Image enhancement Unit 2

The objective of the image enhancement is to increase the quality of the image so that it is more suitable for specific application.

It enhances the features, boundaries, edges, contrast

It can be done by 2 methods.

1. Spatial domain method → Manipulation done directly on pixels
2. Frequency domain method → Manipulation done on Fourier transformed image.
3. Combination method → Combination of first 2 methods.

Spatial domain → The term spatial domain refers to image plane itself. Image with pixel values is nothing but spatial domain

→ Image processing in this is direct manipulation of pixels

→ 2 categories 1. Intensity Transformation – in which modification of intensity will takes place

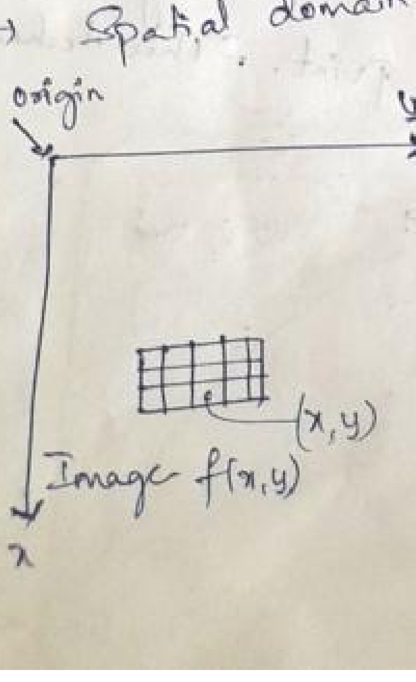
2. Spatial filtering – direct manipulation of pixels using a mask

→ Spatial domain process is denoted by $g(x,y) = T\{f(x,y)\}$

T – operator.

$f(x,y)$ – input image $g(x,y)$ → o/p image

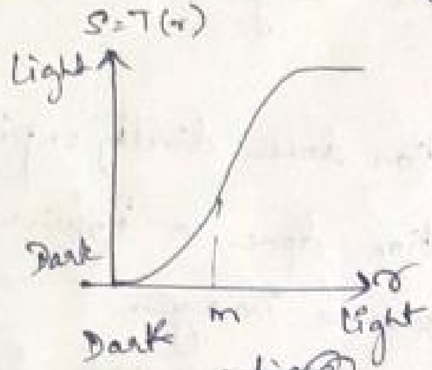
point (x,y) is the arbitrary location in an image



The simplest for T is when the neighbourhood is of the size 1X1

T → gray level transformation / Intensity / Mapping.

$S = T(r)$ S → o/p image pixel value
r → i/p



fig(a)

Contrast Stretching

Gray level transformation.

If T(r) has the form shown in fig(a) the effect of this tranformation is would be to produce an image of higher contrast that the original by darkening the levels below m in brightening the levels above m in the original image.

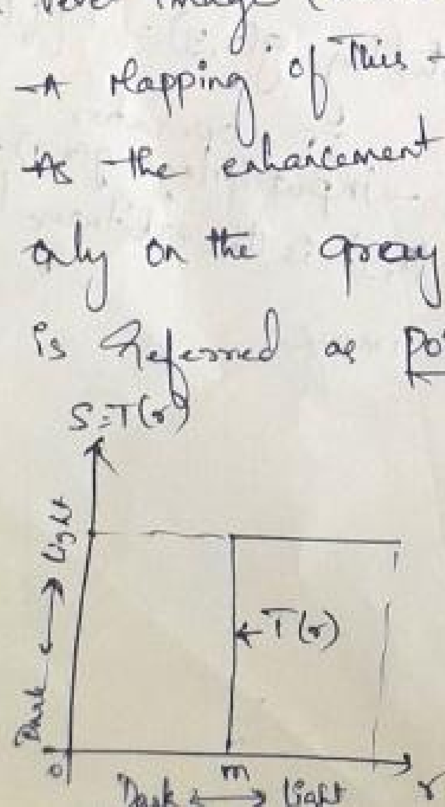
This technique is known as Contrast Stretching.

The value of r below m are compressed by the transformation function to the narrow range of S. towards black.

Fig(b) show is a gray level transformation. which produces 2 level image (also known as binary image).

A mapping of this form is known as threshold function.

As the enhancement at any point in an image depends only on the gray level at that point. This technique is referred as point processing.



* If we consider the image larg so of neighborhood results in more flexibility and hence one principal approach is to use the mask / filter / kernel / template / window.

Mask is a small 2D array [3×3]. The values of the mask co-efficients determine the nature of the process. Enhancement technique based on this approach is referred as mask processing / filtering.

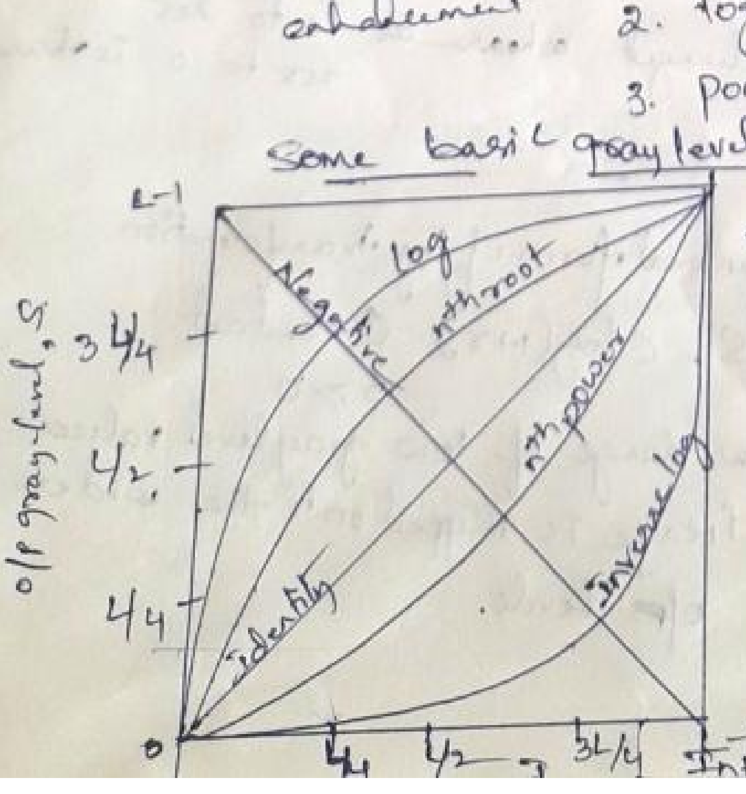
Basic Intensity transformation function

1. Gray level transformation / Intensity transformation.

Can be represented as $S = T(r)$ — $r \to$ value of pixel before processing, $s \to$ value of pixel after processing, T – transformation maps r with S

* Since we are dealing with digital quantities,
* value of Transformation func is stored in an array.
* Mapping from r to S is implemented via r to S.
* Three basic type of transformations are used in Image Gray levels ~~Transformation~~ enhancement
  1. Linear (Negative & Identity transform)
  2. Logarithmic (log & Inverse ~~Antilog~~ transformation)
  3. Power law (n^{th} power & n^{th} root transformation)

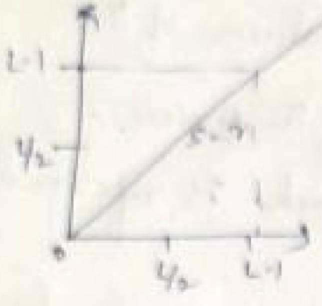
Some basic gray level transformation functions



1. <u>Identity transformation</u>

The image with intensity level $[0, L-1]$ is given, $S = R$

ie after transformation no change in pixel value.



