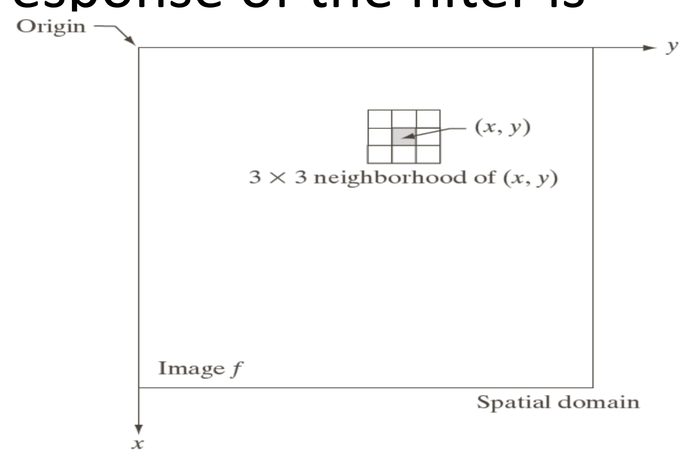
MT390 (DIP): Tutorial 3

Chapter 3:
Intensity Transformations and Spatial Filtering

3.4 Fundamentals of Spatial Filtering

- Filtering operations that are performed directly on the **pixels** of an image are called *spatial filtering to differentiate this type of process from the more traditional* **frequency** domain filtering.
- Spatial filters are **also called** spatial masks, kernels, templates, and windows.

The spatial filter mask is moved from point to point in an image. At each point (x,y), the response of the filter is calculated



The mechanics of spatial filtering

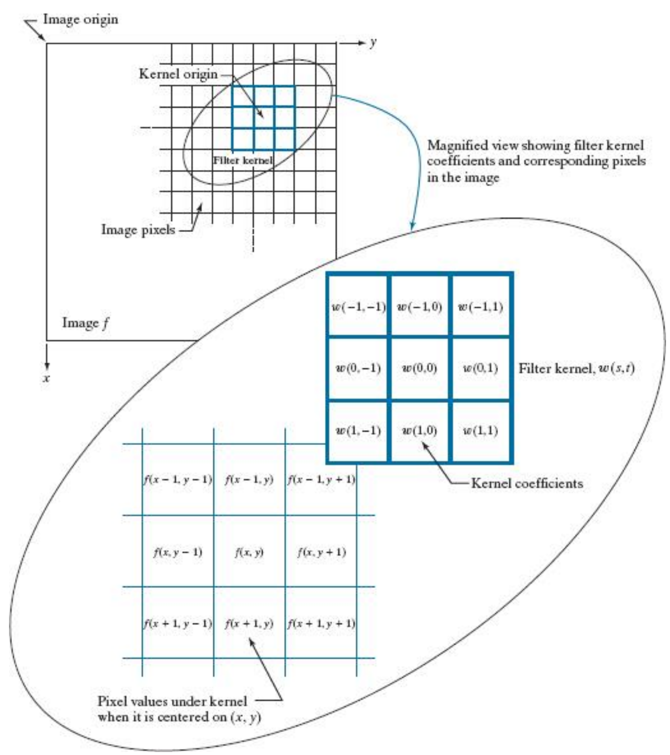
- The **mechanics** of spatial filtering are illustrated in **Fig. 3.28.**
- In general, **linear** filtering of an image f of size MxN with a filter mask of size mxn is given by the expression:

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

- For a mask of size mxn, we assume that m=2a+1 and n=2b+1, where a and b are positive integers.

Figure 3.28

The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.



Linear Spatial Filtering

Linear Spatial Filtering Ops:

- The Linear **Operations** consist of **multiplying** each pixel in the neighborhood by a corresponding Coefficient.
- Results are then **summed** to obtain the response at each point (x,y)

Number of Coefficients Required:

If the neighborhood is of size mxn, then mn coefficients are required,.

Terminology:

The **coefficients** are arranged as a matrix called:

- Filter
- Mask
- Filter Mask
- Kernel
- Template
- Window

Also called:

- Convolution filter
- Convolution mask
- Convolution kernel

Spatial Correlation and Convolution

- **Correlation:** It is the process of moving a filter mask over the image and computing the sum of products at each location.

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

- **Convolution**: It is the same as above except that the filter mask is rotated by 180 degrees.

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

1-D

Correlation	Convolution
Origin f w	Origin f w rotated 180°
(a) 0 0 0 1 0 0 0 0 1 2 3 2 8	0 0 0 1 0 0 0 0 8 2 3 2 1 (i)
(b) 0 0 0 1 0 0 0 0 1 2 3 2 8 Starting position alignment	0 0 0 1 0 0 0 0 (j) 8 2 3 2 1
Zero padding (c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k) 8 2 3 2 1
(d) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8 Position after one shift	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l) 8 2 3 2 1
(e) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8 Position after four shifts	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m) 8 2 3 2 1
(f) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 2 3 2 8 Final position	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n) 8 2 3 2 1
Full correlation result (g) 0 0 0 8 2 3 2 1 0 0 0 0	Full convolution result 0 0 0 1 2 3 2 8 0 0 0 0 (o)
Cropped correlation result (h) 0 8 2 3 2 1 0 0	Cropped convolution result 0 1 2 3 2 8 0 0 (p)

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*.

Let y= [2 1 1 2 1 1 0] and h= [1 1 0];

Find x as convolution of y and h. **show your intermediate steps**.

Solution:

First, we need to invert h to become h^1; h^1= [0 1 1]

The steps for finding the convolution of y and h are as follows:

```
Step1:      2 1 1 2 1 1 0
        0 1 1                    → 2

Step 2:     2 1 1 2 1 1 0
          0 1 1                  → 3

Step 3:     2 1 1 2 1 1 0
            0 1 1                → 2

Step4:      2 1 1 2 1 1 0
              0 1 1              → 3

Step5:      2 1 1 2 1 1 0
                0 1 1            → 3

Step6:      2 1 1 2 1 1 0
                  0 1 1          → 2

Step7:      2 1 1 2 1 1 0
                    0 1 1        → 1

Step8:      2 1 1 2 1 1 0
                      0 1 1      → 0

Step9:      2 1 1 2 1 1 0
                        0 1 1    → 0
```

Final answer: x = [2 3 2 3 3 2 1 0 0]

Table 3.5

Some fundamental properties of convolution and correlation. A dash means that the property does not hold.

