Image Enhancement in *Spatial Domain*

Arithmetical and Logic Operations

- ***Arithmetic/logic operations are performed on a pixel by pixel basis between two or more images.***
- ***Basic arithmetic operations are: Gray-value point operations are used including the following functions***

Operation	*Definition*	*preferred data type*
ADD	**c = a + b**	**integer**
SUB	**c = a - b**	**integer**
MUL	**c = a * b**	**integer or floating point**
DIV	**c = a / b**	**floating point**
LOG	**c = log(a)**	**floating point**
EXP	**c = exp(a)**	**floating point**
SQRT	**c = sqrt(a)**	**floating point**
TRIG.	**c = sin/cos/tan(a)**	**floating point**
INVERT	**c = $(2^B - 1) - a$**	**integer**

Image Enhancement in *Spatial Domain*

Arithmetical and Logic Operations

- ***Basic logic operations are:*** *Binary operations are used including the following functions*

NOT	$c = \bar{a}$
OR	$c = a + b$
AND	$c = a \bullet b$
XOR	$c = a \oplus b = a \bullet \bar{b} + \bar{a} \bullet b$

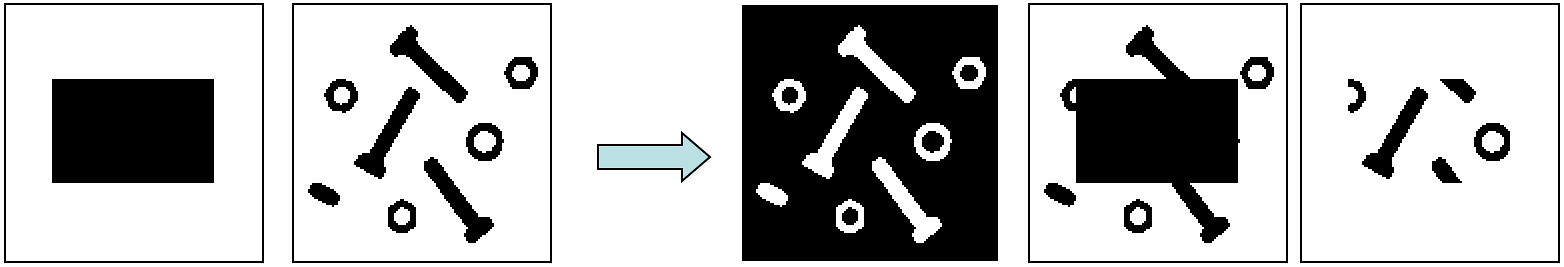


Image *a* **Image *b*** $\bar{b}$ $a.b$ $a + b$

Note: The images can be *binary (bi-level)* images. Each pixel is 1 *(True-white)* or 0 *(False-black)*.

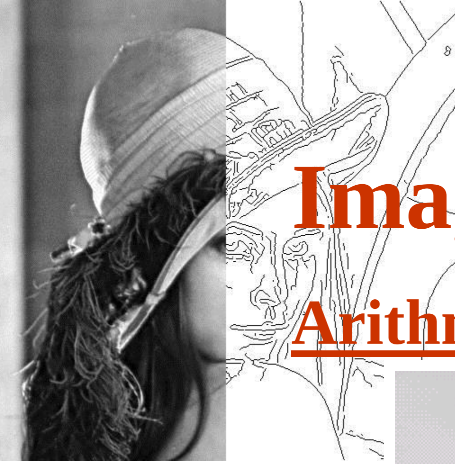
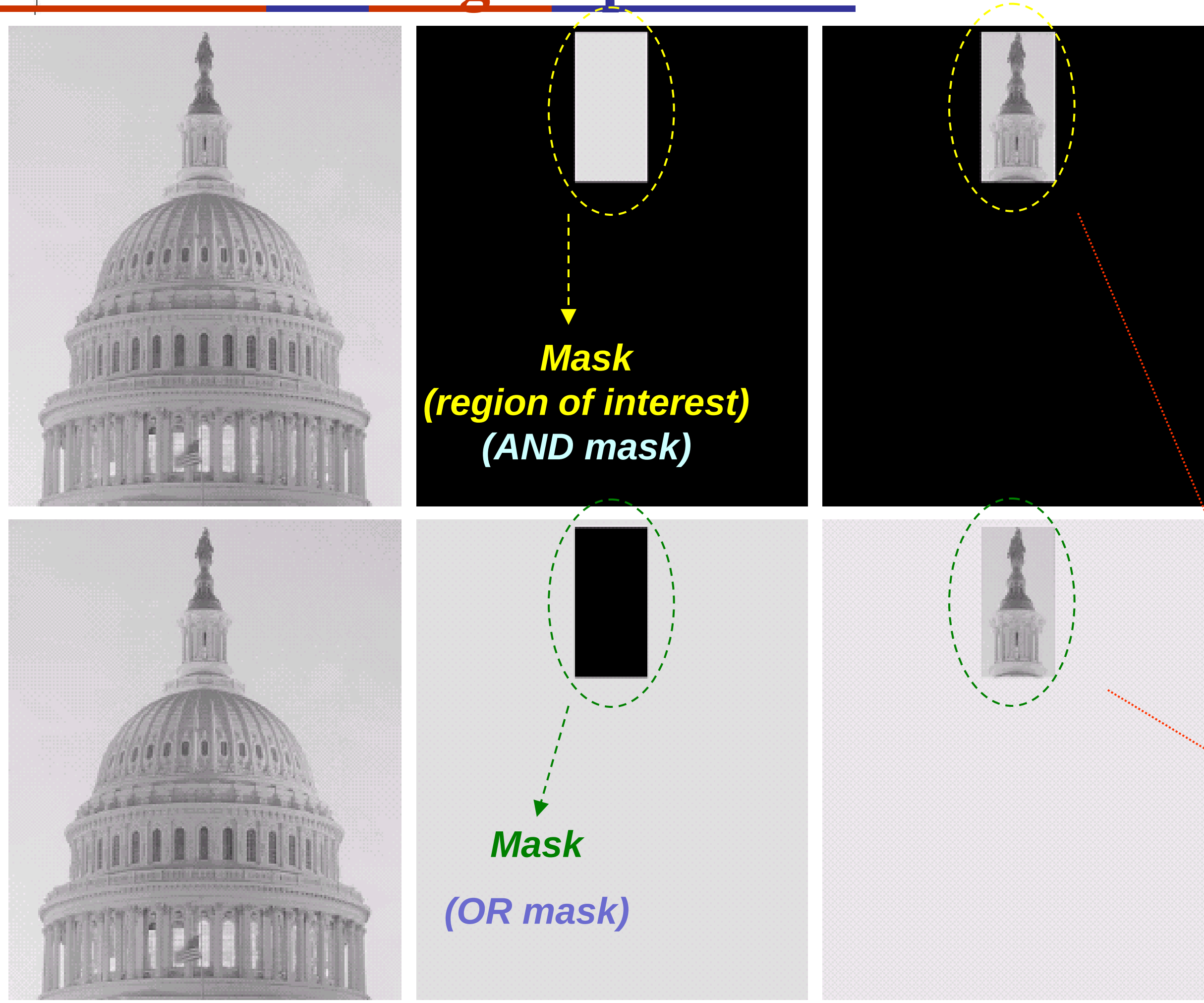


Image Enhancement in *Spatial Domain*

Arithmetical and Logic Operations



a b c
d e f

FIGURE 3.27 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

Note: The images can be *gray-level* images. Each pixel is an 8-bit binary number. Bit by bit operation is used.

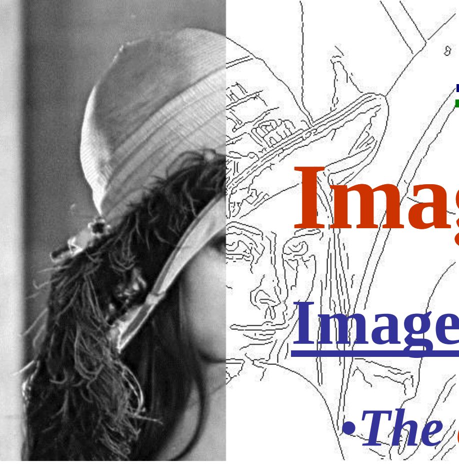
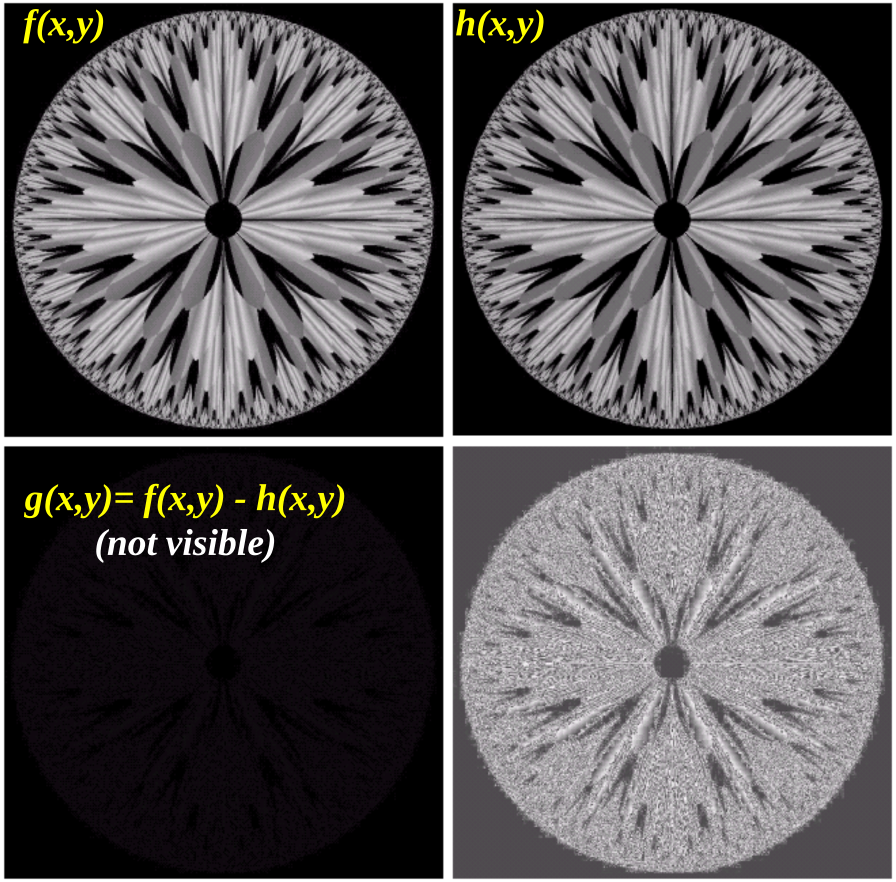


Image Enhancement in *Spatial Domain*

Image Subtraction

•The difference image between two images f(x.y) and h(x,y) can be expressed by:

$$g(x,y) = f(x,y) - h(x,y)$$



a b
c d

FIGURE 3.28 (a) Original fractal image. (b) Result of setting the four lower-order bit planes to zero. (c) Difference between (a) and (b). (d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).

