(4) (5) 72

UNIT - II (PART-A)

INTENSITY TRANSFORMATIONS & SPATIAL FILTERING

[IMAGE ENHANCEMENT]

INTRODUCTION :-

* Image Enhancement techniques are designed to improve the quality of an image as perceived by a human being.

* The main objective is to improve the quality of an image even by the degradation is available. This can be achieved by increasing the dominance of some features or decreasing the ambiguity b/w different regions of image.

* The Intensity transformations operate on single pixels of an image, for the purpose of contrast manipulation & image thresholding.

* Spatial filtering deals with performing operations such as image sharpening by considering every pixel in an image.

Definition :-

* Image Enhancement refers to processed image which is more suitable than original image.

* Enhancement methods are application specific and are often developed empirically.

REASONS TO PREFER ENHANCEMENT TECHNIQUES :-

* Due to bad illumination sources
* For maintaining correct acceptance angle
* For good dynamic range

CLASSIFICATION :-

Image Enhancement techniques can be done in 2 ways. They are:

(a) Spatial domain Enhancement

Masking   Filtering   point operation

(b) Frequency domain Enhancement

BACKGROUND :

Basics of Intensity Transformations & Spatial Filtering :

The spatial domain processes can be denoted by the expression

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

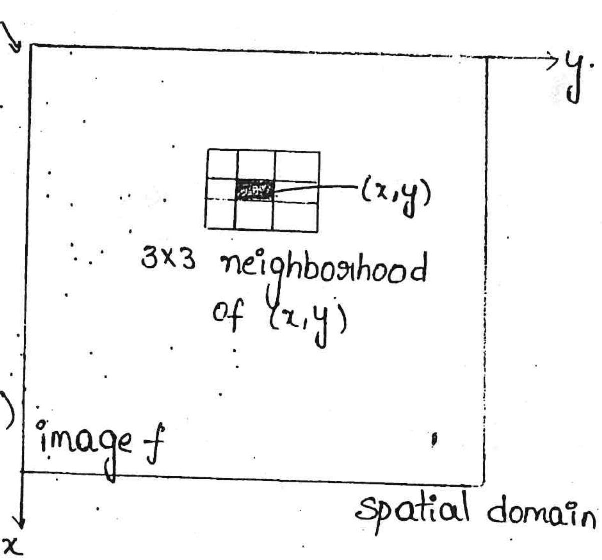
where $f(x,y)$ is i/p image

$g(x,y)$ is o/p image

T is an operator

* The operator (T) can be applied to a single image or to a set of images

Eg:- Let us consider a 3×3 neighborhood about a point (x,y) in an image in spatial domain as shown



* From the figure,

Consider an arbitrary location say (100, 150). Assuming that the origin of neighborhood is at its centre then the o/p $g(100,150)$ is obtained by considering

$g(100,150)$ = Sum of [$f(100,150)$ & its 8 neighbors divided by 9]

The origin of the neighborhood is then moved to next location & the procedure is repeated as discussed above to get the o/p image.

This procedure is called spatial filtering & the neighborhood along with operator 'T' is known as spatial filter.

The smallest possible neighborhood is of size 1×1. In this case the o/p $g(x,y)$ depends only on the value of f at single point (x,y) & T becomes an intensity transformation function given by

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

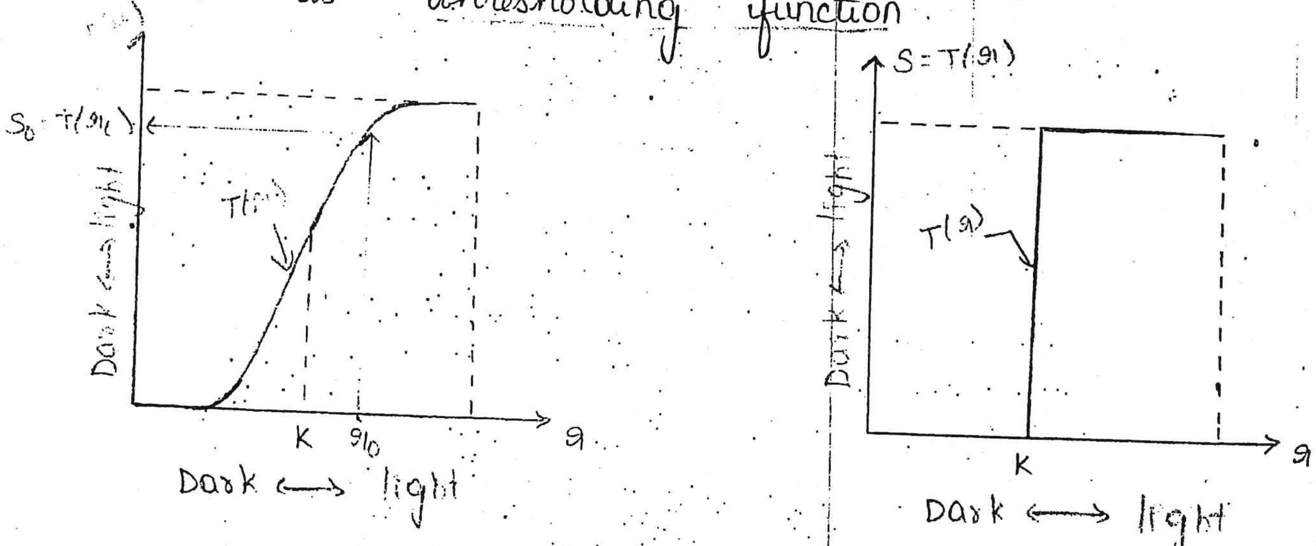
where s & r are variables

For example if $T(r)$ has the form as shown in fig (a) then the effect of applying the transformation results in 2 factors :

(i) To produce an image of <u>higher contrast</u> than the original by darkening the intensity levels below 'k' & brightening the levels above 'k'.

In this case the values of r lower than k are compressed by the transformation function which is known as <u>contrast stretching</u>.

(ii) To produce a two-level (binary) image by considering the values of 'r' higher than 'k' which is known as <u>thresholding function</u>.



(a) Contrast stretching function

(b) Thresholding function

fig :- Intensity transformation functions

TECHNIQUES OF SPATIAL DOMAIN :-

1. Image Negative
2. Contrast stretching
3. Clipping
4. Thresholding
5. Log transformation
6. Level slicing
7. Bit plane slicing
8. power law transformation
9. Histogram Specification
10. Histogram Equalisation

point operation techniques.

1. IMAGE NEGATIVE :-

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

$$s = L-1-r \qquad r \rightarrow \text{Intensity value}$$

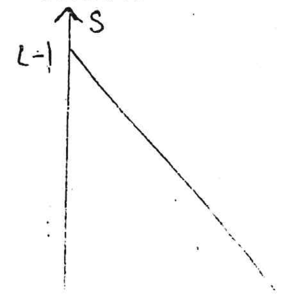
We know that $s = T(r)$ &

$L = 2^b$

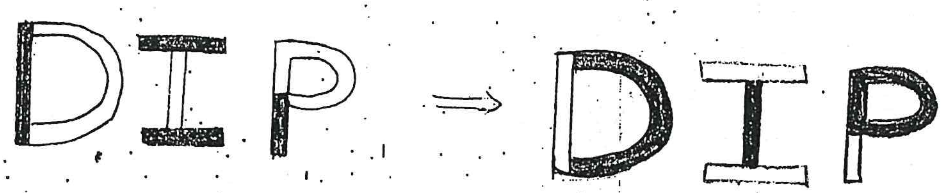
where b is bit-width

Let $b=8$, then

$$s = T(r) = 2^8-1-0$$
$$= 255$$



The main aim of this Image negative is, the dark region is converted into bright & bright into dark.



Eg:- Let us assume an image of pixels

$$\text{i/p image } f(x,y) = \begin{bmatrix} 2 & 4 & 6 & 10 \\ 6 & 7 & 0 & 0 \\ 15 & 15 & 1 & 2 \\ 0 & 15 & 15 & 0 \end{bmatrix}_{4\times4}$$

Here bit width (b) = 4

$$L = 2^b = 2^4 = 16$$

$$S = L-1-r = 16-1-r = 15-r$$

$$\Rightarrow \begin{bmatrix} 15-2 & 15-4 & 15-6 & 15-10 \\ 15-6 & 15-7 & 15-0 & 15-0 \\ 15-15 & 15-15 & 15-1 & 15-2 \\ 15-0 & 15-15 & 15-15 & 15-0 \end{bmatrix}$$

$$\therefore \text{processed image} = \begin{bmatrix} 13 & 11 & 9 & 5 \\ 9 & 8 & 15 & 15 \\ 0 & 0 & 14 & 13 \\ 15 & 0 & 0 & 15 \end{bmatrix}$$

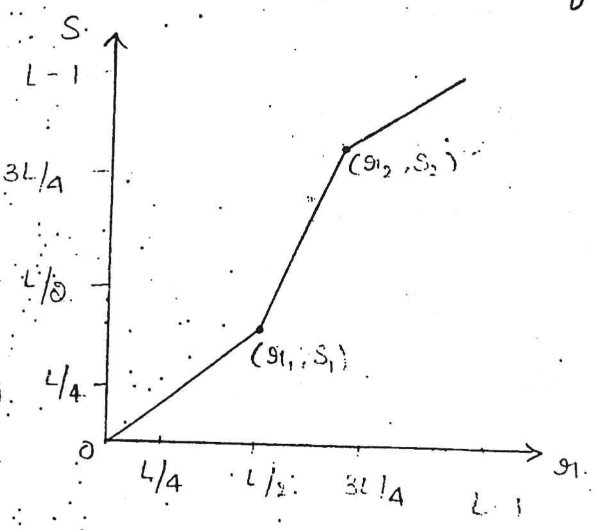
Applications :- Negatives of digital images are useful in numerous applications such as displaying medical images & photographing a screen with monochrome +ve film.

