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.

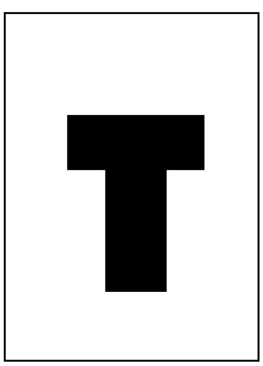
As we mentioned earlier an image is a projection of the environment under consideration, if the environment is 3D then its projection will be in 2D. If we are having an n-Dimensional space then its projection will be of n-1 dimensions. Magnetic resonance images and computerized tomography (CT) images, 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:

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

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.

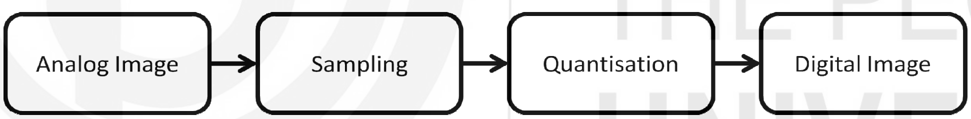


Fig. 2: Analog Image Processing

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.

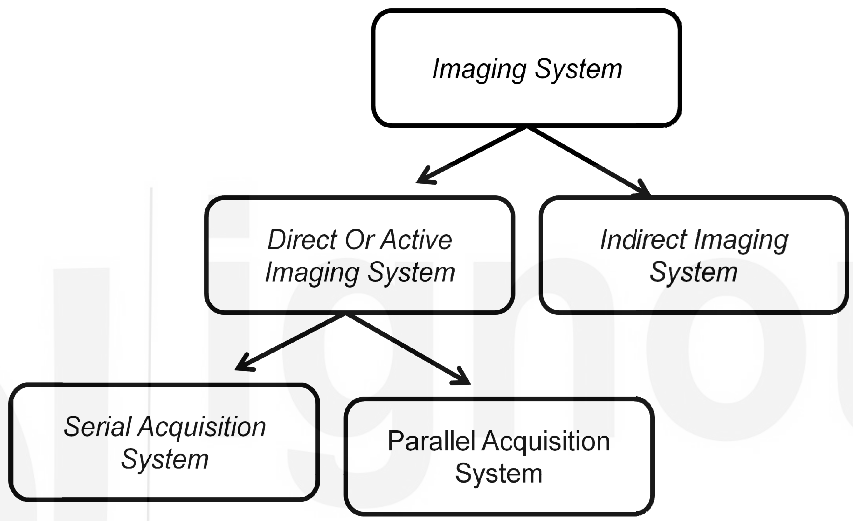


Fig. 3: Types of Imaging and Data Acquisition Systems

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.

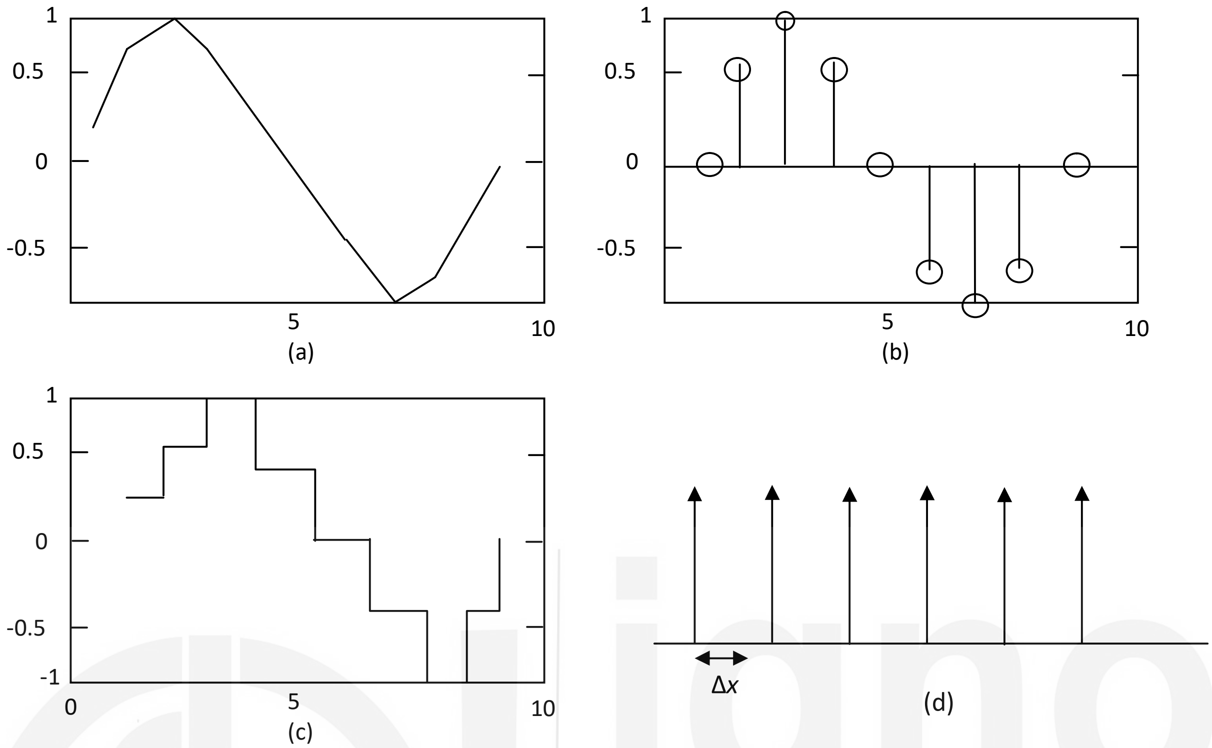


Fig. 4: **Sampling process (a) Original signal** $f(t)$ **(b) Sampled image** $f(n) = f(t)*r(t)$ **(c) Reconstructed image (d) Train of the impulse function** ***r*(t)**

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 $d \le \frac{1}{2f_{max}}$. 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 .

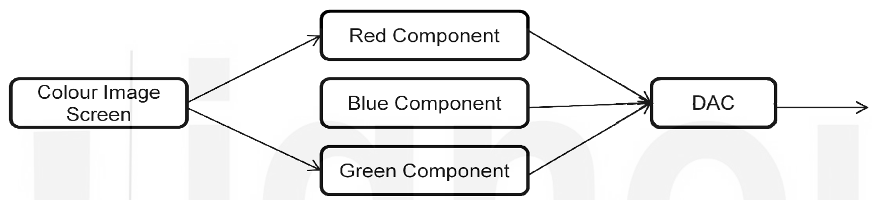


Fig. 5: True ColourImage

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.

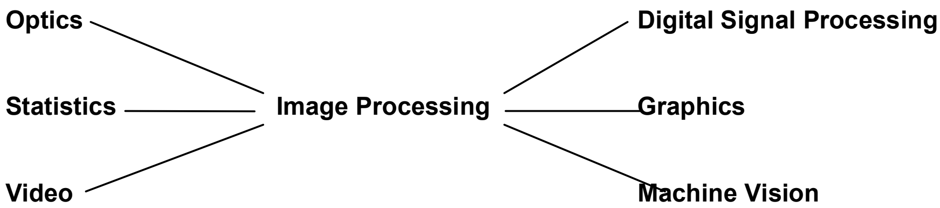


Fig. 11: Image processing and other closely related fields.

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, $f : D \rightarrow P$ 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.

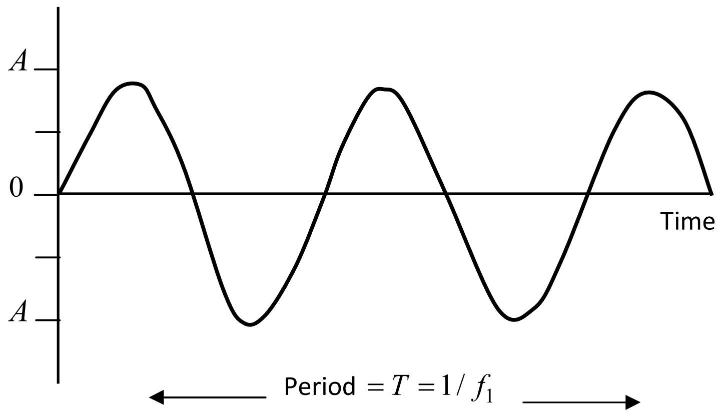


Fig. 1: 1-D Signal.

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.

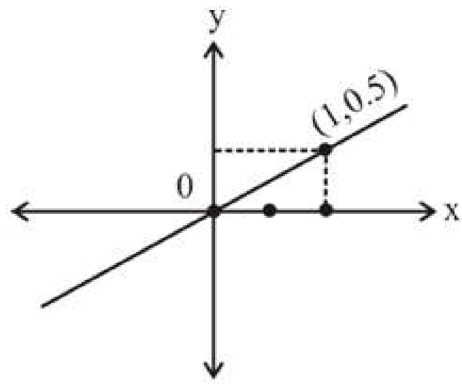


Fig. 2: Representation of $f(x, y) = \delta(x - 2y)$.

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.

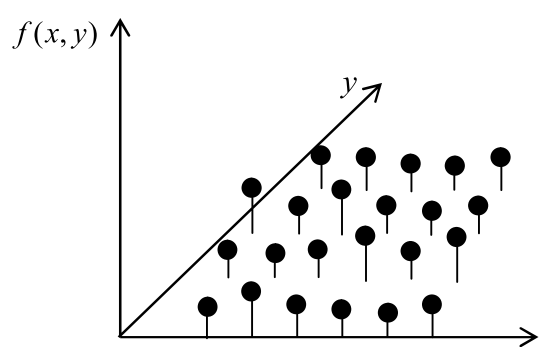


Fig. 3: A Discrete 2-D Signal

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.

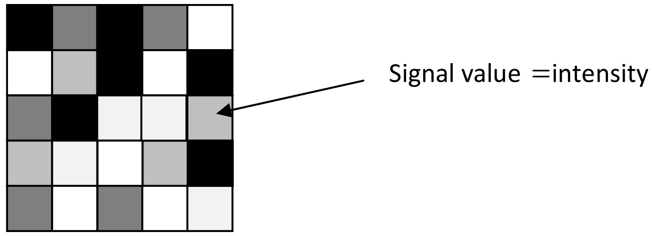
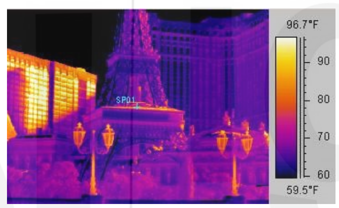


Fig.4: A Gray-Scale Image, An Example of a 2-D Signal.

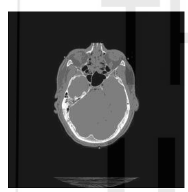
In general, brightness or intensity is usually the value of the function in a image. However, in an image, the signal value can represent physical values such as temperature, pressure, depth, etc. also. We are showing some of these examples in Fig. 5.

(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.

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

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

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

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

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:**

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

ii) Completeness Property:

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

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

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

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

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

$$g(u, v) = A_M\ f\ A_N \tag{9}$$

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

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

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

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

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:

$$\mathbf{g}=\mathbf{T_1}\cdot\mathbf{f}\cdot\mathbf{T}_1^T \qquad (16)$$

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

$$\mathbf{f}=\mathbf{T}_1^{*T}\cdot g\cdot\mathbf{T}_1^{*} \qquad (17)$$

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.

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

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

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

$$\|\mathbf{g}\|^2 = \|\mathbf{f}\|^2 \quad (20)$$

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.

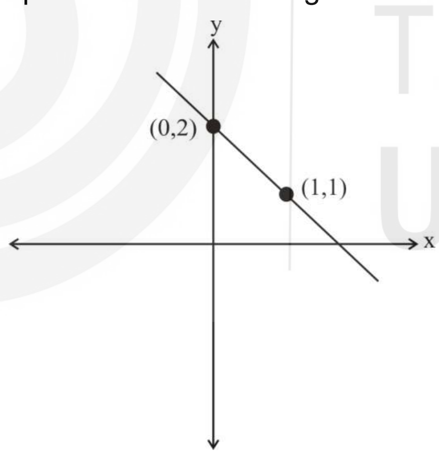


Fig. 7: Representation

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.

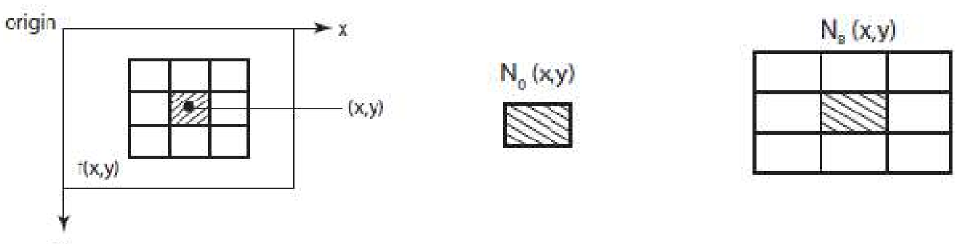
E2) What are the two types of image enhancements? Define them with the help of suitable examples.

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)

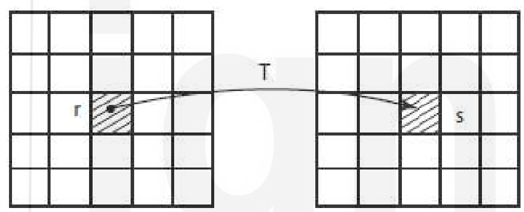
Fig. 1: Point and Neighbourhood Processing

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.

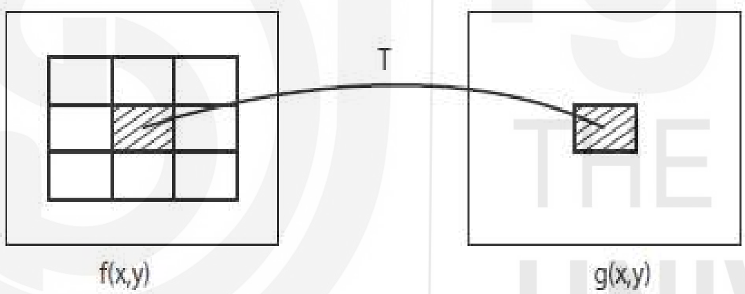


Fig. 3: Neighbourhood Processing

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.

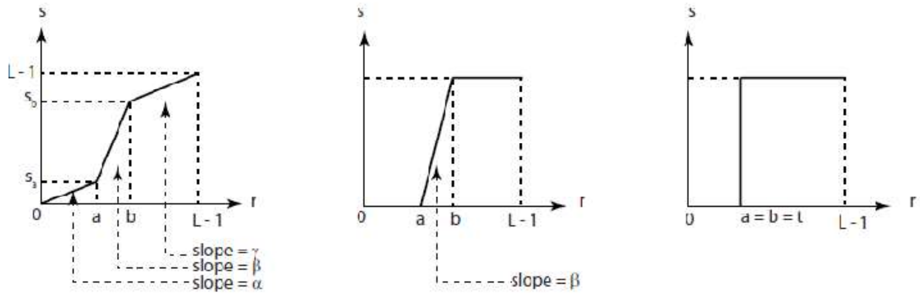
Fig. 4: A low contrast image

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

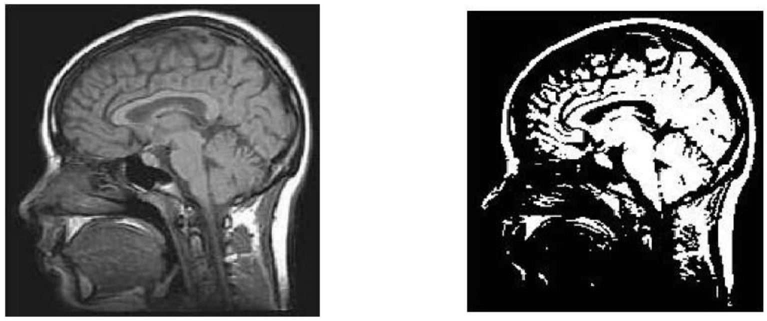
Clipping and thresholding are two special cases of contrast stretching.

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:

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

***

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

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

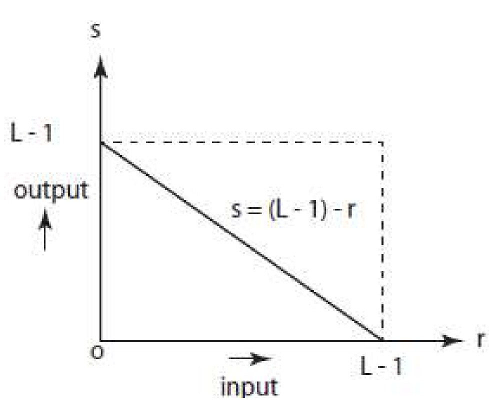


Fig. 8

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

$f(x, y) =$

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

Solution: The negative of the image is given as

$g(x, y) =$

255	245	205	155
250	160	105	55
145	105	165	45
80	45	0	155

***

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 $\{a, \ldots.., b\}$ are set to maximum grey level and all the grey levels outside this range are set to zero as shown in Fig. 10.

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

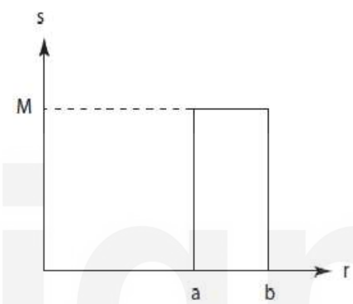


Fig. 10: Transfer Function of Intensity Level Slicing

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

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

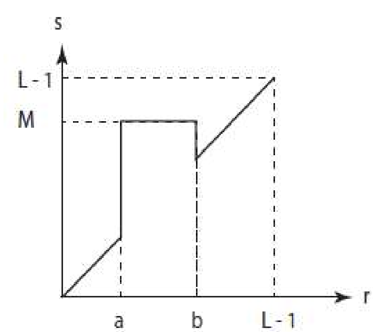
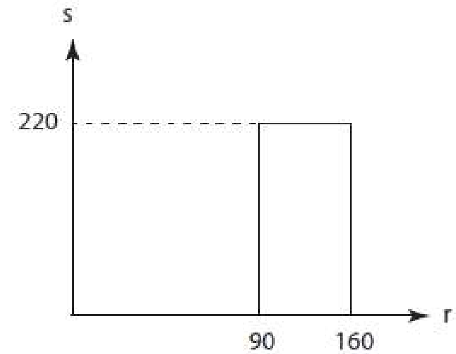


Fig. 11: Transfer Function of Intensity Level Slicing

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

$f(x, y)$

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

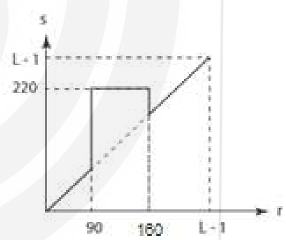
0	0	0	220
0	220	220	0
220	220	0	0
0	0	0	220

$g(x, y)$

***

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 $k_1, k_2, \ldots.., k_8$ 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.

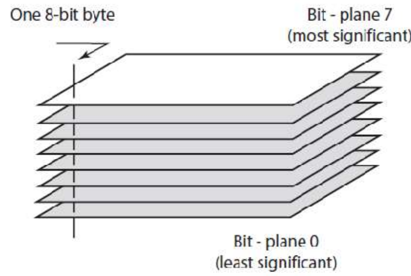
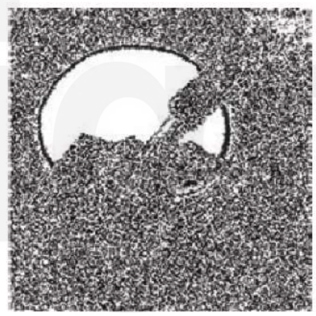
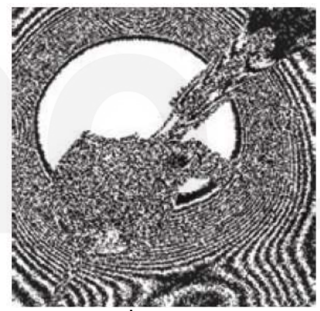


Fig. 12: Bit Plane Slicing

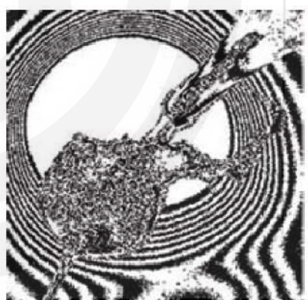
(a) LSB plane



(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 →	1	1	1	1	1	1	1	1
138 →	0	1	0	1	0	0	0	1
30 →	1	1	1	1	1	0	0	0
65 →	1	0	0	0	0	0	1	0
12 →	0	0	1	1	0	0	0	0
201 →	1	0	0	1	0	0	1	1
183 →	1	1	1	0	1	1	0	1
111 →	1	1	1	1	0	1	1	0
85 →	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:

8th plane (MSB)

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

7th plane

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

6th plane

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

5th plane

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

4th plane

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

3rd plane

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

2nd plane

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

1st 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.

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

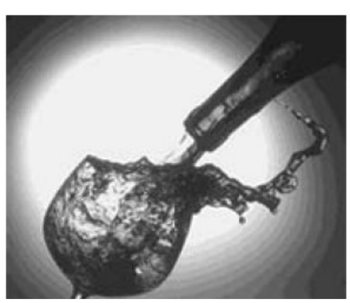
And so on. The resulting image is Fig. 16 (a). There is very small error between the reconstructed image $g(x,y)$ and original image $f(x,y)$ (Fig. 13 (i)). To find the error, we can find $e(x,y) = f(x,y) - g(x,y)$ which is shown in Fig. 16(b). Thus, with only 8th, 7th and 6th plane, the image is very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

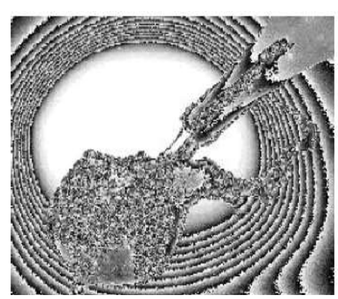
0	0	32	96
32	64	128	196
96	128	160	192
128	192	224	96

$g'(x,y)$

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

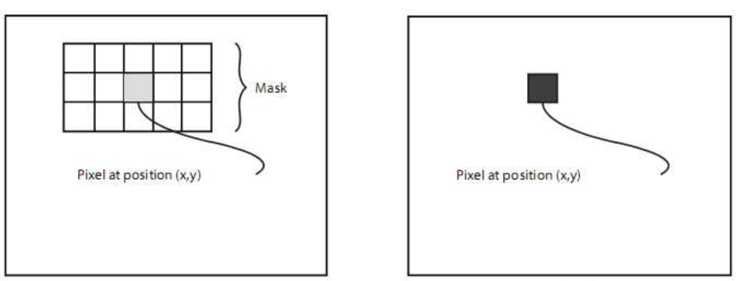
After studying this unit, you should be able to

- 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.

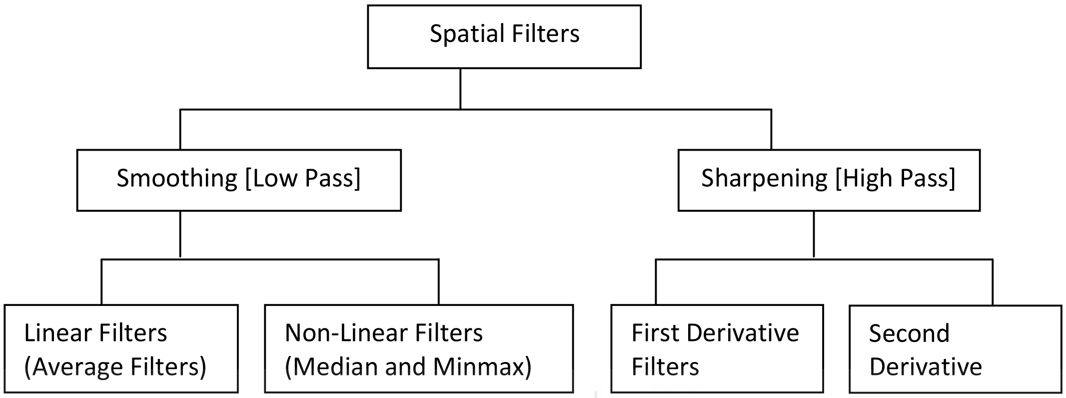


Fig. 2: Types of Spatial Filters

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.

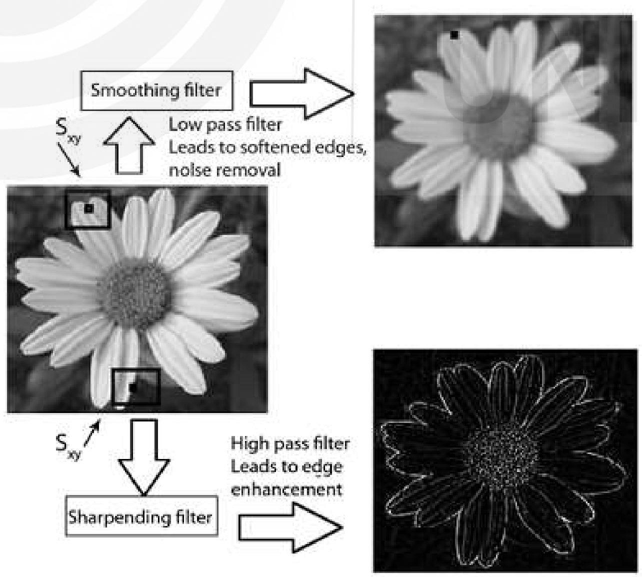


Fig. 3: Smoothing and Sharpening Filters

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

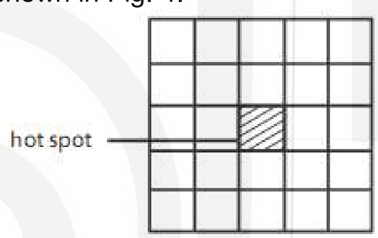
4.3 IMAGE SMOOTHING

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 $5\times5, 7\times7$ or 21×21 (odd sizes) and filter is centered at a coordinate (x, y) called '**Hot Spot**' as shown in Fig. 4.



(b) 5 x 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.

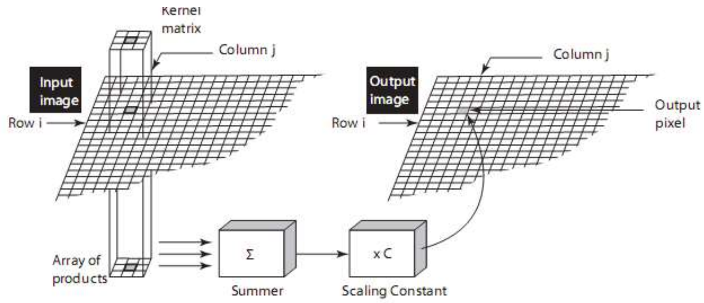


Fig 5: Linear filtering

Let us apply these steps in the following example.

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

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

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

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

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

(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.

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

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

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, 1st 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.

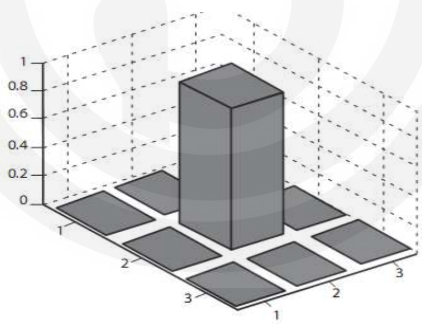
The aim of image smoothing is to diminish the effect of camera noise, spurious pixel values, missing pixel values etc. This is done by a **neighborhood averaging filter.** Each pixel in the smoothened image $g(x, y)$ is obtained from the average pixel value in the neighborhood of (x, y) in input image.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

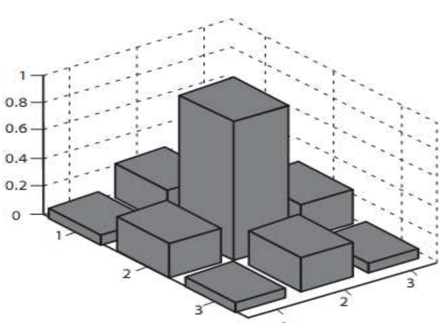
Therefore, smoothing filters are also called averaging filters or low pass filters. Fig. 7 (a) and Fig. 7 (b) shows 3×3 smoothing filter masks. Spatial filter, where all coefficients are equal, is called *box filter*. However, to give maximum importance to pixel at the centre of the mask, it is given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

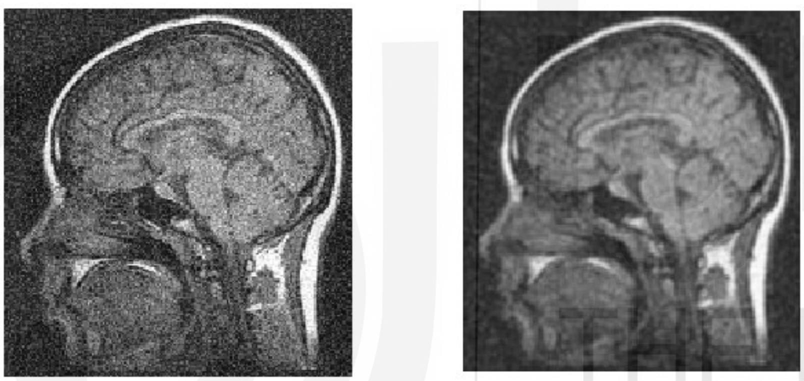
This filter reduces blurring in the smoothing process as the centre pixel is weighted highest.
Smoothing filter is used for the following purposes:

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.

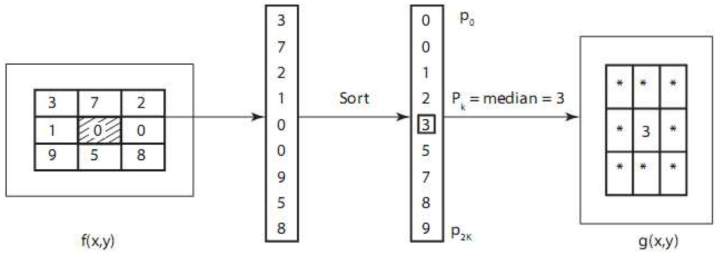


Fig 10: Median Filtering

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.

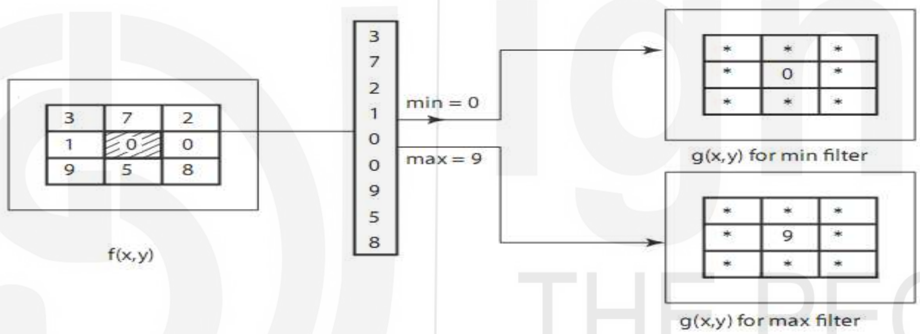


Fig. 11: Min, max filtering

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

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

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

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

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

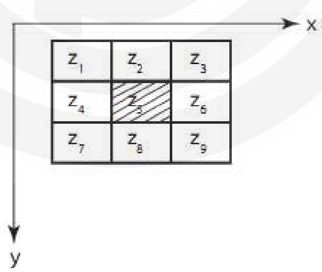
$$\nabla f = \mathrm{mag}(\nabla f(x,y)) = \sqrt{g_x^2 + g_y^2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right] \tag{11}$$

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).



1	0
−1	0

1	−1
0	0

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

1	0
0	−1

0	1
−1	0

(c)

Fig. 12

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

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

$$g_y = z_5 - z_6 \quad (14)$$

$$\nabla f = |z_5 - z_8| + |z_5 - z_6| \quad (15)$$

Another approach is to use cross difference to realized

$$\nabla f \cong |z_5 - z_9| + |z_6 - z_8| \quad (16)$$

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.

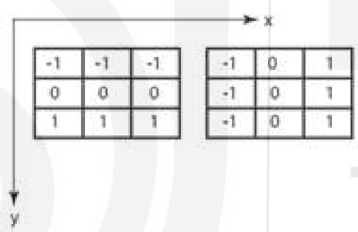


Fig. 13: Perwitt Operator

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.

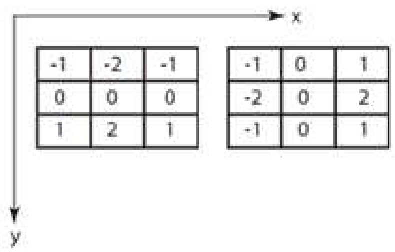


Fig. 14: Sobel Operator

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

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

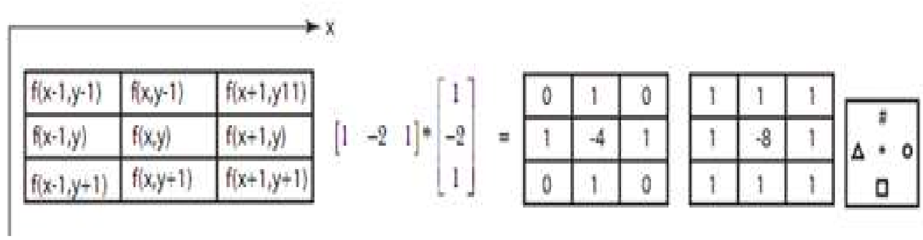
$$\frac{\partial^2 f}{\partial x^2} = f(x-1,y) + 2f(x,y) + f(x+1,y) = [1 \quad -2 \quad 1]$$

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

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)

$$\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(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.

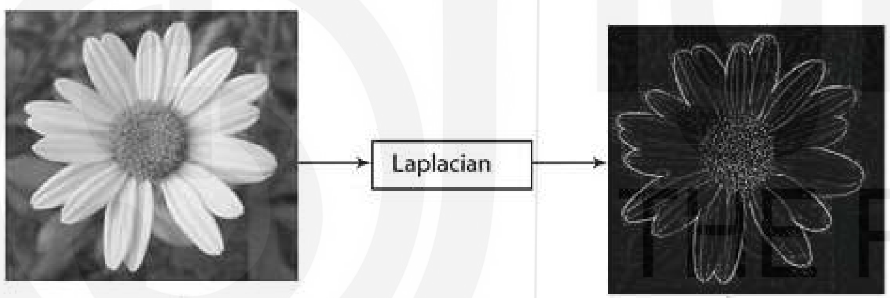


Fig 16: Output after laplacian operator

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

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

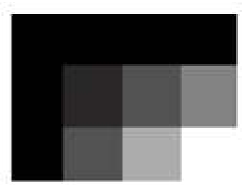
Before we begin with histogram processing, let us define what is the histogram of an image?

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 $0, 1, 2, ..., L-1$.

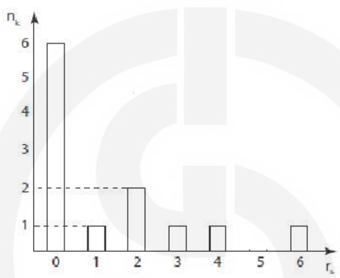
Example 1: Find histogram of image in Fig. 1.



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	$\frac{1}{12}$

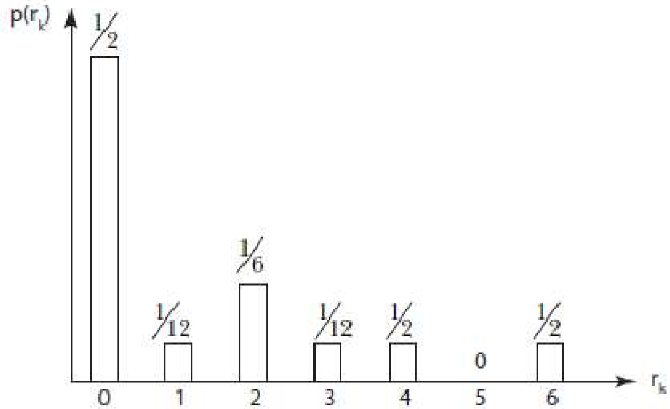


Fig. 3: Normalized Histogram

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.

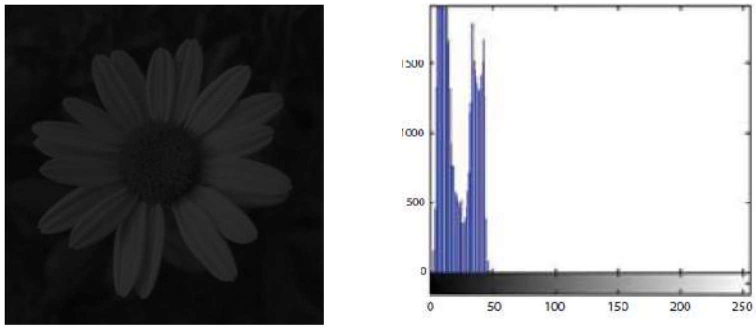
iv) **High contrast image:** Fig. 4 (g) shows a high contrast image and its histogram in Fig. 4 (h). This histogram covers a broad range of grey levels, but distribution of pixels is not uniform. The number of some grey levels is very high as compared to the number of other grey levels.

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.

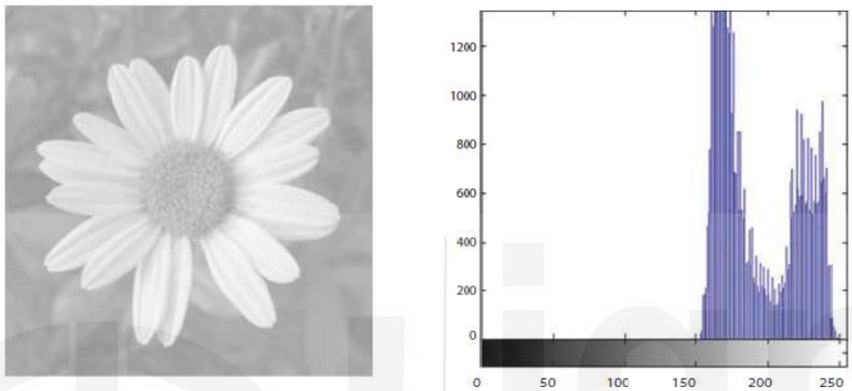
Applications of Histogram

Histogram is widely used in image processing applications.

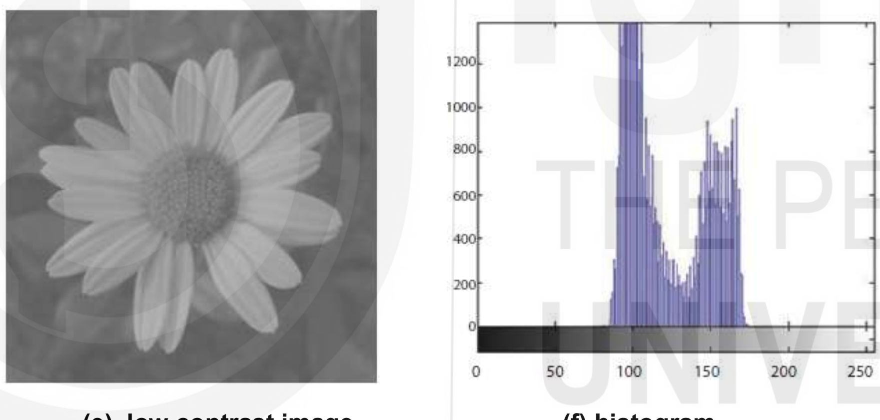
i) Histogram gives a quick indication as to whether or not we have used the entire dynamic range of digitizer. Information is lost if dynamic range is not used fully.



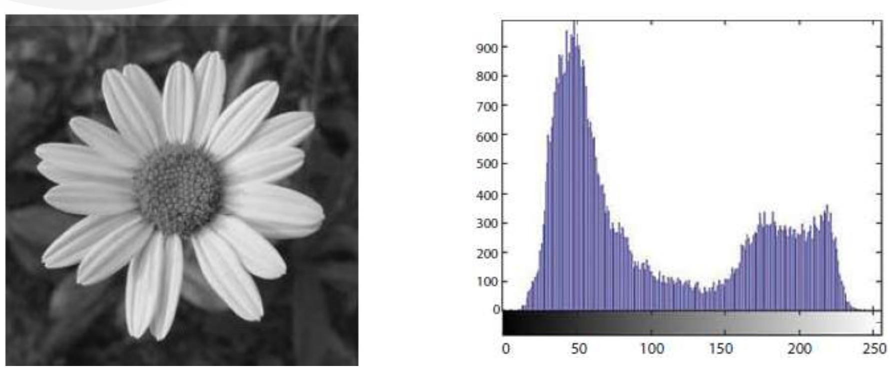
(a) under exposed image **(b) histogram**



(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.

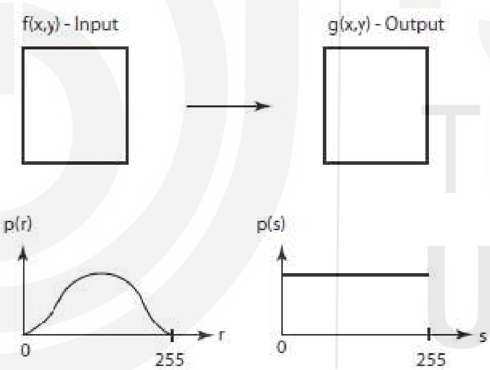


Fig. 5: Histogram Equalization

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:

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

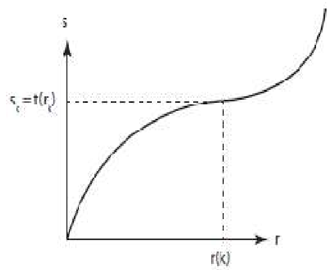
which should satisfy following conditions:

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.

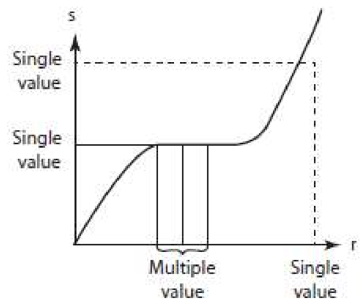
2) $T(r)$ is a monotonically increasing function in the interval $0 \le r \le 1$, Amonotonically increasing function is one which keeps on growing (once it goes up it does not) comedown. This preserves intensity level order in the output image from black to white. This transformation doesn't cause a

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

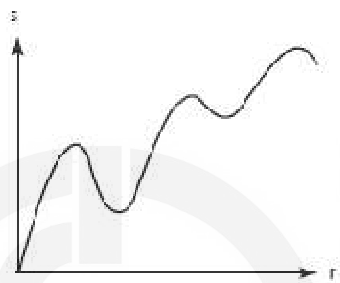
3) The range of r and s is same, that is for $0 \le r \le 1, 0 \le T(r) \le 1$. This guarantees that the output grey levels are in the same range as input grey levels.



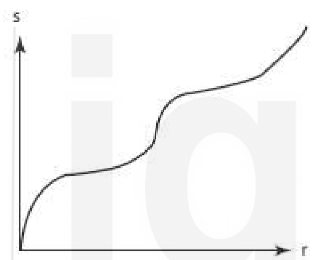
(a) Transfer Function



(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 $0 \le s \le 1$

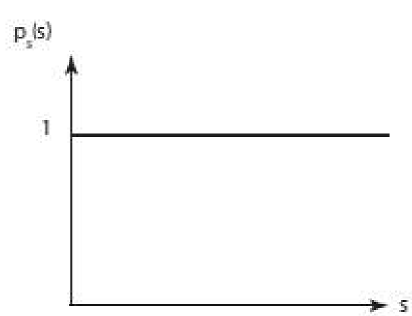


Fig. 7

$p_s(s)$ 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

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

Transformation is given by

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

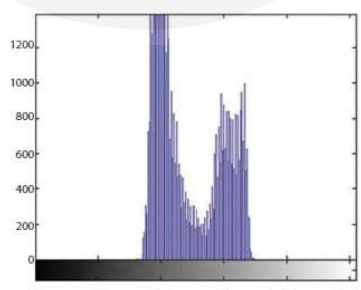
Substituting value of $p_r(r_k)$, we get

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

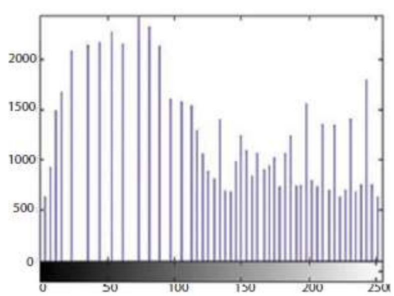
The output image is obtained by mapping each pixel with grey level r_k in the input image to a corresponding pixel with grey level s_k in output image.

