INF-3056

GAUHATI UNIVERSITY

Centre for Distance and Online Education

Third Semester

(Under CBCS)

M.Sc.-IT

Paper: INF 3056

IMAGE PROCESSING

Contents:

SLM Development Team:

HoD, Department of Computer Science, GU
Programme Coordinator, M.Sc.-IT, GUCDOE
Prof. Shikhar Kr. Sarma, Department of IT, GU
Dr. Khurshid Alam Borbora, Assistant Professor, GUCDOE
Dr. Swapnanil Gogoi, Assistant Professor, GUCDOE
Mrs. Pallavi Saikia, Assistant Professor, GUCDOE
Dr. Rita Chakraborty, Assistant Professor, GUCDOE
Mr. Hemanta Kalita, Assistant Professor, GUCDOE

Contributors:

Mrs. Pallavi Saikia Asstt. Prof., GUCDOE	(Block 1 : Units- 1, 2, 3 & 4) (Block 2 : Unit- 2)
Dr. Minakshi Gogoi Prof. & Head, Dept. of Computer Science & Engineering GCU, Guwahati	(Block 1 : Units- 5, 6, 7 & 8)
Mr. Hemanta Kalita Asstt. Prof., GUCDOE	(Block 2 : Unit- 1) (Block 3 : Units- 1, 2 & 3)
Mr. Risheraj Baruah Asstt. Prof., KKHSOU	(Block 2 : Unit- 3) (Block 2 : Unit- 6)
Dr. Sanjib Kr. Kalita Associate Prof., Dept. of Computer Science Gauhati University	(Block 2 : Units- 4 & 5)
Mr. Devaraj Mahanta JRF, Deptt. of Mathematics & Computational Sciences, IASST	(Block 3 : Units- 4, 5 & 6)

Course Coordination:

Dr. Debahari Talukdar	Director, GUCDOE
Prof. Anjana Kakoti Mahanta	Programme Coordinator, GUCDOE Dept. of Computer Science, G.U.
Dr. Khurshid Alam Borbora	Assistant Professor, GUCDOE
Dr. Swapnanil Gogoi	Assistant Professor, GUCDOE
Mrs. Pallavi Saikia	Assistant Professor, GUCDOE
Dr. Rita Chakraborty	Assistant Professor, GUCDOE
Mr. Hemanta Kalita	Assistant Professor, GUCDOE
Mr. Dipankar Saikia	Editor SLM, GUCDOE

Content Editor:

Dr. Kshirod Sarmah	HOD & Assistant Professor, Dept. of Computer Science, PDUAM, Amjonga, Goalpara

Cover Page Designing:

Bhaskar Jyoti Goswami	GUCDOE
Nishanta Das	GUCDOE

ISBN: 978-81-989447-1-9

October, 2024

Published on behalf of Gauhati University Centre for Distance and Online Education by the Director, and printed at Gauhati University Press, Guwahati-781014.

CONTENTS:

BLOCK: I

Unit 1: Fundamentals of Digital Image Processing
Unit 2: Visual Perception
Unit 3: Image Sensing and Acquisition
Unit 4: Image Sampling and Quantization
Unit 5: Basic Concepts of Image Transform
Unit 6: Image Transform- I
Unit 7: Image Transform- II
Unit 8: Intensity Transformation

UNIT: 1
FUNDAMENTALS OF DIGITAL IMAGE PROCESSING

Unit Structure:

1.1 Introduction

The process of manipulating an image for its feature enhancement and also to extract valuable information from it is called Digital Image Processing. Two main applications which led to the interest in Digital Image Processing (DIP) are: improvement of pictorial information for human interpretation and processing of image data for storage, transmission and representation for storage, transmission and representation for autonomous machine perception. Image processing is one of the growing technologies which has a core area in research.

1.2 Objectives

After going through this unit, you will learn-

- *what* is Digital Image Processing?

- *different* applications of Digital Image Processing
- *fundamental* Steps in Digital Image Processing
- *components* of Digital Image Processing Systems

1.3 What is Digital Image Processing?

Digital Image Processing refers to the processing of digital images by means of digital computer is referred to as Digital Image Processing. A digital image consists of a finite number of elements where each one of them has a certain value and position. These elements are referred to as picture elements, pels or pixels. The most widely used term is the pixel. An image may be defined as a two-dimensional function that represents a measure of some characteristic such as brightness or colour of a viewed scene. Mathematically an image can be defined as a two-dimensional function, f(x,y) where x and y are spatial(plane) co-ordinates and amplitude of f at any pair of co-ordinates (x,y) is called the intensity or gray level of the image at that point. Gray level refers to the amount of light falling at an object or the amount of reflection by an object. When the values of x,y and the intensity values of f are all finite and discrete then that image is referred to as digital image.

Digital images can be divided into two types:

- Black and white images: Black and white images are made from different shades of gray. These different shades lie between 0 to 255, where 0 refers to black, 255 refers to white and intermediate values ask different shades of black and white. Gray scale relates to the range of neutral tonal values (shades) of black to white. A normal grayscale image contains 8 bits/pixel data, which has 256 different grey levels. In medical images and astronomy, 12 or 16 bits/pixel images are used.

- Colour images: Colour images are made from coloured pixels. Colour can capture a way broader range of values than grayscale. “The spectrum – the collection of colours created when light passes through a prism – includes billions of colours, of which the human eye can recognize 7 to 10 million”. The electronic capture and display of colour are complicated. RGB (Red, Green, and Blue) is that the most ordinarily adopted colour system. Each color image has 24 bits/pixel means 8 bits for each of the three color band (RGB).

Stop to Consider

The first application of digital images was in the newspaper industry where pictures were first sent through submarine cable between London and New York

Check Your Progress

1. Define pixel
2. What is the difference between grayscale and colour images?

1.4 Applications of Digital Image Processing

Digital Image Processing is now being used in a variety of fields for information extraction.

- Medical Image Retrieval: MRI image are being processed for different medical diagnosis such as brain analysis for Alzheimer’s disease, heart diseases and many more. Analysing MRI images can help the medical experts to detect the pattern of certain diseases and they can take preventive measures earlier. It also helps in detecting diseases like cancer in different organs of a body. It has paved the way for X-ray Imaging, PET Scans, UV Imaging, Medical CT, Cancer Cell Image Processing, and much more

- Remote Sensing: Remote sensing is the process of acquiring information about an object, phenomenon or an area from a remote space using devices like aircraft or satellite. It uses sensor to capture the different images.
- Robotics: Robotics is the use of robots that perceive, understand and interact with the environment. It aims to capture or extract useful information from images, such as object detection, recognition, classification, segmentation, tracking, pose estimation, depth estimation, and scene understanding.
- Image restoration and sharpening: Image restoration is the technique of recovering an image that has been degraded, corrupted or consist of noise. The recovered image should be of high quality with it's original content intact. Image restoration requires prior knowledge of the degradation model that has degraded the image. Image sharpening means to apply effect to give a sharper appearance in an image. It enhances the overall representation of the image highlighting the edges and the fine details of an image.
- Image Reconstruction: Image Reconstruction is the technique of constructing the images that have missing parts that needs to be filled or that are corrupted. This is done by training extensively the image processing software with a lot of images of some other dataset. The knowledge gained through training, can be used to reconstruct the corrupted image or

generate a new image of that type. These are only a few applications of image processing software discussed here because the scope of application of image processing is a huge area.

Check Your Progress
3) Give an application of image processing other than given in the unit.

1.5 Steps in Digital Image Processing

The fundamental steps in Digital Image Processing can be divided into two categories:

1) Methods whose input and output are images.

2) Methods whose outputs are attributes extracted from images.

The diagram of the fundamental steps of image processing depicted in Figure 1 doesn't imply that every method is applied to an image rather it gives an idea that all of these methodologies can be applied to an image according to the application or the objective defined.

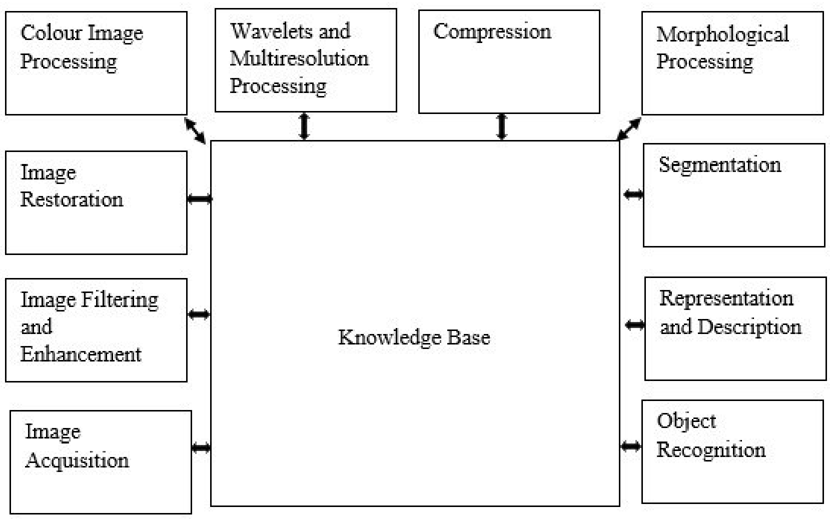


Figure 1: Fundamental Steps in Image Processing

These operations are discussed briefly now as in later chapters it will be explained in details:

- Image acquisition: Image acquisition is the process of retrieving images from different hardware-based sources like cameras, encoders and sensors. Image processing is considered as the first step in image processing as without acquiring the image, there can be no processing. This step also includes some type of pre-processing that will be discussed later in this book.
- Image Filtering and Enhancement: Image filtering and enhancement refers to the manipulation of images using different operations to enhance the quality of the image for using it for a particular application. Image enhancement techniques are problem-oriented. A technique that is suitable for enhancing a satellite image may not be suitable for enhancing X-ray images. There are a wide variety of techniques for image filtering and enhancement.
- Image Restoration: Image restoration is the process of improving the quality of the image if it is degraded and contains noise. This is done to obtain a high-quality image from a corrupted input image to make it applicable for certain purposes. Image restoration techniques are based on mathematical or probabilistic models of image degradation.
- Colour Image Processing: Colour image processing is the process of analysing visual data presented in colour. It also forms the basis for extracting features of interest in an image. Colour image processing got importance when the use of digital images over internet increased significantly.

- Wavelets and Multiresolution Processing: These are the foundation for representing image in various degrees of resolution.
- Image Compression: Image Compression is the technique of reducing the space required to save an image or to reduce the rate at which the data transfer takes place over the network.
- Morphological Processing: Morphological image processing is a collection of non-linear operations for extracting image components that are useful for representing and describing an image.
- Segmentation: Image segmentation is the process of partitioning an image into its constituent parts based on grouping up of pixels having the same attributes. Image segmentation helps in improving the accuracy of an object detection model.
- Representation and Description: It always follows the output of segmentation step that is, raw pixel data, constituting either the boundary of an image or points in the region itself. In either case converting the data to a form suitable for computer processing is necessary
- Object Recognition: Recognition is the process of assigning label to an object based on the features or attributes of the object.
- Knowledge Base: Knowledge about a particular problem or a specific task is incorporated in an Image Processing System in the form of a knowledge database. This can be simple if the knowledge about the location of the area of interest is known thus reducing the task of searching it. But at the same time, it

may be complex when the image database consists of high-resolution satellite image.

1.6 Components of an Image Processing System

The components of a general-purpose image processing system are depicted in Fig 2:

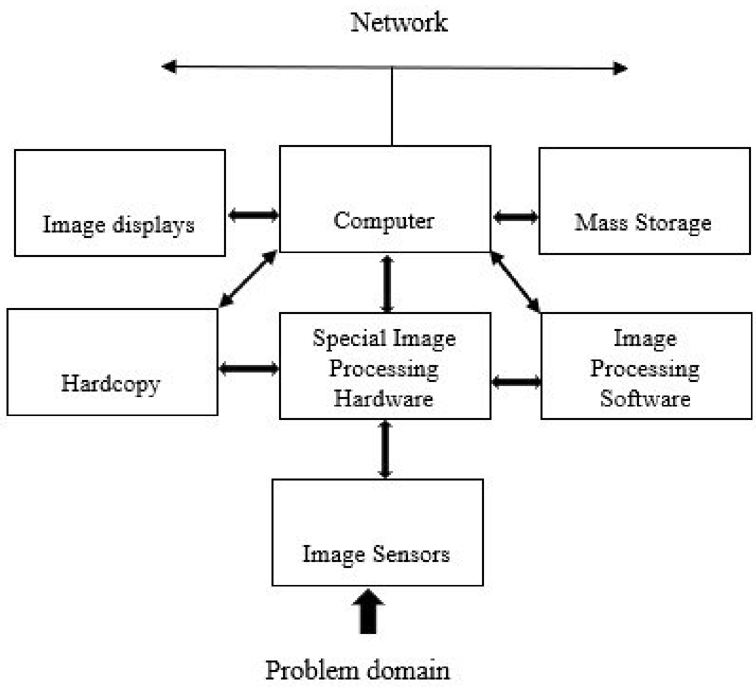


Figure 2: Components of Image Processing System

Let us discuss briefly the different components of an image processing system:

- Image Sensors: Image sensors senses or takes input all the characteristics and features of an image and passes it to the next block i.e. Specialized Image Processing hardware. Image sensors mainly consists of two devices: a physical device that is sensitive to the energy radiated by the object we wish to image and a digitizer to convert the output of the physical device into digital form. It includes the problem domain.

- Specialized Image Processing Hardware: This type of hardware works as a front subsystem and performs tasks that require very fast data throughputs which a computer may not be able to handle. It may be any unit that performs some primitive operations like the ALU that performs arithmetic and logic operations.

- Computer: Based on the image processing application, computers may range from a simple PC that is used in our daily lives to a supercomputer with a very high-level performance.

- Image processing Software: A well-defined package of an Image processing software consists of specialized modules along with different mechanisms as well as algorithms to perform specific tasks for image processing. A user can also write codes for a well-defined package but the user must be well versed in writing such specialized modules for processing images.

- Mass Storage: Mass storage capability is a pre-requisite of any image processing system asmillions of images needs to be processed where the intensity of each pixel of an image isa 8 bit quantity. The storage can be farther divided into three principal categories:

  a) storage that can be used during processing
  b) on-line storage for data retrieval
  c) archival storage that is not a

- Image Display: The processed images are displayed using a monitor that are driven by graphics display cards that form an integral part of any computer.

- Hardcopy: After the image is being processed, it needs to be stored in a hardware device.

- Networking: In every image processing application, large number of images is generated. Amount of data is not an issue but for efficient data transmission to remote site depends on the bandwidth. With optical fibre and other broadband technologies, the data transmission is becoming efficient and reliable.

In this chapter we have discussed some of the initial concepts briefly to get a overview of the concepts that are going to be taught in this subject. In the later chapters each of the concepts and also some new concepts will be introduced and explained in a detailed manner.

1.7 Summing Up

- Digital Image Processing refers to the processing of digital images by means of digital computer is referred to as Digital Image Processing.
- An image is composed of a finite number of elements where each of the elements has a value and a position. These elements are referred to as pixel.
- Digital images are of two types: Grayscale images and Coloured images
- Image processing is widely used in almost every field like medical image retrieval, robotics, remote sensing and many more.
- The different tasks of digital image processing consist of image acquisition, image filtering and enhancement, image restoration, colour image processing, wavelets and multiresolution processing, image compression, morphological processing, segmentation, object recognition and knowledge base.

- A digital image processing system consists of the following components: image sensors, specialized image processing hardware, computer, image processing software, mass storage, image display, hardcopy and network devices.

1.8 Answers to Check Your Progress

1. An image is composed of a finite number of elements where each of the elements has a value and a position. These elements are referred to as pixel.

2. A coloured image has three colour channels red, green and blue whereas a grayscale image has only one colour channel.

3. An application of image processing is pattern recognition. Pattern recognition is used in handwriting recognition, image recognition and also in computer aided diagnosis of different disease

1.9 Possible Questions

1) Define an image
2) What is a pixel?
3) What are the two types of digital image?
4) What is a grayscale image?
5) What is a RGB image?
6) Mention two applications of image processing?
7) How image processing is used in the medical field?
8) What is image acquisition?
9) What is knowledge base in this context?
10) What is an image sensor?
11) What is Digital Image Processing? What are the different applications of Digital Image Processing?

12) Explain the different steps carried out in the process of digital image processing.

13) What are the different components of a Digital Image Processing System? Explain.

1.10 References and Suggested Readings

1. Gonzalez, R. C. (2009). *Digital image processing*. Pearson education India.

*****

UNIT: 2
VISUAL PERCEPTION

Unit Structure:

2.1 Introduction

Visual perception is the process by which a human eye perceives the visual stimulus through the light that enters the eyes. Before having the knowledge about images it's very important to understand the visual perception. The study of visual perception is very crucial for the designers to develop an image processing user interface because it should be in relation to the colours, patterns, and structures perceived through the human eyes.

2.2 Objectives

After going through this unit, you will learn-

- *elements* of visual perception,
- *the* structure of the human eye,
- *how* images are formed in the human eye,

- *the* meanings of brightness, adaptation and discrimination in the context of image processing,
- *what* is light and electromagnetic spectrum.

2.3 Elements of Visual Perception

Human intuition and analysis play a very important role in the field of image processing as it helps in understanding better about the techniques and decide which technique is better than the other, though mathematics and probabilistic formulation forms the backbone of the subject. The choice of the technique is often based on subjective and visual judgements. The three main elements of visual perception are:

- Structure of the human eye
- Image formation in the human eye
- Brightness Adaptation and Discrimination

2.3.1Structure of the Human Eye

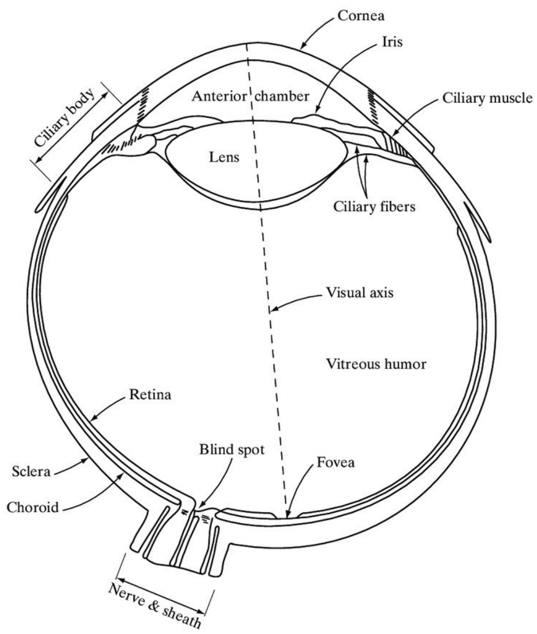


Fig 1: Simplified diagram of a cross section of the human eye

Figure 1shows a simplified horizontal cross section of the human eye. The eye is nearly a sphere, with an average diameter of approximately 20 mm. Three membranes enclose the eye: the cornea and sclera outer cover; the choroid; and the retina. The cornea is a tough, transparent tissue that covers the anterior surface of the eye. Continuous with the cornea, the sclera is an opaque membrane that encloses the remainder of the optic globe. The choroid lies directly below the sclera. This membrane contains a network of blood vessels that serve as the major source of nutrition to the eye. The choroid coat is heavily pigmented and hence helps to reduce the amount of extraneous light entering the eye and the backscatter within the optic globe. At its anterior extreme, the choroid is divided into the ciliary body and the iris. The latter contracts or expands to control the amount of light that enters the eye. The central opening of the iris (the pupil) varies in diameter from approximately 2 to 8 mm. The front of the iris contains the visible pigment of the eye, whereas the back contains a black pigment. The lens is made up of concentric layers of fibrous cells and is suspended by fibres that attach to the ciliary body. It contains 60 to 70% water, about 6% fat, and more protein than any other tissue in the eye. The lens is coloured by a slightly yellow pigmentation that increases with age. In extreme cases, excessive clouding of the lens, caused by the affliction commonly referred to as cataracts, can lead to poor colour discrimination and loss of clear vision. The lens absorbs approximately 8% of the visible light spectrum, with relatively higher absorption at shorter wavelengths. Both infrared and ultraviolet light are absorbed appreciably by proteins with in the lens structure and, in excessive amounts, can damage the eye. The innermost membrane of the eye is the retina, which lines the inside of the wall's entire posterior portion. When the eye is properly focused, light from an object outside the eye is imaged on the retina. Pattern vision is afforded by the distribution of discrete light receptors over the surface of the retina. There are two classes of receptors: cones and rods. The cones in each eye number between 6 and 7 million. They are located primarily in the central portion of the retina, called the fovea, and are highly sensitive to colour. Humans can resolve fine details with these cones largely because each one is connected to its own nerve end. Muscles controlling the eye rotate the eyeball until the image of an object of interest falls on the fovea. Cone vision is called photopic or bright-light vision. The number of rods is much larger: Some 75 to 150 million are distributed over the retinal surface. The larger area of distribution and the fact that several rods are connected to a single nerve end reduce the amount of detail discernible by these receptors. Rods serve to give a general, overall picture of the field of view. They are not involved in colour vision and are sensitive to low levels of illumination. For example, objects that appear brightly coloured in daylight when seen by moonlight appear as colourless forms because only the rods are stimulated. This phenomenon is known as scotopic or dim-light vision. The fovea is a circular indentation in the retina of about 1.5mm in diameter. We can view or interpret fovea as a square sensor array of size 1.5mm x 1.5mm. The density of cones in that area of the retina is approximately 150,000 elements per mm^2. Based on these approximations, the number of cones in the region of highest acuity in the eye is about 337,000.

2.3.2 Image Formation in the Human Eye

After going through the structure of the eye in the previous section we know that the cones and the rods are mainly responsible for the formation of an image in the human eye. The lens of an eye is more flexible compared to an ordinary photographic camera. When an eye focusses on an object, the image is formed on the retina. The lens of a human eye is controlled by the tensions of the fibres of the ciliary body. Whenever the lens of the eye focusses a faraway object the fibre muscles flattens the shape of the lens and when it focusses on a nearby object, it thickens the lens. Depending on the distance between the object and the lens, the shape of the lens changes. The distance between the centre of the lens and the retina is called the focal length and it ranges from 14mm to 17mm.The strongest refractive power of the lens is achieved if the object is nearby and the lowest refractive power is achieved if it is faraway. Let us now see how to retrieve the dimension of an image that is formed in the retina with the help of the figure 2.

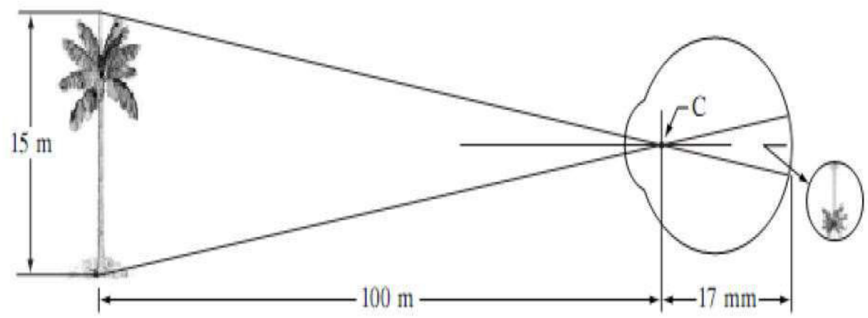


Fig 2:Graphical representation of the eye looking at a palm tree

Suppose a person is focussing on an object from a distance of 100m. Here the object considered is a tree of height 15m. Let c be the centre of the lens and h be the height of the tree that is formed in the retina. The calculation is done as follows:

$$15/100 = h/17$$

Hence we get $h = 2.55$

Fovea is the region where primarily the retinal image is formed.

Check Your Progress

1. The __________ is a tough, transparent tissue that covers the anterior surface of the eye.

2. The innermost membrane of the eye is the ______, which lines the inside of the wall's entire posterior portion.

3. The two classes of receptors are __________ and ___________

4. The lens of a human eye is controlled by the tensions of the fibres of the _________

5. Define focal length.

2.3.3 Brightness Adaptation and Discrimination

The ability of the eye to adapt itself with the different brightness changes is called the brightness adaptation and the ability of the eye to discriminate between the different intensity levels of brightness is called discrimination. The human visual system can adapt itself to an enormous range of intensity levels which is in order of 10^{10}. This range is from the scotopic threshold to the glare limit. Experimentally subjective brightness which is the intensity perceived by the human visual system, is a logarithmic function of the light intensity incident on the eye. The characteristics of light intensity versus subjective is shown in figure 3. In the figure the long curve shows the range of the intensity levels to which a human eye can adapt. The range of photopic alone is in the range of 10^6. But the human visual system cannot perceive over this range simultaneously. So, it adjusts itself to this range by changing its overall sensitivity and the phenomenon is called brightness adaptation.

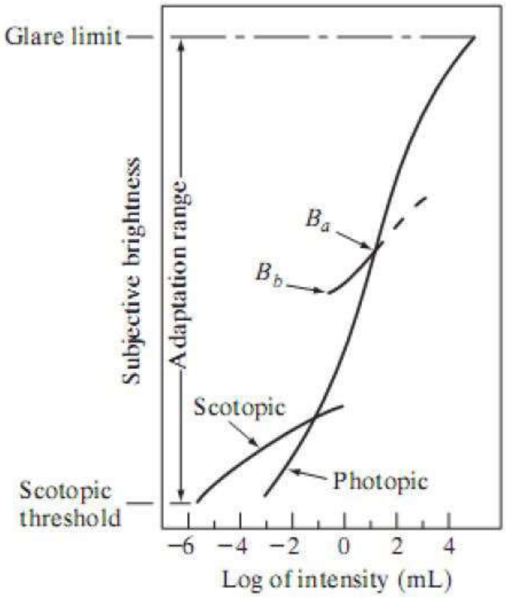


Fig 3: Range of subjective brightness sensations showing a particular adaptation level

The phenomenon that describes the ability of the human eye to discriminate the different distinct intensity levels is called brightness adaptation. For any given set of conditions , the current sensitivity levels of the visual system are called the brightness adaptation level. In figure 3, $B_i$as the brightness adaptation level up to which a human eye can perceive and all other intensity levels at and below B_b is considered as indistinguishable blacks.

Brightness discrimination is the ability of the human eye to distinguish between different light intensity levels. At any specific adaptation level, brightness discrimination of the eye is different. Brightness discrimination is given by the Weber ratio $\Delta I_c/I$ where ΔI_c is the increment of illumination discriminable 50% of the time with background illumination I. Small $\Delta I_c/I$ gives good brightness discrimination whereas large $\Delta I_c/I$ gives poor brightness discrimination. The above fact is represented diagrammatically in figure 4 where a plot of log $\Delta I_c/I$ as a function of log I is shown.

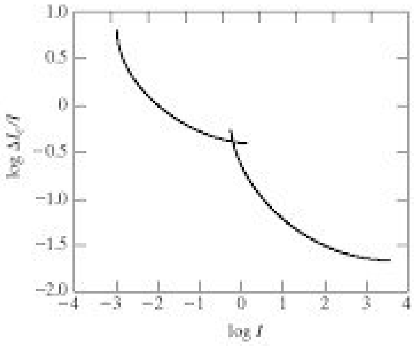


Fig 4: Typical Weber ration as a function of intensity

Check Your Progress

6. What is Weber ratio?

7. Define brightness adaptation.

2.4 Light and the Electromagnetic Spectrum

When a beam of sunlight passes through a glass prism, the reflected light is not white but a continuous spectrum of colors ranging from violet at one end and red at the other end. This was discovered back in 1966 by Sir Issac Newton. The human eye can perceive a very few range of colors that comprises only a small portion of the electromagnetic spectrum. Fig 5 illustrates the Electromagnetic Spectrum.

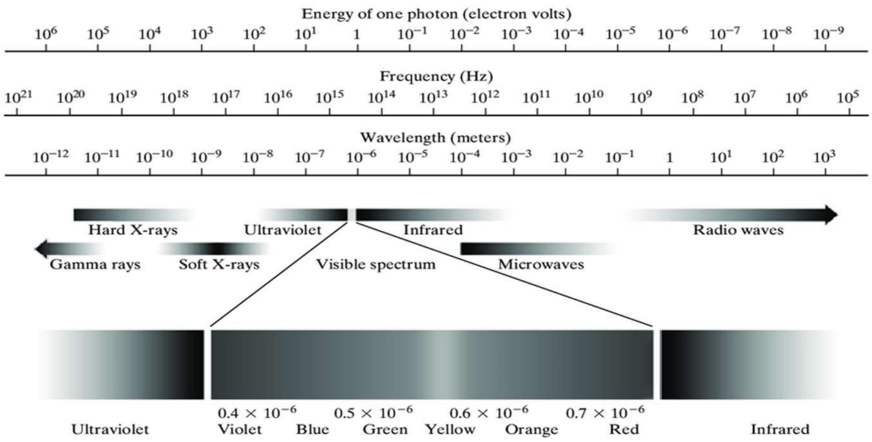


Fig 5: The Electromagnetic Spectrum

The two ends of the spectrum consist of radio waves having wavelengths billions of times longer than those of visible light and gamma rays having wavelengths millions of times smaller than the visible light. Electromagnetic spectrum can be expressed in terms of wavelength(λ), frequency(v) or energy and is given by:

$$\lambda = c/v$$

where c is the speed of light (2.998 x 10^8m). The energy of the various components of the electromagnetic spectrum is given by the expression

$$E = hv$$

where h is Planck's constant. The unit of wavelength is meter, frequency is hertz and energy is electron-volt.

Electromagnetic waves can be visualized as a linear superposition of sinusoidal waves with wavelength or can be assumed as a massless particle that travels in the speed of light. The energy possessed by a massless particle is called photon. As we know that energy is proportional to frequency, so higher the frequency more energy is possessed per photon. So, if we go through fig 5 we can see that the radio waves have the lowest energy, microwaves have more energy than radio-waves, infrared have more energy than that of radio-waves and as we go towards left we will find that Gamma rays is the most energetic of all.

As we can see in figure 5, the range of the visible spectrum or the color spectrum of the electromagnetic spectrum is from 0.43µm to about 0.79µm. The visible spectrum is mainly divided into six main regions: violet, blue, green , yellow, orange and red and every color's range blends smoothly to the next color's range thus preventing no color to stop abruptly. The light that is reflected from an object is perceived by the human eye. A human eye sees the white color if the light reflected from an object has all equal visible wavelengths. Every color has a range in which its wavelength varies and hence provides different shades of the same colour.

A light that has no colour is called monochromatic and its only attribute is its intensity. This intensity varies from black to gray and finally to white. This is termed as the gray level. An image whose intensity ranges from back to white is termed as a grayscale image. We know that the electromagnetic spectrum of chromatic or colour light ranges from approximately 0.43 to 0.79µm. The three main attributes of a chromatic light other than frequency is: radiance, luminance and brightness. Radiance is the total amount of energy that flows from the light source, and it is usually measured in watts (W). Luminance, measured in lumens (lm), gives a measure of the amount of energy an observer perceives from a light source. For example, light emitted from a source operating in the far infrared region of the spectrum could have significant energy (radiance), but an observer would hardly perceive it; its luminance would be almost zero. Brightness is a subjective descriptor that is practically impossible to measure. It embodies the achromatic notion of intensity and is one of the key factors in describing colour sensation.

As we can see that in the left end of the electromagnetic spectrum are the Gamma rays, X-rays and Ultraviolet rays. Gamma rays are used both in medical and astronomical imaging. Medical applications include imaging techniques of Positron Emission Tomography (PET) and radiation therapies for diagnosis of cancerous cells and organ. In astronomical imaging it is used in nuclear environments. Gamma rays' transitions into X-rays which are used mainly for detecting breakage of bones, airport security for baggage scanning as well as also in astronomical purposes. X-rays further transitions into Ultraviolet rays. Ultraviolet rays are used for many purposes like killing harmful bacteria present in water through water purifiers, for fake currency note detection and also for photograph of fingerprints.

After the visible spectrum lies the infrared rays which is used for remote controls and communication, astronomers use it to observe objects in space which can't be detected by the human eye. The infrared ray blends with the microwave which is used in satellite communication. At the right end of the electromagnetic spectrum we can see the radio waves. The main application of radio waves encompasses television as well as AM and FM radios. Radio waves are also used in astronomical observations.

Stop to Consider

The full range of electromagnetic spectrum is used by the NASA's instruments to study the Earth, solar system and the universe beyond.

2.5 Summing Up

- Visual perception is the process by which a human eye perceives the visual stimulus through the light that enters the eyes.
- The three main elements of visual perception are: Structure of the human eye
- Image formation in the human eye, Brightness Adaptation and Discrimination
- The eye is nearly a sphere, with an average diameter of approximately 20 mm.
- Three membranes enclose the eye: the cornea and sclera outer cover; the choroid; and the retina.
- When the eye is properly focused, light from an object outside the eye is imaged on the retina.
- Pattern vision is afforded by the distribution of discrete light receptors over the surface of the retina.
- There are two classes of receptors: cones and rods.
- The cones and the rods are mainly responsible for the formation of an image in the human eye.

- When an eye focusses on an object, the image is formed on the retina.
- The lens of a human eye is controlled by the tensions of the fibres of the ciliary body.
- The distance between the centre of the lens and the retina is called the focal length and it ranges from 14mm to 17mm.
- The ability of the eye to adapt itself with the different brightness changes is called the brightness adaptation and the ability of the eye to discriminate between the different intensity levels of brightness is called discrimination.
- Brightness discrimination is given by the Weber ratio $\Delta I_c/I$ where $\Delta I_c$is the increment of illumination discriminable 50% of the time with background illumination I.
- The human eye can perceive a very few range of colors that comprises only a small portion of the electromagnetic spectrum.
- A light that has no colour is called monochromatic and it's only attribute is it's intensity.
- An image whose intensity ranges from back to white is termed as a grayscale image.
- Radiance is the total amount of energy that flows from the light source, and it is usually measured in watts (W).Luminance, measured in lumens (lm), gives a measure of the amount of energy an observer perceives from a light source

2.6 Answers to Check Your Progress

1. Cornea
2. Retina
3. Cones, rods
4. Ciliary body

5. The distance between the centre of the lens of the eye and the retina is called the focal length

6. Brightness discrimination is given by the Weber ratio $\Delta I_c/I$ where $\Delta I_c$is the increment of illumination discriminable 50% of the time with background illumination I.

7.The phenomenon that describes the ability of the human eye to discriminate the different distinct intensity levels is called brightness adaptation.

2.7 Possible Questions

1. What is visual perception?
2. List the three elements of visual perception.
3. What is the average diameter of an eye?
4. Explain the structure of an eye with the help of an example.
5. Explain with the help of an example how an image is formed in an eye.
6. Define brightness adaptation and discrimination.
7. What is an electromagnetic spectrum?
8.Which electromagnetic wave has the strongest energy?
9. What is a grayscale image?
10. Define radiance and luminance.
11. Mention some applications of Gamma rays and radio waves.

2.8 References and Suggested Readings

1. Gonzalez, R. C. (2009). *Digital image processing*. Pearson education India.

*****

UNIT: 3
IMAGE SENSING AND ACQUISITION

Unit Structure:

3.1 Introduction

Image sensing and acquisition is the process of detecting and accessing the amount of light emitted or reflected from the source object so that the physical properties of it can be represented correctly in another system. Image acquisition is performed by using different image sensors which in turn generate digital images.

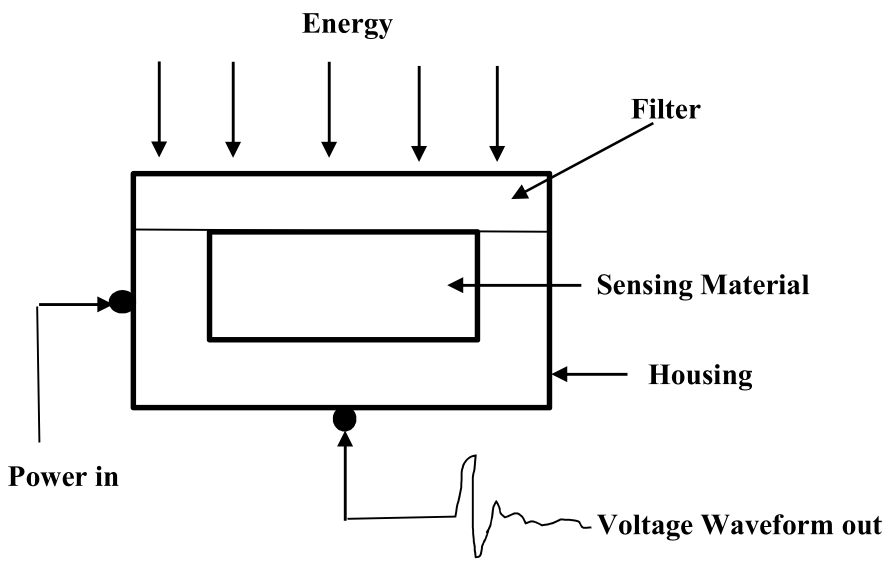
3.2 Objectives

After going through this unit, you will learn:

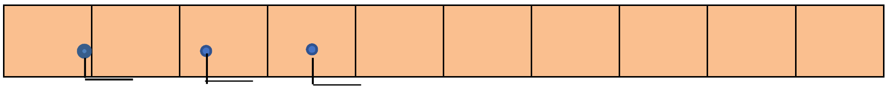
- *image* sensing and acquisition,
- *image* Acquisition using a Single Sensor,
- *image* Acquisition using Sensor Strips,
- *image* Acquisition using Sensor Arrays,
- *a* Simple Image Formation Model.

3.3 Image Acquisition

An image is usually generated by the combination of illumination from the source object and the reflection or absorption of energy of that source by the elements of the scene being imaged. The illumination may originate from the sun or any other electro-magnetic source such as radar, infrared or X-ray stream. The amount of illumination energy reflected from or transmitted through the object depends upon the nature of the source object. For example light can reflect from a planar surface or through a patient's body while generating X-rays. This energy is then transformed into visible light by focussing the reflected energy on to a photo converter. The three principal sensor arrangements used in the conversion of the illumination energy into digital images are: single imaging sensor, line sensor and array sensor illustrated in figure 1 a), b) and c):

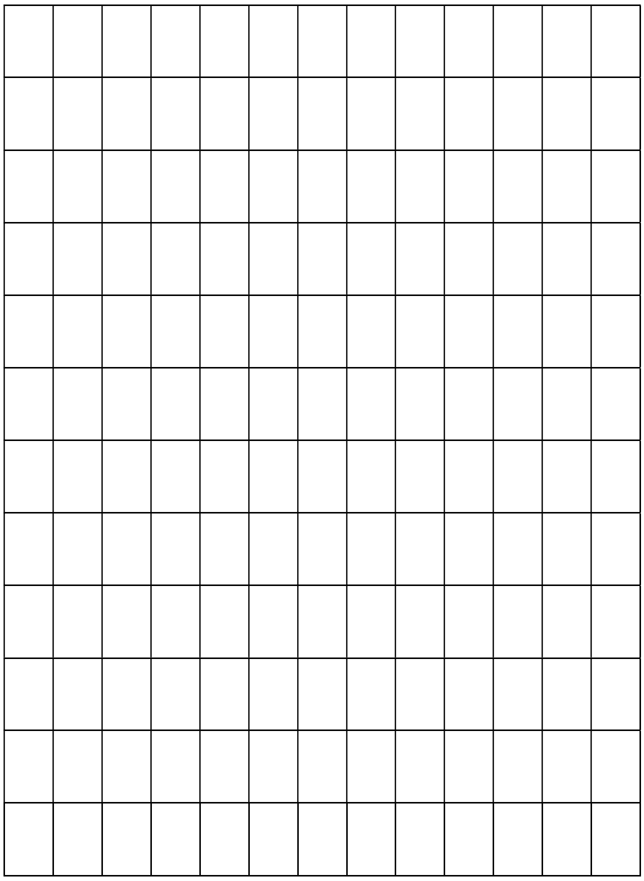


a) Single Imaging Sensor



b) Line Sensor

....



c) Array Sensor

Fig 1: Principal Sensor Arrangements

In figure 1 a) The illuminated energy given as input to the sensor which in combination with the electrical power and the sensor material gets transformed into a voltage. This happens because the sensor material responds to the specific type of energy being detected. The response of the sensor is the voltage waveform which forms the output. Along with the voltage waveform a digital quantity is also generated which is obtained by digitizing the response of the sensor.

Stop to Consider
The three main steps in Image Acquisition process can be summarized as [3]: 1. Optical system which focuses the energy 2. Energy reflected from the object of interest 3. A sensor which measure the amount of energy.

3.3.1 Image Acquisition using a Single Sensor

The most common of this type of sensor is the photodiode which is formed from silicon material and the output, which is a voltage waveform, is proportional to light. A filter placed in front of the sensor helps in selecting a specific colour. For example a green filter when placed in front of a light sensor favours the green band of the colour spectrum. So the output of the sensor will be strong for the green colour rather than the other colours. For 2D image generation, the displacement of the sensor on the object to be imaged should be in both x and y directions.

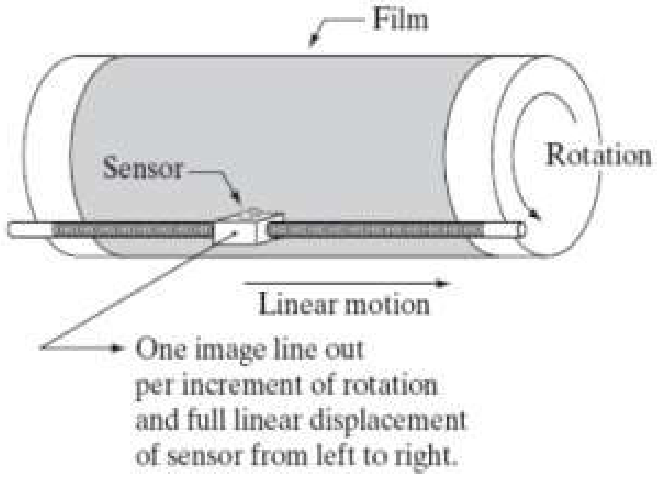


Fig 2: Combining a single sensor with motion to generate a 2D image

In figure 2, we can see that a film negative is mounted on a cylinder drum. On rotating, the drum causes the vertical movement in one dimension. The horizontal movement is caused by the sensor placed on the lead screw. This method is an inexpensive way to generate high resolution 2D images. A flatbed can also be used with a sensor moving in two linear directions. These types of mechanical digitizers are also referred to as microdensitometers. Similarly, a laser source placed coincident with the sensor can also be used in imaging. The outgoing beam is controlled by moving mirrors and also it directs the reflected laser signal onto the sensor.

3.3.2 Image Acquisition using Sensor Strip

In figure 3 we can see a sensor strip that provides imaging elements in one direction. The motion perpendicular to the strip provides imaging in the other direction. This type of imaging is used in flatbed scanners. Sensor strip is mostly used in airborne imaging application.

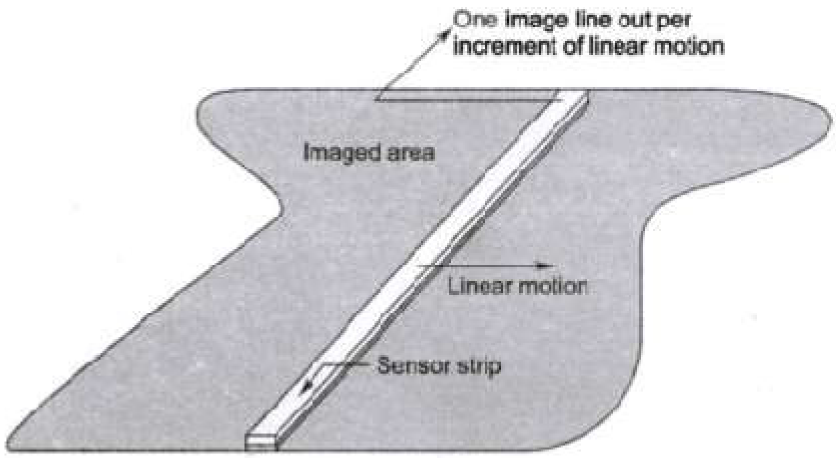


Fig 3: Image Acquisition Using a linear Sensor Strip

A circular sensor strip on the other hand has sensor strips mounted in a ring configuration. These type of sensor strips are used mostly in medical and industrial imaging to obtain cross sectional images of 3D objects. In figure 3 a rotating X-ray source provides illumination and the sensors opposite the source collect the X-ray energy that passes through the object. This form the basis of Computerized Axial Tomography (CAT) imaging technique. The output of these sensors must go through the process of extensive processing by the use of reconstruction algorithms to get meaningful cross-sectional images. Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) are the imaging modalities which works on the principle of CAT.

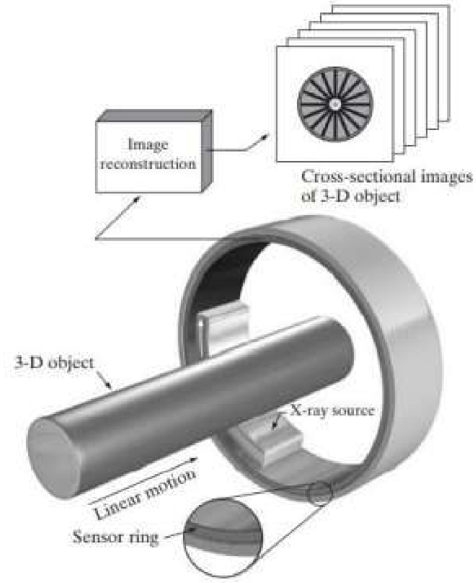


Fig 4: Image acquisition using a circular sensor strip

3.3.3 Image Acquisition using Sensor Arrays

In figure 1 c) we have seen how the individual sensors are arranged in the form of a 2D array. This array format of this sensor consists of a number of electromagnetic and some ultrasonic sensing devices. This type of arrangement is usually found in digital cameras where a CCD array manufactured using a broad range of sensing properties is packaged in rugged arrays of 4000x4000 elements or more. The response of each sensor is proportional to the integral of the light energy projected onto the surface, of the sensor. This property is used in astronomical and other applications requiring low noise image. The input light signal is integrated by the sensor over minutes or hours to achieve noise reduction. In figure 5 c) we can see a 2D sensor. The key advantage of using 2D array is that a complete image can be obtained by focussing the energy pattern onto the surface of the array. Figure 5 shows the principal manner in which array sensors are used. As we can see in figure 5 a) and b) how the illumination from a source is reflected by the object of a scene. The energy reflected by the object is collected as an input by the imaging system and the output is focussed on an image plane which is depicted in figure 5 c) and d) respectively. But when the illumination is light this process is different. Then the front end of the imaging system is a lens which projects the viewed scene onto the lens focal plane, as shown in Figure 5(d). The sensor array, which is coincident with the focal plane, produces outputs proportional to the integral of the light received at each sensor. Digital and analog circuitry then shifts swiftly these outputs and converts them to a video signal, which is then digitized by another section of the imaging system. The output of the digital image is shown in figure 5 e)

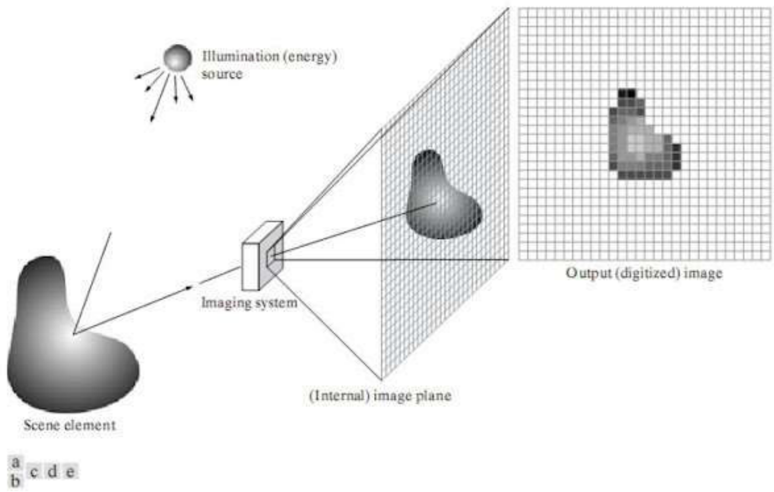


Fig 5: An example of the digital image acquisition process (a) Energy ("illumination")

source (b) An element of a scene (c) Imaging system (d) Projection of the scene onto the
image plane (e) Digitized image

Check Your Progress

1. The illumination may originate from electro-magnetic source such as __________ and ___________ or ___________

2. A filter placed in front of the ____helps in selecting a specific colour.

3. A ______ can also be used with a sensor moving in two linear directions.

4. Mention an application of sensor strip.

5. What does an array sensor consist of ?

3.3.4 A Simple Image Formation Model

As we know that an image is represented by a 2D function of the form f(x, y). The source of the image determines the value or amplitude f, a positive scalar quantity, at the spatial co-ordinates (x, y).The intensity values of an image generated from a physical process is proportional to the energy radiated by a physical source. As a result of this, the function f(x, y) must be non-zero as well as also finite. This can be represented mathematically as below:

$$0<f(x, y) <\infty \text{ -----------------(1)}$$

The two components of f(x, y) may be characterized by two components: (a) the amount of source illumination incident on the scene being viewed which is called as illuminance and is denoted by i(x, y) and (b) the amount of illumination reflected by the objects in the scene which is called as reflectance and is denoted by r(x, y). These two functions are combined as a product to form f(x,y):

$$f(x, y) = i(x, y)r(x, y) \text{ --------------(2)}$$

$$\text{where } 0 < i(x, y) < \infty \text{---------------(3)}$$

$$\text{and } 0 < r(x, y) < 1 \text{ -----------------(4)}$$

The reflectance is bounded by 0 and 1 as we can see in equation 4. The nature of i(x, y) and r(x, y) is determined by the illumination source and characteristics of the imaged objects respectively.

3.4 Summing Up

1. The three principal sensor arrangements used in the conversion of the illumination energy into digital images are: single imaging sensor, line sensor and array sensor.
2. An example of image acquisition using a single sensor is a photodiode.
3. For 2D image generation using a single sensor, the motion should be both in x and y directions.
4. Image acquisition using a line sensor provides imaging only in one direction.
5. Sensor strip is mostly used in airborne imaging application.
6. Image acquisition using an array sensor requires individual sensors to be arranged in the form of a 2-D array.
7. This arrangement is usually found in digital cameras which is in the form of a CCD array.
8. An image is represented by a 2D function of the form f(x, y).
9. The function f(x, y) must be non-zero as well as also finite.
10. The two components of f(x, y) may be characterized by two components: illuminance and reflectance.

3.5 Answers to Check Your Progress

1. Radar, infrared, X-rays
2. Sensor

3. Flatbed

4. Sensor strip is mostly used in airborne imaging application.

5. Sensor consists of a number of electromagnetic and some ultrasonic sensing devices.

3.6 Possible Questions

1. What is image acquisition?
2. Explain the concept of image acquisition.
3. What are the three principal sensor arrangements used in the conversion of the illumination energy into digital images?
4. Explain the process of image acquisition using a single sensor?
5. Explain the structure of a circular sensor strip.
6. What is the basis of CAT imaging technique?
7. Explain the process of image acquisition using sensor arrays.
8. Define: Illuminance, Reflectance

3.7 References and Suggested Readings

1. Gonzalez, R. C. (2009). *Digital image processing*. Pearson education India.
2. Mishra, V., Kumar, S., & Shukla, N. (2017). Image Acquisition and Techniques to Perform Image Acquisition. SAMRIDDHI: A J. *Phys. Sci. Eng. Technol*, *9*.

*****

UNIT: 4
IMAGE SAMPLING AND QUANTIZATION

Unit Structure:

4.1 Introduction

This unit introduces the concept of sampling and quantization. The output of Image sensors is always in analog form. This analog data cannot be applied for digital processing. For digital processing analog data needs to be converted into digital form. As we know that analog data contains continuous data and digital data means discrete data. For creating a digital image, the continuous data must be converted into discrete form. This is done with the help of two step procedure: sampling and quantization.

4.2 Objectives

In this unit you will learn about the-

- *basic* concepts of sampling and quantization
- *types* of sampling

- *spatial* and Intensity resolution
- *image* interpolation

4.3 Image Sampling and Quantization

As we know that an image is continuous not only on it's co-ordinates (x- axis) but also on its amplitude (y-axis), so it needs to be converted into digital form for processing. The digitization of the co-ordinate values is termed as sampling and the digitizing of the amplitude values is termed as quantization. Sampling should be done prior to the quantization process.

4.3.1 Basic Concepts in Sampling and Quantization

The basic concept of image sampling and quantization is illustrated in Fig 1:

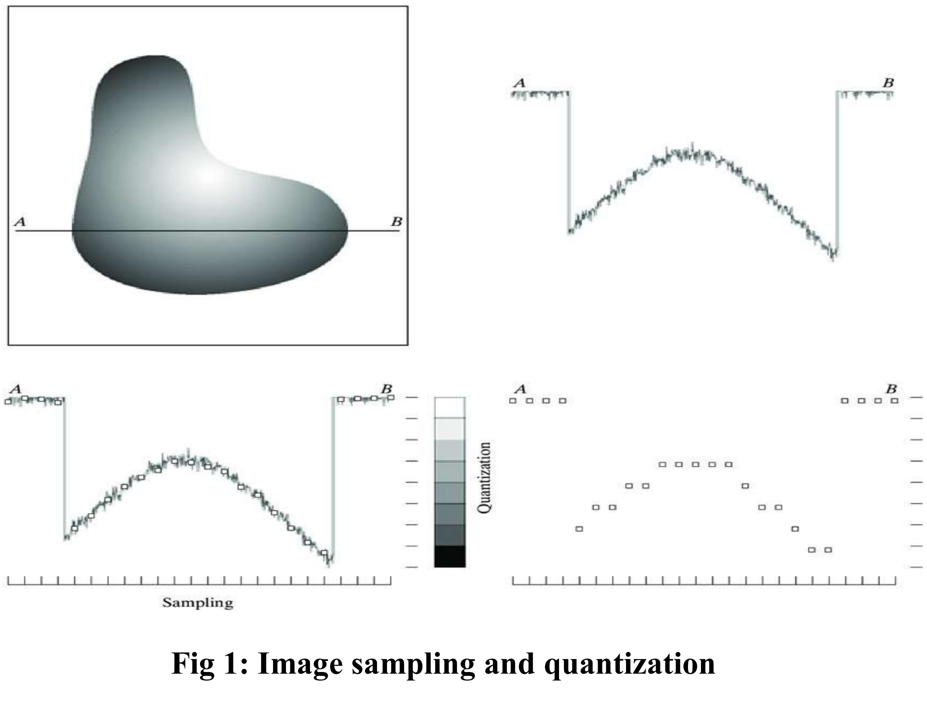


Fig 1: Image sampling and quantization

Let us convert the continuous image f in digital form. The image may be continuous in both the directions x and y and also in amplitude.

