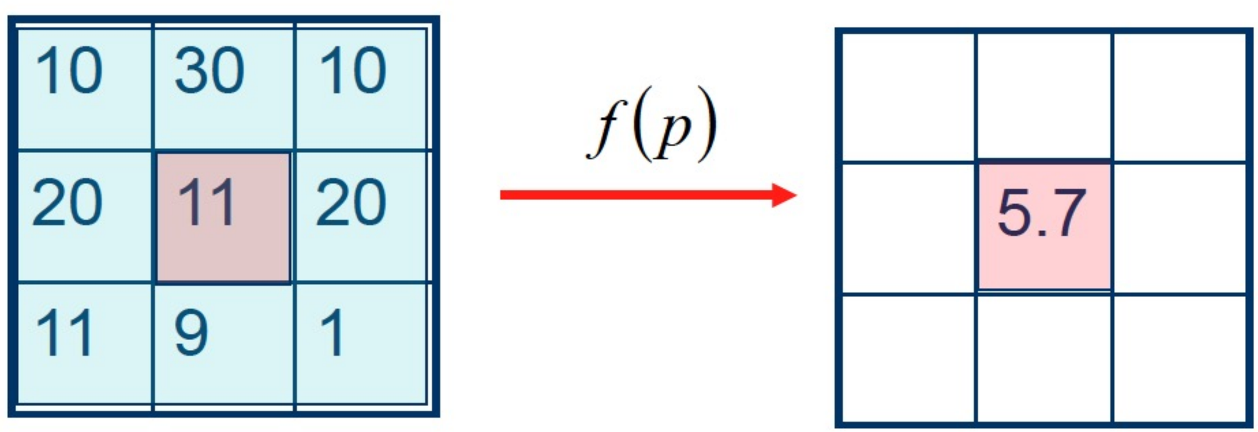
Medical Images- Pre-Processing

Lecture 5

Filtering

- Modify pixels based on some function of the neighborhood



Linear Filtering

- The output is the linear combination of the neighborhood pixels

1	3	0
2	10	2
4	1	1

Image

$\otimes$

1	0	-1
1	0.1	-1
1	0	-1

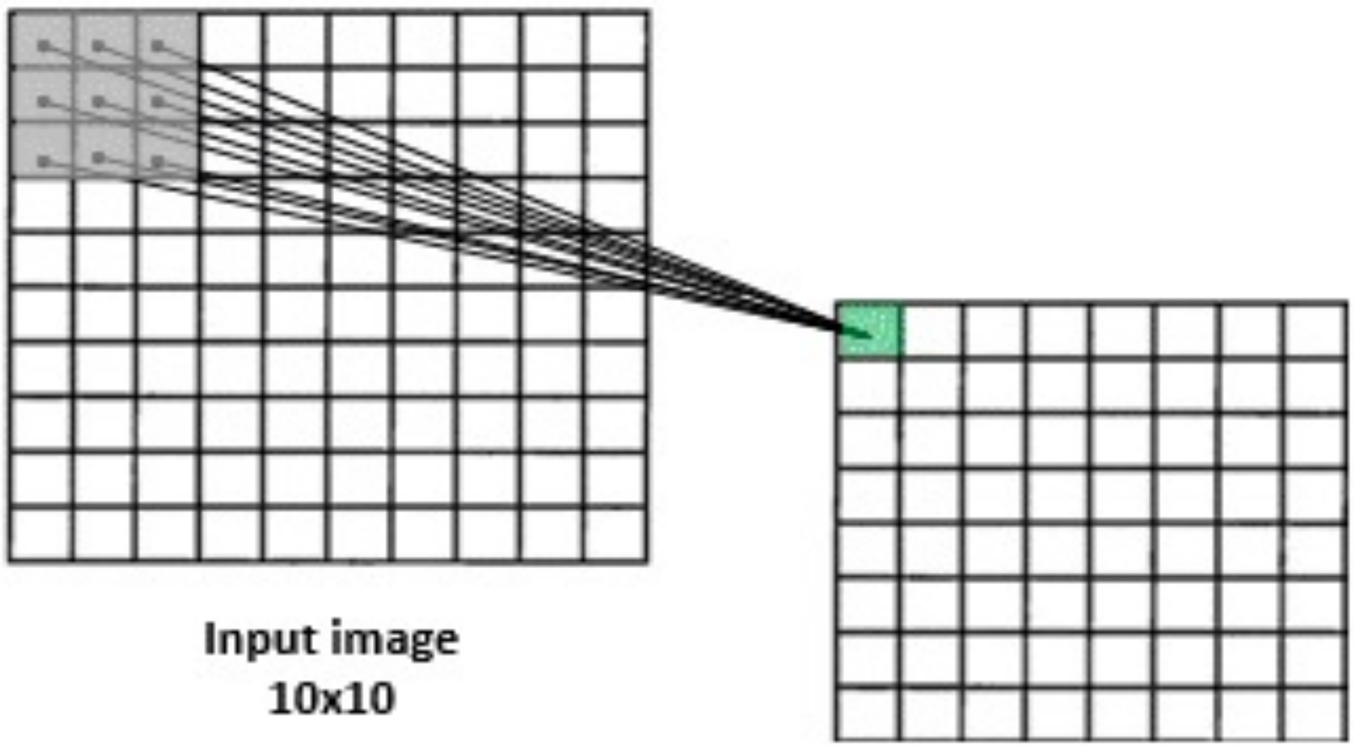
Kernel

=

	5	

Filter Output

Alper Yilmaz, Mubarak Shah, UCF


Input image
10x10

$g[\cdot\,,\cdot\,]$ $\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$

Filtering Operation (Spatial Domain)

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit: S. Seitz

Averages

- Mean

$$I = \frac{I_1 + I_2 + \ldots I_n}{n} = \frac{\sum_{i=1}^{n} I_i}{n}$$

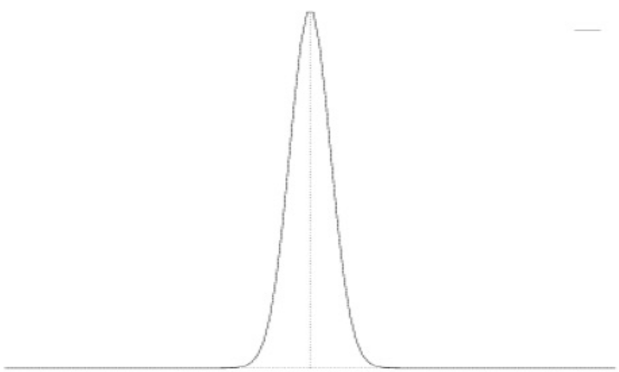
- Weighted mean

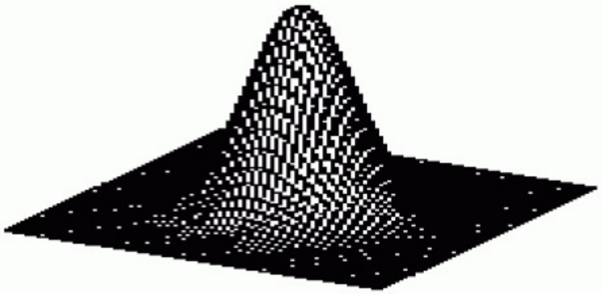
$$I = \frac{w_1 I_1 + w_2 I_2 + \ldots + w_n I_n}{n} = \frac{\sum_{i=1}^{n} w_i I_i}{n}$$

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$

	0.01	0.08	0.01
Gaussian	0.08	0.64	0.08
	0.01	0.08	0.01

Gaussian Filter





$$g(x) = e^{\frac{-x^2}{2o^2}}$$

$$g(x,y) = e^{\frac{-(x^2+y^2)}{2o^2}}$$

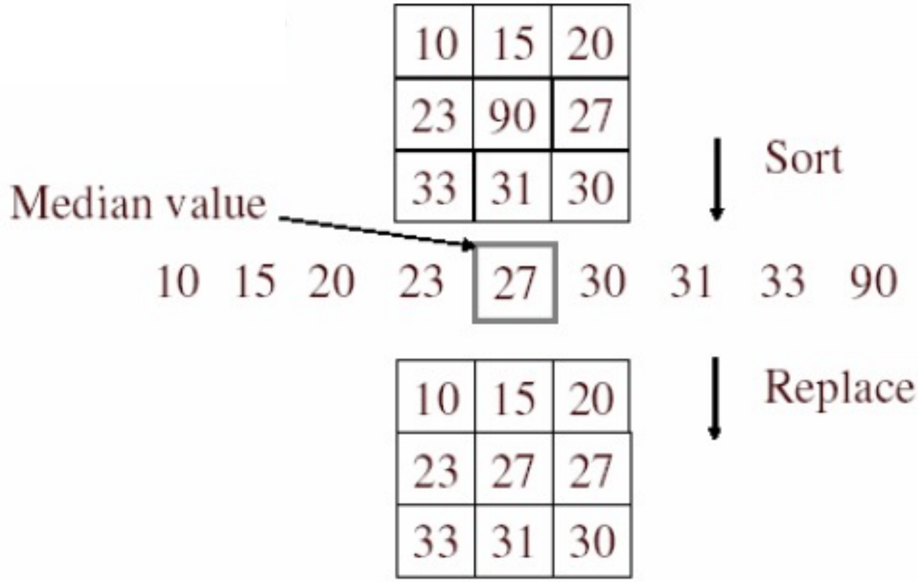
$$g(x) = [.011 \quad .13 \quad .6 \quad 1 \quad .6 \quad .13 \quad .011]$$

o =1

Carl Friedrich Gauss
1777 to 1855

Alternative idea: Median filtering

- A **median filter** operates over a window by selecting the median intensity in the window



- Is median filtering linear?

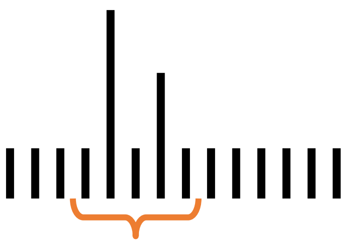
Source: K. Grauman

Median filter

- Replace each pixel by the median over N pixels (5 pixels, for these examples). Generalizes to "rank order" filters.

Median([1 7 1 5 1]) = 1
Mean([1 7 1 5 1]) = 2.8

In:

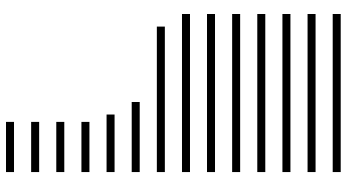


5-pixel neighborhood

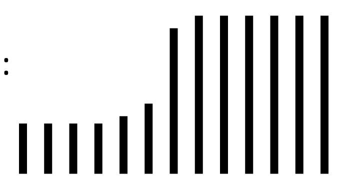
Out:

Spike noise is removed

In:



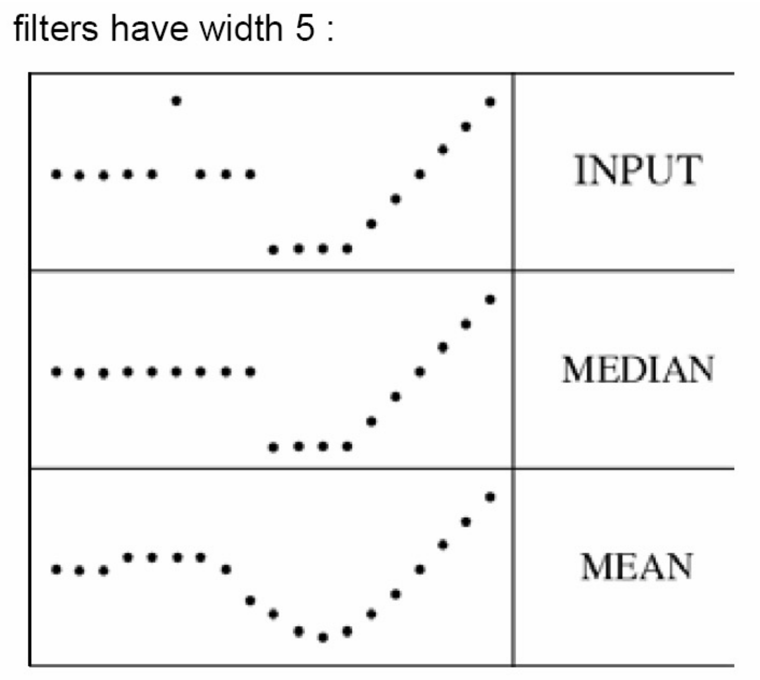
Out:



Monotonic edges remain unchanged

Median filter

- What advantage does median filtering have over Gaussian filtering?
  - Robustness to outliers



Source: K. Grauman

Linear and Non-Linear filters

- Linear means something related to a line. All the linear equations are used to construct a line.
- A non-linear equation is such which does not form a straight line. It looks like a curve in a graph and has a variable slope value.

The general representation of linear equation is; $$y = mx + c$$ Where x and y are the variables, m is the slope of the line and c is a constant value.	The general representation of nonlinear equations is; $$ax^2 + by^2 = c$$ Where x and y are the variables and a,b and c are the constant values
Examples: • $10x = 1$ • $9y + x + 2 = 0$ • $4y = 3x$ • $99x + 12 = 23\,y$	**Examples:** • $x^2+y^2 = 1$ • $x^2 + 12xy + y^2 = 0$ • $x^2+x+2 = 25$

https://byjus.com/maths/difference-between-linear-and-nonlinear-equations/

Convolution

f = Image

h = Kernel

h

h_1	h_2	h_3
h_4	h_5	h_6
h_7	h_8	h_9

$X-flip$

h_7	h_8	h_9
h_4	h_5	h_6
h_1	h_2	h_3

$Y-flip$

f

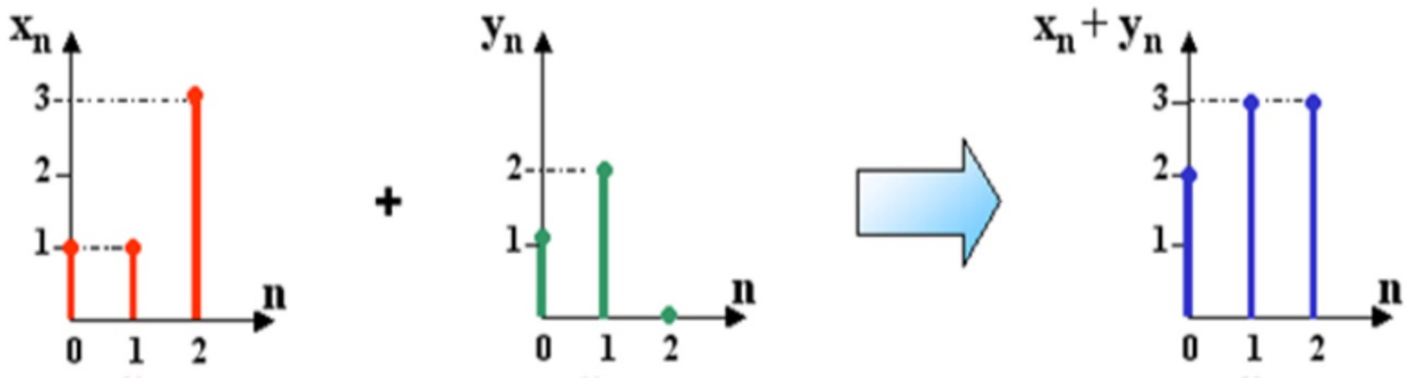
f_1	f_2	f_3
f_4	f_5	f_6
f_7	f_8	f_9

*

h_9	h_8	h_7
h_6	h_5	h_4
h_3	h_2	h_1

$$f * h = f_1 h_9 + f_2 h_8 + f_3 h_7 + f_4 h_6 + f_5 h_5 + f_6 h_4 + f_7 h_3 + f_8 h_2 + f_9 h_1$$

Linear and Non-Linear filters



Linear Filter $F_m(A + \lambda B) = F_m(A) + \lambda F_m(B)$,

$$mean(Xn) + mean(Yn) = mean(Xn + Yn),$$
$$5/3 + 1 = 8/3$$

$$median(Xn) + median(Yn) \neq median(Xn + Yn).$$
$$1 + 1 \neq 3$$

https://dsp.stackexchange.com/questions/14241/what-is-the-difference-between-linear-and-non-linear-filters

Practice with linear filters

Original

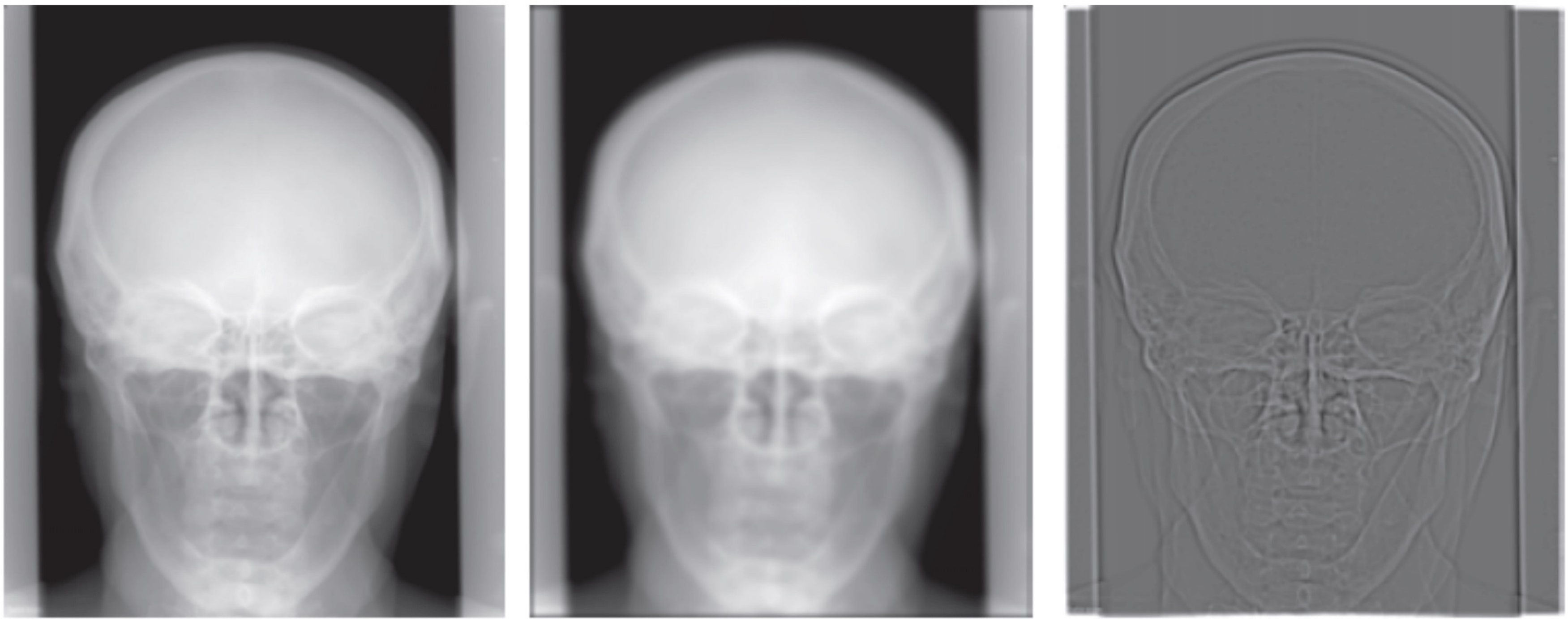
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening filter

- Accentuates differences with local average

Source: D. Lowe

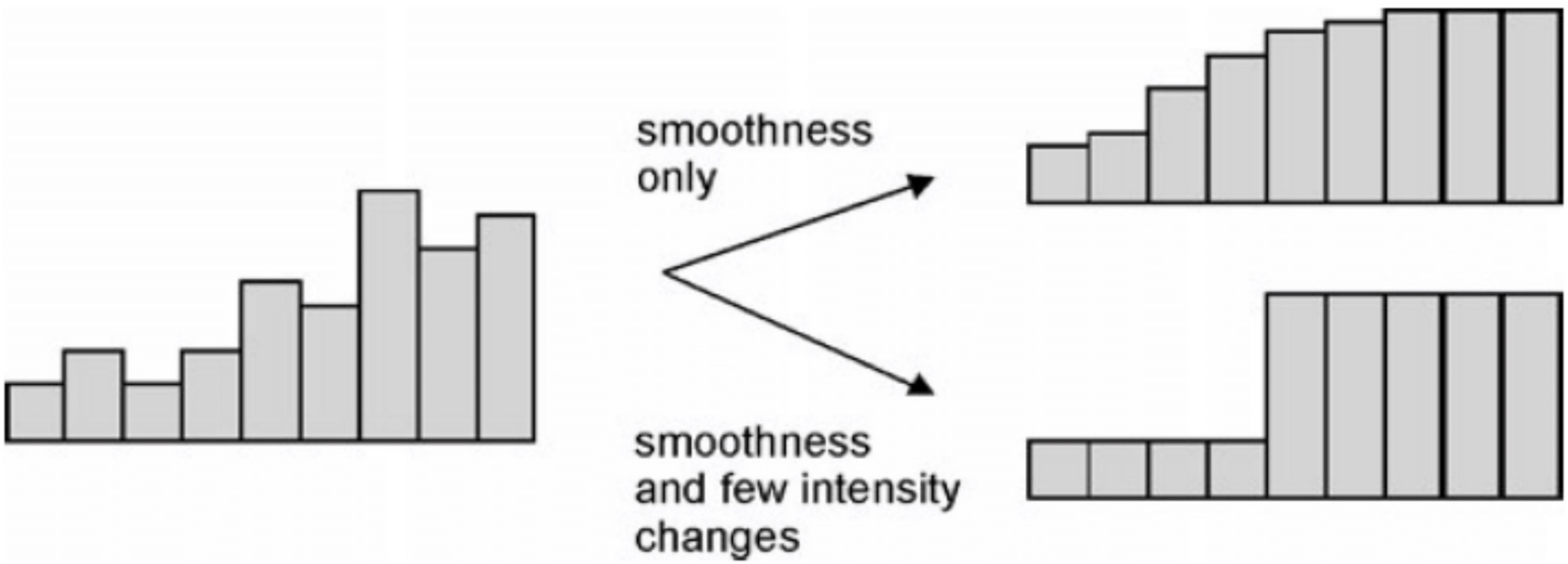
Filtering X-ray



(a) (b) (c)

a. Radiography of the skull, b. low-pass filter with a Gaussian filter (std=15, 20 x 20), c. high-pass Filter obtained from subtracting b from a.

Unsharp Masking
Lower Noise, Higher Contrast



histogram

histogram

Unsharp Masking

- Not only noise removal, but edge enhancement is necessary!

$$I = g * I + (I - g * I)$$

Unsharp Masking

- Not only noise removal, but edge enhancement is necessary!

$$I = \underbrace{g * I}_{\substack{\text{Smoothed image} \\ \text{(low pass)}}} + \underbrace{(I - g * I)}_{\substack{\text{Edge enhanced image} \\ \text{(high pass)}}}$$

Unsharp Masking

- Not only noise removal, but edge enhancement is necessary!

$$I = \underbrace{g * I}_{\text{Smoothed image (low pass)}} + \underbrace{(I - g * I)}_{\text{Edge enhanced image (high pass)}}$$

$$I' = g * I + (1 + \alpha)(I - g * I) \qquad \alpha > 0$$

Reminder: Edges are located in high frequency of the images!

