



# NATIONAL INSTITUTE OF TECHNOLOGY AGARTALA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

---

## Digital Image Processing :

---

A CONCEPT NOTE

*Author*

Jashaswimalya Acharjee  
Undergraduate Student, CSE Dept.

*Instructor*

Dr. Parthasarathi De  
Assistant Professor, CSE Dept.

# Contents

# Chapter 1

## Introduction

### 1.1 Image

An image is a two-dimensional function that represents a measure of some characteristics such as brightness or color of a viewed scene. An image is projection of a 3D scene into a 2D projection plane. It can be defined as a two-variable function  $f(x, y)$  where for each position  $(x, y)$  in the projection plane,  $f(x, y)$  defines the light intensity at this point.

#### 1.1.1 Analog Image

An Analog Image can be mathematically represented as a continuous range of values representing positions and intensity. An Analog Image is characterised by a physical magnitude varying continuously in space.

E.g.: The image produced on the screen of a CRT monitor is analog in nature.

#### 1.1.2 Digital Image

A Digital Image is composed of picture elements called *pixels*. Pixels are the smallest sample of an image. A pixel represents the brightness at one point. Conversion of an analog image into a digital image involves two important operations, namely, sampling and quantisation, which are illustrated in Fig: ??.

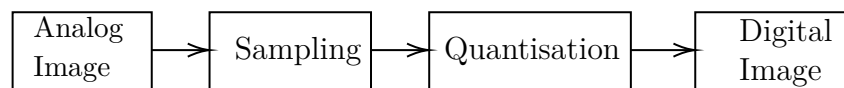


Figure 1.1: Digital Image from Analog Image

#### Advantages of Digital Images:

The advantages of Digital Images are summarised below:

- The Processing of images is faster and cost-effective.
- Digital Images can be effectively stored and efficiently transmitted from one place to another.

- When shooting a digital image, one can immediately see if the image is good or not.
- Copying a digital image is easy. The quality of digital image will not be degraded even if it is copied for several times.
- Whenever the image is in digital format, the reproduction of the image is both faster and cheaper.
- Digital technology offers plenty of scope for versatile image manipulation.

### Drawbacks of Digital Images:

Some of the drawbacks of Digital Images are summarised below:

- Misuse of copyright has become easier because images can be copied from Internet just by clicking the mouse a couple of times.
- A digital file cannot be enlarged beyond a certain size without compromising on quality.
- The memory required to store and process good-quality digital images is very high.
- For real-time implementation of digital-image-processing algorithms, the processor has to be very fast because the volume of data is very high.

## 1.2 Digital Image Processing

The processing of an image by means of a computer is generally termed as **Digital Image Processing**.

The Advantages of computers for the processing of images are summarised below:

- **Flexibility and Adaptability:** The main advantage of digital computers when compared to analog electronic and optical information processing devices is that no hardware modifications are necessary in order to reprogram digital computers to solve different tasks. This feature makes digital computers an ideal devices for processing image signals adaptively.
- **Data Storage and Transmission:** With the development of different image-compression algorithms, the digital data can be effectively stored. The digital data within computers can be easily transmitted from one place to another.

The only limitation of Digital Image and Digital Image Processing are memory and processing speed capabilities of computers. Different Image-Processing techniques include *Image Enhancements*, *image restoration*, *image fusion* and *image watermarking*.

## 1.3 Digital Image Representation

A digital image is a two-dimensional discrete signal. A digital image is also an  $N \times N$  array of elements. Each element in the array is a number which represents the sampled intensity. For e.g. : The representation of a  $4 \times 4$  image in matrix format is shown in Fig: ??.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 1.2: Digital Image Representation

Converting an image into a digital format can be done either with a digital camera, or by a scanner. Digital images can be created directly on a computer screen. However, it is restricted both in spatial coordinates (sampling) and in its allowed intensities (quantisation).

### 1.3.1 Neighbours of a Pixel

A pixel will have four neighbours if the neighbours exist in the EAST, WEST, NORTH and SOUTH directions. The four neighbours of the pixel 'P' are represented in Fig: ??.

