BMS **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**
YELAHANKA – BANGALORE - 64
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Semester: VII ECE

Course: DIGITAL IMAGE PROCESSING

Subject Code: 17EC72

Academic Year: 2020-21Odd Sem

Course coordinators:

Dr.Surekha R. Gondkar,

Prof. Mamatha K. R.

Prof. Shilpa Hiremath

CONTENTS

Digital Image Processing

Module 1

Syllabus: What is DIP?, Origins of DIP, Examples of Fields that use DIP. Fundamental Steps in DIP. Components of an IP system. Elements of Visual Perception. Image Sensing & Acquisition, Image Sampling & Quantization Some basic relationships b/n pixels, Linear & Nonlinear operations.

What is Digital Image Processing? (DIP)

An image may be defined as a 2-D function f(x, y) where x & y are spatial (plane) coordinates, & 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 & intensity values of f are all finite & discrete, we call the image a Digital Image.

DIP:- Processing of digital images by means of a digital computer.

The elements of digital image— pixels, pels or picture elements or image elements.

Pixel is widely used

1) Image Processing → I/p & o/p are images.

2) Image Analysis. (Image Understanding).

3) Computer vision.

2. The origins of DIP :-

One of the first app^ns of digital images was in the newspaper industry, when pictures were first sent by submarine cable b/n London & NewYork.

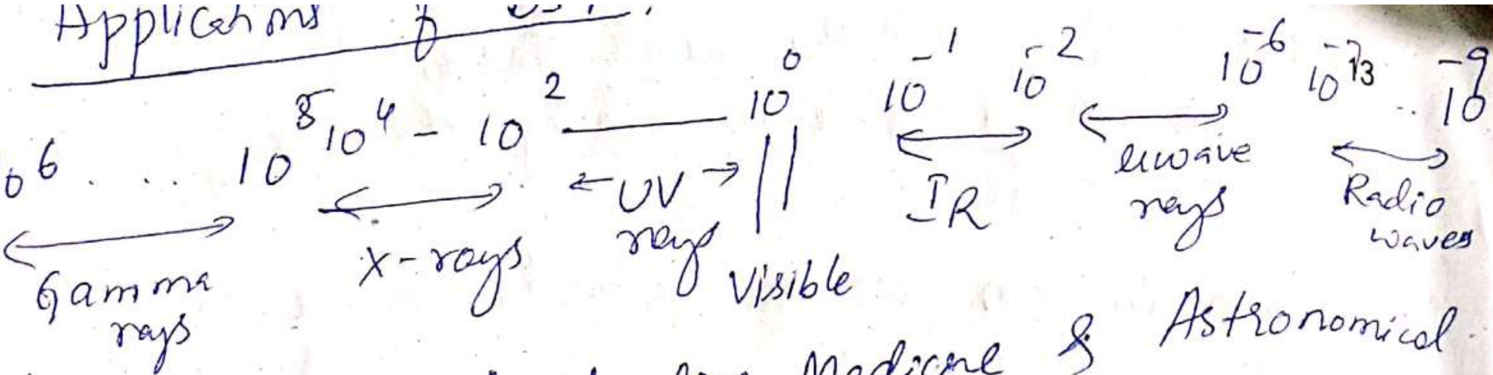
Introduction of the Bartlane cable picture transmission system in the early 1920s reduced the time required to transport a picture across the Atlantic from more than a week to less than 3hrs.

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

Some of the initial problems in improving the visual quality of these early digital pictures were related to the selection of printing procedures & the distribution of intensity levels..

Key advances made in the field of computers like, transistors, ICs, s/w like Cobol, Fortran, μP & VLSI etc helped the advancement in DIP.

3. Applications of



1) Gamma-Ray Imaging: Nuclear Medicine & Astronomical Observations.

Complete bone scan — bone pathology < Infections, Tumors.

→ PET (Positron Emission Tomography) (like to X-Ray tomography)

) X-ray Imaging: Medical diagnostics, industry

Angiography → major appn – images of blood vessels (Angiogram)

CAT – Computerized Axial Tomography.

) Imaging in UV band: Lithography, industrial inspection, microscopy, Lasers, biological imaging & astronomical observations.

Imaging in the Visible & Infrared Bands: Light microscopy, astronomy, remote sensing, industry & law enforcement.

Light microscopy: Pharmaceuticals & microinspection to materials characterization.

Remote sensing: Satellite images – monitoring environmental conditions on the planet, weather observation & prediction also are major appn of multispectral imaging from satellites.

Automated visual inspection of manufactured goods Pills, unfilled bottles, burned flakes, damaged lens etc 14 before packing, vehicle no. reading etc for traffic monitoring.

Imaging in the μwave band: Radar → waves can penetrate thro' clouds, ice, dry sand etc.

Imaging in the Radio band: Medicine & astrono

MRI (Magnetic Resonance Imaging) → Medicine.
places a patient in a powerful magnet & passes radiowaves thro' his/her body in short pulses.

Examples in which other imaging modalities are used

Acoustic imaging

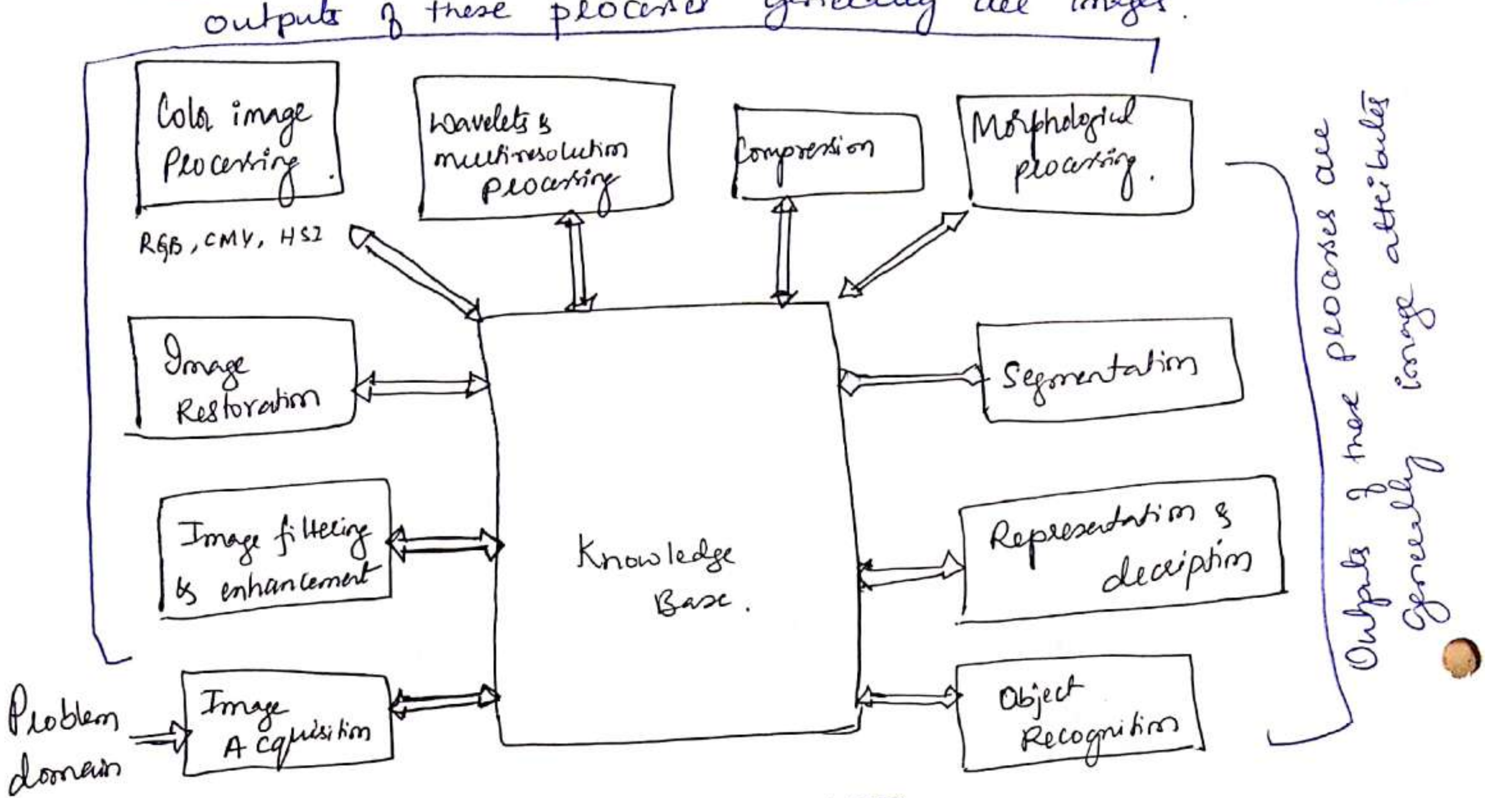
electron microscopy

synthetic imaging (computer-generated)

Mineral & oil exploration

Fundamental Steps in Digital Image Processing

②



Fundamental Steps in DIP

① Image acquisition :- first process in above fig. (DIP) It involves Image collection, preprocessing (such as Scaling)

② Image filtering & enhancement : It is a process of manipulating an image so that more suitable for a specific app^n.

specific → Technique which is suitable for x-Ray enhancement is not suitable for satellite image enhancement.

③ Image Restoration: Improves the appearance of an image based on mathematical model.

Restoration → objective.

Enhancement → Subjective

④ Color Image Processing : This area is gaining importance because of significant increase in the use of digital images over the Internet.

⑤ Wavelets (& multiresolution processing) : Representing images in various degrees of resolution.
images are subdivided into similar regions.

⑥ Compression : Reduces the storage required to save an image or bandwidth required to transmit it.
Eg:- ZIP, JPEG

⑦ Morphological processing : Tools for extracting image components that are useful in the representation & description of shape.

⑧ Segmentation : Procedure partitions an image into its constituent parts or objects.
Autonomous Segmentation — most imp. tasks in DIP.

⑨ Representation & description: O/P of segmentation stage. ie raw pixel data constituting either the boundary of a region or all the points in the region itself.
Description: also called as feature selection. deals with extracting attributes that result in some quantitative information of interest.

⑩ Recognition : assigns a label to an object based on its descriptors.

Imageprocessing is a method to convert an image into digital form & perform some operations on it, in order to get an enhanced image or to extract some useful info from it.

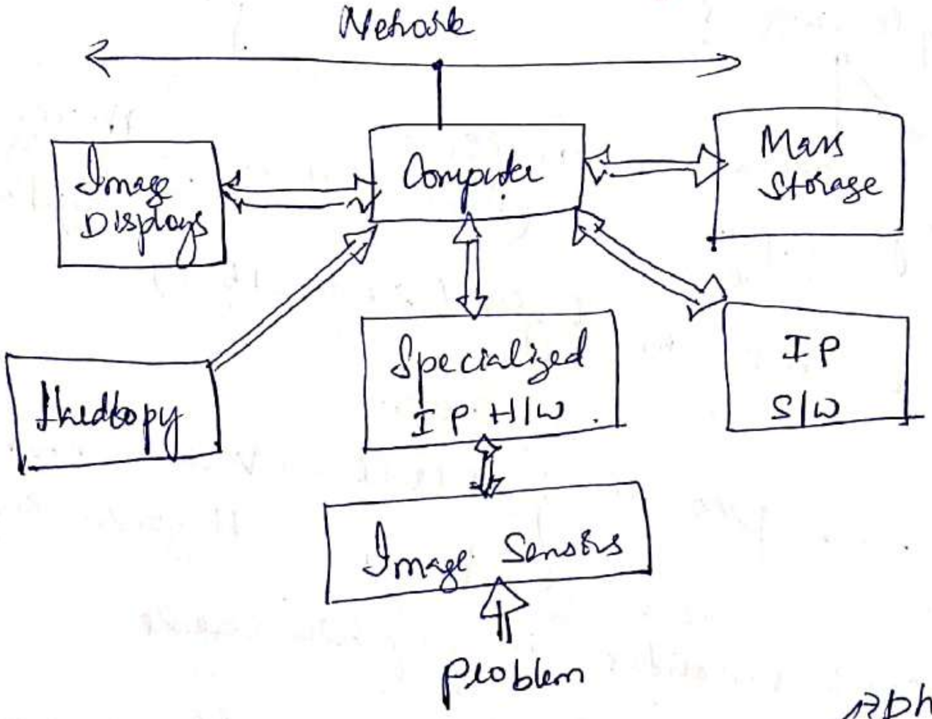
3 steps:-

① Importing an image with optical scanner or by digital photography.

② Analyzing & manipulating the image which includes data compression & image enhancement

③ Output image — result < altered image
report based on image analysis.

Components of an IP system



Sensing:- 2 elements are required < physical device is sensitive to the energy radiated by the object we wish to image.
digitizer. (O/P of device to digital form)

⑥

Specialized IP H/w : Digitizer + ALU (Arithmetic/logical op's in parallel)

Eg:- Averaging.

Computer :- > PC to super computer.
> offline IP tasks.

Software :- Specialized module → specific tasks

Matlab, C, Octave, Scilab, python, Java.

Mass Storage:- is a must

1024x1024 pixels - size in which each pixel is an 8 bit quantity.

1 image → requires 1 Mbyte of storage.

3 principle categories

- "frame buffer" Short term (while processing)
- Online (fast recall)
- archival (infrequent access) → "juke boxes" } Magnetic tapes or optical disks

Storage is measured in bytes (8 bits = 1 byte)

→ provide
- zoom
- Scroll → vertical shift
- pan → Horizontal shift.

Displays:- color monitors + graphic cards.

Hardcopy devices → laser printers, film cameras, heat sensitive devices, inkjet units, CD ROM disks

N/W :- image transmission — key consideration is BW.

optical fiber → & broadband.

Elements of Visual Perception

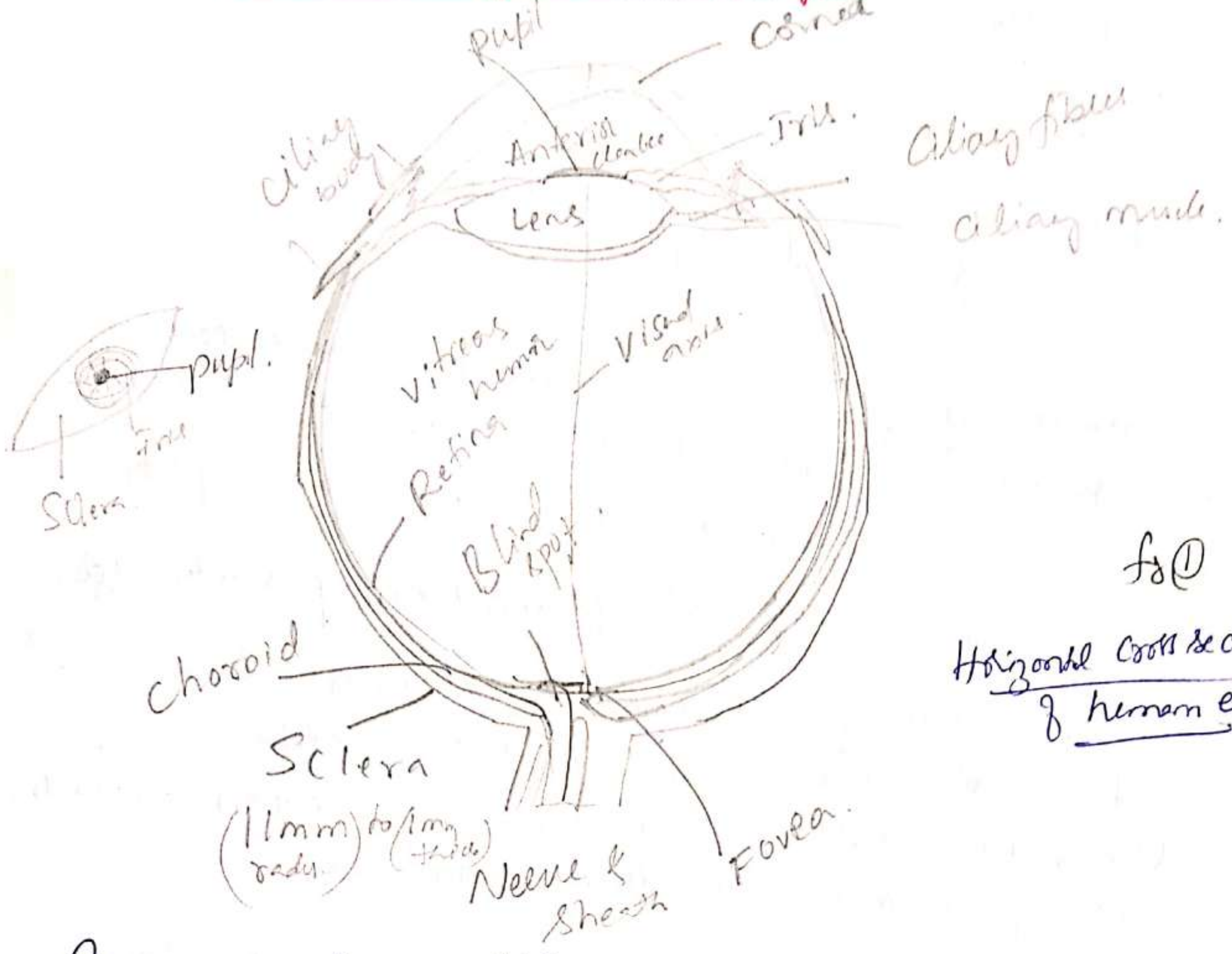


fig ①

Horizontal cross section of human eye

Structure of human eye

→ avg diameter 20mm

→ 3 membranes ① Cornea & Sclera outer cover
② Choroid
③ Retina.

→ Cornea → tough, transparent tissue
Sclera → Opaque membrane

→ Choroid → ① below Sclera
② N/w of blood vessels → major source of nutrition to the eye.

③ even single injury → not serious — blocks blood flow
~~blood~~

④ heavily pigmented & helps to reduce the amount of ex. light entering the eye.

⑤ Choroid — Iris
Ciliary body.

Iris contracts & expands to control the amount of light that enters eye.

Pupil — Varies in diameter 2 to 8 mm.

Front of the iris → Visible pigment of the eye
back → black pigment.

Lens → * made up of concentric layers of fibrous cells & is suspended by fibers that attach to the ciliary body.

- contains 60 to 70 % water, 6 % fat & more protein.
- colored by a slightly yellow pigmentation ↑ses with age.
- Excessive clouding of the eye → Cataracts lead to poor color discrimination & loss of clear vision.
- UV & IR light are absorbed by proteins within the lens if excessive, can damage the eye.

Retina * Innermost membrane of the eye, which lines the inside of the wall's posterior portion.

- When light from an object imaged on retina, eye is properly focused.
- Two classes of receptors ⟨ cones, rods

Cones – 7-8 million located primarily in the central portion of the retina, called the fovea, are sensitive to color.

Cone vision — photopic / bright light vision.

Rods – 75 to 150 million; as many receptors are connected to a single nerve, reduce the amount of detail - receptor.

Scotopic / dim - light vision.

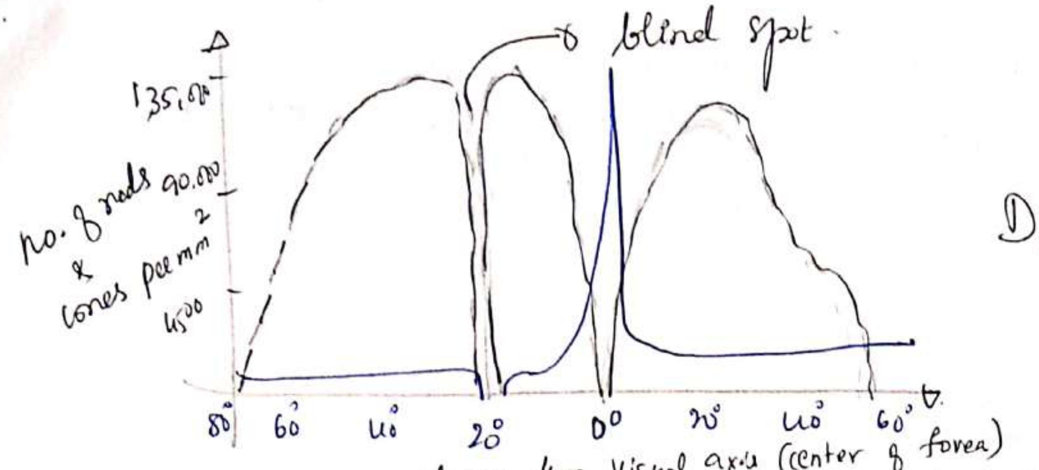


fig ②

Distribution of rods & cones in the retina

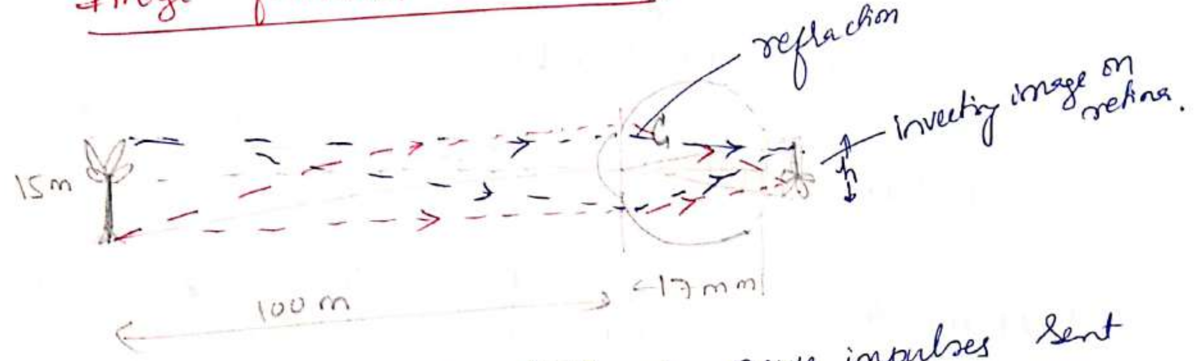
Figure ① ~~above~~ shows the density of rods & cones for a cross section of the right eye passing thro' the region of emergence of the optic nerve from the eye.

The absense of receptors — blind spot.

Receptor density is measured in degrees from the fovea.

$\frac{15}{100} = \frac{h}{17} \Rightarrow h = 2.55$

Image formation in the eye



Cones & rods → convert light into nerve impulses sent to the brain along the optive nerve.

7.)

Images formed anywhere other than on the retina are not transmitted effectively to the brain & hence Visual impairment.

eyesight, vision, seeing ⟶ Visual perception.

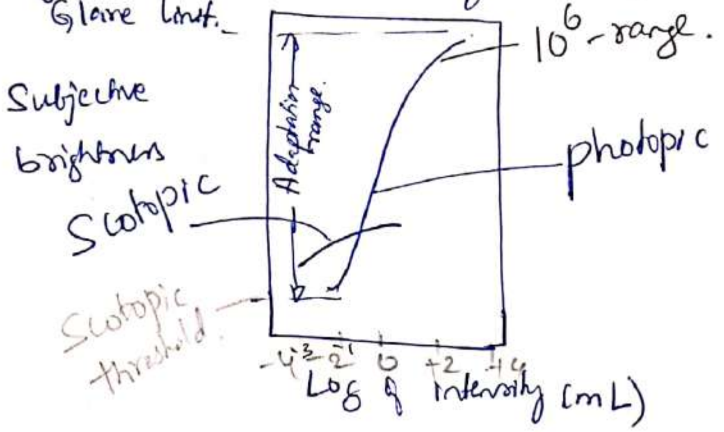
- In human eye, distance b/n the lens & the imaging region (retina) is fixed & the focal length needed to achieve proper focus is obtained by varying the shape of the lens. The fibers in the ciliary body accomplish this, flattening or thickening the lens for distant or near objects respecti-vely.

(10)

Perception then takes place by the relative excitation of light receptors which transform radiant energy into electrical impulses that are ultimately decoded by the brain.

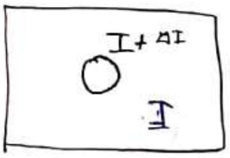
Brightness Adaptation & Discrimination

Experimental evidence indicates ~~fo~~ that Subjective brightness (intensity as perceived by the human visual system) is a logarithmic function of the light intensity incidence on the eye.



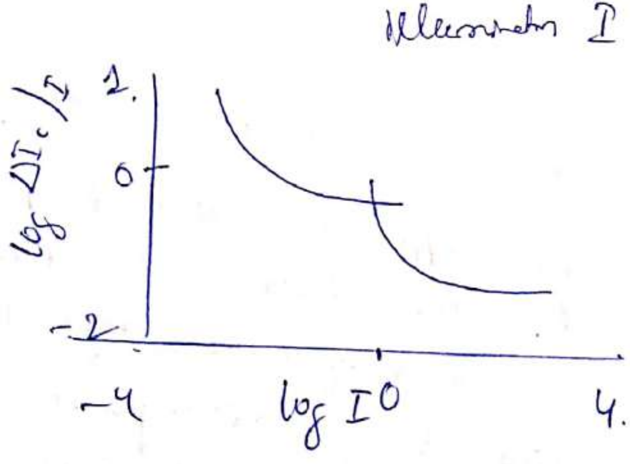
Brightness adaptation - changing its overall sensitivity.

Discrimination:-



$$\frac{\Delta I_c}{I} = \text{Weber ratio.}$$

ΔI_c → increment of illumination discriminable 50% of the time with background illumination I



Weber ratio as a fn of Intensity.

Linear Vs Nonlinear Operations

One of the most imp. classification of an IP method is whether it is linear or nonlinear.

Linearity = additivity + homogeneity.

$$H[f(x,y)] = g(x,y) \qquad H \rightarrow \text{linear operator}$$

$$H[a_i f_i(x,y) + a_j f_j(x,y)] = a_i H[f_i(x,y)] + a_j H[f_j(x,y)]$$

$$= a_i g_i(x,y) + a_j g_j(x,y).$$

$$\Sigma[a_i f_i(x,y) + a_j f_j(x,y)] = a_i \Sigma f_i(x,y) + a_j \Sigma f_j(x,y)$$

$$= a_i g_i(x,y) + a_j g_j(x,y).$$

Let $f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}$ & $f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$

$a_1 = 1$ & $a_2 = -1$.

$$\max\left\{ 1\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1)\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix} \right\} = \max\left\{\begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix}\right\} = -2$$

$$1 \max\left\{\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}\right\} + (-1)\max\left\{\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = 3 + (-7) = -4.$$

max is nonlinear.

Linear operations are exceptionally imp.

Image Sensing & Acquisition

Most of the images in which we are interested are generated by the combination of an "illumination" source & the reflection & absorption of energy from that source by the elements of the 'Scene' being imaged.

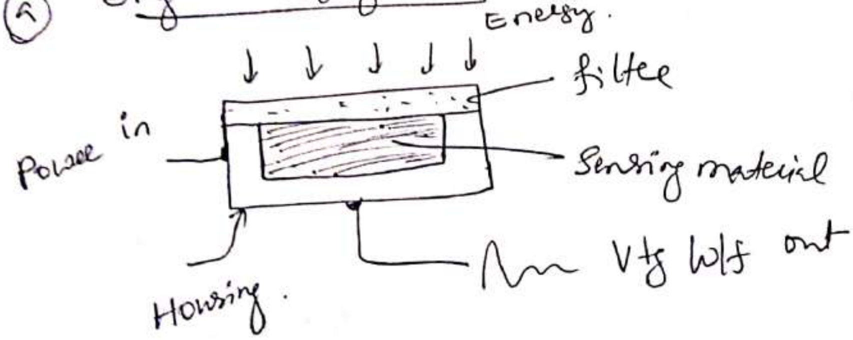
Eg:- illumination may originate from a source of electromagnetic energy such as Radar, infrared, & x-ray system.

X-rays pass thro' a patient's body for the purpose of generating a diagnostic x-ray film.

Three principal sensor arrangements used to transform illumination energy into digital images.

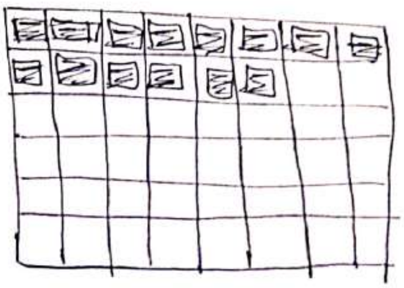
- ⓐ Single imaging Sensor
- ⓑ Line Sensor
- ⓒ Array Sensor.

ⓐ Single Imaging Sensor:



ⓑ

ⓒ



<u>Idea</u>:- Incoming energy is transformed into a voltage by the combination of i/p electrical power & sensor material ie responsive to the particular type of energy being detected.

The o/p Vtg w/f is digitized to get digital quantity.

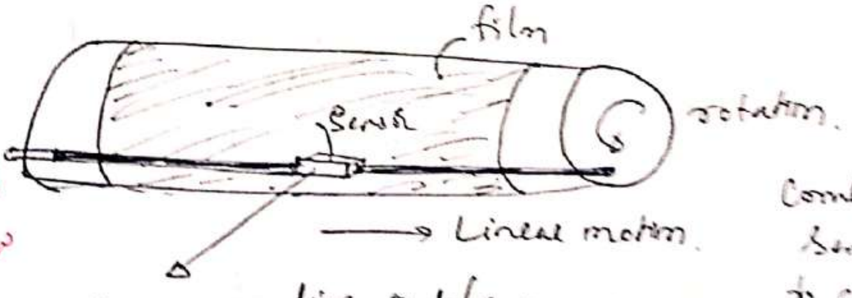
Image Acquisition using a Single Sensor

Fig a. Shows the components of a single sensor → photodiode which is constructed of Silicon material whose o/p vtg is proportional to light.

The use of a filter infront of a sensor improves selectivity.

Eg:- Green filter favours light in the green band of the color spectrum. ie sensor o/p will be stronger for green light than for other components in the visible spectrum.

fig (b) shows an arrangement used in high-precision scanning, where a film negative is mounted onto a drum whose mechanical rotation provides displacement in one dimension



One image line out/increment of rotation & full linear displacement of sensor from left to right.

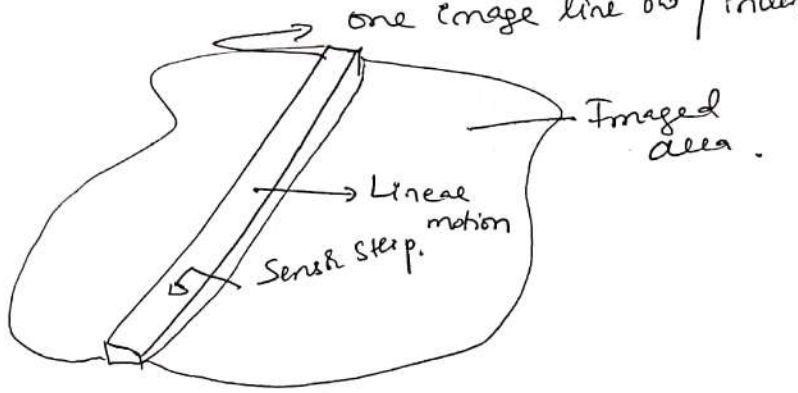
fig (b) Combining a single sensor with motion to generate a 2-D image.

Sensor is mounted on a lead screw

- Single sensor → ⊥r direction motion
    - Expensive method } to obtain high resolution images
    - Slow method. }

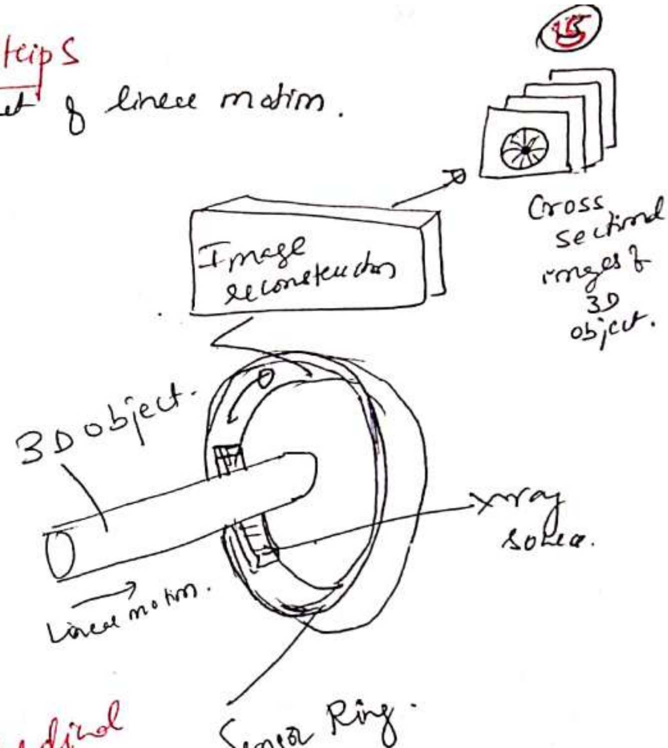
Other devices — microdensitometers → flat bed mech. digitizers.

Image Acquisition Using Sensor Strips



airborne imaging appⁿ

Using linear sensor strip



Medical (CAT) & Industrial imaging.

Circular sensor strip (Ring configuration)

Image Acquisition Toolbox → enable you to connect Industrial & scientific cameras to MATLAB / SIMULINK

Linear sensor strip: In-line sensors are used routinely for airborne imaging appns, in which the imaging system is mounted on an aircraft that flies at a constant altitude & speed over the geographical area to be imaged.

1-D imaging sensor strips that respond to various bands of the electromagnetic spectrum are mounted $\perp^{r}$ to the direction of flight.

Radiance - Total amount of energy that flows from the light source (W)

Luminance (lumens L) = measure of the amount of energy an observer percieves from a light source.

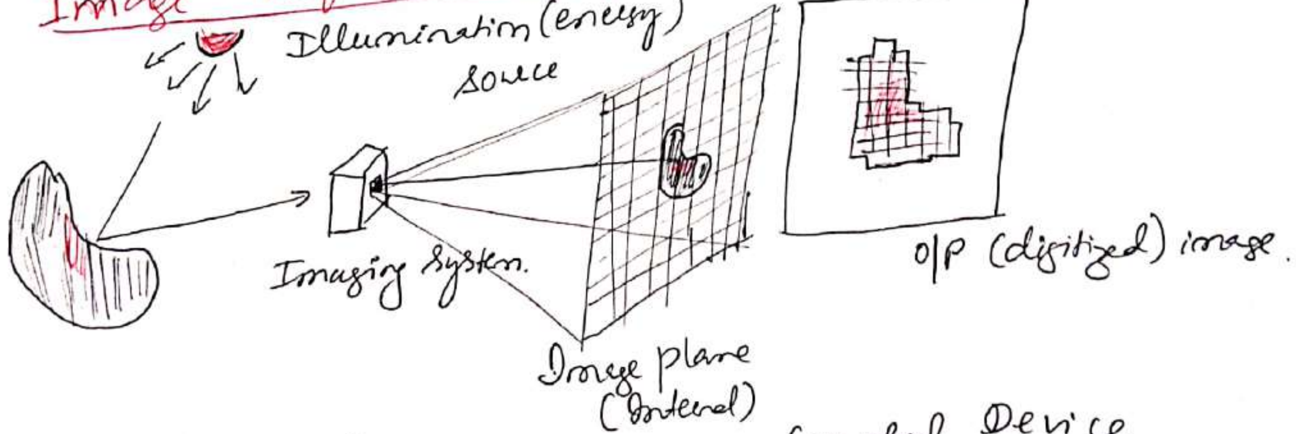
Brightness - Subjective descriptor of light perception practically impossible to measure.

A rotating X-ray source provides illumination & the sensors opp. the source collect the X-ray ~~source provides illumination~~ ~~& the sensors opp~~ energy that passes thro' the object.

This is the basis for medical & industrial CAT - Computerize Axial Tomography.

CAT - principle is also used in MRI - Magnetic Resonance Imaging & PET - Position Emission Tomography.

Image Acquisition Using Sensor Arrays



Typical Sensor — CCD - Charge Coupled Device

Digital Cameras - predominant. & for astronomical appns

Noise reduction is achieved by letting the sensor integrate the i/p light signal over mins or even hours.

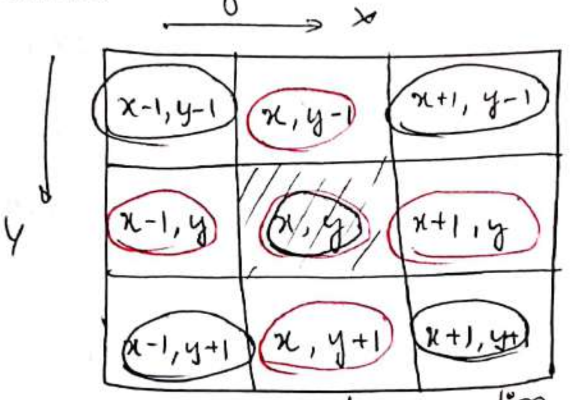
x Motion is not required

II Chapter :- Some basic Relationships between pixels.

$f(x, y) \rightarrow$ Image

① **Neighbors of a pixel :-**

A pixel p at coordinates (x, y) has 4 horizontal & vertical neighbors



x-1, y-1	x, y-1	x+1, y-1
x-1, y	x, y	x+1, y
x-1, y+1	x, y+1	x+1, y+1

coordinate convention

3 type

ⓐ 4-neighbors, N_4

ⓑ Diagonal neighbors, N_D

ⓒ 8-neighbors, N_8.

Together $N_8(P) = N_4(P) \cup N_D(P)$

$N_4(P)$ - red. $N_D(P)$ - black.

(ii) Adjacency:

Connectivity b/n pixels is a fundamental concept that simplifies the defn of numerous digital image concepts, such as regions and boundaries.

To determine if the 2 pixels are connected / adjacent or not, there are 2 conditions:

ⓐ Two pixels should be neighbors
ⓑ Their grey levels should be similar.

3 types of adjacencies are defined:
ⓐ 4-adjacency
ⓑ 8-adjacency
ⓒ m-adjacency.

Binary image → gray level values are 0 or 1.

ⓐ 4-adjacency:

2 pixels p and q are called 4 adjacent if q is in $N_4(p)$
p & q have same value (0 or 1)

0	0	0
0	1 p	1 q_1
0	0 q_2	0

p & q_1 → 4 adjacent
p & q_2 → Not adjacent.

ⓑ 8-adjacency:

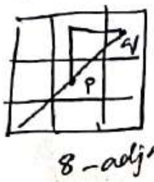
1 q_1	0	0
0	1 p	1 q_2
0	0	0 q_3

p & q_1 → 8 adjacent
p & q_2 → —"—
p & q_3 → are not 8-adjacent

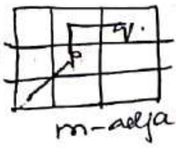
ⓒ m-adjacency (mixed):

(i) q is in $N_4(P)$
(ii) q is in $N_D(P)$ & $\{N_4(P) \cap N_4(q)\}$ is not same as p.

4 → broken paths
8 → multiple paths.



8-adjacency.



m-adjacency.

Graylevel ? In binary image → $V = \{1\}$ Set of gray level values used to define adjacency.

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

$V = \{11, 12 \ldots 25\}$

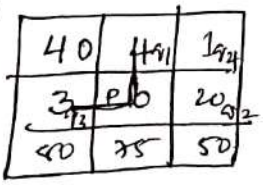
8-adjacent if q is in set $N_8(p)$.

m-adjacent if q is in ⓘ $N_4(p)$ or ⓘⓘ $N_D(p)$ and the set has no pixels whose values are from V.

ie ⓘⓘ q is in $N_D(p)$ & $\{N_4(p) \cap N_4(q)\} \notin V$.

→ gray values.

4-adjacency:



$V = \{0, 1, 2, 3, 4\}$.

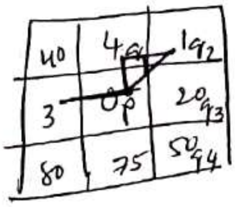
p & q1 → 4 adjacent

p & q3 → —"—

p & q2 → Not. (∵ 20 is not in V)

p & q4 → Not (∵ 1 is not 4 neighbor)

8-adjacency



p & q1 → 8 adjacent

p & q2 → 8 adjacent.

p & q3 → Not 8-adjacent (∵ 20 is not in V)

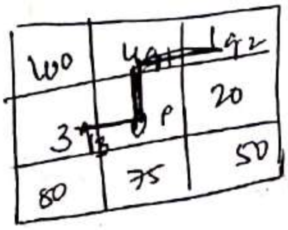
p & q4 → Not —"— (∵ 50 is not in V)

m-adjacency

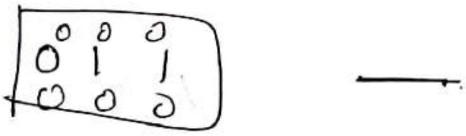
p & q1 → m-adjacent

p & q2 → not m-adjacent.

p & q3 → m adjacent.

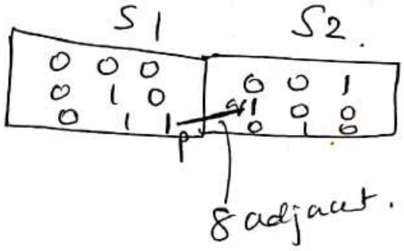


(iii) **Connectivity** :- 2 pixels are connected if they are adjacent.

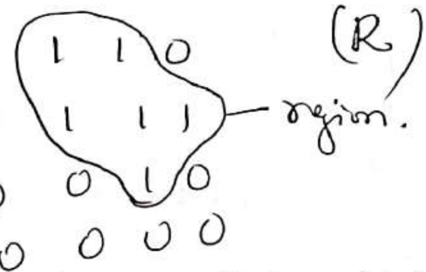


Two subsets are connected or adjacent if some pixel in S_1 is adjacent to some pixel in S_2.

$V = \{1\}$.



(iv) **Region** :-



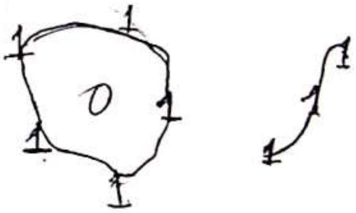
R is a subset of pixels in an image.

Region in an Image.

Every pixel in R is connected to other pixels in R, then R → Region

(v) **Boundary** :- Set of pixels in the region that have one or more neighbors that are not in R.

Edge of a Region → Boundary.



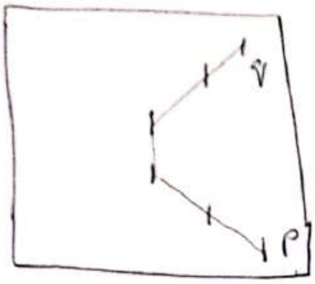
(vi) Path — Count of connected pixels = Length of path.

If first pixel = last pixel then Closed path.

Path

A digital path b/n pixel p having co-ordinates (x,y) to Pixel q with (u,v) coordinates is a sequence of connected pixels

(x,y) $(x_0, y_0)(x_1, y_1)$ . . . . (u,v)



Length of the path is count of connected pixels.

If first pixel is same as last pixel
i.e. (x,y) = (u,w) it is called closed path.

Distance Measure :- Dis(p,q) Distance b/n p & q.

(i) Dis (p,q) ≥ 0 If p = q ⟹ Dis(p,q) = 0

(ii) Dis (p,q) = Dis (q,p)

(iii) Dis (p,z) ≤ Dis (p,q) + Dis (q,z)

① 0 0 ① 0 0 ①
p (x,y) q (s,t) z (v,w)

(iv) Euclidean distance

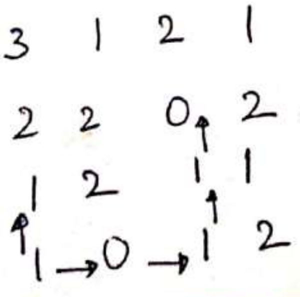
$$Dis_{eq}(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

Eg :- For V = {0,1} find the length of shortest 4, 8 & m-paths b/n p & q. Repeat for V = {1, 2} for the given image.

3 1 2 1 (q)
2 2 0 2
1 2 0 1
(p) 1 0 1 2

Sol

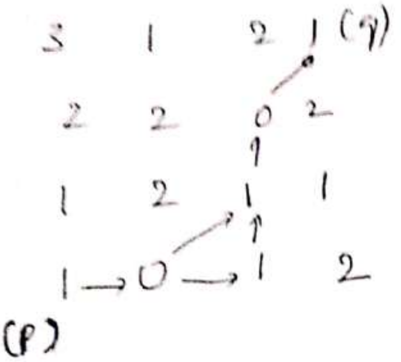
4-path :-



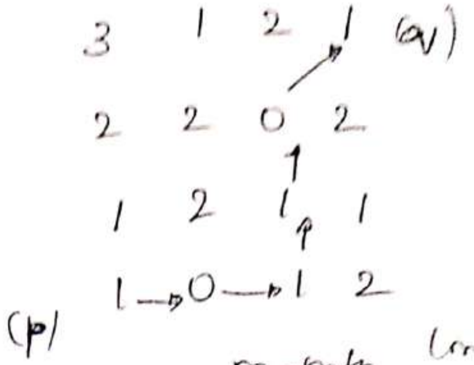
$I_{V} = \{0,1\}$

The path starts from p but doesnot reach q as no path exist b/n q & prev. pixel.

5c



8 path is not unique

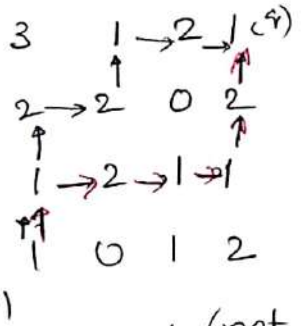


m-path (min length = 5)

II

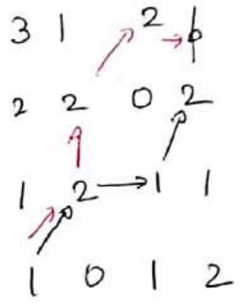
for V = {1, 2}

4 path

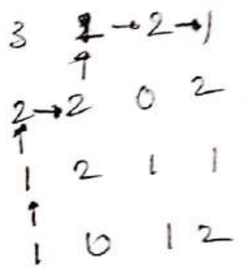


4 path (not unique)

min length = 6



8 path
not unique
min length = 4



m path
min length = 6

H/W. For V = {2, 3, 4} Compute the lengths of shortest 4, 8, & m paths b/n pq for the following image.

Jun 2016 (10m)

	3	4	1	2	0
	0	1	0	4	2 (q)
	2	2	3	1	4
(P)	3	0	4	2	1
	1	2	0	3	4.

③ An image of size 630 × 480 has 24 bit color. Calculate the memory required by the image.

$b = M \times N \times k.$

$= 630 \times 480 \times 24 = 7.2576$ Mbits.

④ Calculate no. of bits required to store a digital image of size 1024 × 1024 & no. of grey levels are 128.

$128 = 2^k \Rightarrow k = 7.$

$\boxed{L = 2^k.}$

$b = 1024 \times 1024 \times 7 = 7.34$ Mb

- $M \times N$ → size of array (pixels) (image)

L → discrete intensity levels

Due to storage & quantizing h/w considerations, no. of intensity levels L → Integer power of 2.

$L = 2^k$ k → bit image   if $k = 8$ then $L = 2^8 = 256$

b → no. of bits required to store a digitized image.

$\boxed{b = M \times N \times k}$

when $M = N$ then $\boxed{b = N^2 k}$

Eg:- 512×512 image with 256 graylevels at 300 baud rate, How many mins would it take to transmit ??

(baud rate → no. of bits transmitted/sec. Assumed each byte is one packet with start bit & stop bit) $L = 2^8$

$b = m \times n \times k$. $k = 8$.

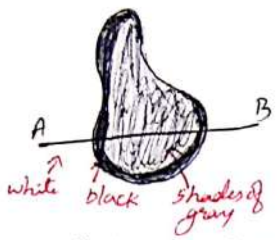
$$\text{Time} = \frac{M \times N \times k}{\text{baud rate}} \text{ secs.}$$

$$= \frac{512 \times 512 \times 8}{300} = \text{ secs.}$$

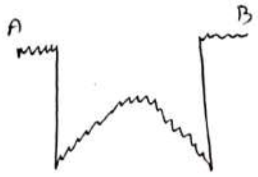
7

⇒ Image Sampling & Quantization

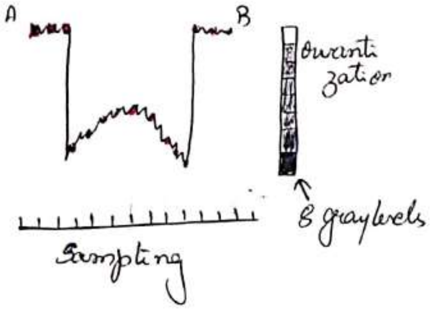
- * Continuous image $f(x,y)$ is converted to digital form
- * Image may be continuous w.r.t x & y co-ordinates & also in amplitude
- * Digitizing the co-ordinate values is called sampling & digitizing the amplitude values is called Quantization



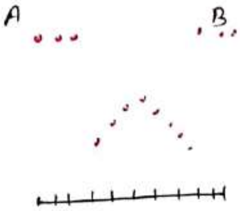
Continuous image



Scan line from A to B

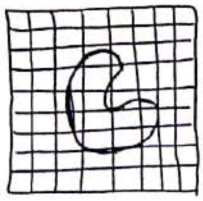


Sampling & quantization

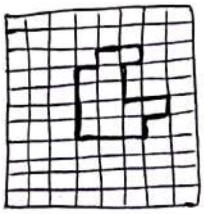


Digital scan line

- * In order to convert to digital function, the gray level values also must be converted (quantized) into discrete quantities
- * Starting at the top of the image & carrying out this procedure line by line produces a 2D digital image



Continuous image projected onto a sensor array



Result of image sampling & quantization

Spatial Domain Image Enhancement

Module 2 ①

Image Enhancement
- Spatial domain
- Transform domain

Spatial domain techniques — operate directly on the pixels of an image as opposed (freq. domain)

Frequency domain operations are performed on the FT of an image, rather than on the image itself.