Property	Convolution	Correlation
Commutative	$f \star g = g \star f$	–
Associative	$f \star (g \star h) = (f \star g) \star h$	–
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \,☆\, (g + h) = (f \,☆\, g) + (f \,☆\, h)$

Vector representation of linear filtering

- When interest lies in the response, *R, of an mxn mask at any point (x, y),* it is common practice to simplify the notation by using the following expression:

$$
\begin{aligned}
R &= w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} \\
&= \sum_{i=1}^{mn} w_i z_i
\end{aligned}
$$

In Matrix form:

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

Generating Spatial Filter Masks

- Generating mxn Linear Spatial Filter requires that we **specify mn mask coefficients**.
- These coefficients are selected based on **what the filter does.**
- E.g: 3x3 **averaging** filter

$$R = \frac{1}{9}\sum_{j=1}^{9} z_j$$

- Each coefficient value is **1/9.**

3.5 Smoothing Spatial Filters

Used for:

1. Blurring: removing small details
2. Noise Reduction

Smoothing Linear Filters (low pass)

- **How it works?** The value of every pixel is replaced by the average of the gray levels in the neighborhood. In other words, the output (response) of a smoothing, linear spatial filter is simply the **average** of the pixels contained in the neighbourhood of the filter mask.
- These filters sometimes are called **averaging** filters. They also are referred to a **low pass** filters.

- **Type of smoothing filters:**

1. Standard average
2. weighted average.

linear

3. Median filter

Order statistics

Low Pass Filters (LPFs):

What is a LPF?

- Allows Low frequencies to pass
- Attenuates High Frequencies (edge in images)

Advantages:

- Noise removal
- Useful for certain object detection purposes

Disadvantages:

- Destroys edge information in images.

Spatial Domain Filtering/Masking Procedure:

- Select an appropriate Filter Mask.
- Imagine the Center of the Mask on top of the Pixel which needs to be processed
- Perform Element by Element multiplication of Image Pixels values and the corresponding mask/filter coefficients.
- Add the results.

Smoothing Filter: Standard Average

Standard Average Spatial Filter Mask:

- Produces Smoothing Effect
- Low Pass Filtering Effect
- The Mask is given by:

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

➢ A spatial averaging filter in which all coefficients are **equal** is sometimes called a ***box filter***.

Examples of smoothing kernels: (a) is a box kernel; (b) is a Gaussian kernel.

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

$$\frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

Figure 3.33

(a) Test pattern of size 1024×1024 pixels. (b)–(d) Results of lowpass filtering with box kernels of sizes $3\times3, 11\times11,$ and $21\times21,$ respectively.

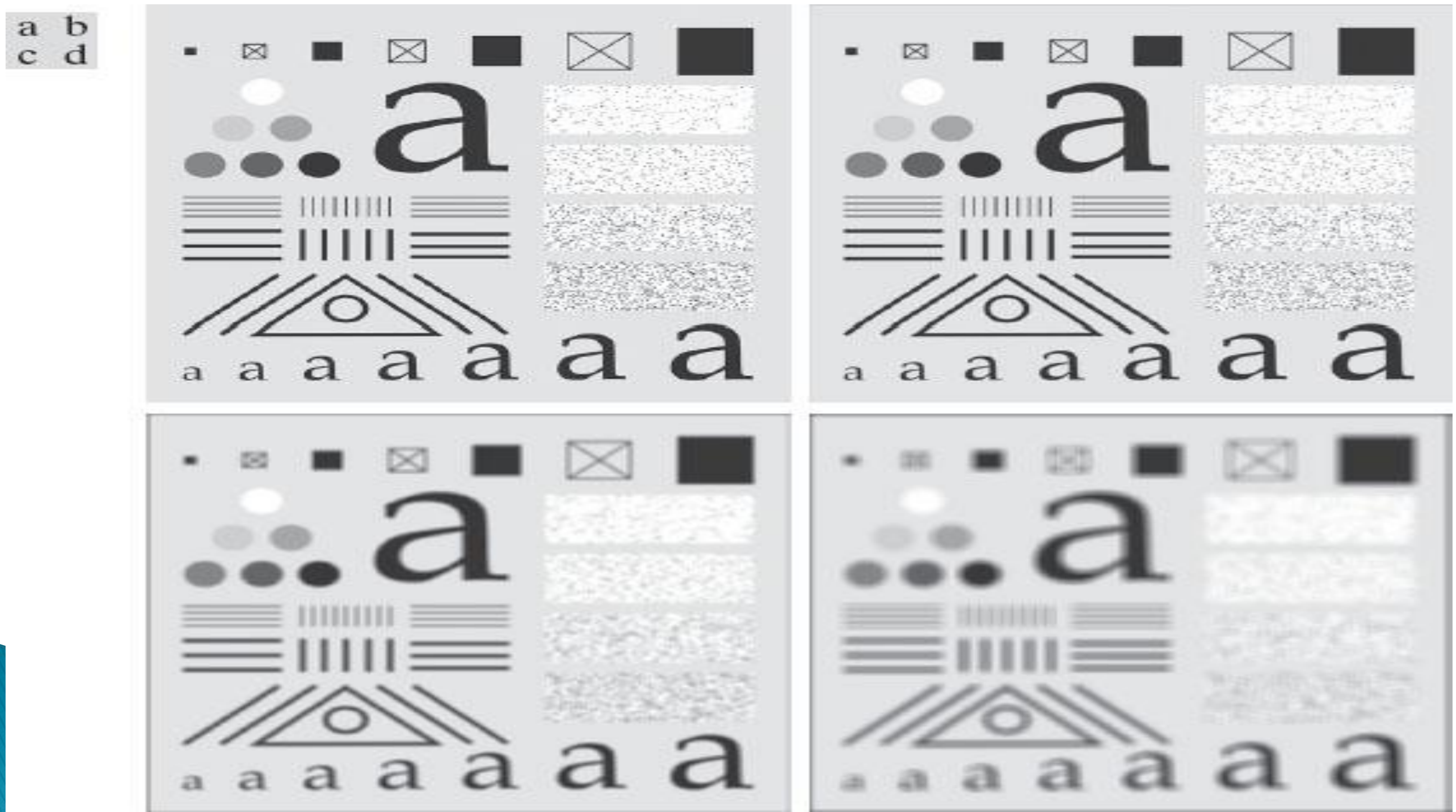


Figure 3.35

(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for $K = 1$ and $\sigma = 1$.

