UNIT 3

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

Structure **Page No.**

5.1 INTRODUCTION

In this unit, we provide an overview of image enhancement techniques in spatial domain. These techniques improve the quality of images. The enhancement process does not increase the information content in the data. But it increases the dynamic range of the selected features so that they can be detected easily. Many point processing enhancement techniques are suggested in this unit. Image enhancement is a very important topic because of its usefulness in virtually all image processing applications.

Now, we shall list the objectives of this unit. After going through the unit, please read this list again and make sure that you have achieved the objectives.

Objectives

After studying this unit, you should be able to

- define image enhancement

- perform image enhancement in spatial domain
- perform point operations on images
- perform image enhancement using the following algorithms
  - Point operations
  - Contrast stretching
  - Clipping and thresholding
  - Digital Negative
  - Intensity levels slicing
  - Bit plane extraction

Let us begin the unit by discussing image enhancement.

5.2 IMAGE ENHANCEMENT

Image enhancement techniques improve the quality of image as perceived by human observer/ machine vision system. Enhancement techniques improve the perception of information in an image for human viewing and provide 'better' input for other automated image processing techniques. The main **objective** is to modify attributes of an image to make it more suitable for a given task or a specific observer. Image quality can degrade because of poor illumination, improper acquisition device, coarse quantization noise during acquisition process etc. The recorded images after acquisition exhibit problems such as.

• Too dark
• Too light
• Not enough contrast
• Noise

Thus, enhancement aims to improve visual quality by 'Cosmetic processing'. A process of improving the visual quality of any image so that it is more suitable for a particular application is termed as enhancement. The enhancement process does not increase the inherent information content in the data. But, it increases the dynamic range of the chosen features so that they can be detected easily. Generally, humans are the ultimate judge of the improved quality. Quality can also be objectively quantified (measured) by metrics like mean square error (MSE).

Suitability of the enhanced image heavily depends on the application. Enhancement is generally one of the preprocessing methods used on an image so that it is more suitable for further processing. For example, a finger print recognition system used for attendance recording of the employees in an organization, uses image enhancement techniques to get best recognition results under all circumstances. During finger print capturing, the quality of finger print can go down because of dust, sweat, noise, etc.Image enhancement techniques make the input more suitable for further processing so that best results can be achieved. If preprocessing is skipped and finger print matching algorithm is applied directly on the input, the algorithm can show a mismatch for an input which is otherwise a match.

Evaluation of image quality by human observer is a very subjective process and is hard to standardize. An image may be good in one person's opinion, may not be good in another person's opinion. But people's view about the quality of an image is very important and cannot be neglected. Generally, a set of 20 (or larger number) people are asked to give their opinion about the enhanced image and average results are taken.

Evaluation task for machine perception is much easier. In this case, a good image is defined as one which gives best machine recognition results. But certain amount of trial & error is generally required before a particular image enhancement approach is selected.

Image enhancement techniques are application specific and produce a 'better' image.

They are broadly classified into two categories namely

i) Spatial domain methods
ii) Frequency domain methods

In spatial (time) domain methods, pixel values are manipulated directly to get an enhanced image, whereas in frequency domain methods, firstly fourier transform of image is taken to convert image into frequency domain. Then the fourier transform is manipulated and the modified spectrum is transformed back to spatial domain to view the enhanced image. Some enhancement techniques operate on combination of these methods to get best results.

Based on what we have discussed so far, you may try the following exercises.

E1) Specify the objectives of image enhancement techniques.

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

E3) What is the importance of image enhancement in image processing?

In the following section, we shall discuss the point processing.

5.3 POINT OPERATIONS

In this unit, we shall discuss various spatial domain enhancement techniques. The pixel values (grey values) of an image are directly manipulated. Such operations simply take the grey value of each pixel/neighbouring pixels, map it to a new value and move on to the next pixel. Let $f(x, y)$ be an input image. An image processing operation in the spatial domain may be expressed as a mathematical function $T[f(x, y)]$ applied to the image $f(x, y)$ to produce a new image $g(v, v)$. Therefore, $g(x, y) = T[f(x, y)]$.

