1

INTRODUCTION TO IMAGE-PROCESSING SYSTEM

Unit structure

1.0 OBJECTIVES

As the name indicates digital image processing helps:

- To study two-dimensional Signals and Systems.
- To understand image fundamentals and transforms necessary for image processing.
- To study the image enhancement techniques in spatial and frequency domain.
- To study image segmentation and image compression techniques

1.1 INTRODUCTION

In current era digital plays an important role in several areas like geography, information technology, medicine etc.

Digital image processing(DIP)refers to management of digital images through a digital computer. It is a subarea of signals and systems but particularly focusses on images. DIP aims on using a computer system that is able to perform several processing of an image. The process contains the input of that system which is a digital image , the system then process that image using effective algorithms, and gives as an output an image. The most general example is Matlab, Adobe Photoshop. It is widely used for processing digital images.

Before proceeding let us define some terms as follows :

1. **IMAGE :** An image is defined as an two dimensional signal. It is defined using mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically called as spatial or plane coordinates.The value of f(x,y) at any point is gives the pixel value at that point of an image.represents a measure of some characteristic such as brightness or color of a viewed scene. An image is a projection of a 3- D scene into a 2D projection plane

2. **ANALOG IMAGE:** An analog image cab ne represented mathematically using continuous range of values which represent intensity and position. Some of the examples include television images, photographs, paintings, and medical images etc.

3. **DIGITAL IMAGE :**It is composed of matrix of small pixels. For operating with the images, there are various software and algorithms. Some of the examples of digital images are color processing, image recognition, video processing, etc. Analog signals can be convted into digital using two important operations i.e sampling and quantisation.

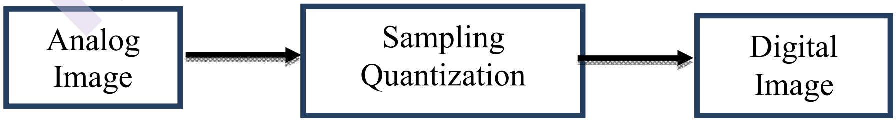


Figure 1

Advantages of Digital Image Processing

- Fast processing and effectiveness of cost
- Easier Image rebuilding and reformatting
- Speedy image storing and retrieval
- Fast and high-quality image distribution.

Disadvantages of Digital Image Processing

- It is very much time-consuming, needs high speed processor.
- Enlargement of image not possible after certain size.
- It is very much expensive depending on the particular system.

4. **Digital Image representation :**

A digital image is an :

- N x N array of elements
- 2-dimensional discrete signal

To convert an image in digital format either scanner or digital camera is used. These can be created directly on computer screens.

5 Neighbours of Pixel

A pixel will have four neighbpurs if they exist. NORTH, SOUTH, EAST and WEST.

	North	
West	P	East
	South	

Figure 2

1.2 IMAGE SAMPLING

Sampling is the procedure of evaluating at discrete spatial points the information of brightness . f(x,y) – a continuous image function can be sampled in the plane with the help of discrete grid of sampling points.

1.2.1 THEORY of 2D SAMPLING

Let the analog image be represented by f(x,y). the corresponding discrete edition of f(x,y) is obtained by defining f(x,y) at particular instances called as sampling intervals $\Delta x, \Delta y$ which are real positive constants.

$$f(m, n) = f(m\Delta x, n\Delta y)$$

Steps in 2D sampling are as follows :

1. Take analog sample f(x,y)
2. Take fourier transform after which we get spectrum of f(x,y) denoted by $F(\Omega_1, \Omega_2)$ as

$$F(\Omega_1, \Omega_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y) e^{-j\Omega_1 x} e^{-j\Omega_2 y}\, dx\, dy$$

3. Then take inverse Fourire transform after which we get

$$f(x,y)=\frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(\Omega_1,\Omega_2)e^{j\Omega_1 x}e^{j\Omega_2 y}d\Omega_1 d\Omega_2$$

4. For discrete signal we specify $\omega_1 = \Omega_1\Delta x \text{ and } \omega_2 = \Omega_2\Delta y$ where ω_1 and ω_2 are expressed in radians

5. Differentiating ω_1 , ω_2 and substituting in Inverse transform in step 3 we get discretised form of f(x,y) as given below

$$f(m,n)=\frac{1}{4\pi^2}\iint_{SQ(k_1,k_2)}\frac{1}{\Delta x\Delta y}\sum_{k_1}\sum_{k_2}F\left(\frac{\omega_1}{\Delta x},\frac{\omega_2}{\Delta y}\right)e^{j\omega_1 m}e^{j\omega_2 n}d\omega_1 d\omega_2$$

6. Replacing the integration by summation , changing limits we have

$$F(\omega_1,\omega_2)=\frac{1}{\Delta x\Delta y}\sum_{k_1}\sum_{k_2}F\left(\frac{\omega_1-2\pi k_1}{\Delta x},\frac{\omega_2-2\pi k_2}{\Delta y}\right)$$

An alertnate approach for the above procedure is multiplying the anlaog image by a 2D comb function.It is a rectangular grid of points as shown in figure 3 . the spaces in between thr grid points are Δx,Δy respectively.

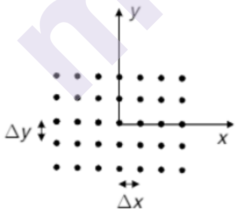


Figure 3

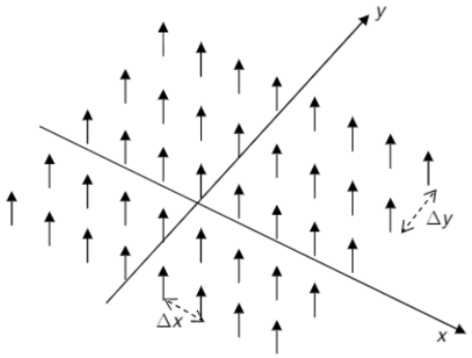


Figure 4

Figure 4 represents a 3 Dimensional view of comb function . The 2D view also called as bed-of-nail function defined by :

$$\text{comb}(x,y,\Delta x,\Delta y)=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}\delta(x-k_1\Delta x,\,y-k_2\Delta y)$$

After multiplying the comb function and analog function f(x,y) we get discrete version of analog image given by

$$f(m,n) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} f(k_1\Delta x, k_2\Delta y)\,\delta\,(x - k_1\Delta x,\, y - k_2\Delta y)$$

Next fourier transform of above product of functions is computed and convolution of signals is applied to obtain

$$F(\omega_1, \omega_2) = \frac{1}{\Delta x}\frac{1}{\Delta y}\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} F(\Omega_1 - k, \Omega_2 - l) \sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} \delta\left(k - \frac{p}{\Delta x}, l - \frac{q}{\Delta y}\right)$$

Summation being an linear operator , interchanging the order we get :

$$F(\omega_1, \omega_2) = \frac{1}{\Delta x}\frac{1}{\Delta y}\sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty} F\left(\Omega_1 - \frac{p}{\Delta x}, \Omega_2 - \frac{q}{\Delta y}\right)$$

We can see that the final equations from both the methods are similar.

1.2.2 RETRIEVING IMAGE FROM ITS SAMPLE

Since discreteness in one domain leads to periodicity in other domain, sampling in spatial domain also leads to a periodic spectrum in the frequency domain as in figure 5

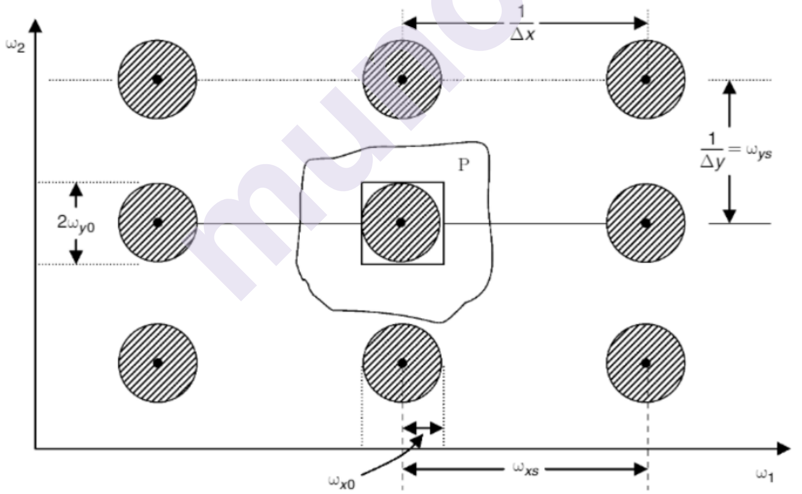


Figure 5 :Periodic spectrum of sampled image.

In order to retrieve the original image from the sampled spectrum we apply the condition : **sampling frequency should be greater than two times the maximum signal frequency which is called "The Sampling Theorem"**

That is (*)

$$\omega_{xs} > 2\omega_{x0} \qquad\qquad \omega_{ys} > 2\omega_{y0}$$

Where $2\omega_{x0}$ *and* $2\omega_{y0}$ are called **Nyquist rates** .

A low pass filter is used to extract the desired spectrum whose transfer function is given by :

$$H(\omega_1, \omega_2) = \begin{cases} \frac{1}{\omega_{1s}\omega_{2s}}, & (\omega_1, \omega_2) \in \text{region of support} \\ 0 & \text{otherwise} \end{cases}$$

The region of support is indicated as P in figure 5. The continuous image can be obtained from sampled spectrum by multiplying the sampled spectrum with low pass filter as given below :

$$\hat{F}(\omega_1, \omega_2) = H(\omega_1, \omega_2) \times F(\omega_1, \omega_2)$$

By taking inverse Fourier transform, we get the continuous image as

$$\hat{F}(x, y) = F^{-1}\{\hat{F}(\omega_1, \omega_2)\}$$

1.2.3 VIOLATION OF SAMPLING CRITERIA

Violation of sampling criteria given in (*) leads to aliasing which fundamentally happens due to under sampling. It also leads to overlapping of spectrum in the ω_1 direction as shown in figure 6.

. Here, $\omega_{xs} < 2_{x0}$, whereas $\omega_{ys} > 2\omega_{y0}$.

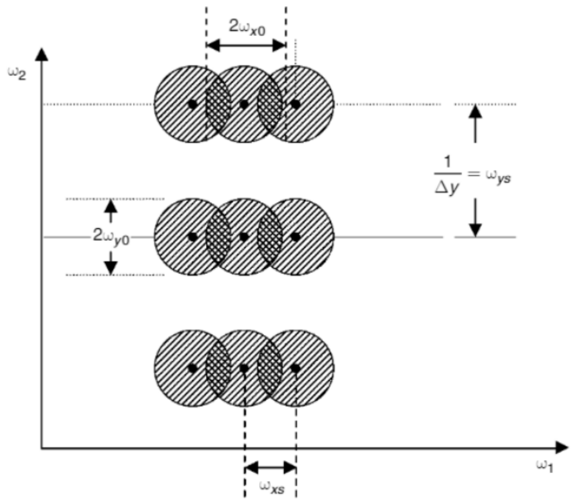


Figure 6 : Along the direction of ω_1 under sampling

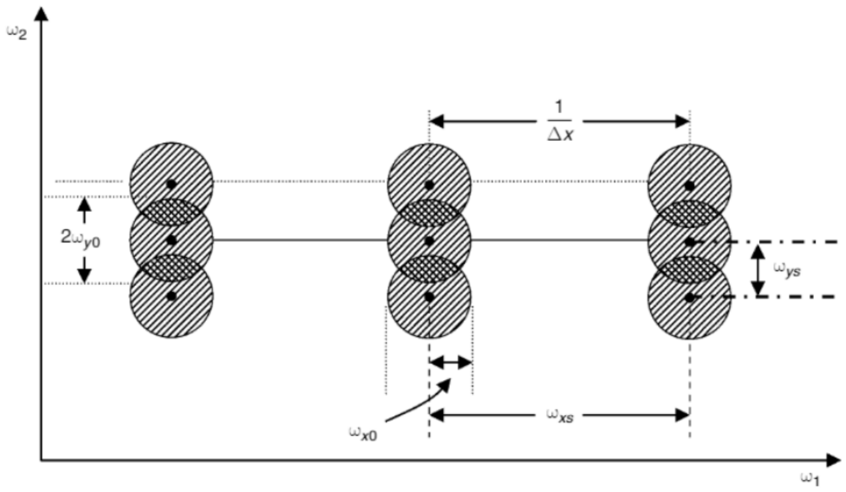


Figure 7 : Along the direction of ω_2 under sampling

In the similar manner violation of sampling criteria given in (*) leads to to overlapping of spectrum in the ω_2 direction as shown in figure 7.

Here, $\omega_{xs} < 2\omega_{x0}$ and $\omega_{ys} < 2\omega_{y0}$.

Violation of sampling criteria given in (*) simultaneously leads to to overlapping of spectrum in both the ω_1 *and* ω_2 direction as shown in figure 8.

Here, $\omega_{xs} < 2\omega_{x0}$ and $\omega_{ys} < 2\omega_{y0}$.

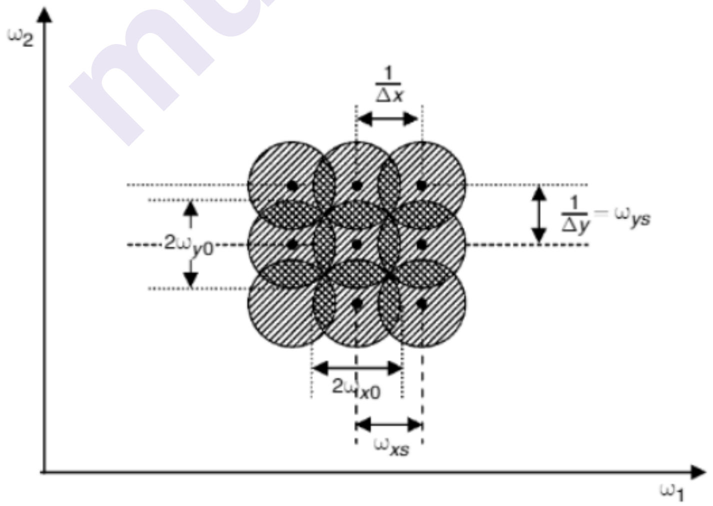


Figure 8 : Along the direction of both ω_1 and ω_2 under sampling

1.3 QUANTIZATION

Quantization is a process of transforming a real valued sampled image to one taking only a finite number of distinct values. Under quantization procedure the amplitude values of the image are digitized. In naïve words, when you are quantizing an image, you are actually partitioning a signal into quanta(partitions).

Quantiser design includes :

- Input decision level
- Output representation level
- Number of levels.

Quantisers are classified in two types :

1. Scalar quantisers
2. Vector quantisers

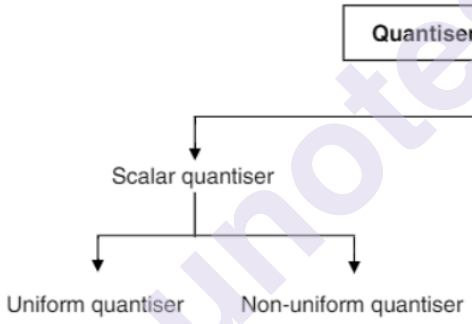


Figure 9 : Classification of quantisers

1.4 RESOLUTION

Resolution contributes the degree of distinguishable details. It is classified as :

1. Spatial resolution :Spatial resolution is defined as the smallest discernible detail in an image.Spatial resolution states that the clarity of an image cannot be determined by the pixel resolution. The number of pixels in an image does not matter.It is the number of independent pixels values per inch.sampling is the principal factor determining this resolution .
2. Gray – level resolution : Gray level resolution refers to the predictable or deterministic change in the shades or levels of gray in an image. In short gray level resolution is equal to the number of bits per pixel.

1.5 HUMAN VISUAL SYSTEM

Human Visual System (HVS) is the most complicated system in life. Many image processing applications are anticipated to produce images that are to be watched by human observers. In HVS , the eyes act as the sensor or camera, neurons act as the connecting cable and the brain acts as the processor. It is hence important to realize the characteristics and limitations of the human visual system to understand the "receiver" of the 2D signals.

1.5.1 ANATOMY OF THE HVS

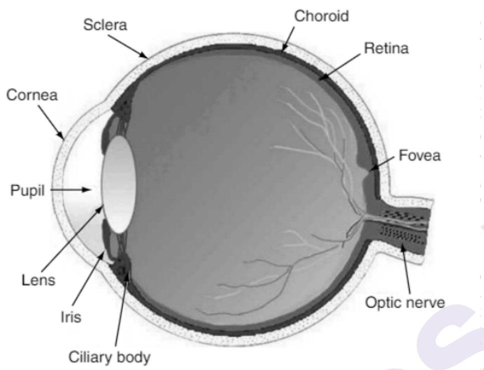


Figure 10 : Vertical section of the Human Eye

The first part of the visual system is the eye. This is shown in figure. Its form is nearly spherical and its diameter is approximately 20 mm. Its outer cover consists of the 'cornea' and 'sclera' The cornea is a tough transparent tissue in the front part of the eye. The sclera is an opaque membrane, which is continuous with cornea and covers the remainder of the eye. Directly below the sclera lies the "choroids", which has many blood vessels. At its anterior extreme lies the iris diaphragm. The light enters in the eye through the central opening of the iris, whose diameter varies from 2mm to 8mm, according to the illumination conditions. Behind the iris is the "lens" which consists of concentric layers of fibrous cells and contains up to 60 to 70% of water. Its operation is similar to that of the man made optical lenses. It focuses the light on the "retina" which is the innermost membrane of the eye. Retina has two kinds of photoreceptors: cones and rods. The cones are highly sensitive to color. Their number is 6-7 million and they are mainly located at the central part of the retina. Each cone is connected to one nerve end. Cone vision is the photopic or bright light vision. Rods serve to view the general picture of the vision field. They are sensitive to low levels of illumination and cannot discriminate colors. This is the scotopic or dim-light vision. Their number is 75 to 150 million and they are distributed over the retinal surface.

Several rods are connected to a single nerve end. This fact and their large spatial distribution explain their low resolution. Both cones and rods transform light to electric stimulus, which is carried through the optical nerve to the human brain for the high level image processing and perception.

1.5.2 SCOTOPIC AND PHOTOPIC VISION

Scotopic vision is the vision of the eye under low light conditions. Cone cells do not function as well as rod cells in low level lighting so scotopic vision occurs completely through rod cells, which are most sensitive to wavelengths of light on the electromagnetic spectrum of 498nm, which would be the blue-green bands of colour.

Photopic vision is the vision of the eye under well-lit conditions, generally usual daylight light intensity. It allows colour perception which is facilitated by cone cells. Cone cells have a higher visual acuity as well as providing the eye's colour sensitivity.

Scotopic vision uses only rods to see, meaning that objects are visible, but appear in black and white, whereas photopic vision uses cones and provides color.

1.5.3 BRIGHTNESS AND CONTRAST

The Contrast and Brightness function enhances the appearance of raster data by modifying the brightness and contrast within the image. Brightness increases the overall lightness of the image—for example, making dark colors lighter and light colors whiter—while contrast adjusts the difference between the darkest and lightest colors. Below is an example of adjustments made to the brightness and contrast of an image.

1.5.4 LAWS OF PHYSICS ASSOCIATED WITH VISION

The important laws associated with vision are :

1. **Weber's law**: It states that the just noticeable difference in stimulus intensity may affect the production of sensations proportionally.

In simple terms, the size of the intensity of stimuli will show a proportionate change in producing the sense experiences.

$\Delta I = kL$

Where ΔI(Delta I) represents the difference threshold, I represents the initial stimulus intensity and k signifies that the proportion on the left side of the equation remains constant despite variations in the I term.

It is denoted by Delta I.

2. **Steven's Law** : If L is physical stimulus and I the perceived sensation then the mathematical form of steven's law is

$I = \alpha L^n$

Where L is the intensity against black background, the value of n is 0.5

3. **Bloch's Law :** It is given by

$$T \times \Delta L = C$$

Where C is constant valid for a duration of shorter than 0.1 second

4. **Ricco's law** : It is given by

$$A \times \Delta L = C'$$

5. **Piper's law** : It is given by

$$\sqrt{A} \times \Delta L = C''$$

1.5.5 MACH BAND EFFECT

Another characteristic of HVS is that it tends to "overshoot" around image edges (boundaries of regions having different intensity). As a result, regions of constant intensity, which are close to edges, appear to have varying intensity. Such an example is shown in Figure11 . The stripes appear to have varying intensity along the horizontal dimension, whereas their intensity is constant. This effect is called Mach band effect. It indicates that the human eye is sensitive to edge information and that it has high-pass characteristics.

Figure 11 : Mach band effect

1.6 CLASSIFICATION OF DIGITAL IMAGES

Digital images can be classified in two types as follows :

1. **Raster or Bitmap Image: A**lso known as bitmap images are made of pixels or tiny dots that use color and tone to produce the image. Pixels appear like little squares on graph paper when the image is zoomed in or enlarged. These images are created by digital cameras, by scanning images into a computer or with raster-based software. Each image can only contain a fixed number of pixels; the amount of pixels determines the quality of the image. This is known as resolution. More pixels result in better quality at the same or larger sizes as the original, but this

also increases the size of the file and the amount of space it takes to store the file. The lower the number of pixels, the lower the resolution. Resolution limits the size the image can be scaled up without being able to see pixels.

Common raster formats are BMP (windows Bitmap), PCX (Paintbrush), TIFF (Tag Inter Leave Format) etc.

Figure 12 : Zooming Of Raster Image

2. **Vector Image :**These images consist of anchored dots and are connected by lines and curves, similar to the connect-the-dot activities you may have done as a kid. Because these graphics are not based on pixels, they are known as resolution independent, which makes them infinitely scalable. Their lines are sharp, without any loss in quality or detail, no matter what their size. These graphics are also device-independent, which means their quality doesn't depend on the number of dots available on a printer or the number of pixels on a screen. Because they consist of lines and anchor points, the size of the file is relatively small.

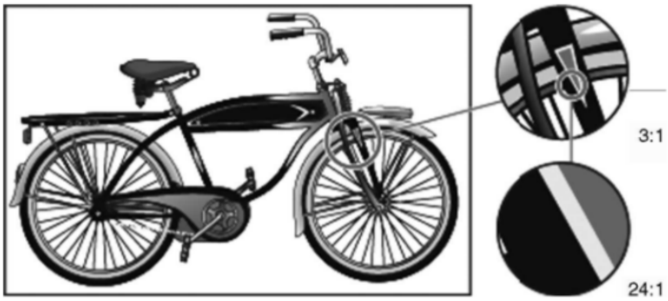


Figure 13 : Zooming Of Vector Image

1.7 IMAGE TYPES

Images are classified into four categories:

1. **Binary image :** Also called as Black and White image ,the binary image as it name sys , contain only two pixel values - 0 and 1.Here 0 refers to black color and 1 refers to white color. It is also know as Monochrome image . A gray scale image can be converted to black and white image by using threshold operation.

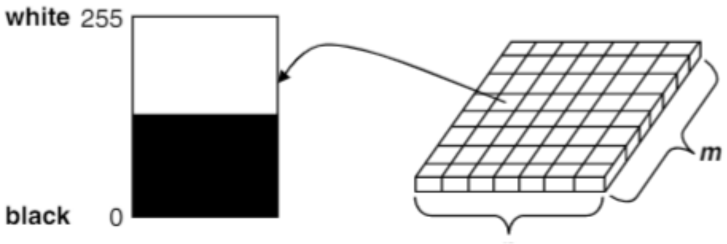


Figure 14 : Binary image representation

2. **Gray scale Image :**Grayscale images, a kind of black-and-white or gray monochrome, are composed exclusively of shades of gray. They conatin only brightness information.The contrast ranges from black at the weakest intensity to white at the strongest.In a gray scale image a pixel is represented by a word or 8-bits, where a value of 0 represents black and 255 white.

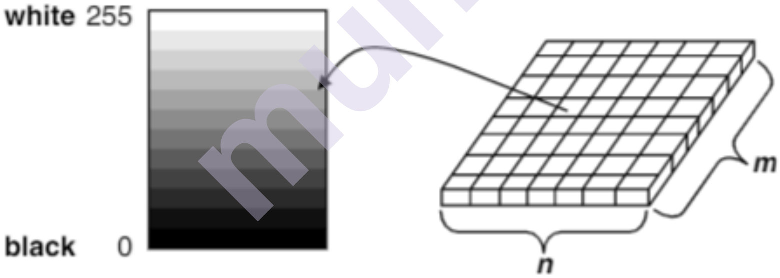


Figure 15 Gray scale image representation

3. **Colour Image :**It has three values per pixel which measure chrominance and intensity of light.Each pixel is avector of colourcomponents.Commoncolour spaces are RGB(RegGren Blue), CMYK(Cyan, Magenta, Yellow, Black) and HSV(Hue, Saturation ,Value).

