# Image Processing [Module - 2]

* Image Enhancement
* Basic Gray level transformation
  - Linear function
  - Logarithmic function
  - Power law function
* Gamma Correction
* Piecewise linear transformation
  - Contrast stretching
  - Gray level slicing
  - Bit plane slicing
* Histogram Processing
  - Histogram Equalization
  - Histogram specification [matching]
* Local Enhancement
* Smoothing Filters

  Linear filters < Averaging, Weighted Average

  Non linear filter < Median, Min/max
* Sharpening Filters
  - Laplacian
  - Unsharp masking and highboost filters
* First order derivative for image sharpening

Radhakrishna. M
Assistant professor
Dept of ECE GAT

* Image enhancement in Frequency domain
  * steps of Filtering
  * smoothing Filters
    - Ideal LPF
    - Butterworth LPF
    - Gaussian LPF
  * Sharpening Filters
    - Ideal HPF
    - Butterworth HPF
    - Gaussian HPF
    - Laplacian
    - Unsharp Masking & highboost filtering
  * Homomorphic filters

* Preliminary concepts
  2D DFT & its properties.

Radhakrishna. M
Assistant Professor
Dept of ECE
GAT, B'lore-98

Image Enhancement → Improves the quality of image so that resulting image is more suitable than the original image for specific application.

Image Enhancement enhances image features such as boundaries, edges or increasing the contrast for analysis.

Image Enhancement can be done using 2 methods

1. spatial domain method → Manipulation is done directly on pixels
2. Frequency domain Method → Manipulation is done on Fourier Transformed image

## Basic Grey level transformation

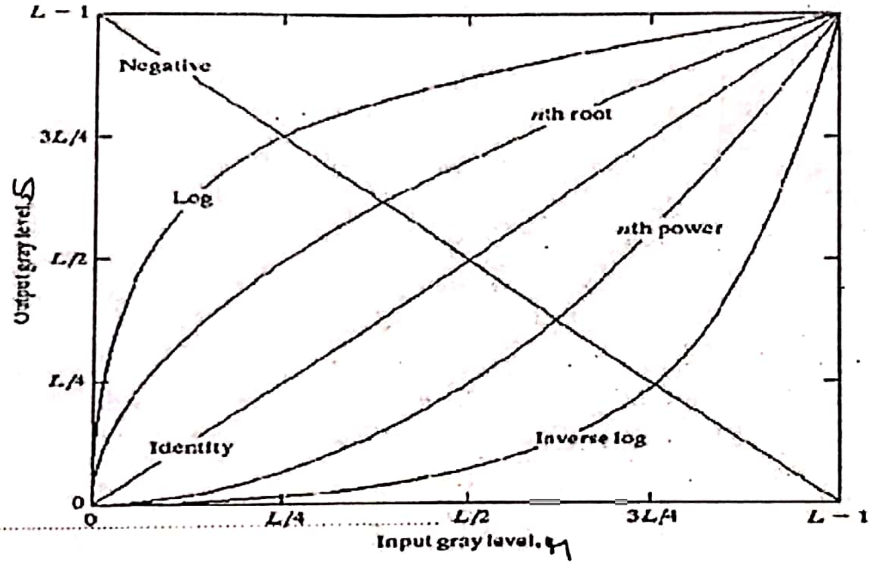



fig:- Gray level transformation functions

$S = T(r)$

Where $s$ = Gray level of $g(x,y)$ (O/P)

$r$ = Gray level of $f(x,y)$ (I/P)

$T$ = Transformation function.

There are 3 basic types of functions used frequently for image enhancement

1) Linear function — Identity transformation. / Negative transformation.

2) Logarithmic function — Log transformation / Inverse log transformation

3) Power law function — $n^{th}$ root transformation / $n^{th}$ Power transformation

## (1) Linear function

### a) Identity transformation

The image with intensity level $[0, L-1]$ is given by $s = r$

i.e after transformation, there is no change in the pixel value.

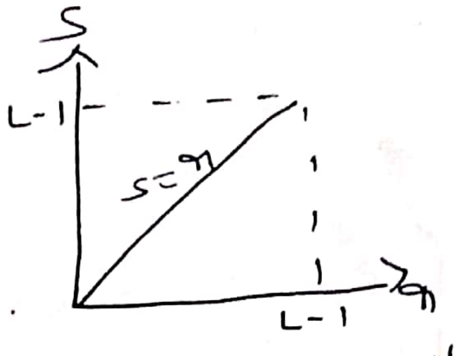



fig: Identity transf'

ex:-

| | | |
|---|---|---|
| 0 | 10 | 50 |
| 5 | 95 | 100 |
| 110 | 150 | 200 |

f(x,y)

$\xrightarrow{s=r}$

| | | |
|---|---|---|
| 0 | 10 | 50 |
| 5 | 95 | 100 |
| 110 | 150 | 200 |

g(x,y)

### b) Negative transformation

The Negative of a digital image is obtained by a transformation function $s = T(r) = (L-1) - r$.

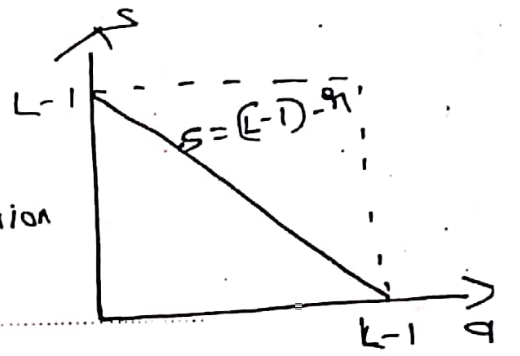



fig: Negative transformati

Where L is the number of Gray levels.

In this transformation, highest gray level is mapped to lowest gray level and vice-versa. Image Negative is useful in numerous applications such as displaying medical images.

Ex:

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

$s = (L-1) - r$
$= (256-1) - r$
$= 255 - r$
$\longrightarrow$

| | | | |
|---|---|---|---|
| 255 | 245 | 205 | 155 |
| 250 | 160 | 105 | 55 |
| 145 | 105 | 165 | 45 |
| 80 | 45 | 0 | 155 |
| 255 | [illegible] | 255 | 0 |

Note that middle grey levels are not changed much ex: [If $r = 110$, $s = 145$], where as dark grey levels is converted to bright & vice versa.

2) Logarithmic functions:-

## Log transformations

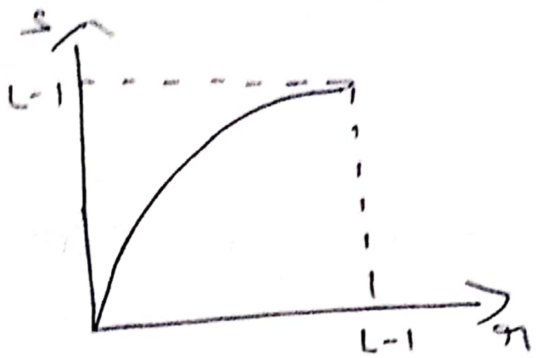



fig: Log transformation

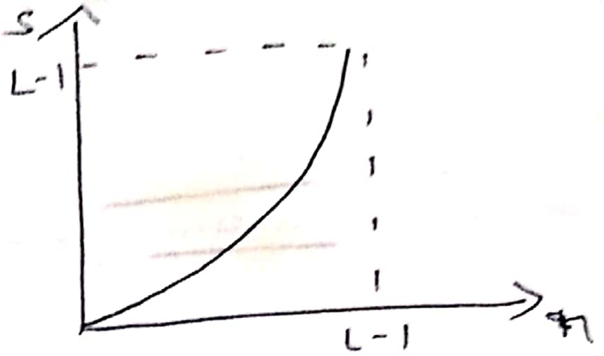



fig: Inverse log transformation

Log transformation is given by

$\boxed{s = c \log(1+r)}$ , where c is constant (c = 1) and $r \geq 0$

ex;

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

$\xrightarrow{s = c\log(1+r)}$

| | | | |
|---|---|---|---|
| 0 | 110 | 181 | 212 |
| 82 | 210 | 231 | 244 |
| 217 | 231 | 242 | 246 |
| 238 | 246 | 255 | 212 |

If c = 1. and r = 0 ⇒ s = 0

If c = 1 and r = 10 ⇒ s = 1.04

If c = 1 and r = 20 ⇒ s = 1.32

If c = 1 and r = 200 ⇒ s = 2.302

If c = 1 and r = 255 ⇒ s = 2.4

Input maximum value is 255. But after log transformati the maximum value is 2.4. Thus the scaling of $\left(\frac{255}{2.4}\right) = 106$ is done, so that maximum value becomes 25

ex: (2.4 × 106 = 255, 2.302 × 106 = 244)

* The shape of log curve in fig (a) shows that this transformation maps a narrow range of low grey leve values in the input image into a wider range of output levels.

* The opposite is true i.e wide range of input is converted to narrow range of output.

* This transformation is used to expand a values of dark pixels in an image while compressing the higher level values. The opposite is true for inverse log transformation.
* The amount of expansion or compression to be done is fixed and cannot be changed. More flexibility is given by power law transformation

3) Power law functions [Power law (Gamma) transformation
Nth root and Nth power transformation

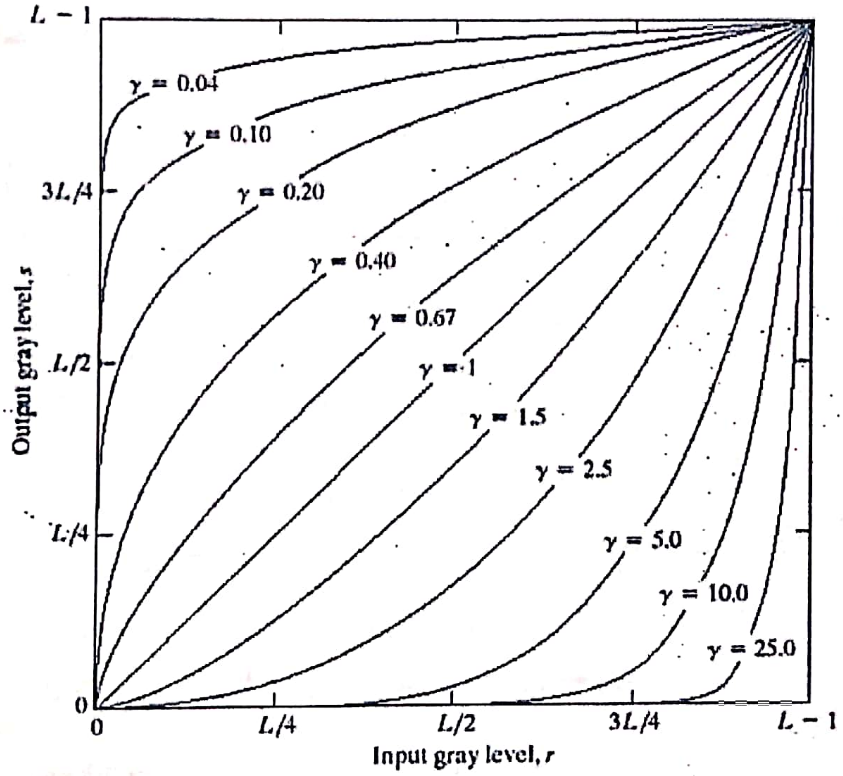



fig:- Plots of equation $s = cr^{\gamma}$ for various values of $\gamma$ $(c=1)$

Power law transformation have the basic form $s = cr^{\gamma}$
where c and $\gamma$ are positive constant.

For fractional values of $\gamma$, the transformation behaves like log transformation i.e narrow range of dark input values is mapped into wider range of output values, with the opposite is being true for higher values of input levels.

Scanned with CamScanner

For large values of $\gamma$ ($\gamma > 1$) the transformation behaves like inverse log transformation. It has opposit effect when $\gamma < 1$. If $c = \gamma = 1$, power law transformation reduces to identity transformation.

| $\gamma$ | $s(\gamma = 2.5)$ | | $s(\gamma = 0.4)$ | |
|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 28 |
| 10 | 316 | 0 | 2.511 | 70 |
| 20 | 1789 | 0 | 3.314 | 92 |
| 30 | 4929 | 1 | 3.896 | 108 |
| 40 | 10119 | 2 | 4.373 | 122 |
| 100 | 100000 | 25 | 6.309 | 175 |
| 210 | 639069 | 157 | 8.489 | 236 |
| 220 | 717888 | 176 | 8.649 | 240 |
| 230 | 802268 | 197 | 8.8042 | 245 |
| 240 | 892335 | 219 | 8.955 | 249 |
| 250 | 988211 | 243 | 9.102 | 253 |
| 255 | 1038365 | 255 | 9.1752 | 255 |

$$\frac{1038365}{255} = 4072 \qquad \frac{255}{9.1752} = 27.8$$

Ex:- $\left[\frac{316}{4072} = 0, \ \frac{1038365}{4072} = 255\right]$ Ex $\left[2.511 \times 27.8 = 70, \ 9.1752 \times 2\right.$ $= 255$

In power law transformation Range compression and expansion is observed.

For $\gamma = 2.5$

$r = [1 \text{ to } 100] \Rightarrow s = [0 \text{ to } 25] \rightarrow$ Range compression.

$r = [210 \text{ to } 255] \Rightarrow s = [157 \text{ to } 255] \rightarrow$ Range expansion.

For $\gamma = 0.4$, reverse is observed

$r = [1 \text{ to } 100] \Rightarrow s = [28 \text{ to } 175] \rightarrow$ Range expansion

$r = [210 \text{ to } 255] \Rightarrow s = [236 \text{ to } 255] \rightarrow$ Range compression.

## Gamma Correction (Application of power law Transformation)

Input image

Gamma corrected image

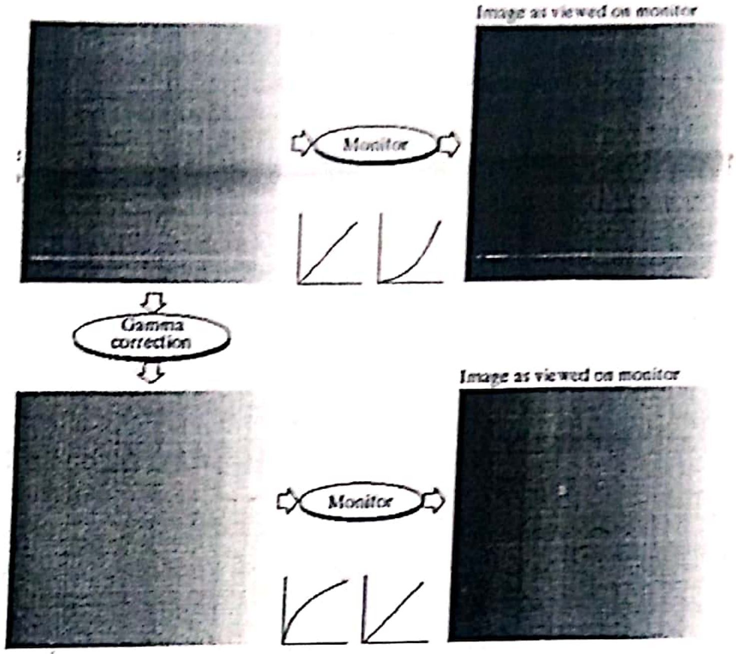



fig:- Gamma Correction

A variety of devices used for image capture, printing and display according to power law. The exponent in power law equation is referred to as $\gamma$. The process used to correct this power law response phenomenon is called gamma correction.

If the value of $\gamma = 2.5$ then such display system would tend to produce an image that are darker than original image. So to avoid this pre-processing of input image is done $(1/2.5 = 0.4)$ before giving it to monitor. Then they produce output that is close to original image. Similar analysis would apply to other imaging devices such as printers, scanner etc.

Ex: For a CRT, $\gamma = 2.5$ shows how gamma correction is implemented on a given image $f(x,y)$ so that it will not appear dark on CRT

$\gamma = 1/2.5 = 0.4$ (Pixel Mapping)

$f(x,y)$

| | | |
|---|---|---|
| 0 | 1 | 2 |
| 100 | 0 | 160 |
| 0 | 127 | 255 |

$g_1(x,y)$

| | | |
|---|---|---|
| 0 | 1 | 1.3 |
| 3.6 | 0 | 7.6 |
| 0 | 6.94 | 9.17 |

$\gamma = 2.5$

$g_2(x,y)$

| | | |
|---|---|---|
| 0 | 1 | 2 |
| 100 | 0 | 160 |
| 0 | 127 | 255 |

## Piecewise linear transformation

In gray level transformation, transformation is applied to whole image. If we required to enhance particular part of image we generally prefer piecewise linear transformation.

