Lecture 4. Edge Detection

A simple edge detector

Juan Carlos Niebles and Jiajun Wu

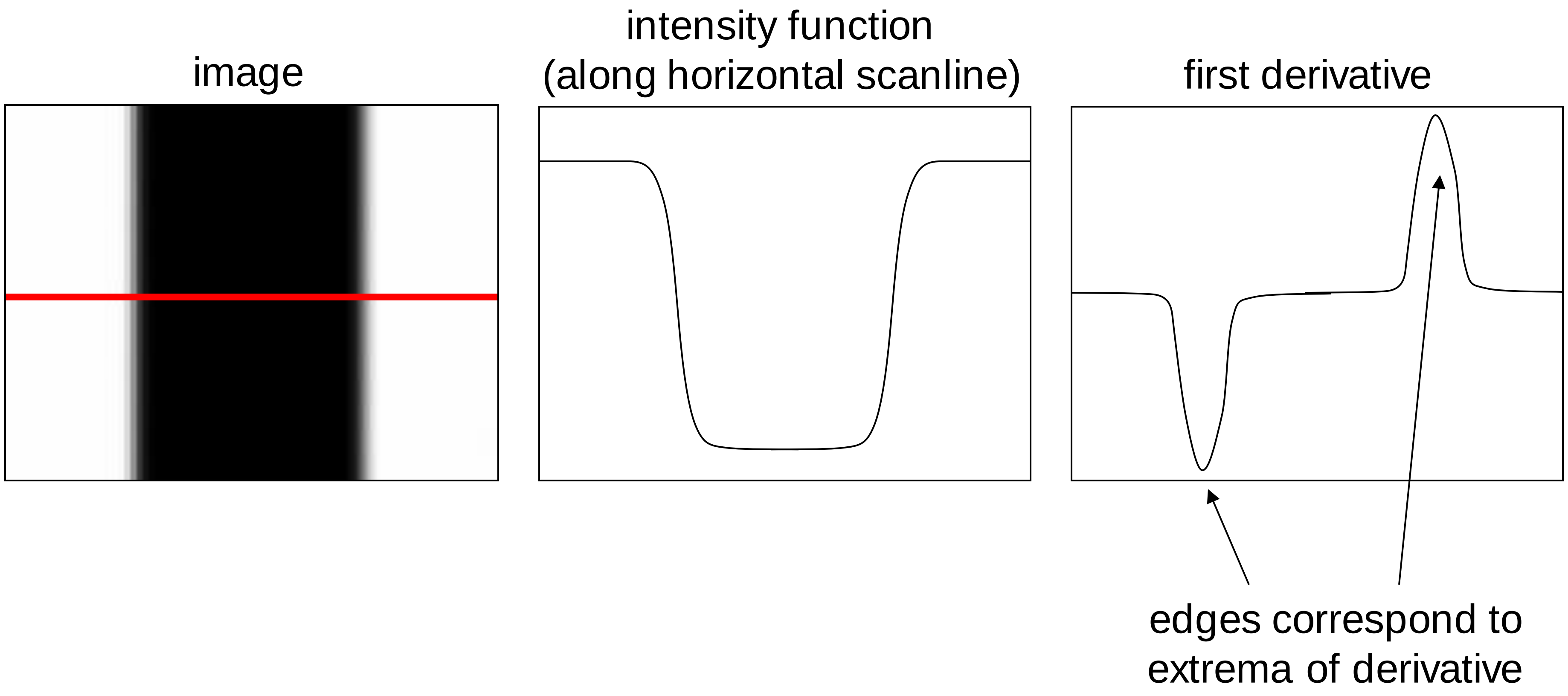
CS131 Computer Vision: Foundations and Applications

What will we learn today?

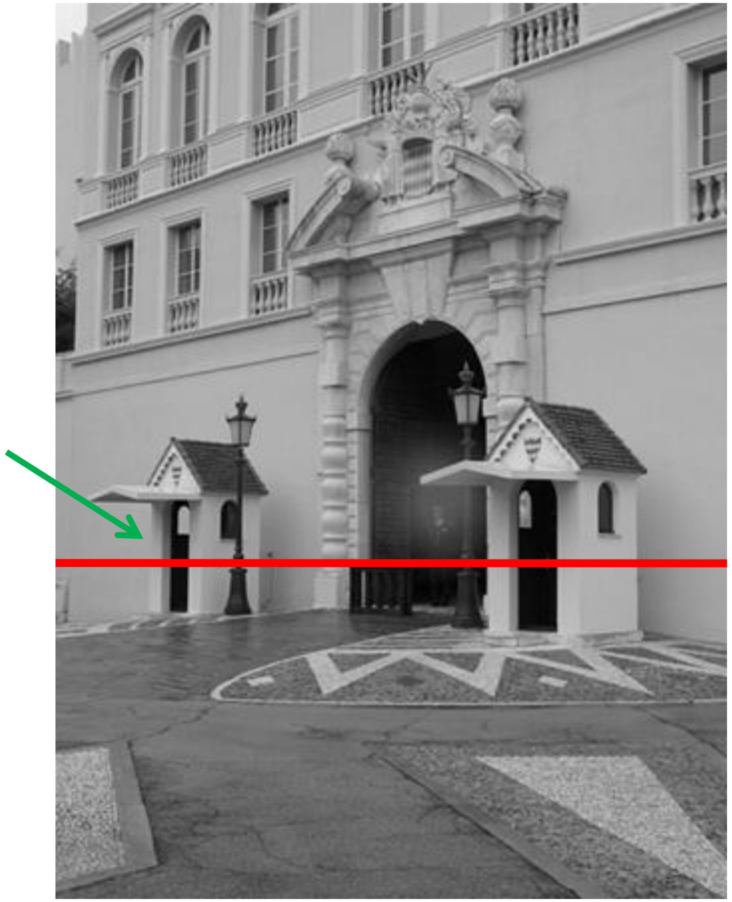
- Characterizing edges
- Effects of noise on edge detection
- Design principles of a good edge detector