4. **Volume Image:** It is three-dimensional image and can be obtained from some medical imaging euquipmentwhere individual data points are called voxels- volume pixels. Example: CAT scan

5. **Range Image :**These are special class of digital images .Here each pixel represents a distance between a reference frame and a visible point in the screen.Also called as depth images

6. **Multispectral Image :**These are images of same object taken in different bands of visible or infrared regions.they contain information outside the normal perception range of human.

1.8 ELEMENTS OF IMAGE PROCESSING SYSTEM

Different elemenst of image processing system are :

1. Image – auisition elements : Image acquisition can be defined as the act of procuring an image from sourcesi.e computer and storing them in devices . This can be done via hardware systems such as cameras, encoders, sensors, etc. Digital cameras are popular nowadays. Most of the digital cameras use CCD or CMOS image sensor. Some of thdefinitios commonly used in solid-state image sensors are:

a. **CCD (Charge-Coupled Device):** A light-sensitive array of silicon cells that is commonly used for digital camera image sensors. It generates electrical current in proportion to light input and allows the simultaneous capture of many pixels with one brief exposure.

b. **Dark current:**the charge of signal collected by pixel in absence of light

c. **Photo site**: it is the portion of silicon that functions as a light sensitive area.

d. **Pixel** : It is the discrete sensitive cell that collects and holda photo charge.

e. **Fill factor** :The fill factor of an imaging sensor is defined as the ratio of a pixel's light sensitive area to its total theoretical area.

f. **Quantum efficiency**:It is the measure of the effectiveness of an imaging device to convert incident photons into electrons

2. **Image** – storage devices : in order to store the image obtained from camera for processing and future use different storing devices are used.Basically there are two types of storage devices :

   i. Disk system

   ii. Flash memory

3. Image – display devices: the final stage of image – processing step is image display.Usually computer moniors are used for the same.some of the factors likw size of monitor, number of colours ,spatial resolution needs to be considered.

1.9 APPLICATIONS OF DIGITAL IMAGE PROCESSING

Some of the most important fields in which digital image processing is widely used are as given below :

- Image sharpening and restoration
- Medical field
- Remote sensing
- Transmission and encoding
- Machine/Robot vision
- Color processing
- Pattern recognition
- Video processing
- Microscopic Imaging

1.10 SUMMARY

In this chapter we learnt about basics of digital image processing like its structure components and conversion.

1.11 EXERCISES

1. Explain 2D image sampling.
2. Explain anatomy of human vision system.
3. Explain sampling and quantization.
4. Explain different element of image processing.
5. Discuss applications of image processing.

1.11 REFERENCES

1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009

2) Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008

3) Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. EsakkirajanAnd T. Veerakumar, 2016

❖❖❖❖

2

CONVOLUTION AND CORRELATION

Unit structure

2.0 OBJECTIVES

In this chapter we are going to learn about theory of signals and some mathematical concepts required in signal processing .

2.1 INTRODUCTION

Signals convey information. Systems transform signals. A signal can be, for example, a sequence of commands or a list of names. Mathematically, we model both signals and systems as functions. A signal is a function that maps a domain, often time or space, into a range, often a physical measure such as air pressure or light intensity. A system is a function that maps signals from its domain—its input signals—into signals in its range—its output signals. Both the domain and the range are sets of signals (signal spaces). Thus, systems are functions that operate on functions.

An example of one D signal is ECG, 2D signal is a still image.

2.2 2D SIGNALS

2D discrete signals are represented as : $x(n_1,n_2)$ where x is a real or complex value and n_1,n_2a pair of integers.

- 2D unit impulse sequence: it is given by $x(n_1,n_2) = \delta(n_1,n_2)$

Which is given by

$$\delta(n_1, n_2) = \begin{cases} 1, & n_1 = n_2 = 0 \\ 0, & else \end{cases}$$

- Line impulse : different types are :
  - Vertical line impulse
  - Horizontal line impulse
  - Diagonal impulse

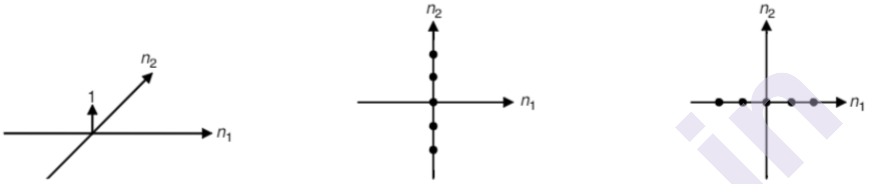


Figure 1 : 2D impulse sequence Vertical line impulse Horizontal Line Impulse

Figure 2 : Diagonal impulse function Diagonal function representation

- Exponential Sequence : they are defined by :

$$x(n_1,n_2) = a^{n_1} b^{n_2}, -\infty < n_1, n_2 < \infty$$

where a and b are complex numbers. When a and b have unit magnitude they can be written a

$$a = e^{j\omega_1}$$

$$\text{and} \quad b = e^{j\omega_2}$$

- Separable Sequence : A signal $x(n_1,n_2)$ is said to be separable if can be represented as product of the function n_1 alone and a function n_2 alone given as $x(n_1,n_2) = x_1(n_1)x_2(n_2)$

- Periodic Sequence : A 2d sequence is periodic if it repeats itself in a regulary spaced interval.hence a sequence is periodic if

$x(n_1,n_2) = x(n_1 + N_1,n_2) = x(n_1,n_2 + N_2)$ where N_1 and N_2 are positive integers

2.3 2D SYSTEMS

A 2D system is a device or algorithm that carries out some operation on a 2D signal.if$x(n_1,n_2)$ is input signal to the system and $y(n_1,n_2)$ is output of the system then the relation is : $y(n_1,n_2) = T[x(n_1,n_2)]$ where T is the transformation by the system to produce output from input.

Classification of 2D systems : It can be classified as :

1. **Linear Vs Non-Linear systems :** A linear system is the one that follows the laws of superposition. This law is necessary and sufficient condition to prove the linearity of the system.the Linearity is defines as :

$$T[ax_1(n_1, n_2) + bx_2(n_1, n_2)] = ay_1(n_1, n_2) + by_2(n_1, n_2)$$

Where a and b are scalar constants.

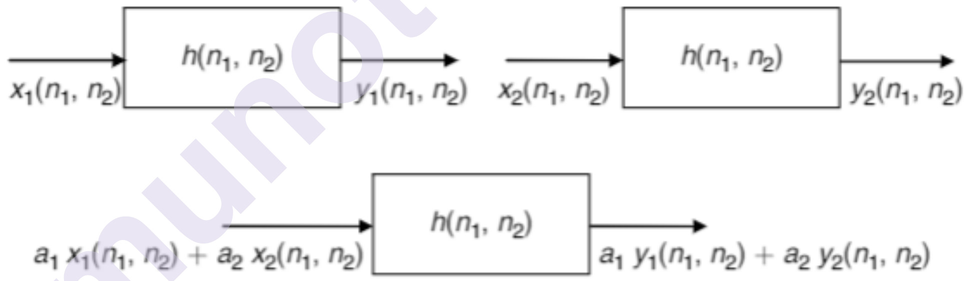


Figure 3 : Superposition Principle

2. **Shift Variant Vs shift -invariant systems :** If the input-output of the systems does not change with time then the system is said to be shift invariant. Mathematically it is given by :

$$T[x(n_1 - m_1, n_2 - m_2)] = y(n_1 - m_1, n_2 - m_2)$$

3. **Static Vs Dynamic system :** If the output of the systems at nay instant depends on most of the input sample but not on past and future samples of the input then the system is said to be static or memoryless. In any other case th system is said to be dynamic or of having memory.

4. **Stable system :** A 2D linear shift invariant system is stable if and only if its impulse response is absolutely summable given by :

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |h(n_1, n_2)| < \infty$$

2.4 2D CONVOLUTION

Convolution and correlation are used to extract information from images.They are linear and shift-invariant operations.Convolution is the relationship between a system's input signal, output signal, and impulse response.Shift-invariant means that we perform the same operation at every point in the image.Linear means that this operation is linear, that is, we replace every pixel with a linear

combination of its neighbors. These two properties make these operations very simple;it's simpler if we do the same thing everywhere, and linear operations are always thesimplest ones.Correlation is a way to identify a known waveform in a noisy background. Correlation is a mathematical operation much very similar to convolution. Just as with convolution, correlation uses two signals to produce a third signal. This third signal is called the cross-correlation of the two input signals. If a signal is correlated with itself, the resulting signal is instead called the autocorrelation.

Convolution has wide range of applications like image fitering , enhancement, restoration ,feature extraction and template matching.

The 2-dimensional discrete convolution between two signals $x[n_1,n_2]$ and $h[n_1,n_2]$ is given by :

$$y[n_1, n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2)h(n_1 - k_1, n_2 - k_2)$$

Convultions can be performed either in spatial domain or frequency domain.

2.4.1 2D CONVOLUTION THROUGH GRAPHICAL METHOD

In 2D convolution through graphical method following operations are performed :

- Folding
- Shifting and
- Addition

The process is performed as follows :

1. Given two matrices as input :x[n,m] and h[,m] ,first determine the dimension of the resultant matrix i.e if x[n,m] is matrix of order 2x3 and h[,m] is of order 3x1 then resultant matrix is of order 4x3

$$\text{Dimension of reslutant matrix} = \begin{cases} (\text{No. of rows of } x(m,n) + \text{No. of rows of } h(m,n) - 1) \\ \times \\ (\text{No. of columns of } x(m,n) + \text{No. of columns of } h(m,n) - 1) \end{cases}$$

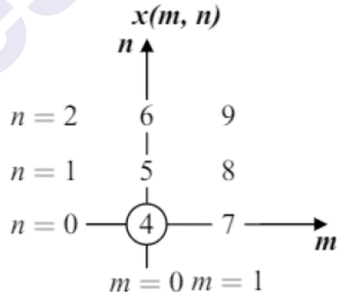
Dimension of resultant matrix $= (2 + 3 - 1) \times (3 + 1 - 1) = 4 \times 3$

2. The resultant matrix is calculated as :

The resultant matrix $y(m, n)$ of size 4×3 is given as

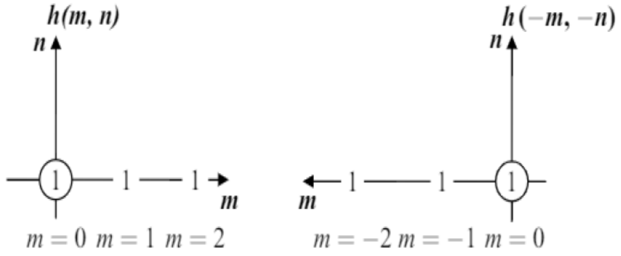
$$y(m,n) = \begin{pmatrix} y(0,0) & y(0,1) & y(0,2) \\ y(1,0) & y(1,1) & y(1,2) \\ y(2,0) & y(2,1) & y(2,2) \\ y(3,0) & y(3,1) & y(3,2) \end{pmatrix}$$

The graphical representation of $x(m, n)$ is shown below:



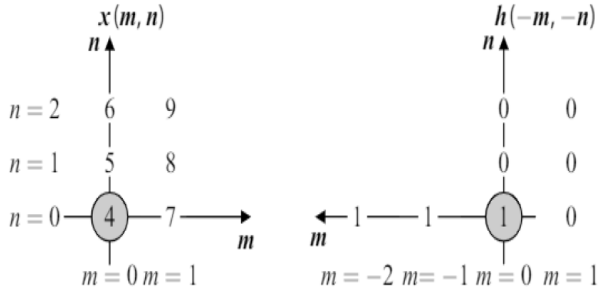
$m = 0 \; m = 1$

Here, the encircled element is the origin. The graphical representation of $h(m, n)$ and it's folded versic $h(-m, -n)$ are given below:



3. Determine the values y(0,0), y(0,1) etc.wherey(0,0) is obtained as :

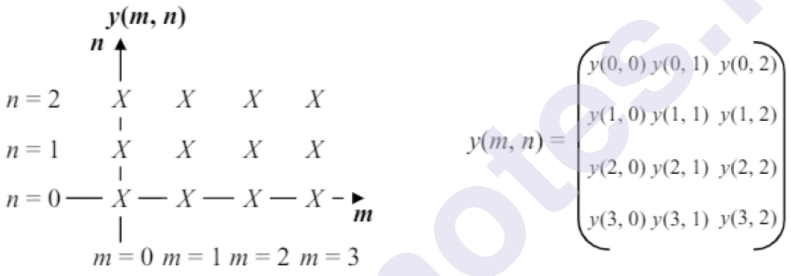
The common values of $x(m, n)$ and $h(-m, -n)$ are multiplied and then added to get the value of $y(0, 0)$. The shaded circle indicates the common area between the two signals.



The value $y(0, 0)$ is obtained as $y(0, 0) = 4 \times 1 = 4$.

Similary values for other elements.

4. The resulatant matrix is :



2.4.2 2D CONVOLUTION THROUGH Z TRANSFORM

The Z-transform is a mathematical tool which is used to convert the difference equations in discrete time domain into the algebraic equations in z-domain.

Mathematically, if x(n) is a discrete time function, then its Z-transform is defined as,

$$Z[x(n)] = X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

The convolution between two signals $x(n_1,n_2)$ and $h(n_1,n_2)$ is given by

$y(n_1,n_2) = x(n_1,n_2)$ ** $h(n_1,n_2)$ where ** indicated convolution

taking Z transform on both the sides we get

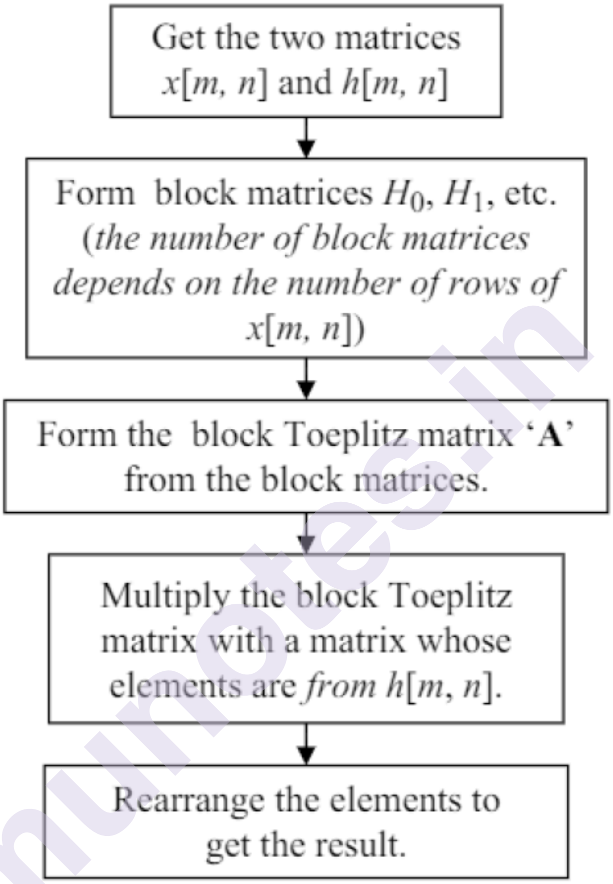
$Y(Z_1,Z_2) = X(Z_1,Z_2) \times H(Z_1,Z_2)$

Convolution is one domain is equal to multiplication in other domain.

2.4.3 2D CONVOLUTION THROUGH MATRIX ANALYSIS

2D convolution can be performed using matrix multiplication with the help of block Teoplitz matrix of one matrix and column matrix of the other .

The flowchart for the same is as given below :



The number of block matrices depend on number of rows of x[m,n]

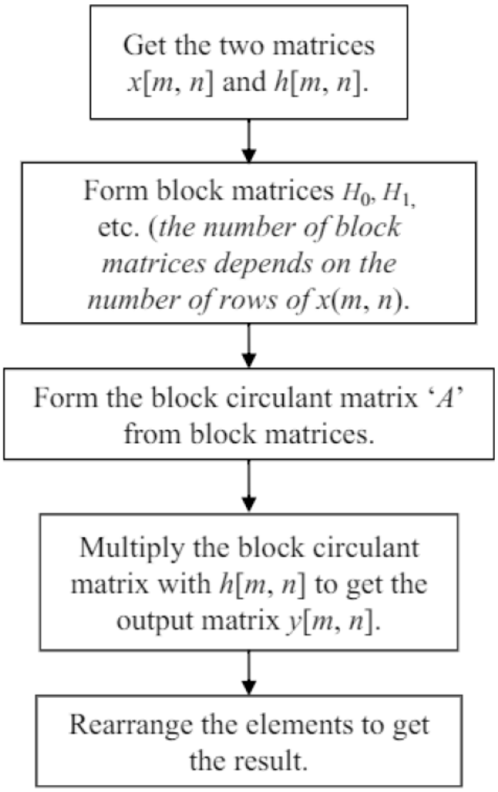
2.4.4 CIRCULAR CONVOLUTION

Circular convolution can be carried out for signals which are periodic in nature.

The convolution between two signals $x(n_1,n_2)$ and $h(n_1,n_2)$ is given by

$$y(n_1,n_2) = x(n_1,n_2) \otimes h(n_1,n_2)$$

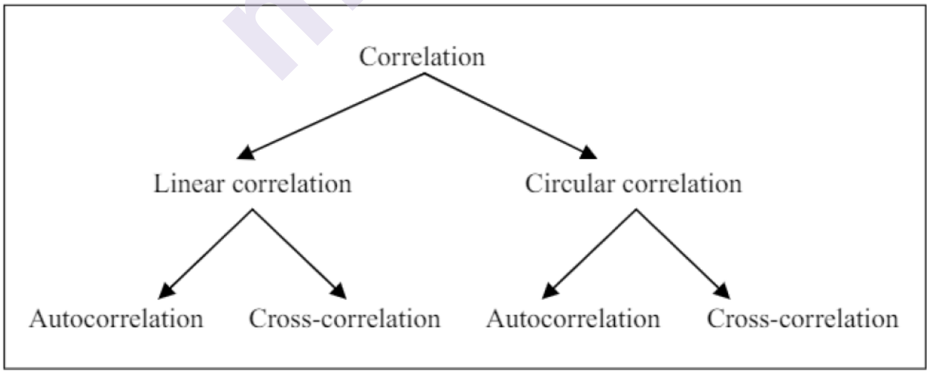
The flowchart for the same is as given below :



Circular correlation is widely used in zooming operation of digital cameras.

2.5 2D CORRELATION

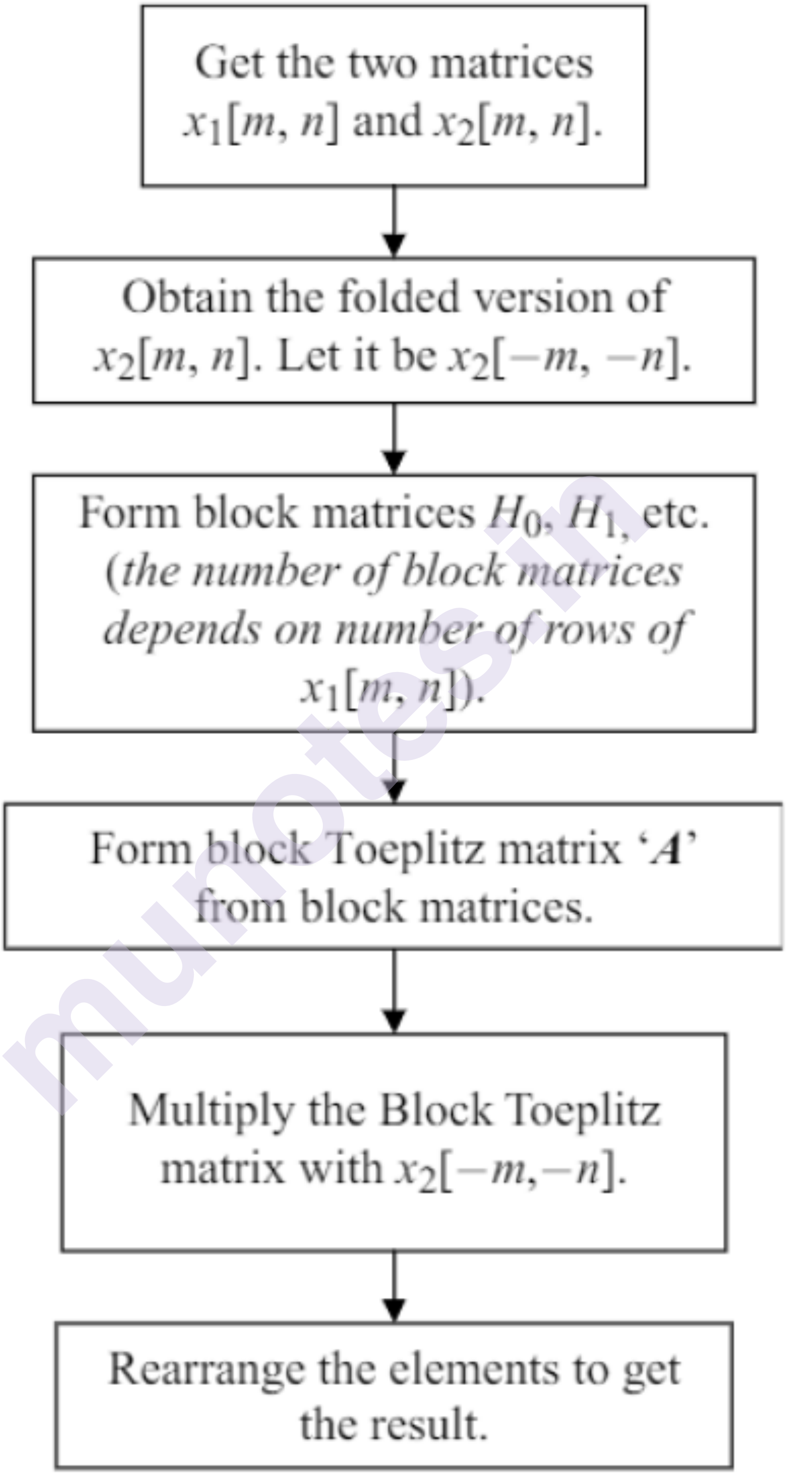
Correlation is a technique of signal matching .It is an important component of radar,sonar, digital communication and other systems.Mathematically similar to convolution it is of following types :



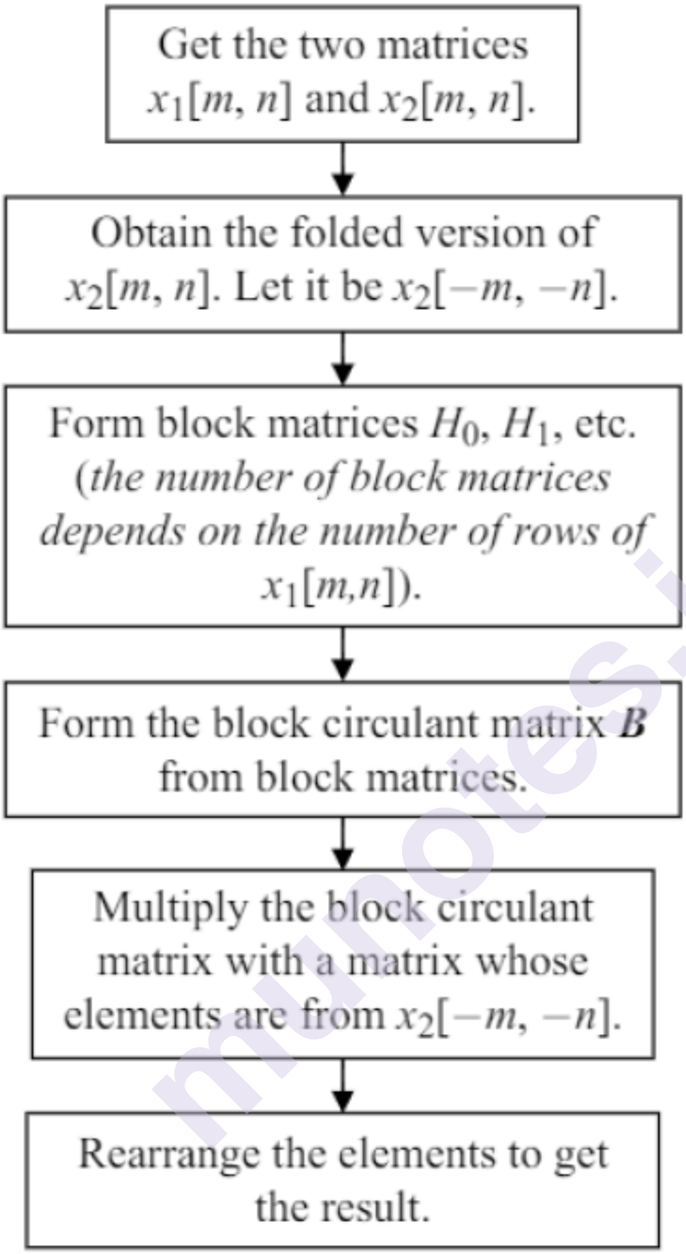
Correlation can be performed using :

- Matrix methods : with the help of block Teoplitz matrix and block Circulant matrix in terms of convolution

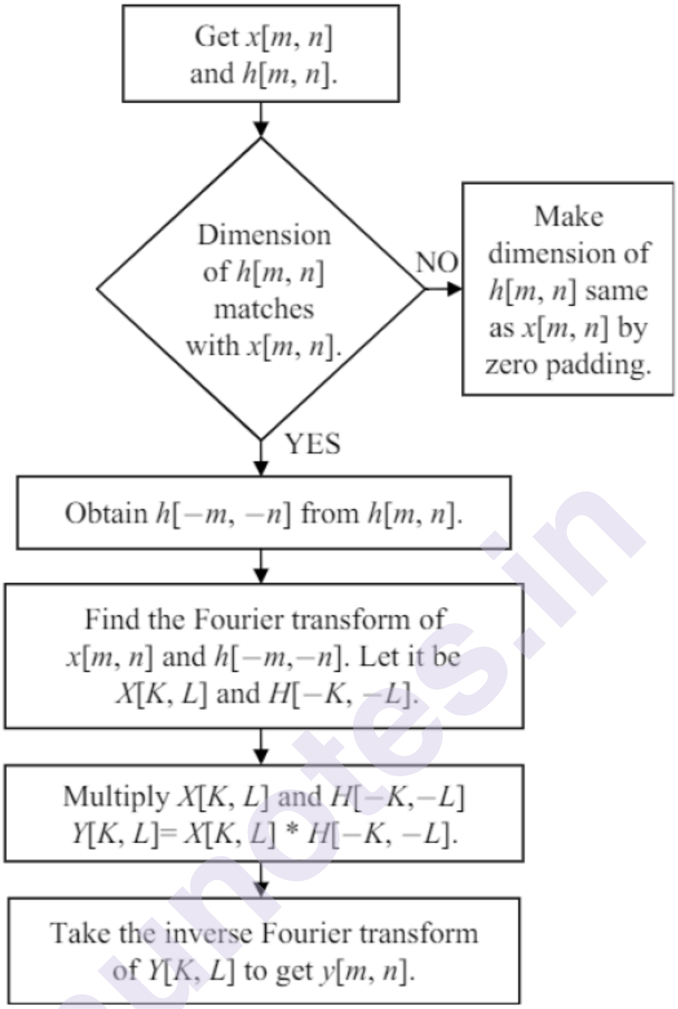
The flowchart for the same is as given below :



Circular correlation can be performed through matrix method as follows :



Circular correlation can be performed through transform method as follows :



2.6 SUMMARY

In this chapter we learnt about convolution that can be carried out for filtering process, its types and methods to perform.we also learnt about correlation which is used to find the similarity between images or part of images.

