# MODULE-4

## Overview of Image Processing

Computers are faster and more accurate than human beings in processing numerical data. However, human beings score over computers in recognition capability. The human brain is so sophisticated that we recognize objects in a few seconds without much difficulty. Human beings use all the five sensory organs to gather knowledge about the outside world. Among these perceptions, visual information plays a major role in understanding the surroundings. Other kinds of sensory information are obtained from hearing, taste, smell and touch.

With the advent of cheaper digital cameras and computer systems, we are witnessing a powerful digital revolution, where images are being increasingly used to communicate effectively.

Images are encountered everywhere in our daily lives. We see many visual information sources such as paintings and photographs in magazines, journals, image galleries, digital libraries, newspapers, advertisement boards, television, and the Internet. Many of us take digital snaps of important events in our lives and preserve them as digital albums. Then from the digital album, we print digital pictures or mail them to our friends to share our feelings of happiness and sorrow. Images are not used merely for entertainment purposes. Doctors use medical images to diagnose problems for providing treatment. With modern technology, it is possible to image virtually all anatomical structures, which is of immense help to doctors in providing better treatment. Forensic imaging application process fingerprints, faces and irises to identify criminals. Industrial applications use imaging technology to count and analyse industrial components. Remote sensing applications use images sent by satellites to locate the minerals present in the earth.

Images are imitations of real-world objects. Image is a two-dimensional (2D) signal f(x,y), where the values of the function f(x,y) represent the amplitude or intensity of the image. For processing using digital computers, this image has to be converted into a discrete form using the process of sampling and quantization, known collectively as **digitization**. In image processing, the term ‘image’ is used to denote the image data that is sampled, quantized and readily available in a form suitable for further processing by digital computers. Image processing is an area that deals with manipulation of visual information.

Major objectives of image processing is to

- Improve the quality of pictorial information for better human interpretation.
- Facilitate the automatic machine interpretation of images.

## Nature of Image Processing

There are three scenarios or ways of acquiring an image

1. Reflective mode Imaging
2. Emissive Type Imaging
3. Transmissive Imaging

The radiation source shown in Figure 4.1 is the light source.

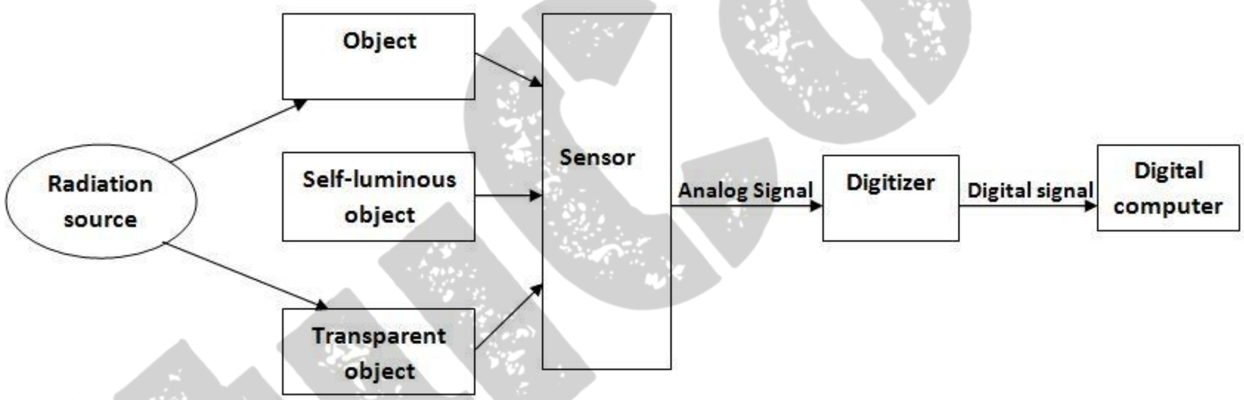



**Figure 4.1: Image processing environment**

Objects are perceived by the eye because of light. The sun, lamps, and clouds are all examples of radiation or light sources. The object is the target for which the image needs to be created. The object can be people, industrial components, or the anatomical structure of a patient. The objects can be two-dimensional, three-dimensional or multidimensional mathematical functions involving many variables. For example, a printed document is a 2D object. Most real-world objects are 3D.

**Reflective Mode Imaging**

Reflective mode imaging represents the simplest form of imaging and uses a sensor to acquire the digital image. All video cameras, digital cameras, and scanners use some types of sensors for capturing the image. Image sensors are important components of imaging systems. They convert light energy to electric signals.

**Emissive Type Imaging**

In Emissive type imaging , images are acquired from self-luminous objects without the help of a radiation source . In emissive type imaging, the objects are self-luminous. The radiation emitted by the object is directly captured by the sensor to form an image. Thermal imaging is an example of emissive type imaging. In thermal imaging, a specialized thermal camera is used in low light situations to produce images of objects based on temperature. Other examples of emissive type imaging are magnetic resonance imaging (MRI) and positron emissive tomography (PET)

**Transmissive Imaging**

In Transmissive imaging, the radiation source illuminates the object. The absorption of radiation by the objects depends upon the nature of the material. Some of the radiation passes through the objects. The attenuated radiation is sensed into an image. This is called transmissive imaging. Examples of this kind of imaging are X-ray imaging, microscopic imaging, and ultrasound imaging.

The first major challenge in image processing is to acquire the image for further processing. Figure 4.1 shows three types of processing – optical, analog and digital image processing.

**Optical Image Processing**

Optical image processing is the study of the radiation source, the object, and other optical processes involved. It refers to the processing of images using lenses and coherent light beams instead of computers. Human beings can see only the optical image. An optical image is the 2D projection of a 3D scene. This is a continuous distribution of light in a 2D surface and contains information about the object that is in focus. This is the kind of information that needs to be captured for the target image. Optical image processing is an area that deals with the object, optics, and how processes are applied to an image that is available in the form of reflected or transmitted light. The optical image is said to be available in optical form till it is converted into analog form.

**Analog Image Processing**

An analog or continuous image is a continuous function f(x,y) where x and y are two spatial coordinates. Analog signals are characterized by continuous signals varying with time. They are often referred to as pictures. The processes that are applied to the analog signal are called analog processes. **Analog image processing** is an area that deals with the processing of analog electrical signals using analog circuits. The imaging systems that use film for recording images are also known as analog imaging systems.

**Digital Image Processing**

The analog signal is often sampled, quantized and converted into digital form using digitizer. **Digitization** refers to the process of sampling and quantization. **Sampling** is the process of converting a continuous-valued image f(x,y) into a discrete image, as computers cannot handle continuous data. So the main aim is to create a discretized version of the continuous data. Sampling is a reversible process, as it is possible to get the original image back. **Quantization** is the process of converting the sampled analog value of the function f(x,y) into a discrete-valued integer. **Digital image processing** is an area that uses digital circuits, systems and software algorithms to carry out the image processing operations. The image processing operations may include quality enhancement of an image, counting of objects, and image analysis.

Digital image processing has become very popular now as digital images have many advantages over analog images. Some of the advantages are as follows:

1. It is easy to post-process the image. Small corrections can be made in the captured image using software.
2. It is easy to store the image in the digital memory.
3. It is possible to transmit the image over networks. So sharing an image is quite easy.
4. A digital image does not require any chemical process. So it is very environment friendly, as harmful film chemicals are not required or used.

5. It is easy to operate a digital camera.

The disadvantages of digital images are very few. Some of the advantages are the initial cost, problems associated with sensors such as high power consumption and potential equipment failure, and other security issues associated with the storage and transmission of digital images.

The final form of an image is the display image. The human eye can recognize only the optical form. So the digital image needs to be converted to optical form through the digital to analog conversion process.

## Image Processing and Related Fields

Image processing is an exciting interdisciplinary field that borrows ideas freely from many fields. Figure 4.2 illustrates the relationships between image processing and other related fields.

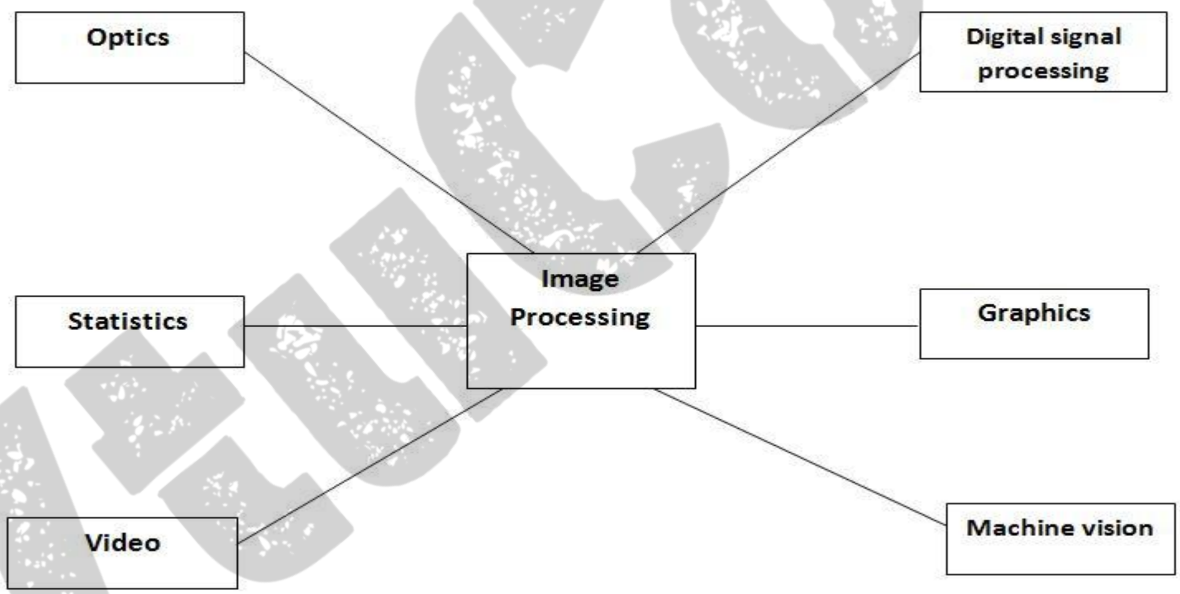



**Figure 4.2: Image Processing and other closely related fields**

### 1) Image Processing and Computer Graphics

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

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

**2) Image Processing and Signal Processing**

Human beings interact with the environment by means of various signals. In digital signal processing, one often deals with the processing of a one-dimensional signal. In the domain of image processing, one deals with visual information that is often in two or more dimensions. Therefore, image processing is a logical extension of signal processing.

**3) Image Processing and Machine Vision**

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

**4) Image Processing and Video processing**

Image processing is about still images. Analog video cameras can be used to capture still images. A video can be considered as a collection of images indexed by time. Most image processing algorithms work with video readily. Thus, video processing is an extension of image processing.

