UNIT - II

Syllabus: Image enhancement in the spatial domain

Introduction, Basic gray -level transformations, histogram processing, enhancement using arithmetic and logic operators, Basics of spatial filtering, smoothing and sharpening spatial filters, combining the spatial enhancement methods.

2.1 Introduction:

Spatial Domain and Point Processing:

- 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). In addition, T can operate on a set of input images, such as performing the pixel -by- pixel sum of K images for noise reduction.
- 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.

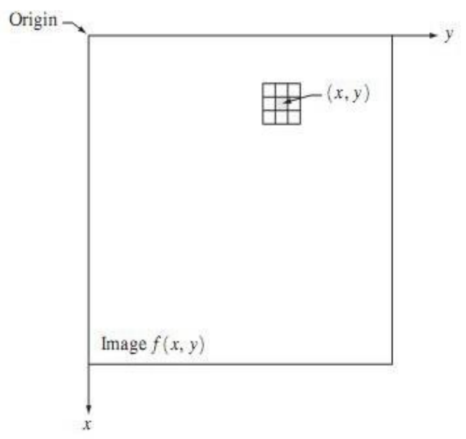


Fig.2.1 : A 3*3 neighborhood about a point (x, y) in an image.

- The simplest form of T is when the neighborhood is of size 1*1 (that is, asingle 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, for simplicity in notation, r and s are variables denoting, , the gray level of f(x, y) and g(x, y) respectively at anypoint (x, y).

- For example, if T(r) has the for m shown in Fig. 2.2(a), the effect of this transformation would be to produce an image of higher contrast than theoriginal 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**.
- Some fairly simple, yet powerful, processing approaches can be formulated with gray - 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** .

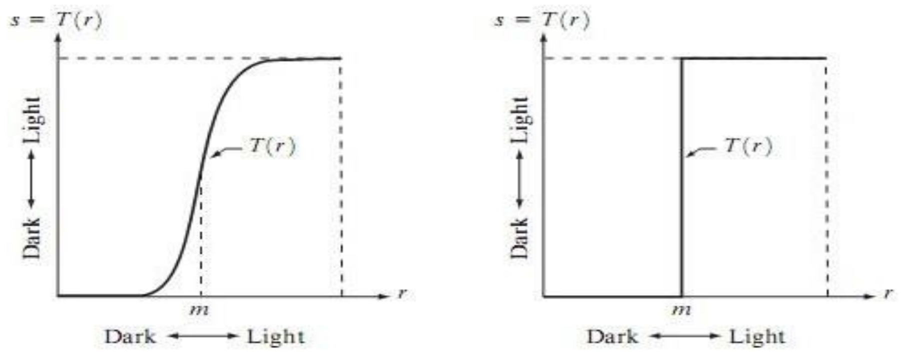


Fig.2.2 Gray level transformation functions for contrast enhancement.

- Larger neighborhoods allow considerably more flexibility. The general approach is to use a function of the values of f in a predefinedneighborhood 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 -calledmasks (also referred to as filters, kernels, templates, or windows). Basically, a mas k is a small (say, 3*3) 2 -D array, such as the one shownin Fig. 2.1, in which the values of the mask coefficients determine the nature of the process, such as image sharpening.

2.2 Basic Gray Level Transformations

- The image enhancement technique is done by gray-level transformation functions. The values of pixels, before and after processing, will be denoted by **r** and **s**, respectively. These values are related by an expression of the form **s=T(r)** , where **T** is a transformation that maps a pixel value **r** into a pixel value **s**.
- Since we are dealing with digital quantities, values of the transformation function typically are stored in a one -dimensional array and the mappings from r to s are implemented via table lookups. For an 8 -bit environment, a lookup table containing the values of T will have 256 entries.

- As an introduction to gray -level transformations, consider Fig. 1.1, which shows **three** basic types of functions used frequently for image enhancement:
    1. linear (negative and identity transformations),
    2. logarithmic (log and inverse-log transformations), and
    3. power-law (nth power and nth root transformations).
- The identity function is the trivial case in which output intensities areidentical to input intensities.

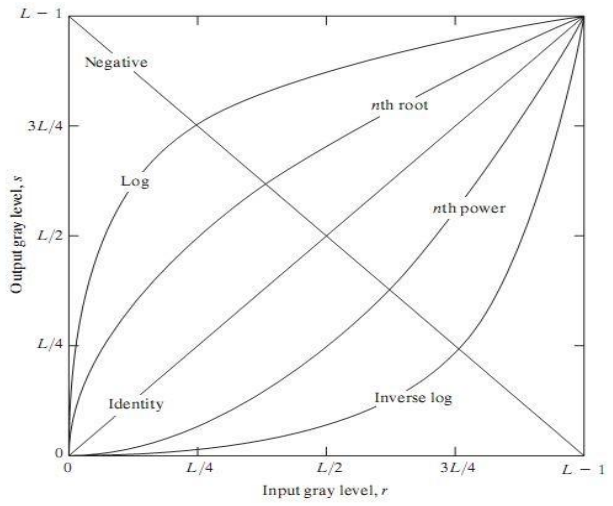


Fig 2.1: Some basic gray -level transformation functions used forimage enhancement

2.2.1 Image Negatives:

- The negative of an image with gray levels in the range [0, L -1] is obtainedby using the negative transformation shown in Fig.1.1, which is given by the expression

$$s = L - 1 - r.$$

- Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative. This type of processing is particularly suited for enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size.

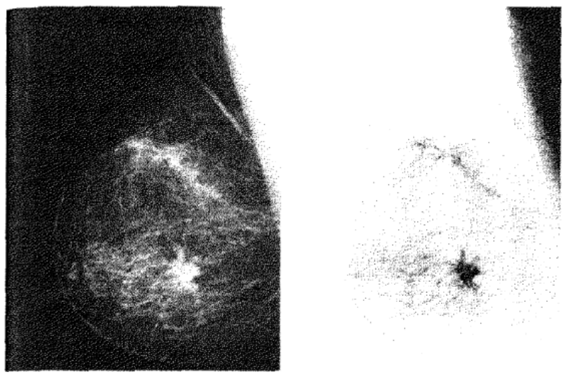


Fig 2.2.1:(a) Original digital mammogram (b) Negative imageobtained using the negative transformation

2.2.2 Log Transformations:

- The general form of the log transformation shown in Fig.1.1 is

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

- Where c is a constant, and it is assumed that r ≥ 0.
- The shape of the log curve in Fig. 2.1 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 transform ation of this type to expand the values of dark pixels in an image while compressing the higher -level values. The opposite is true of the inverse log transformation.

