

## IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN

### Unit structure :

- 3.0 Objectives
- 3.1 Introduction
- 3.2 Basic Grey Level Transform
- 3.3 Identity Transform Function
- 3.4 Image Negatives
- 3.5 Logarithmic Functions
- 3.6 Power-Law Functions
- 3.7 Piecewise-Linear Transformation Functions
- 3.8 Image Contrast Stretching
- 3.9 Image Thresholding
- 3.10 Gray-level Slicing
- 3.11 Bit-plane Slicing
- 3.12 Basics of Histogram Processing
- 3.13 Histogram equalisation (Automatic)
- 3.14 Histogram matching (specification)
- 3.15 Local histogram equalisation
- 3.16 Enhancement using Arithmetic/ Logic Operation
  - 3.16.1 Image Subtraction
  - 3.16.2 Image Averaging
- 3.17 Summary
- 3.18 Unit end exercise
- 3.19 Further Reading

---

### 3.0 OBJECTIVES

---

This chapter Objectives are:

- To understand image in spatial domain.

- Spatial domain representation and manipulation.
- Know how to manipulate histograms for image enhancement; including histogram stretching, equalization and specification.
- Understand the corresponding algorithms and equations.
- Understand gray-scale modification, how it relates to histogram manipulation and understand their mapping equations.
- Understand adaptive contrast enhancement

---

### 3.1 INTRODUCTION TO SPATIAL DOMAIN METHODS

---

- Spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image.
- Frequency domain processing techniques are based on modifying the Fourier transform of the image.
- Suppose we have a digital image which can be represented by a two dimensional random field  $f(x, y)$ .
- Spatial domain processes will be denoted by the expression  $g(x, y) [T = f(x, y)]$  or  $s = T(r)$
- The value of pixels, before and after processing, will be denoted by  $r$  and  $s$ , respectively.
- Where  $f(x, y)$  is the input to the image,  $g(x, y)$  is the processed image and  $T$  is an operator on  $f$ .
- The operator  $T$  applied on  $f(x, y)$  may be defined over:
  - (i) A single pixel  $(x, y)$ . In this case  $T$  is a grey level transformation (or mapping) function.
  - (ii) Some neighborhood of  $(x, y)$ .
  - (iii)  $T$  may operate to a set of input images instead of a single image.
- We are dealing now with image processing methods that are based only on the intensity of single pixels.

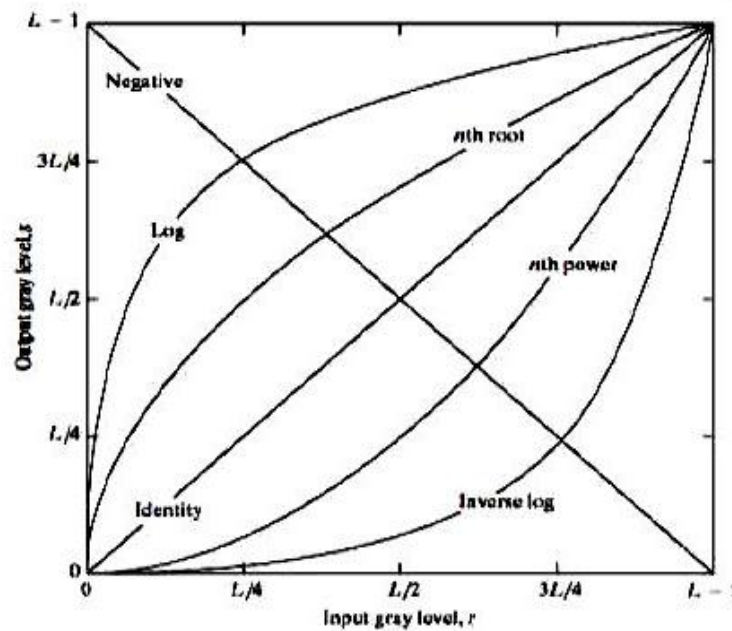
---

### 3.2 BASIC GREY LEVEL TRANSFORM

---

- 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.



**Figure 3.1 : Some of the basic grey level transformation functions used for image enhancement**

The three basic types of functions used frequently for image enhancement:

1. Linear Functions:

- Identity Transformation
- Negative Transformation
- Contrast Stretching
- Thresholding

2. Logarithmic Functions:

Log Transformation  
Inverse-log Transformation

3. Power-Law Functions:

- nth power transformation
- nth root transformation

### 3.3 IDENTITY TRANSFORM FUNCTION

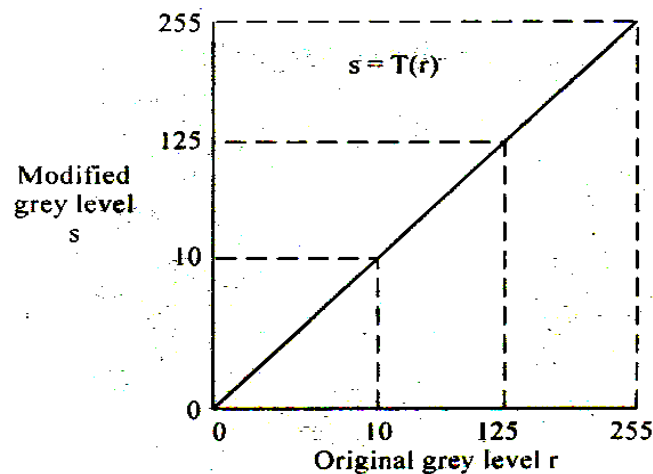


Figure 3.2 Identity Transform Graph

- Output intensities are identical to input intensities, hence called as identity transform function
- This function doesn't have an effect on an image, it was included in the graph only for completeness
- Its expression:

$$s=r$$

- Where  $r$  is original grey scale value of pixel, and  $r$  is transformed grey scale value of same pixel.
- In this case there is no change in the value of  $r$  and  $s$ .

### 3.4 IMAGE NEGATIVES

- Image negative means inverting the grey levels.
- The negative of a digital image is obtained by the transformation function:
 
$$s = T(r) = L - 1 - r$$
- $[0, L-1]$  the range of grey levels
- As shown in the following figure, where  $L$  is the number of grey levels.

- The idea is that the intensity of the output image decreases as the intensity of the input increases.
- This is useful in numerous applications such as displaying medical images.

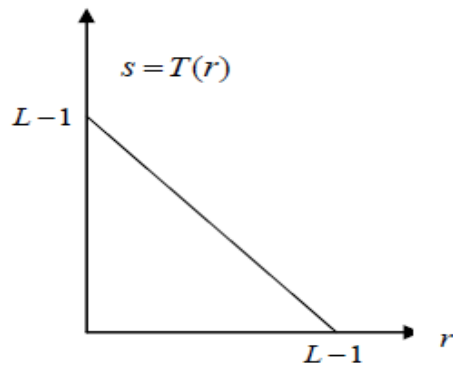


Fig 1



Negative of Fig 1

Figure 3.3 Image Negative

Solved Example to find image negative

Q ] Obtain the digital negative image of the 3 Bit image as shown below:

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0

2	6	6	5	1
0	2	3	2	1

**Solution:**

- For obtaining the negative image :

$$f_2(x,y) = L - 1 - f_1(x,y)$$

Where ,

$L$  : Total No. of Gray Levels

$f_1(x,y)$  : Original Image.

- For a 3 - Bit image :  $L = 2^3 = 8$

$$\begin{aligned} f_2(x,y) &= 8 - 1 - r \\ &= 7 - r \end{aligned}$$

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1
$f_1(x,y)$				
6	5	5	5	5
4	5	3	2	5
5	1	1	0	7
5	1	1	2	6
7	5	4	5	6
Resultant image $f_2(x,y)$				

### 3.5 LOGARITHMIC FUNCTIONS

- Logarithmic transformation further contains two type of transformation.
- 1) Log transformation
  - 2) Inverse log transformation.