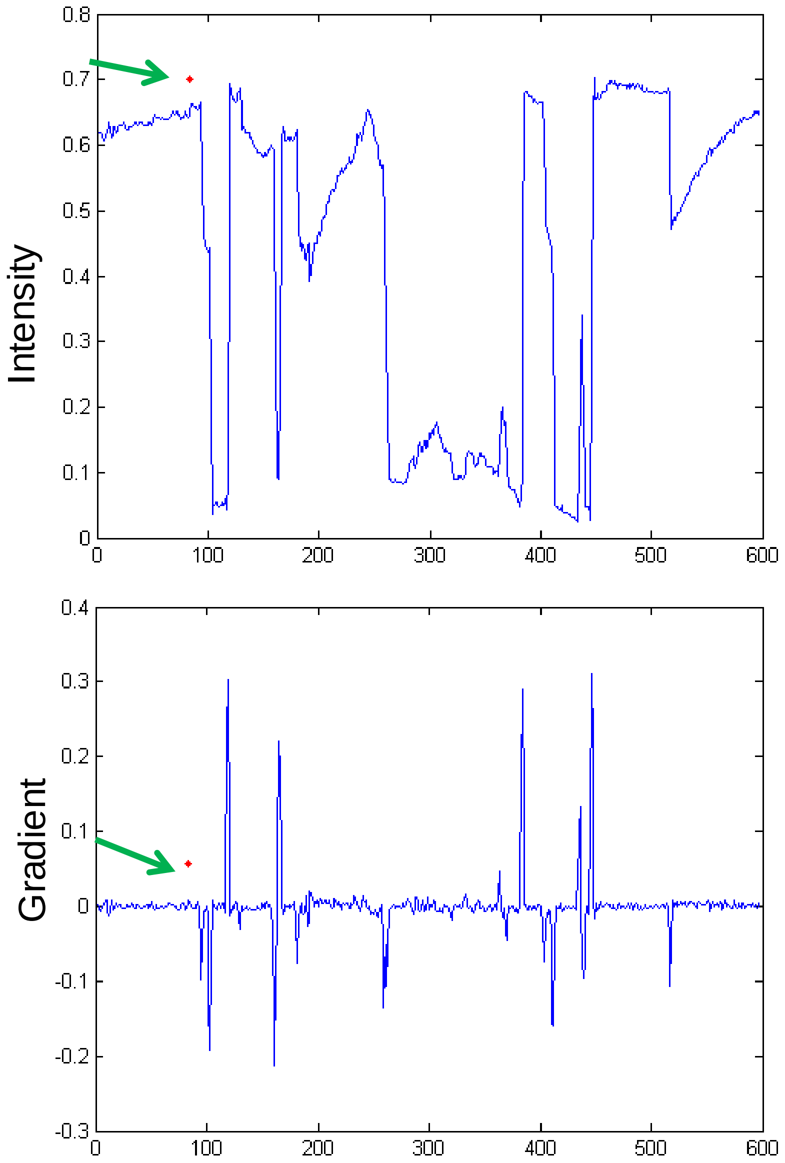
Characterizing edges

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



Intensity profile

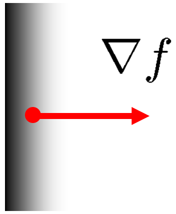




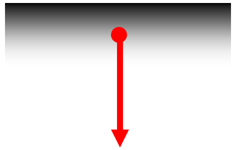
Source: D. Hoiem

Image gradient

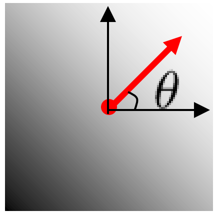
- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$



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



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



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

The gradient vector points in the direction of most rapid increase in intensity

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

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

Source: Steve Seitz

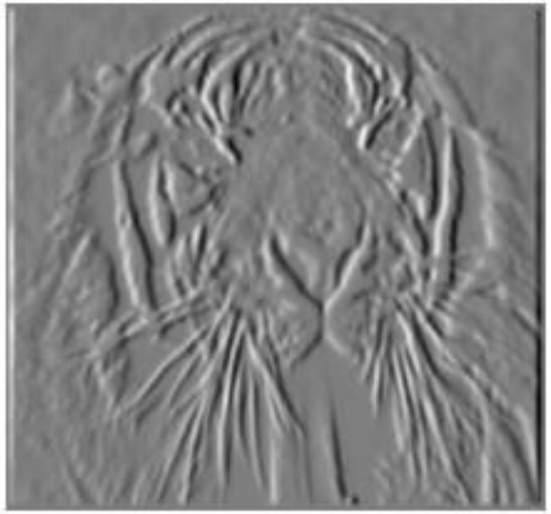
Discrete derivative/gradient: example

- Which one is the gradient in the x-direction? How about y-direction?