2. Contrast stretching :-

The process of expanding the range of intensity levels in an image, inorder to utilise the full range of intensity levels is known as contrast stretching.

It is one of the simplest piece-wise linear functions. Low-contrast images can result from poor illumination.

The adjacent figure shows the typical transformation used for contrast stretching.



Case-1 :- From the figure, if $r_1 = s_1$ & $r_2 = s_2$ then the transformation is linear & produces no change in intensity levels.

Case-2 :- If $r_1 = r_2$; $s_1 = 0$; $s_2 = L-1$ the transformation becomes a thresholding function which creates a binary image.

* The result of contrast stretching is obtained by assuming $(r_1, s_1) = (r_{min}, 0)$ & $(r_2, s_2) = (r_{max}, L-1)$

Eg :- Let us consider an image of 4x4 size

Here $L = 2^6 = 64$

∴ The values of (r_1, s_1) & (r_2, s_2) are

$(r_1, s_1) = (r_{min}, 0) = (r_{min}, 0)$

$(r_2, s_2) = (r_{max}, L-1) = (r_{max}, 63)$.

* Contrast stretching occurs due to

(a) clipping (b) Thresholding

* We know that

$$S = T(r)$$

Here
$$s = \begin{cases} \alpha r & 0 < r < a \\ \beta(r-a) + v_a & a < r < b \\ \gamma(r-b) + v_b & r > b \end{cases}$$

If $\alpha = \gamma = 0$, $\beta(r-a) + v_b$

clipping & thresholding are specical cases in this.

If $a = b = T$ threshold occurs

If $r_1 = r_2$

$\Rightarrow s_1 = 0$; $s_2 = L-1$

If $r_1 \neq r_2$; $s_1 > 0$; $s_2 > s_1$

$s = L-1$

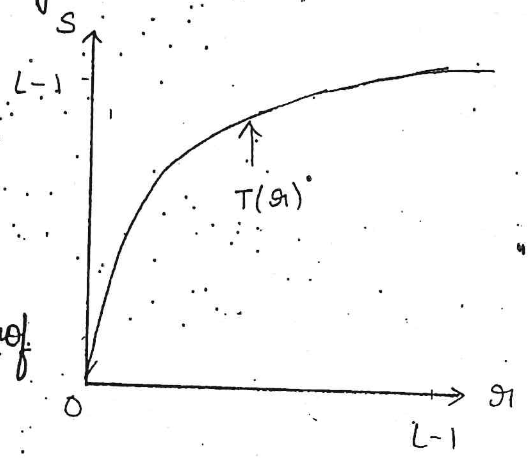
* Log transformation :-

The general form of log transformation is expressed as

$$S = T(r) = c \log(1+r)$$

where c is constant & it is assumed that $r \geq 0$.

The shape of the log curve shows that this transformation maps a narrow range of low gray-level values in the i/p image into a wider range of o/p levels.



The log transformation has the important characteristic of compressing the dynamic range of images with large variations in pixel values.

* LEVEL SLICING :-

The purpose of LEVEL SLICING is to highlight a specific range of gray values.

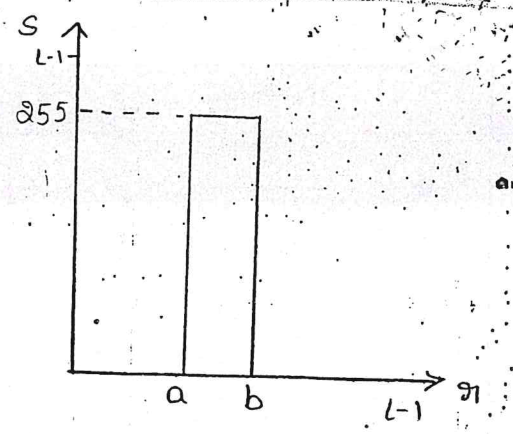
The 'Level Slicing' can be done with 2 approaches. They are:

(a) Level slicing without preserving background.
(b) Level slicing with background.

(a) LEVEL SLICING WITHOUT PRESERVING BACKGROUND :-

In this all the gray levels of particular range are displayed with higher values & remaining gray levels are displayed with lower values.

From the adjacent figure we can observe that only one part is highlighted & remaining are zero's



$$S = T(r) = \begin{cases} L & a \leq r \leq b \\ 0 & \text{otherwise} \end{cases}$$

Drawback :- The main drawback of this approach is, that the background information is discarded.

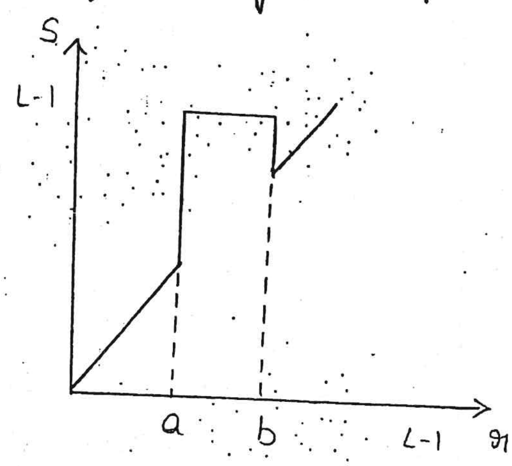
(b) LEVEL SLICING WITH BACKGROUND :-

In this high values are displayed for particular range & original gray level values in other areas.

Since the remaining gray levels does not become zero, it preserves the background of image.

Here we consider

$$S = T(r) = \begin{cases} L & a \leq r \leq b \\ r & 0 \leq r \leq L-1 \end{cases}$$



* BIT-PLANE SLICING :-

The main objective in this is instead of highlighting gray level ranges we should highlight the contribution made to total image by

Considering specific no of bits.

In Bit plane slicing the image is divided according to no. of bits.

Lower planes are considered as Bit-plane 0 (LSB bits) & Upper planes are considered as Bit-plane 7 (MSB bits).

The 3-main objectives of Bit-plane slicing are:

(a) Converting gray level image to binary image

(b) Representing an image with fewer bits & Compressing the image to smaller size

(c) Enhancing the image by focussing.

* For an 8-bit image, 0 is encoded as 00000000 & 255 is encoded as 11111111. Any number between 0 & 255 is encoded as one byte.

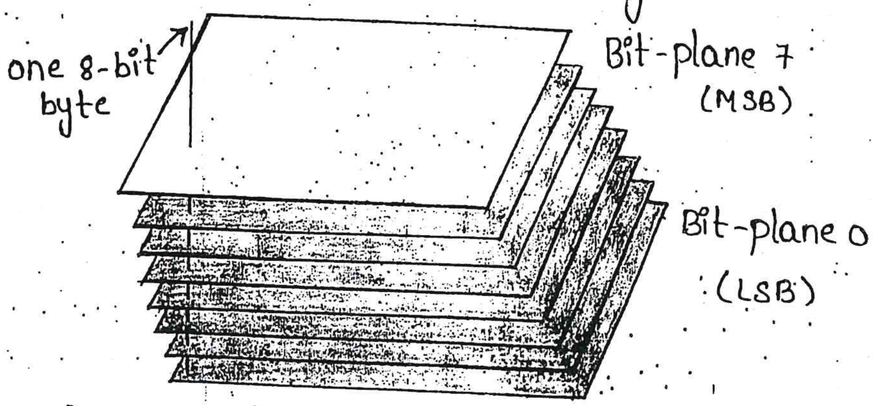


fig: Bit plane representation of an 8-bit image.

* By using Bit-plane slicing image can be Compressed.

* POWER LAW TRANSFORMATION :-

Power law transformations have the basic form

$$s = c r^{\gamma} \quad \text{——} \; (1)$$

where c & γ are positive constants.

