Lecture 10: Edge Detection

( unless otherwise stated slides are taken or adopted from Bob Woodham, Jim Little, Leon Sigal, and Fred Tung )
Menu for Today (February 1, 20)

Topics:
— Estimating Derivatives
— Edge Detection

Readings:
— Today’s Lecture: Forsyth & Ponce (2nd ed.) 5.1 - 5.2
— Next Lecture: Forsyth & Ponce (2nd ed.) 5.3.0 - 5.3.1

Reminders:
— Assignment 2: Face Detection in a Scaled Representation is out
Today’s “fun” Example: Rainbow Illusion
Today’s “fun” Example: Lilac Chaser (a.k.a. Pac-Man) Illusion
Lecture 9: Re-cap

**Template matching** as (normalized) correlation

Template matching is **not robust** to changes in
— 2D spatial scale and 2D orientation
— 3D pose and viewing direction
— illumination

**Scaled representations** facilitate:
— template matching at multiple scales
— efficient search for image-to-image correspondences
— image analysis at multiple levels of detail

A **Gaussian pyramid** reduces artifacts introduced when sub-sampling to coarser scales
Lecture 9: Re-cap

A (discrete) approximation is

\[
\frac{\partial f}{\partial x} \approx \frac{F(X + 1, y) - F(x, y)}{\Delta x}
\]

— “First forward difference”
— Can be implemented as a convolution
— Sensitive to noise: typically smooth the image prior to derivative estimation
Lecture 9: Re-cap

**Derivative** in Y (i.e., vertical) direction

Forsyth & Ponce (1st ed.) Figure 7.4 (top left & top middle)
Lecture 9: Re-cap

**Derivative** in X (i.e., horizontal) direction

Forsyth & Ponce (1st ed.) Figure 7.4 (top left & top right)

NB
Derivatives in the Figure are negated And scaled Range -255 to 255 So /2 and offset by 128
A Short Exercise

Use the “first forward difference" to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)

