IMAGE- PROCESSING CSE395

Module-2

2.0 &2.1–Image Transformation: Some Basic Gray Level Transformations

PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

Private University Estd. in Karnataka State by Act No. 41 of 2013

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

LAKSHMISHA S KRISHNA

Image Acquisition Process

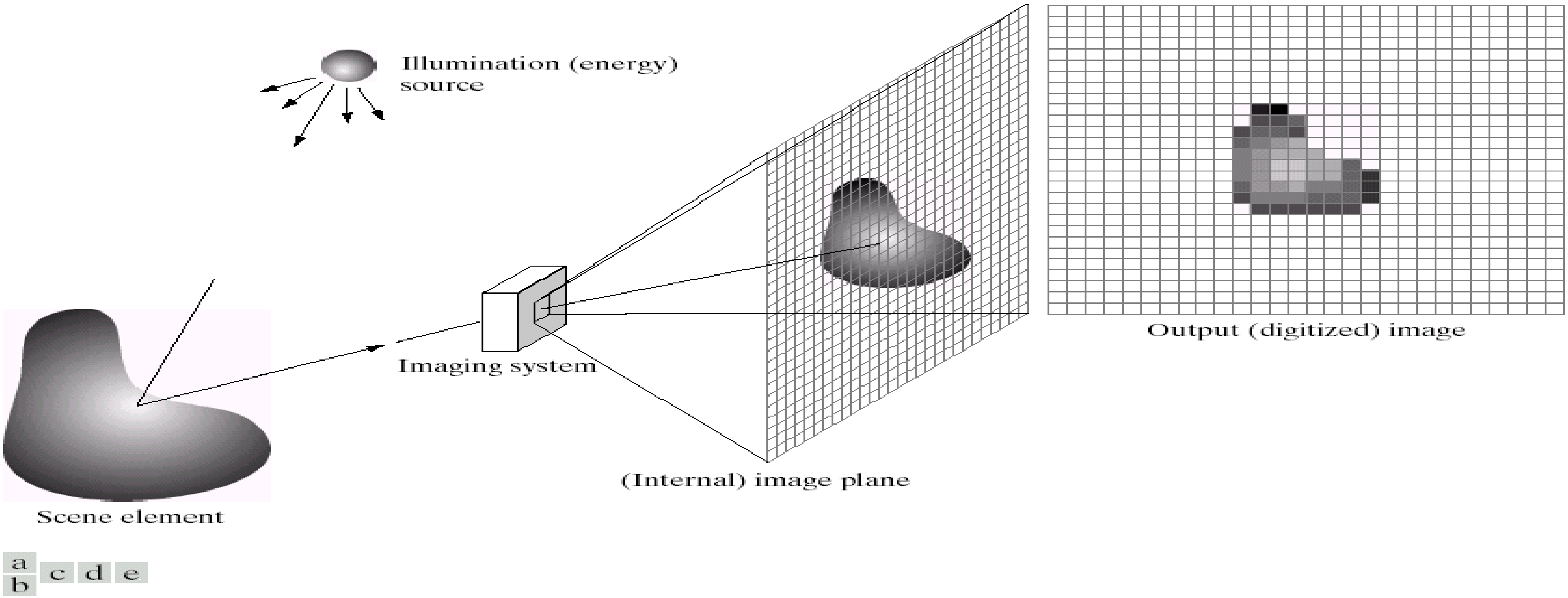


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Image Sampling and Quantization

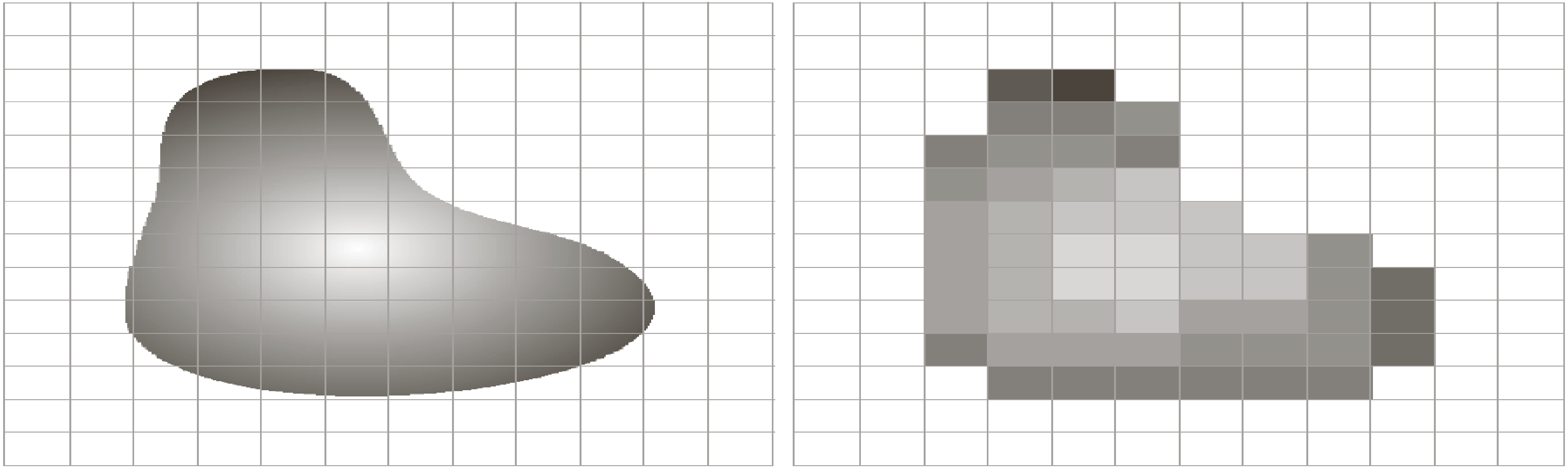
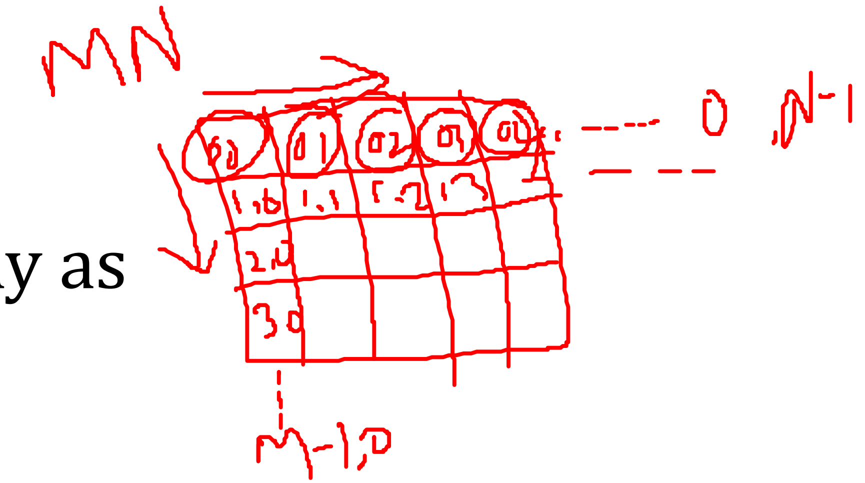


FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Representing Digital Images



- The representation of an M×N numerical array as

$$\boxed{f(x,y)} = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Representing Digital Images

- The representation of an M×N numerical array as

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N-1} \\ \dots & \dots & \dots & \dots \\ a_{M-1,0} & a_{M-1,1} & \dots & a_{M-1,N-1} \end{bmatrix}$$

Representing Digital Images

- The representation of an M×N numerical array Format

$$f(x,y)=\begin{bmatrix} f(1,1) & f(1,2) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \dots & \dots & \dots & \dots \\ f(M,1) & f(M,2) & \dots & f(M,N) \end{bmatrix}$$

Representing Digital Images

- Discrete intensity interval [0, L-1], L=2^k = 2^2 = 4 = L − 1

- The number b of bits required to store a M × N digitized image

Ex: k=2 , M×N = 32×32

$b = M \times N \times k$ = 32 × 32 × 2

Broad Matter = $\boxed{2048}$

300

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

NUMBER OF STORAGE BITS

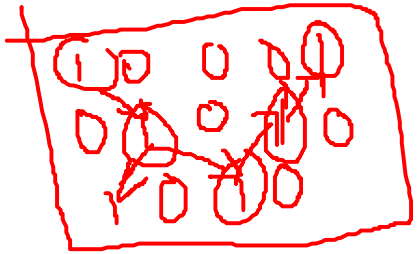
TABLE 2.1

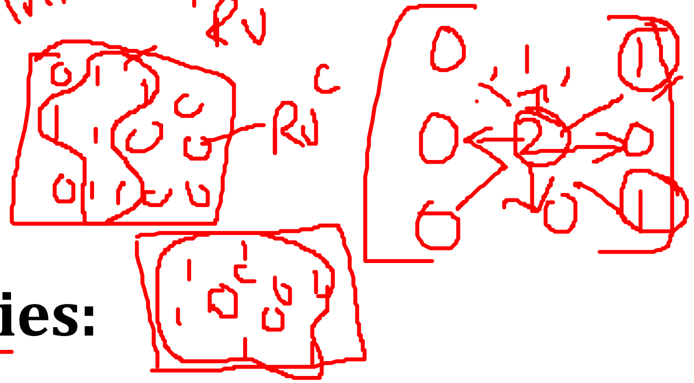
Number of storage bits for various values of N and k.

N/k	1 (L = 2)	2 (L = 4)	3 (L = 8)	4 (L = 16)	5 (L = 32)	6 (L = 64)	7 (L = 128)	8 (L = 256)
32 ×32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64 ×64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

Basic Relationships Between Pixels

- **Neighborhood:**
- **Adjacency:** "V" – intensity value $V = \{1, 2\}$
- **Connectivity:**
- **Paths:**
- **Regions and boundaries:**





(1, 2) (1, 3) (2, 2) (3, 3)
(1, 3) (2, 2) (3, 3)
8 adj

The *boundary* of the region R is the set of pixels in the region that have one or more neighbors that are not in R.

Foregound(R_u)and background $(R_u)^c$ **:**

An image contains K disjoint regions, R_k, k = 1, 2, …, K. Let R_u denote the union of all the K regions, and let $(R_u)^c$ denote its complement.

GRAY SCALE

Lmin 3 4 5 Lmax

$f(x,y)$ T $G(x,y)$

- The interval [Lmin , Lmax] is called the gray scale.
- Common practice is to shift this interval numerically to the interval [0, L-1],
- where L = 0 is considered black and L = L-1 is considered white on the gray scale. All intermediate values are shades of gray varying from black to white
- G (x,y) = T (f(x,y)) depends only on the value of f at (x,y) f(x,y) is the input image, G (x,y) is the output image T is called a gray-level or intensity transformation operator which can apply to single image.

Some Basic Intensity (Gray-level) Transformation Functions

- Grey-level transformation functions (also called, intensity functions), are considered the simplest of all image enhancement techniques.
- The value of pixels, before an after processing, will be denoted by *r* and *s*, respectively. These values are related by the expression of the form:

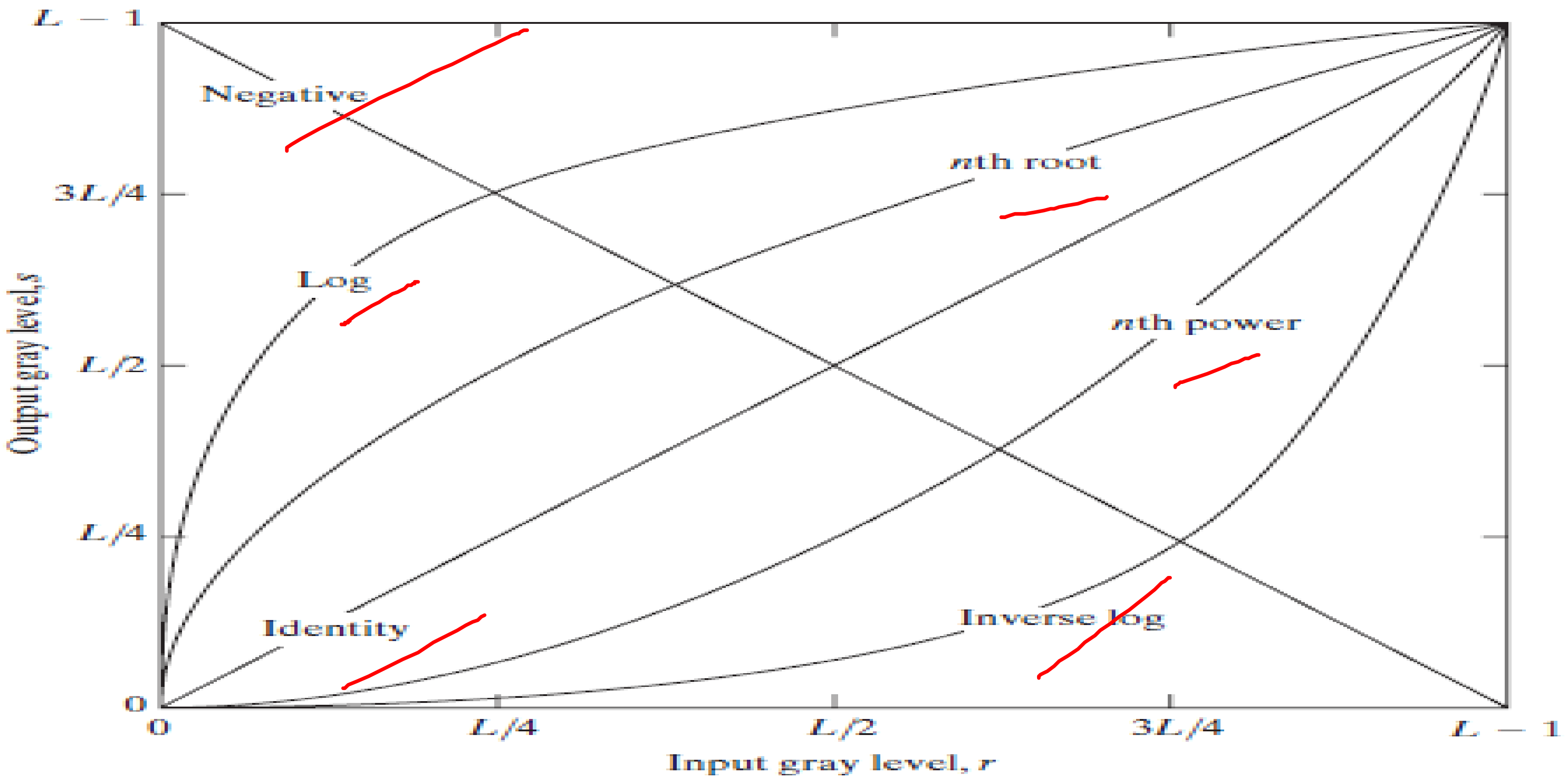
$$\boxed{s = T(r)}$$

where T is a transformation that maps a pixel value r into a pixel value s.

Some Basic Intensity (Gray-level) Transformation Functions

Consider the following figure, which shows three basic types of functions used frequently for image enhancement:

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



Some Basic Intensity (Gray-level) Transformation Functions

- The three basic types of functions used frequently for image enhancement:
  - Linear Functions:
    - Negative Transformation
    - Identity Transformation
  - Logarithmic Functions:
    - Log Transformation
    - Inverse-log Transformation
  - Power-Law Functions:
    - n^{th} power transformation
    - n^{th} root transformation

Linear Functions

$$\begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{matrix}$$

- **Identity Function**
  - Output intensities are identical to input intensities
  - This function doesn’t have an effect on an image, it was included in the graph only for completeness
  - Its expression:

I/P $s = r$ O/P

Linear Functions

- **Image Negatives (Negative Transformation)**
  - The negative of an image with gray level in the range [0, L-1], where L = Largest value in an image, is obtained by using the negative transformation's expression:

$$\boxed{s = L - 1 - r}$$

Which reverses the intensity levels of an input image, in this manner produces the equivalent of a photographic negative.

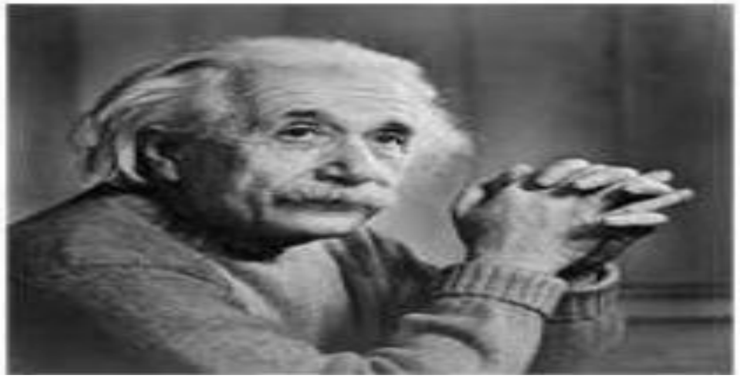
  - The negative transformation is suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area are dominant in size

NEGATIVE TRANSFORMATION EXAMPLE

Graph representation

$s = L - 1 - r$

Input image



Output image

Logarithmic Transformations

- **Log Transformation**

  The general form of the log transformation:

  $$s = c \log(1+r)$$

  Where c is a constant, and $r \geq 0$

  - Log curve maps a narrow range of low gray-level values in the input image into a wider range of the output levels.
  - Used to expand the values of dark pixels in an image while compressing the higher-level values.
  - It compresses the dynamic range of images with large variations in pixel values.

Logarithmic Transformations

a b

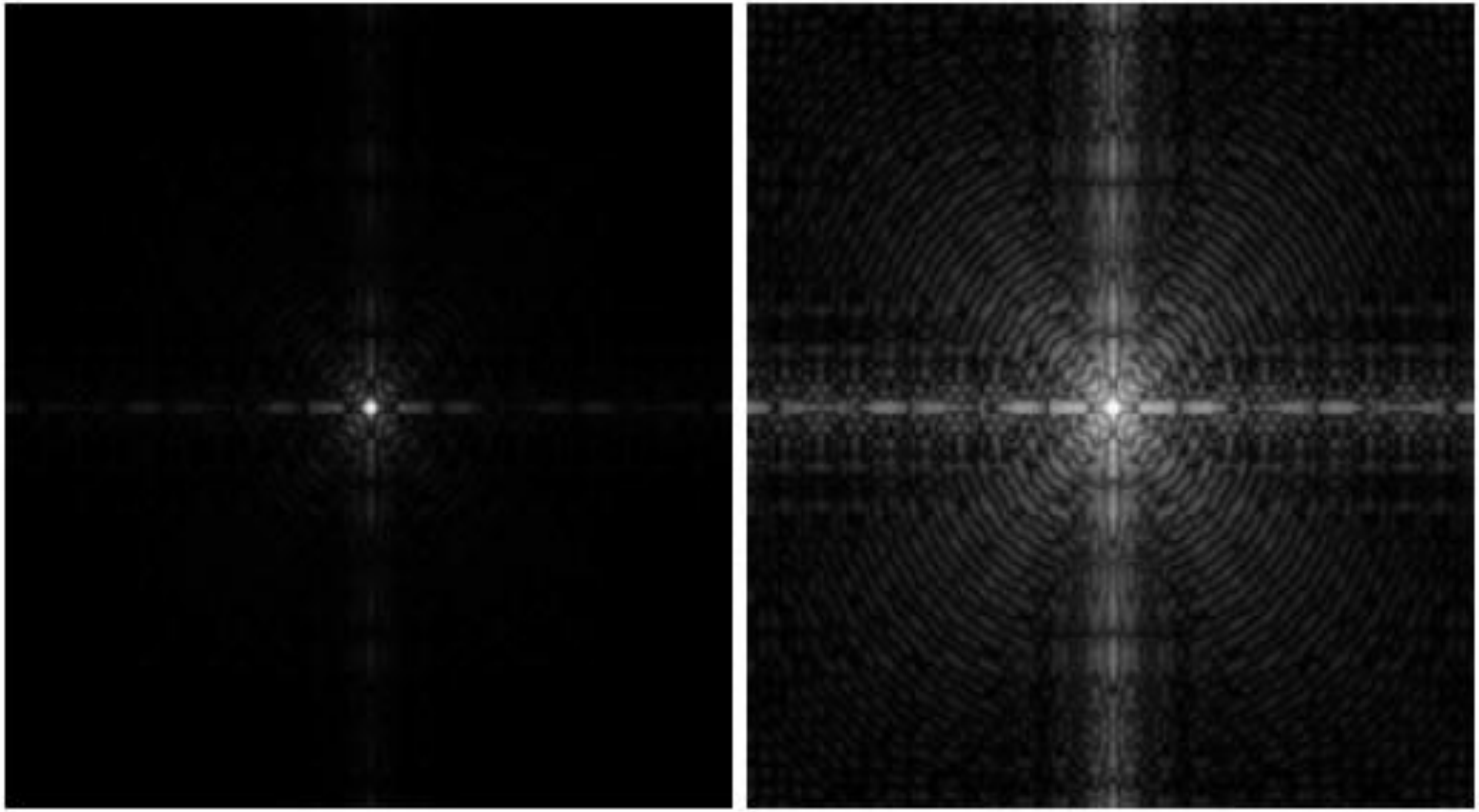
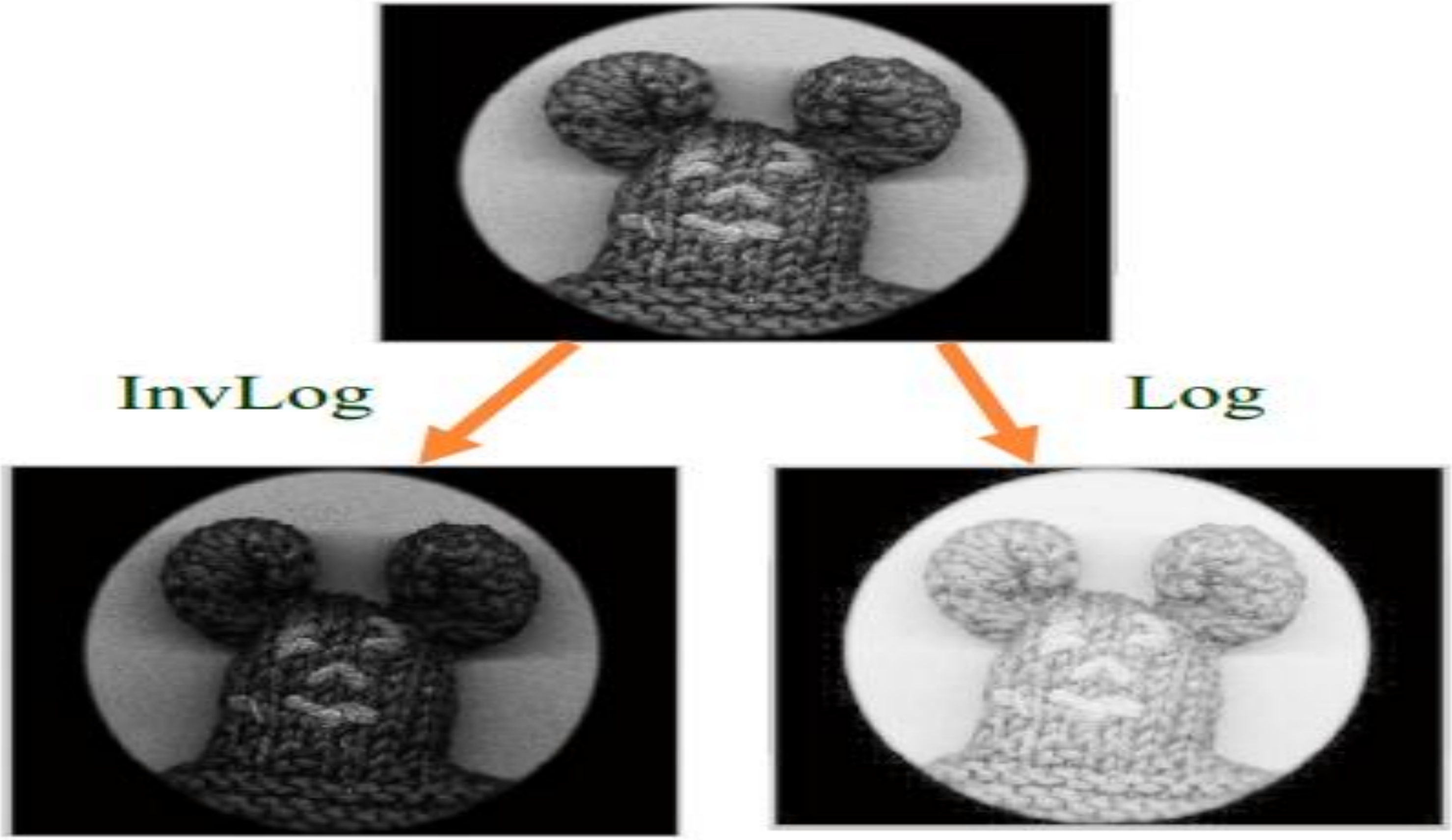


FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.

Logarithmic Transformations

- **Inverse Logarithm Transformation**
  - Do opposite to the log transformations
  - Used to expand the values of high pixels in an image while compressing the darker-level values.

Log transformation example



Power-Law Transformations

- Power-law transformations have the basic form of:

$$s = c.r^{\gamma}$$

Where c and γ are positive constants

O/P - Pixel value

I/P - " "

Vedanis

Power-Law Transformations

- Different transformation curves are obtained by varying γ (gamma)

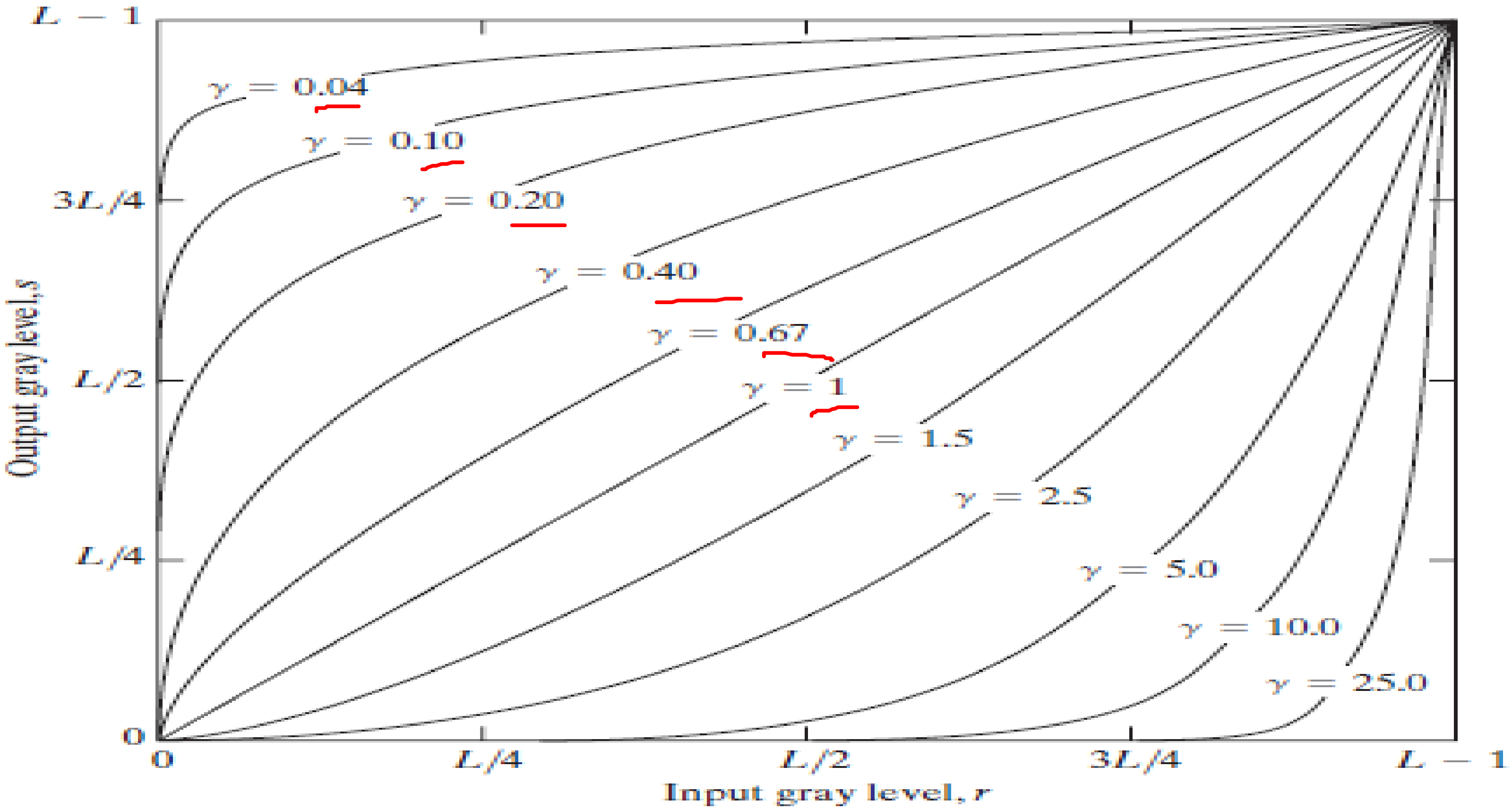


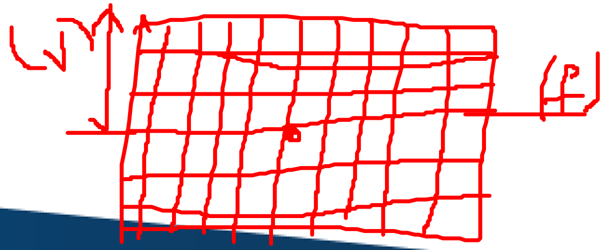
FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Power-Law Transformations

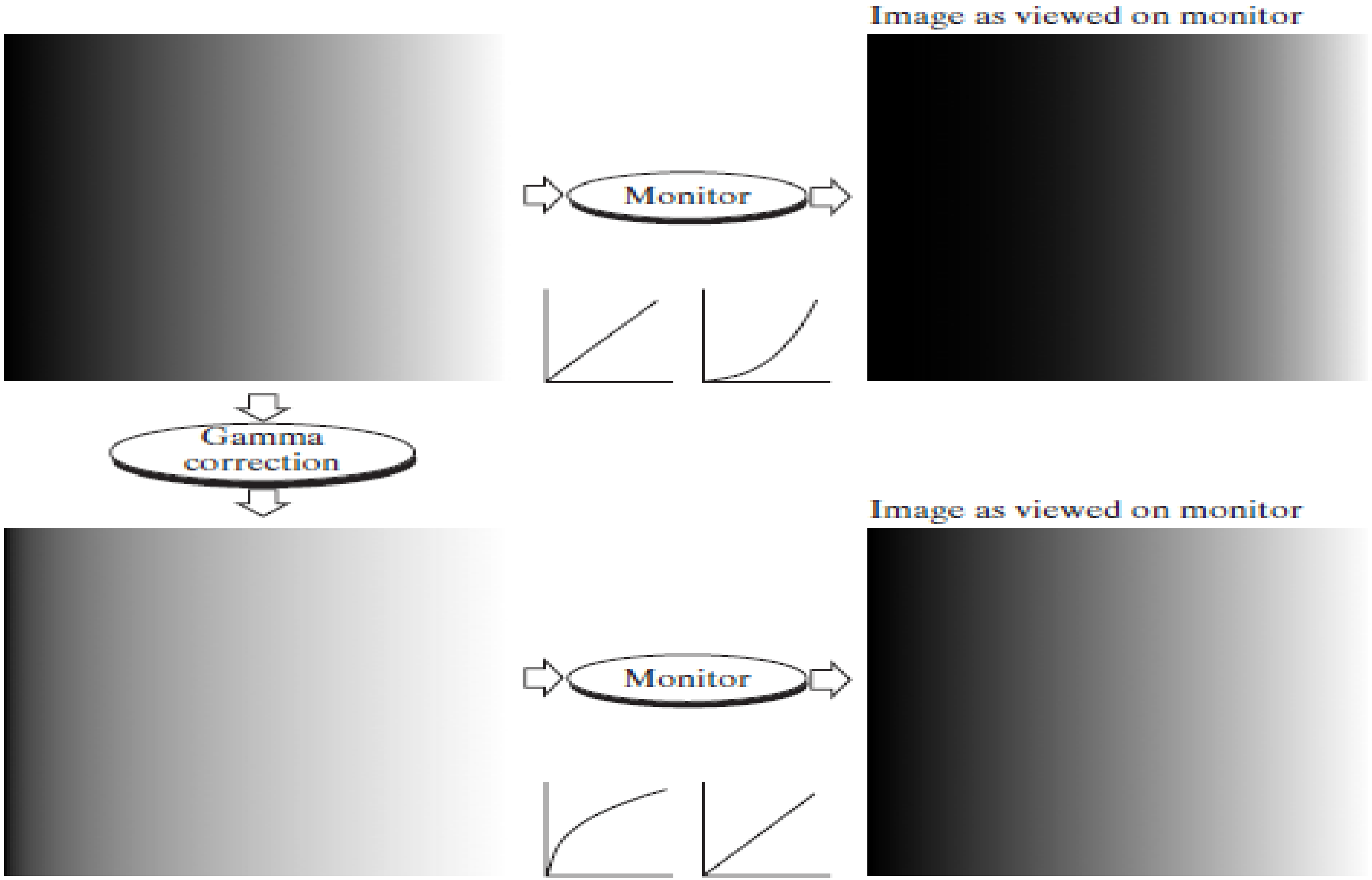
- Variety of devices used for image capture, printing and display respond according to a power law. The process used to correct this power-law response phenomena is called ***gamma correction***.

  For example, cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with exponents varying from approximately 1.8 to 2.5.With reference to the curve for g=2.5 in Fig. 3.6, we see that such display systems would tend to produce images that are darker than intended. This effect is illustrated in Fig. 3.7. Figure 3.7(a) shows a simple gray-scale linear wedge input into a CRT monitor. As expected, the output of the monitor appears darker than the input, as shown in Fig. 3.7(b). Gamma correction

  in this case is straightforward. All we need to do is preprocess the input image before inputting it into the monitor by performing the transformation. The result is shown in Fig. 3.7(c).When input into the same monitor, this gamma-corrected input produces an output that is close in appearance to the original image, as shown in Fig. 3.7(d).



