Lecture 11: Edge Detection (cont.)

( unless otherwise stated slides are taken or adopted from Bob Woodham, Jim Little, Leon Sigal and Fred Tung )
Menu for Today (February 3)

Topics:
— Edge Detection
— Marr / Hildreth and Canny Edges
— Image Boundaries

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  February 10
Today’s “fun” Example: Motion Illusion
Today’s “fun” Example: Rotating Snakes Illusion
Lecture 10: Re-cap

Physical properties of a 3D scene cause “edges” in an image:
— depth discontinuity
— surface orientation discontinuity
— reflectance discontinuity
— illumination boundaries (cast shadow, self shadow)
**Lecture 10: Re-cap**

**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

\[ D \otimes (G \otimes I(X, Y)) = (D \otimes G) \otimes I(X, Y) \]
Lecture 10: Re-cap

The gradient of an image: \[ \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x}, & 0 \end{bmatrix} \]

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

- Original Image
- **Sobel** Gradient
- **Sobel** Edges

**Thresholds are brittle, we can do better!**
Human vision ...

Simple cells:
Response to light orientation

Complex cells:
Response to light orientation and movement

Hypercomplex cells:
Response to movement with end point

Stimulus

Response

Hubel & Wiesel, 1959

Electrical signal from brain

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, cs231n Stanford
David Marr, 1970s

VISION

David Marr

FOREWORD BY Shimon Ulman
AFTERWORD BY Tomaso Poggio

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, cs231n Stanford
David Marr, 1970s

[Stages of Visual Representation, David Marr]

* slide from Fei-Fei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Two generic approaches to **edge point detection**:
— (significant) local extrema of a first derivative operator
— zero crossings of a second derivative operator
Marr / Hildreth Laplacian of Gaussian

A “zero crossings of a second derivative operator” approach

Design Criteria:

1. localization in space
2. localization in frequency
3. rotationally invariant
Marr / Hildreth **Laplacian of Gaussian**

A “zero crossings” of a second derivative operator” approach

**Steps:**

1. Gaussian for smoothing

2. Laplacian ($\nabla^2$) for differentiation where

\[ \nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \]

Combining 1. and 2. we get:

3. Locate zero-crossings in the Laplacian of the Gaussian ($\nabla^2 G$) where

\[ \nabla^2 G(x, y) = \frac{-1}{2\pi\sigma^4} \left[ 2 - \frac{x^2 + y^2}{\sigma^2} \right] \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \]
Marr / Hildreth **Laplacian of Gaussian**

Here’s a 3D plot of the Laplacian of the Gaussian ($\nabla^2 G$)...

... with its characteristic “Mexican hat” shape
1D Example: Continued

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

\[ I(X, 245) \]

\[ \nabla^2 G \]

\[ \nabla^2 G \otimes I(X, Y) \]

Where is the edge? Zero-crossings of bottom graph
Marr / Hildreth **Laplacian of Gaussian**

### 5 x 5 LoG filter

\[
\begin{array}{ccccc}
0 & 0 & -1 & 0 & 0 \\
0 & -1 & -2 & -1 & 0 \\
-1 & -2 & 16 & -2 & -1 \\
0 & -1 & -2 & -1 & 0 \\
0 & 0 & -1 & 0 & 0
\end{array}
\]

### 17 x 17 LoG filter

\[
\begin{array}{cccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & 0 & 0 & 0 \\
0 & 0 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & 0 & 0 \\
0 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & -1 \\
0 & 0 & -2 & 0 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 \\
0 & 0 & -1 & -1 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 \\
0 & 0 & -1 & 0 & -1 & -1 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\end{array}
\]

**Scale (σ)**

*Image From: A. Campilho*
Marr / Hildreth **Laplacian of Gaussian**

- Original Image
- LoG Filter
- Zero Crossings

Scale ($\sigma$)

*Image From: A. Campilho*
Assignment 1: High Frequency Image

original

smoothed (5x5 Gaussian)

original - smoothed (scaled by 4, offset +128)
Assignment 1: High Frequency Image

original - smoothed (5x5 Gaussian) = smoothed - original (scaled by 4, offset +128)
Assignment 1: High Frequency Image

Gaussian \rightarrow \text{delta function} \sim \text{Laplacian of Gaussian}
Canny Edge Detector

A “local extrema of a first derivative operator” approach

Design Criteria:

1. good detection
   — low error rate for omissions (missed edges)
   — low error rate for commissions (false positive)

2. good localization

3. one (single) response to a given edge
   — (i.e., eliminate multiple responses to a single edge)
Example: Edge Detection

**Question**: How many edges are there?

**Question**: What is the position of each edge?
Example: Edge Detection

**Question**: How many edges are there?

**Question**: What is the position of each edge?
**Example**: Edge Detection

**Question**: How many edges are there?

**Question**: What is the position of each edge?
Canny Edge Detector

Steps:

1. Apply directional derivatives of Gaussian

2. Compute gradient magnitude and gradient direction

3. Non-maximum suppression
   — thin multi-pixel wide “ridges” down to single pixel width

4. Linking and thresholding
   — Low, high edge-strength thresholds
   — Accept all edges over low threshold that are connected to edge over high threshold
Non-maximum Suppression

**Idea:** suppress near-by similar detections to obtain one “true” result
Non-maximum Suppression

Idea: suppress near-by similar detections to obtain one “true” result
Non-maximum Suppression

**Idea:** suppress near-by similar detections to obtain one “true” result

**Detected template**

**Correlation map**

*Slide Credit:* Kristen Grauman
Non-maximum Suppression

Forsyth & Ponce (1st ed.) Figure 8.11

Select the image maximum point across the width of the edge
Non-maximum Suppression

Value at $q$ must be larger than interpolated values at $p$ and $r$

Forsyth & Ponce (2nd ed.) Figure 5.5 left
Non-maximum Suppression

Value at $q$ must be larger than interpolated values at $p$ and $r$.
Example: Non-maximum Suppression

Original Image  Gradient Magnitude  Non-maximum Suppression

Slide Credit: Christopher Rasmussen
Example

Forsyth & Ponce (1st ed.) Figure 8.13 top
Example

Forsyth & Ponce (1st ed.) Figure 8.13 top

Figure 8.13 bottom left
Fine scale ($\sigma = 1$), high threshold
Example

Forsyth & Ponce (1st ed.) Figure 8.13 top

Figure 8.13 bottom middle
Fine scale ($\sigma = 4$), high threshold
Example

Forsyth & Ponce (1st ed.) Figure 8.13 top

Figure 8.13 bottom right

Fine scale ($\sigma = 4$), low threshold
Assume the marked point is an **edge point**. Take the normal to the gradient at that point and use this to predict continuation points (either \( r \) or \( s \))
**Edge Hysteresis**

One way to deal with broken edge chains is to use hysteresis

**Hysteresis**: A lag or momentum factor

**Idea**: Maintain two thresholds $k_{\text{high}}$ and $k_{\text{low}}$

— Use $k_{\text{high}}$ to find strong edges to start edge chain
— Use $k_{\text{low}}$ to find weak edges which continue edge chain

Typical ratio of thresholds is (roughly):

$$\frac{k_{\text{high}}}{k_{\text{low}}} = 2$$
**Canny** Edge Detector

**Original Image**

**Strong Edges**

**Strong + connected Weak Edges**

**Weak Edges**
Comparing **Edge** Detectors
Comparing **Edge** Detectors

**Good detection**: minimize probability of false positives/negatives (spurious/missing) edges

**Good localization**: found edges should be as close to true image edge as possible

**Single response**: minimize the number of edge pixels around a single edge
Comparing **Edge Detectors**

**Good detection**: minimize probability of false positives/negatives (spurious/missing) edges

**Good localization**: found edges should be as close to true image edge as possible

**Single response**: minimize the number of edge pixels around a single edge

<table>
<thead>
<tr>
<th>Approach</th>
<th>Detection</th>
<th>Localization</th>
<th>Single Resp</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sobel</strong></td>
<td>Good</td>
<td>Poor</td>
<td>Poor</td>
<td>Results in Thick Edges</td>
</tr>
<tr>
<td>Gradient Magnitude Threshold</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Marr / Hildreth</strong></td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Smooths Corners</td>
</tr>
<tr>
<td>Zero-crossings of 2nd Derivative (LoG)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Canny</strong></td>
<td>Best</td>
<td>Good</td>
<td>Good</td>
<td></td>
</tr>
<tr>
<td>Local extrema of 1st Derivative</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

48
Lecture 12: Laplacian Pyramids (aside for HW2)
Gaussian Pyramid

What happens to the details?
— They get smoothed out as we move to higher levels

What is preserved at the higher levels?
— Mostly large uniform regions in the original image

How would you reconstruct the original image from the image at the upper level?
— That’s not possible

Forsyth & Ponce (2nd ed.) Figure 4.17

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Laplacian Pyramid

Building a Laplacian pyramid:
— Create a Gaussian pyramid
— Take the difference between one Gaussian pyramid level and the next (before subsampling) - the residual

Properties
— Also known as the difference-of-Gaussian (DOG) function, a close approximation to the Laplacian
— It is a band pass filter – each level represents a different band of spatial frequencies
Laplacian Pyramid

At each level, retain the residuals instead of the blurred images themselves.

Why is it called Laplacian Pyramid?

Slide Credit: Ioannis (Yannis) Gkioulkas (CMU)
Why Laplacian Pyramid?

Slide Credit: Ioannis (Yannis) Gkioulkas (CMU)
Laplacian Pyramid

At each level, retain the residuals instead of the blurred images themselves.

Why is it called Laplacian Pyramid?

Can we reconstruct the original image using the pyramid?
— Yes we can!
Laplacian Pyramid

At each level, retain the residuals instead of the blurred images themselves.