The above eqn can also be written as

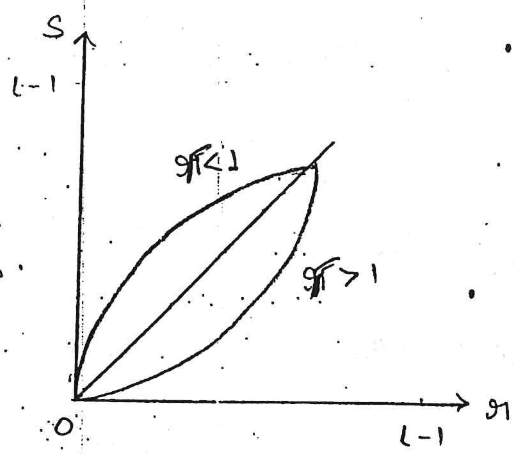
$s = (r+\epsilon)^{\gamma}$ for offset purpose (i.e., measurab o/p when the i/p is zero)

* Since gamma (γ) is used to correct the power law response phenomenon, it is known as gamma correction & this power law transformation is also known as GAMMA TRANSFORMATION.

From (1), if $r = s$ then original intensity levels of image = o/p intensity levels.

$r < 1$ for high intensities

$r > 1$ for low intensities.



Advantage :-

1. This law is used in variety of devices for image capturing, printing & display responding purpose.
2. Used for Gamma correction.
3. power-law transformations are useful for general purpose contrast manipulations.

HISTOGRAM :-

Histogram of an image is defined as the representation including relative frequency of occurance of various gray levels in the image.

Inorder to improve the visual quality of image we use histogram manipulation techniques. The histogram provides more insight about image contrast & brightness

The histogram of an image is a plot of the no. of occurances of gray levels in the image against the grey-level values.

Image near to zero values represent dark one.

Image near to L-1 values represent bright one.

For a low contrast image (dark image) the histogram will not spread equally i.e., the histogram will be narrow.

For a high contrast image (bright image) the histogram will have an equal spread in grey level.

It means the histogram of dark image is clustered towards lower gray level & the histogram of bright image is clustered towards higher gray leve

Formula :-

Let r_k is the kth gray level of i/p image

n_k is the no of pixels in the i/p image

Then the histogram of a digital image with intensity levels in the range $[0, L-1]$ is given as

$$\boxed{h(r_k) = n_k}$$

Since the histogram represents intensity values then <u>normalised histogram</u> is given as

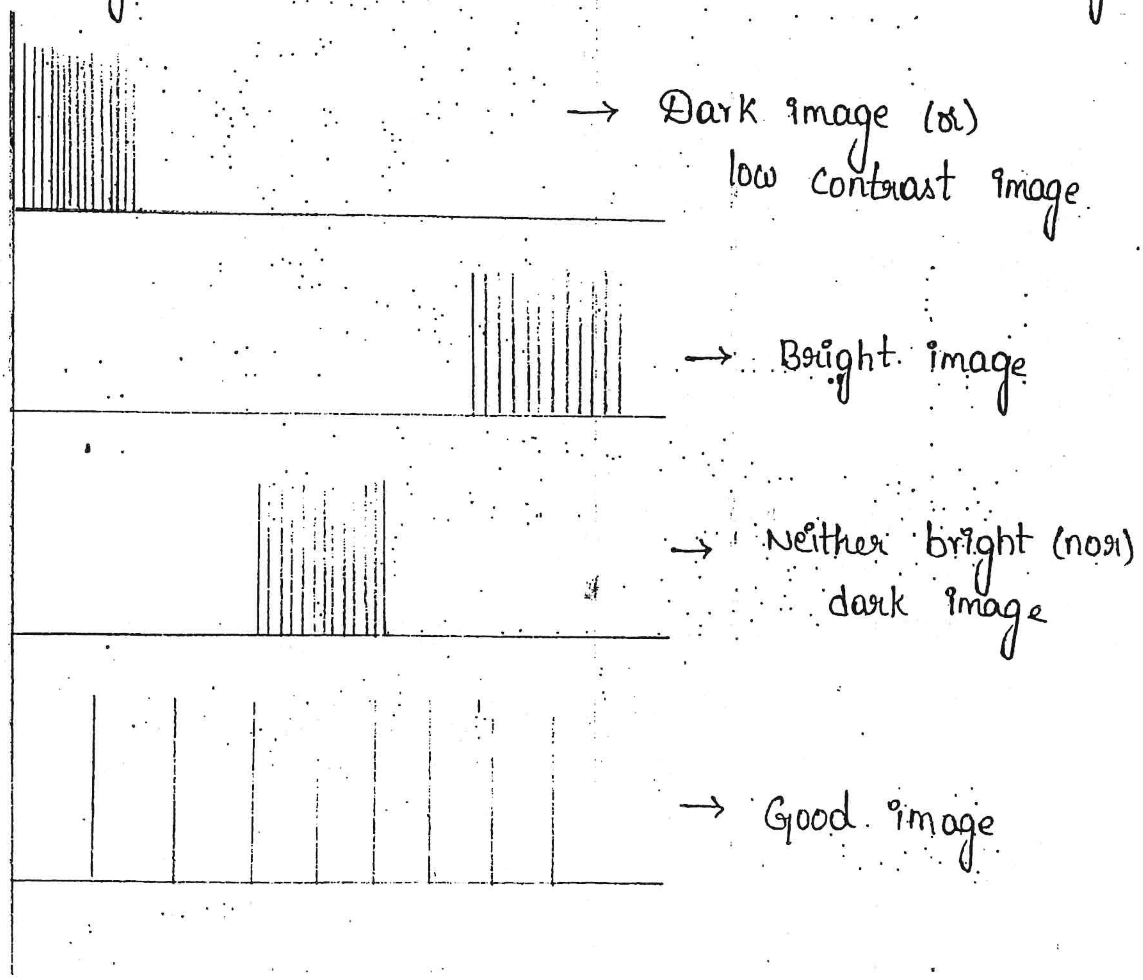
$$h(r_k) = \frac{n_k}{n}$$

when 'n' is no of intensity levels.

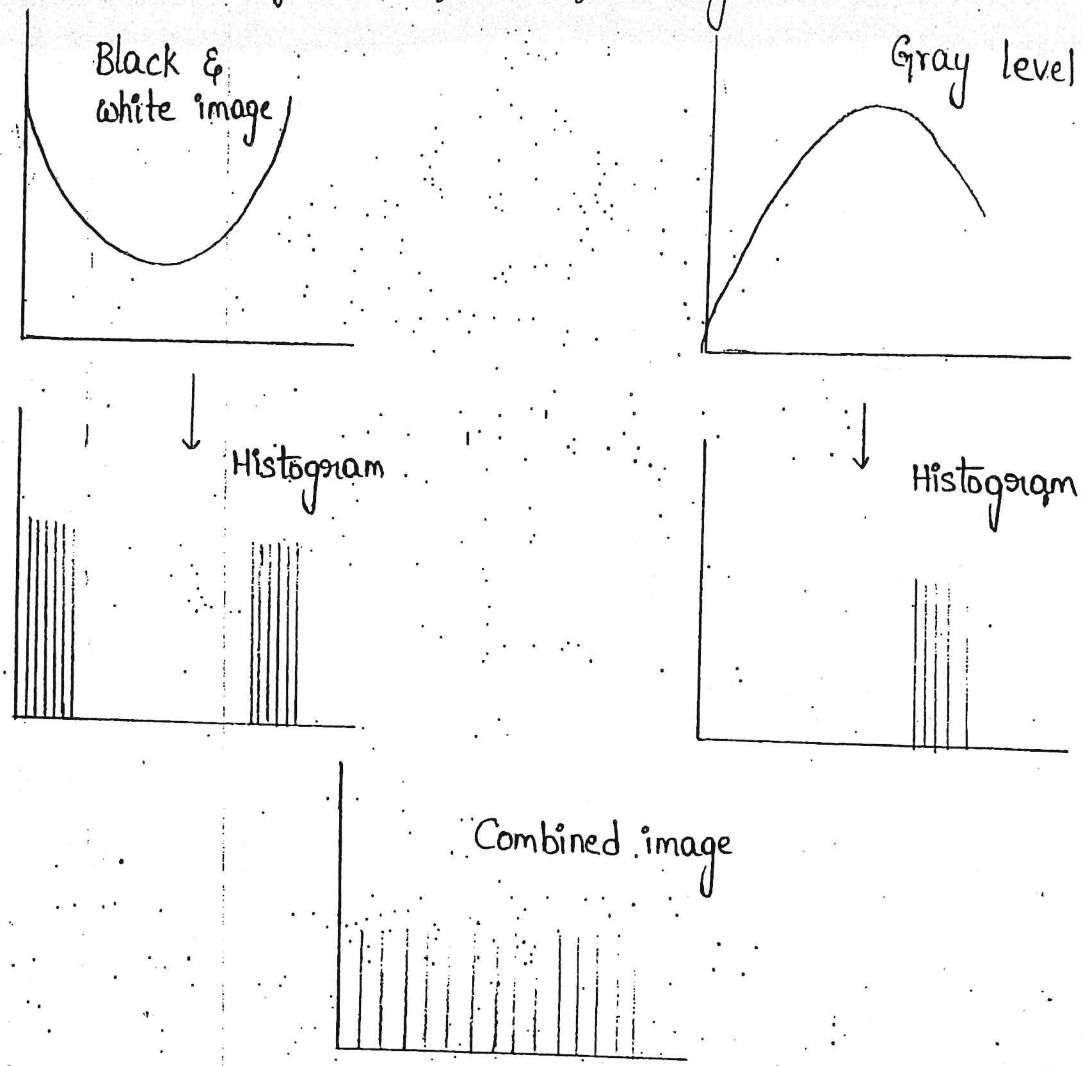
* plot histogram for:

(a) dark image: (b) Bright image (c) Neither bright nor dark image

(d) Good image.