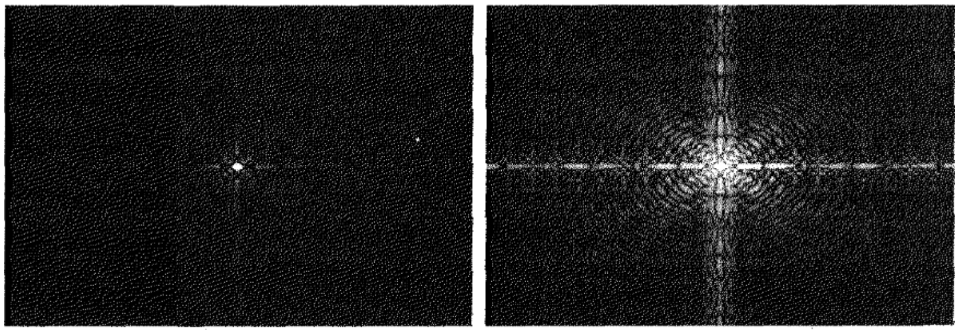


Fig 2.2.2: (a) Fourier spectrum (b) Result of applying the log transformation with c=1

- In the above figure 2.2.2(a) shows a Fourier spectrum with values in the range 0 to 1.5 X 10^6. When these values are scales linearly for display in an 8 bit system, the bright pixels will dominate the display. But if we apply the log transformation with C= 1 to the spectrum values then the range of values of the spectrum becomes 0 to 6.2, then the result scaling image will be as shown in the figure 2.2.2(b).

2.2.3 Power-Law Transformations:

- Power-law transformations have the basic form $s = cr^{\gamma}$ where c and γ are positive constants.
- Sometimes Eq. is written as $s = c(r + \varepsilon)^{\gamma}$ to account for an offset(that is, a measurable output when the input is zero) However, offsetstypically are an issue of display calibration and as a result they are normally ignored in Eq. Plots of s versus r for various values of g areshown in Fig. 2.2.3.

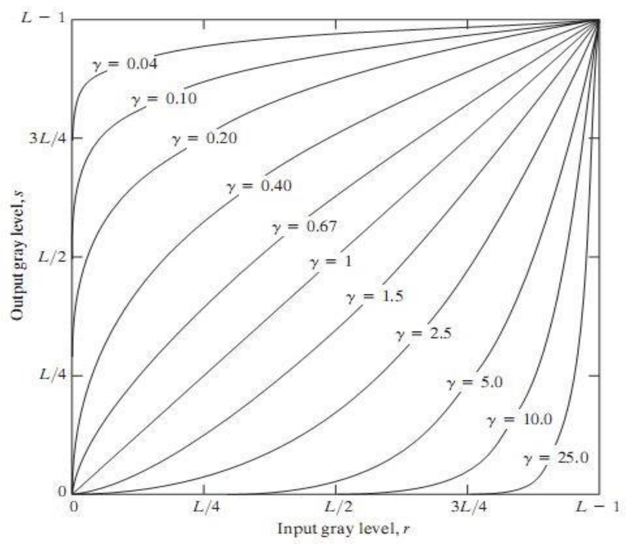


Fig. 2.2.3: Plots of the equation $s = cr^{\gamma}$ for various γ values of (c=1in all cases).

- We see in Fig. 2.2.3 that curves generated with values of γ >1 have exactly the opposite effect as those generated with values of γ <1.Finally, we note that Eq. reduces to the identity transformation when c = γ = 1. A variety of devices used for image capture, printing, and

  display respond according to a power l aw. By convention, the exponent in the power -law equation is referred to as gamma. The process used to correct this power -law response phenomenon 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.2.2.3, we see that such display systems would tend to produce images that are darker than intended.

2.2.4 Piecewise -Linear Transformation Functions:

- The principal advantage of piecewise linear functions over the types of functions we have discussed above is that the form of piecewise functions can be arbitrarily complex. The principal disadvantage of piecewise functions is that their specification requires considerably moreuser input.

a) Contrast stretching:

- 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 the gray levels in the image being processed.

- Figure 2.2.4.1 (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 illustrated in Fig. 2.2.4.1 (b). 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 that the function is single valued and monotonically increasing. This condition preserves the order of gray levels, thus preventing the creation of intensity artifacts in the processedimage.

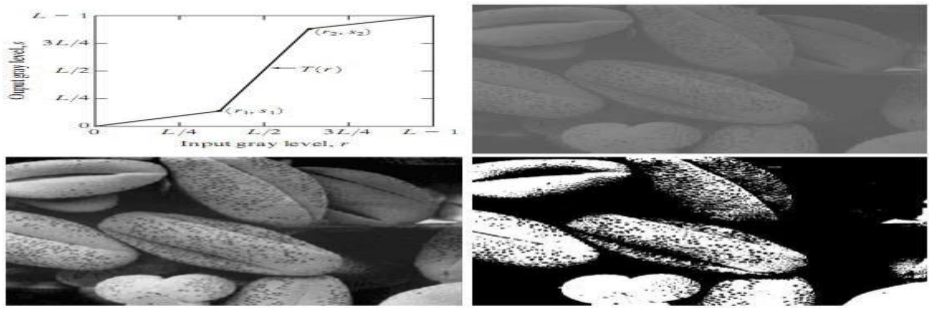


Fig 2.2.4.1: Contrast Stretching (a) Form of Transformation function (b) Alow contrast image (c) Result of contrast stretching (d) Result of thresholding.

- Figure 2.2.4.1 (b) shows an 8-bit image with low contrast. Fig. 2.2.4.1(c) shows the result of contrast stretching, obtained by setting (r1 , s1) = (rmin , 0) and (r2 , s2) = (rmax , L -1) where rmin and rmax de note theminimum 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. 2.2.4 (d) shows the result ofusing the thresholding function defined previously, with r1 = r2 = 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.

b) Gray-level slicing:

- 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. This transformation, shown in Fig. 2.2.4.2 (a), produces a binary image. The second approach, based on the transformation shown in Fig. 2.2.4.2 (b), brightens the desired range of gray levels but preserves the background and gray -level tonalities in the image. Figure 1.4(c) shows a gray-scale image, and Fig. 2.2.4.2 (d) shows the result of using the transformation in Fig. 2.2.4.2 (a).Variations of the two transformations shown in Fig. 2.2.4.2 are easy to formulate.

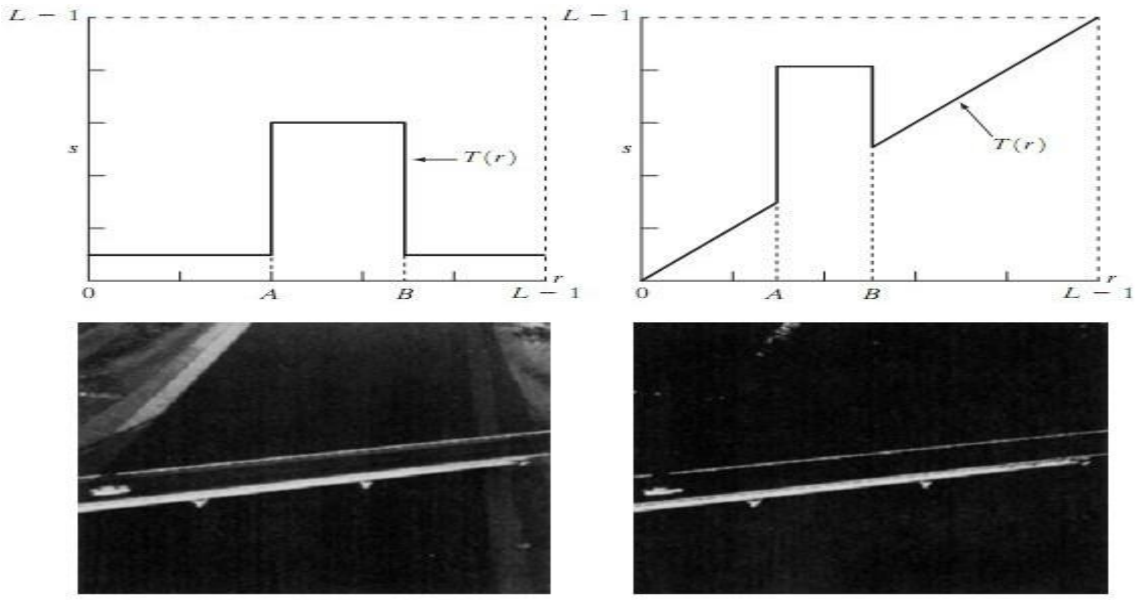


Fig. 2.2.4.2 (a) This transformation highlights range [A, B] of gray levels and reduce all others to a constant level (b) This transformation highlights range [A, B] but preserves all other levels (c) An image (d) Resultof using the transformation in (a).

