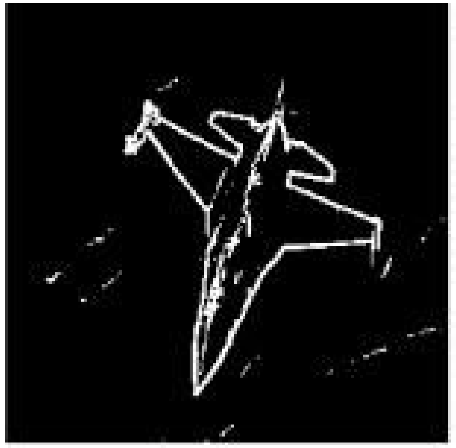
Lecture 07

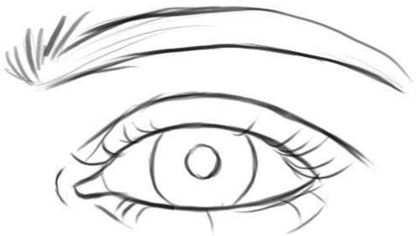
Outline

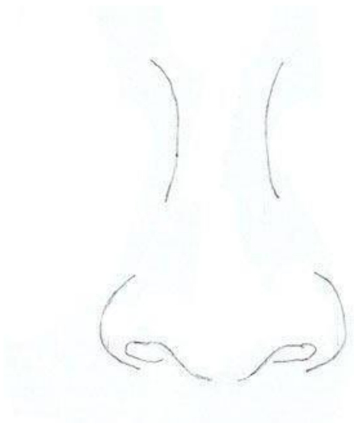
- What is edge detection?
- Why do we need edge detection?
- Challenges
  - Noise
- How to detect edges?
  - Prewit
  - Sobel
  - Laplacian
  - Canny

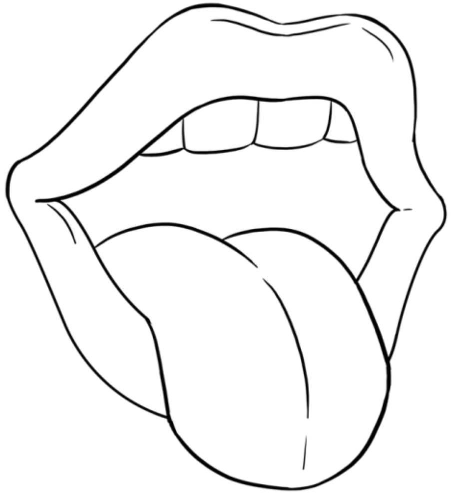
Why edge detection?

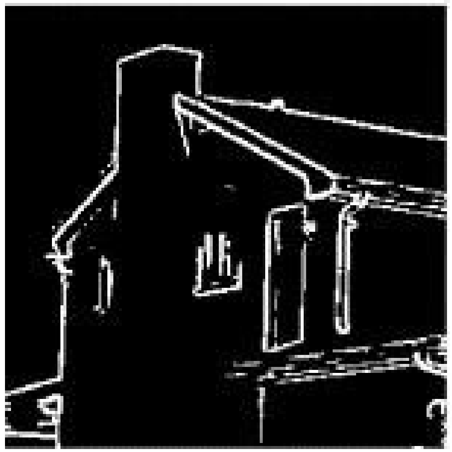
- Extract useful information from images
  - Recognizing objects









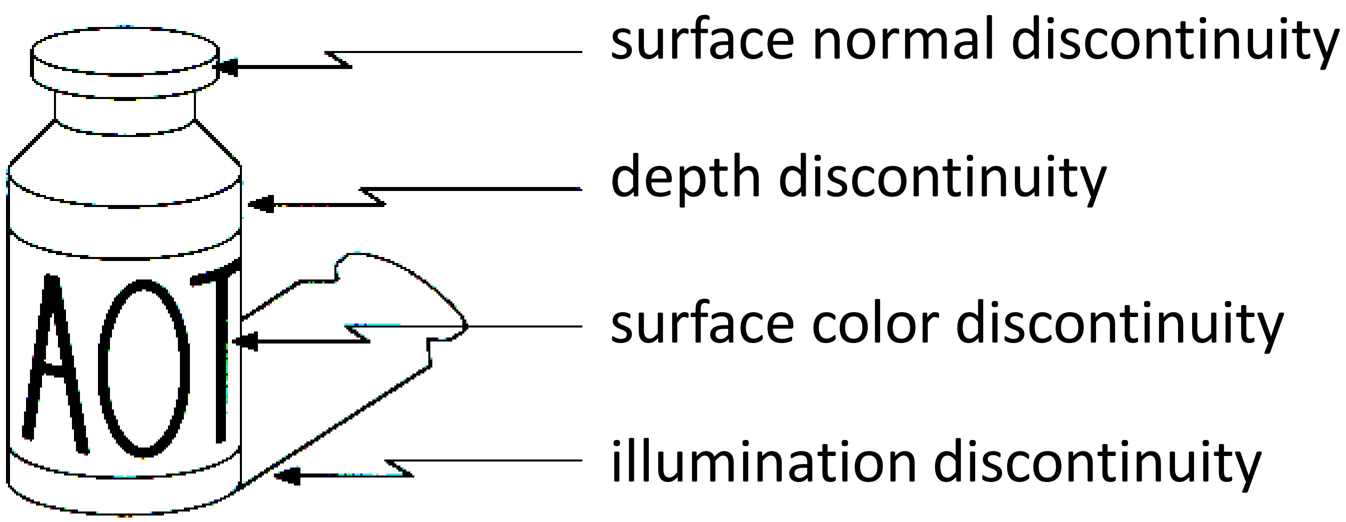


Edge Detection

- Identify sudden changes in an image
  - Semantic and shaped information
  - Marks the border of an object
  - More compact than pixels

Origins of Edges

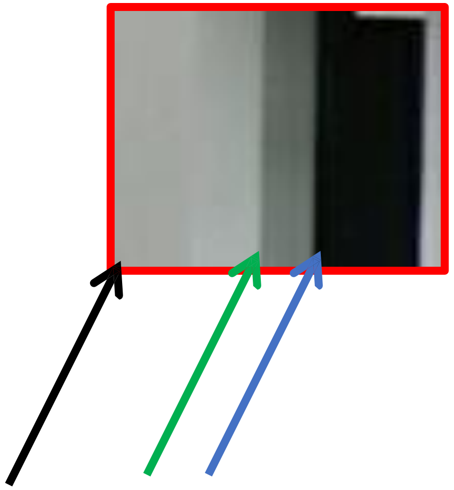
- Edges are caused by a variety of factors