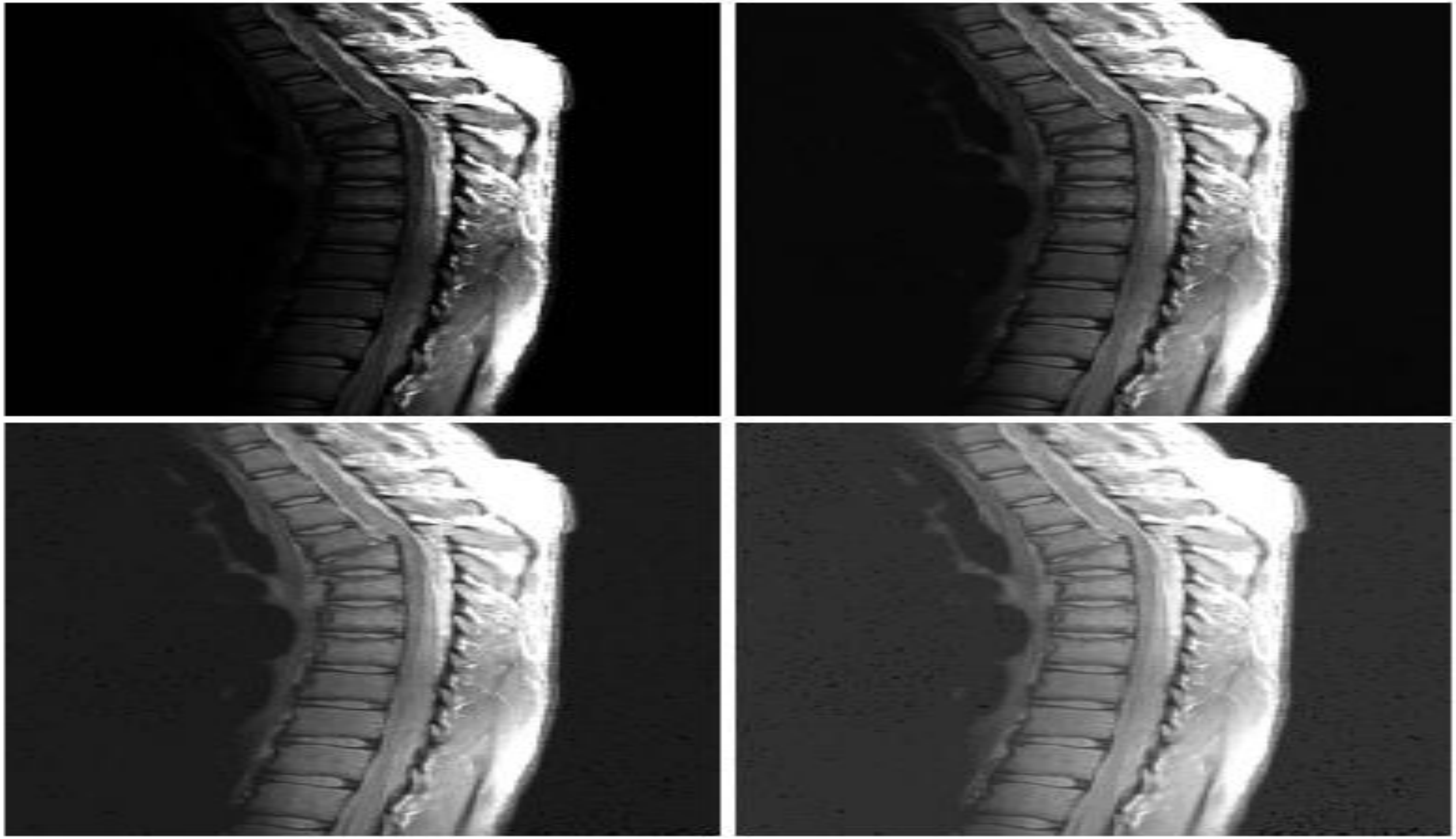
Power-Law Transformation



a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

Power-Law Transformation



a b
c d

FIGURE 3.8 (a) Magnetic resonance (MR) image of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

illustration of Power-Law Transformation

a b
c d

FIGURE 3.9
(a) Aerial image.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0, respectively. (Original image for this example courtesy of NASA.)

Power law transformation example

Gamma=10

Gamma=8

Gamma=6

Piecewise-Linear Transformation Functions

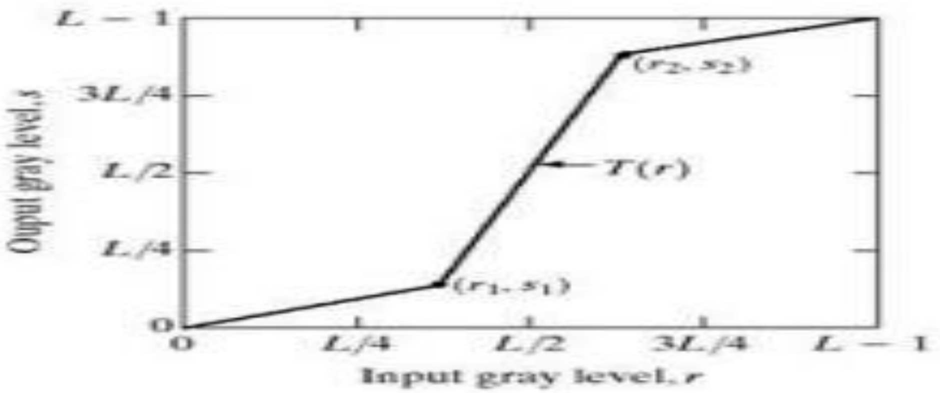
- **Principle Advantage:** Some important transformations can be formulated only as a piecewise function.
- **Principle Disadvantage:** Their specification requires more user input that previous transformations
- **Types of Piecewise transformations are:**
  - Contrast Stretching
  - Gray-level Slicing
  - Bit-plane slicing

Contrast Stretching

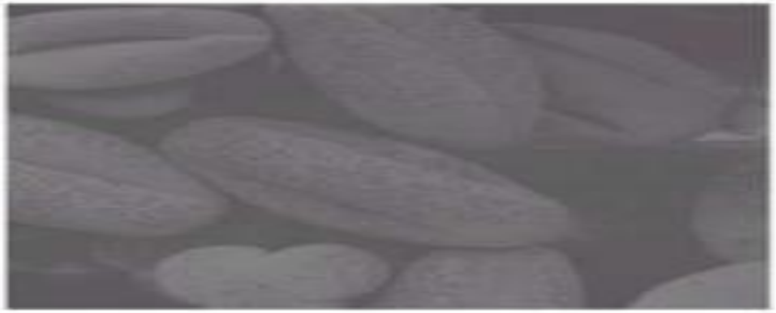
- One of the simplest piecewise linear functions is a contrast-stretching transformation, which is used to enhance the low contrast images.
- Low contrast images may result from:
  - Poor illumination
  - Wrong setting of lens aperture during image acquisition.

CONTRAST STRETCHING EXAMPLE

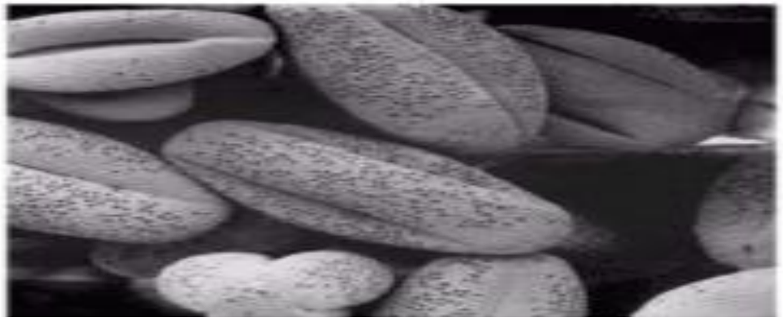
Transformation function



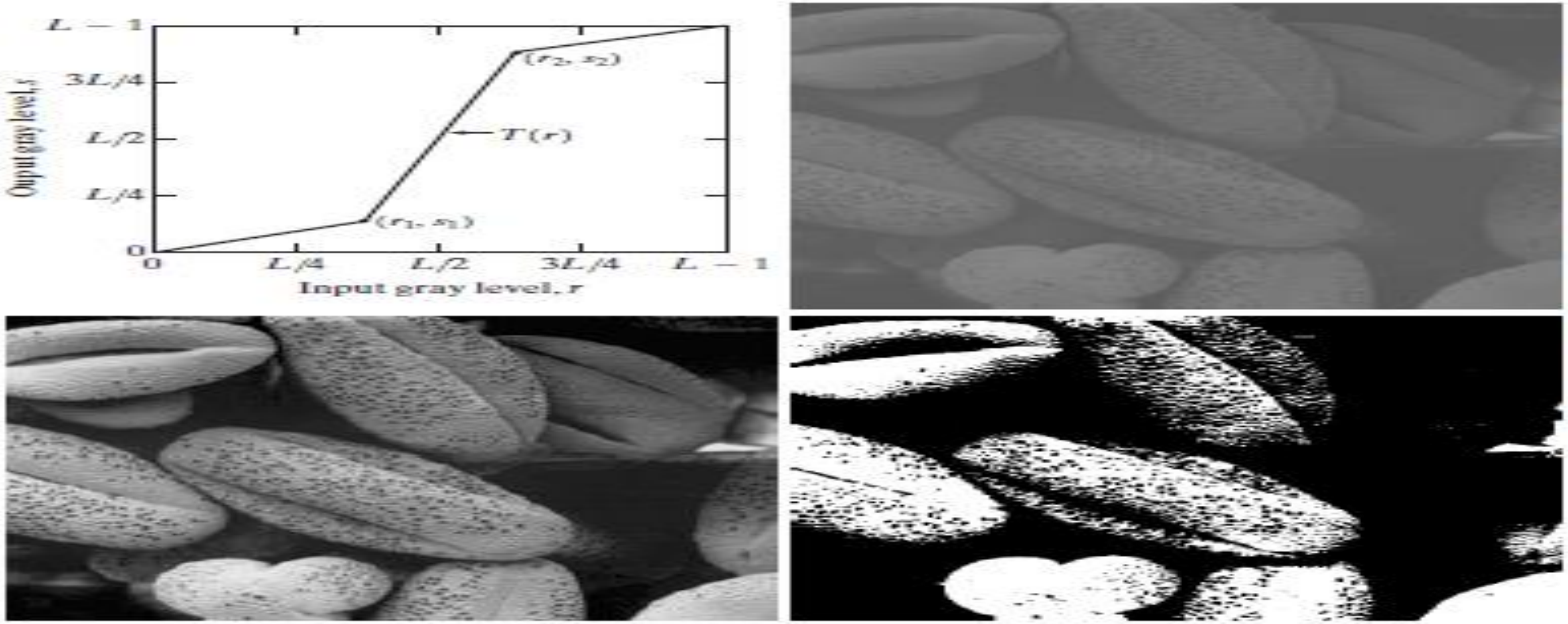
Low contrast image



Contrast stretching image



Contrast Stretching



a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Contrast Stretching

- Figure 3.10(a) shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function.
- If r1 = s1 and r2 = s2, the transformation is a linear function that produces no changes in gray levels.
- If r1 = r2, s1 = 0 and s2 = L-1, the transformation becomes a *thresholding function* that creates a binary image. As shown previously in slide 7.
- Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.
- In general, r1 ≤ r2 and s1 ≤ s2 is assumed, so the function is always increasing.

Contrast Stretching

- Figure 3.10(b) shows an 8-bit image with low contrast.

- Fig. 3.10(c) shows the result of contrast stretching, obtained by setting (r1, s1) = (r_{min}, 0) and (r2, s2) = (r_{max},L-1) where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range [0, L-1].

- Finally, Fig. 3.10(d) shows the result of using the *thresholding function* defined previously, with r1=r2=m, the mean gray level in the image.

Gray-level Slicing

- This technique is used to highlight a specific range of gray levels in a given image. It can be implemented in several ways, but the two basic themes are:
  - One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels. This transformation, shown in Fig 3.11 (a), produces a binary image.
  - The second approach, based on the transformation shown in Fig 3.11 (b), brightens the desired range of gray levels but preserves gray levels unchanged.
  - Fig 3.11 (c) shows a gray scale image, and fig 3.11 (d) shows the result of using the transformation in Fig 3.11 (a).

Gray-level Slicing

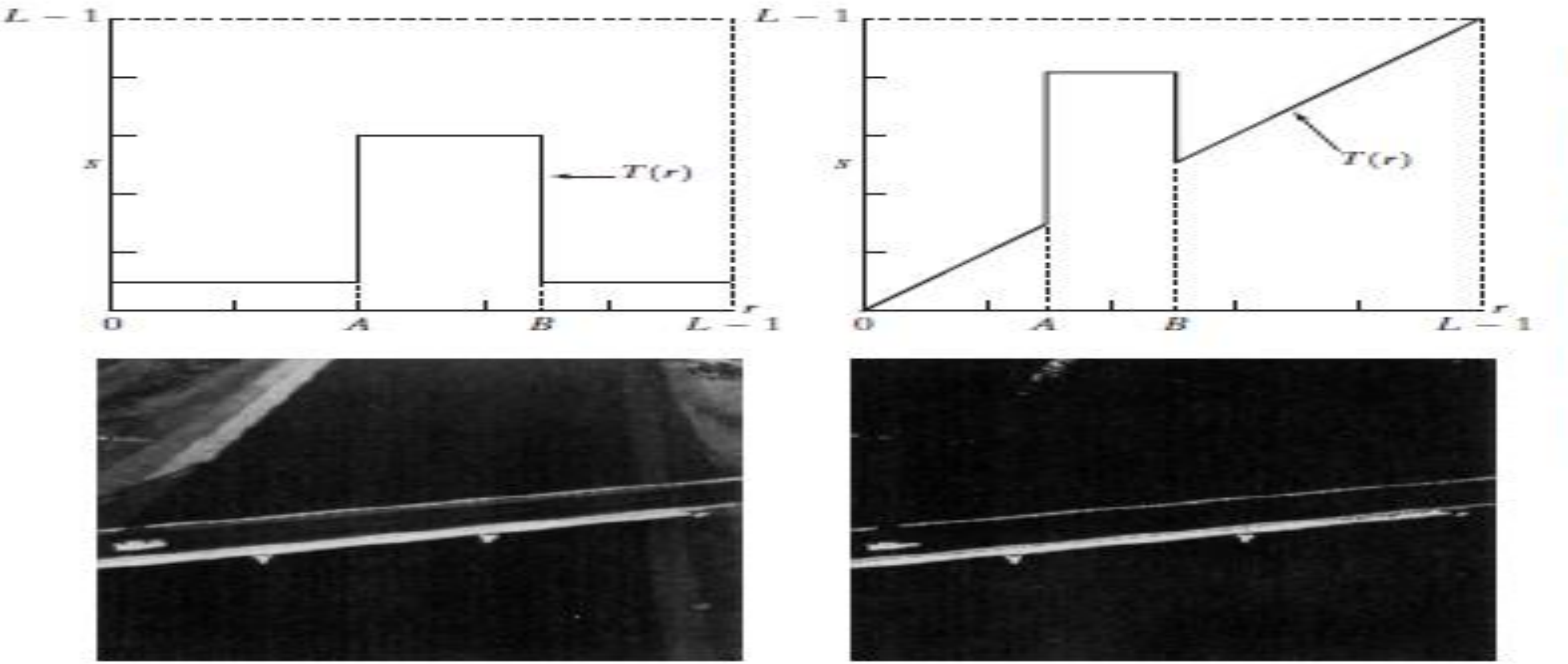


FIGURE 3.11 (a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level. (b) This transformation highlights range $[A, B]$ but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

Input image

Output image

Bit-plane Slicing

- Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit.
- This method is useful and used in image compression.

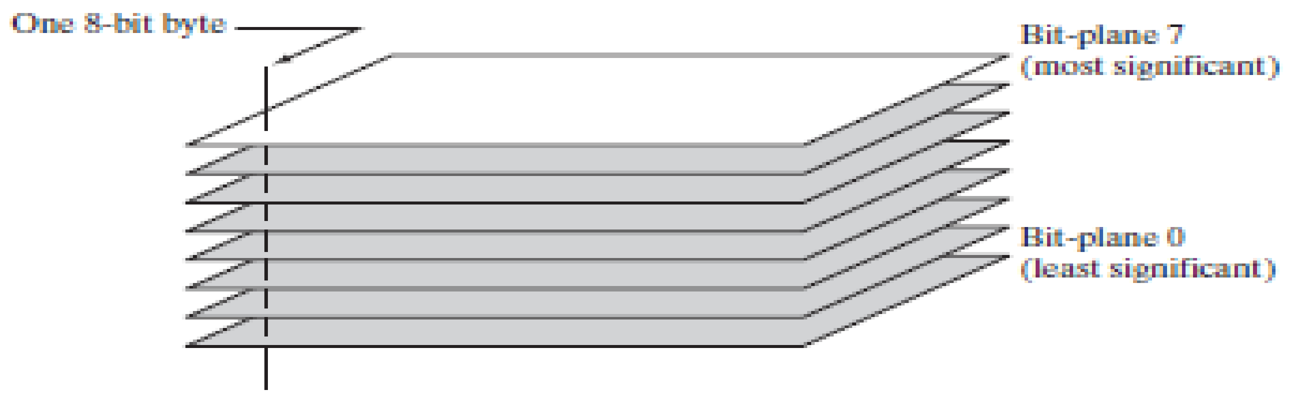


FIGURE 3.12 Bit-plane representation of an 8-bit image.

- Most significant bits contain the majority of visually significant data.

BIT PLANE SLICING EXAMPLE

Original image

Bit plane 7

Bit plane 6

Bit plane 4

Bit plane 1

1. Which of the following expression is used to denote spatial domain process?

a) $g(x,y)=T[f(x,y)]$

b) $f(x+y)=T[g(x+y)]$

c) $g(xy)=T[f(xy)]$

d) $g(x-y)=T[f(x-y)]$

1. Which of the following expression is used to denote spatial domain process?

a) g(x,y)=T[f(x,y)]

b) f(x+y)=T[g(x+y)]

c) g(xy)=T[f(xy)]

d) g(x-y)=T[f(x-y)]

2.Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range[0,L-1] ?

a) s=L+1-r

b) s=L+1+r

c) s=L-1-r

d) s=L-1+r

2.Which expression is obtained by performing the negative transformation on the negative of an image with gray levels in the range[0,L-1] ?

a) s=L+1-r

b) s=L+1+r

c) s=L-1-r

d) s=L-1+r

3.What is the general form of representation of log transformation?

a) $s = c\log_{10}(1/r)$

b) $s = c\log_{10}(1+r)$

c) $s = c\log_{10}(1*r)$

d) $s = c\log_{10}(1-r)$

3.What is the general form of representation of log transformation?

a) $s=c\log_{10}(1/r)$

b) $s=c\log_{10}(1+r)$

c) $s=c\log_{10}(1*r)$

d) $s=c\log_{10}(1-r)$

4.What is the general form of representation of power transformation?

a) $s=cr^{\gamma}$

b) $c=sr^{\gamma}$

c) $s=rc$

d) $s=rc^{\gamma}$

4.What is the general form of representation of power transformation?

a) $s=cr^{\gamma}$

b) $c=sr^{\gamma}$

c) $s=rc$

d) $s=rc^{\gamma}$

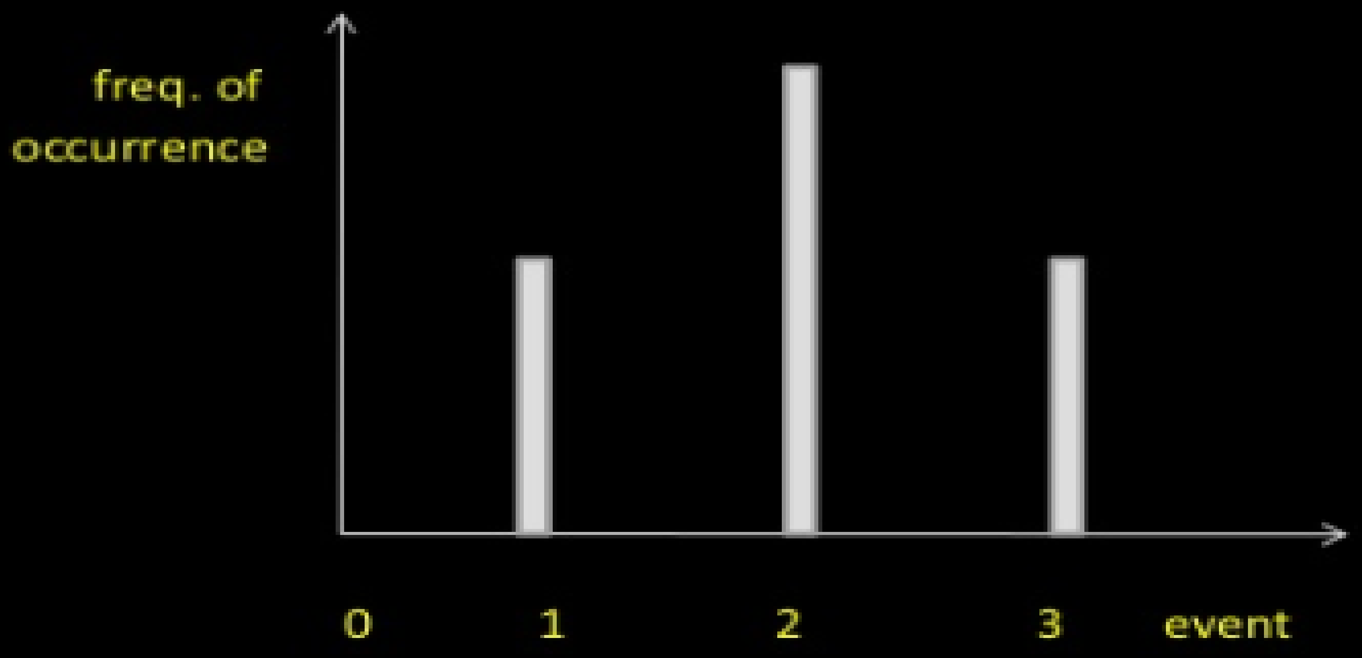
MODULE-2

2.2-Histogram Processing ,Histogram equalization

Histogram Processing

Histogram:

It is a plot of frequency of occurrence of an event.



Histogram Processing

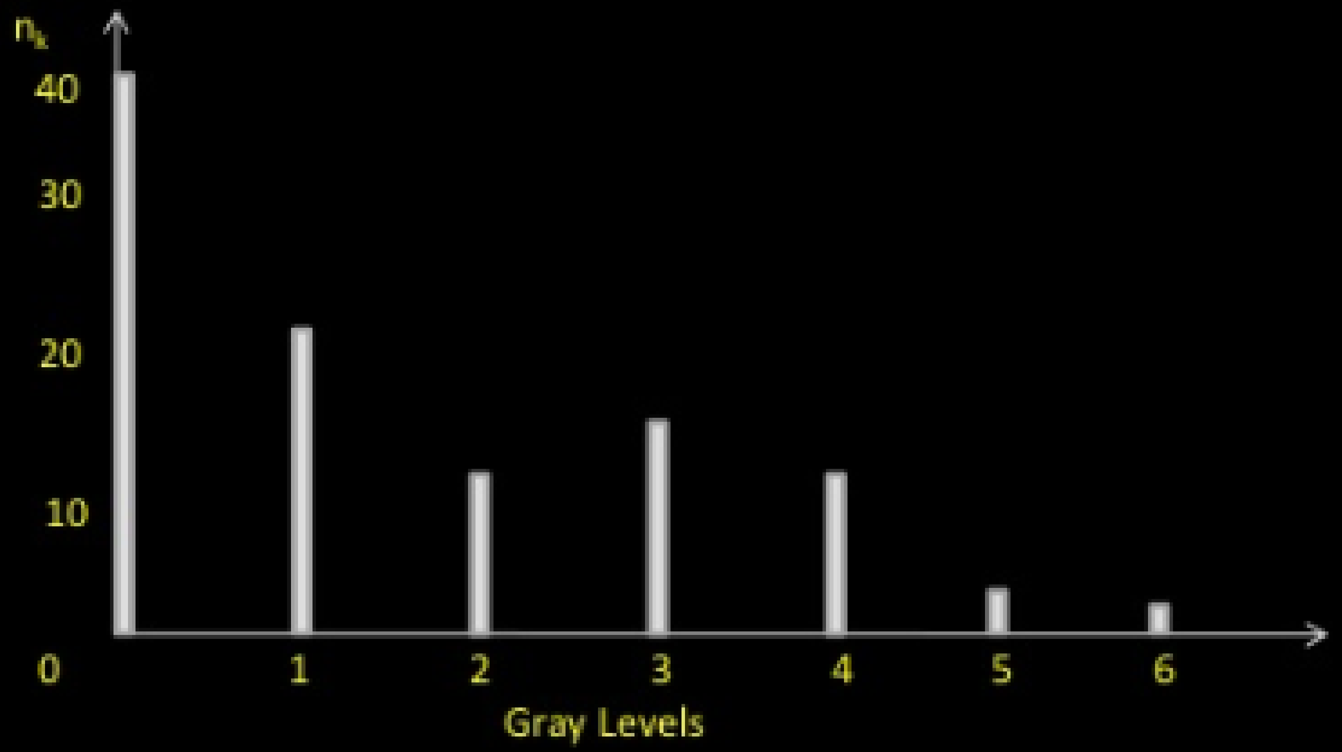
- Histogram of images provide a global description of their appearance.
- Enormous information is obtained.
- It is a spatial domain technique.
- Histogram of an image represents relative frequency of occurrence of various gray levels.
- Histogram can be plotted in two ways:

Histogram Processing

- <u>First Method</u>:

- X-axis has gray levels & Y-axis has No. of pixels in each gray levels.

Gray Level	No. of Pixels (n_k)
0	40
1	20
2	10
3	15
4	10
5	3
6	2



Consider a 5x5 image with integer intensities in the range between zero and seven:

Image

0	7	3	2	3
0	0	0	6	7
7	7	2	2	0
1	1	0	4	1
0	0	7	4	1

Image matrix

Gray scale

Consider a 5x5 image with integer intensities in the range between one and Seven

Image

0	7	3	2	3
0	0	0	6	7
7	7	2	2	0
1	1	0	4	1
0	0	7	4	1

Image matrix

Grey scale

Number of pixel with intensity value 0 [h(r0)] = 8

Image

0	7	3	2	3
0	0	0	6	7
7	7	2	2	0
1	**1**	0	4	**1**
0	0	7	4	**1**

Image matrix

Grey scale

Number of pixel with intensity value 0 [h(r0)] = 8

Similarly for **1** h(r1) = 4

Image

0	7	3	2	3
0	0	0	6	7
7	7	2	2	0
1	1	0	4	1
0	0	7	4	1

Image matrix

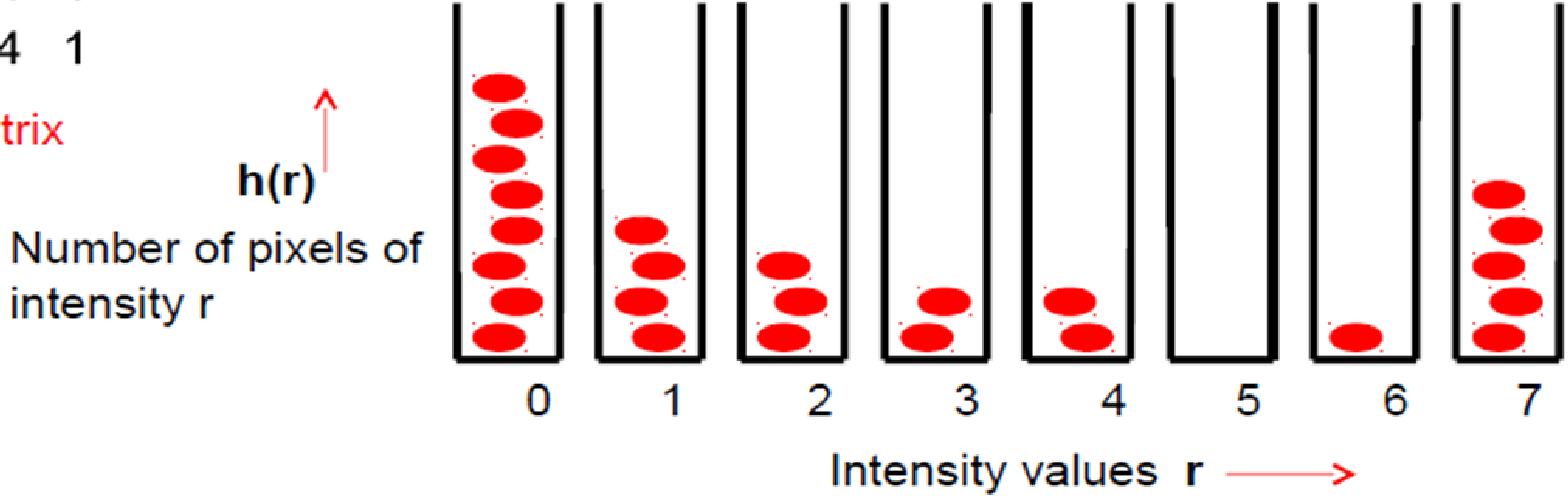
Similarly

INTENSITY r	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
NUMBER of pixels of r h(r)	h(r0)=**8**	h(r1)=**4**	h(r2)=**3**	h(r3)=**2**	h(r4)=**2**	h(r5)=**0**	h(r6)=**1**	h(r7)=**5**

0	7	3	2	3
0	0	0	6	7
7	7	2	2	0
1	1	0	4	1
0	0	7	4	1

Image matrix

r	0	1	2	3	4	5	6	7
h(r)	8	4	3	2	2	0	1	5



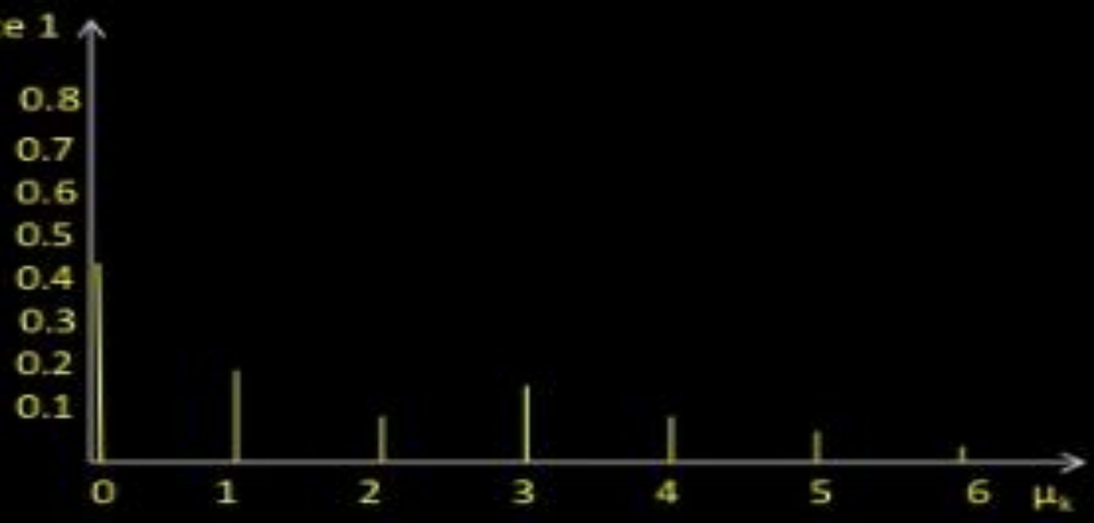
HISTOGRAM

Histogram Processing

- <u>Second Method</u>:
- X-axis has gray levels & Y-axis has probability of occurrence of gray levels.

$$P(\mu_k) = n_k / n; \quad \text{where, } \mu_k - \text{gray level}$$
$$n_k - \text{no. of pixels in } k^{th} \text{ gray level}$$
$$n - \text{total number of pixels in an image}$$

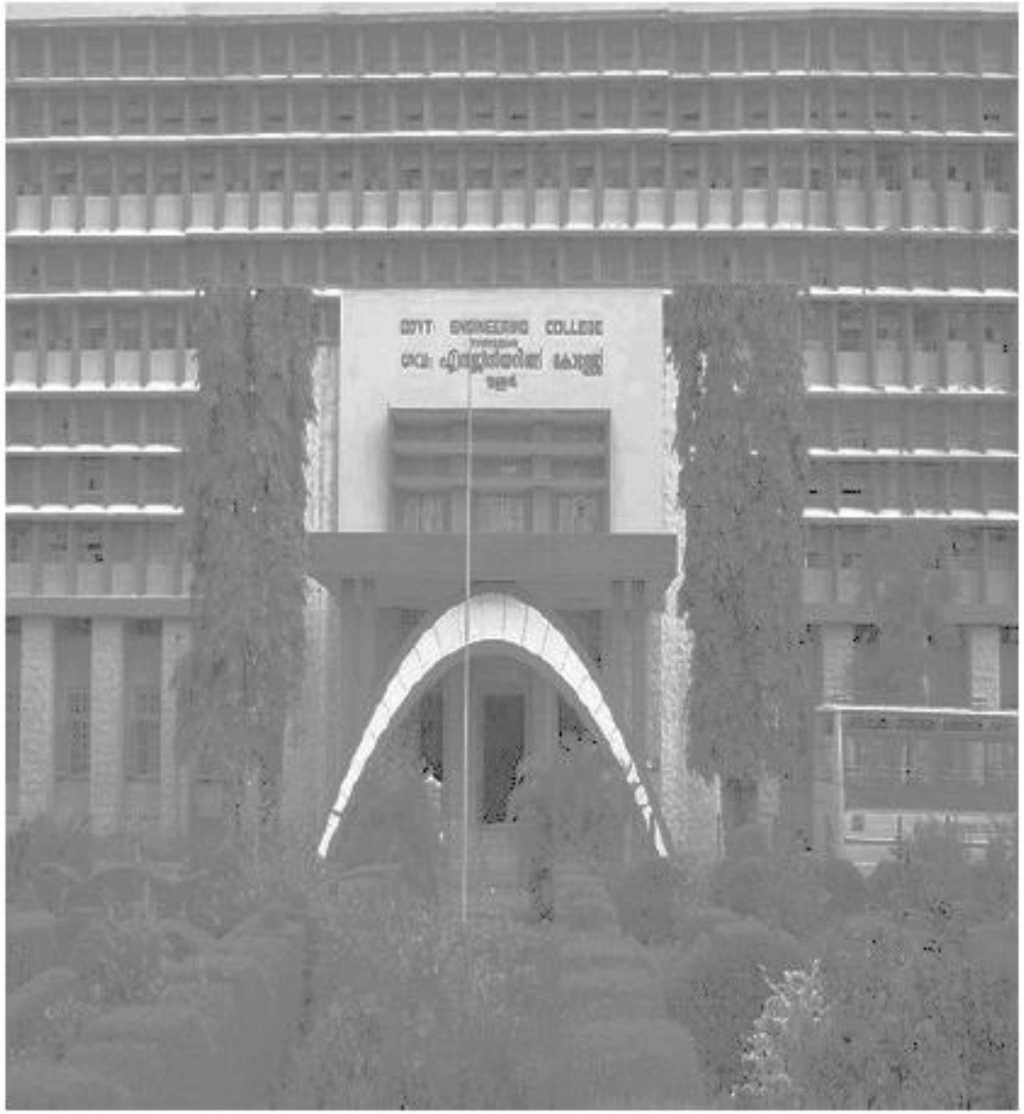
Gray Level	No. of Pixels (n_k)	$P(\mu_k)$
0	40	0.4
1	20	0.2
2	10	0.1
3	15	0.15
4	10	0.1
5	3	0.03
6	2	0.02
	n = 100	1