c)Bit -plane slicing:

- Instead of highlighting gray -level ranges, highlighting the con tribution 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 b it 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 2.2.4.3 illustrates these ideas, and Fig. b shows the various bit planes for the image shown in Fig a . 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 di gital 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 eachpixel.

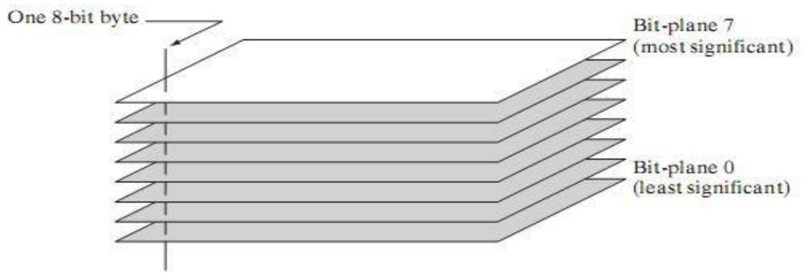
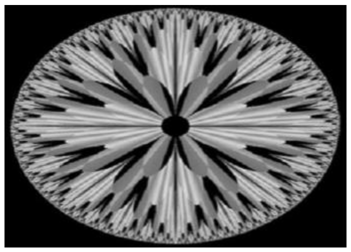


Fig. 2.2.4.3 Bit -plane representation of a n 8 -bit image.

- 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 l evels 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).

Fig. a: An 8 -bit fractal image



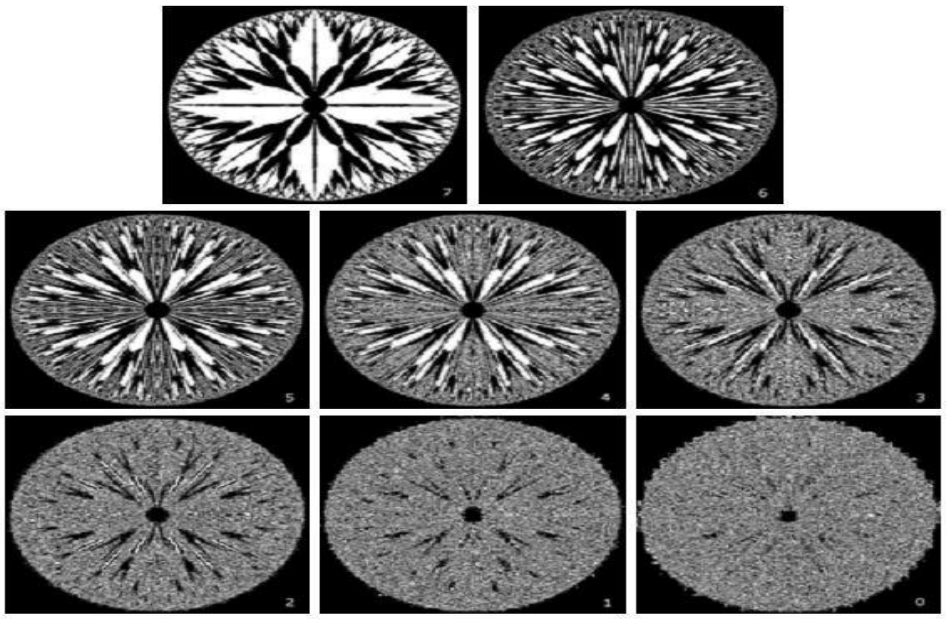


Fig b: The eight bit planes of the image in Fig.1.6. The number at thebottom, right of each image identifies the bit plane.

2.3 <u>Histogram Processing</u>

- The histogram of a digital image with gray levels in the range [0, L -1] is a discrete function h(r k) = (nk), where rk is the kth gray level and n k is the number of pixels in the image having gray level rk. It is common practice to normalize a histogram by dividing each of its values by the total number of pixels in the image, denoted by n. Thus, a normalized histogram is given by

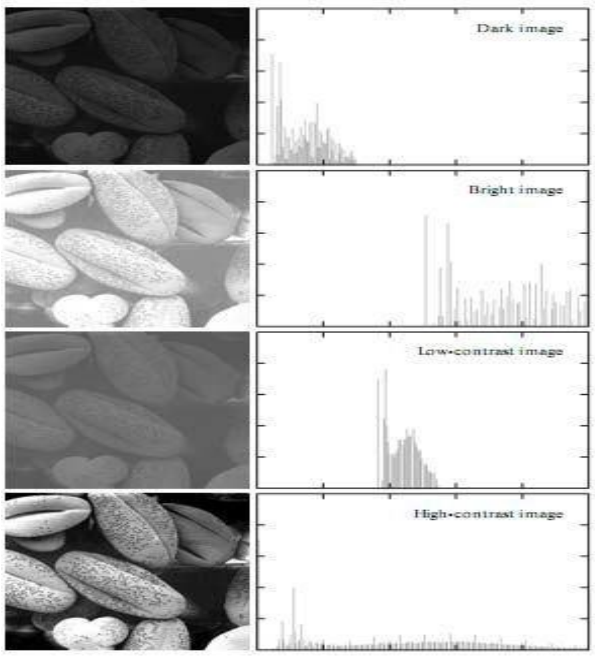
  for k=0,1,…… .,L-1. Loosely speaking, p(r k) gives an estimate of the probability of occurrence of gray level r k. Note that the sum of all components of a normalized histogram

  $$p(r_k) = n_k/n$$

  is equal to 1.
- Histogram manipulation can be used effectively for image enhancem ent for real-time image processing.
- Consider Fig. 3, whi ch is the pollen image shown in four basic gray -level characteristics: dark, light, low contrast, and high contrast. The right

side of the figure shows the histograms corresponding to these images. The horizontal axis of each histogram plot corresponds to gray level values, r_k. The vertical axis corresponds to values of $h(r_k) = n_k$ or $p(r_k) = n_k/n$ if the values are normalized. Thus, as indicated previously, these histogram plots are simply plots of $h(r_k) = n_k$ versus r_k or $p(r_k) = n_k/n$ versus r_k.

Fig 2.3 : Four basic image types: dark, light, low contrast, highcontrast, and



their corresponding histograms.

