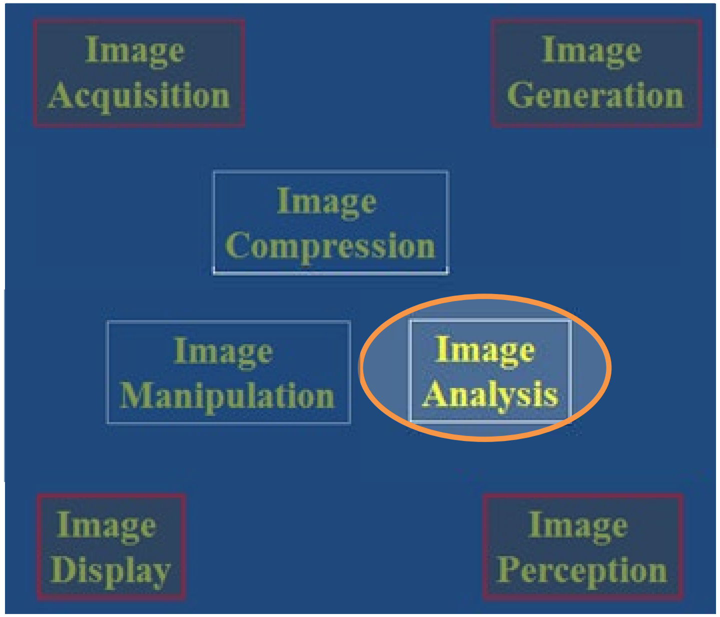
Digital Image Processing

Edge Detection



Brent M. Dingle, Ph.D. 2015
Game Design and Development Program
Mathematics, Statistics and Computer Science
University of Wisconsin - Stout

Material in this presentation is largely based on/derived from presentations by: Sventlana Lazebnik, and Noah Snavely

Lecture Objectives

- Previously
  - Filtering
  - Interpolation
  - Warping
  - Morphing
  - Compression
- Today
  - Edge Detection

Image Manipulation and Enhancement

Image Compression

Image Analysis

Recall: Digital Image Processing

- **Digital Image Processing (DIP)**
  - **Is *computer* manipulation of pictures, or images, that have been converted into numeric form**

  - Typical operations include
    - Image Compression
    - Image Warping
    - Contrast Enhancement
    - Blur Removal
    - Feature Extraction
    - Pattern Recognition

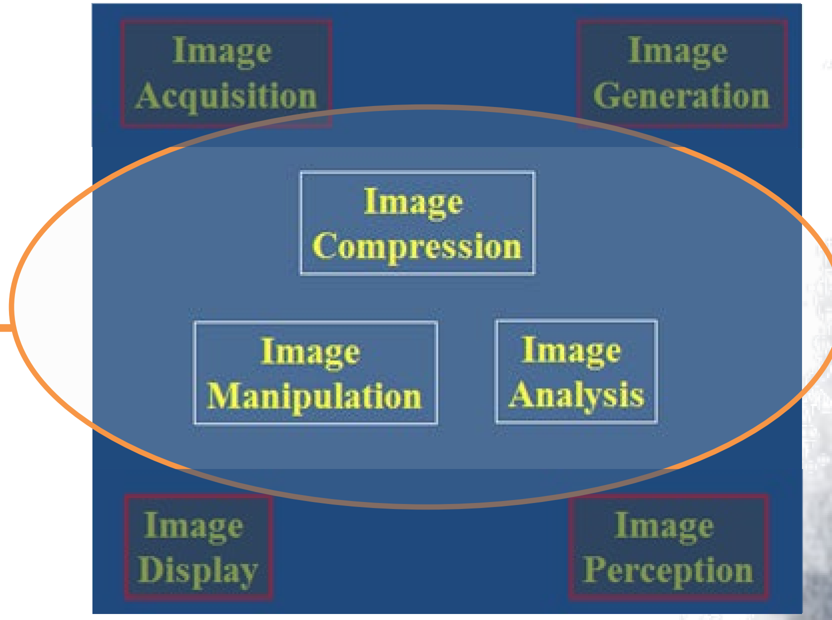


Image Processing **Goals**

- Digital image processing is a **subclass of signal processing specifically concerned with picture images**

– Goals
  - To improve image quality for
    – human perception (subjective)
    – computer interpretation (objective)
  - Develop methods and applications
    – to compress images
    – to provide efficient storage and transmission

Your projects should relate to these goals

Related Fields

- Computer Graphics
- Computer Vision
- Artificial Intelligence
- … others…

Distinction Between Fields

- Image processing is **not just image-in/image-out**
- **But it does work to distinguish relational aspects of some fields**

Output / Input	Image	Description
Image	Image Processing	Computer Vision
Description	Computer Graphics	Artificial Intelligence

