UNIT-I

DIGITAL IMAGE FUNDAMENTALS & IMAGE TRANSFORMS

What Is Digital Image Processing?

- The field of digital image processing refers to processing digital images by means of a digital computer.

What is a Digital Image ?

- An image may be defined as a two- dimensional function, f(x,y) where x and y are *spatial (plane) coordinates, and the amplitude of f at any pair of* coordinates (x, y) is called the *intensity or gray level of the image at that point.*

- When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a *digital image*

Picture elements, Image elements, pels, and pixels

- A digital image is composed of a finite number of elements, each of which has a particular location and value.
- These elements are referred to as *picture elements, image elements, pels, and pixels.*

- Pixel is the term most widely used to denote the elements of a digital image.

The Origins of Digital Image Processing

- One of the first applications of digital images was in the newspaper industry, when pictures were first sent by submarine cable between London and New York.

- Specialized printing equipment coded pictures for cable transmission and then reconstructed them at the receiving end.

- Figure was transmitted in this way and reproduced on a telegraph printer fitted with typefaces simulating a halftone pattern.

- The initial problems in improving the visual quality of these early digital pictures were related to the selection of printing procedures and the distribution of intensity levels

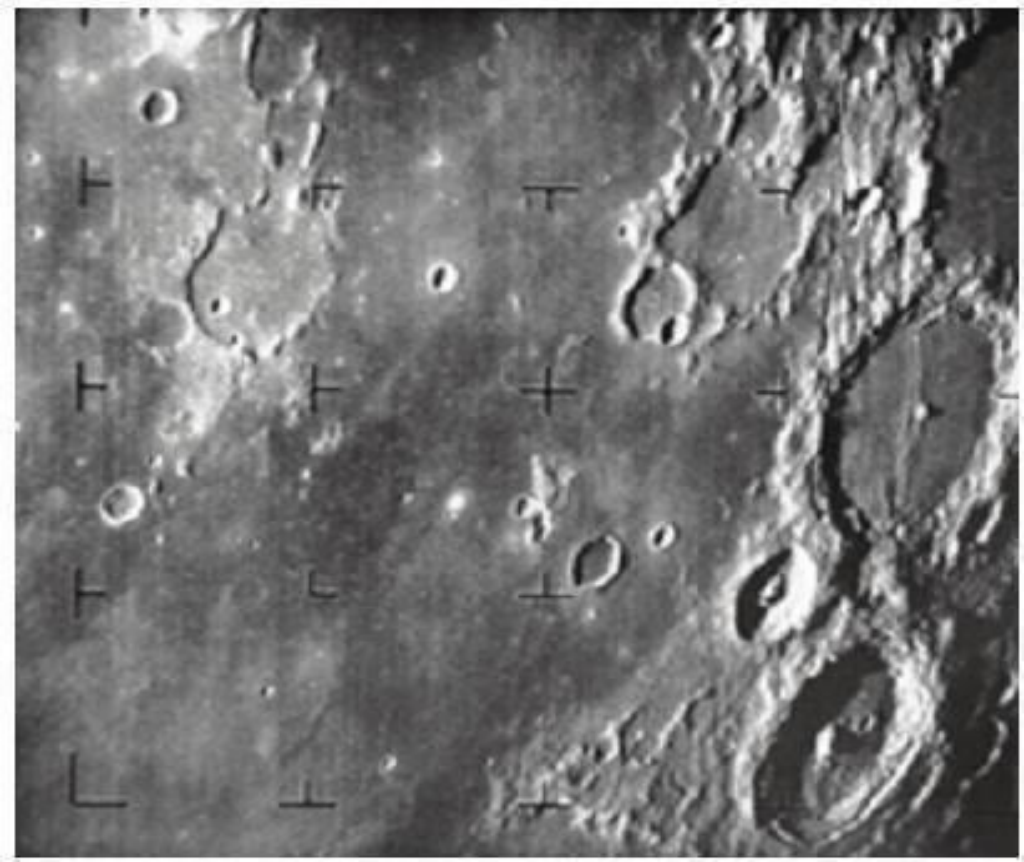
- The printing technique based on photographic reproduction made from tapes perforated at the telegraph receiving terminal from 1921.

- Figure shows an image obtained using this method.
- The improvements are tonal quality and in resolution.

- The early Bartlane systems were capable of coding images in five distinct levels of gray.
- This capability was increased to 15 levels in 1929.

- Figure is typical of the type of images thatcould be obtained using the 15-tone equipment.

- Figure shows the first image of the moon taken by *Ranger*



Applications of DIP

- The field of image processing has applications in medicine and the space program.

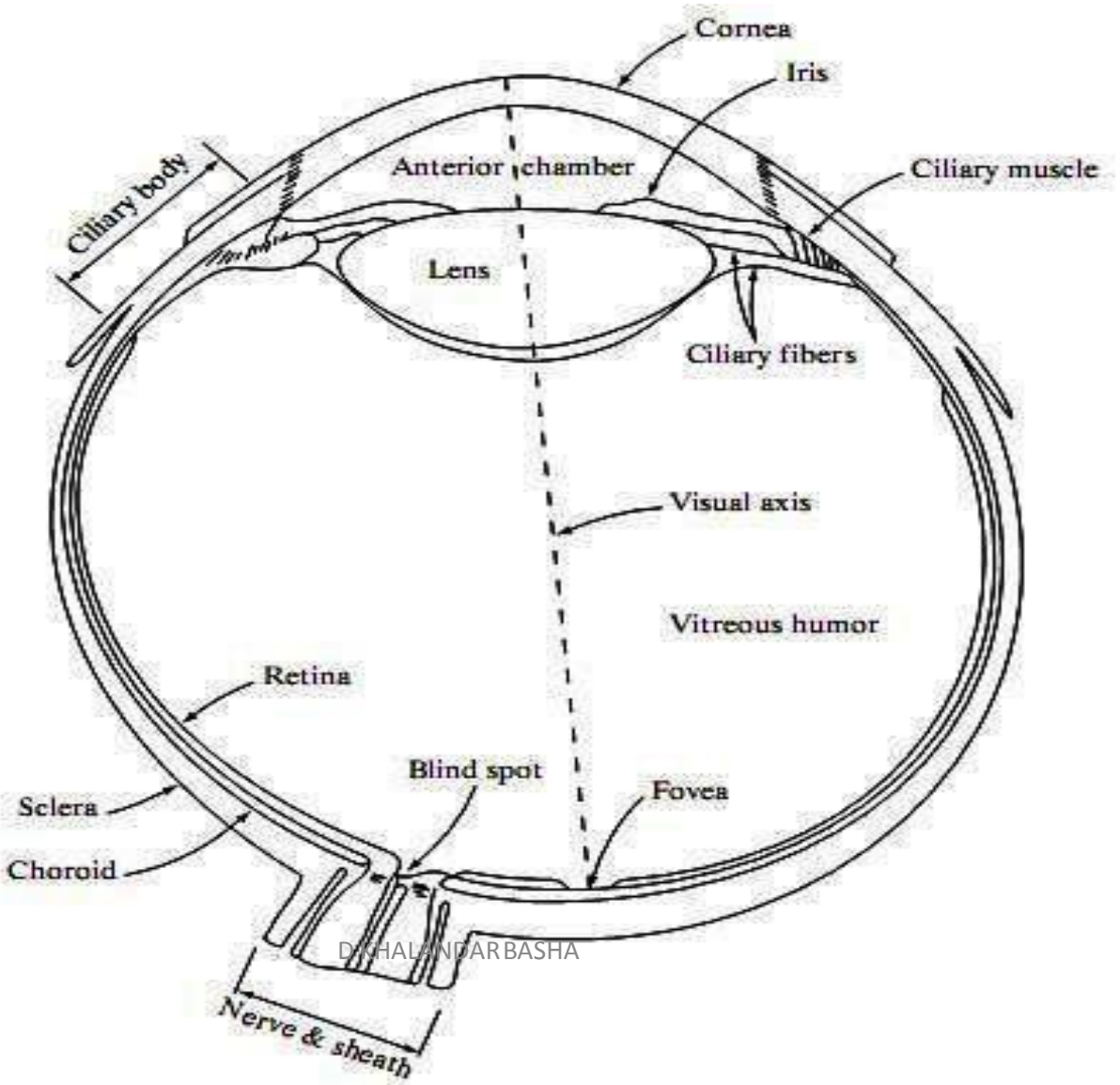
- Computer procedures are used to enhance the contrast or code the intensity levels into color for easier interpretation of X-rays and other images used in industry, medicine, and the biological sciences.

- Geographers use the same or similar techniques to study pollution patterns from aerial and satellite imagery

- **Image enhancement and restoration procedures are used to process degraded images of unrecoverable objects or**

- **Experimental results too expensive to duplicate.**

Structure of the Human Eye



- The eye is nearly a sphere, with an average diameter of approximately 20mm.

- Three membranes enclose the eye:

- The cornea and sclera outer cover the choroid the retina.

Cornea

- The cornea is a tough, transparent tissue that covers the anterior surface of the eye.

- Continuous with the cornea,the sclera is an opaque membrane that encloses the remainder of the optic globe.

Choroid

- The choroid lies directly below the sclera.
- This membrane contains a net- work of blood vessels that serve as the major source of nutrition to the eye.
- The choroid coat is heavily pigmented and hence helps to reduce the amount of extraneous light entering the eye and the backscatter within the optical globe.

- At its anterior extreme, the choroid is divided into the ciliary body and the iris diaphragm.
- The latter contracts or expands to control the amount of light that enters the eye
- The front of the iris contains the visible pigment of the eye, whereas the back contains a black pigment.

- The lens is made up of concentric layers of fibrous cells and is suspended by fibers that attach to the ciliary body.

- It contains 60 to 70% water, about 6% fat, and more protein than any other tissue in the eye.

Retina

- The innermost membrane of the eye is the retina, which lines the Inside of the ‖all's entire posterior portion.

- When the eye is properly focused, light from an object outside the eye is imaged on the retina.

- Pattern vision is afforded by the distribution of discrete light receptors over the surface of the retina.

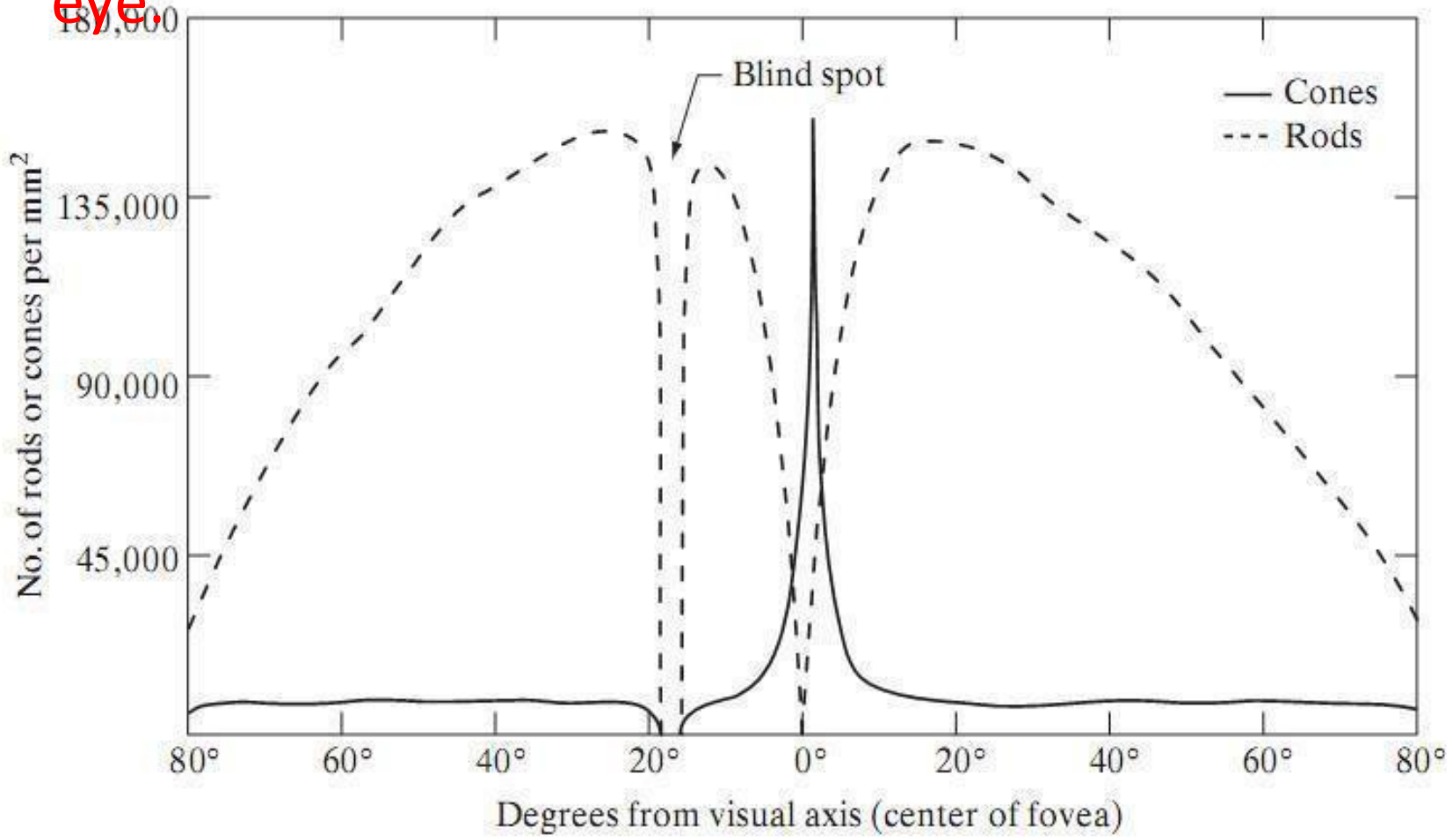
- There are two classes of receptors: cones and rods.
- The cones in each eye number between 6 and 7 million.
- They are located primarily in the central portion of the retina, called the fovea, and are highly sensitive to color.

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

- Cone vision is called photopic or bright-light vision.

- The number of rods is much larger: Some 75 to 150 million are distributed over the retinal surface.

- Figure shows the density of rods and cones for a cross section of the right eye passing through the region of emergence of the optic nerve from the eye.



- The absence of receptors in this area results in the so-called blind spot.

- Fig. shows that cones are most dense in the center of the retina (in the center area of the fovea)

Image Formation in the Eye

- The principal difference between the lens of the eye and an ordinary optical lens is that the former is flexible.
- The shape of the lens is controlled by tension in the fibers of the ciliary body.
- To focus on distant objects, the controlling muscles cause the lens to be relatively flattened.
- Similarly, these muscles allow the lens to become thicker in order to focus on objects near the eye.

- The distance between the center of the lens and the retina called the *focal length varies from approximately 17 mm to about 14 mm, as the refractive power of the lens increases from its minimum to its maximum.*

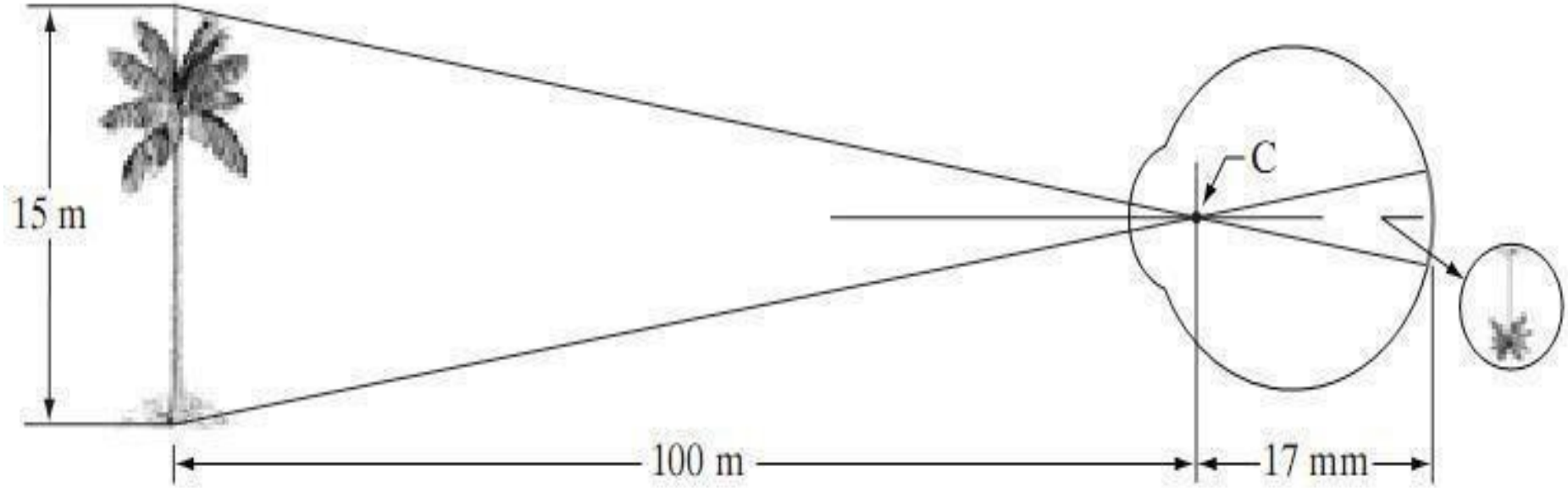
- *When the eye* focuses on an object farther away the lens exhibits its lowest refractive power.

- When the eye focuses on a nearby object, the lens is most strongly refractive.

- For example, the observer is looking at a tree 15 m high at a distance of 100 m.

- If h is the height in mm of that object in the retinal image,the geometry of Fig. yields

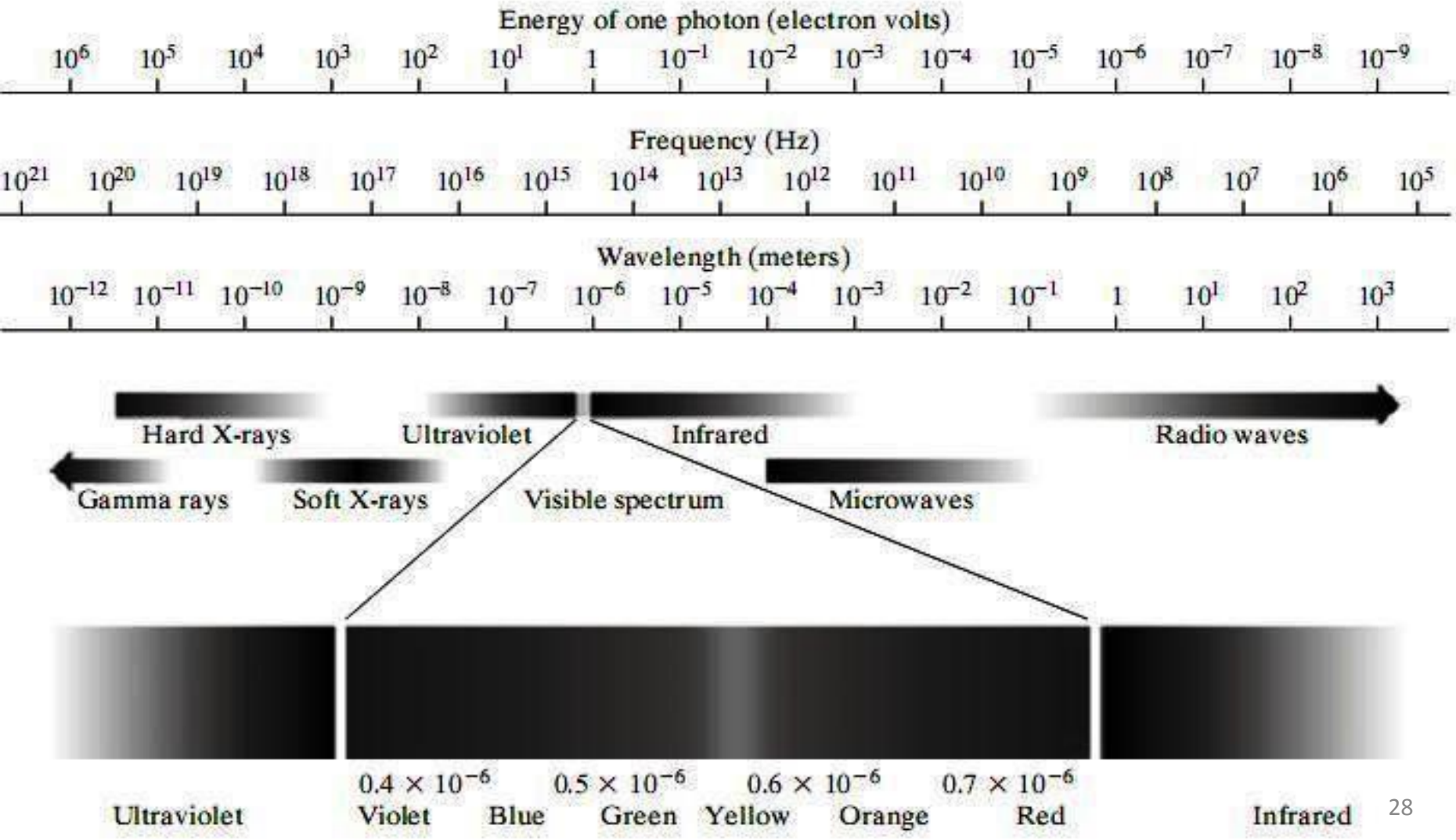
15/100 = h/17 or h=2.55mm.



Light and the Electromagnetic Spectrum

- Sir Isaac Newton discovered that when a beam of sunlight is passed through a glass prism,
- The emerging beam of light is not white but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other.

The electromagnetic spectrum



- The electromagnetic spectrum can be expressed in terms of wavelength, frequency, or energy.
- Wavelength (l)and frequency (n)are related by the expression
- where c is the speed of light ($2.998*10^8$ m s)
- The energy of the electromagnetic spectrum is given by the expression E = hv
- where *h is Plank"s constant*

A Simple Image Formation Model

- Images by two-dimensional functions of the form f(x, y).
- The value or amplitude of f at spatial coordinates (x, y) gives the intensity (brightness) of the image at that point.
- As light is a form of energy, f(x,y) must be non zero and finite.

- The function f(x, y) may be characterized by two components:

  (1)the amount of source illumination incident on the scene being viewed

  (2)the amount of illumination reflected by the objects in the scene.

- These are called the *illumination and reflectance components and are denoted by i(x, y) and r(x, y), respectively.*

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

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

$r(x, y) = 0$ --- total absorption

1 --- total reflection

- The intensity of a monochrome image f at any coordinates (x, y) the *gray level (l) of the image at that point.*

That is, $l = f(x_0, y_0)$

L lies in the range $L_{min} \leq \ell \leq L_{max}$

In practice, $L_{min} = i_{min} r_{min}$ and $L_{max} = i_{max} r_{max}$

GRAY SCALE

- The interval $[L_{min}, L_{max}]$ is called the*gray scale.*
- *Common practice is to shift this interval numerically to the interval [0, L-1],*
- *where L = 0 is considered black and*

L = L-1 is considered white on the gray scale.

All intermediate values are shades of gray varying from black to white.

Basic Relationships Between Pixels

- 1. **Neighbors of a Pixel :-**

A pixel p at coordinates (x, y) has four *horizontal and vertical neighbors whose coordinates are given by* (x+1, y), (x-1, y), (x, y+1), (x, y-1)

- This set of pixels, called the 4-*neighbors of p, is denoted by* N_4(p).

- Each pixel is a unit distance from (x, y), and some of the neighbors of p lie outside the digital image if (x, y) is on the border of the image.

$N_D(p)$ and $N_8(p)$

- The four *diagonal neighbors of p have coordinates*
  (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)
  and are denoted by $N_D(p)$.

- These points, together with the 4-neighbors, arecalled the 8-*neighbors of p, denoted by* $N_8(p)$.

- If some of the points in $N_D(p)$ and $N_8(p)$ fall outside the image if (x, y) is on the border of theimage.

Adjacency, Connectivity, Regions, and Boundaries

- To establish whether two pixels are connected, it must be determined if they are neighbors and
- if their gray levels satisfy a specified criterion of similarity (say, if their gray levels are equal).

- For instance, in a binary image with values 0 and 1, two pixels may be 4-neighbors,

- but they are said to be connected only if they have the same value

- Let *V be the set of gray-level values used todefine connectivity. In a binary image, V={1} for the connectivity of pixels with value 1.*

- *In a grayscale image, for connectivity of pixels with a range of intensity values of say 32, 64 V typically contains more elements.*

- *For example,*

- *In the adjacency of pixels with a range of possible gray-level values 0 to 255,*
- *set V could be any subset of these 256 values.We consider three types of adjacency:*

- **We consider three types of adjacency:**

(a) 4-*adjacency.*

Two pixels p and q with values from V are 4-adjacent if q is in the set N_4(p).

(b) 8-*adjacency.*

***Two pixels p and q with values from V are 8-adjacent if q is* in the set N_8(p).**

(c) m-adjacency (mixed adjacency).

(d) Two pixels p and q with values from V are m-adjacent if

- **(i) q is in N_4(p)**, *or*
- ***(ii) q is in N_D(p) and theset whose values are from V.***

$N_4(p) \cap N_4(q)$

- A *path from pixel p with coordinates (x, y) to pixel q with coordinates (s, t) is a sequence of distinct pixels with coordinates*

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

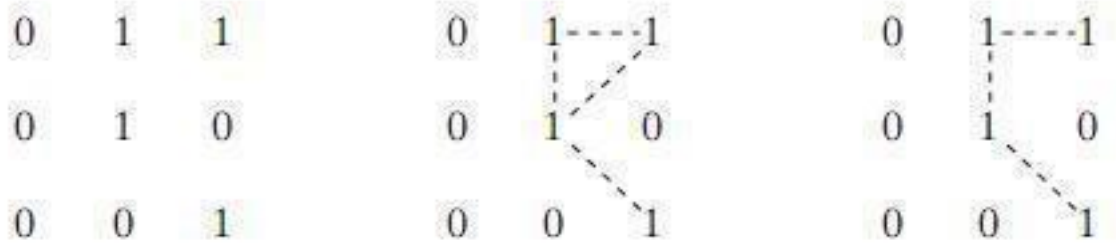
- where $(x_0, y_0) = (x, y)$ and $(x_n, y_n) = (s, t)$, (x_i, y_i) and (x_{i-1}, y_{i-1}) pixels and are adjacent for б ч і ч Ŷ. IŶ this Đase, Ŷ is the *length of the path.*

- *If* $(x_0, y_0) = (x_n, y_n)$ the path is a *closed path*.

.

- Two pixels p and 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 a connected component of S.*



a b c

FIGURE (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) *m*-adjacency.

Relations, equivalence

- A binary relation R on a set A is a set of pairs of elements from A. If the pair (a, b) is in R, the notation used is aRb (ie a is related to b)

- Ex:- the set of points A = { p1,p2,p3,p4} arranged as

P1 p2

P3

p4

- In this case R is set of pairs of points from A that are 4-connected that is R = {(p1,p2), (p2,p1), (p1,p3), (p3,p1)} .

thus p1 is related to p2 and p1 is related to p3 and vice versa but p4 is not related to any other point under the relation .

Reflective - Symmetric - Transitive

- Reflective

if for each a in A, aRb

- Symmetric

if for each a and b in A, aRb implies bRa

- Transitive

if for a, b and c in A, aRb and bRc implies aRc

A relation satisfying the three properties is called an equivalence relation.

Distance Measures

- For pixels p, q,and z,with coordinates (x, y), (s, t),and (u, v) respectively, *D is a distance function or metric if*

  (a) D(p, q)D;p, if p = q),

  (b) D(p, q) = D(q, p)

The *Euclidean distance between p and q is defined as*

$$D_e(p,q) = \left[(x-s)^2 + (y-t)^2\right]^{\frac{1}{2}}.$$

- The *D_4 distance (also called city-block distance) between p and q is defined as*

  $D_4(p, q) = |x - s| + |y - t|$

  •For example, the pixels with D_4 distance ч † from (x, y) (the center point) form the following contours of constant distance:

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

- The pixels with D_4=1 are the 4-neighbors of (x, y).

- The *D_8 distance (also called chess board distance)between p and q is defined as*

D_8 (p, q) = max (| x - s | , | y – t |)

- *For example, the pixels with D_8 distance ≤ 2 frow̃ (x, y) (the center point) form the following contours of constant distance:*

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

- The pixels with D_8=1 are the 8-neighbors of (x, y).

- The D_m distance between two points is defined as the shortest m-path between the points.

- In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.

- For instance, consider the following arrangement of pixels and assume that p, p2 and p4 have value 1 and that p1 and p3 can have a value of 0 or 1:

	p3	p4
P1	p2	
p		

- If only connectivity of pixels valued 1 is allowed, and p1 and p3 are 0 then the m distance between p and p4 is 2.

- If either p1 or p3 is 1, the distance is 3

- If both pi and p2 are 1, the distance is 4

DIGITAL IMAGE PROCESSING

UNIT 2: IMAGE ENHANCEMENT

Process an image so that the result will be more suitable than the original image for a application. specific

Highlighting interesting detail in images Removing noise from images

Making images more visually appealing

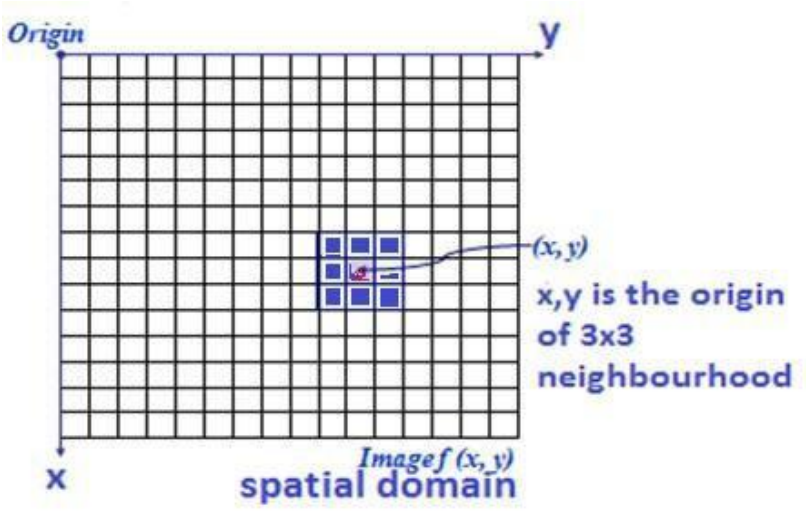
So, a technique for enhancement of x-ray image may not be the best for enhancement of microscopic images.

These spatial domain processes are expressed by:

$$G(x,y) = T(f(x,y))$$ depends only on the value of f at (x,y)

f(x,y) is the input image, G (x,y) is the output image

T is called a gray-level or intensity transformation operator which can apply to single image or binned images.



Window origin is moved from image origin along the 1st row and then second row etc.

At each location, the image pixel value is replaced by the value obtained after applying T operation on the window at the origin.

the neighborhood size may be different. We can have a neighborhood size of 5 by 5, 7 by 7 and so on depending upon the type of the image and the type of operation that we want to have.

Spatial domain techniques

Point Processing: Contrast stretching Thresholding

Intensity transformations / gray level transformations

> Image Negatives
> Log Transformations
> Power Law Transformations

Piecewise-Linear Transformation Functions Contrast stretching
Gray-level slicing Bit-plane slicing

Spatial filters

Smoothening filters Low pass filters Median filters
Sharpening filters High boost filters
Derivative filters

Suppose we have a digital image which can be represented by a two dimensional random field $f(x,y)$.

An image processing operator in the spatial domain may be expressed as a mathematical function $T[\cdot]$ applied to the input image $f(x,y)$ to produce a new image $g(x,y)=T[f(x,y)]$ as follows.

The operator T applied on $f(x,y)$ may be defined over some neighborhood of (x,y)

(i) A single pixel (x,y). In this case T is a gray level transformation (or mapping) function.

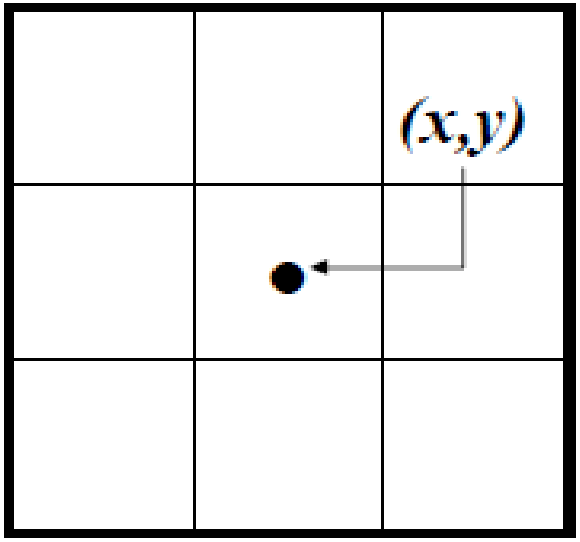
(ii) Some neighborhood of (x,y).