(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.

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

$f(x, y)$

equalized $\Longrightarrow$

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

$g(x, y)$

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

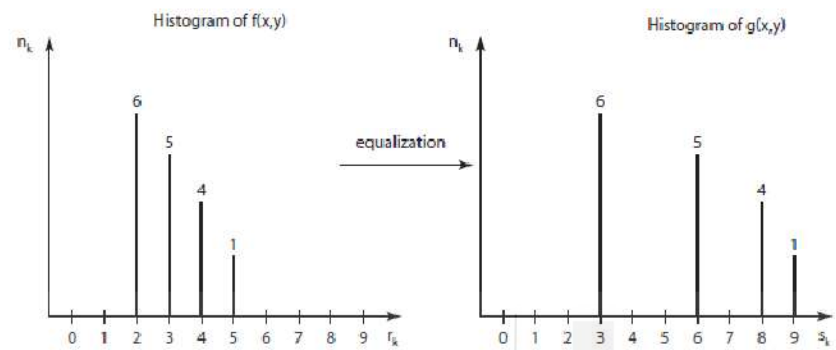


Fig. 9

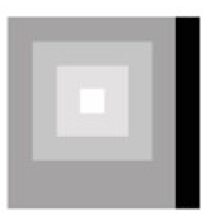
It may be noted that while in the original image the histogram spans between grey values 2 to 5, in the output image, the grey values span from 3 to 9. Although some equalization has been achieved, some of the grey values are still missing. It is because the image has only 9 levels. Histogram equalization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

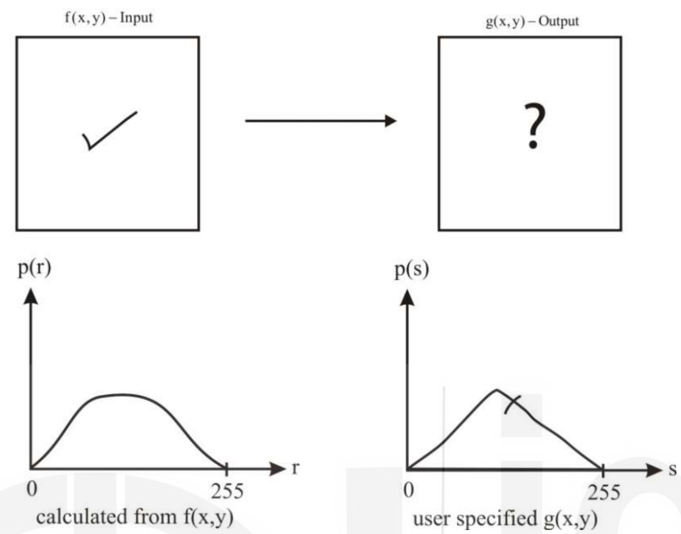


Fig. 9: Histogram specification

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}.

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

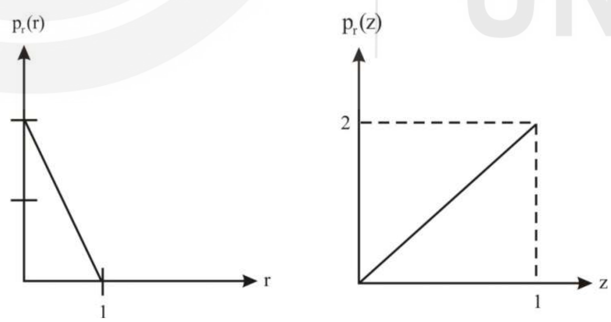
Step 5: Obtain the output image by applying inverse transformation function to all pixels of input image.

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:



(a) (b)

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.

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

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.

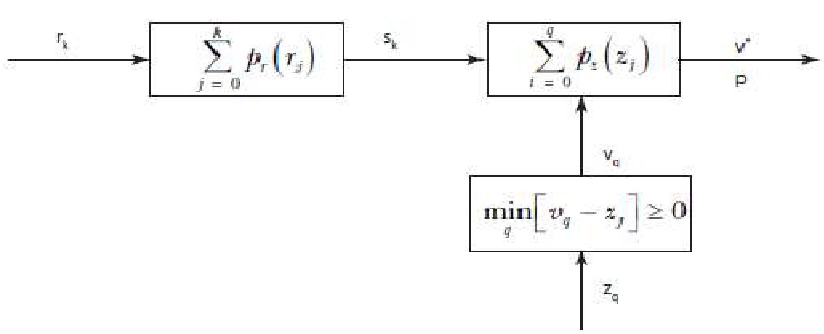


Fig. 11: Histogram Specification

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

0	1	0	2
2	3	3	2
3	1	0	1
1	3	2	0

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

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.

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

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

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.

$r = 0,1,2,3,4$

$p(r_i) = 0.2$ 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 $5\times5, 7\times7$ 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.

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

E6) Max and min filters are defined as:

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

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)

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

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

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

○ Δ □ # *

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

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.

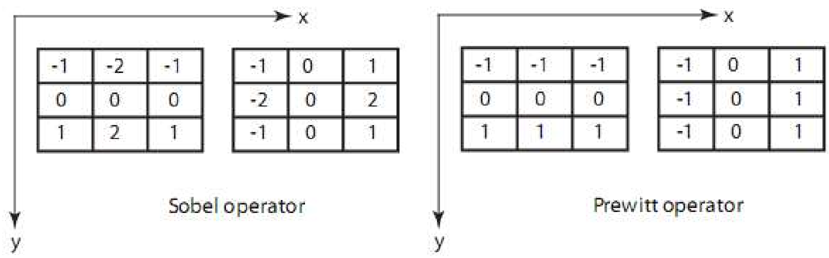
Different image sharpening filters are Laplacian, gradient, Robert, sobel, etc.

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.

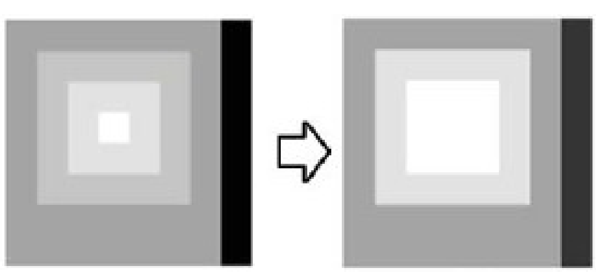
Step 6: Map input levels r_k to output levels s_k using 1st and last row of table above to make table below.

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

UNIT 5

IMAGE TRANSFORMATIONS-FREQUENCY DOMAIN

Structure **Page No.**

5.1 INTRODUCTION

As we saw in Unit 2, an image can be transformed, to show or hide information in the image. Image transformations can be done in both the spatial and the frequency domain. In the spatial domain, image transformation is carried out by changing the value of the pixels based on certain constraints. These transformations can change the brightness and clarity of the images.

In this unit, we shall focus on the transformations in the frequency domain. We recall at this point that an image is also a 2D signal and that the transformations that can be applied on a signal can also be applied on any image. The transformations in the frequency domain provide us information on the frequency content of the image. These transformations can help represent the information in the image in a more compact form, thereby making it computationally easier to store and transmit the images. The transformations may also help in separating the noise and the salient information present in the image.

In Sec. 5.2, we shall focus on very important and useful image transformations, namely the Discrete Fourier transformation (DFT). We shall continue our discussion in Sec. 5.3 with the Discrete Cosine Transformation (DCT). Subsequently, Discrete Wavelet Transform will be discussed in Sec. 5.4 Thereafter, In Sec. 5.5, Haar transform will be discussed . As we go through this unit, we shall see the unique properties of each of these transforms.
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:

- find the Discrete Fourier Transform (DFT)
- compute the Discrete Cosine Transform (DCT)
- find the Discrete Wavelet Transform(DWT)
- find the Haar Transform
- apply the above mentioned transforms

We shall begin the unit with Discrete Fourier Transform(DFT).

5.2 DISCRETE FOURIER TRANSFORM

The Discrete Fourier Transform (DFT) transfers an image from the spatial domain to the frequency domain. It is one of the most important transforms in image processing, which enables us to decompose an image into its sine and cosine components. The output image after applying the Fourier transformation is represented as a linear combination of a collection of sine and cosine waves of different frequencies.

Consider a 1D function, $\{f(x), 0 \le x \le N-1\}$. The general form of a transformation is

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

where $T(u,x)$ is called the **forward kernel** of transformation and $g(u)$ is the transformed image.

If the transformation is the Discrete Fourier Transform.

Then,

$$g(u) = \sum_{x=0}^{N-1} \frac{1}{N} e^{-i2\pi\frac{x}{N}} f(x); u = 0,1,2,...N-1 \tag{2}$$

The inverse 1-D DFT will then be,

$$f(x)=\sum_{x=0}^{N-1} e^{i2\pi\frac{ux}{N}} g(u) \qquad (3)$$

As can be seen the signal is written as a linear combination of an orthogonal set of basis functions. Similarly, an image can be transformed into a set of "basis images", which can be used for representing the image.

We can extend the transform to 2-D image.

Consider an image $f(x,y)$ of size $M \times N$. The 2-D DFT of $f(x,y)$ is defined as follows:

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \qquad (4)$$

And the inverse 2-D DFT is given by:

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{i2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \qquad (5)$$

The 2-D DFT is separable, symmetric and unitary. In case of square images, $M = N$. Many a time in image processing we work with square images. Additionally if the imagesize is a power of 2, then DFT implementation becomes very easy. Computational complexity can be reduced by efficient algorithms such as FFT.

The representation of intensity as a function of frequency is called '**spectrum**'. In the Fourier domain image, each point to a particular frequency contained in the spatial domain image. The coordinates of the Fourier spectrum are the spatial frequencies. The spatial position information of an image is encoded as the difference between the coefficients of the real and imaginary parts. This difference is called the "**phase angle**". The phase information is very useful for recovering the original information. The phase information represents the edge information or boundary information of the objects present in an image. For applications such as medical image analysis, the phase information is very crucial in getting information from the image.

Note that while the image values $f(x,y)$ are going to be real, the corresponding frequency domain data is going to be complex. There will be one matrix containing real values $R(u,v)$ and the other matrix $I(u,v)$ will contain the imaginary component of the complex value. The amplitude spectrum or the magnitude for 2D DFT is given by

$$|F(u,v)| = [R^2(u,v)+I^2(u,v)]^{1/2} \qquad (6)$$

where, R and I are real and imaginary parts of $F(u,v)$ and and all computations are carried out for the discrete variables $u = 0,1,2,...,M-1$ and $v = 0,1,2,...,N-1$. The spectrum tells us the relative magnitude at each frequency.

The power spectrum of the 2-D DFT is defined as

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v) \tag{7}$$

and the phase spectrum of the 2-D DFT is given by

$$\text{ø}(u,v) = \tan^{-1}\frac{I(u,v)}{R(u,v)} \tag{8}$$

Note that the size of the image remains the same as the original image in spatial domain.Therefore, the magnitude, Fourier (phase) spectrum andthe power spectrum are all matrices of size $M \times N$.

Remark: We can find 2-D DFT of an image by simply computing a set of 1-D DFT is for all rows of $f(x,y)$. Thus, the 2-D DFT of an image $f(x,y)$ is

$$F(u,v) = \sum_{x=0}^{M-1} e^{\frac{-2\pi i u x}{M}} \sum_{y=0}^{N-1} f(x,y) e^{\frac{-2\pi i v y}{N}}.$$

$$= \sum_{x=0}^{M-1} F(x,v) e^{\frac{-2\pi i u x}{M}},$$

where $F(x,v) = \sum_{y=0}^{N-1} f(x,y) e^{\frac{-2\pi i v y}{N}}$.

Also, the 2-D DFT can also be found using the Eqn. (4) with the condition of separability as we used in Unit-2.

Let us discuss properties of 2-D DFT.

DFT has several useful properties which makes it an important transformation.

i) **Separability:** It is separable because a 2D transform is separable if $T(u,x,v,y) = T_1(u,x) \cdot T_2(v,y)$.

ii) **Symmetry:** It is symmetric because a 2D transform is symmetric if $T_1(u,x) = T_2(u,x)$.

iii) **Periodicity:**The 2-D DFT and the 2-D IDFT are both periodic, that is, $F(u,v) = F(u+M,v) = F(u,v+N) = F(u+M,v+N)$.

iv) **Conjugate symmetry:** $F(u,v) = F^*(-u+pM,-v+qN)$, where p and q being integers. The property of conjugate symmetry implies that $|F(u,v)| = |F(-u,-v)|$

v) If $f(x,y)$ is real and even then $F(u,v)$ is real and even.

vi) If $f(x,y)$ is real and odd then $F(u,v)$ is imaginary and odd.

vii) Let F be the DFT operator, then

$$F(f(x,y)+g(x,y)=F(f(x,y)+F(g(x,y)$$

However, $F(f(x,y)\cdot g(x,y)\neq F(f(x,y)\cdot F(g(x,y)$

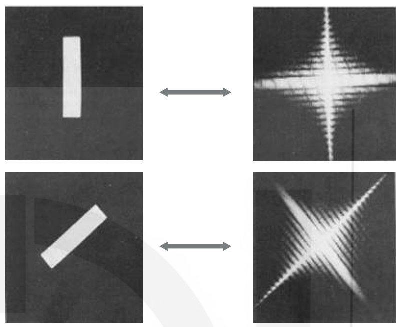
viii) Translation in the spatial domain by (x_0, y_0) implies

$$f(x-x_0, y-y_0)\leftrightarrow F(u,v)e^{-j2\pi\left(\frac{ux_0}{M}+\frac{vy_0}{N}\right)}$$

While translation in the frequency domain by (u_0, v_0) implies

$$F(u-u_0, v-v_0)\leftrightarrow f(x,y)e^{j2\pi\left(\frac{xu_0}{M}+\frac{yv_0}{N}\right)}$$

ix) The average value of the signal is given by



$$\bar{f}(x,y)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)$$

If we see the value of $F(0,0)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\Rightarrow F(0,0)=\bar{f}(x,y)$.

x) **Rotation:** Rotating $f(x,y)$ by θ rotates $F(u,v)$ by θ.

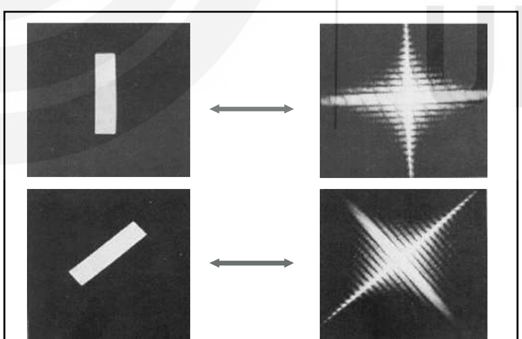


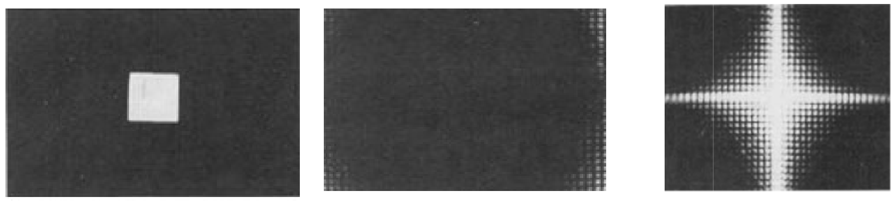
Fig. 1: Rotation in $f(x,y)$

Fig. 1 shows how the rotation in the spatial domain (on left) affects the rotation in the frequency domain (right).

After going through all the properties of DFT, let us see how do we visualize 2-D DFT. We need to translate the origin of the transformed image to the center of the image $(u,v)=(M/2, N/2)$ to be able to display the full period of the 2-D DFT. As we saw above, translating the Fourier image to the center, requires us to use the translation property of $F(u,v)$ with $u_0 = M/2$ and $v_0 = N/2$.

Then, $F\left\{f(x,y)e^{i2\pi\left(\frac{xu_0}{M}+\frac{yv_0}{N}\right)}\right\} = F(u-u_0, v-v_0)$ becomes

$$F\{f(x,y)e^{i\pi(x+y)}\} = F\{f(x,y)(-1)^{(x+y)}\} = F(u-M/2, v-N/2)$$



a) Original Image **b) Original DFT image** **c) Translated DFT image**

Fig. 2: DFT of an Original Image

We can see in Fig. 2 what changes do we see after DFT. Fig 2. (a) is the original image in the spatial domain, (b) is the 2-D DFT image, and (c) is the translated DFT image to show the full period of the 2D DFT of image in (a).

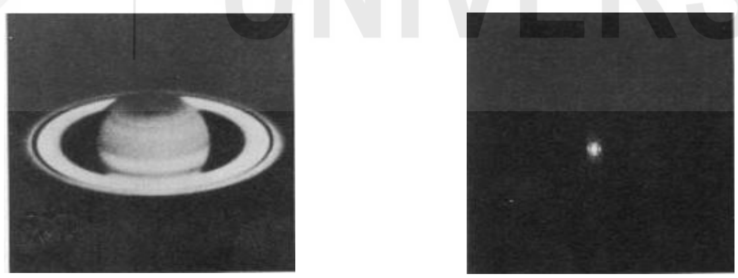
Let us see how do we visualize the range of 2-D DFT.

In general, the range of values the 2-D DFT $F(u,v)$ is very large. Therefore, when we attempt to display the values of $F(u,v)$, smaller values are not distinguishable because of quantization as can be seen in Fig. 3 (b). Therefore, to enhance the small values, weapply a logarithmic transformation given by

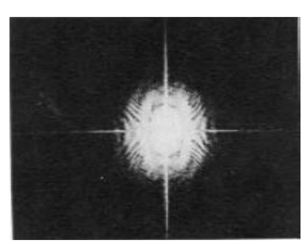
$$D(u,v) = c\log(1+|F(u,v)|)$$

Where, the parameter $c$is chosen so that the range of $D(u,v)$ is $[0,255]$.

$$c = \frac{255}{\log(1+\max\{|F(u,v)|\})}$$



a) The original image **b) The 2D DFT image**



c) The 2D DFT image after log transform

Fig. 3: DFT image after log tranform

We can visualise the display of the amplitude of the 2-D DFT after logarithmic transformation in Fig. 3(b) and Fig. 3(c) respectively for the original image as shown in Fig. 3(a).

Example 1: Compute the DFT of the 1D sequence $f(x)=[1,0,-1,0]$.

Solution: Here $N=4$. Using Eqn. (2) we get

$$g(u)=\frac{1}{4}\sum_{x=0}^{3} f(x)\cdot e^{\frac{-i,2\pi\cdot ux}{4}};\ u=0,1,2,3$$

$$=\frac{1}{4}\sum_{x=0}^{3} f(x)\cdot\left(e^{\frac{-2\pi i}{4}}\right)^{ux};\ u=0,1,2,3$$

$$=\frac{1}{4}\sum_{x=0}^{3} f(x)\ (-i)^{ux};\ u=0,1,2,3$$

$$=\frac{1}{4}[f(0)(-i)^{0}+f(10(-i)^{u}+f(2)(-i)^{2u}+f(3)(i)^{3u}];u=0,.1,2,3$$

$$=\frac{1}{4}[1+0+(-1)(-i)^{2u}+0];u=0,1,2,3$$

$$=\frac{1}{4}[1-(i)^{2u}];u=0,1,2,3$$

This gives $\mathbf{g}=\frac{1}{4}[0,\ 2,\ 0,\ 2]$, which is the DFT of $f(x)$.

***

Example 2: Construct a DFT matrix of order 2.

Solution: Here $N=2$.

$$g(u)=\frac{1}{2}\sum_{x=0}^{1} f(x)e^{\frac{-i.2\pi.ux}{2}};\ u=0,1.$$

$$=\frac{1}{2}\sum_{x=0}^{1} f(x)(-1)^{ux};\ u=0,1$$

$$=\frac{1}{2}[f(0)(-1)^{0}+f(1)(-1)^{u}];\ u=0,1$$

$$=\frac{1}{2}[f(0)+(-1)^{u}f(1);u=0,1$$

Now, $g(0)=\frac{1}{2}[f(0)+f(1)]$

$g(1)=\frac{1}{2}[f(0)-f(1)]$

DFT matrix $=\frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}.$

***

Example 3: Compute the 2-D DFT of the 2×2 image

$$f(x,y)=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}.$$

Solution: Let the DFT of $f(x,y)$ be $F(u,v)$, which is given in Eqn. (4).

$$F(u,v)=\sum_{x=0}^{1}\sum_{y=0}^{1} f(x,y)e^{-2\pi i\left(\frac{ux}{2}+\frac{vy}{2}\right)};\ u,v=0,1$$
$$=\sum_{x=0}^{1}\sum_{y=0}^{1} f(x,y)(-1)^{ux}(-1)^{vy};\ u,v=0,1$$
$$=[f(0,0)(-1)^{0}(-1)^{0}+f(0,1)(-1)^{0}(-1)^{v}+f(1,0)(-1)^{u}(-1)^{0}$$
$$+f(1,1)(-1)^{u}(-1)^{v}];\ u,\ v=0,1$$
$$=[f(0,0)+(-1)^{v}f(0,1)+(-1)^{u}f(1,0)+(-1)^{u}(-1)^{v}f(1,1)];\ u,v=0,1,1.$$

$$F(0,0)=f(0,0)+f(0,1)+f(1,0)+f(1,1)=4$$
$$F(1,0)=f(0,0)+f(0,1)-f(1,0)-f(1,1)=0$$
$$F(0,1)=f(0,0)-f(0,1)+f(1,0)-f(1,1)=0$$
$$F(1,1)=f(0,0)-f(0,1)-f(1,0)+f(1,1)=0$$

Thus, the 2-D DFT of the given image is $\begin{bmatrix}4 & 0\\ 0 & 0\end{bmatrix}$.

F(0,0) is 4 which happens to be the average of all the intensity values in original image. The other values represent frequency values. But since there is no variation in values of original image, there is no frequency involved, and that is why the frequency values in DFT are zeroes.

Alternatively, the 2-D DFT can also be found using the DFT basis matrix formed by finding 1-D DFT of each row of $f(x,y)$ and then using that as kernel.

That is $F(u,v)=$ kernel $\times f(x,y)\times$ (kernel)^{T}

We have already found the DFT matrix of order 2 in Example 2.

Therefore,

$$\text{kernel}=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

Hence, $F(u,v)=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$ [we are skipping $\frac{1}{2}$ as in 2-D we have taken $\frac{1}{MN}$ in inverse DFT]

$$=\begin{bmatrix}2 & 2\\ 0 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$
$$=\begin{bmatrix}4 & 0\\ 0 & 0\end{bmatrix}$$

Both the results are same.

***

Try the following exercises.

E1) Does the implementation of a separable and symmetric transform, such as the DFT in an image requires the sequential implementation of the corresponding one-dimensional transform row-by-row and then column-by-column? Justify your answer.

E2) Find the DFT of the sequence $f(x) = [i, 0, i, 1]$.

E3) Construct a DFT matrix of order 4. Also, check whether DFT matrix is unitary matrix or not.

E4) Find the inverse 2-D DFT of $F(u, v)$ found in Example 3.

In the following section, we shall discuss discrete cosine transform.

5.3 DISCRETE COSINE TRANSFORM

The Discrete Cosine Transform DCT is a family of unitary transformations that transforms the real values of input image to another set of real values. Unlike the DFT that is complex, the DCT is a real transform because it projects the signal onto real cosinewaves.

The 1-D DCT is given as:

$$C(u) = a(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right]; 0 \le u \le N-1, \qquad (9)$$

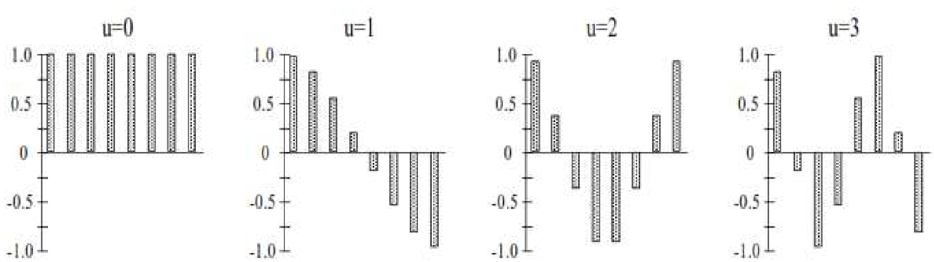
where, $a(u) = \begin{cases} \sqrt{\dfrac{1}{N}}; & u = 0 \\ \sqrt{\dfrac{2}{N}}; & u = 1, \ldots, N-1 \end{cases}$

Then, the inverse transform is given by

$$f(x) = \sum_{u=0}^{N-1} a(u)C(u)\cos\left[\frac{(2x+1)u\pi}{2N}\right], \qquad (10)$$

where $a(u)$ is the same function as used for DCT.

Let us visualize the effect of 1-D DCT through Fig. 4, where in the rows of the 8×8 transformation matrix of the DCT for a signal $f(x)$ with 8 samples are shown.



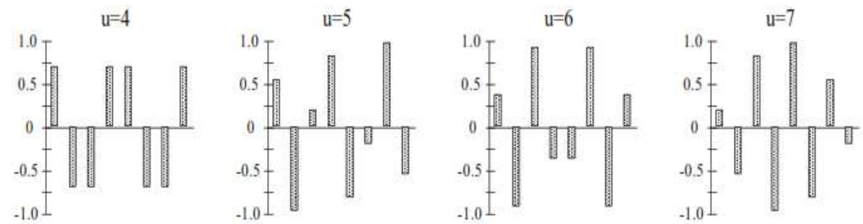


Fig. 4: 1-D DCT

The figures for values of u from 0 to 7 show the various rows of the 8×8 transformation matrix of the DCT for a 1D signal $f(x)$ with 8 samples.

Let us now expand 1-D DCT to 2-D DCT.

Consider an image $f(x,y)$ of size $M\times N$. Then, the 2-D DCT of the image is defined as:

$$C(u,v)=a(u)a(v)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right], \qquad (11)$$
$$0\le u\le M-1, 0\le v\le N-1$$

And the inverse 2-D DCT is given by

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}a(u)a(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \qquad (12)$$
$$0\le x\le M-1, 0\le y\le N-1$$

Where $a(u)$ is same as defined earlier for 1D DCT.

DCT and DFT are very similar, however, DCT has the advantage over DFT that DCT is real while DFT is complex. Moreover. DCT has better energy compaction in comparison to the energy compaction of DFT. Energy compaction is the ability to pack the energy of the spatial sequence into as few frequency coefficients as possible. This property is exploited for image compression and is a very important property.You can see in the Fig.5, that most of the DCT image is dark, which means DCT values are concentrated only in few pixels very near the origin. This indicates that DCT has high compaction.

a) Original image – Lena **b) 2D DCT of Lena**

Fig.5: The 2D DCT in (b) of the image Lena in (a) shows the high compaction capability of DCT.

Example 4: Compute the discrete cosine transform (DCT) matrix for order 2.

Solution: Using Eqn. (9), we substitute $N = 2$, and we get

$$C(u) = a(u)\sum_{x=0}^{1} f(x)\cos\frac{(2x+1)u\pi}{2\times 2};\ 0 \le u \le 1.$$

where $a(u) = \begin{cases} \dfrac{1}{\sqrt{2}} & ;\quad u = 0 \\ \sqrt{\dfrac{2}{2}} = 1; & \quad u = 1 \end{cases}$

At $u = 0,$ we get

$$C(0) = \frac{1}{\sqrt{2}}\sum_{x=0}^{1} f(x)\ \cos\frac{(2x+1)\,\pi\times 0}{4}$$

$$= \frac{1}{\sqrt{2}}\sum_{x=0}^{1} f(x)\cdot 1$$

$$= \frac{1}{2}\sum_{x=0}^{1} f(x)$$

$$= \frac{1}{2}[f(0) + f(1)]$$

At $u = 1,$ we get

$$C(1) = 1\cdot\sum_{x=0}^{1} f(x)\cdot\cos\frac{(2x+1)\cdot 1\cdot\pi}{4}$$

$$= \sum_{x=0}^{1} f(x)\cos\frac{(2x+1)\,\pi}{4}$$

$$= \left[f(0)\cos\frac{\pi}{4} + f(1)\cos\frac{3\pi}{4}\right]$$

$$= \frac{1}{\sqrt{2}}f(0) - \frac{1}{\sqrt{2}}f(1)$$

By collecting the coefficients, we get the required DCT. Therefore, DCT is

$$C(u) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

***

Now try the following exercises.

E5) Why is DCT important for image compression?

E6) Find the DCT of the matrix of order 4.

So far, we have discussed discrete fourier transform and discrete cosine transform.

5.4 DISCRETE WAVELET TRANSFORM

In Block-1 of this course we learned about the Spatial domain, it was learned that the Spatial domain is the normal image space where the term "the domain" refers to the normal image space that is represented as a matrix of pixels, . In Spatial domain, the transformation methods are executed by directly operating on the pixel values of an image. Adjustments in spatial domain are made to the values in order to obtain the desired level of improvement.

In earlier sections of this unit we learned about the second type of domain i.e. the frequency domain, where the pace at which the individual color components in an image shift is referred as the image's frequency and in this domain i.e. frequency domain the prime focus is on the rate at which the pixel values in the spatial domain vary. It is to be noted that, in any image the color changes very quickly, for the regions with high frequencies, whereas in regions that contain low frequencies, the color changes quite gradually.

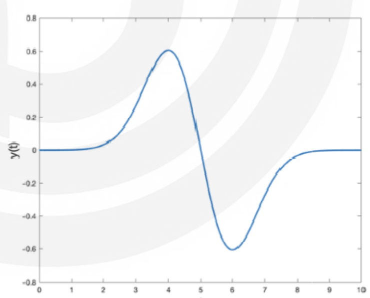
It is essential to keep in mind that, in contrast to the spatial domain, the frequency domain does not provide direct operations on the values. This restriction prevents you from performing some calculations in the frequency domain. In order to begin the processing of the image, it must first go through a transformation that restores it to its original frequency distribution. It is possible to separate the frequency components into two basic sub-components. Components with a high frequency that correlate to the edges of an image and components with a low frequency relates to the smooth regions of an image . This technique does not result in the production of an image as its end product; rather, it produces a transformation as its conclusion. It is necessary to carry out an inverse transformation on the data that was produced as a result of the processing that was done so that the image can be restored to its perfect, original form.

Also, we learned that Fourier transform is a powerful tool that has been available to signal analysts for many years. It gives information regarding the frequency content of a signal. However, the problem with using Fourier transforms is that frequency analysis cannot offer both good frequency and time resolution at the same time. A Fourier transform does not give information about the time at which a particular frequency has occurred in the signal. Hence, a Fourier transform is not an effective tool to analyse a non-stationary signal. To overcome this problem, windowed Fourier transform, or short-time Fourier transform, was introduced. Even though a short-time Fourier transform has the ability to provide time information, multi-resolution is not possible with short-time Fourier transforms. Wavelet is the answer to the multi-resolution problem. A wavelet has the important property of not having a fixed-width sampling window.

The technique of wavelet transformation, which is typically utilised for the analysis of images and the compression of data, will be investigated in this section. In spite of the fact that the frequency domain encompasses a number of other mathematical transformations, including the Fourier transform, the Laplace transform, and the Z transform, the wavelet transformation method is going to be the one that is discussed in detail in this section.

Let's begin with acquiring an understanding of what wavelets are and why we need this transformation before we move on to trying to comprehend the Discrete Wavelet Transformation, often known as the DWT. This will help us get a better grasp on the DWT. According to Wikipedia, "a wavelet is a wave-like oscillation with an amplitude that begins at zero, rises, and then decreases back to zero." It is best to think of it as a "short oscillation" in the majority of situations, which is analogous to what could be captured by a seismograph or heart monitor.

Let’s try to understand this concept of wavelet in a better way, with the explanation given below:
A wavelet is a wave-like oscillation that is localised in time; an example of this type of oscillation is provided further down in this paragraph. Scale and location are the two fundamental features that wavelets possess. How "stretched" or "squished" a wavelet is can be defined by its scale, which can also be referred to as its dilation. This characteristic is connected to frequency in the sense that it is understood for waves. The wavelet's position in time can be determined based on its location (or space).



$$-(x-b)e^{\frac{-(x-b)^2/(2a^2)}{\sqrt{2\pi}a^3}}$$

First derivative of Gaussian Function

The magnitude of the wavelet can be calculated by examining the value of the parameter labelled "a" in the preceding expression. When the value is decreased, the wavelet will take on an appearance that is more compressed. As a direct result of this, it is now possible to acquire information at high frequencies. In contrast, increasing the value of "a" will cause the wavelet to stretch, which will lead to the collection of information at low frequencies. The value of the "b" parameter is what decides where the wavelet is positioned in the image. When the value of "b" is decreased, the wavelet will shift to the left. Increasing the value of "b" will cause it to relocate to the right. In contrast to waves, wavelets only exhibit non-zero behaviour for a small period of time during which their locations are meaningful. This difference in behavior between wavelets and waves is called the wavelet scale. When we perform an analysis of a signal, in addition to being interested in the oscillations that the signal displays, we are also interested in the locations of those oscillations.

The fundamental concept here is to determine the proportion of a wavelet that exists in a signal at a specific scale and location. For those of you who are familiar with convolutions, this is a perfect example. A signal is convolved with a set of wavelets operating at a range of different scales. We go with a wavelet that has a specified scale. After that, we multiply the wavelet and the signal at each time step, and then we slide this wavelet across the entire signal, which means we change where it is located. The result of performing this multiplication provides us a coefficient that corresponds to that wavelet scale at that time step. After that, the wavelet scale is increased, and the procedure is carried out again.

Based on previous explanation, we understood that wavelets are functions that are concentrated in time and frequency around a certain location.

Generally, got confused for waves and wavelets but they are different the fundamental difference between the two is that a wave is an oscillating function of time or space that is periodic. The wave is an infinite length continuous function in time or space. In contrast, wavelets are localised waves. A wavelet is a waveform of an effectively limited duration that has an average value of zero.

A function () can be called a wavelet if it posses the following properties:

1. The function integrates to zero, or equivalently its Fourier transform denoted as $\psi(\omega)$ is zero at the onigin:

$$\int_{-\infty}^{\infty} \psi(x)dx = 0 \tag{12a}$$

This implies $\psi(\omega)|_{\omega=0}$ =0 in the frequency domain.

2. It is square integrable, or equivalently, has finite energy:

$$\int_{-\infty}^{\infty} |\psi(x)|^2 dx < \infty \tag{12b}$$

3. The Fourier transform $\psi(x)$ must satisfy the admissibility condition given by

$$C_\psi \int_{-\infty}^{\infty} = \frac{|\psi(\omega)|^2}{|\omega|} d\omega < \infty \tag{12c}$$

Interpretation of Eqs. (12a), (12b) and (12c)

Equation (12a) suggests that the function is either oscillatory or has a wavey appearance.Equation (12b) implies that most of the energy in $\psi(x)$ is confined to a finite interval, or in other words. $\psi(x)$ has good space localisation. Ideally, the function is exactly zero outside the finite interval. This implies that the function is a compactly supported function. Equation (12c) is useful in formulating the inverse wavelet transform. From Eq. (12c), it is obvious that in $\psi(x)$ must have a sufficient decay in frequency. This means that the Fourier transform of a wavelet is localized, that is, a wavelet mostly contains frequencies from a certain frequency band. Since the Fourier transform is zero at the origin, and the spectrum decays at high frequencies, a wavelet has a bandpass characteristic. Thus a wavelet is a 'small wave' that exhibits good time-frequency localisation. A family of wavelets can be generated by dilating and translating the mother wavelet in $\psi(x)$ which is given by

$$\psi_{(a,\ b)}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right)$$

(12d)

Here, a is the scale parameter and b is the shift parameter.

After understanding the concept of wavelets now it's time to begin with our topic of Wavelet Transform.

Wavelet transforms can be either continuous or discrete, depending on how they are implemented. The Continuous Wavelet Transform (CWT) uses every wavelet that is feasible across a range of scales and places, meaning that it has an endless number of scales and locations to choose from. This is the primary distinction between these two types. While the Discrete Wavelet Transform (DWT) uses a limited number of wavelets, which are defined according to a specific range of sizes and locations, this set of wavelets is not limited in any way. Few ore comparisons between CWT and DWT are given below"

	CWT- Continuous Wavelet Transform	DWT- Discrete Wavelet Transform
1 Scale	At any scale	Dyadic scales
2. Translation	At any point	Integer point
3. Wavelet	Any wavelet that satisfies minimum criteria	Orthogonal, biorthogonal, . . .
4. Computation	Large	Small
5. Detection	Easily detects direction, orientation	Cannot detect minute object if not finely tuned
6. Application	Object Detection, Pattern recognition Feature extraction	Compression, De-noising, Transmission Characterisation

So, the wavelet transform can be broadly classified into (i) continuous wavelet transform, and (ii) discrete wavelet transform. For long signals, continuous wavelet transform can be time consuming since it needs to integrate over all times. To overcome the time complexity, discrete wavelet transform was introduced. Discrete wavelet transforms can be implemented through sub-band coding. The DWT is useful in image processing because it can simultaneously localise signals in time and scale, whereas the DFT or DCT can localise signals only in the frequency domain.

It is to be noted that apart from image processing, the DWT is quite promising tool for the Signal processing also. After the suggestion of Mallat's, that signals may be represented at several resolutions using wavelet decomposition, Discrete Wavelet Transform (DWT) emerged as an extremely flexible tool for signal processing. Because the energy of wavelets is concentrated in time while still retaining the wave-like (periodic) characteristics, we discovered that wavelets make it possible to perform time and frequency analysis of signals at the same time. This was one of the key takeaways from the investigation into wavelets. As a consequence of this, wavelet representation offers a flexible mathematical tool for the analysis of transient, time-variant (non-stationary), signals that are not statistically predictable, particularly in the region of discontinuities. This quality is characteristic of images that have discontinuities at the edges. In DWT, a digital signal splits up into its component sub-bands, so that the lower frequency sub-bands have finer frequency resolution and coarser time resolution compared to the higher frequency sub-bands. .

The wavelet transformation technique overcomes the limitations of the Fourier method. The Fourier transformation, despite the fact that it deals with frequencies, does not reveal any facts regarding the passage of time. In accordance with the Heisenberg's Uncertainty Principle , we can either have a high frequency resolution but a low temporal resolution, or vice versa. The introduction to the Heisenberg's Uncertainty Principle is given below:

The Heisenberg uncertainty principle was originally stated in physics, and claims that it impossible to know both the position and momentum of a particle simultaneously. However, it has an analog basis in signal processing. In terms of signals, the Heisenberg uncertainty principle is given by the rule that it is impossible to know both the frequency and time at which they occur. The time and frequency domains are complimentary. If one is local, the other is global. Formally, the uncertainty principle is expressed as

$$(\Delta t)^2\,(\Delta\omega)^2 \geq \frac{1}{4}$$

In the case of an impulse signal, which assumes a constant value for a brief period of time, the frequency spectrum is infinite; whereas in the case of a step signal which extends over infinite time, its frequency spectrum is a single vertical line. This fact shows that we can always localise a signal in time or in frequency but not both simultaneously. If a signal has a short duration, its band of frequency is wide and vice versa.

The Wavelet Transform offers a number of important benefits, including the following:

- The Wavelet transform has the ability to concurrently extract local spectral and temporal information.
- A selection of different wavelets from a variety to choose

The first significant benefit is one that we have gone over in some detail already. This is most likely the most important advantage of utilising the Wavelet Transform. This may be preferable to employing a method such as a Short-Time Fourier Transform, which needs slicing a signal into segments and then applying a Fourier Transform to each individual segment.

The second essential benefit appears to be more of a logistical consideration. In the end, the most important thing to take away from this is that there is a large variety of wavelets from which to choose in order to get the one that most closely matches the characteristic shape that you are seeking to extract from your signal.

In comparison to the Fourier Transform, the Wavelet Transform has the primary benefit of being able to extract local information that is both spectral and temporal in nature. Analyzing electrocardiogram (ECG) readings, which comprise periodic and transient signals of relevance, is an example of a real-world use of the Wavelet Transform.

As a result, we realised that non-stationary signals are the ideal candidates for the use of the wavelet transform. By applying this transformation, one can obtain a high temporal resolution for high-frequency components while maintaining a decent frequency resolution for low-frequency components. This technique begins with a mother wavelet, which could be a Haar, Morlet, or Daubechies, among other options. After that, the signal is essentially recast as scaled and shifted iterations of the mother wavelet. **We will discuss Haar transformation in the subsequent section 5.5 of this unit**

Important points:

- The wavelet transform is used to decompose a time series; this results in waves that are not only localised in frequency but also in time.
- One of the most significant drawbacks of the Fourier Transform is that it collects global frequency information, which refers to frequencies that are present throughout an entire signal. There are some applications, such as electrocardiography (ECG), in which the signals include brief intervals of distinctive oscillation, that this form of signal decomposition would not suit very well. The Wavelet Transform is an alternate method that may be used, and what it does is it decomposes a function into a group of wavelets.
- A simple comparison between Wavelet Transform and Fourier Transform: The Fourier transform can be thought of as a special case of the wavelet transform. A function is decomposed by the Fourier transform into sine and cosine waves, which serve as the base functions. Although the period lengths of the sine and cosine waves change, the base functions remain the same across the entire interval. The wavelet transform, on the other hand, applies scaling as well as shifting to the basic functions.

Also, the sine and cosine waves are not required to be the base functions, despite the fact that there are some well-known base functions. Instead, we are free to choose whichever functions we want to use as base functions, so long as they fulfills the fundamental condition of a wavelet, which is that it has a finite amount of energy.

5.5 HAAR TRANSFORM

The Haar transform is a wavelet transform. Wavelet transforms are based on small waves called wavelets which are of varying frequencies and limited duration. These are different from the Fourier transform, where the basis functions are sinusoids. Haar transform is a transform whose basis functions are orthonormal wavelets. The Haar transform can be expressed as

$$T = HFH^T \tag{13}$$

where, F is an $N \times N$ image matrix, H is the $N \times N$ Haar transform matrix and T is the resulting $N \times N$ transform.

The Haar transform, H, contains the Haar basis functions, $h_k(t)$. They are defined on a continuous interval, $t \in [0,1]$ for $k = 0,1,...,N-1$, where $N = 2^n$. Then, H is generated by uniquely decomposing the integer k as $k = 2^p + q - 1$, where, $0 \le p \le n-1$ and when $p = 0, q = 0,1; p \ne 0$ then, $1 \le q \le 2^n$.

For example, when $N = 4$, k will take the values $k = 0,1,2,3$. For these the corresponding values of p and q have to satisfy that $k = 2^p + q - 1$.

Therefore, we compute the values of k, p and q in Table 1.

Table 1

k	0	1	2	3
p	0	0	1	1
q	0	1	1	2

Let t take the values from the set $\left\{\frac{0}{N}, \frac{1}{N}, ..., \frac{N-1}{N}\right\}$.

Then, the Haar basis functions are recursively defined as:

- For $k = 0$, the Haar function is defined as a constant

$$h_0(t) = 1/\sqrt{N} \tag{14}$$

- When $k > 0$, the Haar function is defined by

$$h_k(t) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & ; \quad \frac{(q-1)}{2^p} \le t < \frac{(q-0.5)}{2^p} \\ -2^{p/2} & ; \quad \frac{(q-0.5)}{2^p} \le t < \frac{q}{2^p} \\ 0 & ; \quad \text{othewise} \end{cases} \qquad (15)$$

As can be seen from the definition of the Haar basis functions, for the non-zero part of the function, the amplitude and width is determined by p while its position is determined by q.