For converting the image function f into digital form, it needs to be sampled in both directions x and y and in amplitude also. Fig b) displays the plot of amplitude values of the continuous image function f along the line segment AB. We can see the random variations due to the presence of noise in the image. For sampling this function we need to take equally spaced samples along the line AB. The spatial location of each sample is indicated by a vertical tick mark in the bottom part of the figure. The samples are displayed as small white squares in between superimposed in the function. The sampled function is obtained by the set of these discrete locations. As the intensity values of the sample still span a continuous range of intensity, these needs to convert into discrete quantities in order to form a digital function. The intensity scale which is divided into eight discrete intervals ranging from black to white is shown in Fig c). Each of the eight discrete intervals are assigned a specific value which is indicated by the vertical tick marks. The process of quantization is done by assigning the continuous intensity levels into one of the eight values to each sample. The assignment of a sample to a particular value depend upon it's vertical proximity to the vertical tick mark. The final output after sampling and quantization is shown in Fig d)

The above sampling procedure assumes that we have a continuous image that is not only continuous in both the directions x and y but also in amplitude. In practical the method of sampling is determined by the sensor arrangement to generate the image.

Stop to Consider

Sampling is measured by pixels per inch (PPI) or dots per inch (DPI) and quantization is measured by bits per channel

4.3.2 Representing Digital Images

Let us convert a continuous image function f(s,t) consisting of two continuous variables s and t into a digital image by sampling and quantization. Now we assume that we have sampled the continuous image into a 2D array f(x,y) and it has M rows and N columns as shown in Fig 2:

$$
f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) \ldots\ldots\ldots & f(0,N-1) \\ f(1,0) & f(1,1) \ldots\ldots\ldots & f(1,N-1) \\ \ldots\ldots\ldots & \ldots\ldots\ldots & \ldots\ldots\ldots \\ \ldots\ldots\ldots & \ldots\ldots\ldots & \ldots\ldots\ldots \\ \ldots\ldots\ldots & \ldots\ldots\ldots & \ldots\ldots\ldots \\ f(M-1,0) & f(M-1,1) \ldots\ldots & f(M-1, N-1) \end{bmatrix}
$$

Fig 3: Representation of a continuous image in a 2D array

The values of the coordinates (x,y) now become discrete quantities thus the value of the coordinates at orgin become (x,y) =(0,0). The next co-ordinates value along the first signify the image along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled.

A digital image can be represented by both the sides of the equation given above as they are equivalent. The right side of the equation represents a matrix of real numbers. Each of the element in the matrix is referred to as an image element, picture element, pixel or pel.

4.3.3 Uniform and Non-uniform Sampling

In Uniform sampling the spectrum of uniformly spaced samples is also a set of uniformly space spikes. On the other hand in Non-uniform sampling samples at non-uniform locations have a different spectrum i.e, a single spike and noise.

4.3.4 Spatial and Intensity Resolution

Image Resolution can be referred to as the total number of pixels in an image along with how much detail the image portrays. The smallest discernible detail in an image is termed as spatial resolution. Line pairs per unit distance and dots per unit distance are the most common measures that are used for defining spatial resolution. For example if we construct a chart with black and white vertical lines, each having W units, the width of a line pair is thus 2W and there are ½ W line pairs per unit distance. Image resolution can also be defined as the largest number of discernible line pairs per unit distance. Dots per unit distance is one of the measures of image resolution which is most commonly used in printing and publishing industry. In order to define spatial resolution of an image, spatial units with respect to the image is necessary. In Figure 4 we can see an image of 1024 pixels which means 1024 in both width and height. Below it we can see the same image with different resolution size. With higher image resolution, higher is the spatial resolution with higher amount of pixels which results in better image quality.

Fig 4: An image with different resolution size

Image source:
https://miro.medium.com/v2/resize:fit:1400/format:webp/1*UmrcQbhNz9B0DPIz1FIDvQ.png

On the other hand the smallest discernible change in intensity level is termed as intensity resolution. The number of intensity levels usually is an integer power of 2. The most common intensity level is 8 bits followed by 16 bits and 32 bits which are used in some rare applications.Intensity level resolution is usually given in terms of the number of bits used to store each intensity level. In the table we can see the different number of intensity levels assigned to the different number of bits.

Number of Bits	Number of Intensity Levels	Examples
1	2	0,1
2	4	00,01,10,11
4	16	0000,0101,1111
8	256	00110011,01010101
16	65,536	1010101010101010

An image having 8 bits of intensity resolution means the ability to quantize intensity in fixed increments of 1/256 units of intensity amplitude.

Check Your Progress

1. The process of ____________is done by assigning the continuous intensity levels into one of the eight values to each sample.

2. Random variations in an image is due to the presence of ___________

3.__________ can also be defined as the largest number of discernible line pairs per unit distance.

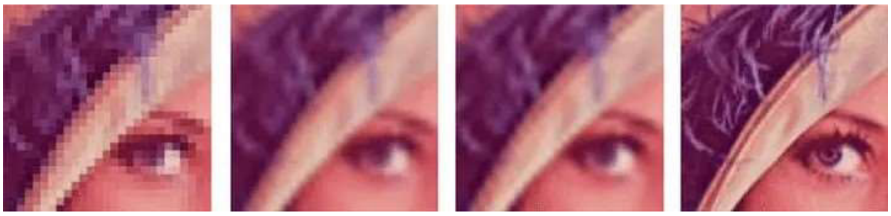
4. ____________distance is one of the measure of image resolution.

5. The number of intensity levels usually is an integer power of _____

4.3.5 Image Interpolation

The process of transferring images from one resolution to another without losing the attributes of an image is called image interpolation. Image interpolation is an important function that is used when we perform zooming, shrinking, rotating and geometric corrections in an image. It is a process of using known data to estimate values in unknown location. Image interpolation also helps in producing high resolution images from it's low resolution version. The application of image interpolation includes image enlargement, image reduction, subpixel image registration, image decomposition and to correct spatial distortions and many more. The two main categories of image interpolation algorithms are adaptive and non-adaptive interpolation algorithm. In non-adaptive method the image features are not considered rather some procedure is applied on all pixels whereas in adaptive interpolation algorithm the features and quality of the image are being considered before the algorithm is being applied.

Let's understand image interpolation with the help of a simple example. Suppose we need to enlarge an image of size 500x500 to 750x750 pixels. To visualize the zooming of an image we have to create an imaginary grid with the same pixel spacing as the original and then shrink it in such a way that it fits exactly over the original image. The pixel spacing in the original image will obviously be higher than the shrunken image grid of 750x750. To assign intensity values to the new 750x750 grid, the closest pixel in the original image is found out and that value is given to the shrunken image. We assign all the other points of the overlay grid in the same manner and once we are done, the shrunken image is expanded to the original specified image to obtain the zoomed image. This method in which the intensities of the pixels are assigned on the basis of the proximity of it's nearest neighbourhood of the original image is called nearest neighbourhood interpolation. Another way of assigning intensities to a given location (x,y) is to consider the four nearest neighbours of the original image. This method is called bilinear interpolation. The next level of complexity is bicubic interpolation in which sixteen nearest neighbourhoods are concerned. This will be discussed in details in the later chapters. Figure 5 depicts the visual comparison of the different interpolation methods.



a) b) c) d)

Fig 5: Visual comparison of different interpolation methods. (a) Nearest-neighbour. (b) Bilinear. (c) Bicubic. (d) Original HR image

Image source: https://www.researchgate.net/publication/287419933/figure/fig2/AS:398896468643841@1472115737243/sual-comparison-of-different-interpolation-methods-a-Nearest-neighbor-b-Bilinear.png

4.4Summing Up

1. For creating a digital image, the continuous data must be converted into discrete form. This is done with the help of two step procedure: sampling and quantization.
2. The digitization of the co-ordinate values is termed as sampling and the digitizing of the amplitude values is termed as quantization.
3. Sampling is always performed before quantization.
4. Uniform sampling have uniformly spaced sampling levels whereas non-uniform sampling have unequal spaced sampling levels.
5. Image Resolution can be referred to as the total number of pixels in an image along with how much detail the image portrays.
6. The smallest discernible detail in an image is termed as spatial resolution.
7. For better image quality, image and spatial resolution should be high as it results in greater number of pixels.
8. The process of transferring images from one resolution to another without losing the attributes of an image is called image interpolation.

4.5 Answers to Check Your Progress

1.Quantization
2. Noise
3. Image resolution
4. Dots per unit
5.Two

4.6 Possible Questions

1. Define sampling and quantization with the help of an example.
2. What is spatial and intensity resolution?
3. What is image interpolation? What are the different types of image interpolation? Explain

4.7 References and Suggested Readings

1. Gonzalez, R. C. (2009). *Digital image processing*. Pearson education India.
2. https://www.slideserve.com/elina/sampling-and-reconstruction

*****

UNIT: 5

BASIC CONCEPTS OF IMAGE TRANSFORM

Unit Structure:

5.1 Introduction

The French mathematician Jean Baptiste Joseph Fourier was born in 1768 in the town of Auxerre, about midway between Paris and Dijon. His most well-known contribution was described in an autobiography written in 1807 and published in his book La Théorie Analitique de la Chaleur (The Analytic Theory of Heat) in 1822. Freeman translated this work into English 55 years later. Any periodic function can be written as the sum of sines and/or cosines of various frequencies, each multiplied by a different coefficient. This sum is now known as a Fourier series. This is essentially Fourier's contribution in this subject. If the function is periodic and meets certain simple mathematical requirements, it can be represented by such a sum regardless of how complex it is. This is now taken for granted but, at the time it first appeared, the concept that complicated functions could be represented as a sum of simple sines and cosines was not at all intuitive, so it is not surprising that Fourier's ideas were met initially with skepticism.

The integral of sines and/or cosines multiplied by a weighting function can be used to define even non-periodic functions with a finite area under the curve. In many theoretical and applied fields, the Fourier transform—the formulation used here—is even more useful than the Fourier series. The key feature of both representations is that a function, whether it is described as a transform or a Fourier series, may be fully recovered (reconstructed) through an inverse process without any information loss. This is one of the most important characteristics of these representations because it allows us to work in the "Fourier domain" and then return to the original domain of the function without losing any information. Ultimately, it was the utility of the Fourier series and transform in solving practical problems that made them widely studied and used as fundamental tools.

The initial application of Fourier's ideas was in the field of heat diffusion, where they allowed the formulation of differential equations representing heat flow in such a way that solutions could be obtained for the first time. During the past century, and especially in the past 50 years, entire industries and academic disciplines have flourished as a result of Fourier's ideas. The advent of digital computers and the "discovery" of a fast Fourier transform (FFT) algorithm in the early 1960s (more about this later) revolutionized the field of signal processing.

5.2 Objectives

This unit is an attempt to explain and implement ideas of some basic image transformation techniques. After going through this unit, you will be able to-

- *explain complex number,*
- *explain Fourier series*,
- *apply* impulses and their shifting properties,
- *explain* Fourier transform for one continuous variable,
- *understand* Convolution in frequency and time domain.

5.3 Complex Numbers

A complex number, C, is defined as

$$C = R + jI$$

where R and I are real numbers, and j is an imaginary number equal to the square of -1; that is$j = \sqrt{-1}$ Here, R denotes the real part of the complex number and I its imaginary part. Real numbers are a subset of complexe numbers in which I = 0 The conjugate of a complex number C, denoted C^* is defined as

$$C^* = R - jI$$

Complex numbers can be viewed geometrically as points in a plane (called the complex plane) whose abscissa is the real axis (values of R) and whose ordinate is the imaginary axis (values of I). That is, the complex number $R + jI$ is point (R, I) in the rectangular coordinate system of the complex plane. Sometimes, it is useful to represent complex numbers in polar coordinates,

$$C = |C|\,(\cos\theta + j\sin\theta)$$

Where $|C| = \sqrt{R^2 + I^2}$ is the length of the vector extending from the origin of the complex plane to point (R, I), and θ is the angle between the vector and the real axis. Drawing a simple diagram of the real and complex axes with the vector in the first quadrant will reveal that $\tan\theta = ({}^{I}/_{R})$ or $\theta = \arctan(\frac{I}{R})$. The arctan function returns angles in the range ${}^{-\pi}/_{2}, {}^{\pi}/_{2}$. However, because I and R can be positive and negative independently, we need to be able to obtain angles in the full range $[-\pi, \pi]$. This is accomplished simply by keeping track of the sign of I and R when computing θ. Many programming languages do this automatically via so called four-quadrant arctangent functions.

5.4 Fourier Series

A function $f(t)$of a continuous variable, t, that is periodic with a period, T, can be expressed as the sum of sines and cosines multiplied by appropriate coefficients. This sum, known as a Fourier series, has the form

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi}{T}t}$$

where

$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)e^{j\frac{2\pi}{T}t}\,dt$ for $n = 0, \pm1, \pm2, \dots\dots\dots$

are the coefficients.

5.5 Impulses and their Shifting property

Impulses and their dynamic characteristics are crucial to understand how systems respond to inputs in image processing. The idea of an impulse and its shifting characteristics are fundamental to the study of linear systems and the Fourier transform. A unit impulse of a continuous variable t located at t=0, and denoted, (t) , is defined as

$$\delta(t) = \begin{cases} \propto & if\ t = 0 \\ 0 & if\ t \neq 0 \end{cases}$$

and is constrained to satisfy the identity

$$\int_{-\propto}^{\propto} \delta(t)dt = 1$$

An impulse can be physically represented as a spike with unit size, zero duration, and infinite amplitude if we take t to be time. An impulse has the so-called shifting property with respect to integration.

$$\int_{-\propto}^{\propto} f(t)\delta(t)dt = f(0)$$

Provided that $f(t)$ is continuous at t=0, a condition typically shifted in practice. Shifting simply yields the value of the function $f(t)$ at the location of the impulse (i.e., at t=0 in the previous equation). A more general statement of the shifting property involves an impulse located at an arbitrary point, t_0, denoted as, $\delta(t - t_0)$. In this case,

$$\int_{-\propto}^{\propto} f(t)\delta(t - t_0)dt = f(t_0) \qquad (5.1)$$

which simply gives the value of the function at the location of the impulse. For example, if $f(t) = \cos(t)$, using the impulse $\delta(t - \pi)$ in Eq. (5.1) yields the result $f(\pi) = \cos(\pi) = -1$. The power of the sifting concept will become evident shortly.

5.6 Fourier Transform of Functions of one continuous variable

Functions that are not periodic but whose area under the curve is finite can be expressed, as the integral of sines and/or cosines multiplied by a weighting function. This formulation is the Fourier transform. Both representations share the important characteristic that a function, expressed in either a Fourier series or transform, can be reconstructed completely via an inverse process, with no less of information. Fourier transform is widely used in the field of image processing. An image is a spatially varying function. One way to analyse spatial variations is to decompose an image into a set of orthogonal functions, one such set being the Fourier functions, A Fourier transform is used to transform an intensity image into the domain of spatial frequency. The important feature of the Fourier transform is that irrespective of the type of signal, it is possible to describe the signal as the sum of a collection of sine and/or cosine waves of different frequencies and amplitudes multiplied by the weighting function. Fourier transformation gives you the frequency components present in the image. And for sampling we must meet the condition that your sampling frequency mast be greater than twice the maximum frequency present in the continuous. The Fourier transform of the continuous function $f(x)$ of a continuous variable$x$, is denoted as $F(u)$, where u represents the spatial frequency.

$$F(u) = \int_{-\propto}^{\propto} f(x)\cos(2\pi ux) - j\, sin(2\pi ux) dx$$

This can be expressed in a concise manner in exponential form as

$$F(u) = \int_{-\propto}^{\propto} f(x) e^{-j2\pi ux} dx \qquad (5.2)$$

Here,

ux - frequency variable

$f(x)$ - must be continuous and integrable

$e^{-j2\pi}$ - complex quantity

$j = \sqrt{-1}$

The Fourier transform treats all image data as complex numbers. Similarly, given $F(u)$, it is possible to get back $f(x)$. This is done using the inverse Fourier transform. The inverse Fourier transform is expressed mathematically as

$$f(x) = \int_{-\propto}^{\propto} F(u)[\cos(2\pi ux) + jsin(2\pi ux)]du$$

In the exponential form it is expressed as

$$f(x) = \int_{-\propto}^{\propto} F(u)e^{+j2\pi ux}du \qquad (5.3)$$

Equations (5.2) and (5.3) known as Fourier transform pairs.

The exponential term has the opposite sign. These two equations (5.2) and (5.3) form the Fourier transform pair. This means any spatial information can be converted to the frequency domain and vice versa without loss of information. It can also be observed that the Fourier transform results in a complex quantity even though the original data is not a complex number. So when the Fourier transform is applied to an image the image is treated in the real/imaginary format with the imaginary component as zero, or in the magnitude/phase format where the phase angle is assumed to be zero.

5.7 Convolution

Convolution in the frequency domain is the Fourier transform of a product of time-domain functions or signals. The convolution theorem states that the Fourier transform of a convolution of two functions is equal to the product of their Fourier transforms. This means that convolution in one domain (time domain) is equal to point-wise multiplication in the other domain (frequency domain). The convolution of two continuous functions or signals $f(t)$ and $g(t)$, of one continuous variable t, denoted by the operator *, is defined as

$$f(t) * g(t) = \int_{-\alpha}^{\alpha} f(\tau) - g(t-\tau) d\tau$$

Where the minus sign accounts for the flipping, t is the displacement needed to slide one function past the other, and τ is a dummy variable that is integrated out. We assume that the functions extend from –α to α. The Fourier transform of convolution operation will be as follows.

$$F(f(t) * h(t) = \int_{-\propto}^{\propto} \left[\int_{-\propto}^{\propto} f(\tau) g(t-\tau) d\tau \right] e^{-j2\pi\mu t} dt$$

$$= \int_{-\propto}^{\propto} f(\tau) \left[\int_{-\propto}^{\propto} g(t-\tau) e^{-j2\pi\mu(t-\tau)} dt \right] e^{-j2\pi\mu\tau} d\tau$$

The term inside the brackets is the Fourier transform of $g(t-\tau)$.

$$= \int_{-\propto}^{\propto} f(\tau) H(w) e^{-j2\pi\mu} \, d\tau$$

$$= H(w) \int_{-\propto}^{\propto} f(\tau) e^{-j2\pi\mu\tau} d\tau$$

$$= H(w).F(w)$$

Hence,

$$f(t) * g(t) = F(w).H(w)$$

If we take the convolution of two signals or functions in time domain, then it is equivalent to the multiplication of the two signal in frequency domain. The reverse is also true.

$$F(w) * H(w) = f(t).g(t)$$

Check Your Progress

1. State True or False

(i) The initial application of Fourier's ideas was in the field of heat diffusion.

(ii) Complex numbers cannot be viewed geometrically.

(iii) An image is a spatially varying function.

(iv) Fourier transformation gives you the frequency components present in the image

(v) The Fourier transform treats all image data as general numbers.

5.8 Summing up

- A transform is basically a representation of signal. A transform changes the representation of a signal by projecting it onto a set of basis functions. The transform does not change the information content present in the signal.
- The transformation matrix A is unitary if it obeys the following condition

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

- The transformation matrix A is orthogonal if it obeys the following relation

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

- The king of all transforms is Fourier series and Fourier transform.

5.9 Answer to Check Your Progress

1. (i) True (ii) False (iii) True (iv)True (v) False

5.10 Possible Questions

1. What is complex number ? Give an example?

2. What is Fourier series. Explain in details

3. What is convolution explain in brief.

4. Explain Fourier transform for one continuous variable? Also write the equation for inverse Fourier transform for one continuous variable.

5.11 References and Suggested Readings

1. Digital Image Processing, 3ed, Rafael C. Gonzalez, Richard E. Woods, Pearson.
2. Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, McGrawHill
3. Digital Image Processing, 2^{nd} ed,S Sridhar, Oxford
4. NPTEL, IITKGP
5. Digital Image Processing, Dr. Sanjay Sarma, SK Kataria & Sons

*****

UNIT: 6
IMAGE TRANSFORM I

Unit Structure:

6.1 Introduction

Image transforms are basically used in image processing and image analysis. Transform is basically a mathematical tool, which allows us to move from one representation to another representation (time domain to the frequency domain or vice versa). The reason to change from one representation to another representation is to perform the task at hand in an easier manner. Image transforms are useful for fast computation of convolution and correlation. Transforms change the representation of a signal by projecting it onto a set of basis functions. The transforms do not change the information content present in the signal or image. Most of the image transforms, like the Fourier transform, discrete cosine transform, wavelet transform, etc., give information about the frequency contents in an image. It is to be noted that all the transforms will not give frequency domain information. Transforms play a significant role in various image-processing applications such as image analysis and image enhancement. To extract more Information Transforms allow us to extract more relevant information. To illustrate this, consider the following example. There are two reasons for transforming an image from one representation to another. First, the transformation may isolate critical components of the image pattern so that they are directly accessible for analysis. Second, the transformation may place the image data in a more compact form so that it can be stored and transmitted efficiently.

6.2 Objectives

This unit is an attempt to explain and implement ideas of some basic image transformation techniques. After going through this unit, you will be able to

- *explain Fourier transform and its inverse transform,*
- *explain properties of Fourier transform*,
- *apply* Fourier transform in 1D function,
- *understand* convolution in frequency domain.

6.3 Introduction to Fourier Transforms

Functions that are not periodic but whose area under the curve is finite can be expressed as the integral of sines and/or cosines multiplied by a weighting function. This is the Fourier transform. Both representations share the important characteristic that a function, expressed in either a Fourier series or transform, can be reconstructed completely via an inverse process, with no loss of information. Fourier transform is widely used in the field of image processing. An image is a spatially varying function. One way to analyse spatial variations is to decompose an image into a set of orthogonal functions, one such set being the Fourier functions, A Fourier transform is used to transform an intensity image into the domain of spatial frequency. The important feature of the Fourier transform is that irrespective of the type of signal, it is possible to describe the signal is the sum of a collection of sine and/or cosine waves of different frequencies and amplitudes multiplied by the weighting function.

6.4 Fourier Transform in Continuous Domain

(a) One variable

Functions that are not periodic but whose area under the curve is finite can be expressed, as the integral of sines and/or cosines multiplied by a weighting function. This formulation is the Fourier transform. Both representations share the important characteristic that a function, expressed in either a Fourier series or transform, can be reconstructed completely via an inverse process, with no less of information. Fourier transform is widely used in the field of image processing. An image is a spatially varying function. One way to analyse spatial variations is to decompose an image into a set of orthogonal functions, one such set being the Fourier functions, A Fourier transform is used to transform an intensity image into the domain of spatial frequency. The important feature of the Fourier transform is that irrespective of the type of signal, it is possible to describe the signal as the sum of a collection of sine and/or cosine waves of different frequencies and amplitudes multiplied by the weighting function. Fourier transformation gives you the frequency components present in the image. And for sampling we must meet the condition that your sampling frequency mast be grater than twice the maximum frequency present in the continuous. The

Fourier transform of the continuous function $f(x)$ of a continuous variable x, is denoted as $F(u)$, where u represents the spatial frequency.

$$F(u) = \int_{-\propto}^{\propto} f(x)\cos(2\pi ux) - j\, sin(2\pi ux)dx$$

This can be expressed in a concise manner in exponential form as

$$F(u) = \int_{-\propto}^{\propto} f(x)e^{-j2\pi ux}dx \qquad (6.1)$$

Here,

ux - frequency variable

$f(x)$ - must be continuous and integrable

$e^{-j2\pi}$ - complex quantity

$j = \sqrt{-1}$

The Fourier transform treats all image data as complex numbers. Similarly, given $F(u)$, it is possible to get back $f(x)$. This is done using the inverse Fourier transform. The inverse Fourier transform is expressed mathematically as

$$f(x) = \int_{-\propto}^{\propto} F(u)[\cos(2\pi ux) + jsin(2\pi ux)]du$$

In the exponential form it is expressed as

$$f(x) = \int_{-\propto}^{\propto} F(u)e^{+j2\pi u}\ du \qquad (6.2)$$

Equations (5.2) and (5.3) known as Fourier transform pairs.

The exponential term has the opposite sign. These two equations (5.2) and (5.3) form the Fourier transform pair. This means any spatial information can be converted to the frequency domain and vice versa without loss of information. It can also be observed that the Fourier transform results in a complex quantity even though the original data is not a complex number. So when the Fourier transform is applied to an image the image is treated in the real/imaginary format with the imaginary component as zero, or in the magnitude/phase format where the phase angle is assumed to be zero.

(b) Two variable

The Fourier transform can be extended to 2D functions. The Fourier transform for the 2D image $f(x,y)$ is given by

$$F(u,v) = \int_{-\propto}^{\propto}\int_{-\alpha}^{\alpha} f(x,y)e^{-j2\pi(ux+\quad)}\,dxdy$$

and the inverse transform is given by

$$f(x,y) = \int_{-\propto}^{\propto}\int_{-\alpha}^{\alpha} F(u,v)e^{j2\pi(ux+v\quad)}\,dudv$$

6.5 Discrete Fourier transform (DFT)

(a) One variable:

The Fourier transform of on sampled, band limited function extending from –α to α is a continuous, periodic function that also extends from –α to α. Since the images are digitized, it is necessary to have a discrete formulation of the Fourier transform. This is achieved by the discrete Fourier transform (DFT), which takes regularly space data values and returns the value of the Fourier transform by replacing the integral by a summation.

The DFT for a one dimensional function takes the form

$$F(u) = \sum_{x=0}^{M-1} f(x)\left[\cos\left(\frac{2\pi ux}{M}\right) - j\sin\left(\frac{2\pi ux}{M}\right)\right]$$

In exponential form, it is written as

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M}$$

For $u = 0,1,2,3, \dots\dots\dots\dots\dots\dots\dots\dots\dots M-1$

Similarly, the inverse DFT is given as

$$f(x) = \sum_{u=0}^{M-1} F(u)\left[\cos\frac{2\pi ux}{M} + j\sin\frac{2\pi ux}{M}\right]$$

In exponential form it is given as

$$f(x) = \sum_{u=0}^{M-1} F(u)e^{+j2\pi ux/M}$$

Example : Apply DFT to the following sequence of numbers.

$$x = \{1, 2, 8, 9\}$$

Sol^{n}: Here M=4. The factor 1/M can be included as part of either the forward DFT or inverse DFT. Let us include 1/M as part of the inverse DFT. So the formula for applying DFT is given as

$$F(u) = \sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M}$$

When $u = 0$,

$$F(0) = \sum_{x=0}^{3} f(x)e^{-j2\pi 0x/4}$$

$$= f(0) + f(1) + f(2) + f(3)$$

$$= 1 + 2 + 8 + 9$$

$= 20$

When $u = 1$,

$$F(1) = \sum_{x=0}^{3} f(x) e^{-j2\pi 1x}/_4$$

$$= \left[f(0)e^{0} + f(1)e^{\frac{-j\pi}{2}} + f(2)e^{-j} \ \ + f(3)e^{\frac{-j3\pi}{2}}\right]$$

$$= 1 + 2(-j) + 8(-1) + 9(j)$$

$$= -7 + 7j$$

When $u = 2$,

$$F(2) = \sum_{x=0}^{3} f(x) e^{-j2\pi 2x}/_4$$

$$= [f(0)e^{0} + f(1)e^{-j\pi} + f(2)e^{-j2\pi} + f(3)e^{-j3\pi}]$$

$$= 1 + 2(-1) + 8(1) + 9(-1)$$

$$= -2$$

When $u = 3$,

$$F(3) = \sum_{x=0}^{3} f(x) e^{-j2\pi 3x}/_4$$

$$= \left[f(0)e^{0} + f(1)e^{\frac{-j3\pi}{2}} + f(2)e^{-j3\pi} + f(3)e^{\frac{-j9\pi}{2}}\right]$$

$$= 1 + 2(j) + 8(-1) + 9(-j)$$

$$= -7 - 7j$$

Therefore, the DFT of the sequence is $\{20, -7 + 7j, -2, -7 - 7j\}$

This can be verified by computing the inverse of this sequence. Let us add this factor 1/M as part of this formula. The formula for computing the inverse DFT is

$$f(x) = \frac{1}{M}\sum_{u=0}^{M-1} F(u)e^{\frac{j2\pi ux}{M}}$$

For $x = 0$,

$$f(0) = \frac{1}{4}\sum_{u=0}^{3} F(u)e^{\frac{j2\pi u.0}{4}}$$

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

$$= \frac{1}{4}[20 + (-7 + 7j) + (-2) + (-7 - 7j)]$$

$$= \frac{1}{4} \times 4$$

$$= 1$$

For $x = 1$,

$$f(1) = \frac{1}{4}\sum_{u=0}^{3} F(u)e^{\frac{j2\pi u.1}{4}}$$

$$= \frac{1}{4} \times [F(0)e^{0} + F(1)e^{\frac{j\pi}{2}} + F(2)e^{j\pi} + F(3)e^{\frac{j3\pi}{2}}]$$

$$= \frac{1}{4}[20 + (-7 + 7j)(1) + (-2)(-1) + (-7 - 7j)(1)]$$

$$= \frac{1}{4} \times [20 - 7 + 7j + 2 - 7 - 7j]$$

$$= \frac{1}{4} \times 8 = 2$$

For $x = 2$,

$$f(2) = \frac{1}{4}\sum_{u=0}^{3} F(u)e^{\frac{j2\pi\ \ .2}{4}}$$

$$= \frac{1}{4} \times [F(0)e^{0} + F(1)e^{\frac{j\pi}{2}} + F(2)e^{j2\pi} + F(3)e^{j3\pi}]$$

$$= \frac{1}{4}[20 + (-7 + 7j)(-1) + (-2)(1) + (-7 - 7j)(-1)]$$

$$= \frac{1}{4} \times [20 + 7 - 7j - 2 + 7 + 7j]$$

$$= \frac{1}{4} \times 32 = 8$$

For $x = 3$,

$$f(3) = \frac{1}{4}\sum_{u=0}^{3} F(u)e^{\frac{j2\pi\ .3}{4}}$$

$$= \frac{1}{4} \times [F(0)e^{0} + F(1)e^{\frac{j3\pi}{2}} + F(2)e^{j3\pi} + F(3)e^{\frac{j9\pi}{2}}]$$

$$= \frac{1}{4}[20 + (-7 + 7j)(-1) + (-2)(-1) + (-7 - 7j)(-1)]$$

$$= \frac{1}{4} \times [20 + 7 - 7j + 2 + 7 + 7j]$$

$$= \frac{1}{4} \times 36 = 9$$

Therefore, IDFT of the sequence is $\{1, 2, 8, 9\}$. Hence, the original data is obtained from the inverse DFT.

Stop to Consider

You can find the different values of $\sin\theta$ and $\cos\theta$ from the following curves

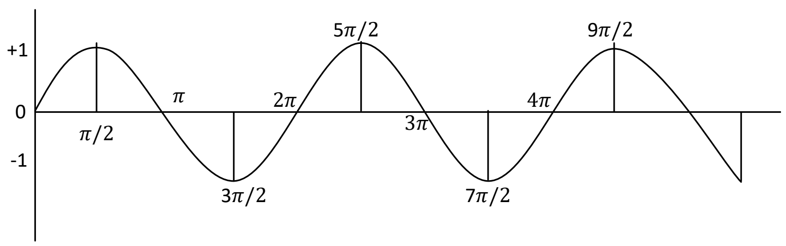
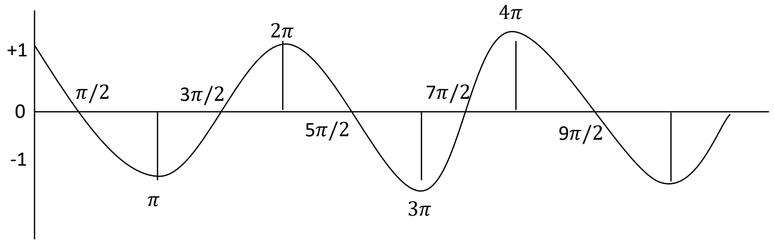


Fig 6.1. Values for sin curve



(b) Two variable: Let $f(x,y)$ for $x = 0,1,2, \dots \dots \dots \dots . M-1$ and $y = 0,1,2, \dots \dots \dots . N-1$ denotes a digital image of size $M \times N$ pixels. The 2D discrete Fourier transform (DFT) of $f(x,y)$, denoted by $F(u,v)$ is given by the equation

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

for $u = 0,1,2, \dots \dots \dots \dots . M-1$ and $v = 0,1,2, \dots \dots \dots . N-1$, where u, v frequency variable. It could be expand the exponential term into sin and cosine functions with the variables u and v determining their frequencies. The frequency domain is the coordinate system spanned by $F(u,v)$ with u and v as frequency variables. The $M \times N$ rectangular region defined by $u = 0,1,2, \dots \dots \dots \dots . M-1$ and $v = 0,1,2, \dots \dots \dots . N-1$ is referred to as frequency rectangle. The frequency rectangle is the same sise as the input image. The inverse discrete Fourier transform (IDFT) is given by

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

for $x = 0,1,2, \dots \dots \dots \dots . M-1$ and $y = 0,1,2, \dots \dots \dots . N-1$

Unitary transform:

A discrete linear transform is unitary if its transform matrix confirmed to the unitary condition.

$$A \times A^H = I$$

Where A - transformation matrix,

A^H – represents Hermitian matrix

$$A^H = {A^*}^T$$

I – identity matrix

When the transform matrix A is unitary, the defined transform is unitary transform.

6.6 Properties of Fourier transform

Some of the Fourier transform properties have been described below

(a) Separable property

The separable property allows a 2D transform to be computed in two steps by successive 1D operations on rows and columns of an image. Mathematically it can be represented as

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

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

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

(i) Performing a 1D transform on each row of image $f(x,y)$ to get $F(x,v)$

(ii) Performing a 1D transform on each column of image $f(x,v)$ to get $F(u,v)$

The main advantage of separability is that a Fourier transform of any dimension can be performed by applying a 1D transform on each dimension. Similarly, in inverse Fourier transform also it is possible to apply separable property.

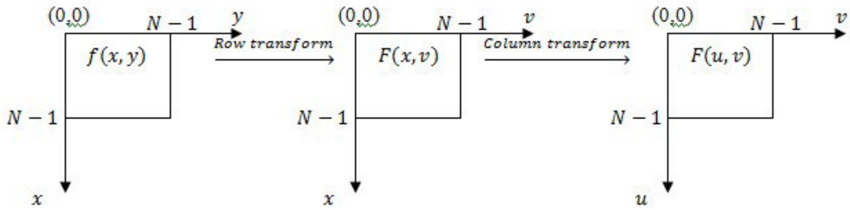


Figure 6.3 : Separable property of Fourier transform

(b) Spatial Shift property

The 2D DFT of a shifted version of the image i.e. $f(x-x_0,y)$ is given by

$$f(x-x_0,y) \xrightarrow{DFT} \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x-x_0,y)e^{\frac{-j2\pi ux}{M}}e^{\frac{-j2\pi vy}{N}}$$
(6.1)

Where x_0 represents the number of times that the function $f(x,y)$ is shifted.

Proof: Adding and subtracting x_0 to $e^{\frac{-j2\pi u}{M}}$ in equation (6.1) we get

$$DFT[f(x-x_0,y)] = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x-x_0,y)e^{\frac{-j2\pi u(x-x_0+x_0)}{M}}e^{\frac{-j2\pi v}{N}}$$
(6.2)

Splitting $e^{\frac{-j2\pi u(x-x_0+x_0}{M}}$ into $e^{\frac{-j2\pi u(x-x_0)}{M}}$ and $e^{\frac{-j2\pi ux_0}{M}}$ results in

$$DFT[f(x-x_0,y)] = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x-x_0,y)e^{\frac{-j2\pi u(x-x_0)}{M}}e^{\frac{-j2\pi ux_0}{M}}e^{\frac{-j2\pi vy}{N}} \quad (6.3)$$

By taking the $e^{\frac{-j2\pi u\ 0}{M}}$ term outside in equation (6.3) we get

$$DFT[f(x-x_0,y)] = e^{\frac{-j2\pi ux_0}{M}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x-x_0,y)e^{\frac{-j2\pi u(x-x_0)}{M}}e^{\frac{-j2\pi vy}{N}} \quad (6.4)$$

From the definition of forward two dimensional discrete Fourier transform, we can write

$$DFT[f(x-x_0,y)] = e^{\frac{-j2\pi ux_0}{M}}F(u,v)$$

This theorem proves that the DFT of a shifted function is unaltered except for a linearly varying phase factor.

(c) Scaling property

Scaling is basically used to increase or decrease the size of an image. According to this property, the expansion of a signal in one domain is equal to compression of the signal in another domain.

$$f(x,y) \xrightarrow{DFT} F(u,v)$$

If DFT of $f(x,y)$ is $F(u,v)$ then

$$f(ax,by) \xrightarrow{DFT} \frac{1}{|ab|} F(u/a, v/b)$$

Proof: The DFT function $f(ax,by)$ is given by

$$f(ax,by) \xrightarrow{DFT} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(ax,by) e^{\frac{-j2\pi u}{N}} e^{\frac{-j2\pi vy}{N}} \quad (6.5)$$

By multiplying and dividing the power of the exponential term $e^{\frac{-j2\pi u}{N}}$ with 'a' and $e^{\frac{-j2\pi vy}{N}}$ with 'b' in equation (6.5)

$$f(ax,by) \xrightarrow{DFT} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(ax,by) e^{\frac{-j2\pi u \ (\frac{a}{a})}{N}} e^{\frac{-j2\pi vy(\frac{b}{b})}{N}} \quad (6.6)$$

$$f(ax,by) \xrightarrow{DFT} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(ax,by) e^{\frac{-j2\pi ax(\frac{u}{a})}{N}} e^{\frac{-j2\pi by(\frac{v}{b})}{N}} \quad (6.7)$$

By substituting DFT equation in equation (6.7) we get

$$f(ax,by) \xrightarrow{DFT} \frac{1}{ab} F(\frac{u}{a}, \frac{v}{b})$$

The scaling theorem tells us that compression in one domain produces a corresponding expansion in the Fourier domain.

(d) Translation property

The translation property state that, if we have a 2D signal $f(x, y)$ and translate it by a vector (x_0, y_0) then the function will be $f(x - x_0, y - y_0)$. Mathematically, it can be written as $f(x, y) \xrightarrow{(x_0,y_0)} f(x - x_0, y - y_0)$. If we take the Fourier transform of the translated signal then it will be the following form.

$$F_t(u, v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x - x_0, y - y_0)e^{-j\frac{2\pi}{N}(u(x-x_0)+v(y-y_0))}$$

By expanding the equation we get

$$F_t(u, v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x - x_0, y - y_0)e^{-j\frac{2\pi}{N}(ux+v\ \ )}e^{-j\frac{2\pi}{N}(ux_0+vy_0)}$$

$$F_t(u, v) = F(u, v)e^{-j\frac{2\pi}{N}(ux_0+vy_0)} \qquad (6.8)$$

From equation (6.8) it is observed that the Fourier spectrum of the signal after translation does not change. It introduces only additional phase difference. So, we can write

$$f(x, y)e^{-j\frac{2\pi}{N}(u_0x+v_0y)} \Leftrightarrow F(u - u_0, v - v_0) \qquad (6.9)$$

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j\frac{2\pi}{N}(ux_0+vy_0)} \qquad (6.10)$$

These two equations (6.9) and (6.10) are known as Fourier transform pair under translation.

(e) Periodicity and Conjugate property

The periodicity property says that both the DFT and IDFT are periodic with a period N.

Proof: The periodicity property says that

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

The Fourier transform $F(u,v)$ is periodic both in x and y directions. The Fourier transform expression is as follows.

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

If we compute Fourier transform of $F(u+N, v+N)$ then we get

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

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

As x, and y is integers the value of $e^{-j2\pi(x+y)}$ always be 1.

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

$$= F(u,v)$$

Therefore,

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

The other property is conjugate. The conjugate property says that if $f(x,y)$ is a real valued function then the Fourier transform will be

$F(u,v) = F^{*}(-u,-v)$

Where F^{*}indicates it is complex conjugate.

(f) Rotation property

The rotation property states that if a function is rotated by the angle, its Fourier transform also rotates by an equal amount. To explain the rotation property we will use polar coordinate. Here we will replace with the following

$x = r\ cos\theta$

$y = r\ sin\theta$

$$u = \omega\ cos\emptyset$$

$$v = \omega\ sin\emptyset$$

$$f(x, y) \Longrightarrow f(r\ cos\theta, r\ sin\theta)$$

$$f(r\ cos\theta, r\ sin\theta) \xrightarrow{DFT} F(\omega \cos \emptyset, \omega \sin \emptyset)$$

$$f(r \cos(\theta + \theta_0), r \sin(\theta + \theta_0))$$

$$\xrightarrow{DFT} F(\omega \cos(\emptyset + \emptyset_0), \omega \sin(\emptyset + \emptyset_0))$$

Stop to Consider

The google colab code to prove the rotation property

```
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from skimage.color import rgb2gray
from skimage.transform import rotate
from numpy.fft import fftshift, fft2
from PIL import Image
image_path = '/content/square.jpg'
# Load the image, convert to grayscale, and normalize
image = Image.open(image_path)
image = np.array(image)
# Convert to grayscale if the image is in RGB
if image.ndim == 3:
    image = rgb2gray(image)
# Rotate the image by 45 degrees
rotated_image = rotate(image, angle=45, resize=True)
# Compute the Fourier Transform of the original and rotated images
original_spectrum = fftshift(np.log(np.abs(fft2(image)) + 1))
rotated_spectrum = fftshift(np.log(np.abs(fft2(rotated_image)) + 1))
# Plot the results
fig, axs = plt.subplots(1, 4, figsize=(15, 5))
# Display original image
axs[0].imshow(image, cmap='gray')
axs[0].set_title("Original Image")
axs[0].axis('off')
# Display rotated image
axs[1].imshow(rotated_image, cmap='gray')
axs[1].set_title("Rotated Image (45°)")
axs[1].axis('off')
# Display spectrum of original image
axs[2].imshow(original_spectrum, cmap='gray')
axs[2].set_title("Original Image Spectrum")
axs[2].axis('off')
# Display spectrum of rotated image
axs[3].imshow(rotated_spectrum, cmap='gray')
axs[3].set_title("Rotated Image Spectrum")
axs[3].axis('off')
plt.show()
```

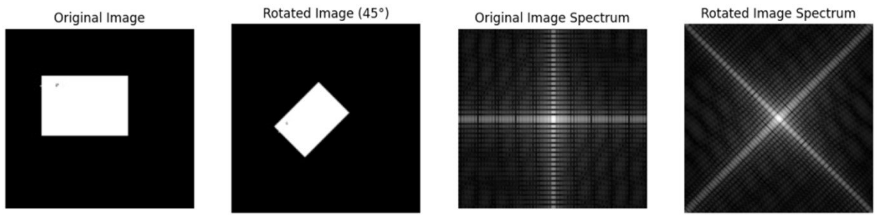


Figure 6.4: Result of Google colab code to show rotation property

(g) Distributive property

The distributive property states that if we take two signals $f_1(x, y)$ and $f_2(x, y)$ and take the summation of both the signal then take the Fourier transform, then

$F\{f_1(x, y) + f_2(x, y)\}= F\{f_1(x, y)\}+ F\{f_2(x, y)\}$

This distributive property is valid only for addition but not valid for multiplication

(h) Convolution property

If we have two functions $f(x)$ and $g(x)$ then the Fourier transform of this two function will be

$$f(x)^{\circ}g(x) \Leftrightarrow F(u) * G(u)$$

$$f(x) * g(x) \Leftrightarrow F(u)^{\circ}G(u)$$

6.7 Convolution theorem

Convolution in the frequency domain is the Fourier transform of a product of time-domain functions or signals. The convolution theorem states that the Fourier transform of a convolution of two functions is equal to the product of their Fourier transforms. This means that convolution in one domain (time domain) is equal to point-wise multiplication in the other domain (frequency domain).

The convolution of two continuous functions or signals $f(t)$ and $g(t)$, of one continuous variable t, denoted by the operator *, is defined as

$$f(t) * g(t) = \int_{-\alpha}^{\alpha} f(\tau) - g(t-\tau)d\tau$$

Where the minus sign accounts for the flipping, t is the displacement needed to slide one function past the other, and τ is a dummy variable that is integrated out. We assume that the functions extend from –α to α. The Fourier transform of convolution operation will be as follows.

$$F(f(t) * g(t) = \int_{-\propto}^{\propto} \left[\int_{-\propto}^{\propto} f(\tau)g(t-\tau)d\tau \right] e^{-j2\pi\mu t} dt$$

$$\int_{-\propto}^{\propto} f(\tau) \left[\int_{-\propto}^{\propto} g(t-\tau)e^{-j2\pi\mu(t-\tau)} dt \right] e^{-j2\pi\mu\tau} d\tau$$

The term inside the brackets is the Fourier transform of $g(t-\tau)$.

$$= \int_{-\propto}^{\propto} f(\tau)H(w)e^{-j2\pi\mu} \; d\tau$$

$$= H(w) \int_{-\propto}^{\propto} f(\tau)e^{-j2\pi\mu\tau} d\tau$$

$$= H(w).F(w)$$

Hence,

$$f(t) * g(t) = F(w).H(w)$$

If we take the convolution of two signals or functions in time domain, then it is equivalent to the multiplication of the two signal in frequency domain. The reverse is also true.

$$F(w) * H(w) = f(t).g(t)$$

Check your Progress

Q1. Check whether the DFT matrix is unitary or not.

6.8 Summing up

- A transform is basically a representation of signal. A transform changes the representation of a signal by projecting it onto a set of basis functions. The transform does not change the information content present in the signal.
- The transformation matrix A is unitary if it obeys the following condition

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

- The transformation matrix A is orthogonal if it obeys the following relation

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

- The king of all transforms is Fourier series and Fourier transform.
- The Fourier transform of the continuous function $f(x)$ of a continuous variable x, is denoted as $F(u)$, where u represents the spatial frequency.

$$F(u) = \int_{-\propto}^{\propto} f(x)e^{-j2\pi ux}dx$$

- The Fourier transform for the 2D continuous function $f(x, y)$ is given by

$$F(u, v) = \int_{-\propto}^{\propto}\int_{-\alpha}^{\alpha} f(x, y)e^{-j2\pi(ux+vy)}\,dxdy$$

and the inverse transform is given by

$$f(x, y) = \int_{-\propto}^{\propto}\int_{-\alpha}^{\alpha} F(u, v)e^{j2\pi(ux+v\ \ )}\,dudv$$

- The DFT for a one dimensional function takes the form

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-j2\pi u}\ /_{M}$$

For $u = 0,1,2,3, \dots \dots \dots \dots \dots \dots \dots \dots \dots M-1$

Similarly, the inverse DFT is given as

$$f(x) = \sum_{u=0}^{M-1} F(u)e^{+j2\pi u\ /M}$$

- The 2D discrete Fourier transform (DFT) of $f(x, y)$, denoted by $F(u, v)$ is given by the equation

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

for $u = 0, 1, 2, \dots \dots \dots \dots . M-1$ and $v = 0, 1, 2, \dots \dots \dots . N-1$, where u, v frequency variable.

- The inverse discrete Fourier transform (IDFT) is given by

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

for $x = 0, 1, 2, \dots \dots \dots \dots . M-1$ and $y = 0, 1, 2, \dots \dots \dots . N-1$

- A discrete linear transform is unitary if its transform matrix confirmed to the unitary condition.

$$A \times A^{H} = I$$

6.9 Answer to Check Your Progress

Solution

Q No 1

Step 1: Determination of the matrix A, Finding 4 – point DFT (where N=4) . The formula to compute a DFT matrix of order 4 is given below.

$$F(u)=\sum_{x=0}^{3} f(x)e^{-j2\pi ux}/_{M}\,, \qquad u=0,1,2,3$$

When $u=0$,

$$F(0)=\sum_{x=0}^{3} f(x)e^{-j2\pi 0x}/_{4}$$

$$=f(0)+f(1)+f(2)+f(3)$$

When $u=1$,

$$F(1)=\sum_{x=0}^{3} f(x)e^{-j2\pi 1x}/_{4}$$

$$=\left[f(0)e^{0}+f(1)e^{\frac{-j}{2}}+f(2)e^{-j\pi}+f(3)e^{\frac{-j3\pi}{2}}\right]$$

$$=f(0)-jf(1)-f(2)+jf(3)$$

When $u=2$,

$$F(2)=\sum_{x=0}^{3} f(x)e^{-j2\pi 2x}/_{4}$$

$$=\left[f(0)e^{0}+f(1)e^{-j}\ +f(2)e^{-j2\pi}+f(3)e^{-j3\pi}\right]$$

$$=f(0)-f(1)+f(2)-f(3)$$

When $u=3$,

$$F(3)=\sum_{x=0}^{3} f(x)e^{-j2\pi 3x}/_{4}$$

$$=\left[f(0)e^{0}+f(1)e^{\frac{-j3\pi}{2}}+f(2)e^{-j3\pi}+f(3)e^{\frac{-j9\pi}{2}}\right]$$

$$=f(0)+jf(1)-f(2)-jf(3)$$

Collecting the coefficients of $F(0), F(1), F(2), and\ F(3)$ we get

$$F(u)=A=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -j & -1 & j\\ 1 & -1 & 1 & -1\\ 1 & j & -1 & -j\end{bmatrix}$$

Step 2: Computation of A^H

To determine A^H, first determine the conjugate and then take its transpose.

$$A \xrightarrow{Conjugate} A^* \xrightarrow{Transpose} A^H$$

Step 2a: Computation of conjugate A^*

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

Step 2b : Determination of transpose of A^*

$$(A^*)^T = A^H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

Step 3: Determination of $A \times A^H$

$$A \times A^H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

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

The result is the identity matrix, which shows that Fourier transform satisfies unitary condition.

6.10 Possible Questions

Q1. What is image transform? Why it is necessary?

Q2. What is Fourier transform. Explain in details.

Q3. What is convolution explain in brief.

Q4. Explain Fourier transform for one continuous variable? Also write the equation for inverse Fourier transform for one continuous variable.

Q5. Check whether DFT matrix is unitary or not.

Q6. Apply DFT to the following sequence of numbers.

$$x = \{2, 4, 6, 7\}$$

6.11 References and Suggested Readings

1. Digital Image Processing, 3ed, Rafael C. Gonzalez, Richard E. Woods, Pearson.
2. Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, McGrawHill
3. Digital Image Processing, 2^{nd} ed,S Sridhar, Oxford
4. NPTEL, IITKGP
5. Digital Image Processing, Dr. Sanjay Sarma, SK Kataria & Sons

*****

UNIT: 7
IMAGE TRANSFORM II

Unit Structure:

7.1 Introduction

Fast Fourier Transform and Inverse Fast Fourier Transform are indispensable algorithms in the field of Digital Signal Processing. They are widely used in different areas of applications such as bio signal data compression, radars, image processing, voice processing etc. FFT algorithm is computationally intensive and need to be processed in real time for most applications. This unit presents some transformation algorithms for 1-D FFT and IFFT.

7.2 Objectives

This unit is an attempt to explain and implement ideas of some image transformation techniques. After going through this unit, you will be able to-

- *apply FFT on an image,*
- *explain the difference between FFT and DFT*,
- *apply* DCT, Walsh and KL transform on images,
- *explain* the mathematical deduction of DCT, Walsh and KL transform.

7.3 Fast Fourier Transform (FFT)

The time complexity of 1D DFT is N^2 and for 2D DFT is N^4. This is quite high for large dataset. This could be simplified and reduce by modifying the 1D DFT and 2D DFT. The separability property could be used for fast implementation of the Fourier Transform. The 1D DFT is written as follows

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j\frac{2\pi}{N}ux} \qquad (7.1)$$

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)W_N^{ux} \qquad \text{Here } W_N = e^{-j\frac{2\pi}{N}}$$

W_N is used for simplification of expression. Let us assumed that the number of samples $N = 2^n$. If the samples can be represented as 2^n, then it also be possible to write N=2M. By substituting the value of N we can rewrite the above equation as follows.

$$F(u) = \frac{1}{2M}\sum_{x=0}^{2M-1} f(x)W_{2M}^{ux}$$

$$F(u) = \frac{1}{2}\left[\frac{1}{M}\sum_{x=0}^{M-1} f(2x)W_{2M}^{u(2x)} + \frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)W_{2M}^{u(2x+1)}\right]$$

(7.2)

Here the first part of the expression gives all the even samples of the input signal from x=0 to M-1. Similarly, the second part of the expression gives all the odd samples of the input signal. Further the expression can be written as

$$F(u) = \frac{1}{2}\left[\frac{1}{M}\sum_{x=0}^{M-1} f(2x)W_M^{ux} + \frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)W_M^{ux}.W_{2M}^{u}\right]$$

(7.3)

From the above expression it is clear that the first expression gives the Fourier Transform of all the even samples of the input signal and the second expression gives the Fourier Transform of all the odd samples of the input signal. Mathematically, it can be written as

$$F(u) = \frac{1}{2}[F_{even}(u) + F_{odd}(u)].W_{2M}^{u} \quad (7.4)$$

From equation (7.4) it is clear that, if we combine both even and odd expression then we get the Fourier Transform for the input signal from 0 to M-1. We can also show that

$W_M^{u+M} = W_M^{u}$ and $W_{2M}^{u+M} = -W_{2M}^{u}$

This tells us that

$$F(u+M) = \frac{1}{2}[F_{even}(u) - F_{odd}(u).W_{2M}^{u}] \quad (7.5)$$

Equation (7.4) gives the coefficients from 0 to M-1 and equation (7.5) gives the coefficients from M to 2M-1. Equation (7.4) and (7.5) gives the coefficients from 0 to N-1. The advantage of this separation is that in the original expression complexity is order of N^2. Now, N numbers of samples have been divided into two equal parts and the time complexity will be $N^2/2$. Further, the odd and even number of samples could be divided recursively, till two samples presents. After computing the Fourier Transform of all the samples and combining it is possible to get the F(u). Finally, the total computation time will be $N\ log_2 N$.