$$\begin{matrix} 0 & 10 & 50 \\ 5 & 95 & 100 \\ 110 & 150 & 200 \end{matrix} \xrightarrow{s=r} \begin{matrix} 0 & 10 & 50 \\ 5 & 95 & 100 \\ 110 & 150 & 200 \end{matrix}$$

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

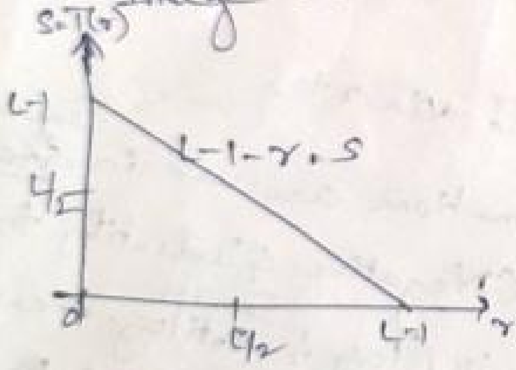
2. <u>Negative transformation</u>

The -ve of the digital image is obtained after the transformation $S = T(r) = L-1-r$ $\quad$ $L \rightarrow$ no of Gray levels

Reversing the intensity levels of an image is done here to produce the equivalent photographic -ve

Highest Gray levels is mapped to lowest Gray level & vice-versa.

Image -ve is used in displaying medical image



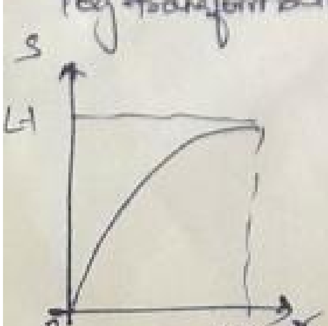
$L = 256$

$$\begin{bmatrix} 0 & 10 & 50 & 100 \\ 5 & 95 & 100 & 200 \\ 110 & 150 & 190 & 210 \\ 175 & 210 & 255 & 100 \\ 0 & 255 & 0 & 255 \end{bmatrix} \xrightarrow[= 255-r]{s = L-1-r} \begin{bmatrix} 255 & 245 & 205 & 155 \\ 250 & 160 & 105 & 55 \\ 145 & 105 & 165 & 45 \\ 80 & 45 & 0 & 155 \\ 255 & 0 & 255 & 0 \end{bmatrix}$$

Middle gray levels are not changed whereas 0 to 255, 255 to 0 is observed

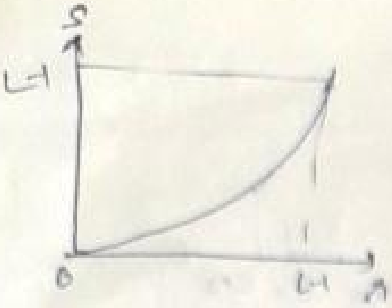
<u>Logarithmic function</u>

log transformation



The general form of log transformation is $S = c\log(1+r)$ $\quad$ C - Constant, $r \geq 0$

The narrow range of low gray level values in the i/p image is mapped onto the wider range of o/p levels.



* the shape of the log curve in fig shows that transformation maps a narrow range of low gray level values in the range of i/p range into wider range of o/p level

Inverse log transformation * The opposite is true ie wide range of i/p is converted to narrow range of o/p.

This transformation is used to expand a value of dark pixels in a imag while compressing the higher level values. The opposite is true in Inverse log transformation.

* The amount of expansion or compression to be done is fixed & cannot be changed. More flexibility is given by <u>Power law transformation</u>.

$$\begin{bmatrix} 0 & 10 & 50 & 100 \\ 5 & 95 & 150 & 200 \\ 110 & 150 & 190 & 210 \\ 125 & 210 & 255 & 100 \end{bmatrix} \xrightarrow{S = c\log(1+r)} \begin{bmatrix} 0 & 110 & 181 & 212 \\ 82 & 210 & 231 & 244 \\ 217 & 231 & 242 & 246 \\ 238 & 246 & 255 & 212 \end{bmatrix}$$

If $C=1$ & $r=0 \rightarrow S=0$

$C=1$ $\quad r=10 \Rightarrow S=1.04$

$C=1$ $\quad r=20 \Rightarrow S=1.32$

$C=1$ $\quad r=200 \Rightarrow S=2.302$

$C=1$ $\quad r=255 \Rightarrow S=2.4$

I/p max value is 255 after log max $= 2.4$ thus scaling of $\frac{255}{2.4} = 1.06$

Thus the max value becomes 255

$2.4 \times 106 = 255$ $\qquad 2.302 \times 106 = \underline{244}$

Power law function -

N^{th} root transformation

N^{th} power transformation.

Power law transformation have the basic form $s = c r^{\gamma}$
where c & γ are +ve constant

→ For fractional values of γ (r/s) the transformation behaves like a log transfer i.e. Narrow range of dark i/p values is mapped into wider range of o/p values, with the opp being true for higher value of n bit levels.

→ for $\gamma > 1$ the transformation behaves like inverse log transform. It has a opposite effect when $\gamma < 1$

→ for $c = \gamma = 1$ power law transformation reduces to Identity transformation.

→ Here power law transformation Range compression & expansion is observed

for $\gamma = 25$ $r = [1 \text{ to } 100]$ $s = [0 \text{ to } 25]$ Range compression
$r = [210 \text{ to } 255]$ $s = [155 \text{ to } 255]$ Range expansion

for $\gamma = 0.4$ reverse is observed.
$r = [1 \text{ to } 100]$ $s\ [25 \text{ to } 175]$ Range expansion
$r = [210 \text{ to } 255]$ $s\ [236 \text{ to } 255]$ Range Compression

Gamma Correction - Application of power law transformation
The exponent used in power law is γ. The process used to correct this power law response phenomenon is called Gamma Correction.

If the value of $\gamma = 2.5$ then such display s/y would tend to produce an image that are darker than the original image. So to avoid this processing of i/p image done ($1/2.5 = 0.4$) before giving to Monitor.

Then they produce o/p that is close to original
Similar analysis would apply to another imaging devices
such as printer scanners.

for CRT $\gamma = 2.5$

0	1	2	$\gamma = 1/2.5 = 0.4$	0	1	1.3	$\xrightarrow{\gamma = 2.5}$	0	1	2
100	0	160		3.1	0	7.6		100	0	160
0	127	255	Pixel mapping	0	6.94	9.17		0	125	255

$f(x,y)$ $\quad$ $g_1(x,y)$ $\quad$ $g_2(x,y)$.

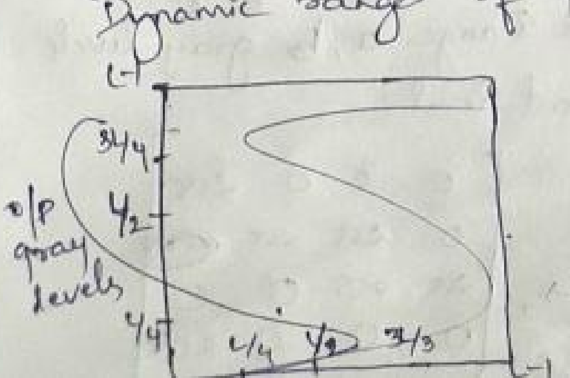
Piece wise linear transformation.

In Gray level transformation, transformation is applied to whole image. If we required to enhance particular part of image, we generally refer to piece wise linear transformation.