(iii) T may operate to a set of input images instead of a single image.

the neighborhood of a point (x, y) is usually a square sub image which is centered at point (x, y).

Mask/Filter

Neighborhood of a point (x,y) can be defined by using a square/rectangular (common used) or circular subimage area centered at (x,y)
The center of the subimage is moved from pixel to pixel starting at the top of the corner



Spatial Processing :
intensity transformation -> works on single pi
contrast manipulation

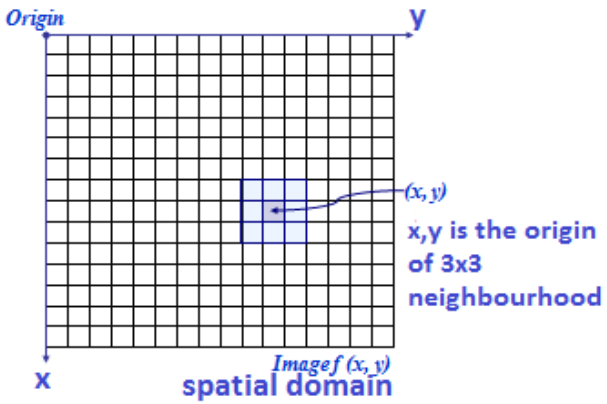


image thresholding

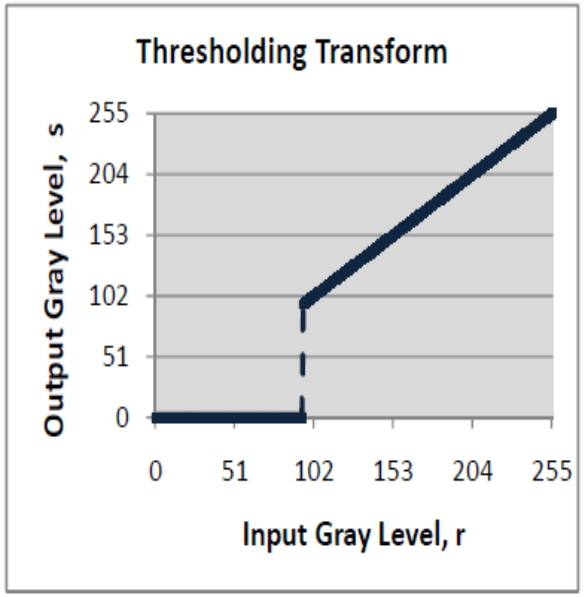
spatial filtering → Image sharpening (working on neighborhood of every pixel) or Neighborhood Processing:

Thresholding (piece wise linear transformation)

Produce a two-level (binary) image

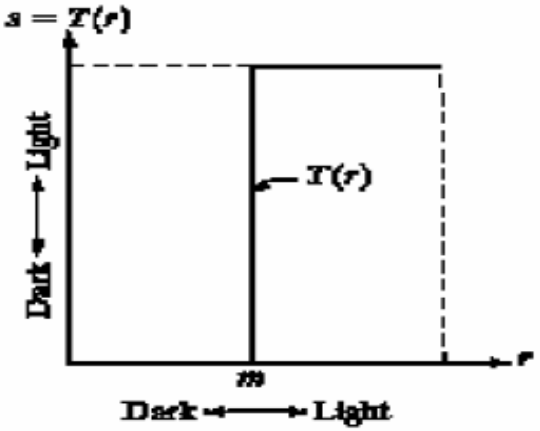
For any $0 < t < 255$ the threshold transform Thr_t can be defined as:

$$s = Thr_t(r) = \begin{cases} 0 & if\ r < t \\ r & otherwise \end{cases}$$

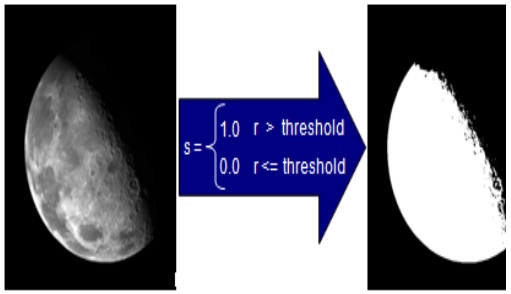


Thresholding has another form used to generate binary images from the gray-scale images, i.e.:

$$s = Thr_t(r) = \begin{cases} 0 & if\ r < t \\ 255 & otherwise \end{cases}$$



Thresholding transformations are particularl useful for segmentation in which we want to isolate an object of interest from a background



UNIT-II

IMAGE ENHANCEMENT (SPATIAL & FREQUENCY DOMAIN)

Spatial domain: Image Enhancement

Three basic type of functions are used for image enhancement. image enhancement point processing techniques:

Linear (Negative image and Identity transformations) Logarithmic transformation (log and inverse log transformations) Power law transforms (nth power and nth root transformations) Grey level slicing

Bit plane slicing

We are dealing now with image processing methods that are based only on the intensity of single pixels.

Intensity transformations (Gray level transformations)

Linear function Negative and identity Transformations

Logarithm function

Log and inverse-log transformation Power-law function

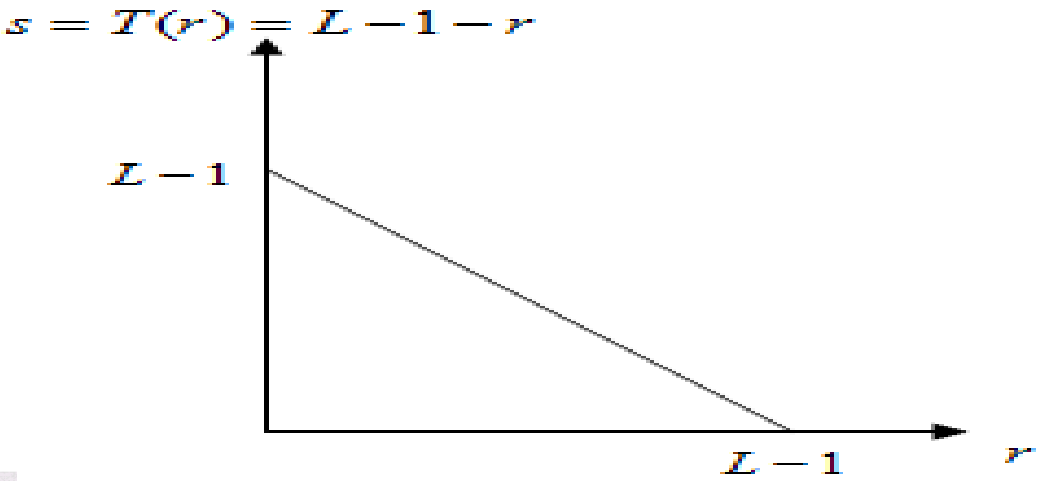
nth power and nth root transformations

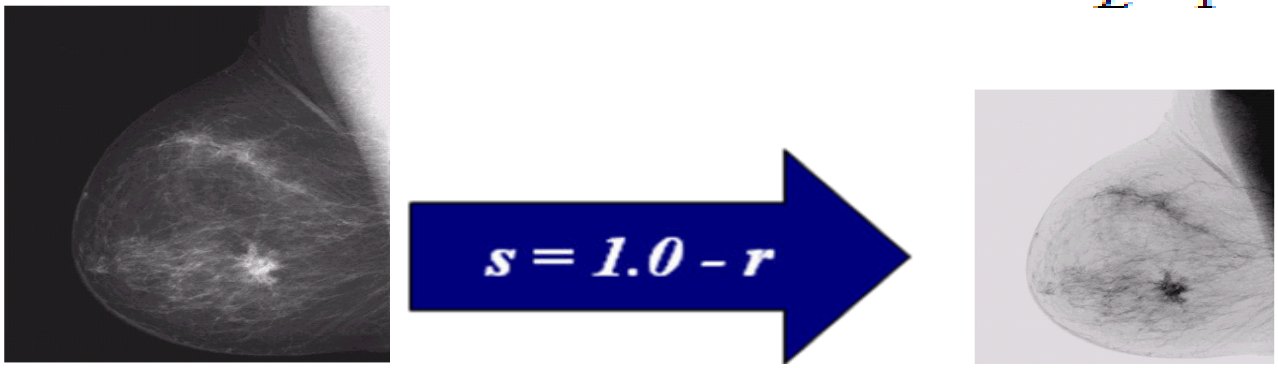
Image Negatives

Here, we consider that the digital image that we are considering that will have capital L number of intensity levels represented from 0 to capital L minus 1 in steps of 1.

The negative of a digital image is obtained by the transformation function

$$s=T(r)=L-1-r$$

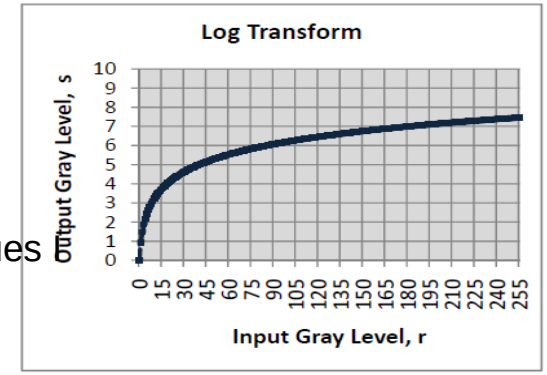




Logarithmic Transformations

The general form of the log transformation is s = c * log (1 + r)

C is a constant and r is assumed to be ≥ 0



The log transformation maps a narrow range of low input grey level values into a wider range of output values. The inverse log transformation performs the opposite transformation s = log(1 + r)

We usually set c to 1. Grey levels must be in the range [0.0, 1.0]

Identity Function

Output intensities are identical to input intensities.

Is included in the graph only for completeness

Power Law Transformations Why power laws are popular?

A cathode ray tube (CRT), for example, converts a video signal to light in a way. The light intensity is proportional to a power (γ) of the source voltage V_S For a computer CRT, γ is about 2.2

nonlinear

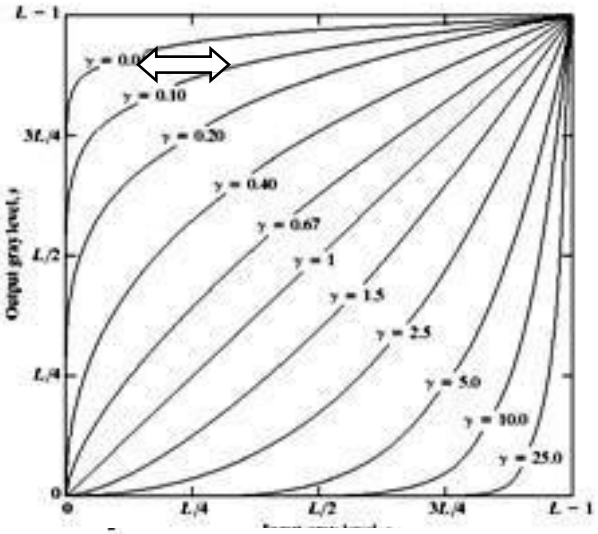
Viewing images properly on monitors requires γ-correction

Power law transformations have the following form s = c * r^{γ} c and γ are positive constants $s = r^{\gamma}$

We usually set c to 1. Grey levels must be in the range [0.0, 1.0]

Gamma correction is used for display improvements

Some times it is also written as $s = c\,(r+\epsilon)^{\gamma}$, and this offset is to provide a measurable output even when input values are zero

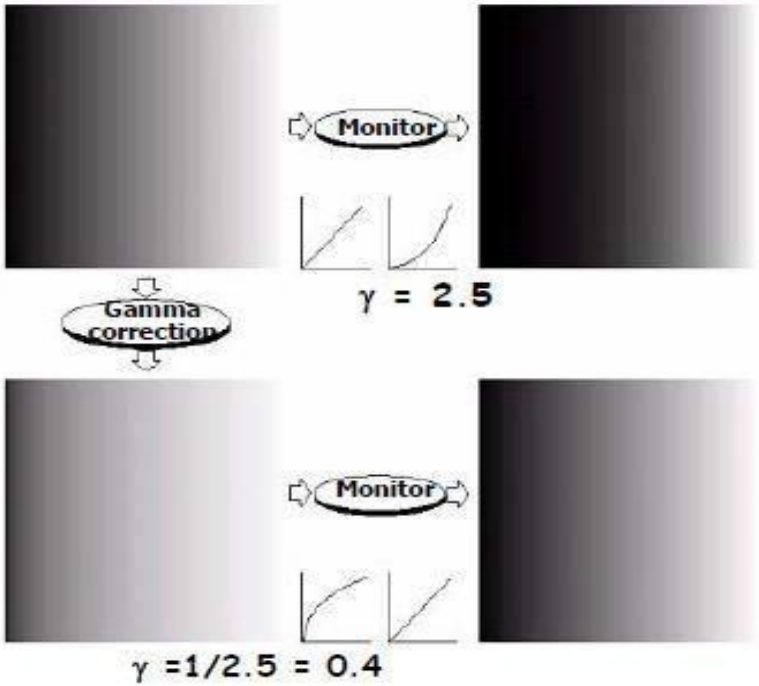


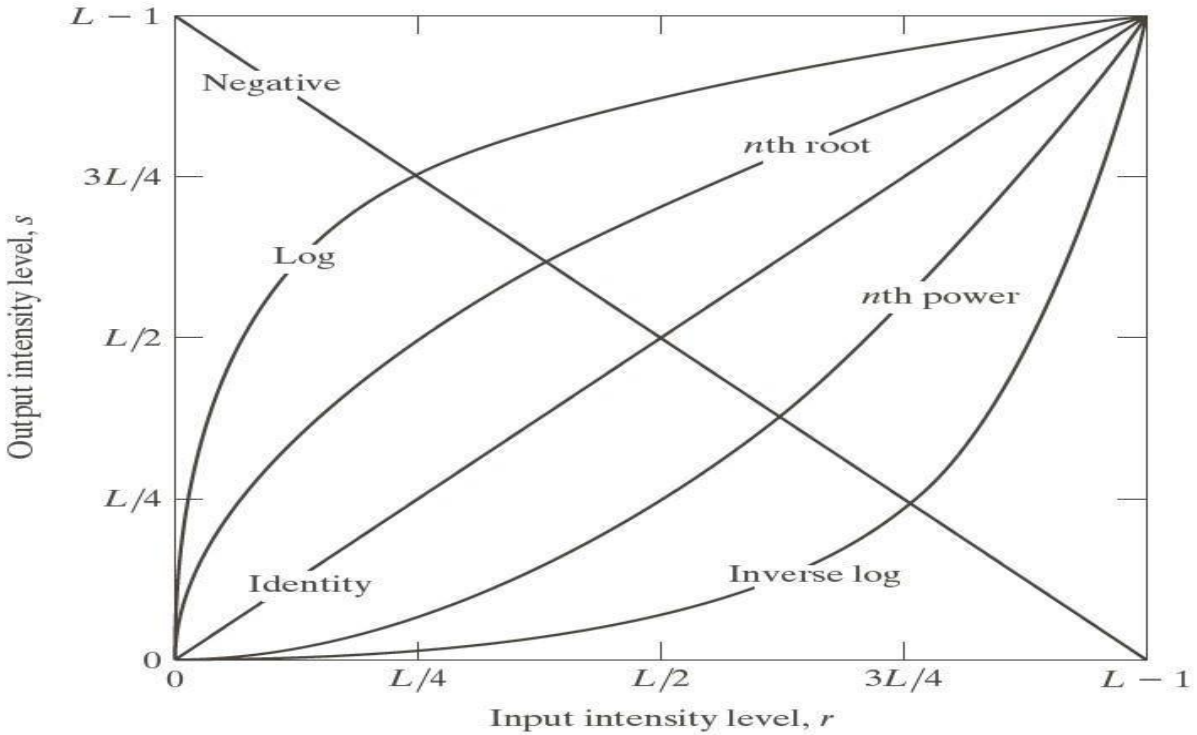
$s = cr^{\gamma}$

c and γ are positive constants

Power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.

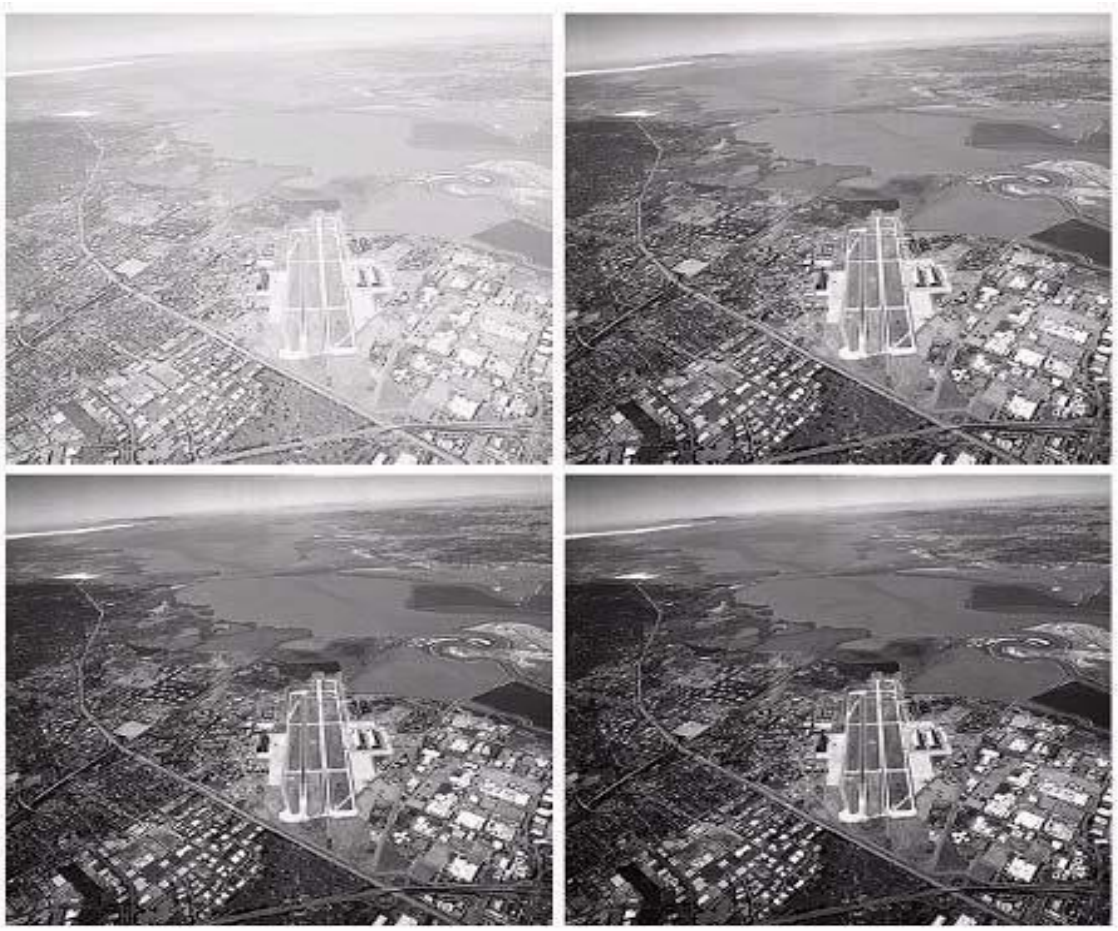
c = γ = 1 Identity function




$L-1$
$3L/4$
$L/2$
$L/4$
0
Output intensity level, s
Negative
nth root
Log
nth power
Identity
Inverse log
0
$L/4$
$L/2$
$3L/4$
$L-1$
Input intensity level, r

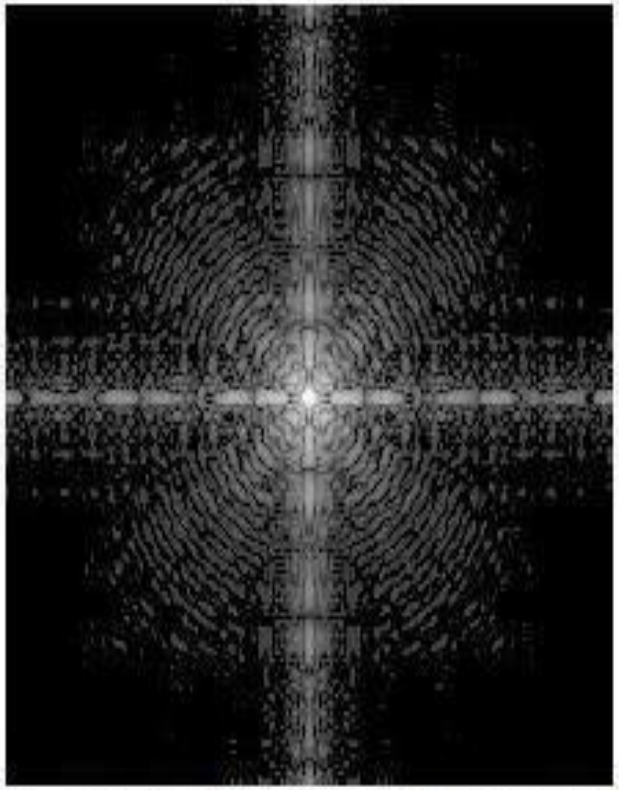
Effect of decreasing gamma

When the γ is reduced too much, the image begins to reduce contrast to the point where the image started to have very slight “wash-out” look, especially in the background



a) image has a washed-out
appearance, it needs a compression of gray levels
needs γ > 1
(b)result after power-law transformation with γ = 3.0
(suitable)
(c)transformation with γ = 4.0
(suitable)
(d)transformation with γ = 5.0
(high contrast, the image has areas that are too dark,
some detail is lost)

(a) Original image



(b) Result of Log transform with $c = 1$

Piecewise Linear Transformation Functions

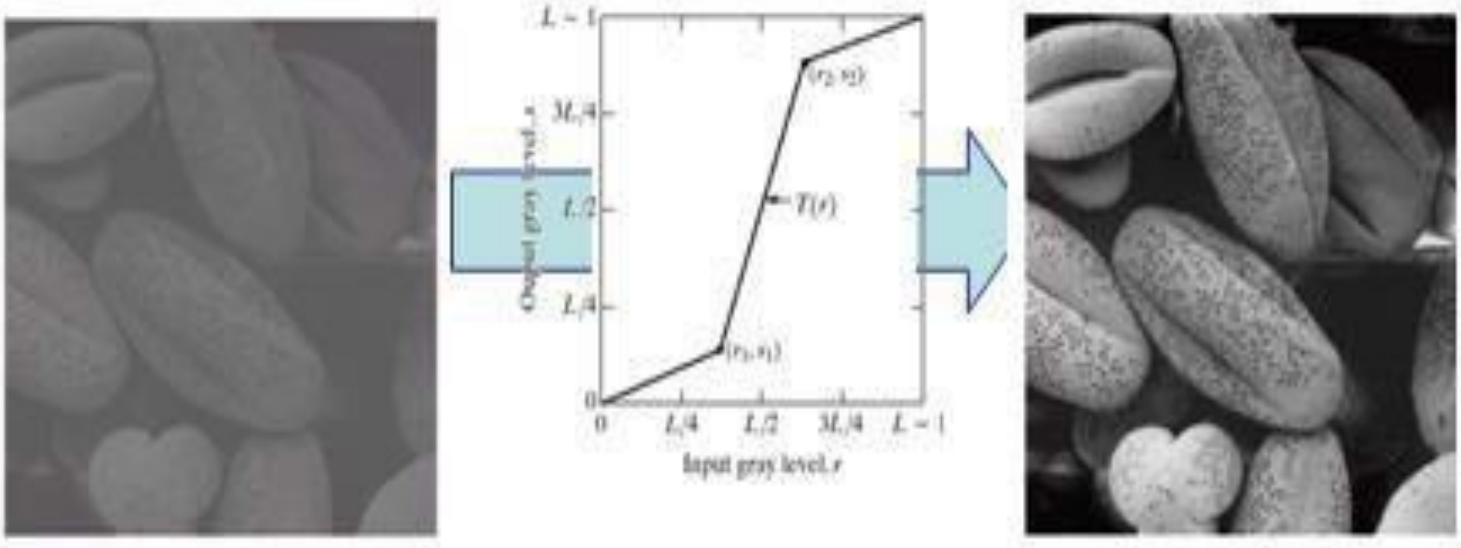
Piecewise functions can be arbitrarily complex

- A disadvantage is that their specification requires significant user input
- Example functions :
  - Contrast stretching
  - Intensity-level slicing
  - Bit-plane slicing

Contrast Stretching

Low contrast images occur often due to poor or non uniform lighting conditions, or due to nonlinearity, or small dynamic range of the imaging sensor.

Purpose of contrast stretching is to process such images so that the dynamic range of the image will be very high, so that different details in the objects present in the image will be clearly visible. Contrast stretching process expands dynamic range of intensity levels in an image so that it spans the full intensity range of the recording medium or display devices.



Control points (r1,s1) and (r2,s2) control the shape of the transform T(r)

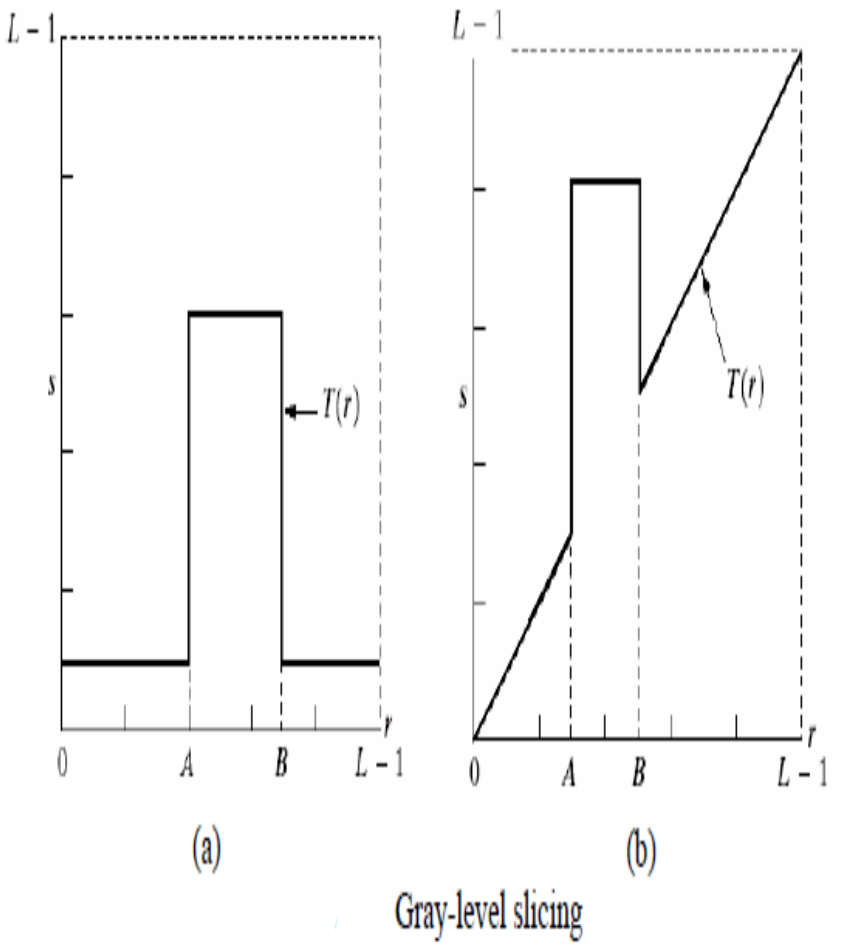
•if r1=s1 and r2=s2, the transformation is linear and produce no changes in intensity levels

•r1=r2, s1=0 and s2=L-1 yields a thresholding function that creates a binary image

•Intermediate values of (r1,s1) and (r2,s2) produce various degrees of spread in the intensity levels

In general, r1≤r2 and s1≤ s2 is assumed so that the junction is single valued and monotonically increasing.

If (r1,s1)=(rmin,0) and (r2,s2)=(rmax,L-1), where rmin and r max are minimum and maximum levels in the image. The transformation stretches the levels linearly from their original range to the full range (0,L-1)



Gray-level slicing

Two common approaches

– Set all pixel values within a range of interest to one value (white) and all others to another value (black)

•Produces a binary image

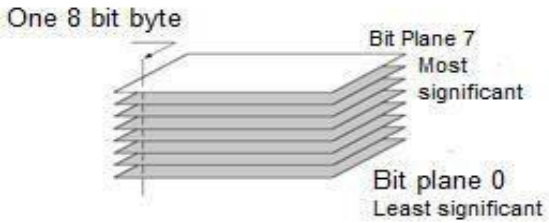
That means, Display high value for range of interest, else low value (‘discard background’)

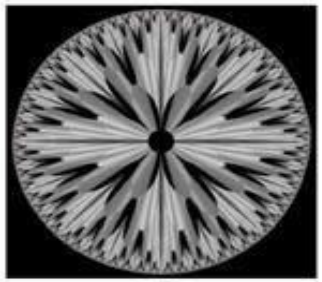
– Brighten (or darken) pixel values in a range of interest and leave all others Unchanged. That means , Display high value for range of interest, else original value (‘preserve background’)

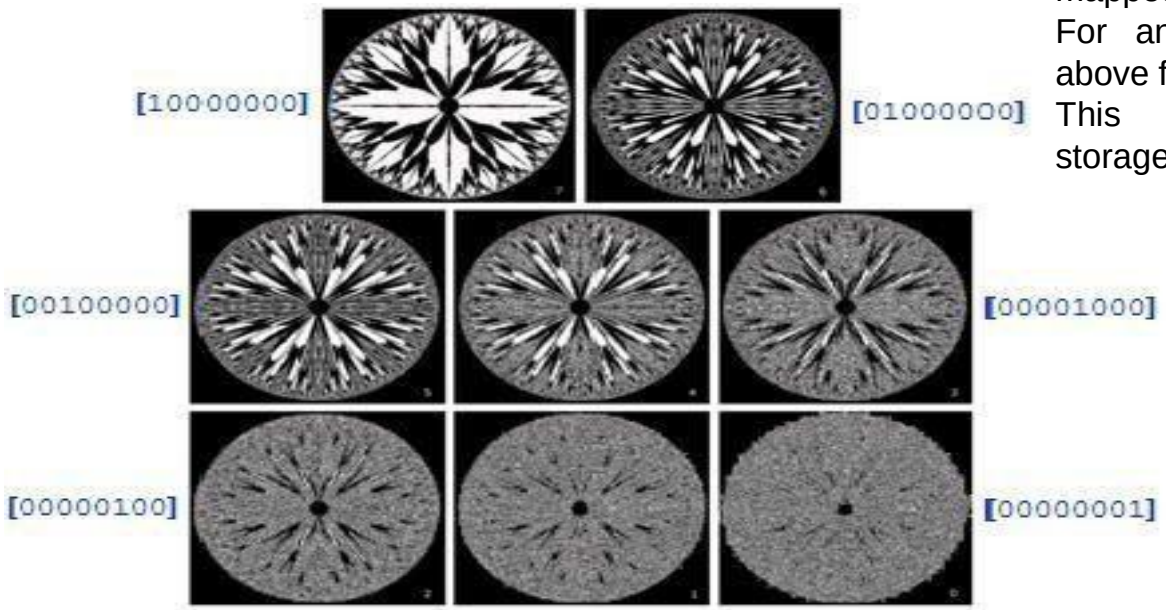
Bit Plane Slicing

Only by isolating particular bits of the pixel values in a image we can highlight interesting aspects of that image.

High order bits contain most of the significant visual information Lower bits contain subtle details







Reconstruction is obtained by:

$$I(i,j)=\sum_{n=0}^{N-1} 2^{n} I_n(i,j)$$

0 to 127 can be mapped as 0 128 to 256 can be mapped as 1

For an 8 bit image, the above forms a binary image. This occupies less storage space.