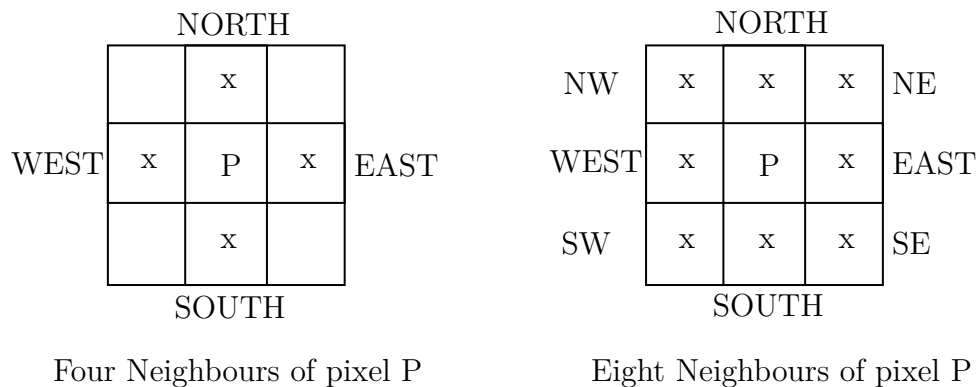


Figure 1.3: Neighbours of Pixels

## 1.4 Classification of Digital Images

Digital Images can be broadly classified into two types and they are:

1. Raster Image
2. Vector Image

### 1.4.1 Raster Image or Bitmap Image

- A raster image file is generally defined as a rectangular array of regularly sampled values known as pixels.
- Scanned graphics and web graphics are the most common forms of raster images.
- Raster images are mapped to grids which are not easily scalable.

- A raster image is resolution dependent because it contains a fixed number of pixels that are used to create the image. Since there are a fixed and limited number of pixels, raster image will lose its quality if it is enlarged beyond that number of pixels as the computer will have to ‘make up’ for the missing information.
- When an image is zoomed by a factor of 24 (say), the clarity is lost.
- Bitmaps are used for photorealistic images, and therefore, involve complex color variations.
- Raster images can show well the gradations of color and detailed images such as photographs.
- The spatial resolution of a raster image is determined by the resolution of the acquisition device and the quality of the original data source.

Common Raster Image formats include: BMP (Windows Bitmap), PCX (Paintbrush), TIFF (Tag Interleave Format), JPEG/JPG (Join Photographics Expert Group), GIF (Graphics Interchange Format), PNG (Portable Network Graphics), PSD (Adobe Photoshop) and CPT (Corel PhotoPaint)

### 1.4.2 Vector Image

- A vector image is defined by objects which are made of lines and curves that are mathematically defined in the computer.
- A vector can have various attributes such as line thickness, length and color.
- Vector images are mathematically defined and hence, they are scalable.
- This implies that vectors can be printed at any size, on any output device, at any resolution, without losing the detail and without altering the resultuion of the image.
- A Vector image, when zoomed by a factor of 24 (say), shows no distortion and clarity is preserved.
- Vector images can be scaled by several factors without altering the resolution of the image.
- Vector images are suitable for typography, line art and illustrations.

## 1.5 Image Types

Images can be broadly classified under four categories:

1. Black & White or Binary Image.
2. Grayscale Image.
3. Color Image.
4. Multispectral Image.

### 1.5.1 Binary Image

- Binary Images take only two values, i.e. either '0' or '1'.
- The brightness graduation cannot be differentiated in Binary Images.
- A Grayscale image can be converted to Binary Image by the threshold operation.
- Geometric Properties of an object can be easily extracted from a binary image.

### 1.5.2 Grayscale Image

- Grayscale images contain only brightness information.
- Each Pixel value in a grayscale image corresponds to an amount or quantity of light.
- The brightness graduation can be differentiated in a grayscale image.
- Each pixel is represented by a 'byte' or 'word'.
- An 8-bit image will have a brightness variation from 0 to 255 where '0' represents Black and '255' represents White.

### 1.5.3 Color Image

- A color image has three values per pixel and they measure the intensity and chrominance of light.
- Each pixel is a vector of color components.
- Common Color spaces are RGB (Red, Green, Blue), HSV (Hue, Saturation, Value) and CMYK (Cyan, Magenta, Yellow, Black).
- The typical uncompressed data rate of a color image is three times the data rate of an uncompressed grayscale image.

