UNIT 1

INTRODUCTION TO DIGITAL IMAGE

Structure **Page No.**

1.1 INTRODUCTION

There is a famous saying that a picture is equivalent to thousands of words, and this turns out to be true when we come to the field of image processing and pattern recognition. It is the interpretation of image which deteremines various types of outcomes viz. for an arts person it may cary a different meaning but for a science person the meaning of that image may be entirely different. Based on the utility of the images, different formats are deviced to store the different type of images, which are stored in different file formats and they contain a variety of information in the form of colour, contrast, intensity, brightness etc., but resolution of these images holds a great importance in the field of image processing, because it affects the quality of image when it is scaled up. This unit relates to the understanding of essentials of image processing and tofind major applications of image processing in our everyday life. Broadly speaking, image processing is an area that deals with manipulation of visual information called images, and one of the major objectives of image processing is to improve the quality of pictorial information for better human interpretation and to facilitate the automatic machine interpretation of images.

We shall begin our discussion by defining an image in Sec. 1.2 and will continue image acquisition and image digitization in Sec. 1.3 and Sec. 1.4 respectively. In the subsequent sections, we shall define the basic terms and concepts, which will be used throughout the course such as representation of image, resolution of image, characteristics of image and types of images. We shall end the unit by discussing various areas in which digital image processing is used.

Now, we shall list the objectives of this unit. After going through the unit, please read this list again and make sure that you have achieved the objectives.

Objectives

After going through this unit, you should be able to

- define and represent an image;
- define the various terms used in digital image processing;
- apply sampling and quantization for image digitization;
- list various classifications of images along with their descriptions;
- relate the areas where digital image processing can be applied.

Now, let us find the answer to the question "what an image is?" in the following section.

1.2 INTRODUCTION TO AN IMAGE?

When we studied optics as a subject in the field of physics, we learned that when two or more rays of light meet at a point an image is formed, and we studied various phenomenons like reflection, refraction, diffraction, dispersion, scattering etc., this means we were in the process of image generation and analysis, even much before the development of computers. But, with the advent of the computers in our day to day life this field of image processing has transformed in to an entirely new field i.e. digital image processing, where sampling and quantization techniques are applied to convert the image into its discrete form, this process of conversion is collectively known as image digitization, after digitization the image is readily available in a form suitable for further processing by digital computers. We shall discuss about the concept of sampling and quantization in the subsequent sections, here we will discuss about the concept of image only.

Many times the term 'picture' is used, but this term relates to the analog or raw image data and the term 'image' is used to refer to digital data that is suitable for the processing of images using digital computers. Infact Images are imitations of a real world objects.Image may be considered as a projection of the real world (to be more precise 2D projection of 3D world). From a photographers point of view it is a photograph (i.e. projection of real world), and from the point of view of a computer engineer an image may be a two-dimensional (2D) signal. Therefore, an image is a two-dimensional function $f(x, y)$ where for each position (x, y) in the projection plane, the values of the function $f(x, y)$ represent the amplitude or light intensity of the image.

Images are of two types, analog and digital. Analog image can be mathematically represented as a continuous range of values representing position and intensity. A digital image is composed of picture elements called **pixels**.

You may note here that through out the course we shall use the word image which refers to digital image.

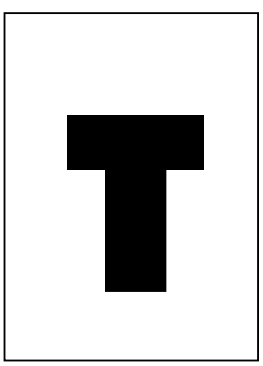
As we mentioned earlier an image is a projection of the environment under consideration, if the environment is 3D then its projection will be in 2D. If we are having an n-Dimensional space then its projection will be of n-1 dimensions. Magnetic resonance images and computerized tomography (CT) images, which are 3D images, are mathematically represented by 3D function $f(x, y, z)$, where x, y, z are spatial coordinates which may represent the amplitude or intensity or any other parameter of the image.

A digital image is defined as a 2D discrete signal that varies over the spatial coordinates x and y, and can be written mathematically as $f(x,y)$. It is also an $n \times n$ array of elements, and each element represents the sampled intensity.

In general, the image $f(x, y)$ is divided into X rows and Y columns. The coordinate ranges are $\{X = 0,1,...,m-1\}$ and $\{Y = 0,1,2,...,n-1\}$. The image can be written as a mathematical function $f(x, y)$ as given below:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2)\ldots\ldots\ldots\ldots & f(0,y-1) \\ f(1,0) & f(1,1) & f(1,2)\ldots\ldots\ldots\ldots & f(1,y-1) \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ f(m-1,0) & f(m-1,1) & f(m-1,2) & f(m-1,n-1) \end{bmatrix}$$

A sample digital image is shown in Fig. 1(a) and its equivalent matrix is shown in Fig.1(b).



0	0	0	0	0	0
0	0	0	0	0	0
0	1	1	1	1	0
0	0	1	1	0	0
0	0	1	1	0	0
		1	1		
		1	1		
		1	1		

Fig. 1: (a)Sample Digital Image and (b) Matrix of the Image

In Fig. 1(b), the elements are shown as either 0 or 1, which represents the value of the function $f(x, y)$, each element is known as **pixels** i.e. 'picture element'. They represent the discrete data of any digital image, and serve as the actual building block of digital images. The value of the function $f(x, y)$ at every point indexed by a row and a column is a number and has no units, and it is known as the **grey value** of intensity of the image at that point.

You may again refer to Fig. 1(b) , the number of rows in a digital image is called **vertical resolution**. The number of columns is called **horizontal resolution**. We shall discuss about these concepts of image resolution, later in this unit. The number of rows and columns describes the **dimensions** of the image. The image size is often expressed in terms of the rectangular pixel dimensions of the array. Images can be of various sizes. Some examples of image size are 256 x 256, 512 x 512, etc. For a digital camera, the image size is defined as the total number of pixels (specified in megapixels). For example an image with resolution 2048x2048 will have 4×10^6 pixels, that is, 4 Megapixels. Millions of pixels combine together to give a digital image, and their meaning varies with context i.e. a pixel can be considered a single sensor, photosite (physical element of the sensor array of a digital camera), element of a matrix, or display element on a monitor.

Generally, the value of the pixel is the intensity value of the image at that point, which is quantized value of the light that is captured by the sensor at the point. Typically the quantization is done in 256 levels. Thus the pixels will have values going from 0 to 255, and every pixel will need 8bits to store this information. But many times, the value of the pixel is not always the intensity value. For example, in the case of an X-ray image, the value of the pixel indicates the attenuation of the X-ray at the point. Similarly, the average Magnetic Resonance (MR) signal intensity denotes the pixel value in an MRI.

You may try the following exercises.

E1) What do you mean by the term image file format? Mention some of the frequently used image file formats.

E2) Differentiate analog image and digital image. Also, give example in support of your answer.

Now, in the following section, we shall discuss image acquisition.

1.3 IMAGE ACQUISITION

According to the fundamental concept of optics, when two or more rays of light meet at a point due to reflection or refraction or any other phenomenon, an image is formed. The information of happened phenomenon is recorded or acquired by the respective sensors as an image of that event. The process of capturing real world images and storing them into a computer is called **image acquisition**.

The question arises here, that how do we get the image in digital form. The answer is that the sensors are actually the electronic devices which taps the various types of signals, like thermal or optical or electromagnetic or any other type. Generally, the data is gathered in the analog form, which is digitized through a digitizer, and this digital signal is finally utilized by the digital computer for its processing. This leads to three types of image processing :–

- Optical Image Processing
- Analog Image Processing
- Digital Image Processing

We define these types briefly:

- **Optical Image Processing:** An optical image of 3D object is is in fact its 2D projection, which is the continuous distribution of light on a 2D surface. This 2D projection holds various types of information like 3D objects that are in focus. It also includes the study of the radiation source, and other optical processes. This optical image is available in the optical form until it is converted into analog form, leading to the area of analog image processing.

- **Analog Image Processing:** This relates to the processing of analog signals using analog circuits. Analog signals are the time varying continuous signals, often referred to as pictures. These analog signals are transformed in to digital image through the process of sampling and quantization, performed by using a digitizer and the process is known as digitization. Fig. 2 shows the steps of analog image processing.

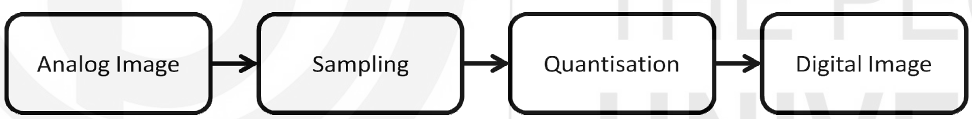


Fig. 2: Analog Image Processing

  You may note that the imaging systems that use film for recording images are also known as analog imaging systems, in medical imaging, still films are used, because these films provide better quality than digital systems.

- **Digital Image Processing:** It relates to the field of using digital circuits, systems, and software algorithms to carry out the image processing operations, which include quality enhancement of an image, counting of objects, image analysis, and many more.

Now, we know that when an interaction happens between object and rays of light (or signals) some optical or thermal or relevant phenomenon happens. The phenomenon is recorded or acquired by the respective sensors as an image of that event. This image relates to the combination of millions of pixels, each of which is carrying the information for function $f(x, y)$ or $f(x, y, z)$, which may be intensity or X-ray attenuation value, or Magnetic Resonance Intensity or any other parameter. This is gathered by using respective sensors or other devices, this data gathering relates to the field of image acquisition, which is actually the first phase of Digital Imaging. This acquired information is stored in to the memory of the digital computer for necessary processing, the final outcome of processed image is transmitted to the concerned as an information. This means that the relevant tasks performed by any digital imaging system are **data acquisition**, **storage**, **manipulation**, and **transmission.**

Thus, a typical digital imaging system or digital imaging workstation involves components for the image acquisition, storage, processing, and transmission. Types of imaging systems and acquisition systems are shown in Fig. 3.

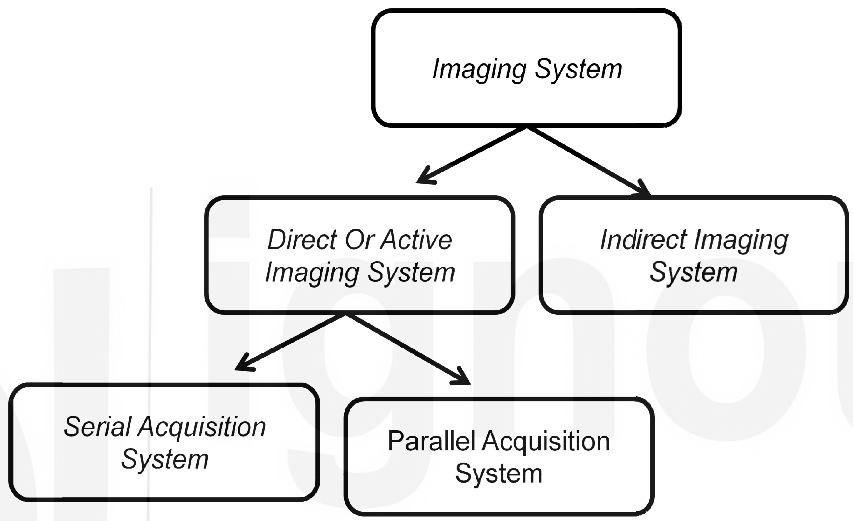


Fig. 3: Types of Imaging and Data Acquisition Systems

Let us now discuss these types briefly.

Direct or Active Imaging Systems: These are the systems where no temporary storage on film or chemical processes is required. In such systems the acquired images are in a recognizable form. We can say that in such systems the final image is a digital image. For example, human eye, digital camera, scanning confocal microscopes, etc.

The direct or active imaging systems are divided in to two sub categories i.e. serial and parallel acquisition systems. The human eye and the digital camera are example of parallel active imaging systems, where as scanning microdensitometer and scanning confocal microscopes are examples of serial acquisition imaging systems. The advantage of direct imaging systems is that the final image is a digital image.

Indirect Imaging Systems: In comparison to Direct imaging systems, the Indirect imaging systems involves data processing or reconstruction before producing an image for observation. In indirect imaging systems, the image is stored in a film, which is rendered observable by a chemical process. Therefore, there is a delay in the production of images from films.

You may now try the following exercises.

E3) Distinguish among optical image processing, analog image processing and digital image processing using examples.

E4) Differentiate between direct and indirect imaging system.

After data or image acquisition, image digitization is to be performed. We shall be using the image digitization process in image digitization.

1.4 DIGITIZATION OF IMAGES

In this section, we will try to find out the answers to three basic questions. The first question is why do we need digitization? Then we will try to find the answer to what is meant by digitization and thirdly, we will go to how to digitize an image.

We have learned from the previous sections that images are actually the imitations of real-world objects, which is generally a two-dimensional (2D) signal $f(x, y)$, where the values of the function $f(x, y)$ represent the amplitude or intensity at different points (pixels) of an image. In the processing of the analog signals or information by using digital computers, we need to convert this analog image into a discrete form using the process of digitization. This process of Digitization involves subprocess of sampling and quantization, performed through a Digitizer. The process of sampling and quantization transforms the acquired data in to the form which is suitable for further processing by digital computers.

Thus, image digitization refers to the process of sampling and quantization of the analog signals, which is required to convert the analog signal in to digital form. The digital image processing is performed using Digitizers, and this digitization process consists of two steps. One is called **sampling** and the other is called **quantization**. Sampling refers to considering the image only at a finite number of points, each image sample is called a pixel and Quantization refers to the representation of the grey level value at the sampling point using finite number of bits. For example 256 level quantization results in image storage of 8 bits per pixel.

Let us discuss the two concepts in detail.

Sampling

Consider any image. As you know that an image can be viewed as a 2 dimensional function given in the form of $f(x, y)$. Now, the image has certain length and certain height. Suppose the height of the image is H and the length of the image is L. The units of length and height would be the same. We can identify the coordinates X any Y at any space on the image. As you have seen the matrix representation of function

$f(x, y)$ in Sec. 1.2, x-axis is taken vertically and y-axis is taken horizontally, so X coordinate will vary from 0 to H and Y coordinate will vary from 0 to L. At any point (X, Y) we identify two features and write XY as product of them. Suppose these features are $r(X, Y)$ and $i(X, Y)$ represent the reflectance of the surface point and intensity of light respectively. If we consider that the maximum and minimum values of intensity are I_{max} and $I_{min,}$ then in case of continuous image $f(x, y)$ cannot attain the values either according to XY or $r(X, Y).i(x, y)$.

From the theory of real numbers you know that given any 2 points, there are infinite numbers of points. So again, when I come to this image as X varies from0 to H, there can be infinite possible values of X between 0 and H. Similarly, there can be infinite values of Y between 0 and L. So effectively, that means that if we want to represent this image in a computer, then this image has to be represented by infinite number of points and secondly when we consider the intensity value at a particular point, it varies between certain minimum I min and certain maximum Imax values, which is infinite in number.

It is clear from here that the number of points in any case would be infinite which would not be possible for computer to work on. Therefore, a way to overcome from this problem is to consider some discrete set of points, and this process of taking discrete data for any image is known as sampling.

From this discussion, we can say that since the computers can not handle the continuous data, so the continuous signal needs to be converted into digital signal and the process of this conversion is referred to as sampling.

We can visualize the process as overlaying an uniform grid on the image and sampling the image function at the center of each grid square. As we make the grid finer, we get better resolution of the image, but as we make the gridcoarser, we observe more "*pixelization*". At each pixel (or at each grid square) we usually represent the gray level value using an integer ranging from 0 for black to 255 for fully white.

To sample the signal, the signal should be frozen, as sampling cannot be applied to moving signals. If the signal is frozen for T seconds, T is called the sampling period. The **sampling period** is measured in seconds, milliseconds, or microseconds. For example, if the sampling rate is 1000Hz, it means that the signal is sampled every millisecond.

The **sampling rate** or **sampling frequency** is the reciprocal of the sampling period and is measured in samples per second or Hertz. The sampling process is the multiplication of a continuous signal $f(t)$ by a railing function $r(t)$. Railing function can be a pulse of unit amplitude at exactly the sampling time instant. We can also say that If the signal is frozen for T seconds ($T$is the sampling period) thensampled function $f(n)$ is mathematically represented as the product of continuous signal $f(t)$ by a railing function $r(t)$,

$$f(n) = f(t) \times r(t) = f(nT)$$

All these components are graphically shown in Fig. 4.

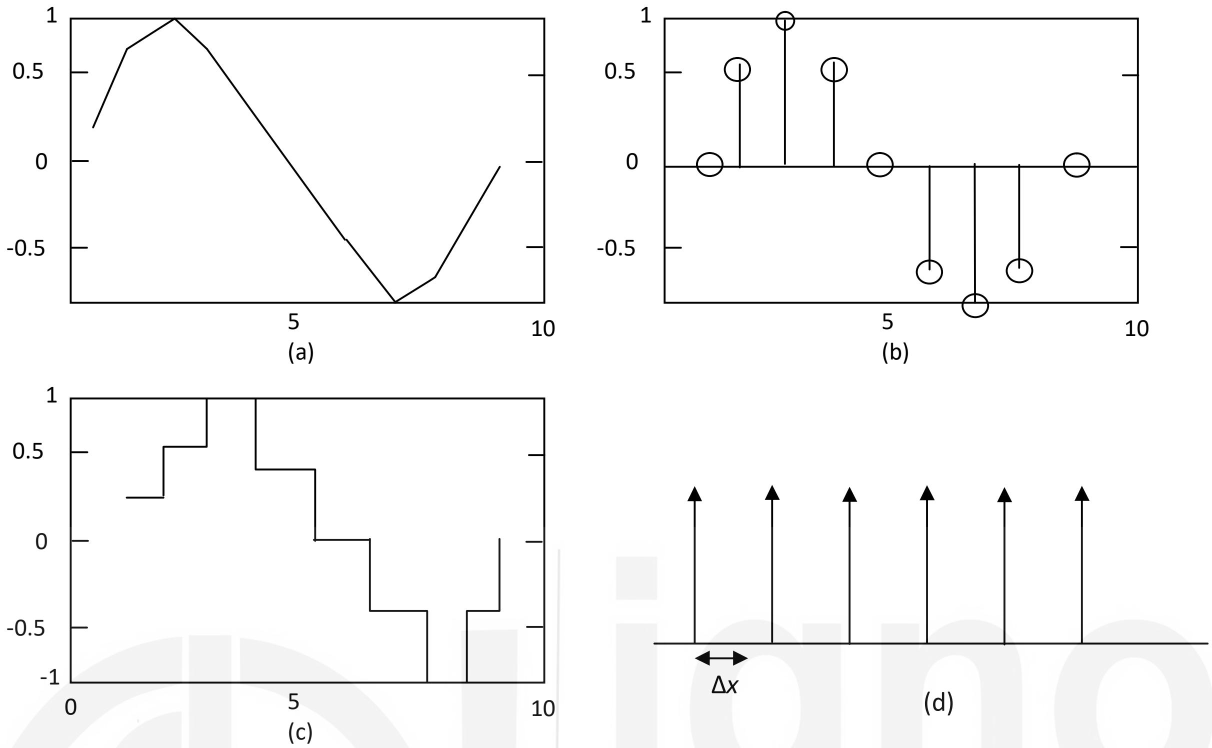


Fig. 4: **Sampling process (a) Original signal** $f(t)$ **(b) Sampled image** $f(n) = f(t) * r(t)$ **(c) Reconstructed image (d) Train of the impulse function *r*(t)**

Since, sampling is a reversible process, the reconstruction of original signal from the sampled signal is possible in both frequency and time domain. In frequency domain, the information of the base band component is required, and this base-band component over the band $\pm f_s$ (where f_s is the sampling frequency) can be extracted from the infinite spectrum of the infinite spectrum of the sampled signal, by using the low pass filters.(This statement is derived from the Shannon's Sampling Theorem).In image processing, the sampling frequency decides the distance between the samples. This distance determines the linear pixel size. Alternatively, the equivalent operation in time domain is the interpolation operation where an ideallow-pass filter is used in convolution with the interpolation function (e.g., sine function). The function $f(t)$, the railing function $r(t)$, and the sampled and reconstructed functions are all shown in Fig. 5.

This idea of one-dimensional sampling can be extended to 2D images also. The 2D railing function is known as **comb function**. It is arranged as a rectangular grid of unit impulses separated by Δx and Δy. The 2D sampling process can again be viewed as the multiplication of the railing function with the continuous function to give discrete samples. The values of Δx and Δy play an important role in image processing. Normally they are kept the same in both the horizontal and the vertical directions, so that the pixels will be a square pixels. This is called **pixel aspect ratio**. In addition, the size of the pixel is important for image quality. If the size is very large, there will be a lesser number of pixels. Hence, the details become less, which makes the image distracting and meaningless. This is called **pixelization error** where the grey level discontinuities at the edges of the pixel become poor. Then question arises that, What should be the ideal size of the pixel? Should it be big or small? The answer is give by the Shannon-Nyquist theorem.

As per Shannon-Nyquist theorem, the sampling frequency should be greater than or equal to $2 \times f_{max}$, where f_{max} is the highest frequency present in the image. Otherwise, the original signal cannot be reconstructed. In other words, the number of samples required is dictated by Shannon-Nyquist theorem, this sampling theorem can be stated in terms of distance d as $d \leq \frac{1}{2f_{max}}$. Alternatively, it should be less than or equal to the smallest detail that is present in the image.

Another frequency that is helpful in image processing is the Nyquist frequency. The Nyquist frequency (f_N) is $1/2 \times$ (sampling frequency). Therefore, the Nyquist frequency (f_N) should be greater than or equal to f_{max}. If the sampled image has frequencies higher than the Nyquist frequency, it results in a condition called aliasing where the high frequencies masquerade as low frequency components. This results in an image where interpretation becomes difficult. This problem is called **aliasing.**

Now, let us solve the following examples to understand this concept better.

Example 1: What should be the physical size of a 2D image of a document with dimensions ~~is~~ 2400×2400, when scanned at 300dpi. Here dpi stands for dots per inch.

Solution: The physical size of image is

$$= \frac{\text{Number of pixels in width}}{\text{Resolution}} \times \frac{\text{Number of pixels in height}}{\text{Resolution}}$$

$$= \frac{2400}{300} \times \frac{2400}{300}$$

$$= 8\,\text{inches} \times 8\,\text{inches}$$

***

Example 2: If the physical size of a medical image is 8×8 inches and the sampling resolution is 5 cycles/mm, then how many pixels per cycle are required to have a better quality image? Will an image of size 256×256 be enough?

Solution: It is given that the sampling resolution is 5 cycles/mm. Therefore, for better quality, 2 pixels per cycle are required. This is because sampling theorem states that the sampling frequency should be greater than twice the maximum signal frequency. This means that 10 pixels per mm are required. So, the pixel size is 0.1 mm.

You may note here that this double rates in both directions are called Nyquist rates.

The given image size (since $1\,\text{inch} = 2.54\,\text{cm}$) is

$= 8 \times 2.54 \times 8 \times 2.54 \text{ cm}^2$

$= 20.32 \times 20.32 \text{cm}^2$

Therefore, the minimum number of required pixels $= 2032 \times 2032$ pixels. So, an image of size 256×256 is not enough.

***

Now try the following exercises.

E5) The dimension of an image is 5 x 8 inches and the frequency is 500 dots per inches in each direction. Find the number of samples required to preserve the information in the image.

Now let us discuss the other step of image digitization, which is image quantization.

Quantization

Image quantization is the process of converting the sampled analog value of the function $f(x, y)$ into a discrete-valued integer. An analog signal has an infinitely larger number of distinct values. So, it is necessary to convert the continuous values into a smaller set through the process of quantization. The sampled analog image i.e. a natural image has continuously varying shades and colours and is known as a continuous tone image. This continuous tone image is required to be converted into a discrete image. The image quantizer maps a continuous value x into a discrete variable x', where discrete points of grey-tone or brightness are used.

The process of quantization involves the partitioning of input values into equally spaced intervals. The end points of the interval are called **decision boundaries**. Let the decision boundaries be given by $B = \{b_0, b_1, ..., b_m\}$, and let the input values be in the range $-X_{max}$ to $+X_{max}$. The length of the interval between successive decision boundaries i.e. the step size (Δ) is given by $\frac{2X_{max}}{m}$. The midpoint between successive decision boundaries is called **output or reconstruction level** and is given by $(R) = \{y_1, y_2, .., y_m\}$. Then quantization is in the range $\left[\text{from} -\frac{\Delta}{2} \text{to} \frac{\Delta}{2}\right]$, and the number of bits necessary to encode the output levels is given by $R = [\log_2 m]$.

Try the following exercises.

E6) What is the role of Sampling and Quantization in the process of Digitization?

E7) What do you mean by pixel aspect ratio and pixelization error?

E8) How Shannon-Nyquist theorem relates to the determination of the ideal size of the pixel?

In the following section, we shall discuss the ways how an image is represented.

1.5 REPRESENTATION OF DIGITAL IMAGE

In conventional image processing, a matrix is used to represent intensity values. Many times we need to convert this matrix to a vector and then use the normal equations or regularization based methods to represent intensity values. The normal matrix representation of image is called the Spatial Domain. There are other representation of images obtained by transforming the spatial domain image like Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) etc. Each of these transformations represents the image in the form of a mathematical formula. Further, some Partial Differential Equations (PDE) based methods use kernels to process images using 2D convolution. Sometimes we use the notion of images as point clouds and then construct a graph based on some distance measure. Then we study the Laplacians (Diffusion maps) or the Hamiltonian operators of these graphs. The Eigen values of these operators lead to the notion of heat kernel signatures and wave kernel signatures. We have these alternate views of an image based on the application, we will study some of these image representation mechanism in this section.