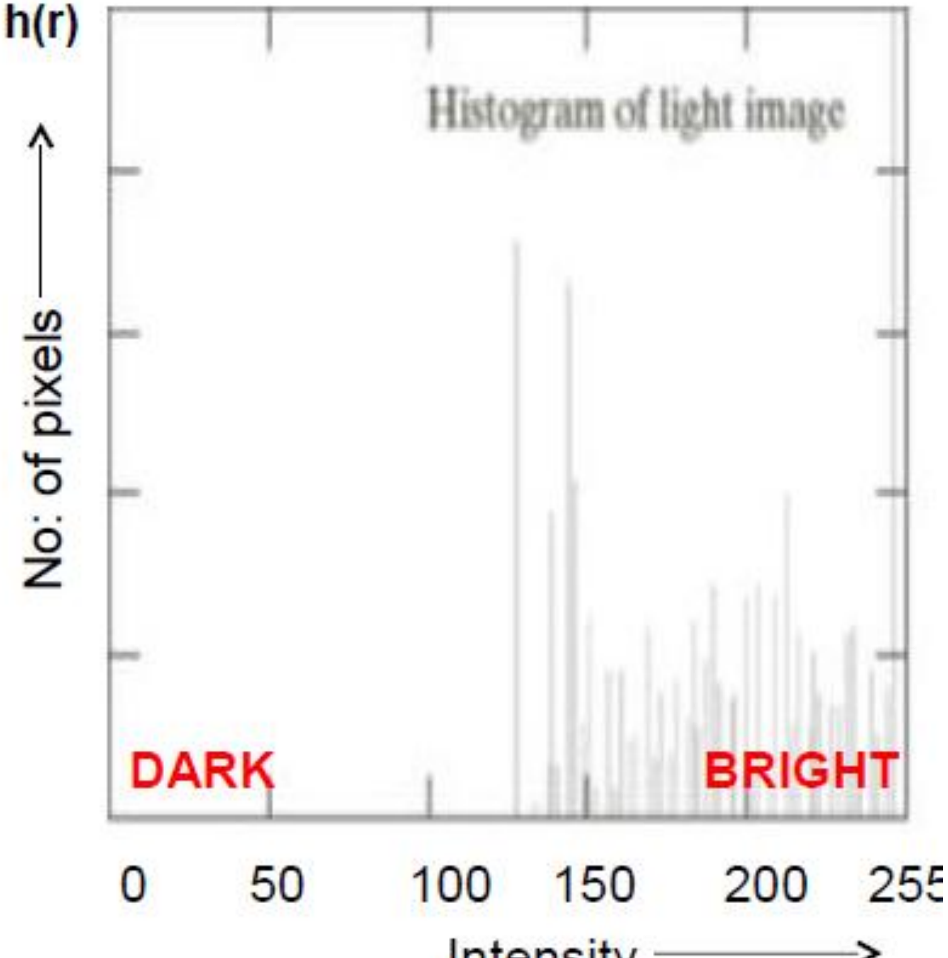
SAMPLE IMAGES AND ITS HISTOGRAM

Bright image
Intensity range 0 - 255

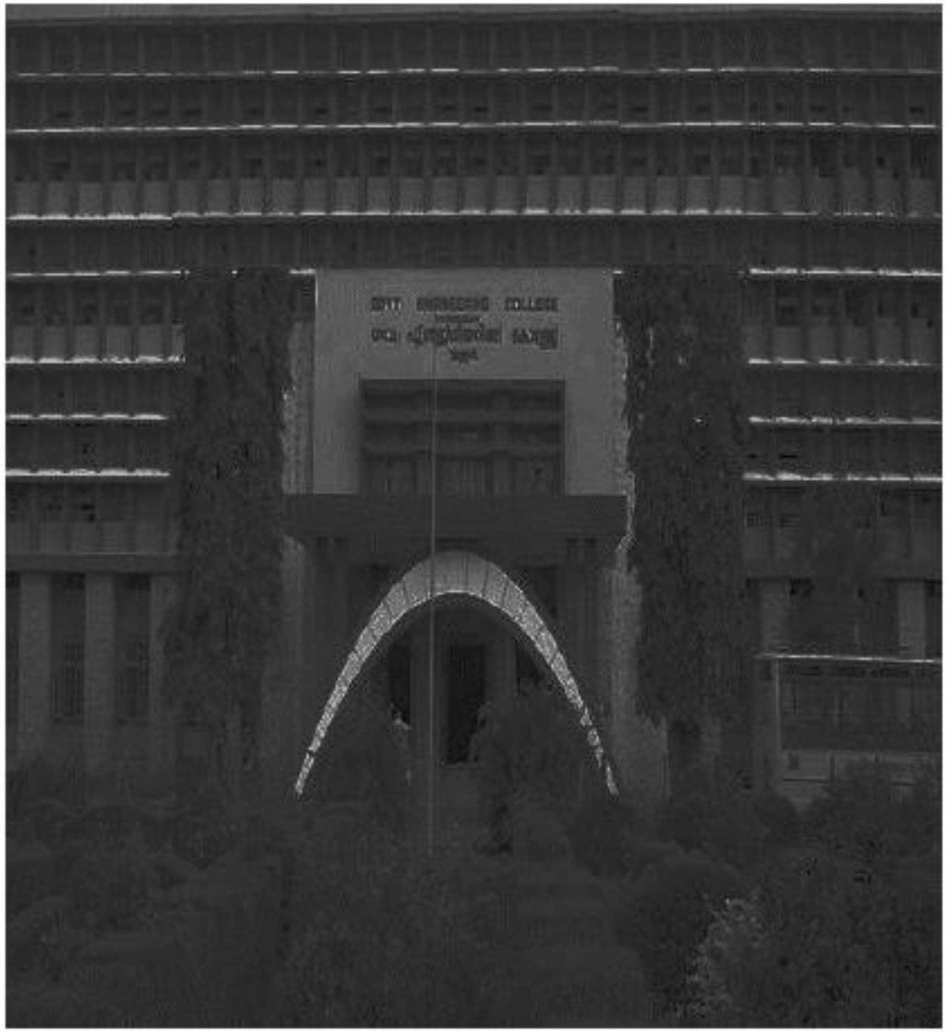
SAMPLE IMAGES AND ITS HISTOGRAM



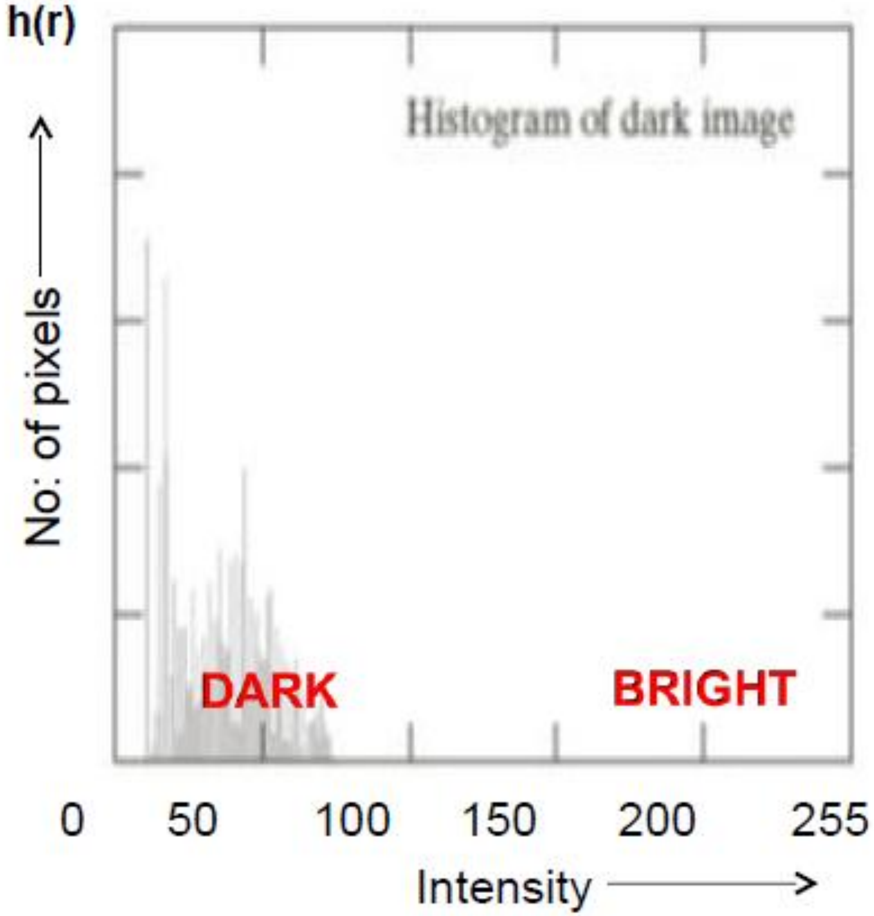
Bright image



SAMPLE IMAGES AND ITS HISTOGRAM

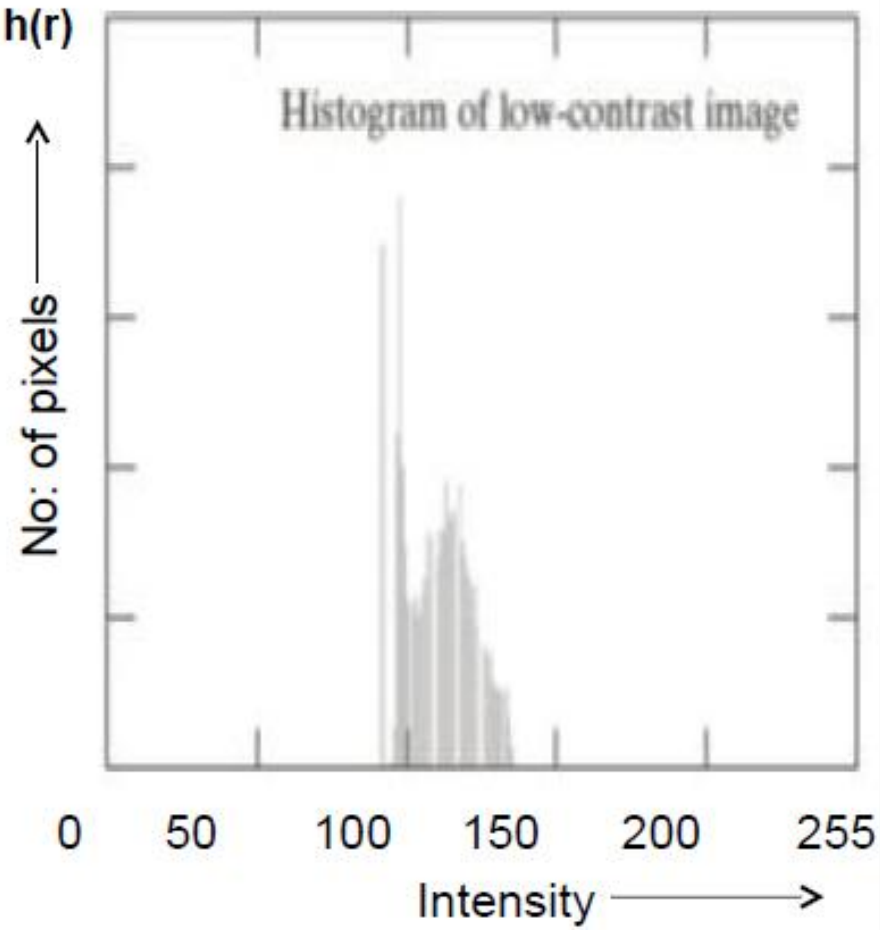


Dark image
Intensity range 0 - 255

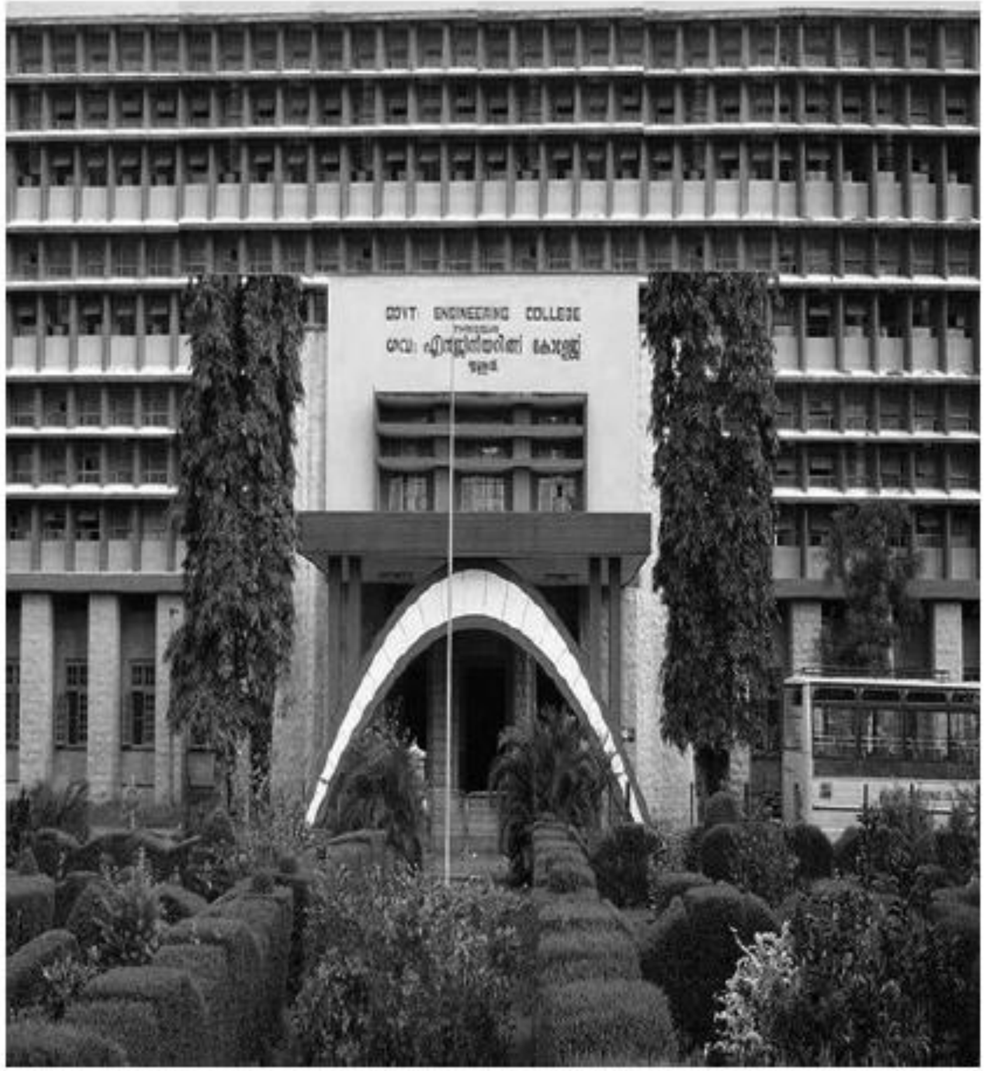


SAMPLE IMAGES AND ITS HISTOGRAM

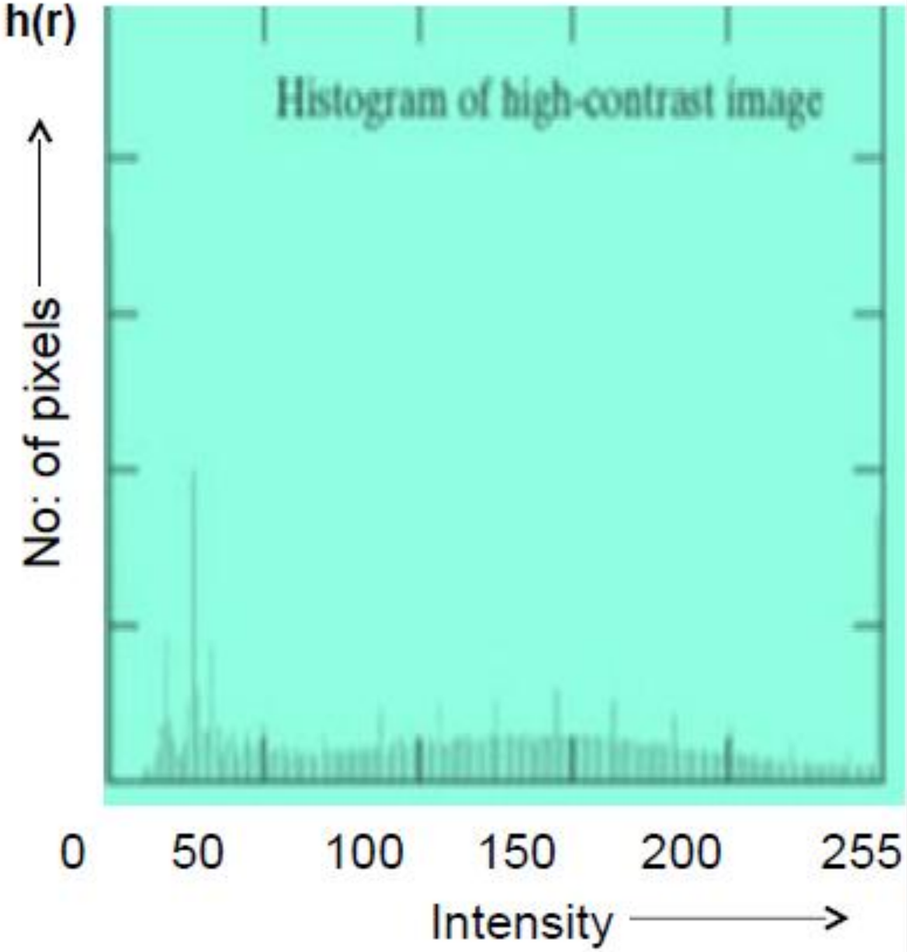
Light image
Intensity range 0 - 255



SAMPLE IMAGES AND ITS HISTOGRAM



High contrast image
Intensity range 0 - 255



Histogram Processing

Histogram $h(r_k) = n_k$

r_k is the k^{th} intensity value

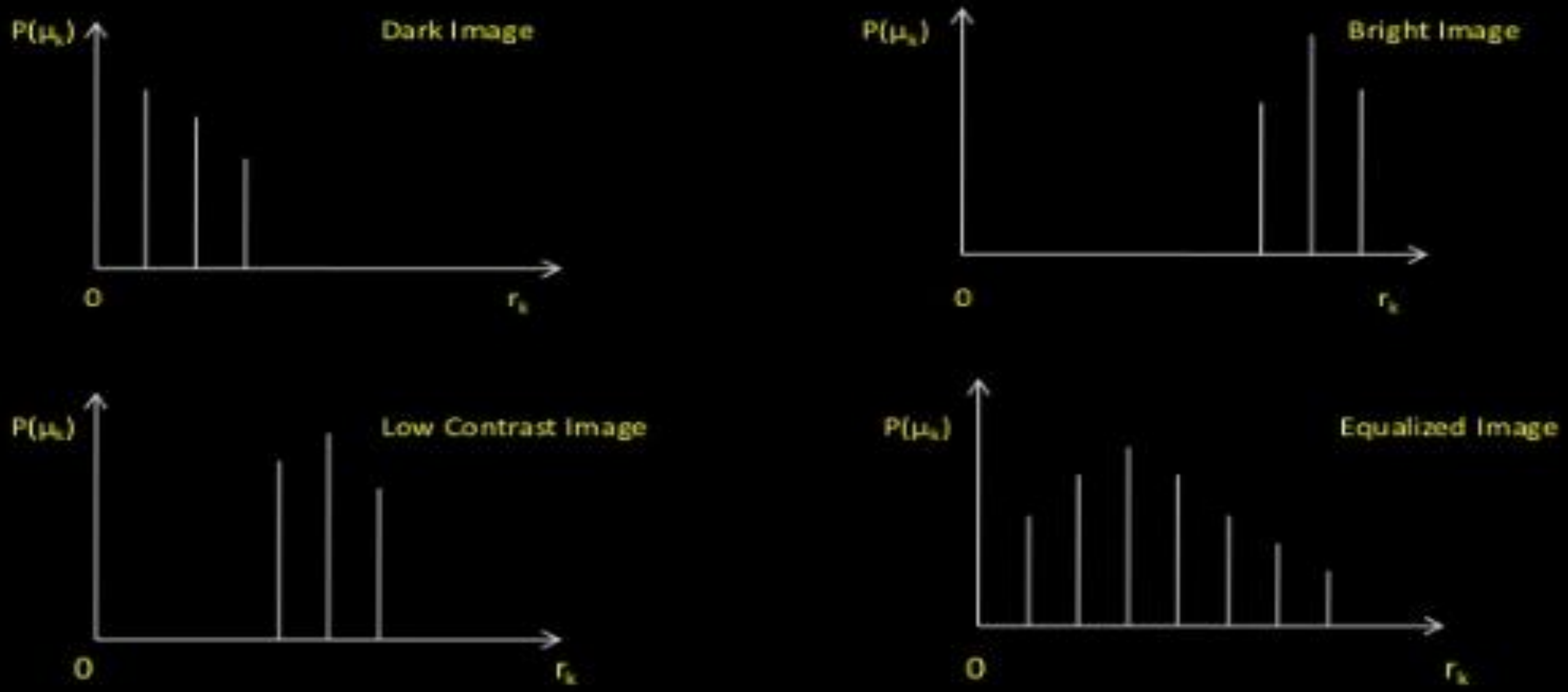
n_k is the number of pixels in the image with intensity r_k

Normalized histogram $p(r_k) = \frac{n_k}{MN}$

n_k: the number of pixels in the image of size $M \times N$ with intensity r_k

Histogram Processing

- Advantage of 2^{nd} method: Maximum value plotted will always be 1.
- White – 1, Black – 0.
- Great deal of information can be obtained just by looking at histogram.
- Types of Histograms:



Histogram Processing

- The last graph represent the best image.
- It is a high contrast image.
- Our aim would be to transform the first 3 histograms into the 4th type.
- In other words we try to increase the dynamic range of the image.

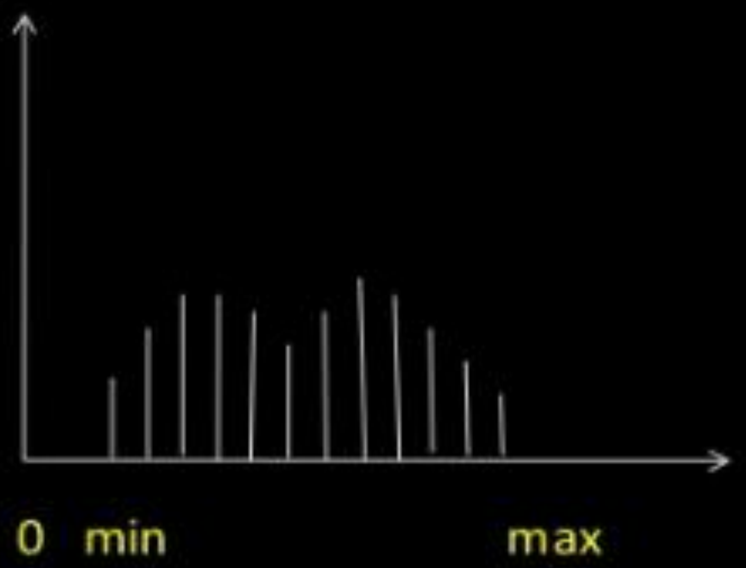
Histogram Stretching

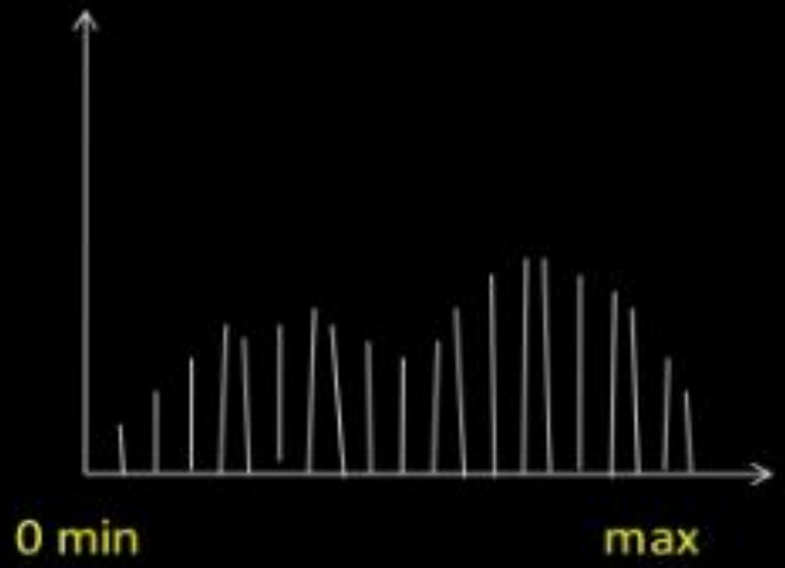
1) <u>Linear stretching:</u>

- Here, we don't alter the basic shape.
- We use basic equation of a straight line having a slope.

$$(S_{max} - S_{min})/(r_{max} - r_{min})$$

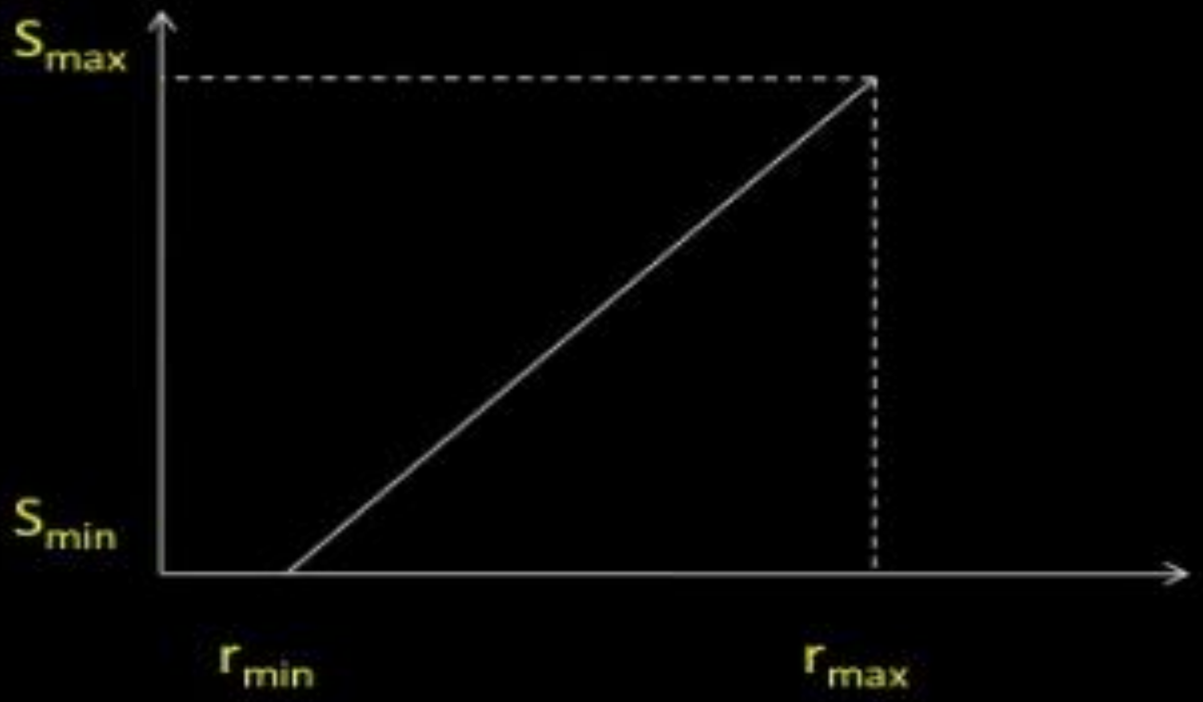
Where, Smax – max gray level of output image
Smin – min gray level of output image
rmax – max gray level of input image
rmin – min gray level of input image





Histogram Processing

$$S = T(r) = ((S_{max} - S_{min})/(r_{max} - r_{min}))(r - r_{min}) + Smin$$



Histogram Processing

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	0	0	50	60	50	20	10	0

Histogram Processing

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	0	0	50	60	50	20	10	0

Sol^n:- $r_{min} = 2$; $r_{max} = 6$; $s_{min} = 0$; $s_{max} = 7$;

slope = $((s_{max} - s_{min})/(r_{max} - r_{min})) = ((7 - 0)/(6 - 2)) = 7 / 4 = 1.75$.

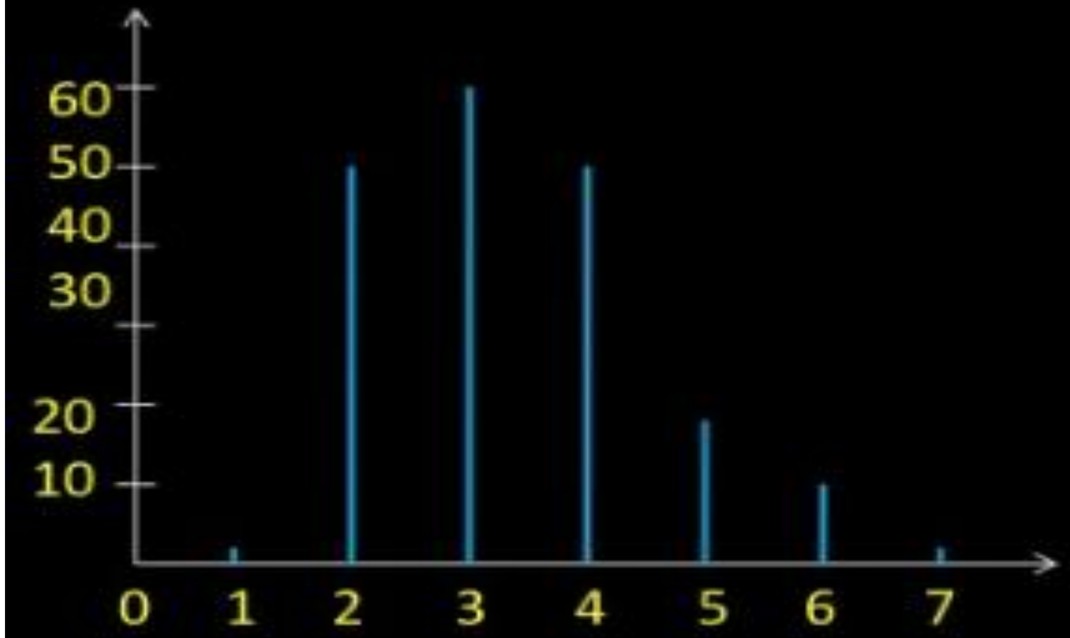
$S = (7 / 4)(r - 2) + 0;$
$S = (7 / 4)(r - 2)$

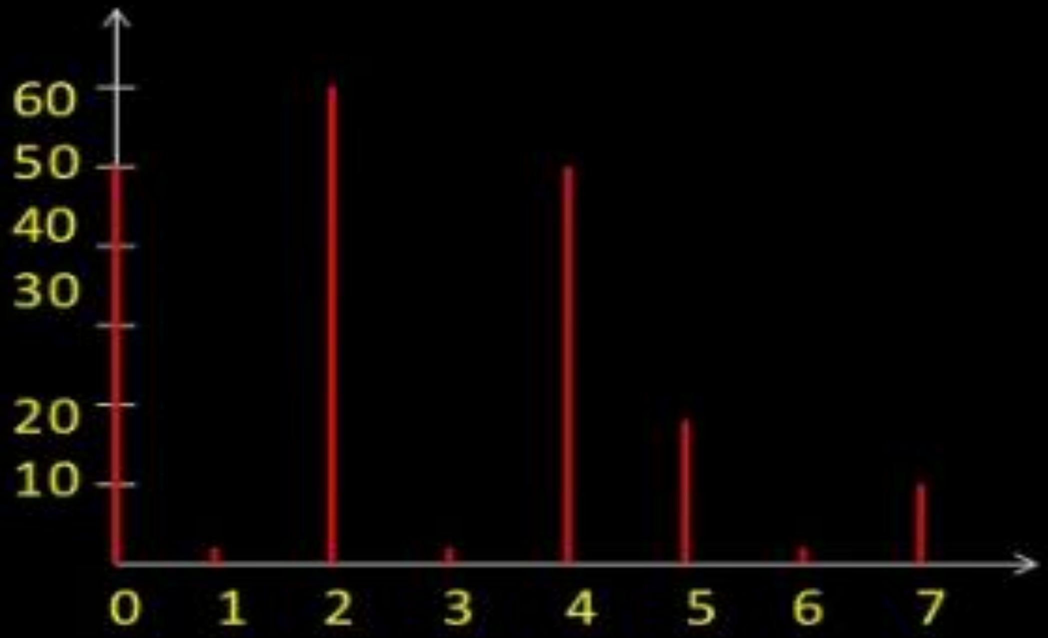
r	(7 / 4)(r - 2) = S
2	0 = 0
3	7/4 = 1.75 = 2
4	7/2 = 3.5 = 4
5	21/4 = 5.25 = 5
6	7 = 7

Histogram Processing

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	50	0	60	0	50	20	0	10





Ex. 2) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 .

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	100	90	85	70	0	0	0	0

- Rmin=0, rmax=3, smin=0, smax=7
- Slope= (7/3)=2.3
- R s
- 0 0
- 1 2
- 2 5
- 3 7

Ex. 2) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 .

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	100	0	90	0	0	85	0	70

Dynamic range increases

BUT

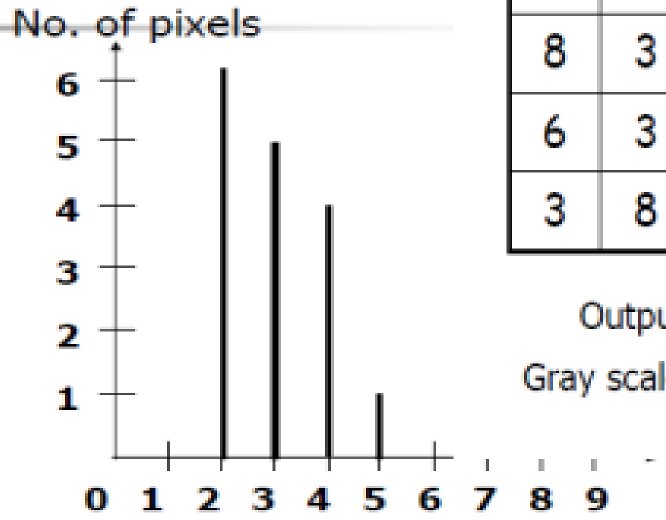
No. of pixels at each gray level remains constant.

Example

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]

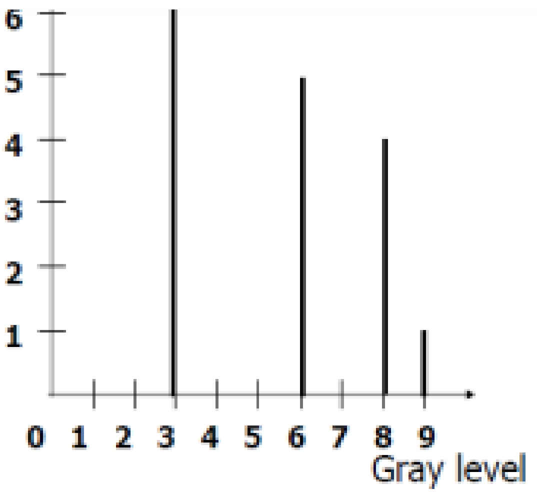


histogram

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]



Histogram equalization

Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
$s \times 9$	0	0	3.3 $\approx$3	6.1 $\approx$6	8.4 $\approx$8	9	9	9	9	9

Histogram Equalization

- What is the histogram equalization?

- The histogram equalization is an approach to enhance a given image. The approach is to design a transformation $T(.)$ such that the gray values in the output is uniformly distributed in [0, 1].

- Let us assume for the moment that the input image to be enhanced has continuous gray values, with $r = 0$ representing black and $r = 1$ representing white.

