Module-2

Image Enhancement in Spatial domain

Spatial Domain: Some Basic Intensity Transformation Functions, Histogram Processing, Fundamentals of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters (Ch 3: Sections 3.2 to 3.6)

Frequency Domain: Preliminary Concepts, The Discrete Fourier Transform (DFT) of Two Variables, Properties of the 2-D DFT, Filtering in the Frequency Domain, Image Smoothing and Image Sharpening Using Frequency Domain Filters, Selective Filtering. (Ch 4: Sections 4.2, 4.5 to 4.10)

Image Enhancement in Spatial domain

2.1 Introduction

The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application. Image enhancement approaches fall into two board categories :

1. Spatial domain methods : The term spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image.
2. Frequency domain methods : Frequency domain processing techniques are based on modifying the Fourier transform of an image.

Enhancing an image provides better contrast and more detailed image as compared to non-enhanced image. Image enhancement is used to enhance medical images, images captured in remote sensing, images from satellite e.t.c.

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.

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

where f(x, y) is the input image, g(x, y) is the processed image, and **T** is an operator on f, defined over some neighborhood of (x, y).

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 neighbourhood of (x, y) .

(iii) T may operate to a set of input images instead of a single image.

The principal approach in defining a neighborhood about a point (x, y) is to use a square or rectangular subimage area centered at (x, y), as Fig. 2.1 shows. The center of the subimage is moved from pixel to pixel starting, say, at the top left corner. The operator T is applied at each location (x, y) to yield the output, g, at that location. The process utilizes only the pixels in the area of the image spanned by the neighborhood.

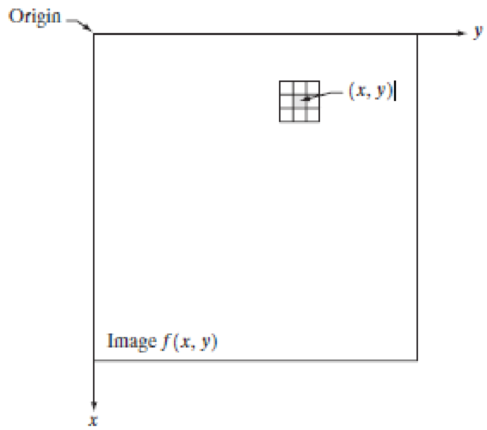


Fig.: 3x3 neighborhood about a point (x,y) in an image.

2.2 BASIC GRAY LEVEL TRANSFORMATIONS:

The simplest form of T is when the neighborhood is 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 gray-level (also called an intensity or mapping) transformation function of the form

$$s = T(r)$$

where

r is the pixels of the input image f(x,y) before processing

s is the pixels of the output image g(x,y). after processing

T is a transformation function that maps each value of "r" to each value of "s".

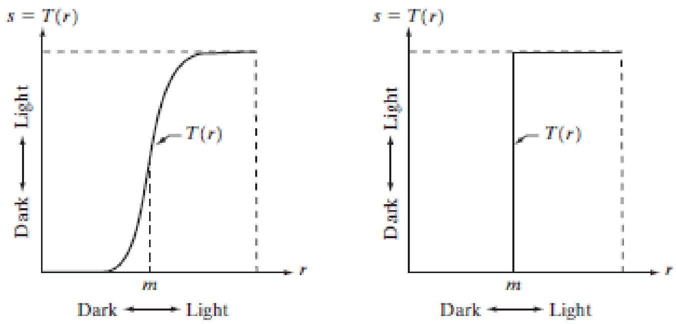


Fig. 2.2 Gray level transformation functions for contrast enhancement.

For example, if T(r) has the form shown in Fig. 2.2(a), the effect of this transformation would be to produce an image of higher contrast than the original by darkening the levels below m and brightening the levels above m in the original image. This technique is known as contrast stretching. In Fig. 2.2(b), T(r) produces a two-level (binary) image. A mapping of this form is called a thresholding function.

There are three basic types of functions used frequently for image enhancement:

- Linear (negative and identity transformations)
- Logarithmic (log and inverse-log transformations)
- Power-law (nth power and nth root transformations).

