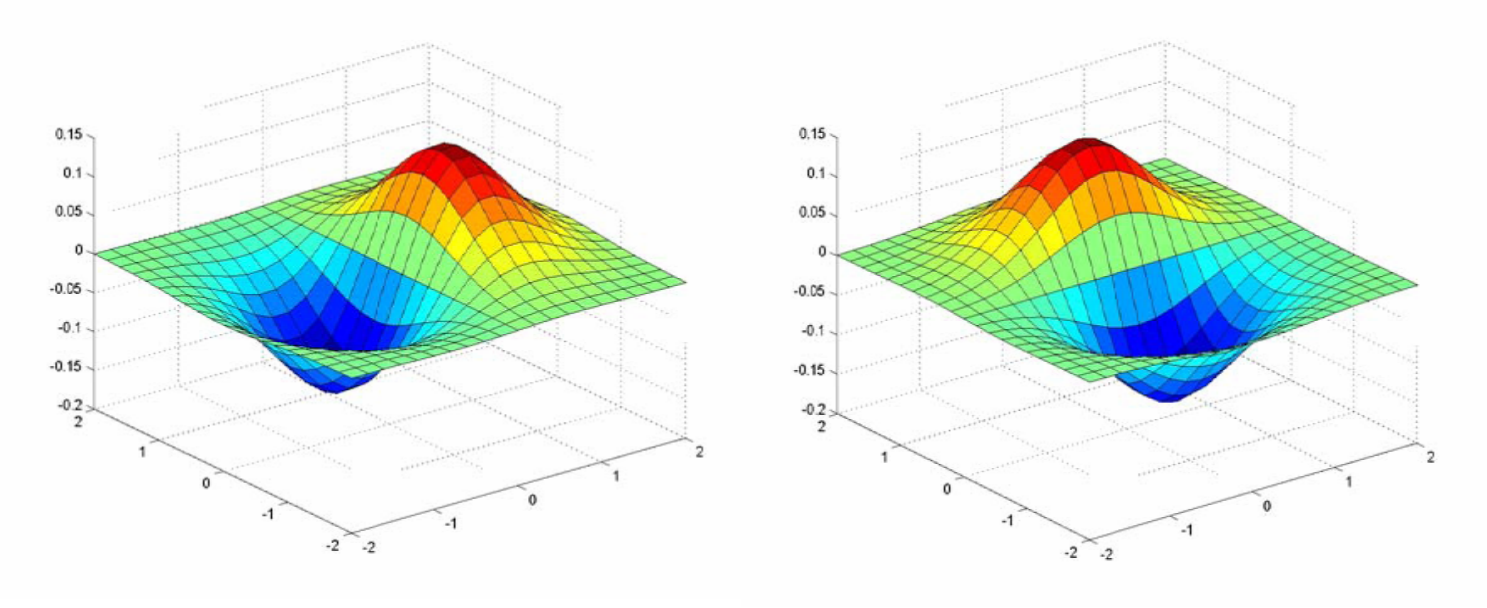
CSCI 497P/597P: Computer Vision



Lecture 4: Gradients and Edge Detection

Announcements

- Week 1 Feedback survey is out on Canvas - fill it out by Friday night.
  - It's worth 1 homework point.
  - It's quick: 4 multiple choice questions with an optional comments field.

Goals

- Understand the limitations of linear filtering
- Know how to compute image derivatives using convolution filters
- Understand how the Sobel filter works to detect edges.

Filtering, so far

- Filtering:
  output pixel depends on input neighborhood

- Linear filtering:
  output pixel is a weighted average of input neighborhood
  (must always the same weights to be linear)

- Cross-correlation is a kind of linear filtering:
  output pixel = weighted average(neighborhood)

- Convolution: cross-correlation, but first flip the kernel horizontally and vertically

Limitations:
What *can't* convolution do?

Problem #1 (= #5 from last lecture) - discuss in groups.

- Maximum filter?

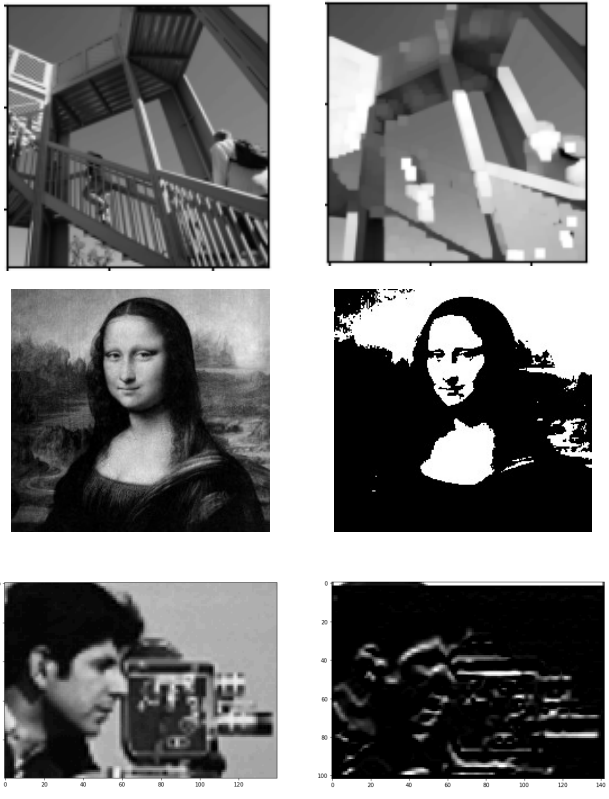
  no!

