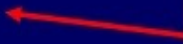
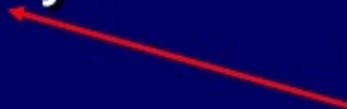


**Digital Image Processing**  
**Lecture 16: Segmentation:**  
**Detection of Discontinuities**  
**May 2, 2005**

**Prof. Charlene Tsai**

# What is segmentation?

- Dividing an image into its constituent regions or objects.
- Heavily rely on one of two properties of intensity values:
  - Discontinuity  Partition based on abrupt changes in intensity, e.g. edges in an image
  - Similarity  Partition based on intensity similarity, e.g. thresholding

We'll discuss both approaches.  
Starting with the first one.

- We want to extract 2 basic types of gray-level discontinuity:
  - Lines
  - Edges
- What have we learnt in previous lectures to help us in this process?
  - CONVOLUTION!

$$g(x, y) = \sum_{s=a}^b \sum_{t=a}^b w(s, t) f(x+s, y+t)$$

Diagram illustrating the convolution operation:

- The term  $f(x+s, y+t)$  is labeled "Grayscale image".
- The term  $w(s, t)$  is labeled "Mask coefficient".

# Line Detection

- Masks for lines of different directions:

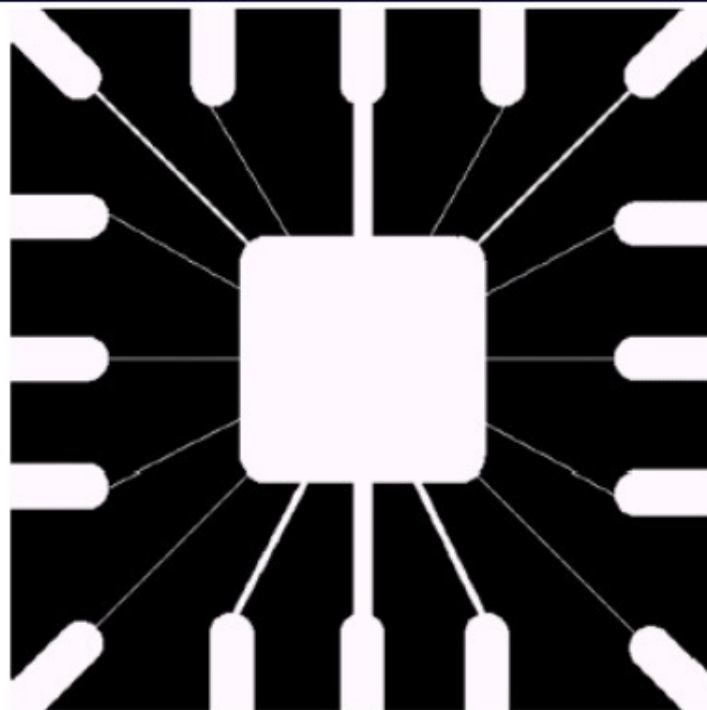
**FIGURE 10.3** Line masks.

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal			+45°			Vertical			-45°		

- Respond more strongly to lines of one pixel thick of the designated direction.
- High or low pass filters?



- If interested in lines of any directions, run all 4 masks and select the highest response
- If interested only in lines of a specific direction (e.g. vertical), use only the mask associated with that direction.
- Threshold the output.
- The strongest responses for lines one pixel thick, and correspond closest to the direction defined by the mask.



a  
b c

#### FIGURE 10.4

Illustration of line detection.

(a) Binary wire-bond mask.

