|| Shri Hari ||

BCSE 0101: Digital Image Processing

Prof. Charul Bhatnagar
Room no. 325, AB - I
Email: charul@gla.ac.in
WA no.: 7055510537

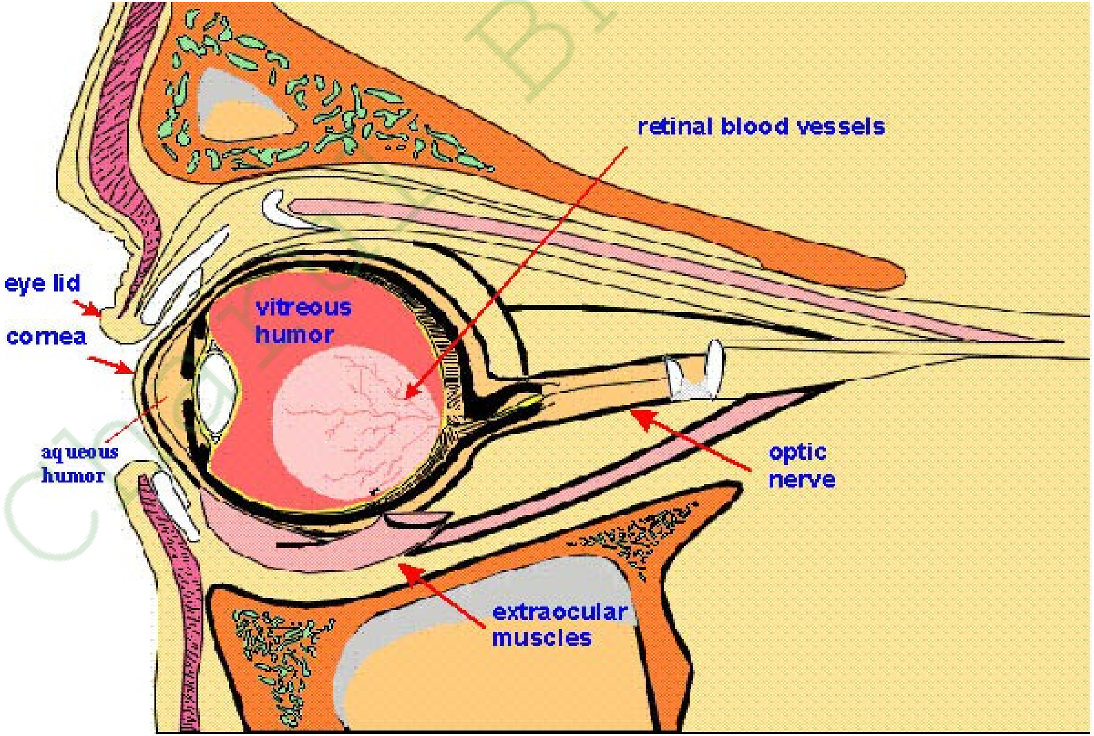
Module I

Introduction and Fundamentals: Motivation and Perspective, Applications, Components of Image Processing System, Element of Visual Perception, A Simple Image Model, Sampling and Quantization, Some Basic Relationships between Pixels.

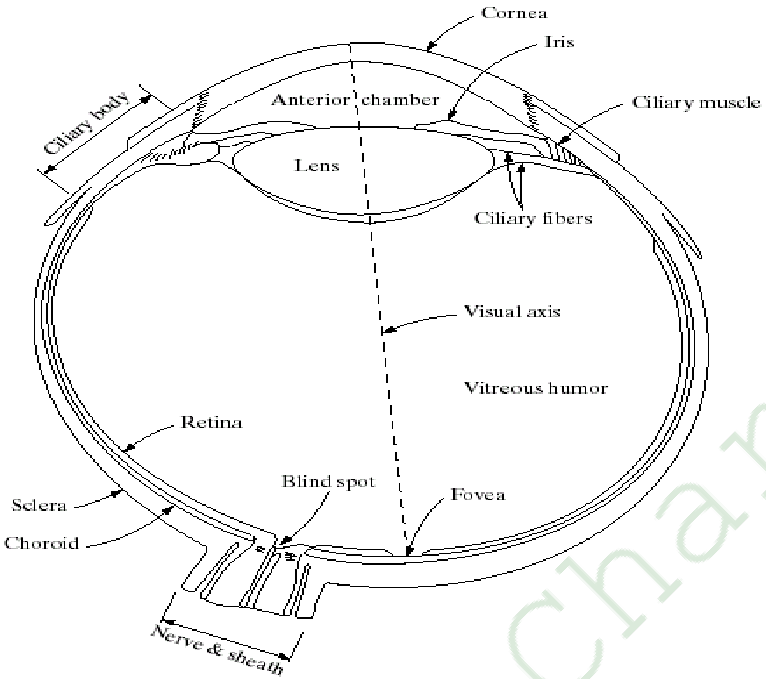
Intensity Transformations and Spatial Filtering: Introduction, Some Basic Intensity Transformation Functions, Histogram Processing, Histogram Equalization, Histogram Specification, Local Enhancement, Enhancement using Arithmetic/Logic Operations – Image Subtraction, Image Averaging, Basics of Spatial Filtering, Smoothing - Mean Filter, Order Statistics Filters, Sharpening – The Laplacian.

Filtering in the Frequency Domain: Fourier Transform and the Frequency Domain, Basis of Filtering in Frequency Domain

Elements of Visual Perception



Human Visual Perception



Cross Section of Human Eye

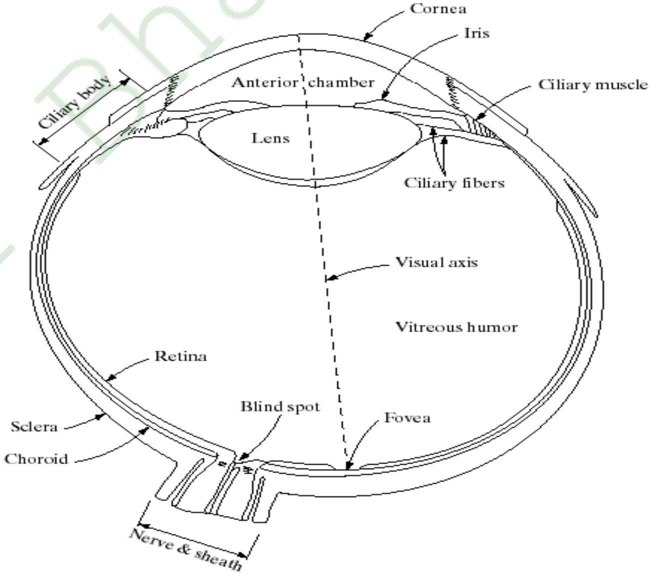
- Avg diameter : 20mm
- 3 membranes enclose the eye

1. Cornea & sclera
2. Choroid
3. Retina

1. Cornea & Sclera

Cornea

- Tough & Transparent tissue
- Covers the anterior surface of the eye

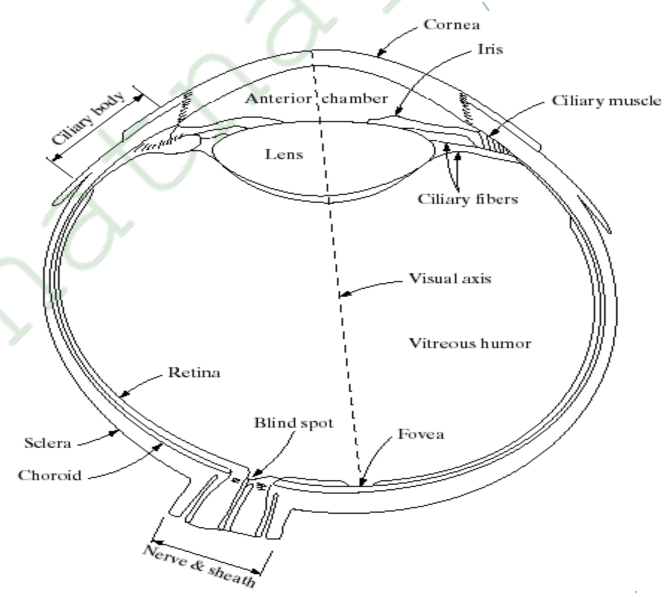


Sclera

- Continuous with cornea
- Opaque membrane
- Encloses the remainder of the optic globe

2. Choroid

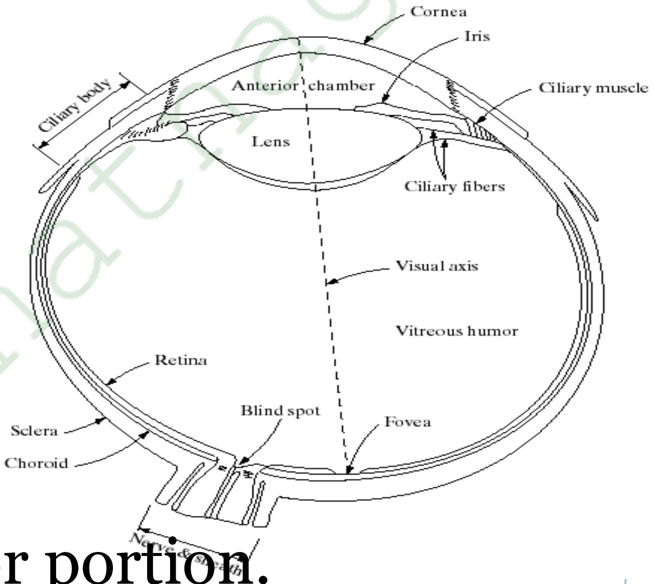
- Choroid contains blood vessels for eye nutrition; it is heavily pigmented to reduce extraneous light entrance and backscatter.
- Divided into the ciliary body & the iris diaphragm, which controls the amount of light that enters the pupil (2 mm ~ 8 mm).



- Lens made up of fibrous cells & is suspended by fibers that attach it to the ciliary body.
- It is slightly yellow and absorbs approx. 8% of the visible light spectrum.

3. Retina

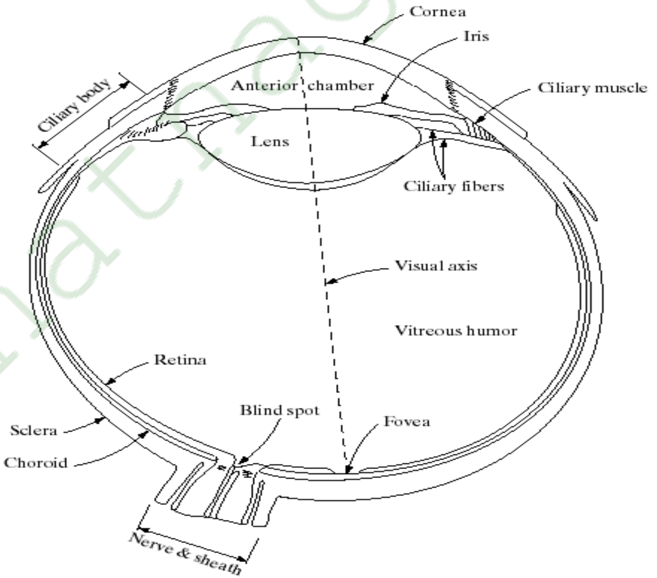
Light from an object is imaged on the retina



- The retina lines the entire posterior portion.
- Discrete light receptors are distributed over the surface of the retina:
  - cones (6-7 million per eye) and
  - rods (75-150 million per eye)

Cones

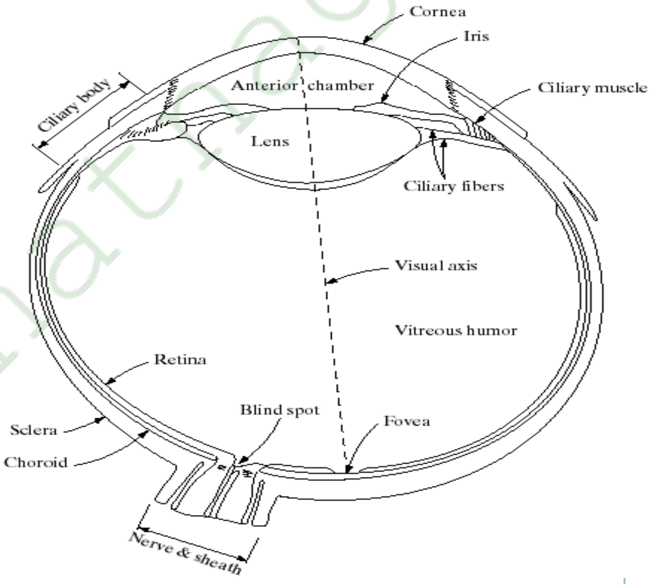
- Cones are located in the fovea and are sensitive to color.
- Each one is connected to its own nerve end.
- Cone vision is called photopic (or bright-light vision).



Muscles controlling the eye rotate the eye ball until the image of an object of interest falls on the fovea.

Rods

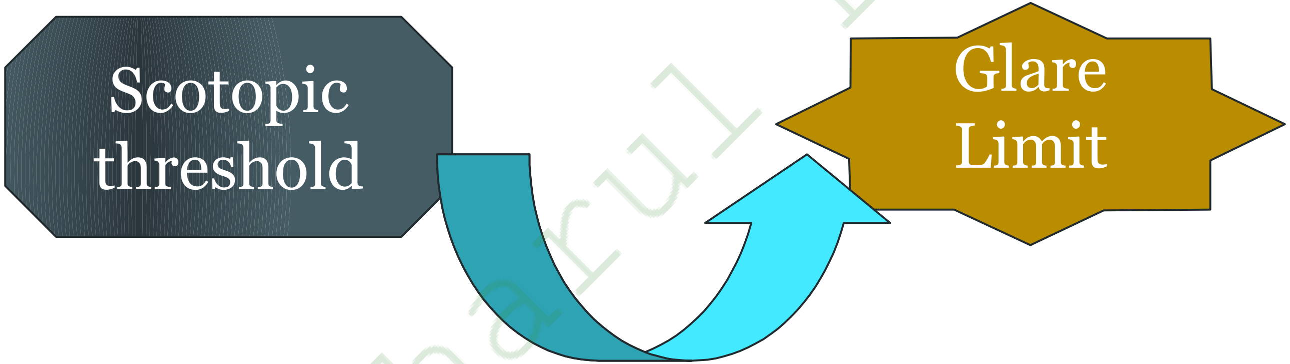
- Rods are distributed over the retinal surface
- Rods give a general, overall picture of the field of view and are not involved in color vision.
- Several rods are connected to a single nerve and are sensitive to low levels of illumination (*scotopic* or dim-light vision).



Objects seen by moon light appear as colourless forms because only rods are stimulated.

Brightness Adaptation & Discrimination

Range of light intensity levels to which human visual system (HVS) can adapt of the order of **1010**

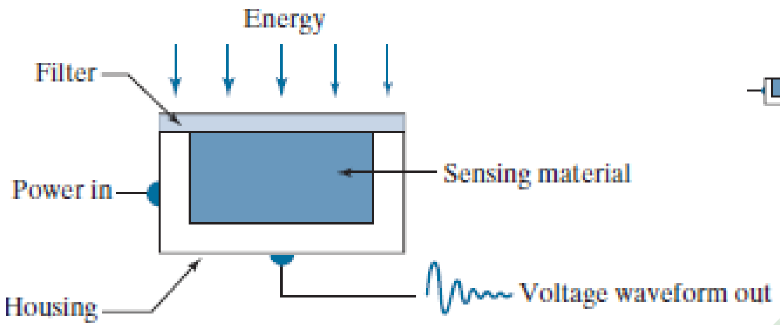


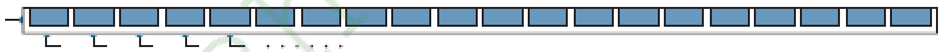
Subjective brightness (i.e. intensity as perceived by the HVS) is a **logarithmic function** of the light intensity incident on the eye.