Related Fields

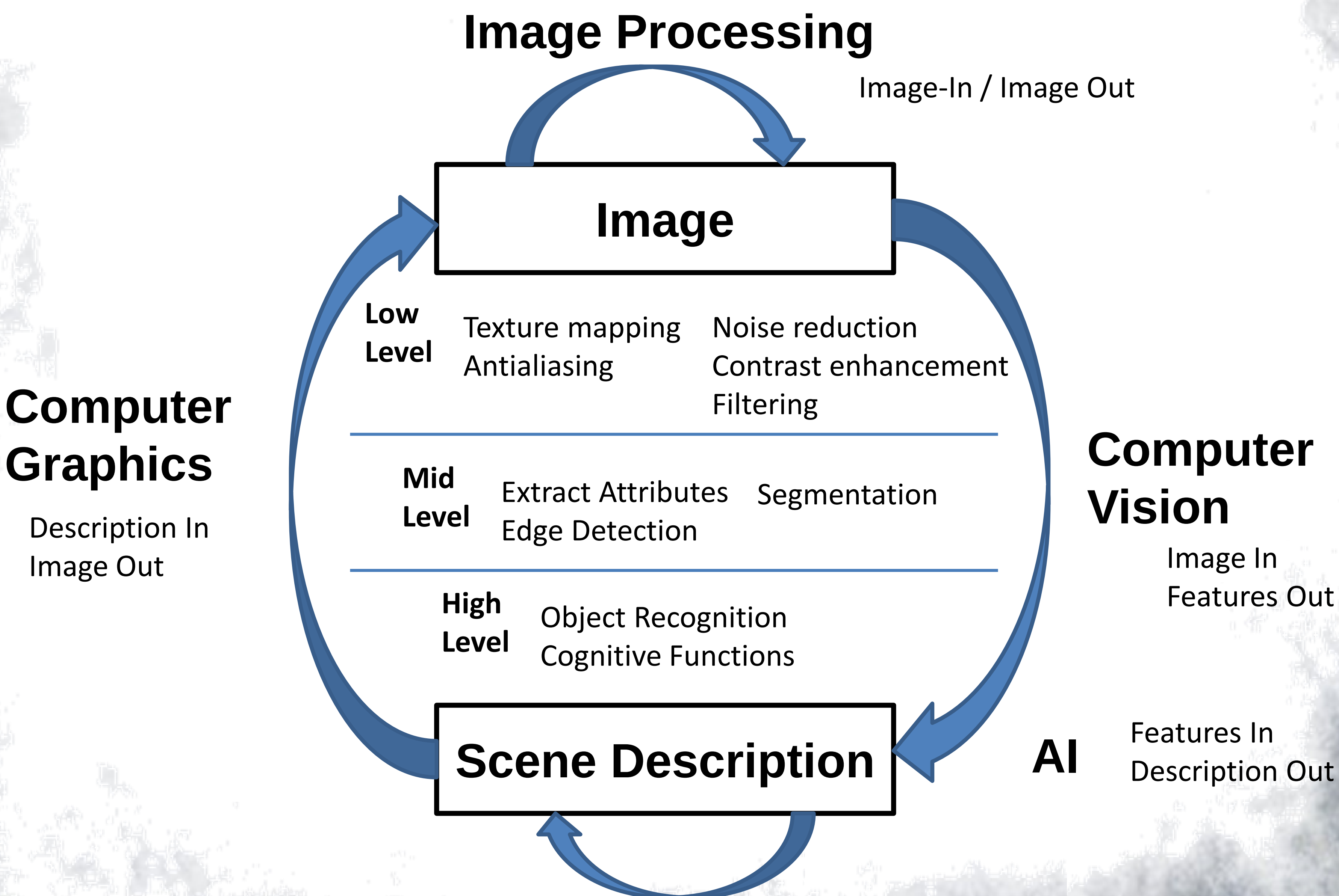
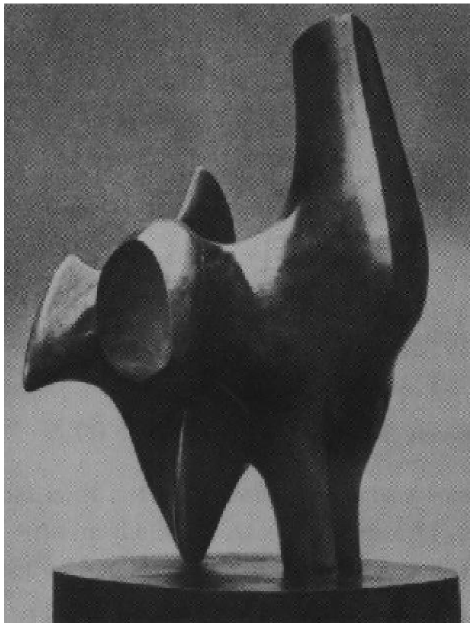
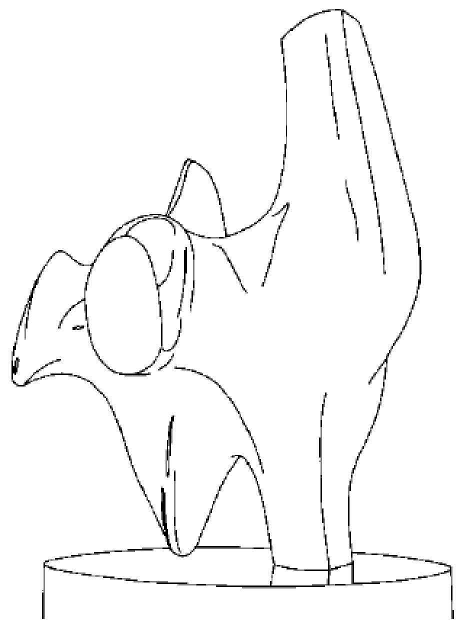


Image Analysis: Edge Detection

- Edge Detection
  - – Relates to filtering
  - – Used in various types of image analysis
  - – Useful in other fields
    - Such as Computer Vision

Edge Detection

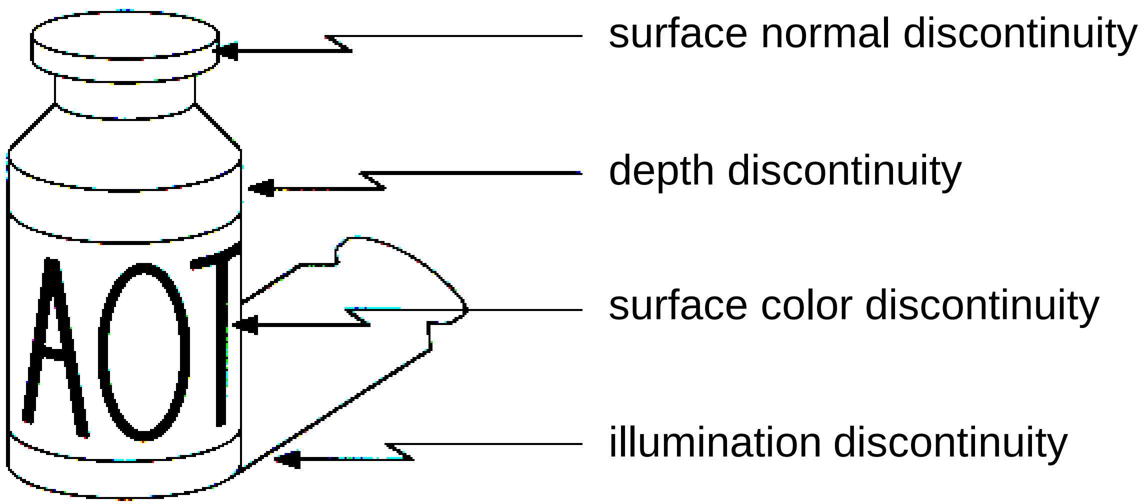




- **Goal**: Identify sudden changes (discontinuities) in an image
  - Much of the 'shape' information of the image can be realized in the edges
- **Ideal Result**: an artist's line drawing
  - *aside: who can use knowledge beyond a single image*

contour line drawing image from: https://www.youtube.com/watch?v=Eu9sUWLfGng

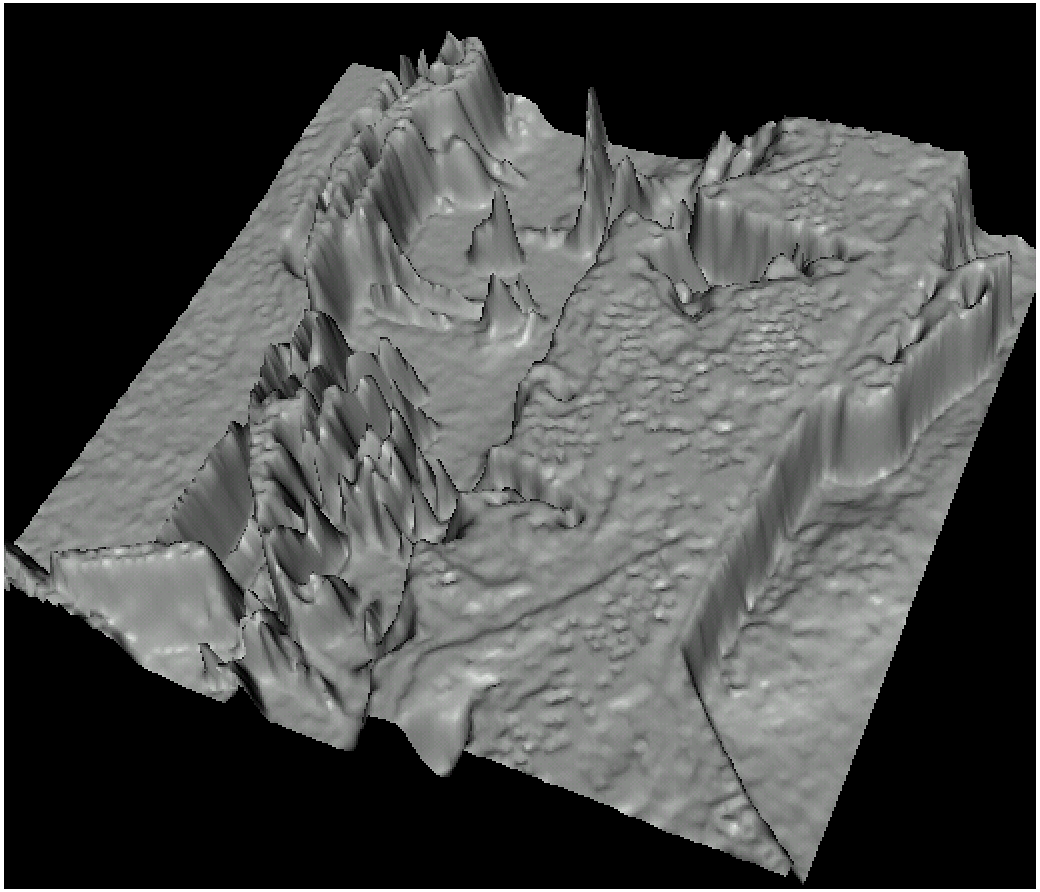
What Makes an Edge?



