EEE-6512: Image Processing and Computer Vision

October 27, 2017

Lecture #8: Edges and Features

Damon L. Woodard, Ph.D.
Dept. of Electrical and Computer Engineering
dwoodard@ece.ufl.edu

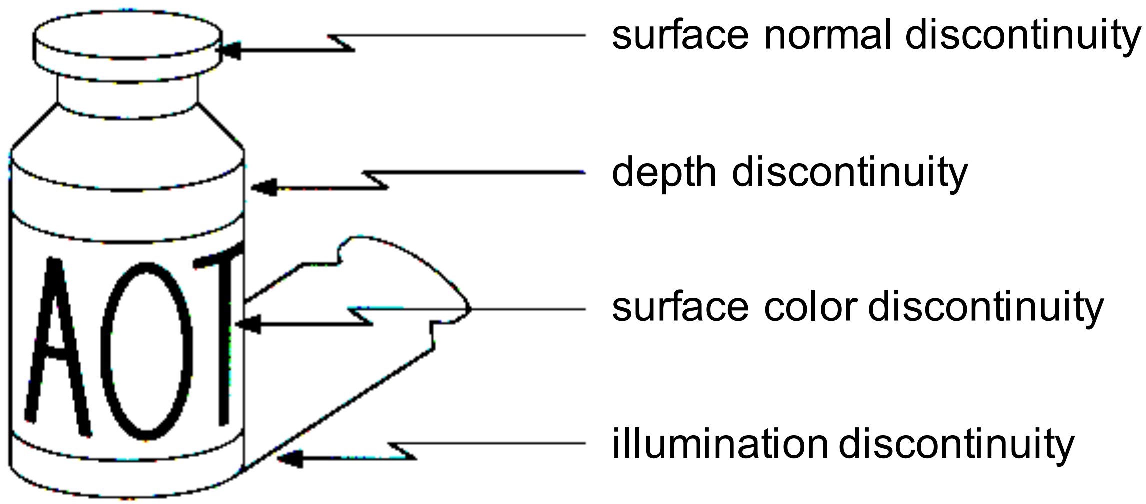
Laplacian Pyramid

- Since the Laplacian is a bandpass operator, convolving the image with a Laplacian of Gaussian (LoG) kernel with increasing variance yields the **Laplacian pyramid**.

$$L^{(i+1)}(x, y) \equiv (I^{(0)}(x, y) \circledast LoG_{(i+1)\sigma^2}(x, y)) \downarrow (i+1)d$$