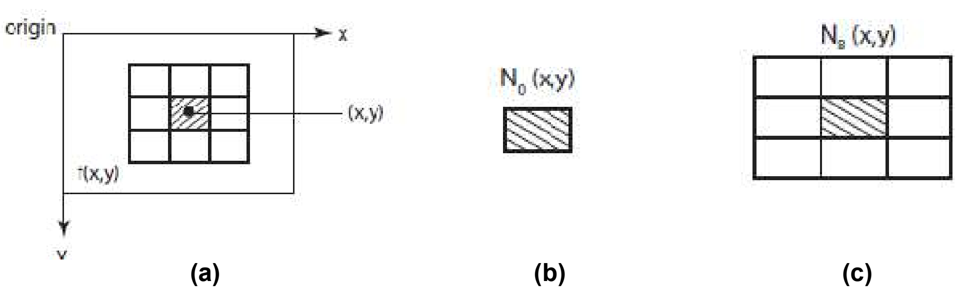


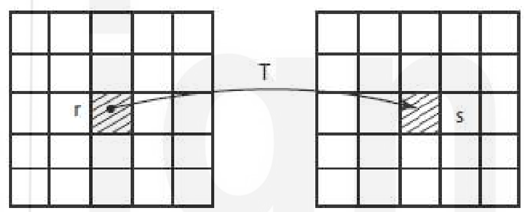
Fig. 1: Point and Neighbourhood Processing

We define the operator T applied on $f(x, y)$ over

i) a single pixel (x, y), which is called ‘ **point processing**’, as shown in Fig. 1 (b).
ii) some neighbourhood of (x, y), which is called ‘**Neighbourhood processing**’, as shown in Fig. 1 (c).
iii) T may operate on a set of input images instead of a single image.

The principal approach defining a neighbourhood about a point (x, y) is to use a square or rectangular sub image are centred at (x, y) as shown in Fig. 1. The centre of the sub image is moved from pixel to pixel starting at the top left corner. The operator T is applied to each location (x, y) to yield the output $g(u, v)$ at that location.

Point processing is the simplest case of spatial domain techniques where output at (x, y) only depends on the input intensity at the same point as shown in Fig. 2. It is a memoryless operation. In this case, pixels of same intensity get the same



transformation.

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

Fig. 2: Point operations

Let us see how these transformations take place.

Example 1: Perform the transformation $g_1(v) = v + 1$ on the image $f(x, y) = \begin{bmatrix} -2 & -1 & 0 \\ 0 & 1 & 2 \end{bmatrix}$.

Solution: $G_1(x, y) = g_1(f(x, y)) = \begin{bmatrix} -2+1 & -1+1 & 0+1 \\ 0+1 & 1+1 & 2+1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}$.

***

Example 2: Perform the transformation $g_2(v) = v^2$ on the image $f(x, y) = \begin{bmatrix} -2 & -1 & 0 \\ 0 & 1 & 2 \end{bmatrix}$.

Solution: $G_1(x, y) = g_2(f(x, y)) = \begin{bmatrix} (-2)^2 & (-1)^2 & 0^2 \\ 0^2 & 1^2 & 0^2 \end{bmatrix} = \begin{bmatrix} 4 & 1 & 0 \\ 0 & 1 & 4 \end{bmatrix}$.

***

Point operations are simplest yet extremely useful and powerful image processing tasks. Point operations are defined as

$g(u, v) = T[f(x, y)]$, also, $s = T(r)$
Where $r =$ grey level of input image $f(x, y)$, and
$s =$ grey level of output image $g(x, y)$

Thus, grey levels from the input image are modified with a function T which is independent of image coordinates. Typical examples of point operations are modifying image brightness, histogram processing etc.