#### Log Transformation

- Log transformation is also known as **Dynamic Range Compression**
- At times, the dynamic range of image exceeds the capability of the display device.
- In image some pixel values are so large that the other low value pixels get obscured.
- We need a technique to see even low value pixels.
- The technique to compress the dynamic range is known as *dynamic range compression*.
- The function used is *log transformation function*.
- The log transformations can be defined by this formula:
 
$$s = c \log(r + 1).$$
- Where  $s$  and  $r$  are the pixel values of the output and the input image and  $c$  is a constant.
- The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then  **$\log(0)$  is equal to infinity.**
- So 1 is added, to make the minimum value at least 1.
- During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values.
- The higher pixel values are kind of compressed in log transformation.
- This result in following image enhancement.
- The value of  $c$  in the log transform adjust the kind of enhancement you are looking for.



$$S = C \log(r + 1)$$

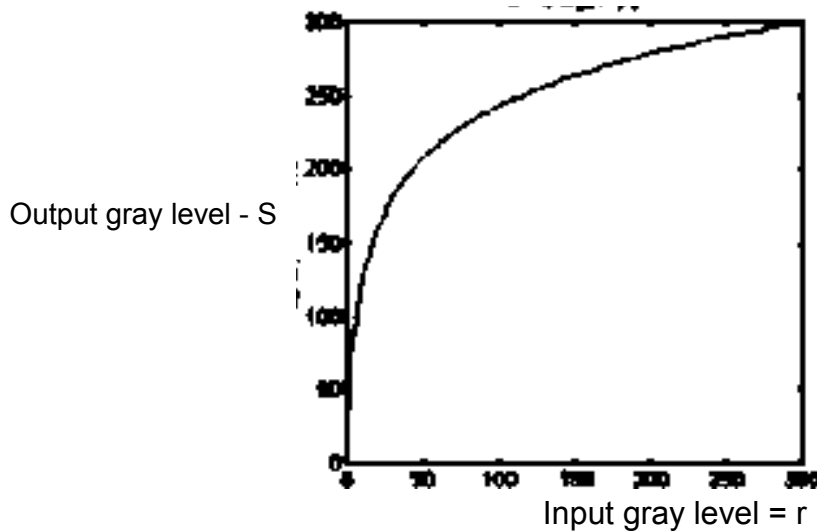
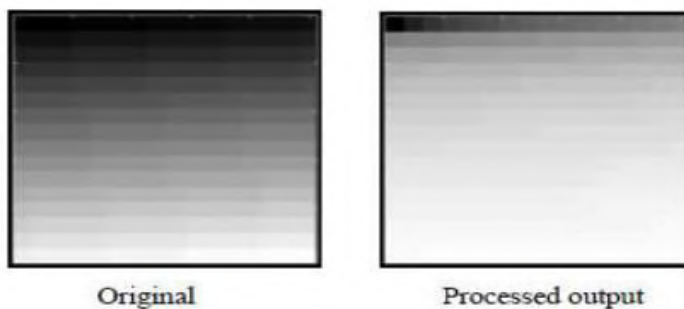


Figure 3.4 : Log Transformation

**Inverse log transformation**

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

**3.6 POWER-LAW FUNCTIONS**

- Power law transformation function further contains two type of transformation.
  - 1) nth power transformation
  - 2) nth root transformation

**nth power transformation**

- Power-law transformations have the basic form of:

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

- Where c and  $\gamma$  are positive constants



- Different transformation curves are obtained by varying  $\gamma$  (gamma)

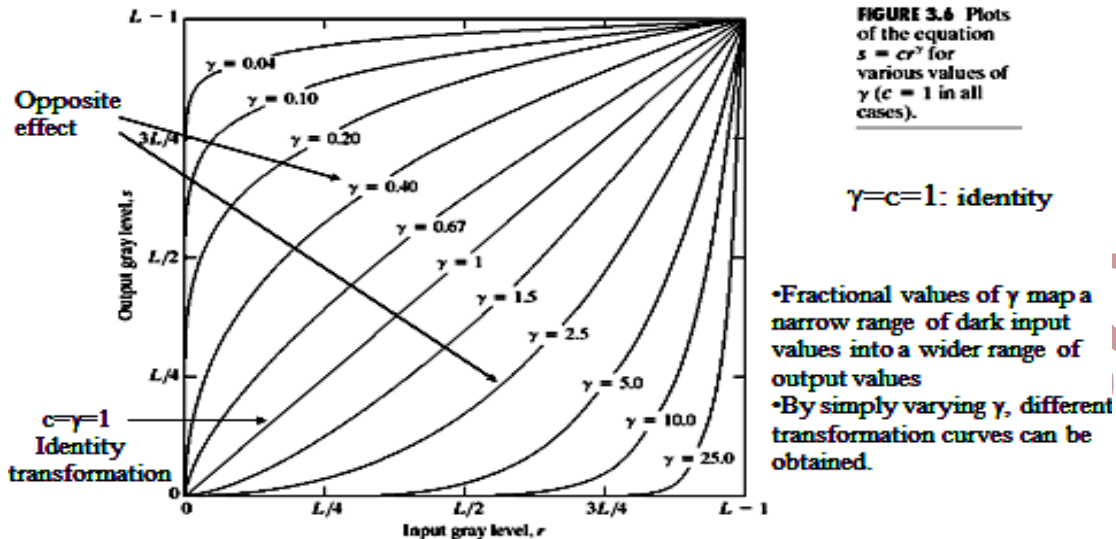
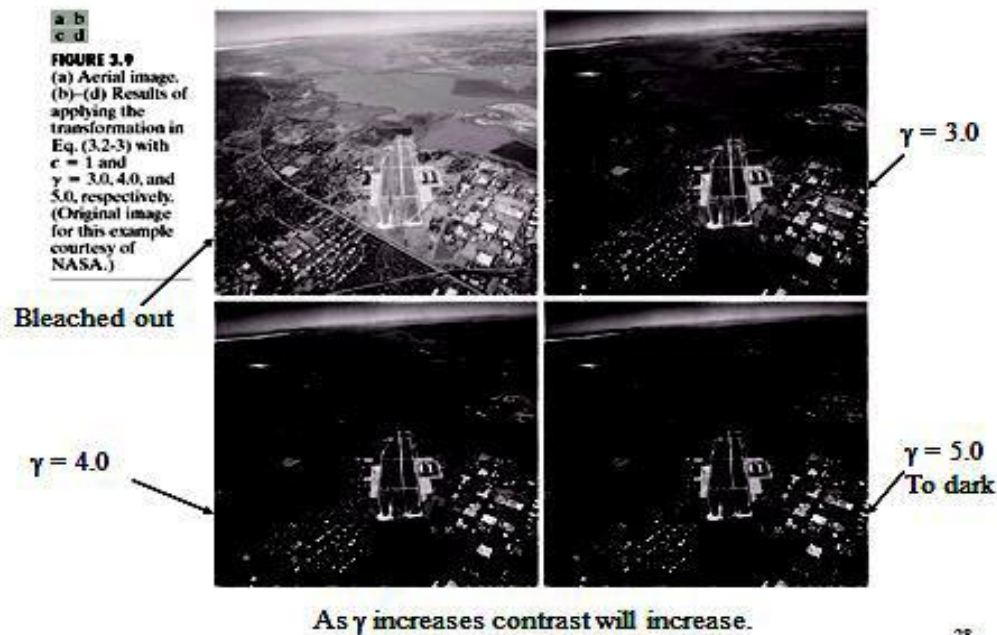