**Why is it called Laplacian Pyramid?**

Can we reconstruct the original image using the pyramid?
— Yes we can!

What do we need to store to be able to reconstruct the original image?

*Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)*
Let’s start by just looking at one level

Does this mean we need to store both residuals and the blurred copies of the original?

Slide Credit: Ioannis (Yannis) Gkioulkas (CMU)
Constructing a **Laplacian** Pyramid

**Algorithm**

```
repeat:
  filter
  compute residual
  subsample
until min resolution reached
```
Constructing a **Laplacian** Pyramid

repeat:
  filter
  compute residual
  subsample
until min resolution reached

**Algorithm**

---

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Constructing a **Laplacian Pyramid**

It’s a Gaussian Pyramid

**Algorithm**

repeat:
  filter
  compute residual
  subsample
until min resolution reached

**Slide Credit:** Ioannis (Yannis) Gkioulekas (CMU)
Reconstructing the Original Image

**Algorithm**

repeat:
  - upsample
  - sum with residual
until orig resolution reached

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Gaussian vs Laplacian Pyramid

Which one takes more space to store?

Shown in opposite order for space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)
Aside: Image Blending

Aside: Image Blending

Aside: Image Blending

**Algorithm:**

1. Build Laplacian pyramid LA and LB from images A and B

2. Build a Gaussian pyramid GR from mask image R (the mask defines which image pixels should be coming from A or B)

3. Form a combined (blended) Laplacian pyramid LS, using nodes of GR as weights: 
   \[ LS(i,j) = GR(i,j) \times LA(i,j) + (1-GR(i,j)) \times LB(i,j) \]

4. Reconstruct the final blended image from LS
Aside: Image Blending
Aside: Image Blending

© david dmartin (Boston College)
Aside: Image Blending

© Chris Cameron
How do humans perceive boundaries?

Edges are a property of the 2D image.

**It is interesting to ask**: How closely do image edges correspond to boundaries that humans perceive to be salient or significant?
How do humans perceive boundaries?

"Divide the image into some number of segments, where the segments represent 'things' or 'parts of things' in the scene. The number of segments is up to you, as it depends on the image. Something between 2 and 30 is likely to be appropriate. It is important that all of the segments have approximately equal importance." (Martin et al. 2004)
How do humans perceive **boundaries**?

*Figure Credit: Martin et al. 2001*
How do humans perceive **boundaries**?

**Figure Credit:** Martin et al. 2001
How do humans perceive boundaries?

Each image shows multiple (4-8) human-marked boundaries. Pixels are darker where more humans marked a boundary.

**Figure Credit:** Szeliski Fig. 4.31. **Original:** Martin et al. 2004
Boundary Detection

We can formulate *boundary detection* as a high-level recognition task — try to learn, from sample human-annotated images, which visual features or cues are predictive of a salient/significant boundary.

Many boundary detectors output a *probability or confidence* that a pixel is on a boundary.
Boundary Detection: Example Approach

— Consider circular windows of radii \( r \) at each pixel \((x, y)\) cut in half by an oriented line through the middle

— Compare visual features on both sides of the cut

— If features are very different on the two sides, the cut line probably corresponds to a boundary

— Notice this gives us an idea of the orientation of the boundary as well
Boundary Detection: Example Approach

— Consider circular windows of radii $r$ at each pixel $(x, y)$ cut in half by an oriented line through the middle

— Compare visual features on both sides of the cut

— If features are very **different** on the two sides, the cut line probably corresponds to a boundary

— Notice this gives us an idea of the orientation of the boundary as well

**Implementation:** consider 8 discrete orientations ($\theta$) and 3 scales ($r$)
Boundary Detection:

For each **feature** type

— Compute non-parametric distribution (histogram) for left side
— Compute non-parametric distribution (histogram) for right side
— Compare two histograms, on left and right side, using statistical test

Use all the histogram similarities as features in a learning based approach that outputs probabilities (Logistic Regression, SVM, etc.)
Boundary Detection:

Features:
— Raw Intensity
— Orientation Energy
— Brightness Gradient
— Color Gradient
— Texture gradient
**Boundary Detection:**

**Features:**
- Raw Intensity
- Orientation Energy
- Brightness Gradient BG
- Color Gradient CG
- Texture gradient TG

**Figure Credit:** Martin et al. 2004
Boundary Detection: Example Approach

Figure Credit: Szeliski Fig. 4.33. Original: Martin et al. 2004
Summary

Physical properties of a 3D scene cause “edges” in an image:
— depth discontinuity
— surface orientation discontinuity
— reflectance discontinuity
— illumination boundaries

Two generic approaches to edge detection:
— local extrema of a first derivative operator → Canny
— zero crossings of a second derivative operator → Marr/Hildreth

Many algorithms consider “boundary detection” as a high-level recognition task and output a probability or confidence that a pixel is on a human-perceived boundary