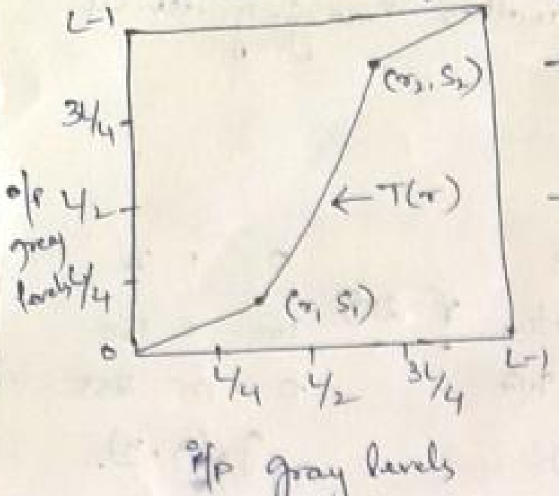
3 types
1. Contrast stretching.
2. Gray level slicing (Intensity level slicing)
3. Bit plane slicing.

1. Contrast stretching – low contrast images occur due to bad & non uniform illumination Cdn due to non-linearity of image acquisition devices, from poor illumination, lack of dynamic range in the imaging sensor.

The idea behind contrast stretching is to increase the Dynamic range of gray levels in the image being processed



Contrast Stretching transformation function



→ This is a typical transformation used for Contrast Stretching.

→ The location of the points (r_1, s_1) (r_2, s_2) Controls the shape of the transformation function

→ If $r_1 = s_1$ $r_2 = s_2$ transformation is a linear function & produces no change in the gray levels

→ If $r_1 = r_2$ $s_1 = 0$ $s_2 = L-1$ → the transformation becomes a thresholding function that creates a binary image ~~binary image~~

Perform thresholding of $f(x,y)$ with $t = 128$

0	10	50	100
5	95	150	200
170	150	190	210

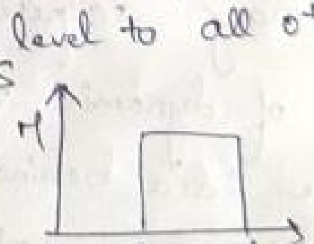
$t = 125$

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

Gray level slicing -

1. It is used to highlight the specific range of gray levels in an image.
2. One approach is to display a high value for all gray levels in the range of interest & low level to all other gray level.

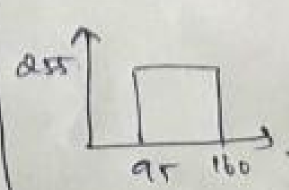
$$s = \begin{cases} M & A \le r \le B \\ 0 & \text{otherwise} \end{cases} \Rightarrow$$



8.

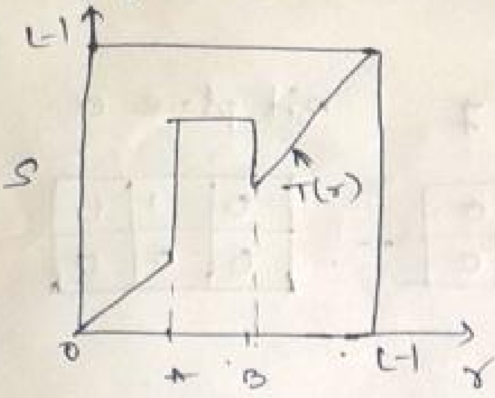
3. This transformation highlights range a, b gray levels and produces all other contrast level

Eg
$$\begin{bmatrix} 0 & 10 & 50 & 100 \\ 5 & 95 & 150 & 200 \\ 110 & 150 & 190 & 210 \\ 175 & 210 & 225 & 150 \end{bmatrix}$$



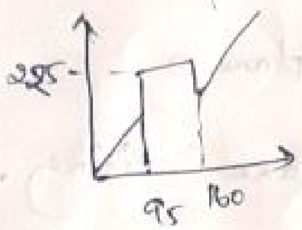
$$\begin{bmatrix} 0 & 0 & 0 & 255 \\ 0 & 255 & 255 & 255 \\ 255 & 255 & 0 & 0 \\ 0 & 0 & 0 & 255 \end{bmatrix}$$

4. The Second approach is based on transformation shown, the fig below. This transformation brightens the desired range of gray levels but preserves the background & graylevel tonalities in the image.



$$S = \begin{cases} H & A \le r \le B \\ r & \text{otherwise} \end{cases}$$

0	10	150	100
5	95	150	200
10	150	190	210
175	210	225	150



0	10	225	225
5	225	225	200
225	225	190	210
175	210	225	225

Bit-plane slicing –

Sometimes it is required to know the Contribution of each bit of the image.

Generally Gray scale image Consists of 8 bits for pixel Representation ie, each pixel in an image is represented by 8 bit. then the pixel Can be divided to 8 planes (Plane 0 – plane 7).

Plane 0 will have LSB's & plane 7 will have MSB's

Seperating the digital image into its 8 bit plane. It is useful for analysing the relative importance played by each bit in an image.

Using this approach One can understand the min no of bits required to represent the image with desired details. & also its decides the size of image required to represent & store the image.

The higher order bits (Bit plane 4,5,6,7) Contains Majority of info. The lower order bits (bit plane 0,1,2,3) Contains

Compute the bit plane slicing for 8bit image shown.

①.

6	7	8
3	2	10

6 - 0 0 0 0 0 1 1 0

7 - 0 0 0 0 0 1 1 1

8 - 0 0 0 0 1 0 0 0

3 - 0 0 0 0 0 1 1 0

2 - 0 0 0 0 0 0 1 0

10 - 0 0 0 0 1 0 1 0

↑ Bitplane 7. ↑ Bitplane 0

Bitplane 7

0	0	0
0	0	0

Bit plane 0

0	1	0
0	0	0

② Compute the bit plane slicing for 8bit image

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

0 0 0 0 0 0 0 0 0

10 0 0 0 0 1 0 1 0

50 0 0 1 1 0 0 1 0

100 0 1 1 0 0 1 0 0

50 0 0 1 1 0 0 1 0

95 0

150 1

200 1 1 0 0 1 0 0 0

110

150

190

210

175

210

255

110

7th plane

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

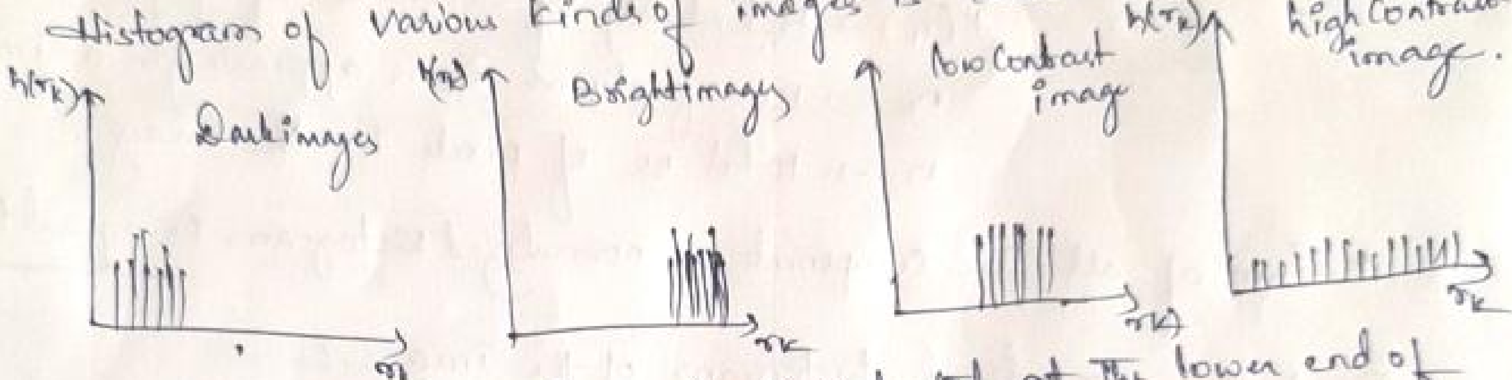
Bit 7.