Now, we shall discuss neighbourhood processing.

In this case, the transformation operator T is defined over the neighbourhood of (x, y). This neighbourhood can be defined using a square/ rectangular/ circular sub image that are centred at (x, y). In this method, the spatial characteristics around the pixel (x, y) can be used which is not possible in point operation.

In the Fig. 3, the operation on nine pixels around (x, y) in the input image results in manipulating the grey level values of (x, y) in the output image. Some examples of neighbourhood processing are spatial filtering, Laplatian operator etc.

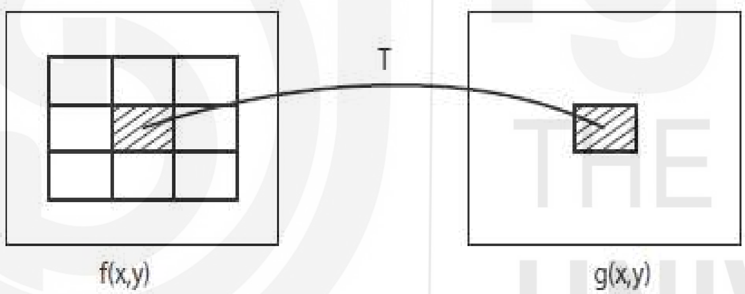


Fig. 3: Neighbourhood Processing

Point Processing

Point operations are zero memory operations where a given gray level values of an individual pixel in the input image $r \in [1, L-1]$ is mapped into a gray levels $s \in [1, L-1]$ of the pixels in the output image using the transformation $T()$.

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

Try the following exercises.

E4) What do you mean by point processing?

E5) Perform the transformation $g_1(v) = 3v$ on the image

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

In the following section, we shall discuss contrast stretching.

5.4 CONTRAST STRETCHING

Contrast ‘c’ between two levels x_1 and x_2 are defined as the absolute values of the difference, between x_1 and $x_2 \Rightarrow c(x_1, x_2) = |x_1 - x_2|$. Good contrast in an image is important to distinguish the details else they will merge in background. Low contrast images occur due to bad or non-uniform illumination conditions or due to nonlinearity of image acquisition devices. Fig. 4 shows an example of low contrast image where the details are lost in background.

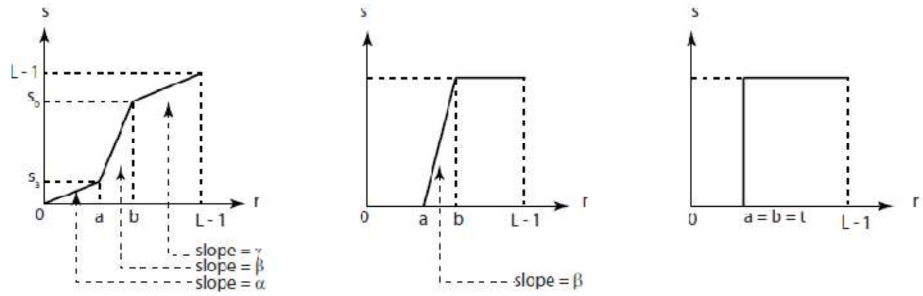
Fig. 4: A low contrast image

Value remapping by contrast stretching is a process that expands the range of levels so that all the levels contribute to the image. The main idea is to reduce the low level gray values, and to increase the mid range gray values, so that an artificial contrast is created between the two sets of gray values. The contrast between gray values is thus stretched from both sides.

The transformation is defined mathematically as follows:

$$s = \begin{cases} \alpha r & 0 \le r \le a \\ \beta(r - a) + s_a & a \le r \le b \\ \gamma(r - b) + s_b & b \le r \le L \end{cases}$$

Where α, β, γ are slopes of different regions as shown in Fig. 5 (a). The slopes determine the amount of contrast stretching (or diminishing). If the slope is greater than one, then corresponding grey levels are stretched because the original grey levels are mapped onto a larger range of grey values. A slope smaller than one means the contrast is diminished. A slope of one indicates no contrast alteration. The parameters a and b are user defined. Fig. 6 shows an example of image enhancement using contrast stretching.