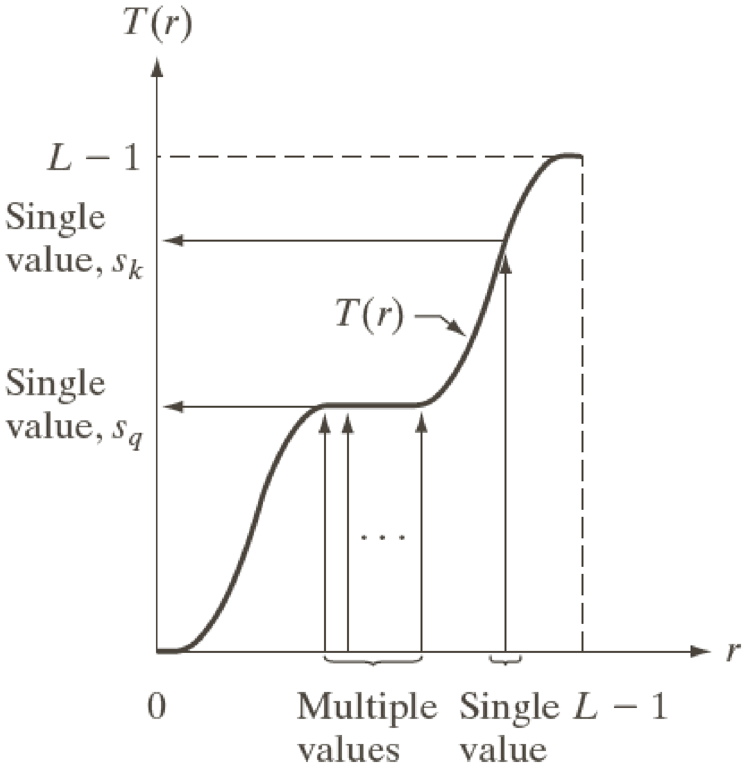
- We need to design a gray value transformation $s = T(r)$, based on the histogram of the input image, which will enhance the image.

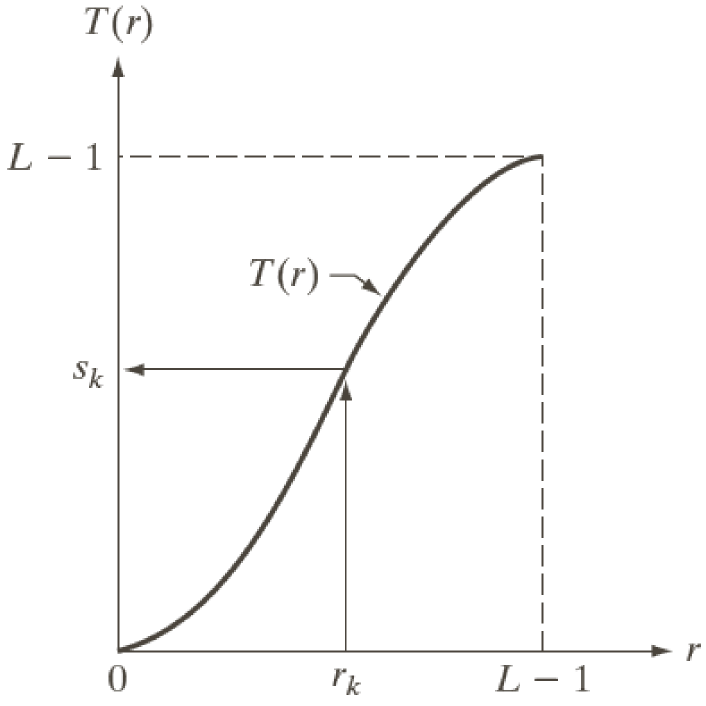
- As before, we assume that:

(1) *T(r)* is a monotonically increasing function for
$0 \leq r \leq 1$(preserves order from black to white).

(2) *T(r)* maps [0,1] into [0,1] (preserves the range of allowed

Gray values).





- Let us denote the inverse transformation by $r = T^{-1}(s)$. We assume that the inverse transformation also satisfies the above two conditions.

- We consider the gray values in the input image and output image as random variables in the interval [0, 1].

- Let $p_{in}(r)$ and $p_{out}(s)$ denote the probability density of the Gray values in the input and output images.

$$p_s(s)ds = p_r(r)dr$$

• If $p_{in}(r)$ and T(r) are known, and $r = T^{-1}(s)$ satisfies condition 1, we can write (result from probability theory):

$$p_{out}(s) = \left[p_{in}(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)}$$

• One way to enhance the image is to design a transformation *T(.)* such that the gray values in the output is uniformly distributed in [0, 1], i.e. $p_{out}(s) = 1, \quad 0 \leq s \leq 1$

• In terms of histograms, the output image will have all gray values in "equal proportion" .

• This technique is called **histogram equalization**.

Next we derive the gray values in the output is uniformly distributed in [0, 1].

- Consider the transformation

$$s = T(r) = \int_0^r p_{in}(w)dw, \qquad 0 \le r \le 1$$

- Note that this is the cumulative distribution function (CDF) of $p_{in}(r)$ and satisfies the previous two conditions.

- From the previous equation and using the fundamental theorem of calculus,

$$\frac{ds}{dr} = p_{in}(r)$$

- Therefore, the output histogram is given by

$$p_{out}(s) = \left[p_{in}(r) \cdot \frac{1}{p_{in}(r)} \right]_{r=T^{-1}(s)} = [1]_{r=T^{-1}(s)} = 1, \qquad 0 \le s \le 1$$

- The output probability density function is uniform, regardless of the input.

- Thus, using a transformation function equal to the CDF of input gray values r, we can obtain an image with uniform gray values.

- This usually results in an enhanced image, with an increase in the dynamic range of pixel values.

How to implement histogram equalization?

Step 1:For images with discrete gray values, compute:

$$p_{in}(r_k) = \frac{n_k}{n} \qquad 0 \le r_k \le 1 \qquad 0 \le k \le L-1$$

L: Total number of gray levels

n_k: Number of pixels with gray value r_k

n: Total number of pixels in the image

Step 2: Based on CDF, compute the discrete version of the previous transformation :

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_{in}(r_j) \qquad 0 \le k \le L-1$$

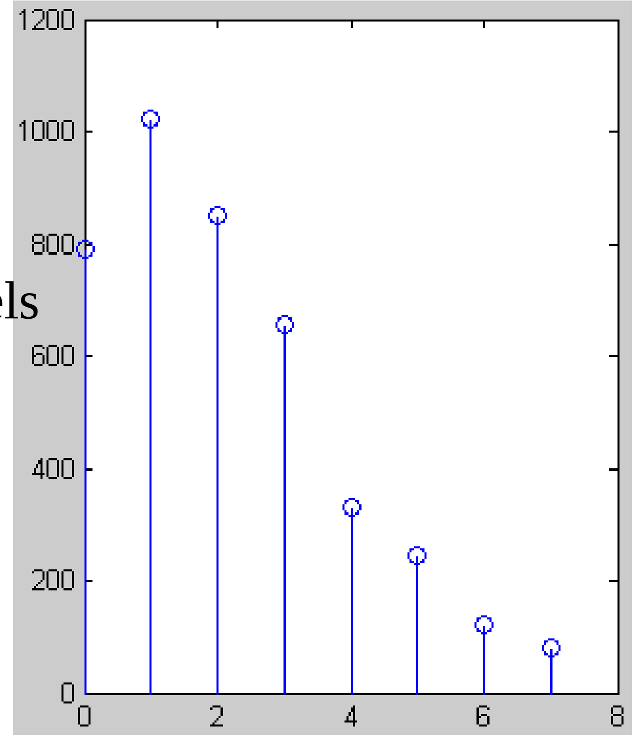
Example:

- Consider an 8-level 64 x 64 image with gray values (0, 1, …, 7). The normalized gray values are (0, 1/7, 2/7, …, 1). The normalized histogram is given below:

k	r_k	n_k	$p(r_k) = n_k/n$
0	0	790	0.19
1	1/7	1023	0.25
2	2/7	850	0.21
3	3/7	656	0.16
4	4/7	329	0.08
5	5/7	245	0.06
6	6/7	122	0.03
7	1	81	0.02

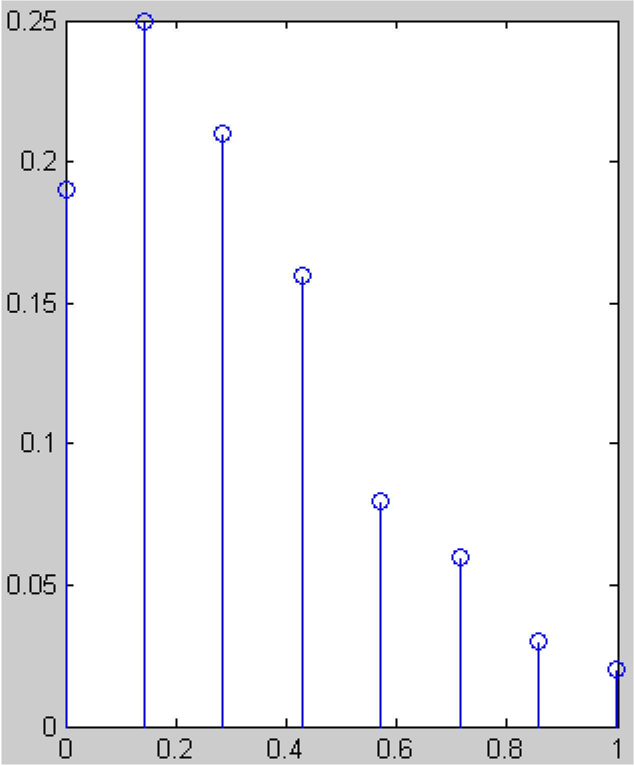
NB: The gray values in output are also (0, 1/7, 2/7, …, 1).

pixels



Gray value

Fraction
of # pixels



Normalized gray value

```
>> clear
>> h=[790 1023 850 656 329 245 122 81];
>> stem(0:7,h)
```

```
>> clear
>> h=[0.19 0.25 0.21 0.16 0.08 0.06 0.03 0.02];
>> stem(0:0.142857:1,h)
```

- Applying the transformation, $s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_{in}(r_j)$ we have

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7\times 0.19 = 1.33 \quad \rightarrow 1$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7\times (0.19 + 0.25) = 3.08 \quad \rightarrow 3$$

$$s_2 = 4.55 \quad \rightarrow 5 \qquad s_3 = 5.67 \quad \rightarrow 6$$

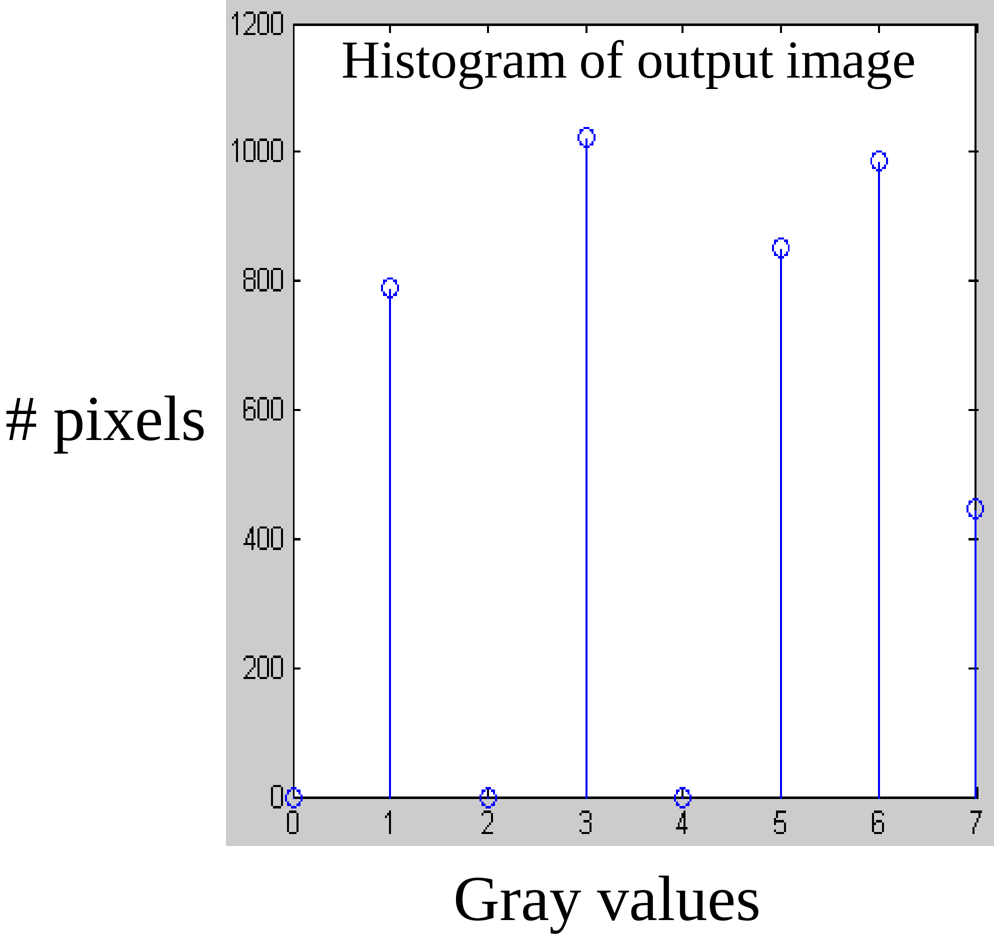
$$s_4 = 6.23 \quad \rightarrow 6 \qquad s_5 = 6.65 \quad \rightarrow 7$$

$$s_6 = 6.86 \quad \rightarrow 7 \qquad s_7 = 7.00 \quad \rightarrow 7$$

- Notice that there are only five distinct gray levels --- (1/7, 3/7, 5/7, 6/7, 1) in the output image. We will relabel them as (s_0, s_1, …, s_4).

- With this transformation, the output image will have histogram

k	s_k	n_k	$p(s_k) = n_k/n$
0	1/7	790	0.19
1	3/7	1023	0.25
2	5/7	850	0.21
3	6/7	985	0.24
4	1	448	0.11

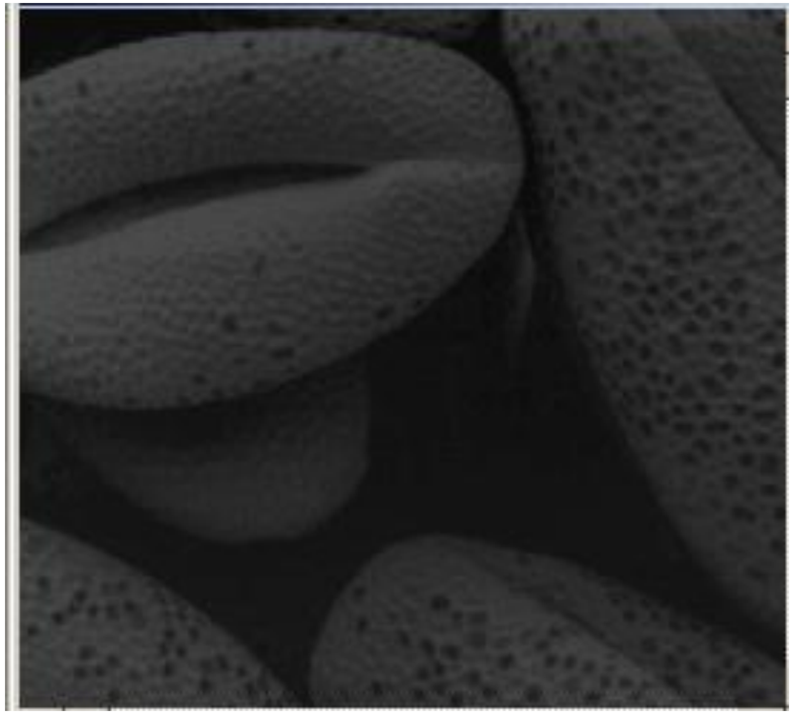


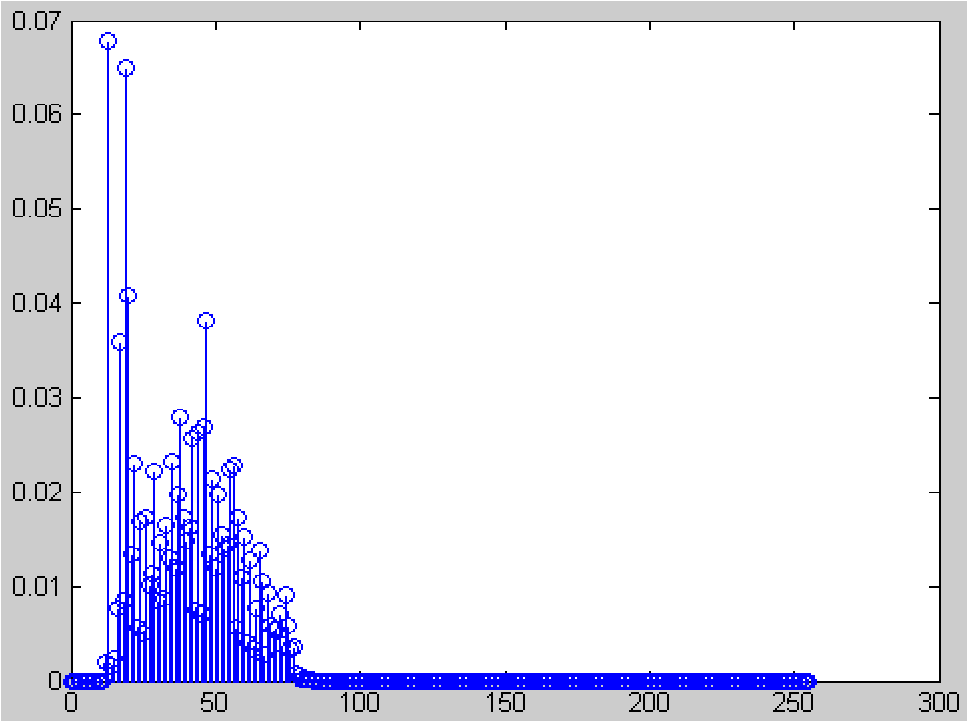

```
>> clear
>> h=[0 790 0 1023 0 850 985 448];
>> stem(0:7,h)
```

• Note that the histogram of output image is only approximately, and not exactly, uniform. This should not be surprising, since there is no result that claims uniformity in the **discrete** case.

Example

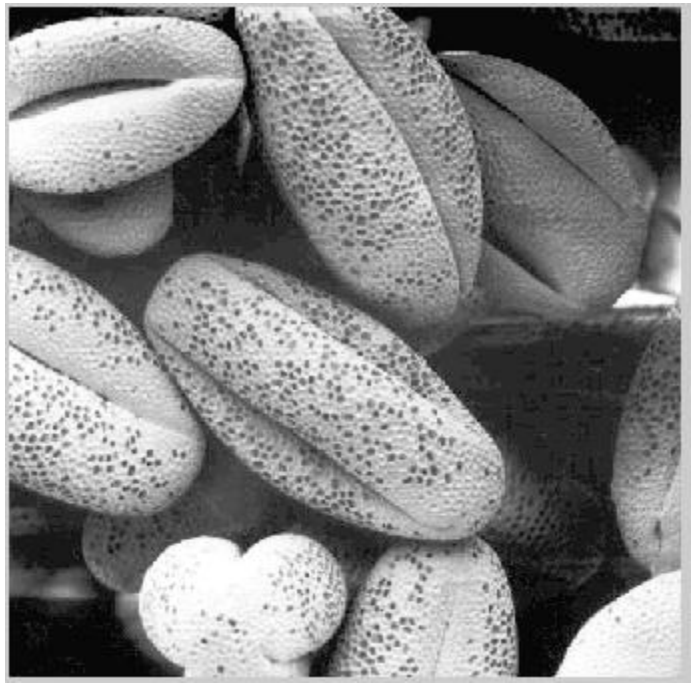
Original image and its histogram

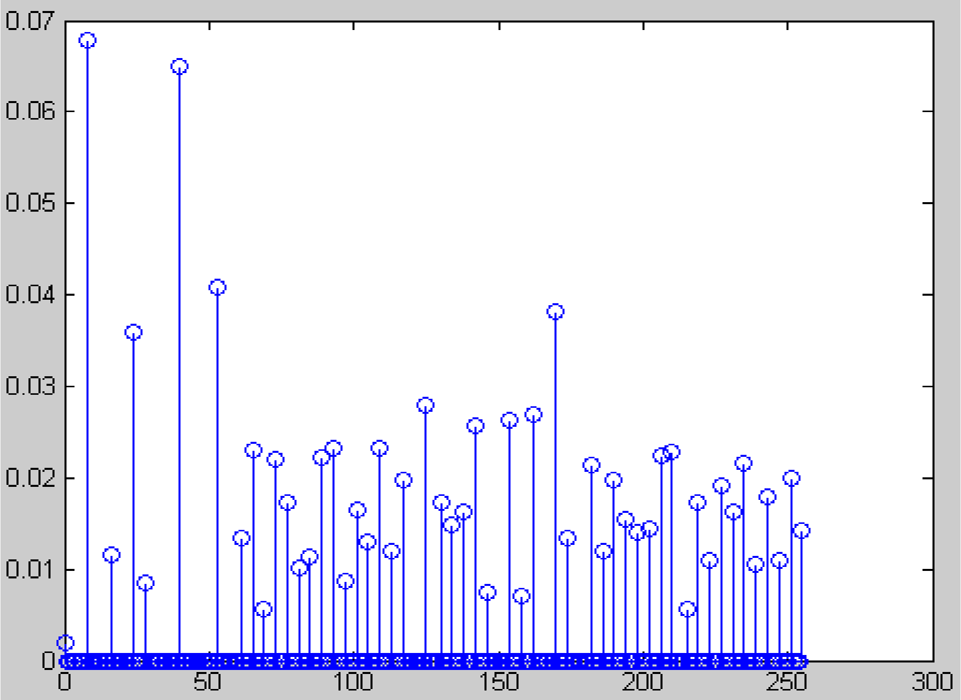





```
>> clear;
[ix,map]=imread('Fig3_15a.jpg');
imshow(ix)
figure;
ix=double(ix);
h=histogram(ix);
stem(0:255,h);
```

Histogram equalized image and its histogram






```
>> clear
>> [ix,map]=imread('Fig3_15a.jpg');
>> imshow(ix);
>> iy=histeq(ix);
>> figure
>> imshow(iy);
>> iy=double(iy);
>> hy=histogram(iy);
>> figure
>> stem(0:255,hy);
```

Gray level(r_k)	No. of Pixels(n_k)	PDF(n_k/N) $p(r_k)$	CDF	(L-1)*CDF	S_k
0	790	0.19	0.19	1.33	1
1	1023	0.25	0.44	3.08	3
2	850	0.21	0.65	4.55	5
3	656	0.16	0.81	5.67	6
4	329	0.08	0.89	6.23	6
5	245	0.06	0.95	6.65	7
6	122	0.03	0.98	6.86	7
7	81	0.02	1.0	7	7
	4096	1			

Modified Image

Gray level	0	1	2	3	4	5	6	7
No. of Pixels	0	790	0	1023	0	850	656+ 329	245+ 122+ 81

Perform the Histogram Specification

Original Image

Gray level	0	1	2	3	4	5	6	7
No. of Pixels	8	10	10	2	12	16	4	2

Desired Image

Gray level	0	1	2	3	4	5	6	7
No. of Pixels	0	0	0	0	20	20	16	8

Histogram Equalization (Input Image)

Gray level(r_k)	No. of Pixels(n_k)	PDF(n_k/N) $P_r(r_k)$	CDF	(L-1)*CDF	H_k
0	8	0.13	0.13	0.91	1
1	10	0.16	0.29	2.03	2
2	10	0.16	0.45	3.15	3
3	2	0.03	0.48	3.36	3
4	12	0.18	0.66	4.62	5
5	16	0.25	0.91	6.37	6
6	4	0.06	0.97	6.79	7
7	2	0.03	1.0	7	7
	64	1			

Histogram Equalization (Target Image)

Gray level(r_k)	No. of Pixels(n_k)	PDF(n_k/N) $P_r(r_k)$	CDF	(L-1)*CDF	S_k
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	20	0.31	0.31	2.17	2
5	20	0.31	0.62	4.34	4
6	16	0.25	0.87	6.09	6
7	8	0.13	1.0	7	7
	64	1			

Mapping

Gray Scale	H	S	Map
0	1	0	4
1	2	0	4
2	3	0	5
3	3	0	5
4	5	2	6
5	6	4	6
6	7	6	7
7	7	7	7

Modified Image

Gray level	0	1	2	3	4	5	6	7
No. of Pixels	0	0	0	0	18	12	28	6

Numeric Example of Equalization

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

(a)

i	$\hat{h}_i$	$\hat{C}_i$	$7\hat{C}_i$
0	1/16	1/16	0
1	3/16	4/16	2
2	4/16	8/16	4
3	4/16	12/16	5
4	3/16	15/16	7
5	1/16	16/16	7
6	0/16	16/16	7
7	0/16	16/16	7

(b)

0	2	5	7
2	4	4	5
2	5	7	7
5	4	7	4

(c)

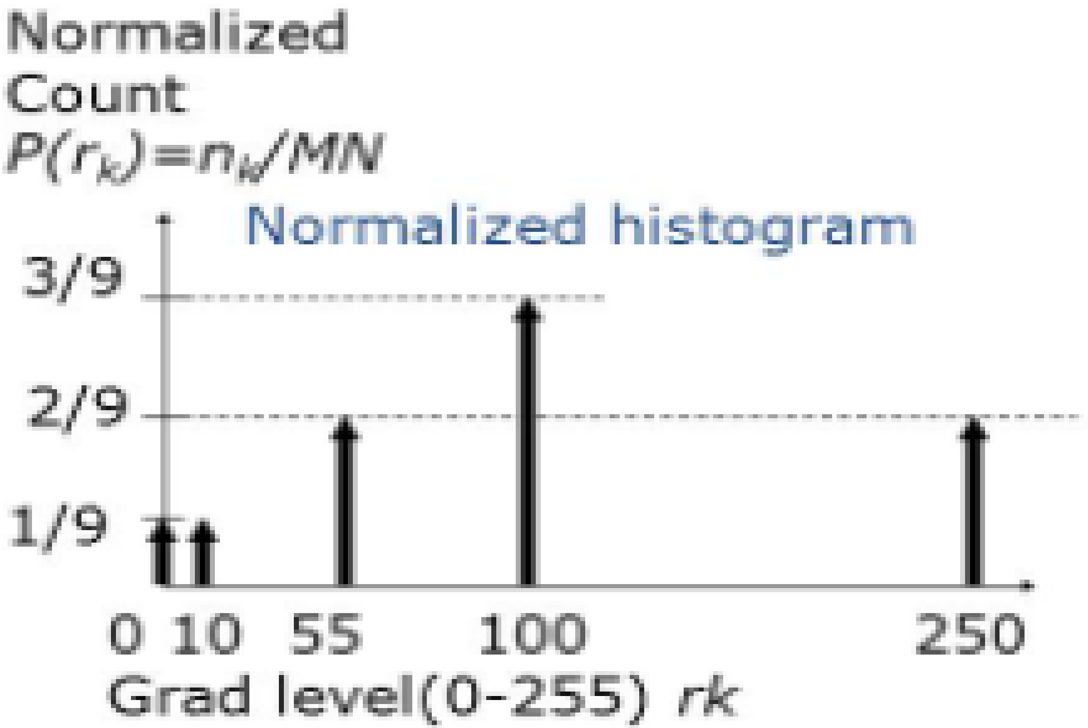
i	$\hat{h}_i$
0	1/16
1	0/16
2	3/16
3	0/16
4	4/16
5	4/16
6	0/16
7	4/16

(d)

Figure 5.9. Numerical example of histogram equalization: (a) a 3-bit image, (b) normalized histogram and CDF, (c) the equalized image, and (d) histogram of the result.

Example: Normalized histogram

- An image of 9 pixels (M=3, N=3)
- K=0,1,2,..,L-1=255.
- L=256 levels



0	250	250
100	100	100
55	55	10

level	count
0	1
10	1
55	2
100	3
250	2

IMAGE- PROCESSING CSE395

Module-2

2.3–Enhancement Using Arithmetic/Logic Operations

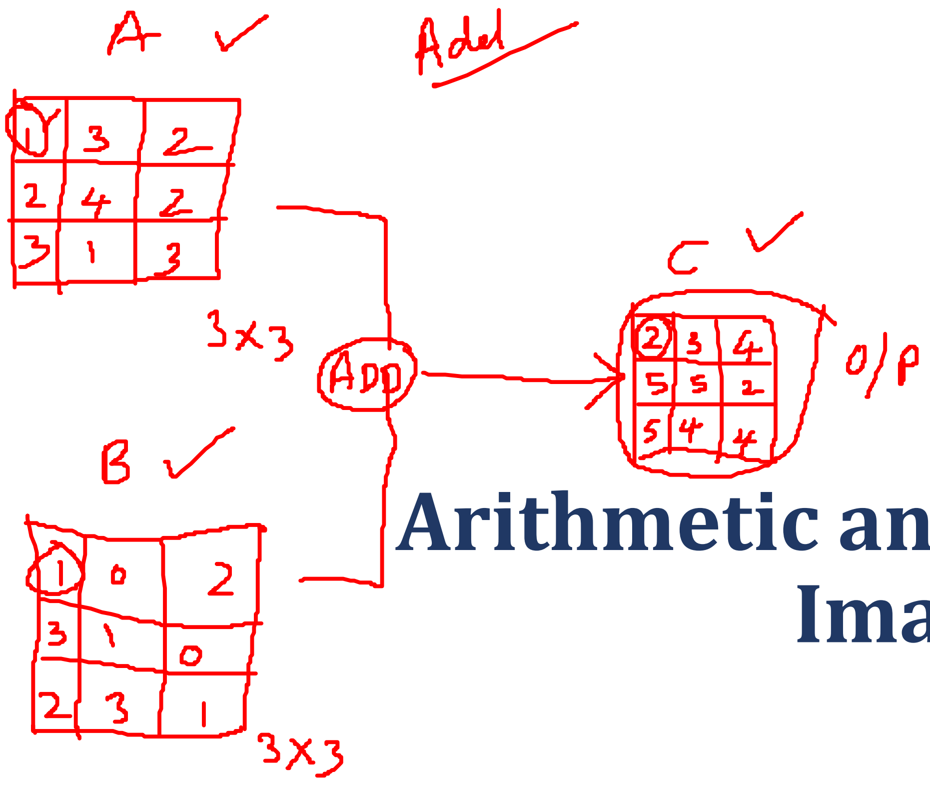


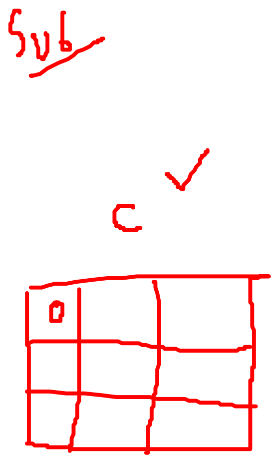
PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

Private University Estd. in Karnataka State by Act No. 41 of 2013

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM





Arithmetic and Logic Operations for Image processing

Enhancement Techniques

- Spatial
  Operates on pixels
- Frequency Domain
  Operates on FT of Image

Image Enhancement Definition

- **Image Enhancement:** is the process that improves the quality of the image for a specific application

Image Enhancement Methods

- **Spatial Domain Methods (Image Plane)**

  Techniques are based on direct manipulation of pixels in an image

- **Frequency Domain Methods**

  Techniques are based on modifying the Fourier transform of the image.

- **Combination Methods**

  There are some enhancement techniques based on various combinations of methods from the first two categories

Spatial Domain Methods

- As indicated previously, the term *spatial domain* refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression:

  $g(x,y) = \boldsymbol{T}\,[f(x,y)]$

  Where f(x,y) in the input image, g(x,y) is the processed image and ***T*** is as operator on f, defined over some neighborhood of (x,y)

- In addition, ***T*** can operate on a set of input images.

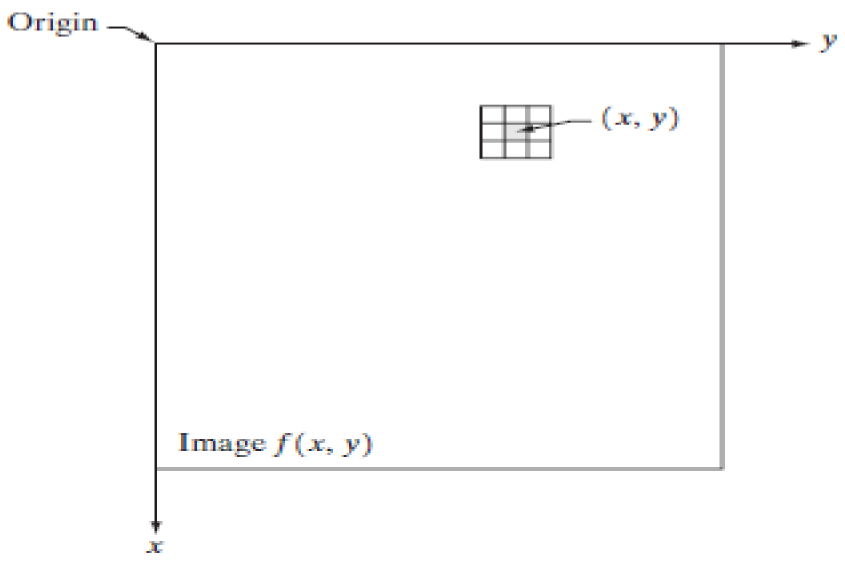


FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.

- The simplest form of ***T***, is when the neighborhood of size 1X1 (that is a single pixel). In this case, g depends only on the value of f at (x,y), and ***T*** becomes a *grey-level* (also called *intensity* or *mapping*) *transformation function* of the form:

$$s = T\,(r)$$

Where, for simplicity in notation, r and s are variables denoting, respectively, the grey level of f(x,y) and g(x,y) at any point (x,y)

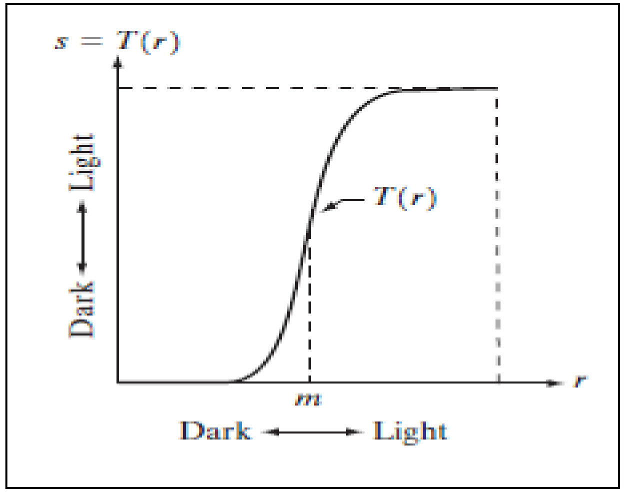
Examples of Enhancement Techniques

- **Contrast Stretching**:

If T(r) has the form as shown in the figure below, the effect of applying the transformation to every pixel of f to generate the corresponding pixels in g would:

Produce higher contrast than the original image, by:

- Darkening the levels below m in the original image
- Brightening the levels above m in the original image

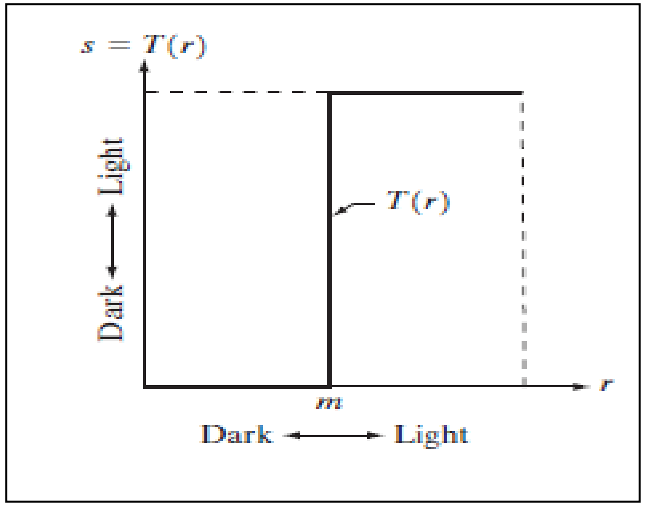


So, Contrast Stretching: is a simple image enhancement technique that improves the contrast in an image by 'stretching' the range of intensity values it contains to span a desired range of values. Typically, it uses a linear function

Examples of Enhancement Techniques

- **Thresholding**

Is a limited case of contrast stretching, it produces a two-level (binary) image.