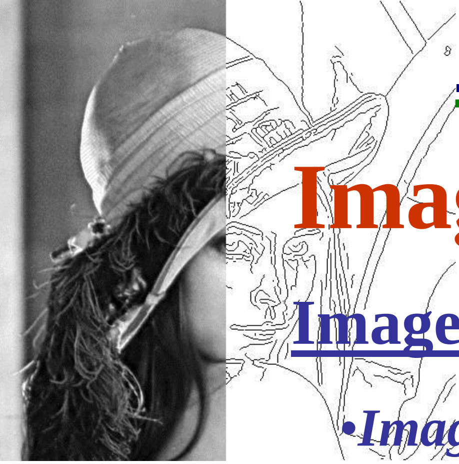
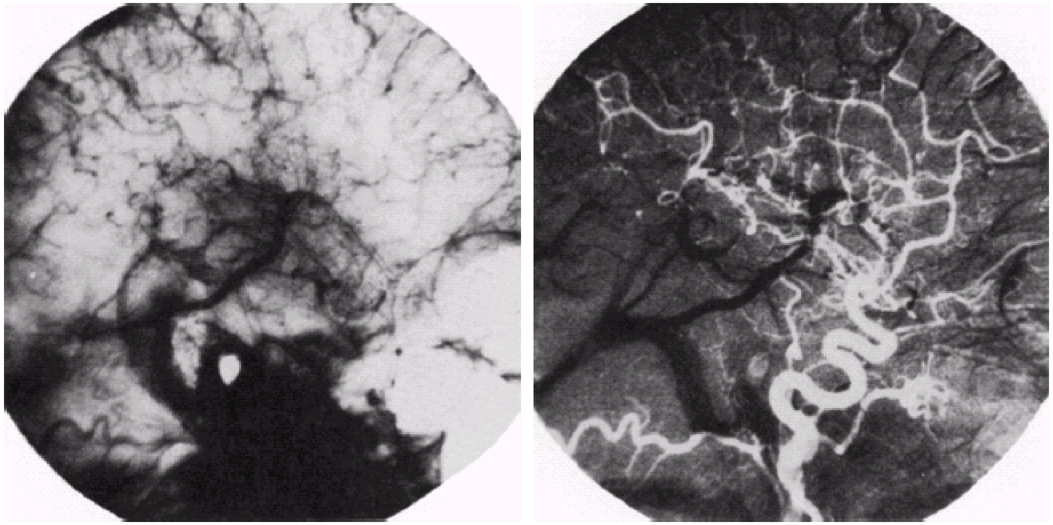


Image Enhancement in *Spatial Domain*

Image Subtraction

- ***Image subtraction is used in medical imaging called mask mode radiography.***

- ***The initial image is captured and used as the mask image, $h(x,y)$. Then after injecting a contrast material into the bloodstream the mask image is subtracted from the resulting image $f(x,y)$ to give an enhanced output image $g(x,y)$.***



a b

FIGURE 3.29 Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

$h(x,y)$ ***$g(x,y) = f(x,y) - h(x,y)$***

(initial image -Mask) ***(Result after Subtraction)***

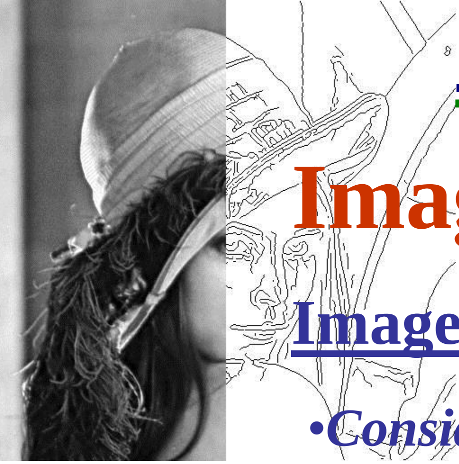


Image Enhancement in *Spatial Domain*

Image Averaging

•*Consider a noisy image, $g(x,y)$, formed by the addition of noise $\eta(x,y)$ to an original image $f(x,y)$:*

$$g(x,y) = f(x,y) + \eta(x,y)$$

•*Consider an uncorrelated noise with zero average value.*

•*An enhanced image , $\bar{g}(x,y)$, can be formed by adding K different noisy images.*

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

•*The expected value of $\bar{g}$:*

$$E\{\bar{g}(x,y)\} = f(x,y)$$

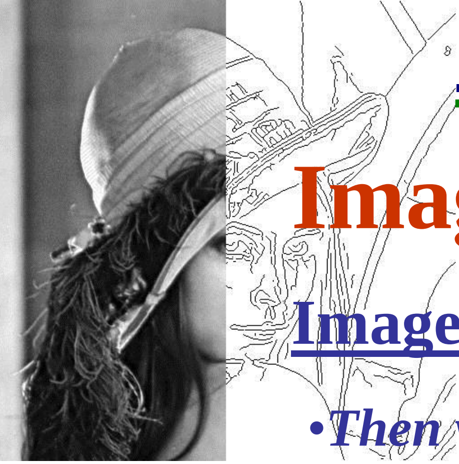


Image Enhancement in *Spatial Domain*

Image Averaging

Variance is dictated by noise

•*Then variances:*

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$$

•***Standard deviations in the average image:***

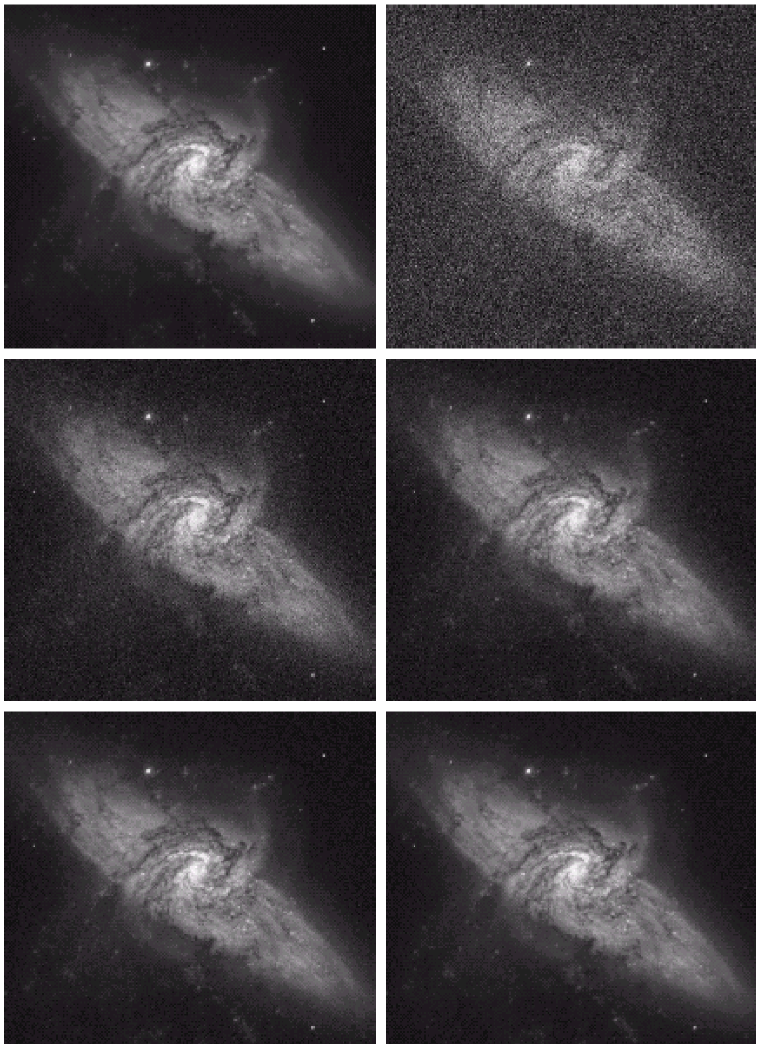
$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)}$$

•***As K increases the variability (noise) of the pixel values at each location (x,y) decreases.***

• ***In other words, the average image,*** $\bar{g}(x,y)$***, approaches the input image*** *f(x,y)* ***as the number of noisy images used in the averaging operation increases.***

Image Enhancement in *Spatial Domain*

Image Averaging



a	b
c	d
e	f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64$, and 128 noisy images. (Original image courtesy of NASA.)

Original im.