Going back to the basics, a digital image represents a mapping on a visual scene recorded by an optical sensor into an array of picture element intensities. Here we are discussing some of the models of image representation.

- **Pixel model:** In this model, the RGB optical sensors maps to a vector of discrete intensities which are integers. Let $p: R \times G \times B \to Z \times Z \times Z$ defined by $p(r, g, b) = (zR, zG, zB)$, where r, g, b are real valued colour intensities and zR, zG, zB are integer-valued colour intensities in the range from 0 to 255. In this case, a pixel is represented by a 1×3 column vector used to "paint" a tiny square in an image display.

- **Vector model:** In this model, the RGB optical sensor values are mapped to vectors and the displayed image conforms to a form of triangulated image space instead of a collection of pixels. In this model, a vector image is a collection of vertices and edges. The end result of vector images is the formation of a basis for applications such as Adobe Photoshop and various video games.

You can try the following exercise.

E9) Compare Pixel Model and Vector Model for the representation of Digital Image.

In the following section we will discuss about the various types of images, and their respective characteristics.

1.6 TYPES OF IMAGES

Classification of images can be performed on the basis of various criteria like attributes, colour, dimension and data types. Based on Attributes criteria the images are classified as raster and vector, whereas on the basis of colour we can classify an image as binary, greyscale, true colour and pseudo colour. Further on the basis of dimensions, we have 2D images and 3D images, and on the basis of data types the images are classified as signed integer type, unsigned integer type, float type, logical and double type. So there is no single accepted way of classifying images. The image classification is described in detail now:

- **Classification of Images on the basis of Attributes:** Based on attributes of any image, it is classified as raster and vector graphic images.
  - **Vector graphics** uses graphic primitives (point/line/circle/ellipse etc)to describe an image. Hence, the notion of resolution is practically not present in graphic.
  - **Raster images** are pixel-based, and hence their quality is dependent on the number of pixels. So, operations such as enlarging or blowing-up of a raster image often result in quality reduction.
- **Classification of images on the basis of colour:** On the basis of colour, the images can be classified into the following categories:
  - **Monochrome images** are the images where the colour component is absent, they are further classified as Grey scale and binary images.
  - **Grey scale images** : The term grey scale refers to the range of shades between white and black or vice versa, such images have many shades of grey, and eight bits ($2^8 = 256$) are enough to represent the grey scale because the human visual system can not distinguish more than 32 different grey levels, and the additional bits are necessary to cover noise margins. Most medical images such as X-rays, CT images, MRIs, and ultrasound images are grey scale images.
  - **Binary images** :The binary images are just special case of grey scale images, where the process of Thresholding is applied, they are actually Bi-level images where the pixels assumes the values of 0 or 1, i.e. only one bit is sufficient to represent the pixel value. The thresholding process involves the comparison of the threshold value i.e. the pixel value is compared with the threshold value, if the pixel value of the grey scale image is greater than the threshold value, the pixel value in the binary

image is considered as 1, Otherwise, the pixel value is 0. So far as the utility of Binary images is concerned, they are often used in representing basic shapes and line drawings. They are also used as masks. In addition, image processing operations produce binary images at intermediate stages.

- **True colour (or full colour)** images are the images where the pixel has a colour that is obtained by mixing the primary colours red, green, and blue. Each colour component is represented like a grey scale image using eight bits. Mostly, true colour images use 24 bits to represent all the colour. Hence true colour image can be considered as three-band images. The number of colours that is possible is 256^3 (i.e. 256*256*256= 1,67,77,216 colours). A display controller then uses a digital-to-analog converter (DAC) to convert the colour value to the pixel intensity of the monitor, refer to Fig.5 .

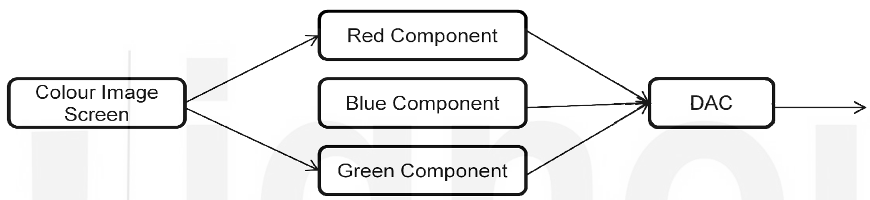


Fig. 5: True ColourImage

True colour images carries the full range of available colours in themselves. Thus, they are quite similar to the actual object and hence called true colour images. Further, the true colour images do not use any lookup table but store the pixel information with full precision.

- **Pseudocolour images** are infact false colour images, their colour component is manipulated artificially, based on the interpretation of data. Like true colour images, pseudocolour images are also used widely in images processing. We know that true colour images are called three-band images. But, in remote sensing application, multi-band images or multi-spectral images are generally used. These images, which are captured by satellites, contain many bands. A typical remote sensing image may have 3 to 11 bands in an image.This information is beyond the human perceptual range. Hence it is mostly not visible to the human observer. So colour is artificially added to these bands, so as to distinguish the bands and to increase operational convenience. These are called artificial colour or pseudocolour images. Pseudocolour images are popular in the medical domain also. For example, the doppler colour image is a pseudocolour image.

We shall discuss few examples for the better understanding of the topic.

Example3: Determine the storage space required to store 1024 x1024 pixels of a binary image?

Solution: For a binary image, one bit is sufficient for representing the pixel value. So, the number of bits required will be 1024 x 1024 x 1 =

10,48,576 bits = [(10,48,576)/8]bytes = 1,31,072 bytes = 128 KB.Since, 1 KB is 1024 bytes, the storage requirement is 128 KB.

***

Example 4: What is the storage requirement for a 1024 x 1024, 24-bit colour image?

Solution: Since colour images are three-band images (red, green, and blue components), the storage requirement is 1024*1024*3 bytes = 31,45,728 bytes. Since 1 KB is 1024 bytes, the storage requirement is 3072 KB or 3 Mega pixels.

***

We shall resume the discussion on image classification.

- **Classification of Images on the basis of Dimensions :**Images can be classified on the basis of dimensions also. Normally, digital images are a 2D rectangular array of pixels. If another dimension, of depth or any other characteristic, is considered, which may be necessary to use, then a higher-order stack of images like 3D images are produced. A good example of a 3D image is a volume image, where pixels are called voxels. By '3D image', it is meant that the dimension of the target in the imaging system (may be a scene or an object.) is three dimensional $(x, y,$ depth). In medical imaging, some of the frequently encountered 3D images are CT images, MRIs, and microscopy images. These are stored basically as 2D image slices taken across the body or the skull. Range images (often used in remote sensing application) are also 3D images as they also incorporate the depth information.

- **Classification of Images on the basis of Data Types:** Images may be classified based on their data type. Sometimes, image processing operations produce images with negative number, decimal fraction, and complex number. For example, Fourier transforms produce image involving complex numbers. To handle negative numbers, signed and unsigned integer types are used. In these data types, the first bit is used to encode whether the number is positive or negative. For example, the signed data type encodes the numbers from – 128 to 127 where one bit is used to encode the sign. In general, an $n-1$ bit signed integer can represent integers from -2^{n-1} to $2^{n-1}-1$, a total of 2^n. Unsigned integers represent all integers from 0 to $2^n - 1$ with n bits.

  Floating-point involves storing the data in scientific notation. For example, 1,230 can be represented as $0.123*10^4$, where 0.123 is called the significant and the power is called exponent. There are many floating-point conventions.

  The quality of such data representation is characterized by parameters such as data accuracy and precision. Data accuracy is the property of how well the pixel values of an image are able to represent the physical properties of the object that is being imaged. Data accuracy is an important parameter, as the failure to capture the actual physical properties of the image leads to the loss of vital information that can affect the quality of the application. While accuracy refers to the correctness of measurement, precision refers to the repeatability of the measurement. In other words, repeated measurements of the physical properties of the object should give the same result. Most software use the data type ‘double’ to maintain precision as well as accuracy.

Now try the following exercises.

E10) Make a list of all the types of classifications of images based on various parameters.

E11) Compare true Colour Images and Monochromatic Images.

In this section, we learned various types of images. Now it is time to learn about the fundamental characteristics of images because to improve the quality of an image, it is necessary to assess its quality, and this assessment requires understanding of characteristics of images, which is discussed in the following section.

1.7 IMAGE CHARACTERISTICS

To improve the quality of an image, it is necessary to assess its quality, and this assessment requires understanding of characteristics of images. some of the essential characteristics of image are Intensity, Contrast, Brightness, Noise and Resolution, we will discuss them one by one.

- **Intensity:** The term Intensity refers to the amount of light or the numerical value of a pixel, it is the measure of energy of a wave, which is directly proportional to square of amplitude of the signal. From the point of view of image processing it is a numerical value which represents a pixel, for *example* for an 8-bit gray scale image the value of a pixel can be in the range of [0,255], where (say) numericvalue “0” represents black colour and the numeric value “255” represents white colour. Therefore, *the higher the value of a pixel(i.e. intensity) whiter the* pixel will appear, i.e. in a grey scale images, intensity is depicted by the grey level value at each pixel i.e., two pixels with grey level value 127 and 220, then it can be interpreted that pixel with value 127 is darker than pixel with value 220 .

- **Contrast:** The term Contrast of an image relates to recording of the differences in the magnitude of the intensity at the surface of an object. There are many ways in which contrast can be measured,i.e.it can be described as a product of the sensor signal contrast (this depends on the energy source and the physical properties of the object) and detector contrast (this depends on the way signal is detected, captured, and stored). But, a common measurer of contrast (C) involves intensity of foreground (I_{object}) and background $(I_{background})$ objects, i.e.

$$C = \frac{I_{object} - I_{background}}{I_{object} + I_{background}}$$

Where, I_{object} is the average pixel intensity of the object pixels and $I_{background}$ is the average pixel intensity of the background. Another useful contrast measure using the same parameters I_{object} and $I_{background}$ is

$$C = \log_{10} \frac{I_{object}}{I_{background}}$$

We know, Contrast of an image relates to recording of the differences in the magnitude of the intensity at the surface of an object. *So, Contrast is the difference between maximum and minimum pixel intensities in an image.* Consider two images A& B having pixel intensities between 30 to 200 and 70 to 130, respectively. Then A has more contrast than B. Again contrast is also relative. Contrast can be simply explained as the difference between maximum and minimum pixel intensity in an image. For example, consider an image, whose maximum and minimum value of intensity is 100 i.e. same then contrast will be zero because, Contrast = maximum pixel intensity – minimum pixel intensity $= 100 - 100 = 0.$

- **Brightness: It** refers to the average pixel intensity of the image, Similar to contrast, the image should be reasonably bright to show all the information to the viewer. However, excessive brightness may affect the quality of the image. An image may have a higher brightness but if the intensity is not optimal, the image again will not exhibit good quality. *Brightness resolution* is defined as the number of identified distinct colours and it is also called as *colour resolution*.

  Brightness is a relative term which can be understood visually you can say the higher the intensity the brighter is the pixel. Since brightness is a relative term, so brightness can be defined as the amount of energy output by a source of light relative to the source we are comparing it to. In some cases we can easily say that the image is bright, and in some cases, it's not easy to perceive. Now, Question is How to make an image brighter? Brightness can be simply increased or decreased by simple addition or subtraction, to the pixel values of the image matrix. Consider a black image of 5 rows and 5 columns. All the entries of the image matrix are going to be zero since it is a completely black image. To make it brighter we shall perform some operation onthis mage. What we will do is, that we will simply add a value of 50 (or any value from 1 to 255) to each of the matrix value of image. After adding the image would become somewhat brighter then its previous state of brightness.

- **Noise:** *Noise* is an unwanted disturbance that causes fluctuations in the pixel value. It is a random or stochastic process and hence its true value cannot be predicted accurately. However, noise obeys all the statistic properties. So a pixel is characterized as a random variable for statistical analysis. It is the next important characteristic, related to the quality of image, Image applications are frequently affected by the noise present in the image.

- **Resolution:** Resolution refers to the number of pixels in an image. Resolution is sometimes identified by the width and height of the image as well as the total number of pixels in the image. For example, an image that is 2048 pixels wide and 1536 pixels high (2048 x 1536) contains (multiply) 3,145,728 pixels (or 3.1 Megapixels). You could call it a 2048 x 1536 or a 3.1 Megapixel image.

You may now try the following exercises.

E12) What do you understand by the term “Brightness of Image”? How it is different from the Contrast of any Image?

So far, we have been using the word resolution and discussed it at several places. In the following section, we shall highlight resolution in detail.

1.8 IMAGE RESOLUTION

How does image resolution play out on my computer monitor? The computer screen you are looking at right now is set at a particular resolution as well. The larger the screen, the larger you likely have your screen resolution set. If you have a 17" monitor, it is likely you have it set at 800 x 600 pixels. If you have a 19" screen it is likely set at 1024 x 768. You can change the settings but these are optimum for those screen sizes.

Now, if your monitor is set to 800 x 600 and you open up an image that is 640 x 480, it will only fill up a part of your screen. If you open up an image that is 2048 x 1536 (3.1 megapixels) then you will find yourself moving the slider bar around to see all the different parts of the image. It just won't fit. Add to that the fact that the computer monitor has a finite number of pixels per inch available (like 72) so if you are going to display your image on a monitor only, you would have to reduce the quality down to 72 to save file space. If you are going to put your image on a webpage or email it to a friend then you will want to first make it a useful size. Not too big, not too small. May be 200 or 300 pixels high would be a nice size. This means that Image resolution is highly related to the size of image and its pixel density.

We learned from the last section that the term Image Resolution relates to pixel density i.e. the number of pixels in an image, which relates to clarity of image. Parameters used to determine the Resolution of any image involves the dimensions of image i.e. its width and height and the total number of pixels in the image, as well. Say, an image has following dimensions i.e. 2048 pixels wide and 1536 pixels high thus total number of pixels in the image is (2048 x 1536) i.e. 3,145,728 pixels (or 3.1 Megapixels), i.e the image has resolution of 3.1 megapixels. As the megapixels in the pickup device in your camera increase so does the possible maximum size image you can produce. This means that a 5 megapixel camera is capable of capturing a larger image than a 3 megapixel camera.

In general the Resolution is classified as Spatial Resolution and Intensity. In simple terms, Spatial Resolution relates to the concentration of pixels in the creation of digital images and Intensity resolution of an image relates to blurredness or sharpness of an image, depending on the intensity of resolution.

- **Intensity resolution** deals with the Intensity of resolution i.e. the number of pixels per square inch, which determines the clarity or sharpness of an image. Whereas, Spatial resolution refers to the number of pixels used in making an image.

- **Spatial resolution***,* in general is defined as the number of pixels per inch of an image, the term deals with the number of pixels, pixel density, and quantization levels. Thus, it depends on the sampling and quantization processes. If the spatial resolution is not satisfactory, the quality of the image would not be satisfactory. In General, if the image has more pixels, the quality of the image is supposed to be higher. but, this is not correct, because an image may have more pixels, but it may not be providing sufficient information regarding other factors such as optical resolution, the distance of an object from the camera ,the field of view etc. which plays an important role in defining image quality.

Images with a higher number of pixels per square inch are sharp and hence said to have a higher Spatial resolution i.e. Spatial resolution is the number of pixels used for the construction of a digital image. Images that have a higher spatial resolution are made of a greater number of pixels and they are much clear as compared to the images with lower spatial resolution. In radiology, spatial resolution is used to differentiate between objects located close to each other. What do we understand by ~~what~~ spatial resolution? If we have to compare two images, to see which one is more clear or which has more spatial resolution, we need to bring the two images to same size.

Since, the spatial resolution is a measure to the clarity of image, so for different devices, different measure has been made to measure it. For example Dots per inch(Dots per inch or DPI is usually used in printers), Lines per inch(Lines per inch or LPI is usually used in laser printers), Pixels per inch (Pixel per inch or PPI is measure for different devices such as tablets , Mobile phones etc.).The higher is the PPI, the higher is the quality. In order to understand that how PPI is calculated, lets calculate the PPI of a mobile phone.

First of all we will use Pythagoras theorem to calculate the diagonal resolution in pixels. It can be given as $C = \sqrt{a^2 + b^2}$, where a and b are the height and width resolutions in pixel and C is the diagonal resolution in pixels. Now we will calculate PPI
PPI = C / diagonal size in inches

We shall see the following example.

Example 5: Calculate PPI or pixel density. For the sample mobile phone with the height and width resolutions in pixel as 1080 x 1920 pixels, and the diagonal size in inches of sample mobile phone is 5.0 inches.

Solution: So, putting those values in Pythagoras theorem i.e. the equation $C = \sqrt{a^2 + b^2}$ gives the result C = 2202.90717, Given, the diagonal size in inches of sample mobile phone is 5.0 inches, therefore PPI = 2202.90717/5.0 = 440.58 = 441 (approx).

That means that the pixel density of sample mobile phone is 441 PPI.

***

Pixel resolution, the term resolution refers to the total number of count of pixels in an digital image. For example. If an image has M rows and N columns, then its resolution can be defined as M X N. If we define resolution as the total number of pixels, then pixel resolution can be defined with set of two numbers. The first number the width of the picture, or the pixels across columns, and the second number is height of the picture, or the pixels across its height (rows).We can say that the higher is the pixel resolution, the higher is the quality of the image.We can calculate mega pixels of a camera using pixel resolution.

Pixel resolution in Megapixels = {Column pixels (width) X Row pixels (height)} / 1 Million.

Thus the size of an image can also be defined by using its pixel resolution.

Size of image = pixel resolution X bpp(bits per pixel)

Example 6: Calculate pixel resolution of a camera in mega pixels, capturing an image of dimension: 2500 X 3192.

Solution: Its pixel resolution = 2500 * 3192 = 7982350 bytes. Dividing it by 1 million = 7.9 = 8 mega pixel (approximately).

***

Now, let us discuss some important terms like *Aspect ratio, Dots per Inch, Lines per Inch*, etc. which are closely associated with the concept of image resolution

- Aspect ratio *is another important concept with the pixel resolution*. Aspect ratio is the ratio between width of an image and the height of an image. It is commonly explained as two numbers separated by a colon (8:9). This ratio differs in different images, and in different screens. The common aspect ratios are: 1.33:1, 1.37:1, 1.43:1, 1.50:1, 1.56:1, 1.66:1, 1.75:1, 1.78:1, 1.85:1, 2.00:1, etc.

  Advantage of Aspect ratio is that it maintains a balance between the appearance of an image on the screen, means it maintains a ratio between horizontal and vertical pixels. It does not let the image get distorted when aspect ratio is increased. Aspect ratio tells us many things. With the aspect ratio, you can calculate the dimensions of the image along with the size of the image.

Example 7: Given an image is an gray scale image with aspect ratio of 6:2 and pixel resolution of 480000 pixels, calculate the following:

a) Resolve pixel resolution to calculate the dimensions of image
b) Calculate the size of the image

Solution: Following are given:
Aspect ratio (i.e $c : r) = 6 : 2$

Pixel resolution (i.e. $r \times c) = 480000$

We know that bits per pixel for a grayscale image is 8bpp.

Here, we are required to find the number of rows(r) and the number of columns(c).

Using the Aspect ratio, we get $c : r = 6 : 2$ that is $c = 6r/2$

Using the pixel resolution, we get $c = 480000/r$

Comparing both relations in c and r, we get

$$\frac{6r}{2} = \frac{480000}{r} \Rightarrow r^2 = \sqrt{\frac{(480000*2}{6}} \text{ i.e. } r = 400$$

Putting $r = 400,$ we get $c = 1200;$

Thus, Rows $= 400$ and Columns $= 1200$

Now we solve for the size of image.

Size of image in bits $=$ rows * cols * bpp

Size of image in bits $= 400 * 1200 * 8 = 3840000$ bits

Size of image in bytes $=$ Size of image in bits /8 = 480000 bytes

Size of image in kilo bytes $= 469\text{kb}$ (approx).

***

- **Dots per inch(DPI):** If you have an image that is 640 x 480. How big a print can you make? Well, the true answer is you can make as big a print as you want but very quickly you will start to see "blocks" (pixelization) and the quality will drop off. To maximize the capability of your printer, you should print a picture a size that the printer can handle. Here we introduce a new term "dots per inch" (dpi) or "pixels per inch" (ppi).

The DPI is often relates to PPI (Pixels Per Inch), whereas there is a difference between these two. DPI or dots per inch is a measure of spatial resolution of printers. In case of printers, DPI means that how many dots of ink are printed per inch when an image gets printed out from the printer. Remember, it is not necessary that each Pixel per inch is printed by one dot per inch. There may be many dots per inch used for printing one pixel. The reason behind this that most of the colour printers uses CMYK model. The colours are limited. Printer has to choose from these colours to make the colour of the pixel whereas within pc, you have hundreds of thousands of colours. The higher is the dpi of the printer, the higher is the quality of the printed document or image on paper. Usually some of the laser printers have dpi of 300 and some have 600 or more.

Example 8: Determine the optimum size of the 640 X 480 image that can be printed at 200 DPI.

Solution: We have a 640 x 480 image and you want to print it at 200 dpi. 640 divided by 200 equals 3.2 and 480 divided by 200 equals 2.4 so if you print this picture at 3.2" x 2.4" you will get a print with 200 dots per inch. We recommend 200 dpi as a minimum for good quality prints.

***

Example 9 : Let's say we want to print an 8" X 10" picture at 300 dpi. What resolution must we have?

Solution:300 times 8 is 2400 and 300 times 10 is 3,000. So, we would need a 3,000 x 2,400 image i.e., 7.2 megapixels.

***

- **Lines per inch:** As dpi refers to dots per inch, liner per inch refers to lines of dots per inch. The resolution of halftone screen is measured in lines per inch. Table 2 shows some of the lines per inch capacity of the printers.

Table 2

Printer	**LPI**
Screen printing	45-65 lpi
Laser printer (300 dpi)	65 lpi
Laser printer (600 dpi)	85-105 lpi
Offset Press (newsprint paper)	85 lpi
Offset Press (coated paper)	85-185 lpi

Try an exercise.

E13) What do you understand by the term “ Resolution of an Image”? How Intensity Resolution differs from Spatial Resolution?

In the following section, we shall discuss various applications where we user digital images.

1.9 APPLICATIONS THAT USE DIGITAL IMAGES

Image processing has got wide variety of application areas, and it is widely utilized by engineers, scientists, media , fine arts professionals and many more. Nowadays it is an exciting interdisciplinary field that borrows ideas freely from many fields. Fig. 11 illustrates the relationships between image processing and other related fields.

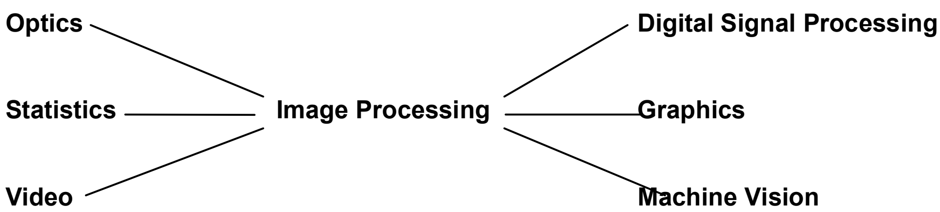


Fig. 11: Image processing and other closely related fields.

Image processing and Computer Graphics: Computer graphics and image processing are very closely related areas. Image processing deals with raster data or bitmaps, whereas computer graphics primarily deals with vector data. Raster data or bitmaps are stored in a 2D matrix form and often used to depict real images. However, vector images are composed of vectors, which represent the mathematical relationships between the objects. Vectors are lines or primitive curves that are used to describe an image. Vector graphics are often used to represent abstract, basic line drawings.

The algorithms in computer graphics often take numerical data as input and produce an image as output. However in image processing, the input is often an image. The goal of image processing is to enhance the quality of the image to assist in its interpretation. Hence, the result of image processing is often an image or the description of an image. Thus, image processing is a logical extension of computer graphics and serves as a complementary field.

Image Processing and Machine Vision : The main goal of machine vision is to interpret the image and to extract its physical, geometric, or topological properties. Thus, the output of image processing operations can be subjected to more techniques, to produce additional information for interpretation. *Artificial vision* is a vast field, with two main subfields- machine vision and computer vision. The domain of *machine vision* includes many aspects such as lighting and camera, as part of the implementation of industrial projects, since most of the applications associated with machine vision are automated visual inspection systems. The applications involving machine vision aim to inspect a large number of products and achieve improved quality controls. *Computer vision* is more ambitious. It tries to mimic the human visual system and is often associated with scene *understanding*. Most image processing algorithms produce results that can serve as the first input for machine vision algorithms.

Image Processing and Video Processing: Image processing is about still images. In fact, analog video cameras can be used to capture still images. A video can be considered as a collection of images indexed by time. Most image processing algorithms work with video readily. Thus, video processing is an extension of image processing. In addition, images are strongly related to multimedia, as the field of multimedia broadly includes the study of audio, video images, graphics, and animation.

Image Process and Optics: Optical image processing deals with lenses, light, lighting conditions, and associated optical circuits. The study of lenses and lighting conditions has an important role in the study of image processing.

Image Processing and Statistics: Image analysis is an area that concerns the extraction and analysis of object information from the image. Imaging applications involve both simple statistics such as counting and Mensuration and complex statistics such as advanced statistical inference. So statistics play an important role in imaging applications.

You may try the following exercise.

E14) Write any three applications of digital image processing.

Now, we summarise what we have studied in the unit.