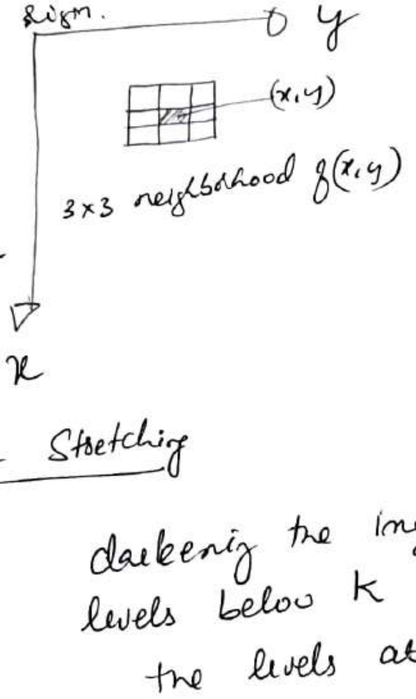
Spatial domain $g(x,y) = T[f(x,y)] \rightarrow ①$ $f(x,y) \rightarrow$ i/p image $g(x,y) \rightarrow$ o/p image

$T \rightarrow$ operator on 'f' defined over a neighborhood of point (x,y)

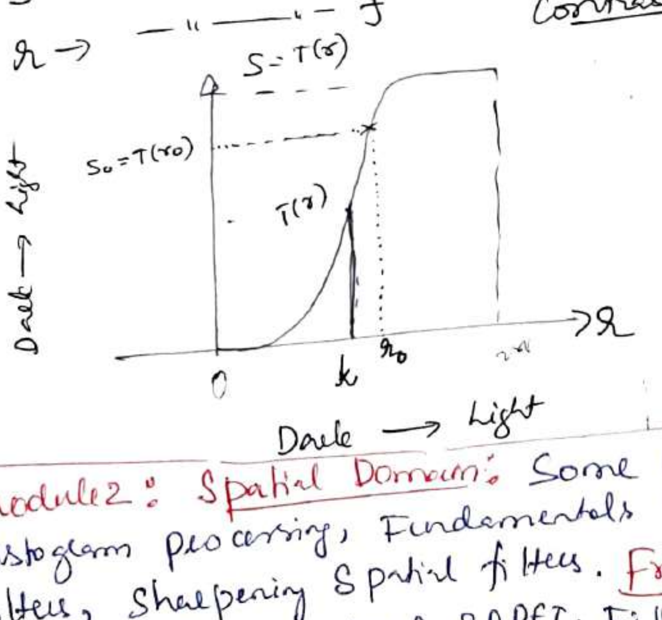
Basic implementation of eq ①.

$S = T(r)$

$T \rightarrow$ intensity transformation f^n



$S \rightarrow$ Intensity of g
$r \rightarrow$ — " — " — f



Contrast Stretching

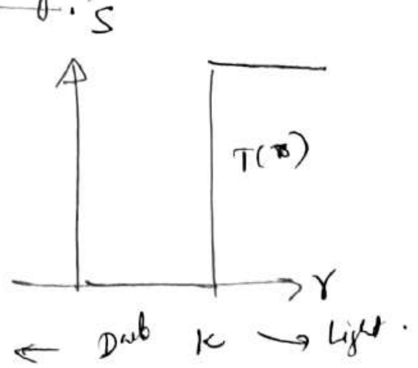
darkening the image intensity levels below k & brightening the levels above k.

Module 2: Spatial Domain: Some basic Intensity Transformation Functions, Histogram processing, Fundamentals of spatial filtering, Smoothing Spatial filters, Sharpening Spatial filters. Freq. domain: Preliminary concepts, DFT of 2 variables, properties of 2D DFT, Filtering in freq. domain, Image Smoothing & Sharpening using freq. domain filters, Selective filtering,

3.2 - 3.6
4.2, 4.5 - 4.10

Thresholding:

②



binary image (only Black & White)

Enhancement:- process of manipulating an image so that the result is more suitable than the original for a specific appⁿ.

X-ray is different

Sat image - IR are different.

No general "theory". Simple & basic.

Viewer is the ultimate judge.

But machine perception → quantify.

Eg. Character recognition → one level of enhancement is sufficient.

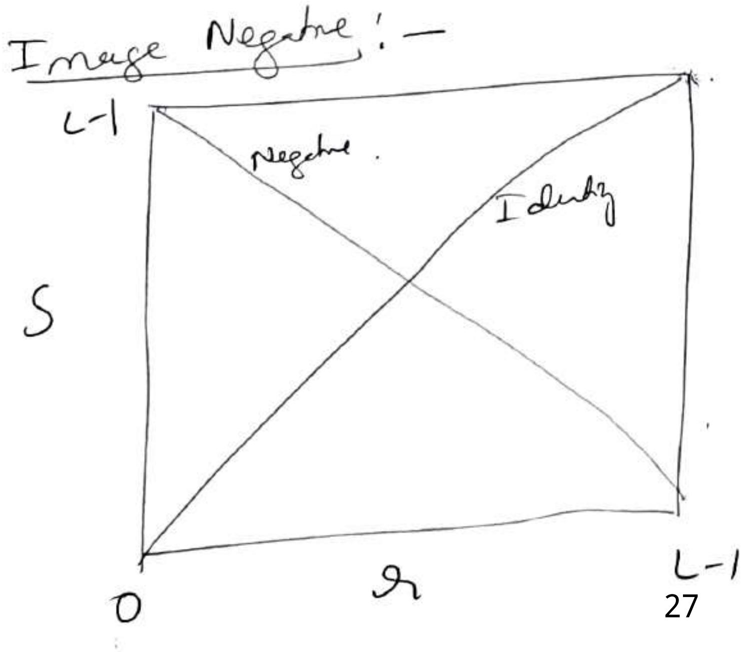
Basic Intensity Transformation functions

① Linear (Negative & Identity)

② Logarithmic (log & inverse log)

③ Power - law (nth power & nth root)

Image Negative:-



$\boxed{S = L-1-r.}$

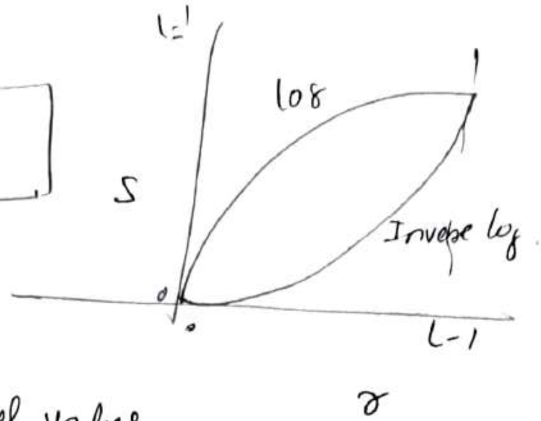
Photographic -ve.

Suitable for white/grey details in dark regions

Eg:- X-ray
Mammogram

② Log transformations :

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



To expand the values of dark pixels in an image while compressing higher-level values.

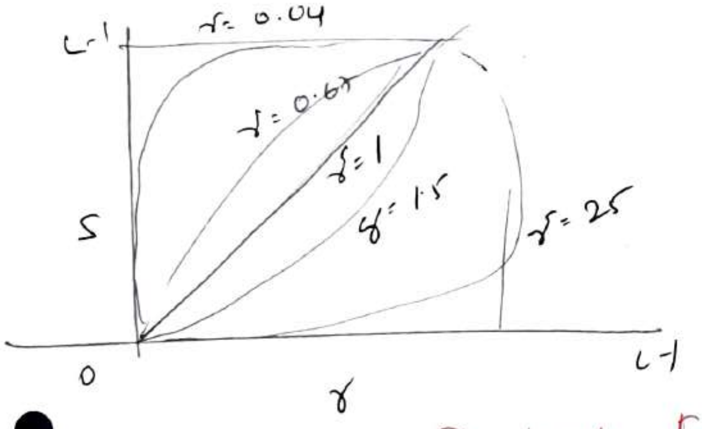
Eg:- Expand Fourier spectrum

③ Power-law (gamma):

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

c & γ → +ve constants

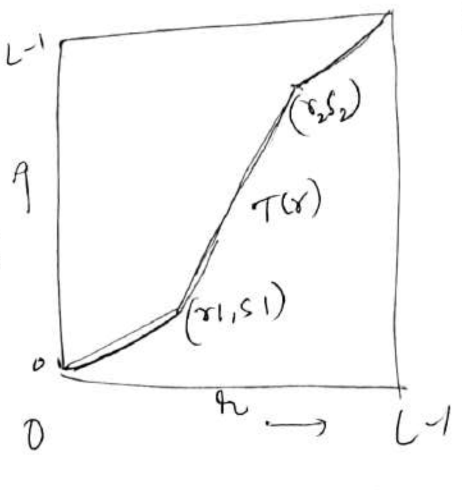
$$\boxed{s = c(r + \underset{\text{offsets (neglected)}}{\epsilon})^{\gamma}}$$



$c = \gamma = 1$ → Identity.

- Piecewise - Linear Transformation Functions

Bit plane slicing
Gray level slicing
Contrast stretching
Intensity level slicing.



Contrast Stretching :

Simplest piecewise - linear transformation fn.

low contrast images → poor illumination
→ lack of dynamic range in the imaging sensor
→ wrong setting of lens aperture during image acquisition

contrast stretch → ↑ dynamic range of the graylevels — Image process ④

locations (r_1, s_1) & (r_2, s_2) control the shape of the transformation.

If $r_1 = s_1$, $r_2 = s_2$ } Linear f^n

If $r_1 = r_2$ & $s_1 = 0$, $s_2 = L-1$ → Thresholding f^n → Binary image

Intermediate values of (r_1, s_1) & (r_2, s_2) → Various degrees of graylevels

In general $r_1 \leq r_2$ & $s_1 \leq s_2$ → so that f^n is single valued & monotonically ↑ing f^n.

If $r_1 = r_2 = m \Rightarrow$ mean gray level

Gray level slicing:

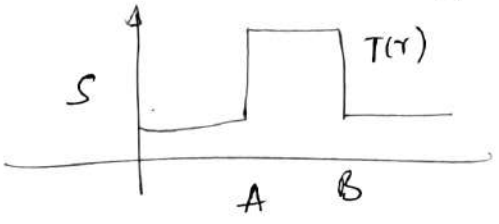
Highlighting a specific range of gray levels in an image often is desired.

Applications — ① water masses in sat. images
② flaws in X-rays.

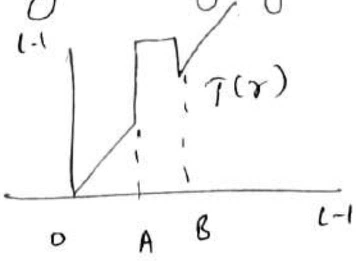
Several ways of slicing but 2 imp. basic themes:

I High value for all gray levels for range of interest & low value for all other gray levels.

II Brightening the desired range of graylevels but preserving background graylevel tonalities



Highlights range (A, B) and others constant level

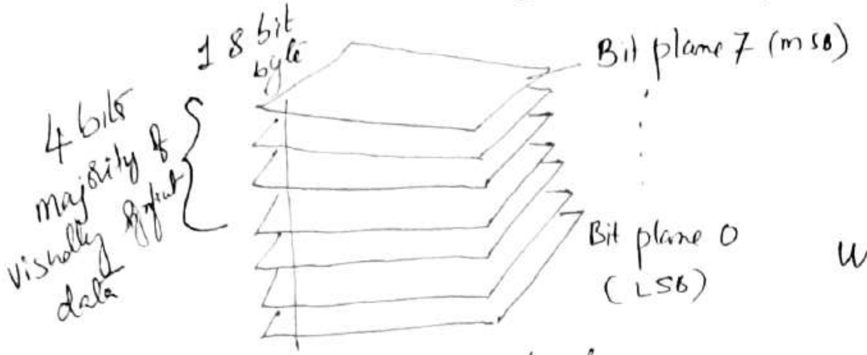


preserves all other levels

Bit-plane Slicing

8 bit image — 8 bit plans

bit plane 0 — LSB
bit plane 7 — MSB



used fr image compression

Bit plane extraction

Histogram Processing

- The histogram of a digital image with intensity levels in the range [0, L-1] is a discrete function $h(r_k) = n_k$, where r_k is the kth intensity value & $n_k \rightarrow$ no. of pixels in the image with intensity r_k.

M × N → row & columns of image matrix.

Normalizing histogram is given by $p(r_k) = \frac{n_k}{MN}$ for $k = 0, 1, 2 .. L-1$

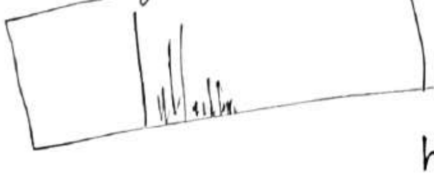
$p(r_k)$ is an estimate of the prob. of occurrence of intensity level r_k in an image.

- The sum of all components of a normalized histogram is equal to 1.

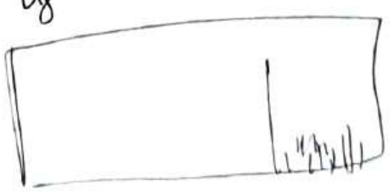
Histograms are ⟶ basis for numerous spatial domain processing techniques

Popular tool in IP. (real time)

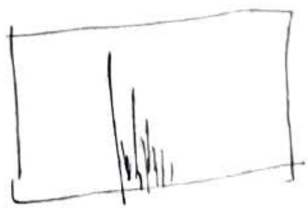
histogram of dark image.



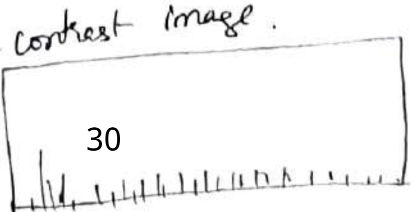
light image



Low contrast image.



high-contrast image.



Histogram Equalization:- process that attempts to spread out the graylevels in an image so that they are evenly distributed across their range.

Histogram equalisation reassigns the brightness values of pixels based on the image histogram.

Procedure:-

① Find the running sum of the histogram values.

② Normalise the values from step① by dividing by the total no. of pixels.

③ Multiply the values from step ② by the max. gray-level value & round.

④ Map the gray level values to the results from step③ using a one-to-one correspondence.

eg① Perform histogram equalization of the image
$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4. \end{bmatrix}$$

Max value = 5

min 3 bits to represent 5.

∴ possible graylevels = $2^3 = 8$ ie (0 to 7)

Total no. of pixels = 25 (5×5)

max.

Gray level	0	1	2	3	4	5	6	7
No. of pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Running Sum / total no. of pixels	$\frac{0}{25}$	$\frac{0}{25}$	$\frac{0}{25}$	$\frac{6}{25}$	$\frac{20}{25}$	$\frac{25}{25}$	$\frac{25}{25}$	$\frac{25}{25}$.
×ly by ⑦ max gray level	0	0	0	$\frac{6}{25}\times 7$ = 2	$\frac{20}{25}\times 7$ ≈ 6	$\frac{25}{25}\times 7$ ≈ 7	1×7 ≈ 7	1×7. ≈ 7

One-to-one correspondence.

⑦

Original grey level.	0	1	2	3	4	5	6	7
Histogram equalized values	0	0	0	2	6	7	7	7

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \xrightarrow[\text{Equalisation}]{\text{Histogram}} \begin{bmatrix} 6 & 6 & 6 & 6 & 6 \\ 2 & 6 & 7 & 6 & 2 \\ 2 & 7 & 7 & 7 & 2 \\ 2 & 6 & 7 & 6 & 2 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

- <u>Code</u> :-

```
clc ;
clear all ;
close all ;
a = imread ('___.png');
b = histeq (a);

imshow(a)      imshow(b)
im hist (a)    Im hist (b)
```

- Consider for a moment continuous intensity values & let 'r' denote the intensities of an image to be processed.

$r \rightarrow [0, L-1]$ range.

$r=0$ — dark $r = L-1 \rightarrow$ white

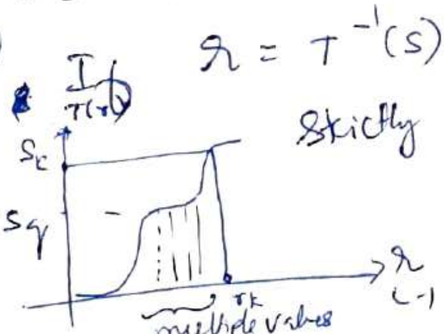
Let $S = T(r)$ $0 \le r \le L-1$ (intensity Transformation)

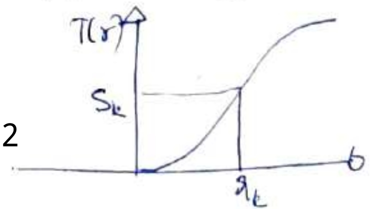
We assume ⓐ $T(r) \rightarrow$ monotonically ↑ fn. in 0 to L-1

ⓑ $0 \le T(r) \le L-1$ for $0 \le r \le L-1$

If $r = T^{-1}(s)$ for $0 \le S \le L-1$ then T(r) is a strictly ↑ fn in $0 \le r \le L-1$ (one to one)

(one to one may to one)





Strictly monotonically ↑ fn

~~Histogr~~

Let $p_r(r)$ & $p_s(s) \longrightarrow$ PDFs of r & s. [prob. density fn]

$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$ PDF of transformed image.

$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$ → CDF [cumulative distribution fn]

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

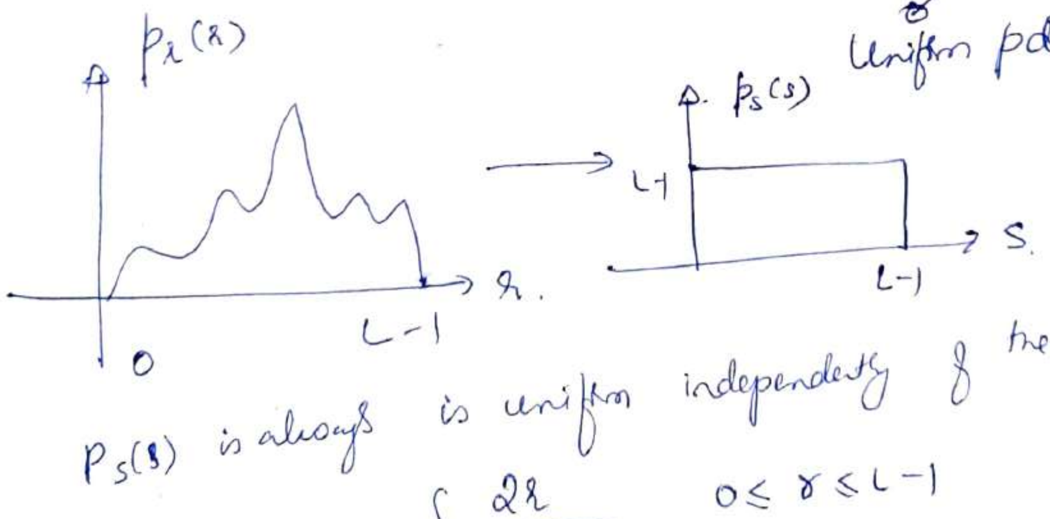
$$= (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right]$$

$$= (L-1)\,p_r(r)$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

$$= p_r(r)\left|\frac{1}{(L-1)\,p_r(r)}\right| = \frac{1}{L-1} \quad 0 \leq s \leq L-1$$

Uniform pdf.



$p_s(s)$ is always is uniform independently of the form of $p_r(r)$.

Eg:-
$$p_r(r) = \begin{cases} \frac{2r}{(L-1)^2} & 0 \leq r \leq L-1 \\ 0 & \text{otherwise.} \end{cases}$$

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = \frac{2}{L-1}\int_0^r w\,dw$$

$$= \frac{r^2}{L-1}$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = \frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right|$$ ⑨

$$= \frac{2r}{(L-1)^2}\left|\left[\frac{d}{dr}\frac{r^2}{(L-1)}\right]^{-1}\right|$$

$$= \frac{\cancel{2r}}{(L-1)^{\cancel{2}}}\left|\frac{\cancel{(L-1)}}{\cancel{2r}}\right| = \frac{1}{L-1}$$ → uniform PDF.

Discrete form of the transformation

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

$p_r(r_k) = \frac{n_k}{MN}$

$k = 0 - L-1$

Histogram → plot of $p_r(r_k)$ v/s r_k.

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

MN → total no. of pixels in the image

n_j → no. of pixels that have intensity value r_j

L → total no. of possible intensity levels in the image.

Computing the transformation fn $G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i)$

$G(z_q) = S_k$.

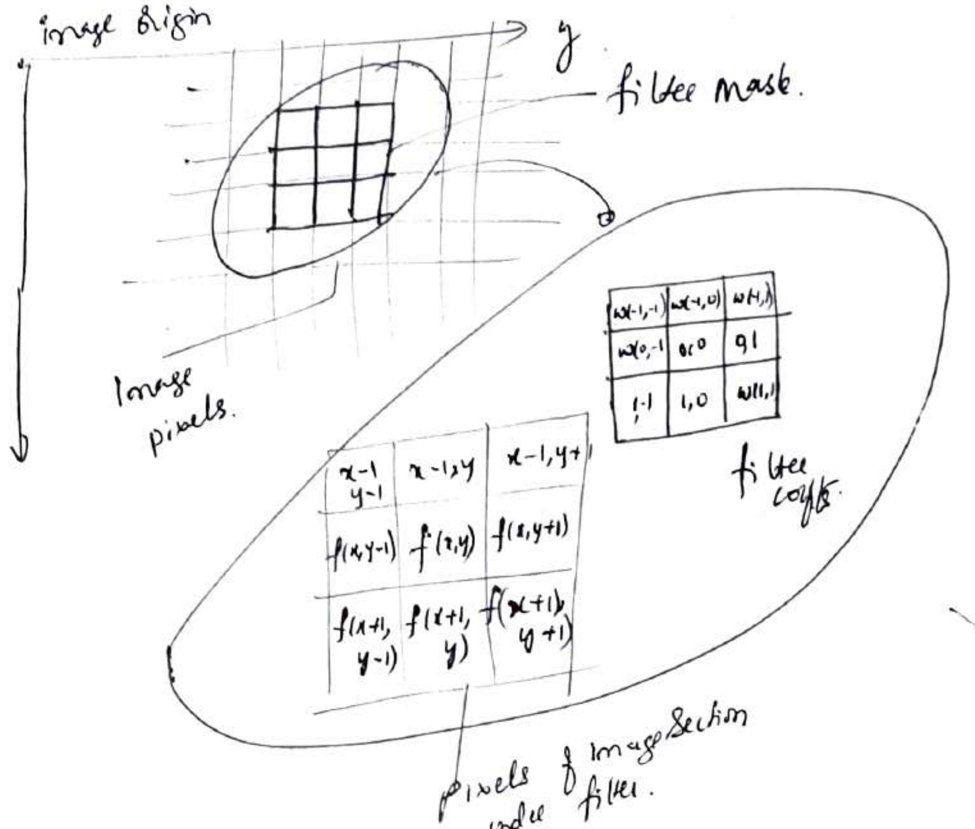
& $z_q = G^{-1}(s_k)$.

Arithmetic / Logic Operations

Neighbourhood operation :- The pixels in an image are modified based on some function of the pixels in their neighbourhood.

Linear filtering: Each pixel in the i/p image is replaced by a linear combination of intensities of neighbouring pixels. ie each pixel value in the o/p image is a weighted sum of the pixels in the neighbourhood of the corresponding pixel in the i/p image.

Linear filtering can be used to smoothen an image as well as sharpen the image.



Mean filter :- (Avg filter / LPF)

Replaces each pixel by the avg of all the values in the local neighbourhood.

Eg:- 3x3 mask $= \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

& 5x5 mask = $\frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

Limitations :-

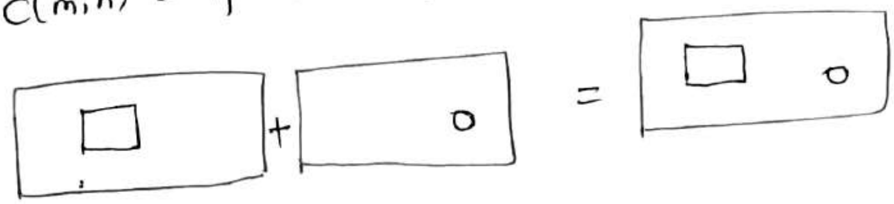
① Avg operation leads to the blurring of an image. Blurring affects feature localisation.

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

(a) Image addition :

" Superimposing images "

$c(m,n) = f(m,n) + g(m,n)$



(b) Image subtraction :

" Detection "

$c(m,n) = f(m,n) - g(m,n)$

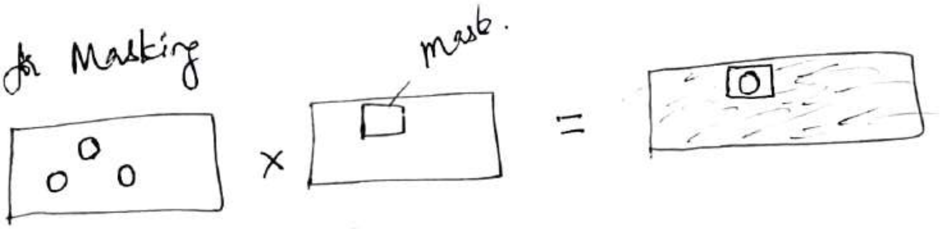
```
a = imread(' ');
b = imread(' ');
c = double(a) + double(b);
imshow(c);
```

(c) Multiplication / Division :

* /

" Background Suppression "

→ used for Masking



```
a = imread(' ');
b = 0.35 + zeros(242, 308);
[m n] = size(b);
  for i = 20:98
        for j = 85:164
        b(i,j) = 1;
     end
  end
c = double(a).*b;
```

Mean filter

Eg:-

$$\begin{vmatrix} 1 & 2 & 3 \\ 2 & \textcircled{4} & 5 \\ 3 & 4 & 3 \end{vmatrix} \xrightarrow{3\times3 \text{ box filter}} \begin{vmatrix} 1 & 2 & 3 \\ 2 & \textcircled{3} & 5 \\ 3 & 4 & 3 \end{vmatrix}$$

$$\frac{1}{9} \times [1+2+3+2+4+5+3+4+3] = \frac{27}{9} = 3$$

Median filter

Median filter – Order-Statistic (Non linear) filter.
effective – salt & pepper noise, impulse noise

$$\begin{bmatrix} 24 & 22 & 33 & 25 & 32 & 24 \\ 34 & \textcircled{255} & 24 & \textcircled{0} & 26 & 23 \\ 23 & 21 & 32 & 31 & 28 & 26 \end{bmatrix}$$

255 – Salt noise

0 – pepper noise.

$21, 22, 23, 24, \textcircled{24}, 32, 33, 34, 255$ – median.

$0, 21, 22, 24, \textcircled{25}, 31, 32, 33, 255$ – median

$0, 24, 25, 26, \textcircled{28}, 31, 32, 32, 33$ – median

$0, 23, 24, 25, \textcircled{26}, 28, 31, 32$

median filtered

$$\begin{bmatrix} 24 & 22 & 33 & 25 & 32 & 24 \\ 34 & \textcircled{24} & \textcircled{25} & \textcircled{28} & \textcircled{26} & 23 \\ 23 & 21 & 32 & 31 & 28 & 26 \end{bmatrix}$$

Other - nonlinear filter – Max filter
Min filter.

Sharpening Spatial filters

Principal objective — to highlight transitions in intensity

Uses — electronic printing, medical imaging - Industry, military app^n.

avg → conti → integration (analogous) → causes blurring.

Sharpening → spatial differentiation

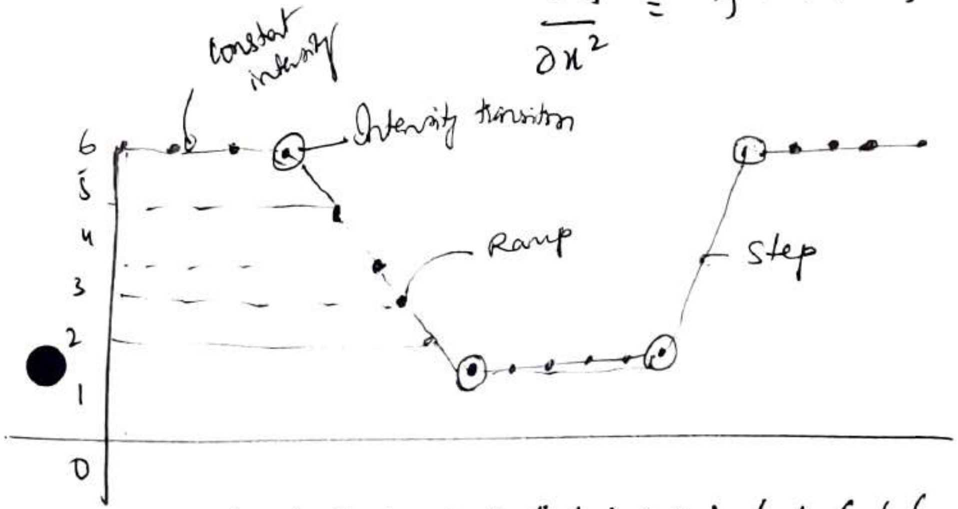
Image differentiation enhances edges & other discontinuities

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

$\frac{\partial f}{\partial x} = \frac{df}{dx}$ (If only one variable there in f^n)

$f(x-1)$	$f(x)$	$f(x+1)$

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



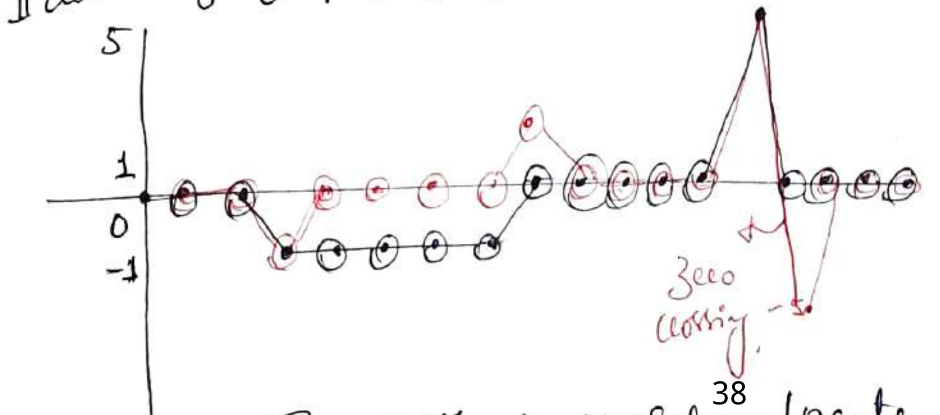
6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6

I derivative 0 0 0 -1 -1 -1 -1 -1 0 0 0 0 0 5 0 0 0 0

II derivative 0 0 -1 0 0 0 0 1 0 0 0 0 5 -5 0 0 0

Observations
① constant intensities
I derivative → zero.
② step | ramp
→ Nonzero.
③ along ramps - Nonzero

II derivative.
① Constant intensities
zero
② onset & end of ramp | step - Nonzero.
③ along ramps - zero.



Zero crossing is useful — locate edges

Using the Second Derivative for Image Sharpening - The Laplacian

Isotropic filters; rotation invariant

Rotating the image & then applying the filter gives the same result as applying the filter to the image first & then rotating the result.

Simplest isotropic derivative operator is the Laplacian.

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

fn of 2 variables.

Laplacian is a linear operator

In the x-direction, we have $$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

Similarly, in the y-direction,

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

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

Masks:

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

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

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

Unsharp Masking & High boost filtering

Subtracting an unsharp (smoothed) version of an image from the original image

3 steps:
① Blur the original image
② Subtract the blurred image from the original
③ Add the mask to the original.

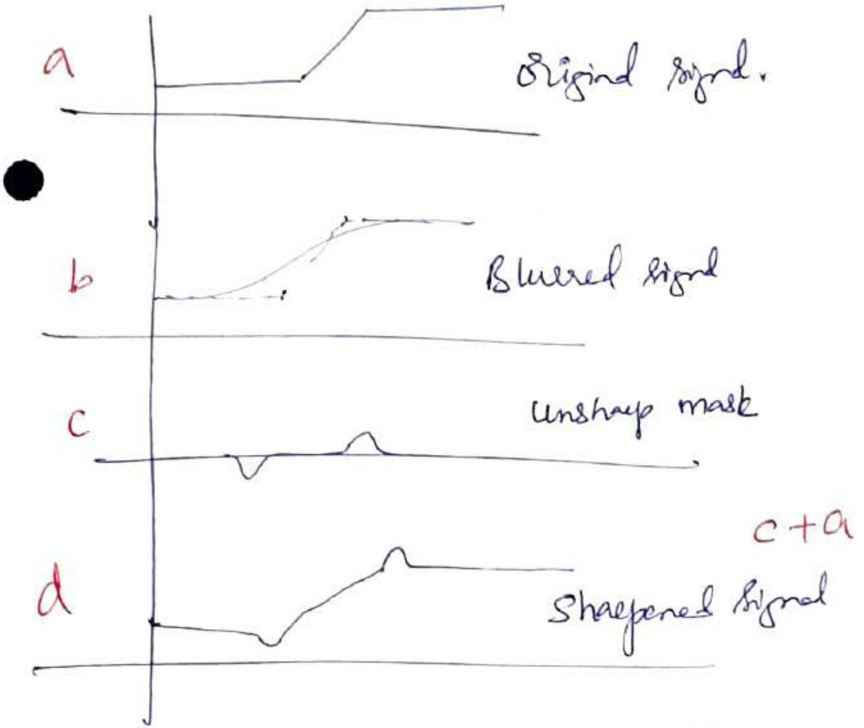
$f(x,y) \rightarrow$ original $\bar{f}(x,y) \rightarrow$ blurred

~~gef~~ $g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)$

$g(x,y) = f(x,y) + k * g_{mask}(x,y)$ $k \rightarrow wt.$

if $k = 1$ → process is called unsharp masking.

if $k > 1$ → process is called Highboost filtering.



1.D illustration of the mechanics of unsharp masking.

- Using I order derivatives for (Nonlinear) image sharpening.

The Gradient.

$$\nabla f = \overset{\text{not isotropic}}{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$$

Vector has imp geometrical property - it points in the direction of the greatest change of f at location (x,y)

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

Roberts operator:-

-1	0
0	1

0	-1
1	0

Sobel operators:-

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

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

(16)

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

$f(x,y)$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

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

I order derivative

$g_x = (z_8 - z_5)$ & $g_y = (z_6 - z_5)$

Roberts proposed $g_x = (z_9 - z_5)$ & $g_y = (z_8 - z_6)$

$$\therefore M(x,y) = \sqrt{(z_9 - z_5)^2 + (z_8 - z_6)^2}$$

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

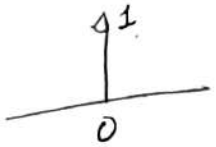
→ Roberts cross-gradient operator.

Filtering in Freq. domain:

Functions of Two Variables

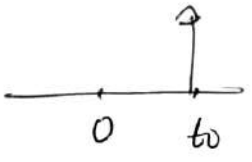
δ(t):

1D: $$\delta(t) = \begin{cases} 1 & \text{if } t=0 \\ 0 & \text{elsewhere.} \end{cases}$$

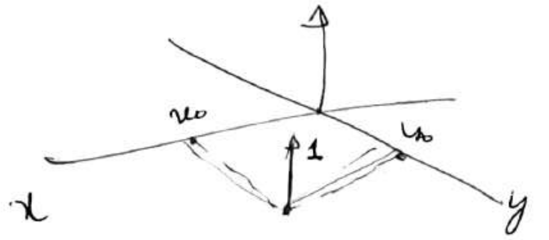


$$\delta(t-t_0) = \begin{cases} 1 & \text{if } t=t_0 \\ 0 & \text{elsewhere} \end{cases}$$

Sifting property



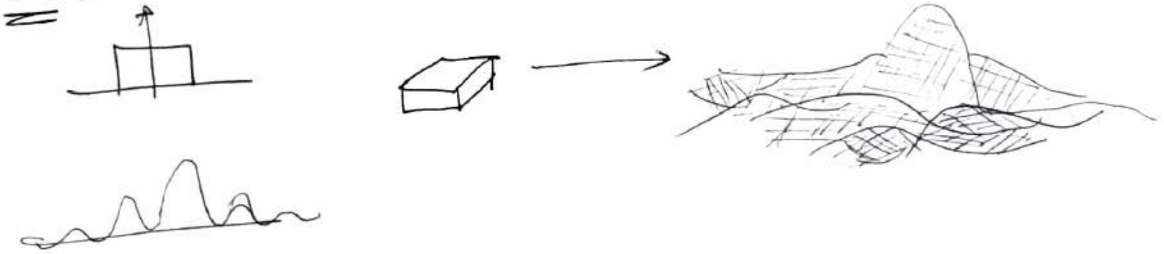
2D:



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

$$\delta(x-x_0,\, y-y_0) = \begin{cases} 1 & x=x_0 \,\&\, y=y_0 \\ 0 & \text{elsewhere.} \end{cases}$$

FT:



2D-DFT:-

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

$f(x,y) \rightarrow$ size $M \times N$

$u = 0, 1, 2 \cdots M-1$
$v = 0, 1, 2 \cdots N-1$.

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

$x \rightarrow 0, 1, 2 \cdots M-1$
$y \rightarrow 0, 1, 2 \cdots N-1$

Properties of 2D-DFT

① Relationships b/n spatial & freq. intervals:

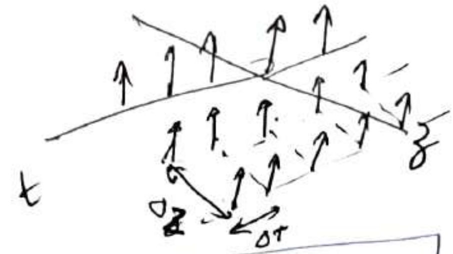
Conti. time $f(t,z)$ is sampled to form $f(x,y) \rightarrow M \times N$ samples

Let ΔT & $\Delta Z \rightarrow$ Separations b/n Samples.

⑩

Then the separations b/n the corresponding discrete freq-domain variables are given by

$$\Delta u = \frac{1}{M\Delta T} \quad \& \quad \Delta v = \frac{1}{N\Delta Z}$$



② **Translation & Rotation:**

Shifts the origin of DFT to (u_0, v_0)

$$f(x,y)\, e^{j2\pi\left(\frac{u_0 x}{M} + \frac{v_0 y}{N}\right)} \longleftrightarrow F\left((u-u_0)(v-v_0)\right)$$

$$\& f(x-x_0, y-y_0) \longleftrightarrow F(u,v)\, e^{-j2\pi\left(\frac{x_0 u}{M} + \frac{y_0 v}{N}\right)}$$

→ Translation Property.

Rotation:- If $x = r\cos\theta$ $\quad$ $u = \omega\cos\phi$ $\quad$ then

$y = r\sin\theta$ $\quad$ $v = \omega\sin\phi$

$$f(r, \theta+\theta_0) \longleftrightarrow F(\omega, \phi+\theta_0)$$

Rotating f(x,y) by an angle θ_0 → rotates F(u,v) by the same angle.

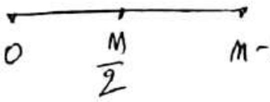
③ **Periodicity:**

$$F(u,v) = F(u+k_1M, v) = F(u, v+k_2N)$$

$$= F(u+k_1M, v+k_2N)$$

$$f(x,y) = f(x+k_1M, y) = f(x, y+k_2N) = f(x+k_1M, y+k_2N)$$

k_1, k_2 → integer.



$$f(x)\, e^{j2\pi\left(\frac{u_0 x}{M}\right)} \longleftrightarrow F(u-u_0)$$

Let $M/2 = u_0$

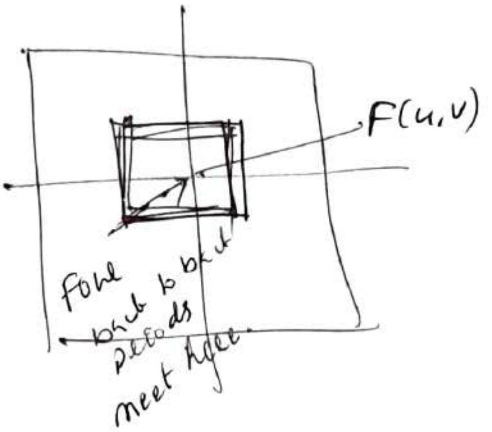
$$f(x)\, e^{j\pi \frac{M}{2} x / M} \longleftrightarrow F(u - M/2)$$

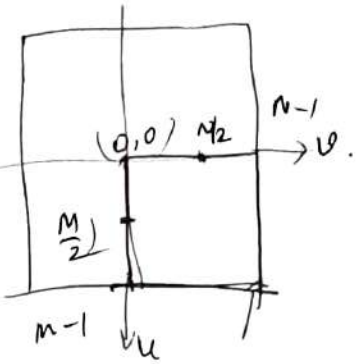
$$f(x)\, (-1)^x \longleftrightarrow F(u - M/2)$$

Multiplying f(x) by $(-1)^x$ shifts F(0) to center of the interval (M/2) 0 to M-1

For 2D DFT,

$$f(x,y)\,(-1)^{x+y} \longleftrightarrow F\left(u-\frac{M}{2},\, v-\frac{N}{2}\right).$$





④ <u>Convolution:</u>

$$f(x,y) * g(x,y) \longleftrightarrow F(u,v)\cdot G(u,v)$$

Defn of conv.

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

⑤ <u>Correlation:-</u>

$$f(x,y) \circ g(x,y) \longleftrightarrow F^*(u,v)\,.\,G.(u,v).$$

⑥ <u>Scaling:-</u>

$$a f(x,y) \longleftrightarrow a\,.\,F(u,v)$$

$$f(ax,\, by) \longleftrightarrow \frac{1}{|ab|}\, F\left(\frac{u}{a}, \frac{v}{b}\right)$$

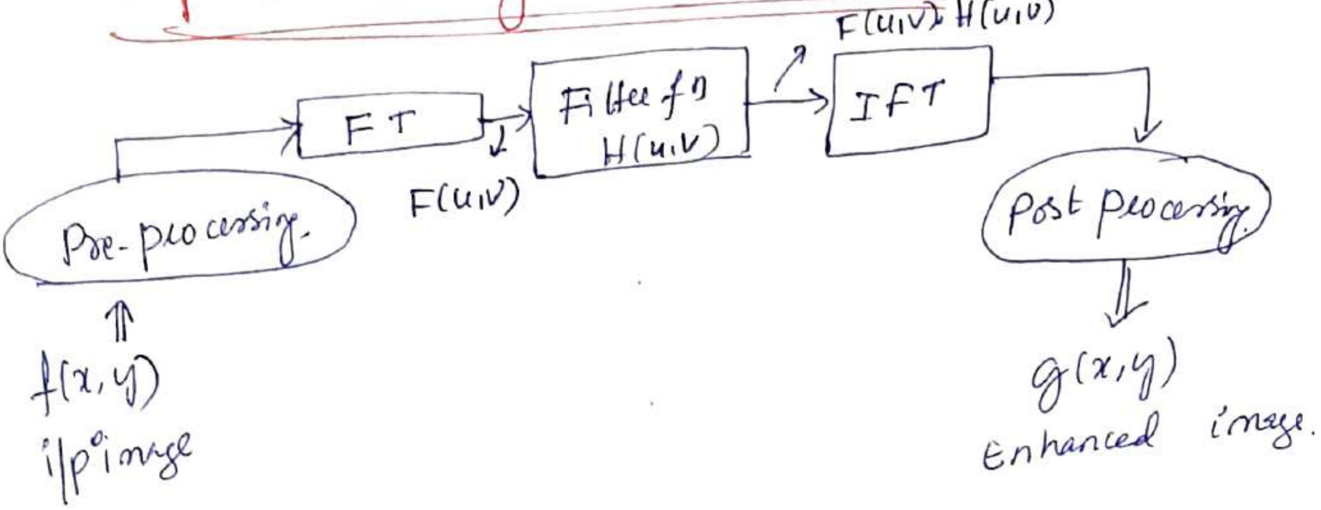
⑥ <u>Symmetry:</u> $\text{DFT}\{f^*(m,n)\} = F^*(-k,-l)$

$$F(k,l) = F^*(-k,-l).$$

⑦ <u>Linearity:-</u>

$$a f(x,y) + b g(x,y) \longleftrightarrow a F(u,v) + b\, G(u,v)$$

Steps for filtering in Freq. domain..



Steps:-

① Multiply the input image by $(-1)^{x+y}$ to center the transform.

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

③ Multiply F(u,v) by a filter fn H(u,v)

④ Compute the IDFT of the result in step ③.

⑤ obtain the real part of the result in step ④

⑥ multiply the result by $(-1)^{x+y}$

Preprocessing → Zero padding.

Post processing → cropping.

H(u,v)

$h(x,y) \longrightarrow$ Impulse response of $H(u,v)$

∵ all quantities in a discrete implementation of $h(x,y) \longleftrightarrow H(u,v)$ are finite, such filters are also called — FIR (finite impulse response)

These are only — linear spatial filters considered.

Some basic filters:- ① $H(u,v) = \begin{cases} 0 & \text{if } (u,v)=(M/2, N/2) \\ 1 & \text{otherwise} \end{cases}$

- It is called Notch filter — It is a constant fn with a hole (notch) at the origin.

Low frequencies in FT are responsible for the general gray-level appearance of an image over smooth areas.

high frequencies in FT are responsible for details such as edges & noise

Filter that attenuates high frequencies — LPF

- " " low " — HPF.

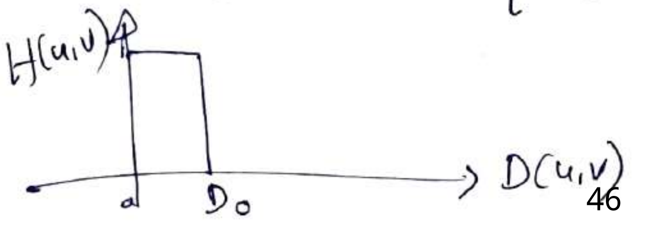
Image Smoothing using Fr. domain filter

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

$D_0 \rightarrow$ +ve constant

$D(u,v) \rightarrow$ distance b/n pt (u,v) & center of fr. rectangle.

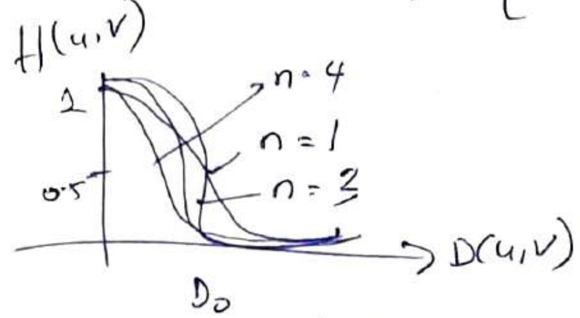
$$D(u,v) = \left\{ \left(u-P/2\right)^2 + \left(v-\frac{Q}{2}\right)^2 \right\}^{1/2}$$



v^2

② **Butterworth LPF**

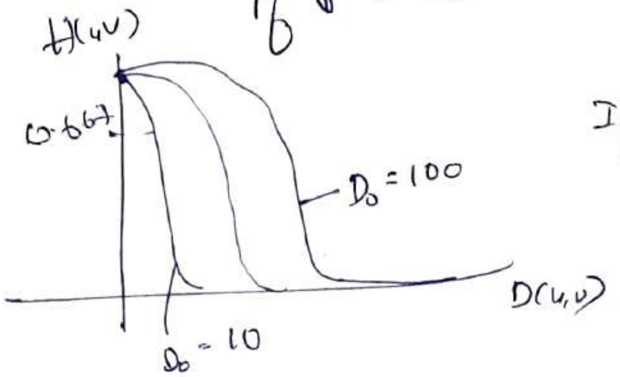
$$H(u,v) = \frac{1}{1 + \left[D(u,v)/D_0\right]^{2n}} \quad \text{order} \rightarrow n$$



③ **Gaussian LPF**

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

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



If $D_0 = D(u,v) \longrightarrow 0.667$ max value. (Down)

LPF → Printing & publishing industry.