One of the noisy images

K=8

Average of K=16 noisy images

K=64

Average of K=128 noisy images

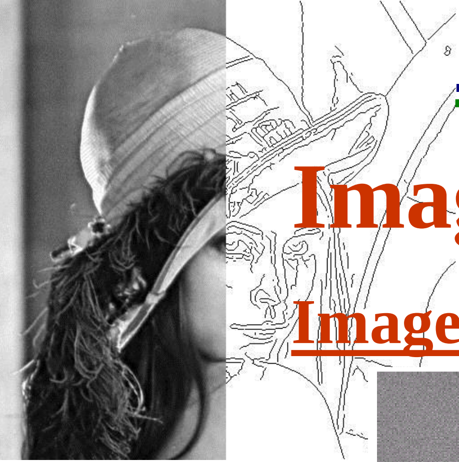


Image Enhancement in *Spatial Domain*

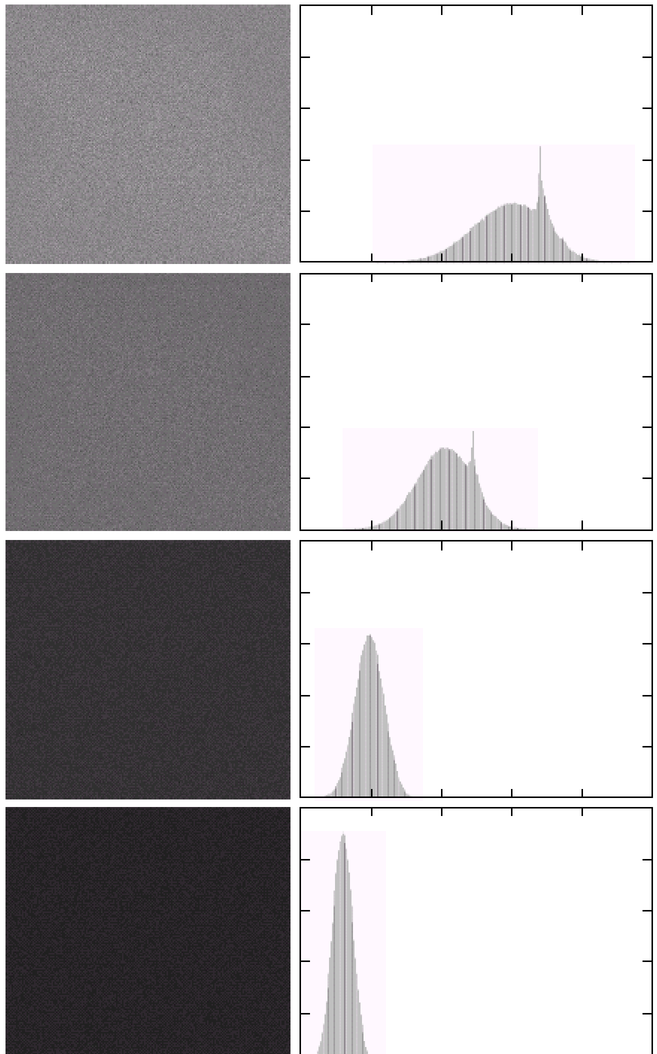
Image Averaging

K = 8

K = 16

K = 64

K = 128



a b

FIGURE 3.31 (a) From top to bottom: Difference images between Fig. 3.30(a) and the four images in Figs. 3.30(c) through (f), respectively. (b) Corresponding histograms.

difference = original - averaged

$$d(x,y) = f(x,y) - \bar{g}(x,y)$$

Prepared By: ***Dr. Hasan Demirel, PhD***

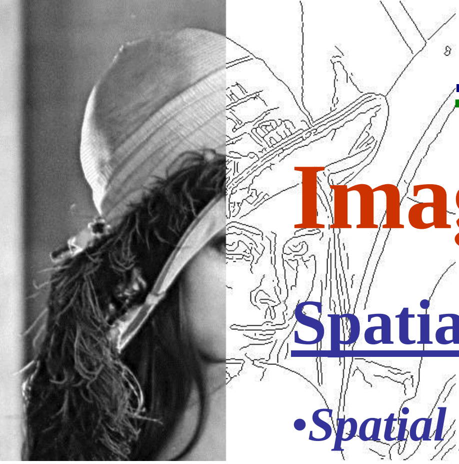


Image Enhancement in *Spatial Domain*

Spatial Filtering

•*Spatial filtering refers to some neighborhood operations working with the values of the image pixels in the neighborhood and the corresponding values of a subimage that has the same dimensions as the neighborhood.*

•*This subimage is called, a filter, mask, kernel, template or a window. The values in a filter is referred to as coefficients.*

•*The filtering can be performed in*
- *spatial domain.*
- *frequency domain (we will study later) and*

•*There are two main types of spatial domain filtering*
- *linear spatial filtering (convolution filter/mask/kernel) and*
- *nonlinear spatial filtering .*

Prepared By: ***Dr. Hasan Demirel, PhD***

Image Enhancement in *Spatial Domain*

Spatial Filtering

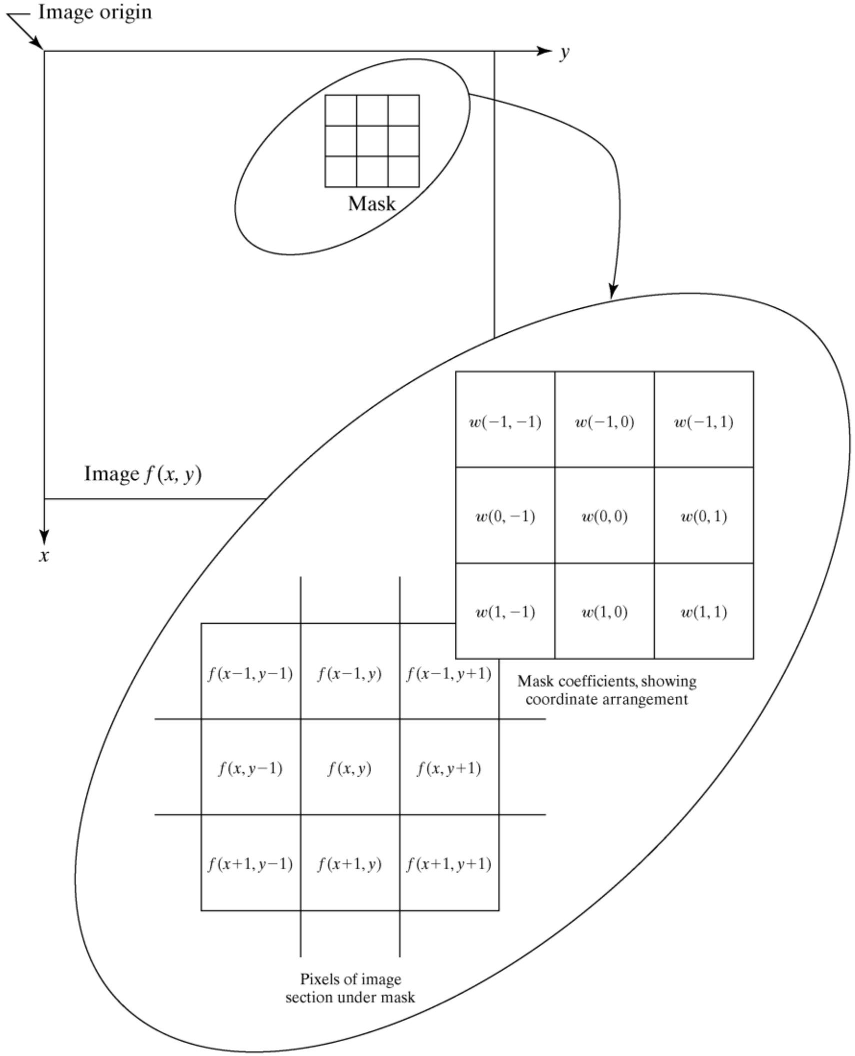


FIGURE 3.12 The mechanics of linear spatial filtering. The magnified drawing shows a 3 × 3 mask and the corresponding image neighborhood directly under it. The neighborhood is shown displaced out from under the mask for ease of readability.

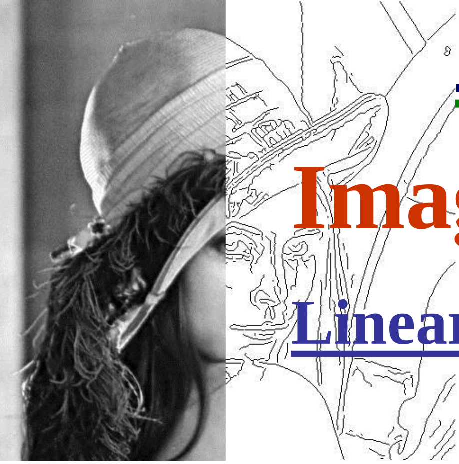
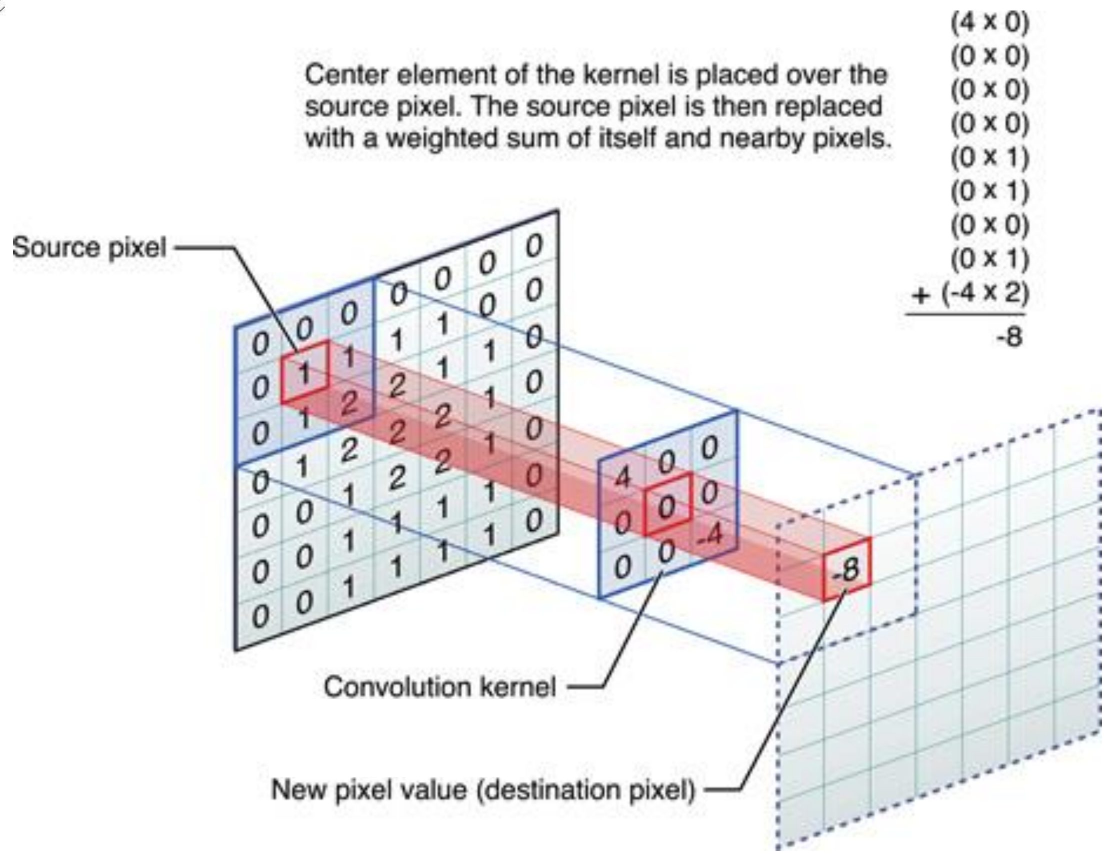