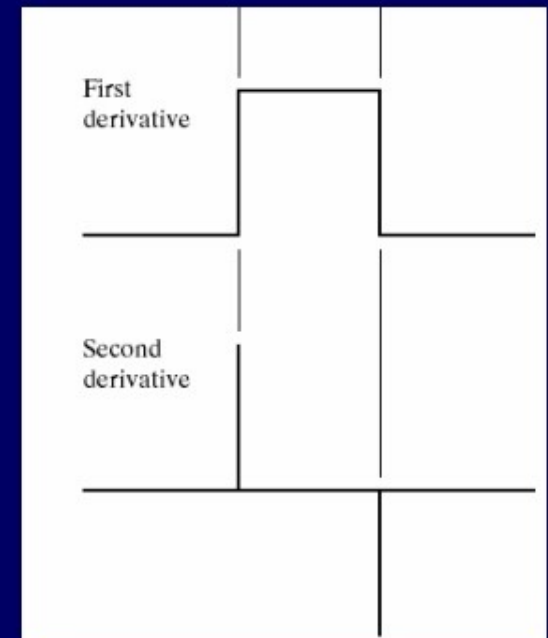
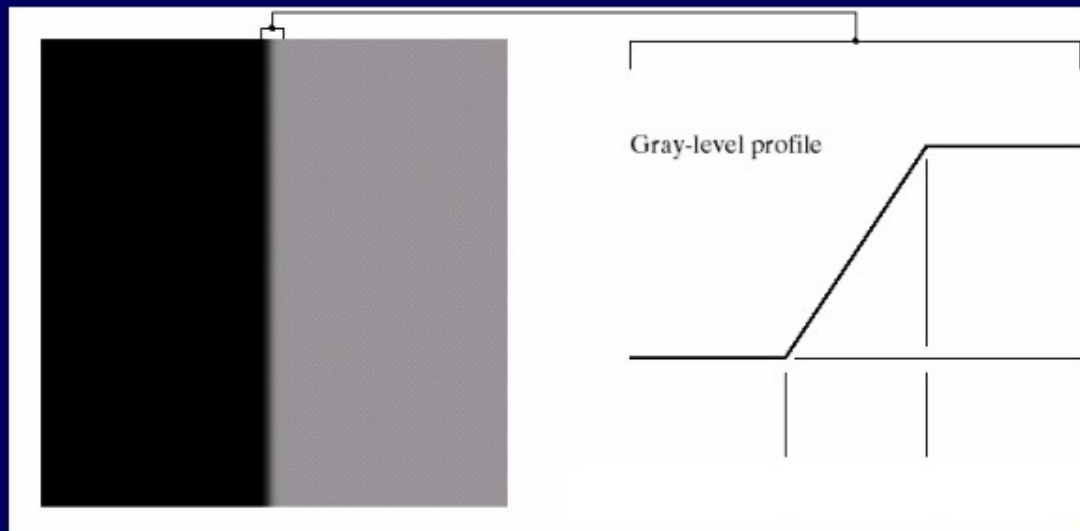
(b) Absolute value of result after processing with  $-45^\circ$  line detector.

(c) Result of thresholding image (b).