1.10 SUMMARY

We have covered the following points:

1. A digital image is defined as a 2D discrete signal that varies over the spatial coordinates x and y, and can be written mathematically as $f(x, y)$. It is also an $n \times n$ array of elements, and each element represents the sampled intensity.
2. Image acquisition can be done using various ways and the concepts required to understand the acquisition of any image.
3. The two steps of image digitization are sampling and quantization.
4. Two models for representing an image.
5. Classification of the image on the basis of various attributes of image like colour, dimension and data types, etc.
6. Various terms which will be used again and again in the course.

1.11 SOLUTIONS/ANSWERS

E1) Image file format is an algorithm or a method used to store and display an image. Some of the frequently used file formats are JPEG, PNG, BMP, GIF, TIFF, etc.

E2) Some of the points of difference in analogue and digital images are:

i. Digital image's pixel value must be discrete whereas Analog image's pixel value must be continuous.

ii. The amplitude of digital image is finite whereas the amplitude of analog image is infinite.

iii. It is quite possible to store all the pixels of digital image whereas It is quite impossible to store all the pixels of analog image.

E3) **Analog Image Processing:** It is applied on analog signals/images and it processes only 2 D images. Examples are television, images, photographs, painting and medical images, etc.

Digital Image Processing: It is applied to digital images (a matrix of small pixels and elements). For example, colour processing, image recognition, video processing, etc.

Optical Image Processing: An optical image of 3D object is is in fact its 2D projection, which is the continuous distribution of light on a 2D surface. This 2D projection holds various types of information like 3D objects that are in focus. It also includes the study of the radiation source, and other optical processes. This optical image is available in the optical form until it is converted into analog form, leading to the area of analog image processing.

E4) **Direct or Active Imaging Systems**: These are the systems where no temporary storage on film or chemical processes is required. In such systems the acquired images are in a recognizable form. We can say that in such systems the final image is a digital image. For example, human eye, digital camera, scanning confocal microscopes, etc.

Indirect Imaging Systems: Incomparison to Direct imaging systems, the Indirect imaging systems involves data processing or reconstruction before producing an image for observation. In indirect imaging systems, the image is stored in a film, which is rendered observable by a chemical process. Therefore, there is a delay in the production of images from films.

E5) The bandwidth = 500 dots per inch in both directions
Therefore sample size = 1000 dots per inch [using sample theorem]

Total number of samples = 5 x 1000 x 8 x 1000 = 40000000

E6) An image may be continuous with respect to the x- and y-coordinates, and also in amplitude. Converting such an image to digital form requires that the coordinates, as well as the amplitude, be digitized.

Sampling is digitizing the coordinate values and quantization is digitizing the amplitude values.

E7) The dimensions of pixel is represented by Δx and Δy, and they are kept the same in both the horizontal and vertical directions, so that the pixels will look like square pixels. This is called pixel aspect ratio. Sometimes the size of the pixel is very large and numbers of pixels and very less, which makes the image distracting and meaning less. This is known as pixelization error.

E8) As per Shannon-Nyquist theorem, the sampling frequency should be greater than or equal to $2 \times f_{max}$, where f_{max} is the highest frequency present in the image. Otherwise, the original signal cannot be reconstructed. In other words, the number of samples required is dictated by Shannon-Nyquist theorem, this sampling theorem can be stated in terms of distance d as $d \leq \frac{1}{2f_{max}}$.

Alternatively, it should be less than or equal to the smallest detail that is present in the image.

E9) **Pixel model:** In this model, the RGB optical sensors maps to a vector of discrete intensities which are integers. Let $p : R \times G \times B \rightarrow Z \times Z \times Z$ defined by $p(r, g, b) = (zR, zG, zB)$, where where r, g, b are real valued colour intensities and zR, zG, zB are integer-valued colour intensities in the range from 0 to 255. In this case, a pixel is represented by a 1×3 column vector used to "paint" a tiny square in an image display.

Vector model: In this model, the RGB optical sensor values are mapped to vectors and the displayed image conforms to a form of triangulated image space instead of a collection of pixels. In this model, a vector image is a collection of vertices and edges. The end result of vector images is the formation of a basis for applications such as Adobe Photoshop and various video games.

E10) The classification of an image can be done on the basis of

i) Attributes
ii) Colour
iii) Dimensions
iv) Data type.

E11) **Monochrome images**are the images where the colour component is absent, they are further classified as Grey scale and binary images.

True colour (or full colour) images are the images where the pixel has a colour that is obtained by mixing the primary colours red, green, and blue. Each colour component is represented like a grey scale image using eight bits. Mostly, true colour images use 24 bits to represent all the colour. Hence true colour image can be considered as three-band images. The number of colours that is possible is 256^3 (i.e. 256*256*256= 1,67,77,216 colours).

E12) **Brightness: It** refers to the average pixel intensity of the image, Similar to contrast, the image should be reasonably bright to show all the information to the viewer. However, excessive brightness may affect the quality of the image. An image may have a higher brightness but if the intensity is not optimal, the image again will not exhibit good quality. *Brightness resolution* is defined as the number of identified distinct colours and it is also called as *colour resolution*.

We know, Contrast of an image relates to recording of the differences in the magnitude of the intensity at the surface of an object. So, Contrast is the difference between maximum and minimum pixel intensities in an image. Consider two images A& B having pixel intensities between 30 to 200 and 70 to 130, respectively. Then A has more contrast than B. Again contrast is also relative. Contrast can be simply explained as the difference between maximum and minimum pixel intensity in an image. For example, consider an image, whose maximum and minimum value of intensity is 100 i.e. same then contrast will be zero because, Contrast = maximum pixel intensity – minimum pixel intensity $= 100 - 100 = 0$.

E13) **Pixel resolution**, the term resolution refers to the total number of count of pixels in an digital image. For example. If an image has M rows and N columns, then its resolution can be defined as $M \times N$. If we define resolution as the total number of pixels, then pixel resolution can be defined with set of two numbers.

E14) Refer Section 1.9 or Browse information over relevant websites.

UNIT 2

IMAGE TRANSFORMATIONS

Structure **Page No.**

2.1 INTRODUCTION

As we discussed in the previous unit, an image is a two dimensional signal. In this unit, we shall look into the definitions of signals and their properties. These properties help in defining the relationship between images and 2D signals. Once an image is considered as a 2D signal, transformations on the signals can be applied to transform images as well. The transformations that we shall focus upon in this unit are the orthogonal transforms and unitary transforms of 2D signals. Finally, we shall discuss the properties of these transforms that are useful in image processing.

You may be familiar with some or all of these concepts from Unit-1. However, a quick look through this unit will help you in refreshing and revising your knowledge in this domain.

In Unit-1, you have gone through the definitions and properties of digital images. Therefore, you already know that when we capture an image from a camera, the image formed is a two-dimensional function that contains certain information regarding the luminance of scene objects in an RGB image or the temperature of the objects such as in a thermal image.

In Sec. 2.2, we shall discuss the definition of 1-D and 2-D signals. We then discuss the relationship between images and 2-D signals in Sec. 2.3. We discuss the orthogonal transforms and unitary transforms in Sec. 2.4 and their properties in Sec. 2.5. Finally, we summarise the discussion in Sec. 2.6 and in Sec. 2.7, we give the solutions/answers/hints to the exercises.

First, we list the objectives of this unit. After going through this unit, please read this list again and make sure that you have achieved the objectives.

Objectives

After studying this unit, you should be able to:

- define the 1-dimensional (1-D) and 2-dimensional (2-D) signals.
- relate how an image is also a 2-dimensional signal
- define and apply the unitary and orthogonal transforms of signals.
- list the properties of unitary transforms, especially useful for image processing.

Let us begin with signals in the following section.

2.2 IMAGE AS SIGNALS

You have read about the relation between image and signals in Unit-1. We begin this section by defining them in more detail. A signal is a function of one or more independent variables, where the variables represent some physical meaning. These signals carry information through variables. The dimension of a signal is defined by the number of independent variables that the function is defined on. For example, an $E \subset G$ signal is one-dimentional, whereas the intensity of a still image is 2-dimensional. You may recall the definition of a function as given below.

A function, f, is a mapping from the domain, D, that is, a set of values to another set of values called the range, P. Therefore, $f: D \to P$ such that for any element $x \in D \Rightarrow f(x) \in P$.

Therefore, a way to differentiate the signals is by the dimension of their domain, which is also the number of independent variables that the function/signal is defined on.

Let us begin our discussion by defining a signal in 1-D.

Definition(1-D Signal): A 1-dimesional (1-D) signal is a function of a single variable. A one-dimensional continuous signal is represented as a function, $f(t)$, of the independent variable t, representing the evolution of a physical phenomenon across time '*t*'. For example, a 1-D audio signal, such as the electrical signal at the output of a microphone is a function of time.

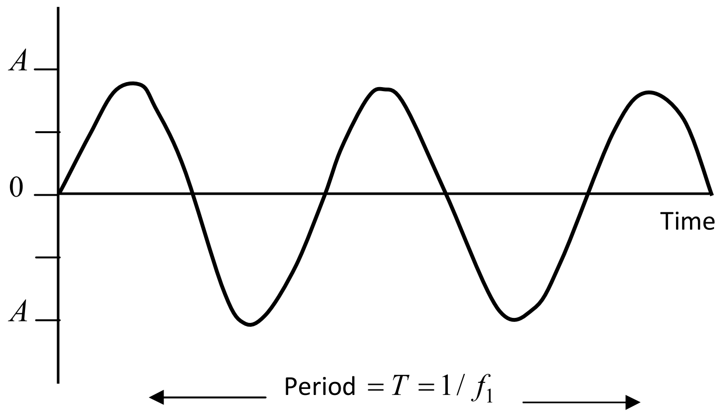


Fig. 1: 1-D Signal.

Now, we shall give the definition of 2-D signals.

Definition (2-D Signals): A 2-dimensional (2 – D) signal $f(x, y)$, is a function of two independent variables, x and y, such that (x, y) indicate a point in the 2D space.

For example, if the function f shows the intensity in a still image, then $f(x, y)$ represents the value of the intensity for the pixel at the location (x, y). Functions for 2-D signals are defined below:

i) The 2-D impulse function is defined by

$$f(x, y) = \delta(x, y) = \begin{cases} 1, & x = y = 0 \\ 0, & \text{otherwise} \end{cases}$$

ii) The line impulses are in horizontal line, vertical line or diagonal line. Accordingly,

$$f(x, y) = \begin{cases} \delta(x), & [\text{horizontal}] \\ \delta(y), & [\text{vertical}] \\ \delta(x \pm y), & [\text{diagonal}] \end{cases}$$

iii) Exponential signals is defined as $f(x, y) = n^x y^x$, where m and n are complex numbers. This function can also be written as $f(x, y) = \cos(z_1 x + z_2 y) + i \sin(z_2 x + z_2 y)$ where, $m = e^{iz_1}$ and $n = e^{iz_2}$.

Example 1: Draw the signal of $f(x, y) = \delta(x - 2y)$. Identify the signal impulse type.

Solution: Using the δ-function, we get $\delta(x - 2y) = \begin{cases} 1, & x - 2y = 0 \\ 0, & \text{otherwise} \end{cases}$.

We plot these points

x	0	1
y	0	0.5

on the graph as shown in Fig. 2.

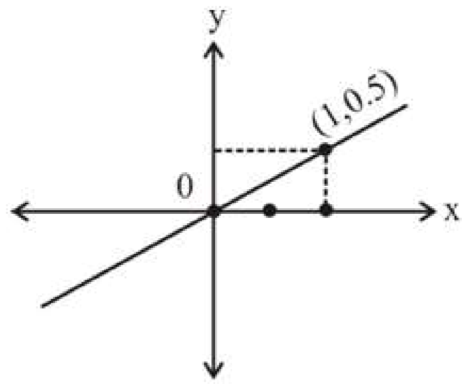


Fig. 2: Representation of $f(x, y) = \delta(x - 2y)$.

The type of signal impulse is diagonal line.

***

Try an exercise.

E1) Draw the signal of $f(x, y) = \delta(x + y - 2)$ and also, identify the type of the signal.

In the following section, we shall see how an image can be related with a 2-D signal.

2.3 IMAGE AS A 2-D SIGNAL

A grayscale image is a discrete 2-D signal $f(x, y)$, having two independent variables, x and y, such that $f(x, y)$ is the value of the signal at a pixel whose location in the image is given by integers x and y. The 2-D signals can be of various types such as seperable, periodic, etc.

Therefore, the 2D signal represents a digital image, where the (x, y) is the location of the pixel and $f(x, y)$ is the value at this pixel. Fig.3 showsthe discrete 2D signal.

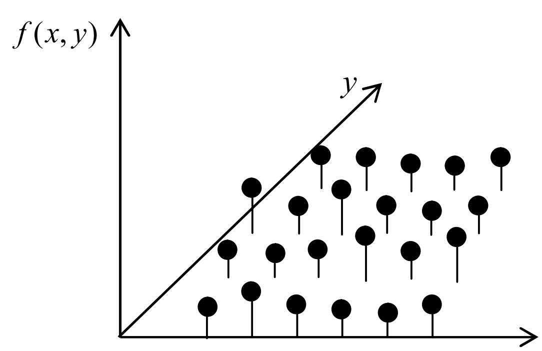


Fig. 3: A Discrete 2-D Signal

Fig.4 is another representation of a gray-scale image, where each square represents a pixel and the colour of the square gives the signal value at that location. In case of an image, the signal value represents the intensity at that location.

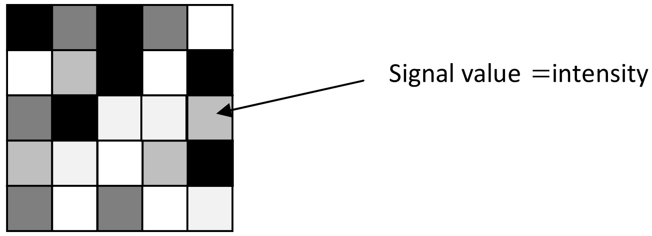
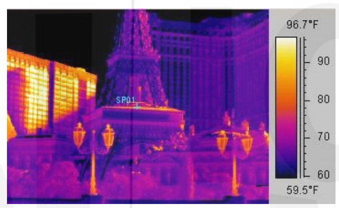


Fig.4: A Gray-Scale Image, An Example of a 2-D Signal.

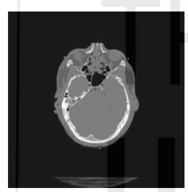
In general, brightness or intensity is usually the value of the function in a image. However, in an image, the signal value can represent physical values such as temperature, pressure, depth, etc. also. We are showing some of these examples in Fig. 5.

(a) ultrasound



(b) temperature

(c) RGB image



(d) CT scan image

Fig.5: Examples of different types of images, depending on the physical phenomenon represented by the 2D signal.

Try an exercise.

E2) Define a 2-D image in terms of 2-D signal.

Sometimes the image in 2-D signals are difficult to process and analyse due to the complexity of the function involved in them. To make such task easier we need to transform the image. Now we discuss transformation of 2-D signals needed to process an image.

2.4 TRANSFORMATIONS OF 2 D SIGNALS

Image transforms are necessary for image analysis and image processing. Transformations are mathematical functions that allow us to convert from one domain to another. Therefore, transformation of a signal $f(x)$ will convert the signal into a new signal, say, $g(y)$, in a different domain. However, transformation does not change the information content present in the signal/image.

Image transforms are important for computing correlation and convolutions. Therefore, image transforms find various applications. Depending on the transform used, the transformed image represents the image data in a more compact form, which helps in storage and transmission of the image easily. Some transforms also separate the noise from the image, making the information in the image clearer. Transformations such as Fourier transformation, cosine transformation which we shall study in the next unit, provide us the information about the frequency content in an image.

For our understanding, we first consider a one-dimensional discrete signal with N samples and represent the signal as $f(x), 0 \le x \le N-1$. Then, a transform of the signal (x) will convert the signal into a new signal $g(y)$, such that if (x) has N samples, then $g(y)$ also has N samples.

The generic form of transforms is:

$$g(u) = \sum_{x=0}^{N-1} T(u,x) f(x), 0 \le u \le N-1 \qquad (1)$$

where, $T(u,x)$ is called the **forward transformation kernel**.

The Eqn. (1) shows that to carry out the transformation, all values of $f(x)$ are required to compute each of the N values of $g(u)$. Since we can write the N values of $f(x)$ and $g(u)$ in a column vector, we can write the transform in form of matrix multiplication.

Let f and g denote these column vectors. Then, the transformation equation will be of the form

$$g = T \cdot f \qquad (2)$$

where, the matrix T is of dimension $N \times N$ and contains the values $T(y,x)$ for different y, x.

The Eqn. (1) can also be extended for transformation of 2-D signals.

A question that comes to our mind is why do we need to transform signals, especially images. Why we cannot do image processing in the spatial domain itself, where it is easy to see the picture? To answer these questions, we need to understand that many a time it is easier to carry out image processing tasks in another domain (or frequency domain) rather than in the spatial domain.

Now, we list the key steps for 2-D image transformations.

The key steps for image transformation from the spatial to the frequency domain are:

Step 1: Transform the image from the spatial domain to the frequency domain.

Step 2: Carry the required image processing task in the transformed domain.

Step 3: Apply inverse transform to return to the spatial domain.

These key steps are also listed in Fig. 6.

$$\underbrace{f(x,y)}_{\text{Spatial Domain}} \rightarrow \underbrace{\boxed{\text{Tranform}} \xrightarrow{T(u,v)} \boxed{\begin{matrix}\text{Operation}\\ R\end{matrix}} \xrightarrow{R[T(u,v)]} \boxed{\begin{matrix}\text{Inverse}\\ \text{Tranform}\end{matrix}} \rightarrow}_{\text{Transform Domain}} \underbrace{g(x,y)}_{\text{Spatial Domain}}$$

Fig. 6: Key Steps of 2-D Image Transformation.

We shall show in subsequent sections how to apply these transformations.

Try an exercise.

E3) List the steps from 2-D signal to 2-D image transformation.

In the following section, we shall discuss orthogonal transformation of 2-D signals.

2.5 ORTHOGONAL TRANSFORMATIONS OF 2-D SIGNALS

A square matrix $A = [A_1 A_2 ... A_n]$ where, A_i is the i^{th} column vector of A is real and has the property that $A^{-1} = A^T$, then it is said to be an orthogonal matrix. If, A is not real, it is said to be a unitary matrix if $A^{-1} = A^T$, that is the inverse of A is equal to its conjugate transpose.

The columns of a unitary and an orthogonal matrix are perpendicular to each other and therefore, are said to be **orthogonal**. Also, the columns of these matrices are of unit length, that is, **normalized**. Therefore, the columns of these matrices are said to be **orthonormal**. Their inner product therefore, satisfies the following:

$$(A_i, A_j) = \delta(i, j) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

The n columns of these matrices can be treated as the **basis vectors** of an n-dimensional vector space.

Let us consider an $N \times N$ image $f(x, y)$. The forward and inverse transforms of $f(x, y)$ are given by

$$g(u, v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} T(u, v, x, y) f(x, y),\ 0 \le u,\ v \in N-1. \qquad (3)$$

$$f(x, y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} I(x, y, u, v) g(u, v),\ 0 \le x, y \in N-1 \qquad (4)$$

where, $g(u, v)$ is the transformed image, $T(u, v, x, y)$ is called the **forward transformation kernel** and $I(x, y, u, v)$ is called the**inverse transformation kernel.** It is said to be an **orthogonal transform**, when T is an **orthogonal matrix**.

The forward transformation kernel satisfies the following properties:

i) **Orthonormality Property:**

$$\sum_{x,=0}^{N-1}\sum_{y=0}^{N-1} T(u, v, x, y) I(u', v', x, y) = \delta(u-u', v-v') \qquad (5)$$

ii) Completeness Property:

$$\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} T(u, v, x, y) I\ (u, v, x', y') = \delta(x-x', y-y') \qquad (6)$$

Eqns. (3) and (4) are not easy to solve. If the transform is restricted to be separable, that is $T(u, v, x, y) = A(u, x)\ B(v, y)$, then we can rewrite Eqn. (3) and Eqn. (4) as

$$g(u, v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} A(u, x) f(x, y) B(v, y) = AfA^T \qquad (7)$$

$$f(x, y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} A^*(u, v) g(u, v)\ B^*(v, y) = A^{*T} g\ A^* \qquad (8)$$

For the rectangular image $M \times N$, we get

$$g(u, v) = A_M\ f\ A_N \qquad (9)$$

$$f(x, y) = A_M^*\ g A_N^{*\,T}. \qquad (10)$$

Example 2: Consider the following orthogonal matrix A and image matrix f

$$A = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad f = \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$$

Apply the orthogonal transform and its inverse.

Solution: The transformed image, obtained according to the Eqn. (3) is

$$g = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 6 & 10 \\ -4 & -4 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 8 & -2 \\ -4 & 0 \end{bmatrix}$$

We compute the outer product of the columns of A^{*T}.

$$A^*(0,0) = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Similarly, we compute $A^*(0,1)$

$$A^*(0,1) = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} = A^{*T}(1,0) \text{ and } A^*(1,1) = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Now, we obtain the inverse transformation. We get

$$A^{*T}gA^* = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 8 & -2 \\ -4 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 4 & -2 \\ 12 & -2 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}.$$

We can verify that the inverse transformation gives us the original image.

***

Try following exercises.

E4) Is the identity matrix, I, an orthonormal matrix? Can it be used to define an orthonormal transform? What will be the inverse transformation kernel in this case?

E5) For the 2×2 transform A and the image f, which are given as:

$$A = \frac{1}{2}\begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix}, \quad f = \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}$$

Calculate the transformed image g and the basis images.

In the following section, we shall discuss unitary transforms.

2.6 UNITARY TRANSFORMS

We first discuss the unitary transforms for one dimensional signals for ease of understanding and will then move on to discussing them for 2D signals.

Unitary Transforms for 1-D signals

A one dimensional sequence $\{f(x), 0 \le x \le N-1\}$ can always be represented as a vector of dimension N, as $\mathbf{f} = [f(0)\,f(1) \ldots f(N-1)]$.

Then, a transformation T of $f(x)$ can be written as:

$$\mathbf{g} = \mathbf{T} \cdot \mathbf{f} \Rightarrow g(u) = \sum_{x=0}^{N-1} T(u,x) f(x),\; 0 \le u \le N-1 \tag{11}$$

where, T(u,x) is called the **forward transformation kernel** and $g(u)$ is the transform of $f(x)$, that is $g(u)$ is the result of applying the transformation $T(u,x)$ on $f(x)$.

There exists an inverse relation that transforms $g(u)$ back to $f(x)$, which is given by $I(x,u)$ such that

$$f(x) = \sum_{u=0}^{N-1} I(x,u) g(u),\; 0 \le x \le N-1 \tag{12}$$

where, $I(x,u)$ is called the **inverse transformation kernel**.

Therefore, in the matrix form, we can write

$$\mathbf{f} = \mathbf{I} \cdot \mathbf{g} = \mathbf{T}^{-1} \cdot \mathbf{g}$$

Recall, that a matrix is said to be a unitary matrix, if the inverse of the matrix is also its conjugate transpose. That is, if

$$\mathbf{I} = \mathbf{T}^{-1} = \mathbf{T}^{*T}$$

then the matrix $\mathbf{T}$ is called **a unitary matrix** and therefore, the transformation is also known as the **unitary transformation**.

In general, the rows (or columns) of an $N \times N$ unitary matrix are orthonormal and hence, they form a complete set of basis vectors in the N- dimensional vector space. Since,

$$\mathbf{f} = \mathbf{I} \cdot \mathbf{g} = \mathbf{T}^{*T} \cdot \mathbf{g} \Rightarrow f(x) = \sum_{u=0}^{N-1} T^{*}(u,x) g(u) \tag{13}$$

therefore, the columns of $\mathbf{T}^{*T}$, that is, the vectors $\mathbf{T}_u^{*} = \left[T^{*}(u,0)\, T^{*}(u,1) \ldots T^{*}(u,N-1)\right]^T$ are called the **basis vectors** of T.

Now let us discuss the unitary transforms of 2-D signals.

Similar to the 1D unitary transform, which allows us to represent a 1D signal by a set of orthonormal basis vectors, there exists unitary transforms in the higher dimensions. The 2D unitary transform allows us to represent a 2D signal/ image as set of basis arrays or basis images.

Given an $N \times N$ image $f(x,y)$ the forward and inverse transforms are given by

$$g(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}T(u,v,x,y)f(x,y) \tag{14}$$

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}I(x,y,u,v)g(u,v) \tag{15}$$

where, $T(u,v,x,y)$ is called the **forward transformation kernel** and $I(x,y,u,v)$ is called the **inverse transformation kernel.** It is said to be a unitary transform when T is a unitary matrix.

In general, the transformations (both orthonormal and unitary) can have the following properties:

i) **Separability:** The transformation kernel is said to be **separable** if $T(u,v,x,y)=T_1(u,x)T_2(v,y)$

ii) **Symmetry**: The transformation kernel is said to be **symmetric** if T_1 is functionally equal to T_2 such that $T(u,v,x,y)=T_1(u,x)T_1(v,y)$

The unitary transformation is an important transformation in image processing. If the forward transformation kernel $T(u,v,x,y)$ is both separable and symmetric, then the transform $g(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}T(u,v,x,y)f(x,y)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}T_1(u,x)T_1(v,y)f(x,y)$ implies:

$$\mathbf{g}=\mathbf{T_1}\cdot\mathbf{f}\cdot\mathbf{T}_1^{T} \tag{16}$$

in the matrix form, where $\mathbf{f}$ denotes the original image of size $N\times N$, and $\mathbf{T}_1$ is an $N\times N$ transformation matrix with elements $t_{ij}=T_1(i,j)$. If, in addition, $\mathbf{T}_1$ is a unitary matrix then the original image is recovered using the following equation

$$\mathbf{f}=\mathbf{T}_1^{*T}\cdot g\cdot\mathbf{T}_1^{*} \tag{17}$$

Such a transformation is called a **separable unitary transformation**.