Image Enhancement in *Spatial Domain*

Linear Spatial Filtering



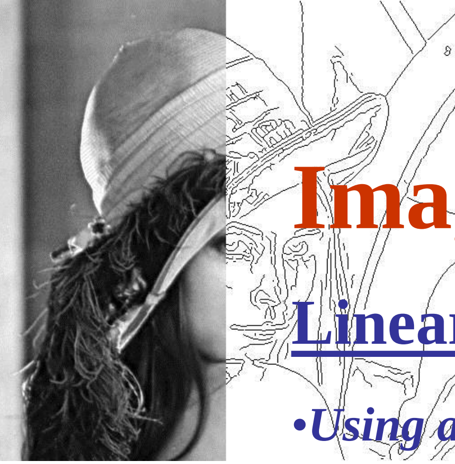


Image Enhancement in *Spatial Domain*

Linear Spatial Filtering

•*Using a* **3 x 3** *mask shown in the previous slide the response, R, of a linear filtering with the filter mask at point (x,y) in the image is:*

$$R = \omega(-1,-1)\,f(x-1,y-1) + \omega(-1,0)\,f(x-1,y) + \ldots$$
$$+\,\omega(0,0)\,f(x,y) + \ldots + \omega(1,0)\,f(x+1,y) + \omega(1,1)\,f(x+1,y+1)$$

•*In general, linear filtering of an image of size* **M x N** *with a filter mask of size* ***m* x *n*** *is given by:*

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

- *where* ***a* =(*m* - 1)/2, *b* =(*n* - 1)/2**

Prepared By: ***Dr. Hasan Demirel, PhD***

Image Enhancement in *Spatial Domain*

Linear Spatial Filtering

•*Linear spatial filtering is often called **convolution** operation and the filter mask is also referred to as **convolution mask**.*

•*Response, R, of a **m x n** mask at any point **(x,y)** in the image can be formulated by:*

$$R = \omega_1 z_1 + \omega_2 z_2 + \ldots + \omega_{mn} z_{mn}$$

$$= \sum_{i=1}^{mn} \omega_i z_i$$

•*Where, **ω**'s are the mask coefficients and **z**'s are the image pixel values.*

•*Given a **3 x 3** mask below the response at any point **(x,y)** is:*

FIGURE 3.33 Another representation of a general 3 × 3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$R = \sum_{i=1}^{9} \omega_i z_i$$

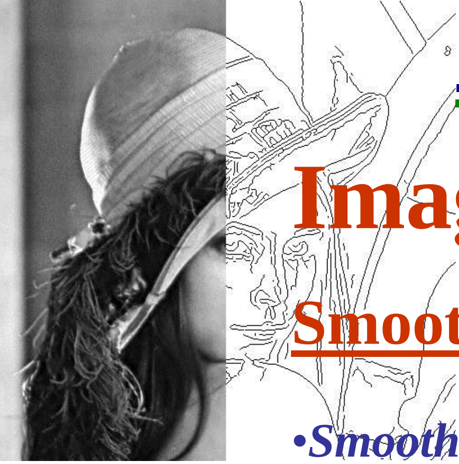


Image Enhancement in *Spatial Domain*

Smoothing Spatial Filters

•*Smoothing filters are used for noise reduction and blurring operations. Blurring can be used as a preprocessing step for other image processing operations.*

• *There are two main types of Smoothing filters:*

- *•Smoothing Linear Filters*
- *•Smoothing Nonlinear Filters*

Smoothing Linear Filters/ Averaging Filters

•*The response of a smoothing linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.*

•*These kind of filters are called averaging filters or lowpass filters.*

Image Enhancement in *Spatial Domain*

Convolution and Correlation

•***Convolution*** *involves calculating the weighted sum of a neighborhood of pixels. The weights are taken from a convolution kernel. Each value from the neighborhood of pixels is multiplied with its opposite on the matrix. For example, the top-left of the neighbor is multiplied by the bottom-right of the kernel. All these values are summed up to calculate the result of the convolution.*

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

$$g=\omega * f$$

Consider a 3x3 neighborhood. Given a convolution kernel (mask) ω, you need to rotate the mask with 180° as follows,

Image Enhancement in *Spatial Domain*

Convolution operation

Convolution kernel, ω

1	-1	-1
1	2	-1
1	1	1

Rotate 180°

1	1	1
-1	2	1
-1	-1	1

Input Image f

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Convolution: Step 1

1	1	1		
-1	4	2	2	3
-1	-2	1	3	3
	2	2	1	2
	1	3	2	2

Input Image, *f*

5			

output Image, *g*

1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Convolution: Step 2

1	1	1	
-2	4	2	3
-2	-1	3	3
2	2	1	2
1	3	2	2

Input Image, ***f***

5	4		

output Image, ***g***

1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Convolution: Step 3

	1	1	1
2	-2	4	3
2	-1	-3	3
2	2	1	2
1	3	2	2

Input Image, *f*

5	4	4	

output Image, *g*

1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Convolution: Step 4

		1	1	1
2	2	-2	6	1
2	1	-3	-3	1
2	2	1	2	
1	3	2	2	

5	4	4	-2

Input Image, *f*

output Image, *g*

1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Convolution: Step 5

1	2	2	2	3
-1	4	1	3	3
-1	-2	2	1	2
	1	3	2	2

Input Image, *f*

5	4	4	-2
9			

output Image, *g*

1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Convolution: Step 6

2	2	2	3
-2	2	3	3
-2	-2	1	2
1	3	2	2

Input Image, ***f***

5	4	4	-2
9	6		

output Image, ***g***

Convolution: Final Result

5	4	4	-2
9	6	14	5
11	7	6	5
9	12	8	5

Final output Image, *g*

Image Enhancement in *Spatial Domain*

Convolution and Correlation

•***Correlation* is nearly identical to convolution with only a minor difference, where instead of multiplying the pixel by the opposite in the kernel, you multiply it by the equivalent (i.e. top-left multiplied by top-left).**

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

$$g=\omega\circ f$$

Consider a 3x3 neighborhood. Given a correlation kernel (mask) ω, and input image f,

Image Enhancement in *Spatial Domain*

Correlation operation

orrelation kernel, ω

1	-1	-1
1	2	-1
1	1	1

Don't rotate use it directly

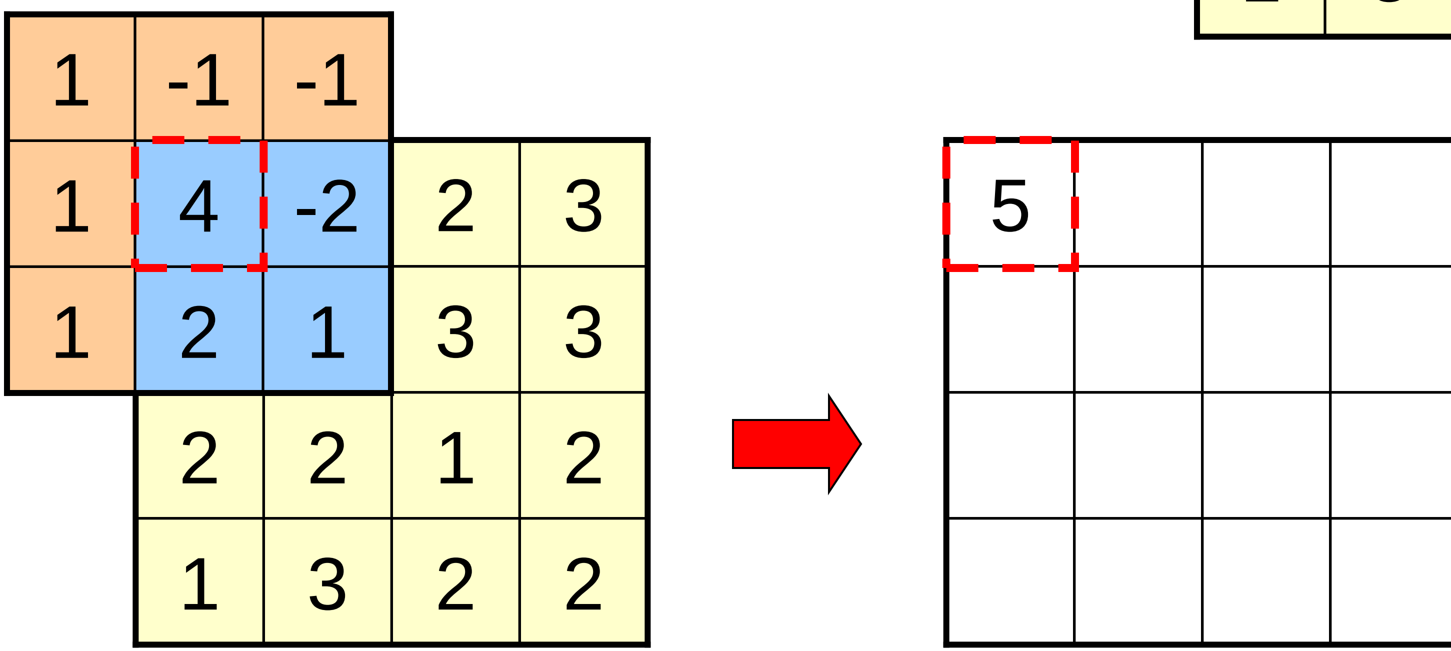
Input Image f

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

1	-1	-1
1	2	-1
1	1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Correlation: Step 1



Input Image, *f*

output Image, *g*

1	-1	-1
1	2	-1
1	1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Correlation: Step 2

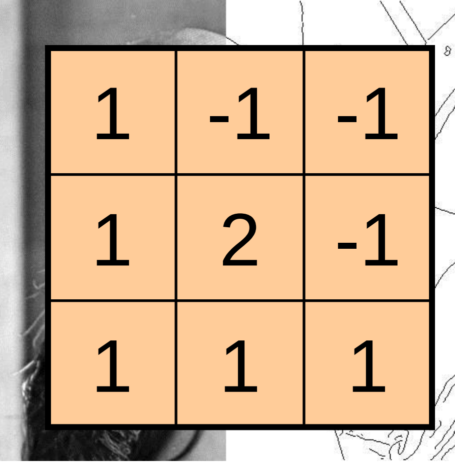
1	-1	-1	
2	4	-2	3
2	1	3	3
2	2	1	2
1	3	2	2