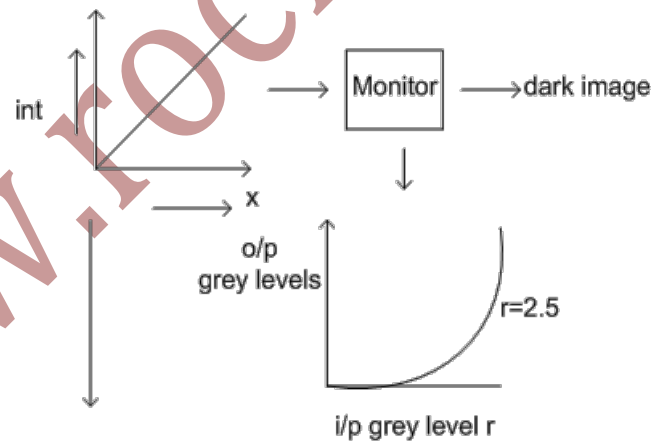
Figure 3.5 Plot of equation  $S = c \cdot r^\gamma$

- The exponent in the power-law equation is referred to as gamma.
- The process used to correct this power-law response phenomena is called a *gamma correction*.
- eg. CRT devices have an intensity-to-voltage response that is a power response.
- Gamma correction matters if you have any interest in displaying an image accurately on a computer screen.
- Gamma correction controls the overall brightness of an image.
- Images which are not properly corrected can look either bleached out (very small value of  $\gamma$ ), or too dark (very large value of  $\gamma$ ).
- Trying to reproduce colors accurately also requires some knowledge of gamma.
- Varying the amount of gamma correction changes not only the brightness, but also the ratios of red to green to blue.
- Almost every computer monitor, from whatever manufacturer, has one thing in common. They all have a intensity to voltage response curve which is roughly a 2.5 power function.
- This just means that if you send your computer monitor a message that a certain pixel should have intensity equal to  $x$ , it will actually display a pixel which has intensity equal to  $x$

$\gamma = 2.5$  Because the range of voltages sent to the monitor is between 0 and 1, this means that the intensity value displayed will be less than what you wanted it to be. ( $0.5^{\gamma = 2.5} = 0.177$  for example) Monitors, then, are said to have a gamma of 2.5.



28



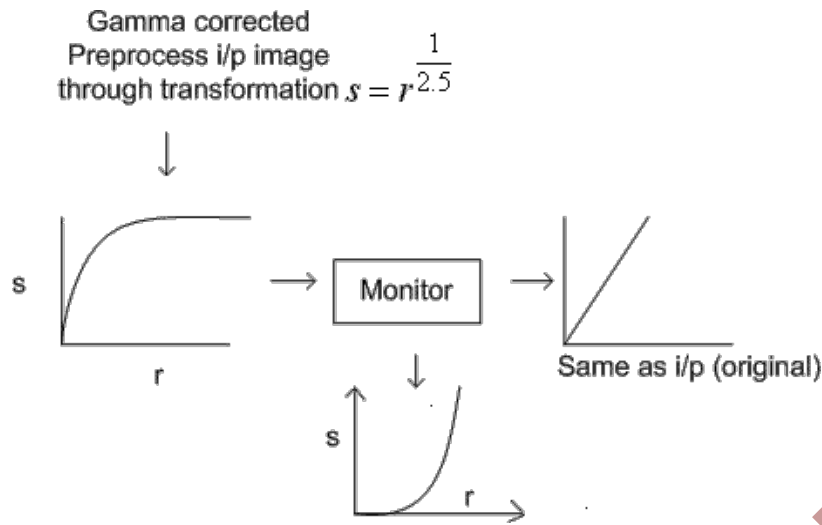


Figure 3.6 : Effects of gamma correction

### 3.7 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 than previous transformations
- Types of Piecewise transformations are:
  - Contrast Stretching
  - Gray-level Slicing
  - Bit-plane slicing

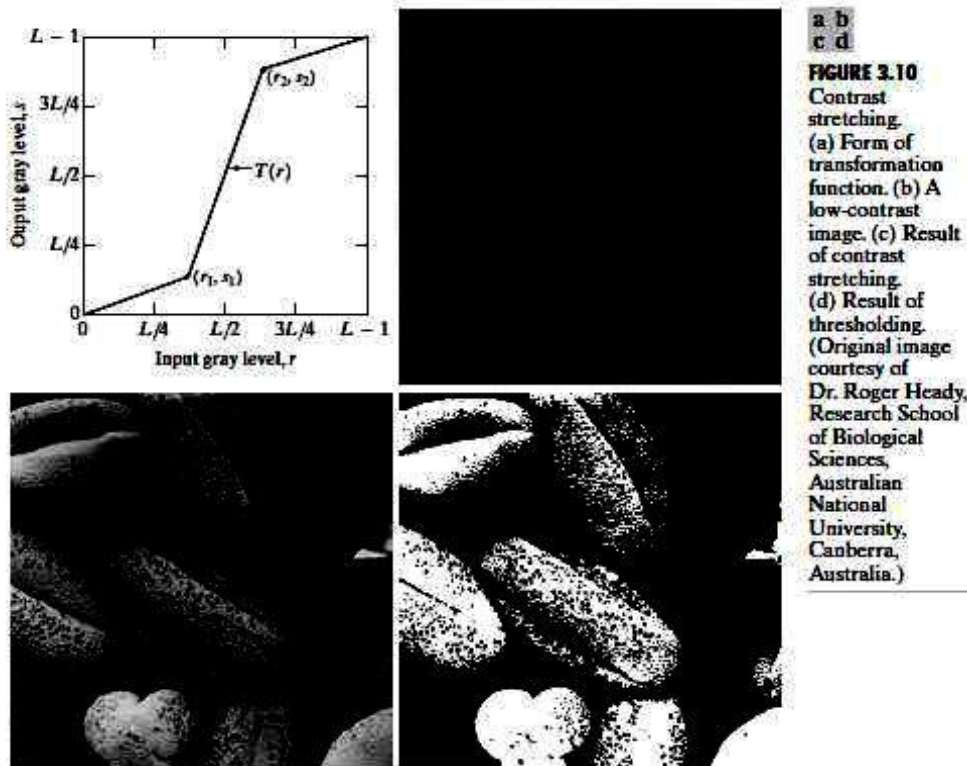
### 3.8 IMAGE CONTRAST STRETCHING

- Many times we obtain low contrast image due to poor illumination.
- Contrast enhancements improve the perceptibility of objects in the scene by enhancing the brightness difference between objects and their backgrounds.
- Contrast enhancements are typically performed as a contrast stretch.
- The idea behind contrast stretching is to increase the contrast of the image by making the dark portion darker and bright portion brighter.

- A high-contrast image spans the full range of gray-level values.
- Therefore, a low contrast image can be transformed into a high-contrast image by remapping or stretching the gray-level values such that the histogram spans the full range.
- Contrast stretching also called as image normalization.
- Before the stretching can be performed it is necessary to specify the upper and lower pixel value limits over which the image is to be normalized.
- Often these limits will just be the minimum and maximum pixel values that the image type concerned allows.
- For example for 8-bit gray level images the lower and upper limits might be 0 and 255. Call the lower and the upper limits  $a$  and  $b$  respectively.
- The simplest sort of normalization then scans the image to find the lowest and highest pixel values currently present in the image.
- Call these  $c$  and  $d$ . Then each pixel  $P$  is scaled using the following function:

$$P_{out} = (P_{in} - c) \left( \frac{b - c}{d - c} \right) + a$$

- Values below 0 are set to 0 and values about 255 are set to 255.
- The problem with this is that a single outlying pixel with either a very high or very low value can severely affect the value of  $c$  or  $d$  and this could lead to very unrepresentative scaling.