- We note in the dark image that the components of the histogram are concentrated on the low (dark) side of the gray scale. Similarly, the components of the histogram of the bright image are biased toward the high side of the gray scale. An image with low contrast has a histogram that will be narrow and will be centered toward the middle of the gray scale. For a monochrome image this implies a dull, washed - out gray look.
- Finally, we see that the components of the histogram in the high -contrastimage cover a broad range of the gray scale and, further, that the

distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others. Intuitively, it is reasonable to conclude that an image, whose pixels tend to occupy the entire range of possible gray levels and, in addition, tend to be distributed uniformly, will have an appearance of high contrast and will exhibit a larg e varietyof gray tones.

Histogram Equalization:

- Consider for a moment continuous functions, and let the variable r represent the gray levels of the image to be enhanced. We assume that r has been normalized to the interval [0, 1], with r=0 representing bl ack and r=1 representing white. Later, we consider a discrete formulation and allow pixel values to be in the interval [0, L -1]. For any r satisfying the aforementioned conditions, we focus attention on transformations of the form

  that produce a level s for every pixel value r in the original image. For reasons that will become obvious shortly, we assume that the transformation function T(r) satisfies the following conditions:

$$s = T(r) \qquad 0 \le r \le 1$$

  a. T(r) is single-valued and monotonically increasing in the interval $0 \le r \le 1$; and

  b. $0 \le T(r) \le 1$ for $0 \le r \le 1$.

- The requirement in (a) that T(r) be single valued is needed to guarantee that the inverse transformation will exist, and the monotonicity conditio npreserves the increasing order from black to white in the output image. Atransformation function that is not monotonically increasing could result in at least a section of the intensity range being inverted, thus producing some inverted gray levels in the output image. Finally, condition (b)guarantees that the output gray levels will be in the same range as the input levels.

- Figure 2.3.1 gives an example of a transformation function that satisfies these two conditions. The inverse transformation from s back to r is denoted

- It can be shown by example that even if T(r) satisfies conditions (a) and (b), it is possible

$$r = T^{-1}(s) \qquad 0 \leq s \leq 1.$$

that the corresponding inverse T -1 (s) may fail to be single valued.

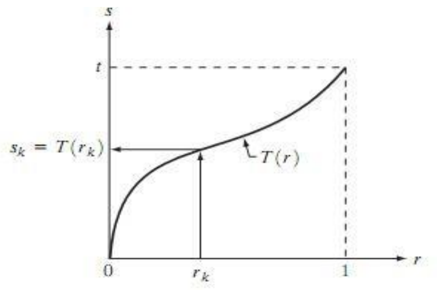


Fig. 2.3.1 A gray-level transformation function that is both single valued and monotonically increasing.

- The gray levels in an image may be viewed as random variables in the interval [0, 1].One of the most fundamental descriptors of a random variable is its probability density function (PDF).Let p r(r) and ps(s) denotethe probability density functions of random variables r and s, respectively, where the subscripts on p are used to denote that pr and ps are different functions. A basic result from an elementary probability theory is that, if pr(r) and T(r) are known and T^{-1} (s) satisfies condition (a),then the probability density function p s(s) of the transformed variable scan be obtained using a rather simple formula:
- Thus, the probability density function of the transformed variable, s, is determined by the gray level PDF of the input image and by the chosen

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

transformation function. A transformation function of particular importance in image processing has the form

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

- Where w is a dummy variable of integration. The right side of Eq. above is recognized as the cumulative distribution function (CDF) of random variable r. Since probability density functions are always positive, and recalling that the integral of a function is the area under the function, it follows that this transformation function is single valued and monotonically increasing, and, therefore, satisfies condition (a). Similarly, the integral of a probability density function for variables in the range [0, 1] also is in the range [0, 1], so condition (b) is satisfied as well. Substituting this result for dr/ds, and keeping in mind that all probability values are positive, yields

$$\begin{aligned} p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| \\ &= p_r(r)\left|\frac{1}{p_r(r)}\right| \\ &= 1 \qquad 0 \le s \le 1. \end{aligned}$$

- Because ps(s) is a probability density function, it follows that it must be zero outside the interval [0, 1] in this case because its integral over all values of s must equal 1.We reco gnize the form of ps(s) as a uniform probability density function. Simply stated, we have demonstrated that performing the transformation function yields a random variable s characterized by a uniform probability density function. It is important to note from Eq. discussed above that T(r) depends on pr(r), but, as indicated by Eq. after it, the resulting ps(s) always is uniform, independent of the form of pr(r). For discrete values we deal with probabilities and summations instead of probability density fun ctions and integrals. The probability of occurrence of gray level r in an image isapproximated by

- Where, n is the total number of pixels in the image, nk is the number of pixels that have

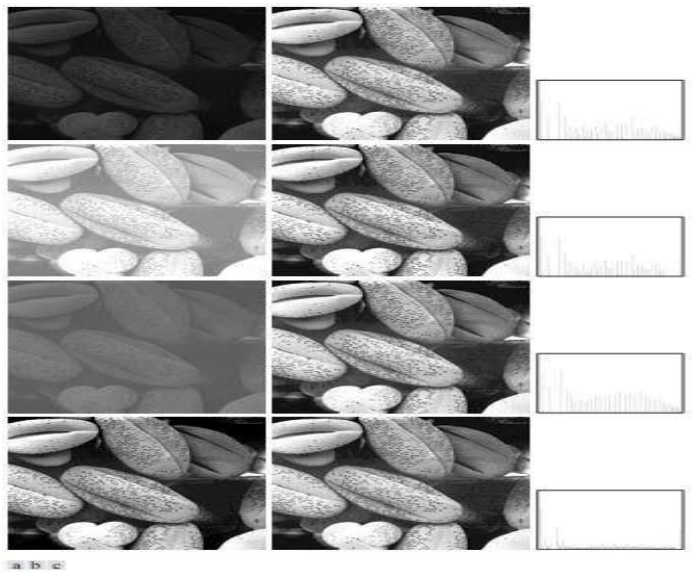
$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \ldots, L-1$$

gray level rk, and L is the total number of possible gray levels in the image. The discrete version of the transformation function given in Eq. is

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0, 1, 2, \ldots, L-1.$$

- Thus, a processed (output) image is obtained by mapping each pixel withlevel rk in the input image into a corresponding pixel with level sk in the output image. As indicat ed earlier, a plot of pr (rk) versus rk is called a histogram. The transformation (mapping) is called histogram equalizationor histogram linearization.

Fig.2.3.2 (a) Images from Fig.3 (b) Results of histogram equalization. (c)Corresponding histograms.



- The inverse transformation from s back to r is denoted by

$$r_k = T^{-1}(s_k) \qquad k = 0, 1, 2, \dots, L - 1$$

Histogram Matching (Specification):

- In particular, it is useful sometimes to be able to specify the shape of the histogram that we wish the processed image to have. The method used togenerate a processed image that has a specified histogram is called histogram matching or histogram specification.

Development of the method:

- In this notation, r and z denote the gray levels of the input and output (processed) images, respectively. We can estimate p r(r) from the given input image, while pz(z) is the specified probability density function that we wish the output image to have. Let s be a random variable with the property
- Where w is a dummy variable of integration. We recognize this expression as the continuo's version of histogram equalization. Suppose next that we define a random

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

variable z with the property

$$G(z) = \int_0^z p_z(t)\,dt = s$$

- Where t is a dummy variable of integration. It then follows from these twoequations that G(z)=T(r) and, therefore, that z must satisfy the condition

$$z = G^{-1}(s) = G^{-1}[T(r)].$$

- The transformation T(r) can be obtained once pr(r) has been estimated from the input image. Similarly, the transformation function G(z) can be obtained because pz(z) is given. Assuming that G-1 exists and that it satisfies conditions (a) and (b) i n the histogram equalization process, the above three equations show that an image with a specified probability density function can be obtained from an input image by using thefollowing procedure:

1. Obtain the transformation function T(r).
2. To obtain the transformation function G(z).
3. Obtain the inverse transformation function G -1
4. Obtain the output image by applying above Eq. to all the pixels inthe input image.