Closeup of edges

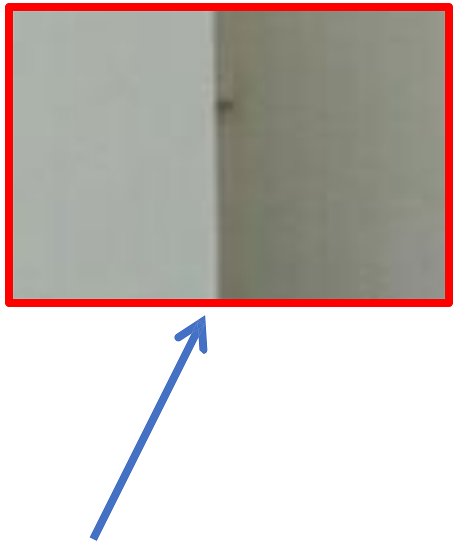
Source: D. Hoiem

Closeup of edges



Source: D. Hoiem

Closeup of edges



Source: D. Hoiem

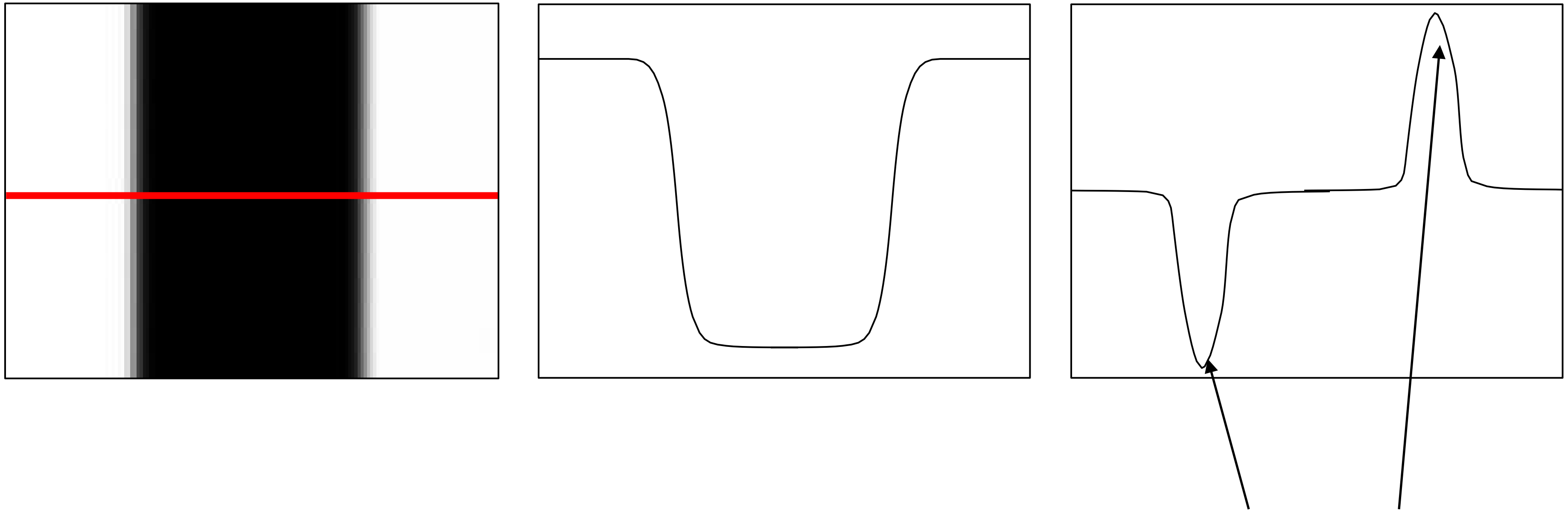
Closeup of edges

Source: D. Hoiem

Characterizing edges

- An edge is a place of rapid change in the image intensity function

image | intensity function (along horizontal scanline) | first derivative



edges correspond to extrema of derivative

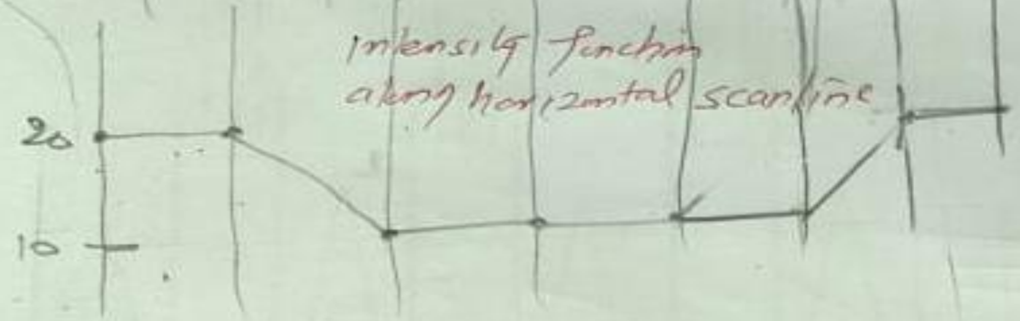
Input Intensity

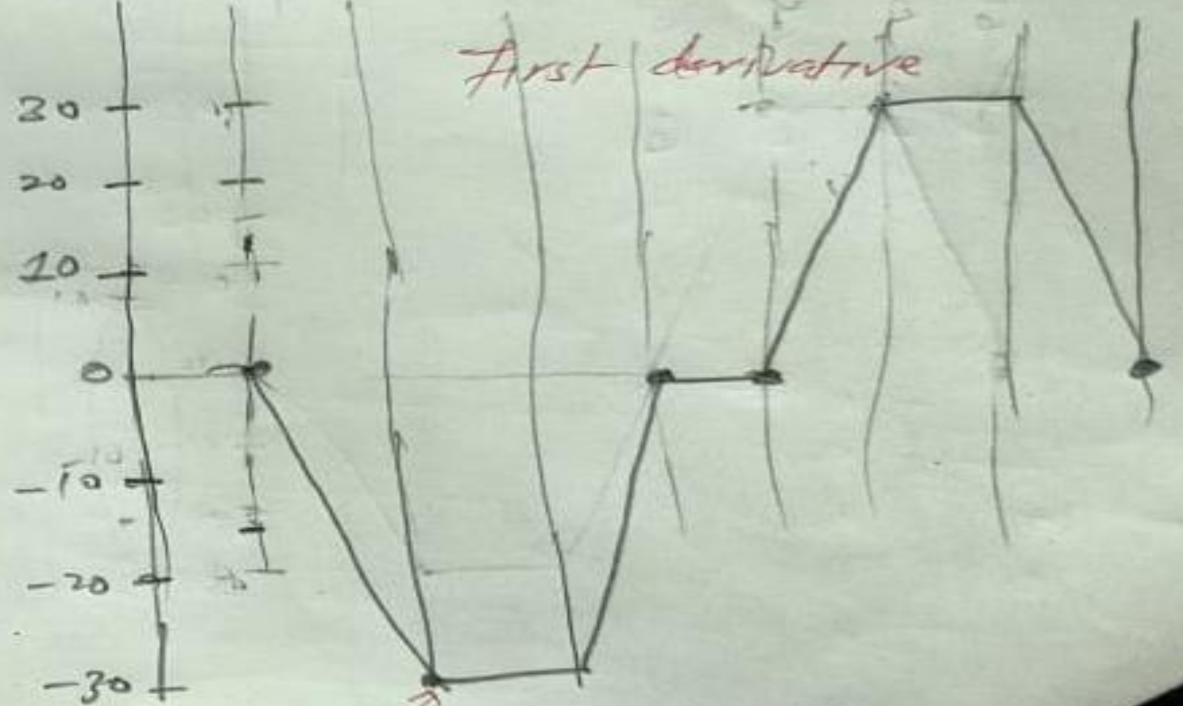
20	20	20	10	10	10	10	20	20
20	20	20	10	10	10	10	20	20
20	20	20	10	10	10	10	20	20
20	20	20	10	10	10	10	20	20
20	20	20	10	10	10	10	20	20

$$\circledast \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

derivative intensity

0	0	0	0	0	0	0	0	0
0	0	-30	-30	0	0	30	30	0
0	0	-30	-30	0	0	30	30	0
0	0	-30	-30	0	0	30	30	0
0	0	-30	-30	0	0	30	30	0
-0	0	-30	-30	0	0	30	0	0

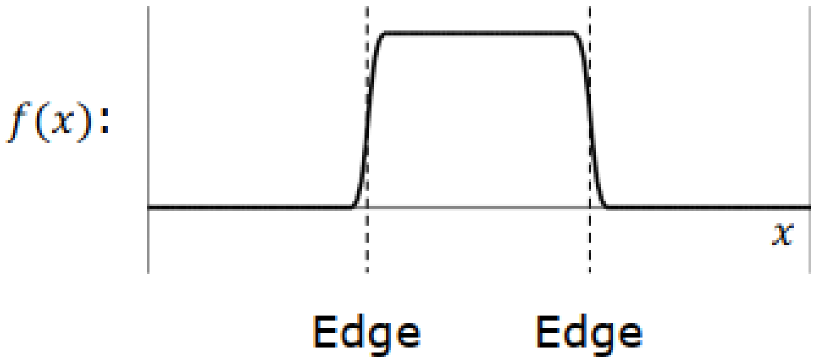




edge corresponds to
extrema of derivative.

1D Edge Detection

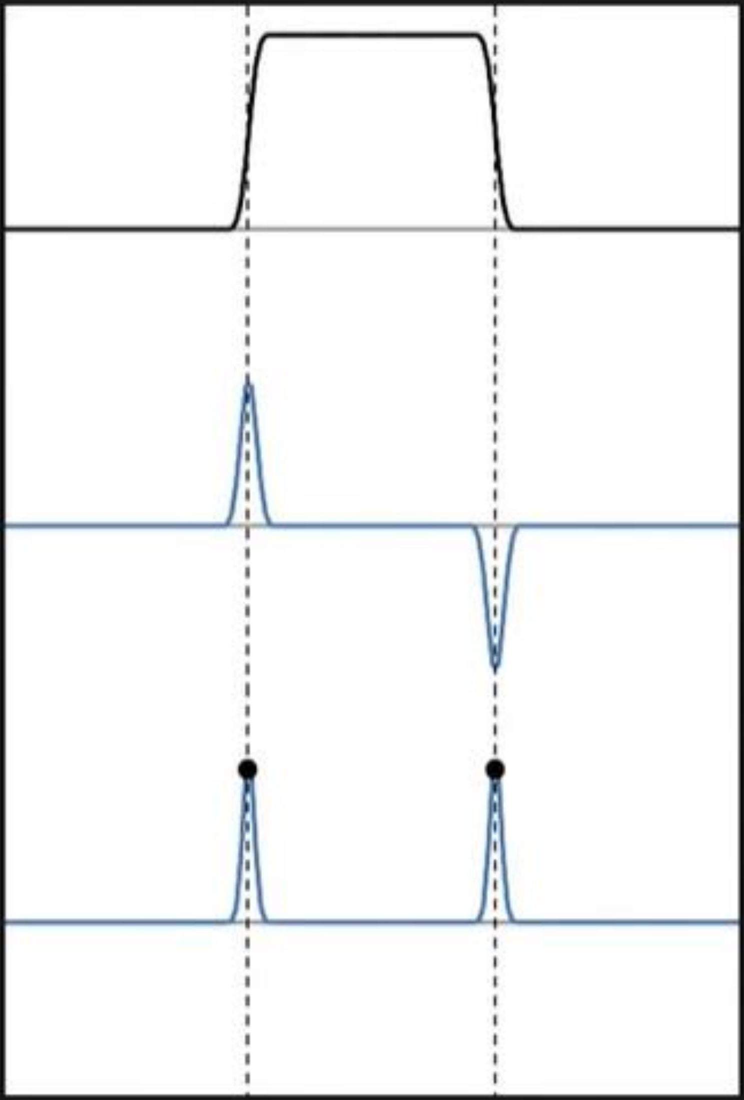
Edge is a rapid change in image intensity in a small region.



Basic Calculus: Derivative of a continuous function represents the amount of change in the function.

Edge Detection Using 1st Derivative

$f(x)$



First Derivative: $\frac{\partial f}{\partial x}$

Local Extrema Indicate Edges

First Derivative Absolute Value: $\left|\frac{\partial f}{\partial x}\right|$

Local Maxima Indicate Edges

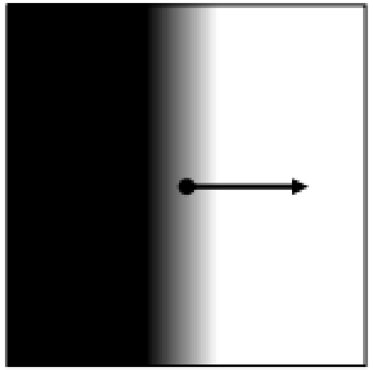
Provides Both Location and Strength of an Edge

Gradient (∇)

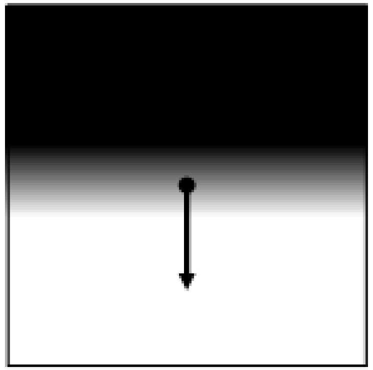
Gradient (Partial Derivatives) represents the direction of most rapid change in intensity

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

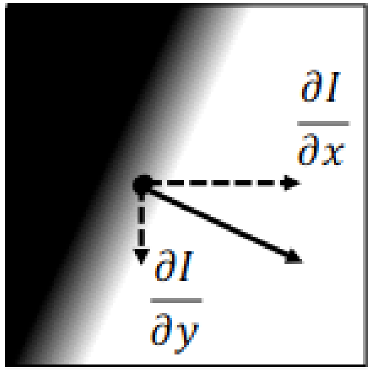
Pronounced as "Del I"