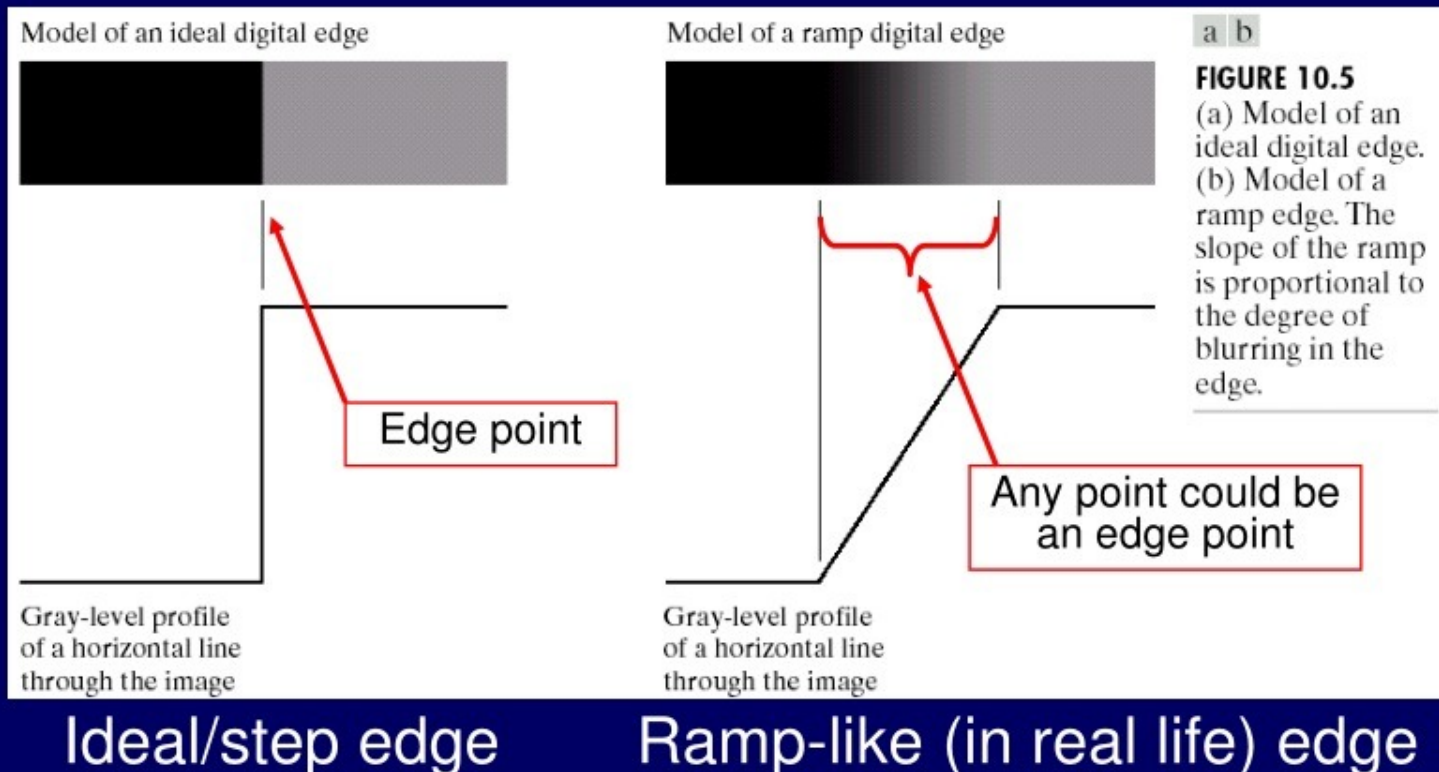
# Edge Detection

- Far more practical than line detection.
- We'll discuss approaches based on
  - 1<sup>st</sup>-order digital derivative
  - 2<sup>nd</sup>-order digital derivative



# What is an edge?

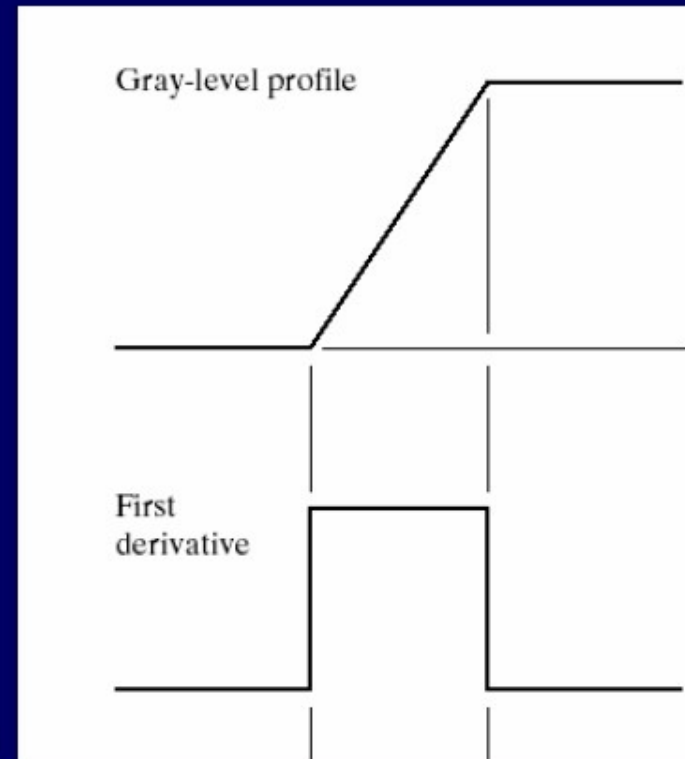
- A set of connected pixels that lie on the boundary between two regions.
- Local concept