**Figure 3.7 : Contrast Stretching**

Image source *Digital Image Processing - R. Gonzalez, R. Woods*

- Figure 3.10(b) (above image) shows an 8-bit image with low contrast.
- Figure 3.10(c) (above image) shows the result of contrast stretching, obtained by setting  $(r_1, s_1) = (r_{\min}, 0)$  and  $(r_2, s_2) = (r_{\max}, L-1)$  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, Figure 3.10(d) (above image) shows the result of using the *thresholding function* defined previously, with  $r_1=r_2=m$ , the mean gray level in the image.

### 3.9 IMAGE THRESHOLDING

- In many vision applications, it is useful to be able to separate out the regions of the image corresponding to objects in which we are interested, from the regions of the image that correspond to background.

- Thresholding provides a way to perform this segmentation on the basis of the different intensities or colors in the foreground and background regions of an image.
- In addition, it is also useful to be able to see what areas of an image consist of pixels whose values lie within a specified range.
- The input to a thresholding operation is typically a grayscale or color image.
- The output is a binary image representing the segmentation.
- Black pixels correspond to background and white pixels correspond to foreground (or *vice versa*).
- The segmentation is determined by a single parameter known as the *intensity threshold*.
- Each pixel in the image is compared with this threshold.
- In a 8 bit grey scale image, If the pixel's intensity is higher than the threshold, the pixel is set to white ie. 255 in the output.
- If it is less than the threshold, it is set to black ie.0.
- Above, Figure 3.10(a) shows a typical transformation used for contrast stretching.
- The locations of points  $(r_1, s_1)$  and  $(r_2, s_2)$  control the shape of the transformation function.
- If  $r_1 = s_1$  and  $r_2 = s_2$ , the transformation is a linear function that produces no changes in gray levels.
- If  $r_1 = r_2$ ,  $s_1 = 0$  and  $s_2 = L-1$ , the transformation becomes a *thresholding function* that creates a binary image.

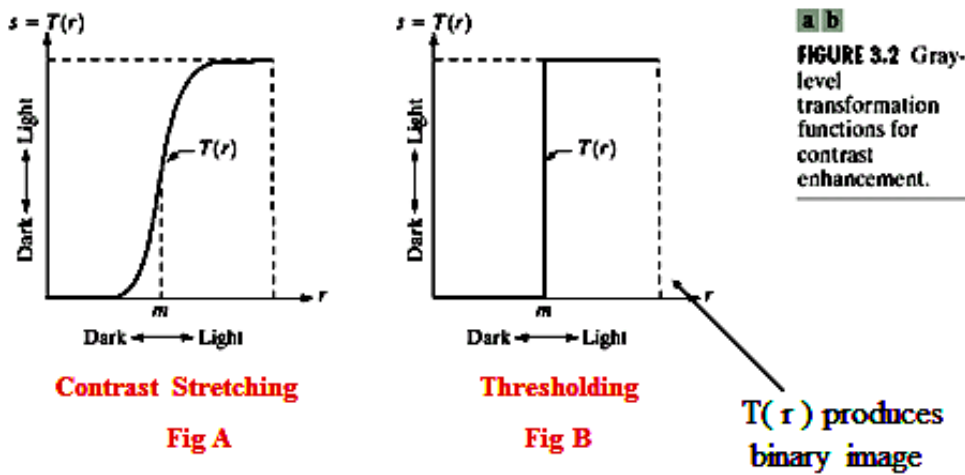


Figure 3.8 Gray Level Transformation

- Intermediate values of  $(r_1, s_1)$  and  $(r_2, s_2)$  produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.
- In general,  $r_1 \leq r_2$  and  $s_1 \leq s_2$  is assumed, so the function is always increasing.

#### Solved example to perform thresholding

**Q]** Obtain the Threshold image for the 3 Bit image shown below.  
Choose appropriate threshold value  $T$ .

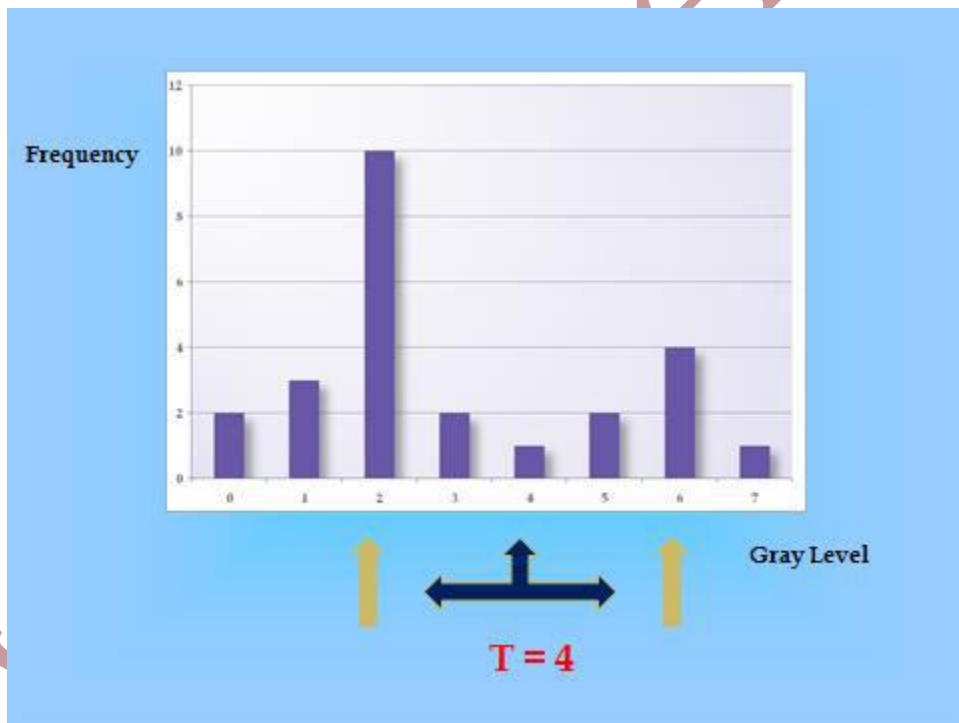
1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

#### Solution:

To get an appropriate threshold value  $T$  we first obtain a frequency table and plot the gray level histogram of the image.

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Gray Level	No. of Pixels
0	2
1	3
2	10
3	2
4	1
5	2
6	4
7	1





For obtaining the Threshold image :

$$f_2(x,y) = \begin{cases} 0 & \text{if } f_1(x,y) \leq T \\ L-1 & \text{if } f_1(x,y) > T \end{cases}$$

$L$  : No. of Gray levels       $(T=4)$

Original image :  $f_1(x,y)$

1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

Threshold image :  $f_2(x,y)$

0	0	0	0	0
0	0	0	7	0
0	7	7	7	0
0	7	7	7	0
0	0	0	0	0

### 3.10 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 allow value for all other gray levels.
- This transformation, shown in Fig 3.11 (a), (figure 3.8) produces abinary image.
- The second approach, based on the transformation shown in Fig 3.11 (b), (figure 3.8) brightens the desired range of gray levels but preserves gray levels unchanged.
- Fig 3.11 (c) (figure 3.8) shows a gray scale image, and
- Fig 3.11 (d) (figure 3.8) shows the result of using the transformation in Fig 3.11 (a).