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

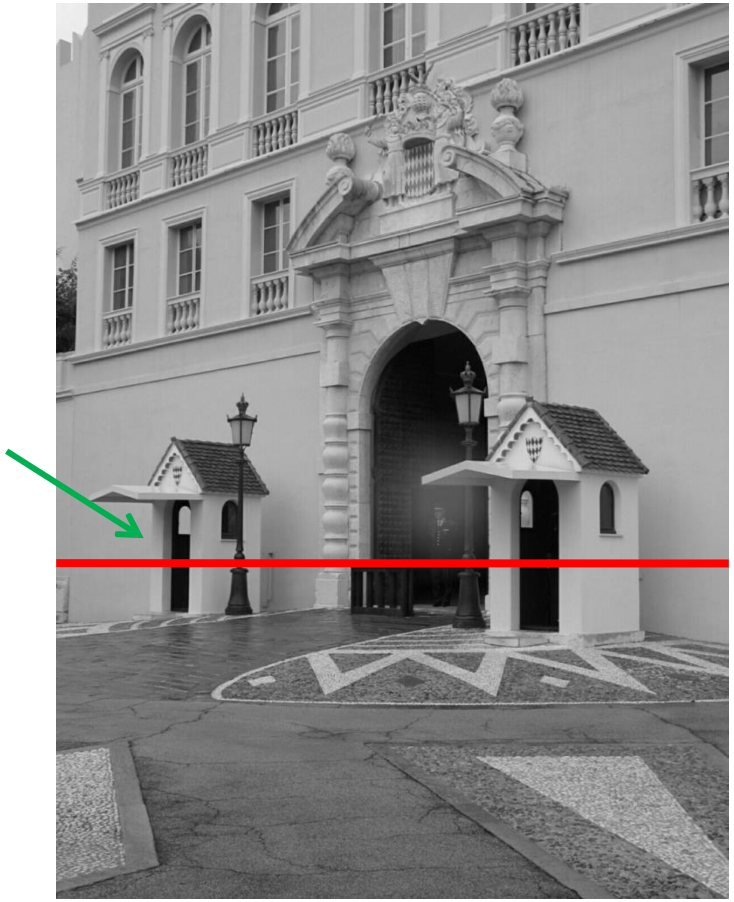


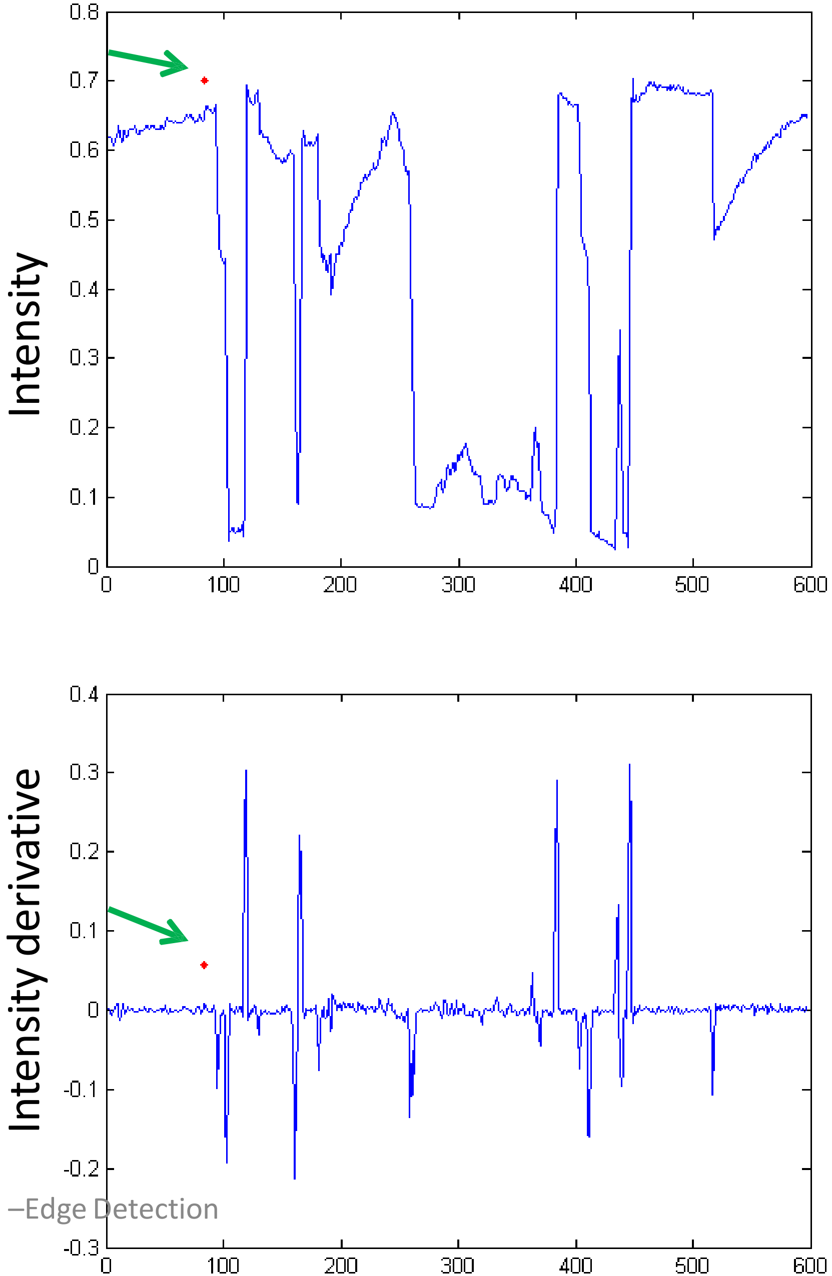
$$\nabla I = \left[0, \frac{\partial I}{\partial y}\right]$$



