

### Basic Gray Level Transformations:

- Image negative
- Log transformations
- Power law transformations
- Piecewise-Linear transformation functions

#### LINEAR TRANSFORMATION:

First we will look at the linear transformation. Linear transformation includes simple identity and negative transformation. Identity transformation has been discussed in our tutorial of image transformation, but a brief description of this transformation has been given here.

Identity transition is shown by a straight line. In this transition, each value of the input image is directly mapped to each other value of output image. That results in the same input image and output image. And hence is called identity transformation. It has been shown below:

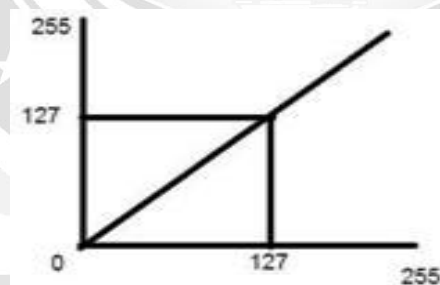


Fig 2.1.1 Linear transformation between input and output

(Source: Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition, 2010. - Page – 108)

#### NEGATIVE TRANSFORMATION:

The second linear transformation is negative transformation, which is invert of identity transformation. In negative transformation, each value of the input image is subtracted from the  $L-1$  and mapped onto the output image

**IMAGE NEGATIVE:** The image negative with gray level value in the range of  $[0, L-1]$  is obtained by negative transformation given by  $S = T(r)$  or

$$S = L - 1 - r$$

Where  $r$  = gray level value at pixel  $(x,y)$

$L$  is the largest gray level consists in the image

It results in getting photograph negative. It is useful when for enhancing white details embedded in dark regions of the image. The overall graph of these transitions has been shown below.

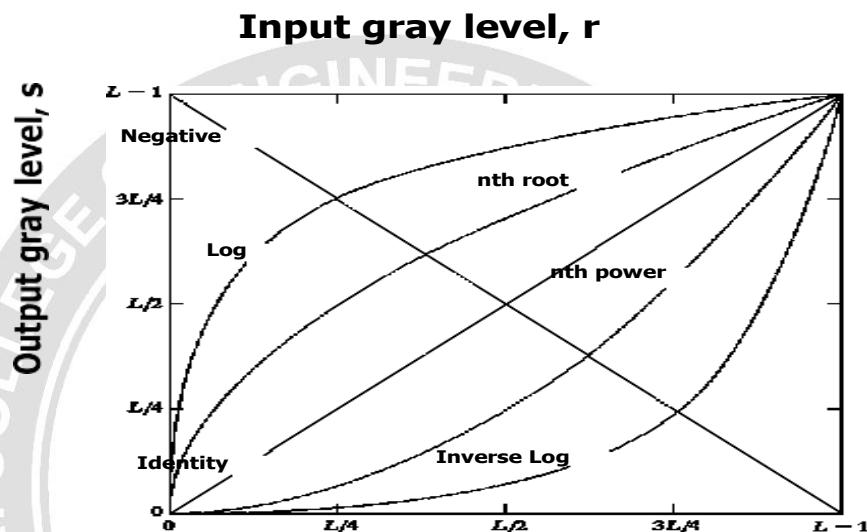


Fig2.1.2: Some basic gray-level transformation functions used for image enhancement.

(Source:-, Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition, 2010. Page- 108)

In this case the following transition has been done.

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

since the input image of Einstein is an 8 bpp image, so the number of levels in this image are 256. Putting 256 in the equation, we get this  $s = 255 - r$

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.

It has been shown in the graph below.

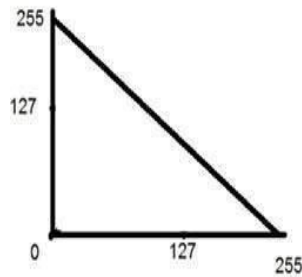


Fig: 2.1.3: Negative transformations.

(Source: Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition, 2010. -Page-108)

### LOGARITHMIC TRANSFORMATIONS:

Logarithmic transformation further contains two type of transformation. Log transformation and inverse log transformation.

#### 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.

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.

An another way of representing LOG TRANSFORMATIONS: Enhance details in the darker regions of an image at the expense of detail in brighter regions.

$$T(f) = C * \log(1+r)$$

- Here  $C$  is constant and  $r \geq 0$ .
- The shape of the curve shows that this transformation maps the narrow range of low gray level values in the input image into a wider range of output image.
- The opposite is true for high level values of input image.

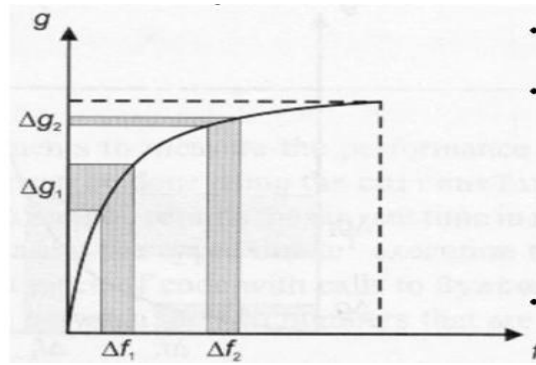


Fig2.1.4: Logarithmic Transformation

(Source: Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Pearson, Third Edition, 2010. – Page- 109)

### POWER – LAW TRANSFORMATIONS:

There are further two transformation is power law transformations, that include  $n$ th power and  $n$ th root transformation. These transformations can be given by the expression:

$$s = cr^\gamma$$

This symbol  $\gamma$  is called gamma, due to which this transformation is also known as gamma transformation.