2.7 EXERCISES

1. Write a note on 2d signals.
2. Explain 2D systems and its classifications.
3. Define convolution . Explain its types of representation.
4. Write a note on cicular convolution.
5. Discuss applications ofCircular Convolution
6. Write a note on Correlation.

2.7 REFERENCES

1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009

2) Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008

3) Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. EsakkirajanAnd T. Veerakumar, 2016

❖❖❖❖

3

IMAGE TRANSFORMS

Unit structure

3.0 OBJECTIVES

In this chapter we are going to learn about image transforms which are widely used in image analysis and image processing .Transform is a mathematical tool basically used to move from one domain to another i.e from time domain to frequency domain . they are useful for fast computation of correlation and convolution.

3.1 INTRODUCTION

An image transform can be applied to an image to convert it from one domain to another. Viewing an image in domains such as frequency or Hough space enables the identification of features that may not be as easily detected in the spatial domain. Common image transforms include:

- Hough Transform, used to find lines in an image
- Radon Transform, used to reconstruct images from fan-beam and parallel-beam projection data
- Discrete Cosine Transform, used in image and video compression
- Discrete Fourier Transform, used in filtering and frequency analysis
- Wavelet Transform, used to perform discrete wavelet analysis, denoise, and fuse images

3.2 NEED FOR TRANSFORM

The main purpose of the transform can be divided into following 5 groups:

1. **Visualization:** The objects which are not visible, they are observed.
2. **Image sharpening and restoration:** It is used for better image resolution.
3. **Image retrieval:** An image of interest can be seen
4. **Measurement of pattern:** In an image, all the objects are measured.
5. **Image Recognition:** Each object in an image can be distinguished.

Transform is a mathematical toolto transform a signa. Hence it is required for :

- Mathematical convenience

- To extract more information .

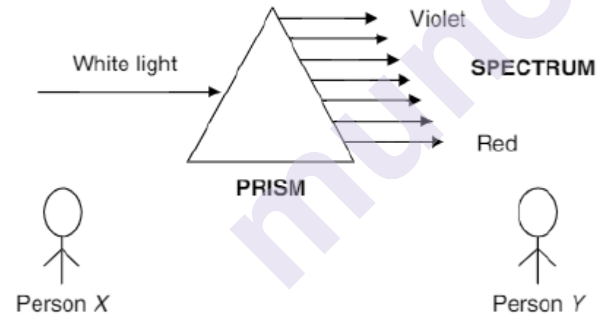


Fig. 4.1 *Spectrum of white light*

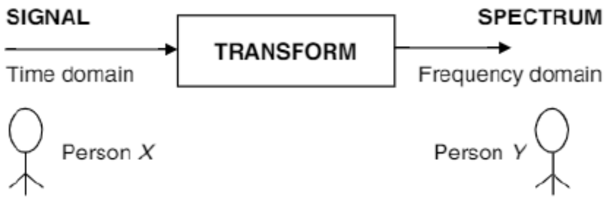


Figure 1 : Transformation Concept

3.3 IMAGE TRANSFORMS

Image transfor is represenatation of image . it is done for following two reasons :

1. To isolate critical component of image so that they are accessible directly.
2. It will make the image data compact so that they can be efficiently stored and transmitted.

Different types of image transforms that wil be discussed are :

- Fourier transform
- Walschtransform
- Hadamardtransform
- Discrete Cosine transform
- Wavelet transform etc

Classification of image transforms : they can be classified on the basis of nature of fuctions as follows :

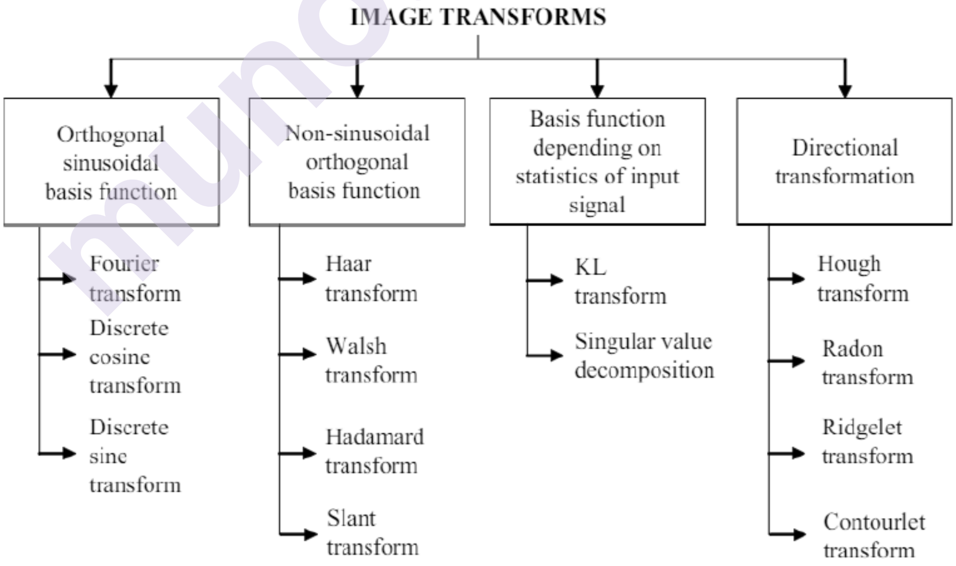


Figure 2 :Classification of Image transforms

3.4 FOURIER TRANSFORM

The Fourier transform states that that the non periodic signals whose area under the curve is finite can also be represented into integrals of the sines and cosines after being multiplied by a certain weight.

The Fourier transform is a representation of an image as a sum of complex exponentials of varying magnitudes, frequencies, and phases. The Fourier transform plays a critical role in a broad range of image processing applications, including enhancement, analysis, restoration, and compression.

The Fourier transform also has many wide applications that include , image compression (e.g JPEG compression) , filtrering and image analysis.

For a continuous time signal $x(t)$, the Fourier transform is defined as $X(\Omega)$

$$x(t) \xrightarrow{\text{CTFT}} X(\Omega)$$

A continuous Time Fourier Transfrom (CTFT) is defined as :

$$X(\Omega) = \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt$$

A continuous time signal x(t) is converted to discrete tie signal x(nT) using sampling process where T is the sampling interval.

$$x(t) \xrightarrow{\text{sampling}} x(nT)$$

The fourier transform of finite energy Discrete time signal is given by :

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(nT)\, e^{-j\Omega nt}$$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j(\Omega t)n}$$

where $X(e^{j\omega})$ is known as Discrete-Time Fourier Transform (DTFT) and is a continuous function of ω.
The relation between ω and Ω is given by

$$\omega = \Omega T$$

After a series of transformations :

The Discrete Fourier Transform (DFT) of a finite duration sequence $x(n)$ is defined a

$$X(K) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn}$$

where $k = 0, 1......, N-1$

3.5 2D DISCRETE FOURIER TRANSFORM

Functioning with the Fourier transform on a computer usually comprises a form of the transform called as the discrete Fourier transform (DFT). A discrete transform is a transform whose input and output values are discrete samples, making it convenient for computer manipulation. There are two principal reasons for using this form of the transform:

- The input and output of the DFT are both discrete, which makes it convenient for computer manipulations.
- There is a fast algorithm for computing the DFT known as the fast Fourier transform (FFT).

The 2D-DFT of a rectangular image $f(m, n)$ of size $M \times N$ is represented as $F(k, l)$

$$f(m, n) \xrightarrow{\text{2D-DFT}} F(k, l)$$

where $F(k, l)$ is defined as

$$F(k, l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j\frac{2\pi}{M}mk} e^{-j\frac{2\pi}{N}nl}$$

$$F(k, l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j\frac{2\pi}{M}nl} e^{-j\frac{2\pi}{N}mk}$$

The inverse 2D Discrete Fourier Transform is given by

$$f(m, n) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k, l) e^{j\frac{2\pi}{N}mk} e^{j\frac{2\pi}{N}nl}$$

The Fourier transform $F(k, l)$ is given by

$$F(k, l) = R(k, l) + jI(k, l)$$

where R(k,l) represents the real part of spectrum and l(k,l) the imaginary part.

3.6 PROPERTIES OF 2D-DFT

There are many properties of the transformation that give vision into the content of the frequency domain representation of a signal and allow us to manipulate singals in one domain or the other.

Following are some of the properties :

1. **Separable property :** It is computed in two steps by successive 1D operations on rows and columns of an image .

Thus, performing a 2D Fourier transform is equivalent to performing two 1D transforms as

(a) Performing a 1D transform on each row of image $f(m, n)$ to get $F(m, l)$
(b) Performing a 1D transform on each column of $F(m, l)$ to get $F(k, l)$

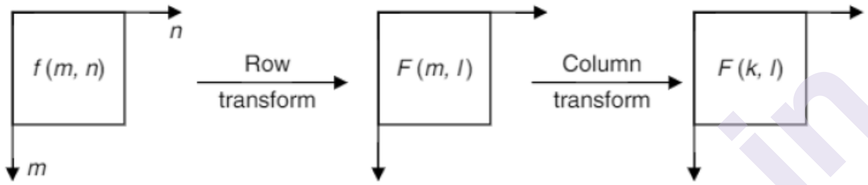


Figure 3 :Computation of 2D-DFT bySeparable Property

2. **Spatial Shift property:** Since both the space and frequency domains are considered periodic for the purposes of the transforms, shifting means rotating around the boundaries.

The 2D DFT of a shifted version of the image $f(m, n)$, i.e., $f(m - m_0, n)$ is given by

$$f(m-m_0, n) \xrightarrow{\text{DFT}} \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m-m_0, n)\, e^{-j\frac{2\pi}{N}mk}\, e^{-j\frac{2\pi}{N}nl}$$

3. **Periodicity Property :**

The 2D DFT of a function $f(m, n)$ is said to be periodic with a period N if

$$F(k, l) \rightarrow F(k+pN, l+qN)$$

4. **Convolution property :** *Convolution* is a mathematical tool for merging two signals to produce a third signal. In other words, the convolution can be defined as a mathematical operation that is used to express the relation between input and output an LTI system.convoltion in spatial domain is same as multiplication in frequency domain .

Convolution of two sequences $x(n)$ and $h(n)$ is defined as

$$x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

5. **Correlation property :** It is used to find relative similarity in between two signals.When we find similarity between of a signal to itself it is called as autocorrelation . On the other hand when we find similarity between two signals it is called as cross correlation. The cross correlation between two signals is same as performing convolution of one sequence with the folded version of other sequence.Thus this property says that correlation of two sequneces in time domain is same as multiplication of DFT of one sequence and time reversal of DFT of another sequence in frequency domain.

6. **Scaling property :** Just as in one dimension, shrinking in one domain causes expansion in the other for the 2D DFT. This means that as an object grows in an image, the corresponding features in the frequency domain will expand.Sclaing is used mainly to shrink or expand the size of an image.

The 2D DFT of a function $f(m, n)$ is defined as

$$f(m,n) \xrightarrow{\text{DFT}} F(k,l)$$

If DFT of $f(m, n)$ is $F(k.l)$ then $\text{DFT}\,[f(am, bn)] = \frac{1}{|ab|} F(k/a, l/b)$

7. **Conjugate symmetry :**

If the DFT of $f(m, n)$ is $F(k, l)$ then the DFT $[f^*(m, n)] = F^*(-k, -l)$

$$F(k, l) = F^*(-k, -l)$$

8. **Orthogonality property :**

The orthogonality property of a 2D DFT is given as

$$\frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_{k,l}(m,n)\, a^*_{k',l'}(m,n) = \delta(k-k', l-l')$$

9. **RoatationProperty :** It states that if a function is rotated y an angle then its Fourier transform also rotates by ana equal amount.

$$f(m,n) \rightarrow f(r\cos\theta, r\sin\theta)$$
$$\text{DFT}[f(r\cos\theta, r\sin\theta)] \rightarrow F[R\cos\Phi, R\sin\Phi]$$
$$\text{DFT}[f(r\cos(\theta+\theta_0), r\sin(\theta+\theta_0))] \rightarrow F[R\cos(\Phi+\Phi_0), R\sin(\Phi+\Phi_0)]$$

10. Multiplication by Exponent:

If the DFT of $f(m, n)$ is $F(k, l)$ then

$$\text{DFT}[e^{j\frac{2\pi}{N}mk_0} e^{j\frac{2\pi}{N}nl_0} f(m,n)] = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}mk_0} e^{j\frac{2\pi}{N}nl_0} f(m,n) e^{-j\frac{2\pi}{N}mk} e^{-j\frac{2\pi}{N}nl}$$

All the above properties in nutshell are as follows :

Property	*Sequence*	*Transform*
Spatial shift	$f(m-m_0, n)$	$e^{-j\frac{2\pi}{N}m_0 k} F(k,l)$
Periodicity	–	$F(k+pN, l+qN) = F(k, l)$
Convolution	$f(m, n)*g(m, n)$	$F(k, l) \times G(k, l)$
Scaling	$f(am, bn)$	$\frac{1}{\lvert ab \rvert} F(k/a, l/b)$
Conjugate symmetry		$F(k, l) = F^*(-k, -l)$
Multiplication by exponential	$e^{j\frac{2\pi}{N}mk_0} e^{j\frac{2\pi}{N}ml_0} f(m,n)$	$F(k-k_0, l-l_0)$
Rotation property	$f(r\cos(\theta+\theta_0), r\sin(\theta+\theta_0))$	$F[R\cos(\Phi+\Phi_0), R\sin(\Phi+\Phi_0)]$
Folding property	$f(-m, -n)$	$F(-k, -l)$

Table 1 : Properties of DFT

3.7 OTHER TRANSFORMS

Some other transforms used in image processing are :

1. **Walsh Transform :** The Walsh-Hadamard transform is a non-sinusoidal, orthogonal transform widely used in signal and image processing. In this transform, the signal is decomposed into a set of basis functions (similar to harmonics in Fourier).

The one-dimensional Walsh transform basis can be given by the following equation:

$$g(n,k) = \frac{1}{N}\prod_{i=0}^{m-1}(-1)^{b_i(n)b_{m-1-i}(k)}$$

The two-dimensional Walsh transform of a function $f(m, n)$ is given by

$$F(k,l) = \frac{1}{N}\sum_{m}\sum_{n} f(m,n)\prod_{i=0}^{p-1}(-1)^{\left[b_i(m)b_{p-1-i}(k)+b_i(n)b_{p-1-i}(k)\right]}$$

$i=0$

Here n represents the time index, k represents the frequency index and N represents the order. Also, m represents the number bits to represent a number and $b_i(n)$ represents the i^{th} (from LSB) bit of the binary value, of n decimal number represented in binary. The value of m is given by $m = \log_2 N$.

2. **Hadamard Transform:** Same as Walsh transform with the difference that rows of the transform matrix are re-ordered.The elements of mutually orthogonal basis vectors of a Hadamard transform are either +1 or -1.

3. **HAAR transform:** Haar wavelet compression is an efficient way to perform both lossless and lossy image compression. It relies on averaging and differencing values in an image matrix to produce a matrix which is sparse or nearly sparse. A sparse matrix is a matrix in which a large portion of its entries are 0.it is based on class of orthogonal matrices whose elements are either +1 , -1 or 0 multiplied by powers of $\sqrt{2}$.Algorithm to generate HAAR basis is :

Step 1 Determine the order of N of the Haar basis.
Step 2 Determine n where $n = \log_2 N$.
Step 3 Determine p and q.

(i) $0 \le p < n-1$
(ii) If $p = 0$ then $q = 0$ or $q = 1$
(iii) If $p \ne 0$, $1 \le q \le 2^p$

Step 4 Determine k.

$$k = 2^p + q - 1$$

Step 5 Determine Z.

$$Z \rightarrow [0,1) \Rightarrow \left\{\frac{0}{N}, \frac{1}{N}, \cdots \frac{N-1}{N}\right\}$$

Step 6

$$\text{If} \quad k = 0 \quad \text{then} \quad H(Z) = \frac{1}{\sqrt{N}}$$

Otherwise,

$$H_k(Z) = H_{pq}(Z) = \frac{1}{\sqrt{N}} \begin{cases} +2^{p/2} & \frac{(q-1)}{2^p} \le Z < \frac{(q-1/2)}{2^p} \\ -2^{p/2} & \frac{(q-1/2)}{2^p} \le Z < \frac{q}{2^p} \\ 0 & \text{otherwise} \end{cases}$$

4. **Slant Transform:** A new unitary transform called the slant transform, specifically designed for image coding, was developed by Enomoto and Shibata. The transformation possesses a discrete sawtoothlike basis vector which efficiently represents linear brightness variations along an image line. The slant transformation has been utilized in several transform image-coding systems for monochrome and color images

5. **Discrete Cosine Transform:** It is a transform that is mainly used in compression algorithms. It transforms data points in a spatial domain into a frequency domain. This makes it easier to find the repetition of patterns. Like any other transform, it is also invertible. This means we can return the actual data points if the transforms are given.It was developed by Ahmad, Natrajan and Rao in 1974.

If $x[n]$ is the signal of length N, the Fourier transform of the signal $x[n]$ is given by $X[k]$ where

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}}$$

Where k varies from 0 to N-1

6. **KL Transform:** Named after Karim Karhunen and Michel Loeve Transform (KLT), it is also known as the **Hotelling trans- form or Principal Component Analysis (PCA).** It is based on the statistical properties of the image and has several important properties that make it useful for image processing particularly for image compression.The main purpose of image compression is to store the image in fewer bits as compared to original image, now data from neighboring pixels in an image are highly correlated.More image compression can be achieved by de-correlating this data. The KL transform does the task of de-correlating the data thus facilitating higher degree of compression.It is used in clustering analysis and image compression.There are four major steps in order to find the KL transform :-

a. Find the mean vector and covariance matrix of the given image x.

b. Find the Eigen values and then the eigen vectors of the covariance matrix

c. Create the transformation matrix T, such that rows of T are eigen vectors

d. Find the KL Transform

Comparison of various image transforms

S No	*Transform name*	*Basis*	*Parameters to be computed*	*Computational complexity*
1	Fourier transform	Complex exponential	Transform coeffients	$2N^2 \log_2{}^N$ (complex)
2	Walsh transform	Walsh basis is either + 1 or − 1	Transform coeffients	$2N^2 \log_2{}^N$ (additions)
3	Slant transform		Transform coeffients	$2N^2 \log_2{}^N$
4	Haar transform		Transform coeffients	$2(N-1)$
5	Cosine transform	Cosine function	Transform coeffients	$4 N^2 \log_2{}^{2N}$
6	Karhunen–Loeve transform	Eigen vector of the covariance matrix	Transform coeffients	N^3

Table 2 :Comparison of Transforms

3.8 SUMMARY

In this chapter we studied why image transforms are required in digital image processing , the different types available .

3.9 EXERCISES

1. Why we need image transforms
2. What are different types of transforms?
3. Explain properties of DFT.
4. What is main difference between Walsh and Hadamard transform?
5. Explain Fourier transform and its properties .
6. What are advantages of Walsh over Fourier transform?

3.10 REFERENCES

1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, Tata McGraw-Hill Education Pvt. Ltd., 2009

2) Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008

3) Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. EsakkirajanAnd T. Veerakumar, 2016

❖❖❖❖

4

IMAGE ENHANCEMENT

Unit Structure

4.0 OBJECTIVES

- To understand the image enhancement in spatial and frequency domain
- To get familiar with point and mask operations in spatial domain
- To learn different types of transformations on an image and the filtering operations

4.1 INTRODUCTION

The goal of image enhancement is to make it easier for viewers to understand the information contained in images. A better-quality image is produced by an enhancement algorithm when it is used for a certain application. This can be accomplished by either reducing noise or boosting image contrast.

For presentation and analysis, image-enhancement techniques are used to highlight, sharpen, or smooth visual characteristics. Application-specific enhancement techniques are frequently created via empirical research. Image enhancement techniques highlight particular image elements to enhance how an image is perceived visually.

Image-enhancement methods can be divided into two groups in general, including

(1) A method in the spatial domain; and (2) A method in the transform domain.

Whereas the transform domain approach works with an image's Fourier transform before transforming it back into the spatial domain, the spatial domain method operates directly on pixels. Because they are quick and easy to use, histogram-based basic improvement approaches can produce results that are acceptable for some applications. By removing a portion of a filtered component from the original image, unsharp masking sharpens the edges. Unsharp masking is a technique that has gained popularity as a diagnostic aid.

4.2 IMAGE ENHANCEMENT IN SPATIAL DOMAIN

The alteration of pixel values is the focus of the spatial domain approach. The three basic categories of the spatial domain technique are i) point operation, (ii) mask operation, and (iii) global operation.

4.2.1 Point operation

In point operation, each pixel is modified by an equation that is not dependent on other pixel values. The point operation is illustrated in Figure 1. The point operation is represented by

$$g(m, n) = T[f(m, n)]$$

In point operation, T operates on one pixel, or there exists a one-to-one mapping between the input image f (m, n) and the output image g (m, n).

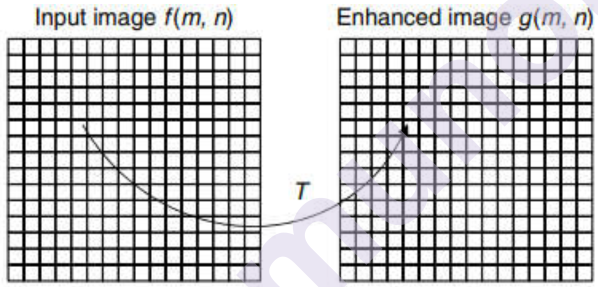


Figure 1: Point operation

4.2.2 Mask operation

As shown in Figure 2, each pixel is altered during the mask operation in accordance with the values in a small neighbourhood. Spatial low-pass filtering with a box filter or median filter are examples of mask operations. In \smask operation, the operator T operates on the neighbourhood of pixels.Here, mask is a little matrix whose elements are frequently referred to as weights. Every mask has a history. A symmetric mask's roots are typically found in its centre pixel position. Depending on the desired usage, any pixel location may be used as the origin for non-symmetric masks.

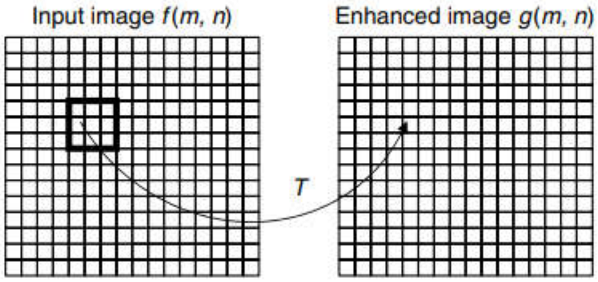


Figure 2: Mask processing

4.2.3 Global operation

All of the image's pixel values are taken into account during global operation. Frequency domain operations are often global operations.