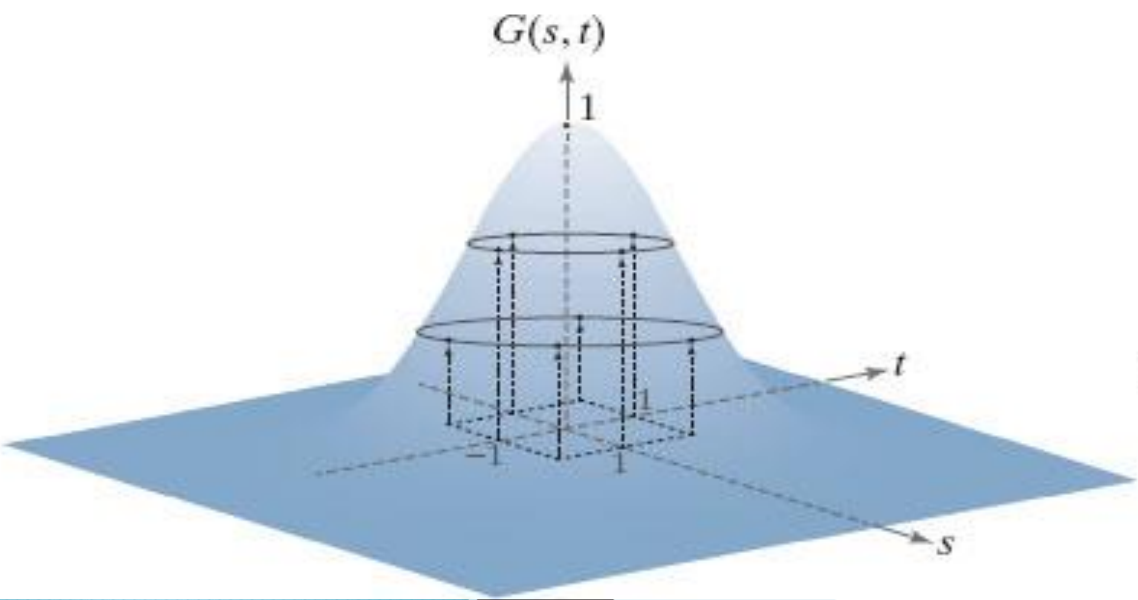
(b) Resulting 3×3 kernel [this is the same as Fig. 3.3(b)].

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

By letting $r = [s^2 + t^2]^{1/2}$ we can write Eq. (3-45) as

$$G(r) = Ke^{-\frac{r^2}{2\sigma^2}}$$

a b



$\frac{1}{4.8976} \times$

0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

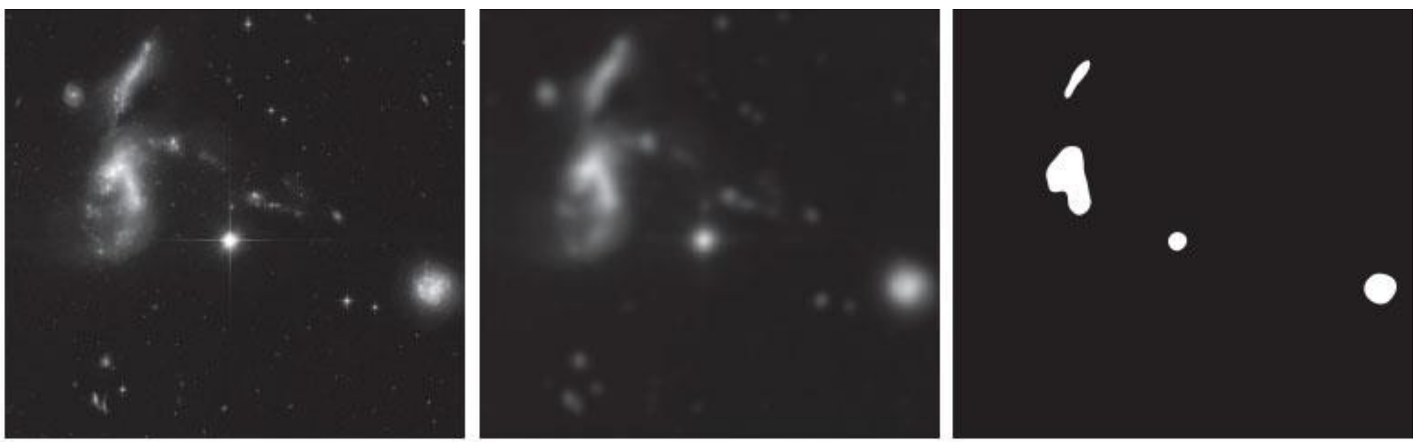
Figure 3.42

(a) A test pattern of size 1024×1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21×21. with standard deviations $\sigma = 3.5$ (c) Result of using a kernel of size 43×43, with $\sigma = 7$. This result is comparable to Fig. 3.39(d). We used K = 1 in all cases.

a b c

Figure 3.16

(a) A 2566×2758, Hubble Telescope image of the Hickson Compact Group. (b) Result of lowpass filtering with a Gaussian kernel. (c) Result of thresholding the filtered image (intensities were scaled to the range [0, 1]). The Hickson Compact Group contains dwarf galaxies that have come together, setting off thousands of new star clusters. (Original image courtesy of NASA.)



a b c

Weighted Average Spatial Filter Mask:

- Produces Smoothing Effect
- Low Pass Filtering Effect
- The Mask values represent Filter Coefficients
- The Mask is given by :

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

Spatial filters : Smoothing Standard and weighted Average- example

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

The mask is moved from point to point in an image. At each point (x,y), the response of the filter is calculated

Standard averaging filter:

(110 +120+90+91+94+98+90+91+99)/9 =883/9 = 98.1

Weighted averaging filter:

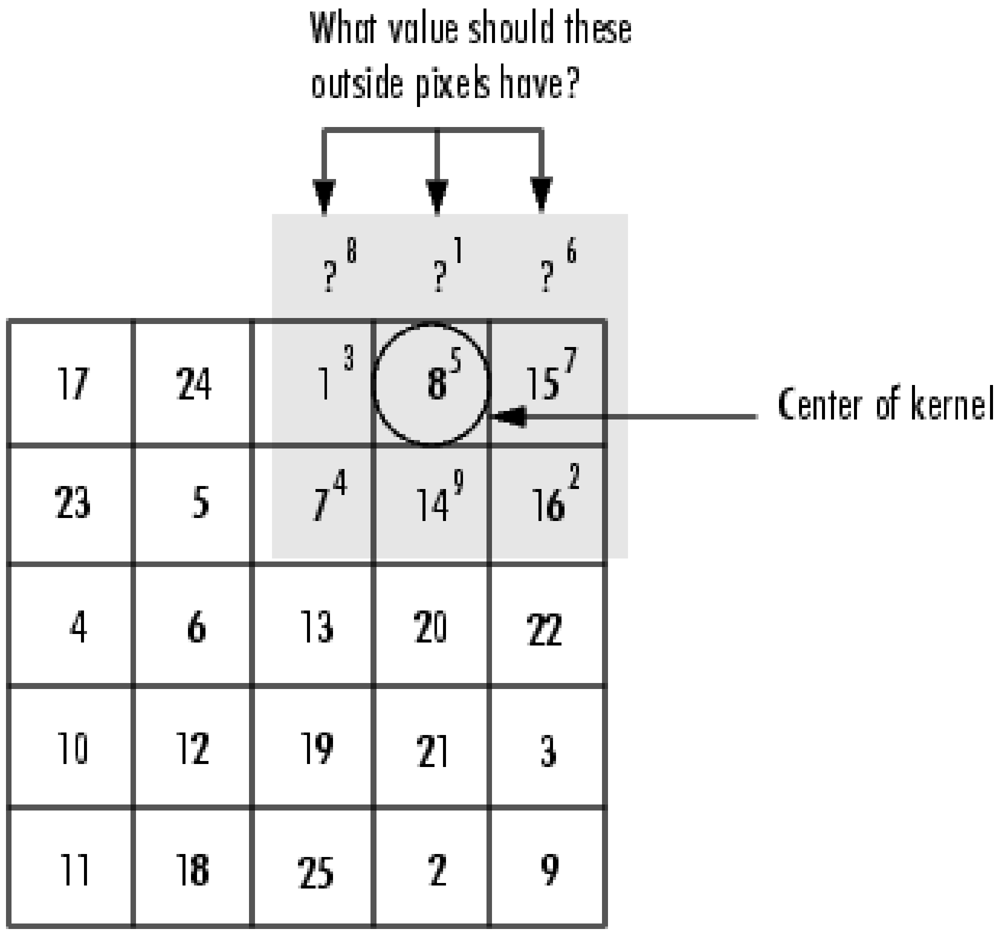
(110 +2 x 120+90+2 x 91+4 x 94+2 x 98+90+2 x 91+99)/16=97.81