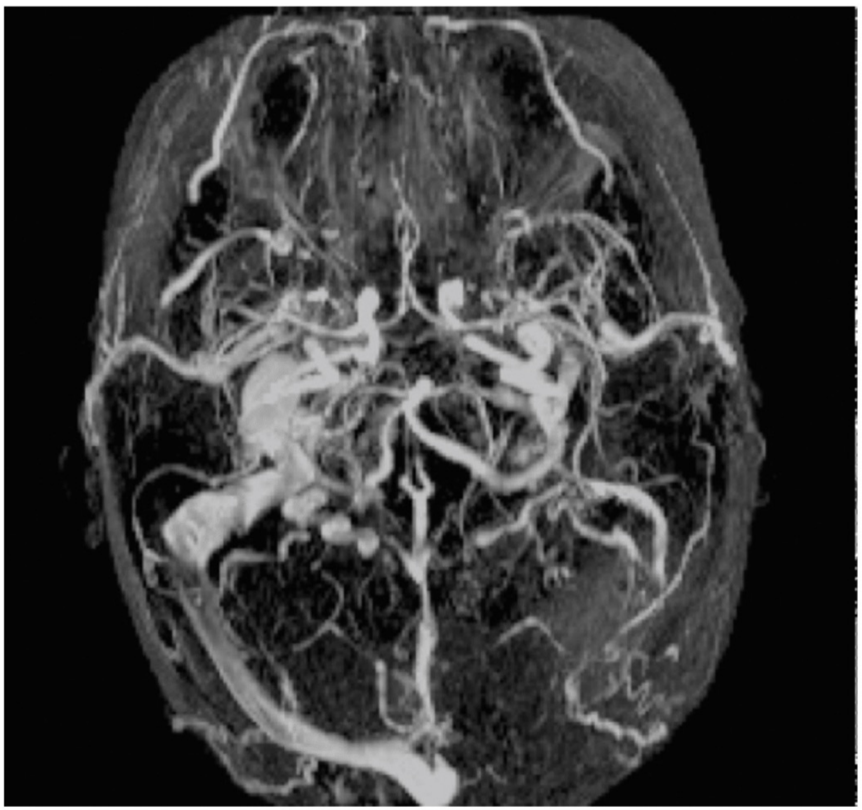
Hand X-ray Unsharp Masking (alpha=0.5)

Original Image

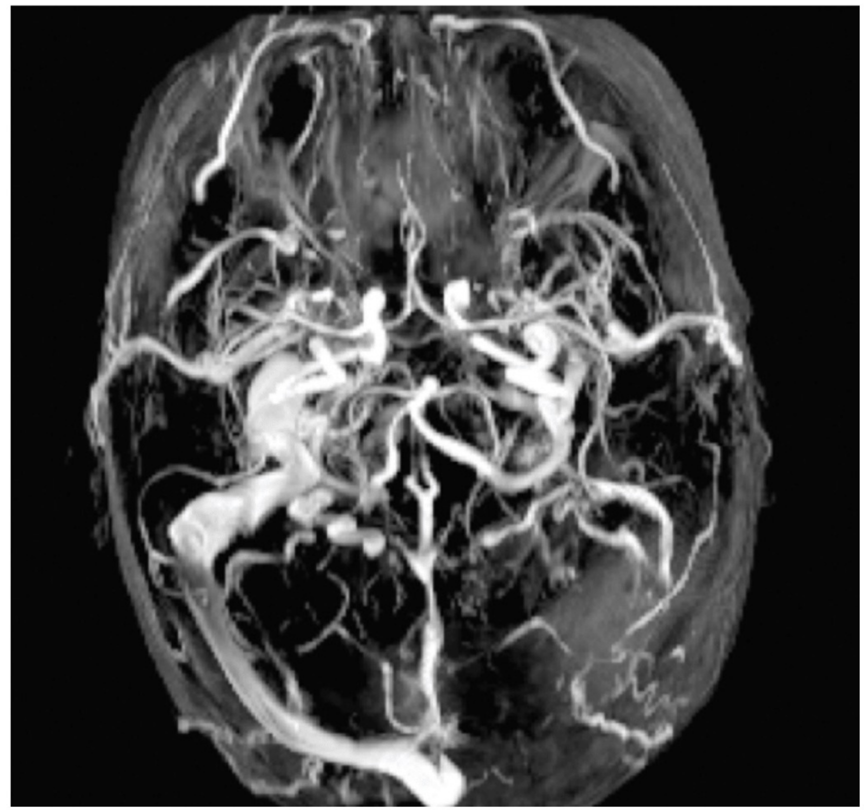
Enhanced Image

Unsharp Masking: Example CT (head, axial)

Adaptive Filtering: Example head MRA

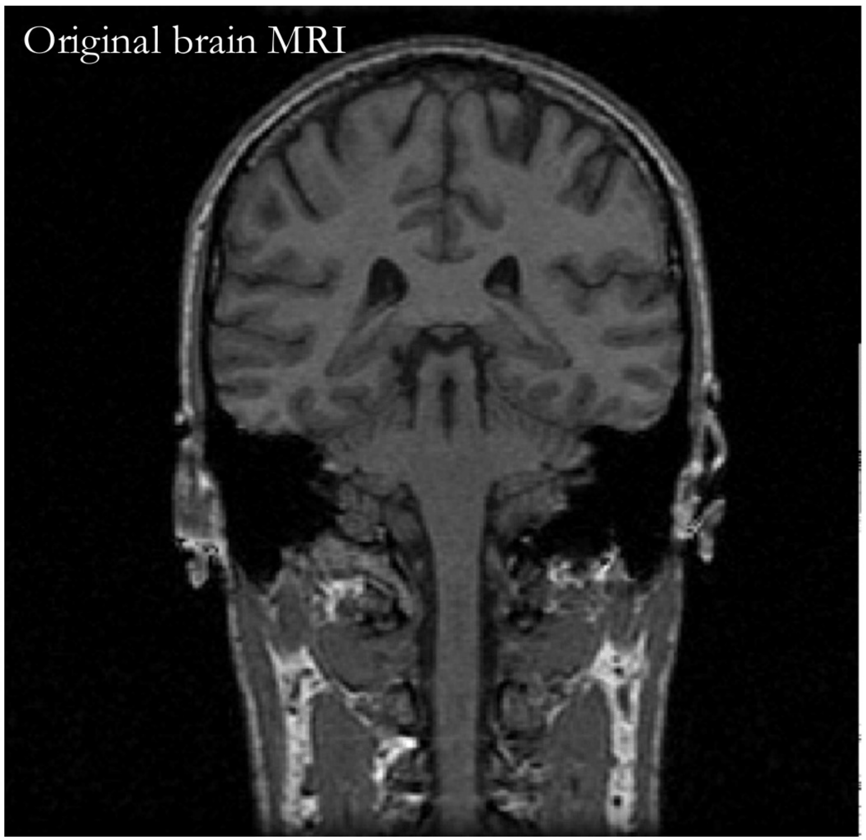


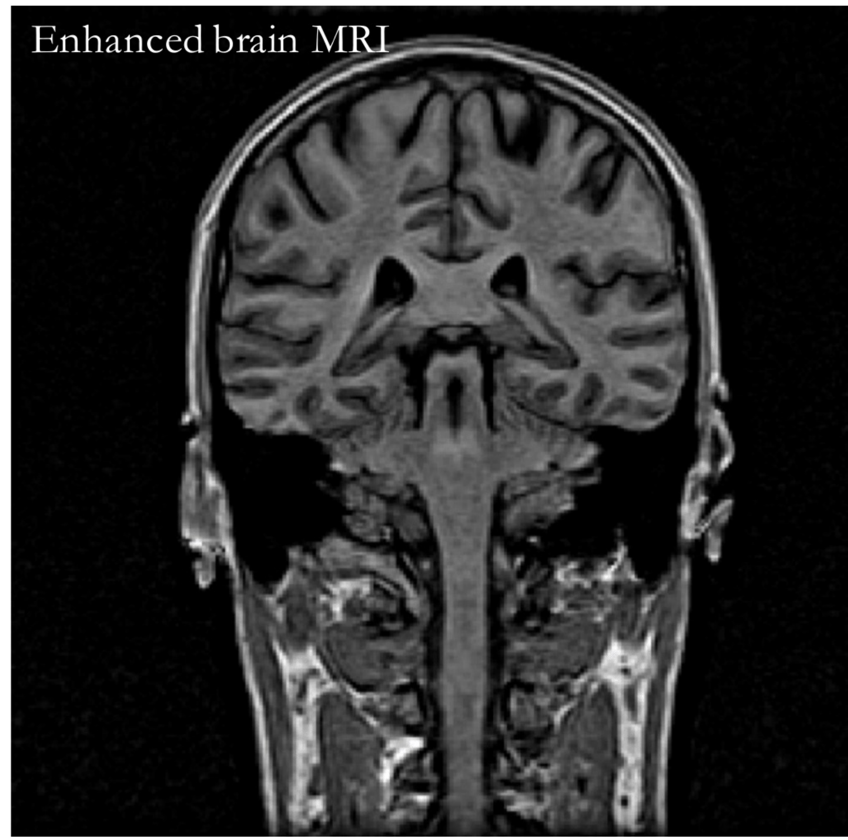
MIP of MRA data before filtering



MIP of MRA data after filtering

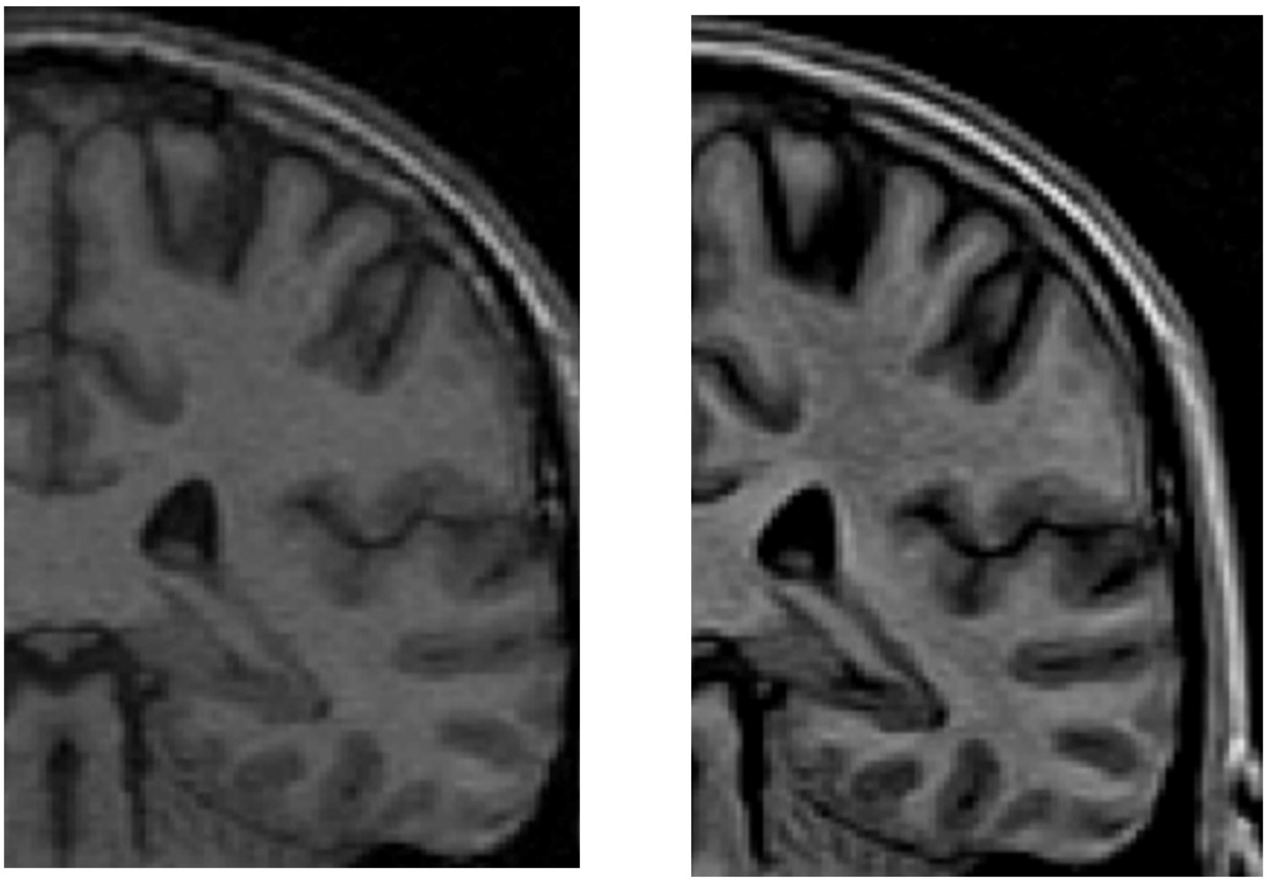
Adaptive Filtering: Example brain MRI





Note the improved contrast between brain and CSF (cerebrospinal fluid)

Adaptive Filtering: Example brain MRI (zoomed)



Note the improved contrast between brain and CSF (cerebrospinal fluid)

Practical matters

- What about near the edge?
  - the filter window falls off the edge of the image
  - need to extrapolate
  - methods:
    - clip filter (black)
    - wrap around
    - copy edge
    - reflect across edge

Source: S. Marschner

Separability example

2D convolution
(center location only)

$$\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} * \begin{bmatrix}2 & 3 & 3\\ 3 & 5 & 5\\ 4 & 4 & 6\end{bmatrix}$$

The filter factors
into a product of 1D
filters:

$$\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} = \begin{bmatrix}1\\ 2\\ 1\end{bmatrix} \times \begin{bmatrix}1 & 2 & 1\end{bmatrix}$$

Perform convolution
along rows:

$$\begin{bmatrix}1 & 2 & 1\end{bmatrix} * \begin{bmatrix}2 & 3 & 3\\ 3 & 5 & 5\\ 4 & 4 & 6\end{bmatrix} = \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix}$$

Followed by convolution
along the remaining column:

Source: K. Grauman

Edges

Brain MRI Image

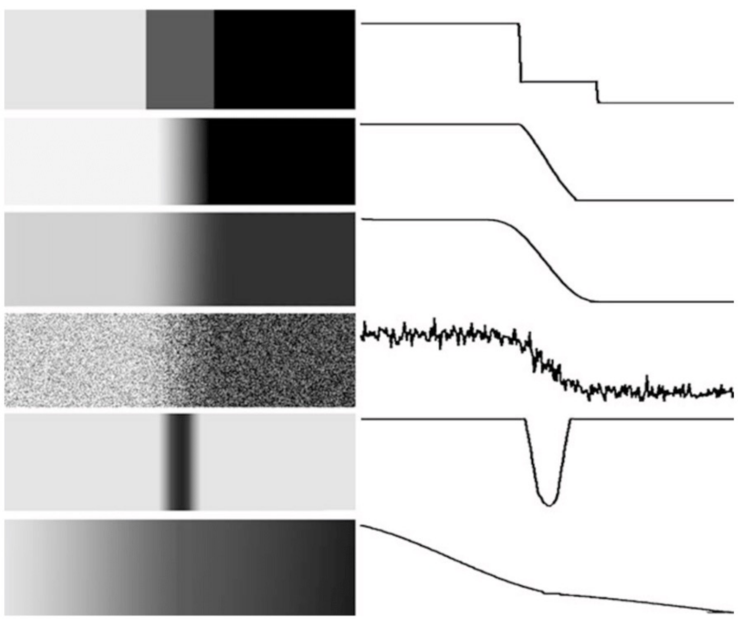
Edge

Edges

- Discontinuities in images are features that are often useful for initializing an image analysis procedure.
- Edges are important information for understanding an image; by moving “non-edge” data we also simplify the data.

Edges

- Discontinuities in images are features that are often useful for initializing an image analysis procedure.
- Edges are important information for understanding an image; by moving “non-edge” data we also simplify the data.



Edges ➔ rate of change

Rate of change ➔ differentiation

Differentiation ➔ difference in digital domain

Edges

- Discontinuities in images are features that are often useful for initializing an image analysis procedure.
- Edges are important information for understanding an image; by moving “non-edge” data we also simplify the data.

- Goal: Identify sudden changes (discontinuities) in an image
  - Most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels
  - Marks the border of an object

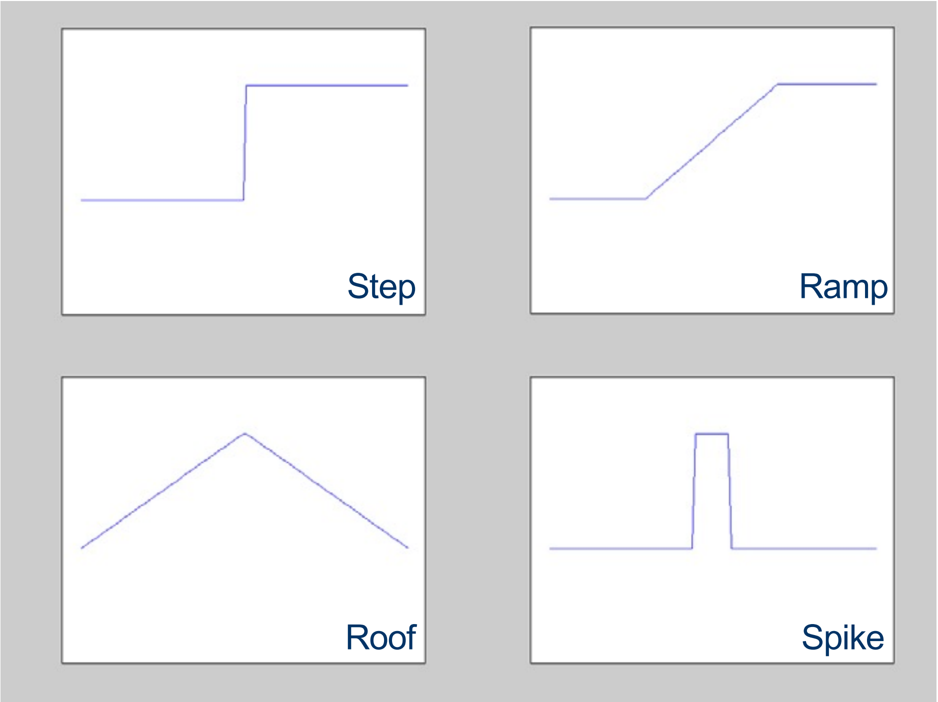
Edges ➔ rate of change

Rate of change ➔ differentiation

Differentiation ➔ difference in digital domain

What is an Edge?

- Discontinuity of intensities in the image
- Edge models
  - Step
  - Roof
  - Ramp
  - Spike



Characterizing Edges

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

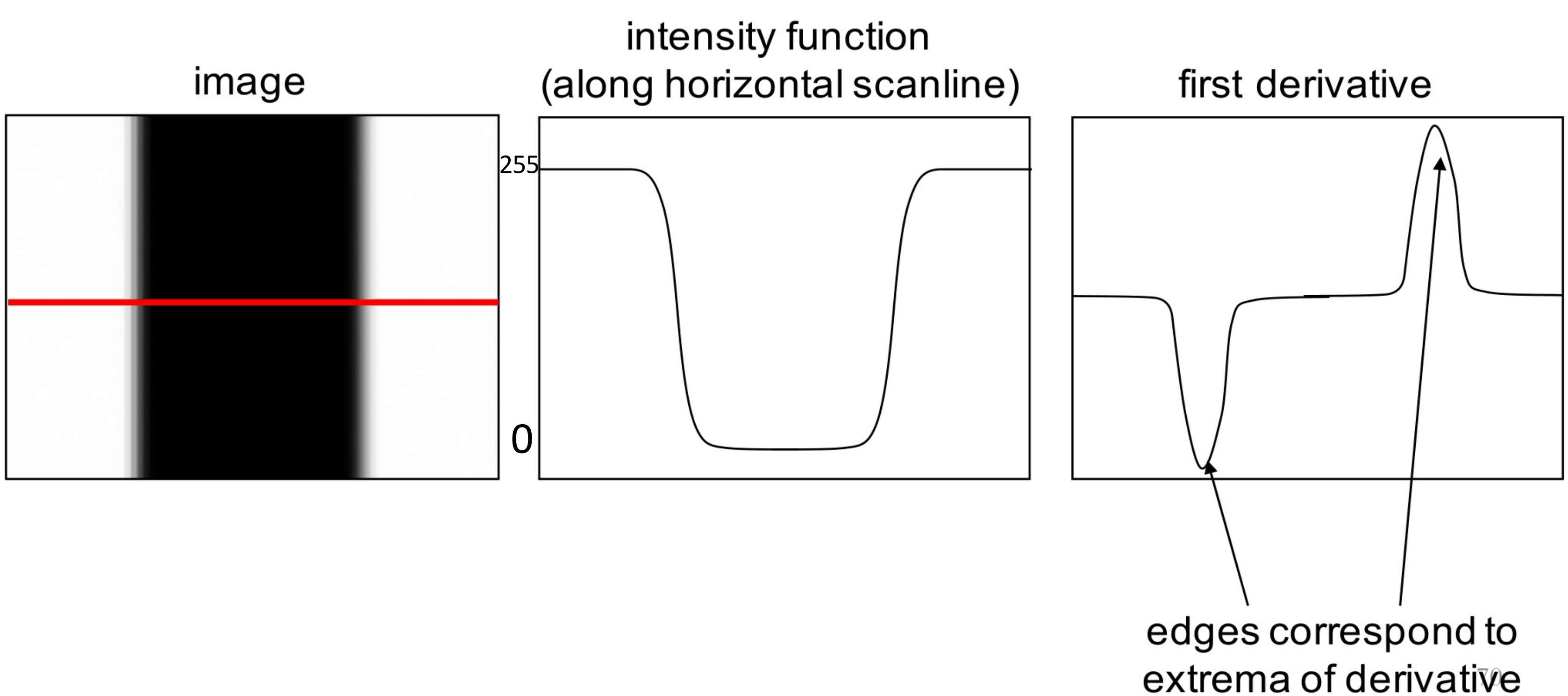


Image Derivatives & Averages

Definitions

Definitions

- Derivative: Rate of change
  - *Speed* is a rate of change of a *distance*
  - *Acceleration* is a rate of change of *speed*

Definitions

- Derivative: Rate of change
  - *Speed* is a rate of change of a *distance*
  - *Acceleration* is a rate of change of *speed*
- Average (Mean)
  - Dividing the sum of N values by N

Derivative

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) = f_x$$

Derivative

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) = \quad f_x$$

$$v = \frac{ds}{dt}$$ speed

$$a = \frac{dv}{dt}$$ acceleration

Examples: Analytic Derivatives

Examples: Analytic Derivatives

$$y = x^2 + x^4$$

$$\frac{dy}{dx} = 2x + 4x^3$$

Examples: Analytic Derivatives

$$y = x^2 + x^4$$

$$\frac{dy}{dx} = 2x + 4x^3$$

$$y = \sin x + e^{-x}$$

$$\frac{dy}{dx} = \cos x + (-1)e^{-x}$$

Discrete Derivative

Discrete Derivative

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)$$

Discrete Derivative

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)$$

$$\frac{df}{dx} = \frac{f(x) - f(x-1)}{1} = f'(x)$$

Discrete Derivative

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x)$$

$$\frac{df}{dx} = \frac{f(x) - f(x-1)}{1} = f'(x)$$

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Discrete Derivative
Finite Difference

Discrete Derivative
Finite Difference

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Discrete Derivative
Finite Difference

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Backward difference

Discrete Derivative
Finite Difference

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Backward difference

$$\frac{df}{dx} = f(x) - f(x+1) = f'(x)$$

Discrete Derivative
Finite Difference

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Backward difference

$$\frac{df}{dx} = f(x) - f(x+1) = f'(x)$$

Forward difference

Discrete Derivative
Finite Difference

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Backward difference

$$\frac{df}{dx} = f(x) - f(x+1) = f'(x)$$

Forward difference

$$\frac{df}{dx} = f(x+1) - f(x-1) = f'(x)$$

Discrete Derivative
Finite Difference

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Backward difference

$$\frac{df}{dx} = f(x) - f(x+1) = f'(x)$$

Forward difference

$$\frac{df}{dx} = f(x+1) - f(x-1) = f'(x)$$

Central difference

Example

Example

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

Example

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

Example

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

Example

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

Example

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

$$f''(x) = 0 \quad 5 \quad -10 \quad 5 \quad 15 \quad -20 \quad 5 \quad 0$$

Example

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

$$f''(x) = 0 \quad 5 \quad -10 \quad 5 \quad 15 \quad -20 \quad 5 \quad 0$$

Derivative Masks

Example

$$f(x) = 10 \quad 15 \quad 10 \quad 10 \quad 25 \quad 20 \quad 20 \quad 20$$

$$f'(x) = 0 \quad 5 \quad -5 \quad 0 \quad 15 \quad -5 \quad 0 \quad 0$$

$$f''(x) = 0 \quad 5 \quad -10 \quad 5 \quad 15 \quad -20 \quad 5 \quad 0$$

Derivative Masks

Backward difference	[-1 1]
Forward difference	[1 -1]
Central difference	[-1 0 1]

Derivatives in 2 Dimensions

Derivatives in 2 Dimensions

Given function $f(x, y)$

Derivatives in 2 Dimensions

Given function $f(x, y)$

Gradient vector $\nabla f(x, y) = \begin{bmatrix} \dfrac{\partial f(x, y)}{\partial x} \\ \dfrac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$

Derivatives in 2 Dimensions

Given function $$f(x, y)$$

Gradient vector $$\nabla f(x, y) = \begin{bmatrix} \dfrac{\partial f(x, y)}{\partial x} \\ \dfrac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Gradient magnitude $$\left|\nabla f(x, y)\right| = \sqrt{f_x^2 + f_y^2}$$

Derivatives in 2 Dimensions

Given function $$f(x, y)$$

Gradient vector $$\nabla f(x, y) = \begin{bmatrix} \dfrac{\partial f(x, y)}{\partial x} \\ \dfrac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Gradient magnitude $$\left|\nabla f(x, y)\right| = \sqrt{f_x^2 + f_y^2}$$

Gradient direction $$\theta = \tan^{-1} \frac{f_x}{f_y}$$

Derivatives of Images

Derivative masks

Derivatives of Images

Derivative masks

$$f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Derivatives of Images

Derivative masks

$$f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad f_y \Rightarrow \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Derivatives of Images

Derivative masks

$$f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad f_y \Rightarrow \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

Derivatives of Images

Derivative masks

$$f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad f_y \Rightarrow \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

Derivatives of Images

Derivative masks

$$f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad f_y \Rightarrow \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Derivatives of Images

Derivative masks

$$f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad f_y \Rightarrow \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \boxed{10} & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Derivatives of Images

Derivative masks

$$f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad f_y \Rightarrow \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \boxed{10} & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Derivatives of Images

Derivative masks

$$f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad f_y \Rightarrow \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Derivatives of Images

Derivatives of Images

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

Derivatives of Images

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad I_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

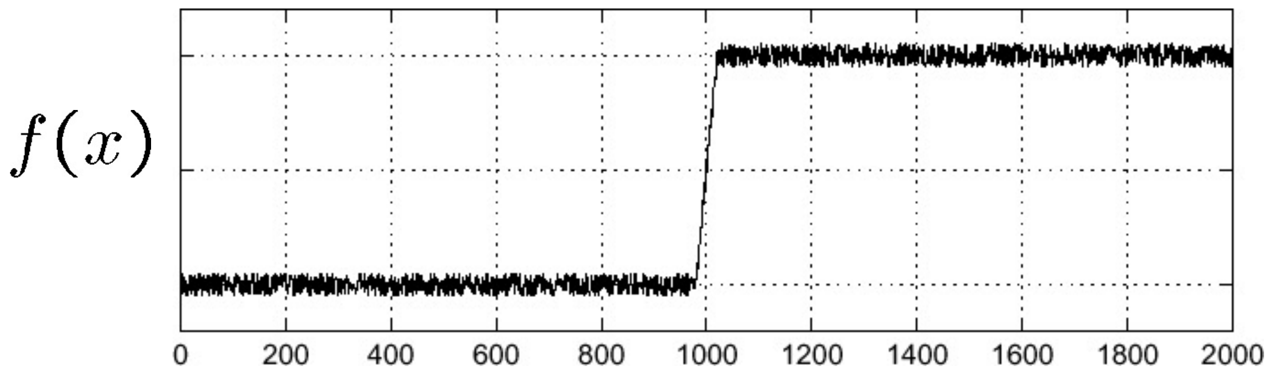
Derivative of Images

Derivative masks $$f_x \Rightarrow \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad f_y \Rightarrow \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$I = \begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix} \qquad I_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \boxed{10} & \boxed{10} & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Effects of noise