Image Sharpening using Freq. domain filter

$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

① Ideal HPF

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

② Butterworth HPF

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

③ Gaussian HPF

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

Laplacian in the frq. domain

Laplacian can be implemented in freq. domain using the filter

$$H(u,v) = -4\pi^2(u^2+v^2)$$

or $$H(u,v) = -4\pi^2\left[\left(u-\frac{P}{2}\right)^2+\left(v-\frac{Q}{2}\right)^2\right]$$

$$= -4\pi^2[D^2(u,v)]$$

$D(u,v) \rightarrow$ distance f^n.

Laplacian image $\nabla^2 f(x,y) = F^{-1}(H(u,v).F(u,v)]$

- $$g(x,y) = f(x,y) + c\nabla^2 f(x,y)$$

$c=-1$ as $H(u,v)$ is negative

$$g(x,y) = F^{-1}\left[F(u,v) - H(u,v).F(u,v)\right\}$$

$$= F^{-1}\left[\{1 - H(u,v)\} F(u,v)\right]$$

$$= F^{-1}\left\{\left[\{1 - 4\pi^2 D^2(u,v)\} F(u,v)\right\}\right.$$

- Unsharp Masking, High boost filtering & High frq. emphasis filtering

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

$$f_{LP}(x,y) = F^{-1}\left[\underset{\substack{\downarrow\\ \text{LPF}}}{H_{LP}(u,v)} . \underset{\substack{\downarrow\\ F\{f(x,y)\}}}{F(u,v)}\right]$$

$$g(x,y) = f(x,y) + k * g_{mask}(x,y)$$

$k>1 \longrightarrow$ highboost filtering

$k\leq 1 \longrightarrow$ Unsharp masking

$$= \left\{f(x,y) + k * [f(x,y) - f_{LP}(x,y)]\right\}$$

$$= f(x,y)\left[1 + k * \left[1 - \frac{f_{LP}(x,y)}{f(x,y)}\right]\right]$$

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

$$\therefore\ g(x,y) = F^{-1}\left\{\underbrace{\left[1 + k * H_{HP}(u,v)\right]} F(u,v)\right\}$$

High frq. Emphasis filter :-

(8)

$$g(x,y) = F^{-1}\left\{\left[k_1 + k_2 * H_{HP}(u,v)\right] F(u,v)\right\}$$

where $k_1 \geq 0$ gives controls to the offset from the origin

& $k_2 \geq 0$ controls the contribution of high frequencies.

Combination of High frq emphasis & histogram equalization is superior to the result

Homomorphic Filtering

$$\text{Let } f(x,y) = i(x,y)\,r(x,y) \longrightarrow ①$$

slow, fast

illumination-reflectance model.

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

Taking ln on b.s.

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

$$F\{z(x,y)\} = F\{\ln\{f(x,y)\}\} = F_i(u,v) + F_r(u,v)$$

$$Z(u,v) = F_i(u,v) + F_r(u,v)$$

We can filter $Z(u,v)$ using a filter $H(u,v)$, so that

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

$$= H(u,v)\,F_i(u,v) + H(u,v)\,F_r(u,v)$$

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

$$s(x,y) = F^{-1}\{H(u,v) F_i(u,v)\} + F^{-1}\{H(u,v) F_r(u,v)\}$$

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

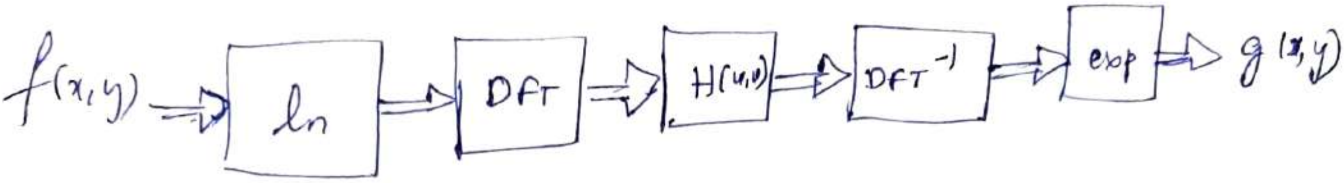
$$r'(x,y) = F^{-1}\{H(u,v) F_r(u,v)\}$$

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

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

$$= i_0(x,y)\, r_0(x,y).$$

Illuminatn & reflectance components of O/P processed image.



Histogram equilization

① 3-bit image ($L=8$) of size 64×64 pixels. ($MN = 4096$) has intensity distribution shown below in table 3.1, where intensity levels are integers in the range $[0, L-1] = [0; 7]$

r_k	n_k	$P_r(r_k) = n_k/MN$
$r_0 = 0$	790	$0.19 = 790/4096$
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

$L = 8$

$MN = 4096$

$$S_K = T(r_K) = (L-1) \sum_{j=0}^{K} P_r(r_j)$$

$$S_0 = T(r_0) = 7 \sum_{j=0}^{0} P_r(r_j) = 7\,P_r(r_0) = 7 \times 0.19$$
$$= 1.33$$

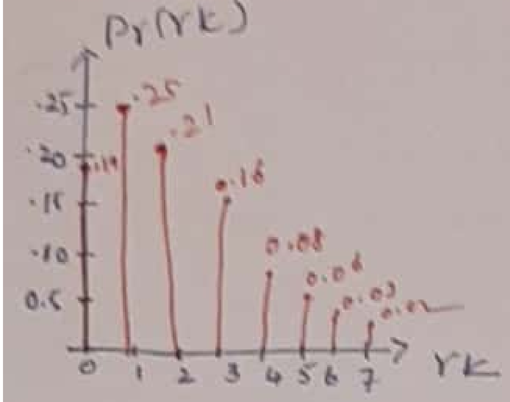
$$S_1 = T(r_1) = 7 \sum_{j=0}^{1} P_r(r_j) = 7\,P_r(r_0) + 7\,P_r(r_1)$$
$$= 7 \times 0.19 + 7 \times 0.25 = \underline{3.08}$$

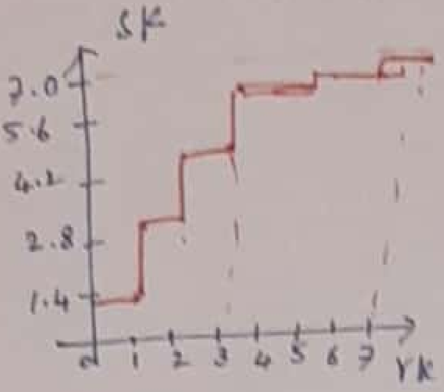
Like this solving we get

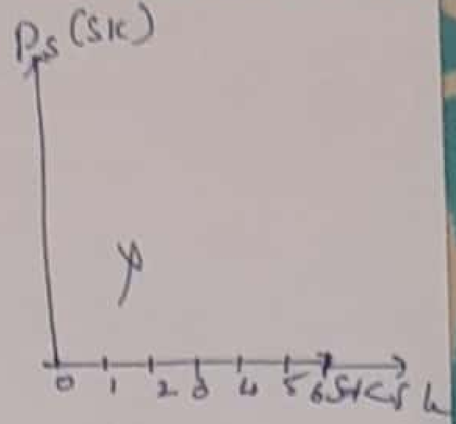
$S_2 = 4.55$, $S_3 = 5.67$, $S_4 = 6.23$, $S_5 = 6.65$,

S~~$_5 = 6.65$~~, $S_6 = 6.86$ & $S_7 = 7.00$

$S_0 = 1.33 \rightarrow 1$	$S_4 = 6.23 \rightarrow 6$
$S_1 = 3.08 \rightarrow 3$	$S_5 = 6.65 \rightarrow 7$
$S_2 = 4.55 \rightarrow 5$	$S_6 = 6.86 \rightarrow 7$
$S_3 = 5.67 \rightarrow 6$	$S_7 = 7.00 \rightarrow 7$







~~$r_0 = 0 \xrightarrow{\text{mapped to}} S_0 = 1$~~ , Th

$r_0 = 0$ is mapped to $s_0 = 1$ ∴ 790 pixels in histogram equalized image

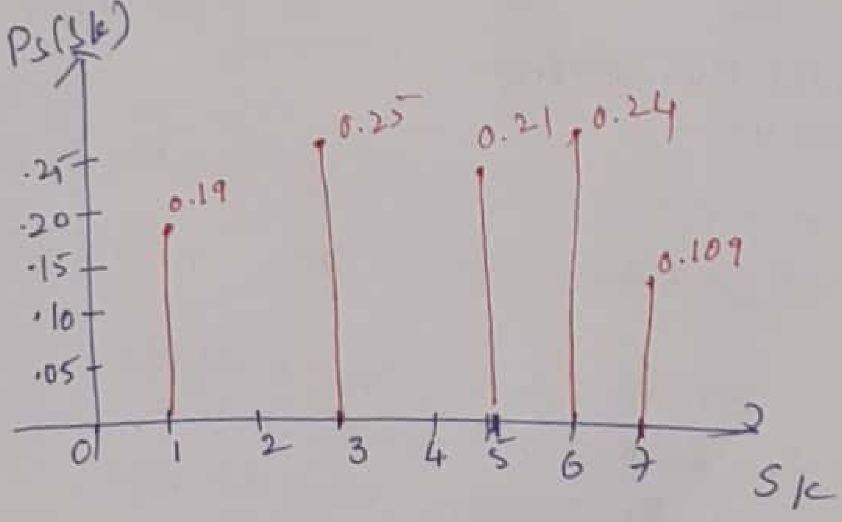
$r_1 = 1$,, ,, $s_1 = 3$ ∴ 1023

$r_2 = 2$ .... $s_2 = 5$ ∴ 850

r_3 & r_4 .... ~~so~~ 6 ∴ $(656 + 329) = 985$

lly r_5, r_6, r_7 --- to 7 ∴ $(245 + 122 + 81) = 448$

Sk	nk	Ps(Sk)
1	790	790/4096 = 0.19
3	1020	= 0.25
5	850	= 0.21
6	985	0.24
7	448	0.109



Histogram Matching (specification)

* Histogram equalization automatically determines a transformation function which produce an output image that has a uniform histogram.

* when automatic ~~etna~~ enhancement is desired, this is a good approach ∴ the results from this technique are predictable & the method is simple to implement.

* for some application, this might not be the best approach ~~of~~ base enhancement ~~on unif~~.

* for some times, we need to specify the shape of the histogram that we want to process the image

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

* Histogram specification is a point operation that maps input image $f(x,y)$ into an output image $g(x,y)$ with a user specified histogram

* Uses: * It improves constrast & brightness of images.

* It is a pre-processing step in comparison of images.

①

Let us recall Algorithm of histogram equalization

$P_r(r) \rightarrow$ pdf of grey level 'r' of input image

$P_z(z) \rightarrow$ pdf of grey level 'z' of specified image

$P_s(s) \rightarrow$ pdf of grey levels 's' of output image

The transformation is

* Histogram equalization of input image

$$S = T(r) = \underset{(L-1)}{[L-1]} \int_0^r P_r(r)\,dr \quad \text{———} \quad ①$$

* Histogram equalization of specified image

$$G(z) = \underset{(L-1)}{[L-1]} \int_0^z P_z(z)\,dz \quad \text{———} \quad ②$$

Then

$$G(z) = S = T(r)$$

$$\Rightarrow \quad z = G^{-1}[S] = G^{-1}[T(r)] \rightarrow ④$$

* Assuming that G^{-1} exists, then we can map i/p grey levels 'r' to o/p grey levels 's'.

Procedure for histogram specification

Step 1:- obtain the transformation T(r) by doing histogram equalization of i/p image

$$S = T(r) = (L-1) \int_0^r P_r(r)\,dr$$

Step2 :- Obtain the Transformation $G(z)$ by doing histogram equalization of specified image

$$G(z) = (L-1)\int_{0}^{z} P_z(z)\, dz$$

Step3 :- Equate $G(z) = s = T(r)$

Step4 :- Obtain inverse transformation function G^{-1}

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

Step5 :- Obtain the output image by applying inverse Transformation function to all pixels of input image.

① Assume an image having ~~gre~~ given grey level pdf $P_r(r)$. Apply histogram specification with given desired pdf function $P_z(z)$ given below

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

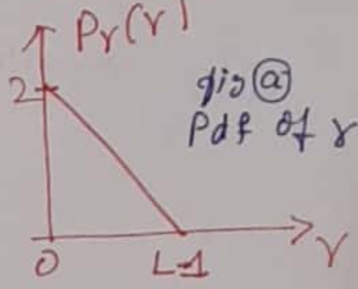


fig ⓐ Pdf of r

$$P_z(r) = \begin{cases} \frac{2z}{(L-1)} \; ; & 0 \leq z \leq L-1 \\ 0 \; ; & \text{otherwise} \end{cases}$$

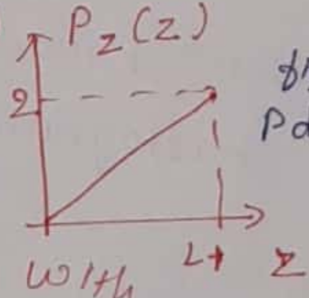


fig ⓑ Pdf of z

Apply histogram specification with the desired pdf function $P_z(z)$ given in fig ⓑ

②

1) Obtain transformation function T(r) by doing histogram equalization of Input Image

$$S = T(r) = \int_0^r P_r(r)\,dr = \int_0^r (-2r+2)\,dr$$

$$= \left[-r^2+2r\right]_0^r$$

$$= -r^2+2r.$$

2) Obtain transformation function G(z)

$$G(z) = \int_0^z P_z(z)\,dz = \int_0^z 2z\,dz$$

$$= \left. z^2 \right]_0^z = z^2$$

(3) Equate $S = T(r) = G(z)$

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

(4) obtain inverse transformation G^{-1}

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

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

//

Discrete formulation

Histogram equalization of i/p image

$$S_K = T(r_K) = \overset{(L-1)}{} \sum_{j=0}^{K} P_r(r_j) \quad , K=0, \ldots L-1$$

$$S_K = (L-1)\sum_{j=0}^{K} \frac{n_j}{n} \qquad \cancel{k} = \frac{(L-1)}{n=MN} \sum_{j=0}^{K} n_j$$

n = total no. of pixels in i/p image

n_j = no of pixels having level j

* ~~transformation fun G(z) can be~~ Ob~~tained using eq ②~~ given $P_z(z)$

---

② $$p_r(r) = \begin{cases} \frac{2r}{(L-1)^2} ; & 0 \le r \le L-1 \\ 0 ; & \text{other values of } r \end{cases}$$

$$P_z(z) = \begin{cases} \frac{3z^2}{(L-1)^3} ; & 0 \le z \le (L-1) \\ 0 ; & \text{other values of } z \end{cases}$$

find the transformation fun

(i) $$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = (L-1)\int_0^r \frac{2w}{(L-1)^2}\,dw$$

$$= \frac{2}{(L-1)}\int_0^r w\,dw = \frac{\cancel{2} r^2}{(L-1)}$$

(2) $$G(z) = (L-1)\int_0^z p_z(w)\,dw = \frac{3}{(L-1)^2}\int_0^2 w^2\,dw$$

$$= \frac{z^3}{(L-1)^2}$$

(3) $$G(z) = s$$

$$\frac{z^3}{(L-1)^2} = s$$

$$\therefore z = [(L-1)^2 s]^{1/3}$$

⑤

If we multiply ~~eac~~ every histogram equalized pixel by $(L-1)^2$ & raise the product to the power by 1/3, the result will be an image whose intensities z have the PDF $P_z(z) = \frac{3z^2}{(L-1)^3}$ in $(0, L-1)$

* since $S = \frac{r^2}{(L-1)}$.

$$Z = \left[(L-1)^2 \cdot \frac{r^2}{(L-1)} \right]^{1/3}$$

$$Z = \left[(L-1) r^2 \right]^{1/3} //$$

* squaring the value of each pixel in the original image & mutiplying the resut by $(L-1)$ & raising the product to the power $(1/3)$ will yield an image whose intensity levels z have the specified PDF.

Histogram equalization of specified image

$$v_q = G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i)\ ,\quad q = 0, \dots L-1;$$

equate $G(z_q) = s_k = T(r_k)$

inverse transformation

$$z_q = G^{-1}[s_k] = G^{-1}[T(r_k)]$$

this operation gives a value of z for each value of s [mapping to s to z)

Procedure for Histogram Specification

Step1 : Equalize input image histogram $[s_k]$

2 :. Equalize specified image histogram $[v_q]$

3 :. For $\min_q [v_q - s] \geq 0$ find corresponding v^* & p.

4 :. Map input pixels to o/p pixels to get output image.

① Apply histogram specification on image in fig below

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

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

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

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

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

③

1 :· Equalize input image histogram.

$P_r(r_i)$ given

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

$S_K = T(r_K)$
$= \sum_{j=0}^{k} \frac{n_j}{n}$

2 :· Equalize specified image histogram.

given → $P_z(z_0)$
$P_z(z_i)$

Z_q	0	1	2	3
$P_z(Z_q)$	0	0.5	0.5	0
V_q	0	0.5	1	1

3 :- Find minimum value of 'q' such that $(V_q - s) \geq 0$. first 3 columns are filled by step 1, next 3 columns are filled by step 2. In this step, last 2 column's are filled by foll procedure

1			2			3	
r_K	$P_r(r_K)$	S_K	Z_q	$P_z(Z_q)$	V_q	V^*	P
0	0.25	0.25	0	0	0	0.5	1
1	0.25	0.5	1	0.5	0.5	0.5	1
2	0.25	0.75	2	0.5	1	1	2
3	0.25	1	3	0	1	1	2

$V_q - S_K$ $\quad$ $V_0^* = V_q$ $\quad$ $P_0 = Z_q$

(a) $q = 0, K = 0 \quad (V_0 - S_0) = (0 - 0.25) = -0.25 \geq 0 \Rightarrow$ NO
increase q

$q = 1, K = 0 \quad (V_1 - S_0) = (0.5 - 0.25) = 0.25 \geq 0 \Rightarrow$ yes

$$\therefore \boxed{\begin{array}{l} V_0^* = V_q = V_1 = 0.5 \\ P_0 = Z_q = Z_1 = 1 \end{array}}$$

(b) $q = 1, K = 1 \quad (V_1 - S_1) = (0.5 - 0.5) = 0 \geq 0 \Rightarrow$ yes

$$\therefore \boxed{\begin{array}{l} V_1^* = V_1 = 0.5 \\ P_0 = Z_1 = 1 \end{array}}$$

(c) $q = 1, K = 2 \quad (V_1 - S_2) = (0-75) \geq -0.75 \geq 0 \Rightarrow$ NO
increase q

$q = 2, K = 2 \quad (V_2 - S_2) = \underset{= 0.25}{(1 - 0.75)} \geq 0 \Rightarrow$ yes

$$\therefore \boxed{\begin{array}{l} V_2^* = V_2 = 1 \\ P_2 = Z_2 = 2 \end{array}}$$

(d) $q = 2, K = 3 \quad (V_2 - S_3) = (1 - 0.75) = 0.4 \geq 0$ yes $\Rightarrow$

$$\boxed{\begin{array}{l} V_3^* = V_2 = 1 \\ P_3 = Z_2 = 2 \end{array}}$$

$\boxed{4}$ Map i/p levels to o/p levels value

rK	0	1	2	3
P	1	1	2	2

$\boxed{5}$ Map i/p pixels to new values to get new image

$$\underset{f(x,y)}{\begin{bmatrix} 0 & 1 & 0 & 2 \\ 2 & 3 & 3 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 3 & 2 & 0 \end{bmatrix}} \Rightarrow \underset{g(x,y)}{\begin{bmatrix} 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 \end{bmatrix}}$$

* Let $P_r(r)$ → Pdf of grey level 'r' of i/p image

$P_z(z)$ → pdf of grey level 'z' of specified image

r & z ⇒ intensity levels of i/p & o/p images resp.

* Transformation ~~of~~ palticular importance in image processing is given by

$$S = T(r) = (L-1)\int_0^r P_r(w)\,dw \rightarrow ①$$

(continuous version of histogram equalizn)

* Let us define a random Valiable 'z' with the property

$$G(z) = (L-1)\int_0^z P_z(t)\,dt = S \rightarrow ②$$

t → dummy valiable

Same as by ①

* from eq ① & ②

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

∴ 'z' must satisfy the condition

$$z = G^{-1}[T(r)] = G^{-1}(s) \rightarrow ③$$

* once $P_r(r)$ has been estimated from i/p image, then T(r) can be obtained by eq ①

consider 64x64 hypothetical image shown in previous example whose histogram ~~ear~~ is shown in below fig ⓐ It is desired to transform this histogram so that it will have the values specified in the second column of this Table 3.2 & fig ⓑ shows a sketch of this histogram.

$p_r(r_k) = n_k/MN$

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

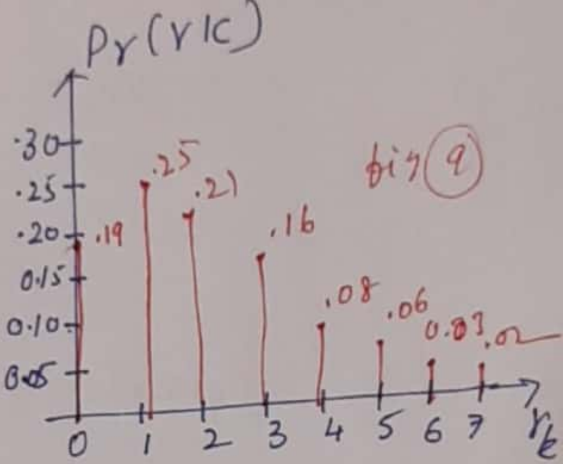
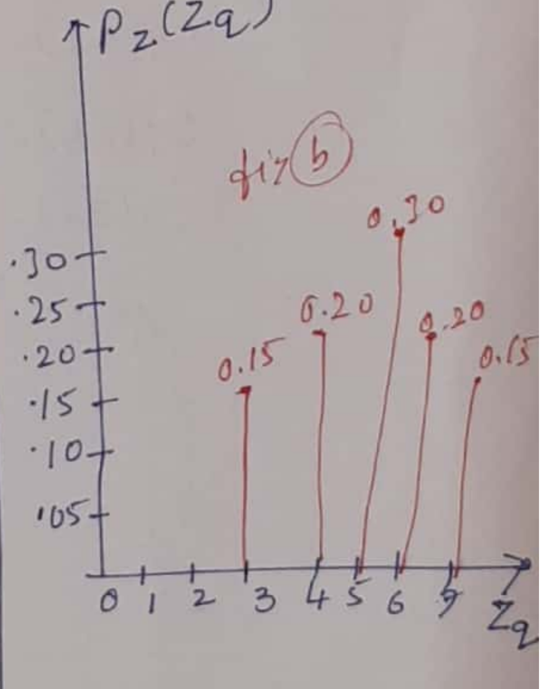


Table 3.2

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



⑤

I to obtain histogram-equalized value

$s_0 = 1$ $s_2 = 5$ $s_4 = 6$ $s_6 = 7$

$s_1 = 3$ $s_3 = 6$ $s_5 = 7$ $s_7 = 7$

II compute all the values of the transformatio
fun G using

$L=8$ $$G(z_q) = (L-1)\sum_{i=0}^{q} P_z(z_i)$$

$$G(z_0) = 7\sum_{j=0}^{0} P_z(z_i) =$$

$$= 7 \times P_z(z_0) = 0.000$$

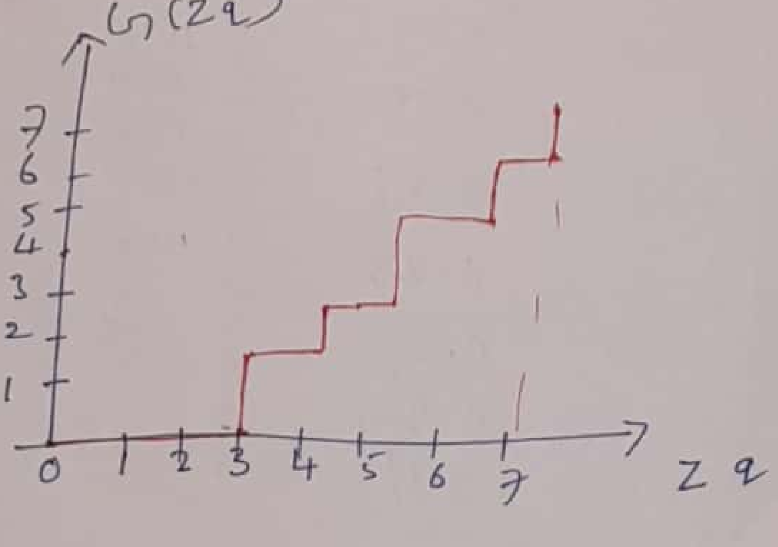
$$G(z_1) = 7\sum_{j=0}^{1} P_z(z_j) = 7\left[P(z_0) + P(z_1)\right] = 0.00$$

||ly

$G(z_2) = 0.00$ $G(z_3) = 1.05$

$G(z_4) = 2.45$ $G(z_5) = 4.55$

$G(z_6) = 5.95$ $G(z_7) = 7.00$



- these fractional values are converted into integer

$G(z_0) = 0.00 \rightarrow 0$

$G(z_1) = 0.00 \rightarrow 0$

$G(z_2) = 0.00 \rightarrow 0$

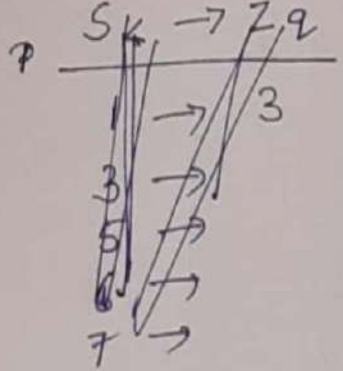
$G(z_3) = 1.05 \rightarrow 1$

$G(z_4) = 2.45 \rightarrow 2$

$G(z_5) = 4.55 \rightarrow 5$

$G(z_6) = 5.95 \rightarrow 6$

$G(z_7) = 7.00 \rightarrow 7$



Zq	G(Zq)
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

III we find smallest value of z_q so that the value $G(z_q)$ is ~~closest~~ closest to S_K.

eg① $S_0 = 1$ & we see $G(z_3) = 1$ which is perfect match in this case

∴ we have correspondence $\boxed{S_0 \rightarrow z_3}$

i.e, every pixel whose value is 1 in the histogram equalized image would map to a pixel valued 3 (in the corresponding location) in the histogram-specified image

Sk	→ Zq
1 →	3
3 →	4
5 →	5
6 →	6
7 →	7

$S_1 = 3$ $G(z_4) = 2$

∴ $S_1 \rightarrow z_4$

:

* To compute $p_z(z_q)$

$s = 1$ maps to $z = 3$

there are 790 pixels in the histogram-equalized image with a value of 1.

$\therefore P_z(z_3) = \frac{790}{4096} = 0.19$

$s = 3 \rightarrow z = 4$

$\therefore P_z(z_4) = \frac{1020}{4096}$
$= 0.25$

$s = 5 \rightarrow z = 5$

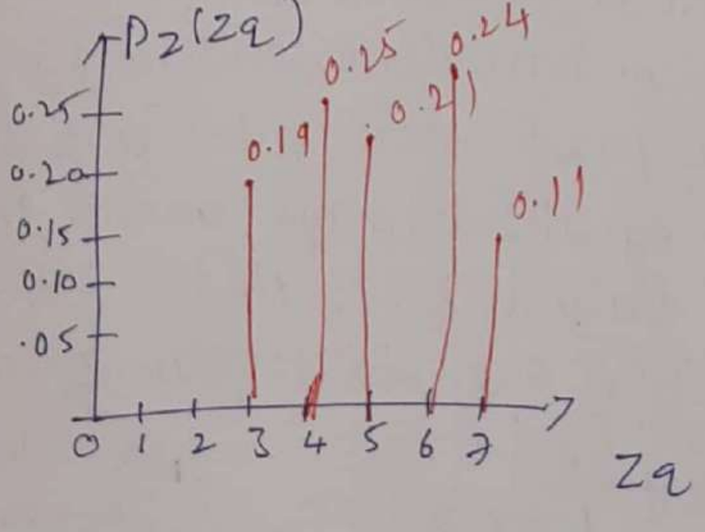
$P_z(z_5) = \frac{850}{4096}$
$= 0.21$

$s = 6 \rightarrow z = 6$

$P_z(z_6) = \frac{985}{4096}$
$= 0.24$

$s = 7 \rightarrow z = 7$

$P_z(z_7) = \frac{448}{4096}$
≈ 0.109
≈ 0.11



Local Histogram Processing

- The histogram process discussed before [histogram equilization & histogram specialization] are global

- In this approach, pixels are modified by a transformation function based on the intensity distribution of an entire image.

- Although this method is suitable for overall enhancement, there are some cases in which it is necessary to enhance details over small areas in an image

- The no of pixels in these areas may have negligible influence on the computation of a global transformation whose shape doesnot necessarily gualantee the desired local enhancement

- The solution is to devise transformation functions based on the intensity distribution in a neighborhood of every pixel in the image

- The procedure is to define a neighborhood and move its center from pixel to pixel

⑦

* At each location, the histogram equalization or histogram specification transformation function is obtained.

* This fun is then used to map the intensity of the pixel centered in the neighborhood

* The centre of the neighborhood region is then moved to an adjacent pixel location & the procedure is repeated

* ∴ only one row or column of the neighborhood changes during a pixel-to-pixel translation of the neighborhood, updating the histogram obtained in previous location with the new data introduced at each motion step is possible

* ~~to~~ this approach has <u>Advantages</u> over ~~repeade~~ repeatedly computing the histogram of all pixels in the neighborhood region each time the region is moved one pixel location

* one more approach used sometimes ~~is~~ to reduce computation is to utilize non-overlapping regions but this method usually produces an undesirable "blocky" effect

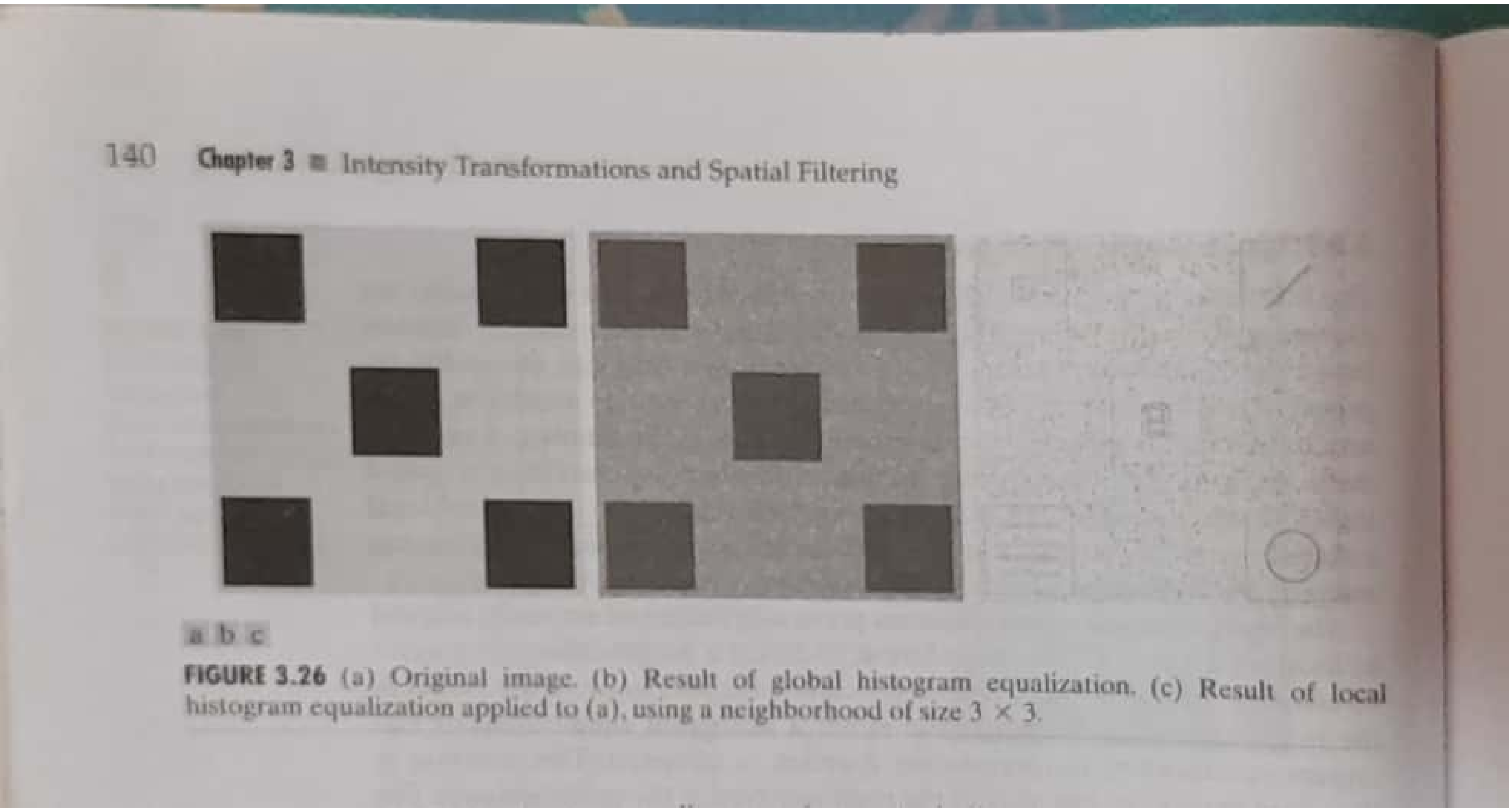


FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3 × 3.

Using Histogram Statistics for Image Enhancement/.

* Statistics obtained directly from an image histogram can be used for image enhancement

* Let 'r' denote ⟹ discrete random variable representing intensity values in the range [0, L-1]

$p(r_i)$ ⟹ the normalized histogram component corresponding to value r_i (or) an estimate of probability that intensity r_i occurs in the image from which the histogram was obtained

* nth moment of 'r' about its mean is defined as

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i) \longrightarrow ①$$

where

m = mean value (average intensity of pixels in the image)

$$m = \sum_{i=0}^{L-1} r_i p(r_i) \longrightarrow ②$$

* The second moment is particularly important & is defined as

$$\mu_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i) \longrightarrow ③$$

⑧

- eq ③ is recognized as intensity variance denoted by σ^2

mean → measure of average intensity in an image

variance ⇒ measure of contrast in an image.
(or std. deviation)

std. dev = squareroot of variance

* once the histogram is computed for an image, all the moments are easily computed using eq ①

* when mean & variance are computed directly from the sample values, without computing the histogram [common practice] then these estimates are called as Sample mean & Sample variance

* $$m = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \rightarrow ④$$

&

$$\sigma^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - m]^2 \rightarrow ⑤$$

* sometimes instead of MN even MN-1 can be used ~~fo~~ is done to obtain (unbiased estimate of variance)

eg consider 2-bit image size 5x5

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

the pixels are represented by 2-bits

- pixels are represented by 2 bits
  ∴ $L =$
- The intensity levels are in the range
  $[0, \quad]$ ↑ $L-1$
- $MN =$
- histogram has the components
  $p(r_i) \Rightarrow$ compute
- compute average value of intensities in the image

* sample mean value ~~&~~ ?

---

* uses of mean & variance for enhancement purpose

* The global mean & variance are computed & are useful for gross adjustments in overall intensity & contrast.

* use of these parameters in local enhancement

* Local mean & variance are used as basis for making changes that depend on image characteristics in a neighbourhood about each pixel in an image

* Let (x, y) ⟹ co-ordinates of any pixel in a given image

$S_{x,y}$ ⟹ neighborhood (subimage) of specified size, centered on (x, y).

* mean value of the pixels in this neighborhood is

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i p_{S_{xy}}(r_i) \longrightarrow ⑥$$

$p_{S_{xy}} \Rightarrow$ histogram of pixels in region S_{xy}.

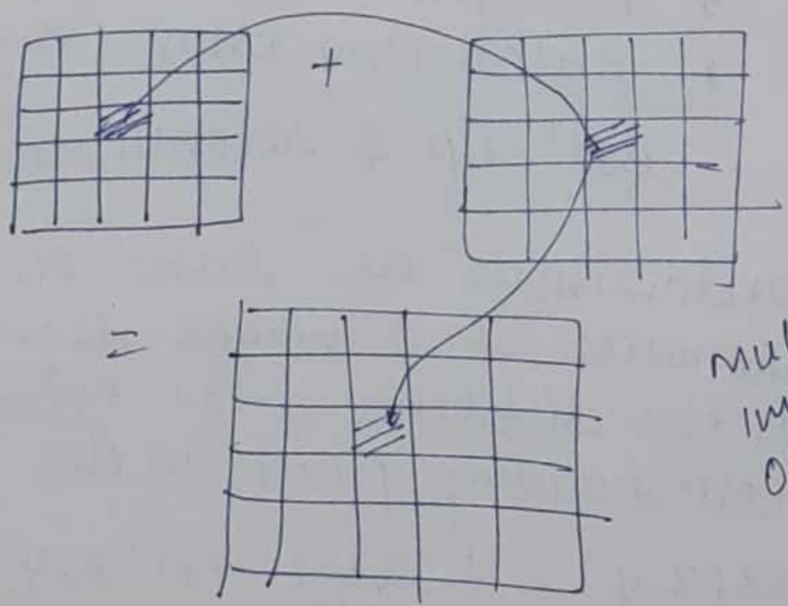
* variance of pixels in the neighborhood

$$\sigma^2_{S_{xy}} = \sum_{i=0}^{L-1} (r_i - m_{S_{xy}})^2 p_{S_{xy}}(r_i) \hookrightarrow ⑧$$

* Local mean ⇒ is a measure of avg intensity in neighborhood S_{xy}

* Local variance ⇒ is a measure of intensity constrast in the neighborhood

Arithmetic | Logic operations



multi image operation

* In multi image operation, grey levels of 2 or more input images are mapped to a single o/p image as shown in above fig

* $g(x,y) = op [f_1(x,y), f_2(x,y)]$

f_1 & f_2 → i/p images

g → o/p ,,

op → operator which is applied pair wise to each pixel in the image

* operations :- are addition, multiplication, subtraction [Arithmetic] and Logical [AND, OR XOR. etc]

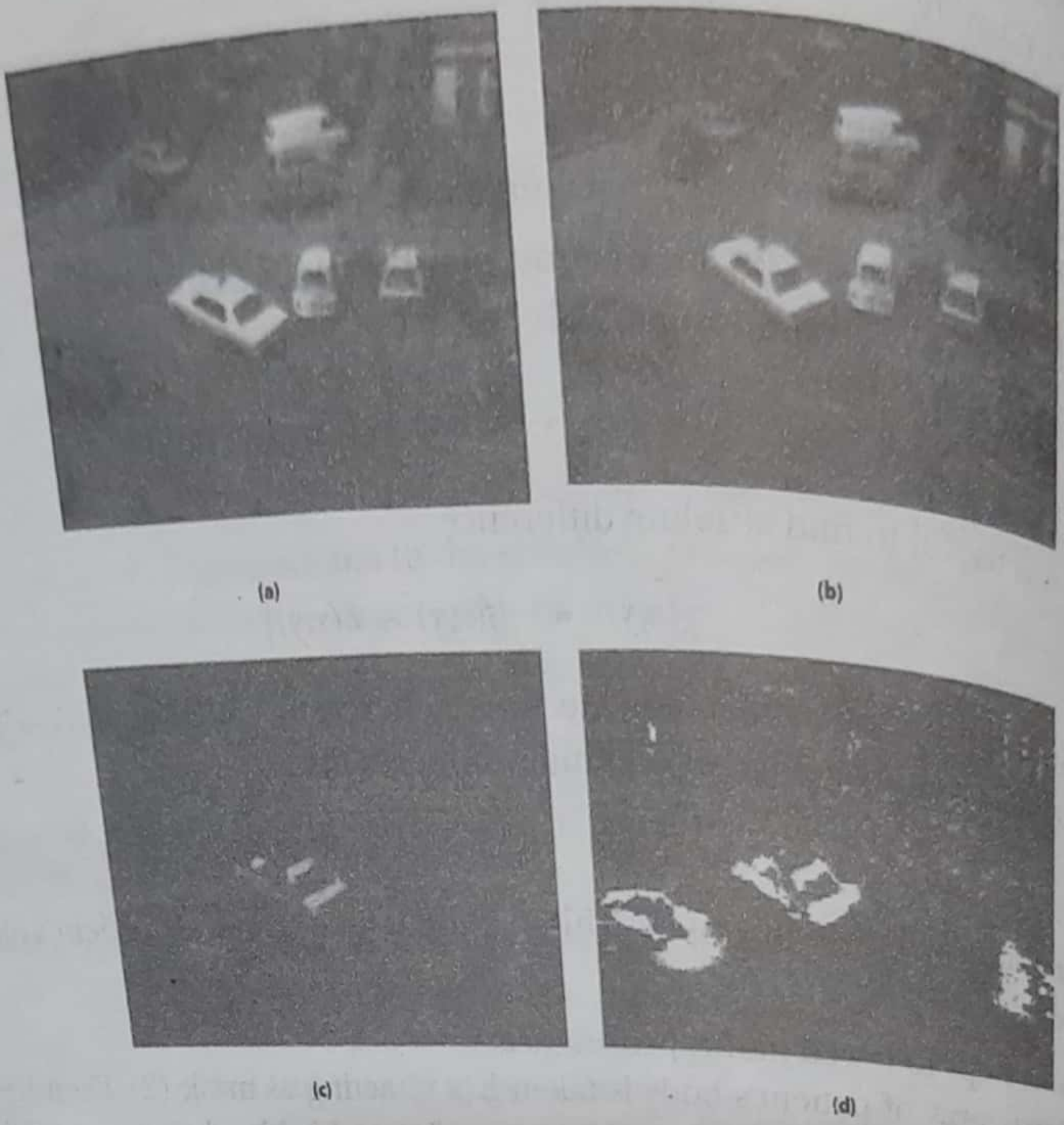
① Image subtraction

Applns:- Image subtraction has numerous applications in image enchancement & segmentation namely

* motion detection
* Background illumination
* calculating error (mean square error) bet' i/p & reconstructed image

* fundamentals are based on ~~subtracting~~ subtraction of 2 images defined on the difference bet' every pair of corresponding pixels in the 2 images

$g(x,y) = f(x,y) - h(x,y)$ — ①



(a) (b)

(c) (d)

FIGURE 3.43: Motion detection: fig (a) and (b) are subtracted to get difference image (c). Fi (c) is thresholded to generate binary image (d).

Sometimes we can find absolute difference

$$g(x,y) = |f(x,y) - h(x,y)| \rightarrow ②$$

Appln! . interesting appln is in medicine where $h(x,y) \Rightarrow$ mask which is subtracted from series images to get very interesting results

① Digital Subtraction Angiography

$h(x,y) \Rightarrow$ x-ray of patient's body

$f(x,y) \Rightarrow$ another x-ray which is obtained by injecting radio opaque dye which spreads into his blood steam

$g(x,y) = f(x,y) - h(x,y) \Rightarrow$ contains only bloodvessel

used to extract patient's blood carrying vessels

② motion detection

③ vide compression — to encode only the differences bet' frames.

④ Automatic checking of industrial parts

(17)

② Image Addition

- to create a double exposure or composites
- $g(x,y) = f_1(x,y) + f_2(x,y)$
- A weighted blend can also be done

$$g(x,y) = \lambda_1 f_1(x,y) + \lambda_2 f_2(x,y)$$

- Image averaging /. to average multiple images of the same scene to reduce noise; eg single image of electron microscope can be very noisy. . one way to reduce such kind of noise is to acquire multiple images of the same scene for long duration & then perform image averaging

$$\underset{\substack{\downarrow \\ \text{avg image}}}{\bar{g}(x,y)} = \frac{1}{n} \sum_{i=1}^{n} f_i(x,y)$$

avg image & $f_1 \ldots f_n$ are n acquired images

$$f(x,y) = \underset{\substack{\downarrow \\ \text{noiseless image}}}{f_{in}(x,y)} + \underset{\substack{\downarrow \\ \text{noise}}}{\eta(x,y)}$$

if n is high. The avg image is closer to noiseless image if no of image is high.

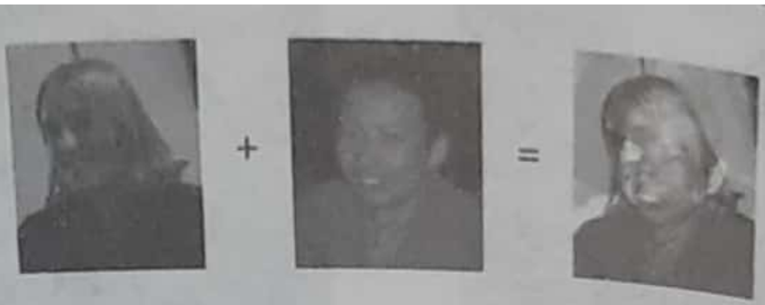


FIGURE 3.45: (a) Image addition

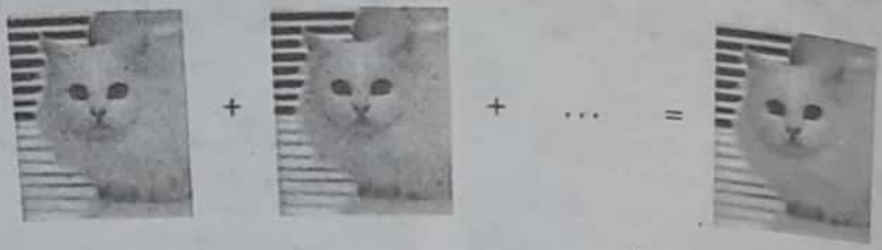


FIGURE 3.45: (b) Image averaging

in fig (d).

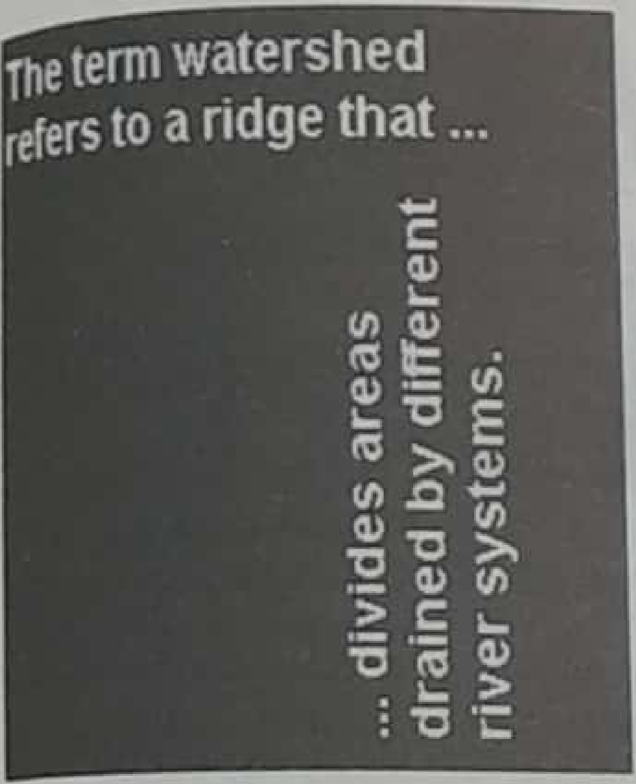


FIGURE 3.46: (a) Input image 1

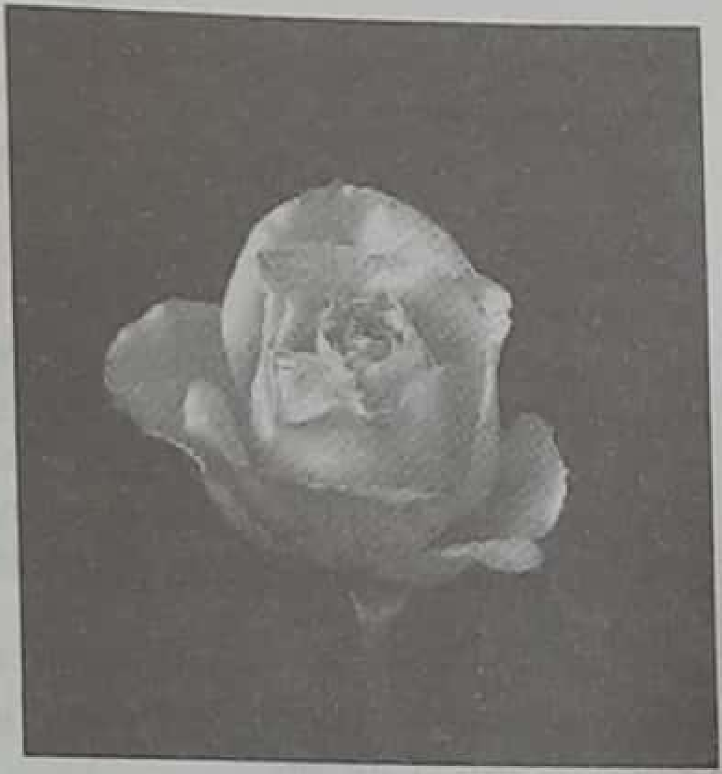


FIGURE 3.46: (b) Input image 2

FIGURE 3.46: (c) Output of image addition

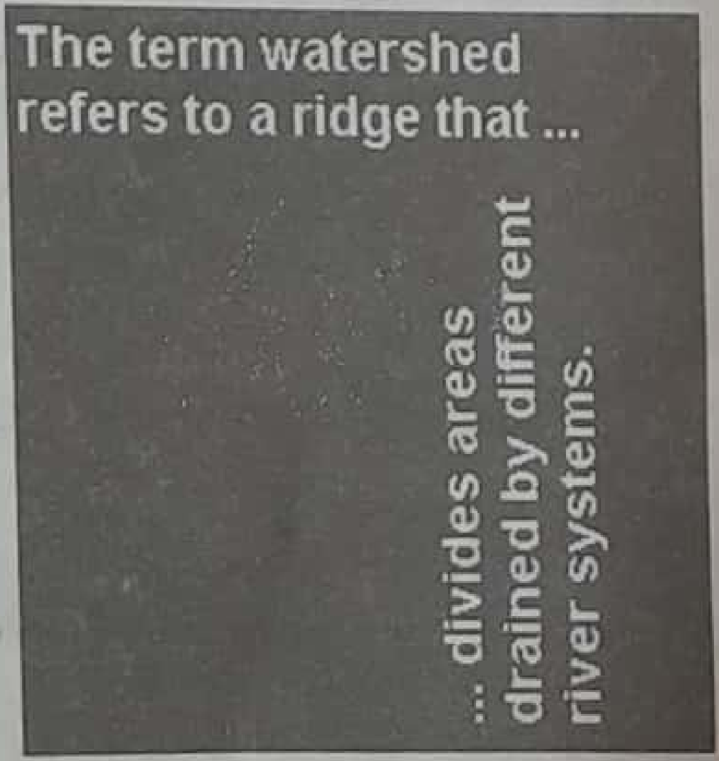


FIGURE 3.46: (d) Output of ex 3.10

Boolean operations

* If binary images need to be combined/operated, we can use boolean operation.

