Indira Gandhi National Open
University School of Sciences

MMTE-003
Pattern Recognition and Image Processing

Block

2

IMAGE IMPROVEMENT I

Curriculum Design Committee

Dr. B.D. Acharya
Dept. of Science & Technology, New Delhi

Prof. Adimurthi
School of Mathematics
TIFR, Bangalore

Prof. Archana Aggarwal
CESP, School of Social Sciences
JNU, New Delhi

Prof. R.B. Bapat
Indian Statistical Institute
New Delhi

Prof. M.C. Bhandari
Dept. of Mathematics
IIT, Kanpur

Prof. R. Bhatia
Indian Statistical Institute, New Delhi

Prof. A.D. Dharmadhikari
Dept. of Statistics
University of Pune

Prof. O.P. Gupta
Dept. of Financial Studies
University of Delhi

Prof. S.D. Joshi
Dept. of Electrical Engineering
IIT, Delhi

Dr. R.K. Khanna
Scientific Analysis Group, DRDO, Delhi

Prof. Susheel Kumar
Dept. of Management Studies
IIT, Delhi

Prof. Veni Madhavan
Scientific Analysis Group
DRDO, Delhi

Prof. J.C. Misra
Dept. of Mathematics
IIT, Kharagpur

Prof. C. Musili
Dept. of Mathematics and Statistics
University of Hyderabad

Prof. Sankar Pal
ISI, Kolkata

Prof. A.P. Singh
PG Dept. of Mathematics
University of Jammu

Faculty Members
School of Sciences, IGNOU
Prof. Poornima Mital
Dr. Atul Razdan
Prof. Parvin Sinclair
Prof. Sujatha Varma
Dr. S. Venkataraman

Course Design Committee

Dr. A.K Sinha
DRDO, Delhi

Prof. Vipula Singh
RNSIT, Bangalore

Prof. Muthukumar Subramanyaom
NIT, Puducherry

Faculty Members
School of Sciences, IGNOU
Prof. Sujatha Varma
Dr. S. Venkataraman
Dr. Deepika

Block Preparation Team

Prof. K.K. Biswas (Retd.) (Editor)
IIT, Delhi

Prof. Vipula Singh
RNSIT, Bangalore

Dr. Deepika
School of Sciences, IGNOU

Course Coordinator: Dr. Deepika

Acknowledgements: Mr. Santosh Kumar Pal for preparing the CRC.

Disclaimer: Any materials adapted from web-based resources in this course are being used for educational purposes only and not for commercial purpose.

June, 2021

ISBN-

Further information on the Indira Gandhi National Open University courses may be obtained from the University's Office at Maidan Garhi, New Delhi-110 068.

Printed and published on behalf of the Indira Gandhi National Open University, New Delhi by the Director, School of Sciences.

BLOCK 2 IMAGE IMPROVEMENT I

In the last block of this course, we defined images & their acquisition process. We also showed digitization process (sampling & quantization) of images & the ways to represent a digital image. We also discussed image characteristics, resolution and relation between the pixels. Various image transforms were also defined and their applications and properties were discussed. Different color models were introduced and image manipulation using pseudo-colour image processing was discussed.

Very often the acquired images are not of very good quality. Sometimes overall image is dark or very bright and lot of interesting details are not very clear. Often the image is noisy because of problems in data acquisition. In this Block we shall see how the quality of images can be enhanced in spatial domain. The spatial domain simply means the image obtained directly from the camera. Later we shall employ enhancement techniques in the frequency domain of the images.

In Unit 5, various point operations for image enhancement in spatial domain will be taken up. First point operations are explained and the procedure to implement point operations is elaborated. Image enhancement algorithms such as: Contrast stretching, Clipping and thresholding, Digital Negative, Intensity levels slicing, Bit plane extraction are explained.

In Unit 6, another point operation namely histogram modeling is introduced. We introduce the concept of histogram and how histogram is used to check the quality of an image is explain in depth. For image enhancement two histogram related techniques are introduced: Histogram equalization and Histogram specification. For both the techniques, we introduce the basic theory and their role in image enhancement.

In Unit 7, we introduce neighbourhood operations for image enhancement in spatial domain, where number of pixels in neighborhood influence the final gray value at a pixel. Spatial linear and non-linear filters are explained. Spatial averaging filter, Spatial low pass filters, Spatial high pass filters are explained with the basic theory and their role in image enhancement. Non-linear filters like Median filters, Min, Max filters are explained with their objective and significance.

UNIT 5

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.

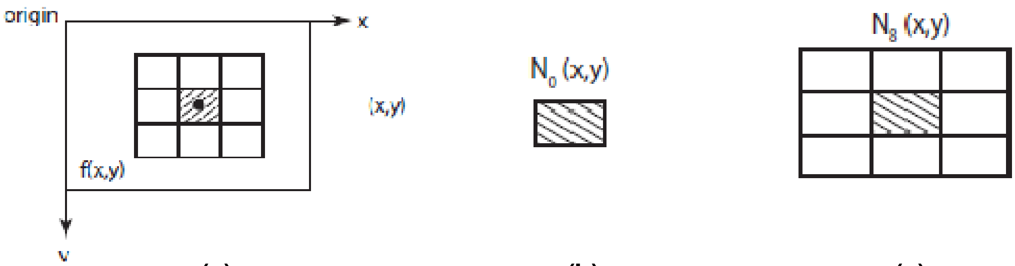
E2) What are the two types of image enhancements? Define them with the help of suitable examples.

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



(a) (b) (c)

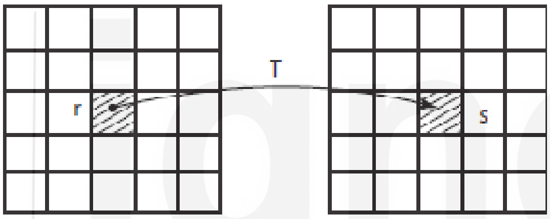
Fig. 1: Point and Neighbourhood Processing

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.

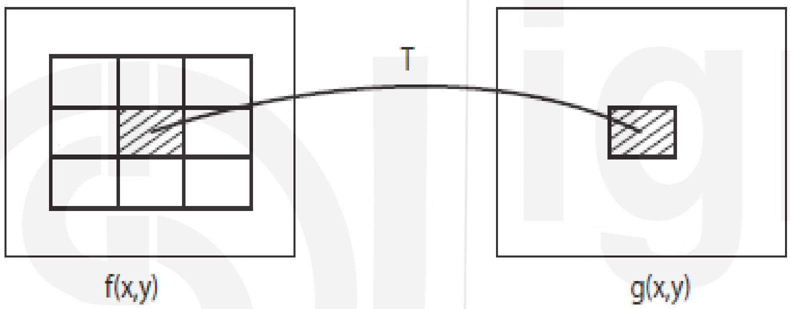


Fig. 3: Neighbourhood Processing

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.

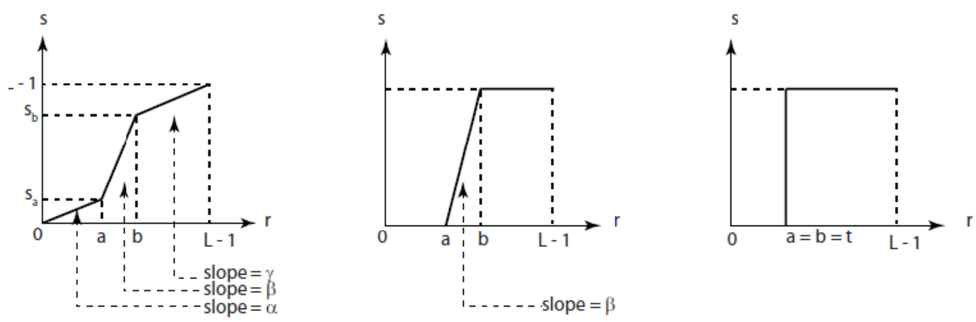
Fig. 4: A low contrast image

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

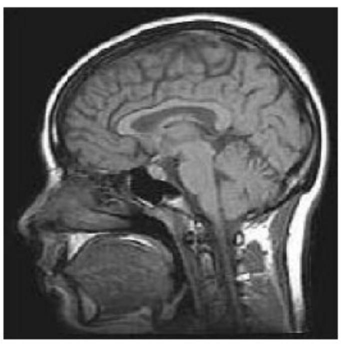
Clipping and thresholding are two special cases of contrast stretching.

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:

$$\begin{array}{|c|c|c|c|}\hline 0 & 10 & 50 & 100 \\ \hline 5 & 95 & 150 & 200 \\ \hline 110 & 150 & 190 & 210 \\ \hline 175 & 210 & 255 & 100 \\ \hline \end{array} \xRightarrow{t-128} \begin{array}{|c|c|c|c|}\hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 255 & 255 \\ \hline 0 & 255 & 255 & 255 \\ \hline 255 & 255 & 255 & 255 \\ \hline \end{array}$$

***

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

$s = (L-1) - r$

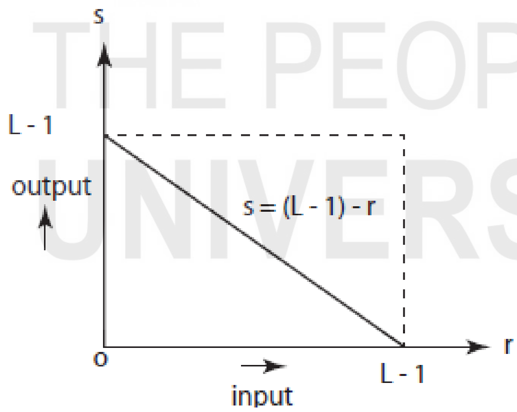
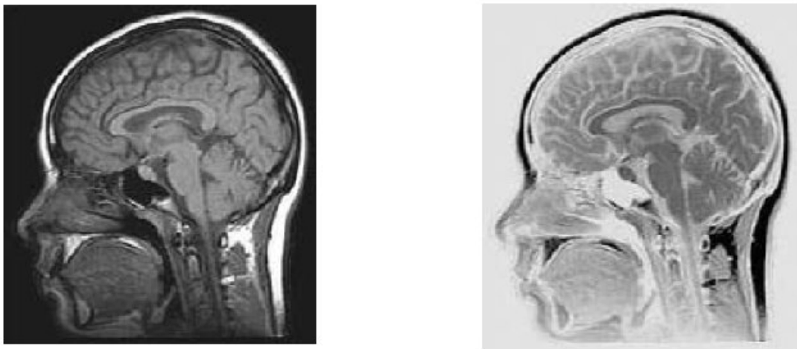


Fig. 8

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 $\{a, \ldots\ldots, b\}$ are set to maximum grey level and all the grey levels outside this range are set to zero as shown in Fig. 10.

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

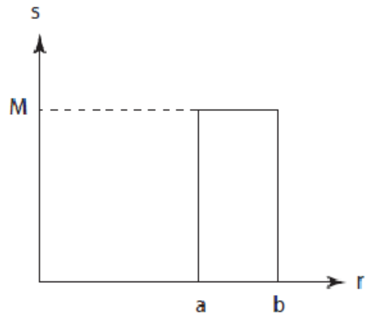


Fig. 10: Transfer Function of Intensity Level Slicing

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

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

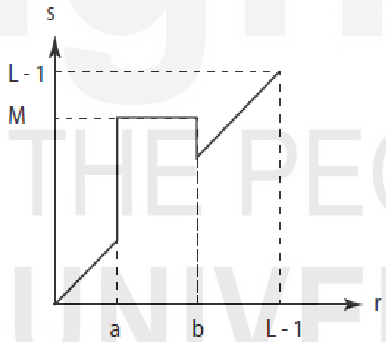
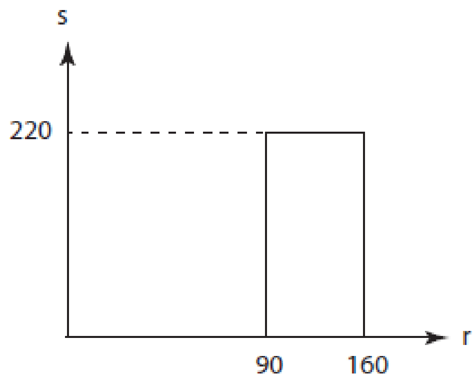


Fig. 11: Transfer Function of Intensity Level Slicing

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

$f(x, y)$

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

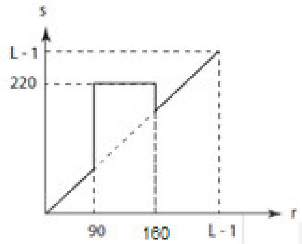
0	0	0	220
0	220	220	0
220	220	0	0
0	0	0	220

$g(x, y)$

***

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 $k_1,....,k_8$ 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.