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

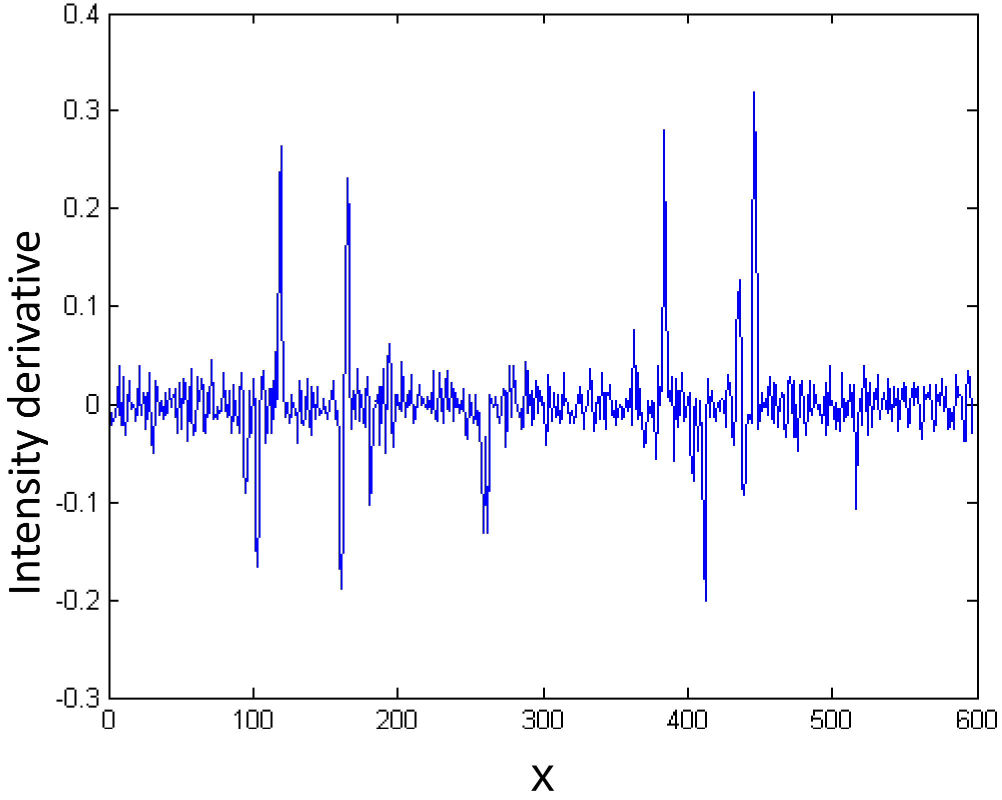
Intensity profile



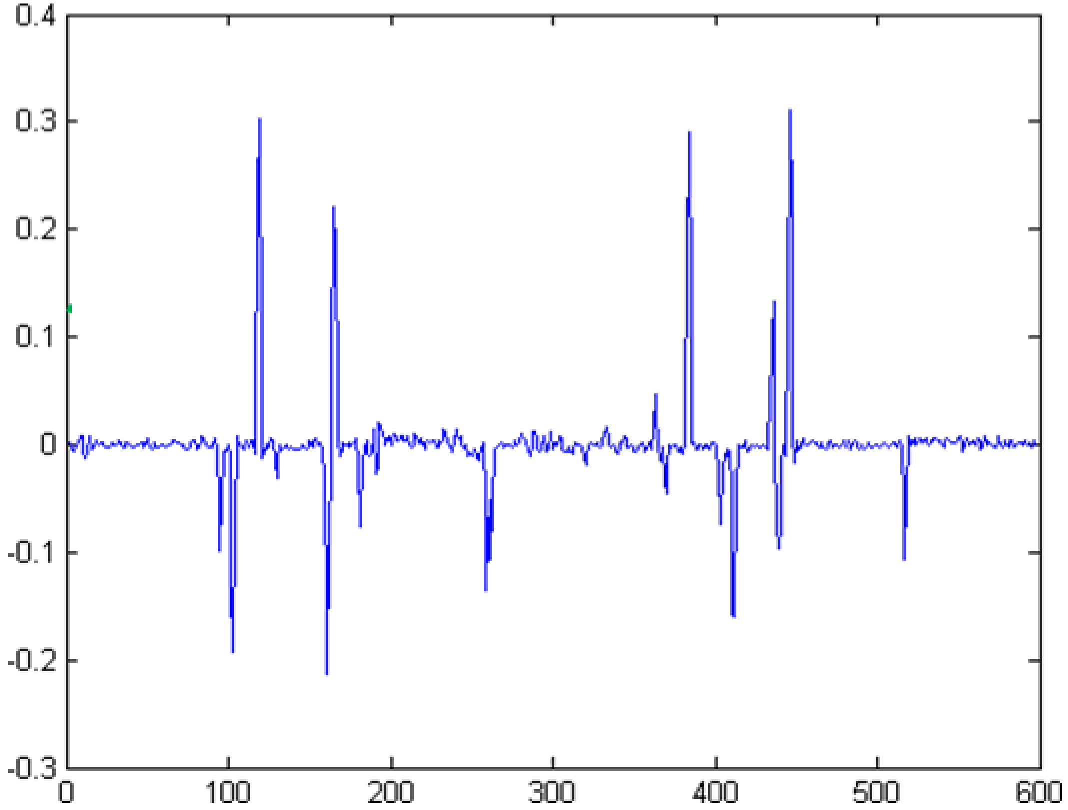


Source: D. Hoiem

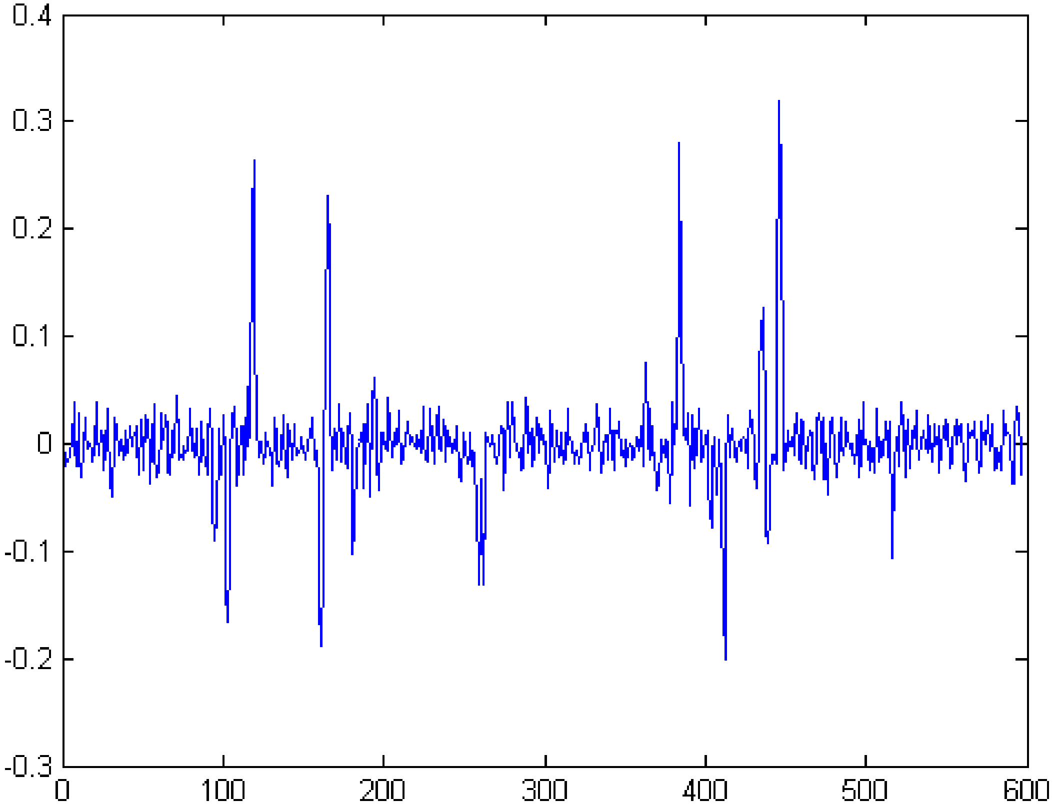
With a little Gaussian noise



Source: D. Hoiem

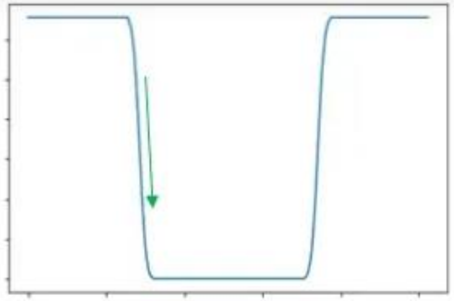


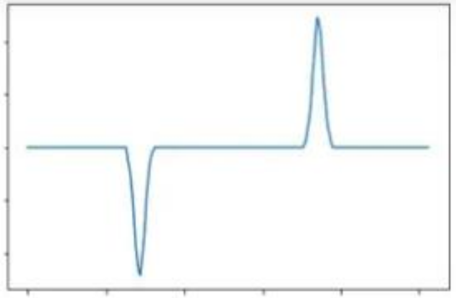
a) Intensity derivatives of clean image



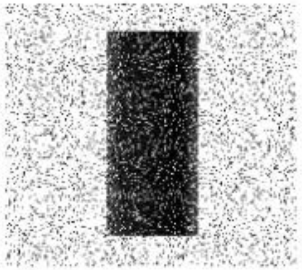
b) Intensity derivative of Image with gaussian noise

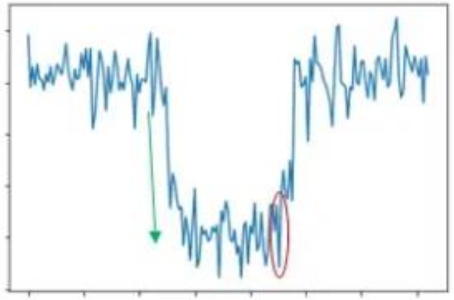
$I[x, y]$

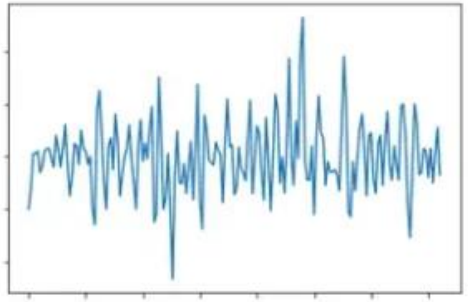




$I[x, y] + Noise$

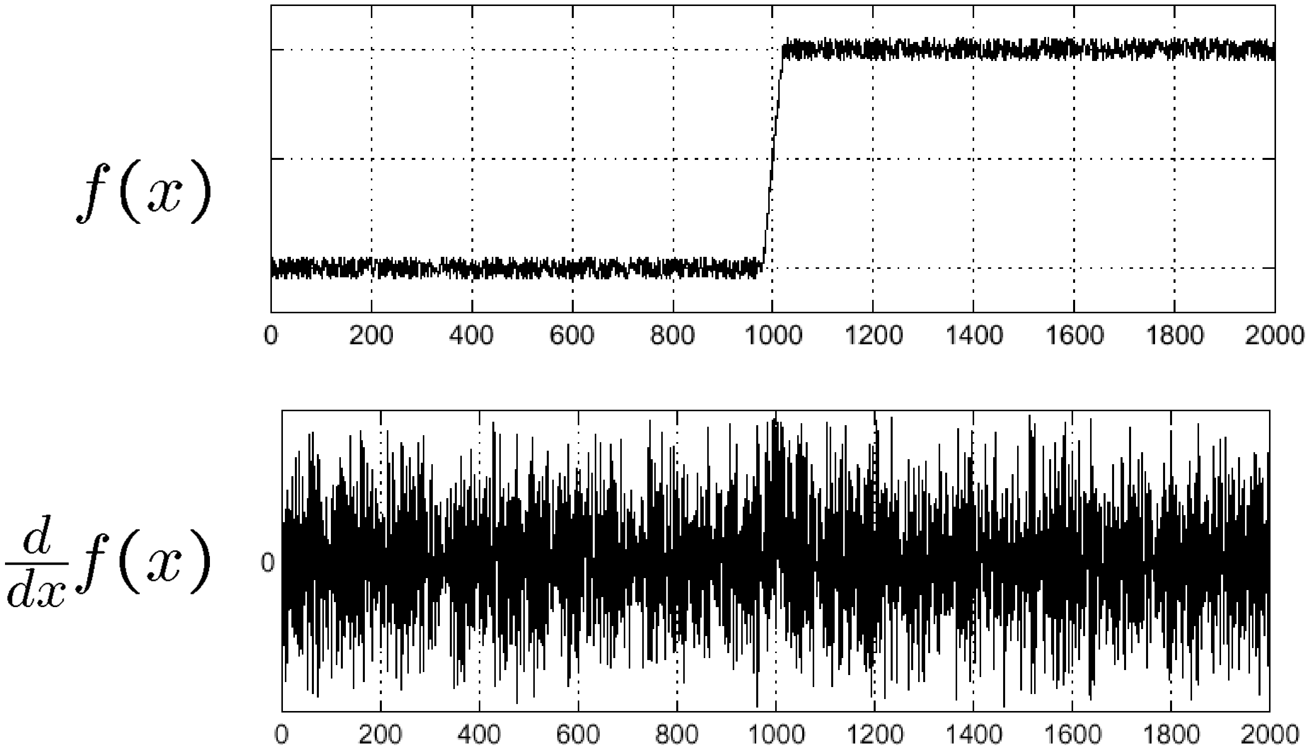






Effects of Noise

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



Where is the edge?

Credit to Seitz.

with little noise

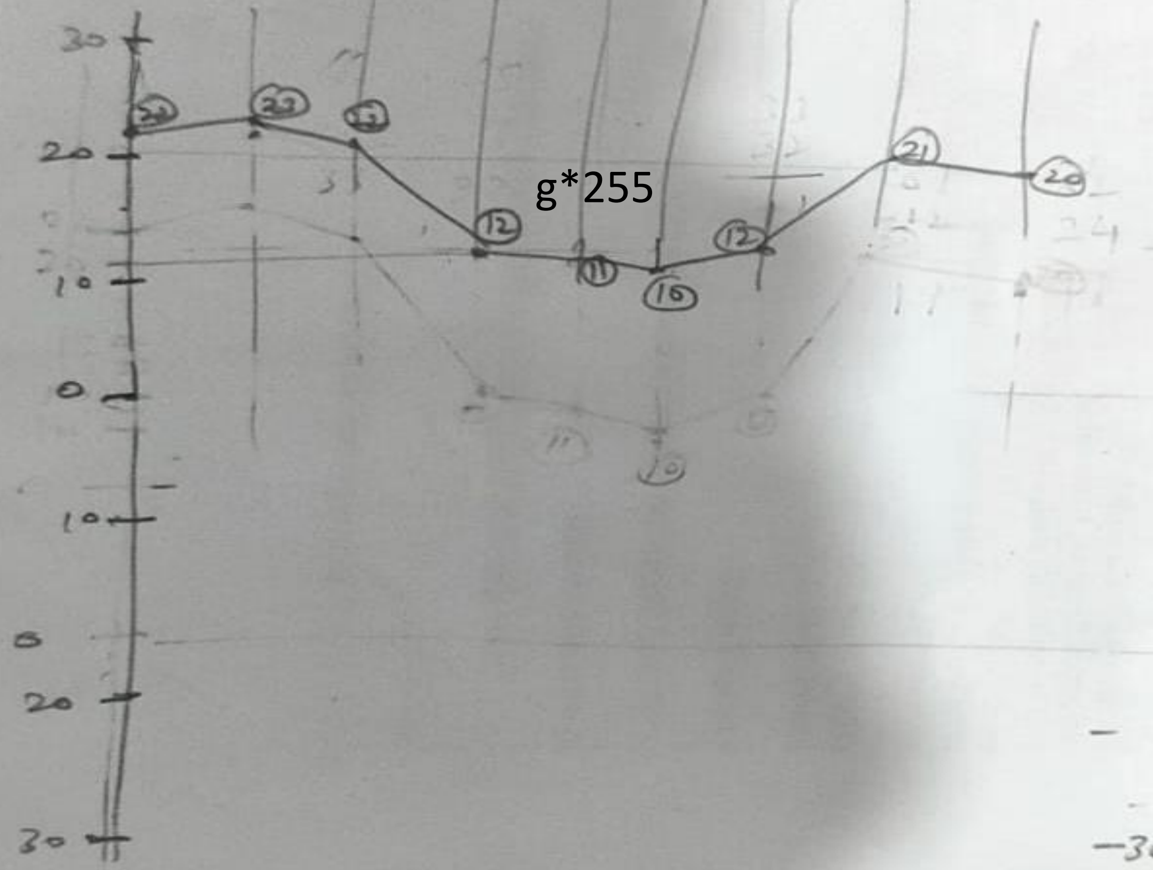
21	22	23	10	12	10	12	20	21
22	23	22	12	11	10	12	21	20
20	21	20	11	10	12	10	20	21
21	20	22	12	11	10	11	21	20
22	21	20	10	12	11	10	20	22