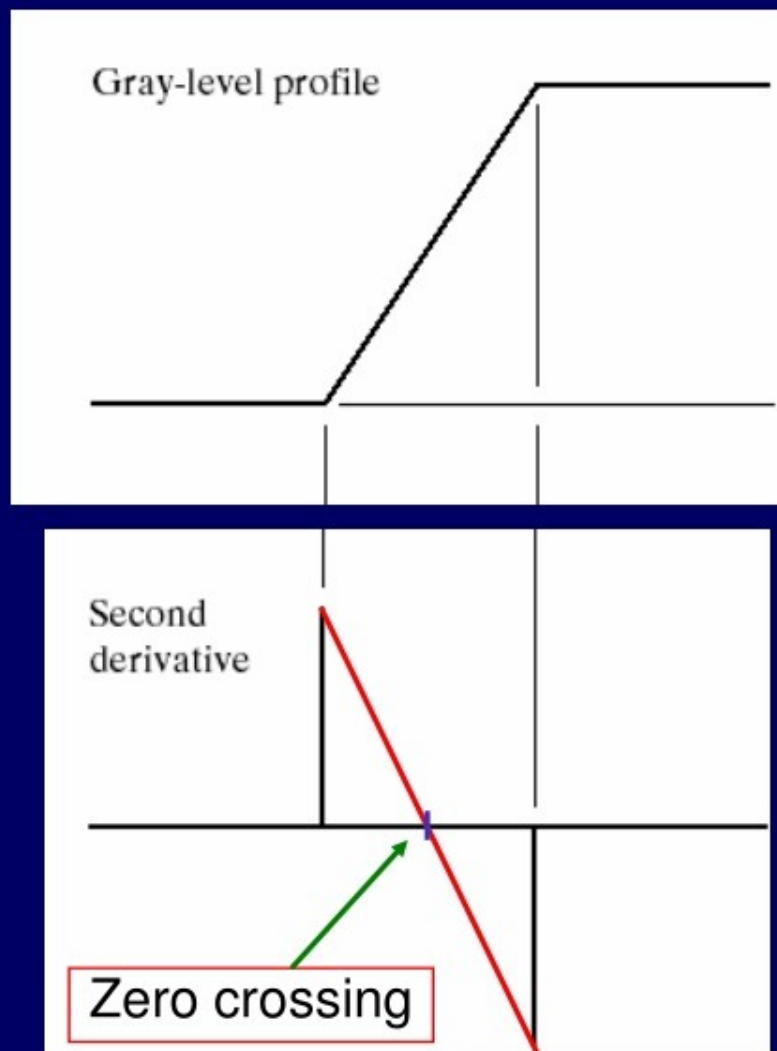
# 1<sup>st</sup> Derivative

- Positive at the points of transition into and out of the ramp, moving from left to right along the profile
- Constant for points in the ramp
- Zero in areas of constant gray Level
- Magnitude for presence of an edge at a point in an image (i.e. if a point is on a ramp)