4.3 ENHANCEMENT THROUGH POINT OPERATIONS

Each pixel value is mapped to a new pixel value in point operation. Point operations essentially have no memory. The enhancement at any position in a point operation depends only on the image value there.The point operation, which is shown in Figure 3, converts the input picture f (m, n) into the output image g (m, n).It is clear from the figure that every f (m, n) pixel with the same grey level maps to a single grey value in the final image.

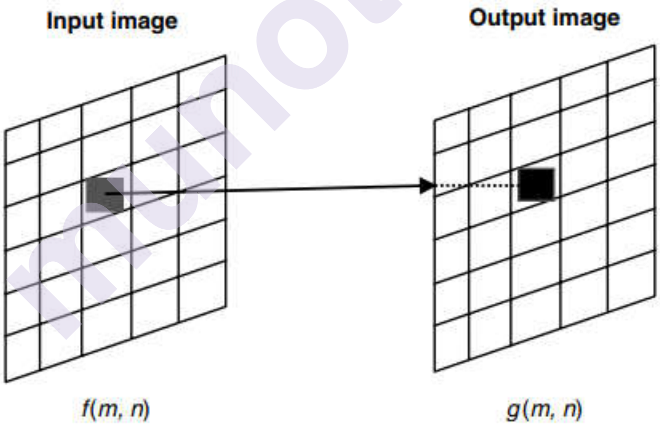


Figure 3: Illustration of point operation

4.3.1 Types of point operations

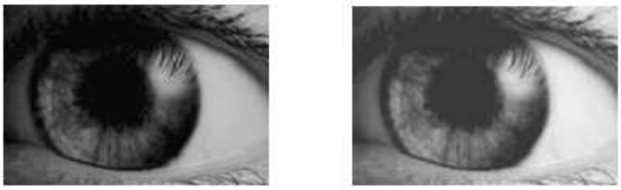
Some of the examples of point operation include (i) brightness modification, (ii) contrast manipulation, and (iii) histogram manipulation

1] Brightness modification

The value assigned to each pixel in an image determines how bright it is. A constant is either added to or subtracted from the luminance of each sample value to alter the brightness of the image. By adding a constant value to each and every pixel in the image, the brightness can be raised.The brightness can also be reduced by deducting a fixed amount from each and every pixel in the image.

- Increasing the brightness of an image: A simple method to increase the brightness value of an image is to add a constant value to each and every pixel of the image. If f [m, n] represents the original image then a new image g [m, n] is obtained by adding a constant k to each pixel of f [m, n].This is represented by

g[m, n] = f [m, n] + k

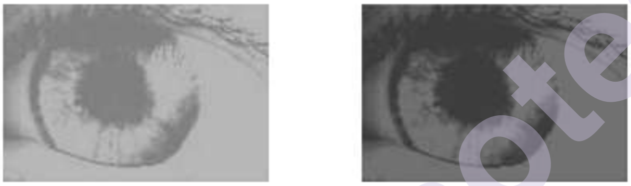


(a) Original image (b) Brightness-enhanced image

Figure 4: Brightness enhancement

- Decreasing the brightness of an image: The brightness of an image can be decreased by subtracting a constant k from all the pixels of the input image f [m,n]. This is represented by

g [m, n] = f [m, n] – k



(a) Original image (b) Brightness-suppressed image

Figure 5: Brightness suppression

2] Contrast adjustment

Contrast adjustment is done by scaling all the pixels of the image by a constant k. It is given by

g [m, n] = f [m, n] ∗ k

Changing the contrast of an image, changes the range of luminance values present in the image

Specifying a value above 1 will increase the contrast by making bright samples brighter and dark samples darker, thus expanding on the range used. A value below 1 will do the opposite and reduce a smaller range of sample values. An original image and its contrast-manipulated images are illustrated in Figure 6.

Figure 6: Contrast manipulation

4.4 HISTOGRAM MANIPULATION

Histogram manipulation generally involves changing the histogram of an input image to enhance the image's visual appeal. One must have a foundational understanding of the histogram of the image in order to comprehend histogram manipulation.

A] Histogram: The gray-level values of an image are plotted against the number of instances of each type of grey in the image to create the histogram. A useful summary of an image's intensities is provided by the histogram, but it is unable to transmit any knowledge of the spatial connections between pixels. Further information on image contrast and brightness is given by the histogram.

1. The lowest grey levels will be concentrated in the histogram of a dark image.
2. A bright image will have a greater grey level cluster in the histogram.
3. The histogram won't be evenly distributed for a low-contrast image; instead, it will be narrow.
4. The histogram will have an equal spread in the grey level for a high-contrast image.

Image brightness may be improved by modifying the histogram of the image.

B] Histogram equalisation: In order to distribute the grey levels of an image uniformly across its range, a procedure called equalisation is used. Based on the image histogram, histogram equalisation reassigns the brightness values of pixels. The goal of histogram equalisation is to produce a picture with a histogram that is as flat as feasible. More aesthetically acceptable outcomes are produced by histogram equalisation across a wider spectrum of photos.

C] Procedure to perform Histogram equalization : Histogram equalisation is done by performing the following steps:

1. Find the running sum of the histogram values.

2. Normalise the values from Step (1) by dividing by the total number of pixels.

3. Multiply the values from Step (2) by the maximum gray-level value and round.

4. Map the gray level values to the results from Step (3) using a one-to-one correspondence.

Example: Perform histogram equalisation on the following image:

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Solution: The maximum value is found to be 5. We need a minimum of 3 bits to represent the number. There are eight possible gray levels from 0 to

7. The histogram of the input image is given below:

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0

Step 1:Compute the running sum of histogram values. The running sum of histogram values is otherwise known as cumulative frequency distribution

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running sum	0	0	0	6	20	25	25	25

Step 2: Divide the running sum obtained in Step 1 by the total number of pixels. In this case, the total number of pixels is 25.

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running sum	0	0	0	6	20	25	25	25
Running Sum/Total number of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25

Step 3:Multiply the result obtained in Step 2 by the maximum gray-level value, which is 7 in this case

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Runningsum / Total number of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25
Multiply the above result by maximum gray level	$\frac{0}{25}*7$	$\frac{0}{25}*7$	$\frac{0}{25}*7$	$\frac{6}{25}*7$	$\frac{20}{25}*7$	$\frac{25}{25}*7$	$\frac{25}{25}*7$	$\frac{25}{25}*7$

The result is then rounded to the closest integer to get the following table:

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Running Sum/Total number of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25
Multiply the above result by maximum gray level	0	0	0	2	6	7	7	7

Step 4: Mapping of gray level by a one-to-one correspondence:

Original gray level	Histogram equalised values
0	0
1	0
2	0
3	2
4	6
5	7
6	7
7	7

The original image and the histogram equalised image are shown side by side.

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \xrightarrow[\text{Equalisation}]{\text{Histogram}} \begin{bmatrix} 6 & 6 & 6 & 6 & 6 \\ 2 & 6 & 7 & 6 & 2 \\ 2 & 7 & 7 & 7 & 2 \\ 2 & 6 & 7 & 6 & 2 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Original image **Histogram equalised image**

4.5 LINEAR GRAY LEVEL TRANSFORMATION

An image's linear transformation is a function that converts each pixel's grayscale value into a different grayscale at the same location using a linear function. The linear transformation is represented by

The linear transformation is given by g(m, n) = T [f (m, n)].

Image negative or inverse transformation:

Light and dark are turned around in the inverse transformation. An image negative is an illustration of an inverse transformation. Each pixel is subtracted from the highest pixel value to create a negative image. The negative picture for an 8-bit image can be created by reversing the scaling of the grey levels according to the transformation

$$g(m, n) = 255 - f(m, n)$$

The graphical representation of negation is shown figure 7.

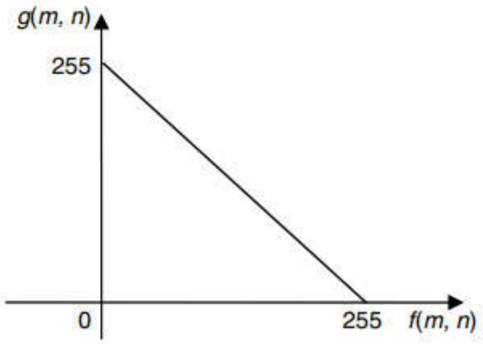


Figure 7: Graphical representation of negations

Negative images are useful in the display of medical images and producing negative prints of images.

4.6 NONLINEAR GRAY LEVEL TRANSFORMATION

Non-linear transformation maps small equal intervals into non-equal intervals. Following are few of the non-linear transformations:

4.6.1 Thresholding

To extract a portion of an image that contains all the information, thresholding is necessary. The problem of segmentation in general includes thresholding. Hard thresholding and soft thresholding are two major categories for thresholding.The procedure of hard thresholding entails reducing coefficients with absolute values below the threshold to zero. Another technique is soft thresholding, which shrinks the nonzero coefficients towards zero by first setting to zero coefficients whose absolute values are below the threshold.

4.6.2 Gray-level slicing

The purpose of gray-level slicing is to highlight a specific range of gray values. Two different approaches can be adopted for gray-level slicing

(a) Gray-level Slicing without Preserving Background This displays high values for a range of interest and low values in other areas. The main drawback of this approach is that the background information is discarded.

(b) Gray-level Slicing with Background In gray-level slicing with background, the objective is to display high values for the range of interest and original gray level values in other areas. This approach preserves the background of the image.

4.6.3 Logarithmic transformation

The logarithmic transformation is given by

$g(m, n) = c\log_2(f(m, n) + 1)$

This type of mapping spreads out the lower gray levels. For an 8-bit image, the lower gray level is zero and the higher gray level is 255. It is desirable to map 0 to 0 and 255 to 255. The function $g(m, n) = c\log(f(m, n) + 1)$ spreads out the lower gray levels.

4.6.4 Exponential transformation

In multiplicative filtering processes, exponential transformation has primarily been utilized in conjunction with logarithmic transformation. An exponential transfer function has the effect of extending high-contrast edges while contracting low-contrast edges in an image. This is not a desired enhancement transformation because it typically results in images with less visible detail than the original.

4.6.5 Power law transformation

A physical device like a CRT does not produce light with a linear relationship to the input signal.The intensity generated at the display's surface is about the applied voltage raised by a factor of 2.5. Gamma is the name given to the numerical value of this power function's exponent. To obtain accurate intensity reproduction, this non-linearity must be adjusted.

The formula for the power law transformation is

$$g(m, n) = [f(m, n)]^{\gamma}$$

where f(m, n) is the input image and g(m, n) is the output image. Gamma (γ) can take either integer or fraction values.

4.7 LOCAL OR NEIGHBORHOOD OPERATION

Neighborhood operation modifies an image's pixels based on how the pixels in their immediate vicinity work. The term "convoluting a mask with an image" is frequently used to describe linear spatial filtering.

Convolution masks or convolution kernels are other names for the filter masks.

4.7.1 Spatial filtering

Using spatial filtering, the original picture pixel value that corresponds to the kernel's centre is replaced with the sum of the original pixel values in the region that the kernel corresponds to, multiplied by the kernel weights.

4.7.2 Linear filtering

Each pixel in the input image is replaced with a linear combination of brightness of nearby pixels in linear filtering. In other words, each pixel value in the output image represents a weighted sum of the pixels nearby the corresponding pixel in the input image. A picture can be both sharpened and smoothed via linear filtering. Using a convolution mask, one can create a spatially invariant linear filter.The linear filter is spatially variable if various filter weights are applied to various portions of the image.

4.7.3 Mean filter/ Average filter/ Low pass filter

The average of all the values in the immediate area is used to replace each pixel with the mean filter. The degree of filtration is determined by the size of the neighbourhood. Each pixel in a spatial averaging operation is replaced by the weighted average of the pixels in its immediate vicinity. The low-pass filter removes the sharp changes that cause blurring and keeps the smooth area of the image. The following is the 3 by 3 spatial mask that may execute the averaging operation:

$$\text{3 by 3 low-pass spatial mask} = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Similarly, the 5 by 5 averaging mask} = \frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

It should be noted that for a low-pass spatial mask, the elements' sum is equal to 1. The larger the mask becomes, the more blurring there will be. In order to precisely find the core pixel, the mask's size is typically unusual.

4.7.4 Weighted average filter

The mask of a weighted average filter is given by

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

The weighted average filter gets its name because it is clear from the mask that the closest pixels to the centre are weighted more heavily than the farthest pixels.The pixel that has to be updated is changed by multiplying the value of the neighbouring pixel by the matrix's weights and dividing the result by the sum of the matrix's coefficients.

4.7.5 Bartlett filter

In the spatial domain, the bartlett filter is a triangle-shaped filter. A bartlett filter can be created by taking the product of two box filters in the frequency domain or by convolving two box filters in the spatial domain.

The 3 × 3 box filter is given by $\frac{1}{9}\times\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$. From this, a Barlett window in the spatial domain is given by

$$\text{Bartlett window} = \frac{1}{9}\times\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * \frac{1}{9}\times\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{8}\times\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}.$$

4.7.6 Gaussian filter

A class of linear smoothing filters known as "Gaussian filters" select their weights based on the characteristics of a Gaussian function. The Gaussian kernel is frequently employed for smoothing. The formula for the Gaussian filter in continuous space is

$$h(m, n) = \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{m^2}{2\sigma^2}}\right) \times \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{n^2}{2\sigma^2}}\right)$$

The expression above demonstrates the separability of a Gaussian filter. A extremely effective filter for eliminating noise derived from a normal distribution is the Gaussian smoothing filter. A particular type of averaging called Gaussian smoothing uses a 2D Gaussian as its kernel.

4.8 MEDIAN FILTER

Similar to the mean filter, the median filter in image processing is typically used to minimize noise in an image. To preserve relevant detail in the image, it frequently performs better than the mean filter.

Median Filter is a simple and powerful non-linear filter.

- It is employed to lessen the degree of intensity difference between adjacent pixels.
- With this filter, the pixel value is swapped out for the median value.
- In order to calculate the median, all the pixel values are sorted first into ascending order, and then the middle pixel value is substituted for the pixel being calculated.

In order to determine if a pixel is typical of its surroundings, the median filter examines each pixel in turn and looks at its close neighbours. The median of those values is used to replace the pixel value rather than just the mean of its neighbours' pixel values. The median is derived by placing the pixel under consideration in the middle of the neighborhood's pixel values, which have been sorted into numerical order. The average of the two middle pixel values is used (if the neighbourhood under consideration has an even number of pixels).

Example: Compute the median value of the marked pixel using 3x3 mask

$$\begin{bmatrix} 1 & 5 & 7 \\ 2 & \textcircled{4} & 6 \\ 3 & 2 & 1 \end{bmatrix}$$

Solution The median value of the marked pixel is computed as follows:

Step 1 First, the pixel values are arranged in ascending order as follows:

1 1 2 2 3 4 5 6 7

Step 2 The median value of the ordered pixel is computed as follows:

~~1~~ ~~1~~ ~~2~~ ~~2~~ 3 ~~4~~ ~~5~~ ~~6~~ ~~7~~

The median value is computed to be 3. Then, the original pixel value of 4 will be replaced by the computed median value of 3.

$$\begin{bmatrix} 1 & 5 & 7 \\ 2 & \textcircled{4} & 6 \\ 3 & 2 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 5 & 7 \\ 2 & \boxed{3} & 6 \\ 3 & 2 & 1 \end{bmatrix}$$

Original image data   After median filtering

4.9 SPATIAL DOMAIN HIGH PASS FILTERING

Image sharpening's primary goal is to draw attention to the image's minute features. The high-frequency components are enhanced through image sharpening. The spatial filter or spatial mask which performs image sharpening is given below:

The spatial mask which performs image sharpening is given as $\frac{1}{9}\times\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

The sum of all the weights is zero; this implies that the resulting signal will have zero dc value.

4.9.1 High boost filtering

High-frequency emphasis filter is another name for a high-boost filter. To help with visual interpretation, a high-boost filter is utilized to keep some of the low-frequency components. Before deleting the low-pass image in high-boost filtering, the input image f(m, n) is amplified by an amplification factor A.

The expression for the high-boost filter then becomes

High boost = A x f(m,n) - low pass

Adding and subtracting 1 with the gain factor, we get

$$\text{High boost} = (A-1)\times f(m,n) + f(m,n) - \text{low pass}$$

$$\text{But } f(m,n) - \text{low pass} = \text{high pass}$$

Thus,

$$\text{high boost} = (A-1)\times f(m,n) + \text{high pass}$$

4.9.2 Unsharp masking

One of the methods frequently used for edge enhancement is unsharp masking. This method tips the balance of the image towards the sharper material by subtracting the smoothed version of the image from the original image. Below is a description of how to accomplish unsharp masking:

1. Apply an image blur filter.

2. Subtract from the original image the result from Step 1.

3. Add a weighted fraction to the result from Step 2 to get the final result.

4. Join the original image to the outcome from Step 3.

The equation for the unsharp masking operation is

$$f'(m,n) = f(m,n) + \alpha\left[f(m,n) - \bar{f}(m,n)\right]$$

where $f(m, n)$ is the original image,
$\bar{f}(m,n)$ is the blurred version of the original image,
α is the weighting fraction, and
$f'(m,n)$ is the sharpened result.

4.10 BIT-PLANE SLICING

The gray level of each pixel in a digital image is stored as one or more bytes in a computer. For an 8-bit image, 0 is encoded as 0 0 0 0 0 0 0 0, and 255 is encoded as 1 1 1 1 1 1 1 1. Any number between 0 and 255 is encoded as one byte. The bit in the far-left side is referred as the Most Significant Bit (MSB), because a change in that bit would significantly change the value encoded by the byte. The bit in the far right is referred as the Least Significant Bit (LSB), because a change in this bit does not change the encoded gray value much. The bit-plane representation of an eight-bit digital image is shown in figure 8.

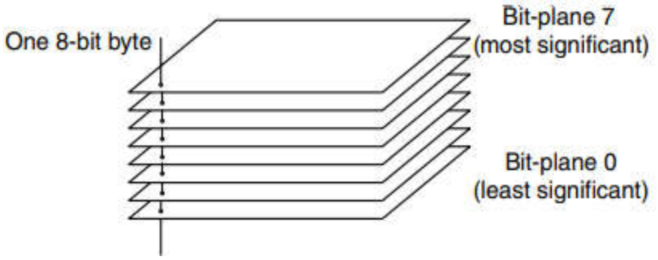


Figure 8: Bit plane representation of a digital image

One or more bits of the byte are used for each pixel when an image is represented using the bit-plane slicing technique. The original grey level is converted to a binary representation because one can only use the MSB to represent a pixel.

Bit plane slicing has three major purposes:

i) converting a grayscale image to a binary image;
ii) representing an image with fewer bits and compressing it to a smaller size; and
iii) enhancing the image by sharpening certain areas.

4.11 IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

Frequency is the rate at which a particular periodic event repeats itself. Spatial frequency in image processing is the fluctuation of image brightness with image position in space. A signal that changes over time can be converted into a sequence of easy periodic variations. A signal is broken down into a collection of sine waves with varying frequency and phase using the Fourier transform. Because a Fourier transform is entirely reversible, if we compute a picture's Fourier transform and then immediately inverse convert the result, we can recover the original image. On the other hand, we can emphasize some frequency components and attenuate others by multiplying each element of the Fourier coefficient by an appropriately selected weighting function.After computing an inverse transform, the corresponding changes in the spatial form can be observed. Frequency domain filtering, often known as Fourier filtering, is the process of selectively enhancing or suppressing frequency components. The adjacency relationship between the pixels is described by the spatial representation of image data. The frequency domain representation, on the other hand, groups the visual data based on the frequency distribution. In frequency domain filtering, the picture data is divided into different spectral bands, each of which represents a particular range of image information. Frequency domain filtering is the method of selective frequency inclusion or exclusion.

It is possible to perform filtering in the frequency domain by specifying the frequencies that should be kept and the frequencies that should be discarded. Spatial domain filtering is accomplished by convolving the image with a filter kernel. We know that

Convolution in spatial domain = Multiplication in the frequency domain

If filtering in spatial domain is done by convolving the input image f (m, n) with the filter kernel h(m, n)

$$\text{Filtering in spatial domain} = f(m,n) * h(m,n)$$

In the frequency domain, filtering corresponds to the multiplication of the image spectrum by the Fourier transform of the filter kernel, which is referred to as the frequency response of the filter

$$\text{Filtering in the frequency domain} = F(k,l) \times H(k,l)$$

Here, F (k, l) is the spectrum of the input image and H (k, l) is the spectrum of the filter kernel. Thus, frequency domain filtering is accomplished by taking the Fourier transform of the image and the Fourier transform of the kernel, multiplying the two Fourier transforms, and taking the inverse Fourier transform of the result. The multiplication of the Fourier transforms needs to be carried out point by point. This point-by-point multiplication requires that the Fourier transforms of the image and the kernel themselves have the same dimensions. As convolution kernels are commonly much smaller than the images they are used to filter, it is required to zero pad out the kernel to the size of the image to accomplish this process.

4.12 HOMOMORPHIC FILTER

An image can be modeled as the product of an illumination function and the reflectance function at every point. Based on this fact, the simple model for an image is given by

$$f(n_1, n_2) = i(n_1, n_2) \times r(n_1, n_2)$$

This model is known as illumination-reflectance model. The illumination-reflectance model can be used to address the problem of improving the quality of an image that has been acquired under poor illumination conditions.

In the above equation, $f(n_1,n_2)$ represents the image, $i(n_1,n_2)$ represents the illumination component and r (n_1,n_2) represents the reflectance component. For many images, the illumination is the primary contributor to the dynamic range and varies slowly in space, while the reflectance component r (n_1,n_2) represents the details of the object and varies rapidly in space. If the illumination and the reflectance components have to be handled separately, the logarithm of the input function f (n_1,n_2) is taken.

Because $f(n_1,n_2)$ is the product of $i(n_1,n_2)$ with $r(n_1,n_2)$, the log of $f(n_1,n_2)$ separates the two components as illustrated below:

$$\ln[f(n_1,n_2)] = \ln[i(n_1,n_2)r(n_1,n_2)] = \ln[i(n_1,n_2)] + \ln[r(n_1,n_2)]$$

Taking Fourier transform on both sides, we get

$$F(k,l) = F_I(k,l) + F_R(k,l)$$

where $F_I(k, l)$ and $F_R(k, l)$ are the Fourier transform of the illumination and reflectance components respectively. Then, the desired filter function H(k,l) can be applied separately to the illumination and the reflectance component separately as shown below:

$$F(k,l) \times H(k,l) = F_I(k,l) \times H(k,l) + F_R(k,l) \times H(k,l)$$

In order to visualize the image, inverse Fourier transform followed by exponential function is applied. First, the inverse Fourier transform is applied as shown below:

$$f'(n_1,n_2) = \Im^{-1}[F(k,l) \times H(k,l)] = \Im^{-1}[F_I(k,l) \times H(k,l)] + \Im^{-1}[F_R(k,l) \times H(k,l)]$$

The desired enhanced image is obtained by taking the exponential operation as given below:

$$g(n_1,n_2) = e^{f'(n_1,n_2)}$$

Here, $g(n_1,n_2)$ represents the enhanced version of the original image $f(n_1,n_2)$. The sequence of operation can be represented by a block diagram as shown in figure 9

Figure 9: Homomorphic filtering block diagram

4.13 ZOOMING OPERATION

When an image has fine features, it could be difficult to see those elements well in a monitor's normal display. Under these circumstances, zooming in on the image enables viewing of fine features. The zooming function allows for the expansion of an image. Holding a magnifying glass in front of the screen is the same as zooming. The simplest method of zooming is the "replication" of pixels. "Put the same value of the pixel in a grid of NxN pixels for every pixel of the image," is the replication principle. Diagrammatic representation of this is in figure 10.

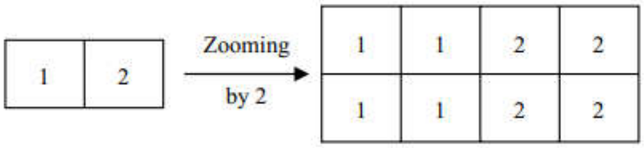


Figure 10: Zooming illustration

4.14 IMAGE ARITHMETIC

Several arithmetic processes, including image addition, subtraction, multiplication and division are taken into account in image arithmetic. Objects can be added to and subtracted from images using addition and subtraction operations.

4.14.1 Image addition

Image addition is used to create double exposure. If f (m, n) and g(m, n) represent two images then the addition of these two images to get the resultant image is given by

$$c(m,n) = f(m,n) + g(m,n)$$

If multiple images of a given region are available for approximately the same date and if a part of one of the images has some noise, then that part can be compensated from other images available through image addition. The concept of image addition is illustrated in figure 11.