- Edges are “caused” by various discontinuities

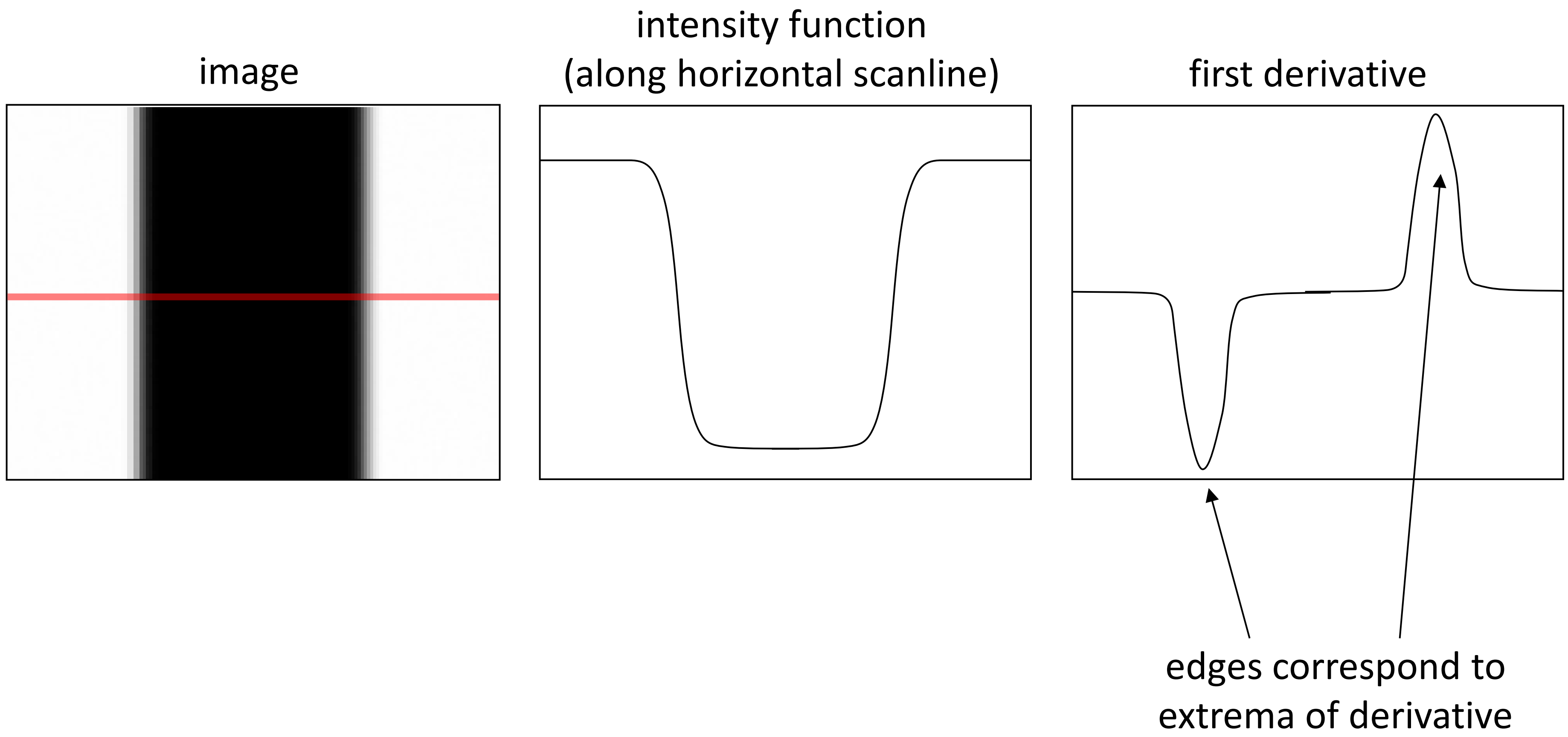
image source: Steve Seitz

Discontinuity --> Sharp Change



Edge Characterization

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



Images from: Svetlana Lazebnik

Image Derivatives

- How can we differentiate a digital image F(x, y)?
- Option 1:
  - Reconstruct a continuous image, *f*, then compute its derivative
- Option 2:
  - Take a discrete derivative (finite difference, convolution)

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

How can this be implemented as a linear filter?

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

H_x

$\frac{\partial f}{\partial y}$:

H_y

Finite Difference Filters

- Other filters for approximating differentiation exist:

Prewitt: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

Sobel: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

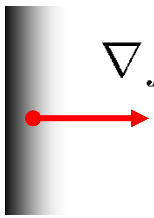
Roberts: $M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

source: K. Grauman

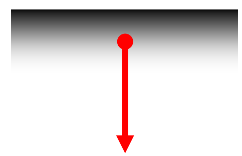
Image Gradient

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

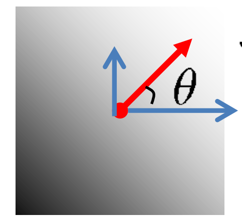
The gradient points in the direction of the most rapid change (increase) in intensity



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



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



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

The edge strength is given by the magnitude of the gradient:

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

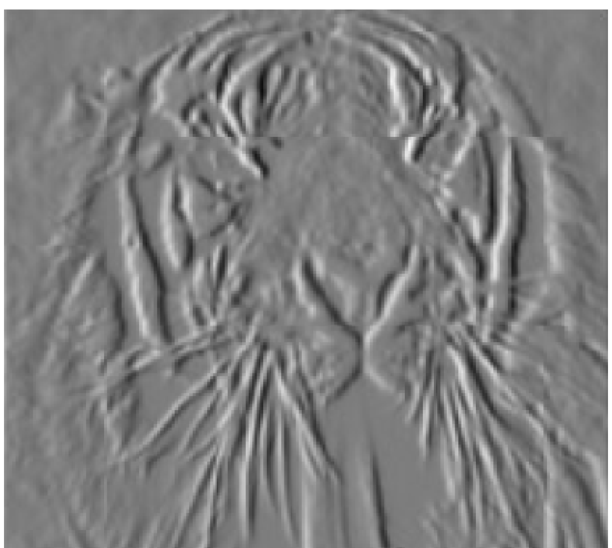
The gradient direction is given by:

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

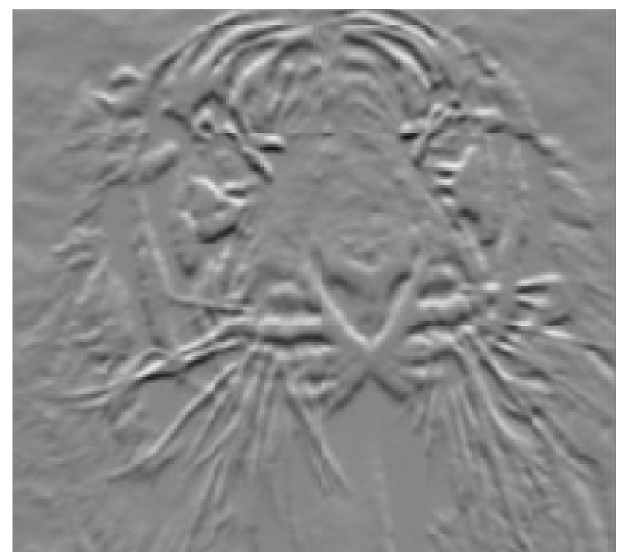
This is perpendicular to the direction of the edge

source: Steve Seitz

Example: Finite Differences



lines in the y-direction

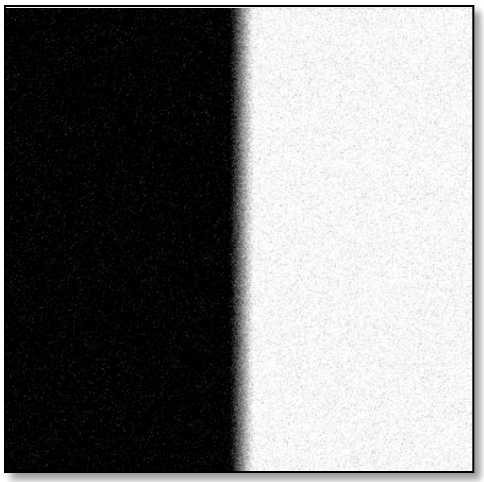


lines in the x-direction

Source: S Lazebnik

Effects of Noise

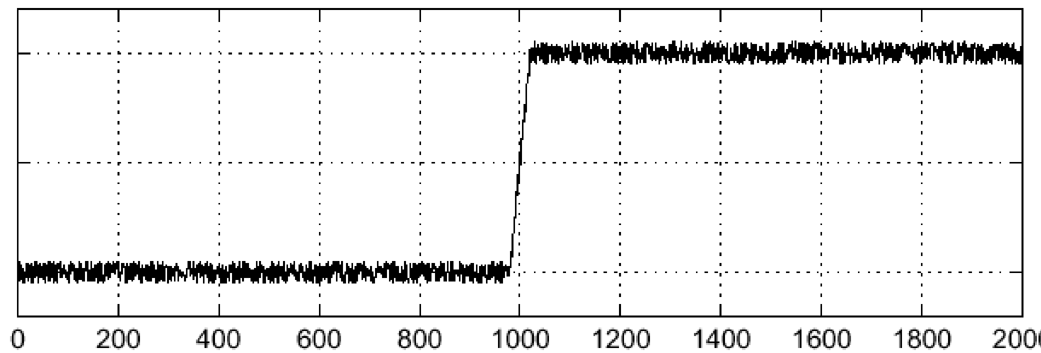
- Consider a single row or column of the image



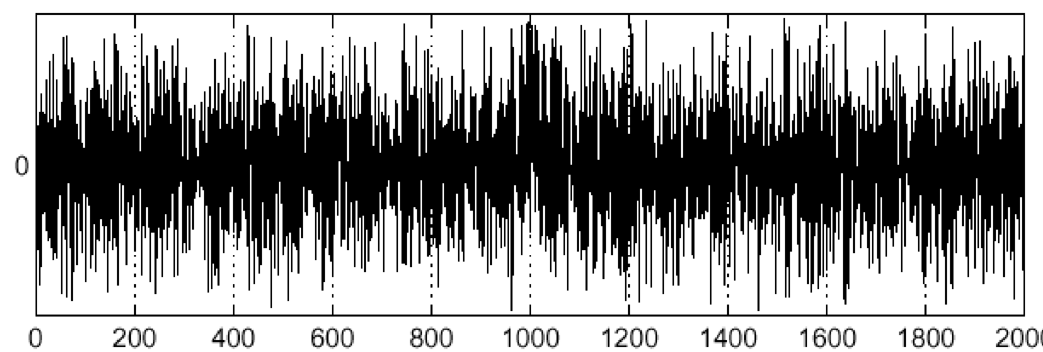
Noisy input image

Plot the intensity as a function of position gives a signal:

$$f(x)$$



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



Where is the Edge?

source: S. Seitz

Effects of Noise

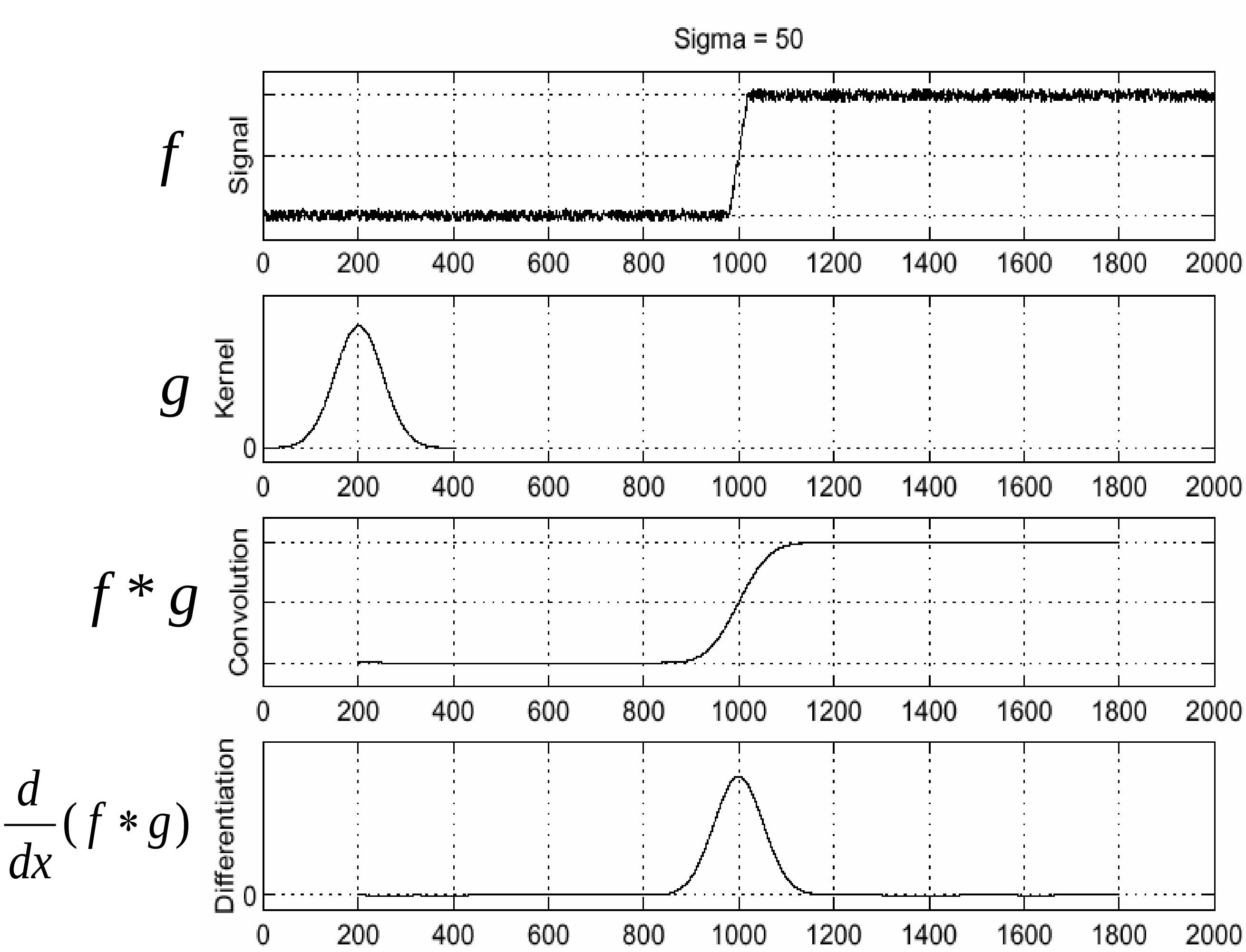
- Finite difference filters respond strongly to noise
  - Image noise results in pixels that look different from their neighbors
  - The “larger” the noise then the stronger the response