- Threshold?

  no!

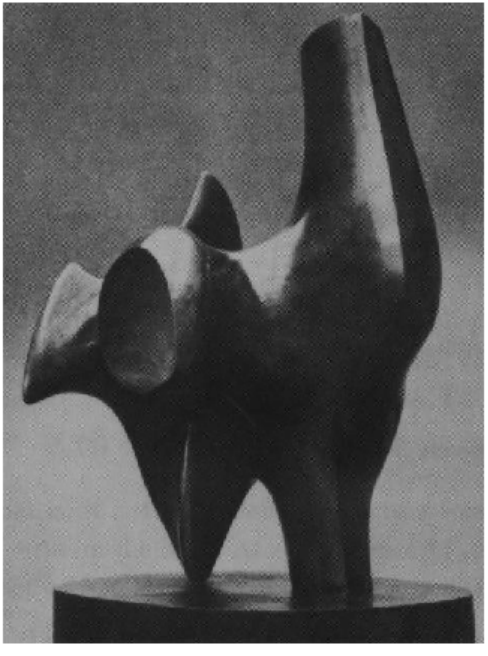
- y partial derivative?

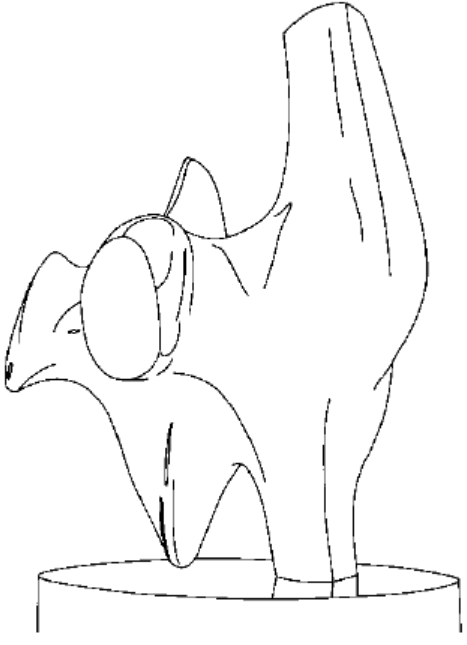
  yes!



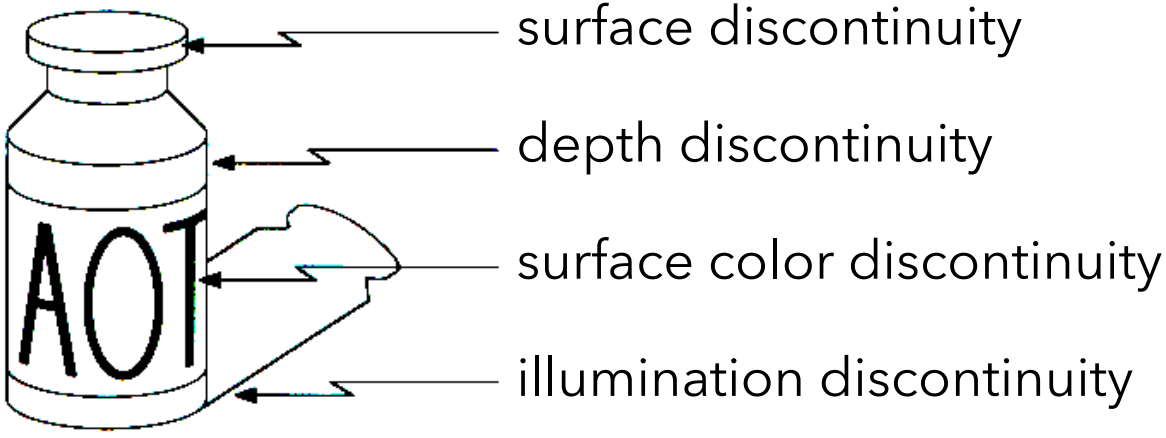
Calculus!?

Edge detection: a classic vision problem.