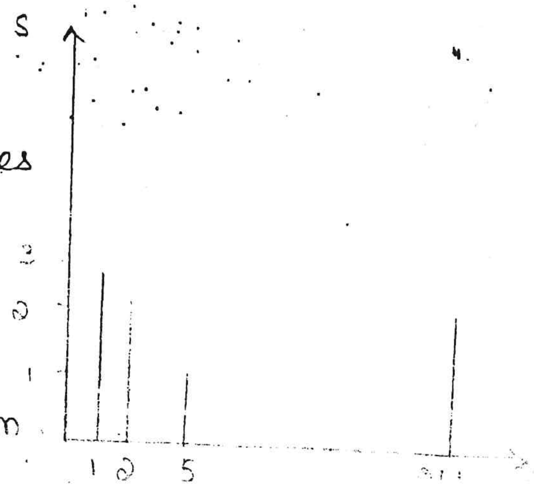
* plot the histogram for following :



Eg :-

For $\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 5 \\ 255 & 255 & 0 \end{bmatrix}$ Draw Histogram.

From the given matrix we observe that '1' is repeated 3 times, hence its value is 3. And 2 is repeated 2 times hence its corresponding value is 2 & so on. The adjacent figure represents the corresponding histogram



* HISTOGRAM EQUALISATION (OR) HISTOGRAM LINEARIZATION :-

Let us consider r is original image &

S is processed image

So that $S = T(r)$ —— ①.

In histogram Equalisation, we consider 'r' & 's' as random variables.

The transformation function in ① should satisfy the following conditions :

(i) r limit is from $0 < r < 1$

(ii) $T(r)$ should be a single valued & monotonically increasing function.

(iii) The transformation should be continuous & differentia

* If 'r' limit is from [0 1] then we get black & white image.

The probability density function of transformed gray levels for ① is obtained as :

$$P_s(s)ds = P_r(r)\, dr$$

$$P_s(s) = P_r(r) \frac{dr}{ds} \quad \text{—— ②}$$

From ①,

$$S = T(r) = \int_0^r P_r(w)\, dw$$

Differentiating the eqn, we get

$$\frac{d(s)}{dr} = \frac{d}{dr}\left(\int_0^r P_r(w)\, dw\right)$$ where w is dummy variable

$$\frac{ds}{dr} = P_r(r)$$

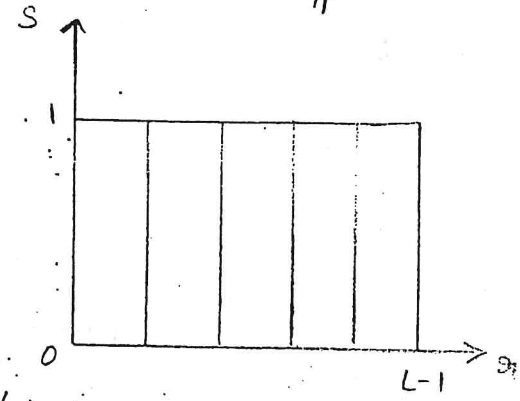
$\therefore$ From ②,

$$P_s(s) = P_r(r) \cdot \frac{1}{P_r(r)}$$

$$\boxed{P_s(s) = 1}$$

It means for all intensity levels the o/p is '1'

$P_s(s) = 1$ represents uniform Equalisation.



Drawback :-

By using Histogram Equalisation we can't get accurate manipulations.

* **Example :-**

perform Histogram Equalisation for the image

$$\begin{bmatrix} 4 & 4 & 4 & 4 \\ 5 & 4 & 3 & 4 \\ 3 & 3 & 4 & 5 \\ 4 & 5 & 2 & 5 \end{bmatrix}_{4 \times 4}$$

Solution :-

The max value of image = 5.

$\therefore$ we need a minimum of 3 bits to represent the number 5 (101). There are 8 possible gray levels from 0 to 7.

The histogram of i/p image is given below:

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	0	1	3	8	4	0	0

Step-1 :- Compute the cumulative sum of above values:

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	0	1	3	8	4	0	0
Cumulative Sum	0	0	1	4	12	16	16	16

Step-2 :- Divide the cumulative sum obtained in step-1 by total no. of pixels.

In this case, the total no. of pixels = 16

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	0	1	3	8	4	0	0
Cumulative sum	0	0	1	4	12	16	16	16
Total no. of Pixels	0/16	0/16	1/16	4/16	12/16	16/16	16/16	16/16

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

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	0	1	3	8	4	0	0
Cumulative Sum	0	0	1	4	12	16	16	16
Total no. of pixels	0/16	0/16	1/16	4/16	12/16	16/16	16/16	16/16
Multiplying the result by 7.	0	0	$7/16 \cong 1$	2	5	7	7	7

Step-4 :- Mapping of gray level by one-to-one correspondence.

original gray level	Histogram equalised values.
0	0
1	0
2	1
3	2
4	5
5	7
6	7
7	7

The original image & the histogram equalised images are shown side by side.

$$\begin{bmatrix} 4 & 4 & 4 & 4 \\ 5 & 4 & 3 & 4 \\ 3 & 3 & 4 & 5 \\ 4 & 5 & 2 & 5 \end{bmatrix} \xrightarrow[\text{Equalisation}]{\text{Histogram}} \begin{bmatrix} 5 & 5 & 5 & 5 \\ 7 & 5 & 2 & 5 \\ 2 & 2 & 5 & 7 \\ 5 & 7 & 1 & 7 \end{bmatrix}$$

original image Histogram equalised image.

* HISTOGRAM SPECIFICATION (OR) HISTOGRAM MATCHING :-

The main drawback in Histogram Equalisation is, it is not suitable for interactive image Enhancement applications.

The method used to generate a processed image that has a specified histogram is called histogram matching (or) histogram specification.

Histogram matching means highlighting the particular part.

Let r is i/p image

s is processed image

z is o/p image

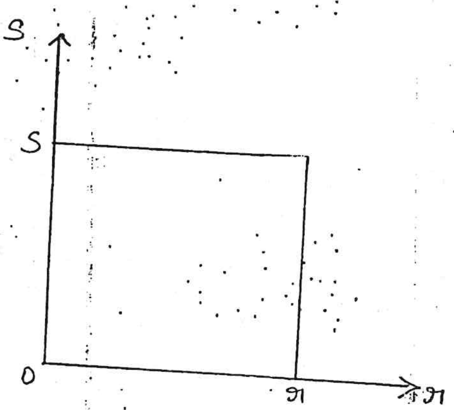
Their probabilities are $P_r(r)$, $P_s(s)$; $P_z(z)$

We know that

$$s = T(r)$$

$$s = G(z)$$

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



$$s = \int_0^r P_r(w)\, dw$$

$$s = \int_0^z P_z(t)\, dt$$

Differentiating the above eqn, we get

$$\frac{ds}{dr} = P_r(r)$$

$$\Rightarrow \frac{ds}{dz} = P_z(z)$$

$$s = G(z) = G^{-1}(s)$$

$$= G^{-1}(T(r))$$

$$= G^{-1}(P_r(r))$$

Discrete case :-

$$s = T(r) = \sum_k \frac{n_k}{n}$$

$$P_r(r) = \sum_k \frac{n_k}{n}$$

$$S = \sum_{K} \frac{n_K}{n}, \quad K = 0 \ldots L-1$$

* SPATIAL MASKING (OR) SPATIAL FILTERING TECHNIQUES :-

Masking:- Some intensities are replaced with other.

Filtering techniques
- Image Smoothing
  - Linear
  - Non-linear
- Image Sharping

Image Smoothing techniques:

1. Low pass filtering (or) Average masking } Linear filters
2. Weighted average Masking } Linear filters
3. Median filter ⟶ Non-linear filter

Image sharping techniques:

4. I order (or) Gradient Masking
5. II order (or) Laplacian Masking
6. Unsharp masking
7. High boost filtering
8. Homo-morphic filtering
9. Sobeal Masking
10. Robert Masking
11. pre-witt Masking

SPATIAL SMOOTHING FILTERS :-

Smoothing filters are used for blurring & for noise reduction.

Blurring is used in pre-processing tasks, such as rem

* LOW PASS FILTER (OR) AVERAGE MASKING :-

It is a linear filter.