General implementation:

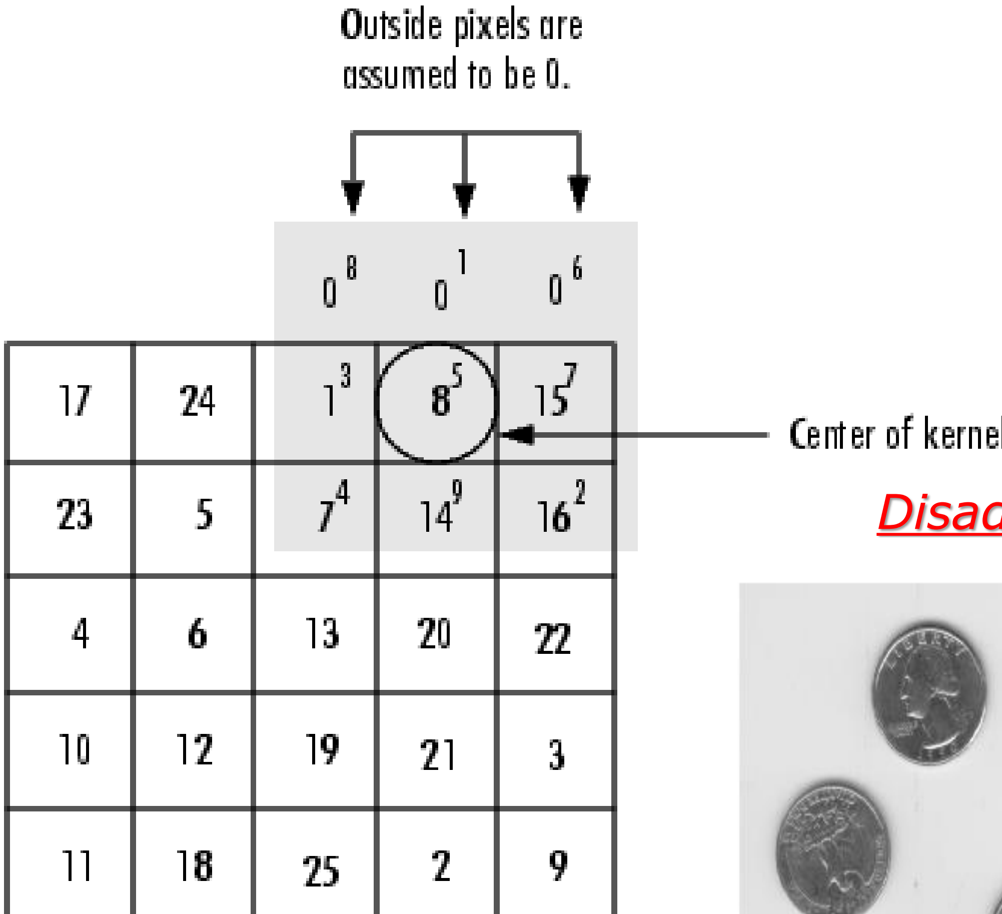
General implementation for filtering an MxN image with a weighted averaging filter of size mxn (m and *n odd) is* given by the expression:

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

What happens when the Values of the Kernel Fall Outside the Image?



First solution :Zero padding:

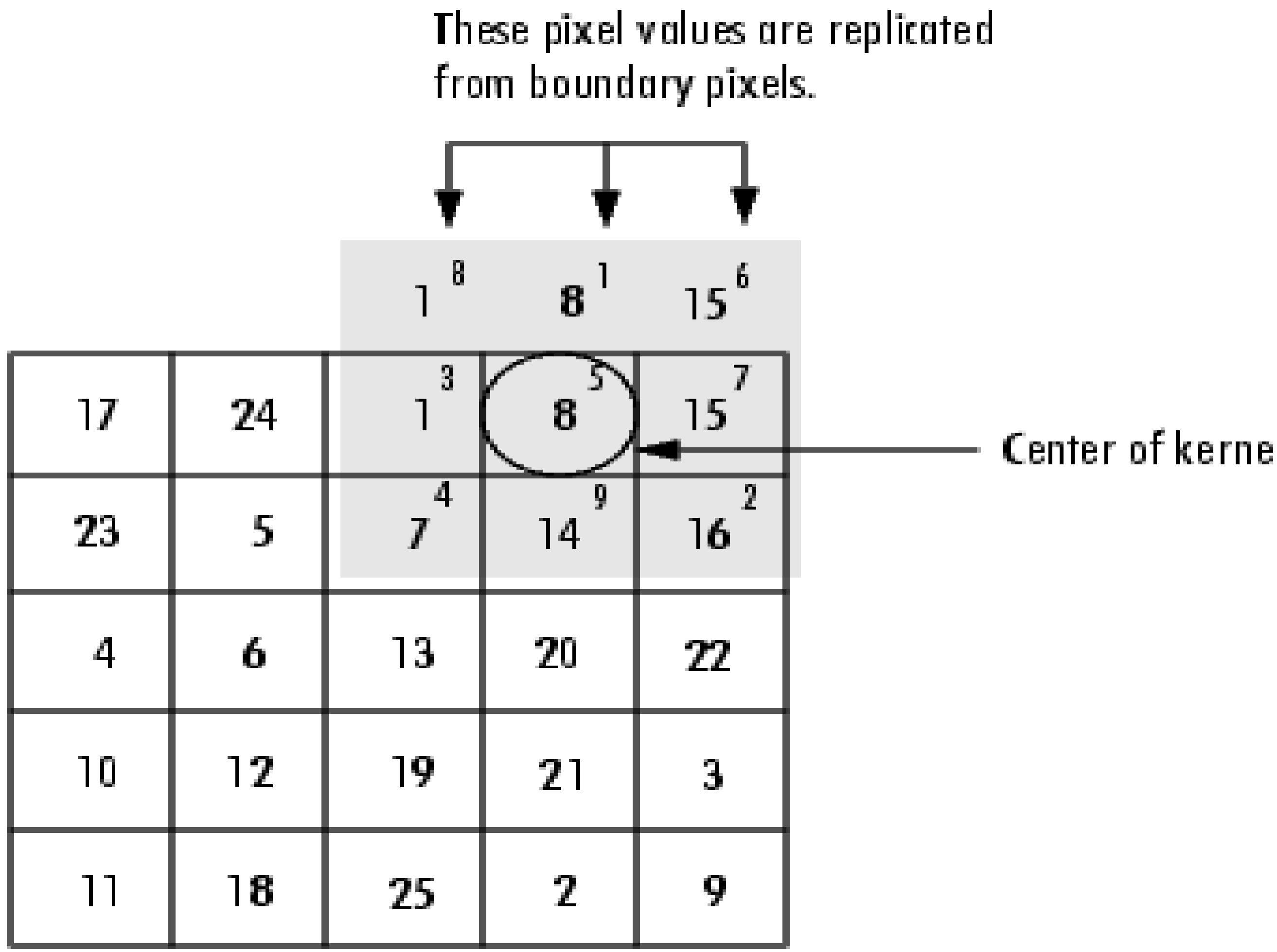


<u>Disadvantage: black border</u>

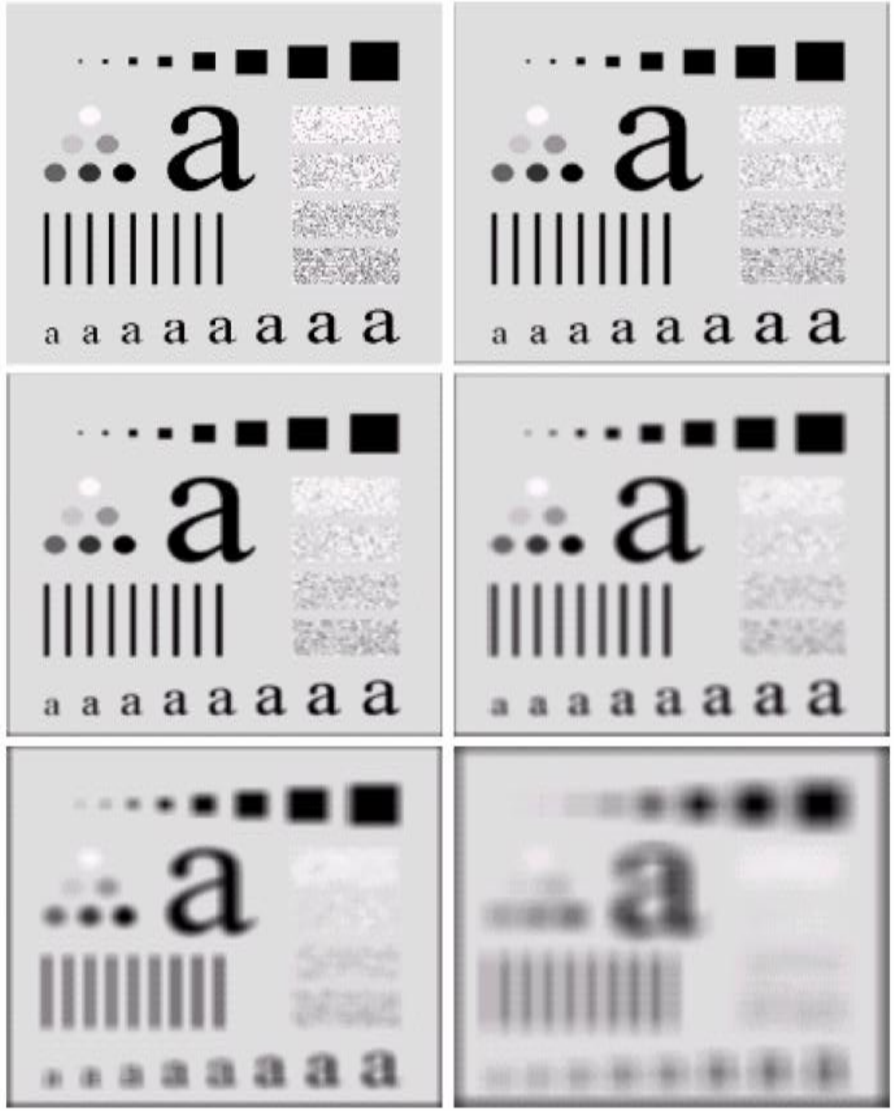
Original Image

Filtered Image with Black Border

Second solution :border padding:



Example:



Original image

3 x 3 averaging

5 x 5 averaging

9 x 9 averaging

15 x 15 averaging

35 x 35 averaging

Order-Statistics (nonlinear) Filters

- Response is based on ordering (**ranking**) the pixels and then replacing the value of the center pixel with the value determined by the ranking result.
- The best-known example in this category is the ***median*** *filter, which, as its name implies, replaces the value of a pixel by the* median of the gray levels in the neighbourhood of that pixel
- Median filters are quite **popular** because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size.
- Median filters are particularly effective in the presence of *impulse noise, also called* ***salt-and-pepper noise*** *because of its* appearance as white and black dots superimposed on an image.

Spatial filters: Smoothing order statistics: Median filter

110	120	90	130
91	94	98	200
90	95	99	100
82	96	85	90

$\rightarrow \boxed{95}$

Steps:

1. Sort the pixels in ascending order:

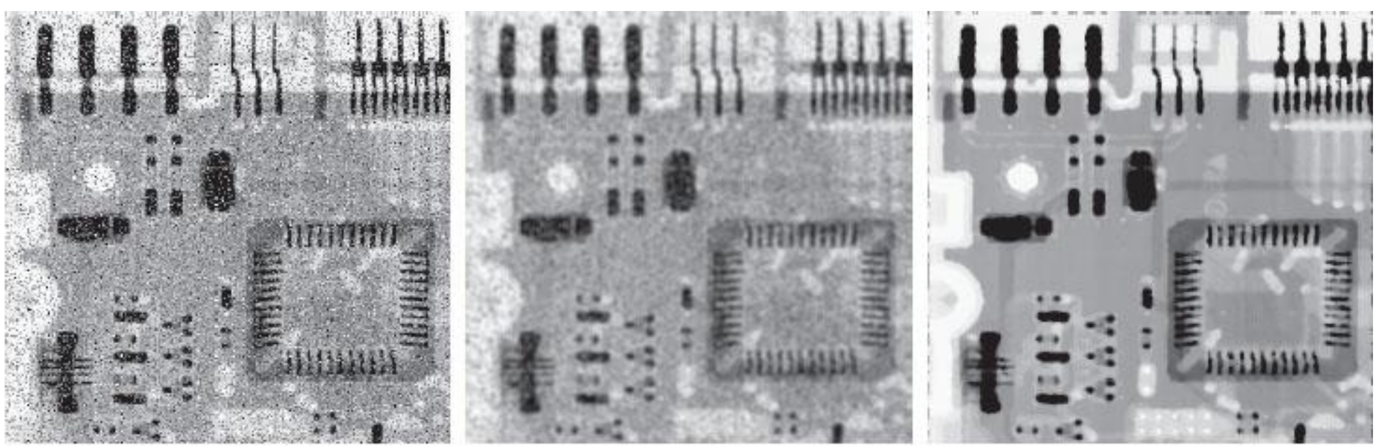
90,90, 91, 94, 95, 98, 99, 110, 120

2. replace the original pixel value by the median : 95

Figure 3.43

(a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma = 3$. (c) Noise reduction using a 7×7 median filter.

(Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



a b c

3.6 Sharpening Spatial Filters: (High Pass)

- The principal **objective** of sharpening is to highlight **transition** in intensity.
- Sharpening is accomplished by spatial **differentiation**.
- Image differentiation **enhances edges** and other discontinuities (such as noise) and deemphasizes areas with slowly varying gray-level values

High Pass Filters (HPFs):

What is a HPF?

- Allows High frequencies to pass
- Attenuates Low Frequencies

Advantages:

- Edge detection in images

Disadvantages:

- Noise amplification

Examples on High pass filter

Input Image

Result of Basic Lap Mask

Output image

Foundation

- The **derivatives** of a digital function are defined in terms of **differences**
- Definitions of the **first and 2nd-order** derivatives of a 1-D function f(x) are the differences:

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

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

Comparing first- and second-order derivatives:

(1) First-order derivatives generally produce thicker edges in an image.

(2) Second-order derivatives have a **stronger response to fine detail, such as thin lines** and isolated points.

(3) First order derivatives generally have a stronger response to a gray-level step.

(4) Second-order derivatives produce a **double respons**e at step changes in gray level.

Using the Second Derivative for Image Sharpening–The Laplacian

- The Laplacian, for a function (image) f(x, y) of two variables, is defined and given below:

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

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

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

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

Figure 3.45 Filter Masks

(a) Laplacian kernel used to implement Eq. (3-53). (b) Kernel used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other Laplacian kernels.

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

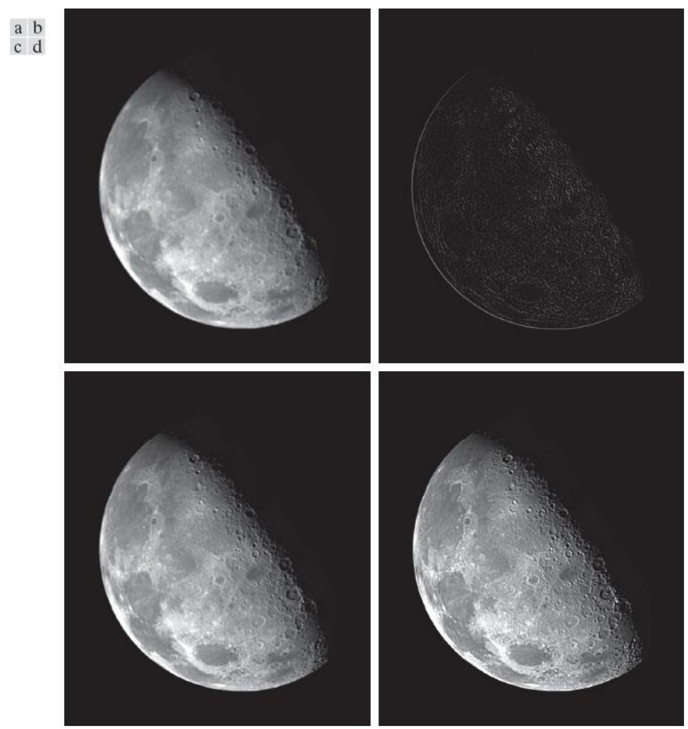
a b c d

Using the Laplacian for image enhancement is as follows:

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$

Figure 3.46

(a) Blurred image of the North Pole of the moon. (b) Laplacian image obtained using the kernel in Fig. 3.45(a). (c) Image sharpened using Eq. (3-63) with $c = -1$. (d) Image sharpened using the same procedure, but with the kernel in Fig. 3.45(b). (Original image courtesy of NASA.)



Example: apply the following Laplace on the highlighted pixel:

0	−1	0
−1	4	−1
0	−1	0

153	157	156	153	155
159	156	158	156	159
155	158	**154**	156	160
154	157	158	160	160
157	157	157	156	155

154x4 – 158- 156-158-158 = -14

So the value after filter = -14

We call the resultant image: **sharpened image.**

Filtered image=original + sharpened image

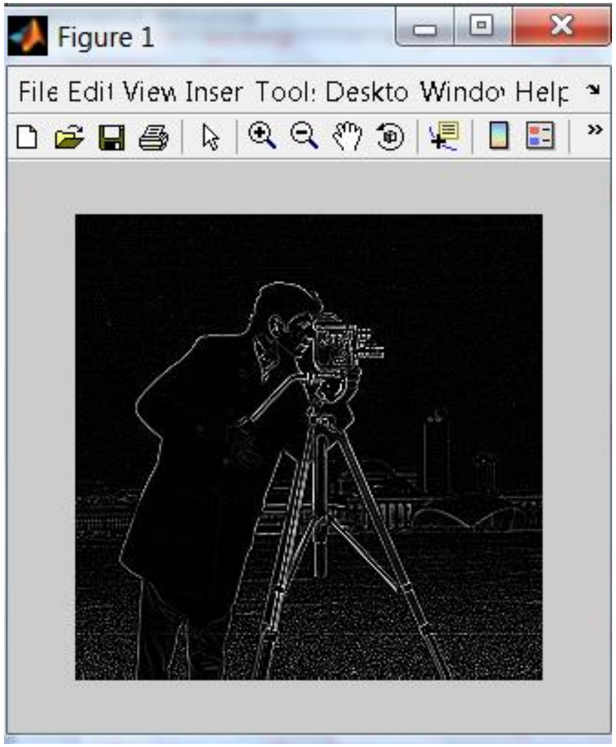
(+ : because: center coefficient of Laplacian mask is positive)

The value in the filter image=154-14 =140

Examples of Laplacian operator:

Original image

Laplace Sharpened image



Laplace filtered image

Un-sharp masking and high-boost filtering

- To sharpen images by **subtracting** a blurred or **un-sharp (smoothed)** version of an image from the original image itself.
- This process, called un-sharp masking, **consists of:**

  1. Blur the original image.

  2. Subtract the blurred image from the original image (the resulting difference is called the mask).

  3. Add the mask to the original.

Unsharp masking and Highboost filtering expressions:

- Let $\bar{f}(x, y)$ denote the **blurred** image.
- The **mask** is obtained as:

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

- A weighted portion of mask is added to original:

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

- **Unsharp Making**: when k=1
- **Highboost filtering**: when $k > 1$

a
b
c
d
e

FIGURE 3.40
(a) Original image.
(b) Result of blurring with a Gaussian filter.
(c) Unsharp mask. (d) Result of using unsharp masking.
(e) Result of using highboost filtering.