To understand this better, let us study the following example.

Example 3: Check whether the matrix $A=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}$ is unitary or not.

Solution: The members of A are real, therefore the condition for A to be unitary is $A^{-1}=A^{T}$. We compute A^{-1} and A^{T}.

$$A^{-1}=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix} \tag{18}$$

and $\quad A^T = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$ (19)

from Eqn (18) and Eqn (19), it is clear that

$$A^{-1} = A^T$$

Hence, A is unitary matrix.

***

Try the following exercises.

E6) Why do we need image transform?

E7) Check whether the matrix $A = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}$ is unitary or not.

Now, we shall discuss some important fundamental properties of unitary transforms.

2.7 FUNDAMENTAL PROPERTIES OF UNITARY TRANSFORMS

Here we discuss the properties of unitary transforms.

i) **Energy conservation and rotation:** Let $\mathbf{T}$ be a unitary transformation, then

$$\mathbf{g} = \mathbf{T} \cdot \mathbf{f}$$

implies that

$$\|\mathbf{g}\|^2 = \|\mathbf{f}\|^2 \tag{20}$$

This means that the signal energy is preserved after the application of a unitary transformation. This property is called **energy preservation property**. It also implies that every unitary transformation is a rotation of the vector $\mathbf{f}$ in the N - dimensional vector space.

ii) Energy compaction

A large fraction of the energy in the image is packed into relatively few transform coefficients in most unitary transforms. By the above mentioned property, since the energy is preserved, this implies that most of the transform coefficients have insignificant values and only a few of the transform coefficients that are close to the origin have significant values. This property is very useful for image compression purposes.

Try an exercise.

E8) Show that if T is a unitary transformation, such that $\mathbf{g} = \mathbf{T} \cdot \mathbf{f}$ then, $\|\mathbf{g}\|^2 = \|\mathbf{f}\|^2$.

Now, let us summarise what we have discussed so far.

2.8 SUMMARY

In this unit, we discussed the following:

1. 1-D and 2-D signals and that an image is also a 2-D signal.

2. We also discussed the orthonormal and the unitary transformation and the properties of the unitary transformation that makes it very useful for image processing, especially image compression.

2.9 SOLUTION AND ANSWER

E1) Here, $x + y - 2 = 0 \Rightarrow y = 2 - x$.

The values of x and y are

x	0	1
y	2	1

We plot these points as shown in Fig. 7.

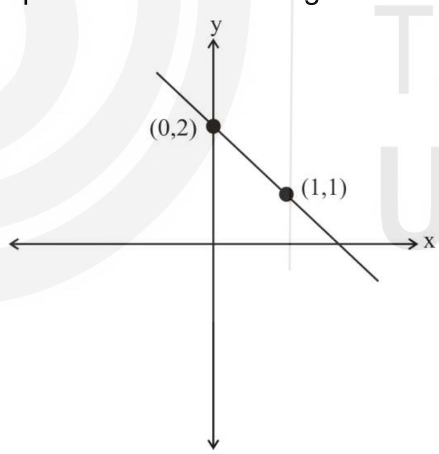


Fig. 7: Representation

E2) A grayscale image is a discrete 2-D signal $f(x, y)$, having two independent variables, x and y, such that $f(x, y)$ is the value of the signal at a pixel whose location in the image is given by x and y where, x and y are integers and the value of the signal is not defined when x and y are non-integers.

E3) First, transform the image from the spatial domain to the frequency domain. Secondly, do the image processing, and finally, apply inverse transform to return to the spatial domain.

E4) The identity matrix $A = I = [E_0, ..., E_i, ..., E_n]$, where the i^{th} column is $E_i = [0, 0, ..., 1, 0, ..., 0]$ with the i^{th} element is 1 and all other values are 0. Then, $A = I$ is an orthonormal matrix since each column is perpendicular to the others and the length of each column is unity. Therefore, it defines an orthogonal transform $Y = IX = X$. Then, the inverse transform is given by:

$$X = AY = \sum_{i=1}^{n} y_i A_i = \sum_{i=1}^{n} x_i E_i$$

Since this is an identical transformation, in this special case, the signal X and its transform, Y are identical.

E5) $$g = \frac{1}{4}\begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix}\begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} \sqrt{3} & -1 \\ 1 & \sqrt{3} \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} 2\sqrt{3}+1 & 3\sqrt{3}+2 \\ -2+\sqrt{3} & -3+2\sqrt{3} \end{bmatrix}\begin{bmatrix} \sqrt{3} & -1 \\ 1 & \sqrt{3} \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} 8+4\sqrt{3} & 8 \\ 0 & 8-4\sqrt{3} \end{bmatrix}$$

$$= \begin{bmatrix} 2+\sqrt{3} & 2 \\ 0 & 2-\sqrt{3} \end{bmatrix}$$

$$f = \frac{1}{4}\begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix}\begin{bmatrix} 2+\sqrt{3} & 2 \\ 0 & 2-\sqrt{3} \end{bmatrix}\begin{bmatrix} \sqrt{3} & -1 \\ 1 & \sqrt{3} \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} 8 & 12 \\ 4 & 8 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}$$

E6) Image transforms make image analysis and image processing easier without changing the information content present in the image.

E7) Check whether $A^{-1} = A^T$

E8) If $\mathbf{g} = \mathbf{T} \cdot \mathbf{f}$, then,

$$\|g\|^2 = \|Tf\|^2 = (Tf)^{*T}(Tf) = f^{*T}T^{*T}Tf = f^{*T}If = \|f\|^2$$

UNIT 3

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

Structure **Page No.**

5.1 INTRODUCTION

In this unit, we provide an overview of image enhancement techniques in spatial domain. These techniques improve the quality of images. The enhancement process does not increase the information content in the data. But it increases the dynamic range of the selected features so that they can be detected easily. Many point processing enhancement techniques are suggested in this unit. Image enhancement is a very important topic because of its usefulness in virtually all image processing applications.

Now, we shall list the objectives of this unit. After going through the unit, please read this list again and make sure that you have achieved the objectives.

Objectives

After studying this unit, you should be able to

- define image enhancement

- perform image enhancement in spatial domain
- perform point operations on images
- perform image enhancement using the following algorithms
  - Point operations
  - Contrast stretching
  - Clipping and thresholding
  - Digital Negative
  - Intensity levels slicing
  - Bit plane extraction

Let us begin the unit by discussing image enhancement.

5.2 IMAGE ENHANCEMENT

Image enhancement techniques improve the quality of image as perceived by human observer/ machine vision system. Enhancement techniques improve the perception of information in an image for human viewing and provide 'better' input for other automated image processing techniques. The main **objective** is to modify attributes of an image to make it more suitable for a given task or a specific observer. Image quality can degrade because of poor illumination, improper acquisition device, coarse quantization noise during acquisition process etc. The recorded images after acquisition exhibit problems such as.

- Too dark
- Too light
- Not enough contrast
- Noise

Thus, enhancement aims to improve visual quality by 'Cosmetic processing'. A process of improving the visual quality of any image so that it is more suitable for a particular application is termed as enhancement. The enhancement process does not increase the inherent information content in the data. But, it increases the dynamic range of the chosen features so that they can be detected easily. Generally, humans are the ultimate judge of the improved quality. Quality can also be objectively quantified (measured) by metrics like mean square error (MSE).

Suitability of the enhanced image heavily depends on the application. Enhancement is generally one of the preprocessing methods used on an image so that it is more suitable for further processing. For example, a finger print recognition system used for attendance recording of the employees in an organization, uses image enhancement techniques to get best recognition results under all circumstances. During finger print capturing, the quality of finger print can go down because of dust, sweat, noise, etc.Image enhancement techniques make the input more suitable for further processing so that best results can be achieved. If preprocessing is skipped and finger print matching algorithm is applied directly on the input, the algorithm can show a mismatch for an input which is otherwise a match.

Evaluation of image quality by human observer is a very subjective process and is hard to standardize. An image may be good in one person's opinion, may not be good in another person's opinion. But people's view about the quality of an image is very important and cannot be neglected. Generally, a set of 20 (or larger number) people are asked to give their opinion about the enhanced image and average results are taken.

Evaluation task for machine perception is much easier. In this case, a good image is defined as one which gives best machine recognition results. But certain amount of trial & error is generally required before a particular image enhancement approach is selected.

Image enhancement techniques are application specific and produce a 'better' image.

They are broadly classified into two categories namely

i) Spatial domain methods
ii) Frequency domain methods

In spatial (time) domain methods, pixel values are manipulated directly to get an enhanced image, whereas in frequency domain methods, firstly fourier transform of image is taken to convert image into frequency domain. Then the fourier transform is manipulated and the modified spectrum is transformed back to spatial domain to view the enhanced image. Some enhancement techniques operate on combination of these methods to get best results.

Based on what we have discussed so far, you may try the following exercises.

E1) Specify the objectives of image enhancement techniques.

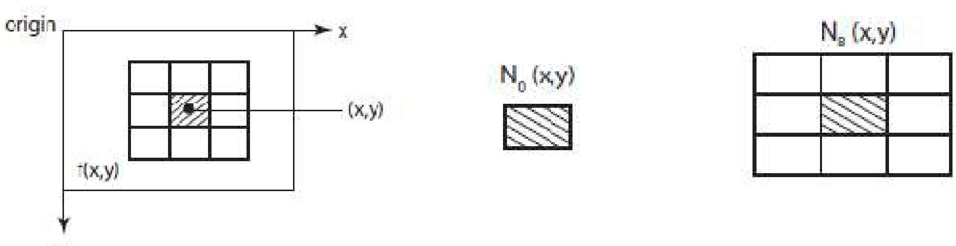
E2) What are the two types of image enhancements? Define them with the help of suitable examples.

E3) What is the importance of image enhancement in image processing?

In the following section, we shall discuss the point processing.

5.3 POINT OPERATIONS

In this unit, we shall discuss various spatial domain enhancement techniques. The pixel values (grey values) of an image are directly manipulated. Such operations simply take the grey value of each pixel/neighbouring pixels, map it to a new value and move on to the next pixel. Let $f(x, y)$ be an input image. An image processing operation in the spatial domain may be expressed as a mathematical function $T[f(x, y)]$ applied to the image $f(x, y)$ to produce a new image $g(v, v)$. Therefore, $g(x, y) = T[f(x, y)]$.



(a) (b) (c)

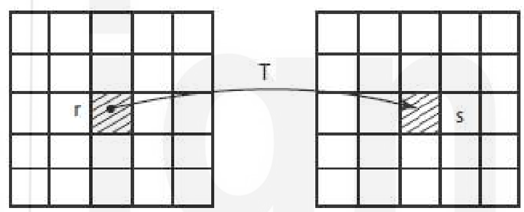
Fig. 1: Point and Neighbourhood Processing

We define the operator T applied on $f(x, y)$ over

i) a single pixel (x, y), which is called ' **point processing**', as shown in Fig. 1 (b).
ii) some neighbourhood of (x, y), which is called '**Neighbourhood processing**', as shown in Fig. 1 (c).
iii) T may operate on a set of input images instead of a single image.

The principal approach defining a neighbourhood about a point (x, y) is to use a square or rectangular sub image are centred at (x, y) as shown in Fig. 1. The centre of the sub image is moved from pixel to pixel starting at the top left corner. The operator T is applied to each location (x, y) to yield the output $g(u, v)$ at that location.

Point processing is the simplest case of spatial domain techniques where output at (x, y) only depends on the input intensity at the same point as shown in Fig. 2. It is a memoryless operation. In this case, pixels of same intensity get the same



transformation.

$f(x, y)$ $g(u, v)$

Fig. 2: Point operations

Let us see how these transformations take place.

Example 1: Perform the transformation $g_1(v) = v + 1$ on the image $f(x, y) = \begin{bmatrix} -2 & -1 & 0 \\ 0 & 1 & 2 \end{bmatrix}$.

Solution: $G_1(x, y) = g_1(f(x, y)) = \begin{bmatrix} -2+1 & -1+1 & 0+1 \\ 0+1 & 1+1 & 2+1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}$.

***

Example 2: Perform the transformation $g_2(v) = v^2$ on the image $f(x, y) = \begin{bmatrix} -2 & -1 & 0 \\ 0 & 1 & 2 \end{bmatrix}$.

Solution: $G_1(x, y) = g_2(f(x, y)) = \begin{bmatrix} (-2)^2 & (-1)^2 & 0^2 \\ 0^2 & 1^2 & 0^2 \end{bmatrix} = \begin{bmatrix} 4 & 1 & 0 \\ 0 & 1 & 4 \end{bmatrix}$.

***

Point operations are simplest yet extremely useful and powerful image processing tasks. Point operations are defined as

$g(u,v) = T[f(x,y)]$, also, $s = T(r)$
Where $r =$ grey level of input image $f(x,y)$, and
$s =$ grey level of output image $g(x,y)$

Thus, grey levels from the input image are modified with a function T which is independent of image coordinates. Typical examples of point operations are modifying image brightness, histogram processing etc.

Now, we shall discuss neighbourhood processing.

In this case, the transformation operator T is defined over the neighbourhood of (x,y). This neighbourhood can be defined using a square/ rectangular/ circular sub image that are centred at (x,y). In this method, the spatial characteristics around the pixel (x,y) can be used which is not possible in point operation.

In the Fig. 3, the operation on nine pixels around (x,y) in the input image results in manipulating the grey level values of (x,y) in the output image. Some examples of neighbourhood processing are spatial filtering, Laplatian operator etc.

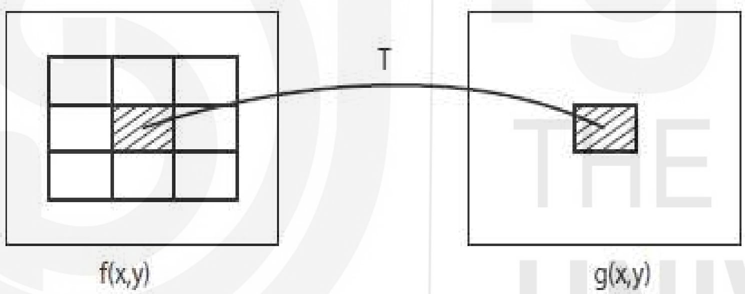


Fig. 3: Neighbourhood Processing

Point Processing

Point operations are zero memory operations where a given gray level values of an individual pixel in the input image $r \in [1, L-1]$ is mapped into a gray levels $s \in [1, L-1]$ of the pixels in the output image using the transformation $T()$.

$$\boxed{s = T(r)}$$

Try the following exercises.

E4) What do you mean by point processing?

E5) Perform the transformation $g_1(v) = 3v$ on the image
$$f(x, y) = \begin{bmatrix} -2 & -1 & 0 \\ 0 & 1 & 2 \end{bmatrix}.$$

In the following section, we shall discuss contrast stretching.

5.4 CONTRAST STRETCHING

Contrast 'c' between two levels x_1 and x_2 are defined as the absolute values of the difference, between x_1 and $x_2 \Rightarrow c(x_1, x_2) = |x_1 - x_2|$. Good contrast in an image is important to distinguish the details else they will merge in background. Low contrast images occur due to bad or non-uniform illumination conditions or due to nonlinearity of image acquisition devices. Fig. 4 shows an example of low contrast image where the details are lost in background.

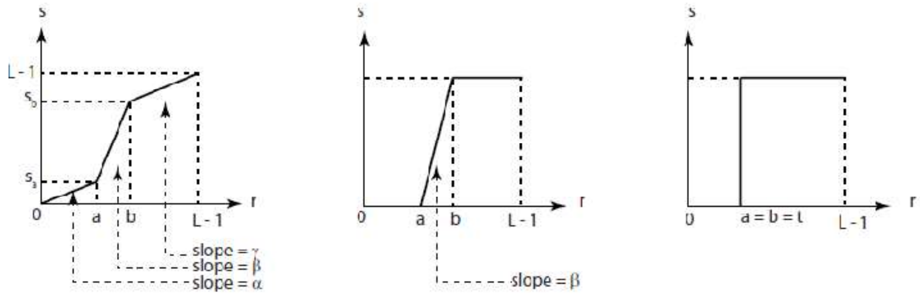
Fig. 4: A low contrast image

Value remapping by contrast stretching is a process that expands the range of levels so that all the levels contribute to the image. The main idea is to reduce the low level gray values, and to increase the mid range gray values, so that an artificial contrast is created between the two sets of gray values. The contrast between gray values is thus stretched from both sides.

The transformation is defined mathematically as follows:

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

Where α, β, γ are slopes of different regions as shown in Fig. 5 (a). The slopes determine the amount of contrast stretching (or diminishing). If the slope is greater than one, then corresponding grey levels are stretched because the original grey levels are mapped onto a larger range of grey values. A slope smaller than one means the contrast is diminished. A slope of one indicates no contrast alteration. The parameters a and b are user defined. Fig. 6 shows an example of image enhancement using contrast stretching.



(a) Contrast stretching **(b) Clipping** **(c) Thresholding**

Fig. 5

(a) low contrast image **(b) enhanced image**

Fig. 6: Effect of Contrast Stretching

Try the following exercise.

E6) What is contrast stretching?

E7) Differentiate between low contrast image and enhanced image.

In the following section, we shall discuss clipping and thresholding.

5.5 CLIPPING AND THRESHOLDING

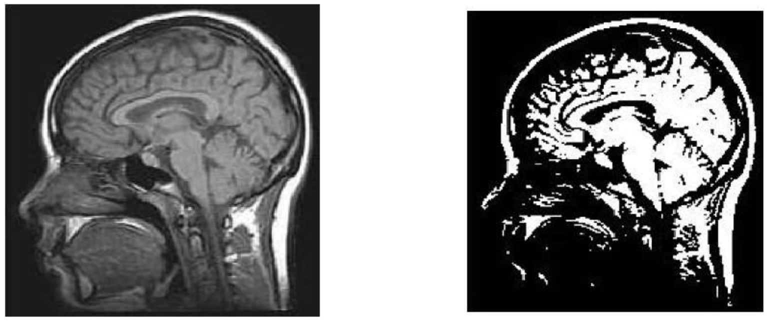
Clipping and thresholding are two special cases of contrast stretching.

a) Clipping

If $\alpha = \gamma = 0$, this is called **'clipping'** (windowing) as shown in, Fig. 5 (b). This operation stretches the contrast to its maximum in a limited range ('window') of original grey level values $\{a,......,b\}$. All the grey levels outside this window are either mapped to zero or the maximum values. Thus, clipping allows us to focus all the available contrast onto the required range of grey level values. This is especially useful when viewing medical images such as CT images. It also helps in viewing under or over exposed images.

b) Thresholding

If $\alpha = \gamma = 0$ and $a = b = t$, this remapping is called **'thresholding'** as shown in Fig. 5 (c). This operation 'binarizes' (only two values present) the image. A suitable threshold value 't' is chosen, all the gray levels smaller than 't' are mapped to zero whereas all the grey levels greater than or equal to 't' are mapped to maximum value. This is a very useful operation in image processing applications. Binarization is generally done before segmentation or region extraction. Fig. 7 shows an example of thresholding with $t = 110$. Note that input image is a gray scale image and output image is binary.



(a) original image **(b) image after thresholding**

Fig. 7: Effect of Thresholding

Example 3: Perform thresholding on image segment $f(x, y)$ with $t = 128$.

Solution:

$$\begin{array}{|c|c|c|c|}\hline 0 & 10 & 50 & 100 \\ \hline 5 & 95 & 150 & 200 \\ \hline 110 & 150 & 190 & 210 \\ \hline 175 & 210 & 255 & 100 \\ \hline \end{array} \overset{t-128}{\Longrightarrow} \begin{array}{|c|c|c|c|}\hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 255 & 255 \\ \hline 0 & 255 & 255 & 255 \\ \hline 255 & 255 & 255 & 255 \\ \hline \end{array}$$

***

Try the following exercises.

E8) What are the two special cases of contrast stretching?

E9) Why thresholding operation results in binary output?

In the following section, we shall discuss digital negatives.

5.6 DIGITAL NEGATIVES

The digital negative of an image with the intensity levels in the range $[0, L-1]$ is obtained by using negative transformation shown in Fig. 8, which is given by the expression

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

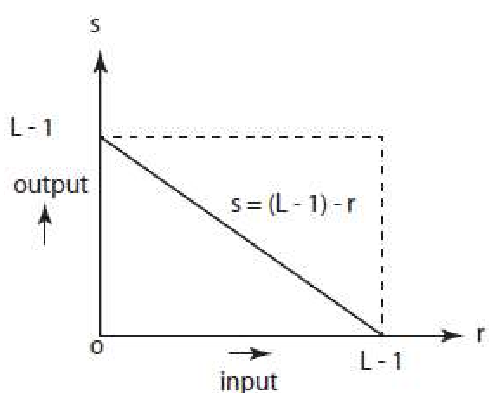


Fig. 8

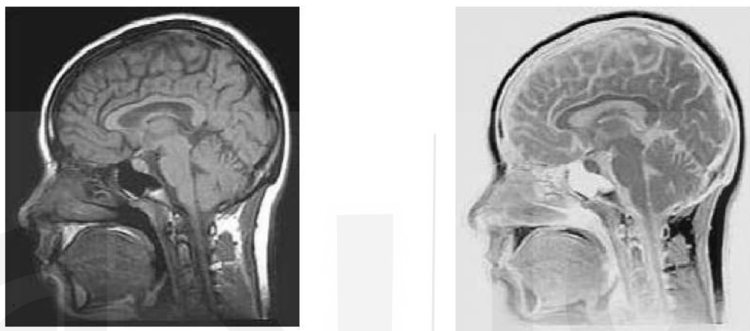
In this transformation, highest grey level is mapped to lowest and vice versa. For an8-bit image, the transformation is

$s = 255 - r$

If $r = 20,$ (dark pixel) $s = 255 - 20 = 235$ (bright pixel)

If $r = 125, s = 255 - 125 = 130$

Note that middle grey level has not changed much whereas dark grey level has become bright. This transformation is generally used to enhance the white details embedded in the dark regions of an image where black is dominant. It is useful in displaying medical images.Fig. shows an example. Even though the visual contents of both images are same, it is much easier to analyse image in negative form.



(a) original image **(b) digital negative of fig (a)**

Fig. 9: Effect of digital negative

Example 4: Find the image negative transformation on an image f given as

$$f(x,y) = \begin{array}{|c|c|c|c|}\hline 0 & 10 & 50 & 100 \\ \hline 5 & 95 & 150 & 200 \\ \hline 110 & 150 & 190 & 210 \\ \hline 175 & 210 & 255 & 100 \\ \hline \end{array}$$

Solution: The negative of the image is given as

$$g(x,y) = \begin{array}{|c|c|c|c|}\hline 255 & 245 & 205 & 155 \\ \hline 250 & 160 & 105 & 55 \\ \hline 145 & 105 & 165 & 45 \\ \hline 80 & 45 & 0 & 155 \\ \hline \end{array}$$

***

Try the following exercise.

E10) Find the negative transformation of the image $f(x,y) = \begin{bmatrix} 1 & 2 & 10 \\ 3 & 4 & 0 \\ 1 & 5 & 6 \end{bmatrix}$.

In the following section, we shall discuss about intensity level slicing.

5.7 INTENSITY LEVEL SLICING

A variant of thresholding is intensity level slicing or double thresholding. It is used to highlight a range of intensity values of interest in an image. In this operation, all the grey levels in a window $\{a, \ldots\ldots, b\}$ are set to maximum grey level and all the grey levels outside this range are set to zero as shown in Fig. 10.

$$s = \begin{cases} M & a \le r \le b \\ r & \text{otherwise} \end{cases}$$

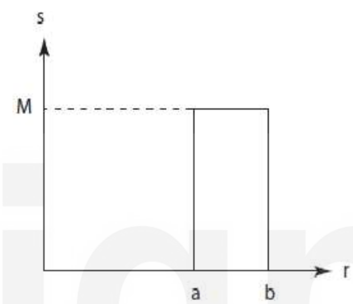


Fig. 10: Transfer Function of Intensity Level Slicing

Intensity level slicing is a binarization operation as the resulting image has only two grey level values (0 and M). Mostly this technique is used to remove unwanted elements (clutter) from an image so that the useful information becomes prominent.

Applications are enhancing certain features which are based on grey level values, such as mass of water in satellite images, cancer cells in a monogram images etc.

Sometimes, we wish to retain the original image as 'background' and highlight the grey levels in a window, following transformation can be used

$$s = \begin{cases} M & a \le r \le b \\ r & \text{otherwise} \end{cases}$$

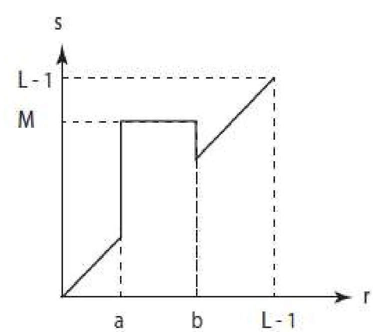
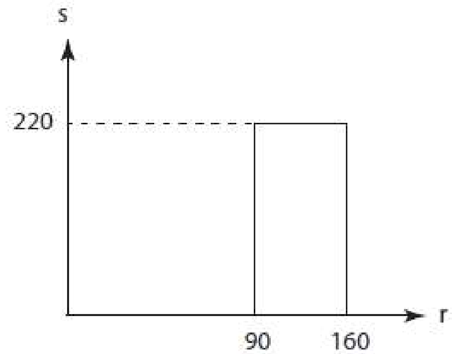


Fig. 11: Transfer Function of Intensity Level Slicing

Example 5: Perform intensity level slicing on $f(x, y)$ as given in Example 4, based on transfer function shown below



Solution: In this case, grey levels less than 90 and greater than 160 are rounded to zero and grey levels between this range are set to 220. This transformation produces a '**binary**' image. In this process, all the background is lost.