Piecewise linear transformation is of 3 types

* contrast stretching.
* Gray level slicing [Intensity level slicing]
* Bit plane slicing.

## Contrast stratching

Low contrast images occur due to bad or non uniform illumination conditions, or due to non linearity of image acquisition devices. Below fig shows an example of low contrast image where the details are lost in background.

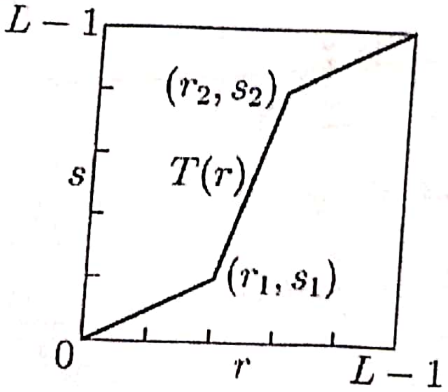



fig :- contrast stretching

(5-5)

Contrast stretching is used to increase the dynamic range of gray level in the image being processed.

In the above fig, Location of the point $(r_1, s_1)$ and $(r_2, s_2)$ control the shape of transformation function.

If $r_1 = s_1$ and $r_2 = s_2$ then the transformation is identity transformation, which produces no change in grey level.

If $r_1 = r_2$ and $s_1 = 0$, $s_2 = L-1$, the transformation becomes thresholding function that creates a binary image (2 grey levels 0 & 1).

Problem

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

| | | | |
|---|---|---|---|
| 0 | 10 | 50 | 100 |
| 5 | 95 | 150 | 200 |
| 110 | 150 | 190 | 210 |

$f(x,y)$

$\xrightarrow{t=128}$

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 255 | 255 |
| 0 | 255 | 255 | 255 |

$g(x,y)$

Gray level slicing

Gray level slicing is used to highlight a specific range of gray level in an image. One approach is to display a high value for all gray levels in the range of interest and low values to all other grey levels

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

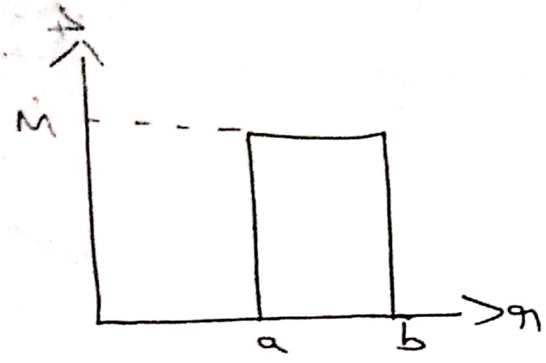



fig ⓐ :- Transformation highlights range A, B grey levels and reducess all other contrast levels.

Ex:

| | | | |
|---|---|---|---|
| 0 | 10 | 50 | 100 |
| 5 | 95 | 150 | 200 |
| 110 | 150 | 190 | 210 |
| 175 | 210 | 225 | 150 |


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

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 225 |
| 0 | 225 | 225 | 0 |
| 225 | 225 | 0 | 0 |
| 0 | 0 | 0 | 225 |

In the below figure grey level in the range of interest is enhanced by preserving all other valu

$$s = \begin{cases} m & A \leq r \leq B. \\ r & \text{otherwise} \end{cases}$$

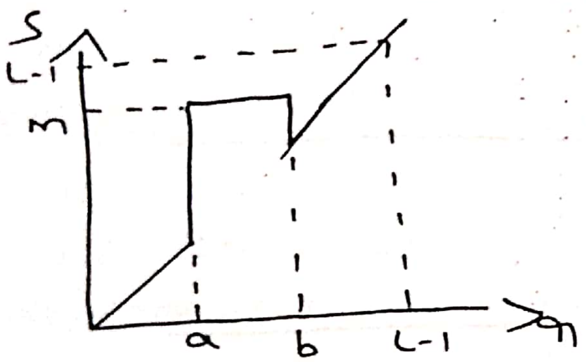



fig ⓑ : Transformation highlights range A, B that preserve all other details.

Ex:

| | | | |
|---|---|---|---|
| 0 | 10 | 50 | 100 |
| 5 | 95 | 150 | 200 |
| 110 | 150 | 190 | 210 |
| 175 | 210 | 225 | 150 |


| | | | |
|---|---|---|---|
| 0 | 10 | 50 | 225 |
| 5 | 225 | 225 | 200 |
| 222 | 225 | 190 | 210 |
| 175 | 210 | 225 | 225 |

# Bit plane slicing

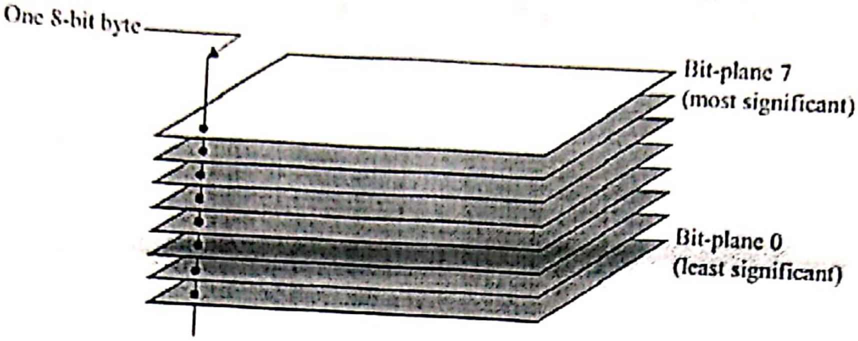



fig: Bit plane slicing

Sometimes it is required to know the contribution of each bit of image. Generally gray scale image consist of 8 bits for pixel representatio i.e each pixel in an image is represented by 8 bit. then the pixel can be divided to 8 planes. (Plane 0 to plane 7).

Plane 0 will has LSB's and Plane 7 will have MSB's seperating the digital image into its 8 bit plane is useful for analysing the relative importance plays by each bit in an image.

Using this approach one can understand the minimum number of bits required to represent the image with desired details, and also it decide the size of image required to represent & store the image.

The higher order bits [Bit plane 4, 5, 6, 7] contains majority of information. The lower order bits [Bit plane 0, 1, 2, 3] contains only small data in an image.

Ex:

| | | |
|---|---|---|
| 6 | 7 | 8 |
| 3 | 2 | 10 |

6 → 00000110
7 → 00000111
8 → 00001000
3 → 00000011
2 → 00000010
10 → 00001010

Bit plane 7 (MSB, leftmost bit) … Bit plane 0 (LSB, rightmost bit)

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 0 |

Bit plane 7

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 0 | 0 |

Bit plane 0

## Problem

* Compute Bit plane slicing for the 8 bit image shown

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

Sol^n:

| | MSB | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|
| 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 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 150 → | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 200 → | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 110 → | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 150 → | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 190 → | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 210 → | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 175 → | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 210 → | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 255 → | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 110 → | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

7th plane (MSB)

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |

6th plane

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |

5th plane

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |

4th plane

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 |

3rd plane

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |

2nd plane

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |

1st plane

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |

0th plane (LSB)

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |

# Histogram Processing

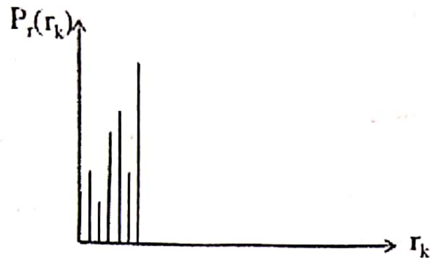



Fig1: dark image


Fig2: light image

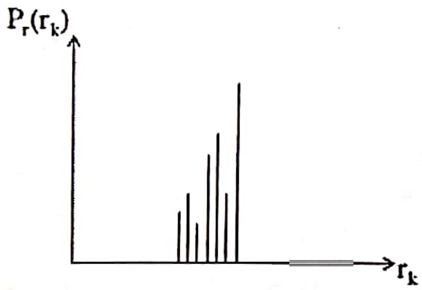



Fig3: low contrast image

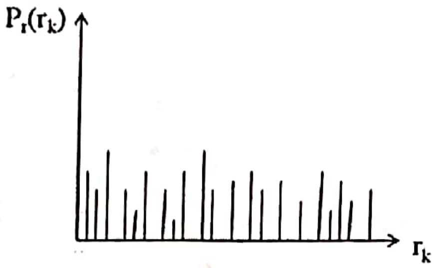



Fig4: high contrast image

**Histogram plots for various kinds of images**

Histogram in an image represents the number of times a particular grey level has occured in a image. It is a graph between various grey levels on x-axis and number of times a grey level has occured in an image on y axis.

Histogram of an image is defined as

$$h(r_k) = n_k \qquad k = 0 \ldots L-1$$

Where $r_k \rightarrow k^{th}$ gray level.

$n_k \rightarrow$ number of pixels in the image having grey level $r_k$.

Histogram manipulation is effectively used for image enhancement

The above fig shows 4 basic grey level char i.e dark, bright, Low contrast & hight contrast.

Dark image → In dark image the histogram are concentrated at the lower end of gray scale.

Bright image → In bright image the histogram are concentrated at the right side of gray scale.

Low contrast image → In this Histogram is narrow and centered towards the middle scale.

High contrast image → In this Histogram covers a broad range of gray levels.

## Normalized Histogram

Normalized Histogram is obtained by dividing the occurance of each pixel having gray level $r_k$ by the total number of pixels in the image.

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

$P(r_k)$ = Probablity of occurrence of gray level $r_k$

$n_k$ = Total number of pixels having gray level $r_k$

$n$ = Total number of pixels.

The sum of all the components of normalized histogram is equal to 1

## Problem

* Find the normalized histogram of image

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 2 | 3 |
| 0 | 2 | 4 | 6 |

Soln:

| $r_k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| $n_k$ | 6 | 1 | 2 | 1 | 1 | 0 | 1 |
| $P(r_k) = n_k/n$ | 6/12 | 1/12 | 2/12 | 1/12 | 1/12 | 0 | 1/12 |

$P(r_k) = \frac{n_k}{n} = \frac{12}{12} = 1$

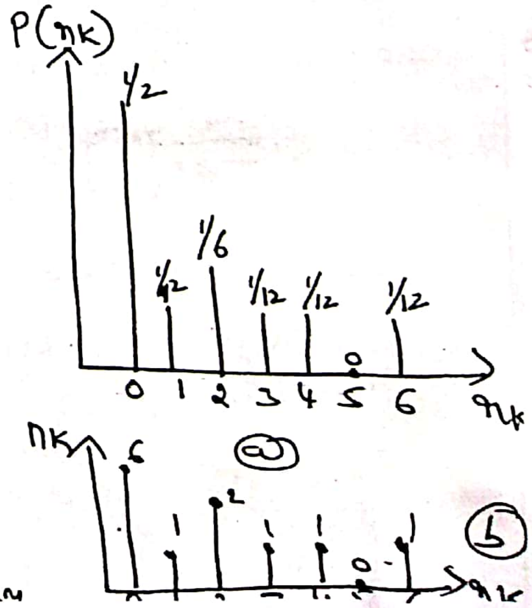
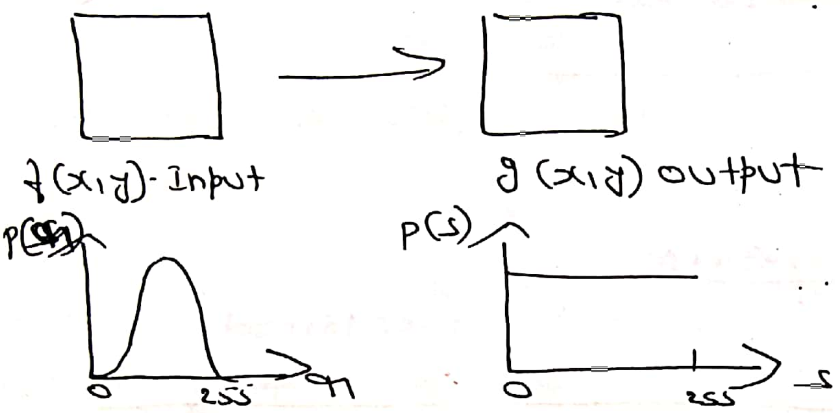



fig: Histogram

# Histogram equalization

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




original image          Histogram equalized image

fig: Histogram equalization

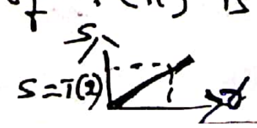
Let us assume that image to be processed has continuous intensity that lies within the interval [0, L-1]
Suppose we divide the image intensity with its maximum value L-1.

Let the variable r represent new gray level in the image, where now $0 \leq r \leq 1$ and let $P_r(r)$ denotes the probablity density function (Pdf) of the variable r. Now we apply the following Transformation function to the r (new gray level)

$$S = T(r) = \int_0^r P_r(w)\,dw \qquad \text{———} \quad (1)$$

Equation (1) should satisfy following conditions.

(i) T(r) is a single valued function (one to one relationship] i.e if $r_2 > r_1 \Rightarrow T(r_2) \geq T(r_1)$
i.e the function of T(r) is increasing with r



(ii) $T(r)$ is a monotonically increasing function in the interval $0 \leq r \leq 1$.


$$s = T(0) = \int_0^0 P_r(w)dw = 0 \text{ and}$$
$$s = T(1) = \int_0^1 P_r(w)dw = 1$$

If the original image has intensities only with a certain range $[r_{min}, r_{max}]$ then

$$s = T(r_{min}) = \int_0^{r_{min}} P_r(w)dw = 0 \text{ and}$$
$$s = T(r_{max}) = \int_0^{r_{max}} P_r(w)dw = 1$$

i.e New intensity $s$ takes always all values within the available range $[0\ 1]$

(iii) For $0 \leq r \leq 1$, $0 \leq T(r) \leq 1$. This guarantees that the output grey levels are in the same of input grey levels.

Proof

The gray levels in an image may be assumed as a random variable in the interval $[0, 1]$

Let $P_r(r)$ and $P_s(s)$ denote the Probablity density function of random variables $r$ and $s$ respectively.

If $P_r(r)$ and $T(r)$ is known and $T^{-1}(s)$ satisfied the condition (ii), then $P_s(s)$ can be obtained by

$$P_s(s) = P_r(r)\left|\frac{dr}{ds}\right| \quad \text{——— (1)}$$

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

$$s = T(r) = \int_0^r P_r(w)dw \quad \text{——— (2)}$$

Where $w$ is a dummy variable.

CDF is an integral of probablity function (always positive). Thus CDF is always single

Valued and monotonically increasing function.
$\therefore$ condition (i) and (ii) is satisfied.

Similarly the integral of probablity density function for variables in the range [0, 1] also in the range [0, 1] so condition (iii) is satisfied.

Differentiating equation (2) w.r.t r

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

$$\boxed{\frac{ds}{dr} = P_r(r)} \quad \text{——— (3)}$$

Substituting eq (3) in equation (1)

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

$$\boxed{P_s(s) = 1} \quad \text{— (4)} \quad \text{where } 0 \le s \le 1$$

$P_s(s)$ is a pdf of output image, whose value is 1 for all values of s. Thus $P_s(s)$ is a uniform pdf.

Discrete transformation function

For discrete values we deal with Probablities and summation instead of PDF and integrals.

The Probablity of occurance of occurence of gray $r_k$ in an image is given by

$$P_r(r_k) = \frac{n_k}{n}, \quad k = 0 \text{----} L-1$$

Transformation is given by

$$s = T(r_k) = \sum_{j=0}^{k} P_r(r_j), \quad k = 0 \text{---} L-1$$

$$\boxed{r_k = \sum_{j=0}^{k} \frac{n_j}{n}} \quad \text{——— (5)}$$

Output image is obtained by mapping each pixel with gray level $r_k$ in the input image to a corresponding pixel with gray level $s_k$ in output image.