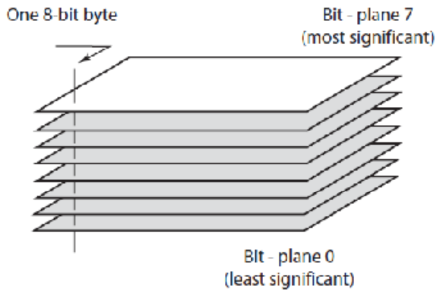


Fig. 12: Bit Plane Slicing

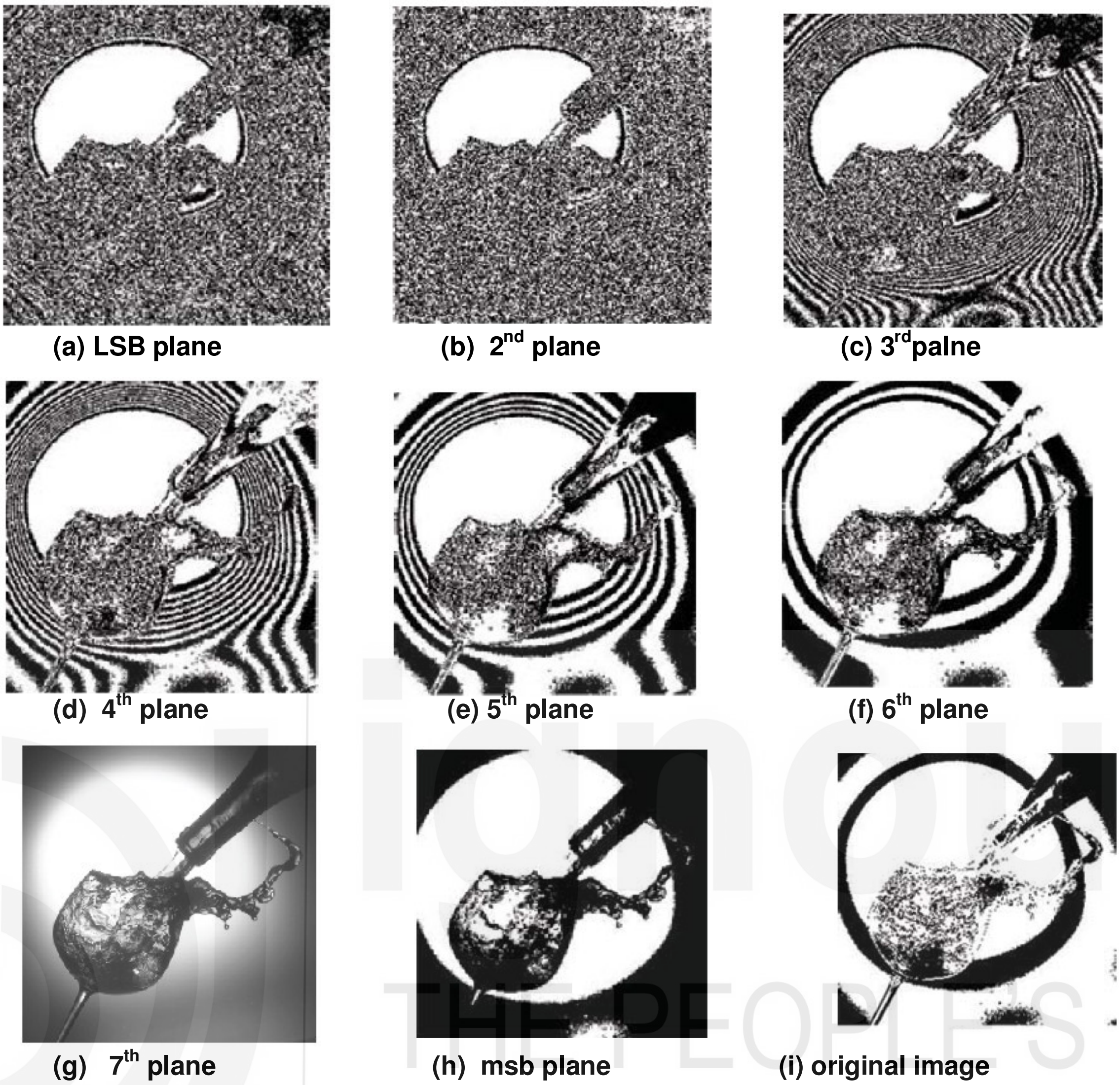


Fig. 13: Bit Plane Slicing Output

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 →	1	1	1	1	1	1	1	1
138 →	0	1	0	1	0	0	0	1
30 →	1	1	1	1	1	0	0	0
65 →	1	0	0	0	0	0	1	0
12 →	0	0	1	1	0	0	0	0
201 →	1	0	0	1	0	0	1	1
183 →	1	1	1	0	1	1	0	1
111 →	1	1	1	1	0	1	1	0
85 →	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.

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

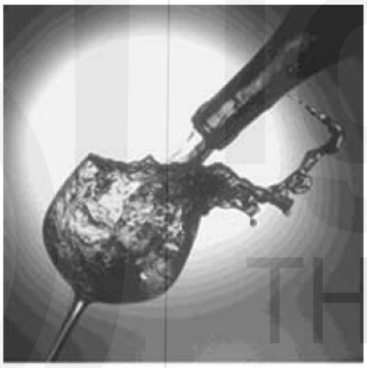
And so on. The resulting image is Fig. 16 (a). There is very small error between the reconstructed image $g(x, y)$ and original image $f(x, y)$ (Fig. 13 (i)). To find the error, we can find $e(x, y) = f(x, y) - g(x, y)$ which is shown in Fig. 16(b). Thus, with only 8th, 7th and 6th plane, the image is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

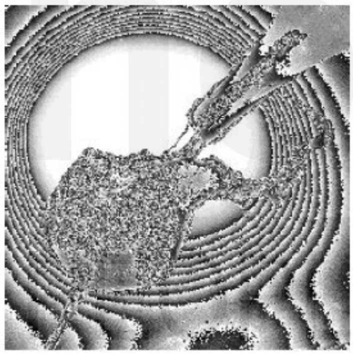
0	0	32	96
32	64	128	196
96	128	160	192
128	192	224	96

$g'(x, y)$

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

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

UNIT 6

HISTOGRAM MODELING

Structure **Page No.**

6.1 INTRODUCTION

In this unit, we introduce the concept of histogram and provide an overview of image enhancement using histogram equalization and histogram specification. The histogram of an image represents the relative frequency of occurrence of the various gray levels in the image. It provides useful image statistics that help us in analyzing the image. Histograms are the basis for many spatial domain processing techniques. Histogram-modeling techniques modify an image so that its histogram has a desired shape. Histogram manipulation is used for image enhancement.

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 histogram;
- generate histogram form given image;
- perform histogram equalization;
- perform histogram specification.

6.2 HISTOGRAM PROCESSING

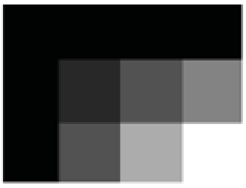
Before we begin with histogram processing, let us define what is the histogram of an image?

Histogram

Histogram of an image represents the number of times a particular grey level has occurred in an image. It is a graph in which the x-axis represents gray levels, and y -axis represents number of pixels for each grey level.

Histogram of an image is defined as $h(r_k) = n_k$, where $r_k = k^{th}$ grey level n_k = number of pixels with grey level r_k, and k takes on the values $0,1,2,...,L-1$.

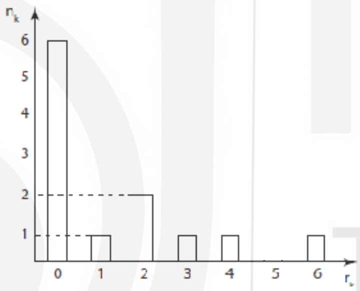
Example 1: Find histogram of image in Fig. 1.



0	0	0	0
0	1	2	3
0	2	4	6

Fig. 1: Image

Solution: A table is generated with r_k and n_k, where r_k contains all possible grey level values 0,1, 2, 3, 4, 5, 6 and n_k consists the number of times that grey value has occurred in the Fig. 1. 0 has occurred 6 times, 1 has occurred 1 time etc. Then histogram is drawn in Fig. 2.



r_k	0	1	2	3	4	5	6
n_k	6	1	2	1	1	0	1

Fig. 2: Histogram

***

Normalized histogram is obtained by dividing the frequency of occurrence of each gray level r_k by the total number of pixels in an image.

$p(r_k) = \frac{n_k}{n}$, where $k = 0........L-1$, where n = total number of pixels in the image $p(r_k)$ gives the probability of occurrence of grey level r_k. The sum of all components of a normalized histogram is equal to 1.

Example 2: Find normalized histogram of the image

0	0	0	0
0	1	2	3
0	2	4	6

Solution: Fig. 2 image is similar to Example 1. A table is made with r_k, n_k and $p(r_k)$ and normalized histogram is drawn in Fig. 3.

r_k	1	1	2	3	4	5	6
n_k	6	1	2	1	1	0	1
$P(r_k)$	$\frac{6}{12}$	$\frac{1}{12}$	$\frac{2}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	0	$\frac{1}{12}$

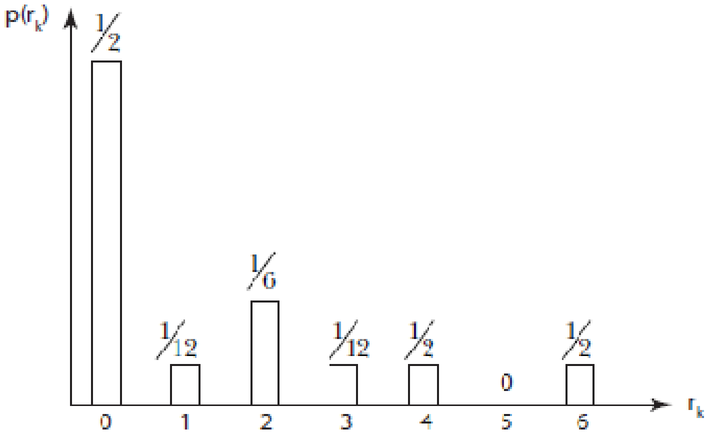


Fig. 3: Normalized Histogram

Now, you might be thinking that why histogram is needed. We shall answer this now.

Histogram gives an insight about the contrast in an image. It tells us the difference between average grey level of an object and that of the surroundings. It also provides a useful image statistics which are helpful in various image processing applications, for example, thresholding, intensity level slicing, segmentation etc. Besides it is very simple to calculate histogram of an image. Intuitively, it tells how vivid or washed out an image appears.

Based on histogram, we can categorize images in the four categories, which are defined below.

i) **Under exposed image:** Fig. 4 (a) shows an under exposed image, and its histogram is shown in Fig. 4 (b). It may be noted that almost all the pixels are concentrated at the lower end of the histogram. The maximum value of grey is less than 50. No details are seen in the image.

ii) **Over-exposed image:** Fig. 4 (c) shows an over-exposed image and its histogram is shown in Fig. 4 (d). The histogram is prominent towards the higher side of the grey values with the lowest grey level value around 150. This image has a washed-out look because of the absence of darker shades of grey.

iii) **Low contrast image:** Fig. 4 (e) shows a low contrast image and its histogram in Fig. 4 (f). The shape of the histogram is narrow and centered towards the middle of the grey scale. Very few grey levels participate in the image formation as neither the lower grey levels nor the higher grey levels are present. Thus, details are not visible in the image.

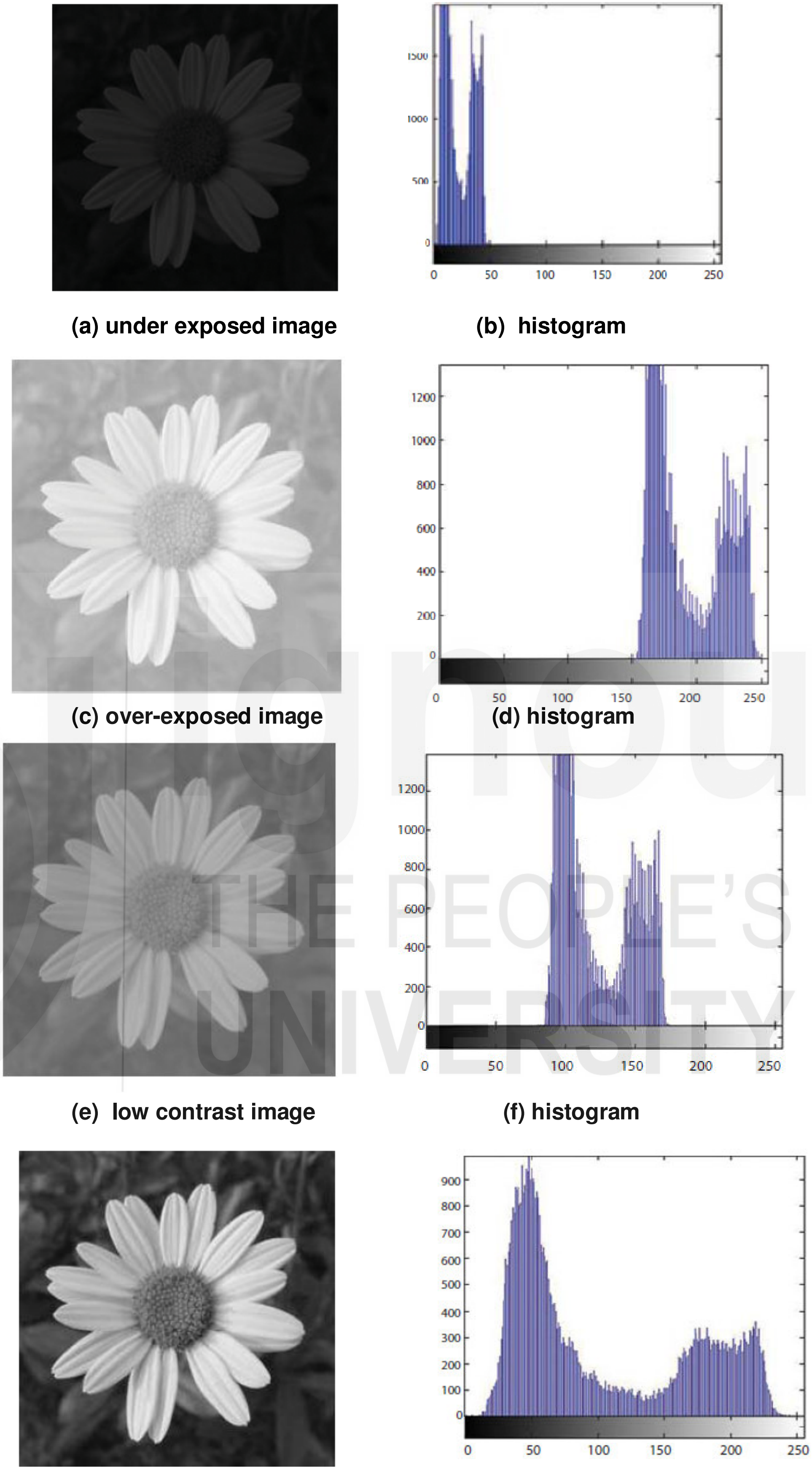
iv) **High contrast image:** Fig. 4 (g) shows a high contrast image and its histogram in Fig. 4 (h). This histogram covers a broad range of grey levels, but distribution of pixels is not uniform. The number of some grey levels is very high as compared to the number of other grey levels.

