

# Spatial Filtering

*Signal and Image processing*

*Lecture 5*

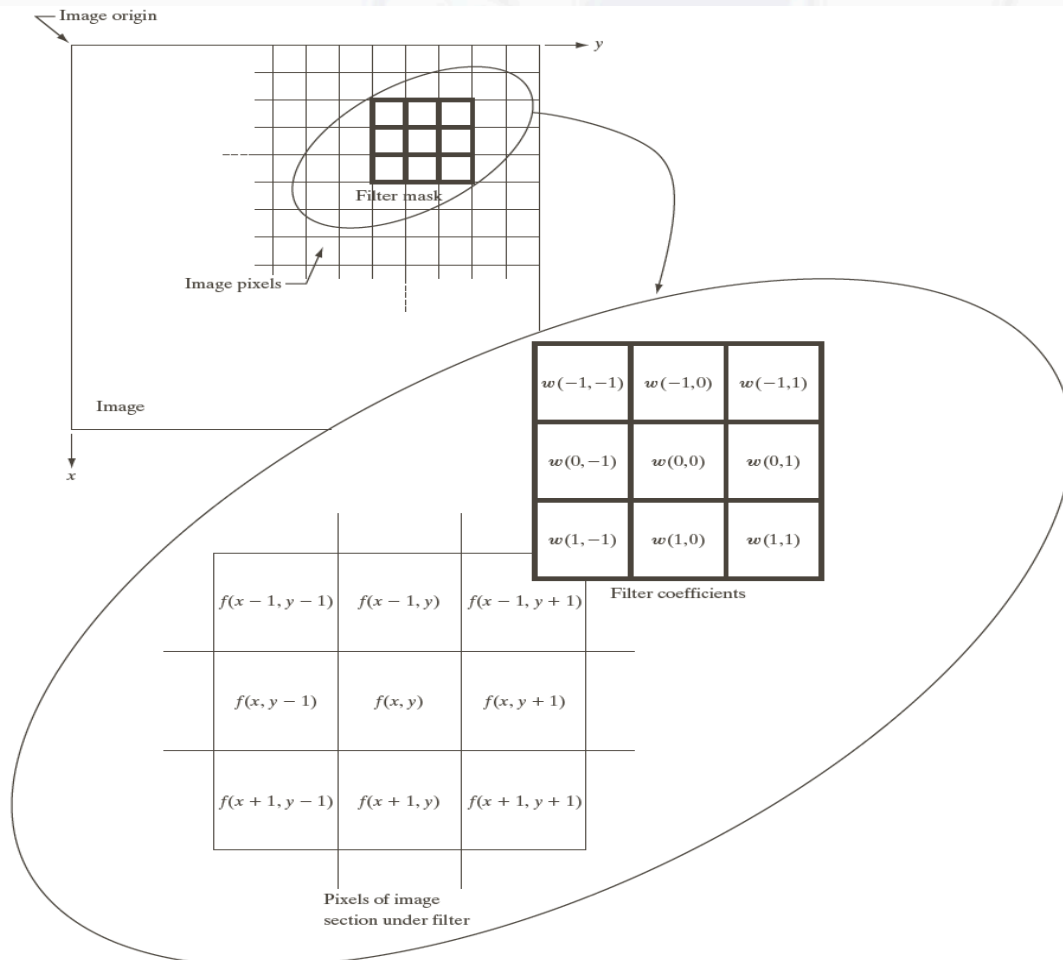


# Basics of Spatial Filtering

- ◆ Spatial filter consists of
  - ◆ Neighborhood (small rectangle)
  - ◆ Predefined operation
    - ◆ Performed on the image pixels encompassed by the neighborhood
- ◆ New pixel is created
  - ◆ coordinates equal to the coordinates of the center of the neighborhood
  - ◆ value is the result of the operation
- ◆ Linear if operation is linear



# Basics of Spatial Filtering



**FIGURE 3.28** The mechanics of linear spatial filtering using a  $3 \times 3$  filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

# Basics of Spatial Filtering

$$g(x, y) = w(-1, -1) f(x-1, y-1) + w(-1, 0) f(x-1, y) + \dots \\ + w(0, 0) f(x, y) + \dots + w(0, 1) f(x, y+1) + w(1, 1) f(x+1, y+1)$$

◆  $m \times n$  mask

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

where,

$$a = (m-1)/2$$
$$b = (n-1)/2$$



# Spatial Correlation and Convolution

## ◆ Correlation

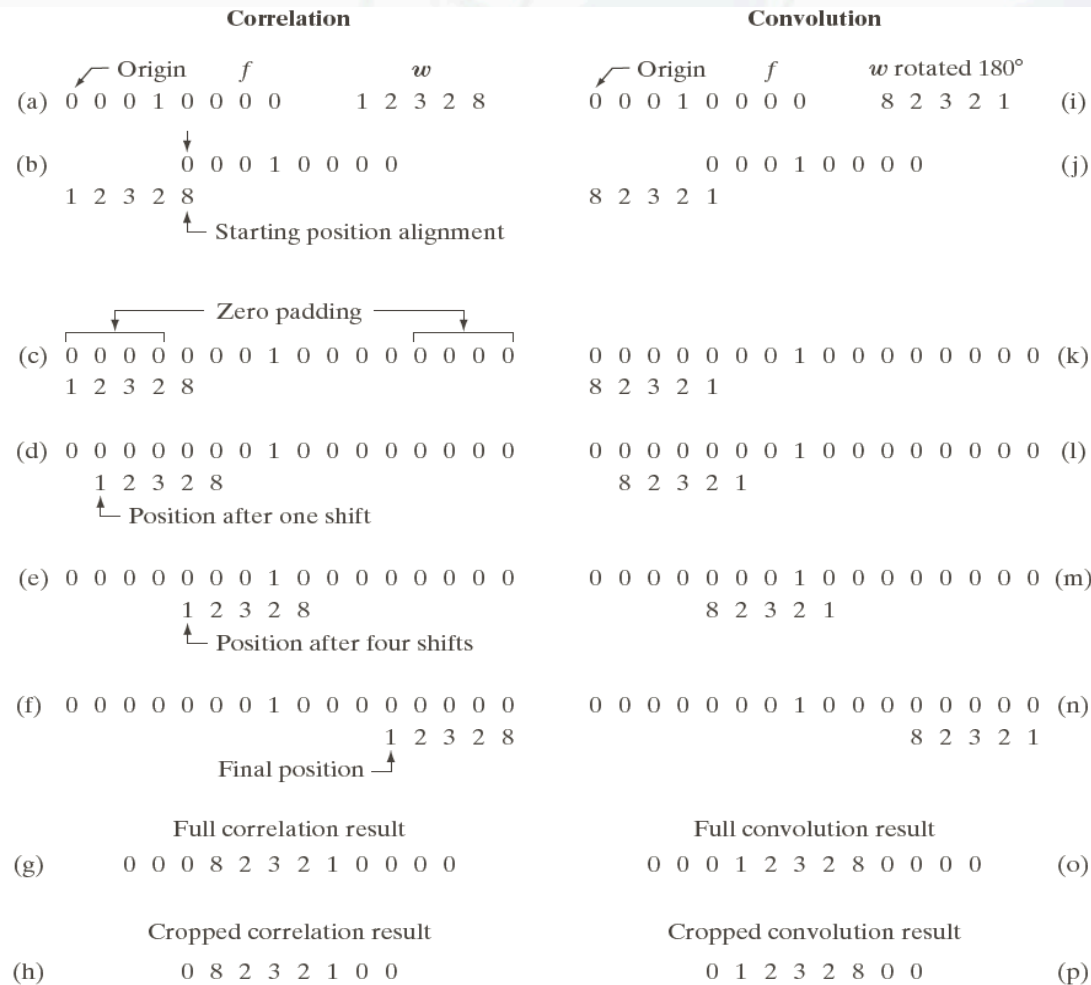
- ◆ Moving a filter mask over the image and computing the sum of products at each location
- ◆ Is a function of displacement of the filter
  - ◆ First value corresponds to 0 displacement of the filter.
- ◆ Correlating a filter  $w$  with a discrete unit impulse function yields a copy of  $w$ , but rotated with 180 degree.
  - ◆ Discrete unit impulse Function contains a single 1 with the rest being 0s

## ◆ Convolution

- ◆ Same as correlation, except that the filter is first rotated by 180 degree.