## Problem

1) For a given 4x4 image having gray scales between [0 to 9], get histogram equalized image and 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 |

Sol^n:

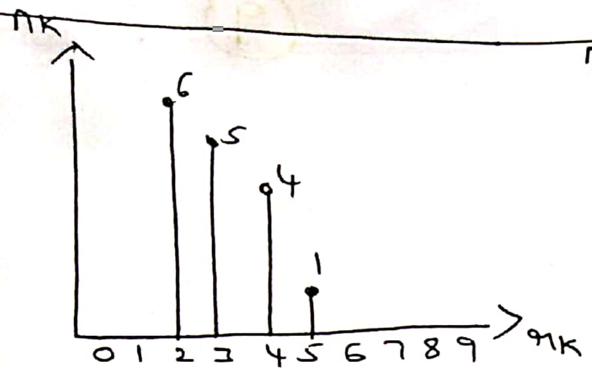
$n$ = Total no of pixels (16)

| Pixel ($r_k$) | No of pixels ($n_j$) | Cummulative sum $\sum_{j=0}^{k} n_j$ | $s = \sum_{j=0}^{k} \frac{n_j}{n}$ | $s \times 9$ | Round of $s_k$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 6 | 6 | 0.375 | 3.375 | 3 |
| 3 | 5 | 11 | 0.68 | 6.18 | 6 |
| 4 | 4 | 15 | 0.93 | 8.43 | 8 |
| 5 | 1 | 16 | 1 | 9 | 9 |
| 6 | 0 | 16 | 1 | 9 | 9 |
| 7 | 0 | 16 | 1 | 9 | 9 |
| 8 | 0 | 16 | 1 | 9 | 9 |
| 9 | 0 | 16 | 1 | 9 | 9 |

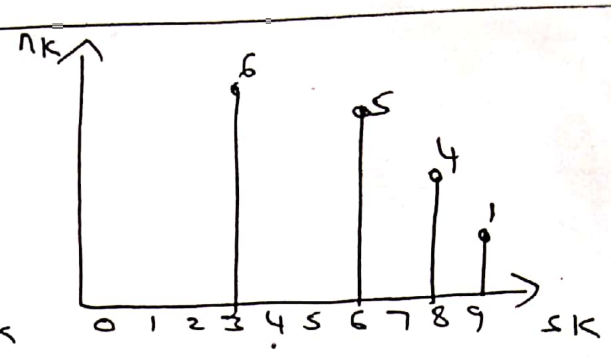
| | | | |
|---|---|---|---|
| 2 | 3 | 3 | 2 |
| 4 | 2 | 4 | 3 |
| 3 | 2 | 3 | 5 |
| 2 | 4 | 2 | 4 |

$\longrightarrow$

| | | | |
|---|---|---|---|
| 3 | 6 | 6 | 3 |
| 8 | 3 | 8 | 6 |
| 6 | 3 | 6 | 9 |
| 3 | 8 | 3 | 8 |




Histogram of input




Histogram of output

② For an image given find histogram equalized image and draw the histogram before & after equalizati

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 3 |
| 3 | 3 | 3 | 3 | 3 |
| 0 | 0 | 0 | 7 | 7 |

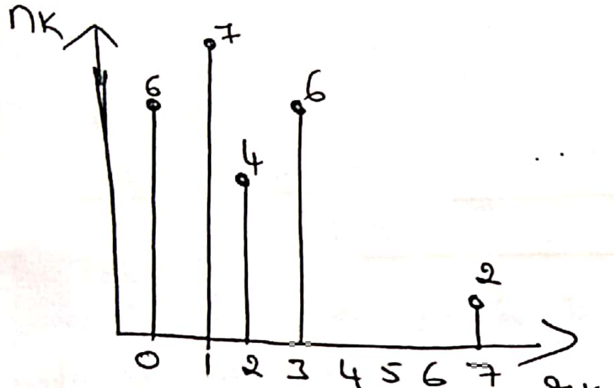
Solⁿ

| Pixels ($r_k$) | $n_k$ | $\sum_{j=0}^{k} n_j$ | $S = \sum_{j=0}^{K} \frac{n_j}{n}$ | $S_k \times 7$ | Round off |
|---|---|---|---|---|---|
| 0 | 6 | 6 | 0.24 | 1.68 | 2 |
| 1 | 7 | 13 | 0.52 | 3.64 | 4 |
| 2 | 4 | 17 | 0.68 | 4.76 | 5 |
| 3 | 6 | 23 | 0.92 | 6.44 | 6 |
| 4 | 0 | 23 | 0.92 | 6.44 | 6 |
| 5 | 0 | 23 | 0.92 | 6.44 | 6 |
| 6 | 0 | 23 | 0.92 | 6.44 | 6 |
| 7 | 2 | 25 | 0.92 | 7 | 7 |

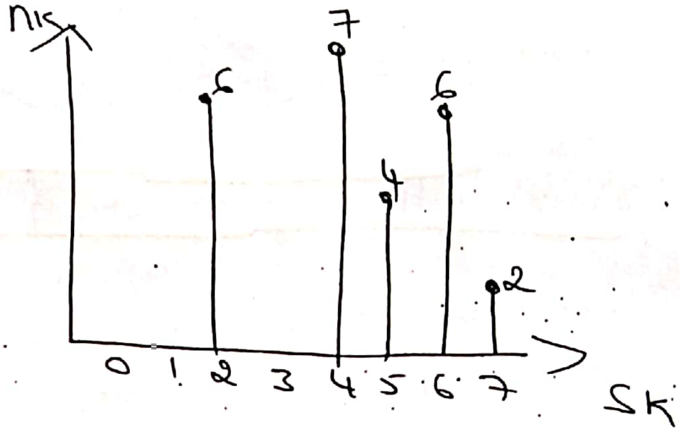
| $r_k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $S_k$ | 2 | 4 | 5 | 6 | 6 | 6 | 6 | 7 |

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

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




Input histogram




Output Histogram

③ Assume an image with a grey level probablity density function $P_r(r)$ as shown in fig ⓐ. obtain equalized histogram for it.

$$P_r(r) = \begin{cases} -2r+2 & 0 \le r \le 1 \\ 0 & \text{otherwise} \end{cases}$$

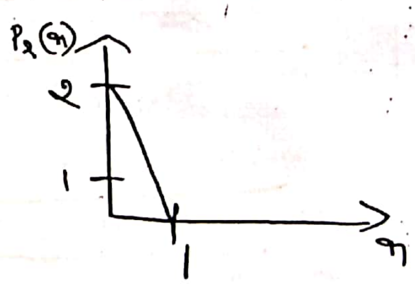



fig ⓐ

Sol$^n$: Transformation function

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

$$= \int_0^r (-2w+2)\,dw$$

$$= \left[\frac{-2w^2}{2} + 2w\right]_0^r$$

$$\boxed{s = -r^2 + 2r}$$

# Histogram specification [Histogram matching]

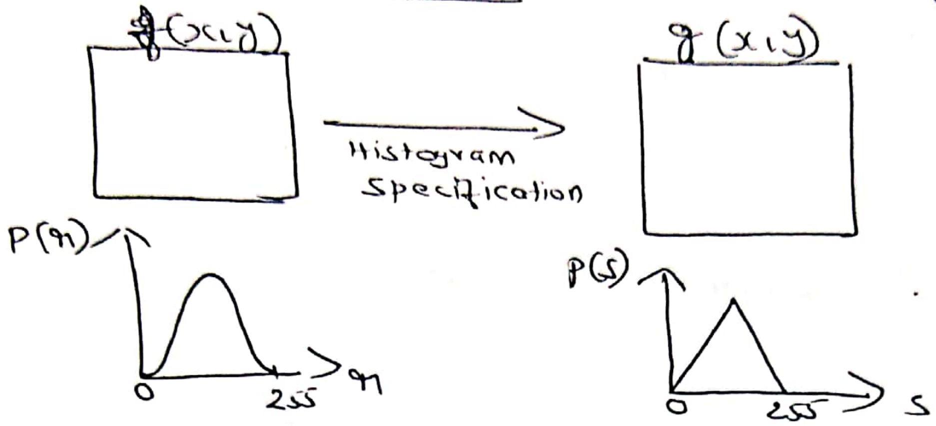



fig: Histogram specification

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.

Suppose we want to specify a particular histogram shape (not uniform) which is capable of highlighting certain grey levels in the image, then histogram specification is performed.

Histogram equalization has a disadvantage that it can generate only one type of output image where the histogram is flat, which may ~~be~~ not be best approach under all circumstances.

In Histogram specification, we can specify the shape of histogram, thus histogram specification gives us flexibility to choose histogram for reference and output image is mapped accordingly.

Let

$P_r(r)$ is Pdf of gray level r of input image

$P_z(z)$ is Pdf of gray level z of specified (desired) image

$P_s(s)$ is Pdf of gray level s of output image

Suppose that histogram equalization is first applied on original image r

$$s = T(r) = \int_0^r P_r(\omega)\,d\omega \quad \text{——— (1)}$$

Suppose that the desired image z is available and histogram equalization is applied

$$V = G(z) = \int_0^z P_z(w)\,dw \quad \text{——— (2)}$$

$P_s(s)$ and $P_z(z)$ are both uniform densities and they can be considered as identical.

Final result of histogram equalization is independent of the density inside the integral, so in eq$^n$ (2) we can use symbol s instead of V.

i.e $G(z)$ is equated to s.

$$G(z) = s = T(r) \quad \text{——— (3)}$$

$$\Rightarrow z = G^{-1}(s) = G^{-1}T(r) \quad \text{——— (4)}$$

Inverse process $z = G^{-1}(s)$ will have desired Pdf

∴ Process of histogram specification can be summarised in following steps.

(i) Take original image and equalize its intensity using relation $s = T(r) = \int_0^r P_r(\omega)\,d\omega$

(ii) From the given Pdf $P_z(z)$, specify pdf $G(z)$

(iii) Apply inverse transformation function

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

## For discrete values

Discrete formulation of equation (4) is given by

$$s_K = T(r_K) = \sum_{j=0}^{K} P_r(r_j) \quad \text{——— (5)}$$

$$= \sum_{j=0}^{K} \frac{n_j}{n}, \quad K = 0, 1, \ldots\ L-1$$

Where n is total number of pixels in the image. $n_j$ is the number of pixels with grey level $r_j$ and L is the number of discrete grey levels

Similarly the discrete formulation is obtained from given histogram $P_z(z_i)$, $i=0,1 \dots L-1$ and has the form

$$V = G(z) = \sum_{j=0}^{K} P_z(z_j) = S_K \qquad K = 0 \dots L-1 \qquad \text{—— (6)}$$

Discrete version of eqn (4) is given by

$$Z_K = G^{-1}[T(r_K)] \qquad K = 0,1,2 \dots L-1 \qquad \text{—— (7)}$$

From eqn (5)

$$Z_K = G^{-1}(S_K) \qquad K = 0,1,2 \dots L-1.$$

Eqn (6) computes transformation function G for a given histogram $P_z(z)$

Finally eqn (7) computes discrete grey levels of the image with that histogram.

Problem

(1) Apply histogram specification on image shown in fig (a)

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 2 |
| 2 | 3 | 3 | 2 |
| 0 | 1 | 0 | 1 |
| 1 | 3 | 2 | 0 |

fig (a)

having $r_i = z_i = 0,1,2,3$

$P_r(r_i) = 0.25$ for $i = 0,1,2,3$

$P_z(z_0) = 0$, $P_z(z_1) = 0.5$

$P_z(z_2) = 0.5$ $P_z(z_3) = 0$

Soln :- Equalize input image histogram

| $r_K$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $P_r(r_K)$ | 0.25 | 0.25 | 0.25 | 0.25 |
| $S_K$ | 0.25 | 0.5 | 0.75 | 1. |

# Equalize specified image histogram

| $z_q$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $P_z(z_q)$ | 0 | 0.5 | 0.5 | 0 |
| $V_q$ | 0 | 0.5 | 1 | 1 |

| $r_k$ | $P_r(r_k)$ | $s_k$ | $z_q$ | $P_z(z_q)$ | $V_q$ | $V^*$ | P |
|---|---|---|---|---|---|---|---|
| 0 | 0.25 | 0.25 | 0 | 0 | 0 | 0.5 | 1 |
| 1 | 0.25 | 0.5 | 1 | 0.5 | 0.5 | 0.5 | 1 |
| 2 | 0.25 | 0.75 | 2 | 0.5 | 1 | 1 | 2 |
| 3 | 0.25 | 1 | 3 | 0 | 1 | 2 | 2 |

a) $q=0, k=0$

$(V_q - s_k) \Rightarrow (V_0 - s_0) = (0-0.25) \geq 0 \Rightarrow$ NO $\Rightarrow$ Increase $q$

$q=1, k=0$

$(V_q - s_k) \Rightarrow (V_1 - s_0) = (0.5-0.25) \geq 0 \Rightarrow$ Yes

$$V_0^* = V_q = V_1 = 0.5$$

$$P_0 = z_q = z_1 = 1$$

b) $q=1, k=1$

$(V_q - s_k) \Rightarrow (V_1 - s_1) = (0.5-0.5) \geq 0 \Rightarrow$ Yes

$$V_1^* = V_q = V_1 = 0.5$$

$$P_1 = z_q = z_1 = 1$$

c) $q=1, k=2$

$(V_q - s_k) \Rightarrow (V_1 - s_2) = (0-0.75) \geq 0 \Rightarrow$ NO $\Rightarrow$ Increase $q$

$q=2, k=2$

$(V_q - s_k) \Rightarrow (V_2 - s_2) = (1-0.75) \geq 0 \Rightarrow$ Yes

$$V_2^* = V_q = V_2 = 1$$

$$P_2 = z_q = z_2 = 2$$

d) $q=2, k=3$

$(V_q - s_k) \Rightarrow (V_2 - s_3) \Rightarrow (1-0.75) \geq 0 \Rightarrow$ Yes

$V_3^* = V_q = V_2 = 1$. $P_3 = z_q = z_2 = 2$

Map input level values output level values

| $r_k$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| P | 1 | 1 | 2 | 2 |

Map input pixels to new values to get new image

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 2 |
| 2 | 3 | 3 | 2 |
| 0 | 1 | 0 | 1 |
| 1 | 3 | 2 | 0 |

$f(x,y)$

$\Longrightarrow$

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 2 |
| 2 | 2 | 2 | 2 |
| 2 | 1 | 1 | 1 |
| 1 | 2 | 2 | 1 |

$g(x,y)$

(2) Perform histogram specification for given data

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

$P(r_i) = 0.2$ for $i = 0 --- 4$

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

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

$P(z_7) = 0$

Soln Equalize input histogram

| $r_k$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| $P(r_k)$ | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| $s_k$ | 0.2 | 0.4 | 0.6 | 0.8 | 1 |

Equalize specified histogram

| $z_q$ | 0 | 2 | 4 | 5 | 7 |
|---|---|---|---|---|---|
| $P(z_q)$ | 0 | 0.2 | 0.4 | 0.4 | 0 |
| $v_q$ | 0 | 0.2 | 0.6 | 1 | 1 |

Find minimum value of q such that $(v_q - s_k) \geq 0$

13

| $r_k$ | $P(r_k)$ | $s_k$ | $z_q$ | $P(z_q)$ | $v_q$ | $v^*$ | $P$ |
|---|---|---|---|---|---|---|---|
| 0 | 0.2 | 0.2 | 0 | 0 | 0 | 0.2 | 2 |
| 1 | 0.2 | 0.4 | 2 | 0.2 | 0.2 | 0.6 | 4 |
| 2 | 0.2 | 0.6 | 4 | 0.4 | 0.6 | 0.6 | 4 |
| 3 | 0.2 | 0.8 | 5 | 0.4 | 1 | 1 | 5 |
| 4 | 0.2 | 1 | 7 | 0 | 1 | 1 | 5 |

a) $q=0,\ k=0$

$(v_q - s_k) = (v_0 - s_0) = (0-0.2) \geq 0 \Rightarrow$ No $\Rightarrow$ Increase $q$

$q=1,\ k=0$

$(v_q - s_k) = (v_1 - s_0) = (0.2-0.2) \geq 0 \Rightarrow$ yes

$v_0^* = v_q = v_1 = 0.2$

$p_0 = z_q = z_1 = 2$

b) $q=1,\ k=1$

$(v_q - s_k) = (v_1 - s_1) = (0.2-0.4) \geq 0 \Rightarrow$ No $\Rightarrow$ Increase $q$