LINEAR TRANSFORMATION:

Linear transformation includes simple identity and negative transformation.

IDENTITY OR LAZY MAN OPERATION

The identity function is the trivial case in which output intensities are identical to input intensities and hence is called identity transformation as shown below:

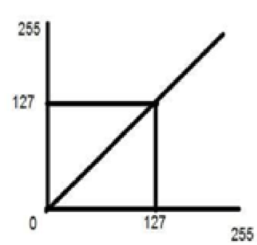


Fig. Linear transformation between input and output.

NEGATIVE TRANSFORMATION (IMAGE NEGATIVE):

Negative transformation is invert of identity transformation. The image negative with gray level value in the range of [0, L-1] is obtained by negative transformation given by

$$S = T(r) \text{ or}$$

$$S = L - 1 - r$$

Where r= gray level value at pixel (x,y), L is the largest gray level in the image

It is useful when for enhancing white details embedded in dark regions of the image.

$$s = L - 1 - r$$

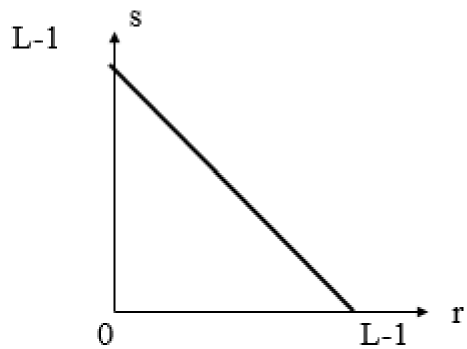


FIG. Negative Trransformation

In this eample, since the input image is an 8 bpp image, so the number of levels in this image are 256. Putting L=256 in the equation,

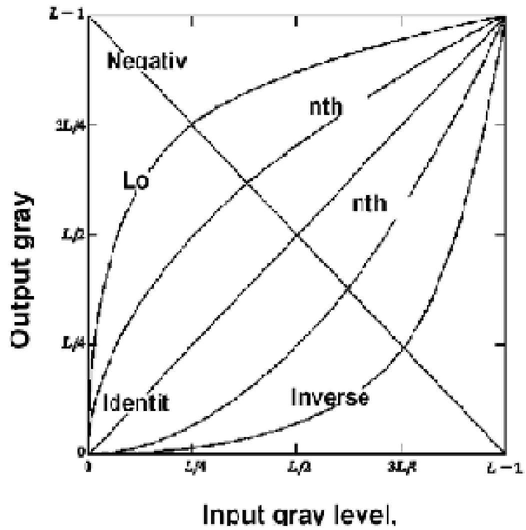
$$S = (L - 1) - r$$

$$S = 255 - r$$

So each value is subtracted by 255 and the result image has been shown above. So what happens is that, the lighter pixels become dark and the darker picture becomes light. And it results in image negative.

LOGARITHMIC TRANSFORMATIONS:

Logarithmic transformation further contains two type of transformation. Log transformation and inverse log transformation as shown in figure below



LOG TRANSFORMATIONS:

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. The shape of the log curve shows that this transformation maps a narrow range of low gray-level values in the input image into a wider range of output levels. The opposite is true of higher values of input levels. We would use a transformation of this type to expand the values of dark pixels in an image while compressing the higher-level values.

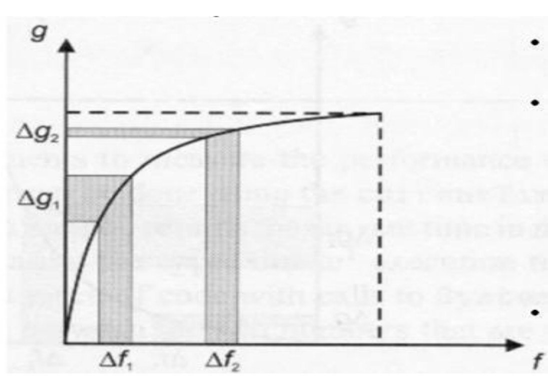


Fig. log transformation curve input vs output

POWER – LAW TRANSFORMATIONS:

The power law transformations include nth power and nth root transformation. These transformations can be given by the expression:

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

where c and γ are positive constants.