Histogram processing.

Histogram of an image Represents the no of times a particular gray level has occured in an image.

Histogram is a graphical Representation of any data. Histogram in image processing is a graphical Representation of digital image. It Represents the relative frequency of occurance of various gray levels.

Histogram of various kinds of images is shown below.



For Dark images - histogram is Concentrated at the lower end of the gray scale

For Bright images histogram is Concentrated at the right side

For low Contrast images it is narrow & Centered towards the middle.

For High Contrast images it Covers the broad range of gray scale with flat profile.

* The Histogram with grayleveh range $[0, L-1]$ is represented as a discrete function $h(r_k) = n_k$ $r_k \rightarrow k^{th}$ gray level

$n_k \rightarrow$ no of times r_k is appearing in an image.

*

Histogram is used to manipulate Contrast & brightness.
Histogram is used to manipulate Contrast & brightness.

A good quality image will have a flat profile in the histogram

~~the~~ So this can be achieved by <u>Normalization</u>..

Normalization histogram is obtained by dividing the no of times the occurance of a ~~pix~~ each pixel by total no of pixel in the image.

$P(r_k) = \frac{n_k}{n}$ $k = 0 \ldots L-1$

$P(r_k) \rightarrow$ Probability of occurance of graylevel r_k

$n_k \rightarrow$ no of times r_k is appearing in an image

$n \rightarrow$ total no of pixels in the image.

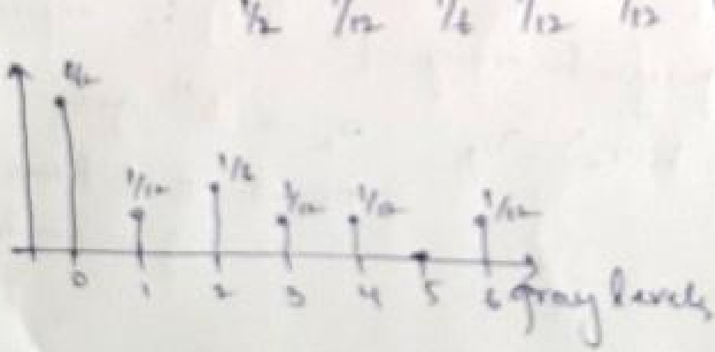
* Sum of all the components of normalized histogram is <u>equal to 1</u>.

Find the normalized histogram of the image!

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

$n = r \times c = 3 \times 4 = \underline{12}$.

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



<u>Histogram equalization</u> or <u>Histogram Linearization</u>

* It is an point operation that maps an i/p image onto o/p image such that there are equal no of pixels at graylevel in o/p.
* It is used for Contrast enhancement

Histogram is used to manipulate Contrast & brightness.

- A good quality image will have a flat profile in the histogram
- So this can be achieved by <u>Normalization</u>..

Normalization histogram is obtained by dividing the no of times the occurance of a ~~pix~~ each pixel by total no of pixel in the image.

$$P(r_k) = \frac{n_k}{n} \qquad k = 0 \ldots L-1$$

$P(r_k) \rightarrow$ Probability of occurance of graylevel r_k

$n_k \rightarrow$ no of times r_k is appearing in an image

$n \rightarrow$ total no of pixels in the image.

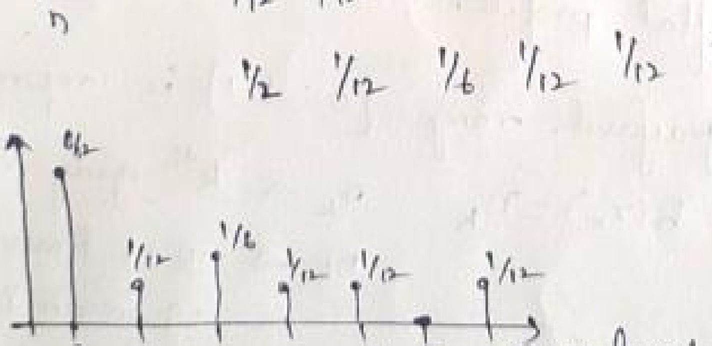
* Sum of all the components of normalized histogram is equal to 1.

Find the normalized histogram of the image?

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

$n = r \times c = 3 \times 4 = 12.$

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



<u>Histogram equalization</u> or <u>Histogram Linearization</u>

* It is an point operation that maps an i/p image onto o/p image such that there are equal no of pixels at graylevel in o/p.
* It is used for Contrast enhancement

1. Perform histogram equalization for 8×8 image shown.

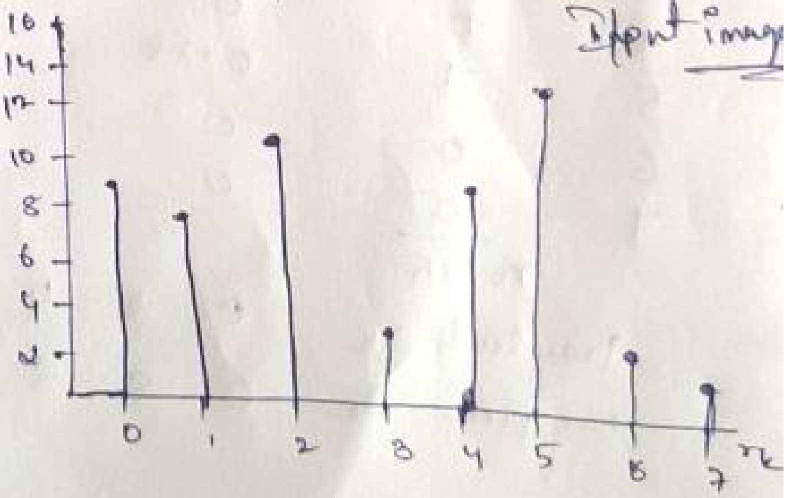
Gray levels	0	1	2	3	4	5	6	7
No of pixels	9	8	11	4	10	15	4	3

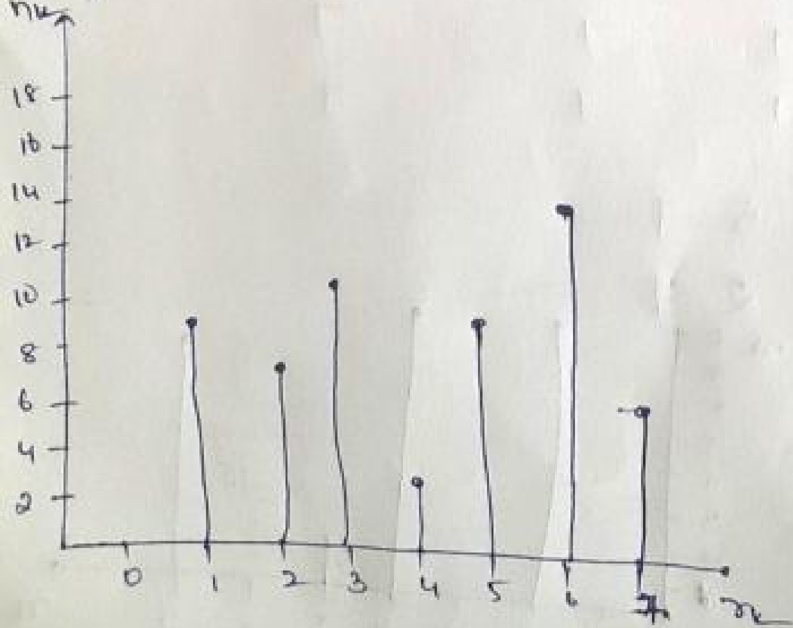
Soln

Gray levels r_k	no of pixels n_k	$P_r(r_k) = \frac{n_k}{n}$	s_k CDF	$S \times 7$	Histogram equalization
0	9	0.141	0.141	0.987	1
1	8	0.125	0.260	1.862	2
2	11	0.172	0.438	3.066	3
3	4	0.0625	0.5005	3.5035	4
4	10	0.156	0.6505	4.5955	5
5	15	0.234	0.8905	6.2336	6
6	4	0.0625	0.953	6.671	7
7	3	0.047	1	7	7
	n = 64				