- Solution?
  - Smoothing the image
  - Forces pixels different from their nhbrs to look more like their nhbrs
    - i.e. Removes “largeness” of the difference

source: D. Forsyth

Smoothing First

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



source: S. Seitz

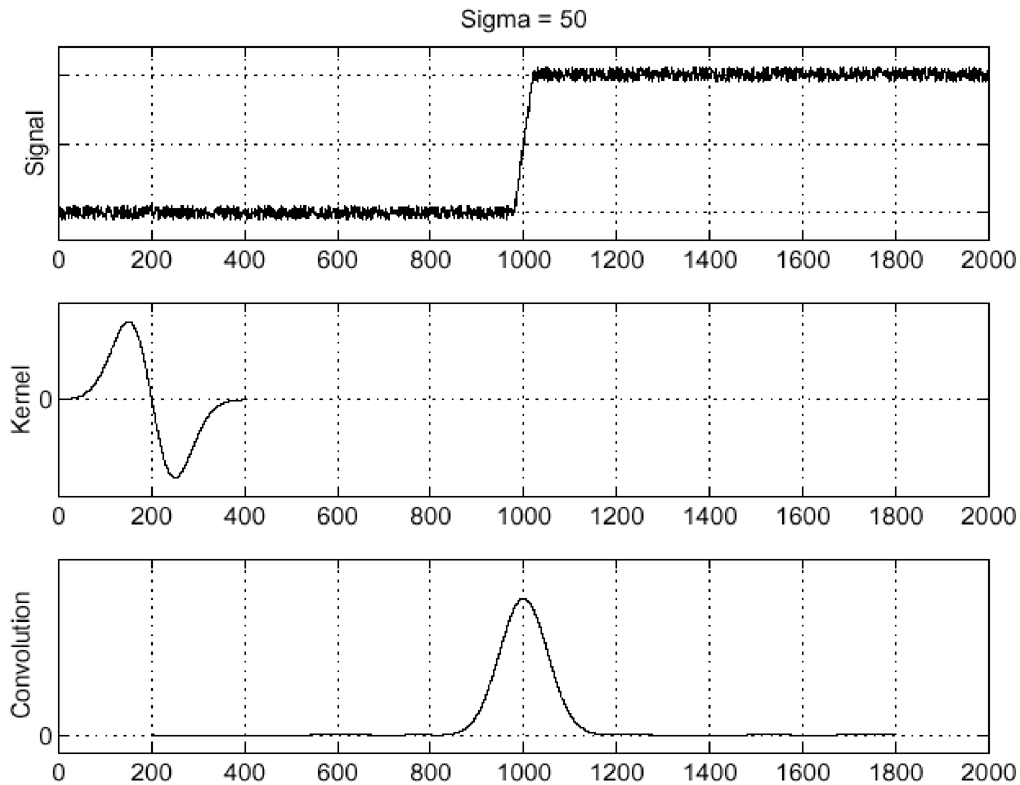
Associative Property of Convolution

- Differentiation is convolution, and convolution is associative: $\frac{d}{dx}(f*g)=f*\frac{d}{dx}g$
- This saves us one operation:

f

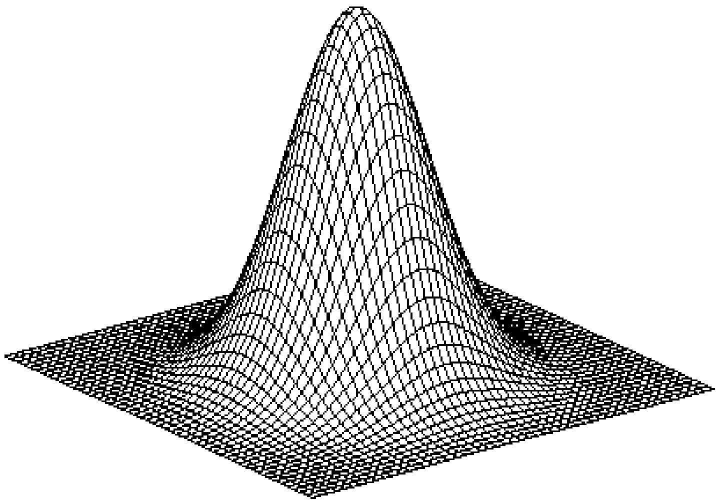
$\frac{d}{dx}g$

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

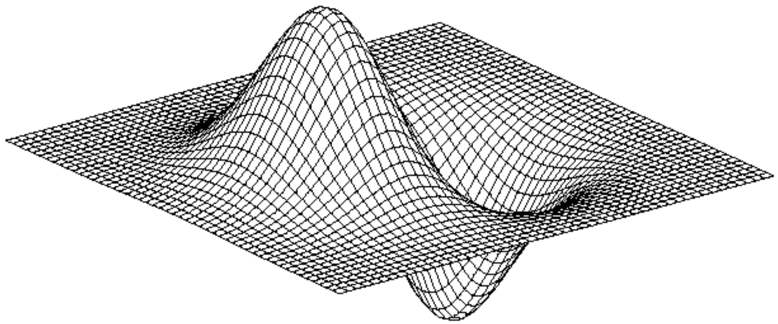


source: S. Seitz

2D Edge Detection Filter



* [1 -1] =



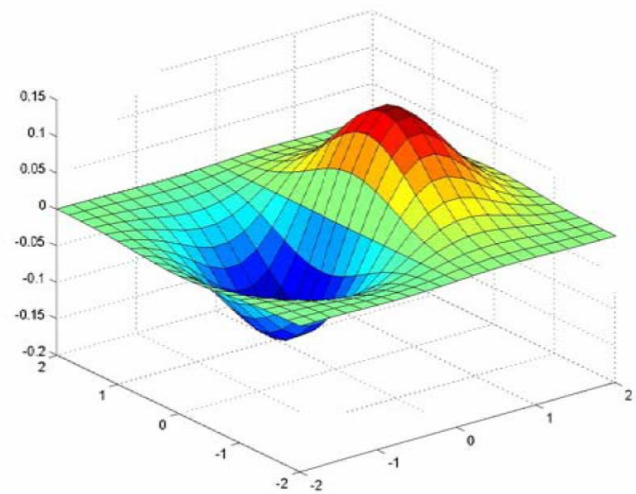
Gaussian

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

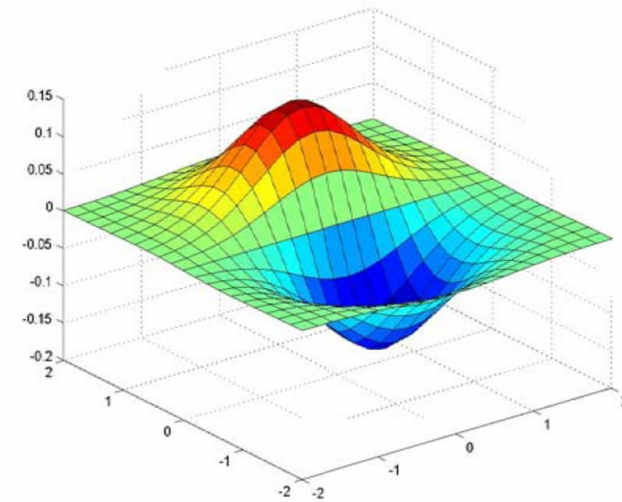
Derivative of Gaussian (x)