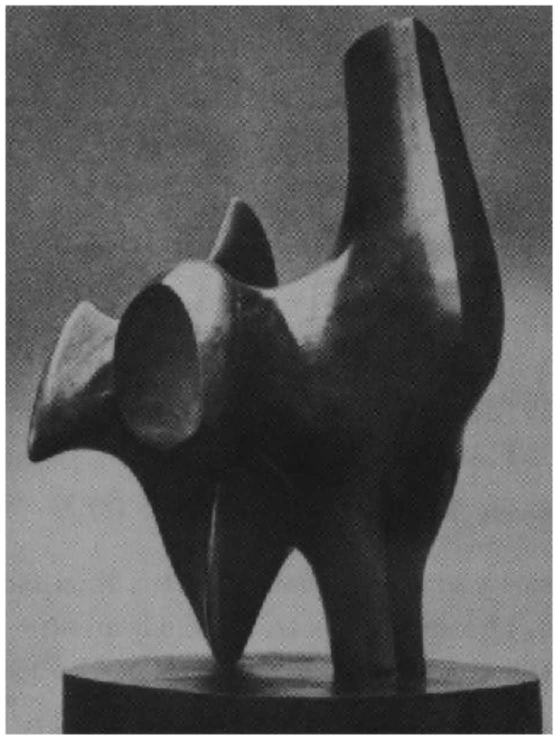
Edge Detection

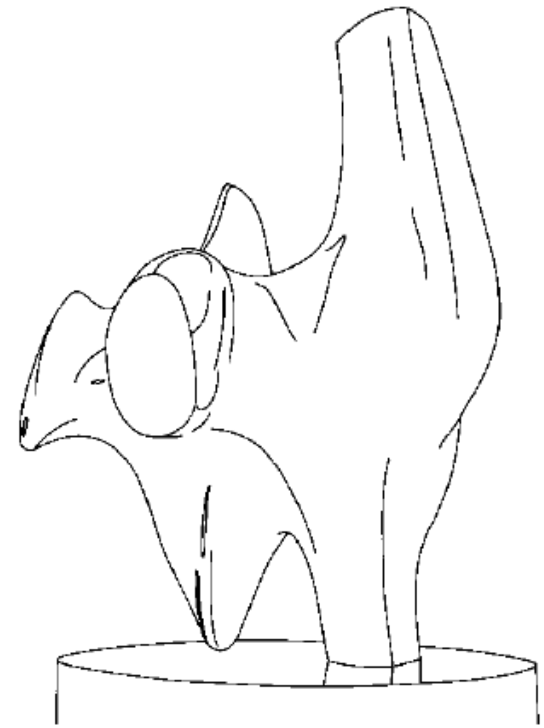
Origin of Edges



Edges are caused by a variety of factors

Edge detection

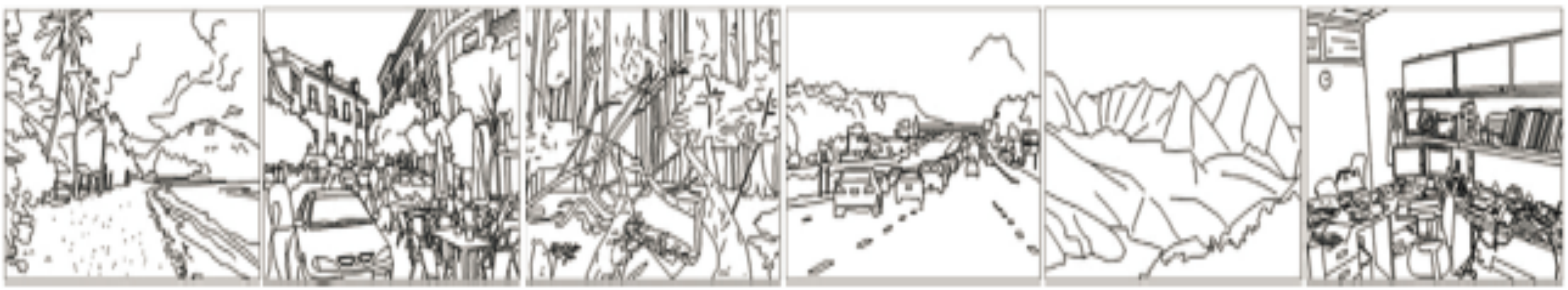




- How can you tell that a pixel is on an edge?

Edge Detection (cont'd)

Figure 7.6 Intensity edges capture a rich representation of the scene. The scenes and objects in these line drawings are, with little difficulty, recognizable by the average human viewer. From Walther et al. [2011]. For the original images, turn to Figure 7.7.



D. B. Walther, B. Chai, E. Caddigan, D. M. Beck, and L. Fei-Fei, "Simple line drawings sffice for functional MRI decoding of natural scene categories," Proceedings of the National Academy of Sciences (PNAS), 108(23):9661-9666, 2011.

D. B. Walther, B. Chai, E. Caddigan, D. M. Beck, and L. Fei-Fei, "Simple line drawings su ce for functional MRI decoding of natural scene categories," Proceedings of the National Academy of Sciences (PNAS), 108(23):9661-9666, 2011.

Figure 7.7 The original images from which the line drawings shown in Figure 7. 6 were obtained. From Walther et al. [2011].

Edge Detection

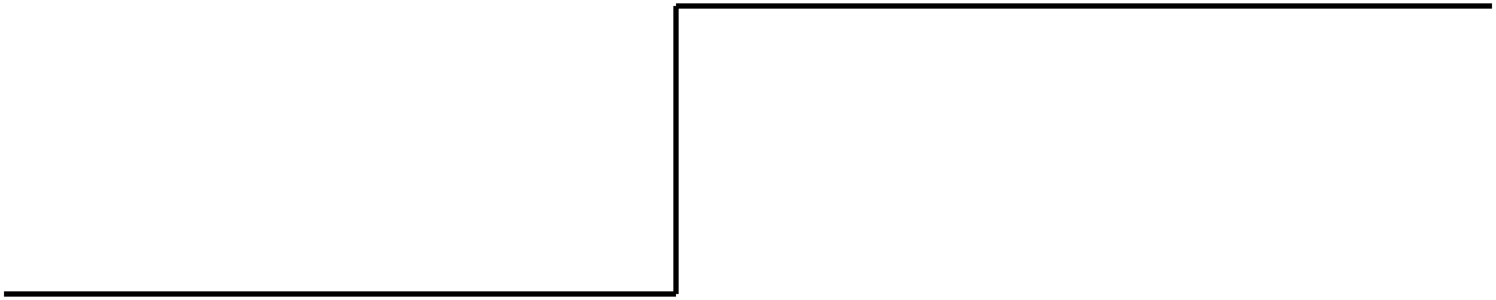
Basic idea: look for a neighborhood with strong signs of change.