Images are strongly related to multimedia, as the field of multimedia broadly includes the study of audio, video, images, graphics and animation.

**5) Image Processing and Optics**

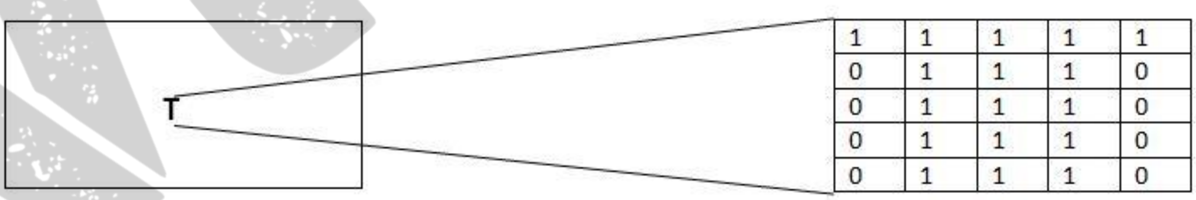
Optical image processing deals with lenses, light, lighting conditions, and associated optical circuits. The study of lenses and lighting conditions has an important role in study of image processing.

**6) Image Processing and Statistics**

Image analysis is an area that concerns the extraction and analysis of object information from the image. Imaging applications involve both simple statistics such as counting and mensuration and complex statistics such as advanced statistical inference. So statistics plays an important role in imaging applications. Image understanding is an area that applies statistical inferencing to extract more information from the image.

## Digital Image Representation

An image can be defined as a 2D signal that varies over the spatial coordinates x and y, and can be written mathematically as f(x,y). Medical images such as magnetic resonance images and computerized tomography(CT) images are 3D images that can be represented as f(x,y,z), where x,y, and z are spatial coordinates. A simple digital image and its matrix equivalent are shown in Figs 4.3(a) and 4.3(b).




(a) (b)

**Figure 4.3: Digital Image Representation (a) Small binary digital image (b) Equivalent image contents in matrix form**

Figure 4.3(a) shows a displayed image. The source of the image is a matrix as shown in Fig. 4.3(b). The image has five rows and five columns. In general, the image can be written as a mathematical function f(x,y) as follows:

$$f(x,y)=\begin{pmatrix} f(0,0) & f(0,1) & f(0,2) & \cdots & f(0,Y-1) \\ f(1,0) & f(1,1) & f(1,2) & \cdots & f(1,Y-1) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ f(X-1,0) & f(X-1,1) & f(X-1,2) & \cdots & f(X-1,Y-1) \end{pmatrix}$$

In general, the image f(x,y) is divided into X rows and Y columns. Thus, the coordinate ranges are {x=0,1,……X-1} and {y=0,1,2,…….Y-1}. At the intersection of rows and columns, pixels are present. Pixels are the building blocks of digital images. Pixels combine together to give a digital image. Pixel represents discrete data. A pixel can be considered as a single sensor, photosite(physical element of the sensor array of a digital camera), element of a matrix, or display element on a monitor.

The value of the function f(x,y) at every point indexed by row and a column is called **grey value** or **intensity** of the image. The value of the pixel is the intensity value of the image at that point. The intensity value is the sampled, quantized value of the light that is captured by the sensor at that point. It is a number and has no units.

The number of rows in a digital image is called **vertical resolution**. The number of columns is called **horizontal resolution**. The number of rows and columns describes the dimensions of the image. The image size is often expressed in terms of the rectangular pixel dimensions of the array. Images can be of various sizes. Some examples of image size are 256 X 256, 512 X 512. For a digital camera, the image size is defined as the number of pixels (specified in megapixels)

Resolution is an important characteristic of an imaging system. It is the ability of the imaging system to produce the smallest discernable details, that is the smallest sized object clearly and differentiate it from the neighbouring small objects that are present in the image. Image resolution depends on two factors- **optical resolution** of the lens and **spatial resolution**.

Spatial resolution of the image is very crucial as the digital image must show the object and its separation from the other spatial objects that are present in the image clearly and precisely.

A useful way to define resolution is the smallest number of line pairs per unit distance. The resolution can then be quantified as 200 line pairs per mm.

Spatial resolution depends on two parameters – the **number of pixels of the image** and the **number of bits necessary for adequate intensity resolution**, referred to as the bit depth. The numbers of pixels determine the quality of the digital image. The total number of pixels that are present in the digital image is the number of rows multiplied by the number of columns.

The choice of bit depth is very crucial and often depends on the precision of the measurement system. To represent the pixel intensity value, certain bits are required. For example, in binary images, the possible pixel values are 0 or 1. To represent two values, one bit is sufficient. The number of bits necessary to encode the pixel value is called **bit depth**. Bit depth is a power of two. It can be written as $2^m$ . In monochrome grey scale images (e.g medical images such as X-rays and ultrasound images), the pixel values can be between 0 and 255. Hence, eight bits are used to represent the grey shades between 0 and 255.(as $2^8$ =256). So the bit depth of grey scale images is 8. In colour images, the pixel value is characterized by both colour value and intensity value. So colour resolution refers to the number of bits used to represent the colour of the pixel. The set of all colours that can be represented by the bit depth is called **gamut** or **palette**.

So, the total number of bits necessary to represent the image is

**Number of rows x Number of columns x Bit depth**

Spatial resolution depends on the number of pixels present in the image and the bit depth. Keeping the number of pixels constant but reducing the quantization levels(bit depth) leads to phenomenon called **false contouring**. The decrease in number of pixels while retaining the quantization levels leads to a phenomenon called **checkerboard effect** (or pixelization error).

A 3D image is a function f(x,y,z) where x,y, and z are spatial coordinates. In 3D images, the term ‘voxel’ is used for pixel. Voxel is an abbreviation of ‘volume element’.

## Types of Images

Images can be classified based on many criteria.

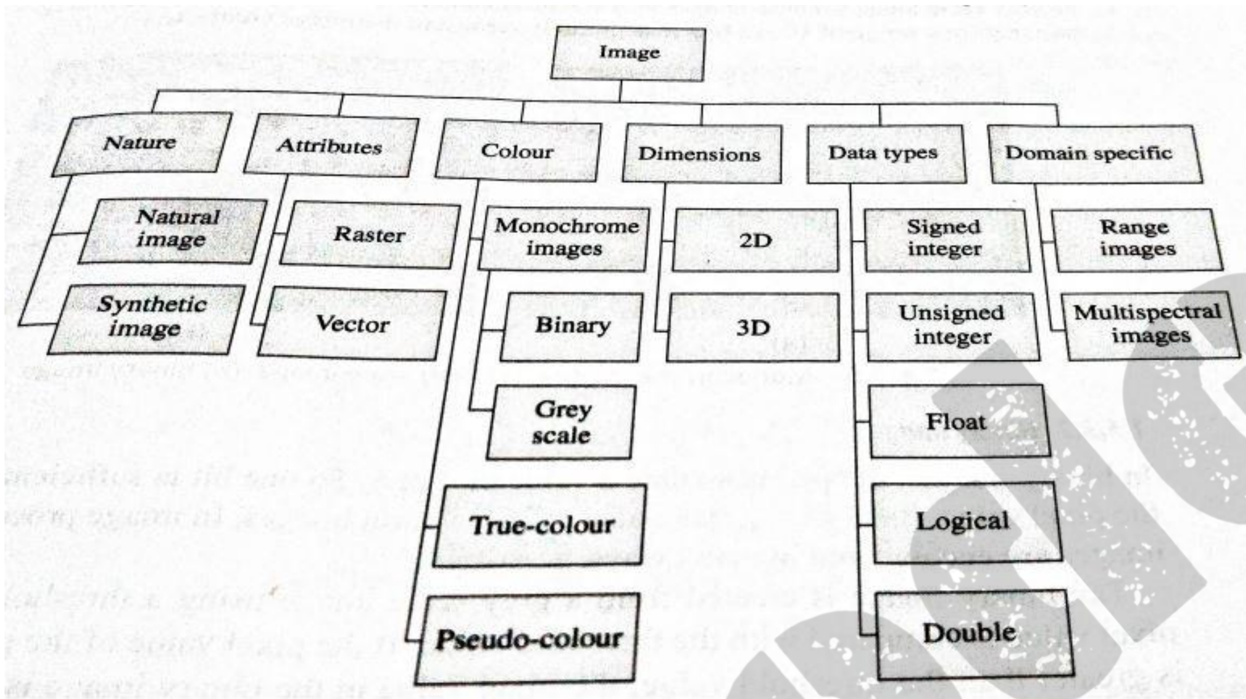



**Figure 4.4: Classification of Images**

**Based on Nature**

Images can be broadly classified as **natural** and **synthetic** images. **Natural images** are images of the natural objects obtained using devices such as cameras or scanners. **Synthetic images** are images that are generated using computer programs.

**Based on Attributes**

Based on attributes, images can be classified as **raster images** and **vector graphics**. **Vector graphics** use basic geometric attributes such as lines and circles, to describe an image. Hence the notion of resolution is practically not present in graphics. **Raster images** are pixel-based. The quality of the raster images is dependent on the number of pixels. So operations such as enlarging or blowing-up of a raster image often result in quality reduction.

**Based on Colour**

Based on colour, images can be classified as **grey scale, binary, true colour** and **pseudocolour images**.

Grayscale and binary images are called monochrome images as there is no colour component in these images. True colour(or full colour) images represent the full range of available colours. So the images are almost similar to the actual object and hence called true colour images. In addition, true colour images do not use any lookup table but store the pixel information with full precision. Pseudocolour images are false colour images where the colour is added artificially based on the interpretation of the data.

**i) Grey scale Images**

Grey scale images are different from binary images as they have many shades of grey between black and white. These images are also called monochromatic as there is no colour component in the image, like in binary images. Grey scale is the term that refers to the range of shades between white and black or vice versa.

Eight bits ($2^8$=256) are enough to represent grey scale as the human visual system can distinguish only 32 different grey levels. The additional bits are necessary to cover noise margins. Most medical images such as X-rays, CT images, MRIs and ultrasound images are grey scale images. These images may use more than eight bits. For example, CT images may require a range of 10-12 bits to accurately represent the image contrast.