In this the value of every pixel in an image is replaced by the average of intensity levels in the local neighborhood.

The size of the neighborhood controls the amount of filtering.

In this masking, all intensities are same.

The general form of Average masking is

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

Eg :- Let us consider a 3x3 low pass spatial mask

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{3\times 3}$$

Then its general form is taken as

$$g(x,y) = \frac{1}{9} \sum_{s,t=-1} \sum f(x+s, y+t)$$

Advantages :-

1. By applying low pass filter noise gets reduced
2. By applying this masking Image gets smoothed.

Dis-advantages :-

1. Average masking leads to blurring of edges; which are desirable features of an image.

2. If the average masking operation is applied to an image, which is corrupted by impulse noise then the impulse noise is attenuated & diffused but not removed.

3. A single pixel with a very unrepresentative value can affect the mean value of all the pixels in its neighborhood significantly.

* **WEIGHTED AVERAGE MASKING :-**

It is a linear filter

To prevent blurring at the edges, since edges consist of high pass components, we go for weighted average technique.

In this technique the pixels nearest to the centre are weighted more than the distant pixels. Since the centre pixel has more weight, blurring at edges is reduced.

Hence it is named as weighted average filter, the pixel to be updated is replaced by a sum of nearby pixel.

The general expression is as below

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

Eg:- weighted average image : $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

we can observe, that the centre pixel has more weight than remaining pixels.

Its general form is

$$g(x,y) = \frac{1}{16} \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

Advantages :-

1. Blurring at sharp edges gets reduced.
2. Noise gets reduced.
3. Image gets smoothed.

* MEDIAN FILTER :-

It is a non-linear technique.

Median filters provide excellent noise reduction capabilities than Linear smoothing filters.

Median filters are used to reduce salt-and pepper noise (impulse noise). A median filter smoothens the image by utilising the median of neighborhood.

Median filter perform the following tasks to find each pixel in the processed image.

1. All pixels in the neighborhood of the original

image are obtained by arranging them in ascending (or) descending order.

2. The median of the sorted value is computed and is choosen as the pixel value of processed image

Eg :- Compute the median value of the marked pixel shown below using 3x3 mask.

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

Solution :- The median value of marked pixel is computed as follows :

Step-1 :- First the pixel values are arranged in ascending order

1 1 2 2 3 4 5 6 7.

Step-2 :- The median value of the ordered pixel is computed as follows :

$\cancel{1}$ $\cancel{1}$ $\cancel{2}$ $\cancel{2}$ 3 $\cancel{4}$ $\cancel{5}$ $\cancel{6}$ $\cancel{7}$

Median value = 3.

Now the original pixel value 4 is replaced by the computed median value 3.

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

original image   After median filtering.

SPATIAL SHARPENING FILTERS :-

The main objective of sharpening is to highlight transitions in intensity.

* FIRST ORDER DERIVATIVE (OR) GRADIENT MASKING (OR) PRE-WITT MASKING

* Image differentiation enhances edges & other discontinuities & de-emphasizes areas with slowly varying intensities.

By using Gradient masking we find out the vertical & horizontal thick values only.

Gradient function, $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$

$$f(x,y) = \left[\left|\frac{\partial f}{\partial x}\right|^2 + \left|\frac{\partial f}{\partial y}\right|^2 \right]$$

* Let us consider an image,

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Differentiation is nothing but difference b/w previous & present images.

$$\frac{\partial f}{\partial x} = w_7 - \cancel{w_4} + w_8 - \cancel{w_5} + w_9 - \cancel{w_6} + \cancel{w_4} - w_1 + \cancel{w_5} - w_2 + \cancel{w_6} - w_3$$

$$= w_7 + w_8 + w_9 - (w_1 + w_2 + w_3)$$

$$\therefore \frac{\partial f}{\partial x} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{\partial f}{\partial y} = w_3 - \cancel{w_2} + w_6 - \cancel{w_5} + w_9 - \cancel{w_8} + \cancel{w_2} - w_1 + \cancel{w_5} - w_4 + \cancel{w_8} - w_7$$

$$= w_3 + w_6 + w_9 - (w_1 + w_4 + w_7)$$

$$\therefore \frac{\partial f}{\partial y} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

From the definition, one-dimensional function $f(x)$ is

- In x-direction, $\frac{\partial f}{\partial x} = f(x+1) - f(x)$

In y-direction, $\frac{\partial f}{\partial y} = f(y+1) - f(y)$

- Two-dimensional function $f(x,y)$ is

In x-direction, $\frac{\partial f}{\partial x} = f(x+1,y) - f(x,y)$

In y-direction, $\frac{\partial f}{\partial y} = f(x,y+1) - f(x,y)$

$$\therefore \nabla f(x,y) = f(x+1,y) + f(x,y+1) - 2f(x,y).$$

Eg:- Take an image as

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

Its equivalent is

$$\begin{pmatrix} 0 & 0 & 0 & & \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 5 & 6 & 7 & 8 \\ & 1 & 2 & 3 & 4 \end{pmatrix}$$

$$\Rightarrow \text{abs}\left[\Sigma\begin{bmatrix}0&0&0\\0&1&2\\0&5&6\end{bmatrix} * \begin{bmatrix}-1&-1&-1\\0&0&0\\1&1&1\end{bmatrix} + \Sigma\begin{bmatrix}0&0&0\\0&1&2\\0&5&6\end{bmatrix} * \begin{bmatrix}-1&0&1\\-1&0&1\\-1&0&1\end{bmatrix}\right]$$

$$\Rightarrow \begin{bmatrix}0&0&0\\2&2&2\\6&6&6\end{bmatrix} + \begin{bmatrix}0&0&0\\-3&0&3\\-11&0&11\end{bmatrix}$$

$$\Rightarrow 24+0 = \underline{24}$$ [Negative values are not taken]

* II-ORDER DERIVATIVE (OR) LAPLACIAN MASKING (OR) HIGH-PASS FILTER MASKING :-

By using II order we find thin lines of an image.

In this if one part gets highlighted then other parts are neglected. Usually centre part may be highlighted (or) dimmed than other pixel values.

Laplacian function, $\nabla^2 f = \dfrac{\partial^2 f}{\partial x^2} + \dfrac{\partial^2 f}{\partial y^2}$

Let the image is a 2-D image, then

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

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

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

Eg :-

(i) $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ (ii) $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ (iii) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ (iv) $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Advantage :-

By applying Laplacian masking, brightness increases once brightness increases we can easily identify the edges & boundaries of image.

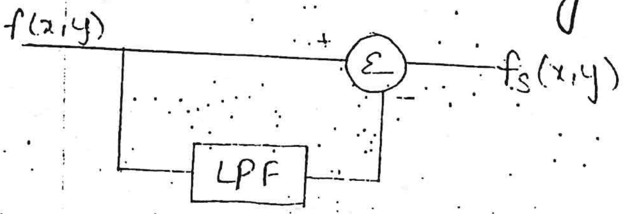
Dis-advantage :-

Because of Laplacian masking noise gets increased

* UNSHARP MASKING :-

The main objective of 'Unsharp masking' is to increase the contrast of an image.

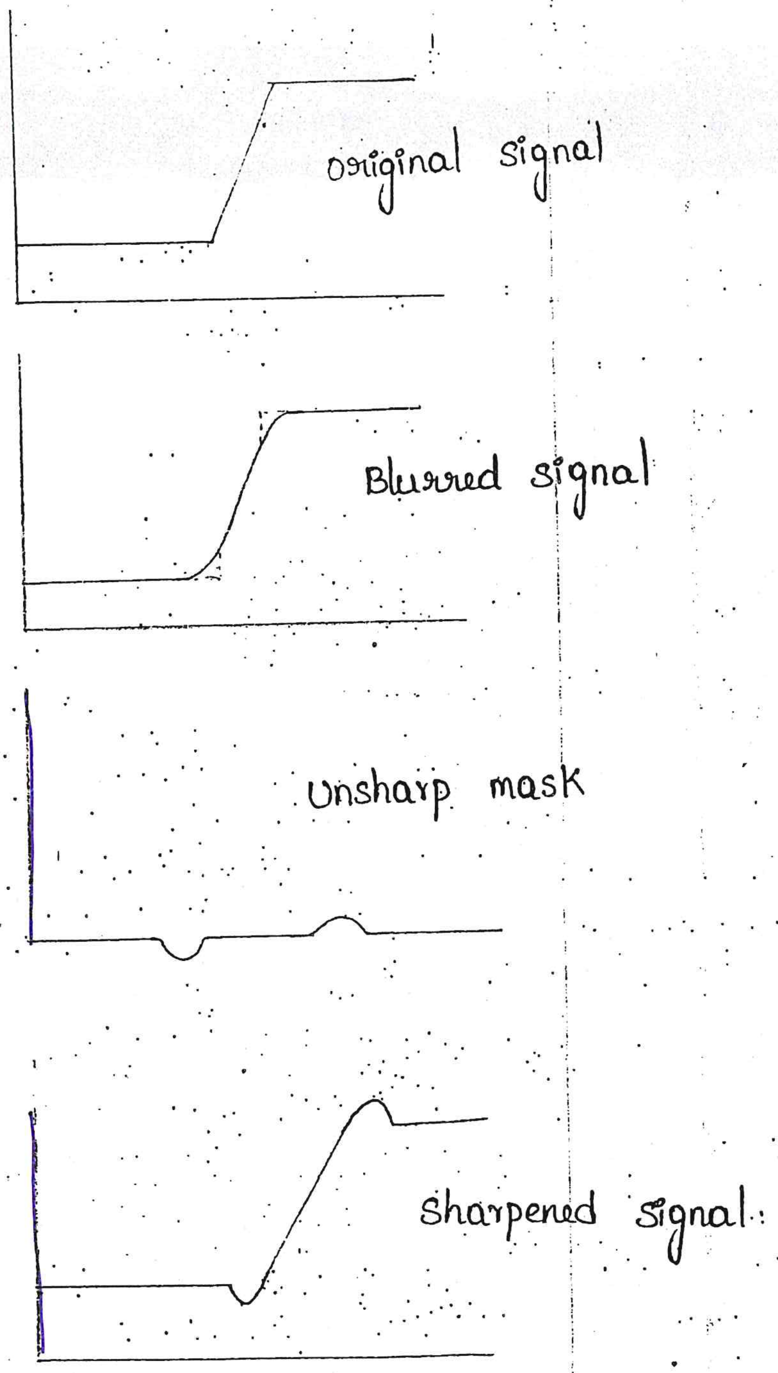
The brightness can be increased by reducing the low pass components & enhancing high pass components