Figure

(a) Unretouched "soft-tone" digital image of size 469×600 pixels

(b) Image blurred using a 31×31 Gaussian lowpass filter with $\sigma = 5$.

(c) Mask. (d) Result of unsharp masking using Eq. (3-56) with $k = 1$. (e) and (f) Results of highboost filtering with $k = 2$ and $k = 3$, respectively.

a b c
d e f

Using First Order Derivatives for (Nonlinear) Image Sharpening—The Gradient

- For a function f(x, y), the **gradient** of *f at coordinates (x, y) is defined* as the two-dimensional column *vector*

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

- *The **magnitude** (length) of above vector is:*

$$\begin{aligned} &= \operatorname{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}. \end{aligned}$$

- *The masks are given in next slide.*

Figure 3.50

(a) A 3×3 region of an image, where the z_s are intensity values. (b)–(c) Roberts cross-gradient operators. (d)–(e) Sobel operators. All the kernel coefficients sum to zero, as expected of a derivative operator.

a
b c
d e

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

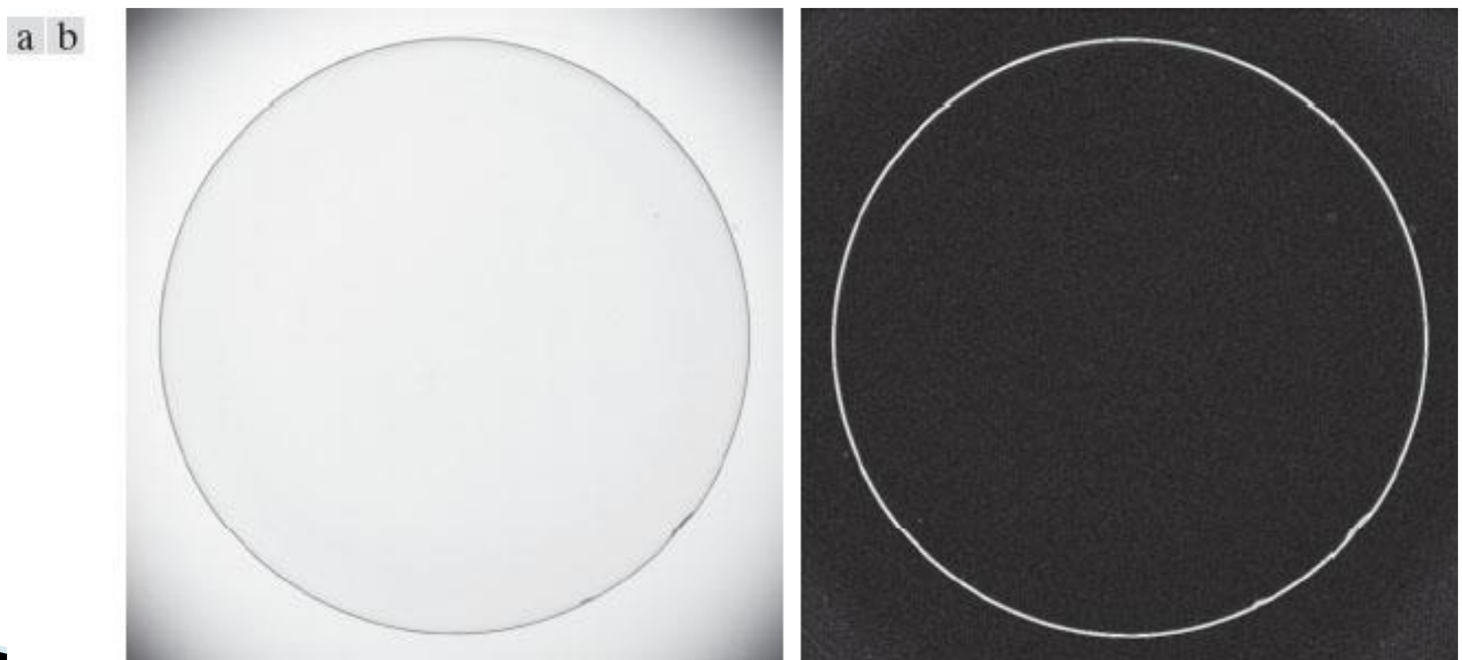
0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