**(a)** **(b)**

**Figure 4.5: Monochrome images (a) Grey scale image (b) Binary image**

**ii) Binary Images**

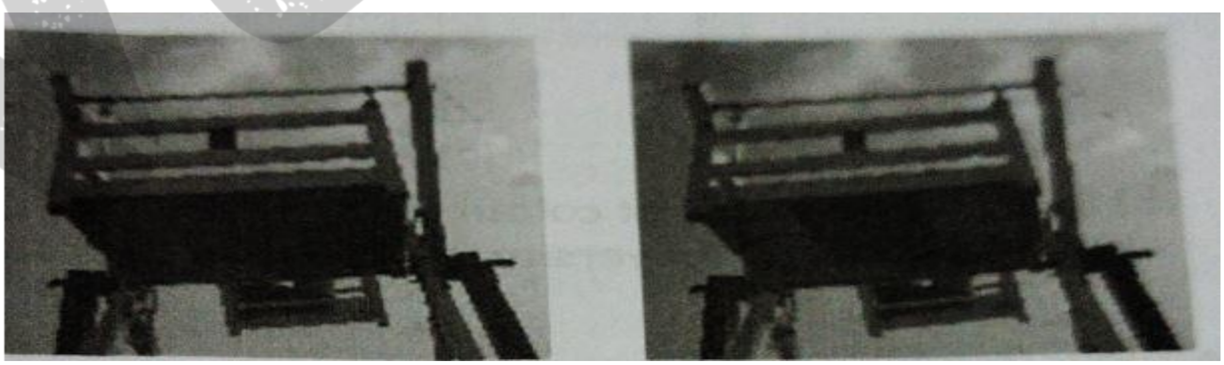
In binary images, the pixels assume a value of 0 or 1. So one bit is sufficient to represent the pixel value. Binary images are also called bi-level images. In image processing, binary images are encountered in many ways.

The binary image is created from a grey scale image using a **threshold process**. The pixel value is compared with the threshold value. If the pixel value of the grey scale image is greater than the threshold value, the pixel value in the binary image is considered as 1.Otherwise, the pixel value is 0. The binary image is created by applying the threshold process on the grey scale image in Fig. 4.5(a) is displayed in Fig. 4.5(b) . It can be observed that most of the details are eliminated. However, binary images are often used in representing basic shapes and line drawings. They are also used as masks. In addition, image processing operations produce binary images at intermediate stages.

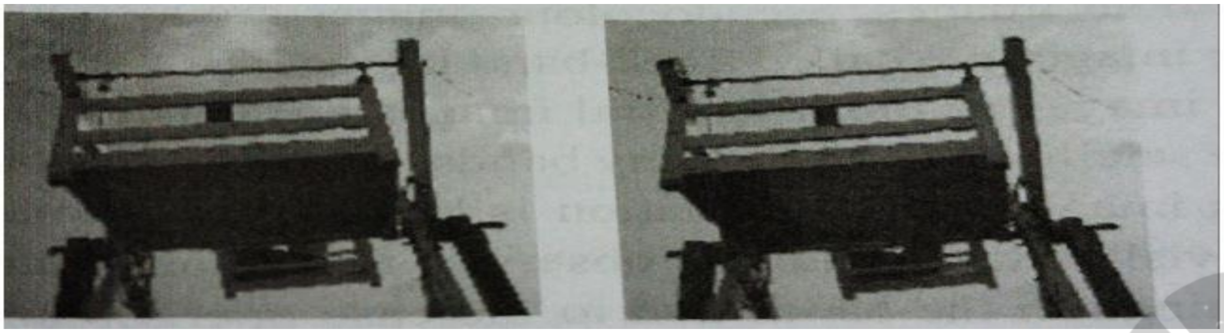
**iii) True Colour Images**

In true colour images, the pixel has a colour that is obtained by mixing the primary colours red,green and blue. Each colour component is represented like a grey scale image using eight bits. Mostly, true colour images use 24 bits to represent all the colours. Hence true colour images can be considered as **three-band images**. The number of colours that is possible is $256^3$ (i.e 256x256x256=1,67,77,216 colours)

Figure 4.6 a) shows a colour image and its three primary colour components. Figure 4.6 b illustrates the general storage structure of the colour image. A display controller then uses a digital-to-analog converter(DAC) to convert the colour value to the pixel intensity of the monitor.



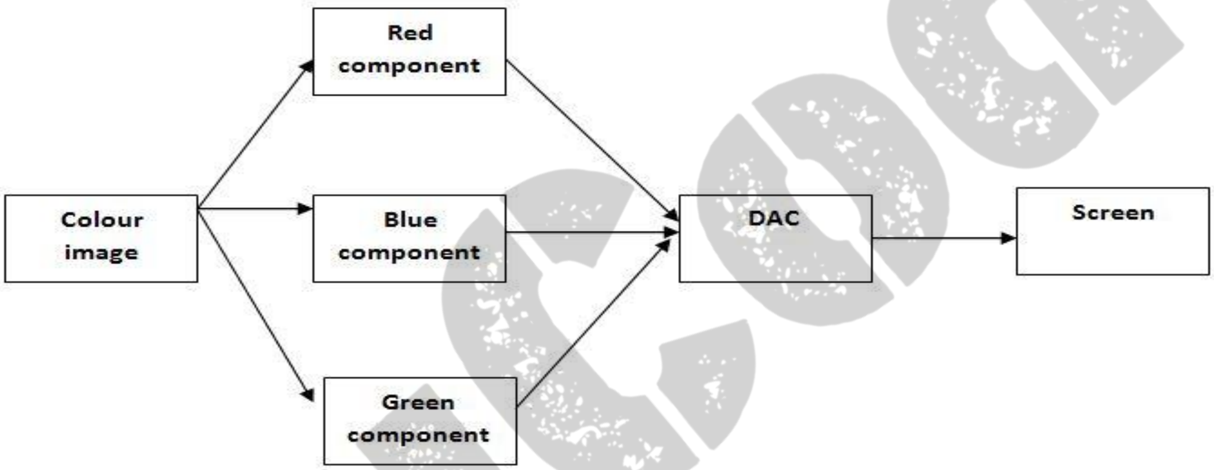
**Original Image** **Red Component**



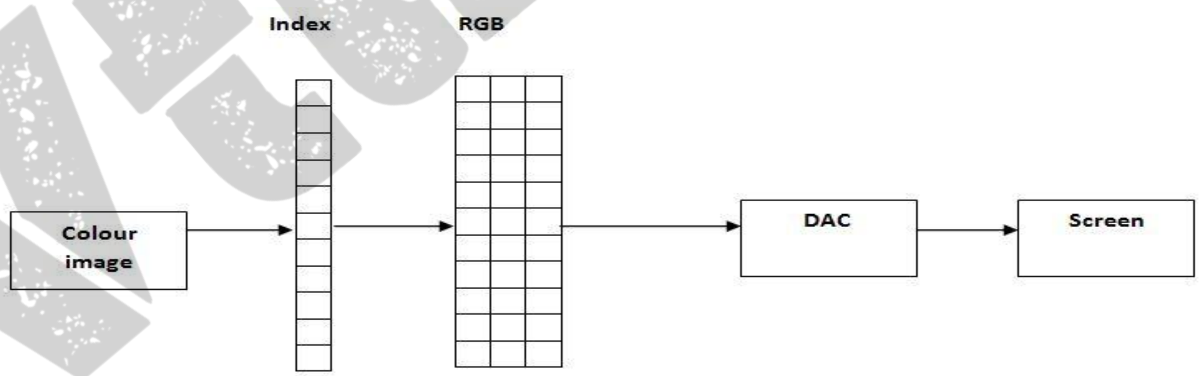
**Green Component** **Blue Component**

**(a)**

**Figure 4.6: True colour images (a) Original image and its colour components**




**(b)**




**(c)**

**Figure 4.6: (b) Storage structure of colour images (c) Storage structure of an indexed image**

A special category of colour images is the indexed image. In most images, the full range of colours is not used. So it is better to reduce the number of bits by maintaining a colour map, gamut, or palette with the image. Figure 4.6(c) illustrates the storage structure of an indexed image. The pixel value can be considered as a pointer to the index, which contains the address of the colour map. The colour map has RGB components. Using this indexed approach, the number of bits required to represent the colours can be drastically reduced. The display controller uses a DAC to convert the RGB value to the pixel intensity of the monitor.

**iv)Pseudocolour Images**

Like true colour images, pseudocolour images are also widely used in image processing. True colour images are called three-band images. However, in remote sensing applications multi-band images or multi-spectral images are generally used. These images, which are captured by satellites contains many bands. A typical remote sensing image may have 3-11 bands in an image. This information is beyond the human perceptual range. Hence it is mostly not visible to the human observer. So colour is artificially added to these bands, so as to distinguish the bands and to increase operational convenience. These are called artificial colour or pseudocolour images. Pseudocolour images are popular in the medical domain also. For example, the Doppler colour image is a pseudocolour image.

**Based on Dimensions**

Images can be classified based on dimension also. Normally, digital images are 2D rectangular array of pixels. If another dimension, of depth or any other characteristics, is considered, it may be necessary to use a higher-order stack of images.  A good example of a 3D image is a volume image, where pixels are called voxels. By '3D image', it is meant that the dimension of the target in the imaging system is 3D. The target of the imaging system may be a scene or an object. In medical imaging, some of the frequently encountered images are CT images, MRIs and microscopy images. Range images, which are often used in remote sensing applications, are also 3D images.

**Based on Data Types**

Images may be classified based on their data type. A binary image is a 1-bit image as one bit is sufficient to represent black and white pixels. Grey scale images are stored as one-byte(8-bit) or two-byte(16-bit) images, With one byte, it is possible to represent $2^8$ , that is 0-255=256 shades and with 16 bits, it is possible to represent $2^{16}$, that is 65,536 shades. Colour images often use 24 or 32 bits to represent the colour and intensity value.

Sometimes, image processing operations produce images with negative numbers, decimal fractions, and complex numbers. For example, Fourier transforms produce images involving complex numbers. To handle, negative numbers, signed and unsigned integer types are used. In these data types, the first bit is used to encode whether the number is positive or negative. Floating-point involves storing the data in scientific notation. For example, 1230 can be represented as $0.1234\text{x}10^4$ , where 0.123 is called the significand and the power is called the exponent. There are many floating-point conventions.

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

**Domain Specific Images**

Images can be classified based on the domains and applications where such images are encountered.

**Range Images**

Range images are often encountered in computer vision. In range images, the pixel values denote the distance between the object and the camera. These images are also referred to as depth images. This is in contrast to all other images whose pixel values denote intensity and hence are often known as intensity images.

**Multispectral Images**

Multispectral images are encountered mostly in remote sensing applications. These images are taken at different bands of visible or infrared regions of the electromagnetic wave. Multispectral images may have many bands that may include infrared and ultraviolet regions of the electromagnetic spectrum.

## Basic Relationship and Distance Matrices

Images can be easily represented as a two-dimensional array of matrix. Pixels can be visualized logically and physically. Logical pixels specify the points of a continuous 2D function. These are logical in the sense that they specify a location but occupy no physical area. Normally, this is represented in the Cartesian first coordinate system. Physical pixels occupy a small amount of space when displayed on the output device. Digitized images indicating physical pixels are represented in the Cartesian fourth coordinate system.