We now show how the Haar transform matrix can be computed at $n = m/N$, where $n = 0,1,...,N-1$ to form the $N \times N$ discrete Haar transform matrix through the following examples.

Example 5: For, $N = 2$, compute the discrete Haar transform of a 2×2 matrix.

Solution: Here, $N = 2$, we know that $N = 2^n$.

Substituting the value of N, we get $2 = 2^n$, which gives $n = 1$.

Since, $0 \le p \le n-1$, we get $0 \le p \le 0$.

Therefore, $p = 0$, and hence $q = 0, 1, 2$.

We determine the value of k using the relation $k = 2^p + q - 1$, we obtain

p	0	0
q	0	1
k	0	1

for $k = 0, h_0(t) = \frac{1}{\sqrt{2}}$ [using Eqn. (14)]

for $k = 1, h_1(t) = \frac{1}{\sqrt{2}} \begin{cases} 1 & ; t = 0 \\ -1 & ; t = 1/2 \end{cases}$

Thus, Haar transform is $h_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

***

Example 6: For $N = 8$, the 8×8 discrete Haar transform matrix.

Solution: As you know we need to find various parameters of Haar transform. So, we find them as follows:

i) Here $N = 8$,

ii) $N = 2^n \Rightarrow n = 3$

iii) when $p = 0,\ q = 0,1$

$p = 1,\ q = 1,2$

$p = 2, q = 1,2,3,4$

iv) All the possible values of k for each set of p and q are given below:

p	0	0	1	1	2	2	2	2
q	0	1	1	2	1	2	3	4

k	0	1	2	3	4	5	6	7

v) Accordingly, $t=\left\{0,\frac{1}{8},\frac{2}{8},\frac{3}{8},\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8}\right\}$.

Now, we compute $h_k(t)$ for each k and t.

For $k=0$, $h_0(t)=0$. for all t.

In general, $$h_k(t)=\frac{1}{\sqrt{8}}\begin{cases}2^{p/2} & ; \frac{q-1}{2^p}\le t<\frac{q-0.5}{2^p}\\ -2^{p/2} & ; \frac{q-0.5}{2^p}\le t<\frac{q}{2^p}\\ 0 & ; \text{otherwise}\end{cases} \quad (16)$$

Now, let us find each $h_k(t)$ for each of the interval of t for a particular k using Eqn. (16) in the following table:

For $k=1$

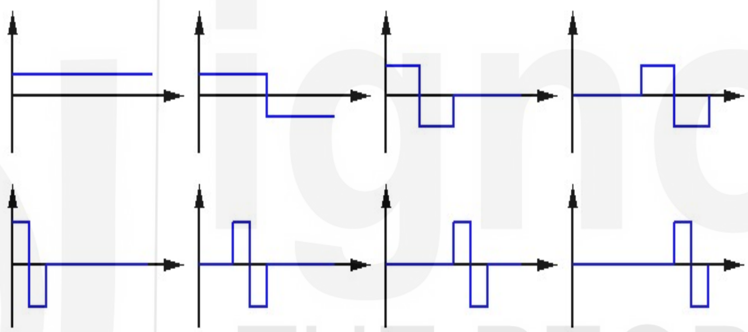
Parameters $\mathbf{k, q, p}$	$\mathbf{h_k(t)}$	**Haar Transform after simplification**
$k=1$, $q=1$, $p=0$	$h_1(t)=\frac{1}{\sqrt{8}}\begin{cases}1; & \frac{0}{2^0}\le t<\frac{1-0.5}{2}\Rightarrow 1\le t<\frac{1}{2}\\ -1; & \frac{1-0.5}{2}\le t<\frac{1}{2^0}\Rightarrow\frac{1}{2}\le t<1\\ 0; & \text{otherwise}\end{cases}$	$h_1(t)=\frac{1}{2\sqrt{2}}$; for $t=0,\frac{1}{8},\frac{2}{8},\frac{3}{8}$ $h_1(t)=\frac{-1}{2\sqrt{2}}$; for $t=\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8}$
$k=2$, $q=1$, $p=1$	$h_2(t)=\frac{1}{\sqrt{8}}\begin{cases}\sqrt{2} & ; 0\le t<\frac{1}{4}\\ -\sqrt{2} & ;\frac{1}{4}\le t<\frac{1}{2}\\ 0 & ; \text{otherwise}\end{cases}$	$h_2(t)=\frac{1}{2}$; for $t=0,\frac{1}{8}$ $h_2(t)=\frac{-1}{2}$; for $t=\frac{2}{8},\frac{3}{8}$ $h_2(t)=0$; for $t=\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8}$.
$k=3$, $q=2$, $p=1$	$h_3(t)=\frac{1}{\sqrt{8}}\begin{cases}\sqrt{2} & ; \frac{1}{2}\le t<\frac{3}{4}\\ -\sqrt{2} & ;\frac{3}{4}\le t<1\\ 0 & ; \text{otherwise}\end{cases}$	$h_3(t)=\frac{1}{2}$; for $t=\frac{4}{8},\frac{5}{8}$ $h_3(t)=\frac{-1}{2}$; for $t=\frac{6}{8},\frac{7}{8}$

		$h_3(t)=0$; for $t=0,\frac{1}{8},\frac{2}{8},\frac{3}{8}$.
$k=4,$ $q=1,$ $p=2$	$h_4(t)=\frac{1}{\sqrt{8}}\begin{cases}2 \;; 0\le t<\frac{1}{8}\\ -2 \;;\frac{1}{8}\le t<\frac{1}{4}\\ 0 \;; \text{otherwise}\end{cases}$	$h_4(t)=\frac{1}{\sqrt{2}}$; for $t=0$ $h_4(t)=\frac{-1}{\sqrt{2}}$; for $t=\frac{1}{8}$ $h_4(t)=0$; for $t=\frac{2}{8},\frac{3}{8},\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8}$
$k=-5,$ $q=2,$ $p=2$	$h_5(t)=\frac{1}{\sqrt{8}}\begin{cases}2 \;;\frac{1}{4}\le t<\frac{3}{8}\\ -2 \;;\frac{3}{8}\le t<\frac{1}{2}\\ 0 \;; \text{otherwise}\end{cases}$	$h_5(t)=\frac{1}{\sqrt{2}}$; for $t=\frac{2}{8}$. $h_5(t)=\frac{-1}{\sqrt{2}}$; for $t=\frac{3}{8}$ $h_5(t)=0$; for $t=0,\frac{1}{8},\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8}$
$k=6,$ $q=3,$ $p=2$	$h_6(t)=\frac{1}{\sqrt{8}}\begin{cases}2 \;;\frac{1}{2}\le t<\frac{5}{8}\\ -2 \;;\frac{5}{8}\le t<\frac{3}{4}\\ 0 \;; \text{otherwise}\end{cases}$	$h_6(t)=\frac{1}{\sqrt{2}}$; for $t=\frac{4}{8}$ $h_6(t)=\frac{-1}{\sqrt{2}}$; for $t=\frac{5}{8}$ $h_6(t)=0$; for $t=0,\frac{1}{8},\frac{2}{8},\frac{3}{8},\frac{6}{8},\frac{7}{8}$
$k=7,$ $q=4,$ $p=2$	$h_7(t)=\frac{1}{\sqrt{8}}\begin{cases}2 \;;\frac{3}{4}\le t<\frac{7}{8}\\ -2 \;;\frac{7}{8}\le t<1\\ 0 \;; \text{otherwise}\end{cases}$	$h_7(t)=\frac{1}{\sqrt{2}}$; for $t=\frac{6}{8}$ $h_7(t)=\frac{-1}{\sqrt{2}}$; for $t=\frac{7}{8}$ $h_7(t)=0$; for

		$t = 0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}.$

The Haar transform is given in the following matrix.

$$h_k(t) = \begin{bmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{-1}{2\sqrt{2}} & \frac{-1}{2\sqrt{2}} & \frac{-1}{2\sqrt{2}} & \frac{-1}{2\sqrt{2}} \\ \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{-1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{-1}{2} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

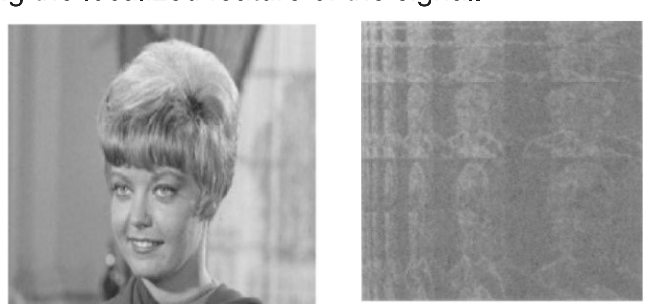


The plot of these 8 basis functions are shown in Fig. 6.

Fig. 6: Haar Basis Functions

As can be seen by Fig.6, all non-zero Haar functions $h_k(t), k > 0$ consists of a square wave and its negative version, and the parametersp defines the magnitude and width of the shape while q specifies the position (or shift) of the shape. This gives the unique property to the Haar transform that it not only represents the signal at different scales based on the different frequencies, but also represents their locations across time.

Moreover, an important property of the Haar transform matrix is that it is real and orthogonal, that is, $H = H^*$ and $H^{-1} = H^T$. The orthogonal property of the Haar transform allows the analysis of the frequency components of the input signal. The Haar transform can also be used for analyzing the localized feature of the signal.



a) Original Image **b) Output of Haar Transform**

Fig. 7: Haar Transform

Fig. 7 (b) shows the output of Haar Transform of the image in Fig. 7 (a).

Try the following exercises.

E7) Let $X = [x[0], x[1], x[2], x[3]]^T = [1, 2, 3, 4]^T$. Then, X is a 4-point signal. Find the Haar transform coefficients and show that the signal can be expressed as a linear combination of the basis functions by the inverse transform.

E8) For $N = 4$, compute h_4, which represents the 4×4 discrete Haar transform matrix.

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

3.5 SUMMARY

In this unit, we discussed transformations which convert the spatial domain image to the frequency domain. We saw that these transform provide a variety of information based on the frequency content of the image.We discussed in depth three very important image transforms, namely the Discrete Fourier transformation (DFT), the Discrete Cosine Transformation (DCT) and the Haar transform. We also discussed the properties of each of these transforms, which shall help us in using them for image filtering in the frequency domain.

3.6 SOLUTIONS AND ANSWERS

E1) Consider an image $f(x, y)$ of size $M \times N$ and the generic image transform T where, x indicates row and y indicates column.

Then,

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

If, T is separable and symmetric, then we can write $g(u, v)$ as

$$g(u, v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} T_1(u, x) T_2(v, y) f(x, y)$$

$$\Rightarrow g(u, v) = \sum_{x=0}^{M-1} T_1(u, x) \sum_{y=0}^{N-1} T_2(v, y) f(x, y)$$

Then, $\sum_{y=0}^{N-1} T_2(v, y) f(x, y)$ is the same as applying the one-dimensional transform long the x-th row of the image. By doing this for each of the M rows of the image, we obtain an intermediate image $F(x, v)$.

Then, $g(u, v) = \sum_{x=0}^{M-1} T_1(u, x) F(x, v)$

Therefore, we can see, the above sum corresponds to applying the one dimensional transform along the v-th column of the intermediate image (x, v).

Therefore, we have shown that the implementation of a separable and symmetric transform in an image requires the sequential implementation of the corresponding one-dimensional transform row-by-row and then column-by-column (or the inverse). We explain this in the Fig. 9:

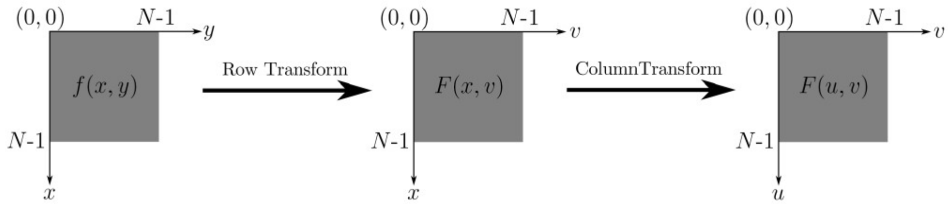


Fig. 9

E2) $$g'(u) = \frac{1}{4}\sum_{x=0}^{3} f(x) e^{\frac{-i\cdot 2\pi ux}{4}}; \quad u = 0, 1, 2, 3$$

$$= \frac{1}{4}[i + i(-i)^{2u} + (-i)^{3u}]; \; u = 0, 1, 2, 3$$

$$= \frac{1}{4}[i + i(-1)^{2u} + (i)^{u}]; \; u = 0, 1, 2, 3$$

$$\mathbf{g} = \frac{1}{4}[1 + 2i, i, -1 + 2i, -i].$$

E3) Here $N = 4$.

$$g(u) = \frac{1}{4}\sum_{x=0}^{3} f(x) e^{\frac{-i2\pi\cdot ux}{4}}; \; u = 0, 1, 2, 3$$

$$g(u) = \frac{1}{4}[f(0) + (-i)^{u} f(1) + (-i)^{2u} f(2) + (-i)^{3u} f(3)]; \; u = 0, 1, 2, 3.$$

$$g(0) = \frac{1}{4}[f(0) + f(1) + f(2) + f(3)]$$

$$g(1) = \frac{1}{4}[f(0) - i f(1) + f(2) + i f(3)]$$

$$g(2) = \frac{1}{4}[f(0) - f(1) + f(2) - f(3)]$$

$$g(3) = \frac{1}{4}[f(0) + i f(1) - f(2) + i f(3)]$$

Hence, the DFT matrix is $A = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}$.

You may check if $A \times A^{*T} = I.$

E4) Using Eqn. (5), $f(x,y) = \frac{1}{2\times 2}\sum_{x=0}^{1}\sum_{y=0}^{1} F(u,v).e^{2\pi i\left(\frac{ux}{2}+\frac{vy}{2}\right)}; u,v = 0,1$

$$= \frac{1}{4}\sum_{x=0}^{1}\sum_{y=0}^{1} F(u,v)(1)^{ux}(1)^{vy}; u,v = 0,1$$

$$= \frac{1}{4}\left[f(0,0) + (1)^{v}F(0,1) + (1)^{u}F(1,0) + (1)^{u}(1)^{v}F(1,1)\right]; u,v = 0,1$$

which gives $f(x,y) = \frac{1}{4}\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

E5) In general, in most of the images, large part of the signal energy lies at the low frequencies which appear in the upper left corner of the DCT image. Since the higher frequencies present in the lower right of the image are small enough to be neglected, the original image can be represented in less number of coefficients, thereby achieving compression. Therefore, as DCT has good compaction property, it can represent the original image in less number of coefficients and therefore, storage and transmission of the image is better and faster. Moreover, the original image can be recreated close to the original from the most important components of the DCT.

E6) $$C(u) = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix}$$

E7) $X = [x[0], x[1], x[2], x[3]]^T = [1, 2, 3, 4]^T$ be the 4-point signal. Then, we shall use the basis matrix, H_4 to compute the Haar transform coefficients.

$$\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ -2 \\ -1\sqrt{2} \\ -1\sqrt{2} \end{bmatrix}$$

The inverse transform will be:

$$\frac{1}{2}\begin{bmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{bmatrix}\begin{bmatrix} 5 \\ -2 \\ -1\sqrt{2} \\ -1\sqrt{2} \end{bmatrix}$$

$$= \frac{1}{2}\left[5\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - 2\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} - \frac{1}{\sqrt{2}}\begin{bmatrix} \sqrt{2} \\ -\sqrt{2} \\ 0 \\ 0 \end{bmatrix} - \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 0 \\ \sqrt{2} \\ -\sqrt{2} \end{bmatrix}\right] = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

As can be seen, X is a linear combination of the basis vectors.

E8) Here $N = 4; n = 2; p = 0,1; t = \left\{0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}\right\}$. Let us write all the values of Haar transform in the following table:

k, q, p	$h_k(t)$	$h_k(t)$ after simplification
$p = 0, q = 0, k = 0$	$h_0(t) = \frac{1}{\sqrt{4}} = \frac{1}{2}$ for all t	$h_0(t) = \frac{1}{2}$ for $t = 0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}$.
$p = 0, q = 1, k = 1$	$h_1(t) = \frac{1}{2}\begin{cases} 1; \ 0 \le t < \frac{1}{2} \\ -1; \frac{1}{2} \le t < 1 \\ 0; \text{otherwise} \end{cases}$	$h_1(t) = \frac{1}{2}; t = 0, \frac{1}{4}$ $h_1(t) = \frac{-1}{2}; t = \frac{2}{4}, \frac{3}{4}$
$p = 1, q = 1, k = 2$	$h_2(t) = \frac{1}{2}\begin{cases} \sqrt{2}; 0 \le t < \frac{1}{4} \\ -\sqrt{2}; \frac{1}{4} \le t < \frac{1}{2} \\ 0; \text{otherwise} \end{cases}$	$h_2(t) = \frac{1}{\sqrt{2}}; t = 0$ $h_2(t) = \frac{-1}{\sqrt{2}}; t = \frac{1}{4}$ $h_2(t) = 0; t = \frac{2}{4}, \frac{3}{4}$.
$p = 1, q = 2, k = 3$	$h_3(t) = \frac{1}{2}\begin{cases} \sqrt{2}; \frac{1}{2} \le t < \frac{3}{4} \\ -\sqrt{2}; \frac{3}{4} \le t < 1 \\ 0; \text{otherwise} \end{cases}$	$h_3(t) = \frac{1}{\sqrt{2}}; t = \frac{2}{4}$ $h_3(t) = \frac{1}{\sqrt{2}}; t = \frac{3}{4}$ $h_3(t) = 0; t = 0, \frac{1}{4}$.

Hence,

$$h_k(t) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{-1}{1} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

UNIT 6

IMAGE ENHANCEMENT & FILTERING IN FREQUENCY DOMAIN

Structure **Page No.**

6.1 INTRODUCTION

In the previous units of this course, we have considered an image in the spatial domain which is the form in which camera captures it. In this unit, we shall view the image as a signal and apply the well known filtering techniques used in signal processing. The only difference here will be that image would be considered as 2-D signal (along x and y axes). We shall see that this view of the image results in number of benefits over the spatial domain treatment.

In this unit, we will discuss various enhancement techniques in frequency (fourier) domain. We discuss the basic issues associated with frequency domain filtering. We also discuss various low pass and high pass filters in frequency domain with their applications and advantages in image enhancement.

By discussing the enhancement techniques in frequency (fourier) domain, we have looked at image improvement without bothering about source which caused a degradation in the quality of the image. If the source is known to us, it is possible to improve the quality of the image in a better way. Thus, it is required to discuss the concept of Image restoration/degradation.

Image restoration is a pre-processing method that suppresses a known **degradation**. Image acquisition devices introduce degradation because of defects in optical lenses, non- linearity of sensors, relative object camera motion, blur due to camera mis-focus, atmospheric turbulence etc. Restoration tries to reconstruct an image that was degraded by a known degradation function. Iterative restoration techniques attempt to restore an image by minimizing some parameter of degradation, whereas blind restoration techniques attempt improve the image without knowing the degradation function. Like image enhancement, image restoration also aims to improve image quality, but it is more objective process where as enhancement is a subjective process. Noise is visually unpleasant, it is bad for compression and bad for analysis. Restoration involves modeling of these degradations and applying inverse process to recover the original image.

So we learned that Image restoration is the process of retrieving an original image from degraded image. The idea is to obtain an image as close to the original image as possible. This is possible by removing or minimizing degradations. This is often difficult in case of extreme noise and blurs, and often called an **inverse problem**. An inverse problem aims to find the cause and extent of degradation.

In this unit we will also learn that the image Restoration involves modeling of the degradations and by applying inverse process we can recover the original image. For the restoration process, it is mandatory that we estimate the degradation process accurately. Else we will not be able to remove it.

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 images in frequency domain
- perform filtering in frequency domain
- apply different types of image smoothing filters
- apply different types of image sharpening filters
- describe images degradation models
- state difference between restoration and enhancement
- apply different noise models
- estimate degradation function
- apply Inverse filtering
- apply Wiener filtering

Let us begin with shifting the centre of the spectrum.

6.2 BASICS OF FILTERING IN FREQUENCY DOMAIN - SHIFTING THE CENTRE OF THE SPECTRUM

To start with we understood that any signal (periodic or non periodic) can be expressed as the summations of sines and/or cosines multiplied by a weighting function. This is carried out by applying Fourier transform on the image.

In 1D signal, the fourier transform takes the form

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx$$

$F(u)$ can be expressed in polar coordinates:

$F(u) = |F(u)| e^{j\phi(u)}$

where $|F(u)| = \left[R^2(u) + I^2(u)\right]^{\frac{1}{2}}$ (magnitude or spectrum)

$\phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$ (phase angle or phase spectrum)

$R(u)$: the real part of $F(u)$

$I(u)$: the imaginary part of $F(u)$

The various benefits of frequency domain analysis are the following:

1) It is convenient to design a filter in frequency domain. As filtering is more intuitive in frequency domain, designing an appropriate filter is easier.
2) Implementation is very efficient with fast DFT via FFT.
3) Convolution in spatial domain reduces to multiplication in frequency domain which is a much simpler process.

However, the image in spatial domain is not continuous but consists of discrete values. The discrete version of fourier transform is

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x) e^{\frac{-2\pi ux}{N}}, u = 0, 1, ..., N-1$$

Also, since the image is two dimensional signal, we need 2D Fourier transform. For a $N \times N$ image it takes the form:

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y) e^{-j2\pi\left(\frac{ux+vy}{N}\right)},$$

where u and v are the frequencies along x and y axes and take the values $0, 1, 2, ..., N-1$.

In the spatial domain we consider the origin to be located at top left corner of the image. For better display in the frequency domain, it is common to shift the origin to centre of the image.

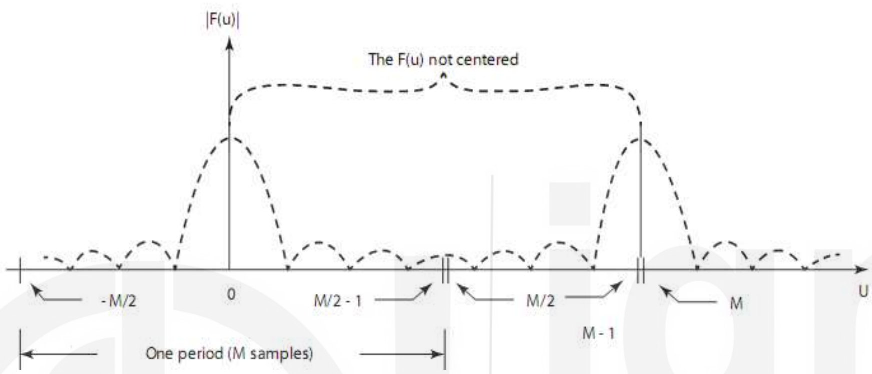
Periodicity of Fourier transform is given by

$$v(k,l) = v(k+M,l) = v(k,l+N) = v(k+M,l+N) \quad (1)$$

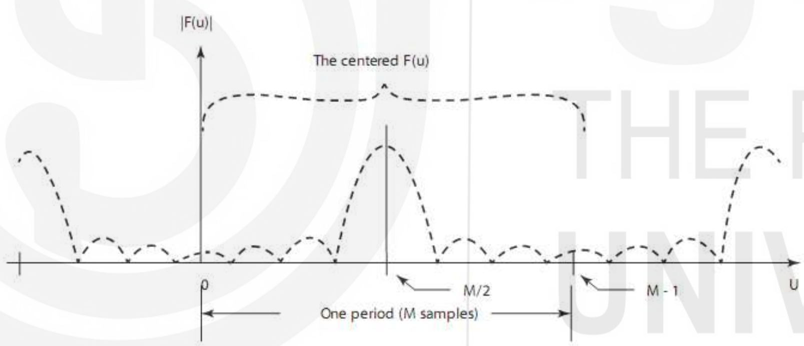
$$u(m,n) = u(m+M,n) = u(m,n+N) = u(m+M,n+N) \qquad (2)$$

Fig 1(a) shows that the values from $N/2$ to $N-1$ are the same as the value from $N-1$ to 0. As DFT has been formulated for value of k in the interval $[0, N-1]$, the result of this formulation yield two back to back half periods in this interval. To display one full interval between 0 to $N-1$ as shown in Fig. 1(b), it is necessary to shift the origin of transform to the point $k = N/2$. To do so we have to take advantage of translation property of Fourier transform.

$$v(m,n)(-1)^{m-n} \xleftrightarrow{FT} v\left(k-\frac{M}{2}, l-\frac{N}{2}\right) \qquad (3)$$



(a) Spectrum of f(x) without shifting centre.



(b) Spectrum of f(x) after shifting centre.

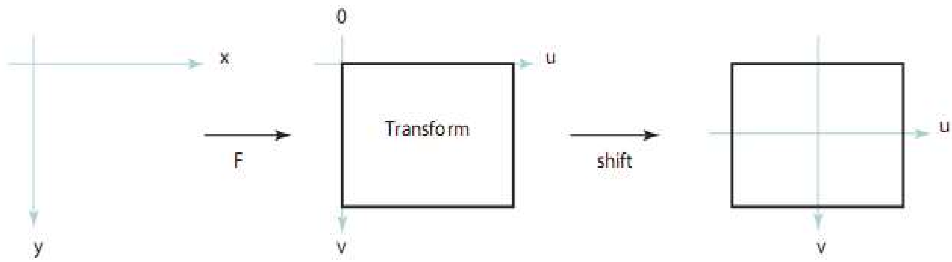
Fig. 1

Fig. 2 (a) and (b) show how the origin shifts from left corner of the image to centre of the image.

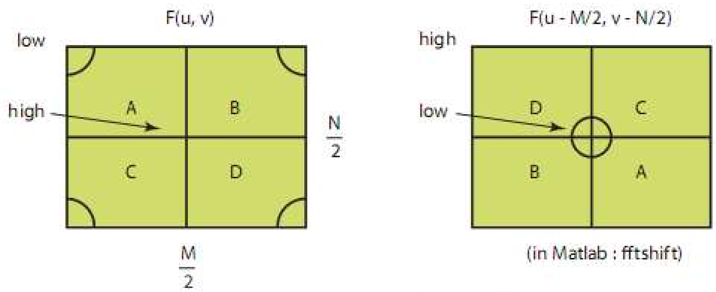
Basic Property of images in Frequency Domain

The forward transform of input image $u(m,n)$ is given by

$$v(k,l) = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{m=0}^{N-1} u(m,n)\, W_N^{km}\, W_N^{ln} \qquad 0 \le k,l \le N-1 \qquad (4)$$



(a) Change of centre in the spectrum of an image.



(b) Change of centre in the spectrum of an image.

Fig. 2

Following properties of the Fourier transform are observed

i) Each term of $v(k,l)$ contains all the values of $u(m,n)$ modified by the values of exponential terms.

ii) Frequency is directly related to the rate of change of grey level values.

iii) DC value or the average grey level value in an image is the slowest varying components corresponding to $u = 0, v = 0$. It is also the largest component in the frequency domain.

iv) Smooth variation of grey levels corresponds to low frequency components. Slow varying components can be the background of an image, hair of a person, skin, or texture etc.

v) Faster grey level changes correspond to high frequency components. These can be the edges/boundary of the objects or noise present in the image.

vi) As we move away from origin, higher frequencies are encountered.

Fig. 3 shows the variation in frequency of a centred spectrum.

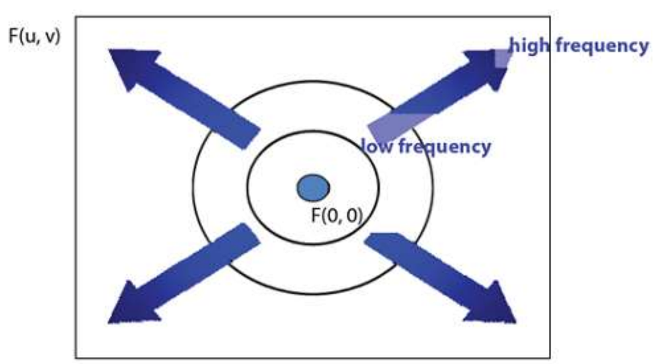


Fig 3: Frequency variation in an image.

Also, note that the rotation of an image in spatial domain causes exactly same rotation in frequency domain.

- Rotating $f(x,y)$ by θ rotates $F(u,v)$ by θ.

Once the image is transformed in frequency domain, it is easy to carry out image processing operations on it. We apply a low pass filter, if we are interested in only slowly varying components of the image (like object shapes), and we wish to suppress high frequency components (like noise). If we are interested in highlighting the edges or special textures, we can employ high pass filters, which will allow high frequency components to be displayed. Filtering in frequency domain is multiplication of a suitable filter $H(u,v)$ by image in Fourier domain $F(u,v)$ to result in $G(u,v)$. By taking inverse Fourier Transform of $G(u,v)$ we get the image back in spatial domain.

Generally, the filters are centred and are symmetric about the centre. Input image should also be centred. Following steps are followed to carry out filtering in frequency domain (Fig. 4):

Step 1: Multiply input image $f(x,y)$ by $(-1)^{x+y}$ to move the origin in the transformed image to $u = \frac{M}{2}$ and $v = \frac{N}{2}$

Step 2: Compute $F(u,v)$, Fourier transform of the output of step 1.

Step 3: Multiply filter function $H(u,v)$ to $F(u,v)$ to get $G(u,v)$.

Step 4: Take inverse Fourier transform of $G(u,v)$ to get $g(x,y)$.

Step 5: Take the real part of $g(x,y)$ to get $g_r(x,y)$

Step 6: Multiply the result of step 5 by $(-1)^{x+y}$ to shift the centre back to origin and enhanced image is generated.

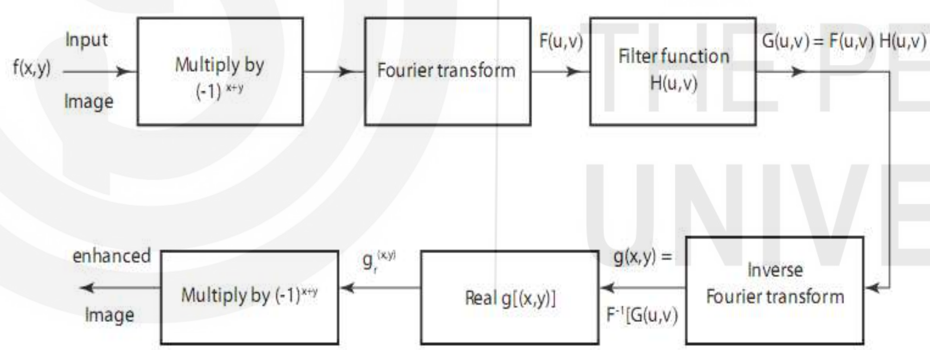


Fig. 4: Block Diagram of Filtering in Frequency Domain.

Types of Frequency Domain Filters

Frequency domain filters are categorized into three types.

1. Smoothing filters
2. Sharpening filters
3. Homomorphic filters

Smoothing filters are low pass filters and are used for noise reduction. It blurs objects. Sharpening filters are high pass filters and produce sharp images with dark background. Laplacian and high boost filters are used to produce sharp images. Homomorphic filters are based on illumination and reflectance model, and create a balance between smoothing and sharpening filtering effect. This classification is shown in Fig. 5.

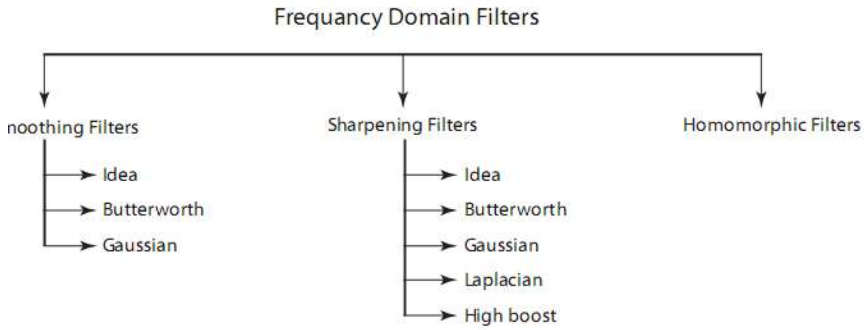


Fig. 5: Types of Frequency Domain Filters.

Try the following exercises.

E1) Write the steps involved in frequency domain filtering with the help of block diagram.

E2) Explain how image enhancement is better in the frequency domain as compared to spatial domain.

In the following section, we will discuss smoothing filters in the frequency domain.

6.3 SMOOTHING FILTERS

Smoothing filters are low pass filters (LPF). Edges, sharp transitions and noise in the grey levels contribute to high frequency contents in an image. A low pass filter only passes low frequency and blocks the high ones. It removes noise but in the process introduces blurring as a side effect in the image. The basic model of filtering is

$$G(u,v) = H(u,v)\,F(u,v) \tag{5}$$

where $F(u,v) =$ Fourier transform of the image to the filtered, $H(u,v) =$ Transfer function of the filter, and $G(u,v) =$ Enhanced image where high frequency components have been alternated.

The transfer function $H(u,v)$ is of three types

a) Ideal LPF
b) Butterworth LPF
c) Gaussian LPF

Ideal filters has sharp slope in transition band whereas Gaussian filter has smooth slope in transition. Butterworth filter has a parameter called filter order which controls the slope of transition band. Higher value of filter order leads to ideal filter.

6.3.1 Ideal Low Pass Filters (ILPF)

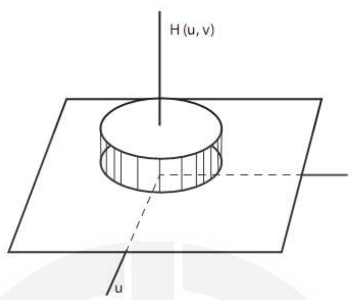
Low pass filter removes all frequencies above a certain frequency components D_0. Ideal low pass filter is defined by the transfer function

$$H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) > D_0 \end{cases}$$

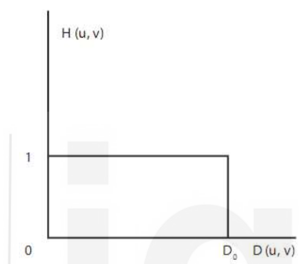
Where $D(u,v)=\left[\left(u-\frac{M}{2}\right)^{1/2}+\left(v-\frac{N}{2}\right)^{1/2}\right]$

$D(u,v)$ is the distance from point (u,v) to the centre $\left(\frac{M}{2},\frac{N}{2}\right)$. If size of an image is $M \times N$, then the centre is at $\left(\frac{M}{2},\frac{N}{2}\right)$. Filter transfer function is symmetric about the midpoint.

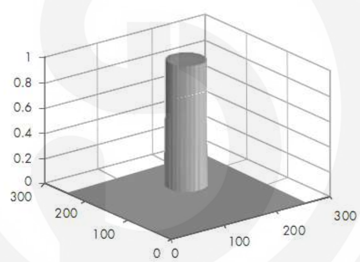
D_0 is non negative quantity specifying the frequency content to be retained. It is also called cut off frequency. Fig. 6(a), Fig. 6(b) is the plot of ILPF and Fig. 6 (c) is perspective plot and Fig. 6(d) is filter displayed as an image.



(a) Transfer Function



(b) Plot of Ideal LPF



c) Prospective plot of ILPF

(d) ILPF displayed as an image with D_0 = 10

Fig. 6

Choice of cut off frequency in an ideal LPF

1. The cutoff frequency D_0 decides the amount of frequency components passed by the filter.

2. Smaller the value of D_0, more are the number of frequency components eliminated by the filter.

3. In general, D_0 is chosen such that most of the frequency components of interest are passed while unnecessary components are eliminated.

4. A useful way to establish a set of standard cut off frequencies is to compute circles having a certain percentage of the total image power.

$$P_T=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}P(u,v)$$

Here $P(u,v) = |F(u,v)|^2$ is the total image power.

5. Consider a circle of radius $D_0(\alpha)$ as the cut off frequency with respect to a threshold a such that

$$\sum_u \sum_v P(u,v) = \alpha P_T .$$

6. Thus, we can fix a threshold α which tells how much of the total energy is retained and obtain an appropriate cut off frequency $D_0(\alpha)$.

Properties of ILPF

As it is an ideal filter, it is non-real, non-actual and non-physical. But it can be stimulated in computers. Fig 7 and 8(b) to (c) show low pass filtered images with different cut off frequencies. As the filter radius increases, less and less power and information is removed which resulted in less blurring. A very noticeable effect that can be seen in the output image is ringing.

Now, the question arises that what is ringing?

Ringing is undesirable and unpleasant lines around the objects present in the image Fig. 7 (b). As the cut of frequency D_0 increases, effect of ringing reduces. Ringing is a side effect of ideal lpf.

Why is there Ringing in Ideal LPF?

Ideal LPF function is a rectangular function as shown in Fig. 6-X. The inverse Fourier transform of a rectangular function, is a sinc function. We can observe two distinctive characteristic of $\sin c$ function:

1. A dominant component at the origin which is responsible for blurring.
2. Concentric circular components are responsible for ringing which is the characteristic of an ideal LPF.

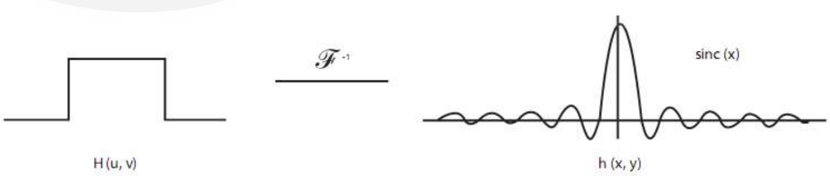


Fig. 6-X: Fourier Inverse of Rectangular Transfer Function

Radius of the centre component $\infty \dfrac{1}{\text{cut off frequency}}$

Number of circles per unit distance from origin $\infty \dfrac{1}{\text{cut off frequency}}$

Thus, as the cut off frequency (D_0) is increased, blurring as well as ringing reduces. The examples are given in Fig. 7 and Fig. 8.

(a) Original image **(b) Output of ILPF with D_0=30** **(c) Output of ILPF with D_0= 50**

Fig. 7

(a) Original image **(b) Output of ILPF with D_0=50** **(c) Output of ILPF with D_0=80**

Fig. 8

6.3.2 Butterworth Low Pass Filters (BLPF)

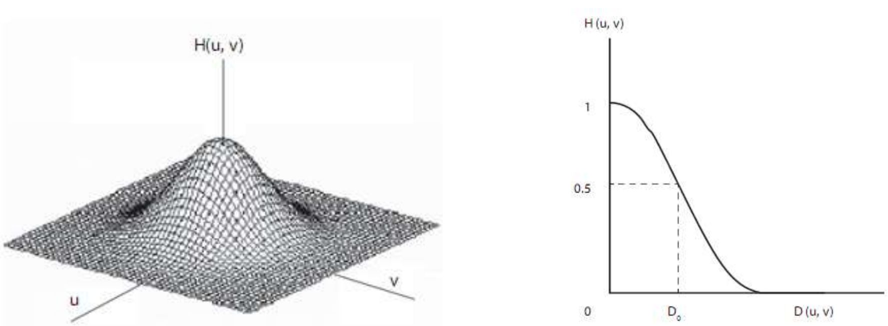
The Butterworth filter replaces the sharp cutoff of Ideal LPF by a smooth cutoff. Frequency response of BLPF does not have a sharp transition between pass band and stop band. It is more appropriate for image smoothing and does not introduce ringing effect for lower order filters. Transfer function of BLPF is given by

$$H(u,v) = \frac{1}{1+\left[D(u,v)/D_0\right]^{2n}},$$

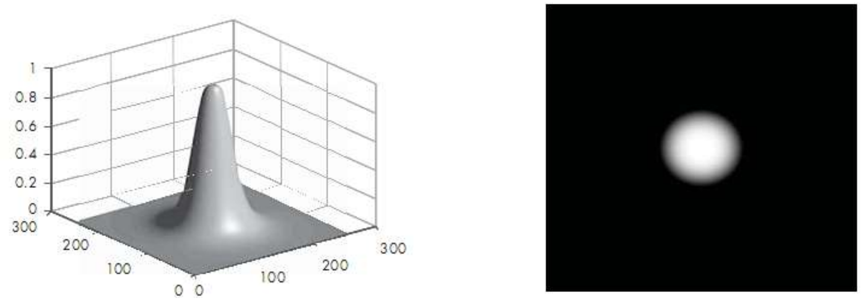
where the cut off frequency or distance from the centre $D_0 = \left(\frac{M}{2}, \frac{N}{2}\right)$, and

the filter order is n, and $D(u,v) = \left[\left(u-\frac{M}{2}\right)^2 + \left(u-\frac{N}{2}\right)^2\right]^{1/2}$

Fig. 9 (a) and Fig. 9 (b) show the transfer function of BLPF. Fig. 9 (c) is the plot of BLPF and Fig. 9 (d) is BLPF displayed as an image.



(a) Plot of BLPF **(b) Plot of BLPF transfer function**



(c) Plot of BLPF **(d) BLPF displayed as an image**

Fig. 9

Transfer function of BLPF does not have sharp transition near the cut off. For $n = 1$, the transition is very smooth. As the filter order increases, the transfer function approaches towards ideal LPF. No ringing is visible on the image filtered by BLPF for $n = 1$. Noise is reduced and blurring is observed in all the images. For $n = 2$, ringing is un-noticeable, but it can becomes more significant for higher values of n. Fig. 10 shows the increasing effect of ringing as n increases from 1 to 20.

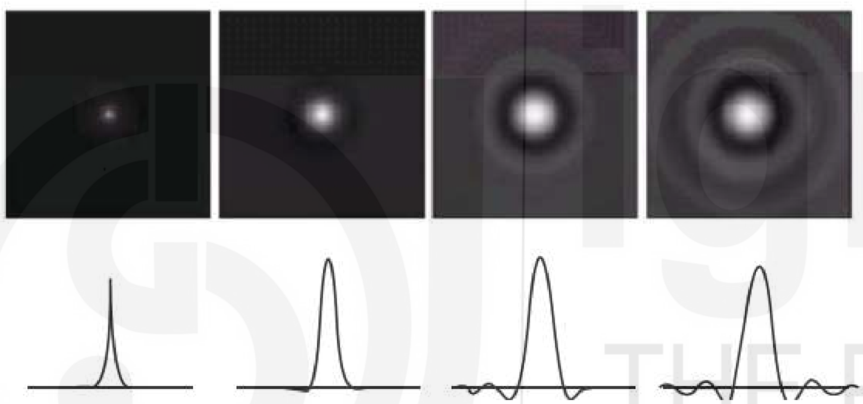
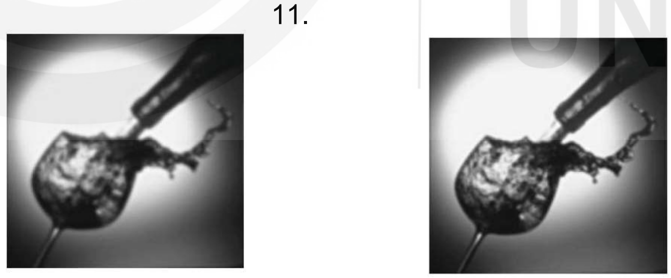
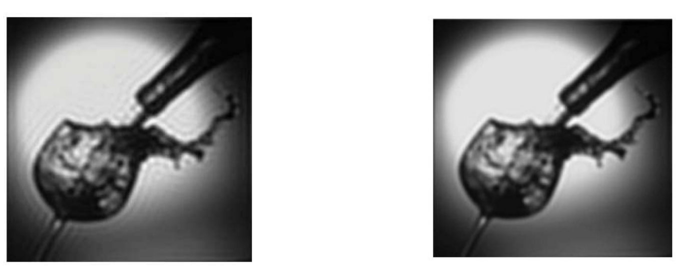


Fig. 10: Spatial Representation of BLPF of order 1, 2, 5 and 20 and Corresponding Intensity Profile

The output corresponding to the change in the values of D_o and n are shown in Fig. 11.