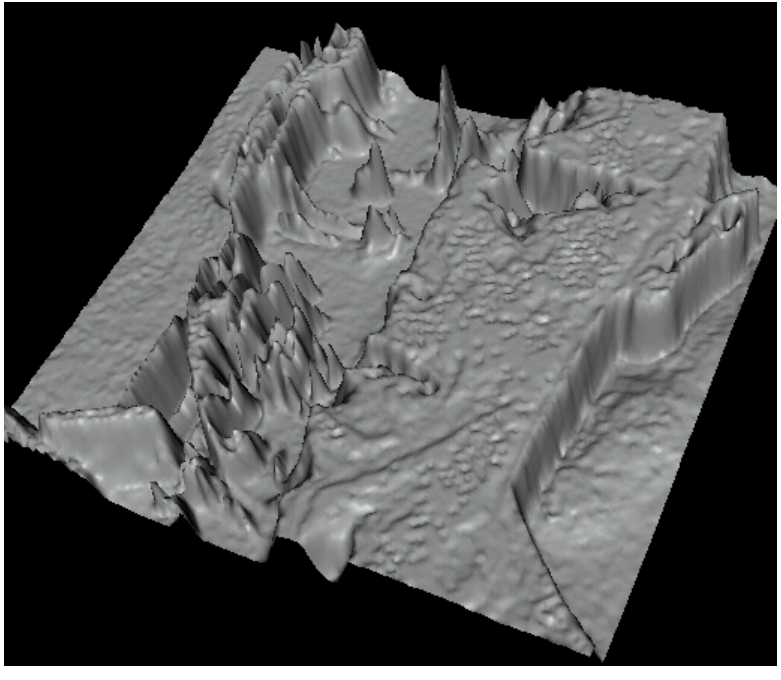


What is an edge?



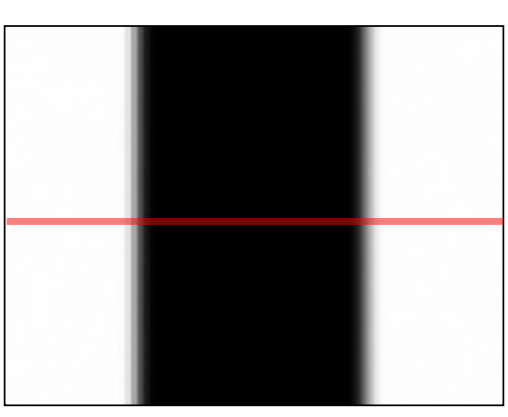
How do we find them?

f(x,y) as brightness

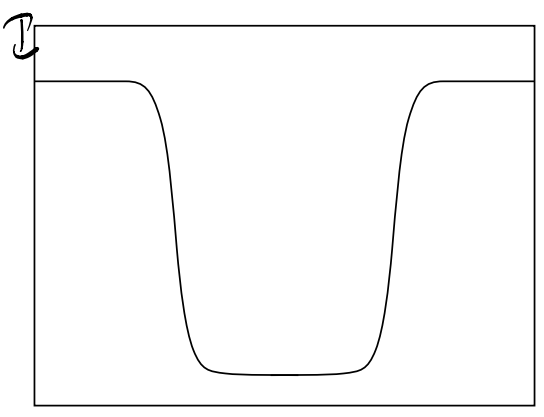


f(x,y) as height

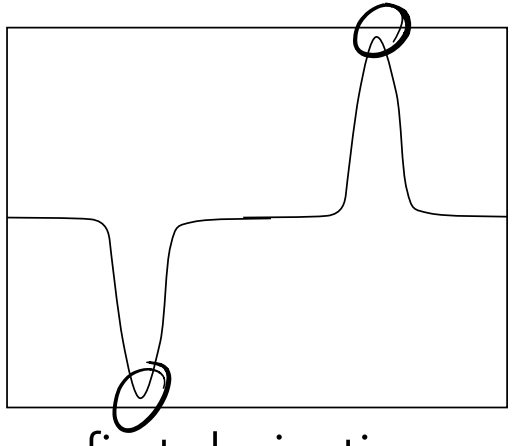
Characterizing edges



image



1D scanline



first derivative

Multivariable Calculus!?

Partial derivative with respect to x: pretend all other variables are constants and differentiate.

$$f(x,y) = x^2 + 2y$$

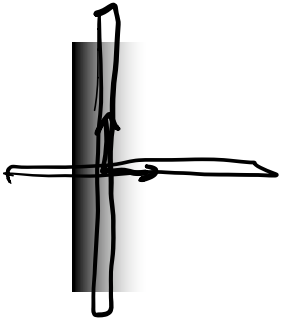
$$\frac{\partial f}{\partial x} = 2x \qquad \frac{\partial f}{\partial y} = 2$$

Gradient: the vector of all partial derivatives

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

Image Derivatives

Images are 2D - have x and y partial derivatives

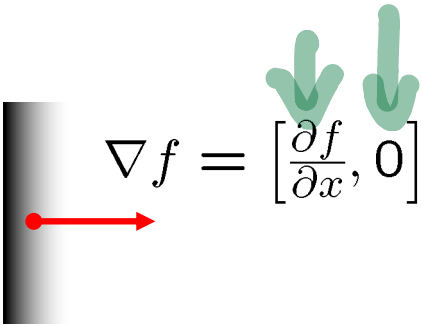


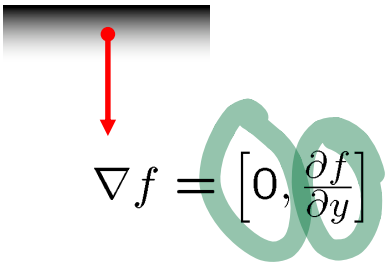
The image gradient is the vector of partial derivatives:

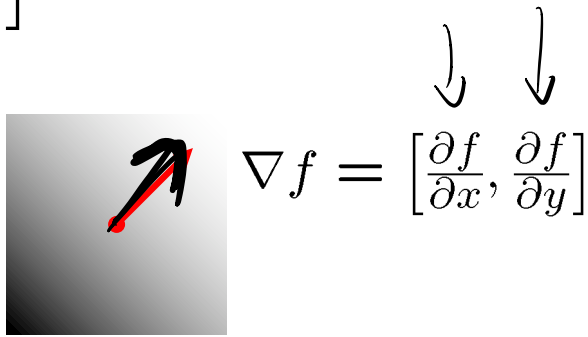
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

Image Gradient as Edge Detector

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







What is the edge **strength**?