Image Dynamic Range, Brightness and Control

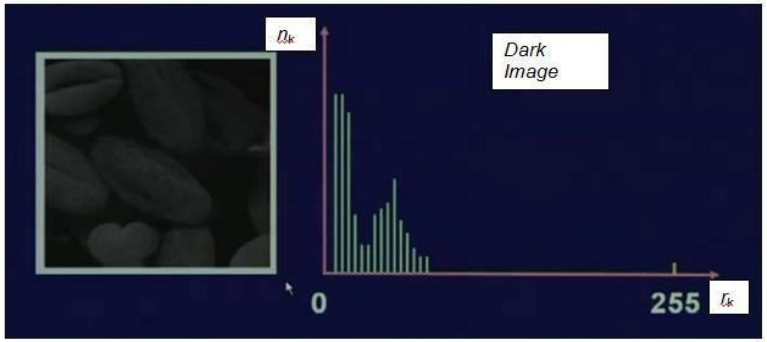
The dynamic range of an image is the exact subset of gray values (0,1,2, L-1) that are present in the image. The image histogram gives a clear indication on its dynamic range.

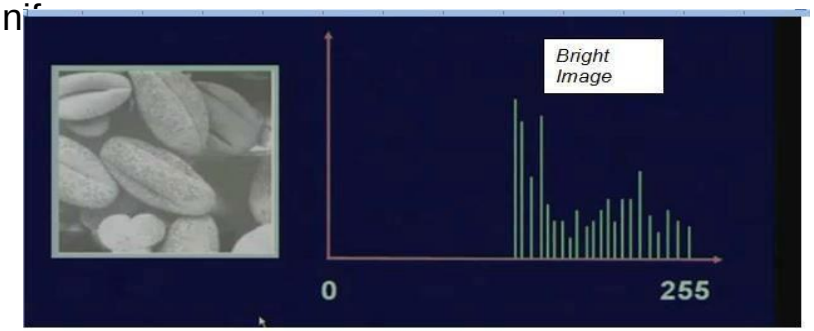
When the dynamic range of the image is concentrated on the lower side of the gray scale, the image will be dark image.

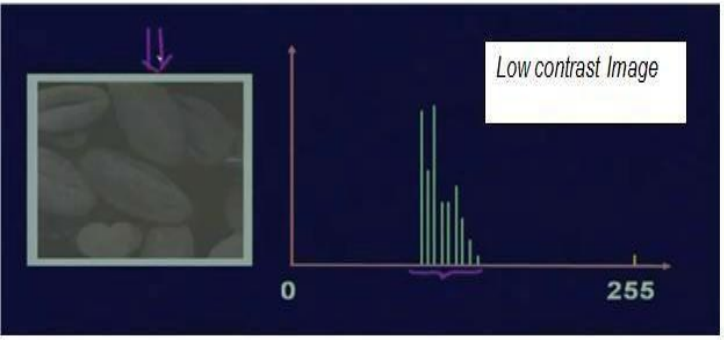
When the dynamic range of an image is biased towards the high side of the gray scale, the image will be bright or light image

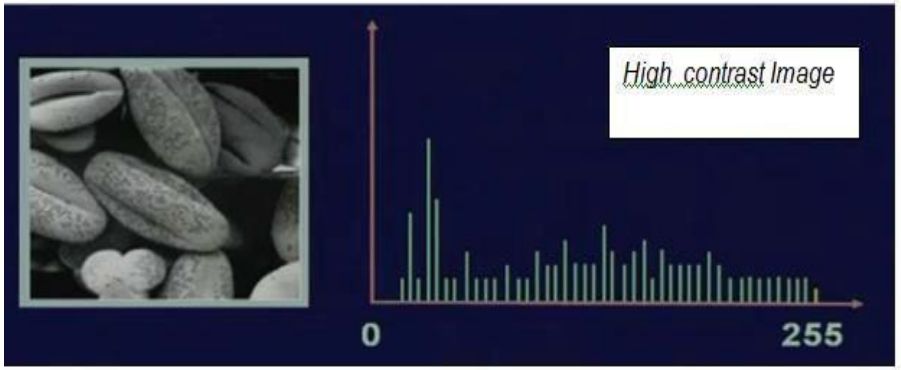
An image with a low contrast has a dynamic range that will be narrow and concentrated to the middle of the gray scale. The images will have dull or washed out look.

When the dynamic range of the image is significantly broad, the image will have a high

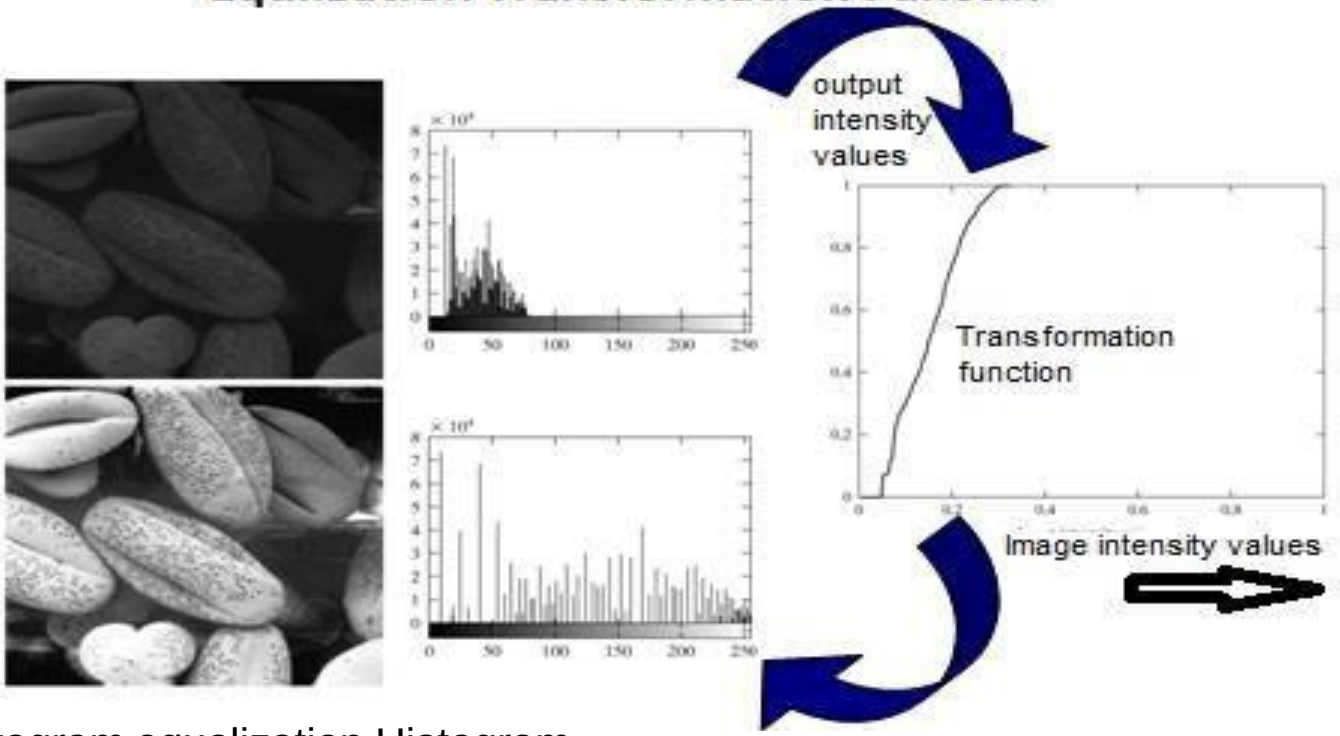








Equlization Transformation Functin



Histogram equalization Histogram Linearisation requires construction of a transformation function sk

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

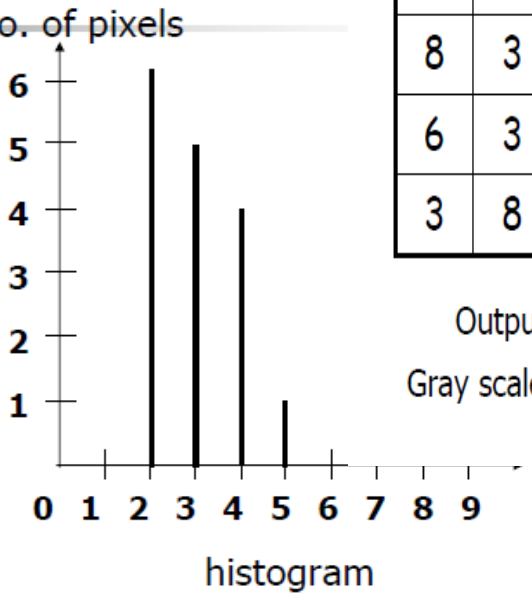
$$= \frac{(L-1)}{M \times N}\sum_{j=0}^{k} n_j \quad k = 0,1,2,\ldots,L-1$$

Example

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

4x4 image

Gray scale = [0,9]

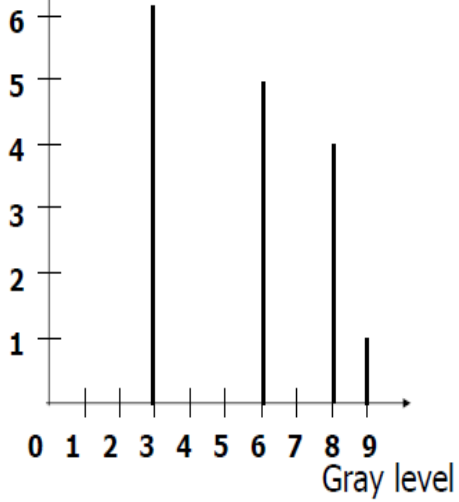


histogram

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]



Histogram equalization

Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
s x 9	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

HISTOGRAM EQUALISATION IS NOT ALWAYS DESIRED.
Some applications need a specified histogram to their requirements
This is called histogram specification or histogram matching

Histogram Matching: Discrete Cases

- Obtain $p_r(r_j)$ from the input image and then obtain the values of s_k, round the value to the integer range [0, L-1].

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

- Use the specified PDF and obtain the transformation function $G(z_q)$, round the value to the integer range [0, L-1].

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

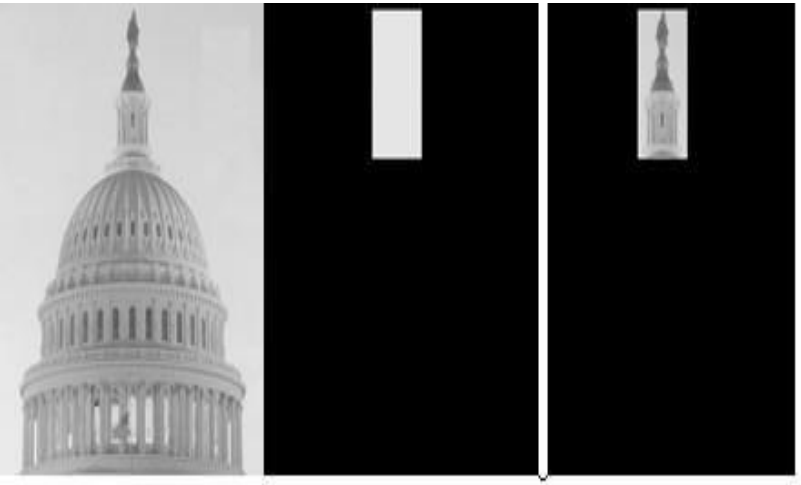
- Mapping from s_k to z_q

$$z_q = G^{-1}(s_k)$$

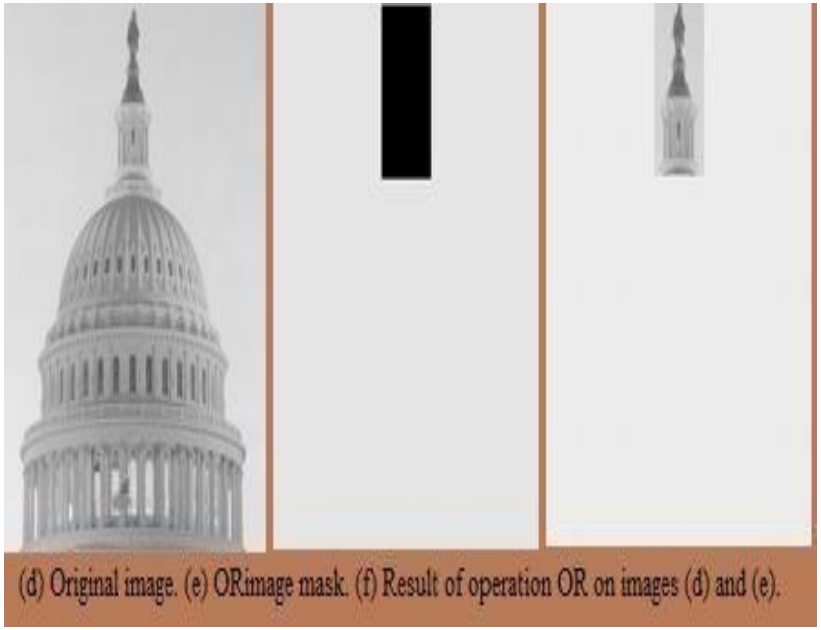
. two-step process

- perform histogram equalization on the image

- perform a gray-level mapping using the inverse of the desired cumulative histogram



(a) Original image. (b)AND image mask. (c) Result of the AND operation on images (a)&b



(d) Original image. (e) ORimage mask. (f) Result of operation OR on images (d) and (e).

Arithmetic operations

Addition:

Image averaging will reduce the noise. Images are to be registered before adding. An important application of image averaging is in the field of astronomy, where imaging with very low light levels is routine, causing sensor noise frequently to render single images virtually useless for analysis

g(x, y) = f(x, y) + η (x, y)

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

As *K* increases, indicate that the variability (noise) of the pixel values at each location (x, y) decreases

In practice, the images gi(x, y) must be registered (aligned) in order to avoid the introduction of blurring and other artifacts in the output image.

Subtraction

A frequent application of image subtraction is in the enhancement of differences between images. Black (0 values) in difference image indicate the location where there is no difference between the images.

One of the most commercially successful and beneficial uses of image subtraction is in the area of medical imaging called *mask mode radiography*

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

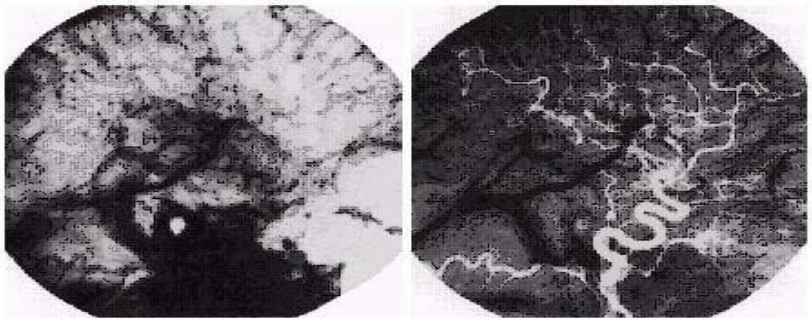
Image of a digital angiography. Live image and mask image with fluid injected. Difference will be useful to identify the blocked fine blood vessels.

The difference of two 8 bit images can range from -255 to 255, and the sum of two images can range from 0 to 510.

Given and f(x,y) image, $f_m = f - min(f)$ *which creates an image whose min value is zero.*

fs = k [fm / max (fm)],

fs is a scaled image whose values of k are 0 to 255. For 8 bit image k=255,



mask image

an image (taken after injection of a contrast medium (iodine) into the bloodstream) with mask

The output pixels are set of elements not in A.All elements in A become zero and the others to 1All

AND operation is the set of coordinates common to A and B

The output pixels belong to either A or B or Both

Exclusive or: The output pixels belong to either A or B but not to Both

An image multiplication and Division

An image multiplication and Division method is used in **shading correction.**

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

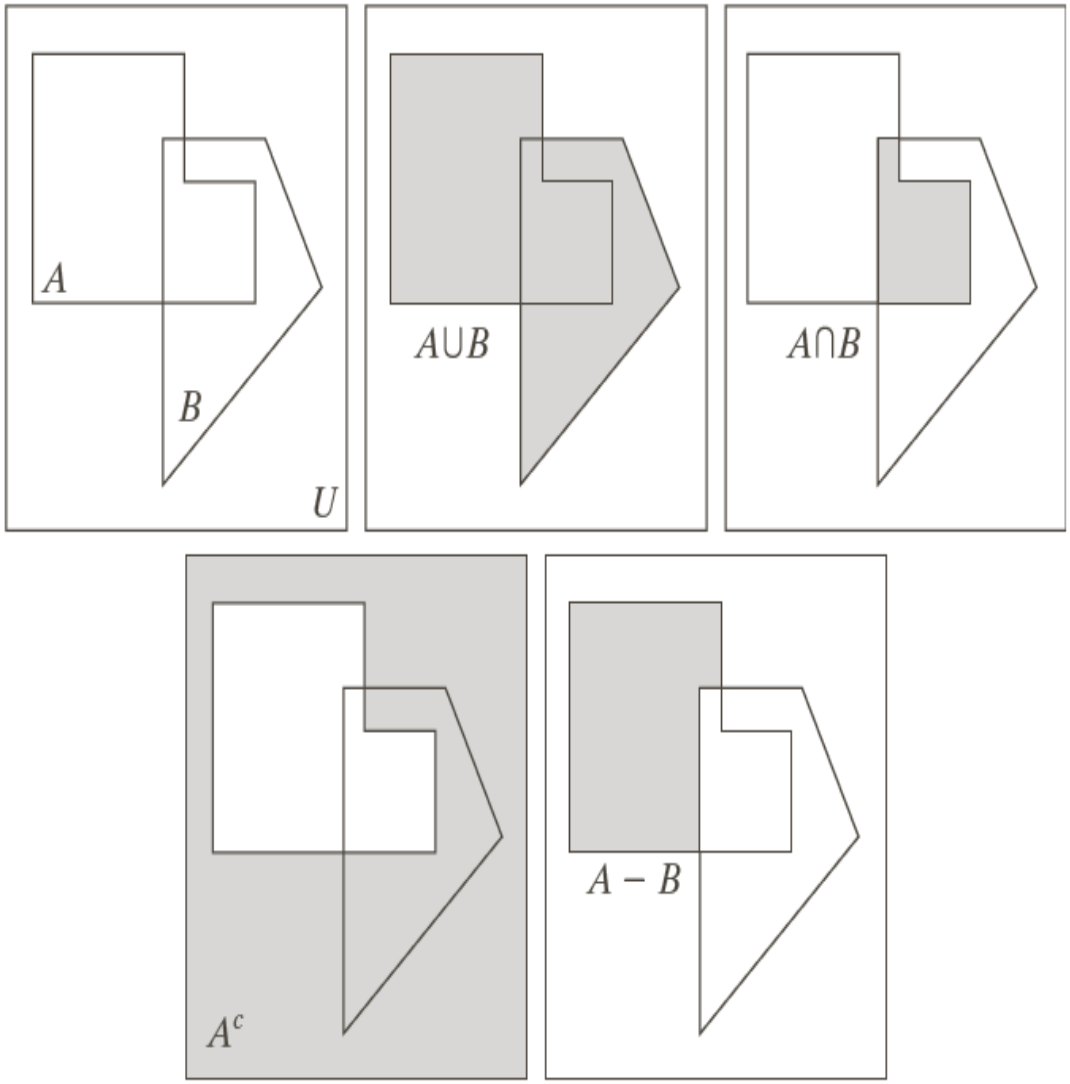
g(x, y) is sensed image

f(x, y) is perfect image

h (x, y) is shading function.

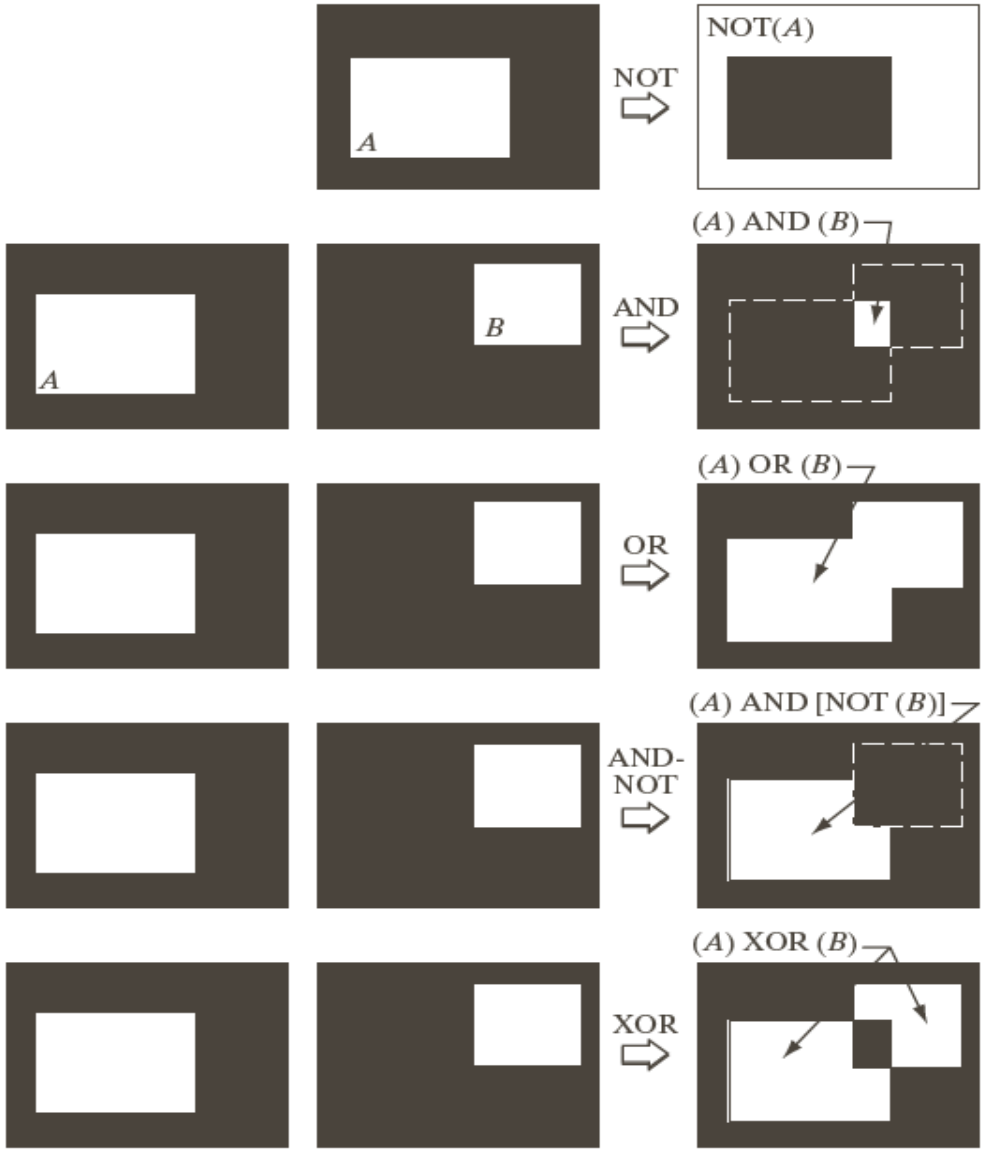
If h(x,y) is known, the sensed image can be multiplied with inverse of h(x,y) to get f(x,y) that is dividing g(x,y) by h(x,y)

Another use of multiplication is Region Of Interest (ROI). Multiplication of a given image by mask image that has 1s in the ROI and 0s elsewhere. There can be more than one ROI in the mask image.



a b c
d e

(a) Two sets of coordinates, A and B, in 2-D space. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B. In (b)–(e) the shaded areas represent the member of the set operation indicated.



The output pixels are set of elements not in A.All elements in A become zero and the others to 1All

AND operation is the set of coordinates common to A and B

The output pixels belong to either A or B or Both

Exclusive or: The output pixels belong to either A or B but not to Both

Let g(x,y) denote a corrupted image by adding noise η(x,y) to a noiseless image f(x,y):

$$g(x,y)=f(x,y)+\eta(x,y)$$

The noise has zero mean value $E[z_i]=0$

At every pair of coordinates $z_i=(x_i,y_i)$ the noise is uncorrelated $E[z_i z_j]=0$

The noise effect is reduced by averaging a set of K noisy images. The new image is

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

Spatial filters : spatial masks, kernels, templates, windows

Linear Filters and Non linear filters based on the operation performed on the image. Filtering means accepting (passing) or rejecting some frequencies. Mechanics of spatial filtering

f(x, y) → Filter → g(x, y)

Neighbourhood (a small rectangle) 3x3, 5x5,7x7 etc

Window centre moves from the first (0,0) pixel and moves till the end of first row, then second row and till the last pixel (M-1, N-1) of the input image

A pre defined operation on the Input Image

Filtering creates a new pixel in the image at window neighborhood centre as it moves.

Filtered image

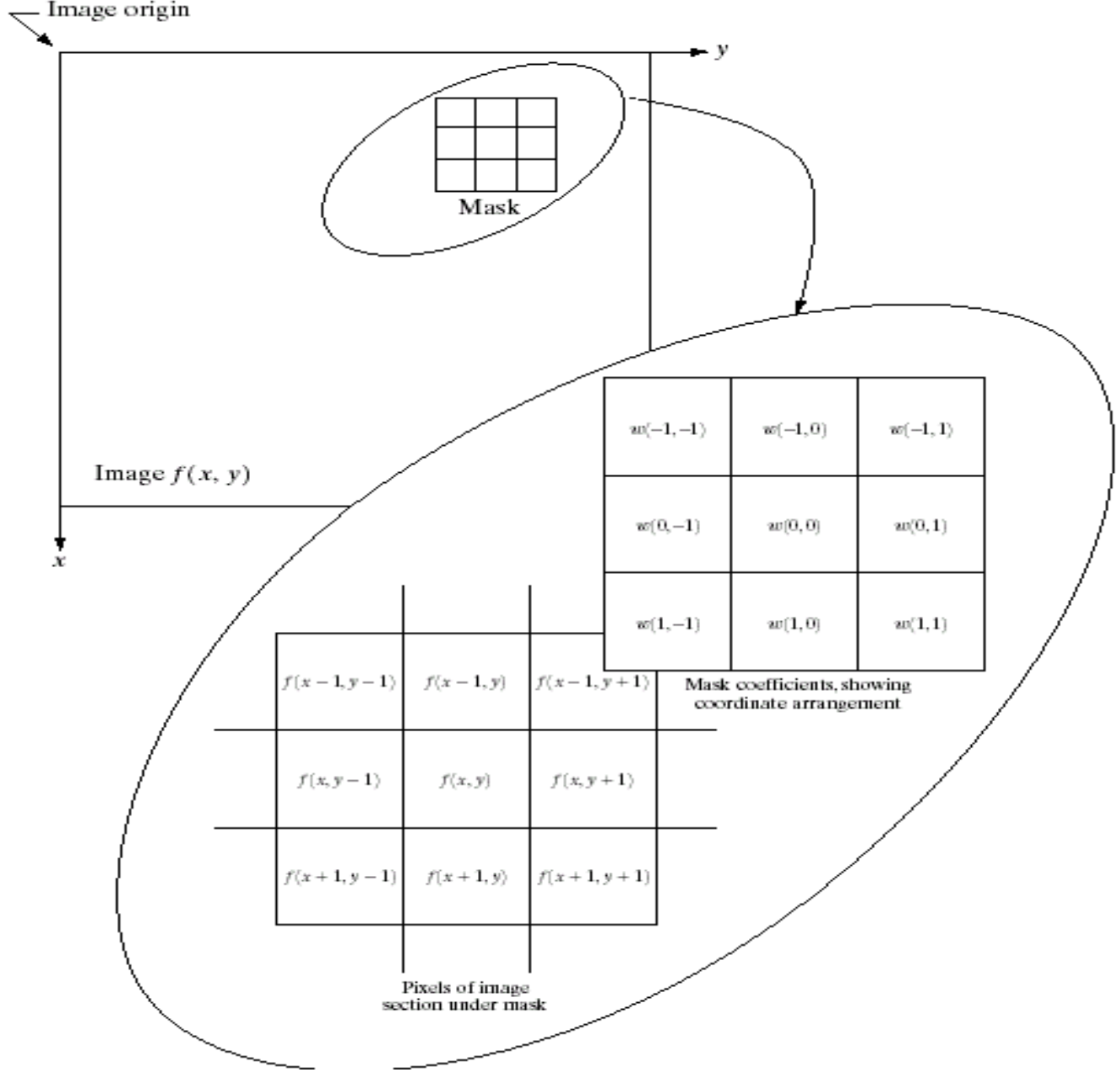
-1,-1 W_1	*-1,0* W_2	*-1,-1* W_3
0,-1 W_4	*0,0* W_5	*0,1* W_6
1,-1 W_7	*1,0* W_8	*1,1* W_9

At any point (x,y) in the image, the response g(x,y) of the filter is the sum of products of the filter coefficients and the image response and the image pixels encompassed by the filter.

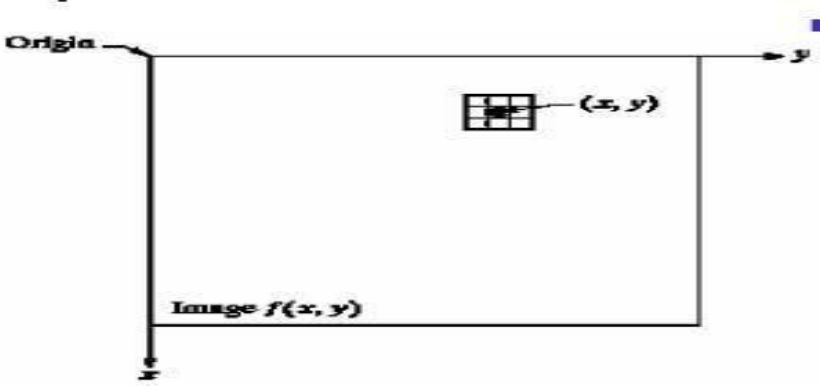
Observe that he filter w(0,0) aligns with the pixel at location (x,y) g(x,y)= w (-1,-1) f(x-1,y-1) + w (-1,-0) f(x-1,y) +

…+w(0,0)f(x,y)

+….+w(1,1)f(x+1,y+1)



The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

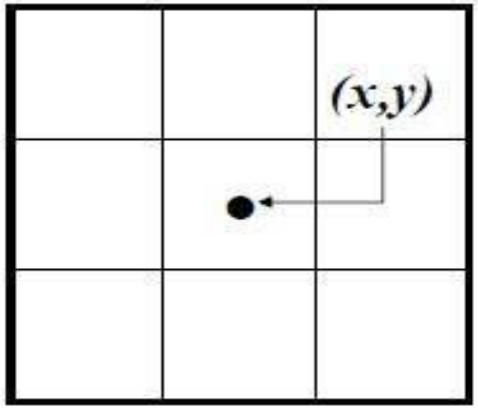


- Procedures that operate directly on pixels.

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

where

- $f(x,y)$ is the input image
- $g(x,y)$ is the processed image
- T is an operator on f defined over some neighborhood of (x,y)



- Neighborhood of a point (x,y) can be defined by using a square/rectangular (common used) or circular subimage area centered at (x,y)
- The center of the subimage is moved from pixel to pixel starting at the top of the corner

simply move the filter mask from point to point in an image.
at each point (x,y), the response of the filter at that point is calculated using a predefined relationship

Smoothening (blurring) filter Low Pass filter integration	*Ideal LF Butterworth LF Gaussian LF*	*Noise reduction by removing sharp edges and Sharp intensity transitions Side effect is: This will blur sharp edges*	*Average: Average Filter R=1/9 [Σz_i], i=1 to 9 Box filter (if all coefficients are equal) Weighted Average: Mask will have different coefficients*	*Linear Filter*
Order statistic		*Salt and pepper noise or impulse noise removal*	*1.Median filter 50 percentile*	*Non linear filter*
Order statistic		*Max filter finds bright objects*	*2.Max filter (100 percentile) t 3.Min filter (zero percentile)*	
Sharpening filters differentiation	*Highlights sharpenin g intensity*	*Image sharpening Second derivative filter is better for edge detection*	*Differentiation or first order or gradient Second derivative (Laplacian filter)*	*gradient is Linear operator magnitude is Non linear*
		Image sharpening	*Second derivative is Laplacian*	*Linear*
Unsharp masking High Boost filtering		*image sharpening*		
First order derivatives for image sharpening		*image sharpening*		*Non Linear*

Smoothing Linear Filter or *averaging filters or Low pass filters*

The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called *averaging filters*. they also are referred to a *lowpass filters*.

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

1	1	1
1	1	1
1	1	1

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

1	2	1
2	4	2
1	2	1

a b

Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

Weighted Average mask: Central pixel usually have higher value. Weightage is inversely proportional to the distance of the pixel from centre of the mask.

