Announcements

- **Project 0 due tomorrow night**

Edge Detection

From Sandlot Science

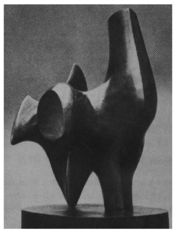
Today's readings

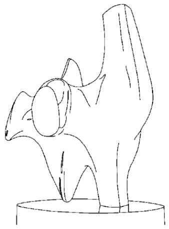
- Cipolla and Gee (handout)
  - supplemental: Forsyth, chapter 9

For Friday

- Watt, 10.3-10.4 (handout)
- Intelligent Scissors
  - http://www.cs.washington.edu/education/courses/490cv/02wi/readings/book-7-revised-a-indx.pdf

Edge detection

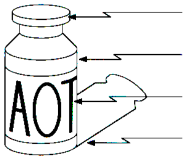




Convert a 2D image into a set of curves

- Extracts salient features of the scene
- More compact than pixels

Origin of Edges



surface normal discontinuity

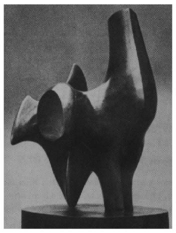
depth discontinuity

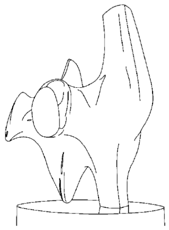
surface color discontinuity

illumination discontinuity

Edges are caused by a variety of factors.

Edge detection





How can you tell that a pixel is on an edge?

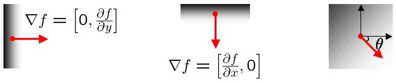
snoop demo

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 change in intensity



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

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

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

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

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

The discrete gradient

How can we differentiate a *digital* image F(x,y)?

- Option 1: reconstruct a continuous image, then take gradient
- Option 2: take discrete derivative (finite difference)

$$\frac{\partial F}{\partial x}[x,y] \approx F[x+1,y] - F[x,y]$$

How would you implement this as a cross-correlation?

H

filter demo

The Sobel operator

Better approximations of the derivatives exist

- The *Sobel* operators below are very commonly used

$$\frac{1}{8}\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad \frac{1}{8}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

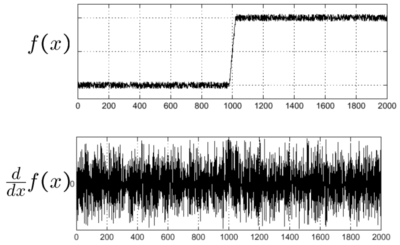
$$H_x \qquad\qquad H_y$$

- The standard defn. of the Sobel operator omits the 1/8 term
  - doesn't make a difference for edge detection
  - the 1/8 term **is** needed to get the right gradient value, however

Effects of noise

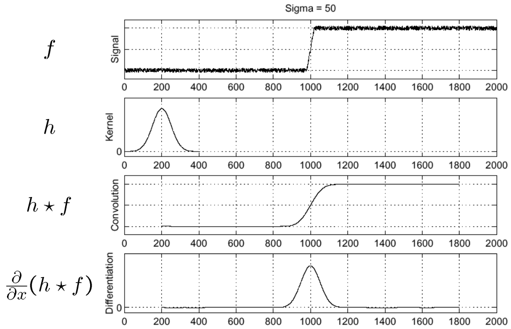
Consider a single row or column of the image

- Plotting intensity as a function of position gives a *signal*



Where is the edge?

Solution: smooth first

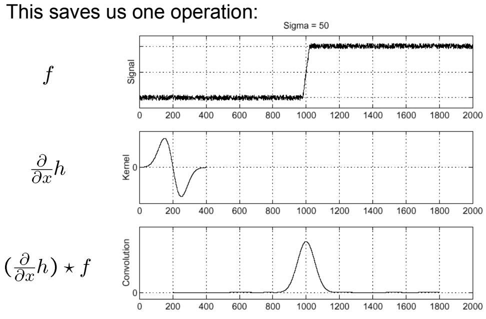


Where is the edge? Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Derivative theorem of convolution

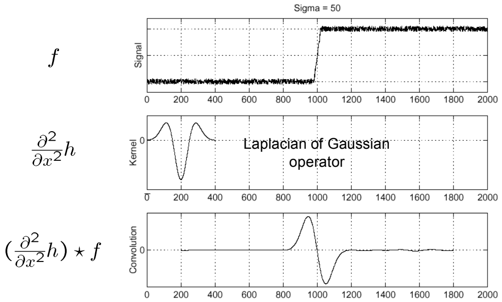
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

This saves us one operation:

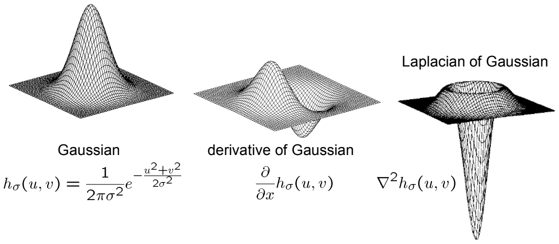


Laplacian of Gaussian

Look for zero-crossings of $\frac{\partial^2}{\partial x^2}(h \star f)$



2D edge detection filters



Laplacian of Gaussian

Gaussian

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

derivative of Gaussian

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

$\nabla^2 h_\sigma(u,v)$

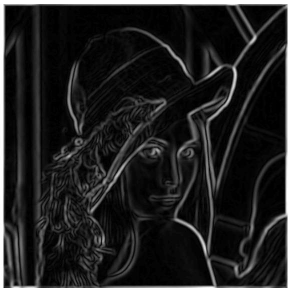
∇^2 is the **Laplacian** operator:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

The Canny edge detector

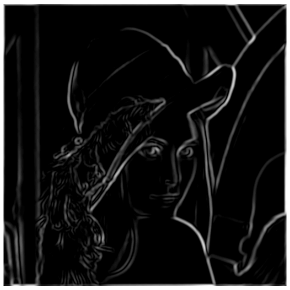
original image (Lena)

The Canny edge detector



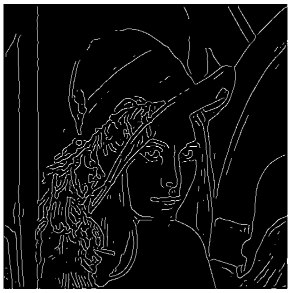
norm of the gradient

The Canny edge detector



thresholding

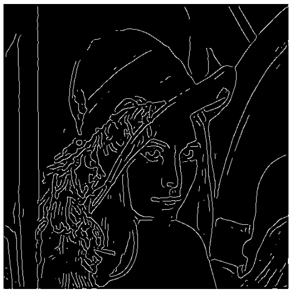
The Canny edge detector



thinning

(non-maximum suppression, edge following)

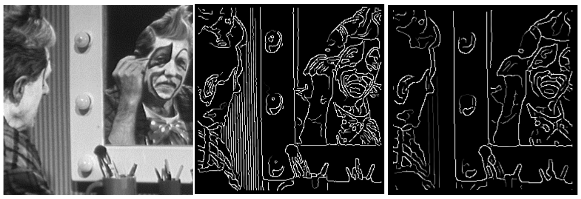
The Canny edge detector



thinning

(non-maximum suppression, edge following)

Effect of Gaussian kernel width



original

Canny with $\sigma = 1$

Canny with $\sigma = 2$

The choice of σ depends on desired behavior

- large σ detects large scale edges
- small σ detects fine features

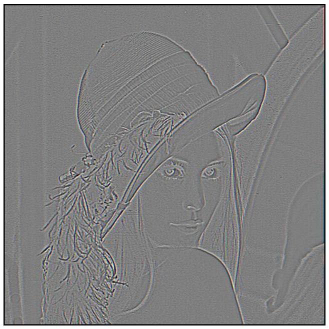
Edge detection by subtraction

original

Edge detection by subtraction

smoothed (5x5 Gaussian)

Edge detection by subtraction

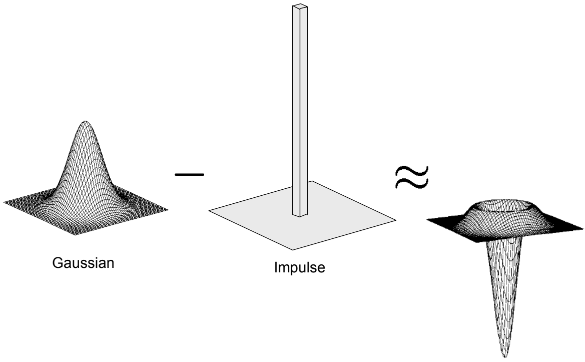


smoothed – original

(scaled by 4, offset +128)

Why does this work?

Gaussian - subtraction filter



Summary

Things to take away from this lecture

- What is an edge and where does it come from
- Edge detection by differentiation
- Image gradients
  - continuous and discrete
  - filters (e.g., Sobel operator)
- Effects of noise on gradients
- Derivative theorem of convolution
- Derivative of Gaussian (DoG) operator
- Laplacian operator
  - Laplacian of Gaussian (LoG)
- Canny edge detector (basic idea)
  - Effects of varying sigma parameter
- Approximating an LoG by subtraction