(a) Output of BLPF for D_0= 30 **(b) Output of BLPF for D0= 40**



(c) Output of BLPF for n = 4, D_0 = 30 **(d) Output of BLPF for n = 20 D_0 = 30**

Fig. 11

6.3.3 Gaussian Low Pass filters (GLPF)

Still a better variant of the low pass filter is the Gaussian Low Pass filter which have smooth transition between pass band and stop band. It does not introduce any ringing in the output image. The transfer function of GLPF is given by

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2},$$

where $D(u,v)$ is the distance from the origin of Fourier transform, and σ is the measure of spread/dispersion of the Gaussian curve.

Larger the value of σ, larger is the cut off frequency and the filter is milder. Let $\sigma = D_0$ then transfer function is given by

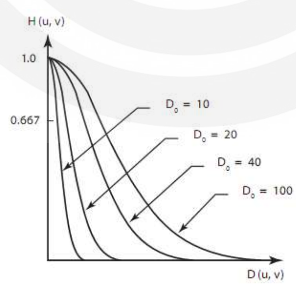
$$H(u,v) = e^{-D^2(u,v)/2{D_0}^2},$$

where D_0 is the cut off frequency.

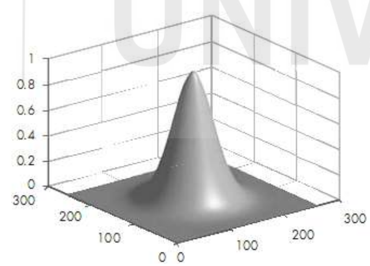
When $D(u,v) = D_0$, the amplitude of transfer function is down to 0.607 of its maximum value of 1.

$D_0 \uparrow$	= Milder filter, more components are passed
$D_0 \uparrow$	= $\downarrow$ blurring

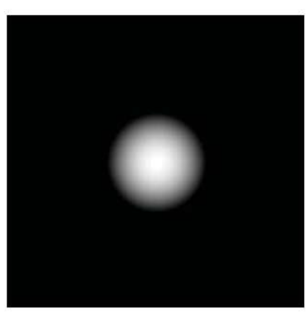
Fig. 12 (a) is GLPF transfer function, Fig. 12 (b) is plot of GLPF and Fig. 12 (c) is GLPF displayed as an image. Fig. 13 (a) to Fig. 13 (c) are GLP filtered images. No ringing is observed in the output, but only blurring is visible. As the cut off frequency increase, blurring reduces. No ringing in the output is a very big advantage of GLPF. These filters can be used in situations where no artifacts are desirable (eg. medical imaging). In medical imaging, GLPF is preferred over ILPF/ BLPF.



(a) GLPF Transfer Function for Various Values of D



(b) Plot of GLPF

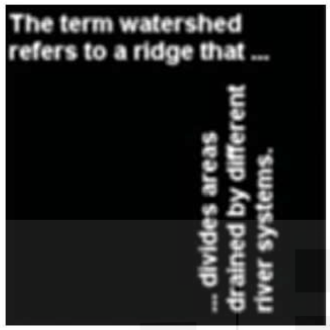


(c) GLPF Displayed as an Image

Fig. 12

(a) Output of GLPF for $D_0 = 10$

(b) Output of GLPF for $D_0 = 300$



(c) Output of GLPF with $D_0 = 50$

Fig. 13

	Ideal	Butter worth	Gaussian
Transfer Function	$H(u,v) = \begin{cases} 1, D(u,v) \le D_0 \\ 0, D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1+[D(u,v)/D_0]^{24}}$	$H(u,v) = e^{-D^2(u,v)/2D^2}$
Applications	Reduce noise	Reduce noise	Reduce noise
Problems	Blurring Ringing	Blurring, Ringing for higher order filters $(n > 2)$	Blurring, No ringing

Let us discuss some of the applications of Low pass filters in frequency domain.

6.3.4 Applications of Low Pass Filters

LPF are generally used as a preprocessing step before an automatic recognition algorithm. It is also used to reduce noise in images. Few examples are listed below.

1. **Character Recognition:** Input to an automatic character recognition system is generally of poor quality. Input may contain noise due to improper acquisition system or there may be gaps in the alphabets. (broken alphabets). Because of these problems, character recognition system fails to give expected results consistently. Hence, LPF is used as a preprocessing step to blur the image. Blurring is used to bridge small gaps in the alphabets. This is done using GLPF with $D_0 = 80$. This increases chances of getting correct result from automatic character recognition system.

2. **Object Counting:** Object counting is to count the number of objects in an image. The output of an object counting algorithm may give wrong output because of poor quantity of input image. If there are small gaps in the boundary of objects, automatic algorithm will not give expected results.

Blurring is used to fill in small gaps in the boundary of objects which helps in producing correct results.

3. **Printing and Publishing Industry:** Unsharp masking is used in publishing industry to sharpen image where blurred version of an image is subtracted from the image itself to get sharpen image.

4. **"Cosmetic"** processing is another use of low pass filter prior to printing. Blurring is used to reduce the sharpness of fine skin lines and small blemishes on human face. Smoothened images look very soft and pleasing and face looks younger.

Try the following exercises.

E3) Give the formula for transform function of a Butterworth low pass filter.

E4) Explain and compare ideal low pass filter and Butterworth filter for image smoothing.

E5) Explain smoothing frequency domain filters. What is ringing effect?

E6) Discuss the applications of image smoothing filters.

In the following section we will discuss sharpening filters.

6.4 IMAGE SHARPENING IN FREQUENCY DOMAIN

In the Fourier transform of an image, high frequency contents correspond to edges, sharp transition in grey levels and noise. Low frequency contents correspond to uniform or slowly varying grey level values.

High pass filtering is achieved by attenuating low frequency components without disturbing high frequency components. High pass filter (HPF) can also be viewed as reverse operation of low pass filter. Transfer function of HPF is given by

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

Here, $H(u,v)$ is the transfer function of a LPF. Thus

Smoothing $\rightarrow$ LPF $\rightarrow$ attenuates high frequency components
Sharpening $\rightarrow$ HPF $\rightarrow$ attenuates low frequency components

Here, we discuss only real and symmetric filters. Following sharpening filters are discussed in this section:

1. Ideal high pass filter
2. Butterworth high pass filter
3. Gaussian high pass filter

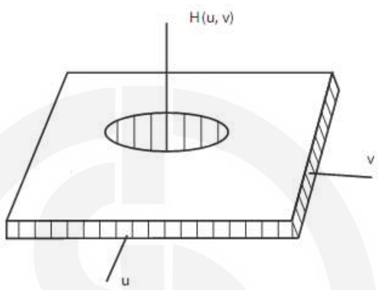
High pass filters are used for enhancing edges. These filters are used to extract edges and noise is enhanced, as a side effect.

6.4.1 Ideal High Pass Filter (IHPF)

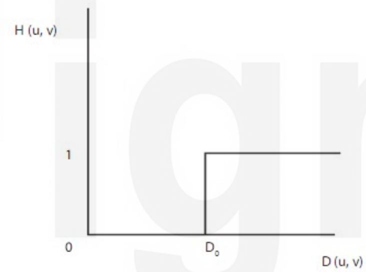
Transfer function of a 2D IHPF is given by

$$H(u,v) = \begin{cases} 1, & \text{if } D(u,v) \geq D_0 \\ 0, & \text{if } D(u,v) < D_0 \end{cases}$$

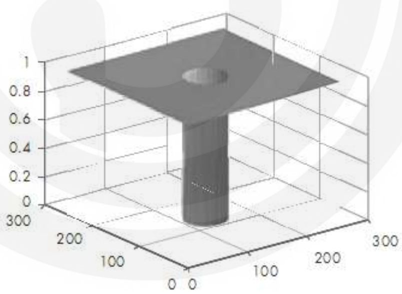
Here, D_0 is the cut off frequency and $D(u,v)$ is the distance from the origin of the Fourier transform. Fig. 14 (a) and Fig. 14 (b) is the IHPF and its transfer function respectively. Fig. 14 (c) is plot of IHPF and Fig. 14 (d) is IHPF as an image. Note that the origin (0,0) is at the centre and not in the corner of the image. The abrupt transition from 1 to 0 of the transfer function $H(u,v)$ cannot be realized in practice. However, the filter can be simulated on a computer. This filter sets to all frequencies inside the circle of radius D and passes all frequencies above D_0 without any attenuation. Ringing is clearly visible in the output (Fig. 15 (b), and Fig. 16(c)) other than sharp edges and boundaries. Output image looks very dark and dull as the high value DC component $G(0,0)$ is eliminated.



(a) Plot of IHPF



(b) Transfer function of IHPF

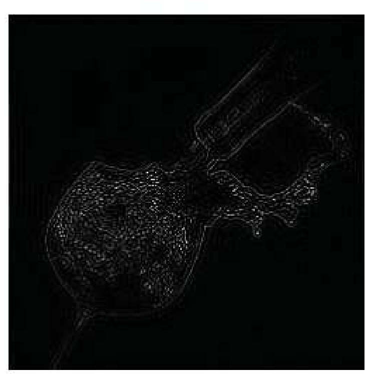


(c) Plot of IHPF

(d) IHPF displayed as an image

Fig. 14

(a) Output of IHPF for D_0= 50



(b) Output of IHPF for D_0 = 60

Fig. 15

6.4.2 Butterworth High Pass Filter (BHPF)

Butterworth filter does not have sharp transition between passband and stop band. The slope depends on the order of the filter. Transfer function of BHPF is

$$H(u,v)=\frac{1}{1+\left[\dfrac{D_0}{D(u,v)}\right]^{2n}},$$

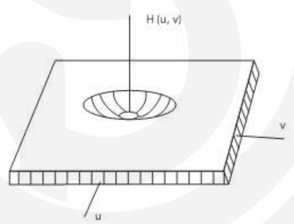
where n is the order of the filter, D_0 is the cut off frequency and $D(u,v)$ is the distance from the origin of Fourier transform.

Fig. 16 (a) and Fig. 16 (b) are BHPF transfer function and Fig. 16 (c) and Fig. 16 (d) are plot and image display of BHPF.

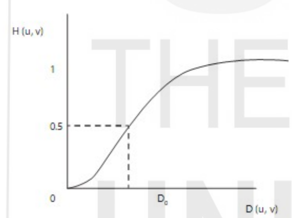
Frequency response does not have a sharp transition as in the ideal HPF.

Thus, less distortion is seen in the output with no ringing effect even for smaller values of cut off frequencies. This filter is more appropriate for image sharpening than ideal HPF as there is no ringing in output.

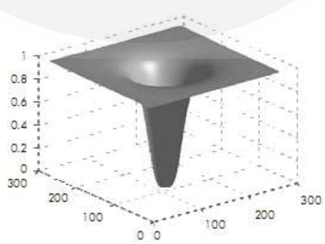
Fig. 16(b) is the plot of GHPF for $D_0=30, n=2,$ and Fig. 16 (c) GHPF displayed as an image. Fig. 17(a) and Fig.17 (b) are the output of GHPF for D_0 = 30 and 130 respectively for $n=2$. It is clear from the output, as D_0 increases, more and more power is removed from the output image. Thus, output looks sharper for higher value of D_0. Fig. 17(d) is the output for $D_0=30\ n=20,$ ringing is clearly visible in the output. As n increases, ringing in butterworth filter increases.



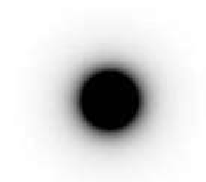
(a) Transfer function of BHPF



(b) Transfer function of BHPF



(c) Plot of BHPF



(d) BHPF displayed as an image

Fig. 16

(a) Output of BHPF with D_0 = 130, n = 2 (b) Output of BHPF with D_0= 30, n = 2

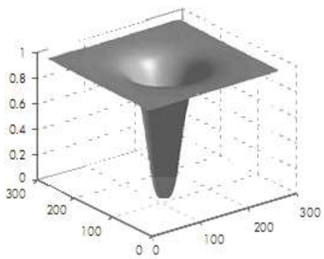
Fig. 17

6.4.3 Gaussian High Pass Filter (GHPF)

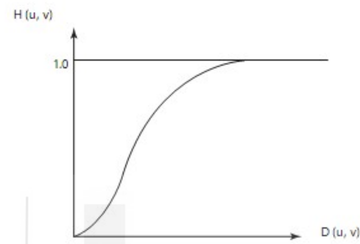
Gaussian high pass filters have smooth transition between passband and stopband near cutoff frequency. The parameter D is a measure of spread of the Gaussian curve. Larger the value D_0, larger is the cut off frequency. Transfer function of GHPF is

$$H(u,v) = 1 - e\frac{-D^2(u,v)}{2D_0^{\,2}},$$

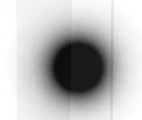
where D_0 is the cut off frequency and $D(u,v)$ is the distance from origin of Fourier transform.



(a) GHPF transfer function



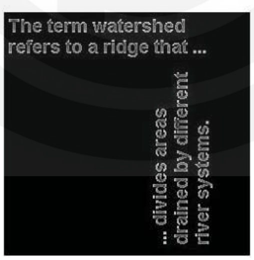
(b) Plot of GHPF



(c) GHPF displayed as an image

Fig. 18

Fig. 18(a) is GHPF Transfer function filter. Plot and image are displayed in Fig. 18 (b) and Fig. 18 (c). Output in Fig. 19 is much smoother than previous two filters with no ringing effect.



(a) Output of GHPF with D_0=30

(b) Output of GHPF with D_0=120

Fig. 19

Let us compare these three high pass filters in frequency domain filters in the following table.

Table 1

	Ideal	**Butterworth**	**Gaussian**
Transfer function	$H(u,v) = \begin{cases} 1, & D(u,v) \le D_0 \\ 0, & D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1+\left[\dfrac{D}{D(u,v)}\right]^{2n}}$	$H(u,v) = 1 - e\dfrac{D^2(u,v)}{2D_0^{\,2}}$
Application	**Edge enhancement**	**Edge enhancement**	**Edge enhancement**

Problems	Ringing	No Ringing	No Ringing

Try the following exercises.

E7) How many types of high pass filters are there in frequency domain? List them.

E8) Give the formula for transform function of a Gaussian high pass filter.

Now, Its time to discuss the concept of image degradation.

6.5 IMAGE DEGRADATION

Overall our objective is to improve image. For that it is important to understand image degradation if we want to remove it. Degradations are of three types

a) Noise
b) Blur
c) Artifacts

Let us define these one by one.

a) **Noise** is a disturbance that causes fluctuations in pixel values. Pixel values show random variations and can cause very disturbing effects on the image. Thus suitable strategies should be designed to model and remove/ reduce noise. Original image is shown in Fig. 20(a) and noisy image with added Gaussian noise is shown in Fig. 20(b).

(a) Original

(b) Noisy Image

Fig. 20

b) **Blur** is a degradation that makes image less clear.This makes image analysis and interpretation very difficult. Motion blur is a very common cause of blurring where blur occurs due to the movement of object or camera. Fig. 21 (a) shows original image and Fig. 21 (b) shows blurred image.

(a) Original Image

(b) Blurred image

Fig. 21

c) **Artifacts** or distortions are extreme intensity or color fluctuations that can make image meaningless. Distortions involve geometric transformations such as translation, rotation or change in scale.

Now, the question arises that what are the sources which contribute to image degradation. Image degradation (as shown in Fig. 22) can happen due to

a) **Sensor Distortions:** Involves quantization, sampling, sensor noise, spectral sensitivity, de-mosaicking, non linearity of sensor etc.

b) **Optical Distortions:** are geometric distortion, blurring due to camera mis-focus.

c) **Atmospheric Distortions:** are haze, turbulence etc.

d) **Other Distortions:** Low illumination, relative motion between object and camera etc.

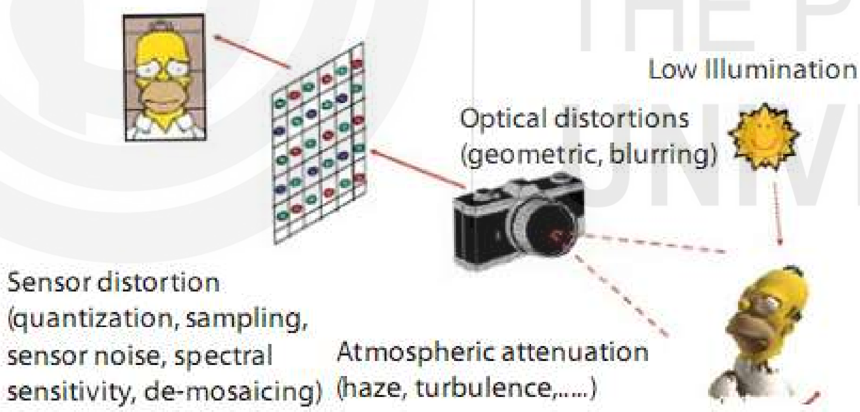


Fig. 22: Typical Degradation Sources

The processes that are used to remove degradation are mainly image enhancement and image restoration.

Restoration is the process of inverting a degradation using knowledge about its nature, whereas enhancement is a process that aims to improve 'bad' quality image so that it looks better. Restoration is distinguished from enhancement, as degradation can be considered as an external influence. Table 2 lists the differences between enhancement and restoration.

Table 2: Enhancement v/s Restoration

	Enhancement	Restoration
1.	It gives better visual representation	It remove effects of sensing environment
2.	No model required	Mathematical model of degradation
3.	It is a subjective process	It is an objective process
4.	Contrast stretching, histogram equalization etc are some enhancement techniques	Inverse filtering, wiener filtering, denoising are some restoration techniques.

Try the following exercises.

E9) What are the factors that can cause image degradation.

In this section we will discuss image degradation/restoration model.

6.6 IMAGE DEGRADATION/RESTORATION MODEL

Consider the block diagram given in Fig. 23 shows the block diagram of degradation/restoration model. Degradation function $h(x, y)$ and noise $n(x, y)$, operate on input image $f(x, y)$ to generate a degraded and noisy image $g(x, y)$.

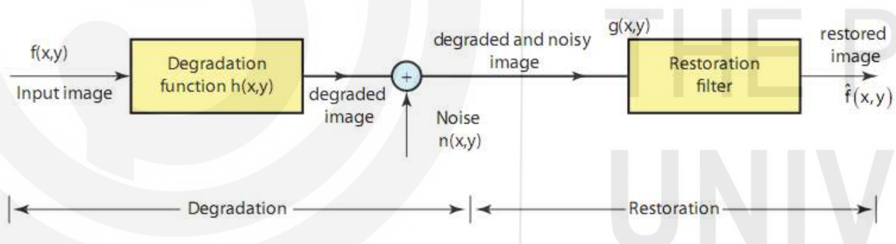


Fig. 23: Block diagram of degradation/restoration model

The notations used in the block diagram are

$f(x, y) =$ original image
$h(x, y) =$ degradation function
$n(x, y) =$ additive noise
$g(x, y) =$ degraded and noisy image
$\hat{f}(x, y) =$ restored image

The objective of restoration process is to estimate $\hat{f}(x, y)$ from the degraded version $g(x, y)$, when some knowledge of degradation function H and noise n is available. The degraded image $g(x, y)$ as shown in Fig. 24 can be expressed mathematically as

$$g(x, y) = h(x, y) * f(x.y) + n(x, y)$$

This equation is in spatial domain and * represents convolution operation. An equivalent frequency domain representation is graphically shown in Fig. 25 and expressed as

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Here $G(u,v) = F[g(x,y)]$

$H(u,v) = F[h(x,y)]$

$F(u,v) = F[g(x,y)]$

$N(u,v) = F[n(x,y)]$

$$F(u,v) = H^4(u,v)[G(u,v) - N(u,v)]$$

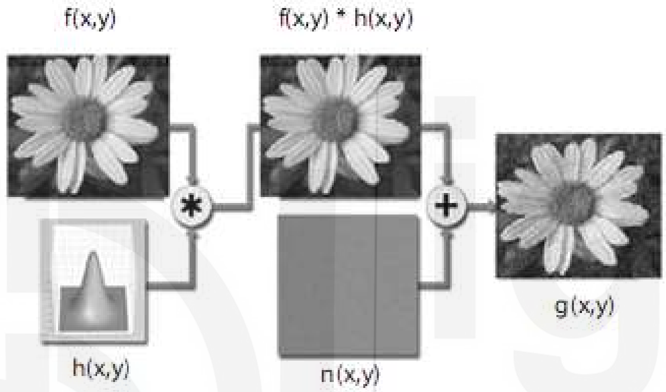


Fig. 24: Image Degradation Model (Spatial Domain)

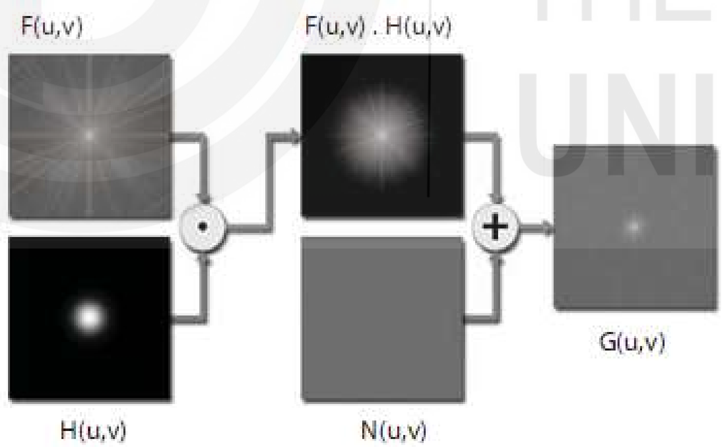


Fig. 25: Image Degradation Model (Frequency Domain)

Restored image can be obtained by the above equation. The problems in implementing this equation are

1) The noise N is unknown. Only the statistical properties of noise can be known.
2) The operation H is singular or ill posed. It is very difficult to estimate H.

Try an exercise.

E10) Explain in detail an image degradation model.

In the following section, we shall discuss noise models in detail.

6.7 NOISE MODELS

Major source of noise in digital images is during image acquisition. Non-ideal image sensors and poor quality of sensing elements contribute to majority of noise. Environmental factors such as light conditions, temperature of atmosphere, humidity, other atmospheric disturbances also account for noise in images. Transmission of image is also a source of noise. Images are corrupted with noise because of interference in the channel, lightning and other disturbances in wireless network. Human interference also plays a part in addition of noise in images.

Properties of Noise

Spatial and frequency characteristics of noise are as follows:

1) Noise is assumed to be 'white noise' (it could contain all possible frequency components), as such, fourier spectrum of noise is constant.

2) Noise is assumed to be independent in spatial domain. Noise is '**uncorrelated**' with the image, that is, there is no correlation between pixel value of image and value of noise components.

The spatial noise descriptor is the statistical behavior of the intensity values in the noise component. Noise intensity is considered as a random variable characterized by a certain probability density function (PDF).

Restoration techniques are oriented towards modeling the degradation (noise in this case) and restore an image to the original state. Most types of noise are modeled as known PDFs. Based on the estimated parameters from the noisy image, a particular noise PDF is chosen. Noise models are divided into two categories:

a) Noise which is independent of spatial location: Gaussian, Rayleight, Gamma, Exponential, Uniform noise are examples of this category.

b) Noise which is dependent on spatial location: Periodic noise is example of this type of noise.

Following are the most commonly occurring noise models:

Gaussian Noise

Gaussian noise model is most frequently used in practice. The PDF of a Gaussian random variable 'z' is given by

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, \qquad (2)$$

where $z =$ intensity/grey level value

$\mu =$ mean (average) value of z

$\sigma =$ standard deviation

Plot of $p(z)$ with respect to z is shown in Fig. 26. 70% of its values are in the range $[(\mu-\sigma),(\mu-\sigma)]$ while 95% of the values are in the range $[(\mu-2\sigma),(\mu+2\sigma)]$. DFT of **Gaussian (normal) noise is another Gaussian process**. This property of Gaussian noise makes it most often used noise model. Some examples where Gaussian model is the most appropriate model are electronic circuit noise, sensor noise due to low illumination or high temperature, poor illumination.

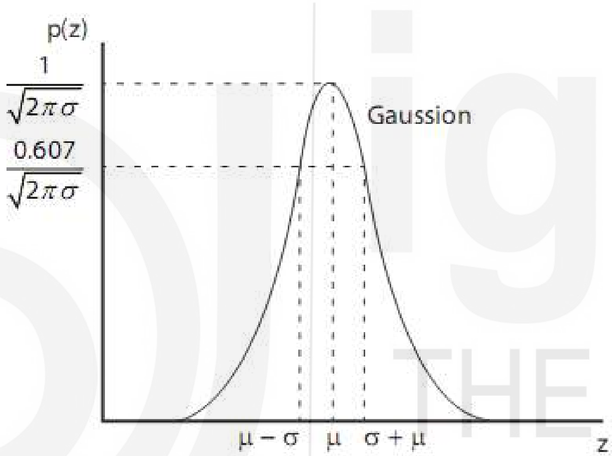


Fig. 26: PDF of Gaussian Noise Model

Gaussian noise is useful for modeling natural processes which introduce noise (e.g. noise caused by the discrete nature of radiation and the conversion of the optical signal into an electrical one – detector/shot noise, the electrical noise during acquisition – sensor electrical signal amplification, etc.).

Rayleigh Noise

Radar range and velocity images typically contain noise that can be modeled by the Rayleigh distribution. Rayleigh distribution is defined by

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} & z \geq a \\ 0 & z < a \end{cases}, \qquad (3)$$

Mean density is given as $\mu = a + \sqrt{\pi^{b/4}}$ and variance is given by $\sigma^2 = \frac{b(4-\pi)}{4}$.

Plot of PDF is shown in Fig 27. As it is clear that the curve doesn't start from origin and is not symmetrical with respect to the centre of the curve. Thus, Rayleigh density is useful for approximating skewed (non-uniform) histograms. This is mainly used in range imaging.

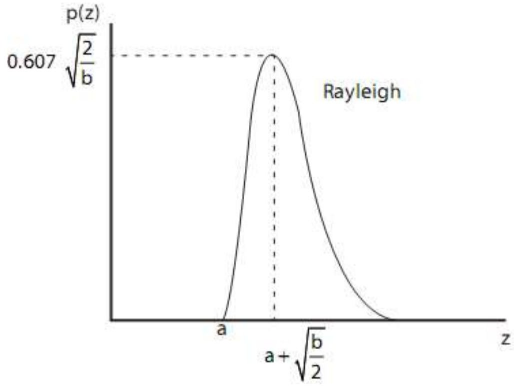


Fig. 27: PDF of Rayleigh Noise

Erlang (Gamma) Noise

Erlang noise is given by

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-nz} & z \geq 0 \\ 0 & z < 0 \end{cases}, \quad (4)$$

Where a and b are positive integers, mean density is given by $\mu = \dfrac{b}{a}$

and variance is $\sigma^2 = \dfrac{b}{a^2}$.

When the denominator is a gamma function, the PDF describes the gamma distribution. Plot is shown in Fig. 28.

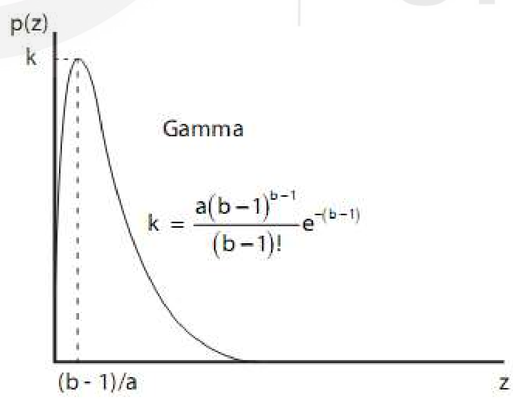


Fig. 28: PDF of Erlang Noise

Uniform Noise

Uniform noise is specified as

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if} \quad a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$

Then mean and variance of uniform noise is given by

$$\mu = \frac{a+b}{2}, \ \sigma^2 = \frac{(b-a)^2}{12}$$

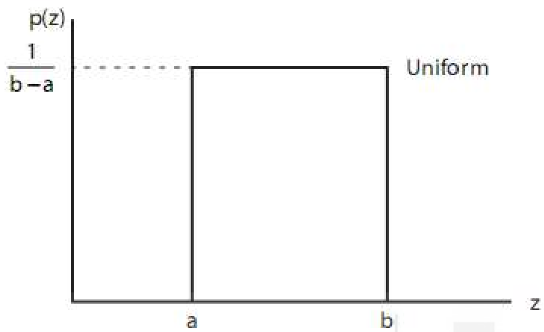


Fig. 29: PDF of Uniform Noise.

Fig. 29 shows the plot of PDF of uniform noise. Uniform noise is least used in practice.

Impulse (Salt and Pepper) Noise

Impulse (salt and pepper) noise is specified as

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \end{cases}, \tag{6}$$

Fig. 30 shows the plot of PDF of impulse noise. If $b > a$, intensity (grey level) ‘b’ will appear as a light dot on the image and ‘a’ appears as a dark dot. This is a **‘bipolar’** noise, If $P_a = 0$ or $P_b = 0 \Rightarrow$ **unipolar noise** Generally, a and b values are saturated (very high or very low value), resulting in positive impulses being white (salt) and negative impulses being black (pepper). If $P_a = 0$ and P_b exists, this is called ‘**pepper noise**’ as only black dots are visible as noise. If $P_b = 0$, only P_a exists, this is called ‘**salt noise**’ as only white dots are visible on the image as noise.

Impulse noise occurs when quick transitions happen, such as faulty switching takes place. Noise parameters are generally estimated based on histogram of small flat area of noisy image.

The salt & pepper noise is generally caused by malfunctioning of camera’s sensor cells, by memory cell failure or by synchronization errors in the image digitizing or transmission.

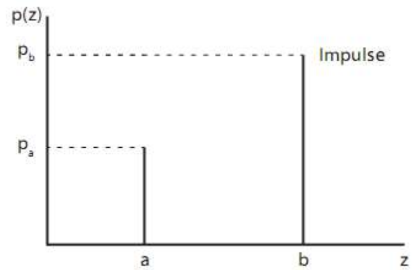


Fig. 30 PDF of Uniform noise.

Fig. 31 shown an example of impulse noise with $p = 0.1$ added to input image and togenerate a noisy image g. Noise level $p = 0.1$ means that approximately 10% of pixelsare contaminated by salt or pepper noise (highlighted by box)

128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128

f(x,y)

⇒

128	128	$\boxed{255}$	$\boxed{0}$	128	128	128	128	128	128
128	128	128	128	$\boxed{0}$	128	128	128	128	0
128	128	128	128	128	128	128	128	128	128
128	128	$\boxed{0}$	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
$\boxed{0}$	128	128	128	128	$\boxed{255}$	128	128	128	128
128	128	128	128	128	128	128	128	128	$\boxed{255}$
128	128	128	128	128	128	128	$\boxed{255}$	128	128

g(x,y)

Fig. 31: Numerical Example of Adding Impulse Noise with P = 0.1.

Fig. 32 shows the flower image with different types of noise. It is very easy to identify the effect of different types of noise on the images. Fig. 32 (a) shows original image, Fig. 32 (b) shows image with Gaussian noise. Fig. 32 (c) shows image with salt and pepper noise and Fig. 32 (d) shows image with uniform noise. The amount of noise added can also vary. If the amount of noise added is more, it becomes very difficult to remove it.

(a) Original image

(b) Image with Gaussian noise

(c) Image with salt & pepper noise

(d) Image with uniform noise

Fig. 32

Let us discuss an important type of noise.

Periodic noise is a spatially dependent noise. During Image acquisition, electrical or electromechanical interference may cause such type of periodic noise. A strong periodic noise can be seen in the frequency domain as equi-spaced dots at a particular radius around the centre (origin) of the spectrum. Fig. 33 shows image with periodic noise.

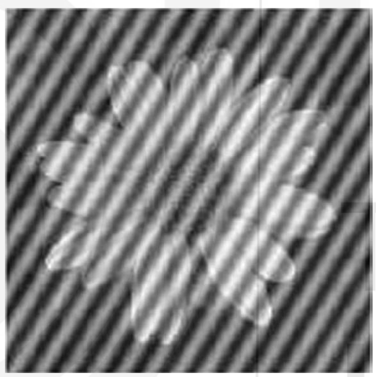


Fig. 33: Image with periodic noise

Now, you may like to try the following exercises.

E11) What is noise? How noise can be eradicated?

E12) Explain the types of noises based on its probability distribution.

In the following section, we will discuss restoration in the presence of noise only-spatial filtering.

6.8 RESTORATION IN THE PRESENCE OF NOISE ONLY – SPATIAL FILTERING

If H is an identity operator and degradation is only due to additive noise,

$$g(x, y) = f(x, y) + n(x, y)$$
$$G(u, v) = F(u, v) + N(u, v)$$

As noise is unknown, $f(x, y) = g(x, y) - n(x, y)$ is not realistic. Thus, spatial filtering is used when additive random noise is present. Mean and median filters are used for noise removal. Band reject and band pass filters are used for periodic noise removal.

6.8.1 Mean Filters

Spatial Smoothing concepts are explained in earlier unit 4 of this course. Now, Consider Fig. 34, $S_{x,y} = \text{Sub}$ image window of size $m \times n$ centred at (x, y). Fig. 35 shows 3×3 and 5×5 sub images. Mean filter computes average value of the corrupted image $g(x, y)$ in the area defined by $S_{x,y}$

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{s,t \in S_{xy}} g(s, t)$$

Such filter smooths local variations in an image, thus reducing noise and introducing blurring. This filter is well suited for random noise like Gaussian, uniform noise.

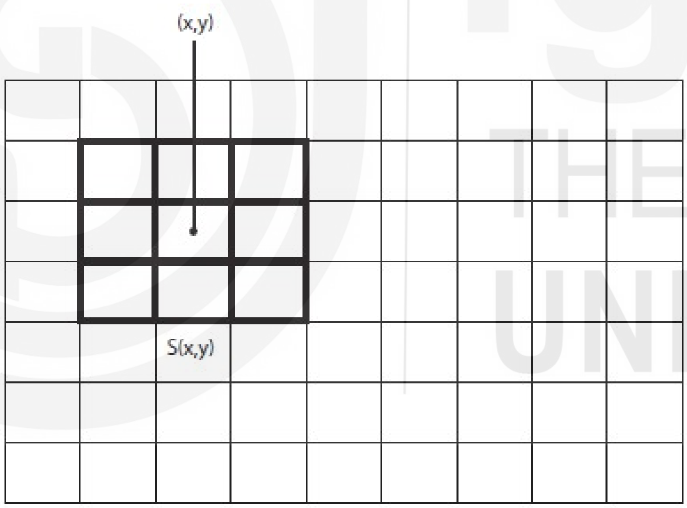
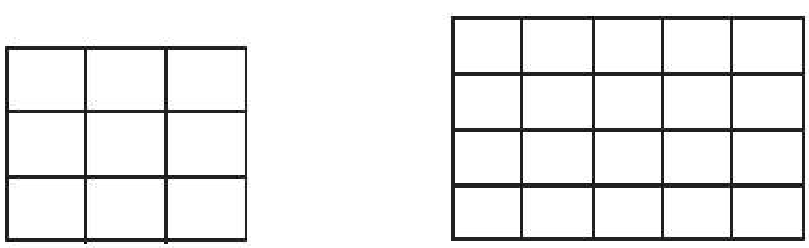


Fig. 34: Graphic Illustration of Sub-Image in An Image



(a) $S(x, y)$ of size 3×3 (b) $S(x, y)$ of size 5×5

Fig. 35: Sub-Image of Various Sizes

Thus, new value at (x, y) in image in Fig. 36 is

$\{g(s,t)\} = 19 = [30+10+20+10+250+25+20+25+3] = 46.7 \approx 47$

30	10	20
10	250	25
20	25	30

→

×	×	×
×	$46.7 \approx 47$	×
×	×	×

Fig. 36: Example of Mean Filtering.

Let us apply this in the following example.

Example 1: Show effect of 3×3 mean filter on a simple image in Fig. 37 (a) and Fig. 38 (a)

Solution: As explained in Unit 4, a 3 × 3 mean filter is overlapped with image and output for that particular pixel is derived and then filter centre is moved to the next pixel. We generate a lower size image because the filter mask doesn't overlap fully on first and last row and column.

0	0	0	0	0
0	0	0	1	1
0	0	1	1	1
0	0	1	20	1
0	0	1	1	1

(a)

Mean filter →

	$1/9$	$3/9$	$5/9$	
	$2/9$	$24/9$	$27/9$	
	$3/9$	$25/9$	$28/9$	

(b)

Fig.37: Input and Output Image.

Mean filter removes random noise by introducing blurring. Random noise value of 20 in Fig. 37 (a) is removed from the resultant image (b). But, importantly the edge is also diluted and blurred. In the second image Fig. 38 (a), which has fairly constant values, the pixel values remain more or less same in Fig. 38 (b).

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

(a)

Mean filter →

	4	5	6	
	4	6.5	8	
	5	6	8	

(b)

Fig.38 Input and Output Image.

Gaussian noise is added to the input image Fig. 39 (a). $3\times3, 5\times5$ and 7×7 mean filters are applied to the noisy image and the output images are displayed in Fig. 39 (b), (c), (d). As it is clear from the output, 3×3 filter (Fig. 39 (b)) does not remove the noise completely. Noise is still seen in the image but blurring is less. In 5×5 (Fig. 39 (c)) filtering more noise is removed but image gets blurred. In 7×7 (Fig. 39(d)), too much blurring is seen in the output.

(a) Original Image

(b) Filtered Image by Mean Filter 3×3

(c) Filtered Image by Mean Filter 5×5

(d) Filtered Image by Mean Filter 7×7.

Fig. 39

Let us discuss median filter.

6.8.2 Median Filters

Median filter replaces the pixel value by the median of the pixel values in the neighbourhood of the centre pixel (x, y). The filtered image is given by

$$\hat{f}(x, y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s, t)\}$$

Fig. 40 shows the procedure of applying 3×3 median filter on an image. As impulse noise appears as black (minimum) or white (maximum) dots, taking median effectively suppresses the noise.

Thus, median filter provides excellent results for salt and pepper noise with considerably less blurring than linear smoothing filter of the same size. These filters are very effective for both bipolar and unipolar noise. But, for higher noise strength, it affects clean pixels as well and a noticeable edge blurring exists after median filtering.

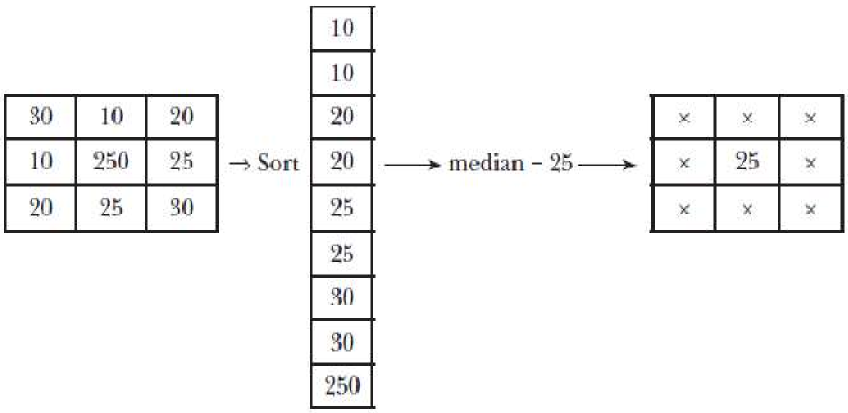


Fig. 40: Example of Median Filtering.

To understand this clearly, see the following example.

Example 2: Show the effect of 3×3 median filter on a simple image given in Fig. 41(a) and Fig. 41(b).

128	128	128	128	128
128	0	128	128	128
128	128	128	128	128
128	128	128	128	128
128	128	128	128	128

Median filter →

	128	128	128	
	128	128	128	
	128	128	128	

(a) Input image **(b) Output image**

Fig. 41

Solution: When a 3×3 median filter is implemented, all 9 pixels around 'hotspot' are arranged in ascending/ descending order. Center pixel is taken as output and center pixel is replaced by the output. This process is repeated for the entire image.

128	128	128	0	128
128	0	128	128	128
0	0	255	255	255
0	0	128	255	0
128	0	0	0	128

Median filter →

	128	128	255	
	0	128	255	
	0	0	128	

(a) Input image **(b) Output image**

Fig. 42

***

It is clear from Example 2, Fig. 41 (a) and Fig. 41 (b) that if noise strength is low in noisy image, output is completely clean. But if noise strength is more (more number of noisy pixels in the image), output is not completely noise free as can be seen in Fig. 42 (a) and Fig. 42 (b).

Let us see the effect of the median filter.

Salt and pepper noise is added to an image given in Fig. 43 (a), noisy image is shown in Fig. 43 (b). 3×3 mean filter and 5 × 5 median filter is applied on it. As it is clear from the result, (Fig. 43 (c) and Fig. 43 (d)) mean filter is not effective in removing salt and pepper noise. But median filter completely removes salt and pepper noise without introducing any blur.

(a) Original Image

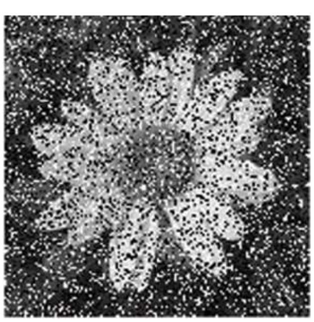
(b) Noisy Image

(c) Filtered Image with Mean Filter

(d) Filtered Image with Media Filter

Fig. 43

Salt and pepper noise with density of 0.3 is added to an image. The noisy image (Fig. 44 (a)) is filtered using $3\times3, 5\times5$ and 7×7, median filter. The results in Fig. 44 (b), (c), (d) show that 3×3 median filter is unable to remove the noise completely as the noise density is high. But 5×5 and 7×7 median filters remove noise completely but some distortions are seen specially in Fig. 44 (d).



(a) Noisy Image

(b) Filtered Image with 3×3 Median Filter

(c) Filtered Image with 5×5 **Median Filter**

(d) Filtered Image with 7×7 **Median Filter**

Fig. 44

Now, in the following section, we shall discuss noise reduction.

6.9 PERIODIC NOISE REDUCTION

Periodic noise is spatially dependent noise and it occurs due to electrical or electromagnetic interference. It gives rise to a regular noise pattern in an image. Frequency domain (fourier domain) techniques are very effective in removing periodic noise. Basic steps in frequency domain filtering remain same as discussed above. Here, we are discussing two frequency domain filters; namely band reject filter and band pass filter.

6.9.1 Band Reject Filter

Removing periodic noise from an image involves removing a particular range of frequencies from the image. Transfer function of ideal band reject filter is

$$H(u,v) = \begin{cases} 1 & D(u,v) < D_0 - \dfrac{W}{2} \\ 0 & D_0 - \dfrac{W}{2} \le D(u,v) \le D_0 + \dfrac{W}{2}, \\ 1 & D(u,v) > D_o + \dfrac{W}{2} \end{cases} \tag{7}$$

where W is the width of the band (band width), D_0 is its radial centre and $D(u,v)$ is the distance from the origin and is given by

$$D(u,v) = \left[\left(u - \frac{M}{2}\right)^{1/2} + \left(v - \frac{N}{2}\right)^{1/2}\right]$$

$D(u,v)$ is the distance measured from the point (u,v) to the centre $\left(\frac{M}{2}, \frac{N}{2}\right)$. If size of an image is $M \times N,$ then the centre is at $\left(\frac{M}{2}, \frac{N}{2}\right)$.