$\|\nabla f\|$

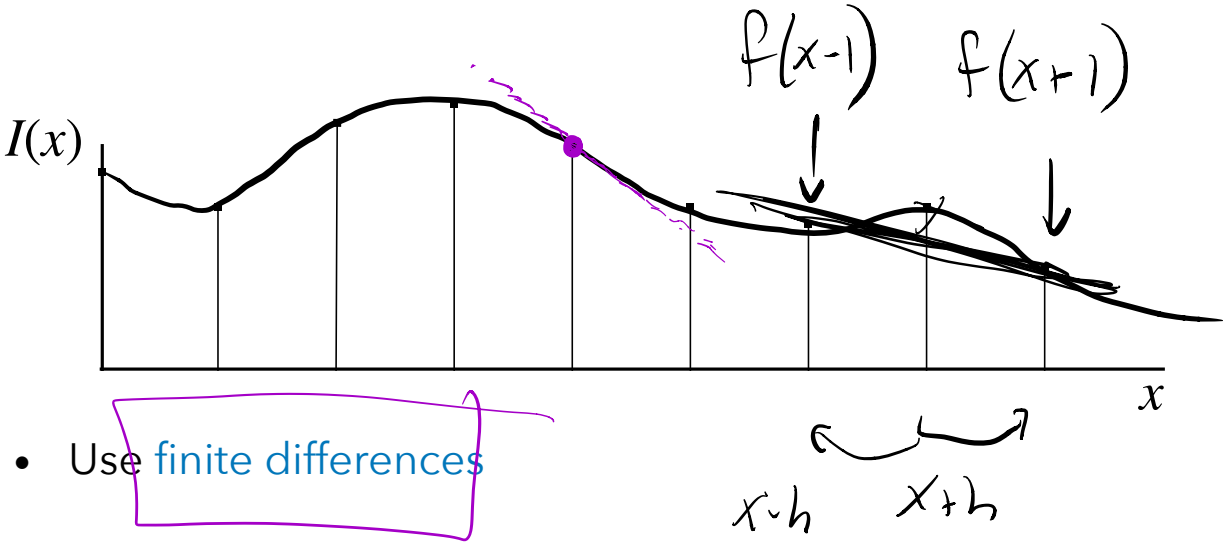
What is the edge **direction**?

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

in practice: atan2(y, x)

Image Derivatives

- How do we differentiate a discrete (sampled) image?
  - Reconstruct a continuous function and compute the derivative



  - Use finite differences

Derivative Filters

- How do we differentiate a discrete digital image?
  - Use finite differences

Example image:

0	0	1
0	0	1
0	1	1

Candidate derivative filters:

not centered

0	0	0
1	-1	0
0	0	0

0	1	0
0	-1	0
0	0	0

centered

0	0	0
1	0	-1
0	0	0

0	-1	0
0	0	0
0	1	0

Problems 2-3: Compute a Derivative

- Same groupwork routine as usual.
- If your group has a question or is stuck, @mention me in your group's text channel.