$q=2,\ k=1$

$(v_q - s_k) = (v_2 - s_1) = (0.6-0.4) \geq 0 \Rightarrow$ yes

$v_1^* = v_q = v_2 = 0.6$

$p_1 = z_q = z_2 = 4$

c) $q=2,\ k=2$

$(v_q - s_k) = (v_2 - s_2) = (0.6-0.6) \geq 0 \Rightarrow$ yes

$v_2^* = v_q = v_2 = 0.6$

$p_2 = z_q = z_2 = 4$

d) $q=2,\ k=3$

$(v_q - s_k) = (v_2 - s_3) = (0.6-0.8) \geq 0 \Rightarrow$ No $\Rightarrow$ Increase

$q=3,\ k=3$

$(v_q = s_k) = (v_3 - s_3) = (0.6-0.6) \geq 0 =$ yes

$v_3^* = v_q = v_3 = 1$ and $p_3 = z_q = z_3 = 5$

e) $q = 3,\ k = 4$

$(V_q - s_k) = (V_3 - s_4) \Rightarrow (1-1) \geq 0 \Rightarrow$ yes

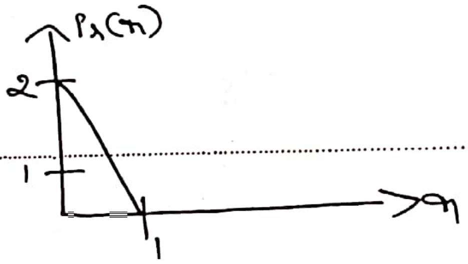
$V_4^* = V_q = V_3 = 1$

$P_4 = z_q = z_3 = 5$

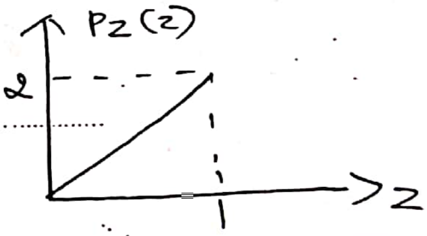
| | | | | | |
|---|---|---|---|---|---|
| $r_k$ | 0 | 1 | 2 | 3 | 4 |
| P | 2 | 4 | 4 | 5 | 5 |

③ Assume an image having given gray level pdf $P_r(r)$. Apply histogram specification with given desired pdf function $P_z(z)$ given below.

$$P_r(r) = \begin{cases} -2r+1 & 0 \leq r \leq 1 \\ 0 & \text{otherwise} \end{cases} \qquad P_z(z) = \begin{cases} 2z & 0 \leq z \leq 1 \\ 0 & \text{otherwise} \end{cases}$$




ⓐ Pdf of r




ⓑ Pdf of z

**Sol^n**: Step 1

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

$$= \int_0^r (-2r+2)\, dr = \left[-r^2 + 2r\right]_0^r = -r^2 + 2r$$

Step 2 (obtain Transformation function $G(z)$)

$$G(z) = \int_0^z P_z(z)\, dz$$

$$= \int_0^z 2z\, dz = \left[z^2\right]_0^z = z^2$$

## Step 3

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

$$-r^2 + 2r = z^2$$

## Step 4

obtain inverse transformation $G^{-1}$

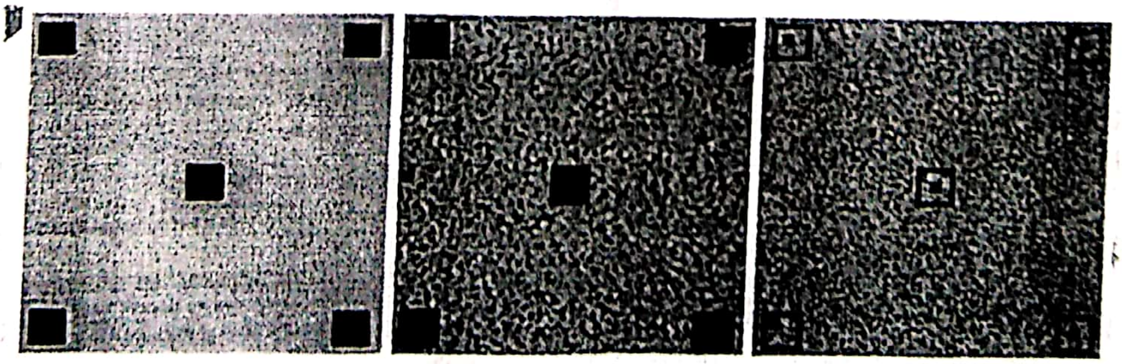
$$z = G^{-1}[T(r)]$$

$$= \sqrt{-r^2 + 2r}$$

## Local Enhancement

Histogram processing methods are global i.e the pixels of entire image are modified by a transformation function. Local enhancement is enhancing detail over the small details in the image. one of the best local enhancement method is local histogram equalization.

To Enhance smaller detail in image, small rectangular or square neighbourhood is chosen with center Moving from pixel to pixel over the entire image every time. In the small neighbourhood histogram is calculated ~~for~~ for the set of pixels, then histogram specification or equalization can be used for enhancement in local area.



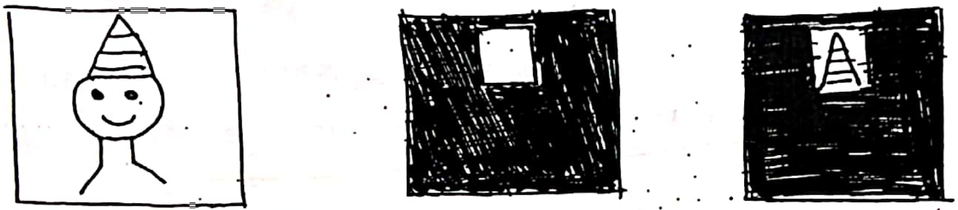
a b c

**FIGURE** [redacted] (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7 × 7 neighborhood about each pixel.

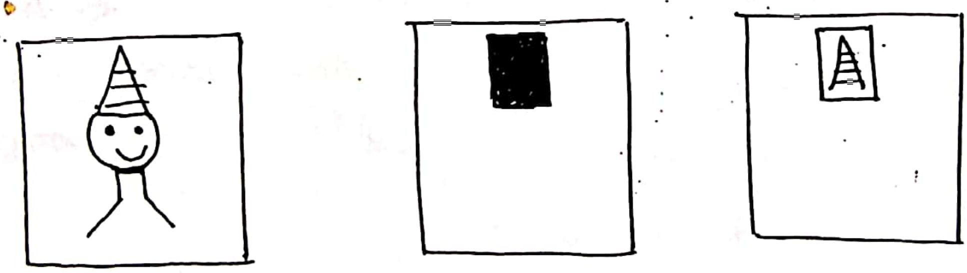
Fig ⓐ shows image in which more detail are not visible. Fig ⓑ shows result of global histogram equalization. In this image contrast is increased little, more details are not visible in dark area. Fig ⓒ is image obtained from local histogram equalization, where all the details in image are visible

## Image Enhancement using Arithmetic and Logical operation

### Logical (Boolean) operation



ⓐ Original image ⓑ AND image mask ⓒ Result of AND operation



ⓓ Original image ⓔ OR image mask ⓕ Result of OR operation.

Arithmetic/Logical operations involving images are performed on a pixel by pixel basis, between 2 or more images. The logical AND and OR is used for masking i.e for selecting sub images in an image.

Fig ⓑ and ⓔ shows AND and OR image mask. Light area represents binary 1 and dark area represents binary 0.

Masking is used only for specific part in an image (Region of interest). In the given image (Above fig) the region of interest is top part of an image

In case of AND mask. If any one of the input is one output follows the other input. If any of the input is zero, the output is always 0 irrespective of other input.

| Mask A | I/p B | output Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Truth table of AND

So AND Mask will pass the image information at light region of Mask, as shown in above fig.

In case of OR mask. If any one of the input is zero, The output follows — other input. If any one of the input is one, output is always one irrespective of other input.

| Mask A | I/P B | O/P Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Truth table of OR

So OR Mask will pass the image information at dark region of mask. as shown in above fig.

## Image subtraction

Image subtraction mainly used in following area

i) Medical - Digital system angiography

First x-ray of patient's body is taken i.e $h(x,y)$. Then a contrast medium is injected into a patient's body (blood stream) and the x-ray is taken after injection. i.e $f(x,y)$

i.e $g(x,y) = f(x,y) - h(x,y)$

usually certain arteries that are not visible in $h(x,y)$ will be bright in output.

Ex.

$$\underset{f(x,y)}{\begin{bmatrix} 0 & 200 \\ 100 & 0 \end{bmatrix}} - \underset{h(x,y)}{\begin{bmatrix} 255 & 0 \\ 0 & 0 \end{bmatrix}} \Rightarrow \underset{g(x,y)}{\begin{bmatrix} -255 & 200 \\ 100 & 0 \end{bmatrix}}$$

In the above example -255 is outside the range, hence we need to scale the output image by adding 255 to every pixel and dividing by 2

$$\underset{g(x,y)}{\begin{bmatrix} \frac{-255+255}{2} & \frac{200+255}{2} \\ \frac{100+255}{2} & \frac{0+255}{2} \end{bmatrix}} \Rightarrow \underset{g(x,y)}{\begin{bmatrix} 0 & 228 \\ \cancel{178} & 128 \end{bmatrix}}$$

178

ii) Motion detection

iii) Automatic checking of industrial parts by subtracting a image of new part.

iv) In automated video surveillance, video camera continuously monitors same area (Bank or store). For any image it acquires, subtraction is done from previously acquired image.

Scanned with CamScanner

# Fundamentals of spatial Filtering

Spatial filtering is one of the tool used in variety of applications such as noise removal, bridging the gaps in object boundaries, sharpening edges etc.

In spatial filtering a mask (rectangle), usually with size of odd length) is made to move on an image. We create a new image where gray level values of pixels are calculated from the values under mask. The values of the mask are modified by a function called filter. If this filter function is a linear function of all gray level values in the mask, then filter is called linear filter, else it is called non linear filters

- Spatial filtering (neighbour hood processing)
  - Smoothing filters
    - Linear filters
      - Averaging filter
      - Weighted averaging filter
    - Non linear filters (order statistics filter)
      - Median filter
      - Max/min filter
  - Sharpening filters
    - Linear filters
      - Laplacian filter
      - 1st derivative filter
      - High boost filter.

# Mechanism of spatial filtering

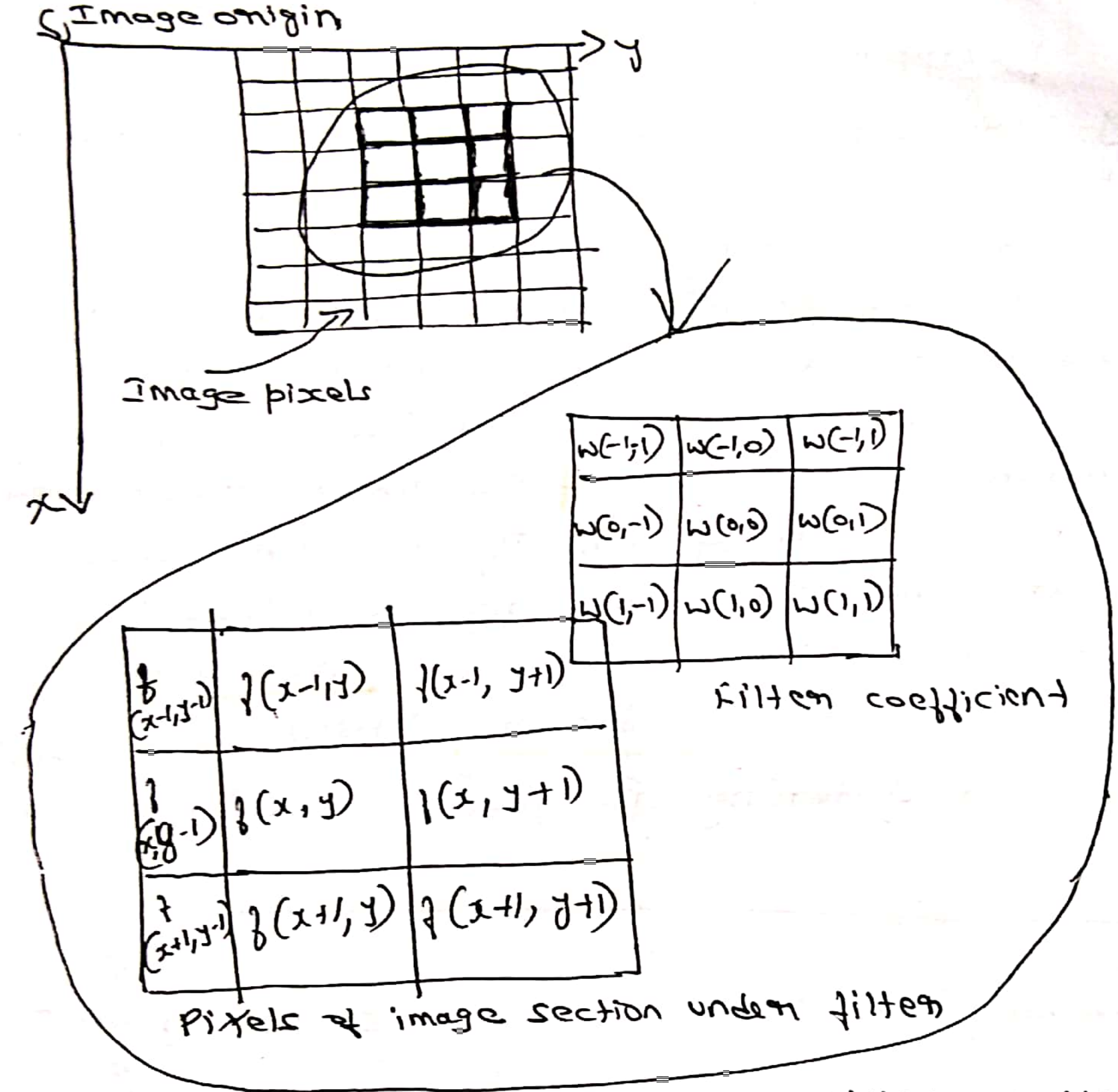



Fig: Spatial Filtering using 3X3 Filter mask

The basic of Spatial filtering is shown in above figure. The filter mask (kernels, templates, windows) is moved from point to point of an image. At any point (x,y) in the image the response g(x,y) of the filter is the sum of products of filter co-efficients and the corresponding image pixels in the area directly under the mask.

$$g(x,y) = w(-1,-1)\, f(x-1, y-1) + w(-1,0)\cdot f(x-1, y) + \cdots\cdots w(0,0)\cdot f(x,y) + \cdots\cdots + w(1,0)\cdot f(x+1, y+1) + w(1,1)\, f(x+1, y+1)$$

For a mask of size $m \times n$, we assume that $m = 2a+1$ and $n = 2b+1$, where a and b are positive integers. In general linear filtering of an image of size $M \times N$ with filter mask of size $m \times n$ is given by

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

Where a and b are integers. It is better to use odd size filter ($3\times3$, $5\times5$, $7\times7$, etc.) because they ~~they~~ have clean center point and easy to locate on the image.

## Spatial correlation and convolution

Correlation is the process of moving a filter mask over the image and computing the sum of products at each location. The Mechanics of convolution are same, except that the filter is first rotated by $180^\circ$

Example:- Spatial correlation [1D correlation of a filter with discrete unit impulse]

a) 0 0 0 1 0 0 0 0 = f and 1 2 3 2 8 = w (filter)

↑ origin

b)

0 0 0 1 0 0 0 0

1 2 3 2 8

c) If filter is of size m, we need m-1 zeros on either side of f

$$\underbrace{0\ 0\ 0\ 0}\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ \underbrace{0\ 0\ 0\ 0}$$

zero padding

1 2 3 2 8

d) Position after one shift

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

1 2 3 2 8

e) Position after four shift

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

1 2 3 2 8

f) Final position

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

1 2 3 2 8

g) Full correlation result

0 0 0 8 2 3 2 1 0 0 0 0

h) Cropped correlation result

0 8 2 3 2 1 0 0

Example :- spatial convolution [illustration of 1D convolution of filter with discrete unit impulse]

a) f = 0 0 0 1 0 0 0 0     w = 1 2 3 2 8

origin

Rotate w by $180^\circ$

then w = 8 2 3 2 1

Scanned with CamScanner

b) starting position alignment

0 0 0 1 0 0 0 0

8 2 3 2 1

c) zero padding

zero padding

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

8 2 3 2 1

d) Position after one shift

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

8 2 3 2 1

e) Position after four shift

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

8 2 3 2 1

f) Final position

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

8 2 3 2 1

g) Full convolution result

0 0 0 1 2 3 2 8 0 0 0 0

h) Cropped Convolution result

0 1 2 3 2 8 0 0

For correlation, of image $f(x,y)$ with a filter $w(x,y)$ of size $m \times n$ is represented as

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

where $a = (m-1)/2$ and $b = (n-1)/2$

In similar way the convolution of $w(x,y)$ and $f(x,y)$ is given by the equation

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

where the minus sign indicates that rotating by $180^\circ$

Below example represents correlation (middle row) and convolution (Last row) of a 2D filter with a 2D discrete unit impulse.