### 1.5.4 Volume Image

- A three-dimensional image is an example of volume image.
- The volume image can be obtained from some medical imaging equipment in which the individual data points are called 'voxels'.
- Voxel stands for Volume Pixel.
- A CAT Scan is an example of Volume Image.

### 1.5.5 Range Image

- Range Images are a special class of digital images.
- Each pixel of a range image expresses the distance between a known reference frame and a visible point in the scene.
- The range image reproduces the 3D structure of a scene.
- Range images are also referred as depth images.

## 1.6 Image File Formats

- A digital image is often encoded in the form of binary files for the purpose of storage and transmission.
- A file format is a method used to store digital data and different file formats exist for storing images.
- Different file formats may compress the image data by differing amounts.
- Whatever be the format, the image file consists of two parts:
  1. File Header
  2. Image Data

### Common Image File Formats:

Various image file formats are widely used for the purpose of digital storage and retrieval. Each of these file formats possesses certain characteristics which makes them valid for various applications. These include: GIF, JPEG/JPG, PNG, TIFF, PSD, EPS.

#### 1.6.1 File Header

The Header part contains some of the vital information like format or version identification, width and height of the image, type of the image (binary, grayscale, color image) and image data format which specifies the order in which pixel values are stored in the image data section. The header also specifies the type of compression mechanism. The length of the File Header is often fixed. The header information must provide all the necessary information to reconstruct the original data and its organisation from the stored data.

## 1.7 Applications of Digital Image Processing:

Digital Image Processing is widely used in different fields like:

1. Medicine
2. Forensics
3. Remote Sensing
4. Communications
5. Automobiles.



# Chapter 2

## Operations on Images

### 2.1 Pixel Connectivity

The relationship between two or more pixels is defined as Pixel Connectivity. Connectivity information is used to establish the boundaries of the 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 grayscale images,  $V$  might be of any range of Gray 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.

In terms of pixel coordinates,

$$(x \pm 1, y) \text{ or } (x, y \pm 1)$$

#### 6-Connectivity:

6-connected pixels are neighbours to every pixel that touches one of their corners in a hexagonal grid or stretched bond rectangular grid. There are several ways to map hexagonal tiles to integer pixel coordinates. With one method, in addition to the 4-connected pixels, the two additional pixel coordinates

$$(x + 1, y + 1) \text{ and } (x - 1, y - 1)$$

are connected to the pixel at  $(x, y)$ .

#### 8-Connectivity:

It is assumed that the pixels  $p$  and  $q$  share a common grayscale value. The pixels  $p$  and  $q$  are said to be in 8-connectivity if  $q$  is in the set  $N_8(p)$ . In simpler terms, in addition to 4-connected pixels, each pixel with coordinates

$$(x \pm 1, y \pm 1)$$

is connected to the pixel at  $(x, y)$ . Eg: Fig: ?? shows 8-connectivity when  $V = \{0, 1\}$ . Here, a multiple path or Loop is present.

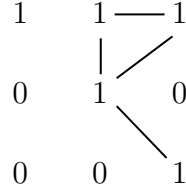


Figure 2.1: 8-Connectivity represented as lines

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

Eg: Fig: ?? shows an  $m$ -connectivity that there is no multiple paths or loops present in the connectivity. It can be observed that multiple paths have been removed.

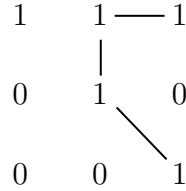


Figure 2.2:  $m$ -connectivity

## 2.2 Distance Measures

The distance between the pixels  $p$  and  $q$  in an image can be given by distance measures such as Euclidean Distance,  $D_4$  distance and  $D_8$  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: ??, the distances between the pixels can be calculated.

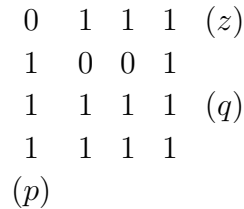


Figure 2.3: Sample Image

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.  $D(p, q) \geq 0$  if  $p = q$ , then  $D(p, q) = 0$ .

3. The distance  $D(p, q) = D(q, p)$ .
4.  $D(p, q) + D(q, z) \geq D(p, z)$ . This is called property of triangular inequality.