Range of adaptable light intensity levels - 1010

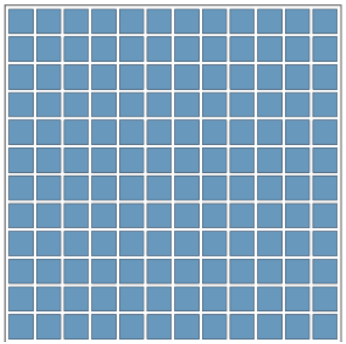
- The HVS cannot operate over such a range simultaneously
- Accomplishes this large variation by changing its overall sensitivity, a phenomenon known as **brightness adaptation.**
- The total range of distinct intensity levels the eye can discriminate simultaneously is rather small

Image Acquisition





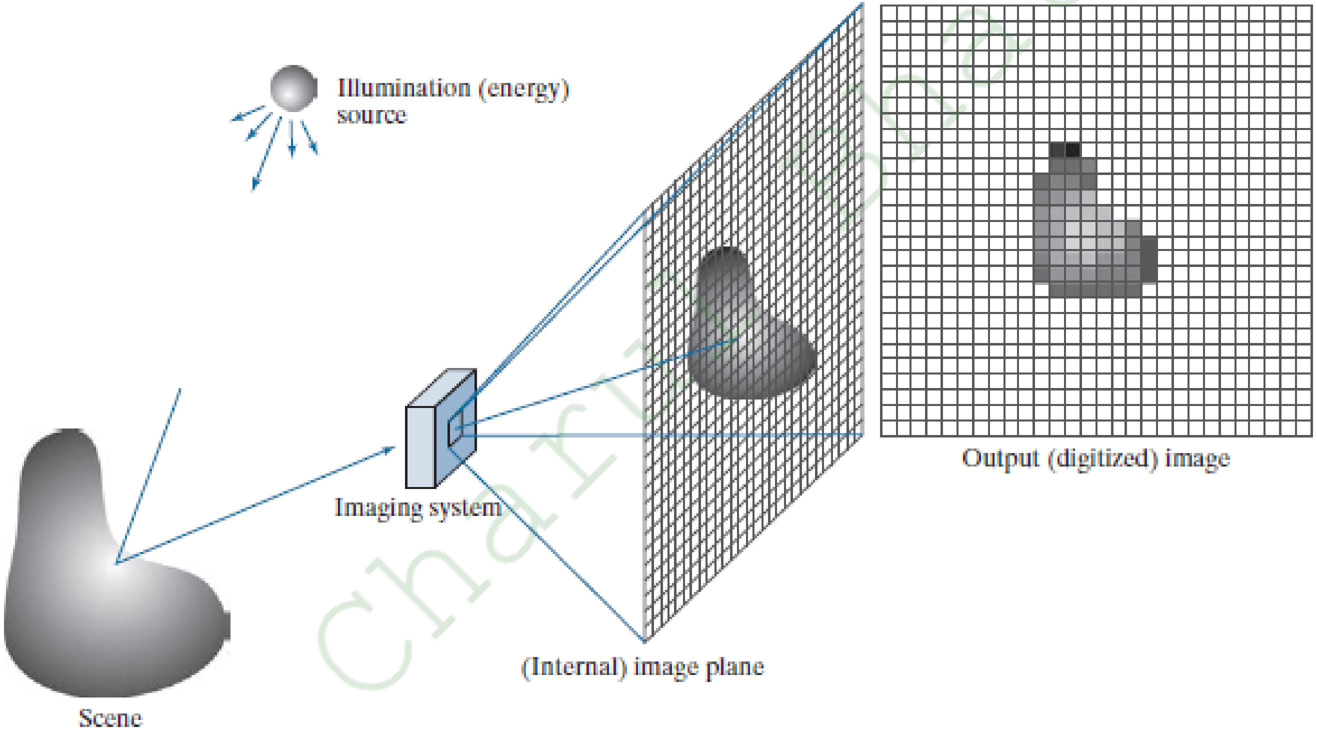
Line Sensor



Array Sensor

Single sensing element.
E.g. photodiode, whose output is a voltage proportional to light intensity.

A SIMPLE IMAGE FORMATION MODEL



Function f(x, y)

Characterized by two components:

1. ***Illumination i(x, y)*** - the amount of source illumination incident on the scene being viewed, and
2. ***Reflectance r(x, y)*** - the amount of illumination reflected by the objects in the scene.

The two functions combine as a product to form $f(x, y)$

$$\mathbf{f(x, y) = i(x, y)r(x, y)}$$

where

$$0 \le i(x, y) < \infty$$

$$0 \le r(x, y) \le 1$$

Typical values of i(x, y)

- On a sunny day, illumination on earth's surface is 90,000 lm/m2
- On a cloudy day it is 10,000 lm/m2
- Full moon yields 0.01 lm/m2
- Commercial office yields 1000 lm/m2

Typical values of r(x, y)

- for black velvet – 0.01
- Stainless steel – 0.65
- Flat white wall paint – 0.90
- Snow – 0.93

Digital Image Representation

Gray scale images can be represented as a function of two variables, **f(x, y)**,
where f(x, y) is the brightness (or grayness or intensity) of the image at the coordinate (x, y)

In colour image, the intensity is measured in 3 wavelengths (red, green, blue), so

$$f(x, y) = [fR(x, y), fG(x, y), fB(x, y)].$$

Image Sampling & Quantization

Numerous ways to acquire images

- o/p of most sensors is a continuous voltage waveform whose amplitude & spatial behaviour are related to the physical phenomenon (e.g. brightness) being sensed.
- For an image to be processed by a computer, it must be represented by an appropriate discrete data structure (e.g. Matrix)
- Two important processes to convert continuous analog image into digital image – **Sampling & Quantization**

Discretization

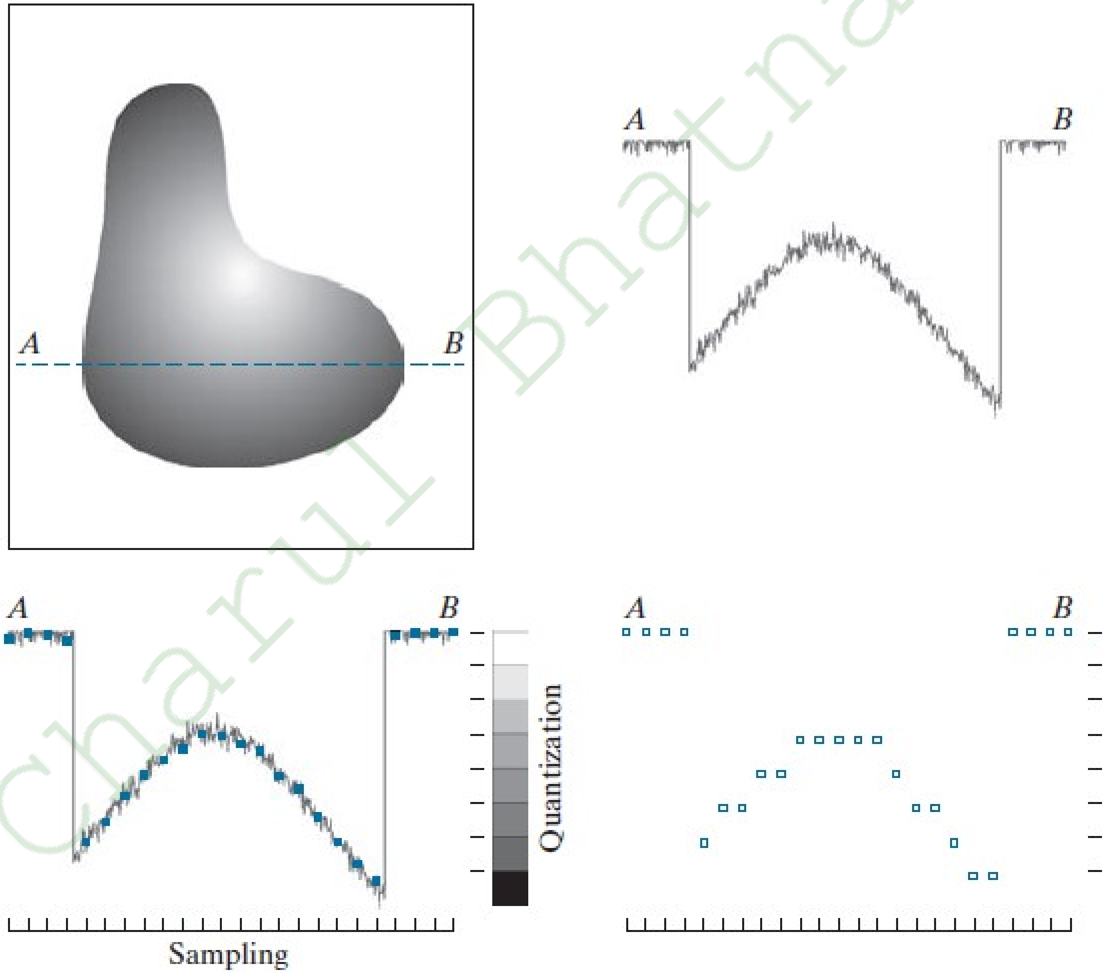
Process in which signals or data samples are considered at regular intervals.

Sampling

It is the discretization of image data in spatial coordinates.

Quantization

It is the discretization of image intensity (gray level) values.



a	b
c	d

FIGURE 2.16
(a) Continuous image. (b) A scan line showing intensity variations along line *AB* in the continuous image. (c) Sampling and quantization. (d) Digital scan line. (The black border in (a) is included for clarity. It is not part of the image).

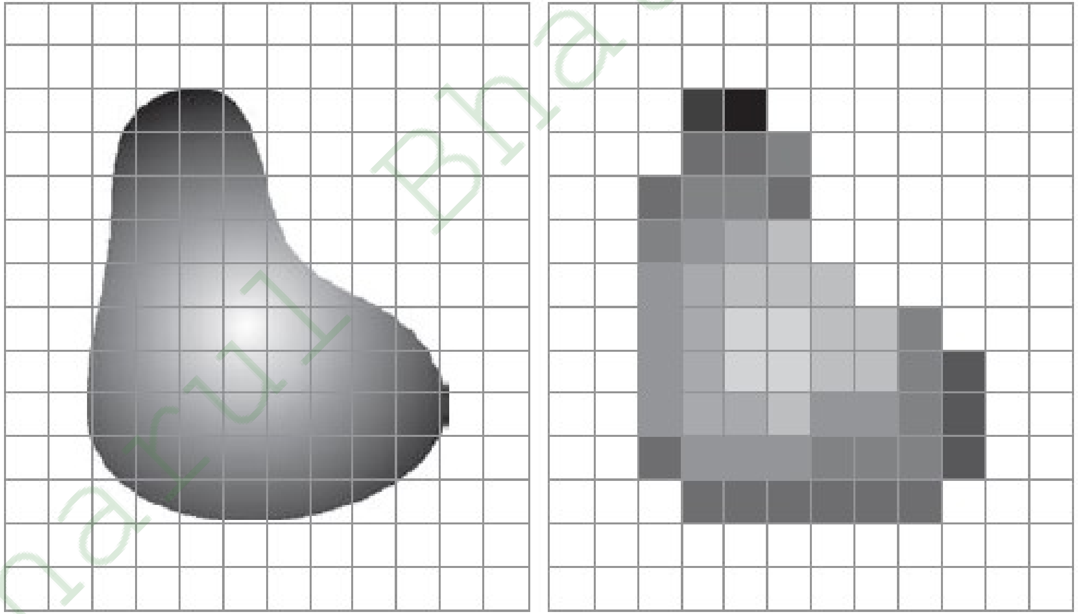
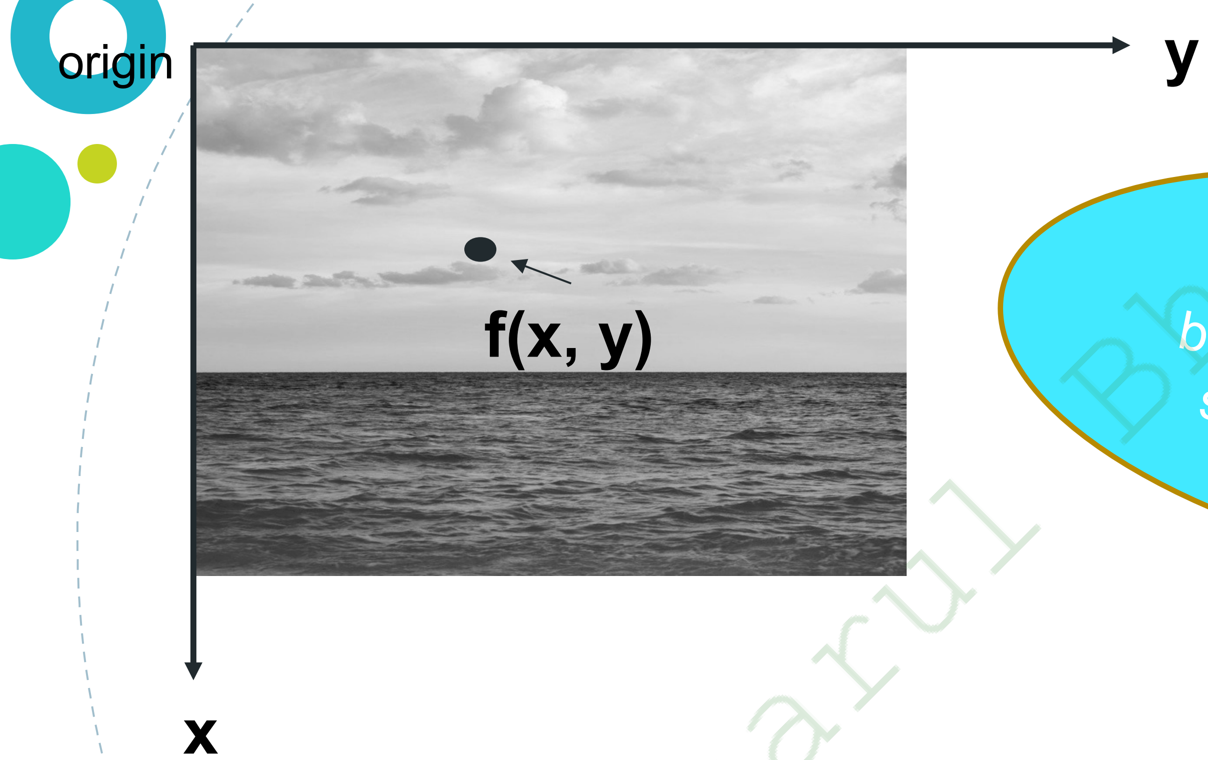


FIGURE 2.17
(a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.



A digital image is an image f(x, y) that has been discretized in both spatial coordinates & brightness.

A digital image can be considered as a matrix, where the matrix element value identifies the gray level at that point.

f(x, y) gives the **intensity** at position *(x, y)*

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

For digital images M, N and L (no. of gray levels) are mostly considered as powers of 2. Suppose

M = 29 , N = 210 & L = 2k = 28

Total number of bits b required to represent an image???

b = 512 x 1024 x 8 = 41,94,304 bits

Resolution of an Image

Spatial Resolution

It is the smallest discernible detail in an image

Gray-level Resolution

It is the smallest discernible change in the gray level of an image.

L-level digital image of size M x N

Spatial resolution – M x N pixels

Gray-level resolution – L levels

Fixed Gray-level
Varying number of samples

A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

(a) & (b)
virtually impossible to tell apart. The level of detail lost is too fine.

(c)
V slight fine checkerboard pattern in the borders & some graininess throughout the image

(d), (e), (f)
These effects much

(a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

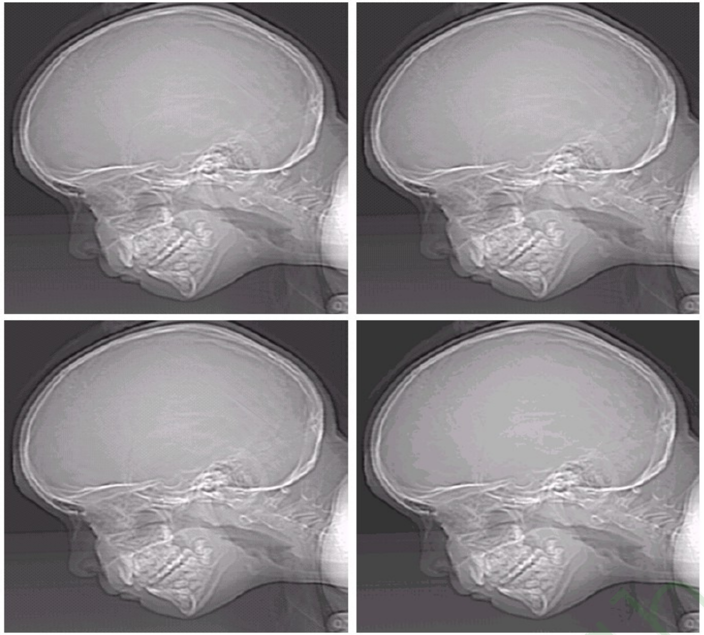
Checkerboard Effect

When the no. of pixels in an image is reduced keeping the no. of gray levels in the image constant, fine checkerboard patterns are found at the edges of the image. This effect is called the checker board effect.

Measuring Gray Level Resolution

- Common practice to refer to the number of bits used to quantize intensity as the “**intensity resolution**.”
- For example, it is common to say that an image whose intensity is quantized into 256 levels has 8 bits of intensity resolution.