The symbol γ is called gamma, due to which this transformation is also known as *gamma transformation.*

Variation in the value of γ varies the enhancement of the images. Different display devices/ monitors have their own gamma correction, thats why they display their image at different intensity.

Plots of s versus r for various values of γ are shown in Fig. below

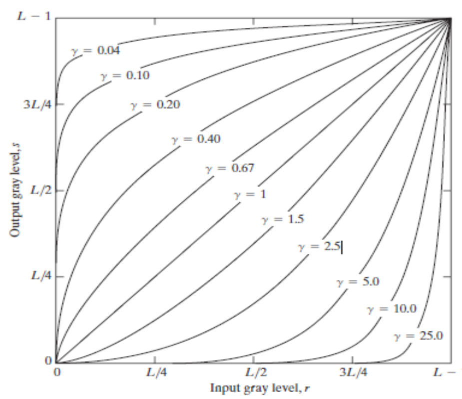


Fig. 2.13 Plot of the equation $S = cr^{\gamma}$ for various values of γ (c =1 in all cases).

$\gamma > 1$ compresses dark values

Expands bright values

$\gamma < 1$ (similar to Log transformation)

Expands dark values

Compresses bright values

When $C = \gamma = 1$, it reduces to identity transformation .

This type of transformation is used for enhancing images for different type of display devices. The gamma of different display devices is different. For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark. Varying gamma (γ) obtains family of possible transformation curves

Piecewise-Linear Transformation Functions:

Low contrast images can occur due to poor or non uniform lighting conditions, due to non linearity in imaging sensors or due to small dynamic range of imaging sensors

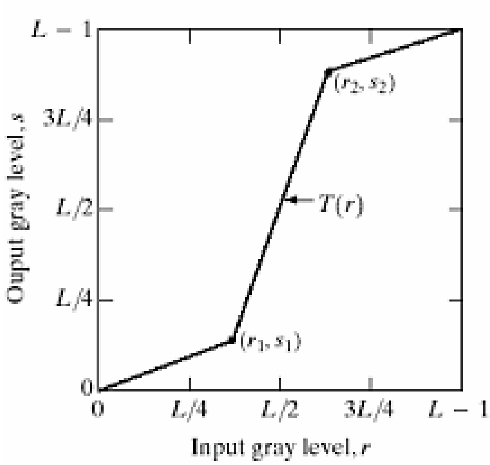
A complementary approach to the methods discussed in the previous three sections is to use piecewise linear functions. The principal advantage of piecewise linear functions is that the form of piecewise functions can be arbitrarily complex but their specification requires considerably more user input. The principal disadvantage of piecewise functions is that their specification requires considerably more user input.

Contrast stretching:

One of the simplest piecewise linear functions is a contrast-stretching transformation. Low-contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture during image acquisition. The idea behind contrast stretching is to increase the dynamic range of gray levels in the image being processed.

$$S = T(r)$$

Figure below shows a typical transformation used for contrast stretching.



If slope>1, stretching of gray values

If slope< 1, contrast reduction

Slope=1, no change

The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function.

a) If r1=r2 and s1=s2, the transformation is a linear function that deduces no change in gray levels.

b) If r1=s1, s1=0 , and s2=L-1, then the transformation become a thresholding function that creates a binary image

c) Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the gray value of the output image thus effecting its contract.

Generally r1≤ r2 and s1 ≤ s2 so that the function is single valued and monotonically increasing.