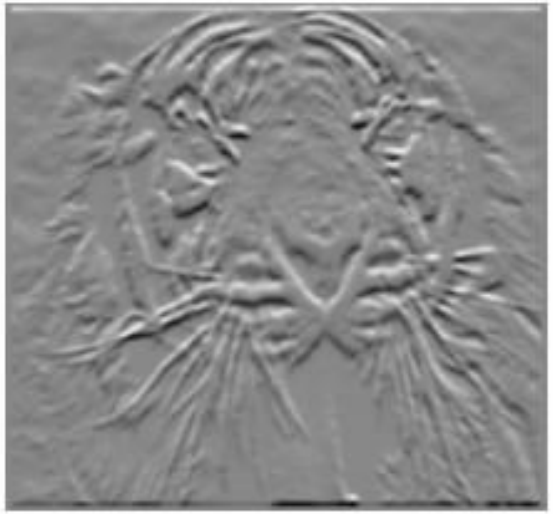
Original Image

Gradient magnitude

x-direction



y-direction

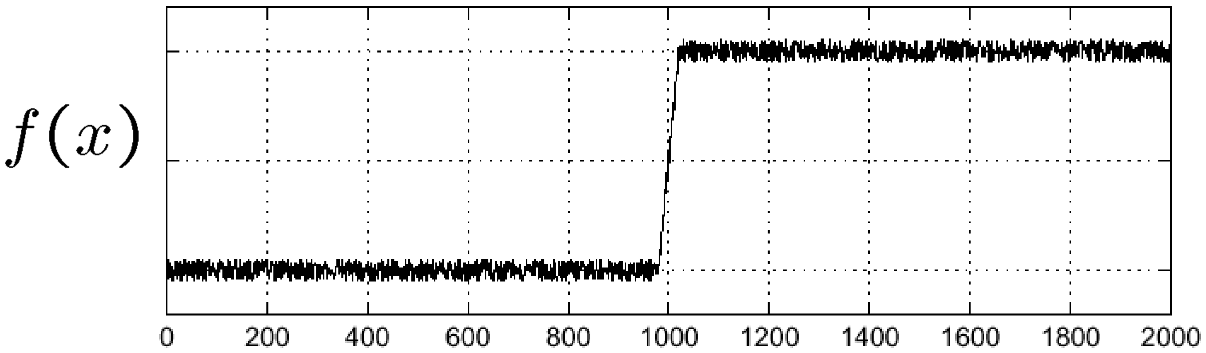


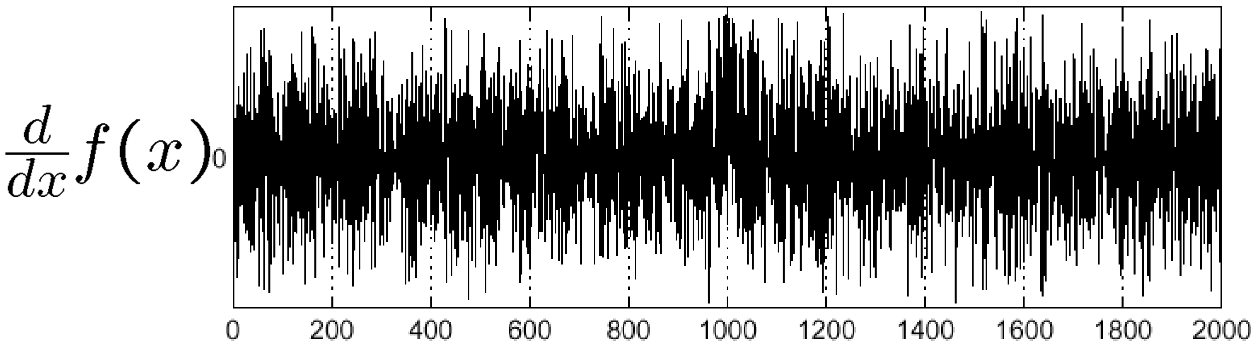
What will we learn today?