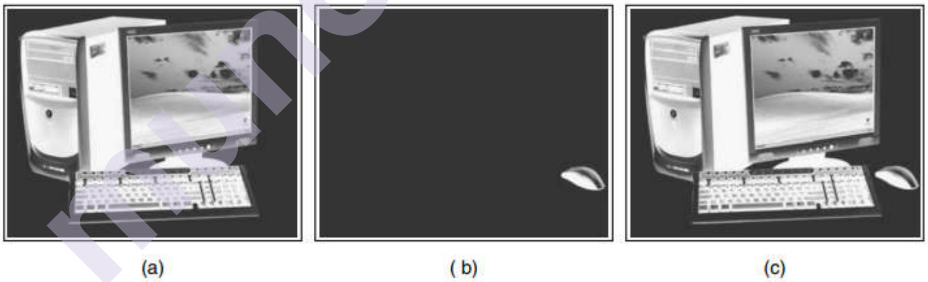


Figure 11: Image addition

In figure 11 there are three images (a), (b) and (c). Here, image (c) is obtained by addition of images in (a) and (b).

4.14.2 Image subtraction

Image subtraction is used to find the changes between two images of a same scene. The mathematical representation of image subtraction is given by

$$c(m,n) = f(m,n) - g(m,n)$$

To assess the degree of change in an area, two dates of co-registered images can be used with the subtraction operation. Image subtraction can be used to remove certain features in the image. The image subtraction is illustrated in the figure 12.

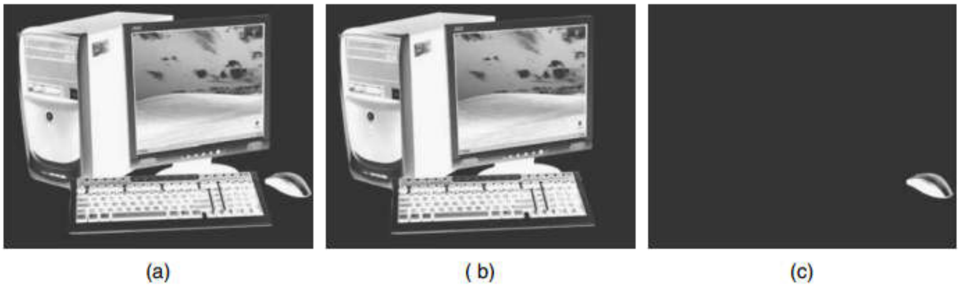


Figure 12: Image subtraction

The image subtraction is illustrated in figure 12.The images (a) and (b) are subtracted to get the image (c).

4.14.3 Image multiplication

Image multiplication is basically used for masking. If the analyst is interested in a part of an image then extracting that area can be done by multiplying the area by one and the rest by zero. The operation is depicted in the figure 13.

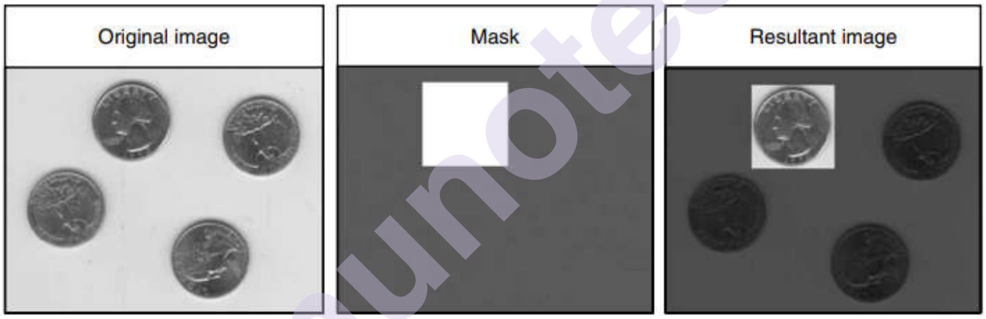


Figure 13: Image multiplication

From the figure, it is obvious that the multiplication operation is used to extract specific information from the image. From the figure, it is obvious that the coin of interest is highlighted.

4.14.4 Image division

Dividing the pixels in one image by the corresponding pixels in a second image is commonly used in transformation.The operation is depicted in the figure 14.From the figure it is obvious that the result of image division is just opposite to that of image multiplication.

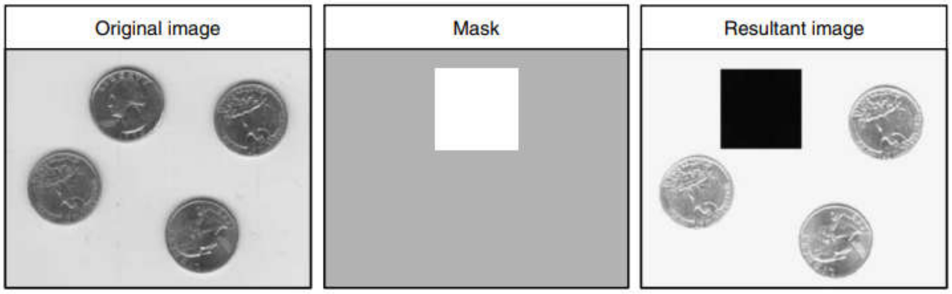


Figure 14: Image division

4.15 SUMMARY

- By modifying intensity functions, picture spectral content, or a combination of these functions, image enhancement aims to alter how the image is perceived.
- Images can be enhanced by removing blurriness and noise, boosting contrast, and bringing out features, for instance.
- Either the spatial domain or the frequency domain can be used for image enhancement. Application determines which image-enhancement method should be used.
- The gray-level changes used in the spatial-domain picture enhancement technique include image negative, image slicing, image thresholding, gamma correction, etc.
- Each pixel's value is recalculated in a point operation in accordance with a specific transformation. A common aspect of point operations is image brightness and contrast correction.
- The frequency of occurrence of the grey levels is indicated by the histogram of an image. The histogram of two photos can be identical. A histogram is rotation-invariant.

4.16 LIST OF REFERENCES

1. Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, Tata McGraw-Hill Education Pvt. Ltd., 2009.
2. Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008.
3. Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. Esakkirajan and T. Veerakumar, 2016 (https://scilab.in/textbook_companion/generate_book/125).

4.17 UNIT END EXERCISES

1) Explain the image enhancement in spatial domain.
2) Describe the enhancement process through point operations.
3) Write a note on Histogram manipulation.
4) Explain the Linear Gray Level Transformation.
5) Write a note on Nonlinear Gray Level Transformation.
6) Describe Median Filter.
7) Write a note on Spatial domain High pass filtering.
8) Explain the concept of Bit-plane slicing.
9) Describe the process of Image Enhancement in frequency domain.
10) Write a note on Homomorphic filter.
11) Describe various zooming operations.
12) State and explain various arithmetic operations performed on an image.

❖❖❖❖

5

BINARY IMAGE PROCESSING

Unit Structure

5.0 OBJECTIVES

- To get familiar with the concept of binary image processing and various morphological operations involved
- To understand several binary image operations
- To get familiar with the fundamentals of distance transform

5.1 INTRODUCTION

The study of form, organization, and appearance is known as morphology. A group of non-linear techniques known as mathematical morphology can be used to an image to eliminate details that are smaller than a particular reference shape. The initial definition of mathematical morphology's operations as set operations demonstrated its use for handling sets of 2D point sets. The morphological operations can be used to skelatonize, filter, and extract the edges of a picture. Dilation, erosion, opening, and closing are some of the fundamental morphological operations covered in this chapter. They are followed by the characteristics of morphological processes.Although binary, grayscale, and colour pictures can all be subjected to morphological processes, our focus in this chapter is on using various morphological procedures on binary images.An important subclass of digital images are binary images, which have only two grey levels. The final product of picture segmentation is typically a binary image.

5.2 MATHEMATICAL MORPHOLOGY

A highly fruitful area of image processing is mathematical morphology. A method for removing visual elements that are helpful for representation and description is mathematical morphology. The technique was originally developed by Matheron and Serra at the Ecole des Mines in Paris. The collection of structural data on the visual domain serves as the inspiration. Set theory serves as the sole foundation for the subject matter of mathematical morphology. Several useful operators described in mathematical morphology can be used by employing set operations. In mathematical morphology, sets stand in for the image's objects.For instance, a binary image's black or white pixels collectively represent a morphological description of the image. The sets in a binary image are the 3-D image domain's individuals with their integer elements. A 3-D tuple with the elements x and y coordinates is used to represent each element in the image. The development of image-segmentation techniques with a variety of applications can be based on mathematical morphology, and it also plays a significant part in techniques for picture description.

5.3 STRUCTURING ELEMENTS

When applied to an image, mathematical morphology is a group of non-linear procedures that can be used to eliminate details that are smaller than a particular reference shape, known as a structuring element. With its various shapes and sizes, the structural element in a morphological operation plays a significant role. A number of 0s and 1s in the structural elements determine shape and size. The resultant value is applied to the circle in figure 1 known as the centre pixel. Depending on the user's perspective, this circle can be located wherever in the structuring element. Figure 1 displays some examples of structural element references.

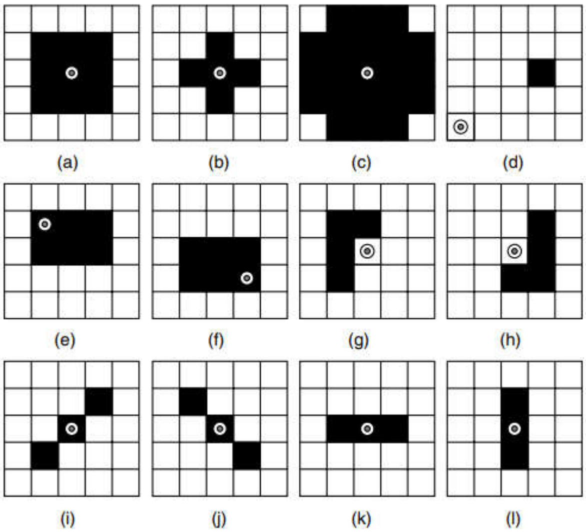


Fig. *Some possibilities of 5 x 5 square structuring elements are shown. They are named as (a) N8 (8 –Neighbourhood centred) (b) N4 (4 –Neighbourhood centred) (c) Flat plus (d) Shifted version (e) 2 × 3 sized rectangular (f) Reflected structuring element of Fig. (e) (g) 'L' Shaped structuring element (h) Reflected version of structuring element Fig. (g) (i) line-structuring element of 45° (j) line-structuring element of 135° (k) Horizontal structuring element with size 1 × 3 (l) Vertical structuring element with size 3 × 1*

Figure 1: Example of structuring element reference

5.4 MORPHOLOGICAL IMAGE PROCESSING

By shifting a structuring element so that it is centred over each pixel of the binary picture to be edited at some time, morphological operations are defined. A logical operation is carried out on the pixels covered by the structuring element when it is centred over a specific area of the image, producing a binary output. As seen in figure 2, the morphological image processing resembles a convolution method. The structural element, like the convolution kernel, can be any size and can include complements of 1s and 0s.A specific logical action is carried out between the structuring element and the underlying binary image at each pixel point. At that pixel position in the output image, the binary outcome of that logical operation is saved. The structuring element's size, content, and type of logical operation all influence the impact that is produced. The binary picture and structural element sets could be defined in 1, 2, or even higher dimensions rather than being limited to sets on the 2D plane. Do the logical operation if the structuring element is precisely positioned on the binary image; else, leave the generated binary image pixel alone.

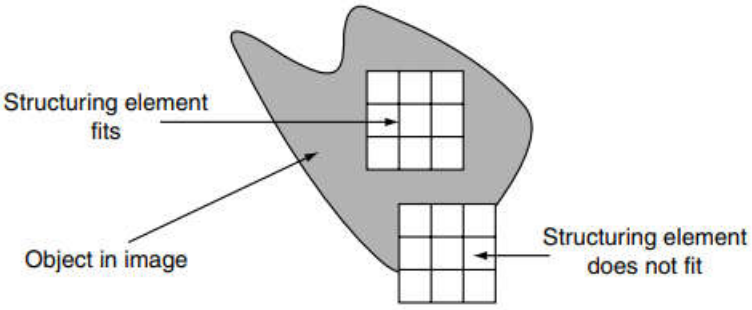


Figure 2: Morphological image processing

5.5 LOGICAL OPERATIONS

Logical operations includes AND, OR, NOT, EXOR operations, etc.

1] AND operation: The AND logic operation is similar to the intersection operation of set theory and the AND operation of two images is shown in Fig. 3. The AND logic operation computes the common area of images A and B as a resultant image.

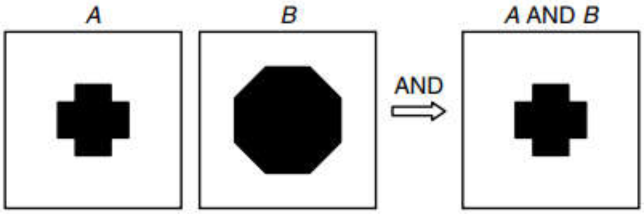


Figure 3: AND operation of images A and B

2] OR operation: The OR operation is similar to the union operation and the OR operation result of images A and B is shown in figure 4.

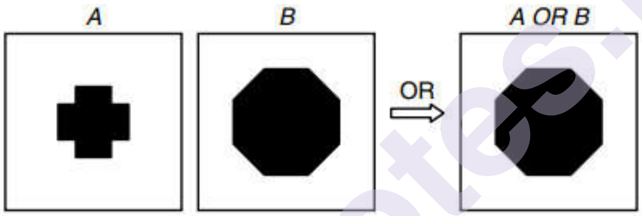


Figure 4: OR operation of images A and B

3] NOT operation: The NOT operation is similar to the complement function of set theory. In the NOT operation, the image pixel 1 is changed to 0 and vice versa. The NOT operation of Image A is shown in figure 5.

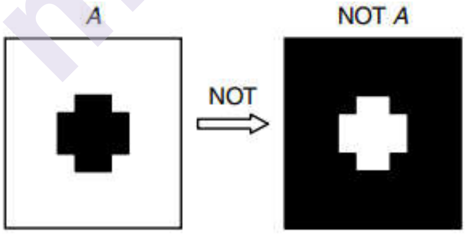


Figure 5: NOT operation of image A

4] EXOR operation: In the XOR operation, if the pixels of Image A and Image B are complementary to each other than the resultant image pixel is black, otherwise the resultant image pixel is white. The XOR operation of images A and B as shown in figure 6.

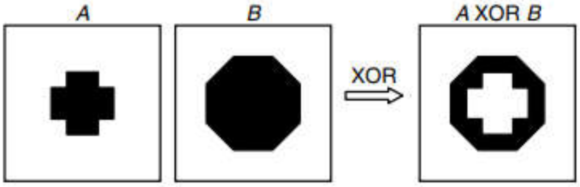


Figure 6: EXOR operation of images A and B

5.6 MORPHOLOGICAL OPERATIONS

Dilatation and erosion are the two primary morphological processes. They can be stated using a kernel that operates on a binary input image called X, where white pixels signify uniform regions and black pixels signify region boundaries.By translating a structural element, B, over the image points and looking at the intersection of the translated kernel coordinates and image coordinates, erosion and dilation operate conceptually.

5.6.1 Dilation

The binary picture is expanded from its original shape during the dilation process. The structural element determines how the binary picture is enlarged. Compared to the image itself, this structuring element is smaller in size; typically, the structuring element is 3 × 3. The structuring element is reflected and shifted from left to right and from top to bottom throughout the dilation process, similar to the convolution process. During each shift, the procedure searches for any overlapping identical pixels between the structuring element and that of the binary picture. The pixels under the centre location of the structuring element will be set to 1 or black if there is an overlap.

Let us define X as the reference image and B as the structuring element. The dilation operation is defined by

$$X \oplus B = \left\{ z \left| \left[(\hat{B})_z \cap X \right] \subseteq X \right. \right\}$$

where Bˆ is the image B rotated about the origin. Equation 10.7 states that when the image X is dilated by the structuring element B, the outcome element z would be that there will be at least one element in B that intersects with an element in X. If this is the case, the position where the structuring element is being centred on the image will be ‘ON’. This process is illustrated in figure 7. The black square represents 1 and the white square represents 0.

Initially, the centre of the structuring element is aligned at position *. At this point, there is no overlapping between the black squares of B and the black squares of X; hence at position * the square will remain white. This structuring element will then be shifted towards right. At position **, we find that one of the black squares of B is overlapping or intersecting with the black square of X. Thus, at position ** the square will be changed to black. Similarly, the structuring element B is shifted from left to right and from top to bottom on the image X to yield the dilated image as shown in figure 7.

The dilation is an expansion operator that enlarges binary objects. Dilation has many uses, but the major one is bridging gaps in an image, due to the fact that B is expanding the features of X.

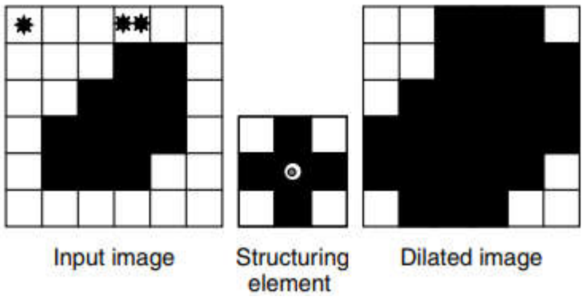


Figure 7: Dilation process

5.6.2 Erosion

Dilation's opposite process is erosion. Erosion decreases an image if dilation enlarges it. The structural element determines how the image is reduced. With a 3x3 size, the structural element is typically smaller than the image. When compared to greater structuring-element sizes, this will guarantee faster computation times. The erosion process will shift the structural element from left to right and top to bottom, almost identical to the dilatation process. The method will check whether there is a complete overlap with the structuring element at the centre location, denoted by the centre of the structuring element, or not.If there is no complete overlapping then the centre pixel indicated by the centre of the structuring element will be set white or 0.

Let us define X as the reference binary image and B as the structuring element. Erosion is defined by the equation

$$X \ominus B = \left\{ z \middle| (B)_z \subseteq X \right\}$$

The above equation states that the outcome element z is considered only when the structuring element is a subset or equal to the binary image X. This process is depicted in figure 8. Again, the white square indicates 0 and the black square indicates 1.

The erosion process starts at position *. Here, there is no complete overlapping, and so the pixel at the position * will remain white. The structuring element is then shifted to the right and the same condition is observed. At position **, complete overlapping is not present; thus, the black square marked with ** will be turned to white. The structuring element is then shifted further until its centre reaches the position marked by ***. Here, we see that the overlapping is complete, that is, all the black squares in the structuring element overlap with the black squares in the image. Hence, the centre of the structuring element corresponding to the image will be black.

Erosion is a thinning operator that shrinks an image. By applying erosion to an image, narrow regions can be eliminated, while wider ones are thinned.

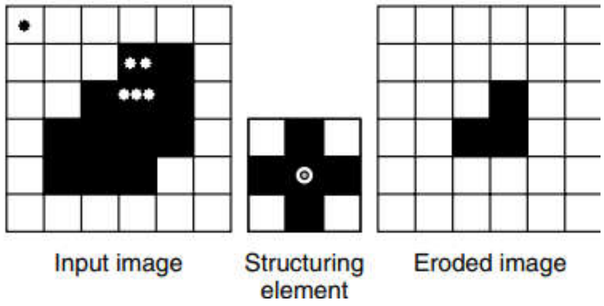


Figure 8: Erosion process

5.6.3 Dilation and erosion-based operations

Erosion and dilation can be combined to solve specific filtering tasks. The most widely used combinations are opening and closing.

1] Opening:Opening is based on the morphological operations, erosion and dilation. Opening breaks up thin strips, smoothes the interior of object contours, and removes thin image areas. It involves applying erosion and dilation procedures to the image in that order.

The opening procedure is used to clean up CCD flaws and image noise. By rounding corners from inside the object where the kernel utilizes fits, the opening filters details and simplifies images. The mathematical representation of the opening procedure is

$$X \circ B = (X \Theta B) \oplus B$$

where X is an input image and B is a structuring element.

2] Closing:The opposite of the opening process is the shutting operation. It is a process of first dilatation and then erosion. A single-pixel object's closure process fills in the tiny imperfections and spaces. It has the same result as an opening operation in that it retains object forms and sizes and smooths out features.

The mathematical representation of the closing procedure is

$$X \bullet B = (X \oplus B) \Theta B$$

where X is an input image and B is the structuring element. Closing protects coarse structures, closes small gaps and rounds off concave corners.

5.7 DISTANCE TRANSFORM

An operator called the distance transform is often exclusively used with binary pictures. Each feature pixel in a binary image is given a value by the distance transform that is equal to its distance from the closest non-feature pixels. It can be used to gather details on the form and placement of the foreground pixels in relation to one another. It has been used in numerous scientific areas, including surface reconstruction, pattern recognition, and medical imaging. When comparing binary images, the distance transforms are crucial, especially for images produced by local feature-detection methods like edge or corner detection.

Depending on how the distance is calculated, there are various types of distance transforms. The Gaussian distance transform, city block, chessboard, and Euclidean distance can all be used to calculate the distance between pixels. The following is a description of the distance metrics utilized.

5.7.1 Euclidean distance

The value at a pixel in the Euclidean distance transform is directly proportional to the distance in Euclidean terms between that pixel and the object pixel that is closest to it. The procedure is noise-sensitive because it determines the value at the pixel of interest using the value at a single object pixel.

The straight-line distance between two pixels is given by

$$D_E\left[(i, j), (k, l)\right] = \left[(i-k)^2 + (j-l)^2\right]^{1/2}$$

The input image and its Euclidean distance transform are shown in the figure 9.

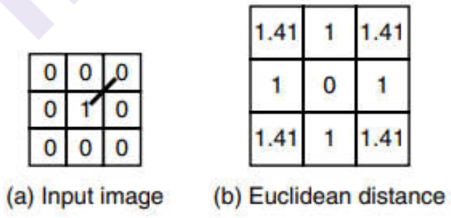


Figure 9: Euclidean distance transform

5.7.2 City-Block distance

The city-block distance of two pixels

$$D_4\left[(i, j), (k, l)\right] = |i-k| + |j-l|$$

This metric measures the path between the pixels based on a four-connected neighbourhood, as shown in figure 10. For a pixel ‘p’ with the coordinates (x, y), the set of pixels given by,

$$N_4(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\}$$

Above equation is called its four neighbours, as shown in Fig. 10. The input and its city block distance are shown in Fig. 11.

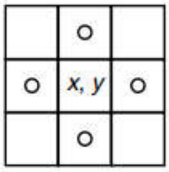


Figure 10: Four-connected neighbours

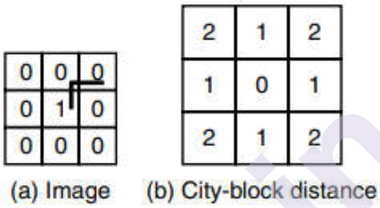


Figure 11: City block distance transform

5.7.3 Chessboard distance

The chessboard distance of two pixels is given by

$$D_8[(i, j), (k, l)] = \max[|i-k|, |j-l|]$$

The chessboard distance metric measures the path between the pixels based on an eight-connected neighbourhood, as shown in figure 13. The set of eight neighbour pixels is defined as

$$N_8(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\} \cup \{(x+1, y+1), (x-1, y+1), (x-1, y+1), (x+1, y-1)\}$$

The above equation is called the eight connected neighbours as shown in Fig. 12. The chessboard distance of the image and the original image is shown in Fig. 13.

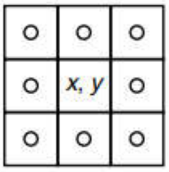


Figure 12: Eight connected neighbours

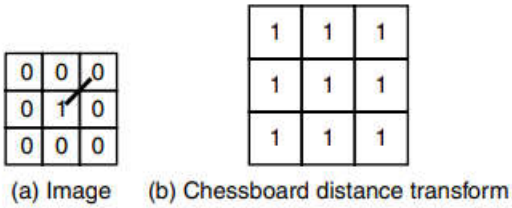


Figure 13: Chessboard distance transform

5.7.4 Quasi- Euclidean distance transform

The quasi-Euclidean metric measures the total Euclidean distance along a set of horizontal, vertical and diagonal line segments. The quasi-Euclidean distance of two pixels is given by

$$D_{QE}[(i,j),(k,l)] = \begin{cases} |i-k| + (\sqrt{2}-1)\times|j-l| & if\ |i-k| > |j-l| \\ (\sqrt{2}-1)\times|i-k| + |j-l| & \text{otherwise} \end{cases}$$

The quasi-Euclidean distance transform of the image is shown in Figure 14

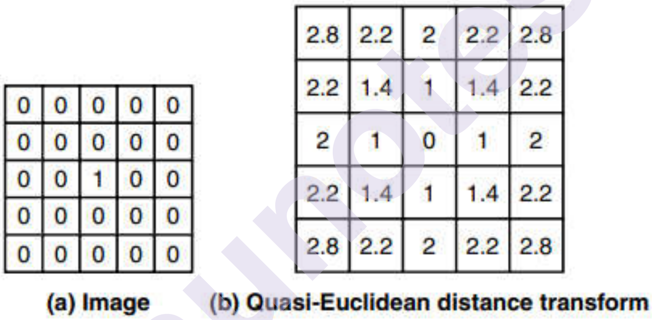


Figure 14: Quasi-Euclidean distance transform

5.8 SUMMARY

- The principle behind morphological image processing is to probe an image with a tiny form or pattern called a structuring element. To carry out a certain activity, structural elements of various shapes may be used.
- Dilatation and erosion are the two primary morphological processes.
- By successively moving the structuring element's origin to all feasible pixel locations throughout the dilation procedure, the image is probed with the structuring element; the result is 1 if the structuring element and the picture have a non-zero intersection and 0 otherwise.
- The erosion procedure involves probing the image with the structuring element by successively moving the structuring element's origin to all

potential pixel locations. If the structuring element is entirely contained in the image, the result is 1, otherwise it is 0.