* Adv/- can be carried out relatively fast on computer

* Boolean operations are used for masking

* mask can be ANDed / Ored with i/p image to extract region of interest

*

* Logical operations are also used in Image quantization when 8 bit inforn has to be reduced to 5/4 bit

NOT(OR (I_1, I_2)) shows invisible area.

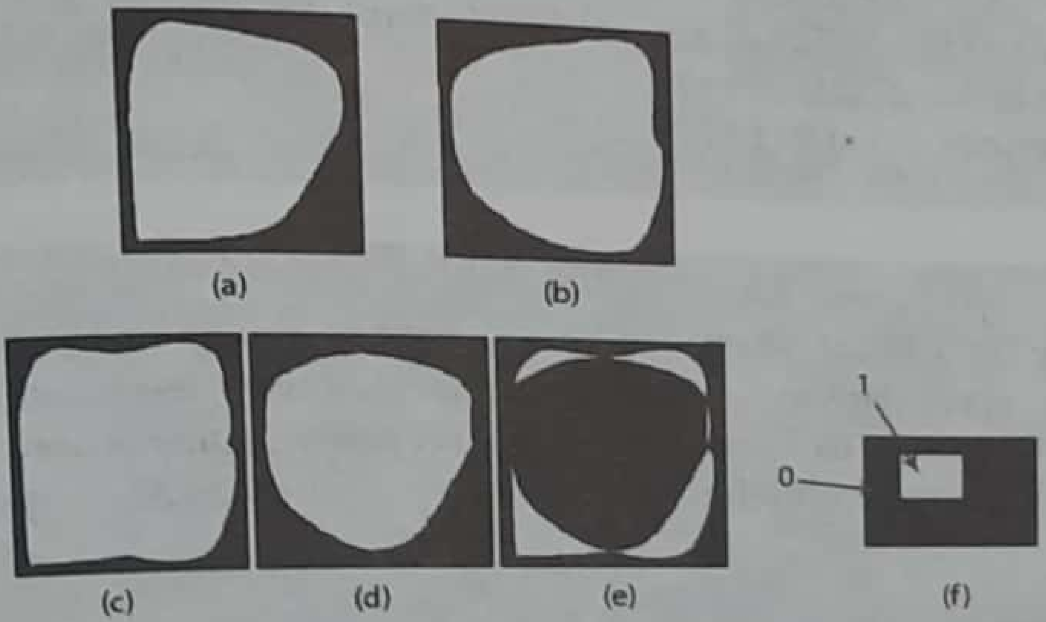


FIGURE 3.47: Boolean operations

Note

Bit wise AND operation is also used in matlab ex 3.6 to extract various bit planes from the image.

FIGURE 3.48: (a) Input image

FIGURE 3.48: (b)

FIGURE 3.48: (c)

FIGURE 3.48: (d)

Fundamentals of spatial filtering

- * spatial filtering is one of the principal tool used in DIP for a broad spectrum of applications. eg:· noise removal, bridging the gaps in object boundaries, Sharping of edges etc.

- * filtering refers to passing (accepting) or rejecting certain frequency components

* spatial filtering involves passing a weighted mask, or kernel over the image and replacing the original image pixel value corresponding to the centre of the kernel with the sum of the original pixel values in the region corresponding to the kernel multiplied by the kernel weights

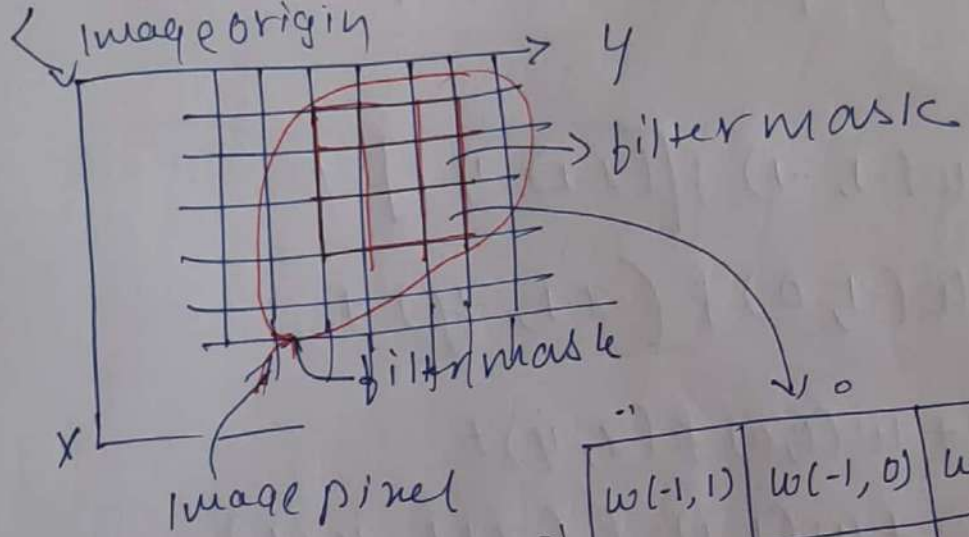
Mechanics of spatial filtering

* spatial filter consists of
  (i) a neighborhood (typically a small rectangle)
  & (ii) a pre-defined operation that is performed on the image pixels encompassed by the neighborhood

* filtering creates a new pixel with co-ordinates equal to the coordinates of the center of the neighborhood & whose values is the result of the filtering operation
* A processed (filtered) image is generated as the center of the filter visits each pixel in the i/p image

* If the operation performed on the image pixels is linear, then the filter is called linear spatial filter. otherwise the filter is non-linear

Linear spatial filtering



	-1	0	1
-1	w(-1,1)	w(-1,0)	w(-1,1)
0	w(0,-1)	w(0,0)	w(0,1)
1	w(1,-1)	w(1,0)	w(1,1)

filter coefficients

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

pixels of image section under filter

x

- fig illustrates the mechanics of linear spatial filtering using 3x3 neighborhood.
- 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 & the image pixels encompassed by the filter

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

- center coefficient of the filter w(0,0) aligns with the pixel at location (x, y)
- for a mask of size m×n, we assume that $m = 2a+1$ & $n = 2b+1$, where a & b +ve integer
- our focus is on filters of odd size

  3x3 ⇒ smallest size ~~being of~~
- linear spatial filter of image size M×N with a filter of size m×n

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

Apply given 3x3 mask "w" of fig ⓐ on the given image f(x,y) defined as

5	1	2	6	7
4	4	7	5	8
2	6	20	6	7
3	1	2	4	5
10	2	1	2	3

f(x,y)

$$\frac{1}{9} \times \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}_{w} \quad (b)$$

i/p image size = 5x5

Sol^n

1.

5	1	2	6	7
4	④	7	5	8
2	6	20	6	7
3	1	2	4	5
10	2	1	2	3

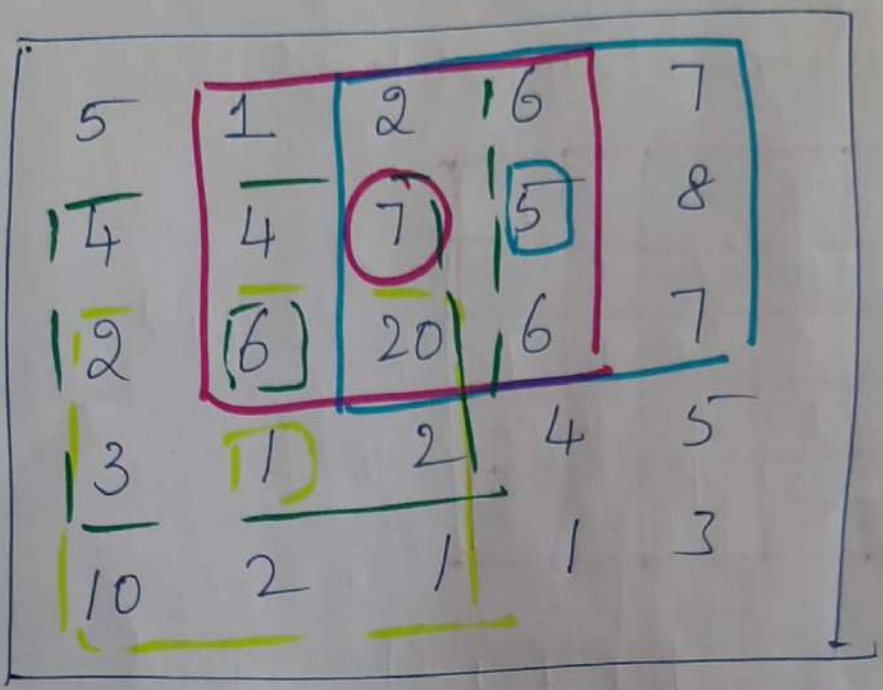
o/p =

$$\frac{1}{9} \times [5\times1 + 1\times1 + 2\times1 + 4\times1 + 4\times1 + 7\times1 + 2\times1 + 6\times1 + 20\times1]$$

$$= \frac{49}{9} \simeq \underline{\underline{5}}$$

Replace grey level value

$\boxed{4 \text{ by } 5}$



5	1	2	6	7
4	4	7	5	8
2	6	20	6	7
3	1	2	4	5
10	2	1	1	3

o/p

*	*	*	*	*
*	5	6	8	*
*	5	6	7	*
*	4	5	6	*
*	*	*	*	*

② o/p = $\frac{1}{9}$ [1×1 + 2×1 + 6×1 + 4×1 + 7×1 + 5×1 + 6×1 + 20×1 + 6×1]

= $\frac{57}{9} \simeq 6$ 7 → 6

③ o/p : $\frac{1}{9}$ [2×1 + 6×1 + 7×1 + 7×1 + 5×1 + 8×1 + 20×1 + 6×1 + 7×1]

= $\frac{68}{9} \simeq 8$ 5 → 8

④ o/p : $\frac{1}{9}$ [4 + 4 + 7 + 2 + 6 + 20 + 3 + 1 + 2] = $\frac{49}{9}$ = 5

5. o/p = $\frac{1}{9}$ [4 + 7 + 5 + 6 + 20 + 6 + 1 + 2 + 4] = $\frac{55}{9} \simeq 6$

6. o/p : $\frac{1}{9}$ [7 + 5 + 8 + 20 + 6 + 7 + 2 + 4 + 5] = $\frac{64}{9}$ = 7

7. o/p : $\frac{1}{9}$ [2 + 6 + 20 + 3 + 1 + 2 + 10 + 2 + 1] : $\frac{38}{9}$ = 4

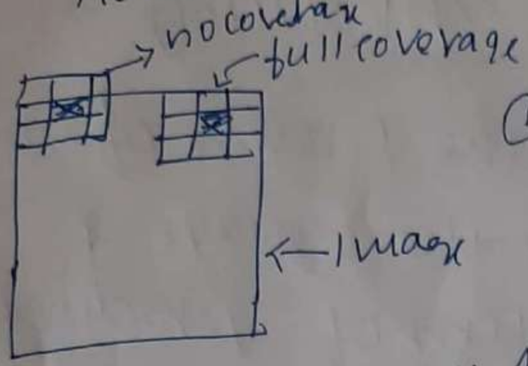
8. o/p : $\frac{1}{9}$ [6 + 20 + 6 + 1 + 2 + 4 + 2 + 1 + 1] : $\frac{44}{9}$ = 5

9. o/p = $\frac{1}{9}$ [20 + 6 + 7 + 2 + 4 + 5 + 1 + 1 + 3] : $\frac{59}{9} \simeq 6$

Spatial Correlation & Convolution

Note: Handling images Borders

① ignoring edges ② padding ③ mirroring



① ignoring edges: → * Apply the mask to only those pixels in the image for which the mask lies fully with the image

* Mask is applied to all pixels in the image except for edges [o/p image is smaller than that of i/p inform]

② padding:

* in this case, the i/p image is padded with zeros at the border.
* This uses the size of i/p image before applying filter

0 0 0 0 0 0 0
0 5 1 2 6 7 0
0 4 4 7 5 8 0
0 2 6 9 30 7 0
0 3 2 1 4 5 0
0 1 2 2 2 2 0
0 0 0 0 0 0 0

③ mirroring:

* mirror image of the known image is created with the border

5 5 1 2 6 7 7
5 5 1 2 6 7 7
4 4 4 7 5 8 8
2 2 6 20 6 7 7
3 3 1 24 5 5
1 1 2 1 2 3 3
1 1 2 1 2 3 3

* copy 1st & last row & column.

Linear spatial filtelling

① Correlation ② convolution

① correlation !- is the process of moving a filter mask over the image & computing the sum of products at each location. [as explained in Linear spatial filtelling]

② convolution:- the mechanism is same except the filter is first rotated by 180°

* Let us explain the above concept using 1-D illustration.

correlation

(a) origin f
0 0 0 1 0 0 0 0

w
1 2 3 2 8

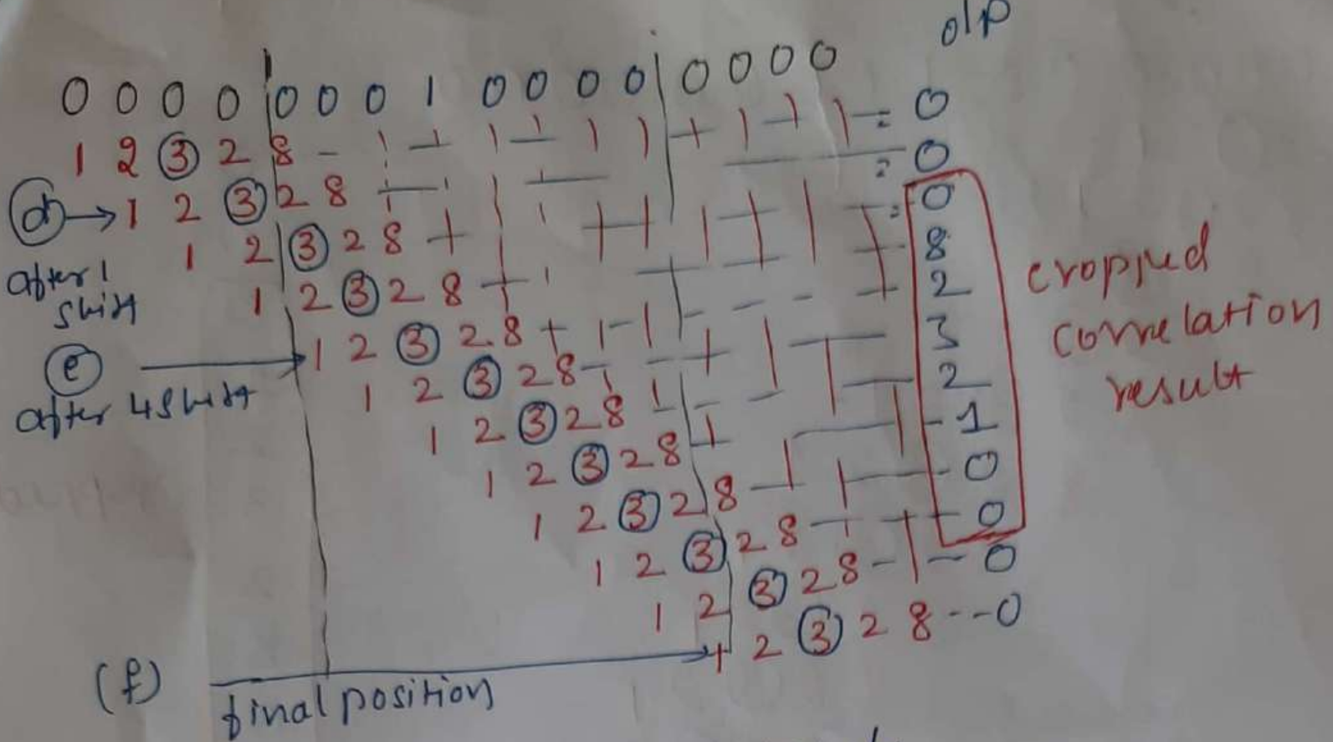
Length of image = 8
length of filter (size) = m = 5

(b) 0 0 0 1 0 0 0 0
1 2 3 2 8
starting position alignment

zero padding

(c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8

(m-1) 0's are padded on ~~other~~ either side of 'f'



(g) Full correlation result

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

(h) cropped correlation result
(the size should be same as f)

0 8 2 3 2 1 0 0

Convolution

f w rotated 180°

(a) 0 0 0 1 0 0 0 0 8 2 3 2 1

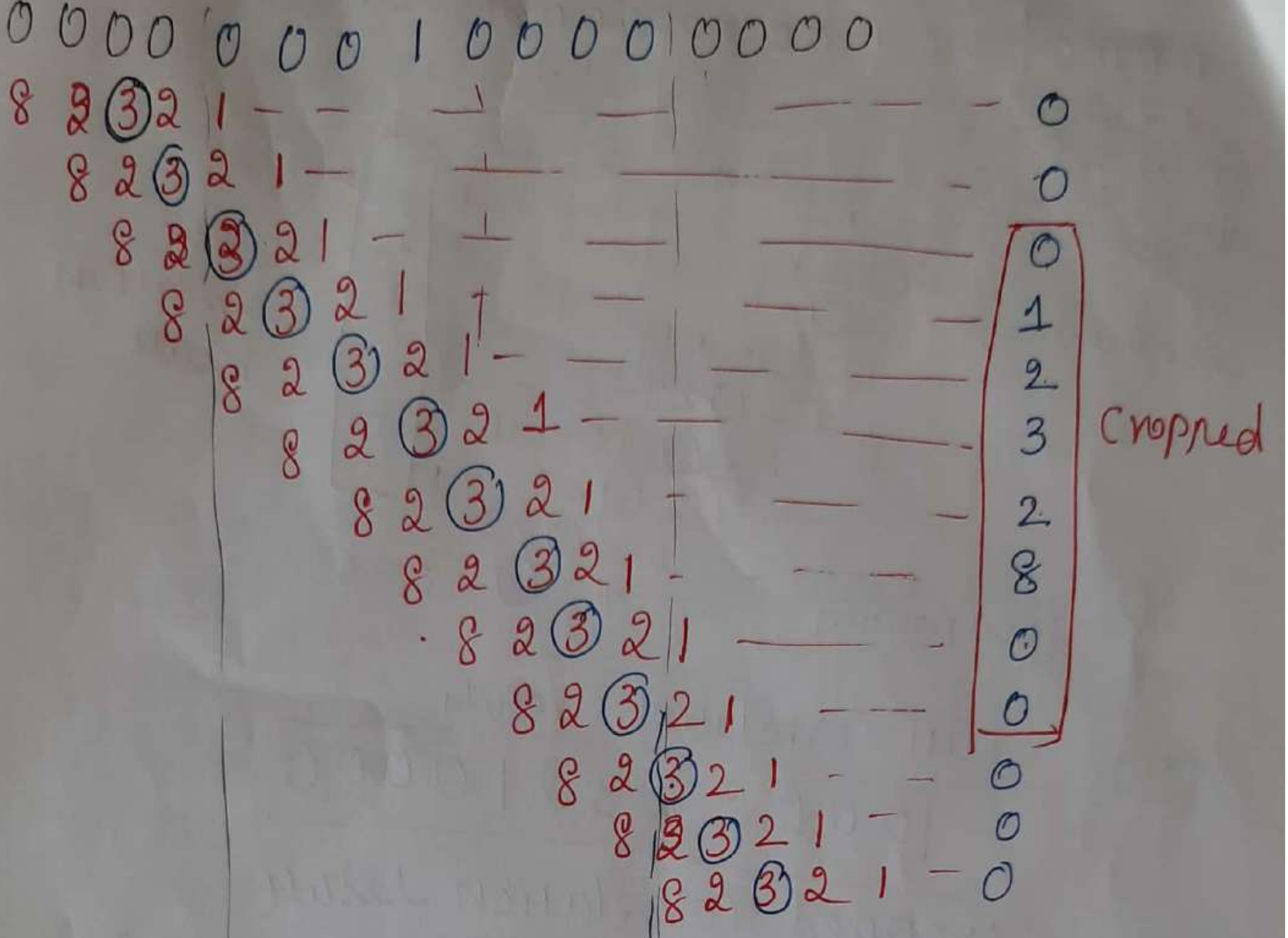
(b) 0 0 0 1 0 0 0 0
8 2 3 2 1

(c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 zero padding
8 3 2 1

(g) Full convolution result
0 0 0 1 2 3 2 8 0 0 0 0

(e) 0 1 2 3 2 8 0 0 → cropped convolution result

2



* Two important points to note from the above discussion.

① * correlation is a function of displacement of the filter.

* 1st value of correlation corresponds to zero displacement of the filter

* 2nd corresponds to one unit displacement & so on . .

② The correlating a filter 'w' with a function that contains all 0's & a single '1' yields a result that is copy of 'w' but roated by 180°

* correlation of a function with a discrete unit impulse yields a rotated version of the function at the location of the impulse

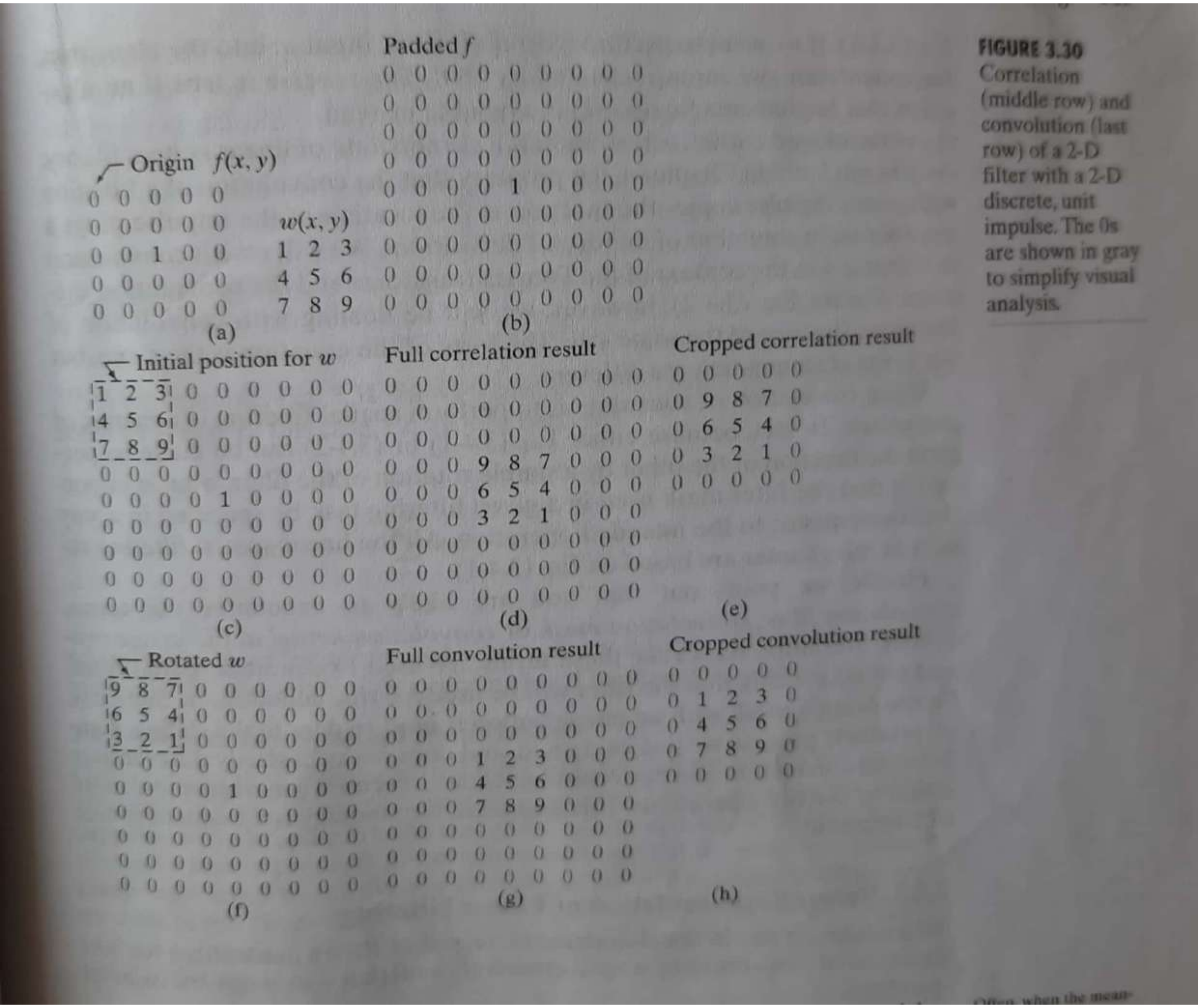
③ convoluting a fun with a unit impulse yields a copy of the function at the location of the impulse

* C~~onvolution~~
* correlation yields a copy of the function also but roated by 180°. ∴ If we pre-rotate the filter & perform the same sliding sum of products, we will obtain desired result

* for images, ~~these~~ the same concepts ~~extende~~d can be applied
* for filter of size m×n, we pad the image with a minimum of m-1 rows of 0's at the top & bottom and n-1 columns of 0's on the left & right

* convolution is cornerstone of a linear system theory

③



Origin $f(x, y)$

$$\begin{matrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{matrix} \qquad w(x, y) \quad \begin{matrix} 1&2&3\\ 4&5&6\\ 7&8&9 \end{matrix}$$

(a)

Padded f

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

(b)

Initial position for w

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

(c)

Full correlation result

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

(d)

Cropped correlation result

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

(e)

Rotated w

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

(f)

Full convolution result

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

(g)

Cropped convolution result

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

(h)

FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Often, when the mean-

- filter → $w(x,y)$ of size m×n,
  image → $f(x,y)$

* correlation of a filter & image

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

* convolution

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

where - sign ⇒ right flip (rotate it by 180°)

Vector Representation of Linear filtering

R ⇒ characteristic response of a mask of either correlation or convolution

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

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

w_k → coefficients of an m×n filter

z_k → corresponding image intensities encompassed by filter

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_8

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

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

Generating spatial filter masks

* Generating an mxn linear spatial filter requires in specifying mn mask coefficients.

* coefficients are selected based on the filter type.

* for example, we want to replace the pixels in an image by the average intensity of a 3x3 neighborhood centered on those pixels..

* Then the average value at any location (x,y) in the image is the sum of the nine intensity values in the 3x3 neighborhood centered on (x,y) divided by 9.

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

* in some applications, we have continuous function of 2 variables & the objective is to obtain a spatial filter mask based on that function.

eg a Gaussian fun of 2 variables has the basic form

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

where σ = std. deviation

x, y = are integers

to generate 3x3 mask from this fun, we sample it about its center

Generating spatial filter contd

* Generating a non-linear filter requires ~~to specifyinthe~~

(i) specifying the size of a neighborhood

& (ii) operation(s) to be performed on the image pixels contained in the neighborhood

* Nonlinear filters are quite powerful & in some applications they can perform functions that are beyond the capabilities of linear filter

* eg. 5x5 maximum filter [which performs max operation] centered at an arbitrary point (x,y) of an image obtains the maximum intensity value of the 25 pixels & assign that value to location (x, y) in the processed image

smoothing spatial filters

* smoothing filters are used for blurring & for noise reduction

* Blurring is used in preprocessing tasks, such as removal of small details from an image prior to (larger) object extraction & bridging of small gaps in lines or ~~cou~~ curves.

* Noise reduction can be accomplished by blurring with a linear filter & also non linear filter

smoothing Linear filters

- The output (response) of a smoothing linear Spatialfilter is simply the average of the pixels contained in the neighborhood of the filter mask.
- These filters are called as averaging filter or Low-pass filter or Mean filter
- In smoothing filters, the value of every pixel in an image is replaced by the average of the intensity levels in the neighborhood defined by the filter mask.
- This process results in an image with reduced sharp transitions in intensities.
- Random noise typically consists of sharp transitions in intensity levels. ∴ most obvious application of smoothing is noise reduction.
- ~~For~~ The edges (which almost always are desirable features of an image) are characterised by sharp intensity transition.
- So averaging filters have undesirable side effect that they blur edges.
- Another application of this type of process includes the smoothing of false contours which results from using insufficient no of intensity levels

* A major use of averaging filter is in the reduction of irrelevant detail in an Image.

[irrelevant ⇒ pixel regions that are small WRT to the size of the filter mask]

3 X 3 smoothing average filter

*

$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

ⓐ box filter ⓑ Weighted average

*ⓘ use of this filter (1st one) yields the standard average of the pixels under the mask $R = \sum_{k=1}^{9} w_k z_k$

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

↓

⇒ average of the intensity levels of

* the pixels in the 3x3 neighborhood defined by the mask

* The coefficients of the filter are all 1's
* The idea here is that it is computationally more efficient to have coefficients valued 1.
* At the end of filtering process, the entire image is ÷ by 9.

* An $m \times n$ Mask would have a normalizing constant equal to $\frac{1}{mn}$.

* A spatial averaging filter in which all coefficients are equal sometimes is called as box-filter

②

* The second type is shown in fig ⓑ above is called as weighted average, in which the pixels are multiplied by different coefficients of filter mask, there by giving more importance (weight) to some pixels at the expenses of other.

* In the filter mask shown, ~~above~~

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

(i) the pixel at the center of the mask is multiplied by a higher value than any other thus giving this pixel more importance in the calculation of the average.

(ii) The other pixels are inversely weighted as a function of their distance from the center of the mask

(iii) The diagonal terms are further away from the center than the orthogonal neighbors (by a factor of $\sqrt{2}$) & are weighted less than the immediate neighbors of the centre pixel

9

* The basic strategy behind weighing the center point the highest & then reducing the value of the coefficients as a function of increasing distance from the origin is to simply an attempt to reduce blurring in the smoothing process

* The general implementation for filtering an M×N image with a weighted averaging filter of size m×n (m & n are odd) is given by

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

* The denominator is the sum of the mask coefficients & ∴ it is a constant that needs to be computed only once

* Appⁿ of spatial averaging is to blur an image for the purpose of getting a gross representation of objects of interest i.e, intensity of smaller objects blends with black ground & larger objects become bloblike & easy to detect

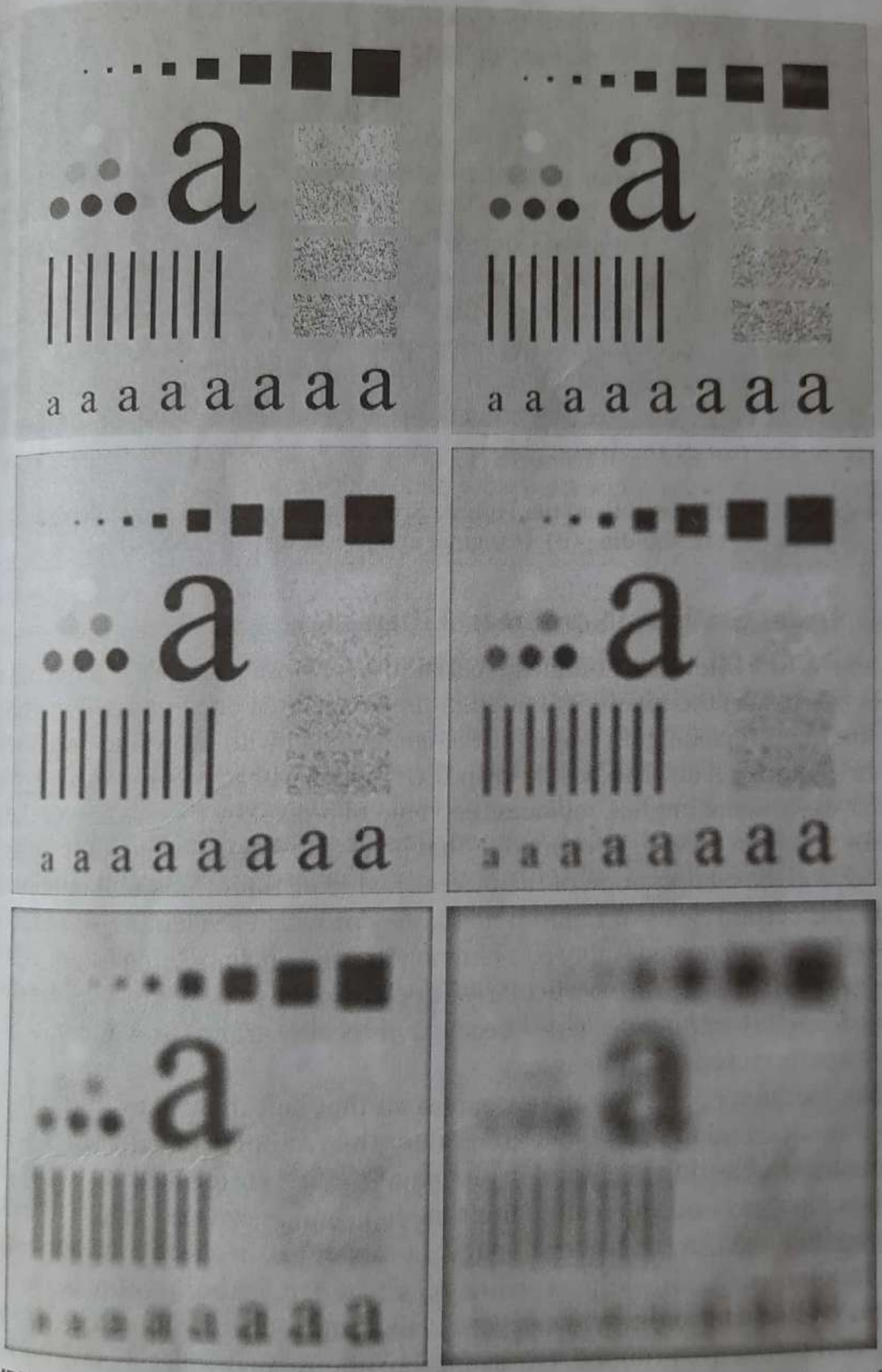
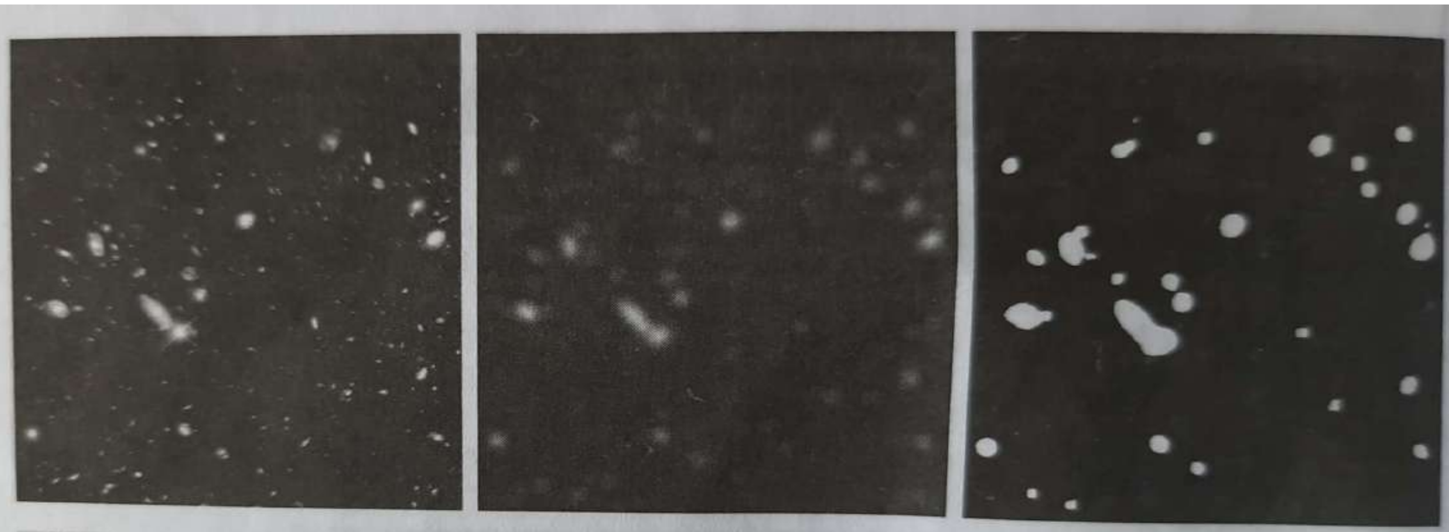


FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

a b
c d
e f



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

3.5.2 Order-Statistic (Nonlinear) Filters

Order- statistic (Non-linear) filters

- order-statistic filter are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter

- and replacing the value of the center pixel with the value determined by the ranking result

- The best-known filter in this category is median filter

- In median filter, the value of a pixel is replaced by the median of the the intensity values in the neighborhood of that pixel.

- Median filters are quite popular because
  - for random noise, they provide excellent noise-reduction capabilities with less blurring than linear smoothing filters
  - are effective in the presence of impulse noise, also called as salt and pepper noise [appearance as white & black dots superimposed on an image]

$$\left.\begin{array}{l}\text{* The median } \xi \\ \text{of aset of} \\ \text{valuy}\end{array}\right\} = \frac{1}{2} \text{ values in the set} \leq \xi$$

---

* The median, ξ of a set of values is such that half the values in the set are less than or equalto ξ & half are greater than or equalue to ξ

* to Perform median filtering at a point in an image

(i) we sort the values of the pixel in the neighborhood

(ii) determine their median

2(iii) assign that value to the corresponding pixel in the filtered image

* eg. in a 3X3 neighborhood, the median is 5th largest value

in a 5x5 neighborhood, it is the 13th largest value & so on

* suppose a 3X3 neighborhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100).

- values are sorted as [10, 15, 20, 20, 20, 20, 20, 25, 100)
- median = 20

* principal function of median filters is to force points with distinct intensity levels to be more like their neighbors

* The isolated clusters of pixels that are light or dark WRT their neighbors & whose area is less than $\frac{m^2}{2}$ [one-half the filter area] are eliminated by a m×m median filter

* in this case elimination means, forced to the median intensity of the neighbors

* larger clusters are affected considerably less.

* median represents ⇒ 50th percentile of a ranked set of the no

* eg 100th percentile ⇒ max filter which is useful for finding the brightest points in an image

* The response of a 3×3 max filter is given by $R = \max\{z_k \mid k = 1, 2, \ldots 9\}$

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

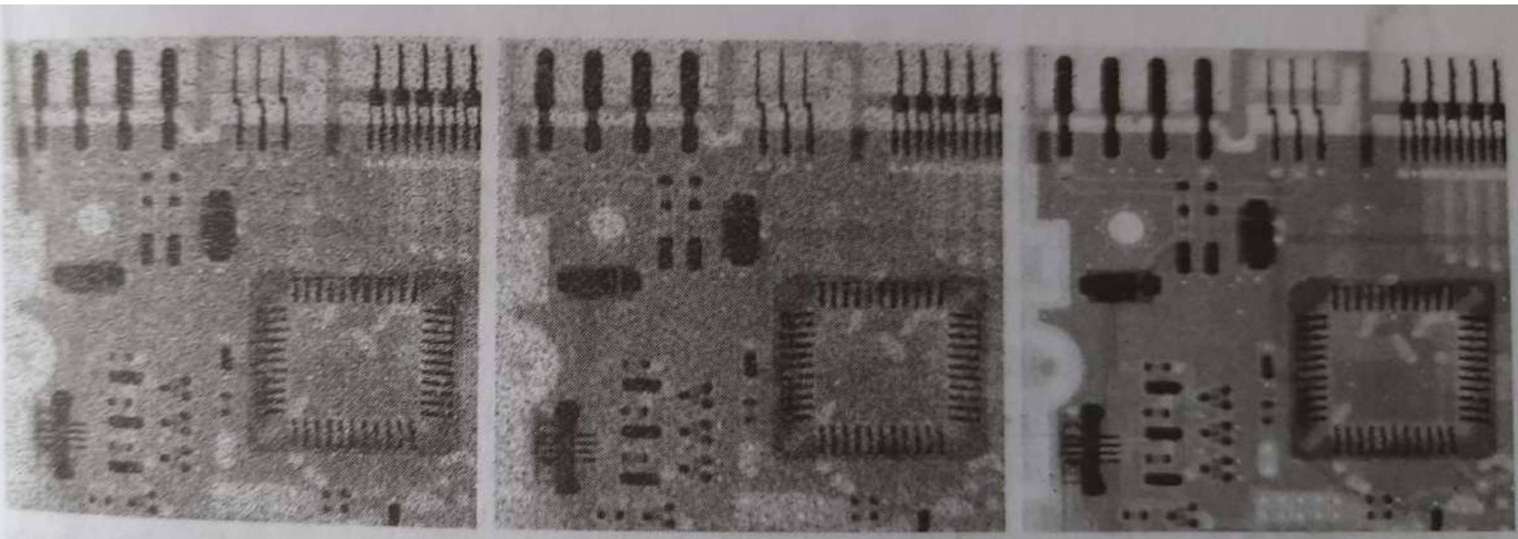


FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

sharpening spatial filters

~~* objective~~

* objective of sharpening is to highlight transitions in intensity.

* applns : ranging from electronic printing & medical imaging, industrial inspection & autonomous guidance in military system.

* image blurring in spatial domain is accomplished by pixel averaging in a neighborhood

* averaging is analogous to integration

* so we can conclude the sharpening can be accomplished by spatial differentia -tion

* the strength of the response of a derivative operator is proportional to the degree of intensity discontinuity of the image at the point at which the operator is applied

* Thus image differentiation enhances edges and other discontinuities (such as noises) & deemphasizes areas with slowly varying intensities.

Foundation

- sharpening filters are based on first & second-order derivatives respectively.
- To simplify the explanation let us initially focus on one-dimensional derivatives
- we are interested in the behavior of these derivatives in the areas of
  - (i) constant intensity
  - (ii) at the onset & end of discontinuities (step & ramp discontinuities)
  - & (iii) along intensity ramps
- These types of discontinuities can be used to model noise points, lines & edges in an image.
- The behavior of derivatives during transitions into & out of these image features also is of interest
- The derivatives of a digital functions are defined in terms of differences
- There are various ways to defines these differences.

1

(1) first derivatives

1 must be zero in areas of constant intensity

2 must be non-zero at the onset of a intensity step or ramp

3 must be nonzero along ramps

(2) second-derivatives

1 must be zero in constant areas

2 must be nonzero at the onset & end of an intensity step or ramp

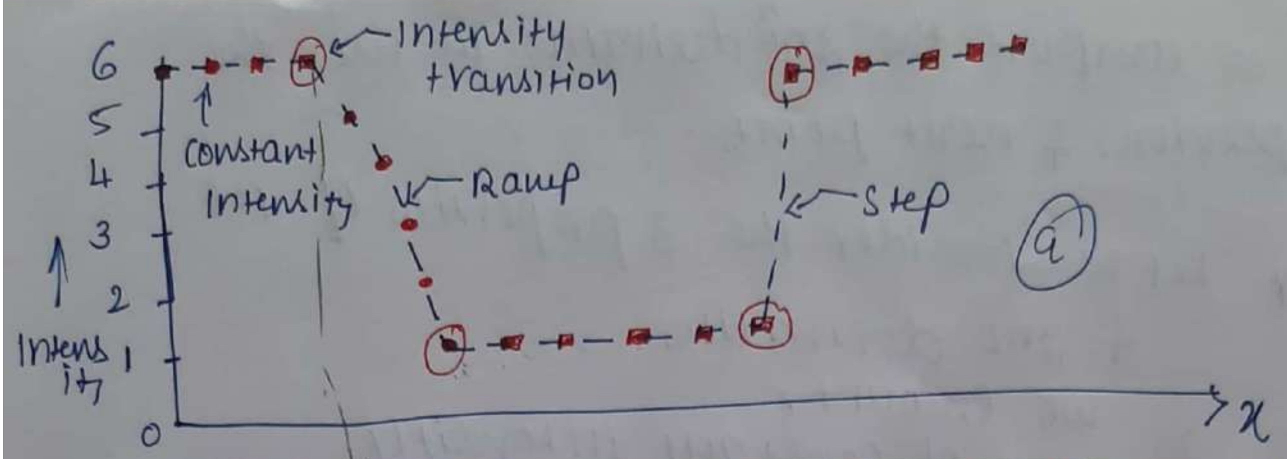
3 must be zero along ramps of constant slope

* basic defn of 1st-order delivative of one-dimensional fun $f(x)$ is

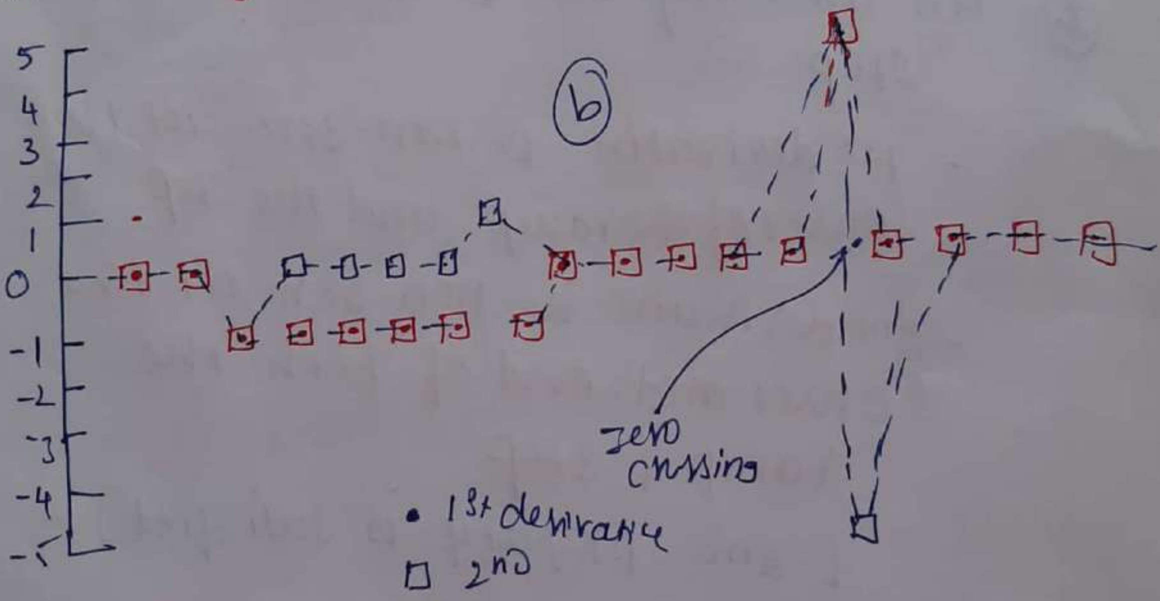
$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \longrightarrow ①$$

* second-order derivative of $f(x)$

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



scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6	→ x
	6	6	6	5	4	3	2	1		1	1	1	1		6	6	6	6		
		6	6	6	6	5	4	3	2	1	1	1	1		1	6	6	6	6	6
1st derivative	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	5	0	0	0	0		
2nd derivative		0	0	-1	0	0	0	0	1	0	0	0	0	5	-5	0	0	0		



* scan line ⇒ values are intensity values which are plotted as dot in fig ⓐ

* fig ⓐ intensity ramp, 3 sections of constant, intensity step

* ○ ⇒ onset or end of intensity transistion

* when computing 1st derivative at loc^n x, we subtract the value of fun at the location from next point. Look-ahead operation

④

* to compute the 2nd derivative we use the previous & next points.