graylevels	1	2	3	4	5	6	7
no of pixels	9	8	11	4	10	15	7

Input image





2. Perform the histogram equalization for the following image.

$$f(x,y) = \begin{bmatrix} 1 & 2 & 1 & 1 & 1 \\ 2 & 5 & 3 & 5 & 2 \\ 2 & 5 & 5 & 5 & 2 \\ 2 & 5 & 3 & 5 & 2 \\ 1 & 1 & 1 & 2 & 1 \end{bmatrix}$$

<u>Soln</u>: As <u>no</u> of graylevels is not given directly consider the <u>max</u> gray level = 5 $2^3 = 8$ ✓ ∴ $L = 8$.

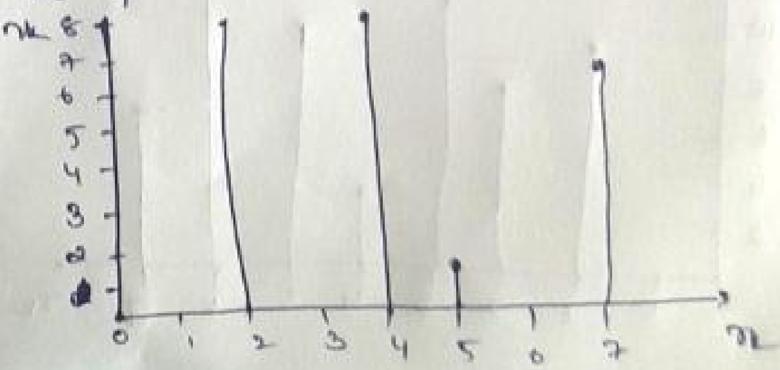
$2^2 = 4$ ✗

∴ the no of gray levels = 8   Range $[0 \text{ to } L-1] \Rightarrow \underline{0 \text{ to } 7}$.

graylevels r_k	0	1	2	3	4	5	6	7
no of pixels n_k	0	8	8	2	0	7	0	0

gray level r_k	no of pixels n_k	$P(r_k) = \frac{n_k}{n}$	S_k	$S_k \times 7$	Histogram equalization
			0	0	0
0	0	0	0.32	2.24	2
1	8	0.32	0.64	4.48	4
2	8	0.32	0.72	5.04	5
3	2	0.08	0.72	5.04	5
4	0	0	1	7	7
5	7	0.28	1	7	7
6	0	0	1	7	7
7	0	0			
	$n = 25$				

gray levels r_k	0	2	4	5	7
no of pixels n_k	0	8	8	2	7

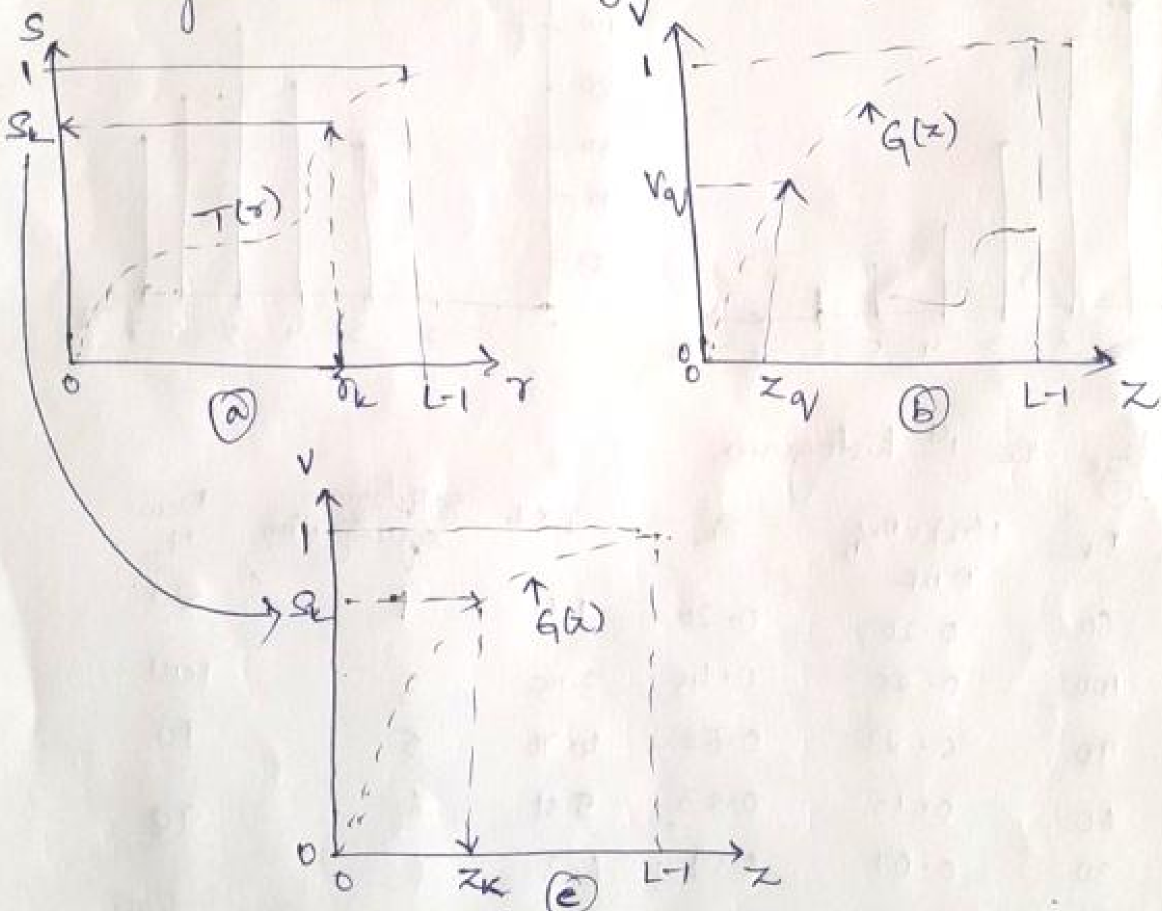


Histogram Matching / Histogram specification.

Histogram Equalization is not applicable for some application. As it Generates histogram of the entire image.
If we want to enhance a specific part of an image we go with histogram matching.

Histogram matching is a method to generate a processed image that has a specified histogram.

The diagram shows how histogram matching is done.



a). Graphical interpretation of mapping from r_k to s_k via $T(r)$.

b). Mapping of z_q to its corresponding value v_q via $G(z)$.

c) Inverse mapping from s_k to its corresponding value of z_k.