Input Image, ***f***

5	10		

output Image, ***g***

Prepared By: ***Dr. Hasan Demirel, PhD***



1	-1	-1
1	2	-1
1	1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Correlation: Step 3

	1	-1	-1
2	2	4	-3
2	1	3	3
2	2	1	2
1	3	2	2

Input Image, ***f***

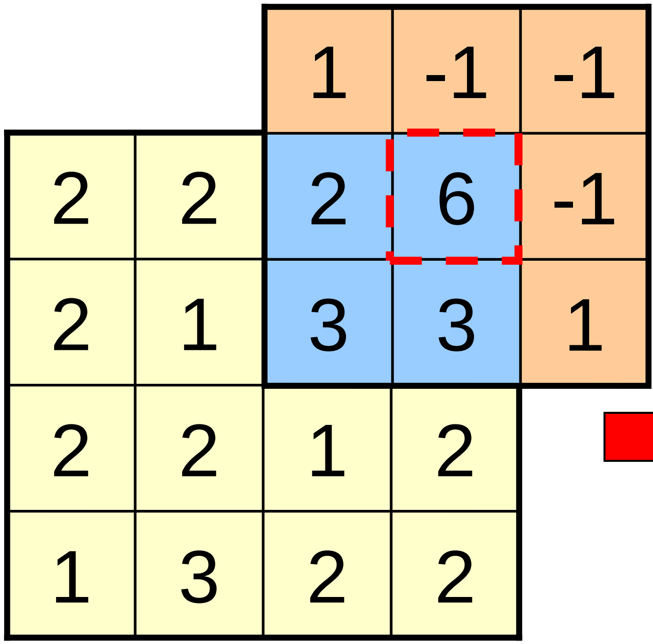
5	10	10	

output Image, ***g***

1	-1	-1
1	2	-1
1	1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Correlation: Step 4



Input Image, *f*

5	10	10	15

output Image, *g*

1	-1	-1
1	2	-1
1	1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

Correlation: Step 5

1	-2	-2	2	3
1	4	-1	3	3
1	2	2	1	2
	1	3	2	2

Input Image, *f*

5	10	10	15
3			

output Image, *g*

1	-1	-1
1	2	-1
1	1	1

Correlation: Step 6

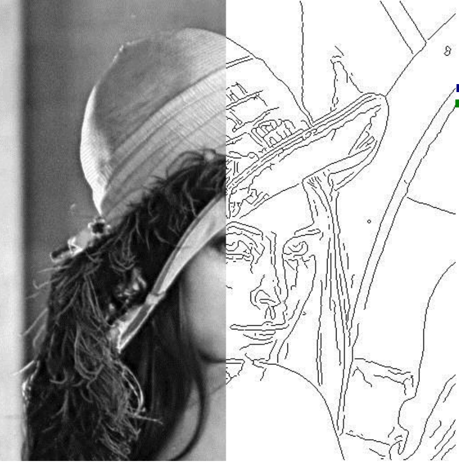
2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

2	-2	-2	3
2	2	-3	3
2	2	1	2
1	3	2	2

Input Image, *f*

5	10	10	15
3	4		

output Image, *g*



Correlation: Final Result

5	10	10	15
3	4	6	11
7	11	4	9
-5	4	4	5

Final output Image, g

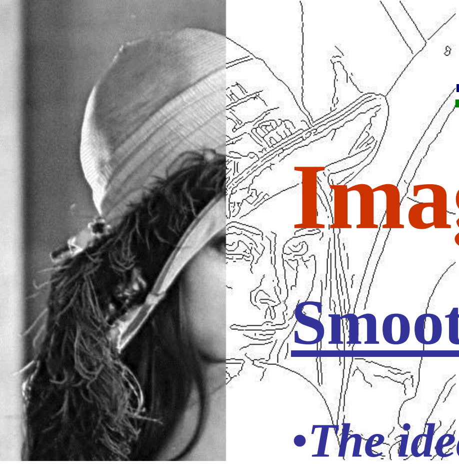


Image Enhancement in *Spatial Domain*

Smoothing Linear Filters/ Averaging Filters

•*The idea behind smoothing filters is to replace the value of every pixel in an image by the average of the gray levels defined by the filter mask.*

•*Random noise consists of sharp transitions in gray levels. So, the most obvious application is noise reduction.*

•*The undesirable effects of the averaging filters is the blurring of edges.*

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

box filter

weighted average filter

a b

FIGURE 3.34 Two 3 × 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

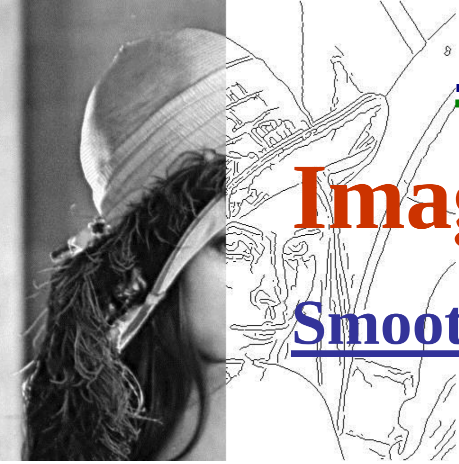
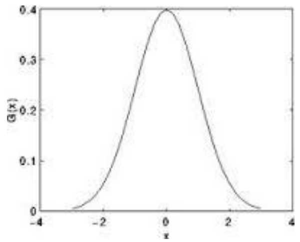


Image Enhancement in *Spatial Domain*

Smoothing Linear Filters/ Gaussian Smoothing

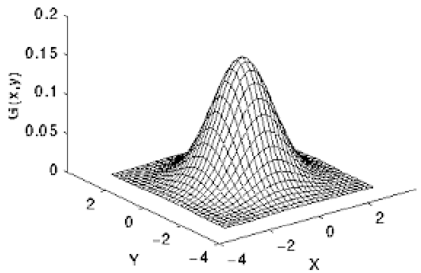
1D Gaussian distribution:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



2D Gaussian distribution:
($\sigma = \sigma_x = \sigma_y$)

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

3x3 mask/kernel

$$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

5x5 mask/kernel

Prepared By: *Dr. Hasan Demirel, PhD*

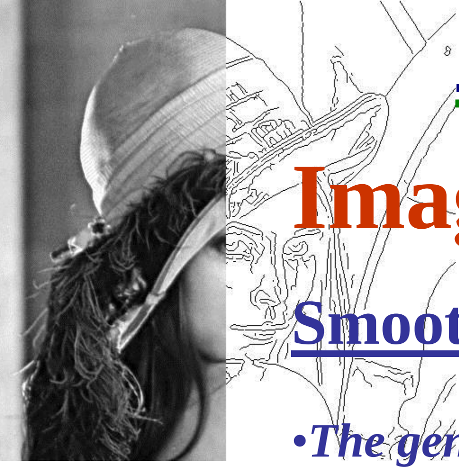


Image Enhancement in *Spatial Domain*

Smoothing Linear Filters/ Averaging Filters

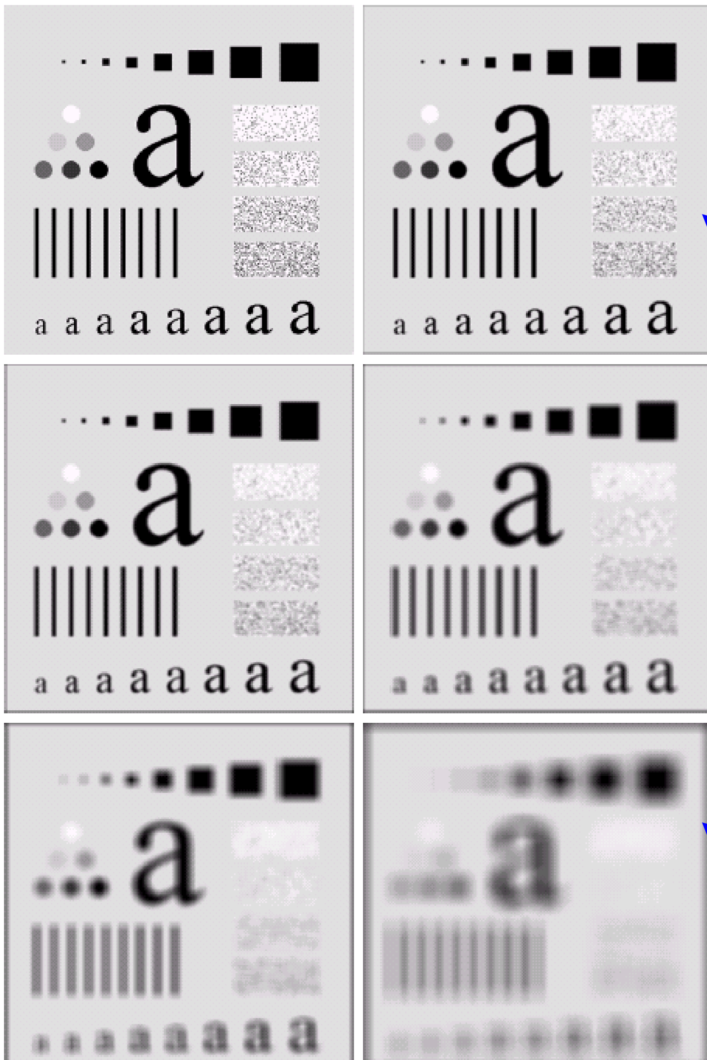
•*The general implementation of an M x N image with a weighted averaging filter of size m x n is given by:*

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

Sum of the mask coefficients, which is constant.