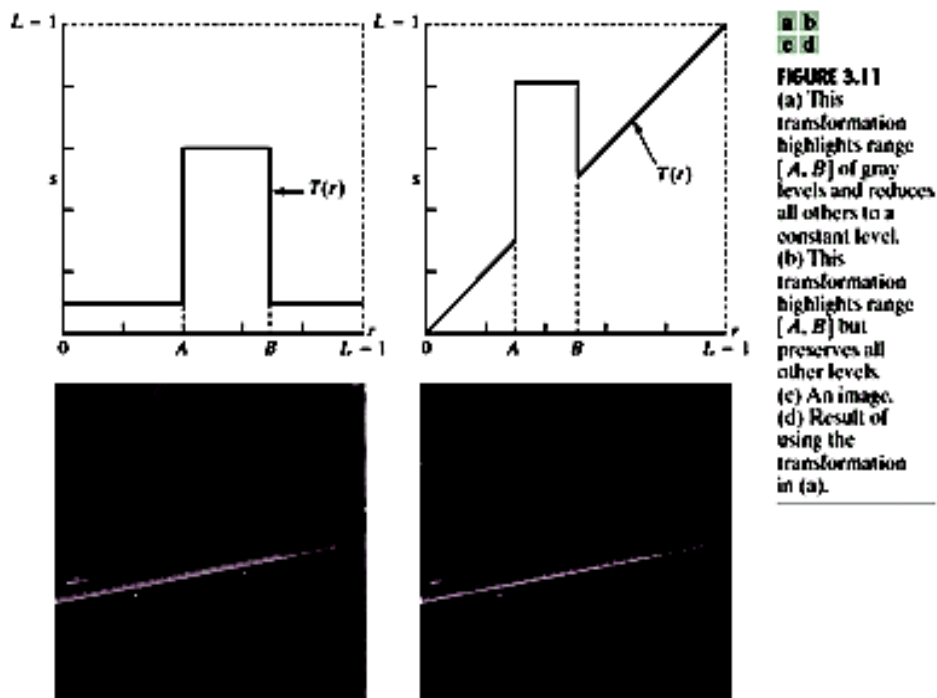


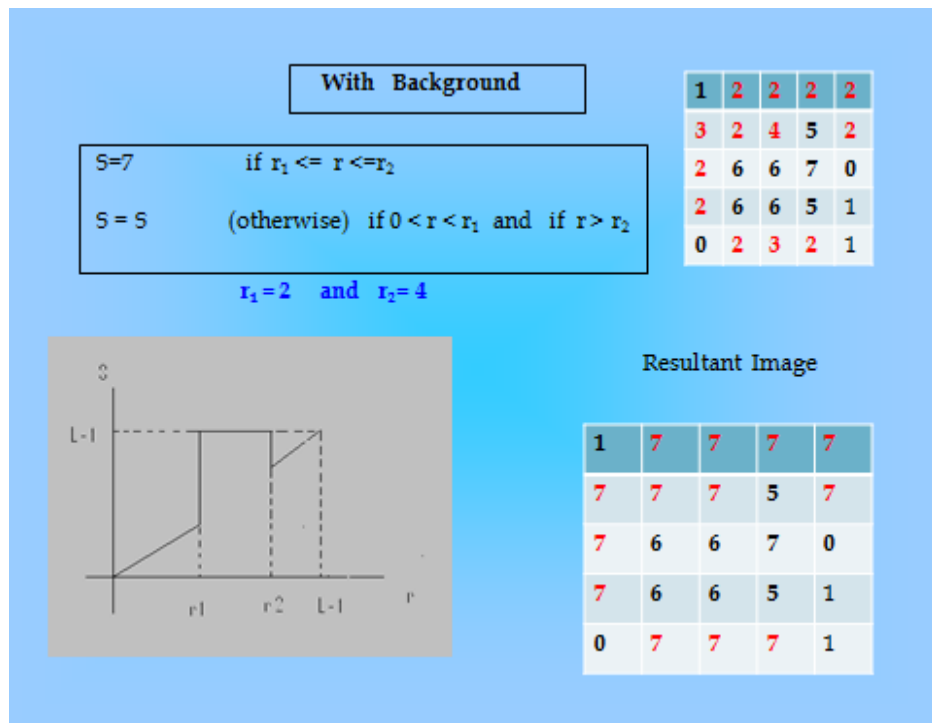
Figure 3.9 : Gray Level slicing without background and with back ground

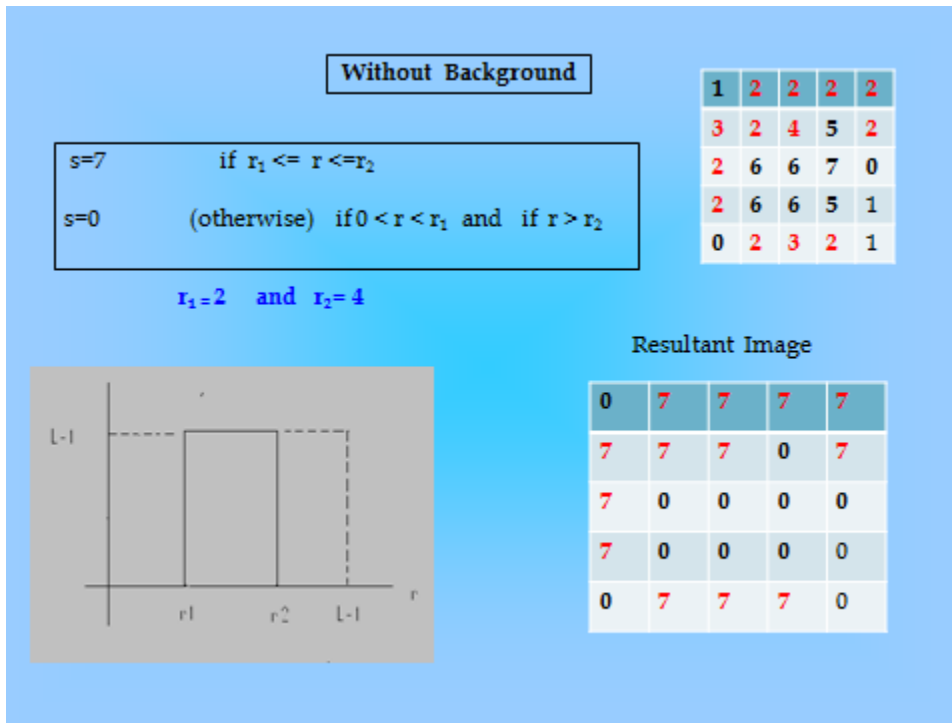
**Solved example to perform gray level slicing**

1. With Background
2. Without Background

**Q] Perform gray level slicing with Background and Without Background on the 3 bit image shown below. ( Assume  $r_1=2$  and  $r_2=4$  ).**

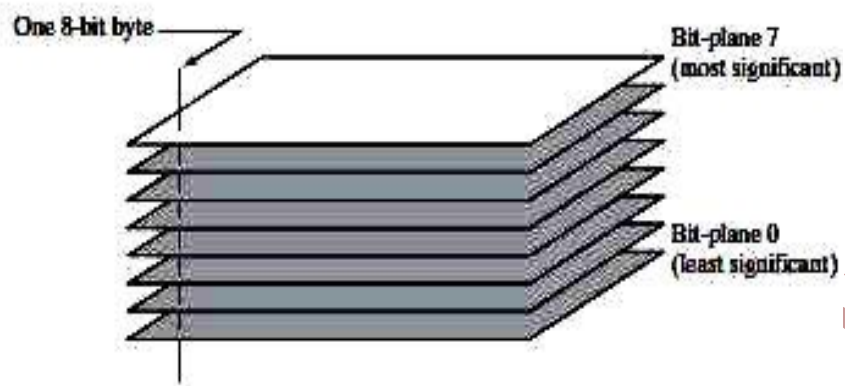
1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1