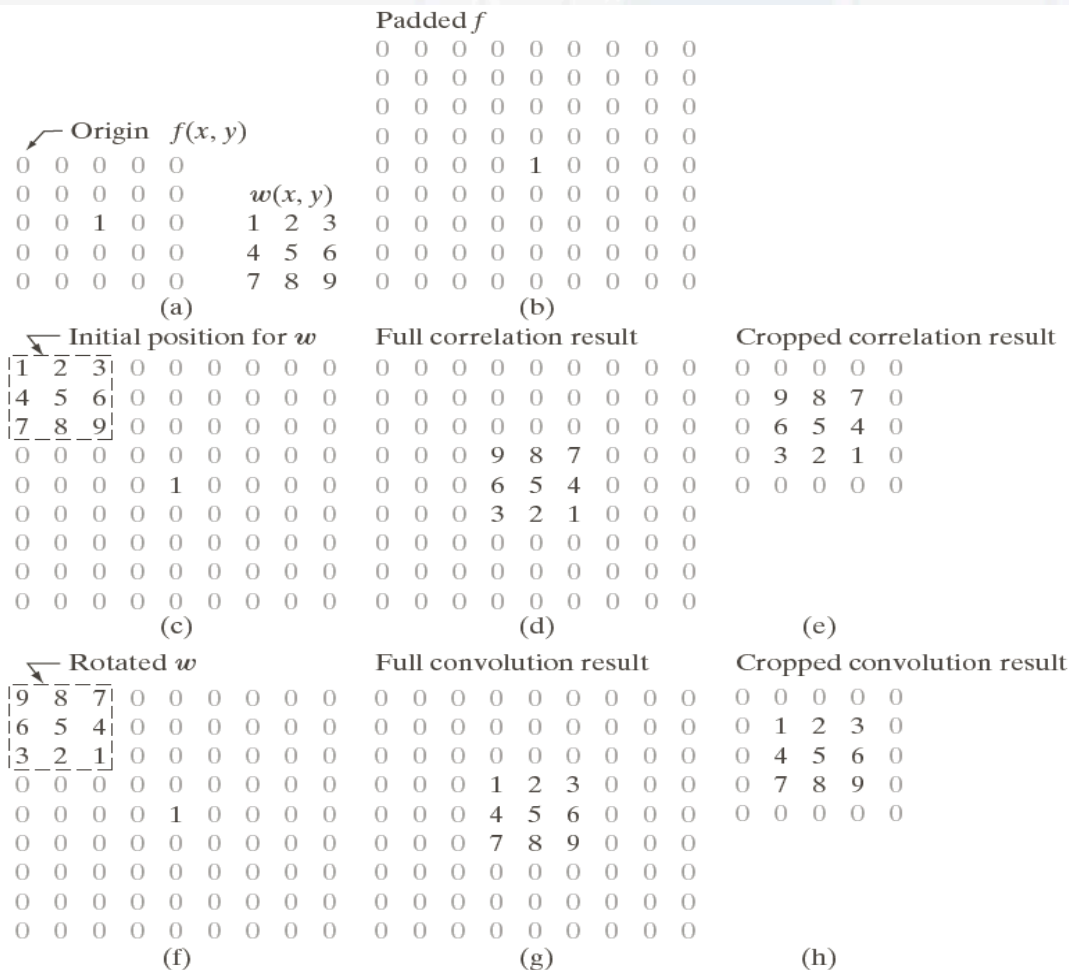
# Spatial Correlation and Convolution



**FIGURE 3.29** Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.



# Spatial Correlation and Convolution



**FIGURE 3.30** Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.



# Spatial Correlation and Convolution

- ◆ Correlation of a filter  $w(x,y)$  of size  $m \times n$  with an image  $f(x,y)$  is given by

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

- ◆ Evaluated for all values so that all elements of  $w$  visit every pixel in  $f$ .
- ◆ Convolution of  $w(x,y)$  and  $f(x,y)$  is given by

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





# Vector Representation of Linear Filtering

$$R = w_1 z_1 + w_2 z_2 + \dots + w_{mn} z_{mn}$$

$$= \sum_{k=1}^{mn} w_k z_k$$

$$= \mathbf{w}^T \mathbf{z}$$

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9$$

$$= \sum_{k=1}^9 w_k z_k$$

$$= \mathbf{w}^T \mathbf{z}$$

- ◆ Generating spatial filter masks
  - ◆ Specify  $mn$  mask coefficients
  - ◆ Selection is based on filter function

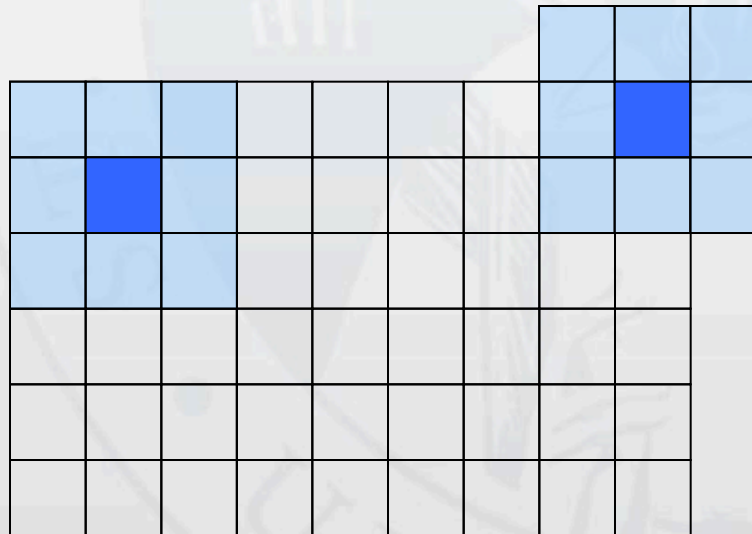
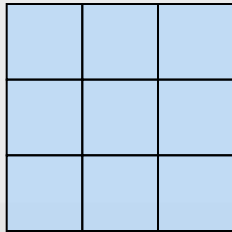


# Basics of Spatial Filtering

## ◆ Boundary Implementation

◆ No processing

◆ Padding



# Smoothing Spatial Filters

- ◆ Smoothing Filters are used for **blurring** and for **noise reduction** (noise typically has sharp transitions in intensity)
- ◆ Blurring is used for
  - ◆ removal of small details prior to object extraction.
  - ◆ bridging of small gaps in lines or curves.
- ◆ Smoothing Linear Filters (Averaging Filters)
  - ◆ low pass filter
  - ◆ replace the average value defined by the filter mask.
  - ◆ have the undesirable effect of blur edges

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$



# Smoothing Spatial Filters

- ◆ weighted average
- ◆ Pixels have different importance
- ◆ The pixel at the center is multiplied by a higher value
- ◆ The other pixels are inversely weighted as a function of distance
- ◆ The weight like this attempts to reduce the blurring effect in the smoothing process
- ◆ Sum of all weights is 16 (easy for computer implementation because it is a power of 2)
- ◆ Other values of weight could accomplish the same general objectives