(a) Contrast stretching **(b) Clipping** **(c) Thresholding**

Fig. 5

(a) low contrast image **(b) enhanced image**

Fig. 6: Effect of Contrast Stretching

Try the following exercise.

E6) What is contrast stretching?

E7) Differentiate between low contrast image and enhanced image.

In the following section, we shall discuss clipping and thresholding.

5.5 CLIPPING AND THRESHOLDING

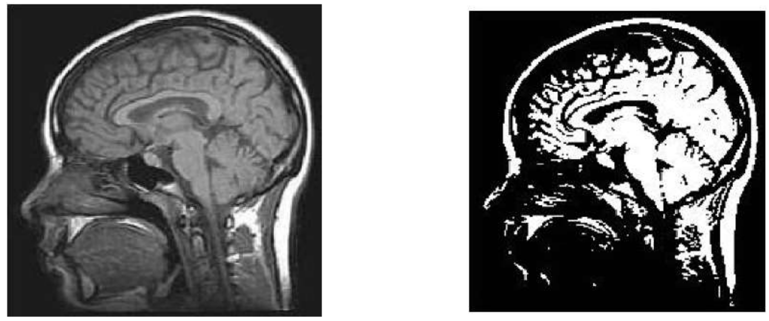
Clipping and thresholding are two special cases of contrast stretching.

a) Clipping

If $\alpha = \gamma = 0$, this is called **'clipping'** (windowing) as shown in, Fig. 5 (b). This operation stretches the contrast to its maximum in a limited range ('window') of original grey level values $\{a,......,b\}$. All the grey levels outside this window are either mapped to zero or the maximum values. Thus, clipping allows us to focus all the available contrast onto the required range of grey level values. This is especially useful when viewing medical images such as CT images. It also helps in viewing under or over exposed images.

b) Thresholding

If $\alpha = \gamma = 0$ and $a = b = t$, this remapping is called **'thresholding'** as shown in Fig. 5 (c). This operation 'binarizes' (only two values present) the image. A suitable threshold value 't' is chosen, all the gray levels smaller than 't' are mapped to zero whereas all the grey levels greater than or equal to 't' are mapped to maximum value. This is a very useful operation in image processing applications. Binarization is generally done before segmentation or region extraction. Fig. 7 shows an example of thresholding with $t = 110$. Note that input image is a gray scale image and output image is binary.



(a) original image **(b) image after thresholding**

Fig. 7: Effect of Thresholding

Example 3: Perform thresholding on image segment $f(x, y)$ with $t = 128$.

Solution:

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

$\xRightarrow{t=128}$

0	0	0	0
0	0	255	255
0	255	255	255
255	255	255	255

***

Try the following exercises.

E8) What are the two special cases of contrast stretching?

E9) Why thresholding operation results in binary output?

In the following section, we shall discuss digital negatives.

5.6 DIGITAL NEGATIVES

The digital negative of an image with the intensity levels in the range $[0, L-1]$ is obtained by using negative transformation shown in Fig. 8, which is given by the expression