You can visualize histograms of different images when you will be doing your scilab sessions as given in Block 5 (Lab Manual). Now let us discuss various applications of histogram.

Applications of Histogram

Histogram is widely used in image processing applications.

i) Histogram gives a quick indication as to whether or not we have used the entire dynamic range of digitizer. Information is lost if dynamic range is not used fully.



(a) under exposed image **(b) histogram**

(c) over-exposed image **(d) histogram**

(e) low contrast image **(f) histogram**

(g) high contrast image **(h) histogram**

Fig. 4

ii) Histogram indicates if clipping is a problem in the image.

iii) It is extensively used in thresholding to separate contrasting objects from background.

iv) It tells us about image contrast.

Now, try the following exercises.

E1) What is a histogram?

E2) Highlight the importance of histograms in image processing.

In the following section, we shall discuss histogram equalization.

6.3 HISTOGRAM EQUALIZATION

The histogram of a poor quality image would show presence of only few grey levels, while most would be missing. The number of pixels for various grey levels vary widely. Histogram equalization is a point operation that maps an input image onto a new output image such that there are (almost) equal numbers of pixels at each grey level in output. Thus it can be used for contrast enhancement.

Given information: Input image $f(x, y)$ from which we can calculate its histogram $h(r_k)$.

Goal: To obtain a uniform histogram for the output image. To devise a point operation $s = T(r)$ that maps input image $f(x, y)$ to an output image $g(x, y)$ that has a flat histogram as shown in Fig. 5.

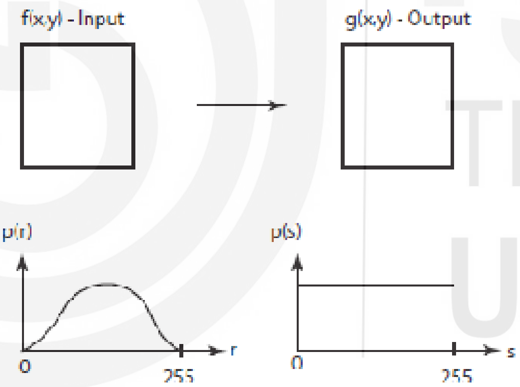


Fig. 5: Histogram Equalization

This is done by adjusting the probability density function (PDF) of the histogram of original image so that probability density function (PDF) spreads equally. We want to find a transformation of the image given in Fig. 6 (a) by the following function:

$$s = T(r), 0 \le r \le 1 \qquad \dots (1)$$

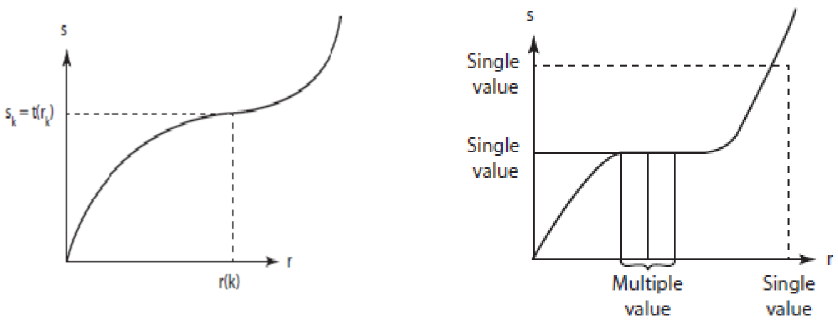
which should satisfy following conditions:

1) $T(r)$ is a single valued function (one to one relationship) as shown in Fig. 6 (b). For each value of r there should be a single value of s. This is needed so that inverse transform exists.

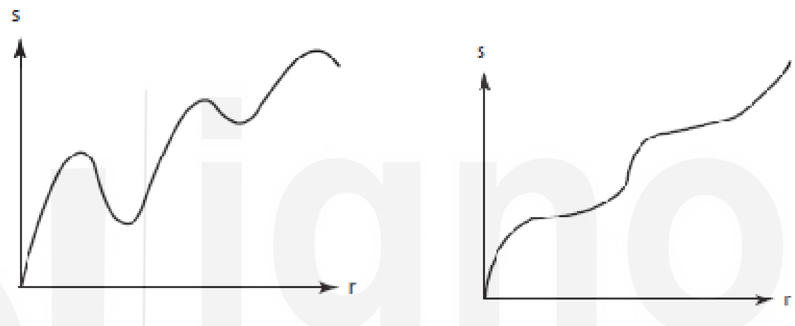
2) $T(r)$ is a monotonically increasing function in the interval $0 \le r \le 1$, Amonotonically increasing function is one which keeps on growing (once it goes up it does not) comedown. This preserves intensity level order in the output image from black to white. This transformation doesn't cause a

negative image. Fig. 6 (c) shows a non-monotonical function and Fig. 6 (d) shows a monotonically increasing function.

3) The range of r and s is same, that is for $0 \le r \le 1, 0 \le T(r) \le 1$. This guarantees that the output grey levels are in the same range as input grey levels.



(a) Transfer Function **(b) Single Valued Function**



(c) Non Monotonical Function **(d) Monotonical Increasing Function**

Fig. 6

Inverse transformation of the function given in Eqn. (1) is defined as $r = T^{-1}(s) 0 \le s \le 1$.

In practice, for integer values of images, many times condition 1 is violated and non-unique inverse transformation is generated.

Probability Density Function (PDF)

The grey levels in an image may be viewed as random variables in the interval $[0,1]$. Probability density function (PDF) is one of the fundamental descriptors of a random variable. PDF of a random variable x is defined as the derivative of cumulative distribution function (CDF)

$$p(x) = \frac{dF(x)}{dx}$$

$F(x) = P(X \le x)$

Where $p(x) =$ PDF of x

$F(x) =$ CDF of x

$p =$ Probability of x

PDF satisfies the following properties:

i) $P(x) \ge 0$ for all x

ii) $\int_{-\infty}^{\infty} p(x)\,dx = 1$

iii) $\int_{-\infty}^{x} p(x)\,d\alpha$ where α is a dummy variable

iv) $p(x_1 \le x \le x_2) = \int_{-x_1}^{x_2} p(x)\,dx$

If a random variable x is transformed by a monotonic transfer function $T(x)$ to produce a new random variable y, then the PDF of y can be obtained from knowledge of $T(r)$ and PDF of x

$$p_y(y) = p_x(x)\left|\frac{dx}{dy}\right|$$

Let $p_r(r)$ be PDF of random variable r of input image.
$P_s(s)$ be PDF of random variables r of output image.
If $p_r(r)$ and $T(r)$ is know then $P_s(s)$ can be obtained by

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

The PDF of transformed variables s is determined by pdf of input image and by chosen transformation function.

Let us now discuss the discrete transformation function.

Transformation function is a cumulative distribution function (CDF) of r.

$$s = T(r) = \int_0^r p_r(w)\,d(w)$$

Where w is dummy variable.

CDF is an integral of a probability function (always positive). Thus CDF is always single valued and monotonically increasing function. CDF satisfies the conditions of transformation function, hence can be considered as $T()$.
Differentiating above equation with respect to r, we get

$$\frac{ds}{dr} = \frac{dt(r)}{dr} = \frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right]$$

$$= p_r(r)$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{p_r(r)}\right| = 1$$

Thus, $\boxed{p_s(s) = 1}$, where $0 \le s \le 1$

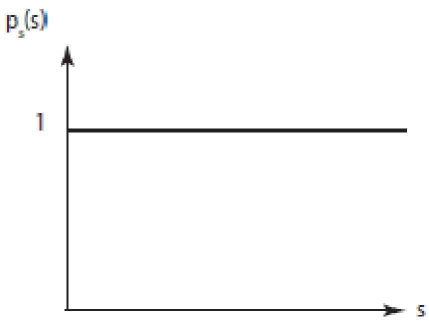


Fig. 7

$p_s(s)$ is PDF of output image, whose value is 1 for all values of s. **Thus $p_s(s)$ is a uniform PDF independent to the form of $p_r(r)$ as shown in Fig. 7.**

Discrete Transformation Function

Although the above discussion shows that any image can be transformed to a flat histogram, it is only possible if grey values are varying continuously. In practice any image will have discrete grey values. Thus the integration above is replaced by a summation function. This means that the output image will not have a flat histogram but only an approximation to it.

Probability of occurrence of grey levels in an image given by

$$p_r(r_k) = \frac{n_k}{n}, \text{where } k = 0..........L-1.$$

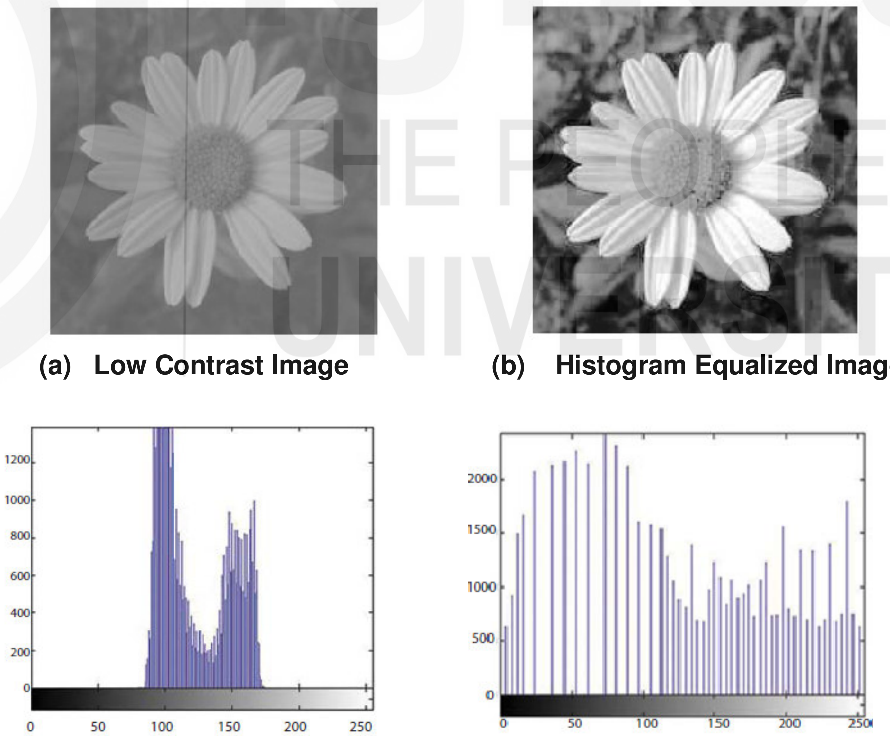
Transformation is given by

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j), \text{ where } k = 0,....L-1.$$

Substituting value of $p_r(r_k)$, we get

$$r_k = \sum_{j=0}^{k} \frac{n_j}{n}$$

The output image is obtained by mapping each pixel with grey level r_k in the input image to a corresponding pixel with grey level s_k in output image.



(a) Low Contrast Image (b) Histogram Equalized Image

(c) Histogram of (a) (d) Histogram of (b)

Fig. 8

Example 3: For the given 4×4 image having grey scales between $[0,9]$, carry out histogram equalization. Also, draw the histogram of image before and after equalization.

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

Solution: Given is

a) The grey levels are between $[0,9]$.
b) Size of image is 4×4.

Following steps are to be followed for histogram equalization:

Step 1: Find the histogram of grey levels of input image by making a table of r_k (grey levels) and number of pixels (n_k) in the input image.

Step 2: For each input grey level, compute the cumulative $\sum_{j=0}^{k} n_j$ sum by adding number of pixels for each grey level.

Step 3: Divide the cumulative sum by the total number of pixels in the image (4 × 4 = 16) to generate the CDF of the given image.

Step 4: Scale the output by multiplying step 3 to maximum grey level value (9 in this case)

Step 5: Round off the output of step 4 to the nearest integer value

We have summarized step 1 to step 5 in the following table.

Grey pixel r_j		0	1	2	3	4	5	6	7	8	9
Step 1	No. of pixels n_j	0	0	6	5	4	1	0	0	0	0
Step 2	Commulative sum $\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
Step 3	$s=\sum_{j=0}^{k}\frac{n_j}{n}$	0	0	$\frac{6}{16}$	$\frac{11}{16}$	$\frac{15}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$	$\frac{16}{16}$
Step 4	$s\times9$	0	0	$\frac{6}{16}\times9$	$\frac{11}{16}\times9$	$\frac{15}{16}\times9$	$\frac{16}{16}\times9$	$\frac{16}{16}\times9$	$\frac{16}{16}\times9$	$\frac{16}{16}\times9$	$\frac{16}{16}\times9$
Step 5	Round off s_k	0	0	≈ 3	≈ 6	≈ 8	≈ 9	9	9	9	9

Step 6: Map input levels (r_k) to output levels s_k in a tabular form from 1st and last row of table above to generate

r_k	0	1	2	3	4	5	6	7	8	9
s_k	0	0	3	6	9	9	9	9	9	9

Step 7: Formulate new image $g(x,y)$ by replacing each level of $f(x,y)$ by a new one using table above.

$$\begin{array}{|c|c|c|c|}\hline 2 & 3 & 3 & 2 \\ \hline 4 & 2 & 4 & 3 \\ \hline 3 & 2 & 3 & 5 \\ \hline 2 & 4 & 2 & 4 \\ \hline \end{array} \xrightarrow{\text{equalized}} \begin{array}{|c|c|c|c|}\hline 3 & 6 & 6 & 3 \\ \hline 8 & 3 & 8 & 6 \\ \hline 6 & 3 & 6 & 9 \\ \hline 3 & 8 & 3 & 8 \\ \hline \end{array}$$

$$f(x, y) \qquad\qquad g(x, y)$$

Step 8: We draw output and input histogram as shown in Fig. 9.