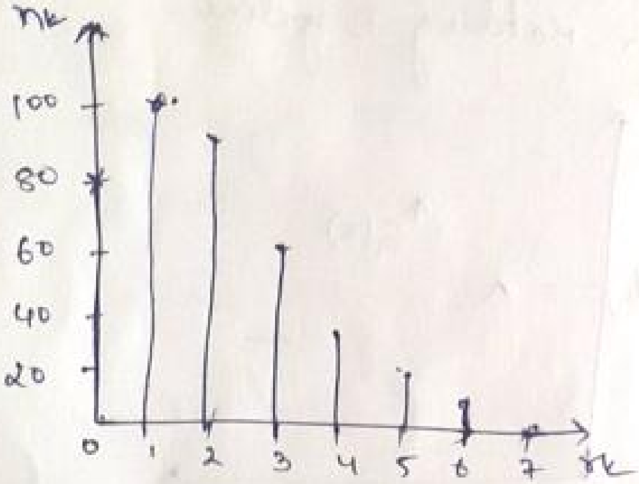
Given below are 2 histograms ① & ② Modify the the histogram (1) as given by histogram 2

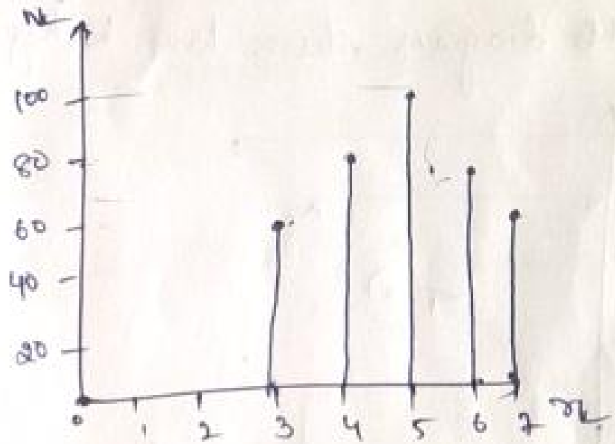
① Gray level (r_k) 0 1 2 3 4 5 6 7

n_k 80 100 90 60 30 20 10 0

② Gray level (r_k) 0 1 2 3 4 5 6 7

n_k 0 0 0 60 80 100 80 70





Equalize the 1st histogram.

Gray level	n_k	$P(r_k) = n_k/n$ PDF	S_k	$S_k \times 7$	Histogram equalization	New n_k
0	80	0.20	0.20	1.4	1	80
1	100	0.25	0.45	3.15	3	100
2	90	0.23	0.68	4.76	5	90
3	60	0.15	0.83	5.81	6	90
4	30	0.07	0.9	6.3	6	
5	20	0.05	0.95	6.65	7	30
6	10	0.02	0.97	6.79	7	
7	0	0	0.97	6.79	7	

$n = 390$

Equalize the 2nd histogram

Grey level	n_k	P.D.F n_k/n	S_k	$S_k \times 7$	Histogram Equilisation
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	60	0.15	0.15	1.05	1
4	80	0.20	0.35	2.45	2
5	100	0.25	0.6	4.2	4
6	80	0.20	0.8	5.6	6
7	70	0.179	0.97	6.79	7

$n = 390$

2nd image | 1st image

r_k	Histogram Equilization (2nd image)	Histogram Equilization (1st image)	new (n_k)
0	0	1	80
1	0	3	100
2	0	5	90
3	1	6	
4	2	6	90
5	4	7	
6	6	7	30
7	7	7	

r_k	0	1	2	3	4	5	6	7
	0	0	0	80	80	100	90	30

Fundamentals of Spatial filtering.

Spatial filter is one of the principle tool used for Image processing. These are used for image enhancement.

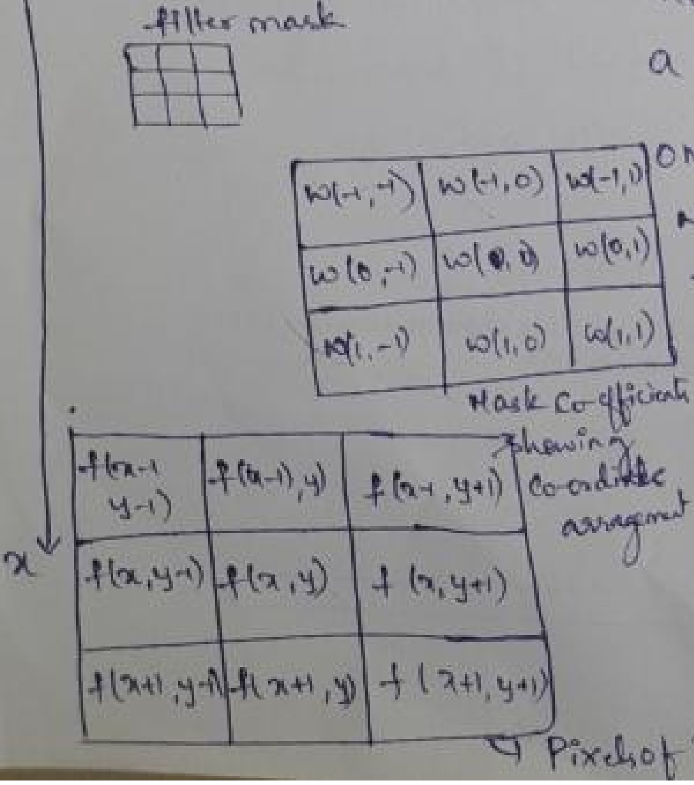
Here filter refers to passing or rejecting some frequency components.

The filter that passes lower frequency is LPF, the overall effects produce by an LPF is to blur an image. or it can be called as Smoothening an image. So we can accomplish a similar smoothening directly on image itself by using spatial filters.

This spatial filter also known as kernel, mask, template or window.

These spatial filters offers more versatality compared to frequency domain because they can also be used for non linear filtering operations.

Mechanics of Spatial filtering.



Pixels of image under Mask.

A spatial filter consists of a neighbour-hood typically a small rectangle a predefined operation that is performed on the image pixel by using the neighbourhood.

The filtering operations will create a new pixel. If the operations performed are linear then the filter is linear spatial filter otherwise they are known as Non linear spatial filters.

~~The process of the~~

The Mechanics of Spatial filter is shown in the fig. a

The process consists of moving this filter mask from point to point in an image.

At each point the response of the filter at that point is calculated using predefined relationship.

For linear spatial filter the response is given by sum of product of the filter co-efficients and the co-efficients in the image pixels.

Eg Take 3×3 Mask as shown in fig ⓐ. The response of of linear filter is given as:- product of filter co-efficient and the corresponding image pixel.

$\therefore R = w(-1,-1) f(x-1, y-1) + w(-1,0) f(x-1, y) + \ldots$

$\ldots w(0,0) f(x, y) + \ldots \ldots w(1,1) f(x+1, y+1)$ — (1)

* The equation indicates the sum of product ~~coefficients~~ of mask coefficients with the co-efficients of the image pixel value.

* $w(0,0)$ co-incides with image value $f(x,y)$ this indicates that the mask is centered at (x, y).

* For a mask size $m \times n$ Assume $m = 2a+1$
  $n = 2b+1$

a, b are non-ve integers. indicates that mask is of odd size means a=1 b=1 mask size 3×3.

so the minimum size is 3×3.

In general

Linear filtering of image of size $M \times N$ & mask size $m \times n$

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t) \quad \text{--- (2)}$$

where $a = \frac{m-1}{2}$ $\quad b = \frac{n-1}{2}$.

Eqn (2) is similar to convolution for this reason the linear spatial filtering is often referred as Convolution m

or Convolution Kernel.

By simplifying

$$R = w_1 z_1 + w_2 z_2 - - - - - + w_{mn} z_{mn}$$

$w \rightarrow$ mask Co-efficients

$z \rightarrow$ The values of the image gray levels.