- The result of this procedure will be an image whose gray levels, z, have the specified probability density function p z(z). Although the procedure described is straightforward in principle, it is possible in practice to obtain analytical expressions for T(r) and for G $^{-1}$.
- Where n is the total number of pixels in the image, nj is the number of pixels with gray

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0, 1, 2, \dots, L-1$$

level rj, and L is the number of discrete gray levels. Similarly, the discrete formulation is obtained from the given histogram pz(zi), i=0, 1, 2,……, L-1, and has the form

$$v_k = G(z_k) = \sum_{i=0}^{k} p_z(z_i) = s_k \qquad k = 0, 1, 2, \dots, L-1.$$

- As in the continuo’s case, we are seeking values of z that satisfy this equation. The variable vk was added here for clarity in the discussion that follows. Finally, the discrete version of the above Eqn. is given by

$$z_k = G^{-1}[T(r_k)] \qquad k = 0, 1, 2, \dots, L-1$$

Or

$$z_k = G^{-1}(s_k) \qquad k = 0, 1, 2, \dots, L-1.$$

2.4 4 Enhancement using arithmetic and logic operators:

a)Image Subtraction:

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

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

  is obtained by computing the difference between all pairs of corresponding pixels from f and h.
- The use of subtraction is the enhancement of differences between images. The higher order bit planes of an image carry a significant amount of visually relevant detail, while the lower planes contribute moreto fine (often imperceptible) detail.
- Figure (a) shows the fractal image used earlier to illustrate the concept of bit planes. Figure (b) shows the result of discarding (setting to zero) the four least significant bit planes of the original image. The images are nearly identical visually, with the exception of a very slight drop inoverall contrast due to less variability of the gray level values in theimage of Fig. (b).The pixel-by-pixel difference between these two images is shown in Fig. (c).The differences in pixel values are so small that the difference image appears nearly black when displayed on an 8 -bitdisplay. In order to bring out more detail, we can perform a contraststretching transformation. We chose histogram equalization, but an appropriate power-law transformation would have done the job also. Theresult is shown in Fig. (d). This is a very useful image for evaluating the effect of setting to zero the lower-order planes.

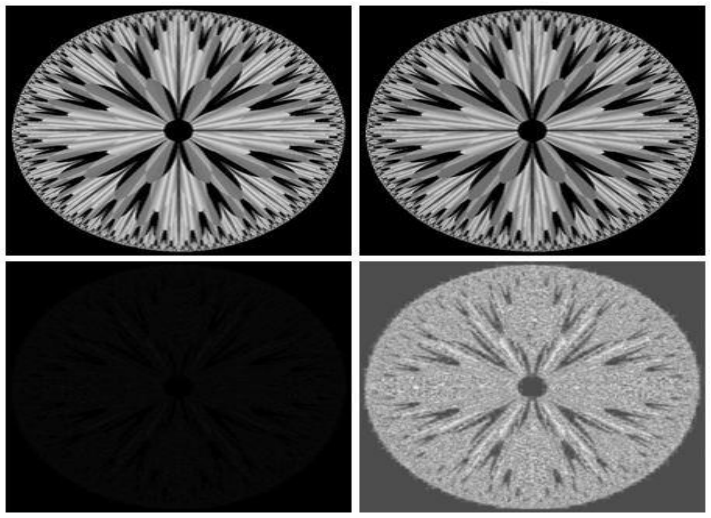


Fig :(a) Original fractal image (b) Result of setting the four lower -order bit planes to zero

(c) Difference between (a) and (b) (d) Histogram equalized difference image.

- The use of image subtraction is in the area of medical imaging called mask mode radiography. In this case h(x, y), the mask, is an X -ray imageof a region of a patient's body captured by an intensified TV camera (instead of traditional X -ray film) located opposite an X -ray source. Theprocedure consists of injecting a contrast medium into the patient's bloodstream, taking a series of images of the same anatomical region as h(x, y), and subtracting this mask from the series of incoming images after injection of the contrast medium. The net effect of subtracting the mask from each sample in the incoming stream of TV images is that the areas that are different between f(x, y) and h(x, y) appear in the output image as enhanced detail. Because images can be captured at TV rates, this procedure in essence gives a movie showing how the contrastmedium propagates through the various arteries in the area being observed.

b) Image averaging:

- Consider a noisy image g(x, y) formed by the addition of noise h(x, y) to an original image f(x, y); that is,

$$g(x, y) = f(x, y) + \eta(x, y)$$

- Where the assumption is that at every pair of coordinates (x, y) the noiseis uncorrelated and has zero average value. The objective of the following procedure is to reduce the noise content by adding a set of noisy images,
  {gi (x, y)}. If the noise satisfies the constraints just stated, it can be shown that if an image

$$\bar{g}(x, y)$$

is formed by averaging K different noisy images,

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

Then it follows that

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

And

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$$

- Where $E\{\bar{g}(x, y)\}$ is the expected value of $\bar{g}$ and $\sigma^2_{\bar{g}(x,y)}$ and $\sigma^2_{\bar{\eta}(x,y)}$ are the variances of $\bar{g}$ and η, all at coordinates (x, y). The standard deviation at any point in the average image is

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)}.$$

- As K increases, the above equations indicate that the variability (noise) of the pixel values at each location (x, y) decreases. Because $E\{\bar{g}(x, y)\} = f(x, y),$ this means thatapproaches f(x,y) $\bar{g}(x, y)$ as the number of noisy images used in the averaging process increases. In practice, the images g i(x, y) must be registered (aligned) in order to avoid the introduction of blurring and other artifacts in the output image.

2.5 5 Basics of Spati al Filtering

- Some neighborhood operations work with the values of the image pixels in the neighborhood and the corresponding values of a sub image that has the same dimensions as the neighborhood. The sub image is called afilter, mask, kernel, template, or window.
- The values in a filter sub image are refer red to as coefficients, rather than pixels. The concept of filtering has its roots in the use of the Fouriertransform for signal processing in the so -called frequency domain. We use the term spatial filtering to differentiate this type of process from themore traditional frequency domain filtering.
- The mechanics of spatial filtering are illustrated in Fig.9.1. 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 poi nt is calculated using a predefined relationship. The response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask. For the 3 x 3 mask shown in Fig. 9.1, the result (or response), R, of linear filtering with the filter mask at apoint (x, y) in the image is
- Which we see is the sum of products of the mask coefficients with the corresponding pixels directly under the mask. Note in particular that the coefficient w(0, 0) coincides with image value f(x, y), indicating that the mask is centered at (x, y) when the

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

computation of the sum of products

takes place. For a mask of size m x n, we assume that m=2a+1 andn=2b+1,where a and b are nonnegative integers.