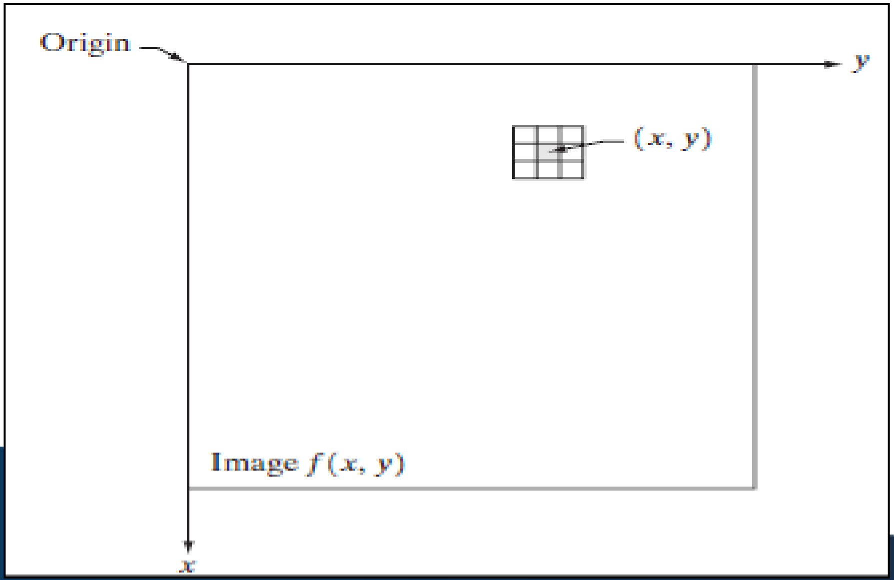
Some fairly simple, yet powerful, processing approaches can be formulated with grey-level transformations. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category often are referred to as *point processing*.

Examples of Enhancement Techniques

Larger neighborhoods allow considerable more flexibility. The general approach is to use a function of the values of *f* in a predefined neighborhood of (x,y) to determine the value of *g* at (x,y).

One of the principal approaches in this formulation is based on the use of so-called *masks* (also referred to as *filters*)

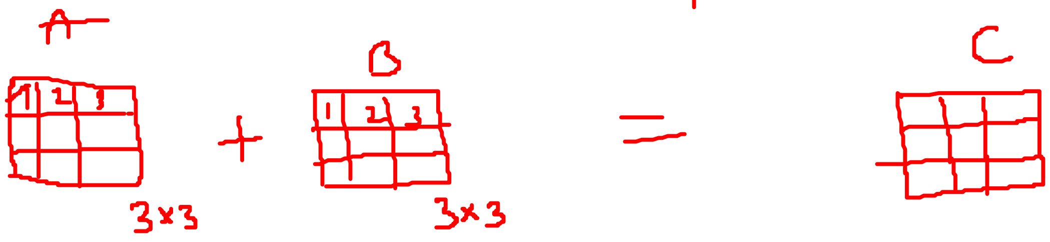
So, a mask/filter: is a small (say 3X3) 2-D array, such as the one shown in the figure, in which the values of the mask coefficients determine the nature of the process, such as *image sharpening*. Enhancement techniques based on this type of approach often are referred to as *mask processing* or *filtering*.

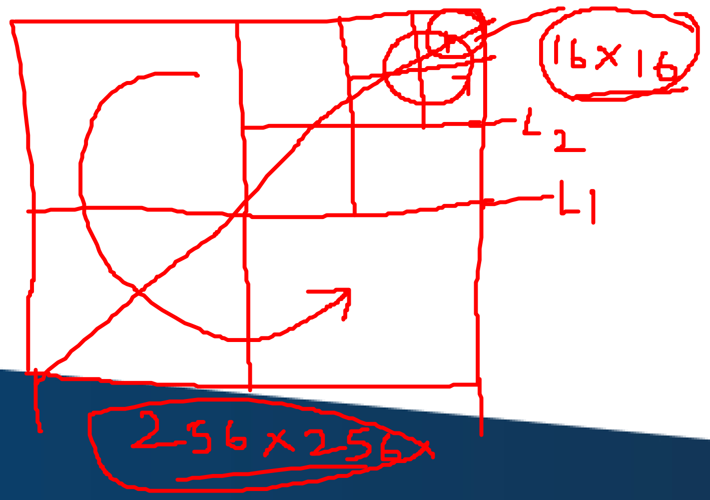


Arithmetic Operations

- Arithmetic operations between images are array operations. The four arithmetic operations are denoted as

$$s(x,y) = f(x,y) + g(x,y)$$
$$d(x,y) = f(x,y) - g(x,y)$$
$$p(x,y) = f(x,y) \times g(x,y)$$
$$v(x,y) = f(x,y) \div g(x,y)$$





Example: Addition of Noisy Images for Noise Reduction

Noiseless image: f(x,y)

Noise: n(x,y) (at every pair of coordinates (x,y), the noise is uncorrelated and has zero average value)

Corrupted image: g(x,y)

$$\boxed{g(x,y)} = f(x,y) + n(x,y)$$

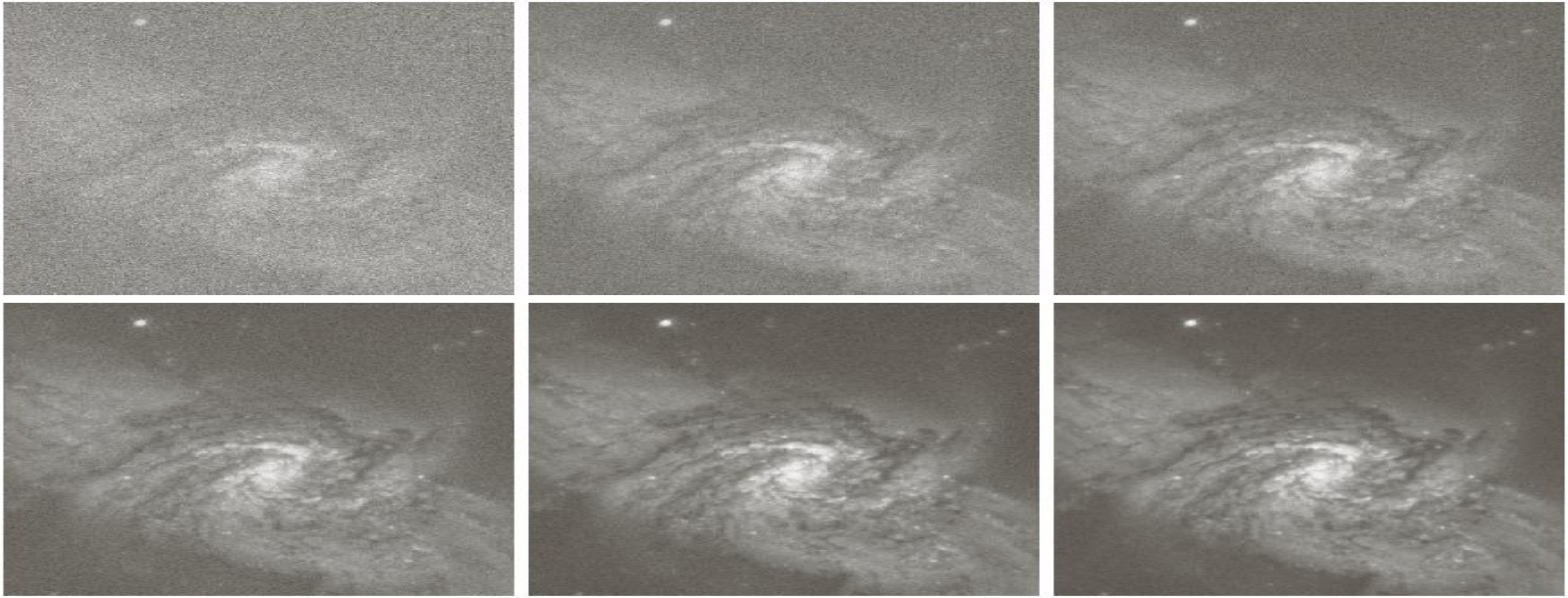
Reducing the noise by adding a set of noisy images, $\{g_i(x,y)\}$

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

Example: Addition of Noisy Images for Noise Reduction

- In astronomy, imaging under very low light levels frequently causes sensor noise to render single images virtually useless for analysis.

- In astronomical observations, similar sensors for noise reduction by observing the same scene over long periods of time. Image averaging is then used to reduce the noise.



a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

An Example of Image Subtraction: Mask Mode Radiography

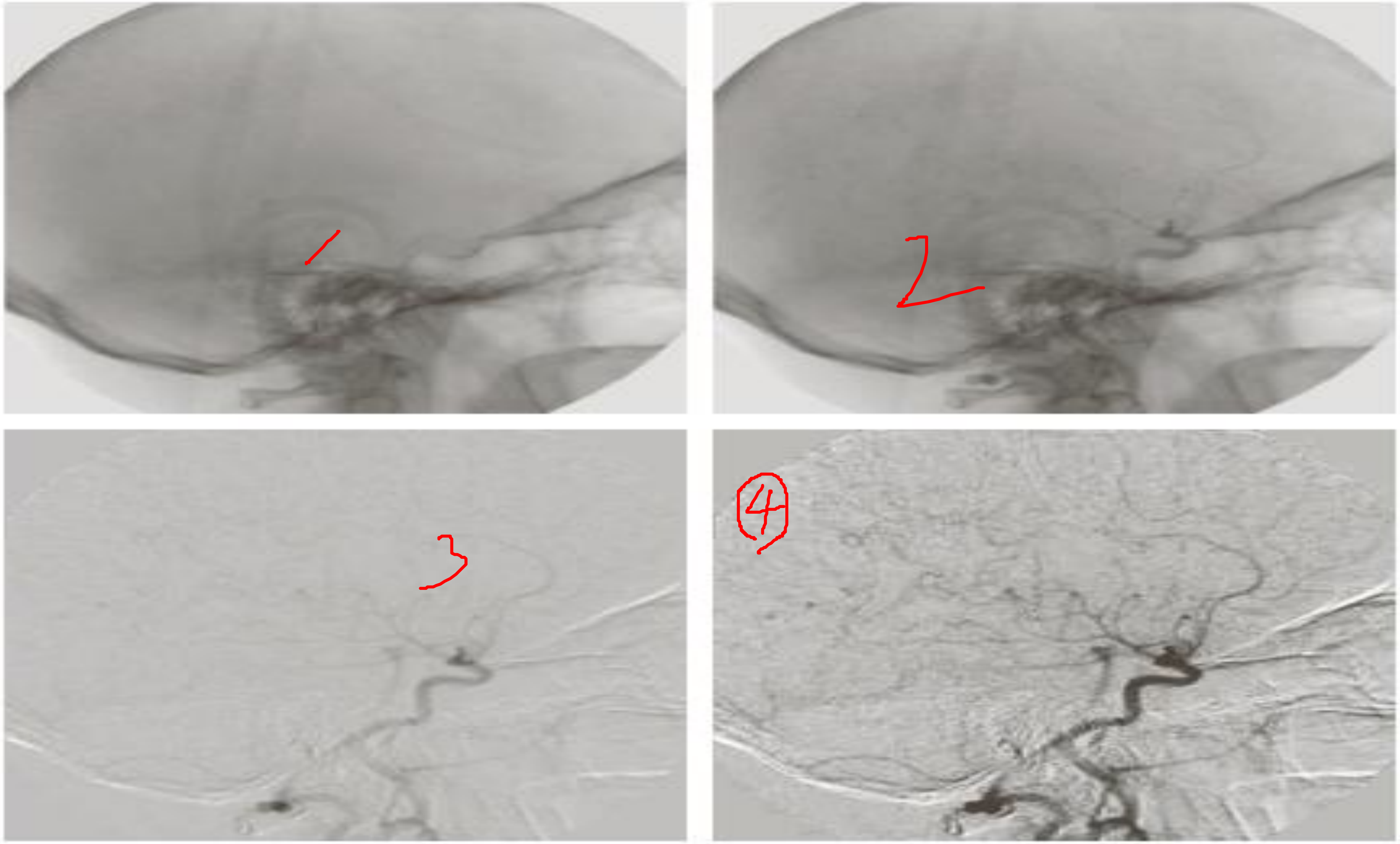
Mask h(x,y): an X-ray image of a region of a patient's body

Live images f(x,y): X-ray images captured at TV rates after injection of the contrast medium

Enhanced detail g(x,y)

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

The procedure gives a movie showing how the contrast medium propagates through the various arteries in the area being observed.



a b
c d

FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

An Example of Image Multiplication

FIGURE 2.29 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Arithmetic and Logic Operations

- Arithmetic and logic operations are often applied a preprocessing steps in image analysis in order to combine images in various way.
- Addition, subtraction, division and multiplication comprise the arithmetic operation, while AND , OR, and NOT make up the logic operations.
- These operation performed on two image , except the NOT logic operation which require only one image, and are done on a pixel by pixel basis.

Image Arithmetic

- For input images f_1 and f_2 and some function Op:

$$g(x, y) = \mathrm{Op}\left(f_1(x, y), f_2(x, y)\right)$$

  The operator is applied pairwise to each pixel in the images.
- Pseudocode:

```
for all pixel positions x, y:
  out[x,y] = func( image1[x,y] , image2[x,y]
)
```

- Possibilities: addition, subtraction, and, or, ...

Arithmetic operations

- Addition: $g(x, y) = f_1(x, y) + f_2(x, y)$
- Subtraction: $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication: $g(x, y) = f_1(x, y) \cdot f_2(x, y)$
- Division: $g(x, y) = f_1(x, y)/f_2(x, y)$

Addition/Blending

Used to create double-exposures or composites:

$$g(x, y) = f_1(x, y) + f_2(x, y)$$

A B

Can also do a weighted *blend*:

$$g(x, y) = \alpha_1 f_1(x, y) + \alpha_2 f_2(x, y)$$

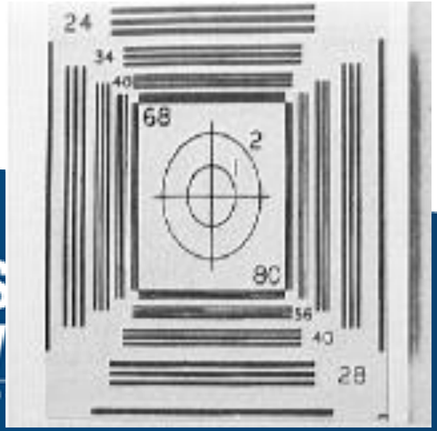
Figure 3.2-6 Image Addition Examples. This example shows one step in the *image morphing* process where an increasing percentage of the second image is slowly added to the first, and a geometric transformation is usually required to align the images. a) first original, b) second original, c) addition of images (a) and (b). This example shows adding noise to an image which is often useful for developing image restoration models. d) original image, e) Gaussian noise, variance = 400, mean = 0, f) addition of images (d) and (e).

a)

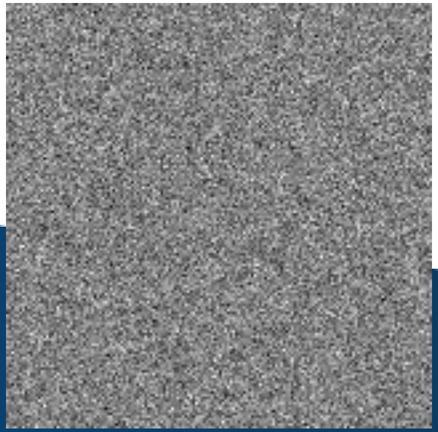
b)

c)

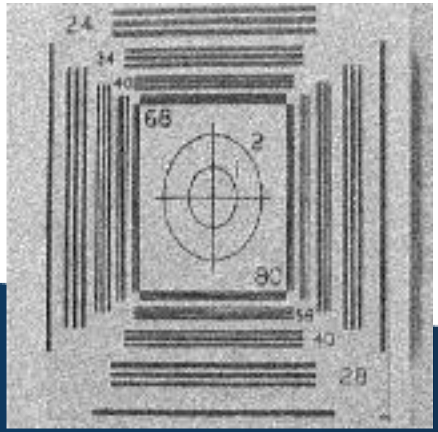
d)



e)



f)



Arithmetic operations

- Addition: $g(x, y) = f_1(x, y) + f_2(x, y)$
- Subtraction: $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication: $g(x, y) = f_1(x, y) \cdot f_2(x, y)$
- Division: $g(x, y) = f_1(x, y)/f_2(x, y)$

Subtraction

- Useful for finding changes between two images

$$g(x, y) = f_1(x, y) - f_2(x, y)$$

- Sometimes more useful to use *absolute difference*

$$g(x, y) = |f_1(x, y) - f_2(x, y)|$$

- What's changed?

Subtraction

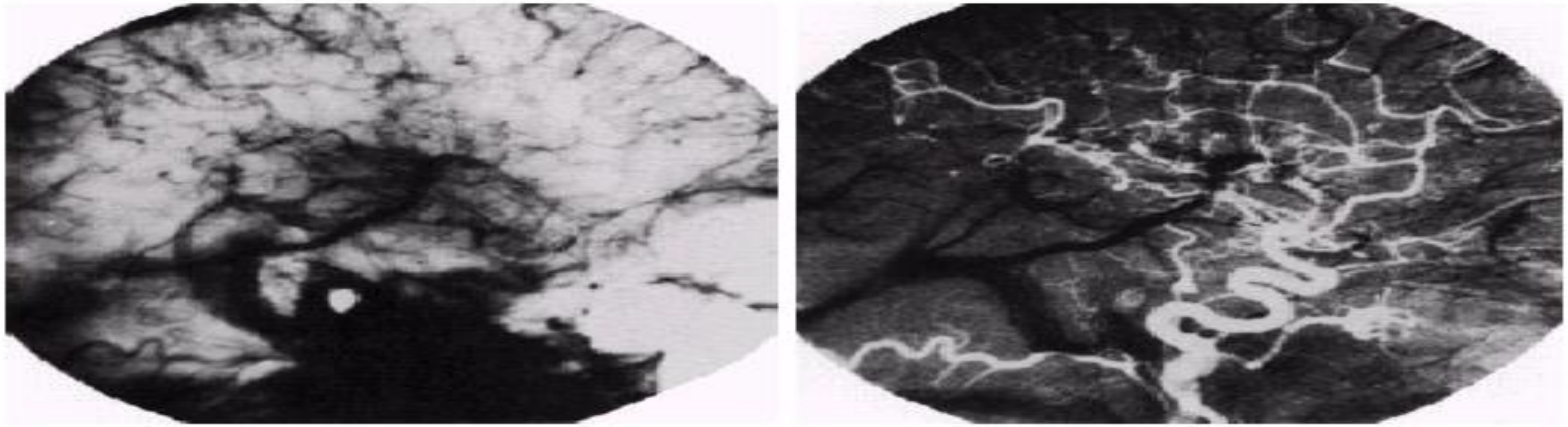
- Subtraction of two image is often used to detect motion.
- Consider the case where nothing has changed in a scene; the image resulting from subtraction of two sequential image is filled with zeros - a black image.
- If something has moved in the scene, subtraction produce a nonzero result at the location of movement.

Subtraction

- Medical imaging often uses this type of operation to allow the doctor to more readily see changes which are helpful in the diagnosis.
- The technique is also used in law enforcement and military applications; for example, to find an individual in a crowd or to detect changes in a military installation.

Digital Subtraction Angiography

1. Take an x-ray
2. Inject patient with a radio-opaque dye (and tell them not to move!)
3. Take another x-ray
4. Subtract the two



a b

FIGURE 3.29 Enhancement b image subtracti (a) Mask image (b) An image (taken after injection of a contrast mediu into the bloodstream) w mask subtracte out.

Arithmetic operations

- Addition: $g(x, y) = f_1(x, y) + f_2(x, y)$
- Subtraction: $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication: $g(x, y) = f_1(x, y) \cdot f_2(x, y)$
- Division: $g(x, y) = f_1(x, y)/f_2(x, y)$

- Addition: $g(x, y) = f_1(x, y) + f_2(x, y)$
- Subtraction: $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication: $g(x, y) = f_1(x, y) \cdot f_2(x, y)$
- Division: $g(x, y) = f_1(x, y) / f_2(x, y)$

Multiplication n Division

- used to adjust the brightness of an image.
  - is done on a pixel by pixel basis and the options are to multiply or divide an image by a constant value, or by another image.
  - Multiplication of the pixel value by a value greater than one will brighten the image (or division by a value less than 1), and division by a factor greater than one will darken the image (or multiplication by a value les than 1).
- Brightness adjustment by a constant is often used as a preprocessing step in image enhancement and is shown in Figure 3.2.8.

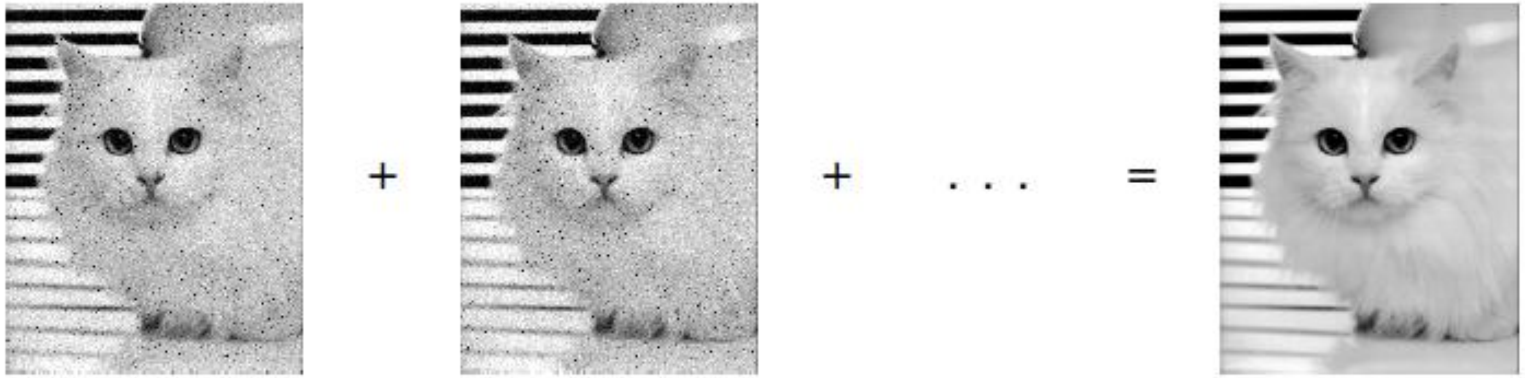
Figure 3.2-8 Image Division. a) original image, b) image divided by a value less than 1 to brighten, c) image divided a value greater than 1 to darken

a) b) c)

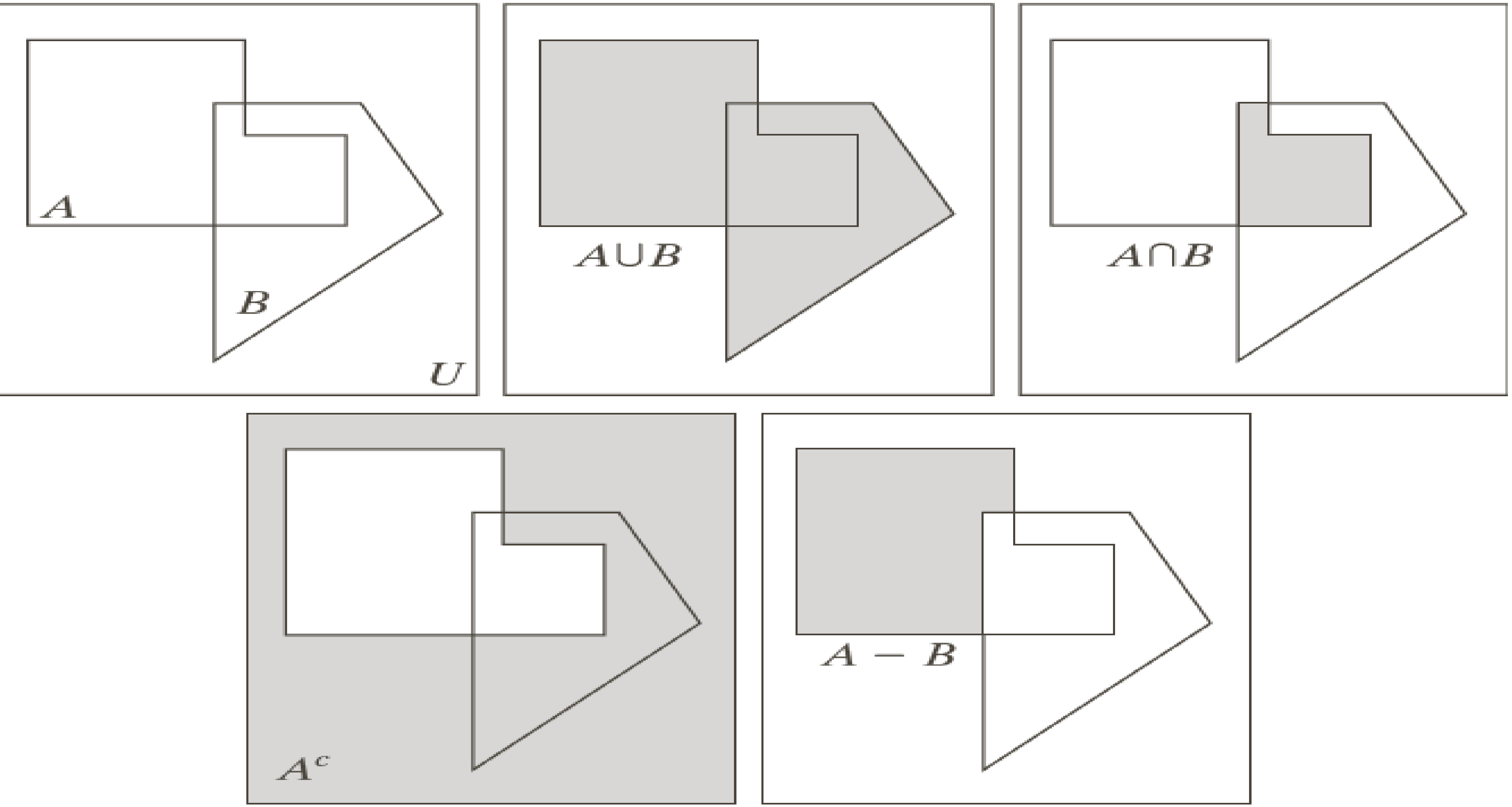
Image Averaging

Idea:
Average multiple pictures of the same scene to reduce noise

Similar in principle to acquiring the image for a longer duration.



Set and Logical Operations



a b c
d e

FIGURE 2.31
(a) Two sets of coordinates, A and B, in 2-D space. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B. In (b)–(e) the shaded areas represent the member of the set operation indicated.

Set and Logical Operations

- Let A be the elements of a gray-scale image

  The elements of A are triplets of the form (x, y, z), where x and y are spatial coordinates and z denotes the intensity at the point (x, y).

$$A = \{(x, y, z) \mid z = f(x, y)\}$$

- The complement of A is denoted A^c

$$A^c = \{(x, y, K - z) \mid (x, y, z) \in A\}$$

$$K = 2^k - 1; k \text{ is the number of intensity bits used to represent } z$$

Set and Logical Operations

- The union of two gray-scale images (sets) A and B is defined as the set

$$A \cup B = \{\max_{z}(a,b) \mid a \in A, b \in B\}$$

Set and Logical Operations

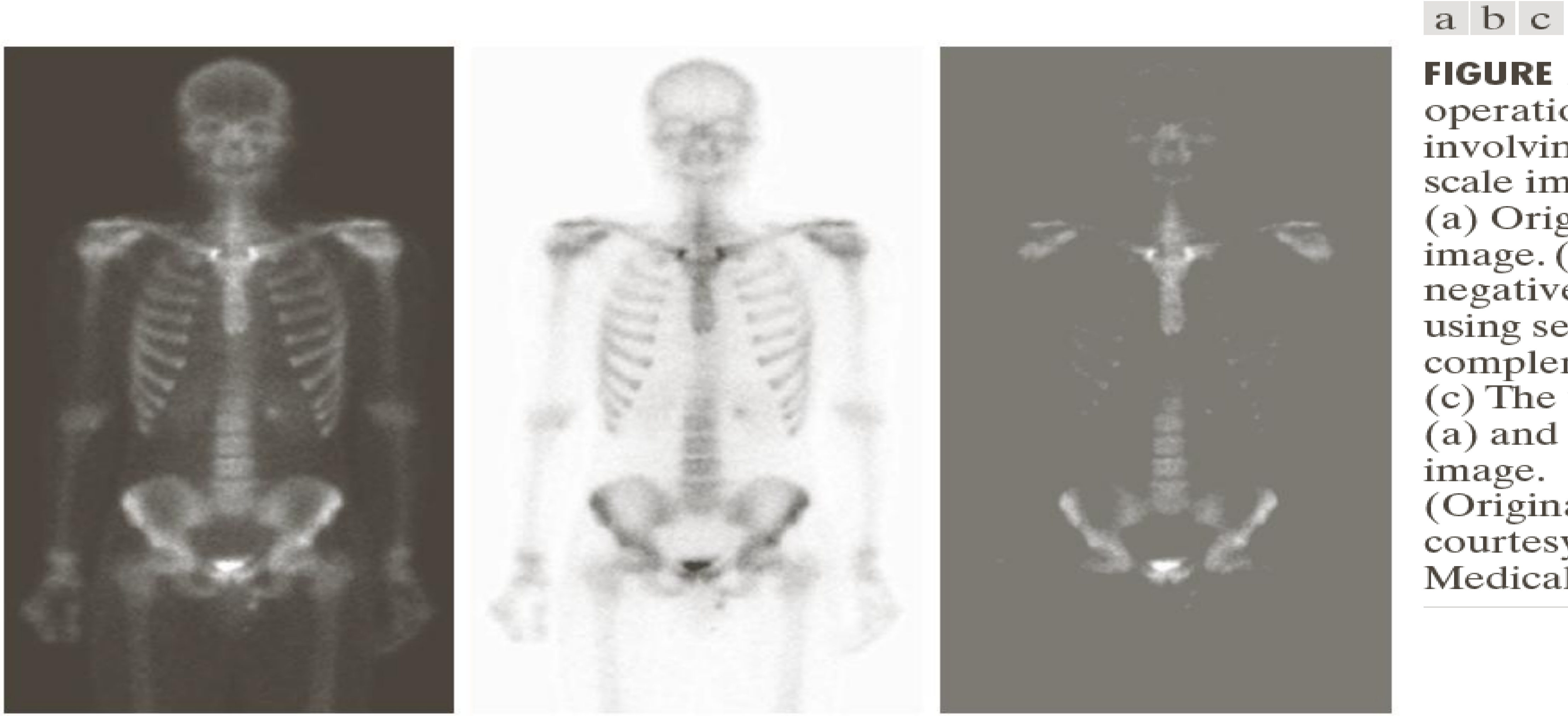


FIGURE 2.32 Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

Set and Logical Operations

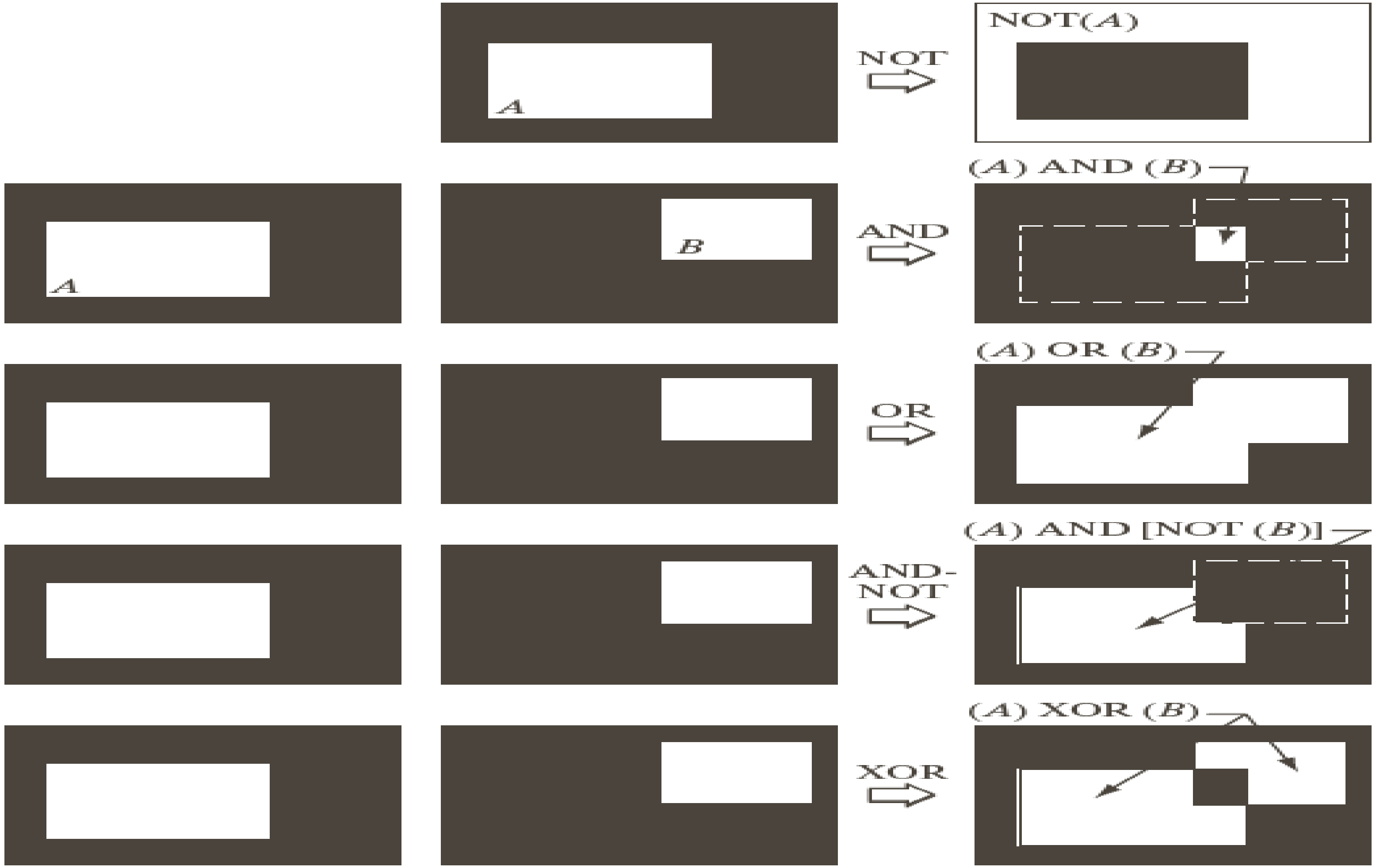


FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Logic operations

- The logic operations AND, 0R and NOT operate in a bit-wise fashion on pixel data.
- Example
  - performing a logic AND on two images. Two corresponding pixel values are 111_{10} in one image and 88_{10} in the second image. The corresponding bit string are:

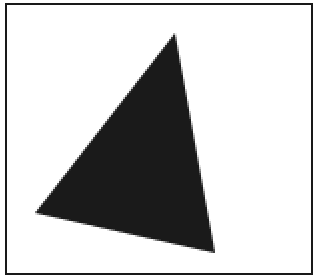
$111_{10} = 01101111_2$ $\qquad\qquad$ $88 = 01011000_2$

$$
\begin{array}{ll}
 & 01101111_2 \\
\text{AND} & \underline{01011000_2} \\
 & 01001000_2
\end{array}
$$

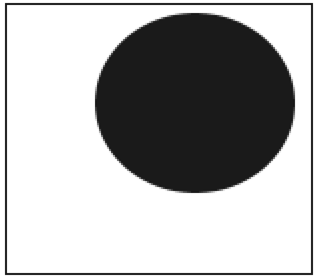
Master Layout 1

Image after logical operation is done

Original Images



IMG_1



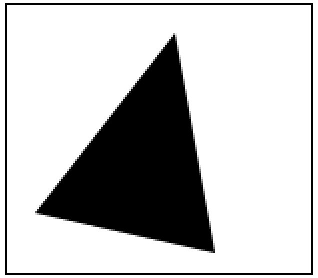
IMG_2

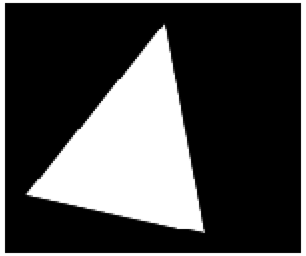
Give a dropdown box to select the operation

- The operations are: NOT, AND, OR, XOR, NOT-AND
- Give Start, Pause, Reset buttons

Step 1:

NOT

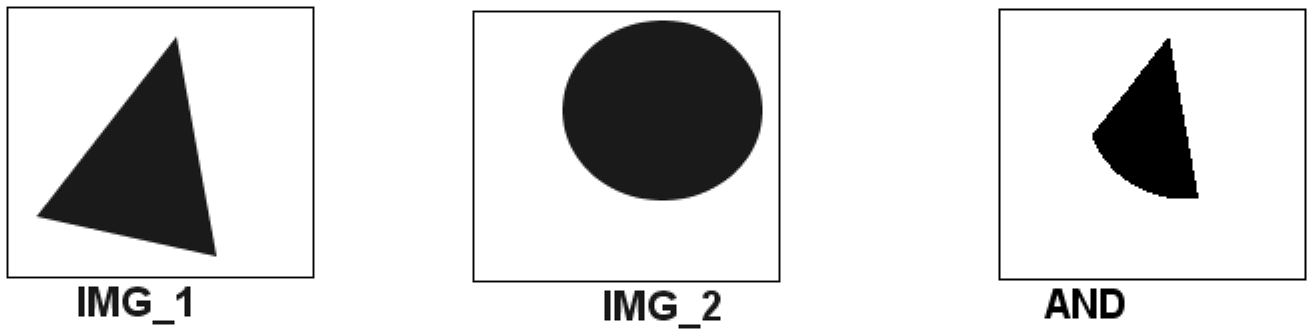




NOT

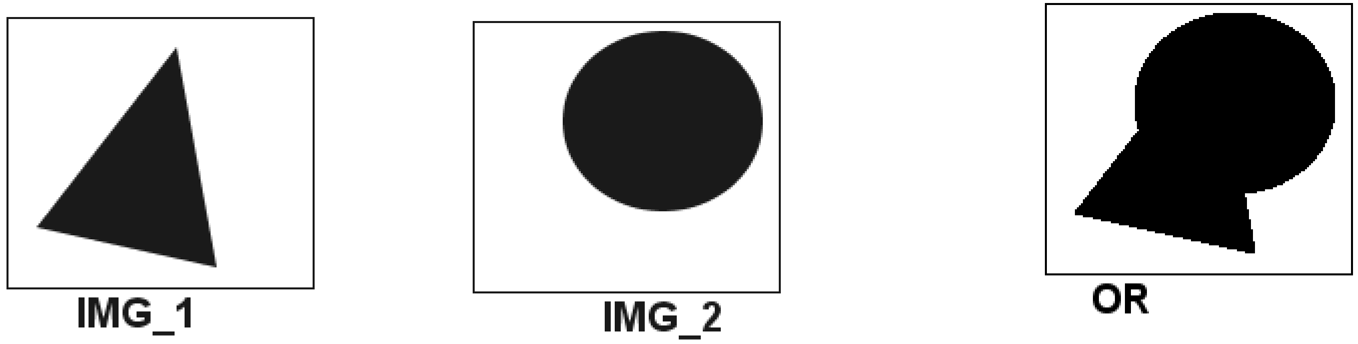
Step 2:

AND



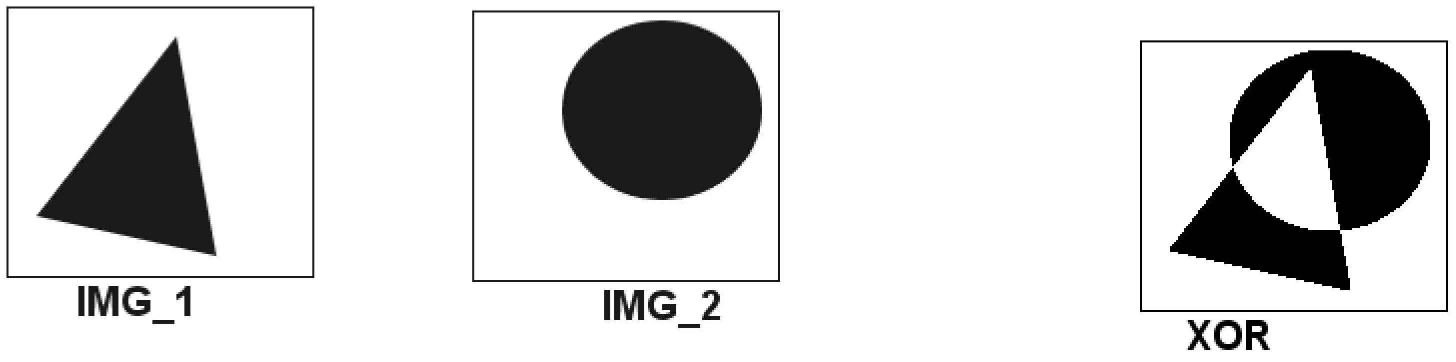
Step 3:

OR



Step 4:

XOR



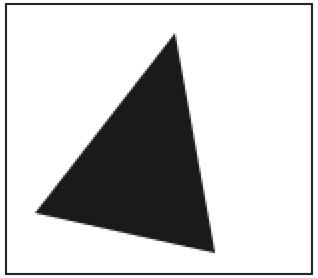
1

2

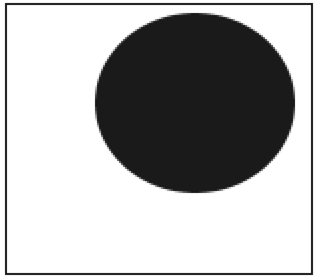
3

Step 6:

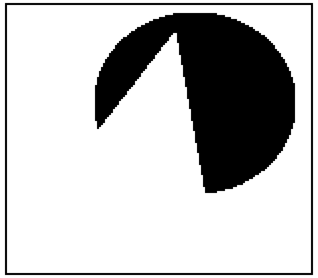
NOT-AND



IMG_1



IMG_2



NOT-AND

- The logic operations AND and OR are used to combine the information in two images.
- This may be done for special effects but a more useful application for image analysis is to perform a masking operation.
- AND and OR can be used as a simple method to extract a ROI from an image.

- For example, a white mask ANDed with an image will allow only the portion of the image coincident with the mask to appear in the output image, with the background turned black; and a black mask ORed with an image will allow only the part f the image corresponding to the black mask to appear in the output image, but will turn the return of the image white.

$f(x.y)$

- This process is called *image masking*

Figure 3.2-10 Image Masking. a) Original image, b) image mask for AND operation, c) Resulting image from (a) AND (b), d) image mask for OR operation, created by performing a NOT on mask (b), e) Resulting image from (a) OR (d).

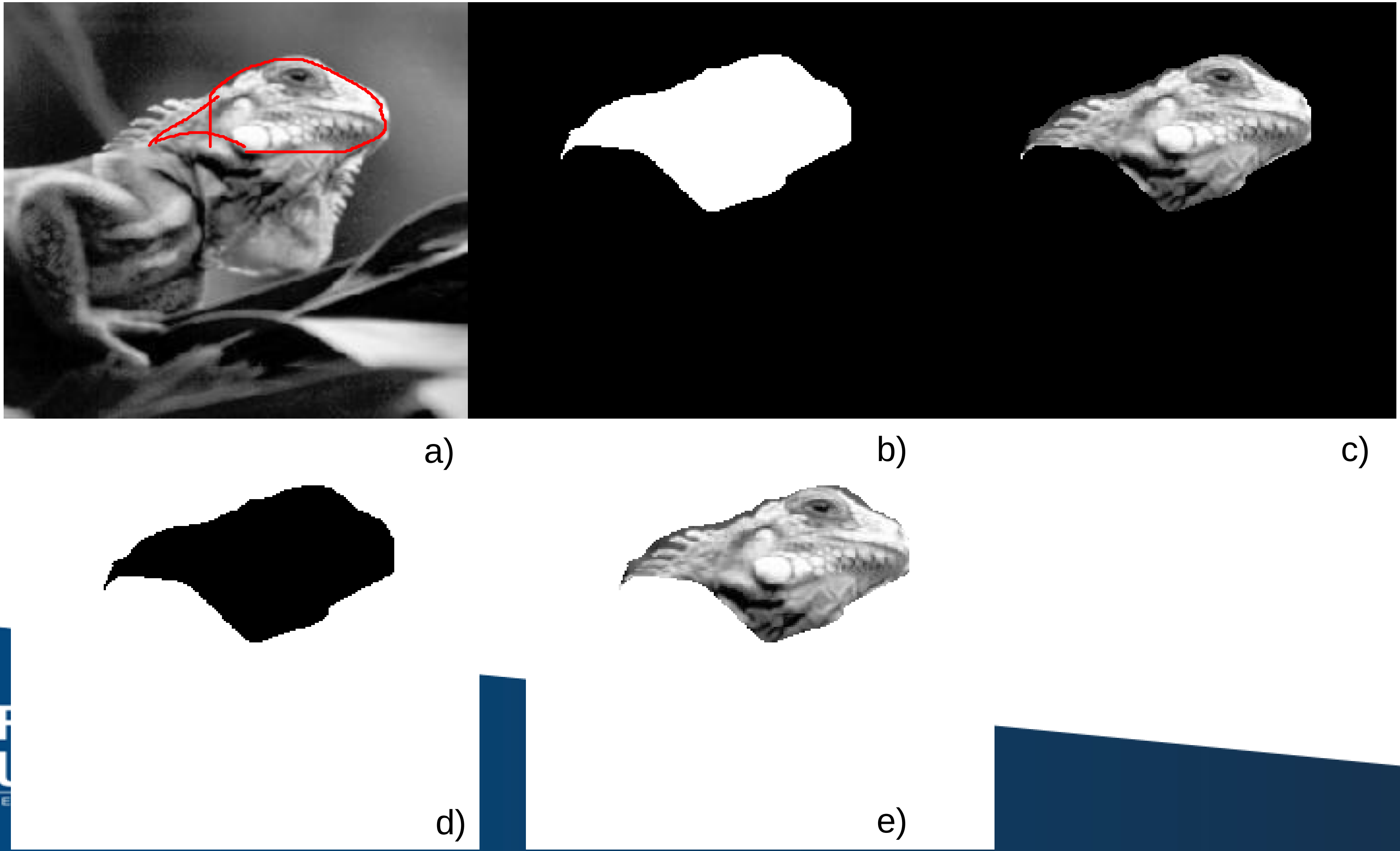


Figure 3.2-11 Complement Image – NOT Operation. a) Original, b) NOT operator applied to the image

a)

b)

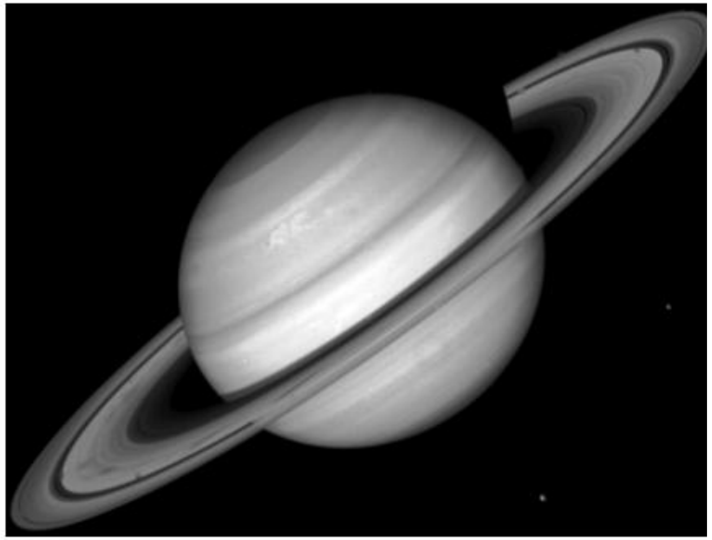
IMAGE- PROCESSING CSE395

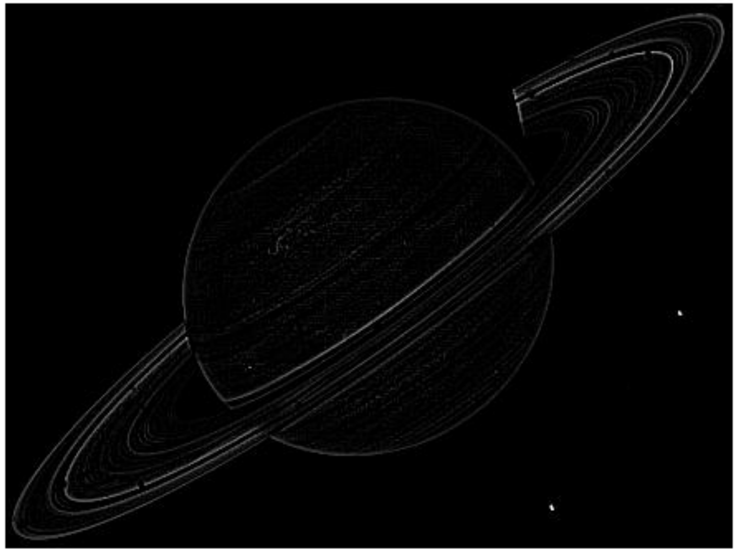
Module-2

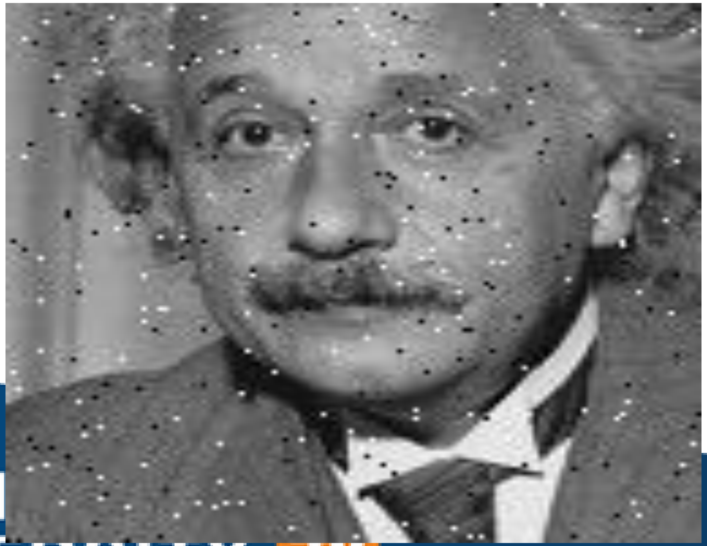
2.4–Basics of Spatial Filtering,

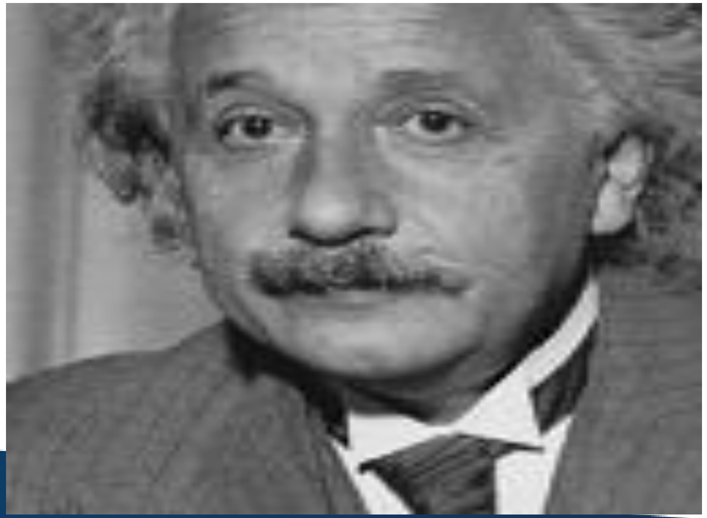
OUTLINE

- Fundamentals Spatial Filtering
- Smoothing Spatial Filters
  - Smoothing filters are used for blurring and for noise reduction
- Sharpening Spatial Filters
  - Highlight fine detail or enhance detail that has been blurred









Spatial Filtering

- The word “filtering” has been borrowed from the frequency domain.

- Filters are classified as:
  Low-pass (i.e., preserve low frequencies)
  High-pass (i.e., preserve high frequencies)
  Band-pass (i.e., preserve frequencies within a band)
  Band-reject (i.e., reject frequencies within a band)

Spatial Filtering

❖ ***Background :***

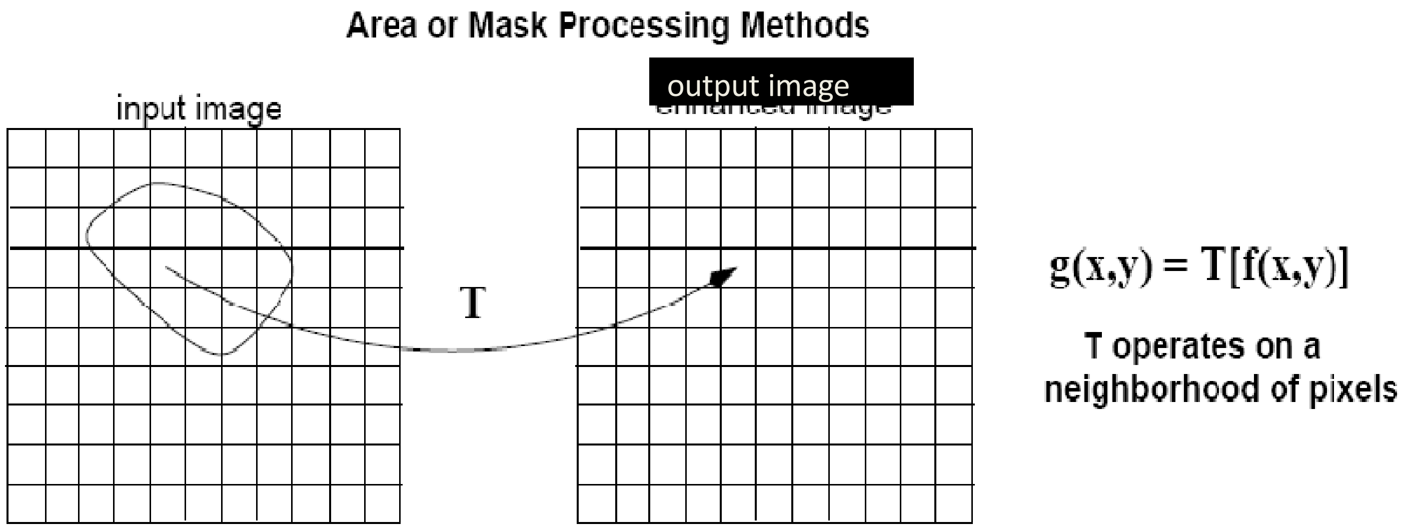
- ➢ Filter term in "Digital image processing" is referred to the subimage
- ➢ There are others term to call subimage such as mask, kernel, template, or window
- ➢ The value in a filter subimage are referred as coefficients, rather than pixels.

❖ ***Basics of Spatial Filtering :***

- ➢ The concept of filtering has its roots in the use of the Fourier transform for signal processing in the so-called frequency domain.
- ➢ Spatial filtering term is the filtering operations that are performed directly on the pixels of an image

❖ *Definition*

- Spatial filtering are defined by:
  - (1) A **neighborhood**
  - (2) A **predefined operation** that is performed on the pixels inside the neighborhood



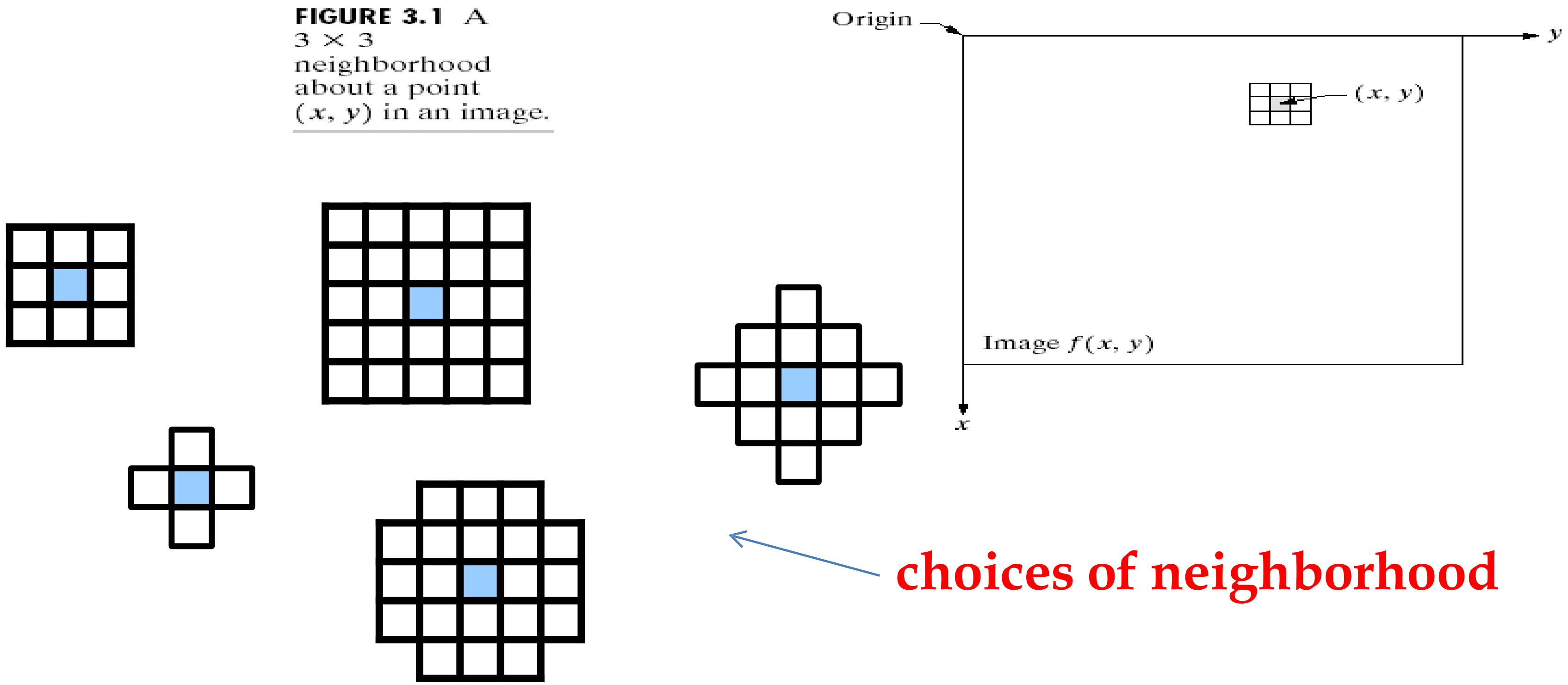
Operation with 3x3 Filter

- 3 x 3 Neighborhood / Mask / Window / Template:

	(y - 1)	y	(y + 1)	Y
(x - 1)	w(-1,-1) f(x-1, y-1)	w(-1,0) f(x-1, y)	w(-1,1) f(x-1, y+1)	
x	w(0,-1) f(x, y-1)	w(0,0) f(x, y)	w(0,1) f(x, y+1)	
(x + 1)	w(1,-1) f(x+1, y-1)	w(1,0) f(x+1, y)	w(1,1) f(x+1, y+1)	
X				

❖Spatial Neighborhood

FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image.



choices of neighborhood

- ❖ Typically, the neighborhood is **rectangular** and its size is much smaller than that of *f(x,y)*
  - e.g., 3x3 or 5x5

❖ *Mechanics of spatial filtering*

- The process consists simply of moving the filter mask from point to point in an image.

- At each point (x,y) the response of the filter at that point is calculated using a predefined relationship

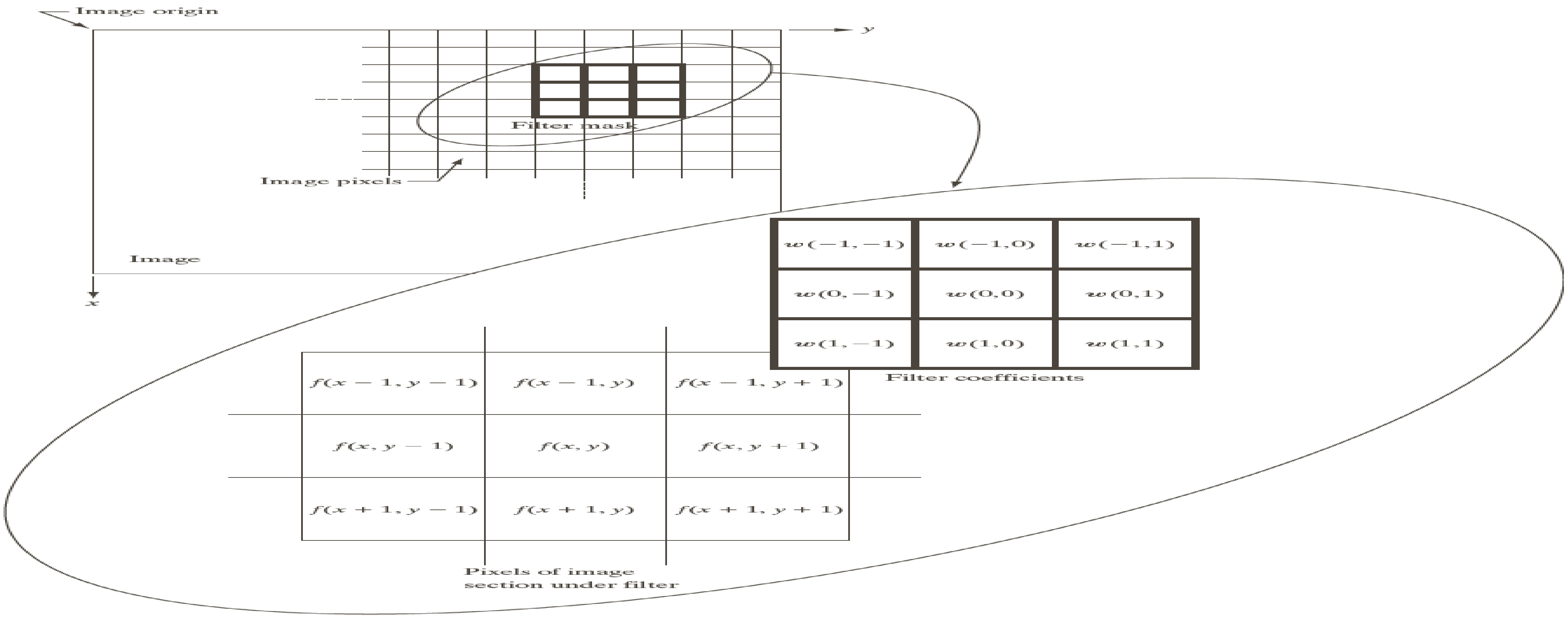


Image origin
y
Filter mask
Image pixels
Image
x
w(−1,−1)
w(−1,0)
w(−1,1)
w(0,−1)
w(0,0)
w(0,1)
w(1,−1)
w(1,0)
w(1,1)
Filter coefficients
f(x − 1, y − 1)
f(x − 1, y)
f(x − 1, y + 1)
f(x, y − 1)
f(x, y)
f(x, y + 1)
f(x + 1, y − 1)
f(x + 1, y)
f(x + 1, y + 1)
Pixels of image
section under filter

❖ *Type*

- Linear spatial filtering
- Nonlinear spatial filtering

➢ A filtering method is linear when the output is a weighted sum of the input pixels

w1	w2	w3
w4	w5	w6
w7	w8	w9

$$z5' = R = w1z1 + w2z2 + \ldots + z9w9$$

➢ Methods that do not satisfy the above property are called non-linear.

z1	z2	z3
z4	z5	z6
z7	z8	z9

$$z_5' = max(z_k, k = 1, 2, \ldots, 9)$$

Linear Spatial Filtering Methods

- Two main linear spatial filtering methods:
  - Correlation
  - Convolution

The correlation of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

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

The convolution of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

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

Correlation & Convolution

- Correlation & Convolution are two closely related concepts used in linear spatial filtering.
- *Correlation*: It is a process of moving a filter mask over an image & computing the sum of products at each location.
- *Convolution*: Here, the mechanics are same, except that the filter is first rotated by 180˚.
- Correlation & Convolution are function of displacement. Correlation & Convolution are exactly same if the filter mask is symmetric.
- 1D correlation and convolution of a filter with a discrete unit impulse is shown below.

Correlation

Origin f w

(a) 0 0 0 1 0 0 0 0   1 2 3 2 8

(b) 0 0 0 1 0 0 0 0
1 2 3 2 8
Starting position alignment

Zero padding

(c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8

(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

(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

(f) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Final position

Full correlation result

(g) 0 0 0 8 2 3 2 1 0 0 0 0

Cropped correlation result

(h) 0 8 2 3 2 1 0 0

Convolution

Origin f w rotated 180°

0 0 0 1 0 0 0 0   8 2 3 2 1 (i)

0 0 0 1 0 0 0 0 (j)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n)
8 2 3 2 1

Full convolution result

0 0 0 1 2 3 2 8 0 0 0 0 (o)

Cropped convolution result

0 1 2 3 2 8 0 0 (p)

Correlation & Convolution

- Correlation is a function of displacement of the filter.
- Correlating a filter w with a function that contains all ‘0’ & single ‘1’ yields a 180˚ rotated copy of w.
- Correlating a function with discrete unit impulse yields a rotated (time inverted) version of the function.
- Convolving a function with a unit impulse yields the same function.
- Thus, to perform convolution all we have to do is rotate one function by 180˚ & perform same operation as in correlation.

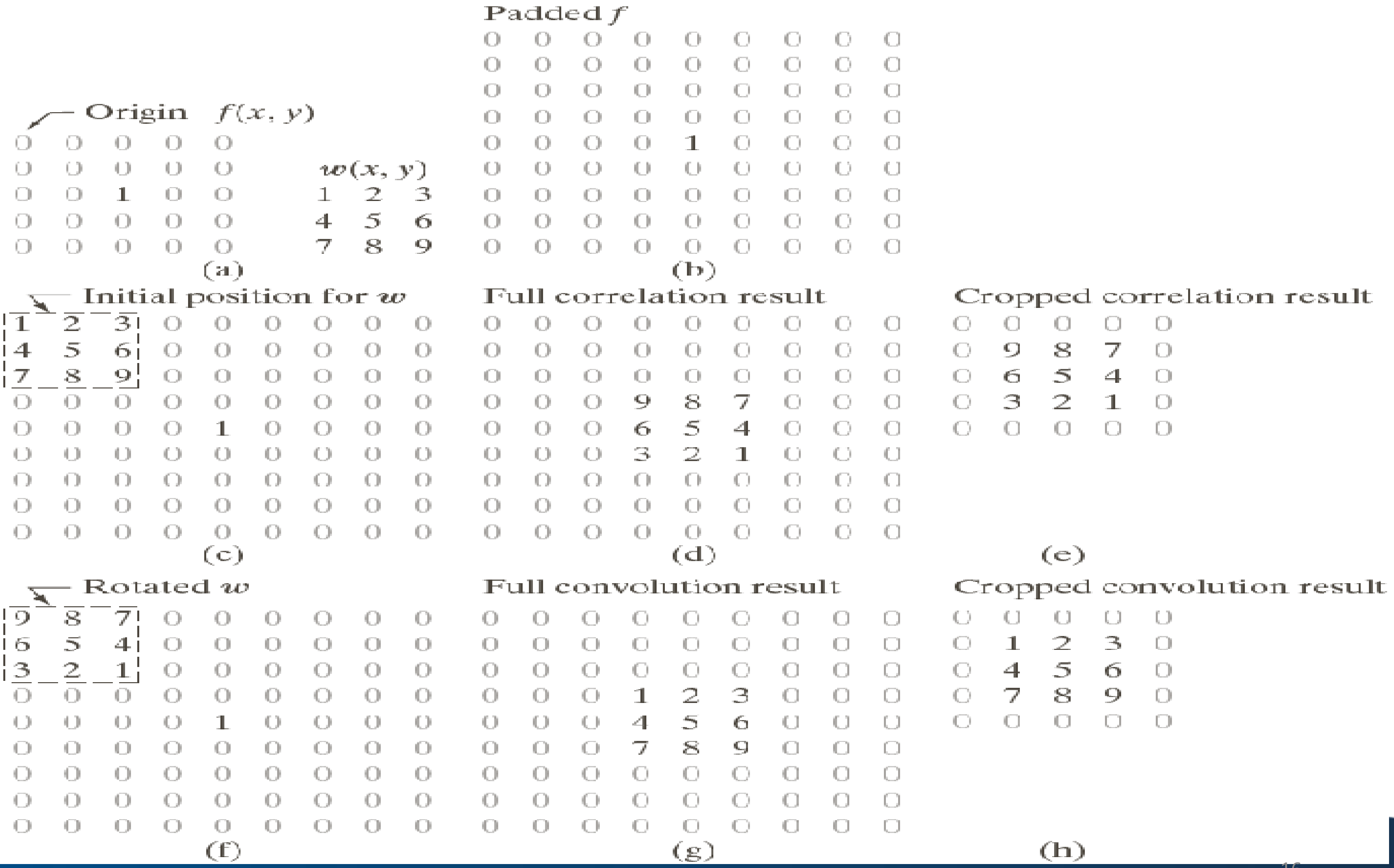


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Some fundamental properties of convolution and correlation

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

❖ Linear spatial filtering

Pixels of image

f(x-1.y-1)	f(x-1,y)	f(x,y-1)
f(x,y-1)	f(x,y)	f(x,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

•The result is the sum of products of the mask coefficients with the corresponding pixels directly under the mask

$$
\begin{aligned}
g(x,y) = {} & w(-1,-1)\,f(x-1,y-1) + w(-1,0)\,f(x-1,y) + w(-1,1)\,f(x-1,y+1) + \\
& w(0,-1)\,f(x,y-1) + w(0,0)\,f(x,y) + w(0,1)\,f(x,y+1) + \\
& w(1,-1)\,f(x+1,y-1) + w(1,0)\,f(x+1,y) + w(1,1)\,f(x+1,y+1)
\end{aligned}
$$

❖ Nonlinear spatial filtering

- Nonlinear spatial filters also operate on neighborhoods, and the mechanics of sliding a mask past an image are the same as was just outlined.
- The filtering operation is based conditionally on the values of the pixels in the neighborhood under consideration

- Ex: Noise reduction can be achieved effectively with a nonlinear filter with a basic function of computing the median gray level value in the neighbourhood

- Computation of median is a non-linear operation, as is that of variance

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

- **Motivation: Limitation of Linear Filters**
  - Frequency shaping

    enhance some frequency components and suppress the others
  - For individual frequency component, cannot differentiate its "desirable" and "undesirable" parts

- Nonlinear Filters
  - Cannot be expressed as convolution
  - Cannot be expressed as frequency shaping

- **"Nonlinear" Means Everything** (other than linear)
  - Need to be more specific
  - Often heuristic
  - We will study some "nice" ones

IMAGE- PROCESSING CSE395

Module-2

2.5–Smoothing Spatial Filters,



PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

Private University Estd. in Karnataka State by Act No. 41 of 2013

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

Smoothing Spatial Filter

Smoothing filters are used for

- blurring
- noise reduction.