**Solution : With Background****Solution : Without Background**



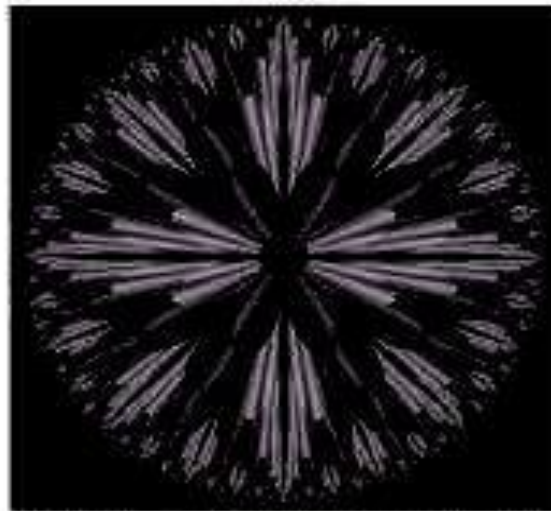
### 3.11 BIT-PLANE SLICING

- Pixels are digital numbers, each one composed of 8 bits.
- The image is composed of 8 1-bit planes.
- Plane 0 contains the least significant bit and plane 7 contains the most significant bit.
- In terms of 8-bits bytes, plane 0 contains all lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all high order bits.
- Instead of highlighting gray-level range, we could highlight the contribution made by each bit
- Only the higher order bits (top four) contain visually significant data.
- The other bit planes contribute the more subtle details.
- This method is useful and used in image compression.



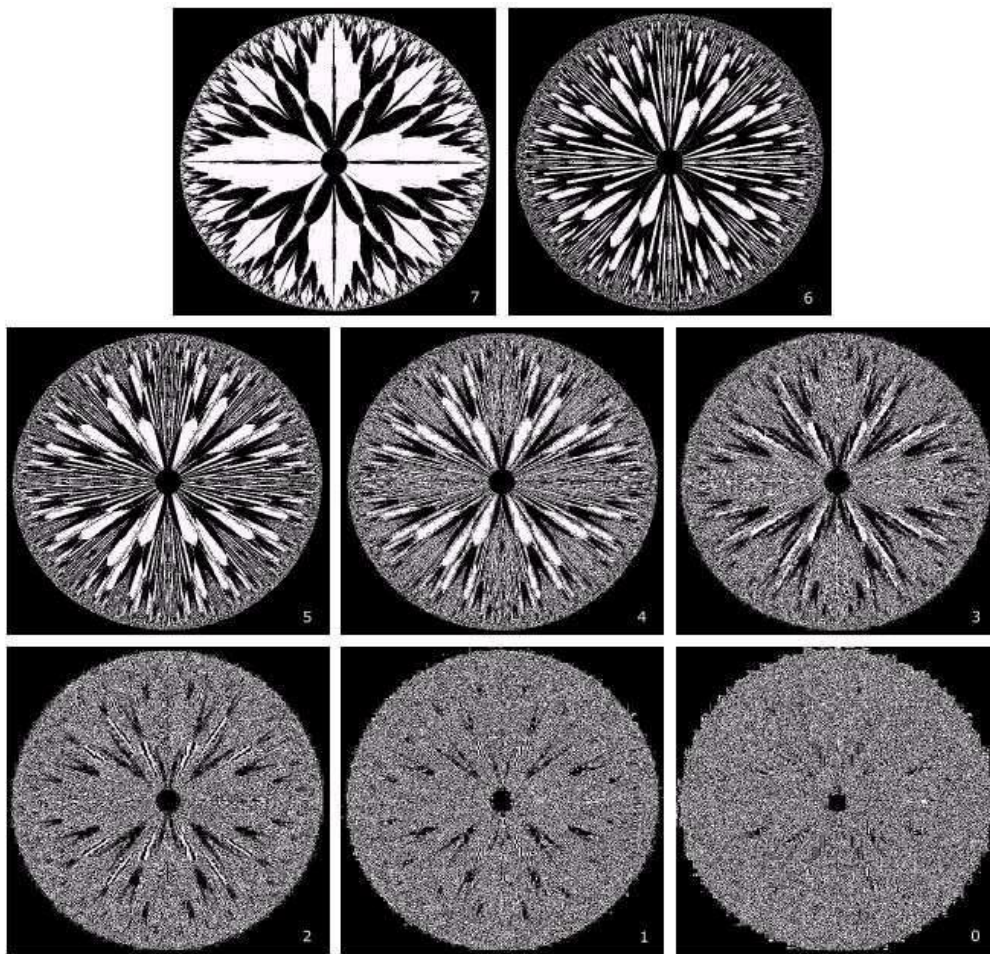
**FIGURE 3.10 : Bit-plane representation of an 8-bit image.**

- Most significant bits contain the majority of visually significant data



**FIGURE 3.11** An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

**Figure 3.11 : Bit fractal image**



**FIGURE 3.12** The eight bit planes of the image in Fig. 3.11. The number at the bottom, right of each image identifies the bit plane.

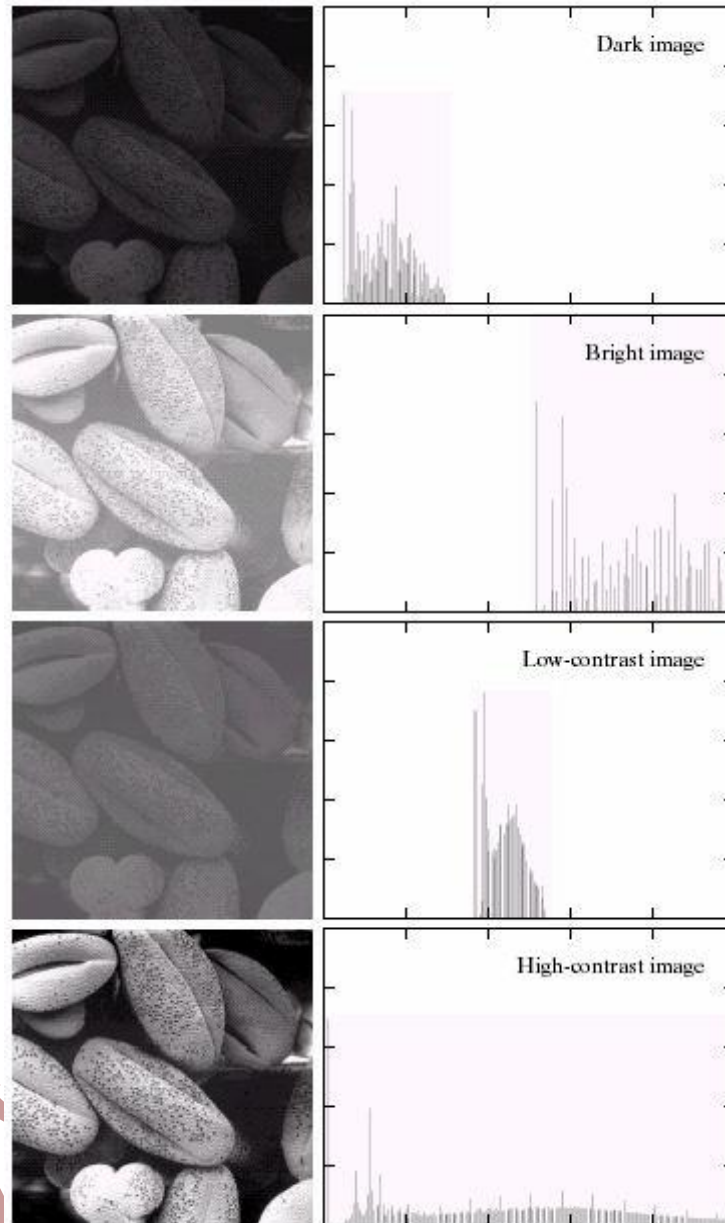
### 3.12 BASICS OF HISTOGRAM PROCESSING

- The histogram of a digital image with gray levels in the range  $[0, L-1]$  is a discrete function of the form  

$$H(r_k) = n_k$$
- where  $r_k$  is the  $k$ th gray level and  $n_k$  is the number of pixels in the image having the level  $r_k$ .
- A normalized histogram is given by the equation  

$$p(r_k) = n_k / n \text{ for } k=0, 1, 2, \dots, L-1$$
- $P(r_k)$  gives the estimate of the probability of occurrence of gray level  $r_k$ .
- The sum of all components of a normalized histogram is equal to 1.