### Euclidean Distance:

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 Euclidean Distance is its simplicity. However it is very computationally costly due to involvement of square root operations.

### $D_4$ Distance or City Block Distance:

The  $D_4$  distance or City Block Distance can be simply calculated as:

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

### $D_8$ Distance or Chessboard Distance:

This is calculated with the help of the following:

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

### Question:

Let  $V = \{0, 1\}$ . Compute the  $D_e$ ,  $D_4$ ,  $D_8$  and  $D_m$  distances between two pixels  $p$  and  $q$ . Let the pixel coordinates of  $p$  and  $q$  be  $(3, 0)$  and  $(2, 3)$  respectively, for the image shown below. Find the Distance Measures.

	0	1	2	3
0	0	1	1	1
1	1	0	0	1
2	1	1	1	1 (q)
3	1 (q)	1	1	1

Figure 2.4: Sample Image

### Solution:

The Euclidean Distance is:

$$\begin{aligned}
 D_e &= \sqrt{(x - s)^2 + (y - t)^2} \\
 &= \sqrt{(3 - 2)^2 + (0 - 3)^2} \\
 &= \sqrt{1 + 9} \\
 &= \sqrt{10}
 \end{aligned}$$

$$\begin{aligned}
D_4 &= \sqrt{|x - s| + |y - t|} \\
&= |3 - 2| + |0 - 3| \\
&= 1 + 3 \\
&= 4
\end{aligned}$$

$$\begin{aligned}
D_8 &= \max(|x - s|, |y - t|) \\
&= \max(|3 - 2|, |0 - 3|) \\
&= \max(1, 3)
\end{aligned}$$

The distances can be checked with Fig: ??.

The distance  $D_m$  depends on the values of the set  $V$ . The implications of the set  $V$  is that the path should be constructed only using the elements of  $V$ . So if the value of the set are changed, the path also changes.

The simplest  $D_m$  distance can be calculated along the diagonal path. Here the distance is 3. Suppose the set  $V = \{1\}$ , the path of the distance  $D_m$  also changes.

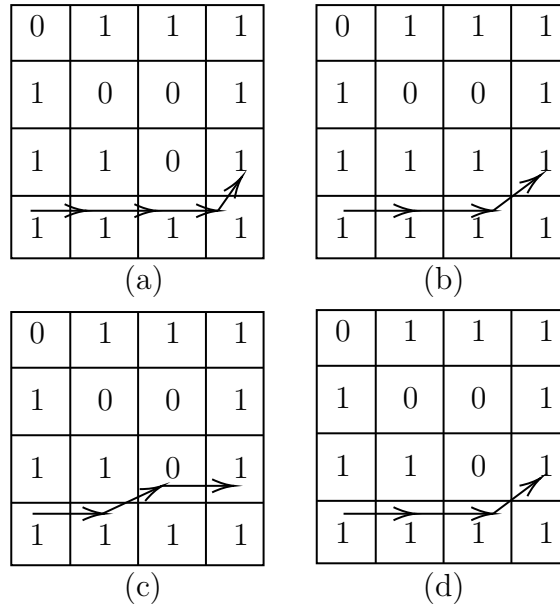


Figure 2.5: Distance Measures: (a) Distance  $D_e$ , (b) Distance  $D_4$  when  $V = \{0, 1\}$ , (c) Distance  $D_8$  when  $V = \{0, 1\}$ , (d) Distance  $D_m$  when  $V = \{1\}$

# Chapter 3

## Image Operations

Image Operations are array operations. These Operations are done on a pixel-by-pixel basis. Array Operations are different from Matrix Operations.

Eg: Consider the two Images:

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

and

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

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

$$F_1 \times F_2 = \begin{pmatrix} AE & BF \\ CG & DH \end{pmatrix}$$

In addition, one can observe that  $F_1 \times F_2 = F_2 \times F_1$ , whereas matrix multiplication is clearly different, since in matrices,  $A \times B \neq B \times A$ . By default, image operations are array operations only.

### 3.1 Arithmetic Operations

Arithmetic operations include Image Addition, Subtraction, Multiplication, Division and Blending.