o/p $f_s(x,y) = f(x,y) - f_{LP}(x,y)$

Unsharp masking involves the following steps:

1. Blurring the original image.
2. Subtracting the blurred image from original. image and add masking to the original image.



* **HIGH BOOST FILTERING :-**

For sharpening the image & to increase the centre pixel value we go for high boost filtering.

we know that

$$f_s(x,y) = A\, f(x,y) - f_{LP}(x,y)$$

$$= A\, f(x,y) + f(x,y) - f(x,y) - f_{LP}(x,y)$$

[Adding & Subtracting $f(x,y)$]

$$= (A-1)\, f(x,y) + f_s(x,y)$$

For the purpose of sharping we use Laplacian transform which takes the form

$$f_s(x,y) = (A-1)\, f(x,y) + \nabla^2 f$$

$\hookrightarrow$ Laplacian function.

Eg:- The examples for high boost filtering are:

(i) $\begin{bmatrix} 0 & -1 & 0 \\ -1 & A+4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ (ii) $\begin{bmatrix} 0 & 1 & 0 \\ 1 & A-4 & \phi \\ 0 & 1 & 0 \end{bmatrix}$ (iii) $\begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ (iv) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & A-8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

* HOMO-MORPHIC FILTERING :-

Image is a combination of illumination & reflectance i.e.,

$$f(x,y) = i(x,y) \cdot r(x,y) \quad \text{——} \; ①$$

Reflection term contains high pass components & Illumination term contains low pass components

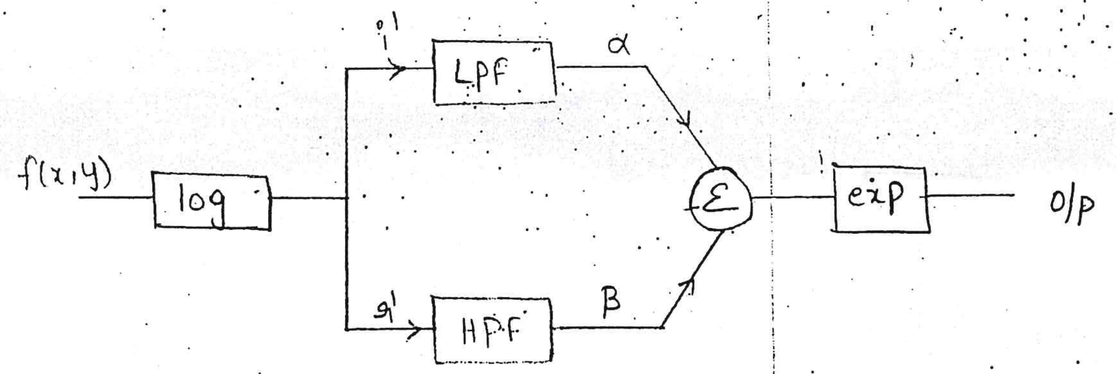
Inorder to seperate low pass & high pass components we have to apply logarithm.

① becomes

$$\log(i(x,y) \cdot r(x,y)) = \log(i(x,y)) + \log(r(x,y))$$

$$\Rightarrow \quad f'(x,y) = i'(x,y) + r'(x,y) \quad \text{——} \; ②$$

Now $i'(x,y)$ is given to LPF & $r'(x,y)$ is given to HPF. These terms (i' & r') are multiplied with α & β



NOW ② becomes,

$$f'(x,y) = \alpha\, i'(x,y) + \beta\, r'(x,y)$$

By applying exponential, we get

$$f'(x,y) = e^{\alpha i'(x,y) + \beta r'(x,y)}$$

$$= e^{\alpha \log(i(x,y)) + \beta \log(r(x,y))}$$

$$= e^{\log i^{\alpha}_{(x,y)} + \log r^{\beta}_{(x,y)}}$$

$[\because m \log a = \log a^m]$

$$= e^{\log\left[i^{\alpha}_{(x,y)} \cdot r^{\beta}_{(x,y)}\right]}$$

$[\because \log m + \log n = \log mn]$

$$\therefore f'(x,y) = i^{\alpha}_{(x,y)} \cdot r^{\beta}_{(x,y)}$$

f(x,y) → log → FFT → IFT → o/p

This gives about Homo-morphic technique.

* SOBEAL MASKING :-

By using sobeal masking sharp edges can be found. It is also similar to Gradient filter but the centre part is doubled.

From the equations of I-order derivative, $\frac{\partial f}{\partial x}$ & $\frac{\partial f}{\partial y}$ becomes

$$\frac{\partial f}{\partial x} = (w_7 + 2w_8 + w_9) - (w_1 + 2w_2 + w_3)$$

$$\frac{\partial f}{\partial y} = (w_3 + 2w_6 + w_9) - (w_1 + 2w_4 + w_7)$$

From the above eqn's we can observe that the centre part is doubled.

$$f(x,y) = |(w_7 + 2w_8 + w_9) - (w_1 + 2w_2 + w_3)| + |(w_3 + 2w_6 + w_9) - (w_1 + 2w_4 + w_7)|$$

Eg:- The examples for Sobeal masking are

(i) $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ w.r.t 'x' (ii) $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ w.r.t 'y'

* ROBERT MASKING :-

This masking is also known as Gradient (or) First order filter.

In this masking we take the cross differences.

Let us consider an image $\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$

Here $f(x,y) = \left[(w_9 - w_5)^2 + (w_8 - w_6)^2\right]^{1/2}$.

* By applying Robert masking we can find out the diagonal values i.e., 45^0 & -45^0

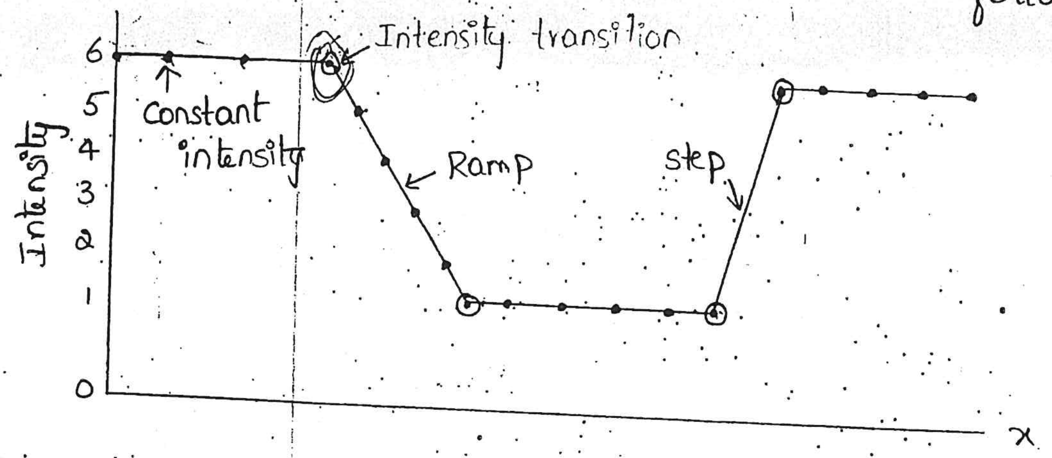
Eg :- The examples for Robert masking are

(i) $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ (ii) $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