# Smoothing Spatial Filters

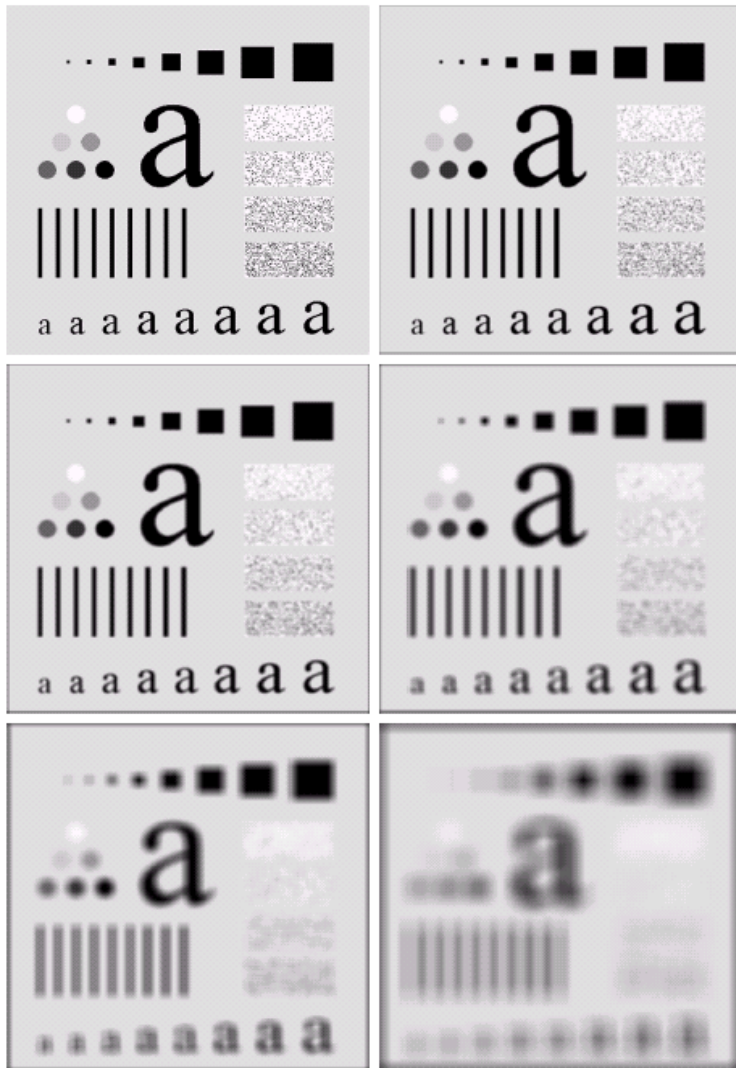
- ◆ The general implementation for filtering an MxN image with weighted averaging filter size m x n is

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



# Smoothing Spatial Filters

- ◆ Excessive blurring is generally used for eliminating small objects from an image

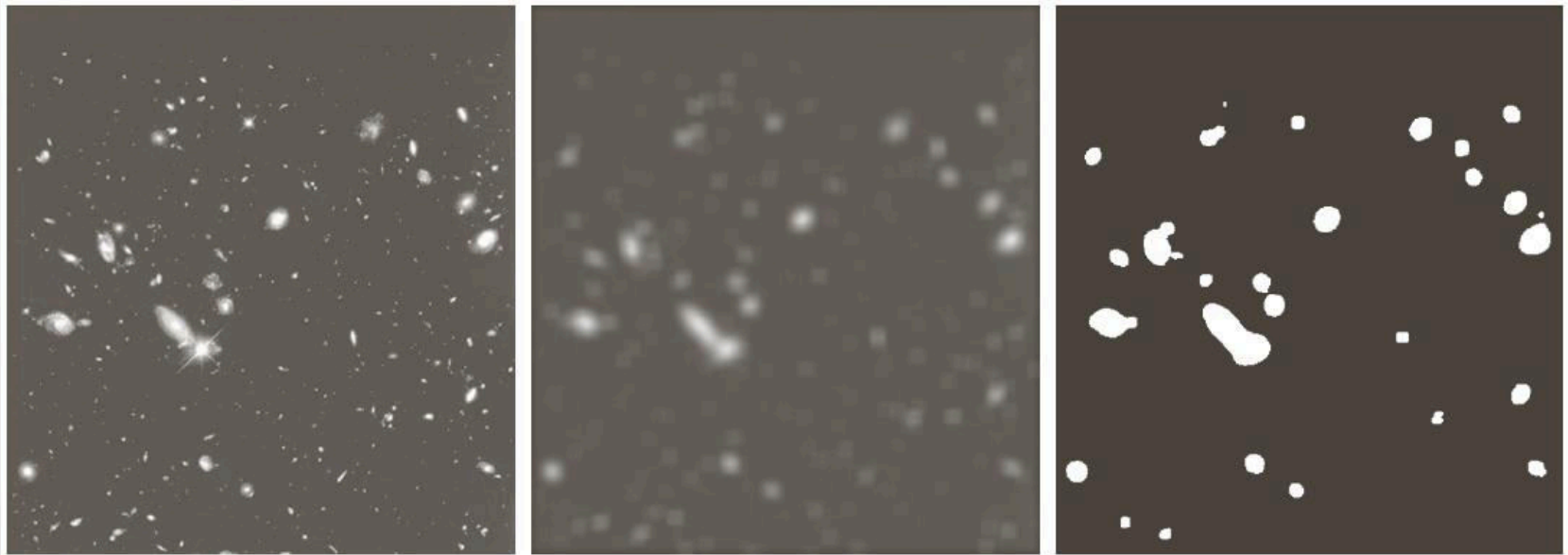


**FIGURE 3.35** (a) Original image, of size  $500 \times 500$  pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes  $n = 3, 5, 9, 15,$  and  $35$ , respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size  $50 \times 120$  pixels.





# Smoothing Spatial Filters



a b c

**FIGURE 3.34** (a) Image of size  $528 \times 485$  pixels from the Hubble Space Telescope. (b) Image filtered with a  $15 \times 15$  averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

# Smoothing Spatial Filters

- **Filter**

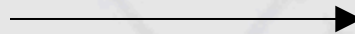
$\frac{1}{6}$	0	1	0
	1	2	1
	0	1	0

- **Input image: A 4x4, 4 bits/pixel**

1	8	6	6
6	3	11	8
8	8	9	10
9	10	10	7

- **Preprocessing: Zero-padding**

1	8	6	6
6	3	11	8
8	8	9	10
9	10	10	7



0	0	0	0	0	0
0	1	8	6	6	0
0	6	3	11	8	0
0	8	8	9	10	0
0	9	10	10	7	0
0	0	0	0	0	0





# Smoothing Spatial Filters

- Move mask across the zero-padded image

$$\frac{1}{6}$$

0	1	0
1	2	1
0	1	0

0	0	0	0	0	0
0	1	8	6	6	0
0	6	3	11	8	0
0	8	8	9	10	0
0	9	10	10	7	0
0	0	0	0	0	0

- Compute weighted sum

- Result:

2.6	4.3	6.2	4.3
4.0	6.5	8.0	7.2
6.5	7.7	9.5	7.3
6.0	7.8	7.7	5.7

round



3	4	6	4
4	7	8	7
7	8	10	7
6	8	8	6



# Order-Statistics Filters

- ◆ nonlinear spatial filters
- ◆ response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter
- ◆ replacing of the center pixel with the value determined by the ranking result
- ◆ Median filter
  - ◆ replaces the pixel value by the median value in the neighborhood
  - ◆ The principal function is to force distinct gray level points to be more like their neighbors.
  - ◆ excellent noise-reduction capabilities with less blurring than linear smoothing filters
  - ◆ effective for impulse noise (salt-and-pepper noise)