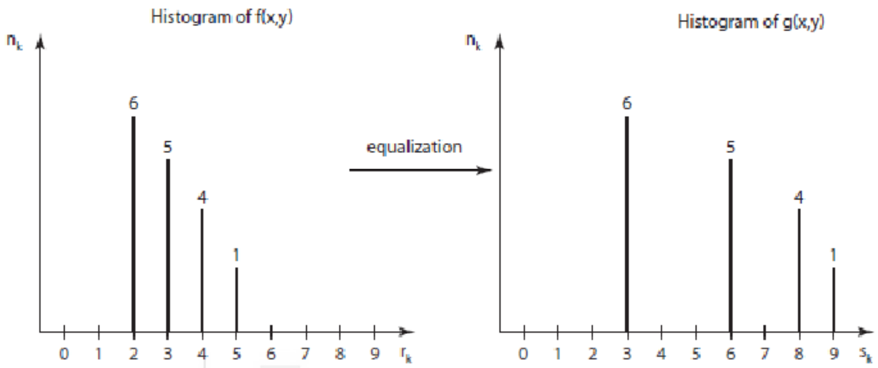


Fig. 9

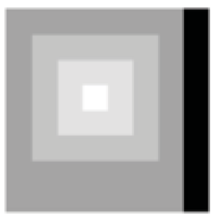
It may be noted that while in the original image the histogram spans between grey values 2 to 5, in the output image, the grey values span from 3 to 9. Although some equalization has been achieved, some of the grey values are still missing. It is because the image has only 9 levels. Histogram equalization will be very effective in an image with 256 grey levels.

Now, try the following exercises.

E3) Write steps for a procedure to perform histogram equalization.

E4) What is the role of histogram equalization in image enhancement? Why this technique yields a flat histogram?

E5) Equalize the histogram of 8×8 image $g(x, y)$ shown below. The input image has grey levels 0, 1, ..., 7.



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

In the next section we shall see how to find transformation for an image, such that the new histogram follows a specific shape. It is called Histogram Specification.

6.4 HISTOGRAM SPECIFICATION

We shall begin with the definition of histogram specification.

Definition: Histogram specification is a point operation that maps input image $f(x, y)$ into an output image $g(x, y)$ with a user specified histogram. An example of specified histogram is shown in Fig. 9.

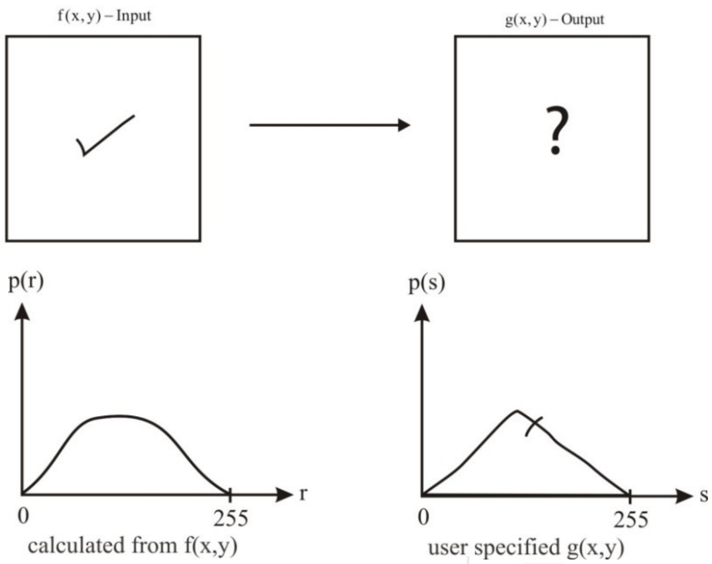


Fig. 9: Histogram specification

The main application of histogram specification is that it improves contrast and brightness of images. It is a pre processing step in comparison of images.

Comparison to histogram equalization

Histogram equalization has a disadvantage that it can generates only one type of output image where histogram is flat, which may not be the best approach under all circumstances. Histogram specification is more general than equalization. It is an interactive enhancement technique where user can draw desired histogram. We can specify the shape of histogram which need not be uniform. Thus, histogram specification gives us the flexibility to choose histogram for reference and output image is mapped accordingly. The basic idea is to carry out histogram equalization of input image histogram, then carry out histogram equalization of specified histogram. The two transformations are used to find the required transformation.

Algorithm for histogram equalization: In this procedure, there are three variables:

$P_r(r)$ is pdf of grey level r of input image

$P_z(z)$ is pdf of grey level z of specified image

$P_s(s)$ is pdf of grey level s of output image

The transformation is $s = T(r) = \int_0^r p_r(r)\,dr$

Histogram equalization of input image $G(z) = \int_0^s p_z(z)\,dz$

To find the histogram equalization of specified image, we equate $G(z)$ is equated to s and an inverse transformation is computed as given below:

$G(z) = s = T(r)$

$\Rightarrow z = G^{-1}[s] = G^{-1}[T(r)]$

Assuming that G^{-1} exists, we can map input grey levels r to output grey levels s.

Procedure for histogram specification

Step 1: Obtain the transformation $T(r)$ by doing histogram equalization of input image.

$$s = T(r) = \int_0^r p_r(r)\,dr$$

Step 2: Obtain the transformation $G(z)$ by doing histogram equalization of specified image.

Step 3: $G(z) = \int_0^z p_z(z)\,dz$

Equate $G(z) = s = T(r)$

Step 4: Obtain inverse transformation function G^{-1}.

$$z = G^{-1}[s] = G^{-1}[T(r)]$$

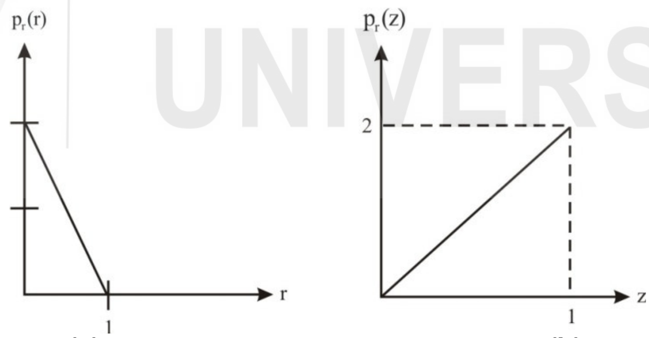
Step 5: Obtain the output image by applying inverse transformation function to all pixels of input image.

Let us now apply these steps in the following example.

Example 4: Assume an image having given grey level PDF $P_r(r)$. Apply histogram specification with given desired PDF function $P_z(z)$ given below

$$p_r(r) = \begin{cases} -2r+1; & 0 \le r \le 1 \\ 0\ ; & \text{otherwise} \end{cases} \quad p_s(r) = \begin{cases} 2z; & 0 \le z \le 1 \\ 0; & \text{otherwise} \end{cases}$$

Apply histogram specification with the desired PDF function $p_z(z)$ given in Fig. 10:



(a) (b)

Fig. 10

Solution: We shall solve it step wise.

Step 1: Obtain transformation function $T(r)$ by doing histogram equalization of input image.

$$s = T(r) = \int_0^r p_r(r)\,dr = \int_0^r (-2r+2)\,dr - [-r^2 + 2r]_0^r = -r^2 + 2r$$

Step 2: Obtain transformation function $G(z)$

$$G(z) = \int_0^z p_z(z)\,dz = \int_0^z 2z\,dz = [z^2]_0^z = z^2$$

Step 3: Equate

$$s = T(r) = G(z)$$
$$-r^2 + 2r = z^2$$

Step 4: Obtain inverse transformation G^{-1}.

$$z = G^{-1}[T(r)]$$
$$z = \sqrt{-r^2 + 2r}$$

***

Now, we shall find the discrete transformation function for an actual image.

Discrete Transformation Function for an actual Image

Histogram equalization of input image.

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j), k = 0,...,L-1$$
$$s_k = \sum_{j=0}^{k} \frac{n_j}{n}$$

where n = total number of pixels in input image
n_j = number of pixels having level j

Histogram equalization of specified image

$$v_q = G(z_q) = \sum_{i=0}^{q} p_z(z_i), q = 0,..,L-1$$

Equate $G(z_q) = s_k = T(r_k)$

Inverse transformation $z_q = G^{-1}[s_k] = G^{-1}[T(r_k)]$

The block diagram of histogram specification is given in Fig 11.

v^* is that value of v_q so that min $[v_q - s] \geq 0$
p = Corresponding value of z.

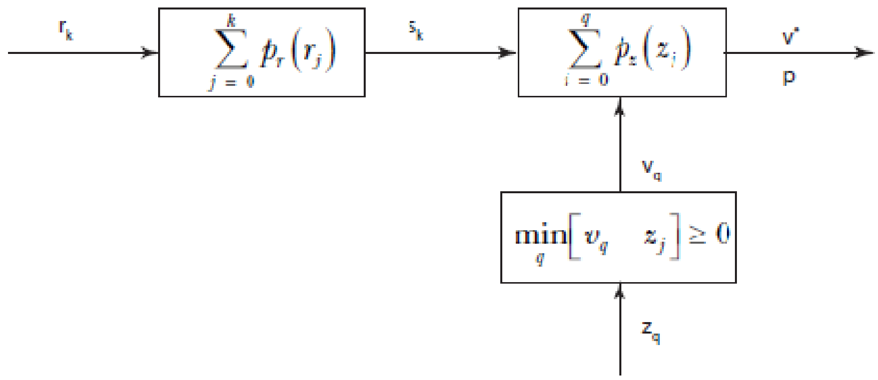


Fig. 11: Histogram Specification

Procedure for histogram specification of an image

Step 1: Equalize input image histogram (s_k)
Step 2: Equalize specified image histogram (v_q)
Step 3: For $\min q[v_q - s] \geq 0$ find corresponding v^* and p.
Step 4: Map input pixels to output pixels to get output image.

We shall apply these steps in the following example.

Example 6: Apply histogram specification on given image having

0	1	0	2
2	3	3	2
3	1	0	1
1	3	2	0

$r_i = z_i = 0,1,2,3$
$p_r(r_i) = .25 \text{ for } i = 0,1,2,3$
$p_z(z_0) = 0, p_z(z_1) = .5,\ p_z(z_2) = .5, p_z(z_3) = 0$

Solution: We shall apply each step of the procedure.

Step 1: Equalize input image histogram.

r_k	0	1	2	3
$p_r(r_k)$	.25	.25	.25	.25
s_k	.25	.5	.75	1

Step 2: Equalize specified image histogram

z_q	0	1	2	3
$p_z(r_z)$	0	.5	.5	0
v_q	0	.5	1	1

Step 3: Find minimum value of q such that $(v_q - s) \geq 0$. First three columns are filled by step 1, next three columns are filled by step 2. In this step, last two columns are filled by the fallowing procedure.

r_k	$p_r(r_k)$	s_k	z_q	$p_z(z_q)$	v_q	v^4	p
0	.25	.25	0	0	0	.5	1
1	.25	.5	1	.5	.5	.5	1
2	.25	.75	2	.5	1	1	2
3	.25	1	3	0	1	2	2

a) $q = 0, k = 0 \quad (v_0 - s_0) = (0 - .25) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increase} q$
$q = 1, k = 0 \quad (v_1 - S_0) = (.5 - .25) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_0 = v_1 = .5$ and $p_0 = z_1 = 1$

b) $q = 1, k = 1 \quad (v_1 - s_1) = (.5 - .5) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_1 = v_1 = .5$ and $p_1 = z_1 = 1$

c) $q = 1, k = 2 \quad (v_1 - s_2) = (0 - .75) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increase}\, q$
$q = 2, k = 2 \quad (v_2 - s_2) = (1 - .75) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_2 = v_2 = 1$ and $p_2 = z_2 = 2$

d) $q = 2, k = 3 \quad (v_2 - s_2) = (1 - .75) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_3 = v_2 = 1$ and $p_3 = z_3 = 2$

Step 4: Map input level values to output level values

r_k	0	1	2	3
p	1	1	2	2

Step 5: Map input pixels to new values to get new image using step 4 output.

$$\begin{array}{|c|c|c|c|}\hline 0&1&0&2\\ \hline 2&3&3&2\\ \hline 0&1&0&1\\ \hline 1&3&2&0\\ \hline \end{array} \Longrightarrow \begin{array}{|c|c|c|c|}\hline 1&1&1&2\\ \hline 2&2&2&2\\ \hline 2&1&1&1\\ \hline 1&2&2&1\\ \hline \end{array}$$

$f(x, y)$ $\qquad$ $g(x, y)$

This pixel mapping is used to map the pixels of input image to generate output image with specified histogram. Histogram specification is a trial and error process. There are no rules for specifying histogram and one must resort to analysis on a case by case basis for any given enhancement task.

***

Try the following exercises.

E6) Define the concept of Histogram matching with appropriate example.

E7) Perform histogram specification from the given data.