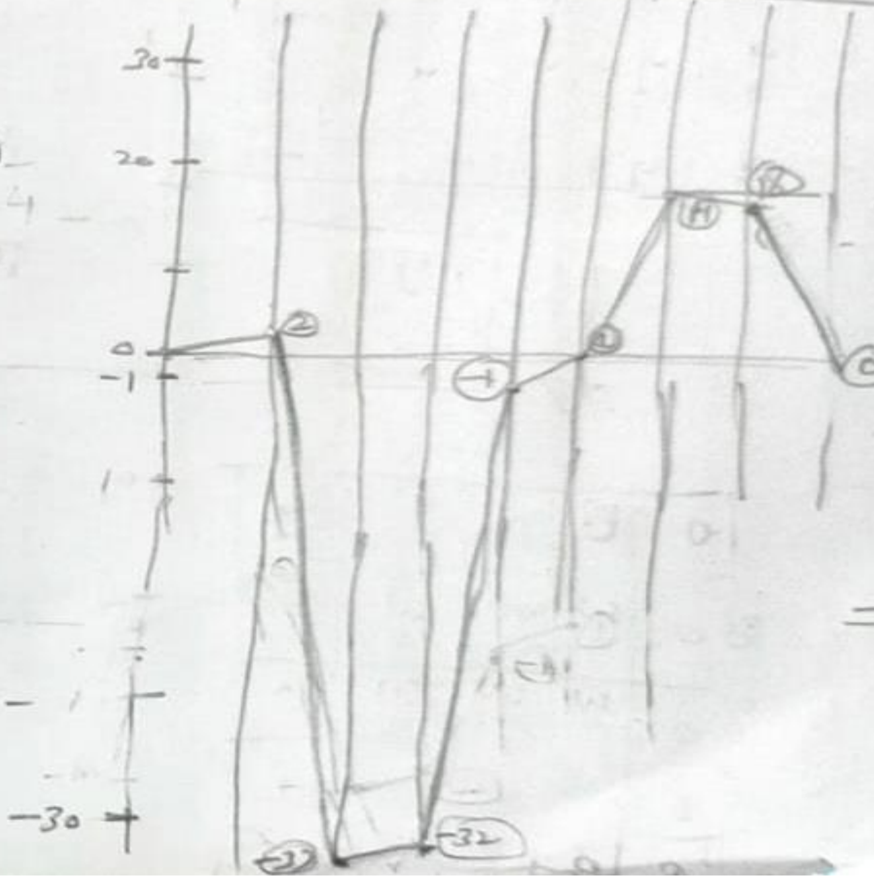
*

-1	0	1
-1	0	1
-1	0	1

→

0	0	0	0	0	0	0	0	0
0	2	-33	-32	-1	1	19	18	0





⇒ Change are detected all over the scan line

⇒ So many Edges are there

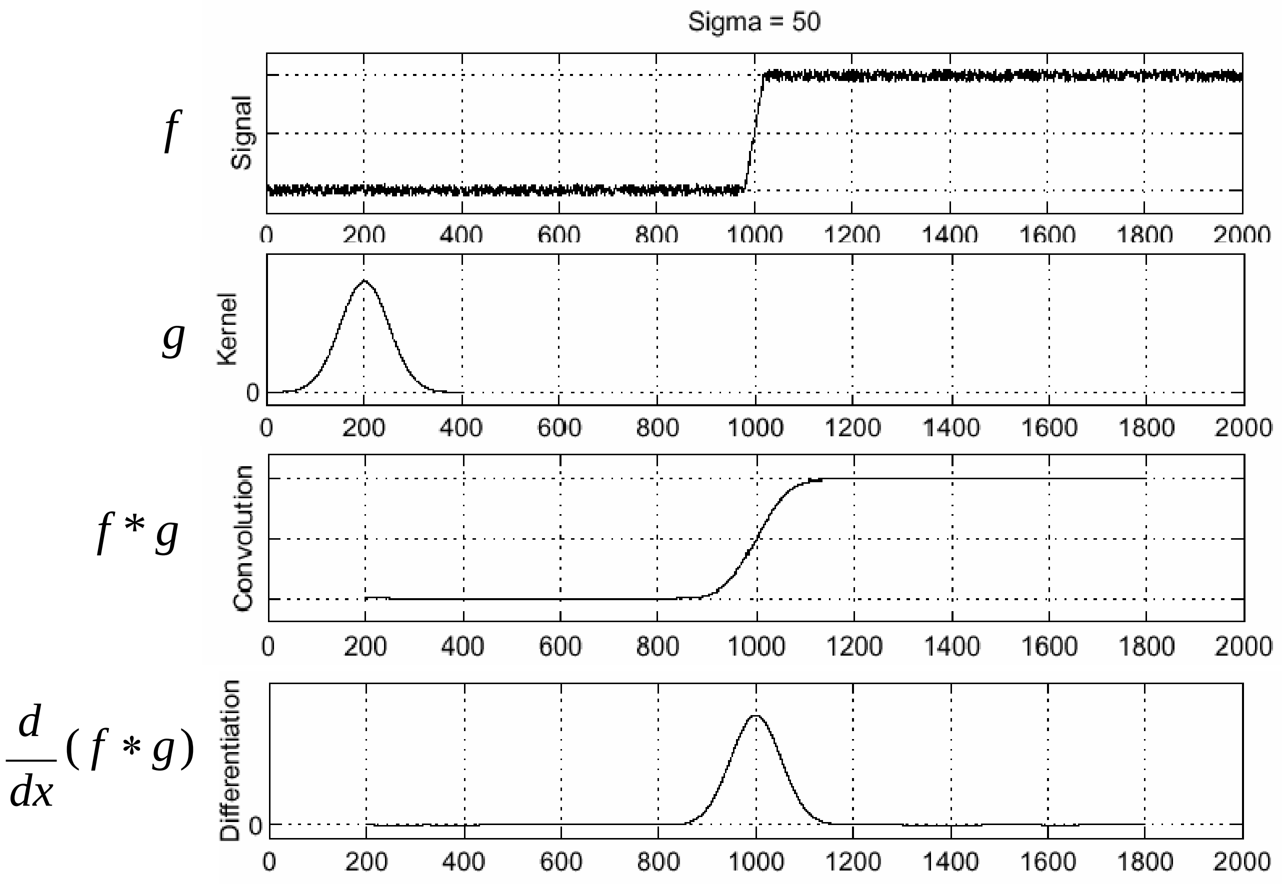
⇒ So detected edge become hard.

Effects of noise

- Difference filters respond strongly to noise
  - Image noise results in pixels that look very different from their neighbors
  - Generally, the larger the noise the stronger the response
- What can we do about it?

Source: D. Forsyth

Solution: smooth first



To find edges, look for peaks in $\frac{d}{dx}(f * g)$

Source: S. Seitz

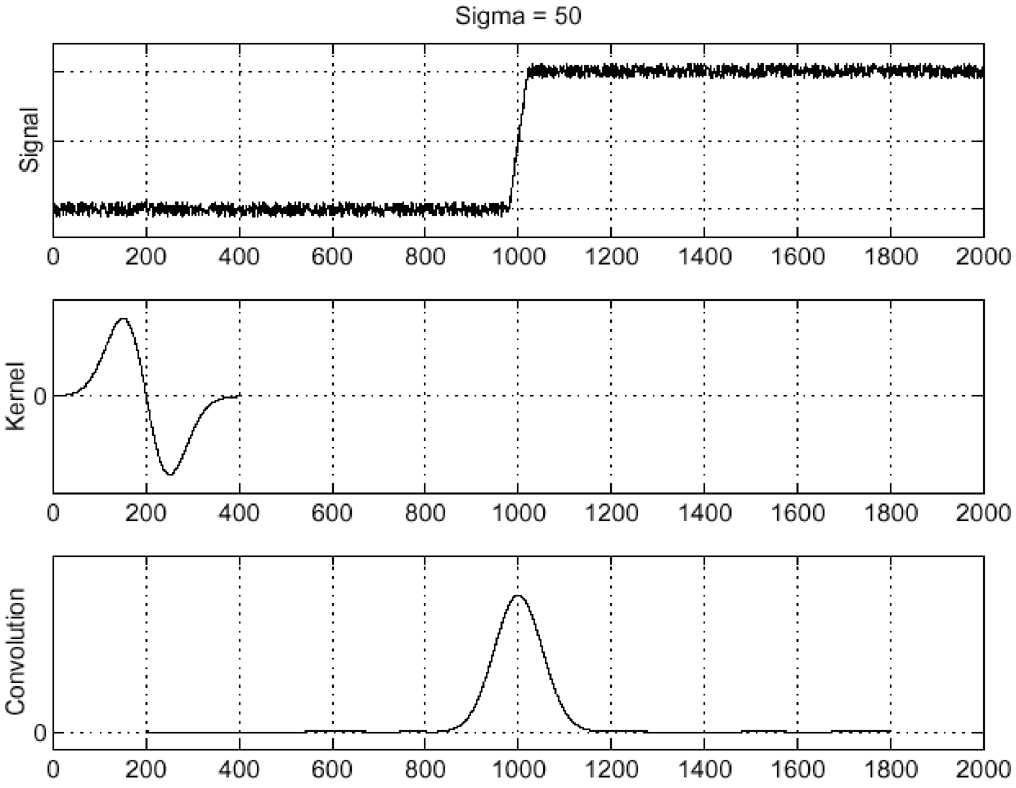
Derivative theorem of convolution

- Convolution is differentiable: $$\frac{d}{dx}(f*g)=f*\frac{d}{dx}g$$
- This saves us one operation:

f

$\frac{d}{dx}g$

$f*\frac{d}{dx}g$



Source: S. Seitz

Prewitt and Sobel Edge Detector

- Compute derivatives
  - In x and y directions
- Find gradient magnitude
- Threshold gradient magnitude

Prewitt Edge Detector

1. Convert image to grey-scale
2. Compute derivatives in x and y directions (f_x and f_y using horizontal and vertical Prewitt filter respectively)
3. Find gradient magnitude at every pixel

$$\sqrt{f_x^2 + f_y^2}$$

4. Threshold gradient magnitude image

Prewitt Filter

Prewitt Filter

- **Horizontal Prewitt Filter**

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

- **Vertical Prewitt Filter**

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Note that these kernels are separable

Here, these columns smooth in vertical direction

$$\underbrace{K_x = \begin{bmatrix} \boxed{\begin{matrix} -1 \\ -1 \\ -1 \end{matrix}} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & \boxed{\begin{matrix} 1 \\ 1 \\ 1 \end{matrix}} \end{bmatrix}}_{\text{First order derivative estimator for horizontal direction}}$$

And these rows smooth in horizontal direction

$$\underbrace{K_y = \begin{bmatrix} \boxed{-1 \quad -1 \quad -1} \\ 0 \quad\;\; 0 \quad\;\; 0 \\ \boxed{\;\;1 \quad\;\; 1 \quad\;\; 1} \end{bmatrix}}_{\text{First order derivative estimator for vertical direction}}$$