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

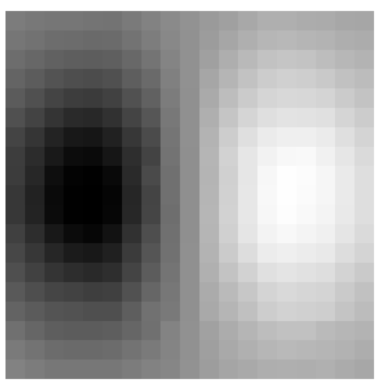
Derivative of Gaussian Filter

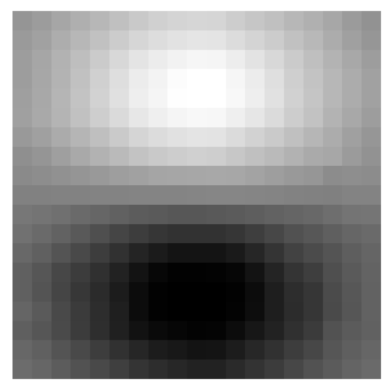


x-direction



y-direction





Which finds horizontal/vertical edges? --- try it and find out!

Options: Sobel Operator

- Remember there are options for approximating the derivative of the Gaussian
  - Prewitt, Sobel, Roberts *(back a few slides)*, results vary

- For illustration consider the Sobel Operator

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

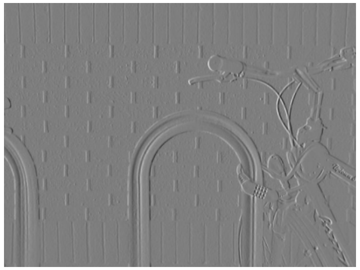
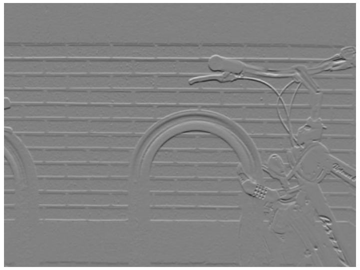
$$s_x \qquad\qquad s_y$$

Most "standard" definition of the Sobel operator omit the 1/8 term

This terms does not matter so much with regard to edge detection

But it ***IS*** needed to calculate the correct gradient value

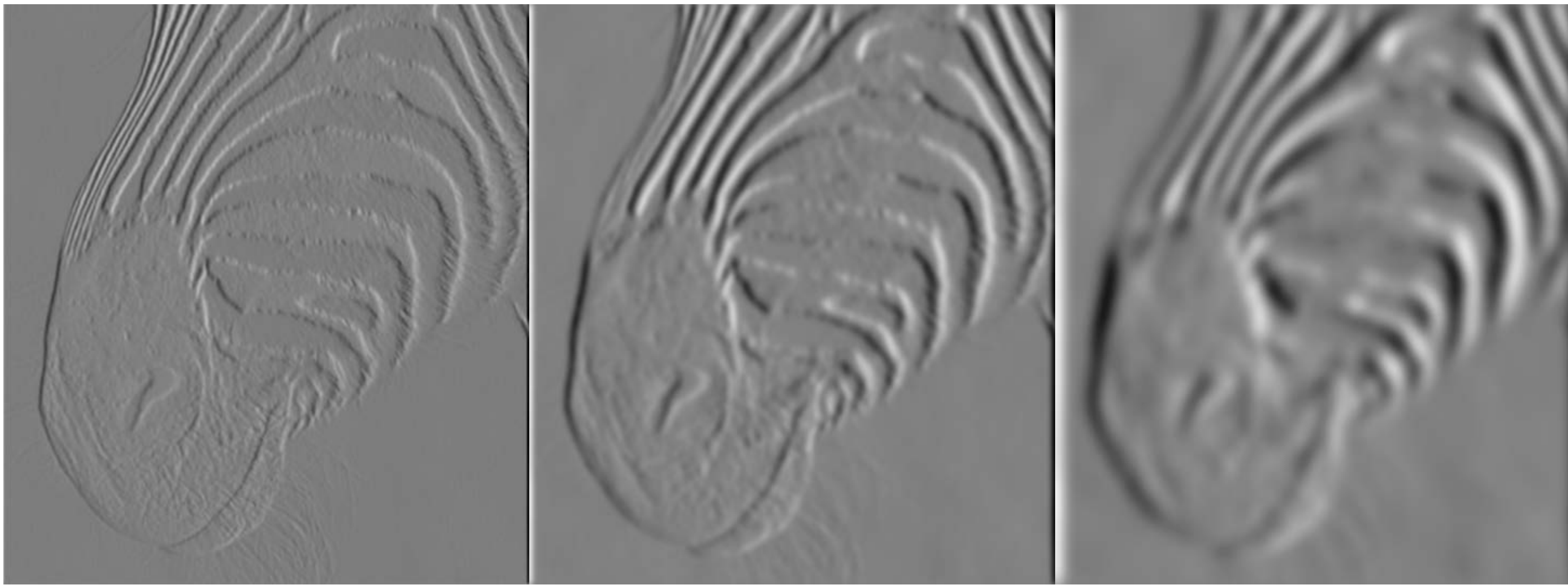
Example: Sobel Operator


Source: Wikipedia

Also Consider: Scale of Gaussian’ Filter

- Scale of Gaussian Derivative Filter



1 pixel | 3 pixels | 7 pixels

Smoothed derivative removes noise
BUT blurs the edges
AND finds edges at different ‘scales.’

source: D. Forsyth

Implementation Issues

- The gradient magnitude is large along a thick trail or ridge
  - How do we identify actual edge points?
  - How do we link the edge points to form curves?