Image Enhancement in *Spatial Domain*

Smoothing Linear Filters/ Averaging Filters



a	b
c	d
e	f

FIGURE 3.35 (a) Original image, of size 500 × 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 × 120 pixels.

original

3x3 *mask*

5x5 *mask*

9x9 *mask*

15x15 *mask*

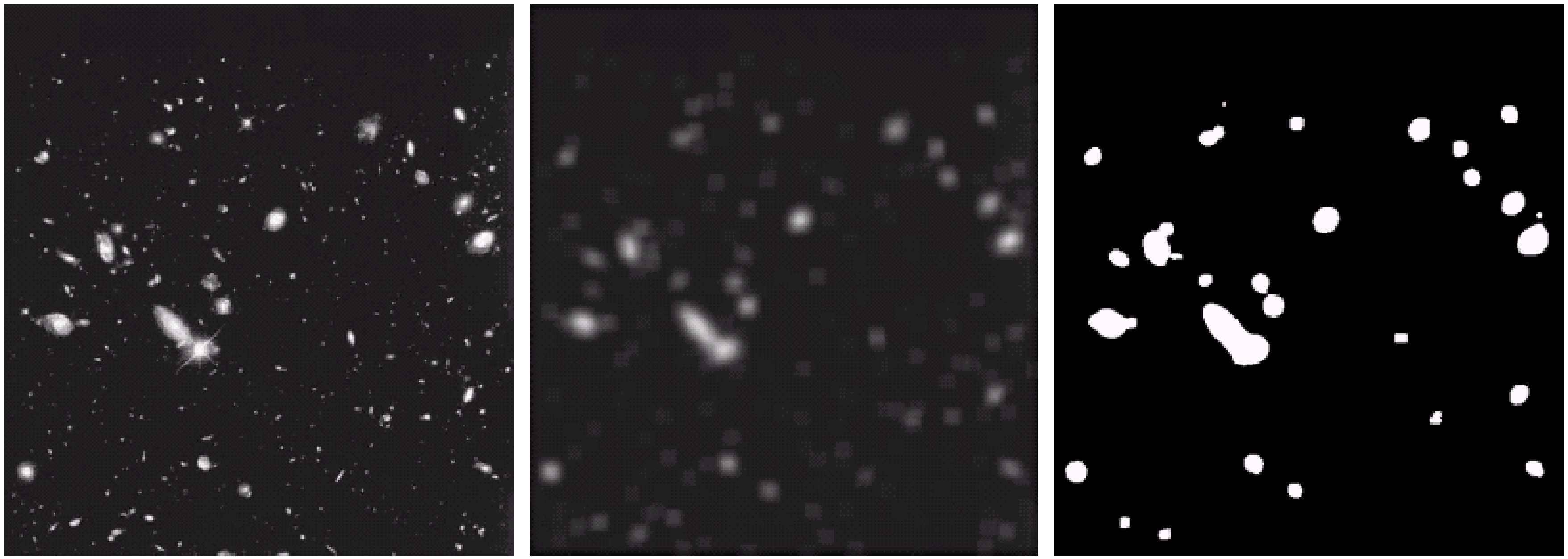
35x35 *mask*

Prepared By: *Dr. Hasan Demirel, PhD*

Image Enhancement in *Spatial Domain*

Smoothing Linear Filters/ Averaging Filters

original *blurred* *thresholded*



a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Image Enhancement in *Spatial Domain*

Order-Statistics Filters

•*The order-statistics filters are nonlinear spatial filters whose response is based on the ordering/ranking of the pixels contained image area encompassed by the filter.*

•*The center pixel is replaced with the value determined by the ranking result.*

•*The best known ordered –statistics filter is the median filter.*

•*The median filter is excellent for random noise reduction with considerably less blurring than the linear smoothing filters.*

•*Median filters is very effective for impulsive noise which is also called salt-and-pepper noise (noise introducing white and black dots on the image)*

•*Given a* **3x3** *neighborhood having (10, 20 ,20 ,20 ,100 ,20 ,20 ,25 ,15) gray level values. The sorted values of the neighborhood will be: (10, 15 ,20 ,20 ,20 ,20 ,20 ,25 ,100) and the center pixel will be forced to the median value which is 20.*

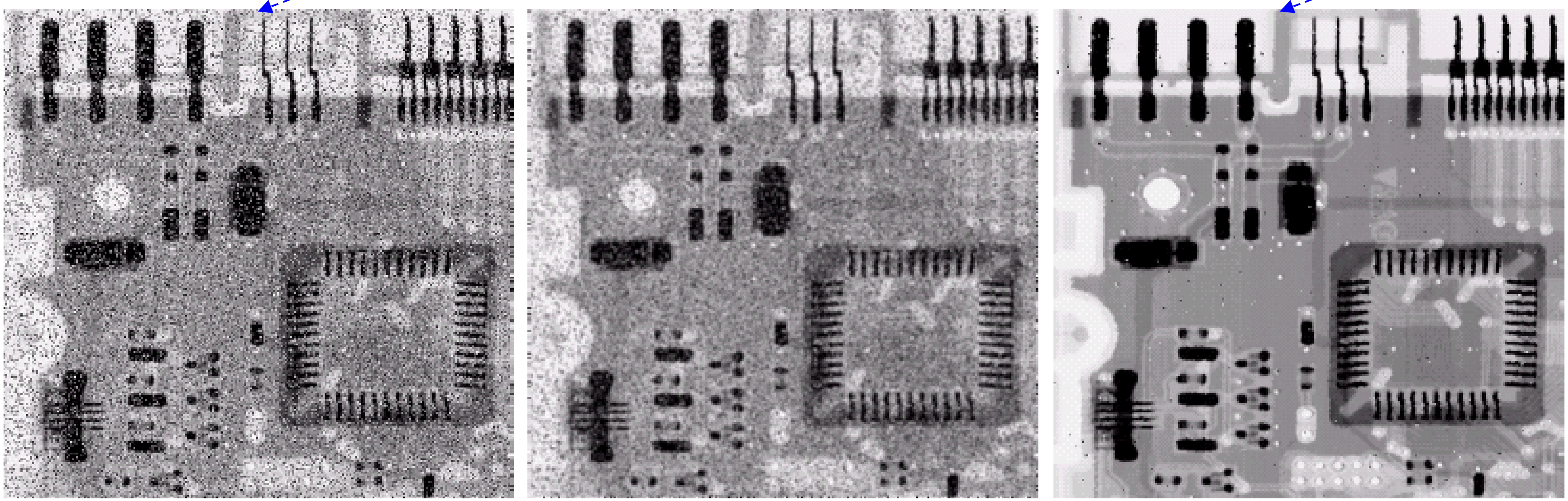
Image Enhancement in *Spatial Domain*

Order-Statistics Filters

Corrupted by salt and pepper noise

Averaging filter

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 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

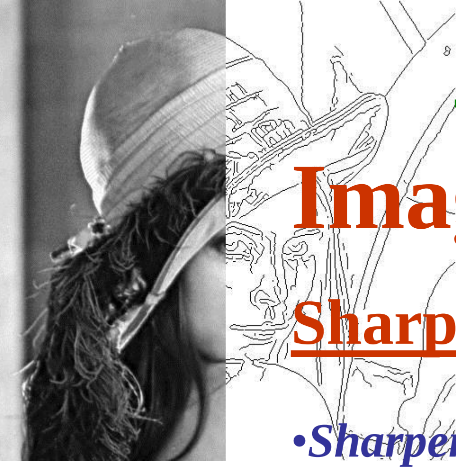


Image Enhancement in *Spatial Domain*

Sharpening Spatial Filters

• ***Sharpening* is the operation to highlight fine details or enhance the details that has been blurred.**

• ***Blurring* is based on the *averaging* in a neighborhood which is analogous to *integration*. Therefore the *sharpening* could be accomplished by *differentiation*.**

• ***The derivative of a digital function is defined in terms of differences, where a first order derivative of a one dimensional function f(x) is:***

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

• ***Second order derivative can be defined by:***

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= f(x+1) - f(x) - (f(x) - f(x-1)) \\ &= f(x+1) + f(x-1) - 2f(x)\end{aligned}$$

Image Enhancement in *Spatial Domain*

Sharpening Spatial Filters

•The effect of the first and second-order derivatives on an image are:

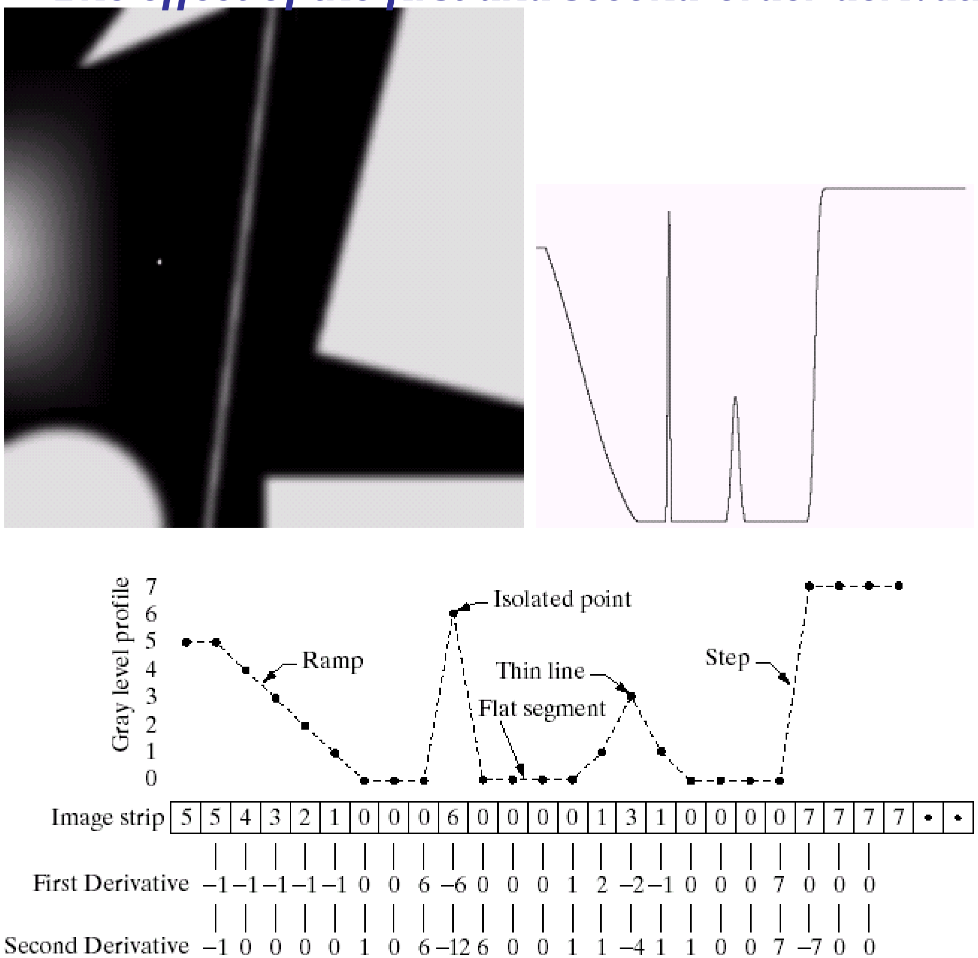


FIGURE 3.38 (a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point. (c) Simplified profile (the points are joined by dashed lines to simplify interpretation).