For example, an analog image of size 3x3 is represented in the first quadrant of the Cartesian coordinate system as shown in Fig 4.7.

$$
\begin{array}{r|lll}
2 & f(0,2) & f(1,2) & f(2,2) \\
1 & f(0,1) & f(1,1) & f(2,1) \\
y=0 & f(0,0) & f(1,0) & f(2,0) \\
\hline
 & x=0 & 1 & 2
\end{array}
$$

**Figure 4.7: Analog image f(x,y) in the first quadrant of Cartesian coordinate system**

Figure 4.7 illustrates an image f(x,y) of dimension 3x3, where f(0,0) is the bottom left corner. Since it starts from the coordinate position(0,0), it ends with f(2,2), that is x=0,1,2,…..M-1 and y=0,1,2,…….,N-1. x and y define the dimensions of the image.

In digital image processing, the discrete form of the image is often used. Discrete images are usually represented in the fourth quadrant of the Cartesian coordinate system. A discrete image f(x,y) of dimension 3x3 is shown in Fig. 4.8(a)

Many programming environments including MATLAB starts with an index of (1,1). The equivalent representation of the given matrix is shown in Fig 4.8(b)

$$\begin{array}{c|ccc} & y=0 & y=1 & y=2 \\ \hline x=0 & f(0,0) & f(0,1) & f(0,2) \\ x=1 & f(1,0) & f(1,1) & f(1,2) \\ x=2 & f(2,0) & f(2,1) & f(2,2) \end{array} \qquad \begin{array}{c|ccc} & y=1 & y=2 & y=3 \\ \hline x=1 & f(1,1) & f(1,2) & f(1,3) \\ x=2 & f(2,1) & f(2,2) & f(2,3) \\ x=3 & f(3,1) & f(3,2) & f(3,3) \end{array}$$

(a) (b)

**Figure 4.8: Discrete image (a) Image in the fourth quadrant of Cartesian coordinate system (b) Image coordinates as handled by software environments such as MATLAB**

The coordinates used for discrete image is, by default, the fourth quadrant of the Cartesian system.

## Image Topology

Image topology is a branch of image processing that deals with the fundamental properties of the image such as image neighbourhood, paths among pixels, boundary, and connected components. It characterizes the image with topological properties such as neighbourhood, adjacency and connectivity. **Neighbourhood** is fundamental to understanding image topology. Neighbours of a given reference pixel are those pixels with which the given reference pixel shares its edges and corners.

In $N_4(p)$, the reference pixel p(x,y) at the coordinate position (x,y) has two horizontal and two vertical pixels as neighbours. This is shown graphically in Fig. 4.9.

$$\begin{bmatrix} 0 & X & 0 \\ X & p(x,y) & X \\ 0 & X & 0 \end{bmatrix}$$

**Figure 4.9: 4-Neighbourhood $N_4$(p)**

The set of pixels {(x+1,y),(x-1,y),(x,y+1),(x,y-1)}, called the **4-neighbours** of p is denoted as **$N_4$(p)** .Thus, 4-neighbourhood includes the four direct neighbours of the pixel p(x,y). The pixel may have four diagonal neighbours. They are (x-1,y-1), (x+1,y+1),(x-1,y+1) and (x+1,y-1). The diagonal pixels for the reference pixel p(x,y) are shown graphically in Fig 4.10.

$$\begin{bmatrix} X & 0 & X \\ 0 & p(x,y) & 0 \\ X & 0 & X \end{bmatrix}$$

**Figure 4.10: Diagonal elements $N_D$(p)**

The diagonal neighbours of pixel p(x,y) are represented as $N_D$ (p). The 4-neighbourhood and $N_D$ are collectively called the **8-neighbourhood**. This refers to all the neighbours and pixels that share a common corner with the reference pixel p(x,y). These pixels are called indirect neighbours. This is represented as $N_8$(p) and is shown graphically in Fig 4.11.

The set of pixels $N_8(x)=N_D(x) \cup N_4(x)$

$$\begin{bmatrix} X & X & X \\ X & p(x,y) & X \\ X & X & X \end{bmatrix}$$

**Figure 4.11: 8-Neighbourhood $N_8$(p)**

## Connectivity

The relationship between two or more pixels is defined by **pixel connectivity**. Connectivity information is used to establish the boundaries of objects. The pixels p and q are said to be connected if certain conditions on pixel brightness specified by the set V and spatial adjacency are satisfied. For a binary image, this set V will be {0,1} and for grey scale images, V might be any range of grey levels.

**4-Connectivity** : The pixels p and q are said to be in 4-connectivity when both have the same values as specified by the set V and if q is said to be in the set $N_4$(p). This implies any path from p to q on which every other pixel is 4-connected to the next pixel.

**8-Connectivity** : It is assumed that the pixels p and q share a common grey scale value. The pixels p and q are said to be in 8-connectivity if q is in the set $N_8$(p)

**Mixed Connectivity:** Mixed connectivity is also known as m-connectivity. Two pixels p and q are said to be in m-connectivity when

1. q is in $N_4(p)$
2. q is in $N_D(p)$ and the intersection of $N_4(p)$ and $N_4(q)$ is empty.

For example, Fig 4.12 shows 8-connectivity when V={0,1}

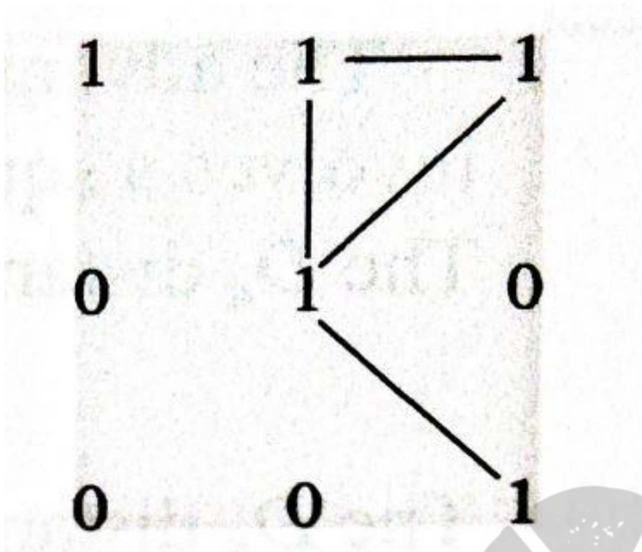



**Figure 4.12 : 8-connectivity represented as lines**

8- Connectivity is shown as lines. Here, a multiple path or loop is present. In m-connectivity, there are no such multiple paths. The m-connectivity for the image in Fig 4.12 is as shown in Fig 4.13.

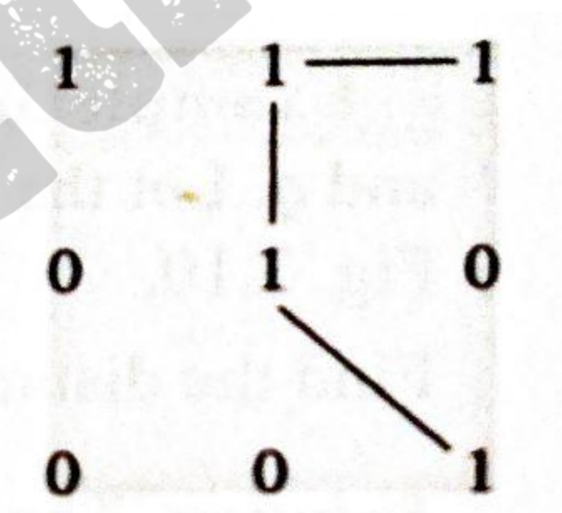



**Figure 4.13 : m-Connectivity**

It can be observed that the multiple paths have been removed.

## Relations

A binary relation between two pixels a and b, denoted as aRb, specifies a pair of elements of an image.

For example, consider the image pattern given in Fig 4.14. The set is given as A $=\{x_1,x_2,x_3\}$. The set based on the 4-connecivity relation is given as A$=\{x_1,x_2\}$. It can be observed that x3 is ignored as it is not connected to any other element of the image by 4-connectivity.

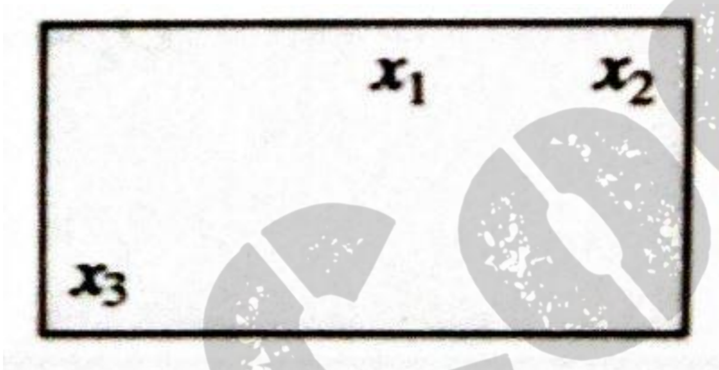



**Figure 4.14 : Image pattern**

The following are the properties of the binary relations:

**Reflexive:** For any element a in the set A, if the relation aRa holds, this is known as a reflexive relation.

**Symmetric:** If aRb implies that bRa also exists, this is known as a symmetric relation.

**Transitive :** If the relation aRb and bRc exist, it implies that the relationship aRc also exists. This is called the transitivity property.

If all these three properties hold, the relationship is called an equivalence relation.

## Distance Measures

The distance between the pixels p and q in an image can be given by distance measures such as **Euclidian distance, D4 distance** and **D8 distance**. Consider three pixels p,q, and z. If the coordinates of the pixels are P(x,y),Q(s,t) and Z(u,w) as shown in Fig.4.15, the distances between the pixels can be calculated.

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 1(z) |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1(q) |
| 1 | 1 | 1 | 1 |
| (p) | | | |

**Figure 4.15 : Sample image**

The distance function can be called metric if the following properties are satisfied:

1. D(p,q) is well-defined and finite for all p and q.
2. D(p,q) ≥0 if p=q, then D(p,q)=0
3. The distance D(p,q)=D(q,p)
4. D(p,q)+D(q,z) ≥ D(p,z). This is called the property of triangular inequality.

The Euclidean distance between the pixels p and q, with coordinates (x,y) and (s,t) respectively can be defined as

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

The advantage of the Euclidean distance is its simplicity. However, since its calculation involves a square root operation, it is computationally costly.