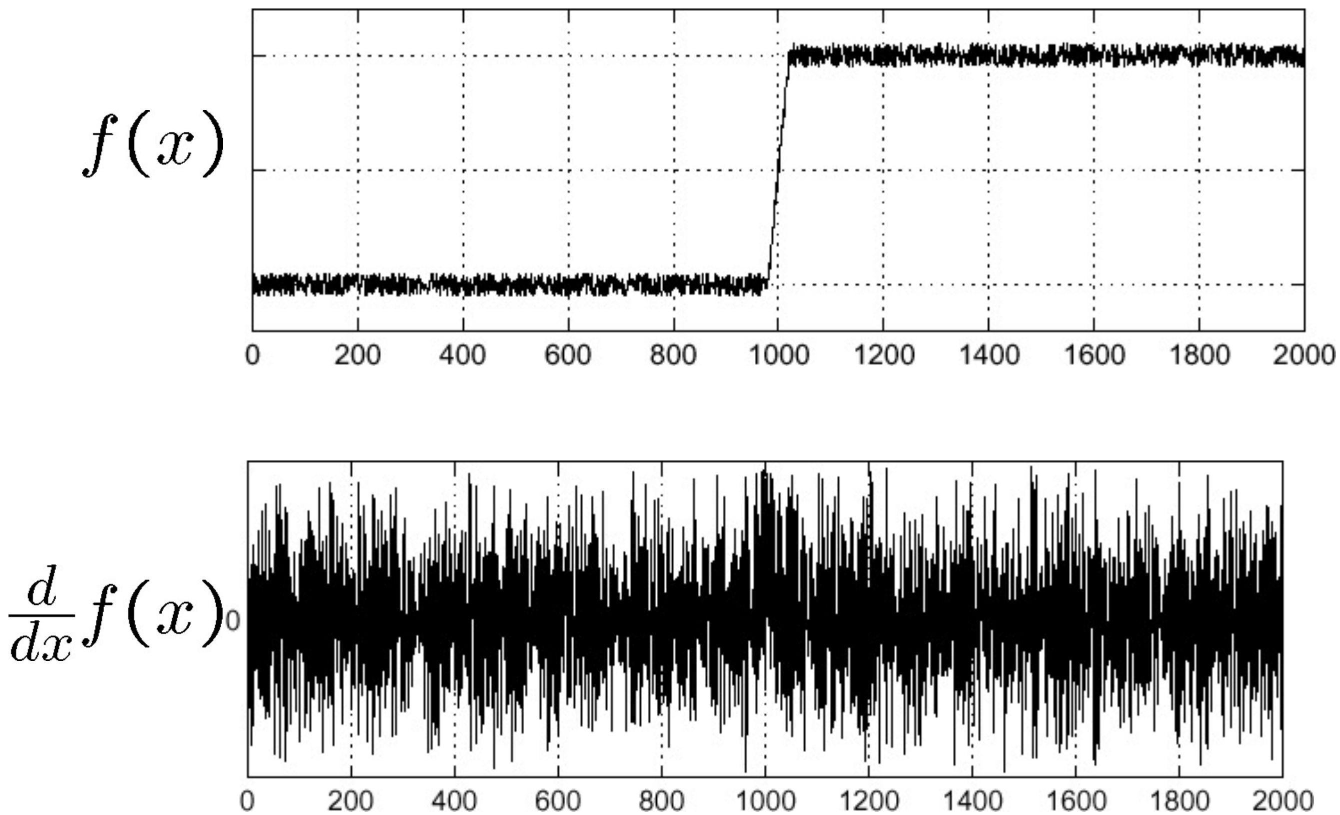
- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal



Source: S. Seitz

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

Source: D. Forsyth

Effects of noise

- Difference filters respond strongly to noise

Source: D. Forsyth

Effects of noise

- Difference filters respond strongly to noise
  - Image noise results in pixels that look very different from their neighbors

Source: D. Forsyth

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

Source: D. Forsyth

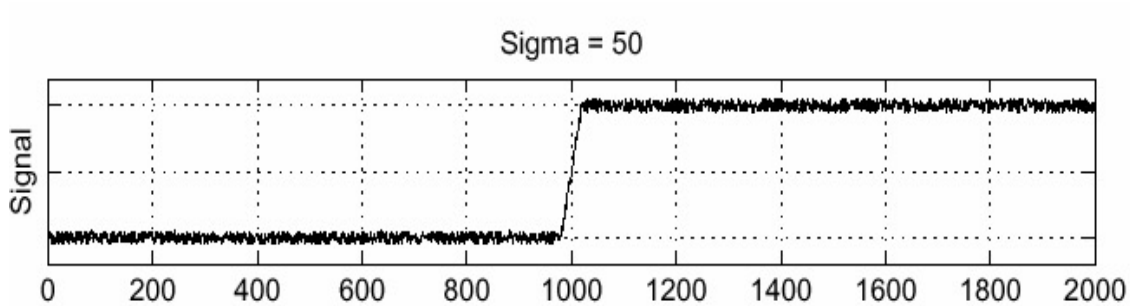
Effects of noise

- 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

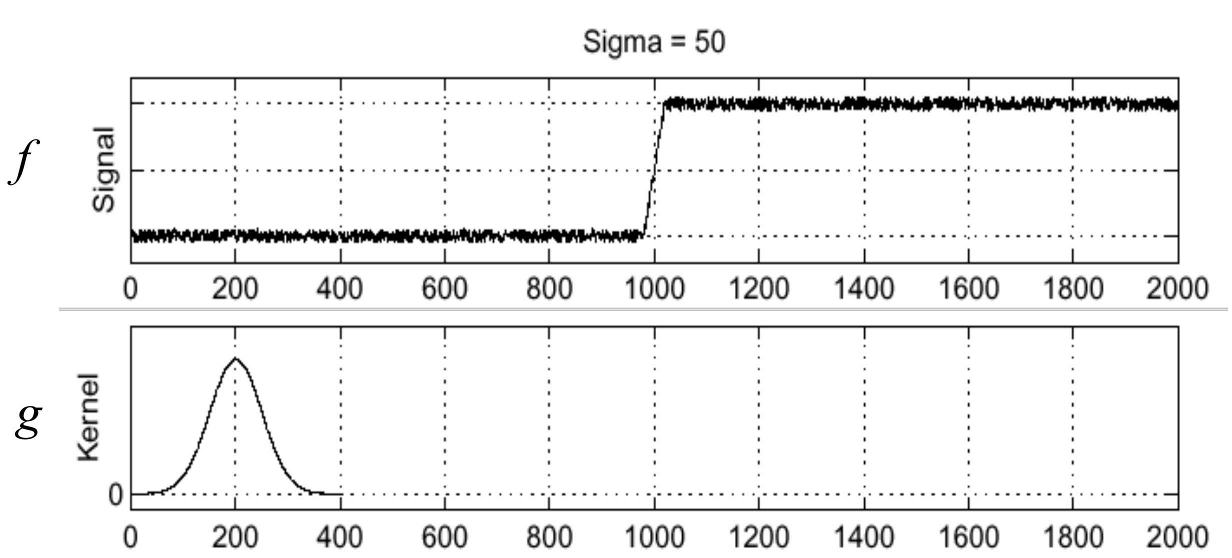
Solution: smooth first

f



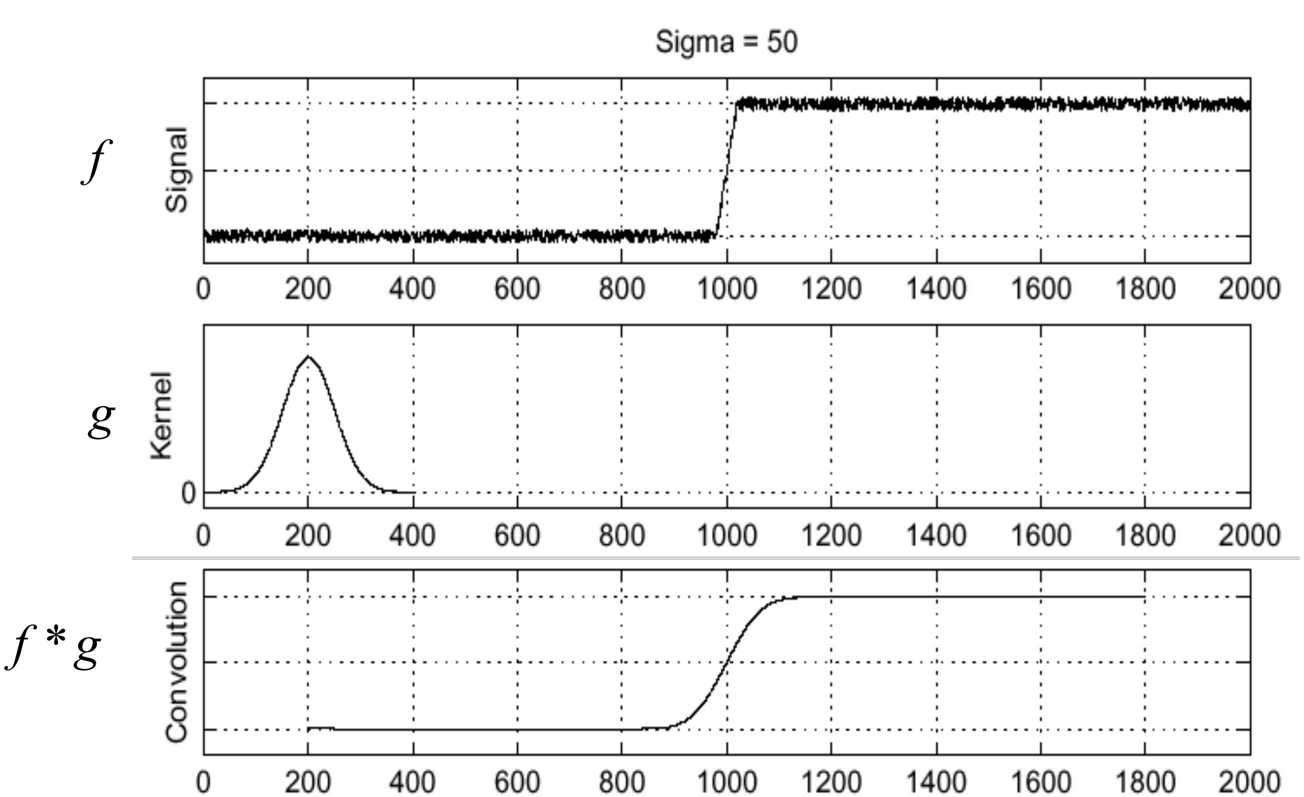
Source: S. Seitz

Solution: smooth first



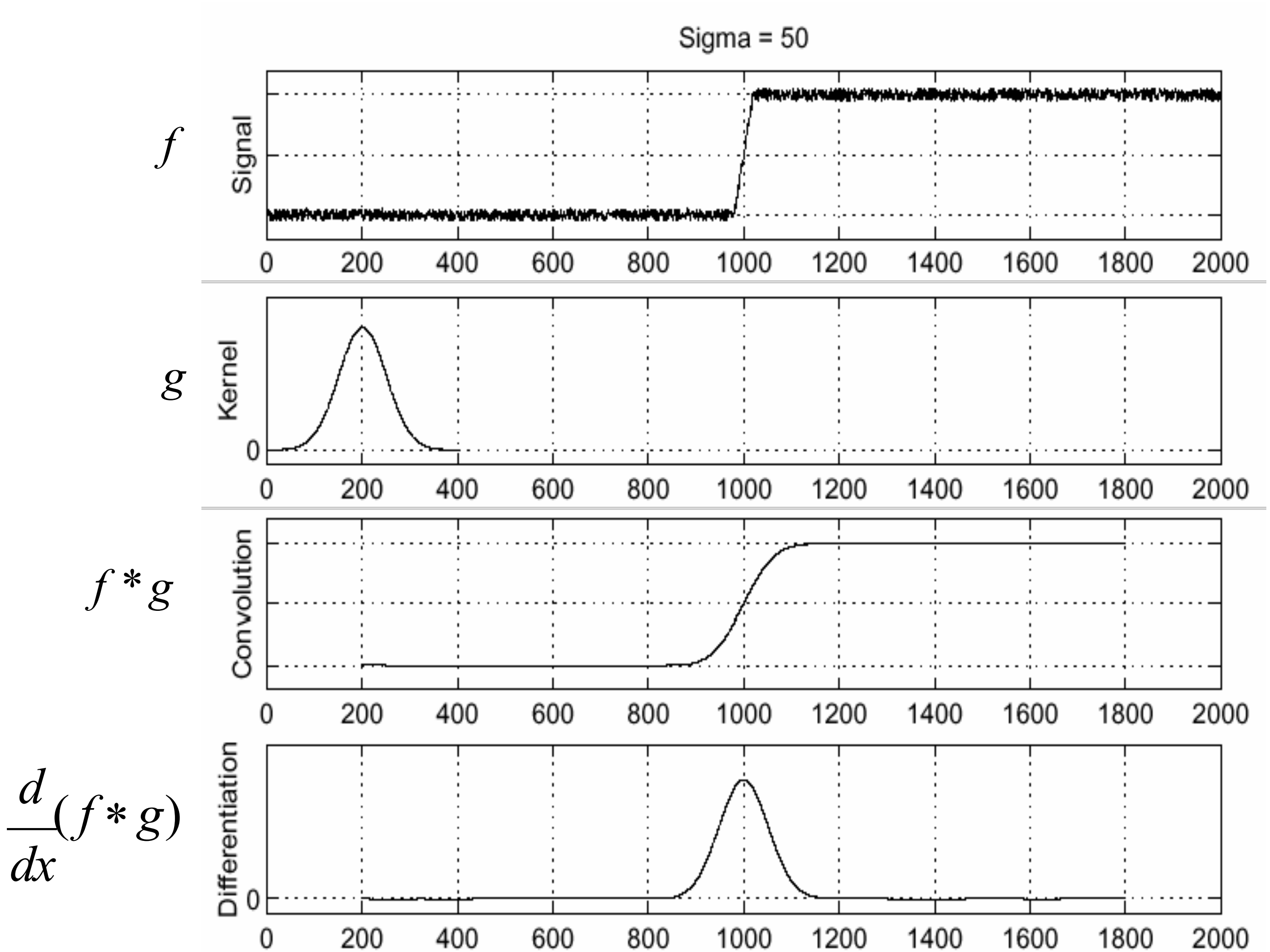
Source: S. Seitz

Solution: smooth first



Source: S. Seitz

Solution: smooth first



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

Source: S. Seitz

Derivative theorem of convolution

Source: S. Seitz

Derivative theorem of convolution

- Differentiation is convolution, and convolution is associative:

Source: S. Seitz

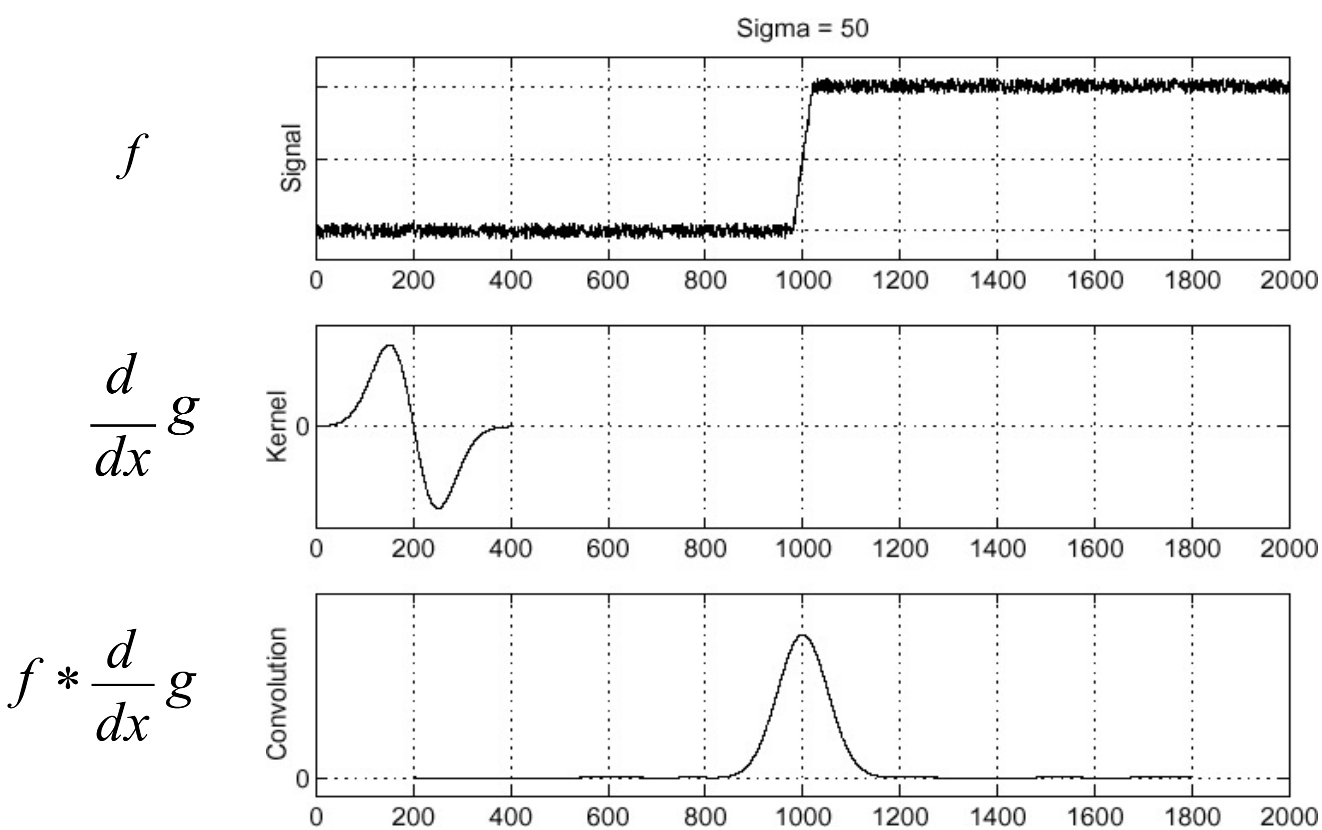
Derivative theorem of convolution

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

Source: S. Seitz

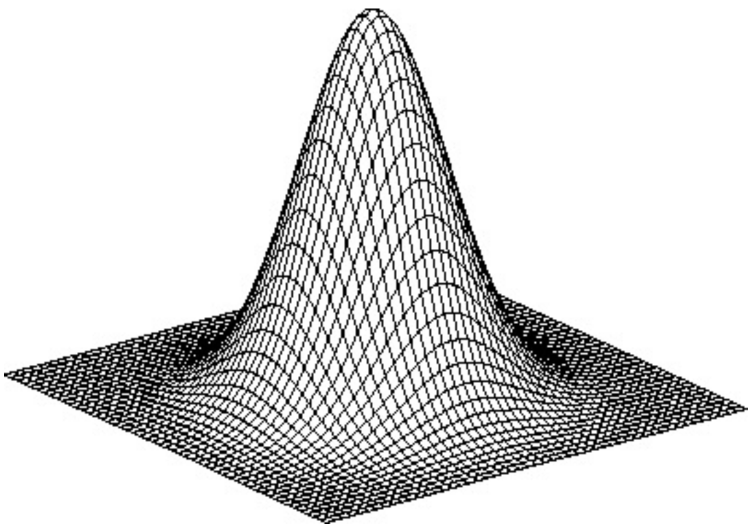
Derivative theorem of convolution

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

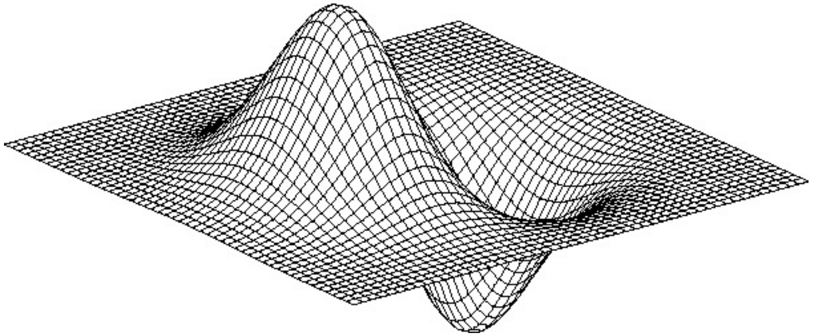


Source: S. Seitz

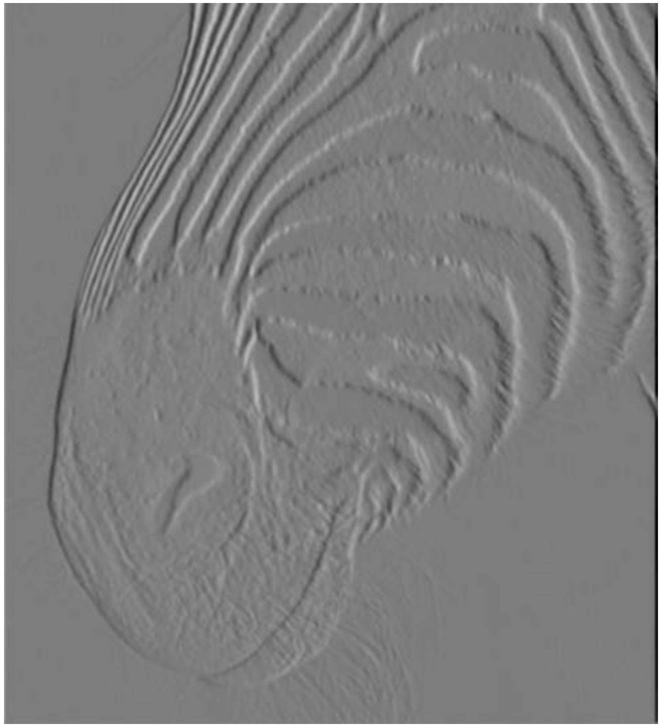
Derivative of Gaussian filter



* [1 -1] =



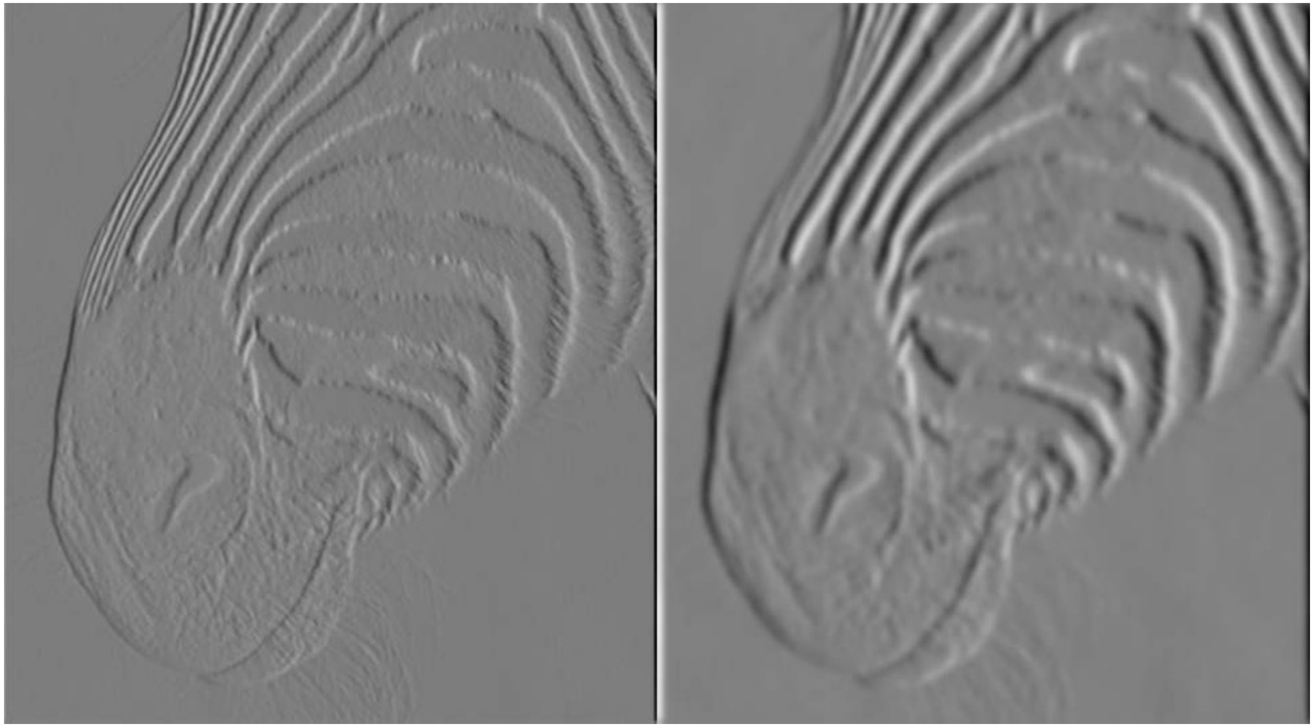
Tradeoff between smoothing and localization