Example: Apply FFT for the sequence of numbers {1, 2, 8, 9}

Solution: Here M=4 as there are four samples and M=2K, therefore K=2

Now apply the FFT formula as follow

$$F_{even}(u) = \frac{1}{K}\sum_{x=0}^{K-1} f(2x)W_K^{ux}$$

$$F_{odd}(u) = \frac{1}{K}\sum_{x=0}^{K-1} f(2x+1)W_K^{ux}$$

When u=0

$$F_{even}(0) = \frac{1}{2}\sum_{x=0}^{1} f(2x)W_2^{0.x} = \frac{1}{2}[1+8] = \frac{9}{2}$$

$$F_{odd}(0) = \frac{1}{2}\sum_{x=0}^{1} f(2x+1)W_2^{0.x} = \frac{1}{2}[2+9] = \frac{11}{2}$$

Now, when u=0 then $F(0)$ can be computed as follows

$$F(0) = \frac{1}{2}[F_{even}(0) + F_{odd}(0).W_4^0] = \frac{1}{2}\left[\frac{9}{2}+\frac{11}{2}\right] = 5$$

When u=1

$$F_{even}(1) = \frac{1}{2}\sum_{x=0}^{1} f(2x)W_2^{1.x} = \frac{1}{2}[1-8] = -\frac{7}{2}$$

$$F_{odd}(1) = \frac{1}{2}\sum_{x=0}^{1} f(2x+1)W_2^{1.x} = \frac{1}{2}[2-9] = -\frac{7}{2}$$

$$F(1) = \frac{1}{2}[F_{even}(1) + F_{odd}(1).W_4^1] = \frac{1}{2}\left[-\frac{7}{2}-\frac{7}{2}(-j)\right]$$

$$= \frac{1}{4}[-7+7j]$$

Once the first half is computed, the second half can be computed as follows

$$F(u+K) = \frac{1}{2}[F_{even}(u) - F_{odd}(u) \times W_{2k}^{u}]$$

Therefore,

$$F(2) = \frac{1}{2}[F_{even}(0) - F_{odd}(0) \times W_4^0] = \frac{1}{2}\left[\frac{9}{2} - \frac{11}{2}(1)\right] = -\frac{1}{2}$$

and

$$F(3) = \frac{1}{2}[F_{even}(1) - F_{odd}(1) \times W_4^1] = \frac{1}{2}\left[-\frac{7}{2} - \left(-\frac{7}{2}\right)(-j)\right]$$

$$= \frac{1}{4}[-7 - 7j]$$

So, the FFT of the input sequence numbers {1, 2, 8, 9} will be

$$5,\ \tfrac{1}{4}[-7 + 7j],\ -\tfrac{1}{2},\ \tfrac{1}{4}[-7 - 7j]$$

7.4 Inverse Fast Fourier Transform (IFFT)

Inverse Fast Fourier Transform (IFFT) is a fast algorithm to perform inverse (or backward) Fourier transform (IDFT), which undoes the process of DFT. IDFT of a sequence $F(u)$ that can be defined as

$$f(x) = \frac{1}{N}\sum_{u=0}^{N-1} F(u)e^{j2\pi ux/N}$$

If an IFFT is performed on a complex FFT result computed by Origin, this will in principle transform the FFT result back to its original data set.

Applications of IFFT

IFFT is commonly used in

a. **Signal Reconstruction**: Given the frequency-domain representation, IFFT reconstructs the original time-domain signal.

b. **Filtering**: In digital signal processing, IFFT is used to implement filters (e.g., low-pass, high-pass) in the time domain.
c. **Channel Equalization**: In communication systems, IFFT compensates for channel distortion caused by multipath propagation.
d. **Audio and Image Compression**: IFFT plays a role in lossy compression techniques like MP3 audio and JPEG image compression.

7.5 Discrete Cosine Transform

The discrete cosine transform was developed by Ahmed, Natrajan and Rao in 1974. It is a technique for converting a signal into elementary frequency components and it is widely used in image compression. The discrete cosine transform (DCT) for one dimensional function takes the form

$F(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) cos\left\{\frac{(2x+1)\pi u}{2N}\right\}$, where $0 \leq u \leq N-1$

$\alpha(u) = \sqrt{\frac{1}{N}}$, if $u = 0$

$\alpha(u) = \sqrt{\frac{2}{N}}$, if $u \neq 0$

The process of reconstructing a set of spatial domain samples from the DCT coefficients is called the inverse discrete cosine transform (IDCT). The inverse discrete cosine transformation is given by

$f(x) = \alpha(u) \sum_{x=0}^{N-1} F(u) cos\left\{\frac{(2x+1)\pi u}{2N}\right\}$, where $0 \leq x \leq N-1$

The forward 2D discrete cosine transform of a signal $f(x, y)$ is given by

$$F(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\, cos\left\{\frac{(2x+1)\pi u}{2N}\right\} cos\left\{\frac{(2y+1)\pi v}{2N}\right\}$$

$\alpha(u) = \sqrt{\frac{1}{N}}$, if $u = 0$

$\alpha(u) = \sqrt{\frac{2}{N}}$, if $u \neq 0$

Similarly,

$\alpha(v) = \sqrt{\frac{1}{N}}$, if $v = 0$

$\alpha(v) = \sqrt{\frac{2}{N}}$, if $v \neq 0$

The 2D inverse discrete cosine transformation is given by

$$f(x,y) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} \alpha(u)\alpha(v)F(u,v)\, cos\left\{\frac{(2x+1)\pi u}{2N}\right\} cos\left\{\frac{(2y+1)\pi v}{2N}\right\}$$

In case of forward transformation $\alpha(u)$ and $\alpha(v)$ are outside the double summation but in case of inverse transformation it is inside the summation.

Example: Compute the discrete cosine transform (DCT) matrix for N=4

Solution: The formula to compute the DCT matrix is given by

$F(u) = \alpha(u)\sum_{x=0}^{N-1} f(x) cos\left\{\frac{(2x+1)\pi u}{2N}\right\}$, where $0 \leq u \leq N-1$

$\alpha(u) = \sqrt{\frac{1}{N}}$, if $u = 0$

$\alpha(u) = \sqrt{\frac{2}{N}}, \qquad \text{if } u \neq 0$

Here, the value of N=4, substituting N=4 in the expression of $F(u)$ we get

$$F(u) = \alpha(u) \sum_{x=0}^{3} f(x) cos \left\{\frac{(2x+1)\pi u}{8}\right\} \qquad (7.6)$$

Substituting u=0 in equation (7.6) we get

$$F(0) = \sqrt{\frac{1}{4}} \times \sum_{x=0}^{3} f(x) cos \left\{\frac{(2x+1)\pi \times 0}{8}\right\}$$

$$= \frac{1}{2} \times \sum_{x=0}^{3} f(x) cos\,(0) = \frac{1}{2} \sum_{x=0}^{3} f(x) \times 1 = \frac{1}{2} \sum_{x=0}^{3} f(x)$$

$$= \frac{1}{2} \times \{f(0) + f(1) + f(2) + f(3)\}$$

$$= \frac{1}{2} \times f(0) + \frac{1}{2} \times f(1) + \frac{1}{2} \times f(2) + \frac{1}{2} \times f(3)$$

Substituting u=1 in equation (7.6) we get

$$F(1) = \sqrt{\frac{2}{4}} \times \sum_{x=0}^{3} f(x) cos \left\{\frac{(2x+1)\pi \times 1}{8}\right\}$$

$$= \sqrt{\frac{1}{2}} \times \sum_{x=0}^{3} f(x) cos \left\{\frac{(2x+1)\pi}{8}\right\}$$

$$= \sqrt{\frac{1}{2}} \times \left\{f(0) cos \left(\frac{\pi}{8}\right) + f(1) cos \left(\frac{3\pi}{8}\right) + f(2) cos \left(\frac{5\pi}{8}\right) + f(3) cos \left(\frac{7\pi}{8}\right)\right\}$$

Here,

$$cos \frac{\pi}{8} = 0.9239$$

$cos\left(\frac{3\pi}{8}\right)$=0.3826

$$cos\left(\frac{5\pi}{8}\right) = -0.3826$$

$$cos\left(\frac{7\pi}{8}\right) = -0.9239$$

$$\therefore F(1) = 0.707 \times \{f(0) \times 0.9239 + f(1) \times 0.3827 + f(2) \times (-0.3827) + f(3) \times (-0.9239)\}$$

$$\therefore F(1) = 0.6532 \times f(0) + 0.2706 \times f(1) - 2706 \times f(2) - 0.6532 \times f(3)$$

Substituting u=2 in equation (7.6) we get

$$F(2) = \sqrt{\frac{2}{4}} \times \sum_{x=0}^{3} f(x) cos\left\{\frac{(2x+1)\pi \times 2}{8}\right\}$$

$$= \sqrt{\frac{1}{2}} \times \sum_{x=0}^{3} f(x) cos\left\{\frac{(2x+1)\pi}{4}\right\}$$

$$= \sqrt{\frac{1}{2}} \times \left\{f(0) cos\left(\frac{\pi}{4}\right) + f(1) cos\left(\frac{3\pi}{4}\right) + f(2) cos\left(\frac{5\pi}{4}\right) + f(3) cos\left(\frac{7\pi}{4}\right)\right\}$$

$$F(2) = 0.7071 \times \{f(0) \times 0.7071 + f(1) \times (-0.7071) + f(2) \times (-0.7071) + f(3) \times (0.7071)\}$$

$$= 0.7071 \times \{f(0) \times (-0.7071) + f(1) \times (-0.7071) + f(2) \times (-0.7071) + f(3) \times (0.7071)\}$$

$$= 0.5 \times f(0) - 0.5 \times f(1) - 0.5 \times f(2) 0.5 \times f(3)\}$$

Substituting u=3 in equation (7.6) we get

$$F(3) = \sqrt{\frac{2}{4}} \times \sum_{x=0}^{3} f(x) cos\left\{\frac{(2x+1)\pi \times 3}{8}\right\}$$

$$= \sqrt{\frac{1}{2}} \times \sum_{x=0}^{3} f(x) cos \left\{\frac{(2x+1)3\pi}{8}\right\}$$

$$= \sqrt{\frac{1}{2}} \times \left\{ f(0) cos \left(\frac{3\pi}{8}\right) + f(1) cos \left(\frac{9\pi}{8}\right) + f(2) cos \left(\frac{15\pi}{8}\right) + f(3) cos \left(\frac{21\pi}{8}\right) \right\}$$

$$= 0.7071 \times \{f(0) \times (0.3827) + f(1) \times (-0.9239) + f(2) \times (0.9239) + f(3) \times (-0.3827)\}$$

$$= 0.2706 \times f(0) - 0.6523 \times f(1) + 0.6533 \times f(2) - 0.2706 \times f(3)\}$$

Collecting all the coefficients of $f(0), f(1), f(2)$ *and* $f(3)$ from $F(0), F(1), F(2)$ *and* $F(3)$ we get

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F[3] \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2705 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix}$$

This is the required DCT matrix

Stop to Consider

Q1. Find the value of $cos\,\frac{\pi}{8}$

Sol^n: From trigonometric rule we know that

$cos2\theta = cos^2\theta - sin^2\theta$

$\Rightarrow cos2\theta = cos^2\theta - (1 - cos^2\theta) \qquad [\because sin^2\theta = 1 - cos^2\theta]$

$\Rightarrow cos2\theta + 1 = 2cos^2\theta$

$\Rightarrow cos^2\theta = \dfrac{1 + cos2\theta}{2}$

$\therefore cos\theta = \sqrt{\dfrac{1 + \cos 2\theta}{2}}$

$$\therefore cos\frac{\pi}{8} = \sqrt{\frac{1 + \cos 2\frac{\pi}{8}}{2}} = \sqrt{\frac{1 + \cos\frac{\pi}{4}}{2}} = \sqrt{\frac{1 + {}^{1}\!/_{\sqrt{2}}}{2}}$$

$$= \sqrt{\frac{\frac{\sqrt{2}+1}{\sqrt{2}}}{2}} = \sqrt{\frac{\sqrt{2}+1}{2\sqrt{2}}} = \sqrt{\frac{1.4142+1}{2\times1.4142}} = \sqrt{\frac{2.4142}{2.82842}} = \sqrt{0.85355} = 0.9239$$

7.6 Walsh Transform

Fourier analysis is the process of representing a signal using a collection of orthogonal sinusoidal waveforms. The waveforms are arranged according to frequency, and the coefficients in this format are referred to as frequency components. To express these functions, Walsh presented a full set of orthonormal square-wave functions in 1923. Walsh functions take only two values, +1 and -1, and are therefore real, which makes them computationally simple. Walsh Transform can be used in many different applications, such as power spectrum analysis, filtering, processing speech and medical signals, multiplexing and coding in communications, characterizing non-linear signals, solving non-linear differential equations, and logical design and analysis. The 1 D walsh transform can be defined as follow.

$$W(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\left[\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}\right] \qquad (7.7)$$

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

Equation (7.7) is equivalent to

$$W(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\left[(-1)^{\sum_{i=1}^{n-1} b_i(x)b_{n-1-i}(u)}\right]$$

The 1 D inverse Walsh transform can be defined as follow.

$$f(x) = \sum_{x=0}^{N-1} W(u)(-1)^{\sum_{i=1}^{n-1} b_i(x)b_{n-1-i}(u)}$$

The array formed by the inverse Walsh matrix is identical to the one formed by the forward Walsh matrix apart from a multiplicative factor N.

The 2 D Walsh transform can be defined as follow.

$$W(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)(-1)^{\sum_{i=1}^{n-1}(b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v))}$$

The 2 D inverse Walsh transform can be defined as follow.

$$f(x,y) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} W(u,v)(-1)^{\sum_{i=1}^{n-1}(b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v))}$$

7.7 Karhunen-Loeve Transform (KL Transform)

The KL transform is named after Kari Karhunen and Michel Loeve who developed it as a series expansion method for continuous random processes. Originally, Harold Hotelling studied the discrete formulation of the KL transform and for this reason, the KL. transform is also known as the Hotelling transform.

The KL transform is a reversible linear transform that exploits the statistical properties of a vector representation. The basic functions of the KL transform are orthogonal eigen vectors of the covariance matrix of a data set. A KL transform optimally decorrelates the input data. After a KL transform, most of the 'energy' of the transform coefficients is concentrated within the first few components. This is the energy compaction property of a KL transform.

Drawbacks of KL Transforms

The two serious practical drawbacks of KL transform are the following:

i. A KL transform is input-dependent and the basic function has to be calculated for each signal model on which it operates. The KL bases have no specific mathematical structure that leads to fast implementations.

ii. The KL transform requires O(m²) multiply/add operations. The DFT and DCT require O(log") multiplications.

Applications of KL Transforms

(i) **Clustering Analysis**: The KL transform is used in clustering analysis to determine a new coordinate system for sample data where the largest variance of a projection of the data lies on the first axis, the next largest variance on the second axis, and so on. Because these axes are orthogonal, this approach allows for reducing the dimensionality of the data set by eliminating those coordinate axes with small variances. This data-reduction technique is commonly referred as Principle Component

(ii) **Image Compression**: The KL transform is heavily utilised for performance evaluation of compression algorithms since it has been proven to be the optimal transform for the compression of an image sequence in the sense that the KL spectrum contains the largest number of zero-valued coefficients.

Example: Perform KL transform for the following matrix.

$$X = \begin{bmatrix} 4 & -2 \\ -1 & 3 \end{bmatrix}$$

Solution

Step 1 Formation of vectors from the given matrix

The given matrix is a 2 x 2 matrix; hence two vectors can be extracted from the given matrix. Let it be x_0 and x_1.

$x_0 = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$ and $x_1 = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$

Step 2 Determination of covariance matrix

` The formula to compute covariance of the matrix is $cov(x) = E[xx^T] - \bar{x}\bar{x}^T$

In the formula for covariance matrix, $\bar{x}$ denotes the mean of the input matrix. The formula to comput the mean of the given matrix is given below:

$$\bar{x} = \frac{1}{M}\sum_{k=0}^{M-1} x_k$$

Where M is the number of vectors in x.

$$\bar{x} = \frac{1}{2}\sum_{k=0}^{1} x_k = \frac{1}{2}\{x_0 + x_1\} = \frac{1}{2}\left\{\begin{bmatrix} 4 \\ -1 \end{bmatrix} + \begin{bmatrix} -2 \\ 3 \end{bmatrix}\right\}$$

$$= \frac{1}{2}\left\{\begin{bmatrix} 2 \\ 2 \end{bmatrix}\right\} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The mean value is calculated as $\bar{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Now multiplying the mean value with its transpose gives

$$\bar{x}\bar{x}^T = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Now to find the $E[xx^T]$ we use the formula $E[xx^T] = \frac{1}{M}\sum_{k=0}^{M-1} x_k x_k^T$

In our case, M=2 hence

$$E[xx^T] = \frac{1}{2}\sum_{k=0}^{1} x_k x_k^T = \frac{1}{2}\left\{\begin{bmatrix} 4 \\ -1 \end{bmatrix}\begin{bmatrix} 4 & -1 \end{bmatrix} + \begin{bmatrix} -2 \\ 3 \end{bmatrix}\begin{bmatrix} -2 & 3 \end{bmatrix}\right\}$$

$$E[xx^T] = \frac{1}{2}\left\{\begin{bmatrix} 16 & -4 \\ -4 & 1 \end{bmatrix} + \begin{bmatrix} 4 & -6 \\ -6 & 9 \end{bmatrix}\right\} = \begin{bmatrix} 10 & -5 \\ -5 & 5 \end{bmatrix}$$

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

Now using the value of $E[xx^T]$ and $\bar{x}\bar{x}^T$, we find the covariance matrix,

$$cov(x) = E[xx^T] - \bar{x}\bar{x}^T$$

$$cov(x) = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix}$$

Step 3 Determination of eigen values of the covariance matrix

To find the eigen values λ, we solve the characteristics equation,

$$|cov(x) - \lambda I| = 0$$

$$det\left(\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

$$det\left(\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) = 0$$

$$det\left(\begin{bmatrix} 1 - \lambda & -2 \\ -2 & -\lambda \end{bmatrix}\right) = 0$$

$$(1 - \lambda)(-\lambda) + 2(-2) = 0$$

$$-\lambda + \lambda^2 - 4 = 0$$

$$\lambda^2 - \lambda - 4 = 0$$

From the above equation, we have to find the eigen values λ_0 and λ_1. Solving the above equation, we get

$$\lambda = \frac{1 \pm \sqrt{1 + 16}}{2} = \frac{1 \pm \sqrt{17}}{2} = \frac{1 \pm 4.1231}{2}$$

$$\lambda_0 = \frac{1 + 4.1231}{2} = 2.5615$$

$$\lambda_1 = \frac{1 - 4.1231}{2} = -1.5615$$

Therefore, the eigen values of $cov(x)$ are $\lambda_0 = 2.5615$ and $\lambda_1 = -1.5615$

Step 4 Determination of eigen vectors of the covariance matrix.

The first eigen vector $\emptyset_0$ is found from the equation,

$$(cov(x) - \lambda_0 I)\emptyset_0 = 0$$

$$(cov(x) - \lambda_0 I)\emptyset_0$$

$$= \left[\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} 2.5615 & 0 \\ 0 & 2.5615 \end{bmatrix}\right]\begin{bmatrix} \emptyset_{00} \\ \emptyset_{01} \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -1.5615 & -2 \\ -2 & -2.5615 \end{bmatrix}\begin{bmatrix} \emptyset_{00} \\ \emptyset_{01} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Using row detection method, we have found $\emptyset_{01}$ to be a free variable. So, we choose the value of $\emptyset_{01}$ as 1.

$$-1.5615\emptyset_{00} - 2\emptyset_{01} = 0$$

$$\emptyset_{00} = \frac{2}{-1.5615} = -1.2808$$

The eigen vector $\emptyset_0 = \begin{bmatrix} -1.2808 \\ 1 \end{bmatrix}$

Similarly, find the next eigen vector $\emptyset_1$; the eigen value is $\lambda_1 = -1.5615$

$$(cov(x) - \lambda_1 I)\emptyset_1$$

$$= \left[\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} -1.5615 & 0 \\ 0 & -1.5615 \end{bmatrix}\right]\begin{bmatrix} \emptyset_{10} \\ \emptyset_{11} \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2.5615 & -2 \\ -2 & -1.5615 \end{bmatrix}\begin{bmatrix} \emptyset_{10} \\ \emptyset_{11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Using row detection method, we have found $\emptyset_{11}$ to be a free variable. So, we choose the value of $\emptyset_{11}$ as 1.

$$2.5615\emptyset_{10} - 2\emptyset_{11} = 0$$

$$\emptyset_{10} = \frac{2}{2.5615} = 0.7808$$

The eigen vector $\emptyset_1 = \begin{bmatrix} 0.7808 \\ 1 \end{bmatrix}$

Step 5 Normalization of the eigen vectors

The normalization formula to normalize the eigen vector $\emptyset_0$ is given below

$$\frac{\emptyset_0}{||\emptyset_0||} = \frac{1}{\sqrt{\emptyset_{00}^2 + \emptyset_{01}^2}} \begin{bmatrix} \emptyset_{00} \\ \emptyset_{01} \end{bmatrix}$$

$$\frac{\emptyset_0}{||\emptyset_0||} = \frac{1}{\sqrt{(-1.2808)^2 + 1^2}} \begin{bmatrix} -1.2808 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.7882 \\ 0.6154 \end{bmatrix}$$

Similarly, the normalization of the eigen vector $\emptyset_1$ is given by

$$\frac{\emptyset_1}{||\emptyset_1||} = \frac{1}{\sqrt{(0.7808)^2 + 1^2}} \begin{bmatrix} 0.7808 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.6154 \\ 0.7882 \end{bmatrix}$$

Step 6 KL transformation matrix from the eigen vector of the covariance matrix

From the normalized eigen vector, we have to form the transformation matrix.

$$T = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}$$

Now we have to check, which is orthogonal (i.e.,)$TT^T = TT^{-1} = I$

$$TT^T = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}$$
$$= \begin{bmatrix} 0.9999 & 0 \\ 0 & 0.9999 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Step 7 KL transformation of the input matrix

To find the KL transform of the input matrix, the formula is $Y = T[x]$

$$Y_0 = T[x_0] = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \end{bmatrix} = \begin{bmatrix} -3.7682 \\ 1.6734 \end{bmatrix}$$

$$Y_1 = T[x_1] = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3.4226 \\ 1.1338 \end{bmatrix}$$

The final transform matrix

$$Y = \begin{bmatrix} -3.7682 & 3.4226 \\ 1.6734 & 1.1338 \end{bmatrix}$$

Step 8 Reconstruction of input values from the transformed coefficients

From the transform matrix, we have to reconstruct value of the given sample matrix X using the formula $X = T^T Y$

$$x_0 = T^T Y_0 = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} -3.7682 \\ 1.6734 \end{bmatrix} = \begin{bmatrix} 3.9998 \\ -1 \end{bmatrix}$$

$$x_1 = T^T Y_1 = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} \begin{bmatrix} 3.4226 \\ 1.1338 \end{bmatrix} = \begin{bmatrix} -1.9999 \\ 2.9999 \end{bmatrix}$$

$$X = [x_0 \quad x_1] = \begin{bmatrix} 4 & -2 \\ -1 & 3 \end{bmatrix}$$

7.8 Check Your Progress

Q No 1: Co Q No 1: Compute the basis of the KL transform for the input data $x_1=(4, 4, 5)^T$, $x_2=(3, 2, 5)^T$, $x_3=(5, 7, 6)^T$ and $x_4=(6, 7, 7)^T$.

7.9 Summing Up

- A transform is basically a representation of signal. A transform changes the representation of a signal by projecting it onto a set of basis functions. The transform does not change the information content present in the signal.

- Different types of image transforms are Fourier, Walsh, Hadamard, Slant, Cosine Sine, KL, Radon and Singular Value Decomposition.
- The transformation matrix A is unitary if it obeys the following condition

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

- The transformation matrix A is orthogonal if it obeys the following relation

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

- The king of all transforms is Fourier transform.
- A set of mutually orthonormal basis functions, with values +1 or -1 constitutes the Walsh transform kernels.
- The computation of Walsh coefficients is simple and it involves only addition and subtraction operations.
- A KL transform depends on the second-order statistics of the Input data. It is an optimal transform with respect to energy compaction.
- Discrete Cosine Transform (DCT) is real and orthogonal. DCT has excellent energy compaction for highly correlated data. DCT is used in JPEG standard.
- Haar functions are non-sinusoidal orthogonal functions. The Haar transform is the simplest discrete wavelet transform.
- The slant transform is real and orthogonal. Slant transforms possess good energy compaction property.
- Some of the applications of image transform include filtering, restoration, compression, enhance- ment and image analysis.

- Many common unitary transforms tend to pack a large fraction of signal energy into a few transform coefficients which is generally termed energy compaction.

7.10 Answer To Check Your Progress

Q. No1

Solution:

Step1 Calculation of mean of the input vectors

$$m_x = \frac{1}{4} \times \begin{bmatrix} 18 \\ 20 \\ 23 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 5.0 \\ 5.75 \end{bmatrix}$$

Step 2 Computation of autocorrelation matrix

The expression for the autocorrelation matrix is given by

$$R_x = \frac{1}{M} \sum_{i=1}^{n} x_i x_i^T - m_x m_x^T$$

Where n is the number of input vectors.

$$m_x m_x^T = \begin{bmatrix} 4.5 \\ 5 \\ 5.75 \end{bmatrix} \times [4.5 \quad 5 \quad 5.75]$$

$$= \begin{bmatrix} 20.25 & 22.5 & 25.875 \\ 22.5 & 25.0 & 28.75 \\ 25.875 & 28.75 & 33.0625 \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1.25 & 2.25 & 0.88 \\ 2.25 & 4.5 & 1.5 \\ 0.88 & 1.5 & 0.69 \end{bmatrix}$$

The eigen values of R_x are $\lambda_1 = 6.1963$, $\lambda_2 = 0.2147$ and $\lambda_3 = 0.0264$

The corresponding Eigen vectors are

$$u_1 = \begin{bmatrix} 0.4385 \\ 0.8471 \\ 0.3003 \end{bmatrix}, u_2 = \begin{bmatrix} 0.4460 \\ -0.4952 \\ 0.7456 \end{bmatrix} \text{ and } u_3 = \begin{bmatrix} -0.7803 \\ 0.1929 \\ 0.5949 \end{bmatrix}$$

From this, the KL transform basis is given by

$$T = \begin{bmatrix} 0.4385 & 0.8471 & 0.3003 \\ 0.4460 & -0.4952 & 0.7456 \\ -0.7803 & 0.1929 & 0.5949 \end{bmatrix}$$

Since the rows of T are orthonormal vectors, the inverse transform is just the transpose $T^{-1}=T^{T}$.

7.11 Possible Questions

Q1. What is Fast Fourier Transform(FFT) ? What is the advantage of FFT over DFT?

Q2 Explain any three image transformation techniques in brief.

Q3 Give the advantages of Walsh transform over Fourier Transform.

7.12 References and Suggested Readings

1. Digital Image Processing, 3ed, Rafael C. Gonzalez, Richard E. Woods, Pearson.
2. Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, McGrawHill
3. Digital Image Processing, 2^{nd} ed,S Sridhar, Oxford
4. NPTEL, IITKGP
5. Digital Image Processing, Dr. Sanjay Sarma, SK Kataria & Sons

*****

UNIT: 8
INTENSITY TRANSFORMATION

Unit Structure:

8.1 Introduction

In this unit, we will discuss some functions of intensity transformation used in image processing. Intensity transformations operate on single pixels of an image for the purpose of image enhancement. The techniques that operate directly on the pixel of an image are known as spatial domain. It is more efficient computationally and requires less processing resources to implement. The spatial domain processes can be denoted by the expression.

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

Where, $f(x, y)$ is the input image, $g(x, y)$ is the output image and T is an operator on f. The operator can apply to a single image or a set of image, such as performing the pixel-by-pixel sum of a sequences of images for noise reduction.

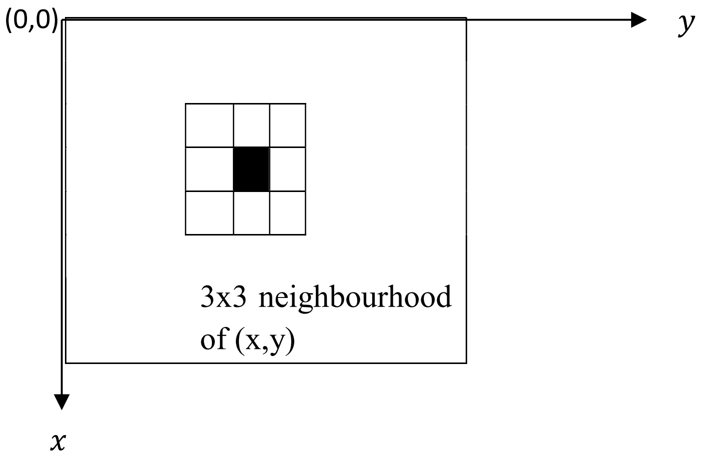


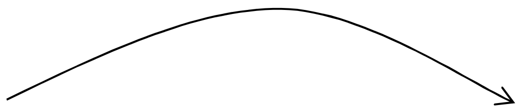
Figure 8.1 : A 3x3 neighbourhood about a point (x,y) in an image in the spatial domain

Figure 8.1 shows the basic idea of intensity transformation on a single image. The point (x, y) shown in an arbitrary location in the image, and the small region shown containing the point is a neighbourhood of (x, y). The process that Figure 1 describes moving the origin of the neighbourhood from pixel to pixel and applying the operator T to the pixel in the neighbourhood to get the output at that location. Thus for any specific location (x, y) the value of the output image g at those coordinates is equal to the result of applying T to the neighbourhood with origin at (x, y) in f. For example, consider a neighbourhood of size 3x3 and the operator T is defined as "compute the average intensity of the neighbourhood". Consider an arbitrary location in an image, say (100,150). Assuming that origin of the neighbourhood is at its centre, the result g(100,150), at that location is computed as the sum of intensity value of $f(100,150)$ and its 8–neighbours, divided by 9. The origin of the neighbourhood is then moved to the next location and the process repeated. Typically, the process starts at the top left of the input image and proceeds pixel by pixel in horizontal scan one row at a time. The smallest possible neighbourhood is of size 1x1. In this case, g depends only on the intensity value of f at a single point (x, y) and T. In general, the intensity transformation function has the form

$$s = T(r)$$

Where, r and s are variables denoting the intensity of f and g at any point (x,y).

Example: Intensity transformation where T is defined as "compute the average intensity of the 3x3 neighbourhood".



$$g(x, y) = \frac{4 + 5 + 6 + 1 + 2 + 3 + 7 + 8 + 9}{9} = 5$$

Figure 8.2 : Intensity transformation

Figure 8.2 depicts the intensity transformation of a pixel using 3x3 neighbourhood.

8.2 Objectives

This unit is an attempt to explain and implement ideas of some intensity transformation functions on gray scale images. After going through this unit, you will be able to

- *apply image negative on gray scale image,*
- *explain the process of intensity transformation functions*,
- *apply* power law transform on images,

- *explain* the process of piecewise linear transformation and histogram processing.

		4	5	6			
		1	2	3			
		7	8	9			

		4	5	6			
		1	5	3			
		7	8	9			

8.3 Image Negative

The negative of an image with intensity levels in the range [0, L-1] is obtained by using the negative transformation. The transformation function may be denoted as $s = L - 1 - r$ where r is the input intensity level and s is the output intensity level. Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative. Image negative is applicable for enhancing white or gray embedded in dark regions of an image. Figure 8.3 shows some basic intensity transformation functions.

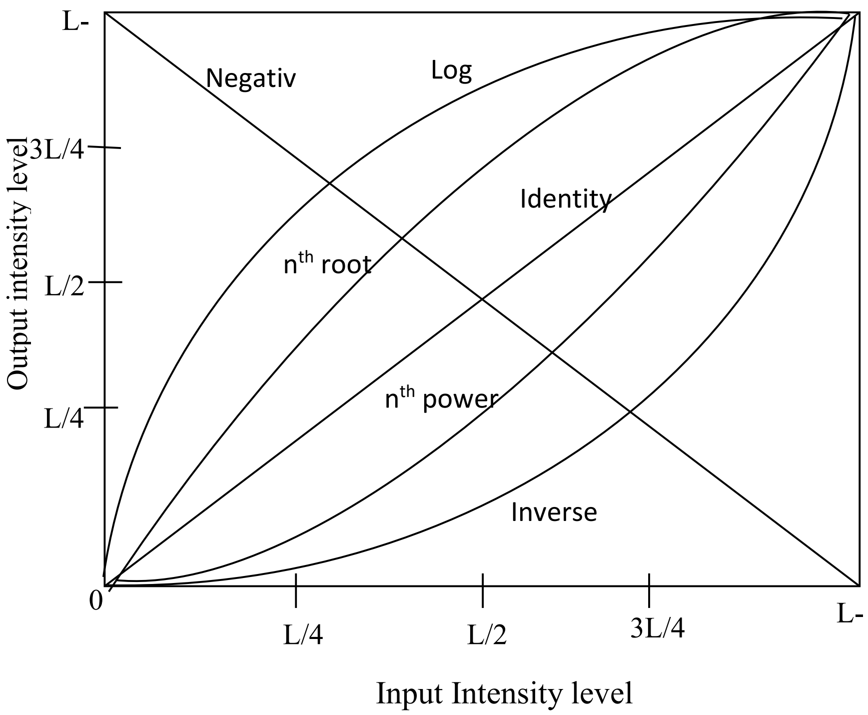


Figure 8.3: Some basic intensity transformation functions.

SAQ

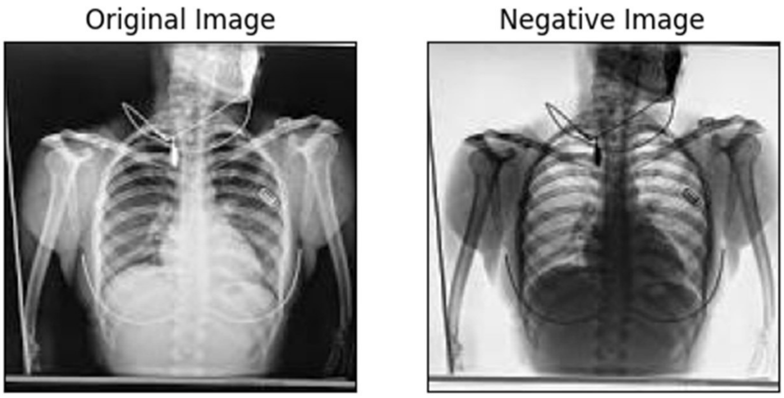
Explain the concept of Intensity transformation

Stop to Consider

Google colab code to read a colour image and find the image negative.

```
#Image Negative
import cv2
import numpy as np
import matplotlib.pyplot as plt
image = cv2.imread('/content/chest.jpeg', 0)
neg_image = 255 - image
plt.subplot(1, 2, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.xticks([]), plt.yticks([])
plt.subplot(1, 2, 2)
plt.imshow(neg_image, cmap='gray')
plt.title('Negative Image')
plt.xticks([]), plt.yticks([])
```

Outputs



8.4 Log Transformations

This transformation transfers a small range of low intensity values in the input into a wider range of output levels, as the shape of the log curve in Fig. 8.3 illustrates. Higher input level values have the opposite effect. This kind of adjustment is what we apply to compress the higher-level values in an image while increasing the values of the dark pixels. The inverse log transformation has the opposite effect. This spreading/compressing of intensity levels in a picture might be achieved by any curve with the overall shape of the log functions illustrated in Fig. 8.3. The general form of the log transformation is

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

Where s is output intensity, r input intensity and c is a constant and value of r is greater than 0.

Stop to Consider

Google colab code to read a colour image and find the log transformation.

```
#Log transformation
import cv2
import numpy as np
import matplotlib.pyplot as plt
c=1
image = cv2.imread('/content/Lena.jpeg', 0)
log_image = c * (np.log(1 +
image.astype(np.float64)))
plt.subplot(1, 2, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.xticks([]), plt.yticks([])
plt.subplot(1, 2, 2)
plt.imshow(log_image, cmap='gray')
plt.title('Log transformation')
plt.xticks([]), plt.yticks([])
```

Outputs

Figure 8.6 : Log transformation

8.5 Power Law (gamma) Transformations

The basic form of power law transformation is defined as

$$s = cr^{\gamma}$$

where c and γ are positive constants. Plots of s versus r for various values of γ are shown in Figure 8.7. As in the case of the log transformation, power-law curves with fractional values of y map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.

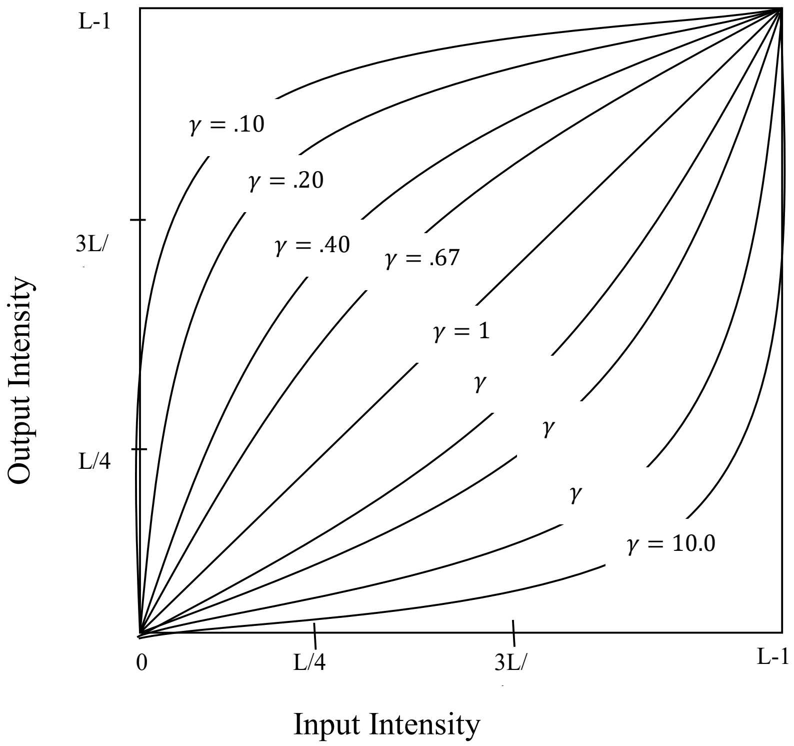


Figure 8.7: Plots of the equation $s = cr^{\gamma}$ for different values of γ(c=1 in all curves)

Stop to Consider

Google colab code to read an image and find the gamma transformation.

```
#Gamma transformation
import cv2
import numpy as np
import matplotlib.pyplot as plt
c=1
gamma=0.5
image = cv2.imread('/content/Lena.jpeg', 0)
gamma_image = c *
np.power(image.astype(np.float64) / 255.0,
gamma) * 255
plt.subplot(1, 2, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.xticks([]), plt.yticks([])
plt.subplot(1, 2, 2)
plt.imshow(gamma_image, cmap='gray')
plt.title('Gamma transformation(gamma=0.5)')
plt.xticks([]), plt.yticks([])
```

Output

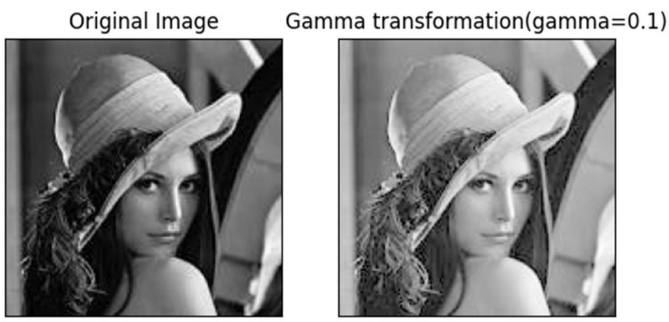


Figure 8.8: Gamma Transform

8.6 Piecewise Linear Transformation

One of the simplest piecewise linear functions is a contrast-stretching transformation. Low-contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even the wrong setting of a lens aperture during image acquisition. Contrast stretching is a process that expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device.

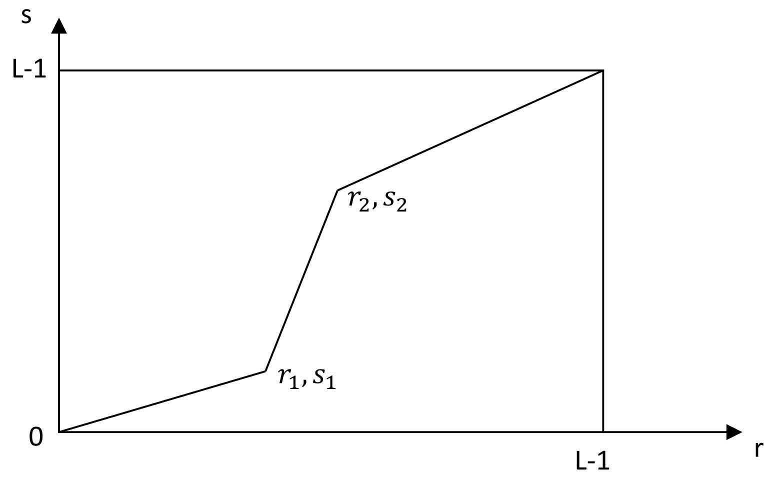


Figure 8.9: Contrast stretching

Figure 8.9 shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function. If r1 = s1 and r2 = s2, the transformation is a linear function that produces no changes in intensity levels. If r1= r2, s1=0 and s2= L-1, the transformation becomes a thresholding function that creates a binary image. Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the intensity levels of the output image, thus affecting its contrast. In general, r1≤r2 and s1≤s2 is assumed so that the function is single valued and monotonically increasing. This condition preserves the order of intensity levels, thus preventing the creation of intensity artifacts in the processed image.

STOP TO CONSIDER

Google colab code to read an image and find the Piecewise linear transformation.

```
#4. Piecewise Linear Transformation
def piecewise_linear_transform(image, r1,
s1, r2, s2):
    # Define a piecewise linear transform
    def piecewise_pixel(val):
        if 0 <= val <= r1:
            return s1 / r1 * val
        elif r1 < val <= r2:
            return ((s2 - s1) / (r2 - r1))
* (val - r1) + s1
        else:
            return ((255 - s2) / (255 -
r2)) * (val - r2) + s2
    vectorized_transform =
np.vectorize(piecewise_pixel)
    return  vectorized_transform(image)

import cv2
import numpy as np
import matplotlib.pyplot as plt
image = cv2.imread('/content/Lena.jpeg',
0)
piecewise_image =
piecewise_linear_transform(image, r1=50,
s1=20, r2=200, s2=130)
plt.subplot(1, 2, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')
plt.xticks([]), plt.yticks([])
plt.subplot(1, 2, 2)
plt.imshow(piecewise_image, cmap='gray')
plt.title('Piecewise image
transformation')
plt.xticks([]), plt.yticks([])
```

Outputs

Figure 8.10: Piecewise linear transformation

8.7 Histogram processing

(a) Histogram: A plot of an image's grey level values against the number of grey level occurrences is called the histogram. The histogram is a useful tool for summarizing the intensities in an image, but it cannot provide information about the spatial relationships between individual pixels. More information on image brightness and contrast can be found in the histogram. The following are some properties of image histogram.

1. A dark image's histogram will be grouped towards the lower grey level.

2. A bright image's histogram will be grouped towards a higher grey level.

3. The histogram for a low-contrast image will be narrow since it won't be dispersed evenly.

4. The histogram will have an equal spread in the grey level for a high contrast photograph.

In order to enhance an input image's visual quality, histogram manipulation essentially alters the image's histogram. A fundamental understanding of the image's histogram is required in order to understand histogram manipulation. The following section provides an overview of image histograms and the histogram-equalization approach, which enhances an image's visual quality.

(b**) Histogram Equalisation**: Equalisation is the process of distributing an image's grey levels such that they are equally distributed throughout its range. Pixel brightness values are reassigned using histogram equalisation using the image histogram as a guide. Using a technique called histogram equalisation, the resulting image's histogram is made as flat as possible. Equalisation of histograms provides more visually pleasing results across a wider range of images.

(c) Steps to Perform Histogram Equalisation

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 8.1: Perform histogram equalisation of the following gray scale image.

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

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	5	4	5	6	5	0	0

Step 1 Compute the running sum of histogram values.

The running sum of histogram values is known as cumulative frequency distribution.

Gray Level	0	1	2	3	4	5	6	7
Number of Pixels	0	5	4	5	6	5	0	0
Running Sum	0	5	9	14	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	5	4	5	6	5	0	0
Running Sum	0	5	9	14	20	25	25	25
Running Sum/Total number of pixels	0/25	5/25	9/25	14/25	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	5	4	5	6	5	0	0
Running Sum	0	5	9	14	20	25	25	25
Running Sum/Total number of pixels	0/25	5/25	9/25	14/25	20/25	25/25	25/25	25/25
Multiply the above result by maximum gray level	$\frac{0}{25} \times 7$	$\frac{5}{25} \times 7$	$\frac{9}{25} \times 7$	$\frac{14}{25} \times 7$	$\frac{20}{25} \times 7$	$\frac{25}{25} \times 7$	$\frac{25}{25} \times 7$	$\frac{25}{25} \times 7$

The result is then rounded to the nearest integer to obtain the following table.