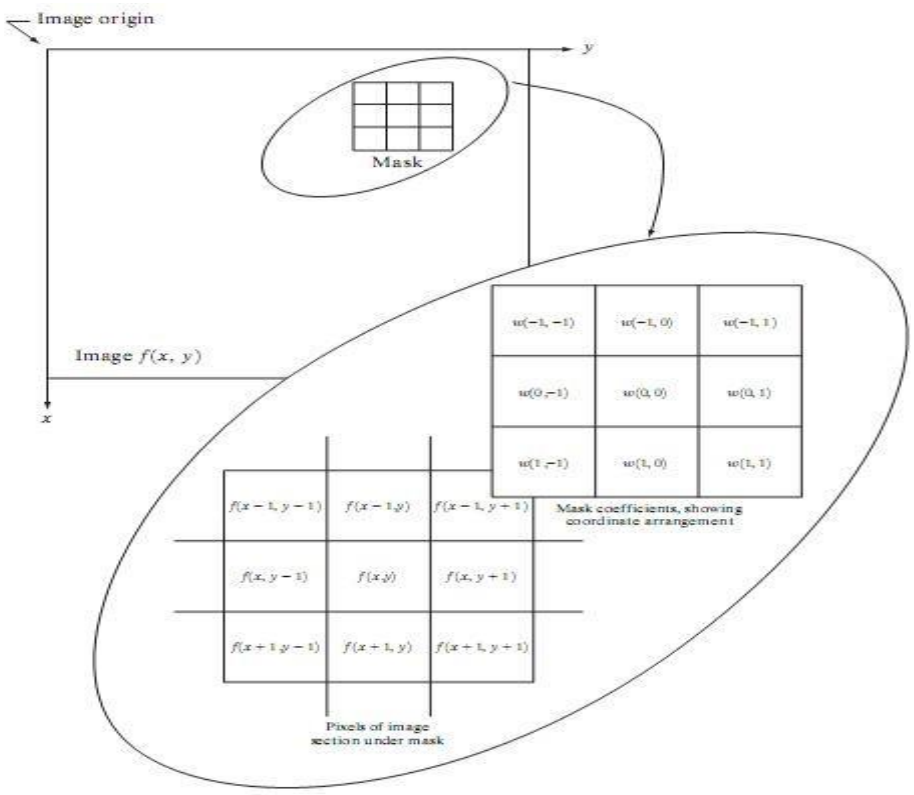


Fig 2 .5 : The mechanics of spatial filtering. The magnified drawing showsa 3X3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

- In general, linear filtering of an image f of size M x N with a filt er mask ofsize m x n is given by the expression:

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

- Where, from the previous paragraph, a=(m -1)/2 and b=(n-1)/2. To generate a complete filtered image this equation must be applied for x=0,1,2,……, M-1 and y=0,1,2,……, N-1. In this way, we are assured thatthe mask processes all pixels in the image. It is easily verified when m=n=3 that this expression reduces to the example given in the previous paragraph.

- The process of linear filtering is similar to a frequency domain concept called convolution. For this r eason, linear spatial filtering often isreferred to as “convolving a mask with an image.” Similarly, filter masks are sometimes called convolution masks. The term convolution kernel also is in common use. When interest lies on the response, R, of an m xn mask at any point (x, y), and not on the mechanics of implementing mask convolution, it is common practice to simplify the notation by using the following expression:
- where the w’s are mask coefficients, the z’s are the values of the image gray levels corresponding to those coefficients, and mn is the total number of coefficients in the mask. For the 3 x 3 general mask shown in Fig.9.2 the response at any point (x, y) in the

$$\begin{aligned} R &= w_1 z_1 + w_2 z_2 + \dots + w_{mn} z_{mn} \\ &= \sum_{i=1}^{mn} w_i z_i \end{aligned}$$

image is given by

$$\begin{aligned} R &= w_1 z_1 + w_2 z_2 + \dots w_9 z_9 \\ &= \sum_{i=1}^{9} w_i z_i . \end{aligned}$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

- An important consideration in implementing neighborhood op erations for spatial filtering is the issue of what happens when the center of the filter approaches the border of the image. Consider for simplicity a square mask of size n x n. At least one edge of such a mask will coincide with the border of the image when the center of the mask is at a distance of (n-1)/2 pixels away from the border of the image. If the center of the mask moves any closer to the border, one or more rows or columns of the mask will be located outside the image plane. There are several ways to handle this situation. The simplest is to limit the excursions of the center of the mask to be at a distance no less than (n-1)/2 pixels from the border. The resulting filtered image will be smaller than the original, but all the pixels in the filtered imaged will have been processed with the full mask. If the result is required to be the same size as the original, then the approach typically employed is to filter all pixels only with the section of the mask that is fully contained in the image. With this approach, there will be bands of pixels near the border that will have been processed with a partial filter mask. Other approaches include "padding" the image by adding rows and columns of 0's (or other constant gray level), or padding by replicating rows or columns. The padding is then stripped off at the end of the process.

- This keeps the size of the filtered image the same as the original, but the values of the padding will have an effect near the edges that becomes more prevalent as the size of the mask increases. The only way to obtain a perfectly filtered result is to accept a somewhat smaller filtered image by limiting the excursions of the center of the filter mask to a distance noless than (n-1)/2 pixels from the border of the original image.

2.6 <u>Smoothing and sharpening Spatial Filters</u>

- Smoothing filters are used for blurring and for noise reduction. Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves. Noise reduction can be accomplished by blurring with a linear filter and also by non -linear filtering.

(1) Smoothing Linear Filters:

- The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called averaging filters. The idea behind smoothing filters is straightforward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask, this process results in an image with reduced "sharp" transitions in gray levels. Because random noise typically consists of sharp transitions in gray levels, the most obvious application of smoothing is noise reduction. However, edges (which almost always are desirable features of an image) also are characterized by sharp transitions in gray levels, so averaging filters have the undesirable side effect that they blur edges. Another application of this type of process includes the smoothing of false contours that result from using an insufficient number of gray levels.

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}\qquad \frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

Fig 2.6.1: Two 3 x 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

- A major use of averaging filters is in the reduction of “irrelevant” detail inan image. By “irrelevant” we mean pixel regions that are small with respect to the size of the filter mask.
- Figure 2.6.1shows two 3 x 3 smoothing filters. Use of the first filter yieldsthe standard average of the pixels under the mask. This can best be seen by substituting the coefficients of the mask in

$$R=\frac{1}{9}\sum_{i=1}^{9} z_i,$$

- which is the average of the gray levels of the pixels in the 3 x 3 neighborhood defined by the mask. Note that, instead of being 1/9, the coefficients of the filter are all 1’s.The idea here is that it is computationally more efficient to have coefficients valued 1. At the end of the filtering process the entire image is divided by 9. An m x n mask would have a normalizing constant equal to 1/mn.
- A spatial averaging filter in which all coefficients are equal is sometimes called a box filter.
- The second mask shown in Fig. 2.6.1 yields weighted average, terminology used to indicate that pixels a re multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of others.
- In the mask shown in Fig. 2.6.1(b) the pixel at the center of the mask ismultiplied by a higher value than any other, thus giving this pixel more importance in the calculation of the average. The other pixels are inversely weighted as a function of their distance from the center of the mask. The diagonal terms are further away from the center than the orthogonal neighbors (by a factor of √2) and, thus, are weighed less thanthese immediate neighbors of the center pixel.
- The basic strategy behind weighing the center point the highest and thenreducing the value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process. We could have picked other weights to accomplish the same

general objective. However, the sum of all the coefficients in the mask of Fig. 2.6.1(b) is equal to 16, an attractive feature for computer implementation because it has an integer power of 2. In practice, it is difficult in general to see differences between images smoothed by usingeither of the masks in Fig. 10.1, or similar arrangements, because the area these masks span at any one locat ion in an image is so small.