Problems:

- neighborhood size
- how to detect change

Edge is Where Change Occurs

- Change is measured by derivative in 1D
- Biggest change, derivative has maximum magnitude
- Or 2^{nd} derivative is zero.



Edge Detection

- **Intensity edges** are pixels in the image where the intensity (or graylevel) function changes rapidly.
- **Step edge**: occurs when a light region is adjacent to a dark region.
- **Line edge:** occurs when a thin light (or dark) object, such as a wire, is in front of a dark (or light) background.
- **Roof edge**: the change is not in the lightness itself but rather the derivative of the lightness.
- **Ramp edge:** occurs when the lightness changes slowly across a region.

Edge Detection (cont'd)

Figure 7.8 Four types of intensity edges.

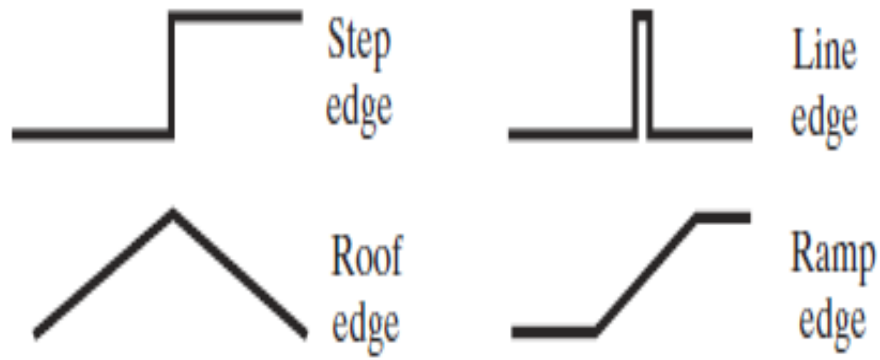
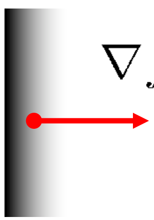


Image gradient

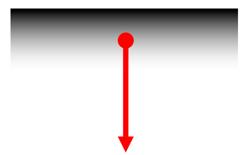
- The gradient of an image:

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

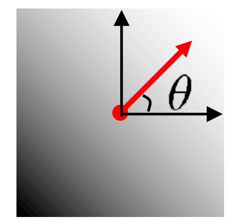
- The gradient points in the direction of most rapid change in intensity



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



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



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

The gradient direction is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \bigg/ \frac{\partial f}{\partial x}\right)$$

The *edge strength* is given by the gradient magnitude

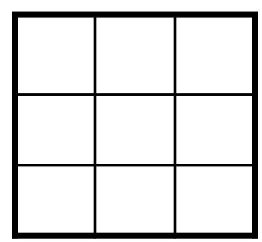
$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

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

Differential Operators

Differential operators:

- attempt to approximate the gradient at a pixel via masks

- threshold the gradient to select the edge pixels

Example: Sobel Operator

$$Sx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Sy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