Gray Level	0	1	2	3	4	5	6	7
Number of Pixels	0	5	4	5	6	5	0	0
Running Sum	0	5	9	14	20	25	25	25
Running Sum/Total number of pixels	0/25	5/25	9/25	14/25	20/25	25/25	25/25	25/25
Multiply the above result by maximum gray level	0	1	3	4	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	1
2	3
3	4
4	6
5	7
6	7
7	7

The original image and histogram equalized image are as follows

$$\begin{bmatrix} 2 & 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 & 5 \\ 4 & 3 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \xrightarrow{\text{Histogram Equalization}} \begin{bmatrix} 3 & 4 & 4 & 4 & 4 \\ 7 & 7 & 7 & 7 & 7 \\ 6 & 6 & 3 & 3 & 3 \\ 6 & 6 & 6 & 6 & 6 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Original image — Histogram equalized image

Stop to Consider

Google colab code to read an image and find the Histogram equalization.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
# Load the image in grayscale
image = cv2.imread('/content/Lena.jpeg', 0)
# Apply histogram equalization
equalized_image = cv2.equalizeHist(image)
# Plot original and equalized image
plt.figure(figsize=(8, 4))
plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(image, cmap='gray')
plt.axis('off')
plt.subplot(1, 2, 2)
plt.title('Equalized Image')
plt.imshow(equalized_image, cmap='gray')
plt.axis('off')
```

(Continued…)

Stop to Consider

```
plt.show()
# Plot the histograms
plt.figure(figsize=(8, 4))
plt.subplot(1, 2, 1)
plt.title('Original Histogram')
plt.hist(image.ravel(), 256, [0, 256])
plt.xlim([0, 256])
plt.subplot(1, 2, 2)
plt.title('Equalized Histogram')
plt.hist(equalized_image.ravel(), 256, [0,
256])
plt.xlim([0, 256])
plt.show()
```

Output

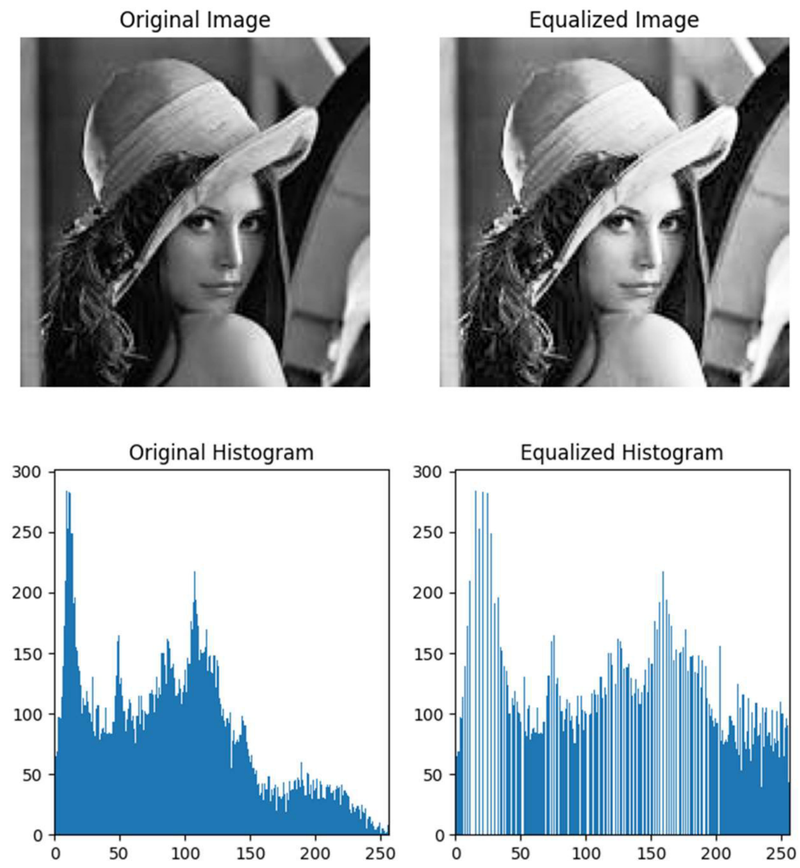


Figure 8.11: Results of histogram equalization

8.8 Check Your Progress

Q No 1: Perform histogram equalisation of the following gray scale image.

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 0 \\ 5 & 5 & 5 & 0 & 5 \\ 4 & 3 & 0 & 2 & 2 \\ 4 & 0 & 4 & 4 & 4 \\ 0 & 0 & 0 & 6 & 6 \end{bmatrix}$$

8.9 Summing Up

- Intensity transformation is the process of reassigning the pixel intensity values to enhance the image quality.
- Image negative is the transformation function that reverts the intensity of an image.
- Log transformations, power law transformations, and piecewise linear transformations are the functions that reassign the pixel intensity of an image.
- Histogram processing is a technique that gives the intensity relationship among pixels of an image.

8.10 Answer to Check Your Progress

Q. No1

Solution: The maximum value is found to be 6. 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	7	0	6	1	5	4	2	0

Step 1 Compute the running sum of histogram values.

The running sum of histogram values is known as cumulative frequency distribution.

Gray Level	0	1	2	3	4	5	6	7
Number of Pixels	7	0	6	1	5	4	2	0
Running Sum	7	7	13	14	19	23	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	7	0	6	1	5	4	2	0
Running Sum	7	7	13	14	19	23	25	25
Running Sum/Total number of pixels	7/25	7/25	13/25	14/25	19/25	23/25	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	7	0	6	1	5	4	2	0
Running Sum	7	7	13	14	19	23	25	25
Running Sum/Total number of pixels	7/25	7/25	13/25	14/25	19/25	23/25	25/25	25/25
Multiply the above result by maximum gray level	$\frac{7}{25} \times 7$	$\frac{7}{25} \times 7$	$\frac{13}{25} \times 7$	$\frac{14}{25} \times 7$	$\frac{19}{25} \times 7$	$\frac{23}{25} \times 7$	$\frac{25}{25} \times 7$	$\frac{25}{25} \times 7$

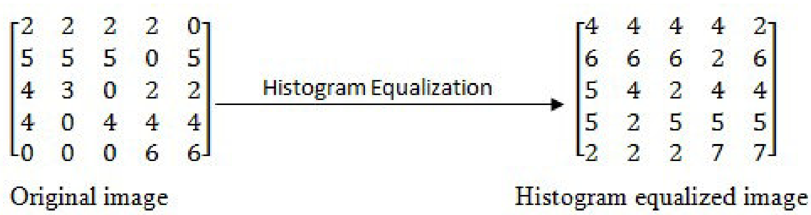
The result is then rounded to the nearest integer to obtain the following table.

Gray Level	0	1	2	3	4	5	6	7
Number of Pixels	7	0	6	1	5	4	2	0
Running Sum	7	7	13	14	19	23	25	25
Running Sum/Total number of pixels	7/25	7/25	13/25	14/25	19/25	23/25	25/25	25/25
Multiply the above result by maximum gray level	2	2	4	4	5	6	7	7

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

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

The original image and histogram equalized image are as follows



This is the required boundary of the image or object.

8.11 Possible Questions

Q1. What is intensity transformation? Explain intensity transformation with examples?

Q2 Explain any three intensity transformation function

Q3 Write python code to perform the followings.

I. Image Negatives
II. Log Transformations
III. Power Law Transformations
IV. Piecewise Linear Transformation
V. Histogram equalization

8.12 References and Suggested Readings

1. Digital Image Processing, 3ed, Rafael C. Gonzalez, Richard E. Woods, Pearson.

2. Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, McGrawHill

3. Digital Image Processing, 2^{nd} ed,S Sridhar, Oxford

4. NPTEL, IITKGP

5. Didital Image Processing, Dr. Sanjay Sarma, SK Kataria & Sons

*****

Block: II

Unit 1: Fundamentals of Spatial Filtering
Unit 2: Filtering in Frequency Domain
Unit 3: Image Restoration
Unit 4: Morphological Image Processing
Unit 5: Basic Morphological Algorithms
Unit 6: Colour Image Processing

UNIT: 1
FUNDAMENTALS OF SPATIAL FILTERING IN IMAGE PROCESSING

Unit Structure:

1.1 Introduction

The spatial domain refers to the image plane and the different processing method which are based on the direct manipulation of pixels of an image. There are two major categories of spatial processing one is intensity transformation and the other is spatial filtering. As we learn that the intensity transformation operate on a single pixel and it's main purpose is of contrast manipulation and the image thresholding. But the spatial filtering deals with performing operation which includes image sharpening, by working in a neighborhood of every pixel in an image. Filtering refers accepting or rejecting some of the frequency components. For example if the filter passes through the low frequencies is called lowpass filter. The effect of the low pass filter is to blur or smooth an image. We can carry out similar smoothing directly on the image itself by using spatial filters.

1.2 Objectives

After going through this unit learner will able to-

- *understand* concept of spatial filtering,
- *learn* the concept of the mechanics of spatial filtering,
- *learn* different image enhancement techniques in spatial domain,
- *understand* the concept of smoothing and sharpening spatial filters.

1.3 The Mechanics of Spatial Filtering

The spatial filter consists of a neighborhood, typically a small rectangle and a predefined operation that is performed on the image pixels surrounded by the neighborhood. Filtering will creates a new pixel and the coordinates of this pixel is equal to the coordinates of the centre of the neighborhood. A filtered image is generated as the center of filter visits all the pixels in the input image. If the operation performed on the image pixel linearly, then the filter is called a linear spatial filter otherwise the filter is called nonlinear filter. The Figure 1.1 explains the mechanics of the linear spatial filtering using 3×3 neighborhood. At any point (x,y) in the image, the response g(x,y) of the filter is shown in the following equation (1.1)

$$g(x,y) = w(-1,-1)\, f(x-1, y-1) + w(-1,0)\, f(x-1, y) + \ldots + w(0,0)\, f(x, y) + \ldots + w(1,1)\, f(x+1, y+1) \quad \ldots\ldots(1.1)$$

Here we observe that the centre coefficient of the filter, w(0,0), align with the pixel at the location(x,y). For a mask m×n , we assume that m = 2a + 1 and n=2b + 1 , a and b are two positive integers. This means we only focus on filters of odd size. Generally linear spatial filtering of an image of size M × N with a filter of size m × n is given by the following equation (1.2).

$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$.......(1.2)

where x, y are two variable and each pixel in w visits every pixel in f.

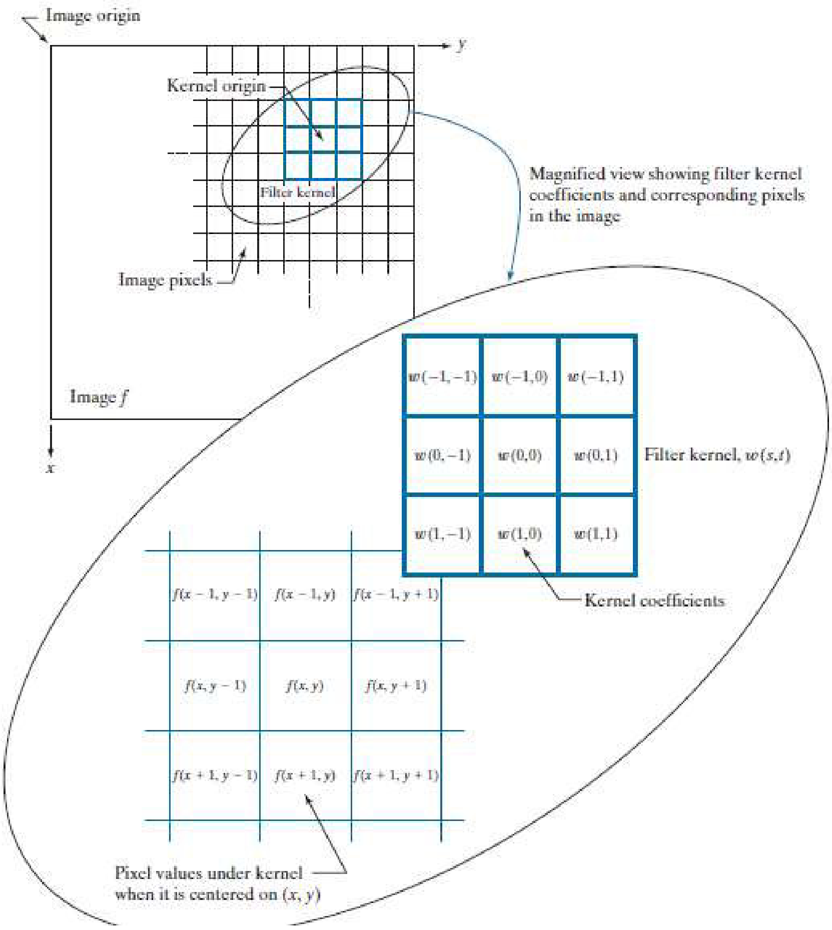


Figure 1.1 The mechanics of linear spatial filtering

Image source: (Gonzalez & Woods, 2016)

1.4 Image Enhancement in spatial domain

As we discussed earlier that image enhancement is the process of manipulating an image so that the result is more suitable in comparison to the original image for a particular application. There is no any general theory of image enhancement. In the image processing, image enhancement is one of the most visually appealing area.

1.4.1 Some Basic Intensity Transformation Functions

As an introduction to intensity transformation there are basic types of functions used frequently for image enhancement. These functions are as follows

- **Image Negatives**

  The negative of an image with the intensity level is in the range [0, L-1]. This is obtained by using the negative transformation which is shown in the Figure 1.2 and is can be expressed by the equation (1.3) as follows

  s = L - 1 - r …………………….(1.3)

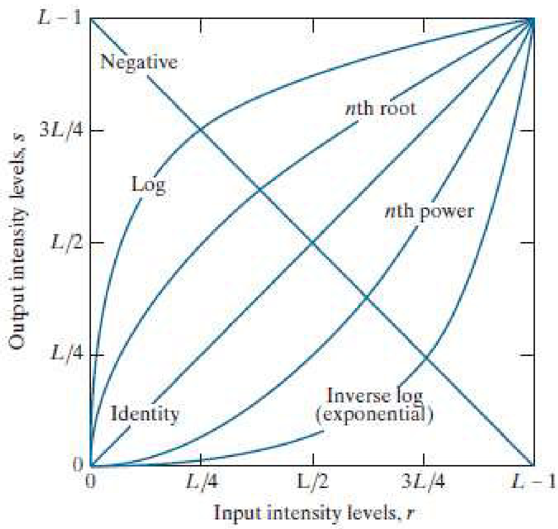


Figure 1.2 Some basic intensity transformation functions

Image source: (Gonzalez & Woods, 2016)

If the intensity level of an image is reversed, then it will produce the equivalent of a photographic negative. This type of processing is particularly suited for enhancing white or gray detail embedded in a dark region of an image.

- **Log Transformations**

The general form of the log transformation in Figure 1.2 is expressed as shown in the equation (1.4)

$$s = c \log (1 + r) \qquad (1.4)$$

Here c → is a constant and $r \geq 0$. If we consider the long curve in the Figure (1.2) shows that the transformation maps a narrow range of low intensity values in the input into wider range of output levels. The reverse is true of higher values of input levels. The uses of this type of transformation to expand the values of dark pixels in an image while compressing the higher-level values. The opposite is true for the inverse log transformation.

- **Power-Law (Gamma) Transformations**

The basic form of Power-Law transformation is shown in the equation (1.5)

$$s = cr^{\gamma} \qquad (1.5)$$

Where c and γ are positive constants. In some of the cases the equation can also be written as $s = c\,(r+\epsilon)^{\gamma}$ to account for an offset. The varieties of devices are used for capturing, printing and displaying the respond according to a power law. The exponent of the power law equation is referred to as gamma. The process used to correct these power-law responses in different situation is called gamma correction.

- **Piecewise-linear transformation Functions**

The form of piecewise linear function can be arbitrarily complex. But in some situations, some important transformation can be formulated only as piecewise functions. The major disadvantage of piecewise linear transformation function is that their specification considerably requires more user input.

- **Contrast stretching**

One of the simplest level of piecewise linear functions is a contrast stretching transformation. It is a process of that expands the range of intensity levels in an image so that it can be spans the full intensity range of the recording medium or display device.

- **Intensity-level slicing**

In an image, focusing on a specific range of intensities often is of interest. The process of some applications such as enhancing flaws in X-ray images, enhancing features such as masses of water in satellite imagery are the example of intensity level slicing. These processes can be implemented in several ways, but most are variations of two basic themes. One approach is to display in one value say white of all the values in the range of interest and in another say black all other intensities. This transformation produces a binary image which is shown in Figure 1.3. The second approach based on the transformation which is shown in the Figure 1.4, brightens or darkens the desired range of intensities but leaves all other intensity levels unchanged in the image.

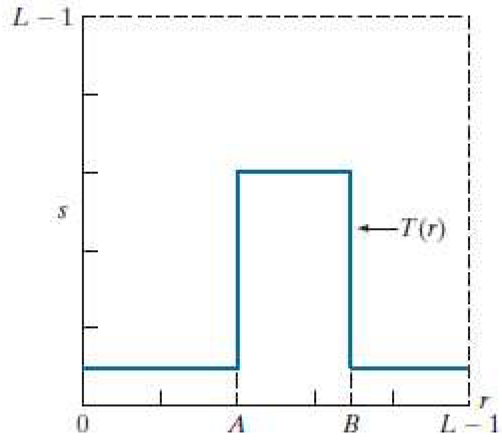


Figure 1.3 The transformation highlights intensity range [A, B] and reduces all other intensities to a lower level.
Image source: (Gonzalez & Woods, 2016)

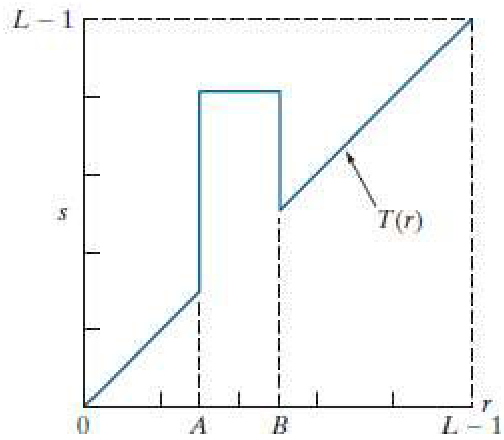


Figure 1.4: This transformation highlights range [A,B] and preserves all other intensity levels
Image source: (Gonzalez & Woods, 2016)

Stop to Consider

In addition to gamma correction, power-law transformations are useful for general-purpose contrast manipulation

1.4.2 Histogram Processing

Histograms are the basis for numerous spatial domain processing techniques. Histogram manipulation can be used for image enhancement. Histograms are simple to calculate in software and it makes them a popular tool for real time image processing. The histogram of a digital with the intensity level in the range [0, L-1] is a discrete function $h(r_k) = n_k$, where r_k is the kth intensity value and n_k is the number of pixels in the image with the intensity r_k.

A normalized histogram is given by the $p(r_k) = n_k / N$, for k= 0,1,2,………., L -1. N is the total number of pixels.

In histogram processing, there are four basic intensity characteristics: dark, light, low contrast and high contrast. The components of the histogram of the dark images are concentrated on the low side of the intensity scale. Similarly, the components of the histogram of the light images are biased toward the high side of the scale. An image with low contrast has a narrow histogram and placed toward the middle of the of the intensity scale. Finally an image with high-contrast covers a wide range of the intensity scale.

- **Histogram Equalization**

Histogram equalization is a technique used in image processing to improve the contrast of an image. A perfect image is which have equaled no pixels in all each gray level. Hence our objective is not only to spread the dynamic range but also to have equal pixel in all the grey levels. As usual, we assume that r is in the range [0, L -1], with r =0 representing black and r = L – 1 representing white. We consider the transformation of the form which is shown in the equation (1.6)

$$s = T(r) \qquad 0 \leq r \leq L-1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.6)$$

The equation (1.6) produces an output intensity level s for every pixel in the input image with the intensity r. We assume for this purpose as follows

- T(r) is a monotonically increasing function in the interval $0 \leq r \leq L - 1$ and
- $0 \leq T(r) \leq L - 1$ for $0 \leq r \leq L - 1$

For a discrete value we deal with the probabilities and the summation instead of probability density functions. As mentioned earlier, the probability of occurrence of intensity level r_k in a digital image is approximate by the equation (1.7)

$$p_r(r_k) = \frac{n_k}{N} \quad k = 0,1,2,\ldots\ldots\ldots\ldots,L-1 \quad \ldots\ldots\ldots\ldots(1.7)$$

Where, N is the total of pixels in the image.

Example: Equalize the given histogram for 3-bit image (L=8).

Gray Level	0	1	2	3	4	5	6	7
No of pixels	790	1023	850	656	329	245	122	81

Solution: The above histogram can be equalized which is shown in the table 1.1 .

Table 1.1: Histogram equalized Table

Gray Level	n_k	PDF($p_r(r_k)$ $=\frac{n_k}{N}$	CDF $s_k = \sum(pr(rk)$	Equalized value= (L-1) × s_k	Rounding off
0	790	0.1929	0.1929	1.33	1
1	1023	0.2498	0.4427	3.08	3
2	850	0.2075	0.6502	4.55	5
3	656	0.1602	0.8104	5.67	6
4	329	0.0803	0.8907	6.23	6
5	245	0.0598	0.9505	6.65	7
6	122	0.0298	0.9803	6.86	7
7	81	0.0198	1.0000	7	7

Here, N = 4096

PDF → Probability Density Function

CDF → Cumulative Distribution Function

Comparison of the values of old and new gray level is shown in the Table 1.2.

Table 1.2: Comparison of gray level values

Old gray level	Number of pixels	New gray level
0	790	1
1	1023	3
2	850	5
3	656	6 Sum up
4	329	6
5	245	7
6	122	7 Sum up
7	81	7

Since the gray level 3 and 4 mapped to the same value 6 so there are 985(656 +329) pixels in the equalized image with this value. Similarly, gray level 5, 6 and 7 mapped to the same value 7 so there are 448(245+122+81) pixels with a value of 7 in the histogram equalized image. Final Equalized histogram table with pixel values is shown in the table 1.3

Table 1.3: Final equalized histogram table with pixel values

Equalized gray level	Number of pixels
0	0
1	790
2	0
3	1023
4	0
5	850
6	985
7	448

Check Your Progress-I

1. **Multiple Choice Questions**

(i) Low pass filters passes through the

a) High frequency

b) Low frequency

c) Moderate frequency

d) None of these

(ii) Histogram manipulation can be used for

a) Image segmentation

b) Image enhancing

c) Image restoration

d) Threshoding

(iii)The process that expands the range of intensity levels of a image is known as

a) Image negative

b) Image positive

c) Contrast stretching

d) Image sorting

(iv) The process used to correct the power-law responses in different situation is called

a) gamma correction.

b) image correction

c) negative thresholding

d) log correction

2. State True or False

(i) The form of piecewise linear function can be arbitrarily complex.

(ii) Histograms are the basis for numerous spatial domain processing techniques.

(iii) An image with high contrast has a narrow histogram.

(iv) Filtering refers accepting or rejecting some of the frequency components

1.5 Smoothing Spatial Filters

Smoothing filter is basically used for reducing the noise and for blurring the images. Image blurring is used in the preprocessing tasks which includes removal of small details from an image prior to large object extraction. Noise reduction can be achieved by blurring the images with linear and nonlinear filters.

1.5.1 Smoothing linear filters

The output or results of smoothing, linear spatial filter is the average of the pixels contained in neighborhood of the filter mask. In some cases these filters are called averaging filters. The average filters are also called low-pass filters. The idea behind the smoothing filter is achieved by replacing the value of every pixel in an image by the average intensity levels in the neighborhood defined by the filter mask. This will reduce sharp transition in intensities. The major use of average filters is in the reduction of irrelevant details in an image. The Figure 1.5 shows two 3 × 3 smoothing filters.

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

Figure 1.5 Two 3 × 3 smoothing filter mask

The first filter produces the standard average of the pixels under the mask. This can be expressed by subtracting the coefficients of the mask as in the equation (1.8).

$$\mathrm{R} = \frac{1}{9} \sum_{i=1}^{9} z_i \text{ .... (1.8)}$$

The above equation is the average of the intensity levels of the pixels in the 3 × 3 neighborhood defined by the mask. The first filter is computationally more efficient to have coefficients valued 1. At the end of the filtering process the entire image is divided by 9. An m × n mask would have a normalizing constant which is equal to 1/mn. In some cases, such types of spatial smoothing (averaging) filter in which all the coefficients are equal is called a box filter.

In the second mask, the pixel at the centre of the mask is multiplied by the higher value than any other. This mask will produce a weighted average, which will give more importance that means weight to some pixels at expense of others. This pixel is more importance in the calculation of the average. The sum of all the coefficient of the second mask is equal to 16. As 16 is the power of 2, it is an attractive feature for computer implementation. In practical it is not an easy task to see differences between images smoothed by using these two masks shown in Figure 1.5 or the similar arrangement. This is because the area spanned by these masks at any one location in an image is so small. An example of image smoothing with masks of various size are shown in the Figure 1.6 which shows an original image and the its corresponding smoothed result. The result is obtained using square averaging filters of sizes m = 3, 5, 9,15 and 35 pixels respectively. The black square at the top of the image are of sizes 3,5,9,15,25,35,45 and 55 pixels and their border are 25 pixels. The letters in the bottom of the images are in size of 10 to 24 points which are the increment of 2 points and that are 10,12,14,16,18,20,22 and 24 respectively. The large letter at the top is 60 points and the vertical bars are 5 pixels wide and 100 pixel high. The diameters of the circles are 25 pixels and their borders are 15 pixels apart. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

Figure 1.6 Results of smoothing with square averaging filter
Image source: (Gonzalez & Woods, 2016)

For m = 3 ,we observe general slight blurring throughout the entire image , but the details that are of approximately the same size as the filter mask are affected considerably more. The result for m = 5 is somewhat similar with a slight further increase in blurring. For m = 9, we observe considerably more blurring. The results for m = 15 and 35 are extreme with respect to the sizes of the objects in the image. This type of strong blurring generally used to eliminate small objects from an image. In the final image of Figure 1.3 three small squares, two of the circles and most of the noisy rectangle areas have been blended into the background of the image.

1.5.2 Order-Statistic (Nonlinear) Filters

Order-static filters are the non-linear spatial filter. The response of non-linear filter is based on ordering the pixels contained in the image area surrounded by the filter and then replacing the value of the centre pixel with the value determined by the ranking result. The example of non-linear filter is the median filter. By virtue of name median filter replaces the value of a pixel by the median of the intensity values in the neighborhood of that pixel. For certain types of random noises, the median filter gives tremendous result in noise reduction. When median filter is used the images are less blurring than the linear smoothing filters of similar size. The median filters are quite effective in the presence of impulsive noise, which is also called salt-and-pepper noise. In the impulsive or salt-and-pepper noise the white and black dots superimposed on an image. In linear filters like the mean or Gaussian filter compute a weighted sum of neighboring values, the median filter uses the ranking of the values. In median filter, the process of finding the median involves it will first sorting the pixel values in the neighborhood and then selecting the middle value. Suppose a 3 × 3 neighborhood has values (34, 7, 200,6,9,3,8,15,1). Now these values are sorted as (1,3,6,7,8,9,15,34,200) which results in a median of 8. The centre value in this case 9 is replaced by 8.

1.6 Sharpening Spatial Filters

The main objective of sharpening is to highlight transitions in intensity. The requirement of image sharpening is varies in different situations. We have already discussed that image blurring could be accomplished in the spatial domain by pixel averaging in a neighborhood. It is logical to conclude that sharpening can be accomplished by the spatial differentiation. There are various ways of defining and implementation of operators for sharpening by digital differentiation. The derivatives of a digital function are defined in terms of differences. There are various ways to define these differences. However any definition we use for a first derivative required the followings

- First derivative must be zero in areas of constant intensity
- First derivative must be non-zero at the onset of an intensity step or ramp
- First derivative must be nonzero along ramps.

In the similar way any definition of a second derivative

- Must be zero in areas of constant intensity
- Must be non-zero at the onset and end of an intensity step or ramp
- Must be zero along ramps of constant slope.

As we dealing with the digital quantities whose values are finite, the maximum possible intensity change also is finite. The shortest distance over which the change can occur is between adjacent pixels.

Using the second order derivative for image sharpening is basically consists of defining a discrete formulation and then produces a filter mask based on that formulation. Here isotropic filters can be used for this purpose. The response of isotropic filter is independent of the direction of the discontinuities in the image to which the filter is applied. So it can be stated that isotropic filters are rotation invariant. The term rotation invariant means rotating the image and then applying the filter gives the same result as applying the filter first and then rotating the result.

The simplest isotropic derivative operator is the Laplacian, which can be derive for a function f(x,y) of two variable is defined in the equation (1.9)

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(1.9)}$$

To express this equation (1.9), we have to carry a second variable. In x-direction the equation (1.9) can be expressed and this can be shown in the equation (1.10)

$\frac{\partial^2 f}{\partial x^2}$ = f(x + 1,y) + f (x -1,y) – 2f (x,y)………………(1.10)

Similarly the equation (1.9) can be shown in the y-direction and it can be shown in the equation (1.11)

$\frac{\partial^2 f}{\partial y^2}$ = f(x, y +1) + f(x, y-1) -2f(x,y) …………(1.11)

Therefore it follows from preceding three equations that the discrete Laplacian of two variables can be expressed as in the equation (1.12)

∇^2f = f(x + 1,y) + f (x -1,y) + f(x, y +1) + f(x, y-1) - 4f(x,y)

…………………….(1.12)

The equation (1.11) can be implemented using the filter mask which are shown in the Figure 1.7 . This mask gives an isotropic result for rotation in increments of 90^0. The mask shown in the Figure 1.8 is used to implement an extension of the equation (1.11) that includes the diagonal terms. The masks shown the Figure (1.9) are the two other implementation of the Laplacian found frequently in practice.

0	1	0
1	-4	1
0	1	0

Figure 1.7 Filter mask used to implement equation (1.11)

1	1	1
1	-8	1
1	1	1

Figure 1.8 Mask is used to implement an extension of the equation (1.11) that includes the diagonal terms.

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 1.9 Masks implementation of Laplacian found frequently in practice.

- **Unsharp Masking and Highboost Filtering**

The process that has been used to sharpen images consists of subtracting an unsharp smoothed version of an image from the original image is called unsharp masking. The process of unsharp masking has the following steps:

(i) Blur the original image

(ii) Subtract the blur image from the original. Here the subtraction result is called the mask.

(iii) Add the mask to the original.

Now let us consider $\bar{f}$(x,y) denote the blurred image, the unsharp masking is expressed in the equation(1.13)

g_{mask} (x,y) = f(x,y) - $\bar{f}$(x,y)…………………………………..(1.13)

Then we add a weighted portion of the mask back to the original image which is shown in the equation (1.14)

g (x,y) = f(x,y) + k * g_{mask}(x,y)………………………………(1.14)

Where we included a weight, k (k ≥0), for generality. When k =1, we have unsharp masking , as defined above. The process is defined as highboost filtering, when k > 1. Choosing k < 1 minimize the contribution of the un-sharp mask.

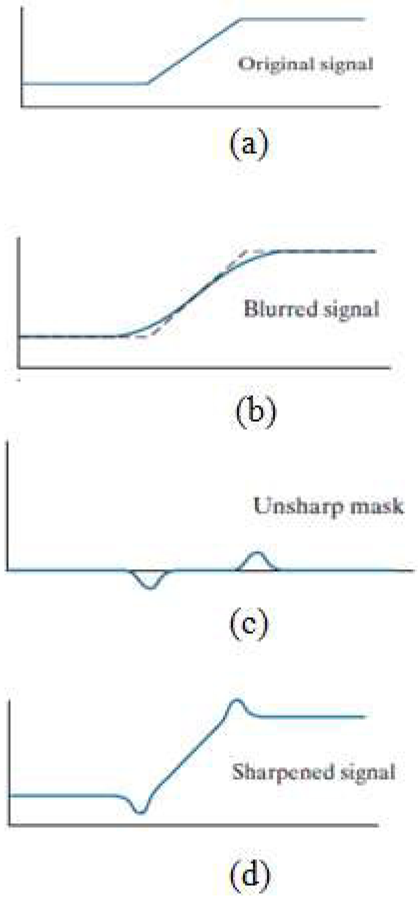


Figure 1.10: Illustration of the mechanics of unsharp masking
Image source: (Gonzalez & Woods, 2016)

Figure 1.10 explains the working of unsharp masking. The intensity profile in Figure 1.10(a) can be interpreted as a horizontal scan line through a vertical edge that transition from a dark to a light region in an image. Figure 1.10 (b) shows the result of smoothing, superimposed on the original signal which is shown as dashed line. Figure 1.10 (c) is the unsharp mask, obtained by subtracting the blurred signal from the original one. Final sharpened result is shown in the Figure 1.10 (d) which is obtained by adding the mask to the original signal.

Check Your Progress-II

3. State True or False

(i) The simplest isotropic derivative operator is the Laplacian.

(ii) Image sharpening is constant in different situations.

(iii) The median filters are effective in the presence of salt-and-pepper noise.

(iv) Gaussian filter is an example of non linear filters.

(v) The average filters are also called low-pass filters.

1.7 Summing up:

- The spatial domain refers to the image plane and the different processing method which are based on the direct manipulation of pixels of an image.
- The spatial filtering deals with performing operation which includes image sharpening, by working in a neighborhood of every pixel in an image.
- There is no any general theory of image enhancement. In the image processing, image enhancement is one of the most visually appealing area.

- Filtering refers accepting or rejecting some of the frequency components.
- If the operation performed on the image pixel linearly, then the filter is called a linear spatial filter otherwise the filter is called nonlinear filter.
- Histograms are simple to calculate in software and it makes them a popular tool for real time image processing.
- In histogram processing, there are four basic intensity characteristics which are dark, light, low contrast and high contrast.
- Smoothing filter is basically used for reducing the noise and for blurring the images.
- Order-static filters are the non-linear spatial filter. The response of non-linear filter is based on ordering the pixels contained in the image area surrounded by the filter and then replacing the value of the centre pixel with the value determined by the ranking result.
- The main objective of sharpening is to highlight transitions in intensity and the requirement of image sharpening is varies in different situations.
- The process that has been used to sharpen images consists of subtracting an unsharp smoothed version of an image from the original image is called unsharp masking.
-

1.8 Answer to check your progress

1.(i) (b) (ii) (b) (iii) (c) (iv) (a)

2. (i) True (ii) True (iii)False (iv) True

3. (i) True (ii) False (iii)True (iv)False (v) True

1.9 Possible Questions

1. Explain the mechanics of spatial filtering.
2. What is image negative?
3. Explain the log transformation.
4. Explain the piecewise-linear transformation Functions.
5. What is histogram processing?
6. Explain the histogram equalization process.
7. What are the uses of smoothing linear filter in image processing?
8. How are non-linear filters reduce the noise of an image?
9. Explain the main objectives of sharpening spatial filters.
10. Explain the various ways of defining and implementation of operators for sharpening by digital differentiation.
11. How is isotropic derivative operator derived for the function f(x,y)?
12. What are the uses of median filters?
13. What is unsharp masking?
14. Explain the mechanics of unsharp masking.
15. Explain the steps consists of unsharp masking process.

1.12 References and Suggested Readings

Gonzalez & Woods (2016). *Digital image processing*. Pearson education india.

Jain, A. K. (1989). Fundamentals of digital image processing. Prentice-Hall, Inc..

Digital Image Processing, Dr. Sanjay Sarma, SK Kataria & Sons

NPTEL, IITKGP

Digital Image Processing, S Jayaraman, S Esakkirajan, T Veera Kumar, Mc Graw Hill.

*****

UNIT: 2
FILTERING IN FREQUENCY DOMAIN

Unit Structure:

2.1 Introduction

In the previous chapter we have learned in details about the Fourier transform and its importance in image processing. As we have now a very good knowledge about the concepts of Fourier transform in continuous domain, Discrete Fourier transform, properties of Fourier transform and convolution theorem, we can proceed towards understanding the basics of filtering in frequency domain. We will also learn the smoothing and the sharpening technique in the frequency domain. We will conclude the unit with the concept of Homomorphic filtering.

2.2 Objectives

After going through this unit, you will learn-

- *basic* concepts of filtering in frequency domain,
- *image* smoothing using frequency domain filters,
- *different* types of filters for image smoothening,
- *image* sharpening using frequency domain filters,
- *different* types of filter for image sharpening,
- *homomorphic* filtering.

2.3 Basics of Filtering in Frequency Domain

Filtering in frequency domain is a powerful technique in image processing that involves modifying the Fourier Transform of an image to get a processed image by computing the inverse transform of that image. The basic filtering equation of a digital image f(x,y) with size MXN has the form:

$$g(x,y) = \mathfrak{I}^{-1} [H(u, v) F(u, v)] \text{--------------------------} (1)$$

where $\mathfrak{I}^{-1}$ is the IDFT,

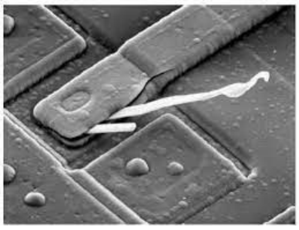
F(u, v) is the DFT of the input image f(x, y),

H(u, v) is the filter function

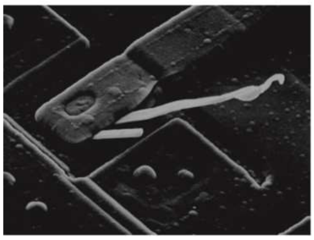
g(x, y) is the filtered image.

F,H and g are all arrays with size same as the input digital image i.e MXN. H (u, v) F (u, v) is the product derived from array multiplication. Filter function is responsible for transforming the input digital image into the processed image g(x, y)

The simplest filter that we can construct is a filter H(u, v) that is 0 at the center of the transform and 1 elsewhere. The filter will reject only the dc term and all other terms will be passed. As dc term is responsible for the average intensity of an image, setting it to zero will reduce the average intensity of the output image to zero. Figure 1 [1]illustrates this fact



a)



b)

Fig 1 a) shows the original image of a damaged integrated circuit and b) shows the output of the original image after applying the filter

We can see the output image is much darker due to presence of negative intensities. Though the above concept is illustrated in the Figure but this may not be considered as true representation as all the negative intensities were clipped. Low intensities in the transform are caused due to slowly varying components in an image whereas high frequencies are caused due to sharp transitions in intensity.

A low pass filter H(u, v) is a filter that passes low-frequency components while attenuating high-frequency components.On the other hand, a high pass filter H(u, v) is a filter that passes high frequency components while attenuating low-frequency components. High pass filters also eliminate the dc term; thus its result will be same as shown in Figure 1 b). Now if we add a small constant value to the filter, it does not affect sharpening but it helps prevent the elimination of the dc term thus preserving tonality.

Thus, after adding a small constant value, it becomes similar to the original image in Figure 2 a). [1]

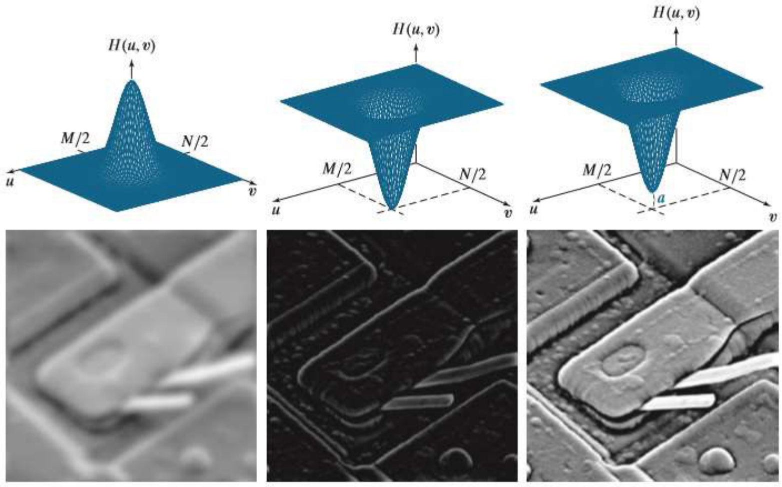


Fig 2: The top row exhibits different types of frequeny domain filters. The bottom row shows the corresponding filtered images derived by using equation 1. In the rightmost is the result of applying a small constant value a=0.85 to the high pass filter which results in a image that is similar to the original image in Figure 1.

A Gaussian low pass filter eliminates high frequency (sharp) features and the practical effect on the image is the blurr effect thus losing the details. The result of blurring is not uniform in the image. So, padding is applied to the input image to avoid the wraparound error.

Let us consider two functions f(x) and h(x) composed of A and B samples respectively. Now if we append zeros to both the functions so that they may have the same length, denoted by P then wraparound error is avoided by choosing

$$P \geq A+B+1 \text{---------------------------------} 2$$

Padding is applied to the input image by using the following equations respectively.

$$P \geq 2M - 1 \text{----------------------------------} 3$$

And

$$Q \geq 2N-1 \text{------------------------------------} 4$$

The resulting padded images are of size PXQ. If both the arrays are of same size MXN then we require the above two conditions. After padding equation 1 is being applied to the padded image. The effect of Gaussian filter with an without padding is demonstrated in Figure 4[1]

a) b) c)

Figure 4:a)The leftmost image is the original image. b) The image in the center is the result of blurring with a Gaussian lowpass filter without padding. c) The rightmost image is the result of low pass filtering with padding. The change in the center and the right image can be observed if vertical edges are considered.

2.3.1 Steps for Filtering in the Frequency domain

The steps for filtering in frequency domain are:

1. Given an input image f(x,y) of size MXN, obtain the padding parameters P and Q from equations 3 and 4. Typically we select P=2M and Q=2N.

2. Form a padded image, $f_p(x, y)$, of size PXQ by appending the necessary number of zeros to f(x, y).

3. Multiply $f_p(x, y)$ by $(-1)^{x+y}$ to center it's transformed.

4. Compute the DFT, F(u,v), of the image from step 3.

5. Generate a real, symmetric filter function , H(u,v) of size PXQ with center at coordinates (P/2, Q/2). Form the product G(u, v) = H (u, v) F(u, v) using array multiplication; that is G(i, k)F(i,k).

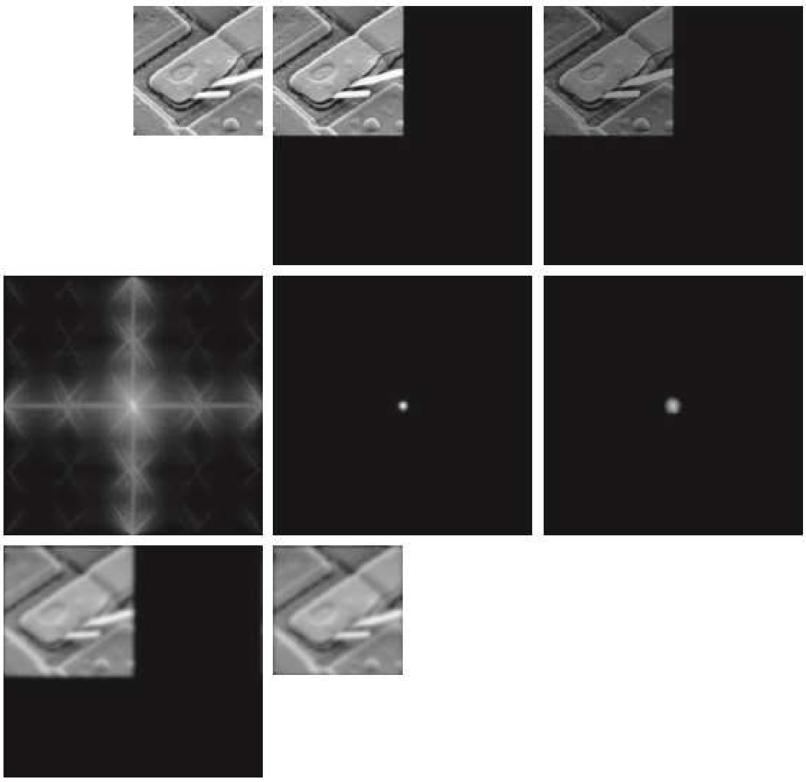
6. Obtain the processed image:

$$g_p(x, y) = \{real[\Im^{-1} [G(u, v)]]\}(-1)^{x+y}$$

where the real part is selected in order to ignore parasitic complex components resulting from computational inaccuracies and the subscript p indicates dealing with padded arrays.

7. Obtain the final processed result, g(x, y), by extracting the MXN region from the top, left quadrant of $g_p(x,y)$.

Figure 5 [1] illustrates the steps of filtering in frequency domain.



a b c

d e f

g h

Figure 5: a) An MXN image f. b) Padded image f_p of size PXQ. c) Result of multiplying f_p by $(-1)^{x+y}$. d) Spectrum of $f_{p.}$ e) Centred Gaussian lowpass filter, H, of size PXQ. f) Spectrum of the product HF_p g) $g_{p,}$ the product of $(-1)^{x+y}$ and the real part of the IDFT of $HF_{p.}$h) Final result, g, obtained by cropping the first M rows and N columns of g_p.

2.4 Image Smoothing Using Frequency Domain Filters

Smoothing or blurring in Frequency domain is achieved by high frequency attenuation that is by low pass filtering. Mainly three types of low pass filters are considered: ideal, Butterworth and Gaussian. These three low pass filters cover the range from very sharp (ideal) to very smooth (Gaussian) filtering. The Butterworth filter has a parameter filter order which determines how the filter is going to behave. For higher order values, the Butterworth filter becomes like an ideal filter. On the other hand, for lower order values, the Butterworth filter is more like a Gaussian filter. Thus, Butterworth filter may be viewed as providing a transition between two extremes. All filtering follows the steps described above , so all filter functions, H(u, v) are understood to be discrete functions of size PXQ; the discrete frequency variables are in the range u=0,1,2……P-1 and v=0,1,2………..Q-1.

2.4.1 Ideal Lowpass Filters

A 2D low pass filter that cuts off all high frequency components of the DFT that are at a distance greater than a specified distance D_0 from the origin of transform is called an ideal low pass filter (ILPF). An ILPF is specified by the function:

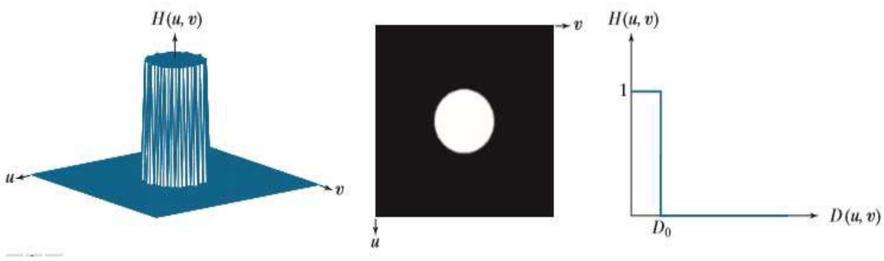
$$H(u,v) = \begin{cases} 1, if\ D(u,v) \le D_0 \\ 0, if\ D(u,v) > D_0 \end{cases} \text{----------------------5}$$

Where D_0 is a positive constant and D(u, v) is the distance between a point (u, v) in the frequency domain and center of the frequency rectangle; that is

$$D(u, v) = [(u- P/2)^2 + (v- Q/2)^2]^{1/2} \text{---------------------} 6$$

where P and Q are the padded sizes.

Figure 6 [1] shows an ideal low pass filter. The ILPF is radially symmetric about the origin, which means that the filter is completely defined by a radial cross section. Rotating the cross section by 360^0 yields the filter in 2-D

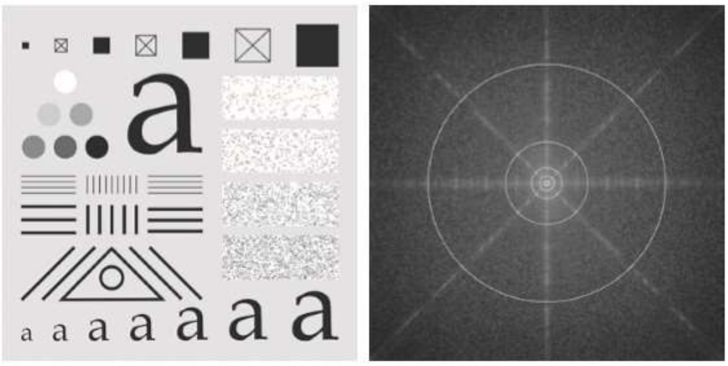


a) b) c)

Figure 6: a) An ideal low pass filter b) Filter displayed as an image c) Filter radial cross section.

For an ILPF cross section, the point of transition between H(u, v) =1 and H(u, v) = 0 is called the cutoff frequency.

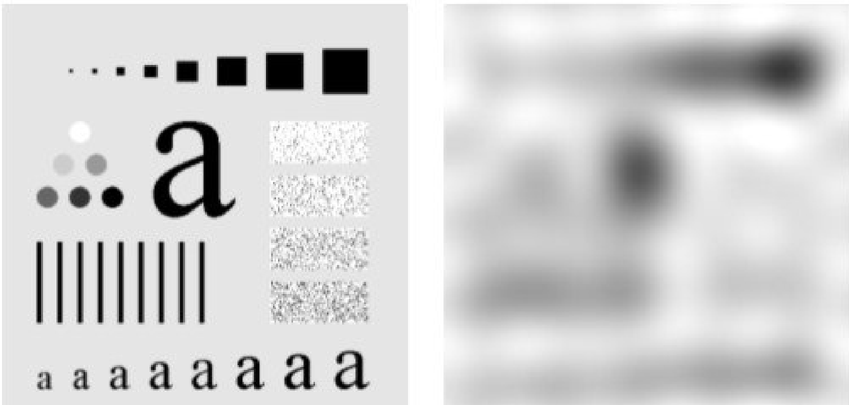
The next figure 7 [1] shows a grayscale image and it's corresponding Fourier Spectrum. The circles superimposed on the spectrum have radii of 10,30,60,160 and 460 pixels respectively.



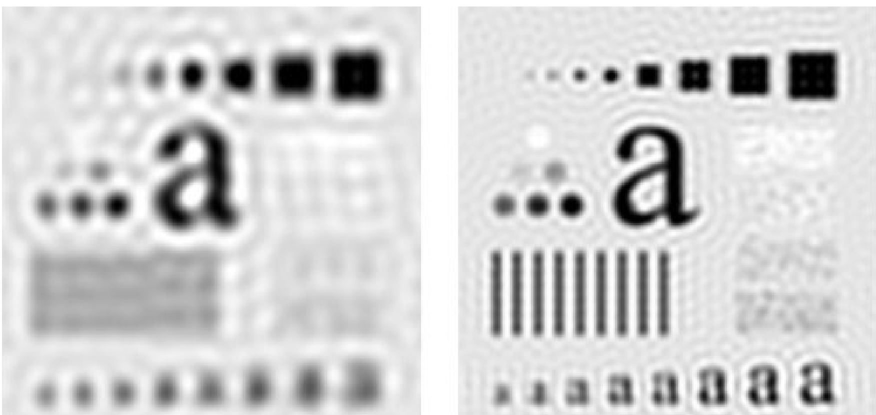
a) b)

Figure 7: a) Grayscale image b) Fourier Spectrum

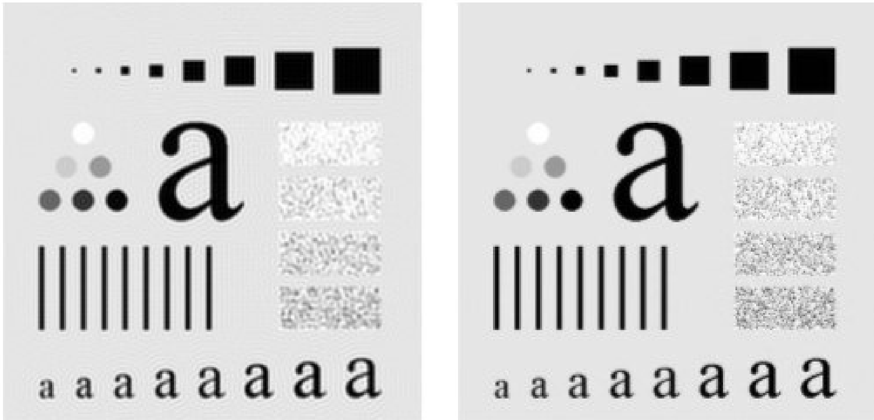
Applying ILPF to the above grayscale image with cut off frequencies set at the previous radii values 10, 30, 60, 160 and 460 pixels will result in Figure 8 b)-f).[1]



a) b)



c) d)



e) f)

Figure 8: a) Original image. b)-f) Results of filtering using ILPF's with cutoff frequencies set at the radii values 10,30, 60, 160 and 460 respectively. The power removed by these filters was 13, 6.9, 4.3, 2.2 and 0.8% of the total, respectively.

The severe blurring in the image is a clear indication that most of the sharp detail information in the picture is contained in the 13% power removed by the filter. Again, we can also see Ringing effect in the images which produces dark and light ripples around bright features of an image.

We can clearly see the effects of ILPF in the images given in Figure 8 and conclude that:

a) Blurring effect decreases as cutoff frequency increases.

b) Ringing effect becomes finer as cutoff frequency decreases.

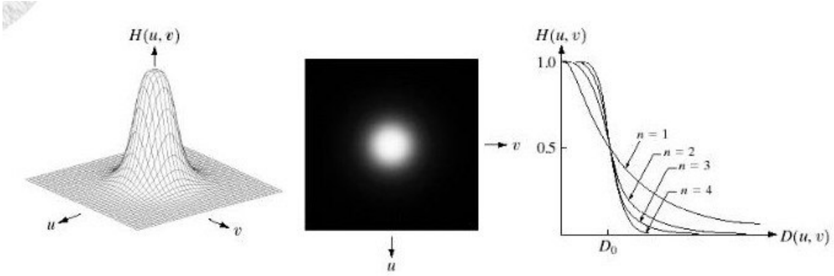
2.4.2 Butterworth Lowpass Filters

The transfer function of Butterworth low pass filter (BLPF) of order n with cutoff frequency at a distance D_0 from the origin is defined as

$$H(u, v) = \frac{1}{1+[\frac{D(u,v)}{D_0}]^{2n}} \text{------------------------------} 7$$

Where D(u, v) is given by equation 6.

Figure 9[1]displays the results of a Butterworth lowpass filter.

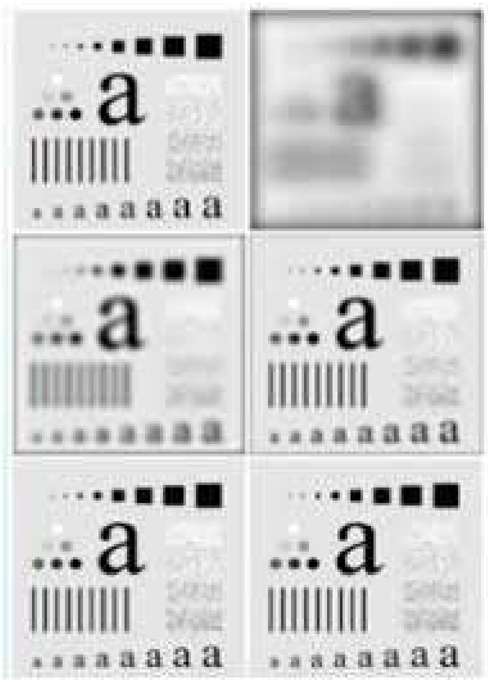


a) b) c)

Figure 9: a) Perspective plot of Butterworth low pass filter transfer function. b) Filter displayed as an image. c) Filter radial cross sections of orders 1 through 4.

BLPF transfer function does not have a sharp discontinuity like ILPF that gives a clear cut off between passed and filtered frequencies.

Figure 10[1] shows the effect of using BLPF's of order 2 with different cutoff frequencies.



a b
c d
e f

Figure 10: a) Original image. b)-f) Results of filtering BLPF's of order 2 with cutoff frequencies at the radii equal to 10, 30, 60, 160 and 460 respectively.

From figure 10 we can observe that in BLPF, there is a smooth transition in blurring as a function of increasing cutoff frequency and there is no ringing effect which was observed in ILPF.

A BLPF of order 1 has no ringing in the spatial domain. Ringing effect is negligible in case filters of order 2 but can become distinct when there are higher order filters. So, a BLPF of order 1 have neither ringing effect nor negative values. A BLPF of order 2 have mild ringing and small negative values. So BLPF of filter order 2 can be considered as compromise between effective low pass filtering and acceptable ringing. On the other hand, with higher order values for example a BLPF with order 20 exhibits characteristics of ILPF.

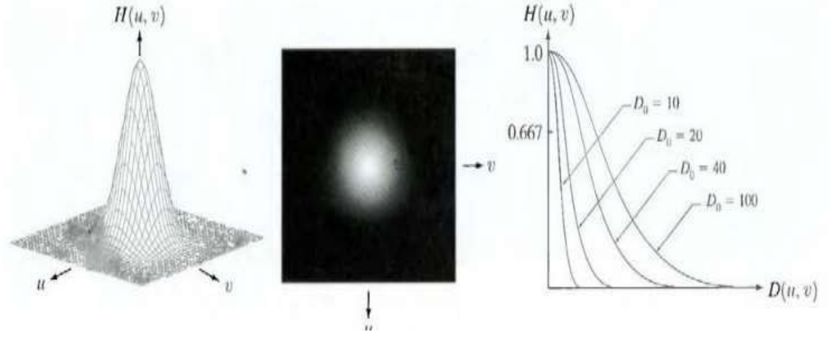
2.4.2 Gaussian Lowpass Filters

The transfer function of Gaussian Lowpass Filter in two dimensions is given by

$$H(u, v) = e^{-D^2(u-v)/2D_0^2} \text{----------------------------------} 8$$

where D_0 is the cutoff frequency. When $D(u, v) = D_0$, the GLPF is down to 0.607 of its maximum value.

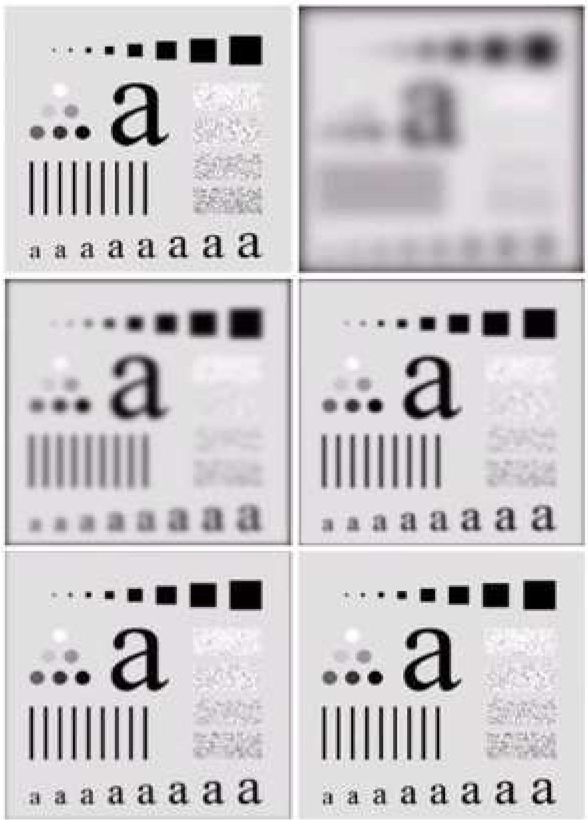
Figure 11 [1]shows the effect of Gaussian low pass filter



a) b) c)

Figure 11:a) Perspective plot of a GLPF transfer functions. b) Filter displayed as an image. c) Filter radial cross sections for various values of D_0

The results of filtering using GLPFs with cutoff frequencies at the different radii is shown in Figure 12.[1]



a b

c d

e f

Figure 12: a) Original image b) -f) Results of filtering using GLPF's with cutoff frequencies at the radii equal to 10, 30, 60, 160 and 460 respectively.

From Figure 12 we can observe that GLPF with filter order 2 if compared to BLPF, there is less smoothening. This is because the profile of GLPF is less tight than the profile of BLPF. Also, we can see that GLPF doesn't have any ringing effect. GLPF can be used to bridge small gaps in broken characters by blurring it and hence it is useful in character recognition. GLPF can be used for cosmetic processing prior to printing and publishing.

2.5 Image Sharpening Using Frequency Domain Filters

Image sharpening can be achieved in the frequency domain by high pass filtering, which attenuates the low frequency components without disturbing high frequency information in the Fourier transform. As edges and other abrupt changes in intensities are associated with high frequency components, so to enhance and sharpen significant details, high pass filters are used. A highpass filter can be obtained from any low pass filter using the equation

$$H_{HP}(u, v) = 1 - H_{LP}(u, v) \text{-------------------------------} 9$$

where $H_{LP}(u, v)$ is the transfer function of the low pass filter.

2.5.1 Ideal Highpass Filters

A 2D ideal highpass filter (IHPF) is defined as

$$H(u, v) = \begin{cases} 0, & if\ D(u, v) \leq D_0 \\ 1, & if\ D(u, v) > D_0 \end{cases} \text{----------------------} 10$$

where D_0 is the cutoff frequency and D(u, v) is given by Eqn 6). Figure 13 [1] shows the perspective plot , image representation and cross section of a typical ideal highpass filter.

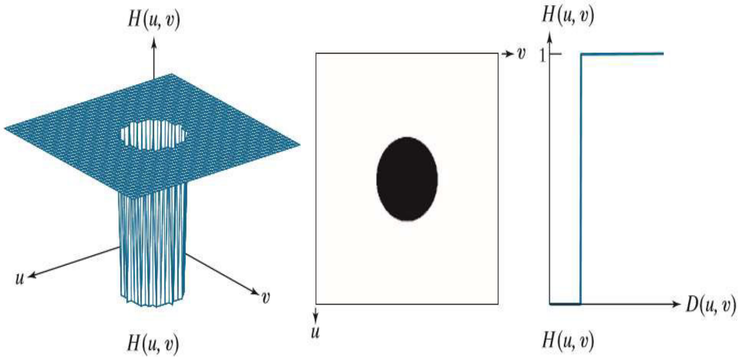


Figure 13: Perspective plot, image representation and cross section of typical ideal high pass filter.

The IHPF sets to zero all frequencies inside the circle of radius D_0 while passing, without attenuation , all frequencies outside the circle. The IHPF is not physically realizable.

Figure 14 [1]shows the results of highpass filtering image

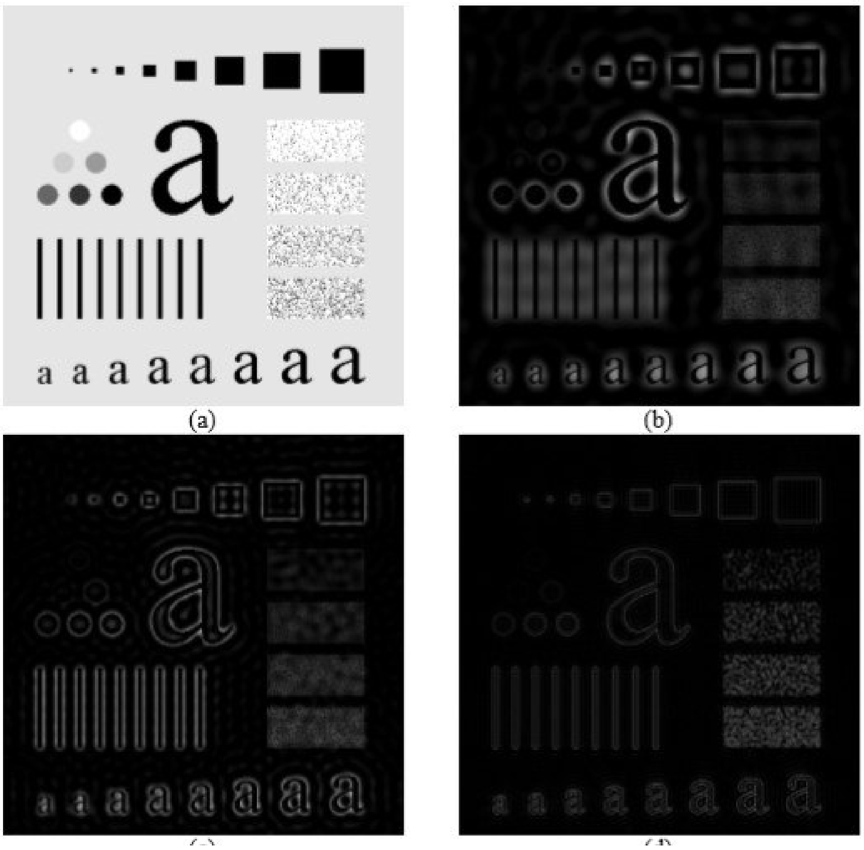


Figure 14 :a) Original image b)-d) Results of highpass filtering the image using an IHPF with $D_0 = 30$, 60 and 160.

From the Figure 14 we can observe the ringing and edge distortion effect and these decreases as cut off frequencies increases.

2.5.2 Butterworth Highpass Filters

A 2-D Butter-worth high-pass filter (BHPF) of order n and cutoff frequency D0 is defined as

$$H(u, v) = \frac{1}{1+[D_0/D(u,v)]^{2n}} \text{ ----------------- } 11$$

where D(u, v) is given by Eqn 6.

Figure 15 [1]shows the perspective plot , image representation and cross section of a typical ideal highpass filter.

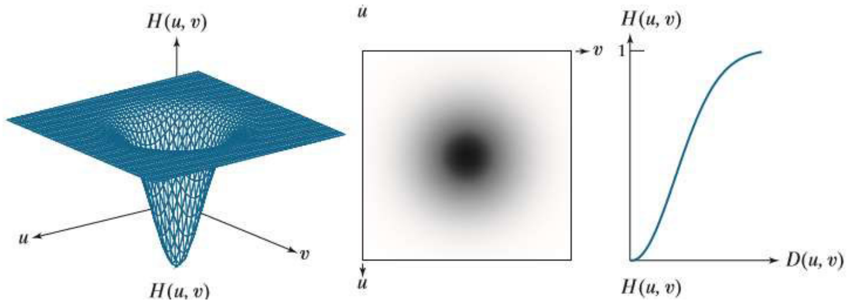


Figure 15: Perspective plot , image representation and cross section of a typical ideal highpass filter.

Figure 16[1] displays results of high pass filtering the image.

Figure 15: Results of highpass filtering the image using a BHPF of order 2 with $D_0 = 30,60$ and 160

From the images we can observe that the boundaries are much less distorted then IHPF even for the smallest value of cutoff frequency. The transition into higher values of cutoff frequencies is much smoother with the BHPF.

2.5.3 Gaussian Highpass Filter

The Gaussian Highpass Filter (GHPF) with cutoff frequency locus at a distance $D_0$from the center of the frequency rectangle is given by

$$H(u, v) = 1 - e^{-D^2(u,v)/\,2D_0^2} \text{----------------------------------}12$$

Where D(u, v) is given by Eqn 6

Figure 16 [1]displays the perspective plot, image and cross section of the GHPF function.

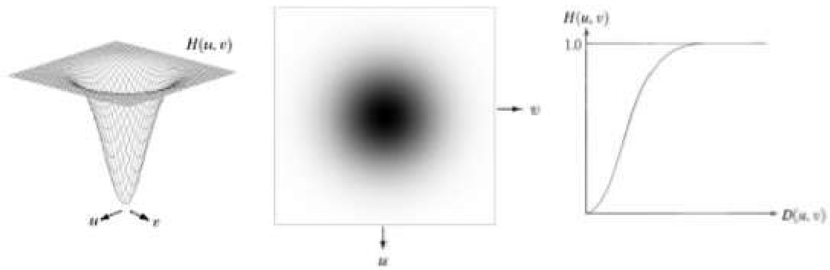


Figure 16: Perspective plot, image and cross section of GHPF

The results of passing an image through a Gaussian Highpass Filter with different cutoff frequencies is demonstrated in Figure 17 [1]

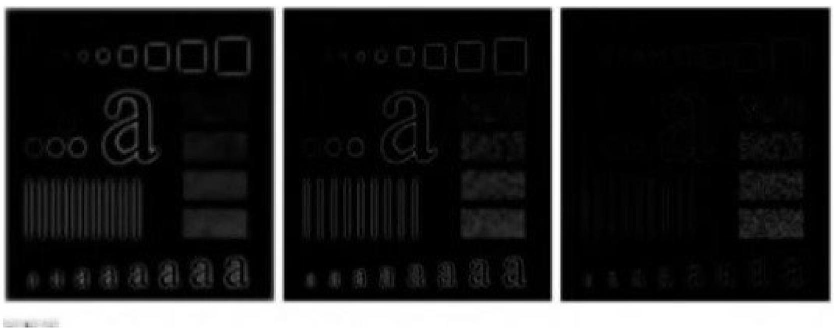


Figure 17: Results of highpass filtering the image using a GHPF with D_0 = 30,60 and 160 respectively.

We can observe from Figure 17 that there is no ringing effect and the results obtained are much smoother than those obtained by IHPF. Also, there is less edge distortion in GHPF.

2.6 Homomorphic Filtering

Homomorphic filtering is a generalized technique for signal and image processing involving a non-linear mapping to a different domain in which linear filter techniques are applied, followed by mapping back to the original domain. It is also used in image enhancement. It simultaneously normalizes the brightness across an image and increase contrast. Homomorphic filtering can be used for improving the appearance of a grayscale image by simultaneously intensity range compression and contrast enhancement.

An image f(x, y) can be expressed as the product of it's illumination , i(x, y) and reflectance r(x, y). The two functions combine as a product to form f(x, y):

$f(x, y) = i(x, y), r (x, y)$

such that $0 < i (x, y) < \infty$ and $0 < r(x, y) < 1$

The fourier transform of the product of two functions is not seperable

$\Im [f(x, y)] \neq \Im [i(x, y)] \Im [r(x, y)]$ -------------------------------------13

Suppose if we define

$z(x,y) = \ln f(x,y) = \ln f(x, y)$

$= \ln i(x,y) + \ln r(x,y)$-------------------------------------14

Then,

Suppose if we define $z(x,y) = \ln f(x,y)$

$= \ln i(x,y) + \ln r(x,y)$---------------------15

Then,

$$\Im\{z(x, y)\} = \Im\{\ln f(x, y)\}$$

$$= \Im\{\ln i(x, y)\} + \Im\{\ln r(x, y)\} \text{------------------} 16$$

Or

$$Z(u, v) = F_i(u, v) + F_r(u,v) \text{--------------------------------------} 17$$

Where $F_i(u, v)$ and $F_r(u, v)$ are the Fourier transforms of $\ln i(x, y)$ and $\ln r(x, y)$ respectively.

Now, if we process $Z(u, v)$ using a filter function $H(u, v)$ then

$$S(u, v) = H(u, v)Z(u,v)$$

$$= H(u, v) F_i(u, v) + H(u, v) F_r(u, v) \text{-------------------} 18$$

The filtered image in the spatial domain is

$$s(x, y) = \Im^{-1}\{S(u, v)\}$$

$$= \Im^{-1}\{H(u, v)F_i(u, v)\} + \Im^{-1}\{H(u, v)F_r(u, v)\} \text{----} 19$$

By defining,

$$i'(x, y) = \Im^{-1}\{H(u, v)F_i(u, v)\} \text{--------------------} 20$$

and

$$r'(x, y) = \Im^{-1}\{H(u, v)F_r(u, v)\} \text{-------------------} 21$$

So, now we can express Eqn 19 as

$$s(x, y) = i'(x, y) + r'(x, y) \text{-----------------------} 22$$

Therefore, the enhanced image $g(x, y)$ will be

$$g(x, y) = e^{s(x, y)}$$

$$= e^{i'(x, y)} e^{r'(x, y)}$$

$$= i_0(x, y) r_0(x, y)$$

where

$$i_0(x, y) = e^{i'(x, y)}$$

and

$r_0(x, y) = e^{r'(x, y)}$

are the illumination and reflectance components of the output image.

The filtering process described above can be summarized as in Figure 18.

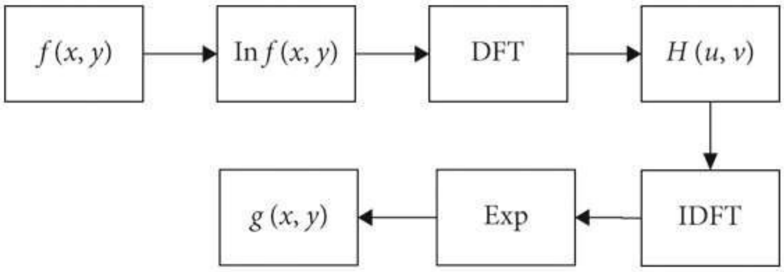


Figure 18: Summary of steps in homomorphic filtering

Imgsrc:https://www.researchgate.net/publication/347236768/figure/fig2/AS:1084168708587569@1635497371752/Basic-steps-of-homomorphic-filtering.jpg

The illuminance component of an image generally is characterized by slow spatial variations. Reflectance components tend to vary abruptly at the junctions of dissimilar objects. Control can be gained over the illuminance and reflectance components with a homomorphic filter. This control requires specification of a filter function H(u, v) that affects the low and high frequency components of the Fourier Transform in different controllable ways. The shape of the function in Figure 19 can be approximated using the basic form of a highpass filter. For example, using a slightly modified form of the Gaussian highpass filter yields the function

$H(u, v) = (\gamma_H - \gamma_L)\ 1 - e^{-c[D^2(u,v)/\ D_0^2}\ ]$---------------------------23

Where D(u, v) is defined in Eqn 6 and the constant c controls the sharpness of the slope of the function as it transitions between γ_L and γ_H.

The effect of homomorphic filtering results in sharper image, increase in contrast and increase in dynamic range compression.

Check Your Progress

State True or False

1.The simplest filter that we can construct is a filter H(u, v) that is 0 at the center of the transform and 1 elsewhere.

2. A low pass filter H(u, v) is a filter that passes high-frequency components while attenuating low-frequency components.

3. A Gaussian low pass filter eliminates high frequency (sharp) features and the practical effect on the image is the blurr effect thus losing the details.

4.The Butterworth filter has a parameter filter order which determines how the filter is going to behave.

5.In ILPF blurring effect increases as cutoff frequency increases.

6.A BLPF of order 2 has no ringing in the spatial domain.

7. GLPF can be used for cosmetic processing prior to printing and publishing.

8. The effect of homomorphic filtering results in blurred image.

2.7 Summing Up

1.Filtering in frequency domain is a powerful technique in image processing that involves modifying the Fourier Transform of an image to get a processed image by computing the inverse transform of that image.

2.The simplest filter that we can construct is a filter H(u, v) that is 0 at the center of the transform and 1 elsewhere.

3.A low pass filter H(u, v) is a filter that passes low-frequency components while attenuating high-frequency components.

4.A high pass filter H(u, v) is a filter that passes high frequency components while attenuating low-frequency components.

5.A Gaussian low pass filter eliminates high frequency (sharp) features and the practical effect on the image is the blurr effect thus losing the details.

6.Smoothing or blurring in Frequency domain is achieved by high frequency attenuation that is by low pass filtering.

7.There are mainly three types of low pass filters are considered: ideal, Butterworth and Gaussian.

8.These three low pass filters cover the range from very sharp (ideal) to very smooth (Gaussian) filtering.

9.Image sharpening can be achieved in the frequency domain by high pass filtering, which attenuates the low frequency components without disturbing high frequency information in the Fourier transform.

10. Homomorphic filtering can be used for improving the appearance of a grayscale image by simultaneously intensity range compression and contrast enhancement.

2.8 Answers to Check Your Progress

1. True	2. False	3. True
4. True	5. False	6.False
7. True	8. False	

2.9 Possible Questions

1. What is the basic filtering equation of a digital image?
2. What is the difference between a lowpass filter and a highpass filter?
3. What is a Gaussian lowpass filter?
4. Write the steps for filtering in the frequency domain.
5. What is image smoothening?
6. What are the three main types of filters used in image smoothing ? Explain with the help of an example.
7. What are the effects of a ILPF on an image?
8. What is a butterworth low pass filter?
9. How does BLPF differ from ILPF?
10. What is image sharpening?
11. What are the different filters used in image sharpening? Explain with the help of examples.
12. What is homomorphic filtering? What are the effects of homomorphic filtering?

2.10 References and Suggested Readings

Gonzalez, R. C. (2009). *Digital image processing*. Pearson education India.

*****

UNIT: 3
IMAGE RESTORATION

Unit Structure:

3.1 Introduction

Image restoration is a critical field in image processing that aims to recover an original, high-quality image from its degraded version. This degradation can result from various factors, including noise, blurring, and other distortions introduced during image acquisition or transmission. The primary goal of image restoration is to enhance the visual quality of an image and restore it as closely as possible to its original state, thereby improving its utility for various applications.

Images serve as vital representations of three-dimensional scenes, often used in numerous fields such as astronomy, medicine, and forensics. However, the captured images frequently suffer from imperfections due to noise, blurring, or other distortions, leading to a compromised depiction of the scene. Image restoration techniques address these issues by applying mathematical and computational methods to improve image quality, enabling better analysis, interpretation, and decision-making.

3.2 Objectives

After completion of this unit, you will be able to-

- *understand* the concept of image restoration,
- *explore* applications of image restoration,
- *understand* noise and noise models,
- *analyze* image degradation,
- *implement* restoration techniques.

3.3 Image Restoration

An image is acquired in order to obtain a two-dimensional (2-D) representation of a three-dimensional (3-D) scene. Unfortunately, many images represent scenes in an unsatisfactory manner. A recorded image often represents a degraded version of the ideal 2-D mapping of the scene. Image restoration addresses the problem of unsatisfactory scene representation. The goal of image restoration is to manipulate an image in such a way that it will in some sense more closely depict the scene that it represents.

Restoration techniques is useful in—

1. **Astronomy**

   Telescopes and cameras have imperfections that can introduce artifacts or blurring. Image restoration algorithms, such as deconvolution, can compensate for these instrumental effects, yielding sharper images.

2. **Medicine**

   Image restoration in the field of medicine enhances the quality of medical images, facilitating better diagnosis, treatment planning, and patient care. Techniques such as histogram equalization and contrast-limited adaptive histogram equalization (CLAHE) are used to improve the contrast of medical images, making it easier to distinguish between different tissues and structures.

3. **Forensics**

   Enhancing images from crime scenes to reveal hidden details and provide clearer evidence. Improving the quality of images from CCTV cameras to identify suspects, vehicles, and other critical details.

3.4 Noise Model

Noise is an unwanted variation in pixel values that do not originate from the scene being captured. Noise produces undesirable effects such as artifacts, unrealistic edges, unseen lines, corners, blurred objects and disturbs background scenes. To reduce these undesirable effects, prior learning of noise models is essential for further processing. Signal-to-Noise Ratio is the measure of the level of the desired signal compared to the level of background noise. A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise. Higher SNR indicated a clearer image.

3.5 Types of Noise and the Methods Used in Noise Modeling:

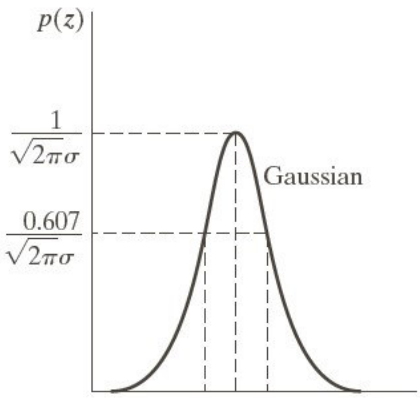
3.5.1 Gaussian Noise Model

Gaussian noise is statistical noise having probability distribution function (PDF) equal to that of the normal distribution, which is also known as the Gaussian distribution. Gaussian noise model is also called normal noise model Gaussian noise is caused by natural sources such as thermal vibration of atoms and discrete nature of radiation of warm objects.

Gaussian noise generally disturbs the gray values in digital images. The probability density function of a Gaussian random variable is given by:

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

Where x = gray level, σ = standard deviation and μ = mean value. Generally Gaussian noise mathematical model represents the correct approximation of real-world scenarios.



In this noise model, the mean value is zero, variance is 0.1 and 256 gray levels in terms of its probability distribution function.

Due to this equal randomness the normalized Gaussian noise curve look like in bell shaped. The PDF of this noise model shows that 70% to 90% noisy pixel values of degraded image in between μ-σ and μ+σ.

3.5.2 Impulse Valued Noise (Salt and Pepper Noise)

This is also called data drop noise because statistically its drop the original data values. Salt-and-pepper noise is a type of impulse noise where the original value of the pixels is lost and the image pixel values are replaced by corrupted pixel values either maximum 'or' minimum pixel value i.e., 255 'or' 0 respectively, if number of bits are 8 for transmission. These pixels are referred to as corrupted or noisy pixels whose origin can be due to, for instance, by malfunctioning pixels in camera sensors, faulty memory locations in hardware or transmissions in a noise channel.

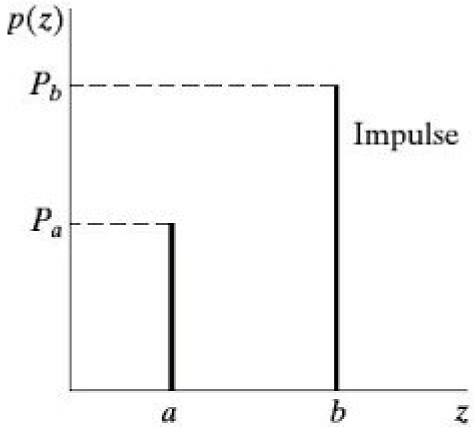
Let us consider 3x3 image matrices. Suppose the central value of matrices is corrupted by Pepper noise. Therefore, this central value i.e., 212 and is replaced by value zero.

In this connection, we can say that, this noise inserted is dead pixels either dark or bright. So, in a salt and pepper noise, progressively dark pixel values are present in bright region and vice versa

254	207	210
97	212 ◄	32
62	106	20

254	207	210
97	0	32
62	106	20

Inserted dead pixel in the picture is due to errors in analog to digital conversion and errors in bit transmission.



P(z) = P_a for z = a

P_b for z = b

0 otherwise

The figure above shows the PDF of Salt and Pepper noise, if mean is zero and variance is 0.05. Here we will meet two spike one is for bright region (where gray level is less) called 'region a' and another one is dark region (where gray level is large) called 'region b', we have clearly seen here the PDF values are minimum and maximum in 'region a' and 'region b', respectively.

3.5.3 Brownian Noise (Fractal Noise)

Brownian noise, often referred to as fractal noise, it is a type of noise with a power density that decreases with increasing frequency, proportional to the inverse square of the frequency. Brownian noise is caused by Brownian motion, it is seen due to the random movement of suspended particles in the fluid. Brownian noise is referred to as fractal noise. Fractal noise is caused by natural process. It is different from Gaussian process.

A fractional Brownian motion is mathematically represented as a zero mean Gaussian process (BH) which is showing in the following equation respectively.

B_H (0) = 0 and expected value of fractional Brownian motion is $E\{|B_H(t) - |B_H(t - \Delta|)^2\} = \sigma^2 |\Delta|^{2H}$

3.5.4 Periodic Noise

Periodic noise is a common issue in digital images, arising from electrical interference during the image capture process. When an image is affected by periodic noise, it appears as if a repeating pattern has been superimposed on the original image. In the frequency domain, this type of noise manifests as discrete spikes. If you encounter periodic noise in your images, consider using frequency domain filtering techniques to reduce it effectively. These methods isolate the frequencies associated with the noise and suppress them using band-reject filters.

3.5.5 White Noise

White noise is a random signal that contains equal intensity across different frequencies, resulting in a constant power spectral density. White noise is a combination of all imaginable tones a human can hear—like static from an untuned radio or television. White noise spans the entire audible spectrum, making it a type of broadband noise.

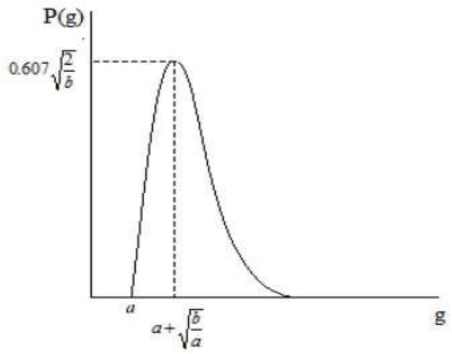
The total power of a white noise signal is theoretically infinite because white noise is defined to have a flat power spectral density over an infinite bandwidth.

3.5.6 Rayleigh Noise

The Rayleigh distribution is a continuous probability distribution for positive valued random variables. It is often observed when the magnitude of a vector is related to its directional components PDF of Rayleigh noise is defined by

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

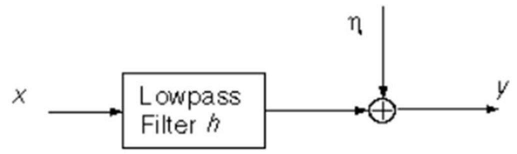
The Mean & Variance of distribution is given by$\mu = a + \left(\sqrt{\frac{(\pi b)}{4}}\right)$and $\sigma^2 = b\,\frac{(4-\pi)}{4}$ respectively.



3.6 Image Degradation

Image degradation is the process by which an image's quality is diminished or compromised. This can happen for a number of causes, including noise, blur, or compression, and can negatively affect the image's appearance and readability. Noise is one of the most frequent causes of image degradation. Noise can affect an image's clarity and detail by causing random variations in the intensity or color of the pixels. Image restoration methods like denoising, deblurring, or decompression may be required to address image degradation in an effort to improve the image's quality. These methods can be used to enhance the image's visibility and readability, but they also run the risk of adding more artifacts or distortions, which could degrade the quality of the restored image.

The block diagram for our general degradation model is



where *g* is the corrupted image obtained by passing the original image *f* through a low pass filter (blurring function) *b* and adding noise to it.

We present four different ways of restoring the image.

3.6.1 Inverse Filtering

Inverse filtering is a technique used in signal processing and image processing to recover an original signal or image from a degraded or distorted version of it. It's based on the idea of reversing the effects of a known filter or degradation process.

The idea of inverse filtering is to obtain an estimate of the original image F (u, v) by dividing the degraded image G(u, v) by the degradation function H(u, v):

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

In the spatial domain, this corresponds to deconvolution. However, this approach is sensitive to noise because if H (u, v) has small values, the division can amplify the noise N (u, v) significantly.

Implementation Steps

1. **Fourier Transform**: Compute the Fourier transform of the degraded image g(x, y) to get G(u, v).
2. **Division**: Divide G(u, v) by the Fourier transform of the PSF, H(u, v), to obtain the estimated Fourier transform of the original image $\hat{F}$ (u, v)
3. **Inverse Fourier Transform**: Compute the inverse Fourier transform of $\hat{F}$ (u, v) to get the restored image $\hat{f}(x, y)$.

3.6.2. Weiner Filtering

Wiener filtering is an advanced technique used in digital image processing for image restoration. It is designed to handle the challenges of inverse filtering, especially in the presence of noise. The Wiener filter aims to minimize the mean square error between the estimated and the true image by considering both the degradation function and the statistical properties of the noise and the original image.

The Wiener filter uses a probabilistic approach to find the best estimate of the original image F(u,v) given the observed image G(u, v), the degradation function H(u, v), and the noise characteristics. The filter is defined in the frequency domain as:

$$F(u,v) = H \quad \frac{(u,v)}{|H(u,v)^2|} + \frac{K}{|K(u,v)^2|} G(u,v)$$

where,

- $\hat{F}$ (u, v) is the estimated Fourier transform of the original image.
- G (u, v) is the Fourier transform of the degraded image.

- H (u, v) is the Fourier transform of the degradation function (PSF).
- H* (u, v) is the complex conjugate of H (u, v).
- K is a constant related to the noise-to-signal power ratio.

The Wiener filter balances between inverse filtering and noise suppression. When the signal-to-noise ratio is high, the filter behaves like an inverse filter. When the signal-to-noise ratio is low, it suppresses the noise.

Implementation Steps

1. Fourier Transform: Compute the Fourier transform of the degraded image *g(x, y)* to get *G(u, v).*

2. Weiner Filter: Apply Wiener Filter formula to compute $\hat{F}(u, v)$.

3. Inverse Fourier Transform: Compute the inverse Fourier transform of $\hat{F}(u, v)$to get the restored image $\hat{f}(x, y)$.

Check Your Progress

Multiple Choice Questions

1. What is the primary goal of image restoration?

a) To segment objects within the image.
b) To enhance the visual quality by removing noise and distortions.
c) To classify the image into different categories.
d) To compress the image for storage efficiency.

2. Which of the following is a common noise model used in image restoration?

a) Poisson Noise

b) Gaussian Noise

c) Salt-and-Pepper Noise

d) All of the above

3. What type of noise is typically characterized by random bright and dark pixels scattered across an image?

a) Gaussian Noise

b) Salt-and-Pepper Noise

c) Speckle Noise

d) Poisson Noise

4. Which mathematical function is typically used to describe Gaussian noise?

a) Exponential Distribution.

b) Uniform Distribution.

c) Normal Distribution.

d) Binomial Distribution.

5. Which of the following factors commonly causes image degradation?

a) Poor lighting conditions.

b) Motion blur.

c) Low resolution.

d) All of the above.

3.7 Summing Up

1. Image restoration is a vital process in image processing aimed at recovering a high-quality image from its degraded version.

2. Degradation can occur due to noise, blurring, and other distortions during image acquisition or transmission.

3.The primary objective of image restoration is to enhance image quality, making it a crucial technique in fields such as astronomy, medicine, and forensics.

4.A significant part of image restoration involves understanding noise and noise models. 5.Different types of noise, including Gaussian, salt-and-pepper, Brownian, periodic, white, and Rayleigh noise, are explored, along with their probability distribution functions and characteristics.

6. Recognizing these noise types is essential for selecting appropriate restoration techniques to address each specific type of noise effectively.

7. Understanding the principles and techniques of different image degradation techniques allows significant improvements in image quality, making degraded images more useful and reliable for various practical applications.

3.8 Answers to Check Your Progress

Ans. 1-(b), 2-(d),3-(b),4-(c),5-(d)

3.9 Possible Questions

1. Define Image restoration?
2. State how Brownian noise is caused?
3. Define Inverse Filtering?
4. Name three noise model?
5. Draw the block diagram of general degradation model.
6. Discuss the different types of noise modeling.
7. State the various application where Image Restoration can be implemented.
8. Briefly describe image degradation.
9. Write a short note on Inverse Filtering and Weiner Filtering.

3.10 References and Suggestive Readings

1. Gonzalez, Rafael C. “Digital image processing.” Prentice Hall, 2008.

2. Thiruvikraman, P. K. "Image Restoration." Published November 2019. Copyright © IOP Publishing Ltd 2020, pp. 6-1 to 6-20.

3. “Image Restoration.” CRC Press eBooks, 2018, https://doi.org/10.1201/b12693.

*****

UNIT: 4
MORPHOLOGICAL IMAGE PROCESSING

Unit Structure:

4.1 Introduction

Morphology is the technique used to manipulate the appearance and shape of an object. Mathematical morphology is a collection of operations that can be applied to images to manipulate the shape and structure of the image. The operations of mathematical morphology were originally defined as set operations and have been shown to be useful for processing sets of 2D points in general and images in particular. There are many applications of morphological operations, such as extracting the edges of an image, boundary detection of an image, enlarging an image, shrinking an image, sharpening an image, blurring an image, filtering an image, skelatonizing an image, etc. The basic morphological operations, discussed in this unit are binarization, set operation, erosion, dilation, opening and closing, Hit or Miss Transformation, boundary detection. The morphological operations can be performed on binary, grayscale, and colour images, but in this unit we will discuss only morphological operations on binary images. A binary image has only two gray levels mainly black and white.

4.2 Objectives

This unit is an attempt to analyse the ideas of manipulating shape and structure of binary image. After going through this unit you will be able to-

- *explain* binary image manipulation,
- *apply* detection of boundary and template matching,
- *apply* the process of enlarging and shrinking an image,
- *explain* the process of noise reduction.

4.3 Introduction to Binarization

Due to the advancement of technology, most digital image capturing devices are colour sensitive. The meaning is that when we capture an image with the help of a camera, we mostly capture a colour image. Though clolour images have good visual quality, they occupy a larger amount of storage. In many image processing applications, colour is not essential. Colour images occupy more storage space, and it takes more computing time. Many applications can be performed with the help of a binary image. So, in many situations, we need to convert a colour image to a binary image. Colour images are converted to grayscale images, then converted to binary images. The process of converting a grayscale image to a black-and-white or binary image is called binarisation. A grayscale image contains a pixel intensity range of 0 to 255 levels, as each pixel stores 8 bits. Binarisation can be performed using global thresholding techniques. Global thresholding sets all pixels above a certain intensity value to white, and the rest of the pixels to black in the image. It is very important to decide the appropriate threshold value to binaries the image, though it is difficult to decide a global value that is suitable for all images. The common technique is global thresholding. Global thresholding sets all pixels above a defined threshold value to white (0) and the rest of the pixels to black (1) in the image. Figure 4.1 shows an example of grayscale and binary images. Figure 4.1(b) has been created using Figure 4.1(a), taking the global threshold value 128. Each pixel has a value of either 0 or 1. The black pixel (digit 1) refers to the foreground of the image, and the white pixel (digit 0) refers to the background of the image.

a) Grayscale image

b) Binary image

Figure 4.1: Image examples

STOP TO CONSIDER

Google colab code to read a colour image and convert it to grayscale then binary.

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow

# Load the color image

image = cv2.imread('/content/gucodl.jpg')

# Convert the image to grayscale

gray_image = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)

# Apply a binary threshold to the grayscale
image

_, binary_image = cv2.threshold(gray_image,
128, 255, cv2.THRESH_BINARY)

# Display the original and binary images

cv2_imshow(gray_image)    # Gray Image

cv2_imshow(binary_image)  # Binary Image

# Save the binary image

cv2.imwrite('/content/binary_image.jpg',
binary_image)
```