- The histogram plots are simple plots of  $H(r_k) = n_k$  versus  $r_k$ .



**Figure 3.13 : Histogram of Image**

- In the dark image the components of the histogram are concentrated on the low (dark) side of the gray scale.
- In case of bright image the histogram components are biased towards the high side of the gray scale.
- The histogram of a low contrast image will be narrow and will be centered towards the middle of the gray scale.



- The components of the histogram in the high contrast image cover a broad range of the gray scale.
- The net effect of this will be an image that shows a great deal of gray levels details and has high dynamic range.

### 3.13 HISTOGRAM EQUALISATION (AUTOMATIC)

- Histogram equalization is a common technique for enhancing the appearance of images.
- Suppose we have an image which is predominantly dark.
- Then its histogram would be skewed towards the lower end of the grey scale and all the image detail are compressed into the dark end of the histogram.
- If we could 'stretch out' the grey levels at the dark end to produce a more uniformly distributed histogram then the image would become much clearer.
- Let there be a continuous function with  $r$  being gray levels of the image to be enhanced.
- The range of  $r$  is  $[0, 1]$  with  $r=0$  representing black and  $r=1$  representing white.
- The transformation function is of the form

$$S=T(r) \text{ where } 0 < r < 1$$

- It produces a level  $s$  for every pixel value  $r$  in the original image.
- The transformation function is assumed to fulfill two condition  $T(r)$  is single valued and monotonically increasing in the interval

$$0 < T(r) < 1 \quad \text{for } 0 < r < 1$$

- The transformation function should be single valued so that the inverse transformations should exist.
- Monotonically increasing condition preserves the increasing order from black to white in the output image.
- The second conditions guarantee that the output gray levels will be in the same range as the input levels.
- The gray levels of the image may be viewed as random variables in the interval  $[0, 1]$
- The most fundamental descriptor of a random variable is its probability density function (PDF)
- $Pr(r)$  and  $Ps(s)$  denote the probability density functions of random variables  $r$  and  $s$  respectively.
- Basic results from an elementary probability theory states that if  $Pr(r)$  and  $Tr$  are known and  $T^{-1}(s)$  satisfies conditions (a), then the probability density function  $Ps(s)$  of the transformed variable  $s$  is given by the formula-



$$P_s(s) = P_r(r) \frac{dr}{ds},$$

- Thus the PDF of the transformed variable s is determined by the gray levels PDF of the input image and by the chosen transformation function.
- A transformation function of a particular importance in image processing

$$s = T(r) = \int_Q^r P_r(w) dw.$$

This is the cumulative distribution function of r.

- Using this definition of T we see that the derivative of s with respect to r is

$$\frac{ds}{dr} = P_r(r).$$

- Substituting it back in the expression for  $P_s$  we may get

$$P_s(s) = P_r(r) \frac{1}{P_r(r)} = 1$$

- An important point here is that  $T_r$  depends on  $P_r(r)$  but the resulting  $P_s(s)$  always is uniform, and independent of the form of  $P(r)$ .
- For discrete values we deal with probability and summations instead of probability density functions and integrals.
- The probability of occurrence of gray levels  $r_k$  in an image as approximated

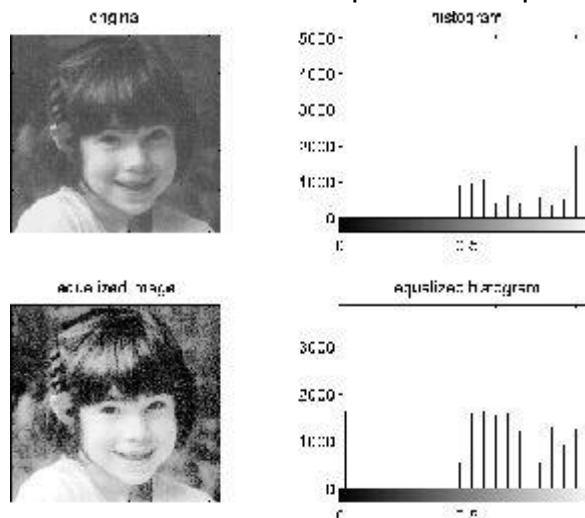
$$P_r(r) = nk / N$$

- N is the total number of the pixels in an image.
- nk is the number of the pixels that have gray level  $r_k$ .
- L is the total number of possible gray levels in the image.
- The discrete transformation function is given by

$$\begin{aligned} S_k = T(r_k) &= \sum_{i=Q}^k \frac{n_i}{N} \\ &= \sum_{i=Q}^k P_r(r_i) \end{aligned}$$

- Thus a processed image is obtained by mapping each pixel with levels  $r_k$  in the input image into a corresponding pixel with level  $s_k$  in the output image.
- A plot of  $P_r(r_k)$  versus  $r_k$  is called a histogram.

- The transformation function given by the above equation is the called histogram equalization or linearization.
- Given an image the process of histogram equalization consists simple of implementing the transformation function which is based information that can be extracted directly from the given image, without the need for further parameter specification.



**Figure 3.14 : Histogram Equalization**

- Equalization automatically determines a transformation function that seeks to produce an output image that has a uniform histogram.
- It is a good approach when automatic enhancement is needed

### 3.14 HISTOGRAM MATCHING (SPECIFICATION)

- In some cases it may be desirable to specify the shape of the histogram that we wish the processed image to have.
- Histogram equalization does not allow interactive image enhancement and generates only one result: an approximation to a uniform histogram.
- Sometimes we need to be able to specify particular histogram shapes capable of highlighting certain gray-level ranges.
- The method use to generate a processed image that has a specified histogram is called histogram matching or histogram specification.
- Algorithm
  1. Compute  $s_k = P_f(k)$ ,  $k = 0, \dots, L-1$ , the cumulative normalized histogram of  $f$ .
  2. Compute  $G(k)$ ,  $k = 0, \dots, L-1$ , the transformation function, from the given histogram  $h_z$
  3. Compute  $G^{-1}(s_k)$  for each  $k = 0, \dots, L-1$  using an iterative method (iterate on  $z$ ), or in effect, directly compute  $G^{-1}(P_f(k))$

4. Transform  $f$  using  $G^{-1}(P_f(k))$ .

### 3.15 LOCAL HISTOGRAM EQUALISATION

- In earlier methods pixels were modified by a transformation function based on the gray level of an entire image.
- It is not suitable when enhancement is to be done in some small areas of the image.
- This problem can be solved by local enhancement where a transformation function is applied only in the neighborhood of pixels in the interested region.
- Define square or rectangular neighborhood (mask) and move the center from pixel to pixel.
- For each neighborhood
  1. Calculate histogram of the points in the neighborhood
  2. Obtain histogram equalization/specification function
  3. Map gray level of pixel centered in neighborhood
  4. The center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated.