1 pixel

Source: D. Forsyth

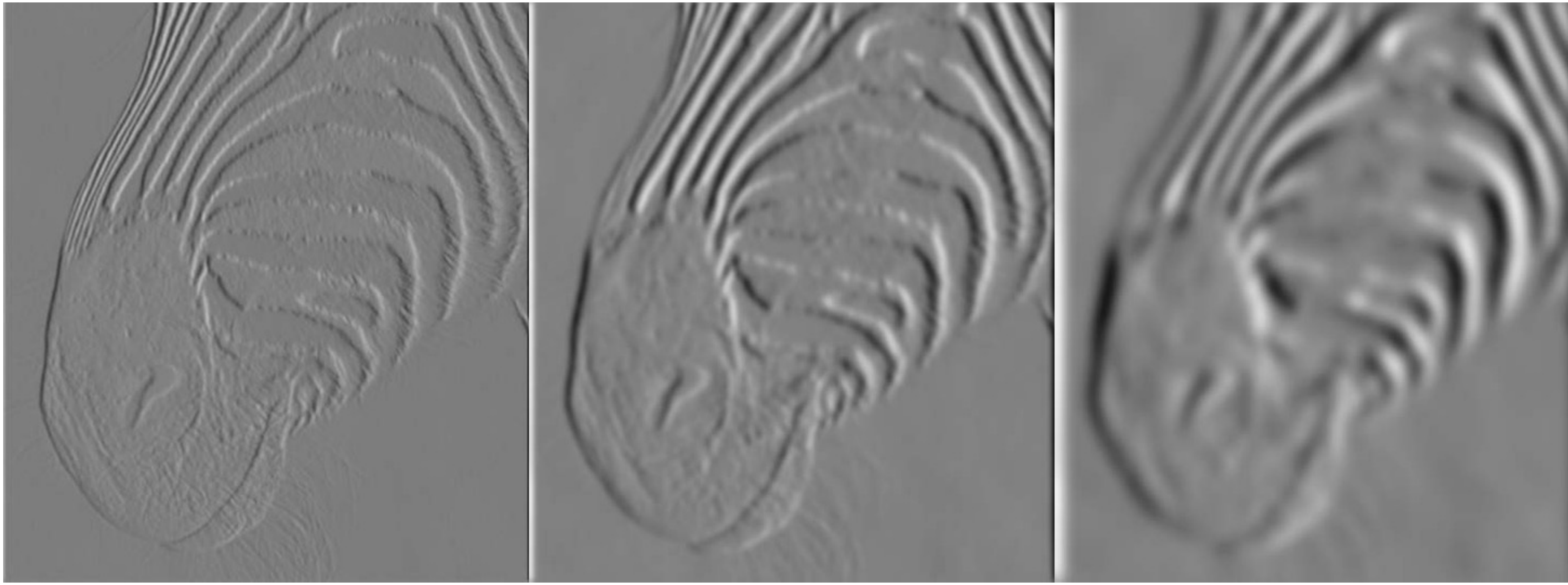
Tradeoff between smoothing and localization



1 pixel 3 pixels

Source: D. Forsyth

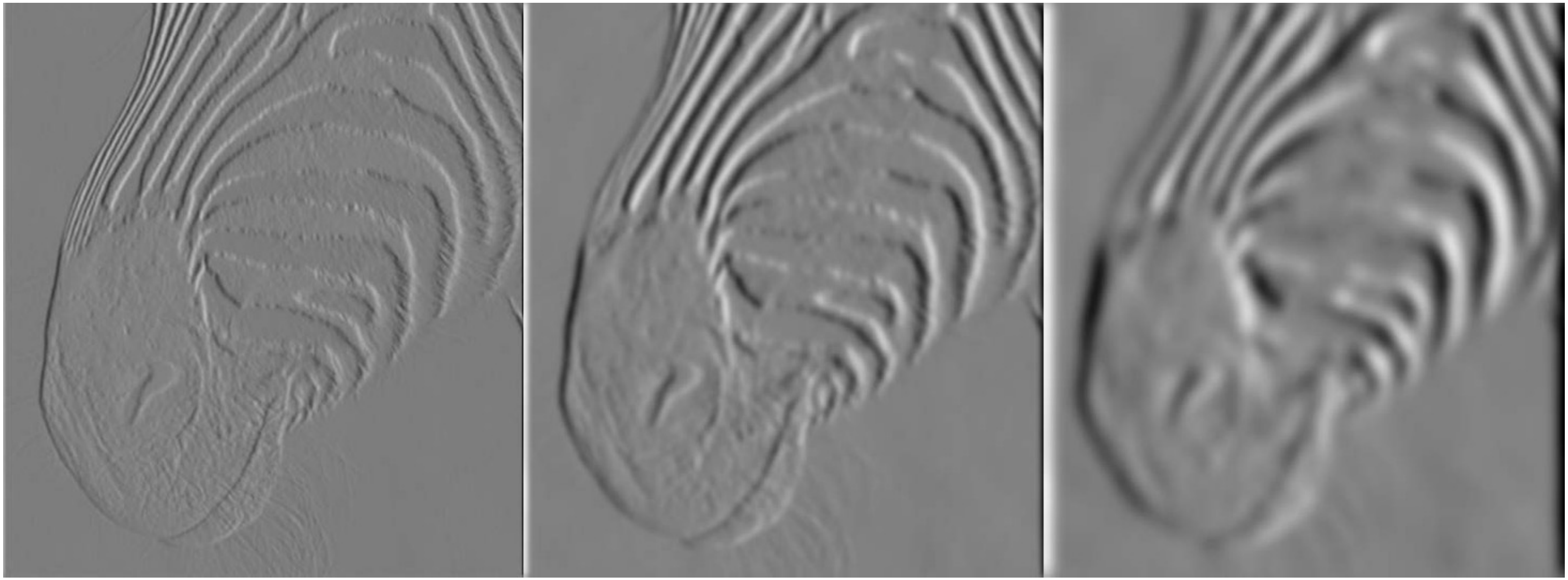
Tradeoff between smoothing and localization



1 pixel 3 pixels 7 pixels

Source: D. Forsyth

Tradeoff between smoothing and localization

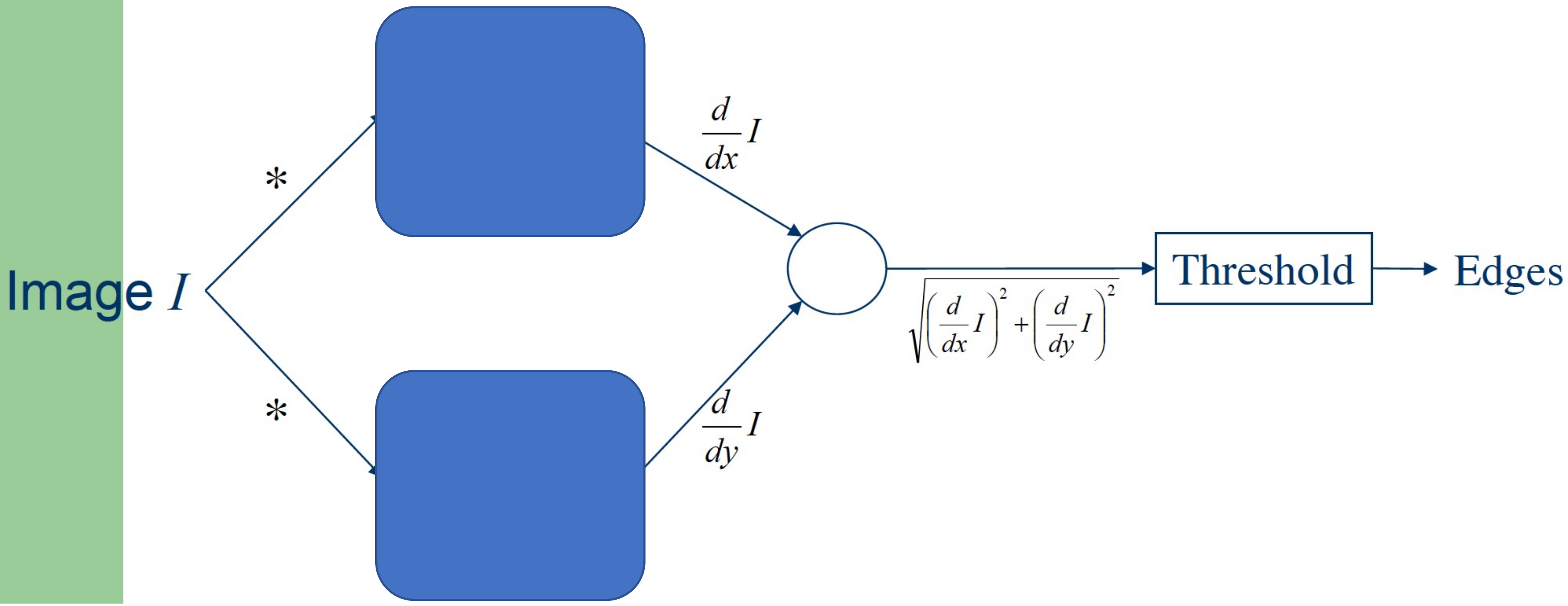


1 pixel 3 pixels 7 pixels

- Smoothed derivative removes noise, but blurs edge. Also finds edges at different “scales”.

Source: D. Forsyth

Edge Detectors



Edge Detectors

- Gradient operators
  - Prewit
  - Sobel

Edge Detectors

- Gradient operators
  - Prewit
  - Sobel
- Laplacian of Gaussian (Marr-Hildreth)

Edge Detectors

- Gradient operators
  - Prewit
  - Sobel
- Laplacian of Gaussian (Marr-Hildreth)
- Gradient of Gaussian (Canny)

Prewitt and Sobel Edge Detector

Prewitt and Sobel Edge Detector

- Compute derivatives
  - In x and y directions

Prewitt and Sobel Edge Detector

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

Prewitt and Sobel Edge Detector

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

Prewitt Edge Detector

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

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

Sobel Edge Detector

X-derivative

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

Y-derivative

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

Prewitt Edge Detector

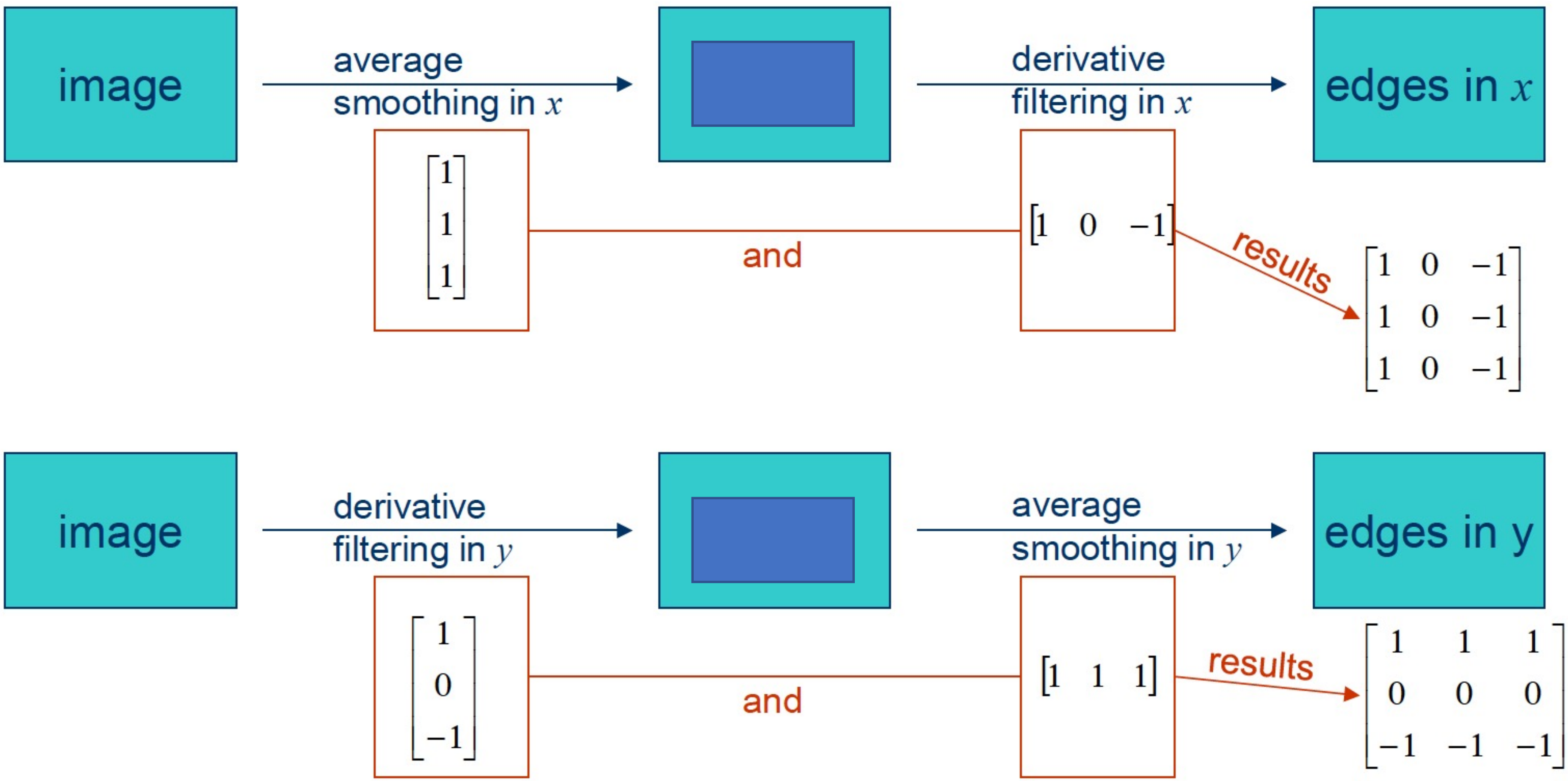
X-derivative

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

Y-derivative

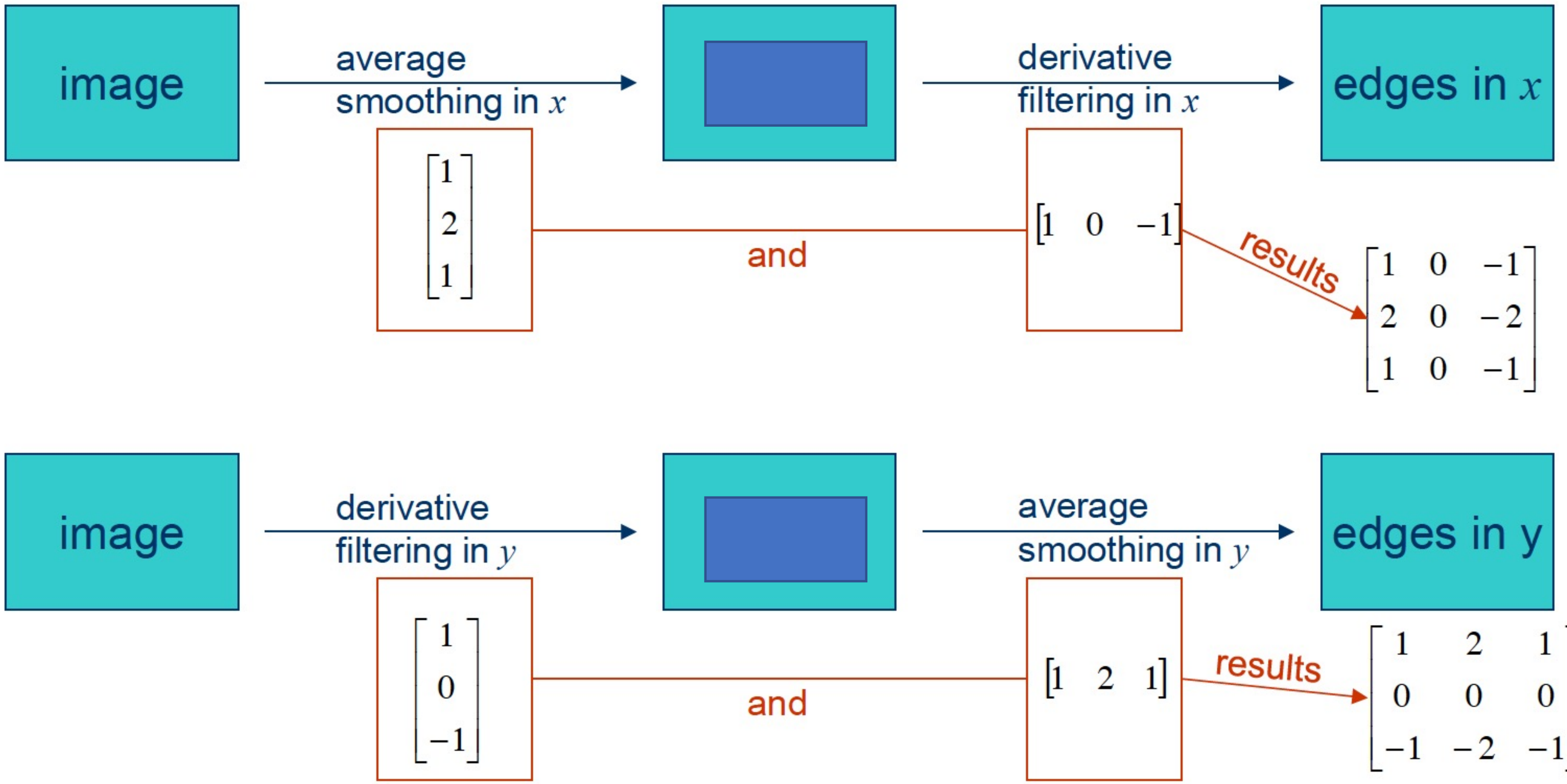
$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

Prewitt Edge Detector



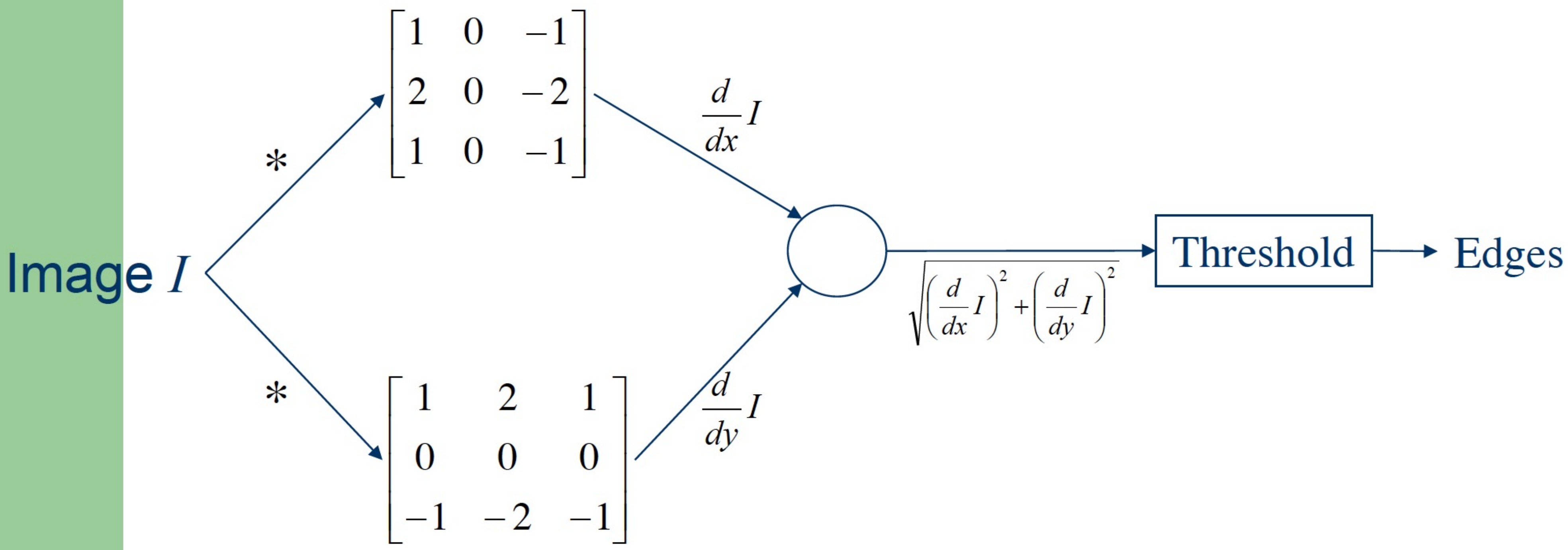
Alper Yilmaz, Mubarak Shah Fall 2012UCF

Sobel Edge Detector

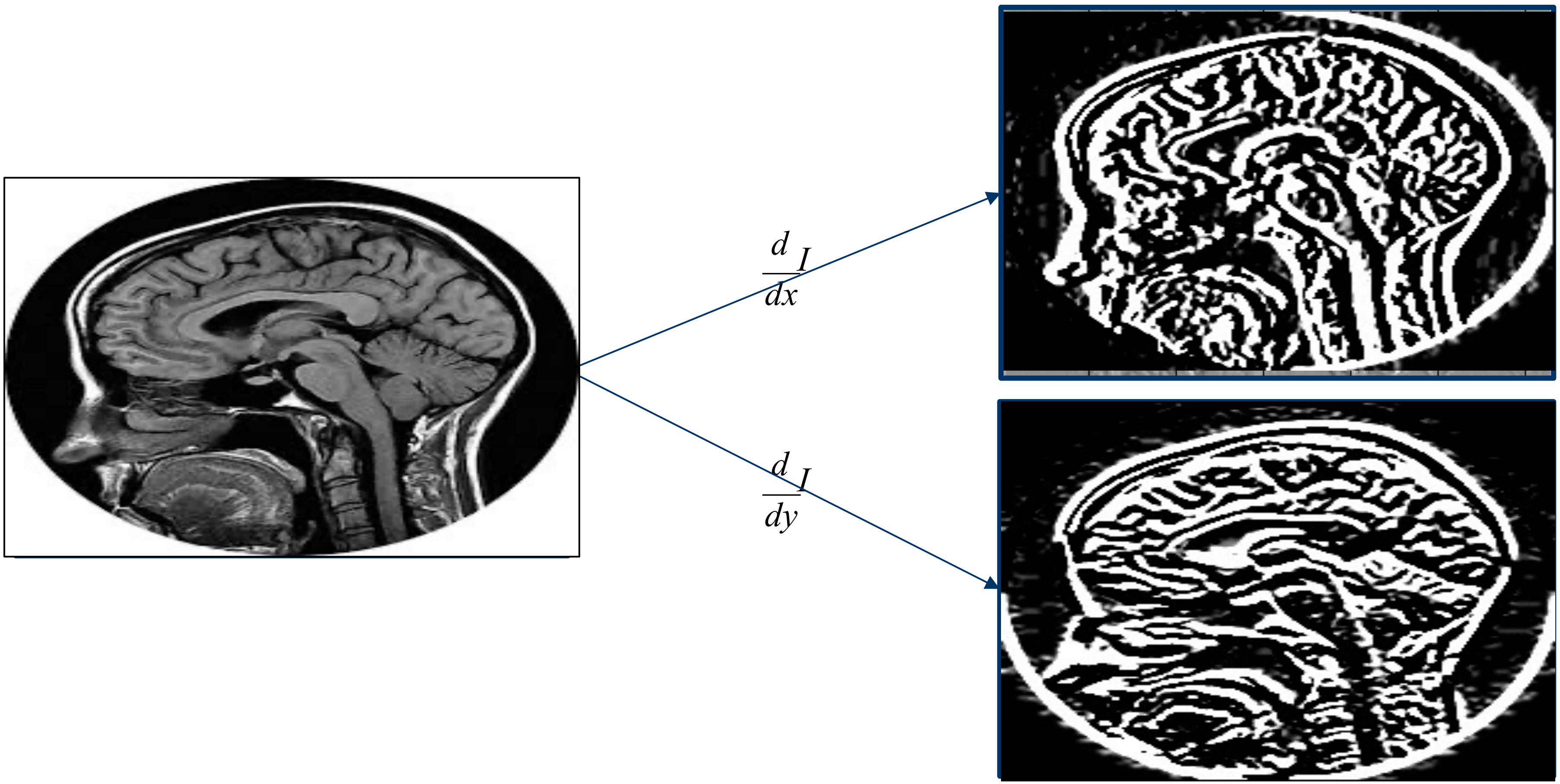


Alper Yilmaz, Mubarak Shah Fall 2012UCF

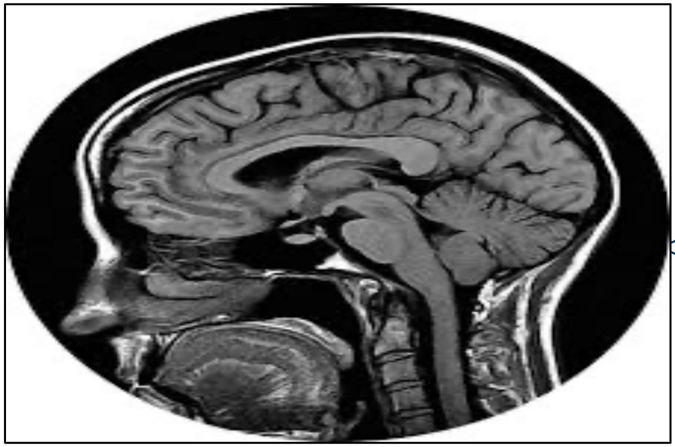
Sobel Edge Detector



Sobel Edge Detector



Sobel Edge Detector



$$\Delta \geq Threshold = 150$$

$$\Delta = \sqrt{\left(\frac{d}{dx} I\right)^2 + \left(\frac{d}{dy} I\right)^2}$$

Laplacian of Gaussian (LOG) Edge Detector

- Smooth image by Gaussian filter → S

Laplacian of Gaussian (LOG) Edge Detector

- Smooth image by Gaussian filter → S
- Apply Laplacian to S
  - Used in mechanics, electromagnetics, wave theory, quantum mechanics and Laplace equation
- Find zero crossings
  - Scan along each row, record an edge point at the location of zero-crossing.
  - Repeat above step along each column

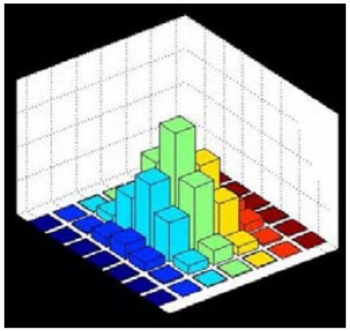
Finding Zero Crossings

- Four cases of zero-crossings :
  - {+,-}
  - {+,0,-}
  - {-,+}
  - {-,0,+}
- Slope of zero-crossing {a, -b} is |a+b|.
- To mark an edge
  - compute slope of zero-crossing
  - Apply a threshold to slope

Laplacian of Gaussian (LOG) Edge Detector

- Gaussian smoothing

$$\overbrace{S}^{\text{smoothed image}} = \overbrace{g}^{\text{Gaussian filter}} * \overbrace{I}^{\text{image}} \qquad g = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



- Find Laplacian

$$\Delta^2 S = \overbrace{\frac{\partial^2}{\partial x^2} S}^{\text{second order derivative in } x} + \overbrace{\frac{\partial^2}{\partial y^2} S}^{\text{second order derivative in } y}$$

- ∇ is used for gradient (first derivative)
- Δ^2 is used for Laplacian (Second derivative)

Alper Yilmaz, Mubarak Shah Fall 2012, UCF

Laplacian of Gaussian (LOG) Edge Detector

- Deriving the Laplacian of Gaussian (LoG)

$$\Delta^2 S = \Delta^2 (g * I) = (\Delta^2 g) * I \qquad g = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Difference of Gaussians ~ Laplacian