* Let us consider the 3 properties of 1st & 2nd derivatives we encounter

① area of constant intensity
- both derivaties are zero
[so condn 1 is statisfied for bot]

② an intensity ramp followed by a step
- 1st derivative is non-zero at the onset of the ramp and the step
- 2nd derivative is non-zero at the onset and end of both the ramp & step
[2nd property is satisfied]

③ 1st derivative is non zero & 2nd is zero along the ramp

Note that the sign of the second derivative changes at the onset and end of a step or a ramp

* fig ⓑ in a step transition a line joining these 2 values crosses the horizontal axis midway bet' 2 extremes. This zero crossing property

* This zero crossing property is quite useful for locating edges.

* edges in digital images often are ramp like transitions in intensity

in which case

* 1st derivative of the image would result in thick edges ∵ the derivative is non-zero along the ramp

* 2nd derivative would produce a double edge one pixel thick, separated by zero

∴ 2nd derivative enhances fine detail much better than the 1st derivative & is ideally suited for sharpening images

using the second derivative for Image Sharpening - The Laplacian

* ~~2~~ Implementation of 2-D 2nd order derivatives & their uses for image sharpening.

* The approach consists of defining a discrete formulation of the second-order derivative & then constructing a filter mask ~~based~~ based on that formulation

* isotropic filters, whose response is independent of the direction of the discontinuities in the image to which the filter is applied

* istoropic filters are rotation invariant [rotating the image & then applying the filter give same result as applying the filter to the image first & then rotating the result].

* simplest isotropic derivative operator is Laplacian which for a function (image) f (x,y) of 2 valiables is defined as [Rosenfed & Kak 1982]

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \longrightarrow ③$$

* ∵ derivatives of any order are linear opeen, the Laplacian is a linear operator

using eq ② [2nd order]

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \rightarrow ④$$

→ (x-direⁿ)

& in y-dirⁿ

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \rightarrow ⑤$$

∴ the discrete Laplacian of 2 variables is

$$\nabla^2 f(x,y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x,y) \longrightarrow ⑥$$

- This eqⁿ can be implemented using the filter mask shown in fig 3.37 ⓐ which gives an isotropic result for rotations in increments of 90°

- The diagonal directions can be incorporated in the defⁿ of the digital laplacian by adding 2 more terms in eq 4 + ⑤

∴ each diagonal term also contains $-2f(x,y)$

	y+1	y	y-1
x+1	x+1, y+1	x+1 y	x+1, y-1
x	x, y+1	x y	x, y-1
x-1	x-1 y+1	x-1 y	x-1 y-1

∴ total subtracted from the difference term now would be $-8f(x,y)$.

- * this can be used for filter mask implementation of fig 3.37 ⓑ.
- * This mask yields ~~iso~~ isotropic results in increments of 45°

90°

0	1	0
1	-4	1
0	1	0

3.37 ⓐ

45°

1	1	1
1	-8	1
1	1	1

3.37 ⓑ

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

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

- * because the Laplacian is derivative operator, it uses highlights intensity discontinuities in an image & deemphasizes regions with slowly varying intensity levels
- * produce images that have grayish edge lines & other discontinuities in ~~an image~~ ~~& deemphasizes regions wi~~ all superimposed on a dark featureless background

* Laplacian for image sharpening

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

$f(x,y) \rightarrow$ i/p image

$g(x,y) \rightarrow$ sharpened image

$c = -1$ = constant (sub)

1 if other filter are used (add)

unsharp masking & Highboost filtering

* A process that has been used by the printing & publishing industry for many years is to sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image
* This process is called unsharp masking

* & consists of foll steps

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

* $\bar{f}(x,y)$ ⇒ denote blurred image

* unsharp masking is expressed in eq^n form as follows

$$\boxed{g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)} \rightarrow ⑧$$

* Then we add a weighted portion of the mask back to the original image

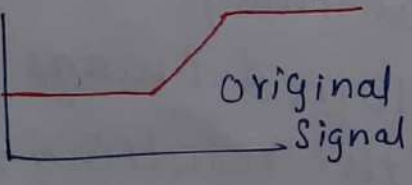
$$\boxed{g(x,y) = f(x,y) + K * g_{mask}(x,y)} \rightarrow ⑨$$

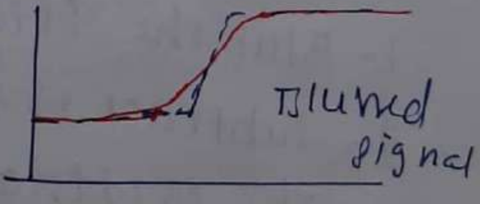
$K \geq 0$ for generality

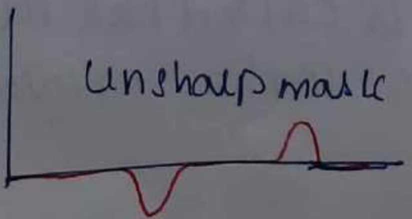
$K = 1$ we have unsharp masking

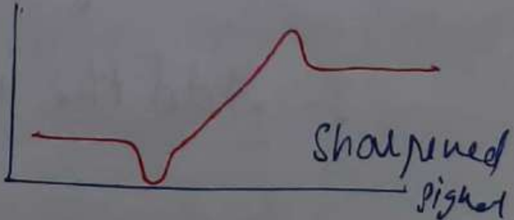
$K > 1$, process is referred as high boost filtering

$K < 1$, de-emphasizes the contribution of the un-sharp mask









using first-order derivatives for (Non-linear) Image sharpening - The Gradient

* 1st derivatives in image processing are implemented using the magnitude of the gradient

* For a function $f(x,y)$, the gradient of "f" at co-ordinates (x,y) is defined as the 2-dimensional column vector

$$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \rightarrow \text{⑩}$$

* The vector has important geometrical property that It points in the direction of the greatest rate of change of 'f' at location (x,y)

* magnitude (length) of vector ∇f, denoted as $M(x,y)$, where

$$\boxed{M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}} \rightarrow \text{⑪}$$

is the value at (x,y) of the rate of change in the direction of the gradient vector.

$$M(x,y) \approx |g_x| + |g_y| \rightarrow \text{⑫}$$

⑧

partial derivatives of eq ⑯ are not rotation invariant (isotropic) but the magnitude of the gradient vector is

* we define discrete approximation to the preceding eqn & from there formulate the appropriate filter mask

* 3x3 region of image [Zs are intensity value]

fig ⓐ

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

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

① Roberts cross-gradient operators

-1	0
0	1

0	-1
1	0

fig ⓑ

$g_x = (z_8 - z_5)$ & $g_y (z_6 - z_5)$

* 2 other defns proposed by Roberts in the early development of digital image processing use cross differences

$g_x = (z_9 - z_5)$ & $g_y = (z_8 - z_6) \rightarrow$ ⑬

use eq 11 & 13 we can compute gradient image as

$$M(x,y) = \sqrt{g_x^2 + g_y^2}$$

$$M(x,y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2} \rightarrow ⑭$$

If we use eq ⑫ & ⑬

$$M(x,y) \simeq |g_x| + |g_y|$$

$$M(x,y) \simeq |z_9 - z_5| + |z_8 - z_6| \rightarrow ⑮$$

- * The partial derivatives terms in eq ⑬ ~~it~~ can be implemented using 2 linear filter as shown in fig ⑥

- * These masks are refered as **Roberts-cross-gradient operator**

(ii) Sobel operator

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

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

fig ⑦

* masks of even sizes don't have a center of symmetry

* The smallest filter mask is 3 X 3

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad ⑯$$

&

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad ⑰$$

* These eqns can be implemented using masks of fig ⓒ.

* Substituting g_x & g_y in ⑫

$$M(x,y) \simeq |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)| \quad ⑱$$

* The masks are called Sobel operators

FIGURE 3.40 (a) Original image. (b) Result of blurring with a Gaussian filter. (c) Unsharp mask. (d) Result of using unsharp masking. (e) Result of using highboost filtering.

[Module-3]

Filtering, Image Restoration

Preliminary concepts, The DFT of one Valiable, Extension to functions of 2 variables, some properties of the 2-D DFT, freq domain filtering, A Model of the image degradation/ Restoration process. Noise models, Restoration in the presence of noise, only - spatial filtering, homomorphic filtering

Ch 4 :- 4.2 to 4.7, 4.9.6 , Ch 5 :- 5.2, 5.3

[4.5] Extension to function of 2 variables]

4.5.1 The 2-D impulse and its shifting property

* The impulse, $\delta(t, z)$ of 2 continuous Valiables t & z is defined as in

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

$$\delta(t, z) = \begin{cases} \infty ; & \text{if } t = z = 0 \\ 0 ; & \text{otherwise} \end{cases} \longrightarrow ①$$

&

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(t, z) \, dt \, dz = 1 \longrightarrow ②$$

* 2-D impulse exhibits shifting property as in 1-D

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t, z) \, \delta(t, z) \, dt \, dz = f(0,0) \longrightarrow ③$$

①

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)\,\delta(t-t_0, z-z_0)\,dt\,dz = f(t_0,z_0) \longrightarrow ③$$

[Shifting property yields the value of the fun $f(t,z)$ at the location of the impulse.

- for discrete variables x & y, the 2-D discrete impulse is defined as

$$\delta(x,y) = \begin{cases} 1 ; & \text{If } x=y=0 \\ 0 ; & \text{otherwise} \end{cases} \quad — ④$$

& its shifting property is

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\cdot\delta(x,y) = f(0,0) \longrightarrow ⑤$$

or

$$\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty} f(x,y)\cdot\delta(x-x_0, y-y_0) = f(x_0,y_0) \longrightarrow ⑥$$

4.5.2 The 2-D continuous Fourier ~~Trasf~~ Transform pair

Let $f(t,z) \longrightarrow$ continous function of 2 continuous variables t & z.

- 2-dimensional, continuous FT pair (CFT) is given by

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

$$\longrightarrow ⑦$$

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

$$\longrightarrow ⑧$$

μ & v → freq variables

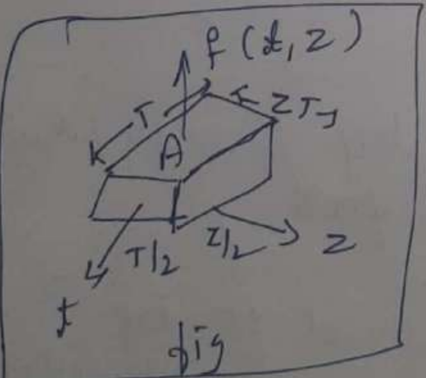
t & z → are interpreted to be continuous spatial variables

① Fig shows a 2-D fun analogous to 1-D

$$F(t,z) = A\, e^{\cancel{j2\pi(t}}, z)$$

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

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



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

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

4.5.3

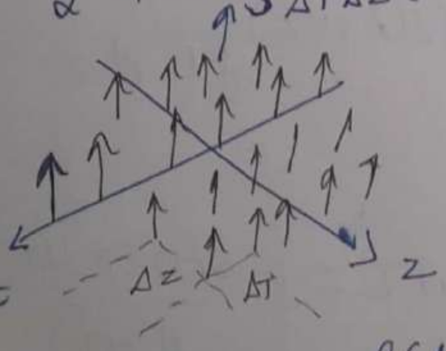
TWO - Dimensional sampling & 2-D Sampling Theorem

* sampling in 2-dimensions can be modeled using the sampling function (2-D impulse train]

$$S_{\Delta T \Delta Z}(t, z) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t - m\Delta T, z - n\Delta Z) \longrightarrow ⑨$$

where ΔT & ΔZ are the separations bet' samples along 't' axis & Z-axis of the continuous fun f(t, z).

* eq ⑨ represents a set of periodic impulses extending infinetly along the 2 axis as shown in fig below.



* multiplying f(t, z) by $S_{\Delta T \Delta Z}(t, z)$ yields the sampled fun

* function f(t, z) is said to be band-limited, if its fourier transform is 0 outside a rectangle established by the intervals [−μmax, μmax] & [−Vmax, Vmax].

$F(\mu, v) = 0$ for $|\mu| \geq \mu_{max}$ & $|v| \geq v_{max}$ → ⑩

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

$$\Delta T < \frac{1}{2\mu_{max}} \quad — ⑪$$

& $$\Delta Z < \frac{1}{2V_{max}} \quad — ⑫$$

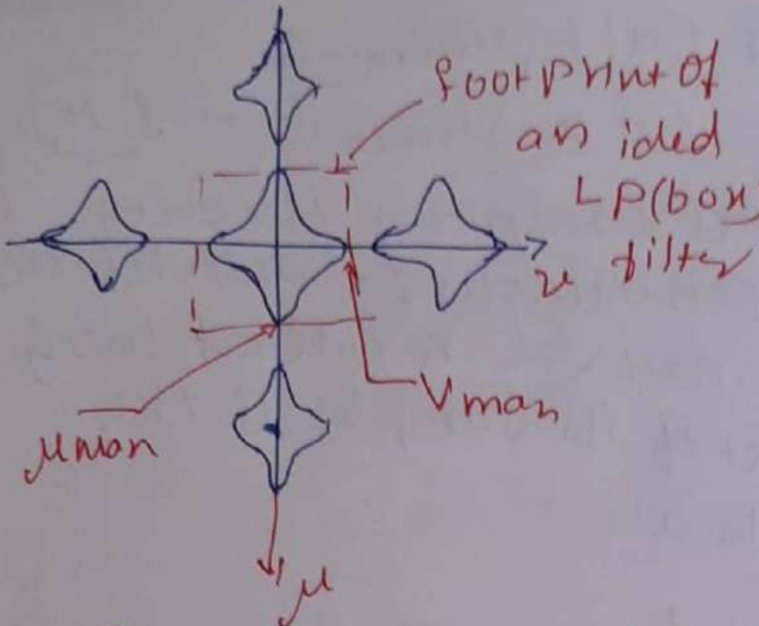
or expressed in terms of the sampling rate if

$$\frac{1}{\Delta T} > 2\mu_{max} \quad — ⑬$$

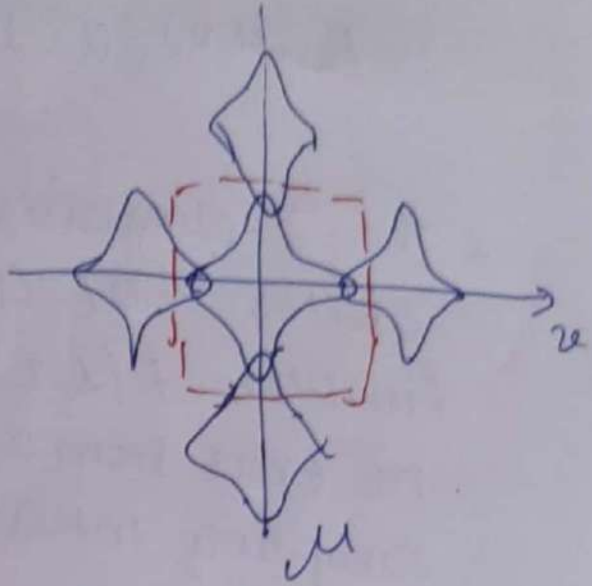
& $$\frac{1}{\Delta Z} > 2V_{max} \quad — ⑭$$

* no information is lost if a 2-D, band-limited continuous fun is represented by samples acquired at rates greater than twice the highest freq contents of the function in both μ & v- directions

③



ⓐ an oversampled fun



ⓑ under-sampled fun

4.5.4 Aliasing in Images

- concept of aliasing to images & several aspects related to image sampling & resampling is discussed.

- f(t, z) of 2 continuous variables t & z can be band-limited in general only if it extends infinitely in both coordinate directions.
- ~~The~~ By limiting the duration of the function, introduces corrupting the freq components extending to infinity in the freq-domain
- ∴ we cannot sample a fun infinitely, aliasing is always present in digital images

* There are 2 principal manifestations of aliasing in images
(i) spatial aliasing
& (ii) Temporal aliasing

* spatial aliasing: is due to undersampling

* Temporal aliasing: is ~~related~~ ~~due to~~ related to time intervals bet' images in a sequence of images.

eg. "wagon wheel" effect in which wheels with ~~spokes~~ spokes in a sequence of images (for eg in a movie) appear to be rotating backward. This is caused by the frame rate being too low WRT the speed of wheel rotation in the sequence

* spatial aliasing:
- The key concerns with spatial aliasing in images are the introduction of artifacts such as jaggedness in line features, suprious highlights & the appearance of freq patterns not present in the original image

* The effects of aliasing can be reduced by slightly defocusing the scene to the digitized so that high frequencies are attenuated

* anti-aliasing filtering has to be done at the "front-end" before the image is sampled.

* blurring a digital image can reduce additional aliasing artifacts caused by resampling

image interpolation & resampling

* perfect reconstruction of a bandlimited image function from a set of its samples requires 2-D convolution in spatial domain with a sinc fun.

* WKT a perfect reconstruction requires interpolation using infinite summation

* one of the most common applns of 2-D interpolation in image processing is in image resing [zooming & shrinking).

* Zooming : may ~~be~~ viewed as over-sampling while shrinking may be viewed as under-sampling

* They are applied to digital images

* A special case of nearest neighbor interpolation that ties in nicely with oversampling is zooming by pixel replication [which is applicable when we want to ~~inc~~ the size of an image an integer no of times]

* If we need to double the size of the image, we duplicate each column which doubles image size in horizontal direction.

* Then we duplicate each row of the enlarged image to double the size in vertical direction

* The same Procedure is used to enlarge the image any integer no of times

* The intensity level assignment of each pixel is predetermined by the fact that new locations are exact duplicates of old location

* Image shrinking is done in a manner similar to zooming.

* under sampling is achieved by row-column deletion. ~~(eg~~.

* example:. to shrink an image by 1/2, we delet every other row & column.

* To reduce aliasing, it is good idea to blur an image slightly before shrinking it.

* An alternate technique is to supersample the original scene & then reduce (resample) its size by row & column deletion.

* This yield sharper results than with smoothing. (clear access to orignal image is needed)

* If no access to original scene, supersampling is not an option.

* for image which have strong edge content, the effects of aliasing are seen as block-like image components called Jaggies

moire patterns: another type of artifacts which result from sampling scenes with periodic or nearly periodic components

eg in digital image, the problem arises when scanning media print such as newspapers, magzines.

* superimposing one pattern on the other creats a beat pattern that has frequencies not present in either of the original pattern.

* the morie effect produced by 2 patterns of dots is discussed further

* Newspapers & other printed materials make use of so called halftone dots which are black dots or ellipses whose sizes & various joining schemes are used to stimulate gray tones.

4.5.5 The 2-D discrete Fourier Transforms & its inverse

* 2-D discrete Fourier Transform DFT

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad ⑮$$

$f(x,y) \rightarrow$ digital image of size $M \times N$

$u, v \rightarrow$ discrete values ranging from 0 to $M-1$ & 0 to $N-1$ resp.

* inverse DFT

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad ⑯$$

⑥

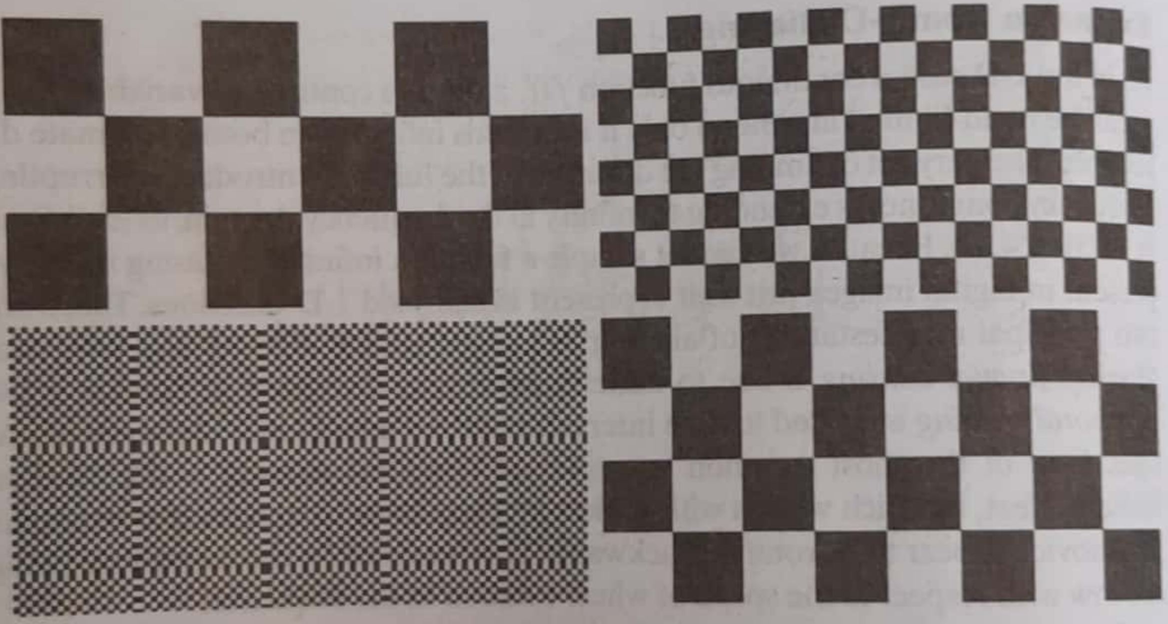
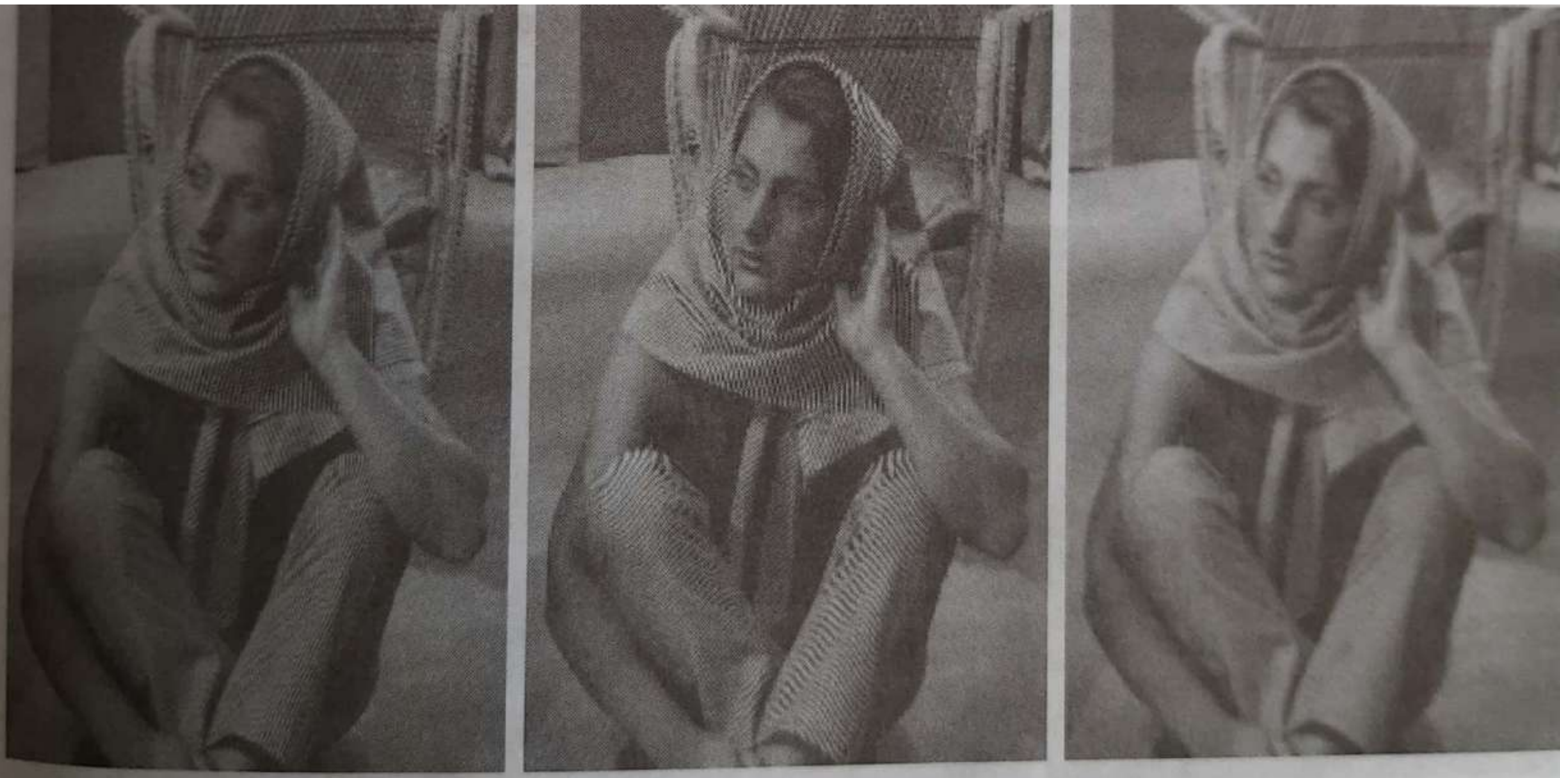


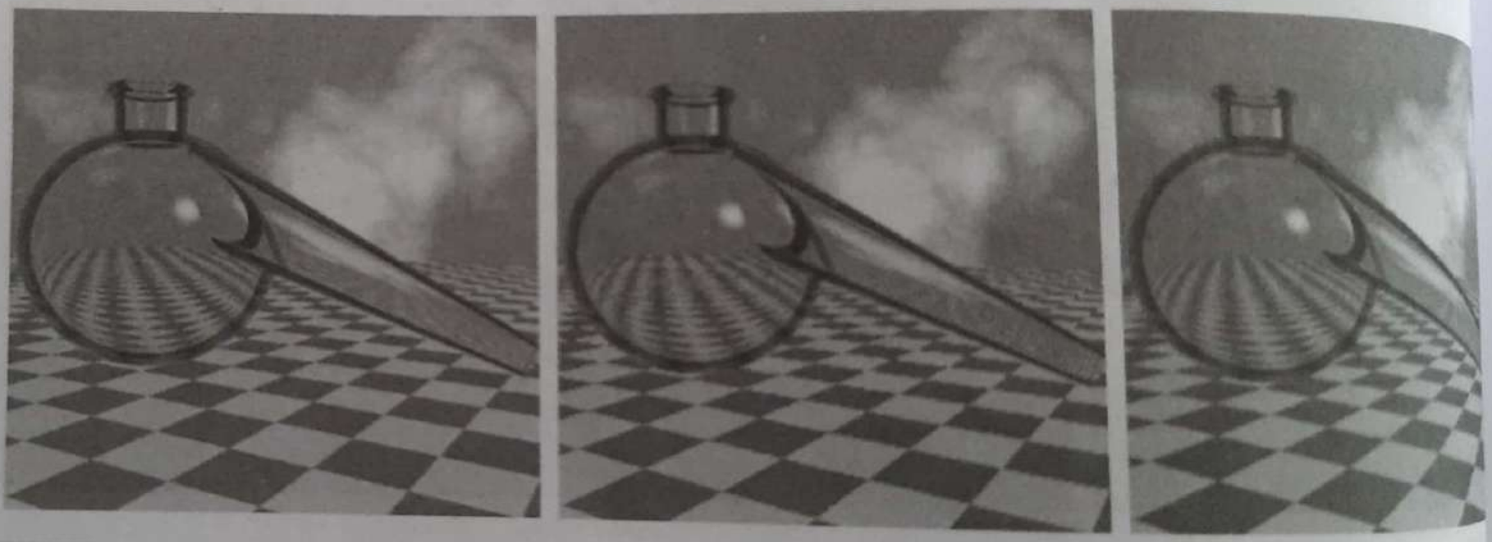
FIGURE 4.16 Aliasing in images. In (a) and (b), the lengths of the sides of the squares are 16 and 6 pixels, respectively, and aliasing is visually negligible. In (c) and (d), the sides of the squares are 0.9174 and 0.4798 pixels, respectively, and the results show significant aliasing. Note that (d) masquerades as a "normal" image.



a b c

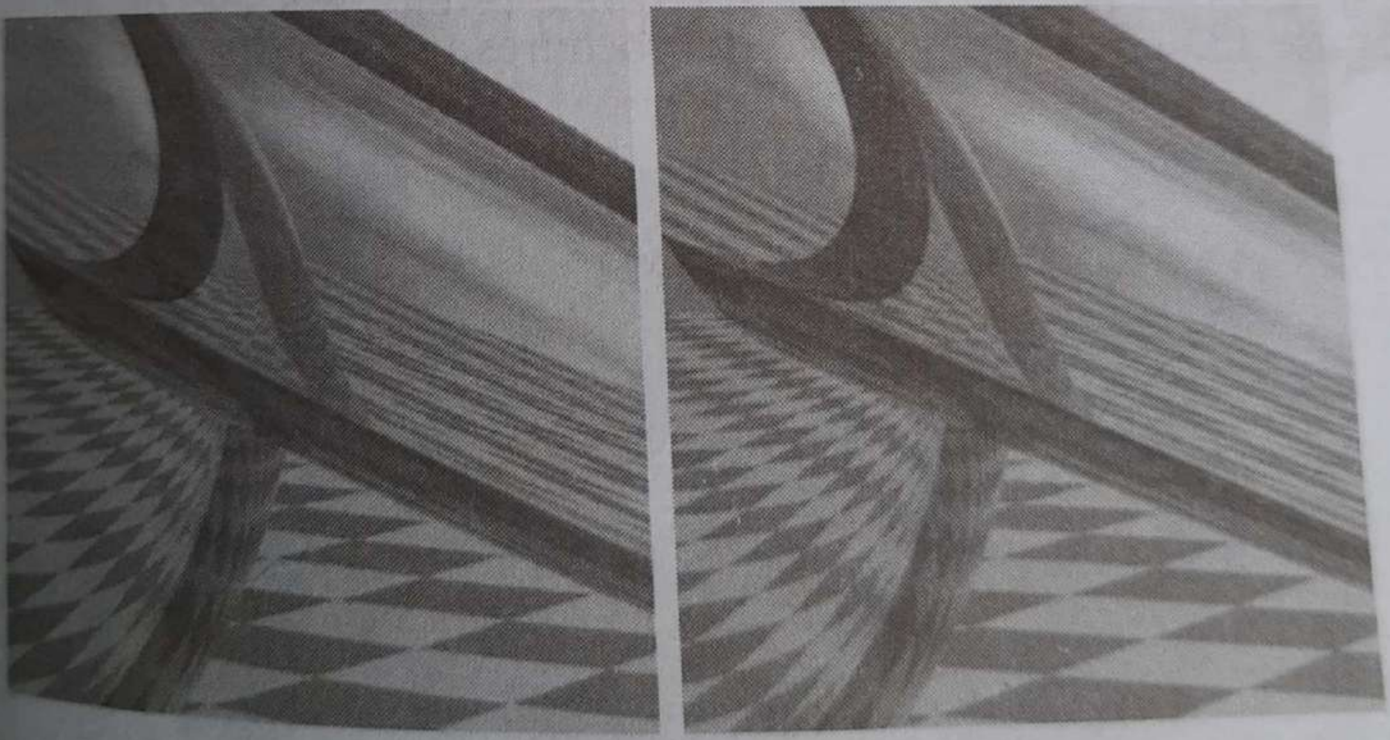
FIGURE 4.17 Illustration of aliasing on resampled images. (a) A digital image with negligible visual aliasi (b) Result of resizing the image to 50% of its original size by pixel deletion. Aliasing is clearly visib (c) Result of blurring the image in (a) with a 3 × 3 averaging filter prior to resizing. The image is sligh more blurred than (b), but aliasing is not longer objectionable. (Original image courtesy of the Sig Compression Laboratory, University of California, Santa Barbara.)

generate Fig. 4.18(b).



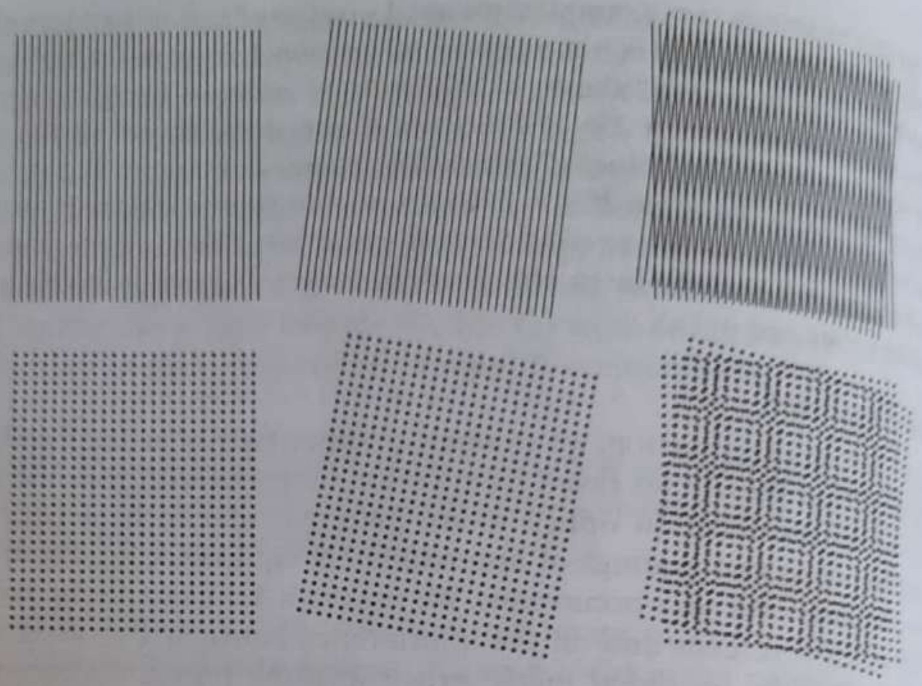
a b c

FIGURE 4.18 Illustration of jaggies. (a) A 1024 × 1024 digital image of a computer-generated sce negligible visible aliasing. (b) Result of reducing (a) to 25% of its original size using bilinear interp (c) Result of blurring the image in (a) with a 5 × 5 averaging filter prior to resizing it to 25% using b interpolation. (Original image courtesy of D. P. Mitchell, Mental Landscape, LLC.)



a b

FIGURE 4.19 Image [illegible] generated by



a b c
d e f

FIGURE 4.20 Examples of the moiré effect. These are ink drawings, not digitized patterns. Superimposing one pattern on the other is equivalent mathematically to multiplying the patterns.

a beat pattern that has frequencies not present in either of the original patterns. Note in particular the moiré effect produced by two patterns of dots, as this is the effect of interest in the following discussion.

Newspapers and other printed materials make use of so called *halftone dots*, which are black dots or ellipses whose sizes and various joining schemes are used to simulate gray tones. As a rule, the following numbers are typical: newspapers are printed using 75 halftone dots per inch (*dpi* for short), magazines use 133 dpi, and high-quality brochures use 175 dpi. Figure 4.21 shows

Color printing uses red, green, and blue dots to produce the sensation in the eye of continuous color.

FIGURE 4.21 A newspaper image of size 246 × 168 pixels sampled at 75 dpi showing a moiré pattern. The moiré pattern in this image is the interference pattern created between the ±45° orientation of the halftone dots and the north–south orientation of the sampling grid used to digitize the image.

* 4.6 Some properties of the 2-D Discrete Fourier Transform [DFT]

→ $\boxed{4.6.1}$ Relationship between spatial & frequency Intervals

* $f(t,z) \Rightarrow$ continuous fun
* $f(x,y) \Rightarrow$ sampled form of $f(t,z)$ digital image which consists of M X N samples taken 't' & 'z' directions resp.

* Let ΔT & ΔZ → denote the sepa'n bet' samples.

* separations bet' the corresponding discrete, frequency domain variables are given by

$$\Delta u = \frac{1}{M \Delta T} \longrightarrow ①$$

$$\& \quad \Delta v = \frac{1}{N \Delta T} \longrightarrow ②$$

separation bet' samples in freq domain are inversely proportional both to the spacing bet' spatial samples & no of samples

4.6.2 Translation & Rotation

FT pairs ~~statis~~ satisfies the foll translation Properties

$$f(x,y)\, e^{j2\pi(u_0x/M + v_0y/N)} \Longleftrightarrow F(u-u_0, v-v_0) \quad ③$$

&

$$f(x-x_0, y-y_0) \Longleftrightarrow F(u,v)\, e^{-j2\pi(x_0u/M + y_0v/N)} \quad ④$$

- x'lying f(x,y) by exponential shows shifts the origin of DFT to (u_0, v_0)
- conversely xling F(u,v) by negative of exponential shifts the origin of f(x,y) to (x_0, y_0)
- using the polar co-ordinates $x = r\cos\theta$, $y = r\sin\theta$, $u = \omega\cos\varphi$ $v = \omega\sin\varphi$

results in

$$f(r, \theta+\theta_0) \Longleftrightarrow F(\omega, \varphi+\theta_0)$$

$\Rightarrow$ rotating f(x,y) by an angle θ_0 rotates F(u,v) by the same angle. Conversly, rotating F(u,v) rotates f(x,y) by the same angle

⑦

* 4.6.3 periodicity

2-D FT & its inverse are ~~infinet~~ infinitely periodic in the u & v directions ie

$$F(u,v) = F(u + k_1 M, v) = F(u, v + k_2 N) = F(u + k_1 M, v + k_2 N) \rightarrow ⑥$$

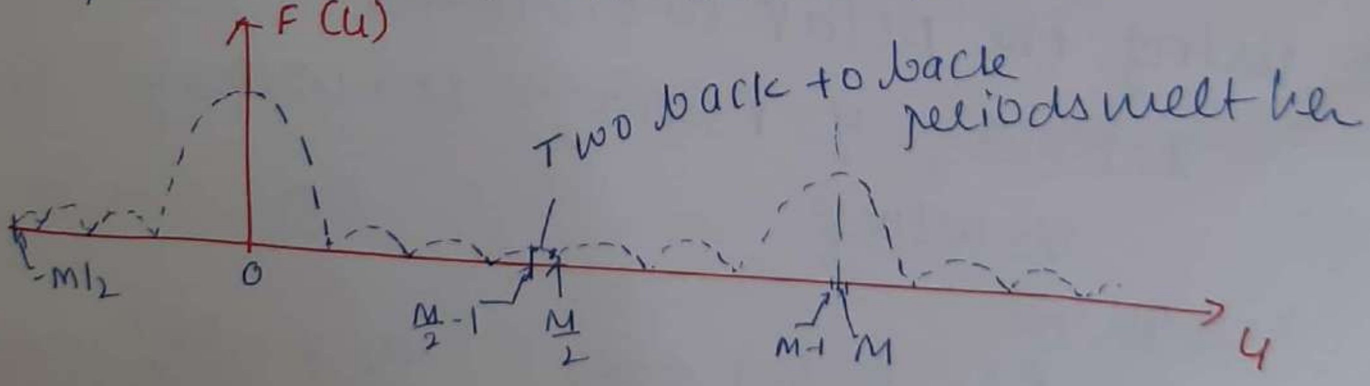
&

$$f(x,y) = f(x + k_1 M, y) = f(x, y + k_2 N) = f(x + k_1 M, y + k_2 N) \longrightarrow ⑦$$

where k_1 & k_2 are integers

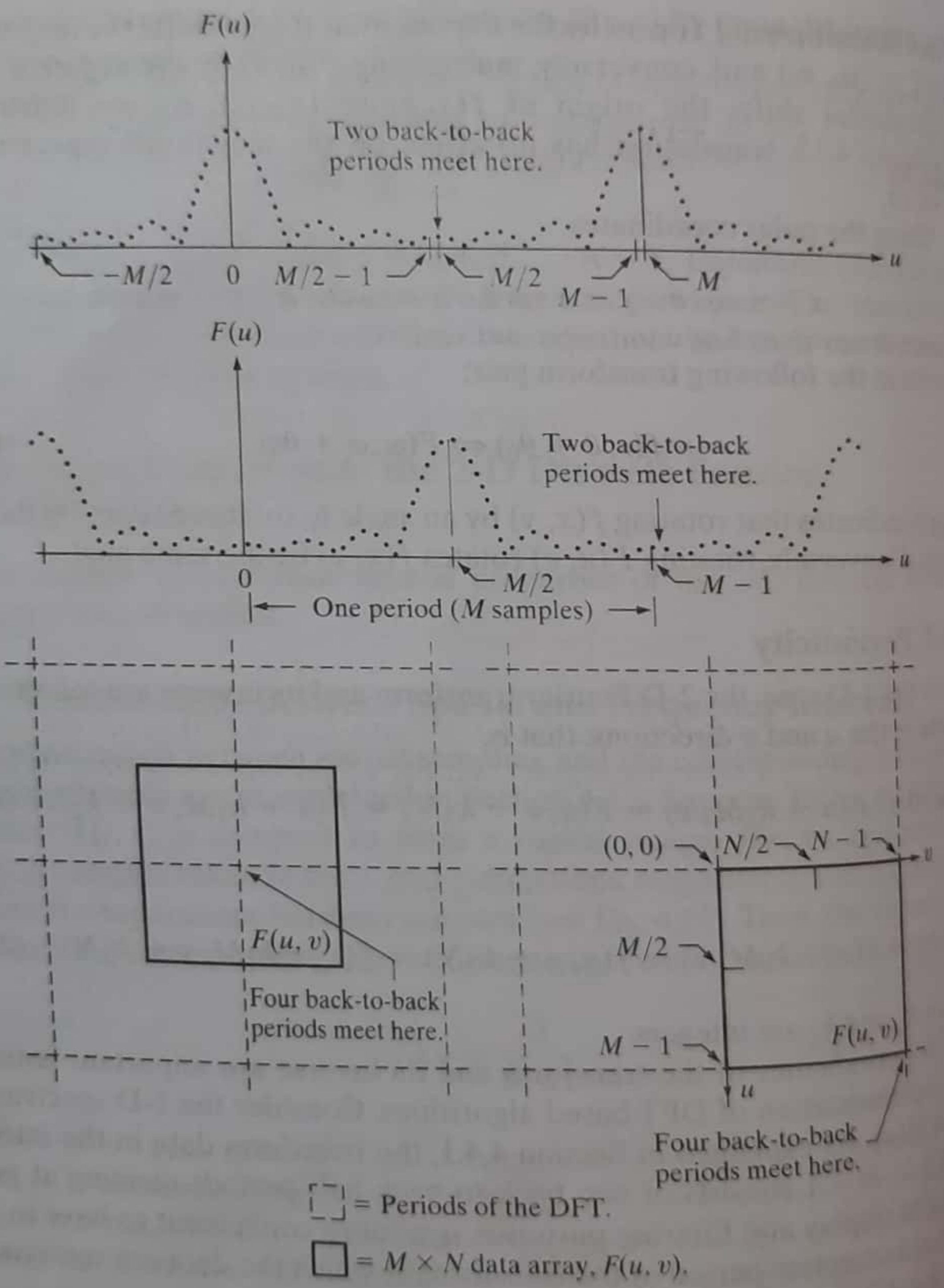
* The periodicities of the Transforms & its inverse are important issues in the implementation of DFT-based algorithms

* ~~The transform data~~ in the



a
b
c d

FIGURE 4.23 Centering the Fourier transform. (a) A 1-D DFT showing an infinite number of periods. (b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$. (c) A 2-D DFT showing an infinite number of periods. The solid area is the $M \times N$ data array, $F(u, v)$, obtained with Eq. (4.5-15). This array consists of four quarter periods. (d) A Shifted DFT obtained by multiplying $f(x, y)$ by $(-1)^{x+y}$ before computing $F(u, v)$. The data now contains one complete, centered period, as in (b).



* consider the 1-D spectrum in above fig.

* The transform data in the interval from 0 to M-1 consists of 2 - back to back half periods meeting at a point $\frac{M}{2}$.

* For displaying & filteling purposes, it is more convenient to have in this interval a complete period of transform in which data are contiguous as shown in fig ⓑ

$$f(x) e^{j2\pi (u_0 x/M)} \Longleftrightarrow F(u-u_0)$$

* multiplying f(x) by exponential term shown shifts the data so that the origin F(0) is located to u_0

* Let $u_0 = M/2$, the exponential term becomes $e^{j\pi x}$ which is equal to $(-1)^x$. ∵ x = integer

* in this case
$$f(x)(-1)^x \Longleftrightarrow F(u-M/2)$$

* xlying f(x) by $(-1)^x$ shifts the data so that F(0) is at the center of the interval [0, M-1]

* The principal is same for 2-D

⑧

* instead of 2 half periods, there are now 4 quarter periods meeting at the pt $(M/2, N/2)$

* The dashed line corresponds to the infinite no of periods of 2-D DFT

* If we shift data so that $F(0,0)$ is at $(M/2, N/2)$.

$(u_0, v_0) = (M/2, N/2)$

eq ⑦ becom

$$f(x,y)(-1)^{x+y} \Longleftrightarrow F(u-\frac{M}{2}, u-\frac{M}{2}) \rightarrow ⑧$$

*

4.6.4 Symmetry Properties

Any real or complex fun $w(x,y)$ can be expressed as the sum of an even & odd part (each of which can be real or complex)

$$w(x,y) = w_e(x,y) + w_o(x,y) \rightarrow ⑨$$

where even & odd parts are defined as

$$w_e(x,y) \triangleq \frac{w(x,y) + w(-x-y)}{2} \text{ --- } (10a)$$

& $w_o(x,y) \triangleq \frac{w(x,y) - w(-x-y)}{2} \longrightarrow$ (10b)

~~using~~

$w_e(x,y) = w_e(-x,-y) \longrightarrow$ 11(a)

& $w_o(x,y) = -w_o(-x,-y) \longrightarrow$ 11(b)

* even fun's are said to be symmetric
& odd fun's are antisymmetric

$w_e(x,y) = w_e(M-x, N-y) \longrightarrow$ 12(a)

& $w_o(x,y) = -w_o(M-x, N-y) \longrightarrow$ 12(b)

where M & N ⇒ no of rows & columns of a 2-D array

* WKT product of 2 even & 2 odd fun's is even & product of an even & an odd fun is odd

* The only way a discrete function be odd is if all its samples sum to zero.

* These properties lead to

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} w_e(x,y)\, w_o(x,y) = 0 \quad \text{—— (13)}$$

w_e = even & w_o = odd

- because ~~the~~ arguments of eq ⑬ is ~~zero~~ odd the result of summation is 0 ⑨

eg
consider the 1-D seq^n

$f = \{ f(0), f(1), f(2), f(3) \}$

$= \{ 2, 1, 1, 1 \}$ $\boxed{M = 4}$

(*) to test for evenness the cond^n

$f(x) = f(M-x) = f(4-x)$

$f(0) = f(4)$; $f(1) = f(3)$

$f(2) = f(2)$; $f(3) = f(1)$

* any -4 point even seq^n has to have the form

$\{ a, b, c, b \}$ 2^nd & ~~4~~ last pt must be equal

② An odd seq^n

$g = \{ g(0), g(1), g(2), g(3) \}$

$= \{ 0, -1, 0, 1 \}$

$g(x) = -g(4-x)$

$g(1) = -g(3)$

$= \{ 0, -b, 0, b \}$

* when M is an even no , a 1-D odd sign has the property that the points at location 0 & M/2 always are zero

* when M is odd, the 1st term still has to be 0, but the remaining terms form pairs with equal value but opposite sign.

* eg { 0, -1, 0, 1, 0 } is neither odd nor even. even though the basic structure appears to be odd

* $$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -2 & 0 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$ is odd

* adding another row & column of 0's would give a result i.e, neither odd nor even.