Note that these kernels are separable

Here, these columns smooth in vertical direction

$$K_x = \underbrace{\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}}_{\text{First order derivative estimator for horizontal direction}}$$

Vertical smoothing filter (Averaging filter)

$$K_x = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

1D first order derivative estimator for horizontal direction

2D kernel $\longrightarrow$ Product of two 1D kernels

Note that these kernels are separable

Instead of 2D, two 1D kernels can be convolved with image sequentially

Here, these columns smooth in vertical direction

$$K_x = \underbrace{\begin{bmatrix} \boxed{\begin{matrix} -1 \\ -1 \\ -1 \end{matrix}} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & \boxed{\begin{matrix} 1 \\ 1 \\ 1 \end{matrix}} \end{bmatrix}}_{\text{First order derivative estimator for horizontal direction}}$$

Vertical smoothing filter
(Averaging filter)

$$K_x = \boxed{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}} \boxed{\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}}$$

1D first order derivative estimator
for horizontal direction

2D kernel $\longrightarrow$ Product of two 1D kernels

These kernels are separable too, as in Prewitt

$$\underbrace{K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}}_{\text{First order derivative estimator for horizontal direction}}$$

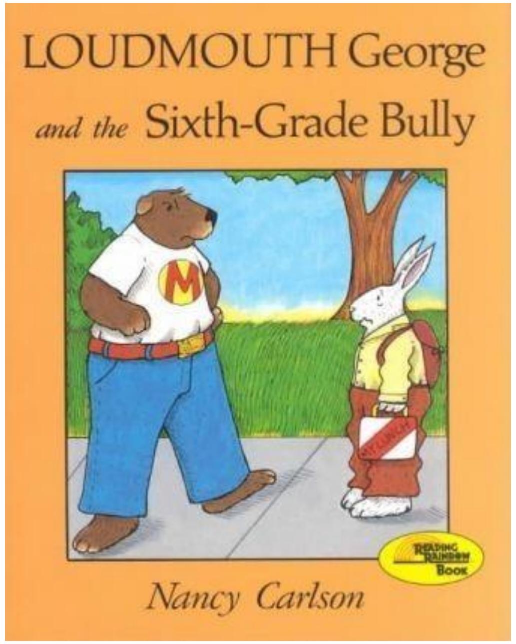
$$K_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \boxed{\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}}$$

1D first order derivative estimator for horizontal direction

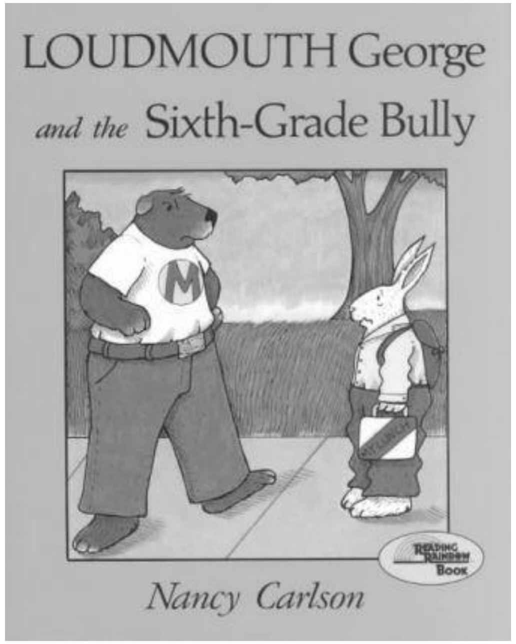
2D kernel $\longrightarrow$ Product of two 1D kernels

Prewitt Edge Detector: Step 1

Convert image to grey-scale



Gray $\longrightarrow$

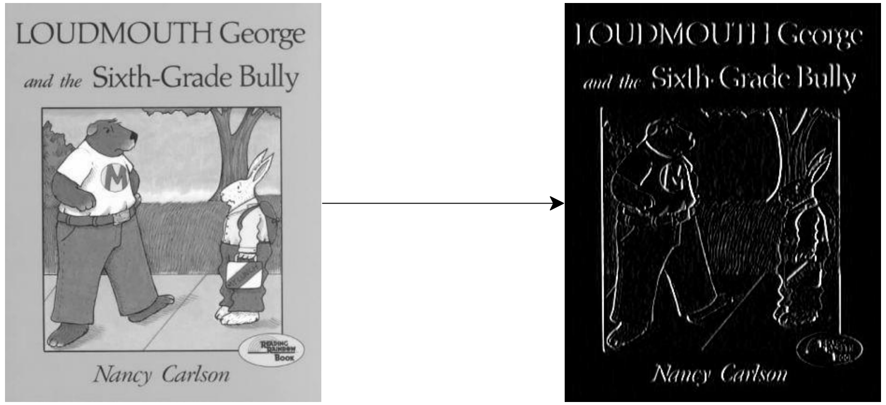


Original Image

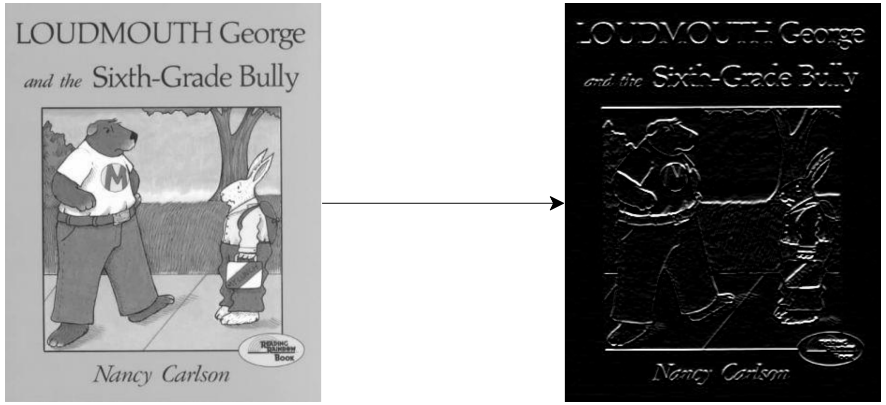
$f(x, y)$

Prewitt Edge Detector: Step 2

Compute f_x i.e. convolve gray image with horizontal Prewitt filter



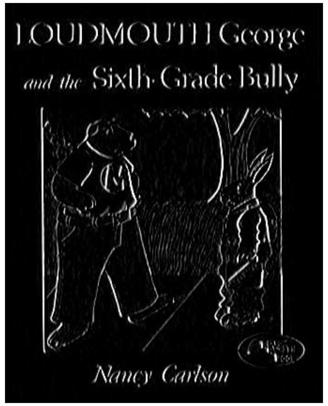
Compute f_y i.e. convolve gray image with vertical Prewitt filter