False contouring

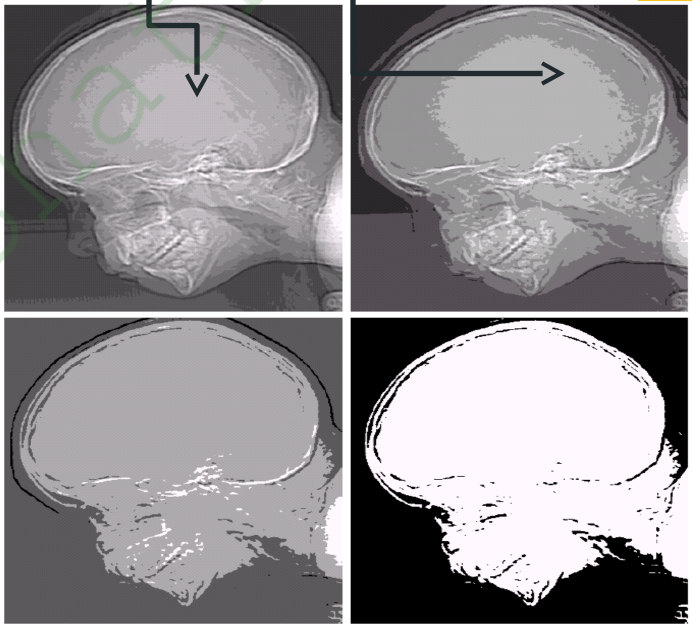


a b
c d

FIGURE 2.21
(a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

e f
g h

(e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)



False Contouring

When the no. of gray-levels in the image is low, the foreground details of the image merge with the background details of the image, causing ridge like structures.This degradation phenomenon is known as false contouring.

(The ridges resemble topographic contours in a map)

SOME BASIC RELATIONSHIPS BETWEEN PIXELS

Neighbours of a Pixel

NW	N	NE
W	**X**	E
SW	S	SE

N4 (p)

X3	X2	X1
X4	**p**	X0
X5	X6	X7

Pixel p at coordinates (x, y) has 4 horizontal & vertical neighbours.

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

ND (p)

Pixel p at coordinates (x, y) has 4 diagonal neighbours.

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

N4(p) ND(p) N8(p)

- If pixel p lies on the boundary, then some of the neighbours of p will lie outside the digital image.
- The set of image locations of the neighbors of a point p is called the neighborhood of p.
- The neighborhood is said to be closed if it contains p.
- Otherwise, the neighborhood is said to be open.

Adjacency

V

The set of gray-level values used to define adjacency

In a gray-scale image, V typically contains more than one elements. E.g. in an image of 265 gray-levels, V could be any subset of these 256 values.

In a binary image, V = {1} for finding the adjacency of a pixel with value 1.

Types of Adjacency

I) 4-adjacency

Two pixels p & q with values from V are 4-adjacent if q is in the set N4(p).

II) 8-adjacency

Two pixels p & q with values from V are 8-adjacent if q is in the set N8(p).

III) m-adjacency (mixed adjacency)

Two pixels p & q with values from V are m-adjacent if

a) q is in the set N4(p), or

b) q is in ND(p) and the set N4(p) ∩ N4(q) has no pixels whose values are from V.

Mixed adjacency is a modification of 8-adjacency. It is used to eliminate ambiguities that may arise when 8-adjacency is used.

$$\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Find 8-adjacency and m-adjacency of the pixel in the centre.

Note: V = {1}

$$\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

8 adjacency

$$\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

m - adjacency

Digital Path or Curve

From pixel p with coordinates (x0, y0) to pixel q with coordinates (xn, yn)

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

where pixels (x_i, y_i) & (x_{i-1}, y_{i-1}) are adjacent for $1 \le i \le n$

n is the length of the path

If (x0, y0) = (xn, yn) then the path is closed

- S – subset of pixels in an image.

Pixels p & q are said to be *connected in S*, if there exists a path between them consisting entirely of pixels in S.

For any pixel p in S, *the set of pixels* that are connected to it in S, is called the *connected component of S.*
If S has only one connected component, then S is called a *connected set.*

Region

R – subset of pixels in an image.

- R is a ***region*** of an image if **R is a connected set**
- Ri & Rj are ***adjacent*** if their union forms a connected set; otherwise the regions are said to be *disjoint*.

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

Are the two regions adjacent?

Distance Measures

Pixel	Coordinate
p	(x, y)
q	(s, t)

Euclidean distance between p & q

$$De(p, q) = [(x - s)2 + (y - t)2]1/2$$

City-block distance between p & q

$$D4(p, q) = |x - s| + |y - t|$$

Chessboard distance between p & q

$$D8(p, q) = \max(|x - s| , |y - t|)$$

Intensity Transformations and Spatial Filtering

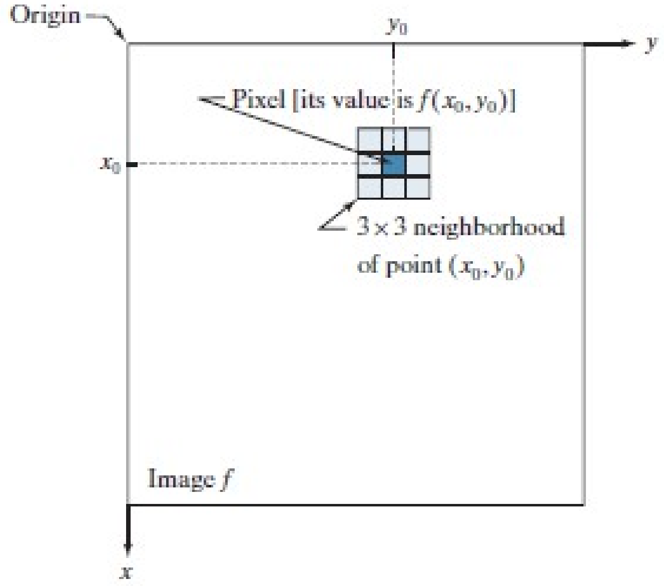
We will now see some Image Processing Techniques implemented in **Spatial Domain** (the plane containing the pixels of an image)

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

Output Image

Operator on f defined over a neighbourhood of point (x, y)

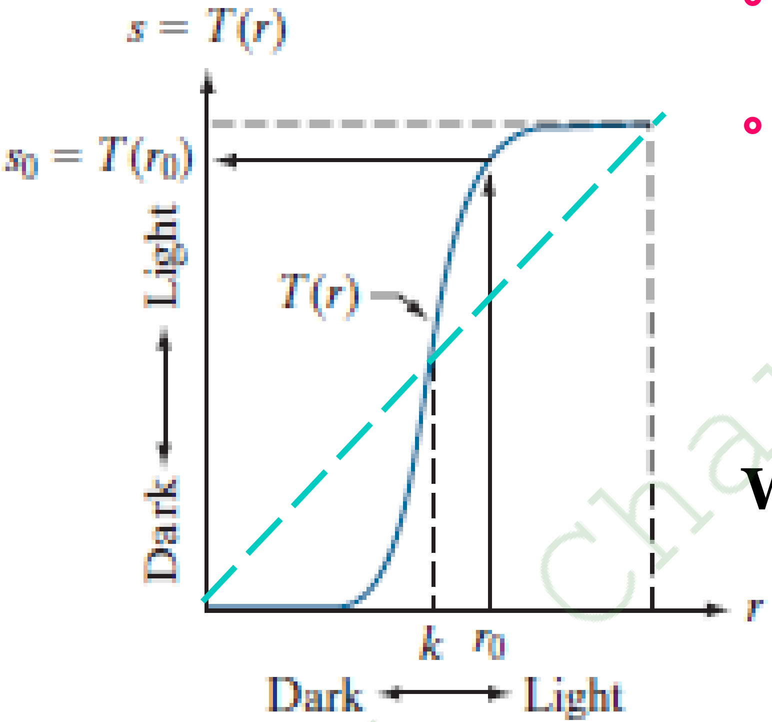
Input Image



The point (x_o, y_o) *is an arbitrary location in the* image

- The small region shown is a *neighborhood of* (x_o, y_o).
- Move the center of the neighborhood (x_o, y_o), from pixel to pixel, and apply the operator *T to the pixels in the neighborhood* to yield an output value at that location.
- For any specific location (x_o, y_o), the value of the output image *g* at those coordinates is equal to the result of applying T to the neighborhood with origin at (xo , yo) in f.

Intensity Transformation Function
(aka called a gray-level, or mapping)



- The smallest possible neighborhood is of size 1 × 1.
- In this case, *g depends only* on the value of *f at a single point (x, y)*

$$s = T(r)$$

What is this transformation doing?

Contrast Stretching

Principal Objective Of Enhancement

To process an image so that the result is more suitable than the original image for a specific application

- Enhance otherwise hidden information
- Filter important image features
- Discard unimportant image features

Amongst the simplest of all image enhancement techniques

$s = T(r)$

- **Linear transformations**
  - Image negative
- **Non-linear transformations**
  - Logarithmic transformation
  - Power Law (Exponential) transformation
- **Piecewise-linear transformations**
  - Contrast stretching
  - Gray-level slicing
  - Bit plane slicing

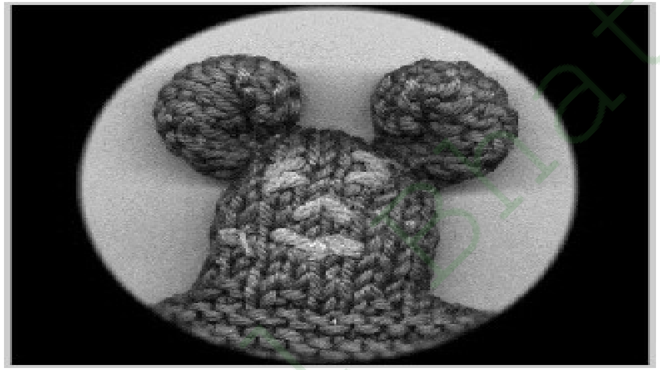
1. Image Negatives

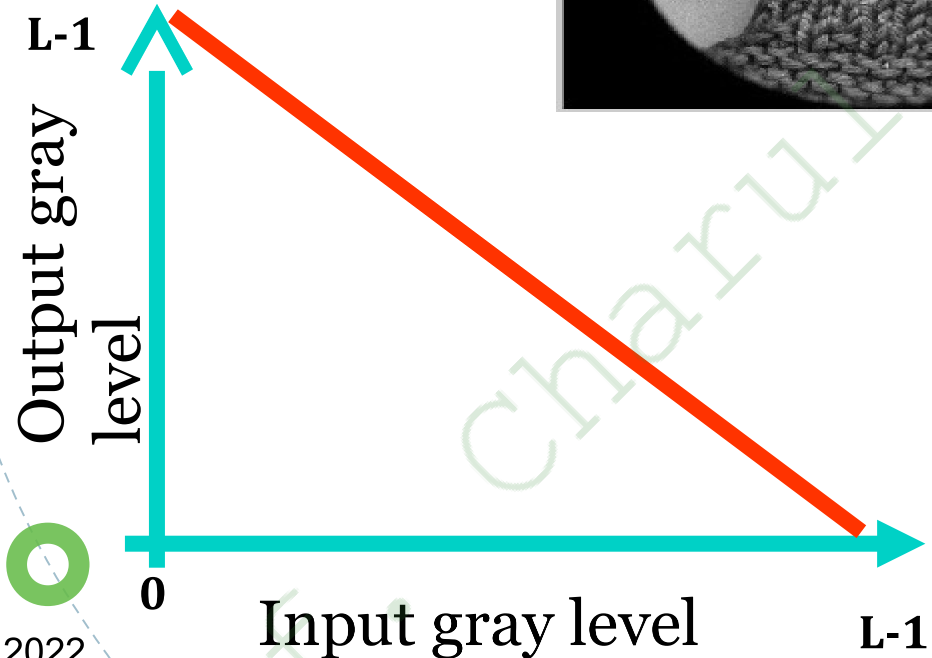
- Gray levels of the original image in the range of [0, L-1]
- Then the negative of an image can be obtained by reversing the intensity levels, the expression for which is given as:

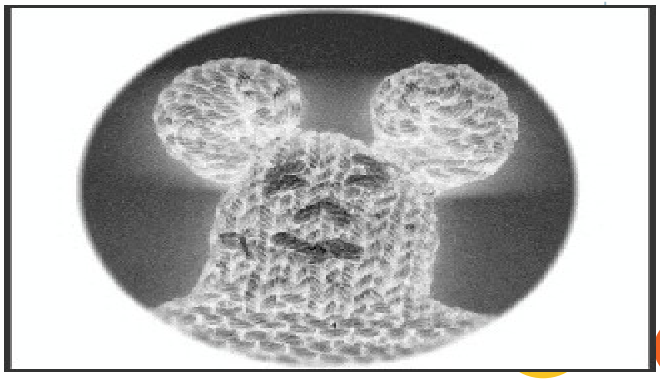
$$s = L - 1 - r$$

- This type of processing is suited for enhancing white or gray detail embedded in dark regions of an image, especially when black areas are dominant.

Image Negatives







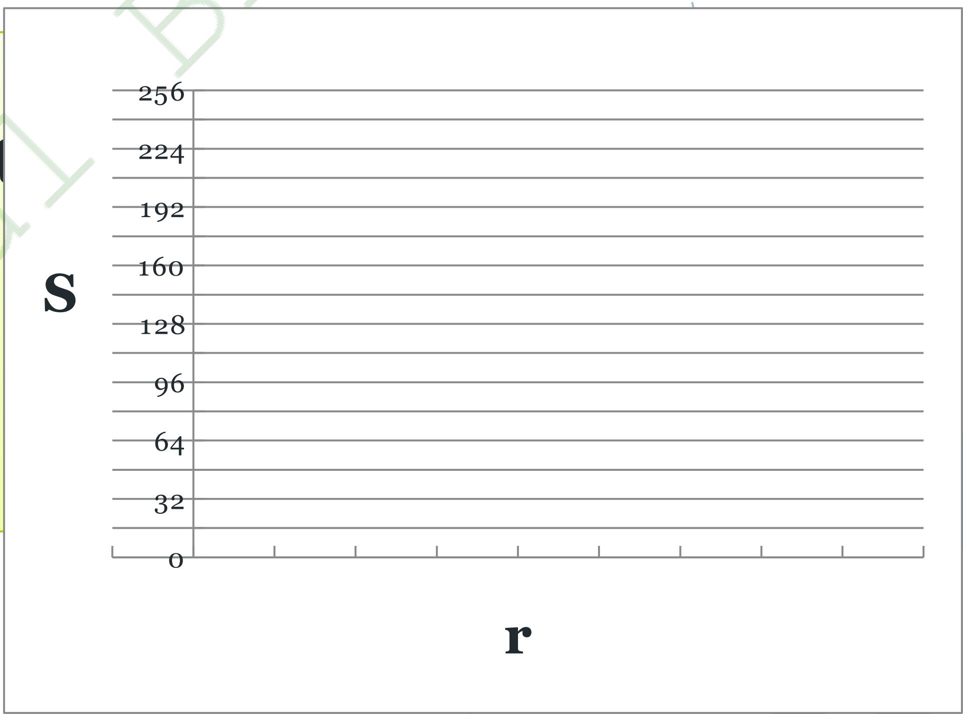
2. Log Transformations

General form:

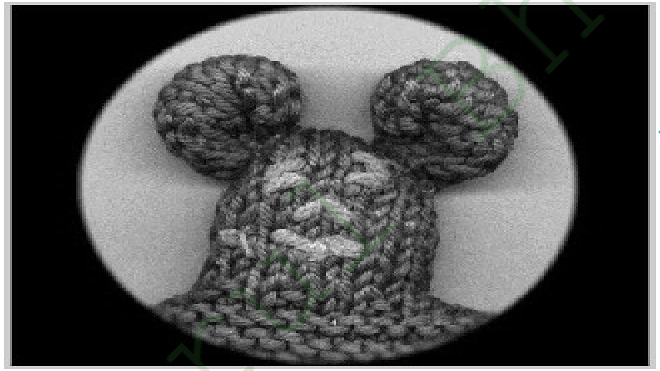
$$s = c \log (1 + r)$$

where c is a constant & $r \geq 0$.

- ✓ Maps a narrow range of low gray levels in the i/p image int wider range of o/p levels.
- ✓ Opposite is true at higher values of i/p levels.



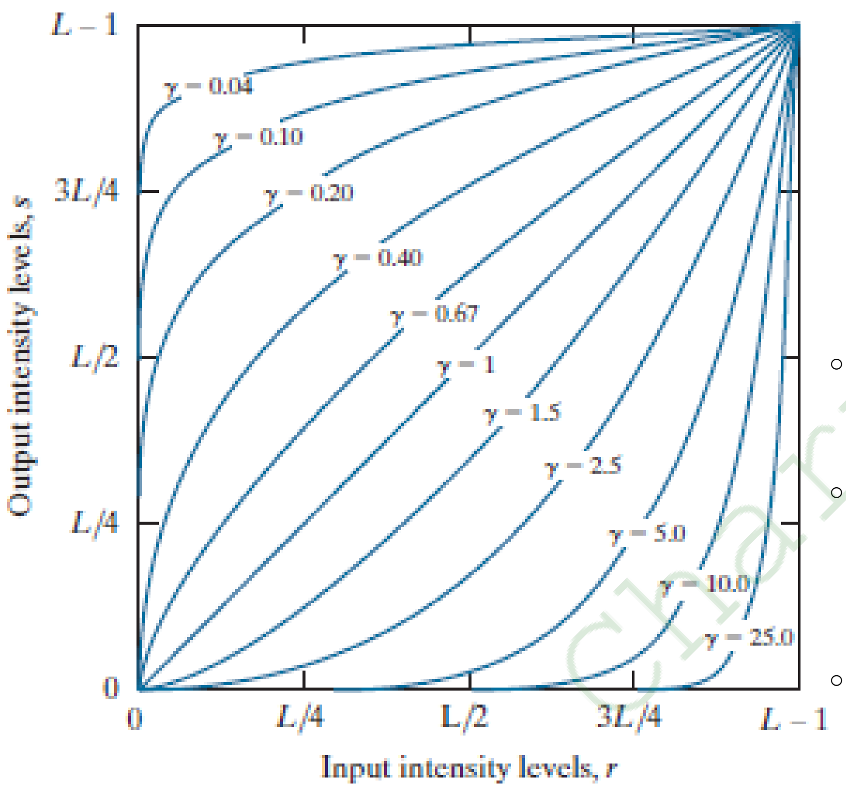
This type of processing is used to expand the values of dark pixels in an image while compressing the higher level values.