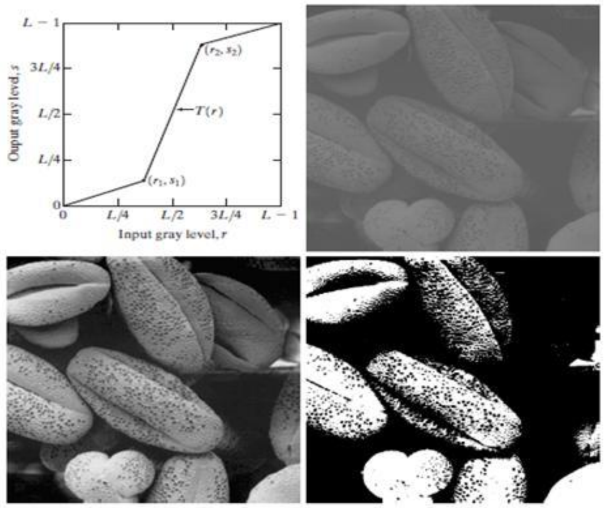
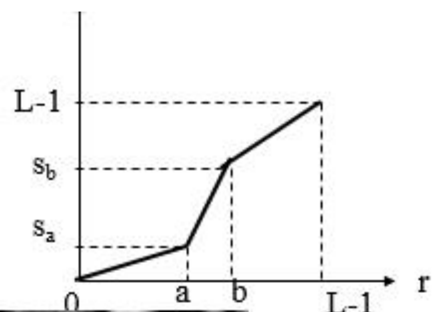
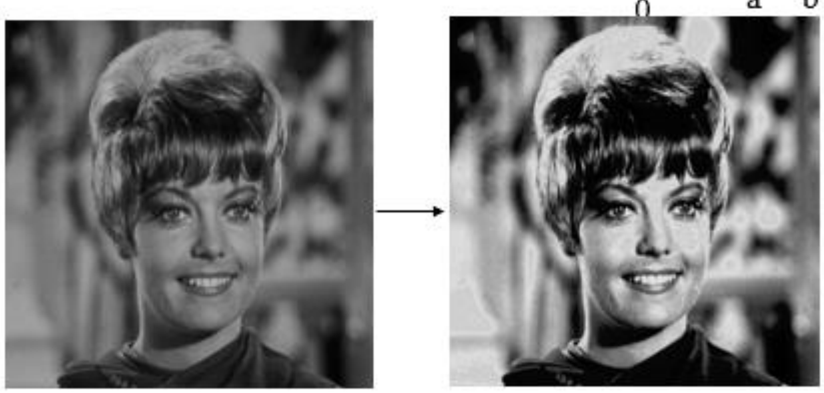


Fig. x Contrast stretching. (a) Form of transformation function. (b) A low-contrast stretching. (c) Result of contrast stretching. (d) Result of thresholding (original image courtesy of Dr.Roger Heady, Research School of Biological Sciences, Australian National University Canberra Australia).

$$s = \begin{cases} \alpha r & 0 \le r < a \\ \beta(r-a) + s_a & a \le r < b \\ \gamma(r-b) + s_b & b \le r < L \end{cases}$$





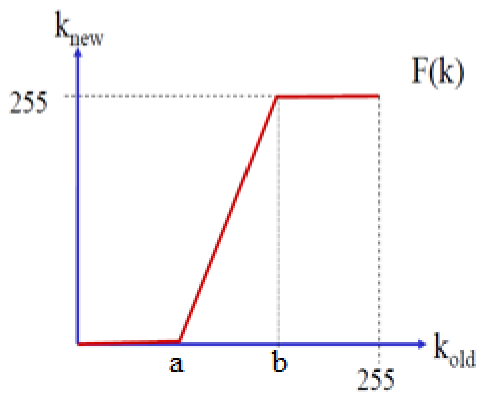
Find new image pixel values after applying Contrast stretching

$$a = 50, b = 150, \alpha = 0.2, \beta = 2, \gamma = 1, s_a = 30, s_b = 200$$

0	**10**	**50**	**100**
5	95	150	200
110	150	190	210
175	210	255	100

Clipping Function

If most of the gray-levels in the image are in [a b], the following mapping increases the image contrast. All other gray values are mapped to either zero or maximum value. This is useful in medical images like CT, MRI images