#### 3.1.1 Image Addition

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

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

The pixels of the input images  $f_1(x, y)$  and  $f_2(x, y)$  are added to obtain the resultant image  $g(x, y)$ . However, it should be noted that the sum should not cross the allowed range for the datatype used.

Similarly it is possible to add any constant value to an image by:

$$g(x, y) = f_1(x, y) + k$$

where,  $k$  is any constant.

If the value of  $k$  is larger than 0, the overall ‘brightness’ of the image is increased.

### **Some Practical Applications of Image Addition:**

#### **Brightness of an 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. Thus adding a constant to all the pixels of an image increases its brightness and vice-versa.

#### **Double Exposure:**

Double Exposure is the technique of superimposing an image on another image to produce the resultant. This gives the scenario equivalent to exposing a film to two pictures or Applying Watermarking on an image. This is done by adding One image on top of the other.

### **3.1.2 Image Subtraction**

The Subtraction of two images can be done as follows:

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

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:

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

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. As discussed earlier, the decrease in the average intensity reduces the brightness of the image.

#### **Some practical applications of Image Subtraction are as follows:**

- **Background Elimination:** This is done by subtracting the background image from the original image.
- **Brightness reduction:** This is done by subtracting a constant value from the whole image.
- **Change Detection:** This is done by subtracting two images and if the resultant value is non-zero, then there is a change between the two images.

### 3.1.3 Image Multiplication

Image Multiplication can be done in the following manner:

$$g(x, y) = f_1(x, y) \times f_2(x, y)$$

Here,  $f_1(x, y)$  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 by that specific data type.

Similarly, scaling by a constant can be performed as:

$$g(x, y) = f(x, y) \times k$$

where,  $k$  is a constant.

If  $k$  is greater than 1, then the overall contrast increases. If  $k$  is less than 1, then the overall contrast decreases.

**NOTE:** The Brightness and Contrast can be manipulated together and also simultaneously by:

$$g(x, y) = a \times f(x, y) + k$$

Here, the parameters  $a$  and  $k$  are used to manipulate the contrast and brightness of the image  $f(x, y)$  respectively,  $g(x, y)$  is the output/resultant image.

**Some of the practical applications of Image Multiplication are as follows:**

- **Manipulate Contrast:** If a fraction of  $\geq 1$  multiplied with the image, the overall contrast increases and vice-versa.
- **Filter Masks:** Useful for designing filter masks.
- **Highlighting AoI:** Used to create a mask to highlight area of interest.

### 3.1.4 Image Division

Similar to other previously discussed operations, the division operation can be performed in the following manner:

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

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

Improper Data type specification may result in loss of information.

Division using a constant can also be performed, like:

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

where,  $k$  is a constant.

**Some of the practical applications of Image Division are as follows:**

- Change Detection.
- Separation of luminance and reflectance components.
- Contrast Reduction.

## 3.2 Geometric Operations

Let's discuss the Geometrical Operations used in Image Processing. These include: Translation, Scaling, Zooming, Linear Interpolation, Mirror or Reflection.

### 3.2.1 Translation Operation

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  $X'$ .

This translation can be represented as:

$$\begin{aligned}x' &= x + \delta x \\y' &= y + \delta y\end{aligned}$$

However, other transformations such as scaling and rotation are multiplicative in nature. The transformation process for **rotation** is given by  $F' = RF$ , where  $\mathbf{R}$  is the transform matrix for performing rotation. Whereas, the transformation process for **scaling** is given by  $F' = SF$ . Here,  $\mathbf{S}$  is the scaling transformation matrix.

To solve such a discrepancy and 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  $w \neq 0$ .

The properties of homogeneous coordinates are as follows:

- In homogeneous coordinates, at least one point should be non-zero. Thus  $(0, 0, 0)$  does not exist in the homogeneous coordinate system.
- 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  $3 \times (1, 3, 5)$ .
- The point  $(x, y, w)$  in the homogeneous coordinate system corresponds to the point  $(\frac{x}{w}, \frac{y}{w})$  in 2D space.