← origin

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \qquad (a)$$

$f(x,y)$ $\qquad$ $w(x,y)$ = Filter.

For the filter of size $m \times n$, pad the image with $m-1$ rows of 0's at top and bottom, and $n-1$ columns of 0's on the left and right [~~Ex~~ above filter is $3 \times 3$ so 2 rows top & bottom and 2 columns up and down]

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad (b)$$

Initial position of w

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 5 | 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 8 | 9 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(c)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 9 | 8 | 7 | 0 | 0 | 0 |
| 0 | 0 | 0 | 6 | 5 | 4 | 0 | 0 | 0 |
| 0 | 0 | 0 | 3 | 2 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(d) ↑ Full correlation result

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 9 | 8 | 7 | 0 |
| 0 | 6 | 5 | 4 | 0 |
| 0 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Cropped correlation result

(e)

Rotated w

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | 8 | 7 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 5 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(7)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 2 | 3 | 0 | 0 | 0 |
| 0 | 0 | 0 | 4 | 5 | 6 | 0 | 0 | 0 |
| 0 | 0 | 0 | 7 | 8 | 9 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(8) Full convolution result

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 2 | 3 | 0 |
| 0 | 4 | 5 | 6 | 0 |
| 0 | 7 | 8 | 9 | 0 |
| 0 | 0 | 0 | 0 | 0 |

(c) → Cropped convolution result

## Vector representation of Linear filtering

(i) Correlation

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

$$= \sum_{k=1}^{mn} w_k z_k$$

$$= w^T z$$

where $R$ = Response of mask

$w_k$ = co-efficients of $m \times n$ filter

$z_k$ = corresponding image encompassed by filter

(ii) Convolution

Rotate the mask by $180^\circ$

The general $3 \times 3$ mask equation is

$$R = w_1 z_1 + w_2 z_2 + \text{---} + w_9 z_9$$

$$= \sum_{k=1}^{9} w_k z_k = \underline{\underline{w^T z}}$$

| | | |
|---|---|---|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

fig: Representation of general $3 \times 3$ filter mask.

# Generating a spatial filter mask



Generating an $m \times n$ linear spatial filter requires specification of $mn$ mask co-efficients. These co-efficients are selected based on what the filter is supposed to do keeping in mind that all we can do with linear filtering is to implement a sum of products.

Assuming that we need to replace the pixels in an image with the average pixel intensities of $3 \times 3$ neighborhood centered on those pixels. If $z_i$ are the intensities, the average is

$$R = \frac{1}{9} \sum_{i=1}^{9} z_i$$

In some applications, we have a continuous function of two variables and the objective is to obtain the spatial filter mask based on that function

ex:- Gaussian function of 2 variables has the basic form

$$h(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where $\sigma$ = standard deviation

$x$ and $y$ are integers

To generate $3 \times 3$ filter mask from this function, we sample it about its center. Thus $w_1 = h(-1,-1)$, $w_2 = h(-1,0)$ ---- $w_9 = h(1,1)$. An $m \times n$ filter mask is generated in similar manner

# Smoothing spatial filters

Smoothing filters are used for blurring and for noise reduction.

Blurring is used in preprocessing task, such as removal of small details from an image prior to object extraction, and to fill the small gaps in lines or curves.

## (a) Smoothing linear filters

The output of smoothing linear filters is simply the average of pixels contained in the neighborhood of the filter mask. These filters are also called averaging filters (or low pass filters).

### (i) Averaging filter

In Averaging filter (mean filter), we replace the value of every pixel in an image by the average/arithmetic mean of gray levels in the neighborhood defined by filter mask.

The advantage of averaging filter is, the very small irrelevant areas of images are removed (as small areas have very small values and if we take further averaging of these areas, then the result will be smaller and negligible). But disadvantage is that, we may loose the edges (sharp information) of an image.

Below figure shows basic 3×3 filter used for averaging

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

fig:- 3×3 filter for averaging

and $R = \frac{1}{9} \sum_{i=1}^{9} z_i$

## (ii) Weighted Averaging Filter

Sometimes it is necessary to protect some information of our choice during averaging for that purpose we use weighted averaging.

In this type of averaging, we multiply the pixels of an image with different coefficients. The pixels having high importance are multiplied by co-efficints of high values in comparision to others. Sometimes we use high values coefficients at center and decreasing the values of coefficients as we go away from the center.

for ex:-

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

Here the sum of all co-efficients of mask is 16. so for averaging, we divide the total values by 16.

Filtering an $M \times N$ image with a weighted averaging filter of size $m \times n$ is given by

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

(5) Non linear smoothing spatial filtering [order statistics filters]

When we require a response of filtering based on the ordering or ranking of pixels, then non linear filters are used. Some of the nonlinear smoothing spatial filtering are

(i) Median filter

It replaces the value of a pixel by the median of the intensity levels in the neighborhood of that pixel.

Median filters provides excellent noise reduction with less blurring than linear smoothing filters of similar size.

In 3X3 neighborhood the median is $5^{th}$ largest value. In 5X5 neighborhood median is $13^{th}$ largest value & so on.

## (ii) Maximum and minimum filtering



Median filter represents 50th percentile of ranked set of pixels. For some purposes, we may use 100th percentile or 0th percentile filter. The filter that is using 100th percentile of ranked set of pixels is called max filter. The filter that is using 0th percentile of ranked set of pixels is called min filter.

Max filter is useful for finding brightest point of an image and minimum filter is useful for finding darkest point of image.

## Sharpening spatial filters

Image sharpening is done to highlight fine details and edges in an image.

Image sharpening is reverse of smoothing.

So smoothing (Blurring) = Average of neighbouring pixels
= Integration of pixels

but

sharpening = Derivative of pixels

Mathematically, derivative can be of

(i) First derivative that can be represented by

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

(ii) second derivative that can be represented by

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

* First derivative (i) should be zero at the area of constant gray levels [also called flat segments of image]

(ii) must be non zero at onset of gray level step or ramp

(iii) must be non zero along ramps

* similarly, the <u>second derivative</u>

(i) must be zero at constant areas

(ii) must be nonzero at the onset and end of an intensity step or ramp.

(iii) must be zero along the ramps of constant slope.

<u>Example</u>

① Find the first and second derivative of given data $f(x)$:

$$f(x) = [4 \quad 3 \quad 2 \quad 5 \quad 9]$$

<u>Sol^n</u> :- First order derivative of a 1D signal $f(x)$ is

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

$$= [3-4, \; 2-3, \; 5-2, \; 9-5]$$

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

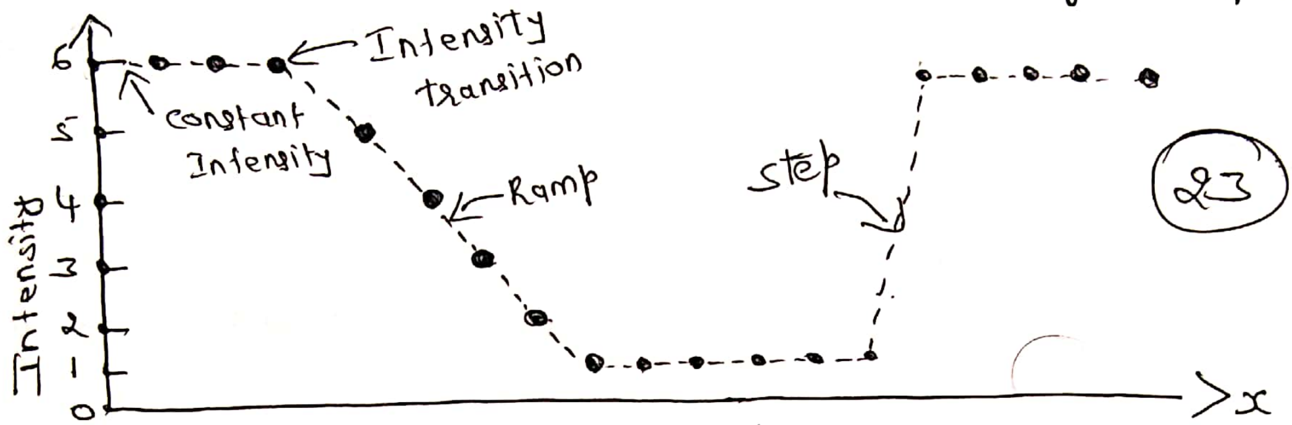
2nd order derivative

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

$$= [2+4-6, \; 5+3-4, \; 2+9-10]$$

$$= [0, 4, 1]$$

To examine the similarities ~~between~~ differences between first and second derivatives of a digital function consider the following example




| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Scan line | 6 | 6 | 6 | 6 | 5 | 4 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 6 | 6 | 6 | 6 | 6 |

1st Derivative 0 0 0 -1 -1 -1 -1 -1 0 0 0 0 0 5 0 0 0 0 → x

2nd Derivative 0 0 -1 0 0 0 0 1 0 0 0 0 5 -5 0 0 0

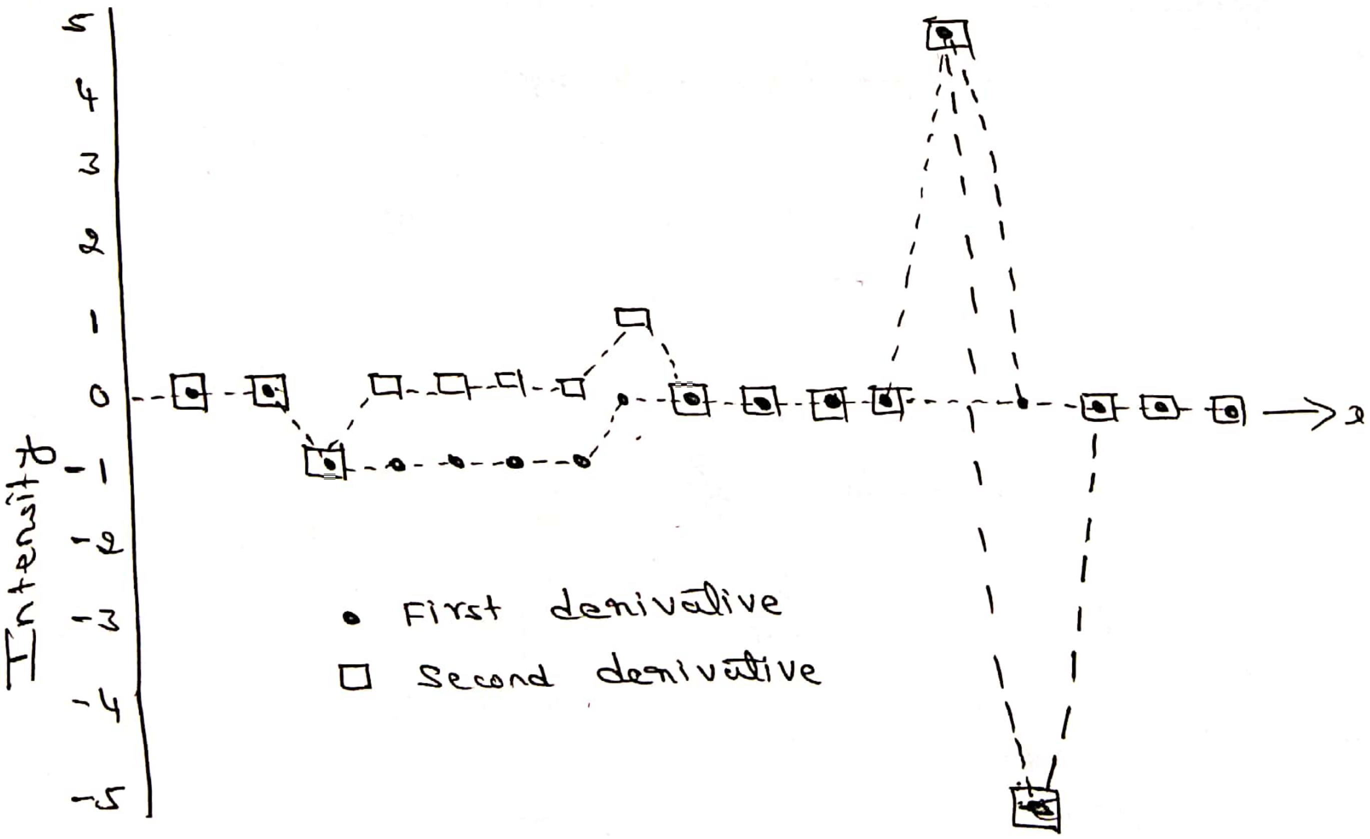



fig:- Illustration of first & second derivatives of 1D digital function

24

## Using the second derivative for image sharpening – The Laplacian

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

Laplacian formulate a 2nd order derivative & construct a filter mask based on formulation

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

Laplacian (Second derivative) is rotation invariant and linear operation and it is defined as

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

In $x$ direction, we have

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

and similarly in $y$ direction we have

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

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

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

The above equation is represented in terms of mask

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

+ve Laplacian

When the diagonals also considered, then

$$\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)$$
$$+ f(x-1,y-1) + f(x+1,y-1) + f(x-1,y+1)$$
$$+ f(x+1,y+1) - 8f(x,y) \quad (24)$$

The mask representation of above equation is

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | -8 | 1 |
| 1 | 1 | 1 |

The other implementation of Laplacian mark (when eqn 1 and 2 are negative)

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

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

Laplacian operator

→ Highlights, gray level discontinuities in an image

→ It deemphasizes regions with slowly varying gray levels

→ Tends to produce an image with grayish edge lines with dark featureless background

The basic way in which we use Laplacian for image sharpening is

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

where $g(x,y)$ = sharpened image

$f(x,y)$ = Input image

$c = -1$ if the center co-efficient of Laplacian mask is -ve

$c = 1$ —"—"— +ve

i.e $g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ \text{or} \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$

In positive Laplacian, the center element of the mask should be -ve and corner elements of mask should be 0 [+ve Laplacian operator is used to take out the outward edges in an image]

In -ve Laplacian, the center element should be positive, all the elements in the corner should be zero and rest of all the elements in the mask should be -1. [-ve Laplacian operator is used to take out the inward edges in an image]

# Unsharp masking and high boost filtering

The sharpened (high pass filtered) image is generated by subtracting blurred (Low pass filtered/smoothed) image from original image

The process of unsharp masking consist of following steps

1. Blur the original image
2. Subtract the blurred image from original (The resulting ~~image~~ difference is called mask)
3. Add the mask to original

$$g_{mask}(x,y) = f(x,y) - \bar{f}(x,y) \quad \text{— (1)}$$

Mask = $g_{mask}(x,y)$

$f(x,y)$ = original image

$\bar{f}(x,y)$ = Blurred image

then we add weighted portion of mask back to original image

$$g(x,y) = f(x,y) + K * g_{mask}(x,y) \quad \text{— (2)}$$

when $K=1$, we have unsharp masking

when $K>1$, the process is called high boost filtering

when $K<1$, it deemphasizes the contribution of unsharp masking.

The sharpened image generated by unsharp masking has average background intensity near to black. This is because sharpening is a high pass filter operation and elliminates dc component (zero frequency). solution to this problem is to add a portion of image to the filtered image. High boost filter does the same.