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

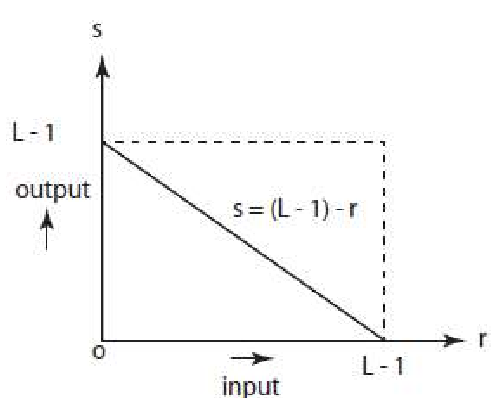


Fig. 8

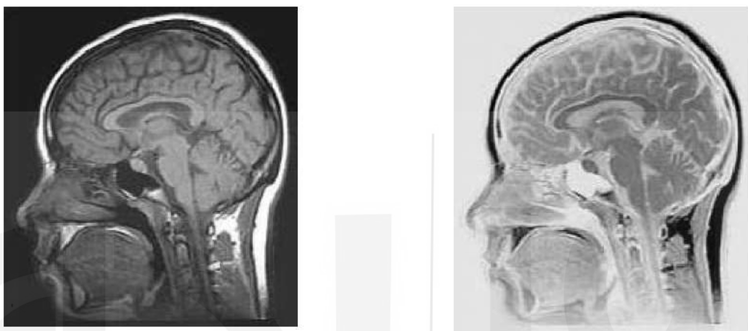
In this transformation, highest grey level is mapped to lowest and vice versa. For an8-bit image, the transformation is

$s = 255 - r$

If $r = 20$, (dark pixel) $s = 255 - 20 = 235$ (bright pixel)

If $r = 125, s = 255 - 125 = 130$

Note that middle grey level has not changed much whereas dark grey level has become bright. This transformation is generally used to enhance the white details embedded in the dark regions of an image where black is dominant. It is useful in displaying medical images.Fig. shows an example. Even though the visual contents of both images are same, it is much easier to analyse image in negative form.



(a) original image **(b) digital negative of fig (a)**

Fig. 9: Effect of digital negative

Example 4: Find the image negative transformation on an image f given as

$f(x, y) =$

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

Solution: The negative of the image is given as

$g(x, y) =$

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

***

Try the following exercise.

E10) Find the negative transformation of the image $f(x, y) = \begin{bmatrix} 1 & 2 & 10 \\ 3 & 4 & 0 \\ 1 & 5 & 6 \end{bmatrix}$.

In the following section, we shall discuss about intensity level slicing.

5.7 INTENSITY LEVEL SLICING

A variant of thresholding is intensity level slicing or double thresholding. It is used to highlight a range of intensity values of interest in an image. In this operation, all the grey levels in a window $\{a, \ldots, b\}$ are set to maximum grey level and all the grey levels outside this range are set to zero as shown in Fig. 10.

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

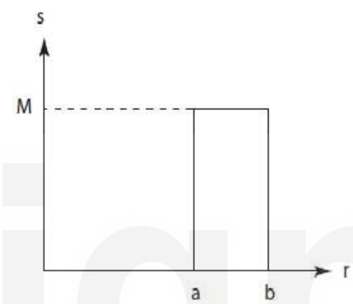


Fig. 10: Transfer Function of Intensity Level Slicing

Intensity level slicing is a binarization operation as the resulting image has only two grey level values (0 and M). Mostly this technique is used to remove unwanted elements (clutter) from an image so that the useful information becomes prominent.

Applications are enhancing certain features which are based on grey level values, such as mass of water in satellite images, cancer cells in a monogram images etc.

Sometimes, we wish to retain the original image as 'background' and highlight the grey levels in a window, following transformation can be used

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

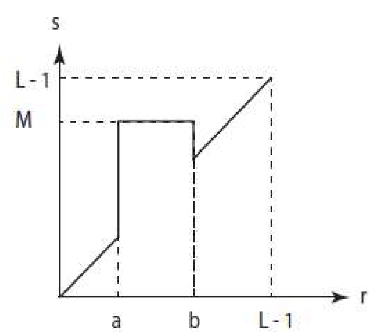
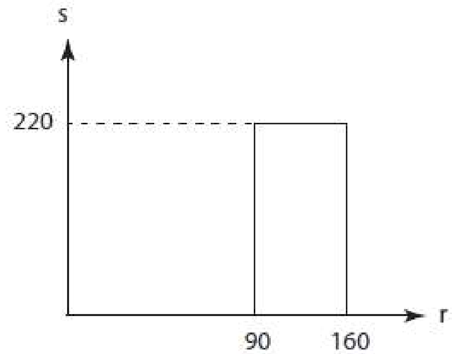