SAQ

What is image binarisation? Why it is required?

4.4 Applications

Mathematical morphology is a tool for finding image components that are useful for representation and description. Morphological image processing play an important role in image preprocessing. In real life applications an image need to manipulate before fed in to the model. This include mainly image enhancement, extracting the edges of an image, boundary detection of an image, enlarging an image, shrinking an image, sharpening an image, blurring an image, filtering an image, skelatonizing an image, etc.

4.5 Basic set operation

Mathematical morphology is based on basic set theory. This section describes the basic set theory used in image processing.

4.5.1 Union

The union of two binary images A and B is written as AUB. AUB is the set whose elements are either the elements of A or B or both. Mathematically, it is written as

$$AUB = \text{def}\{x | x \in A \; or \; x \in B\}$$

The union of images A and B has been shown in Figure 4.2

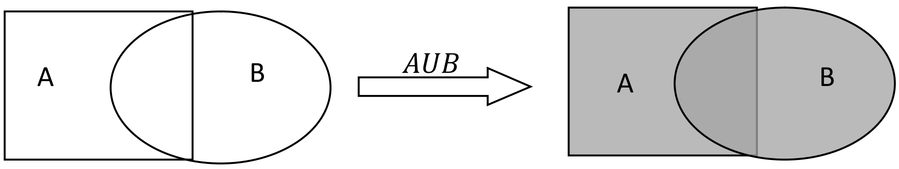


Figure 4.2: Union of two images A and B

4.5.2 Intersection

The intersection of two binary images A and B is written as A∩B. A∩B is the set whose elements are common to both images A and B. Mathematically, it is written as

$$A \cap B = \text{def}\{x | x \in A \; and \; x \in B\}$$

The intersection of images A and B has been shown in Figure 4.3

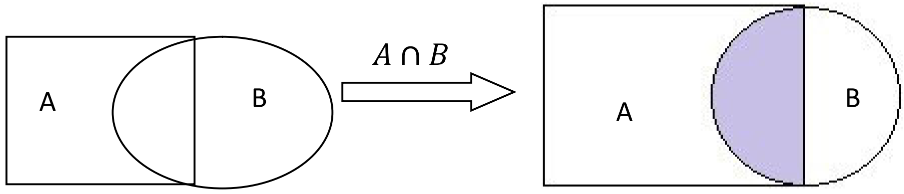


Figure 4.3: Intersection of two images A and B

4.5.3 Difference

The difference of two binary images A and B is written as A – B. A – B is the set whose elements are in image A and not in image B. Mathematically, it is written as

$$A - B = \text{def}\{x | x \in A \; and \; x \notin B\}$$

The difference of images A and B has been shown in Figure 4.4

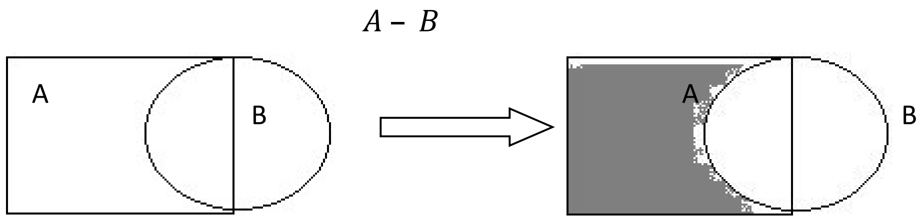


Figure 4.4: Difference of two images A and B

4.5.4 Complement

The complement of image A is the set consisting of everything but not in image. It is written as A^c. The definition of complement operation is represented by

$$A^c = \text{def}\{x | x \notin A$$

The result of complement operation is shown in Figure 4.5

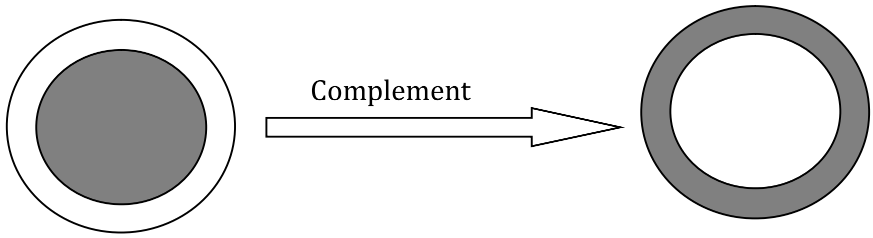


Figure 4.5: Complement of two images A and B

4.5.5 Reflection

Reflection of an image A can be viewed as mirror of that image. It is defined as

$$A_1 = \text{def}\{x | x = -a \; for \; a \in A$$

The result of reflection operation has been depicted in Figure 4.6

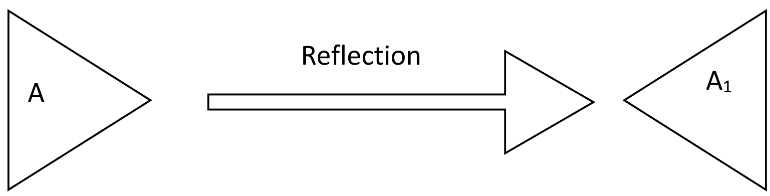


Figure 4.6: Reflection of image A

4.5.6 Translation

The translation operation shifts the origin of image A from point (x,y) to some other point (x_1, y_1). Translation operation of image A is defined as

$$A_1 = \text{def}\{x | x = a + z \; for \; a \in A$$

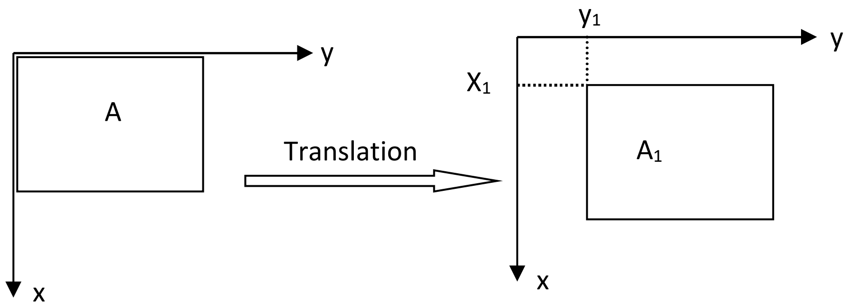


Figure 4.7: Translation of image A

The result of translation operation has been depicted in Figure 4.7

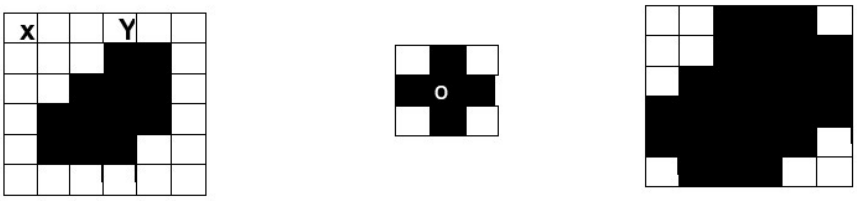
4.6 Dilation

Dilation is one of the two basic operators in the area of mathematical morphology. It is typically applied to binary and grayscale images. The basic effect of this operator on a binary image is that it gradually increases the boundaries of regions of the image. The way the image increases is determined by the structuring element. The structuring element is small in size, normally 3x3, 5x5, 7x7, etc. The dilation operation is performed using the structuring element shifted from left to right and from top to bottom. At each shift, the process will look for any overlapping similar pixels between the structuring element and the binary image. If there is any overlapping, the pixel under the centre position of the structuring element will be turned to 1 or black. Let us define X as the reference image and B as the structuring element. The dilation operation is defined by

$$X \oplus B = \{z | [\ ((\hat{B})_Z \cap X\ ] \subseteq X \qquad (4.1)$$

Where $\hat{B}$ is the image B rotated about the origin. Equation 4.1 states that when the image X is dilated by the structuring element B, the outcome element z will be that there will be at least one element in B that intersects with an element in X.

Example



(a) Input image (b) Structuring element (c) Dilated image

Figure 4.8: Dilation operation using 3x3 structuring element

Figure 4.8 shows the dilation operation. The black square represents 1, and the white square represents 0. Initially, the centre of the structuring element is placed at position X in Figure 13.10(a). At this point, there is no overlapping between any black square of B and the black square of X. So, at position X, the square will remain white. The structuring element will then be shifted by one pixel to the to the right. The shifting process will continue until the last pixel of the row. At position Y, one black square (pixel) of B is overlapping or intersecting with the black square of X. Thus, at position Y, the square will be changed to black. The structuring element B is shifted from left to right and from top to bottom on the image X to get the dilated image, as shown in Figure 4.8(c).

STOP TO CONSIDER

Google colab code for image dilation

```
# coding: utf-8
# Image dialation using 3x3, 5x5 and 7x7
structuring element
import cv2

import numpy as np

from matplotlib import pyplot as plt

# Reading the image

img=cv2.imread('/home/rusa/Desktop/j.png',0)

#Declaring the structuring element

kernel1 = np.ones((3,3),np.uint8)

kernel2 = np.ones((5,5),np.uint8)

kernel3 = np.ones((7,7),np.uint8)

# Performing dilation operation

dilation1 = cv2.dilate(img, kernel1, iterations =
1)

dilation2 = cv2.dilate(img, kernel2, iterations =
1)

dilation3 = cv2.dilate(img, kernel3, iterations =
1)

# Display the images

plt.subplot(221), plt.imshow(img)

plt.axis("off")

plt.subplot(222), plt.imshow(dilation1)

plt.axis("off")

plt.subplot(223), plt.imshow(dilation2)

plt.axis("off")

plt.subplot(224), plt.imshow(dilation3)

plt.axis("off")

.
```

Output

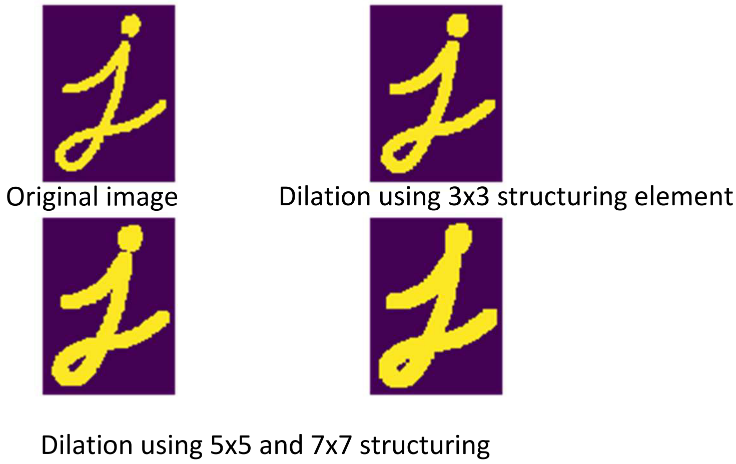


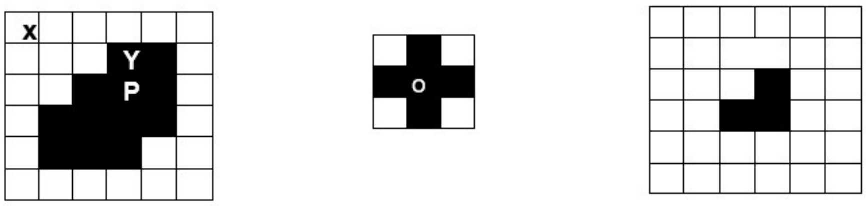
Figure 4.9: Output of dilation operation

4.7 Erosion

Erosion is the reverse process of dilation. Dilation increases the boundary region of an image, whereas erosion shrinks the image. The amount of reduction depends on the size of the structuring element. The structuring element is normally smaller than the image. The smaller the structuring element, the shorter the computation time. Similar to the dilation process, the erosion process shifts the structuring element from left to right and from top to bottom. At the centre position, indicated by the centre of the structuring element, the process will look for whether there is a complete overlap with 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 to white or 0. Mathematically, it may be defined as

$$X \ominus B = \{z \mid (B)_Z \subseteq X\} \qquad (4.2)$$

Where X is the reference binary image and B is the structuring element, Equation 4.2 states that the outcome element z is considered only when the structuring element is a subset of or equal to the binary image X.



a) Input image (b) Structuring element (c) Eroded image

Figure 4.10: Erosion operation using 3x3 structuring element

Figure 4.10 shows the erosion operation. The black square represents 1, and the white square represents 0. Initially, the center of the structuring element is placed at position X in Figure 4.10(a). At this point, there is no overlapping between any black square of B and the black square of X. So, the corresponding X position in the output image, the square will remain white. The structuring element will then be shifted by one pixel to the right in the original image. The shifting process will continue until the last pixel of the row. At position Y, complete overlapping or intersecting with the black square of X is not found, thus the black square marked with Y will be turned white. The structuring element is then shifted further until its center reaches position P. At position P, overlapping is complete; that is, all the black squares in the structuring element overlap with the black squares in the image. Thus, the center of the structuring element corresponding to the image will be black. The structuring element B is shifted from left to right and from top to bottom on the image X to get the eroded image, as shown in Figure 4.10(c).

STOP TO CONSIDER

Python code for image erosion

```
# coding: utf-8
# Image erosion using 3x3, 5x5 and 7x7 structuring
element
import cv2
import numpy as np
from matplotlib import pyplot as plt
#Reading the image from a folder in binary mode
img=cv2.imread('/home/rusa/Desktop/j.png',0)
#defining the structuring element
kernel1 = np.ones((3,3),np.uint8)
kernel2 = np.ones((5,5),np.uint8)
kernel3 = np.ones((7,7),np.uint8)
# erosion operation
erosion1 = cv2.erode(img, kernel1, iterations = 1)
erosion2 = cv2.erode(img, kernel2, iterations = 1)
erosion3 = cv2.erode(img, kernel3, iterations = 1)
# Display the outputs in four window
plt.subplot(221), plt.imshow(img)
plt.axis("off")
plt.subplot(222), plt.imshow(erosion1)
plt.axis("off")
plt.subplot(223), plt.imshow(erosion2)
plt.axis("off")
plt.subplot(224), plt.imshow(erosion3)
plt.axis("off")
```

Output

Original image Erosion using 3x3 structuring element

Erosion using 5x5 and 7x7 structuring element

4.8 Opening

Opening is based on morphological operations, erosion, and dialation. Opening smoothes the inside of the object conture, breaks narrow strips, and eliminates thin portions of the image. It is done by first applying erosion and then dilation operations to the image. Mathematically, the opening process can be defined as

$$X \circ B = (X \ominus B) \oplus B \qquad (4.3)$$

Where X is the reference binary image and B is the structuring element

4.9 Closing

The closing operation is the reverse of the opening operation. It is a dilation operation followed by an erosion operation. The closing operation fills the small holes and gaps in a single pixel object. It has the same effect as the opening operation in that it smoothes contours and maintains the shapes and sizes of objects. Mathematically, the closing process can be defined as

$$X \bullet B = (X \oplus B) \ominus B \qquad (4.4)$$

4.10 Duality

Duality is the property of binary morphological operation. This property of duality can be defined as follows:

$$X \ominus B \leq (X^c \oplus B)^c \qquad (4.5)$$

$$X \oplus B \leq (X^c \ominus B)^c \qquad (4.6)$$

$$X \circ B \leq (X^c \bullet B)^c \qquad (4.7)$$

$$X \bullet B \leq (X^c \circ B)^c \qquad (4.8)$$

The equations (4.5) and (4.6) state that dilation is the dual of erosion operation. The meaning can be defined as that dilation is the complement operation of erosion and vice versa. Similarly, opening and closing are dual operations. The closing of X corresponds to the opening of the complemented image X^c. The equations (4.6) and (4.7) represents that opening is the complement operation of closing and vice versa.

Check Your Progress

Q1 Write True or false

a. Each pixel of a binary image can take only 0 and 1 value

b. A grayscale image pixel can take only one value from 256 colour intensities irrespective of colour code

c. Size of structuring element can be larger than reference input image

d. Morphological operations are used for colour conversion

e. Set operation can be performed on colour image

f. Dialation operation shrinks the original image

g. Structuring element can be of any shape

4.11 Summing up

- There are basically three types of images, binary, gray scale and colour.
- Binarisation is the process of converting a gray scale image into binary image.
- Mathematical morphology is a tool for finding image components that are useful for representation and description.
- Structuring element is a set of pixels normally in numbers 9, 25, 49 etc.

- There are four basic morphological operations mainly used in image processing i.e. dialation, erosion, opening and closing.
- Dialation operation increases the size of an image
- Erosion operation reduces the size of an image. The increase or shrink of an image depends on the structuring element

4.12 Answer to Check Your Progress

Q. No1

a. True

b. False

c. False

d. False

e. True

f. False

g. False

4.13 Possible Questions

Q1. What is binary image? What are the difference between binary and grayscale image?

Q2 What is mathematical morphology? How morphology is used in image processing

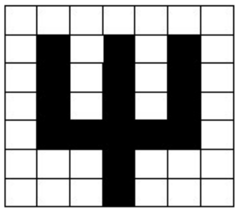
Q3 Explain the different set theory used in image processing.

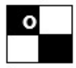
Q4 Explain the different logical operations used in image processing.

Q5 Explain any one standard binary morphological operations used in image processing.

Q6 Using the input image and structuring element as given below, find the dilated version of the input image

Input image Structuring element





4. 14 References and Suggested Readings

1. Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, McGrawHill
2. Digital Image Processing, 2nd ed,S Sridhar, Oxford
3. NPTEL, IITKGP

*****

UNIT: 5
BASIC MORPHOLOGICAL ALGORITHMS

Unit Structure:

5.1 Introduction

In this unit, we will discuss some applications of morphological techniques. Morphological operators are mostly useful in a variety of image processing applications. They can be used to extract the boundary, remove noise, identify components, convex hull, and so on. Some useful applications are explained in this unit. The first operation is boundary extraction. Boundary extraction is a morphological operation that extracts the boundary of an object. Then we will discuss some important morphological operations like convex set, convex hull. and thickening.

5.2 Objectives

This unit is an attempt to explain and implement ideas of some morphological operations on binary images. After going through this unit, you will be able to-

- *extract the boundary of an image,*
- *apply* the convex set property to a binary image,
- *apply* the process of enlarging and shrinking an image,
- *explain* the process of thinning and thickening.