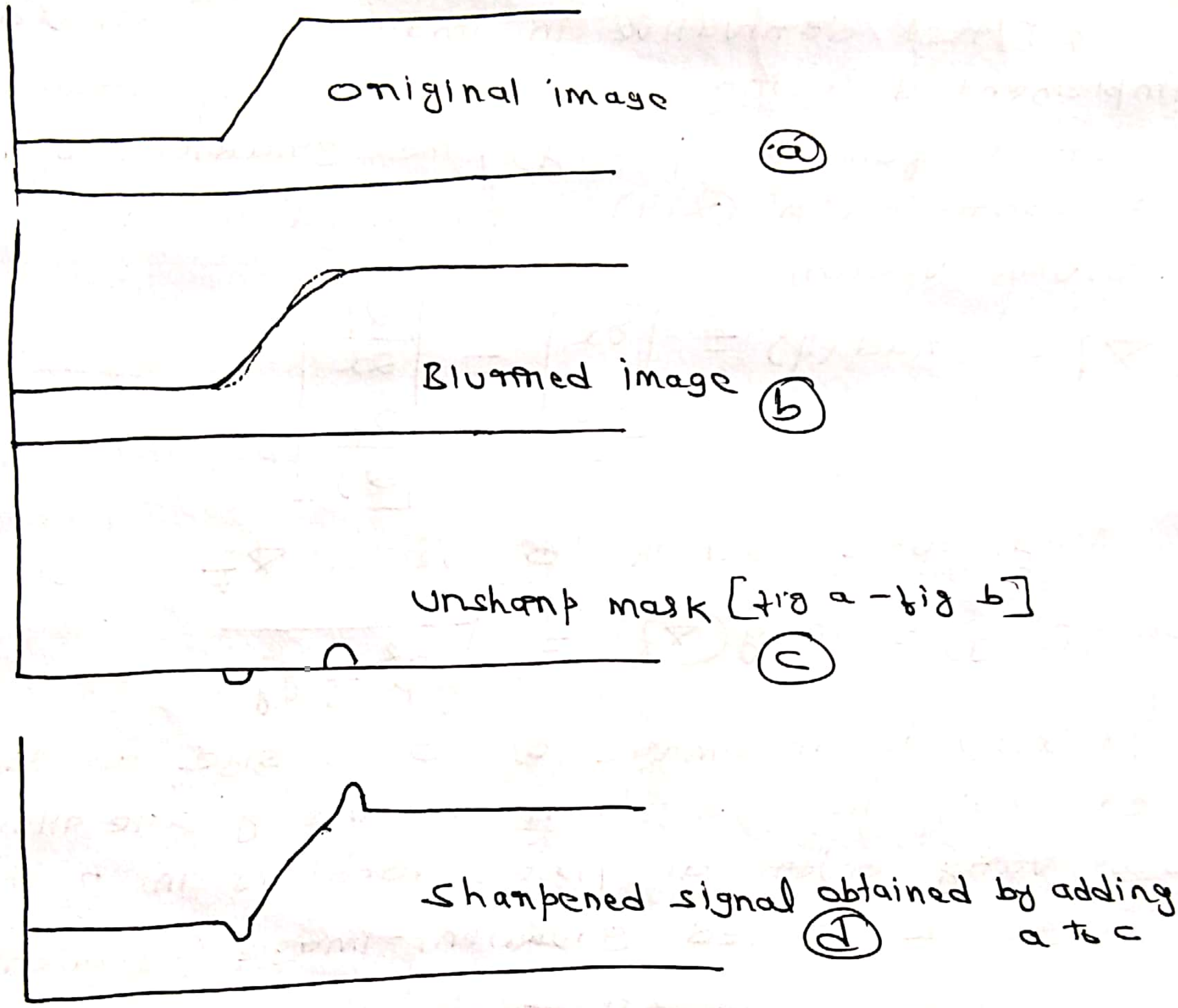



fig:- Illustration of Mechanism of unsharp masking

# First order derivative for image sharpening (Gradient)



First derivatives are directional operators and are defined as

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{vmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{vmatrix}$$

First derivative are implemented using the magnitude of the gradient

$$\nabla f = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$$\simeq |G_x| + |G_y| = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right| \quad -(1)$$

Note :- 1) Gradient with absolute values are simpler to implement

2) Gradient of an image measure the change in image function in $F(x,y)$ in $x$ (column) and $y$ (rows) direction

$\rightarrow x$, $\downarrow y$

| | | |
|---|---|---|
| $z_1$ | $z_2$ | $z_3$ |
| $z_4$ | $z_5$ | $z_6$ |
| $z_7$ | $z_8$ | $z_9$ |

fig: a 3x3 region of an image

In the figure $z_5$ denotes $f(x,y)$ at arbitrary location $(x,y)$, $z_1$ denotes $f(x-1,y-1)$ & so on.

Ex:- $G_x = z_8 - z_5$

$G_y = z_6 - z_5$

From equation (1)

$$\nabla f \simeq |z_8 - z_5| + |z_6 - z_5|$$

(i) Robert operator

Roberts cross gradient operator can be written as

$G_x = (z_9 - z_5)$ , $G_y = (z_8 - z_6)$

From eqn ①

$\nabla f = |z_9 - z_5| + |z_8 - z_6|$

| | |
|---|---|
| -1 | 0 |
| 0 | 1 |

| | |
|---|---|
| 0 | -1 |
| 1 | 0 |

fig: Roberts cross gradient operators

(ii) Sobel operator

sobel operator, 3×3 can be written as

$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$

$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$

$\nabla f = |G_x| + |G_y|$

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

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

fig: sobel operator

Note: 1) All the mask co-efficients in Robert and sobel operator sum to zero, as expected for derivative operator

2) As the center value of sobel operator is zero, it does not include original value of image but it calculates the difference of right and left pixel values (or Top & bottom)

## Basic steps of filtering in Frequency domain

Filtering in frequency domain is multiplication of a suitable filter to input image in fourier domain.

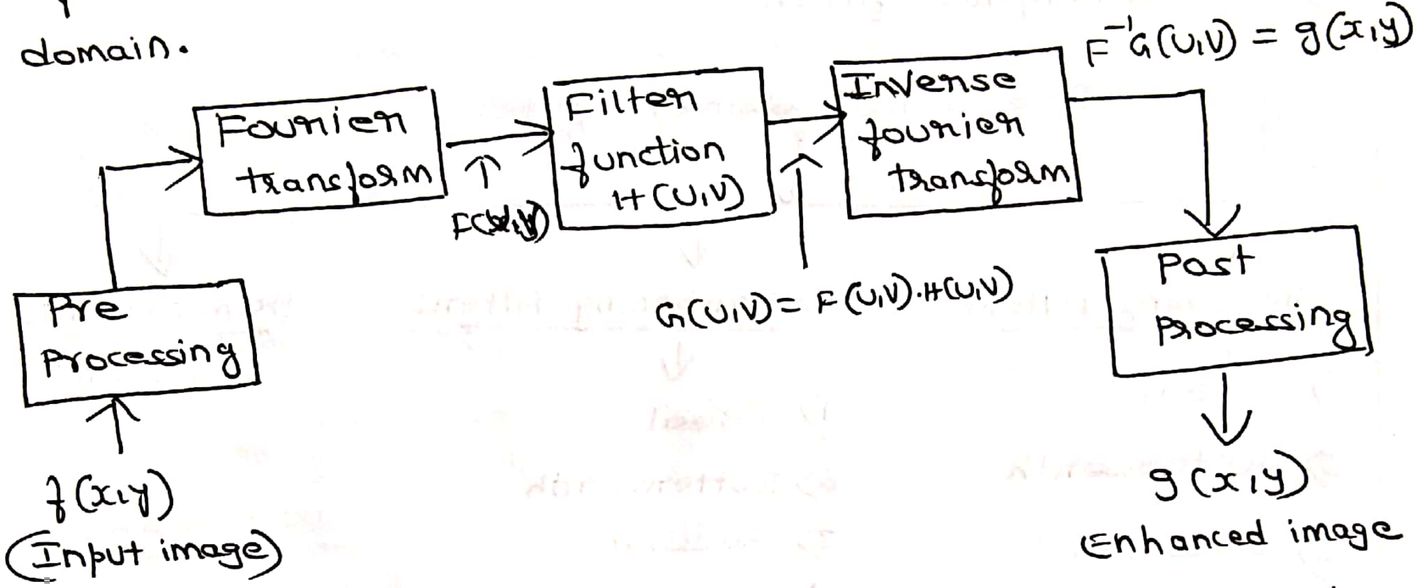



Fig:- Block diagram of filtering in frequency domain

Following steps are followed to do filtering in frequency domain.

1) Multiply the input image $f(x,y)$ by $(-1)^{x+y}$ to center the transform (to take the image at its original place after transformation). It is called pre-processing an image.

2) Now compute $F(u,v)$ i.e Fourier transform of output of step1.

3) Now for enhancement of image, we multiply any filter [According to the requirement] $H(u,v)$ to the image so $G(u,v) = H(u,v) \cdot F(u \cdot v)$

4) Now calculate IDFT of image of step 3 so filtered image $F^{-1}[G(u,v)] = g(x,y)$

5) Now to compensate step1 again multiply the image by $(-1)^{x+y}$ [shift the center back to origin]. This process is called post processing.

Finally the enhanced image is generated.

**Types of frequency domain filters**

1) Smoothing filters
2) Sharpening filters
3) Homomorphic filter

Frequency domain filter

- **Smoothing filters**
  1) Ideal
  2) Butterworth
  3) Gaussian
- **Sharpening filters**
  1) Ideal
  2) Butterworth
  3) Gaussian
  4) Laplacian
  5) High boost
- **Homomorphic filter**

→ Smoothing filters are low pass filters and are used for noise reduction. It blurs object

→ Sharpening filters are high pass filters and produces sharp images i.e enhancing blurred information of image.

→ Homomorphic filters are based on illumination and reflection model.

**Smoothing filters**

* Smoothing filters are LPF
* A LPF only passes low frequency and supress high frequency component from image.

Edges, transitions and noise in grey levels contribute to high frequency contents in an image.

* smoothing filters removes noise and introduce blurring as a side effect in image.

The basic model of filtering is

$G(U,V) = H(U,V) \cdot F(U,V)$

$G(U,V)$ = Enhanced image

$H(U,V)$ = Transfer function of the filter.

$F(U,V)$ = Fourier transform of the image to be filtered.

* Transfer function $H(U,V)$ is of 3 types

1) Ideal LPF
2) Butterworth LPF
3) Gaussian LPF

① Ideal LPF

→ Ideal LPF is defined by transfer function

$$H(U,V) = \begin{cases} 1 & D(U,V) \le D_0 \\ 0 & D(U,V) > D_0 \end{cases}$$

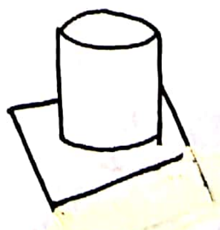
where $D(U,V) = \left[\left(U - \frac{M}{2}\right)^2 + \left(V - \frac{N}{2}\right)^2\right]^{1/2}$

$D(U,V)$ is the distance from point $(U,V)$ to the center $\left(\frac{M}{2}, \frac{N}{2}\right)$

→ ~~[illegible]~~ pass filter pass all frequency components inside $D_0$ and all frequency above $D_0$ [or outside the circle] are suppressed.


ⓐ

fig a.b: Low pass filter
plot of ideal



ⓑ

fig: Filter displayed as an image

→ $D_0$ is the cut off frequency of filter.

Due to sharp cut off frequency. Ideal Low pass filter has ringing problem.

→ cutting the high frequency component results in blurring of image. so that effect on noise in the image will be reduced.

→ $D_0$ is chosen such that most of the frequency component of interest are passed, while unnecessary components are elliminated.

→ Smaller the value of $D_0$, more number of frequency components are elliminated by the filter.

→ As the filter radius increases, less power and information is removed, which results in less blurring.

→ As the cut of frequency reduces, a ringing effect can be seen in image. Ringing is a undesirable and unpleasent lines around the objects present in the image.

→ Total power is given by $P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u,v)$

where $u = 0 --- M-1$

$v = 0 --- N-1$

## Butterworth LPF (BLPF)

→ BLPF does not have a sharp transition between pass band and stop band.

→ It is more appropriate for image smoothing and does not introduce ringing for low order filters

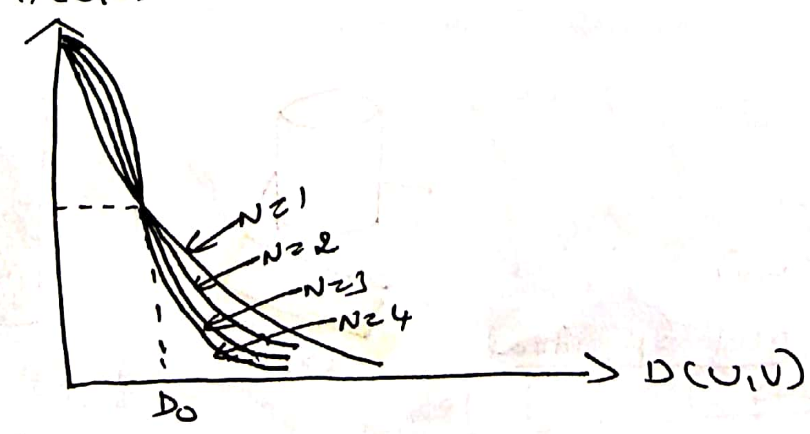



fig: Butterworth LPF

→ The transfer function of BLPF of order n and with cut off frequency at a distance do from the origin is defined by the relation.

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

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

→ For n = 1, the transition is very smooth and has no ringing problem, but as the filter order increases the transfer function approaches towards ideal LPF.

→ For n = 2, Ringing is not seen properly, but as n increases, it becomes more significant.

## Gaussian Low pass filter (GLPF)

→ Gaussian LPF have smooth transition between Pass band and stop band.

→ It does not introduce any ringing in the output image, but only blurring is visible. As the cut-off frequency increases blurring reduces.

→ The Transfer function of GLPF is given by

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

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

& $\sigma$ = spread/Dispersion of Gaussian curve

→ Larger the value of $\sigma$, Larger is the cut off frequency

→ If $\sigma = D_0$ then

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

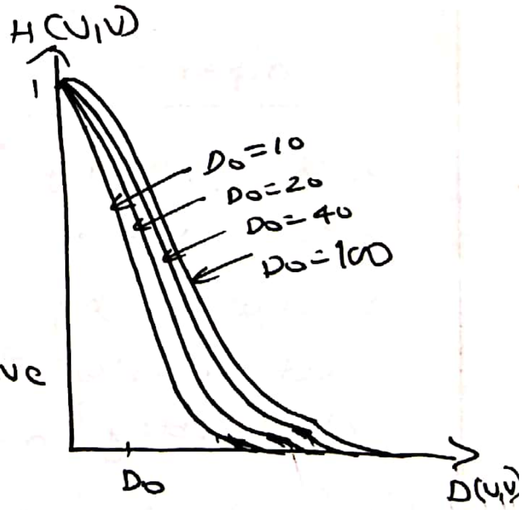



fig: Gaussian LPF

## Applications of smoothing filters (LPF)

1) **Character Recognition** :- Input to an automatic character recognition system is generally poor quality. Input may contain noise due to improper acquisition system. therefore character recognition system fails to give expected result consistently. Hence LPF is used as preprocessing step to blur the image. (Blurring is used to bridge small gaps in the alphabets)

2) **Object counting** : To count number of objects in an image. The o/p of an object counting algorithm may give wrong o/p because of poor quality of I/p image. If there is a small gap in the boundary of objects blurring (LPF) is used, to fill small gaps.

3) **Printing and publishing industry** : Blurred version of image is substracted from the image itself to get better output.

4) **Cosmetic processing** (CP): cosmetic processing is done prior to printing. In CP, blurring is used to reduce the sharpness of fine skin lines and small blemishes on human face.

## Sharpening Filters

→ sharpening ~~filter~~ of image means enhancing blurring information of image.

→ sharpening filters are High pass filters, which attenuates low frequency components without disturbing high frequency component.

→ Transfer function of HPF is given by

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

where $H_{lp}$ is Transfer function of LPF

i.e HPF is reverse operation of LPF

## Ideal High pass filter (IHPF) (30)

Transfer function of 2D IHPF is given by

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

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

$D_0$ is the cut off frequency measured from the origin $(\frac{M}{2}, \frac{N}{2})$.