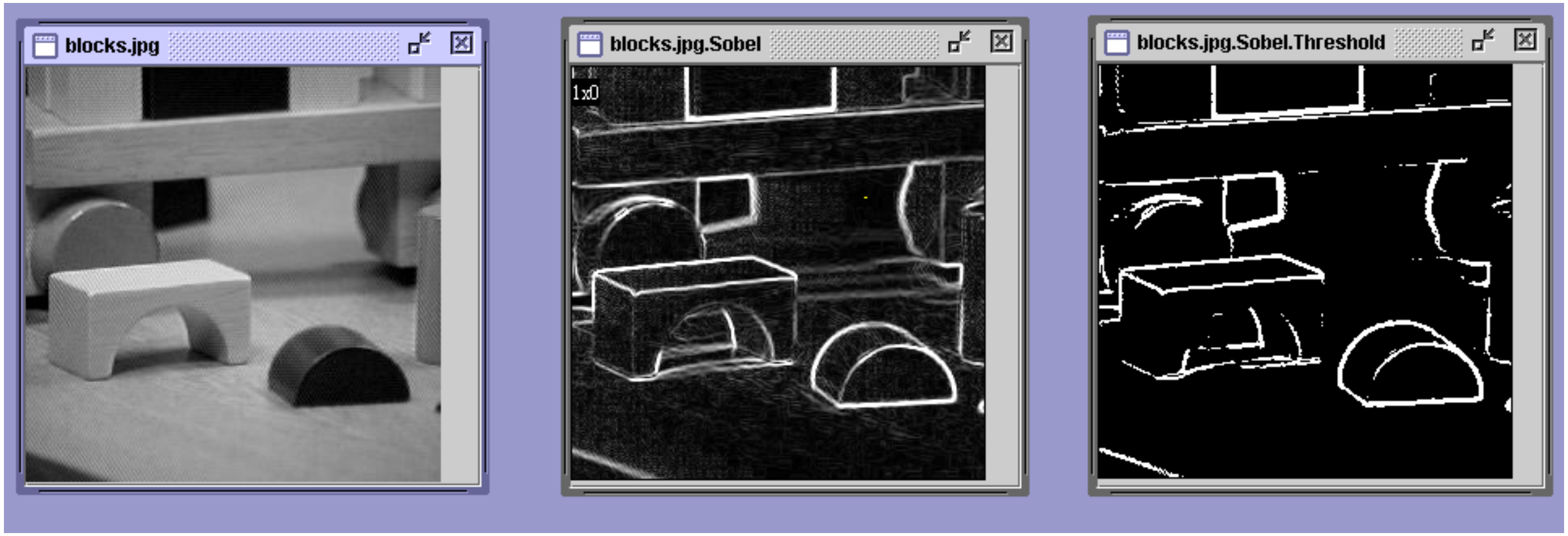
On a pixel of the image I

- let gx be the response to Sx
- let gy be the response to Sy

Then the gradient is

$$\nabla I = [gx \;\; gy]^T$$

Sobel Operator on the Blocks Image



original image

gradient magnitude

thresholded gradient magnitude

Common Masks for Computing Gradient

- Sobel:

$$S_x = \boxed{\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}} \qquad S_y = \boxed{\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}}$$

- Prewitt:

$$\boxed{\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}} \qquad \boxed{\begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}}$$

- Roberts

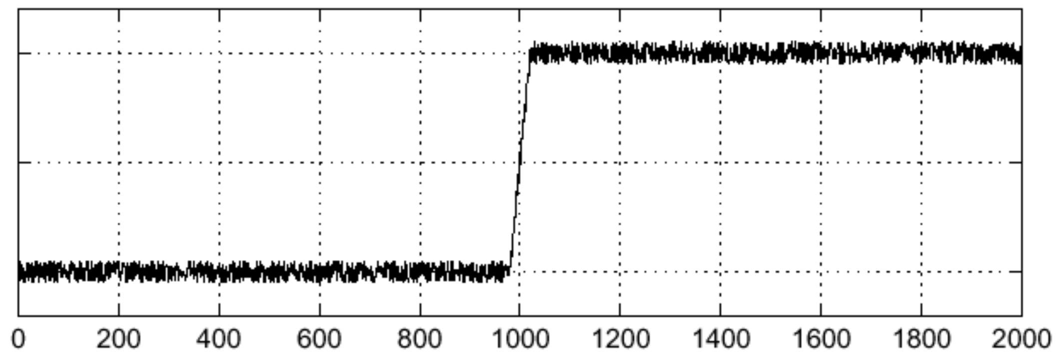
$$\boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}} \qquad \boxed{\begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}}$$