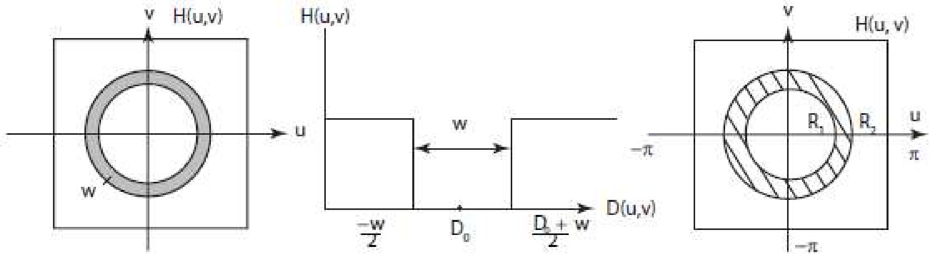


Fig 45: Frequency response of ideal band reject filter

Transfer function of butter worth band reject filter of order 'n' is given by

$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)W}{D^2(u,v) - D_0^2}\right]^{2n}} \tag{8}$$

Gaussian band reject filter is given by

$$H(u,v) = 1 - e^{\frac{1}{2}\left[\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right]} \tag{9}$$

Fig. 46 gives the plots of ideal, butterworth and gaussian band reject filters.

(a) Ideal **(b) Butterworth (order 1)** **(c) Gaussian**

Fig. 46: Plots of Band Reject Filters

6.9.2 Band Pass Filter

Band pass filter performs just opposite to band reject filter. The transfer function of band pass filter can be obtained from band pass filters.

$$H_{bp}(u,v) = 1 - H_{br}(u,v), \tag{10}$$

Where, H_{bp} is transfer function of band pass filter and H_{br} is transfer function of band reject filter.

Ideal band pass filter is given by

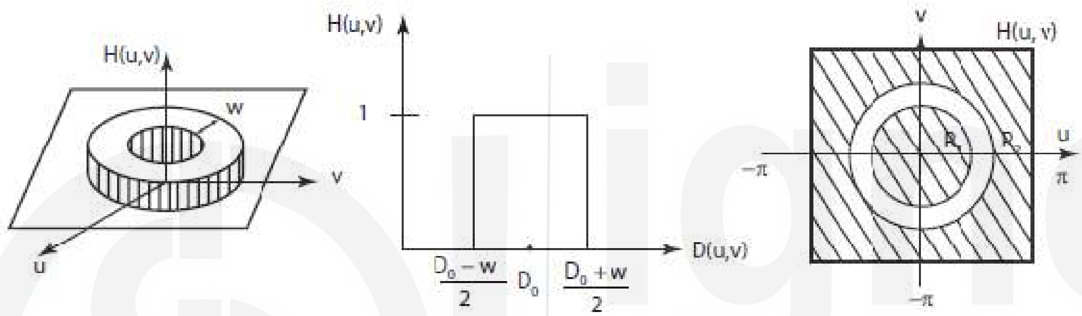
$$H(u,v) = \begin{cases} 0 & D(u,v) \le D_0 - \frac{W}{2} \\ 1 & D_0 - \frac{W}{2} < D(u,v) < D_0 + \frac{W}{2}, \\ 0 & D(u,v) \ge D_0 + \frac{W}{2} \end{cases} \tag{11}$$

Where $D(u,v)$ is the distance from origin, W is the band width D_0 is the radial centre or the cut off frequency.

Fig. 47 shows the transfer function of ideal band pass filter.
Butterworth band pass filter of order 'n' is given by

$$H(u,v) = 1 - H(u,v)_{\text{butterworth band reject}} \tag{12}$$

$$H(u,v) = 1 - \frac{1}{1+\left[\frac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}} = \frac{\left[\frac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}}{1+\left[\frac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}} \tag{13}$$



(a) (b) (c)

Fig. 47: Frequency Response of Ideal Band Pass Filter

Similarly, Gaussian band pass filter is given by

$$H(u,v) = 1 - H(u,v)_{\text{gaussian band pass}}$$

$$= 1 - \left[1 - e^{-1/2\left[\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right]^2}\right]$$

$$= e^{-1/2\left[\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right]^2}$$

6.10 ESTIMATION OF DEGRADATION FUNCTION

In order to restore the image, we need to estimate the degradation function. There are three principal ways to estimate the degradation function to be used in restoration:

1) Observation
2) Experimentation
3) Mathematical modelling

Once the degradation function has been estimated, then, restoration is a de convolution process also called **Blind Deconvolution.**

6.10.1 Observation

In restoration using observation, we assume that an image $g(x, y)$ is degraded with an unknown degradation function H. We try to estimate H from the information gathered from the image itself. For example, in case of burred image, a small rectangular section of image containing a part of object and background is taken (Fig. 48). To reduce the effect of noise, the chosen part should be such that it shows presence of a strong signal. We try to un-blur that sub-image manually as much as possible and generate $f_s(x, y)$ from $g_s(x, y)$.

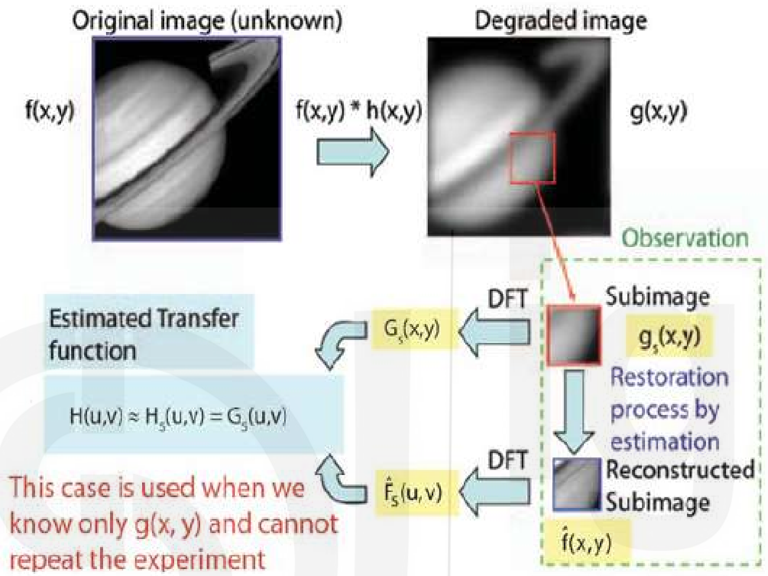


Fig. 48: Estimation by image observation

Here, $g(x, y)$ is the original sub image

$\hat{f}_s(x, y)$ is the restored version of $g_s(x, y)$

Thus, degradation can be estimated for the sub image by

$$H_s(u, v) = \frac{g_s(u, v)}{\hat{F}_s(u, v)}$$

From the characteristics of $H_s(u, v)$, we try to deduce the complete degradation function $H_s(u, v)$ based on the assumption of position invariance. For example, if $H_s(u, v)$ has a Gaussian shape, we can construct $H(u, v)$ on a larger scale with the same (Gaussian)shape. This is a very involved process and is used in very specific situations.

6.10.2 Experimentation

It is possible to estimate the degradation function accurately if the equipment used to acquire the degraded image is available. The processes is shown in Fig. 49.

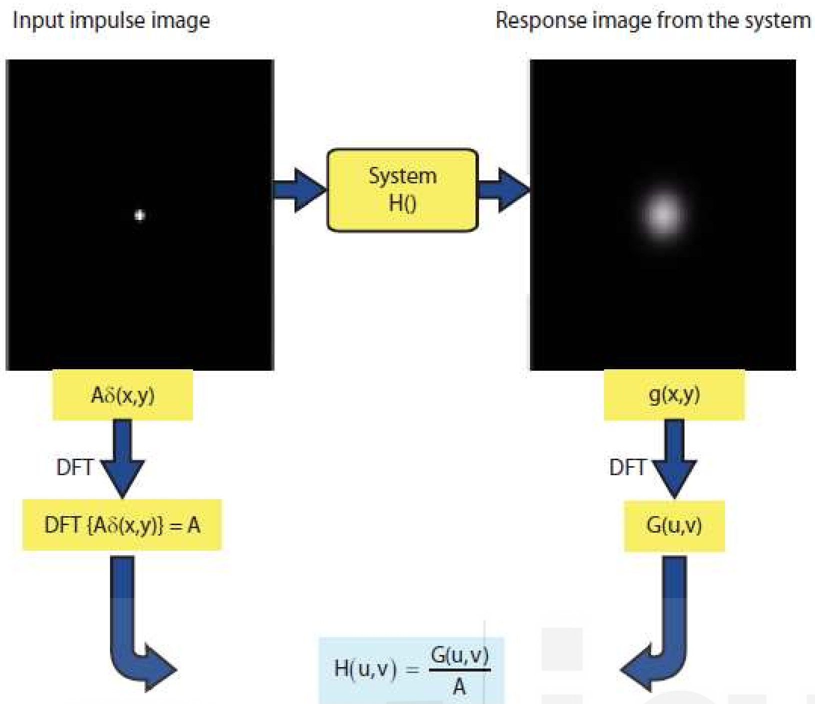


Fig. 49: Estimation by experimentation

We now list the steps performed for estimation.

Step 1: First step is to adjust the equipment by varying the system setting such that the image obtained is similar to the degraded image that needs to be restored.

Step 2: Second step is to obtain the impulse response of the degradation by imaging an impulse using the same system setting, since a linear space – invariant system is completely characterized by its impulse response. An impulse is simulated by a maximally bright dot of light. As shown in the Fig. 2, impulse response is given by

$$H(u,v) = \frac{G(u,v)}{A},$$

where $G(u,v) = DFT[g(x,y)] = DFT$[degraded impulse], and A is the constant describing the strength of the impulse.

6.10.3 Modelling

Modelling is used to estimate the degradation function. Scientists have studied several environmental conditions and other processes which cause degradation, and have formulated several fundamental models for the degradation functions. Degradation model based on atmospheric turbulence blur is given as

$$H(u,v) = e^{-k(u^2+v^2)5/6}$$
$$h(x,y) = e^{-k(x^2+y^2)5/6},$$

where k is a constant that depend on the nature of blur. Various values used for the constant k along with their type of turbulence are given as

$k = 0.0025$ for server turbulence
$k = 0.001$ for wild turbulence
$k = 0.00025$ for low turbulence

This is commonly used in remote sensing and axial imaging applications. Degradation model for uniform out of focus blur (optical blur).

$$h(x,y) = \begin{cases} \frac{1}{L^2}; & \frac{-L}{2} \le x^0, y^0 \le \frac{L}{2} \\ 0 \;; & \text{otherwise} \end{cases} \tag{1}$$

$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin(ua+vb) e^{-j\pi(ua+vb)} \tag{2}$$

Blur can be due to motion (camera and object moving with respect to each other). With suitable values of T, a and b, blurred image can be generated using this transfer function. Fig. 50(a) shows original image and Fig. 50(b) shows blurred image.

(a) Original Image **(b) Blurred Image**

Fig. 50

Try an exercise.

E13) What are the different methods of estimation of image degradation function?

In the following section, we shall discuss inverse filtering.

6.11 INVERSE FILTERING

Inverse filter is also known as reconstruction filter. Deblurring is very important in restoration applications because blurring is visually annoying. It is bad for analysis, and de-blurred images have plenty of applications. Applications include astronomical imaging, law enforcement (identifying criminals), biometrics etc. In this unit we will discuss Inverse filtering, pseudo-inverse filtering and Wiener filtering for deblurring.

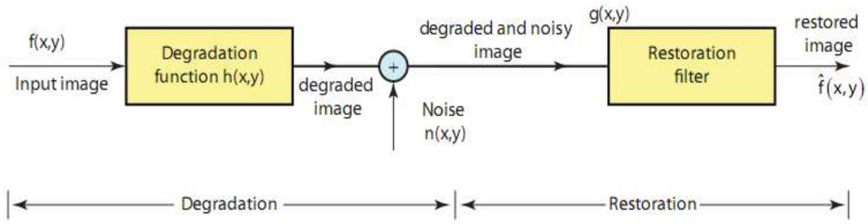


Fig. 51: Block diagram of degradation/restoration model

As in the absence of noise, degradation model becomes,

$$G(u,v) = F(u,v)H(u,v) \tag{3}$$

Simplest approach to restoration is direct inverse filtering, where we can compute an estimate $\hat{F}(u,v)$, of the transform of the original image simply by dividing transform of the degraded image by the degradation function.

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \tag{4}$$

In presence of noise, degradation model as shown in Fig. 51 becomes

$$G(u,v) = F(u,v)H(u,v) + N(u,v) \tag{5}$$

After applying inverse filtering

$$\hat{F}(u,v) = H_R(u,v)G(u,v) \tag{6}$$

Substituting the values of $H_R(u,v)$ and $G(u,v)$, we get

$$\begin{aligned}\hat{F}(u,v) &= \frac{1}{H(u,v)}\left[F(u,v)H(u,v) + N(u,v)\right] \\ &= \frac{F(u,v)H(u,v)}{H(u,v)} + \frac{N(u,v)}{H(u,v)} \\ &= F(u,v) + \frac{N(u,v)}{H(u,v)}\end{aligned} \tag{7}$$

Thus, in case of noisy degraded images, output is also noisy.

If $H(u,v) \approx 0 \dfrac{N(u,v)}{H(u,v)} \rightarrow \infty$

noise is amplified and it dominates the output.

Limitations of inverse filtering are:

1) It is an unstable filter
2) It is sensitive to noise. In practice, inverse filter is not popularly used.

To remove the limitations of inverse filter, pseudo inverse filters are used. Pseduo Inverse filter is defined as,

$$H_R(u,v) = \begin{cases} \dfrac{1}{H(u,v)} & |H(u,v)| \geq \in \\ 0 & |H(u,v)| < \in \end{cases}$$

where, $\in$ is a small value.

Pseudo-inverse filters eliminates the first problem of inverse filters (un-stability).

As $H(u,v) \rightarrow 0, H_R(u,v) = 0.$

Hence, it does not allow $H_R(u,v) \rightarrow \infty.$

It is a stable filter. This filter also has a problem of noise amplification. Inverse filtering is applied to the image blurred with a Gaussian (Fig. 52(b)). The output image (Fig. 52(c)) is very close to the original image (a). Then same inverse filtering is used when the blurred image is subjected to additive noise with different strengths. Outputs are shown in Fig. 52(d), (e) and (f). As noise increases, the output of filter goes down.

(a) Original Image

(b) Image Blurred with a Gaussian

(c) Inverse Filter Applied to Noiseless Blurred

(d) Inverse Filter Applied to Blurred Image Plus


(e) Inverse Filter Applied to Blurred Image Plus Noise (0.1)


(f) Inverse Filter Applied to Blurred Plus Noise (0.5)

Fig. 52

Try an exercise.

E14) Explain in brief the inverse filtering approach and its limitations in image restoration.

In the following section, we shall discuss wiener filter.

6.12 WIENER FILTER

Wiener filter is also known as minimum mean square error. This approach includes both the degradation function and power spectrum of noise characteristics in developing the restoration filter. Wiener filter restores the image in the presence of blur as well as noise.

This method is founded by considering image and noise as random variables and objective is to find as estimate $\hat{f}$ of the uncorrupted image f such that the mean square error between them is minimized. This error is given by

$$e^2 = E\{(f - \hat{f})^2\}, \tag{9}$$

where, $E\{.\}$ is the expected value of the argument. Noise and image are assumed to be uncorrelated. Filter transfer function is given by

$$H_R(u,v) = \frac{S_{fg}(u,v)}{S_{gg}(u,v)}, \tag{10}$$

where, $S_{fg}(u,v)$ is the power spectral density of recovered image and noisy image and $S_{gg}(u,v)$ is the power spectral density of noisy image

$$H_R(u,v) = \frac{S_{fg}(u,v)}{S_{gg}(u,v)} = \frac{H^*(u,v)S_{ff}(u,v)}{|H(u,v)|^2 S_{ff}(u,v) + S_{nn}(u,v)}, \tag{11}$$

where, $H(u,v) =$ degradation function.

$$|H(u,v)|^2 = H(u,v)H^*(u,v)$$

$S_{nn}(u,v) =$ power spectral density of noise

$S_{ff}(u,v) =$ power spectral density of undergraded image.

If $S_{ff}(u,v), S_{nn}(u,v)$ and $H(u,v)$ is known $H_R(u,v)$ is completely known. Wiener filter works very well for specific applications and is not suitable for general images. For example, if a wiener filter $H_R(u,v)$ is working well for faces, same filter would not work for landscapes etc. Now we discuss several cases to test wiener filter.

Case 1: When, there is no noise $S_{nn}(u,v) = 0$

$$H_R(u,v) = \frac{H^*(u,v)S_{ff}(u,v)}{|H(u,v)|^2 S_{ff}(u,v) + 0}$$

$$H_R(u,v) = \frac{1}{H(u,v)} = \text{inverse filter}$$

Thus, if there is no noise, Wiener filter = Inverse filter.

Case 2: Noise is present, but there is no degradation, $H(u,v) = 1$

$$\therefore H_R(u,v) = \frac{1.S_{ff}(u,v)}{1.S_{ff}(u,v) + S_{nn}(u,v)}$$

$$= \frac{S_{ff}(u,v)/S_{nn}(u,v)}{\dfrac{S_{ff}(u,v)}{S_{nn}(u,v)} + 1}$$

$$= \frac{SNR}{SNR + 1}$$

$$SNR = \text{Signal to noise ratio} = \frac{S_{ff}(u,v)}{S_{nn}(u,v)} = \frac{\text{PSD of signal}}{\text{PSD of noise}}$$

$$\Rightarrow SNR >> 1 \;\; H_R(u,v) \approx \frac{SNR}{SNR} = 1$$

a) If signal to noise ratio is high

Thus, if SNR high, wiener filter acts like pass band and allows all the signal to pass through without any attenuation.

b) If $SNR << 1$, if Signal to noise ratio is low, then

$$H_R(u,v) = \frac{SNR}{1} = SNR$$

$$= \text{a very low value}$$

$$\approx 0$$

Thus, if SNR is low and noise level very high, $H_R(u,v) \approx 0$, acts as a stop band for signal and doesn't allow signal to pass, thus attenuating noise. If noise is high in the signal, wiener filter reduces it after filtering.

$$SNR = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|^2}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|N(u,v)|^2}$$

SNR gives a measure of the level of information bearing signal power (i.e. of the original, undegraded image) to the level of noise power. Images with low noise tend to have high SNR and conversely, the same image with higher noise level has a high SNR.

The mean square error is given by

$$MSE = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y) - \hat{f}(x,y)]^2$$

Here, $f(x,y)$ is the original image and $\hat{f}(x,y)$ is the restored image.

Wiener filter is also called minimum mean square error (MMSE) or Least square (LS) filtering because it minimizes the error between the image and its estimate.

$$H_R(u,v) = \frac{H^*(u,v)S_{ff}(u,v)}{|H(u,v)|^2 S_{ff}(u,v) + S_{nn}(u,v)}$$

$$= \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_{nn}(u,v)}{S_{ff}(u,v)}} \longrightarrow \text{difficult to estimate}$$

Wiener filter in this form is not very useful, as it is difficult of estimate noise power spectrum S_{nn} and undergraded image power spectrum S_{ff},

$S_{nn}(u,v) \rightarrow$ difficult to estimate

$S_{ff}(u,v) \rightarrow$ difficult to estimate

To solve this, we can do an approximation

$\frac{S_{nn}(u,v)}{S_{ff}(u,v)} \approx k$ (approximated by constant k)

$$H_R(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + k}$$

And estimated restored image

$$\hat{F}(u,v) = H_R(u,v)G(u,v)$$

$$= \left[\frac{H^*(u,v)}{H(u,v)|^2 + k}\right]G(u,v)$$

$$= \left[\frac{1}{H^*(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + k}\right]G(u,v)$$

k is chosen experimentally and iteratively for best results. In Fig. 53, small noise is added to a blurred image, which is restored by wiener filter in Fig. 53(b).If the amount of added noise is increased Fig. 53(c), the restored image by wiener filter (Fig. 53(d)) is not good. Thus, it is apparent that the wiener filter only works well when the noise is small.

(a) Blurred Image with Small Additive noise

(b) Image Restored by by Wiener Filter

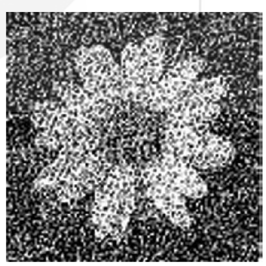
(c) Blurred Image with Increase Additive Noise

(d) Image Restored by Wiener Filter

Fig. 53

(a) Original Image

(b) Blurred Image



(c) Restored image

Fig.54: Applying Wiener Filter

Image is blurred using linear motion = 15, angle = 5 shown in Fig. 54(b). Wiener filter is used to deconvolve the blurred image. The output (Fig. 54(c)) is not clear as the wiener filter does not use any prediction about noise density.

Now, try an exercise.

E15) Discuss the minimum mean square error (Wiener) filtering.

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

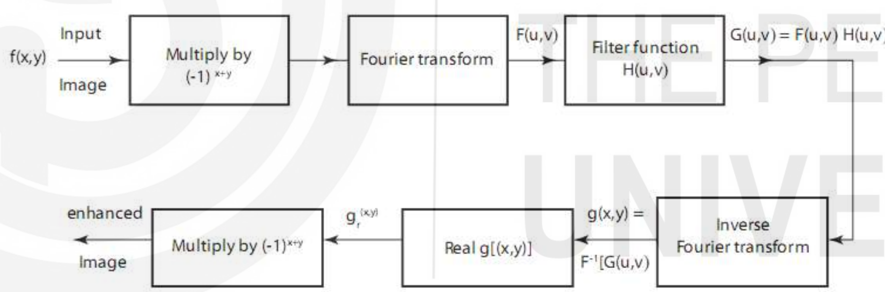
6.13 SUMMARY

In this unit, we have discussed the following points.

1. Image characteristics in frequency domain
2. Filtering in frequency domain
3. Basic steps of frequency domain filtering
4. Various low pass and high pass filters
5. Various image smoothing filters in frequency domain
6. Various image sharpening filters in frequency domain
7. Sources of degradation.
8. Difference between enhancement and restoration.
9. Image degradation/restoration model.
10. Various types of noises with their pdfs.
11. Mean and median filters for noise reduction
12. Band reject and band pass filters for periodic noise reduction.
13. Methods of estimation of degradation function.
14. Inverse filtering.
15. Wiener filtering.

6.14 SOLUTIONS AND ANSWERS

E1)



$u = \frac{M}{2}$ and $v = \frac{N}{2}$.

1. Multiply input image $f(x, y)$ by $(-1)^{x+y}$ to centre the transform to
2. Compute $F(u, v)$, Fourier transform of the output of step 1.
3. Multiply filter function $H(u, v)$ to $F(u, v)$ to get $G(u, v)$.
4. Take inverse Fourier transform of $G(u, v)$ to get $g(x, y)$.
5. Take the real part of $g(x, y)$ to get $g_r(x, y)$
6. Multiply the result of step 5 by $(-1)^{x+y}$ to shift the centre back to origin and enhanced image is generated.

E2) Image enhancement can be done very effectively in frequency domain. High frequency noise, undesirable breakages in the edges and other imperfections can be taken care by filtering in frequency domain. Low pass and high pass filters are implemented with ease and perfection in frequency domain.

E3) $$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}},$$

where $D_0 =$ Cut off frequency or distance from the centre $n =$ filter order $\left(\frac{M}{2}, \frac{N}{2}\right)$

E4)

	Ideal	**Butterworth**	**Gaussian**
Transfer function	$H(u,v) = \begin{cases} 1, & D(u,v) \le D_0 \\ 0, & D(u,v) > D_0 \end{cases}$	$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{24}}$	$H(u,v) = e^{-D^2(u,v)/2D^2}$
Application	**Reduce noise**	**Reduce noise**	**Reduce noise**
Problems	**Blurring Ringing**	**Blurring, Ringing for higher order filters**	**Blurring no ringing**

E5) Smoothing filters are low pass filters (LPF). Edges, sharp transitions and noise in the grey levels contribute to high frequency contents in an image. A low pass filter only passes low frequency and blocks the high ones. It removes noise and introduces blurring as a side effect in the image.

Ringing is undesirable and unpleasant lines around the objects present in the image. As the cut of frequency D_0 increases, effect of ringing reduces. Ringing is a side effect of ideal *lpf*.

$\uparrow D_0$ ———— $\downarrow$ Ringing + blur

E6) LPF are generally used as a preprocessing step before an automatic recognition algorithm. It is also used to reduce noise in images. Few examples are listed below.

Character Recognition, Object counting, Printing and publishing industry, "Cosmetic" processing etc.

E7) The sharpening filters are listed as follows:

1. Ideal high pass filter
2. Butterworth high pass filter
3. Gaussian high pass filter

High pass filters are used for enhancing edges. These filters are used to extract edges and noise is enhanced, as a side effect.

E8) Gaussian high pass filters have smooth transition between passband and stop band near cut off frequency. The parameter D is a measure of spread of the Gaussian curve. Larger the value D_0, larger is the cut off frequency. Transfer function of GHPF is

$$H(u,v) = 1 - e\frac{-D^2(u,v)}{2D_0^2},$$

where $D_0 =$ cut off frequency and $D(u, v)$ is the distance from origin of Fourier transform.

E9) Image degradation can happen due to

a) **Sensor distortions:** Involves quantization, sampling, sensor noise, spectral sensitivity, de-mosaicking, non linearity of sensor etc.
b) **Optical distortions:** are geometric distortion, blurring due to camera mis-focus.
c) **Atmospheric distortions:** are haze, turbulence etc.

d) **Other distortions:** Low illumination, relative motion between object and camera etc.

E10) Fig shows the block diagram of degradation/restoration model. Degradation function $h(x, y)$ and noise $n(x, y)$, operate on input image $f(x, y)$ to generate a degraded and noisy image $g(x, y)$.

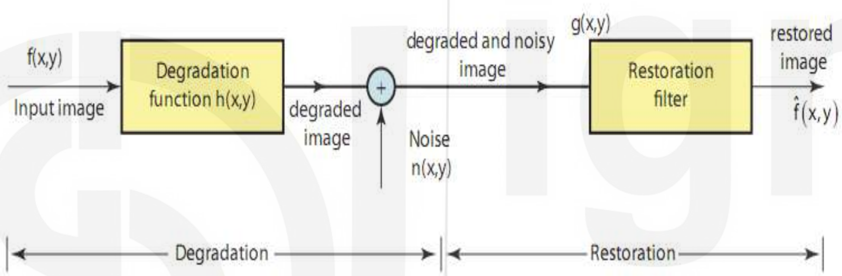


Fig.: Block diagram of degradation/restoration model

$f(x, y) =$ original image
$h(x, y) =$ degradation function
$n(x, y) =$ additive noise
$g(x, y) =$ degraded and noisy image
$\hat{f}(x, y) =$ restored image

E11) Noise is a disturbance that causes fluctuations in pixel values. Pixel values show random variations and can cause very disturbing effects on the image. Thus suitable strategies should be designed to model and remove/ reduce noise. Major source of noise in digital images is during image acquisition. Non-ideal image sensors and poor quality of sensing elements contribute to majority of noise. Environmental factors such as light conditions, temperature of atmosphere, humidity, other atmospheric disturbances also account for noise in images. Transmission of image is also a source of noise. Images are corrupted with noise because of interference in the channel, lightning and other disturbances in wireless network. Human interference also plays a part in addition of noise in images.

Properties of Noise

Spatial and frequency characteristics of noise are as follows:

1) Noise is assumed to be 'white noise' (it could contain all possible frequency components), as such,Fourier spectrum of noise is constant.

2) Noise is assumed to be independent in spatial domain. Noise is '**uncorrelated**' with the image, that is, there is no correlation between pixel value of image and value of noise components.

Based on noise properties and types of noise, different filters are used to reduce/remove noise.

E12) **Gaussian Noise**

Gaussian noise model is most frequently used in practice. The PDF of a Gaussian random variable 'z' is given by

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-u)^2}{2\sigma^2}},$$

where $z =$ intensity/grey level value
$\mu =$ mean (average) value of z
$\sigma =$ standard deviation

Rayleigh Noise

Radar range and velocity images typically contain noise that can be modeled by the Rayleigh distribution. Rayleigh distribution is defined by

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} & z \geq a \\ 0 & z < a \end{cases}$$

Mean density is given $\mu = a + \sqrt{\pi b/4}$ as $\sigma^2 = \frac{b(4-\pi)}{4}$

Erlang (Gamma) Noise

Erlang noise is given by

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-z} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

a and b are positive integers. Mean density is given by

and variance $\sigma^2 = \frac{b}{a^2}$ is $\mu = \frac{b}{a}$

Uniform Noise

Uniform noise is specified as

$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$

Then mean and variance of uniform noise is given by

$$\mu = \frac{a+b}{2}, \sigma^2 = \frac{(b-a)^2}{12}$$

Impulse (Salt and Pepper) noise

Impulse (salt and pepper) noise is specified as

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \end{cases}$$

E13) There are three methods of estimation of degradation function:
a) Observation
b) Experimentation
c) Modelling

E14) In presence of noise, degradation model as shown in figure 4 becomes

$$G(u,v) = F(u,v)H(u,v) + N(u,v)$$

After applying inverse filtering

$$\hat{F}(u,v) = H_R(u,v)G(u,v)$$

Substituting values of $H_R(u,v)$ and $G(u,v)$

$$\begin{aligned}\hat{F}(u,v) &= \frac{1}{H(u,v)}\left[F(u,v)H(u,v) + N(u,v)\right] \\ &= \frac{F(u,v)H(u,v)}{H(u,v)} + \frac{N(n,v)}{H(u,v)} \\ &= F(u,v) + \frac{N(u,v)}{H(u,v)}\end{aligned}$$

Limitations of inverse filtering are:

1) It is an unstable filter
2) It is sensitive to noise. In practice, inverse filter is not popularly used.

E15) This approach includes both the degradation function and power spectrum of noise characteristics in developing the restoration filter. Wiener filter restores the image in the presence of blur as well as noise.

This method is founded by considering image and noise as random variables and objective is to find as estimate $\hat{f}$ of the uncorrupted image f such that the mean square error between them is minimized. This error is given by

$$e^2 = E\{(f - \hat{f})^2\})$$

Where $E\{.\}$ is the expected value of the argument. Noise and image are assumed to be uncorrelated.

$$\Rightarrow \boxed{H_R(u,v) = \frac{S_{fg}(u,v)}{S_{gg}(u,v)}}$$

$$\begin{aligned}\hat{F}(u,v) &= H_R(u,v)G(u,v)\\ &= \left[\frac{H^*(u,v)}{|H(u,v)|^* + k}\right]G(u,v)\\ &= \left[\frac{1}{H^*(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + k}\right]G(u,v)\end{aligned}$$

k is chosen experimentally and iteratively for best results.

UNIT 7

COLOUR IMAGE PROCESSING

Structure **Page No.**

7.1 INTRODUCTION

The purpose of this unit is to introduce the concepts related to colour image processing. As we have been working with grayscale images till now, we would like to have an in-depth discussion of how colour images are formed and the various colour models that exist.

We shall first discuss the human vision system in Sec. 7.2. A healthy vision system is capable of seeing the world in colour.

As you read further, we shall discuss the various colour models that exist and the advantages and disadvantages of each in Sec. 7.3.

Finally, we shall discuss pseudo-colour processing, also called false colour, which is the process of assigning colours to grey values based on specified conditions. We shall discuss colour processing in Sec. 7.5. We finally summarise the discussion in Sec. 7.6 and in Sec. 7.7, we give the solutions/answers/hints to the exercises.

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:

- to differentiate between thousands of thousands of colours and their shades in the colour images.
- To define the different colour models and use them as per the requirements.
- To apply different pseudo colour models

A colour image is a powerful source of information. Human visual system has the ability to differentiate between hundreds of colours and their shades. Therefore, colour images contain a large amount of extra information compared to grey-scale images, that give a better understanding of the contents of the image, for example, in object detection and segmentation. If an image is captured by a full-colour sensor, then the resulting image is a full colour image.

A grayscale image can be converted into a colour image using the technique of pseudo-colour processing, where each intensity is assigned a colour.

Full colour image processing is primarily used in most applications such as visualisation and publishing. We start with discussion on human vision system in the following section.

7.2 HUMAN VISION SYSTEM

The human eye is nearly spherical in shape with a 20mm diameter on an average, as shown in Fig. 4.1. The three main parts of the eye are:

i) The cornea and sclera outer cover,

ii) the choroid and

iii) the retina.

Let us discuss them one by one briefly.

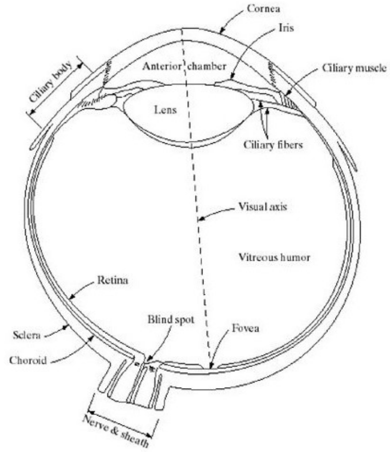


Fig.1: Structure of the human eye source

i) As you see in Fig. 1, the sclera is an opaque member that encloses theoptic globe all around, except at the anterior end, which is covered by the cornea. The cornea is a tough, transparent cover of the anterior chamber.

ii) Choroid is the layer under the sclera. The membrane choroid contains a network of blood vessels. These blood vessels form the major source of nutrition for the eyes. If the choroid is damaged and inflamed, it can restrict blood flow in the eye, resulting in serious damage of the eye cells. The role of the choroid is also to control the amount of light entering the eye as well as reduce the backscatter inside the eye.

    The choroid is divided into two parts:

    a. The ciliary muscles which relax and tighten to enable the lens to focus by changing its shape,

    b. The iris diaphragm, that contracts and expands to control the amount of light that enters the eye.

    The lens is a transparent, biconvex structure that helps to refract light into the eye such that the image is formed on the retina. The lens is flexible and ca change shape to change the focal length of the eye. THSI ensures that objects at various distances can be focussed upon and their images can be formed on the retina.

iii) The retina is the innermost membrane of the eye. It lines the wall of the complete posterior portion of the eye. The retina can be thought of as the image plane in the eye, since on properly focussing the eye on an object, light from that object is passed through the lens such that the image is formed on the retina.

    The retina consists of two types of cells called rods and cones. The cones are highly sensitive to colour and are around 6-7 million in a human eye. The cones are located on the focea which is the central portion of the retina.

    However, there are 75-150 million rod cells which are completely distributed all over the retina. The rod cells give the overall picture of the object in the scene, and reduce the amount of detail. Rods are also responsible for low light vision, also known as **SCOTOPIC vision**, while cones are responsible for bright light vision, also known as **PHOTOPIC vision**.

    In the human eye, the distance between the retina and lens, that is the focal length, varies between 14 and 17 mm as the refractive power of the lens increases from min to max. For a nearby object, the lens is most strongly refractive. Moreover, the lens of the eye is very flexible and is flattened by controlling muscles to enable the eye to focus on distant objects.

    To allow the eye to focus on objects close to the eye, the controlling muscles allow the lens to become thicker.

Here, you might be wondering how human eye adapts to different levels of brightness and how it discriminates various levels of brightness. The answer to your question is given below.

Brightness Adaptation and Discrimination: Human vision system is highly complex and can adapt to an enormous range of light intensity levels-of the order of $10^{\wedge}10$. The range starts from the scotopic threshold and goes upto the glare limit. The subjective brightness, the perceived intensity by the human eye, has been experimentally found to be a logarithmic function of the light intensity that falls on the eye. Since the human eye cannot interpret this dynamic range simultaneously, brightness adaptation is carried out by the eye. The eye can discriminate only a small range of distinct intensity levels simultaneously. Brightness adaption level is the current sensitivity level of a human eye for a given set of conditions.

Now, try the following exercises.

E1) If an observer is looking at a tree that is 100m far and if h is the height of the tree in mm in the retinal image, what is h?

So, by now you know the fundamental concepts about human vision system. In the following section, we are going to highlight various colour models. You must have heard about some of them in your day-to-day life.

7.3 COLOUR FUNDAMENTALS

Every colour is defined using three quantities that are independent of each other, however, taken together they define a particular shade of a colour. These quantities are:

i) **Hue:** Hue component of a colour is defined by the dominant wavelength. The wavelength range on the electromagnetic spectrum that defines the visible colour spectrum lies between 400nm [nm is nanometre] that represents the violet colour and 700nm that represents the red colour as can be seen in Fig. 2.

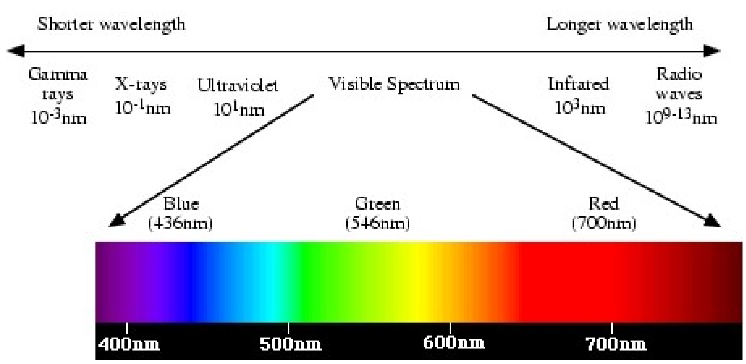


Fig. 2: Part of the electromagnetic spectrum that shows the visible spectrum Source:

ii) **Saturation**: The excitation purity of the colour is determined by the quantity known as **saturation.**It is dependent on the amount of white light that is mixed with hue of that colour. A fully saturated colour implies that no white light is mixed with that hue.

iii) **Chromaticity**: The sum of hue and saturation constitutes the *chromaticity* of the colour. Therefore, if there is no colour, it is called achromatic light.

iv) **Intensity**: The amount of light actually present defines the *intensity*. Therefore, intensity is a physical quantity. If more light is present, the colour is more intense. Achromatic light has only intensity but no colour. Grayscale images have only intensity

v) **Luminance or Brightness**: The perception of colour is the quantity known as luminance or brightness. For example, given two colours of the same intensity, such as blue and green, it is perceived that blue is much darker than green.

vi) **Reflectance**: The ability of an object to reflect light, is the reflectance property of the object. The reflectance property determines the colour of the object, since, we see those colours that are reflected back and not the ones that are absorbed. For example, an object that reflects green, absorbs all other colours in the white light spectrum except green.

There are about 6 to 7 million cones in the human eye and they are responsible for recognising colours. Nearly, 65% of the cones recognise red, 33% are sensitive to green and about 2% to blue. Red, green and blue are known as the primary colours and nearly all other colours are seen as a combination of these primary colours. However, there is a difference between the primary colours of light and the primary colours of pigments. The primary colours of light are red, blue and green and they can be added to produce the secondary colours of light that are yellow(red plus green), magenta (red+blue) and cyan (green + blue). Moreover, the primary colours of pigments are said to be those that absorbs a primary colour of light and reflects the other two. Therefore, in tHSI case, the primary colours are cyan, magenta and yellow while the secondary colours are red, blue and green.

For standardisation, in 1931, the *Commission Internationale de l'Éclairage (CIE)* defined specific wavelengths for the three primary colours: red = 700nm, blue: 435.8 nm and green = **546.1** nm. The amounts of red, blue and green required to form a colour are known as the Tristimulus values and are denoted as X, Y and Z.

Given X, Y and Z the tristimulus coefficients which define a colour. If we define x, y and z as the relative values of the primary colours, then these values can be found by

$$x = \frac{X}{X+Y+Z}, \; Y = \frac{Y}{X+Y+Z} \text{ and } z = \frac{Z}{X+Y+Z} \qquad \text{... (1)}$$

It is obvious that

$$x + y + z = 1. \qquad \text{... (2)}$$

Thus a 2-D diagram is adequate to show the coordinates x and y.

If we specify colours as a composition as x (red) and y (green). Then, given the values of x and y, the value of z (blue) can be computed as:

$$z = 1 - (x + y) \quad \text{... (3)}$$

Here, we can see only two variables are independent. Therefore, we can show these variables in 2-D coordinate system.

The point on the boundary of the chromaticity chart is fully saturated, while as a point moves farther from the boundary, more white light is added and is therefore, less saturated. The saturation is zero at the point of equal energy. A straightine joining any two points in the chromaticity diagram, determines all possible colours that can be obtained by combining the two colours at the endpoints of the segment. This can be extended to combining three colours. The three line segments joining the points pairwise form a triangle and various combinations of the colours at the vertices of this triangle give all colours inside the triangle or on the boundary of the triangle.

To understand this more clearly, we shall discuss few examples.

Example 1: Consider the coordinates of warm white $(0.45, 0.4)$ and the coordinates of deep blue $(0.15, 0.2)$. Find the percentage of the three colours red (X), green (Y) and blue (Z).

Solution: We first find the trichromatic coefficients x, y and z. At the point warm white, $x = 0.45$ and $y = 0.4$, therefore

$$\begin{aligned} z &= 1 - (x + y) \\ &= 1 - (0.45 + 0.4) \\ &= 0.15 \end{aligned}$$

Now, we shall find the tristimulus values X, Y and Z.

$$\frac{X}{X+Y+Z} = 0.45$$
$$\frac{Y}{X+Y+Z} = 0.4$$
$$\frac{Z}{X+Y+Z} = 0.15$$

Here, $X : Y : Z = 0.45 : 0.4 : 0.15$

Therefore the percentage of each colour would be as follows:

Percentage of red $(X) = 45\%$
Percentage of green $(Y) = 40\%$
Percentage of Blue $(Z) = 15\%$

At the point deep blue, $x = 0.15,\ y = 0.2,$ therefore $z = 0.65.$

We can find the percentage of each colour as we found in case of warm white. We get percentage of red colour as $15\%,$ percentage of green colour as 20% and the percentage of blue colour as $65\%.$

We can see the percentage is justified for each colour name.

***

Example 2: Find the relative percentage of colours warm white and deep blue which mixes the give the colour which lies on the line joining them. Use the coordinates of these points as given in Example 1.

Solution: Let the colour C lies on the line have the coordinate $(x, y).$

The distance of C from the warm white colour $= \sqrt{(x-0.45)^2 + (y-0.4)^2}$

Similarly, the distance of C from the deep blue colour
$= \sqrt{(0.15-x)^2 + (0.2-y)^2}$

The percentage of warm white in

$$C = \frac{\sqrt{(x-0.45)^2 + (y-0.4)^2} - \sqrt{(0.15-x)^2 + (0.2-y)^2}}{\sqrt{(0.45-0.15)^2 + (0.4-0.2)^2}} \times 100$$

This expression can be used to find the percentage of warm white colour at C by substituting the coordinates of the point C as per the situation. Also, the percentage of the deep blue colour would be $(100$ - percentage of warm white colour).

***

Now try the following exercises.

E2) Derive an expression to find the percentage of each colours $C_1, C_2,$ and C_3 at the point C which lies within the triangle having vertices as $C_1, C_2,$ and $C_3.$

In the following section, we shall discuss the most commonly used colour models such as the RGB (red, green, blue), CMY (Cyan, magenta, yellow) and HSI (hue, saturation, intensity).

7.4 COLOUR MODELS

Colour models or Colour spaces or Colour systems have been introduced so as to be able to specify each colour in a generally accepted manner. There are various colour models or colour spaces. Each colour space specifies a particular colour in a standard manner, by specifying a 3-D coordinate system and a subspace that contains all possible colours in that colour model. Then, each colour in that colour space is represented as a point in that subspace, given by three coordinates $(x, y, z).$ These colour models are either oriented towards specific hardware or image processing applications. In this section, we shall discuss three important colour models and the conversion of one colour model into other.

Before we discuss each colour model, let us discuss the principles of absorption of colours of any model by human eye.

i) The human eye has absorption characteristics of colours and recognises them as variables. Thus, the colours red (R), green (G) and blue (B) are called **primary colours** of light.

ii) Secondary colours of light are produced by adding primary colours. For example red and blue produces magenta, red and green produces yellow, green and blue produces cyan, etc.

iii) Proportion of primary and secondary colours in appropriate amount produces white light.