•First-order derivative

- Zero in flat segments.

- Nonzero at ramps.

•Second-order derivative

- Zero in Flat segments.

- Nonzero at the beginning and the end of ramps.

- Zero in along ramps.

Image Enhancement in *Spatial Domain*

Second-order Derivatives for Enhancement - The Laplacian

- *Second-order derivative is used to construct a Laplacian filter mask. Laplacian is an isotropic filter where the response of the filter is independent of the direction of the discontinuities in the image.*

- *Isotropic filters are rotation invariant, which means that if you rotate and filter the image or if you filter and then rotate the image you get the same result.*

- *Given an image f(x,y) the Laplacian operator is defined by:*

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- *The operator is a linear operator and can be expressed in discrete form in x-direction by:*

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

Image Enhancement in *Spatial Domain*

Second-order Derivatives for Enhancement - The Laplacian

- *The operator can be expressed in discrete form in y-direction by:*

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

- *Then,2-D Laplacian is:*

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

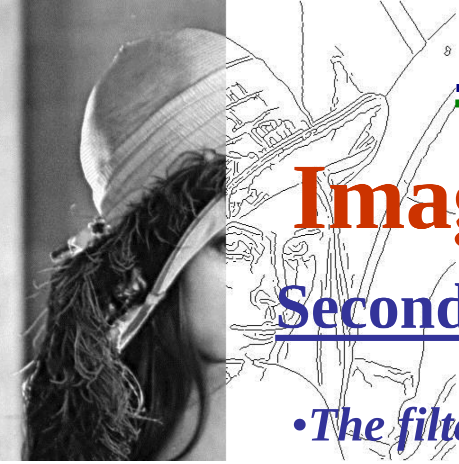


Image Enhancement in *Spatial Domain*

Second-order Derivatives for Enhancement - The Laplacian

• *The filter masks used to implement the digital Laplacian:*

considers x and y coordinates
Isotropic results for 90º

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

considers x, y and two diagonal coordinates.
Isotropic for 45º

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.

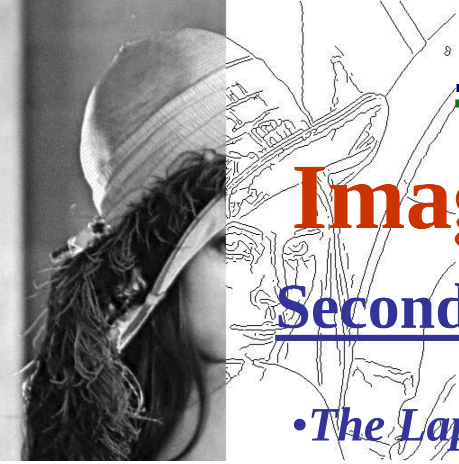


Image Enhancement in *Spatial Domain*

Second-order Derivatives for Enhancement - The Laplacian

- *The Laplacian operator highlights gray level discontinuities and de-emphasizes the slowly varying gray-levels.*
- *The result of Laplacian operator will give edge lines and other discontinuities on a dark and featureless background.*
- *The background features can be recovered and sharpening effect can be preserved by adding the Laplacian image to the original image.*
- *Depending on the choice of the Laplacian coefficients the following criteria is used for enhancement:*

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

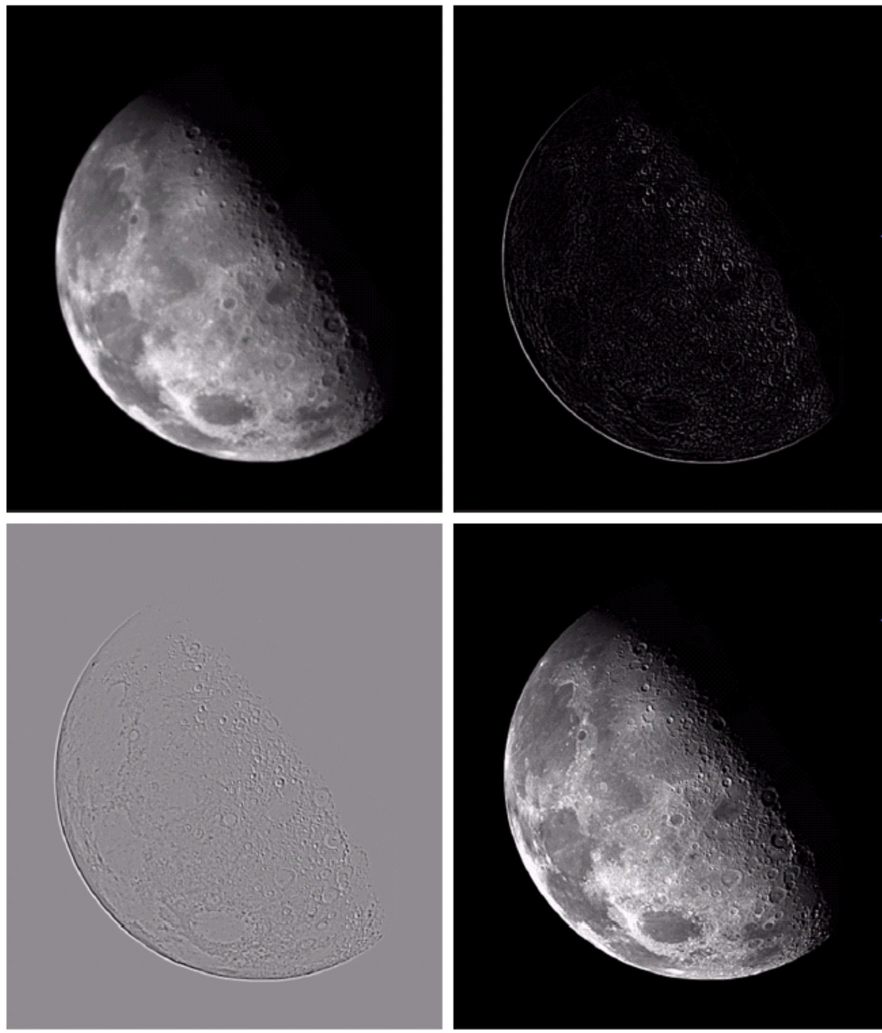
- *If the center coefficient of the Laplacian mask is negative*
- *If the center coefficient of the Laplacian mask is positive*

Image Enhancement in *Spatial Domain*

Second-order Derivatives for Enhancement - The Laplacian

a b
c d

FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)



Original image
$$f(x,y)$$

Laplacian image
$$\nabla^2 f(x,y)$$

Enhanced image
$$f(x,y)+\nabla^2 f(x,y)$$

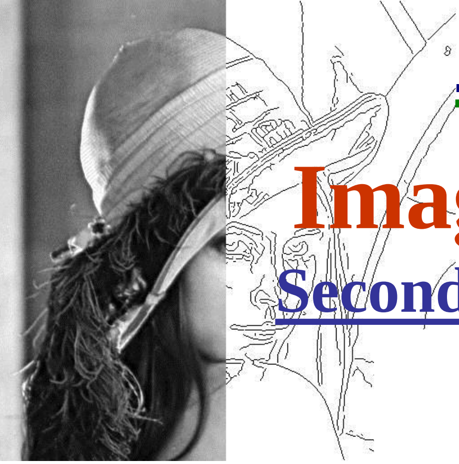


Image Enhancement in *Spatial Domain*

Second-order Derivatives for Enhancement - The Laplacian

Original image

Enhanced image using the mask with Center coefficient -8

Enhanced image using the mask with Center coefficient -4

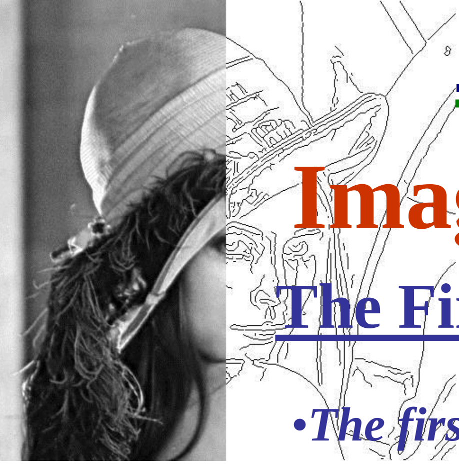


Image Enhancement in *Spatial Domain*

The First Derivatives for Enhancement - The Gradient

•The first derivatives in image processing are implemented by using the magnitude of the gradient.

• The gradient of f at coordinates (x,y) is defined by the two-dimensional column vector:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

•The magnitude of this vector, is referred to as the gradient, which is :

$$\nabla f = mag(\nabla \mathbf{f}) = \left[G_x{}^2 + G_y{}^2\right]^{1/2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

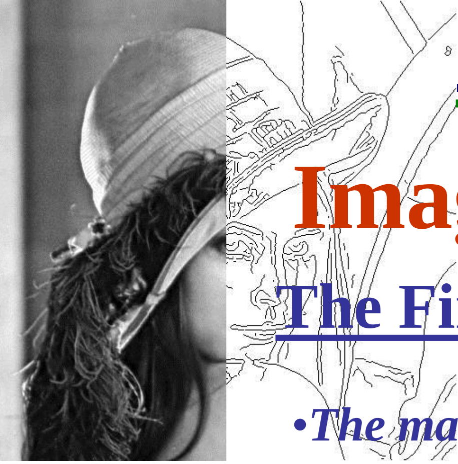


Image Enhancement in *Spatial Domain*

The First Derivatives for Enhancement - The Gradient

- *The magnitude of the gradient can be approximated by using the absolute values instead of squares and square roots, which is cheaper to compute and still preserves changes in the gray levels.*

$$\nabla f \approx |G_x| + |G_y|$$

- *If we consider a **3x3** filter mask then an approximation around the center pixel will be as follows:*

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$\nabla f \approx |G_x| + |G_y| = |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