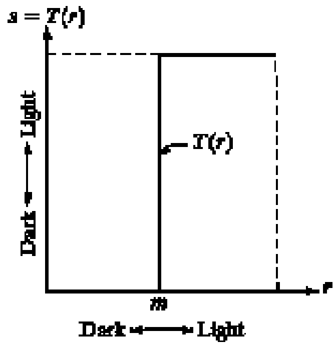
Here $\alpha = \gamma = 0$

For the given sub-image, if a=50 and b =170, β=2, Find the new image after gray level slicing

0	**10**	**50**	**100**
5	95	150	200
110	150	190	210
175	210	255	100

Thresholding

Produces a two-level image or binary image

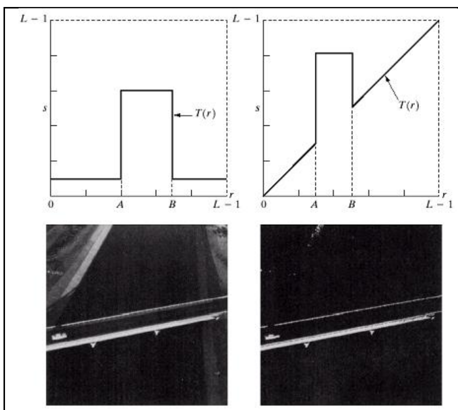


Here $\alpha = \gamma = 0$ and a=b=t

All gray values above t are mapped to maximum value and below t are mapped to minimum value

Gray-level slicing:

Highlighting a specific range of gray levels in an image. Displays a high value of all gray levels in the range of interest and a low value for all other gray levels. There are several ways of doing level slicing, but most of them are variations of two basic themes. 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. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images.

 (a) (b)	(a) transformation highlights range [A,B] of gray level and reduces all others to a constant level (b) transformation highlights range [A,B] but preserves all other levels

This transformation, shown in Fig. (a), produces a binary image. The second approach, based on the transformation shown in Fig. (b), brightens the desired range of gray levels but preserves the background and gray-level tonalities in the image.

BIT-PLANE SLICING:

Instead of highlighting gray-level ranges, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits. Imagine that the image is composed of eight 1-bit planes, ranging from bit-plane 0 for the least significant bit to bit plane 7 for the most significant bit. In terms of 8-bit bytes, plane 0 contains all the lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all the high-order bits.

Figure below illustrates these ideas, and shows the various bit planes. Note that the higher-order bits (especially the top four) contain the majority of the visually significant data.The other bit planes contribute to more subtle details in the image. Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, a process that aids in determining the adequacy of the number of bits used to quantize each pixel.

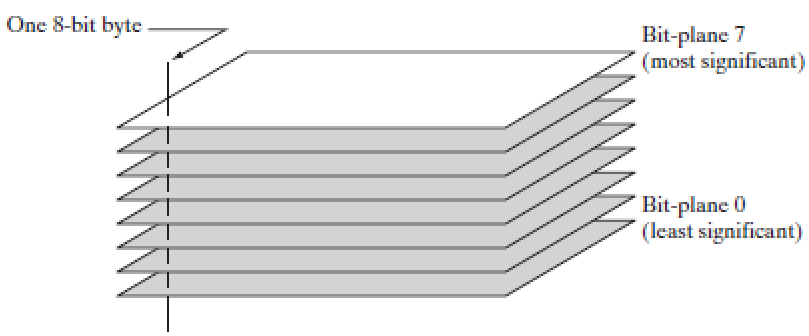


FIGURE Bit-plane representation of an 8-bit image.

In terms of bit-plane extraction for an 8-bit image, it is not difficult to show that the (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation function that (1) maps all levels in the image between 0 and 127 to one level (for example, 0); and (2) maps all levels between 129 and 255 to another (for example, 255).