$m \times n \rightarrow$ Total no of Co-efficients.

$$R = \sum_{i=1}^{mn} w_i z_i \quad \text{--- (3)}$$

For 3×3 general mask.

$$R = w_1 z_1 + w_2 z_2 \cdots w_9 z_9.$$

$$R = \sum_{i=1}^{9} w_i z_i \quad \text{--- (4)}$$

Once the operation is performed the value of z_5 is changed to the Resultant Response.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Subimage

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

mask Coefficients 3x

Smoothing Spatial filters

→ used for Blurring & Noise Reduction

→ Blurring : is used in preprocessing such as Removal of small details from an image, prior to object extraction.

→ Noise reduction : can be accomplished by blurring with a linear or Non linear filter.

Smoothing linear filters –

The o/p of smoothing filters using the linear spatial filter is average of pixels contained in the neighborhood of the filter mask.

It is also known as averaging filter / Low pass filter

eg. ⓐ

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

3×3 smoothing filter mask.

Standard average filter mask.

ⓑ

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 0 \\ \hline \end{array} \times \frac{1}{16}$$

weighted average filter mask

→ Smothing or filter is Replacing each pixel in an image by the average gray level values of the filter mask.

→ Application → Noise reduction

→ Side effects → blur edges.

Fig a. The spatial averaging filter where all the co-efficients are equal are sometimes known as Box filters.

Standard average of pixel values.

→ M×n ~~matrix~~ mask → image is normalized by 1/m×n

Fig b is the example of weighted smoothing filter

→ The pixel at the center of mask is given more importance by multiplying it by a higher value. This is to reduce blurring during Smoothing process

① The general implementation for filter w with image M×N & mask m×n

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

w(s,t) → Co-efficients of filter

f → Co-efficients of image gray levels

~~Order Statistics filters:~~

→ ~~are also known as Non linear spatial filters~~

~~Spa~~ **Sharpening of Spatial filters:**

The principle objective of sharpening is to highlight the finate details or to enhance details that has been blurred either in error or as a natural method during image acquisit

Application → Electronic printing.
Medical imaging
Industrial Inspections &
autonomous Guidance in Military systems.

→ As Image blurring can be accomplished by pixel averaging since blurring can be done in integration so sharpening can be obtained by spatial differentiation

Image differentiation → enhances edges and noise & deemphasizes areas with slowly varying gray-level values.

Foundation of image sharpening

→ First order & Second order derivaties to sharpen the image

→ Derivatives of a digital function are defined in terms of differences

→ The defination used for 1st derivative

(a) Must be zero in flat segment.

(b) must be Non zero at onset of a graylevel, step or ramp

(c) must be nonzero along ramp.

→ For 2nd Order derivative

(a) must be zero in flat area

(b) must be nonzero at the onset (starting) & end of graylevel ~~set~~ step or ramp.

(c) must be zero along ramp of constant slope.

→ The shortest distance over which changes can occur is b/n adjacent pixels.

→ The basic definition of

1st order derivative of 1-D $f(x)$

$$\frac{\delta f}{\delta x} = f(x+1) - f(x)$$

2nd order derivative of 2-D $f(x,y)$

$$\frac{\delta f}{\delta x^2} = f(x+1) + f(x-1) - 2f(x)$$

③ Use of Second order derivatives for Enhancement

— The Laplacian.

Let us understand a practical filter based on second order derivative

Laplacian filter are filter defined based on both x & y co-ordinate.

Defn. $\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$

In x- direction

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

In y- dirn

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

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

Consider a 3×3 filter

f(x-1, y-1)	f(x-1, y)	f(x-1, y+1)
f(x, y-1)	f(x, y)	f(x, y+1)
f(x+1, y-1)	f(x+1, y)	f(x+1, y+1)

0	1	0
1	-4	1
0	1	0

→ is one the Laplacian filter designed using the above eqn.

Different filters.

0	1	0
1	-4	1
0	1	0

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

1	1	1
1	-8	1
1	1	1

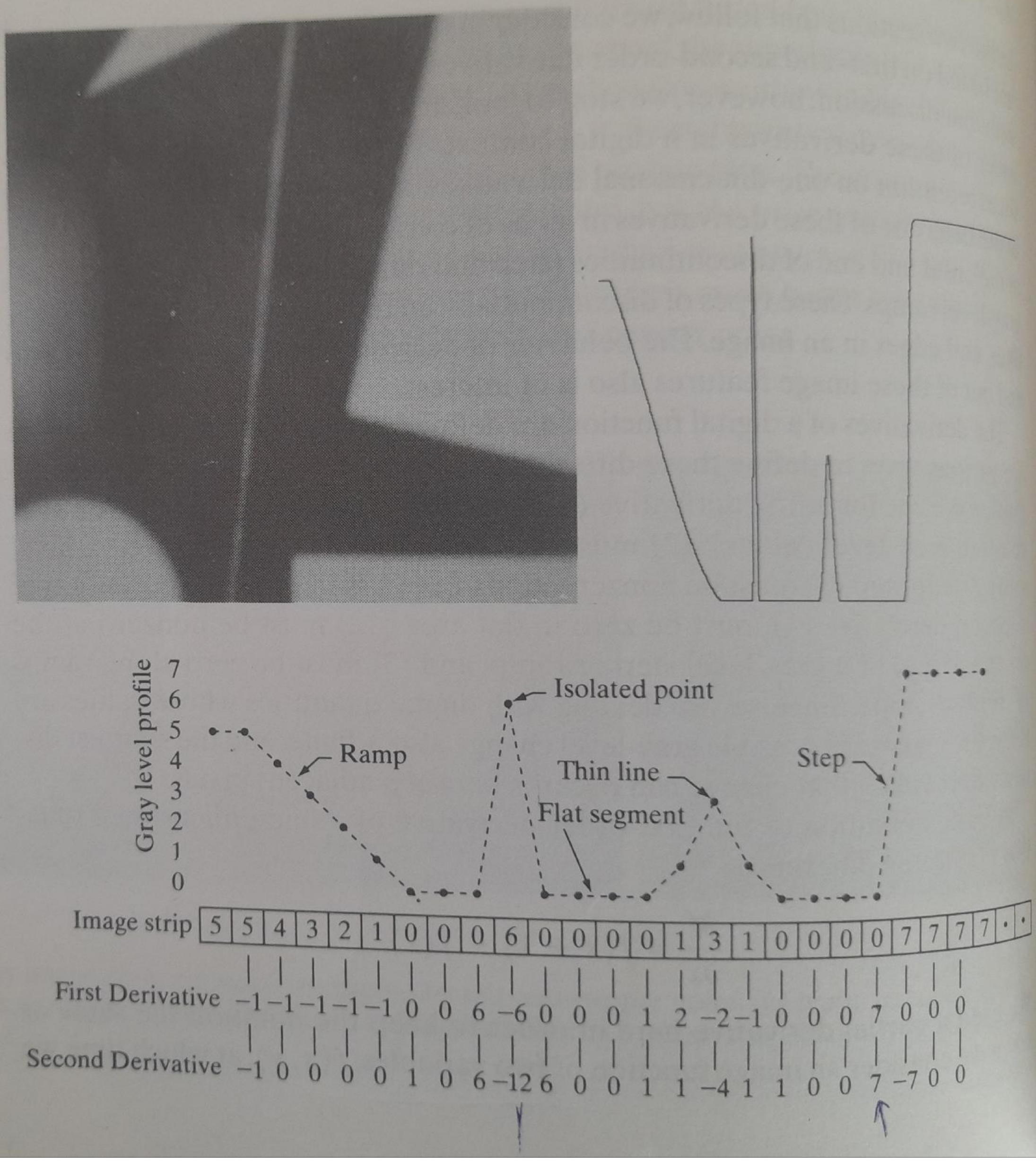
-1	-1	-1
-1	8	-1
-1	-1	-1

Center pixel will be

to highlight the centre pixel compared to other pixel value.

a b
c

FIGURE 3.38
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point.
(c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



* Conclusion - (5)

→ The 1st order derivative produces thick edges.