- Opening and closing morphological processes are morphological filters that eliminate undesirable elements from an image.

5.9 LIST OF REFERENCES

1] Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009.

2] Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008.

3] Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. Esakkirajanand T. Veerakumar, 2016

(https://scilab.in/textbook_companion/generate_book/125).

5.10 UNIT END EXERCISES

1] Explain the mathematical morphology.

2] What do you mean bystructuring elements?

3] Write a note on Morphological image processing.

4] Explain thelogical operations.

5] State and describe the Morphological operations.

6] What is Dilation?

7]What is Erosion?

8] Describe the Dilation and Erosion-based operations.

9] Write a note on Distance Transform.

10] Explain Euclidean distance.

11] Describe City-Block distance.

12] What is Chessboard distance?

13] Explain Quasi- Euclidean distance transform.

❖❖❖❖

6

COLOUR IMAGE PROCESSING

Unit Structure

6.0 OBJECTIVES

- To understand the human perception and concepts related to colour image processing
- To learn different colour models
- To get familiar with colour histogram equalization

6.1 INTRODUCTION

The perception of colour is the result of incoming visible light on the retina. The most visually arresting aspect of every image is its colour, which also has a big impact on how scenically beautiful it is.

Understanding the nature of light is required in order to comprehend colour. Light has two distinct natures. It exhibits wave and particle characteristics. After striking the retina, photons cause electric impulses that, once they reach the brain, are interpreted into colour. Light waves with various wavelengths are seen as having various colours.The human eye cannot, however, see every wavelength. The visible spectrum is made up of wavelengths between 380 and 780 nanometers.

6.2 FORMATION OF COLOUR

A variety of physiochemical processes result in the colours that people see every day.Some things act as light sources, while others do little more than reflect incident light. Fundamentally, there are three different ways that colours are formed: three processes: additive processing, subtractive processing andpigmentation.

1] Additive colour formation

The spectral distributions for two or more light rays are added in additive colour creation.The total amount of photons in the same range that are present in all of the component colours makes up the final colour.TV monitors use the additive color-formation principle.

2] Subtractive colour formation

When light is transmitted through a light filter, subtractive colour creation happens. A light filter partially transmits and partially absorbs the light that it receives. For instance, a green filter allows radiation from the green portion of the spectrum to pass while blocking radiation from other wavelengths. The resultant colour is made up of the wavelengths that can pass through each filter when used in sequence. The projection of colour slides onto a screen result in subtractive colour generation.

3] Colour formation by pigments

Colored particles suspended in a liquid make up a pigment. The light that strikes these particles can be absorbed or reflected by them. A light beam is scattered by the particles as it encounters a surface covered in pigment, with consecutive and simultaneous actions of reflection, transmission, and absorption. These occurrences define the type (colour) of light that the surface reflects. One can see colours in a painting thanks to colour pigmentation.

6.3 HUMAN PERCEPTION OF COLOUR

The 400 to 700 nm range of the light spectrum is detectable by the human visual system. The two most crucial senses through which humans perceive their environment are sight and hearing. According to estimates, humans get 75% of their information visually. Perception, sight, or comprehension are terms used to describe the use of visual information. The idea for creating analogous paradigms for computers came from an understanding of human perceptual processing capabilities. A brief explanation of the anatomy of the human visual system is the best way to introduce the intricate aspects of visual perception.The Human Visual System (HVS) is a system for processing information that receives spatially sampled images from the cones and rods in the eye and infers the characteristics of the objects it sees. Cones are the photoreceptors in charge of colour vision. The retina contains three different kinds of cones. They are

i) L-receptors which are sensitive to light of long wavelengths
ii) M-receptors which are sensitive to light of middle wavelengths
iii) S-receptors which are sensitive to light of short wavelengths

The RGB sensors are often identified by the Greek letters rho (red), gamma (green), and beta (blue). Fig. 1 displays the sensitivity curves for rho, gamma, and beta.

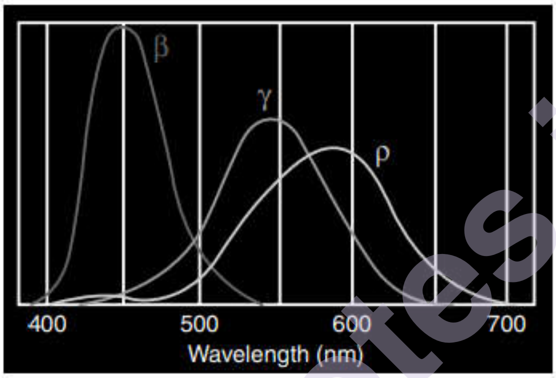


Figure 1: Sensitivity of rho, gamma and beta curves

The strength of the colours one receives for each of the wavelengths in the visual spectrum is determined by the sensitivity curves of the rho, gamma, and beta sensors in the human eye. The visual system can create the perception of colour by selective sensing of various light wavelengths.The sensitivity of light or the lighting of things affects colour perception as well. Cones are not activated at very low illumination levels. Yet, even with very little illumination, rods are triggered.

6.4 COLOUR MODEL

By specifying a 3D coordinate system and a subspace that encompasses all constructible colours within a specific model, colour models offer a uniform approach to identify a specific colour. A model can be used to specify any colour. Each colour model is tailored for a particular piece of software or image-processing tool.

6.4.1 RGB colour model

In an RGB colour model, the three primary colours red, green and blue form the axis of the cube which is illustrated in Fig. 2. Each point in the cube represents a specific colour. This model is good for setting the electron gun for a CRT.

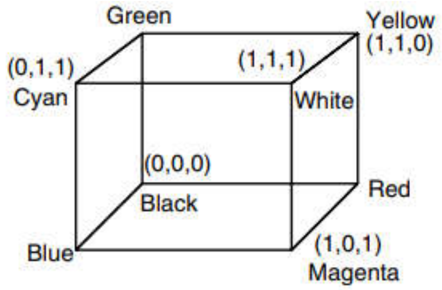


Figure 2: RGB colour cube

RGB is an additive colour model. From Fig. 3, it is obvious that

Magenta = Red + Blue

Yellow = Red + Green

Cyan = Blue + Green

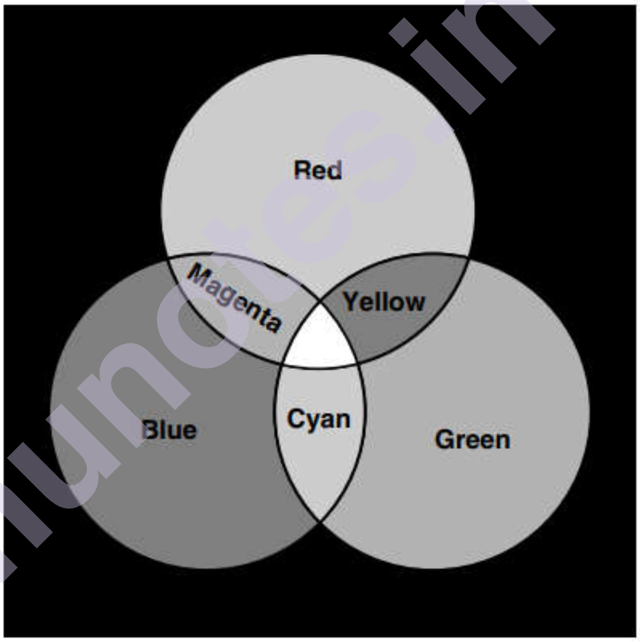


Figure 3: RGB colour model

Limitations of RGB model:

(i) The RGB colour coordinates depend on the device. This suggests that the RGB model will generally fail to accurately replicate the same colour on different displays.

(ii) The RGB model is not consistently perceived. The implication is that in all regions of the colour space, a unit of coordinate distance does not equate to the same perceived colour difference.

(iii) Because this model is based on device signals rather than display brightness values, it is challenging to relate it to the look of colours.

6.4.2 CMY colour model

Printers produce an image by reflective light, which is basically a subtractive process. Printers commonly employ the CMY model. The CMY model cube is illustrated in Fig. 4. The conversion between RGB and CMY is done as follows

C = 1-R

M = 1-G

Y = 1-B

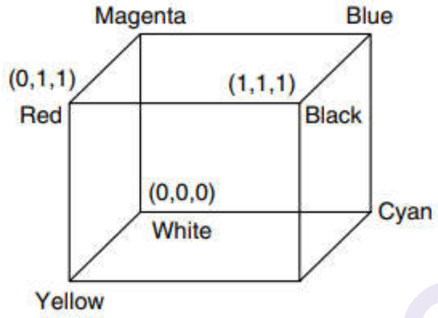


Figure 4: CMY colour cube

CMY is a subtractive colour model. From Fig. 5, it is obvious that

Magenta = White – Green

Cyan = White – Red

Yellow = White –Blue

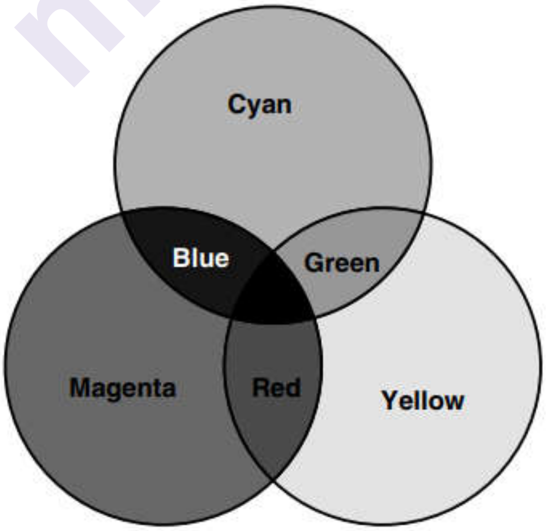


Figure 5: CMY colour model

6.4.3 HIS colour model

Hue, Saturation, and Intensity is referred to as HSI. The observer's perceived main colour is represented by hue. It is a characteristic connected to the predominant wavelength. The term "saturation" describes the degree of purity or the quantity of white light combined with a hue. Brightness is reflected in intensity. As hue and saturation correspond to how colours are seen by humans, HSI decouples colour information from intensity information, making this representation particularly helpful for creating image-processing algorithms. HIS colour space is well-liked because it is based on how people see colour.

The conversion from RGB space to HSI space is given below:

$$I = \frac{1}{3}(R + G + B)$$

$$S = 1 - \frac{3}{(R + G + B)}[\min(R, G, B)]$$

and

$$H = \cos^{-1}\left\{\frac{0.5[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}}\right\}$$

The HIS colour model can be considered as a cylinder, where the coordinates r, θ, and z are saturation, hue and intensity respectively. The cylindrical representation of the HIS colour model is illustrated in Fig 6.

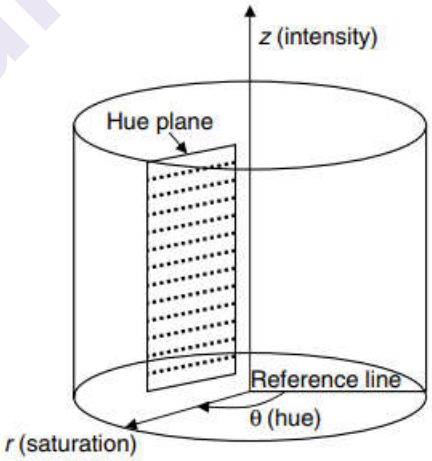


Figure 6: Cylindrical representation of HIS colour model

6.4.4 YIQ colour model

The YIQ colour model is defined by the National Television System Committee (NTSC). In this model, Y represents the luminance; and I and Q describe the chrominance. Conversion between RGB and YIQ is given below

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.956 & 0.621 \\ 1.0 & -0.272 & -0.649 \\ 1.0 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

6.4.5 YCbCr colour coordinate

The YCbCr colour coordinate was developed as part of the ITU-R BT.601 during the establishment of a worldwide digital video component standard. The YCbCr signals are scaled and offset versions of the YIQ format. Y is defined to have a nominal range of 16 to 235; Cb and Cr are defined to have a range of 16 to 240, with zero signal corresponding to level 128. There are several YCbCr formats such as 4:4:4, 4:2:2 and 4:1:1. The sampling format implies that the sampling rates of Cb and Cr are one-half of that of Y.

6.5 COLOUR IMAGE QUANTISATION

The process of dividing the original colour space into larger cells in order to reduce the size of a colour space is known as colour quantization. The method of color-image quantization involves lowering the number of colours in a digital colour image. A lossy image compression operation is essentially what color-image quantization is. There are two main processes in the process of quantizing colour images:

(i) Choose a small number of colours from the available combinations of red, green, and blue to create a colour map.

(ii) Assigning a colour from the colour map to each colour pixel in the colour image.

The basic goal of colour image quantization is to transfer the original colour image's set of colours to the quantized image's significantly smaller range of colours. A colour palette is a more condensed collection of exemplary hues. The discrepancy between the original and the quantized images should be as little as possible after the mapping.

6.5.1 Classification of colour quantisation technique

Colour quantisation, in general, can be classified into (i) uniform colour quantisation, and (ii) non-uniform colour quantisation

1] Uniform colour quantisation

The simplest colour quantization technique is uniform quantization. Each colour cell in uniform colour quantization represents a cube whose edges are parallel to the colour axes and each cell is the same size. Fig. 7 provides a visual picture of uniform colour quantization.

The formula for the uniform colour quantization scheme is

$$C = \{(R_i, G_j, B_k) \mid it \le (i+1)\cdot t, jt \le G < (j+1)\cdot t, kt \le B < (k+1)\cdot t; \quad i, j, k = 0, 1.. n-1\}$$

where (R, G, B) are the original colours

C is the colour space after quantisation

(R_i, G_j, B_k) is a chosen colour that corresponds to the (i, j, k)th cell

n is the number of quantisation cells in each dimension

t is the size of the quantisation cells such that $nt = 256$

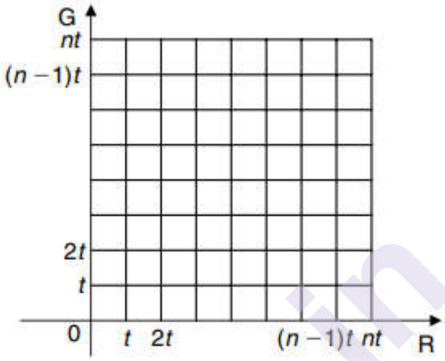


Figure 7: Representation of uniform quantisation in two dimensions

2] Non-uniform colour quantisation

By utilizing non-uniform quantization, different thresholds are used on each colour axis individually to split the colour space. Based on an investigation of the initial distribution in the colour space, the thresholds are determined. Fig. 8 provides a visual illustration of non-uniform quantization.

Non-uniform quantization over various thresholds is mathematically represented as follows

$$C = \{(R_i, G_j, B_k) \mid Rt_i\, R < Rt_{i+1}, Gt_j\, G < Gt_{j+1}, Bt_k\, B < Bt_{k+1};$$

$$i = 0, 1, \ldots r-1, j = 0, 1, \ldots, g-1, k = 0, 1, \ldots\ldots b-1\}$$

where (R, G, B) is the original colour

C is the colour space after quantisation

Rt_i, Gt_j and Bt_k are chosen thresholds

Rt_0, Gt_0 and Bt_0 are equal to zero while Rt_r, Gt_g and Bt_b are equal to 256. The number of cells in the three dimensions are not necessarily equal and are represented as r, g and b. The triple (R_i, G_j, B_k) represents a chosen colour that represents the corresponding (i, j, k)th cell. In this method, the cells might not have the same size. The total number of cells can be calculated by $r \cdot g \cdot b$.

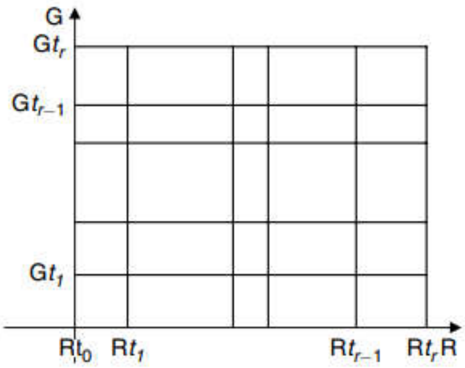


Figure 8:Representation of non-uniform quantisation in two dimensions

6.6 HISTOGRAM OF A COLOUR IMAGE

The histogram of the image displays the grey level's occurrence frequency. It provides the frequency with which a specific grey level appears in the image. The histogram is anticipated to provide the frequency with which a specific colour appears in a colour image.

6.6.1 Histogram equalization of a colour image

A colour image's histogram equalisation is done by first converting the RGB image to the YIQ(NTSC) format. The Y component alone is then subjected to histogram equalisation. The I and Q parts are unaltered. The equalised Y, unmodified I, and Q components are then converted to the RGB format in the histogram.

6.7 SUMMARY

- Between 400 nm (blue) and 700 nm is the range of visible light (red).
- Three different types of color-sensitive pigments, each responding to a different spectrum of wavelengths, are present in cones.
- The tristimulus model of colour vision, which is commonly used to describe how the responses of these many sensors combine to give us colour perception.
- Red, green, and blue are replaced by the three primaries X, Y, and Z in CIE.
- The Y component was selected to match luminous efficacy;Y= luminance X, Z = chromaticity
- The colour space spanned by a set of primary colours is called a colour gamut.
- The different colour models are RGB, CMY, YCbCr, YIQ, etc.

- The RGB model is used most often for additive models. In an RGB model, adding a colour and its complement create white.
- The most common subtractive model is the CMY model. In a CMY model, adding a subtractive colour and its complement create black.
- Each pixel of the colour image will have three values, one each for the red, green and blue components.
- The histogram of a colour image can be considered to consist of three separate one-dimensional histograms, one for each tricolour component.

6.8 LIST OF REFERENCES

1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009.

2) Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008.

3) Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. Esakkirajan and T. Veerakumar, 2016 (https://scilab.in/textbook_companion/generate_book/125).

6.9 UNIT END EXERCISES

1) Explain the formation of colour.
2) Describe the human perception of colour.
3) Discuss different colour models.
4) Write a note on colour image quantization.
5) Explain the classification of colour quantisation technique.
6) Write a note on Histogram of a colour image.
7) Describe the Histogram equalization of a colour image.

❖❖❖❖

7

IMAGE SEGMENTATION

Unit Structure

7.0 OBJECTIVES

- To separate or combine different image elements that can be used to develop objects of interest that can be the subject of investigation and interpretation.
- To get familiar with different segmentation techniques along with their applicability

7.1 INTRODUCTION

Image segmentation is the division of a picture into groups of pixels that are homogeneous in terms of a particular criterion. Adjacent groupings must be diversified and different groups must not intersect. Instead,then being pixel-oriented, segmentation algorithms are area-oriented. The division of the image into linked regions is the outcome of segmentation. Therefore, segmentation is the process of splitting an image into useful sections.

7.2 IMAGE SEGMENTATION TECHNIQUES

Image segmentation is classified into two categories:Local segmentation and Global segmentation

i] Localsegmentation: Local segmentation divides sub-images, which are tiny windows on a larger image, into separate segments. Local segmentation has a lot fewer pixels accessible than global segmentation. Local segmentation needs to use pixel data sparingly.

ii] Global segmentation:Segmenting an entire image is what global segmentation is all about. The majority of global segmentation's work involves segments with a sizable number of pixels. This strengthens the reliability of estimated parameter values for global segments.

7.3 REGION APPROACH

A region in an image is a collection of related pixels that are joined together. The region technique assigns each pixel to a specific object or region. The goal of the boundary technique is to identify and pinpoint the actual regional borders. The edges are first found in the edge approach, and then they are connected to create the necessary boundaries.

7.3.1 Region growing

An approach to picture segmentation known as "region growing" involves looking at nearby pixels and adding them to a region class if no edges are found. Each border pixel in the area goes through this process one more time. If neighbouring regions are discovered, a region-merging technique that preserves strong edges while dissolving weak edges is employed.

Starting with a seed is necessary for region growing. The seed could be a single pixel, but ideally it would be a region. By including as many nearby pixels as feasible that match the homogeneity requirement, a new segment is formed from the seed. After then, the resulting section is dropped from the procedure. The remaining pixels are used to choose a new seed. This keeps happening until every pixel has been assigned to a segment. The settings for each segment must be updated when pixels are gathered. The initial seed selected and the sequence in which neighbouring pixels are evaluated may have a significant impact on the segmentation that results. The choice of homogeneity criteria in image growth depends not only on the issue at hand but also on the kind of image that needs to be divided into segments. The criteria used to determine whether a pixel should be included in the area or not, the connectivity type used to identify neighbours, and the approach taken to visit neighbouring pixels all affect how regions evolve.

Comparing region-growing to traditional segmentation methods has various benefits. The boundaries of the growing zones are incredibly intertwined and narrow. In terms of noise, the algorithm is likewise highly steady.

7.3.2 Region splitting

Top-down strategies involve region division. Starting with a complete image, it is divided up so that the divided portions are more homogeneous than the entire. Since splitting substantially restricts the forms of segments, splitting alone is insufficient for acceptable segmentation. Therefore, the split-and-merge algorithm, splitting followed by a merging phaseis always preferred.

7.3.3 Region splitting and merging

An image-segmentation method that takes spatial information into account is region splitting and merging. The following is the region-splitting-and-merging method:

- Splitting

1]Let R stand for the full image in split 1.

2] Split or partition the image into successively smaller quadrant portions using the predicate P.

As seen in figure 1, a quadtree structure provides a practical way to visualize the splitting method. The root of a quadtree represents the complete image, and each node represents a subdivision.

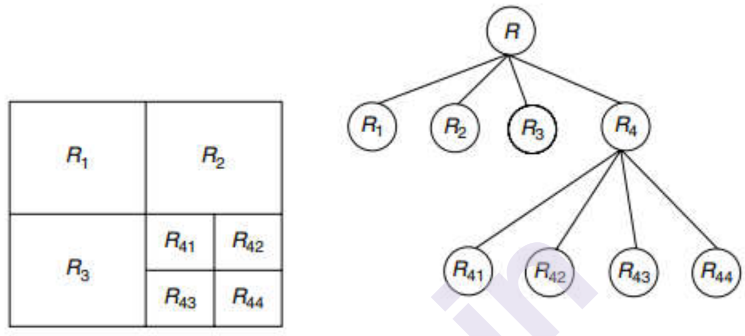


Figure 1: (a) Splitting of an image (b) Representation by a quadtree

It is probable that contiguous zones with the same characteristics will be found in the final partition. Applying merging and only merging nearby regions whose combined pixels fulfil the predicate P will address this flaw.

- Merging

Any adjacent regions that are similar enough should be merged. The split-and-merge algorithm's process is described below:

1. Commence with the entire picture.

2. Divide the variation into quadrants if it is too large.

3. Combine any nearby locations that are sufficiently comparable.

4. Continue iteratively repeating procedures (2) and (3) until no more splitting or merging takes place.

The image that has been separated and combined is shown in Figure 1(a). Figure 1(b) depicts the image's quadtree representation.

7.4 CLUSTERING TECHNIQUES

By grouping the patterns into clusters or groups so that they are more similar to one another than to patterns belonging to different clusters, the clustering technique aims to access the links between patterns of the data set. In other words, clustering is the division of things into groups based on shared characteristics. The clustering technique makes an effort to extract a feature vector from the image's localised regions. Each pixel is given the class of the closest cluster mean as part of the conventional clustering process. The two types of clustering techniques are partitional and hierarchical. There are numerous types of clustering methods within each category.

7.4.1 Hierarchical clustering

A proximity matrix is used in hierarchical clustering algorithms to show how similar each pair of data points is to the others. The outcome is a tree of clusters that displays the nested group of patterns and the similarity thresholds at which groups shift. While the root node is set aside for the full dataset and the leaf nodes are for individual data samples, the resulting clusters are always formed as the internal nodes of the tree. The procedures used to combine two small clusters or divide a large cluster vary depending on the clustering techniques.