Sx **Sy**

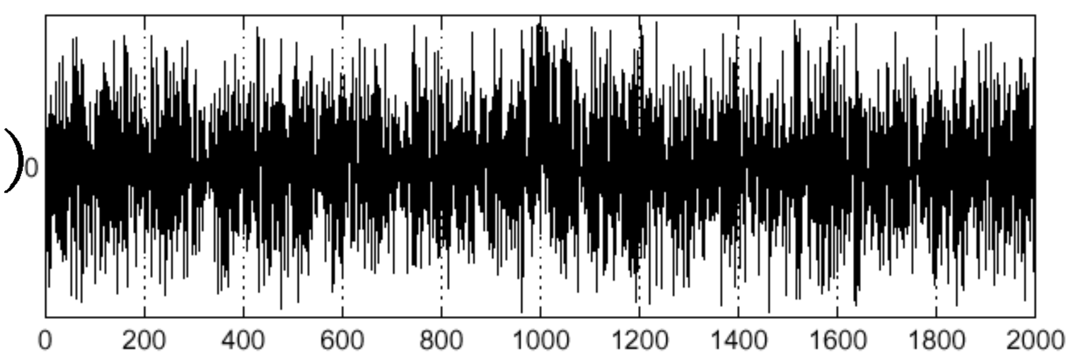
Effects of noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal

$f(x)$



$\frac{d}{dx}f(x)$



Where is the edge?