# Median Filter

6	2	0
3	97	4
19	3	10

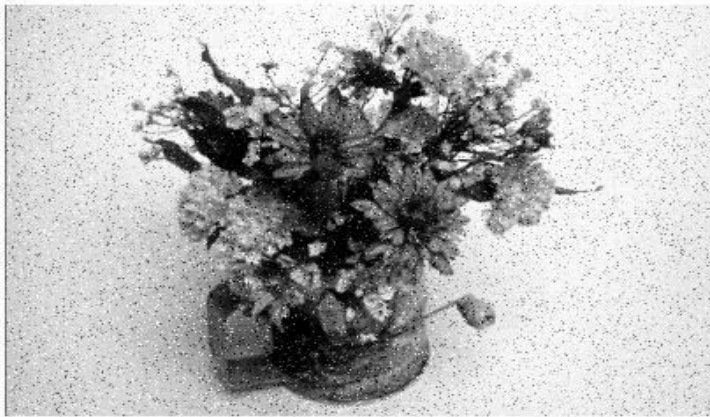


*	*	*
*	4	*
*	*	*

*order (0, 2, 3, 3, 4, 6, 10, 19, 97)*

*Median : 4*

*median filtered*



*original image with noise*



*3x3 median filtered*

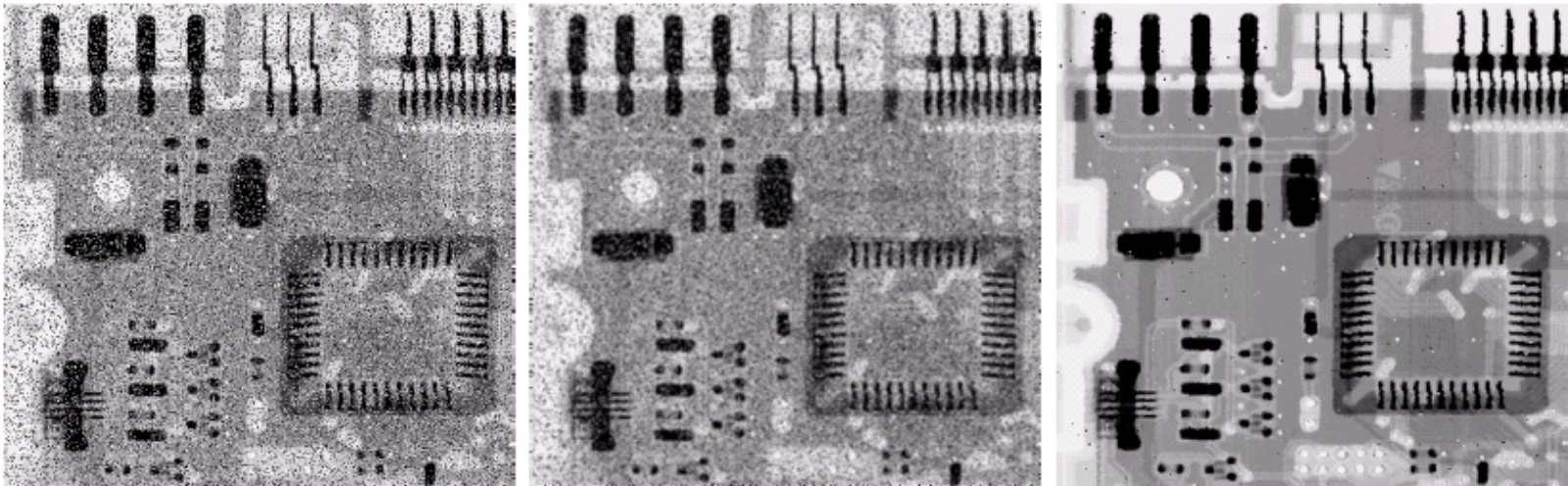
# Median Filter

- ◆ Isolated clusters that are dark or light and whose area is less than  $n^2/2$  (0.5 filter area) are eliminated by  $n \times n$  median.
  - ◆ Median represents the 50<sup>th</sup> percent.
    - **Min:** Set the pixel value to the minimum in the neighbourhood
    - **Max:** Set the pixel value to the maximum in the neighbourhood
- max filter  $\rightarrow R = \max \{z_k \mid k = 1, 2, \dots, 9\}$
- min filter  $\rightarrow R = \min \{z_k \mid k = 1, 2, \dots, 9\}$





# Median Filter



a b c

**FIGURE 3.37** (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a  $3 \times 3$  averaging mask. (c) Noise reduction with a  $3 \times 3$  median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

❖ ***Median filtering is much better suited than averaging for the removal of salt-and-pepper noise.***

# Sharpening Filters

- ◆ The principal objective of sharpening is **to highlight transition of intensity, highlight fine detail in an image, enhance detail that has been blurred**
- ◆ Averaging is analogous to integration, sharpening could be accomplished by spatial differentiation
- ◆ Image differentiation enhances edges and other discontinuities
- ◆ **First-order derivatives**
  - ◆ Zero in flat segments
  - ◆ Nonzero at the onset of a gray-level step or ramp
  - ◆ Nonzero along ramps

$$\frac{\partial f}{\partial x} \equiv f(x+1) - f(x)$$

- ◆ **Second-order derivatives**
  - ◆ Zero in flat areas
  - ◆ Nonzero at the onset and end of step or ramp
  - ◆ Zero along ramps of constant slope

$$\frac{\partial^2 f}{\partial x^2} \equiv f(x+1) + f(x-1) - 2f(x)$$

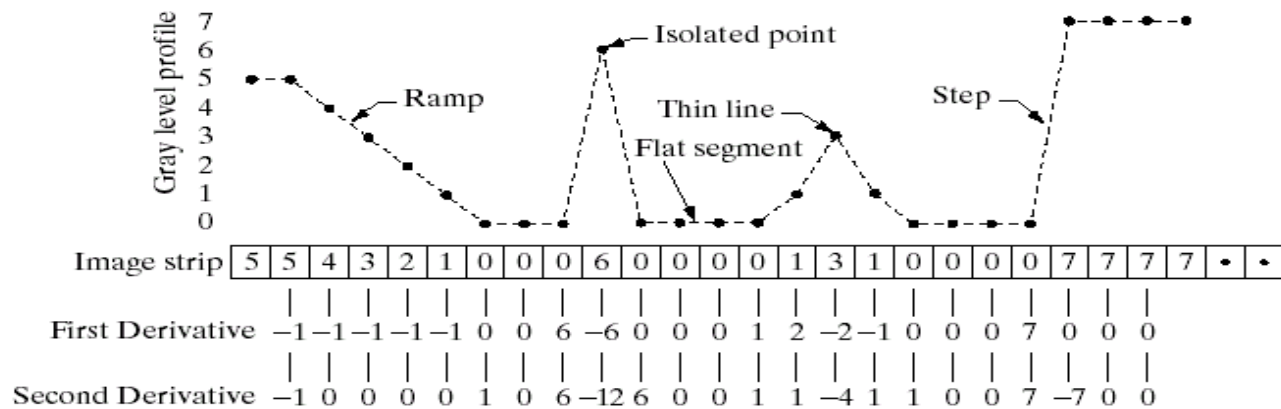
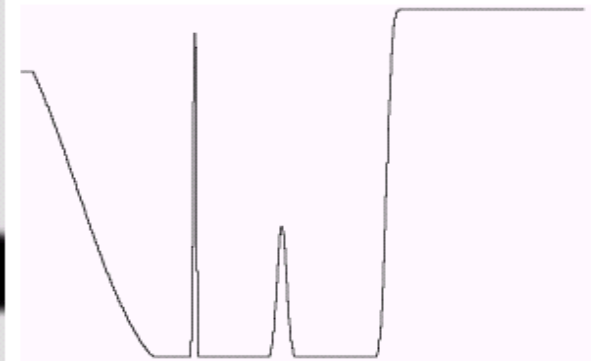
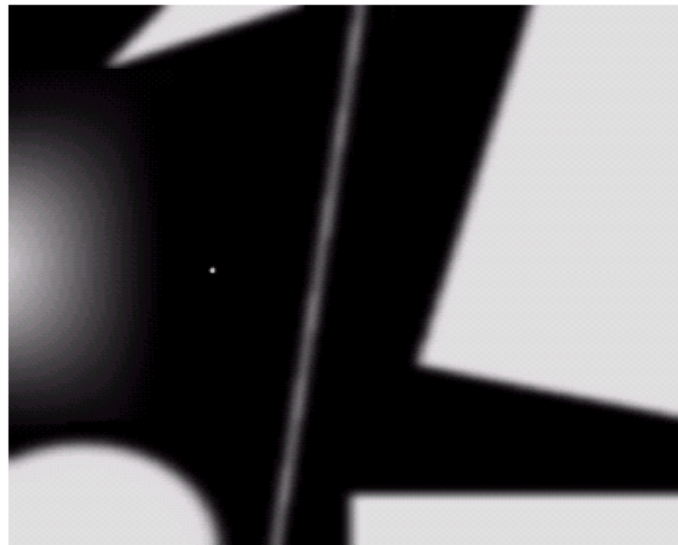


# First- and second-order derivatives

a b  
c

**FIGURE 3.38**

(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



# First- and second-order derivatives

## ◆ First-order derivatives

- ◆ Produce thicker edges in an image (nonzero along the entire ramp)
- ◆ Stronger response to a gray-level step
- ◆ Edge extraction