The agglomerative and divisive algorithm categories are the two main types employed in the hierarchical clustering framework. Agglomerative algorithms start with N singlepoint clusters and attempt to combine them into larger and larger clusters. There are three categories into which the algorithm can be subdivided: (1) single-link algorithm, (2) complete-link algorithm, and (3) minimum-variance algorithm. According to the shortest distance between data samples from two clusters, the single-link algorithm combines two clusters. In light of this, the algorithm permits a propensity to create clusters with elongated shapes. The complete-link approach, however, consistently yields compact clusters despite incorporating the maximum distance between data samples in clusters.The definition of the dissimilarity measurement between two groups affects the quality of hierarchical clustering. By minimising the cost function and creating a new cluster with the least amount of cost function rise, the minimum-variance algorithm joins two clusters. This approach, known as the pairwise-nearest-neighbor algorithm, has generated a lot of attention in vector quantization.

Divide and conquer clustering starts with the entire dataset in one cluster and splits it repeatedly until single-point clusters are found on leaf nodes. In order to combat agglomerative clustering, it employs a reverse clustering approach. The divisive algorithm performs an exhaustive search for all pairings of clusters for data samples on each node.

Hierarchical algorithms include COBWEB, CURE, and CHAMELEON.

7.4.2 Partitional clustering

The iterative optimization process used in partition-based clustering tries to minimize an objective function f that gauges the effectiveness of clustering. The division of each pattern to the nearest cluster and the computing of the cluster centroids are the two learning processes that make up partition-based clustering.Partition-based clustering frequently begin with an initial solution that contains a predetermined number of clusters. The optimality criterion is typically used to compute the cluster centroids so that the objective function is minimized. The two types of partitional algorithms are density-based partitioning and partitioning relocation techniques.The first category of algorithms is further divided into probabilistic clustering, K-medoids, and K-means. Density-based partitioning, the second category of partitional algorithms, includes DBSCAN, OPTICS DBCLAS, and DENCLUE.Compared to hierarchical clustering techniques, partitional clustering techniques, such K-means clustering and ISODATA, have the advantage of maximizing some criterion function.

7.4.3 k-means clustering

The simplest method for unsupervised classification is the K-means approach. There is no need for training data for the clustering methods. Iterative processes such as K-means clustering are used. The K-means clustering technique segments the image by categorizing each pixel in the class with the nearest mean, iteratively computing a mean intensity for each class, and clustering the data. The K-means clustering algorithm's steps are listed below:

1] Choose K initial clusters $z_1(l)$, $z_2(l)$, ………. $z_k(l)$.

2]Distribute the samples x among the K clusters at the kth iterative step using the connection

$$x \in C_j(k) \quad \text{if } \|x - z_j(k)\| < \|x - z_i(k)\|$$

For i = 1, 2, …, K, i ≠ j, where $C_j(k)$ denotes the set of samples whose cluster centre is $z_j(k)$.

3] Determine the new cluster centres$z_j(k + 1)$, j = 1, 2,..., K, with the goal of minimizing the sum of the squared distances between all locations in C_j (k) and the new cluster. The sample mean of C_j (k) is the measurement that minimizes this. Consequently, the new cluster centre is provided by

$$z_j(k+1) = \frac{1}{N_j} \sum_{x \in C_j(k)} x, \quad j = 1, 2, \ldots\ldots\ldots K$$

where N_j is the number of samples in C_j (k)

4] If z_j (k + 1), j = 1, 2, ……, K, the algorithm has converged and the procedure is terminated. Otherwise go to Step 2

The K-means algorithm's disadvantage is that, once K is selected, the number of clusters is fixed and it always returns K cluster centres.

7.4.4 Fuzzy clustering

Depending on whether a pattern data belongs only to one cluster or to numerous clusters with different degrees, clustering methods can be categorized as either hard or fuzzy. In contrast to fuzzy clustering, which assigns a value between zero and one to each pattern, hard clustering assigns a membership value of zero or one to each pattern's data. Since they can more accurately depict the relationship between the input pattern data and clusters, fuzzy clustering algorithms can generally be regarded as being superior to those of their hard counterparts. Fuzzy clustering takes advantage of the fact that each pattern has some graded membership in each cluster in order to reduce a heuristic global cost function. The clustering criterion permits numerous cluster assignments for each pattern. The fuzzy K-means algorithm employs a gradient descent method to estimate the class membership function and iteratively update the cluster centroid.

Dissimilarity function

The dissimilarity function is a crucial parameter for determining how different data patterns are from one another. The diversity and size of the characteristics contained in patterns necessitates careful selection of the dissimilarity functions. Different clustering outcomes are produced by using various clustering dissimilarities. The Minkowski metricis a popular way to express how different two data vectors are which is given by,

$$d_p(x_i, x_j) = \left(\sum_{s=1}^{\dim} \left\| x_{i,s} - x_{j,s} \right\|^p \right)^{1/p}$$

p =1 means the L1 distance or Manhattan distance

p =2 means the Euclidean distance or L2 distance

Below are the pre requisite of a distance function that must be fulfilled by a metric

$$d(x, x) = 0, \quad \forall x \in X$$
$$0 \leq d(x, y), \quad \forall x, y \in X$$
$$d(x, y) = d(y, x)$$

The triangle inequality, a stricter requirement for the definition of metric, is defined as follows:

$$d(x, z) \leq d(x, y) + d(y, z) \quad \text{and} \quad d(x, y) = 0 \Leftrightarrow x = y$$

The case with l = ∞ in Minkowski metric comes to the maximum metric:

$$d(x_i, x_j) = \max_{l \leq s \leq \dim} \left| x_{i,s} - x_{j,s} \right|$$

These measurements are translation-invariant but scaling-variant. The Canberra metric is one of the scale-invariant differences given by

$$d(x_i, x_j) = \sum_{s=1}^{\dim} \frac{\left| x_{i,s} - x_{j,s} \right|}{\left| x_{i,s} \right| + \left| x_{j,s} \right|}$$

The cosine similarity is given by

$$d(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \times \|x_j\|}$$

The dot in this instance is the L2 norm in Euclidean space and indicates the inner product of two vectors. The Mahalanobis distanceis another method for generalizing the Euclidean distance to the scale invariant situation and is expressed as

$$d(x_i, x_j) = (x_i - x_j)^T \sum{}^{-1} (x_i - x_j)$$

Where Σ: covariance matrix with respect to the entire data set

$$\sum(X) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$$

When using a model-based clustering technique like the EM clustering algorithm, the Mahalanobis distance is used as a measure of dissimilarity. When used within the context of K-means clustering, the Mahalanobis distance application is always hampered by the singularity problem and a high computational cost.

7.5 THRESHOLDING

Segments made using thresholding techniques have pixels with comparable intensity. Setting limits in photos with solid objects resting on a contrasting background using thresholding is a valuable method. There are many segmentation techniques based on grey levels that use either global or local image data. The thresholding approach requires a background with a varied intensity level and an object with uniform intensity. Such a picture can be divided into two areas using straightforward thresholding.

7.5.1 Global Thresholding

The simplest and most used segmentation technique is global thresholding. In global thresholding, the following stipulation is imposed along with a threshold value of θ:

$$f(m, n) = \begin{cases} 1 & \text{if } f(m, n) \geq \theta \\ 0 & \text{else} \end{cases}$$

The equation mentioned above provides a detailed explanation of a binarization technique but makes no mention of how to choose the threshold parameter's value. It is necessary to choose the value of in the best possible way. When pixels from distinct segments employ intensities that overlap, global thresholding will suffer. If the cause is noise, a method like the minimum-error method can be used to determine the parameters of the underlying cluster and select the thresholds that will minimize the classification error. Variable thresholding could be utilized if the overlap is brought on by differences in illumination across the image. One way to see this is as a type of local segmentation.

7.5.2 Adaptive Thresholding

Global thresholding, also known as fixed thresholding, functions well when the background is uneven but has a generally uniform grey level and the objects of interest have an inner grey level that is reasonably consistent. In many instances, the object contrast within an image varies while the backdrop grey level is not consistent. In these circumstances, a threshold that performs well in one section of the image could perform poorly in other areas. In these situations, it is practical to employ a threshold grey level that progressively changes depending on where it is in the image.

7.5.3 Histogram-based threshold selection

The gray-level histogram of an image with an object against a contrasting background is bimodal. The comparatively large number of points inside and outside the object are represented by the two peaks. The comparatively few spots along the object's edge correlate to the dip between the peaks. It is usual practice to determine the threshold grey level using this dip.

The area function derivative for an object whose border is determined via thresholding is the histogram

$$H(D) = -\frac{d}{dD}A(D)$$

where D is the gray level, A(D) is the area of the object obtained by thresholding at gray level D, and H(D) is the histogram

If D is close to the histogram dip, raising the threshold from D to D +ΔD just slightly reduces the area. As a result, setting the threshold near the histogram's dip reduces the area measurement's susceptibility to minute threshold selection errors.

The histogram will be noisy if the image that contains the object is noisy and small. Unless the dip is exceptionally sharp, the noise can obscure or at least make the placement unpredictable from one image to the next. By applying a convolution filter or the curve-fitting method to smooth the histogram, this effect can be somewhat mitigated.

7.6 EDGE-BASED SEGMENTATION

Edge-based segmentation uses the edges of a picture to extract spatial information. Edges are when the homogeneity criteria for segments discontinues. Local linear gradient operators like Prewitt, Sobel, and Laplacian filters are typically used for edge detection. These operators perform best with images that have few noise sources and sharp edges. Some edge connecting may be necessary since the boundaries identified by these operators may not always consist of a collection of closed linked curves.

7.7 EDGE DETECTION

Edge detection is a technique for spotting significant changes in an image. One of the main functions of the lowest levels of image processing is edge detection. The boundaries between various objects are often formed by the locations where there are abrupt changes in brightness. By calculating intensity differences in local picture regions, these locations can be located. That is, an area with high signals of change should be sought after by the edge-detection algorithm. The majority of edge detectors measure the intensity gradient at a specific location in the image.

7.7.1 Gradient operator

A gradient is a two-dimensional vector that points to the direction in which the image intensity grows fastest. The gradient operator ∇ is given by

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}$$

If the operator ∇ is applied to the function f then

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

The gradient magnitude and gradient orientation are the two functions that can be represented in terms of the directional derivatives. It is feasible to determine the gradient's magnitude ||∇f|| and direction □(∇f).

The strength of the edge is determined by the gradient's magnitude, which reveals how much the neighborhood's pixels differ from one another. The gradient's size is given by

$$|\nabla f| = \left\| \begin{bmatrix} G_x \\ G_y \end{bmatrix} \right\| = \left\| \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \right\| = \left[G_x^2 + G_y^2 \right]^{\frac{1}{2}}$$

The greatest rate of growth of f (x, y) per unit distance in the gradient orientation of |∇f| is determined by the gradient's size.

The gradient orientation indicates the direction of the biggest shift, which is probably the edge-crossing direction. The gradient's direction is determined by

$$\phi(\nabla f) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

where the angle is measured with respect to the x-axis.

7.7.2 Edge detection using first order derivates

A digital pixel grid's derivative can be described in terms of differences. The first derivative of an image with gray-value pixels must meet the following requirements: it must be non-zero at the start of a gray-level step or ramp, and it must be non-zero along the ramp (constant change in grey values). It must also be zero in flat segments, or in areas of constant gray-level values. Using a one-dimensional function f (x), the first-order derivative can be found by

$$df/dx = f(x+1) - f(x)$$

A function of two variables, f (x, y), is an image. Only the partial derivative along the x-axis is mentioned in the previous equation. It is possible to identify pixel discontinuity in eight different directions, including up, down, left, right, and along the four diagonals.

By estimating the finite difference, the second technique of obtaining the first-order derivative is provided:

$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x+h, y) - f(x, y)}{h}$$

$$\frac{\partial f}{\partial y} = \lim_{h \to 0} \frac{f(x, y+h) - f(x, y)}{h}$$

The finite difference can be approximated as

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

and

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

Using the pixel coordinate notation and considering that j corresponds to the direction of x, and i corresponds to the y direction, we have

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

and

$$\frac{\partial f}{\partial y} = f(i, j) - f(i+1, j)$$

The majority of edge-detecting algorithms can be compared to gradient estimators. The gradient computations must be estimated since the gradient is a continuous function notion and the input signal (the image) is a finite signal. Convolution is most frequently used in gradient calculation because derivatives are shift-invariant and linear. For identifying edges, numerous kernels have been presented.

7.7.3 Roberts kernel

The main goal is to calculate the differences between adjacent pixels. To directly utilize{+ 1, -1} which calculates these differences, is one technique to detect an edge. These are referred to as forward differences in mathematics. In reality, the Roberts kernels are too small to accurately locate edges when there is noise. The Roberts operator for a cross-gradient method is the easiest approach to implement the first-order partial derivative.

$$\frac{\partial f}{\partial x} = f(i, j) - f(i+1, j+1)$$

and

$$\frac{\partial f}{\partial y} = f(i+1, j) - f(i, j+1)$$

Implementing the partial derivatives shown above involves roughly converting them to two 2x2 masks. The providers of the Roberts operator masks are

$$G_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

These filters have the shortest aid, which results in a more precise positioning of the edges, but the short support of the filters has the drawback of being susceptible to noise.

7.7.4 Prewitt kernel

Judy Prewitt is honoured by the name of the Prewitt kernels. The concept of central difference is the foundation of Prewitt kernels. In comparison to the Roberts operator, the Prewitt edge detector is a considerably better operator. Take a look at how the pixels are arranged around the core pixel [i, j] in the illustration below:

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & [i, j] & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

The partial derivates of the Prewitt operator are calculated as

$$G_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

and

$$G_y = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2)$$

In the above equations c is the constant thatsignifies the emphasis given to pixels closer to the centre of the mask. G_x and G_y are the approximations at [i, j].

Setting c = 1, the Prewitt operator mask is obtained as

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

The support is longer on the Prewitt masks. The edge detector is less susceptible to noise because the Prewitt mask differentiates in one direction and averages in the other.

7.7.5 Sobel kernel

The Sobel kernels bear Irwin Sobel's name. The Sobel kernel is based on central differences, however when averaging, it gives more weight to the central pixels. The Sobel kernels can be viewed as 3x3 approximations to Gaussian kernels' first derivatives. The Sobel operator's partial derivates are calculated as

$$G_x = (a_2 + 2a_3 + a_4) - (a_0 + 2a_7 + a_6)$$

and

$$G_y = (a_6 + 2a_5 + a_4) - (a_0 + 2a_1 + a_2)$$

The Sobel masks in matrix form are given as

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

A Sobel mask has superior noise-suppression qualities compared to a Prewitt mask.

7.7.6 Frei-Chen edge detector

By mapping the intensity vector with a linear transformation and then identifying edges based on the angle between the intensity vector and its projection onto the edge subspace, Frei-Chen mask edge detection is performed. The normalized weights allow for the realization of Frei-Chen edge detection. Unique Frei-Chen masks are masks that include all of the basis vectors. This suggests that the weighted sum of nine Frei-Chen masks is used to represent a 3x3 image area. The image is initially confounded with each of the nine masks. The convolution outcomes of each mask are then combined to produce an innerproduct. Following are the nine Frei-Chen masks:

$$G_1 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix};\ G_2 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix};\ G_3 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix}$$

$$G_4 = \frac{1}{2\sqrt{2}}\begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix};\ G_5 = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix};\ G_6 = \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

$$G_7 = \frac{1}{6}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix};\ G_8 = \frac{1}{6}\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix};\ G_9 = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The first four Frei-Chen masks are used to represent edges, the next four to represent lines, and the last mask to represent averages. The image is projected onto the required masks for edge detection.

7.7.7 Second derivative method of detecting edges in an image

Finding the ideal edge is comparable to determining the largest or least derivative point. It is possible to determine a function's maximum or minimum value by differentiating the provided function and looking for locations where the derivative is zero. The second derivative is obtained by differentiating the first derivative. It is analogous to looking for locations where the second derivative is zero to locate the best edges. Images can be processed using differential operators, although zeros rarely land perfectly on a pixel. They typically lie between pixels. By locating the zero crossings, the zeros can be isolated. When one pixel is positive and its neighbour is negative, this is known as a zero crossover. The following are issues with zero-crossing methods:

- Zero crossing methods produce two-pixel thick edges.
- Zero crossing methods are extremely sensitive to noise.

For images, there is a single measure, similar to the gradient magnitude, that measures the second derivative, which is obtained by taking the dot product of ∇ with itself

$$\nabla \cdot \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}$$

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

The operator $\nabla \cdot \nabla = \nabla^2$ is called the Laplacian operator.

The Laplacian operator when applied to the function f, we get

$$\nabla^2 f = \left(\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \right) f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

The following difference equations can be used to express the Laplacian operation:

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

and

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

Also

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

This implies that

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

The 3 × 3 Laplacian operator is given by

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

The brightness values of each of the surrounding pixels are subtracted from the central pixel via the Laplacian operator. The Laplacian's output is non-zero when the neighbourhood has a discontinuity in the form of a point, line, or edge. Depending on how close to the edge the central point is, it could be either positive or negative.

Rotational invariance characterises the Laplacian operator. As long as they are orthogonal, the Laplacian operator does not depend on the direction.

7.7.8 Laplacian of gaussian (LOG)

Noise in the input image is a significant factor in the Laplacian operator's performance degradation. Prior performing edge enhancement, the image might be smoothed to reduce the effects of noise. The Laplacian-of-Gaussian (LOG) operator smoothes the image via convolution with a kernel that has a Gaussian form and is then applied after.

7.7.9 Difference of gaussian filters (DoG)

The DoG filter is obtained by taking the difference of two Gaussian functions. The expression of a DoG filter is given by

$$h(m, n) = h_1(m, n) - h_2(m, n)$$

where $h_1(m, n)$ and $h_2(m, n)$ are two Gaussian functions which are given by

$$h_1(m, n) = e^{\frac{r^2}{2\sigma_1^2}} \quad \text{and} \quad h_2(m, n) = e^{\frac{r^2}{2\sigma_2^2}}$$

Hence $$h(m, n) = e^{\frac{r^2}{2\sigma_1^2}} - e^{\frac{r^2}{2\sigma_2^2}}$$

7.7.10 Canny edge detectors

The fact that a Laplacian zero-crossing merely combines the principal curvatures together makes it difficult to use as an edge detector. In other words, it doesn't actually determine the gradient's maximum magnitude. Edges are defined by the Canny edge detector as second derivative zero-crossings in the direction of the largest first derivative. The Canny operator performs their work in stages. The image is initially smoothed using a Gaussian convolution. Then, to highlight the areas of the image with high spatial derivatives, a 2D first derivative operator is used to the smoothed image. In the image of the gradient magnitude, edges give birth to ridges. Non-maximal suppression is the method by which the algorithm tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top in order to produce a thin line in the output. The two thresholds T1 and T2, with T1 > T2, regulate the hysteresis of the tracking process. Only at a position on a ridge higher than T1 may tracking start. From there, tracking continues in both directions until the height of the ridge drops below T2. Hysteresis like this prevents noisy edges from becoming many edge fragments. The three parameters that make up a Canny edge detector are (i) the width of the Gaussian kernel, (ii) the higher threshold, and (iii) lower threshold used by the tracker

The detector's sensitivity to noise is decreased by widening the Gaussian kernel, but some of the image's finer details are lost in the process. As the Gaussian width is raised, the localization inaccuracy in the identified edges also significantly rises. The Canny edge detector uses gaussian smoothing for two reasons. In addition to noise reduction, it can be used to first regulate how much detail emerges in the edge image.

For optimal results, the upper tracking threshold is often set relatively high and the lower threshold value is set quite low. If the lower threshold is set too high, noisy edges will fragment. If the higher threshold is set too low, more false and undesired edge fragments will be produced.

7.8 EDGE LINKING

One can threshold an edge-magnitude image and thin the resulting binary image down to single pixel-wide closed, linked boundaries if the edges are reasonably strong and the noise level is low. However, such an edge picture will contain gaps that need to be filled under less-than-ideal circumstances. Small gaps can be filled by looking for other end points in a neighbourhood of 5 by 5 pixels or greater, centred on the end point, and then filling in border pixels as necessary to join them. This, however, has the potential to oversegment images with several edge points.

- **Edge linking using Heuristic Search Algorithm**

The heuristic search strategy to connect two edge points P and Q is presented below if there is a gap between them:

A. Determine if P's neighbours qualify as the first step towards Q.

1. Name three P neighbours that are about in the same general area as Q.
2. For each of the points from Step 1, compute the edge quality function from P.
3. Decide on the option that improves edge quality from P to those points.
4. Begin the subsequent iteration from the location established in Step 2.
5. Continue in this manner until you reach point Q.

B. Establish and qualify the Path.

1. Evaluate the newly formed path's edge quality function against a threshold.
2. Discard the freshly formed edge if it is insufficiently robust.
3. Use the resulting graph as the boundary if the newly formed edge is robust enough.

7.9 HOUGH TRANSFORM

The straight-line y = mx + b isrepresented in polar coordinate as

$\rho = x \cos(\theta) + y \sin(\theta)$

where ρ, θ defines a vector from the origin to the nearest point on the straight line. This vector will be perpendicular from the origin to the nearest point to the line as shown in figure:

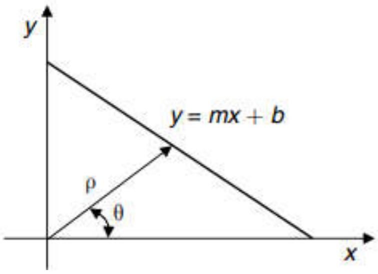


Figure 2: Straight line

A point in the 2D space determined by the parameters and corresponds to any line in the x, y plane. So, a single point in the ρ, θ space is the Hough transform of a straight line in the x, y space. Each straight line that traverses a certain point in the x, y plane, x_i and y_i maps to a point in the ρ, θ space, and these points should meet the aforementioned equation with x_1 and y_1 as constants. Any point in the x, y plane corresponds to a certain sinusoidal curve in the, space, and as a result, the locus of all such lines in the x, y space is a sinusoid in parameter ρ, θ space.

Suppose we have a set of edge points x_i, y_i that lie along a straight-line having parameters ρ_0, θ_0. Each edge point plots to a sinusoidal curve in the ρ, θ space, but these curves must intersect at a point ρ_0, $\theta_0$since this is a line, they all have common.

7.10 SUMMARY

- The goal of image segmentation is to divide an image into discernible, essentially homogeneous sections.
- Global segmentation deals with segmenting an entire image, whereas local segmentation focuses on segmenting sub-images, which are tiny windows on a larger image.
- There are three methods for performing image segmentation: the region approach, boundary approach, and edge approach.
- An approach to picture segmentation known as "region growing" involves looking at nearby pixels and adding them to a region class if no edges are found. With a top-down strategy known as region splitting, an entire image is split into portions that are more homogeneous than the whole.
- By grouping the patterns into clusters or groups so that they are more similar to one another than to patterns belonging to different clusters, the clustering technique aims to access the links between patterns of the data set.
- By using a thresholding process, a picture can be segmented, and the threshold value can be chosen using the image's histogram as a tool.

- Edges are essentially discontinuities in the intensity of the image brought on by adjustments to the image structure.
- It is possible to identify the edges in a picture using the Robert, Prewitt, Sobel, and Frei-Chen kernels.

7.11 LIST OF REFERENCES

1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009.
2) Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008.
3) Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. Esakkirajan and T. Veerakumar, 2016

(https://scilab.in/textbook_companion/generate_book/125).

7.12 UNIT END EXERCISES

1) What is content-based image retrieval?
2) What is an 'edge' in an image? On what mathematical operation are the two basic approaches for edge detection based on?
3) What are the three stages of the Canny edge detector? Briefly explain each phase.
4) Compare the Canny edge detector with the Laplacian of Gaussian edge detector
5) Describe various edge detection methods.
6) What is edge linking?
7) Describe Hough transform.
8) Discuss on image segmentation techniques.
9) Describe the region-based approach.

❖❖❖❖

8

IMAGE COMPRESSION

Unit Structure

8.0 OBJECTIVES

To understand and get familiar with the following concepts:

- need for image compression along with their metrics and techniques
- lossless and lossy image compression
- spatial domain and frequency domain

8.1 INTRODUCTION

The demand for digital information has rapidly increased as a result of the development of multimedia technologies over the past few decades. The widespread use of digital photographs is largely due to technological advancements. Applications like medical and satellite imagery frequently use still photos. Digital photos contain a tremendous quantity of data. As digital images find more applications, image data size reduction for both storage and transmission are becoming more and more crucial. A mapping from a higher dimensional space to a lower dimensional space is what image compression is. Many multimedia applications, including image storage and transmission, heavily rely on image compression.The fundamental objective of image compression is to represent an image with the fewest possible bits while maintaining acceptable image quality. All image-compression methods aim to reduce the amount of data as much as possible while removing statistical redundancy and taking advantage of perceptual irrelevancy.

8.2 NEED FOR IMAGE COMPRESSION

The amount of information that computers handle has increased tremendously over the past few decades thanks to advancements in Internet, teleconferencing, multimedia, and high-definition television technology. Consequently, a significant issue with multimedia systems is the storage and transmission of the digital image component. To exhibit photographs with an acceptable level of quality, a tremendous amount of data is needed. Large amounts of storage space and transmission bandwidth are needed for high-quality image data, which the existing technology cannot provide both technically and economically. Compressing the data to save on storage space and transmission time is one method that could work for this issue.