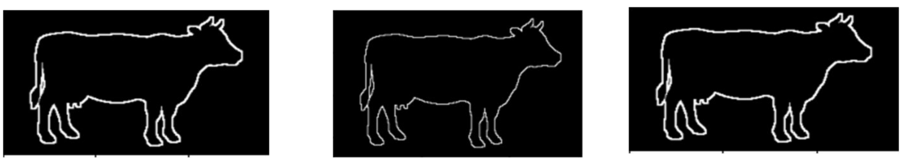
5.3 Boundary Extraction

One of the major applications of morphological operation is finding the boundary. Boundary can play avital role in object identification. The difference between the original image and the eroded image creates a boundary. Let us assume that X is the input image and B is the structuring element. Two types of boundaries can be obtained as follows:

$$Y = X - (X \ominus B)$$

$$Y = (X \oplus B) - X$$

Where Y is the boundary image, $\ominus$ denotes erosion operator, and $\oplus$ denotes dilation operator. The original image, boundary extraction with a 3x3 mask, and boundary extraction with a 5x5 mask have been shown in Figure 5.1.



a. Original Image | b. Boundary detection, 3x3 mask | c. Boundary detection, 5x5 mask

Figure 5.1: Boundary extraction

SAQ

Explain the concept of boundary extraction

STOP TO CONSIDER

Google colab code to read a colour image and detect the boundary using 3x3 and 5x5 mask.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Read the colour image
image = cv2.imread('/content/cow1.png',
cv2.IMREAD_GRAYSCALE)
# Ensure the image is binary
_, binary_image = cv2.threshold(image, 127, 255,
cv2.THRESH_BINARY)
# Create kernels for erosion
kernel = np.ones((3, 3), np.uint8)
kernel1 = np.ones((5, 5), np.uint8)
# Erode the binary image
eroded_image = cv2.erode(binary_image, kernel,
iterations=1)
eroded_image1 = cv2.erode(binary_image, kernel1,
iterations=1)
# Subtract the eroded image from the original image
to get the boundary
boundary_image = cv2.subtract(binary_image,
eroded_image)
boundary_image1 = cv2.subtract(binary_image,
eroded_image1)
# Display the images
plt.figure(figsize=(10, 5))

plt.subplot(1, 3, 1)
plt.title('Original Binary Image')
plt.imshow(binary_image, cmap='gray')

plt.subplot(1, 3, 2)
plt.title('Eroded Image')
plt.imshow(boundary_image, cmap='gray')

plt.subplot(1, 3, 3)
plt.title('Boundary Image')
plt.imshow(boundary_image1, cmap='gray')