The $D_4$ distance or city block distance can be simply calculated as

$D_4(p,q)=|x-s|+ |y-t|$

The $D_8$ distance or chessboard distance can be calculated as

$D_8(p,q)=\max(|x-s|,|y-t|)$

## Important Image Characteristics

Some important characteristics of images are as follows:

1. The set of pixels that has connectivity in a binary image is said to be characterized by the connected set.
2. A digital path or curve from pixel p to another pixel q is a set of points $p_1,p_2,\ldots,p_n$. If the coordinates of those points are $(x_0,y_0)$ ,$(x_1,y_1),\ldots\ldots(x_n,y_n)$, then $p=(x_0,y_0)$ and $q=(x_n,y_n)$. The number of pixels is called the length. If $x_0=x_n$ and $y_0=y_n$ , then the path is called a closed path.
3. R is called a region if it is a connected component.
4. If a path between any two pixels p and q lies within the connected set S, it is called a connected component of S. If the set has only one connected component, then the set S is called a connected set. A connected set is called a region.
5. Two Regions $R_1$ and $R_2$ are called adjacent if the union of these sets also forms a connected component. If the regions are not adjacent, it is called disjoint set. In Fig 4.16, two regions $R_1$ and $R_2$ are shown. These regions are 8-connected because the pixels (underlined pixel ‘1’) have 8-connectivity. If the regions are not adjacent, they are called disjoint.
6. The border of the image is called contour or boundary. A boundary is a set of pixels covering a region that has one or more neighbours outside the region. Typically, in a binary image, there is a foreground object and a background object. The border of the foreground object may have at least one neighbor in the background. If the border pixels are within the region itself, it is called inner boundary. This need not be closed.
7. Edges are present whenever there is an abrupt intensity change among pixels. Edges are similar to boundaries, but may or may not be connected. If edges are disjoint, they have to be linked together by edge linking algorithms. However boundaries are global and have a closed path. Figure 4.17 illustrates two regions and an edge. It can be observed that edges provide an outline of the object. The pixels that are covered by the edges lead to regions.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| Region 1 | | | Region 2 | | |

**Figure 4.16 : Neighbouring regions**

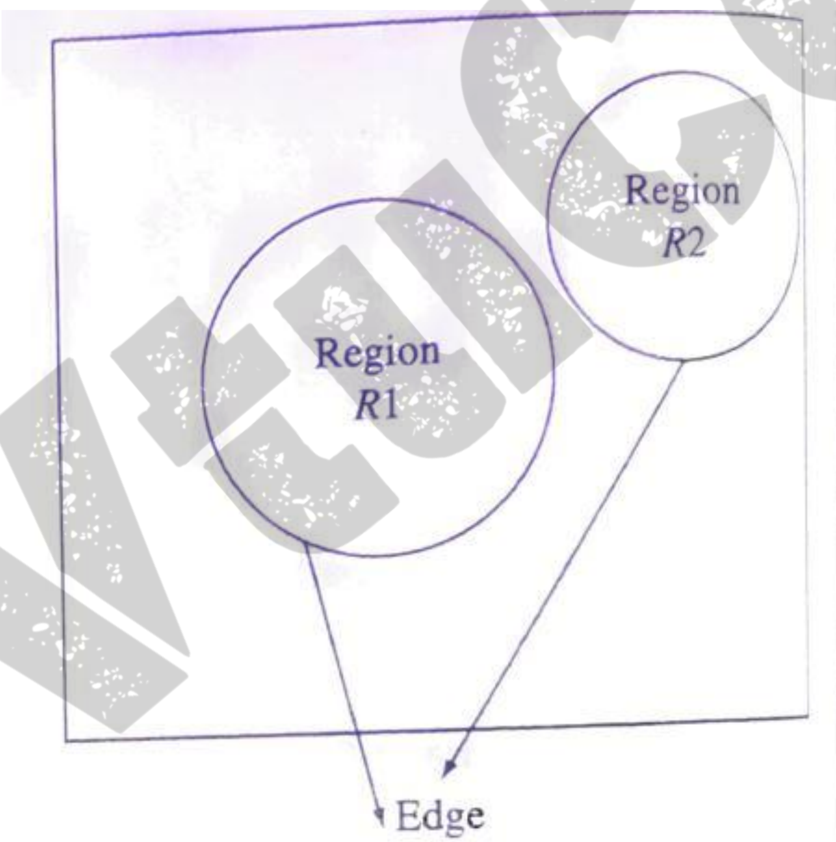



**Figure 4.17 : Edge and regions**

## Classification of Image Processing Operations

There are various ways to classify image operations. The reason for categorizing the operations is to gain an insight into the nature of the operations, the expected results, and the kind of computation burden that is associated with them.

One way of categorizing the operations based on neighbourhood is as follows:

1. Point Operations
2. Local Operations
3. Global Operations

Point operations are those whose output value at a specific coordinate is dependent only on the input value. A local operation is one whose output value at a specific coordinate is dependent on the input values in the neighbourhood of that pixel. Global operations are those whose output value at a specific coordinate is dependent on all the values in the input image.

Another way of categorizing operations is as follows:

1. Linear Operations
2. Non-linear Operations

An operator is called a linear operator if it obeys the following rules of additivity and homogeneity.

**1. Property of additivity**

$$H(a_1f_1(x,y)+a_2f_2(x,y))= H(a_1f_1(x,y)) + H(a_2f_2(x,y))$$

$$= a_1H(f_1(x,y)) + a_2H(f_2(x,y))$$

$$=a_1 \times g_1(x,y)+a_2 \times g_2(x,y)$$

**2. Property of homogeneity**

$$H(kf_1(x,y))=kH(f_1(x,y))=kg_1(x,y)$$

A non-linear operator does not follow these rules.

Image operations are array operations. These operations are done on a pixel-by-pixel basis. Array operations are different from matrix operations. For example, consider two images

$F_1 = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$

$F_2 = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$

The multiplication of $F_1$ and $F_2$ is element-wise, as follows:

$F_1 \times F_2 = \begin{bmatrix} AE & BF \\ CG & HD \end{bmatrix}$

By default, image operations are array operations only.

## Arithmetic Operations

Arithmetic operations include image addition, subtraction, multiplication, division and blending.

**Image Addition**

Two images can be added in a direct manner, as given by

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

The pixels of the input images $f_1(x,y)$ and $f_2(x,y)$ are added to obtain the resultant image $g(x,y)$.

Figure 4.18 shows the effect of adding a noise pattern to an image. However, during the image addition process, care should be taken to ensure that the sum does not cross the allowed range. For example, in a grey scale image, the allowed range is 0-255, using eight bits. If the sum is above the allowed range, the pixel value is set to the maximum allowed value. Similarly, it is possible to add a constant value to a single image, as follows:

$$\mathbf{g(x,y)=f_1(x,y)+k}$$

If the value of k is larger than 0, the overall brightness is increased. Figure 4.18(d) illustrates the addition of the constant 50 increases the brightness of the image.

The brightness of an image is the average pixel intensity of an image. If a positive or negative constant is added to all the pixels of an image, the average pixel intensity of the image increases or decreases respectively. The practical application of image addition is as follows:

1. To create double exposure. Double exposure is the technique of superimposing an image on another image to produce the resultant. This gives a scenario equivalent to exposing a film to two pictures.
2. To increase the brightness of an image

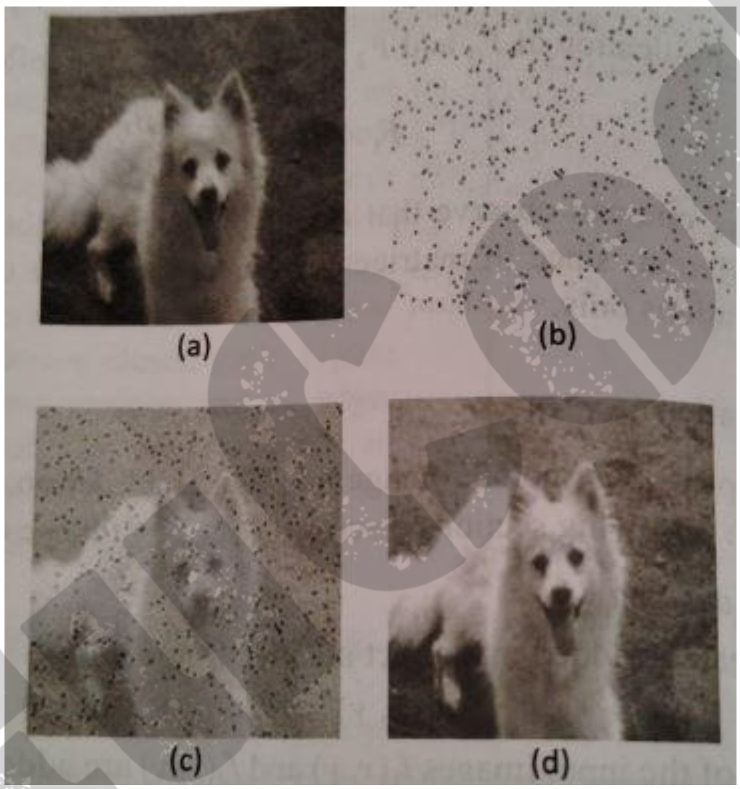



**Figure 4.18: Results of the image addition operations (a) Image1 (b) Image 2 (c) Addition of images 1 and 2 (d) Addition of image 1 and constant 50**

## Image Subtraction

The subtraction of two images can be done as follows. Consider

$$g(x,y)=f_1(x,y)-f_2(x,y)$$

where $f_1(x,y)$ and $f_2(x,y)$ are two input images and $g(x,y)$ is the output image. To avoid negative values, it is desirable to find the modulus of the difference as

$$g(x,y) = | f_1(x,y) - f_2(x,y) |$$

It is also possible to subtract a constant value k from the image

i.e $g(x,y) = | f_1(x,y) - k |$, as k is constant. The decrease in the average intensity reduces the brightness of the image. Some of the practical applications of image subtraction are as follows:

1. Background elimination
2. Brightness reduction
3. Change detection

If there is no difference between the frames, the subtraction process yields zero, and if there is any difference, it indicates the change. Figure 4.19 (a) -4.19(d) show the difference between the images. In addition, it illustrates that the subtraction of a constant results in a decrease of the brightness.