Example: Lena

Original Image

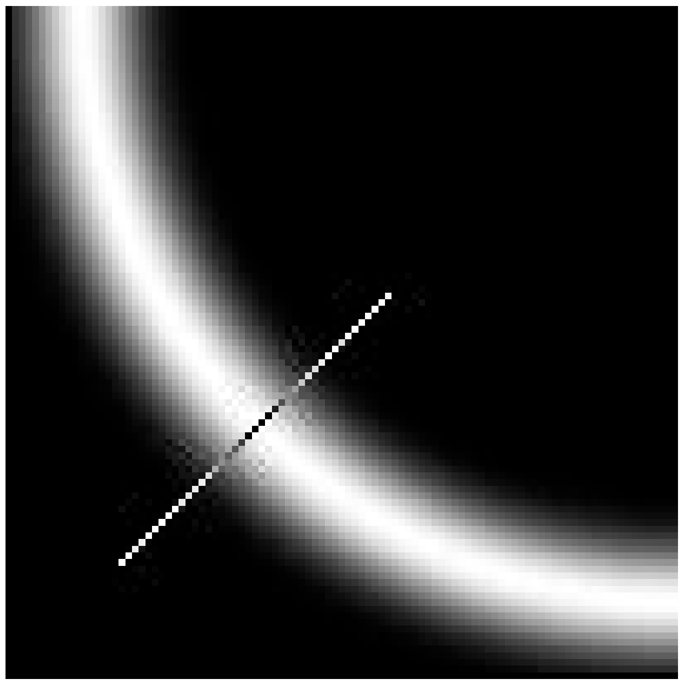
Finding Edges

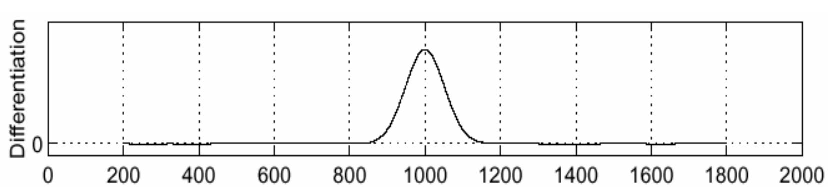
Gradient Magnitude

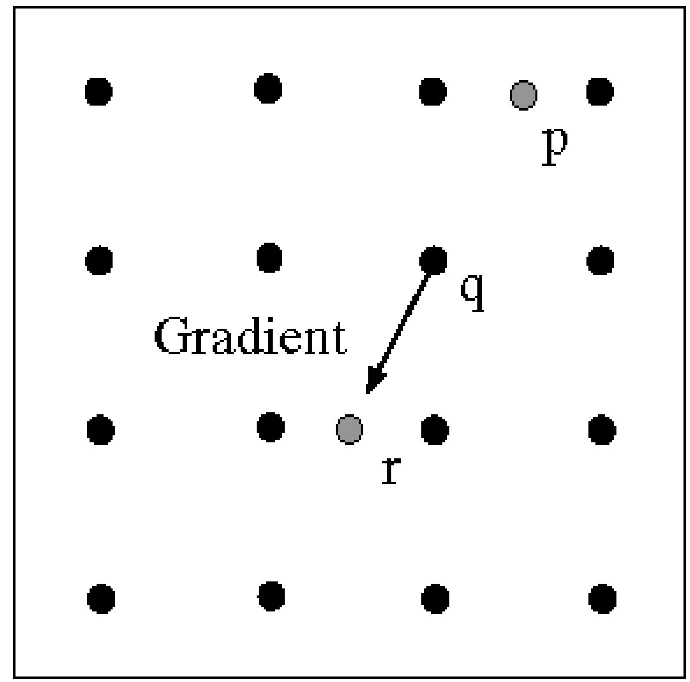
Finding Edges

where is the edge?

Non-Maximum Suppression







- Check if pixel is a local maximum along gradient direction
  - ***q*** is a maximum if the value is larger than those at both ***p*** and ***r***
    - *note: p and r values would be interpolated values*

source: D. Forsyth

Finding Edges

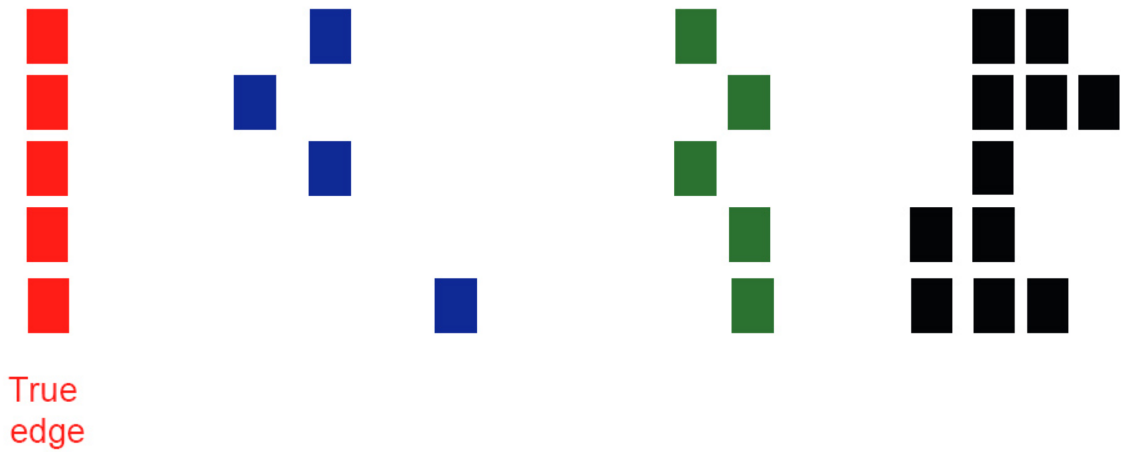
Thresholding

Finding Edges

Thinning
(non-maximum suppression)

Designing an Edge Detector

- Criteria for an 'optimal' edge detector
  - **Good Detection**
    - Minimizes the probability of false positives and false negatives
  - **Good Localization**
    - The detected edges must be as close as possible to the true edges
  - **Single Response**
    - Returns only one point for each true edge point
      - Minimize the number of local maxima around the true edge



source: L. Fei-Fei

Canny Edge Detector

- Originated in 1986
  - J. Canny, A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

- Still widely used today

- Depends on several parameters
  - σ : width (size of kernel) of the Gaussian blur
  - high and low threshold

Canny Edge Detector

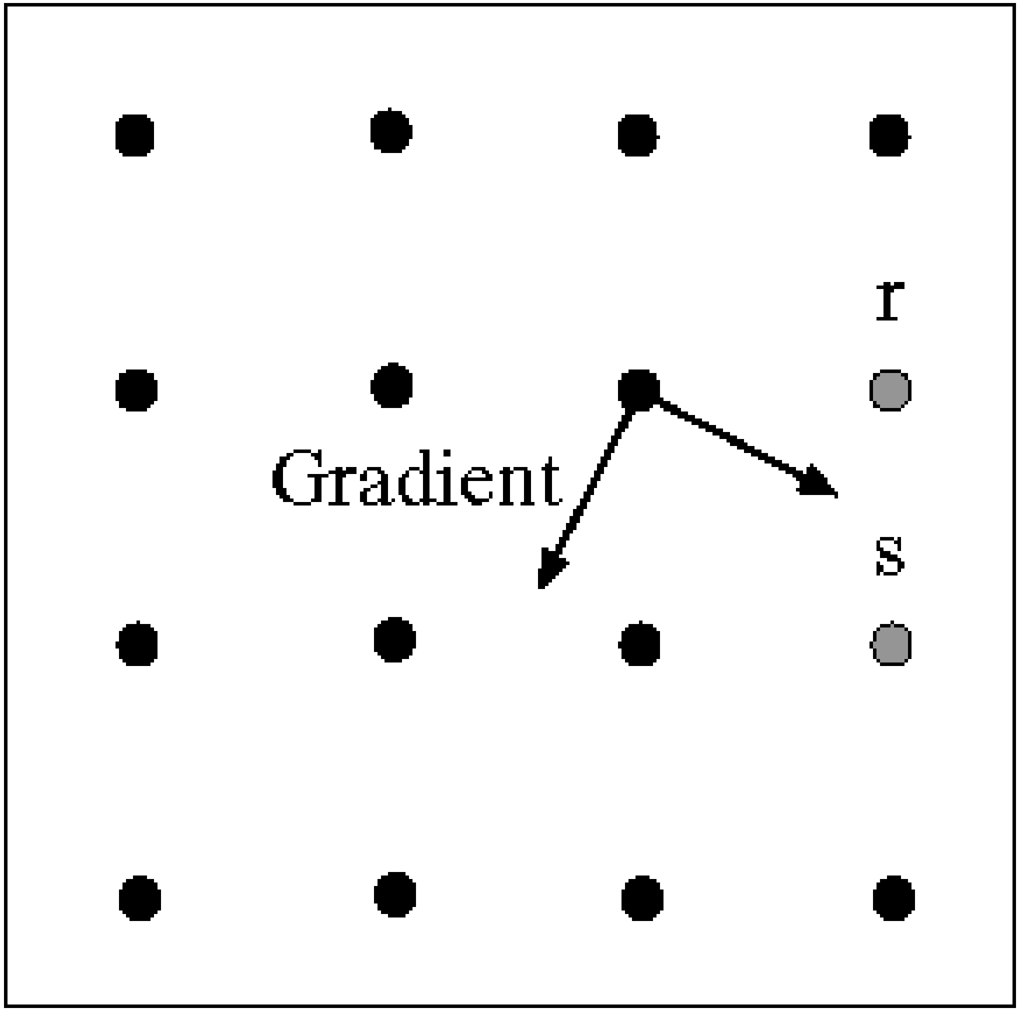
- Filter image with derivative of Gaussian