- Characterizing edges
- Effects of noise on edge detection
- Design principles of a good edge detector

Effects of noise

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

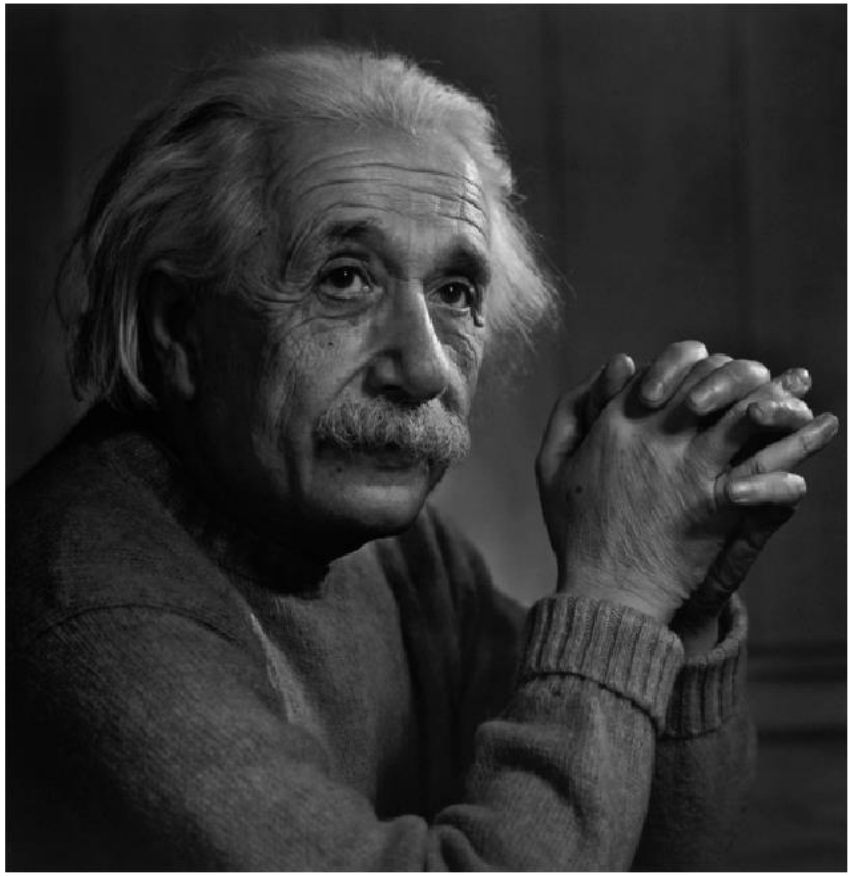


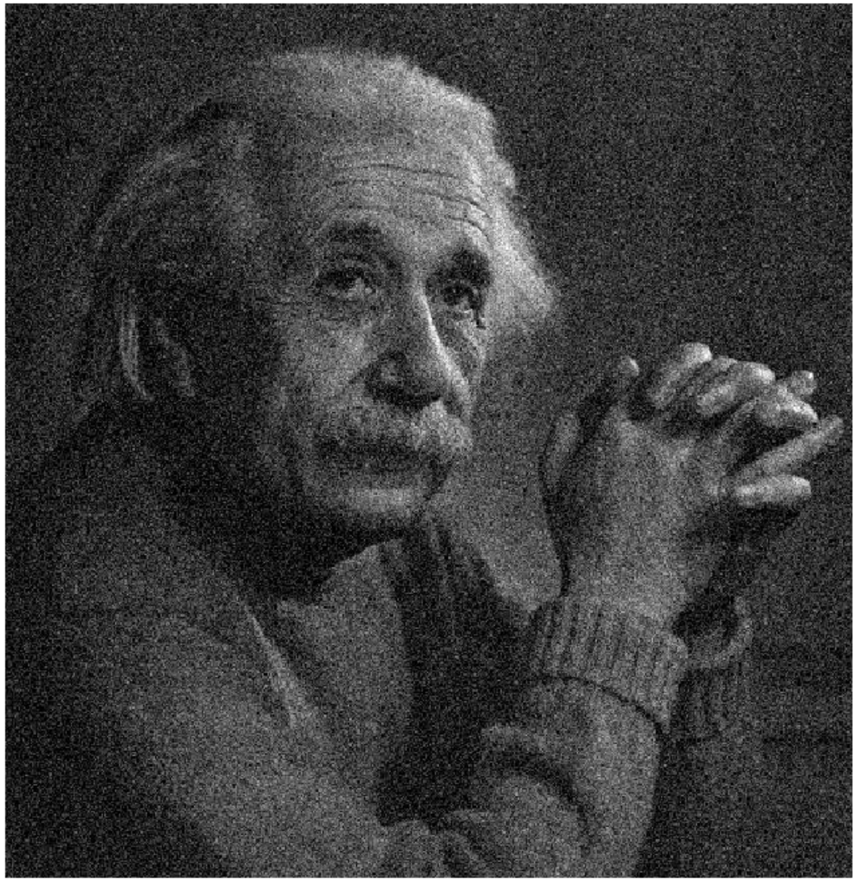


Where is the edge?