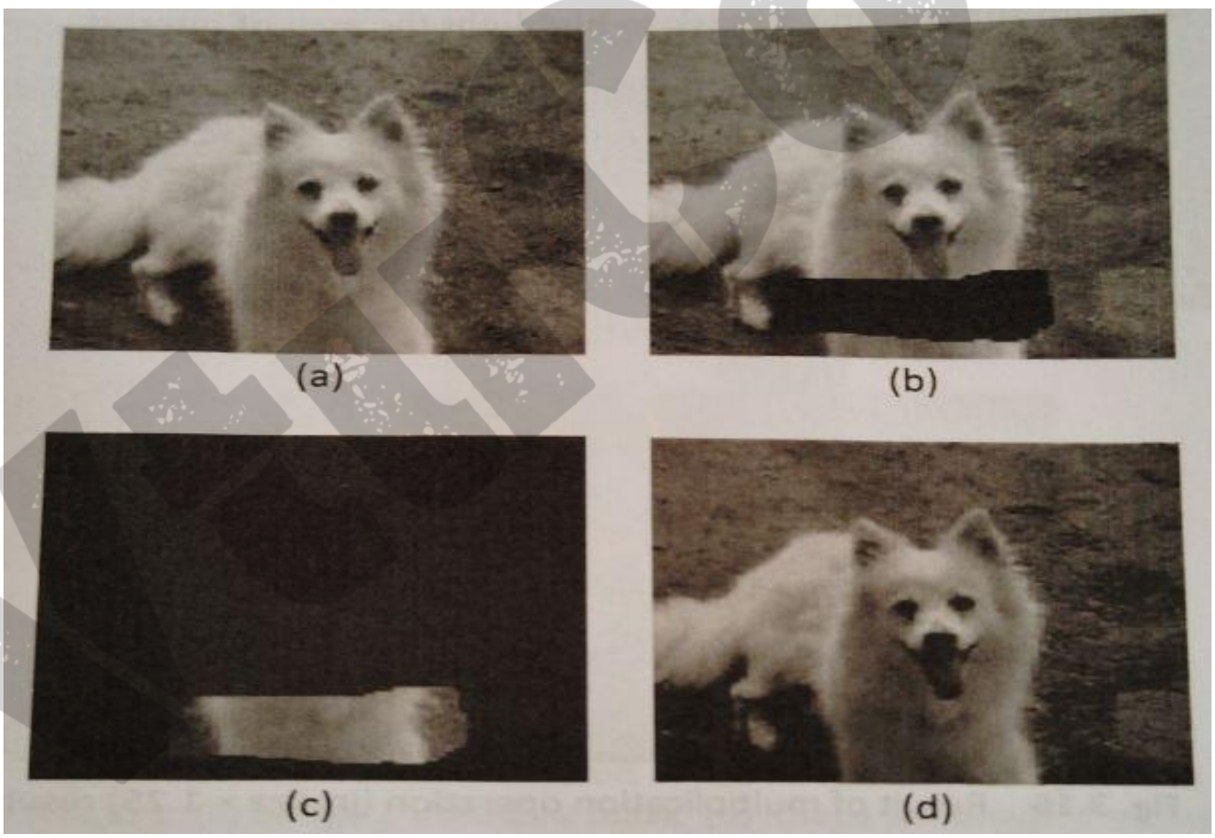



**Figure 4.19: Results of the image subtraction operation (a) Image 1 (b) Image 2 (c) Subtraction of images 1 and 2 (d) Subtraction of constant 50 from image 1**

**Image Multiplication**

Image multiplication can be done in the following manner:

Consider

$$\mathbf{g(x,y)=f_1(x,y)\ x\ f_2(x,y)}$$

$f_1(x,y)$ and $f_2(x,y)$ are two input images and $g(x,y)$ is the output image. If the multiplied value crosses the maximum value of the data type of the images, the value of the pixel is reset to the maximum allowed value. Similarly, scaling by a constant can be performed as

$$\mathbf{g(x,y)=f(x,y)x\ k}$$

where k is a constant.

If k is greater than 1, the overall contrast increases. If k is less than 1, the contrast decreases. The brightness and contrast can be manipulated together as

$g(x,y)=af(x,y)+k$

Parameters a and k are used to manipulate the brightness and contrast of the input image. g(x,y) is the output image. Some of the practical applications of image multiplication as follows:

1. It increases contrast. If a fraction less than 1 is multiplied with the image, it results in decrease of contrast. Figure 4.20 shows that by multiplying a factor of 1.25 with the original image, the contrast of the image increases.
2. It is useful for designing filter masks.
3. It is useful for creating a mask to highlight the area of interest.

**Figure 4.20: Result of multiplication operation (image x 1.25) resulting in good contrast**

## Image Division

Division can be performed as

$$g(x,y) = f_1(x,y)/f_2(x,y)$$

where $f_1(x,y)$ and $f_2(x,y)$ are two input images and $g(x,y)$ is the output image.

The division process may result in floating-point numbers. Hence, the float data type should be used in programming. Improper data type specification of the image may result in loss of information. Division using a constant can also be performed as

$$g(x,y) = f(x,y)/k \quad \text{where } k \text{ is a constant}$$

Some of the practical applications of image division is as follows:

1. Change detection
2. Separation of luminance and reflectance components
3. Contrast reduction

Figure 4.21 (a) shows such an effect when the original image is divided by 1.25.

Figure 4.21 (b)-4.21(e) show the multiplication and division operations used to create a mask. It can be observed that image 2 is used as a mask. The multiplication of image 1 with image 2 results in highlighting certain portions of image 1 while suppressing the other portions. It can be observed that division yields back the original image.

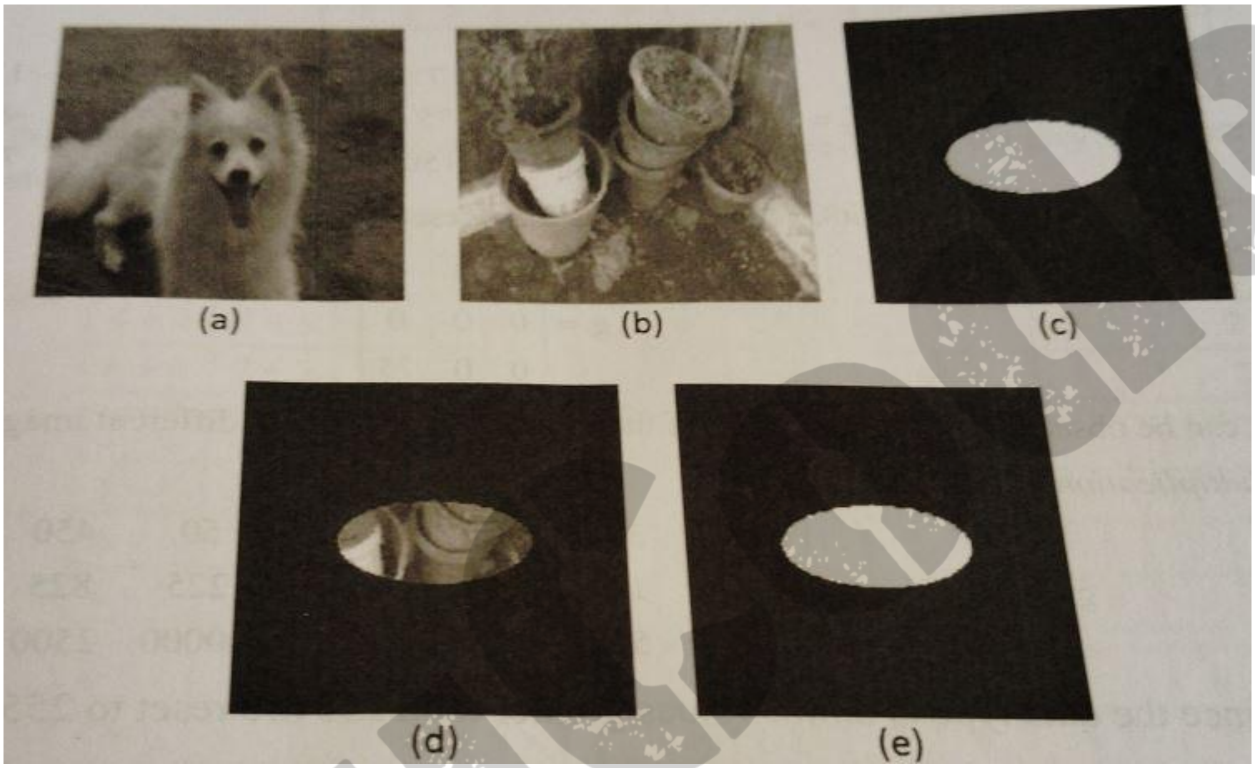



**Figure 4.21: Image division operation (a) Result of the image operation (image/1.25) (b) Image 1 (c) Image 2 used as a mask (d) Image 3=image 1 x image 2 (e) Image 4=image 3/image 1**

## Applications of Arithmetic Operations

Arithmetic operations can be combined and put to effective use. For example, the image averaging process can be used to remove noise. Noise is a random fluctuation of pixel values, which affects the quality of the image. A noisy image can be considered as an image plus noise:

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

where f(x,y) is the input image and g(x,y) is the output image. Several instances of noisy images can be averaged as

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

where M is the number of noisy images. As M increases the averaging process reduces the intensity of the noise and it becomes so low that it can automatically be removed. As M becomes large, the expectation

$$E\{\bar{g}(x, y)\} = f(x, y)$$

## Logical Operations

Bitwise operations can be applied to image pixels. The resultant pixel is defined by the rules of the particular operation. Some of the logical operations that are widely used in image processing are as follows:

1.AND/NAND

2. OR/NOR

3. EXOR/EXNOR

4. Invert/ Logical NOT

**1. AND/NAND**

The truth table of the AND and NAND operators is given in Table 4.2

**Table 4.2: Truth table of the AND and NAND operators**

| A | B | C(AND) | C(NAND) |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

The operators AND and NAND take two images as input and produce one output image. The output image pixels are the output of the logical AND/NAND of the individual pixels. Some of the practical applications of the AND and NAND operators are as follows:

1. Computation of the intersection of images
2. Design of filter masks
3. Slicing of grey scale images; for example, the pixel value of the grey scale image may be 1100 0000. The first bits of the pixels of an image constitute one slice. To extract the first slice, a mask of value 1000 0000 can be designed. The AND operation of the image pixel and the mask can extract the first bit and first slice of the image.

Figure 4.22 (a) -4.22(d) shows the effect of the AND and OR logical operators. It illustrates that the AND operator shows overlapping regions of the two input images and the OR operator shows all the input images with their overlapping.


**Figure 4.22: Results of the AND and OR logical operators (a) Image 1 (b) Image 2 (c) Result of image 1 OR image 2 (d) Result of image 1 and AND image 2**

**2. OR/NOR**

The truth table of the OR and NOR operators is given in Table 4.3

**Table 4.3: Truth table of the OR and NOR operators**

| A | B | C(OR) | C(NOR) |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

The practical applications of the OR and NOR operators are as follows:

1. OR is used as the union operator of two images.
2. OR can be used as a merging operator

**3. XOR/NOR**

The truth table of the XOR and XNOR operators is given in Table 4.4.

| **A** | **B** | **C(XOR)** | **C(XNOR)** |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

The practical applications of the XOR and XNOR operations are as follows:

1. Change detection
2. Use as a subcomponent of a complex imaging operation. XOR for identical inputs is zero. Hence it can be observed that the common region of image 1 and image 2 in Figures 4.22(a) and 4.22 (b), respectively, is zero and hence dark. This is illustrated in Fig. 4.23.