\[
\begin{array}{cccc}
1 & 1 & 0.6 & 0.3 & 0 & 0 \\
1 & 1 & 0.6 & 0.3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)
A Short **Exercise**: Derivative in X Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)
A Sort Exercise: Derivative in X Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)

```
1 1 0.6 0.3 0 0
1 1 0.6 0.3 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```
A Sort **Exercise**: Derivative in X Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>0.6</th>
<th>0.3</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.6</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.6</td>
<td>0.3</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ \frac{\partial f}{\partial x} \]

\[ \frac{\partial f}{\partial y} \]
A Sort Exercise: Derivative in X Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)
A Sort Exercise: Derivative in X Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)

\[
\begin{array}{cccccc}
1 & 1 & 0.6 & 0.3 & 0 & 0 \\
1 & 1 & 0.6 & 0.3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
A Sort Exercise: Derivative in X Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)
A Sort **Exercise**: Derivative in X Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.6</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.6</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-0.4</td>
<td>-0.3</td>
<td>-0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>-0.4</td>
<td>-0.3</td>
<td>-0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
A Short Exercise: Derivative in Y Direction

Use the “first forward difference" to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)
A Short **Exercise**: Derivative in Y Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)
A Short **Exercise**: Derivative in Y Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of \( \frac{\partial f}{\partial x} \) values and one of \( \frac{\partial f}{\partial y} \) values.)
A Sort **Exercise**: Derivative in Y Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>0.6</th>
<th>0.3</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.6</td>
<td>0.3</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\begin{array}{ccc}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{array}
\]
A Sort Exercise: Derivative in Y Direction

Use the “first forward difference” to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)

\[
\begin{array}{cccccc}
1 & 1 & 0.6 & 0.3 & 0 & 0 \\
1 & 1 & 0.6 & 0.3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
Estimating Derivatives

**Question**: Why, in general, should the weights of a filter used for differentiation sum to 0?
Estimating **Derivatives**

**Question**: Why, in general, should the weights of a filter used for differentiation sum to 0?

**Answer**: Think of a constant image, $I(X, Y) = k$: The derivative is 0. Therefore, the weights of any filter used for differentiation need to sum to 0.
Estimating Derivatives

**Question:** Why, in general, should the weights of a filter used for differentiation sum to 0?

**Answer:** Think of a constant image, $I(X, Y) = k$. The derivative is 0. Therefore, the weights of any filter used for differentiation need to sum to 0.

\[
\sum_{i=1}^{N} f_i \cdot k = k \sum_{i=1}^{N} f_i = 0 \implies \sum_{i=1}^{N} f_i = 0
\]
Edge Detection

**Goal**: Identify sudden changes in image intensity

This is where most shape information is encoded

**Example**: artist’s line drawing (but artist also is using object-level knowledge)
What Causes **Edges**?

- Depth discontinuity
- Surface orientation discontinuity
- Reflectance discontinuity (i.e., change in surface material properties)
- Illumination discontinuity (e.g., shadow)

*Slide Credit: Christopher Rasmussen*
Smoothing and Differentiation

**Edge**: a location with high gradient (derivative)

Need smoothing to reduce noise prior to taking derivative

Need two derivatives, in $x$ and $y$ direction

We can use **derivative of Gaussian** filters

— because differentiation is convolution, and

— convolution is associative

Let $\otimes$ denote convolution

\[
D \otimes (G \otimes I(X, Y)) = (D \otimes G) \otimes I(X, Y)
\]
1D Example

Let's consider a row of pixels in an image:

\[ I(X, 245) \]

Where is the edge?
1D Example: Derivative

Let's consider a row of pixels in an image:

\( I(X, 245) \)

\( \frac{\partial I(X, 245)}{\partial x} \)

Where is the edge?
1D Example: Smoothing + Derivative

Let's consider a row of pixels in an image:

\[ I(X, 245) \]

\[ G \]

\[ G \otimes I(X, Y) \]
1D Example: Smoothing + Derivative

Lets consider a row of pixels in an image:

\[ I(X, 245) \]

\[ G \]

\[ G \otimes I(X, Y) \]

\[ \frac{\partial G \otimes I(X, Y)}{\partial x} \]
1D Example: Smoothing + Derivative (efficient)

Let's consider a row of pixels in an image:

\[ I(X, 245) \]

\[ \frac{\partial G}{\partial x} \]

\[ \frac{\partial G}{\partial x} \otimes I(X, Y) \]
Partial Derivatives of Gaussian

Slide Credit: Christopher Rasmussen
Gradient Magnitude

Let $I(X, Y)$ be a (digital) image.

Let $I_x(X, Y)$ and $I_y(X, Y)$ be estimates of the partial derivatives in the $x$ and $y$ directions, respectively.

Call these estimates $I_x$ and $I_y$ (for short). The vector $[I_x, I_y]$ is the gradient.

The scalar $\sqrt{I_x^2 + I_y^2}$ is the gradient magnitude.
Image **Gradient**

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

\[ \nabla f = \left[ \frac{\partial f}{\partial x}, 0 \right] \hspace{1cm} \nabla f = \left[ 0, \frac{\partial f}{\partial y} \right] \]
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 of intensity:
The gradient of an image: $\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$

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

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

(how is this related to the direction of the edge?)
Image Gradient

The gradient of an image:

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

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

The gradient direction is given by:

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

(how is this related to the direction of the edge?)

The edge strength is given by the gradient magnitude:

\[ \| \nabla f \| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \]
Increased **smoothing**:
— eliminates noise edges
— makes edges smoother and thicker
— removes fine detail
Sobel Edge Detector

1. Use **central differencing** to compute gradient image (instead of first forward differencing). This is more accurate.

2. **Threshold** to obtain edges

   ![Images](Original Image, Sobel Gradient, Sobel Edges)

   *Thresholds are brittle, we can do better!*