Fig. 11: Transfer Function of Intensity Level Slicing

Example 5: Perform intensity level slicing on $f(x, y)$ as given in Example 4, based on transfer function shown below



Solution: In this case, grey levels less than 90 and greater than 160 are rounded to zero and grey levels between this range are set to 220. This transformation produces a '**binary**' image. In this process, all the background is lost.

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

$f(x, y)$

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

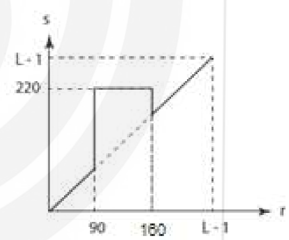
0	0	0	220
0	220	220	0
220	220	0	0
0	0	0	220

$g(x, y)$

***

Try the following exercise.

E11) Perform intensity level slicing on $f(x, y)$ as given in Example 4, based on transformation shown below:



In the following section we shall be discussing another sliding called it extraction.

5.8 BIT EXTRACTION (BIT PLANE SLICING)

Generally, a grey scale image consists of 8 bits for pixel representation. Thus, for overall appearance of the image, there is contribution by each of the 8 bits. For a pixel P at coordinates (x, y), contribution from each bit plane is

$$f(x, y) = k_1 2^7 + k_2 2^6 + k_3 2^5 + k_4 2^4 + k_3 2^3 + k_6 2^2 + k_8 2^0$$

where $k_1, \ldots, k_8$ are either 0 or 1. Images can be assumed to be composed of 8 one bit planes with plane 1 containing the lowest order bits of all pixels in the image and plane 8 containing all highest order bits (fig 12). Fig 13 (a) to (h) show various bit planes of the image in fig 13 (i). Observe that four highest order bit planes have most of the visually significant data. The lower order planes contribute to more subtle intensity level details in the image. For example, if pixel value is 194 (11000010 in binary form) then values of $k_1, k_2, ..., k_8$ are 1, 1, 0, 0, 0, 0, 1, 0.

As the contribution of higher order planes are much more in the image, reconstruction by only these planes results to an image very close to the original image to a great extent. To reconstruct the image using only 8th and 7th plane only is done by multiplying bit plane 8 by 128, bit plane 7 by 64 and then adding the two planes.

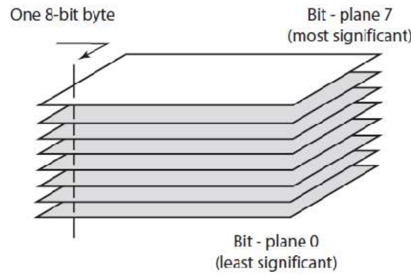
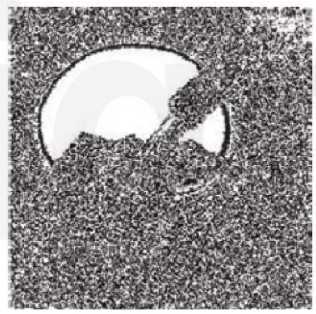
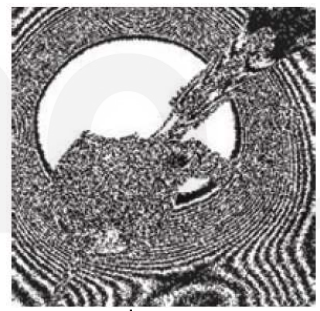
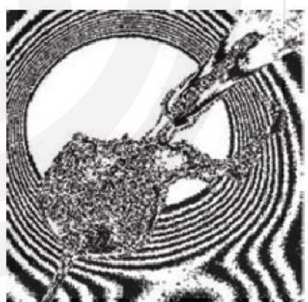