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

$f(x, y)$

$\xrightarrow{s=T(r)}$

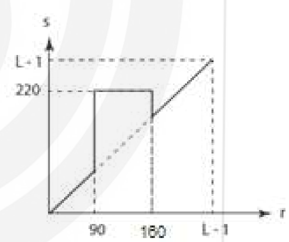
0	0	0	220
0	220	220	0
220	220	0	0
0	0	0	220

$g(x, y)$

***

Try the following exercise.

E11) Perform intensity level slicing on $f(x, y)$ as given in Example 4, based on transformation shown below:



In the following section we shall be discussing another sliding called it extraction.

5.8 BIT EXTRACTION (BIT PLANE SLICING)

Generally, a grey scale image consists of 8 bits for pixel representation. Thus, for overall appearance of the image, there is contribution by each of the 8 bits. For a pixel P at coordinates (x, y), contribution from each bit plane is

$$f(x, y) = k_1 2^7 + k_2 2^6 + k_3 2^5 + k_4 2^4 + k_3 2^3 + k_6 2^2 + k_8 2^0$$

where $k_1, ..., k_8$ are either 0 or 1. Images can be assumed to be composed of 8 one bit planes with plane 1 containing the lowest order bits of all pixels in the image and plane 8 containing all highest order bits (fig 12). Fig 13 (a) to (h) show various bit planes of the image in fig 13 (i). Observe that four highest order bit planes have most of the visually significant data. The lower order planes contribute to more subtle intensity level details in the image. For example, if pixel value is 194 (11000010 in binary form) then values of $k_1, k_2, ..., k_8$ are 1, 1, 0, 0, 0, 0, 1, 0.

As the contribution of higher order planes are much more in the image, reconstruction by only these planes results to an image very close to the original image to a great extent. To reconstruct the image using only 8th and 7th plane only is done by multiplying bit plane 8 by 128, bit plane 7 by 64 and then adding the two planes.

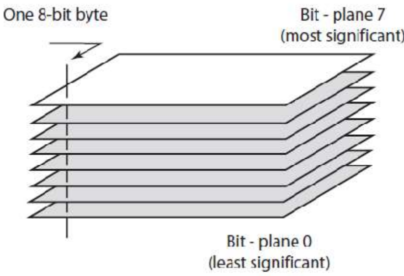
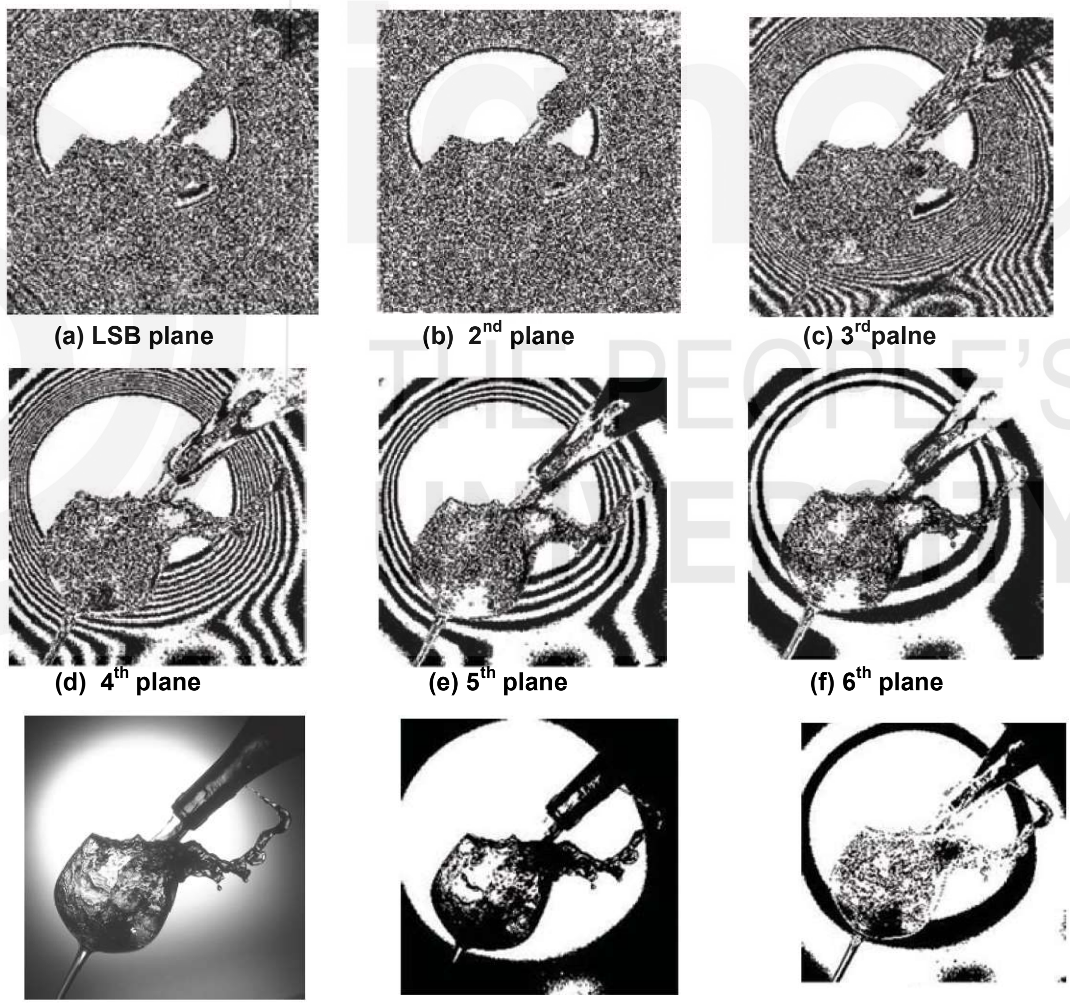


Fig. 12: Bit Plane Slicing



(a) LSB plane **(b) 2nd plane** **(c) 3rdpalne**

(d) 4th plane **(e) 5th plane** **(f) 6th plane**

(g) 7th plane **(h) msb plane** **(i) original image**

Fig. 13: Bit Plane Slicing Output

Example 6: Find MSB and LSB planes for the given image

255	138	30
65	12	201
183	111	85

Solution: Represent all grey level values in binary format.

	LSB					MSB		
255 $\rightarrow$	1	1	1	1	1	1	1	1
138 $\rightarrow$	0	1	0	1	0	0	0	1
30 $\rightarrow$	1	1	1	1	1	0	0	0
65 $\rightarrow$	1	0	0	0	0	0	1	0
12 $\rightarrow$	0	0	1	1	0	0	0	0
201 $\rightarrow$	1	0	0	1	0	0	1	1
183 $\rightarrow$	1	1	1	0	1	1	0	1
111 $\rightarrow$	1	1	1	1	0	1	1	0
85 $\rightarrow$	1	0	1	0	1	0	1	0

MSB plane consists of MSB's of all grey levels as shown in Fig. 14(a) similarly LSB plane consists of LSB's of all grey level values (Fig. 14(b)).

1	1	0
0	0	1
1	0	0

MSB plane
(a)

1	0	1
1	0	1
1	1	1

LSB plane
(b)

Fig. 14

***

Example 7: Compute various bit planes of the following 8-bit image.

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

Solution: To decompose grey level value 100 into 8-bit plane, convert 100 into binary $\rightarrow 01100100$.

Then bit for 8th plane is 0,7th plane is 1, 6th plane is 1 and so on. Similarly, binary representation of 50 is 00110010. Then bit for 8th plane 0, 7th plane is 0, 6th plane is 1 and so on.

Thus, to decompose image grey levels into various bit planes, two steps are followed:

a) Convert the grey level value into binary.
b) Allocate bits into various planes starting from MSB.

Various bit planes for the given image are:

8^{th} plane (MSB)

0	0	0	0
0	0	1	1
0	1	1	1
1	1	1	0

7^{th} plane

0	0	0	1
0	1	0	1
1	0	0	1
0	1	1	1

6^{th} plane

0	0	1	1
1	0	0	0
1	0	1	0
1	0	1	1

5^{th} plane

0	0	0	0
0	0	1	1
0	1	1	1
1	1	1	0

4^{th} plane

0	0	0	1
0	1	0	1
1	0	0	1
0	1	1	1

3^{rd} plane

0	0	1	1
1	0	0	0
1	0	1	0
1	0	1	1

2^{nd} plane

0	1	1	0
1	1	1	0
1	1	1	1
1	1	1	1

1^{st} plane

0	0	0	0
0	1	0	0
0	0	0	0
1	0	1	0

***

Try the following exercises.

E12) From all bit plane of the Example 4, generate the original image.

E13) Generate the image using only 8th, 7th plane and 6th plane.

E14) What is meant by bit plane slicing?

Now, we shall summarise the unit.

5.9 SUMMARY

In this unit, we have discussed the following points.

1. Stated Image enhancement in spatial domain.
2. Explained point processing and neighbourhood processing
3. Explained various point operations such as Contrast stretching, Clipping and thresholding, Digital Negative, Intensity levels slicing, Bit plane extraction.
4. Used special cases of Contrast stretching: Clipping and thresholding, which are very import preprocessing steps in image processing algorithms.

5.10 SOLUTIONS/ANSWERS

E1) Enhancement techniques improve the perception of information in an image for human viewing and provide ‘better’ input for other automated image processing techniques. The main **objective** is to modify attributes of an image to make it more suitable for a given task or a specific observer.

E2) Image enhancement can be broadly classified into two categories namely
  i. Spatial domain methods
  ii. Frequency domain methods

E3) Image enhancement techniques improve the quality of image as received by human observer/ machine vision system. The enhancement process does not increase the information content in the data. But it increases the dynamic range of the selected features so that they can be detected easily. Many point processing enhancement techniques are suggested in this unit. Image enhancement is a very important topic because of its usefulness in virtually all image processing applications.

E4) Point operations are zero memory operations where a given gray level values of an individual pixel in the input image $r \in [1, L-1]$ is mapped into a gray levels $s \in [1, L-1]$ of the pixels in the output image using $s = T(r)$.

E5) $$G_1(x, y) = g_2(f(x, y))$$
$$= \begin{bmatrix} -6 & -3 & 0 \\ 0 & 3 & 6 \end{bmatrix}$$

E6) Is the transformation for contrast stretching. It is used to improve the contrast of low contrast images.

$$s = \begin{cases} \alpha r & 0 \le r \le a \\ \beta(r-a) + s_a & a \le r \le b \\ \gamma(r-b) + s_b & b \ge r \le 1 \end{cases}$$

E7) You may like to differentiate low contrast image and enhance image yourself.

E8) Clipping and thresholding are two special cases of contrast stretching.

E9) Thresholding results in binary output because the output can be either 0 or 255. Multiple input gray level values are mapped to only two output vales.

E10) The negative transformation is $\begin{bmatrix} 254 & 253 & 245 \\ 252 & 251 & 255 \\ 254 & 250 & 249 \end{bmatrix}$.

E11)

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

$f(x, y)$

$\xrightarrow{s=T(r)}$

0	10	50	220
50	220	220	200
220	220	190	210
175	210	255	220

$g(x, y)$

In this case, grey levels in the window $\{90,160\}$ are set to 220 (the circled pixels in $g(x, y)$), but other grey levels are not changed. **This transformation does not produce a binary image**.

E12) To generate original image back from the bit planes we need to multiply 128 to 8th bit, 64 to 7th bit, 32 to 6th bit, 16 to 5th bit, 8 to 4th bit, 4 to 3rd bit, 2 to 2nd bit and 1 to 1st bit for each pixel and add all of it.

Thus to get 100 = 128 × 0 + 64 × 1 + 32 × 1 + 16 × 0 + 8 × 0 + 4 × 1 + 2× 0 + 1 × 0.

Similarly 50 = 128 × 0 + 64 × 1 + 32 × 1 + 16 × 0 + 8 × 0 + 4 × 0 +2 × 0 + 1 × 0 and so on the result is original image is exactly reconstructed back as shown below.

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

E13) As we have to consider only 8th 7th and 6th plane, only these plane values are added, remaining values are not considered.

$$g(0,0) = 128\times 0 + 64\times 0 + 32\times 0 = 0$$
$$g(0,1) = 128\times 0 + 64\times 0 + 32\times 0 = 0$$
$$g(0,2) = 128\times 0 + 64\times 0 + 32\times 1 = 32$$
$$g(0,3) = 128\times 0 + 64\times 1 + 32\times 1 = 96$$
$$g(1,0) = 128\times 0 + 64\times 0 + 32\times 1 = 32$$

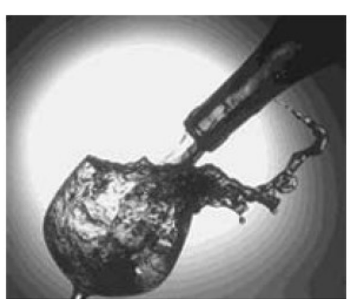
And so on. The resulting image is Fig. 16 (a). There is very small error between the reconstructed image $g(x, y)$ and original image $f(x, y)$ (Fig. 13 (i)). To find the error, we can find $e(x, y) = f(x, y) - g(x, y)$ which is shown in Fig. 16(b). Thus, with only 8th, 7th and 6th plane, the image is very close to original image. The quality keeps on increasing as more and more bit planes are added.

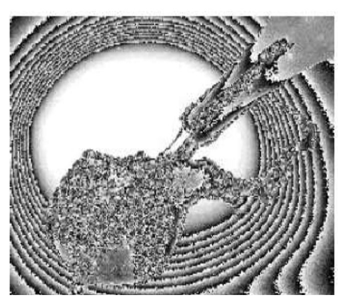
0	0	32	96
32	64	128	196
96	128	160	192
128	192	224	96

$g'(x, y)$

0	10	18	4
18	31	22	8
14	22	30	18
47	18	31	14

$e(x, y)$





(a) image reconstructed with 7th (b) error image 6th and 5th plane

Fig. 16

E14) A grey scale image consists of 8 bits for pixel representation. Thus, for overall appearance of the image, there is contribution by each bit of each pixel. For a pixel P at coordinates (x, y), contribution from each bit plane is

$$f(x, y) = k_1 2^7 + k_2 2^6 + k_3 2^5 + k_4 2^4 + k_3 2^3 + k_6 2^2 + k_8 2^0$$

UNIT 4

IMAGE FILTERING OPERATIONS IN SPATIAL DOMAIN

Structure ______________________ Page No.

4.1 INTRODUCTION

In the previous units of this course, we had seen how image transformation carried out on a specific pixel depends on the gray value of that pixel. In this unit, we shall consider more powerful transformation where the gray values of the neighbouring pixels also play a useful part. We have discussed point processing operations of the type $s = T(r)$, where r is the grey level at a single pixel in the input and s is the new value of that pixel. The capabilities of point operations are limited as the relation between a pixel and its neighbours is not exploited.

In this unit, we will be considering a neighbourhood of pixels from the input image. This is termed as **spatial filtering**. We introduce image enhancement with the help of image sharperning (spatial high pass) and image smoothing (low pass filters). Spatial filtering as generally used in preprocessing operations for noise removal or for edge detection/enhancement applications.

Subsequently, In this unit, we introduce the concept of histogram and provide an overview of image enhancement using histogram equalization and histogram specification. The histogram of an image represents the relative frequency of occurrence of the various gray levels in the image. It provides useful image statistics that help us in analyzing the image. Histograms are the basis for many spatial domain processing techniques. Histogram-modeling techniques modify an image so that its histogram has a desired shape. Histogram manipulation is used for image enhancement.

Now, we shall list the objectives of this unit. After going through the unit, please read this list again and make sure that you have achieved the objectives.

Objectives

After studying this unit, you should be able to

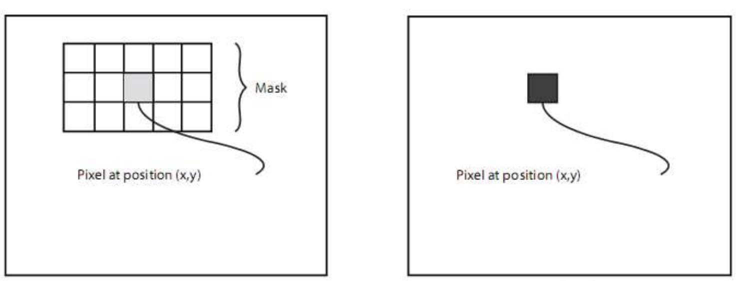
- apply spatial filtering
- perform linear filtering in spatial domain
- Use differentiate low pass & high pass filtering in spatial domain
- Use differentiate non-linear filtering and linear filtering
- Perform median, min & max filters in spatial domain

- define histogram;
- generate histogram form given image;
- perform histogram equalization;
- perform histogram specification.

4.2 SPATIAL FILTERING

Filtering is a process that removes some unwanted components or small details in an image. In digital image processing, filter is basically a subimage and is known by various names such as mask, kernel, template or window. Filters can be of two types: Spatial filters and frequency domain filters.

We have discussed point processing operations of the type $s = T(r)$, where r is the grey level at a single pixel in the input image and s is the new value of that pixel. The capabilities of point operations are limited as the relation between a pixel and its neighbors is not exploited. In this section, we will be considering a neighborhood of pixels from the input image. **Spatial filtering** is one of the main tools used in variety of applications such as noise removal, bridging the gaps in object boundaries, sharpening of edges etc.



(a) Original Image **(b) Image after filtering**

Fig. 1: Linear Filtering

The idea of spatial filtering is to move a 'mask', a square of odd size (such as 3x3, 5x5 etc.) over the whole image pixel by pixel. At each pixel, the corresponding value of the mask and the image are multiplied and added up to replace the original grey value of the pixel. This process is known as "**convolution**".

By this process, we create a new image where grey level values of the pixels are calculated from the values under the mask. The values under the mask are modified by a function called '**filter**'. If this filter function is a linear function of all grey level values in the mask, then filter is called a '**linear filter**', else it is called '**non linear filter**'.

Spatial filters can be of various types as shown in Fig. 2.

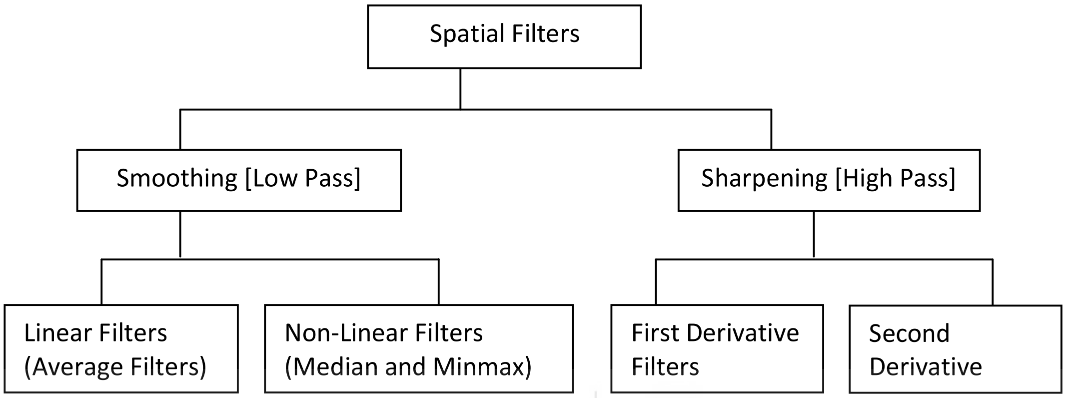


Fig. 2: Types of Spatial Filters

In this unit, we will discuss two major types of spatial filters.

i) **Smoothing:** This type of filter is used to blur (smooth) the image, therefore called smoothing filter also. These filters are also used for noise reduction. Noise reduction can be done by blurring with a linear filter, (in which operation performed on the image pixels is linear) or a non-linear filter.

ii) **Sharpening:** This filter is used to highlight transitions in intensity. These are based on first and second derivatives.

The effect of these filters is shown in Fig. 3.

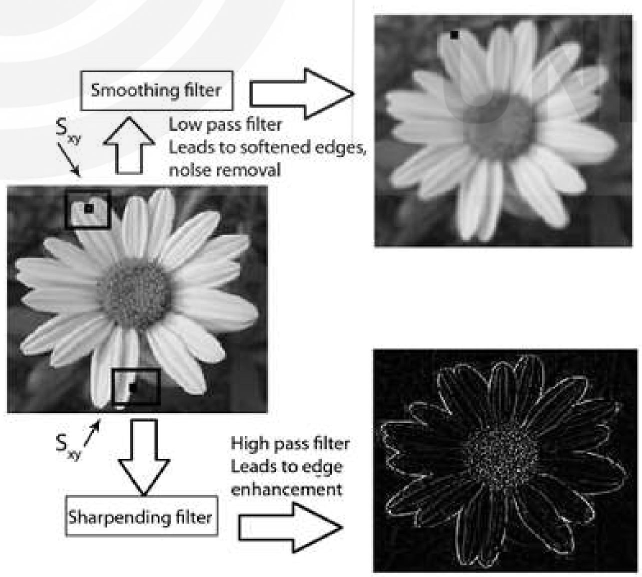


Fig. 3: Smoothing and Sharpening Filters

In the following section, we discuss image smoothing filters in detail.

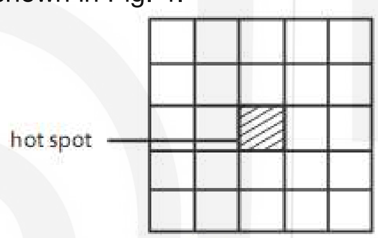
4.3 IMAGE SMOOTHING

We discussed spatial filters in previous section. Spatial filter consists of a typically small rectangle called neighbourhood and a predefined operation which is performed on the pixels of the image encompassed by the neighbourhood. Such filtering creates a new pixel with coordinates equal to the coordinates of the center of the neighbourhood, and whose value is the outcome of the filtering operation. Thus, a filtered image is generated as the center of the filter visits each pixel in the input image. If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter. Otherwise, the filter is nonlinear. Here, we shall discuss first linear filters and then some simple nonlinear filters.

Let us discuss linear spatial filters to start with.

4.3.1 Linear Filters

Linear filtering is a spatial domain process where a filter (mask/ kernel/ template) with some integer coefficient values is applied to input image to generate the filtered/ output image. Generally, filter size is either 3×3 or $5\times 5, 7\times 7$ or 21×21 (odd sizes) and filter is centered at a coordinate (x, y) called '**Hot Spot**' as shown in Fig. 4.



(b) 5 × 5 mask

Fig. 4

Linear filtering of an image $f(x, y)$ of size $M\times N$ and filter mask of size $m\times m$ is given by

$$g(x, y) = \sum_{(i,j)\in R_n} f(x+i,\ y+j)\, w(i, j) \text{ where } 0 \le x \le M-1 \text{ and } 0 \le y \le N-1. \qquad (1)$$

The right hand side of Eqn. (1) denotes the set of coordinates covered by filter. For a 3×3 filter, with coefficients $w(i, j)$ the output image at coordinate (x, y) is calculated as

$$g(x, y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x+i, y+j)\, w(i, j) \qquad (2)$$

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

We carry out the following steps for linear filtering:

Step 1: Position the mask over the current pixel such that hotspot w(0,0) coincides with current pixel.

Step 2: Form all products of filter elements with the corresponding elements in the neighbourhood.

Step 3: Add up all the products and store it at current position in the output image.

Step 4: Divide it by the scaling constant (sum of all coefficients of the mask).This must be repeated for every pixel in the image.

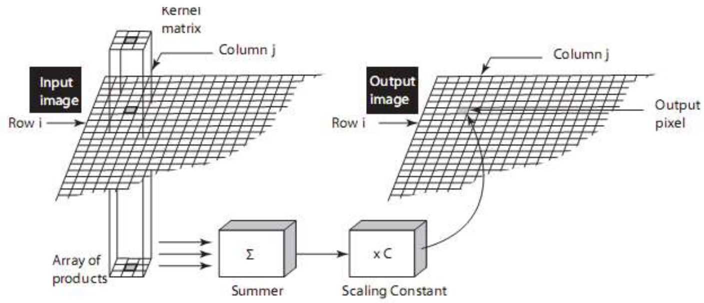


Fig 5: Linear filtering

Let us apply these steps in the following example.