Gaussian

0.01	0.08	0.01
0.08	0.64	0.08
0.01	0.08	0.01

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

0.05	0	−0.05
0.34	0	−0.34
0.05	0	−0.05

$\frac{\partial}{\partial y}$

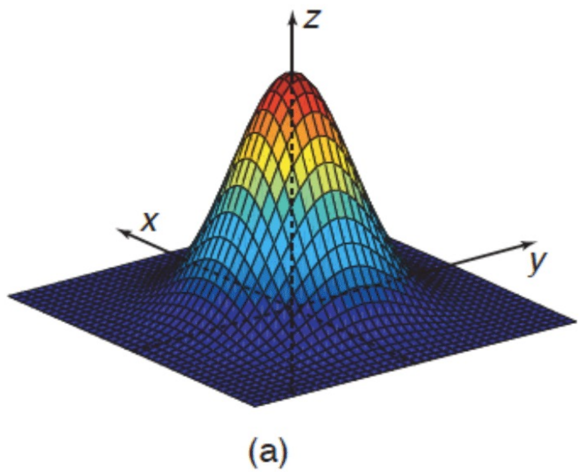
0.05	0.34	0.05
0	0	0
−0.05	−0.34	−0.05

∇^2

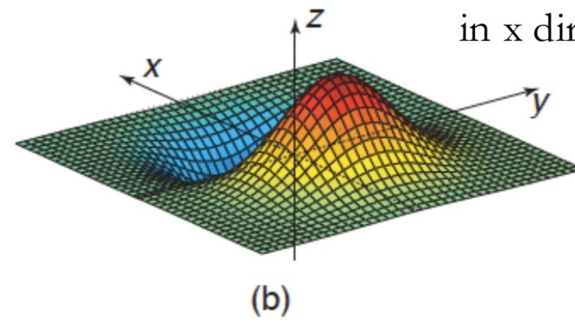
0.3	0.7	0.3
0.7	−4	0.7
0.3	0.7	0.3

Laplacian: Difference of Gaussians

Gaussian

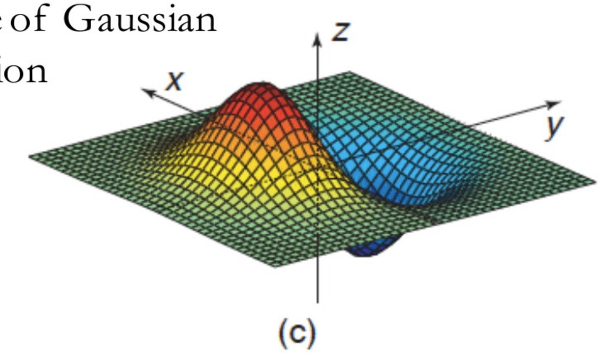


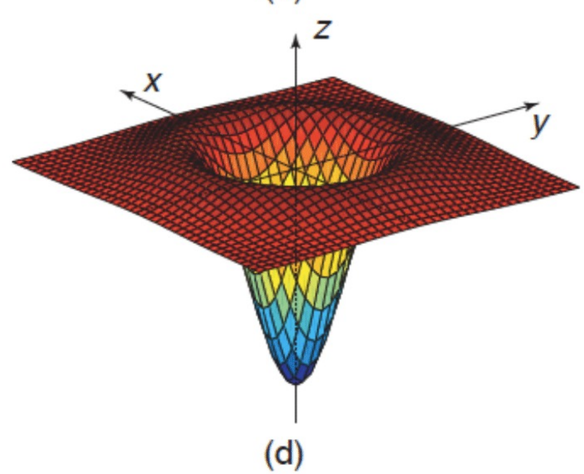
Derivative of Gaussian
in x direction
(gradient)



Derivative of Gaussian
in y direction

(gradient)





Laplacian of
Gaussian

Quality of an Edge

Quality of an Edge

- Robust to noise

Quality of an Edge

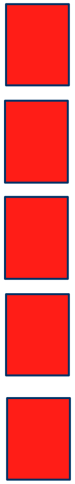
- Robust to noise
- Localization

Quality of an Edge

- Robust to noise
- Localization
- Too many or too less responses

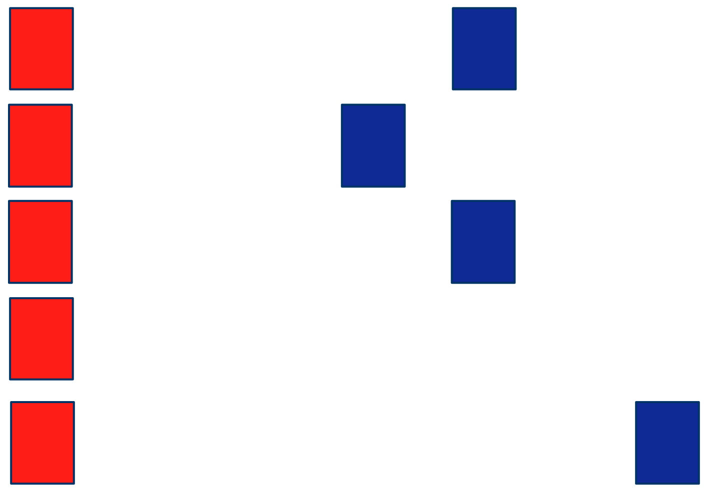
Quality of an Edge

Quality of an Edge



True
edge

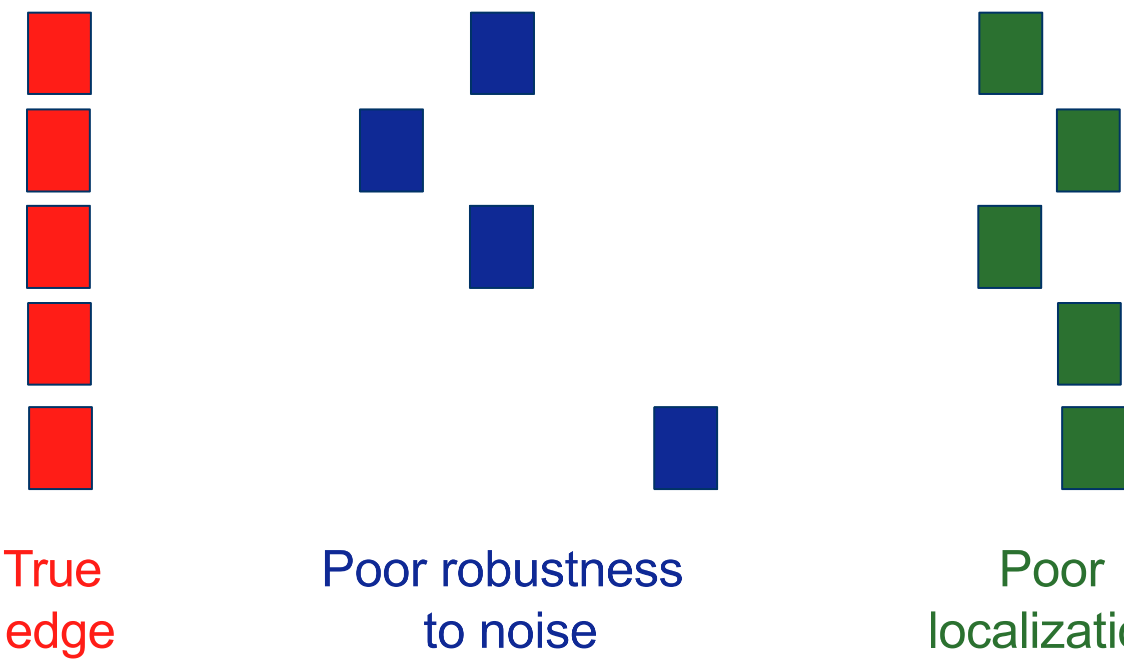
Quality of an Edge



True edge

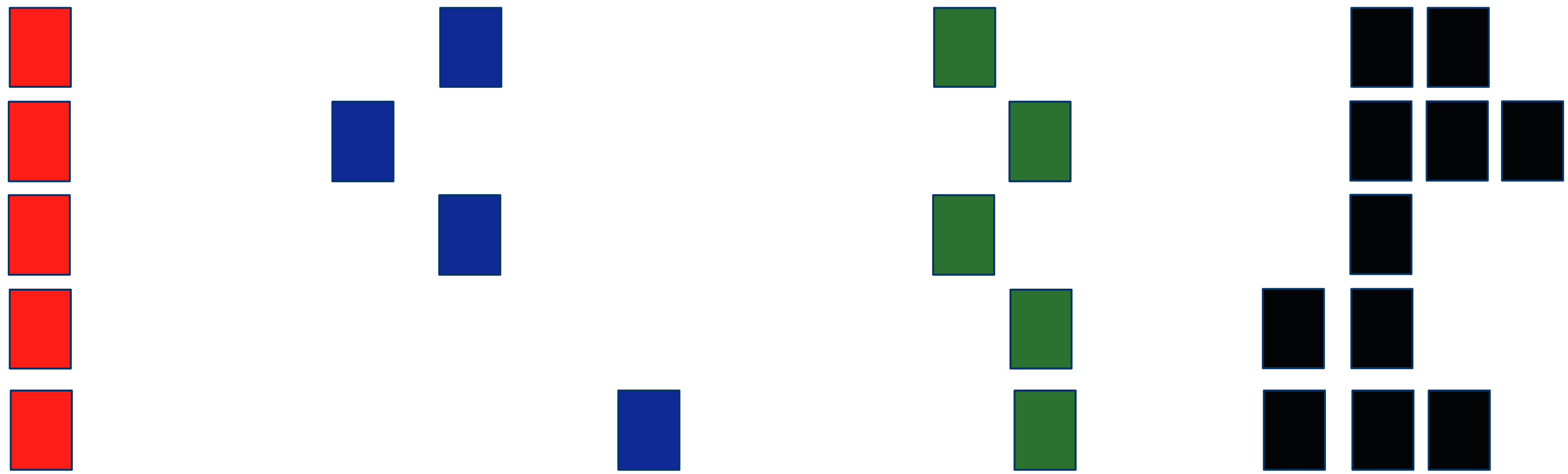
Poor robustness to noise

Quality of an Edge



Alper Yilmaz, Mubarak Shah Fall 2012, UCF

Quality of an Edge



True edge

Poor robustness to noise

Poor localization

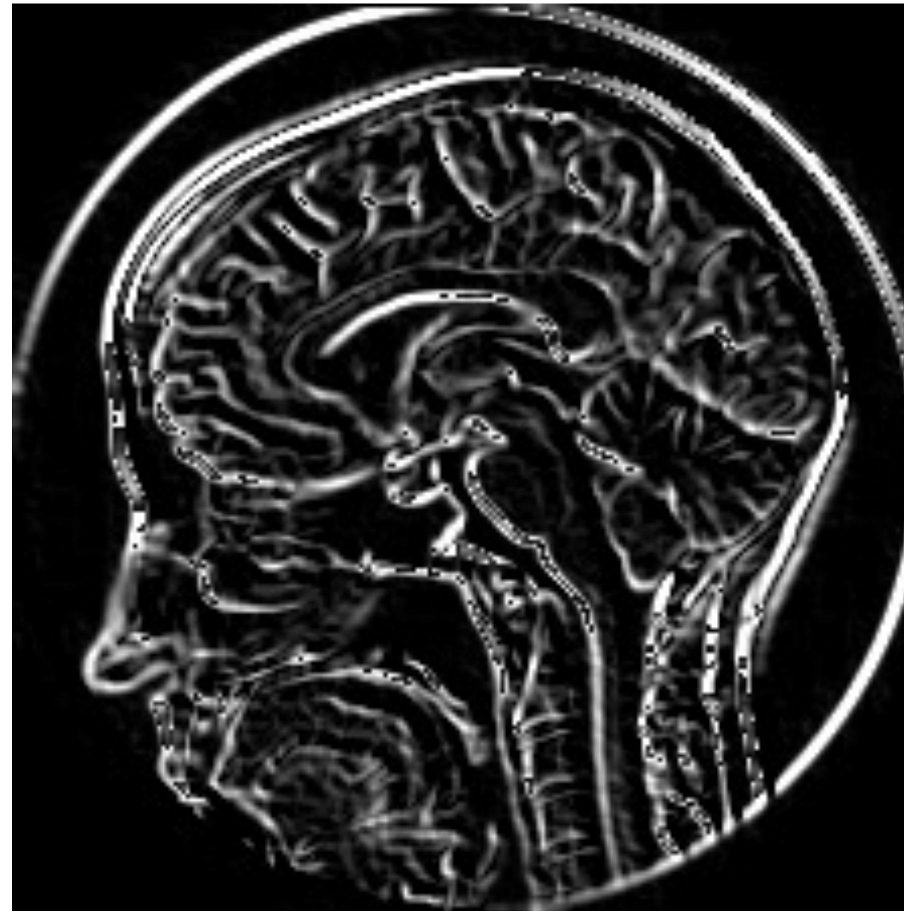
Too many responses

Alper Yilmaz, Mubarak Shah Fall 2012, UCF

Brain MRI Image

Sobel Edge

Brain MRI Image



Prewitt Edge

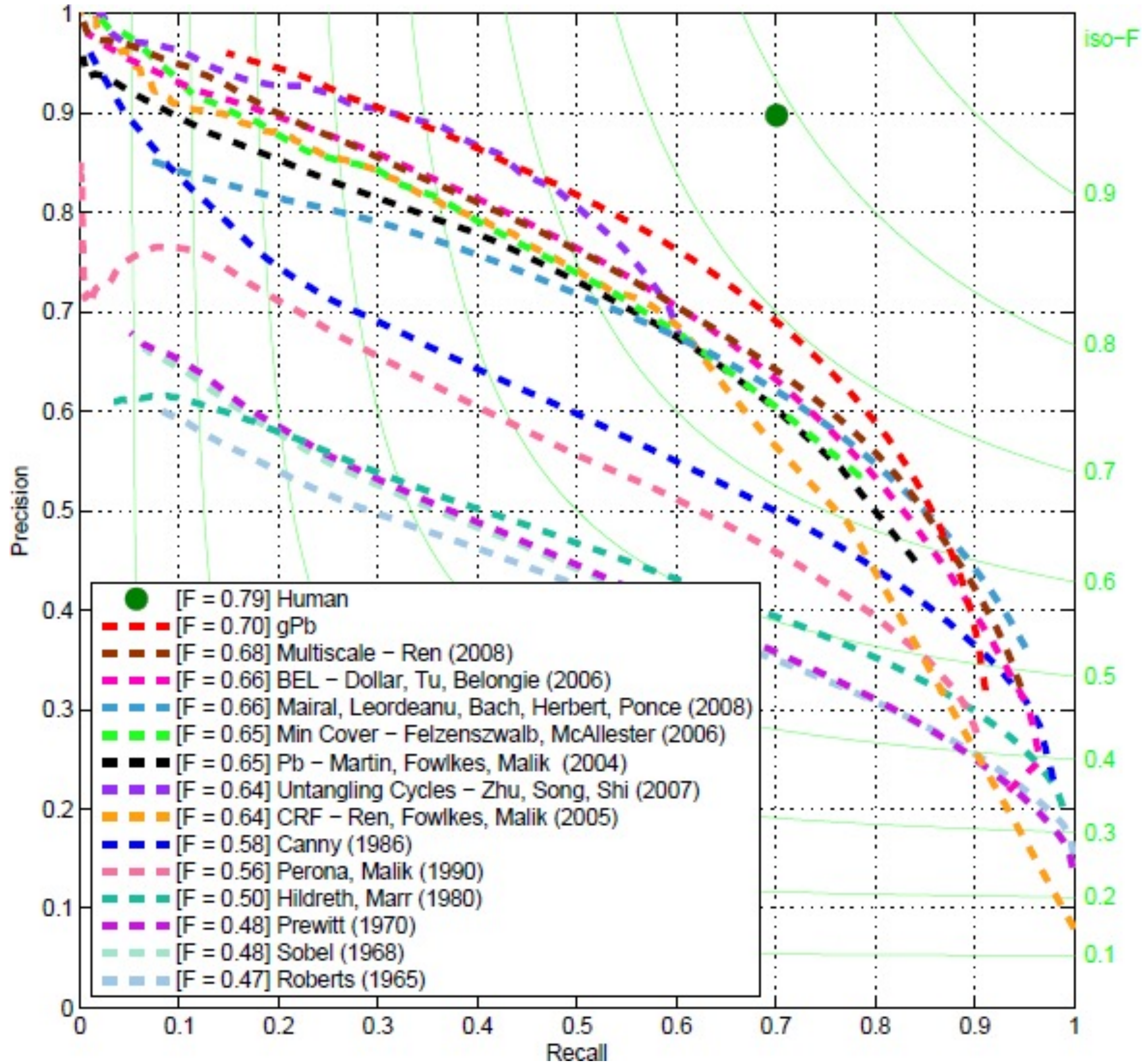
Brain MRI Image

Laplacian

Brain MRI Image

Canny Edge

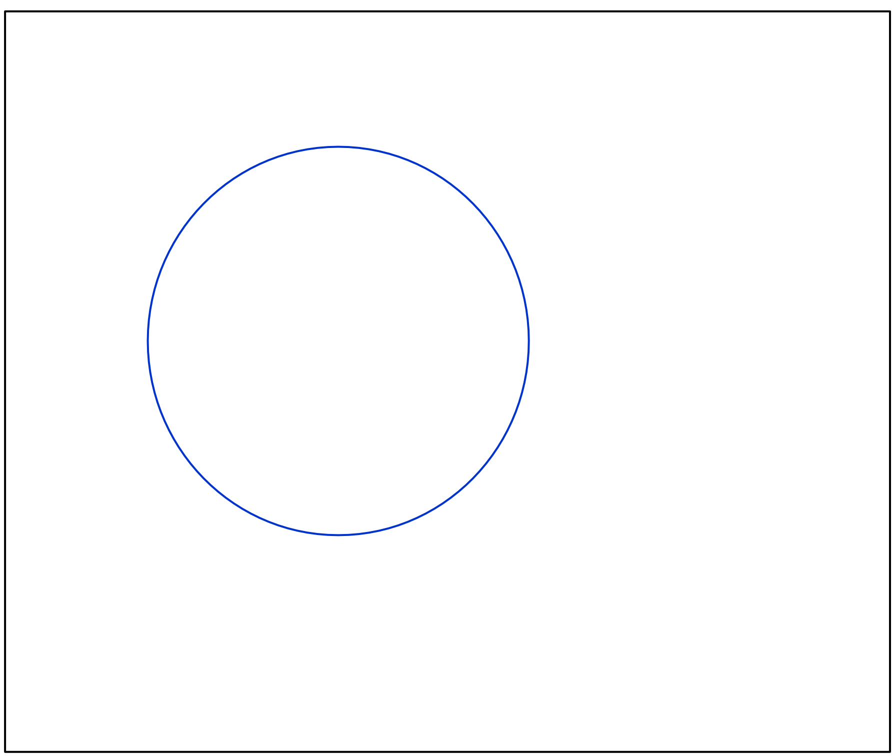
45 years of boundary detection



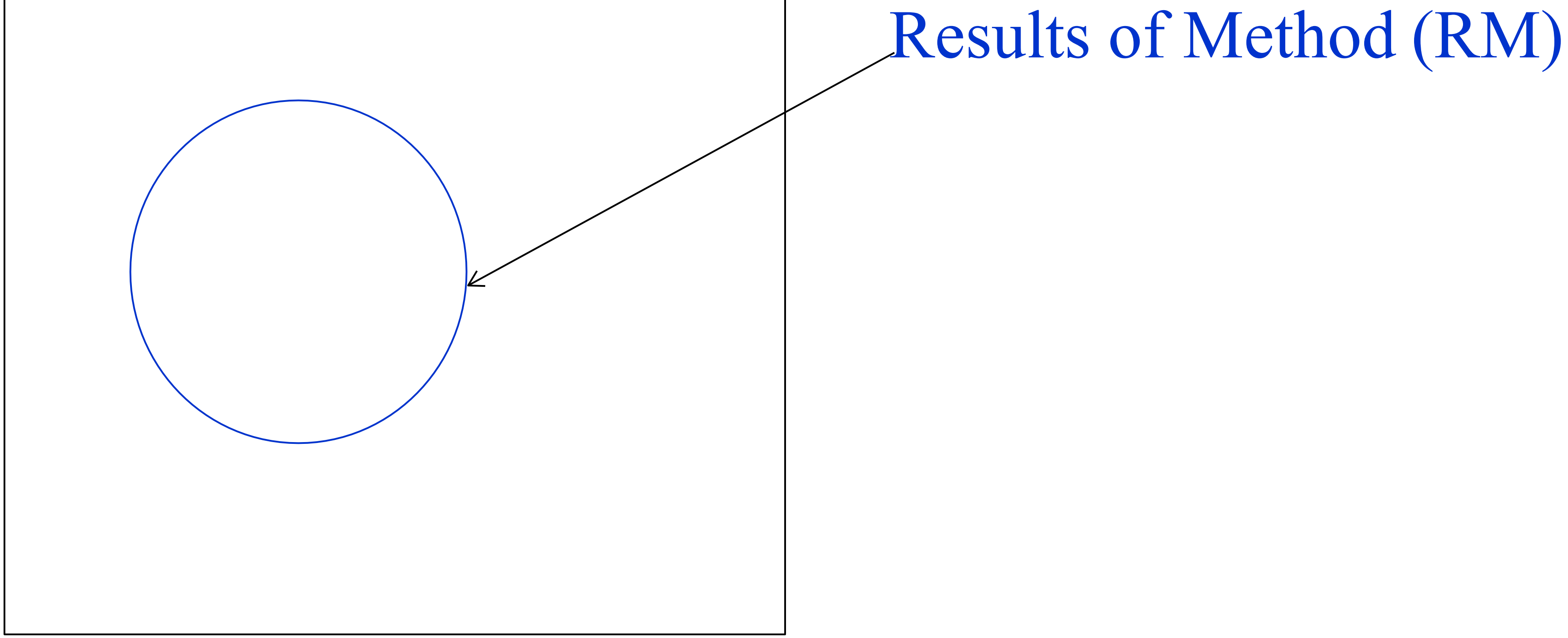
Source: Arbelaez, Maire, Fowlkes, and Malik. TPAMI 2011 (pdf)