Blurring is used in preprocessing steps to removal of small details from an image prior to object extraction and bridging of small gaps in lines or curves

Noise reduction can be accomplished by blurring

Types of Smoothing Filter

- There are 2 way of smoothing spatial filters
  - LinearFilters –operations performed on image pixel

Order-Statistics (non-linear) Filters - based on ranking the pixels

Linear Filter

Linear spatial filter is simply the average of the

pixels contained in the neighborhood of the filter mask.

The idea is replacing the value of every pixel in an image by the average of the gray

levels in the neighborhood defined by the filter mask.

Linear Filter (cont..)

This process result in an image reduce the sharp transitions in intensities. Two mask

- Averaging filter
- Weighted averaging filter

Averaging Filter

A major use of averaging filters is in the reduction of irrelevant detail in image.

mxn mask would have a normalizing constant equal to 1/mn.

Its also known as low pass filter.

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

Averaging Filter - Example

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

Smoothing Filters: Averaging

$$\frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(b)

$$\frac{1}{49} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(c)

Weighted Average Filter - Example

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

Weighted Averaging Filter

- The general implementation for filtering an MxN image with a weighted averaging filter of size m x n is given by the expression

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

- For complete filtered image apply **x = 0,1,2,3……..m-1** and **y=0,1,2,3,……..n-1** in the above equation.

PRESIDENCY UNIVERSITY

Ex. 1) 8x8 Pseudo image with a single edge (High Frequency) of 10 & 50. Remove using a 3x3 size averaging mask.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

0	0	0						
0	10	10	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

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

0	0	0						
0	4.44	10	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 10 \\ 0 & 10 & 10 \end{bmatrix}$$

0	0	0	0					
0	4.44	6.66	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix} 0 & 0 & 0 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

0	0	0	0	0				
0	4.44	6.66	6.66	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix} 0 & 0 & 0 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

0	0	0	0	0	0			
0	4.44	6.66	6.66	6.66	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix} 0 & 0 & 0 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44
0	6.66	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	10	10	10	10	10	10	10	0
0	50	50	50	50	50	50	50	50	0
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{bmatrix} 0 & 10 & 10 \\ 0 & 10 & 10 \\ 0 & 50 & 50 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	50	50	50	50	50	50	0
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{bmatrix} 10 & 10 & 10 \\ 50 & 50 & 50 \\ 50 & 50 & 50 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	36.66	36.66	36.66	36.66	36.66	24.44	0
0	33.33	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

$$\frac{1}{9}\begin{bmatrix} 50 & 50 & 50 \\ 50 & 50 & 50 \\ 50 & 50 & 50 \end{bmatrix}$$

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	36.66	36.66	36.66	36.66	36.66	24.44	0
0	33.33	50	50	50	50	50	50	33.33	0
0	33.33	50	50	50	50	50	50	33.33	0
0	22.22	33.33	33.33	33.33	33.33	33.33	33.33	22.22	0
	0	0	0	0	0	0	0	0	0

$$\frac{1}{9}\begin{bmatrix} 50 & 50 & 0 \\ 50 & 50 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	23.33	23.33	23.33	23.33	23.33	23.33	10
50	36.66	36.66	36.66	36.66	36.66	36.66	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

PRES
UNIV
Private University Estd. in Karnatak

- In the resultant image the Low frequency region has remained unchanged.
- Sharp transition between 10 & 50 has changed from 10 23.33 to 36.66 and finally to 50.
- Thus, Sharp edges has become blurred.
- Best result when used over image corrupted by Gaussia noise.
- Other types of low pass averaging mask are:

$$\frac{1}{6}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Smoothing filters – Example

input image

smoothed image

Order-Statistics Filter

- Order-statistics filters are nonlinear spatial filters.
- It is based on ordering (ranking) the pixels contained in the image area encompassed by the filter,
- It replacing the value of the center pixel with the value determined by the ranking result.

Order- statics filter

- The filter selects a sample from the window, does not average
- Edges are better preserved than with liner filters
- Best suited for “salt and pepper” noise

Types of order-statics filter

Different types of order-statics filters are

- Minimum filter
- Maximum filter
- Median filter

Minimum Filter

- The 0th percentile filter is the min filter.
- Minimum filter selects the smallest value in the window and replace the center by the smallest value
- Using comparison the minimum value can be obtained fast.(not necessary to sort)
- It enhances the dark areas of image

Minimum Filter : Example

(mask size =3 x 3)

(mask size =7 x 7)

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013

Maximum Filter

- The maximum filter selects the largest value within of pixel values, and replace the center by the largest value.

- Using comparison the maximum value can be obtained fast.(not necessary to sort)
- Using the 100th percentile results in the so-called *max filter*
- it enhances bright areas of image

Maximum Filter

mask (3 x 3)

mask (7 x 7)

Median Filter

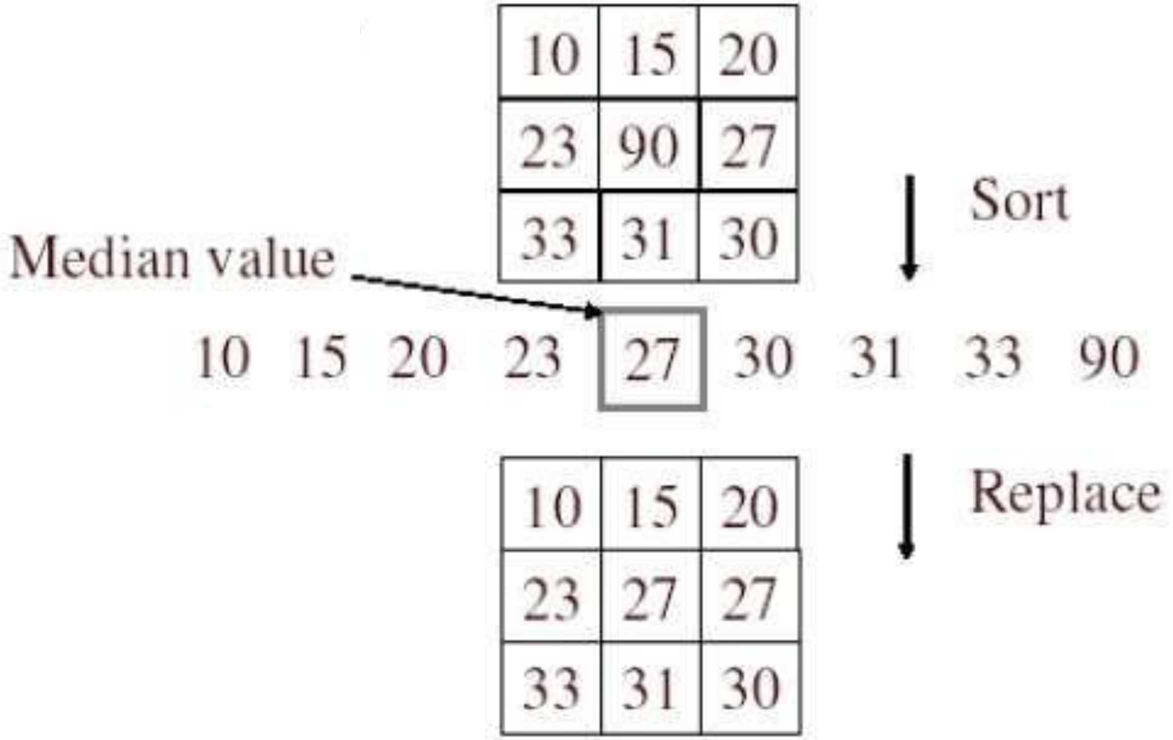
- Three steps to be followed to run a median filter:

1. Consider each pixel in the image

2. Sort the neighboring pixels into order based upon their intensities

3. Replace the original value of the pixel with the median value from the list.

Median Filter - Process



Ex. 2) 8x8 Pseudo image with a single edge (High Frequency) of 10 & 50. Remove using a 3x3 size median filter mask.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

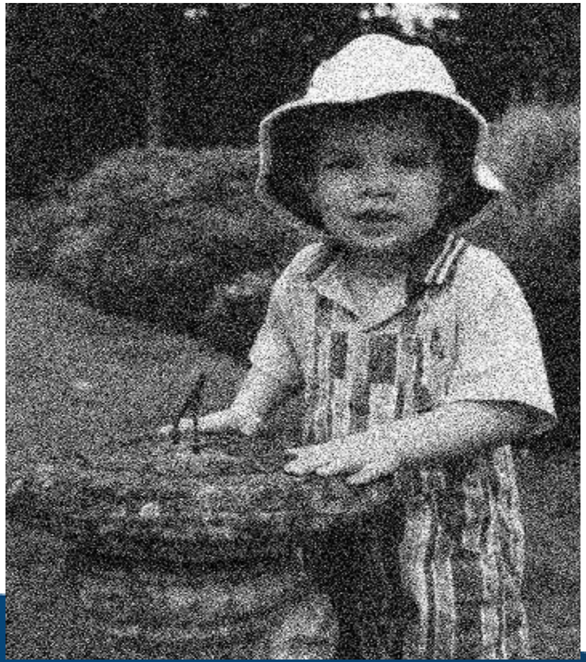
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

Median Filter - Example



Median Filter size =3 x 3

Median Filter size =7 x 7

Conclusion

- A linear filter cannot totally eliminate impulse noise, as a single pixel which acts as an intensity spike can contribute significantly to the weighted average of the filter.

- Non-linear filters can be robust to this type of noise because single outlier pixel intensities can be eliminated entirely.

Original image

Median filter

Mean filter

IMAGE- PROCESSING CSE395

Module-2

2.6 & 2.7– Sharpening Spatial Filters, Combining Spatial Enhancement Methods,



PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

Sharpening Filters(High pass filter)

1. The concept of sharpening filter
2. First and second order derivatives
3. Laplace filter
4. Unsharp mask
5. High boost filter
6. Gradient mask

Sharpening Spatial Filters

- To highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.

- Blurring vs. Sharpening

  - Blurring/smooth is done in spatial domain by pixel averaging in a neighbors, it is a process of integration

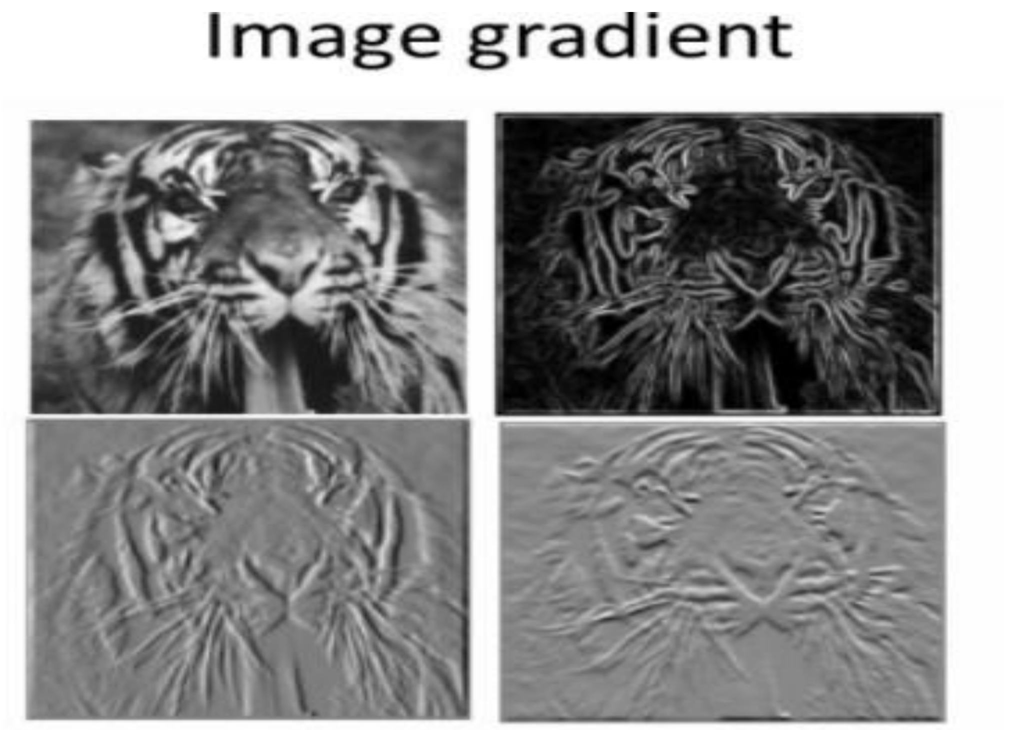
  - Sharpening is an inverse process, to find the difference by the neighborhood, done by spatial differentiation.

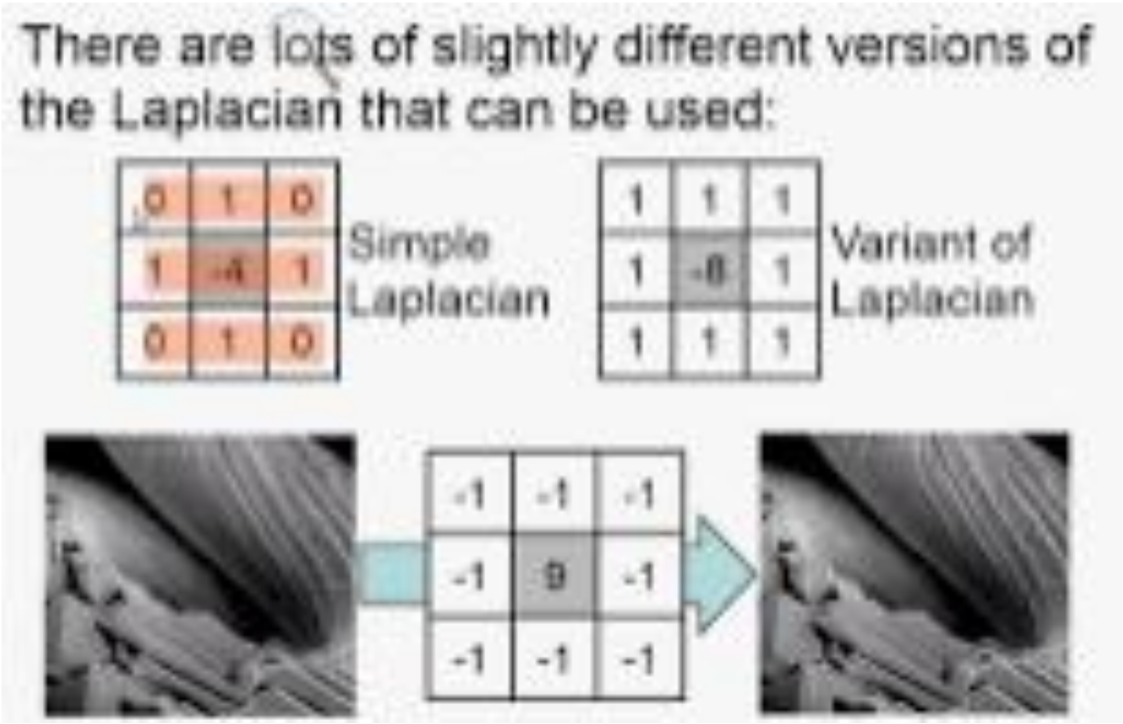
Derivative operator

- The strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.

- Image differentiation
  - enhances edges and other discontinuities (noise)
  - deemphasizes area with slowly varying gray-level values.

Gradiant and Laplacian
Discuss their role in image enhancement.





First and second order difference of 1D

- The basic definition of the first-order derivative of a one-dimensional function f(x) is the difference

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

- The second-order derivative of a one-dimensional function f(x) is the difference

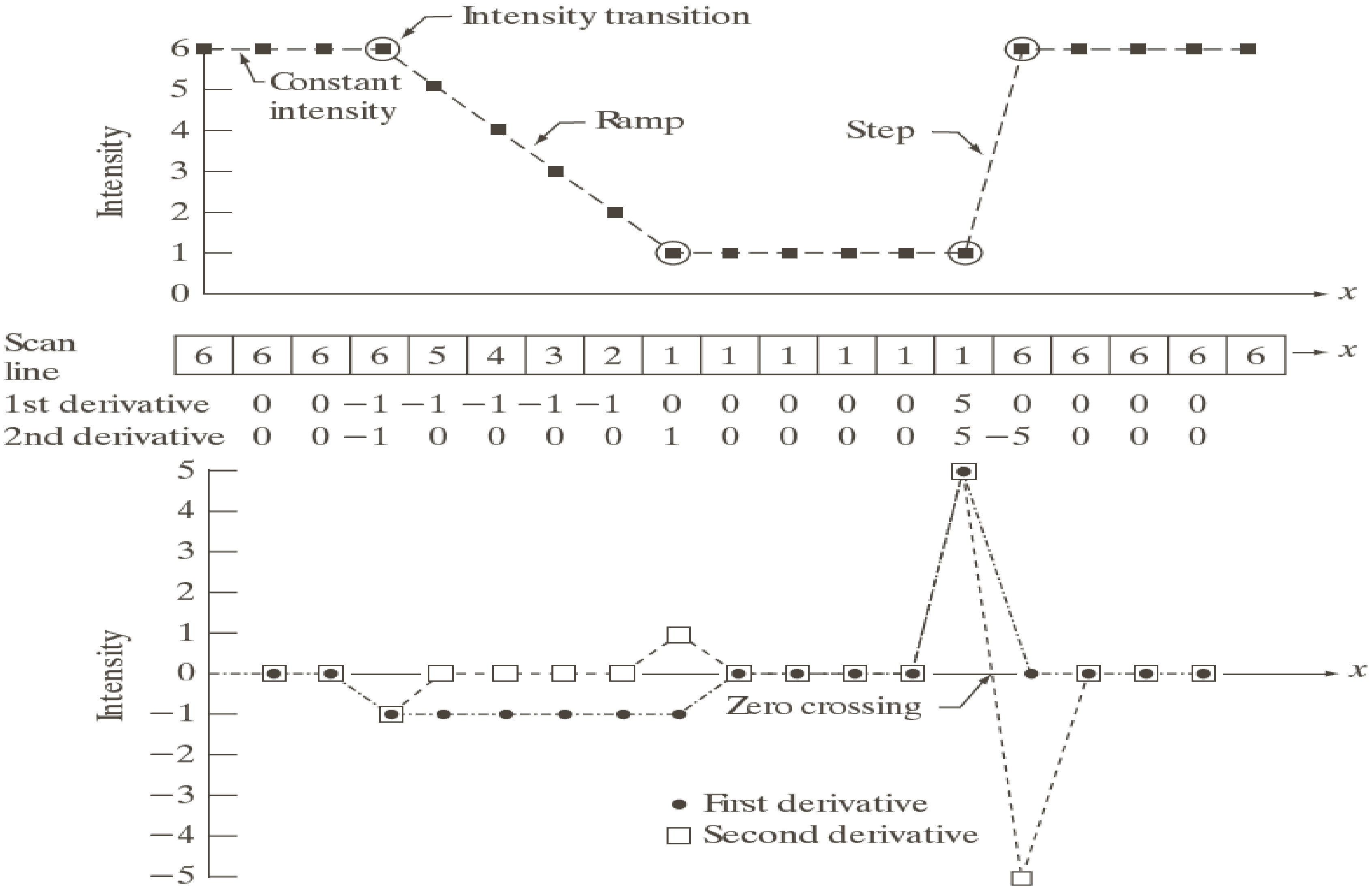
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

Private University Estd. in Karnataka State by Act No. 41 of 2013

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM



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.

First and Second-order derivative of 2D

- when we consider an image function of two variables, f(x, y), at which time we will dealing with partial derivatives along the two spatial axes.

Gradient operator

(linear operator)

$$\nabla \mathrm{f} = \frac{\partial f(x,y)}{\partial x \partial y} = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}$$

Laplacian operator (non-linear)

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

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

Discrete form of Laplacian

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

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

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

Result Laplacian mask

0	1	0
1	−4	1
0	1	0

Laplacian mask implemented an extension of diagonal neighbors

1	1	1
1	−8	1
1	1	1

Other implementation of Laplacian masks

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

give the same result, but we have to keep in mind that when combining (add / subtract) a Laplacian-filtered image with another image.

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
40 YEARS OF ACADEMIC WISDOM

Effect of Laplacian Operator

- as it is a derivative operator,
  - it highlights gray-level discontinuities in an image
  - it deemphasizes regions with slowly varying gray levels
- tends to produce images that have
  - grayish edge lines and other discontinuities, all superimposed on a dark,
  - featureless background.

Correct the effect of featureless background

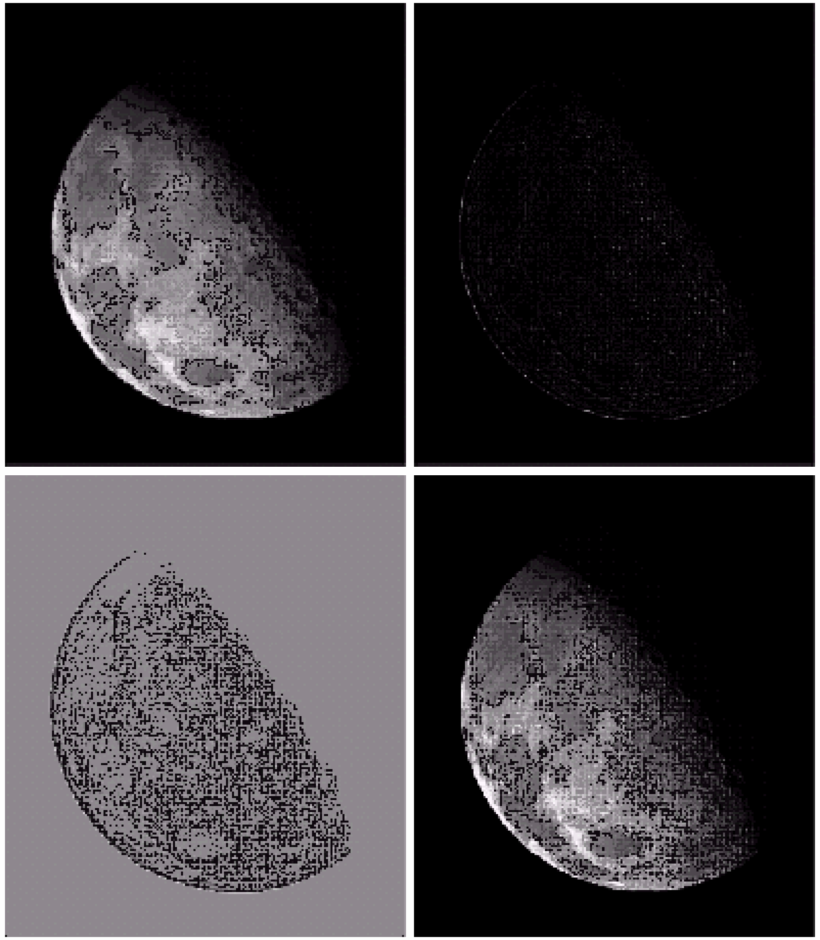
- easily by adding the original and Laplacian image.
- be careful with the Laplacian filter used

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

if the center coefficient of the Laplacian mask is negative

if the center coefficient of the Laplacian mask is positive

Example



- a). image of the North pole of the moon
- b). Laplacian-filtered image with

1	1	1
1	-8	1
1	1	1

- c). Laplacian image scaled for display purposes
- d). image enhanced by addition with original image

Mask of Laplacian + addition

- to simply the computation, we can create a mask which do both operations, Laplacian Filter and Addition the original image.

Mask of Laplacian + addition

$$g(x,y) = f(x,y) - [f(x+1,y) + f(x-1,y)$$
$$+ f(x,y+1) + f(x,y-1) + 4f(x,y)]$$
$$= 5f(x,y) - [f(x+1,y) + f(x-1,y)$$
$$+ f(x,y+1) + f(x,y-1)]$$

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

PRESIDENCY UNIVERSITY

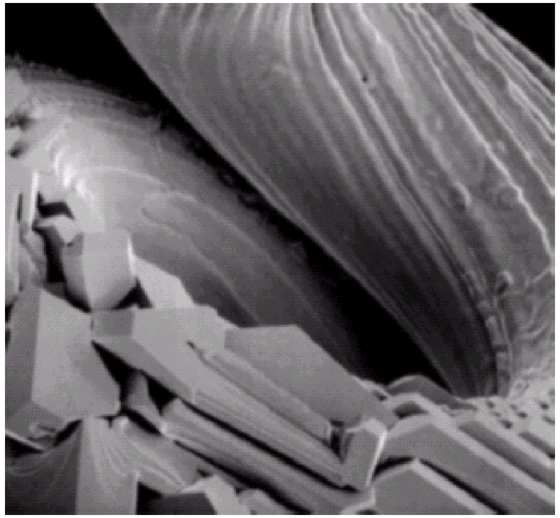
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

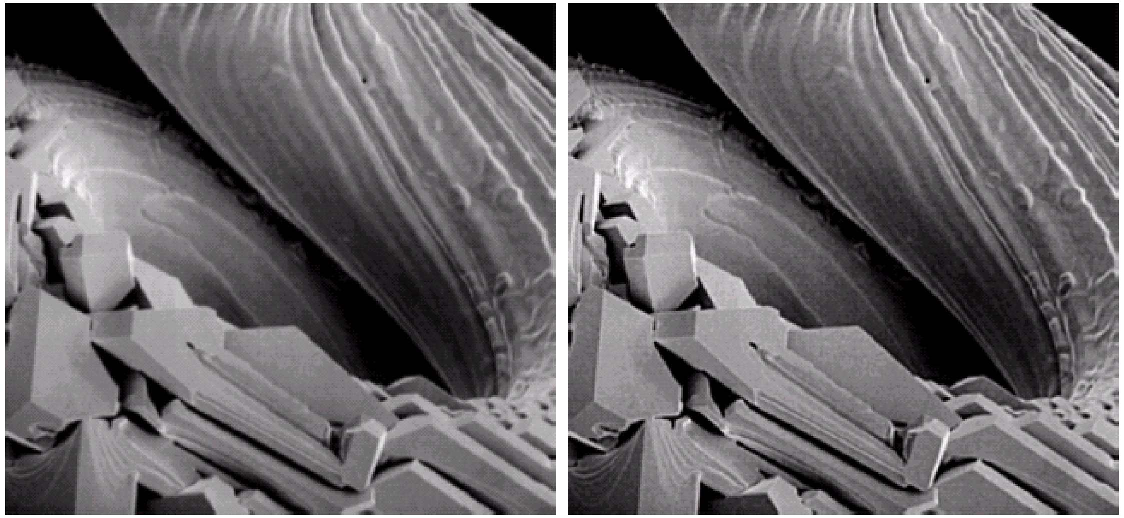
Private University Estd. in Karnataka State by Act No. 41 of 2013

Example

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1





a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Note

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

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 9 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 8 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

Unsharp masking

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

sharpened image = original image – blurred image

sharpened image = original image – blurred image

- to subtract a blurred version of an image produces sharpening output image.

Unsharp mask

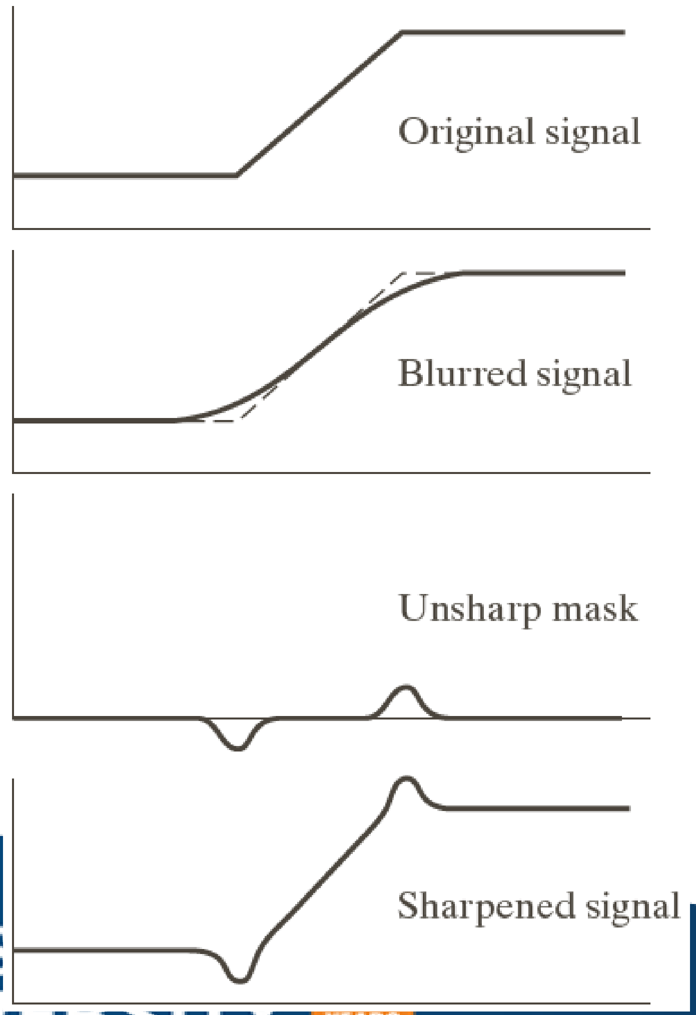
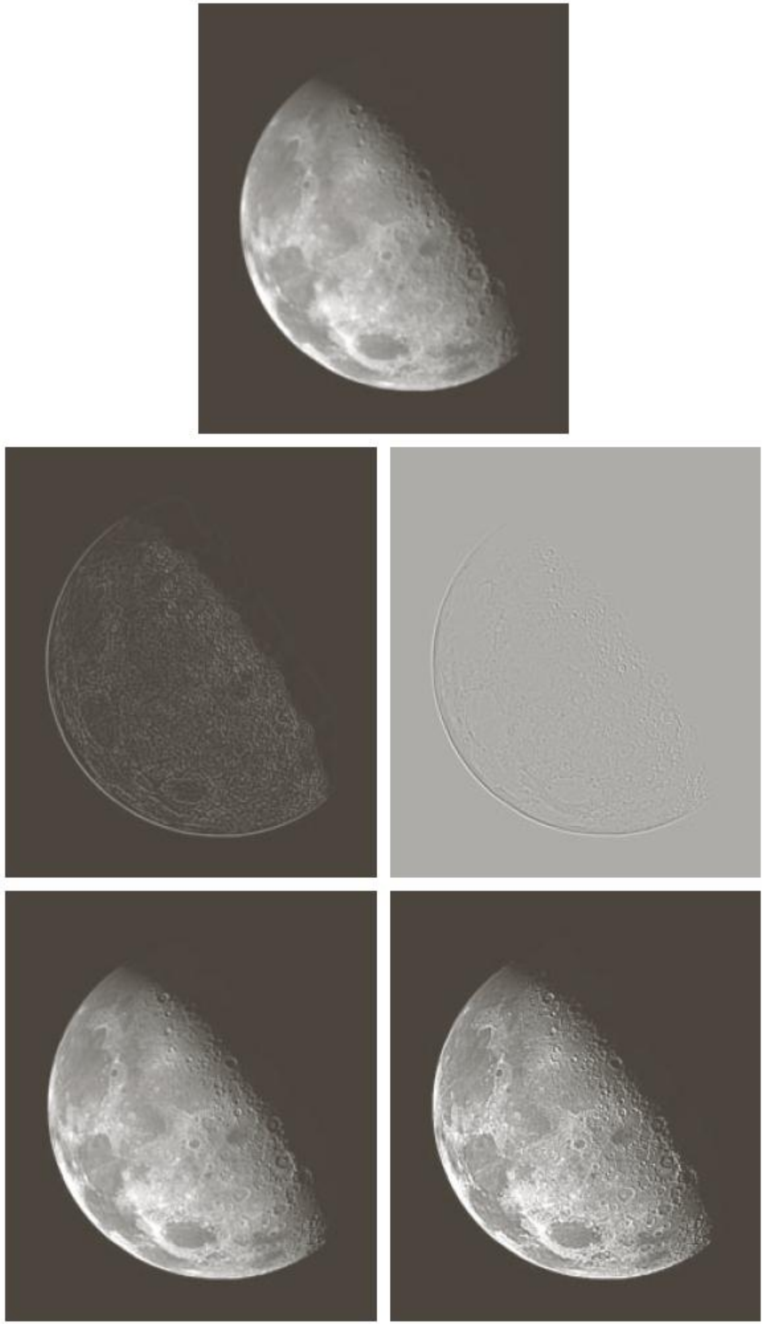


FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).



	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.)