$r = 0,1,2,3,4$

$p(r_i) = 0.2$ for $i = 0.......4$

$z_i = 0,2,4,5,7$

$p(z_0) = 0,\ p(z_2) = 0.2,\ p(z_4) = 0.4,\ p(z_5) = 0.4,$

$p(z_7) = 0$

Let us now summarise what we have studied in this unit.

6.5 SUMMARY

1) Defined histogram
2) Understood how to draw histogram from the image
3) Perform histogram equalization
4) Perform histogram specification

6.6 SOLUTIONS/ANSWERS

E1) Histogram is a graph between various grey levels on x-axis and the number of times a grey level has occurred in an image, on y-axis. Histogram of an image is defined as

$$h(r_k) = \frac{nk}{k}, = 0............L-1$$

$r_k = k^{th}$ grey level

$n_k =$ number of birds grey level values as r_k.

E2) The histogram of an image represents the relative frequency of occurrence of the various gray levels in the image. It also provides a useful image statistics which are helpful in various image processing applications, for example, thresholding, intensity level slicing, segmentation etc. Besides it is very simple to calculate histogram. Intuitively, it tells how vivid or washed out an image appears. Histograms are the basis for many spatial domain processing techniques.

E3) Histogram equalization is done by the transformation

$$\boxed{s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)} \qquad , k = 0, \ldots\ldots L-1$$

E4) Histogram equalization is a point operation that maps an input image onto output image such that there are equal number of pixels at each grey level in output. It is used for contrast enhancement.

Given information: Input image $f(x, y)$ from which we can calculate its histogram $h(r_k)$

Goal: To obtain a uniform histogram for the output image. To device a point operation $s = T(r)$ that maps input image $f(x, y)$ to an output image $g(x, y)$ that has a flat histogram

E5) Given size: 8×8, levels $[0\ 7]$

	r_j	0	1	2	3	4	5	6	7
Step 1	n_k	8	0	0	0	31	16	8	1
Step 2	$\sum_{j=0}^{k} n_j$	8	8	8	8	39	55	63	64
Step 3	$S_k = \sum_{j=0}^{k} \frac{n_j}{n}$	$\frac{8}{64}$	$\frac{8}{64}$	$\frac{8}{64}$	$\frac{8}{64}$	$\frac{39}{64}$	$\frac{55}{64}$	$\frac{63}{64}$	$\frac{64}{64}$
Step 4	$s_k \times 7$	$\frac{8}{64}\times 7$	$\frac{8}{64}\times 7$	$\frac{8}{64}\times 7$	$\frac{8}{64}\times 7$	$\frac{39}{64}\times 7$	$\frac{55}{64}\times 7$	$\frac{63}{64}\times 7$	$\frac{64}{64}\times 7$
Step 5	Round off	≈ 1	≈ 1	≈ 1	≈ 1	4	6	7	7

Step 1: Find the histogram of input image by making a table of r_k and n_k.

Step 2: For each input level, find cumulative sum

Step 3: Divide cumulative sum by total number of pixels $n = 64$.

Step 4: Scale the output by multiplying step 3 by 7.

Step 5: Round off to nearest integer value to find s_k.

Step 6: Map input levels r_k to output levels s_k using 1st and last row of table above to make table below.

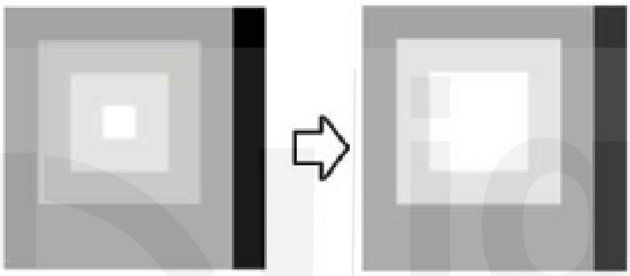
r_k	0	1	2	3	4	5	6	7
s_k	1	1	1	1	4	6	7	7

Step 7: Formulate a new image $g(x, y)$ by replacing each level of $f(x, y)$ by a new one using table.

Step 8: Draw input and output histograms.

$$\begin{matrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 \\ 4 & 5 & 5 & 5 & 5 & 5 & 4 & 0 \\ 4 & 5 & 6 & 6 & 6 & 5 & 4 & 0 \\ 4 & 5 & 6 & 7 & 6 & 5 & 4 & 0 \\ 4 & 5 & 6 & 6 & 6 & 5 & 4 & 0 \\ 4 & 5 & 5 & 5 & 5 & 5 & 4 & 0 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 0 \end{matrix} \Rightarrow \begin{matrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 1 \\ 4 & 6 & 6 & 6 & 6 & 6 & 4 & 1 \\ 4 & 6 & 7 & 7 & 7 & 6 & 4 & 1 \\ 4 & 6 & 7 & 7 & 7 & 6 & 4 & 1 \\ 4 & 6 & 7 & 7 & 7 & 6 & 4 & 1 \\ 4 & 6 & 6 & 6 & 6 & 6 & 4 & 1 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 1 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 & 1 \end{matrix}$$

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



f(x) g(x)

E6) Histogram specification is more general than equalization. It is an interactive enhancement technique where user can draw desired histogram. We can specify the shape of histogram which needs not to be uniform. Thus, histogram specification gives us the flexibility to choose histogram for reference and output image is mapped accordingly

E7) **Step 1:** Equalize input histogram.

r_k	0	1	2	3	4
$p(r_k)$	.2	.2	.2	.2	.2
s_k	.2	.4	.6	.8	1

Step 2: Equalize specified histogram

r_k	0	1	2	3	4
$p(r_k)$	.2	.2	.2	.2	.2
s_k	.2	.4	.6	.8	1

Step 3: Find minimum value of q such that $[vq - s] \geq 0$

1) $q = 0, k = 0$ $(v_0 - s_0) = (0 - .2) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increaseq}$

$q = 1, k = 0$ $(v_1 - s_0) = (.2 - .2) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_0 - v_1 = .2$ and $p_0 = z_1 = 2$

2) $q = 1, k = 1$ $(v_1 - s_1) = (.2 - .4) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increase}\, q$

$q = 2, k = 1 \quad (v_2 - s_2) = (.2 - .4) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_1 = v_2 = .6$ and $p_1 = z_2 = 4$

3) $q = 2, k = 2 \quad (v_2 - s_2) = (.6 - .6) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_2 = v_2 = .6$ and $p_2 = z_2 = 4$

4) $q = 3, k = 3 \quad (v_2 - s_2) = (.6 - .8) \geq 0 \Rightarrow \text{No} \Rightarrow \text{increase}\, q$
$q = 3, k = 3 \quad (v_q - s_q) = (1 - .8) \geq 0 \Rightarrow \text{Yes} \Rightarrow v^*{}_q = v_q = 1$ and $p_q = z_q = 5$

5) $q - 3, k - 4 \quad (v_3 - s_4) = (1 - 1) \geq 0 \Rightarrow \text{Yes} \Rightarrow\Rightarrow v^*{}_4 = v_3 = 1$ and $p_4 = z_3 = 5$

S.No.	r_k	$p(r_k)$	s_k	z_q	$p(z_q)$	v_q	v^*	p
1	0	.2	.2	0	0	0	.2	2
2	1	.2	.4	2	.2	.2	.6	4
3	2	.2	.5	4	.4	.6	.6	4
4	3	.2	.8	5	.4	1	1	5
5	4	.2	1	7	0	1	1	5

Step 4: Input and output grey level mapping from 2nd and last row of previous output

r	0	1	2	3	4
p	2	4	4	5	5

UNIT 7

NEIGHBOURHOOD OPERATIONS

Structure ______________________ Page No.

7.1 INTRODUCTION

In the previous unit, we had seen how image transformation carried out on a specific pixel depends on the gray value of that pixel. In this unit, we shall consider more powerful transformation where the gray values of the neighbouring pixels also play a useful part. We have discussed point processing operations of the type $s = T(r)$, where r is the grey level at a single pixel in the input and s is the new value of that pixel. The capabilities of point operations are limited as the relation between a pixel and its neighbours is not exploited.

In this unit, we will be considering a neighbourhood of pixels from the input image. This is termed as **spatial filtering**. We introduce image enhancement with the help of image sharperning (spatial high pass) and image smoothing (low pass filters). Spatial filtering as generally used in preprocessing operations for noise removal or for edge detection/enhancement 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

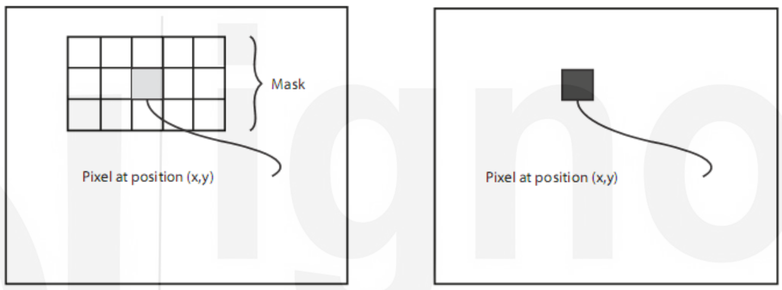
- apply spatial filtering
- perform linear filtering in spatial domain

- Use differentiate low pass & high pass filtering in spatial domain
- Use differentiate non-linear filtering and linear filtering
- Perform median, min & max filters in spatial domain

7.2 SPATIAL FILTERING

Filtering is a process that removes some unwanted components or small details in an image. In digital image processing, filter is basically a subimage and is known by various names such as mask, kernel, template or window. Filters can be of two types: Spatial filters and frequency domain filters.

We have discussed point processing operations of the type $s = T(r)$, where r is the grey level at a single pixel in the input image and s is the new value of that pixel. The capabilities of point operations are limited as the relation between a pixel and its neighbors is not exploited. In this section, we will be considering a neighborhood of pixels from the input image. **Spatial filtering** is one of the main tools used in variety of applications such as noise removal, bridging the gaps in object boundaries, sharpening of edges etc.



(a) Original Image **(b) Image after filtering**

Fig. 1: Linear Filtering

The idea of spatial filtering is to move a 'mask', a square of odd size (such as 3x3, 5x5 etc.) over the whole image pixel by pixel. At each pixel, the corresponding value of the mask and the image are multiplied and added up to replace the original grey value of the pixel. This process is known as "**convolution**".

By this process, we create a new image where grey level values of the pixels are calculated from the values under the mask. The values under the mask are modified by a function called '**filter**'. If this filter function is a linear function of all grey level values in the mask, then filter is called a '**linear filter**', else it is called '**non linear filter**'.

Spatial filters can be of various types as shown in Fig. 2.

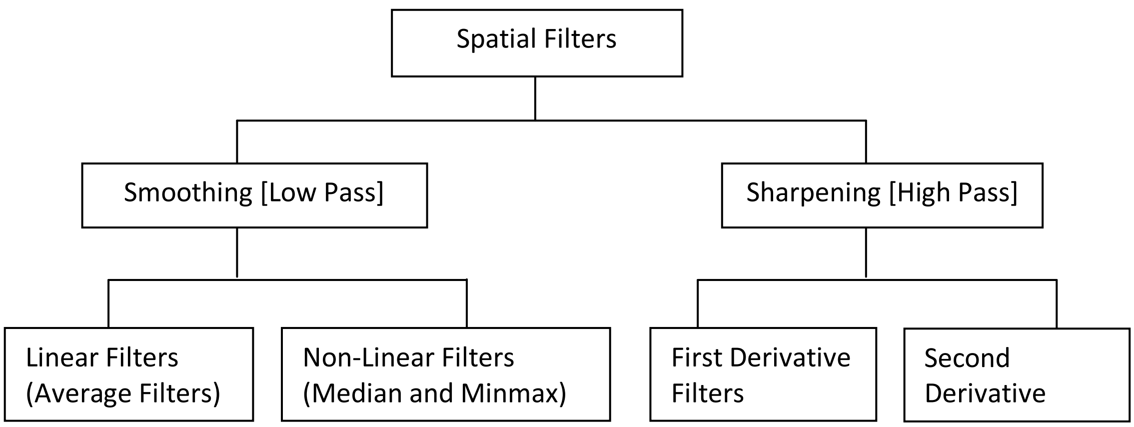


Fig. 2: Types of Spatial Filters

In this unit, we will discuss two major types of spatial filters.

i) **Smoothing:** This type of filter is used to blur (smooth) the image, therefore called smoothing filter also. These filters are also used for noise reduction. Noise reduction can be done by blurring with a linear filter, (in which operation performed on the image pixels is linear) or a non-linear filter.

ii) **Sharpening:** This filter is used to highlight transitions in intensity. These are based on first and second derivatives.

The effect of these filters is shown in Fig. 3.

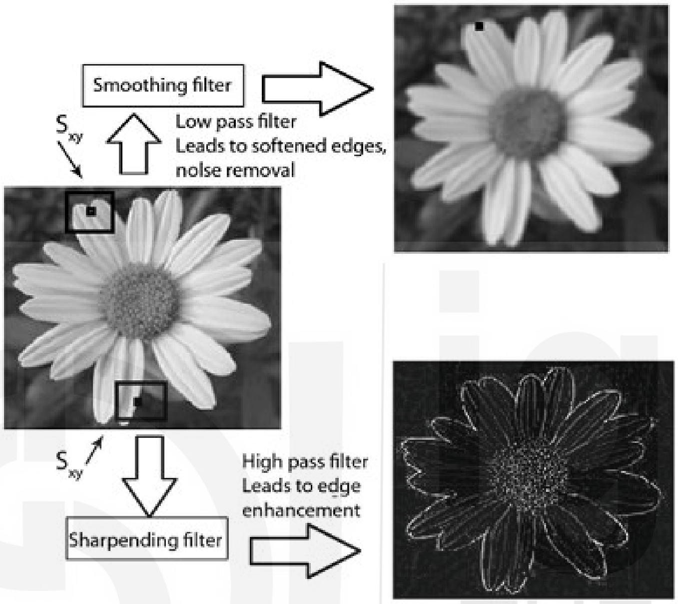


Fig. 3: Smoothing and Sharpening Filters

In the following section, we discuss image smoothing filters in detail.