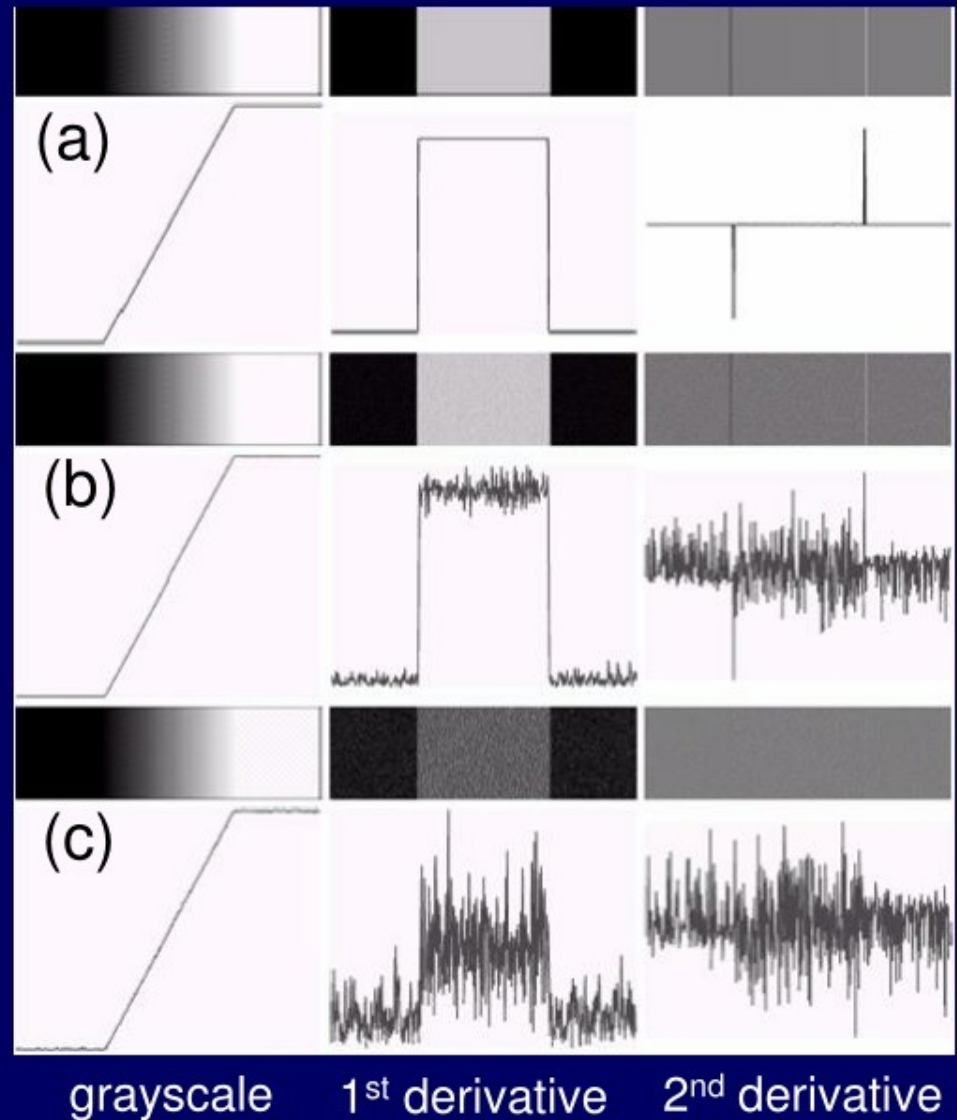
## 2<sup>nd</sup> Derivative

- Positive at the transition associated with the dark side of the edge
- Negative at the transition associated with the bright side of the edge
- Zero elsewhere
- Producing 2 values for every edge in an image.
- Center of a thick edge is located at the zero crossing



# Effect of Noise

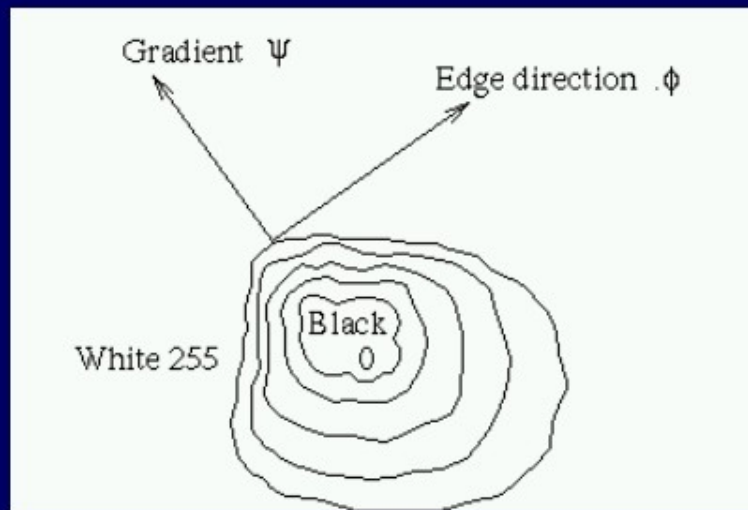
- ❑ Corrupted by Random Gaussian noise of mean 0 and standard deviation of
- ❑ (a) 0
- ❑ (b) 0.1
- ❑ (c) 1.0
- ❑ Conclusion???
- ❑ Sensitivity of derivative to noise





# Some Terminology

- An edge element is associated with 2 components:
  - magnitude of the gradient, and
  - and edge direction  $\phi$  , rotated with respect to the gradient direction  $\psi$  by -90 deg.





