Image Enhancement

Image enhancement approaches fall into two broad categories: spatial domain methods and frequency 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.

Frequency domain processing techniques are based on modifying the Fourier transform of an image. Enhancing an image provides better contrast and a more detailed image as compare to non enhanced image. Image enhancement has very good applications. It is used to enhance medical images, images captured in remote sensing, images from satellite e.t.c. 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 theexpression.

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

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 principal approach in defining a neighborhood about a point (x, y) is to use a square or rectangular sub image area centered at (x, y), as Fig. 2.1 shows. The center of the sub image 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.

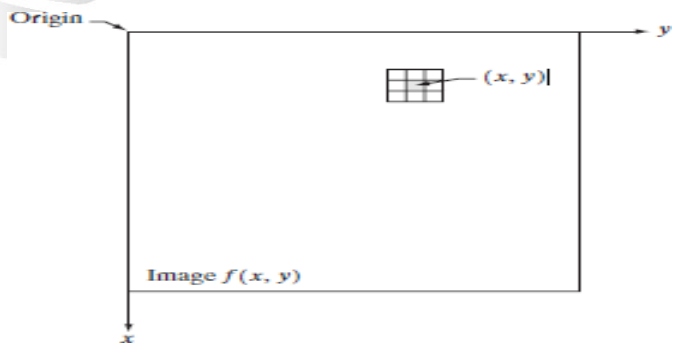

Fig2.5.1: 3x3 neighborhood about a point (x,y) in an image.

Source: Rafael C. Gonzalez, Richard E. Woods, ‗Digital Image Processing', Pearson,Third Edition,
2010.- Page- 105

The simplest form of T is when the neighborhood is of size 1*1 (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 and s is the pixels of the output image. T is a transformation function that maps each value of „r" to each value of „s".

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. In this technique, known as contrast stretching, the values of r below m are compressed by the transformation function into a narrow range of s, toward black.The opposite effect takes place for values of r above m.

In the limiting case shown in Fig. 2.2(b), T(r) produces a two-level (binary)image. A mapping of this form is called a thresholding function.

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.

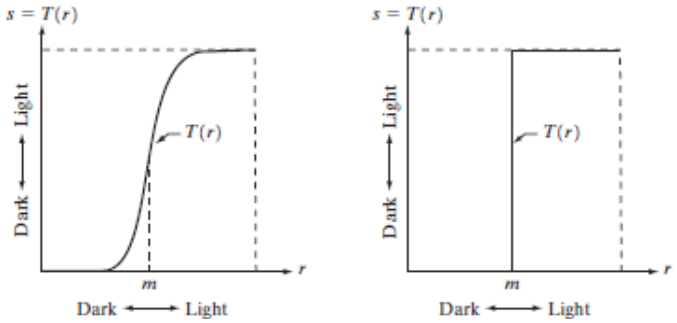

Fig. 2.5.2 Gray level transformation functions for contrast enhancement.

(Source: Rafael C. Gonzalez, Richard E. Woods, ‗Digital Image Processing', Pearson, Third Edition, 2010.-Page 106)

GRAY-LEVEL SLICING:

Highlighting a specific range of gray levels in an image often is desired. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images.

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 graylevels.

This transformation, shown in Fig. y(a), produces a binary image. The second approach, based on the transformation shown in Fig.y (b), brightens the desired range of gray levels but preserves the background and gray-level tonalities in the image. Figure y (c) shows a gray-scale image, and Fig. y(d) shows the result of using the transformation in Fig. y(a).Variations of the two transformations shown in Fig. are easy to formulate.

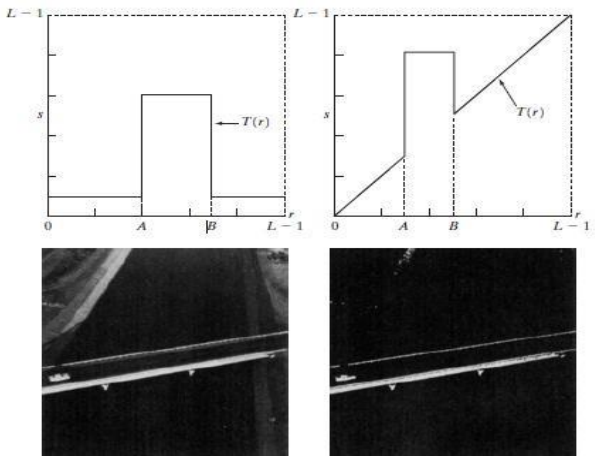

Fig.2.5.3y (a)This transformation highlights range [A,B] of gray levels and reduces all others to a constant level (b) This transformation highlights range [A,B] but preserves all other levels.

(c) An image . (d) Result of using the transformation in(a).

(Source: Rafael C. Gonzalez, Richard E. Woods, ‗Digital Image Processing', Pearson, Third Edition, 2010.-Page:107)

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 significantbit. Intermsof 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 3.12 illustrates these ideas, and Fig. 3.14 shows the various bit planes for the image shown in Fig. 3.13. 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.

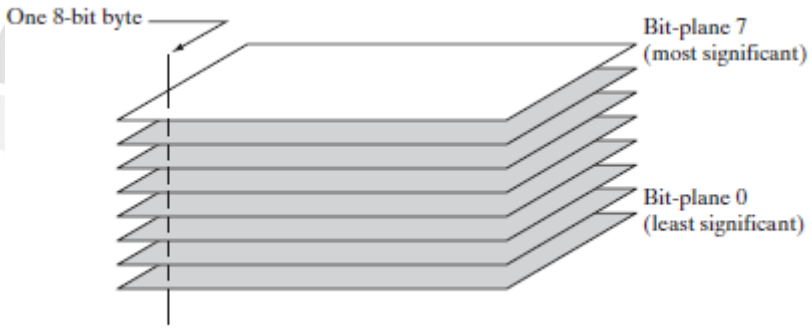

FIGURE 2.5.4 Bit-plane representation of an 8-bit image.
(Source: Rafael C. Gonzalez, Richard E. Woods, ‗Digital Image Processing', Pearson, Third Edition, 2010.-Page-118)

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 o ne level (for example, 0); and (2) maps all levels between 129 and 255 to another (for example, 255).The binary image for bit-plane 7 in Fig. 3.14 was obtained in just this manner. It is left as an exercise (Problem 3.3) to obtain the gray-level transformation functions that would yield the other bit planes.