High-boost filtering

$$f_{hb}(x, y) = Af(x, y) - \bar{f}(x, y)$$

$$f_{hb}(x, y) = (A-1)f(x, y) - f(x, y)\bar{f}(x, y)$$
$$= (A-1)f(x, y) - f_s(x, y)$$

- generalized form of Unsharp masking
- $A \geq 1$

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

High-boost filtering

$$f_{hb}(x,y) = (A-1)f(x,y) - f_s(x,y)$$

- if we use Laplacian filter to create sharpen image $f_s(x,y)$ with addition of original image

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

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

High-boost Masks

0	−1	0
−1	$A + 4$	−1
0	−1	0

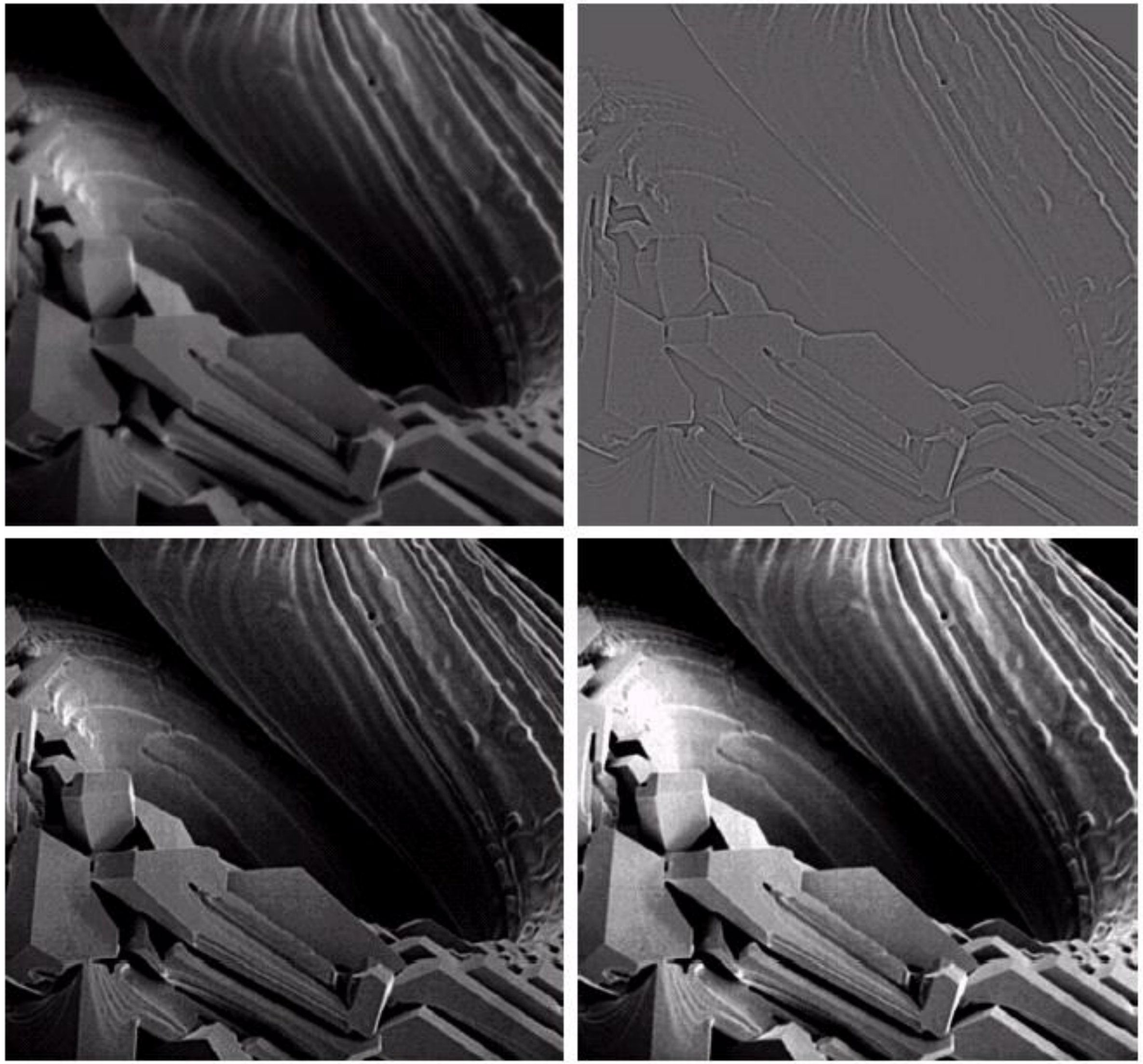
−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

- $A \geq 1$
- if A = 1, it becomes “standard” Laplacian sharpening

Example

a b
c d

FIGURE 3.43
(a) Same as Fig. 3.41(c), but darker. (a) 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 $A = 1.7$.



Gradient Operator

- first derivatives are implemented using the magnitude of the gradient.

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

$$\nabla f = mag(\nabla \mathrm{f}) = [G_x^2 + G_y^2]^{1/2}$$

$$= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

commonly approx.

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

the magnitude becomes nonlinear

Gradient Mask

- simplest approximation, 2x2

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$G_x = (z_8 - z_5) \quad \text{and} \quad G_y = (z_6 - z_5)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_8 - z_5)^2 + (z_6 - z_5)^2]^{1/2}$$

$$\nabla f \approx |z_8 - z_5| + |z_6 - z_5|$$

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Roberts cross-gradient operators, 2x2

$$G_x = (z_9 - z_5) \quad \text{and} \quad G_y = (z_8 - z_6)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

-1	0
0	1

0	-1
1	0

Gradient Mask

- Sobel operators, 3x3

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

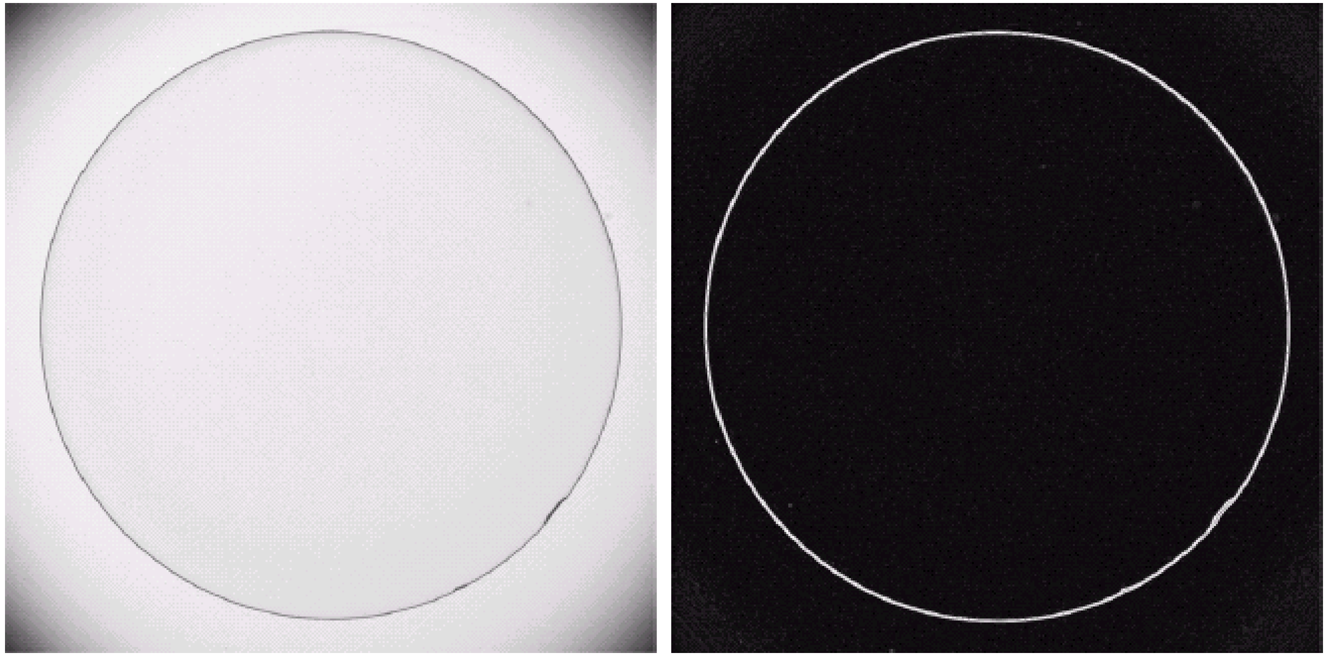
$$\nabla f \approx |G_x| + |G_y|$$

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

the weight value 2 is to achieve smoothing by giving more important to the center point

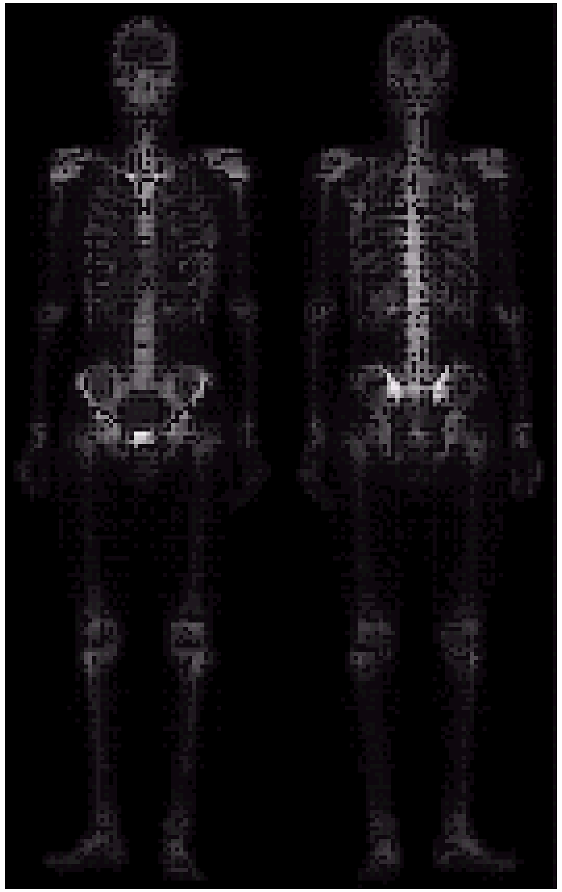
Example



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.)

Combining Spatial Enhancement Methods



- want to sharpen the original image and bring out more skeletal detail.
- problems: narrow dynamic range of gray level and high noise content makes the image difficult to enhance

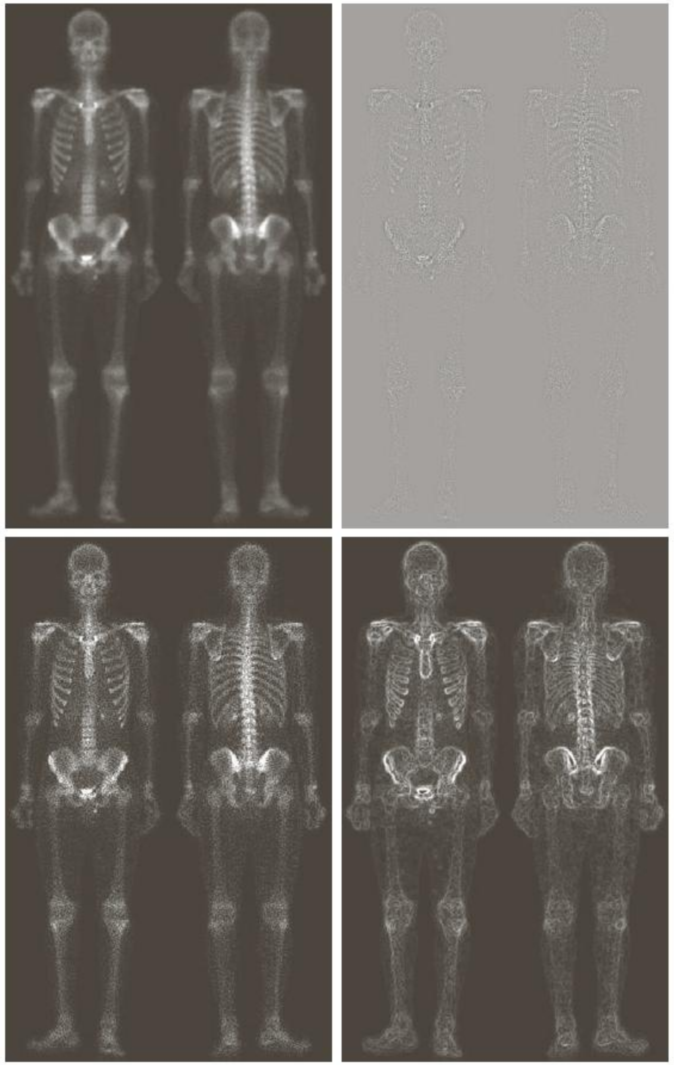
Example of Combining Spatial Enhancement Methods

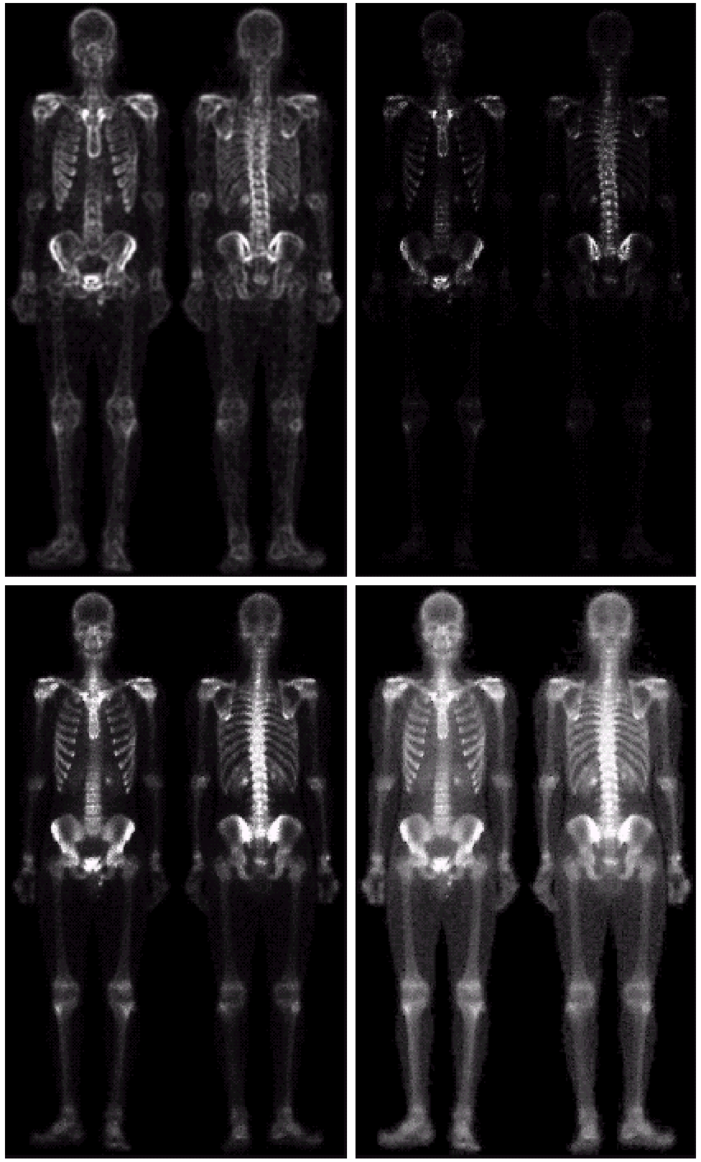
To solve :

1. Laplacian to highlight fine detail
2. gradient to enhance prominent edges
3. gray-level transformation to increase the dynamic range of gray levels

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).





e f
g h

FIGURE 3.46
(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.)

IMAGE- PROCESSING CSE395

Module-2

2.8– Frequency-Domain Filters

PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

Private University Estd. in Karnataka State by Act No. 41 of 2013

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

Frequency-Domain Filters

Basic model for filter in the frequency domain is given by the following equation Where F(u,v) is the Fourier transform of the image to be smoothed.

$$G(u,v) = H(u,v)F(u,v)$$

The objective is to select a filter transfer function H(u,v) that yields G(u,v) by attenuating the high-frequency components of F(u,v)

Frequency-Domain Filters

H(u,v) is called a filter is because it suppresses certain frequencies in the transform while leaving others unchanged.

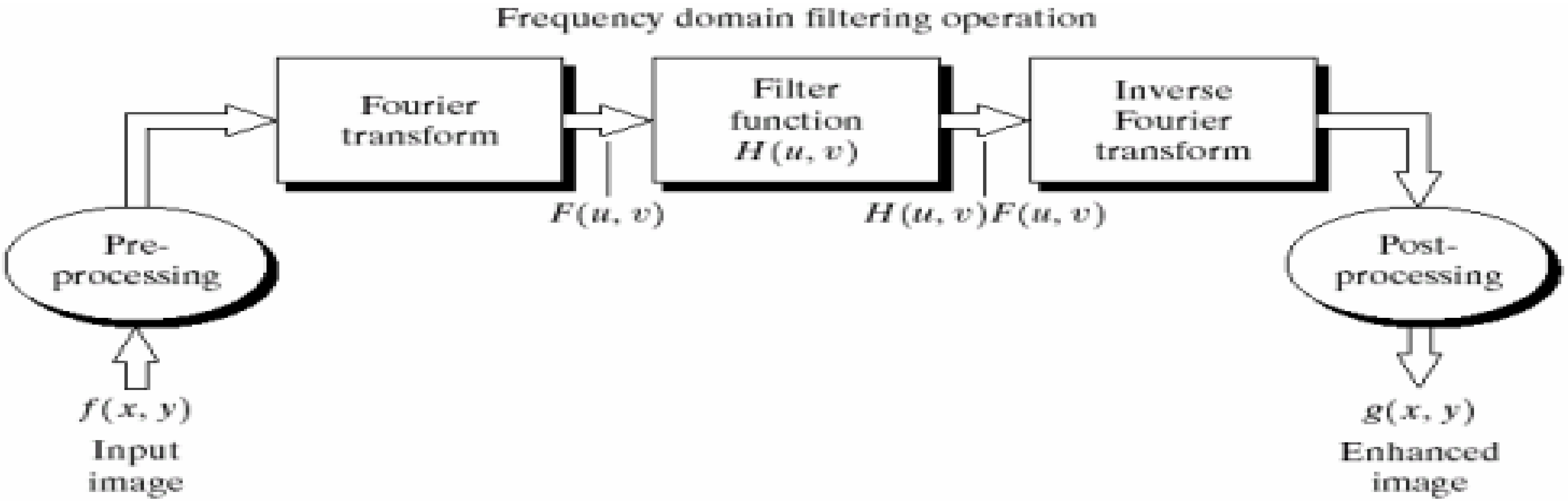
The Fourier transform of the output image is given by

$$G(u,v) = H(u,v)F(u,v)$$

The filtered image is obtained simply by taking the inverse Fourier transform of G(u,v):

$$\text{Filtered Image} = \Im^{-1}[G(u,v)]$$

Frequency-Domain Filters



The discrete convolution of two functions f(x,y) and h(x,y) of size M*N is denoted by f(x,y)*h(x,y) and is defined by the expression

$$f(x,y) * h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x-m, y-n)$$

Letting F(u,v) and H(u,v) denote the Fourier transforms of f(x,y) and h(x,y), the following result holds:

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v) H(u,v)$$

$$f(x,y) h(x,y) \Leftrightarrow F(u,v) * H(u,v)$$

Smoothing Frequency-Domain Filters

Basic model for filter in the frequency domai is given by the following equation

$$G(u,v) = H(u,v)F(u,v)$$

Where F(u,v) is the Fourier transform of the image to be smoothed.

The objective is to select a filter transfer function H(u,v) that yields G(u,v) by attenuating the high-frequency components of F(u,v).

Ideal low pass filter

The transfer function of a two-dimensional(2-D) ideal lowpass filter(ILPF) is:

$$H(u,v)=\begin{cases}1 & \text{if} \quad D(u,v)\le D_0\\ 0 & \text{if} \quad D(u,v)>D_0\end{cases}$$

The distance from any point (u,v) to the center of the Fourier transform is given by

$$\mathbf{D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}.}$$

a b
c d
e f

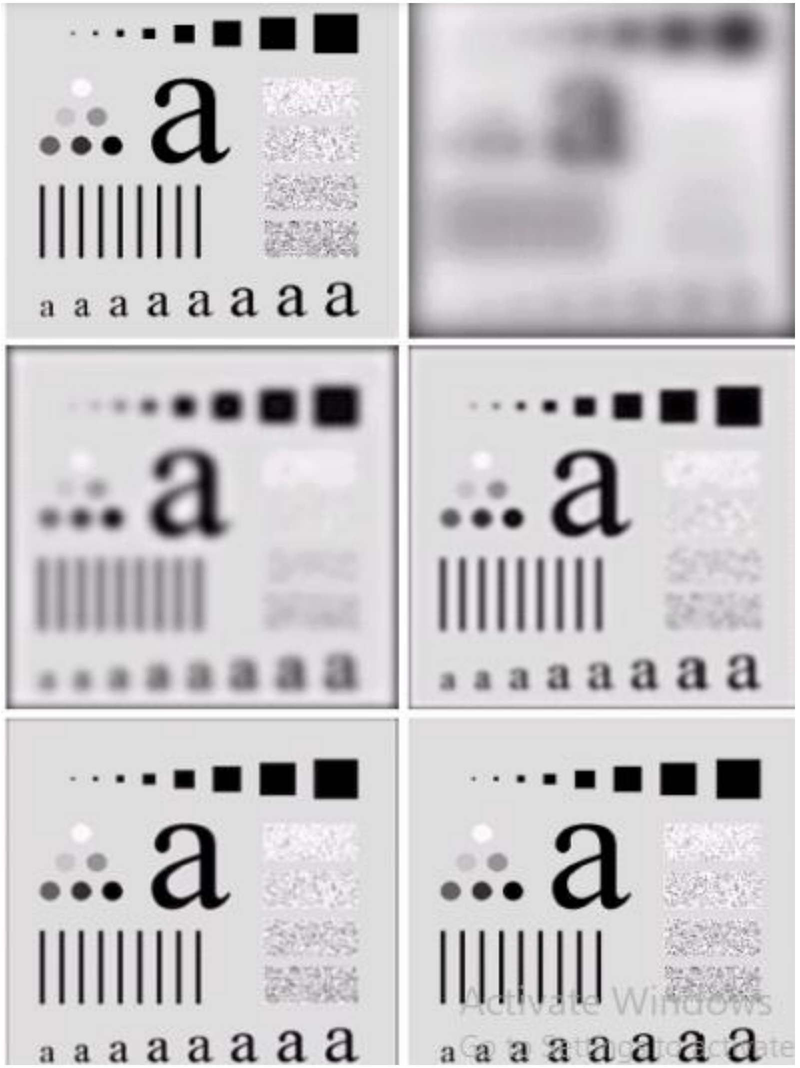
FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

PRESIDENCY UNIVERSITY
GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS
Private University Estd. in Karnataka State by Act No. 41 of 2013
PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

Butterworth Low pass Filters

Ideal filtering simply cuts off the Fourier transform. It is easy to implement, however, it has the disadvantage of introducing unwanted artifacts (ringing) into the result.

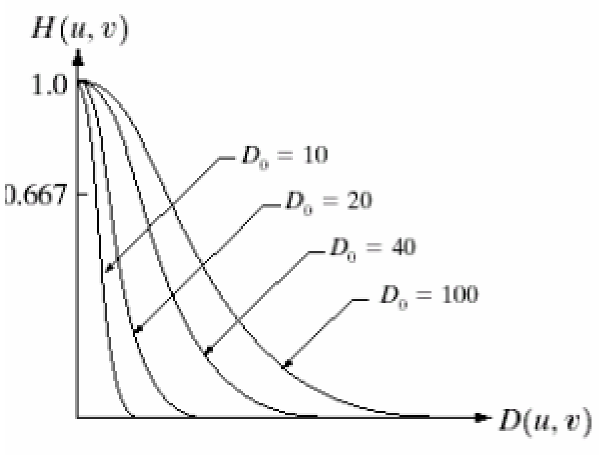
One way of avoiding these artifacts is to use a filter matrix a circle with a cutoff that is less sharp.



a b
c d
e f

FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

Gaussian Lowpass Filter



Sharpening Frequency Domain Filters

Image sharpening can be achieved by a high pass filtering process, which attenuates the low-frequency components without disturbing high-frequency information.

Zero-phase-shift filters: radially symmetric and completely specified by a cross section

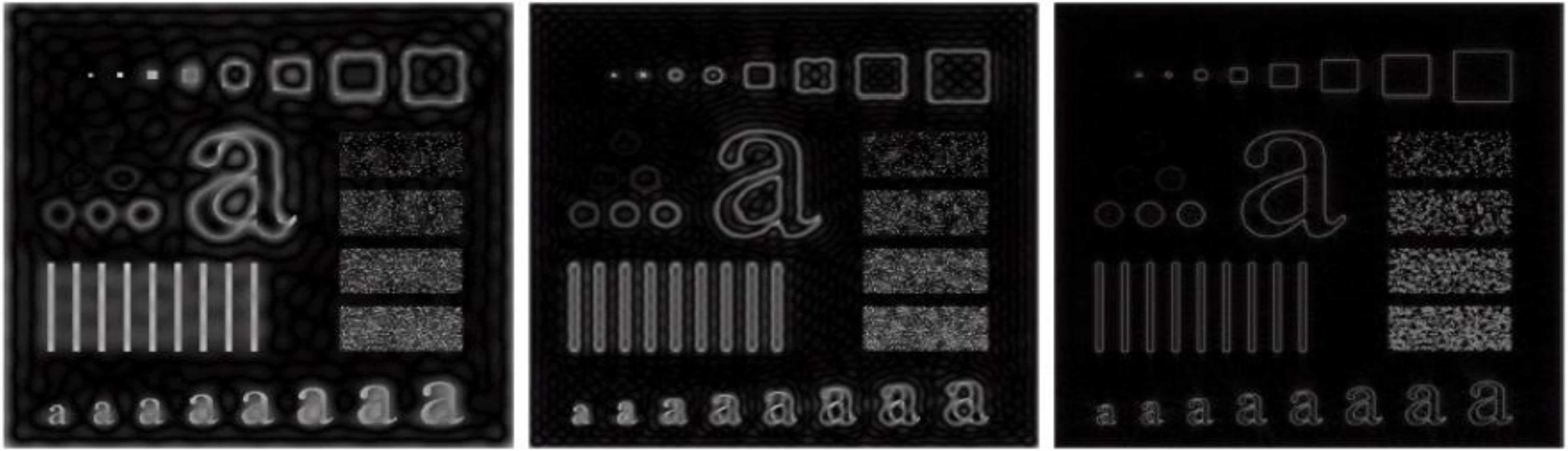
$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

Ideal High pass Filters

A 2-D ideal high pass filter (IHPF) is defined as

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

D0 is the cutoff distance measured.
This filter is the opposite of the ideal lowpass filter



a b c

FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

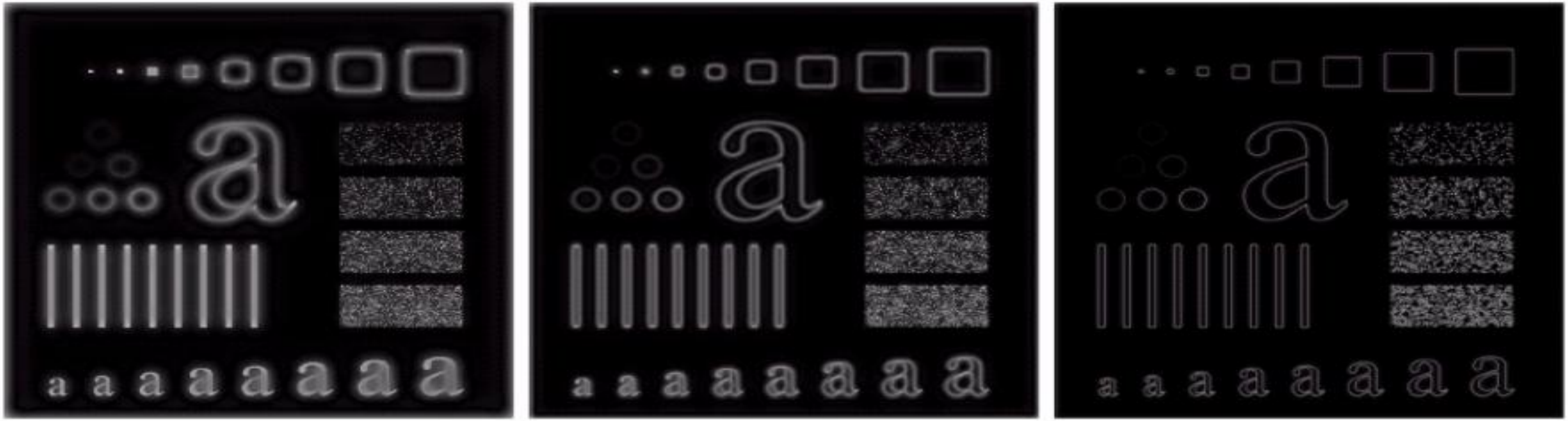
Butterworth High pass Filters

The transfer function of the Butterworth high pass filter (BHPF) of order n and will cutoff frequency locus at distance D0 from the origin is given by

$$H(u,v) = \frac{1}{1+[\frac{D_0}{D(u,v)}]^{2n}}$$

High-frequency emphasis: Adding a constant to a high pass filter to preserve the low-frequency components.

Since the center spot sizes of the IHPF and the BHPF are similar, the performance of the two filters I terms of filtering the smaller objects is comparable. The transition into higher values of cutoff frequencies is much smoother with the BHPF



a b c

FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

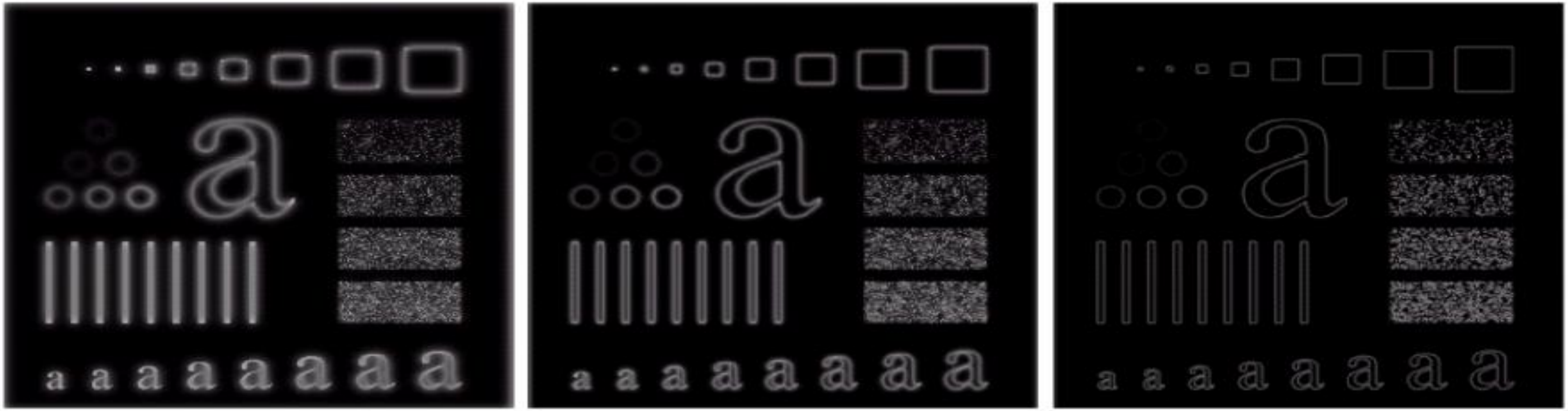
Gaussian High pass Filters

The transfer function of the Gaussian High pass Filters (GHPF) with cutoff frequency locus at distance D0 from the origin is given by

$$H(u,v) = 1 - \exp\left(\frac{-D^2(u,v)}{2D_0^{\ 2}}\right)$$

the results obtained are smoother than with the previous two filters. Even the filtering of the smaller objects and thin bars cleaner with the Gaussian filter.

Gaussian High pass Filters



a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

Spatial Vs Frequency

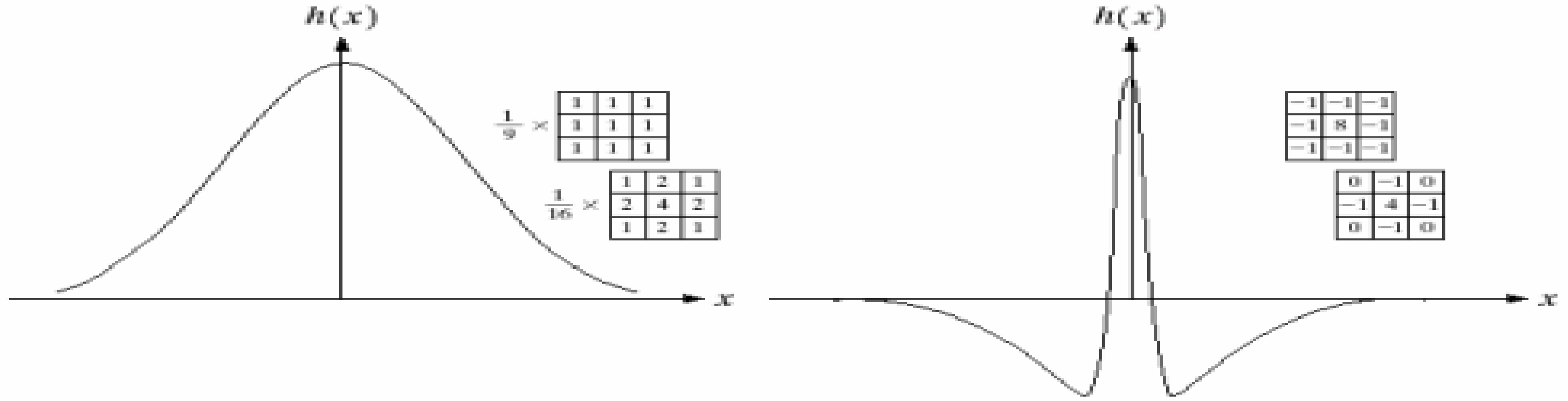


IMAGE- PROCESSING CSE395

Module-2

2.9– Explain the concept of homomorphic filtering.

PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE REACH GREATER HEIGHTS

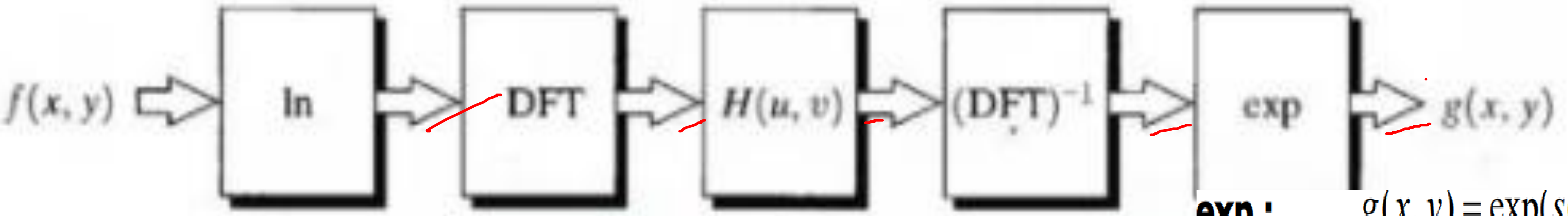
Private University Estd. in Karnataka State by Act No. 41 of 2013

PRESIDENCY GROUP OVER 40 YEARS OF ACADEMIC WISDOM

Explain the concept of homomorphic filtering.

DFT : $Z(u,v) = F_i(u,v) + F_r(u,v)$

$(DFT)^{-1}$: $s(x,y) = i'(x,y) + r'(x,y)$



exp : $g(x,y) = \exp(s(x,y)) = i_0(x,y)r_0$

H(u,v) : $S(u,v) = H(u,v)Z(u,v)$

Homomorphic filtering approach for image enhancement

Homomorphic $\Rightarrow$ $f(x,y) = L(x,y) \times R(x,y)$

Image illu Refl

Apply logfunction ✓

$\hookrightarrow Z(x,y) = \ln f(x,y) = \ln i(x,y) + \ln r(x,y)$

DFT $\hookrightarrow \mathcal{F}\{z(x,y)\} = \mathcal{F}\{\ln f(x,y)\} = \{\ln i(x,y)\} + \{\ln r(x,y)\}$

$Z(u,v) = F_i(u,v) + F_r(u,v)$

Filter with $H(u,v)$

$$\hookrightarrow S(U,V) = H(U,V)\, Z(U,V)$$

$$= H(U,V)\, F_i(U,V) + H(U,V)\, F_r(U,V)$$

DFT^{-1}

$$\hookrightarrow \mathcal{F}^{-1}\{S(U,V)\}\ \mathcal{F}^{-1}\{H(U,V)\, F_i(U,V)\} + \mathcal{F}^{-1}\{H(U,V)\, F_r(U,V)\}$$

$$= i'(x,y) + r'(x,y)$$

Apply Antilog function

$$\hookrightarrow g(x,y) = e^{s(x,y)} = e^{i'(x,y)} e^{r'(x,y)} = i_0(x,y)\, r_0(x,y)$$

$$g(x,y) = i_0(x,y)\; r_0(x,y)$$

The enhanced image is given by

$$g(x,y) = \exp[s(x,y)] = \exp[i'(x,y)] + \exp[r'(x,y)]$$

$$\boxed{g(x,y) = i_0(x,y)\,r_0(x,y).}$$

where $i_0(x,y) = \exp[i'(x,y)]$ and $r_0(x,y) = \exp[r'(x,y)]$.

The Homomorphic filtering steps are given as