* DIFFERENCES b/w I - ORDER & II - ORDER DERIVATIONS :-

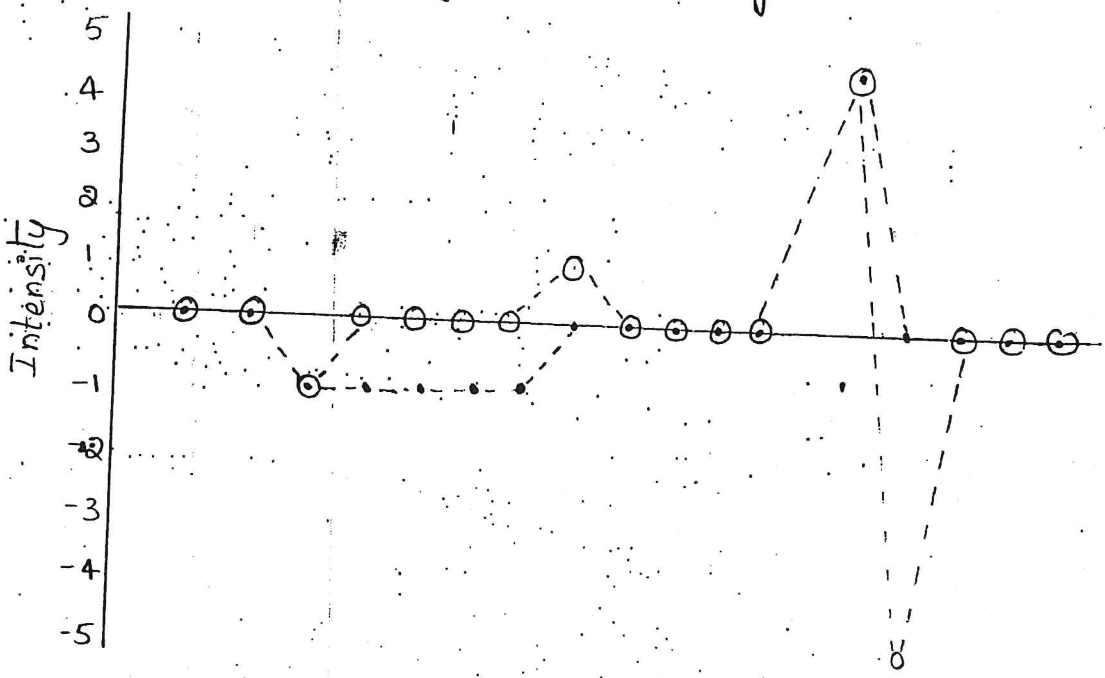
I - order derivative	II - order derivative.
1. The first order derivative of one dimensional function is $\frac{\partial f}{\partial x} = f(x+1) - f(x)$.	1. The II - order derivative of 1-D function f(x) is $\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$.
2. In areas of constant intensity the I - order derivative value is zero.	2. It is also zero at constant areas.
3. For unit - step function its value is non - zero.	3. II - order value is also non - zero at unit step function.
4. Along ramp functions also its value is non - zero.	4. This derivative value must be zero along ramp functions.
5. For isolation case, the value of I - order derivative is its peak value.	6. The II - order derivative value is doubled for isolation case.

Eg:- To get a clear view about the differences b/w I & II order derivations consider the following



Scan line	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
I- derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	5	0	0	0	0
II derivative	0	0	-1	0	0	0	0	1	0	0	0	0	5	-5	0	0	0

The corresponding o/p image is obtained as



• → I - derivative

o → II - derivative

⊙ → same values for both derivative.

* COMBINING SPATIAL ENHANCEMENT METHODS :-

We know that, to obtain a task we require applications of several complementary techniques inorder to achieve an acceptable result. The main objective of combining spatial enhancement method is to enhance the image by sharpening it by combining various techniques.

We use Laplacian to highlight the prominent edges & to increase the dynamic range of the intensity level we use intensity transformation.

Median filter is used to reduce noise. However Median filtering is a non-linear process capable of removing image features, this is unacceptable in medical image processing.

The gradient has a stronger response in areas of ramp & step functions. The Laplacian function produces higher noise than Gradient. The noise can be further lowered by smoothing the gradient with averaging filter. By using Sobeal masking we can sharp the edges of an image. The smallest possible value of gradient image is zero.

By using the product of Laplacian & Smoothed gradient we can increase the sharpness of the image. This type of improvement would not have been possible by using the Laplacian or gradient alone.

The dynamic range can be sharpened by using power law transformation. Histogram equalisation is not suitable for this purpose since it has dark image distributions. For this case it is better to use Histogram specification.

Eg:- These techniques are found in printing industry, in image based product inspection, in forensics, in microscopy, in surveillance.

Basics of filtering in the frequency domain:

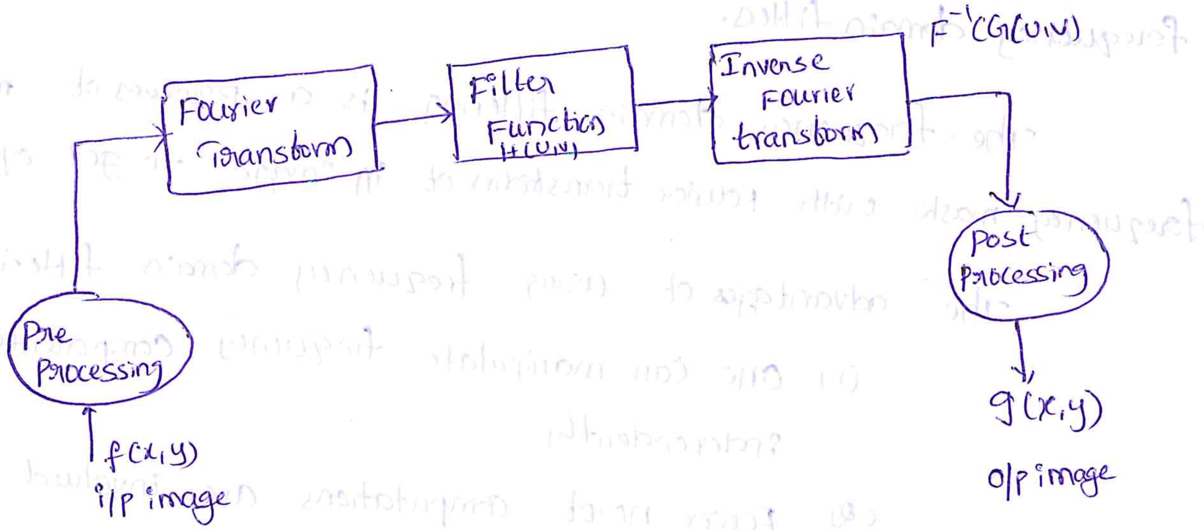


fig: Basic steps for filtering in frequency domain

The algorithm for frequency domain filtering is given as

(1) Let f(x,y) be original image for which filtering is required. obtain Fourier transform of image. Read the spectrum and multiply by $(-1)^{x+y}$ to centre the transform

(2) Design a frequency domain filter matrix function h(x,y). The mask can be any shape depends on application requirement. obtain Fourier transform of h(x,y) to get H(u,v).

(3) Multiply the Fourier spectrum of the filter with the Fourier spectrum of the image by element wise multiplication

$$G(u,v) = H(u,v) \cdot F(u,v)$$

↑

o/p image in Frequency domain

(4) Apply inverse Fourier transform to G(u,v) to retrieve the filtered image in spatial domain.

(5) Extract real components and multiply by $(-1)^{x+y}$ to offset the effect in step 1

6. Display the images and exit.

This algorithm is general and used to implement many frequency domain filters.

The frequency domain filtering is a process of multiplying frequency mask with Fourier transform of i/p image to get o/p image.

The advantages of using frequency domain filtering are

(1) one can manipulate frequency components independently

(2) Fewer no. of computations are involved.

The spatial domain filtering is flexible upto 9x9 mask but for larger masks filtering in frequency domain is preffered.

For example:

consider a mask H(D). If all the values of mask are 1 i.e

H(D) = 1. ⇒ It represents zero attenuation where all the frequency components are allowed

H(D) = 0 ⇒ It represents maximum attenuation where all the frequency components are blocked.

By controlling the weights of mask, we can control the attenuation of frequency components.

Image smoothing in Frequency domain:

An ideal Lowpass filter which allows the frequencies up to a certain cutoff frequency and removes all frequencies beyond that, then the transfer function is given as

$$H(D) = \begin{cases} 1 & \text{for } D \leq D_0 \\ 0 & \text{for } D > D_0 \end{cases}$$

By multiply F(D) in 1D preserves with H(D) which preserves the frequencies upto D_0

Similarly highpassfilter which allows the frequencies more than cutoff frequencies and removes all the frequencies below it, then the transfer function is given as

$$H(D) = \begin{cases} 1 & \text{for } D > D_0 \\ 0 & \text{for } D \leq D_0 \end{cases}$$

2D image:

In general images are two Dimensional. Here the transfer function should be applied first along the rows (M) of image and the results should be stored in intermediate image. Then H(D) applied to columns of intermediate image to yield a 2D mask.