- The general implementation for filtering an M x N image with a weighted averaging filter of size m x n (m and n odd) is given by the expression

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

(2) Order -Statistics Filters:

- Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result. The best -knownexample in this category is the median filter, which, as its name implies, replaces the value of a pixel by the median of the gray levels in theneighborhood of that pixel (the original value of the pixel is included in the computation of the median). Median filters are quite popular because, for certain types of random noise, they provide excellent noise – reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective inthe presence of impulse noise, also called salt-and-pepper noise becauseof its appearance as white and black dots superimposed on an image.
- The median, ε, of a set of values is such that half the values in the setare less than or equal to ε, and half are greater than or equal to ε. In order to perform median filtering at a point in an image, we first sort the values of the pixel in question and its neighbors, determine their median,and assign this value to that pixel. For example, in a 3 x 3 neighborhood the median is the 5th largest value, in a 5 x 5 neighborhood the 13th largest value, and so on. When several values in a neighborhood are the same, all equal values are grouped. For example, suppose that a 3 x 3 neighborhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100). These values are sorted as (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in a median of 20. Thus, the principal function of median filters is to force points with distinct gray levels to be more like their neighbors. In fact, isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2 / 2$ (one-half the filter area), are eliminated by an n x n median filter. In this case “eliminated” means forced to the median intensity of the neighbors. Larger clusters are affected considerably less.

(3) Use of Second Derivatives for Enhancement –The Laplacian:

- The approach basically consists of defining a discrete formulation of the second-order derivative and then constructing a filter mask based onthat formulation. We are inte rested in isotropic filters, whose response is independent of the direction of the discontinuities in the image to which the filter is applied.

- In other words, isotropic filters are rotation invariant; in the sense that rotating the image and then applyin g the filter gives the same result as applying the filter to the image first and then rotating the result.

Development of the method:

- It can be shown (Rosenfeld and Kak [1982]) that the simplest isotropic derivative operator is the Laplacian, which, for a function (image) f(x, y) oftwo variables, is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

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

- Because derivatives of any order are linear operations, the Laplacian is a linear operator. In order to be useful for digital image processing, this equation needs to be expressed in discrete form. There are several ways to define a digital Laplacian using neighborhoods. digital second Taking into account that we now have two variables, we use the followingnotation for the partial second-order derivative in the x-direction:

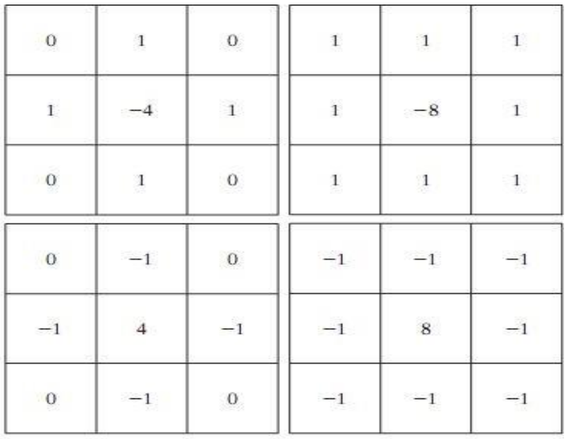
  and, similarly in the y-direction, as

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

- The digital implementation of the two -dimensional Laplacian in Eq. is obtained by summing these two components

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

- This equation can be implemented using the mask shown in Fig. 2.6.3(a), which gives an isotropic result for rotations in incr ements of 90°.
- The diagonal directions can be incorporated in the definition of the digitalLaplacian by adding two more terms to Eq., one for each of the two diagonal directions. The form of each new term is the same as either Eq.



0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Fig. 2.6.3 . (a) Filter mask used to implement the digital Laplacian (b) Maskused to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the **Laplacian.**

- but the coordinates are along the diagonals. Since each diagonal term also contains a –2f(x, y) term, the total subtracted from the difference terms now would be –8f(x, y). The mask used to implement this new definition is shown in Fig. 2.6.3(b). This mask yields isotropic results for increments of 45°. The other two masks shown in Fig. 2.6.3 also are usedfrequently in practice.
- They are based on a definition of the Laplacian that is the negative of the one we used here. As such, they yield equivalent results, but the difference in sign must be kept in mind when co mbining (by addition or subtraction) a Laplacian -filtered image with another image.
- Because the Laplacian is a derivative operator, its use highlights gray - level discontinuities in an image and deemphasizes regions with slowly varying gray levels. This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background. Background features can be "recovered" while still preserving the sharpening effect of the Laplacian operation simply by adding the original and Laplacian images. As noted in the previous paragraph, it is important to keep in mind which definition of the Laplacian is used. If the definition used has a negative center coefficient, then we subtract, rather than add, the Laplacia n image to obtain asharpened result. Thus, the basic way in which we use the Laplacian for image enhancement is as follows:

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$

Use of First Derivatives for Enhancement —The Gradient:

- First derivatives in image processing are implemented using the magnitude of the gradient. For a function f(x, y), the gradient of f at coordinates (x, y) is defined as the two -dimensional column vector

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

- The magnitude of this vector is given by

$$\begin{aligned} \nabla f &= \text{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}. \end{aligned}$$

- The components of the gradient vector itself are linear operators, but the magnitude of this vector obviously is not because of the squaring and square root operations. On the other hand, the partial derivatives are notrotation invariant (isotropic), but the magnitude of the gradient vector is. Although it is not strictly correct, the magnitude of the gradient vector often is referred to as the gradient.
- The computational burden of implementing over an entire image is not trivial, and it is common practice to approximate the magnitude of the gradient by using absolute va lues instead of squares and square roots:

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

- This equation is simpler to compute and it still preserves relative changes in gray levels, but the isotropic feature property is lost in general. However, as in the case of the Laplacian, the isotropic propertie s of the digital gradient defined in the following paragraph are preserved only for a limited number of rotational increments that depend on the masks used to approximate the derivatives. As it turns out, the most popular masks used to approximate the grad ient give the same result only for vertical and horizontal edges and thus the isotropic properties of the gradient are preserved only for multiples of 90°.