*

* A property used frequently is that FT of a real fun f(x,y) is conjugate symmetric

$$F^*(u,v) = F(-u,-v) \longrightarrow \textcircled{14}$$

* If f(x,y) is imaginary, its FT is conjugate antisymmetric

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

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

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

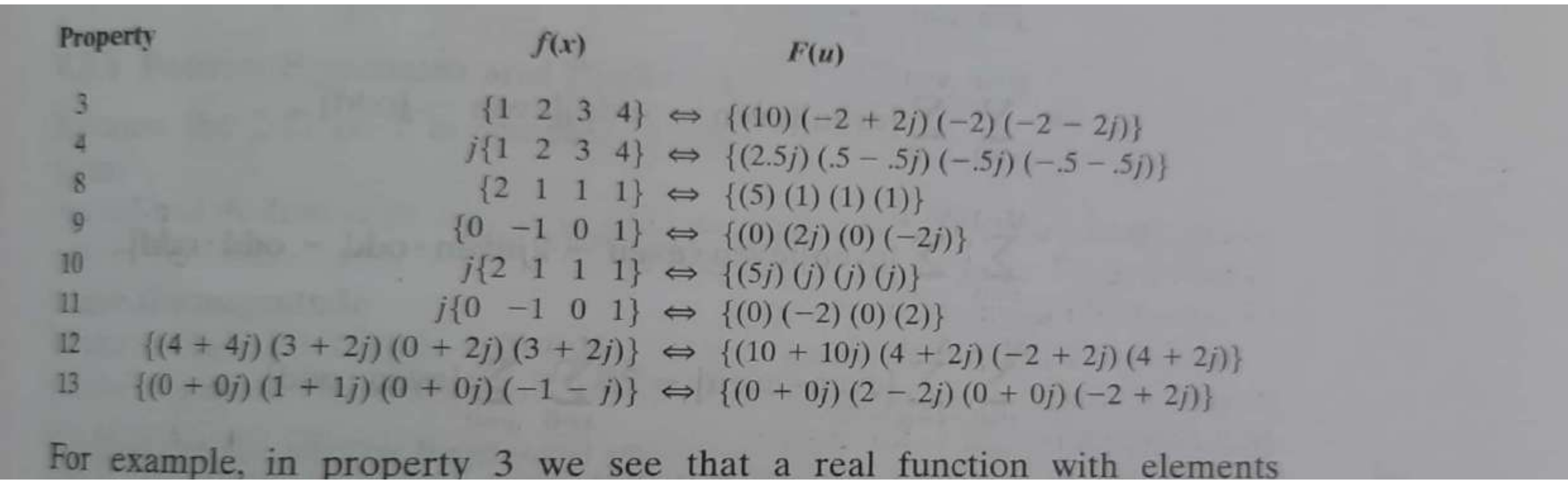
$$= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi\left[\frac{(-u)x}{M} + \frac{[-v]y}{N}\right]}$$

$$= F(-u,-v)$$

LE 4.1 Some
mmetry
perties of the
DFT and its
erse. $R(u,v)$
d $I(u,v)$ are the
l and imaginary
rts of $F(u,v)$,
spectively. The
rm *complex*
dicates that a
nction has
onzero real and
maginary parts.

	Spatial Domain[†]		Frequency Domain[†]
1)	$f(x, y)$ real	$\Leftrightarrow$	$F^*(u, v) = F(-u, -v)$
2)	$f(x, y)$ imaginary	$\Leftrightarrow$	$F^*(-u, -v) = -F(u, v)$
3)	$f(x, y)$ real	$\Leftrightarrow$	$R(u, v)$ even; $I(u, v)$ odd
4)	$f(x, y)$ imaginary	$\Leftrightarrow$	$R(u, v)$ odd; $I(u, v)$ even
5)	$f(-x, -y)$ real	$\Leftrightarrow$	$F^*(u, v)$ complex
6)	$f(-x, -y)$ complex	$\Leftrightarrow$	$F(-u, -v)$ complex
7)	$f^*(x, y)$ complex	$\Leftrightarrow$	$F^*(-u - v)$ complex
8)	$f(x, y)$ real and even	$\Leftrightarrow$	$F(u, v)$ real and even
9)	$f(x, y)$ real and odd	$\Leftrightarrow$	$F(u, v)$ imaginary and odd
10)	$f(x, y)$ imaginary and even	$\Leftrightarrow$	$F(u, v)$ imaginary and even
11)	$f(x, y)$ imaginary and odd	$\Leftrightarrow$	$F(u, v)$ real and odd
12)	$f(x, y)$ complex and even	$\Leftrightarrow$	$F(u, v)$ complex and even
13)	$f(x, y)$ complex and odd	$\Leftrightarrow$	$F(u, v)$ complex and odd

[†] Recall that x, y, u, and v are *discrete* (integer) variables, with x and u in the range $[0, M - 1]$, and y, and v in the range $[0, N - 1]$. To say that a complex function is *even* means that its real *and* imaginary parts are even, and similarly for an odd complex function.



Property	$f(x)$		$F(u)$
3	$\{1 \;\; 2 \;\; 3 \;\; 4\}$	$\Leftrightarrow$	$\{(10)\,(-2+2j)\,(-2)\,(-2-2j)\}$
4	$j\{1 \;\; 2 \;\; 3 \;\; 4\}$	$\Leftrightarrow$	$\{(2.5j)\,(.5-.5j)\,(-.5j)\,(-.5-.5j)\}$
8	$\{2 \;\; 1 \;\; 1 \;\; 1\}$	$\Leftrightarrow$	$\{(5)\,(1)\,(1)\,(1)\}$
9	$\{0 \;\; -1 \;\; 0 \;\; 1\}$	$\Leftrightarrow$	$\{(0)\,(2j)\,(0)\,(-2j)\}$
10	$j\{2 \;\; 1 \;\; 1 \;\; 1\}$	$\Leftrightarrow$	$\{(5j)\,(j)\,(j)\,(j)\}$
11	$j\{0 \;\; -1 \;\; 0 \;\; 1\}$	$\Leftrightarrow$	$\{(0)\,(-2)\,(0)\,(2)\}$
12	$\{(4+4j)\,(3+2j)\,(0+2j)\,(3+2j)\}$	$\Leftrightarrow$	$\{(10+10j)\,(4+2j)\,(-2+2j)\,(4+2j)\}$
13	$\{(0+0j)\,(1+1j)\,(0+0j)\,(-1-j)\}$	$\Leftrightarrow$	$\{(0+0j)\,(2-2j)\,(0+0j)\,(-2+2j)\}$

For example, in property 3 we see that a real function with elements

① $f(x) = \{1, 2, 3, 4\}$

$F(u) = \{10, [-2+2j], -2, [-2-2j]\}$

if $f(x,y)$ real ⟺ then $R(u,v)$ even.
$I(u,v)$ odd

$R(u,v) = \{10, -2, -2, -2\}$ is even

$I(u,v) = \{0, +2j, 0, -2]$ is odd

② $f(x) = j\{1, 2, 3, 4\} \Leftrightarrow F(u) = \{(2.5j),$
$\Updownarrow$ $(0.5$

$F(u) = \{2.5j, 0.5-0.5j, -0.5j, -0.5-0.5j\}$

$R(u,v) = \{0, 0.5, 0, -0.5\}$ is odd

I ~~(u,v) = { 0, 0.5, 0, -0.5~~

$I(u,v) = \{2.5, -0.5, -0.5, -0.5\}$ is even

Property 3:·

If (x,y) is real fun, the real part of its DFT is even & the odd part is odd

proof: If $F(u,v)$ is complex.

$F(u,v) = R(u,v) + j\, I(u,v)$

Then $F^*(u,v) = R(u,v) - j\, I(u,v)$

$F(-u,-v) = R(-u,-v) + jI(-u,-v)$

WKT if $f(x,y)$ is real, then

$F^*(u,v) = F(-u,-v)$

$R(u,v) = R(-u,-v) = \text{even}$

& $I(u,v) = -I(-u,-v) = \text{odd}$.

=

Property 8:

ST if (x,y) is real & even, then the imaginary part of $F(u,v)$ is 0 real & even

[to prove property 8, we need to show if $f(x,y)$ is real & even Imaginary part of $F(u,v)$ is 0]

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

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} [f_r(x,y)]\, e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right]}$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} [f_r(x,y)]\, e^{-j2\pi\left(\frac{ux}{M}\right)} \cdot e^{-j2\pi\left(\frac{uy}{N}\right)}$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} [\text{even}]\,[\text{even} - j\text{odd}]\,[\text{even} - j\text{odd}]$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} [\text{even}]\,[\text{even}\cdot\text{even} - 2j\text{even}\cdot\text{odd} - \text{odd}\cdot\text{odd}]$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} [\text{even.even}] - 2j\overset{=0}{\cancel{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} [\text{even.odd}]}}$$

$$= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} [\text{even.even}]$$

= Real

The second term is imaginary component $= 0$ according to $F^*(u,v) = F(-u,-v)$

4.6.5 Fourier spectrum & Phase Angle

- 2-D DFT is complex in general, we can express in polar form

$$F(u,v) = |F(u,v)|\, e^{j\phi(u,v)} \longrightarrow (15)$$

where the magnitude

$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2} \hookrightarrow (16)$$

is called Fourier spectrum [freq spectrum]

$$\phi(u,v) = \arctan\left[\frac{I(u,v)}{R(u,v)}\right] \text{—} (17)$$

is the phase angle

[atan2 (Imag, Real)] → MATLAB

12

* Power spectrum

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

$$= R^2(u,v) + I^2(u,v) \longrightarrow (18)$$

R → real part of F(u,v)
I → Imaginary — ,, —
$u = 0, 1, 2, \ldots M-1$
$V = 0, 1, 2 \ldots N-1$

$|F(u,v)|$, $\phi(u,v)$ & $P(u,v)$ ⇒ arrays of size $M \times N$.

* FT of a real funⁿ is congugate symmetric $F^*(u,v) = F(-u,-v)$ which implies that the spectrum has even symmetry about the origin

$$|F(u,v)| = |F(-u,-v)| \longrightarrow (19)$$

* The phase ange exhibits the $\phi(u)$ odd symmetry about the origin

$$\phi(u,v) = -\phi(-u,-v) \longrightarrow (20)$$

* $$F(0,0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

* which indicates the zero. freq term is α to the average value of $f(x,y)$

$$F(0,0) = MN \cdot \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

$$= MN\bar{f}(x,y) \longrightarrow ㉑$$

$\bar{f} \Rightarrow$ avg value of f

then

$$|F(0,0)| = MN|\bar{F}(x,y)| \rightarrow ㉒$$

because the proportionality constant MN usually is large, $|F(0,0)|$ typically is the largest component of the spectrum by a factor that can be ~~serted~~ ~~O~~ several orders of magnitude larger than other term.

* $F(0,0) \Rightarrow$ dc component of transform

⑬

2-D Convolution Theorem

* 2-D circular convolution

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

↳ ㉓

for $x = 0, 1, 2 \cdots M-1$
$y = 0, 1, 2 \cdots N-1$

* The 2-D convolution theorem is given by the expressions

$$\boxed{f(x,y) \circledast h(x,y) \overset{FT}{\Longleftrightarrow} F(u,v)\, H(u,v)}$$

↳ ㉔

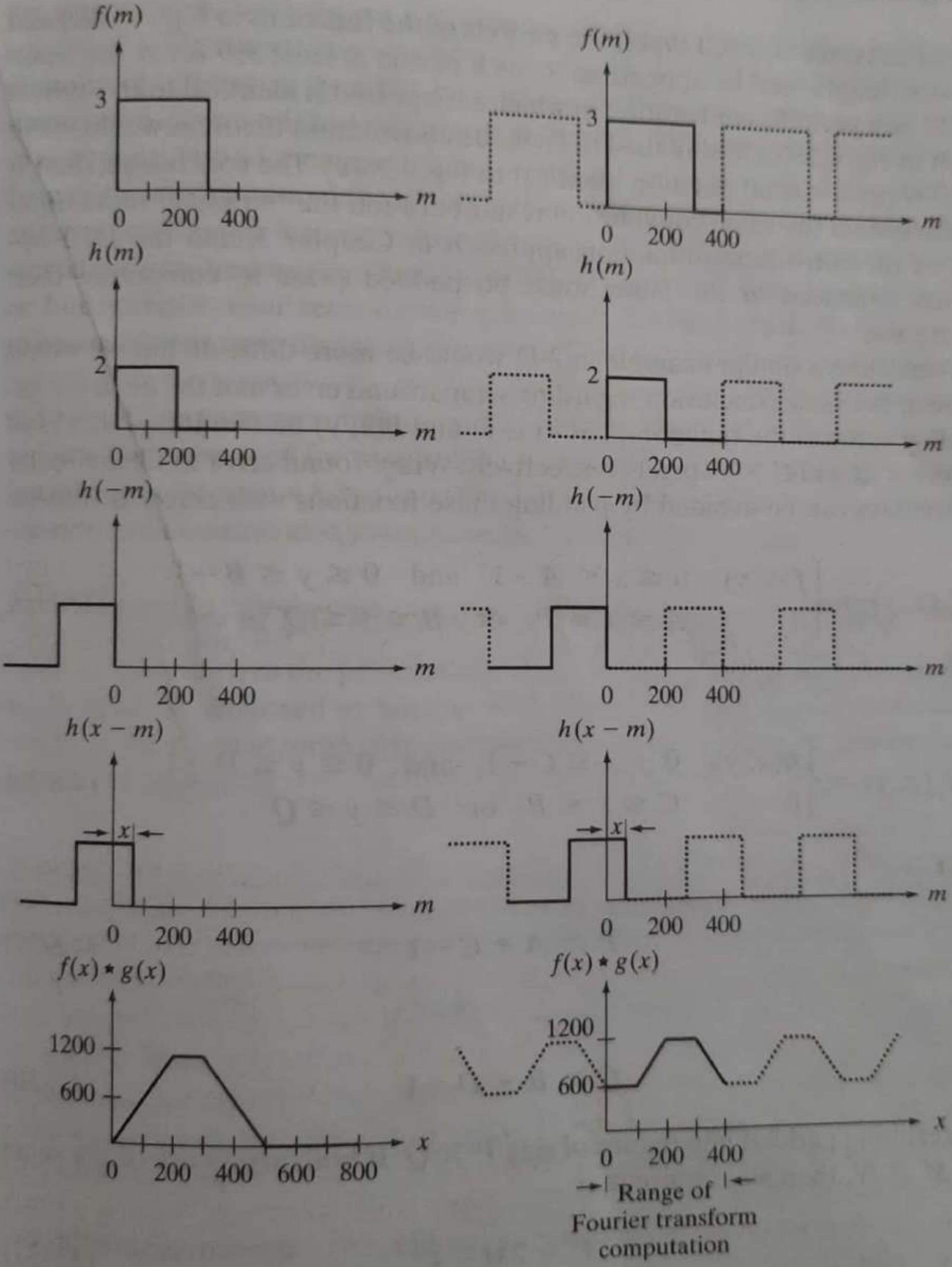
& the converse

$$\boxed{f(x,y)\, h(x,y) \overset{FT}{\Longleftrightarrow} F(u,v) \circledast H(u,v)}$$

↳ ㉕

* 1-D

$$f(x) \circledast h(x) = \sum_{m=0}^{M-1} f(x)\, h(x-m)$$



a f
b g
c h
d i
e j

FIGURE 4.28 Left column: convolution of two discrete functions obtained using the approach discussed in Section 3.4.2. The result in (e) is correct. Right column: Convolution of the same functions, but taking into account the periodicity implied by the DFT. Note in (j) how data from adjacent periods produce wraparound error, yielding an incorrect convolution result. To obtain the correct result, function padding must be used.

the inverse DFT, we would have obtained

- If we use DFT & the convoln theorem, to obtain the same result as in the left column of fig 4.28, we must take into account the periodicity inherent in the expression of DFT.

- This is equivalent to convolving the 2 periodic function (4.28 (f) & (g))

- The Procedure is ~~simple~~. same.

- Proceeding in the similar manner will yield the result shown in fig 4.28(j) which is obviously incorrect

- Since we are convolving 2 periodic signals, the result itself is periodic

- The closeness of the periods is such that they interfere with each other to cause ~~was~~ wrap around error

- This problem ~~is~~ can be solved by using zero padding method

- If we append ~~ze~~ zeros to both funs so that they have same length denoted by P: $P \geq A+B-1$ —— (276)

2-D

* Let $f(x,y)$ & $h(x,y)$ be 2 image arrays of sizes $A \times B$ & $C \times D$ respectively.

* wraparound error in their convolution can be avoided by padding these functions with zero's as follows

$$f_p(x,y) = \begin{cases} f(x,y) ; & 0 \le x \le A-1 \text{ \& } 0 \le y \le B-1 \\ 0 ; & A \le x \le P \text{ or } B \le y \le Q \end{cases} \longrightarrow \textcircled{27}$$

&

$$h_p(x,y) = \begin{cases} h(x,y) ; & 0 \le x \le C-1 \text{ \& } 0 \le y \le D-1 \\ 0 ; & C \le x \le P \text{ or } D \le y \le Q \end{cases} \longrightarrow \textcircled{28}$$

with

$$P \ge A+C-1 \text{ ——— } 29$$

$$\& \; Q \ge B+D-1 \text{ ——— } \textcircled{30}$$

- The resulting padded images are of size $P \times Q$. If both arrays are of the same size $M \times N$, then we require

$$P \ge 2M-1 \longrightarrow \textcircled{31}$$

$$\& \; Q \ge 2N-1 \quad \textcircled{32}$$

* If one or both of the fun's of 4.28 ⓐ & ⓑ were not zero at the end of the interval, then a discontinuity would be created when zeros were appended to the fun^n to eliminate wraparound error

* This is analogous to x'y a fun by a box, which in the freq domain would imply convol^n of original transform with a sinc fun.

* This would create frequency leakage caused by high freq components of sinc fun.

* This produces a blocky effect on image

* This can be reduced, by xling the sampled fun by another fun that tapers smoothly to near zero at both ends of the sampled record to ~~daper~~ dampen the sharp transiti (high freq comp) of the box.

* This approach is windowing or apodizing

① compute the linear convolution bet

$x[m,n] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ & $h[m,n] = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

thro' Matrix method

① size of $x[m,n] = M_1 \times N_1 = 2 \times 2$

$h[m,n] = M_2 \times N_2 = 2 \times 2$

∴ convoluted Matrix size will be $y[M,N] = M_3 \times N_3$

$M_3 = M_1 + M_2 - 1 = 2+2-1 = 3$

$N_3 = N_1 + N_2 - 1 = 2+2-1 \quad 3$

$y[m,n] = 3 \times 3$

② The block matrix

- no of block matrix depends on the no of rows of $x[m,n]$
- In this case $x[m,n]$ has 2 rows
  ∴ no of block matrix is 2
  H_0 & H_1

No of zeros to be appended in H_0 = No of columns in $h[m,n] - 1$

$x[m,n] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $\rightarrow$ used to form Ho $\rightarrow$ used to form H_1

③ steps in formation of block matrix Ho

(1) 1st element is inserted in H_0

$H_0 = [1 \quad 0]$ only one zero is inserted as

no of zeros = no of columns in h[m-n] - 1

2 - 1 = 1

(2) 2nd elem

$H_0 = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$

(3) $H_0 = \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 0 & 2 \end{bmatrix}$ element is shifted

③ **H_1**

$H_1 = \begin{bmatrix} 3 & 0 \\ 4 & 3 \\ 0 & 4 \end{bmatrix}$

⑤ Steps in the formation of block Toeplitz matrix

<u>no</u> of zeros to be appended in 'A'

= <u>no</u> of rows of $h[m,n]-1$

$$A = \begin{bmatrix} H_0, & 0 \\ H_1, & H_0 \\ 0, & H_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 4 & 3 & 2 & 1 \\ 0 & 4 & 0 & 2 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 4 & 3 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

0 → group of zeros

⑥ $$y[m,n] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 4 & 3 & 2 & 1 \\ 0 & 4 & 0 & 2 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 4 & 3 \\ 0 & 0 & 0 & 4 \end{bmatrix} \times \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 5 \\ 16 \\ 12 \\ 22 \\ 60 \\ 40 \\ 21 \\ 52 \\ 32 \end{bmatrix}$$

$$y[m,n] = \begin{bmatrix} 5 & 16 & 12 \\ 22 & 60 & 40 \\ 21, & 52, & 32 \end{bmatrix}$$

② circular convolution

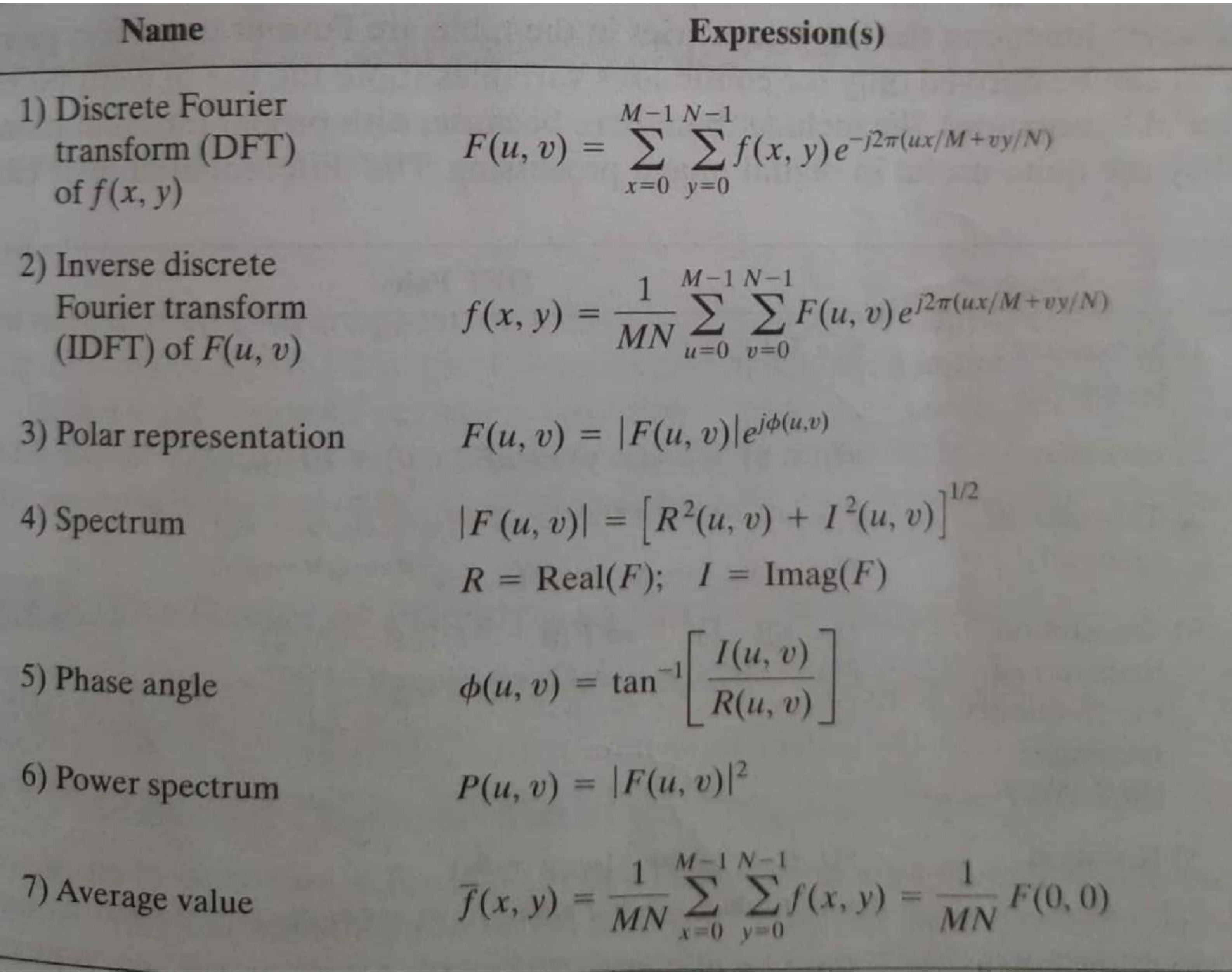
$x(m,n) = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ & $h(m,n) = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

① $H_0 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$ $H_1 = \begin{bmatrix} 3 & 4 \\ 4 & 3 \end{bmatrix}$

② $A = \begin{bmatrix} H_0 & H_1 \\ H_1 & H_0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$

③ $y = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 70 \\ 68 \\ 62 \\ 60 \end{bmatrix}$

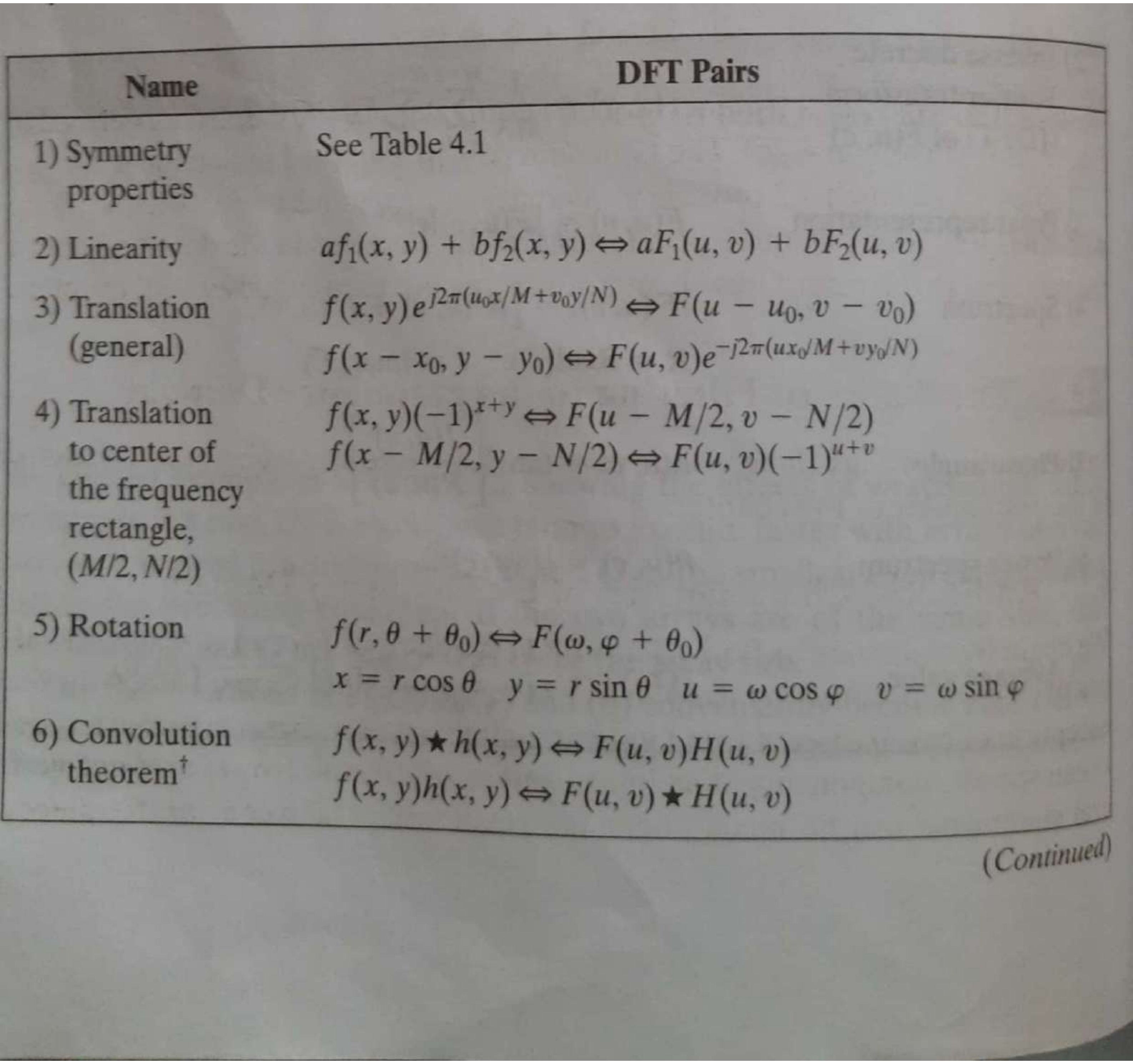
$y(m,n) = \begin{bmatrix} 70 & 68 \\ 62 & 60 \end{bmatrix}$



Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x, y)$	$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$
2) Inverse discrete Fourier transform (IDFT) of $F(u, v)$	$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$
3) Polar representation	$F(u, v) = \lvert F(u, v) \rvert e^{j\phi(u,v)}$
4) Spectrum	$\lvert F(u, v) \rvert = \left[R^2(u, v) + I^2(u, v) \right]^{1/2}$ $R = \text{Real}(F); \quad I = \text{Imag}(F)$
5) Phase angle	$\phi(u, v) = \tan^{-1}\left[\frac{I(u, v)}{R(u, v)} \right]$
6) Power spectrum	$P(u, v) = \lvert F(u, v) \rvert^2$
7) Average value	$\bar{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} F(0, 0)$

Name	**Expression(s)**
8) Periodicity (k_1 and k_2 are integers)	$F(u, v) = F(u + k_1M, v) = F(u, v + k_2N)$ $= F(u + k_1M, v + k_2N)$ $f(x, y) = f(x + k_1M, y) = f(x, y + k_2N)$ $= f(x + k_1M, y + k_2N)$
9) Convolution	$f(x, y) \bigstar h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)$
10) Correlation	$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n)h(x + m, y + n)$
11) Separability	The 2-D DFT can be computed by computing 1-D DFT transforms along the rows (columns) of the image, followed by 1-D transforms along the columns (rows) of the result. See Section 4.11.1.
12) Obtaining the inverse Fourier transform using a forward transform algorithm.	$MNf^*(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v)e^{-j2\pi(ux/M + vy/N)}$ This equation indicates that inputting $F^*(u, v)$ into an algorithm that computes the forward transform (right side of above equation) yields $MNf^*(x, y)$. Taking the complex conjugate and dividing by MN gives the desired inverse. See Section 4.11.2.

Table 4.3 summarizes some important DFT pairs. Although our focus is on discrete functions, the last two entries in the table are Fourier transform pairs that can be derived only for continuous variables (note the use of continuous variable notation). We include them here because, with proper interpretation, they are quite useful in digital image processing. The differentiation pair can



Name	DFT Pairs
1) Symmetry properties	See Table 4.1
2) Linearity	$af_1(x, y) + bf_2(x, y) \Leftrightarrow aF_1(u, v) + bF_2(u, v)$
3) Translation (general)	$f(x, y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0/M+vy_0/N)}$
4) Translation to center of the frequency rectangle, $(M/2, N/2)$	$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$
5) Rotation	$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$ $x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$
6) Convolution theorem[†]	$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$ $f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$

(Continued)

Name	DFT Pairs
7) Correlation theorem[†]	$f(x, y) \star h(x, y) \Leftrightarrow F^*(u, v)\, H(u, v)$ $f^*(x, y) h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$
8) Discrete unit impulse	$\delta(x, y) \Leftrightarrow 1$
9) Rectangle	$\text{rect}[a, b] \Leftrightarrow ab \dfrac{\sin(\pi ua)}{(\pi ua)} \dfrac{\sin(\pi vb)}{(\pi vb)} e^{-j\pi(ua+vb)}$
10) Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $j\dfrac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) - \delta(u - Mu_0, v - Nv_0)\right]$
11) Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $\dfrac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) + \delta(u - Mu_0, v - Nv_0)\right]$
The following Fourier transform pairs are derivable only for continuous variables, denoted as before by t and z for spatial variables and by μ and ν for frequency variables. These results can be used for DFT work by sampling the continuous forms.	
12) *Differentiation* (The expressions on the right assume that $f(\pm\infty, \pm\infty) = 0$.)	$\left(\dfrac{\partial}{\partial t}\right)^m \left(\dfrac{\partial}{\partial z}\right)^n f(t, z) \Leftrightarrow (j2\pi\mu)^m (j2\pi\nu)^n F(\mu, \nu)$ $\dfrac{\partial^m f(t, z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu, \nu); \dfrac{\partial^n f(t, z)}{\partial z^n} \Leftrightarrow (j2\pi\nu)^n F(\mu, \nu)$
13) *Gaussian*	$A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+\nu^2)/2\sigma^2}$ (A is a constant)

TABLE 4.
(Continu

[†] Assumes that the functions have been extended by zero padding. Convolution and correlation are associative, commutative, and distributive.

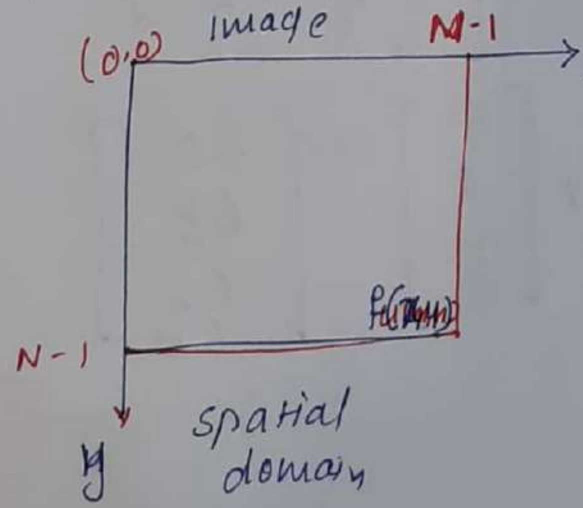
be used to derive the frequency-domain equivalent of the Laplacian defined in

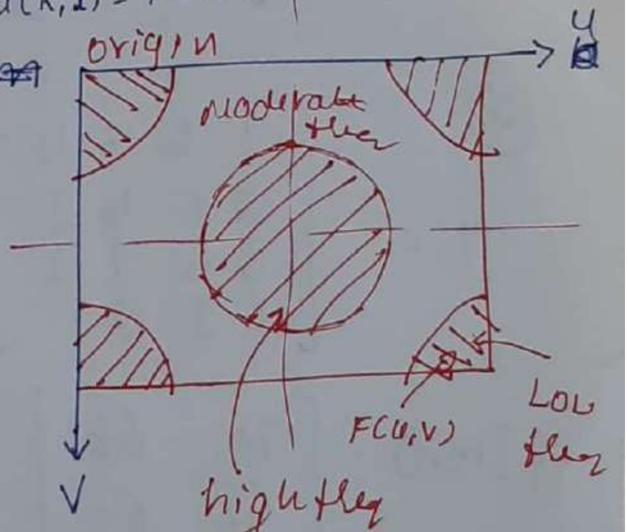
Understanding the image in Fourier Domain

* DFT tells us what frequencies are present in the image & their relative strengths.

* Frequency is directly related to the rate of change ∴ we can associate DFT with patterns of intensity variations in the image.

* Few observations $u(m,n) = f(x,y)$
$u(k,l) = F(u,v)$





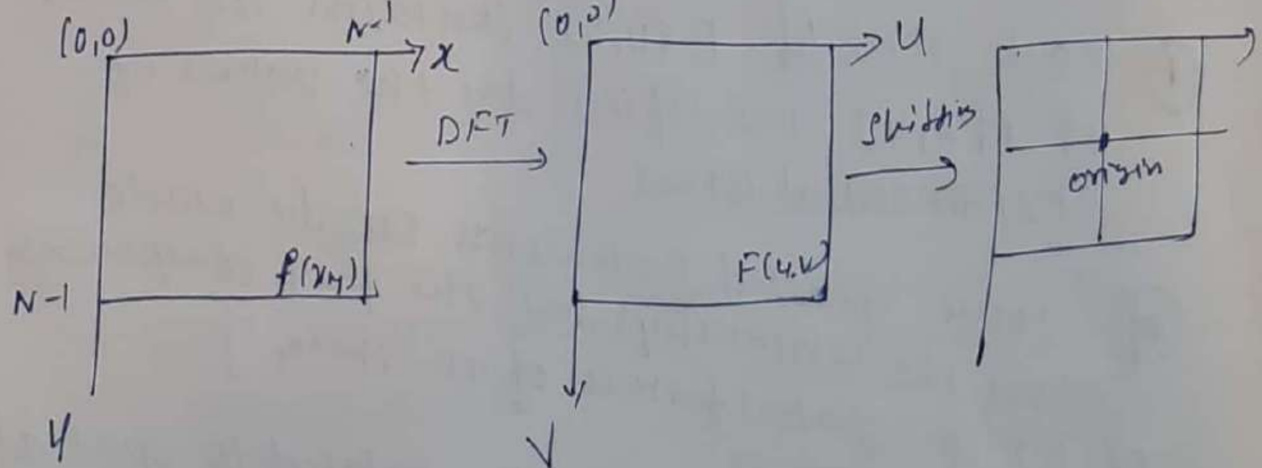
(i) $U=V=0$ is a DC component (zero freq)
$x=y=0$

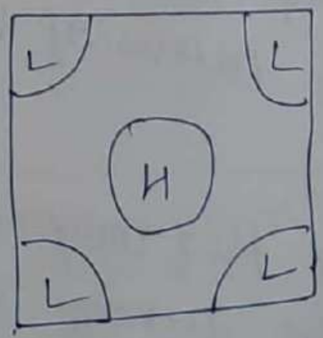
(ii) Near the origin $(x=y=0)$ of freq space, low frequencies exist which correspond to slowly varying components in the image
~~ex~~ background in any image is smooth grey-level variations

(iii) As we move away from origin, we encounter terms in freq space at higher freq's.
[faster & faster grey level variation in the image]
Edges of objects & other components (noise) of an images are characterized by abrupt change in grey level

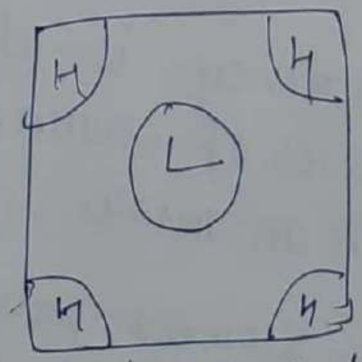
eg. ~~fundde~~ sudden change in grey level is boundary of a petal in an image

- To avoid problems with displaying complex-valued transform F(u,v) of an image f(x,y), a common approach is to display only the magnitude ~~|V(K,L)|~~ |F(u,v)| & ignore the phase of F(u,v).
- origin of the image is shifted to image centre





Before centralization



after centralization

- After shifting the origin to center of image, lowest freq comp are at centre & freq rises as we go away from centre

(18)

~~4.7.2 Frequen~~

4.7 The BASICS of Filtering in frequency domain

[image enhancement in freq domain : 4.3 Basic properties of freq domain
4.4 basic steps of filtering in freq
4.5 of Vipula]

4.7.1 Additional characteristics of freq domain

Let us consider the 2-D DFT eqn

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

(4).

- each term of F(u,v) contains all values of f(x,y) modified by the values of exponential terms

- [some general statements can be made about the relationship bet' the freq components of FT & spatial feature of an image]

- Frequency is directly related to spatial rates of change of intensity variations in an image

- The slowest varying freq component (u=v=0) is proportional to the average intensity of an image.

- As we move away from the origin of the transform, the low frequencies correspond to the slowly varying intensity components of ~~the~~ an image

- As we move further away from the origin, the higher frequencies begin to correspond to faster & faster intensity changes in the image [edges of objects & other components of image characterized by abrupt changes in intensity]

- Filtering techniques in freq domain are based on modifying the FT to achieve a specific objective & then computing the IDFT to get back to the image domain

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

  · WKT the 2 components of DFT are magnitude (spectrum) & the Phase angle.

- visual analysis of phase component is not very useful.

Frequency Domain filtering fundamentals

- Filtering in frequency domain consists of modifying the FT of an image & then computing the inverse transform to obtain the processed result.

- For a given digital image $f(x,y)$ of size $M \times N$, the basic filtering eq^n is of the form

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

where

$F^{-1} \rightarrow$ IDFT

$F(u,v) \rightarrow$ DFT of i/p image

$H(u,v) \rightarrow$ DFT of a filter fun

$g(x,y) \rightarrow$ filtered o/p image

the size of all the functions are $M \times N$ same as i/p image

- The filter fun, modifies the transform of the i/p image to yield a processed o/p $g(x,y)$.

- $H(u,v)$ is simplified considerably by using fun's that are symmetric about their center.

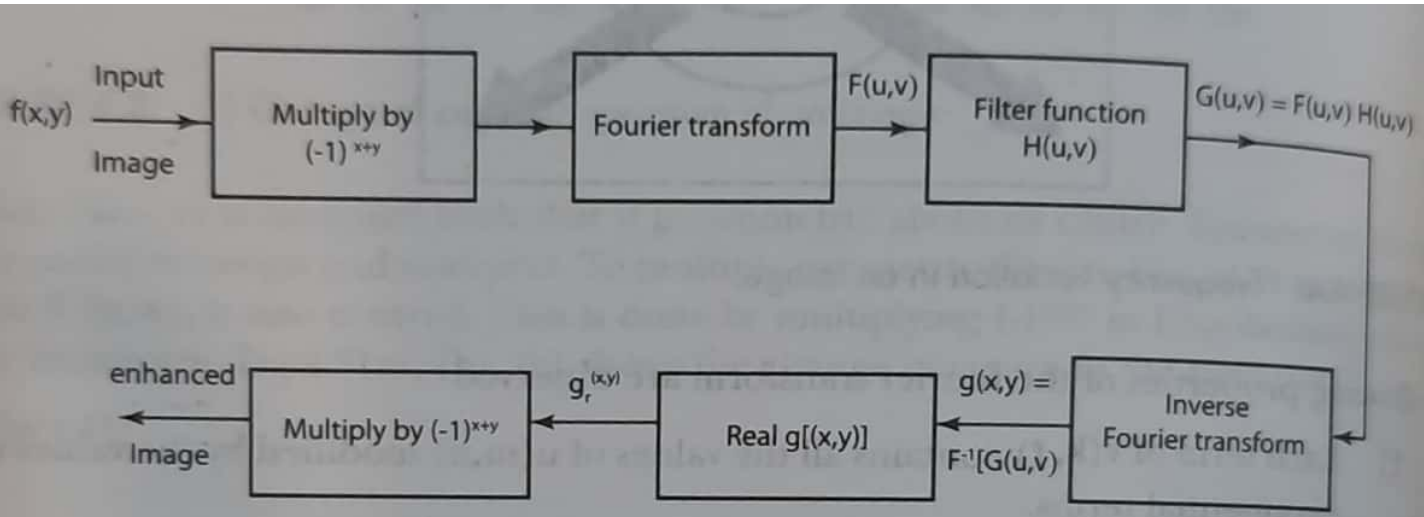


FIGURE 4.5: Block diagram of filtering in frequency domain

* This is accomplished by xling the i/p image by $(-1)^{x+y}$ prior to computing its transforms $f(x)(-1)^x \Longleftrightarrow F(u-M/2)$
[shifts the data so that $F(0)$ is at the center of the interval $[0, M-1]$

* One of the simplest filters we can construct is a filter $H(u,v)$ is '0' at the center of the transform & '1' elsewhere
* This filter would reject the dc term & pass all other terms of $F(u,v)$.

* WKT $F(0,0) = MN \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$

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

$\bar{f}$ = avg value

from above eqn WKT the dc-term is responsible for the average intensity of an image. (fig 4.30)

* so setting it to zero will reduce the avg intensity of the o/p image to zero
* The image becomes much darker.
[An avg of zero ⇒ existence of -ve intensity)

FIGURE 4.29 (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

FIGURE 4.30 Result of filtering the image in Fig. 4.29(a) by setting to 0 the term $F(M/2, N/2)$ in the Fourier transform.

* low frequencies in the transform are related to slowly varying intensity components in the image (walls of room / cloudless sky in an outdoor scene)

& high frequencies are caused by sharp transitions in intensity such as edges & noise

* ∴ we would expect that a filter H(u,v) that attenuates high freq's while passing low freq's [LPF] would blur an image.

while a filter with opposite property [high pass filter] would enhance sharp details but cause a reduction in constrast in the image (4.31 fig)

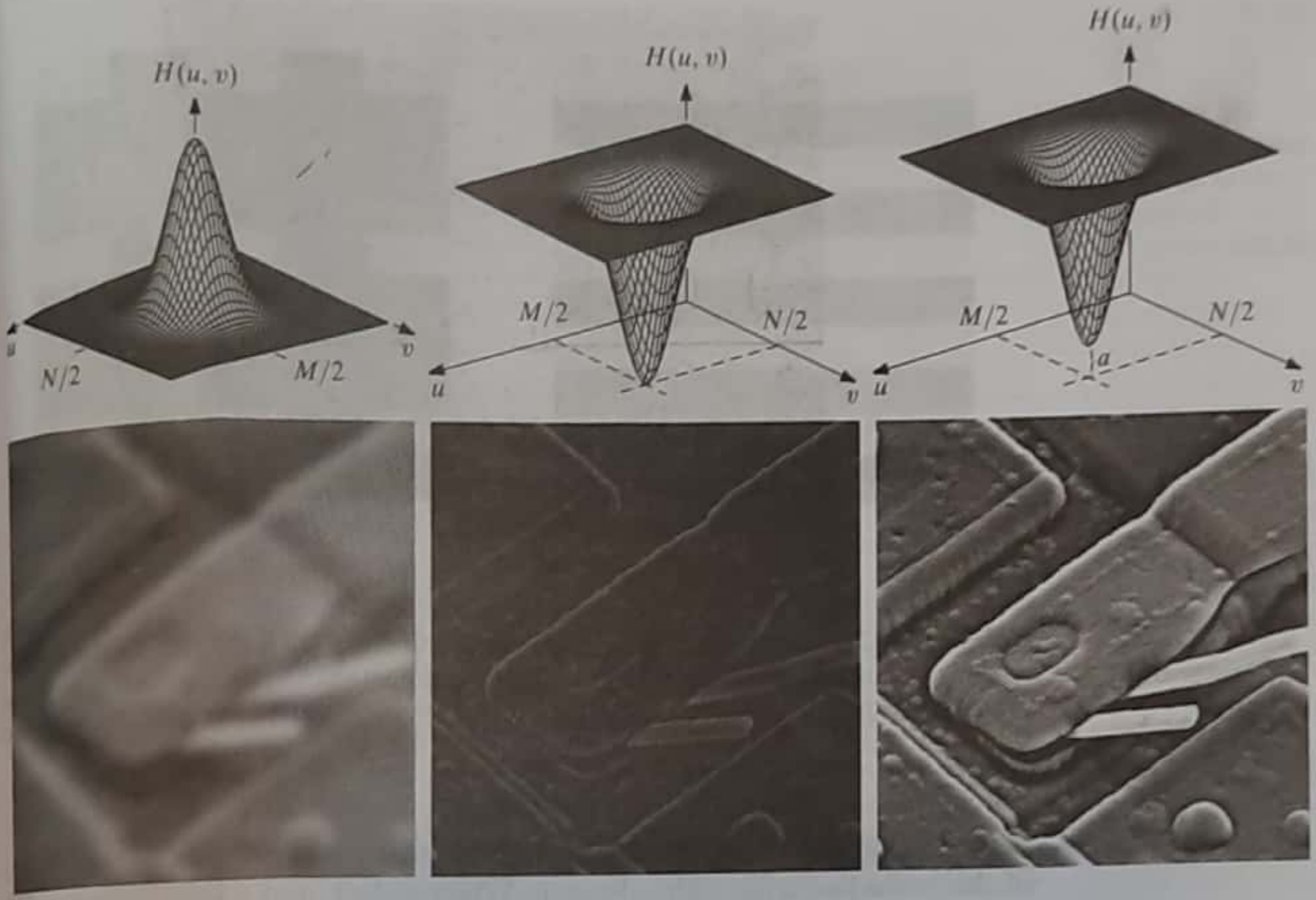
[HPF eliminates the dc term]

* eq ⑪ $g(x,y) = F^{-1}[H(u,v), F(u,v)]$

product of 2 fun's in freq domain

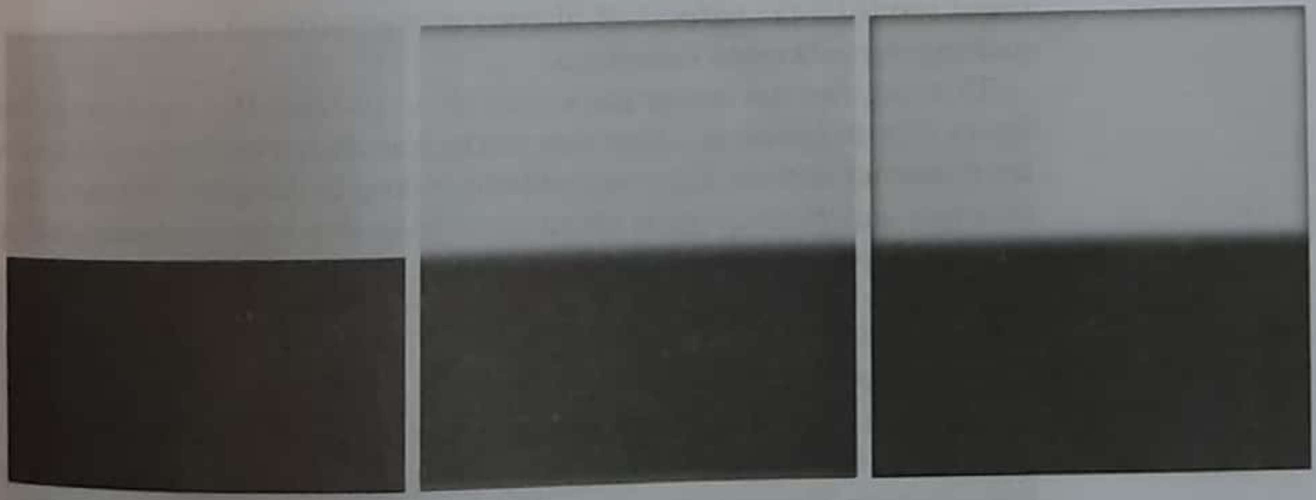
= convol^n in spatial domain.