InvLog

Log

3. PowerLaw (Gamma) Transformations



General form:

$$s = c\ r^{\gamma}$$

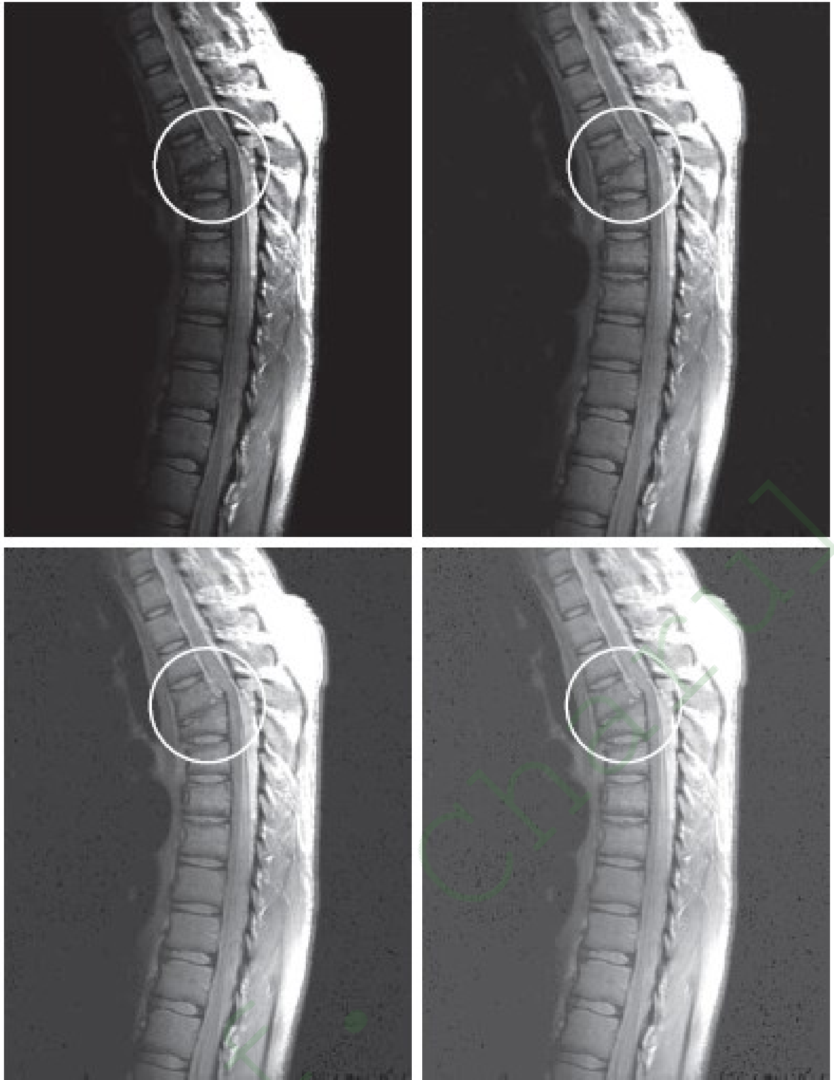
Where c & γ are positive constants.

When γ < 1

- map a narrow range of dark i/p values into wider range of o/p values.
- Opposite is true for higher i/p values.

When γ > 1

- Exactly the opposite effect as those generated with values of γ < 1.



(a) Magnetic resonance image (MRI) of a fractured human spine (the region of the fracture is enclosed by the circle). (b)–(d) Results of applying the Gamma Transformation with $c = 1$ *and* $\gamma = 0.6$, 0.4, and 0.3, respectively.

- Form of piecewise functions can be arbitrarily complex.
- Practical implementation of some important transformations can be formulated only as piecewise functions

- Their specification requires considerably more user i/p.

4. Contrast Stretching

- Simplest of piecewise linear functions.
- Basic idea – to increase the dynamic range of the gray levels in the image being processed.

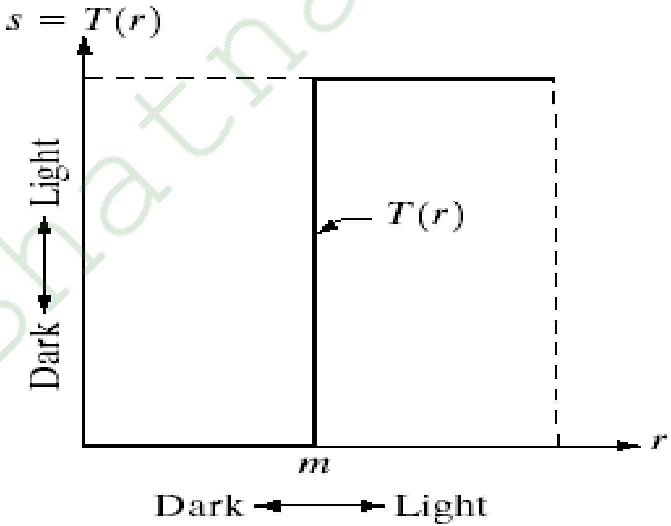
During image acquisition, low contrast images may result due to

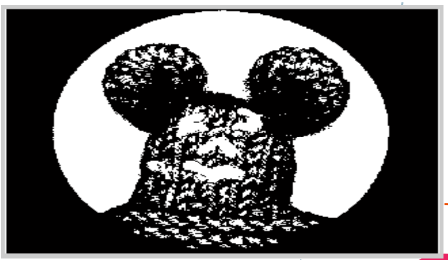
- Poor illumination
- Lack of dynamic range in image sensor
- Wrong setting of the lens aperture

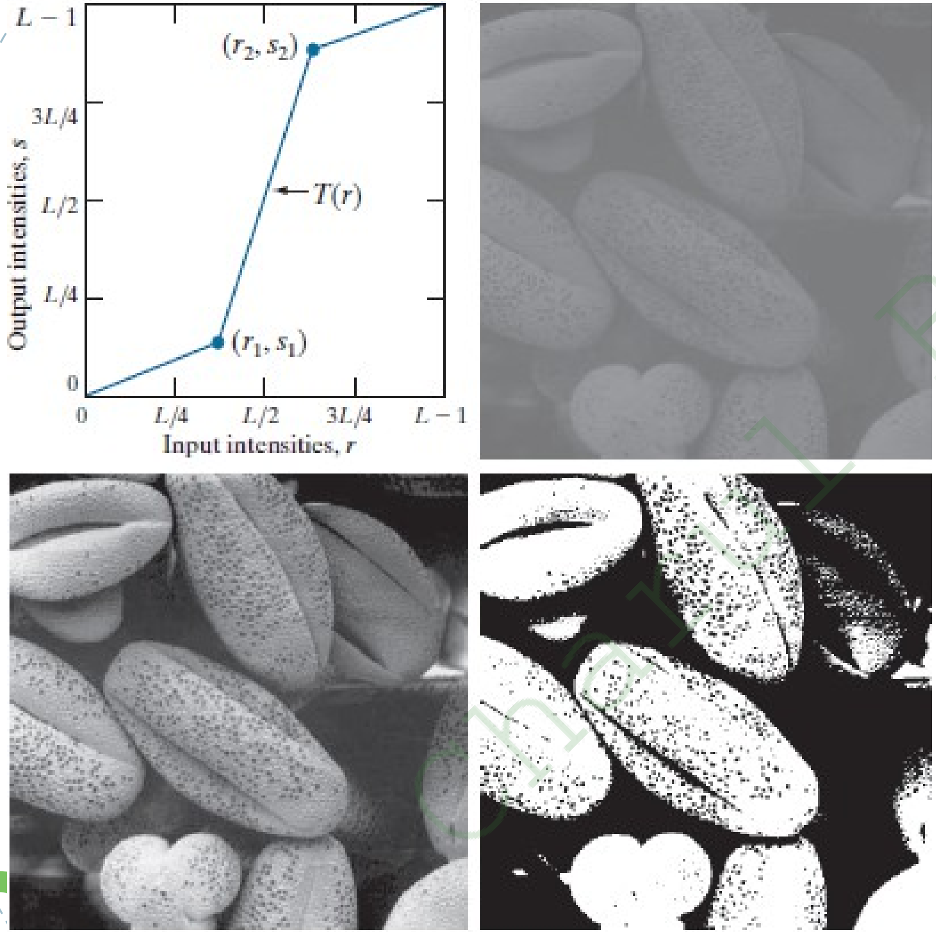
Thresholding Function

m = ‘threshold level’

$$s = \begin{cases} L-1 & \text{if } r > m \\ 0 & \text{otherwise} \end{cases}$$







Contrast stretching.

(a) Piecewise linear Transformation function.

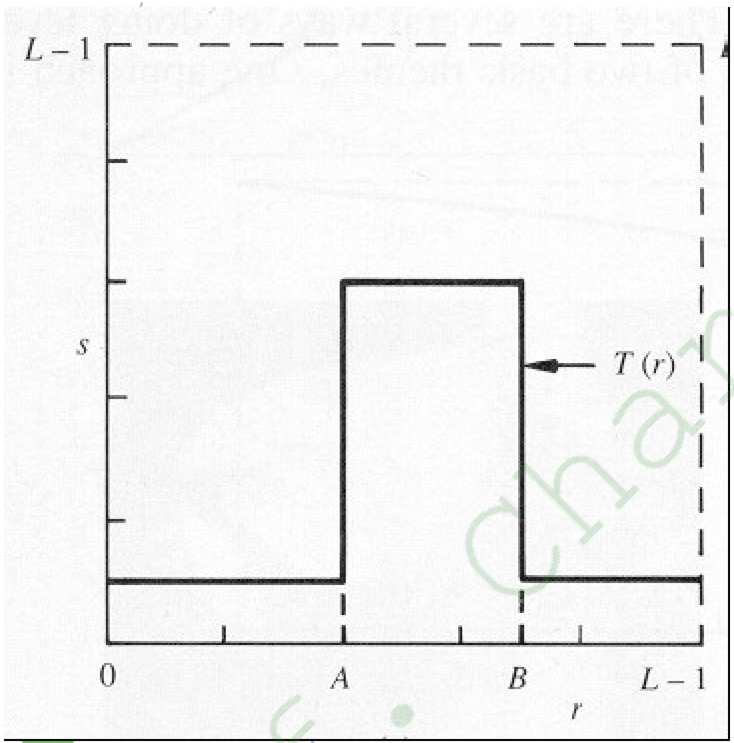
(b) A low contrast electron microscope image of pollen, magnified 700 times.

(c) Result of contrast stretching.

(d) Result of thresholding.

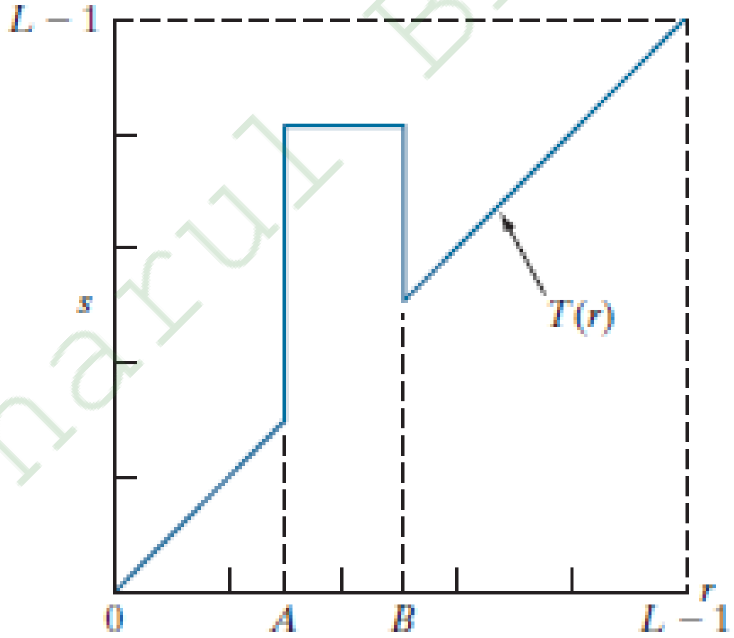
5. Gray Level / Intensity-Level Slicing

Used to highlight a specific range of gray values

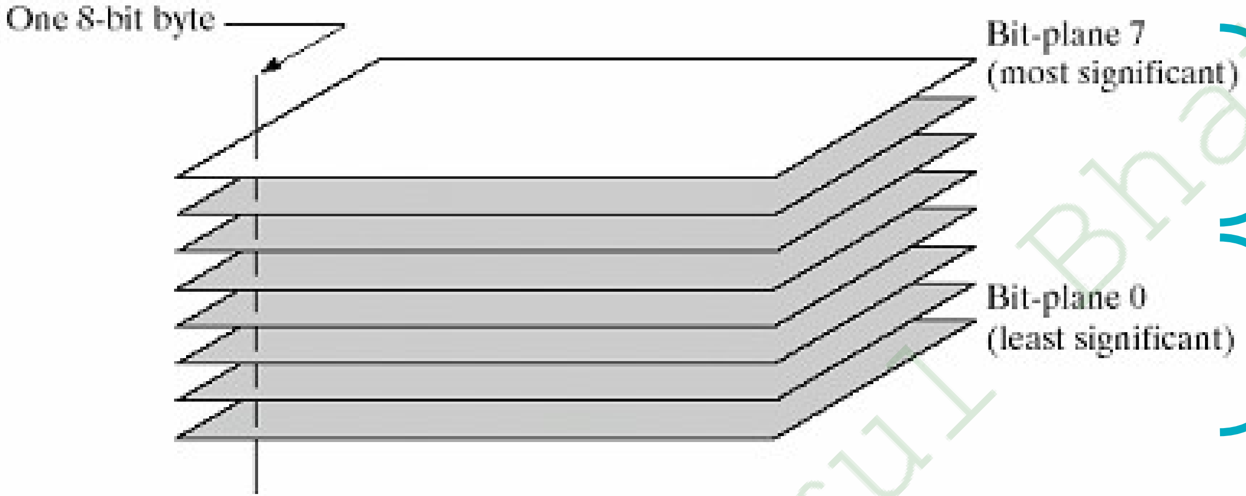


One way is to display a high value for all gray levels in the range of interest and a low value for all other gray levels (binary image).

What should we do to brighten the desired range of gray levels and leave all other intensities unchanged?



6. Bit Plane Slicing



Visually significant data

Subtle details

Extracts the information of a single bit plane

Instead of highlighting gray-level ranges, highlighting the contribution made to the total image appearance by specific bits might be desired.