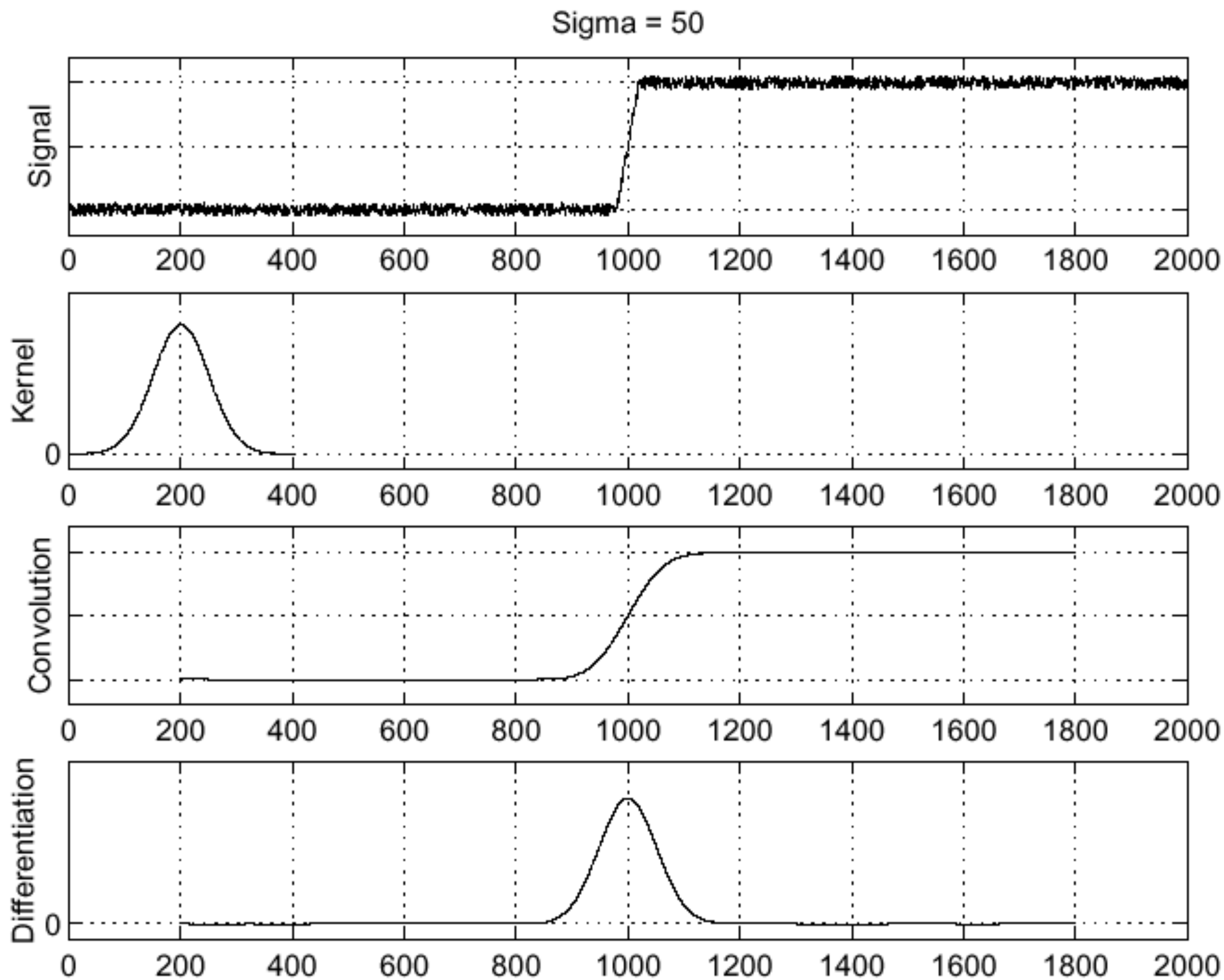
Solution: smooth first

f

h

$h \star f$

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



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

Derivative theorem of convolution

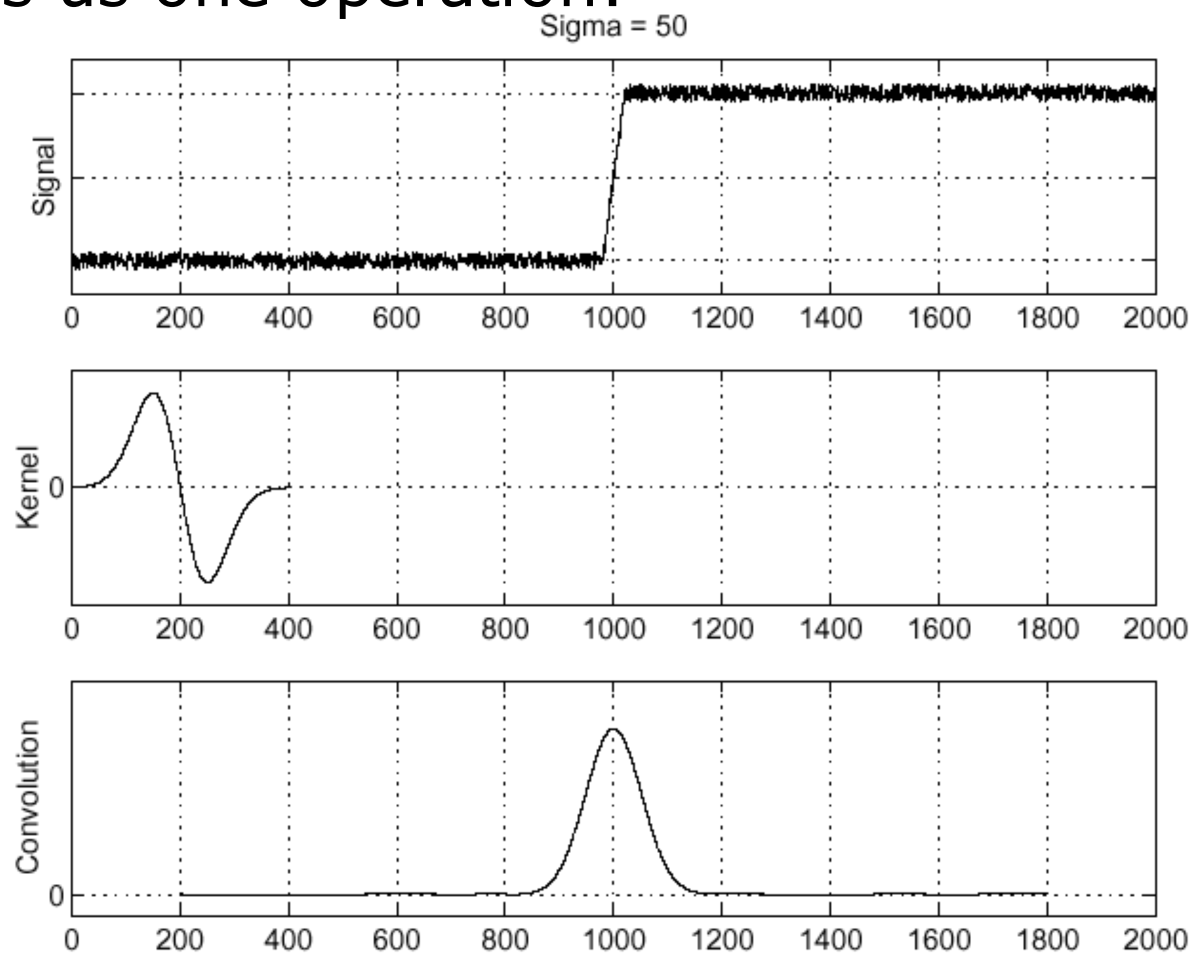
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