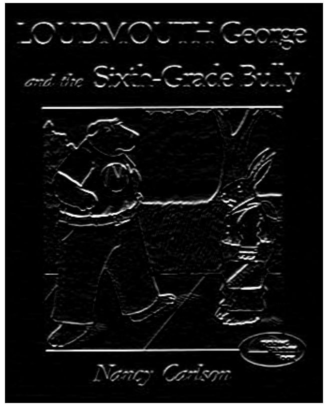
$f(x, y)$ $\qquad$ f_y

Prewitt Edge Detector: Step 3

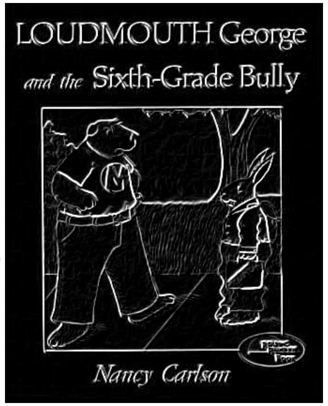
Find gradient magnitude at each pixel



f_x



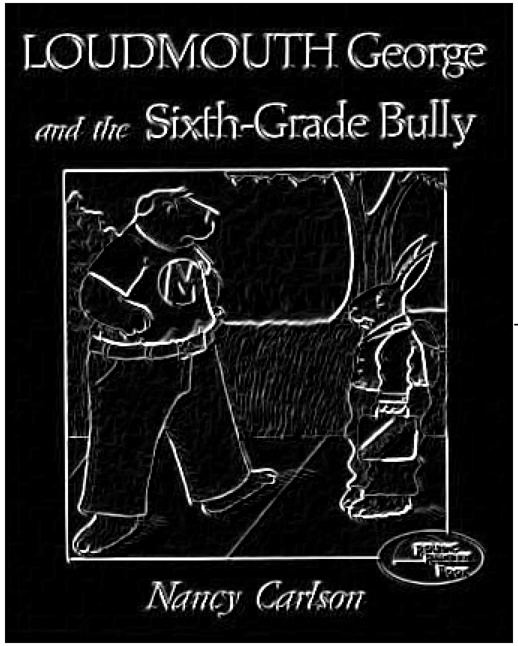
f_y



$\|\nabla f(x, y)\|$

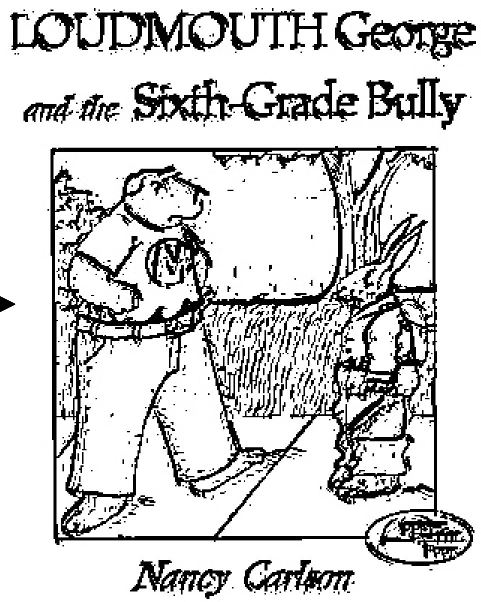
Prewitt Edge Detector: Step 4

Threshold gradient magnitude image



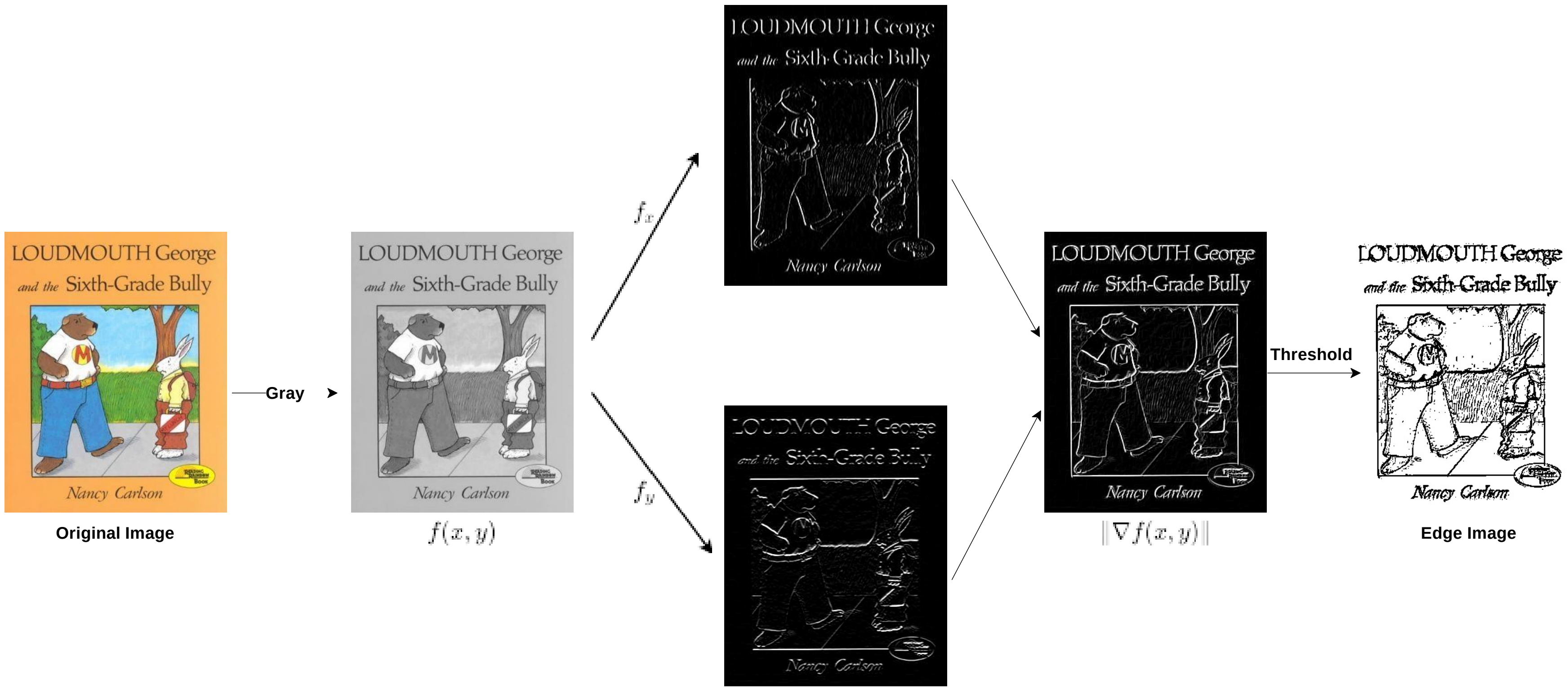
$\|\nabla f(x,y)\|$

Threshold →



Edge Image

Prewitt Edge Detector Complete Pipeline



Sobel Edge Detector

1. Convert image to grey-scale
2. Compute derivatives in x and y directions (f_x and f_y using horizontal and vertical Sobel filter respectively)
3. Find gradient magnitude at every pixel

$$\sqrt{f_x^2 + f_y^2}$$

4. Threshold gradient magnitude image

Sobel Filter

- **Horizontal Sobel Filter**

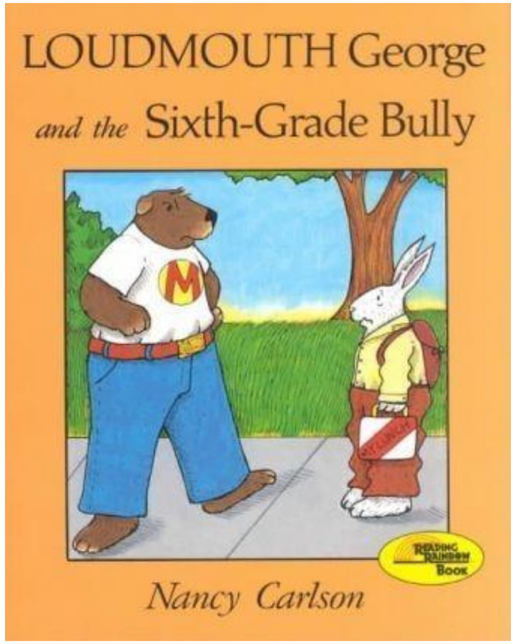
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- **Vertical Sobel Filter**