Source: S. Seitz

Effects of noise





Effects of noise

- Discrete gradient 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 is to be done?
  - Smoothing the image should help, by forcing pixels different to their neighbors (=noise pixels?) to look more like neighbors

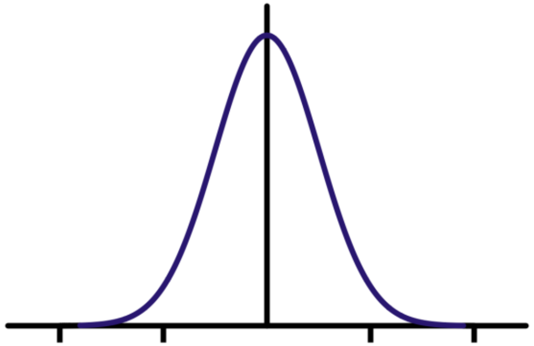
Source: D. Forsyth

Smoothing filters

- Mean smoothing

$$\begin{bmatrix}1\\1\\1\end{bmatrix} \qquad \begin{bmatrix}1 & 1 & 1\end{bmatrix}$$

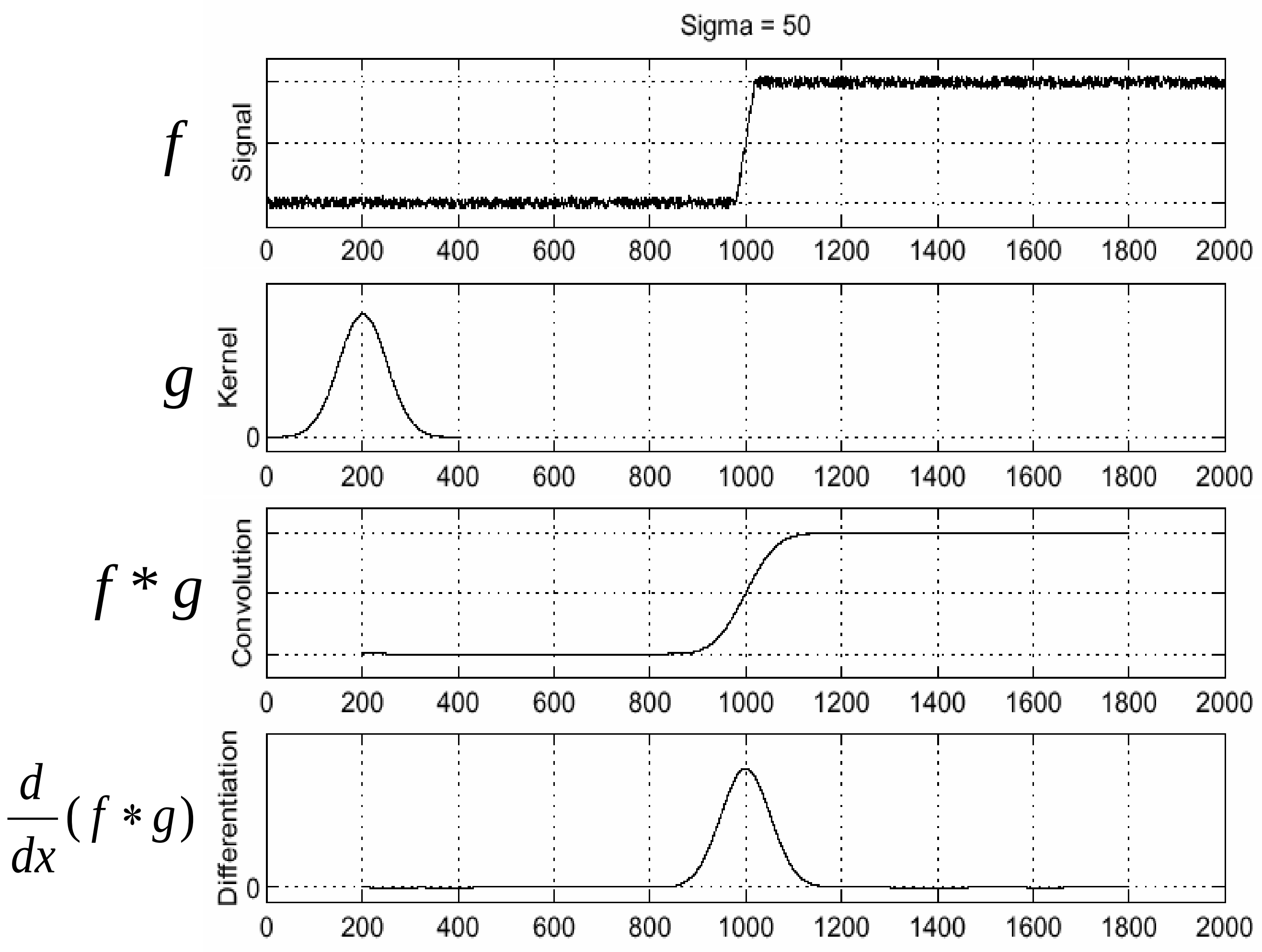
- Gaussian smoothing



$$\begin{bmatrix}1\\2\\1\end{bmatrix} \qquad \begin{bmatrix}1 & 2 & 1\end{bmatrix}$$

Slide credit: Steve Seitz

A simple edge detector: smoothing + peaks



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

Source: S. Seitz

Derivative theorem of convolution

- This theorem gives us a very useful property:

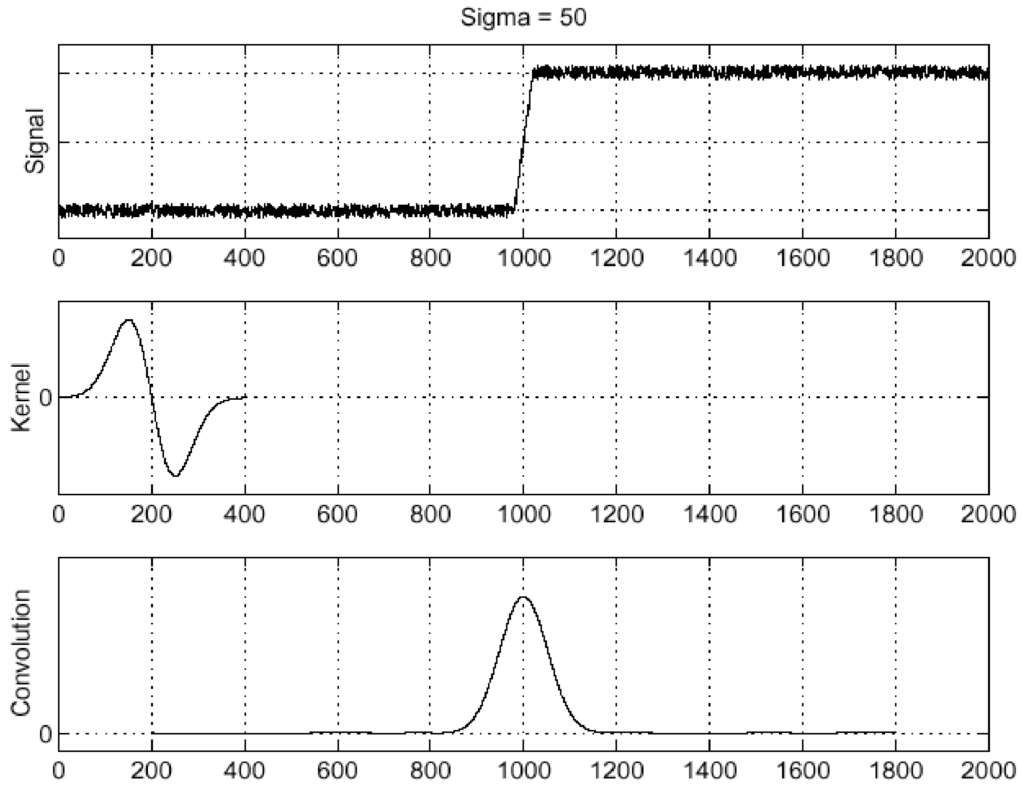
$$\frac{d}{dx}(f * g) = f * \frac{d}{dx} g$$

- This saves us one operation:

$$f$$

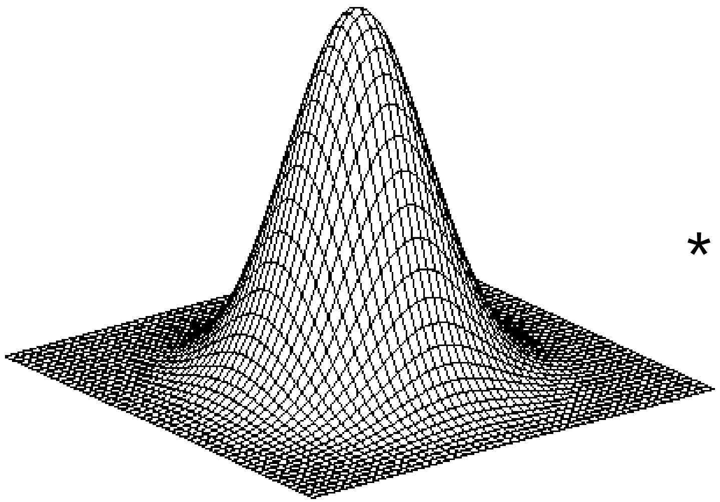
$$\frac{d}{dx} g$$

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

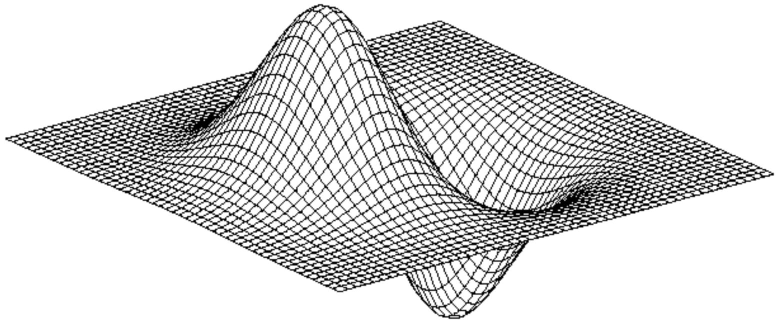


Source: S. Seitz

Derivative of Gaussian filter



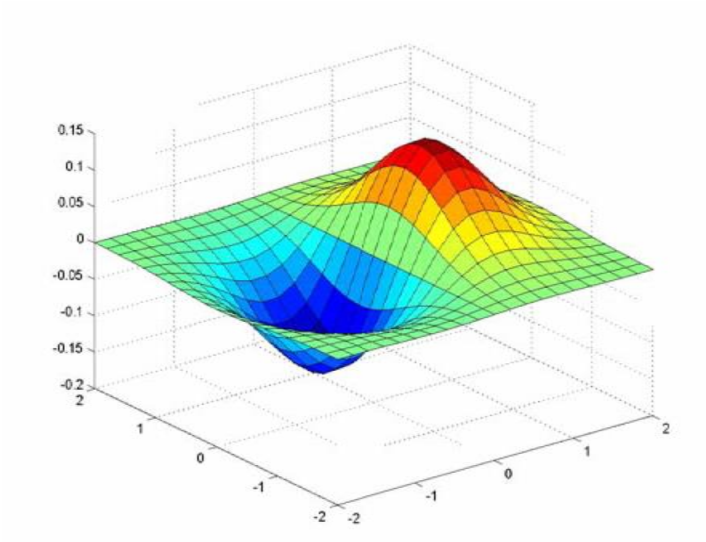
* [1 0 -1] =



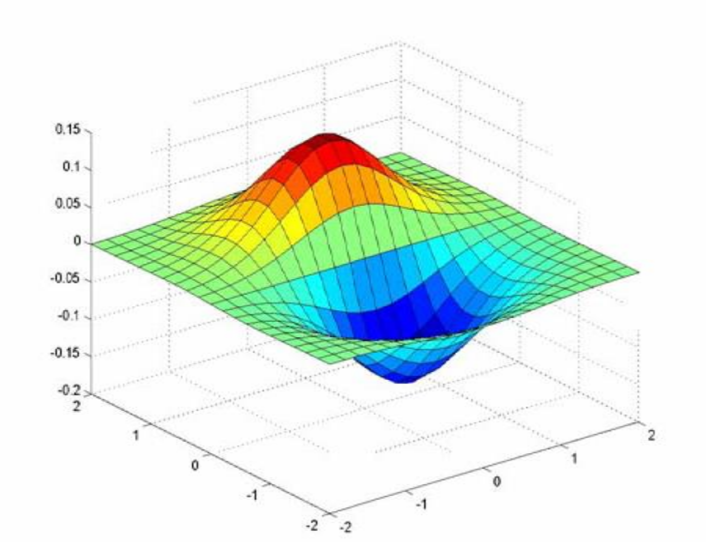
2D-gaussian

x - derivative

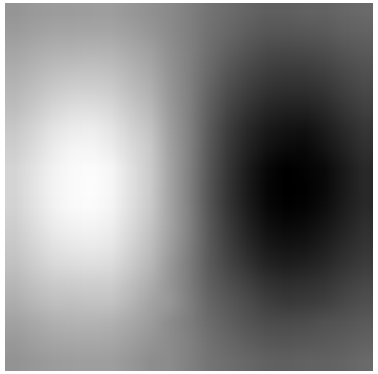
Derivative of Gaussian filter

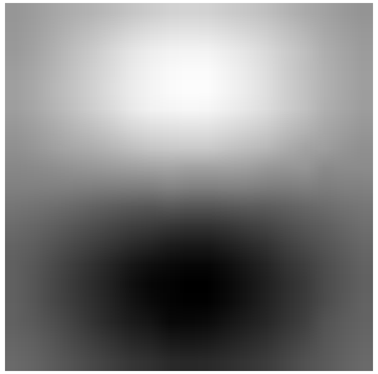


x-direction

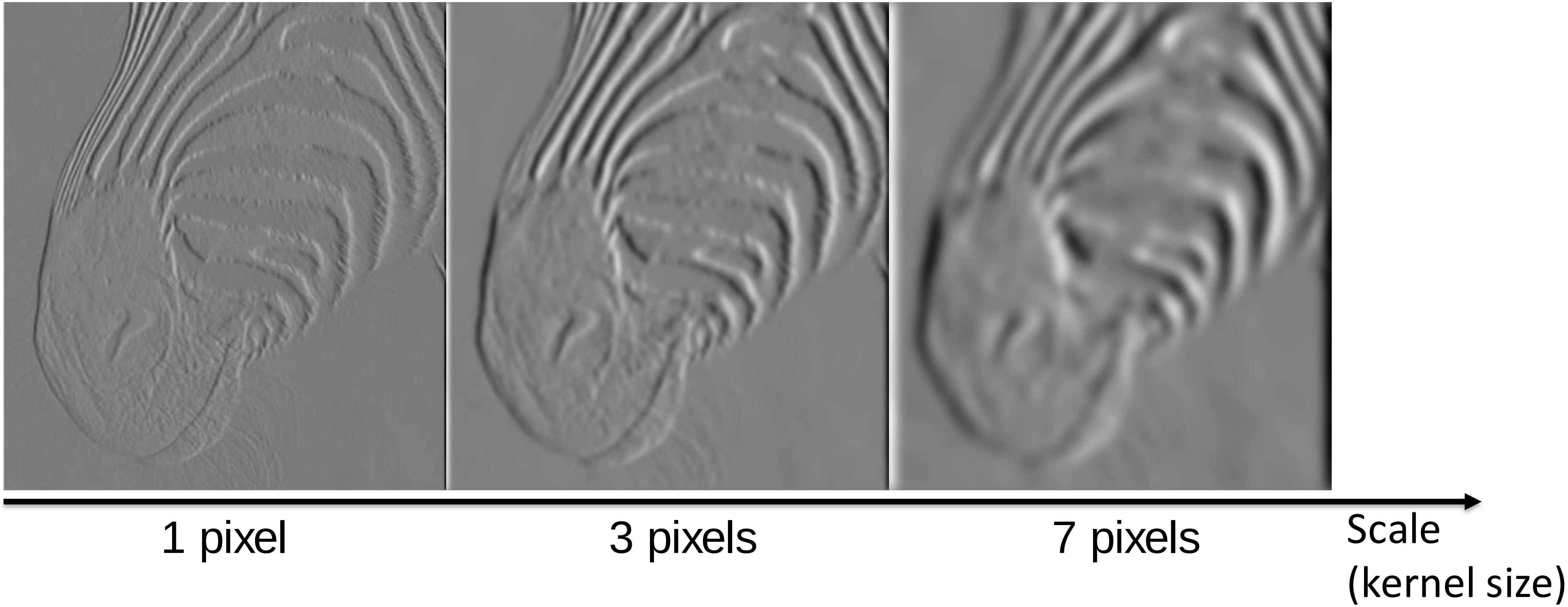


y-direction





Tradeoff between smoothing and localization



- Stronger smoothing removes noise, but blurs edges.
- Finds edges at different “scales”.

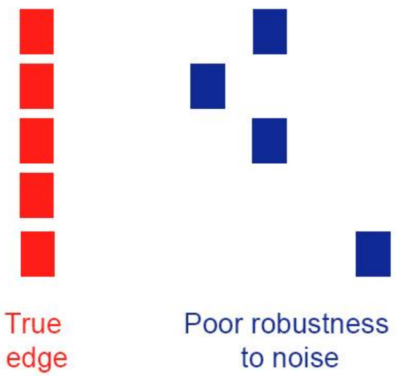
Source: D. Forsyth

What will we learn today?

- Characterizing edges
- Effects of noise on edge detection
- Design principles of a good edge detector

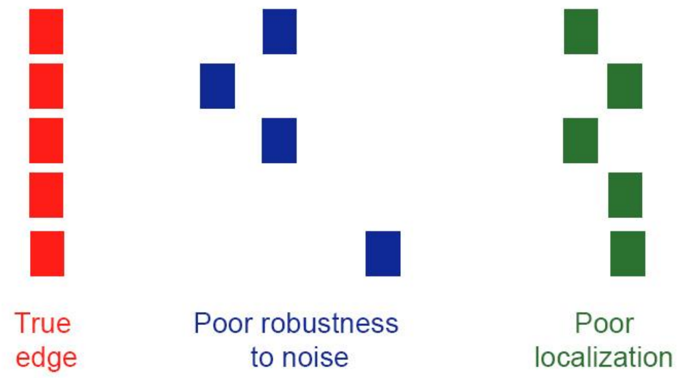
Designing an edge detector

- Criteria for an "optimal" edge detector:
  - **Good detection:** the optimal detector must minimize the probability of false positives (detecting spurious edges caused by noise), as well as that of false negatives (missing real edges)