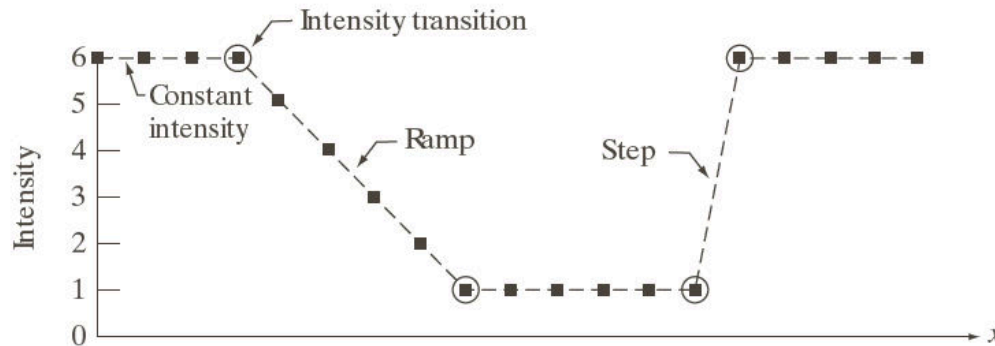
## ◆ Second-order derivatives

- ◆ Stronger response to fine detail, more aggressive (thin lines, isolated points)
- ◆ Transition from positive back to negative Produce double response at step changes in gray level (shown as thin double line)
- ◆ Suit for image enhancing sharp changes

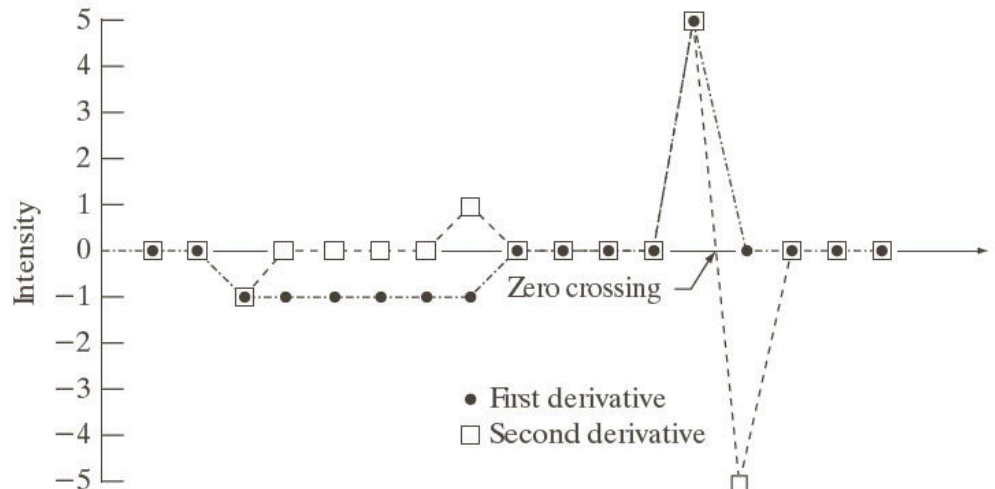




# First- and second-order derivatives



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	5	0	0	0	0	0
2nd derivative	0	0	-1	0	0	0	0	1	0	0	0	0	0	5	-5	0	0	0	0



a  
b  
c

**FIGURE 3.36**  
Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.



# The Laplacian

- ◆ the partial derivative in the  $x$  direction is defined as follows:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

- ◆ and in the  $y$  direction as follows

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$



# The Laplacian

- ◆ So, the Laplacian can be given as follows:

$$\nabla^2 f(x, y) = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

- ◆ We can easily build a filter based on this

0	1	0
1	-4	1
0	1	0



# The Laplacian

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

a	b
c	d

**FIGURE 3.39**

(a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4).

(b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.



# The Laplacian

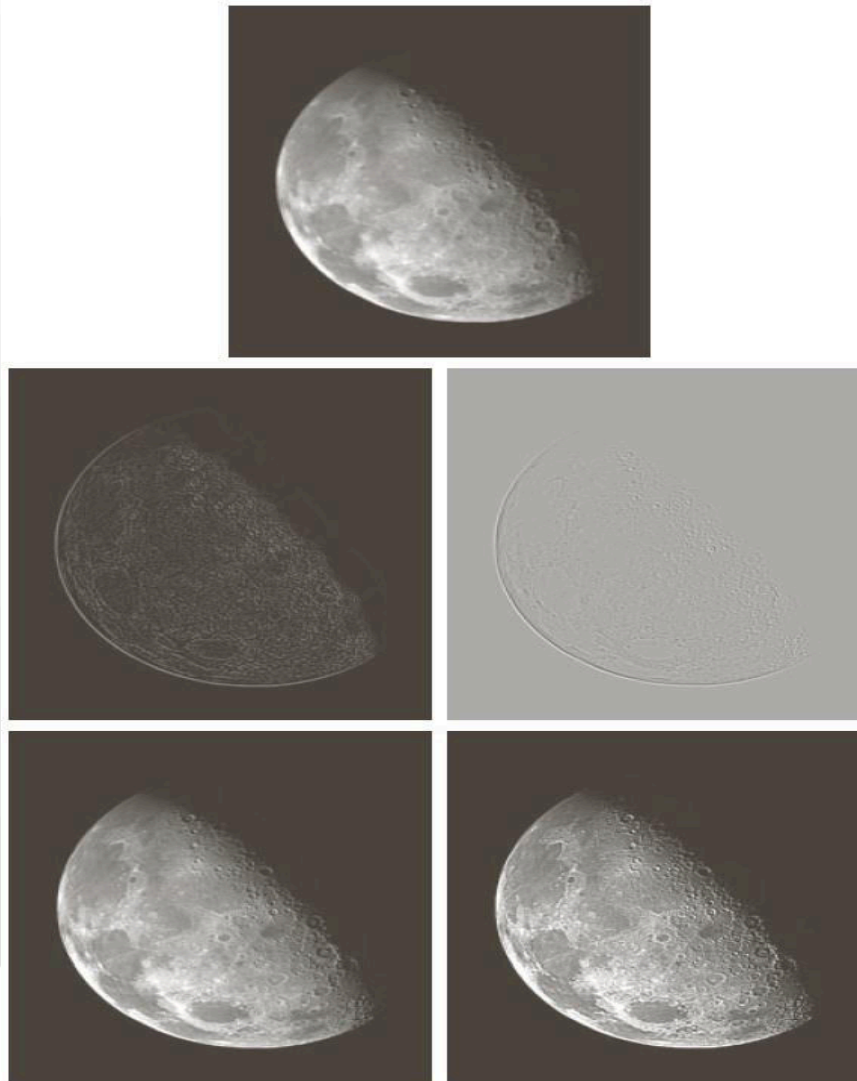
- ◆ Highlights intensity discontinuities in an image
- ◆ Deemphasizes regions with slowly varying gray levels
- ◆ Produces a new image that have grayish edge lines and other discontinuities, **(all superimposed on dark, featureless background)**
- ◆ The result of a Laplacian filtering is not an enhanced image
- ◆ We have to do more work in order to get our final image
- ◆ Add the Laplacian result from the original image to generate our final sharpened enhanced image

$$g(x, y) = f(x, y) + c \left[ \nabla^2 f(x, y) \right]$$

- ◆  $C = -1$  if negative center
- ◆  $C = 1$  if others



# The Laplacian



a  
b c  
d e

**FIGURE 3.38**

(a) Blurred image of the North Pole of the moon.  
(b) Laplacian without scaling.  
(c) Laplacian with scaling.  
(d) Image sharpened using the mask in Fig. 3.37(a).  
(e) Result of using the mask in Fig. 3.37(b).  
(Original image courtesy of NASA.)