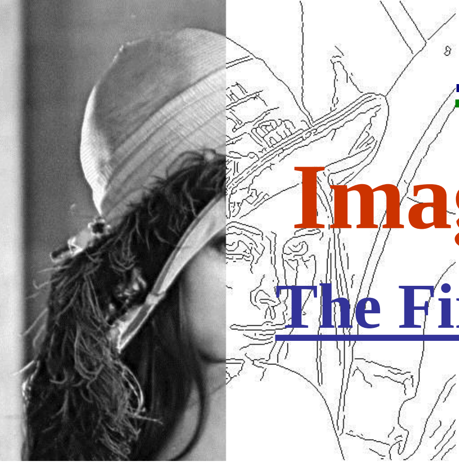


Image Enhancement in *Spatial Domain*

The First Derivatives for Enhancement - The Gradient

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

- ***The above masks are called the Sobel operators and can be used to implement gradient operation.***
- ***The idea behind using weight value of 2 is to achieve some smoothing by giving more importance to the center point.***
- ***The mask on the left approximates the derivative in x-direction (row 3- row 1).***
- ***The mask on the right approximates the derivative in y-direction (col 3- col 1).***

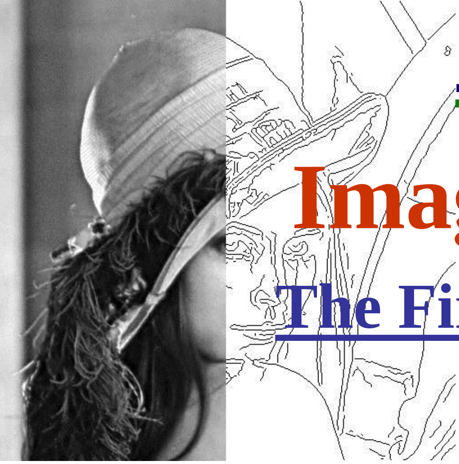
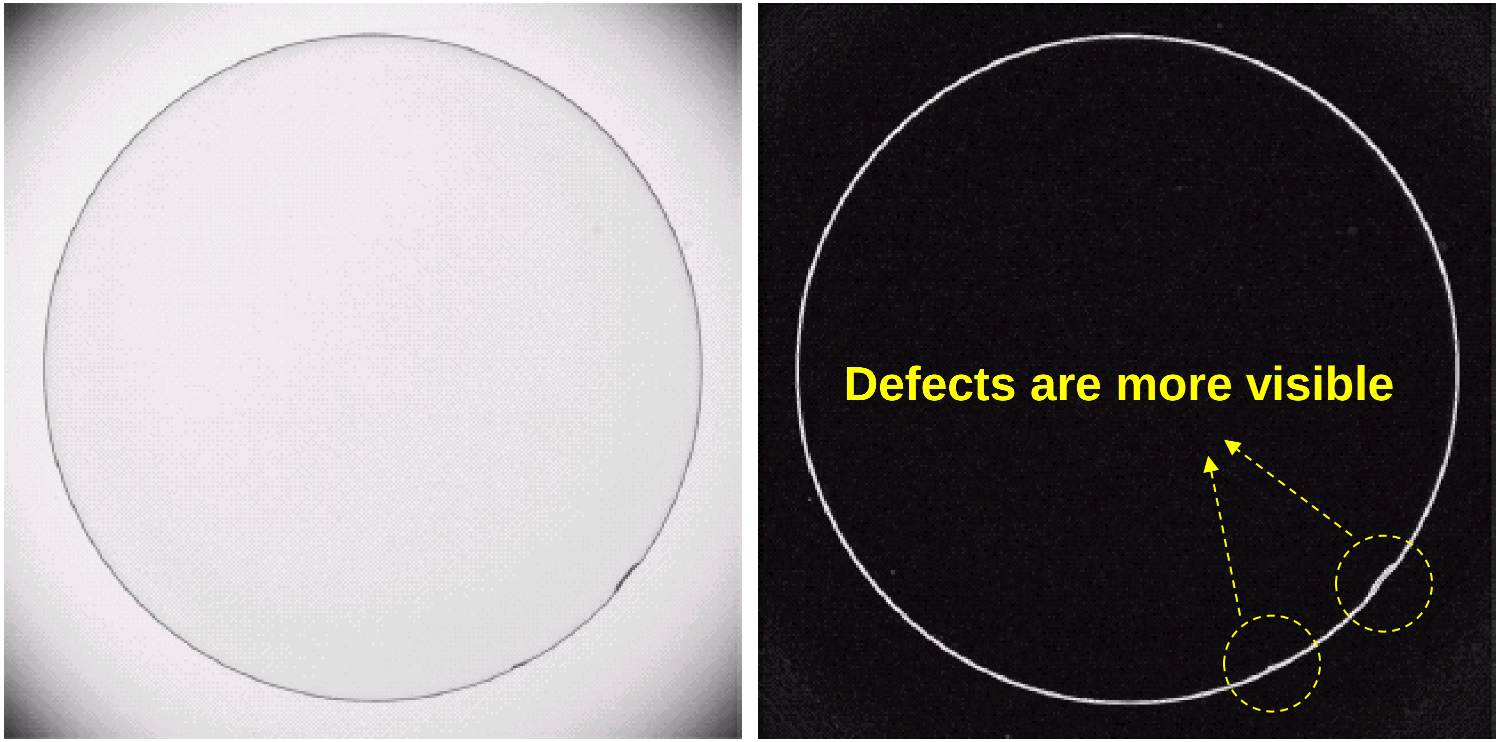


Image Enhancement in *Spatial Domain*

The First Derivatives for Enhancement - The Gradient



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

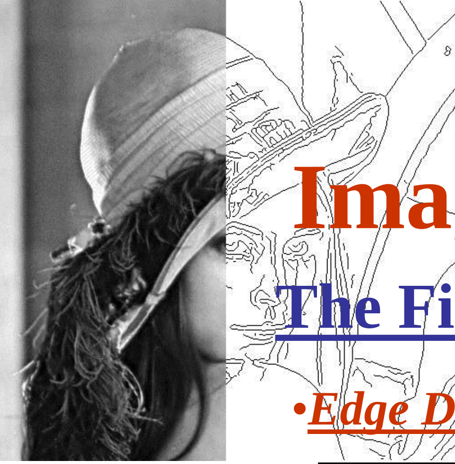
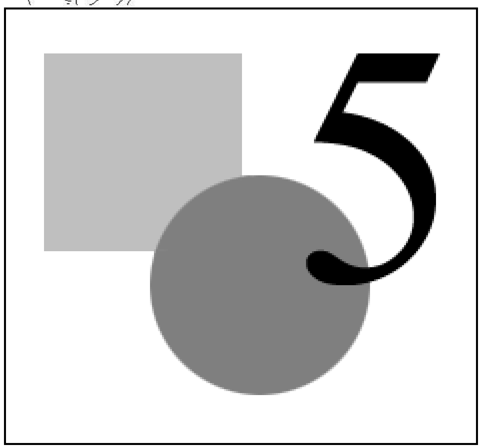


Image Enhancement in *Spatial Domain*

The First Derivatives for Enhancement - The Gradient

•***Edge Detection:*** *Consider the following image and the respective* ***sobel*** *operators*



−1	−2	−1
0	0	0
1	2	1

Horizontal Operator

−1	0	1
−2	0	2
−1	0	1

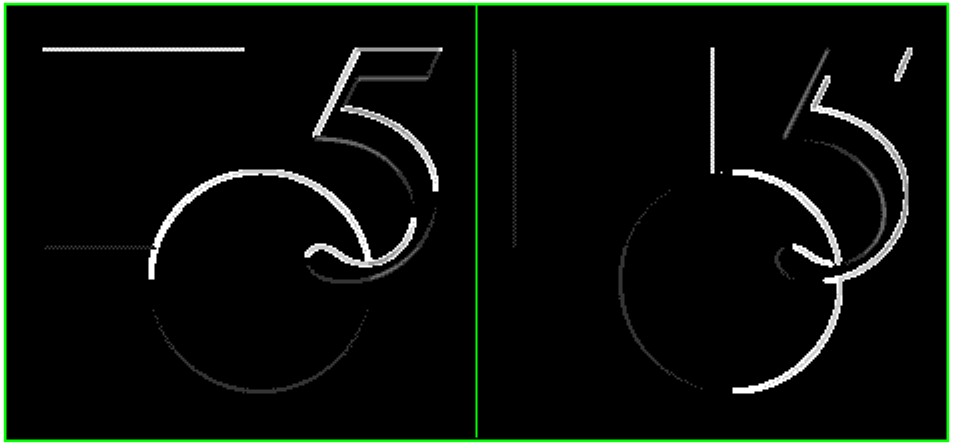
Vertical Operator

•*If the image is convolved(or correlated) by using the sobel operators given above. Then,*

Image Enhancement in *Spatial Domain*

The First Derivatives for Enhancement - The Gradient

• ***Edge Detection:***



g_h
Horizontal Sobel Operator highlights the horizontal edges

g_v
Vertical Sobel Operator highlights the vertical edges

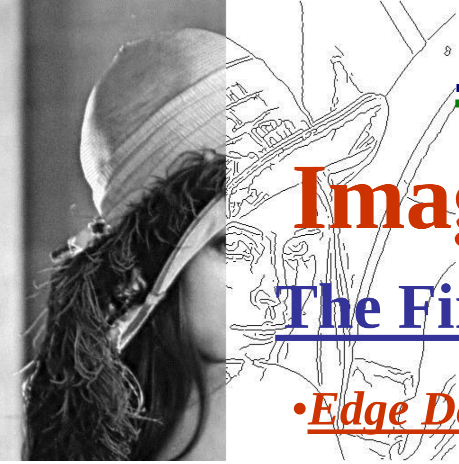
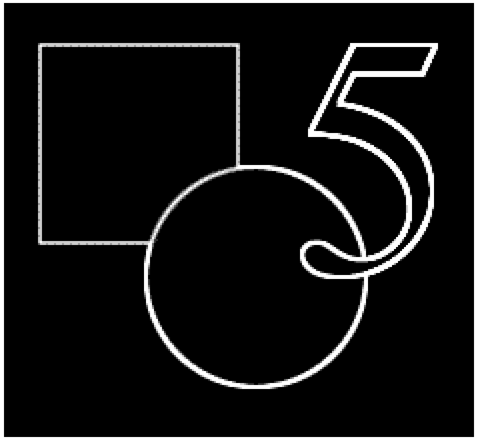


Image Enhancement in *Spatial Domain*

The First Derivatives for Enhancement - The Gradient

•***Edge Detection:***

•*The Gradient for each pixel can be defined to extract edges.*



$$g(x,y)=\sqrt{g(x,y)_h{}^2+g(x,y)_v{}^2}$$

•*Edges are detected by combining horizontal and vertical images obtained using respective Sobel operators.*