A more effective approach is to use a single filter and apply radially along the frequency range of image. Some of the masks are

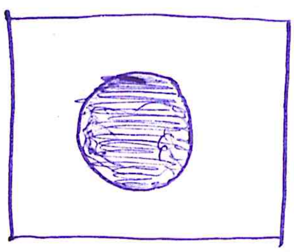


fig: low pass filter circular mask

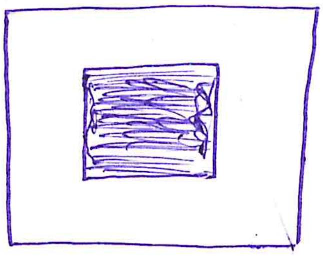


fig: square mask

The masks can be rectangular, Circular & any shape

The centre of frequency rectangle is

$$(u,v) = \left(\frac{M}{2}, \frac{N}{2}\right)$$

The radial frequency

$$D(u,v) = \left[\left(u-\frac{M}{2}\right)^2 + \left(v-\frac{M}{2}\right)^2\right]^{\frac{1}{2}}$$

In 2D the radial cutoff frequency is D_0 and it is specified interms of pixels. For circular mask the cutoff freq is the radius of circle.

For 2D image

The transfer function of lowpass filter is

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

The transferfunction for a highpass filter is

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

→ The ideal lowpass filter produces ringing effect which is also known as Gibbs ringing. i.e decreasing the intensities in parallel to edges.

To overcome this effect use

→ Gaussian lowpass filters

→ Butterworth lowpass filters

The transferfunction on Gaussian filter mask is

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

where D_0: cutoff frequency

The values of mask changes from 0 to 1. The Gaussian mask is controlled by σ, as the value of σ changes the cut off frequency changed. The Gaussian filter never cause ringing artefacts

The transferfunction on Butterworth filter mask is

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

$n \rightarrow$ order of filter

$D_0 \rightarrow$ cutoff frequency

$H : 0 \rightarrow 1$ mask magnitude

As n value increases the filter becomes sharper with increased in ringing artefacts.

if $n=0$ No ringing effect

$n=2$ Small amount of ringing present.

Image sharpening in frequency domain:

High pass filter equivalents are used to attenuate the low frequency components and allows high frequency components such as edges, boundaries and other abrupt changes of image.

The transfer function of high pass filter is

$$H_{hp}(u,v) = 1 - H_{lp}(u,v).$$

where $H_{lp}(u,v)$ = transfer function of LPF.

✓ High pass filters doesn't have ringing effects because it eliminates the zero (DC) components.

The transfer function for High pass Gaussian filter is

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

The transfer function for high pass Butterworth filter is

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

$n \rightarrow$ order of filter that gives sharpness of cut off value

→ Frequency emphasis filter is used for image sharpening this filter emphasizes frequencies by adding a portion of high frequencies to the image. It is given as

$$g(x,y) = IFFT\left[\left[1+K(1-H_{LP}(U,V))\right] F(U,V)\right]$$

$1+K(1-H_{LP}(U,V))$ is a term called as high freq emphasis filter.

The Parameter k controls the proportion of high frequencies in the image. The most general form of filter is

$$g(x,y) = IFFT\{[(k_1+k_2) H_{LP}(u,v)] F(u,v)\}$$

k_1 controls offset
k_2 controls contribution of high frequencies.

Selective Filtering:

Selective filters are allows & blocks the frequency components within its range. Some of those are

1. Band pass filters
2. Band stop filters
3. Notch filters.

Band pass filters:

Band pass filters allow frequency components if they fall in the range $D_l - D_h$.

For 1D Band pass filters

$$H(D) = 1 \text{ for } D_l \le D \le D_h$$
$$0 \text{ for } D \geq D_0$$

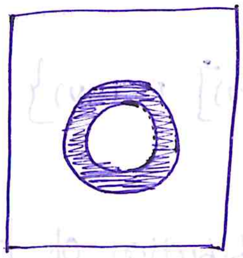
Here D_0 is cutoff frequency

For 2D Band pass filters the transfer function

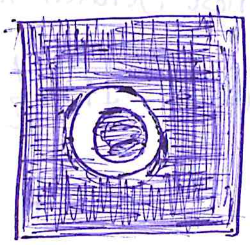
$$H(u,v) = 1 \text{ if } D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2}$$
$$0 \text{ else}$$

Here D_0 is cutoff freq.

$D(u,v)$ is the Distance of the point (u,v) from the centre and W is width of Band



ⓐ Band Pass filter



ⓑ Band reject filter

Frequency Domain mask.

Band reject filters:

The Band reject filter which blocks the frequency components in its range $D_l - D_h$. It is the complement of Band Pass filter.

For 1D, the transfer function is

$$H(D) = \begin{cases} 0 & \text{for } D_l \le D \le D_h \\ 1 & \text{for } \cancel{\text{else}}\ D \ge D_0 \end{cases}$$

For 2D image the transfer function is

$$H(u,v) = \begin{cases} 0 & \text{for } D_0 - \frac{\omega}{2} \le D(u,v) \le D_0 + \frac{\omega}{2} \\ 1 & \text{for else} \end{cases}$$

The transfer function for Band reject filter $= 1 - H_{BP}(u,v)$

Notch filters: A Notch filter is a special form Band reject filters. Instead of removing the entire range of frequencies, it only removes selective frequency components.

It is used to remove periodic noise and ringing effects. and also removes the electrical interference caused by electrical disturbance.

Homomorphic filtering:

This filtering process an image with adequate brightne by simultaneous intensity range compression and contrast enhancement. i.e reducing high intensity values and enhancing dark intensity value at a time.

An image $f(x,y)$ can be expressed as the product of its illumination $i(x,y)$ and reflectance $r(x,y)$ components

$$f(x,y) = i(x,y) \cdot r(x,y). \rightarrow (1)$$

Since $F.T[x \cdot y] \neq F.T[x] \cdot F.T[y]$, so we use logarithm to equation (1) to split the terms

$$\ln(f(x,y)) = \ln(i(x,y) \cdot r(x,y))$$

$$g(x,y) = \ln(i(x,y)) + \ln(r(x,y)) \qquad [\because g(x,y) = \ln(f(x,y))]$$

$$F.T[g(x,y)] = F.T[\ln(i(x,y)) + \ln(r(x,y))]$$

$$G(u,v) = I'(u,v) + R'(u,v) \rightarrow (2)$$

Now by applying filtering mask $H(u,v)$, the o/p image in Frequency domain is

$$G(u,v) = H(u,v) \cdot Z(u,v)$$

$$= H(u,v) [I'(u,v) + R'(u,v)]$$

$$G(u,v) = H(u,v) \cdot I'(u,v) + H(u,v) R'(u,v).$$

The filtered image in spatial domain is obtained by applying inverse fourier transform

$$\Rightarrow IFT[G(U,V)] = IFT[H(U,V) \cdot I'(U,V) + H(U,V) R \ldots$$

$\because H(U,V$

$$g(x,y) = i'(x,y) + r'(x,y)$$

$$g(x,y) = e^{g(x,y)}$$

$$= e^{i'(x,y) + r'(x,y)}$$

$$e^{f(x,y)} = e^{i'(x,y)} \cdot e^{r'(x,y)}$$

$$f_0(x,y) = i_0(x,y) \cdot r_0(x,y)$$

Algorithm for applying homomorphic filters:

1. Apply log transformation to the image i.e

$$\ln(f(x,y)) = \ln(i(x,y)) + \ln(r(x,y))$$

2. Apply Fourier transform to log of these components
3. Design filters seperately for illumination and reflectance components. The transfer function of these components are different
4. Apply inverse Fourier transform to filtered image
5. To offset the logirithm applied in step 1. apply antilog function.

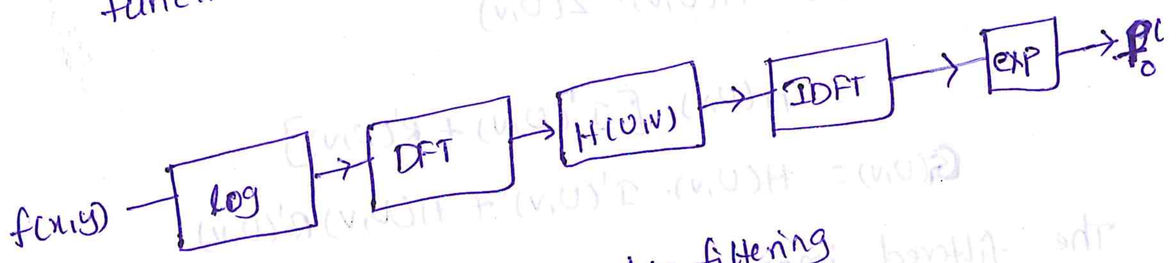


fig: Steps in Homomorphic filtering