2nd order derivative produces fine details. (thin lines &

→ The 1st order - graylevel step - stronger Response

2nd order → produces a double response in the gray level.

(b) Order statistics filters.

→ These are Non linear filters
→ Their response is based on the ordering/Ranking the Pixels in the image encompassed by the filter
Eg arrange the pixels in ascending/descending then find the response.
→ These filters will replace the value of center pixel value with value determined by ranking result.

Types
1. Median filter → This filter replaces the value of a pixel by the median of the gray level.
2. It is a most popular as it provides excellent noise reduction capabilities
→ It produces less blurring compared to linear spatial filters
→ These are practically effective for Impulse Noise - also known as salt & pepper Noise.

Eg

10	20	20
20	15	20
20	25	100

⟹

10	20	20
20	20	20
20	25	100

10 15 20 20 20 20 20 25 100

2. Max filters: Finding the brightest point

$R = \max \{ z_k \mid k = 1, 2, 3 \ldots 9 \}$

Max value = 100 brightest point

3. Min filters - Finding the darkest point

$R = \min \{ z_k \mid k = 1, 2, 3 \ldots 9 \}$

Min value = 10 darkest point

(7) Image Enhancement in frequency domain.

Here images from spatial domain are converted to frequency domain then processed.

Inverse transform is applied to bring back images into spatial domain.

In frequency domain filter are used for smoothening & sharpening of an image by removing high & low freq component.

~~The for~~ Once we apply frequency domain filter the change will takes place on the whole image. unlike in spatial domain where the manipulation ~~was~~ has taking place pixel by pixel.

Types of filter

a. Low pass filters → Removes all the high frequency components
→ Mainly used for smoothening the image
→ Used to remove the noise from the image

b. ~~High~~ High pass filter → removes all the low frequency components
→ Mainly used for sharpening the image

Both Low pass & High pass filter can be classified into 3 types
Ideal filter
Butterworth filter
Gaussian filter

Fourier transform.

→ Provides the Relationship b/n Spatial & frequency domain.
→ Fourier transform is used for the image enhancement in frequency domain.

– 1-D Discrete fourier transform.

Consider $x(n) \xrightarrow{DFT} X(k)$
Spatial domain — frequency domain.

$$\therefore X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} \quad ; 0 \le k \le N-1$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, e^{\frac{j2\pi}{N}kn} \qquad 0 \le n \le N-1.$$

8

The fourier Spectrum

$$|F(u)| = \left[R^2(u) + I^2(u)\right]^{1/2}$$

Phase angle

$$\phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$$

Power Spectrum.

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

2-D Discrete fourier transform

$$f(x,y) \xleftrightarrow{\text{2-D-DFT}} F(u,v)$$

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

$$F(u,v) \xrightarrow{\text{2D IDFT}} f(x,y)$$

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

u & v → transformed or frequency variable

x & y → Spatial or In

Fourier Spectrum

$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$$

Phase angle

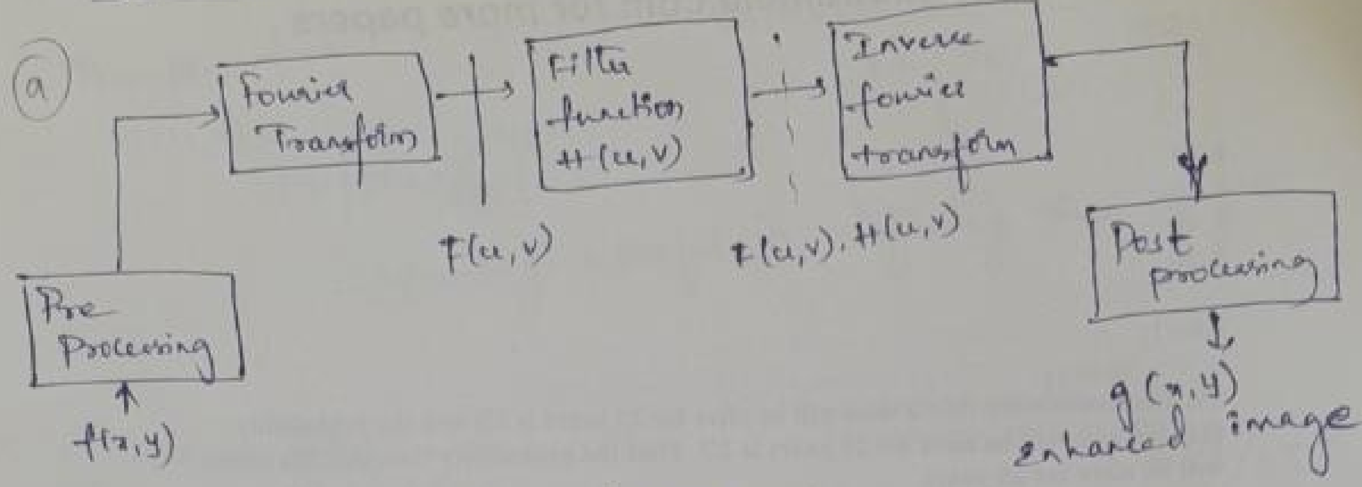
$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

Power spectrum

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

Steps for filtering in frequency domain

(a)



1. Take the image $f(x,y)$ of size $N \times N$
2. Preprocessing $\to f(x,y)(-1)^{x+y}$
3. $F(u,v) \xrightarrow{FT}$ FT of $[f(x,y)]$
4. $H(u,v) \to$ filter in freq domain
   $G(u,v) \to H(u,v) \cdot F(u,v)$.
5. $g(x,y) \longrightarrow IFT\ [F(u,v)]$

Image smothering & sharpening using frequency domain filter

1. Lowpass filter
   - → Removes high freq component
   - → Used for Smothering
   - → Ideal Lowpass filter

$$H(u,v) = \begin{cases} 1 & ; D(u,v) \le D_0 \\ 0 & ; D(u,v) > D_0 \end{cases}$$

$D_0 \to$ Non negative quantity

$D(u,v) \to$ distance from point (u,v)

$f(x,y) \to$ of size $M \times n$.

$$D(u,v) = \left[u - \frac{m}{2}\right]^2 + \left[\frac{u-N}{2}\right]^2$$

High pass filter

Removes low freq components

→ Sharpen

Ideal Highpass filter

$$H(u,v) = \begin{cases} 0 & ; D(u,v) \le D_0 \\ 1 & D(u,v) > D_0 \end{cases}$$

Disadvantages blurred image

2. Butterworth LPF

Transform func

(10) $$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

Butterworth HPF

$$H(u,v) = \frac{1}{1 + [D_0/D(u,v)]^{2n}}$$

Advantages:- Useful in defining the edges

3. Gaussian LPF

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

Removes low freq noise

Gaussian HPF

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Removes high freq noise.

2. Butterworth LPF

Transform func

(10) $$H(u,v) = \frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$$

Butterworth HPF

$$H(u,v) = \frac{1}{1+\left[D_0/D(u,v)\right]^{2n}}$$

Advantages:- Useful in defining the edges.

3. Gaussian LPF

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

Removes low freq noise

Gaussian HPF

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Removes high freq noise.