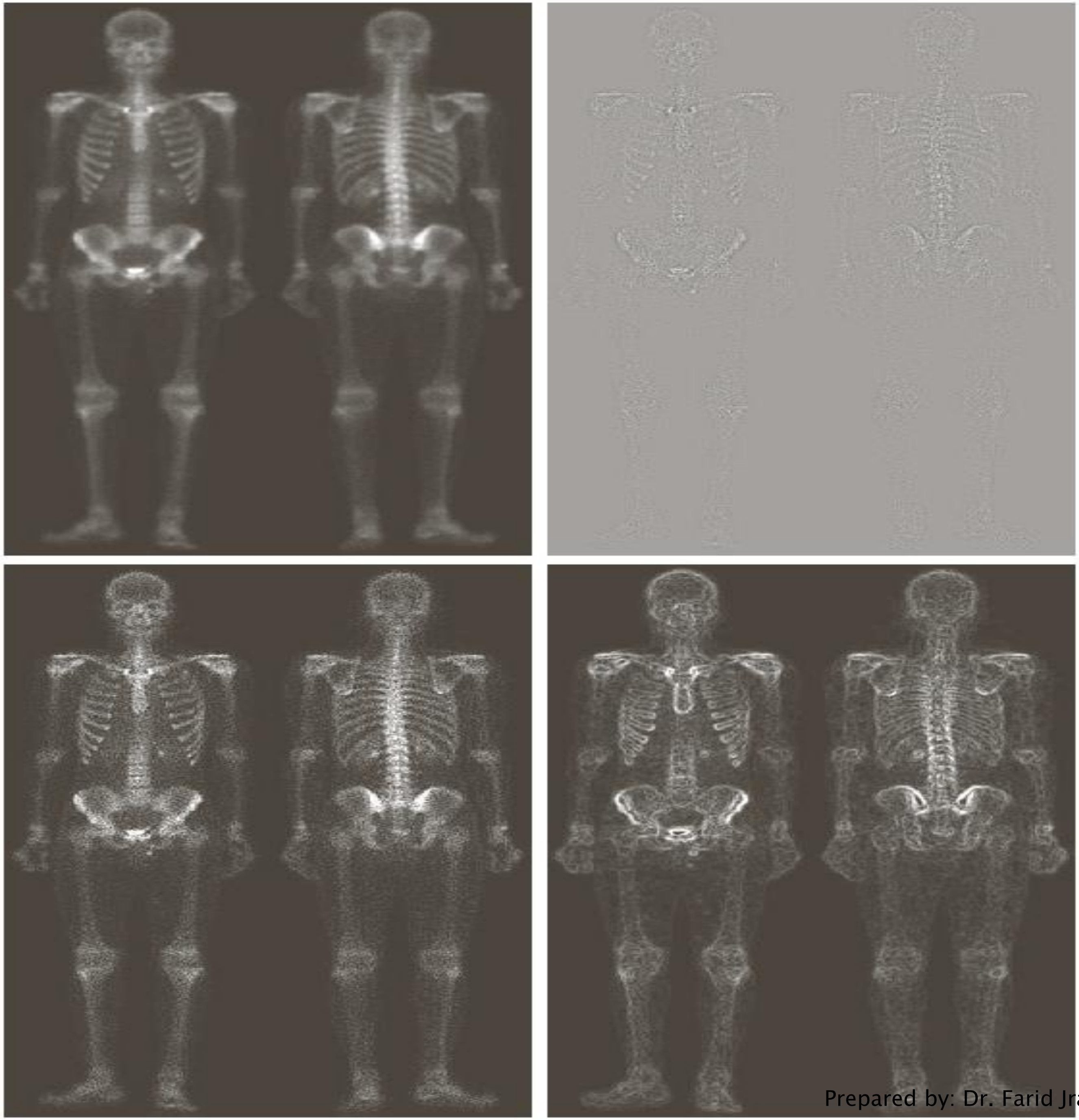
Figure 3.51

(a) Image of a contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Perceptics Corporation.)



3.8 Combining Spatial Enhancement Methods

- **Fig. 3.43(a)** is a nuclear whole body bone scan image
- Our **objective** is to enhance this image by sharpening it and by bringing out more of the skeletal detail.
- The narrow dynamic range of the gray levels and high noise content make this image **difficult** to enhance.
- The **strategy** we will follow is:
  - to utilize the **Laplacian** to highlight fine detail, and
  - the **gradient** to enhance prominent edges.
  - a smoothed version of the gradient image will be used to mask the Laplacian image
  - Finally, we will attempt to increase the dynamic range of the gray levels by using a **gray-level transformation**.



a b
c d

FIGURE 3.43
(a) Image of whole body bone scan.
(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b).
(d) Sobel gradient of (a).

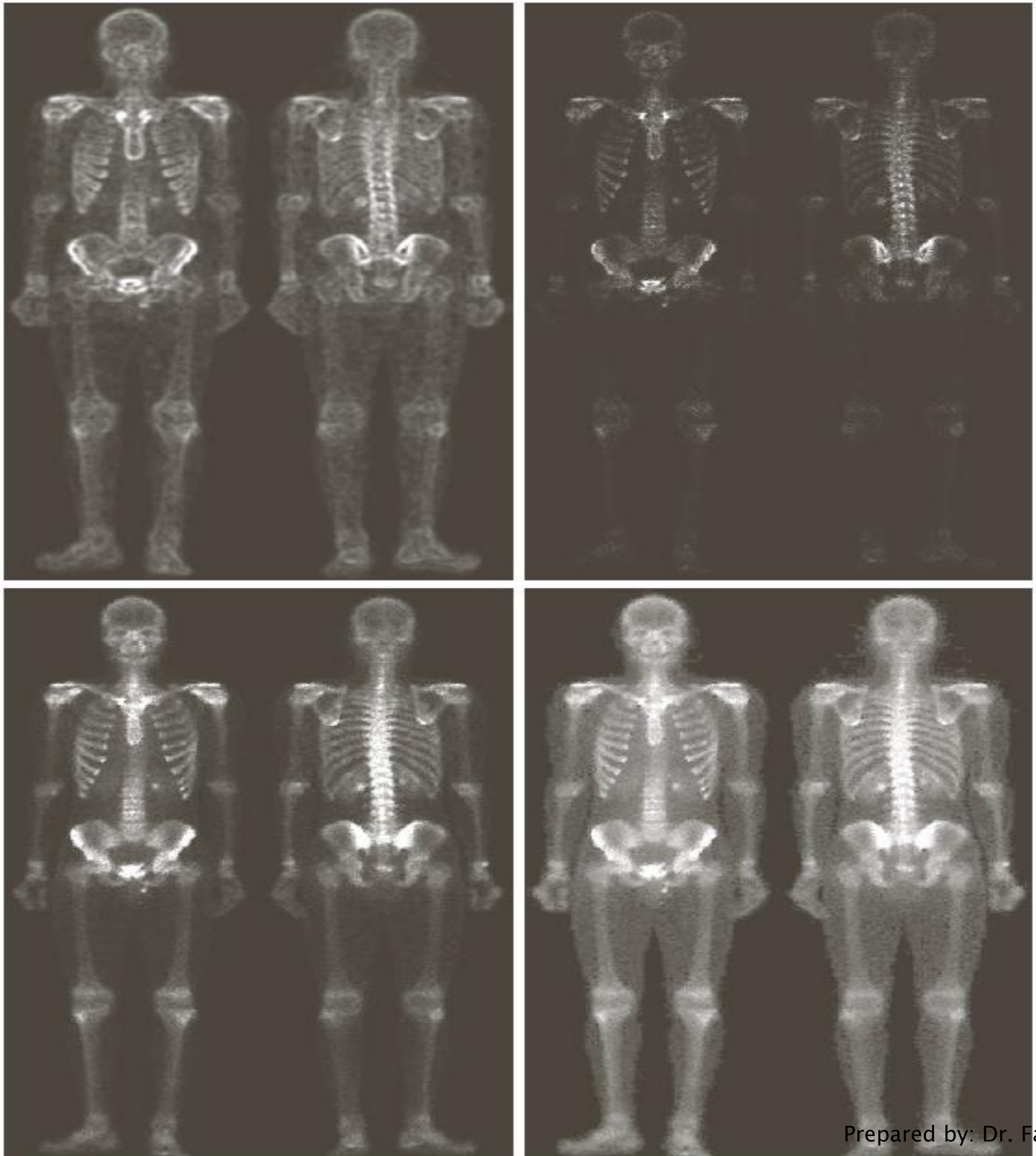


FIGURE 3.43 *(Continued)* (e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

Lab Hints:

Matlab help on following image related functions:

- Histeq
- adapthisteq
- imadjust
- imnoise
- medfilt2
- fspecial,
- filter2,
- Imfilter etc