- Find magnitude and orientation of gradient

- Apply non-maximum suppression
- Linking and thresholding (hysteresis)
  - Define 2 thresholds: low and high
  - Use the high threshold to start edge curves and the low threshold to continue them

source: L. Fei-Fei, D. Lowe

Edge Linking

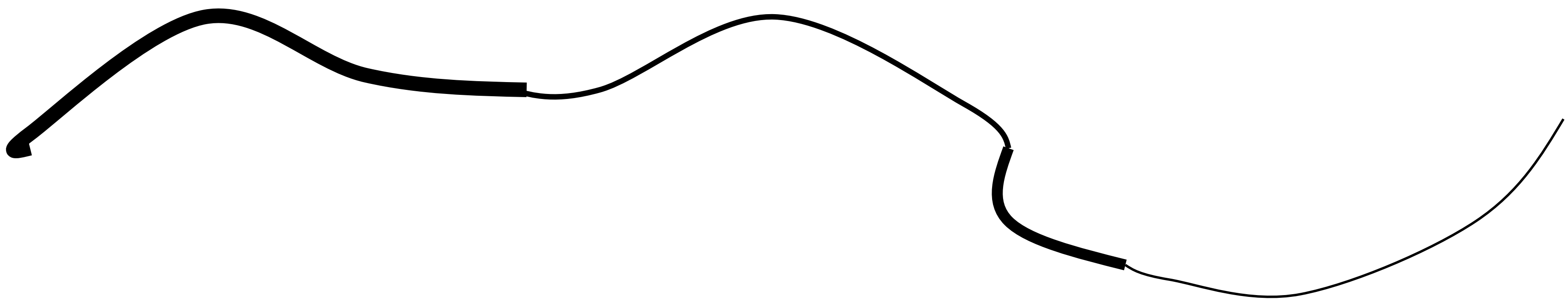


- Assume the marked point is an edge point
- Construct the tangent to the edge curve
  - which is normal to the gradient at that point
- Use this to predict the next points
  - either ***r*** or ***s*** as shown

source: D. Forsyth

Hysteresis Thresholding

- Use a high threshold to start edge curves and a low threshold to continue them



source: S. Seitz

Example: Hysteresis Thresholding

original image

high threshold
(strong edges)

low threshold
(weak edges)

hysteresis threshold

source: L. Fei-Fei

Effect of σ (size of kernel)

original

Canny with $\sigma = 1$

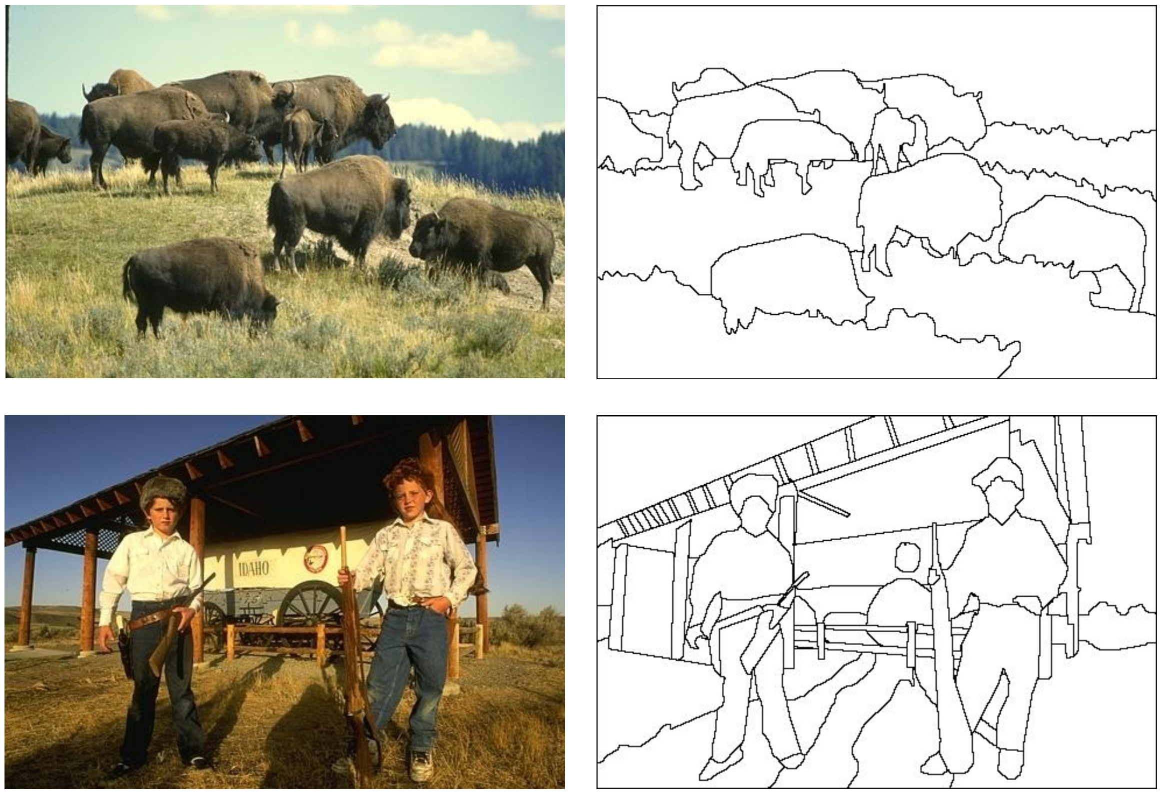
Canny with $\sigma = 2$

- Large σ detects large scale edges
- Small σ detects small (finer) scale edges

source: S. Seitz

Edge Detection is **Just the Beginning**

image segmentation



Questions?

- Beyond D2L
  - Examples and information can be found online at:
    - *http://docdingle.com/teaching/cs.html*

    - *Continue to more stuff as needed*

Extra Reference Stuff Follows

Credits

- Much of the content derived/based on slides for use with the book:
  - *Digital Image Processing,* Gonzalez and Woods

- Some layout and presentation style derived/based on presentations by
  - Donald House, Texas A&M University, 1999
  - Sventlana Lazebnik, UNC, 2010
  - Noah Snavely, Cornell University, 2012
  - Xin Li, WVU, 2014
  - George Wolberg, City College of New York, 2015
  - Yao Wang and Zhu Liu, NYU-Poly, 2015