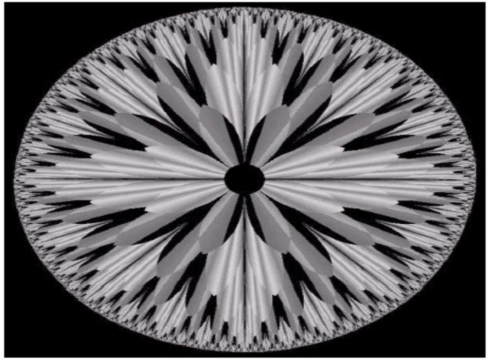


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

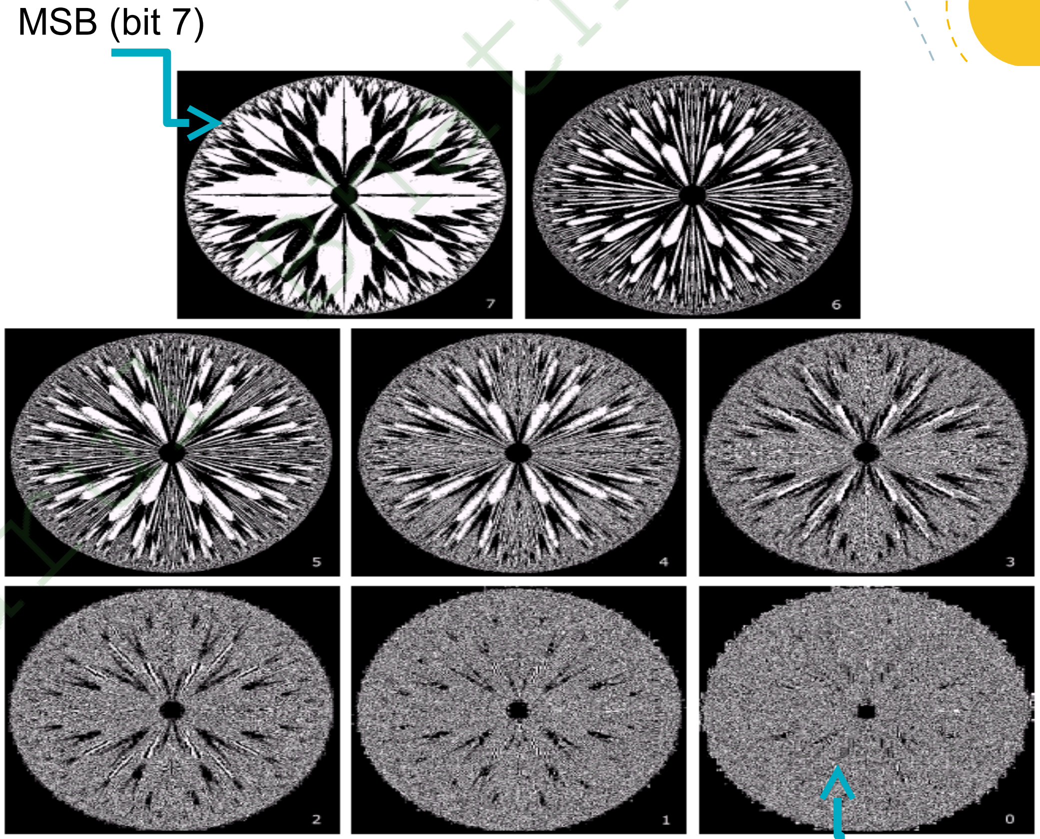


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

Histograms

The **histogram** of a digital image with grey levels from 0 to L-1 is a discrete function

$$\mathbf{h(rk) \;=\; nk}$$

or for a **normalized Histogram** it is

$$\mathbf{p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN}}$$

where:

- rk is the k-th grey level
- nk is the no. of pixels in the image with that grey level
- M, N are the no. of rows & columns in the image
- k = 0, 1, 2, …, L-1

Discrete values:

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

$$= (L-1)\sum_{j=0}^{k} \frac{n_j}{MN} = \frac{L-1}{MN}\sum_{j=0}^{k} n_j \qquad k=0,1,\ldots, L-1$$

M – no. of rows
N – no. of columns
MN – total no. of pixels in the image
L – no. of gray levels

Example: Histogram Equalization

Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in following table.

Get the histogram equalization transformation function

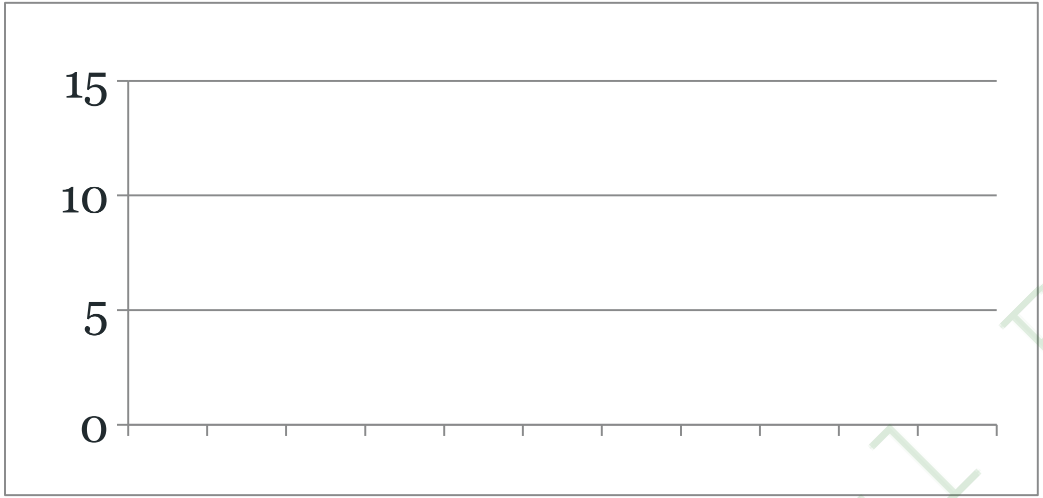
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

I/p Gray Level (r_k)	no. of pixels (n_k)	$p(r_k) = n_k/MN$	Σ	(L-1)Σ (sk)	O/p Gray Level
0	790	0.19	0.19	1.33	1
1	1023	0.25	0.44	3.08	3
2	850	0.21	0.65	4.55	5
3	656	0.16	0.81	5.67	6
4	329	0.08	0.89	6.23	6
5	245	0.06	0.95	6.65	7
6	122	0.03	0.98	6.86	7
7	81	0.02	1.00	7.00	7

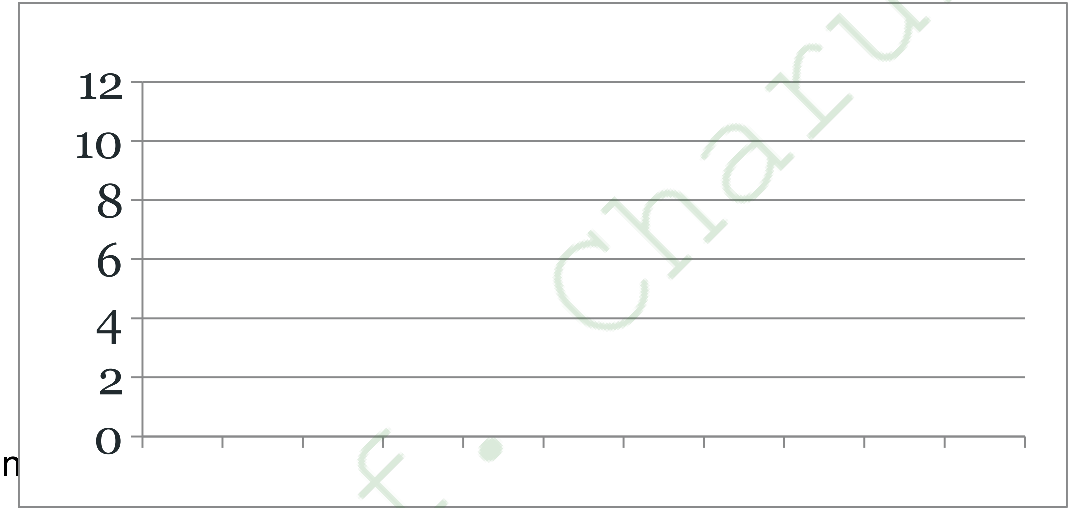
Grey level	nk	pr(rk)	sk
0	790	0.19	1
1	1023	0.25	3
2	850	0.21	5
3	656	0.16	6
4	329	0.08	6
5	245	0.06	7
6	122	0.03	7
7	81	0.02	7

What will be the histogram of the new image?

Equalized Grey level	nk	
0	0	
1	790	
2	0	
3	1023	
4	0	
5	850	
6	656 + 329 =	985
7	245 + 122 + 81 =	448



Histogram of the dark image



Equalized Histogram of the image

Histogram Matching / Specification

- There are applications in which attempting to base enhancement on a uniform histogram is not the best approach
- Sometimes it is useful to be able to specify the shape of the histogram that we wish the processed image to have.
- For a image, whose enhancement is to be done, we are **given an histogram, G(zk),** that actually shows **how the processed image's histogram should look** after applying the transformation function to the i/p image.

Steps for Histogram Matching

1. Equalize the histogram of the input image. Round the resulting values, sk , to the integer range [0, L – 1].

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

2. Equalize the image whose histogram we want to match. Round the resulting values, vk , to the integer range [0, L – 1]. Store the rounded values of G in a lookup table.

$$v_k = G(z_k) = \sum_{j=0}^{k} p_z(z_j)$$

3. For every value of s_k , $k = 0,1, 2,\ldots, L-1$, use the stored values of G to find the corresponding value of z_q so that $G(z_q)$ is closest to s_k . Store these mappings from s to z. When more than one value of z_q gives the same match (i.e., the mapping is not unique), choose the smallest value by convention.

4. Form the histogram-specified image by mapping every equalized pixel with value s_k to the corresponding pixel with value z_q in the histogram-specified image, using the mappings

Example: Histogram Matching

- Suppose that a 3-bit image (L=8) of size 64 × 64 pixels (MN = 4096) has the intensity distribution shown in the following table (on the left).
- Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

1. Equalize the histogram of the input image. Round the resulting values, sk , to the integer range [0, L – 1].

I/p Gray Level (rk)	no. of pixels (nk)	p(rk) = nk/MN	Σ	(L-1)Σ	O/p Gray Level (s)
0	790	0.19	0.19	1.33	1
1	1023	0.25	0.44	3.08	3
2	850	0.21	0.65	4.55	5
3	656	0.16	0.81	5.67	6
4	329	0.08	0.89	6.23	6
5	245	0.06	0.95	6.65	7
6	122	0.03	0.98	6.86	7
7	81	0.02	1.00	7.00	7

2. Equalize the image whose histogram we want to match. Round the resulting values, v_k, to the integer range [0, L – 1]. Store the rounded values of G in a lookup table.

Gray Level (z_k)	$p(z_k) = n_k/MN$	Σ	$(L-1)\Sigma$	O/p Gray Level $G(z_k)$
0	0.00			
1	0.00			
2	0.00			
3	0.15			
4	0.20			
5	0.30			
6	0.20			
7	0.15			

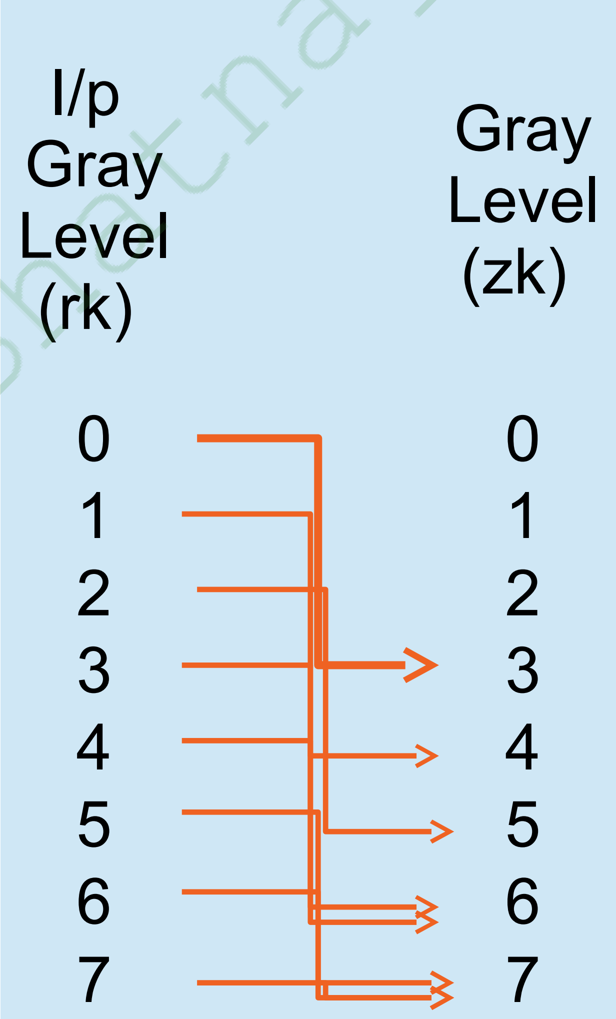
3. For every value of sk , k = 0,1, 2,…, L– 1, use the stored values of G to find the corresponding value of zq so that G (zq) is closest to sk . Store these mappings from s to z. When more than one value of zq gives the same match (i.e., the mapping is not unique), choose the smallest value by convention.

I/p Gray Level (rk)	O/p Gray Level (sk)
0	1
1	3
2	5
3	6
4	6
5	7
6	7
7	7

O/p Gray Level G(zk)	Gray Level (zk)
0	0
0	1
0	2
1	3
2	4
5	5
6	6
7	7

I/p Gray Level (r_k)	O/p Gray Level (s_k)
0	1
1	3
2	5
3	6
4	6
5	7
6	7
7	7

O/p Gray Level $G(z_k)$	Gray Level (z_k)
0	0
0	1
0	2
1	3
2	4
5	5
6	6
7	7



Histogram Matching

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

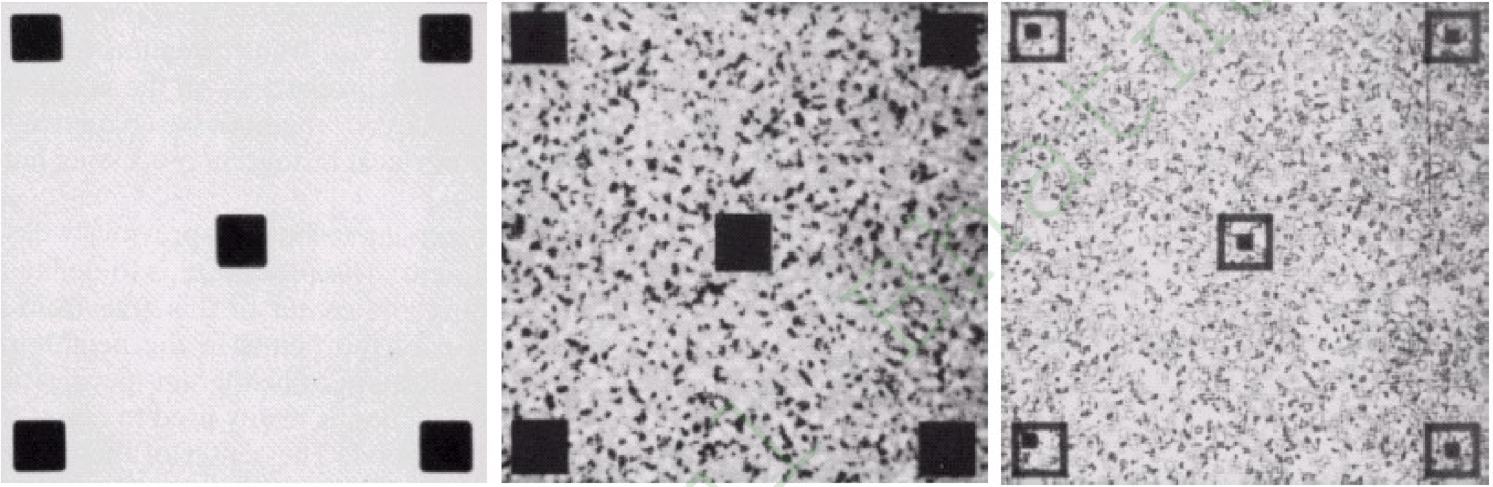
rk	zq	No. of pixels
0	3	
1	4	
2	5	
3	6	
4	6	
5	7	
6	7	
7	7	

Local Enhancement

- The histogram processing methods discussed are *global*
- Suitable for overall enhancement, but generally fails when the objective is to enhance details over small areas in an image.
  - the number of pixels in small areas have negligible influence on the computation of global transformations.
- The solution is to devise transformation functions based on the intensity distribution of pixel neighborhoods.

Local Enhancement

- Define a neighborhood (e.g. a square or rectangle) and move its center from pixel to pixel.
- At each location, the histogram of the points in the neighborhood is computed. Either histogram equalization or histogram specification transformation function is obtained
- Map the intensity of the pixel centered in the neighborhood
- Move to the next location and repeat the procedure



a b c

(a) Original image.

(b) Result of global histogram equalization.

(c) Result of local histogram equalization using a 7 x 7 neighbourhood about each pixel.

Enhancement using Arithmetic/Logic Operations

Arithmetic/Logic operations are performed on the pixels of two or more images

Image Subtraction

$$g(x, y) = f(x, y) - h(x, y)$$

Obtained by computing the difference between all pairs of corresponding pixels from images f & h.