# The Laplacian

- ◆ The entire enhancement can be combined into a single filtering operation

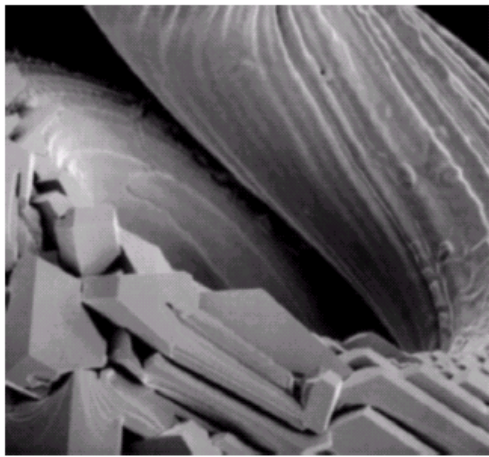
$$\begin{aligned}g(x, y) &= f(x, y) - \nabla^2 f \\ &= f(x, y) - [f(x+1, y) + f(x-1, y) \\ &\quad + f(x, y+1) + f(x, y-1) \\ &\quad - 4f(x, y)] \\ &= 5f(x, y) - f(x+1, y) - f(x-1, y) \\ &\quad - f(x, y+1) - f(x, y-1)\end{aligned}$$



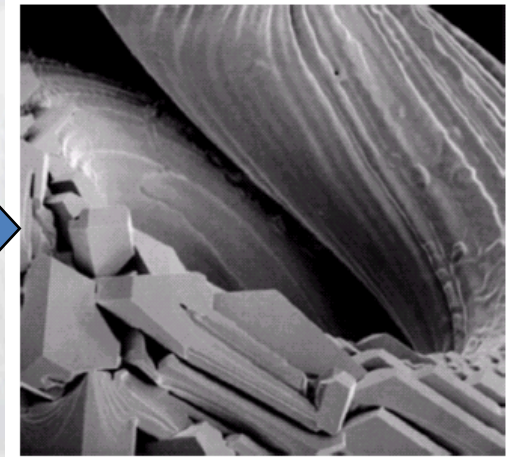


# The Laplacian

- ◆ This gives us a new filter which does the whole job for us in one step



0	-1	0
-1	5	-1
0	-1	0





# The Laplacian

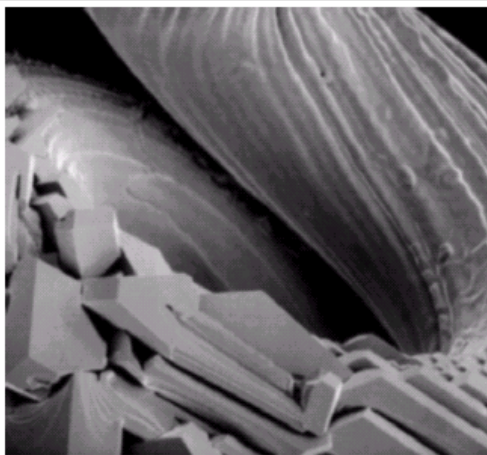
- ◆ There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

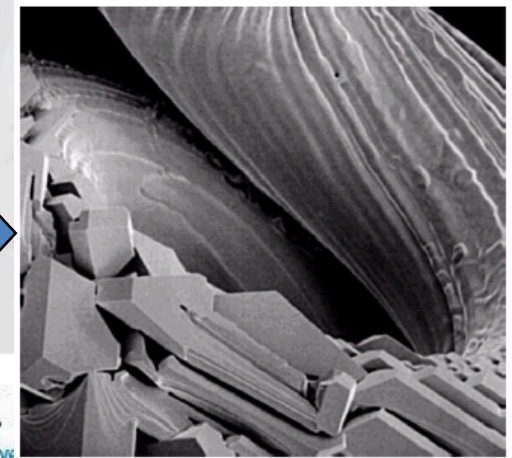
Simple  
Laplacian

1	1	1
1	-8	1
1	1	1

Variant of  
Laplacian



-1	-1	-1
-1	9	-1
-1	-1	-1



# Unsharp Masking And High-Boost Filtering

## ◆ Unsharp masking

- ◆ Blur the original image
- ◆ Subtract a blurred version from the original (diff = mask)
- ◆ Add the mask to the original

$$g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y)$$

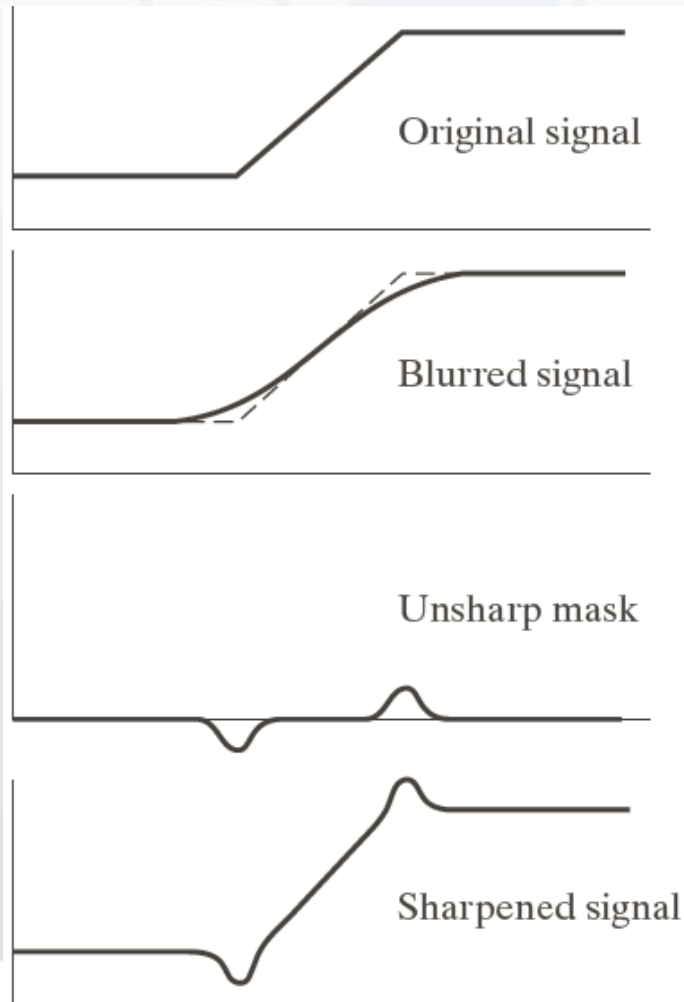
add a weighted portion of the mask back to the original

$$g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y)$$

- ◆  $K = 1$ , unsharp masking
- ◆  $K > 1$ , the process is referred to as High-boost filtering
- ◆  $K < 1$  de-emphasize the contribution of the unsharp mask



# Unsharp Masking And High-Boost Filtering



a  
b  
c  
d

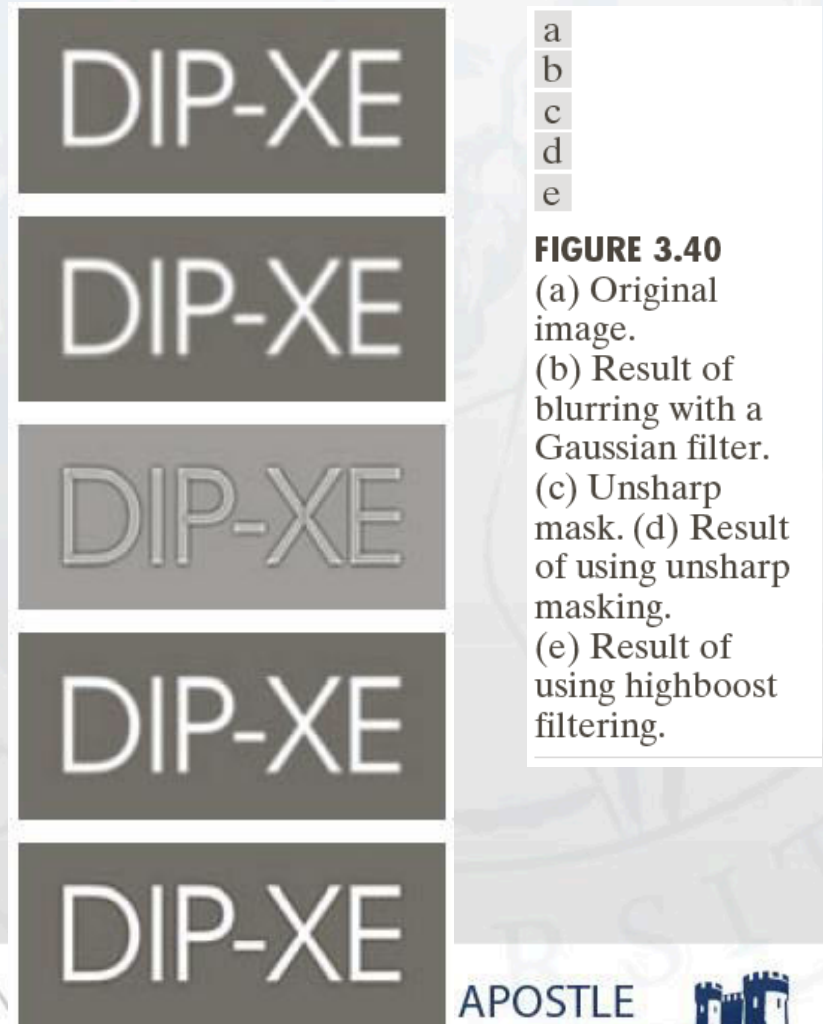
**FIGURE 3.39** 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).