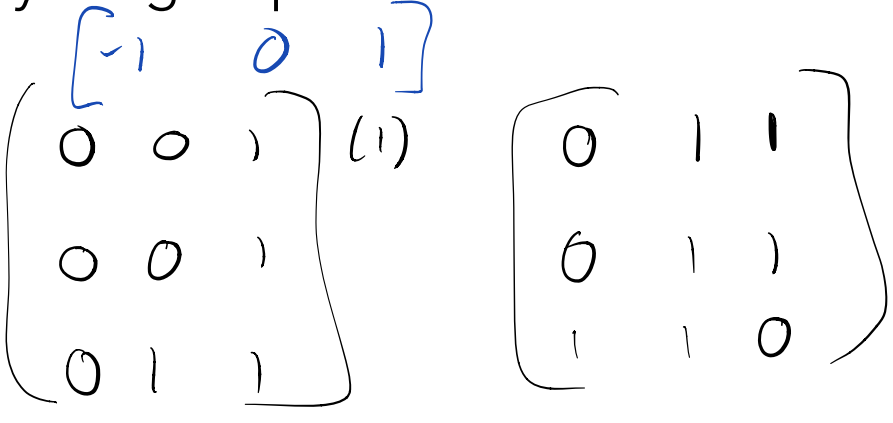
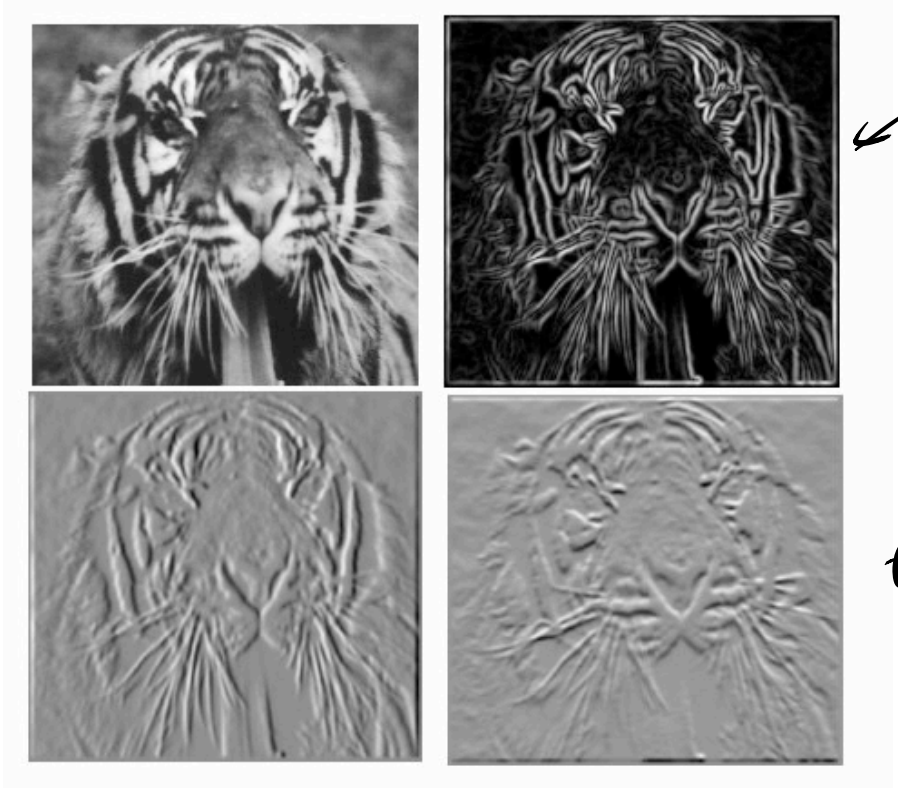
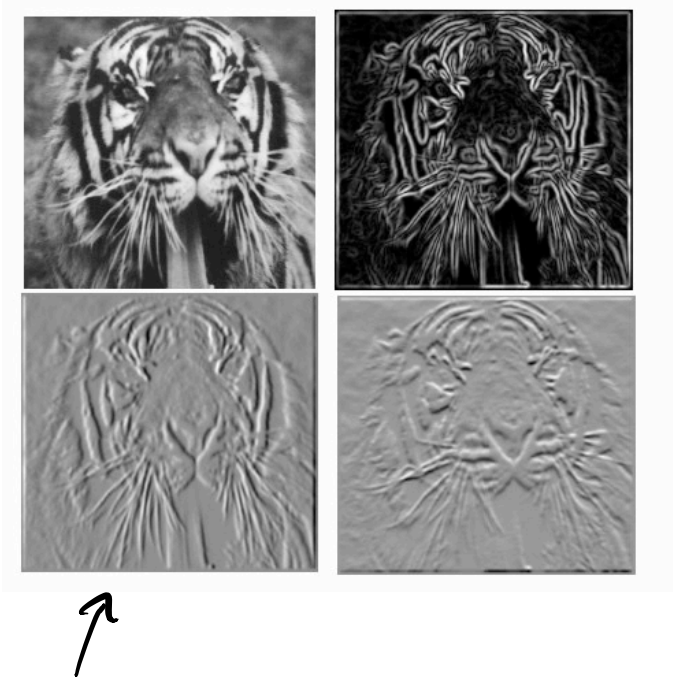


Image Gradient: Visually

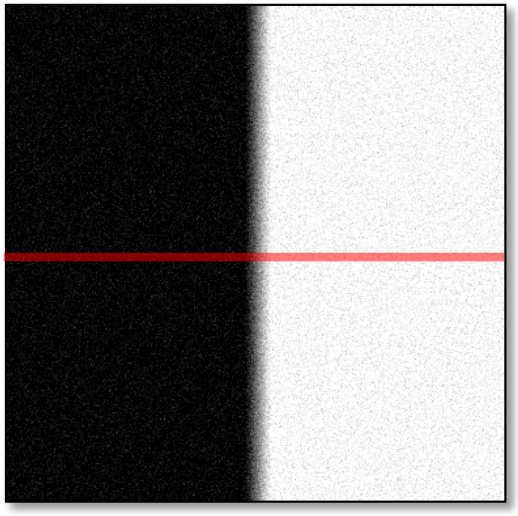


Aside: why are the derivatives grayish, not blackish?