# 1<sup>st</sup> Derivative (Gradient)

- The gradient of an image  $f(x,y)$  at location  $(x,y)$  is defined as

$$\begin{aligned} \nabla f &= \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} \\ \text{mag} \nabla f &= \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \\ \angle \nabla f &= \arctan \left( \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right) \end{aligned}$$

# Finite Gradient - Approximation

- Central differences (not usually used because they neglect the impact of the pixel (x,y) itself)

$$\nabla_x f(x, y) \approx \frac{f(x+h, y) - f(x-h, y)}{2h}$$

$$\nabla_y f(x, y) \approx \frac{f(x, y+h) - f(x, y-h)}{2h}$$

- h** is a small integer, usually 1.
- h** should be chosen small enough to provide a good approximation to the derivative, but large enough to neglect unimportant changes in the image function.

# Convolution of Gradient Operators



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

Prewitt

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Sobel



Better noise-suppression



# Illustration

- One simple method for approximating zero-crossing:
  - Setting all + values to white, - values to black.
  - Scanning the thresholded image and noting the transition between black and white.



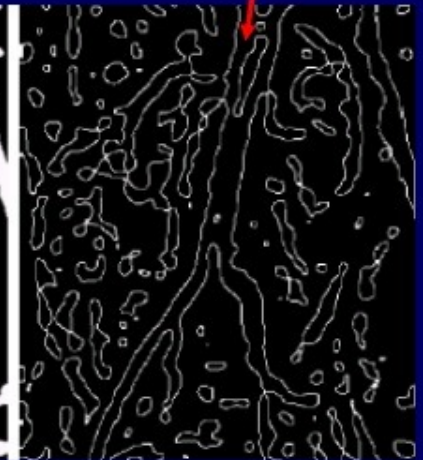
original



LoG



thresholded



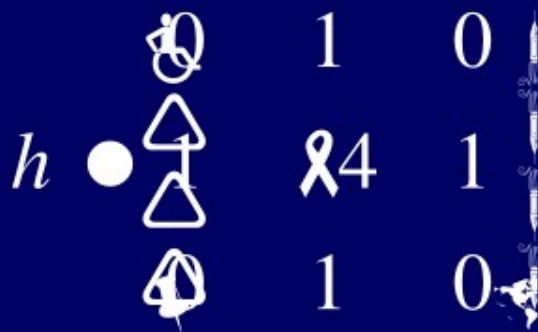
Closed loops  
(spaghetti effect)

zero crossing

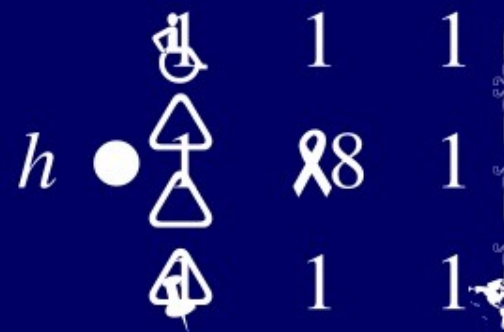


## 2<sup>nd</sup> Derivative: Laplacian Operator

- Review: The Laplace operator ( $\nabla^2$ ) is a very popular operator approximating the second derivative which gives the gradient magnitude only.
- We discussed this operator in lecture 5 (spatial filtering)
- It is isotropic



4-neighborhood



8-neighborhood

# Issues with Laplacian

- Problems:
  - Unacceptably sensitive to noise
  - Magnitude of Laplacian results in double edges
  - Does not provide gradient
- Fixes:
  - Smoothing
  - Using zero-crossing for edge location
  - Not for gradient direction, but for establishing whether a pixel is on the dark or light side of an edge

# Smoothing for Laplacian

- Our goal is to get a second derivative of a smoothed 2D function  $f(x, y)$
- We have seen that the Laplacian operator gives the second derivative, and is non-directional (isotropic).
- Consider then the Laplacian of an image  $f(x, y)$  smoothed by a Gaussian.
- This operator is abbreviated as **LoG**, from **Laplacian of Gaussian**:  
$$\nabla^2 h(x, y) * f(x, y)$$
- The order of differentiation and convolution can be interchanged due to linearity of the operations:

$$\nabla^2 h(x, y) * f(x, y)$$



# Laplacian of Gaussian (LoG)

- Gaussian smoothing effectively suppresses the influence of the pixels that are up to a distance  $3\sigma$  from the current pixel; then the Laplace operator is an efficient and stable measure of changes in the image.
- The location in the LoG image where the zero level is crossed corresponds to the position of the edges.
- The advantage of this approach compared to classical edge operators of small size is that a larger area surrounding the current pixel is taken into account; the influence of more distant points decreases according to  $\frac{1}{r^3}$