Example 1: Apply given 3×3 mask w on the following image $f(x,y)$.

$$f(x,y)=\begin{array}{|ccccc|}\hline 5 & 1 & 2 & 6 & 7\\ 4 & 4 & 7 & 5 & 8\\ 2 & 6 & 20 & 6 & 7\\ 3 & 1 & 2 & 4 & 5\\ 10 & 2 & 1 & 2 & 3\\ \hline\end{array},\ w(i,j)=\frac{1}{9}*\begin{array}{|ccc|}\hline 1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\\ \hline\end{array}$$

Solution: First we consider top left 3×3 neighborhood in image $f(x,y)$ as shown in Fig. 6 (a) with hot spot coinciding with grey level 4 in $f(x,y)$ and calculate output for that pixel position by using equation

$$\text{Output } =\frac{1}{9}[5\times1+1\times1+2\times1+4\times1+4\times1+7\times1+2\times1+6+1+20\times1]$$

$$=\frac{49}{9}\approx 5$$

$$\begin{array}{|ccc|cc|}\hline 5 & 1 & 2 & 6 & 7\\ 4 & 4 & 7 & 5 & 8\\ 2 & 6 & 20 & 6 & 7\\ \hline 3 & 1 & 2 & 4 & 5\\ 10 & 2 & 1 & 1 & 3\\ \hline\end{array}\qquad\qquad\begin{array}{|c|ccc|c|}\hline 5 & 1 & 2 & 6 & 7\\ 4 & 4 & 7 & 5 & 8\\ 2 & 6 & 20 & 6 & 7\\ \hline 3 & 1 & 2 & 4 & 5\\ 10 & 2 & 1 & 1 & 3\\ \hline\end{array}$$

(a) **(b)**

Fig. 6

Shift the mask one pixel towards right such that grey level 7 coincides with hot spot (Fig. 6 (b)). Calculate output for that pixel position and replace in output image. Shift it right again and locate the mask on pixel with grey value 5, and repeat the process.

$$\text{Output } =\frac{1}{9}[1\times1+2\times1+6\times1+4\times1+7\times1+5\times1+6\times1+20\times1+6\times1]$$

$$= \frac{57}{9} \approx 6$$

Now go back to the first position of the mask and shift it down so that it is now centered on pixel with grey value 6 (just below 4), and repeat the process.

$$\text{Output } = \frac{1}{9}[2\times1+6\times1+7\times1+7\times1+5\times1+8\times1+20\times1+6\times1+7\times1]$$

$$= \frac{68}{9} \approx 8$$

Continue doing it over all the possible pixels in the given image.

$$\text{Output } = \frac{1}{9}[4+4+7+2+6+20+3+1+2] = \frac{49}{9} \approx 5$$

$$\text{Output } = \frac{1}{9}[4+7+5+6+20+6+1+2+4] = \frac{55}{9} \approx 6$$

$$\text{Output } = \frac{1}{9}[7+5+8+20+6+7+2+4+5] = \frac{64}{9} \approx 7$$

$$\text{Output } = \frac{1}{9}[2+6+20+3+1+2+10+2+1] = \frac{38}{9} \approx 4$$

$$\text{Output } = \frac{1}{9}[6+20+6+1+2+4+2+1+2] = \frac{44}{9} \approx 5$$

$$\text{Output } = \frac{1}{9}[20+6+7+2+4+5+1+2+3] = \frac{59}{9} \approx 6$$

So, by shifting the mask over the whole image a new image is generated. These values are rounded off to nearest integer values and put in the output image as given below at corresponding pixel locations.

$$g(x,y) = \begin{array}{|ccccc|} \hline * & * & * & * & * \\ * & 5 & 6 & 8 & * \\ * & 5 & 6 & 7 & * \\ * & 4 & 5 & 6 & * \\ * & * & * & * & * \\ \hline \end{array}$$

***

The filtering effect is visible in the new image where the noisy values 10, 20 in original image have been replaced by smoothed values.

Remark: There is a small problem in applying a filter at the edge of the image, where the mask partly falls outside the image. In Example 1, input image size $= 5\times5$, and the mask size $= 3\times3$. Therefore, the output image size $= 3\times3$. The size of output image is smaller than input image size as the mask does not overlap fully in the 1st row, 1^{st} column last column and last row as shown in image $g(x, y)$.

However, it is not a serious problem as in practical situations the image sizes are much larger such as 200×200 and loss of few values at the boundary of the image is not even visible to us.

Now, try the following exercises.

E1) For a 3×3 mask, explain the mechanics of linear filtering with necessary equations.

E2) List various applications of linear filtering.

Now, we discuss one of the commonly used linear filter known as a mean filter.

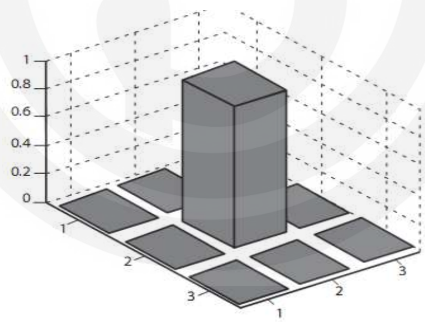
The aim of image smoothing is to diminish the effect of camera noise, spurious pixel values, missing pixel values etc. This is done by a **neighborhood averaging filter.** Each pixel in the smoothened image $g(x, y)$ is obtained from the average pixel value in the neighborhood of (x, y) in input image. Such a mask is also known as **a Mean filter.**

$$g(x, y) = \frac{1}{M} \sum_{(m,n) \in S_{xy}} f(m, n)\, w(m, n) \tag{3}$$

Where $Sxy =$ neighbourhood, $M =$ Number of pixels in Sxy and w is the mask for averaging.

Two masks averaging and weighted averaging masks are shown in Fig. 7. The output is average of pixels contained in the neighbourhood of filter mask.

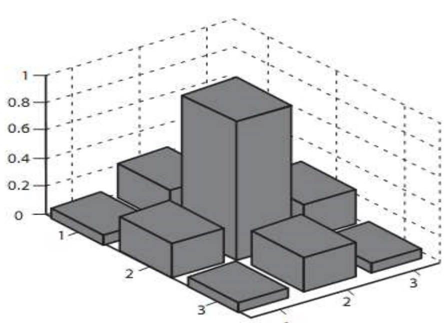
Therefore, smoothing filters are also called averaging filters or low pass filters. Fig. 7 (a) and Fig. 7 (b) shows 3×3 smoothing filter masks. Spatial filter, where all coefficients are equal, is called *box filter*. However, to give maximum importance to pixel at the centre of the mask, it is given maximum weight, and the weights of the neighnouring pixels are progressively reduced.



$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

(a) Box Filter (3×3 Averaging Mask)



$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

(b) 3×3 Weighted Averaging Mask

Fig. 7

In this case centre pixel is given the most importance and other pixels are inversely weighted as a function of their distance from the centre of the mask.

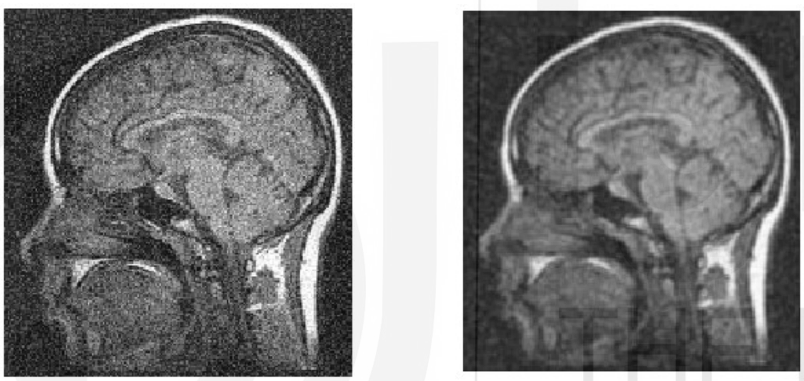
This filter reduces blurring in the smoothing process as the centre pixel is weighted highest.
Smoothing filter is used for the following purposes:

i) It reduces ‘sharp’ transitions in grey levels.
ii) It reduces noise.
iii) It blurs edges. This is a **side effect.**
iv) It helps in smoothing false colours.
v) It reduces ‘irrelevant’ details in an image.

‘**Blurring**’ (a side effect of smoothing filter) is used in preprocessing steps for following applications:

i) Removal of small details from an image prior to object extraction. Intensity of smaller objects blends with the background and larger object become ‘blob like’ and easy to detect.

ii) Bridging small gaps in lines or curves of the boundaries of objects and text. Small gaps in boundary can lead to enormous results in object extraction. Smoothing fills up all smaller gaps and helps in producing correct result.



(a) noisy image **(b) output of 3×3 averaging filter**

Fig. 8

See this example.

Example 2: Apply averaging filter to input image $f(x, y)$ as given in Fig. 10 to produce output image $g(x, y)$ in Fig. 10.

5	1	2	6	7
4	4	7	5	8
2	6	20	6	7
3	1	2	4	5
10	2	1	2	3

$f(x, y)$

$\xrightarrow[\text{filter}]{\text{smoothing}}$

5	6	8
5	6	7
4	5	6

$g(x, y)$

Fig. 9

Solution: Smoothing filter replaces every pixel of the input image by the average of the grey levels in the neighborhood. It reduces sharp transitions in grey levels. Note that pixel value 20 in $f(x, y)$ has been changed to 6 in the output image.

Sharp transitions can be due to

• Random noise in the image
• Edge of objects in the image

Smoothing filter reduces noise (desirable) and also blurs edges (undesirable). In the image $f(x, y)$, pixel value 20(high value as compared to neighborhood) becomes 6, pixel value 1 (low value as compared to neighborhood) becomes 4, whereas pixel value 7(similar value as compared to neighborhood) becomes 6.

Thus, noisy pixel or edges (20, 1: arbitrarily high/ low values) are reduced to average value, closer to neighborhood values. Note that smoothing effect will be more if instead of 3x3 mask, a large sized mask such as 7×7 or 11×11 mask is used. The size of the mask will depend on amount of noise and size of the image.

***

Try these exercises.

E3) With various spatial masks and equations, explain spatial averaging technique for enhancement.

E4) Discuss image smoothing filter in the spatial domain.

4.3.2 Nonlinear Filters

Smoothing linear filters have an important disadvantage, image structures such as individual points, edges and lines are blurred. Overall quality of the image reduces in some applications. These side effects are not tolerated which limits the usage of linear filters. Order statistics filters are non-linear filters whose response is based on ordering (ranking) the pixels contained in the image area encountered by the filter. Like all other spatial filters, non linear filters compute the result at some position (x, y) from the pixels inside the moving region S of the original image. These filters are called **non-linear** because source pixels are processed by some non-linear function.

Here, we discuss some simple non linear filters such as median filter and Min/Max filters.

Median Filters

Median filters are edge preserving smoothing filters, where the level is set to the median of pixel values in the neighborhood of that pixel. It is impossible to design a filter that removes only noise and retains all the important image structures intact, because no filter can discriminate which image content is important to the viewer and which is not. Median filter replaces every image pixel by median of the pixels in the corresponding filter region Sxy.

$$g(x, y) = \text{median}\{f(x+i, y+j) \mid (i, j) \in Sxy\} \qquad (5)$$

Median of $2k+1$ pixel values is defined as median $(p_0, p_1, p_2, ..., p_k, p_{k+1}, ..., p_{2k}) \hat{=} p_k$. Median is the centre value p_k if the sequence $(p_0, p_1, ..., p_{2k})$ is sorted.

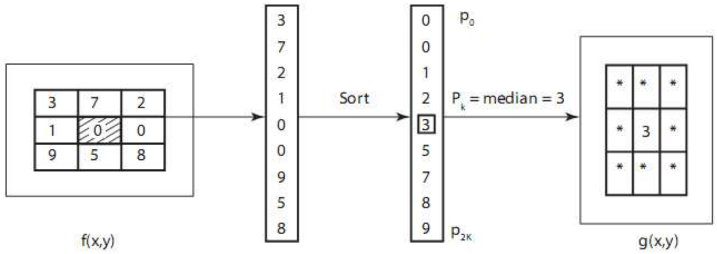


Fig 10: Median Filtering

If the number of elements are even (2k) the median of the sorted sequence $(p_0, p_1, ..., p_{k-1}, p_k, ..., p_{2k-1})$ is defined as arithmetic of two middle values

Thus, median $(p_0, p_1, ..., p_{k-1}, p_k, ..., p_{2k-1}) = \frac{p_{k-1} + p_k}{2}$

Fig. 10 shows median filtering process and generating output image. Median filter is very popular because for impulse noise (salt and pepper noise, randomly placed white and black dots), it provides excellent noise reduction capabilities with considerably less blurring than linear smoothing filter of same size. Median filters force the points with distinct levels to be more like their neighbors. It eliminates isolated clusters of pixels that are dark or light with respect to their neighbors and whose area is less than $k^2/2$ (one half of filter area).

Advantages of Median Filters

- It has excellent noise reduction capability.
- It preserves edges.
- It introduces less blurring than smoothing linear filters.
- It is very effective in the presence of impulse noise.

Disadvantages

- It performs poorly if number of noise pixels in S
- is greater than half the number of pixels in the windowxy
- It performs poorly in the presence of gaussian noise.

Steps to Implement Median Filter as shown in Fig. 10.

Step 1: Sort the values of the pixel in question and its neighbours.
Step 2: Determine their median value.
Step 3: Assign median value to that pixel.

Example 3: Compute the median value of the marked pixels show below using a 3×3 mask

$$\begin{bmatrix} 18 & 22 & 33 & 25 & 32 & 24 \\ 34 & 128 & 24 & 172 & 26 & 33 \\ 22 & 19 & 32 & 31 & 28 & 26 \end{bmatrix}$$

Solution: i) Median $(18,\ 22,\ 33,\ 34,\ 128,\ 24,\ 22,\ 19,\ 32)$

$= \text{Median } (18,\ 19,\ 22,\ 22,\ 24,\ 32,\ 33,\ 34,\ 128) = 24$

ii) Median $(22, 33, 25, 128, 24, 172, 19, 32, 31)$

$= \text{median}(19, 22, 24, 25, 31, 32, 33, 128, 172) = 31$

iii) Median $(33, 25, 32, 34, 172, 26, 32, 31, 28)$

$= \text{median}(24, 25, 26, 28, 31, 32, 32, 33, 172) = 31$

iv) Median $(25, 32, 24, 172, 26, 23, 31, 28, 26)$

$= \text{median}(23, 24, 25, 26, 26, 28, 31, 32, 172) = 26$

***

Minimum and Maximum Filter

Max and min filters are defined as:

$$g(x, y) = \min\{f(x+i, y+j) \mid (i, j) \in S_{xy}\} \quad (6)$$

$$g(x, y) = \max\{f(x+i, y+j) \mid (i, j) \in S_{xy}\}, \quad (7)$$

where S_{xy} denotes the filter region, usually a size of 3×3 pixels. **Min filter** removes salt noise (white dots with large grey level values) because any large grey level with in a 3×3 filter region is replaced by one of its surrounding pixels with smallest value. As a side effect, min filter introduces dark structures in the image. The reverse effect is expected from a **max filter**. It removes pepper noise (black dots with small grey level values) because any black dot within 3×3 filter region is replaced by one of its surrounding pixels with the largest value. White dots/bright structures are widened as a side effect and black dots (pepper noise) will disappear. Fig. 12 shows the process of min/ max filter.

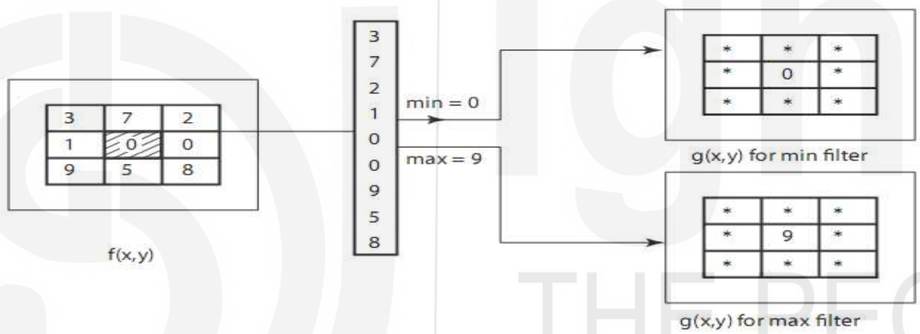


Fig. 11: Min, max filtering

Example 4: For the image segment $f(x, y)$ given below.

$f(x,y) =$

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

Apply

a) Smoothing filter
b) Weighted average filter
c) Median filter
d) Min filter
e) Max filter

Solution: (a) Smoothing filter of size 3×3

Smoothing filter of size 5×5

$$g_1(2, 2) = \frac{1}{9}[3 + 1 + 2 + 2 + 7 + 5 + 0 + 6 + 5] = \frac{31}{9} = 3.44 \approx 3$$

$$g_2(2,2)=\frac{1}{25}[0+1+0+6+5+2+3+1+2+5+1+2+7+5+4+1+0+6+5+$$
$$2+2+3+5+5+7+6]=\frac{81}{9}=9$$

b) 3×3 weighted average mask is

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

Thus for weighted average filter,

$$g_3(2,2)=\frac{1}{16}[3\times1+1\times2+2\times1+2\times2+7\times4+5\times2+0\times1+2\times6+5\times1]$$

$$=\frac{66}{16}=4.125\approx 4$$

c) Median filter of 3×3, sort the values in ascending order

$$g_4(2,2)=\text{median}\{0,1,2,2,3,5,5,6,7\}=3$$

Median filter of size 5×5, sort the values in ascending order

$$g_5(2,2)=\text{median}[0,0,0,1,1,1,1,2,2,2,2,2,3,3,4,5,5,5,5,5,6,6,6,7,7]=3$$

d) For min filter of size 3×3

$$g_6(2,2)=0$$
$$g_7(2,2)=0$$
$$g_8(2,2)=7$$

For min filter of size 5×5

e) For max filter of size 3×3 and 5×5

***

Try these exercises.

E5) What is a Median filter?

E6) What is maximum filter and minimum filter?

E7) Differentiate linear spatial filter and non-linear spatial filter.

Now, in the following section, we shall discuss sharpening spatial filters.

4.4 IMAGE SHARPENING

Image sharpening is opposite of image smoothing. This is done to highlight fine details and edges in an image. Applications of image sharpening are outlining object boundaries in industrial applications, identifying tumor boundaries, or bone structures in medical imaging etc. Smoothing is achieved by pixel averaging which is analogous to integration. Image sharpening is just the reverse process, and is achieved by differentiation. The derivative operation enhances the degree of discontinuity in an image.

The advantages of image sharpening are the following.

- It enhances edges and other discontinuities (noise) in an image.
- It de-emphasizes area with slowly varying grey levels (background) in an image. First order derivative of a one-dimensional signal f(x) is defined as

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \qquad (8)$$

Similarly 2nd order derivative $f(x, y)$ is given by

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \qquad (9)$$

Example 5: Find first and second derivative of given data $f(x): f(x) = [4 \quad 3 \quad 2 \quad 5 \quad 9]$.

Solution: $\frac{\partial f}{\partial x} = [3-4, 2-3, 5-2, 9-5]$

$= [-1, -1, -3, 4]$

$\frac{\partial^2 f}{\partial^2 x} = [4+2-6, 3+5-4, 9+2-10]$

$= [0, 4, 1]$

***

With respect to the first and second derivatives, following things are observed

i) In flat regions (region of constant intensity levels) both derivatives produce zero.

ii) In ramp (constant slope), first derivatives gives a non-zero value, whereas second derivative gives non zero values (with different sign) only at the beginning and end of ramp and zero along the ramp. First derivative produces a thick edge called '**double edge effect**' along the ramp as the output is non zero. Second derivative produces a double edge one pixel thick separated by zeros.

iii) For a noisy pixel, first derivative produces a positive or negative value leading to one zero crossing. Whereas second derivative produces two zero crossing making it easier to identify.

iv) In case of an edge (step change in intensity values), first derivative produces a sign change leading to a zero crossing. This makes edge detection easier by second derivative. Second derivative enhances the details and edges much better as compared to first derivative. Thus for image sharpening applications, second derivative is most suitable.

Here, we will discuss some sharpening filters base on first derivatives and second derivatives.

4.4.1 First Derivatives Filters

Gradient Operator

An edge is the boundary between two regions with distinct grey level properties. Derivative operators are used for most edge detection techniques. The magnitude of first derivative calculated within a neighborhood around the pixel of interest is used to detect presence of edge in an image. First derivatives are directional operators and is defined as

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \tag{10}$$

Magnitude of this vector is given by

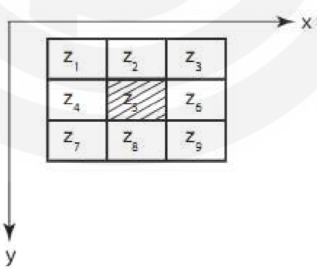
$$\nabla f = \text{mag}(\nabla f(x,y)) = \sqrt{g_x^2 + g_y^2} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right] \tag{11}$$

In common practice, gradient with absolute values are simpler to implement. Thus

$$\nabla f = |g_x| + |g_y| = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right| \tag{12}$$

Robert Operator

Consider the a pixel of interest $f(x,y) = z_5$ and a rectangular neighborhood of size 3 × 3 in Fig. 12 (a).



$$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline -1 & 0 \\ \hline \end{array} \qquad \begin{array}{|c|c|}\hline 1 & -1 \\ \hline 0 & 0 \\ \hline \end{array}$$

(a) Robert Operator **(b)**

$$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \qquad \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$$

(c)

Fig. 12

Equation can be approximated at point z_5 in a number of ways. Simplest is

$$g_x = z_5 - z_8 \tag{13}$$

$$g_y = z_5 - z_6 \tag{14}$$

$$\nabla f = |z_5 - z_8| + |z_5 - z_6| \tag{15}$$

Another approach is to use cross difference to realized

$$\nabla f \cong |z_5 - z_9| + |z_6 - z_8| \tag{16}$$

Equations can be implemented by the masks in Fig. 12 (a) and Fig. 12 (c). The original image is convolved with both the masks separately and the absolute values of the two outputs of convolution are added.

Although Roberts operator illustrates the derivative process, there is a difficulty in implementation. Since only two neighboring pixels are used, it is not clear as to exactly where the derivative value should be stored in the output image. This difficulty is taken care of in other filters like Prewitt and Sobel filters which are symmetric in nature.

Prewitt Operator

In the Prewitt Operator, the edge value at $f(x, y)$ is calculated by taking the -difference of $f(x+1, y)$ and $f(x-1, y)$. To make sure that the difference is not coming from two arbitrary points, three pairs of points are taken together to result in one edge value. This creates a 3×3 mask.

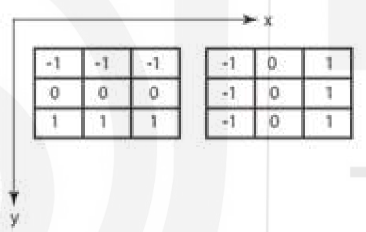


Fig. 13: Perwitt Operator

The above mask in Fig. 13 can be used to identify the presence of a vertical edge in the following image:

30	30	30	80	80	80
30	30	30	80	80	80
30	30	30	80	80	80
30	30	30	80	80	80
30	30	30	80	80	80
30	30	30	80	80	80

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

Sobel Operator

Another sharpening first order filter is the Sobel Operator. It is slightly different from the Prewitt filter, in that the middle pair of pixels are given higher weight compared to other two pairs of pixels. Both the filters are also called EDGE FILTERS. Sobel is the most popular 3×3 Edge operator. It is described by

$$\Delta f = |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

Sobel mask is shown in Fig. 14. Notice that all mask coefficients sum up to zero, thus giving no response in the area of constant intensity.

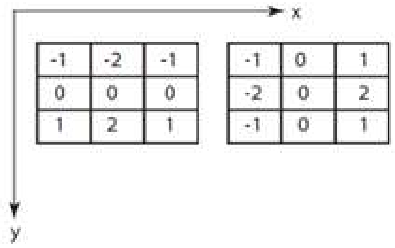


Fig. 14: Sobel Operator

Sobel operator and Prewitt operator are similar to each other. Sobel operator is also used to detect two kinds of edges in an image; vertical and horizontal.

You can observe from the masks given in Fig. 14, that first masks has only one difference in the members of first and third rows that is of sign, that is it has "2" and "–2" values in center of first and third rows. When applied on an image this mask will highlight the horizontal edges. Similarly, the second mask in Fig. 14 shows the vertical edges.

When we apply this mask on the image it prominent vertical edges. It simply works like as first order derivative and calculates the difference of pixel intensities in an edge region.

As the center column is of zero so it does not include the original values of an image but rather it calculates the difference of right

Now, let us discuss the filters based on second derivatives.

4.4.2 Laplacian Operator

For a 2D function $f(x, y)$, the gradient (first derivative) is defined as

$$\Delta f = \frac{\partial f(x,y)}{\partial x\, \partial y} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}$$

Laplacian (second derivative) is a **rotation invariant** and **linear** operator and it is defined as

$$\Delta^2 f = \frac{\partial^2 f(x,y)}{\partial x^2 \partial y^2} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial f^2}{\partial y^2} \qquad (17)$$

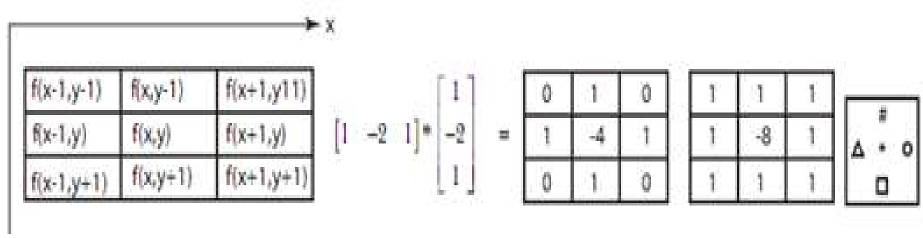
$$\frac{\partial^2 f}{\partial x^2} = f(x-1,y) + 2f(x,y) + f(x+1,y) = [1 \quad -2 \quad 1]$$

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

Thus $\Delta^2 f = [f(x+1,y) + f(x-1,y) - 2f(x,y)] + [f(x,y+1), f(x,y-1) - 2f(x,y)$

$$\Delta^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

O Δ □ # *



(a) (b) (c) (d)

$$\begin{array}{|ccc|}\hline -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \\ \hline\end{array} \qquad \begin{array}{|ccc|}\hline 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \\ \hline\end{array} \qquad \begin{array}{|ccc|}\hline 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \\ \hline\end{array}$$

(e) **(f)** **(g)**

Fig. 15: Masks for laplacian operator

Fig. 15 (b) shows a 3×3 sharpening filter mask. Fig. 15 (a) shows the coordinate notations and Fig. 15 (d) shows the location of non zero elements. Notice that **addition of all the coefficients of the mask produces zero**. In this mask, diagonal elements are not included. Fig. 15 (c) shows a mask with diagonal elements also. Fig. 15 (e) shows a weighted Laplacian mask. Notice that the centre element is negative. Fig. 15 (f) and Fig. 15 (g) show two Laplacian masks with centre element positive which gives the same results. Laplacian operation is a derivative operation which highlights the intensity discontinuities of an image and de-emphasize region with slowly varying intensity values. This tends to produce image that have greyish edge lines with dark featureless background. (Fig.16). Background features can be '**recovered**' while '**preserving**' the sharpened effect of Laplacian by subtracting Laplacian from the image. Thus, sharpened imaged is generated by the following operation.