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 kth 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

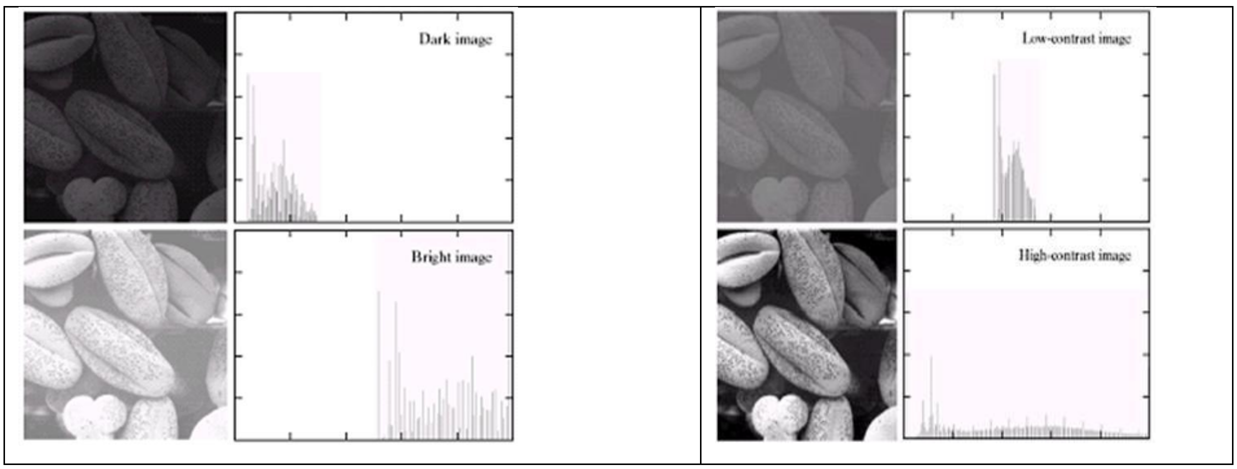
$$p(r_k)=n_k/n \text{ for } k=0,1,2,\ldots,L-1$$

$P(r_k)$ 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.

  - 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 baised 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.



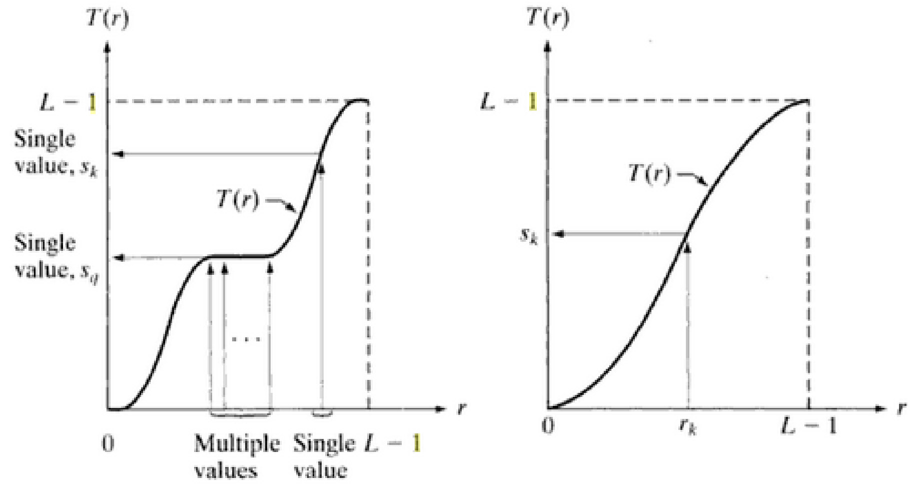
Histogram Equalization:

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 repressing 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.



a b

FIGURE
(a) Monotonically increasing function, showing how multiple values can map to a single value.
(b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

The transformation function is assumed to fulfill two condition T(r) is single valued and monotonically increasing in the internal 0<T(r)<1 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 is given by the formula

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

Thus the PDF of the transformed variable s is the determined by the gray levels PDF of the input image and by the chosen transformations function.

A transformation function of a particular importance in image processing

$$s = T(r) = (L - 1)\int_0^r p_r(w)\, dw$$

This is the cumulative distribution function of r.

L is the total number of possible gray levels in the image.

One of the principal approaches in this formulation is based on the use of so-called masks (also referred to as filters, kernels, templates, or windows). Basically, a mask is a small (say, 3*3) 2-D array, such as the one shown in Fig. 2.1, 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.