T the general implementation for filtering an MxN image with a weighted averaging filter of size m x n (m and *n* odd) is given by the expression, m=2a+1 and n=2b+1,where a and b are nonnegative integers. an important application of spatial averaging is to blur an image for the purpose getting a gross representation of objects of interest, such that the intensity of smaller objects blends with the background; after filtering and thresholding

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

Smoothing Spatial Filter

Weighted average

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

$$g(x,y) = \frac{1}{16} \sum_{i=-1}^{1} \sum_{j=-1}^{1} w_{i,j}\, f(x+i, y+j)$$

Image from Hubble telescope, image processed by 5x5 averaging window, and image after thresholding (nasa)

Examples of Low Pass Masks (Local Averaging)

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

$$\frac{1}{25}\times\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

$$\frac{1}{49}\times\begin{bmatrix}1&1&1&1&1&1&1\\1&1&1&1&1&1&1\\1&1&1&1&1&1&1\\1&1&1&1&1&1&1\\1&1&1&1&1&1&1\\1&1&1&1&1&1&1\\1&1&1&1&1&1&1\end{bmatrix}$$

Popular techniques for lowpass spatial filtering

Uniform filtering

The most popular masks for low pass filtering are masks with all their coefficients positive and equal to each other as for example the mask shown below. Moreover, they sum up to 1 in order to maintain the mean of the image.

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

Gaussian filtering

The two dimensional Gaussian mask has values that attempts to approximate the continuous function. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero more than about three standard deviations from the mean, and so we can truncate the kernel at this point. The following shows a suitable integer-valued convolution kernel that approximates a Gaussian with a of 1.0.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}}$$

Order-Statistics (non linear)Filters

The best-known example in this category is the M*edian filter*, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median).

Order static filter / ;ŶoŶ-liŶeaɗ filter) / median filter Objective:Replace the valve of the pixel by the median of the intensity values in the neighbourhood of that pixel

Although the median filter is by far the most useful order-statistics filter in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but the reader will recall from basic statistics that ranking lends itself to many other possibilities. For example, using the 100th percentile results in the so-called *max filter*, which is useful in finding the brightest points in an image. The response of a 3*3 max filter is given by R=max [zk| k=1, 2, ,… 9]

The 0th percentile filter is the *min filter*, used for the opposite purpose. Example nonlinear spatial filters

–Median filter: Computes the median gray-level value of the neighborhood. Used for noise reduction.

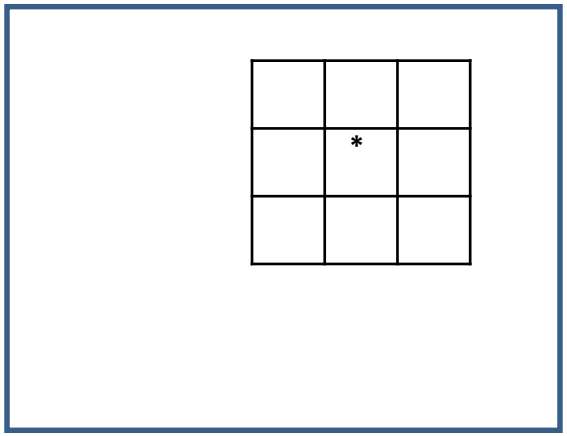
– Max filter: Used to find the brightest points in an image

–Min filter: Used to find the dimmest points in an image R = max{z | k =1,2,...,9}

R = min{z | k =1,2,...,9}

Non linear Median Filter

f (x,y)



101	86	99
100	106	103
91	102	109

f (x,y)

86
91
99
100
101
102
103
106
109

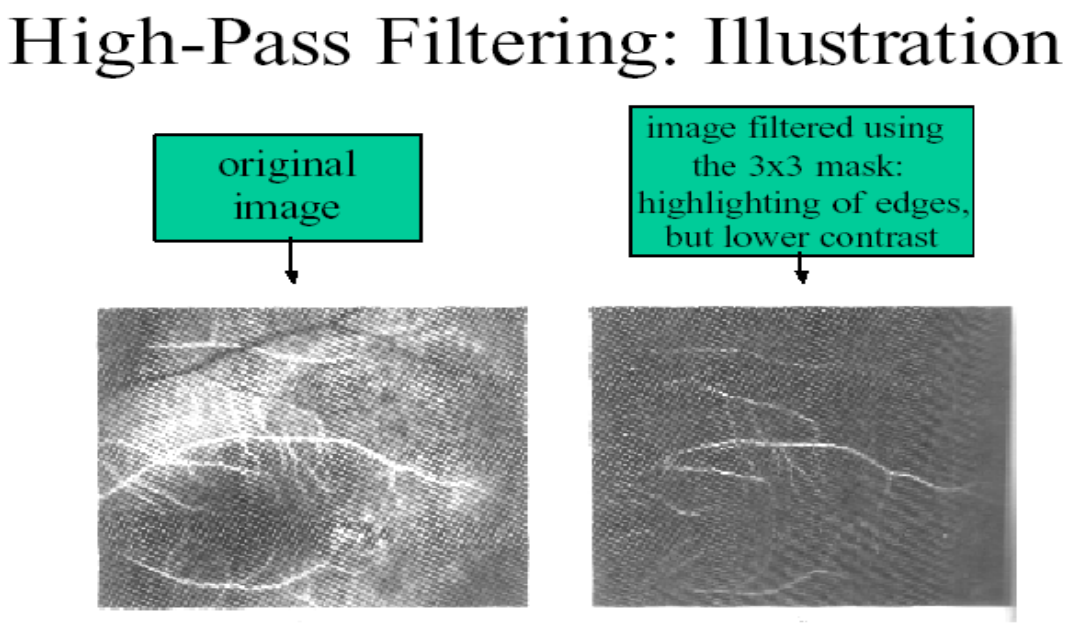
	101	

g (x,y)

High pass filter example

A high pass filtered image may be computed as the difference between the original image and a lowpass filtered version of that image as follows

High pass = Original – Low pass



Multiplying the original by an amplification factor yields a highboost or high-frequency-emphasis filter

Highpass filter example Unsharp masking

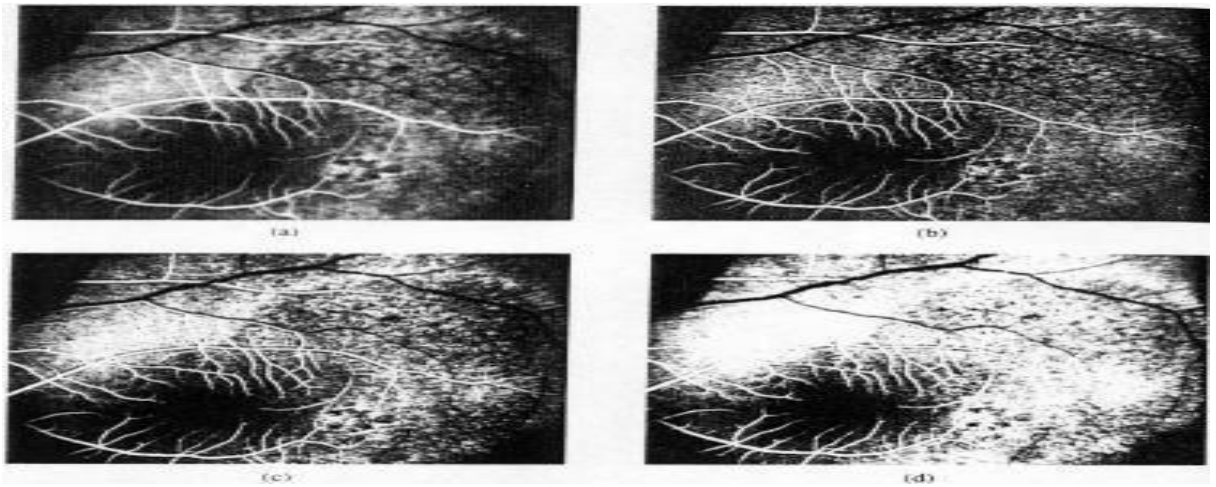
A high pass filtered image may be computed as the difference between the original image and a lowpass filtered version of that image as follows

Highpass = Original – Lowpass

•Multiplying the original by an amplification factor yields a highboost or high-frequency-empha

$$Highboost = A(Original) - Lowpass$$
$$= (A-1)(Original) + Original - Lowpass$$
$$= (A-1)(Original) + Highpass$$

A=1 for Highpass Filter



A=1.1

A=1.15

A=1.2

0	0	0
0	A	0
0	0	0

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

-1	-1	-1
-1	-1	-1
-1	-1	-1

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

-1	-1	-1
-1	$9A-1$	-1
-1	-1	-1

The high-boost filtered image looks more like the original with a degree of edge enhancement, depending on the value of .

A determines nature of filtering

High-Pass Filtering

- Shape of impulse response: +ve coefficients near its centre, -ve coefficients in periphery.
- E.g. 3x3 mask with +ve value in the middle, surrounded by 8 neighbours of –ve values.

1/9 x

–1	–1	–1
–1	8	–1
–1	–1	–1

Sharpening Spatial Filters

Since averaging is analogous to **integration**, it is logical to conclude that sharpening could be accomplished by **spatial differentiation.**

This section deals with various ways of defining and implementing operators for Image sharpening by digital differentiation.

Fundamentally, the strength the response of a **derivative operator** is proportional to the degree of discontinuity of the image at the point at which the operator is applied. Thus, **image differentiation enhances edges** and other discontinuities (such as noise) and deemphasizes areas with slowly varying gray- level values.

Use of first derivatives for Image Sharpening (Non linear) (EDGE enhancement)

About two dimensional high pass spatial filters

An edge is the boundary between two regions with relatively distinct grey level properties. The idea underlying most edge detection techniques is the computation of a local derivative operator.

The magnitude of the first derivative calculated within a neighborhood around the pixel of interest, can be used to detect the presence of an edge in an image.

First derivatives in image processing are implemented using the magnitude of the gradient.

For a function f(x, y), the gradient of f at coordinates (x, y) is defined as the two-dimension

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

The magnitude of this vector is given by

$$\begin{aligned} \nabla f &= \text{mag}(\nabla \mathbf{f}) \\ &= [G_x^2 + G_y^2]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}. \end{aligned}$$

The magnitude M (x,y) of this vector, generally referred to simply as the gradient

is $$\nabla f(x,y) = \text{mag}(\nabla f(x,y)) = \left[\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2\right]^{1/2}$$

Size of M(x,y) is same size as the original image. It is common practice to refer to this image as **gradient image** or simply as gradient.

Common practice is to approximate the gradient with absolute values which is simpler to implement as follows.

$$\nabla f(x,y) \cong \left|\frac{\partial f(x,y)}{\partial x}\right| + \left|\frac{\partial f(x,y)}{\partial y}\right|$$

A basic definition of the **first-order derivative of a one-dimensional function** f(x) is the difference

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

Similarly, we define a second-order derivative as the difference

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

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

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

In x and y directions

Sharpening Spatial Filters

First derivative

(1) must be zero in flat segments (areas of constant gray-level values);

(2) must be nonzero at the onset of a gray-level step or ramp; and

(3) must be nonzero along ramps.

Similarly, any definition of a second derivative

(1) must be zero in flat areas;

(2) must be nonzero at the onset and end of a gray-level step or ramp;

(3) must be zero along ramps of constant slope.

The digital implementation of the two-dimensional Laplacian is obtained by summing these two components:

$$\frac{\partial^2 f}{\partial^2 x^2} + \frac{\partial^2 f}{\partial^2 y^2} = \nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y).$$

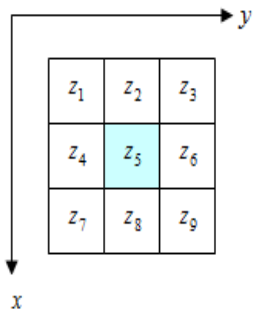
Laplacian operator (for enhancing fine details)

The **Laplacian** of a 2-D function $f(x,y)$ is a second order derivative defined as

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

In practice it can be also implemented using a 3x3 mask $\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$

Consider a pixel of interest $f(x,y) = z_5$ and a rectangular neighborhood of size $3 \times 3 = 9$ pixels (including the pixel of interest) as shown below.



The main disadvantage of the Laplacian operator is that it produces double edges

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

(a) Filter mask used to implement the digital Laplacian

0	−1	0
−1	4	−1
0	−1	0

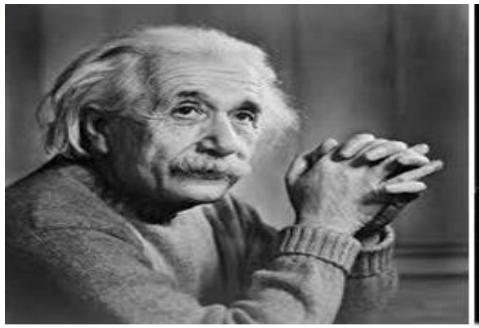
−1	−1	−1
−1	8	−1
−1	−1	−1

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

(b) Mask used to implement an extension that includes the diagonal neighbors.

(d) Two other implementations of the

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$



LAPLACIAN + ADDITION WITH ORIGINAL IMAGE DIRECTLY

0	-1	0
-1	5	-1
0	-1	0

=

0	0	0
0	1	0
0	0	0

+

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

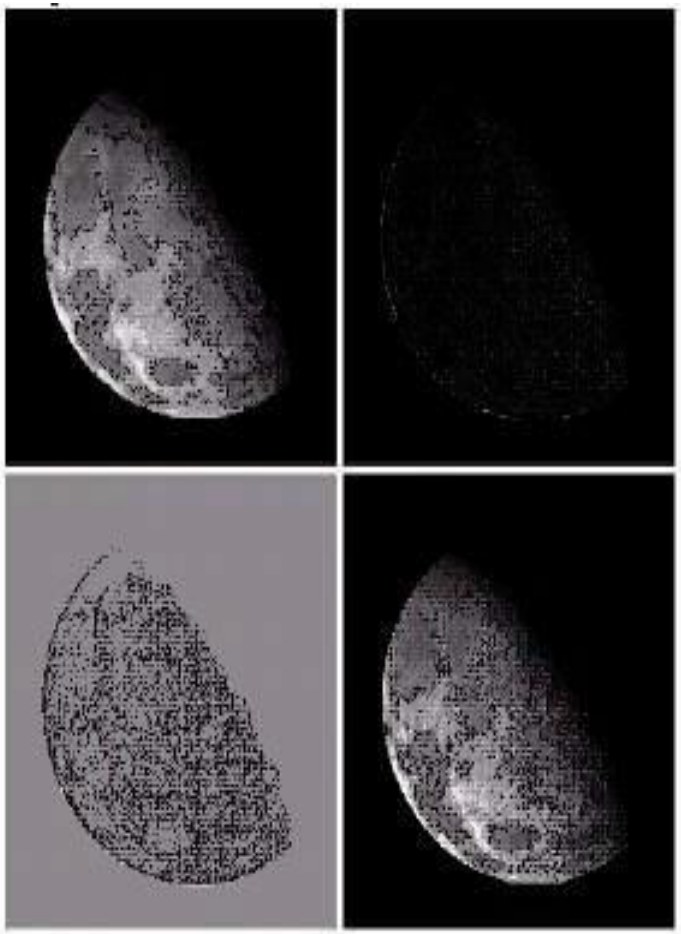
-1	-1	-1
-1	9	-1
-1	-1	-1

=

0	0	0
0	1	0
0	0	0

+

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



- a). image of the North pole of the moon
- b). Laplacian-filtered image with

1	1	1
1	-8	1
1	1	1

- c). Laplacian image scaled for display purposes
- d). image enhanced by addition with original image

Use of first derivatives for Image Sharpening (Non linear)

About two dimensional high pass spatial filters

An edge is the boundary between two regions with relatively distinct grey level properties. The idea underlying most edge detection techniques is the computation of a local derivative operator.

The magnitude of the first derivative calculated within a neighborhood around the pixel of interest, can be used to detect the presence of an edge in an image.

First derivatives in image processing are implemented using the magnitude of the gradient.

For a function f(x, y), the gradient of *f* at coordinates (x, y) is defined as the two-dimensional column vector

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

The magnitude of this vector is given by

$$\begin{aligned} \nabla f &= \text{mag}(\nabla \mathbf{f}) \\ &= [G_x^2 + G_y^2]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}. \end{aligned}$$

The magnitude M (x,y) of this vector, generally referred to simply as the gradient ∇f is

$$\nabla f(x,y) = \text{mag}(\nabla f(x,y)) = \left[\left(\frac{\partial f(x,y)}{\partial x} \right)^2 + \left(\frac{\partial f(x,y)}{\partial y} \right)^2 \right]^{1/2}$$

Size of M(x,y) is same size as the original image. It is common practice to refer to this image as **gradient image** or simply as gradient.

Common practice is to approximate the gradient with absolute values which is simpler to implement as follows.

$$\nabla f(x,y) \cong \left| \frac{\partial f(x,y)}{\partial x} \right| + \left| \frac{\partial f(x,y)}{\partial y} \right|$$

Gradient Mask

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- simplest approximation, 2x2

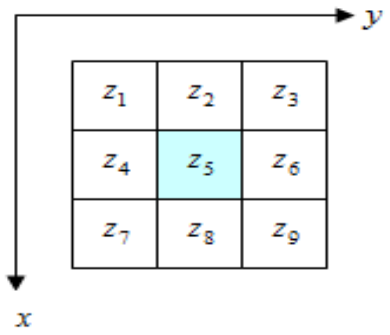
$$G_x = (z_8 - z_5) \quad \text{and} \quad G_y = (z_6 - z_5)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_8 - z_5)^2 + (z_6 - z_5)^2]^{1/2}$$

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

DERIVATIVE OPERATORS

Consider a pixel of interest $f(x,y)=z_5$ and a rectangular neighborhood of size $3\times3=9$ pixels (including the pixel of interest) as shown below.



Roberts operator

Above Equation can be approximated at point Z_5 in a number of ways. The simplest is to use the difference $(Z_5 - Z_8)$ in the x direction and $(Z_5 - Z_6)$ in the y direction. This approximation is known as the **Roberts** operator, and is expressed mathematically as follows

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

Another approach for approximating the equation is to use cross differences

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

Above Equations can be implemented by using the following masks. The original image is convolved with both masks separately and the absolute values of the two outputs of the convolutions are added.

1	0
-1	0

1	-1
0	0

Roberts operator

1	0
0	-1

0	1
-1	0

Roberts operator

- Roberts cross-gradient operators, 2x2

$$G_x = (z_9 - z_5) \quad \text{and} \quad G_y = (z_8 - z_6)$$

$$\nabla f = [G_x^2 + G_y^2]^{1/2} = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2}$$

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

-1	0
0	1

0	-1
1	0

Prewitt operator

$$\nabla f(x, y) \cong \left|\frac{\partial f(x, y)}{\partial x}\right| + \left|\frac{\partial f(x, y)}{\partial y}\right|$$

Another approximation to the above equation, but using a 3 x 3 matrix is:

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

The difference between the first and third rows approximates the derivative in the *x direction*

- The difference between the first and third columns approximates the derivative in the *y direction*
- The *Prewitt operator masks may be used to implement the* above approximation

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

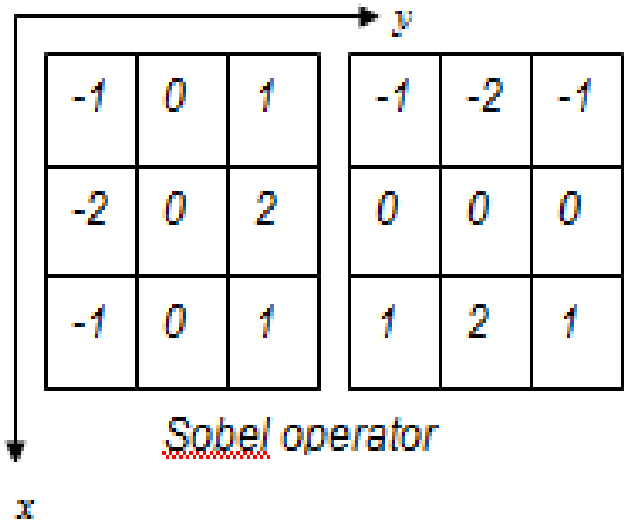
-1	0	1
-1	0	1
-1	0	1

Sobel operator.

Definition and comparison with the Prewitt operator (gives weightage to centre pixel)
The most popular approximation of equation (1) but using a 3×3 mask is the following.

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

This approximation is known as the Sobel operator.



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

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

Sobel operator

If we consider the left mask of the Sobel operator, this causes differentiation along the y direction.

- Sobel operators, 3x3

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

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

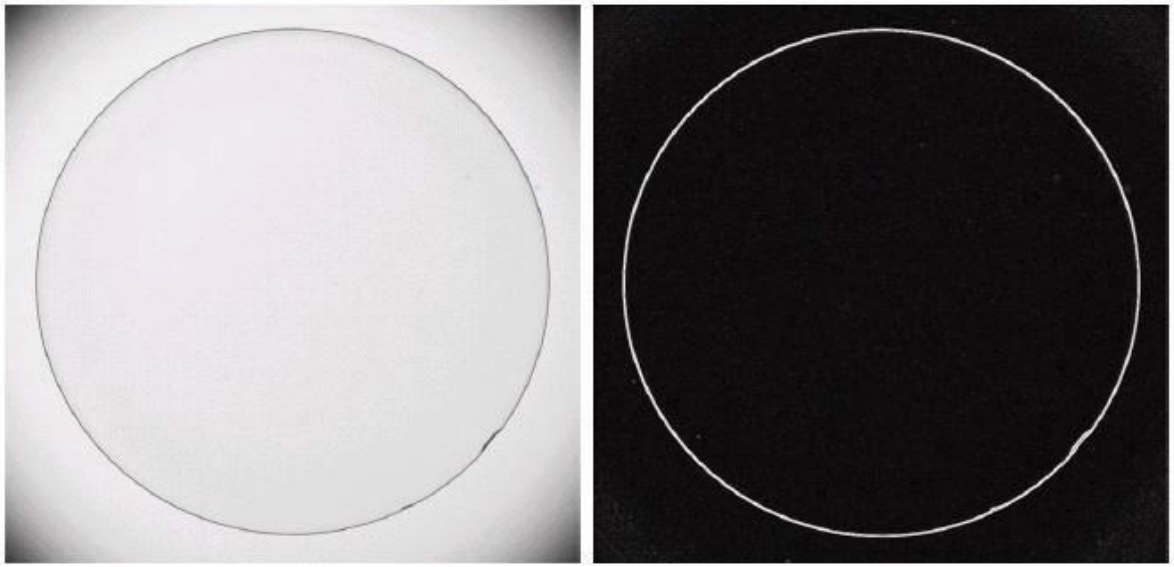
$$\nabla f \approx |G_x| + |G_y|$$

the weight value 2 is to achieve smoothing by giving more important to the center point

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

150

the summation of coefficients in all masks equals 0, indicating that they would give a response of 0 in an area of constant gray level



a b

FIGURE 3.45
Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Filtering in the Frequency Domain

Filters in the frequency domain can be divided in four groups: Low pass filters

…….IMAGE BLUR

Remove frequencies away from the origin
Commonly, frequency response of these filters is symmetric around the origin;
The largest amount of energy is concentrated on low frequencies, but it represents just image luminance and visually not so important part of image.

High pass filters ……EDGES DETECTION
Remove signal components around and further away from origin
Small energy on high frequency corresponds to visually very important image features such as edges and details. Sharpening = boosting high frequency pixels

Band pass filters
Allows frequency in the band between lowest and the highest frequencies;

Stop band filters

Remove frequency band.

To remove certain frequencies, set their corresponding F(u) coefficients to zero

Low pass filters (smoothing filters)

Ideal Low Pass filters ILPF
Butterworth low pass filters BLPF
Gaussian low pass filters GLPF

High Pass Filters (Sharpening filters)

Ideal High pass filters IHPF
Butterworth High pass filters BHPF
Gaussian High pass filters Laplacian in frequency domain GHPF
High boost , high frequency emphasis filters

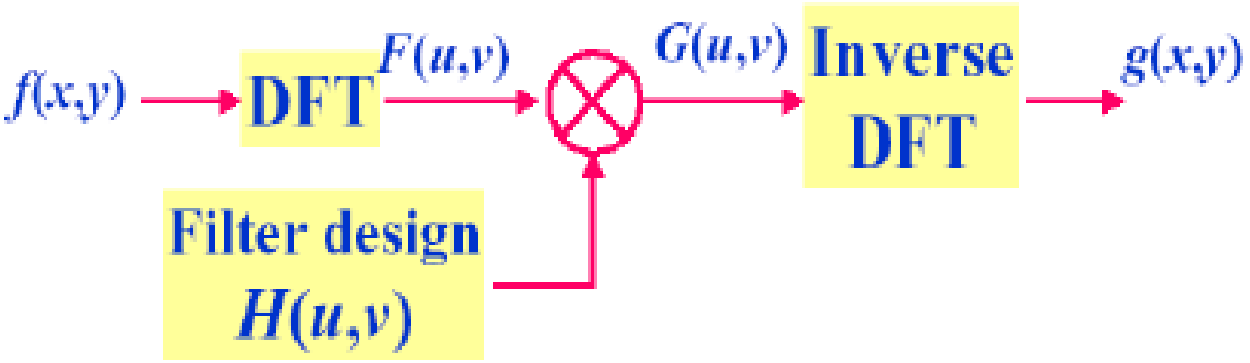
Homomorthic filters

log n or ln

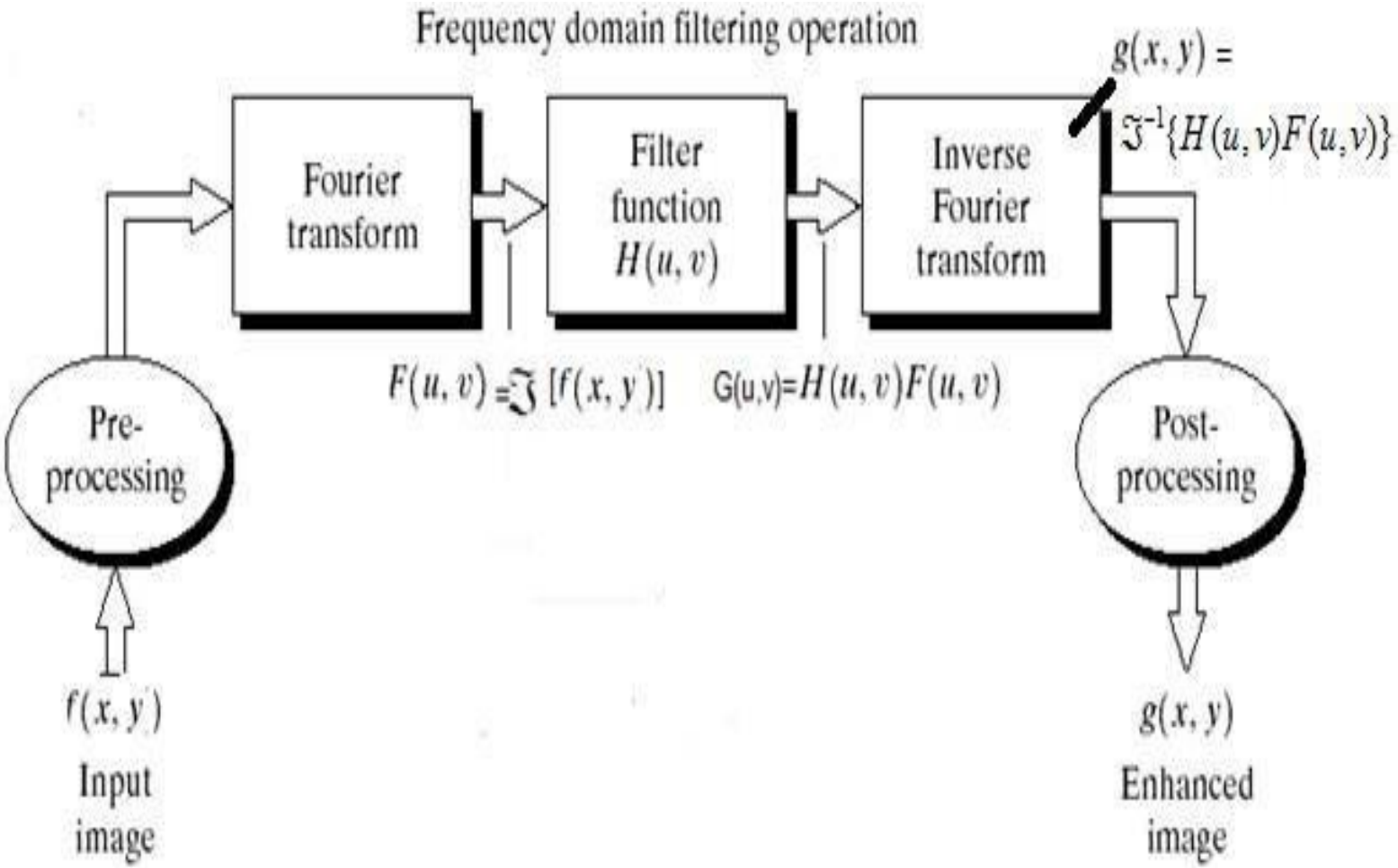
f(x,y)= i(x,y) r(x,y)

F[f(u,v)]= F [log n [i(x,y) r(x,y)] = F [log n [i(x,y)] + F [log n [r(x,y)]

Basic steps for filtering in the frequency domain

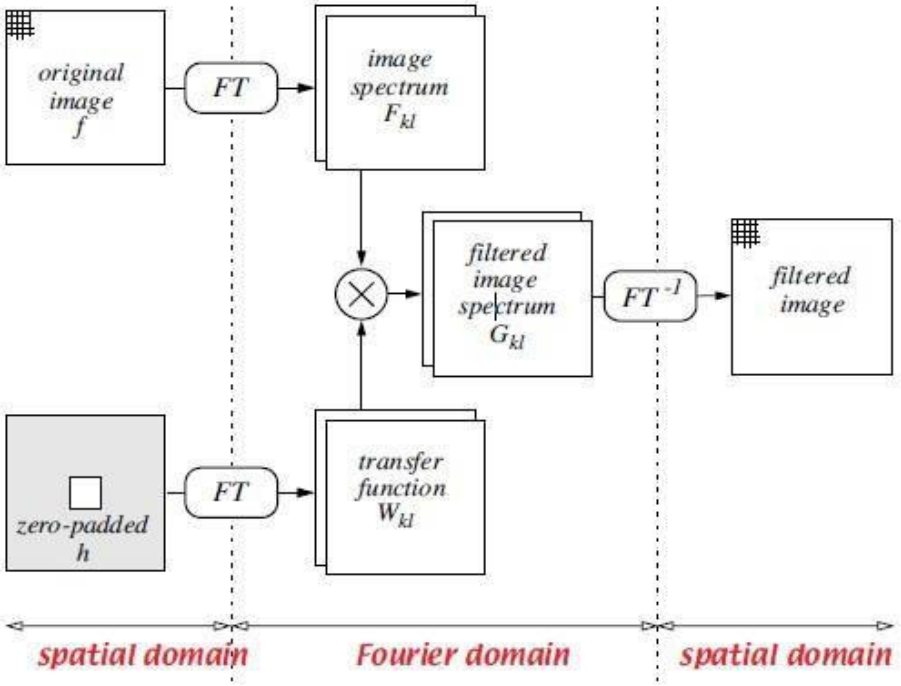


- $H(u,v)$ plays an important role⇨ determine how the enhanced image looks like
- A lowpass $H(u,v)$ results in a *blurring* effect; a highpass $H(u,v)$ results in a *sharpening* effect
- High frequency: edges and sharp details in an image
- Low frequency: slowly varying characteristics of an image (e.g. overall contrast and average intensity)



Basic steps for filtering in the frequency domain.

IMAGE ENHANCEMENT III (Fourier)



Note, multiplication in Fourier domain is full complex operation

Filtering in the Frequency Domain

•***Basic Steps for zero padding***

Zero Pad the input image f(x,y) to p =2M-1, and q=2N-1, if arrays are of same size.