Evaluation Metrics

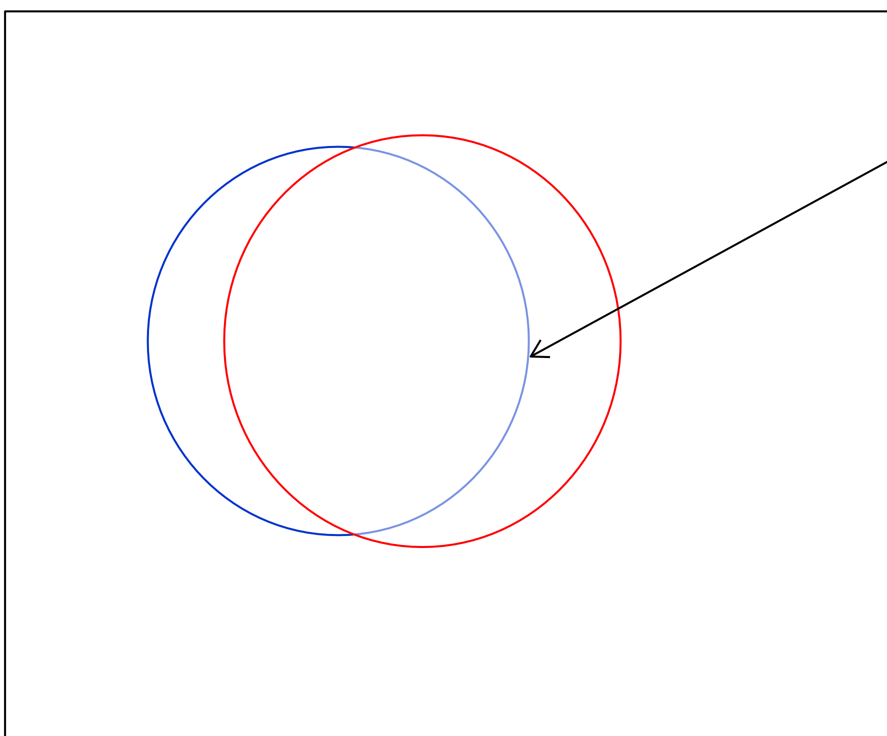
Evaluation Metrics



Evaluation Metrics

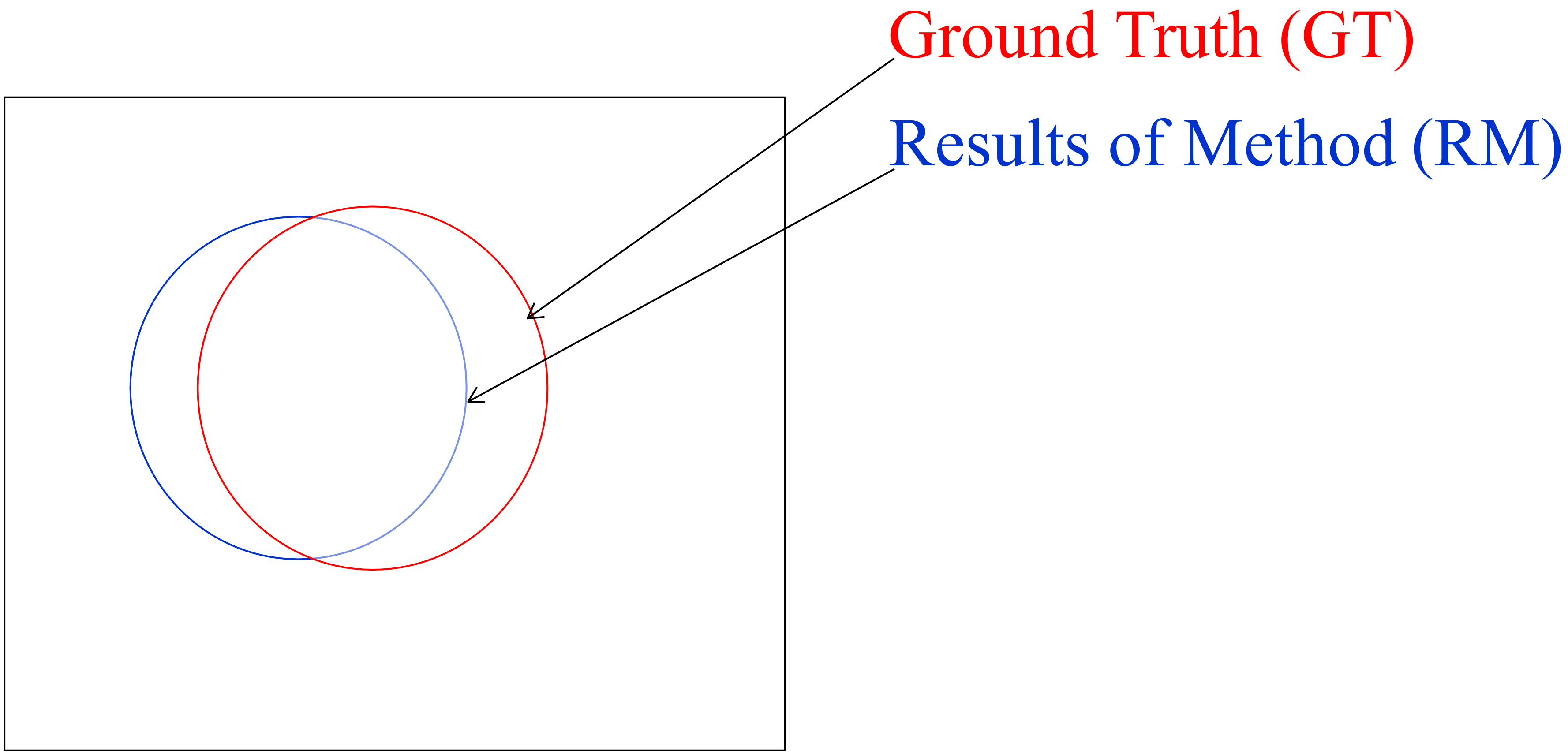


Evaluation Metrics

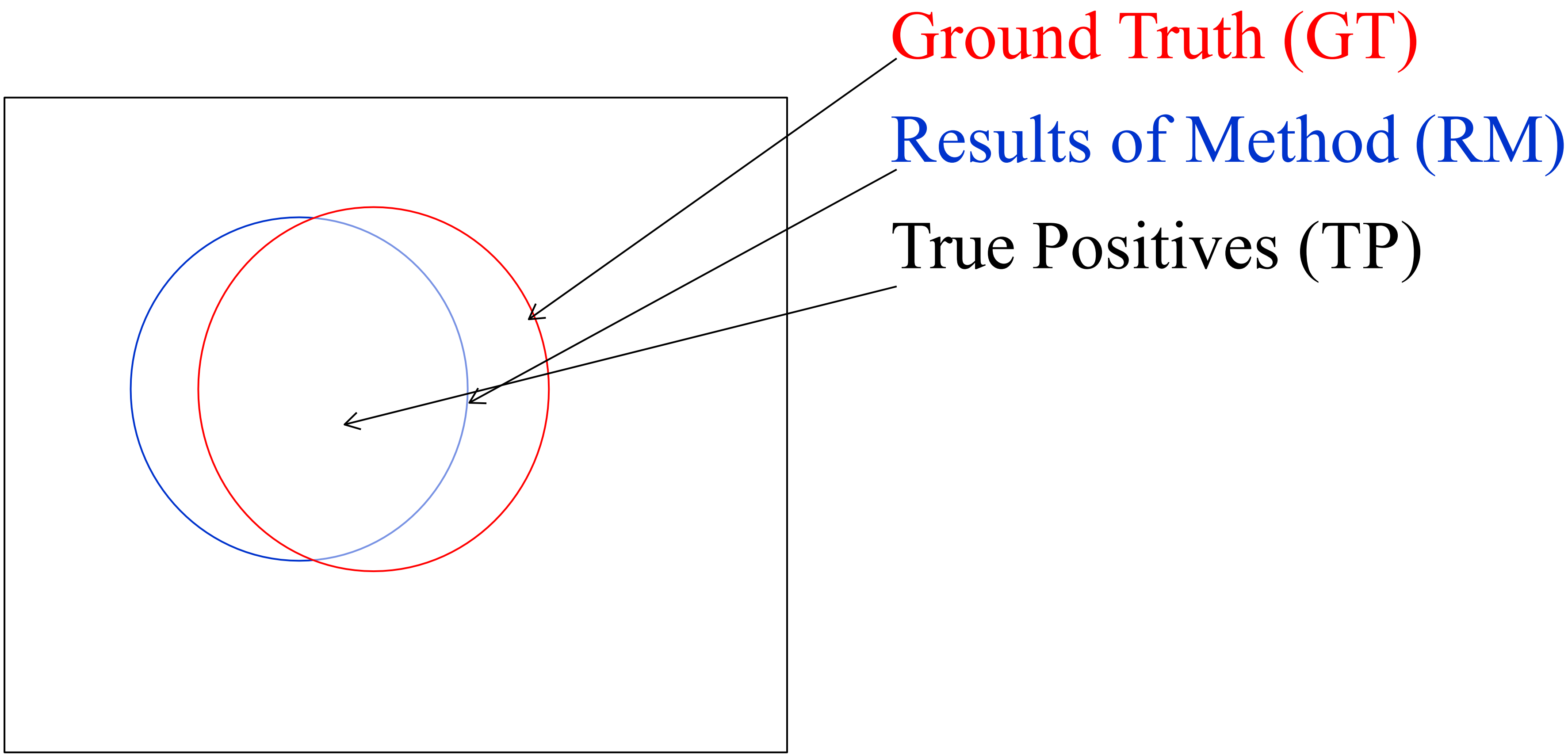


Results of Method (RM)