# Unsharp Masking And High-Boost Filtering

◆ One of the principal applications of boost filtering is when the input image is darker than desired. By varying the boost coefficient, it generally is possible to obtain an overall increase in average gray level of the image, thus helping to brighten the final result.

◆  $K = 4.5$





# Unsharp Masking And High-Boost Filtering

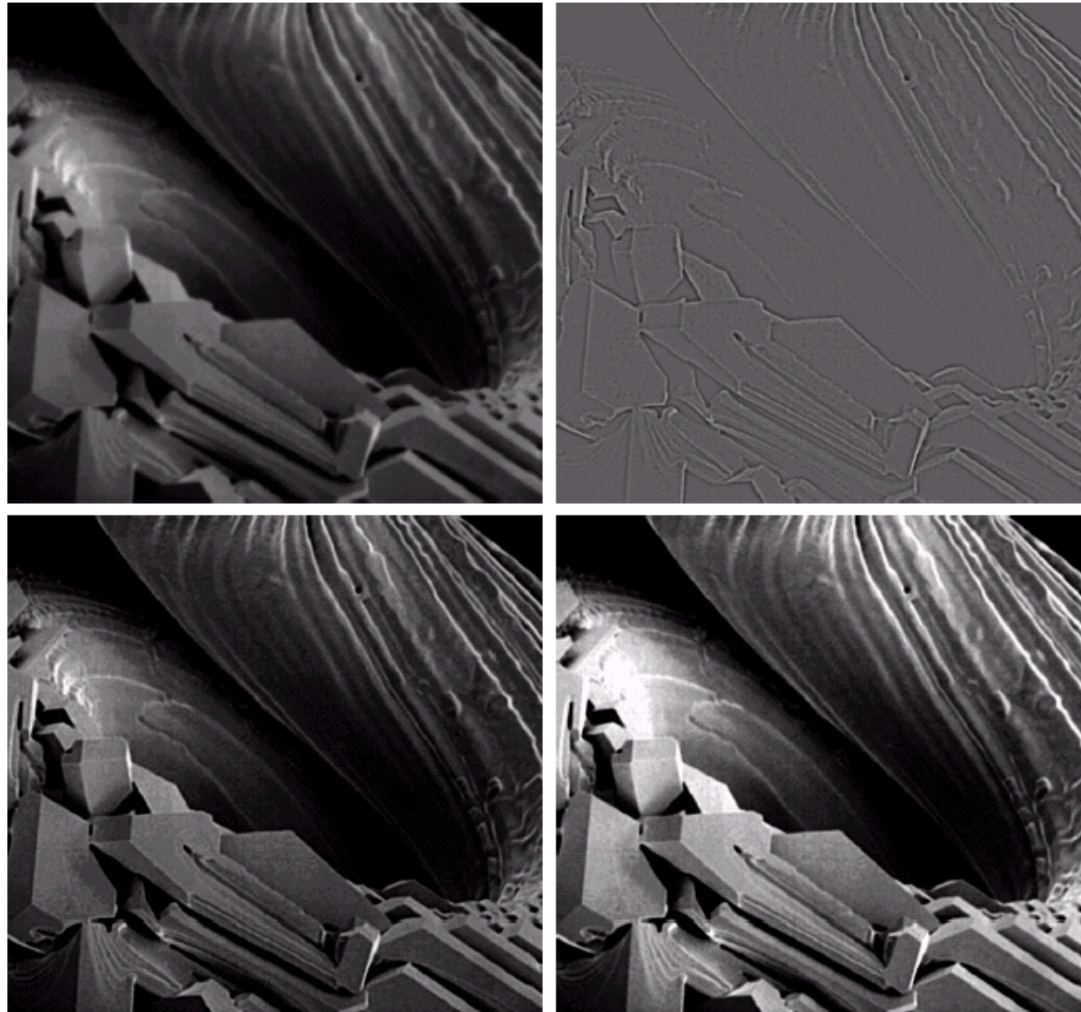
a b  
c d

**FIGURE 3.43**

(a) Same as Fig. 3.41(c), but darker.

(b) Laplacian of (a) computed with the mask in Fig. 3.42(b) using  $A = 0$ .

(c) Laplacian enhanced image using the mask in Fig. 3.42(b) with  $A = 1$ . (d) Same as (c), but using  $A = 1.7$ .



# First Derivative - The Gradient

- ◆ First derivatives are implemented using the magnitude of the gradient
- ◆ For a function  $f(x, y)$  the gradient of  $f$  at coordinates  $(x, y)$  is given as the column vector

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$



# First Derivative - The Gradient

- ◆ The magnitude (length) of this vector is given by:

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

- ◆ Is the value at (x,y) of the rate of change in the direction of the gradient vector.
- ◆ Components of the gradient vector itself are linear, the magnitude is not (square, root)
- ◆ The result is called the gradient image



# First Derivative - The Gradient

- ◆ For practical reasons this can be simplified by using absolute values instead of squares and square roots

$$M(x, y) \approx |g_x| + |g_y|$$

- ◆ Preserve relative changes in gray levels





# First Derivative - The Gradient

- ◆ There is some debate as to how best to calculate these gradients but we will use
- ◆ The simplest approximation to a first order derivative that satisfy the conditions stated are

$$g_x = (z_8 - z_5) \quad g_y = (z_6 - z_5)$$

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$



# First Derivative - The Gradient

## Robert cross-gradient operators

$$g_x = (z_9 - z_5) \quad g_y = (z_8 - z_6)$$

$$M(x, y) = \left[ (z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2}$$

$$M(x, y) \approx |z_9 - z_5| + |z_8 - z_6|$$

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

$g_x$

-1	0
0	1

$g_y$

0	-1
1	0



# Sobel operators

- The result of the Sobel operator at an image point which is in a region of constant image intensity is a zero vector
- At a point on an edge is a vector which points across the edge, from darker to brighter values

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$M(x, y) \approx \left| (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \right| \\ + \left| (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \right|$$



# Sobel operators

$g_x$

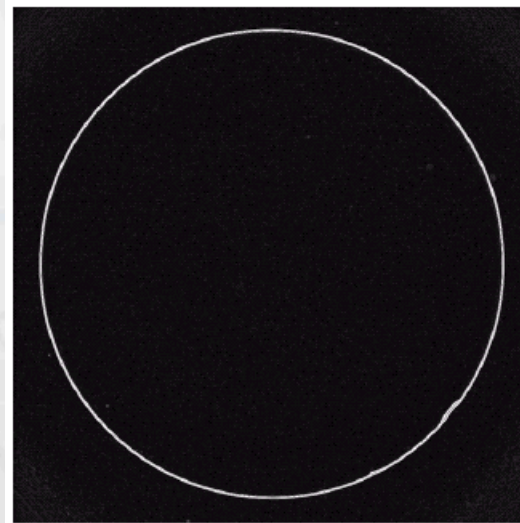
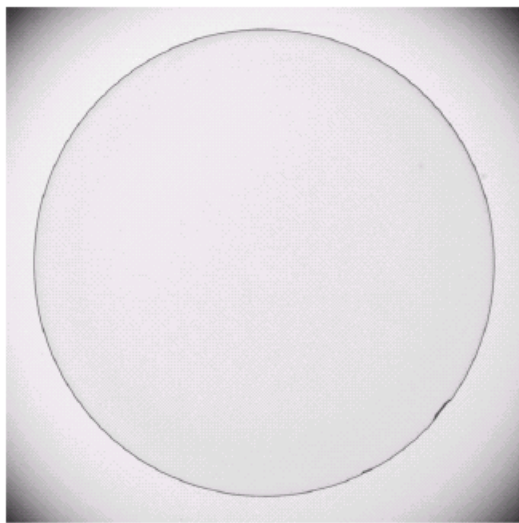
-1	-2	-1
0	0	0
1	2	1