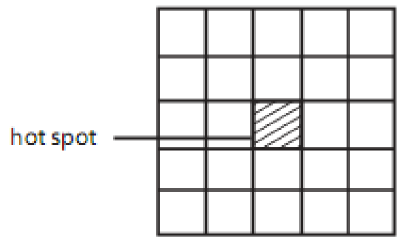
7.3 IMAGE SMOOTHING

We discussed spatial filters in previous section. Spatial filter consists of a typically small rectangle called neighbourhood and a predefined operation which is performed on the pixels of the image encompassed by the neighbourhood. Such filtering creates a new pixel with coordinates equal to the coordinates of the center of the neighbourhood, and whose value is the outcome of the filtering operation. Thus, a filtered image is generated as the center of the filter visits each pixel in the input image. If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter. Otherwise, the filter is nonlinear. Here, we shall discuss first linear filters and then some simple nonlinear filters.

Let us discuss linear spatial filters to start with.

7.3.1 Linear Filters

Linear filtering is a spatial domain process where a filter (mask/ kernel/ template) with some integer coefficient values is applied to input image to generate the filtered/ output image. Generally, filter size is either 3×3 or $5\times5, 7\times7$ or 21×21 (odd sizes) and filter is centered at a coordinate (x, y) called '**Hot Spot**' as shown in Fig. 4.



(b) 5 x 5 mask

Fig. 4

Linear filtering of an image $f(x,y)$ of size $M \times N$ and filter mask of size $m \times m$ is given by

$$g(x,y) = \sum_{(i,j)\in R_n} f(x+i,\ y+j)\, w(i,j) \text{ where } 0 \le x \le M-1 \text{ and } 0 \le y \le N-1. \qquad (1)$$

The right hand side of Eqn. (1) denotes the set of coordinates covered by filter. For a 3×3 filter, with coefficients $w(i,j)$ the output image at coordinate (x,y) is calculated as

$$g(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x+i, y+j)\, w(i,j) \qquad (2)$$

$$\begin{aligned} g(x,y) = {} & f(x,y)\,w(0,0) + f(x,y+1)\,w(0,1) + f(x,y-1)w(0,-1) + f(x-1,y-1) \\ & w(-1,-1) + f(x-1,y)\,w(-1,0) + f(x-1,y+1)w(-1,1) + \\ & f(x+1,y-1)\,w(1,-1) + f(x+1,y)\,w(1,0) + f(x+1,y+1)\,w(1,1) \end{aligned}$$

We carry out the following steps for linear filtering:

Step 1: Position the mask over the current pixel such that hotspot w(0,0) coincides with current pixel.

Step 2: Form all products of filter elements with the corresponding elements in the neighbourhood.

Step 3: Add up all the products and store it at current position in the output image.

Step 4: Divide it by the scaling constant (sum of all coefficients of the mask).This must be repeated for every pixel in the image.

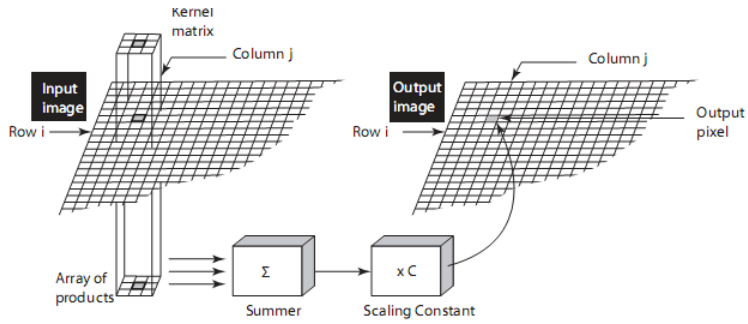


Fig 5: Linear filtering

Let us apply these steps in the following example.

Example 1: Apply given 3×3 mask w on the following image $f(x,y)$.

$$f(x,y)=\begin{array}{|ccccc|}\hline 5 & 1 & 2 & 6 & 7\\ 4 & 4 & 7 & 5 & 8\\ 2 & 6 & 20 & 6 & 7\\ 3 & 1 & 2 & 4 & 5\\ 10 & 2 & 1 & 2 & 3\\ \hline\end{array},\ w(i,j)=\frac{1}{9}*\begin{array}{|ccc|}\hline 1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\\ \hline\end{array}$$

Solution: First we consider top left 3×3 neighborhood in image $f(x,y)$ as shown in Fig. 6 (a) with hot spot coinciding with grey level 4 in $f(x,y)$ and calculate output for that pixel position by using equation

$$\text{Output } =\frac{1}{9}[5\times 1+1\times 1+2\times 1+4\times 1+4\times 1+7\times 1+2\times 1+6+1+20\times 1]$$

$$=\frac{49}{9}\approx 5$$

$$\begin{array}{|ccccc|}\hline 5 & 1 & 2 & 6 & 7\\ 4 & 4 & 7 & 5 & 8\\ 2 & 6 & 20 & 6 & 7\\ 3 & 1 & 2 & 4 & 5\\ 10 & 2 & 1 & 1 & 3\\ \hline\end{array}\qquad\qquad \begin{array}{|ccccc|}\hline 5 & 1 & 2 & 6 & 7\\ 4 & 4 & 7 & 5 & 8\\ 2 & 6 & 20 & 6 & 7\\ 3 & 1 & 2 & 4 & 5\\ 10 & 2 & 1 & 1 & 3\\ \hline\end{array}$$

(a) **(b)**

Fig. 6

Shift the mask one pixel towards right such that grey level 7 coincides with hot spot (Fig. 6 (b)). Calculate output for that pixel position and replace in output image. Shift it right again and locate the mask on pixel with grey value 5, and repeat the process.

$$\text{Output } =\frac{1}{9}[1\times 1+2\times 1+6\times 1+4\times 1+7\times 1+5\times 1+6\times 1+20\times 1+6\times 1]$$

$$=\frac{57}{9}\approx 6$$

Now go back to the first position of the mask and shift it down so that it is now centered on pixel with grey value 6 (just below 4), and repeat the process.

$$\text{Output } =\frac{1}{9}[2\times 1+6\times 1+7\times 1+7\times 1+5\times 1+8\times 1+20\times 1+6\times 1+7\times 1]$$

$$=\frac{68}{9}\approx 8$$

Continue doing it over all the possible pixels in the given image.

$$\text{Output } =\frac{1}{9}[4+4+7+2+6+20+3+1+2]=\frac{49}{9}\approx 5$$

$$\text{Output } =\frac{1}{9}[4+7+5+6+20+6+1+2+4]=\frac{55}{9}\approx 6$$

$$\text{Output } = \frac{1}{9}[7+5+8+20+6+7+2+4+5] = \frac{64}{9} \approx 7$$

$$\text{Output } = \frac{1}{9}[2+6+20+3+1+2+10+2+1] = \frac{38}{9} \approx 4$$

$$\text{Output } = \frac{1}{9}[6+20+6+1+2+4+2+1+2] = \frac{44}{9} \approx 5$$

$$\text{Output } = \frac{1}{9}[20+6+7+2+4+5+1+2+3] = \frac{59}{9} \approx 6$$

So, by shifting the mask over the whole image a new image is generated. These values are rounded off to nearest integer values and put in the output image as given below at corresponding pixel locations.

$$g(x,y) = \begin{array}{|ccccc|} \hline * & * & * & * & * \\ * & 5 & 6 & 8 & * \\ * & 5 & 6 & 7 & * \\ * & 4 & 5 & 6 & * \\ * & * & * & * & * \\ \hline \end{array}$$

***

The filtering effect is visible in the new image where the noisy values 10, 20 in original image have been replaced by smoothed values.

Remark: There is a small problem in applying a filter at the edge of the image, where the mask partly falls outside the image. In Example 1, input image size $= 5\times5$, and the mask size $= 3\times3$. Therefore, the output image size $= 3\times3$. The size of output image is smaller than input image size as the mask does not overlap fully in the 1st row, 1st column last column and last row as shown in image $g(x, y)$.

However, it is not a serious problem as in practical situations the image sizes are much larger such as 200×200 and loss of few values at the boundary of the image is not even visible to us.

Now, try the following exercises.

E1) For a 3×3 mask, explain the mechanics of linear filtering with necessary equations.

E2) List various applications of linear filtering.

Now, we discuss one of the commonly used linear filter known as a mean filter.

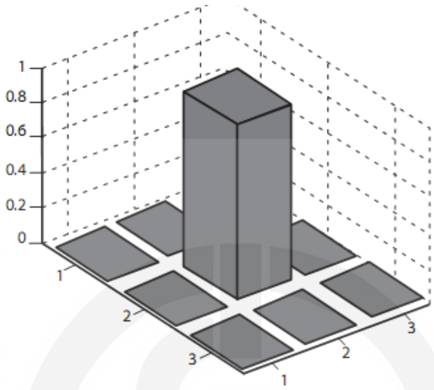
The aim of image smoothing is to diminish the effect of camera noise, spurious pixel values, missing pixel values etc. This is done by a **neighborhood averaging filter.** Each pixel in the smoothened image $g(x, y)$ is obtained from the average pixel value in the neighborhood of (x, y) in input image. Such a mask is also known as **a Mean filter.**

$$g(x, y) = \frac{1}{M} \sum_{(m,n) \in S_{xy}} f(m, n)\, w(m, n) \qquad (3)$$

Where $Sxy =$ neighbourhood, $M =$ Number of pixels in Sxy and w is the mask for averaging.

Two masks averaging and weighted averaging masks are shown in Fig. 7. The output is average of pixels contained in the neighbourhood of filter mask.

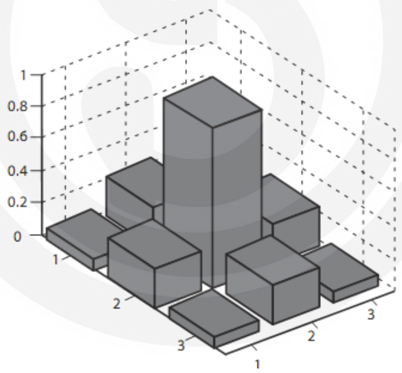
Therefore, smoothing filters are also called averaging filters or low pass filters. Fig. 7 (a) and Fig. 7 (b) shows 3×3 smoothing filter masks. Spatial filter, where all coefficients are equal, is called *box filter*. However, to give maximum importance to pixel at the centre of the mask, it is given maximum weight, and the weights of the neighnouring pixels are progressively reduced.



$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

(a) Box Filter (3×3 Averaging Mask)



$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

(b) 3×3 Weighted Averaging Mask

Fig. 7

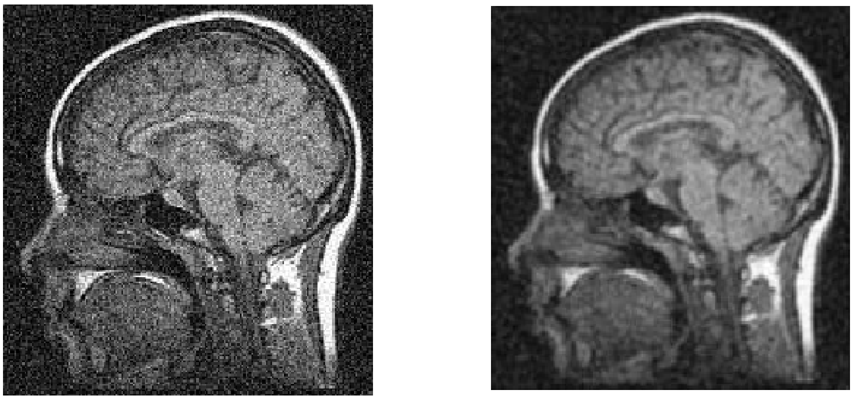
In this case centre pixel is given the most importance and other pixels are inversely weighted as a function of their distance from the centre of the mask. This filter reduces blurring in the smoothing process as the centre pixel is weighted highest.
Smoothing filter is used for the following purposes:

i) It reduces ‘sharp’ transitions in grey levels.
ii) It reduces noise.
iii) It blurs edges. This is a **side effect.**
iv) It helps in smoothing false colours.
v) It reduces ‘irrelevant’ details in an image.

‘**Blurring**’ (a side effect of smoothing filter) is used in preprocessing steps for following applications:

i) Removal of small details from an image prior to object extraction. Intensity of smaller objects blends with the background and larger object become 'blob like' and easy to detect.

ii) Bridging small gaps in lines or curves of the boundaries of objects and text. Small gaps in boundary can lead to enormous results in object extraction. Smoothing fills up all smaller gaps and helps in producing correct result.



(a) noisy image **(b) output of 3×3 averaging filter**

Fig. 8

See this example.

Example 2: Apply averaging filter to input image $f(x, y)$ as given in Fig. 10 to produce output image $g(x, y)$ in Fig. 10.

$$\underset{f(x,y)}{\begin{array}{|ccccc|}\hline 5 & 1 & 2 & 6 & 7 \\ 4 & 4 & 7 & 5 & 8 \\ 2 & 6 & 20 & 6 & 7 \\ 3 & 1 & 2 & 4 & 5 \\ 10 & 2 & 1 & 2 & 3 \\ \hline\end{array}} \xrightarrow[\text{filter}]{\text{smoothing}} \underset{g(x,y)}{\begin{array}{|ccc|}\hline 5 & 6 & 8 \\ 5 & 6 & 7 \\ 4 & 5 & 6 \\ \hline\end{array}}$$

Fig. 9

Solution: Smoothing filter replaces every pixel of the input image by the average of the grey levels in the neighborhood. It reduces sharp transitions in grey levels. Note that pixel value 20 in $f(x, y)$ has been changed to 6 in the output image.