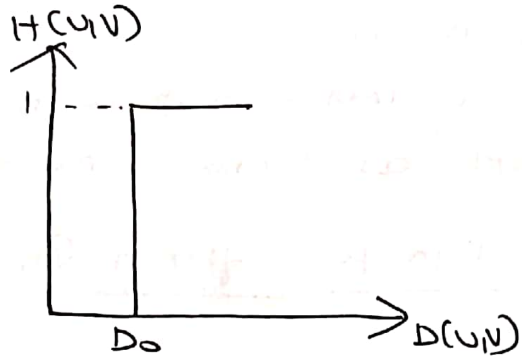



fig: Transfer function of IHPF

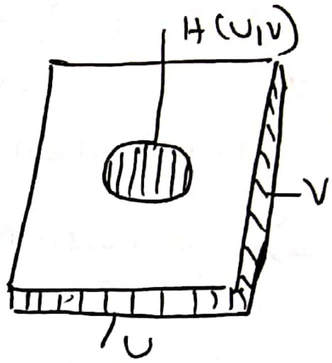



fig: plot of ideal HPF

-> Ideal HPF attenuates all the frequency components inside Do and allow all the frequency components to be passed outside Do. Ringing is clearly visible in the output other than sharp edges and boundaries

## Butterworth High pass filter (BHPF)

* Butterworth HPF does not have sharp transition between pass band and stop band. The slope depends on order of the filter.

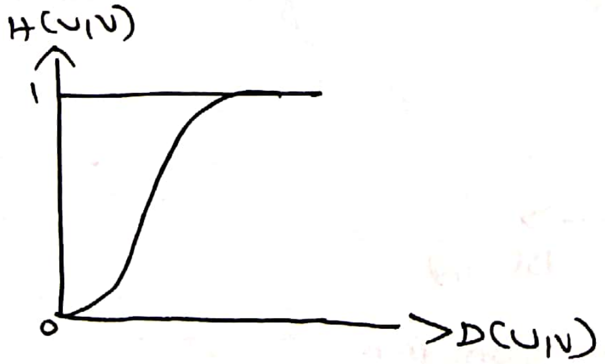



fig: Transfer function of BHPF

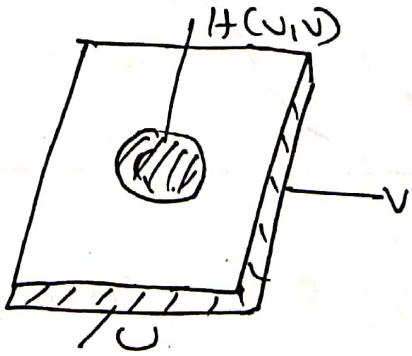



fig: plot of BHPF

* The transfer function of BHPF is given by

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

n = order of filter

Do = cut off frequency

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

* Less distortion is seen in the output with no ringing effect, even ~~after~~ for smaller values of cut off frequencies.
* This filter is more appropriate for image sharpening than ideal LPF as there is no ringing in output

3) Gaussian High pass filter (GHPF)

* Gaussian high pass filter have smooth transition between passband and stop band near cut-off frequency.
* The transfer function of GHPF with cut-off frequency Do from origin is defined by

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

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

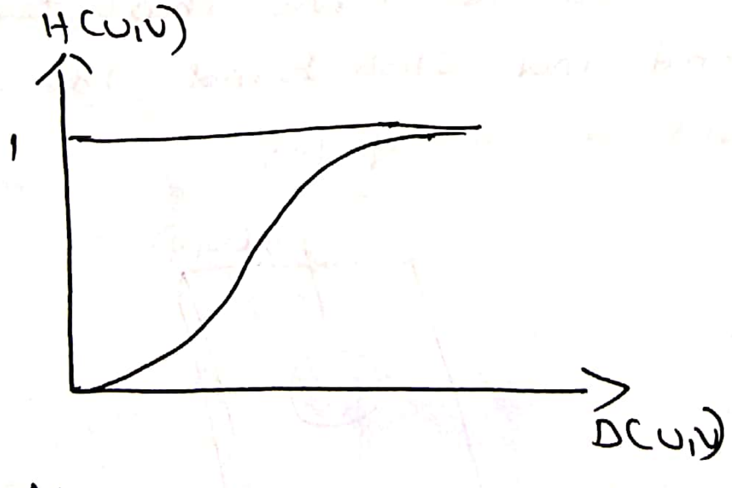



fig: GHPF Transfer function

* Larger the value of Do, larger is cut off frequency.

Scanned with CamScanner

## Laplacian in Frequency domain



Laplacian can be implemented in frequency domain using filter

$$H(u,v) = -4\pi^2 (u^2 + v^2) \quad \text{—— (1)}$$

As the origin of input image $f(x,y)$ is shifted center by multiplying $(-1)^{x+y}$ before taking transform. thus origin of $H(u,v)$ also has to shift from $(0,0)$ to $\left(\frac{M}{2}, \frac{N}{2}\right)$ ∴ Transfer function of Laplacian becomes

$$H(u,v) = -4\pi^2 \left( (u - M/2)^2 + (v - N/2)^2 \right)$$

$$= -4\pi^2 D^2 (u,v) \quad \text{—— (2)}$$

The Laplacian image is obtained by

$$\nabla^2 f(x,y) = F^{-1} [H(u,v) \cdot F(u,v)] \quad \text{—— (3)}$$

where $F(u,v)$ is the DFT of $f(x,y)$

→ Enhancement is done using the equation

$$g(x,y) = f(x,y) + c \nabla^2 f(x,y) \quad \text{—— (4)}$$

Here $c = -1$ because $H(u,v)$ is negative (eq (1))

→ In frequency domain equation 4 is written as

$$g(x,y) = F^{-1} [F(u,v) - H(u,v) \cdot F(u,v)]$$

$$= F^{-1} [[1 - H(u,v)] F(u,v)] \quad \text{—— (5)}$$

substituting equation (2) in equation 5

$$g(x,y) = F^{-1} \{ [1 - (-4\pi^2 D^2(u,v))] F(u,v) \}$$

$$\boxed{g(x,y) = F^{-1} \{ [1 + 4\pi^2 D^2 (u,v)] F(u,v) \}}$$

## Unsharp masking and High boost filtering

Unsharp mask is obtained by subtracting blurred image (low pass filtered image) from the original image.

$$g_{mask}(x,y) = f(x,y) - f_{lp}(x,y)$$

where

$$f_{lp}(x,y) = F^{-1}[H_{LP}(u,v)\, F(u,v)]$$

i.e $H_{LP}(u,v)$ is a Low pass filter &

$F(u,v)$ is the fourier transform of $f(x,y)$

$f_{lp}(x,y)$ is smoothed image ($\bar{f}(x,y)$)

→ Now add the mask back to original image

$$g(x,y) = f(x,y) + K * g_{mask}(x,y) \quad \text{—(1)}$$

when $K=1$, then it is unsharp masking

when $K>1$, then it is high boost filtering

we can express equation (1) in terms of Low pass filter as

$$g(x,y) = F^{-1}\left[\underline{1 + K * [1 - H_{LP}(u,v)]}\, F(u,v)\right]$$

eqn (1) in terms of High pass filter is written as

$$g(x,y) = F^{-1}\left\{\underline{[1 + K * H_{hp}(u,v)]}\, F(u,v)\right\}$$

The expression within square brackets is called high frequency emphasis filter. High pass filter set the DC term to '0', thus reducing the average intensity in the filtered image to 0. The high frequency emphasis filter does not have this problem because of '1' that is added to high pass filter.

[K gives the control over the proportion of high frequencies that influence the final result.

## Homomorphic filtering



Homomorphic filtering is frequency domain procedure to improve the appearance of an image.

An image $f(x,y)$ captured by camera is formed by multiplication of illumination and reflectance.

i.e $\quad f(x,y) = i(x,y) \cdot r(x,y)$ ——— (1)

For image enhancement illumination and reflectance have to be treated separately, which is not possible in frequency domain as

$$F[f(x,y)] \neq F[i(x,y)] \cdot F[r(x,y)] \quad \text{——— (2)}$$

To seperate illumination and reflectance component Homomorphic filters are used.

$$f(x,y) \rightarrow \boxed{\ln} \xrightarrow{z(x,y)} \boxed{DFT} \xrightarrow{Z(U,V)} \boxed{H(U,V)} \xrightarrow{S(U,V)} \boxed{DFT^{-1}} \xrightarrow{s(x,y)} \boxed{\exp} \rightarrow g(x,y)$$

fig: Block diagram of homomorphic filtering

**Step 1 :** Take Natural logarithm of input image

$$z(x,y) = \ln f(x,y)$$
$$= \ln [i(x,y) \cdot r(x,y)]$$
$$\boxed{z(x,y) = \ln i(x,y) + \ln r(x,y)} \quad \text{——— (3)}$$

**Step 2 :** Take Fourier transform on both sides

$$F[z(x,y)] = F[\ln i(x,y) + \ln r(x,y)]$$
$$\boxed{Z(U,V) = F_i(U,V) + F_r(U,V)} \quad \text{——— (4)}$$

Where $F_i(U,V) = F[\ln i(x,y)]$
& $F_r(U,V) = F[\ln r(x,y)]$

**Step 3 :** Multiply with filter $H(U,V)$ with eqn 4

$$S(U,V) = H(U,V)\, Z(U,V)$$
$$= H(U,V)\,[F_i(U,V) + F_r(U,V)]$$
$$\boxed{S(U,V) = H(U,V)\, F_i(U,V) + H(U,V)\, F_r(U,V)} \quad \text{——— (5)}$$

Step 4 :- Take inverse fourier transform on both sides

$$s(x,y) = F^{-1}[S(u,v)]$$

$$= F^{-1}[H(u,v)\cdot F_i(u,v) + H(u,v)\cdot F_r(u,v)]$$

$$= F^{-1}[H(u,v)\cdot F_i(u,v)] + F^{-1}[H(u,v)\cdot F_r(u,v)]$$

$$\boxed{s(x,y) = i'(x,y) + r'(x,y)} \quad \text{—— (6)}$$

Where $i'(x,y) = F^{-1}[H(u,v)\cdot F_i(u,v)]$

& $r'(x,y) = F^{-1}[H(u,v)\cdot F_r(u,v)]$

Step 5 : Take inverse log transformation

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

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

$$\boxed{g(x,y) = i_0(x,y)\cdot r_0(x,y)} \quad \text{—— (7)}$$

Where $i_0(x,y) = e^{i'(x,y)}$

$r_0(x,y) = e^{r'(x,y)}$

$g(x,y)$ = enhanced image

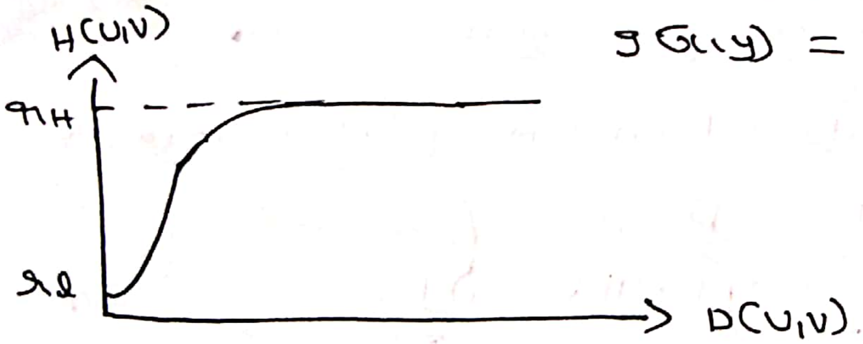



fig: Homomorphic filter transfer function

→ Goal of Homomorphic filtering is to supress low frequencies associated with input images. To acheive this filter is designed in such a way that illumination component is suppressed and reflectance is enhanced as shown in fig.

→ Low frequencies are associated with illumination and high frequencies are associated with reflectance.

$r_L < 1 \Rightarrow$ To Decrease contribution made by low frequency (illumination)

$r_H > 1 \Rightarrow$ To increase contribution made by high frequency (Reflactance)

Radhakrishna M
GAT, B'lore

Scanned with CamScanner

# Aliasing and Moire Pattern

The sine and cosine components with highest frequency determine highest frequency component of the function. suppose this highest frequency is finite and function is of unlimited duration, than according to shannon's sampling theorem, if a function is sampled at a rate equal to or greater than twice its highest frequency, then it is possible to recover completly the original functions from its samples.

If the function is undersampled then the phenomenon aliasing occurs in an image. The aliasing is a form additional frequency content being introduced in sampled function. These are called aliasing frequency.

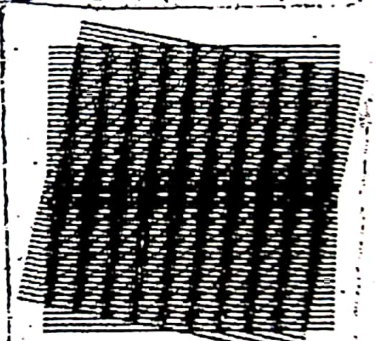


fig: Aliasing & Moire pattern

consider the sample data which are finite in duration. we can model the process of converting a function of unlimited duration into a function of finite duration, simply by multiplying the unlimited function by gating function [1 or 0]

The principle approach for aliasing effect on a image is to reduce its highest frequency component by blurring the image prior to sampling, however aliasing is always present in sampled image. The effect of aliased frequency can be seen in the form of moire pattern.

## Preliminary concepts



### ① Complex numbers

A complex number c is defined as

$$\boxed{c = R + jI}$$

Where R and I are real numbers

and $j = \sqrt{-1}$

The conjugate of a complex number is defined as

$$\boxed{c^* = R - jI}$$

complex number in polar coordinates is represented as

$$\boxed{c = |c| (\cos\theta + j\sin\theta)}$$

Where $|c| = \sqrt{R^2 + I^2}$

$\theta$ = Angle between the vector and real axis.

$\tan\theta = \frac{I}{R} \Rightarrow \theta = \tan^{-1}(I/R)$

using Euler's formula

$e^{j\theta} = \cos\theta + j\sin\theta$ where $e = 2.71828$

so complex numbers in polar coordinates can be represented as $c = |c| e^{j\theta}$

### ② Fourier series

A function $f(t)$ of a continuous variable t that is periodic with period T, can be expressed as the sum of sines and cosines multiplied by appropriate coefficients. This sum is known as Fourier series

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{T}t}$$

Where $c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\, e^{-j\frac{2\pi n}{T}t}\, dt$ for $n = 0, \pm1, \pm2 \dots$

## ③ Impulses and their shifting property

A unit impulse of a continuous variable $t$ located at $t=0$ is defined as

$$\delta(t) = \begin{cases} \infty & \text{if } t=0 \\ 0 & \text{if } t\neq 0 \end{cases} \quad \text{and} \int_{-\infty}^{\infty} \delta(t)\,dt = 1$$

An impulse has shifting property w.r.t integration i.e

$$\int_{-\infty}^{\infty} f(t)\delta(t)\,dt = f(0)$$

Provided that $f(t)$ is continuous at $t=0$

shifting property involves an impulse located at an arbitary point $t_0$ denoted by $\delta(t-t_0)$. In this case shifting property becomes

$$\int_{-\infty}^{\infty} f(t)\,\delta(t-t_0)\,dt = f(t_0)$$

Let $x$ represent a discrete variable. The unit discrete impulse $\delta(x)$ is defined as

$$\delta(x) = \begin{cases} 1 & x=0 \\ 0 & x\neq 0 \end{cases} \quad \text{and} \sum_{x=-\infty}^{\infty} \delta(x) = 1$$

The shifting property for discrete variable has form

$$\sum_{x=-\infty}^{\infty} f(x)\,\delta(x) = f(0)$$

For $x = x_0$

$$\sum_{x=-\infty}^{\infty} f(x)\,\delta(x-x_0) = f(x_0)$$

Scanned with CamScanner

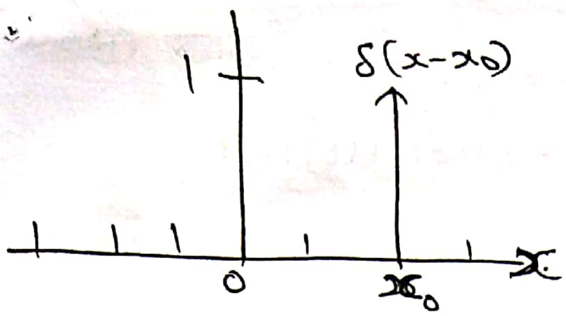



fig:- A discrete impulse located at $x = x_0$. variable $x$ is discrete and $\delta$ is 0 everywhere except at $x = x_0$