If functions f(x,y) and h(x,y) are of size MXN and KXL, respectively, choose to pad with zeros:

P ш M + N - 1

Q ш K + L- 1

Zero-pad h and f

- Pad both to at least
- Radix-2 FFT requires power of 2

For example, if M = N = 512 and K = L = 16, then P = Q = 1024

- Results in linear convolution
- Extract center MxN

Practical implementation: Overlap-add partitions image into I x J smaller blocks, pads each block and filter h to same size, filters each block separately, and recombines:

Filtering in the Frequency Domain

•***Basic Steps for Filtering in the Frequency Domain:***

1. Multiply the input padded image by (-1)$^{x+y}$ to center the transform.
2. Compute F(u,v), the DFT of the image from (1).
3. Multiply F(u,v) by a filter function H(u,v).
4. Compute the inverse DFT of the result in (3).
5. Obtain the real part of the result in (4).
6. Multiply the result in (5) by (-1)$^{x+y}$.

Given the filter H(u,v) (filter transfer function OR filter or filter function) in the frequency domain, the Fourier transform of the output image (filtered image) is given by:

G (u,v)= H (u,v) F (u,v) ***Step (3) is array multiplication***

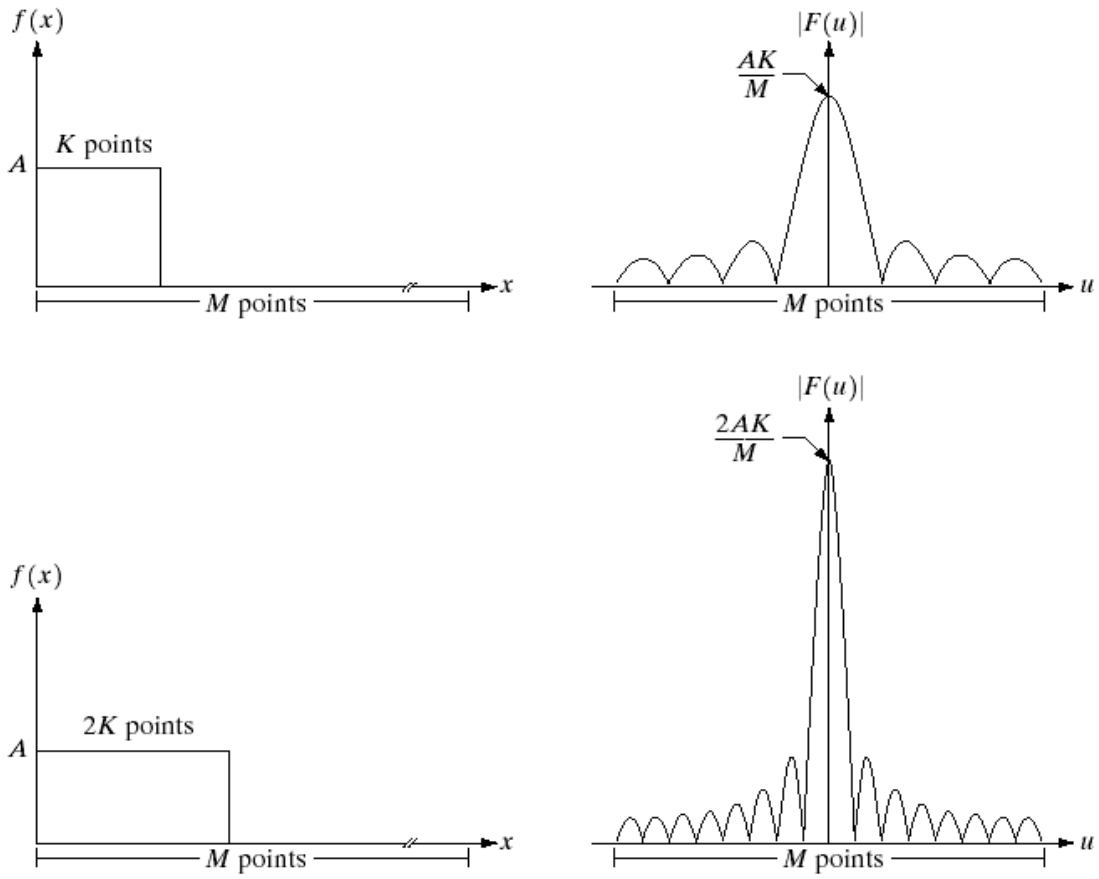
The filtered image g(x,y) is simply the inverse Fourier transform of G(u,v).

g (x,y) = F^{-1} [G(u,v)] = F^{-1} [H (u,v) F (u,v)] ***Step (4)***

$$g_p(x, y) = \{\text{real}\left[\Im^{-1}[G(u, v)]\right]\}(-1)^{x+y}$$

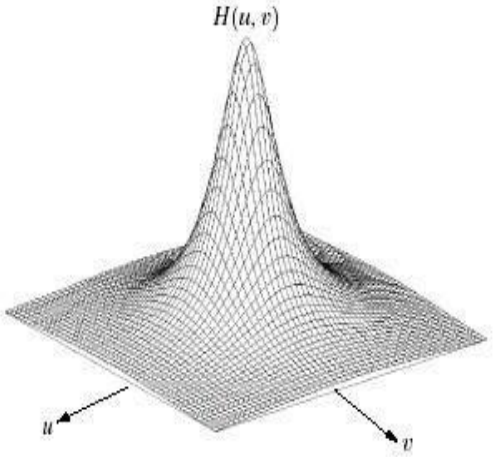
F, H, g, are arrays of same size as in input image. F^{-1} is IDFT.

1. Multiply the input image by $(-1)^{x+y}$ to center the transform
2. Compute F(u,v), the DFT of the image from (1)
3. Multiply F(u,v) by a filter function H(u,v)
4. Compute the inverse DFT of the result in (3)
5. Obtain the real part of the result in (4)
6. Multiply the result in (5) by $(-1)^{x+y}$

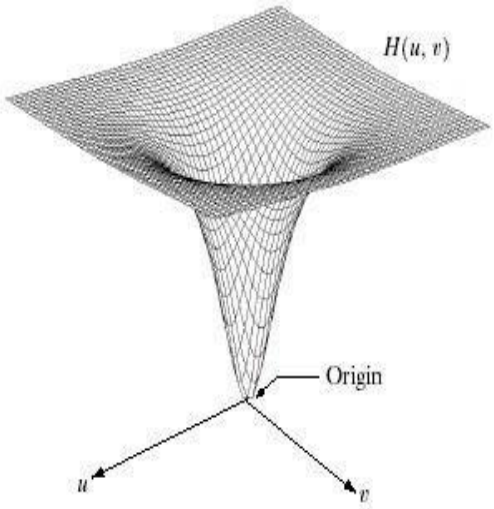


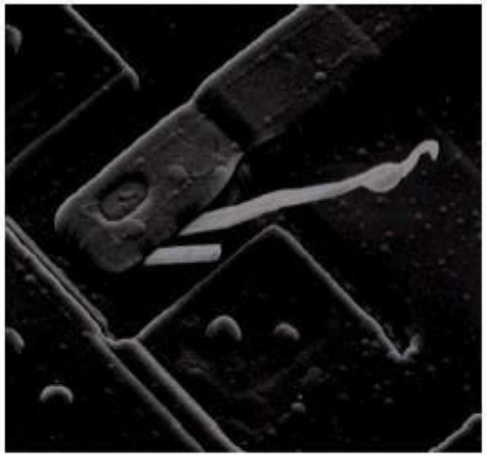
a b
c d

FIGURE 4.2 (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.



Low Pass Filter attenuate high frequencies while passing low frequencies.





High Pass Filter attenuate low frequencies while passing high frequencies.

a b
c d

(a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image
(c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image

Correspondence between filtering in spatial and frequency domains

Filtering in frequency domain is multiplication of filter times fourier transform of the input image

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

Let us find out equivalent of frequency domain filter H (u,v) in spatial domain.

Consider f (x,y) = ω(x,y), we know f(u,v) =1
Then filtered output F^{-1} [H (u,v) F (u,v)] = F^{-1} [H (u,v)]

But this is inverse transform of the frequency domain filter
But this is nothing but filter in the spatial domain.
Conversely, if we take a forward Fourier transform of a spatial filter, we get its fourier domain representation

Therefore, two filters form a transform pair
h(x,y) ⟺ H (u,v),

Since h(x,y) can be obtained from the response of a frequency domain filter to an impulse, h(x,y) spatial filter is some times referred as r finite impulse response filter (FIR) of H(u,v)

$f(x,y) * h(x,y) \Leftrightarrow F(u,v)\,H(u,v)$

Spatial domain processing, in general, is faster than frequency domain processing.

In some cases, it is desirable to generate spatial domain mask that approximates a given frequency domain filter.

The following procedure is one way to create these masks in a *least square error sense.*

Recall that filter processing in frequency domain, which is product of filter and function, becomes convolution of function and filter in spatial domain.

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

Output image — *Filter Transfer Function* — *Input image*

➢ The analogous operation in spatial domain can be implemented as follow

$$g(x,y) = \sum_{i=0}^{N-1}\sum_{k=0}^{N-1} h(x-i, y-k)\, f(i,k)$$

with x, y = 0, 1, 2, ……., *N*-1.

➢ Note that all functions are properly extended to generate a linear convolution in frequency domain.

➢ h is the spatial domain representation of the $H(u,v)$ and it is called *spatial convolution mask.*

➢ The relationship between $h(x,y)$ and $H(u,v)$ is defined as follow

$$H(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} h(x,y)\, e^{-j2\pi\frac{ux+vy}{N}}$$

where $u, v = 0, 1, 2, \ldots\ldots, N-1$.

➢ Now, if we restrict $h(x,y)$ as follow

$$\hat{h}(x,y) = \begin{cases} h(x,y), & 0 \le x, y \le n-1 \\ 0 \quad , & n \le x, y \le N-1 \end{cases}$$

➢ This restriction in effect creates an n x n convolution mask $\hat{h}$, with Fourier transform of

$$\hat{H}(u,v) = \frac{1}{N}\sum_{x=0}^{n-1}\sum_{y=0}^{n-1}\hat{h}(x,y)\,e^{-j2\pi\frac{ux+vy}{N}}$$

where $u, v = 0, 1, 2, \ldots\ldots, N\text{-}1$.

➢ The objective is to find the coefficients of $\hat{h}(x,y)$ such that mean square error is minimized.

$$e^2 = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\left|\hat{H}(u,v) - H(u,v)\right|^2$$

Consider the following filter transfer function:

$$H(u,v) = \begin{cases} 0 & if\ (u,v) = (M/2, N/2) \\ 1 & otherwise \end{cases}$$

This filter will set F(0,0) to zero and leave all the other frequency components. Such a filter is called the notch filter, since it is constant function with a hole (notch) at the origin.

HOMOMORPHIC FILTERING

an image can be modeled mathematically in terms of illumination and reflectance as follow:

$f(x,y) = I(x,y)\, r(x,y)$

Note that:

$F\{ f(x, y)\} \neq F\{i(x, y)\}\ F\{r(x, y)\}$

To accomplish separability, first map the model to natural log domain and then take the Fourier transform of it. $z(x, y) = ln\{ f(x, y)\} = ln\{i(x, y)\} + ln\{r(x, y)\}$

Then,

$F\{z(x, y)\} = F\{ln\ i(x, y)\} + F\{ln\ r(x, y)\}$

or

$Z(u, v) = I(u, v) + R(u, v)$

Now, if we process Z(u,v) by means of a filter function H(u,v) then,

➢ Now, if we process Z(u,v) by means of a filter function H(u,v) then,

$$S(u,v) = H(u,v)Z(u,v) =$$
$$= H(u,v)I(u,v) + H(u,v)R(u,v)$$

➢ Taking inverse Fourier transform of S(u,v) brings the result back into natural log domain,

$$s(x,y) = F^{-1}\{S(u,v)\}$$
$$= F^{-1}\{H(u,v)I(u,v)\} + F^{-1}\{H(u,v)R(u,v)\}$$

By letting

$$i'(x,y) = F^{-1}\{H(u,v)I(u,v)\}$$
$$r'(x,y) = F^{-1}\{H(u,v)R(u,v)\}$$

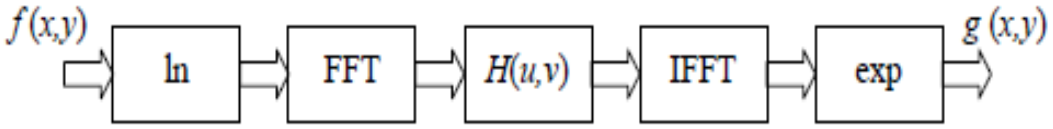
- Now, to get back to spatial domain, we need to get inverse transform of natural log, which is exponential,

$$s(x,y) = i'(x,y) + r'(x,y)$$
$$g(x,y) = \exp[s(x,y)]$$
$$= \exp[i'(x,y)] \cdot \exp[r'(x,y)]$$
$$= i_o(x,y) r_o(x,y)$$

Where $i_o(x,y)$ is illumination and $r_o(x,y)$ is reflectance components of the output image.

- This method is based on a special case of a class of systems known as *homomorphic systems*.

- The overall model in block diagram will look as follow:

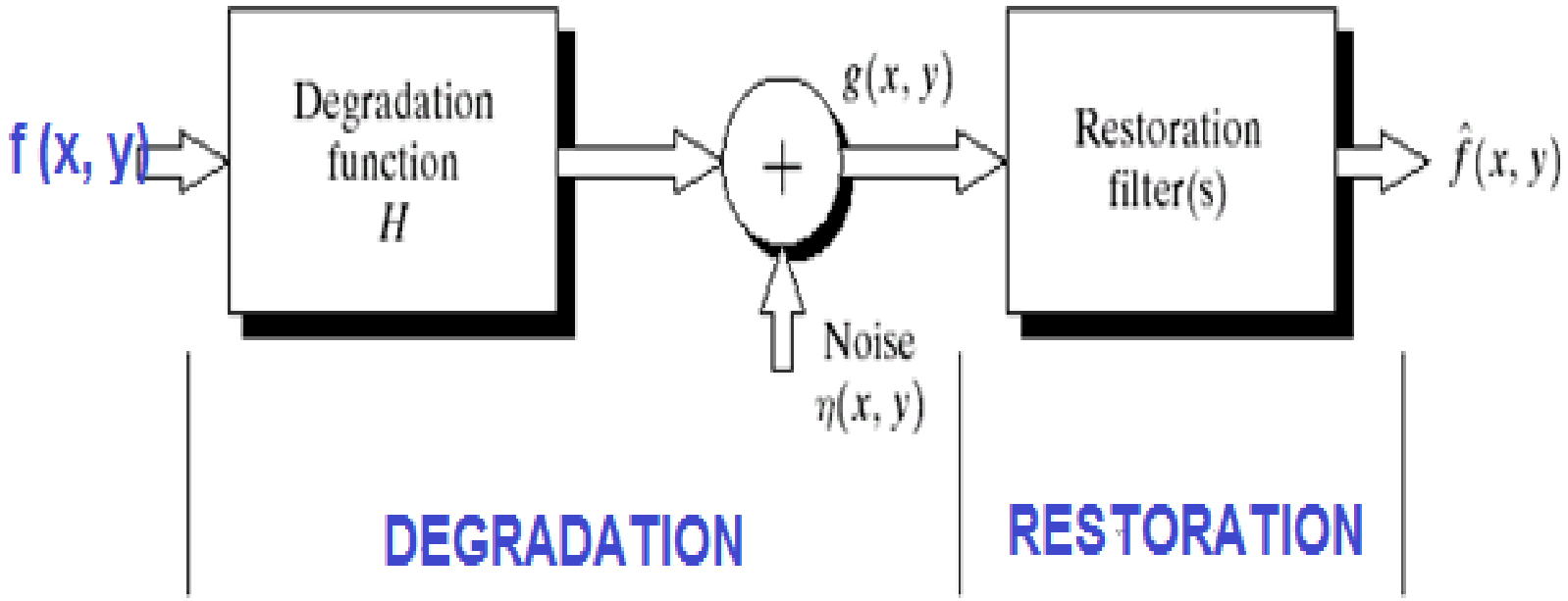


UNIT-III

IMAGE RESTORATION

Concept of Image Restoration

Image restoration is to restore a degraded image back to the original image while image enhancement is to manipulate the image so that it is suitable for a specific application.



Use a priori knowledge of the degradation

Modeling the degradation and apply the inverse process

Model for image degradation/restoration process

The objective of restoration is to obtain an estimate for the original image from its degraded version $g(x,y)$ while having some knowledge about the degradation function H and additive noise $\eta(x,y)$.

– Additive noise

$$g(x, y) = f(x, y) + \eta(x, y)$$

Linear blurring

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

– Linear blurring and additive noise

$$g(x, y) = f(x, y) * h(x, y) + \eta(x, y)$$

the degraded image in **spatial domain** is

$$g(x, y) = h(x, y) \otimes f(x, y) + \eta(x, y)$$

convolution

Therefore, in the **frequency domain** it is

$$G(u,v) = H(u, v)F(u, v) + N(u,v)$$

Degradation Models

Image degradation can occur for many reasons, some typical degradation models are

$$h(i,j) = \begin{cases} 1 & ai + bj = 0 \\ 0 & otherwise \end{cases}$$

Motion Blur: due to camera panning or subject moving quickly.

$$h(i,j) = Ke^{-\left(\frac{i^2+j^2}{2\sigma}\right)}$$

Atmospheric Blur: long exposure

$$h(i,j) = \begin{cases} \frac{1}{L^2} & -\frac{L}{2} \le i,j \le \frac{L}{2} \\ 0 & otherwise \end{cases}$$

Uniform 2D Blur

$$h(i,j) = \begin{cases} \frac{1}{\pi R^2} & i^2 + j^2 \le R^2 \\ 0 & otherwise \end{cases}$$

Out-of-Focus Blur

Restoration in the presence of noise only – Spatial filteri

$$g(x,y) = f(x,y) + \eta(x,y)$$

$$G(u,v) = F(u,v) + N(u,v)$$

- **Mean filters (, Order-statistic filters**

Get more data ! Capture N images of the same scene

gi(x,y) = f(x,y) + ni(x,y)

– Average to obtain new image

$$g_{ave}(x,y) = f(x,y) + n_{ave}(x,y)$$

Estimation of Noise

Consists of finding an image (or subimage) that contains only noise, and then using its histogram for the noise model

- Noise only images can be acquired by aiming the imaging device (e.g. camera) at a blank wall

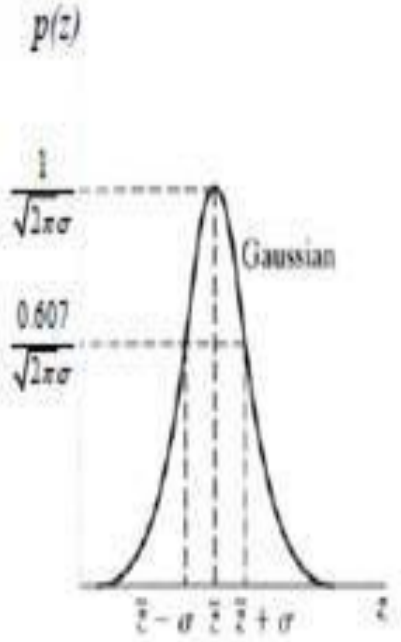
In case we cannot find "noise-only" images, a portion of the image is selected that has a known histogram, subtract the known values from the histogram, and what is left is our noise model.

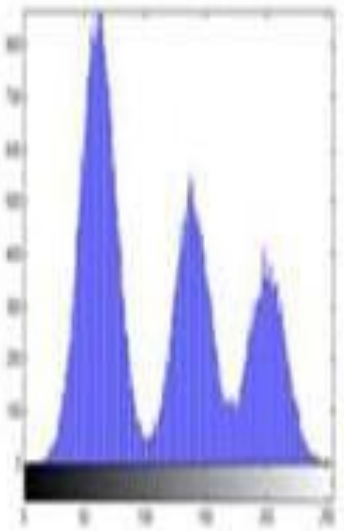
- To develop a valid model **many sub-images** need to be evaluated

Noise pdfs

1. Gaussian (normal)

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}}$$





Here z represents intensity, $\bar{z}$ is the mean (average) value of z and σ is its standard deviation. σ^2 is the variance of z.

Electronic circuit noise, sensor noise due to low illumination or high temperature.

2. Rayleigh noise is specified as

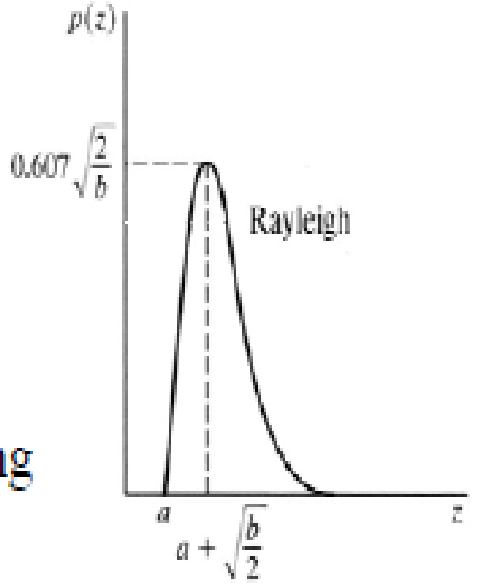
$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} & z \geq a \\ 0 & z < a \end{cases}$$

The mean and variance are given by

$$\bar{z} = a + \sqrt{\pi b/4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}$$

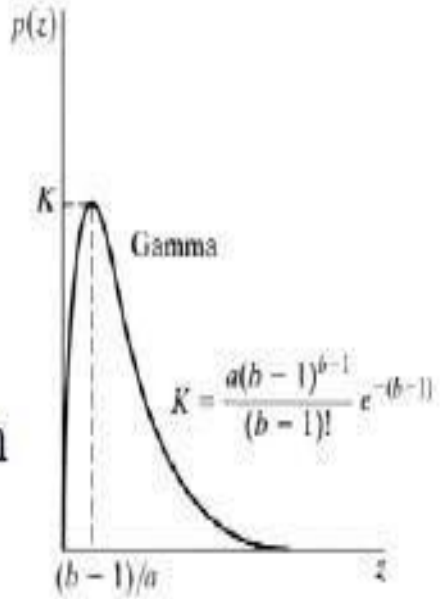
The Rayleigh density is useful for approximating skewed histograms. Used in range imaging.



Radar range and velocity images typically contain noise that
can be modeled by the Rayleigh distribution

3. Erlang noise is specified as

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$



Here $a > 0$ and b is a positive integer. The mean and variance are given by

$$\bar{z} = b/a$$

$$\sigma^2 = b/a^2$$

When the denominator is the gamma function, the pdf describes the gamma distribution.

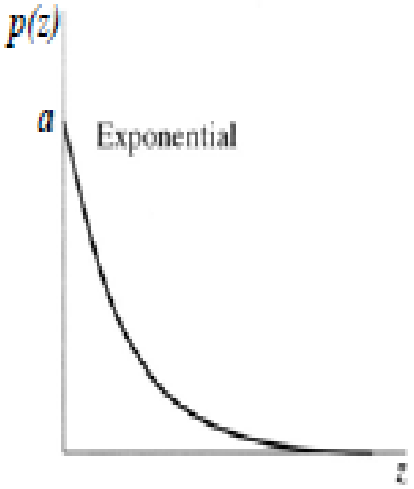
4. Exponential noise is specified as

$$p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Here $a > 0$. The mean and variance are given by

$$\bar{z} = 1/a$$

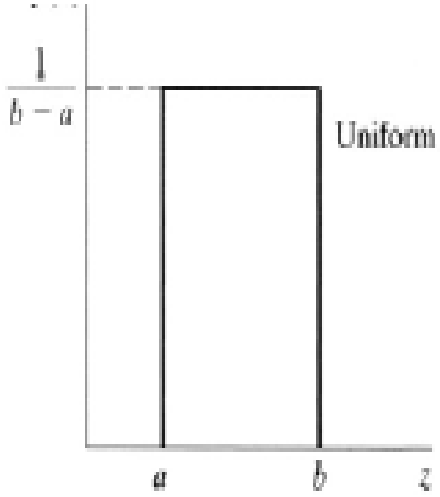
$$\sigma^2 = 1/a^2$$



Exponential pdf is a special case of Erlang pdf with $b = 1$.
Used in laser imaging.

5. Uniform noise is specified as

$$\text{Histogram Uniform} = \begin{cases} \dfrac{1}{b-a} & a \le z \le b \\ 0 & otherwise \end{cases}$$



The mean and variance are given by

$$\bar{z} = \frac{a+b}{2}$$

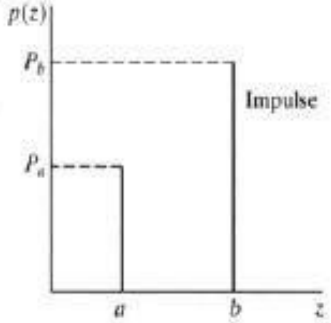
$$\sigma^2 = \frac{(b-a)^2}{12}$$

The gray level values of the noise are evenly distributed across a specific range

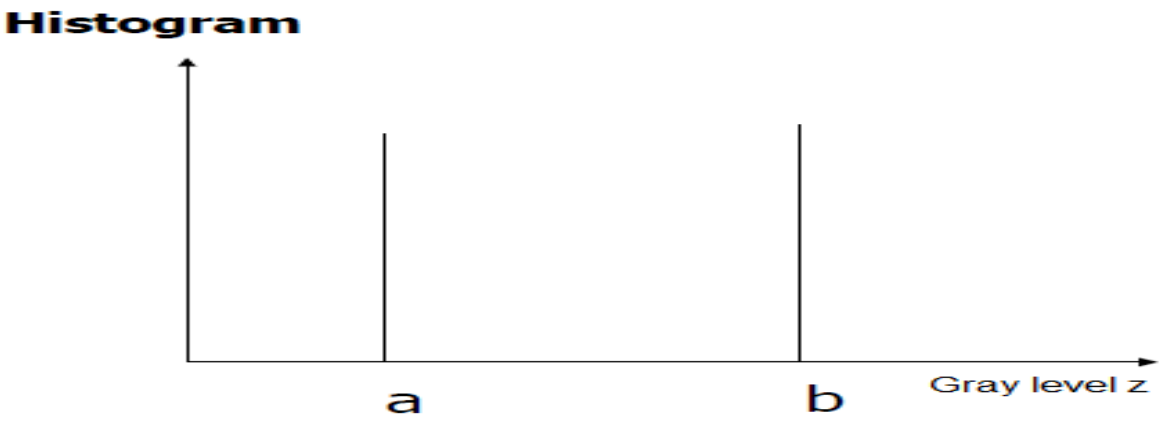
- Quantization noise has an approximately uniform distribution

6. Impulse (salt-and-pepper) noise (bipolar) is specified as

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & otherwise \end{cases}$$



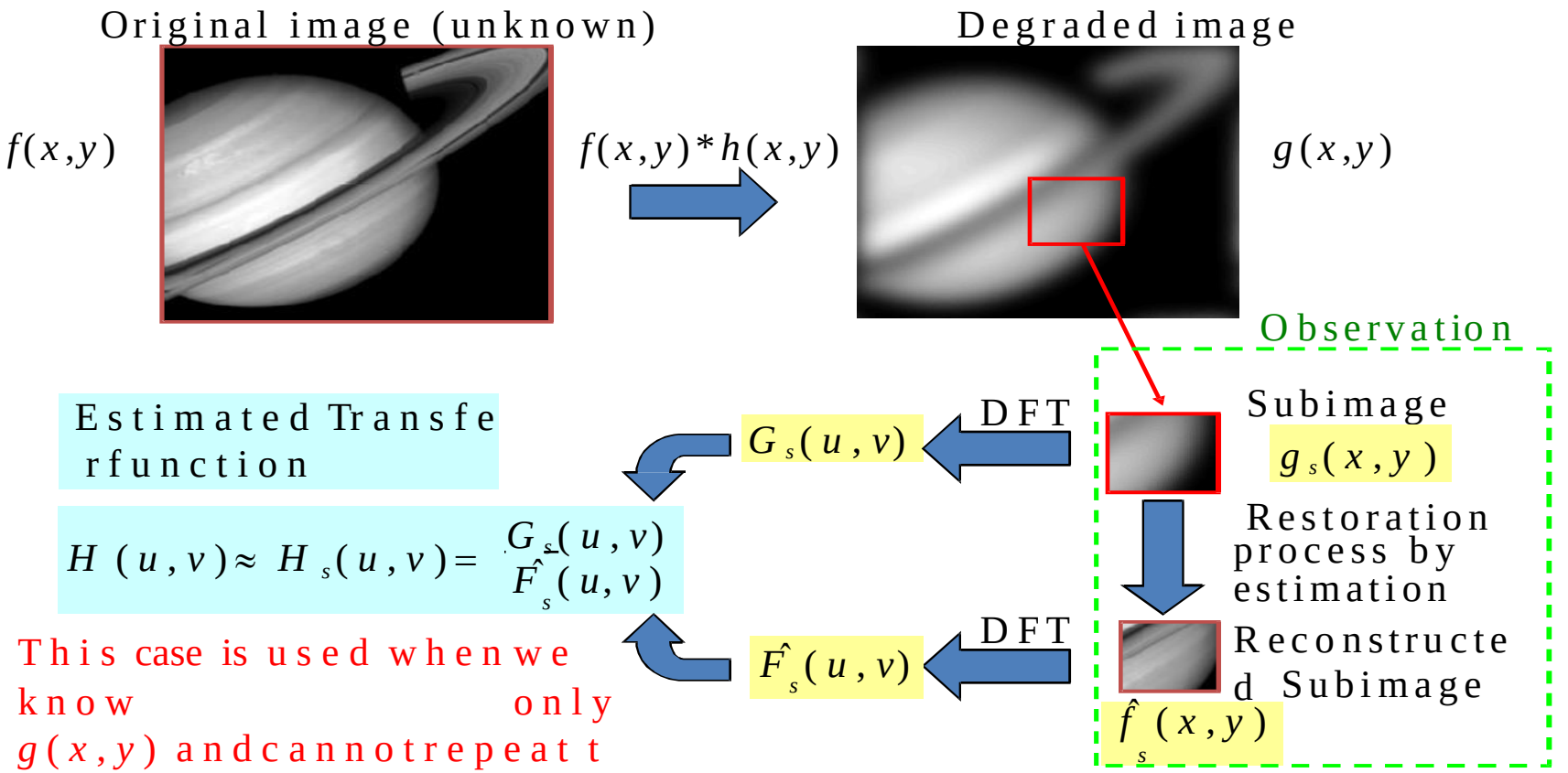
If $b>a$, intensity b will appear as a light dot on the image and a appears as a dark dot. If either P_a or P_b is zero, the noise is called *unipolar*. Frequently, a and b are ***saturated*** values, resulting in positive impulses being white and negative impulses being black. This noise shows up when quick transitions – such as faulty switching – take place.



Three principal methods of estimating the degradation function for Image Restoration: (Blind convolution: because the restored imagewill be only an estimation.)

1.Observation, 2) Experimentation, 3) Mathematical modeling

Estimation by Image Observation



Estimation by Mathematical Modeling:

Sometimes the environmental conditions that causes the degradation can be modeled by mathematical formulation

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}$$

k is a constant that depends on the nature of the Turbulence

If the value of K is large, that means the turbulence is very strong whereas if the value of K is very low, it says that the turbulence is not that strong

If the value of K is large, that means the turbulence is very strong whereas if the value of K is very low, it says that the turbulence is not that strong

Inverse Filtering: (un

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$ co , $\hat{F}(u,v)$ *is the Fourier transform of the restored image*

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

Unknown random function

Must not be very small. Otherwise the noise dominates

even if H (u, v) is known exactly, the perfect reconstruction may not be possible because N (u,v) is not known.

Again if H(u,v) is near zero, N(u,v)/H(u,v) will dominate the

F'(u,v) estimate.