Now, let us discuss each model separately.

7.4.1 The RGB Model

The RGB colour is based on a cartesian coordinate system, where the colour subspace is a cube with axes representing red, green and blue. A colour in the RGB model is therefore, specified as a 3-tuple (R, G, B) where, R, G and B represent the amount of red, green and blue, respectively, present in that colour. The geometry of the RGB colour model is a cube as shown in Fig. 3. It is assumed that all colour values are normalised, that is, it is a unit cube and all values of R, G and B lie between 0 and 1. It is clear from Fig. 3, that it is a model of a cube with the three axes for Red, Green and blue. The grayscale values lies on the diagonal of the cube, which joins the black $(0,0,0)$ and white $(1,1,1)$ vertices of the cube.

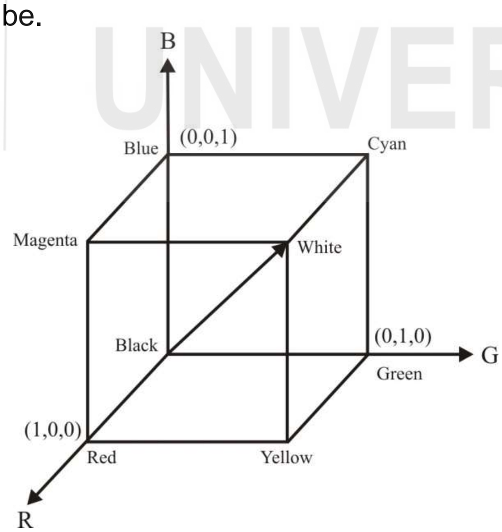


Fig. 3: The RGB colour (Image taken from [1])

A colour image in the RGB model consists of three images corresponding to each of the three colours: Red (R), Green (G) and Blue (B) colours. These three images combine to form one composite colour image on a monitor. To covert an RGB image to a grayscale image, the intensity of the gray-pixel is given by the average of R, G and B values. The RGB colour model is mainly used for colour monitors and screens.

Now the question arises how do we find the composite colour in RGB colour model at any point. For this we follow the following steps:

Step 1: Pixel depth is the number of bits used to represent each pixel. If an image in RGB model has 8-bit image in each of its three colours, then each RGB pixel has a depth of 3 image planes $\times$8-bit per plane that is 24 bits. This gives rise to 2^{24} colour shades.

Step 2: We fix one of the three colours and let the other two colours to vary. Suppose we fix $R = 127$ and let G and B to vary. Then the colour at any point on the plane parallel to GB plane would be $(127, G, B)$, where, $G, B = 0, 1, ..., 255.$

Example 3: In a RGB image, the R and B components are at mid and the G component is at 1, then which colour would be seen by a person?

Solution: At the given point, we have

$$\frac{R}{2} + \frac{B}{2} + G = \frac{1}{2}(R + G + B) + \frac{G}{2}$$
$$= \text{mid grey} + \frac{1}{2}G.$$
$$= \text{Pure green with some grey component.}$$

***

Now, try an exercise.

E3) How many different shades of grey are there in a colour RGB system if each RGB image is an 8 bit image?

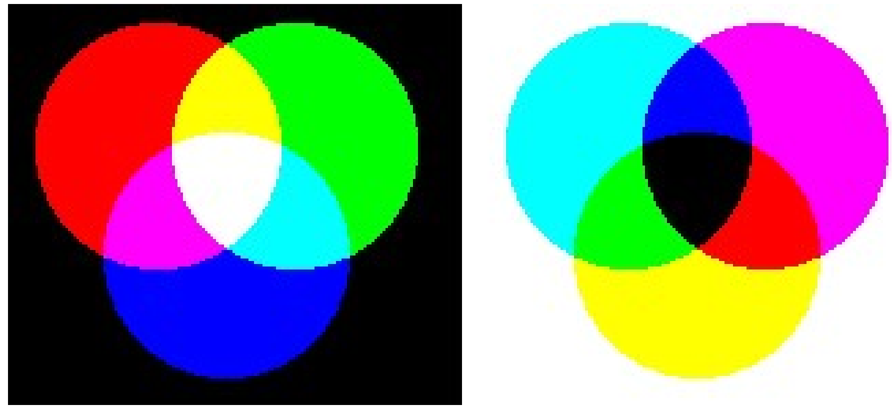
After discussing RGB model, we discuss CMY and CMYK colour models.

7.4.2 The CMY and CMYK Colour Model

Cyan (C), Magenta (M) and Yellow (Y) are the primary colours of pigments and the secondary colours of light. The CMY model is a subtractive model, implying that it subtracts a colour from the white light and reflects the rest. For example, when white light is reflected on cyan, it subtracts Red and reflects the rest. While RGB is an additive model, where something is added to black (0,0,0) to get the desired colour CMY is a **subtractive model**. The conversion between CMY and RGB model is given by the Equation below.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \text{... (4)}$$

where, the RGB values have been normalised. THSI also gives a method to convert from RGB to CMY to enable printing hardcopy, since the CMY model is used by printers and plotters.



(a) **(b)**

Fig. 4: (a) RGB and (b) CMY

Fig. 4(a) shows the RGB model, in which the white colour is produced by adding the three primary colours Red, Green and Blue. Fig. 4 (b) shows the CMY model, where black is obtained as the sum of Cyan, Magenta and Yellow. The inverse relation between the RGB and CMY models are also shown by these two images.

In practice, black is obtained by combining cyan, yellow and magenta, however, this leads to a muddy looking black. For publishing, the black colour plays an important role, therefore, in the CMYK colour model, black is added as the fourth colour, where K stands for black.

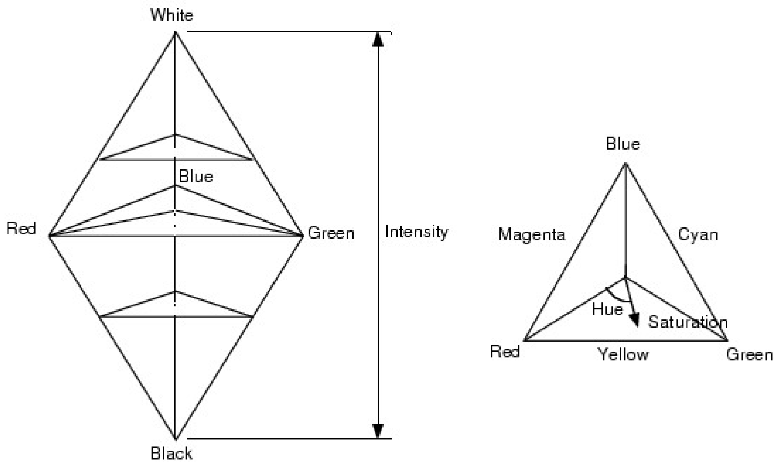
Now check by doing the following exercise, what have you understood.

E4) Why do we get green coloured paint on mixing blue and yellow coloured paints?

Now, let us discuss HSI model.

7.4.3 The HSI Model

This colour model is very close to human colour perception which uses the hue, saturation and intensity components of a colour, when we see a colour, we cannot describe it in terms of the amount of cyan, magenta and yellow that the colour contains. Therefore, the HSI colour model was introduced to enable describing a colour by its hue, saturation and intensity/ brightness. Hue describes the pure colour, saturation describes the degree of purity of that colour while intensity describes the brightness or colour, sensation. In a grayscale image, intensity defines the graylevel. Fig. 6 shows the HSI colour model and the way colours may be specified by this colour model.



(a) (b)

Fig. 5: (a) HSI and (b) RGB

In Fig.5, the HSI colour model is represented and its relation to RGB model is shown in the Fig. 5 (b). The HSI triangle in Fig. 5 (b) shows a slice from the HSI solid at a particular intensity as shown in Fig. 5 (a).

You may notice that in Fig. 5, the hue, saturation and intensity values required to form the HSI colour space can be computed using the RGB values.

Try the following exercises.

E6) Write the full form of HSI and define each of the components.

E7) What is colour space? Mention its classification.

Now we indicate how a RGB colour model is converted to a HSI colour model.

To convert an image in RGB format to HSI colour space, the RGB value of each pixel in the image, is converted to the corresponding HSI value in the following manner. Hue, H is given by

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases} \quad \text{... (5)}$$

where,

$$\theta = \cos^{-1}\left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right\}$$

Saturation, S is given by

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)]$$

And, intensity, I is given by

$$I = \frac{1}{3}(R + G + B)$$

Where, the RGB values have been normalised in the range $[0,1]$ and the angle θ is measured with respect to the red axis in the HSI space.

Now we would convert HSI colour model to RGB colour space.

Given pixel values in the HSI colour space in the interval $[0,1]$, the RGB values can be computed in the same range. However, depending on the H value, the RGB values are computed in different sectors, based on the separation of the RGB colours by 120° intervals.

First, multiply the H value by 360°, to get the hue value in the interval [0°, 360°].

In the RG sector, H takes the value in the interval [0°, 120°[,

that is, $0^\circ \le H < 120^\circ$ and the RGB values are given by

$$B = I(1 - S) \qquad \text{... (8)}$$

$$R = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right] \qquad \text{... (9)}$$

$$G = 3I - (R + B) \qquad \text{... (10)}$$

In the **GB Sector,** $120^\circ \le H < 240^\circ$ then in this case, we first convert the value of H as

Then, the RGB values are computed as

$$R = I(1 - S) \qquad \text{... (11)}$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right] \qquad \text{... (12)}$$

$$B = 3I - (R + B) \qquad \text{... (13)}$$

In BR sector, when, $240^\circ \le H \le 360^\circ$, we first convert H as $H = H - 240^\circ$

Then, the RGB values are

$$G = I(1 - S) \qquad \text{... (14)}$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right] \qquad \text{... (15)}$$

$$R = 3I - (R + B) \qquad \dots (16)$$

Example 4: Consider the image with different colours as given in Fig. 6 Write the RGB colours which would appear on monochrome display. You may assume that all colours are at maximum intensity and saturation. Also show each of the colour in black and white considering them as 0 and 255 respectively.

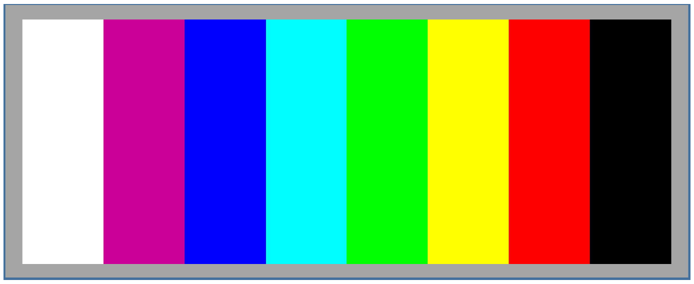


Fig. 6

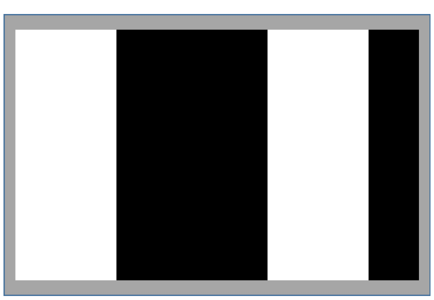
Solution: It is given here that the intensity and saturation are maximum, therefore the value of each of the components RGB would be 0 or 1.

Let us check each colour of the image one by one by starting from the left most colour.

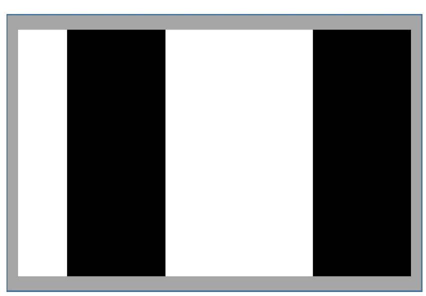
Colour	**RGB combination**	**Intensity/Saturation**			**Monochrome colours**		
		R	**G**	**B**	**R**	**G**	**B**
White	R+G+B	1	1	1	255	255	255
Magenta	R+B	1	0	1	255	0	255
Blue	B	0	0	1	0	0	255
Cyan	G+B	0	1	1	0	255	255
Green	G	0	1	0	0	255	0
Yellow	R+G	1	1	0	255	255	0
Red	R	1	0	0	255	0	0
Black	NIL	0	0	0	0	0	0

Now hence forth we shall follow the conversion that 0 represents black and 255 represents white. Also, the grey is represented by 128. You see that the table has R colour series as 255, 255, 0, 0, 255, 255, 0. Thus, it would show W, W, B, B, B, W, W, B in monochrome display, which is shown in Fig. 7 (a).

Similarly monochrome display of green colour would be shown by the series W,B,B,W,W,W,B,B and blue would be shown as W, W, W, W, B, B, B, B as shown in Fig. 7 (b) and Fig. 7 (c).



(a)



(b)

(c)

Fig. 7

***

Example 5: Let us sketch the HSI components of the image considered in Fig. 6 [Given in Example 4] on a monochrome display.

Solution: We transform HSI by computing values of H, S and I for each colour.

For white $R=1, G=1, B=1$

Using Eqn. (5), we can say that H does not exist as denominator is zero while computing θ.

Using Eqn. (7), we get $I=\frac{1}{3}(1+1+1)=1$ and using Eqn. (6), we get

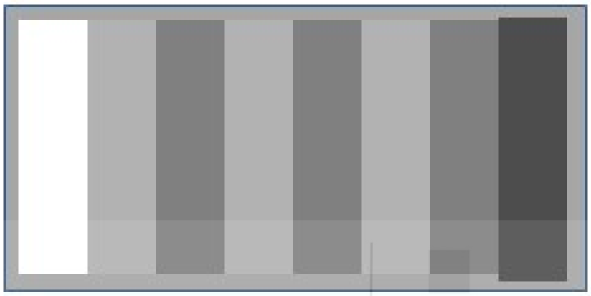
$S=1-\frac{3}{1+1+1}[\min(1,1,1)]=0.$

Similarly we can find H, S, I for each of the colours as shown in the following table.

Colour	R	G	B	H	S	I	Monochromatic		
							H	S	I
White	1	1	1	Cannot be computed	0	1	–	0	255
Magenta	1	0	1	$\frac{5}{6}$	1	$\frac{2}{3}$	213	255	170
Blue	0	0	1	$\frac{2}{3}$	1	$\frac{1}{3}$	170	255	85
Cyan	0	1	1	$\frac{1}{2}$	1	$\frac{2}{3}$	128	255	170
Green	0	1	0	$\frac{1}{3}$	1	$\frac{1}{3}$	85	255	85
Yellow	1	1	0	$\frac{1}{6}$	1	$\frac{2}{3}$	43	255	170
Red	1	0	0	0	1	$\frac{1}{3}$	0	255	85
Black	0	0	0	–	0	0	–	–	0

(a)

(b)



(c)
Fig. 8

The output is given in Fig. 8 (a), Fig. 8 (b) and Fig. 8 (c) for each of the attribute H,S and I.

***

Now try the following exercises.

E8) Describe how the grey levels vary in RGB primary images that make up the font face of the colour cube.

E9) Transform the RGB cube by its CMY cube. Lable all the vertices. Also, interpret the colours at the edges with respect to saturation.

In the following section, we discuss pseudocolour image processing.

7.5 PSEUDOCOLOUR IMAGE PROCESSING

Pseudocolour image processing is the process of assigning colour to each pixel of a grayscale image based on specific conditions. As mentioned above, colour carries with it a large amount of information regarding the objects that we are viewing and therefore, for better visualisation, converting a grayscale image to a colour image helps in improved interpretation of the image.

Intensity slicing or density slicing is one of the simplest forms of pseudocolour image processing technique. In this technique, the image is interpreted as a 3D function and can be imagined as a set of 2D grid which are parallel to the coordinate planes and placed at each intensity value. Each plane can then be thought of as a slice of the image function in the area of intersection. For example, the plane at $f(x, y) = I_i$ slices the image function into two parts. Then, any pixel whose graylevel is on or above the plane can be coded in one colour and whose graylevel is below the plane can be coded in another colour, thereby converting the grayscale image into a two colour image.

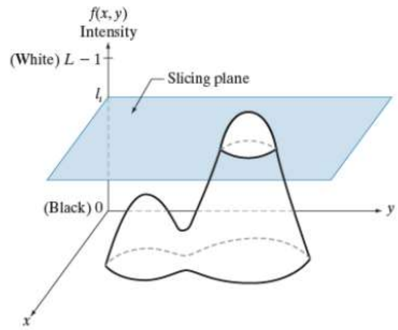


Fig. 9: Intensity slicing technique represented as slicing planes perpendicular to the intensity axis at various intensity levels.

In general, intensity slicing technique is given as follows:

Let $[0, L-1]$ be the existing grayscale values such that black is represented by I_0, that is, $f(x,y)=0$ and white is represented by I_{L-1}, that is, $f(x,y)=L-1$. Then, we define P planes such that they are perpendicular to the intensity axis at levels $I_1, I_2, ..., I_P$, where $0 < P < L-1$. These P planes are parallel to the coordinate plane. The grayscale intensity levels are partitioned into $P+1$ intervals, namely, $V_1, V_2, ... V_{P+1}$. Then, assign a colour to each pixel in the following manner

$$f(x,y) = c_k \text{ if } f(x,y) \in V_k$$

where, V_k is the k^{th} intensity interval defined by planes at levels $Ik-1$ and Ik and c_k is the colour associated with V_k.

You may try an exercise:

E10) Define an application of intensity level slicing.

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

7.6 SUMMARY

In this unit, we discussed the following points:

1. The need for colour image processing. Since the human eye has the wonderful capability of seeing millions of colour, we realise that colour gives a large amount of information about the objects and scene in the images.
2. We first discussed the structure of the human eye and then the tristimulus theory that connects the perception of colour with the various colour models that exist.
3. We then discussed the main colour models or colour spaces that are mainly used in both TV and print.

7.7 SOLUTIONS/ANSWERS

E1) Since when object is far, the focal length is 17 mm for the human eye, therefore, $15/100 = h/17 \Rightarrow h(17*15)/100 = 2.55\,m$

E2) THSI problem is the extention of the problem solved in Example 2. Here, we consider two possibilities.

i) When the point C at which percentage of colours C_1, C_2 and C_3 to be found is on the sides of triangle. In this case the percentage is found by considering the point on the line joining the corresponding vertices as we solved in Example 2. There would be 0% from the vertex which does not lie on the line. For example, if the point lies on the line joining C_1 and C_2, then the percentage of C_1 and C_2 can be found as given below.

Let the coordinates of C_1 be $(x_1, y_1), C_2$ be $(x_2, y_2), C_3$ be (x_3, y_3) and C be (x, y).

Percentage of C_1 in

$$C = \frac{\sqrt{(x-x_1)^2 + (y-y_1)^2} - \sqrt{(x_2-x)^2 + (y_2-y)^2}}{\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}} \times 100$$

Percentage of C_2 in $C = (100 -$ percentage of $C_1)\%$

Percentage of C_3 in $C = 0\%$.

ii) When the point C does not lie on the sides of the triangle with vertices C_1, C_2 and C_3.

In HSI case we join the point C with any of the vertices say C_3. We follow the following steps.

iii) Join the points C and C_3 and extend the line towards the side C_1C_2. Suppose it intersects C_1C_2 at C_4.

iv) Find the percentage of C_1 and C_2 at C_4.

v) Use the concept that the ratio of C_1 and C_2 will remain same at each of the points on the line C_3C_4.

vi) Now, we can easily find the coordinates of the point C_4 by writing equation of the lines C_1C_2 and C_3C. C_4 is the point of intersection of C_1C_2 and C_3C.

vii) Finally, we can find the percentage of C_4 and C_3 for the colour C.

E3) For an 8-bit image, there are $2^8 = 256$ possible values. A colour will be grey if each of the colour in RGB is same. Therefore, there can be 256 shades of grey.

E4) You can see in Fig. 5, yellow paint is made by combining green and red while imperfections in blue leads to reflection of some amount of green from blue paint also. Therefore, when both blue and yellow are mixed, both reflect the green colour, while all other colours are absorbed. Therefore, green coloured paint results from mixing of blue and yellow paints.

E5) H stands for Hue, which represents dominant colour as observed by an observer and the corresponding wavelength is also dominant. S stands for Saturation, which is the amount of white light mixed with a hue. I stands for intensity which reflects the brightness.

E7) A colour space allows one to represent all the colour perceived by human eye. The colour space can be broadly classified into (i) RGB, (ii) GMY and (iii) HSI colour space.

E8) Each of the components in RGB model would vary from 0 to 255. Here, we are discussing the front face. So, we fix all pixel values in the Red image as 255 and let the columns to vary from 0 to 255 in the green image and rows to vary from 255 to 0 in the blue image.

E9) The vertices would be as given below:

White $=(0,0,0)$

Cyan $=(1,0,0)$

Magenta $=(0,1,0)$

Blue $=(1,1,0)$

Green $=(1,0,1)$

Red $=(0,1,1)$

Black $=(1,1,1)$.

The edges which are free from black or white pixels are fully saturated. The saturation decreases towards the ends having black or white pixel.

E10) Detecting blockages in medical image processing is an application of intensity level slicing.

References

[1] R.C. Gonzoles and R.E. Woods, Digital Image Processing, Addision-wesley, 1992.

[2] A.K. Jain, Fundamentals of Digital Image Processing, PHI.

UNIT 12

OBJECT RECOGNITION USING SUPERVISED LEARNING APPROACHES

Structure **Page No.**

12.1 INTRODUCTION

Obect recognition is the discipline of building machines to perform perceptual tasks which humans are particularly good at. e.g. recognize faces, voice, identify species of flowers identify diseases in plants by observing the leaves, identify animals, etc. There are practical needs to find more efficient ways of doing things. e.g. read hand-written symbols, diagnose diseases, identify incoming missiles from radar or sonar signals. The machines may perform these tasks faster, more accurately and cheaply.

The goal of Object recognition is to clarify complicated mechanisms of decision making processes and automatic these functions using computers.

In a classification task, we are given an object or **pattern** and the task is to classify it into one out of c classes. In the previous units it was discussed that how the decision boundary surface between various classes can be used to assign a class to each point in the test data. In this unit, a different approach is being proposed, where the class assigned will depend on how close the point of the test data (that is, the pattern) is to a particular class. This gives rise to Minimum Distance Classifier.

And now, we will list the objectives of this unit. After going through the unit, please read this list again make sure you have achieved the objectives.

Objectives

After studying this unit, you should be able to

- define pattern recognition
- apply different types of classifiers
- describe discriminant Functions (linear and non-linear)
- use Bayesian classification.
- find minimum distance classifiers;
- apply machine learning algorithm;
- describe supervised learning approach;
- describe unsupervised learning approach.

We begin the unit by discussing some basic concepts in the following section.

12.2 BASIC CONCEPTS

Although humans can perform some of the perceptual tasks with ease, there is not sufficient understanding to duplicate the performance with a computer. Because the complex nature of the problems, many pattern recognition research has been concerned with more moderate problems of pattern classification — the assignment of a physical object or event to one of several pre-specified categories.

For example, a lumber mill producing assorted hardwoods wants to automate the process of sorting finished lumber according to the species of trees. Optical sensing is used to distinguish birch lumber from ash lumber. A camera takes pictures of the lumber and passes to on to a feature extractor.

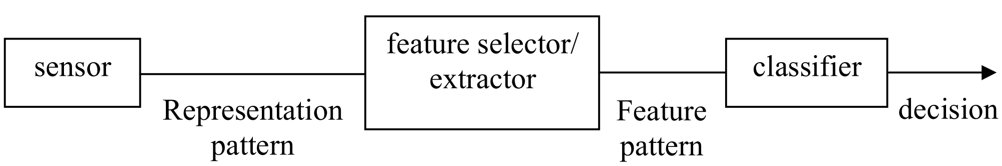


Fig. 1: A pattern classification system

The feature extractor reduces the quantum of data by measuring certain relevant properties that distinguish pictures of birch lumber from pictures of ash lumber. These features are then passed to a classifier that evaluates the evidence presented and makes a final decision about the lumber type.

Suppose that somebody at the lumber mill tells us that birch is often lighter colored than ash. Then brightness becomes an obvious feature. We might attempt to classify the lumber merely by seeing whether or not the average brightness ‘x’ exceeds some critical value.

One characteristic of human pattern recognition is that it involves a teacher. Similarly a machine pattern recognition system needs to be trained. A common mode of learning is to be given a collection of labeled examples, known as training data set. From the training data set, structure information is distilled and used for classifying new inputs.

Try an exercise.

E1) Explain a pattern classification system.

Goal of pattern recognition is to reach an optimal decision rule to categorize the incoming data into their respective categories. A pattern recognition investigation may consist of several stages, enumerated below. Not all stages may be present; some may be merged together so that the distinction between two operations may not be clear, even if both are carried out; also, there may be some application-specific data processing that may not be regarded as one of the stages listed. However, the points below are fairly typical.

1. **Formulation of the problem:** gaining a clear understanding of the aims of the investigation and planning the remaining stages.
2. **Data collection:** making measurements on appropriate variables and recording details of the data collection procedure (ground truth).
3. **Initial examination of the data:** checking the data, calculating summary statistics and producing plots in order to get a feel for the structure.
4. **Feature selection or feature extraction:** selecting variables from the measured set that are appropriate for the task. These new variables may be obtained by a linear or nonlinear transformation of the original set (feature extraction). To some extent, the division of feature extraction and classification is artificial.
5. S**upervised pattern classification:** The system is trained by providing the relevant feature samples with each sample labeled as belonging to a particular class. This help to develop the classifier.
6. **Apply discrimination:** or regression procedures as appropriate. The classifier is designed using a training set of exemplar patterns.
7. **Assessment of results:** This may involve applying the trained classifier to an independent test set of labeled patterns.

8. **Interpretation:** Result interpretation is very important. In case, results are not satisfactory, cycle may be started again from data collection.

We now define the basic terms which are used in the process of pattern recognition.

Feature

Feature is a property (or characteristics) of an object (quantifiable or non quantifiable) which is used to distinguish between (or classify) two different objects. For example, sorting incoming fish on a conveyor according to species using optical sensing. Two category of spices are there: Sea bass, Salmon.

Some properties that could be possibly used to distinguish between the two types of fishes are

• Length
• Lightness (Dark colour or light colour)
• Width
• Number and shape of fins
• Position of the mouth, etc…

This is the set of all suggested features to explore for use in the classifier.

Feature Vector

A Single feature may not be useful always for classification. A set of features used for classification form a feature vector. For example, here the relevant feature vector could be

Fish $x^T = [x1, x2] =$ [Lightness, Width]

Feature Space

The samples of input (when represented by their features) are represented as points in the feature space. If a single feature is used, then the feature space is one- dimensional feature space shown in fig. 2. If number of features is 2, then we get points in 2D space as shown in the Fig. 3. We can also have an n-dimensional feature space.

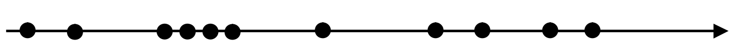


Fig. 2: Point Representing Samples in 1-D Space

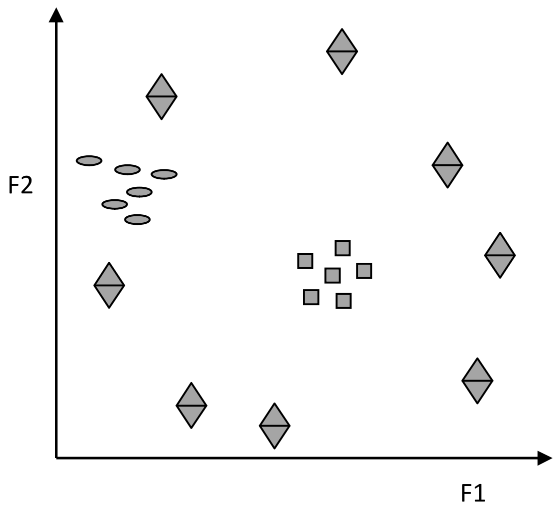
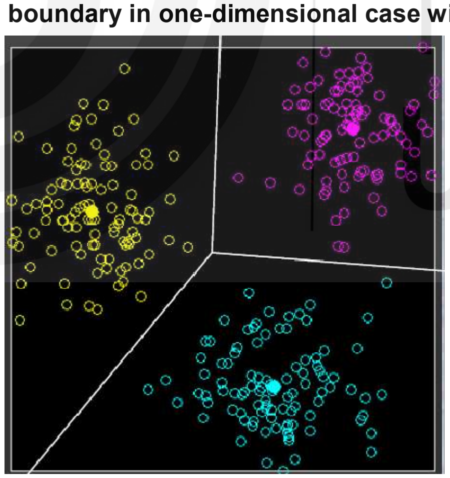


Fig. 3: Sample Points in a 2-Dimensional Feature Space

Decision Region and Decision Boundary

As we know that of pattern recognition is to reach an optimal decision rule to categorize the incoming data into their respective categories (classes). The decision boundary separates points belonging to one class from points of other. The decision boundary partitions the feature space into decision regions. The nature of the decision boundary is decided by the discriminant function which is used for decision. It is a function of the feature vector.

(a) Decision boundary in one-dimensional case with two classes



b) Decision Boundary in 2 (or 3-Dimensional Case with Three Classes

Fig. 4

Hyper Planes and Hyper Surfaces

For two category case, the decision boundary is a line $g(x) = 0$, where $g(x)$ is the discriminant function. When the coordinates of a point are substituted in $g(x)$ and this results in a positive value, we say it belongs to class 1, while a negative value decides the other class. If the number of dimensions is three, then the decision boundary will be a plane or a

3-D surface. The decision regions become semi-infinite volumes. If the number of dimensions increases to more than three, then the decision boundary becomes a hyper-plane or a hyper-surface. The decision regions become semi-infinite hyperspaces.

Learning

The classifier to be designed is built using input samples which is a mixture of all the classes. The classifier learns how to discriminate between samples of different classes. In supervised learning, the classifier is first given a set of training samples and the optimal decision boundary found, and then the classification is done.

Error

The accuracy of classification depends on two things, which are

i) The optimality of decision rule used: The central task is to find an optimal decision rule which categories the training samples correctly, and can generalise to correctly categorising unseen samples as far as possible. This decision theory leads to a minimum error-rate classification.

ii) The accuracy in measurements of feature vectors: This inaccuracy is because of presence of noise. Hence the classifier should deal with noisy and missing features too.

There are various types of classifiers used. We define them as following:

a) **Nonparametric:** Nonparametric techniques do not rely on a set of parameters/weights.

b) **Parametric:** These models are parameterized, with its parameters/weights to be determined through some parameter optimization algorithm, which are then determined by fitting the model to the training data set.

c) **Supervised:** The training samples are given as some input/output pairs. The output is the desired response for the input. The parameters/weights are adjusted so as to minimize the errors between the response of the networks and the desired response.

d) **Unsupervised:** Suppose that we are given data samples without being told which classes they belong to. There are schemes that are aimed to discover significant patterns in the input data without a teacher (labeled data samples).

Try the following exercises.

E2) What is a feature space?

E3) You are given set of data $S=${dolphin, Pomeranian dog, humming bird, frog, rattlesnake, bat}. Developing a suitable classification strategy of classify the given set S according to their nature of existence.

In the following section, we shall discuss discriminant functions.

12.3 DISCRIMINANT FUNCTIONS

Discriminant functions (DFs) are useful for representing classifiers in a simpler way. If we have a set of K classes then we may define a set of K discriminant functions $y_k(x)$, one for each class. Data point 'x' is assigned to class 'c' if

$$y_c(x) > y_k(x) \qquad \text{for all } k \neq c.$$

Formulating a pattern classification problem in terms of discriminant functions is useful since it is possible to estimate the discriminant functions directly from data. Direct estimation of the decision boundaries is sometimes referred to as discriminative modeling. The choice of discriminant function may depend on prior knowledge about the patterns to be classified or may be a particular functional form whose parameters are adjusted by a training procedure. Many different forms of discriminant function have been considered in the literature, varying in complexity from the linear discriminant function (in which g is a linear combination of the x_i) to multi-parameter nonlinear functions such as the multilayer perceptron.
Here, we will discuss some discriminant functions.

Linear Discriminant Functions (LDF)

If no probability distribution or parameters are known, we can to estimate parameters of the discriminant function with Labeled data. The shape of discriminant functions is known such as shown in Fig. 5. If we have samples from 2 classes $x_1, x_2, \ldots, x_n$, we assume that 2 classes can be separated by a linear boundary l (θ) with some unknown parameters θ. Fit the "best" boundary to data by optimizing over parameters θ by minimizing classification error on training data as shown in Fig. 6.

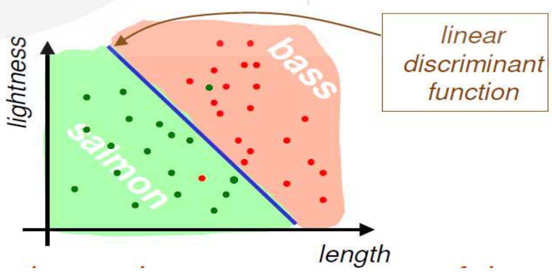


Fig. 5: Linear Discriminant Function

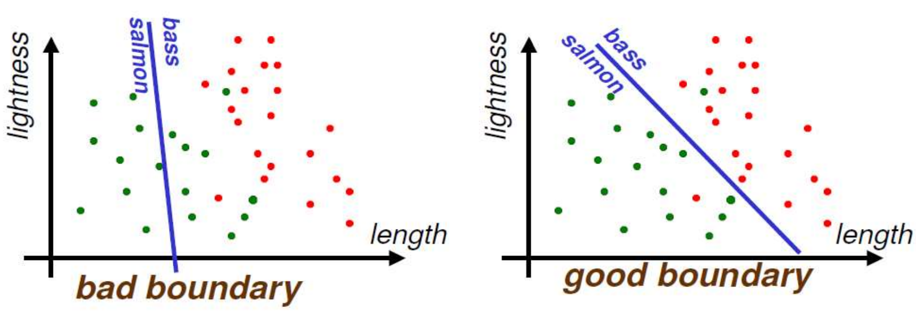


Fig. 6: Linear Discriminant Function for 2 Classes Showing 2 Boundaries

There are many different ways to representa two class pattern classifier. One way is in terms of a discriminant function $g(x)$.

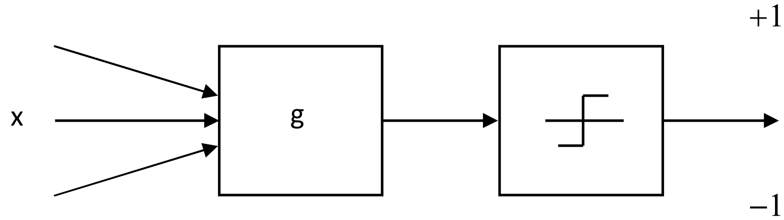


Fig. 7: Block Diagram for Linear Discriminant Function

For a new sample x and a given discriminant function, we can decide on x belongs to Class 1, if $g(x) > 0$, otherwise it belongs to class 2.
A discriminant function that is a linear combination of the components of x can be written as

$$g(x) = w^T x + W_0,$$

where w is called the weight vector and w_0 the threshold weight (also referred to as bias). These are the parameters that we want to estimate (learn) based on training data. A classifier based entirely on linear discriminant functions is called a linear classifier or a linear machine.

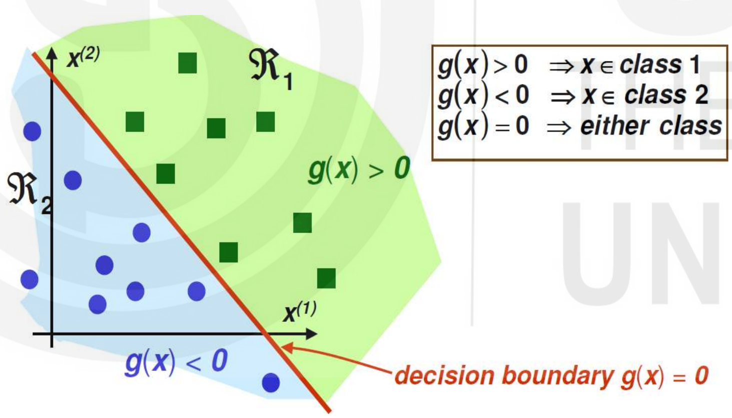


Fig. 8: Linear Discriminant Function for 2 Classes

Decision boundary surface that separates data samples assigned to Class 1 from data samples assigned to Class 2 is given by
$g(x) = w^T x + w_0 = 0.$

The equation $g(x) = 0$ defines the decision boundary. This is a hyperplane when $g(x)$ is linear. Set of vectors x are chosen, for some scalars $a_0, \ldots, a_d$ satisfy $a_0 + a_1 x(1) + \cdots + a_d x(d) = 0$. A hyperplane is a point in 1-D, a line in 2-D, a plane in 3-D as shown in Fig. 9.

Let us consider the family of discriminant functions that are linear combinations of the components of $x = [x_1, \ldots :; x_p]^T$,

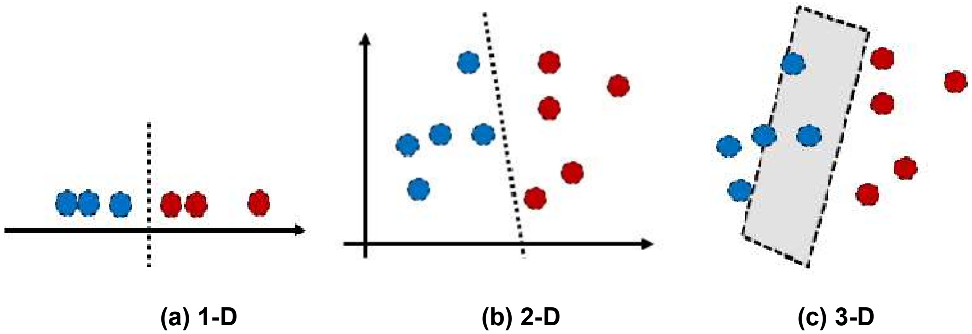


Fig. 9: Discriminant Function $g(x)$

You may see clearly that in Fig. 9(a) the discriminant function is simply a cut off, and in Fig. 9(b), the discriminant function is a line and in Fig. 9(c), the discriminant function is a plane.

$$g(x) = w^T x + w_0 = \sum_{i=1}^{p} w_i x_i + w_0$$

This is a linear discriminant function, a complete specification of which is achieved by prescribing the weight vector w and threshold weight w_0. Equation of $g(x)$ is the equation of a hyperplane with unit normal in the direction of w and a perpendicular distance $|w_0|/|w|$ from the origin. The value of the discriminant function for a pattern x is a measure of the perpendicular distance from the hyperplane shown in Fig. 10.

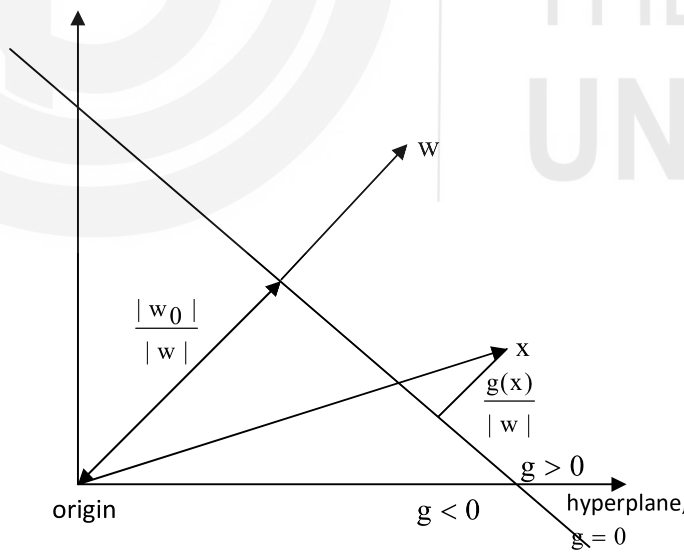


Fig. 10: Geometry of Linear Discriminant Function

Some of the properties of LDA are listed below:

- LDA assumes that the data follows a Gaussian distribution. More specifically, it assumes that all classes share the same covariance matrix.

- LDA finds linear decision boundaries in a $K-1$ dimensional subspace. As such, it is not suited if there are higher-order interactions between the independent variables.
- LDA is well-suited for multi-class problems but should be used with care when the class distribution is imbalanced because the priors are estimated from the observed counts. Thus, observations will rarely be classified to infrequent classes.
- Similarly to Principle Component Analysis (PCA), LDA can be used as a dimensionality reduction technique. Note that the transformation of LDA is inherently different to PCA because LDA is a supervised method that considers the outcomes.

The following are the situation, in which Linear Discriminant Analysis is useful.

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.
- If n is small and the distribution of the predictors X is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.
- Linear discriminant analysis is popular when there are more than two response classes.

Let us understand this through the following example.

Example 1: In order to select the best candidates, an over-subscribed secondary school sets an entrance exam on two subjects of English and Mathematics. The marks of 5 applicants as listed in the Table 1 below and the decision for acceptance is passing an average mark of 75.

(i) Show that the decision rule is equivalent of the method of linear discriminant function.

(ii) Plot the decision hyperplane, indicating the half planes of both Accept and Reject, and location of the 5 applicants.

Table 1

Candidate No.	English	Math	Decision
1	80	85	Accept
2	70	60	Reject
3	50	70	Reject
4	90	70	Accept
5	85	75	Accept

Solution: i) Denote marks of English and Math as x_1 and x_2, respectively. The decision rule is as follows:

If $x_1 + x_2 > 75,$ accept, otherwise reject.

This is equivalent to using a linear discriminant function. $g(x) = x_1 + x_2 - 150$ with the following decision rule:

If $g(x) > 0,$ accept, otherwise reject.

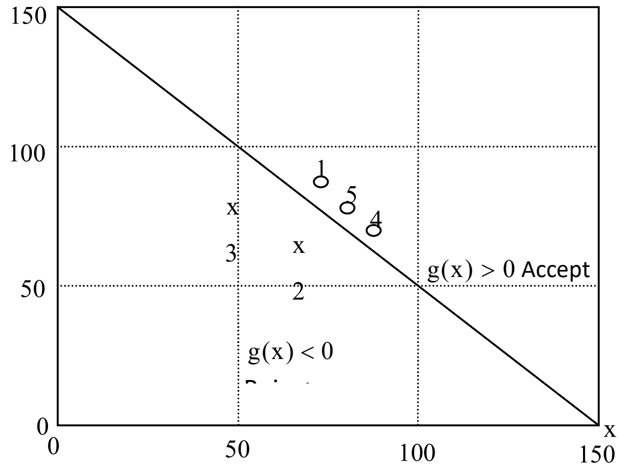


Fig. 11

(ii) To plot $g(x) = 0,$ the easiest way is to set $x_1 = 0,$ and find the value of x_2 so that $g(x) = 0.$

For this, $0 = 0 + x_2 - 150,$ so $x_2 = 150.$

$[0,150]^T$ is on the hyperplane.

Likewise we can also set $x_2 = 0,$ find the value of x_1 so that $g(x) = 0.$

i.e. $0 = x_1 + 0 - 150,$ so, $x_1 = 150.$ $[150, 0]^T$ is on the hyperplane.

Plot a straight line linking $[0,150]^T$ and $[150, 0]^T$ as shown in Fig. 11.

***

Next, we shall discuss another discriminant function.