Subtraction enhances the differences between the two images

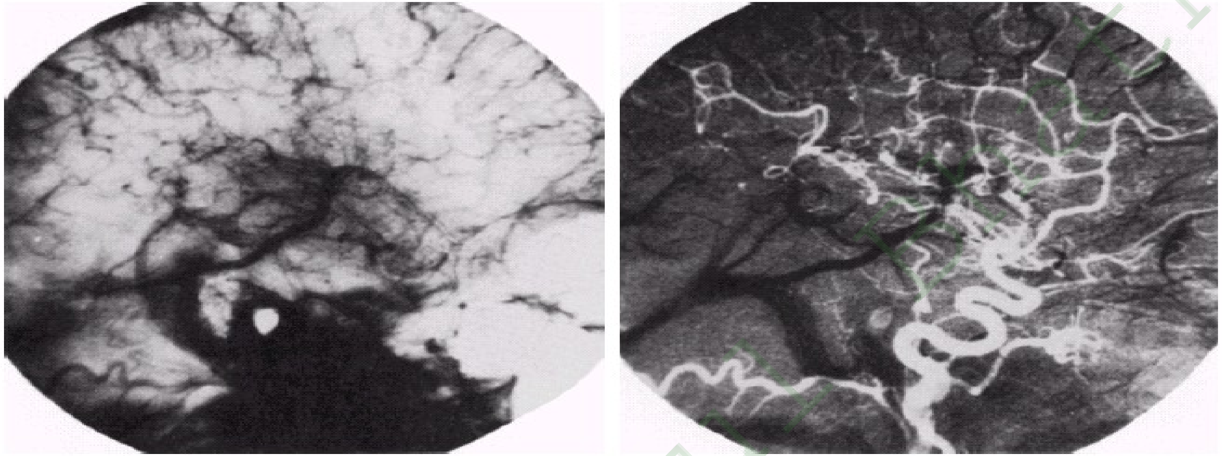
Mask Mode Radiography

One of the most successful commercial applications of image subtraction

Example: imaging blood vessels and arteries in a body. Blood stream is injected with a dye and X-ray images are taken before and after the injection

– f(x, y): image after injecting a dye
– h(x, y): image before injecting the dye (the mask)

• The difference of the 2 images yields a clear display of the blood flow paths.



a b

Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

(a) Mask image

(b) Image (after injection of dye into the bloodstream) with mask subtracted out

Image Averaging

- A noisy image:

$$g(x, y) = f(x, y) + n(x, y)$$

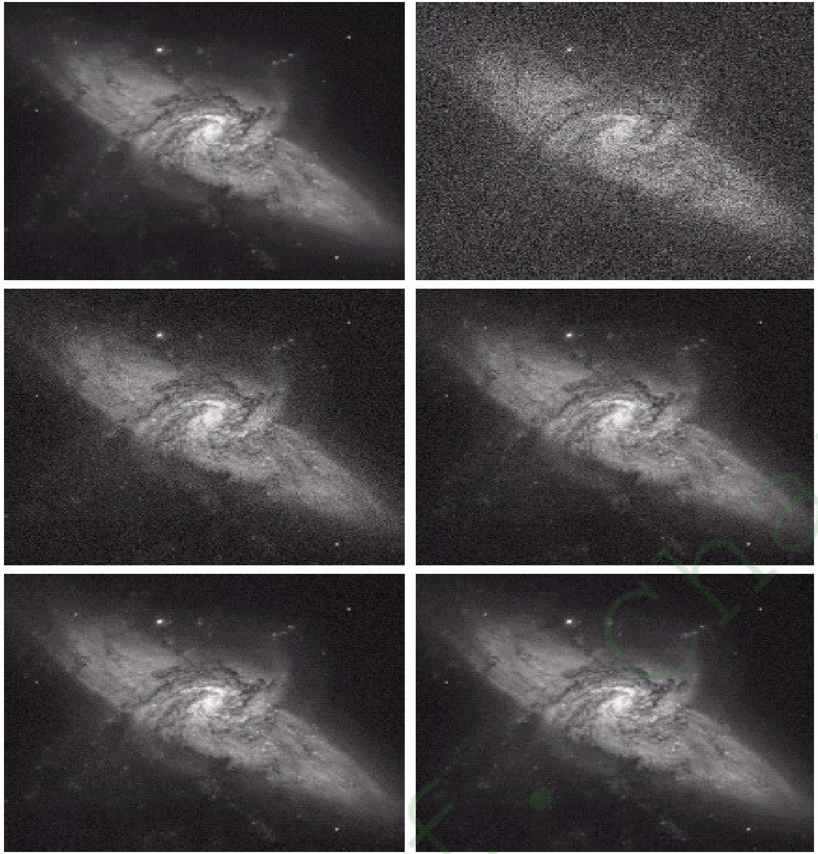
- Averaging M different noisy images:

$$\bar{g}(x, y) = \frac{1}{M}\sum_{i=1}^{M} g_i(x, y)$$

- As M increases, the variability of the pixel values at each location decreases.

- This means that g(x, y) approaches f(x, y) as the number of noisy images used in the averaging process increases.

An important application of image averaging is in the field of astronomy



a	b
c	d
e	f

(a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64$, and 128 noisy images. (Original image courtesy of NASA.)

Image Division

Shading Correction is an important applications of image division

An imaging sensor produces image g(x, y), where

$$g(x, y) = f(x, y)\, h(x, y)$$

By dividing g(x, y) by h(x, y),
we can get the perfect image.

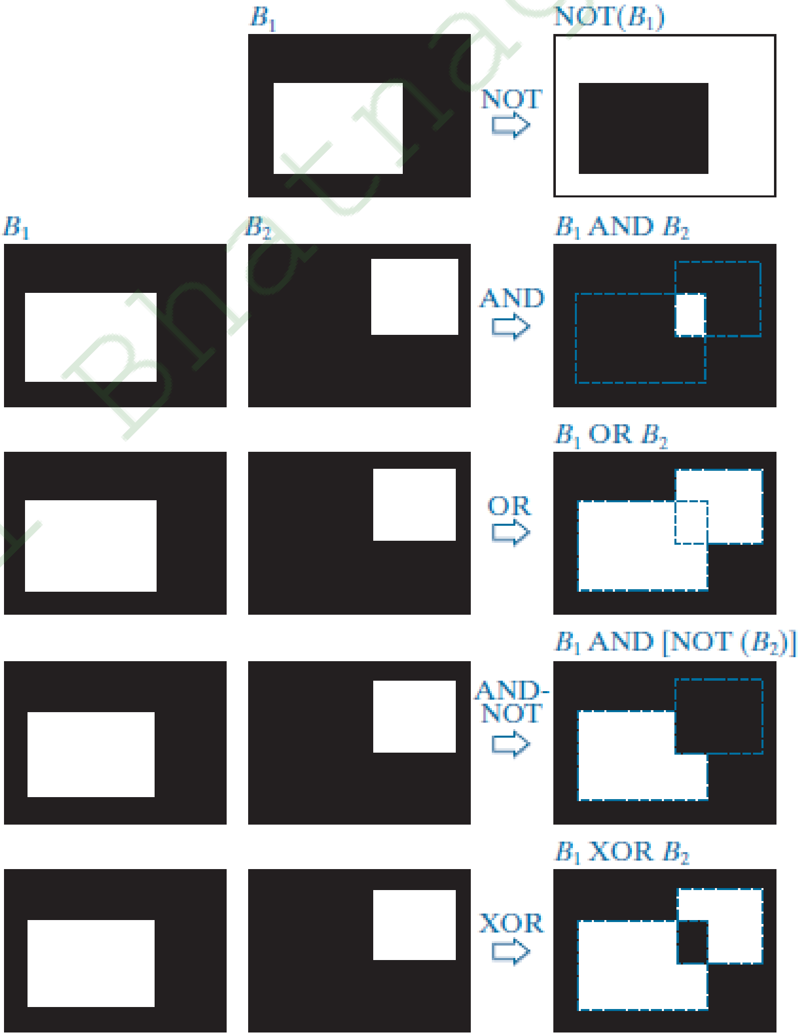
Shading correction.
(a) Shaded test pattern.
(b) Estimated shading pattern.
(c) Division of (a) by (b).

Image Multiplication

An important use of image multiplication is in *masking operations, also called region of interest (ROI).*

Logical Operations

- Illustration of logical operations involving foreground (white) pixels.
- Black represents binary 0's and white binary 1's.
- The dashed lines are shown for reference only. They are not part of the result.



Spatial Filtering

Spatial filtering is the filtering operations that are performed directly on the pixels of an image

If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter.

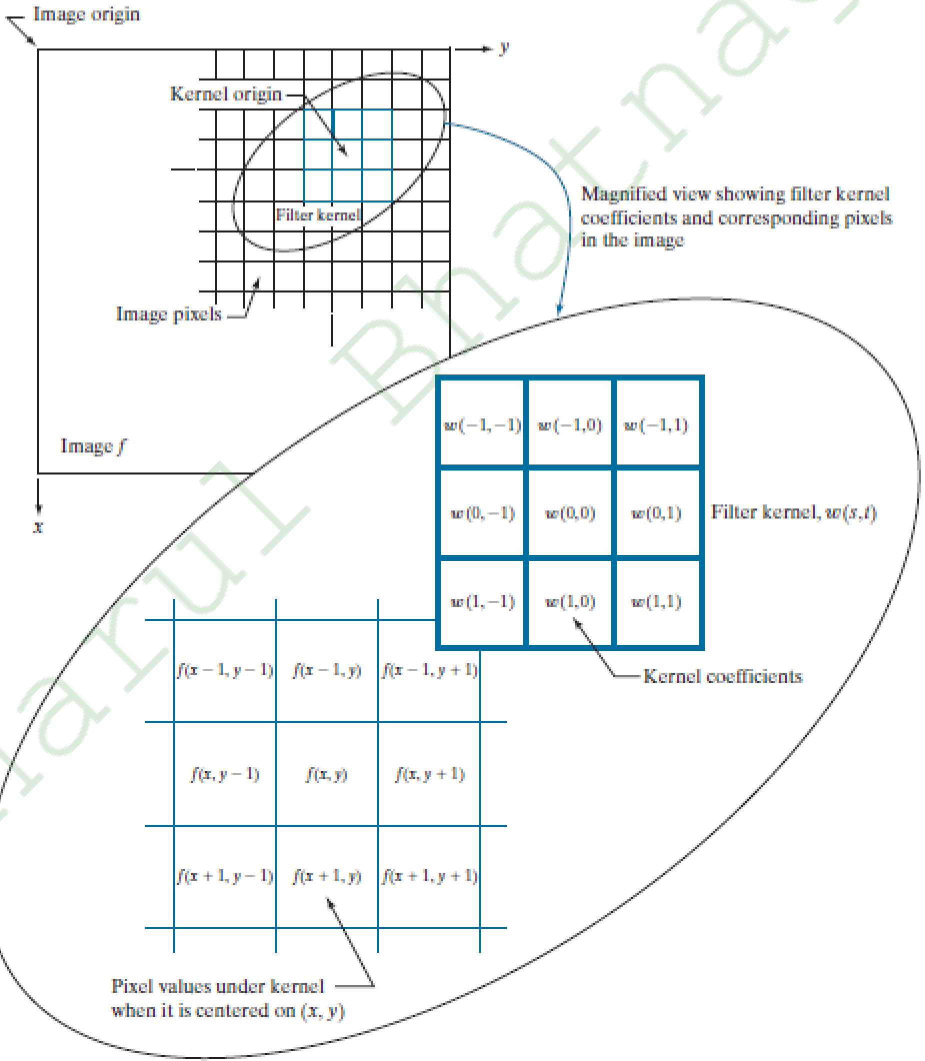
Otherwise, the filter is a nonlinear spatial filter.

THE MECHANICS OF LINEAR SPATIAL FILTERING

- A linear spatial filter performs a sum-of-products operation between an image f and a filter kernel, w.
- The kernel is an array whose size defines the neighborhood of operation, and whose coefficients determine the nature of the filter.
- At any point (x, y) in the image, the response, g(x, y), of the filter is the sum of products of the kernel coefficients and the image pixels encompassed by the kernel:

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

- The mechanics of linear spatial filtering using a 3×3 kernel.
- The origin of the image is at the top left
- The origin of the kernel is at its center.
- Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.



Note: Linear filtering

- Coefficie e f(x, y), indicatin
- For a ma 2a+1 and n=2b+1, er. Then m and n are
- In general, linear filtering of an image f of size M x N with a filter mask of size m x n is given by the expression

To generate a complete filtered image, the following equation must be applied for

x = 0, 1, …, M-1

y = 0, 1, …, N-1

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

a = (m – 1) / 2 & b = (n – 1) / 2

SPATIAL CORRELATION AND CONVOLUTION

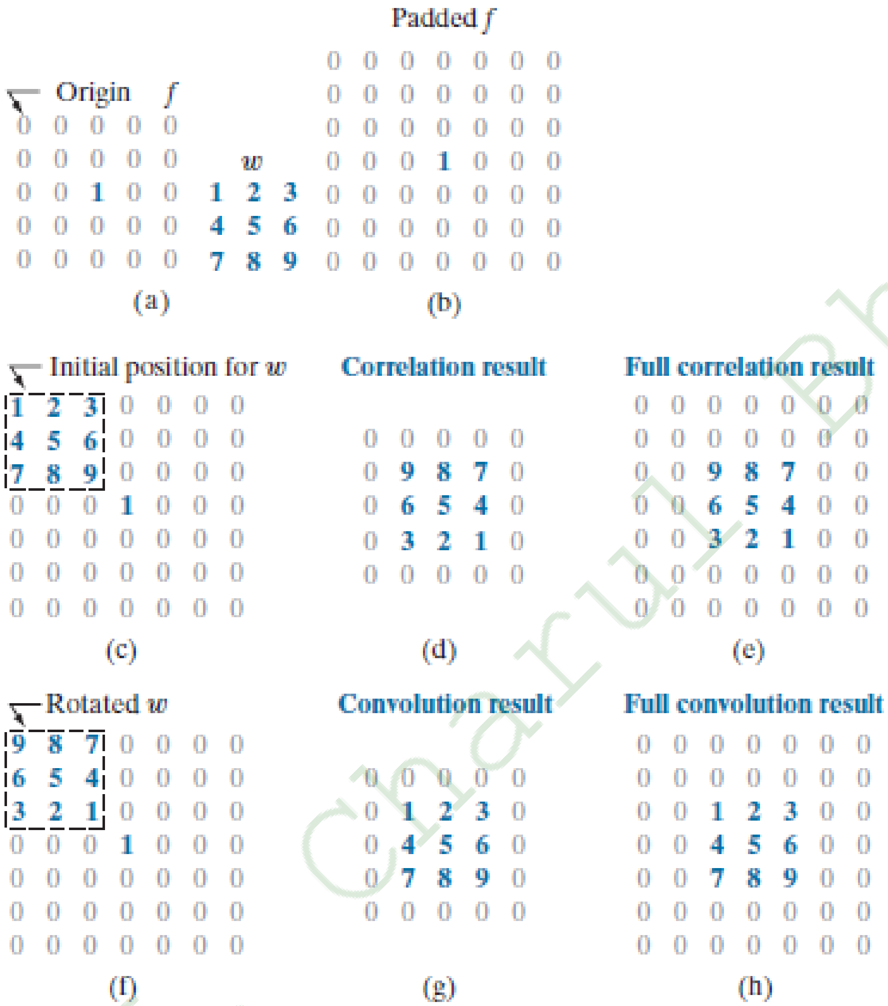
- Correlation consists of moving the center of a kernel over an image, and computing the sum of products at each location.
- The mechanics of spatial convolution are the same, except that the correlation kernel is rotated by 180°.
- When the values of a kernel are symmetric about its center, correlation and convolution yield the same result.
- Following is a 1-D illustration for Correlation

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

SPATIAL CORRELATION AND CONVOLUTION IN IMAGE

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

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



Correlation and Convolution of a 2-D kernel with an image

Smoothing Spatial Filters

Smoothing filters are used for blurring and for noise reduction.

- Blurring is used in preprocessing steps, such as removal of small details from an image prior to object extraction, and bridging of small gaps in lines or curves
- Noise reduction can be accomplished by blurring with a linear or non linear filter.

Types of Smoothing filters

There are 2 types of smoothing spatial filters

- Smoothing Linear Filters
- Order-Statistics Filters

Smoothing Linear Filters

aka Averaging Filters / Lowpass Filters

- Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.
- Sometimes called “averaging filters”.
- The idea is replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask.

Two 3x3 Smoothing Linear Filters

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

Standard average

$$\frac{1}{16}\times\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

Weighted average

Order-Statistics Filters

Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

Best-known "median filter"

Process of Median filter

	10	15	20	
	20	100	20	
	20	20	25	

10, 15, 20, 20, 20, 20, 20, 25, 100

↑

5th

- Crop region of neighborhood
- Sort the values of the pixel in the region
- In the M x N mask the median is (M x N +1) div 2

Median Filters are quite popular …

Max & Min Filters

Using the 100th percentile results gives us the ***max filter***, which is useful in finding the brightest points in the image. The response of a 3 x 3 max filter is given by

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

The 0th percentile filter is the ***min filter***, used for the opposite purpose.

Sharpening Spatial Filters

The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as an natural effect of a particular method of image acquisition.