plt.show()
```

5.4 Convex Set

If we take any pair of points in the set S and join a straight line connecting these two points, we say that the set S is convex. In such cases, if all the points lying in the straight line also belong to the particular set S, then we say that the set S is convex. On the other hand, if any point on the straight line does not belong to the set S, then we say that set S is not convex. As an example, if we take two point sets S1 and S2 on Figure 5.2(a) and 5.2(b) and consider any two points on both the point sets, then all the points lying on the straight line in Figure 5.2(a) are in set S1, but all the points lying on the straight line in Figure 5.2(b) are not in set S2. So, set S1 is a convex set, and set S2 is not a convex set.

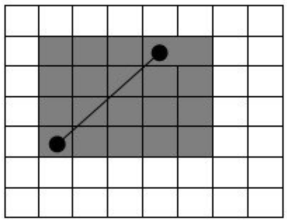


Figure 5.2(a): Point Set S1(convex).

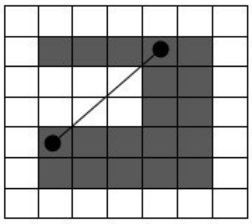


Figure 5.2(b): Point Set S2(not convex)

5.5 Convex Hull

Given a set S, the convex hull of this set S will be the minimal set containing set S, which is convex. The original set S may be convex or may not be convex. The minimal point set that is convex is called the convex hull of set S. If H is the convex hull of set S, then H – S is called convex deficiency, i.e., the set difference between the convex hull and the original point set is convex deficiency. The convex deficiency can be used as one of the descriptors of a given set, which may be useful for high level image understanding purposes. Let us design an algorithm that, given a point set S, we can find out the convex hull of that given set S. Here, instead of using single structuring elements, we use a set of structuring elements. Let us assume B^i is a structuring element where i varies from 1 to 4. We are using four structuring elements for performing the particular operations. The four structuring elements that are used for determining the convex hull are given in Figure 5.3.

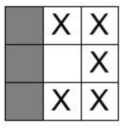


Figure 5.3(a): B^1

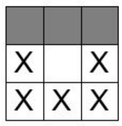


Figure 5.3(a): B^2

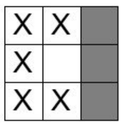


Figure 5.3(a): B^3

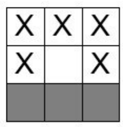


Figure 5.3(a): B^4

Figure 5.3: Structuring element B^1 to B^4

The iterative algorithm for finding out the convex hull will be as follows

$$X_k^i = \left(X_{K-1}^i \circledast B^i\right) \cup \mathrm{A}$$

Here A is the original point set, $\circledast$ is Hit or Miss transform. This iterative algorithm has to be applied for each and every structuring elements. The initial condition is

$$X_0^i = A$$

This iterative algorithm will converge when the following condition meets.

$$X_k^i = X_{k-1}^i$$

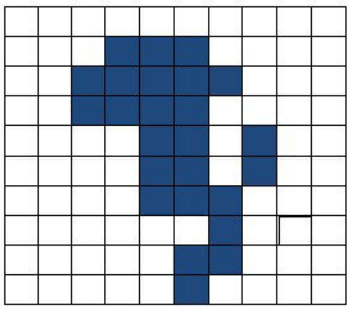
i.e. in two subsequent iterations the outputs does not change. The final output when the algorithm converges it is marked as

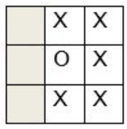
$$X_{conv}^i = D^i$$

Then the convex hull C(A) will be represented as

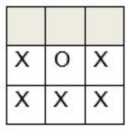
$$C(A) = \bigcup_{i=1}^{4} D_i$$

Example: Find the convex hull for the following image and the structuring elements.

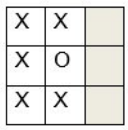




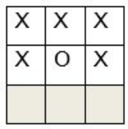
B^1



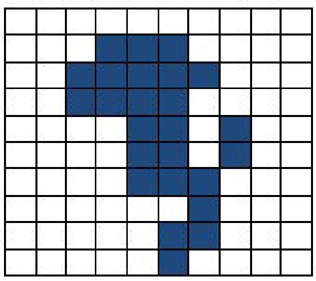
B^2



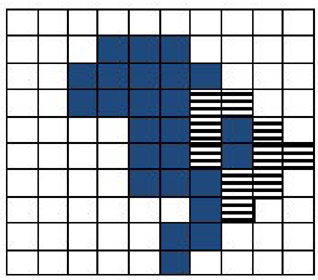
B^3



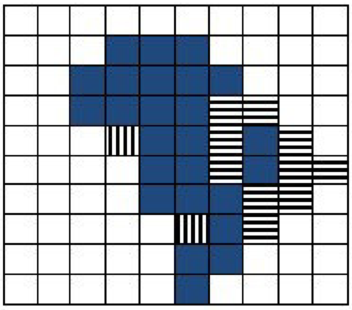
B^4



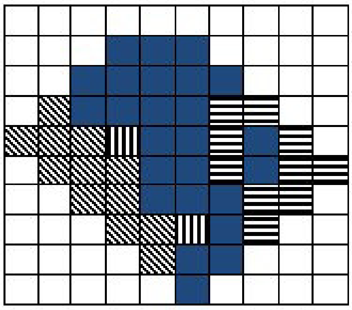
$X_0^1 = A$



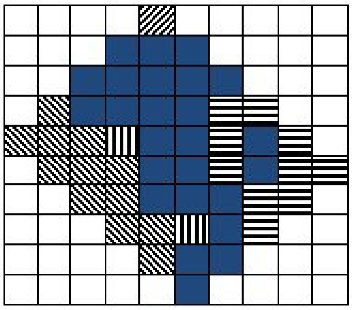
$X_4^1 = D^1$



$X_1^2 = D^2$



$X_5^3 = D^3$



$X_1^4 = D^4$

This is the desired convex hull for the given point set.

5.5 Thinning

Thinning is a morphological operation that is used to find the skeleton of an image. Skeletons identify the structure of the image and are useful to detect or identify objects. The thinning operation of a point set 'A' by a structuring element B is defined as

$$A \otimes B = A - (A \circledast B)$$

$$= \boldsymbol{A} \cap (\boldsymbol{A} \circledast B)^c$$

Here, A represents the original point set, B represents the structuring element, ⊗ represents the thinning operation and ⊛ represent Hit or Miss transform. Like the convex hull operation, here instead of using a single structuring element, a set of structuring elements is used. The set of structuring elements may be defined as

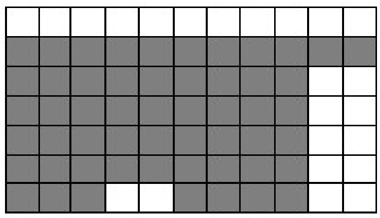
$$\{B\} = \{B^1, B^2, B^3, \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots . B^n\}$$

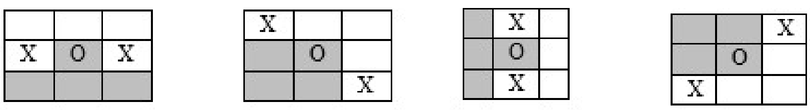
Where, all these structuring elements in this set follow a particular property, i.e., B^i is the rotated version of B^{i-1}. The meaning is that if we rotate the structuring element B^1 then we get structuring element B^2, similarly, if we rotate the structuring element B^2 then we get the structuring element B^3 and so on. Using this set of structuring elements, the thinning operation performs iteratively as follows:

$$A \otimes \{B\} = ((\dots \dots \dots \dots . \left((A \otimes B^1) \otimes B^2\right) \dots \dots \dots . .) \otimes B^n)$$

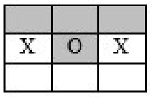
The thinning operation applied iteratively, first thin point set A with structuring element B^1, then thin it with structuring element B^2 and so on and finally thin it with B^n. This process continue until convergence occurs.

Example: Perform thinning operation on the following point set using the following 8 structuring elements.

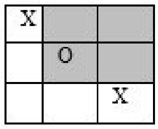




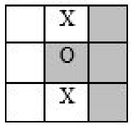
B^1 B^2 B^3 B^4



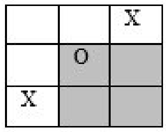
B^5



B^6

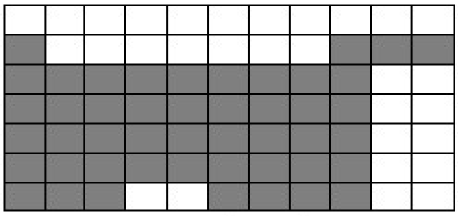


B^7

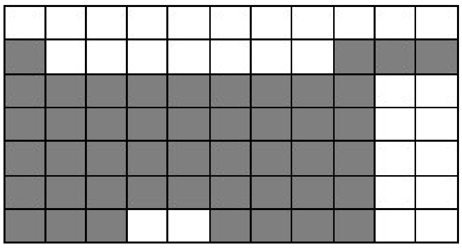


B^8

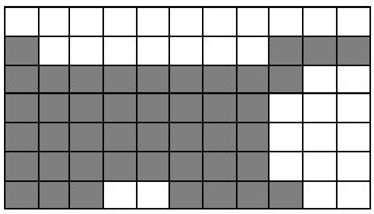
Solution



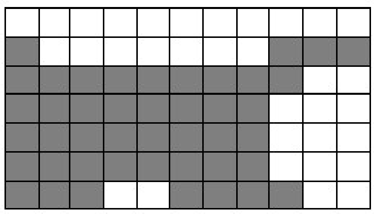
$(A \otimes B^1) = A^1$



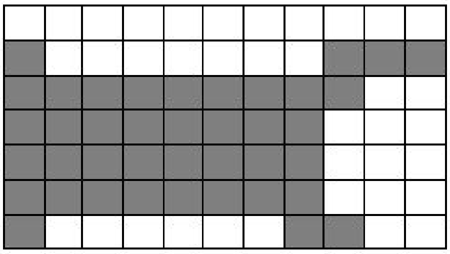
$(A^1 \otimes B^2) = A^2$



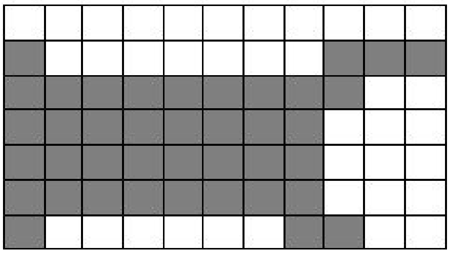
$(A^2 \otimes B^3) = A^3$



$(A^3 \otimes B^4) = A^4$



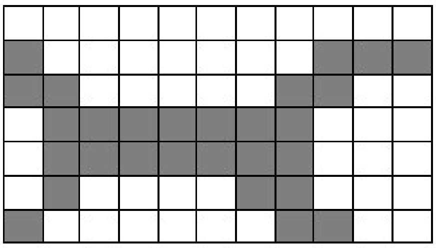
$$(A^4 \otimes B^5) = A^5$$



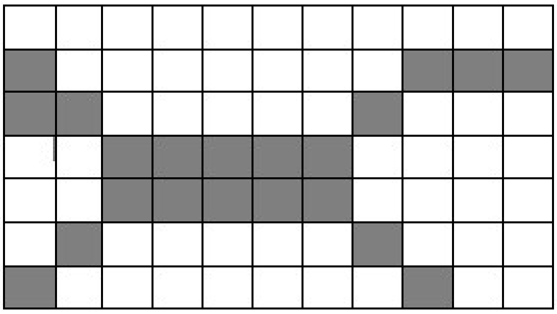
$$(A^5 \otimes B^6) = A^6$$

Here, we thin input image A with structuring element B^1 and we get the output A^1, then we thin A^1 with structuring element B^2 and in this stage there is no change occurring and we get the output A^2 where there is no change, now we thin A^2 with structuring element B^3 and get the output A^3. Now, perform the thin operation considering A^3 with structuring element B^4 and in this step there is no change.

In this way, after performing all the thinning operations iteratively, the final output will be as follows:



To make the thin output m-connected some more points will be removed as follows.



This is the thin version of the input image A.

5.6 Thickening

Thickening is a morphological operation that enlarges selected foreground pixel regions in binary images. It is similar to dilation and closing operations. The nature of thickening is determined by the structuring element. Thickening is the dual operation of thinning, which means that thickening the foreground is the same as thinning the background. The thickening operation of a point set 'A' by a structuring element B is defined as

$$A \odot B = A \cup (A \circledast \mathrm{B})$$

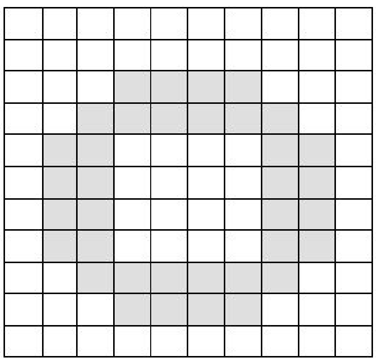
Here, A represent original point set, B represent structuring element, $\odot$ represent thickening operation and $\circledast$ represent Hit or Miss transform. Like thinning operation, here instead of using single structuring element, a set of structuring element is used. The set of structuring elements may be defined as

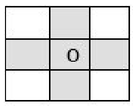
$$\{B\} = \{B^1, B^2, B^3, \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots . B^n\}$$

Where, all these structuring elements in this set follow a particular property, i.e., B^i is the rotated version of B^{i-1}. The meaning is that if we rotate the structuring element B^1 then we get the structuring element B^2, similarly, if we rotate the structuring element B^2 then we get the structuring element B^3 and so on. Using these set of structuring elements the thickening operation perform iteratively as follows.

$$A \odot \{B\} = ((\ldots \ldots \ldots \ldots.((A \odot B^1) \odot B^2) \ldots \ldots \ldots..) \odot B^n)$$

The thinning operation is applied iteratively, first thicken point set A with structuring element B^1, then thicken it with structuring element B^2 and so on, and finally thicken it with B^n. This process continues until convergence occurs. Since, thickening is dual of thinning, one way of implementing the thickening operation is that for the given set A, first take the complement of A. Then perform the thinning of A^c with the structuring element B. After performing the thinning operation again, perform the complement of the thin version of the point set A. This output will be the thickening of A. Mathematically, it can be written as



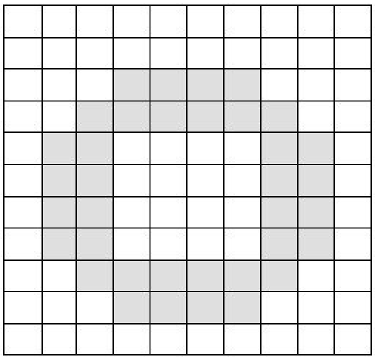


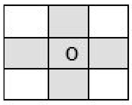
Structuring element B

Input image A

CHECK YOUR PROGRESS

Q No 1 : Perform boundary extraction for the following binary image with the structuring element shown below.





Structuring element B

Input image A

5.7 Summing up

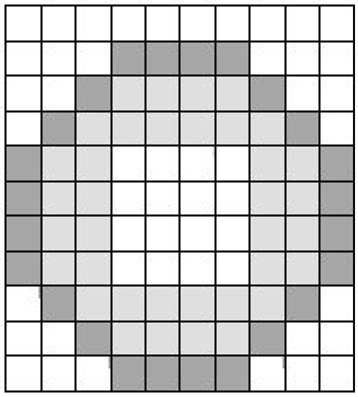
- Boundary extraction is a morphological operation applied on binary image to extract the boundary of an image.
- Convex set or convex hull is a property on binary image.
- Thinning is a morphological operation applied on binary image to find the skeleton on a binary image.
- Thickening is a morphological operation applied on binary image to enlarge the foreground of a binary image.

5.8 Answer to check your progress

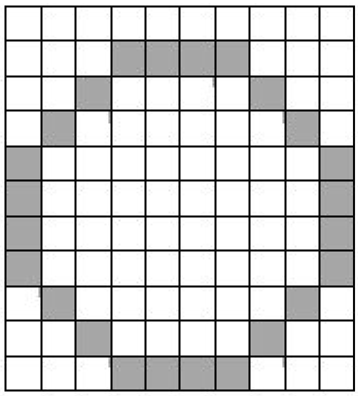
Q. No1

Solution: To find the boundary of the input binary image we use the following morphological operation

$Y = (X \oplus B) - X$



$(X \oplus B)$



$(X \oplus B) - X$

This is the required boundary of the image or object.

5. 9 Possible Questions

Q1. What is boundary extraction? Explain how to extract boundary of a binary image?

Q2 Explain the importance of convex set and convex hull in image processing

Q3 Explain thinning operation with example.

Q4 Explain thickening operation with examples.

Q5 What is the difference between thickening and thinning.

5.10 References and Suggested Readings

1.Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, Mc. Graw Hill

2. Digital Image Processing, 2^{nd} ed,S Sridhar, Oxford

3. NPTEL, IITKGP

4. Didital Image Processing, Dr. Sanjay Sarma, SK Kataria & Sons

*****

UNIT: 6
COLOR IMAGE PROCESSING

Unit Structure:

6.1 Introduction

A color model is a mathematical framework used to represent colors in a structured and standardized way, enabling consistent color manipulation and reproduction across various devices and applications. Different color models serve distinct purposes based on how colors are generated and perceived. For example, the RGB color model, comprising Red, Green, and Blue components, is used in digital displays and cameras, leveraging additive color mixing. Conversely, the CMYK model, with Cyan, Magenta, Yellow, and Black components, is employed in printing, utilizing subtractive color mixing. Other models like HSV (Hue, Saturation, Value) and Lab focus on human color perception and are utilized in image editing and color correction. Understanding these models is essential for tasks ranging from digital imaging to printing, ensuring accurate and consistent color representation.

6.2 Objectives

After going through this unit you will be able to:

- *understand* the concept of color models and their role in digital image processing,
- *differentiate* between additive and subtractive color models, and explain their practical applications,
- *describe* the RGB color model and its use in representing digital color images,
- *explain* the CMY and CMYK color models, particularly in the context of printing and color reproduction,
- *interpret* the HSI color model, and understand its advantages for image analysis and enhancement,
- *define* and apply color quantization techniques to reduce the number of colors in an image while maintaining visual quality,
- *analyze* and interpret the histogram of a color image, understanding how pixel intensity distributions relate to image characteristics such as brightness and contrast,
- *apply* knowledge of color models and histograms to solve real-world problems in image processing, such as enhancement, segmentation, and classification.

6.3 Color Model

Color can be defined through its three characteristics, *Hue*, *Saturation* and *Luminance*. *Hue* is the dominant wavelength of reflected light that we perceive; *Saturation* is the purity of the color, and *Luminance* is the lightness or intensity of the color. *Hue* and *saturation* are called the *chroma* components of color and Luminance is called the *Luma* component of color. Two different light sources with suitably chosen intensities can be used to produce a range of other colors. If the two sources combine to produce white light, they are referred to as *complementary colors*.

A color model is an abstract mathematical model that describes how colors can be represented as a set of numbers (e.g., a triple in RGB or a quad in CMYK). Color models can usually be described using a coordinate system, and each color in the system is represented by a single point in the coordinate space.

For a given color model, to interpret a tuple or quad as a color, we can define a set of rules and definitions used to accurately calibrate and generate colors, i.e. a mapping function. A color space identifies a specific combination of color models and mapping functions. Identifying the color space automatically identifies the associated color model. For example, Adobe RGB and sRGB are two different color spaces, both based on the RGB color model.

A color model ensures consistent and accurate color representation across different devices and applications. It standardizes color manipulation, enhances interoperability, and aligns with human perception for precise color reproduction. This facilitates efficient processing, storage, and communication of color information, ensuring reliable and high-quality visuals in digital imaging, printing, and multimedia.

6.3.1 Additive Color Model

- These types of models use light which is emitted directly from a source to display colors.

- These models mix different amount of RED, GREEN, and BLUE (primary colors) light to produce rest of the colors.
- Adding these three primary colors results in WHITE image.

Example: RGB model is used for digital displays such as laptops, TVs, tablets, etc.

6.3.2 Subtractive Color Model

- These types of models use printing inks to display colors.
- Subtractive color starts with an object that reflects light and uses colorants to subtract portions of the white light illuminating an object to produce other colors.
- If an object reflects all the white light back to the viewer, it appears white, and if it absorbs all the light then it appears black.

Example: Graphic designers used the CMYK model for printing purpose.

6.3.4 RGB Model

The model's name comes from the initials of the three additive primary colors, red, green, and blue. The RGB color model is an additive color model in which red, green, and blue are added together in various ways to reproduce a wide range of colors

Usually, in RGB a pixel is represented using 8 bits for each red, green, and blue. This creates a total of around 16.7 million colors (2^{24}). Equal values of these three primary colors represents shade of gray color ranging from black to white.

Let's plot these three primary colors on a 3-dimensional plane in the form of a cube, the RGB values will be at the corners present on the three axes. The origin (0, 0, 0) will be black, and the diagonal opposite to the origin will be white (1, 1, 1). The rest three corners of the cube will be cyan, magenta, and yellow. The imaginary line connecting the Black and White is the Gray Line. Inside the cube, we get a variety of colors represented by the RGB vector (origin at black).

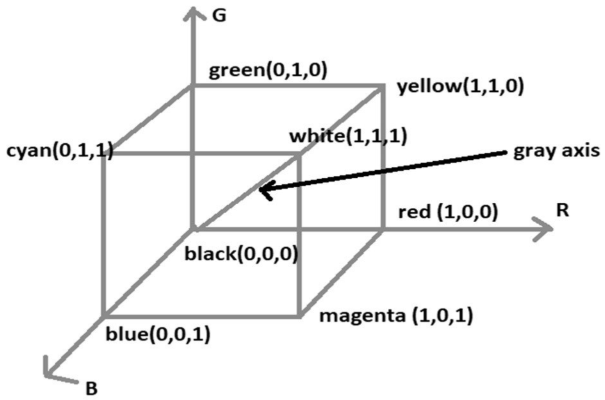


Fig. 1: RGB colour space visualized as a cube

For finding the composite color in RGB color model at any point. The following steps need to be followed:

Steps 1: Pixel depth is the number of bits used to represent each pixel. If an image in RGB model has 8-bits image in each of its three colors, then each RGB pixel has a dept of 3 image planes *8 -bit per plane that is 24 bits. This gives rise to 2^{24} color shades.

Step 2 : We fix one of the three colors and let the other two colors to vary. Suppose we fix R=127 and let G and B to vary. Then the color at any point on the plane parallel to GB plane would be (127, G, B), where, G and B=0, 1……….255.

Fig. 2 (a)

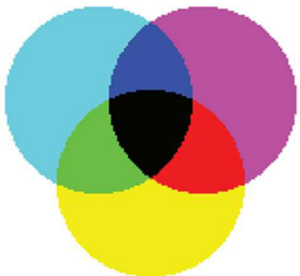


Fig. 2 (b)

Fig. 2(a) shows the RGB model, in which the white color is produced by adding the three primary colors Red, Green and Blue. Fig. 2(b) shows the CMY model, where black is obtained as the sum of Cyan, Magenta and Yellow. The inverse relation between the RGB and CMY models are also shown by these two images. In practice, black is obtained by combining cyan, yellow and magenta. However, this leads to a muddy looking black.

With the help of the primary colors, we can generate secondary colors (Yellow, Cyan, and Magenta) as follows.

Color combination:

Green(255) + Red(255) = Yellow

Green(255) + Blue(255) = Cyan

Red(255) + Blue(255) = Magenta

Red(255) + Greeb(255) + Blue(255) = White

All the display devices fundamentally use the RGB color model to produce various possible colors. The display devices use *additive color mixing* to produce other possible colors.

Example: Consider the image with different colors as given in Fig 3(a). Let us see how the RGB colors which would appear on monochrome display. You may assume that all colors are maximum intensity and saturation. Also show each of the color in black and white while considering them as 0 and 255 respectively.

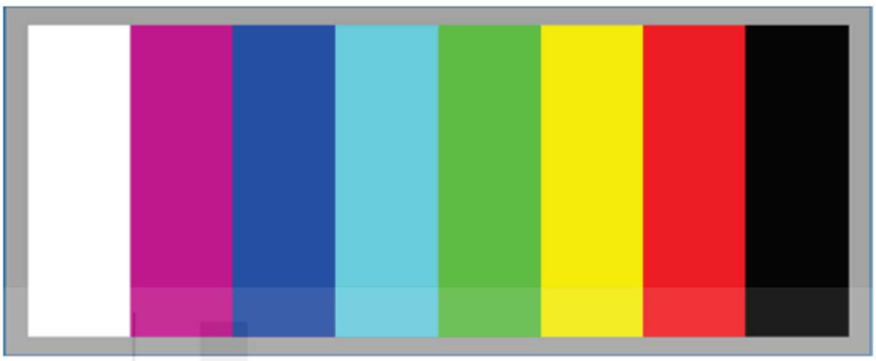


Fig. 3 (a)

It is given here that the intensity and saturation are maximum, therefore the value of each of the components RGB would be 0 or 1.

Let us check each color of the image one by one by starting from the left most color.

Colour	**RGB Combination**	**Intensity/ Saturation**			**Monochrome colours**		
		R	**G**	**B**	**R**	**G**	**B**
White	R+G+B	1	1	1	255	255	255
Magenta	R+B	1	0	1	255	0	255
Blue	B	0	0	1	0	0	255
Cyan	G+B	0	1	1	0	255	255
Green	G	0	1	0	0	255	0
Yellow	R+G	1	1	0	255	255	0
Red	R	1	0	0	255	0	0
Black	NIL	0	0	0	0	0	0

Now hence forth we shall follow the conversion that 0 represents black and 255 represents white. Also, the grey is represented by 128. You see that the table has R colors series as 255,255,0,0,255,255,0. Thus, it would show W, W, B, B, B, W, W, B in monochrome display, which is shown in Fig. 4(a).

Similarly, monochrome display of green color would be shown by the series W, B, B, W, W, W, B, B and blue would be shown as W, W, W, W, B, B, B, B, B as shown in Fig. 4(b) and Fig. 4(c)

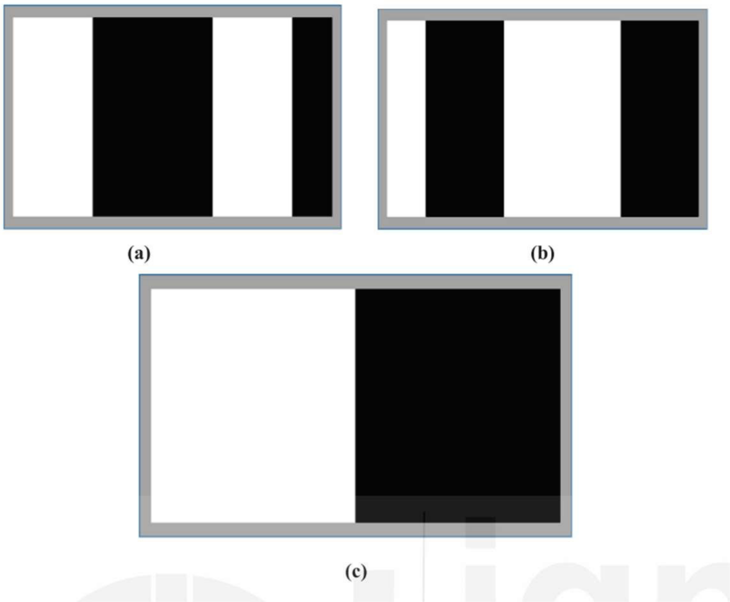


Fig. 4

6.3.5 CMY Model

The CMY color model is a subtractive color model in which cyan, magenta, and yellow (secondary colors) pigments or dyes are mixed in different ways to produce a broad range of colors. The secondary colors are also called the primary color pigments. The CMY color model itself does not describe what is meant by cyan, magenta, and yellow calorimetrically, so the mixg results are not specified as absolute but relative to the primary colors. When the exact chromaticity of the cyan, magenta, and yellow primaries is defined, the color model then becomes an absolute color space.

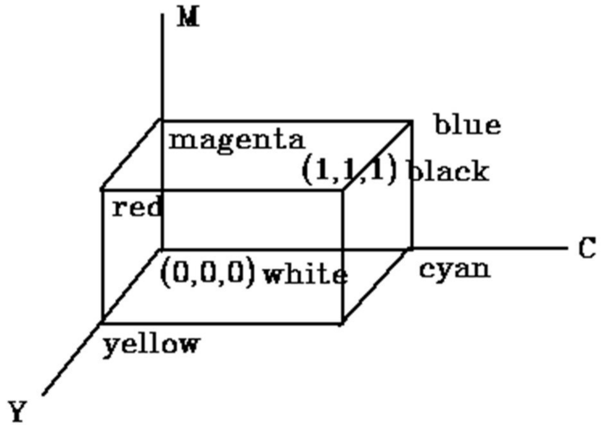


Fig. 5: CMY colour space visualized as a cube

Let's plot these three primary colors on a 3-dimensional plane in the form of a cube, along the three principle axes, we have now Cyan (C), Magenta (M), and Yellow (Y) colors. At the origin (0,0,0) we have White, and on the diagonally opposite vertex at (1,1,1) we have Black. The imaginary line connecting the White and Black vertices forms the Gray Line.

when all the three, the C, M, Y are combined in their purest forms they are supposed to produce Black. But it is not so, and we get close to 99% Black, and we can make it 100% Black, by adding an extra Black pigment. This technique is used in printing industry, where Black is needed for printing on the paper and the inks used are C, M, Y. The addition of this Black pigment is indicated with a suffix symbol '*K*' to make it CMYK color model.

6.3.6 CMYK Model

The CMYK color model describes colors based on their percentage of Cyan, Magenta, Yellow and Black. Many computer printers and traditional "four-color" printing presses use the CMYK model. In the CMYK model, by mixing cyan, magenta, yellow and black inks or paints, you can create nearly any color desired.

CMYK is a subtractive color model, which means colors get darker when mixed. Each of the mixed paints or inks absorbs different components of the light. If the right combination of paints is mixed together, all of the components of light are absorbed and the result is a near black.

CMYK (Cyan, Magenta, Yellow, Key/Black) is the color model used for printed materials. A printing machine creates images by combining CMYK colors to varying degrees with ink.

The Process of Color Subtraction

The methodology of color subtraction is a valuable way of predicting the ultimate color appearance of an object if the color of the incident light and the pigments are known. The relationship between the RGB and CMY color models is given by:

RGB = 1 – CMY or CMY = 1 – RGB

Similarly, if we throw a magenta light, a combination of red and blue, on a yellow pigment, the result will be a red light because the yellow pigment absorbs the blue light.

6.3.7 HSI Model

The RGB and CMY color models are ideally suited for hardware implementations; however, these are not reasonably suited for representing colors in terms that are practical for human interpretation. RGB is not a particularly intuitive way to describe colors. HSI stands for Hue, Saturation, and Intensity. When humans view a color object, its hue, saturation, and brightness are described.

Hue: It is a color attribute that describes a pure color.

Saturation: It measures the extent to which a pure color is diluted by white light.

Brightness: It depends upon color intensity, which is a key factor describing the color sensation. The intensity is easily measurable, and the results are also easily interpretable.

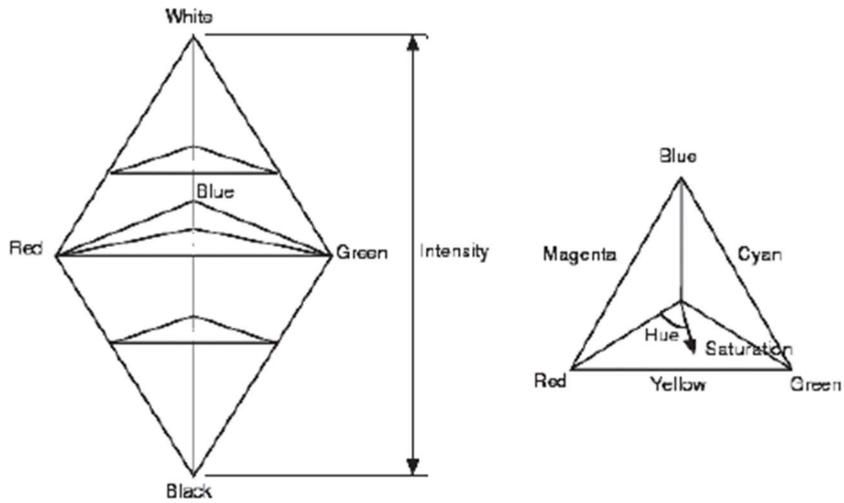


Fig. 6: HSI and RGB colour model

In Fig. 6, the HSI colour model is represented and its relation to RGB model is shown. The triangle shows a slice from the HSI solid at a particular intensity. The hue, saturation and intensity values required to form the HSI colour space can be computed using the RGB values.

Now, let us see how a RGB colour model is converted into a HSI colour model.

To convert an image in RGB to HSI colour space, the RGB value of each pixel in the image is converted to the corresponding HSI value using the equations given below:

Hue, H is given by

$$H=\begin{cases} \theta, & if\ B \leq G \\ 360-\theta, & if\ B > G \end{cases}$$

Where

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

Saturation, S is given by

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

And, intensity, I is given by

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

Here the RGB values have been normalized in the range [0,1], and the angle θ is measured with the respect to the red axis in the HSI space.

We can also convert HSI colour model to RGB colour space. The pixel values in the HSI colour space in the interval [0,1], the RGB values can be computed in the same range. However, depending on the H value, the RGB values are computed in different sectors based on separation of the RGB colours by 120 d intervals.

6.4 Color Quantization

Color quantization is a digital image processing technique that aims to reduce the number of distinct colors in an image while maintaining its overall visual quality. This is achieved by grouping similar colors together and representing them with a single representative color, effectively reducing the image's color palette. The primary goal of color quantization is to optimize image storage and transmission by reducing the amount of data required to represent the image without significantly affecting its appearance or quality.

Color quantization is a process that involves several key steps to reduce the number of colors in an image while maintaining its overall appearance. The main steps include:

1. **Color space analysis**: Examine the colors present in the original image to understand their distribution and frequency.
2. **Palette selection**: Choose a target color palette with fewer colors, using algorithms that consider color distribution and visual perception factors.
3. **Color clustering**: Group similar colors in the original image, assigning them a representative color from the selected palette.
4. **Color mapping**: Replace the original colors in the image with their corresponding representative colors from the reduced palette, minimizing visual differences between the original and quantized images.

By following these steps, color quantization can balance image quality and data efficiency, making it a valuable technique for various applications in digital image processing.

Some of the key situations where color quantization can be useful include:

- **Image compression**: Reducing the number of colors in an image can lead to smaller file sizes, making it ideal for optimizing web images and reducing bandwidth usage.
- **Indexed color**: When creating images for specific devices or formats that support a limited color palette, color quantization can help you stay within those constraints.
- **Stylized visuals**: Artists and designers may choose to limit the color palette for creative reasons, such as evoking a particular mood, aesthetic, or retro-inspired look.

- **Data visualization**: Simplifying the color scheme can make complex data visualizations easier to understand and interpret by reducing visual clutter.
- **Color printing**: Color quantization can minimize the number of inks or toners required for printing, potentially reducing costs and improving print quality.

Pixels in images could have associated 24-bits containing at most 224 = 16,777,216 different colors. These colors are represented as three dimensional vectors, each vector element with 8-bit dynamic range, allowing 256 different values. These vectors are often called RGB triplets. A smaller set of representative colors of the image is called color palette.

6.5 Histogram of a color Image

A histogram of a color image represents the distribution of pixel intensities across the individual color channels—Red, Green, and Blue—of an RGB image. Unlike grayscale histograms, which reflect overall brightness, color histograms capture the frequency of intensity values in each channel separately, helping to analyze the color composition and contrast of an image.

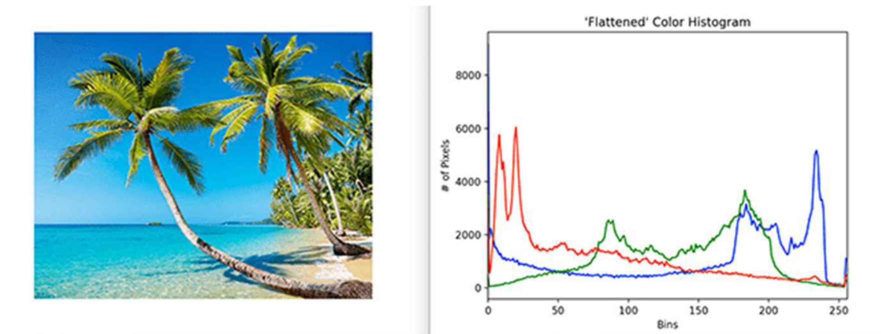


Fig. 7: Colour histogram for a RGB image

A color histogram in digital image processing typically includes the following components:

1. **Color Channels**: A color image is composed of three separate color channels. Histograms are calculated independently for each channel (Red, Green, and Blue in RGB format). This means that for a single image, we usually generate and examine three histograms. Each histogram provides insights into the tonal distribution of its respective color.
2. **Bins or Intensity Levels**: The intensity range of pixel values is divided into discrete intervals called bins. For an 8-bit image, there are 256 possible intensity levels, ranging from 0 (black or no intensity) to 255 (full intensity). The histogram counts the number of pixels that fall within each bin for each channel.
3. **Visualization and Interpretation**: Color histograms are typically visualized as line plots or bar graphs, with the x-axis representing intensity levels and the y-axis representing the frequency of those intensities. By analyzing these plots:
    - Peaks in a histogram indicate dominant intensities.
    - A wider spread suggests higher contrast.
    - A shift toward higher or lower intensity values can indicate an image that is too bright or too dark, respectively.
4. **Applications**: Color histograms are used in various image processing and computer vision tasks, including:
    - **Image enhancement**: Techniques like histogram equalization use histograms to improve contrast.
    - **Object recognition**: Histograms can be used as features in image classification tasks.
    - **Image retrieval**: In content-based image retrieval (CBIR), histograms are compared to find similar images based on color distribution.

5. **Normalization**: To compare histograms across images of different sizes, histograms can be normalized. This involves dividing the frequency count in each bin by the total number of pixels, resulting in a probability distribution of intensities.
6. **Cumulative Histograms**: Sometimes, cumulative histograms are used, where each bin represents the sum of its own frequency and those of all previous bins. This is useful for operations like thresholding and histogram equalization.

Check Your Progress

Multiple choice questions:

1. Which color model is based on the human perception of color and separates chromatic content from intensity?
   a) RGB
   b) CMY
   c) HSI
   d) YUV
2. In the additive color model, combining red, green, and blue in full intensity results in:
   a) Black
   b) White
   c) Yellow
   d) Cyan
3. Which color model is commonly used in color printers?
   a) RGB
   b) HSV
   c) CMY
   d) XYZ

4. How many intensity levels are typically used in an 8-bit color image histogram for each channel?
   a) 128
   b) 64
   c) 512
   d) 256
5. Color quantization is primarily used for:
   a) Increasing resolution
   b) Converting grayscale images to color
   c) Reducing the number of colors in an image
   d) Enhancing image brightness

6.6 Summing Up

1. This unit provided a comprehensive overview of colour models used in digital image processing. It began by introducing the concept of colour representation and the need for standardized color models.

2. The Additive Colour Model, which combines red, green, and blue light to produce other colours, was explained alongside the Subtractive Colour Model, which removes wavelengths using cyan, magenta, and yellow pigments.

3. The RGB Model, is widely used in digital screens and imaging devices.

4. The CMY Model, is primarily used in color printing, along with its extended form, CMYK.

5. The HSI Model, which separates color information into hue, saturation, and intensity—making it more intuitive for human interpretation and useful for tasks like image enhancement and segmentation.

6. Colour quantization, is a crucial process in image compression and palette reduction, explaining both uniform and non-uniform quantization techniques.

7. The histogram of a colour image, shows how separate histograms for red, green, and blue channels provide insight into the tonal and color distribution of an image.

6.7 Answers to Check Your Progress

Ans. 1. (C) 2. (B) 3.(C) 4.(D) 5. (C)

6.8 Possible Questions

1. What is the main difference between the additive and subtractive color models?
2. Why is the HSI model preferred in image processing for segmentation and enhancement?
3. What is color quantization and where is it used?
4. Explain the RGB color model. How does it represent colors and where is it commonly used?
5. Describe the CMY and CMYK color models. How do they differ from the RGB model?
6. What is a color image histogram? Explain its structure and importance in digital image processing.

6.9 References and Suggested Readings

1. R.C. Gonzoles and R.E. Woods, Digital Image Processing, Addision-wesley, 1992

2. A.K. Jain, Fundamentals of Digital Image Processing, PHI

*****

Block: III

Unit 1: Fundamentals of Image Segmentation
Unit 2: Image Segmentation- I
Unit 3: Image Segmentation- II
Unit 4: Image Compression- I
Unit 5: Image Compression- II
Unit 6: Image Compression- III

UNIT: 1
FUNDAMENTALS OF IMAGE SEGMENTATION

Unit Structure:

1.1 Introduction

Image segmentation subdivided an image into some region or objects and this process should stop when the objects or regions of interest in an application have been detected. Here in this unit we will discuss different types of image characteristics of discontinuity based approaches. Also in this unit we will discuss the classification of edge model and how first order derivatives is used to detecting the edge from an image. In this unit we also discuss the computation of gradient of an image using masks and discuss the uses of different gradient operators.

1.2 Objectives

After going through this unit learner will able to

- *understand* the basic properties of intensity values,
- *understand* the edge, line and isolated point detection,
- *learn* edge model and its classification,
- *understand* the concept of basic edge detection,
- *understand* the concept of gradient computation,
- *learn* the uses of different gradient operators like Roberts, Prewitt and Sobel operators.

1.3 Image Segmentation

Image segmentation is a process for segmenting the images in the different regions where each region has its specific meaning. The segmentation or subdivision is always application dependent. Image segmentation is the most important as well as the most difficult tasks in the image analysis. Most of the segmentation algorithm is based on the two approaches one is discontinuity and the other is similarity. The discontinuity based approaches are based on the partition on region of the image where the abrupt changes are occurs such as edge of the image. The similarity based approaches are based on the partition of the images according to the set of predefined criteria. The primary goal of discontinuity based segmentation is to identify the boundaries and using these boundaries images are segmented into some meaningful regions. Under discontinuity based approaches, we are interested in identification of isolated points or identification of lines or identification of edges of the image. Under similarity based approaches we try to group those pixels in an image which are similar in nature. The approaches under similarity based techniques are thresholding, region growing and region splitting and merging operations.

1.4 Point, line and edge detection

1.4.1 Edge detection

Three types of image characteristics of discontinuity based approaches are edge, line and isolated point detection. Edges pixels are pixels where the intensity of the image changes abruptly. A line may view as thin edge segment where intensity of the background may be higher or lower. The point may be viewed as a foreground pixel surrounded by background pixels.

We obtain the Taylor's series of f(x) by expanding the function f(x+Δx) is as follows

$$f(x+\Delta x) = f(x) + \Delta x\frac{\partial f(x)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x)}{\partial x^2} + \frac{(\Delta x)^3}{3!}\frac{\partial^3 f(x)}{\partial x^3} + \cdots$$

$$= \sum_{n=0}^{\infty} \frac{(\Delta x)^n}{n!}\frac{\partial^n f(x)}{\partial x^n} \qquad (1)$$

Here Δx is the separation between the samples of x

When $\Delta x = 1$ the above equation becomes

$$f(x+1) = f(x) + \frac{\partial f(x)}{\partial x} + \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2} + \frac{1}{3!}\frac{\partial^3 f(x)}{\partial x^3} + \cdots$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!}\frac{\partial^n f(x)}{\partial x^n} \qquad (2)$$

Similarly when $\Delta x = -1$

$$f(x-1) = f(x) - \frac{\partial f(x)}{\partial x} + \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2} - \frac{1}{3!}\frac{\partial^3 f(x)}{\partial x^3} + \cdots$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{n!}\frac{\partial^n f(x)}{\partial x^n} \qquad (3)$$

To compute the intensity difference we have to use the few terms of the Taylor's series. For the first order derivative we only use the linear terms, the differences can be form in the following ways

The forward difference can be obtained from the equation (2) as follows

$$f^{/}(x)=\frac{\partial f(x)}{\partial x} = f(x+1) - f(x) \qquad (4)$$

We obtain an expression for the second derivative by differentiating equation (4) with respect to x is

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

$$= f(x+2) - f(x+1) - f(x+1) + f(x)$$

$$= f(x+2) - 2f(x+1) + f(x)$$

The above expansion is about the point (x+1). Our main focus on the second derivative about point x, so we subtract 1 from the arguments in the preceding expression and obtain the following result

$$\frac{\partial^2 f(x)}{\partial x^2} = f^{//}(x) = f(x+1) - 2f(x) + f(x-1) \qquad (5)$$

The first order derivative produces "thick" edges and the second order derivatives thinner in comparison to the first order derivative. The second order derivatives have a stronger response to the details, such as thin lines, isolated point and noise. Second order derivative produce a double-edge response at ramp and step transition in intensity. The sign of the second order derivative can be used for the transition into an edge from light to dark or dark to light. The spatial convolution is used for computing first and second derivative at every pixel location in an image.

The approach for computing first and second derivatives at every pixel location in an image is to use spatial filters.

Consider the following 3x3 spatial filter kernel

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

The above filter is used to compute the sum of products of the kernel coefficient with the intensity values in the region hold by the kernel. The response of the filter at the centre point is as explained by the following equation

$$R = w_1z_1 + w_2z_2 + w_3z_3 + \text{---------------} + w_9z_9$$

$$= \sum_{k=1}^{9} w_k z_k \qquad (6)$$

Where z_k is the intensity of the pixel whose spatial location is corresponding to the location of the k^{th} kernel coefficient in the mask.

1.4.2 Detection of isolated point

Point detection refers to identifying a specific point of interest of an image, called an interest point. The point has significant intensity variation in compared to its surrounding regions on which the kernel is centered. At that point the absolute value of the response of the filter exceeds a specified threshold.

These types of point are designated as 1 and all other terms are termed as 0. That means it will produce a binary image. In an expression it can be explained as follows

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & \text{otherwise} \end{cases}$$

Where g(x,y)→ is the output image

T→ non-negative Threshold

And R is given by the equation (6). This formula simply measures the weighted difference between a pixel and its 8 neighbors. This inherits the intensity of an isolated point is completely different from its neighboring and will be easily detected by this type of kernel filter. The portion where the differences of intensity is large enough which is determined by T is considered as isolated point.

Stop to Consider

For a derivative mask, the coefficient sum to zero, indicating that the mask response will be zero in areas of constant intensity

The example of point detection (Laplacian) mask is as follows

1	1	1
1	-8	1
1	1	1

1.4.3 Line Detection

We know that for detection of line the second derivative give strong response and it is used to produce thinner lines than first derivatives. Thus we can use the Laplacian mask for detection of line also. Whenever the Laplacian mask is used for line detection purpose it should be keep in our mind that the double-line effect of the second derivative must be handled properly. In an image, lines can be in any direction and detecting these lines it would require different masks. The following masks are required for detecting line in different directions.

-1	-1	-1
2	2	2
-1	-1	-1

Figure.1 (a) Horizontal

-1	-1	2
-1	2	-1
2	-1	-1

Figure. 1 (b) $+45^0$

-1	2	-1
-1	2	-1
-1	2	-1

Figure. 1 (c) Vertical

2	-1	-1
-1	2	-1
-1	-1	2

Figure.1 (d) -45^0

All the above masks have a sum equal to zero and hence all of them are high pass masks. The first mask in Figure 1(a) is strongly response to a horizontal line. The second mask in Figure 1(b) response to a line at an angle of $+45^0$. The third mask in Figure 1(c) response to a vertical line and fourth mask in Figure 1(d) response to line at an angle -45^0.

Check Your Progress-I

1. State True or False
   - (i) Point detection is a discontinuity based approach.
   - (ii) The first order derivatives have a stronger response to the details, such as thin lines, isolated point and noise.
   - (iii) We cannot use Laplacian mask for detecting a line.
   - (iv) The sign of the second order derivative can be used for the transition into an edge from light to dark.
   - (v) The second derivative gives strong response for detection of a line.

1.5 Edge Models

The approach of edge detection is used most frequently for segmenting images based on abrupt changes in intensity. According to the intensity profile edge model are classified to different edges as follows

Step edge: A step edge involves a transition between two intensity levels occurring over the distance of one pixel. For example the image generated by a computer is use for the areas like modeling and animation. The boundary of an object usually produces a step edge because the intensity of the object and the background is different. The vertical step edge is shown in the figure 2.

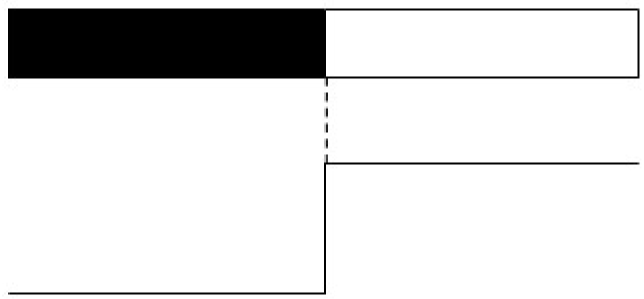


Figure 2. A step edge and their corresponding intensity profiles

Ramp edge:

In the ramp model, we have no longer a thin path. Unlike the step edge, where the change of intensity is abrupt; a ramp edge has a smooth transition. Ramp edge could be represented a surface which is gradually changes its lighting or color like a slope or shadowed area. An edge point in this model is any point in the ramp, and an edge segment is the set of such point that are connected. A ramp edge is shown in the Figure 3.

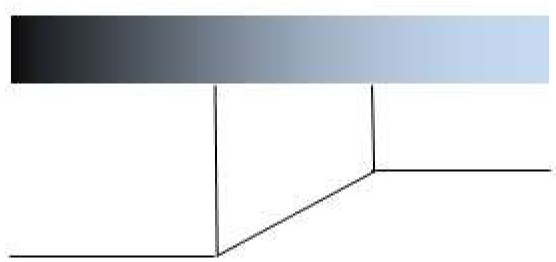


Figure 3. A ramp edge and their corresponding intensity profiles

Roof edge:

Roof edges are models of lines through a region, with the width of the base of a roof edge is determined by the thickness and sharpness of the line. It occurs where there is a peak of intensity value of an image, resembling the shape of a roof. The intensity value of a roof edge is gradually increases to maximum or gradually decreases to minimum value. In terms of intensity value roof edge are symmetric. Detecting the roof edge is more complex in comparison to step edge because it is occur in the border area. Roof edge can appear in the image where there are ridges or folds in the object. For example the image of folded cloth, the fold can create a roof edge. A roof edge is shown in the Figure 4.

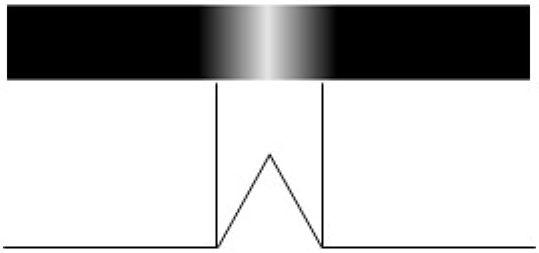


Figure 4. A roof edge and their corresponding intensity profiles

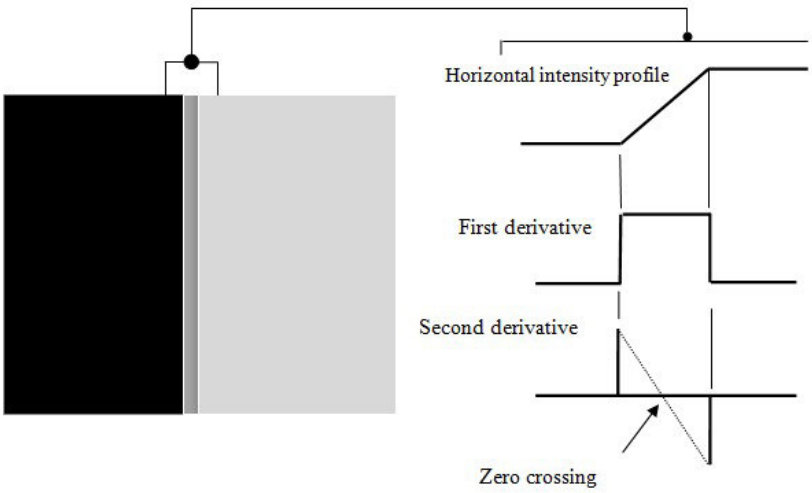


Figure 5. Two regions of constant intensity separated by an ideal vertical ramp edge and horizontal intensity profile with its first and second derivatives.

The Figure 5 shows the horizontal intensity profile along with its first order derivative and second order derivative. The first derivative is positive at the onset of the ramp and it is zero in the area of constant intensity. The second derivative is positive at the initial state of the ramp, and thus it is negative at the end of the ramp. The sign of the derivatives is reversed for an edge that is transit from light to dark. In the second derivative marks a point where the intersection between the zero intensity axis and a line extending between the extrema is called zero crossing. So we can conclude that the magnitude of the first derivative can be used to detect the presence of an edge at a point in an image. The sign of the second derivative is used to determine whether the edge pixel is lies on the light or dark of an edge. We observed the second derivative has two additional properties, which are

(i) It will produce two values for every edge in an image.

(ii) Its zero crossing is used for locating the centre of the thick edges.

1.6 Basic edge Detection

Edges can be detected using the first order derivative and second order derivatives. First we have to discuss how first order derivatives is used to detecting the edge from an image. The first order derivative is computed by the gradient operator. If we have a function f(x, y) then the gradient, denoted by ∇f and defined as the vector

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad \ldots\ldots\ldots\ldots (7)$$

The magnitude of the vector ∇f, denoted as M(x,y), where

$$\mathbf{M}(x,y) = \mathbf{mag}\ (\nabla f) = \sqrt{g_x^2 + g_y^2} \quad \ldots\ldots\ldots\ldots \ (8)$$

g_x, g_y and M(x,y) are images of same size as the original image created and x, y are allowed to vary over all the pixel location in f.

g_x → is the gradient in the x- direction

g_y → is the gradient in the y-direction

This tells what is the strength of the edge at location (x,y), it does not tell us anything about what is the direction of the edge at the point (x,y). So we have to compute the direction of the edge that is the direction of gradient vector.

The direction of the gradient vector is given by the following angle

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(9)}$$

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

Example of properties of gradient

Let us consider a zoomed section of an image which contains a straight edge segment. Each square represented to a pixel and we are interested to obtaining the edge direction at the highlighted point in the square. Here the pixel of the gray value is taken as 0 and the pixel of white value is taken as 1. This example has an approach to compute the x-derivatives and y-derivatives using a 3×3 neighborhood centered about a point. For obtaining the partial derivative in the x-direction we subtract the pixels in the top row of the neighborhood from the pixels in the bottom row. Similarly, we subtract the pixels in the left column from the pixels in the right column to obtain the partial derivative in the y-direction. Using this assumption and the differences as our estimates of the partial derivative we get $\frac{\partial f}{\partial x}$ = -2 and $\frac{\partial f}{\partial y}$= 2 at the point in the question. Then the gradient of f is as follows

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

From the above we get $M(x,y)=2\sqrt{2}$ at that particular point . The direction of the gradient vector at the same point is as follows

$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right] = -45^0$ this is measured as 135^0 in the positive direction with respect to the x-axis. Here the direction angle of the edge in this example is $\alpha - 90^0 = 45^0$. All edge points in the Figure 6 (a) have the same gradient, so the entire edge segment is in the same direction.

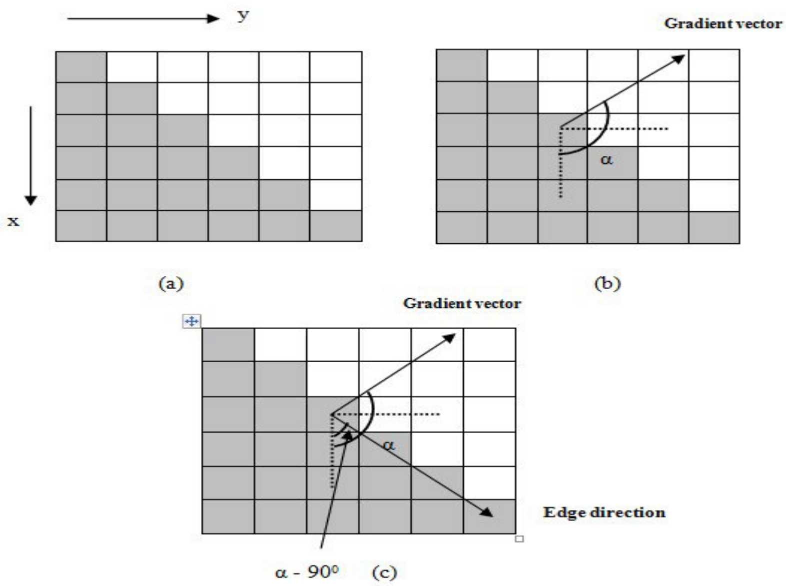


Figure 6. Using the gradient to determine the edge strength and direction at a point

1.7 Computing the Gradient

Case 1:

If the output of the function y=f(x) is a single variable then the derivative of y with respect to x is

$\frac{dy}{dx} = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h} = f'(x)$

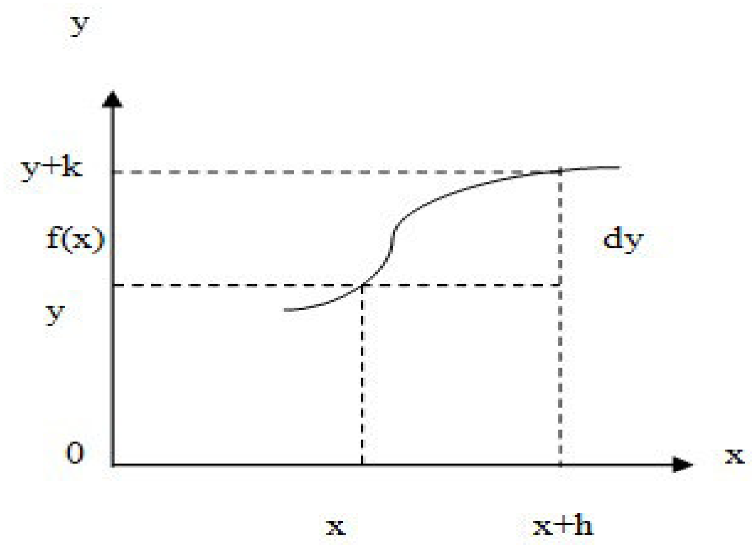


Figure 7: Derivative of y with respect to x.

In Figure. 7 the slope is defined by $\frac{dy}{dx}$

Case 2: If output of the function has two variables then both these variables will change to x+h and y+k. Using the partial derivatives here we get

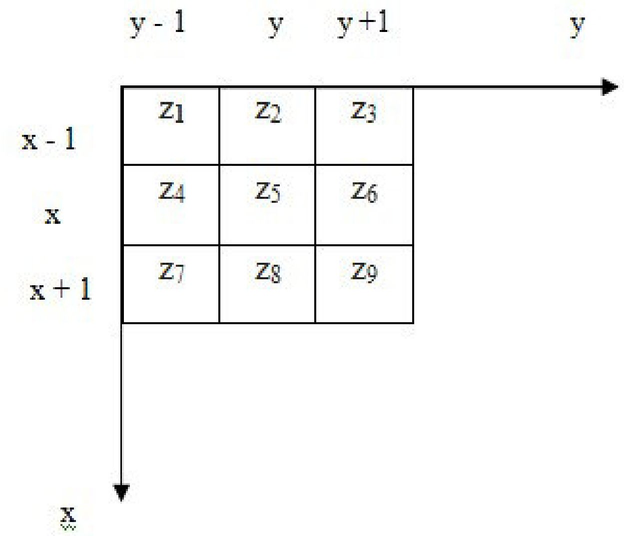
$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x+h, y) - f(x, y)}{h}$$

Similarly keeping x is fixed and changing y, we get

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

Now finding the gradient using mask

The resultant 3x3 mask we have taken as



In a discrete domain h=k=1

From the above two equation, we get

$\frac{\partial f}{\partial x}$ = f (x+1, y) – f (x,y) ………………………… (10)

$\frac{\partial f}{\partial y}$ = f (x, y+1) – f(x,y) …………………………..(11)

From the 3x3 neighborhood we have

$\frac{\partial f}{\partial x} = z_8 - z_5$

$\frac{\partial f}{\partial y} = z_6 - z_5$

Since we have

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$|\nabla f| = \sqrt{(z_8 - z_5)^2 + (z_6 - z_5)^2}$$

$$|\nabla f| = |z_8 - z_5| + |z_6 - z_5|$$

$$|\nabla f| = |z_5 - z_8| + |z_5 - z_6|$$

It can be implemented using two masks

$|z_5 - z_8|$ =

z_5	0
$-z_8$	0

=

1	0
-1	0

The above mask is termed as mask-1

Again for

$|z_5 - z_6|$ =

z_5	$-z_6$
0	0

=

1	-1
0	0

It is termed as mask-2

Therefore to compute the gradient of an image the following steps are required

(i) Processing the original image with mask-1 which gives the gradient along x-axis

(ii) Processing the original image with mask-2 which gives the gradient along y-axis

(iii) Add the result of (i) and (ii)

When diagonal edge direction is of interest, we need a 2-D mask. The Roberts cross gradient operators found that the better result could be obtained if cross difference would be taken instead of straight difference. The Roberts operator can be implemented as

$$\frac{\partial f}{\partial x} = |z_9 - z_5| \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (12)$$

$$\frac{\partial f}{\partial y} = |z_8 - z_6| \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (13)$$

Roberts mask values are

-1	(0	-1
0	]1	0

The 2×2 masks are conceptually simple, but these masks are not so useful for computing the edge directions. In comparison to the 2×2 mask the 3×3 masks provide more information for computing the direction of edge. The simplest approximation to the partial derivatives using the mask of size 3×3 is given by

$$\frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (14)$$

and

$$\frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (15)$$

The equation (14) is the difference between the third and first row of the 3×3 region. It is the derivative in the x-direction. Again in the equation (15) is the difference between third and the first columns and it approximates the derivative in y-direction. We would expect this approximation is more accurate than the approximation obtained from the Roberts operators. The equation (14) and (15) can be implemented over an entire image with the following two masks. These masks are called the ***Prewitt operators***.

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

A slight variation is done in the above two equations by using a weight of 2 in the center coefficient is given as

$$\frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \text{ ……………………… } (16)$$

and

$$\frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \text{ ……………………..} (17)$$

The equations (16) and (17) can be implemented by using the following two masks and these masks are called *Sobel operators* .

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

From the above we have seen that the Prewitt masks are simpler to implement than the Sobel masks. But the Sobel masks have better noise suppression which makes these masks are more preferable. As we know that noise suppression or smoothing is an important issue when we dealing with the derivatives.

1.8 Laplacian Operator

The Laplacian operator is the second order derivative. It can be defined for a function f(x,y) of two variables as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad \text{.......................................} \ (18)$$

From the equation (5), keeping in mind we have to carry a second variable. In the x-direction, we have

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x,y) \quad \text{........................}(19)$$

Similarly, in the y-direction we have

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x,y) \quad \text{..........................}(20)$$

Therefore from the above three equations (18), (19) and (20) that the discrete Laplacian of two variable is

$$\nabla^2 f(x, y) = f(x+1,y) + f(x-1,y) + f(x, y+1) + f(x,y-1) - 4f(x,y) \quad \text{...............}(21)$$

The equation (21) can be implemented with the following mask

0	1	0
1	-4	1
0	1	0

Figure 8. Filter mask to implement equation (21)

The diagonal direction can be added in the definition of the digital Laplacian by adding two more term in equation (21). The form of each new term is same as either equation (19) or (20), but the co-ordinates are along the diagonals. Since the each diagonal term contain a -2f(x,y) term , the total subtracted from the difference terms would be -8f(x,y). The following filter mask is used to implement the new definition.

1	1	1
1	-8	1
1	1	1

Figure 9. Mask used to implement an extension of equation (21) including diagonal terms.

The following two Laplacian masks can be obtained from the definition of the second derivative that are negatives as we used in the equations (19) and (20) are as follows

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 10: two other masks for implementations of Laplacian operator

Laplacian is a derivative operator and the uses of it, highlights the intensity discontinuities in an image. If the definition used has a negative center co-efficient, then we subtract, the Laplacian image rather than add, to obtain a good result. Thus, the basic way of uses of Laplacian for image sharpening is

$$g(x,y) = f(x, y) + c[\nabla^2 f(x,y)] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (22)$$

Where f(x,y) and g(x,y) are the input and sharpened images, respectively. The constant is c = -1 the masks in Figure 6 and Figure 7 are used and for c = 1 the masks in Figure 8 are used.

Check Your Progress-II

2. State True or False
   - (i) Edge model are classified according to the intensity profile.
   - (ii) The intensity value of a roof edge is constant.
   - (iii) Edges cannot be detected using second order derivative.
   - (iv) The Prewitt masks are simpler to implement than the Sobel masks.
   - (v) Sobel masks have better noise suppression.

1.9 Summing up

- Image segmentation is a process for segmenting the images in the different regions where each region has its specific meaning.
- Three types of image characteristics of discontinuity based approaches are edge, line and isolated point detection.
- Edges pixels are pixels where the intensity of the image changes abruptly.
- A line may view as thin edge segment where intensity of the background may be higher or lower.

- The point may be viewed as a foreground pixel surrounded by background pixels.
- According to the intensity profile edge model are classified to different edges.
- The boundary of an object usually produces a step edge because the intensity of the object and the background is different.
- Ramp edge could be represented a surface which is gradually changes its color like a slope or shadowed area.
- Roof edge occurs where there is a peak of intensity value of an image, resembling the shape of a roof.
- In comparison to the 2×2 mask the 3×3 masks provide more information for computing the direction of edge.
- Sobel masks have better noise suppression which makes these masks are more preferable.
- The Laplacian operator is the second order derivative and the uses of it highlights the intensity discontinuities in an image.

1.10 Answer to Check Your Progress

1. (i) True	(ii) False	(iii) False	(iv) True	(v) True
2. (i) True	(ii) False	(iii) False	(iv) True	(v) True

1.11. Possible Questions

1. What are the different approaches of image segmentation?
2. What is the goal of discontinuity based approach?
3. How can you detect an isolated point in an image?

4. How Laplacian masks can be used in detection a line of an image?
5. What is edge model? Explain the classification of edge model.
6. What is zero-crossing?
7. Explain how first order derivatives is used to detecting the edge from an image.
8. How can you compute the gradient of an image?
9. Explain the process of implementation of Roberts mask.
10. Differentiate between Prewitt and Sobel operators.
11. What is Laplacian operator?
12. Explain the uses of Laplacian operator for image sharpening.

1.12 References and Suggested Readings

Gonzalez & Woods (2016). *Digital image processing*. Pearson education india.

Jain, A. K. (1989). Fundamentals of digital image processing. Prentice-Hall, Inc.

Digital Image Processing, Dr. Sanjay Sarma, SK Kataria& Sons

NPTEL, IITKGP

Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, McGrawHill

*****

UNIT: 2
IMAGE SEGMENTATION-I

Unit Structure:

2.1 Introduction

Thresholding is one of the segmentation techniques that create a grayscale image. Here in this unit we will discuss the about the thresholding technique used in image segmentation. We also discuss how the basic intensity thresholding related to the width and depth of the histogram valleys. Here in this unit we will discuss the basic global thresholding segmentation technique and optimum global thresholding using Otsu's method. We will also discuss here the edge linking and fundamental approaches of edge linking technique.

2.2 Objectives

After going through this unit learner will able to

- *understand* the thresholding segmentation technique,
- *learn* how thresholding related to the width and depth of a histogram,
- *understand* the concept of basic global thresholding,
- *understand* the Otsu's method,
- *learn* the concept of edge linking,
- *understand* the fundamental approaches of edge linking.

2.3 Thresholding

Thresholding technique is used to create a binary image from a grayscale image. In this technique we have to set a threshold value and then turning every pixel in the image to either white or black according to the intensity value of that threshold. For a grayscale image each pixel intensity is a value between 0 (black) and 255 (white). Now we have to set threshold value and this threshold value can be between 0 and 255.For applying this threshold value if the pixel intensity is greater than the threshold, then set the pixel value to 255 (white). Again if the pixel intensity value is less than or equal to the threshold, then set the pixel value to 0(black). So the ultimate result is a binary image where all pixels are either black or white.

2.3.1 The basics of intensity thresholding

Suppose a function f(x,y) composed of light objects on a dark background in such a way that the object and background pixels have intensity values that grouped into two modes. The intensity histogram in Figure 1(a) corresponds to the function *f(x,y)*. If the threshold T separate these two modes, then any point (x,y) in the image at which *f(x,y)* > T is called the object point, otherwise the point is called backgroung point. The segmented image, g(x,y), is given by

$$g(x,y)=g(x,y)=\begin{cases}1 & if\ f(x,y) > T\\ 0 & if\ f(x,y) \le T\end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Stop to Consider

We follow convention in using 0 intensity for the background and 1 for object pixels, any two distinct values can be used for the equation (1)

The process given in this equation (1) is referred to as global thresholding, when T is a constant applicable over an entire image. When T is changes, it is termed as variable thresholding. To denote the variable thresholding sometimes it termed as regional thresholding or local thresholding depending upon the value of T at any point (x,y) in an image depends on the properties of neighborhood. The variable thresholding termed as dynamic or adaptive thresholding if the value of T is depends upon the spatial co-ordinates (x,y) themselves.

In Figure 1(b) shows a histogram with three modes. This histogram is more complex in comparison to the previous one. Here the multiple thresholding classifies a point (x,y) as belonging to the background if $f(x,y) \le T_1$, to one object class if $T_1 < f(x,y) \le T_2$ and to the other object class if $f(x,y)>T_2$.

Then the segmented image is given by

$$g(x,y) = \begin{cases} a & if\ f(x,y) > T_2\\ b & if\ T_1 < f(x,y) \le T_2\\ c & if\ f(x,y) \le T_1\end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Here a,b,c are three intensity values.

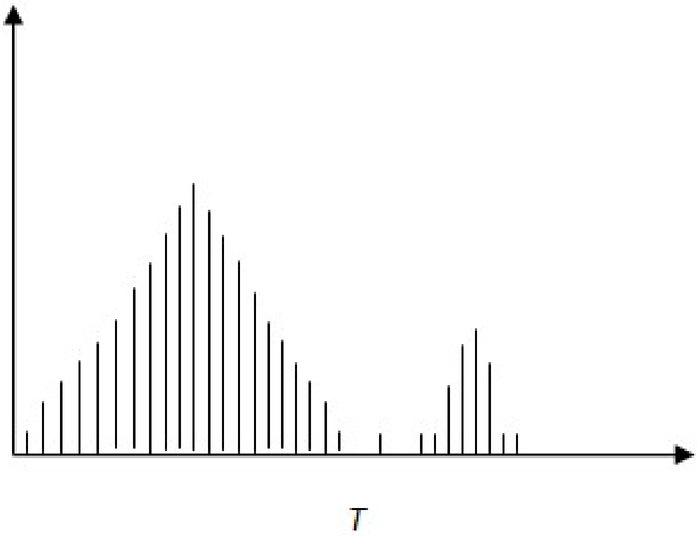


Figure 1(a) Histogram partitioned by a single threshold

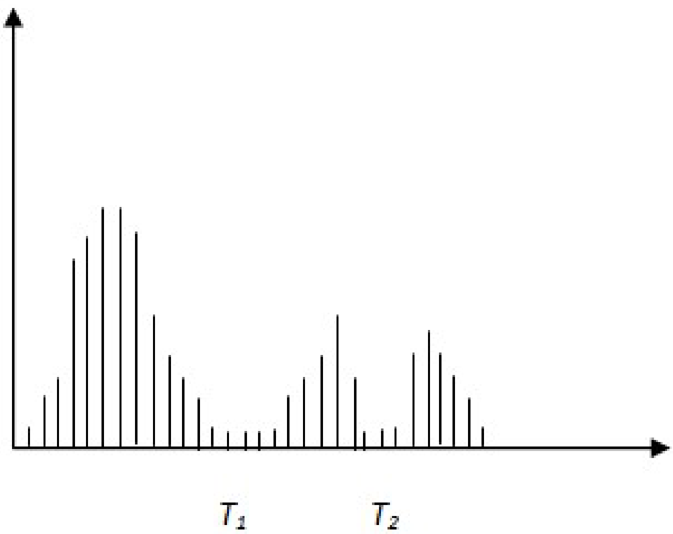


Figure 1 (b) Histogram partitioned by a dual threshold

From the above we can conclude that the success of intensity thresholding is directly related to the width and depth of the histogram valleys.

2.3.2 Basic Global Thresholding

Basic global thresholding is popular and widely used image segmentation technique in image processing. If the intensity distribution of objects and background pixels are sufficiently distinct, it is use a global threshold over the entire image. In most of the applications an algorithm is capable of arranging automatically the threshold value for each image is required. The following iterative algorithm (Gonzalez & Woods, 2016) can be used for this purpose

- For the global threshold *T,* select an initial estimate.
- Segment the image using the global threshold T in Equation (1). This will produce two groups of pixels: P_1 consisting of all pixels with intensity values > *T*, and P_2, respectively.
- Compute the average (mean) intensity values x_1 and x_2 for the pixels in P_1 and P_2 , respectively.
- Compute a new threshold value:

$$T = \frac{1}{2}(x_1+x_2)$$

- Repeat Steps 2 through 4 until the difference between values of *T* in successive iterations is smaller than a predefined parameter ΔT.

The above algorithm is work properly if there is a distinct or clear valley between the modes of the histogram related to objects and background. The parameter ΔT is used to control the number of iterations. The algorithm performs where there is larger ΔT. The average intensity of the image is a good choice for setting the value of *T*.

2.3.3 Optimum Global Thesholding Using Otsu's Method

The main objective of thresholding is to minimize the average error occurred in assigning pixel to two or more classes. It may be viewed as statistical-decision theory problem which have an elegant closed-form solution known as the Bayes decision rule. The solution is based on two parameters, one is probability density function (PDF) and another one is probability that each class occurs in an application. But the estimation of PDFs is not trivial matter and in that situation the problem is solved by making workable assumptions about the form of PDFs. Otsu's method is a popular and effective technique for automatically determining the best threshold value to segment of image into foreground and background. Otsu's method has the important property and it can be easily obtainable using 1-D array. The details Otsu's method is explained by (Gonzalez & Woods, 2016) as follows

Let {0,1,2,………..,L-1} denote the L distinct intensity levels in a digital image of size M×N pixels. Let n_i denote the number of pixels with the intensity i. Then the total number of pixels

MN is defined by MN= $n_0 + n_1 + n_2 + n_3 + - - - + n_{L-1}$. The components of normalized histogram is $p_i = n_i / MN$ and from which it follows that

$$\sum_{i=0}^{L-1} p_i = 1 \qquad p_i \geq 0 \ \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Suppose we select the threshold T (k) =k, 0 < k < L – 1, and using it to the threshold the input image in to two classes C_1 and C_2.

Here C_1 consists of all the pixel in the image with intensity values in the range [0,k] and C_2 consists the values in the range [k + 1, L – 1]. The cumulative sum can be expressed as

$$P_1(k) = \sum_{i=0}^{k} p_i \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

If we set k = 0, the probability of class C_1 having any pixel assigned to it is zero.

Similarly, the probability of class C_2 is

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (5)$$

The mean intensity value of the pixels assigned to the class C_1 is

$$m_1(k) = \sum_{i=0}^{k} iP(i/C_1)$$

$$= \sum_{i=0}^{k} iP(C_1/i)P(i)/P(C_1) \ \ldots\ldots\ldots\ldots\ldots\ldots\ (6)$$

$$= 1/P_1(k) \sum_{i=0}^{k} ip_i$$

The term $P(i/C_1)$ in the first line of the equation (6) is the probability of value i, given that i comes from the class C_1.

The second line in the equation follows from Bayes' formula which can be explained as

$$P(A/B) = P(B/A)P(A)/P(B)$$

The third line follows from $P(C_1/i)$, that is the probability of C_1 given i, is **1** because here we only dealing with the values of i which is from the class C_1.

$P(i)$ is the probability of the ith value, which is simply the ith component of the histogram, p_i. Finally, $P(C_1)$ is the probability of class C_1 which follows from the equation (4) and is equal to $P_1(k)$.

Similarly, the mean intensity value of the pixels assigned to class C_2 is

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i/C_2)$$

$$= 1/P_2(k) \sum_{i=k+1}^{L-1} ip_i \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7)$$

The cumulative mean that is average intensity up to level *k* is given by

$$m(k) = \sum_{i=0}^{k} ip_i \text{ .... } (8)$$

The average intensity of the entire image that is the global mean is obtained by the following equation

$$m_G = \sum_{i=0}^{L-1} ip_i \text{ ....}(9)$$

The validity of the following two equations can be verified by direct substitution of the preceding result

$$P_1 m_1 + P_2 m_2 = m_G \text{ ....}(10)$$

and

$$P_1 + P_2 = 1 \text{ ....}(11)$$

Here we have omitted the *ks* term temporarily in favor of notational clarity.

For evaluating the goodness of the threshold at the level k we use the normalized dimension of metric as follows

$$\eta = \frac{\sigma_B^2}{\sigma_G^2} \text{ ....}(12)$$

Where $\sigma^2\ G$ is the global varience and it can be expressed as

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i \text{ ....}(13)$$

And $\sigma^2\ B$ between-class variance is defined as follows

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \text{ ....}(14)$$

The above equation (14) can also be written as

$$\sigma_B^2 = P_1 P_2 (m_1 - m_2)^2$$

$$= \frac{(m_G P_1 - m)^2}{P_1(1 - P_1)} \text{ ....}(15)$$

The first line of the equation (15) is obtained from the equations (14), (10) and (11). The second line follows from the equations (5) through (9). The second line expression is more efficient in computational purpose because the global mean, m_G is computed only once so only two parameters m and P_1 are need to be calculated for any value of k.

The term σ_B^2will indicate that between the class variance is a measure of separability between the classes m_1 and m_2.

Because σ_G^2 is a constant, this means η is also a measure of separability and maximizing this metric is equivalent to maximizing σ_B^2 .

The equation (12) assumes implicitly that $\sigma_G^2 > 0$. This variance can be zero only when all the intensity levels in the image are the same, this implies the existence of only one class of pixels. This means that η = 0 for a constant image, since the separability of a single class from itself is zero.

Now re-introducing k, we have the final results as follows

$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$ ……………………………………..(16)

and

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[(1 - P_1(k)]} \text{ ………………..(17)}$$

Then, the optimum threshold is the value of k^* , which maximizes $\sigma^2 B(k)$

$\sigma_B^2(k^*) = max_{0 \le k \le L-1} \sigma_B^2(k)$ …………………………..(18)

Therefore the above equation (18) will evaluate the k^* for all the integer values of k and selecting that value of k that yielded the maximum σ_B^2 (k). If the maximum exists for more than one value of k , then taking the average value of k for which σ_B^2 (k) is maximum.

Once k^* has been obtained, the input image f(x,y) is segmented as

$$g(x,y) = \begin{cases} 1 & if\ f(x,y) > k^* \\ 0 & if\ f(x,y) \leq k^* \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(19)$$

Here x = 0, 1, 2 ,………………, M -1.

and y= 0,1,2,……………….., N-1.

Here all the quantities required to evaluate equation (19) are obtained from the histogram of *f(x,y)*.

The normalized metric η , evaluated at the optimum threshold value, $\eta(k^*)$. This optimum threshold is used to requiring a quantitative estimate of the separability of classes. This will give an idea of the ease of thresholding a given image. The range of values is as follows

$$0 \leq \eta(k^*) \leq 1$$

The lower bound is achieved only by images with a single, constant intensity level and the upper bound is attainable by two values which is equal to 0 and L – 1.

Otsu's algorithm can summarized as follows

1. In the first step compute the normalized histogram of the input image. Denote the components of the histogram by, p_i , $i=0,1,2,\ldots,\ldots,L-1$.

2. Now compute the cumulative sums, $P_1(k)$, for $k = 0,1,2,\ldots\ldots\ldots\ldots..,L-1$, using the equation (4).
3. Compute the cumulative mean means, $m(k)$, where $k = 0,1,2,\ldots\ldots\ldots\ldots..,L-1$,, using the equation (8).
4. In this step, compute the global intensity mean, m_G , using the equation (9).
5. Compute between-class variance, $\sigma_B^2(k)$ for the values of $k = 0,1,2,\ldots\ldots\ldots,L-1$ using the equation (17)
6. Here in this step, obtain the Otsu threshold which is designated as k^* , as the value of k for which $\sigma_B^2(k)$ is maximum. If the maximum value is not unique, then averaging the values of k corresponding to the various maximum values to obtain k^*.
7. Finally obtain the separability measure , η^* , by evaluating the equation (16) at $k = k^*$

Check your Progress – I

1. State True or False
   (i) Thresholding technique creates a grayscale image.
   (ii) Grayscale image each pixel intensity value is between 0 and 255.
   (iii) Thresholding is not related to the width and depth of the histogram valleys.
   (iv) The main objective of thresholding is to maximize the average error in the pixel.
   (v) Otsu's method is effective technique for automatically determining the best threshold value.

2.4 Edge Linking

The process that connects detected edge pixels into continuous and meaningful edge boundaries in known as edge linking. After edge detection an image typically contain some discrete image pixels that may fragmented or noisy. The main aim of edge linking is to assemble these noisy edge pixels into meaningful edge region boundaries. The fundamental approaches of edge linking techniques are as follows

2.4.1 Local processing

Local processing requires knowledge about the edge point in a local region. In this approach, for linking edge points is to analyze the properties of pixels in a small neighborhood. All points those are matching or similar according to the predefined criteria are linked edge of pixels that share common properties used according to the specified criteria.

In this kind of analysis for establishing the similarity there are two principle properties one is magnitude and other one is the direction of the gradient vector.

Let S_{xy} , denote the set of a neighborhood and the centre at the point (x,y) in an image. The edge pixel with the corresponding *(s,t)* in S_{xy} is similar in magnitude to the pixel at (x,y) if the following expression hold

$$|M(s,t) - M(x,y)| \leq E \quad \ldots\ldots\ldots\ldots\ldots\ldots(20)$$

Where E ia s positive threshold

The direction angle of the gradient vector is given by the following

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(21)$$

An edge pixel with coordinates *(s,t)* in S_{xy} has an angle similar to the pixel at (x,y) if

$$|\alpha(s,t) - \alpha(x,y)| \leq A \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(22)$$

The pixel with coordinates*(s,t)* in S_{xy} is linked to the pixel at (x,y) if both the criteria that is the magnitude and direction are satisfied. This process is repeated at every location of the image. This is computationally expensive because all the neighbors of every point have to be checked.

2.4.2 Regional processing

The location of region of interest in an image can be determined and it implies that the knowledge is available of the regional membership of a pixel, corresponding to the edge of the image. In this situation we can use the regional processing technique for linking the pixels on a regional basis. One approach to this type of processing is functional approximation where we fit 2-D curve to a known point. For this purpose we can take polygonal approximation which is simple and it can capture the essential shape feature of a region while keeping the representation of the boundary of the image.

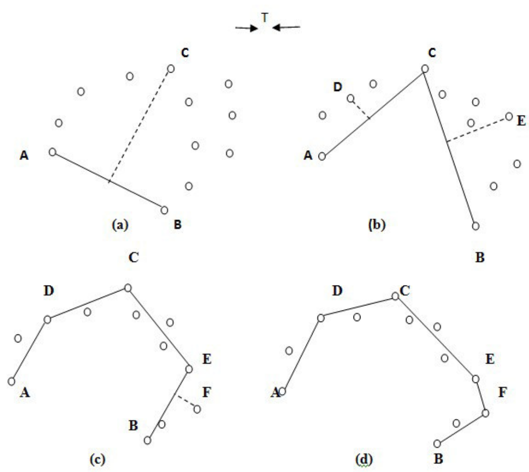


Figure 2 Iterative process of polygon fit algorithm

The Figure 2 shows the iterative process of polygon fit algorithm where a set of points representing an open curve and the end points are labeled as A and B. These two points A and B are the vertices of the polygon. We begin for computing parameters of a straight line passing through A and B. Now we compute the perpendicular distance from all other points in the curve to the line connecting the two points A and B. Select the point which is situated at the maximum distance from the line. Finally we have obtained a regional approximation to the curve which is shown in the Figure 2(d).

If this distance is exceeds the specified threshold value T then the corresponding point is declared as the vertex. This is shown in the Figure 2 (a). Now two lines are established as A to C and C to B. Now distance from all points between A and C are computed and obtained the point which is in maximum distance from the line AC and taken as the vertex. In the figure 2(b) the vertex is taken as D, if the distance exceeds the specified threshold value. A similar procedure is applied for the CB and the vertex is taken as E. This process is continued until no points satisfy the threshold test.

2.4.3 Global processing using the Hough transform

In the local and regional processing, pixel belonging to individual objects at least partially available. In case of regional processing, it makes sense to link a given set of pixels only if we understand that the pixels are part of the boundary of a meaningful region. In general it is often that we have to work with an unstructured environments, in which all we have is an edge image and there is no knowledge where objects of interest might be in.

In such situation, all pixels are candidates for linking and it is accepted or rejected based on the predefined global properties.

Hough [1962] proposed an alternative approach, commonly referred Hough transform. Hough transform is a mapping from a special domain to a parameter space. For this purpose, consider a point (x_i,y_i) in the xy-plane. The general equation of a straight line in the slope-intercept form, $y_i = ax_i + b$. Infinitely many lines pass through (x_i,y_i), but they all satisfy the equation $y_i = ax_i + b$ where a,b are two variables. If we write the equation as $b = -ax_i + y_i$ and here we considering the ab plane. This ab-plane also called the parameter space. The result of the equation $b = -ax_i + y_i$ form a single line for the fixed pair (x_i,y_i) .The second point (x_j,y_j) also has a line in parameter space . Unless they are parallel, this line intersects the line (x_i,y_i) at some point $(a^/,b^/)$, where $a^/$ is the slope and $b^/$ is the intercept of the line which contain both the lines (x_i,y_i) and (x_j,y_j) in the xy-plane. Actually all the points on this line have lines in parameter space that intersect at the point $(a^/,b^/)$. Figure 3 and Figure 4 are explaining the concept of these.

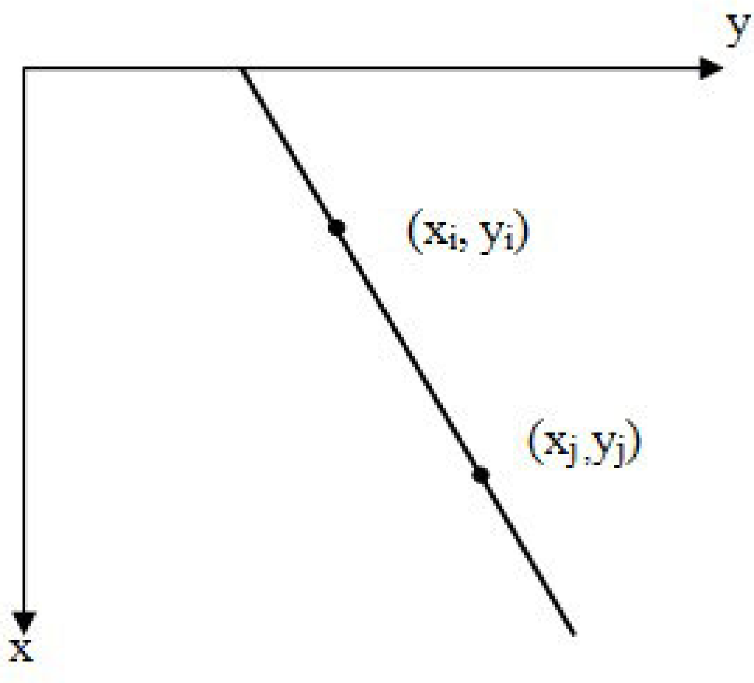


Figure 3 xy-plane

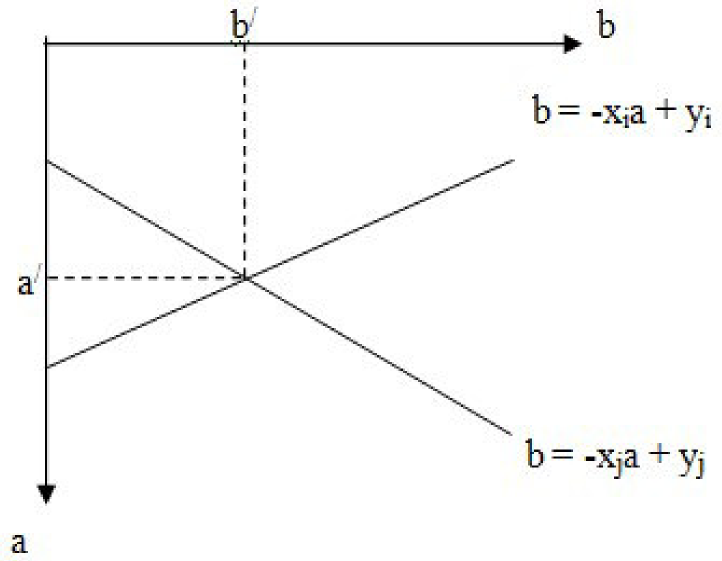


Figure 4 Parameter space.

In the principle, the parameter-space lines corresponding to all points (x_k , y_k) in the xy-plane could be plotted. The principal lines in the xy-plane could be found by identifying points in parameter space where large numbers of parameter-space lines intersect. Now the problem comes when this straight line tries to be vertical, that is parallel to x-axis then the slope "a" tends to be infinity. So in this formulation we cannot take the value of the slope "a". To solve this problem instead of considering the slope intercept form we can make use of the normal representation of a straight line.

The normal representation of a line is given by

$$\rho = x\cos\Theta + y\sin\Theta$$

Where

ρ is the length of the perpendicular drawn on the line from the origin of xy-plane.

Θ is the angle formed by this perpendicular with the x-axis.

The geometrical representation of ρ and Θ can be shown in the figure 4.

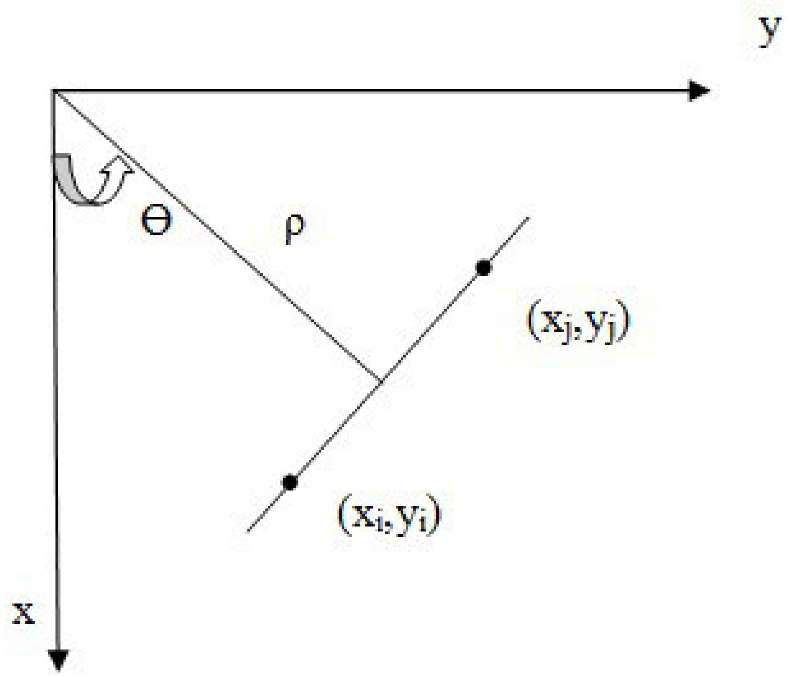


Figure 4 geometrical representations of ρ and Θ

A horizontal line has $\Theta=0^0$ and the vertical line has an angle of $\Theta= 90^0$. For $\Theta=0^0$, ρ is equal to positive x-intercept and for the $\Theta= 90^0$, ρ is equal to negative y-intercept.

