation of the derivative when \( h \) is small
This definition is based on the ‘forward difference’ but ...
it turns out that using the ‘central difference’ is more accurate

\[ f'(x) = \lim_{h \to 0} \frac{f(x + 0.5h) - f(x - 0.5h)}{h} \]

How do we compute the derivative of a discrete signal?

| 10 | 20 | 10 | 200 | 210 | 250 | 250 |
it turns out that using the ‘central difference’ is more accurate

$$f'(x) = \lim_{h \to 0} \frac{f(x + 0.5h) - f(x - 0.5h)}{h}$$

How do we compute the derivative of a discrete signal?

| 10 | 20 | 10 | 200 | 210 | 250 | 250 |

\[ f'(x) = \frac{f(x + 1) - f(x - 1)}{2} = \frac{210 - 10}{2} = 100 \]
Decomposing the Sobel filter

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

Sobel  
What this?
Decomposing the Sobel filter

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

(weighted average and scaling)
Decomposing the Sobel filter

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

Sobel

What this?

weighted average and scaling
Decomposing the Sobel filter

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

= \[
\begin{array}{c}
1 \\
2 \\
1 \\
\end{array}
\]

weighted average
and scaling

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

x-derivative
The Sobel filter only returns the x and y edge responses.

How can you compute the image gradient?
How do you compute the image gradient?

Choose a derivative filter

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

What is this filter called?

Run filter over image

$$\frac{\partial f}{\partial x} = S_x \otimes f \quad \frac{\partial f}{\partial y} = S_y \otimes f$$

What are the dimensions?

Image gradient

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

What are the dimensions?
Matching that Gradient!

(a) \[ \nabla f = \begin{bmatrix} 0, \frac{\partial f}{\partial y} \end{bmatrix} \]

(b) \[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, 0 \end{bmatrix} \]

(c) \[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \end{bmatrix} \]
Image Gradient

\[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, 0 \end{bmatrix} \]

Gradient in x only

\[ \nabla f = \begin{bmatrix} 0, \frac{\partial f}{\partial y} \end{bmatrix} \]

Gradient in y only

\[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \end{bmatrix} \]

Gradient in both x and y

Gradient direction

? 

Gradient magnitude

?
Image Gradient

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

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

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

Gradient direction
\[ \theta = \tan^{-1} \left( \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right) \]

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

How does the gradient direction relate to the edge?
What does a large magnitude look like in the image?
Common ‘derivative’ filters

<table>
<thead>
<tr>
<th></th>
<th>Sobel</th>
<th>Prewitt</th>
<th>Roberts</th>
<th>Scharr</th>
<th>Schar</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 0 -1</td>
<td>1 0 -1</td>
<td>1 0 -1</td>
<td>3 0 -3</td>
<td>3 1 0</td>
</tr>
<tr>
<td></td>
<td>2 0 -2</td>
<td>1 0 -1</td>
<td>1 0 -1</td>
<td>10 0 -10</td>
<td>10 0 0</td>
</tr>
<tr>
<td></td>
<td>1 0 -1</td>
<td>-1 0 -1</td>
<td>1 0 -1</td>
<td>3 0 -3</td>
<td>-3 -10 -3</td>
</tr>
<tr>
<td></td>
<td>1 0 -1</td>
<td>0 0 0</td>
<td>-1 0 -1</td>
<td>0 0 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 -1</td>
<td>-1 -1 -1</td>
<td>-1 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 -1</td>
<td></td>
<td>-1 -1 -1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How do you find the edge from this signal?

Intensity plot
How do you find the edge from this signal?

Intensity plot

Use a derivative filter!
How do you find the edge from this signal?

Intensity plot

Use a derivative filter!

Derivative plot

What happened?
How do you find the edge from this signal?

Use a derivative filter!

Derivative filters are sensitive to noise
Don’t forget to smooth before running derivative filters!
Laplace filter
A.K.A. Laplacian, Laplacian of Gaussian (LoG), Marr filter, Mexican Hat Function
Laplace filter
A.K.A. Laplacian, Laplacian of Gaussian (LoG), Marr filter, Mexican Hat Function
Laplace filter
A.K.A. Laplacian, Laplacian of Gaussian (LoG), Marr filter, Mexican Hat Function
first-order finite difference

$$f'(x) = \lim_{h \to 0} \frac{f(x + 0.5h) - f(x - 0.5h)}{h}$$

derivative filter

1 0 -1

second-order finite difference

$$f''(x) \approx \frac{\delta_h^2[f](x)}{h^2} = \frac{f(x + h) - 2f(x) + f(x - h)}{h^2}.$$ 

Laplace filter

? ? ?
first-order finite difference

\[ f'(x) = \lim_{{h \to 0}} \frac{f(x + 0.5h) - f(x - 0.5h)}{h} \]

derivative filter

second-order finite difference

\[ f''(x) \approx \frac{\delta^2_h[f](x)}{h^2} = \frac{f(x + h) - 2f(x) + f(x - h)}{h^2} . \]

Laplace filter

\[ 1 \ -2 \ 1 \]
Zero crossings are more accurate at localizing edges
Second derivative is noisy
2D Laplace filter

1D Laplace filter

2D Laplace filter
2D Laplace filter

1D Laplace filter

2D Laplace filter

hint
If the Sobel filter approximates the first derivative, the Laplace filter approximates ....?
Laplace filter

with smoothing

Laplace filter

without smoothing
What’s different between the two results?
Zero crossings are more accurate at localizing edges (but not very convenient)
2D Gaussian Filters

\[ h_\sigma(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}} \]

\[ \frac{\partial}{\partial x} h_\sigma(u, v) \]

\[ \nabla^2 h_\sigma(u, v) \]