- Images are nonnegative
- Derivatives can be negative!
- Scale and shift derivative values to display in the range 0-255
  - Gray is zero,
    darker is negative,
    lighter is positive

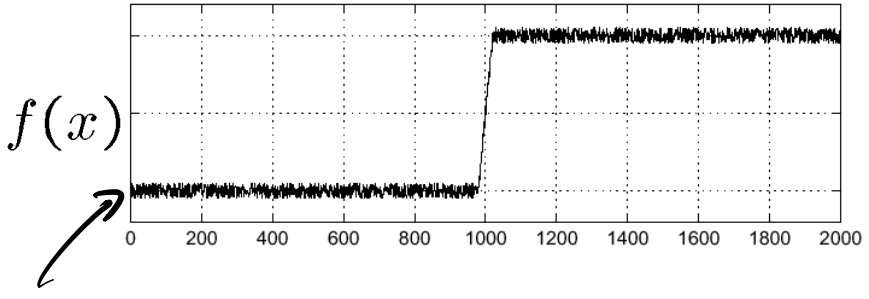


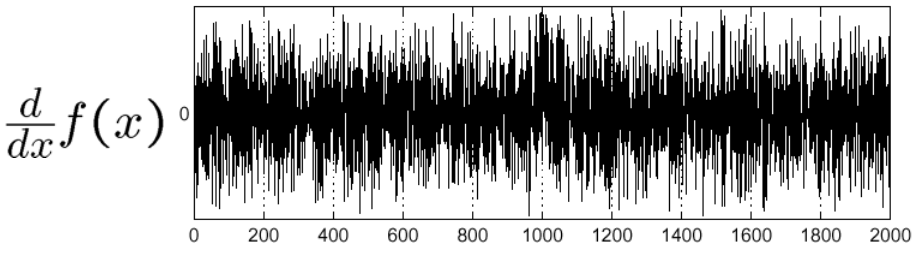
Images (still) aren't perfect



Noisy image

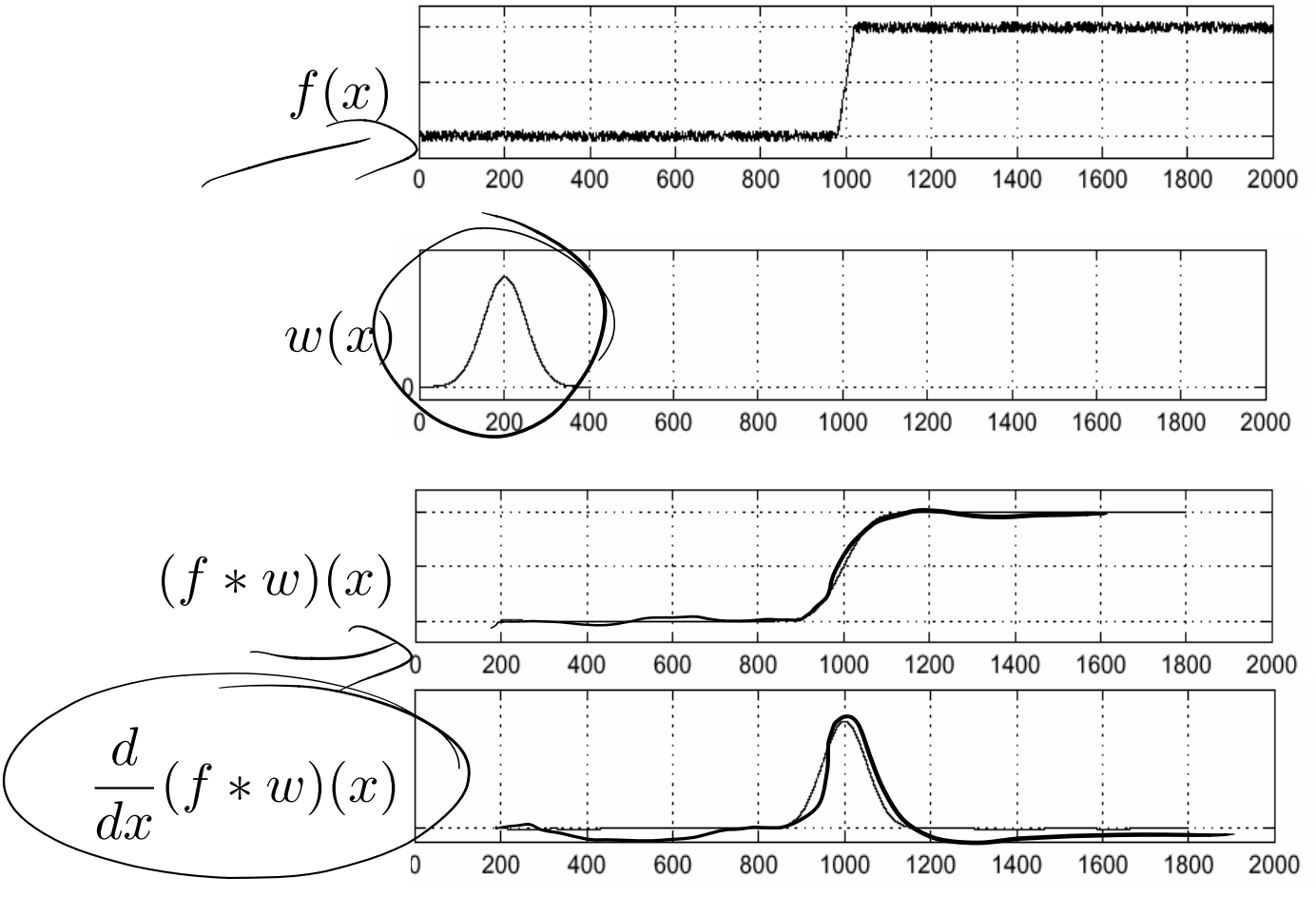
scanline





Can you find the edge?