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

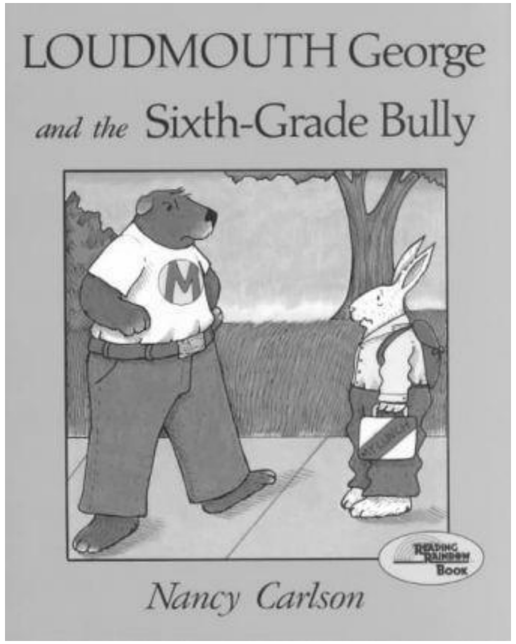
Sobel Edge Detector: Step 1

Convert image to grey-scale



Original Image

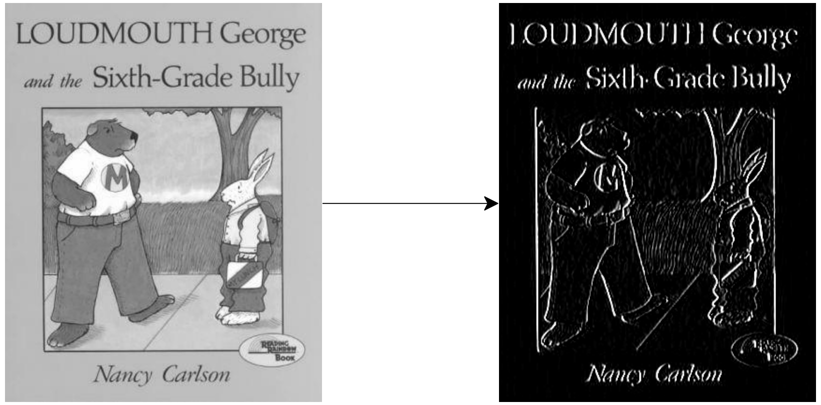
Gray →



$f(x, y)$

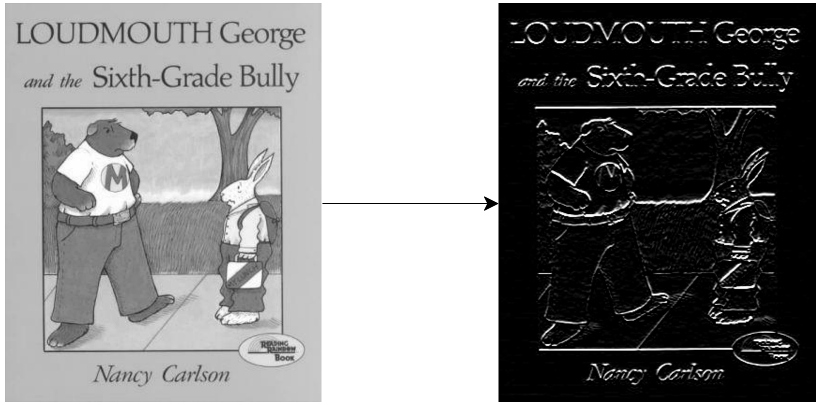
Sobel Edge Detector: Step 2

Compute f_x i.e. convolve gray image with horizontal Sobel filter



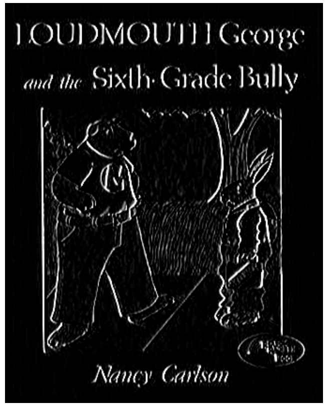
$f(x, y)$ f_x

Compute f_y i.e. convolve gray image with vertical Sobel filter

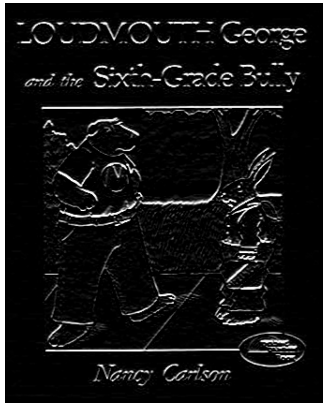


Sobel Edge Detector: Step 3

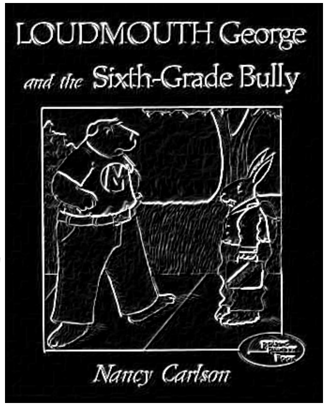
Find gradient magnitude at each pixel



f_x



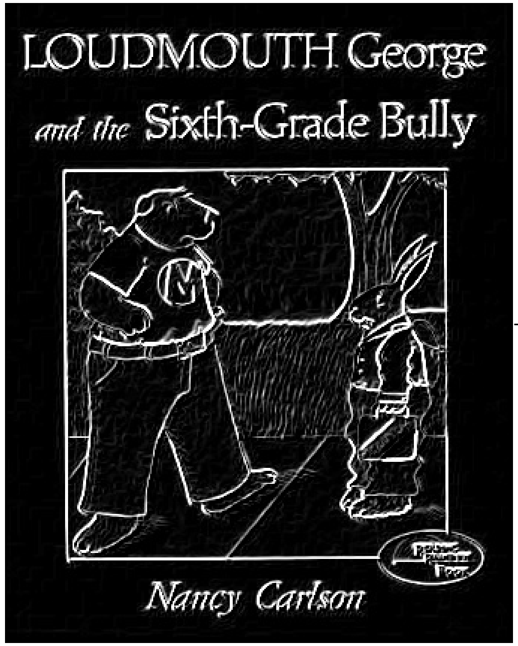
f_y



$\|\nabla f(x, y)\|$

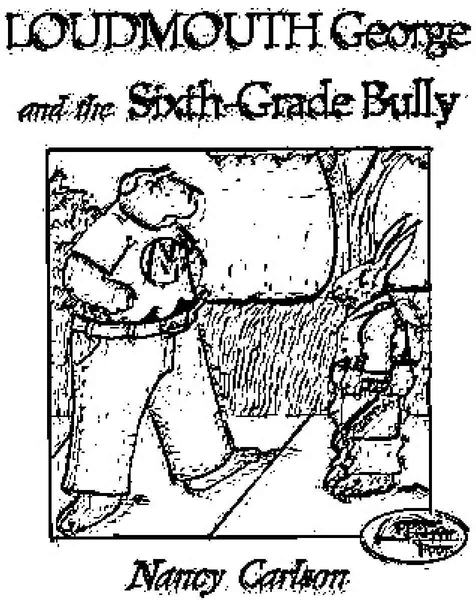
Sobel Edge Detector: Step 4

Threshold gradient magnitude image



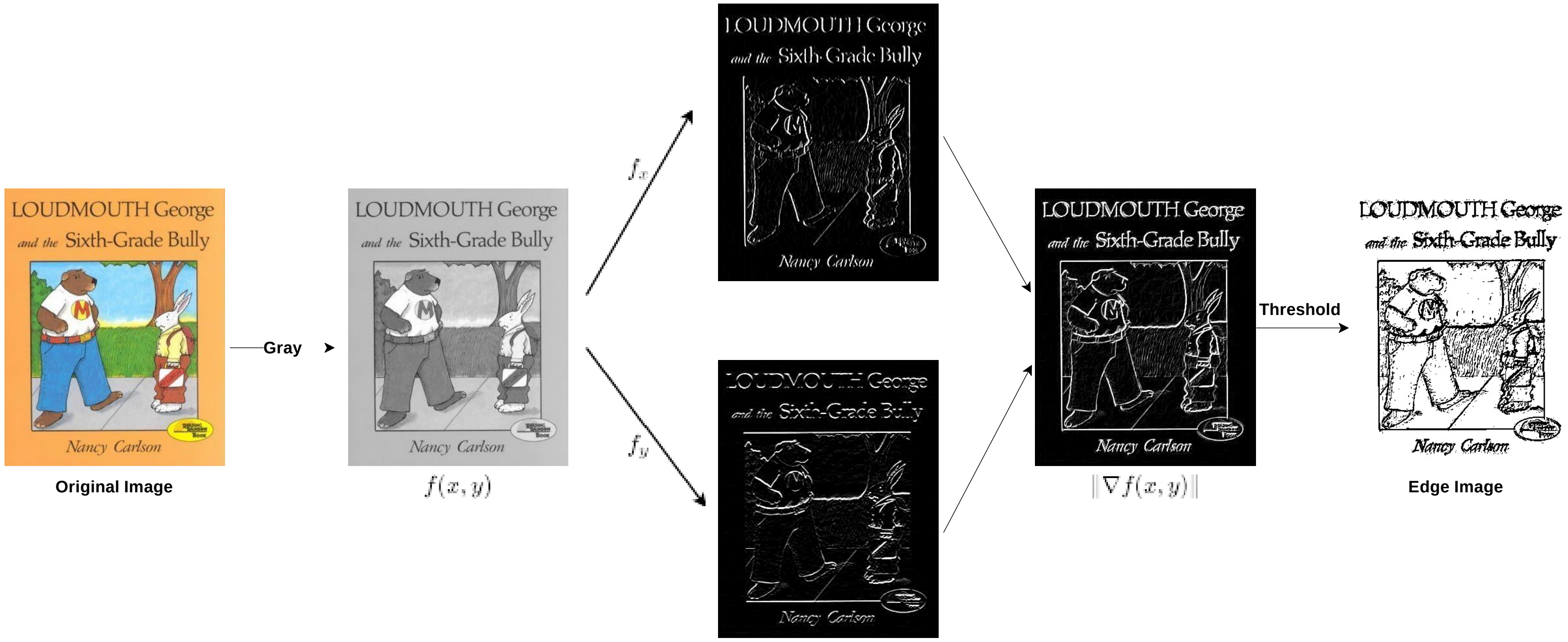
$\|\nabla f(x, y)\|$

Threshold

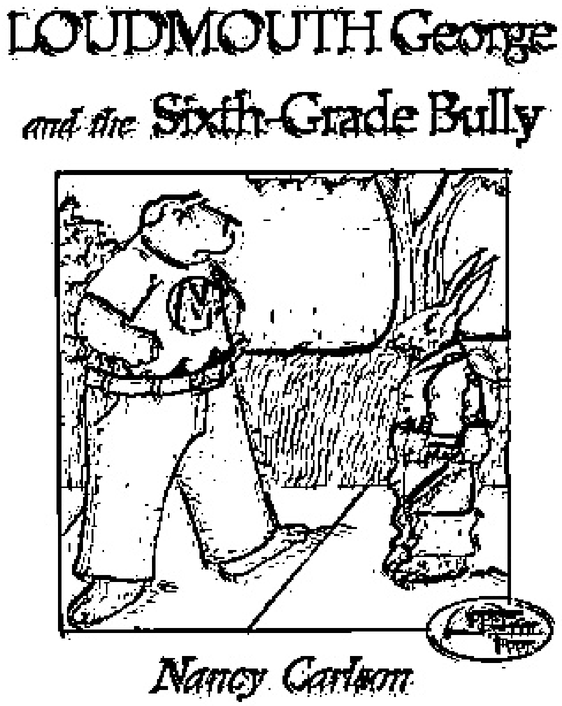


Edge Image

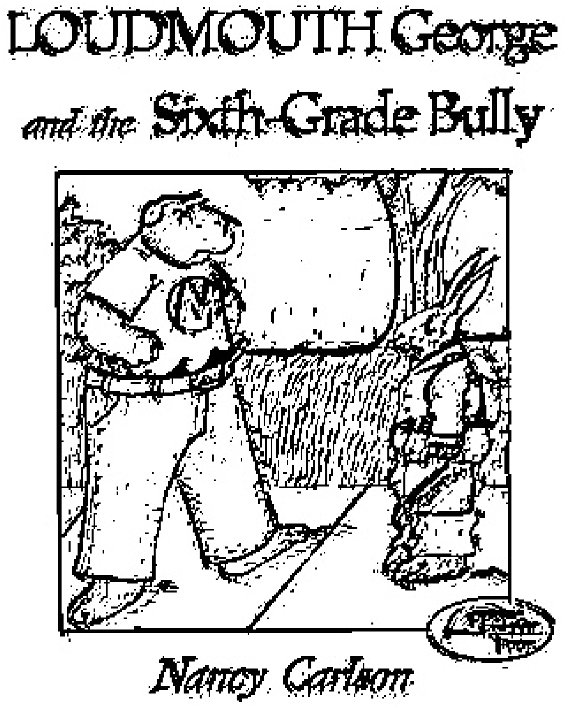
Sobel Edge Detector Complete Pipeline



Prewitt vs. Sobel Results

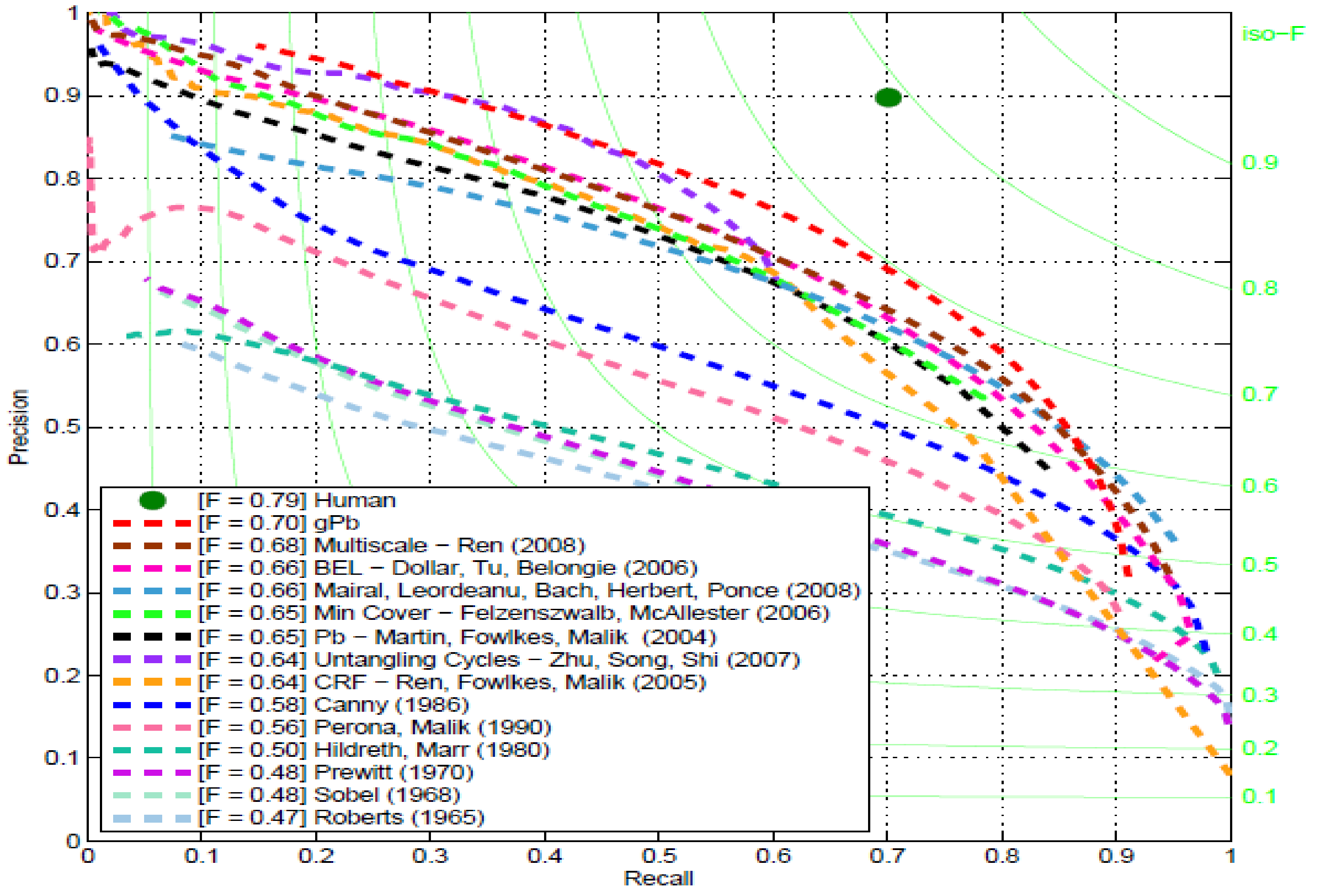


Prewitt



Sobel

Sobel vs Prewitt



Source: Arbelaez, Maire, Fowlkes, and Malik. TPAMI 2011 (pdf)

Source: Dr Rawat(AP UCF)

Questions?

Sources for this lecture include materials from works by Mubarak Shah, Abhijit Mahalanobis, and D. Lowe

Other sources from James Hays, Lana Lazebnik, Steve Seitz, David Forsyth, David Lowe, Fei-Fei Li, and Derek Hoiem