- This saves us one operation:

$$f$$

$$\frac{\partial}{\partial x}h$$

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

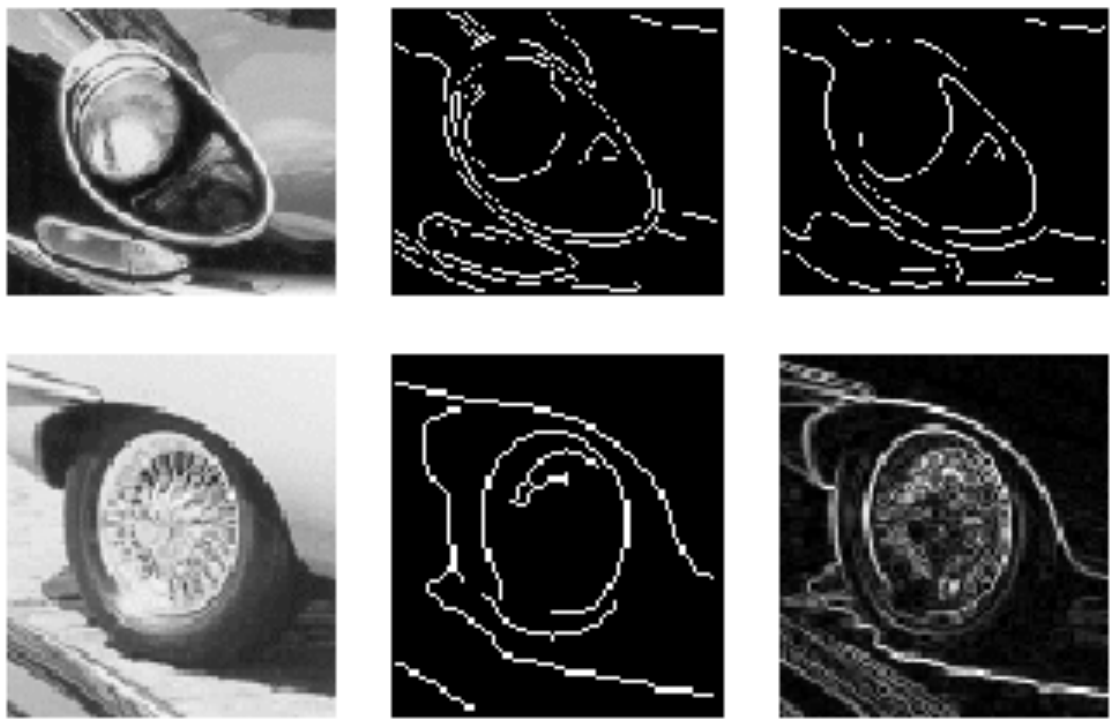


Canny Edge Detector

- Smooth the image with a Gaussian filter with spread σ.
- Compute gradient magnitude and direction at each pixel of the smoothed image.
- Zero out any pixel response $\leq$ the two neighboring pixels on either side of it, along the direction of the gradient.
- Track high-magnitude contours.
- Keep only pixels along these contours, so weak little segments go away.

Canny Examples

Canny σ=1 Canny σ=4



Canny σ=1 Roberts 2X2

Canny Characteristics

- The Canny operator gives single-pixel-wide images with good continuation between adjacent pixels

- It is the most widely used edge operator today; no one has done better since it came out in the late 80s. Many implementations are available.

- It is very sensitive to its parameters, which need to be adjusted for different application domains.

Localization-detection tradeoff

- A large sigma yields a better signal-to-noise ratio (SNR), but a smaller sigma yields a more accurate location for the edge. This dilemma is known as the **localization-detection tradeoff.**

- To derive the optimal step detector, two criteria are specified:
  - The detector should yield low false positive and false negative rates.
  - The detected edge should be close to the true edge (that is, good *localization*).

Questions?

Slide Credits

Lectures Slides adapted from slides of:

Prof. Linda Shapiro (University of Washington)

Prof. George Stockman (Michigan State)

Dr. Simon Baker (Microsoft Research)