- As we know that blurring can be done in spatial domain by **pixel averaging in a neighbors**

- **Averaging is analogous to integration**

- Therefore, logically it can be concluded that the **sharpening must be accomplished** by **spatial differentiation.**

Sharpening Filters based on first- and second-order derivatives

Definition for a first order derivative

- Must be zero in areas of constant intensity
- Must be nonzero at the onset of an intensity step or ramp
- Must be nonzero along intensity ramps

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

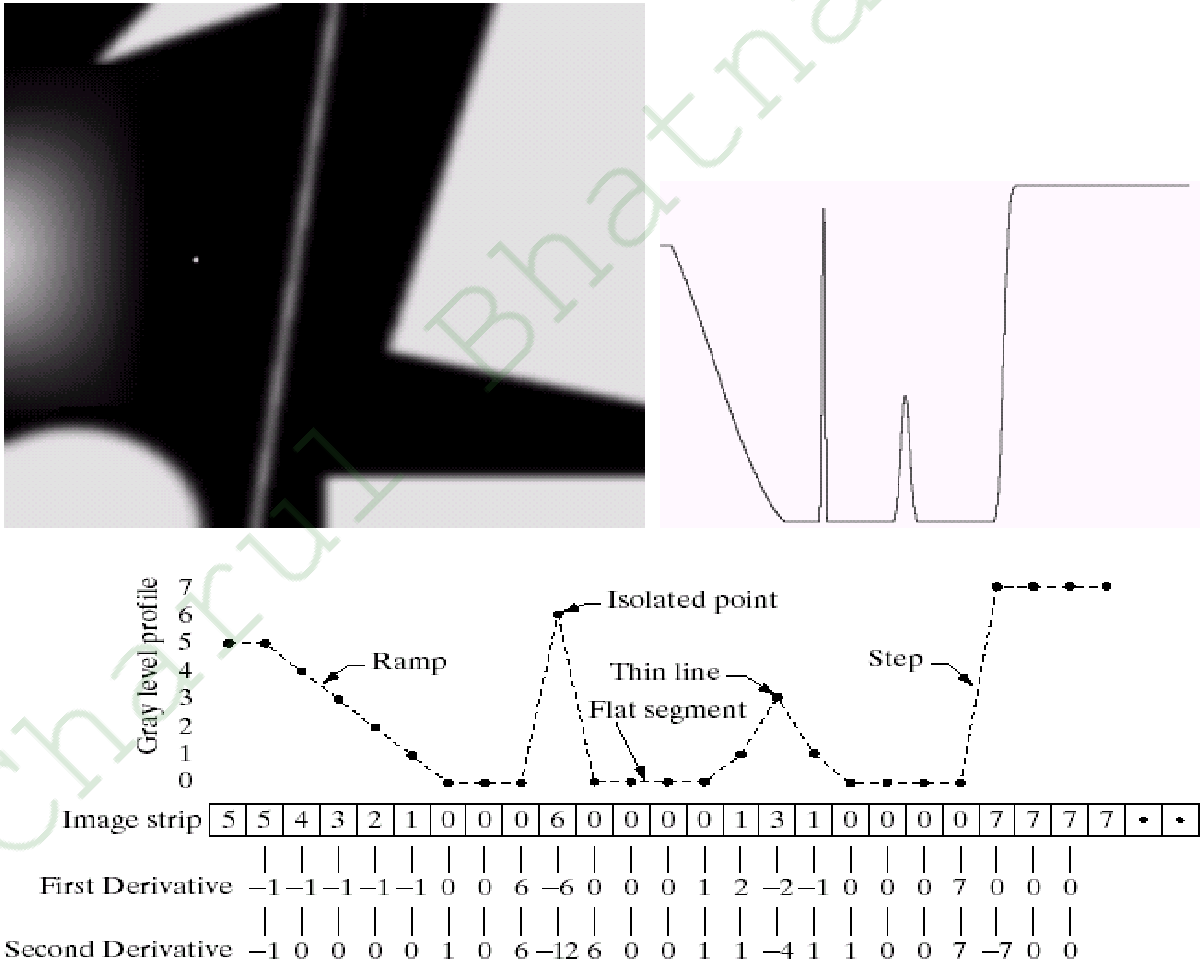
Definition for a second order derivative

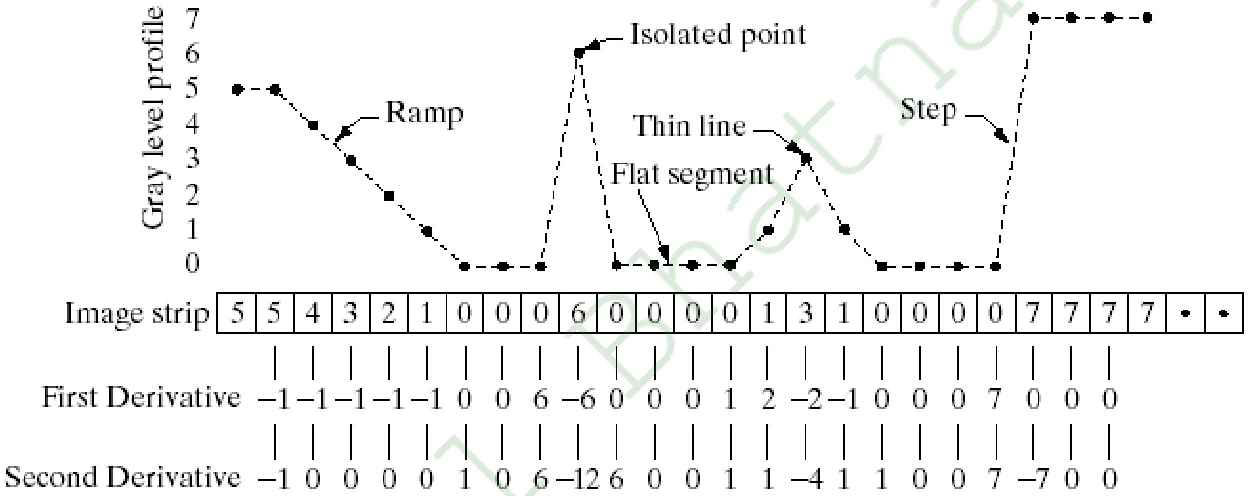
- Must be zero in areas of constant intensity
- Must be nonzero at the onset and end of a intensity step or ramp
- Must be zero along intensity

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

a b
c

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





Gray level profile
7
6
5
4
3
2
1
0
Isolated point
Ramp
Thin line
Flat segment
Step
Image strip 5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7 • •
First Derivative −1 −1 −1 −1 −1 0 0 6 −6 0 0 0 1 2 −2 −1 0 0 0 7 0 0 0
Second Derivative −1 0 0 0 0 1 0 6 −12 6 0 0 1 1 −4 1 1 0 0 7 −7 0 0

1)

2)

3)

Summary

- First-order derivatives generally produce thicker edges in an images.
- Second-order derivatives have a stronger response to fine detail (sa, thin lines or isolated points).
- First-order derivatives generally have a stronger response to a gray-level step.
- Second-order derivatives produce a double response at step changes in gray level

THE LAPLACIAN

USING THE SECOND DERIVATIVE FOR IMAGE SHARPENING

The approach basically consists of the following steps:

1. Defining a discrete formulation of the second-order derivative.
2. Constructing a filter mask based on that formulation.

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

0	1	0
1	-4	1
0	1	0

This mask gives an isotropic result for rotations in increments of 90°.

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a b c d

(a) Laplacian kernel

(b) Kernel used to implement an extension of this equation that includes the diagonal terms.

(c) and (d) Two other Laplacian kernels.

Background features can be "recovered" while still preserving the sharpening effect of the Laplacian operator simply by adding the original & the Laplacian images.

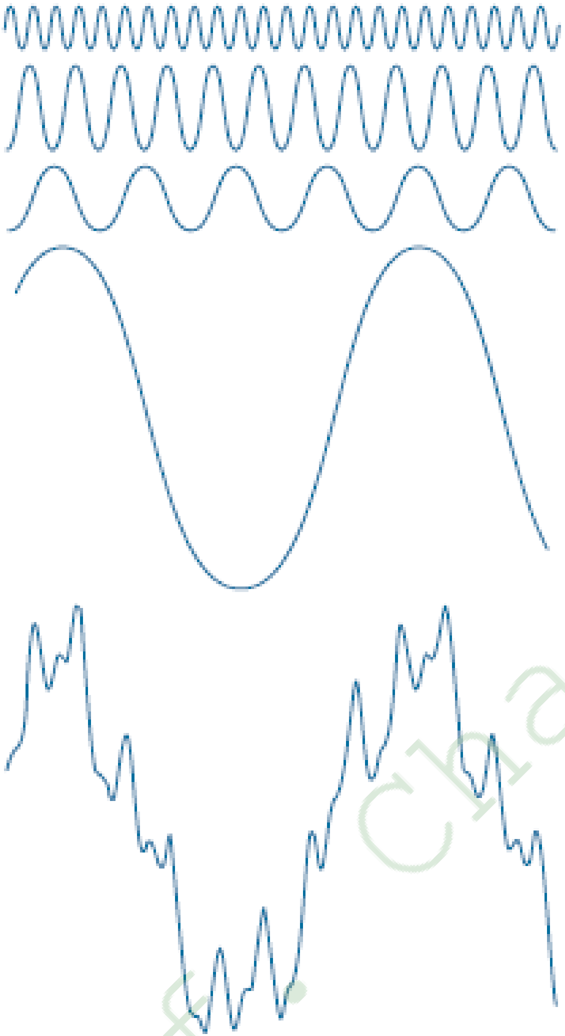
$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \textit{If the center coefficient is negative} \\ f(x,y)+\nabla^2 f(x,y) & \textit{If the center coefficient is positive} \end{cases}$$

Where f(x, y) is the original image

$\nabla^2 f(x,y)$ is Laplacian filtered image

g(x, y) is the sharpened image

Filtering in the Frequency Domain



The function at the bottom is the sum of the four functions above it.

Fourier Transform & The Frequency Domain

The One-Dimensional Fourier Transform & its Inverse

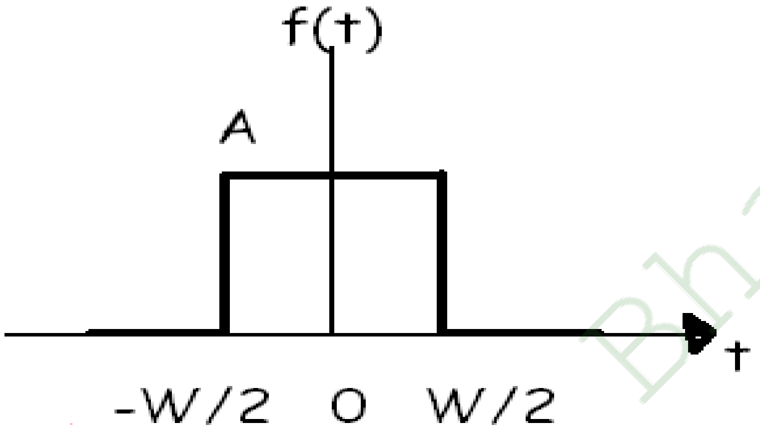
Let f(x) be a continuous function of x.
The Fourier transform of f(x) is:

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux}dx$$

$$= \int_{-\infty}^{\infty} f(x)[cos2\pi ux - jsin2\pi ux]dx$$

The inverse Fourier transform of f(x) is:

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux}du$$



Find its Fourier Transform.

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

$$F(u) = \int_{-\frac{W}{2}}^{\frac{W}{2}} A e^{-j2\pi ut} dt$$

$$F(u) = \int_{-\frac{W}{2}}^{\frac{W}{2}} A e^{-j2\pi ut} dt$$

$$F(u) = \frac{-A}{j2\pi u} [e^{-j2\pi ut}]_{\frac{-W}{2}}^{\frac{W}{2}}$$

$$F(u) = \frac{-A}{j2\pi u} [e^{-j\pi uW} - e^{j\pi uW}]$$

$$F(u) = \frac{A}{j2\pi u} [e^{j\pi uW} - e^{-j\pi uW}]$$

$$F(u) = \frac{A}{j2\pi u}\left[e^{j\pi uW} - e^{-j\pi uW}\right]$$

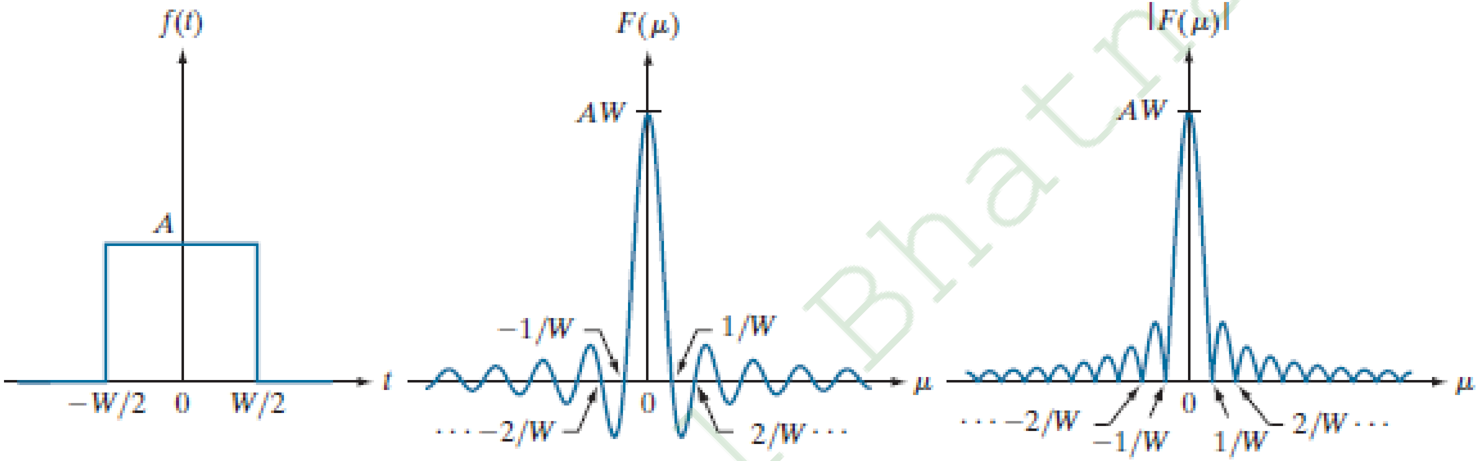
We know that

$$\sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}$$

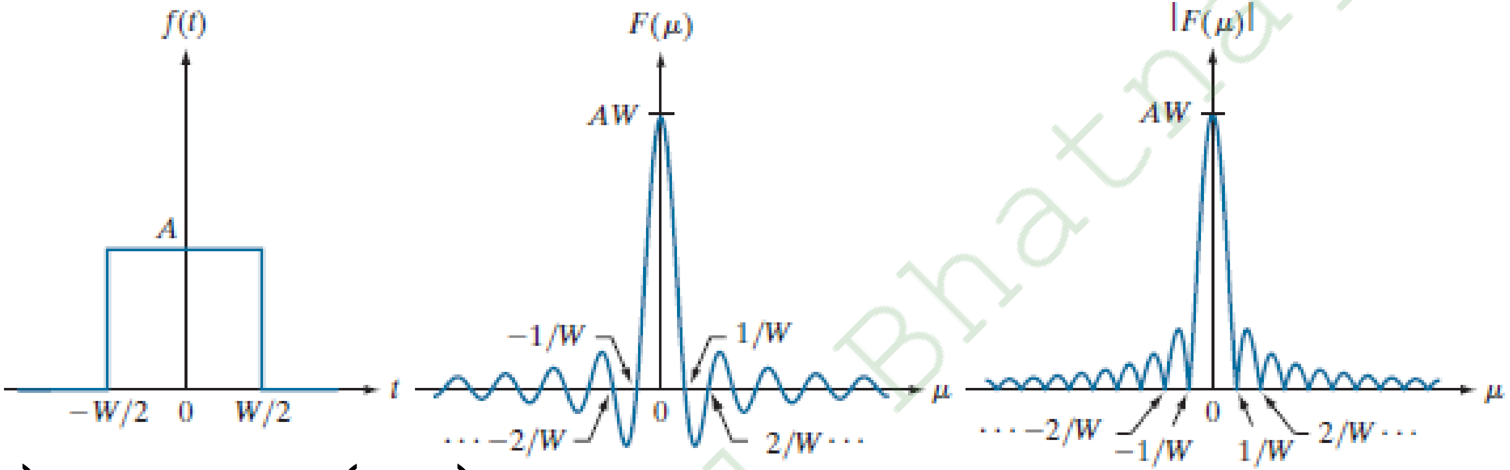
Therefore,

$$F(u) = \frac{A}{\pi u}\sin(\pi uW)$$

$$= AW\frac{\sin(\pi uW)}{\pi uW}$$

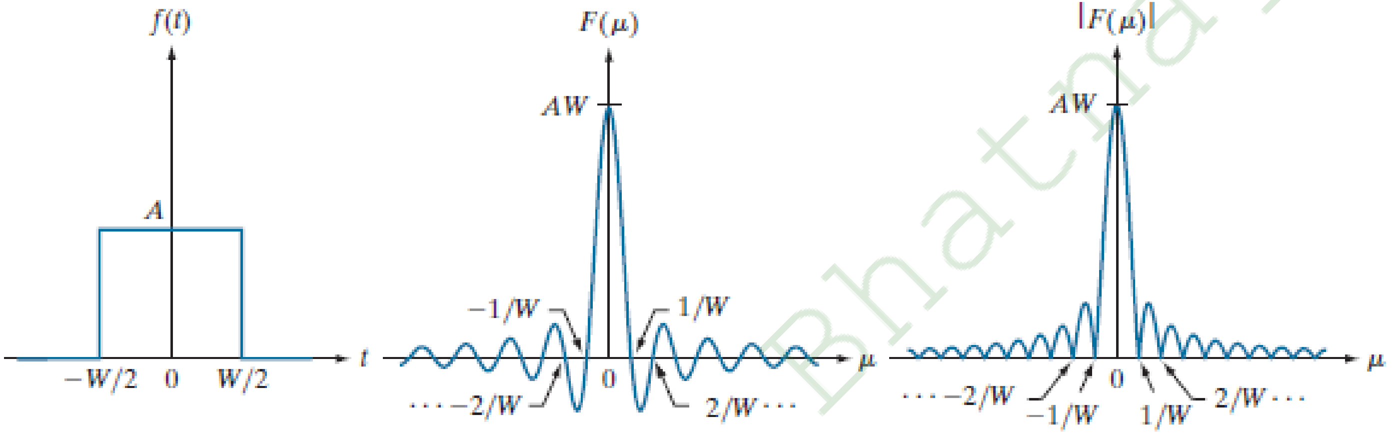


(a) A box function,
(b) its Fourier transform,
(c) its spectrum. All functions extend to infinity in both directions. Note the inverse relationship between the width, *W, of the function and the zeros of the transform*.



$$F(u) = AW \operatorname{sinc}(uW)$$

u	F(u)	Value of F(u)
0	AW sinc(0)	AW
1/W	AW sinc(1)	0
2/W	AW sinc(2)	0



- The Fourier transform contains complex terms
- Customary for display purposes to wo with the magnitude of the transform (a real quantity)

$$|F(\mu)| = AW\left|\frac{\sin(\pi\mu W)}{(\pi\mu W)}\right|$$

- This is called the *Fourier spectrum or the frequency spectrum:*

The One-Dimensional Discrete Fourier Transform (DFT) & its Inverse

The Fourier transform of a discrete function of one variable, f(x), x = 0, 1, ..., M-1, is given by the following equation

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-\frac{j2\pi ux}{M}} \qquad \text{for } u = 0, 1, \ldots, M-1$$

where $j = \sqrt{-1}$

Conversely, given F(u), f(x) can be obtained by means of the *inverse* Fourier transform, i.e. we can obtain the original function back using the inverse DFT

$$f(x) = \sum_{u=0}^{M-1} F(u)e^{j2\pi ux/M} \quad \text{for } x = 0, 1, \ldots, M-1$$

$$f(x) \Leftrightarrow F(u)$$

***denotes a* Fourier transform pair.**

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-\frac{j2\pi ux}{M}} \qquad \text{for } u = 0, 1, \ldots, M-1$$

To compute F(u):

1. Substitute u = 0 in the exponential term & then sum for *all* values of x.
2. Next substitute u = 1 in the exponential & repeat the summation over all values of x.
3. Repeat this process for all M values of u in order to obtain the complete Fourier transform.