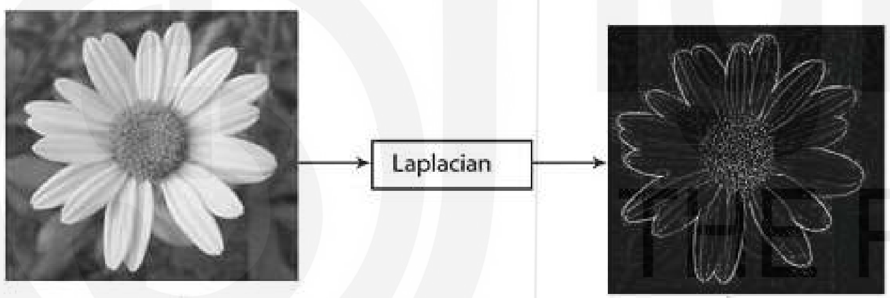


Fig 16: Output after laplacian operator

$$\begin{aligned} g(x,y) &= f(x,y) - \Delta^2 f(x,y) \\ &= f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)] \\ &= 5f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1) \end{aligned}$$

A mask is generated for sharpening of images.

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline\end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array} - \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array}$$

Similarly

$$\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline\end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array} - \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & -8 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

These are high frequency enhancement filters that **pass all frequencies and enhances high frequencies** . Notice that all the coefficients in the laplacian mask add up to one, so convolution with this mask will not change the pixel values in the part of the image that has constant grey levels. But if the pixels have higher or low grey level values than its neighbours, this contrast will be enlarged by the filter.

Example 5: For the image segment $f(x,y)$ given below. Apply laplacian filter of size 3×3 and 5×5 on the centre pixel.

Solution: Laplacian 4-connectivity mask is

0	1	0
1	−4	1
0	1	0

Thus, for Laplacian filter,

$$g_9(2,2) = [1+2+5+6-4\times7] = -14$$

Laplacian 8-connectivity mask is

1	1	1
1	−8	1
1	1	1

Thus for Laplacian filter

$$g_{10}(2,2) = [3+1+2+2+5+0+6+5-8\times7] = -22$$

***

Try these exercises.

E8) How are various filter mask generated to sharpen images in spatial filtering?

E9) Explain spatial filtering in image enhancement.

E10) What is meant by Laplacian filter? Using the second derivative, develop a Laplacian mask for image sharpening

E11) What are image sharpening filters? Explain the various types of q sharpening filters.

E12) Name the different types of derivative filters.

E13) Suggest typical derivative masks for Image enhancement. a) Roberts b) Prewitt c) Sobel.

E14) Write the applications of sharpening filters.

4.5 HISTOGRAM PROCESSING

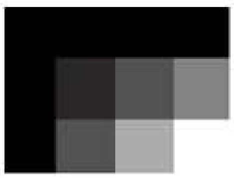
Before we begin with histogram processing, let us define what is the histogram of an image?

Histogram

Histogram of an image represents the number of times a particular grey level has occurred in an image. It is a graph in which the x-axis represents gray levels, and y -axis represents number of pixels for each grey level.

Histogram of an image is defined as $h(r_k) = n_k$, where $r_k = k^{th}$ grey level $n_k =$ number of pixels with grey level r_k, and k takes on the values $0,1,2,...,L-1$.

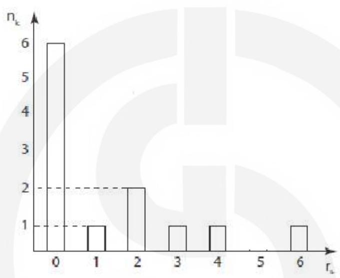
Example 1: Find histogram of image in Fig. 1.



0	0	0	0
0	1	2	3
0	2	4	6

Fig. 1: Image

Solution: A table is generated with r_k and n_k, where r_k contains all possible grey level values 0,1, 2, 3, 4, 5, 6 and n_k consists the number of times that grey value has occurred in the Fig. 1. 0 has occurred 6 times, 1 has occurred 1 time etc. Then histogram is drawn in Fig. 2.



r_k	0	1	2	3	4	5	6
n_k	6	1	2	1	1	0	1

Fig. 2: Histogram

***

Normalized histogram is obtained by dividing the frequency of occurrence of each gray level r_k by the total number of pixels in an image.

$p(r_k) = \frac{n_k}{n}$, where $k = 0........L-1$, where $n =$ total number of pixels in the image $p(r_k)$ gives the probability of occurrence of grey level r_k. The sum of all components of a normalized histogram is equal to 1.

Example 2: Find normalized histogram of the image

0	0	0	0
0	1	2	3
0	2	4	6

Solution: Fig. 2 image is similar to Example 1. A table is made with r_k, n_k and $p(r_k)$ and normalized histogram is drawn in Fig. 3.

r_k	1	1	2	3	4	5	6
n_k	6	1	2	1	1	0	1
$P(r_k)$	$\frac{6}{12}$	$\frac{1}{12}$	$\frac{2}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	0	$\frac{1}{12}$

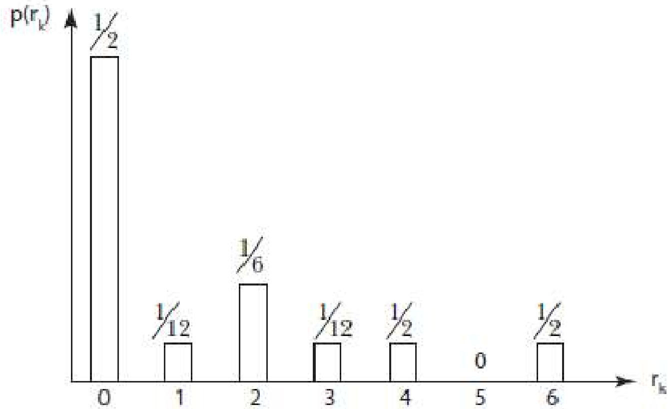


Fig. 3: Normalized Histogram

Now, you might be thinking that why histogram is needed. We shall answer this now.

Histogram gives an insight about the contrast in an image. It tells us the difference between average grey level of an object and that of the surroundings. It also provides a useful image statistics which are helpful in various image processing applications, for example, thresholding, intensity level slicing, segmentation etc. Besides it is very simple to calculate histogram of an image. Intuitively, it tells how vivid or washed out an image appears.

Based on histogram, we can categorize images in the four categories, which are defined below.

i) **Under exposed image:** Fig. 4 (a) shows an under exposed image, and its histogram is shown in Fig. 4 (b). It may be noted that almost all the pixels are concentrated at the lower end of the histogram. The maximum value of grey is less than 50. No details are seen in the image.

ii) **Over-exposed image:** Fig. 4 (c) shows an over-exposed image and its histogram is shown in Fig. 4 (d). The histogram is prominent towards the higher side of the grey values with the lowest grey level value around 150. This image has a washed-out look because of the absence of darker shades of grey.

iii) **Low contrast image:** Fig. 4 (e) shows a low contrast image and its histogram in Fig. 4 (f). The shape of the histogram is narrow and centered towards the middle of the grey scale. Very few grey levels participate in the image formation as neither the lower grey levels nor the higher grey levels are present. Thus, details are not visible in the image.

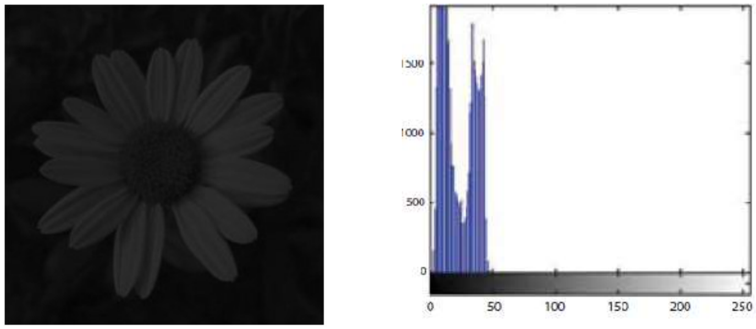
iv) **High contrast image:** Fig. 4 (g) shows a high contrast image and its histogram in Fig. 4 (h). This histogram covers a broad range of grey levels, but distribution of pixels is not uniform. The number of some grey levels is very high as compared to the number of other grey levels.

You can visualize histograms of different images when you will be doing your scilab sessions as given in Block 5 (Lab Manual). Now let us discuss various applications of histogram.

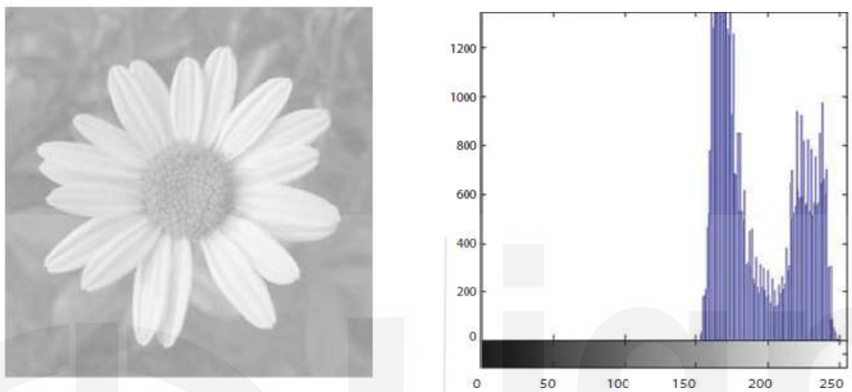
Applications of Histogram

Histogram is widely used in image processing applications.

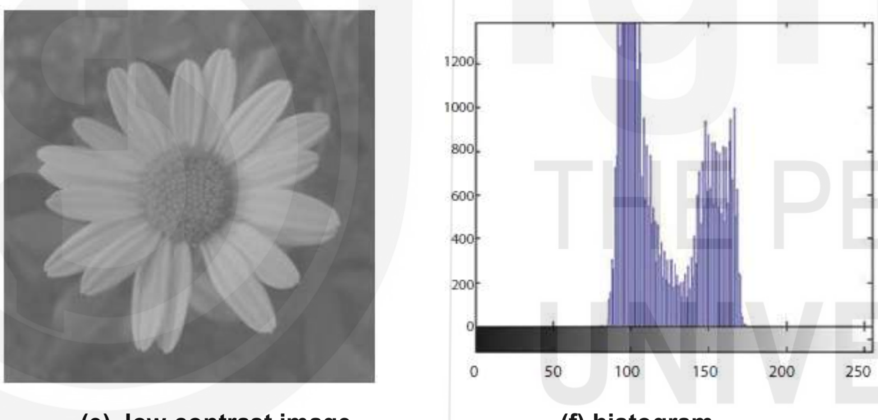
i) Histogram gives a quick indication as to whether or not we have used the entire dynamic range of digitizer. Information is lost if dynamic range is not used fully.



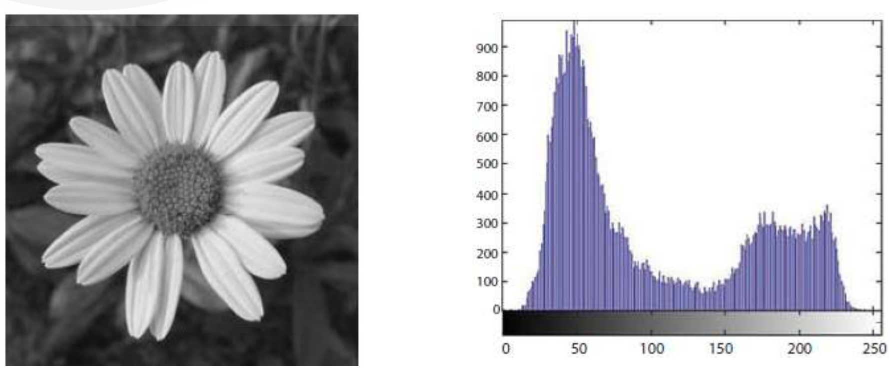
(a) under exposed image (b) histogram



(c) over-exposed image (d) histogram



(e) low contrast image (f) histogram



(g) high contrast image (h) histogram

Fig. 4

ii) Histogram indicates if clipping is a problem in the image.
iii) It is extensively used in thresholding to separate contrasting objects from background.
iv) It tells us about image contrast.

Now, try the following exercises.

E15) What is a histogram?

E16) Highlight the importance of histograms in image processing.

In the following section, we shall discuss histogram equalization.

4.6 HISTOGRAM EQUALIZATION

The histogram of a poor quality image would show presence of only few grey levels, while most would be missing. The number of pixels for various grey levels vary widely. Histogram equalization is a point operation that maps an input image onto a new output image such that there are (almost) equal numbers of pixels at each grey level in output. Thus it can be used for contrast enhancement.

Given information: Input image $f(x, y)$ from which we can calculate its histogram $h(r_k)$.

Goal: To obtain a uniform histogram for the output image. To devise a point operation $s = T(r)$ that maps input image $f(x, y)$ to an output image $g(x, y)$ that has a flat histogram as shown in Fig. 5.

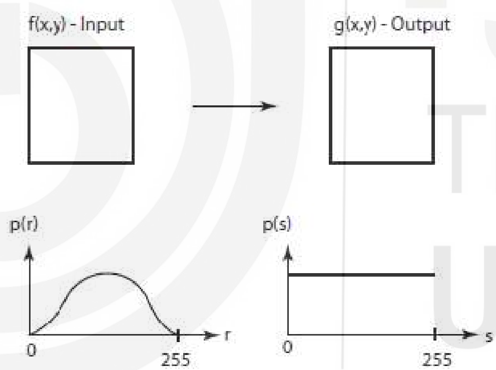


Fig. 5: Histogram Equalization

This is done by adjusting the probability density function (PDF) of the histogram of original image so that probability density function (PDF) spreads equally. We want to find a transformation of the image given in Fig. 6 (a) by the following function:

$$s = T(r), 0 \le r \le 1 \quad \dots (1)$$

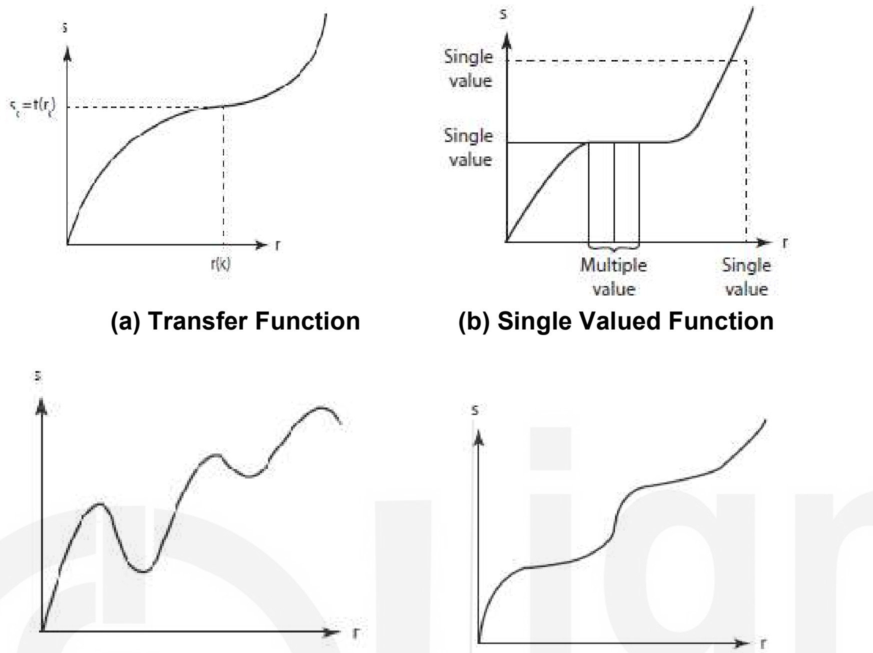
which should satisfy following conditions:

1) $T(r)$ is a single valued function (one to one relationship) as shown in Fig. 6 (b). For each value of r there should be a single value of s. This is needed so that inverse transform exists.

2) $T(r)$ is a monotonically increasing function in the interval $0 \le r \le 1$, Amonotonically increasing function is one which keeps on growing (once it goes up it does not) comedown. This preserves intensity level order in the output image from black to white. This transformation doesn't cause a

negative image. Fig. 6 (c) shows a non-monotonical function and Fig. 6 (d) shows a monotonically increasing function.

3) The range of r and s is same, that is for $0 \le r \le 1, 0 \le T(r) \le 1$. This guarantees that the output grey levels are in the same range as input grey levels.



(a) Transfer Function **(b) Single Valued Function**

(c) Non Monotonical Function **(d) Monotonical Increasing Function**

Fig. 6

Inverse transformation of the function given in Eqn. (1) is defined as $r = T^{-1}(s) 0 \le s \le 1$.

In practice, for integer values of images, many times condition 1 is violated and non-unique inverse transformation is generated.

Probability Density Function (PDF)

The grey levels in an image may be viewed as random variables in the interval $[0,1]$. Probability density function (PDF) is one of the fundamental descriptors of a random variable. PDF of a random variable x is defined as the derivative of cumulative distribution function (CDF)

$$p(x) = \frac{dF(x)}{dx}$$

$F(x) = P(X \le x)$

Where $p(x) =$ PDF of x

$F(x) =$ CDF of x

$p =$ Probability of x

PDF satisfies the following properties:

i) $P(x) \ge 0$ for all x

ii) $\int_{-\infty}^{\infty} p(x)\,dx = 1$

iii) $\int_{-\infty}^{x} p(x)\,d\alpha$ where α is a dummy variable

iv) $p(x_1 \le x \le x_2) = \int_{-x_1}^{x_2} p(x)\,dx$

If a random variable x is transformed by a monotonic transfer function $T(x)$ to produce a new random variable y, then the PDF of y can be obtained from knowledge of $T(r)$ and PDF of x

$$p_y(y) = p_x(x)\left|\frac{dx}{dy}\right|$$

Let $p_r(r)$ be PDF of random variable r of input image.
$P_s(s)$ be PDF of random variables r of output image.
If $p_r(r)$ and $T(r)$ is know then $P_s(s)$ can be obtained by

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

The PDF of transformed variables s is determined by pdf of input image and by chosen transformation function.

Let us now discuss the discrete transformation function.

Transformation function is a cumulative distribution function (CDF) of r.

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

Where w is dummy variable.

CDF is an integral of a probability function (always positive). Thus CDF is always single valued and monotonically increasing function. CDF satisfies the conditions of transformation function, hence can be considered as $T()$.
Differentiating above equation with respect to r, we get

$$\frac{ds}{dr} = \frac{dt(r)}{dr} = \frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right]$$

$$= p_r(r)$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{p_r(r)}\right| = 1$$

Thus, $\boxed{p_s(s) = 1}$, where $0 \le s \le 1$

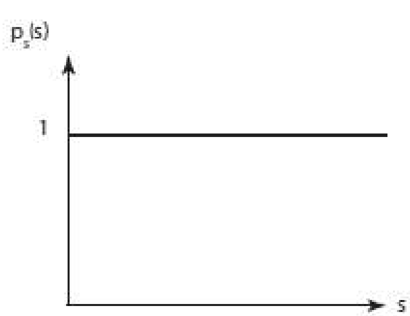


Fig. 7

$p_s(s)$ is PDF of output image, whose value is 1 for all values of s. **Thus $p_s(s)$ is a uniform PDF independent to the form of $p_r(r)$ as shown in Fig. 7.**

Discrete Transformation Function

Although the above discussion shows that any image can be transformed to a flat histogram, it is only possible if grey values are varying continuously. In practice any image will have discrete grey values. Thus the integration above is replaced by a summation function. This means that the output image will not have a flat histogram but only an approximation to it.

Probability of occurrence of grey levels in an image given by

$p_r(r_k) = \frac{n_k}{n}$, where $k = 0............L-1$.

Transformation is given by

$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$, where $k = 0,....L-1$.

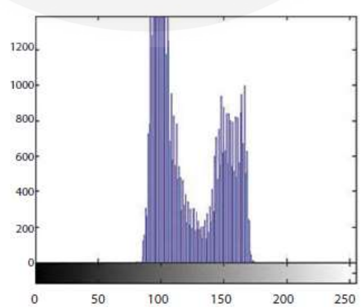
Substituting value of $p_r(r_k)$, we get

$r_k = \sum_{j=0}^{k} \frac{n_j}{n}$

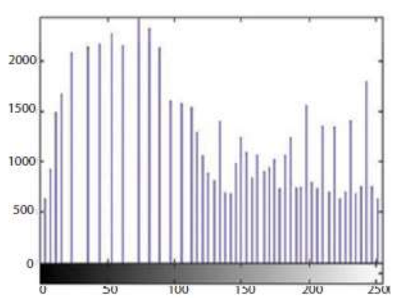
The output image is obtained by mapping each pixel with grey level r_k in the input image to a corresponding pixel with grey level s_k in output image.

(a) Low Contrast Image

(b) Histogram Equalized Image



(c) Histogram of (a)



(d) Histogram of (b)

Fig. 8

Example 3: For the given 4×4 image having grey scales between $[0,9]$, carry out histogram equalization. Also, draw the histogram of image before and after equalization.

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

Solution: Given is

a) The grey levels are between $[0,9]$.
b) Size of image is 4×4.

Following steps are to be followed for histogram equalization:

Step 1: Find the histogram of grey levels of input image by making a table of r_k (grey levels) and number of pixels (n_k) in the input image.

Step 2: For each input grey level, compute the cumulative $\sum_{j=0}^{k} n_j$ sum by adding number of pixels for each grey level.

Step 3: Divide the cumulative sum by the total number of pixels in the image (4 × 4 = 16) to generate the CDF of the given image.

Step 4: Scale the output by multiplying step 3 to maximum grey level value (9 in this case)

Step 5: Round off the output of step 4 to the nearest integer value

We have summarized step 1 to step 5 in the following table.

Grey pixel r_j		0	1	2	3	4	5	6	7	8	9
Step 1	No. of pixels n_j	0	0	6	5	4	1	0	0	0	0
Step 2	Commulative sum $\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
Step 3	$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	$\frac{6}{16}$	$\frac{11}{16}$	$\frac{15}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$
Step 4	$s \times 9$	0	0	$\frac{6}{16} \times 9$	$\frac{11}{16} \times 9$	$\frac{15}{16} \times 9$	$\frac{16}{16} \times 9$	$\frac{16}{16} \times 9$	$\frac{16}{16} \times 9$	$\frac{16}{16} \times 9$	$\frac{16}{16} \times 9$
Step 5	Round off s_k	0	0	≈ 3	≈ 6	≈ 8	≈ 9	9	9	9	9

Step 6: Map input levels (r_k) to output levels s_k in a tabular form from 1st and last row of table above to generate

r_k	0	1	2	3	4	5	6	7	8	9
s_k	0	0	3	6	9	9	9	9	9	9

Step 7: Formulate new image $g(x, y)$ by replacing each level of $f(x, y)$ by a new one using table above.

$$\underset{f(x,y)}{\begin{array}{|c|c|c|c|}\hline 2 & 3 & 3 & 2 \\ \hline 4 & 2 & 4 & 3 \\ \hline 3 & 2 & 3 & 5 \\ \hline 2 & 4 & 2 & 4 \\ \hline \end{array}} \xrightarrow{\text{equalized}} \underset{g(x,y)}{\begin{array}{|c|c|c|c|}\hline 3 & 6 & 6 & 3 \\ \hline 8 & 3 & 8 & 6 \\ \hline 6 & 3 & 6 & 9 \\ \hline 3 & 8 & 3 & 8 \\ \hline \end{array}}$$

Step 8: We draw output and input histogram as shown in Fig. 9.

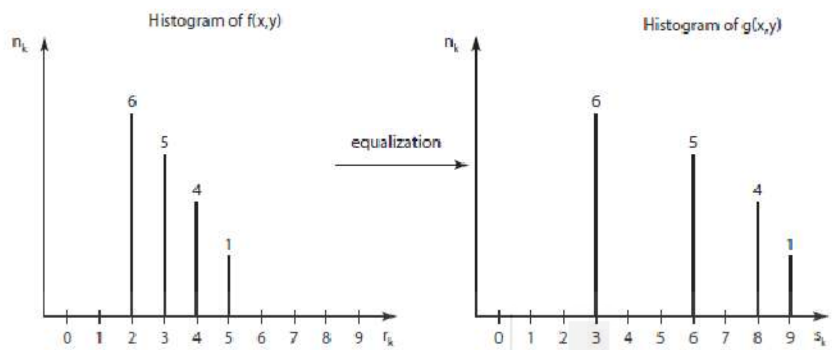


Fig. 9

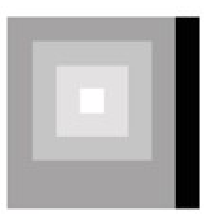
It may be noted that while in the original image the histogram spans between grey values 2 to 5, in the output image, the grey values span from 3 to 9. Although some equalization has been achieved, some of the grey values are still missing. It is because the image has only 9 levels. Histogram equalization will be very effective in an image with 256 grey levels.