Fig. 12: Bit Plane Slicing

(a) LSB plane **(b) 2nd plane** **(c) 3rdpalne**



(d) 4th plane **(e) 5th plane** **(f) 6th plane**



(g) 7th plane **(h) msb plane** **(i) original image**

Fig. 13: Bit Plane Slicing Output

Example 6: Find MSB and LSB planes for the given image

255	138	30
65	12	201
183	111	85

Solution: Represent all grey level values in binary format.

	LSB				MSB			
255 $\rightarrow$	1	1	1	1	1	1	1	1
138 $\rightarrow$	0	1	0	1	0	0	0	1
30 $\rightarrow$	1	1	1	1	1	0	0	0
65 $\rightarrow$	1	0	0	0	0	0	1	0
12 $\rightarrow$	0	0	1	1	0	0	0	0
201 $\rightarrow$	1	0	0	1	0	0	1	1
183 $\rightarrow$	1	1	1	0	1	1	0	1
111 $\rightarrow$	1	1	1	1	0	1	1	0
85 $\rightarrow$	1	0	1	0	1	0	1	0

MSB plane consists of MSB's of all grey levels as shown in Fig. 14(a) similarly LSB plane consists of LSB's of all grey level values (Fig. 14(b)).

1	1	0
0	0	1
1	0	0

MSB plane
(a)

1	0	1
1	0	1
1	1	1

LSB plane
(b)

Fig. 14

***

Example 7: Compute various bit planes of the following 8-bit image.

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

Solution: To decompose grey level value 100 into 8-bit plane, convert 100 into binary $\rightarrow 01100100$.

Then bit for 8th plane is 0,7th plane is 1, 6th plane is 1 and so on. Similarly, binary representation of 50 is 00110010. Then bit for 8th plane 0, 7th plane is 0, 6th plane is 1 and so on.

Thus, to decompose image grey levels into various bit planes, two steps are followed:

a) Convert the grey level value into binary.
b) Allocate bits into various planes starting from MSB.

Various bit planes for the given image are:

8^{th} plane (MSB)

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

7^{th} plane

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

6^{th} plane

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

5^{th} plane

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

4^{th} plane

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

3^{rd} plane

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

2^{nd} plane

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

1^{st} plane

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

***

Try the following exercises.

E12) From all bit plane of the Example 4, generate the original image.

E13) Generate the image using only 8th, 7th plane and 6th plane.

E14) What is meant by bit plane slicing?

Now, we shall summarise the unit.

5.9 SUMMARY

In this unit, we have discussed the following points.

1. Stated Image enhancement in spatial domain.
2. Explained point processing and neighbourhood processing
3. Explained various point operations such as Contrast stretching, Clipping and thresholding, Digital Negative, Intensity levels slicing, Bit plane extraction.
4. Used special cases of Contrast stretching: Clipping and thresholding, which are very import preprocessing steps in image processing algorithms.

5.10 SOLUTIONS/ANSWERS

E1) Enhancement techniques improve the perception of information in an image for human viewing and provide 'better' input for other automated image processing techniques. The main **objective** is to modify attributes of an image to make it more suitable for a given task or a specific observer.

E2) Image enhancement can be broadly classified into two categories namely
   i. Spatial domain methods
   ii. Frequency domain methods

E3) Image enhancement techniques improve the quality of image as received by human observer/ machine vision system. The enhancement process does not increase the information content in the data. But it increases the dynamic range of the selected features so that they can be detected easily. Many point processing enhancement techniques are suggested in this unit. Image enhancement is a very important topic because of its usefulness in virtually all image processing applications.

E4) Point operations are zero memory operations where a given gray level values of an individual pixel in the input image $r \in [1, L-1]$ is mapped into a gray levels $s \in [1, L-1]$ of the pixels in the output image using $s = T(r)$.