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-\frac{j2\pi ux}{M}} \qquad \text{for } u = 0, 1, \ldots, M-1$$

- Each term of the Fourier transform is composed of the sum of all values of the function f(x)
- The domain (values of u) over which the values of F(u) range is called the Frequency domain because u determines the frequency components of the transform
- Each of the M terms of F(u) are known as Frequency Component
  of the transform

M = 4?

Do you remember Euler's Formula:

$$ej\theta = \cos\theta + j\sin\theta$$

$F(0) = 1$ $\quad$ $F(1) = -1/2$

$F(2) = 0$ $\quad$ $F(3) = -1/2$

From these values, now find the values of f(x).

Easier way to calculate DFT

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-\frac{j2\pi ux}{M}} \qquad \text{for } u = 0, 1, \dots, M-1$$

Twiddle Factor

A new factor defined as

$$W_M = e^{\frac{-j2\pi}{M}}$$

The DFT now reduces to

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) W_M^{xu}$$

To solve this equation, form a square matrix WM of size M x M

The equation then reduces to

$$\mathbf{F} = \mathbf{W} \,.\, \mathbf{f}$$

If M = 4, then WM will be a 4x4 matrix

$$
W_4 = \begin{array}{c|c}
u/x & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\
\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} &
\begin{pmatrix}
W_M^0 & W_M^0 & W_M^0 & W_M^0 \\
W_M^0 & W_M^1 & W_M^2 & W_M^3 \\
W_M^0 & W_M^2 & W_M^4 & W_M^6 \\
W_M^0 & W_M^3 & W_M^6 & W_M^9
\end{pmatrix}
\end{array}
$$

$$\boxed{W_M = e^{\frac{-j2\pi}{M}}}$$

$$W_M^1 = e^{\frac{-j2\pi.1}{4}} = \cos\left(-\frac{\pi}{2}\right) + j\sin\left(-\frac{\pi}{2}\right) = -j$$

$$W_M^0 = ??$$

$$W_M^0 = e^{\frac{-j2\pi.0}{4}} = 1$$

F(0) = 1 F(1) = -1/2

F(2) = 0 F(3) = -1/2

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

The Two-Dimensional Discrete Fourier Transform (DFT) & its Inverse

Two Dimensional Discrete Fourier Transform (DFT) Pair

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

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

Q) Find the DFT of the following image.

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

Hint:

First do row-wise transformation & then column-wise.

$$F(0,v) = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} 4 \\ -2 \\ 0 \\ -2 \end{bmatrix}$$

Likewise find the DFT of other rows.

$$\begin{bmatrix} 1 & \frac{-1}{2} & 0 & \frac{-1}{2} \\ 2 & \frac{-1}{2} & 0 & \frac{-1}{2} \\ 3 & \frac{-1}{2} & 0 & \frac{-1}{2} \\ 2 & \frac{-1}{2} & 0 & \frac{-1}{2} \end{bmatrix}$$

Now find the 1-D DFT along the columns of this intermediate image.

$$F(u,0) = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} 8 \\ -2 \\ 0 \\ -2 \end{bmatrix}$$

Likewise find the DFT of other columns.

The final DFT of the entire image is:

$$\begin{bmatrix} 2 & \frac{-1}{2} & 0 & \frac{-1}{2} \\ \frac{-1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{-1}{2} & 0 & 0 & 0 \end{bmatrix}$$

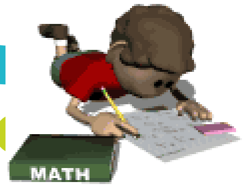
Filtering In Frequency Domain

Steps for Filtering in Frequency Domain

1. Multiply the i/p image by (-1)x+y to centre the transform

$$\mathscr{F}[f(x,y)(-1)^{x+y}] = F\left(u-\frac{M}{2}, v-\frac{N}{2}\right)$$

$\mathscr{F}$ denotes the Fourier transform of the argument



2. Compute F(u, v), the DFT of the image

3. Multiply F(u, v) by a *filter* function H(u, v)

4. Compute the inverse DFT

5. Obtain the real part of the result

6. Multiply the result by (-1)x+y

Fourier Transform of the o/p image

?

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

Filter

Fourier Transform of the i/p image f(x, y)

Multiplication of H & F On an element by element basis

Some Basic Filters

Filtering techniques in the frequency domain are based on modifying the Fourier transform to achieve a specific objective, and then computing the inverse DFT to get us back to the spatial domain

Frequency Domain Filtering Fundamentals

Given (a padded) digital image, f (x, y), of size P × Q pixels, the basic filtering equation has the form:

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

$\boxed{\text{IDFT}}$

$\boxed{\text{Filter}}$

Low frequencies in the transform are related to slowly varying intensity components in an image, such as the walls of a room or a cloudless sky in an outdoor scene.

High frequencies are caused by sharp transitions in intensity, such as edges and noise.

Lowpass Filter

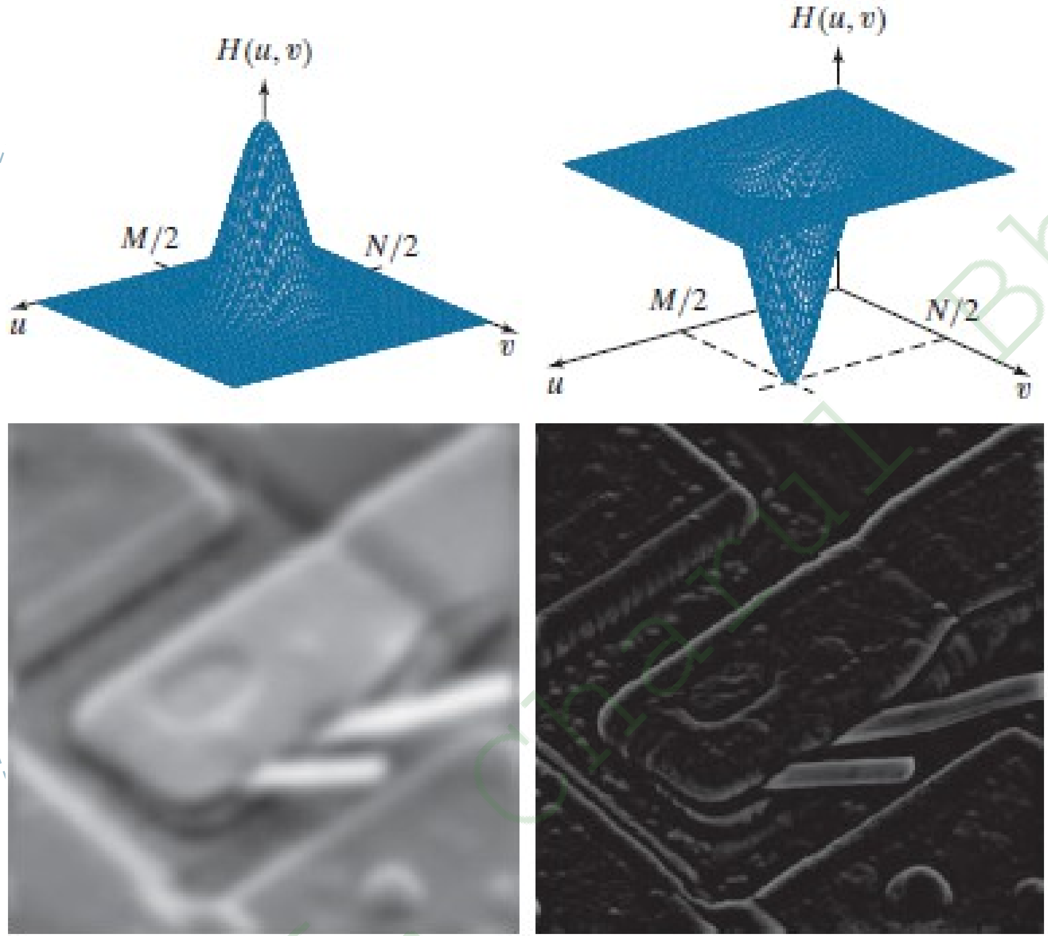
- Attenuates high frequencies
- Passes low frequencies

Lowpass filtered image have less sharp details than the original image as high frequencies have been attenuated

Highpass Filter

- Attenuates low frequencies
- Passes high frequencies

Highpass filtered images have less gray level variations in the smooth areas & emphasized transitional (e.g. edge) gray level details. Such an image appears sharper.



Top row: Frequency domain filter transfer functions of

(a) a lowpass filter,

(b) a highpass filter,

Bottom row: Corresponding filtered images

Smoothing Frequency Domain Filters

Smoothing (blurring) is achieved in the frequency domain by attenuating a specified range of high frequency components in the transform of a given image

Gaussian Lowpass Filters

The transfer function of a GLPF is given as:

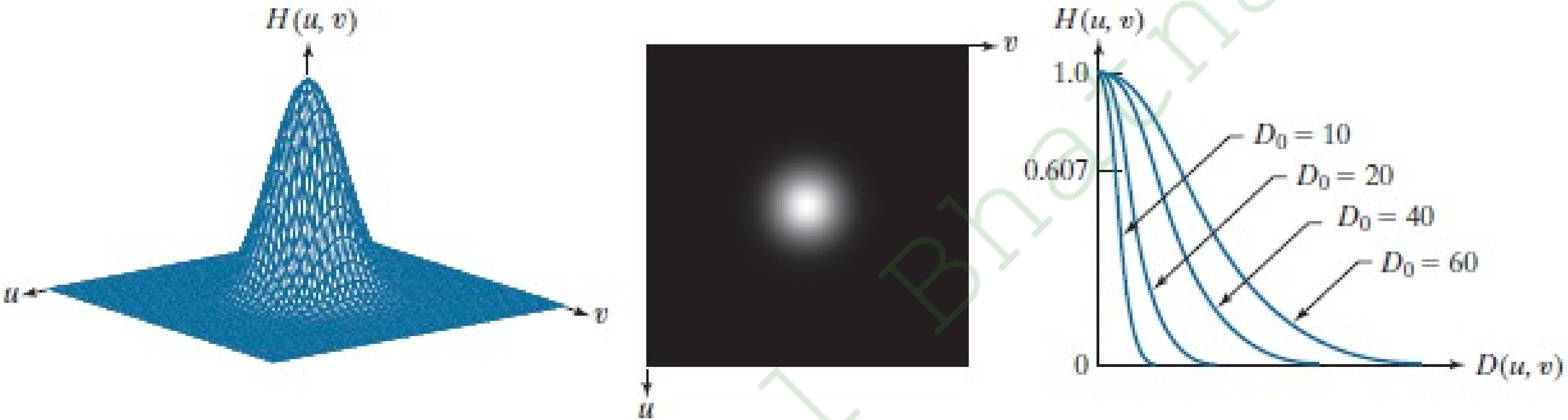
$$\boxed{H(u,v) = e^{-D^2(u,v)/2\sigma^2}}$$

σ is a measure of the spread of the Gaussian curve

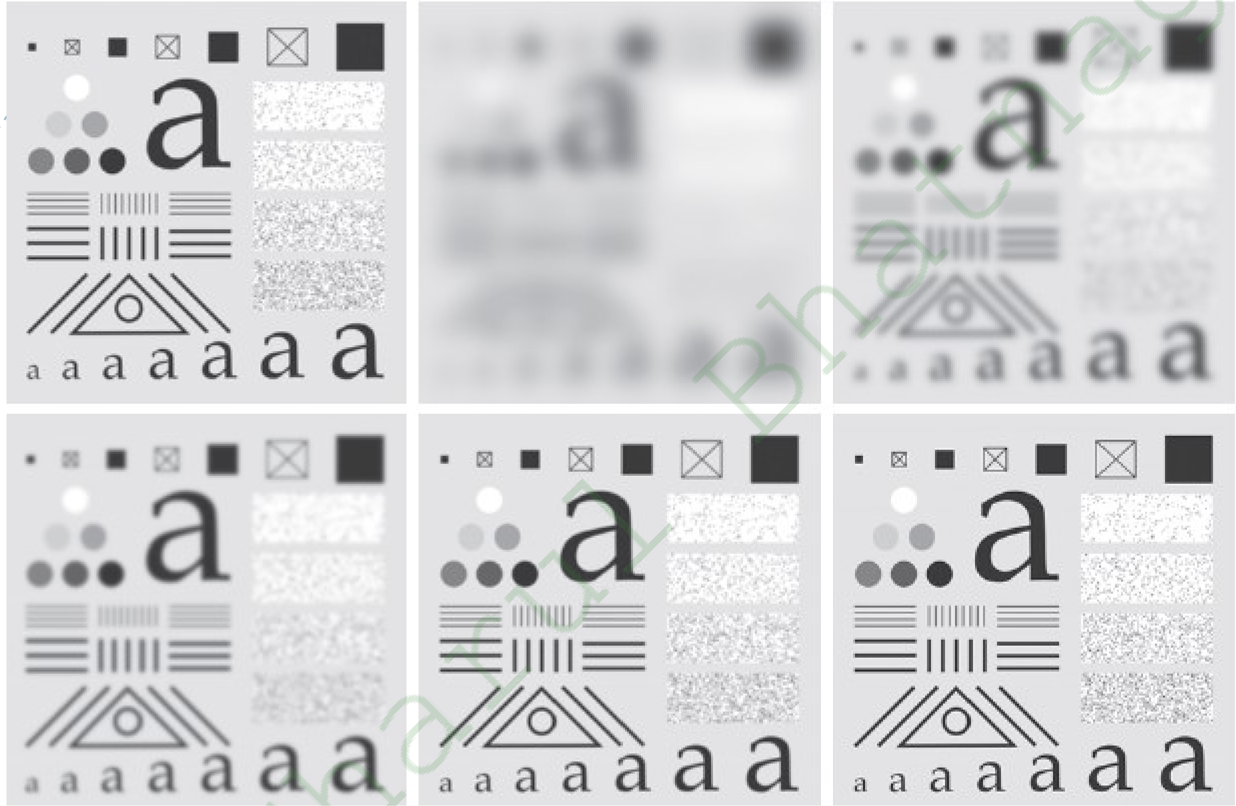
Inverse Fourier Transform of a GLPF is also Gaussian

Advantages:

- ❑ Since the transform & its inverse both are Gaussian, therefore they are real. This facilitates analysis because we do not have to be concerned with complex nos.
- ❑ Gaussian curves are intuitive & easy to manipulate.
- ❑ These functions behave reciprocally wrt one another. This helps considerably in developing a solid understanding of the properties of filtering in both spatial & frequency domains because they lend themselves to familiar analytical interpretation.



(a) Perspective plot of a GLPF transfer function.
(b) Function displayed as an image.
(c) Radial cross sections for various values of *Do* .



(a) Original image of size 688 × 688 pixels. (b)–(f) Results of filtering using GLPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460

Gaussian Highpass Filters

The transfer function of a GHPF is given as:

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

σ is a measure of the spread of the Gaussian curve