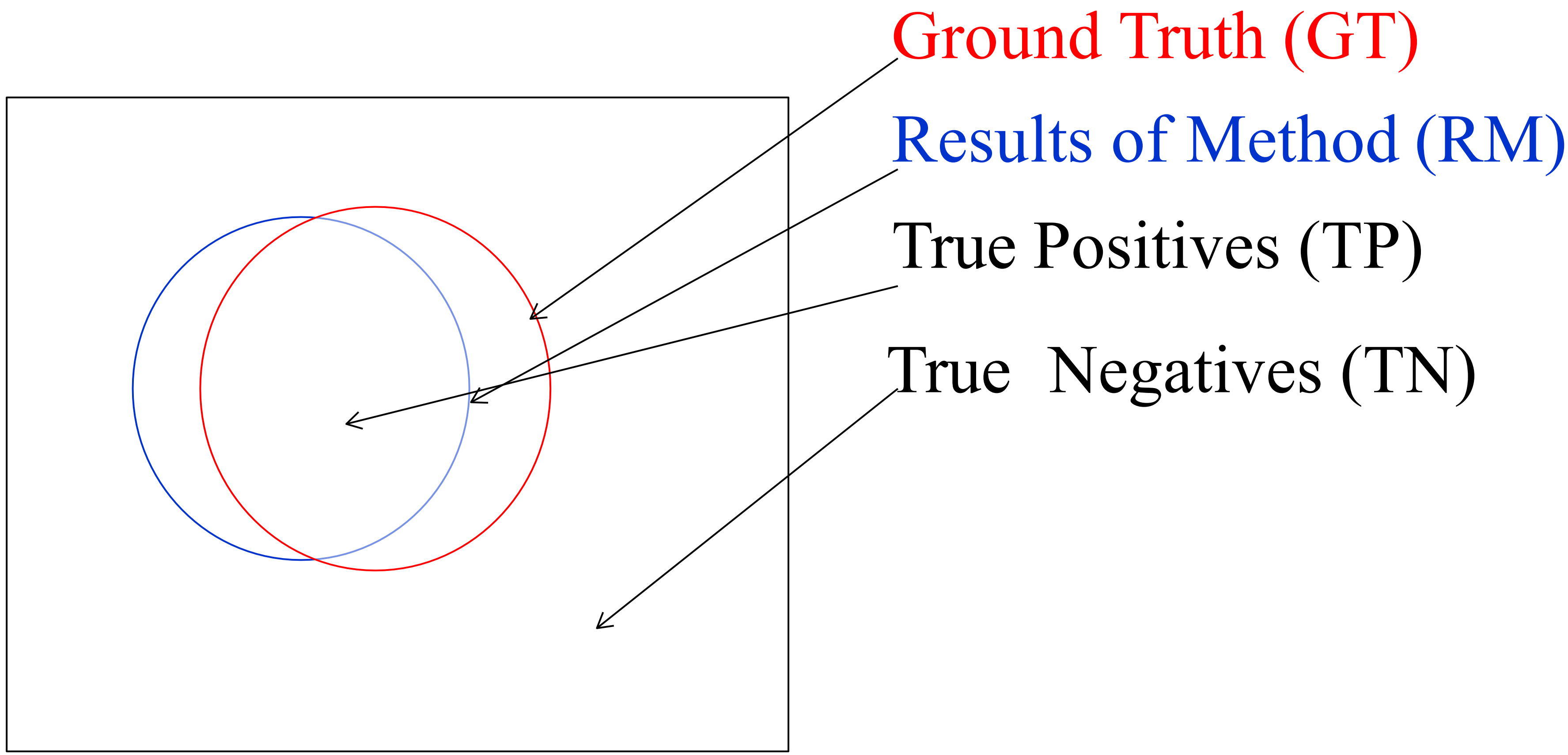
Evaluation Metrics



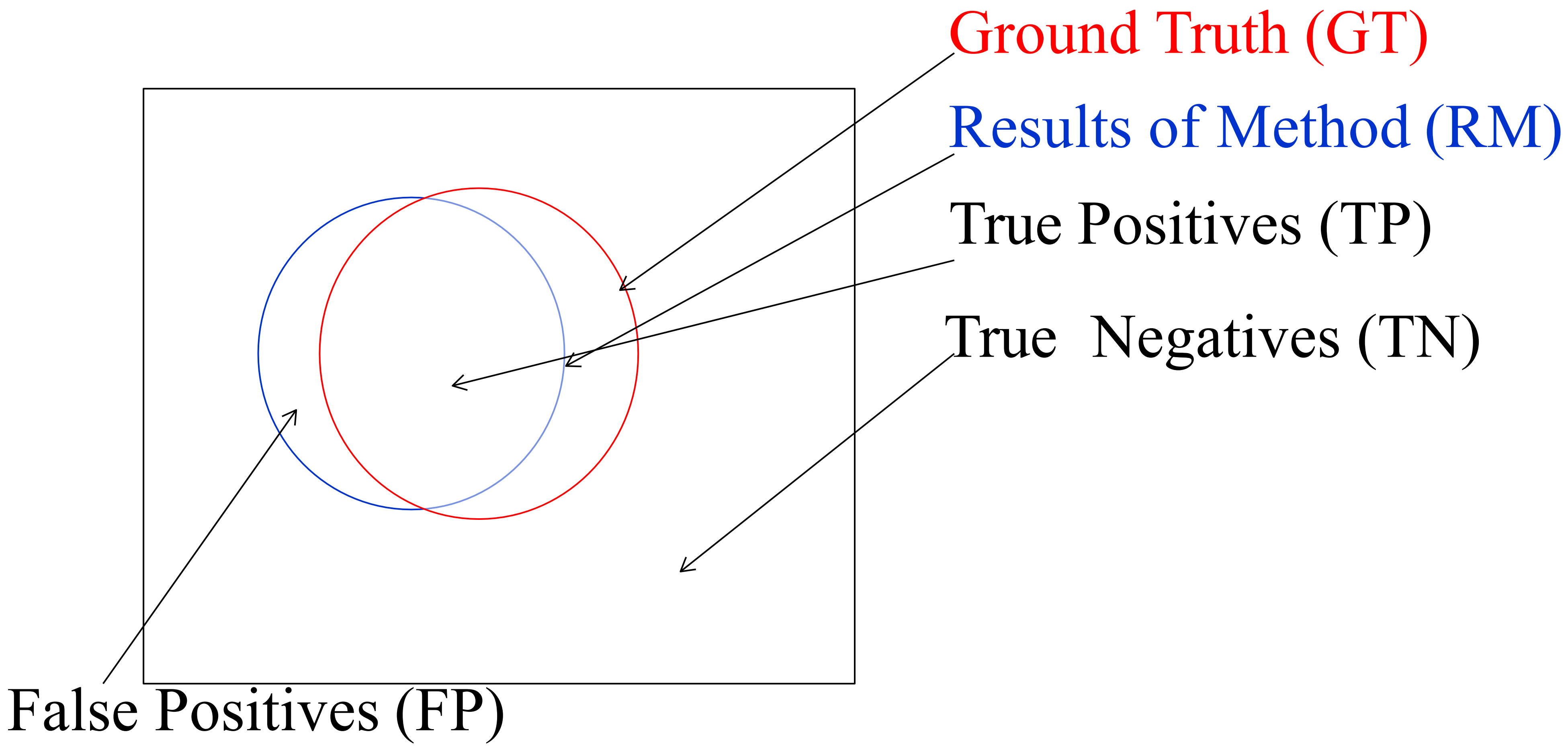
Evaluation Metrics



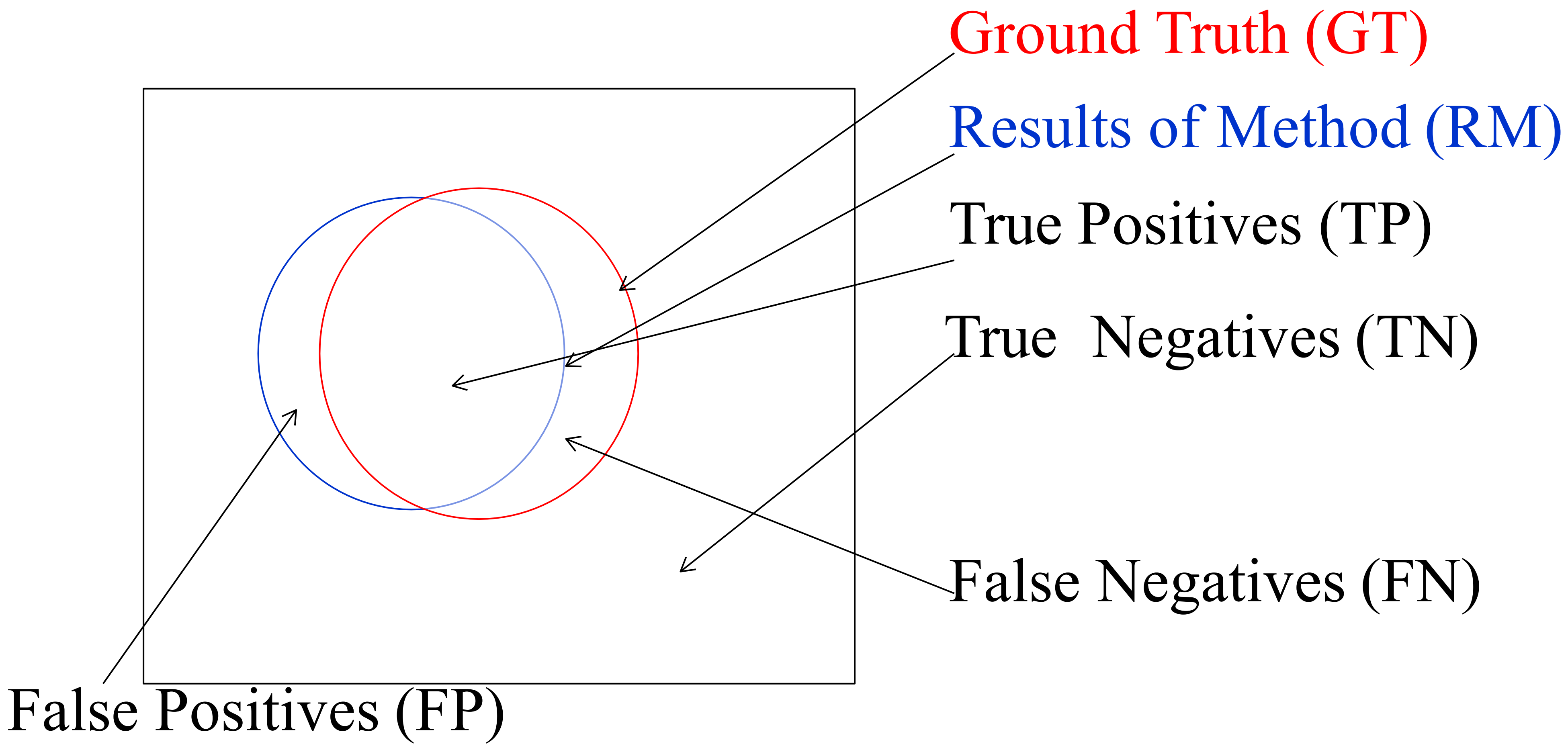
Evaluation Metrics



Evaluation Metrics



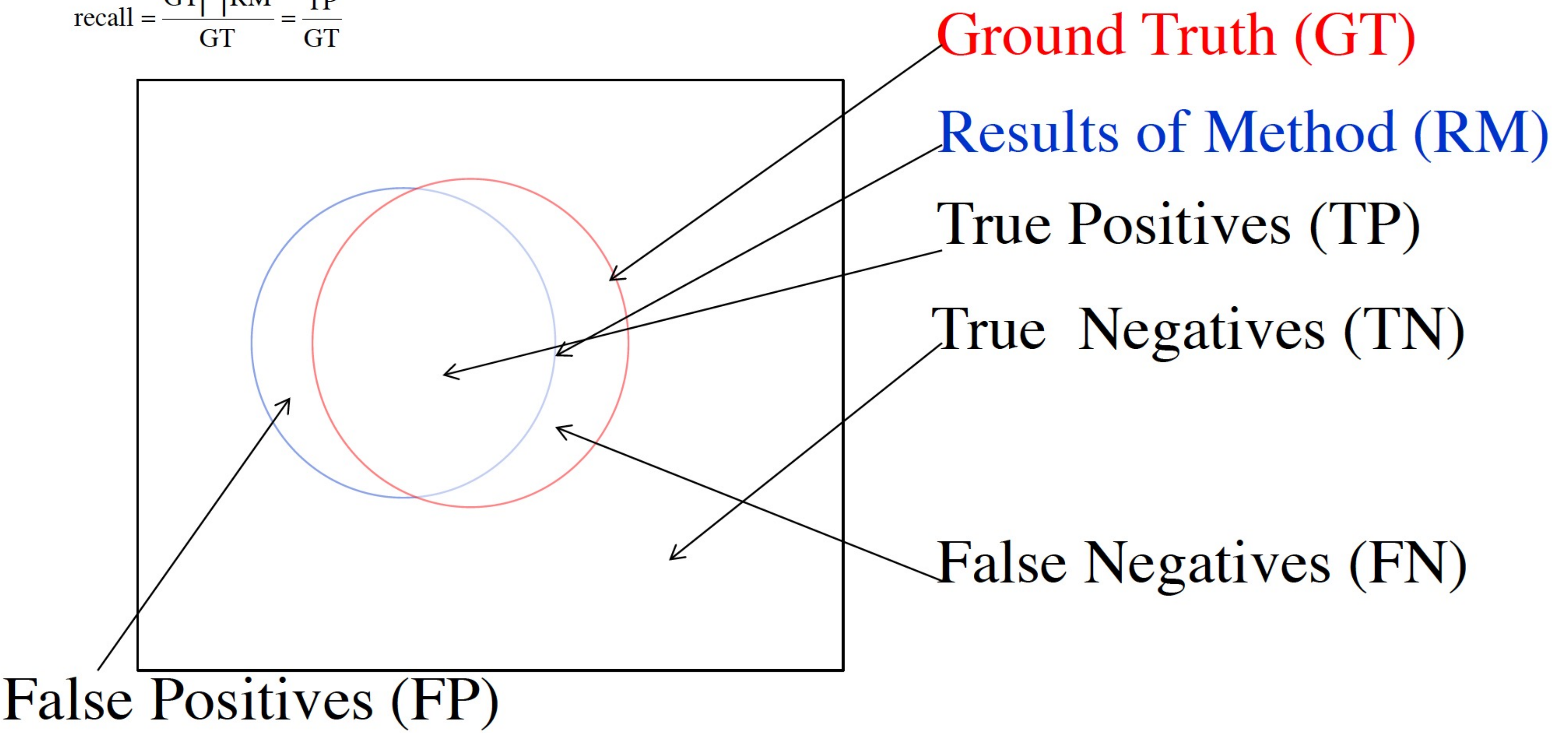
Evaluation Metrics

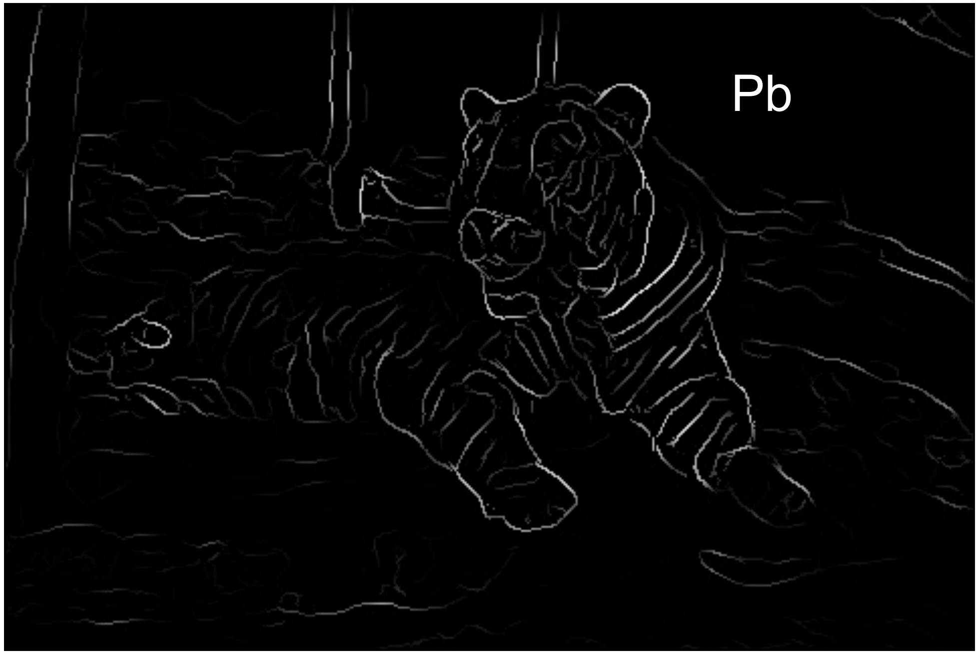


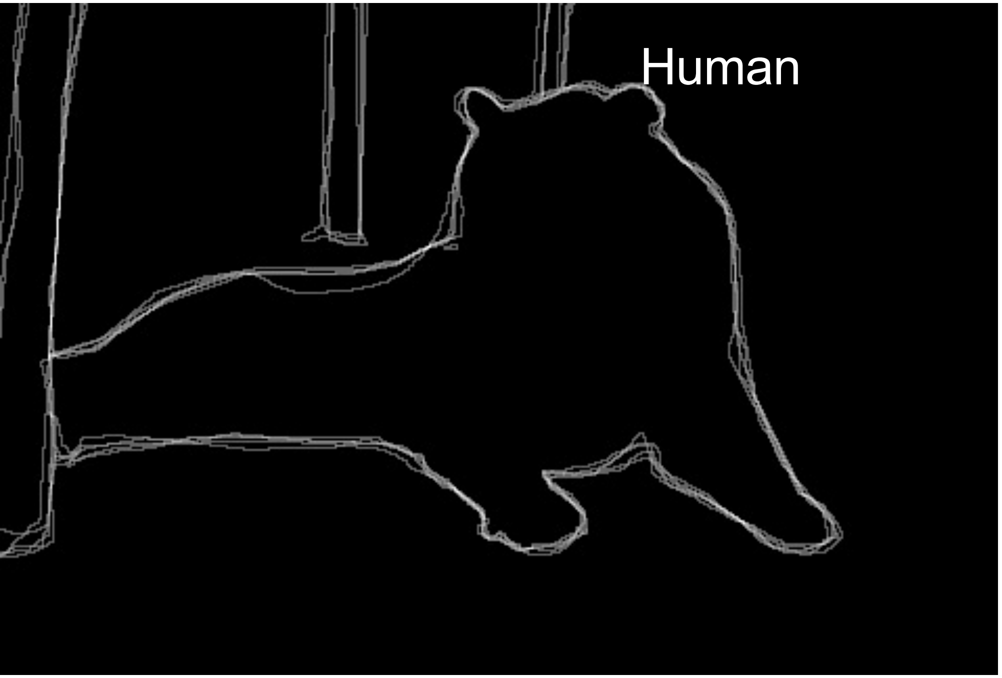
Evaluation Metrics

$$\text{precision} = \frac{GT \bigcap RM}{RM} = \frac{TP}{RM}$$

$$\text{recall} = \frac{GT \bigcap RM}{GT} = \frac{TP}{GT}$$



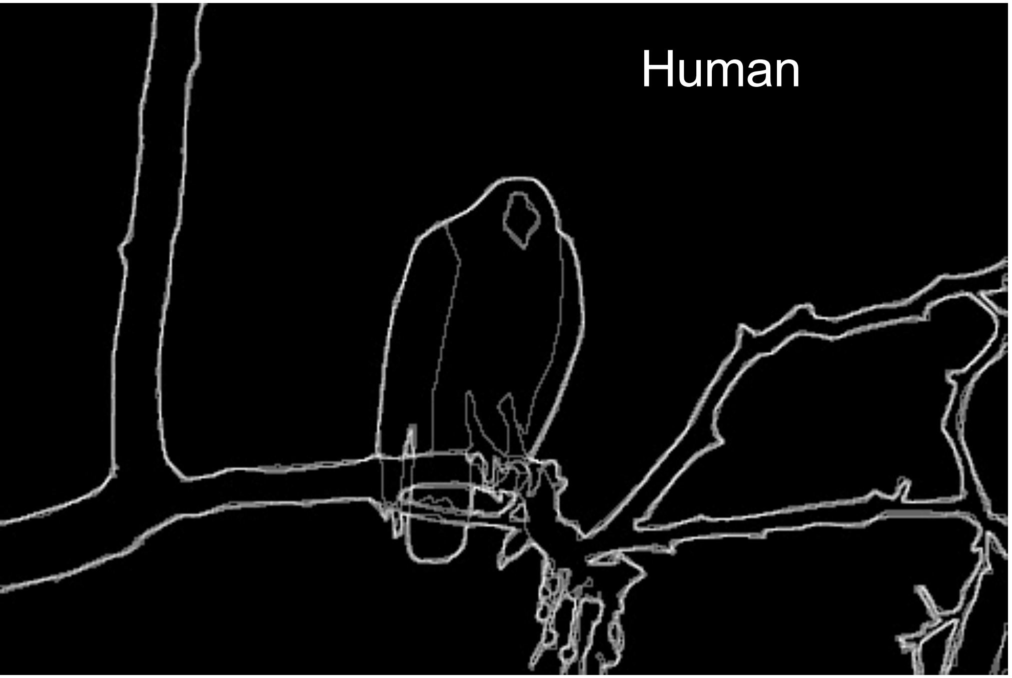


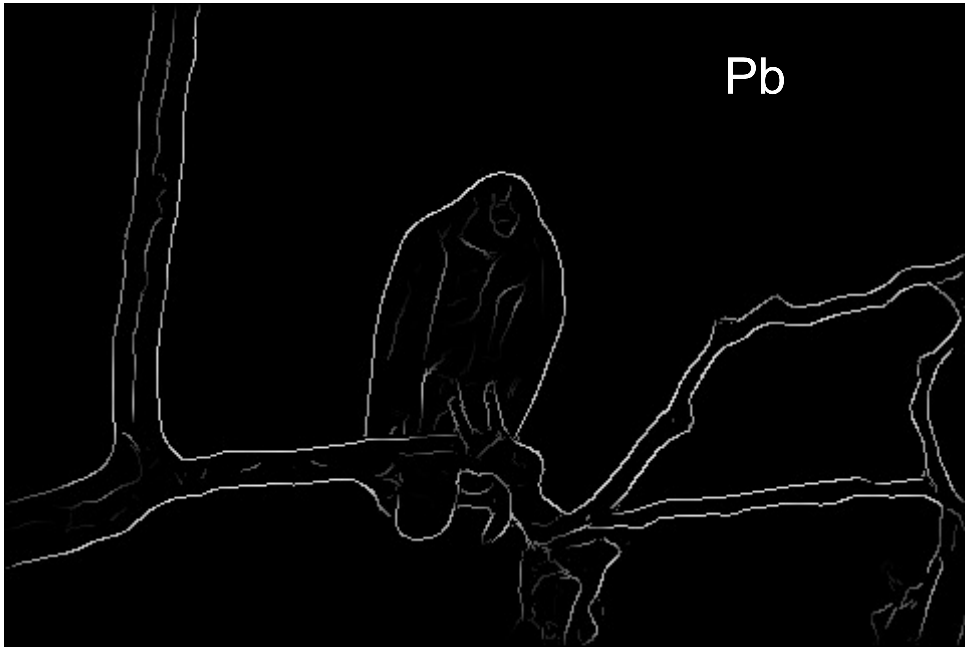


Slide Credit: James Hays

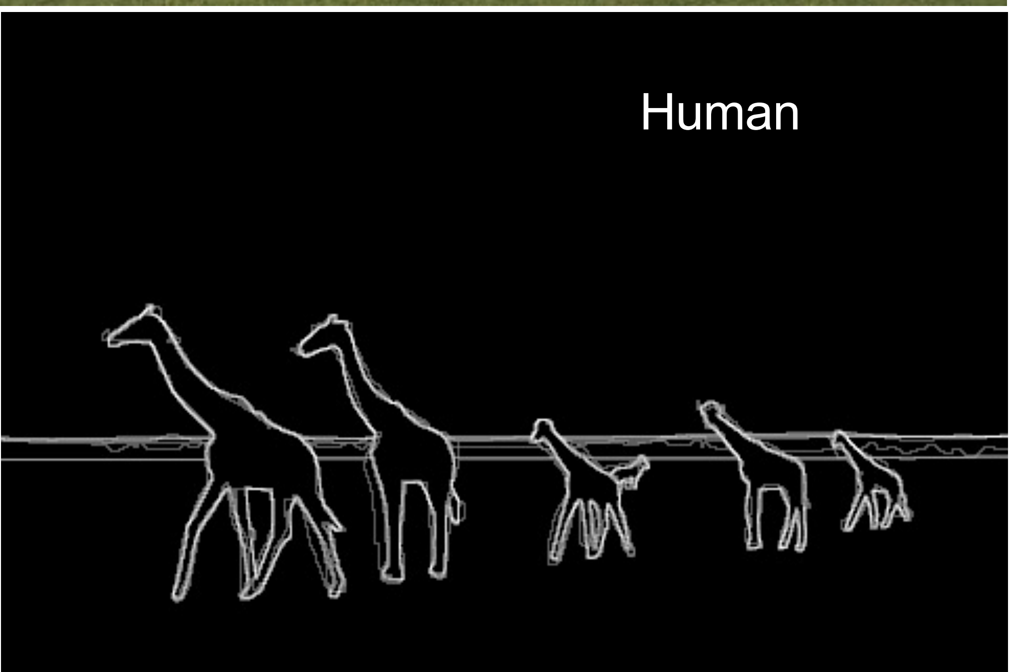
For more:
http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/108082-color.html

Results

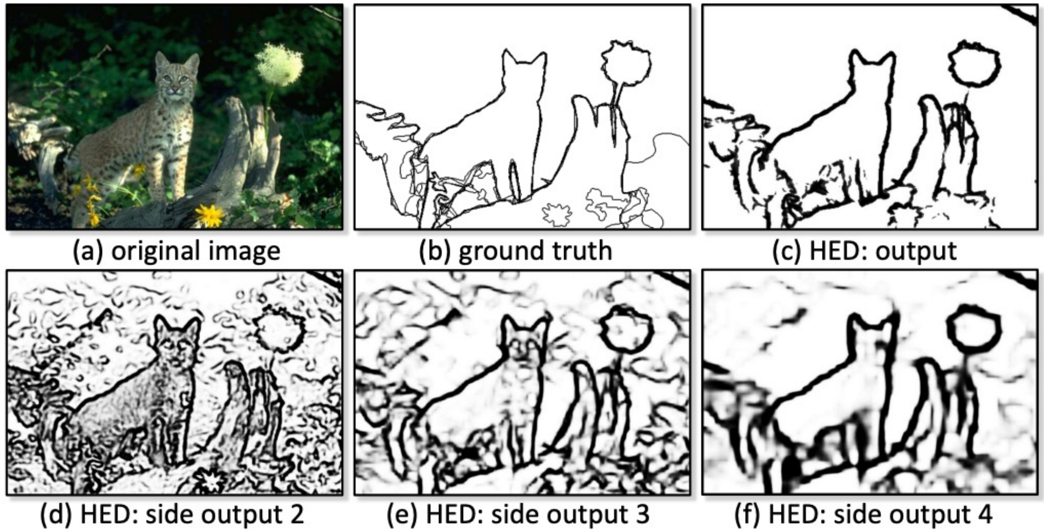


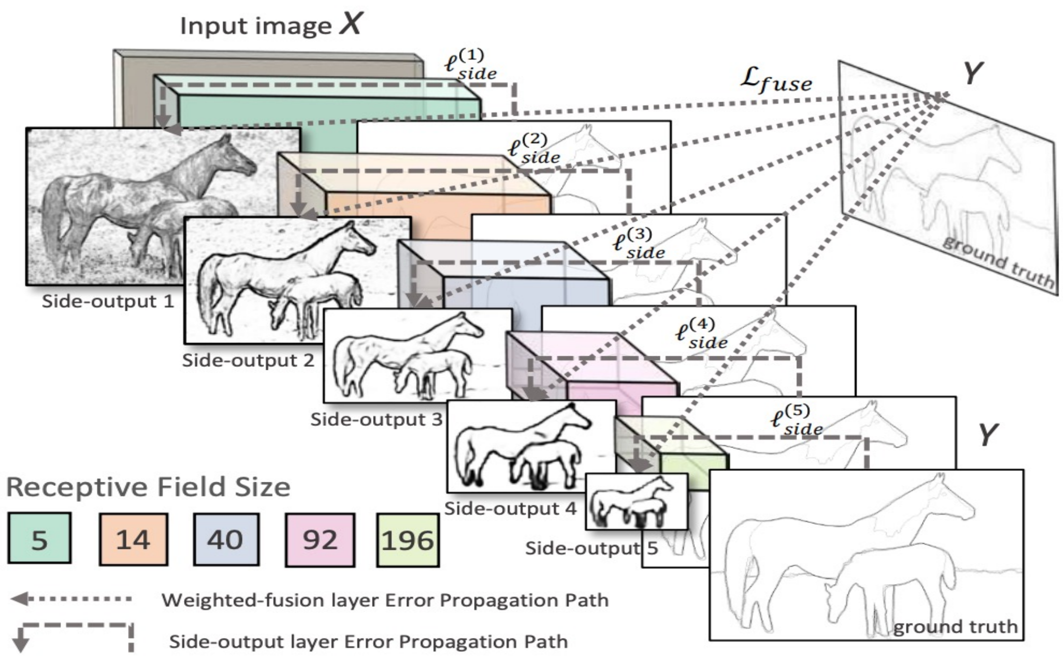


Slide Credit: James Hays



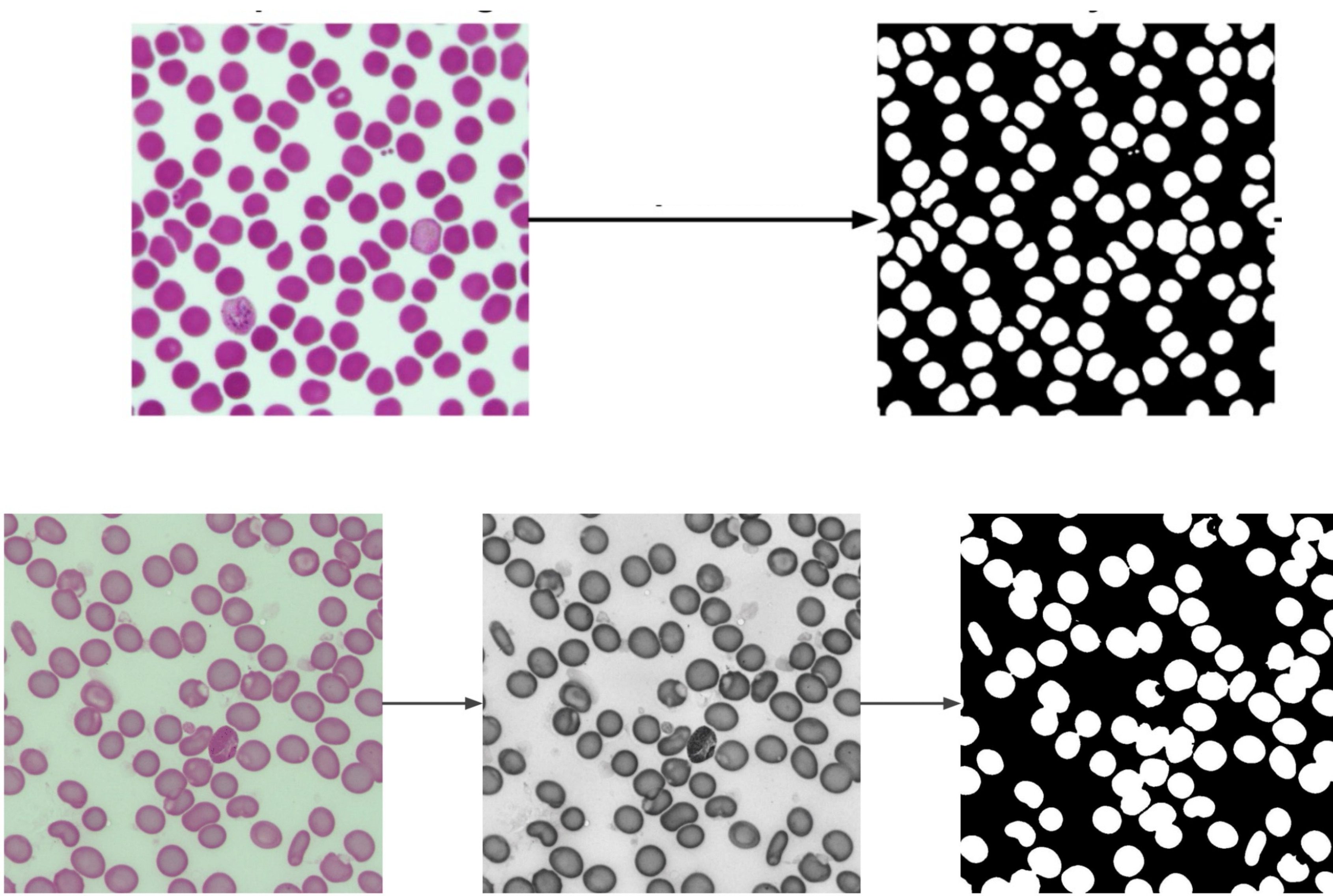
Slide Credit: James Hays



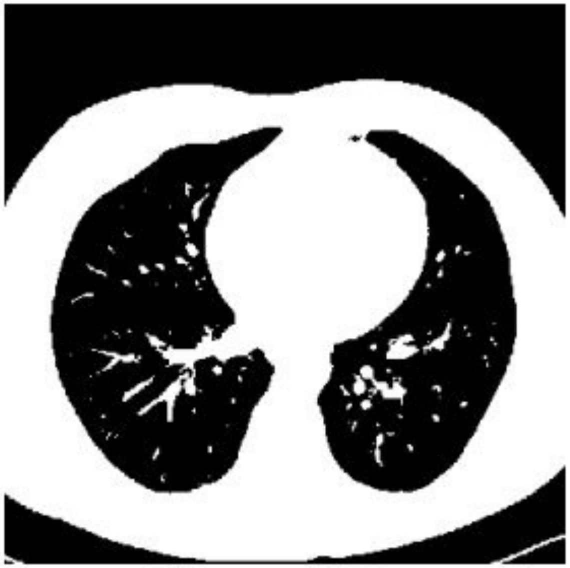


https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Xie_Holistically-Nested_Edge_Detection_ICCV_2015_paper.pdf