Minimum Mean Square Error (Wiener) Filtering

Least Square Error Filter

Wiener filter (constrained)

Direct Method (Stochastic Regularization)

•Degradation model:

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

•Wiener filter: a statistical approach to seek an estimate $\hat{f}$ that minimizes the statistical function (mean square error):

$$e^2 = E\{ (f - \hat{f})^2 \}$$

Mean Square Error $= MSE = \frac{1}{MN} \sum_{X=0}^{M-1} \sum_{Y=0}^{N-1} \left[f(x,y) - \dot{f}(x,y) \right]$

Restoration with a Wiener filter

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

$\hat{F}(u,v) = W(u,v)\,G(u,v)$

- In frequency domain:

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right] G(u,v)$$

* $H(u, v)$: degradation function
* $|H(u, v)|^2 = H^*(u, v)H(u, v)$ → $H^*(u, v)$: complex conjugate

* Hence, $$\hat{F}(u,v) = \left|\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + R}\right| G(u,v)$$

* $S_f(u, v) = |F(u, v)|^2$: the power spetrum of the undegraded image
* $S_\eta(u, v)/S_f(u, v)$: noise-to-signal power ratio
  → If $S_\eta(u, v) = 0$: inverse filter

UNIT-IV

IMAGE SEGMENTATION & MORPHOLOGICAL IMAGE PROCESSING

IMAGE SEGMENTATION

Input is Image output is features of images

Segmentation is an approach for Feature extraction in an image

Features of Image: Points, lines, edges, corner points, regions

Attributes of features :

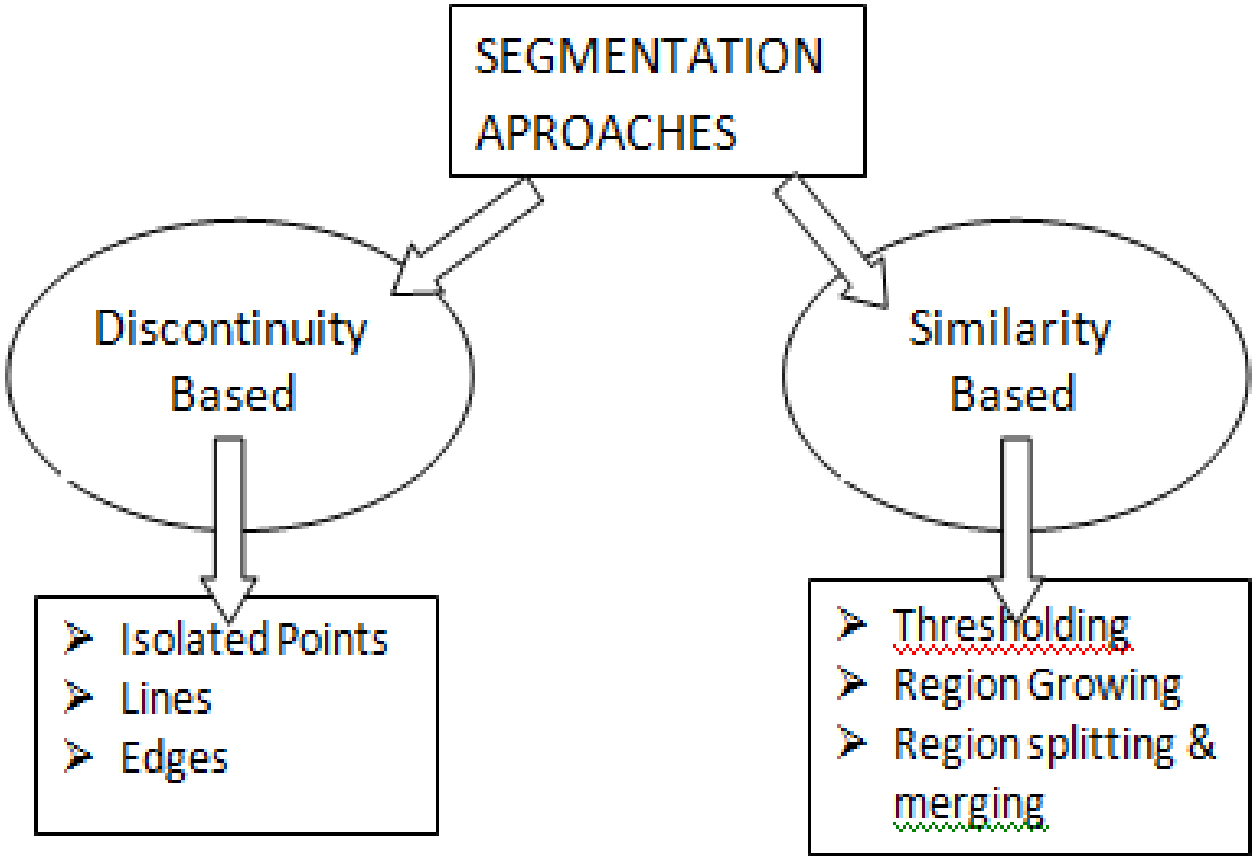
Geometrical (orientation, length, curvature, area, diameter,
perimeter etc

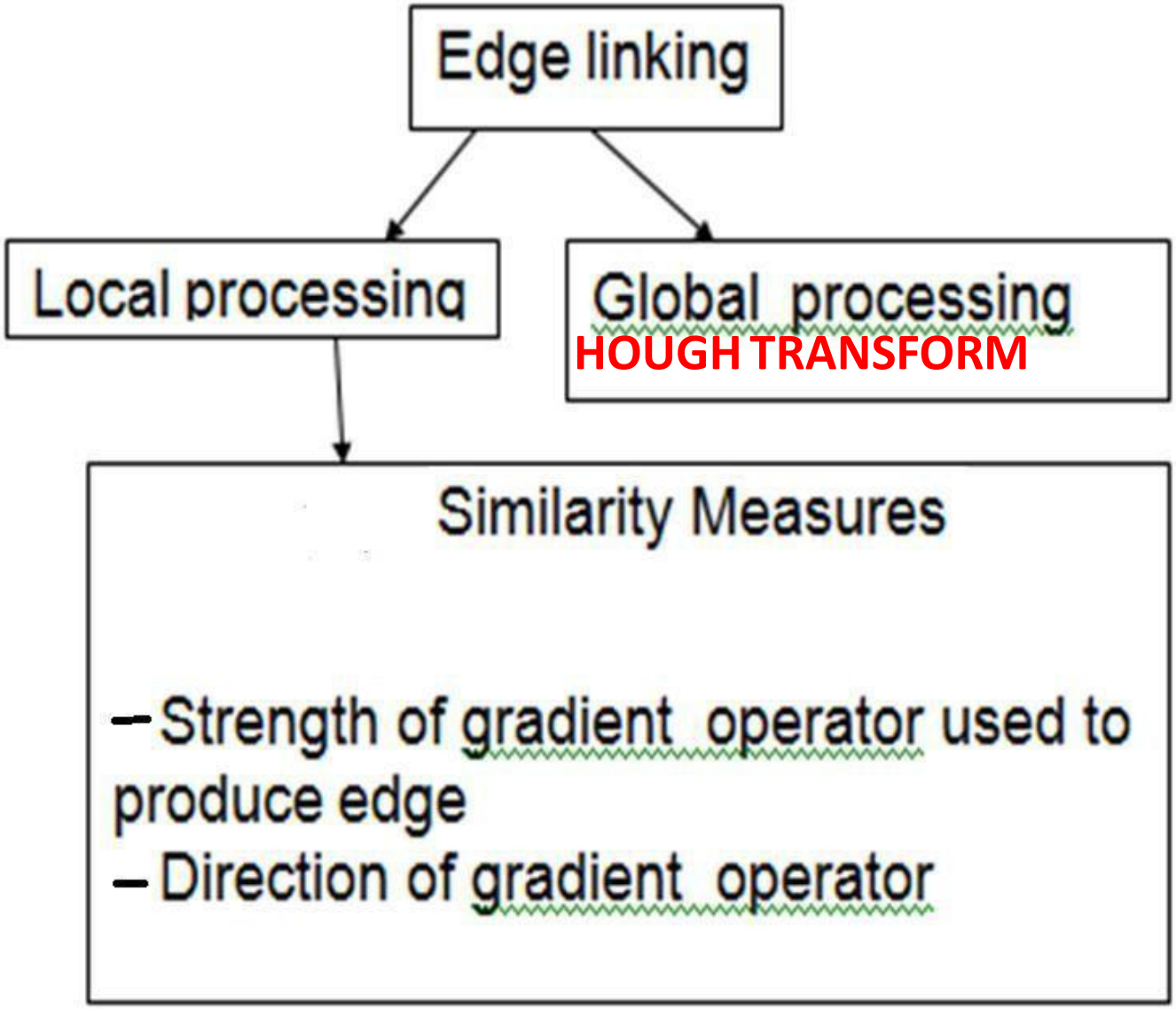
Topological attributes: overlap, adjacency, common end point, parallel, vertical etc

Image segmentation refers to the process of partitioning an image into groups of pixels which are **homogeneous** with respect to some criterion..

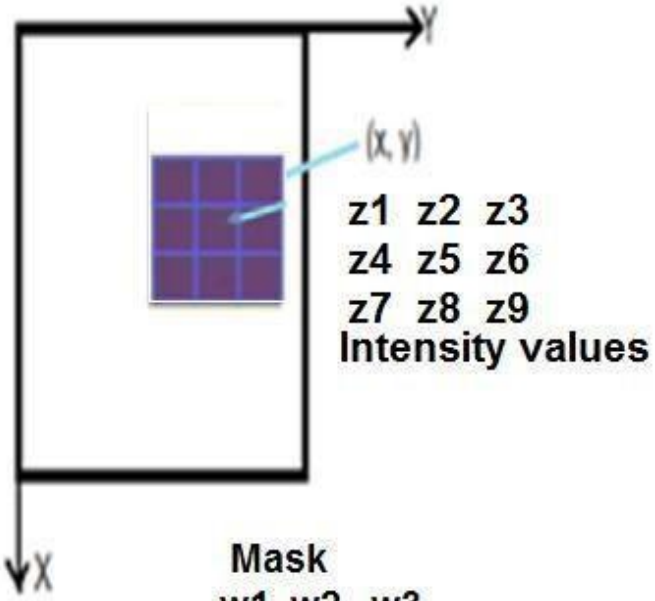
APPROACHES SEGMENTATION

Based on gray level value properties, segmentation is mainly categorized into one of the two categories;




Edge linking
Local processing
Global processing
HOUGH TRANSFORM
Similarity Measures
– Strength of gradient operator used to produce edge
– Direction of gradient operator

Point detection:



$$\begin{matrix} w_{-1,-1} & w_{-1,0} & w_{-1,1} \\ w_{0,-1} & w_{0,0} & w_{0,1} \\ w_{1,-1} & w_{1,0} & w_{1,1} \end{matrix}$$

$$R = \sum_{i=-1}^{1} \sum_{j=-1}^{1} W_{i,j}\, f(x+i, y+j)$$

Or $R = \sum_{k=1}^{9} w_k Z_k$; zk is the intensity at the pixel, wk is the 3x3 mask coefficient. Response of the mask at the center point of the region is R.

Detection of Lines,

Apply all the 4 masks on the

-1	-1	-1		2	-1	-1		-1	2	-1		-1	-1	2
2	2	2		-1	2	-1		-1	2	-1		-1	2	-1
-1	-1	-1		-1	-1	2		-1	2	-1		2	-1	-1

Horizontal	- 45 deg	vertical	45 deg

$|R_i| > |R_j|$, for all values of j not equal to i $\forall_{j \neq i}$

There will be four responses R1, R2, R3, R4.

Suppose that at a particular point,
$|R_1| > |R_j|$, where j=2,3,4 and j≠1

Then that point is on Horizontal line.
Like this, we can decide which point is associated with which line.
Line detection is a low level feature extraction.

Detection of an edge in an image:

What is edge:

An ideal Edge can be defined as a set of connected pixels each of which is located at an orthogonal step transition in gray level

Gray level profile of a horizontal line through the image

Gray level profile of a horizontal line through the image

Calculation of Derivatives of Edges:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}.$$

The magnitude of this vector is given by

$$\begin{aligned} \nabla f &= \operatorname{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}. \end{aligned}$$

$$\nabla f \approx |G_x| + |G_y|.$$

There are various ways in which this first derivative operators can be implemented

Prewitt Edge Operator

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

Horizontal Gx Vertical Gy

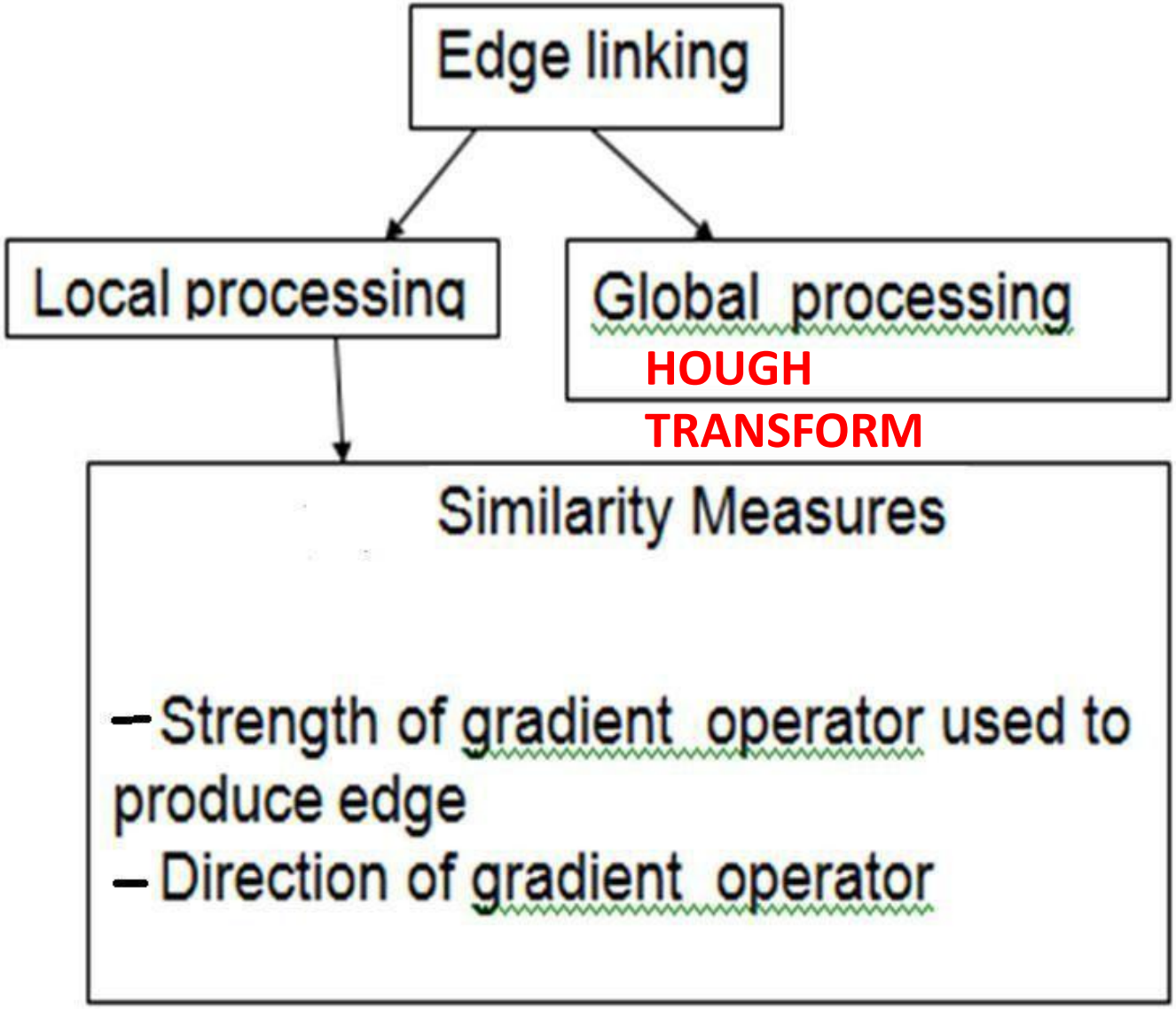
Sobel Edge Operator (noise is taken care)

$$\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix} \qquad \begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$$

Horizontal Gx vertical Gy

The direction of the edge that is the direction of gradient vector f. Direction **$\alpha(x,y) = \tan^{-1}(Gy / Gx)$**

The direction of an edge at a pixel point (x,y) is orthogonal to the direction $\alpha(x,y)$


Edge linking
Local processing
Global processing
HOUGH
TRANSFORM
Similarity Measures
– Strength of gradient operator used to produce edge
– Direction of gradient operator

EDGE LINKING BY LOCAL PROCESSING

A point (x, y) in the image which is already operated by the sobel edge operator. T is threshold

In the edge image take two points x,y and x',y' and to link them

Use similarity measure

first one is the strength of the gradient operator

the direction of the gradient

By sobel edge operator.

$$\nabla f \approx |G_x| + |G_y|.$$

$$\left| \nabla f(x,y) - \nabla f\left(x^{'},y^{'}\right) \right| \leq T$$

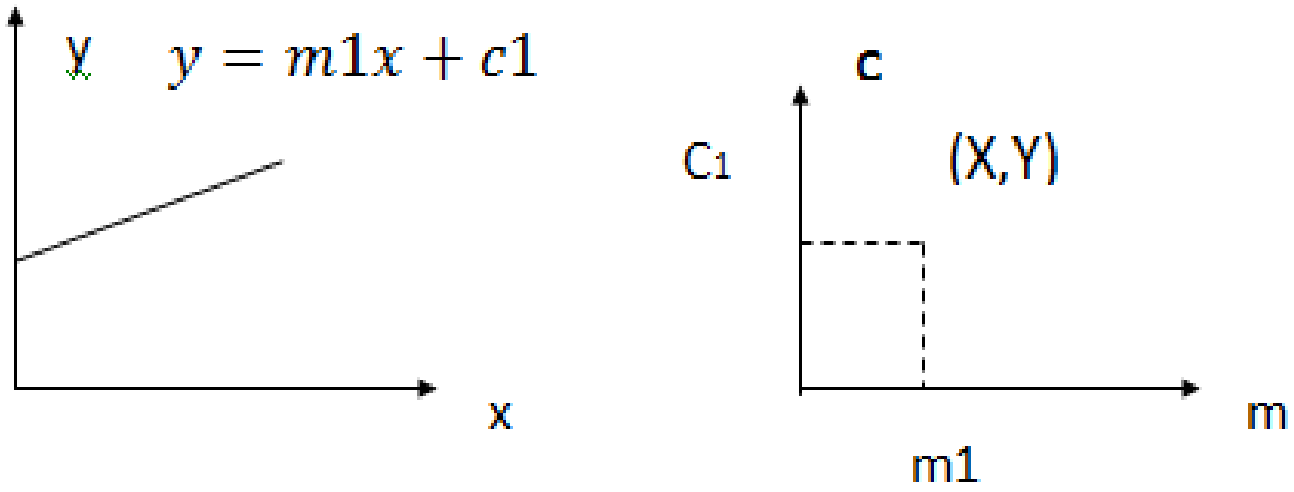
$$\left[\alpha\,(x,y) - \alpha\left(x^{'},y^{'}\right) \right] \leq A$$

These 2 points are similar and those points will be linked together and such operation has to be done for each and do this for every other point in the edge detected image

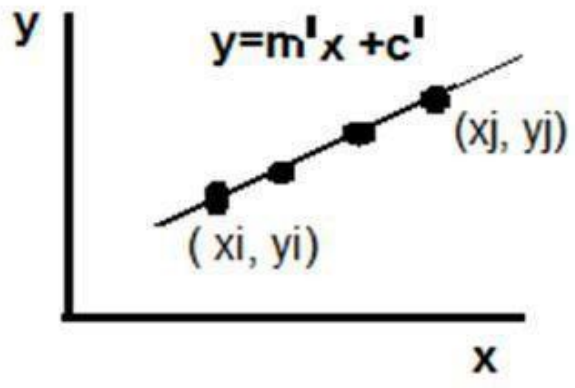
HOUGH TRANSFORM Global processing

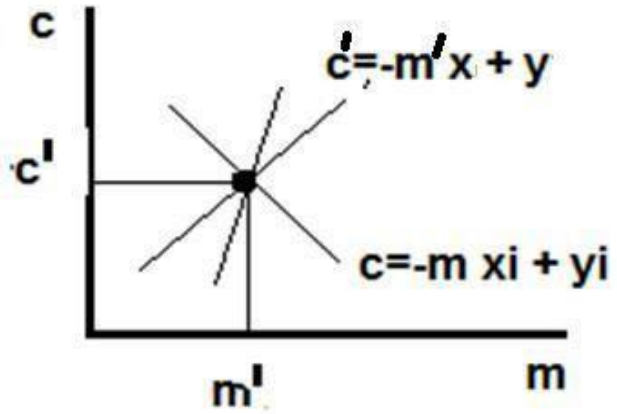
The Hough transform is a mapping from the spatial domain to a parameter space for a particular straight line, the values of m and c will be constant

Spatial domain → Parameter



. Mapping this straight line in the parameter space.





So, we have seen 2 cases

Case one: a straight line in the xy plane is mapped to a point in the mc plane and

Case two: if we have a point in the xy plane that is mapped to a straight line in the mc plane

and this is the basis of the Hough transformation by using which we can link the different edge points which are present in the image domain

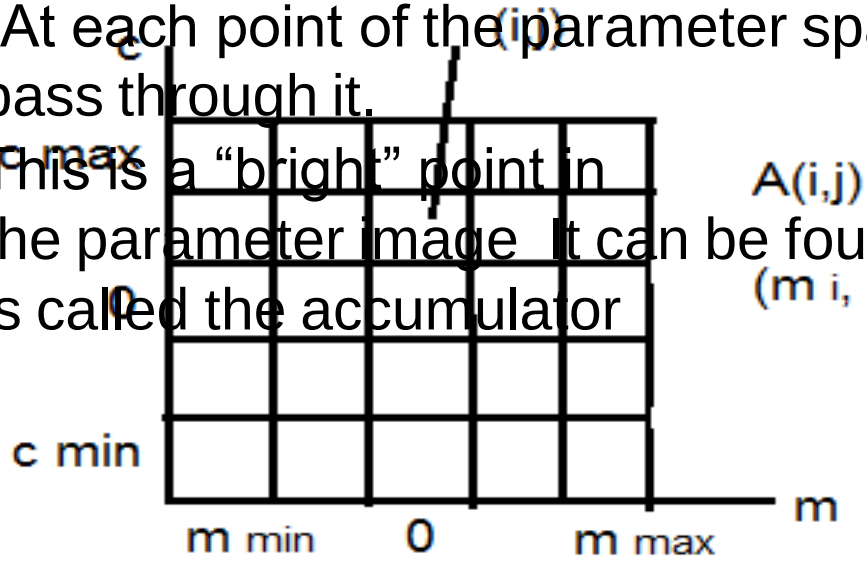
Image Space
Lines
Points
Collinear points

Parameter Space
Points
Lines Intersecting lines

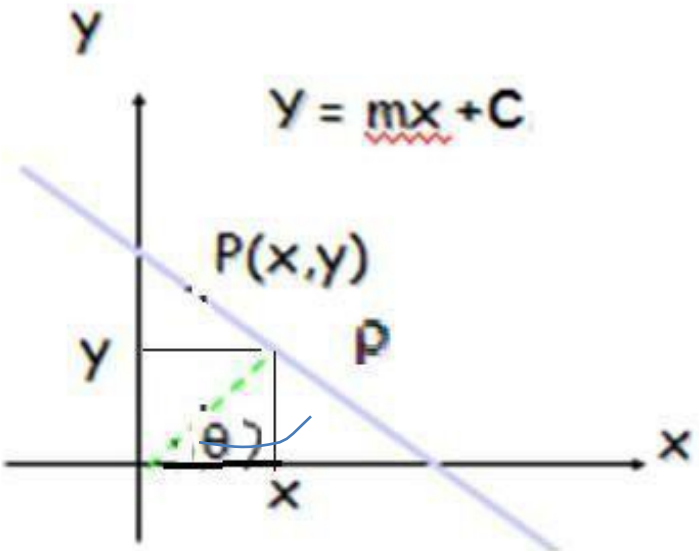
So for **implementation of Hough Transform**, what we have to do is this entire mc space has to be subdivided into a number of accumulator cells.

•At each point of the parameter space, count how many lines pass through it.

This is a "bright" point in the parameter image. It can be found by thresholding. This is called the accumulator



when this straight line tries to be vertical, the slope m tends to be infinity ; to solve this make use of the normal representation of a straight line Use the "Normal" equation of a line:



$\rho = x\cos\theta + y\sin\theta$

θ Is the line orientation

ρ Is the distance between the origin and the line

A Point in Image Space is now represented as a SINUSOID

$\rho = x\cos\theta + y\sin\theta$ Therefore, use (ρ,θ) space

$\rho = x\ \cos\theta + y\sin\theta$

ρ = magnitude

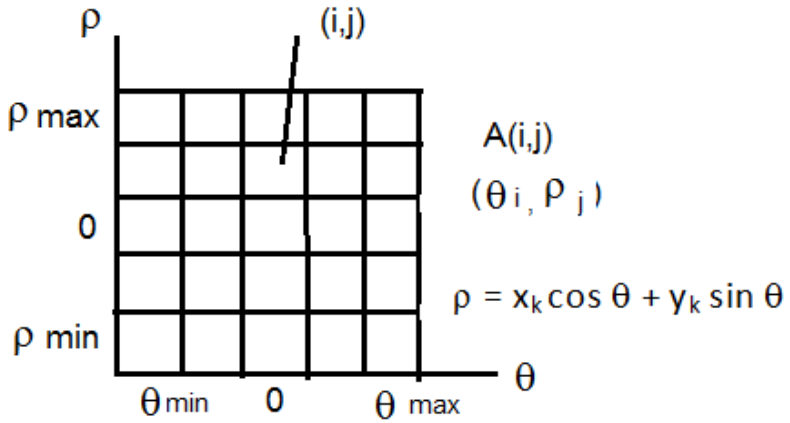
drop a perpendicular from origin to the line

θ = angle perpendicular makes with x-axis

So, unlike in the previous case where the parameters were the slope m and c, now parameters become ρ, and θ.

- Use the parameter space (ρ, θ)
- The new space is FINITE
- $0 < \rho < D$, where D is the image diagonal $\rho = \sqrt{(M^2 + N^2)}$, MxN is image size.
- $0 < \theta < \pi$ (or $\theta = \pm 90$ deg)
- In (ρ,θ) space

point in image space == sinusoid in (ρ,θ) space where sinusoids overlap, accumulator = max maxima still = lines in image space



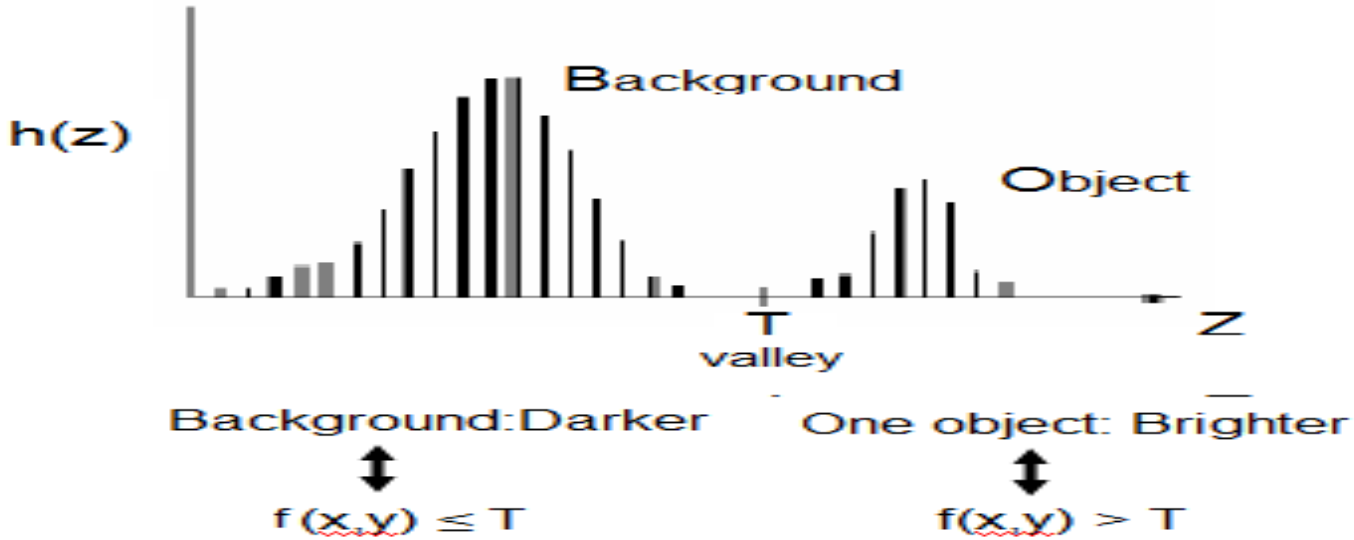
Global Thresholding : a threshold value is selected where the threshold value depends only on the pixel intensities in the image

Dynamic or adaptive thresholding: Threshold depends on pixel value and pixel position. So, the threshold for different pixels in the image will be different.

Optimal thresholding : estimate that how much is the error incorporated if we choose a particular threshold. Then, you choose that value of the threshold where by which your average error will be minimized

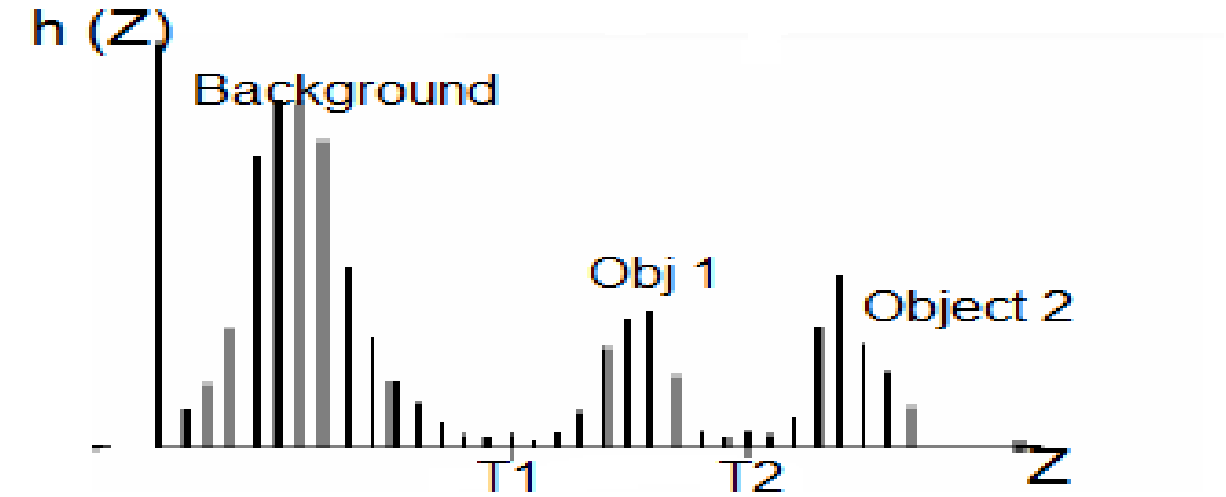
THRESHOLDING

Region based segmentation operations
thresholding
region growing and
the region splitting and merging techniques



So, for such a bimodal histogram, you find that there are two peaks.
Now, the simplest form of the segmentation is, choose a threshold value say T in the valley region
if a pixel at location x,y have the intensity value f (x, y) ≥ T; then we say that
these pixel belongs to object
whereas if f (x, y) < T, then these pixel belongs to the background.

Thresholding In a multi modal



$f(x, y) > T_2 \Rightarrow (x, y) \in$ Object 2

$T_1 < f(x, y) \leq T_2 \Rightarrow (x, y) \in$ Object 1

$f(x, y) < T_1 \Rightarrow (x, y) \in$ Background

So, you will find that the basic aim of this thresholding operation is we want to create a thresholded image g (x, y) which will be a binary image containing pixel values either 0 or 1 depending upon whether the intensity f (x, y) at location (x, y) is greater than T or it is less than or equal to T

This is called global thresholding.

Automatic Thresholding

1. Initial value of Threshold T
2. With this threshold T, Segregate the pixels into two gr2oups G1 and G2
3. Find the mean values of G1 and G2. Let the means be μ1 and μ2
4. Now Choose a new threshod. Find the average of the means

$T_{new} = (\mu 1 + \mu 2)/2$

5. With this new threshold, segregate two groups and repeat the procedure. $|T - T_{new}| > \Delta T'$, back to step. Else stop.

Basic Adaptive Thresholding is
– Divide the image into sub-images and use local threshold s

But, in case of such non uniform illumination, getting a global threshold which will be applicable over the entire image is very very difficult

So, if the scene illumination is non uniform, then a global threshold is not going to give us a good result. So, what we have to do is we have to **subdivide the image into a number of sub regions and find out the threshold value for each of the sub regions** and segment that sub region using this estimated threshold value and here, because your threshold value is position dependent, it depends upon the location of the sub region; so the kind of thresholding that we are applying in this case is an adaptive thresholding.