A solution: smooth it first



An Edge Detection Filter

- Blur, then take the derivative:

$$f * \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

An Edge Detection Filter

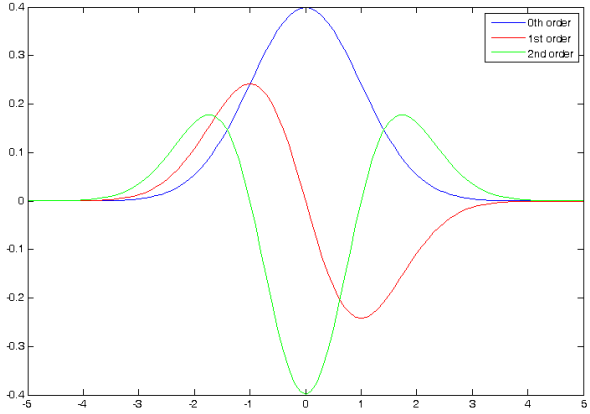
- Blur, then take the derivative:

$$f * \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Or, do the composition in the continuous domain then build a discrete approximation:

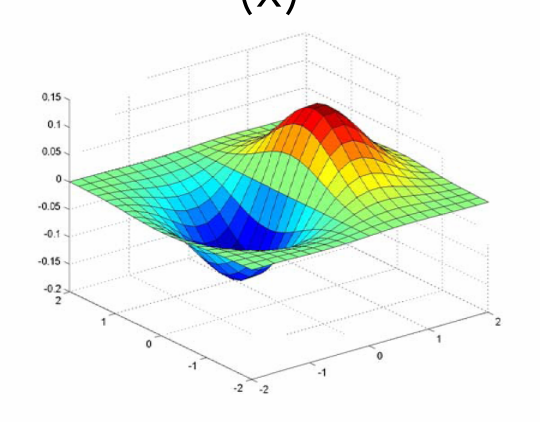
$$G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$G'_\sigma(x) = \frac{d}{dx} G_\sigma(x) = -\frac{1}{\sigma}\left(\frac{x}{\sigma}\right) G_\sigma(x)$$

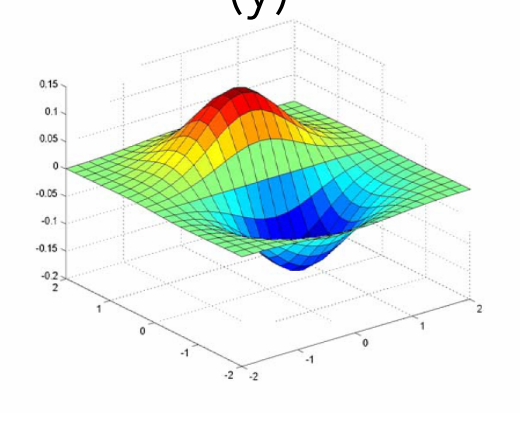


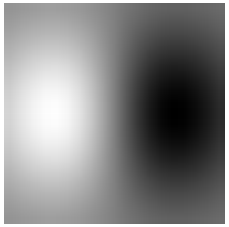
Derivative-of-Gaussian Filter

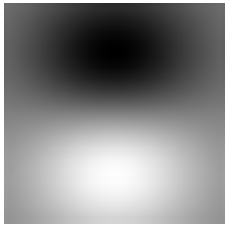
(x)



(y)







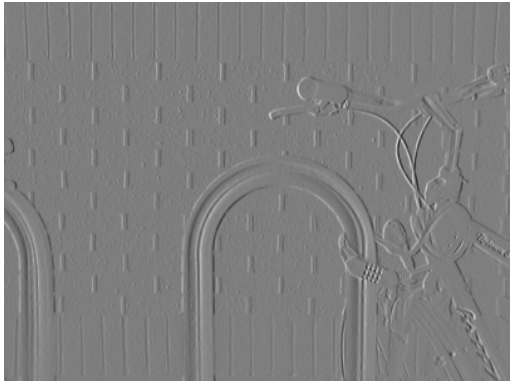
Sobel filter: a 3x3 approximation of the DoG:

$$\frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

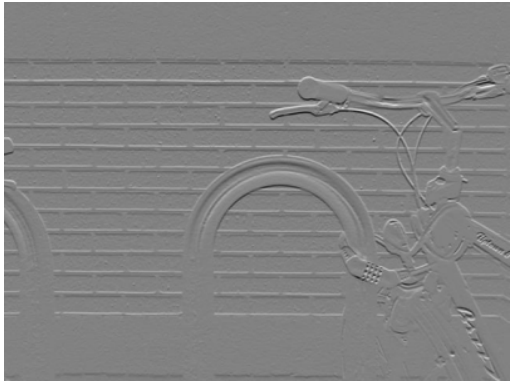
$$\frac{1}{8}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Sobel filter: example