E5) $$G_1(x, y) = g_2(f(x, y)) = \begin{bmatrix} -6 & -3 & 0 \\ 0 & 3 & 6 \end{bmatrix}$$

E6) Is the transformation for contrast stretching. It is used to improve the contrast of low contrast images.

$$s = \begin{cases} \alpha r & 0 \le r \le a \\ \beta(r-a) + s_a & a \le r \le b \\ \gamma(r-b) + s_b & b \ge r \le 1 \end{cases}$$

E7) You may like to differentiate low contrast image and enhance image yourself.

E8) Clipping and thresholding are two special cases of contrast stretching.

E9) Thresholding results in binary output because the output can be either 0 or 255. Multiple input gray level values are mapped to only two output vales.

E10) The negative transformation is $\begin{bmatrix} 254 & 253 & 245 \\ 252 & 251 & 255 \\ 254 & 250 & 249 \end{bmatrix}$.

E11)

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

$f(x, y)$

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

0	10	50	220
50	220	220	200
220	220	190	210
175	210	255	220

$g(x, y)$

In this case, grey levels in the window $\{90,160\}$ are set to 220 (the circled pixels in $g(x, y)$), but other grey levels are not changed. **This transformation does not produce a binary image**.

E12) To generate original image back from the bit planes we need to multiply 128 to 8th bit, 64 to 7th bit, 32 to 6th bit, 16 to 5th bit, 8 to 4th bit, 4 to 3rd bit, 2 to 2nd bit and 1 to 1st bit for each pixel and add all of it.

Thus to get 100 = 128 × 0 + 64 × 1 + 32 × 1 + 16 × 0 + 8 × 0 + 4 × 1 + 2× 0 + 1 × 0.

Similarly 50 = 128 × 0 + 64 × 1 + 32 × 1 + 16 × 0 + 8 × 0 + 4 × 0 +2 × 0 + 1 × 0 and so on the result is original image is exactly reconstructed back as shown below.

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

E13) As we have to consider only 8th 7th and 6th plane, only these plane values are added, remaining values are not considered.

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

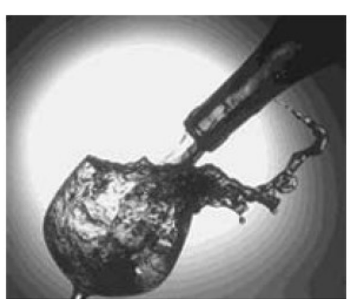
And so on. The resulting image is Fig. 16 (a). There is very small error between the reconstructed image $g(x, y)$ and original image $f(x, y)$ (Fig. 13 (i)). To find the error, we can find $e(x, y) = f(x, y) - g(x, y)$ which is shown in Fig. 16(b). Thus, with only 8th, 7th and 6th plane, the image is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

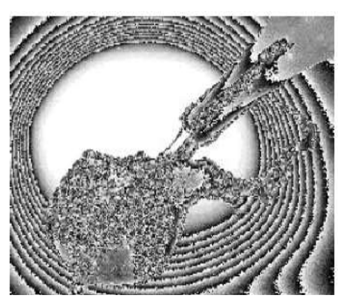
0	0	32	96
32	64	128	196
96	128	160	192
128	192	224	96

$g'(x, y)$

0	10	18	4
18	31	22	8
14	22	30	18
47	18	31	14

$e(x, y)$





(a) image reconstructed with 7th (b) error image 6th and 5th plane

Fig. 16

E14) A grey scale image consists of 8 bits for pixel representation. Thus, for overall appearance of the image, there is contribution by each bit of each pixel. For a pixel P at coordinates (x, y), contribution from each bit plane is

$$f(x, y) = k_1 2^7 + k_2 2^6 + k_3 2^5 + k_4 2^4 + k_3 2^3 + k_6 2^2 + k_8 2^0$$