Now, try the following exercises.

E17) Write steps for a procedure to perform histogram equalization.

E18) What is the role of histogram equalization in image enhancement? Why this technique yields a flat histogram?

E19) Equalize the histogram of 8 × 8 image $g(x, y)$ shown below. The input image has grey levels 0, 1, ..., 7.



$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 \\ 4 & 5 & 5 & 5 & 5 & 5 & 4 & 0 \\ 4 & 5 & 6 & 6 & 6 & 5 & 4 & 0 \\ 4 & 5 & 6 & 7 & 6 & 5 & 4 & 0 \\ 4 & 5 & 6 & 6 & 6 & 5 & 4 & 0 \\ 4 & 5 & 5 & 5 & 5 & 5 & 4 & 0 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 \end{bmatrix}$$

In the next section we shall see how to find transformation for an image, such that the new histogram follows a specific shape. It is called Histogram Specification.

4.7 HISTOGRAM SPECIFICATION

We shall begin with the definition of histogram specification.
Definition: Histogram specification is a point operation that maps input image $f(x, y)$ into an output image $g(x, y)$ with a user specified histogram. An example of specified histogram is shown in Fig. 9.

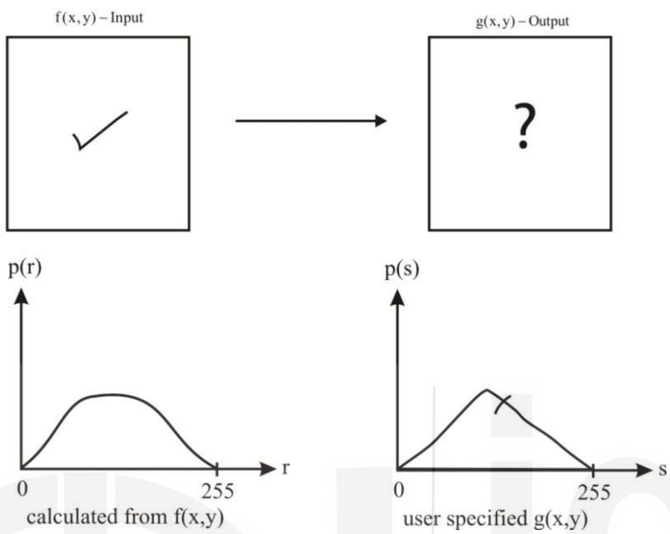


Fig. 9: Histogram specification

The main application of histogram specification is that it improves contrast and brightness of images. It is a pre processing step in comparison of images.

Comparison to histogram equalization

Histogram equalization has a disadvantage that it can generates only one type of output image where histogram is flat, which may not be the best approach under all circumstances. Histogram specification is more general than equalization. It is an interactive enhancement technique where user can draw desired histogram. We can specify the shape of histogram which need not be uniform. Thus, histogram specification gives us the flexibility to choose histogram for reference and output image is mapped accordingly. The basic idea is to carry out histogram equalization of input image histogram, then carry out histogram equalization of specified histogram. The two transformations are used to find the required transformation.

Algorithm for histogram equalization: In this procedure, there are three variables:

$P_r(r)$ is pdf of grey level r of input image
$P_z(z)$ is pdf of grey level z of specified image
$P_s(s)$ is pdf of grey level s of output image

The transformation is $s = T(r) = \int_0^r p_r(r)\,dr$

Histogram equalization of input image $G(z) = \int_0^s p_z(z)\,dz$

To find the histogram equalization of specified image, we equate $G(z)$ is equated to s and an inverse transformation is computed as given below:

$G(z) = s = T(r)$

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

Assuming that G^{-1} exists, we can map input grey levels r to output grey levels s.

Procedure for histogram specification

Step 1: Obtain the transformation $T(r)$ by doing histogram equalization of input image.

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

Step 2: Obtain the transformation $G(z)$ by doing histogram equalization of specified image.

Step 3: $G(z) = \int_0^z p_z(z)\,dz$

Equate $G(z) = s = T(r)$

Step 4: Obtain inverse transformation function G^{-1}.

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

Step 5: Obtain the output image by applying inverse transformation function to all pixels of input image.

Let us now apply these steps in the following example.

Example 4: Assume an image having given grey level PDF $P_r(r)$. Apply histogram specification with given desired PDF function $P_z(z)$ given below

$$p_r(r) = \begin{cases} -2r+1; & 0 \le r \le 1 \\ 0\;; & \text{otherwise} \end{cases} \qquad p_s(r) = \begin{cases} 2z; & 0 \le z \le 1 \\ 0; & \text{otherwise} \end{cases}$$

Apply histogram specification with the desired PDF function $p_z(z)$ given in Fig. 10:

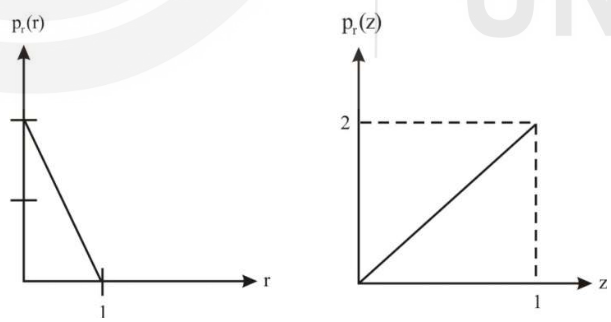


Fig. 10

Solution: We shall solve it step wise.

Step 1: Obtain transformation function $T(r)$ by doing histogram equalization of input image.

$$s = T(r) = \int_0^r p_r(r)\,dr = \int_0^r (-2r+2)\,dr - [-r^2 + 2r]_0^r = -r^2 + 2r$$

Step 2: Obtain transformation function $G(z)$

$$G(z) = \int_0^z p_z(z)\,dz = \int_0^z 2z\,dz = [z^2]_0^z = z^2$$

Step 3: Equate

$$s = T(r) = G(z)$$
$$-r^2 + 2r = z^2$$

Step 4: Obtain inverse transformation G^{-1}.

$$z = G^{-1}[T(r)]$$
$$z = \sqrt{-r^2 + 2r}$$

***

Now, we shall find the discrete transformation function for an actual image.

Discrete Transformation Function for an actual Image

Histogram equalization of input image.

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

where $n =$ total number of pixels in input image
$n_j =$ number of pixels having level j

Histogram equalization of specified image

$$v_q = G(z_q) = \sum_{i=0}^{q} p_z(z_i), q = 0, .., L-1$$

Equate $G(z_q) = s_k = T(r_k)$

Inverse transformation $z_q = G^{-1}[s_k] = G^{-1}[T(r_k)]$

The block diagram of histogram specification is given in Fig 11.

v^* is that value of v_q so that min $[v_q - s] \geq 0$
$p =$ Corresponding value of z.

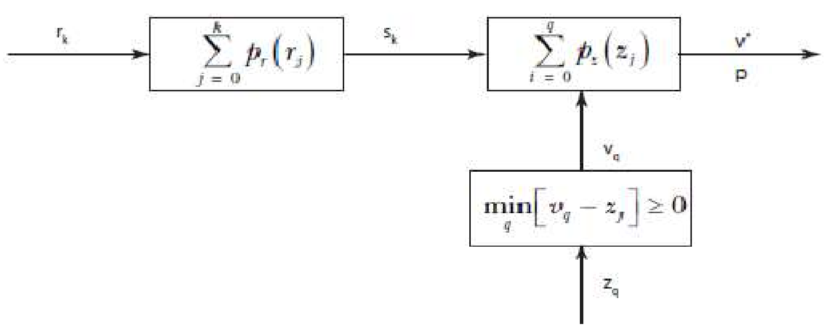


Fig. 11: Histogram Specification

Procedure for histogram specification of an image

Step 1: Equalize input image histogram (s_k)
Step 2: Equalize specified image histogram (v_q)
Step 3: For $\min q[v_q - s] \geq 0$ find corresponding v^* and p.
Step 4: Map input pixels to output pixels to get output image.

We shall apply these steps in the following example.

Example 6: Apply histogram specification on given image having

$$\begin{array}{|cccc|} 0 & 1 & 0 & 2 \\ 2 & 3 & 3 & 2 \\ 3 & 1 & 0 & 1 \\ 1 & 3 & 2 & 0 \end{array}$$

$$r_i = z_i = 0,1,2,3$$
$$p_r(r_i) = .25 \text{ for } i = 0,1,2,3$$
$$p_z(z_0) = 0, p_z(z_1) = .5,\ p_z(z_2) = .5, p_z(z_3) = 0$$

Solution: We shall apply each step of the procedure.

Step 1: Equalize input image histogram.

r_k	0	1	2	3
$p_r(r_k)$	.25	.25	.25	.25
s_k	.25	.5	.75	1

Step 2: Equalize specified image histogram

z_q	0	1	2	3
$p_z(r_z)$	0	.5	.5	0
v_q	0	.5	1	1

Step 3: Find minimum value of q such that $(v_q - s) \geq 0$. First three columns are filled by step 1, next three columns are filled by step 2. In this step, last two columns are filled by the fallowing procedure.

r_k	$p_r(r_k)$	s_k	z_q	$p_z(z_q)$	v_q	v^4	p
0	.25	.25	0	0	0	.5	1
1	.25	.5	1	.5	.5	.5	1
2	.25	.75	2	.5	1	1	2
3	.25	1	3	0	1	2	2

a) $q = 0, k = 0 \quad (v_0 - s_0) = (0 - .25) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increase} q$
$q = 1, k = 0 \quad (v_1 - S_0) = (.5 - .25) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_0 = v_1 = .5$ and $p_0 = z_1 = 1$

b) $q = 1, k = 1 \quad (v_1 - s_1) = (.5 - .5) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_1 = v_1 = .5$ and $p_1 = z_1 = 1$

c) $q = 1, k = 2 \quad (v_1 - s_2) = (0 - .75) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increase} q$
$q = 2, k = 2 \quad (v_2 - s_2) = (1 - .75) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_2 = v_2 = 1$ and $p_2 = z_2 = 2$

d) $q = 2, k = 3 \quad (v_2 - s_2) = (1 - .75) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_3 = v_2 = 1$ and $p_3 = z_3 = 2$

Step 4: Map input level values to output level values

r_k	0	1	2	3
p	1	1	2	2

Step 5: Map input pixels to new values to get new image using step 4 output.

$$\begin{array}{|c|c|c|c|}\hline 0&1&0&2\\\hline 2&3&3&2\\\hline 0&1&0&1\\\hline 1&3&2&0\\\hline\end{array} \Longrightarrow \begin{array}{|c|c|c|c|}\hline 1&1&1&2\\\hline 2&2&2&2\\\hline 2&1&1&1\\\hline 1&2&2&1\\\hline\end{array}$$

$f(x, y)$ $\qquad$ $g(x, y)$

This pixel mapping is used to map the pixels of input image to generate output image with specified histogram. Histogram specification is a trial and error process. There are no rules for specifying histogram and one must resort to analysis on a case by case basis for any given enhancement task.

***

Try the following exercises.

E20) Define the concept of Histogram matching with appropriate example.

E21) Perform histogram specification from the given data.

$$r = 0,1,2,3,4$$
$$p(r_i) = 0.2 \text{ for } i = 0.......4$$
$$z_i = 0,2,4,5,7$$
$$p(z_0) = 0,\ p(z_2) = 0.2,\ p(z_4) = 0.4,\ p(z_5) = 0.4,$$
$$p(z_7) = 0$$

Now let us, summarise what we have discussed in this unit.

4.8 SUMMARY

In this unit, we have discussed the following points.

1) Filtering is a process that removes some unwanted components or small details in an image. In digital image processing, filter is basically a subimage or a mask or kernel or template or window. Filters can be of two types: Spatial filters and frequency domain filters.

2) A filter that passes low frequencies is called a lowpass filter. This type of filter is used to blur (smooth) the image, therefore called smoothing filter also. These filters are also used for noise reduction. Noise reduction can be done by blurring with a linear filter, (in which operation performed on the image pixels is linear) or a non-linear filter.

3) This filter is used to highlight transitions in intensity. These are based on first and second derivatives.

4) Linear filtering is a spatial domain process where a filter (mask/ kernel/ template) with some integer coefficient values is applied to input image to generate the filtered/ output image.

5) Each pixel in the smoothened image $g(x, y)$ is obtained from the average pixel value in the neighborhood of (x, y) in input image. Such a mask is also known as **a Mean filter.**

6) Like all other spatial filters, non linear filters compute the result at some position (x, y) from the pixels inside the moving region S of the original image. These filters are called **non-linear** because source pixels are processed by some non-linear function.

7) Median filters are edge preserving smoothing filters, where the level is set to the median of pixel values in the neighborhood of that pixel.

8) **Min filter** removes salt noise (white dots with large grey level values) because any large grey level with in a 3×3 filter region is replaced by one of its surrounding pixels with smallest value. As a side effect, min filter introduces dark structures in the image. The reverse effect is expected from a **max filter**. It removes pepper noise (black dots with small grey level values) because any black dot within 3×3 filter region is replaced by one of its surrounding pixels with the largest value. White dots/bright structures are widened as a side effect and black dots (pepper noise) will disappear.

9) Image sharpening is opposite of image smoothing. This is done to highlight fine details and edges in an image.

10) Defined histogram

11) Understood how to draw histogram from the image

12) Perform histogram equalization

13) Perform histogram specification

4.9 SOLUTIONS/ANSWERS

E1) Linear filtering is a spatial domain process where a filter (mask/kernel/template/window) with some coefficient values is applied to input image to generate the filtered/ output image. Generally, filter size is either 3×3 or $5 \times 5, 7 \times 7$ or 21×21 (odd sizes) and filter is centered at a coordinate (x, y) called ' Hot Spot'.

E2) Smoothing linear filters are used to reduce 'sharp' transitions in grey levels, reduce noise, bluredges, help in smoothing false colours, reduce 'irrelevant' details in an image.

Sharpening linear filters are used to detect edges, lines, points in the image. It is also used to highlight all high frequency components in the image.

E3) Each pixel in the smoothened image $g(x, y)$ is obtained from the average pixel value in the neighborhood of (x, y) in input image.

$$g(x, y) = \frac{1}{M} \sum_{(m,n)\in S_{xy}} f(m,n)w(m,n)$$

Various masks are $\frac{1}{9}\times\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}\frac{1}{16}\times\begin{bmatrix}1 & 2 & 1\\2 & 4 & 2\\1 & 2 & 1\end{bmatrix}$

Following steps are required for spatial averaging

1) Position the mask over the current pixel such that hotspot w(0,0) coincides with current pixel.
2) Form all products of filter elements with the corresponding elements in the neighbourhood.
3) Add up all the products and store it at current position in the output image. This must be repeated for every pixel in the image.

E4) The aim of image smoothing is to diminish the effect of camera noise, spurious pixel values, missing pixel values etc.

E5) Median filters are edge preserving smoothing filters, where the level is set to the median of pixel values in the neighborhood of that pixel. It is impossible to design a filter that removes only noise and retains all the important image structures intact, because no filter can discriminate which image content is important to the viewer and which is not. Median filter replaces every image pixel by median of the pixels in the corresponding filter region Sxy.

$$g(x, y) = \text{median } \{f(x+i, y+j) | (i, j) \in Sxy\}$$

E6) Max and min filters are defined as:

$$g(x, y) = \min\{f(x+i, y+j) | (i, j) \in S_{xy}\}$$
$$g(x, y) = \max\{f(x+i, y+j) | (i, j) \in S_{xy}\}$$

where S_{xy} denotes the filter region, usually a size of 3×3 pixels.

E7) Linear filters produce an output that is a linear combination of the input. For example, smoothing linear filters produce output as the average of input while sharpening linear filters produce output as the first or second derivative of the input.

Non linear filters produce a statistical output where median filter produce output as median of the input values, min filter produces output as minimum of all input values and max filter produces output that is maximum of all input values.

E8) $\Delta^2 f = \frac{\partial^2 f(x,y)}{\partial x^2 \partial y^2} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial f^2}{\partial y^2}$

E9)

$$\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Spatial filtering is one of the main tools used in variety of applications such as noise removal, bridging the gaps in object boundaries, sharpening of edges etc.The idea of spatial filtering is to move a 'mask', a rectangle (usually with size of odd length) or other shape over the image. By this process, we create a new image where grey level values of the pixels are calculated from the values under the mask. The values under the mask are modified by a function called 'filter'. If this filter function is a linear function of all grey level values in the mask, then filter is called a 'linear filter', else it is called 'non linear filter'.

E10) **Laplacian** (second derivative) is a **rotation invariant** and **linear** operator and it is defined as

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

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

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

Thus

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

O Δ □ # *

$$\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

E11) Image sharpening is done to highlight fine details and edges in an image. Applications of image sharpening are industrial applications, medical imaging etc. Sharpening is reverse of smoothing which is achieved by pixel averaging which is analogous to integration. Thus, sharpening is achieved by differentiation. The derivative operation enhances the degree of discontinuity in an image.

Sharpening

- Enhances edges and other discontinuities (noise) in an image.
- De-emphasizes area with slowly varying grey levels (background) in an image.

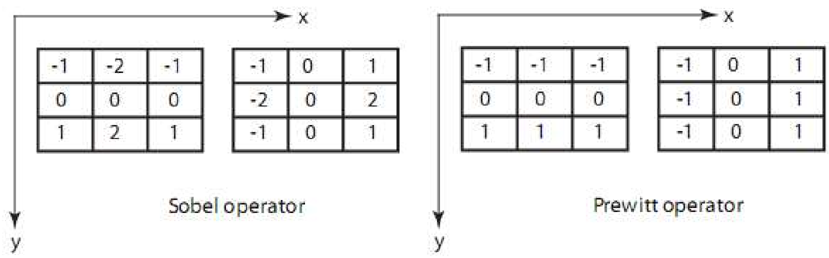
Different image sharpening filters are Laplacian, gradient, Robert, sobel, etc.

E12) For a 2D function $f(x, y)$, the gradient (first derivative) is defined as **Laplacian** (second derivative) is a **rotation invariant** and **linear** operator and it is defined as

$$\nabla f = \frac{\partial f(x,y)}{\partial x \partial y} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}$$

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

E13)



$$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline -1 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|}\hline 1 & -1 \\ \hline 0 & 0 \\ \hline \end{array} \qquad \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \quad \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$$

Robert operator

E14) Sharpening filters are used for edge enhancement applications. Sharpening linear filters are used to detect edges, lines, points in the image. It is also used to highlight all high frequency components in the image.

E15) Histogram is a graph between various grey levels on x-axis and the number of times a grey level has occurred in an image, on y-axis.

Histogram of an image is defined as

$$h(r_k) = \frac{nk}{k}, = 0............L-1$$

$r_k = k^{th}$ grey level

$n_k =$ number of birds grey level values as r_k.

E16) The histogram of an image represents the relative frequency of occurrence of the various gray levels in the image. It also provides a useful image statistics which are helpful in various image processing applications, for example, thresholding, intensity level slicing, segmentation etc. Besides it is very simple to calculate histogram. Intuitively, it tells how vivid or washed out an image appears. Histograms are the basis for many spatial domain processing techniques.

E17) Histogram equalization is done by the transformation

$$\boxed{s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)} \qquad ,k = 0,.......L-1$$

E18) Histogram equalization is a point operation that maps an input image onto output image such that there are equal number of pixels at each grey level in output. It is used for contrast enhancement.

Given information: Input image $f(x,y)$ from which we can calculate its histogram $h(r_k)$

Goal: To obtain a uniform histogram for the output image. To device a point operation $s = T(r)$ that maps input image $f(x, y)$ to an output image $g(x, y)$ that has a flat histogram

E19) Given size: 8×8, levels $[0\ 7]$

	r_j	0	1	2	3	4	5	6	7
Step 1	n_k	8	0	0	0	31	16	8	1
Step 2	$\sum_{j=0}^{k} n_j$	8	8	8	8	39	55	63	64
Step 3	$S_k = \sum_{j=0}^{k} \frac{n_j}{n}$	$\frac{8}{64}$	$\frac{8}{64}$	$\frac{8}{64}$	$\frac{8}{64}$	$\frac{39}{64}$	$\frac{55}{64}$	$\frac{63}{64}$	$\frac{64}{64}$
Step 4	$s_k \times 7$	$\frac{8}{64}\times 7$	$\frac{8}{64}\times 7$	$\frac{8}{64}\times 7$	$\frac{8}{64}\times 7$	$\frac{39}{64}\times 7$	$\frac{55}{64}\times 7$	$\frac{63}{64}\times 7$	$\frac{64}{64}\times 7$
Step 5	Round off	≈ 1	≈ 1	≈ 1	≈ 1	4	6	7	7

Step 1: Find the histogram of input image by making a table of r_k and n_k.

Step 2: For each input level, find cumulative sum

Step 3: Divide cumulative sum by total number of pixels $n = 64$.

Step 4: Scale the output by multiplying step 3 by 7.

Step 5: Round off to nearest integer value to find s_k.

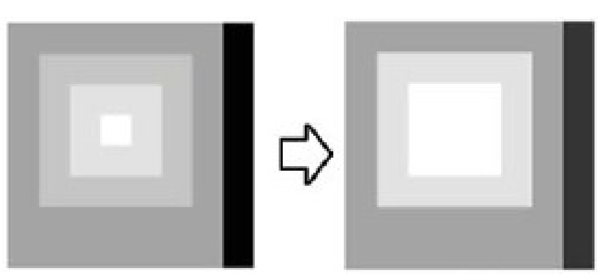
Step 6: Map input levels r_k to output levels s_k using 1st and last row of table above to make table below.

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

Step 7: Formulate a new image $g(x, y)$ by replacing each level of $f(x, y)$ by a new one using table.

Step 8: Draw input and output histograms.

$$
\underset{f(x,y)}{\begin{array}{cccccccc}
4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 \\
4 & 5 & 5 & 5 & 5 & 5 & 4 & 0 \\
4 & 5 & 6 & 6 & 6 & 5 & 4 & 0 \\
4 & 5 & 6 & 7 & 6 & 5 & 4 & 0 \\
4 & 5 & 6 & 6 & 6 & 5 & 4 & 0 \\
4 & 5 & 5 & 5 & 5 & 5 & 4 & 0 \\
4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 \\
4 & 4 & 4 & 4 & 4 & 4 & 4 & 0
\end{array}}
\Rightarrow
\underset{g(x,y)}{\begin{array}{cccccccc}
4 & 4 & 4 & 4 & 4 & 4 & 4 & 1 \\
4 & 6 & 6 & 6 & 6 & 6 & 4 & 1 \\
4 & 6 & 7 & 7 & 7 & 6 & 4 & 1 \\
4 & 6 & 7 & 7 & 7 & 6 & 4 & 1 \\
4 & 6 & 7 & 7 & 7 & 6 & 4 & 1 \\
4 & 6 & 6 & 6 & 6 & 6 & 4 & 1 \\
4 & 4 & 4 & 4 & 4 & 4 & 4 & 1 \\
4 & 4 & 4 & 4 & 4 & 4 & 4 & 1
\end{array}}
$$



f(x) g(x)

E20) Histogram specification is more general than equalization. It is an interactive enhancement technique where user can draw desired histogram. We can specify the shape of histogram which needs not to be uniform. Thus, histogram specification gives us the flexibility to choose histogram for reference and output image is mapped accordingly

E21) **Step 1:** Equalize input histogram.

r_k	0	1	2	3	4
$p(r_k)$	.2	.2	.2	.2	.2
s_k	.2	.4	.6	.8	1

Step 2: Equalize specified histogram

r_k	0	1	2	3	4
$p(r_k)$	.2	.2	.2	.2	.2
s_k	.2	.4	.6	.8	1

Step 3: Find minimum value of q such that $[vq - s] \geq 0$

1) $q = 0, k = 0 \quad (v_0 - s_0) = (0 - .2) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increase}\, q$

   $q = 1, k = 0 \quad (v_1 - s_0) = (.2 - .2) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_0 - v_1 = .2$ and $p_0 = z_1 = 2$

2) $q = 1, k = 1 \quad (v_1 - s_1) = (.2 - .4) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increase}\, q$

   $q = 2, k = 1 \quad (v_2 - s_2) = (.2 - .4) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_1 = v_2 = .6$ and $p_1 = z_2 = 4$

3) $q = 2, k = 2 \quad (v_2 - s_2) = (.6 - .6) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_2 = v_2 = .6$ and $p_2 = z_2 = 4$

4) $q = 3, k = 3 \quad (v_2 - s_2) = (.6 - .8) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increase}\, q$

   $q = 3, k = 3 \quad (v_q - s_q) = (1 - .8) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_q = v_q = 1$ and $p_q = z_q = 5$

5) $q - 3, k - 4 \quad (v_3 - s_4) = (1 - 1) \geq 0 \Rightarrow \text{Yes} \Rightarrow\Rightarrow v^*{}_4 = v_3 = 1$ and $p_4 = z_3 = 5$

S.No.	r_k	$p(r_k)$	s_k	z_q	$p(z_q)$	v_q	v^*	p
1	0	.2	.2	0	0	0	.2	2
2	1	.2	.4	2	.2	.2	.6	4
3	2	.2	.5	4	.4	.6	.6	4
4	3	.2	.8	5	.4	1	1	5
5	4	.2	1	7	0	1	1	5

Step 4: Input and output grey level mapping from 2nd and last row of previous output

r	0	1	2	3	4
p	2	4	4	5	5