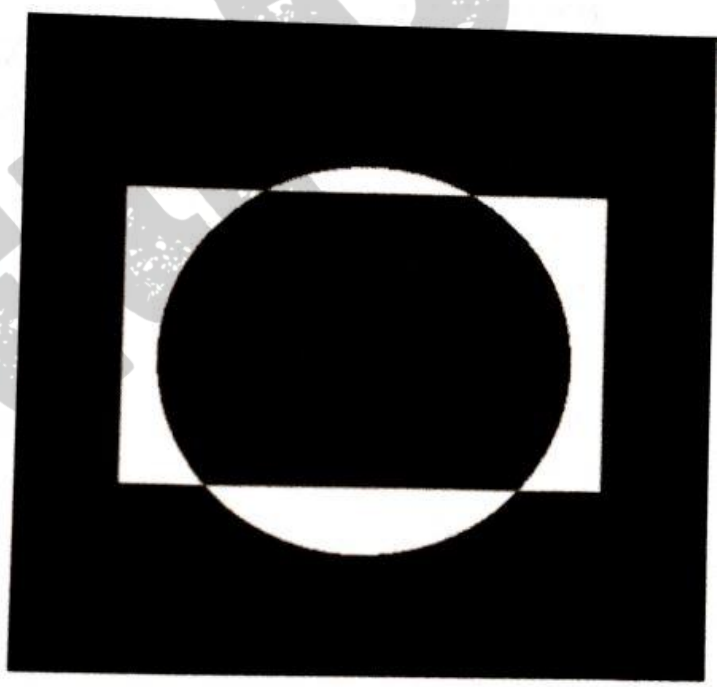


**Figure 4.23: Result of the XOR operation**

## 4. Invert/Logical NOT

The truth table of the NOT operator is given in Table 4.5.

**Table 4.5: Truth table of the OR and NOR operators**

| A | C(NOT) |
|---|---|
| 0 | 1 |
| 1 | 0 |

For grey scale values, the inversion operation is described as

$$g(x,y) = 255 - f(x,y)$$

The practical applications of the inversion operator are as follows:

1. Obtaining the negative of an image. Figure 4.24 shows the negative of the original image shown in Fig. 4.22(a)
2. Making features clear to the observer
3. Morphological processing

(a) (b)

**Figure 4.24: Effect of the NOT operator (a) Original Image (b) NOT of original Image**

Similarly, two images can be compared using operators such as

- = Equal to
- >Greater than
- >= Greater than or equal to
- < Less than

- <= Less than or equal to
- ≠ Not equal to

The resultant image pixel represents the truth or falsehood of the comparisons. Similarly shifting operations are also very useful. Shifting of I bits of the image pixel to the right results in division by $2^I$. Similarly, shifting of I bits of the image pixel to the left results in multiplication by $2^I$.

Shifting operators are helpful in dividing and multiplying an image by a power of two. In addition, this operation is computationally less expensive.

## Geometrical Operations

### Translation

Translation is the movement of an image to a new position. Let us assume that the point at the coordinate position X=(x,y) of the matrix **F** is moved to the new position X′ whose coordinate position is (x’,y’). Mathematically, this can be stated as a translation of a point X to the new position X′. The translation is represented as

$$\mathbf{x' = x + \delta x}$$

$$\mathbf{y' = y + \delta y}$$

The translation of the floor image by (25,25) is shown in Figure 4.25.

**Figure 4.25 : Result of translation by 50 units**

In vector notation, this is represented as **F′=F+T**, where δx and δy are translations parallel to the x and y axes. **F** and **F′** are the original and the translated images respectively. However, other transformations such as scaling and rotation are multiplicative in nature. The transformation process for rotation is given as **F′=RF**, where R is the transform matrix for performing rotation, and the transformation process for scaling is given as **F′=SF**. Here, S is the scaling transformation matrix.

To create uniformity and consistency, it is necessary to use a homogeneous coordinate system where all transformations are treated as multiplications. A point (x,y) in 2D space is expressed as (wx,wy,w) for w≠0. The properties of homogeneous coordinates are as follows:

1. In homogeneous coordinates, at least one point should be non-zero. Thus (0,0,0) does not exist in the homogeneous coordinate system.
2. If one point is multiplicative of the other point, they are same. Thus, the points(1,3,5) and (3,9,15) are same as the second point is 3× (1,3,5)
3. The point (x,y,w) in the homogeneous coordinate system corresponds to the point (x/w,y/w) in 2D space.

In the homogeneous coordinate system, the translation process of point(x,y) of image F′ to the new point (x′,y′) of the image F′ is described as

$x' = x + \delta x$

$y' = y + \delta y$

In matrix form, this can be stated as

$$[x',y',1] = \begin{bmatrix} 1 & 0 & \delta x \\ 0 & 1 & \delta x \\ 0 & 0 & 1 \end{bmatrix} [x,y,1]^T$$

Sometimes, the image may not be present at the origin. In that case, a suitable negative translation value can be used to bring the image to align with the origin.

## Scaling

Depending on the requirement, the object can be scaled. Scaling means enlarging and shrinking. In the homogeneous coordinate system, the scaling of the point (x,y) of the image F to the new point (x′,y′) of the image F′ is described as

$$\mathbf{x' = x \ xS_X}$$

$$\mathbf{y' = y \ xS_y}$$

$[x',y'] = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \; [x,y]$

$S_x$ and $S_y$ are called scaling factors along the x and y axes respectively. If the scale factor is 1, the object would appear larger. If the scaling factors are fractions, the object would shrink. Similarly, if $S_x$ and $S_Y$ are equal, scaling is uniform. This is known as **isotropic scaling**. Otherwise, it is called **differential scaling**. In the homogeneous coordinate system, it is represented as

$[x',y',1] = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \; [x,y,1]^T$

The matrix $S = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is called scaling matrix.

## Mirror or Reflection Operation

This function creates the reflection of the object in a plane mirror. The function returns an image in which the pixels are reversed. This operation is useful in creating an image in the desired order and for making comparisons. The reflected object is of the same size as the original object, but the object is in the opposite quadrant. Reflection is also described as rotation by 180˙. The reflection along the x-axis is given by

$F' = [-x,y] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ x } [x,y]^T$

Similarly, the reflection along the y-axis is given by

$F' = [x,-y] = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ x } [x,y]^T$

Similarly, the reflection about the line y=x is given as

F'=[x,-y]= $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ x $[x,y]^T$

The reflection about the line y=-x is given as

F'=[x,-y]= $\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$ x $[x,y]^T$

The reflection operation is illustrated in Fig 3.22(a) and 3.22(b). In the homogeneous coordinate system, the matrices for reflection can be given as

$$R_{y\text{-axis}}=\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{x\text{-axis}}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{origin}=\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The reflection along a line can be given as

$$R_{y=x}=\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{y=-x}=\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

## Shearing

Shearing is a transformation that produces a distortion of shape. This can be applied either in the x-direction or the y-direction. In this transformation, the parallel and opposite layers of the object are simply sided with respect to each other.

Shearing can be done using the following calculation and can be represented in the matrix form as

$$x'=x+ay$$

$$y'=y$$

$$X_{shear}=\begin{bmatrix}1 & a & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} \quad \text{(where } a=sh_x\text{)}$$

$\mathbf{Y_{shear}}$ can be given as

$$x'=x$$

$$y'=y+bx$$

$$\mathbf{Y_{shear}}=\begin{bmatrix}1 & 0 & 0\\b & 1 & 0\\0 & 0 & 1\end{bmatrix} \quad \text{(where } b=sh_y\text{)}$$

where $\mathbf{sh_x}$ and $\mathbf{sh_y}$ are shear factors in the x an y directions, respectively.

## Rotation

An image can be rotated by various degrees such as $90^0$,$180^0$ or $270^0$. In the matrix form it is given as

$$[x',y']=\begin{bmatrix}cos\theta & -sin\theta\\sin\theta & cos\theta\end{bmatrix}[x,y]^T$$

This can be represented as **F′=RA**. The parameter θ is the angle of rotation with respect to the x-axis. The value of θ can be positive or negative. A positive angle represents counter clockwise rotation and a negative angle represents clockwise rotation. In homogeneous coordinate system, rotation can be expressed as

$$[x',y',1]=\begin{bmatrix}cos\theta & -sin\theta & 0\\sin\theta & cos\theta & 0\\0 & 0 & 1\end{bmatrix}[x,y,1]$$

If θ is substituted with -θ, this matrix rotates the image in the clockwise direction.

## Affine Transform

The transformation that maps the pixel at the coordinates(x,y) to a new coordinate position is given as a pair of transformation equations. In this transform, straight lines are preserved and parallel lines remain unchanged. It is described mathematically as

$x' = T_x(x,y)$

$y' = T_y(x,y)$

$T_x$ and $T_y$ are expressed as polynomials. The linear equation gives an affine transform.

$x' = a_0x+a_1y+a_2$

$y' = b_0x+b_1y+b_2$

This is expressed in matrix form as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a0 & a1 & a2 \\ b0 & b1 & b2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The affine transform is a compact way of representing all transformations. The given equation represents all transformations.

- Translation is the situation where $a_0 = 1$, $a_1 = 0$, and $a_2 = \delta_x$.
- Scaling transformation is a situation where $a_0 = s_x$ and $b_1 = s_y$ and $a_1 = 0$, $a_2 = 0$, $b_0 = 0$ and $b_2 = 0$.
- Rotation is a situation where $a_0 = \cos\theta$, $a_1 = -\sin\theta$, $b_0 = \sin\theta$, $b_1 = \cos\theta$, $a_2 = 0$ and $b_2 = 0$.
- Horizon shear is performed with $a_0 = 1$, $b_0 = 1$, $a_1 = Sh_x$, $a_2 = 0$, $b_1 = Sh_y$ and $b_2 = 0$

## Inverse Transformation

The purpose of inverse transformation is to restore the transformed object to its original form and position. The inverse or backward transformation matrices are given as follows:

Inverse transform for translation= $\begin{bmatrix} 1 & 0 & -\delta x \\ 0 & 1 & -\delta y \\ 0 & 0 & 1 \end{bmatrix}$

Inverse transform for scaling = $\begin{bmatrix} 1 & 0 & -\delta x \\ 0 & 1 & -\delta y \\ 0 & 0 & 1 \end{bmatrix}$

Inverse transform for rotation can be obtained by changing the sign of the transform term. For example, the following matrix performs inverse transform.

$$\begin{bmatrix} cos\theta & +sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3D Transforms

Some medical images such as computerized tomography(CT) and magnetic resonance imaging(MRI) images are three-dimensional images. To apply translation, rotation, and scaling on 3D images, 3D transformations are required. 3D transformations are logical extensions of 2D transformations. These are summarized and described as follows:

Translation = [ ]

Image Interpolation Techniques

Transforms can be of two types. Affine transforms often produce pixels of the resultant image that cannot be fit as some of the pixel values are non-integers and often go beyond the acceptable range. This results in gaps(or holes) and issues related to number of pixels and range. So interpolation techniques are required to solve these issues.

Forward Mapping

Forward mapping is the process of applying transformations iteratively to every pixel in the image, yielding a new coordinate position and copying the values of the pixel to a new position.