Thus, in the homogeneous coordinate system, the translation process of the point  $(x, y)$  to the new point  $(x', y')$  of the image  $F$  is described as:

$$\begin{aligned}x' &= x + \delta x \\y' &= y + \delta y\end{aligned}$$

In matrix form, this can be stated as:

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

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



### 3.2.2 Scaling Operation

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

$$\begin{aligned} x' &= x \times S_x \\ y' &= y \times S_y \end{aligned} \quad \begin{bmatrix} x' & y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x & y \end{bmatrix}$$

Here,  $S_x$  and  $S_y$  are called *scaling factors* along  $x$  and  $y$  axes, respectively. If the scaling factor is 1, then the object would appear larger. If the scaling factors are fractions, the object would shrink.

### 3.2.3 Zooming Operation

The image can be zoomed using a process called pixel replication or interpolation. Replication is called a zero-order hold process, where each pixel along the scan line is repeated once. The scan line is repeated. The aim is to increase the number of pixels, thereby increasing the dimension of the image.

Eg: For the following image  $F$ :

$$\begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} = \begin{matrix} 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \quad (3.1)$$

This process is called **zero-order hold** process. Once 0s are inserted, the pixels are replicated to yield the following:

$$\begin{matrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 3 & 3 \\ 1 & 1 & 3 & 3 \end{matrix} \quad (3.2)$$

### 3.2.4 Linear Interpolation Operation

Consider the image:

$$H = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$$

Linear interpolation is equivalent to fitting a straight line by taking the average along the rows and columns. The process is described as follows:

1. For eg., the matrix  $H$  can be zero-interlaced as:

$$H = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

2. Interpolate the rows first. This is achieved by taking the average of the columns of that row. This yields:

$$\begin{bmatrix} 2 & 1.5 & 1 & 0.5 \\ 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 1.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3. Interpolate Columns next. This is achieved by taking the average of the rows of that column. This yields:

$$\begin{bmatrix} 2 & 1.5 & 1 & 0.5 \\ 1.5 & 1.75 & 2 & 1 \\ 1 & 2 & 3 & 1.5 \\ 0.5 & 1 & 1.5 & 0.75 \end{bmatrix}$$

### 3.2.5 Mirror or Reflection Operation

This function creates the reflection of the object in a plane mirror. In other words, this 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  $180^\circ$ .

The reflection along  $x$ -axis is given by:

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

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

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

Similarly, the reflection along the  $y = x$  is given by:

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

Similarly, the reflection along the  $y = -x$  is given by:

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

In homogeneous coordinate system, the matrices for reflection can be given as:

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

# Chapter 4

## Image Enhancement

### 4.1 Histogram Manipulation

Histogram Manipulation basically modifies the histogram of an input image so as to improve the visual quality of the image. In order to understand histogram manipulation, it is necessary to understand what a ‘Histogram’ is and how ‘histogram equalisation’ technique affects the visual quality of the image.

#### 4.1.1 Histogram

The Histogram of an image is a plot or visual representation of the number of occurrences of each gray level present in the image. The Histogram provides a convenient summary of the intensities of an image, but it is unable to convey any information regarding spatial relationships between individual pixels. The histogram provides more insight about image *contrast* and *brightness*.

Some properties of a typical histogram are listed as follows:

1. The histogram of a dark image will be clustered towards the lower gray levels.
2. The histogram of a bright image will be clustered towards the higher gray levels.
3. For a low-contrast image, the histogram will not be spread equally, i.e., the histogram will be narrow.
4. For a high-contrast image, the histogram will have an equal spread in the gray levels.

The Image Brightness may be improved by modifying the histogram of the image.

#### 4.1.2 Histogram Equalisation

Equalisation is a process that attempts to spread out the gray levels in an image so that they are evenly distributed across their range. It flattens the histogram to create a better quality image. Histogram equalisation reassigns the brightness values of pixels based on the image histogram. Histogram equalisation is a technique where the histogram of the resultant image is as flat as possible. It also provides a more visually pleasing result across a wider range of images.

### Procedure to Perform Histogram Equalisation:

Histogram equalisation is done by performing the following steps:

1. Create the frequency distribution table for each gray level from the given input image.
2. Find the running sum of the histogram values.
3. Normalise the values from the previous step, by dividing by the total number of pixels.
4. Multiply the values from the previous step, by the maximum gray-level value and round it to nearest integer.
5. Map the gray level values to the result obtained in the previous step, using a one-to-one correspondence.