Sharp transitions can be due to

- Random noise in the image
- Edge of objects in the image

Smoothing filter reduces noise (desirable) and also blurs edges (undesirable). In the image $f(x, y)$, pixel value 20(high value as compared to neighborhood) becomes 6, pixel value 1 (low value as compared to neighborhood) becomes 4, whereas pixel value 7(similar value as compared to neighborhood) becomes 6.

Thus, noisy pixel or edges (20, 1: arbitrarily high/ low values) are reduced to average value, closer to neighborhood values. Note that smoothing effect will be more if instead of 3x3 mask, a large sized mask such as 7×7 or 11×11

mask is used. The size of the mask will depend on amount of noise and size of the image.

***

Try these exercises.

E3) With various spatial masks and equations, explain spatial averaging technique for enhancement.

E4) Discuss image smoothing filter in the spatial domain.

7.3.2 Nonlinear Filters

Smoothing linear filters have an important disadvantage, image structures such as individual points, edges and lines are blurred. Overall quality of the image reduces in some applications. These side effects are not tolerated which limits the usage of linear filters. Order statistics filters are non-linear filters whose response is based on ordering (ranking) the pixels contained in the image area encountered by the filter. Like all other spatial filters, non linear filters compute the result at some position (x, y) from the pixels inside the moving region S of the original image. These filters are called **non-linear** because source pixels are processed by some non-linear function.

Here, we discuss some simple non linear filters such as median filter and Min/Max filters.

Median Filters

Median filters are edge preserving smoothing filters, where the level is set to the median of pixel values in the neighborhood of that pixel. It is impossible to design a filter that removes only noise and retains all the important image structures intact, because no filter can discriminate which image content is important to the viewer and which is not. Median filter replaces every image pixel by median of the pixels in the corresponding filter region Sxy.

$$g(x, y) = \text{median}\{f(x+i, y+j) \mid (i, j) \in Sxy\} \quad (5)$$

Median of $2k+1$ pixel values is defined as median $(p_0, p_1, p_2, ..., p_k, p_{k+1}, ..., p_{2k}) \triangleq p_k$. Median is the centre value p_k if the sequence $(p_0, p_1, ..., p_{2k})$ is sorted.

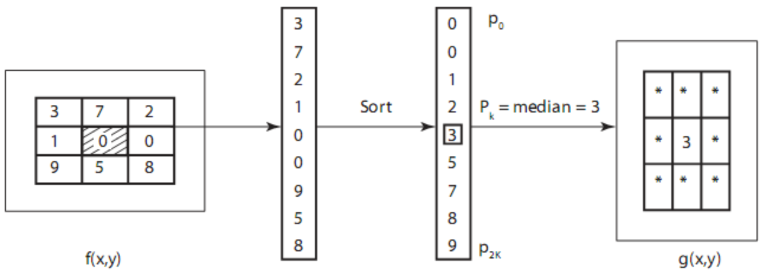


Fig 10: Median Filtering

If the number of elements are even (2k) the median of the sorted sequence $(p_0, p_1, ..., p_{k-1}, p_k, ..., p_{2k-1})$ is defined as arithmetic of two middle values

Thus, median $(p_0, p_1, ..., p_{k-1}, p_k, ..., p_{2k-1}) = \frac{p_{k-1} + p_k}{2}$

Fig. 10 shows median filtering process and generating output image. Median filter is very popular because for impulse noise (salt and pepper noise, randomly placed white and black dots), it provides excellent noise reduction capabilities with considerably less blurring than linear smoothing filter of same size. Median filters force the points with distinct levels to be more like their neighbors. It eliminates isolated clusters of pixels that are dark or light with respect to their neighbors and whose area is less than $k^2/2$ (one half of filter area).

Advantages of Median Filters

- It has excellent noise reduction capability.
- It preserves edges.
- It introduces less blurring than smoothing linear filters.
- It is very effective in the presence of impulse noise.

Disadvantages

- It performs poorly if number of noise pixels in S
- is greater than half the number of pixels in the windowxy
- It performs poorly in the presence of gaussian noise.

Steps to Implement Median Filter as shown in Fig. 10.

Step 1: Sort the values of the pixel in question and its neighbours.
Step 2: Determine their median value.
Step 3: Assign median value to that pixel.

Example 3: Compute the median value of the marked pixels show below using a 3×3 mask

$$\begin{bmatrix} 18 & 22 & 33 & 25 & 32 & 24 \\ 34 & 128 & 24 & 172 & 26 & 33 \\ 22 & 19 & 32 & 31 & 28 & 26 \end{bmatrix}$$

Solution: i) Median $(18, 22, 33, 34, 128, 24, 22, 19, 32)$
$= \text{Median } (18, 19, 22, 22, 24, 32, 33, 34, 128) = 24$

ii) Median $(22, 33, 25, 128, 24, 172, 19, 32, 31)$
$= \text{median}(19, 22, 24, 25, 31, 32, 33, 128, 172) = 31$

iii) Median $(33, 25, 32, 34, 172, 26, 32, 31, 28)$
$= \text{median}(24, 25, 26, 28, 31, 32, 32, 33, 172) = 31$

iv) Median $(25, 32, 24, 172, 26, 23, 31, 28, 26)$
$= \text{median}(23, 24, 25, 26, 26, 28, 31, 32, 172) = 26$

***

Minimum and Maximum Filter

Max and min filters are defined as:

$$g(x, y) = \min\{f(x + i, y + j) | (i, j) \in S_{xy}\} \quad (6)$$

$$g(x,y) = \max\{f(x+i, y+j) \mid (i,j) \in S_{xy}\}, \quad (7)$$

where S_{xy} denotes the filter region, usually a size of 3×3 pixels. **Min filter** removes salt noise (white dots with large grey level values) because any large grey level with in a 3×3 filter region is replaced by one of its surrounding pixels with smallest value. As a side effect, min filter introduces dark structures in the image. The reverse effect is expected from a **max filter**. It removes pepper noise (black dots with small grey level values) because any black dot within 3×3 filter region is replaced by one of its surrounding pixels with the largest value. White dots/bright structures are widened as a side effect and black dots (pepper noise) will disappear. Fig. 12 shows the process of min/max filter.

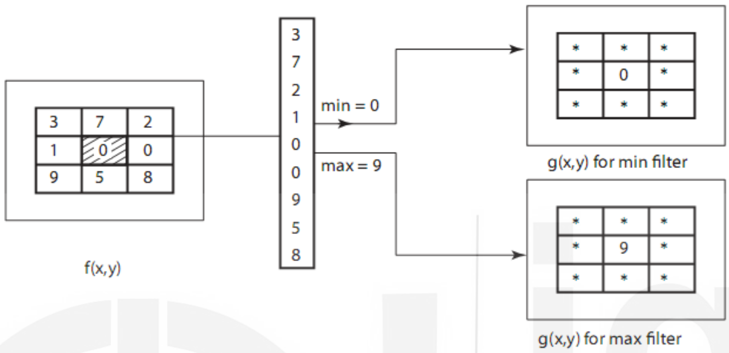


Fig. 11: Min, max filtering

Example 4: For the image segment $f(x,y)$ given below.

f(x,y) =

0	1	0	6	5
2	3	1	2	5
1	2	<u>7</u>	5	4
1	0	6	5	2
2	3	5	7	6

Apply

a) Smoothing filter
b) Weighted average filter
c) Median filter
d) Min filter
e) Max filter

Solution: (a) Smoothing filter of size 3×3

Smoothing filter of size 5×5

$$g_1(2,2) = \frac{1}{9}[3+1+2+2+7+5+0+6+5] = \frac{31}{9} = 3.44 \approx 3$$

$$g_2(2,2) = \frac{1}{25}[0+1+0+6+5+2+3+1+2+5+1+2+7+5+4+1+0+6+5+$$
$$2+2+3+5+5+7+6] = \frac{81}{9} = 9$$

b) 3×3 weighted average mask is

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

Thus for weighted average filter,

$$g_3(2,2) = \frac{1}{16}[3\times1 + 1\times2 + 2\times1 + 2\times2 + 7\times4 + 5\times2 + 0\times1 + 2\times6 + 5\times1]$$

$$= \frac{66}{16} = 4.125 \approx 4$$

c) Median filter of 3×3, sort the values in ascending order

$$g_4(2,2) = \text{median}\{0,1,2,2,3,5,5,6,7\} = 3$$

Median filter of size 5×5, sort the values in ascending order

$$g_5(2,2) = \text{median}[0,0,0,1,1,1,1,2,2,2,2,2,3,3,4,5,5,5,5,5,6,6,6,7,7] = 3$$

d) For min filter of size 3×3

$$g_6(2,2) = 0$$
$$g_7(2,2) = 0$$
$$g_8(2,2) = 7$$

For min filter of size 5×5

e) For max filter of size 3×3 and 5×5

***

Try these exercises.

E5) What is a Median filter?

E6) What is maximum filter and minimum filter?

E7) Differentiate linear spatial filter and non-linear spatial filter.

Now, in the following section, we shall discuss sharpening spatial filters.

7.4 IMAGE SHARPENING

Image sharpening is opposite of image smoothing. This is done to highlight fine details and edges in an image. Applications of image sharpening are outlining object boundaries in industrial applications, identifying tumor boundaries, or bone structures in medical imaging etc. Smoothing is achieved by pixel averaging which is analogous to integration. Image sharpening is just the reverse process, and is achieved by differentiation. The derivative operation enhances the degree of discontinuity in an image.

The advantages of image sharpening are the following.

- It enhances edges and other discontinuities (noise) in an image.

- It de-emphasizes area with slowly varying grey levels (background) in an image. First order derivative of a one-dimensional signal f(x) is defined as

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \tag{8}$$

Similarly 2nd order derivative $f(x, y)$ is given by

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \tag{9}$$

Example 5: Find first and second derivative of given data $f(x): f(x) = [4 \quad 3 \quad 2 \quad 5 \quad 9]$.

Solution: $\frac{\partial f}{\partial x} = [3-4, 2-3, 5-2, 9-5]$

$= [-1, -1, -3, 4]$

$\frac{\partial^2 f}{\partial^2 x} = [4+2-6, 3+5-4, 9+2-10]$

$= [0, 4, 1]$

***

With respect to the first and second derivatives, following things are observed

i) In flat regions (region of constant intensity levels) both derivatives produce zero.

ii) In ramp (constant slope), first derivatives gives a non-zero value, whereas second derivative gives non zero values (with different sign) only at the beginning and end of ramp and zero along the ramp. First derivative produces a thick edge called '**double edge effect**' along the ramp as the output is non zero. Second derivative produces a double edge one pixel thick separated by zeros.

iii) For a noisy pixel, first derivative produces a positive or negative value leading to one zero crossing. Whereas second derivative produces two zero crossing making it easier to identify.

iv) In case of an edge (step change in intensity values), first derivative produces a sign change leading to a zero crossing. This makes edge detection easier by second derivative. Second derivative enhances the details and edges much better as compared to first derivative. Thus for image sharpening applications, second derivative is most suitable.

Here, we will discuss some sharpening filters base on first derivatives and second derivatives.

7.4.1 First Derivatives Filters

Gradient Operator

An edge is the boundary between two regions with distinct grey level properties. Derivative operators are used for most edge detection techniques.

The magnitude of first derivative calculated within a neighborhood around the pixel of interest is used to detect presence of edge in an image. First derivatives are directional operators and is defined as

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \tag{10}$$

Magnitude of this vector is given by

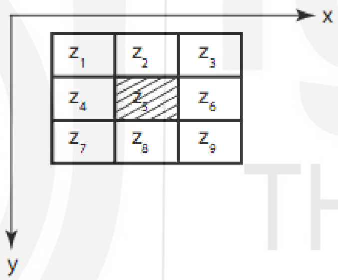
$$\nabla f = \mathrm{mag}(\nabla f(x, y)) = \sqrt{g_x^2 + g_y^2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right] \tag{11}$$

In common practice, gradient with absolute values are simpler to implement. Thus

$$\nabla f = |g_x| + |g_y| = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| \tag{12}$$

Robert Operator

Consider the a pixel of interest $f(x, y) = z_5$ and a rectangular neighborhood of size 3 × 3 in Fig. 12 (a).



$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline -1 & 0 \\ \hline \end{array} \qquad \begin{array}{|c|c|} \hline 1 & -1 \\ \hline 0 & 0 \\ \hline \end{array}$$

(a) Robert Operator **(b)**

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \qquad \begin{array}{|c|c|} \hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$$

(c)

Fig. 12

Equation can be approximated at point z_5 in a number of ways. Simplest is

$$g_x = z_5 - z_8 \tag{13}$$

$$g_y = z_5 - z_6 \tag{14}$$

$$\nabla f = |z_5 - z_8| + |z_5 - z_6| \tag{15}$$

Another approach is to use cross difference to realized

$$\nabla f \cong |z_5 - z_9| + |z_6 - z_8| \tag{16}$$

Equations can be implemented by the masks in Fig. 12 (a) and Fig. 12 (c). The original image is convolved with both the masks separately and the absolute values of the two outputs of convolution are added.

Although Roberts operator illustrates the derivative process, there is a difficulty in implementation. Since only two neighboring pixels are used, it is not clear as to exactly where the derivative value should be stored in the output image. This difficulty is taken care of in other filters like Prewitt and Sobel filters which are symmetric in nature.