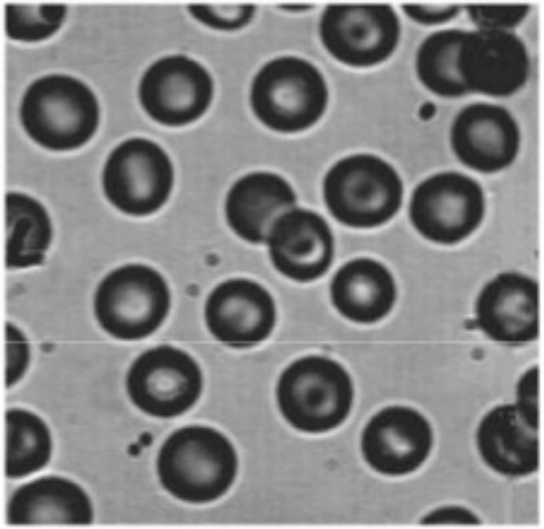
Image Thesholding



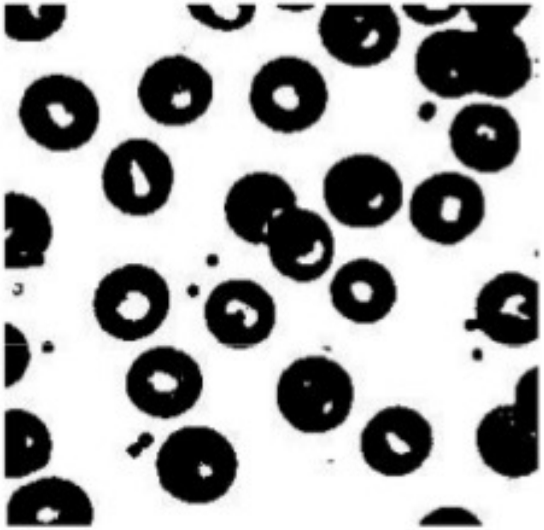
An image



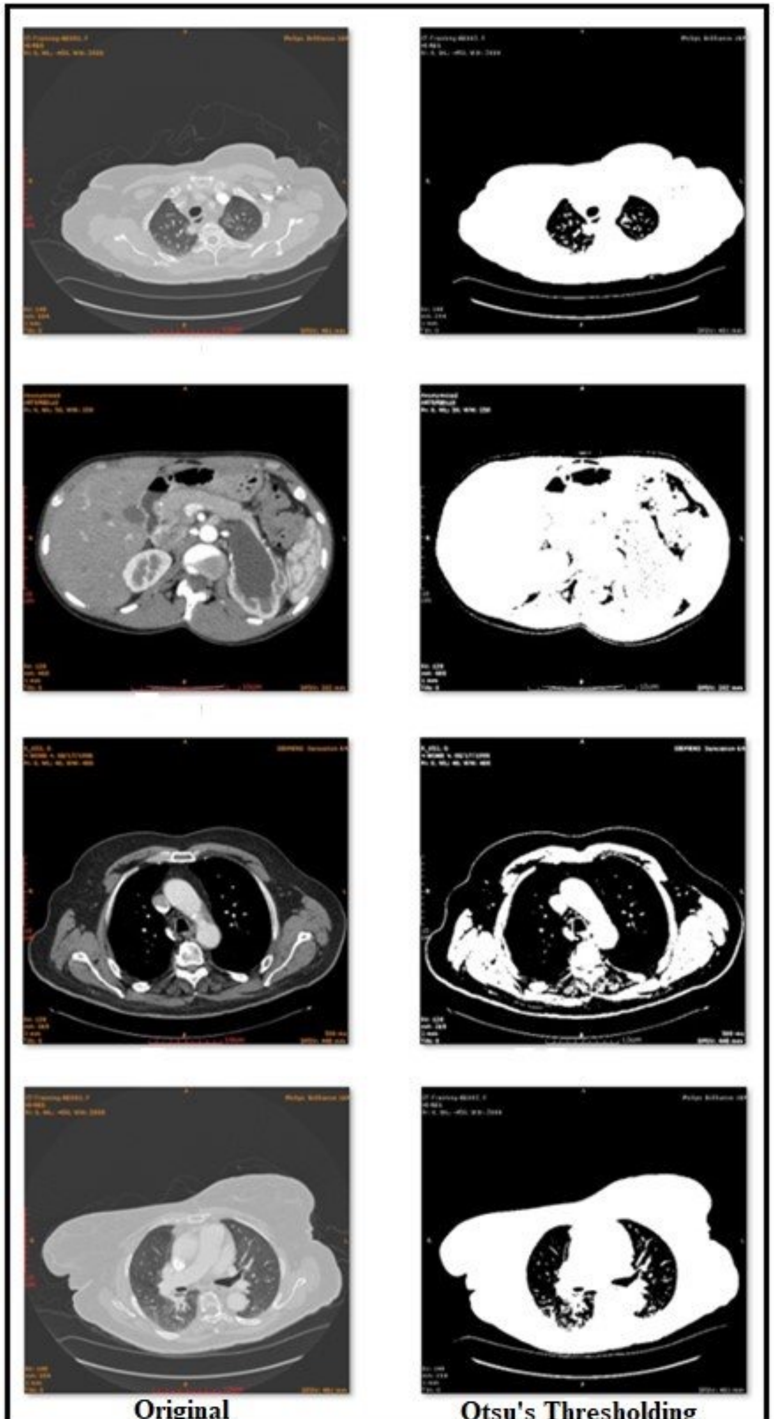
Binary image
by Otsu's method



Red Blood Cells Grayscale Image

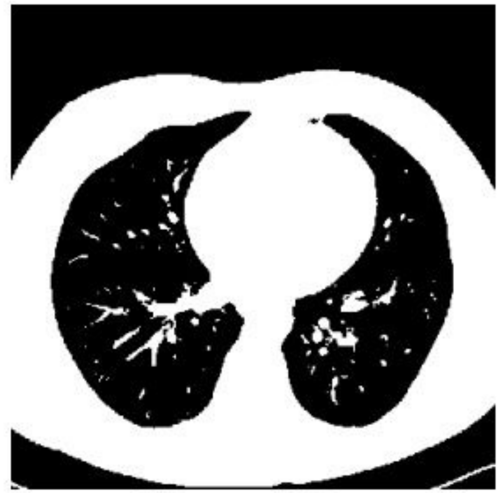


Red Blood Cells Binary Image

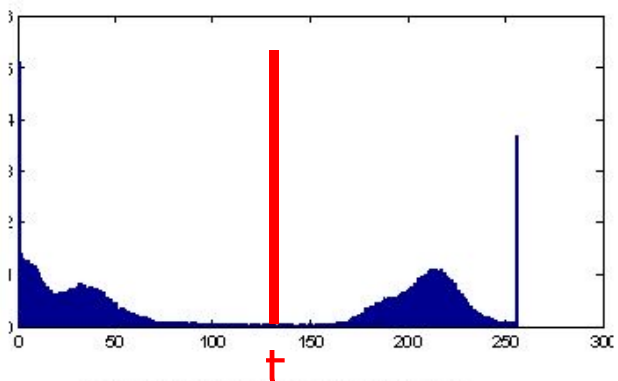

Original
Otsu's Thresholding

Otsu Thresholding

An image



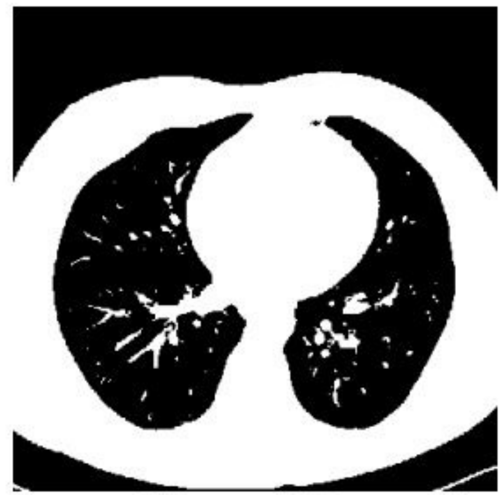
Binary image
by Otsu's method



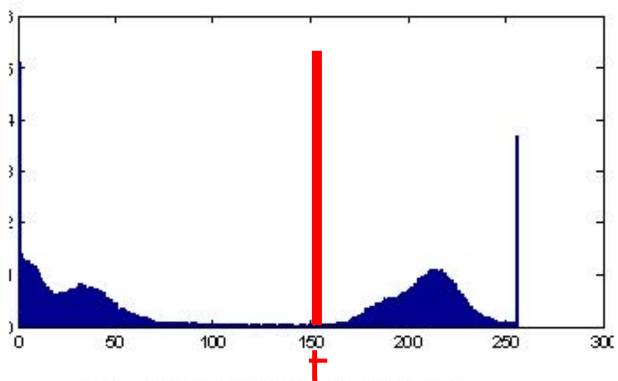
Gray level histogram

Otsu Thresholding

An image



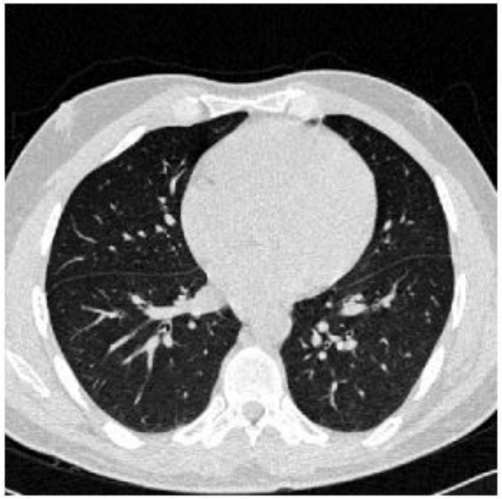
Binary image
by Otsu's method

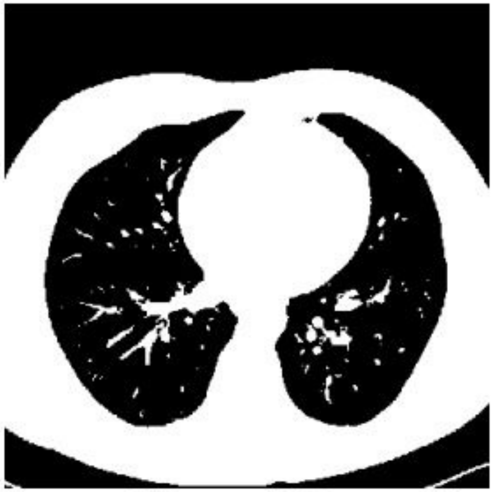


Gray level histogram

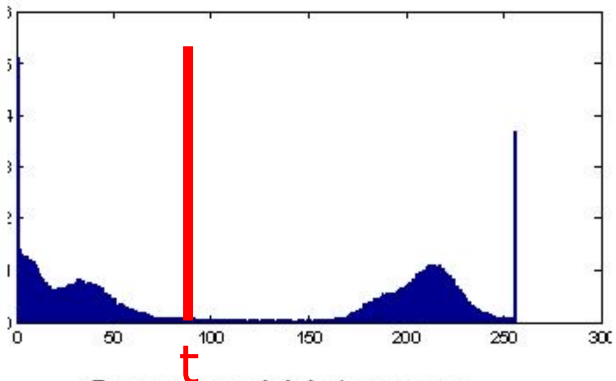
Otsu Thresholding

An image





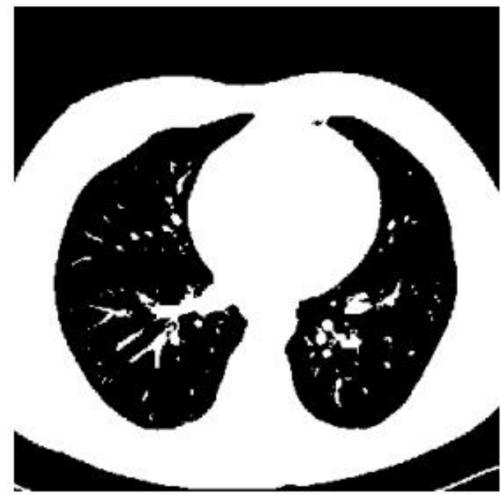
Binary image
by Otsu's method



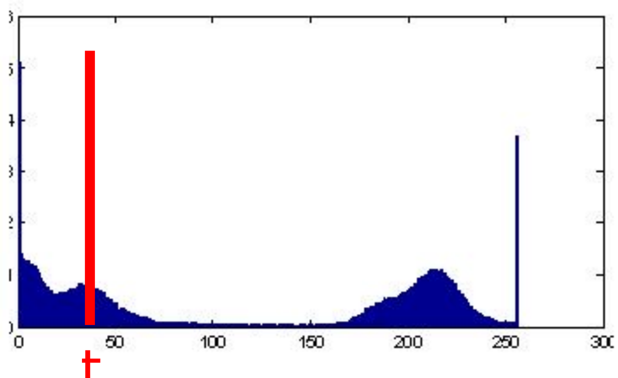
Gray level histogram

Otsu Thresholding

An image



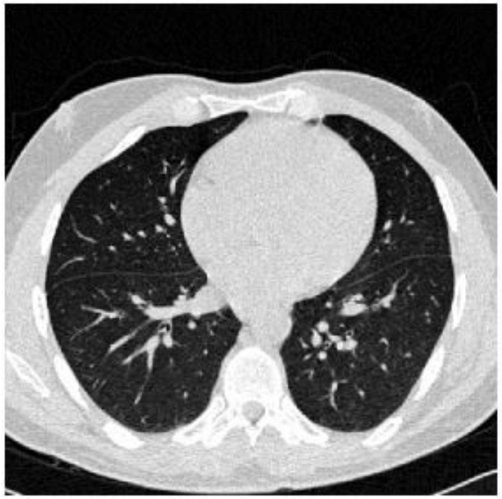
Binary image
by Otsu's method

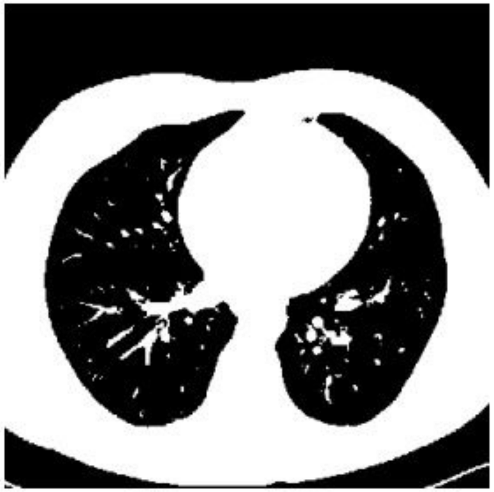


Gray level histogram

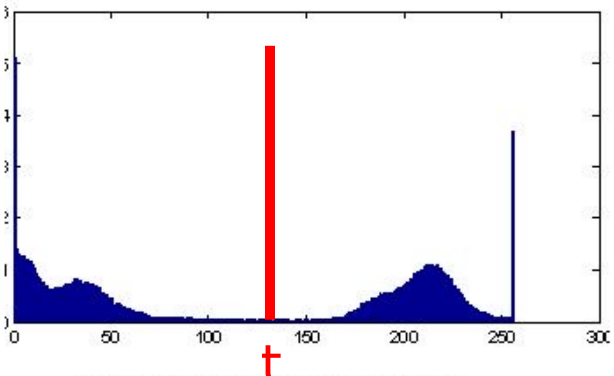
Otsu Thresholding

An image





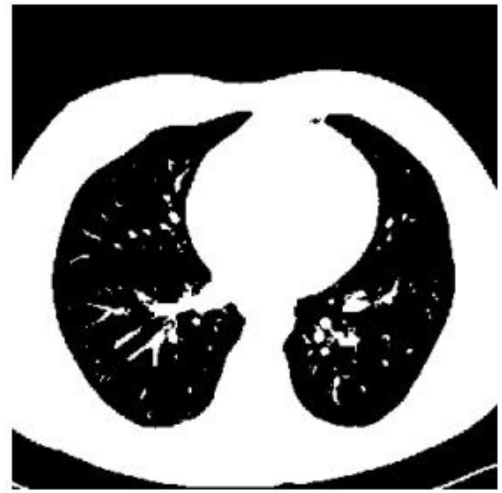
Binary image
by Otsu's method



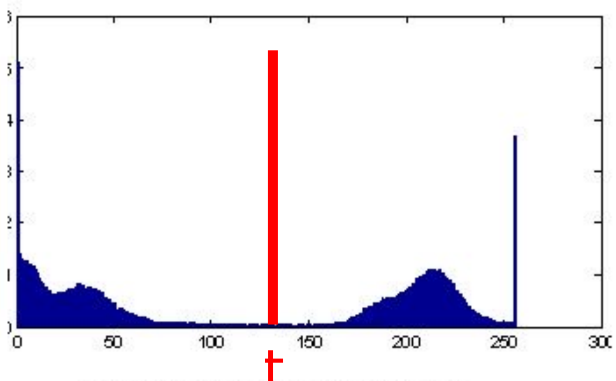
Gray level histogram

Otsu Thresholding

An image

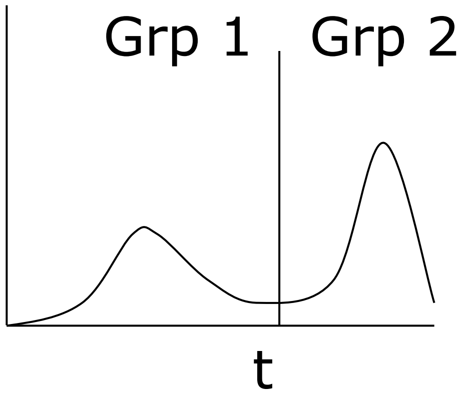


Binary image
by Otsu's method



Gray level histogram

Automatic Thresholding: Otsu's Method



Assumption: the histogram is bimodal

Method: find the threshold t that minimizes the **weighted sum of within-group variances for the two groups** that result from separating the gray tones at value t.

Works well if the assumption holds.

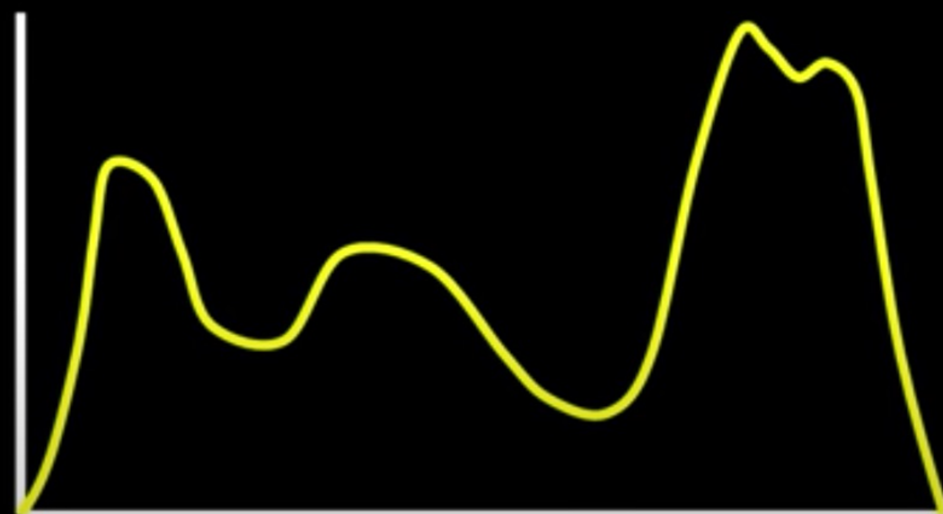
Automatic Thresholding: Otsu's Method

Weighted sum of within-group variances for the two groups

- Weighted
- Within-group
- Variances

Histogram-Directed Thresholding

How can we use a histogram to separate an image into 2 (or several) different regions?



Is there a single clear threshold? 2? 3?

Otsu's method algorithm

- Searches for the threshold intensity I_t which maximizes the *between class variance* σ_B^2

$$\sigma_B^2 = W_b W_f \left(\mu_b - \mu_f\right)^2$$

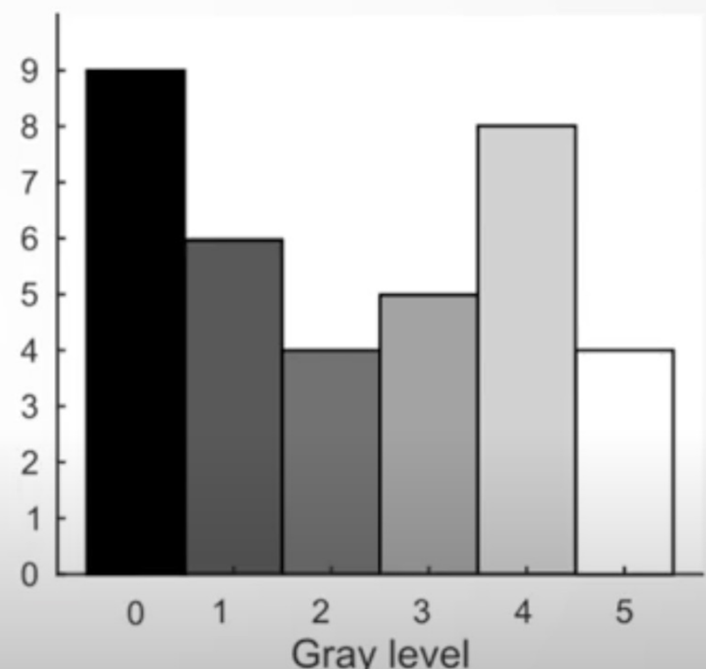
$W_{b,f}$ = Number of pixels in background (foreground)/Total number of pixels

$\mu_{b,f}$ = Mean intensity of background (foreground)

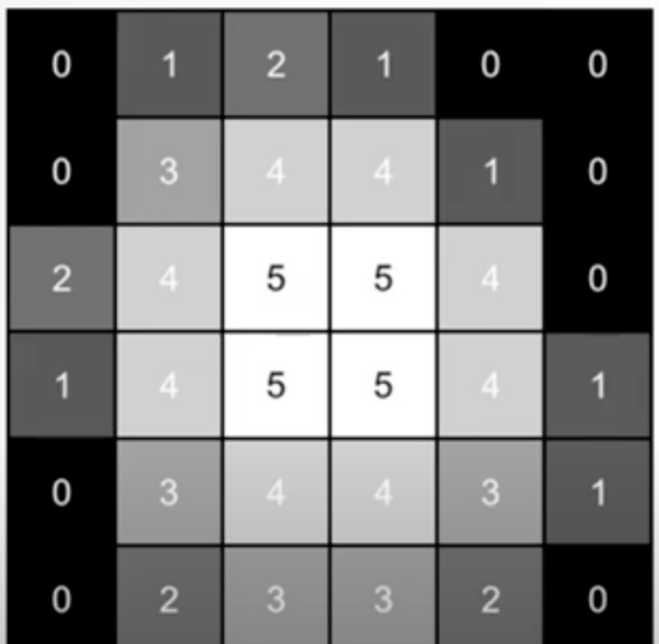
https://www.youtube.com/watch?v=GuqBJ4W7_58&t=78s

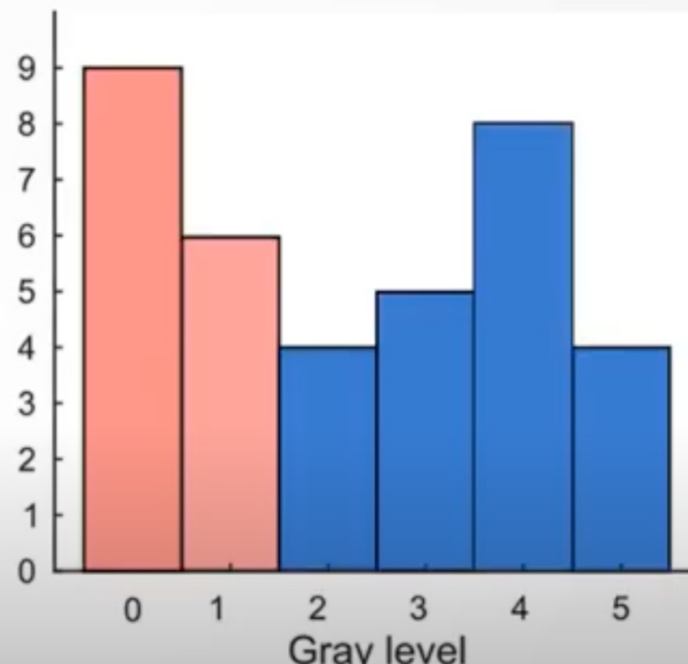
Example

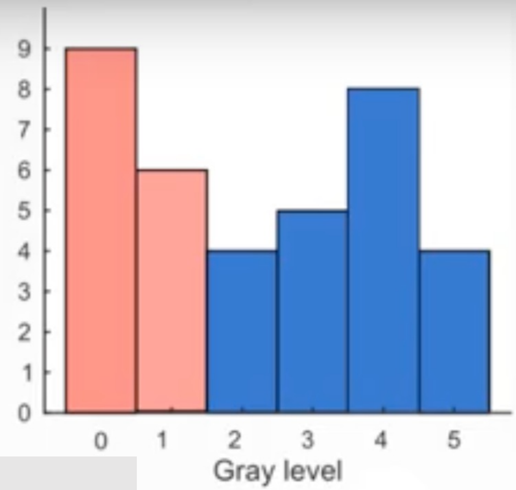
0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

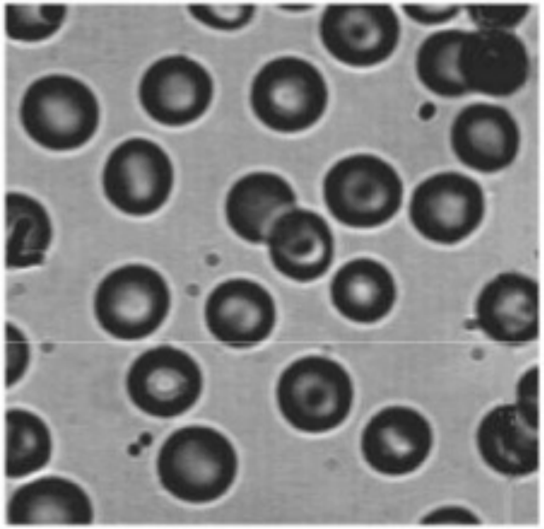


Example

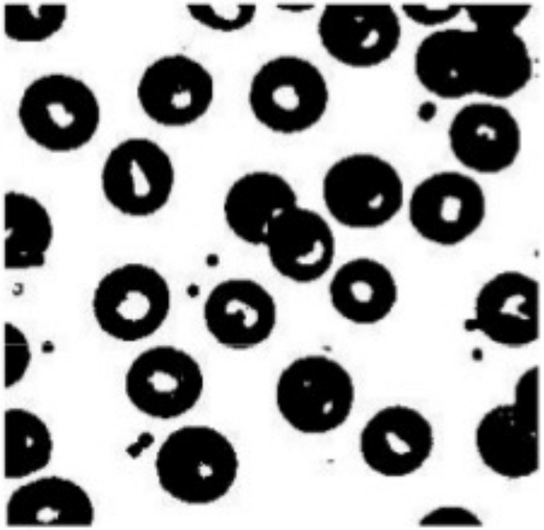

0 1 2 1 0 0
0 3 4 4 1 0
2 4 5 5 4 0
1 4 5 5 4 1
0 3 4 4 3 1
0 2 3 3 2 0


0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5
Gray level


0
1
2
3
4
5
6
7
8
9
Gray level



Red Blood Cells Grayscale Image



Red Blood Cells Binary Image

How to improve (thresholded) Segmentation?

Mathematical Morphology

(Dilation, Erosion, Closing, Opening)

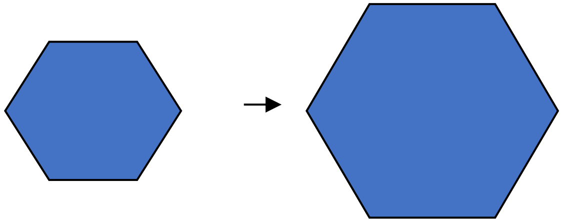
- Dilation

Dilation expands the connected sets of 1s of a binary image.

It can be used for

1. growing features

2. filling holes and gaps



Mathematical Morphology

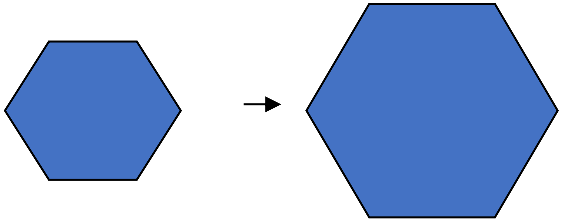
(Dilation, Erosion, Closing, Opening)

- Dilation

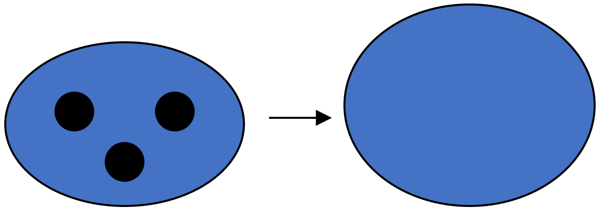
Dilation expands the connected sets of 1s of a binary image.

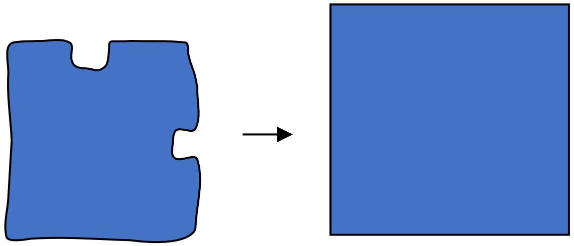
It can be used for

1. growing features



2. filling holes and gaps



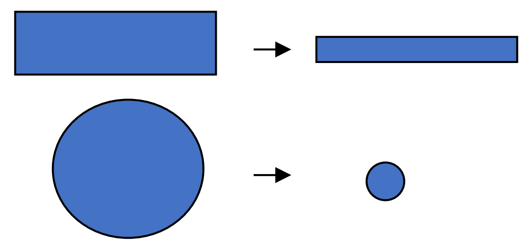


• Erosion

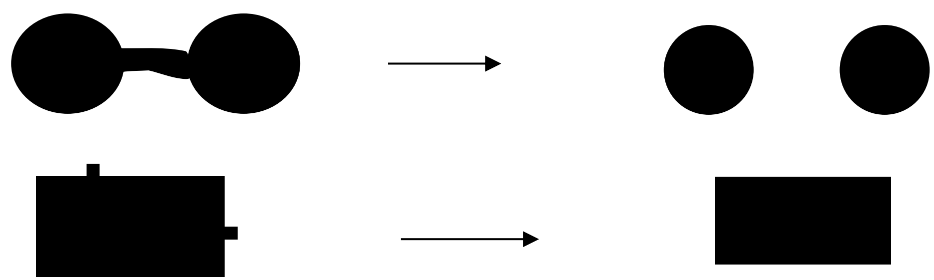
Erosion shrinks the connected sets of 1s of a binary image.

It can be used for

1. shrinking features



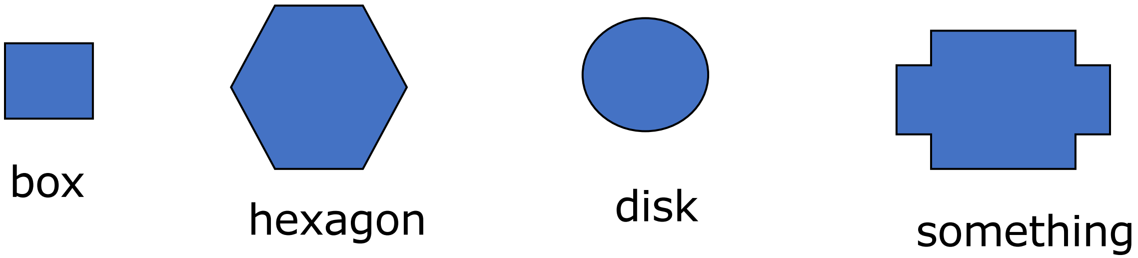
2. Removing bridges, branches and small protrusions



Structuring Elements

A structuring element is a shape mask used in the basic morphological operations.

They can be any shape and size that is digitally representable, and each has an origin.



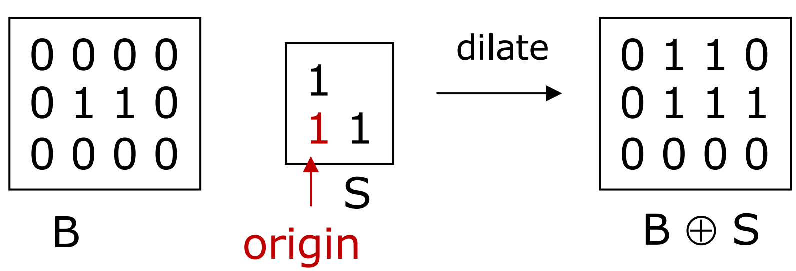
box(length,width) disk(diameter)

Dilation with Structuring Elements

The arguments to dilation and erosion are

1. **a binary image B**
2. **a structuring element S**

dilate(B,S) takes binary image B, places the origin of structuring element S over each 1-pixel, and ORs the structuring element S into the output image at the corresponding position.



Erosion with Structuring Elements

erode(B,S) takes a binary image B, places the origin of structuring element S over every pixel position, and ORs a binary 1 into that position of the output image only if every position of S (with a 1) covers a 1 in B.

origin

$$\underbrace{\begin{array}{|ccccc|}\hline 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}}_{B} \quad \underbrace{\begin{array}{|c|}\hline 1 \\ \color{red}{1} \\ 1 \\ \hline \end{array}}_{S} \quad \xrightarrow{\text{erode}} \quad \underbrace{\begin{array}{|ccccc|}\hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}}_{B \ominus S}$$

Opening and Closing

- Closing is the compound operation of dilation followed by erosion (with the same structuring element)

- Opening is the compound operation of erosion followed by dilation (with the same structuring element)

Original

After Opening

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

a) Binary image B

1	1	1
1	**1**	1
1	1	1

b) Structuring Element S

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1			

c) Dilation $B \oplus S$

				1	1		
				1	1		
				1	1		

d) Erosion $B \ominus S$

	1	1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1				

e) Closing $B \bullet S$

			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	

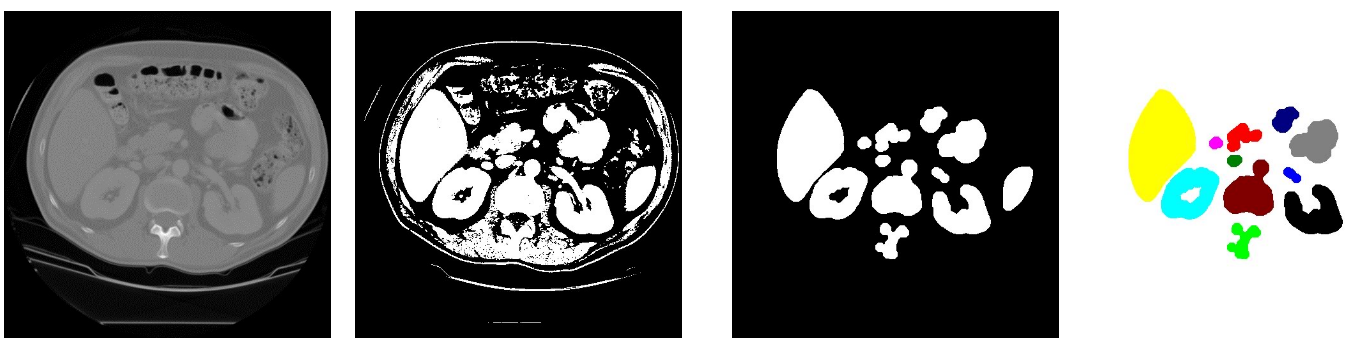
f) Opening $B \circ S$

Connected Components Labeling

Once you have a binary image, you can identify and then analyze each **connected set of pixels**.

The connected components operation takes in a binary image and produces a **labeled image** in which each pixel has the integer label of either the background (0) or a component.

original thresholded opening+closing components



Methods for CC Analysis

1. Recursive Tracking (almost never used)

2. Parallel Growing (needs parallel hardware)

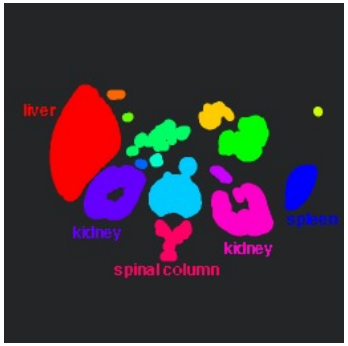
3. Row-by-Row (most common)

   a. propagate labels down to the bottom, recording equivalences

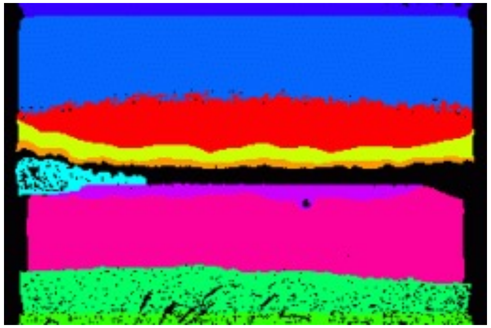
   b. Compute equivalence classes

   c. Replace each labeled pixel with the label of its equivalence class.

Labelings shown as Pseudo-Color



connected components of 1's from cleaned, thresholded image



connected components of cluster labels

References and Slide Credits

- Jayaram K. Udupa, MIPG of University of Pennsylvania, PA.
- P. Suetens, Fundamentals of Medical Imaging, Cambridge Univ. Press.
- N. Bryan, Intro. to the science of medical imaging, Cambridge Univ. Press.
- CAP 5415 Computer Vision (Fall 2016) Lecture Presentations
- Computer Vision (Lecture Presentations) by Dr. Mohsen Ali