- As in the case of the Laplacian, we now define digital approximations to the preceding equations, an d from there formulate the appropriate filter masks. In order to simplify the discussion that follows, we will use the notation in Fig. 11.2 (a) to denote image points in a 3 x 3 region. For example, the center point, z5 , denotes f(x, y), z1 denotes f(x -1, y-1), and so on. The simplest approximations to a first -order derivative that satisfy the conditions stated in that section are G x = (z8 –z5) and Gy = (z6 – z5) . Two other definitions proposed by Roberts [1965] in the early development of digital image processing use cross differences:
- we compute the gradient as

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

- If we use absolute values, then substituting the quantities in the equations gives us the

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

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

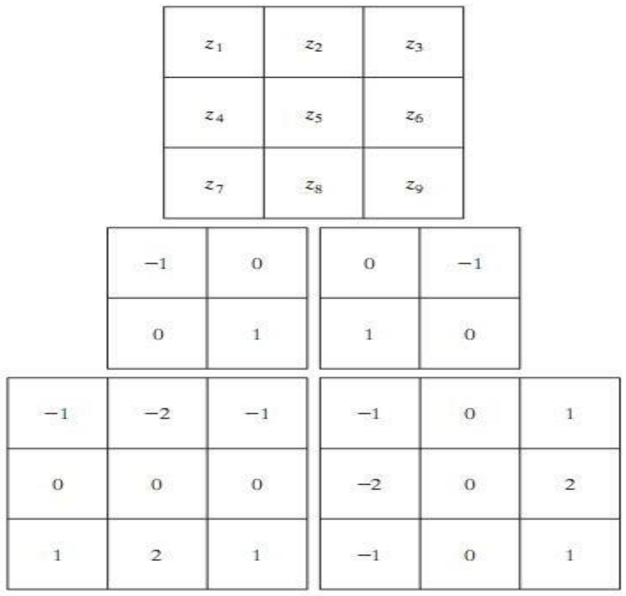
following approximation to the gradient:

- This equation can be implemented with the two masks shown in Figs.
11.2 (b) and(c). These masks are referred to as the Roberts cross -gradientoperators. Masks of even size are awkward to implement. The smallest filter mask in which we are interested is of size 3 x 3.An approximation using absolute values, still a t point z5 , but using a 3*3 mask, is
- The difference between the third and first rows of the 3 x 3 image region approximates the

$$\nabla f \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|.$$

derivative in the x -direction, and the difference between the third and first columns approximates the derivative in the y -direction. The masks shown in Figs. 2.6.4 (d) and (e), called the Sobel operators. The idea behind using a weight value of 2 is to achieve some smoothing by giving more importance to the center point. Note that the coefficients in all the masks shown in Fig. 2.6.4 sum to 0, indicating that they would give a response of 0 in an area of constant gray level, as expected of a derivative operator.

Fig 2.6.4: A 3 x 3 region of an image (the z's are gray -level values) andmasks



z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

used to compute the gradient at poin t labeled z5. All masks coefficients sum to zero, as expected of a derivative operator.

UNIT -I I

Assignment -Cum -Tuto rial Questions

Section - A

Objective Questions

1. In spatial domain, which of the following operation is done on the pixels in sharpening the image? [ ]

a. Integration b. Average c. Median d. Differentiation

2. Sum of all components i n normalized histogram is equal to [ ] a. 100 b.2 c. 0 d. 1

3. If r be the gray-level of image before processing and s after processing then which expression defines the negative transformation, for the gray-level in the range [0, L -1]? [ ]

a. s = L – 1 – r

b. s = cr□, c and □ are positive constants

c. s = c log (1 + r), c is a constant and r ≥ 0

d. none of the mentioned

4. The power-law transformation is given as: s = cr □, c and □ are positive constants, and r is the gray -level of image before processing and s after processing. Then, for what value of c and □ does power-law transformation becomes identity transformation? [ ]

a. c = 1 and □ < 1 b. c = 1 and □ > 1

c. c = –1 and □ = 0 d. c = □ = 1

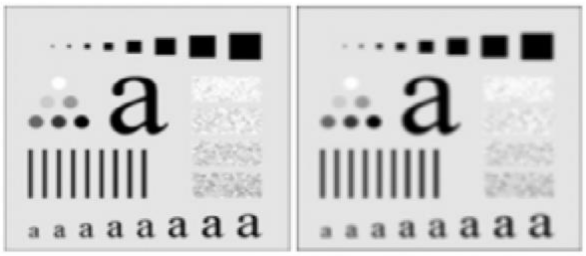
5. The power-law transformation is given as: s = cr □, c and □ are positive constants, and r is the gray -level of image before processing and s after processing. What happens if we increase the gamma value from 0.3 to 0.7?

a. The contrast increases and the detail increases [ ]b. The contrast decreases and the detail decreases

c. The contrast increases and the detail decreases

d. The contrast decreases and the detail increases

6. If the size of the averaging filter us ed to smooth the original image to first image is 9, then what would be the size of the averaging filter used in smoothing the same orig inal picture to second in second image?



a.3 b.5 c.9 d.15 [ ]

7. In power-law transformation what happens if we change the gamma value from 0.9 to 0.2? [ ]

a. The contrast increases and the detail increases b. The contrast decreases and the detail decreasesc. The contrast increases and the detail decre asesd. The contrast decreases and the detail increases

8. What is the maximum area of the cluster that can be eliminated by using an n×n median filter? [ ]

a.n^2 b. $n^2/2$ c. $2*n^2$ d. n

9. In contrast stretching, if r1=s1 and r2=s2 then which of the following is true? [ ]

a. The transformation is not a linear function that produces no changes in gray levels
b. The transformation is a linear function that produces no changes in gray levels
c. The transformation is a linear function that produces changes in gray levels d. The transformation is not a linear function that produces changes in gray levels

10. If f(x,y) is an image function of two variables, then the first order derivative
of a one dimensional function, f(x) is: [ ]

a. f(x+1)-f(x) b. f(x)-f(x+1) c. f(x-1)-f(x+1) d. f(x)+f(x-1)

11. The derivative of digital function is defined in term s of difference. Then,which of the following defines the second order derivative $\partial 2\ f/\partial x2 =$ ____________of a one-dimensional function f(x)? [ ]

a. f(x+1)-f(x) b. f(x+1)+f(x-1)-2f(x)

c. All of the mentioned depending upo n the time when partial derivative will bedealt along two spatial axes

d. None of the mentioned.

12. Discernible small details of image is [ ]

a. wide domain b. spatial domain

b. frequency domain d. algebraic domain

13.___________is the effect cau sed by the use of an insufficient number of intensity levels in smooth areas of a digital image. [ ]

a. Gaussian smooth b. Contouring

c. False Contouring d. Interpolation

Section – B

DESCRIPTIVE QUESTIONS

1. What is the objective of image enhanc ement? Define spatial domain. Definepoint processing.
2. What is meant by image enhancement by point processing? Discuss any twomethods in it.

3. Define histogram of a digital image. Explain how histogram is useful in image enhancement?
4. Write about histogram equalization.
5. Write about histogram specification.What is meant by image subtraction? Discuss various areas of application of image subtraction.
6. Explain about image averaging process.
7. Discuss about the mechanics of filtering in spatial domain. Mention the points to be considered in implementation neighborhood operations for spatial filtering.
8. Write about Smoothing Spatial filters.
9. What is meant by the Gradiant and the Laplacian? Discuss their role inimage enhancement.
10. Implement the two-dimensional Laplac ian equation for a pixel (x,y)
11. Implement first derivative enhancement function
12. What effect would setting to zero the lower -order bit planes have on thehistogram of an image in general?
13. What would be the effect on the histogram if we set to zero the higher orderbit planes instead?
14. Develop a procedure for computing the median of an n*n neighborhood.
15. Propose a technique for updating the median as the center of the neighborhood is moved from pixel to pixel.
16. Give a 3*3 mask for performing unsharp masking in a single pass throughan image.