* If the functions in questions are not padded we can expect wraparound error (discussed earlier



a b c
d e f

FIGURE 4.31 Top row: frequency domain filters. Bottom row: corresponding filtered images obtained using Eq. (4.7-1). We used $a = 0.85$ in (c) to obtain (f) (the height of the filter itself is 1). Compare (f) with Fig. 4.29(a).



a b c

FIGURE 4.32 (a) A simple image. (b) Result of blurring with a Gaussian lowpass filter without padding. (c) Result of lowpass filtering with padding. Compare the light area of the vertical edges in (b) and (c).

* when we apply eq ① without padding
fig 4.32(b) then the image when filtered '
using Gaussian LPF would result in
blurring. {

* blurring is not uniform [top white edges are blurred but side white edges are not fig 4.32 (b)]

* so padding the i/p image ~~would results~~ before applying eq ① results in the filtered image where blurring is uniform.

* [padding the image with 0s can create a uniform border around the periodic seqn
fig 4.33 & then convolving the blurring fun with the padded mosaic gives correct result]

* padding is done in spatial domain
* eq ① involves a filter that can be specified either in spatial or freq domain

* the way to handle padding of a frequency domain filter is to construct the filter to be of the same size as the image

21

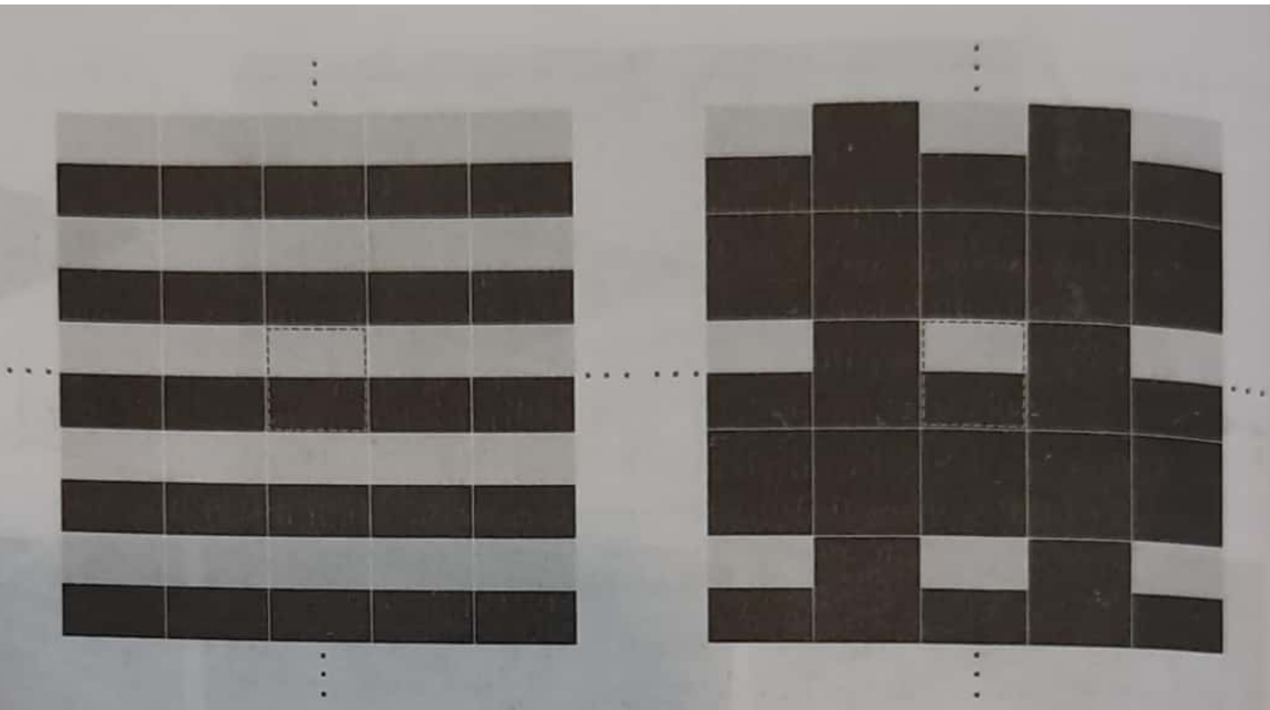
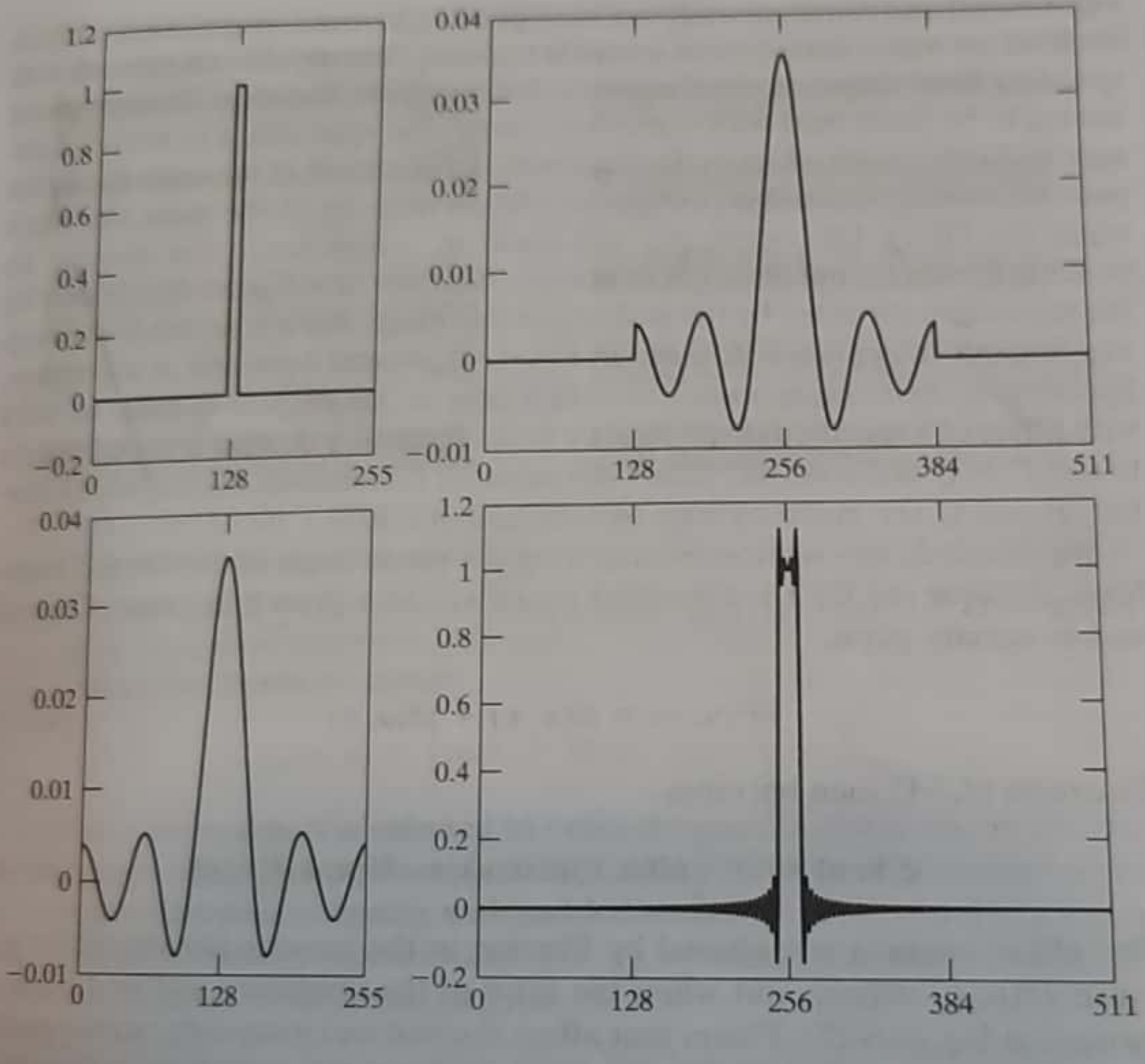


FIGURE 4.33 2-D image periodicity inherent in using the DFT. (a) Periodicity without image padding. (b) Periodicity after padding with 0s (black). The dashed areas in the center correspond to the image in Fig. 4.32(a). (The thin white lines in both images are superimposed for clarity; they are not part of the data.)



a c
b d

FIGURE 4.34 (a) Original filter specified in the (centered) frequency domain. (b) Spatial representation obtained by computing the IDFT of (a). (c) Result of padding (b) to twice its length (note the discontinuities). (d) Corresponding filter in the frequency domain obtained by computing the DFT of (c). Note the ringing caused by the discontinuities in (c). (The curves appear continuous because the points were joined to simplify visual analysis.)

compute IDFT of the filter to obtain the corresponding spatial filter.

- pad that filter in spatial domain & then compute its DFT to return to the freq domain

(fig 4.3.4)

* to work with specified filter shapes in freq domain w/o having to be concerned with truncation issues

- one approach is to zero-pad images & then create filters in freq domain to be of the same size when using the DFT

* Let us analyze the phase angle of the filtered transform

$\because$ DFT is complex & can be expressed as

$$F(u,v) = R(u,v) + jI(u,v) \rightarrow ②$$

Then eq ①

$$g(x,y) = F^{-1}\left[H(u,v)\,R(u,v) + jH(u,v)\,I(u,v)\right] \rightarrow ③$$

* ~~filters~~

- phase angle is not altered by filtering because H(u,v) cancels out when the ratio of imaginary & real part is formed $\left[\frac{I(u,v)}{R(u,v)}\right]$

- filters that affect real & imaginary parts equally & thus have no effect on the phase & are called zero-phase-shift filter.

4.7.3

2)

from multiplying the angle array in Eq. (4.6-15) by 0.5, without changing

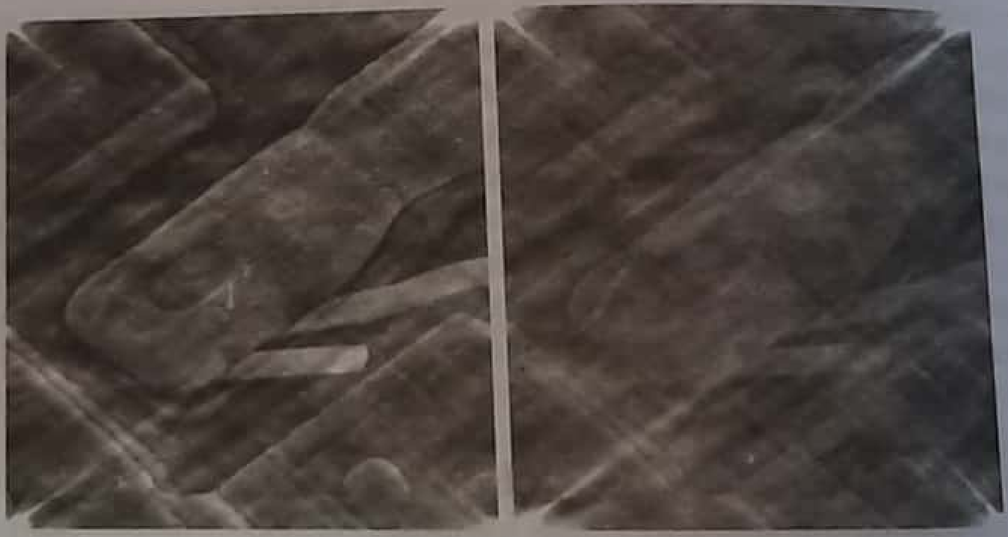

a b

FIGURE 4.35 (a) Image resulting from multiplying by 0.5 the phase angle in Eq. (4.6-15) and then computing the IDFT. (b) The result of multiplying the phase by 0.25. The spectrum was not changed in either of the two cases.

4.7.3 Summary of Steps for Filtering in the Frequency Domain

The material in the previous two sections can be summarized as follows:

1. Given an input image $f(x, y)$ of size $M \times N$, obtain the padding parameters P and Q from Eqs. (4.6-31) and (4.6-32). Typically, we select $P = 2M$ and $Q = 2N$.
2. Form a padded image, $f_p(x, y)$, of size $P \times Q$ by appending the necessary number of zeros to $f(x, y)$.
3. Multiply $f_p(x, y)$ by $(-1)^{x+y}$ to center its transform.
4. Compute the DFT, $F(u, v)$, of the image from step 3.
5. Generate a real, symmetric filter function, $H(u, v)$, of size $P \times Q$ with center at coordinates $(P/2, Q/2)$.[†] Form the product $G(u, v) = H(u, v)F(u, v)$ using array multiplication; that is, $G(i, k) = H(i, k)F(i, k)$.
6. Obtain the processed image:

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

   where the real part is selected in order to ignore parasitic complex components resulting from computational inaccuracies, and the subscript p indicates that we are dealing with padded arrays.
7. Obtain the final processed result, $g(x, y)$, by extracting the $M \times N$ region from the top, left quadrant of $g_p(x, y)$.

As
ing
the
in g
func
cent
men

Figure 4.36 illustrates the preceding steps. The legend in the figure explains the source of each image. If it were enlarged, Fig. 4.36(c) would show black dots interleaved in the image because negative intensities are clipped to 0 for display. Note in Fig. 4.36(h) the characteristic dark border exhibited by lowpass filtered images processed using zero padding.

4.7.4 correspondence bet' filtering in spatial & freq domain.

- The link between filtering in the spatial & freq domains is the convolution Theorem
- WKT, filtering in freq domain is defined as xlion of a filter function $H(u,v)$ times $F(u,v)$, the FT of i/p Image
- Given a filter $H(u,v)$, if we want to find its equivalent representation in spatial domain
- If let $f(x,y) = \delta(x,y)$

  FT of $\delta(x,y) = 1$

  $\therefore F(u,v) = 1$, Then

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

  then filtered o/p from above eqn

  is $F^{-1}\{H(u,v)\}$.
- inverse FT of freq domain filter which is corresponding filter in the spatial domain
- given spatial filter, we can obtain its freq domain repn. by taking FT of the spatial filter

$$h(x,y) \xLeftrightarrow{FT} H(u,v) \rightarrow ④$$

Impulse response

If the quantities in eq ④ are finite, such filters are called as FIR filter

- ⁎ one way to take advantage of the properties of both domains is to specify a filter in freq domain, compute its IDFT, & then use the resulting full-size spatial filter as a guide for constructing smaller spatial filter masks

- ⁎ Let us discuss, by using Gaussian filters, how freq domain filters can be used as guides for specifying the coefficients of some of the small masks [box filter, weighed avg, Sobel, Roberts laplacian)

- ⁎ Filters based on Gaussian functions are of particular interest, because, both the forward & inverse FT of a real Gaussian fun are Gaussian fun's

- ⁎ Let H(u) ⟹ denoted 1-D freq domain Gaussian filter

$$H(u) = A e^{-u^2/2\sigma^2} \longrightarrow ⑤$$

where σ = std. deviation of Gaussian curve

* The corresponding filter in spatial domain is obtained by taking IFT of H(u)

$$h(x) = \sqrt{2\pi}\,\sigma A e^{-2\pi^2\sigma^2 x^2} \longrightarrow ⑥$$

- These eqns are important because

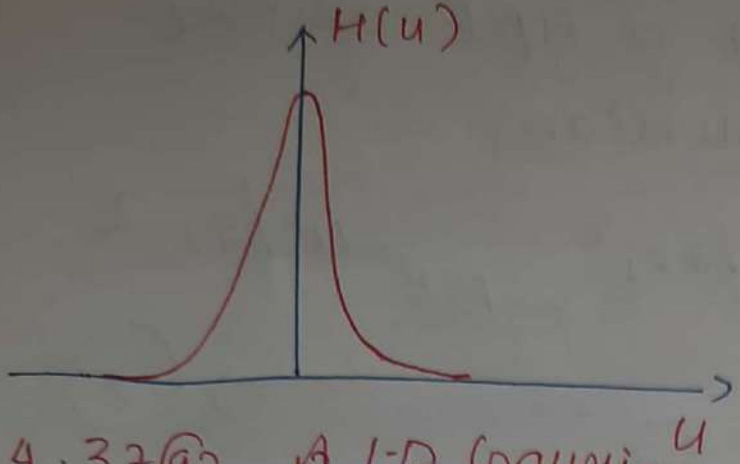
* (i) They are FT pair, both components of which are Gaussian & real.
  ∴ no need to be concerned with complex nos. Gaussian curves are intuitive & easy to manipulate

  (ii) The fun behaves reciprocally. when H(u) has a broad profile (large value of σ), h(x) has a narrow profile & viceversa

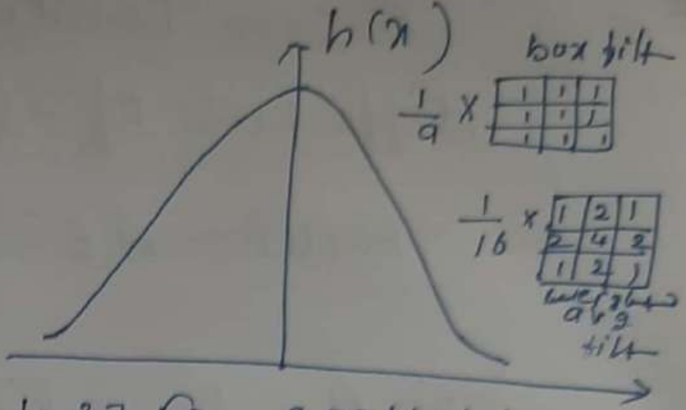
  - if σ approaches infinity, then H(u) tends towards constant fun & h(x) tends towards an impulse which implies no filtering in freq & spatial domains respectively

* fig 4.37 ⓐ & ⓑ shows plots of Gaussian LPF in freq domain & the corresponding filter in spatial domain

with small filter



4.37 ⓐ A 1-D Gaussian LPF in freq domain



4.37 ⓑ spatial LPF

- * If we want to use the shape of h(x) in fig 4.37 ⓑ as guide for specifying coefficients of a small spatial mask.
- * the similarity bet' 2 filters is that all their values are +ve
- * ∴ we conclude that we can implement LP Filtering in spatial domain by using a mask with all positive coefficients
- * The narrower, the freq domain filter, the more it will attenuate the low freq's, resulting in ~~[illegible]~~ blurring
- * In spatial domain this means that a larger mask must be used to ~~[illegible]~~ blurring
- * more complex filters can be constructed using the basic Gaussian fun of eq ⑤ H(u)

* eg we can construct a HPF as the difference of Gaussians

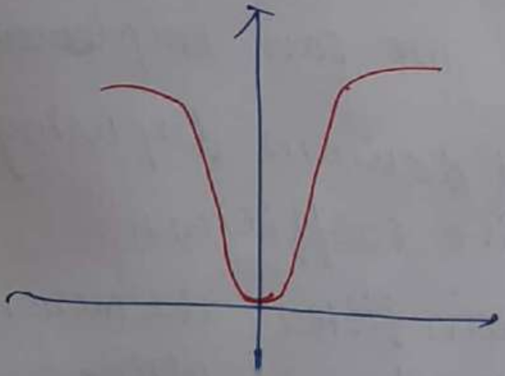
$$H(u) = Ae^{-u^2/2\sigma_1^2} - Be^{-u^2/2\sigma_2^2} \hookrightarrow ⑦$$

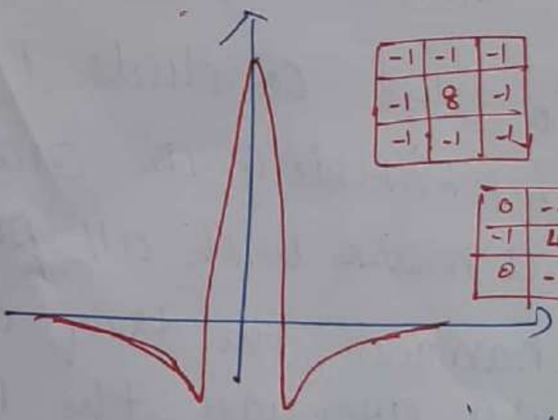
with $A \geq B$ & $\sigma_1 > \sigma_2$

* The corresponding filter in spatial domain is

$$h(x) = \sqrt{2\pi}\,\sigma_1 A e^{-2\pi^2\sigma_1^2 x^2} - \sqrt{2\pi}\,\sigma_2 B e^{-2\pi^2\sigma_2^2 x^2} \hookrightarrow ⑧$$

* fig 4.37 ⓒ & ⓓ shows the plot





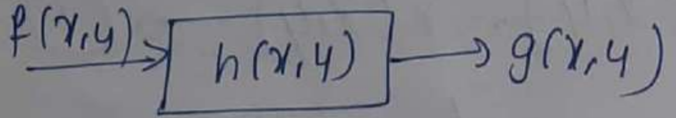
Laplacian

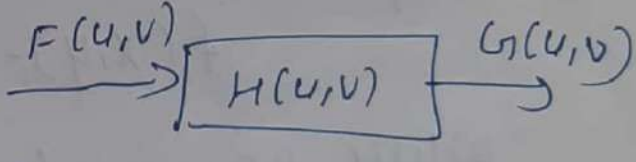
-1	-1	-1
-1	8	-1
-1	-1	-1

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

* The most important feature here is that h(x) has a +ve center term with -ve terms on either side
* These 2 masks are sharpening filters which are now HPF
* In spatial domain, filtering is implemented by convolution bet' i/p image & filter

- convolution filtering with small filter mask is preferred ∵ of speed & ease of implementation in H/w
- But filtering is more intuitive in freq domain,
- Here filtering is implemented by xlion of FT of i/p image & TF of a filter





$g(x,y) = f(x,y) * h(x,y) \xleftrightarrow{F} G(u,v) = F(u,v) \cdot H(u,v)$

↑ Spatial filt

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

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

↑ filter TF

Spatial

freq

Homomorphic Filtering

* Homomorphic filtering is a freq domain procedure to improve the appearance of an image by (a) Grey level range compression
(b) Contrast enhancement

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

* Reflectance model is

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

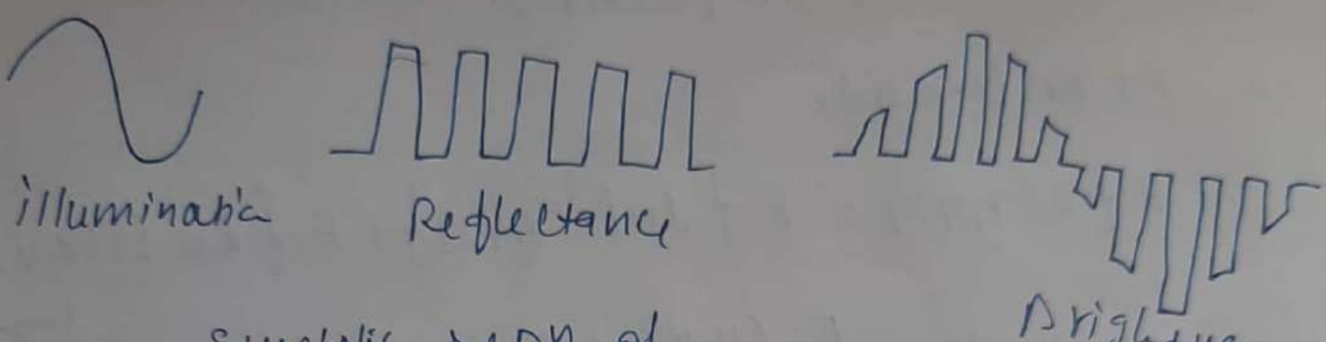
where $f(x,y)$ = Brightness of an image
$i(x,y)$ = illumination component
$r(x,y)$ = reflectance component

* some cases when the scene is not illuminated properly, or camera angle is not correct, some part of the image appears dark.

* In order to improve these types of images, reflectance & illumination has to be treated independently

(*) $i \rightarrow$ slowly varying $\Rightarrow$ low freq component
illumination changes "slowly" across the scene, Thus it is related to low freq

(2) $r \rightarrow$ fast varying ⟹ High freq component. surface reflection changes "sharply" across the scene. Thus it is associated to high freq



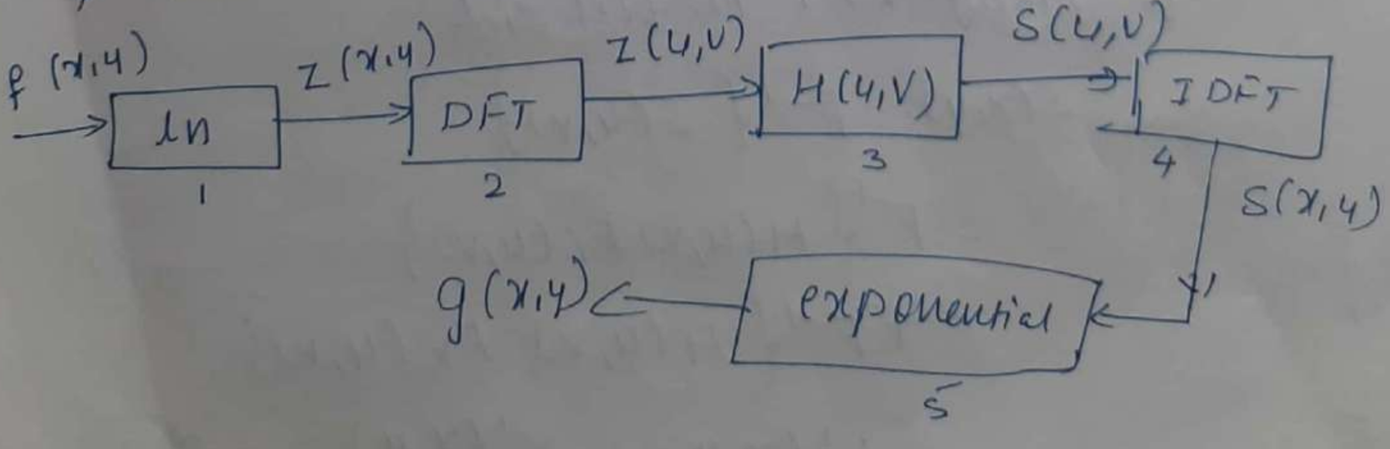
Symbolic repn of reflectance model

* For image enhancement, illumination & reflectance have to be treated separately which is not possible in freq domain as

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

* To separate the reflectance & illumination component, Homomorphic filters are used

* The block dig is shown below



1. Take natural logarithm of i/p image

$$z(x,y) = \ln[f(x,y)]$$

$$= \ln[i(x,y) \cdot r(x,y)] \longrightarrow \textcircled{3}$$

$$= \ln[i(x,y)] . \ln[r(x,y)]$$

2. FT on both side

$$F\{z(x,y)\} = F\{\ln[i(x,y)]\} + F\{\ln[r(x,y)]\} \longrightarrow \textcircled{4}$$

$$Z(u,v) = F_i(u,v) + F_r(u,v) \longrightarrow \textcircled{4'}$$

here $Z(u,v) = F\{z(u,v)\}$

$$F_i(u,v) = F\{\ln[i(x,y)]\}$$

$$F_r(u,v) = F\{\ln[r(x,y)]\}$$

3. X'ly with filter $H(u,v)$ with eq $\textcircled{4'}$

$$S(u,v) = H(u,v)\, Z(u,v)$$

$$= H(u,v)\, F_i(u,v) + H(u,v)\, F_r(u,v) \longrightarrow \textcircled{5}$$

4. The filtered image in spatial domain is taking IFT on both side

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

$$= F^{-1}\{H(u,v)\, F_i(u,v)\} + F^{-1}\{H(u,v)\, F_r(u,v)\} \longrightarrow \textcircled{6}$$

$$= i'(x,y) + r'(x,y) \longrightarrow \textcircled{7}$$

where

$$i'(x,y) = F^{-1}\{H(u,v)\,F_i(u,v)\} \rightarrow ⑧$$

& $r'(x,y) = F^{-1}\{H(u,v)\,F_r(u,v)\} \rightarrow ⑨$

⑤ Take inverse log transform

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

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

$$= i_0(x,y) \cdot r_0(x,y) \longrightarrow ⑩$$

where $i_0(x,y) = e^{i'(x,y)} \longrightarrow ⑪$

& $r_0(x,y) = e^{r'(x,y)} \longrightarrow ⑫$

are illumination & reflectance components of the o/p (processed) image

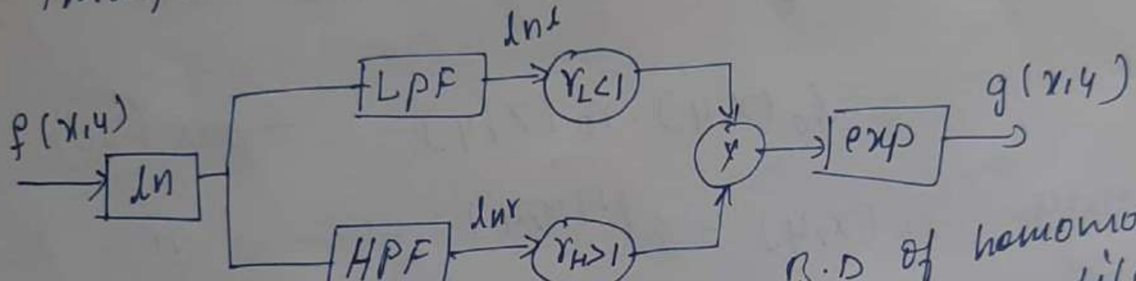
$g(x,y) =$ enhanced image

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

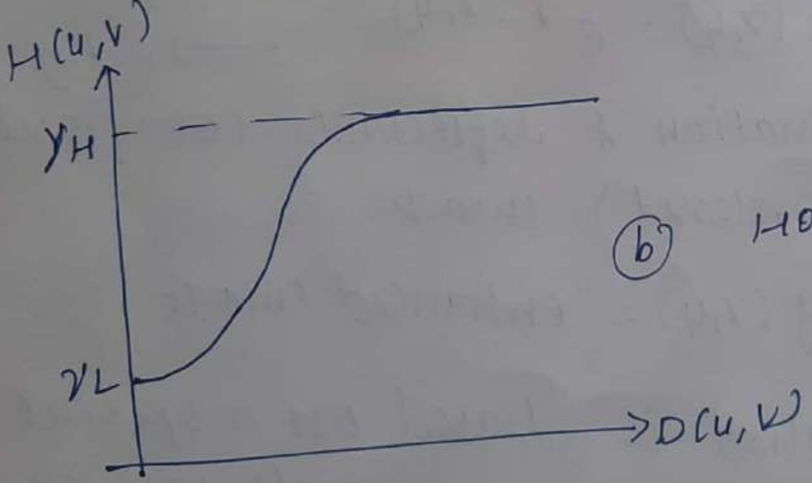
- The homomorphic filter fun $H(u,v)$ is indicated in eq ⑤.

* illumination component of an image is characterised by slow spatial variations while the reflectance component tends to vary abruptly, particularly at the junctions of dissimilar objects.

* The goal of homomorphic filtering is to suppress low frequencies associated with i/p image so that the net effect is enhacement enhanced.



B.D of homomorphic filter



(b) Homomorphic filter Transfer fun

* To achieve the above mentioned goal, a filter has to be designed in such a way that illumination component is suppressed & reflectance is enhanced as shown in above B.D

* Low freq's of FT of a log of an image are associated with illumination & high freq's are associated with reflectance

* Although these are approximate associations but can be used for image enhancement

* Transfer fun is controlled in such a way that low freq's are attenuated & high freq's are passed untouched as shown in fig ⑥.

* fig ⑥ shows the cross section of such a filter

* ~~The~~ If parameters γ_L & γ_H are choosen so that

$\gamma_L < 1 \Rightarrow$ tends to attenuate the contribution made by low freq's (illumination)

& $\gamma_H > 1 \Rightarrow$ amplify the contribution made by high freq's (reflectance)

* The net result is simultaneous dynamic range compression & contrast enhancement

* using a slightly modified form of the Gaussian HPF yields to

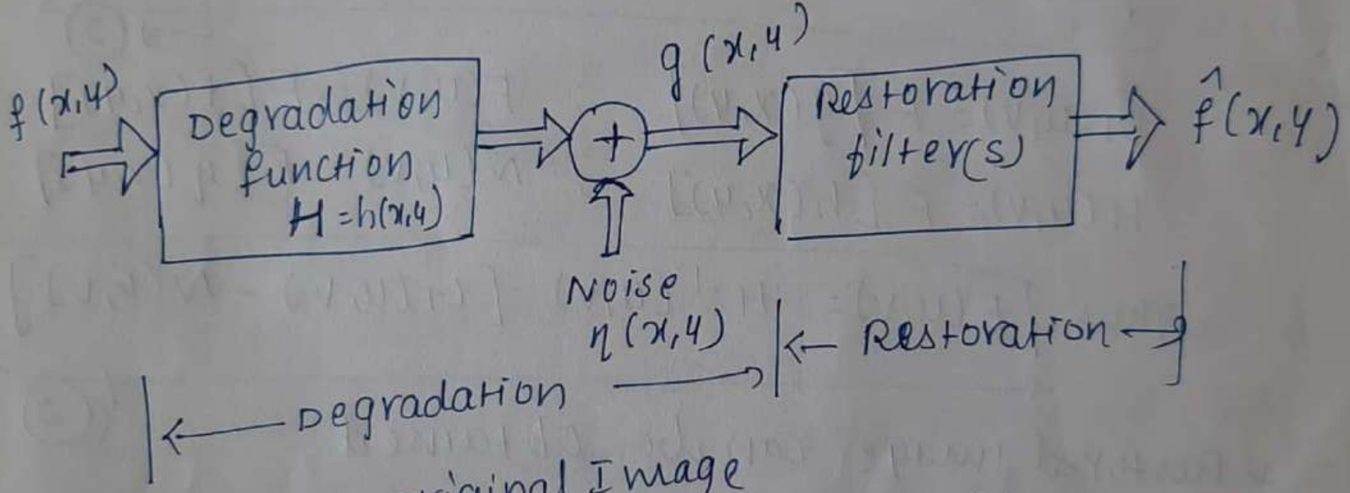
$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c[D^2(u,v)/D_0^2]}\right] + \gamma_L \rightarrow ⑱$$

IMAGE RESTORATION

5.1 A Model of Image degradation/Restoration Process

- Restoration is the process of inverting a degradation using knowledge about its nature

- Fig 5.1 below shows a model for the degradation/restoration process.



$f(x,y)$ = original Image
$h(x,y)$ = degradation function H
$\eta(x,y)$ = additive noise term.
$g(x,y)$ = degraded & noisy image
$\hat{f}(x,y)$ = estimate of the original image ie restored image

- The objective of restoration process is to estimate $\hat{f}(x,y)$ from the degraded version $g(x,y)$, when some knowledge of degradation function 'H' & noise 'η' is there.

①

* The degraded image $g(x,y)$ can be mathematically expressed as

$$\boxed{g(x,y) = h(x,y) * f(x,y) + \eta(x,y)} \rightarrow ①$$

spatial domain

$* \Rightarrow$ convoln.

* An equivalent freq domain representation is

$$\boxed{G(u,v) = H(u,v) F(u,v) + N(u,v)} \rightarrow ②$$

$G(u,v) = F[g(x,y)]$; $F(u,v) = F[f(x,y)]$

$H(u,v) = F[h(x,y)]$; $N(u,v) = F[\eta(x,y)]$

Thus $$\boxed{F(u,v) = H^{-1}(u,v) \cdot [G(u,v) - N(u,v)]} \rightarrow ③$$

- Restored image can be obtained by eq ③.

* The problems in implementing this eqn is

(1) The noise N is unknown. only the statistical properties of noise can be known.

(2) The operation H is singular or ill posed It is very difficult to estimate H

5.2 Noise Models

- The principal sources of noise in digital image arise during image acquisition and/or transmission
- The performance of imaging sensors is affected by a variety of factors such as environmental conditions ~~such as~~ during image acquisition & by the quality of the sensing element themselves
- eg. when acquiring images with a CCD camera, light levels & sensor temperature are major factors affecting the amount of noise in the resulting image.
- Images are corrupted during transmission due to interference in the channel used for trion. eg. an image tred using a wireless N/w might be corrupted as a result of lightning or other atmospheric disturbance

5.2.1 Spatial & frequency properties of noise →

~~* spatial characteristics of noise~~

spatial & freq characteristics of noise are as follows:

(1) Noise is assumed to be 'white noise' i.e. fourier spectrum of noise is constant

②

(2) Noise is assumed to be independent in spatial domain. Noise is uncorrelated with image ie, there is no correlation bet' pixel value of image & value of noise components

- The spatial noise descriptor is the statistical behaviour of the intensity values in the noise component
- Noise intensity is considered as a random variable characterized by a certain probability density function (PDF)

- Frequency properties refer to the freq content of noise in the Fourier sense

eg when the Fourier spectrum of noise is constant, the noise usually is called white noise.

5.2.2 Some important Noise Probability Density functions

Let us discuss the most common PDF's found in image processing applications

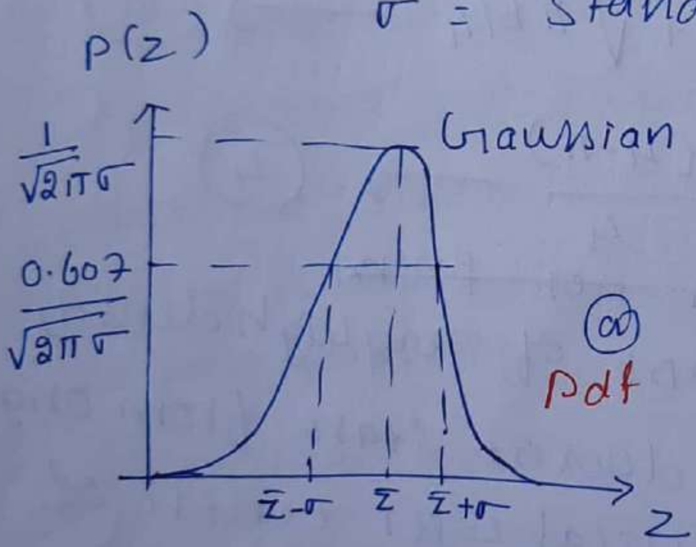
① Gaussian noise:

- Gaussian noise models (normal noise models) are used frequently in practice
- . The PDF of a Gaussian random variable 'z' is given by

* The PDF of a Gaussian random variable 'z' is given by

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

where
z = intensity values
$\bar{z}$ = mean (average) value of z. (we can use also μ)
σ = standard deviation



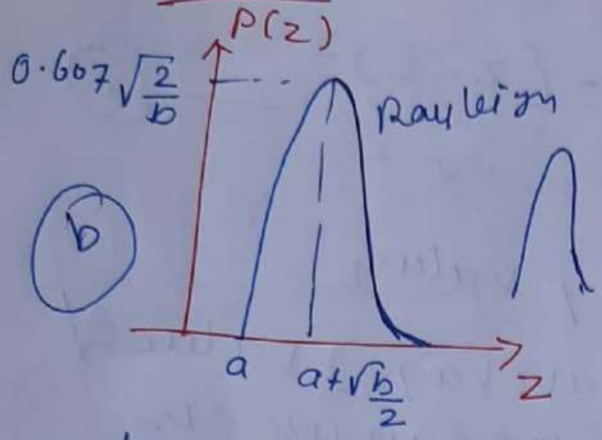
* The plot of this fun is shown in fig ⓐ

* when z is described by eq ⓑ, 70% of its value will be in the range $[(\bar{z}-\sigma), (\bar{z}+\sigma)]$ & about 95% will be in the range $[(\bar{z}-2\sigma), (\bar{z}+2\sigma)]$

* DFT of gaussian noise is another gaussian process. ∴ this property of gaussian noise makes it most oftenly used noise model

* ~~ef.~~ where gaussian model is used are electronic ckts noise, sensor noise due to low illumination or high temp, poor illumination

② Rayleigh noise



* The PDF of Rayleigh noise is given by

$$P(z) = \begin{cases} \frac{2}{b}(z-a)\,e^{-\frac{(z-a)^2}{b}} & ;\ \text{for } z \geq a \\ 0 & ;\ \text{for } z < a \end{cases} \hookrightarrow ②$$

* The mean & variance of this density are given by

$$\bar{z} = a + \sqrt{\pi b/4} \longrightarrow ③$$

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

* ~~displacement is not from~~

* fig ⓑ shows the PDF of Rayleigh density
* Note that curve doesn't start from origin & is not symmetrical WRT centre of curve
* The basic shape of Rayleigh density is skewed to the right. & ∴ can be useful for approximating skewed histograms

③ Erlang (Gamma) Noise

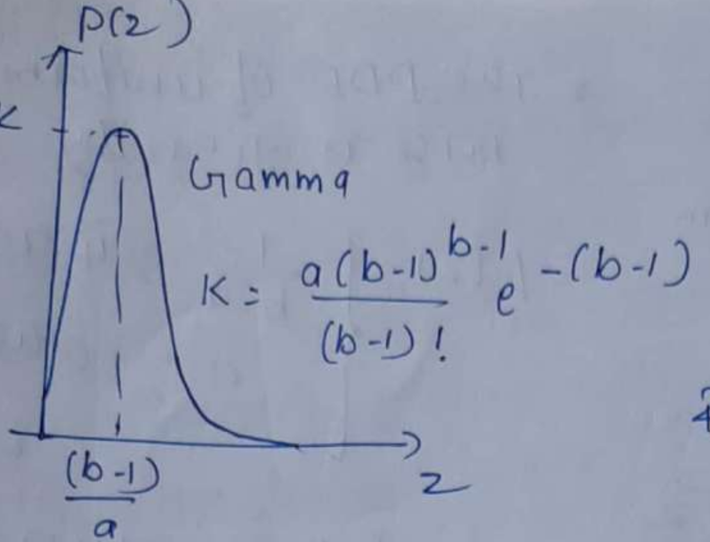
The PDF of Erlang noise is given by

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!}\,e^{-az} & ;\ \text{for } z \geq 0 \\ 0 & ;\ \text{for } z < 0 \end{cases} \longrightarrow ⑤$$

a & b are +ve integers a>0 & b= +ve integer

! ⇒ factorial

fig ⓒ



* The mean & varianc of this density are given by

$$\bar{z} = \frac{b}{a} \longrightarrow ⑥$$

& $$\sigma^2 = \frac{b}{a^2} \longrightarrow ⑦$$

* eq ⑤ is referred to as the gamma density, ~~strictly~~ strictly speaking this is correct only when the denominator is the gamma fun $\Gamma(b)$.

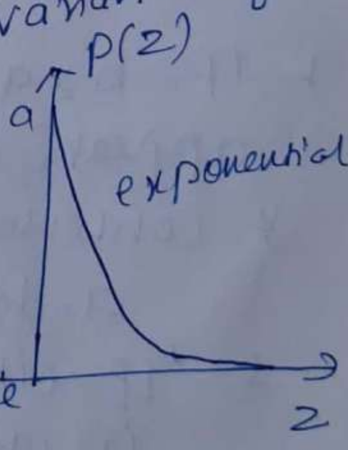
* when the denominator is as shown, the density is more appropriately called the Erlang density

④ Exponential noise

The PDF of exponential noise is given by

$$p(z) = \begin{cases} ae^{-az} & ; \text{ for } z \geq 0 \\ 0 & ; \text{ for } z < 0 \end{cases} \longrightarrow ⑧$$

where $a > 0$, The mean & variance of this density fun are



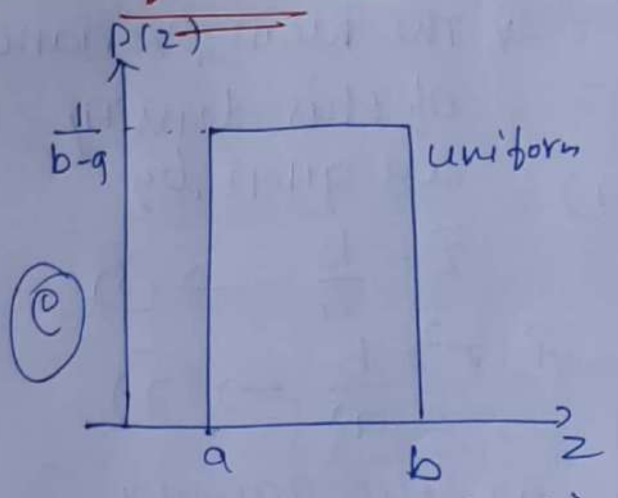
$$\bar{z} = \frac{1}{a} \longrightarrow ⑨$$

& $$\sigma^2 = \frac{1}{a^2} \longrightarrow ⑩$$

this PDF is a special case of the Erlang PDF, with $b=1$ & shown in fig ⓓ

(ii)

⑤ uniform noise



* The PDF of uniform noise is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & ; \text{ if } a \le z \le b \\ 0 & ; \text{ otherwise} \end{cases} \rightarrow ⑪$$

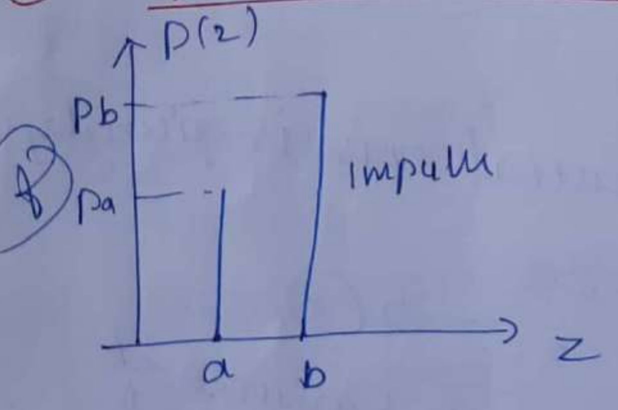
* The mean of this density fun is given by

$$\bar{z} = \frac{a+b}{2} \rightarrow ⑫$$

& its variance by

$$\sigma^2 = \frac{(b-a)^2}{12} \rightarrow ⑬$$

⑥ impulse (salt & pepper) noise



* The PDF of (bipolar) impulse noise is given by

$$p(z) = \begin{cases} P_a & , \text{ for } z = a \\ P_b & ; \text{ for } z = b \\ 0 & ; \text{ otherwise} \end{cases} \rightarrow ⑭$$

- If $b > a$, intensity b will appears as a light dot in the image
- Conversely, if level a will appear like a dark dot
- If either P_a or P_b is zero, the impulse noise is called unipolar
- If neither probability is zero, & if they are approximately equal, impulse noise values will resemble salt & pepper granules randomly distributed over the image

* for this reason, bipolar impulse noise is also called as salt & pepper noise

* Generally a & b values are saturated (very high or very low value), resulting in +ve impulses being white (salt) & negative impulses being black (pepper)

* If Pa=0 & only Pb exists ie, called pepper noise as only black dots are visible as noise

* If Pb=0 & only Pa exists, this is called as 'salt noise' as only white dots are visible on the image as noise

* Impulse noise occurs when quick transitions happen, such as faulty switching takes place

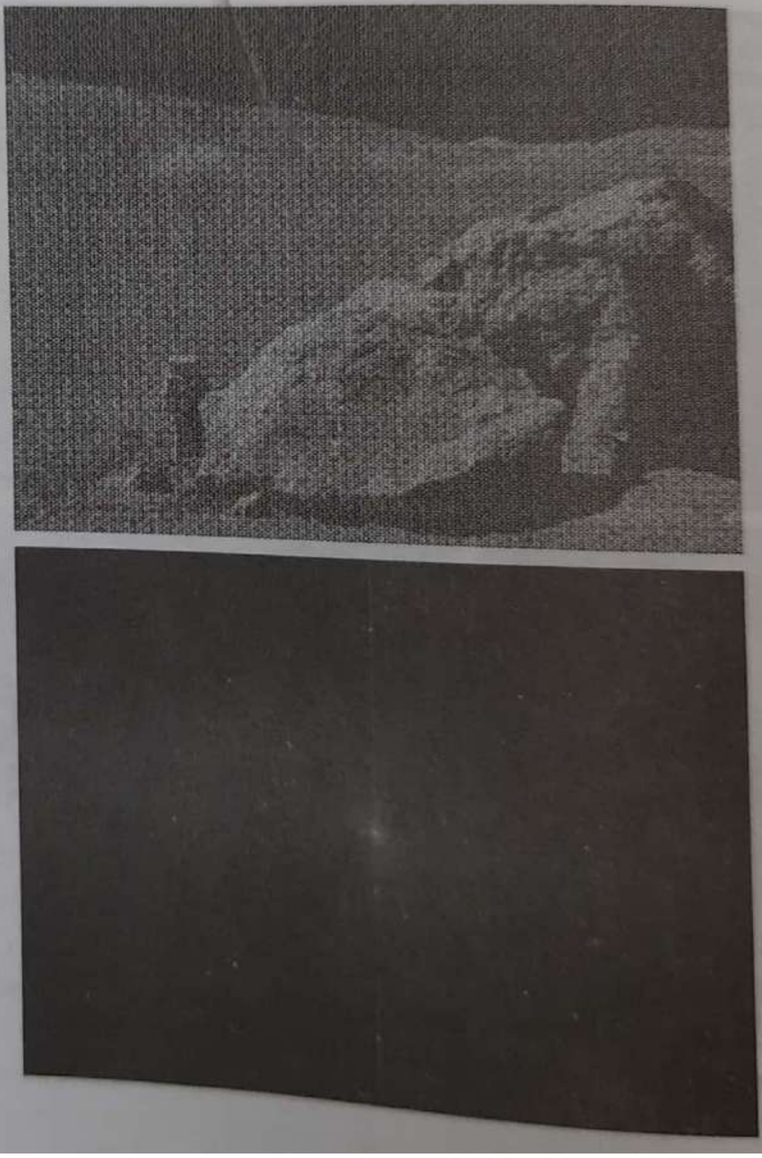
* Noise parameters are generally estimated based on histogram of small flat area of noisy image

* each pixel in an image has a probability of P/2 (0<p<1) being contaminated by either white dot (salt) or a black dot o (pepper)

fig 5.4

a
b

FIGURE 5.5
(a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)



obscuring the

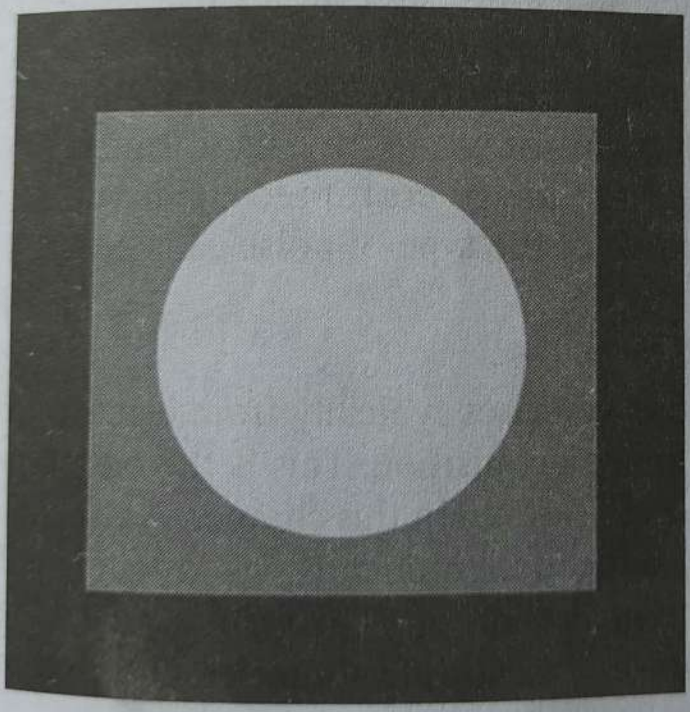
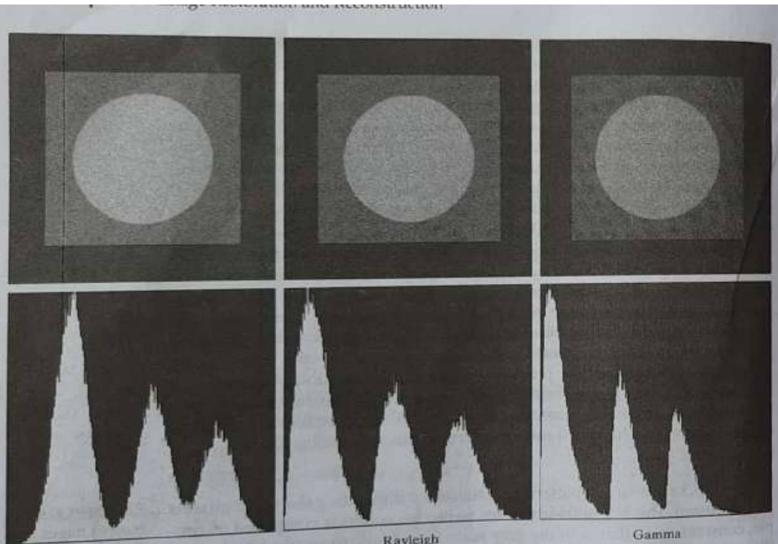


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

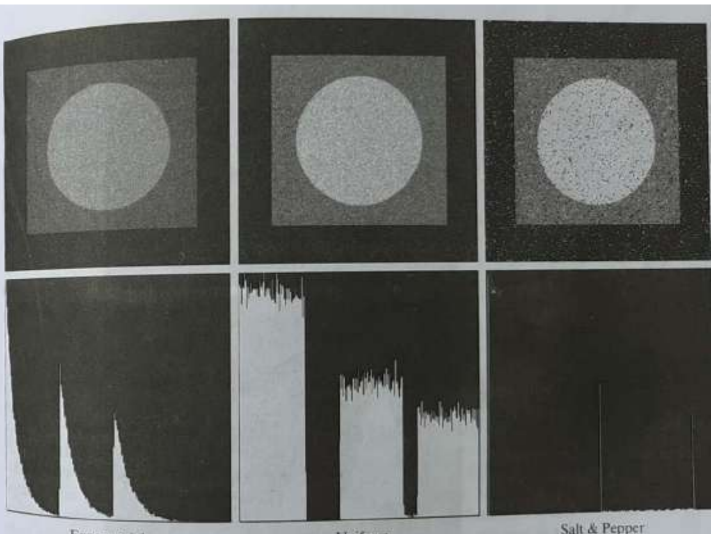

Gaussian

Rayleigh

Gamma

a b c
d e f

FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.