# Laplacian of Gaussian (LoG)

- Gaussian smoothing effectively suppresses the influence of the pixels that are up to a distance  $3\sigma$  from the current pixel; then the Laplace operator is an efficient and stable measure of changes in the image.
- The location in the LoG image where the zero level is crossed corresponds to the position of the edges.
- The advantage of this approach compared to classical edge operators of small size is that a larger area surrounding the current pixel is taken into account; the influence of more distant points decreases according to  $\frac{1}{r^3}$

# Laplacian of Gaussian (LoG)

- Gaussian smoothing effectively suppresses the influence of the pixels that are up to a distance  $3\sigma$  from the current pixel; then the Laplace operator is an efficient and stable measure of changes in the image.
- The location in the LoG image where the zero level is crossed corresponds to the position of the edges.
- The advantage of this approach compared to classical edge operators of small size is that a larger area surrounding the current pixel is taken into account; the influence of more distant points decreases according to  $\frac{1}{r^3}$



- ❑ LoG kernel become large for larger  $\sigma$  (e.g. 40x40 for  $\sigma = 4$ )
- ❑ Fortunately, there is a separable decomposition of the LoG operator that can speed up computation considerable.
- ❑ The practical implication of Gaussian smoothing is that edges are found reliably.
- ❑ If only globally significant edges are required, the standard deviation of the Gaussian smoothing filter may be increased, having the effect of suppressing less significant evidence.
- ❑ Note that since the LoG is an isotropic filter, it is not possible to directly extract edge orientation information from the LoG output, unlike Roberts and Sobel operators.

# Illustration

- One simple method for approximating zero-crossing:
  - Setting all + values to white, - values to black.
  - Scanning the thresholded image and noting the transition between black and white.



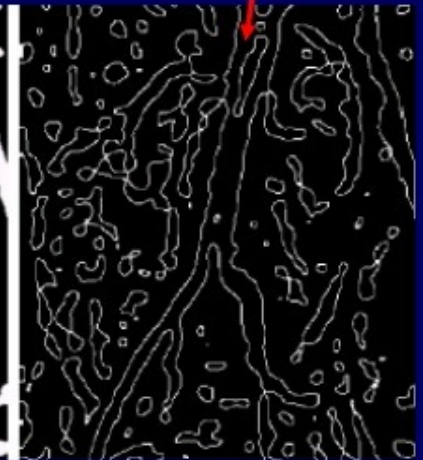
original



LoG



thresholded



Closed loops  
(spaghetti effect)

zero crossing



# LoG variant - DoG

- It is possible to approximate the LoG filter with a filter that is just the difference of two differently sized Gaussians. Such a filter is known as a *DoG* filter (short for 'Difference of Gaussians').
- Similar to Laplace of Gaussian, the image is first smoothed by convolution with Gaussian kernel of certain width  $\sigma_1$

$$g_1(x, y) = G_{\sigma_1}(x, y) * f(x, y)$$

- With a different width  $\sigma_2$ , a second smoothed image can be obtained:

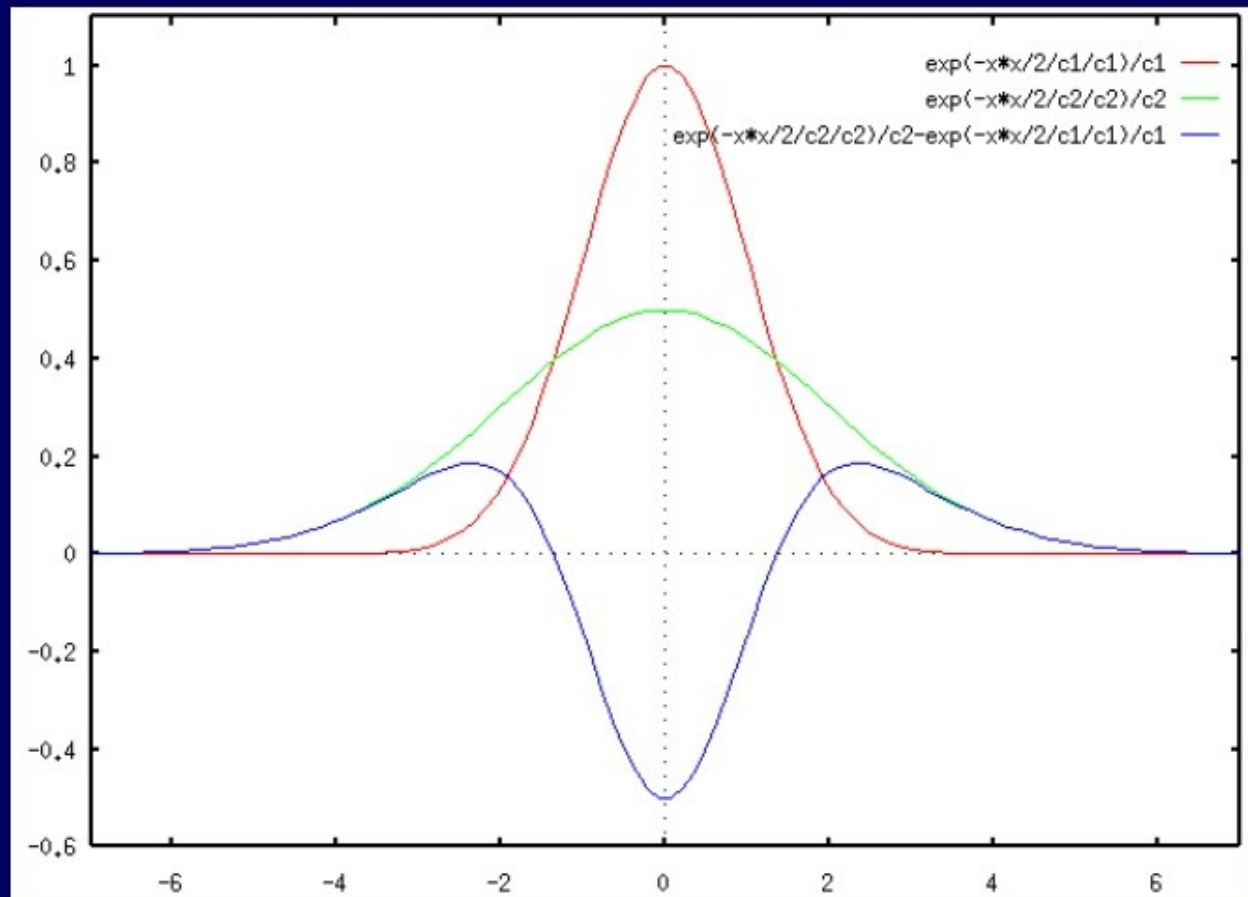
$$g_2(x, y) = G_{\sigma_2}(x, y) * f(x, y)$$

- The difference of these two Gaussian smoothed images, called **difference of Gaussian** (*DoG*), can be used to detect edges in the image.

$$\begin{aligned}
 &g_1(x, y) - g_2(x, y) \\
 &= G_{\sigma_1}(x, y) * f(x, y) - G_{\sigma_2}(x, y) * f(x, y) \\
 &= (G_{\sigma_1} - G_{\sigma_2}) * f(x, y) = DoG * f(x, y)
 \end{aligned}$$

- The DoG as an operator or convolution kernel is defined as

$$DoG = G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma_1} e^{-\frac{x^2 + y^2}{2\sigma_1^2}} - \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma_2} e^{-\frac{x^2 + y^2}{2\sigma_2^2}}$$



- The discrete convolution kernel for DoG can be obtained by approximating the continuous expression of DoG given above. Again, it is necessary for the sum or average of all elements of the kernel matrix to be zero.



- Line detection
- Edge detection based on
  - First derivative
    - Provides gradient information
  - 2<sup>nd</sup> derivative using zero-crossing
    - Indicates dark/bright side of an edge