For instance, the amount of storage space needed to hold a 1600 x 1200 colour photograph is

$1200 \times 1600 \times 8 \times 3 = 46{,}080{,}000$ bits

= 5,760,000 bytes

= 5.76 Mbytes

1.44 Mb is the maximum amount of space on a floppy disc. The maximum space available if we have three floppies is 1.44 x 4 = 5.76 Mbytes. In other words, a 1600x1200 RGB image requires at least four floppies to store.

Every year, the amount of data carried across the Internet doubles, and a sizable chunk of that data is made up of photographs. Any given device will experience significant cost savings and become more accessible to use if its bandwidth requirements are reduced. photos can be represented more compactly through the use of image compression, allowing for speedier transmission and storage of photos.

8.3 REDUNDANCY IN IMAGES

Because images are typically highly coherent and contain redundant information, image compression is achievable. Redundancy and irrelevancy reduction help in compression. Duplication is referred to as redundancy, while irrelevancy is the portion of an image's information that the human visual system will not pick up on. Figure 1 depicts the classification of redundancy

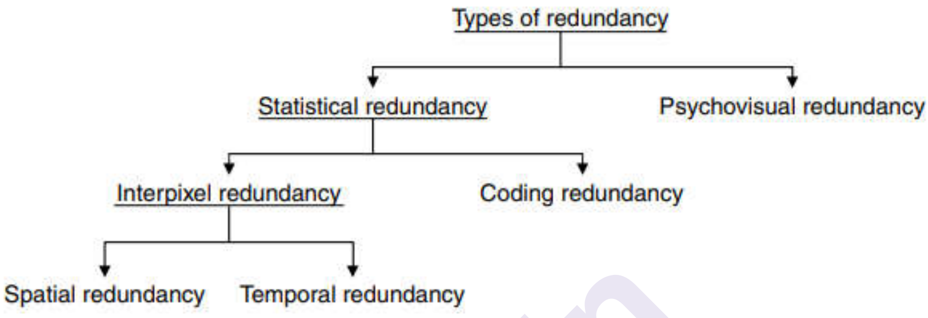


Figure 1: Classification of redundancy

- **Statistical redundancy**

As previously mentioned, there are two kinds of statistical redundancy: interpixel redundancy and coding redundancy.

When adjacent pixels in an image are correlated, interpixel redundancy results. The implication is that the pixels next to each other are not statistically independent. Interpixel redundancy is the term used to describe the interpixel correlation.

Information representation is connected to redundancy in coding. Codes are used to represent the information. Examples of codes include the arithmetic codes and the Huffman code. The efficiency of different codes may vary. To properly compress the image, codes must be effective.

- **Spatial redundancy**

In an image, the statistical relationship between adjacent pixels is represented by spatial redundancy. The term "spatial redundancy" refers to the relationship between adjacent pixels in an image. Each pixel in an image need not be individually represented. Instead, a pixel's neighbours can be used to forecast it. The fundamental idea behind differential coding, which is commonly used in picture and video reduction, is to eliminate spatial redundancy by prediction.

- **Temporal redundancy**

The statistical relationship between pixels from subsequent frames in a video clip is known as temporal redundancy.Interframe redundancy is another name for the temporal redundancy. To lessen temporal redundancy, motion compensated predictive coding is used. Effective video compression is achieved by removing a significant amount of temporal redundancy.

- **Psychovisual redundancy**

The traits of the human visual system (HVS) are connected to psychovisual redundancy. Visual information is not perceived similarly in the HVS. It's possible that certain information is more crucial than other information. Perception won't be impacted if less data is used to represent less significant visual information. This suggests that visual information is redundant in a psychovisual sense. Effective compression results from removing the psychovisual redundant information.

8.4 IMAGE-COMPRESSION SCHEME

Figure 2 depicts the general block diagram of an image-compression technique. While the channel encoder and decoder pair is often referred to as a channel codec module, the source encoder and decoder pair is frequently referred to as a source codec module. Over the years, Shannon's well-known separation theorem has been used to support the two coding modules' independent designs.

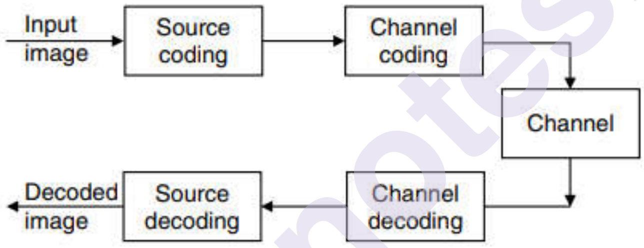


Figure 2: Image compression scheme

- Source Coding: The effective translation of the source data—the input image data—into a series of bits is the aim of source coding. By reducing entropy, the source coder lowers the average number of bits needed to represent the image. In essence, source decoding is an inverse process.

- Channel: The mathematical representation of the medium used for communication is the "channel."

- Channel Coding: The source encoder compresses the output and adds controlled redundancy using the channel encoder. The channel encoder's job is to shield the communication system from channel noise and other transmission mistakes. To recreate the compressed bits, the channel decoder makes use of the redundant bits in the bit sequence.

8.5 FUNDAMENTALS OF INFORMATION THEORY

Both lossless and lossy compression are mathematically based on information theory. Entropy, average information, and rate-distortion principles are covered in this section.

8.5.1 Entropy and mutual information

For a random variable X generated by a discrete memoryless source from a finite alphabet set X = $\{x_1, x_2, \ldots\ldots\ldots\ldots x_N\}$ with p_x being the probability density function, Shannon defined a measure of the information content of X, called entropy, as

$$H(X) = \sum_{x \in X} p_x(x) \log \frac{1}{p_x(x)}$$

The entropy H(X) is the lower bound of the bit rate that can represent the source without distortion. The conditional entropy of a random variable Y with a finite alphabet set y is defined as

$$H(X/Y) = \sum_{x \in X,\ y \in Y} p_{XY} \log_2 \frac{1}{p_{x|y}(x \mid y)}$$

where $p_{xy}(x, y)$ is the joint probability density function of X and Y and $p_x|$ y(x, y) is the conditional probability density function of X given Y. The mutual information I (X; Y) between two random variables X and Y is a measurement of the reduction in uncertainty due to conditioning of Y, which is defined as

$$I(X;Y) = H(X) - H(X/Y)$$

$$I(X;Y) = H(Y) - H(Y/X)$$

The above equation implies that mutual information is symmetric.

8.5.2 Shannon's source coding theorem

The aim of source coding is to effectivelyexpress a random experiment's sequence of outcomes. According to Shannon's source-coding theorem "Let X be the ensemble of letters from a discrete memoryless source with finite entropy H (X). Blocks of J symbols from the source are encoded into code words of length N from a binary alphabet. For any $\varepsilon > 0$, the probability P_e of a block-decoding failure can be made arbitrarily small if

$$R \equiv \frac{N}{J} \geq H(X) + \varepsilon$$

and J is sufficiently large. Conversely, if $R \leq H(X) - \varepsilon$ then P_e becomes arbitrarily close to 1 as J is made sufficiently large. The source-coding theorem demonstrates that the source entropy H (X) is the lower bound of the average number of bits/symbols required to encode the output of a discrete memoryless source without distortion.

8.5.3 Rate-distortion theory

The task of reducing redundancy from a source while adhering to a predetermined fidelity requirement is the focus of the rate distortion theory. The maximum distortion of any source-coding scheme that satisfies the rate constraint R is the definition of the rate-distortion function represented by R(D) of the source X. The rate-distortion function for a discrete memoryless source using the mean square error criterionis given by

$$R(D) = \inf_{pY|X,\ \mathrm{MSE}(p_{Y|X}) \leq D} I(X;Y)$$

where $_{P}$Y|X implies source-coding rule and MSE the MSE of the encoding rule.

8.6 RUN-LENGTH CODING

Long sequences of the same symbol can be effectively encoded using run-length coding (RLC). By encoding the total amount of symbols in a run, run-length coding takes advantage of the spatial redundancy. The word "run" denotes the repetition of a symbol, while the word "run-length" denotes the quantity of repeated symbols. A series of numbers is converted into a series of symbol pairs (run, value) using run-length coding. For this type of compression, images with sizable regions of consistent shade make suitable candidates. The bitmap file format used by Windows employs run-length coding.There are two types of run-length coding: (i) 1D run-length coding (ii) 2D run-length coding. As opposed to 2D RLC, which uses both horizontal and vertical correlation between pixels, 1D RLC only uses the horizontal correlation between pixels on the same scan line.

8.6.1 1-DRun-length coding

Each scan line in 1D run-length coding is separately encoded. A series of alternating, independent white runs and black runs can be thought of as each scan line. The first run in each scan line is taken into consideration to be a white run as per an agreement between the encoder and decoder. The run-length of the initial white run is set to zero if the first actual pixel is black. There is end-of-line (EOL) at the conclusion of each scan line. When an EOL codeword is encountered, the decoder is aware that the scan line has ended.

Example: Consider figure 3 for an example of a binary image and its associated binary representation. Using run-length coding, the image must be encoded.Run-length coding transmits two values: the first value represents the quantity of times a specific symbol has appeared, and the second value is the sign itself. It is clear from Fig. 3 that each row is successively scanned, and the associated run-length code is delivered. An illustration of run-length coding is provided for the fourth row of the input image. The bit stream for the fourth row in the example is 4, 0, 10, 1, 4, 0.

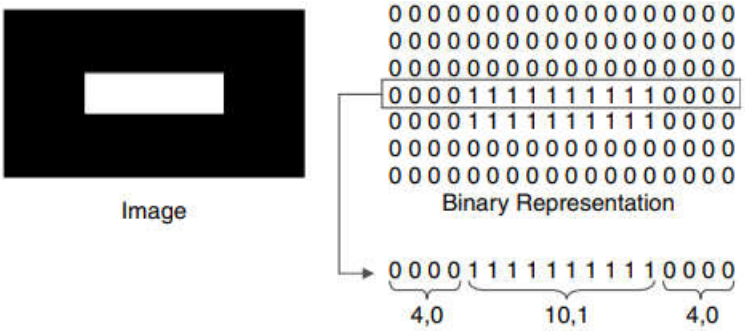


Figure 3: Example ofrun-length coding

8.6.2 2-DRun-length coding

The correlation between pixels in a scanline is used in 1D run-length coding. 2D run-length coding was created to take use of the correlation between pixels in nearby scan lines to increase coding effectiveness.

8.7 SHANNON-FANO CODING

A top-down binary tree approach, Shannon-Fano coding uses binary trees. The Shannon-Fano coding algorithm is provided below:

1. The collection of source symbols is arranged in reverse probability order.
2. The interpretation of each symbol is that of a tree's root.
3. The list is split into two groups with roughly equal overall probabilities for each category.
4. A 0 is added to the first group's code phrase.
5. The number one is added to the second group's code phrase.
6. For each group, steps (3) to (4) are repeated until there is only one node in each subset.

Example: Construct the Shanon–Fano code for the word MUMMY

Solution: The given word is MUMMY. The number of alphabets in the word MUMMY (N) is five. Step 1 Determining the probability of occurance of each character Probability of occurrence of each character in MUMMY is given by

$$P(M)=\frac{3}{5};\; P(U)=\frac{1}{5};\; P(Y)=\frac{1}{5}.$$

Step 2 Determining the number of steps Now, the number of steps to be involved in this problem is number of steps = number of symbols − 1

$$= 3 - 1 = 2$$

Step 3 Computation of the code word for each symbol the steps involved in the computation of the code words is shown in the table given below

Symbol	Probability of occurrence	Step 1	Step 2	Code
M	3/5	0		0
U	1/5	1	0	10
Y	1/5	1	1	11

Step 4 Determination of entropy

After determining the probability and the code for each character, the next step is to compute the entropy. The entropy is given by

$$H(s) = -\sum_{k=0}^{N-1} P(k)\log_2^{P(k)}$$

$$H(s) = -\frac{\sum_{k=0}^{N-1} P(k)\log_{10}^{P(k)}}{\log_{10}^{2}}$$

$$H(s) = -\frac{1}{0.3010}\left\{\frac{3}{5}\log\left(\frac{3}{5}\right)+\frac{1}{5}\log\left(\frac{1}{5}\right)+\frac{1}{5}\log\left(\frac{1}{5}\right)\right\}$$

$$H(s) = \frac{-1}{0.3010}\{0.6\times(-0.2218)+0.2\times(-0.6990)+0.2\times(0.6990)\}$$

$$H(s) = -\frac{1}{0.3010}\{-0.4127\} = 1.3711$$

Step 5 Computation of average length

The average length is given by

$$\bar{L} = \sum_{k=0}^{N-1} P(k)l(k)$$

$$\bar{L} = \left\{\frac{3}{5}\times 1+\frac{1}{5}\times 2+\frac{1}{5}\times 2\right\} = \left\{\frac{3+2+2}{5}\right\} = \frac{7}{5} = 1.4$$

Step 6 Computation of efficiency

The efficiency of the code η is given by

$$\eta = \frac{H(s)}{\bar{L}} = \frac{1.3711}{1.4} = 0.9794 = 97.94\%$$

8.8 HUFFMAN CODING

In 1952, D A Huffman created the Huffman coding system. The algorithm was created by David Huffman in 1950 while he was a student at MIT taking an information theory course. One symbol is mapped to one code word in Huffman codes, which are the best codes. It is assumed in Huffman coding that each pixel's intensity has a specific probability of occurrence, and that this probability is spatially invariant. Each intensity value is given a binary code through Huffman coding, with shorter codes going to intensities with higher probabilities. The Huffman codes table can be fixed in both the encoder and decoder if the probabilities can be calculated a priori. If not, the compressed image data must also be transmitted with the coding table to the decoder. Entropy, Average length, Efficiency, Variance are some of the parameters involved in Huffman coding.

In the first step of Huffman's method, the probabilities of the symbols under consideration are ranked, after which the lowest probability symbols are combined into a single symbol and substituted in the following source reduction. This binary coding technique is seen in figure 4. A hypothetical collection of source symbols and their probabilities are arranged from top to bottom in decreasing order of probability values at the far left. The lowest two probabilities, 0.06 and 0.04, are combined to create a "compound symbol" with probability 0.1 to create the initial source reduction.In order to rank the probabilities of the reduced source from most to least likely, this compound symbol and its corresponding probability are placed in the first source reduction column. Then, this process is continued until the reduced source (at the far right) has two symbols.

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Figure 4: Huffman source reductions

The coding of each reduced source, proceeding backwards from the smallest to the largest, is the second stage in Huffman's method. The symbols 0 and 1 are, of course, the smallest length binary code for a two-symbol source. These symbols are given to the two symbols on the right, as seen in Fig. 5 (The assignment is arbitrary; it would also be effective to reverse the order of the 0 and 1). The 0 used to code the reduced source symbol with probability 0.6 is now assigned to each of these symbols, and a 0 and 1 are arbitrarily appended to each to distinguish them from one another. This is because the reduced source symbol with probability 0.6 was created by combining two symbols in the reduced source to its left.This operation is then repeated for each reduced source until the original source is reached. The final code appears at the far left in Fig. 5. The average length of this code is

$$\begin{aligned} L_{avg} &= (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5) \\ &= 2.2 \text{ bits/pixel} \end{aligned}$$

and the entropy of the source is 2.14 bits/symbol.

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Figure 5: Huffman code assignment procedure

When symbols are coded one at a time, Huffman's method finds the best possible code for a collection of symbols and probabilities. Once the code has been written, it may be easily decoded without error using a simple search table. The code is a block code that can be instantly and exclusively decoded. Because each source symbol is translated into a predetermined order of code symbols, it is known as a block code. Because each code word in a string of code symbols may be deciphered without referring to subsequent symbols, it is instantaneous.It can only be decoded in one method for each given string of symbols, making it uniquely decodable. As a result, any string of Huffman-encoded symbols may be decoded by looking at each symbol in the string from left to right.For the binary code of Fig. 5, a left-to-right scan of the encoded string 010100111100 reveals that the first valid code word is 01010, which is the code for symbol a_3. The next valid code is 011, which corresponds to symbol a1. Continuing in this manner reveals the completely decoded message to be a_3 a_1 a_2 a_2 a_6.

8.9 ARITHMETIC CODING

In arithmetic coding, which dates back to Elias's work (Abramson [1963]), source symbols and code words do not always correlate exactly. Instead, a single arithmetic code word is given to an entire series of source symbols (or a message). By itself, the code word designates a range of real values between 0 and 1. The interval used to represent the message's increasing number of symbols gets smaller as well as the quantity of information units (let's say, bits) needed to do so.The size of the gap is decreased for each symbol in the message in line with the likelihood that it will appear. The technique achieves (but only in theory) the bound set by Shannon's first theorem since it does not require, as does Huffman's approach, that each source symbol translate into an integral number of code symbols (i.e., that the symbols be coded one at a time).

The procedure of fundamental arithmetic coding is shown in Figure 6. Here, $a_1a_2a_3a_3a_4$, a five-symbol sequence or message, is coded from a four-symbol source. The message is first believed to take up the entire half-open interval [0, 1) before the coding process even begins. This interval is originally partitioned into four sections depending on the probabilities of each source symbol, as shown in Table 1. For instance, the symbol a1 is related to the subinterval [0, 0.2).The message interval is initially reduced to [0, 0.2) because it is the first symbol of the message that is being coded.

As a result, in Fig. 6, the range [0, 0.2) is expanded to fill the entire height of the figure, with the values of the restricted range designating its end points. The next message symbol is then added to the narrower range, and the procedure is repeated according to the probability of the origin source symbol.In this manner, symbol a2 narrows the subinterval to [0.04, 0.08), a3 further narrows it to [0.056, 0.072), and so on. The range is reduced to [0.06752, 0.0688) by the last message symbol, which must be reserved as a unique end-of-message signal. Of course, the message can be represented by any integer falling inside this subinterval, such 0.068. The five symbols of the arithmetically coded message shown in Fig. are represented by three decimal digits. In comparison to the source's entropy, which is 0.58 decimal digits per source symbol, this equates to 0.6 decimal digits per source symbol.The resulting arithmetic code approaches the bound defined by Shannon's first theorem as the length of the sequence being coded grows. The use of finite precision arithmetic and the addition of the end-of-message indicator, which is required to distinguish one message from another, are the two variables that cause coding performance to fall short of the bound in practise. The latter issue is addressed by scaling and rounding strategies introduced in practical arithmetic coding systems. The scaling approach divides each subinterval in accordance with the symbol probabilities after renormalizing it to the [0, 1) range. The rounding technique ensures that the coding subintervals are faithfully represented despite the truncations brought on by finite precision arithmetic.

Table 1:Arithmetic coding example

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

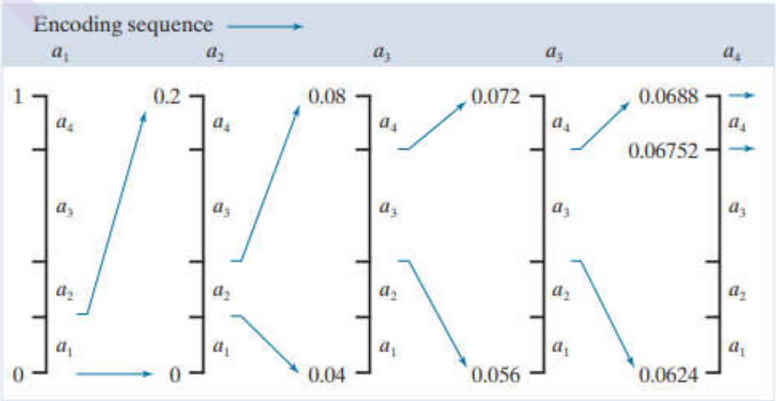


Figure 6:Arithmetic coding procedure

8.10 TRANSFORM-BASED COMPRESSION

The method most frequently employed in image-coding applications is image-transform coding. A set of linear transform coefficients, which are typically scalar-quantised and entropycoded for transmission, are created by linearly transforming an image. Transform coding, then, is a mathematical process that divides a large number of highly correlated pixels into a smaller set of uncorrelated coefficients. The foundation of transform coding is the existence of a specific degree of connection between pixels in an image. When compressing images, transformation is a very helpful technique. It is used to convert time-domain image data to frequency-domain data. The spatial redundancy in the time domain can be reduced by converting the data into frequency domain.The advantage of using transformation is that the energy of the transformed data is mainly condensed in the lowfrequency region, and is represented by a few transform coefficients. Thus, most of these coefficients can be discarded without significantly affecting the reconstructed image quality. In a transform coder, the discrete data signal is segmented into non-overlapping blocks, and each block is expressed as a weighted sum of discrete basis functions. The purpose of transform coding is to decompose the correlated signal samples into a set of uncorrelated transform coefficients, such that the energy is concentrated into as few coefficients as possible. The block diagram of transform-based image coding scheme is shown in Fig. 7.

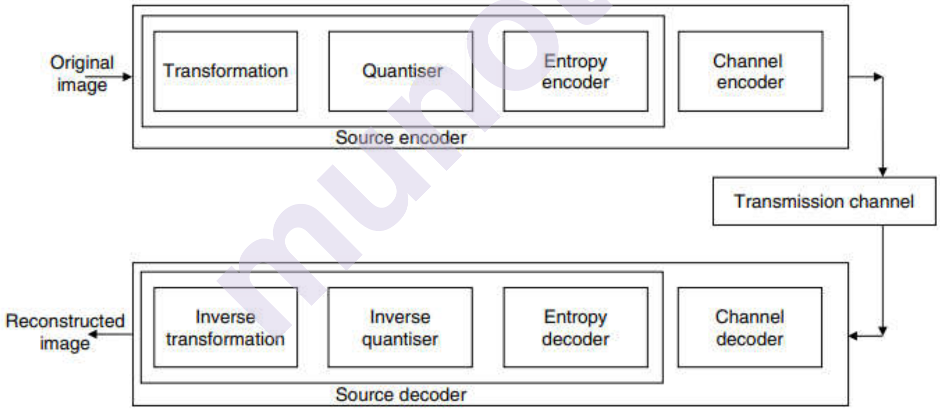


Figure 7:Transform-based image-coding scheme

Transformation: The image's correlation qualities are altered as a result of the transformation's "reorganization" of the grey values. The process of transformation condenses the image data into a tiny set of coefficients. The transform should have the following characteristics for effective compression:

- Decorrelation: The transform should generate less correlated or uncorrelated transform coefficients to achieve high compression ratio.
- Linearity: Linearity principle allows one-to-one mapping between pixel values and transform coefficients.
- Orthogonality: Orthogonal transforms have the feature of eliminating redundancy in the transformed image

Quantization: It is the process of limiting a quantity's range of possible values in order to minimize the number of bits required to represent it. By itself, the picture transformations don't lead to any compression. Instead, they represent the image in a distinct domain where the image data is divided into parts with varied degrees of significance. In essence, quantization is an irreversible process.

Entropy encoder: The reduction of the number of bits needed to represent each symbol at the quantizer output is the goal of an entropy encoder. Techniques for entropy-coding that are frequently employed include run-length coding, arithmetic coding, and Huffman coding. A lossless coding system is essentially what entropy coding is.

Essentially, the encoder's output is a bit stream that is sent over the channel to the decoder. If one is just focused on source code, the channel is typically taken to be lossless. If one is interested in joint source and channel coding, one should be aware that the mistake happens when the bitstream traverses the channel. In essence, the decoder reverses the encoder's process to produce the reconstructed image.

8.11 SUMMARY

- Compact representation is compression. The representation of a picture with the fewest possible bits is known as image compression.

- Redundancies in an image are reduced to accomplish compression. Spatial redundancy, temporal redundancy, and psychovisual redundancy are three different types of redundancies in an image.

- Image compression can be divided into two categories: lossless compression and lossy compression. The rebuilt image closely resembles the original image when using lossless compression. Information is lost during lossy compression.

- These methods are frequently used in lossless picture compression: Run-length coding, Huffman coding, arithmetic coding, Shannon-Fano coding, and dictionary-based methods like LZW are only a few examples.

- Quantization is basically approximation. The process of quantization is non-linear and irreversible. Scalar quantization and vector quantization are the two basic categories under which quantization may be classified.

8.12 LIST OF REFERENCES

1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009.

2) Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008.

3) Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. Esakkirajan and T. Veerakumar, 2016 (https://scilab.in/textbook_companion/generate_book/125).

8.13 UNIT END EXERCISES

1) Explain the need for image compression.
2) Describe Redundancy in images.
3) Explain Image-compression scheme.
4) What is Entropy and mutual information?
5) Explain Shannon's source coding theorem.
6) Describe Rate-distortion theory.
7) What is Run-length coding? Explain along with its types.
8) Write a note on Shannon-Fano coding.
9) Describe Huffman Coding.
10) Explain Arithmetic Coding.
11) Explain Transform-based compression.

❖❖❖❖