Basic Global and Local Thresholding

Simple tresholding schemes compare each pixels gray level with a single global threshold. This is referred to as **Global Tresholding.**

If T depends on both f(x,y) and p(x,y) then this is referred to a

Local Thresholding

Adaptive thresholding Local Thresholding

Adaptive Thresholding is
– Divide the image into sub-images and use local thresholds,
Local properties (e.g., statistics) based criteria can be used for adapting the threshold.

Statistically examine the intensity values of the local neighborhood of each pixel. The statistic which is most appropriate depends largely on the input image. Simple and fast functions include the *mean* of the *local* intensity distribution,

T= Mean, T= Median, T= (Max + Min) / 2 You can simulate the effect with the following steps:

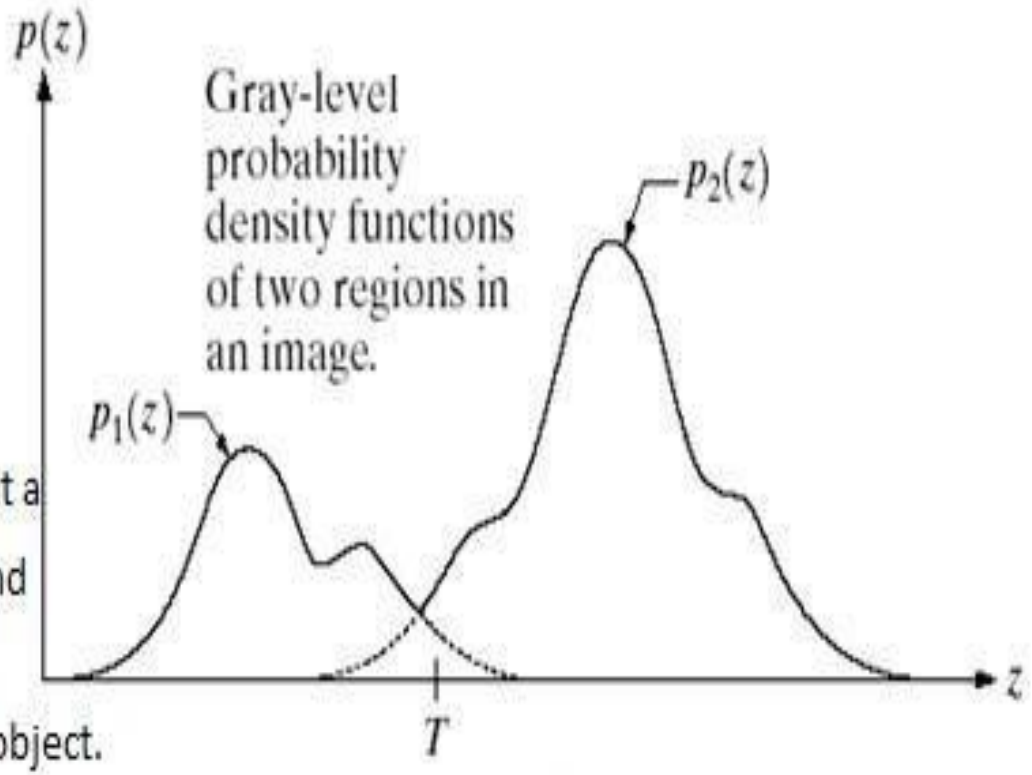
1. Convolve the image with a suitable statistical operator, *i.e.* the *mean* or *median*.
2. Subtract the original from the convolved image.
3. Threshold the difference image with *C*.
4. Invert the thresholded image.

P(z) is histogram

P(z)=P1 P1(z) + P2 P2(z)

P1 + P2 =1

Capital P_1 indicates the probability that a pixel will belong to the background and capital P_2 indicates that indicates the probability that a pixel belongs to an object.

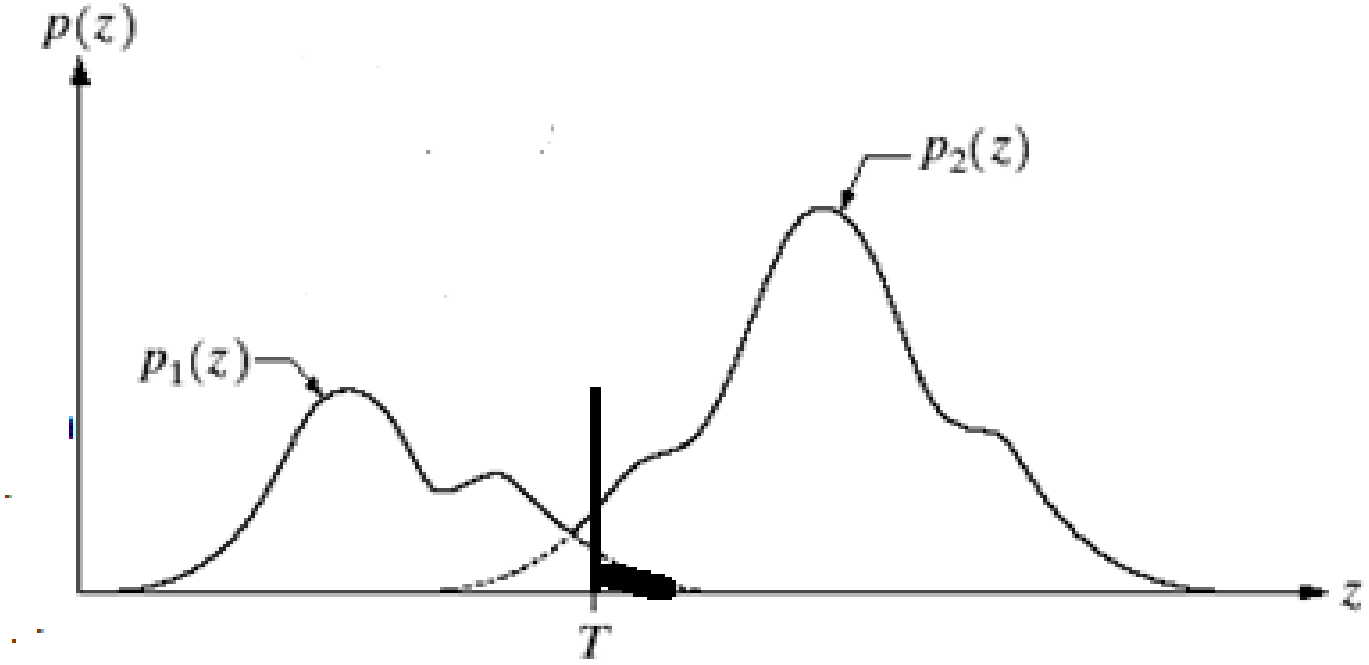


The normalized histogram can be viewed as a probability density function p (z) of this random variable z.

The overall histogram that is p (z) can be represented as the combination of $p_1(z)$ and $p_2(z)$

OTIMAL THRESOLDING

Now, what is our aim in this particular case? Our aim is that we want to determine a threshold T which will minimize the average segmentation error.



f(x,y) > T, (x,y) belongs to Object

Overall probability of error is given by: E (T) = P2 E1(T) + P1 E2(T)

Now, for minimization of this error

$\partial E(T) / \partial T = 0$

By assuming Gaussian probability density function,

$$E(z) = \frac{P1}{(\sqrt{2\pi})\sigma 1} e^{\frac{-(z-\mu 1)^2}{2\sigma 1^2}} + \frac{P2}{(\sqrt{2\pi})\sigma 2} e^{\frac{-(z-\mu 2)^2}{2\sigma 2^2}}$$

The value of T can now be found out as the solution for T is given by, solution of this particular equation

$AT^2 + BT + C = 0$ $A = \sigma_1^2 - \sigma_2^2$

$B = 2 (\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2)$

$C = (\mu_2^2 \sigma_1^2 - \mu_1^2 \sigma_2^2) + 2 \sigma_1^2 \sigma_2^2 \ln \left(\frac{\sigma 2 P1}{\sigma 1 P2} \right)$

$\sigma^2 = \sigma_1^2 = \sigma_2^2$

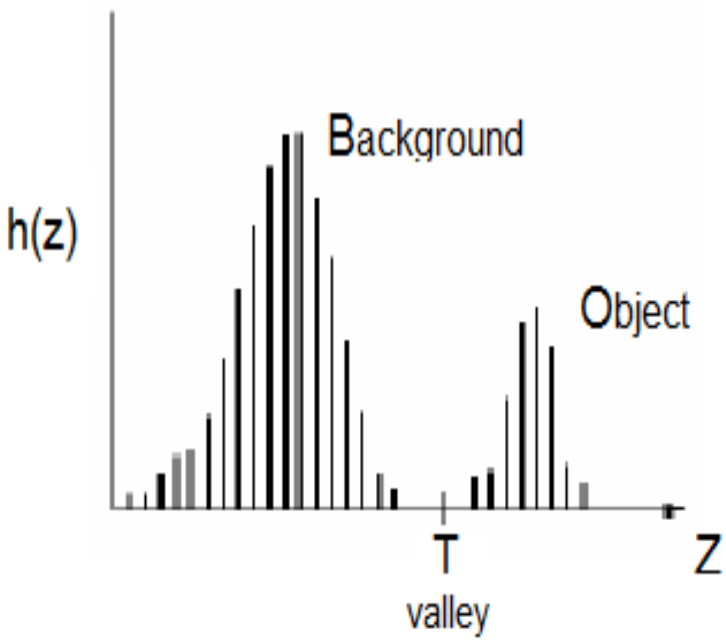
Optimal Threshold is obtained

$T = (\mu_1 + \mu_2)/2 + [\sigma^2 / (\mu_1 - \mu_2)] \ln (P_2/P_1)$

The capital P_1 and capital P_2, they are same; in that case, the value of T will be simply μ_1 plus μ_2 by 2 that is the mean of the average intensities of the foreground region and the background region.

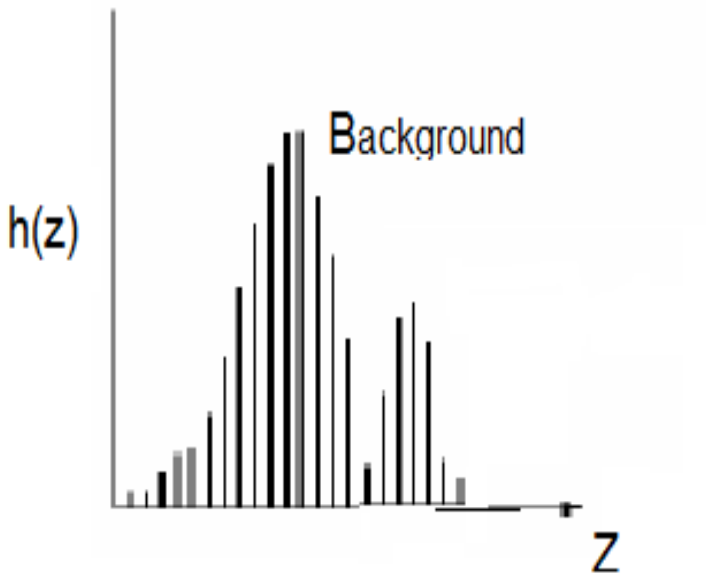
Boundary characteristics for Histogram Thresholding

Use of Boundary Characteristics for Histogram Improvement and Local Thresholding



In this historam, it is easy to find the threshod in the valley

Peaks are separate Peaks are symmetric



So, the bimodal of nature of the histogram is not very visible and is dominated by a single mode by the pixels which belong to the background

Region growing:

starting from this particular pixel, you try to grow the region based on connectivity or based on adjacency and similarity. So, this is what is the region growing based approach

Group pixels from sub-regions to larger regions

– Start from a set of 'seed' pixels and append pixels with similar properties

- Selection of similarity criteria: color, descriptors (gray level + moments)
- Stopping rule

Basic formulation

–Every pixel must be in a region

–Points in a region must be connected

–Regions must be disjoint

–Logical predicate for one region and for distinguishing between regions

Region splitting& merging – Quadtree decomposition

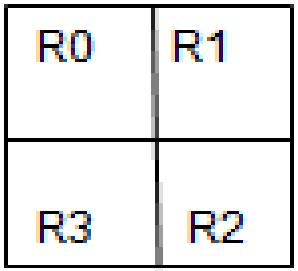
If all the pixels in the image are similar, leave it as it is

Let R denote the Full image.

R

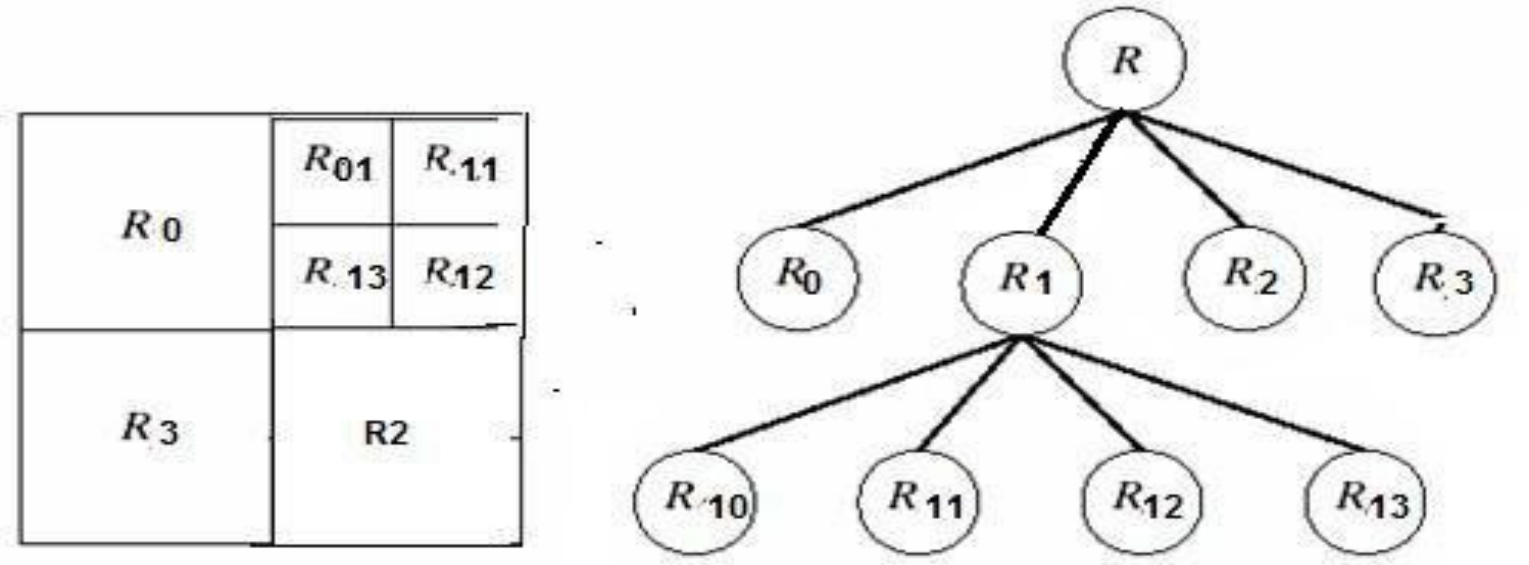
If they are not similar,
then you break this image into quadrants.
make 4 partitions of this image.
Then, check each and every partitionis similar

If it is not similar, again you partition that partic



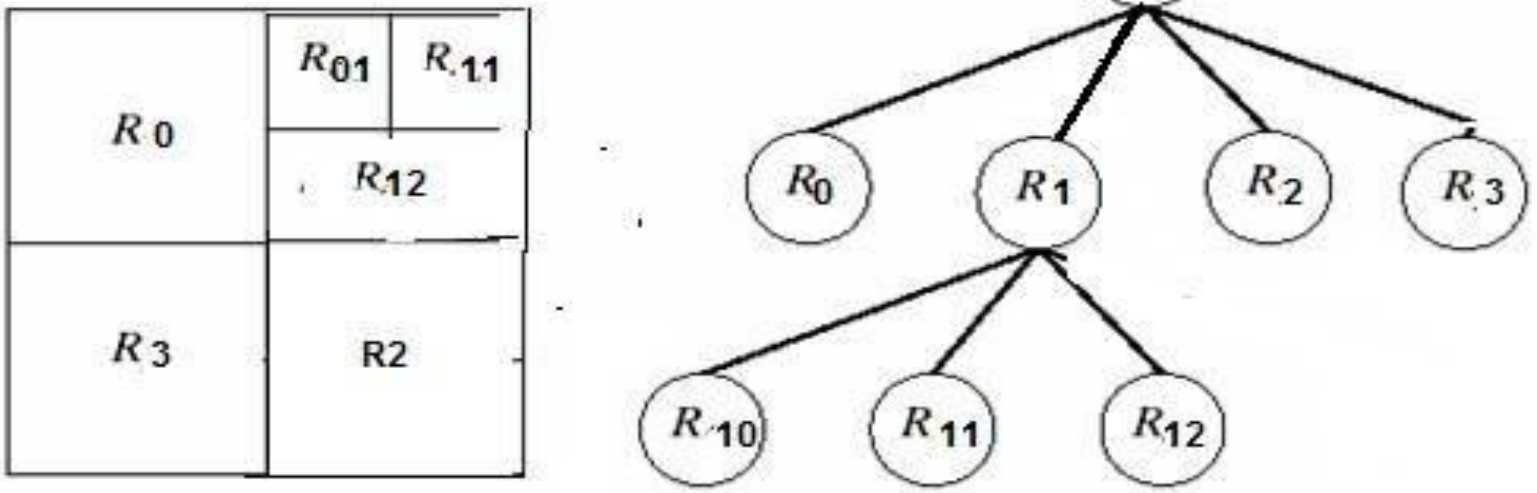
Let us suppose that all the pixels in R are not similar; **(say VARIANACE IS LARGE)**

Now say, this R1 is not uniform,
so partition R1 region again making it R10 R11 R12 R13
and
you go on doing this partitioning until and unless you come to a partition size which is the smallest size permissible or you come to a situation where the partitions have become uniform, or so you cannot partition them anymore.
And in the process of doing this, we have a quad tree representation of the image.



So, in case of quad tree representation, if root node is R, initial partition gives out 4 nodes - R0 R1 R2 and R3. Then R1 gives again R10 R11 R12 and R13. Once such partitioning is completed, then what you do is you try to check all the adjacent partitions to see if they are similar.

If they are similar, you merge them together to form a bigger segment. Say, if R12 and R13 are similar. Merge them.



So, this is the concept of splitting and merging technique for segmentation.

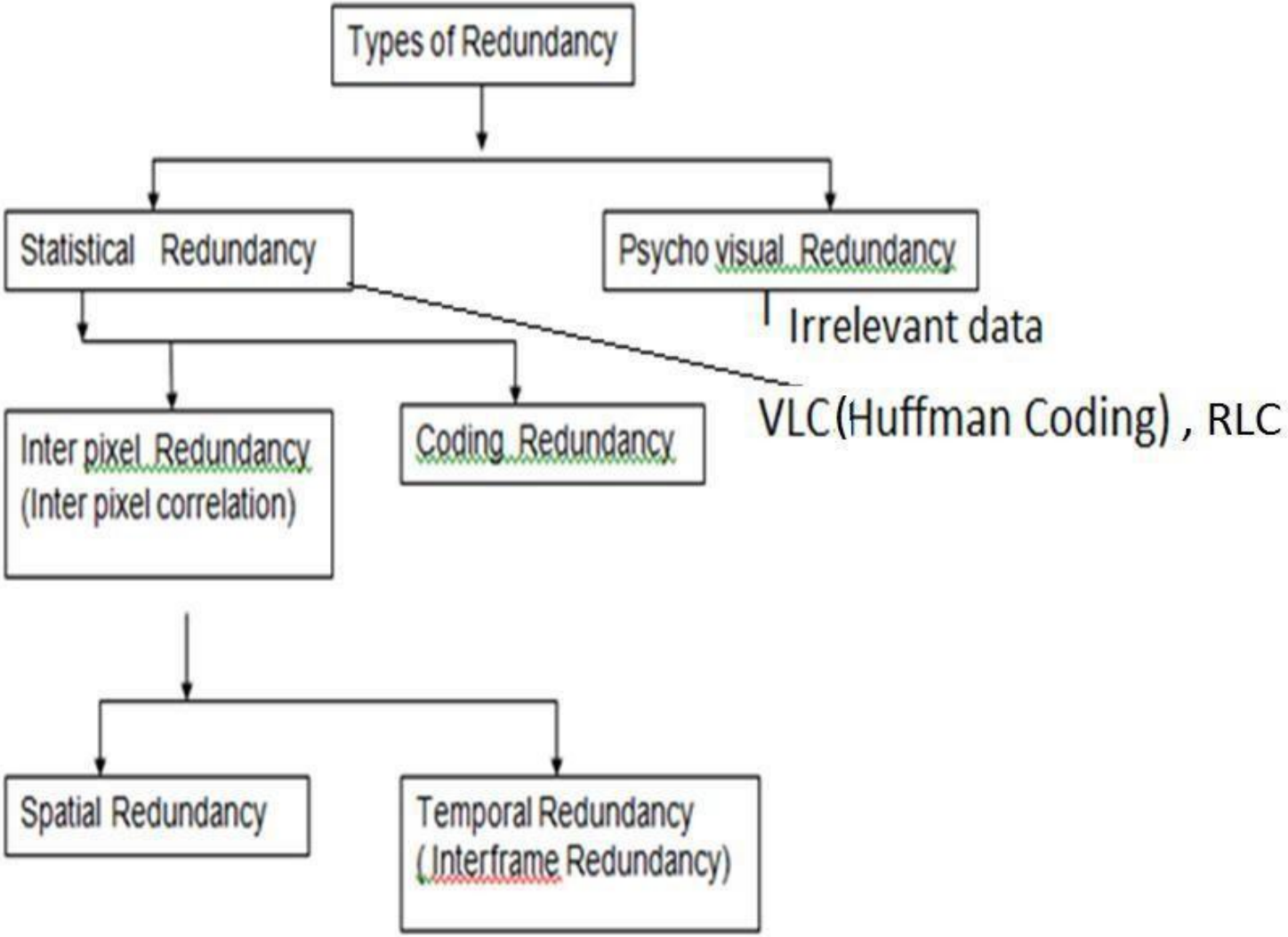
Now at the end, leave it if no more partition is possible ie. reached a minimum partition size or every partition has become uniform;

then look for adjacent partitions which can be combined together to give me a bigger segment.

UNIT-V

IMAGE COMPRESSION

Types of Redundancy



Data redundancy is the central concept in image compression and can be mathematically defined.

Data Redundancy

Because various amount of data can be used to represent the same amount of information, representations that contain irrelevant or repeated information are said to contain redundant data.

- The Relative data redundancy R_D of the first data set, n1, is defined by:

$$R_D = 1 - \frac{1}{C_R}$$

C_R refers to the compression ratio compression ratio (CR) or bits per pixel (bpp) and is defined by:

$$Compression\ Ratio\ C_R = \frac{uncompressed\ file\ size}{Compressed\ file\ size} = \frac{n_1}{n_2}$$

If n1 = n2 , then C_R=1 and R_D=0, indicating that the first representation of the information contains no redundant data.

Coding Redundancy :

•Code: a list of symbols (letters, numbers, bits , bytes etc.)
•Code word: a sequence of symbols used to represent a piece of information or an event (e.g., gray levels).
•Code word length: number of symbols in each code word

Ex: **101** Binary code for 5, Code length 3, symbols 0,1

The gray level histogram of an image can be used in construction of codes to reduce the data used to represent it.

'am of a gray level image where

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0,1,2,\ldots,L-1$$

r_k is the pixel values defined in the interval [0,1] and p_r r(k) is the probability of occurrence of rk. L is the number of gray levels. nk is the number of times that kth gray level appears in the image and n is the total number of pixels (n=MxN)

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	0001	5
$r_6 = 6/7$	0.03	110	3	00001	6
$r_7 = 1$	0.02	111	3	00000	6
8 gray levels		Fixed 3 bit code		Variable length code	

Example of Variable Length Coding

The average number of bit used for fixed 3-bit code:

$$L_{avg} = \sum_{k=0}^{7} l_1(r_k) p_r(r_k) = 3\sum_{k=0}^{7} p_r(r_k) = 3 \times 1 = 3 \; bits$$

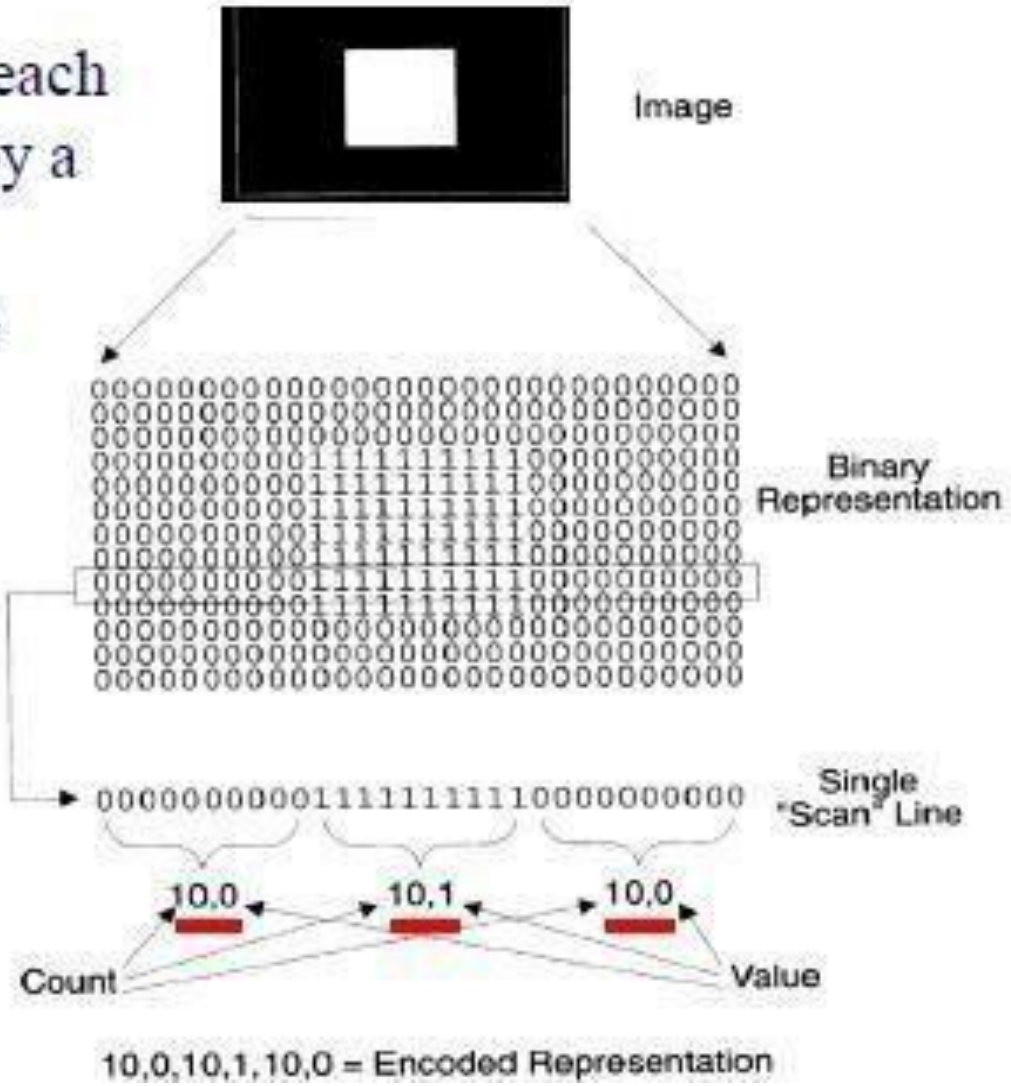
$$L_{avg} = \sum_{k=0}^{7} l_2(r_k) p_r(r_k) = 2(0.19) + 2(0.25) + 2(0.21) +$$
$$3(0.16) + 4(0.08) + 5(0.06) + 6(0.03) + 6(0.02)$$
$$= 2.7 bits$$

- *The compression ratio:* $C_R = \frac{3}{2.7} = 1.11$
- *The relative Data Redundancy:* $R_D = 1 - \frac{1}{1.11} = 0.099 \Rightarrow \sim \%10$

Inter pixel Redundancy or Spatial Redundancy

Pixel redundancy

- In run-length coding, each run R_i is represented by a pair (ri, gi) with
  g_i = gray level of R_i
  r_i = length of R_i



The gray level of a given pixel can be predicted by its neighbors and the difference is used to represent the image; this type of transformation is called **mapping**

Run-length coding can also be employed to utilize inter pixel redundancy in image compression

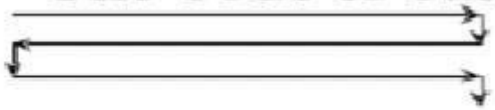
Removing inter pixel redundancy is lossless

Difference coding

$$f(x_i) = \begin{cases} x_i \text{ if } i=0, \\ x_{i+1} - x_i \text{ if } i>0 \end{cases}$$

Ex original:	56	56	56	82	82	82	83	80	80	80	80
code $f(x_i)$:	56	0	0	26	0	0	1	-3	0	0	0

- The code is calculated row by row.



Irrelevant information

One of the simplest ways to compress a set of data is to remove superfluous data For images, information that is ignored by human visual system or is extraneous to the intended use of an image are obvious candidate for omission. The “gray” image, since it appears as a homogeneous field of gray, can be represented by its average intensity alone – a single 8-bit value. Therefore, the compression would be

$$\frac{256 \cdot 256 \cdot 8}{8} = 65,536:1$$

***Psychovisual Redundancy* (EYE CAN RESOLVE 32 GRAY LEVELS ONLY)**

The eye does not respond with equal sensitivity to all visual information. The method used to remove this type of redundancy is called **quantization** which means the mapping of a broad range of input values to a limited number of output values.

Fidelity criteria

$\hat{f}(x,y)$ delity criteria is used to measure information loss and can be divided into two classes.

1)Objective fidelity criteria (math expression is used): Measured mathematically about the amount of error in the reconstructed data.

1)Subjective fidelity criteria: Measured by human observation

Objective fidelity criteria:

When information loss can be expressed as a mathematical function of the input and output of the compression process, $\hat{f}(x,y)$ is based on an objective fidelity criterion. For instance, a ***root-mean- square*** (rms) error between two images.

Let $f(x,y)$ be an input image and be an approximation of $f(x,y)$

resulting from compressing and decompressing the input image. For

$$e_{MSE} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y) - f(x,y)\right]^2$$

- ***The root mean-square-error:***

$$e_{RMSE} = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y) - f(x,y)\right]^2}$$

the *mean-square signal-to-noise ratio* of the output image is defined as

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y) - f(x,y)\right]^2}$$

The rms value of SNR is

$$SNR_{rms} = \sqrt{SNR_{ms}}$$

Subjective criteria:

•Subjective fidelity criteria:

•A Decompressed image is presented to a cross section of viewers and averaging their evaluations.

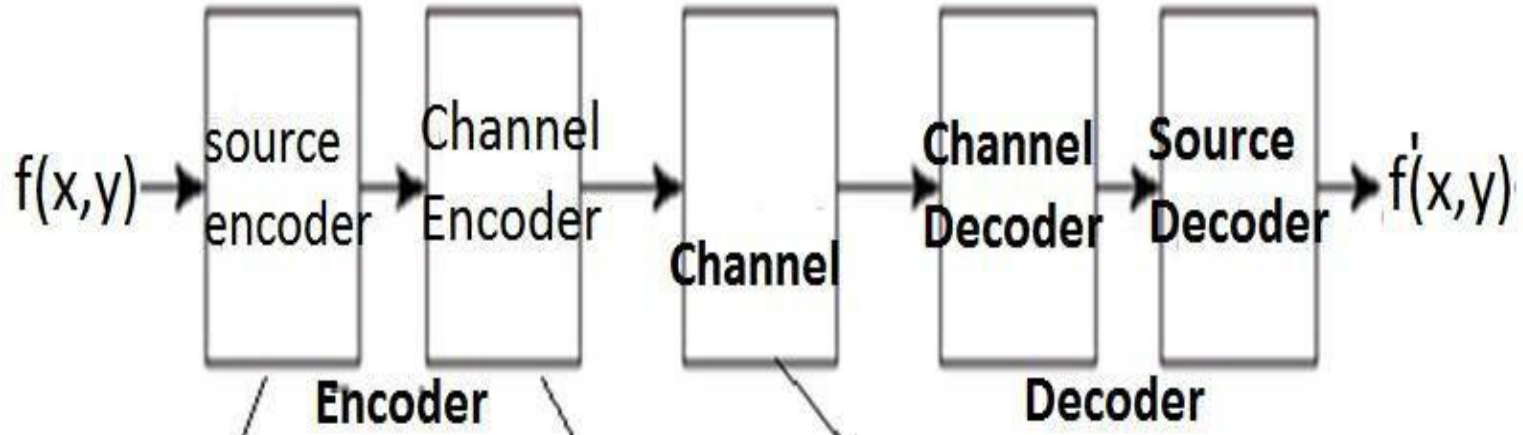
•It can be done by using an absolute rating scale

Or

•By means of side by side comparisons of f(x, y) & f'(x, y).