### Question:

Perform histogram equalisation of the following image.

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

*Solution:* The maximum value in this case is found to be 5. A minimum of 3 bits to represent the number. There are eight possible gray levels from 0 to 7. The histogram of the input image is given below:

Gray Level	0	1	2	3	4	5	6	7
Number of Pixels	0	0	0	6	14	5	0	0

Step 1: Compute the running sum of histogram values.

Gray Level	0	1	2	3	4	5	6	7
Number of Pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25

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

Gray Level	0	1	2	3	4	5	6	7
Number of Pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Running Sum / Total Number of Pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25

Gray Level	0	1	2	3	4	5	6	7
Number of Pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Running Sum	0	0	0	6	20	25	25	25
Multiplying by Maximum Gray Level	$\frac{0}{25} \times 7$	$\frac{0}{25} \times 7$	$\frac{0}{25} \times 7$	$\frac{6}{25} \times 7$	$\frac{20}{25} \times 7$	$\frac{25}{25} \times 7$	$\frac{25}{25} \times 7$	$\frac{25}{25} \times 7$

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

*Rounding off to the closest Integer values.*

Gray Level	0	1	2	3	4	5	6	7
Number of Pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Running Sum	0	0	0	6	20	25	25	25
Multiplying by Maximum Gray Level	0	0	0	2	6	7	7	7

Step 4: Mapping of Gray Levels by one-to-one correspondence.

Original Gray Level	Histogram Equalised Values
0	0
1	0
2	0
3	2
4	6
5	7
6	7
7	7

Here, the Original Image and Histogram Equalised Image are shown side by side:

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

## 4.2 Bit-Plane Slicing

The Gray Level of each pixel in a digital image is stored as one or more bytes in a computer. For an 8-bit image, 0 is encoded as 0 0 0 0 0 0 0 0, and 255 is encoded as 1 1 1 1 1 1 1 1. Any Number between 0 and 255 is encoded as one byte. The bit in the far left side is referred as the Most Significant Bit (MSB), because a change in that bit would significantly change the value encoded by the byte. The bit in the far right is referred as

the Least Significant Bit (LSB), because a change in this bit does not change the encoded gray value much.

Bit plane slicing is a method of representing an image with one or more bits of the byte used for each pixel. One can use only the MSB to represent a pixel, which reduces the original gray level to a binary image. The three main goals of bit plane slicing are:

1. Converting a gray level image to a binary image.
2. Representing an image with fewer bits and compressing the image to a smaller size.
3. Enhancing the image by focusing.

### 4.2.1 Bit-Plane Transformations

It is possible to transform a gray level image into a sequence of binary images by taking advantage of its bit pattern storage. For example, a gray level image is stored as an 8-bit image. Therefore, the image can be transformed into an 8-level image where the zero planes consist of the last bit of each gray level. The first plane consists of the first bit of each gray value. Similarly, the most significant bit plane has the greatest effect in terms of the magnitude of the image. It is roughly equivalent to the threshold of the image at level 127. Depending on the requirement, the bit planes can be retained or ignored.

#### Question:

Show the bit-plane slicing of the following image:

$$\begin{bmatrix} 7 & 6 & 5 \\ 4 & 3 & 2 \\ 1 & 1 & 0 \end{bmatrix}$$

*Solution:* The Binary Equivalent of the pixels is:

$$\begin{bmatrix} 111 & 110 & 101 \\ 100 & 011 & 010 \\ 001 & 001 & 000 \end{bmatrix}$$

*Case-I:* When LSB is changed to 0, this image is reduced to:

$$\begin{bmatrix} 110 & 110 & 100 \\ 100 & 010 & 010 \\ 000 & 000 & 000 \end{bmatrix}$$

The Equivalent Image of the above:

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

*Case-II:* When the MSB is changed to 0, this image changes to:

$$\begin{bmatrix} 011 & 010 & 001 \\ 000 & 011 & 010 \\ 001 & 001 & 000 \end{bmatrix}$$

The Equivalent Image of the above Binary Image:

$$\begin{bmatrix} 3 & 2 & 1 \\ 0 & 3 & 2 \\ 1 & 1 & 0 \end{bmatrix}$$

Thus, Bit-Plane Slicing Leads to Transformation of Images.