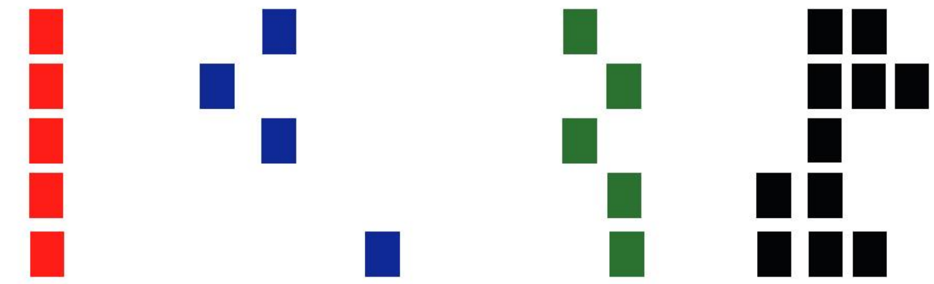
Designing an edge detector

- Criteria for an "optimal" edge detector:
  - **Good detection:** the optimal detector must minimize the probability of false positives (detecting spurious edges caused by noise), as well as that of false negatives (missing real edges)
  - **Good localization:** the edges detected must be as close as possible to the true edges



Designing an edge detector

- Criteria for an “optimal” edge detector:
  - **Good detection:** the optimal detector must minimize the probability of false positives (detecting spurious edges caused by noise), as well as that of false negatives (missing real edges)
  - **Good localization:** the edges detected must be as close as possible to the true edges
  - **Single response:** the detector must return one point only for each true edge point; that is, minimize the number of local maxima around the true edge



Summary

- Characterizing edges
- Effects of noise on edge detection
  - Smoothing filters
  - Design an edge detector with image smoothing
- Design principles of a good edge detector