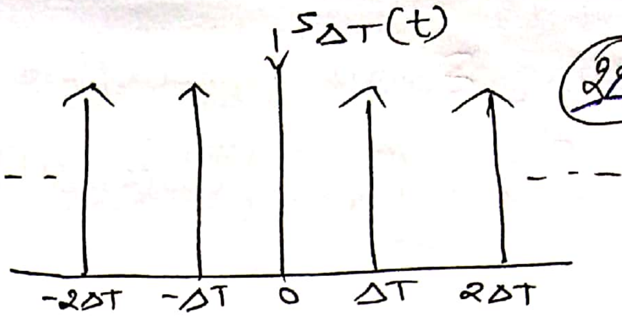



fig:- Impulse Train

## (4) The Fourier transform of function of one continuous variable

The fourier transform of a continuous function $f(t)$ of a continuous variable $t$; is defined by

$$F\{f(t)\} = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi \mu t}\, dt$$

where $\mu$ is a continuous variable and $F\{f(t)\}$ is a function only of $\mu$

$$\therefore\ F(\mu) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi \mu t}\, dt \quad — (1)$$

If $F(\mu)$ is given, then we can obtain $f(t)$ back by using Inverse fourier transform

$$f(t) = F^{-1}\{F(\mu)\}$$

$$= \int_{-\infty}^{\infty} F(\mu)\, e^{j2\pi\mu t}\, d\mu \quad — (2)$$

Equation 1 & 2 are called fourier transform pair

Using Eulers formula, we can express equation ① as

$$F(\mu) = \int_{-\infty}^{\infty} f(t)\left[\cos(2\pi\mu t) - j\sin(2\pi\mu t)\right] dt$$

⑤ **Convolution**

The convolution of 2 continuous functions $f(t)$ and $h(t)$ of one continuous variable $t$ can be denoted by operator $*$ and is defined as

$$f(t) * h(t) = \int_{-\infty}^{\infty} f(\tau)\, h(t-\tau)\, d\tau \quad — ①$$

Taking the F·T of equation ①

$$F\{f(t) * h(t)\} = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(\tau)\, h(t-\tau)\, d\tau\right] e^{-j2\pi\mu t} dt$$

$$= \int_{-\infty}^{\infty} f(\tau)\left[\int_{-\infty}^{\infty} h(t-\tau)\, e^{-j2\pi\mu t} dt\right] d\tau \quad — ②$$

Again $F[h(t-\tau)] = H(\mu)\, e^{-j2\pi\mu\tau}$

where $H(\mu) = F\cdot T\,[h(t)]$

So putting this value in equation ②

$$F\{f(t) * h(t)\} = \int_{-\infty}^{\infty} f(\tau)\left[H(\mu)\, e^{-j2\pi\mu\tau}\right] d\tau$$

$$= H(\mu)\int_{-\infty}^{\infty} f(\tau)\, e^{-j2\pi\mu\tau}\, d\tau$$

$$= H(\mu)\, F(\mu) \quad — ③$$

→ So $f(t) * h(t)$ and $H(\mu)\, F(\mu)$ are F.T pairs

## The 2-D impulse and its shifting property



The impulse $\delta(t,z)$, of 2 Continuous Variables $t$ and $z$ is defined as

$$\delta(t,z) = \begin{cases} \infty & \text{if } t=z=0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(t,z)\, dt\, dz = 1$$

The 2D impulse exibits the shifting property under integration

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\,\delta(t,z)\, dt\, dz = f(0,0)$$

i.e for an impulse located at coordinates $(t_0, z_0)$,

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\,\delta(t-t_0, z-z_0)\, dt\, dz = f(t_0, z_0)$$

For discrete variables $x$ and $y$, the 2D impulse is defined as

$$\delta(x,y) = \begin{cases} 1 & \text{if } x=y=0 \\ 0 & \text{otherwise} \end{cases}$$

and its shifting property is

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\,\delta(x,y) = f(0,0)$$

For impulse located at coordinates $(x_0, y_0)$ the shifting property is

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\,\delta(x-x_0, y-y_0) = f(x_0, y_0)$$

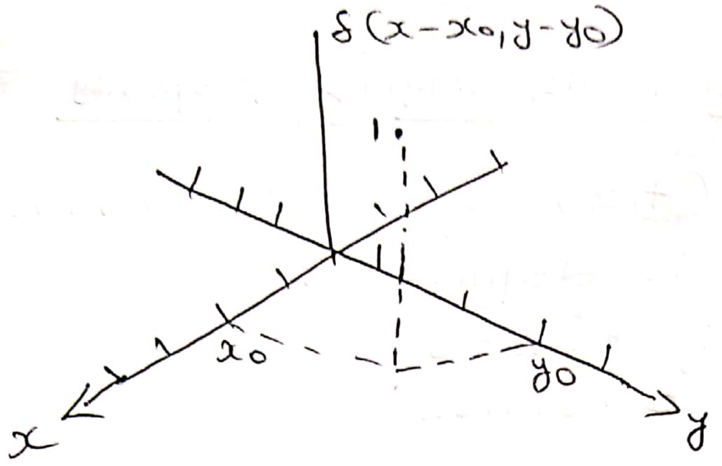



Fig: 2D unit discrete impulse. Variables $x$ and $y$ are discrete and $\delta$ is 0 everywhere except at coordinates $(x_0, y_0)$.

## The 2D continuous Fourier transform pair

Let $f(t,z)$ be a continuous function of 2 continuous variables $t$ and $z$. The 2D, continuous Fourier Transform pair is given by

$$F(\mu,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\, e^{-j2\pi(\mu t + vz)}\, dt\, dz$$

and

$$f(t,z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu,v)\, e^{j2\pi(\mu t + vz)}\, d\mu\, dv$$

where $\mu$ and $v$ are frequency variables

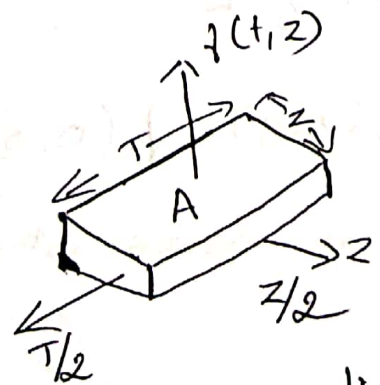



fig: 2D function

## Example: obtaining a 2D fourier transform of a simple function

(26)

$$F(\mu,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\, e^{-j2\pi(\mu t + vz)}\, dt\, dz$$

$$= \int_{-T/2}^{T/2}\int_{-Z/2}^{Z/2} A\, e^{-j2\pi(\mu t + vz)}\, dt\, dz$$

$$= ATZ \left[\frac{\sin(\pi\mu T)}{(\pi\mu T)}\right]\left[\frac{\sin \pi v Z}{(\pi v Z)}\right]$$

The magnitude is given by the expression

$$|F(\mu,v)| = ATZ \left|\frac{\sin \pi\mu T}{\pi\mu T}\right| \left|\frac{\sin \pi v Z}{\pi v Z}\right|$$

## Two dimensional sampling and 2D sampling theorem

Sampling in 2D can be modeled using the sampling function (2D impulse train):

$$S_{\Delta T \Delta Z}(t,z) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \delta(t - m\Delta T, z - n\Delta Z) \quad -(1)$$

where $\Delta T$ and $\Delta Z$ are the seperations between samples along t and z axis of a continuous function $f(t,z)$

Equation ① describes a set of periodic impulses extending infinitely along the two axes as shown in below figure

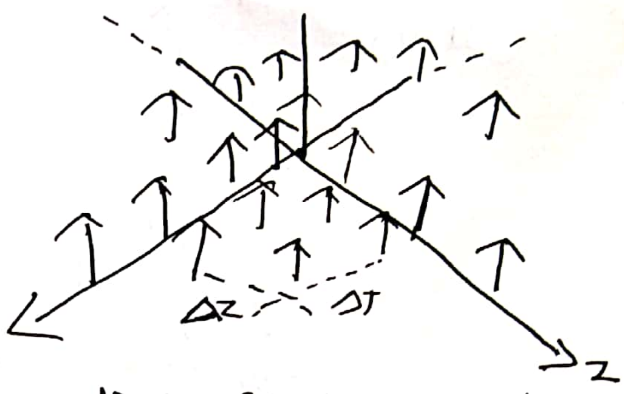



fig: 2D impulse train

Function $f(t,z)$ is said to be band limited if its fourier transform is 0 outside a rectangle established by the intervals $[-\mu_{max}, \mu_{max}]$ and $[-v_{max}, v_{max}]$ i.e

$$F(\mu,v)=0 \text{ for } |\mu| \geq \mu_{max} \text{ and } |v| \geq v_{max}$$

The 2D sampling theorem states that a continuous, band limited function $f(t,z)$ can be recovered with no error from a set of its samples if the sampling intervals are

$$\Delta T < \frac{1}{2\mu_{max}}$$

and $$\Delta Z < \frac{1}{2v_{max}}$$

or in terms of sampling rate

$$\frac{1}{\Delta T} > 2\mu_{max}$$

and $$\frac{1}{\Delta Z} > 2v_{max}$$

i.e No information is lost, if a 2D Band limited continuous function is represented by samples acquired at rates greater than twice the highest frequency content of the function in both $\mu$ and $v$ direction

# The 2D Discrete Fourier transform and its inverse

The 2D DFT is defined as

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

The 2D IDFT is given by

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{\frac{j2\pi ux}{M}} \cdot e^{\frac{j2\pi vy}{N}} \quad -(2)$$

## Properties of 2D DFT

### (1) Relationship between Spatial and Frequency Intervals

Suppose a continuous function $f(t,z)$ is sampled to form a digital image $f(x,y)$, consisting of $M \times N$ samples taken in $t$ and $z$ directions respectively.

Let $\Delta T$ and $\Delta Z$ denote the seperations between the samples. Then the corresponding discrete, frequency domain variables are given by

$$\Delta u = \frac{1}{M\Delta T} \quad \text{and} \quad \Delta v = \frac{1}{N\Delta Z} \quad \text{respectively}$$

Note:- Separations between samples in frequency domain are inversely proportional to the spacing between spatial samples and the number of samples

### (2) Translation and Rotation

Fourier Transform pair satisfies the following translation property as

$$f(x,y)\, e^{j2\pi(u_0 x/M + v_0 y/N)} \Longleftrightarrow F(u-u_0, v-v_0) \quad -(A)$$

[From eq^n 1]

and

$$f(x-x_0, y-y_0) \Leftrightarrow F(u,v) e^{-j2\pi (x_0 u/M + y_0 v/N)} \quad \text{--- (B)}$$

(From eqⁿ (2))

Translation has no effect on magnitude spectrum of $F(u,v)$

i.e multiplying $f(x,y)$ by the exponential, will shift the origin of DFT to $(u_0, v_0)$ (eqⁿ A)

→ multiplying $F(u,v)$ by the -ve of that exponential, shifts the origin of $f(x,y)$ to $x_0, y_0$

## Rotation property

Rotation property states that, if a function is rotated by the angle, its Fourier Transform also rotates by an equal amount.

Using the polar coordinates

$x = r\cos\theta$, $y = r\sin\theta$

$u = w\cos\phi$, $v = w\sin\phi$

we can write as

$$f(r, \theta + \theta_0) \Leftrightarrow F(w, \phi + \theta_0)$$

## (3) Periodicity

2D Fourier transform and its inverse are infinitely periodic in u and v direction

$$F(u,v) = F(u+k_1 M, v) = F(u, v+k_2 N) =$$
$$F(u+k_1 M, v+k_2 N) \text{ and}$$

$$f(x,y) = f(x+k_1 M, y) = f(x, y+k_2 N) =$$
$$f(x+k_1 M, y+k_2 N)$$

Where $k_1$ and $k_2$ are integers

# (4) Symmetry property

Any Real or complex function $w(x,y)$ can be expressed as sum of an even and odd part (can be real or complex)

$$w(x,y) = w_e(x,y) + w_o(x,y) \quad \text{—} (1)$$

where the even and odd parts are

$$w_e(x,y) \triangleq \frac{w(x,y) + w(-x,-y)}{2} \quad \text{—} (2)$$

and $w_o(x,y) \triangleq \dfrac{w(x,y) - w(-x,-y)}{2}$ — (3)

Putting equation (2) and (3) in eqn (1) we get

$$w(x,y) \equiv w(x,y)$$

i.e $w_e(x,y) = w_e(-x,-y)$

$w_o(x,y) = -w_o(-x,-y)$

→ Even functions are symmetric and odd functions are anti-symmetric

→ we can write

$$w_e(x,y) = w_e(M-x, N-y) \quad \text{—} (4)$$

$$w_o(x,y) = -w_o(M-x, N-y) \quad \text{—} (5)$$

where M and N are no. of rows and columns of 2D array

i.e $\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} w_e(x,y) \cdot w_o(x,y) = 0$

Scanned with CamScanner

$\therefore$ sum of samples of an discrete function is zero for odd function

⑤ Fourier spectrum and phase angle

As 2D DFT is complex in general, it can be expressed in polar form as

$$F(u,v) = |F(u,v)| e^{-j\phi(u,v)}$$

Where the magnitude $|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$ is called Fourier spectrum (or frequency spectrum) and $\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$ is phase angle

→ The power spectrum is defined as

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

R and I are Imaginary parts of $F(u,v)$ for $u = 0,1,2 \ldots M-1$ and $v = 0,1,2 \ldots N-1$

The spectrum has even symmetry about the origin i.e $|F(u,v)| = |F(-u,-v)|$ and

Phase angle exhibits odd symmetry about the origin

$$\phi(u,v) = -\phi(-u,-v)$$

⑥ 2D convolution theorem

convolution in spatial domain is equal to multiplication in Frequency domain

convolution of two sequence $x(n)$ & $h(n)$ is defined as

$$x(n) * h(n) = \sum_{K=-\infty}^{\infty} x(K)\, h(n-K)$$

Two dimentional convolution of two arrays (or matrices) $f(x,y)$ and $h(x,y)$ is given by

$$f(x,y) * h(x,y) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n)\, h(x-m, y-n)$$

Proof:

DFT of the convolution of two arrays $f(x,y)$ and $h(x,y)$ is given by.

34

$$DFT\left\{f(x,y) * h(x,y)\right\} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\left[\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, h(x-m, y-n)\right] \cdot e^{-j\frac{2\pi xU}{N}} \cdot e^{-j\frac{2\pi yV}{N}} \quad \text{--- (1)}$$

By taking all $\Sigma$ outside, we get

$$DFT\left\{f(x,y) * h(x,y)\right\} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, h(x-m, y-n) \cdot e^{-j\frac{2\pi}{N}(x-m+m)U} \cdot e^{-j\frac{2\pi}{N}(y-n+n)V} \quad \text{---} \bigcirc$$

Splitting the terms $e^{-j\frac{2\pi}{N}(x-m+m)U}$ and $e^{-j\frac{2\pi}{N}(y-n+n)V}$

$$= \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, h(x-m, y-n)\, e^{-j\frac{2\pi}{N}(x-m)U}\, e^{-j\frac{2\pi}{N}mU} \cdot e^{-j\frac{2\pi}{N}(y-n)V} \cdot e^{-j\frac{2\pi}{N}nV}$$

$$= \underbrace{\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\pi}{N}mU} \cdot e^{-j\frac{2\pi}{N}nV}}\ \underbrace{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} h(x-m, y-n)\, e^{-j\frac{2\pi}{N}(x-m)U} \cdot e^{-j\frac{2\pi}{N}(y-n)V}}$$

$$= F(K, l) \cdot h(K, l)$$

i.e. $\boxed{DFT\left\{f(x,y) * h(x,y)\right\} = F(U,V) \cdot h(U,V)}$

# Questions

1) What is Image Enhancement? Explain the basic Gray level transformations
2) explain the different piecewise linear transformations
3) Write a Note on Gamma correction
4) Explain Histogram equalization for image Enhancement
5) Explain Histogram specification for image Enhancement.
6) write a Note on Local Enhancement
7) Explain image enhancement using arithmetic and logical operations.
8) Write a Note on spatial Filtering.
9) What is smoothing filters? Explain different Smoothing linear filters.
10) Explain order statistics filters / non linear Smoothing spatial filters.
11) Explain First and second derivative for image sharpening
12) Explain Laplacian for image sharpening
13) Explain unsharp masking and high boost filtering in spatial domain.
14) Write a note on gradient for image sharpening.
15) Explain the block diagram of filtering in frequency domain.

16) Explain different smoothing frequency domain filters.

17) Explain different sharpening frequency domain filters

18) Explain Laplacian in frequency domain

19) Explain unsharp masking and highboost filtering in frequency domain

20) With block diagram, explain homomorphic filtering

21) Explain the 2D-DFT properties.

22) Write a note on Aliasing & Moire pattern

Problems