Piecewise Linear discriminant Functions

Suppose we have 'm' classes, define m linear discriminant functions

$$g_i(x) = w_j^t x + w_{i0} \quad i = 1, \ldots, m$$

Given $x,$ assign class $c_i,$ if

$$g_i(x) \geq g_j(x) \qquad \forall j \neq i$$

Such classifier is called a linear machine that divides the feature space into c decision regions, with $g_i(x)$ being the largest discriminant if x is in the region $R_i.$

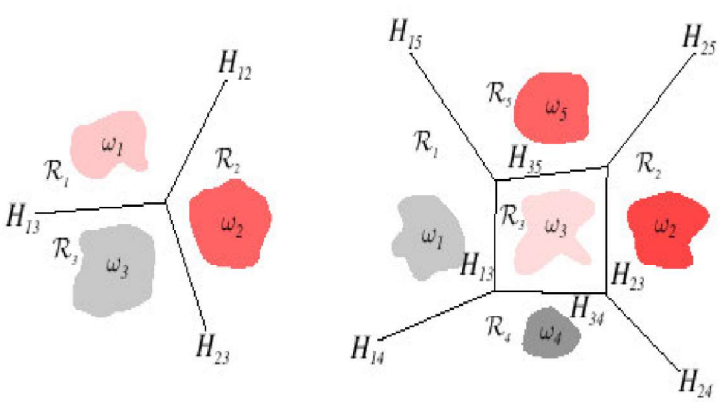


Fig. 12: Linear Discriminant Function for Multiple Classes

For a two contiguous regions R_i and R_j; the boundary that separates them is a portion of hyperplane H_{ij} defined by:

$$g_i(x) = g_j(x) \Leftrightarrow w_i^t x + w_{i0} = w_j^t x + w_{j0}$$
$$\Leftrightarrow (w_i - w_j)^t x + (w_{i0} - w_{j0}) = 0$$

Thus $w_i - w_j$ is normal to H_{ij}.. The distance from x to H_{ij} is given by

$$d(x, H_{ij}) = \frac{g_i(x) - g_j(x)}{\|w_i - w_j\|}.$$

Fig. 13: Groups not Separable by a Linear Discriminant

In a multi-class problem, a pattern x is assigned to the class for which the discriminant function has the maximum value. A linear discriminant function divides the feature space by a hyperplane whose orientation is determined by the weight vector w and distance from the origin by the weight threshold w_0.

Next discriminant function is quadratic discriminant function.

Quadratic Discriminant Function

A quadratic discriminant function is a mapping $g: X \to R$ with $g(x) = \frac{1}{2} x^T W x + w^T x + w_0$.

In quadratic discriminant function, the model parameter is $\theta = \{W; w, w_0\}$. Depending on W, the geometry of g could be convex, concave, or neither. Fig. 14 shows a quadratic discriminant function separating an inner and an outer cluster of data points.

A quadratic discriminant function is able to classify data using quadratic surfaces. This example shows an ellipsoid surface for separating an inner and outer cluster of data points. QDF is not really that much different from LDF except that you assume that the covariance matrix can be different for each class and so, we will estimate the covariance matrix separately for each class $k, k, = 1, 2, \ldots, K$.

$$\delta_k(x) = -\frac{1}{2}\log\left|\sum k\right| - \frac{1}{2}(x - \mu_k)^T \sum_k^{-1} (x - \mu_k) + \log \pi_k$$

Classification rule:

$$\hat{G}(x) = \arg\max_k \delta_k(x)$$

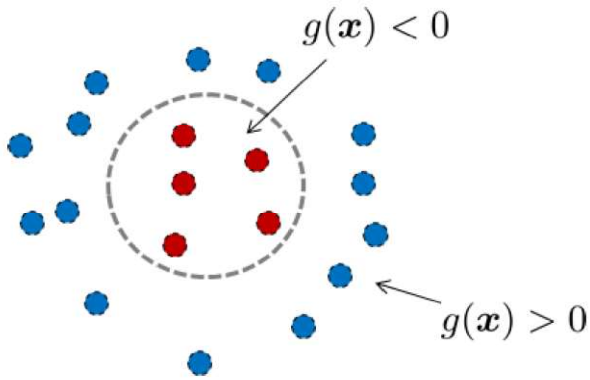


Fig. 14: Quadratic Discriminant Function

The classification rule is similar as well. You just find the class k which maximizes the quadratic discriminant function. The decision boundaries are quadratic equations in x. QDF allows more flexibility for the covariance matrix, tends to fit the data better than LDF, but then it has more parameters to estimate. The number of parameters increases significantly with QDF as a separate covariance matrix is required for every class. If you have many classes and not so many sample points, this can be a problem.

After quadratic discriminant function, let us now discuss the non-linear discriminant function.
Now, try the following exercises.

E4) Explain Linear Discremination Function.

E5) What are the properties of LDA?

In the following section, we shall discuss Bayesian classification.

12.4 BAYESIAN CLASSIFICATION

Goal of most classification procedures is to estimate the probabilities that a pattern to be classified belongs to various possible classes, based on the values of some feature or set of features.

Here, we are discussing Bayesian decision making or Bayes Classifier. This method refers to choosing the most likely class, given the value of the feature/s. Bayes theorem calculates the probability of class membership. In most cases, we decide which is the most likely class. We need a mathematical decision making algorithm, to obtain classification.

Let us recall Bayes theorem used in probability. Accordingly to it,

$P(w_i \mid \vec{X}) = \dfrac{P(\vec{X} \mid w_i) P(w_i)}{P(\vec{X})}$, where $P(w_i)$ is the prior probability for class $w_i, P(X)$ is the probability (Unconditional) for feature vector $X, P(w_i \mid X)$

is measured-conditioned or posteriori probability and $P(X \mid w_i)$ is the probability (Class-conditional) of feature vector X in class w_i.

$P(X)$ is the probability distribution for feature X in the entire population. It is also called unconditional density function. $P(w_i)$ is the prior probability that a random sample is a member of the class C_i. $P(X \mid w_i)$ is the class conditional probability (or likelihood) of obtaining feature value X given that the sample is from class w_i. It is equal to the number of times (occurrences) of X, if it belongs to class w_i.
The goal is to measure $P(w_i \mid X)$, posteriori probability, from the above three values. This is the probability of any vector X being assigned to class w_i.

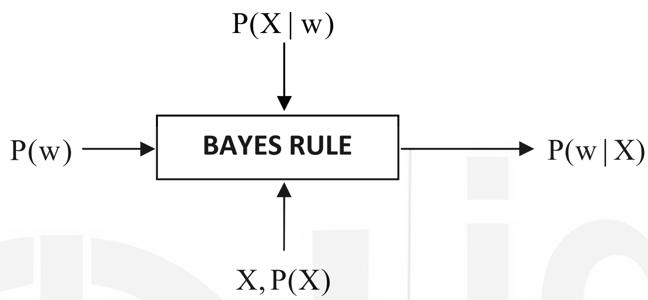


Fig. 15: Bayes Theorem

Here is another model, called the naive Bayes probabilistic model. The probability model for a classifier is a conditional model

$$P(X \mid W_1, \ldots, W_n)$$

over a dependent class variable X with a small number of outcomes or classes, conditional on several feature variables W_1 through W_n. The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes' theorem, we write

$$P(X \mid W_1, \ldots, W_n) = \frac{p(X)\, p(W_1, \ldots, W_n \mid X)}{p(W_1, \ldots, W_n)}$$

In simple words, $\text{posterior} = \dfrac{\text{prior x likelihood}}{\text{evidence}}$

In practice, we are only interested in the numerator of that fraction, since the denominator does not depend on X and the values of the features W_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model $p(X \mid W_1, \ldots, W_n)$, which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$\begin{aligned} p(X \mid W_1, \ldots, W_n) &= p(X)\, p(W_1, \ldots, W_n \mid X_n) \\ &= p(X)\, p(W_1 \mid X)\, p(W_2, \ldots, W_n \mid X, W_1) \end{aligned}$$

$$= p(X)\,p(W_1 \mid X)\,p(W_2 \mid X, W_1)\,p(W_3, \ldots, W_n \mid X, W_1, W_2)$$
$$= p(X)\,p(W_1 \mid X)\,p(W_2 \mid X, W_1), \ldots,$$
$$p(W_n \mid X, W_1, W_2, \ldots, W_{n-1})$$

Now the "naive" conditional independence assumptions come into play: assume that each feature W_i is conditionally independent of every other feature W_i for $i \neq j$. This means that

$$p(W_i \mid X, W_j) = p(W_i \mid X)$$

for $i \neq j$ and the joint expression can be expressed as

$$p(X \mid W_i, \ldots, W_n) = p(X)\,p(W_1 \mid X)\,p(W_2 \mid X)\ldots$$
$$= p(C)\prod_{i=1}^{n} p(w_i \mid X).$$

This means that under the above independence assumptions, the conditional distribution over the class variable can be expressed like this:

$$p(X \mid W_1, \ldots, W_n) = \frac{1}{z}\,p(C)\prod_{i=1}^{n} p(w_i \mid X),$$

where Z (the evidence) is a scaling factor dependent only on $W_i, \ldots W_n$, i.e., a constant, if the values of the feature variables are known.

Let us now discuss how Parameter estimation is done.

All model parameters (i.e., class priors and feature probability distributions) can be approximated with relative frequencies from the training set. These are maximum likelihood estimates of the probabilities. A class' prior may be calculated by assuming equiprobable classes (i.e., priors = 1 / (number of classes)), or by calculating an estimate for the class probability from the training set (i.e., (prior for a given class) = (number of samples in the class) / (total number of samples)).

To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set. If one is dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

For example, suppose the training data contains a continuous attribute, x. We first segment the data by the class and then compute the mean and variance of x in each class. Let μ_c be the mean of the values in x associated with class c, and let $\sigma_c{}^2$ be the variance of the values x in associated with class c. Then, the probability of some value given a class, $p(x = v \mid c)$, can be computed by plugging into the equation for a Normal distribution parameterized by μ_c and $\sigma_c{}^2$. That is,

$$p(x = v \mid c) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(v-\mu_c^2)^2}{2\sigma_c^2}}$$

Another common technique for handling continuous values is to use binning to discretize the values. In general, the distribution method is a better choice if there is a small amount of training data, or if the precise distribution of the data is known. The discretization method tends to do better if there is a large amount of training data because it will learn to fit the distribution of the data. Since naive Bayes is typically used when a large amount of data is available (as more computationally expensive models can generally achieve better accuracy), the discretization method is generally preferred over the distribution method.

If a given class and feature value never occurs together in the training set then the frequency-based probability estimate will be zero. This is problematic since it will wipe out all information in the other probabilities when they are multiplied. It is therefore often desirable to incorporate a small-sample correction in all probability estimates such that no probability is ever set to be exactly zero.

Now, we shall Construct a classifier from the probability model.
The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the *maximum a posteriori* or *MAP* decision rule. The corresponding classifier is the function classify defined as follows:

$$\text{classify}(W_1, \ldots, W_n) = \arg\ \max p(C = c) \prod_{i=1}^{n} p(W_i = w_i \mid C = c)$$

Despite the fact that the far-reaching independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution.

This in turn helps to alleviate problems stemming from the curse of dimensionality, such as the need for data sets that scale exponentially with the number of features. Like all probabilistic classifiers under the MAP decision rule, it arrives at the correct classification as long as the correct class is more probable than any other class; hence class probabilities do not have to be estimated very well. In other words, the overall classifier is robust enough to ignore serious deficiencies in its underlying naive probability model.

Properties of Bayes Classifiers

1. **Incrementality:** with each training example, the prior and the likelihood can be updated dynamically. It is flexible and robust to errors.

2. **Combines prior knowledge and observed data:** prior probability of a hypothesis is multiplied with probability of the hypothesis given the training data.
3. **Probabilistic hypotheses:** outputs is not only a classification, but a probability distribution over all classes.
4. **Meta-classification:** the outputs of several classifiers can be combined, e.g., by multiplying the probabilities that all classifiers predict for a given class.

Now, let us see the following examples.

Example 2: (Two class problem): Let us define variables, Cold (C) and not-cold (C'). Feature is fever (f). Prior probability of a person having a cold, $P(C) = 0.01$. Prob. of having a fever, given that a person has a cold is, $P(f \mid C) = 0.4$. Overall prob. of fever $P(f) = 0.02$. Find the Prob. that a person has a cold, given that she (or he) has a fever.

Solution: Using Bayes theorem, the Prob. that a person has a cold, given that she (or he) has a fever is:

$$P(C \mid f) = \frac{p(f \mid c)\,P(c)}{P(f)} = \frac{0.4 * 0.001}{0.02} = 0.2$$

let us take an example with values to verify:

Total Population =1000.

Thus, people hav ing cold = 10.

People having both fever and cold = 4.

Thus, people having only cold = 10 – 4 = 6.

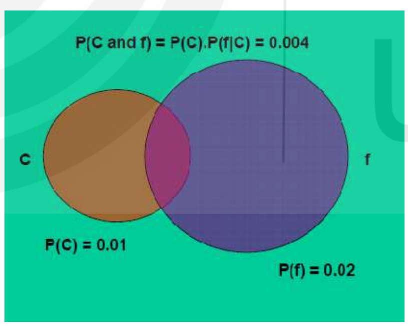


Fig. 15: A Venn Diagram

People having fever (with and without cold) = 0.02 * 1000 = 20.

People having fever without cold = 20 – 4 = 16

So, probability (percentage) of people having cold along with fever, out of all those having fever, is $= 4/20 = 0.2 (20\%)$.

Probability of a joint event - a sample comes from class C and has the feature value X:

$$\begin{aligned} P(C \text{ and } X) &= P(C).P(X \mid C) \\ &= 0.01 * 0.4 \end{aligned}$$

Or, $P(C \text{ and } X) = P(X).P(C \mid X)$
$$= 0.02 * 0.2$$

Also verify, for a K class problem:

$$P(X) = P(w_1)P(X \mid w_1) + P(w_2)P(X \mid w_2) + \cdots + P(w_k)P(X \mid w_k)$$

$$P(w_i \mid \vec{X}) = \frac{P(\vec{X} \mid w_i)P(w_i)}{P(w_1)P(X \mid w_1) + P(w_2)P(X \mid w_2) + \cdots + P(w_k)P(X \mid w_k)}$$

We get

$$P(f) = P(C)P(f \mid C) + P(C')P(f \mid C')$$
$$= 0.01 * 0.4 + 0.99 * 0.01616 = 0.02$$

***

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.
An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Try following exercises.

E6) Explain Bayes classifier.

E7) Explain properties of Bayes classifier.

12.5 MINIMUM DISTANCE CLASSIFIERS

Minimum distance classifier is a pattern classification scheme defined by distance functions. Here pixels which are close to each other in feature space are likely to belong to the same class. The measure of similarity is the "distance" between pixels in feature space (n-D histogram). All dimensions should be in comparable units and distance may be scaled in pixels, radiance, reflectance etc. the classification is most effective if the clusters are disjoint. It requires the least amount of prior information to operate.

Distance in feature space is the primary measure of similarity in minimum distance classifier algorithms.Pixels that are "close" in feature space willbe grouped in the same class.The relative distances may change when data are calibrated, atmospherically corrected or rescaled in ways that treat different features differently. If two features have different units, theymust be scaled to provide comparable variance. Otherwise the "distance" will be biased.

Distance in feature space is the primary measure of similarity in all clustering algorithms.

1) Euclidean distance:

$$d_i(x) = -D_i(x) = -[(x - z_i)^T (x - z_i)]^{1/2}$$

2) Square of the Euclidean distance:

$$d_i^2(x) = -D_i^2(x) = -[(x - z_i)^T (x - z_i)]$$

3) Square of Euclidean distance after eliminating components that are independent of class:

$$d_i(x) = -[x^T z_i + 1/2(z_i^T z_i)]$$

4) Taxicab distance

$$d_i(x) = -D_i(x) = -\sum_{k=1}^{N_b} | X_k - Z_{ki} |$$

The **decision boundary** for the single prototype, simple distance discriminant function is the set of planar surfaces perpendicular to and bisecting the lines connecting pairs of prototypes as shown in Fig. 1. This is a minimum-distance classifier. If the prototype is the mean value of the training pixels for each class, it is called a minimum-distance-to-mean classifier.

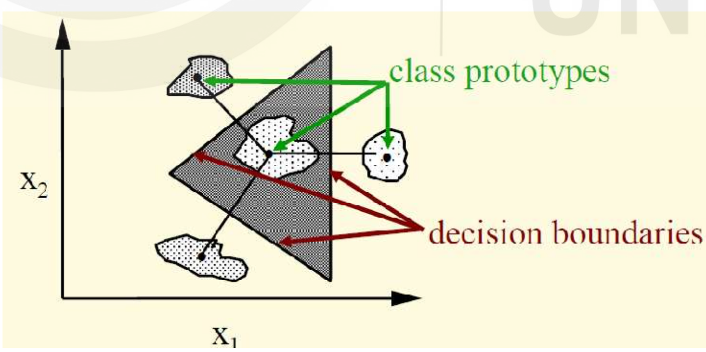


Fig. 1: Decision boundary

The results of clustering will depend strongly on the choice of the prototype. Alternatives for prototype selection are

1. Let the user select prototypes, i.e., one "example" pixel per class.(Reduces the utility of a clustering procedure.)
2. Devise an unbiased procedure for selecting prototypes (random selection, selection at vertices of an arbitrary grid etc)

3. Use the user-selected prototype or unbiased selection procedure as the starting point of an optimization procedure.

We shall discuss Euclidean distance classifier and Mahalanobis distance classifiers here.

The Euclidean Distance Classifier

The optimal Bayesian classifier is significantly simplified under the following assumptions:

- The classes are equiprobable.
- The data in all classes follow Gaussian distributions.
- The covariance matrix is the same for all classes.
- The covariance matrix is diagonal and all elements across the diagonal are equal. That is, $S = \sigma^2 I$, where I is the identity matrix.

Under these assumptions, it turns out that the optimal Bayesian classifier is equivalent to the minimum Euclidean distance classifier. Each class (pattern) is represented by a single prototype vector, z. Assume that there are 'm' classes and that these classes are represented by the prototype vectors, $z_1, z_2, z_3, \dots, z_m$.

The **Euclidean distance**, $D_i(x)$, of a measurement vector, x, from the prototype vector, z_i:

$$D_i(x) = |x - z_i| = [(x - z_i)^T (x - z_i)]^{1/2}$$

The discriminant function is usually defined as the negative of the separation distance:

$$d_i(x) = -D_i(x)$$

The larger (less negative) $d_i(x)$, the closer the measurement vector lies relative to the prototype vector z_i. The maximum value of $d_i(x)$ is zero and occurs when x matches the prototype vector exactly.

Algorithm

Step 1: Select a threshold, T. T is a representative distance in measurement space. The choice of T in this algorithm is entirely arbitrary; it is also the only input required of the user.

Step 2: Select a pixel with measurement vector, x. The selection scheme is arbitrary. Pixels could be selected at random.

Step 3: Let the first pixel be taken as the first cluster center, z_1.

Step 4: Select the next pixel from the image.

Step 5: Compute the distance functions, $D_i(x)$.Compute the distance function for each of the classes established at this point, i.e., compute $D_i(x)$, for $i = 1,\dots,N$ where $N =$ the number of classes. ($N = 1$ initially.)

Step 6: Compare the $D_i(x)$ with T.

a) if $D_i(x) < T$, then $x \in w_i$.

b) if $D_i(x) < T$, for all i, then let x become a new prototype vector: Assign $x \to z_{N+1}$. (Do not compute D_{N+1} for pixels already assigned to an existing class.)

Step 7: Return to step #4 until all pixels are assigned to a class.

Step 8: After all pixels have been assigned to a cluster center, re-compute the $D_i(x)$ and reassign pixels according to the minimum $D_i(x)$.

This simple clustering algorithm is extremely sensitive to the value of the threshold, T, and the order in which pixels are selected. For a given T, two different selection patterns can yield very different results. For the same selection pattern, a different value of T T will lead to different results. These flaws are typical of clustering algorithms. All are sensitive to the starting selection of cluster centers and to the particular specification of the clustering criterion. The better algorithms handle the problems cleverly and without the severe problems that would be apparent with the above algorithm.

That is, given an unknown x, assign it to class ω_i if

$$\|x - m_i\| \equiv \sqrt{(x - m_i)^T (x - m_i)} < \|x - m_j\|, \quad \forall_i \neq j$$

It must be stated that the Euclidean classifier is often used, even if we know that the previously stated assumptions are not valid, because of its simplicity. It assigns a pattern to the class whose meanis closest to it with respect to the Euclidean norm.

The Mahalanobis Distance Classifier

In many appliacations, the range of all feature value may differ widely. One could be in hundreds while the other could be in decimal fractions. If this issue is overlooked some feature values will get neglected. If one relaxes the assumptions required by the Euclidean classifier and removes the last one, the one requiring the covariance matrix to be diagonal and with equal elements, the optimal Bayesian classifier takes the form of minimum Mahalanobis distance classifier. That is, given an unknown x, it is assigned to class ω_i if

$$\sqrt{(x - m_i)^T S^{-1} (x = m_i)} < \sqrt{(x - m_j)^T S^{-1} (x - m_j)}, \quad \forall_j \neq i,$$

where S is the common covariance matrix. The presence of the covariance matrix accounts for the shape of the Gaussians distributions of various features.

Try the following exercises.

E8) What distance measure is used by Euclidean Distance Classifier?

E9) What is the discriminant function for Euclidean distance classifier?

In the following section, we shall discuss Machine learning algorithm.

12.6 MACHINE LEARNING ALGORITHMS

In order to understand how Machine learning works, we need some essential ML vocabulary first:

- ***Parameter***: A value that is learnt during the training of a machine learning program. Based on existing parameters, the training algorithm "trains" and tries to continuously improve the parameters. The inference algorithm uses the parameters to calculate results.
- ***Model***: A trained machine learning method. A model is essentially a (sometimes very large) set of parameters and instructions on how to use them – a ready-to-run ML procedure and the result of the training process. The training algorithm generates the model, the inference algorithm uses it to perform a task.
- ***Inference***: Running a model using a second algorithm. This algorithm uses the model to create a prediction based on an input, classify an input or create some interesting value based on an input. Hence the term "inference algorithm" or "prediction algorithm". Inference is a technical term for "execute the ML model and output a result".

As we see, Machine learning use the combination of a training algorithm and a prediction (or inference) algorithm. The training algorithm uses data to gradually determine parameters. The set of all learned parameters is called a model, basically a "set of rules" established by the algorithm, applicable even to unknown data. The inference algorithm then uses the model and applies it to any given data. Finally, it delivers the desired results.

Equipped with the right vocabulary, we can take a closer look at the execution of a machine learning project:

- We select the machine learning method for which we want to train a model. The choice will depend on the problem to be solved, the available data, the experience and also on gut feeling.
- Then we divide the available data into two parts: The training data and the test data. We apply our training data and thus obtain our model. The model is checked on the unknown test data. It is most important that the test data aren't used during the training phase under any circumstances. The reason is obvious: Computers are great at learning by heart. Complex models like neural networks can

actually start to memorize by themselves. The following results might be quite remarkable. There's only one flaw: They're not based on a model formulated by the program, but on "memorized" data. This effect is called "over fitting". However, the test data are supposed to simulate the "unknown" during quality control and to check whether the model has really "learned" something. A good model achieves about the same error rate on the test data as on the training data without ever having seen it before.

- We use the training data to develop the model with the learning algorithm. The more data we have, the "stronger" the model becomes. Using up all available data for the training algorithm is called an "epoch".
- In order to test it, the trained model is used on the test data unknown to it and makes predictions. If we did everything right, the predictions on unknown data should be as good as on the training data – the model can generalize and solve the problem. Now it is ready for practical use.
- At its most basic, machine learning uses programmed algorithms that receive and analyse input data to predict output values within an acceptable range. As new data is fed to these algorithms, they learn and optimise their operations to improve performance, developing 'intelligence' over time.

Machine learning algorithms are organized into taxonomy, based on the desired outcome of the algorithm. Common algorithm type includes:

- **Supervised learning** --- the algorithm generates a function that maps inputs to desired outputs. One standard formulation of the supervised learning task is the classification problem: the learner is required to learn (to approximate the behaviour of) a function which maps a vector into one of several classes by looking at several input-output examples of the function.

The performance and computational analysis of machine learning algorithms is a branch of statistics known as computational learning theory. Machine learning is about designing algorithms that allow a computer to learn. Learning is not necessarily involves consciousness but learning is a matter of finding statistical regularities or other patterns in the data. Thus, many machine learning algorithms will barely resemble how human might approach a learning task. However, learning algorithms can give insight into the relative difficulty of learning in different environments.

Now we shall discuss supervised learning approach in detail in the following section.

12.7 SUPERVISED LEARNING APPROACH

Supervised learning is fairly common in classification problems because the goal is often to get the computer to learn a classification system that we have created. Digit recognition, once again, is a common example of classification learning. More generally, classification learning is appropriate for any problem where deducing a classification is useful and the classification is easy to determine. In some cases, it might not even be necessary to give predetermined classifications to every instance of a problem if the agent can work out the classifications for itself. This would be an example of unsupervised learning in a classification context.

Supervised learning often leaves the probability for inputs undefined. This model is not needed as long as the inputs are available, but if some of the input values are missing, it is not possible to infer anything about the outputs. Unsupervised learning, all the observations are assumed to be caused by latent variables, that is, the observations is assumed to be at the end of the causal chain. Examples of supervised learning and unsupervised learning are shown in the Fig. 2.

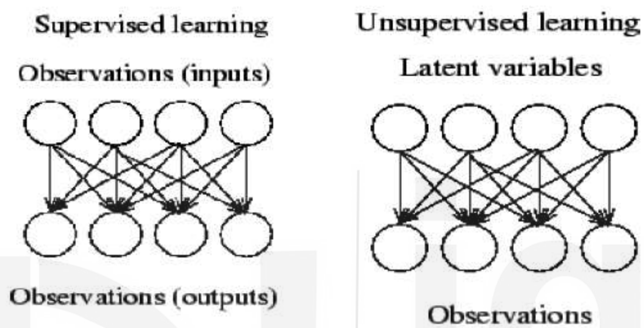


Fig. 2: Examples of Supervised and Unsupervised Learning

Supervised learning is the most common technique for training neural networks and decision trees. Both of these techniques are highly dependent on the information given by the pre-determined classifications. In the case of neural networks, the classification is used to determine the error of the network and then adjust the network to minimize it, and in decision trees, the classifications are used to determine what attributes provide the most information that can be used to solve the classification puzzle. Both of these examples have some "supervision" in the form of pre-determined classifications.

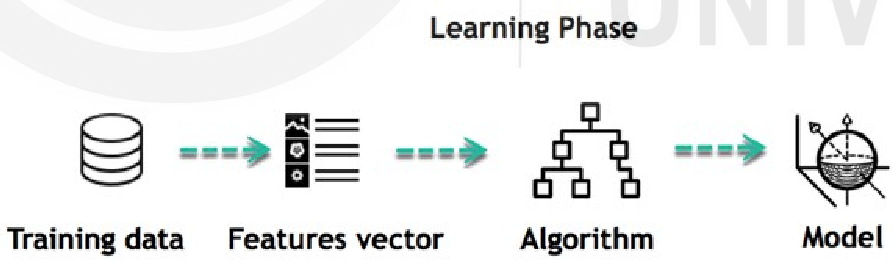


Fig. 3: Learning phase of a supervised learning algorithm

Inductive machine learning is the process of learning a set of rules from instances (examples in a training set), or more generally speaking, creating a classifier that can be used to generalize from new instances. The process of applying supervised ML to a real world problem is described below

Step 1 **(Collect the dataset):** If a requisite expert is available, then s/he could suggest which fields (attributes, features) are the most informative. If not, then the simplest method is that of measuring everything available in the hope that the right (informative, relevant) features can be isolated.

Step 2 (Data preparation and data pre-processing): Depending on the circumstances, there are a number of methods to choose from to handle missing data, outlier (noise) detection. There is a variety of procedures for sampling instances from a large dataset. Feature subset selection is the process of identifying and removing as many irrelevant and redundant features as possible. This reduces the dimensionality of the data and enables data mining algorithms to operate faster and more effectively.

Step 3 (Define a training set): The goal of the learning algorithm is to minimize the error with respect to the given inputs. These inputs, often called the "training set", are the examples from which the agent tries to learn. But learning the training set well is not necessarily the best thing to do. For instance, if I tried to teach you exclusive-or, but only showed you combinations consisting of one true and one false, but never both false or both true, you might learn the rule that the answer is always true. Similarly, with machine learning algorithms, a common problem is over-fitting the data and essentially memorizing the training set rather than learning a more general classification technique. As you might imagine, not all training sets have the inputs classified correctly. This can lead to problems if the algorithm used is powerful enough to memorize even the apparently "special cases" that don't fit the more general principles. This, too, can lead to over fitting, and it is a challenge to find algorithms that are both powerful enough to learn complex functions and robust enough to produce generalizable results.

Step 4: Then we need to select a suitable algorithm and perform training and evaluate the results.

Step 5: If we are satisfied with the results, classifier is trained else parameter tuning is done which includes updating of data preprocessing and training set.

Supervised Machine Learning Categorization

It is important to remember that all supervised learning algorithms are essentially complex algorithms, categorized as either classification or regression models as shown in Fig. 4.

1) **Classification Models** — Classification models are used for problems where the output variable can be categorized, such as “Yes” or “No”, or “Pass” or “Fail.” Classification Models are used to predict the category of the data. Real-life examples include spam detection, sentiment analysis, scorecard prediction of exams, etc.

2) **Regression Models** — Regression models are used for problems where the output variable is a real value such as a unique number, dollars, salary, weight or pressure, for example. It is most often used to predict numerical values based on previous data observations. Some of the more familiar regression algorithms

include linear regression, logistic regression, polynomial regression, and ridge regression.

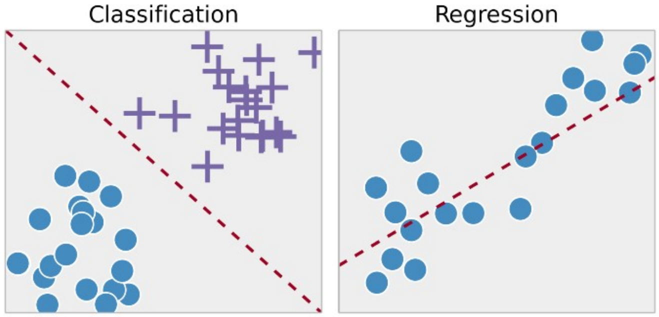


Fig. 4: Categories of Supervised Machine Learning algorithms

There are some very practical applications of supervised learning algorithms in real life, including:

- Text categorization
- Face Detection
- Signature recognition
- Customer discovery
- Spam detection
- Weather forecasting
- Predicting housing prices based on the prevailing market price
- Stock price predictions, among others

Try the following exercises.

E10) What are the different categories of supervised machine learning algorithms?

E11) What are different applications of supervised machine learning algorithms?

Let us list the Differences between Supervised and Unsupervised Learning Approaches.

Parameters	**Supervised machine learning technique**	**Unsupervised machine learning technique**
Process	In a supervised learning model, input and output variables will be given.	In unsupervised learning model, only input data will be given
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data which is not labeled
Algorithms Used	Support vector machine, Neural network, Linear and logistics regression,	Unsupervised algorithms can be divided into different categories: like

	random forest, and Classification trees.	Cluster algorithms, K-means, Hierarchical clustering, etc.
Computational Complexity	Supervised learning is a simpler method.	Unsupervised learning is computationally complex
Use of Data	Supervised learning model uses training data to learn a link between the input and the outputs.	Unsupervised learning does not use output data.
Accuracy of Results	Highly accurate and trustworthy method.	Less accurate and trustworthy method.
Real Time Learning	Learning method takes place offline.	Learning method takes place in real time.
Number of Classes	Number of classes is known.	Number of classes is not known.
Main Drawback	Classifying big data can be a real challenge in Supervised Learning.	You cannot get precise information regarding data sorting, and the output as data used in unsupervised learning is labeled and not known.

The following are the various situations, where supervised or unsupervised learning approaches are used.

- In Supervised learning, the machine is trained using data which is well "labeled."
- Unsupervised learning is a machine learning technique, where you do not need to supervise the model.
- Supervised learning allows you to collect data or produce a data output from the previous experience.
- Unsupervised machine learning helps you to finds all kind of unknown patterns in data.
- For example, you will able to determine the time taken to reach back come base on weather condition, Times of the day and holiday.
- For example, Baby can identify other dogs based on past supervised learning.
- Regression and Classification are two types of supervised machine learning techniques.
- Clustering and Association are two types of Unsupervised learning.
- In a supervised learning model, input and output variables will be given while with unsupervised learning model, only input data will be given

Now the question arises, When to Choose Supervised Learning / Unsupervised Learning?

In manufacturing, a large number of factors affect which machine learning approach is best for any given task. And, since every machine learning problem is different, deciding on which technique to use is a complex process.

In general, a good strategy for honing in on the right machine learning approach is to:

- **Evaluate the data.** Is it labeled/unlabelled? Is there available expert knowledge to support additional labeling? This will help to determine whether a supervised, unsupervised, semi-supervised or reinforced learning approach should be used
- **Define the goal.** Is the problem recurring, defined one? Or, will the algorithm be expected to predict new problems?
- **Review available algorithms** that may suit the problem with regards to dimensionality (number of features, attributes or characteristics). Candidate algorithms should be suited to the overall volume of data and its structure
- **Study successful applications** of the algorithm type on similar problems

Let us now summarize, what we have discussed in this unit.

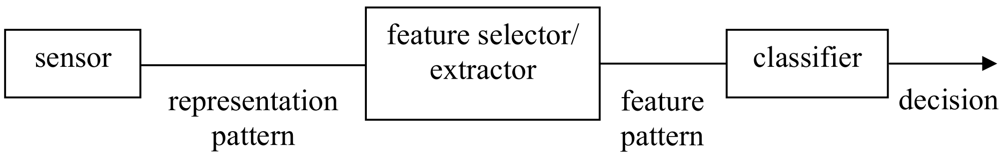
12.8 SUMMARY

We have discussed the following points.

1) Concept of Classification.
2) Various terminologies in pattern recognition problem.
3) Linear discriminant function, piecewise discriminant function, quadratic discrimint function and non-linear discriminant function.
4) Bayes classifier combines prior knowledge with observed data: assigns a posterior probability to a class based on its prior probability and its likelihood given the training data. It computes the maximum a posterior (MAP) hypothesis or the maximum likelihood (ML) hypothesis.
5) Naive Bayes classifier assumes conditional independence between attributes and assigns the MAP class to new instances.
6) Likelihoods can be estimated based on frequencies. Sparse data poses a huge problem. A possible solution is to use m -estimate.
7) Concept of minimum distance classifier.
8) Supervised Learning.
9) Unsupervised Learning.
10) Differences between Supervised Learning/ Unsupervised Learning.

12.9 SOLUTION/ANSWERS

E1)

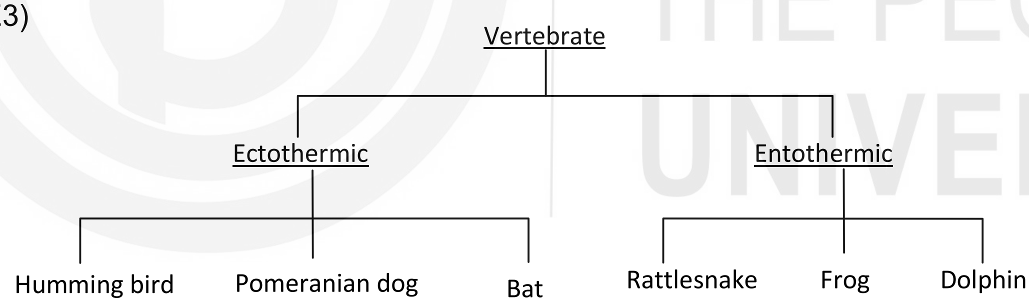


Optical sensing is used to distinguish two patterns. A camera takes pictures of the object and passes to on to a feature extractor. The feature extractor reduces the data by measuring certain “properties” that distinguish pictures of one object to the other. These features are then passed to a classifier that evaluates the evidence presented and makes a final decision about the object type.

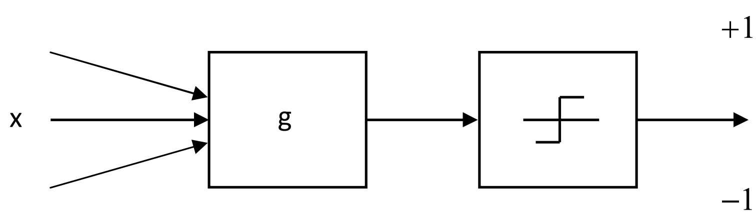
One characteristic of human pattern recognition is that it involves a teacher. Similarly a machine pattern recognition system needs to be trained. A common mode of learning is to be given a collection of labeled examples, known as training data set. From the training data set, structure information is distilled and used for classifying new inputs.

E2) The samples of input (when represented by their features) are represented as points in the feature space. If a single feature is used, then work on a one- dimensional feature space. If number of features is 2, then we get points in 2D space. We can also have an n-dimensional feature space.

E3)



E4)



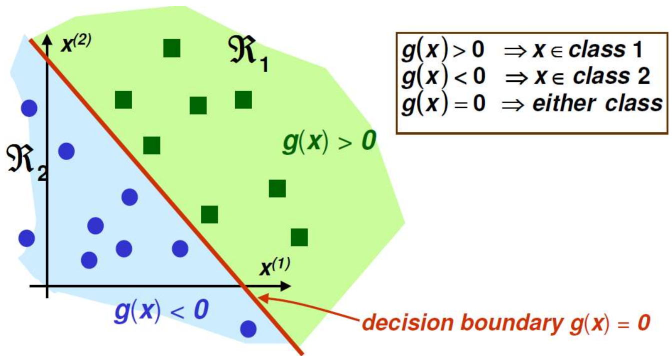
For a new sample x and a given discriminant function, we can decide on x belongs to

Class1 if $g(x) > 0,$ otherwise it’s Class 2.

A discriminant function that is a linear combination of the components of x can be written as

$g(x) = w^T x + w_0$

where w is called the weight vector and w_0 the threshold weight. These are the parameters that we want to estimate based on training data. A classifier based entirely on linear discriminant functions is called a linear classifier or a linear machine.



E5) LDF assumes that the data are Gaussian. More specifically, it assumes that all classes share the same covariance matrix.

- LDF finds linear decision boundaries in a $K-1$ dimensional subspace. As such, it is not suited if there are higher-order interactions between the independent variables.
- LDF is well-suited for multi-class problems but should be used with care when the class distribution is imbalanced because the priors are estimated from the observed counts. Thus, observations will rarely be classified to infrequent classes.
- Similarly to PCA, LDA can be used as a dimensionality reduction technique. Note that the transformation of LDA is inherently different to PCA because LDA is a supervised method that considers the outcomes.

E6) The naive Bayes classifier combines bayes model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the *maximum a posteriori* or *MAP* decision rule. The corresponding classifier is the function classify defined as follows:

$$\text{classify}(W_1, \ldots, W_n) = \arg\ \max p(C = c) \prod_{i=1}^{n} p(W_i = w_i \mid C = c)$$

Despite the fact that the far-reaching independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution.

E7)

- **Incrementality:** with each training example, the prior and the likelihood can be updated dynamically. It is flexible and robust to errors.

- **Combines prior knowledge and observed data:** prior probability of a hypothesis is multiplied with probability of the hypothesis given the training data.
- **Probabilistic hypotheses:** outputs is not only a classification, but a probability distribution over all classes.
- **Meta-classification:** the outputs of several classifiers can be combined, e.g., by multiplying the probabilities that all classifiers predict for a given class.

E8) The **Euclidean distance**, $D_i(x)$, of a measurement vector, x, from the prototype vector, Z_i:

$$D_i(x) = -|(x - z_i| = [(x - z_i)^T (x - z_i)]^{1/2}$$

E9) The discriminant function is usually defined as the negative of the separation distance:

$$d_i(x) = -D_i(x)$$

E10) Supervised learning algorithms are essentially complex algorithms, categorized as either classification or regression models

1) **Classification Models** — Classification models are used for problems where the output variable can be categorized, such as “Yes” or “No”, or “Pass” or “Fail.” Classification Models are used to predict the category of the data. Real-life examples include spam detection, sentiment analysis, scorecard prediction of exams, etc.
2) **Regression Models** — Regression models are used for problems where the output variable is a real value such as a unique number, dollars, salary, weight or pressure, for example. It is most often used to predict numerical values based on previous data observations. Some of the more familiar regression algorithms include linear regression, logistic regression, polynomial regression, and ridge regression.

E11) Applications of supervised learning algorithms in real life, including:

- Text categorization
- Face Detection
- Signature recognition
- Customer discovery
- Spam detection
- Weather forecasting
- Predicting housing prices based on the prevailing market price
- Stock price predictions, among others.

UNIT 13

OBJECT CLASSIFICATION USING UNSUPERVISED LEARNING

Structure **Page No.**

13.1 INTRODUCTION

Clustering is a technique for identifying similarity groups in data, called clusters. i.e., it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters. Clustering is often called an unsupervised learning task as no class values denoting an a priori grouping of the data instances are specified beforehand, as is done in supervised learning. Due to historical reasons, clustering is often considered synonymous with unsupervised learning.

And now, we will list the objectives of this unit. After going through the unit, please read this list again make sure you have achieved the objectives.

Objectives

After studying this unit, you should be able to

- define clustering
- define and use different clustering techniques

- apply Hierarchial Clustering
- use partition based clustering,
- apply K - NN clustering

13.2 INTRODUCTION TO CLUSTERING

The problem of pattern clustering may be regarded as one of discriminating the input data, not between individual patterns, but between populations. These populations are searched for a match with the new object with the help of its features. The objective is to categorize patterns into classes so that patterns belonging to different classes are well separated. The process may start with or without any knowledge of the feature space. When we have prior knowledge about the class of a subset of data, it is called **supervised classification or classification**. When nothing is known a prior, the scheme is called **unsupervised classification or clustering**. In a case of supervised classification, the labelled subset of data is called **training data**.
Being unsupervised in nature, clustering is a very difficult task, as the same data may reveal many different inherent structures depending on the shape and size of its distribution.

A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A cluster is therefore a collection of objects which are “similar to each other and are “dissimilar” to the objects belonging to other clusters. Cluster analysis is also used to form descriptive statistics to ascertain whether or not the data consists of a set distinct subgroups, each group representing objects with substantially different properties. The latter goal requires an assessment of the degree of difference between the objects assigned to the respective clusters.

The notion of cluster is not well defined. To better understand the difficulty of deciding what constitutes a cluster, consider Fig. 1, which shows twenty points and three different ways of dividing them into clusters. The shapes of the markers indicate cluster membership. Fig. 1(b) and Fig.1 (d) divide the data into two and six parts, respectively. The apparent division of each of the two larger clusters into 3 sub-clusters may simply be an artifact of human visual system. In Fig. 1(c), there are four clusters. The fig illustrates that the definition of a cluster is imprecise and that the best definition depends on the nature of data and the desired result.

(a) Original Point **(b) Two Clusters**

(c) Four Clusters **(d) Six Clusters**

Fig. 1: Different ways of Clustering Same Data Points

The various examples of clustering applications are as follows:

1) **Marketing**: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs.
2) **Land use**: Identification of areas of similar land use in an earth observation database.
3) **Insurance**: Identifying groups of motor insurance policy holders with a high average claim cost.
4) **City-planning**: Identifying groups of houses according to their house type, value, and geographical location.
5) **Earth-quake studies**: Observed earth quake epicenters should be clustered along continent faults.
6) **Image processing**: Clustering parts of image having similar RGB values, so that image is clustered into regions such as sky, greenery, road, house, etc.

In fact, clustering is one of the most utilized data mining techniques. It has a long history, and used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc. In recent years, due to the rapid increase of online documents, text clustering becomes important.

Let us see some real-life examples of clustering.

Example 1: Groups people of similar size to gather to make “small”, “medium” and “large” T-Shirts.

– Tailor-made for each person: too expensive
– One-size-fits-all: does not fit all.

***

Example 2: In marketing, segment customers according to their similarities

– To do targeted marketing.

***

Example 3: Given a collection of text documents, we want to organize them according to their content similarities,

– To produce a topic hierarchy

***

See, how quality of clustering is decided?