input

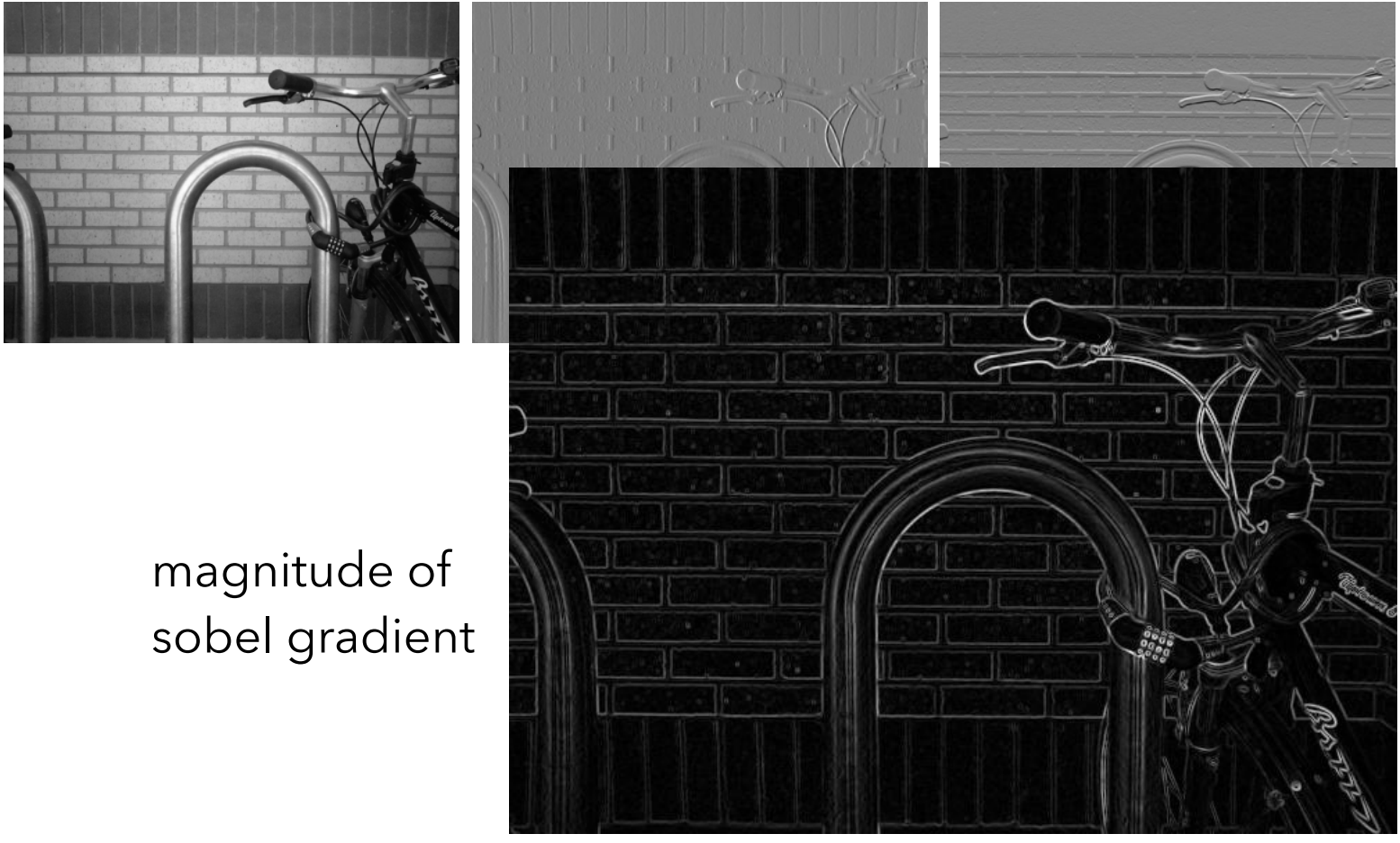


x sobel filtered



y sobel filtered

Sobel filter: example



magnitude of
sobel gradient

Edge Detection

- Fancier edge detectors exist, to try to:
  - more precisely localize the edges (sub-pixel)
  - detect only "salient" edges

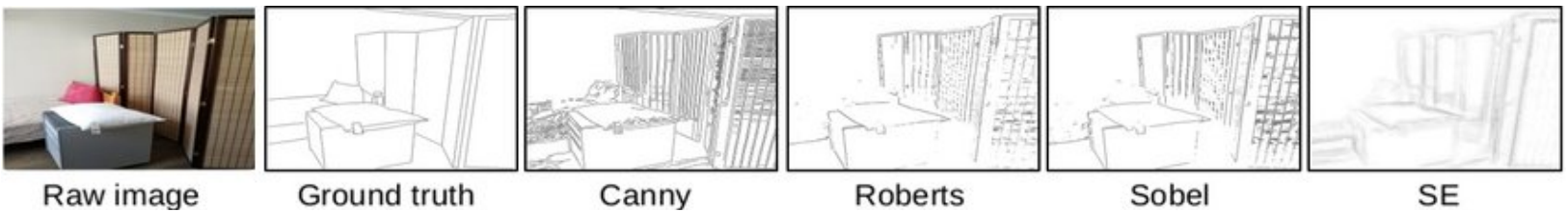


Figure source: Sun et al.: Structural Edge Detection: A Dataset and Benchmark

Questions?

Problem 4

- Derive the Sobel filter for yourself.