Backward Mapping

Backward mapping is the process of checking the pixels of the output image to determine the position of the pixels in the input image. This is used to guarantee that all the pixels of the input image are processed.

During the process of both forward and backward mapping, it may happen that pixels cannot be fitted into the new coordinates. For example, consider the process of rotation of a point(10,5) by $45^0$. This yields

$x' = x\cos\theta - y\sin\theta$

$= 10\cos(45^0) - 5\sin(45^0)$

$= 10(0.707) - 5(0.707)$

$= 3.535$

$y' = x\sin\theta + y\cos\theta$

$= 10 \sin(45^0)+5\cos(45^0)$

$= 10(0.707)+5(0.707)$

$=10.605$

Since these new coordinate positions are not integers, the rotation process cannot be carried out. Thus, the process may leave a gap in the new coordinate position, which creates poor quality output. Therefore, whenever a geometric transformation is performed, a resampling process should be carried out so that the desirable quality is achieved in the resultant image. The resampling process creates new pixels so that the quality of the output is maintained. In addition, the rounding off of the new coordinate position (3.535,10.605) should be carried out as (4,11). This process of fitting the output to the new coordinates is called interpolation.

Interpolation is the method of calculating the expected values for a function with known pixels. Some of the popular interpolation techniques are:

Nearest neighbor technique

Bilinear technique

Bicubic technique

The most elementary form of interpolation is nearest neighbor interpolation or zero-order interpolation. This technique determines the closest pixel and assigns it to every pixel in the new image matrix, that is, the brightness of the pixels is equal to the closest neighbor. Sometimes, this may result in pixel blocking and can degrade the resulting image, which may appear spatially disordered. These distortions are called aliasing.

A more accurate interpolation scheme is bilinear interpolation. This is called first-order interpolation. Four neighbours of the transformed original pixels that surround the new pixel are obtained and are used to calculate the new pixel value. Linear interpolation is used in both the directions. Weights are assigned based on the proximity. Then the process takes the weighted average of the brightness of the four pixels that surround the pixels of interest.

$g(x,y)=(1-a)(1-b)f(x',y')+(1-a)bf(x',y'+1)+a(1-b)f(x'+1,y')+abf(x'+1,y'+1)$

Here g(x,y) is the output image and f(x,y) is the image that undergoes the interpolation operation. If the desired pixel is very close to one of the four nearest neighbor pixels, its weight will be much higher. This technique leads to blurring of the edges.However, it reduces aliasing artefacts.

High-order interpolation schemes takes more pixels into account. Second-order interpolation is known as cubic interpolation. It uses a neighbourhood of 16 pixels. Then it fits two polynomials to the 16 pixels of the transformed original matrix and the new image pixel. This technique is very effective and produces images that are very close to the original. In extreme cases, more than 64 neighbouring pixels can be used. However, as the number of pixel increases, the computational complexity also increases.

Set Operations

An image can be visualized as a set. For example, the following binary image(Fig3 3.25) can be visualizd as a set A={(0,0),(0,2),(2,2)}. The coordinates values represent the value of 1. Set operators can then be applied to the set to get the resultant, which is useful for image analysis.

The complement of set A can be defined as the set of pixels that does not belong to the set A.

$A^c=\{c/c\notin A\}$

The reflection of the set is defined as

$A=\{c=-a,\ a\in A\}$

The union of two sets, A and B can be represented as

$A\cup B=\{c/(c\in A)\vee(c\in B)\}$

Where the pixel c belongs to A,B or both.

The intersection of two sets is given as $A\cap B=\{c/(c\in A)\wedge(c\in B)\}$. The pixel c belongs to A,B or both.

The difference can be expressed as

$A-B=\{c/(c\in A)\wedge(c\notin B)\}$

Which is equivalent to $A\cap B^C$

Morphology is a collection of operations based on set theory, to accomplish various tasks such as extracting boundaries, filling small holes present in the image, and removing noise present in the image.

Mathematical morphology is a very powerful tool for analyzing the shapes of the objects hat are present in the images. The theory of mathematical morphology is based on set theory. One can visualize a binary object as a set. Set theory can then be applied to the sample set. Morphological operators often take a binary image and a mask known as structuring element as input. The set operators such as intersection, union, inclusion and complement can then be applied to images. Dilation is one of the two basic operators. It can be applied to binary as well as grey scale images. The basic effect of this operator on a binary image is that it gradually increases the boundaries of the region, while the small holes that are present in the images become smaller.

Let us assume that A and B are a set of pixel coordinates. The dilation of A by B can be denoted as

$A\oplus B=\{(x,y)+(u,v):\ (x,y)\in A,(u,v)\in B\}$

Where x and y corresponds to the set A, and u and v corresponds to the set B. The coordinates are added and the union is carried out to create the resultant set. These kinds of operations are based on Minkowski algebra.

Statistical Operations

Statistics plays an important role in image processing. An image can be assumed to be a set of discrete points. Statistical operations can be applied to the image to get the desired results such as manipulation of brightness and contrast.Some of the very useful statistical operations include mean,median, mode and mid-range. These measures are useful in image processing. The measures of data dispersion also includes quartiles,inter-quartile range and variance.

Some of the frequently used statistical measures are the following

Mean

Mean is the average of all the values in the sample(population) and is denoted as

The overall brightness of the grey scale image is measured using the mean. This is calculated by summing all the values of the pixels of an image and dividing it by the number of pixels in the image.

$$\mu = \frac{1}{n}\sum_{i=0}^{n-1} I_i$$

Sometimes the data is associated with a weight. This is called weighted mean. The problem of mean is its extreme sensitivity to noise. Even small changes in the input affect the mean drastically.

Median

Median is the value where the given $X_i$ is divided into two equal halves, with half of the values being lower than the median and the other half higher. The procedure for obtaining the median is to sort the values of the given $X_i$ in ascending order. If the given sequence has an odd number of values, the middle value is the median. Otherwise, the median is the arithmetic mean of the two middle values.

Mode

Mode is the value that occurs most frequently in the dataset. The procedure for finding the mode is to calculate the frequencies for all of the values in the data. The mode is the value(or values) with the highest frequency. Normally, based on the mode, the dataset is classified as unimodal, bimodal,and trimodal. Any dataset that has two modes is called bimodal.

Percentile

Percentiles are data that are less than the coordinate by some percentage of the total value. For example, the median is the $50^{th}$ percentile and can be denoted as $Q_{0.50}$.The $25^{th}$ percentile is called the first quartile and the $75^{th}$ percentile is called third quartile. Another measure that is useful to measure dispersion is the inter-quartile range. The inter-quartile is defined as $Q_{0.75}$ $Q_{0.25}$.Semi quartile range is =0.5 Xiqr

Unimodal curves are slightly skewed and the empirical relation is

Mean-Mode=3x(Mean-Median)

The interpretation of the formula is that the mode for the unimodal frequency curve is moderately skewed. The mid-range is also used to assess the central tendency of the dataset. In a normal distribution, the mean, median, and mode are the same. In symmetrical distributions, it is possible for the mean and median to be the same even though there may be several modes. By contrast, in asymmetrical distributions, the mean and median are not the same. These distributions are said to be skewed data where more than half the cases are either above or below the mean.

Standard Deviation and Variance

The most commonly used measures of dispersion are variance and standard deviation. The mean does not convey much more than a middle point. For example, the following datasets {10,20,30} and {10,50,0}, both have a mean of 20. The difference between these two sets is the spread of data. Standard deviation is the average distance from the mean of the dataset to each point. The formula for standard deviation is given by

Sometimes, we divide the value by N-1 instead of N. The reason is that in a larger, real-world scenario, division by N-1 gives an answer that is closer to the actual value. In image processing, it is a measure of how much a pixel varies from the mean value of the image. The mean value and the standard deviation characterize the perceived brightness and contrast of the image. Variance is another measure of the spread of the data. It is the square of standard deviation. While standard deviation is a more common measure, variance also indicates the spread of the data effectively.

Entropy

This is the measure of the amount of orderliness that is present in the image. The entropy can be calculated by assuming that the pixels are totally uncorrelated. An organized system has low entropy and a complex system has a very high entropy. Entropy also indicates the average global information content. Its unit is bits per pixel. It can be computed using the formula

Entropy H=-$\sum_{i=1}^{n} log2pi$

Where $p_i$ is the prior probability of the occurrence of the message. Let us consider a binary image, where the pixel assumes only two possible states,0 or 1 and the occurrence of each state is equally likely. Hence, the probability is ½. Therefore, the entropy H=-[1/2log(1/2)+1/2log(1/2)]=1 bit

Therefore, 1 bit is sufficient to store the intensity of the pixel.Therefore, binary images are less complex.

Thus, entropy indicates the richness of the image. This can be seen visually using a surface plot where pixel values are plotted as a function of pixel position.

Convolution and Correlation Operations

The imaging system can be modeled as a 2D linear system.Let f(x,y) and g(x,y) represent the input and output images, respectively. Then, they can be written as g(x,y)=t*(f(x,y)).Convolution is a group process, that is, unlike point operations, group processes operate on a group of input pixels to yield the result. Spatial convolution is a method of taking a group of pixels in the input image and computing the resultant output image. This is also known as a finite impulse response(FIR) filter. Spatial convolution moves across pixel by pixel and produces the output image. Each pixel of the resultant image is dependent on a group of pixels(called kernel).

The one-dimensional convolution formula is as follows:

g(x)=t*f(x)

$=\sum_{i=-n}^{n} t(i)f(x-i)$

The convolution window is a sliding window that centres on each pixel of the image to generate the resultant image. The resultant pixel is thus calculated by multiplying the weight of the convolution mask by pixel values and summing these values. Thus, the sliding window is moved for every corresponding pixel in the image in both direction. Therefore, convolution is called 'shift-add-multiply' operation.

To carry out the process of convolution, the template or mask is first rotated by $180^0$. Then the convolution process is carried out. Consider the process of convolution of two sequences – F, whose dimension is 1 x 5 and a kernel or template T, whose dimension is 1x3.

Let F={0,0,2,0,0} and the kernel be {7 5 1}. The template has to be rotated by $180^0$. The rotated mask of this original mask [7 5 1] is a convolution template whose dimensions is 1 x 3 with value {1,5,7}

To carry out the convolution process first, the process of zero padding should be carried out. Zero padding is the process of creating more zeros and is done as shown in Table 3.7.

Convolution is the process of shifting and adding the sum of the product of mask coefficients and the image to give the centre value.

Correlation is similar to the convolution operation and it is very useful in recognizing the basic shapes in the image. Correlation reduces to convolution if the kernels are symmetric. The difference between the correlation and convolution processes is that the mask or template is applied directly without any prior rotation., as in the convolution process.

The correlation of these sequences is carried out to observe the difference between these processes. The correlation process also involves the zero padding process.