$g_y$

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



# Sobel operators

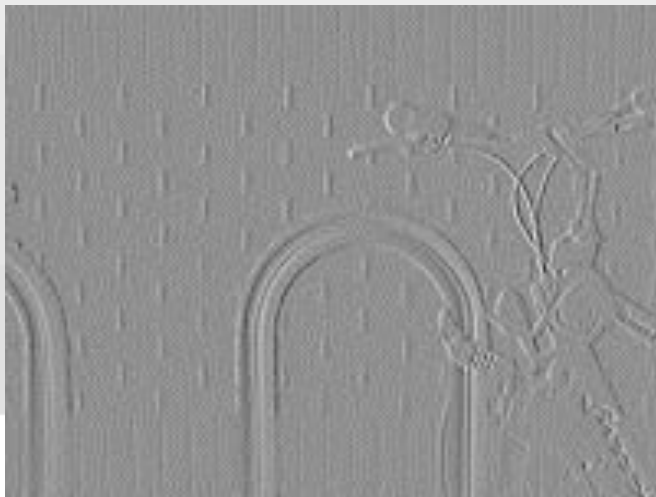


**An image of a contact lens which is enhanced in order to make defects (at four and five o'clock in the image) more obvious**

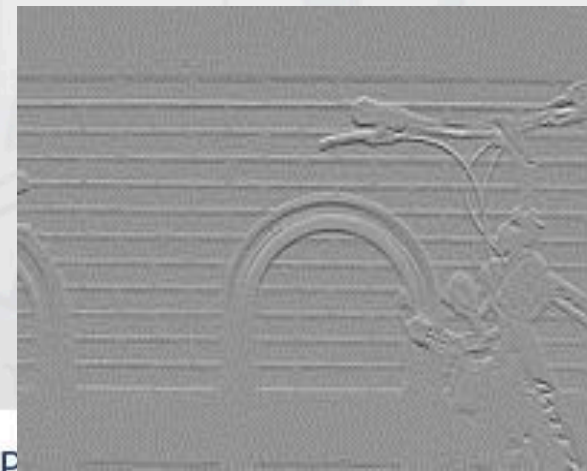
# Sobel operators



Grayscale image of a brick wall & a bike rack



Normalized sobel y-gradient image of bricks & bike rack



Normalized sobel x-gradient image of bricks & bike rack



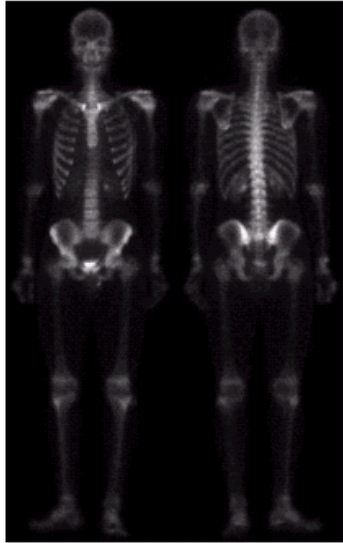


# Combining Spatial Enhancement Methods

- ❖ Successful image enhancement is typically not achieved using a single operation
- ❖ Rather we combine a range of techniques in order to achieve a final result
- ❖ This example will focus on enhancing the bone scan to the right

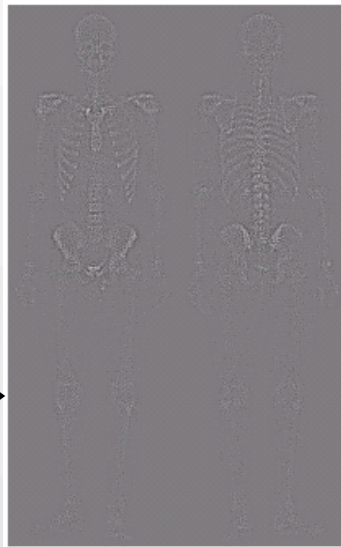


# Combining Spatial Enhancement Methods



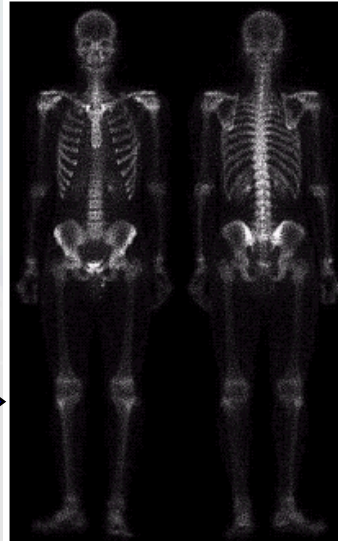
(a)

Laplacian filter of bone scan (a)



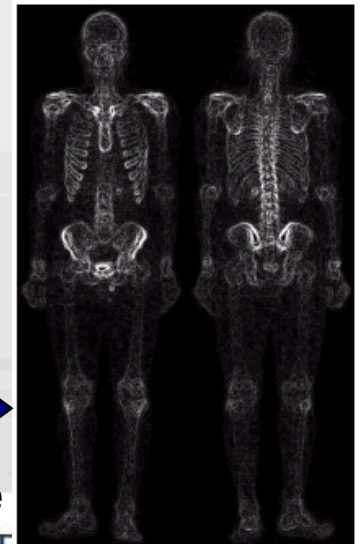
(b)

Sharpened version of bone scan achieved by subtracting (a) and (b)



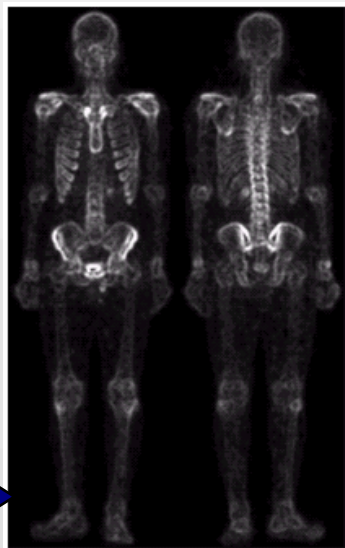
(c)

Sobel filter of bone scan (a)



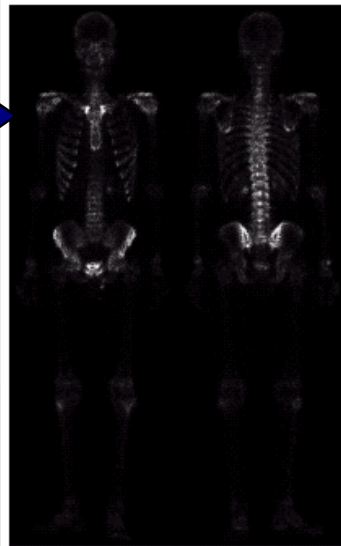
(d)

The product of (c) and (e) which will be used as a mask



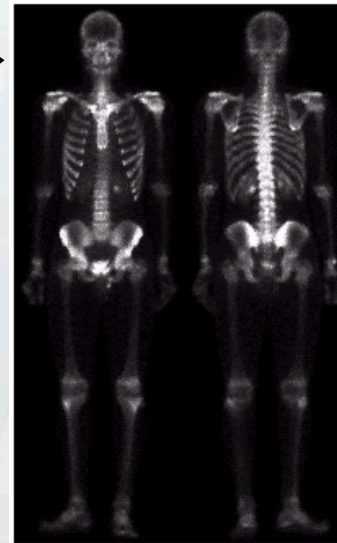
(e)

Sharpened image which is sum of (a) and (f)



(f)

Result of applying a power-law trans. to (g)



(g)

(h)

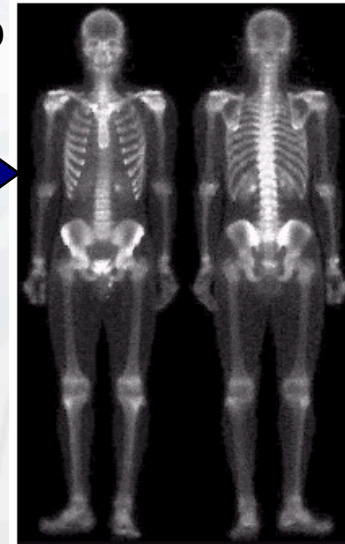


Image (d) smoothed with a 5\*5 averaging filter

# Combining Spatial Enhancement Methods