Prewith Operator

In the Prewitt Operator, the edge value at $f(x, y)$ is calculated by taking the - difference of $f(x+1, y)$ and $f(x-1, y)$. To make sure that the difference is not coming from two arbitrary points, three pairs of points are taken together to result in one edge value. This creates a 3×3 mask.

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Fig. 13: Perwitt Operator

The above mask in Fig. 13 can be used to identify the presence of a vertical edge in the following image:

30	30	30	80	80	80
30	30	30	80	80	80
30	30	30	80	80	80
30	30	30	80	80	80
30	30	30	80	80	80
30	30	30	80	80	80

$$\nabla f \cong |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$

Sobel Operator

Another sharpening first order filter is the Sobel Operator. It is slightly different from the Prewitt filter, in that the middle pair of pixels are given higher weight compared to other two pairs of pixels. Both the filters are also called EDGE FILTERS. Sobel is the most popular 3×3 Edge operator. It is described by

$$\Delta f = |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

Sobel mask is shown in Fig. 14. Notice that all mask coefficients sum up to zero, thus giving no response in the area of constant intensity.

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Fig. 14: Sobel Operator

Sobel operator and Prewitt operator are similar to each other. Sobel operator is also used to detect two kinds of edges in an image; vertical and horizontal.

You can observe from the masks given in Fig. 14, that first masks has only one difference in the members of first and third rows that is of sign, that is it has “2” and “–2” values in center of first and third rows. When applied on an image this mask will highlight the horizontal edges. Similarly, the second mask in Fig. 14 shows the vertical edges.

When we apply this mask on the image it prominent vertical edges. It simply works like as first order derivative and calculates the difference of pixel intensities in an edge region.

As the center column is of zero so it does not include the original values of an image but rather it calculates the difference of right

Now, let us discuss the filters based on second derivatives.

7.4.2 Laplacian Operator

For a 2D function $f(x, y)$, the gradient (first derivative) is defined as

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

Laplacian (second derivative) is a **rotation invariant** and **linear** operator and it is defined as

$$\Delta^2 f = \frac{\partial^2 f(x,y)}{\partial x^2 \partial y^2} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial f^2}{\partial y^2} \tag{17}$$

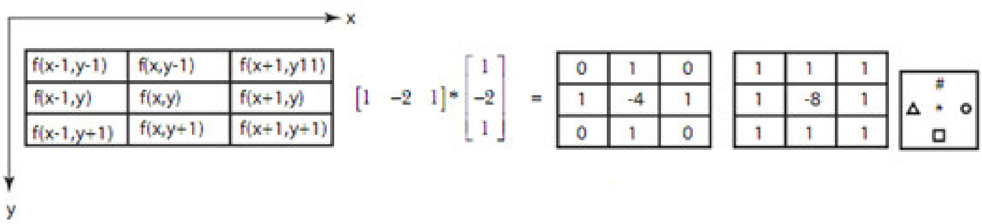
$$\frac{\partial^2 f}{\partial x^2} = f(x-1,y) + 2f(x,y) + f(x+1,y) = [1 \quad -2 \quad 1]$$

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

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

$$\Delta^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

O Δ □ # *



(a) (b) (c) (d)

$$\begin{array}{|rrr|} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{array} \qquad \begin{array}{|rrr|} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{array} \qquad \begin{array}{|rrr|} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{array}$$

(e) **(f)** **(g)**

Fig. 15: Masks for laplacian operator

Fig. 15 (b) shows a 3×3 sharpening filter mask. Fig. 15 (a) shows the coordinate notations and Fig. 15 (d) shows the location of non zero elements. Notice that **addition of all the coefficients of the mask produces zero**. In this mask, diagonal elements are not included. Fig. 15 (c) shows a mask with diagonal elements also. Fig. 15 (e) shows a weighted Laplacian mask. Notice that the centre element is negative. Fig. 15 (f) and Fig. 15 (g) show two Laplacian masks with centre element positive which gives the same results. Laplacian operation is a derivative operation which highlights the intensity discontinuities of an image and de-emphasize region with slowly varying intensity values. This tends to produce image that have greyish edge lines with dark featureless background. (Fig.16). Background features can be '**recovered**' while '**preserving**' the sharpened effect of Laplacian by subtracting Laplacian from the image. Thus, sharpened imaged is generated by the following operation.

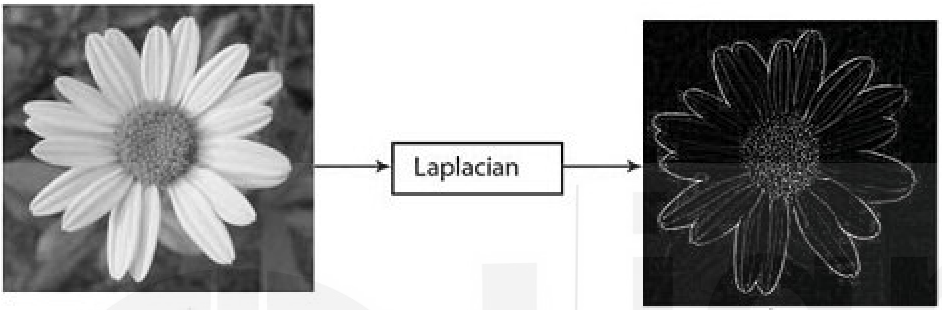


Fig 16: Output after laplacian operator

$$\begin{aligned} g(x,y) &= f(x,y) - \Delta^2 f(x,y) \\ &= f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)] \\ &= 5f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1) \end{aligned}$$

A mask is generated for sharpening of images.

$$\begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Similarly

$$\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & -8 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

These are high frequency enhancement filters that **pass all frequencies and enhances high frequencies** . Notice that all the coefficients in the laplacian mask add up to one, so convolution with this mask will not change the pixel values in the part of the image that has constant grey levels. But if the pixels have higher or low grey level values than its neighbours, this contrast will be enlarged by the filter.

Example 5: For the image segment $f(x,y)$ given below. Apply laplacian filter of size 3×3 and 5×5 on the centre pixel.

Solution: Laplacian 4-connectivity mask is

0	1	0
1	−4	1
0	1	0

Thus, for Laplacian filter,

$$g_9(2,2) = [1+2+5+6-4\times 7] = -14$$

Laplacian 8-connectivity mask is

1	1	1
1	−8	1
1	1	1

Thus for Laplacian filter

$$g_{10}(2,2) = [3+1+2+2+5+0+6+5-8\times 7] = -22$$

***

Try these exercises.

E8) How are various filter mask generated to sharpen images in spatial filtering?

E9) Explain spatial filtering in image enhancement.

E10) What is meant by Laplacian filter? Using the second derivative, develop a Laplacian mask for image sharpening

E11) What are image sharpening filters? Explain the various types of q sharpening filters.

E12) Name the different types of derivative filters.

E13) Suggest typical derivative masks for Image enhancement. a) Roberts b) Prewitt c) Sobel.

E14) Write the applications of sharpening filters.

Now let us, summarise what we have discussed in this unit.

7.8 SUMMARY

In this unit, we have discussed the following points.

1) Filtering is a process that removes some unwanted components or small details in an image. In digital image processing, filter is basically a subimage or a mask or kernel or template or window. Filters can be of two types: Spatial filters and frequency domain filters.

2) A filter that passes low frequencies is called a lowpass filter. This type of filter is used to blur (smooth) the image, therefore called smoothing filter also. These filters are also used for noise reduction. Noise reduction can be done by blurring with a linear filter, (in which operation performed on the image pixels is linear) or a non-linear filter.

3) This filter is used to highlight transitions in intensity. These are based on first and second derivatives.

4) Linear filtering is a spatial domain process where a filter (mask/ kernel/ template) with some integer coefficient values is applied to input image to generate the filtered/ output image.

5) Each pixel in the smoothened image $g(x, y)$ is obtained from the average pixel value in the neighborhood of (x, y) in input image. Such a mask is also known as **a Mean filter.**

6) Like all other spatial filters, non linear filters compute the result at some position (x, y) from the pixels inside the moving region S of the original image. These filters are called **non-linear** because source pixels are processed by some non-linear function.

7) Median filters are edge preserving smoothing filters, where the level is set to the median of pixel values in the neighborhood of that pixel.

8) **Min filter** removes salt noise (white dots with large grey level values) because any large grey level with in a 3×3 filter region is replaced by one of its surrounding pixels with smallest value. As a side effect, min filter introduces dark structures in the image. The reverse effect is expected from a **max filter**. It removes pepper noise (black dots with small grey level values) because any black dot within 3×3 filter region is replaced by one of its surrounding pixels with the largest value. White dots/bright structures are widened as a side effect and black dots (pepper noise) will disappear.

9) Image sharpening is opposite of image smoothing. This is done to highlight fine details and edges in an image.

7.9 SOLUTIONS/ANSWERS

E1) Linear filtering is a spatial domain process where a filter (mask/kernel/template/window) with some coefficient values is applied to input image to generate the filtered/ output image. Generally, filter size is either 3×3 or $5\times5, 7\times7$ or 21×21 (odd sizes) and filter is centered at a coordinate (x, y) called ‘ Hot Spot’.

E2) Smoothing linear filters are used to reduce ‘sharp’ transitions in grey levels, reduce noise, bluredges, help in smoothing false colours, reduce ‘irrelevant’ details in an image.

Sharpening linear filters are used to detect edges, lines, points in the image. It is also used to highlight all high frequency components in the image.

E3) Each pixel in the smoothened image $g(x, y)$ is obtained from the average pixel value in the neighborhood of (x, y) in input image.

$$g(x, y) = \frac{1}{M} \sum_{(m,n)\in S_{xy}} f(m, n) w(m, n)$$

Various masks are $\frac{1}{9}\times\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}\frac{1}{16}\times\begin{bmatrix}1 & 2 & 1\\2 & 4 & 2\\1 & 2 & 1\end{bmatrix}$

Following steps are required for spatial averaging

1) Position the mask over the current pixel such that hotspot w(0,0) coincides with current pixel.
2) Form all products of filter elements with the corresponding elements in the neighbourhood.
3) Add up all the products and store it at current position in the output image. This must be repeated for every pixel in the image.

E4) The aim of image smoothing is to diminish the effect of camera noise, spurious pixel values, missing pixel values etc.

E5) Median filters are edge preserving smoothing filters, where the level is set to the median of pixel values in the neighborhood of that pixel. It is impossible to design a filter that removes only noise and retains all the important image structures intact, because no filter can discriminate which image content is important to the viewer and which is not. Median filter replaces every image pixel by median of the pixels in the corresponding filter region Sxy.

$$g(x,y) = \text{median}\ \{f(x+i, y+j) \mid (i,j) \in Sxy\}$$

E6) Max and min filters are defined as:

$$g(x,y) = \min\{f(x+i, y+j) \mid (i,j) \in S_{xy}\}$$
$$g(x,y) = \max\{f(x+i, y+j) \mid (i,j) \in S_{xy}\}$$

where S_{xy} denotes the filter region, usually a size of 3×3 pixels.

E7) Linear filters produce an output that is a linear combination of the input. For example, smoothing linear filters produce output as the average of input while sharpening linear filters produce output as the first or second derivative of the input.

Non linear filters produce a statistical output where median filter produce output as median of the input values, min filter produces output as minimum of all input values and max filter produces output that is maximum of all input values.

E8) $\Delta^2 f = \frac{\partial^2 f(x,y)}{\partial x^2 \partial y^2} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial f^2}{\partial y^2}$

E9)

$$\begin{array}{|ccc|}\hline -1 & 2 & -1\\ 2 & -4 & 2\\ -1 & 2 & -1\\ \hline\end{array}\qquad\begin{array}{|ccc|}\hline 0 & -1 & 0\\ -1 & 4 & -1\\ 0 & -1 & 0\\ \hline\end{array}\qquad\begin{array}{|ccc|}\hline 1 & 1 & 1\\ 1 & -8 & 1\\ 1 & 1 & 1\\ \hline\end{array}$$

Spatial filtering is one of the main tools used in variety of applications such as noise removal, bridging the gaps in object boundaries, sharpening of edges etc.The idea of spatial filtering is to move a 'mask', a rectangle (usually with size of odd length) or other shape over the image. By this process, we create a new image where grey level values of the pixels are calculated from the values under the mask. The values under the mask are modified by a function called 'filter'. If this filter function is a linear function of all grey level values in the mask, then filter is called a 'linear filter', else it is called 'non linear filter'.

E10) **Laplacian** (second derivative) is a **rotation invariant** and **linear** operator and it is defined as

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

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

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

Thus

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

O Δ □ # *

$$\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

E11) Image sharpening is done to highlight fine details and edges in an image. Applications of image sharpening are industrial applications, medical imaging etc. Sharpening is reverse of smoothing which is achieved by pixel averaging which is analogous to integration. Thus, sharpening is achieved by differentiation. The derivative operation enhances the degree of discontinuity in an image.

Sharpening

- Enhances edges and other discontinuities (noise) in an image.
- De-emphasizes area with slowly varying grey levels (background) in an image.

Different image sharpening filters are Laplacian, gradient, Robert, sobel, etc.

E12) For a 2D function $f(x,y)$, the gradient (first derivative) is defined as **Laplacian** (second derivative) is a **rotation invariant** and **linear** operator and it is defined as

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

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

E13)

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Sobel operator

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Prewitt operator

$$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline -1 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|}\hline 1 & -1 \\ \hline 0 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \quad \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$$

Robert operator

E14) Sharpening filters are used for edge enhancement applications. Sharpening linear filters are used to detect edges, lines, points in the image. It is also used to highlight all high frequency components in the image.