Each sinusoidal curve in Figure 5 represents the family of line. These lines are pass through a particular point (x_k,y_k) which in the xy plane. For computational purpose Hough transform, subdivided the ρΘ parameter space, called accumulator cells. In Figure 6 illustrates where (ρ_{min}, ρ_{max}) and $(\Theta_{min}, \Theta_{max})$ has the expected ranges of parameter values $-90^0 \leq \Theta \leq 90^0$ and $-D \leq \rho \leq D$, where D is the maximum distance between opposite corners in an image. Initially these cells are set to zero. Now for every non –background point (x_k,y_k) in the xy-plane, we suppose Θ equal each of the allowed subdivision values on the Θ-axis .It will solve for the corresponding ρ using the equation $\rho = x_k \cos \Theta + y_k \sin \Theta$. Then the ρ values are taken by rounding off the nearest allowed cell value along the ρ axis. If a choice of Θ_p results in solution ρ_q, then we let A (p,q) = $A(p,q)+1$. Finally a value of p points in the xy-plane lie on the line $x\cos \Theta_j + y \sin \Theta_j = \rho_i$. The number of subdivisions in the ρΘ-plane determines the accuracy of the colinearity of these points.

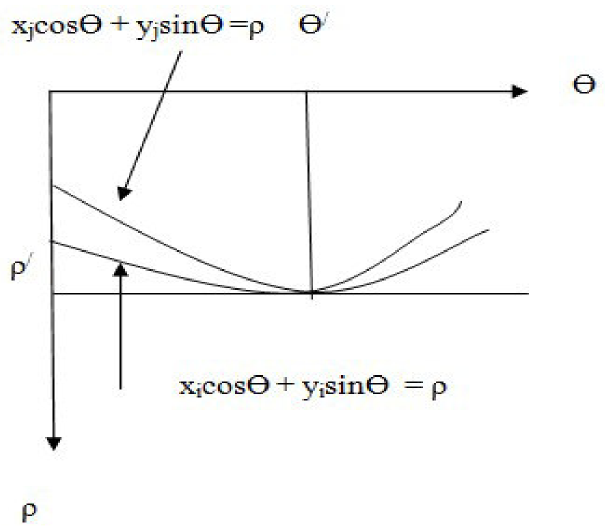


Figure 5. Sinusoidal curves in the ρΘ-plane

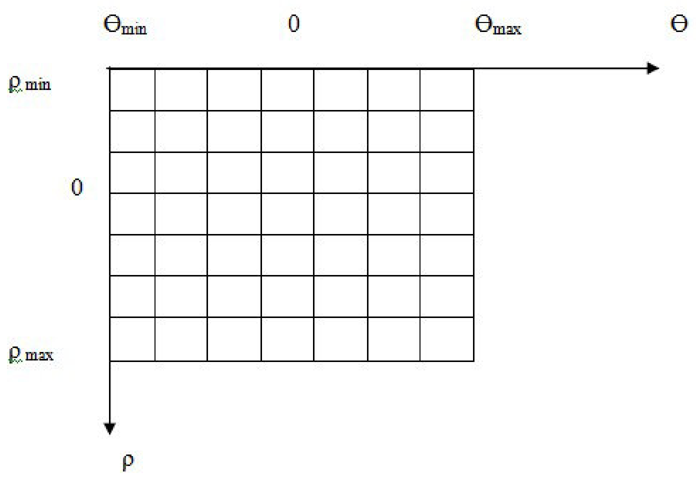


Figure 6. Division of the ρΘ-plane into accumulator cells.

Check Your Progress II

2. State True or False

(i) The main aim of edge linking is to assemble the noisy edge pixels into meaningful edge region boundaries.

(ii) Local processing is the fundamental approaches of edge linking techniques.

(iii) Regional processing technique for linking the pixels on a local basis.

(iv) In global processing, pixel belonging to individual objects at least partially available.

(v) Hough transform is a mapping from a special domain to a parameter space.

2.5. Summing up

- Thresholding is one of the segmentation techniques that create a grayscale image.
- For a grayscale image each pixel intensity is a value between 0 (black) and 255 (white).
- The success of intensity thresholding is directly related to the width and depth of the histogram valleys.
- Basic global thresholding is popular and widely used image segmentation technique in image processing.
- The main objective of thresholding is to minimize the average error occurred in assigning pixel to two or more classes.
- Otsu's method is a popular and effective technique for automatically determining the best threshold value to segment of image into foreground and background.

- The process that connects detected edge pixels into continuous and meaningful edge boundaries in known as edge linking.
- The main aim of edge linking is to assemble these noisy edge pixels into meaningful edge region boundaries.
- Local processing requires knowledge about the edge point in a local region.
- In local processing, for linking edge points is to analyze the properties of pixels in a small neighborhood.
- The location of region of interest in an image can be determined and it implies that the knowledge is available of the regional membership of a pixel, corresponding to the edge of the image.
- In the local and regional processing, pixel belonging to individual objects at least partially available.
- Hough transform is a mapping from a special domain to a parameter space.

2.6. Answer to Check your Progress

1. (i) True (ii) True (iii) False (iv)False (v)True

2. (i)True (ii)True (iii) False (iv) False (v) True

2.7. Possible Questions

(1) What is the principle of thresholding?

(2) Explain how thresholding related to a histogram.

(3) Explain the basic global thresholding.

(4) Explain the optimum global thesholding using Otsu's method.

(5) What is Edge linking?

(6) Explain the local processing approach of edge linking.

(7) Explain the regional processing approach of edge linking.

(8) Explain the Global processing using the Hough transform.

2.8. References and Suggested Readings

Gonzalez & Woods (2016). *Digital image processing*. Pearson education india.

Jain, A. K. (1989). Fundamentals of digital image processing. Prentice-Hall, Inc..

Digital Image Processing, Dr. Sanjay Sarma, SK Kataria& Sons

NPTEL, IITKGP

Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, McGrawHill

*****

UNIT: 3
IMAGE SEGMENTATION II

Unit Structure:

3.1 Introduction

Segmentation is the process of separating the image in one or more regions. There are mainly three method of segmentation. One is pixel based segmentation, second one is the edge based segmentation and the third one is the region based segmentation. Here in this unit we will discuss about the different techniques of region based segmentation. We will also discuss here the image texture analysis and different approaches of defining texture of image.

3.2 Objectives

After going through this unit learner will able to

- *understand* the concept of Region-based segmentation,

- *learn* region how growing technique helps to segmenting images,
- *understand* the concept of region splitting and merging,
- *understand* the concept of texture analysis,
- *learn* about the different approaches of defining texture of the image,
- *learn* about the different method of texture segmentation an its applications.

3.3 Region-based Segmentation

The basic objective of region-based segmentation is to partition an image. In image processing, the region based segmentation is based on the approaches that divide the region in some important criteria, such as pixel intensity, color and texture. The region based approaches based on the distribution of pixel properties. There are different techniques for region based segmentation. These techniques are as follows

3.3.1. Region Growing

The region growing is a process that groups pixels into some larger regions based on some predefined criteria for growth. The basic approach is to start with a set of seed points and from these the region is grown by adding or appending to each seed those neighboring pixels that meet the similarity criteria. Based on the nature of the problem, the selection of the starting point may be one or more. The selection of similarity criteria depends on the availability of the image data. One problem of region growing is to formulation of a stopping rule. When no more pixels satisfy the criteria for inclusion in a region then the region growth must be stopped. In the history of the region growth do not include some criteria such as intensity value, texture and color that are local in nature. The criteria that increase the power of region growth algorithm utilize the concept of size, likeness between the base pixel and grown pixel. Let *f(x,y)* denote an input image and *R(x,y)* denote the seed array containing 1s at the location at the seed points and 0s in elsewhere. Let Q denote a predicate to be applied at each location *(x,y)*. Suppose the size of the arrays f and R is in same size. A basic region-growing algorithm (Gonzalez & Woods, 2016) based on 8-connectivity may be described as follows

(i) Find all the component *R(x,y)* and reduce each connected component to one pixel:

label all such pixels found as 1. All other pixels in *R* are labeled as 0.

(ii) From an image f_Q such at each point, let f_Q (x,y) =1 if the input image satisfies the given predicate, Q at those coordinates; otherwise, let $f_Q(x,y)=0$.

(iii) Let *p* be an image formed by appending to each seed point in *R* then all the 1-valued points in f_Q that are 8-connected to that seed point.

(iv) Label each connected component in *p* with a different region label. This is the final image obtained by region growing.

The above steps can be explained as follows

1. Seed selection: The algorithm starts by selecting one or seed points in the image. These seed point usually chosen by some predefined criteria.
2. Region Criteria: The similarity criteria determine which neighboring pixels can be added to the region.

3. Region Expansion: The algorithm iteratively checks the neighboring pixels of the active region. If the similarity criteria satisfied by the neighboring pixel then it is added to the region and process continues from that pixel.
4. Stopping Condition: When no more neighboring pixels satisfy the similarity criteria or the predefined threshold is reached the process is then stop.

Example of Region growing

Suppose a gray-scale image and we want to segment a particular object in the image .Now we will follow the steps of region growing for segmenting the image

Consider a 2D array where each pixel has an intensity value from 0(black) to 255(white). For the simplicity we consider the object, which we want to segment has pixel intensity of 200. The background has pixel intensities of 50.

Let us consider the image as follows

50,	50,	50,	50,	50,	50,	50
50,	200,	200,	200,	50,	50,	50
50,	200,	200,	200,	50,	50,	50
50,	50,	50,	50,	50,	50,	50

Step 1: Seed point selection

Here in our example we choose a seed point inside the object to start growing the region. Let us consider the pixel at the location (1,1) with a value of 200.

Step 2: Similarity Criteria

The next step is to define similarity criteria. Here we set a threshold of ±10, this means that any neighboring pixel with an intensity value between 190 to 210 will be added in the region.

Step 3: Region growing

In this step, we start from the seed point, we check it's neighboring pixels as follows:

The seed point (1,1) has a value of 200, which matches the threshold. We check the 4-connected neighbors (considering 4-connectivity). For checking the 4-connected neighbor, we consider the left, right, above and below pixel.

Pixel at (0,1) = 50 [which is not added, that is it does not match the criteria]

Pixel at (2,1) = 200 [added to the region]

Pixel at (1,0) = 50 [not added]

Pixel at (1,2) = 200[added]

Next we expand the region to the newly added pixels and repeat the process. For the pixel (2,1) , It's neighbors are checked. Any neighbor has a value 190 and 210 are added in the region. The process will continue until all connected pixels within the threshold are included in the region.

Step 4: Stopping conditions:

The algorithm will stop, when there are no more neighboring pixels within the threshold to add to the region. In our example, the region growing process is stop, after all the pixels with values of 200 are included. The segmented object is the group of pixels where the region growing has successfully expanded. The final segmented region after region growing is as follows

-,	-,	-,	-,	-,	-,	-
-,	200,	200,	200,	-,	-,	-
-,	200,	200,	200,	-,	-,	-
-,	-,	-,	-,	-,	-,	-

Advantages:

(i) The region growing process is easy to implement and understanding.

(ii) User can put the seed value manually which gives more accurate result in segmentation.

Disadvantages:

(i) Noise: Region growing is much more sensitive to the noise. If the similarity criteria not defined properly, the region growing may grow unwanted region.

(ii) Selection of seed point: The success of the algorithm is depends upon the selection of the seed point.

(iii) Computationally Intensive: In region growing, it needs to check the pixels iteratively. So for a large image, It is much more slow.

3.3.2. Region splitting and merging:

Region splitting and merging is another image segmentation technique in which image is initially subdivided into a set of arbitrary disjoint regions and then merges and split the region on the basis of certain homogeneity criteria.

Let R represent the whole image region and select a predicate Q. One approach for segmenting R is to subdivided successfully into some smaller and smaller quadrant region so that for any region R_i , $Q(R_i)$= TRUE. If Q(R) =FALSE, then subdivided image into quadrant and so on. If the value of Q is False, to any quadrant, we then subdivided that quadrant into some sub-quadrants and so on. This splitting technique is called the “quadtrees” that is trees in which each node has exactly four descendants. The partitioned image and the corresponding quadtrees for R are shown in the Figure 1 and Figure 2 respectfully.

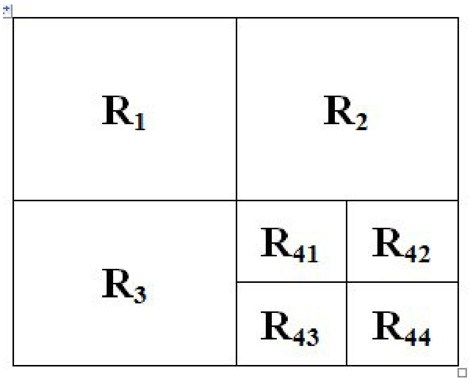


Figure 1. Partitioned image

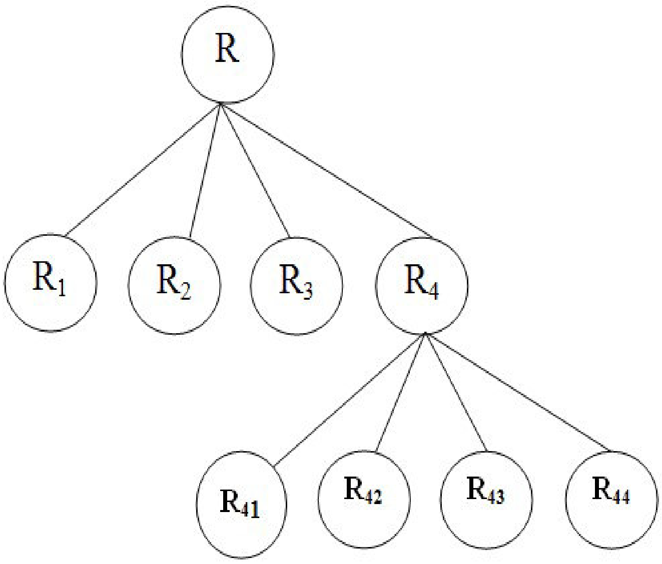


Figure 2. Corresponding quadtree.

The above discussion can be summarized as follows

1. For any region R_m split the region into four quadrant for which $Q(R_{m)}$=FALSE.
2. When splitting is not possible for further, merge any adjacent region Rn and R_k for which $Q(R_n \cup R_k)$ =TRUE.
3. Stop when further merging is not possible.

Let us take the example shown in the Figure 3 for applying the splitting and merging technique

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

Figure 3. 8×8 image in one region R

Here we assume that the threshold value is ≤ 3.

Step :1 first we take the entire image as one region R. Now we have to find out the maximum and minimum values from R.

For the region R the maximum value = 7

the minimum value = 0

maximum value– minimum value = $7 \leq 3$ therefore the condition is false. So we have to split the region R. R can be divided into the 4-quadrant as R1, R2, R3 and R4 as follows. Now we have to check the regions R1, R2, R3 and R4 can be split or not.

	6	5	6	6	7	7	6	6	
	6	7	6	7	5	5	4	7	
R1	6	6	4	4	3	2	5	6	R2
	5	4	5	4	2	3	4	6	
	0	3	2	3	3	2	4	7	
	0	0	0	0	2	2	5	6	
R3	1	1	0	1	0	3	4	4	R4
	1	0	1	0	2	3	5	4	

Figure 4. 4-quadrant of region R

Step 2: Consider the region R1 and find out the maximum and minimum values from this region.

The maximum value = 7

Minimum value = 4

Now maximum – minimum = 7-4=3 ≤ 3, condition is true. So, no further split in region R1.

Step 3: Consider the region R2 and find out the maximum and minimum value

Maximum value = 7

Minimum value = 2

Now maximum value – minimum value = 7-2 = 5 ≤ 3. The condition is false. So the region R2 split again. The R2 region is splitting into 4-quadrant as R21,R22,R23 and R24.Now we have to check whether further splitting technique is operated on four regions of R2. For this purpose consider the region R21

Maximum value = 7

Minimum value =5

Maximum value – Minimum value = 7-5= 2 ≤ 3. So, no further split in the region R21.

For the region R22

We get the maximum value – minimum value = 7 - 4 = 3 ≤ 3. So, no further split in the region R22.

For the region R23, we get maximum value – minimum value = 3-2 = 1≤ 3. So, no further split in the region R23.

For the region R24, we get maximum value – minimum value = 6-4 = 2 ≤ 3. So, no further split in the region R24.

Step 4: Considering the region R3, we get maximum value – minimum value = 3-0=3 ≤ 3, so the region R3 cannot be split further more.

Step 5: Considering the region R4, we get the maximum value – minimum value = 7-0= 7 ≤ 3 ,so the condition is false. So, the region R4 split again. The R4 region is splitting into 4-quadrant as R41,R42, R43 and R44. Now we have to check whether further splitting technique is operated on four regions of R4.

Considering the region R41, we get the maximum value – minimum value = 3-2 =1≤ 3.So, no further split in the region R41.

Considering the region R42, we get the maximum value – minimum value = 7-4=3 ≤ 3.So, no further split in the region R42.

Considering the region R43, we get the maximum value – minimum value = 3-0=3 ≤ 3. So, no further split in the region R43.

Considering the region R44, we get the maximum value – minimum value = 5-4=1≤ 3. So, no further split in the region R44.

After applying the splitting operation in the all four region R1, R2, R3 and R4, we have got total 10 regions in the resultant image as shown in the Figure

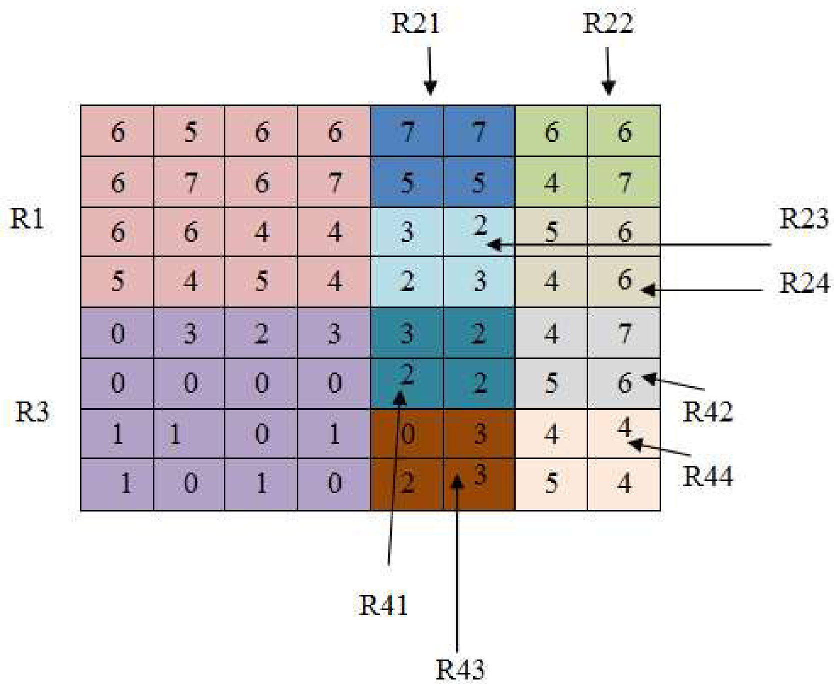


Figure 5. Region R is splitting into ten different regions

Since the splitting process is stop. Now we are going to merge the region, and merging is possible if the existence regions sharing their common properties. In this particular example we have to check the following conditions for merging two regions.

Maximum (region 1) – Minimum (region 2) ≤ 3 at the same time

Maximum (region 2) – Minimum(region 1) ≤ 3

<u>Considering the two regions R1 and R3</u>

Maximum value (R1) – Minimum value(R3) = 7- 0 = 7 $\leq$ 3, condition is false

Maximum value (R3) – Minimum value (R1)= 3-4 = 1 ≤ 3(magnitude of -1), condition is true

Overall condition is false, we cannot merge region R1 and R3

Considering the two regions R1 and R21

Maximum value (R1) – Minimum value (R21) = 7- 5 = 2 ≤ 3, condition is true

Maximum value (R21) – Minimum value (R1)= 7-4 = 3 ≤ 3 , condition is true

Thus two regions R1 and R21 can be merged.

Considering the two regions R1 and R23

Maximum value (R1) – Minimum value (R23) = 7- 2 = 5 ≤ 3, condition is False

Maximum value (R23) – Minimum value (R1) = 3- 4= 1≤ 3, condition is true

We cannot merge region R1 and R23

Considering the two regions R21 and R22

Maximum value (R21) – Minimum value (R22) = 7- 4 = 3 ≤ 3, condition is true

Maximum value (R22) – Minimum value (R21) = 7- 5 = 2 ≤ 3, condition is true

The regions R21 and R22 can be merged

Considering the two regions R22 and R24

Maximum value (R22) – Minimum value (R24) = 7- 4 = 3 ≤ 3, condition is true

Maximum value (R24) – Minimum value (R22) = 6- 4 = 2 ≤ 3, condition is true

The regions R22 and R24 can be merged

Considering the two regions R23 and R24

Maximum value (R23) – Minimum value (R24) =3- 4 = -1=1 ≤ 3, condition is true

Maximum value (R24) – Minimum value (R23) = 6-2 = 4 ≤ 3, condition is false.

We cannot merge the regions R23 and R24.

Considering the two regions R41 and R43

Maximum value (R41) –Minimum value (R43) = 3-0=3 ≤ 3, condition is true

Maximum value (R43) – Minimum value (R41) = 3-2=1≤ 3, condition is true

The regions R41 and R43 can be merged.

Considering the two regions R43 and R44

Maximum value (R43) – Minimum value (R44) = 3-4= -1=1≤ 3, condition is true

Maximum value (R44) – Minimum value (R43) = 5-0= 5 ≤ 3, condition is false

We cannot merge the regions R43 and R44.

After completing the splitting and merging technique we get the final region shown in the figure

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

Figure 6. Final region after splitting and merging technique

Check Your Progress –I

1. State True or False

(i) The basic objective of region-based segmentation is to partition an image.

(ii) The region based approaches based on the distribution of pixel properties

(iii) Region growing is not sensitive to the noise.

(iv) A quadtree is a spatial data structure which has four branches attached to the branch point or node.

(v) In region growing, it needs to check the pixels only once.

3.4 Texture analysis:

Image texture is a spatial distribution of gray level intensity values. The basic element of texture is called the texel . It means a group of pixel having homogeneous property. If texel is repeated spatially then we get a particular texture. For image classification and object recognition texture can be combined with the other features like color feature, motion feature shape feature etc. Texture analysis studies the structural patterns in the image and it can be often used in the fields such as medical imaging, remote sensing and computer vision. Image segmentation separated different texture, which helps us to analyze and interpret data in each region easily. Consider an example shown in the Figure 7 which has three different images with same intensity distribution. The images have 50% black and 50% white distribution of pixels

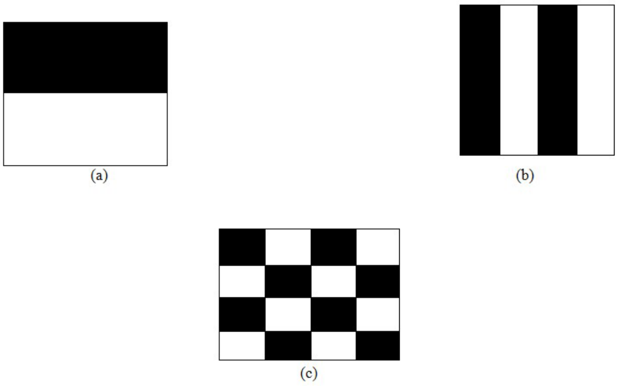
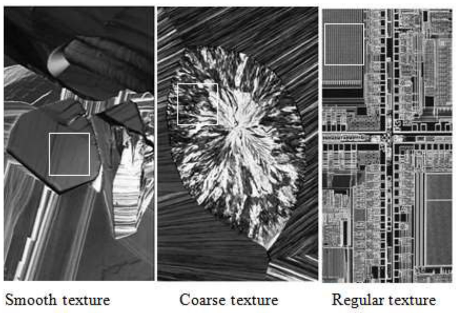


Figure 7. Three different images with different textures.

There are three approaches of defining a texture is

(i) Statistical approach: The statistical approaches produce of characterizations of textures as smooth, coarse, grainy and so on. Texture is a quantitative measure of the arrangement of intensities in a region. This approach is useful when texture primitive are small and extraction of some statistical parameters.

One of the simplest ways to describing texture is to use statistical moments of the intensity of an image. Considering three types of texture one is smooth texture, the second one is coarse texture and third one is the regular texture. In Figure 8 shows these three texture and measures for the images.



Texture	Mean	Standard deviation	R(normalized)	Third moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	-0.105	0.026	5.434
Coarse	143.56	74.63	0.079	-0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

Figure 8.White boxes mark smooth, coarse and regular texture and the texture measures for these images (Gonzalez & Woods, 2016)

These three textures can be mathematically described as below.

We can use statistical moment computed from an image histogram is defined in equation (1)

$$\mu_n(z) = \sum_{i=0}^{L-1}(z_i - m)^n p(z_i) \quad \text{…. (1)}$$

Where m is the mean value of z that is the average intensity defined by

$$m = \sum_{i=0}^{L-1} z_i \; p(z_i) \quad \text{….. (2)}$$

Here we can determine the second moment that is the variance which will give the measure of smoothness.

Variance is shown

$$\sigma^2 = \sum_{i=0}^{L-1}(z_i - m)^2 \, p(z_i) \quad \text{….. (3)}$$

Roughness factor can be defined by the equation (4)

$$R = 1 - \frac{1}{1+\sigma^2} \quad \text{….. (4)}$$

Based on the roughness factor, we will get two types of textures. For R=0, corresponds to the smooth texture and for R=1 corresponds coarse texture.

The third moment also determined and this is designated as the measure of skewness. The skewness parameter is defined by equation (5)

$$\mu_3(z) = \sum_{i=0}^{L-1}(z_i - m)^3 \, p(z_i) \quad \text{….. (5)}$$

From this skewness parameter, we can determine the entropy. The average entropy can be determined by the formula (6).

$$e(z) = -\sum_{i=0}^{L-1} p(z_i)\ \log_2 p(z_i) \qquad \text{..... (6)}$$

The fourth moment is the measure of its relative flatness.

(i) Structural approach: This approach deals with the arrangement of image primitives, such as description of the texture based on regularly spaced parallel lines. Structural description of the texture is the description of the texels and specification of the spatial relationship.

(ii) Spectral approach: This approaches based on the Fourier spectrum. It is mainly used to detect global periodicity in an image by identifying high energy, narrow peaks in the spectrum. Considering three features of the Fourier spectrum which are useful for texture description. First one stated that prominent peaks in the spectrum give the principal direction of the texture patterns. The second one stated that the location of the peaks in the frequency plane gives the fundamental spatial period of the patterns. The third one described that any periodic components using some filtering technique leaves non-periodic image element, which can be described by statistical techniques.

Stop To Consider

Texture primitive can be used to form more complex texture patterns by means of some rules that limit the number of possible arrangements of the primitives.

3.5 Methods of texture segmentation:

There are different methods for segmenting images based on texture are as follows

(1) Statistical Methods:

(i) The statistical properties of pixel values include the first order statics or second-order statics. The first-order statistics includes mean and variance and second order statistics includes co-occurrence matrices.

(ii) Haralick features: Haralick features derived from the Gray-Level-Co-Occurrence Matix (GLCM). These matrices describe how often given intensity gray-level value occurs at some position which is relative to another pixel. The features of GLCM are contrast, correlation, energy, homogeneity etc. which are very much useful for texture segmentation.

(2) Filtering methods: For filter based methods different types of filtering techniques such as Gabor filters, wavelets or Laws's texture energy measures to capture texture patterns. Gabor filter is named after Dennis Gabor, which is basically determines the frequency components of the image with a particular orientation in a localized region around the point of evaluation. Wavelets are useful to identify textures at different scales.

(3) Model-based Methods:

(i) Markov Random Fields (MRFs): It is a statistical model that captures the structure of the texture and is often used in texture segmentation.

(ii) Fractal Models: Fractal dimension is a parameter to characterize roughness in an image. It can be used in texture segmentation.

(4) Clustering-based Methods:

In clustering-based methods includes k-means clustering, mean shift or Gaussian Mixture Models (GMMs) which can be applied to segment textures based on extracted feature vectors. Region growing and merging the regions with similar texture features which are grouped together.

(5) Deep Learning Methods:

This is a popular techniques in image texture segmentation .Convolution Neural Network (CNN) and U-net are the popular deep learning methods that are used in texture-based image segmentation. These methods are used to segmenting the texture of the images by learning automatically its features.

Application of Texture Segmentation

(i) Medical Image Processing: Texture segmentation is very much helpful by distinguishing different types of tissue in MRI or CTscans based on texture. In tumor detection it helps identifying abnormal region from normal tissue by analyzing texture variance in MRI or CTscan. In digital pathology texture segmentation plays an important role for diagnosis of diseases.

(ii) Remote Sensing and Satellite images: Texture segmentation helps to identify different land types. It differentiates between the forest area, urban area, river, road etc. by their corresponding texture images, which enables better land use management. Texture segmentation also helps crop monitoring, classification of seeds, identification of diseases in plants and various sectors in agricultural domain.

(iii) Industrial Inspection: In the manufacturing industries, texture segmentation helps to identify defects on surfaces of products which help improving the quality control of the products. By detecting texture anomalies in fabric industries, texture segmentation helps the measure the quality of fabric products.

(iv) Biometric Systems: In fingerprint recognition texture segmentation is applied in analyzing the ridges and valleys in fingerprints for biometric identification. Texture segmentation also helps iris recognition. It helps to analyzing unique texture patterns in the human iris for secure identity verifications.

Check Your Progress-II

2. State True or False

(i) Image texture is a spatial distribution of gray level intensity values.

(ii) The basic element of texture is called the pixel.

(iii) Structural approaches are based on the Fourier spectrum.

(iv) If texel is repeated spatially then we get a particular texture.

(v) Second order statistics includes co-occurrence matrices.

3.6 Summing up:

- In image processing, the region based segmentation is based on the approaches that divide the region in some important criteria, such as pixel intensity, color and texture.
- The region growing is a process that groups pixels into some larger regions based on some predefined criteria for growth.
- Region splitting and merging is another image segmentation technique in which image is initially subdivided into a set of

arbitrary disjoint regions and then merge and split the region on the basis of certain homogeneity criteria.

- Image texture is a spatial distribution of gray level intensity values.
- The basic element of texture is called the texel .
- Texture analysis studies the structural patterns in the image and it can be often used in the fields such as medical imaging, remote sensing and computer vision.
- Image segmentation separated different texture, which helps us to analyze and interpret data in each region easily.
- Spectral approach is mainly used to detect global periodicity in an image by identifying high energy, narrow peaks in the spectrum.
- The statistical approaches produce of characterizations of textures as smooth, coarse, grainy and so on.
- In clustering-based methods includes k-means clustering, mean shift or Gaussian Mixture Models (GMMs) which can be applied to segment textures based on extracted feature vectors.
- Convolution Neural Network (CNN) and U-net are the popular deep learning methods that are used in texture-based image segmentation.

3.7 Answer to Check your Progress

1. (i) True (ii) True (iii) False (iv) True (v) False
2. (i) True (ii) False (iii) False (iv) True (v) True

3.8 Possible Questions:

(1) What is region based segmentation?

(2) What is region growing?

(3) Explain the basic region growing algorithm based on 8-connectivity.

(4) Explain the advantages and disadvantages of region growing technique.

(5) Explain the region splitting and merging technique with a suitable example.

(6) What is texture?

(7) Explain the statistical approach of defining texture of an image.

(8) Explain the spectral approach of defining texture of an image.

(9) Explain the different methods of texture segmentation.

(10) Explain the application of texture segmentation in different areas.

3.9 References and Suggested Readings

Gonzalez & Woods (2016). *Digital image processing*. Pearson education india.

Jain, A. K. (1989). *Fundamentals of digital image processing*. Prentice-Hall, Inc.

Digital Image Processing, Dr. Sanjay Sarma, SK Kataria& Sons

NPTEL, IITKGP

Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar, McGrawHill

*****

UNIT: 4
IMAGE COMPRESSION I

Unit Structure:

4.1 Introduction

In this chapter we will explore the fundamentals of image compression, delving into techniques like predictive coding and transform coding, which reduce image file sizes by eliminating redundancies. Understanding the trade-offs between compression ratios and image quality is crucial in selecting the right method for different applications, whether it's for fast web delivery or high-fidelity storage

4.2 Unit Objectives

After going through this unit you will be able to understand

- *fundamentals* of Image compression,
- *basic* principle how image compression works,
- *techniques* and algorithms for reducing Image Size,
- *different* image compression models,
- *different* compression methods,
- *image* formats and role of image containers,
- *lossy* and Lossless Formats,
- *different* Compression Standards.

4.3 Fundamentals of Image Compression

Image compression is an essential technology in the digital world. From social media platforms to medical imaging, compressed images allow us to store and transmit visual data efficiently. As the demand for high-resolution images grows, understanding the fundamentals of compression becomes critical.

4.3.1 Basic Principles of How Image Compression Works

At its core, image compression involves reducing the amount of data required to represent an image without significantly compromising its visual quality. This can be achieved by eliminating redundancy—either by reducing repetitive information or by discarding less important data. There are three main types of redundancy that image compression techniques exploit:

1. **Spatial Redundancy**: This refers to redundant data in the spatial domain of the image, where neighbouring pixels are often similar. Techniques like run-length encoding and predictive coding are used to compress spatial redundancy.
2. **Spectral Redundancy**: This type of redundancy exists when multiple colour channels in an image shares similar information. For instance, in RGB images, the red, green, and blue components may carry overlapping data. Compression methods take advantage of this by reducing duplication across channels.
3. **Temporal Redundancy**: In video or sequences of images, many frames are similar to one another, leading to redundancy over time. While not strictly part of still image compression, the concept is relevant when considering how individual frames of a video are compressed.

To manage these redundancies, compression methods use different strategies, including **lossless** and **lossy** techniques.

4.3.2 Techniques and Algorithms for Reducing Image Size:

Several algorithms have been developed to optimize image compression:

- **Run-Length Encoding (RLE)**: This is one of the simplest forms of compression. RLE works by replacing sequences of repeated values (such as colour pixels) with a single value and a count of how many times it repeats. For instance, a sequence of ten white pixels followed by five black pixels might be compressed to "10W5B". Although RLE is not highly efficient for complex images, it works well for simple graphics and is used in formats like BMP and GIF.

- **Predictive Coding**: This method leverages the fact that neighbouring pixels in an image are often similar. Instead of encoding each pixel independently, the algorithm predicts the value of each pixel based on its neighbours, storing only the difference between the predicted value and the actual value. If the prediction is accurate, the difference is small, allowing for more efficient encoding. Predictive coding is often used in lossless compression techniques.
- **Transform Coding**: In transform coding, the image data is transformed into another domain (typically frequency or wavelet domain) where it can be more efficiently compressed. The Discrete Cosine Transform (DCT) is one of the most widely used transformations, particularly in the JPEG compression standard. DCT works by separating the image into parts of differing importance, allowing the less important details (such as high-frequency noise) to be discarded or encoded with fewer bits.

4.3.3 Understanding the Trade-off Between Compression Ratio and Image Quality:

The compression ratio is a measure of how much an image has been compressed. It is defined as the ratio between the original image size and the compressed image size. For example, a compression ratio of 10:1 means that the compressed image is 10 times smaller than the original.

In **lossless compression**, the compression ratio tends to be lower (e.g., 2:1 or 3:1), but there is no loss of quality. The decompressed image is identical to the original image. On the other hand, **lossy compression** can achieve much higher compression ratios (10:1 or more) by discarding some image information. However, as more information is discarded, the image quality degrades, especially if the compression ratio is too high. Visible artifacts, such as blockiness or blurring, can appear if the image is compressed too much.

It is essential to balance compression ratio and image quality, depending on the application. For example, in medical imaging, lossless compression is often required to preserve the integrity of the image. In web applications, lossy compression might be acceptable since file size and load times are critical.

4.4 Image Compression Models

Several models are used to compress images, each with its unique strengths and applications. Understanding these models is vital for choosing the right method based on the requirements of the image and the application.

4.4.1 Exploration of Different Models Used for Image Compression:

1. **Statistical Redundancy Models**: These models focus on eliminating redundancy in the data by assigning shorter codes to frequently occurring values. Common methods in this category include **Huffman Coding** and **Arithmetic Coding**:
    - **Huffman Coding**: This is a form of entropy encoding that assigns shorter binary codes to frequently used symbols and longer codes to less frequent symbols.

Huffman coding is used in many image compression formats, including JPEG.

- **Arithmetic Coding**: This method improves on Huffman coding by representing a sequence of symbols as a single number between 0 and 1. Arithmetic coding can achieve better compression than Huffman coding, especially for images with complex statistical distributions.

2. **Transform-Based Models**: Transform models work by converting the image data into another domain where it can be more efficiently compressed. The most common transform used in image compression is the **Discrete Cosine Transform (DCT)**, which is at the core of the JPEG compression standard.

    - **Wavelet Transforms**: Wavelet-based methods, such as those used in JPEG 2000, offer more flexibility in representing images at different scales and can achieve better compression efficiency, particularly at high compression ratios.

3. **Predictive Models**: Predictive models compress images by exploiting the correlation between neighbouring pixels. For example, in **differential pulse code modulation (DPCM)**, the value of each pixel is predicted based on the value of its neighbours, and only the prediction error is encoded.

4. **Subband Coding**: In this approach, the image is divided into multiple frequency bands using filters. Each band can be compressed separately, allowing more efficient compression of different image components. Subband coding is particularly effective for images with varying levels of detail.

4.4.2 Key Compression Methods (Predictive and Transform Coding):

- **Predictive Coding**: In predictive coding, the value of each pixel is predicted based on its neighbouring pixels. If the predicted value is close to the actual value, the difference (error) can be encoded using fewer bits. This method is commonly used in **lossless compression** schemes like PNG.
- **Transform Coding**: Transform coding is a powerful technique for both lossy and lossless compression. It works by transforming the image into a frequency domain where high-frequency details (which are less visible to the human eye) can be discarded or compressed more aggressively. In the JPEG standard, the **Discrete Cosine Transform (DCT)** is used to convert the image into its frequency components. Higher frequencies are quantized (rounded off) more aggressively than lower frequencies, achieving significant compression without a noticeable loss of quality.

4.4.3 Examples of Popular Compression Models:

1. **JPEG (Joint Photographic Experts Group)**: JPEG is the most widely used image compression model. It is based on transform coding using the DCT. JPEG achieves high compression ratios by discarding some high-frequency components of the image, making it a lossy compression method. However, JPEG allows for adjustable quality settings, allowing users to balance between file size and quality.
2. **JPEG 2000**: This is a more modern image compression standard that uses **wavelet transforms** instead of DCT. Wavelets allow for more flexible representation of image data, leading to better image quality at high compression

ratios. JPEG 2000 supports both lossy and lossless compression, but it is not as widely adopted as the original JPEG format.

3. **PNG (Portable Network Graphics)**: PNG is a lossless image compression format that uses **DEFLATE compression** (a combination of LZ77 and Huffman coding). It is ideal for images that require transparency or when exact reproduction of the original image is required. PNG is commonly used for web graphics, icons, and images with text.
4. **GIF (Graphics Interchange Format)**: GIF is a lossless compression format that is limited to 256 colours. It uses Lempel-Ziv-Welch (LZW) compression, making it suitable for simple graphics like icons and animations. However, due to its colour limitations, it is less suited for high-quality photographic images.

4.5 Image Formats and Containers

Choosing the right image format is essential for optimizing both the storage and display of images. Different image formats offer different levels of compression, quality, and compatibility with software and hardware.

4.5.1 Overview of Common Image Formats:

1. **JPEG (Joint Photographic Experts Group)**: JPEG is the standard format for compressing photographic images. It achieves high compression ratios by using lossy compression. JPEG works best for images with smooth colour transitions, such as photographs, but it is less suited for images with sharp edges or text.

2. **PNG (Portable Network Graphics)**: PNG is a lossless format that supports transparency. It is ideal for images that require exact reproduction, such as logos or images with text. PNG uses DEFLATE compression, which combines LZ77 and Huffman coding for efficient lossless compression.
3. **GIF (Graphics Interchange Format)**: GIF is a lossless format that is limited to 256 colours, making it suitable for simple graphics or animations. It uses LZW compression and is widely supported by web browsers. However, it is not suitable for high-quality images due to its colour limitations.
4. **BMP (Bitmap Image File)**: BMP is a format that stores image data without compression, resulting in large file sizes. It is mainly used in situations where compression is undesirable, such as in some medical or scientific applications. BMP is less commonly used in modern applications due to its inefficiency.
5. **TIFF (Tagged Image File Format)**: TIFF is a lossless format used in professional environments, such as printing, scanning, and medical imaging. It supports a wide range of image data, including multiple layers, transparency, and metadata. TIFF files can be quite large, making them less suited for everyday use, but they are ideal for applications where high image quality is essential.

4.5.2 The Role of Image Containers:

An image container can be thought of as a "wrapper" that holds not only the image data but also additional information such as colour profiles, metadata, and multiple image frames (in the case of formats like GIF or TIFF). Some formats, like **TIFF**, support multiple layers and pages, making them useful for professional applications like desktop publishing and scanning.

4.5.3 Detailed Comparison Between Lossy and Lossless Formats:

1. **Lossy Formats**:
    - **JPEG** and **WebP** are the most common lossy formats. These formats discard some of the image data to achieve higher compression ratios. The degree of loss is often adjustable, allowing users to trade-off between file size and image quality. Lossy formats are best suited for photographs and natural images where some quality degradation is acceptable to save storage space or bandwidth.
2. **Lossless Formats**:
    - **PNG**, **GIF**, and **TIFF** are examples of lossless formats that preserve all of the original image data. These formats are best used in situations where image quality is paramount, such as for archival purposes, professional printing, or images with text. However, lossless formats typically result in larger file sizes than their lossy counterparts.

4.6 Compression Standards

Compression standards are essential for ensuring that compressed images can be stored, transmitted, and viewed consistently across different devices and platforms. These standards define the methods and parameters used for compressing and decompressing images.

4.6.1 Discussion of Established Standards:

1. **JPEG (Joint Photographic Experts Group)**: JPEG has been the industry standard for image compression since the early 1990s. It uses a combination of DCT-based compression and

entropy encoding (Huffman coding) to achieve high compression ratios with minimal quality loss. JPEG is supported by virtually every device and software, making it the go-to format for photographic images.

2. **JPEG 2000**: A more modern image compression standard, JPEG 2000 uses wavelet-based compression instead of DCT. It offers better image quality and compression efficiency than the original JPEG standard, especially at high compression ratios. JPEG 2000 also supports both lossy and lossless compression, but it has not been as widely adopted due to its higher computational requirements and limited software support.
3. **PNG (Portable Network Graphics)**: PNG is the standard for lossless image compression on the web. It is particularly well-suited for images with transparency or sharp edges, such as icons, logos, and images with text. PNG's DEFLATE compression ensures that file sizes are kept manageable without any loss of quality.

4.6.2 Insight into Newer Standards (WebP, AVIF):

1. **WebP**: Developed by Google, WebP is a newer image format that offers both lossy and lossless compression. WebP provides better compression than JPEG or PNG, making it ideal for web applications where load times and bandwidth are important. WebP supports transparency (like PNG) and animations (like GIF), making it a versatile format for modern web design.
2. **AVIF (AV1 Image File Format)**: AVIF is based on the AV1 video codec and offers even better compression efficiency than WebP and JPEG. AVIF is capable of both lossy and

lossless compression and provides superior image quality at smaller file sizes. Although support for AVIF is still growing, it is expected to become a major player in the world of image compression, especially for web applications.

4.6.3 Examination of Industry-Wide Compression Standards:

Image compression standards evolve in response to the growing demand for higher resolution images and faster load times, especially on mobile and web platforms. Standards like **JPEG**, **PNG**, and **WebP** have become ubiquitous, but newer standards like **AVIF** are pushing the boundaries of compression efficiency.

Compression standards ensure that compressed images can be reliably viewed across different devices, software, and platforms. They also guarantee that images compressed on one system can be decompressed on another, ensuring compatibility across the digital ecosystem. As new image formats are developed, they must strike a balance between compression efficiency, image quality, and computational complexity.

Check Your Progress

Fill in the blanks

1. _______________refers to redundant data in the spatial domain of the image, where neighbouring pixels are often similar.

2. _________works well for simple graphics and is used in formats like BMP and GIF.

3.In medical imaging,__________ is often required to preserve the integrity of the image.

4. ________ coding is particularly effective for images with varying levels of detail.

5. An ________________can be thought of as a "wrapper" that holds not only the image data but also additional information such as colour profiles, metadata, and multiple image frames

4.7 Summing Up

1. Image compression involves reducing the amount of data required to represent an image without significantly compromising its visual quality.

2. There are three main types of redundancy that image compression techniques exploit: spatial, spectral and temporal redundancy.

3. The algorithms that have been developed to optimize image compression: Run Length Encoding (RLE), Predictive coding and Transform coding.

4. The compression ratio is a measure of how much an image has been compressed. It is defined as the ratio between the original image size and the compressed image size.

5. In lossless compression, the compression ratio tends to be lower (e.g., 2:1 or 3:1), but there is no loss of quality. The decompressed image is identical to the original image.

6. lossy compression can achieve much higher compression ratios (10:1 or more) by discarding some image information. However, as more information is discarded, the image quality degrades, especially if the compression ratio is too high.

7. The different models used for image compression are Statistical Redundancy Models, Transform-Based Models, Predictive Models and Sub-band coding.

8. Different image formats offer different levels of compression, quality, and compatibility with software and hardware.

9. Compression standards are essential for ensuring that compressed images can be stored, transmitted, and viewed consistently across different devices and platforms.

10. Compression standards ensure that compressed images can be reliably viewed across different devices, software, and platforms. They also guarantee that images compressed on one system can be decompressed on another, ensuring compatibility across the digital ecosystem.

4.8 Answers to Check Your Progress

1. Spatial redundancy
2. RLE
3. Lossless compression
4.Sub-band coding
5. Image container

4.9 Possible Questions:

- What are the key differences between lossy and lossless compression?
- How does predictive coding work, and in what scenarios would you use it?
- What are the advantages of using the Discrete Cosine Transform (DCT) in image compression?

4.10 References and Suggested Readings

1. Gonzalez, R. C., & Woods, R. E. **Digital Image Processing** (4th Edition), Pearson.

2. Pennebaker, W.B., & Mitchell, J.L. **JPEG: Still Image Data Compression Standard**, Springer.

3. Solomon, C. **Fundamentals of Digital Image Processing**, Wiley-Interscience.

*****

UNIT: 5
IMAGE COMPRESSION II

Unit Structure:

5.1 Introduction

In this chapter, we will examine both lossy and lossless compression techniques, focusing on how data can be efficiently reduced while maintaining either exact reproduction (lossless) or acceptable visual quality (lossy). Algorithms like RLE and Huffman coding, as well as the DCT in block-based transform coding, play key roles in achieving efficient compression in formats like JPEG and PNG.

5.2 Objectives

After going through this unit, you will be able to understand

- *lossy* compression and it's different techniques,
- *impact* of lossy compression on image quality,
- *lossless* compression and it's different methods,
- *formats* that preserve the image quality,
- *run* length encoding and Huffman coding,
- *practical* applications of Run length encoding and Huffman coding,
- Block Transform Coding.

5.3 Lossy Compression

Lossy compression is a widely used image compression technique where some amount of image data is discarded to achieve higher compression ratios. This approach sacrifices perfect fidelity to reduce the file size significantly, making it suitable for applications where storage space or bandwidth is limited, and slight quality degradation is acceptable.

5.3.1 Techniques that Discard Some Image Data to Achieve Higher Compression Ratios

In lossy compression, data that is less critical to human perception is discarded or encoded in a way that reduces the overall file size. Some of the most common techniques include:

- **Quantization**: This method reduces the precision of pixel values. In the frequency domain (such as in JPEG), high-frequency components, which represent fine details and noise, are quantized more aggressively, allowing them to be stored with fewer bits or even discarded. This is based on the assumption that the human eye is less sensitive to fine details, particularly in certain colour channels.
- **Chroma Subsampling**: This technique takes advantage of the human eye's greater sensitivity to brightness (luminance) than to colour (chrominance). By reducing the resolution of the colour information (chrominance) while keeping the brightness data (luminance) intact, chroma subsampling can achieve a significant reduction in file size without a noticeable impact on visual quality. Formats like JPEG use chroma subsampling to discard colour information at a lower perceptual cost.
- **Transform Coding (Discrete Cosine Transform - DCT)**: In JPEG, the image is first broken down into 8x8 pixel blocks. Each block is transformed into the frequency domain using the Discrete Cosine Transform (DCT). High-frequency details, which are less noticeable to the human eye, are then quantized and compressed. This technique allows JPEG to achieve impressive compression while keeping the image relatively close to its original appearance.

5.3.2 Examples of Lossy Compression (JPEG, WebP)

1. **JPEG (Joint Photographic Experts Group)**: JPEG is one of the most popular lossy compression standards, particularly suited for photographic images. It uses a combination of DCT and quantization to compress image data. JPEG allows users

to adjust the quality of the compression by controlling the quantization level, making it versatile for different applications where varying trade-offs between file size and image quality are required.

2. **WebP**: Developed by Google, WebP is a newer format that supports both lossy and lossless compression. Its lossy compression uses techniques similar to JPEG but with more efficient entropy coding (VP8 encoding) and better handling of transparency. WebP achieves smaller file sizes than JPEG at comparable visual quality, making it an excellent choice for web images.

5.3.3 The Perceptual Impact of Lossy Compression on Image Quality

The primary impact of lossy compression is **image degradation**. Depending on the level of compression, this degradation can range from imperceptible to highly noticeable. Common artifacts introduced by excessive lossy compression include:

- **Blockiness**: This occurs when the image is divided into blocks (such as 8x8 pixel blocks in JPEG), and compression causes visible discontinuities between blocks. These block artifacts become more pronounced at higher compression ratios.
- **Blurring**: Loss of fine details, especially around edges, is common with high levels of lossy compression. Fine textures and patterns may be smoothed out, causing the image to appear blurry.
- **Colour Banding**: Chroma subsampling can result in visible colour bands where there should be smooth colour transitions.

This is most noticeable in areas with subtle gradients, such as skies.

The goal of lossy compression algorithms is to minimize these perceptual impacts by discarding information that the human visual system is less sensitive to. For instance, slight variations in colour might be discarded, while more attention is given to preserving edge sharpness and brightness, which are more important for image clarity.

5.4 Loss-less Compression

Loss-less compression is a method of image compression where no data is lost during the process. The original image can be perfectly reconstructed from the compressed version. This is crucial in applications where image integrity must be preserved, such as in medical imaging, legal documents, and archival storage.

5.4.1Methods for Compressing Images Without Losing Any Data

Loss-less compression works by reducing the redundancy in image data without discarding any information. Several algorithms are used to achieve this:

- **Run-Length Encoding (RLE)**: This is a simple form of lossless compression that replaces consecutive repeated values with a single value and a count of how many times it repeats. It is highly effective for images with large areas of uniform colour, such as simple graphics or black-and-white line drawings.
- **Lempel-Ziv-Welch (LZW) Compression**: LZW is a dictionary-based algorithm used in formats like GIF and TIFF. It works by identifying repetitive sequences of data and encoding them with shorter representations. For example, a

sequence of pixels that appear multiple times can be stored once with a reference to its position in the dictionary.

- **DEFLATE Compression**: Used in PNG, DEFLATE combines two lossless compression algorithms—LZ77 and Huffman coding. LZ77 replaces repeated sequences of data with references to earlier occurrences, while Huffman coding reduces the overall data size by assigning shorter codes to more frequent pixel values.

5.4.2 Formats That Preserve the Original Quality (PNG, GIF)

1. **PNG (Portable Network Graphics)**: PNG is one of the most widely used lossless compression formats. It uses DEFLATE compression to reduce file size without losing any image data. PNG is especially suitable for images with sharp edges, text, and transparency. Its lossless nature makes it ideal for images where the highest fidelity is required, such as logos and scientific illustrations.
2. **GIF (Graphics Interchange Format)**: GIF is another lossless format that uses LZW compression. It is limited to 256 colours, making it suitable for simple graphics like icons and animations, but not for high-quality photographic images. GIF is popular for web-based graphics and short animated loops.

5.4.3 Scenarios Where Loss-less Compression Is Preferable

1. **Medical Imaging**: Loss-less compression is essential in medical imaging, where even the smallest loss of detail could affect a diagnosis. Formats like DICOM (Digital Imaging and Communications in Medicine) use lossless compression to

ensure that medical professionals can access every detail of an image.

2. **Archival Storage**: When storing historical or legal documents, it is critical to preserve the exact original form of the image. Loss-less compression ensures that no data is lost during compression, making it ideal for archiving.
3. **Graphics with Text**: Images that contain text or sharp edges (such as logos or diagrams) are prone to visible artifacts with lossy compression. Loss-less formats like PNG are preferred for such images, as they maintain the crispness and clarity of the text and edges.

5.5 Run-Length Encoding (RLE) and Huffman Coding

Explanation of RLE as a Form of Lossless Compression:

Run-Length Encoding (RLE) is one of the simplest forms of lossless compression. It is particularly effective for images that contain long runs of identical pixel values, such as black-and-white images, graphics, or images with large uniform areas. The basic principle behind RLE is to store repeated values as a single value followed by a count of repetitions.

For example, consider an image row with the following pixel values:

White White White White Black Black Black Black White White

Instead of storing each pixel individually, RLE compresses this data as:

4W 4B 2W

This significantly reduces the size of the data when there are long runs of repeated values. However, RLE is less effective for images with a lot of colour variation or noise, where few long runs of identical values exist.

5.5.1 Use of Huffman Coding to Reduce Data Size

Huffman Coding is an entropy encoding technique that assigns shorter binary codes to more frequent symbols and longer codes to less frequent symbols. By doing so, Huffman coding can efficiently compress image data based on the frequency of pixel values.

For instance, in an image with a limited colour palette, some colours may appear more frequently than others. Huffman coding assigns shorter binary codes to these frequent colours, reducing the overall size of the compressed data. Conversely, less frequent colours receive longer codes.

Huffman coding is used as part of many image compression algorithms, including **JPEG** (for compressing the quantized DCT coefficients) and **DEFLATE** (used in PNG).

5.5.2 Practical Applications of RLE and Huffman Coding in Real-World Compression Tasks

- **RLE** is commonly used in **BMP**, **TIFF**, and **GIF** formats for compressing simple images with large uniform areas of colour. It is also used in **fax machines**, where documents typically contain long runs of black or white pixels.
- **Huffman Coding** is used in formats such as **JPEG** (as part of the entropy coding step after quantization) and **DEFLATE** (used in PNG), where it helps reduce file size by compressing common patterns in the image data.

5.6 Block Transform Coding

Introduction to Block Transform Methods Like Discrete Cosine Transform (DCT):

Block Transform Coding is a technique that divides an image into smaller blocks (typically 8x8 pixels), applies a mathematical transformation (such as the Discrete Cosine Transform, or DCT) to each block, and then compresses the resulting coefficients. This approach is the cornerstone of the **JPEG compression standard**.

The **Discrete Cosine Transform (DCT)** converts the pixel data from the spatial domain (the image as we see it) into the frequency domain. In the frequency domain, the image is represented as a sum of sinusoidal waves, with different frequencies corresponding to different levels of detail in the image. By quantizing and discarding the higher-frequency components (which typically correspond to fine details and noise), the image can be compressed significantly.

5.6.1 How These Techniques Are Used in Popular Formats Like JPEG

In **JPEG**, the image is first divided into 8x8 pixel blocks. Each block is then transformed using the DCT, which converts the spatial data into frequency data. After transformation, each frequency coefficient is quantized to reduce its precision (this is where data loss occurs in JPEG).

The lower-frequency components (representing large, smooth areas) are preserved with high precision, while higher-frequency components (representing fine details) are quantized more aggressively or even discarded. The remaining coefficients are then compressed using entropy coding (such as Huffman coding) to further reduce file size.

5.6.2 The Mathematics Behind Block-Based Image Compression and Its Impact on Efficiency

The DCT represents each block of pixels as a weighted sum of cosine waves at different frequencies. The resulting coefficients indicate the contribution of each frequency to the overall image block:

- **Low-frequency coefficients**: These represent the overall smoothness or brightness of the image block.
- **High-frequency coefficients**: These represent finer details and sharp transitions within the block.

By compressing the high-frequency coefficients more aggressively, block transform coding achieves high compression ratios without significantly degrading image quality.

Block-based compression is efficient because it reduces the overall amount of data needed to store the image, while also allowing localized control over the compression process (since each block is compressed independently). However, at high compression ratios, block-based methods can introduce visible artifacts, such as **blocking artifacts**—where the boundaries between 8x8 blocks become noticeable.

Check Your Progress

State true or false

1. Quantization increases the precision of pixel values.

2 GIF is a lossless compression format.

3. Huffman coding can efficiently compress image data based on the frequency of pixel values.

4. The Discrete Cosine Transform (DCT) converts the pixel data from the frequency domain into the spatial domain.

5. At high compression ratios, block-based methods can introduce visible artifacts,

5.7 Summing Up

1. Lossy compression is a widely used image compression technique where some amount of image data is discarded to achieve higher compression ratios.

2. The techniques that discard some image data to achieve higher compression ratios are quantization, chroma subsampling and transform coding.

3. The primary impact of lossy compression is image degradation.

4. Loss-less compression is a method of image compression where no data is lost during the process.

5. Lossless compression is preferable in medical imaging, archival storage and graphic with text.

6. Run-Length Encoding (RLE) is one of the simplest forms of lossless compression.

7. Huffman Coding is an entropy encoding technique that assigns shorter binary codes to more frequent symbols and longer codes to less frequent symbols.

8. Huffman coding is used as part of many image compression algorithms.

9. Block Transform Coding is a technique that divides an image into smaller blocks, applies a mathematical transformation to each block, and then compresses the resulting coefficients.

10. Block-based compression is efficient because it reduces the overall amount of data needed to store the image, while also allowing localized control over the compression process.

5.8 Answers to Check Your Progress

1. False 2. True 3.True 4. False 5. True

5.9 Possible Questions

1. What are the main differences between Run-Length Encoding (RLE) and Huffman coding in image compression?

2. How does block-based transform coding work in JPEG compression?

3. Why might you choose lossless compression methods like PNG over lossy formats for specific use cases?

5.10 References and Suggested Readings

1. Taubman, D. & Marcellin, M.W. **JPEG 2000: Image Compression Fundamentals, Standards and Practice**, Springer.
2. Sayood, K. **Introduction to Data Compression**, Morgan Kaufmann.
3. Wallace, G.K. **The JPEG Still Picture Compression Standard**, Communications of the ACM.

*****

UNIT: 6
IMAGE COMPRESSION III

Unit Structure:

6.1 Introduction

In this chapter, we will explore how modern web technologies such as AJAX, ISAPI, and .NET can be utilized to handle images dynamically and compress them efficiently in real-time. We will also learn newer image formats like WebP and AVIF, which provide superior compression for web use cases, and discussed how to balance performance and quality in dynamic web environments.

6.2 Objectives

After going through this unit, you will learn

- *different* modern web-based compression techniques,
- AJAX, ISAPI and .Net technologies,
- Best practices for using the above technologies in dynamic web environments.

6.3. Advanced Web Technologies for Image Compression

In the modern web, efficient image handling is essential for delivering fast, responsive, and engaging user experiences. As web technologies evolve, so do the strategies for handling and compressing images to ensure that websites and applications can load quickly, even on limited bandwidth. Advanced web technologies, such as AJAX, ISAPI, and .NET, provide powerful tools for managing and transmitting compressed images dynamically.

6.4 Introduction to More Modern Web-Based Compression Techniques

The demand for faster web applications has driven the development of more sophisticated image compression techniques specifically designed for web environments. Traditional compression methods like JPEG, PNG, and GIF are still widely used, but new formats like WebP and AVIF have emerged, offering superior compression ratios and performance.

- **WebP**: Developed by Google, WebP is designed to provide high-quality lossy and lossless image compression for the web. WebP images are smaller than JPEG and PNG, making them ideal for use in websites and applications where speed and bandwidth are critical.
- **AVIF (AV1 Image File Format)**: AVIF is a more recent image format based on the AV1 video codec. It provides exceptional compression efficiency, allowing for smaller file sizes while maintaining high visual quality. AVIF is particularly useful in web environments where delivering large numbers of high-resolution images efficiently is essential.
- **Responsive Images**: With the growing use of mobile devices and high-resolution displays, delivering appropriately sized images has become critical. Web technologies like srcset and the <picture> element allow developers to serve different image sizes based on the user's device, reducing the need for unnecessary bandwidth.
- **Lazy Loading**: Lazy loading is a technique where images are only loaded as they come into view in the user's browser. This reduces the initial load time of a webpage, especially on image-heavy sites. Combined with compression, this strategy can significantly improve the performance of web applications.

6.4.1 Key Web-Based Image Compression Techniques

1. **Server-Side Compression**: Many modern websites use server-side techniques to compress images before sending them to clients. This ensures that images are optimized for the

user's connection speed and device type, resulting in faster load times and reduced data usage.

2. **Client-Side Compression**: Client-side compression involves compressing or optimizing images in the user's browser using JavaScript or Web APIs. Libraries like **Compressor.js** can be used to resize and compress images in the browser before uploading them to a server, reducing both server load and the bandwidth required for uploads.
3. **CDN-Based Compression**: Content Delivery Networks (CDNs) can automatically compress and optimize images for different devices and regions. CDNs like **Cloudflare** and **AWS CloudFront** offer image optimization services, ensuring that users receive the smallest possible image without compromising quality.

6.5 AJAX (Asynchronous JavaScript and XML) for Efficient Image Handling

AJAX (Asynchronous JavaScript and XML) is a technology that allows web applications to send and retrieve data from a server asynchronously, without refreshing the entire page. This capability is critical for modern web applications that handle images dynamically.

6.5.1 Introduction to AJAX in Image Handling

AJAX enables dynamic image loading and manipulation in the background, enhancing user experiences by allowing images to be fetched and displayed without interrupting the user's interaction with the web page. For example, in a photo gallery, images can be loaded on demand as the user scrolls through the page, rather than all images being loaded at once.

- **Image Loading on Demand**: By using AJAX, developers can load images when needed, reducing the initial page load time. This is especially important for pages that include a large number of images or high-resolution images.
- **Dynamic Image Compression**: With AJAX, developers can request compressed versions of images from the server based on factors like the user's device, screen size, or internet connection. This ensures that users are always served the most appropriate image for their circumstances.

6.5.2 Best Practices for Using AJAX with Image Compression

1. **Use Appropriate File Formats**: When delivering images dynamically via AJAX, it's important to use modern, web-optimized formats like WebP or AVIF. These formats provide better compression ratios than older formats like JPEG or PNG, resulting in faster load times.
2. **Leverage Caching**: To reduce the load on both the client and the server, caching mechanisms can be used to store compressed images locally on the client's browser or within a CDN. AJAX requests can check if an image is already cached before making a server request, saving time and bandwidth.
3. **Minimize Server Requests**: When loading images dynamically, it's important to minimize the number of AJAX requests made to the server. Techniques like **batching requests** or **lazy loading** can reduce the frequency of image requests, improving overall performance.

6.6 ISAPI (Internet Server Application Programming Interface) for Image Compression

ISAPI is a technology primarily used in Microsoft's Internet Information Services (IIS) that allows developers to create high-performance web applications with server-side functionality. ISAPI extensions and filters can be used to compress images and improve the efficiency of image handling and transmission on the server.

6.6.1 Introduction to ISAPI and Image Handling

ISAPI provides a powerful mechanism for managing image compression at the server level. With ISAPI, developers can create custom extensions and filters that intercept HTTP requests and responses to compress images dynamically before they are sent to the client. This can significantly reduce the amount of data transferred over the network, improving website performance, especially for image-heavy sites.

- **ISAPI Extensions**: Extensions can be used to implement custom image compression algorithms or integrate third-party compression libraries directly into the server's image handling pipeline.
- **ISAPI Filters**: Filters can be applied to compress images before they are transmitted to the client. For example, a filter could be used to compress images based on the user's device capabilities, sending lower-quality images to mobile devices and higher-quality images to desktop users.

6.6.2 Best Practices for Using ISAPI for Image Compression

1. **Implement Compression Filters**: Use ISAPI filters to automatically compress images before they are sent to the client. This can be particularly useful for optimizing performance on image-heavy websites where bandwidth is a concern.

2. **Integrate CDN Support**: Combine ISAPI-based compression with CDN services to deliver compressed images more efficiently. CDNs can cache the compressed images at edge locations, reducing the time it takes to deliver images to users in different regions.
3. **Monitor and Optimize Server Performance**: While ISAPI can improve image handling efficiency, it's important to monitor server performance to ensure that image compression tasks do not overload the server. Consider balancing image compression workloads with other server processes to maintain optimal performance.

6.7 .NET Technologies for Image Handling and Compression

The **.NET** framework offers a wide range of tools and libraries for handling and compressing images in web applications. By leveraging .NET's robust capabilities, developers can ensure that images are processed efficiently on both the server and client sides.

6.7.1 Introduction to .NET for Image Handling

With .NET, developers can integrate image processing and compression directly into their web applications. This is useful for tasks such as resizing, compressing, and converting image formats dynamically before delivering them to users.

- **System.Drawing**: The System.Drawing namespace in .NET provides a set of classes for working with images. Developers can use this namespace to load, manipulate, and save images in various formats, as well as compress images before sending them to the client.

- **Image Processing Libraries**: .NET supports third-party image processing libraries like **ImageSharp** and **Magick.NET**, which provide advanced image manipulation and compression capabilities. These libraries allow developers to perform complex operations, such as resizing, cropping, and compressing images on the server before transmitting them.

6.7.2 Best Practices for Using .NET Technologies for Image Compression

1. **Optimize Image Formats**: When working with images in .NET, it's important to choose the most efficient file format for the application. For example, use WebP for web-based images to minimize file size while maintaining quality.
2. **Use Asynchronous Processing**: When processing large images on the server, use asynchronous methods in .NET to avoid blocking other tasks. This ensures that the server remains responsive, even when handling resource-intensive image compression tasks.
3. **Integrate Cloud Services**: Leverage cloud-based image handling services, such as **Azure Blob Storage** or **AWS S3**, to offload image storage and compression tasks from the server. This allows the .NET application to focus on core logic while leaving image processing to more specialized services.

6.8 Best Practices for Using These Technologies in Dynamic Web Environments

The effective use of AJAX, ISAPI, and .NET for image compression in dynamic web environments requires careful planning and implementation. Here are some best practices to follow:

1. Combine Compression with Responsive Design:

Ensure that images are appropriately compressed and resized based on the user's device and screen resolution. Use tools like **srcset** and <picture> to serve different image sizes for different devices, ensuring optimal performance and visual quality across desktops, tablets, and smartphones.

2. Leverage Caching and CDNs:

Use Content Delivery Networks (CDNs) to cache compressed images closer to users. This reduces latency and improves load times, especially for global websites. Combining CDN caching with compression ensures that users always receive optimized images, no matter where they are located.

3. Implement Progressive Loading:

When delivering large images, consider using progressive image formats like **Progressive JPEG**. This allows users to see a lower-quality version of the image while the rest of the image is loading, improving perceived performance.

4. Use Lazy Loading:

Lazy loading is an essential technique for modern web development. Images that are not immediately visible should not be loaded until the user scrolls to them. This reduces initial page load times and improves overall user experience.

5. Monitor Performance:

Use performance monitoring tools like **Google Lighthouse** or **WebPageTest** to track how image compression impacts your website's performance. Ensure that compressed images are delivering the desired results in terms of both speed and visual quality.

Check Your Progress

Fill in the blanks

1. __________is a technique where images are only loaded as they come into view in the user's browser

2. _______ and __________ offer image optimization services, ensuring that users receive the smallest possible image without compromising quality.

3. By using__________, developers can load images when needed, reducing the initial page load time

4. ________provides a powerful mechanism for managing image compression at the server level.

5._____________ is used to cache compressed images closer to users

6.9 Summing Up

1. Advanced web technologies, such as AJAX, ISAPI, and .NET, provide powerful tools for managing and transmitting compressed images dynamically.

2. The new image formats like WebP and AVIF have emerged, offering superior compression ratios and performance.

3. **AJAX** (Asynchronous JavaScript and XML) is a technology that allows web applications to send and retrieve data from a server asynchronously, without refreshing the entire page

4. **ISAPI** is a technology primarily used in Microsoft's Internet Information Services (IIS) that allows developers to create high-performance web applications with server-side functionality.

5. The **.NET** framework offers a wide range of tools and libraries for handling and compressing images in web applications.

6.10 Answers to Check Your Progress

1. Lazy loading
2. Cloudflare, AWS CloudFront
3. AJAX
4. ISAPI
5. Content Delivery Networks

6.11 Possible Questions:

1.What are the advantages of using AJAX for asynchronous image loading in a web application?

2. How does ISAPI help improve server-side image compression?

3. In what scenarios would you prefer to use WebP or AVIF over traditional formats like JPEG or PNG?

6.12 References and Suggested Readings:

1. Duckett, J. **JavaScript and jQuery: Interactive Front-End Web Development**, Wiley.
2. Troelsen, A. **Pro C# 8.0 and the .NET 4 Framework**, Apress.
3. **WebP Compression and Its Benefits**, Google Developers Documentation.

*****