Variation in the value of  $\gamma$  varies the enhancement of the images. Different display devices / monitors have their own gamma correction, that's why they display their image at different intensity.

where  $c$  and  $g$  are positive constants. Sometimes Eq. (6) is written as  $S = C (r + \epsilon)^\gamma$  to account for an offset (that is, a measurable output when the input is zero). Plots of  $s$  versus  $r$  for various values of  $\gamma$  are shown in Fig. 2.10. As in the case of the log transformation, power-law curves with fractional values of  $\gamma$  map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels. Unlike the log function, however, we notice here a family of possible transformation curves obtained simply by varying  $\gamma$ .

In Fig that curves generated with values of  $\gamma > 1$  have exactly The opposite effect as those generated with values of  $\gamma < 1$ . Finally, we Note that Eq. (6) reduces to the identity transformation when  $c = \gamma = 1$ .

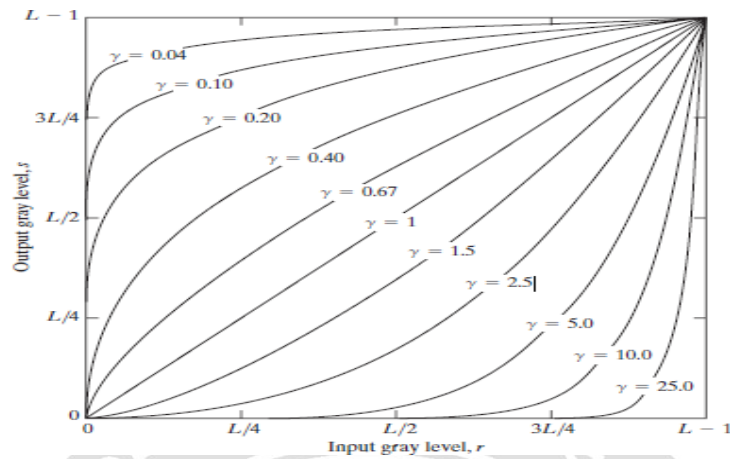


Fig. 2.1.5 Plot of the equation  $S = cr^\gamma$  for various values of  $\gamma$  ( $c = 1$  in all cases).

(Source:- Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Pearson, Third Edition, 2010. Page- 110)

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 ( $\gamma$ ) obtains family of possible transformation curves  $S = C * r^\gamma$ .

Here  $C$  and  $\gamma$  are positive constants. Plot of  $S$  versus  $r$  for various values of  $\gamma$  is  $\gamma > 1$  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.

### CORRECTING GAMMA:

$$s = cr^\gamma \quad s = cr \quad (1/2.5)$$

The same image but with different gamma values has been shown here.

### Piecewise-Linear Transformation Functions:

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 over the types of functions we have discussed thus far is that the form of piecewise functions can be arbitrarily complex.

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.

$$S = T(r)$$

Figure x(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, as illustrated in fig. 2.2(b).

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 that the function is single valued and monotonically increasing

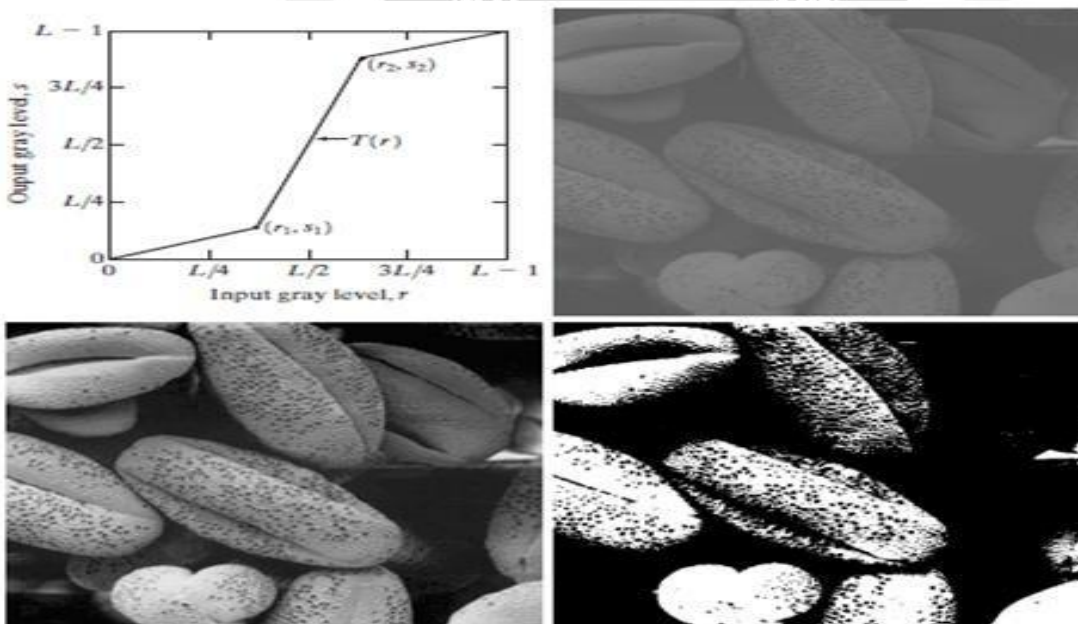


Fig. 2.1.6 Contrast stretching. (a) Form of transformation function. (b) A low-contrast stretching. (c) Result of contrast stretching. (d) Result of thresholding (original image courtesy of Dr.Roger Heady, Research School of Biological Sciences, Australian National University Canberra Australia.

Figure x(b) shows an 8-bit image with low contrast. Fig. x(c) 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, Fig. x(d) shows the result of using the thresholding function defined previously, with  $r_1=r_2=m$ , the mean gray level in the image. The original image on which these results are based is a scanning electron microscope image of pollen, magnified approximately 700 times.

Source: Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition, 2010. – Page 110