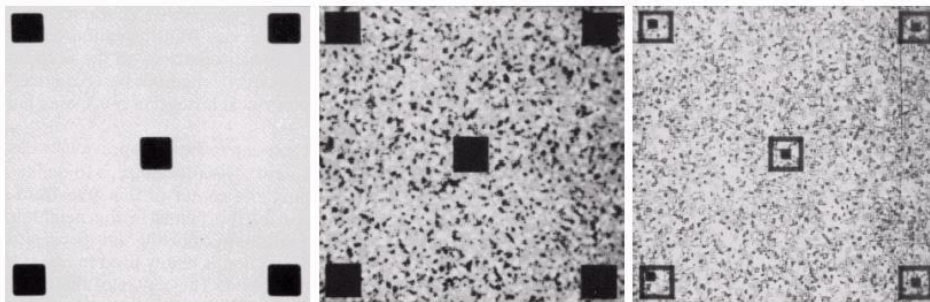


Figure 3.15 : Local Histogram Equalization

### 3.16 ENHANCEMENT USING ARITHMETIC/ LOGIC OPERATION

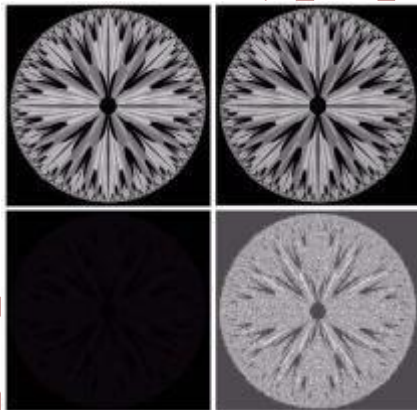
- These operations are performed on a pixel by basis between two or more images excluding not operation which is performed on a single image.
- It depends on the hardware and/or software that the actual mechanism of implementation should be sequential, parallel or simultaneous.
- Logic operations are also generally operated on a pixel by pixel basis.
- Only AND, OR and NOT logical operators are functionally complete.
- Because all other operators can be implemented by using these operators.

- While applying the operations on gray scale images, pixel values are processed as strings of binary numbers.
- The NOT logic operation performs the same function as the negative transformation.
- Image Masking is also referred to as region of Interest (RoI) processing.
- This is done to highlight a particular area and to differentiate it from the rest of the image.
- Out of the four arithmetic operations, subtraction and addition are the most useful for image enhancement.

### **3.16.1 Image Subtraction**

- The difference between two images  $f(x,y)$  and  $h(x,y)$  is expressed as  

$$g(x,y) = f(x,y) - h(x,y)$$
- It is obtained by computing the difference between all pairs of corresponding pixels from  $f$  and  $h$ .
- The key usefulness of subtraction is the enhancement of difference between images.
- This concept is used in another gray scale transformation for enhancement known as bit planeslicing
- The higher order bit planes of an image carry a significant amount of visually relevant detail while the lower planes contribute to fine details.
- If we subtract the four least significant bit planes from the image
- The result will be nearly identical but there will be a slight drop in the overall contrast due to less variability in the graylevel values of image.



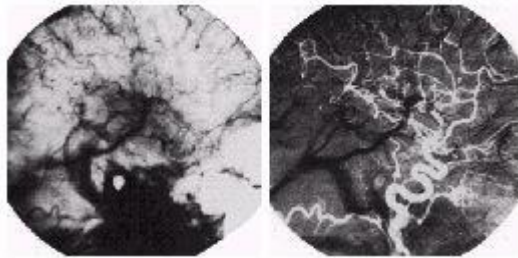
**Figure 3.16 Image Subtraction**

- The use of image subtraction is seen in medical imaging area named as mask mode radiography.
- The mask  $h(x,y)$  is an X-ray image of a region of a patient's body this image is captured by using an intensified TV camera located opposite to the x-ray machine then a consistent

medium is injected into the patient's blood stream and then a series of images are taken of the region same as

$h(x,y)$ .

- The mask is then subtracted from the series of incoming image.
- This subtraction will give the area which will be the difference between  $f(x,y)$  and  $h(x,y)$  this difference will be given as enhanced detail in the output image.



**Figure 3.17 : Image Subtraction**  $f(x,y) - h(x,y)$

### 3.16.2 Image Averaging

- Consider a noisy image  $g(x,y)$  formed by the addition of noise  $n(x,y)$  to the original image  $f(x,y)$

$$g(x,y) = f(x,y) + n(x,y)$$

- Assuming that at every point of coordinate  $(x,y)$  the noise is uncorrelated and has zero average value
- The objective of image averaging is to reduce the noise content by adding a set of noise images,

$$\{g_i(x,y)\}$$

- If an image is formed by image averaging  $K$  different noisy images

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

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

- As  $k$  increases the variability (noise) of the pixel value at each location  $(x,y)$  decreases.
- $E\{g(x,y)\} = f(x,y)$  means that  $g(x,y)$  approaches  $f(x,y)$  as the number of noisy images used in the averaging process increases.
- Image averaging is important in various applications such as in the field of astronomy where the images are low light levels.

## 3.10 SUMMARY

- Spatial domain refers to the image plane itself.
- The three basic types of functions used frequently for image enhancement:

1. Linear Functions:

- Identity Transformation
- Negative Transformation
- Contrast Stretching
- Thresholding

2. Logarithmic Functions:

- Log Transformation
- Inverse-log Transformation

3. Power-Law Functions:

- nth power transformation
- nth root transformation

- The negative of a digital image is obtained by the transformation function:

$$s = T(r) = L - 1 - r$$

- The log transformations can be defined by this formula:

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

- Power-law transformations have the basic form of:

$$s = c.r^Y$$

- Types of Piecewise transformations are:

- Contrast Stretching
- Gray-level Slicing
- Bit-plane slicing

- Contrast enhancements improve the perceptibility of objects in the scene by enhancing the brightness difference between objects and their backgrounds.
- Thresholding provides a way to perform this segmentation on the basis of the different intensities or colors in the foreground and background regions of an image.
- Gray level slicing is the technique used to highlight a specific range of gray levels in a given image.

- The image is composed of 8 1-bit planes.
- Plane 0 contains the least significant bit and plane 7 contains the most significant bit.
- Only the higher order bits (top four) contain visually significant data.
- In an image histogram, the x axis shows the gray level intensities and the y axis shows the frequency of these intensities.
- Histogram equalization is a common technique for enhancing the appearance of images.
- Equalization automatically determines a transformation function that seeks to produce an output image that has a uniform histogram.
- The difference between two images  $f(x,y)$  and  $h(x,y)$  is expressed as

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

---

### 3.11 UNIT END EXERCISE

---

- 1) Specify the objective of image enhancement technique and List the 2 categories of image enhancement.
- 2) What is the purpose of image averaging?
- 3) What is meant by masking?
- 4) Define histogram. What is meant by histogram equalization?
- 5) State the different types of processing used for image enhancement.
- 6) Explain the Basic Gray Level Transformations used in image enhancement.
- 7) Explain: i) Contrast stretching ii) Gray level slicing iii) Bit plane slicing.
- 8) Explain image enhancement using Arithmetic & Logic operations.
- 9) A particular digital image with eight quantization levels has the following histogram . Perform histogram equalization and derive transformation function. Give new equalized histogram.

Gray level $r$	0	1	2	3	4	5	6
No. of pixels with gray level $r$	200	170	130	60	60	80	140

- 10) Obtain the Threshold image for the 3 Bit image below.

Choose appropriate threshold value  $T$ .

1	2	2	2	2
3	3	5	6	2
7	4	4	7	0
7	6	6	5	0
0	7	3	7	1

11) Obtain the digital negative image of the 3 Bit image

	2	1	4
7	0	7	0
5	4	6	2
4	4	3	1

12) Perform gray level slicing with Background and Without Background on the 3 bit image shown

below. ( Assume  $r_1 = 3$  and  $r_2 = 6$  ).

4	3	6	7
1	2	5	3
4	3	0	2
0	7	6	5

### 3.13 FURTHER READING

1. R.C.Gonsales R.E.Woods, .Digital Image Processing., Second Edition, Pearson Education.



2. B. Chanda, D. Dutta Majumder, *.Digital Image Processing and Analysis.*, PHI.
3. Digital Image Processing by Dhananjay Thekedat



www.rocktheit.com