•Side by Side comparison can be done with a scale such as **{-3, -2, -1, 0, 1, 2, 3}** to represent the subjective valuations {much worse, worse, slightly worse, the same, slightly better, much better} respectively. One possible absolute rating scale: **Excellent, fine, average, poor**

A General Compression System Model

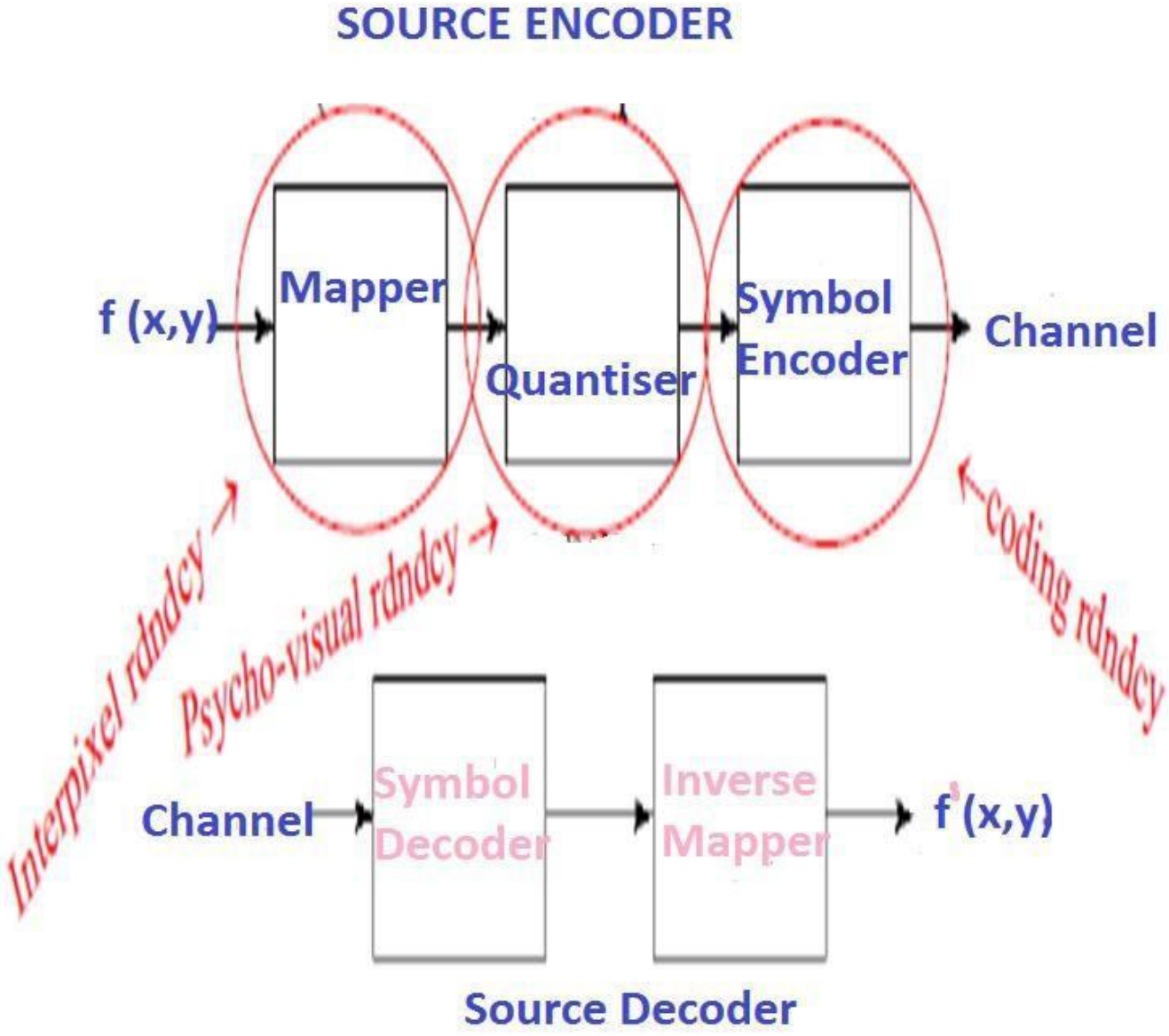


Reduces input image redundancies

Storage or transmission

Reduces impact of channel noise by adding extra bits

Reverse of encoder process


SOURCE ENCODER
f (x,y)
Mapper
Quantiser
Symbol Encoder
Channel
Interpixel rdndcy
Psycho-visual rdndcy
coding rdndcy
Channel
Symbol Decoder
Inverse Mapper
f'(x,y)
Source Decoder

All the three stages are present in every compression model.
If error free compression is needed, Quantizer part will be omitted.
In predict compression system, Quantizer + Mapper = Single Block

Mapper: Transforms the image into array of coefficients reducing inter pixel redundancies. This is a reversible process which is not lossy. Run-length coding is an example of mapping. In video applications, the mapper uses previous (and future)
frames to facilitate removal of temporal redundancy.

•Quantizer: This process reduces the accuracy and hence psycho visual redundancies of a given image is irreversible and therefore lossy.

•Symbol Encoder: Removes coding redundancy by assigning shortest codes for
the most frequently occurring output values.

Huffman Coding:

Source reduction process:

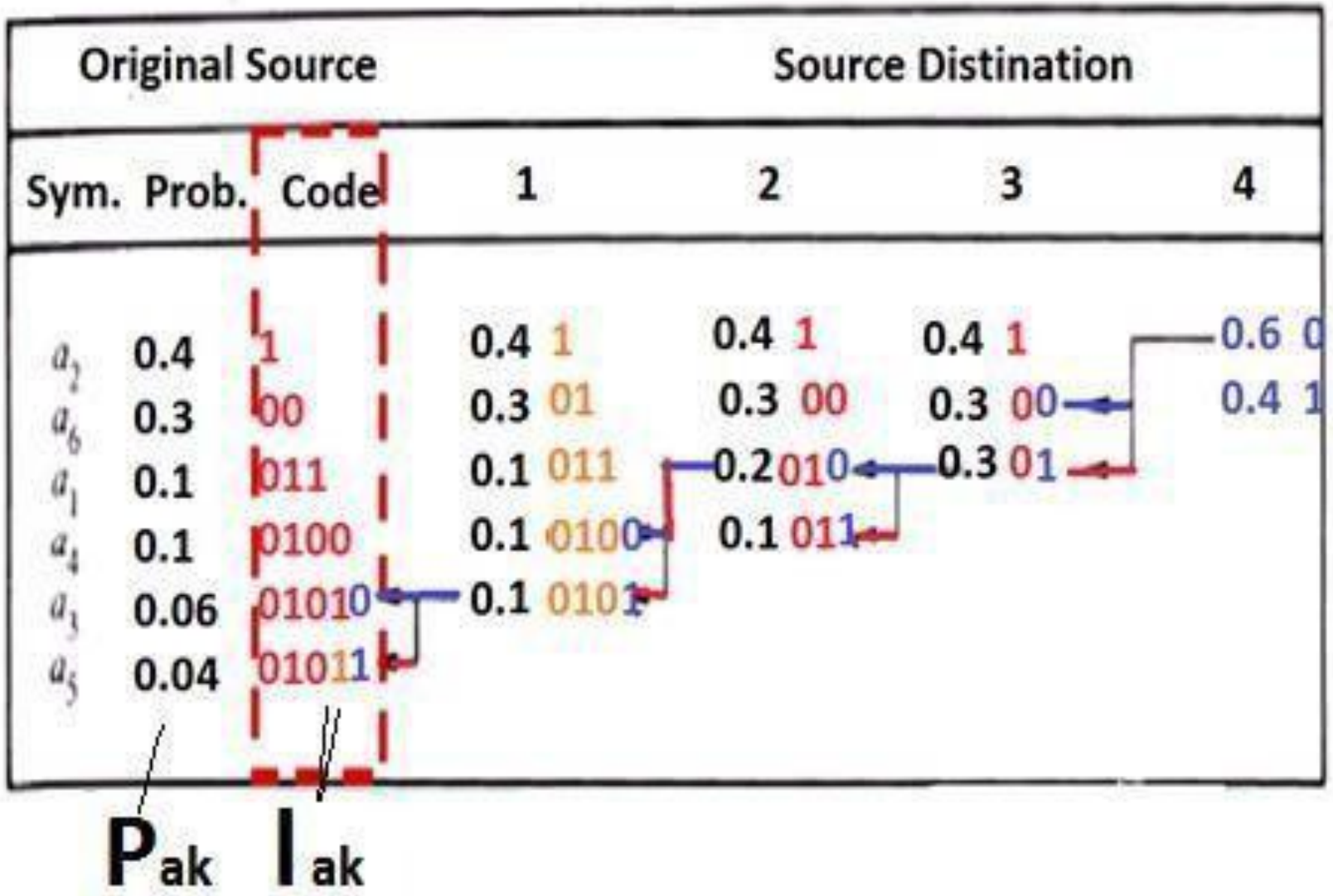
ORIGINAL SOURCE		SOURCE REDUCTION			
Symbol	Probability	1	2	3	4
a2	0.4	0.4	0.4	0.4	0.6
a6	0.3	0.3	0.3	0.3	0.4
a1	0.1	0.1	0.2	0.3	
a4	0.1	0.1	0.1		
a3	0.06	0.1			
a5	0.04				

Entropy of the source or Source Reduction

$$Ez = -\sum_{i=1}^{n} P(ai)\log P(ai)$$

= - [0.4 log (0.4) + 0.3 log (0.3) + 0.1 log (0.1 + 0.1 log (0.1) + 0.06 log (0.06) + 0.04 log (0.04)] = 2.14

Codeword construction process:



Huffman Coding: Note that the shortest codeword (1) is given for the symbol/pixel with the highest probability (a2). The longest codeword (01011) is given for the symbol/pixel with the lowest probability (a5). The average length of the code is given by:

$$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5)$$

$$= 2.2 \; bits / symbol$$

Transform Coding:

(lossy image compression)

In digital images the spatial frequencies are important as they correspond to important image features. High frequencies are a less important part of the images. This method uses a reversible transform (i.e. Fourier, Cosine transform) to map the image into a set of transform coefficients which are then quantized and coded.

Transform Selection: The system is based on discrete 2D transforms. The choice of a transform in a given application depends on the amount of the reconstruction error that can be tolerated and computational resources available.

•Consider a sub image NxN image f(x,y), where the forward discrete transform T(u,v) is given by:

$$T(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)g(x,y,u,v)$$

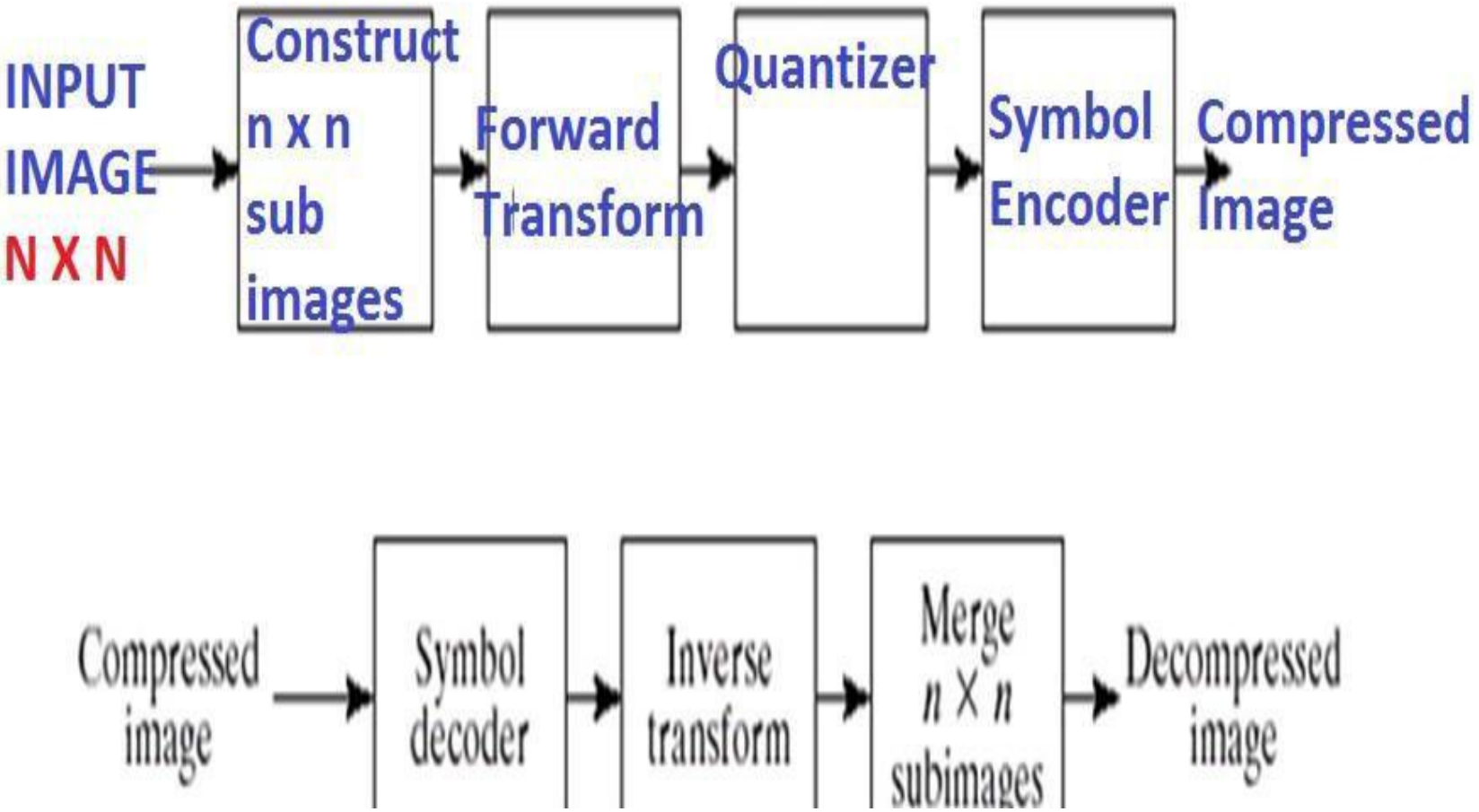
•***For u, v=0,1,2,3,..,N-1***

General scheme

•The transform has to decorrelate the pixels or to compact as much information as possible into the smallest number of transform coefficients

•The quantization selectively eliminates or more coarsely quantizes the less informative coefficients

•Variable-length coding eliminates the remained coding redundancy



A Transform Coding System

DCT

$$c(u,v) = \alpha(u)\,\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left\{ f(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \right\}$$

$$\alpha(u) = \sqrt{\frac{1}{N}}$$

if u=0 ,

$$\alpha(u) = \sqrt{\frac{2}{N}}$$

if u=6,†,....N-1

Same for α (v)

Inverse DCT

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \left\{ \alpha(u)\,\alpha(v)\,c(u,v) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \right\}$$

JPEG Standard

JPEG exploits spatial redundancy

Objective of image compression standards is to enhance the **interoperability and compatibility** among compression systems by different vendors.

JPEG Corresponds to ISO/IEC international standard 10928-1, digital compression and coding of continuous tone still images.

JPEG uses DCT.

JPEG became the Draft International Standard in 1991 and International standard IS in 1992.

JPEG Standard

Different modes such as sequential, progressive and hierarchical modes and options like lossy and lossless modes of the JPEG standards exist.

JPEG supports the following modes of encoding

Sequential : The image is encoded in the order in which it is scanned. Each image component is encoded in a single left-to- right, top-to-bottom scan.

Progressive : The image is encoded in multiple passes. (web browsers). Group DCT coefficients into several spectral bands. Send low-frequency DCT coefficients first AND Send higher- frequency DCT coefficients next

Hierarchical : The image is encoded at multiple resolutions to accommodate different types of displays.

JPEG(Joint Photographic Experts Group)

Applications : color FAX, digital still camera, multimedia computer, internet

JPEG Standard consists of

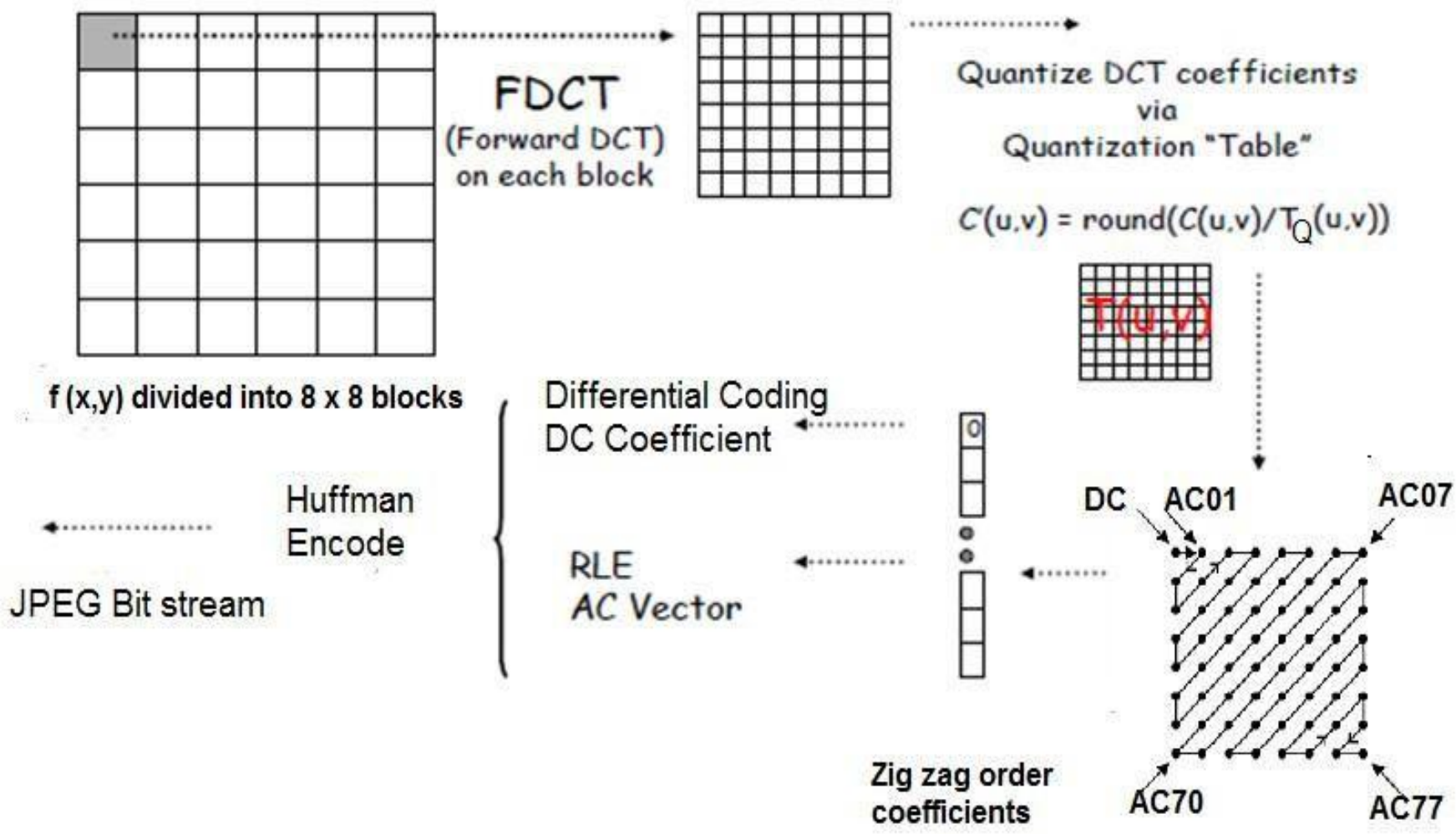
Algorithm: DCT + quantization + variable length coding

Steps in JPEG Compression

- Divide the file into 8 X 8 blocks.
- Apply DCT. Transform the pixel information from the spatial domain to the frequency domain with the Discrete Cosine Transform.
- Each value in the spectrum is divided by the matching value in the quantization table, and the result rounded to the nearest integer.
- Modified spectrum is converted from an 8x8 array into a liner sequence .Look at the resulting coefficients in a zigzag order.

  Do a run-length encoding of the coefficients ordered in this manner.

Follow by Huffman coding.



The coefficient with zero frequency is called DC coefficients, and the remaining 63 coefficients are AC coefficients.

For JPEG decoding, reverse process is applied

Applications of JPEG-2000 and their requirements

- Internet
- Color facsimile
- Printing
- Scanning
- Digital photography
- Remote Sensing
- Mobile
- Medical imagery
- Digital libraries and archives
- E-commerce

Each application area has some requirements which the standard should fulfill.

Improved low bit-rate performance: It should give acceptable quality below 0.25 bpp. Networked image delivery and remote sensing applications have this requirements.

Progressive transmission: The standard should allow progressive transmission that allows images to be reconstructed with increasing pixel accuracy and resolution.

Region of Interest Coding: It should preferentially allocate more bits to the regions of interest (ROIs) as compared to the non-ROI ones.

Content based description: Finding the desired image from a large archive of images is a challenging task. This has applications in medical images, forensic, digital libraries etc. These issues are being addressed by MPEG-7.

- Image Security: Digital images can be protected using watermarking, labeling, stamping, encryption etc.

DWT for Image Compression

2D discrete wavelet transform (1D DWT applied alternatively to horizontal and vertical direction line by line) converts images into "sub-bands" Upper left is the DC coefficient
Lower right are higher frequency sub-bands.

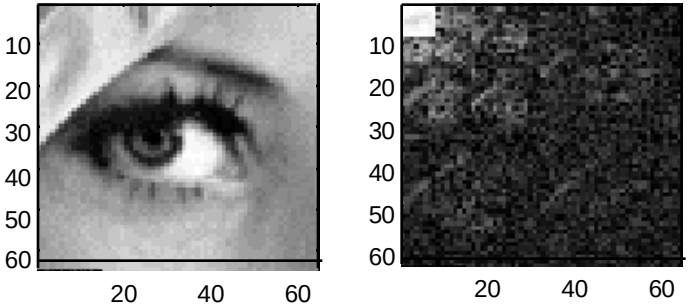


Image decomposition Scale 1

4 subbands : LL1, LH1,HL1,HH1

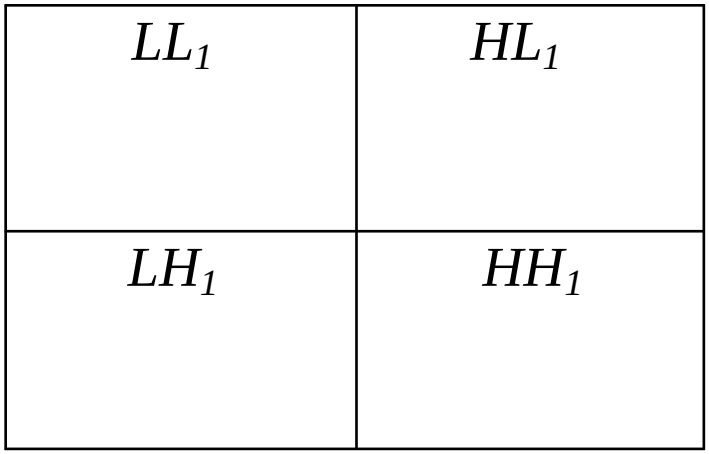


Image decomposition Scale 2

4subbands : LL2, LH2,HL2,HH2

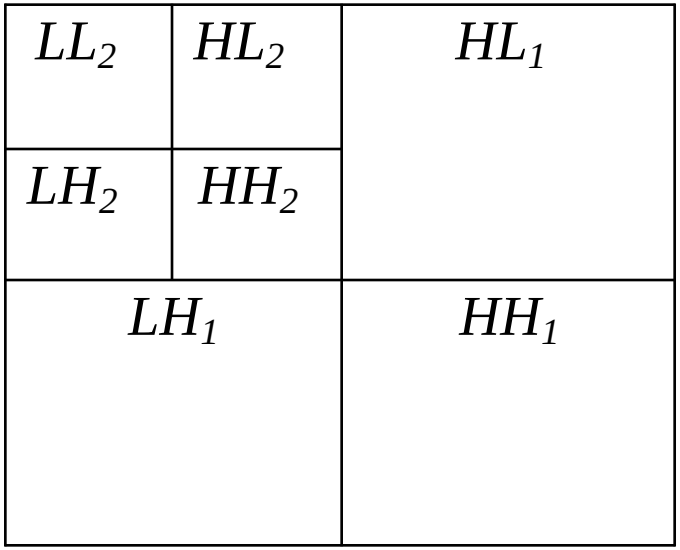
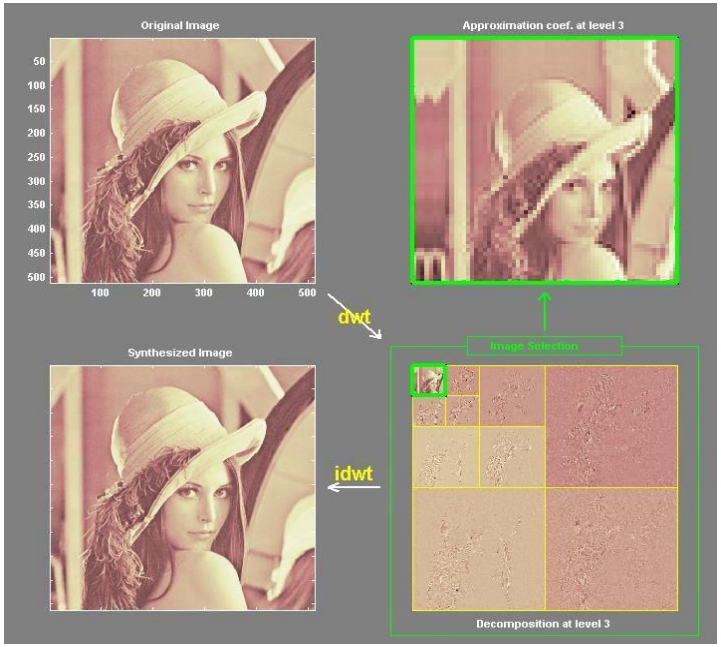
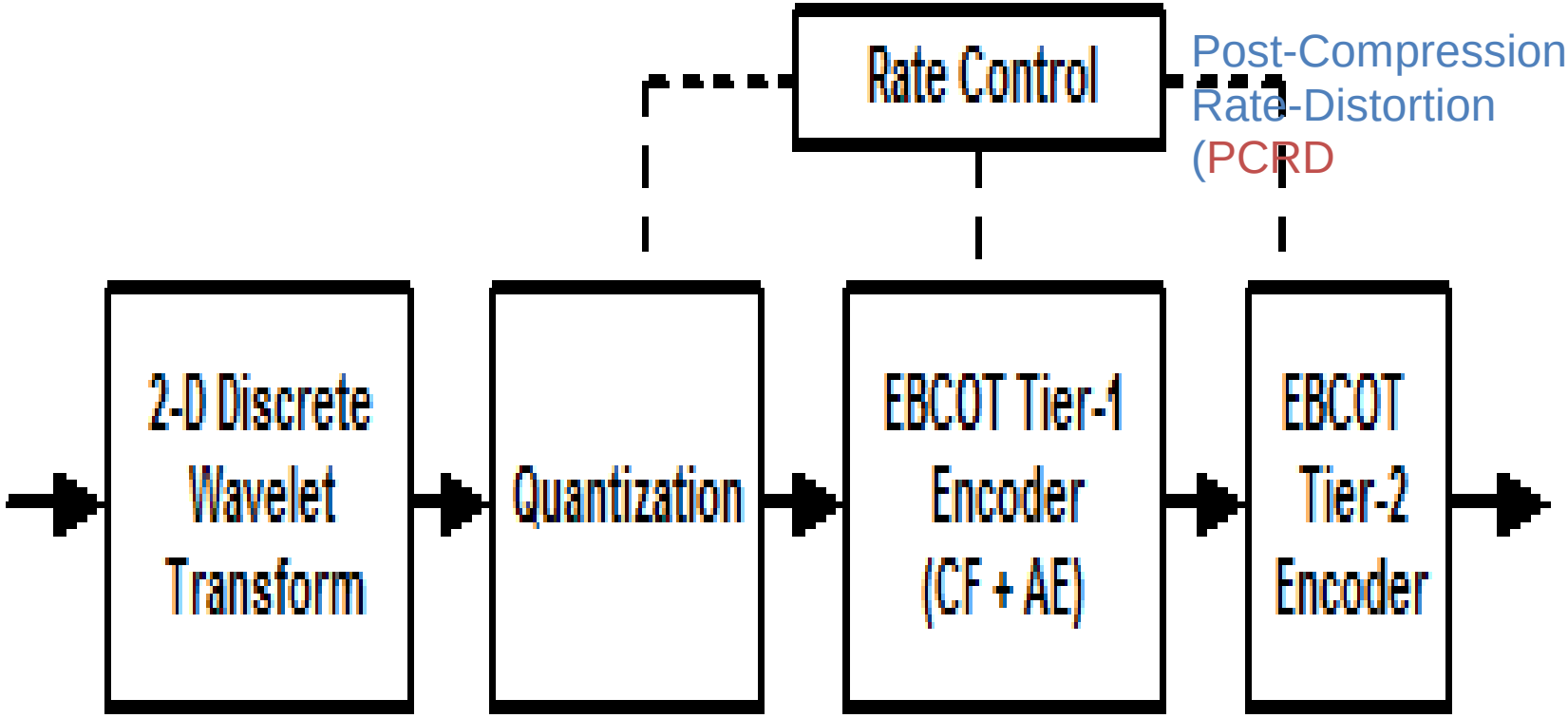


Image Decomposition





transform **quantize** **coding**

J2K uses 2-D Discrete Wavelet Transformation (DWT)

Embedded Block Coding with Optimized Truncation of bit-stream (EBCOT), which can be applied to wavelet packets and which offers both resolution scalability and SNR scalability.

Each sub band is partitioned into small non-overlapping block of samples, known as *code blocks. EBCOT generates an embedded bit-stream for each code block. The bit-stream associated with each code block may be truncated to any of a collection of rate-distortion optimized truncation points*

Steps in JPEG2000

Tiling:

Smaller non-overlapping blocks of image are known as tiles The image is split into *tiles*, rectangular regions of the image. Tiles can be any size. Dividing the image into tiles is advantageous in that the decoder will need less memory to decode the image and it can opt to decode only selected tiles to achieve a partial decoding of the image.

Wavelet Transform: Either CDF 9/7 or CDF 5/3 bi-orthogonal wavelet transform.

Quantization: Scalar quantization

All operations, such as component mixing, DWT, quantization and entropy coding are therefore done independently for each tile.

'Compress Once, Decompress Many Ways"

A Single Original Codestream

By resolutions

By layers

Region of Interest

MPEG1 MOVING PICTURE EXPERT GROUP

MPEG exploits temporal redundancy. Prediction based.

Compare each frame of a sequence with its predecessor and only pixels that have changed are updated,

MPEG-1 standard is for storing and retrieving video information on digital storage media.

MPEG-2 standard is to support digital video broadcasting, HDTV H.261 standard for telecommunication applications systems.

MPEG1 COMPRESSION ALGORITHM: MPEG Digital Video Technology

Temporal compression algorithm: Temporal compression algorithm relies on similarity between successive pictures using prediction in motion compensation Spatial compression algorithm: relies upon redundancy with in small areas of a picture and is based around the DCT transform, quantization an entropy coding techniques.

MPEG-1 was up to 1.5 Mbit/s. MPEG-2 typically over 4MBit/s but can be up to 80 Mbit/s.

MPEG-1(ISO/IEC 11172) and MPEG-2(ISO/IEC 13818) , MPEG-4(ISO/IEC 14496)