A **good clustering** method will produce high quality clusters with

a) high intra-class similarity (in the same class)
b) low inter-class similarity (between two classes)

The quality of a clustering result depends on both the similarity measure used by the method and its implementation. The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

We can measure the quality of clustering by dissimilarity/similarity metric. Similarity is expressed in terms of a distance function, which is typically metric: $d(i, j)$. There is a separate "quality" function that measures the "goodness" of a cluster. The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, and ordinal variables. Weights should be associated with different variables based on applications and data semantics. It is hard to define "similar enough" or "good enough". The answer is typically highly subjective.

Let us define a cluster in the following definition.

Clusters can be defined as collection of similar object groups together. A cluster is a set of entities which are alike and at the same time entities from different cluster are not alike.

In general Clusters may be defined as collection of points in a test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point outside the cluster.

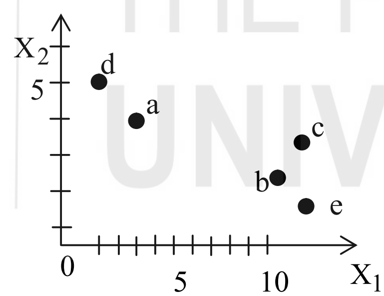
In general, similarity and dissimilarity between data points is measured as a function of the distance between them. The objects may also be grouped into clusters based on different shapes and sizes.

Cluster analysis embraces a variety of techniques, the main objective of which is to group observations or variables into homogeneous and distinct clusters. A simple numerical example will help explain these objectives.

The daily expenditures on food (X1) and clothing (X2) of five persons are shown in Fig. 2.

Person	X_1	X_2
a	2	4
b	8	2
c	9	3
d	1	5
e	$8 \cdot 5$	1

(a) Illustrative Data



(b) Grouping of Observations

Fig. 2

Fig. 2 suggests that the five observations form two clusters. The first consists of persons a and d, and the second of b, c and e. It can be noted that the observations in each cluster are similar to one another with respect to expenditures on food and clothing, and that the two clusters are quite distinct from each other.

These conclusions concerning the number of clusters and their membership were reached through a visual inspection of Fig. 2. This inspection was possible because only two variables were involved in grouping the observations. The question is: Can a procedure be devised for similarly grouping observations when there are more than two variables or attributes?

It may appear that a straightforward procedure is to examine all possible clusters of the available observations, and to summarize each clustering according to the degree of proximity among the cluster elements and of the separation among the clusters. Unfortunately, this is not feasible because in most cases in practice the number of all possible clusters is very large and out of reach of current computers. Cluster analysis offers a number of methods that operate much as a person would in attempting to reach systematically a reasonable grouping of observations or variables.

Since clustering is the grouping of similar instances/objects, some sort of measure that can determine whether two objects are similar or dissimilar is required. There are two main type of measures used to estimate this relation: distance measures and similarity measures. Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects.

Depending on which formula is used to compute the distance between two data points can lead to different classification results. Domain knowledge must be used to guide the formulation of a suitable distance measure for each particular application. For high dimensional data, a popular measure is the Minkowski Metric:

$$d(x_i, x_j) = \left(\sum_{k=1}^{d} | x_{i,k} - x_{j,k} |^p \right)^{\frac{1}{p}},$$

where d is the dimensionality of the data.

Special Cases:

If $p = 2$, then the distance is Euclidean distance, and if $p = 1$, then the distance is Manhattan distance.

The commonly used Euclidean distance between two objects is achieved when $p = 2$.

$$d_{ij} = ((x_{i1} - x_{j1})^2 + (x_{i2} - x_{j_2})^2 + \cdots + (x_{id} - x_{jd})^2)^{\frac{1}{2}}.$$

Another well-known measure is the Manhattan distance which is defined when $p = 1$.

$$d_{ij} = | ((x_{i1} - x_{j1}) + (x_{i2} - x_{j2}) + \cdots + (x_{id} - x_{jd})) |.$$

The Mahalanobis distance is another very important distance measure used in statistics that measures the statistical distance between two populations of Gaussian mixtures having mean μ_i and μ_j and a common covariance matrix $\sum_{ij}$. This measure is given by

$$d_{ij} = (\mu_i - \mu_j)^T \sum_{ij} (\mu_i - \mu_j).$$

The distance measure described in the last section may be easily computed for continuous-valued attributes. In the case of instances described by categorical, binary, ordinal or mixed type attributes, the distance measure should be revised.

In the case of binary attributes, the distance between objects may be calculated based on a contingency table. A binary attribute is symmetric if both of its states are equally valuable. In that case, using the simple matching coefficient can assess dissimilarity between two objects:

$$d(x_i, x_j) = \frac{r+s}{q+r+s+t}$$

where q is the number of attributes that equal 1 for both objects; t is the number of attributes that equal 0 for both objects; and s and r are the number of attributes that are unequal for both objects.

A binary attribute is asymmetric, if its states are not equally important (usually the positive outcome is considered more important). In this case, the denominator ignores the unimportant negative matches (t). This is called the Jaccard coefficient:

$$d(x_i, x_j) = \frac{r+s}{q+r+s}.$$

When the attributes are nominal, two main approaches may be used:

i) Simple Matching: $d(x_i, x_j) = \frac{p-m}{p}$, where, p is the total number of attributes and m is the number of matches.

ii) Creating a binary attribute for each state of each nominal attribute and computing their dissimilarity as described above.

Try the following exercise.

E1) What are different distance measures used for clustering?

Now, we shall discuss major clustering applications in the following section.

13.3 MAJOR CLUSTERING APPROACHES

We shall begin this section by describing the major clustering approaches.

1) **Partitioning algorithms:** Construct various partitions and then evaluate them by some criterion.
2) **Hierarchy algorithms:** Create a hierarchical decomposition of the set of data (or objects) using some criterion .
3) **Density-based:** Density based clustering approach is based on connectivity and density functions.
4) **Grid-based:** It is based on a multiple-level granularity structure.
5) **Model-based:** A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other.
6) **Sample based**
7) **High-dimensional algorithm**
8) **Constraint based**

Clustering algorithms may be classified as listed below:

• ***Exclusive Clustering***

In exclusive clustering data are grouped in an exclusive way, so that a certain datum belongs to only one definite cluster. K-means clustering is one example of the exclusive clustering algorithms.

• ***Overlapping Clustering***

The overlapping clustering uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership.

• ***Hierarchical Clustering***

Hierarchical clustering algorithm has two versions: agglomerative clustering and divisive clustering.

• ***Agglomerative clustering*** It is based on the union between the two nearest clusters. The beginning condition is realized by setting every datum as a cluster. After a few iterations it reaches the final clusters wanted. Basically, this is a bottom-up version

• ***Divisive clustering*** It starts from one cluster containing all data items. At each step, clusters are successively split into smaller clusters according to some dissimilarity. Basically this is a top-down version.

• ***Probabilistic Clustering***

Probabilistic clustering, e.g. mixture of Gaussian, uses a completely probabilistic approach.

The following requirements should be satisfied by clustering algorithm.

1) Scalability
2) Dealing with different types of attributes
3) Discovering clusters of arbitrary shape
4) Ability to deal with noise and outliners
5) High dimensionality
6) Insensitivity to the order of attributes
7) Interpretability and usability

Major problems encountered with clustering algorithms are:

- Dealing with large number of dimensions and a large number of objects can be prohibitive due to time complexity.
- The effectiveness of an algorithm depends on the definition of similarity measure.
- The outcome of an algorithm can be interpreted in different ways.

Now, try an exercise.

E2) Classify the clustering algorithms, along with examples.

In the following section, we shall discuss clustering methods.

13.4 CLUSTERING METHODS

In this section, we shall cover the major clustering techniques.

1. **Hierarchic versus Non-hierarchic Methods:** This is a major distinction involving both the methods and the classification structures designed with them. The hierarchic methods generate clusters as nested structures, in a hierarchical fashion; the clusters of higher levels are aggregations of the clusters of lower levels. Non-hierarchic methods result in a set of un-nested clusters. Sometimes, the user, even when he utilizes a hierarchical clustering algorithm, is interested rather in partitioning the set of the entities considered.

2. **Agglomerative versus Divisive Methods:** Agglomerative method is a bottom up approaching and involves merging smaller clusters into larger ones while the Divisive method is a top-down approach where large clusters are split into smaller ones. Agglomerative methods have been developed for processing mostly similarity/dissimilarity data while the divisive methods mostly work with attribute-based information, producing attribute-driven subdivisions (conceptual clustering).

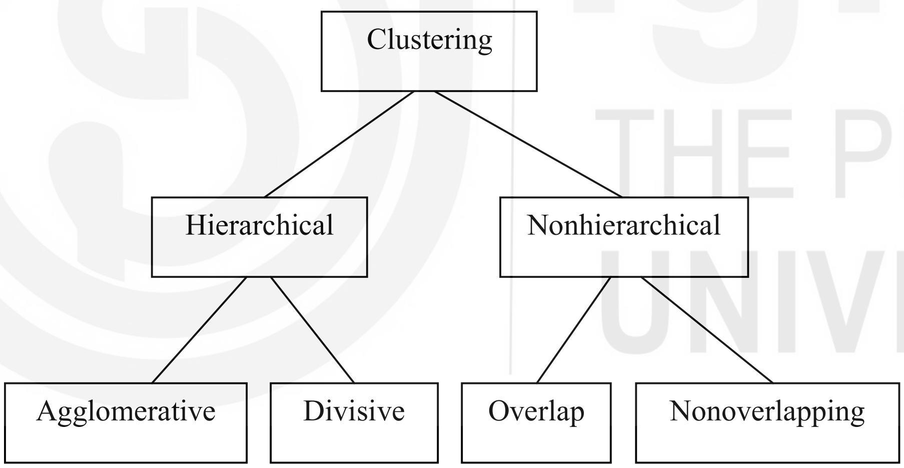


Fig 3: Classical Taxonomy of Clustering Methods

Try an exercise.

E3) How are different clustering methods classified?

Now, let us discuss hierarchical clustering is detail in the following section.

13.5 HIERARCHICAL CLUSTERING

These methods construct the clusters by recursively partitioning the instances in either a top-down or bottom-up fashion. A hierarchy can be represented by a tree structure such as the simple one shown in Fig. 4.

For example, patients in an animal hospital are composed of two main groups, dogs and cats, each of which can be sub-divided to further subgroups. Each of the individual animals, 1 through 5, is represented at the lowest level of the tree. Hierarchical clustering refers to a clustering process that organizes the data into large groups, which contain smaller groups and so on. A hierarchical clustering can be drawn as a tree or dendrogram. The finest grouping is at the bottom of the dendrogram, each sample by itself forms a cluster. The coarsest grouping is at the top of the dendrogram, where all samples are grouped into a cluster.

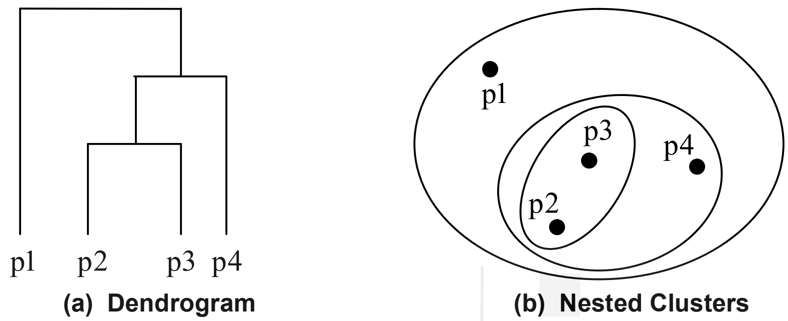


Fig. 4: A Hierarchical Clustering of Four Points

Logically, several approaches are possible to find a hierarchy associated with the data. The popular approach is to construct the hierarchy level-by-level, from bottom to top (agglomerative clustering) or from top to bottom (divisive clustering). Let us discuss hicrachical clustering methods one by one in detail.

Agglomerative Hierarchical Clustering

Agglomerative hierarchical techniques are the more commonly used methods for clustering. Each object initially represents a cluster of its own. Then clusters are successively merged until the desired cluster structure is obtained. Divisive hierarchical clustering. All objects initially belong to one cluster. Then the cluster is divided into sub-clusters, which are successively divided into their own sub-clusters. This process continues until the desired cluster structure is obtained. The result of the hierarchical methods is a dendrogram, representing the nested grouping of objects and similarity levels at which groupings change. A clustering of the data objects is obtained by cutting the dendrogram at the desired similarity level. The merging or division of clusters is performed according to some similarity measure, chosen so as to optimize some criterion (such as a sum of squares).

The steps of general agglomerative clustering algorithm are as follows:

Step 1: Begin with N clusters. Each cluster consists of one sample.

Step 2: Repeat Step 2 a total of $\mathrm{N}-1$ times.

Step 3: Find the most similar clusters $\mathrm{C_i}$ and $\mathrm{C_j}$ and marge $\mathrm{C_i}$ and $\mathrm{C_j}$ into one cluster. If there is a tie, merge the first pair found.

Types of Agglomerative Clustering

Nearest neighbor Method (also called Single-link clustering connectedness, the minimum method or the nearest neighbor method) methods that consider the distance between two clusters to be equal to the shortest distance between any member of one cluster to any member of the other cluster.

If the data consist of similarities, the similarity between a pair of clusters is considered to be equal to the greatest similarity from any member of one cluster to any member of the other cluster. This method has a tendency to cluster together at an early stage objects that are distant from each other in the same cluster because of a chain of intermediate objects in the same cluster. Such clusters have elongated sausage-like shapes when visualized as objects in space.

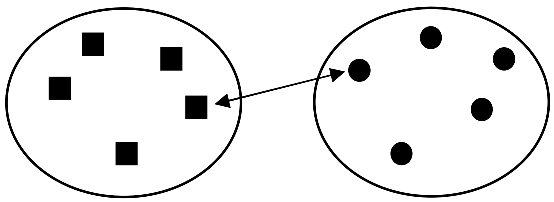


Fig. 5: Cluster Distance in Nearest Neighbour Method

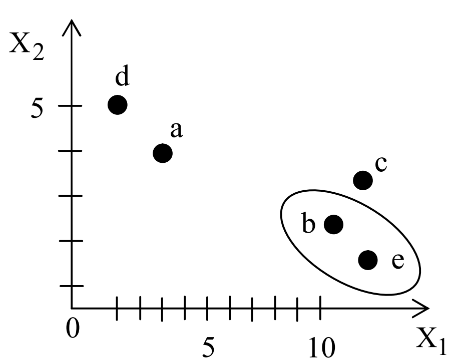
Example 4: Let us suppose that Euclidean distance is the appropriate measure of proximity. Consider the five observations given as a, b, c, d and are shown in Fig. 6(b), and are forming its own cluster. The distance between each pair of observations is shown in Fig. 6(a).
For example, the distance between a and b is

$$\sqrt{(2-8)^2 + (4-2)^2} = \sqrt{36+4} = 6.325.$$

Observations b and e are nearest (most similar) and, as shown in Fig. 6(b), are grouped in the same cluster. Assuming the nearest neighbor method is used, the distance between the cluster (be) and another observation is the smaller of the distances between that observation, on the one hand, and b and e, on the other.

Cluster	a	b	c	d	e
a	0	6.325	7.071	1.414	7.159
b		0	1.414	7.616	1.118
c			0	8.246	2.062
d				0	8.500
e					0

(a)



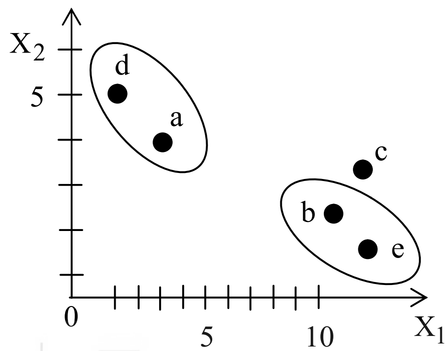
(b)

Fig. 6: Nearest Neighbour Method, (Step 1).

For example, $D(be,a) = \min\{D(b,a), D(e,a)\} = \min\{6.325, 7.159\} = 6.325.$

The four clusters remaining at the end of this step and the distances between these clusters are shown in Fig. 7(a).

Cluster	(be)	a	c	d
(be)	0	6.325	1.414	7.614
a		0	7.071	1.414
c			0	8.246
d				0



(a) **(b)**

Fig. 7: Nearest Neighbour Method, (Step 2).

Two pairs of clusters are closest to one another at distance 1.414; these are (ad) and (bce). We arbitrarily select (ad) as the new cluster, as shown in Fig. 7(b).

The distance between (be) and (ad) is

$$D(be, ad) = \min\{D(be,a), D(be,d)\} = \min\{6.325, 7.616\} = 6.325,$$

while that between c and (ad) is

$$D(c, ad) = \min\{D(c,a), D(c,d)\} = \min\{7.071, 8.246\} = 7.071.$$

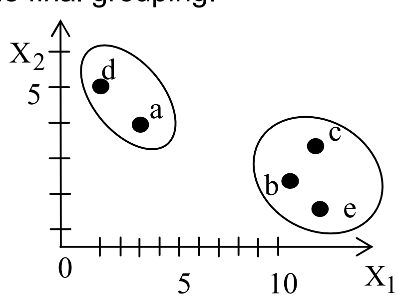
The three clusters remaining at this step and the distances between these clusters are shown in Fig. 8 (a). We merge (be) with c to form the cluster (bce) shown in Fig. 8 (b).

The distance between the two remaining clusters is

$$D(ad, bce) = \min\{D(ad, be), D(ad, c)\} = \min\{6.325, 7.071\} = 6.325.$$

The grouping of these two clusters, it will be noted, occurs at a distance of 6.325, a much greater distance than that at which the earlier groupings took place. Fig. 9 shows the final grouping.

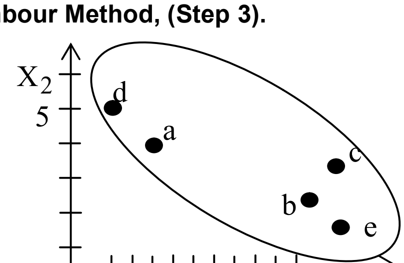
Cluster	(be)	(ad)	c
(be)	0	6.325	1.414
(ad)		0	7.071
c			0



(a) **(b)**

Fig. 8: Nearest Neighbour Method, (Step 3).

Cluster	(bce)	(ad)



(bce)	0	6.325
(ad)		0

(a) **(b)**

Fig. 9: Nearest Neighbour Method, (Step 4).

The groupings and the distance between the clusters are also shown in the tree diagram (dendrogram) of Fig.10. One usually searches the dendrogram for large jumps in the grouping distance as guidance in arriving at the number of groups. In this example, it is clear that the elements in each of the clusters (ad) and (bce) are close(they were merged at a small distance), but the clusters are distant (the distance at which they merge is large).

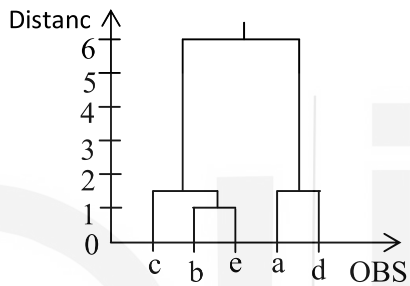


Fig. 10: Nearest neighbour method, (Dendrogram)

***

Complete-link clustering (also called the diameter method, the maximum method or the furthest neighbour method) - methods that consider the distance between two clusters to be equal to the longest distance from any member of one cluster to any member of the other cluster. The nearest neighbour is not the only method for measuring the distance between clusters. Under the furthest neighbor (or complete linkage) method, the distance between two clusters is the distance between their two most distant members. This method tends to produce clusters at the early stages that have objects that are within a narrow range of distances from each other. If we visualize them as objects in space the objects in such clusters would have a more spherical shape as shown in Fig. 11.

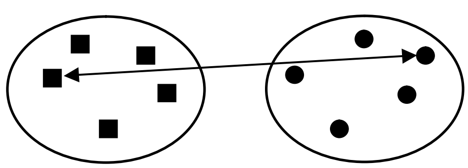


Fig. 11: Cluster Distance (Furthest Neighbour Method)

maximum distance $\cong$ minimum similarity

$$d_{complete}(A,B) := \max_{a \in A, b \in B} d(a,b) \cong \min_{a \in A, b \in B} s(a,b).$$

Now let us understand this through following example:

Example 5: Consider the example data presented in Fig. 6. Therefore, the furthest neighbor method also calls for grouping band e at Step 1.

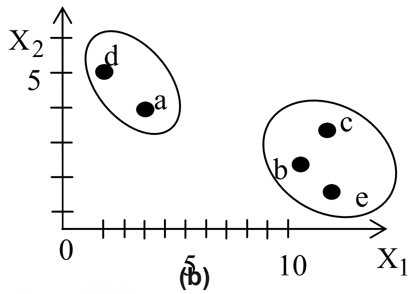
However, the distances between (be), on the one hand, and the clusters (a), (c), and (d), on the other, are different:

$$D(be, a) = \max\{D(b, a), D(e, a)\} = \max\{6.325, 7.159\} = 7.159$$
$$D(be, c) = \max\{D(b, a), D(e, c)\} = \max\{1.414, 2.062\} = 2.062$$
$$D(be, d) = \max\{D(b, d), D(e, d)\} = \max\{7.616, 8.500\} = 8.500$$

The four clusters remaining at Step 2 and the distances between these clusters are shown in Fig. 12(a).

Cluster	(be)	a	c	d
(be)	0	7.159	2.062	8.500
a		0	7.071	1.414
c			0	8.246
d				0

(a)



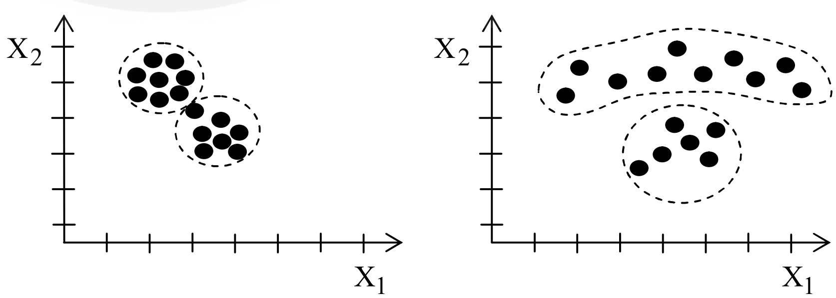
(b)

Fig.12: Furthest Neighbour Method (Step 2).

The nearest clusters are (a) and (d), which are now grouped into the cluster (ad). The remaining steps are similarly executed.

***

You may confirm from the Example 4 and Example 5 that the nearest and furthest neighbour methods produce the same results. In other cases, however, the two methods may not agree. Consider Fig. 13(a) as an example. The nearest neighbour method will probably not form the two groups perceived by the naked eye. This is so because at some intermediate step the method will probably merge the two “nose" points joined in Fig. 13(a) into the same cluster, and proceed to string along the remaining points in chain-link fashion. The furthest neighbour method, will probably identify the two clusters because it tends to resist merging clusters the elements of which vary substantially in distance from those of the other cluster. On the other hand, the nearest neighbour method will probably succeed in forming the two groups marked in Fig. 13(b), but the furthest neighbor method will probably not.



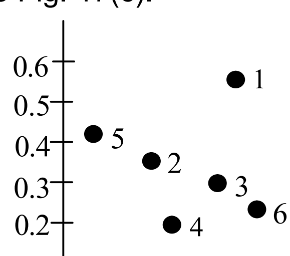
(a) (b)

Fig. 13: Two Cluster Patterns

Now, try the following exercises:

E4) Consider the data given in Fig. 17(a) to Fig. 17(c).

Point	x - **Coordinate**	y - **Coordinate**



p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	045	

(a) $x - y$ **coordinates for 6 points (b) Graph for 6 two-dimensional points**

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

(c) Euclidean Distance Matrix for 6 Points

Fig. 17

Perform clustering using

(i) single link clustering

(ii) complete link clustering

(iii) average link clustering

(iv) Ward's method

In the following section, we shall discuss partitioned clustering.

13.6 PARTITIONAL CLUSTERING

Partitioning clustering begins with a starting cluster partition which is iteratively improved until a locally optimal partition is reached. The starting clusters can be either random or the cluster output from some clustering pre-process (e.g. hierarchical clustering). In the resulting clusters, the objects in the groups together add up to the full object set. Partitioning procedures differ with respect to the methods used to determine the initial partition of the data, how assignments are made during each pass or iteration, and the clustering criterion used. The most frequently used method assigns objects to the clusters having the nearest centroid. This procedure creates initial partitions based on the results from preliminary hierarchical cluster procedures such as the average linkage method or Ward's method, a procedure that resulted in partitioning methods being referred to as two-stage cluster analysis. Some partitioning methods use multiple passes during which cluster centroids are recalculated and objects are re-evaluated, whereas other methods use a single-pass procedure. Partitioning methods also differ with respect to how they evaluate an object's distance from cluster centroids. Some procedures use simple distance and others use more complex multivariate matrix criteria. Finally, most partitioning methods require that the user specify a priori how many clusters will be formed. Let us discuss an important algorithm known as Frogy's algorithm, which is used for partitional clustering.

Frogy's algorithm: One of the simplest partitional clustering algorithm is Frogy's algorithm. Input to the algorithm consists of data, k, number of clusters to be constructed and k samples called **seed points**. Seed points could be chosen randomly or some knowledge of the desired cluster structure could be the starting point.

The following steps are performed:

Step 1: Initialize the cluster centroid to the seed points.

Step 2: For each sample, find the cluster centroid nearest to it. Put the samples in the cluster identified with the nearest cluster centroid.

Step 3: If no samples changed clusters in Step 2, stop.

Step 4: Compute the centroids of the resulting clusters and go to step 2.

Let us apply these steps in the following example.

Example 6: Perform partitional clustering using Frogy's method for the data given in Fig. 18 (a) with k-2 (two clusters). Use first two sample points (4,4) and (8,4) as seed points.

	x	y
1	4	4
2	8	4
3	15	8
4	24	4
5	24	12

(a) x-y Coordinates for 5 Points

Sample	Nearest cluster centroid
(4,4)	(4,4)
(8,4)	(8,4)
(15,8)	(8,4)
(24,4)	(8,4)
(24,12)	(8,4)

(b) First Iteration

Sample	Nearest cluster centroid
(4,4)	(4,4)
(8,4)	(4,4)
(15,8)	(17.75,7)
(24,4)	(17.75,7)
(24,12)	(17.75,7)

(c) Second Iteration

Sample	Nearest cluster centroid
(4,4)	(6,4)
(8,4)	(6,4)
(15,8)	(21,8)
(24,4)	(21,8)
(24,12)	(21,8)

(d) Third Iteration

Fig. 18

For Step 2, find the nearest cluster centroid for each sample. Fig. 18(b) shows the results. The clusters {(4,4)} and {(8,4), (15,8), (24,4), (24,12)} are produced.

For Step 4, we compute the centroid of the clusters. The centroid of first cluster is (4,4). The centroid of second cluster is (17.75,7) as

(8+15+24+24)/4=17.75 and (4+8+4+12)/4 =7.

As samples change clusters, go to Step 2.

Find cluster centroid nearest each sample. Fig. 18(c) shows the results. The clusters {(4,4),(8,4), } and { (15,8), (24,4), (24,12)} are produced. For Step 4, we compute the centroid (6,4) and (21,8) of the clusters. As sample (8,4) changed cluster, return to Step 2.

Find cluster centroid nearest each sample. Fig. 17(d) shows the results. The clusters {(4,4),(8,4), } and { (15,8), (24,4), (24,12)} are produced.

For Step 4, we compute the centroid (6,4) and (21,8) of the clusters. As no sample changed clusters, the algorithm terminates.

***

Try an exercise.

E5) Consider the data

Sample	x	y
1	0	0
2	1	0
3	0	2
4	2	2
5	3	2
6	6	3
7	7	3

Perform a partitional clustering using

(i) $k = 2$ and use the first two samples in the list as seed points.
(ii) $k = 3$ and use the first three samples in the list as seed points.

In the following section, we discuss k-means clustering.

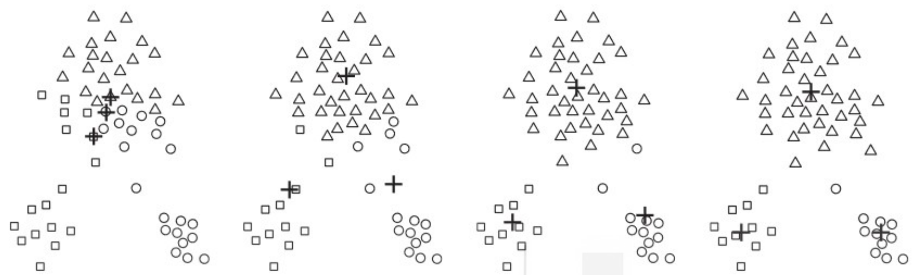
13.7 K-MEANS CLUSTERING

The K-means clustering technique is simple, and we first choose k initial centroids, where k is a user-specified parameter, namely, the number of clusters desired. Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated based on the points assigned to the cluster. We repeat the assignment and update steps until no point changes clusters, or equivalently, until the centroids remain the same. In its simplest form, the k-means method follows the following steps.

Step 1: Specify the number of clusters and, arbitrarily or deliberately, the members of each cluster.
Step 2: Calculate each cluster's \centroid" (explained below), and the distances between each observation and centroid. If an observation is nearer the centroid of a cluster other than the one to which it currently belongs, re-assign it to the nearer cluster.
Step 3: Repeat Step 2 until all observations are nearest the centroid of the cluster to which they belong.
Step 4: If the number of clusters cannot be specified with confidence in advance, repeat Steps 1 to 3 with a different number of clusters and evaluate the results.

The operation of K-means are shown in Fig. 19, which shows how, starting from three centroids, the final clusters are found in four assignment-update steps. In these and other figures displaying K-means clustering, each subfigure shows (1) the centroid sat the start of the iteration and (2) the assignment of the points to those centroids. The centroids are indicated by the "+" symbol. All points belonging to the same cluster have the same marker shape.

In the first step, shown in Fig. 19(a), points are assigned to the initial centroids, which are all in the larger group of points. For this example, we use the mean as the centroid. After points are updated again. In steps 2, 3, and 4, which are shown in Fig. 19(b), (c), and (d), respectively, two of the centroids move to the two small groups of points



(a) First Iteration (b) Second iteration (c) Third Iteration (d) Fourth Iteration

Fig.19: Using the K-Means Algorithm

at the bottom of the figures. When the K-means algorithm terminates in Fig. 19(d), because no more changes occur, the centroids have identified the natural groupings of points. Centroid at the beginning of the step and the assignment of points to those centroids. In the second step, points are assigned to the updated centroids, and the centroids.

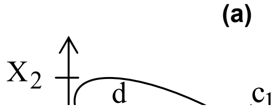
Let us understand this in the following example.

Example 7: Suppose two clusters are to be formed for the observations listed in Fig. 20(a). We begin by arbitrarily assigning a, b and d to Cluster 1, and c and e to Cluster 2. The cluster centroids are calculated as shown in Fig. 20(a).

The cluster centroid is the point with coordinates equal to the average values of the variables for the observations in that cluster. Thus, the centroid of Cluster 1 is the point $(X_1 = 3.67, X_2 = 3.67)$, and that of Cluster 2 the point (8.75, 2). The two centroids are marked by C_1 and C_2 in Fig. 20(a). The cluster's centroid, therefore, can be considered the center of the observations in the cluster, as shown in Fig. 20(b). We now calculate the distance between a and the two centroids.

Cluster 1			Cluster 2		
Observation	X_1	X_2	Observation	X_1	X_2
a	2	4	c	9	3
b	8	2	e	8.5	1
d	1	5			
Average	3.67	3.67	Average	8.75	2

(a)



(b)

Fig. 20: Means Method (Step 1)

$$D(a,abd)=\sqrt{(2-3.67)^2+(4-3.67)^2}=1.702.$$
$$D(a,ce)=\sqrt{(2-8.75)^2+(4-2)^2}=7.040.$$

Observe that a is closer to the centroid of Cluster 1, to which it is currently assigned. Therefore, a is not reassigned. Next, we calculate the distance between b and the two cluster centroids:

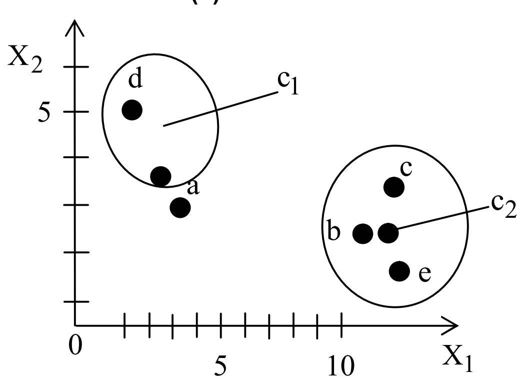
$$D(b,abd)=\sqrt{(8-3.67)^2+(2-3.67)^2}=4.641.$$
$$D(b,cc)=\sqrt{(8-8.75)^2+(2-2)^2}=0.750.$$

Since b is closer to Cluster 2's centroid than to that of Cluster 1, it is reassigned to Cluster 2. The new cluster centroids are calculated as shown in Fig. 21(a).The new centroids are plotted in Fig. 21(b). The distances of the observations from the new cluster centroids are shown in Fig. 21(c).

(an asterisk indicates the nearest centroid):

Cluster 1			Cluster 2		
Observation	X_1	X_2	Observation	X_1	X_2
a	2	4	c	9	3
d	1	5	e	8.5	1
			b	8	2
Average	1.5	4.5	Average	8.5	2

(a)



(b)

Distance from

Observation	Cluster 1	Cluster 2
a	0.707*	6.801
b	6.964	0.500*
c	7.649	1.118*
d	0.707*	8.078
e	7.826	1.000*

(c)

Fig. 21: Means Method (Step 2)

Every observation belongs to the cluster to the centroid of which it is nearest, and the k-means method stops. The elements of the two clusters are shown in Fig. 21(c).

***

Now, we list the benefits and drawbacks of k-means methods.

Benefits:

1) Very fast algorithm (O (k . d . N), if we limit the number of iterations)
2) Convenient centroid vector for every cluster
3) Can be run multiple times to get different results

Limitations:

1) Difficult to choose the number of clusters, k
2) Cannot be used with arbitrary distances
3) Sensitive to scaling – requires careful preprocessing
4) Does not produce the same result every time
5) Sensitive to outliers (squared errors emphasize outliers)
6) Cluster sizes can be quite unbalanced (e.g., one-element outlier clusters)

Try an exercise.

E6) What are advantages and disadvantages of k-means clustering methods?

Now let us summaries what we have learnt in this unit.

13.8 SUMMARY

We have discussed the following points:

1) Concept of clustering.
2) Various distance measures.
3) Various clustering methods.
3) Analyzed various Hierarchical clustering algorithms in detail.
4) Analyzed various Partitional clustering and k-nn clustering algorithms.

13.9 SOLUTION/ ANSWERS

E1) Different formula in defining the distance between two data points can lead to different classification results. Domain knowledge must be used to guide the formulation of a suitable distance measure for each particular application. For high dimensional data, a popular measure is the Minkowski

$$d(x_i, x_j) = \left(\sum_{k=1}^{d} | x_{i,k} - x_{j,k} |^p \right)^{\frac{1}{p}}$$ Metric:

Where d is the dimensionality of the data.
Special Cases:

• p=2: Euclidean distance

• p=1: Manhattan distance

The commonly used Euclidean distance between two objects is achieved when $p = 2$.

$$d_{ij} = ((x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{id} - x_{jd})^2)^{\frac{1}{2}}$$

Another well-known measure is the Manhattan distance which is defined when $p = 1$.

$$d_{ij} = | (x_{i1} - x_{j1}) + (x_{i2} - x_{j2}) + \cdots + (x_{id} - x_{jd}) |$$

The Mahalanobis distance is another very important distance measure used in statistics that measures the statistical distance between two populations of Gaussian mixtures having mean μ_i and μ_j and a common covariance matrix $\sum_{ij}$. This measure is given by

$$d_{ij} = (\mu_i - \mu_j)^T \sum_{ij} (\mu_i - \mu_j).$$

E2) i) Exclusive clustering

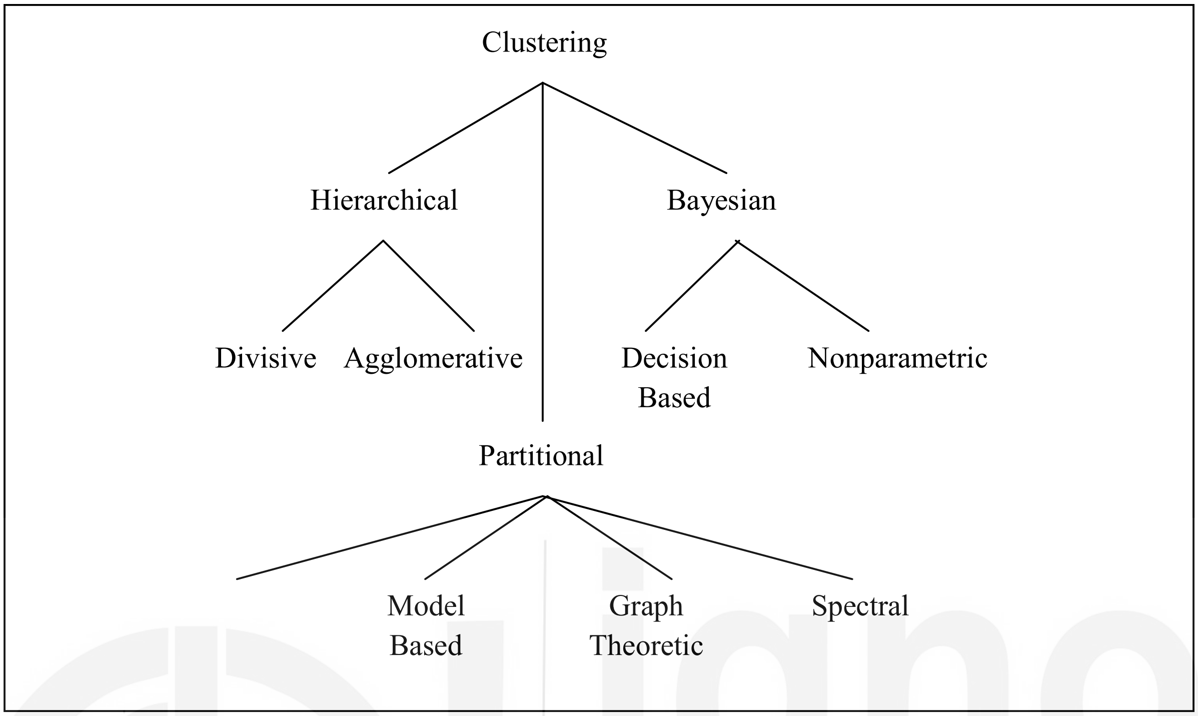
ii) Overlapping clustering

iii) Agglomerative clustering

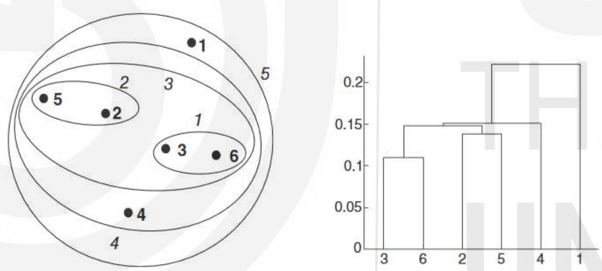
iv) Divisive clustering

v) Probabilistic clustering

E3)



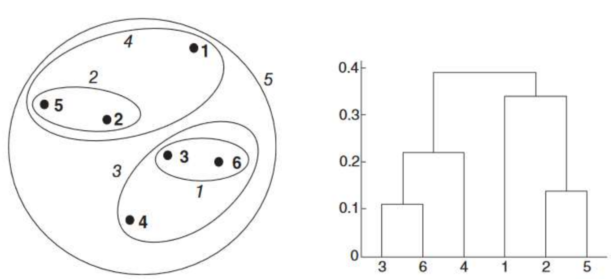
E4) (i) Single link clustering



(a) Single Link Clustering **(b) Single Link Dendrogram**

$$
\begin{aligned}
\text{dist}(\{3,6\},\{2,5\}) &= \min(\text{dist}(3,2), \text{dist}(6,2), \text{dist}(3,5), \text{dist}(6,5)) \\
&= \min(0.15, 0.25, 0.28, 093) \\
&= 0.15.
\end{aligned}
$$

(ii) Complete link clustering



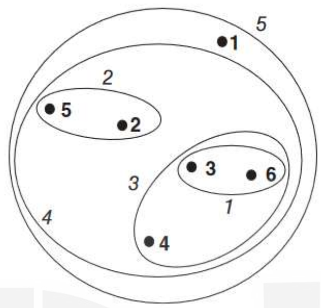
(a) Complete Link Clustering **(b) Complete Link Dendrogram**

$$\begin{aligned}\mathrm{dist}(\{3,6\}),\{4\}) &= \max(\mathrm{dist}(3,4),\mathrm{dist}(6,4)) \\ &= \max(0.15, 0.22) \\ &= 0.22\end{aligned}$$

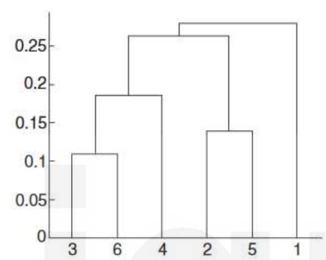
$$\begin{aligned}\mathrm{dist}(\{3,6\},\{2,5\} &= \max(\mathrm{dist}(3,2),\mathrm{dist}(6,2),\mathrm{dist}(3,5),\mathrm{dist}(6,5)) \\ &= \max(0.15, 0.25, 0.28, 0.39) \\ &= 0.39.\end{aligned}$$

$$\begin{aligned}\mathrm{dist}(\{3,6\},\{1\}) &= \max(\mathrm{dist}(3,1),\mathrm{dist}(6,1)) \\ &= \max(0.22, 0.23) \\ &= 0.23.\end{aligned}$$

(iii) Average link clustering



(a) Group Average Clustering



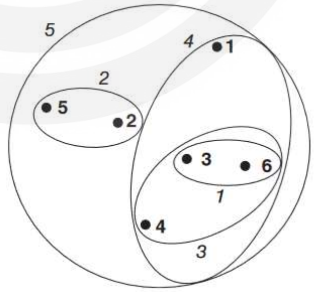
(b) Group Average Dendrogram

$$\begin{aligned}\mathrm{dist}(\{3,6,4\},\{1\}) &= (0.22 + 0.37 + 0.23)/(3*1) \\ &= 0.28\end{aligned}$$

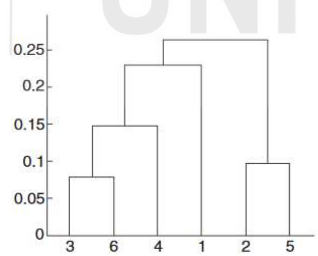
$$\begin{aligned}\mathrm{dist}(\{2,5\},\{1\}) &= (0.2357 + 0.3421)/(2*1). \\ &= 0.2889\end{aligned}$$

$$\begin{aligned}\mathrm{dist}(\{3,6,4\},\{2,5\}) &= (0.15 + 0.28 + 0.25 + 0.39 + 0.20 + 0.29)/(6*2) \\ &= 0.26\end{aligned}$$

iv) Clustering using Ward's method



(a) Ward's Clustering



(b) Ward's Dendrogram

E6) Benefits of k-nnalgorithm:

1) Very fast algorithm (O(k . d . N), if we limit the number of iterations)
2) Convenient centroid vector for every cluster
3) Can be run multiple times to get different results

Limitations of k-nn algorithm:

1) Difficult to choose the number of clusters, k
2) Cannot be used with arbitrary distances
3) Sensitive to scaling – requires careful preprocessing
4) Does not produce the same result every time
5) Sensitive to outliers (squared errors emphasize outliers)
6) Cluster sizes can be quite unbalanced (e.g., one-element outlier clusters)