Exponential

Uniform

Salt & Pepper

g h i
j k l

FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt-an pepper noise to the image in Fig. 5.3.

5.2.3 Periodic Noise

- Periodic noise in an image arises typically from electrical or electromechanical interference during image acquisition
- This is the only type of spatially dependent noise
- Periodic noise can be reduced significantly via freq domain filtering
- A strong periodic noise can be seen in frequency domain as equi spaced dots at a particular radius around the centre (origin) of the spectrum
- For eg If the image is severely corrupted by (spatial) sinusoidal ~~image~~ noise of various frequencies
- The FT of a pure sinusoid is a pair of conjugate impulses located at the conjugate freq's of the sine wave
- Thus If the amplitude of a sine wave in the spatial domain is strong, then we would expect to see a pair of impulses in the spectrum. for each sine wave

5.2.4 Estimation of Noise parameter

* The parameters of periodic noise are estimated by inspection of the Fourier spectrum of the image.

* periodic noise tends to produce freq spikes that often can be detected by visual analysis.

* Another approach is to attempt to infer the periodicity of noise components directly from the image, this is possible for simplistic cases.

* Automated analysis is possible in situations in which the noise spikes are either ~~exp~~ exceptionally pronounced or when knowledge is available about the general location of the freq components of the interference

* The parameters of noise PDF's may be known partially from sensor specification but it is often required to estimate them ~~for~~ for a particular imaging arrangement

* If the imaging system is available, then one simple way to study the characteristics of system noise is to capture, a set of images of "flat" environment and estimate the parameters of the PDF from small patches of reasonably constant background intensity.

⑤

* The simplest ~~use~~ use of the data from the image strips is for calculating the mean & variance of intensity Levels.

* Consider a strip (subimage) denoted by 'S' &

Let $P_S(z_i)$; where $i = 0, 1, 2 \ldots L-1$, denote the Probability estimates (normalized histogram values) of the intensities of the pixelins. where L = no of possible intensities in the entire image.

* The mean & variance of the pixels 'S' can be calculated as

$$\bar{z} = \sum_{i=0}^{L-1} z_i P_S(z_i) \longrightarrow ⑮$$

&

$$\sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 P_S(z_i) \longrightarrow ⑯$$

* The shape of the histogram identifies the ~~closet~~ closest PDF match

* If the shape is approximately Gaussian, then mean & variance are need.

* for other shapes, mean & variance are used to solve for the parameters a & b

* for impulse noise, the heights of the peaks corresponding to black & white pixels are the estimates of Pa & Pb.

53

Restoration in the presence of Noise only - spatial filtering

- When only degradation present in image is noise then

$$g(x,y) = f(x,y) + \eta(x,y) \longrightarrow ①$$

$$\& \quad G(u,v) = F(u,v) + N(u,v) \longrightarrow ②$$

- noise term is unknown so subtracting them from g(x,y) or G(u,v) i.e,

$$f(x,y) = g(x,y) - \eta(x,y)$$ is not a realistic option.

- Thus spatial filtering is used when additive random noise is present

Noise removal / restoration method

Mean filters
- Arithmetic meanfilter
- Geometric
- Harmonic
- Contra-harmonic

Other statistics filter
- Median filter
- Max & min,
- Midpoint.
- Alpha trimmed

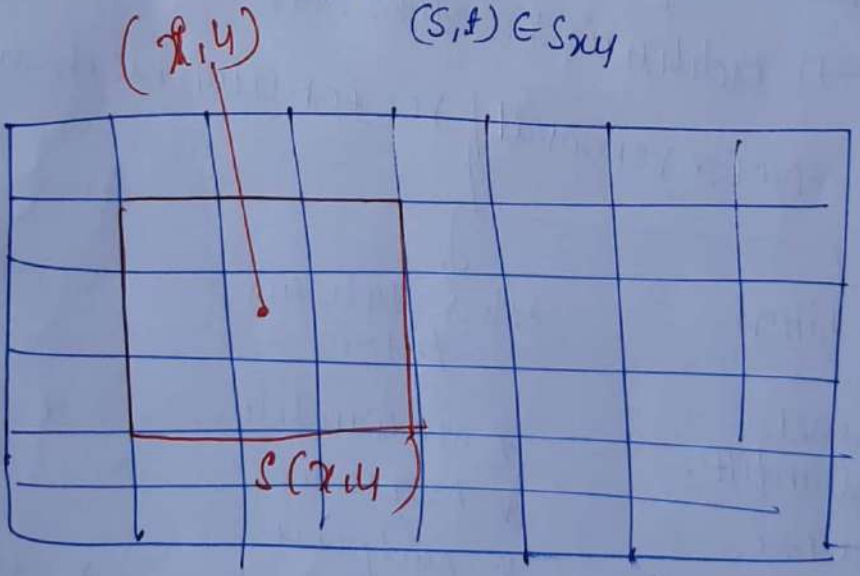
Adaptive filter
- Adaptive local noise reduction
- Adaptive median filter

$$\boxed{\text{Noisy image } g} \longrightarrow \boxed{\text{Filtering Alg}} \longrightarrow \text{Denoised image } \hat{f}$$

mean filters

(i) Arithmetic mean filter

- simplest form of mean filter
- S_{xy} => set of coordinates in a rectangular sub image window (neighborhood) of size m×n, centred at a point x,y.
- mean filter computes avg value of the corrupted image g(x,y) in the area defined by $S_{x,y}$
- $$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t) \longrightarrow ①$$



- such a filter smooths local variations in an image thus reducing noise & introducing blurring.
- This filter is well suited for random noise like Gaussian, uniform noise

Thus new value at (x,y) in image 6.12 = mean {g(s,t)} = $\frac{1}{9}$ [30 + 10 + 20 + 10 + 250 + 25 + 20 + 25 + 3] = 46.7 ≈ 47

30	10	20
10	250	25
20	25	30

×	×	×
×	46.7 ≈ 47	×
×	×	×

FIGURE 6.12: Example of mean filtering

Example 6.2

Show effect of 3 × 3 mean filter on a simple image in fig 6.13 (a) and (c)

Solution:

0	0	0	0	0
0	0	0	1	1
0	0	1	1	1
0	0	1	20	1
0	0	1	1	1

(a)

Mean filter

	$^1/_9$	$^3/_9$	$^5/_9$	
	$^2/_9$	$^{24}/_9$	$^{27}/_9$	
	$^3/_9$	$^{25}/_9$	$^{28}/_9$	

(b)

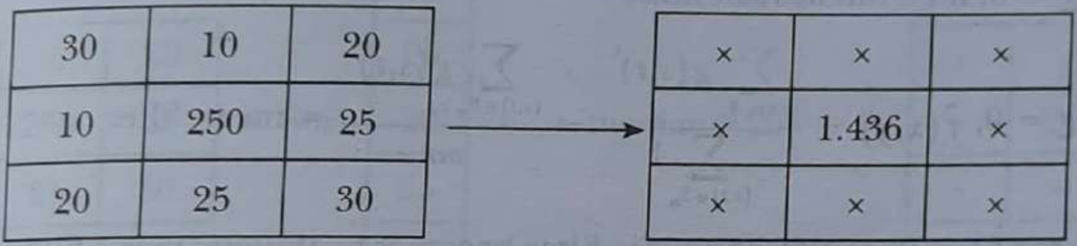
b. Geometric Mean Filter

Restored image by a geometric mean filter is given by

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t)\right]^{1/mn} \tag{6.20}$$

Thus new value at (x,y) in image 6.15 $= \underset{s,t\in Sxy}{\text{Geometric mean}}[g(s,t)]$

$$= [30 \times 10 \times 20 \times 10 \times 250 \times 25 \times 20 \times 25 \times 30]^{1/81} = 1.436$$



30	10	20
10	250	25
20	25	30

×	×	×
×	1.436	×
×	×	×

FIGURE 6.15: Example of geometric mean filter

Geometric mean filter achieves less smoothing as compared to the arithmetic mean filters but it preserves more details.

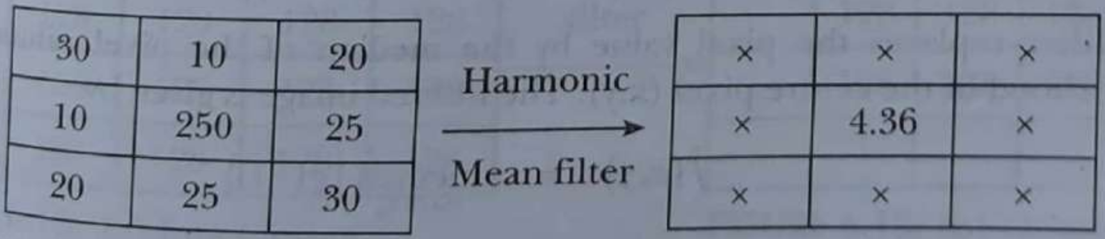
c. Harmonic Mean Filter

Harmonic mean filtered image is given by,

$$\hat{f}(x,y) = \frac{mn}{\sum\limits_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}} \tag{6.21}$$

Thus new value at (x,y) in image 6.16 $= \underset{s,t\in S^{xy}}{\text{Harmonic mean}}[g(s,t)]$

$$= \frac{9}{\frac{1}{30}+\frac{1}{10}+\frac{1}{20}+\frac{1}{10}+\frac{1}{250}+\frac{1}{25}+\frac{1}{20}+\frac{1}{25}+\frac{1}{30}}$$



30	10	20
10	250	25
20	25	30

×	×	×
×	4.36	×
×	×	×

FIGURE 6.16: Example of Harmonic mean filter

Harmonic mean works well for salt noise and gaussian noise, but fails for pepper noise.

d. Contra Harmonic Mean Filter

Restored image from contra harmonic filter is

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}} \tag{6.22}$$

Here, Q is the order of the filter. This filter reduces salt & pepper (impulse) noise.

For $Q > 0$, it eliminates pepper noise.

For $Q < 0$, it eliminates salt noise.

For $Q = 0$, $$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{1}}{\sum_{(s,t)\in S_{xy}} 1} = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{1}}{mn} = \text{mean filter}$$

Thus for $Q = 0$, contra-harmonic filter becomes arithmetic mean filter.

For $Q = -1$, $$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{0}}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}} = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

$$= \text{Harmonic mean filter}$$

Thus, for $Q = -1$, it becomes harmonic mean filter. Q has to be chosen properly. Wrong Q gives disastrous results.

Harmonic mean works well for salt noise and gaussian noise, but fails for pepper noise.

d. Contra Harmonic Mean Filter

Restored image from contra harmonic filter is

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}} \tag{6.22}$$

Here, Q is the order of the filter. This filter reduces salt & pepper (impulse) noise.

For Q > 0, it eliminates pepper noise.

For Q < 0, it eliminates salt noise.

For Q = 0, $\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{1}}{\sum_{(s,t)\in S_{xy}} 1} = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{1}}{mn}$ = mean filter

Thus for Q = 0, contra-harmonic filter becomes arithmetic mean filter.

For Q = -1, $\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{0}}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}} = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$

= Harmonic mean filter

Thus, for Q = -1, it becomes harmonic mean filter. Q has to be chosen properly. Wrong Q gives disastrous results.

6.5.2 Order Statistics Filter

Order statistics filter are **non-linear** spatial filters. Its response is based on ordering the pixels contained in sub – image area. Filter is implemented by replacing the centre pixel value with the value determined by the ranking result. As shown in table 6.2, four types of order statistics filters are discussed here.

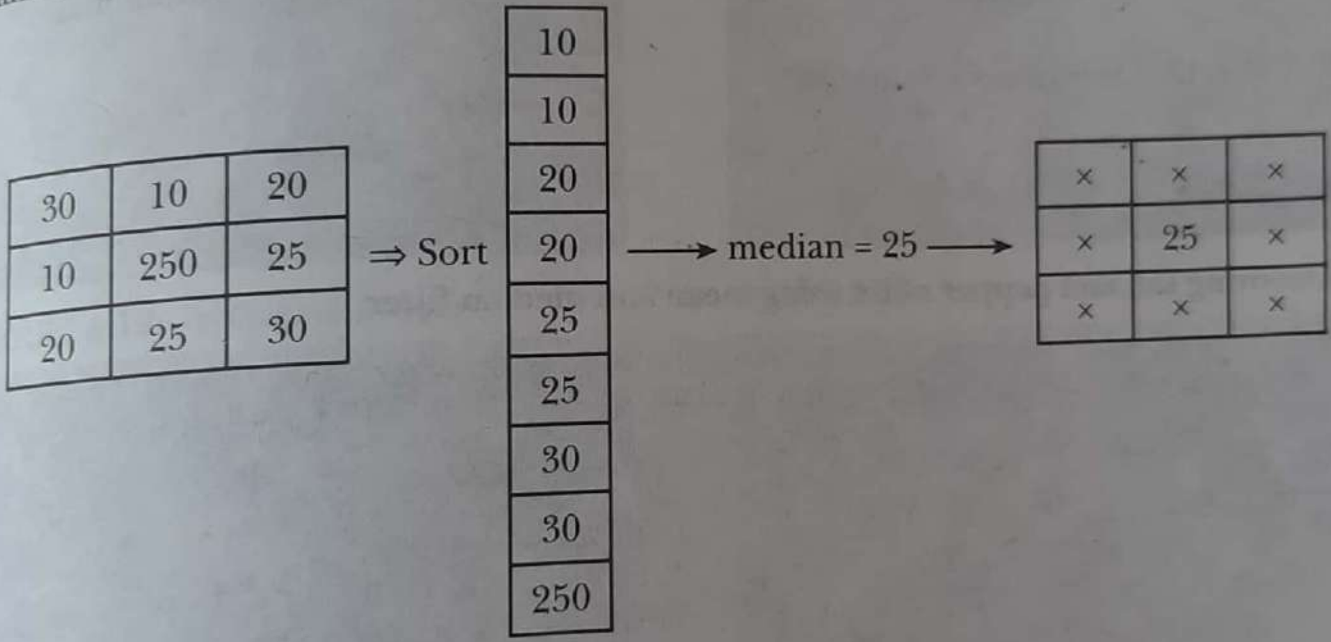
a. Median Filter

Median filter replaces the pixel value by the median of the pixel values in the neighbourhood of the centre pixel (x,y). The filtered image is given by

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}} \{g(s,t)\} \tag{6.23}$$

Fig 6.17 shows the procedure of applying 3 × 3 median filter on an image. As impulse noise appears as black (minimum) or white (maximum) dots, taking median effectively suppresses the noise. It is clear from example 6.3, fig 6.18 (a,b)that if noise strength is low in noisy image, output is completely clean. But if noise strength is more (more number of noisy pixels in the image), output is not completely noise free as can be seen in fig 6.18 (c,d)

Thus, median filter provides excellent results for salt and pepper noise with considerably less blurring than linear smoothing filter of the same size. These filters are very effective for both bipolar and unipolar noise. But, for higher noise strength, it affects clean pixels as well and a noticeable edge blurring exists after median filtering.



30	10	20
10	250	25
20	25	30

10
10
20
20
25
25
30
30
250

×	×	×
×	25	×
×	×	×

FIGURE 6.17: Example of median filtering

Example 6.3

Example 6.3 show the effect of 3 × 3 median filter on a simple image in fig 6.18 (a and c).

Solution

128	128	128	128	128
128	0	128	128	128
128	128	128	128	128
128	128	128	128	128
128	128	128	128	128

FIGURE 6.18: (a) Input image

Median filter →

	128	128	128	
	128	128	128	
	128	128	128	

FIGURE 6.18: (b) Output image

128	128	128	0	128
128	0	128	128	128
0	0	255	255	255
0	0	128	255	0
128	0	0	0	128

FIGURE 6.18: (c) Input image

Median filter →

	128	128	255	
	0	128	255	
	0	0	128	

FIGURE 6.18: (d) Outputimage

FIGURE 6.18: Example of median filter

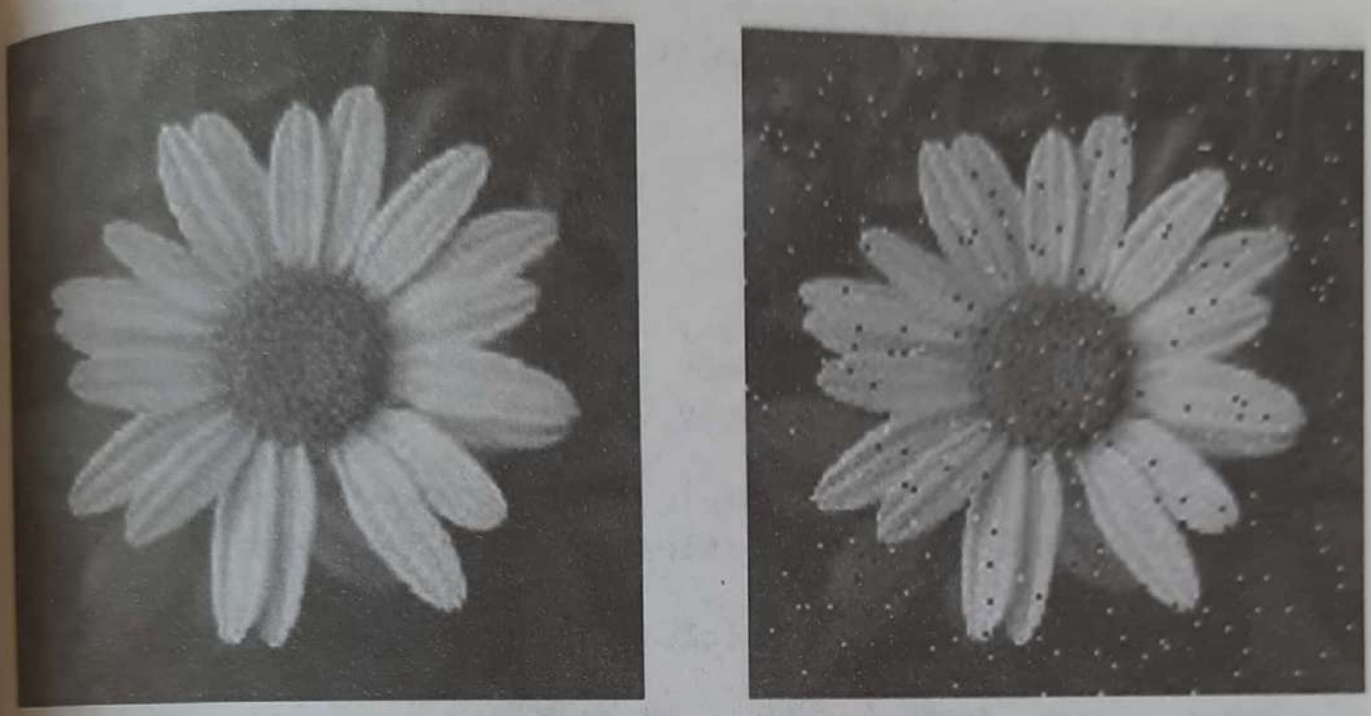


FIGURE 6.19: (a) Original image

FIGURE 6.19: (b) Noisy image

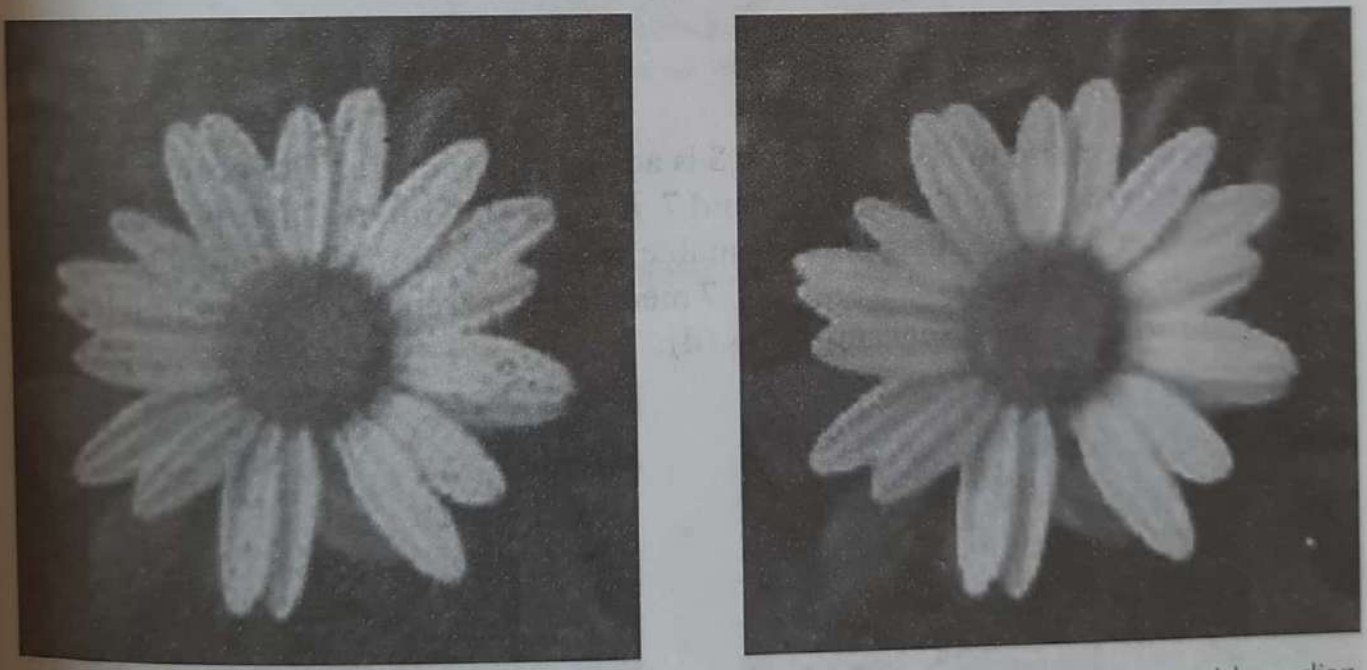


FIGURE 6.19: (c) Filtered image with mean filter

FIGURE 6.19: (d) Filtered image with median filter

FIGURE 6.19: (a) Input image (b) noisy image, image filtered by (c) mean (d) Median filter

Matlab Ex 6.4

Explanation

Salt and pepper noise with density of 0.3 is added to an image. The noisy image (fig 6.20 (a)) is filtered using 3 × 3, 5 × 5 and 7 × 7, median filter. The results in fig 6.20 b,c,d show that 3 × 3 median filter is unable to remove the noise completely as the noise density is high. But 5 × 5 and 7 × 7 median filters remove noise completely but some distortions are seen specially in fig (d).

FIGURE 6.20: (a) Noisy image

FIGURE 6.20: (b) Filtered image with 3 × 3 median filter

FIGURE 6.20: (c) Filtered image with 5 × 5 median filter

FIGURE 6.20: (d) Filtered image with 7 × 7 median filter

FIGURE 6.20: (a) Noisy image, image filtered by median filter of size (b) 3 × 3 (c) 5 × 5 (d) 7 × 7

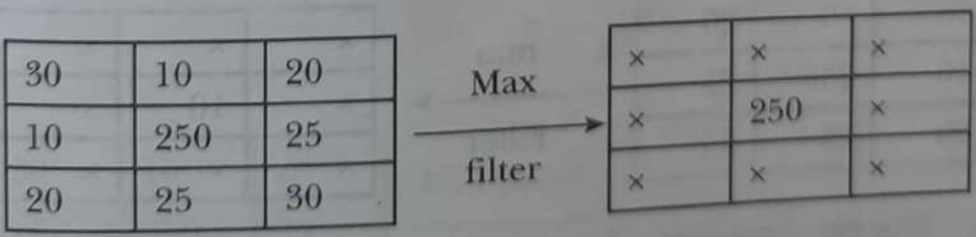
b. Max and Min Filter

The restored image from a max filter is given by

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\} \tag{6.23}$$

Thus new value at (x,y) in fig 6.21 $= \max_{s,t\in S_{xy}} \{g(s,t)\}$ $= \max \{ 30, 10, 20, 10, 250,25, 20,25, 30$

$= 250$



30	10	20
10	250	25
20	25	30

×	×	×
×	250	×
×	×	×

FIGURE 6.21: Example of max filter

Example 6.3

Show the effect of 3 × 3 max on image in fig 6.22 (a)

Solution

128	128	128	128	128
128	0	128	0	128
128	128	0	128	128
128	0	255	0	128
128	128	128	128	128

(a) Input image

$\xrightarrow[\text{Filter}]{\text{Max}}$

	128	128	128	
	255	255	255	
	255	255	255	

(b) Output image

FIGURE 6.22: Example of max filter

This filter is useful in finding the brightest points in an image, therefore it is effective against pepper noise. Problem occurs when both salt & pepper noise is there and there are more noisy pixels. In this case, even non- noisy pixel values are also replaced by salt noise values. As it is clear from example 6.3, 128 pixel value is non noisy.

0 → pixel affected by pepper noise, 255 → pixel affected by salt noise

After the application of filter in fig 6.22 (b), only the first row values are non-noisy, other rows have noise values (255).

Image restored from a **min filter** is given by

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\} \qquad (6.24)$$

Thus new value at (x,y) in fig 6.23 $= \min_{s,t\in S_{xy}} \{g(s,t)\}$ $= \min \{ 30, 10, 20, 10, 250, 25, 20, 25, 30$

$= 10$

30	10	20
10	250	25
20	25	30

$\xrightarrow[\text{Filter}]{\text{min}}$

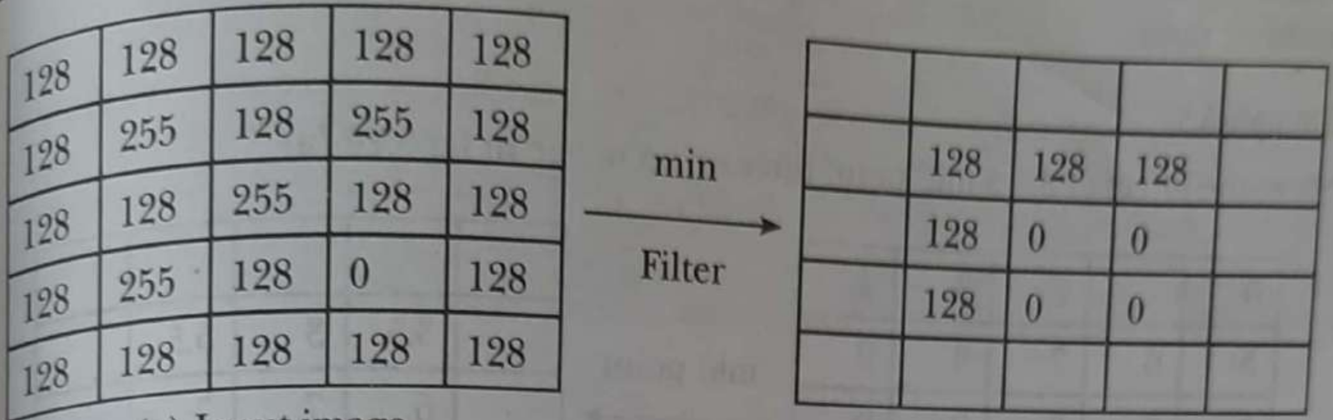
×	×	×
×	10	×
×	×	×

FIGURE 6.23: Example of min filter

Example 6.4

Show the effect of 3 × 3 min filter on image in fig 6.24 (a).

Solution



128	128	128	128	128
128	255	128	255	128
128	128	255	128	128
128	255	128	0	128
128	128	128	128	128

(a) Input image

	128	128	128	
	128	0	0	
	128	0	0	

(b) Output image

FIGURE 6.24: Example of min filter

In the above example 6.4, 128 pixel is non noisy value

255 → pixel affected by salt noise, 0 → pixel affected by pepper noise

In the output Fig 6.24 (b) first row has non noisy pixel values, where as 2nd and 3rd row has pepper noise values a output.

This filter is useful in finding darkest points in an image, it is effective against only salt noise. The problem occurs when both salt and pepper noise is present in an the image, even non-noisy pixel values are replaced by pepper noise.

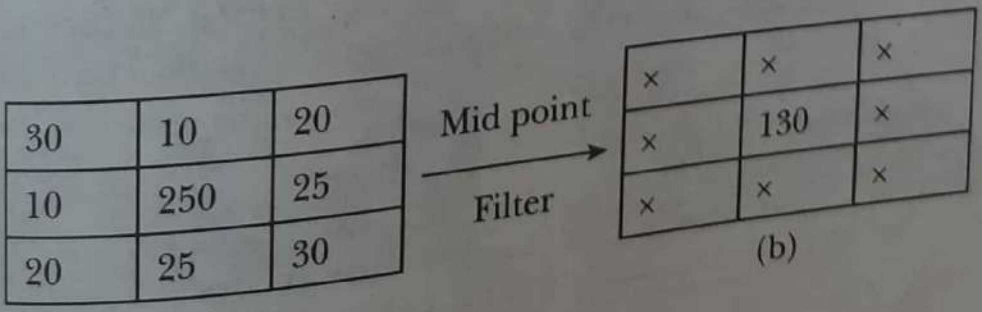
c. Midpoint Filter

This filter computes the mid point of maximum and minimum values of intensities.

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right] \tag{6.25}$$

The new value at (x,y) in image in fig 6.25 $= \frac{1}{2}\left[\max\{g(s,t)\} + \min\{g(s,t)\}\right]$

$$= \frac{1}{2}[250 + 10] = 130$$



30	10	20
10	250	25
20	25	30

(a)

×	×	×
×	130	×
×	×	×

(b)

FIGURE 6.25: Example of mid point filter

This filter is a combination of order statistics and averaging. It works well for Gaussian uniform noise.

Example 6.5

Show the effect of 3 × 3 mid point filter on an image in fig 6.26 (a)

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

(a) Input image

$\xrightarrow[\text{filter}]{\text{mid point}}$

	3.5	3	5.5	
	6	7	7	
	5	7	8	

(b) Output image

FIGURE 6.26: Example of mid point filter

Explanation

Salt and pepper noise is added to an input image shown in fig 6.27 (a). Median filter is implemented by ordfilt2 command by choosing 5 (center value in 3 × 3 = 9 pixels). Max filter is implemented by choosing 9th (highest value in 9 pixel) and min filter is implemented by choosing 1(minimum value in 9 pixels). Mid point filter is implemented by taking average of min and max filter values. As it is clear from the output (fig 6.27 (c)) that, median filter completely removes salt and pepper noise. But max filter fig (d) removes only pepper noise (black dots) but salt noise remains and same distortions in terms of salt noise is added in the output (fig d). Similarly, min filter removes only salt noise (white dots) completely but pepper noise remains and same distortions in terms of pepper noise is added in the output (fig e). In case of mid point filter, noise values and other pixel values are also replaced by average value(125). Therefore lot of grey pixels are seen in the image (fig f).

FIGURE 6.27: (a) Original image

FIGURE 6.27: (b) Noisy imgae

FIGURE 6.27: (c) Filtered image using median filter

FIGURE 6.27: (d) Filtered image using max filter

FIGURE 6.27: (e) Filtered image using min filter

FIGURE 6.27: (f) Filtered image using mid point filter

FIGURE 6.27: Original image (b) noisy image, filtered image using (c) median (d) max (e) min (f) mid point filter

d. Alpha-trimmed Mean Filter

Let there be m × n pixels in neighbourhood S_{xy}. Remove $d/2$ lowest and $d/2$ highest grey level valued pixels. Number of remaining pixels are (mn – d) which are represented by $g_r(s,t)$. Restored image by alpha – trimmed mean filter is given by

$$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(s,t)\in S_{xy}} g_r(s,t) \tag{6.26}$$

Here d can range from 0 to mn – 1.

For d = 0, alpha trimmed filter = Arithmetic filter

For d = $\frac{mn-1}{2}$ alpha trimmed filter = median filter

30	10	20
10	250	25
20	25	30

(a) Input image

$\xrightarrow{d = 2}$

alpha-trimmed mean filter

×	×	×
×	23	×
×	×	×

(b) Output image

FIGURE 6.28: Example of alpha-trimmed filter with d = 2

Let d = 2, we remove $\frac{d}{2}$ = 1 min value (10 in this case) and $\frac{d}{2}$ = 1 max value (250 in this case) and then the value at (x,y) in image in fig 6.28 (a) = $\frac{1}{(9-2)}$ [30 + 10 + 20 + 25 + 20 + 25 + 30] = 22.85 ≈ 23

For d = 4, remove 2 min (10, 10 in this case) and 2 max (250, 30 in this case) values and the new value at (x,y) in image fig 6.28 (c) = $\frac{1}{(9-4)}$[30 + 20 + 25 + 20 + 25] = 24

30	10	20
10	250	25
20	25	30

(a) Input image

$\xrightarrow{d = 4}$

alpha-trimmed mean filter

×	×	×
×	24	×
×	×	×

(b) Output image

FIGURE 6.28: Example of alpha-trimmed filter with d = 4

This filter removes a combination of salt & pepper and Gaussian noise.

Adaptive filters

- Mean filters & order stastics filters are not capable of distinguishing noise from pixel values.

~~* Mead~~

- These filters replace all pixel values with mean/median which causes distortions
- Adaptive filters are capable of superior performance because its behaviours adapts to the change in characteristics of image area being filtered.
- This ries the complexity of the filter

(a) Adaptive Local Noise Reduction Filter

- This filter changes its action based on statistical properties of the pixels in region Sxy.
- The simplest statistical measure of a random variables are its mean & variance.
- These are the quantities closely related to appearance of an image
- mean gives a measure of avg intensity in the region over which mean is computed
- & * Variance gives a measure of contrast in that region
- These 2 parameters are chosen to change the behavior of adaptive local noise reduction filter

- filter is operated on a local region S_{xy}.

- The response of the filter at any point (x,y) on which the region is centered is to be based on 4 quantities

(i) $g(x,y) \rightarrow$ value of the noisy image at (x,y)

(ii) $\sigma_n^2 \Rightarrow$ variance of noise corrupting $f(x,y)$ to form $g(x,y)$

(iii) m_L = Local mean of the pixels in S_{xy}.

(iv) σ_L^2 : Local variance of the pixels in S_{xy}.

- Behaviour of noise reducing filter should be as follows

(1) If $\sigma_n^2 = 0$; the filter should return simply the value of $g(x,y)$. [in case of no noise]

$\therefore \; g(x,y) = \hat{f}(x,y)$.

(2) - If the local variance is high relative to σ_n^2, the filter should return a value close to $g(x,y)$.

- A high variance typically is associated with edges & these should be preserved.

(3) • If the 2 variances are equal, we want the fiter to return the arithmetic mean value of the pixels in Sxy.

* This condition occurs when the local area has the same properties as the overall image & the local noise is to be reduced by averaging.

* Adaptive filter is given by

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_n^2}{\sigma_L^2}\left[g(x,y) - m_L\right] \rightarrow ①$$

σ_n^2 = is the only quantity that needs to be known or estimated is the variance of the overall noise σ_n^2.

* The other parameters are computed from the pixels in Sxy, at each location (x,y) on which the filter window is centered.

* σ_L^2 & M_L is estimated for the selected area

(1) In case of no noise $\sigma_n^2 = 0$, then eq ① becomes

$$\hat{f}(x,y) = g(x,y)$$

2

(2) In case of edges $\sigma_n^2 < \sigma_L^2$

Then $\frac{\sigma_n^2}{\sigma_L^2} \approx 0$

substuting this in eq ①

$$\hat{f}(x,y) = g(s,t) - 0[g(s,t) - m_L]$$

$$\approx g(s,t)$$

(3) In case of presence of noise

If $\sigma_n^2 = \sigma_L^2$ then $\frac{\sigma_n^2}{\sigma_L^2} = 1$

Then eq ①

$$\hat{f}(x,y) = g(s,t) - [g(s,t) - m_L]$$

$$\hat{f}(x,y) = m_L$$

- Adaptive filter achieves approximately the same performance in noise reduction as the mean filter, but introduces less blurring than the mean filter.
- Thus adpative filter yields considerably better results in overall performance at the price of filter complexity
- If the noise variance is not estimated correctly, filter gives undesirable results
- If estimated variance value is too low as compared to actual variance, noise correction will be smaller than it should be

& If the estimate is too high, the noise correction is large & o/p image loose dynamic range

(b) Adaptive median filter /.

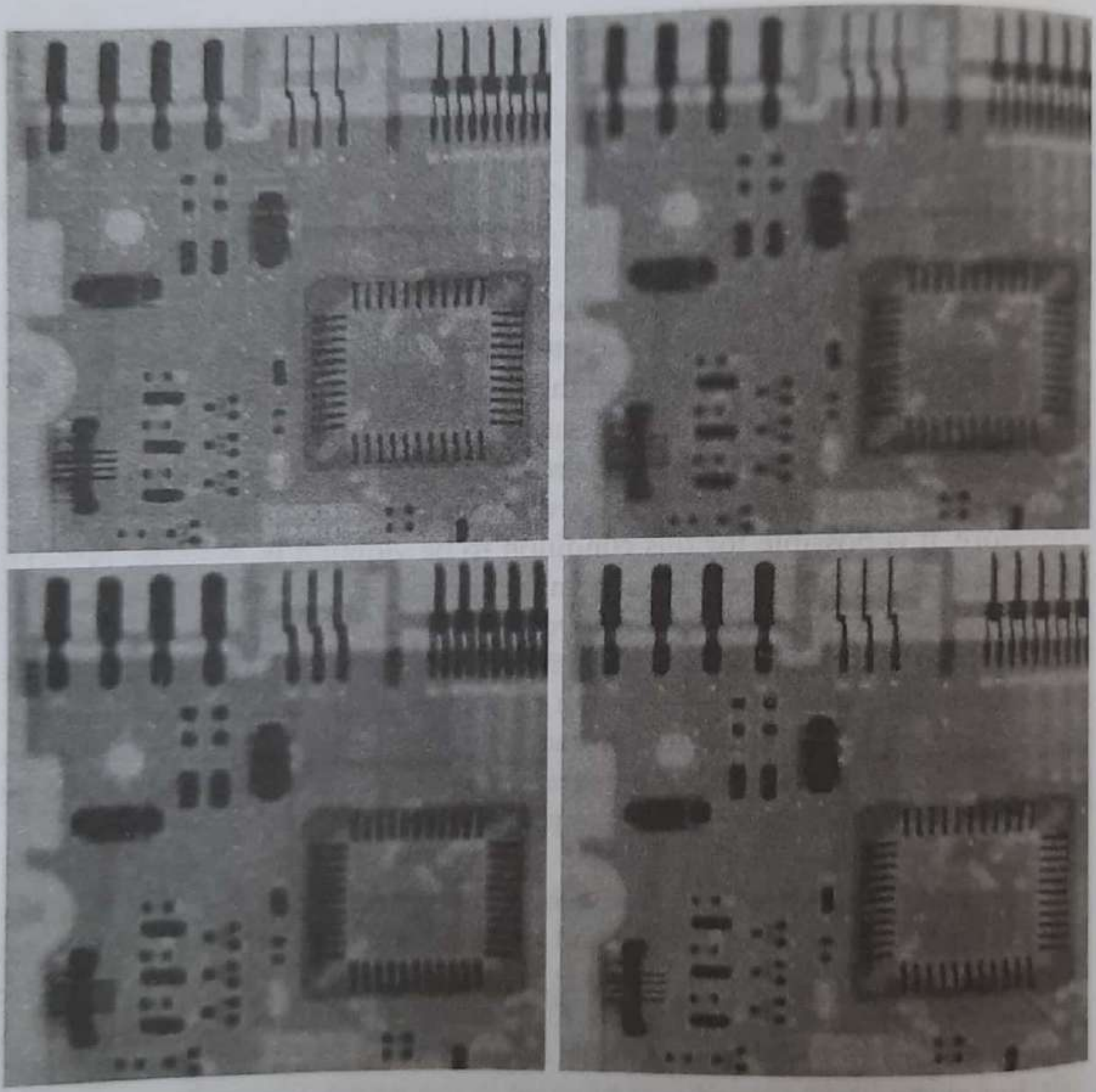
- median filter performs well if the spatial density of the impulse noise is not large ie, impulse noise with smaller probability (P_a & P_b < 0.2).

- Adaptive median filtering can handle impulse noise with probabilities larger than these

- Additional benefit of the adaptive mean filter is it seeks to preserve detail while smoothing nonimpulse noise.

- main objective of the adaptive median filter is
  - To remove salt & peper (impulse) noise
  - To smoothen noise ~~oth~~ other than impulse noise
  - To reduce distortion of thinning & thickening of edges.

- adaptive median filter works in a rectangular window area S_{xy} ~~unlike as~~ like other filters



a b
c d

FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.

* unlike other filters, the adaptive ~~mean~~ median filter changes (increases) the size of S_{xy} during filter operation depending on certain conditions.

* o/p of the filter is a single value used to replace the value of the pixel at (x,y), the point on which the window $S_{x,y}$ is centered at a given time

* variables used in this algorithm are

S_{xy} = rectangular window whose size vary during operation of adaptive filter centered at (x,y)

Z_{min} = min grey level value in S_{xy}

Z_{max} = max grey level value in S_{xy}

Z_{med} = median of grey values in $S_{x,y}$

Z_{xy} = grey level at (x,y)

S_{max} = Max allowed size S_{xy}.

* In the algorithm, Z_{min} & Z_{max} are considered to be "impulse like" noise

Algorithm of Adaptive median filter

Stage A :-

If $A_1 = Z_{med} - Z_{min}$ } (or) If ~~$Z_{min}<$~~ $Z_{min} < Z_{med} < Z_{max}$
$A_2 = Z_{med} - Z_{max}$ }

If $A_1 > 0$ AND $A2 < 0$ go to stage B
else increase the window size
If window size $\leq S_{max}$ repeat stage A
else output $= Z_{med}$.

Stage B :- $B_1 = Z_{xy} - Z_{min}$
$B_2 = Z_{xy} - Z_{max}$
if $B1 > 0$ AND $B2 < 0$, o/p Z_{xy} [do not filter]
else output Z_{med}

* ~~Explanati~~on
* To understand, the mechanism of this algorithm, the kep is to keep in mind that it has 3 main purpose.

▣ to remove salt & peper (impulse) noise
▣ to provide smoothing of other noise that may not be impulsive
▣ to reduce distortion such as ~~considering~~ excessive thinning or thickening of object boundaries

* The values Zmin & Zmax are considered to be impulse-like noise component [Zmin = pepper noise Zmax : salt noise]

* Zxy = pixel value which is to be filtered.

* If Zxy is either salt noise or pepper noise, it should be replaced by median value

* In the region Sxy, centered at (x,y) find the median value Zmed.

* Stage A checks is Zmed is impulse or not.

* Stage A :- If Zmed ≠ impulse, then go to stage B. In stage B, we check if Zxy is impulse or not

* Stage B :- If Zxy ≠ impulse, then there is no need to filter & o/p value is same as Zxy

  If Zxy = impulse (Zxy = Zmin ||
  Zxy = Zmax),
  then o/p = median value (which is not noisy checked at stage A).

* thus here we are ensuring 2 things.

(1) In case of non noisy pixel ⇒ no filter action should take place, o/p = Zxy

(2) In case Zxy is noisy, then it should be replaced by a non-noisy median value (If it is noisty stage A takes care).

* In case, the 1st statement in stage A fails, then Zmed is either salt noise or pepper noise then in this case Zmed can't be used to replace a noisy pixel Zxy at ~~test~~. stage B.

* In stage B we ensure that median is never a noisy value

* To do this size of window is ↑sed & Zmed is tested again for Zmin < Zmed < Zmax

If the condn is true, we go to stage B else again size of window 's' is ↑sed till it reaches Smax.

* If max limits of window is reached & still Zmed is noisy then o/p = Zxy we don't filter Zxy & o/p is not Zmed which is also noisy

* every time o/p is generated,
window shifts & algorithim is reinitialized

* Advantage of this filter

a. only a noisy pixel is filtered
b. if filtering is done, we make sure that the median values is not noise

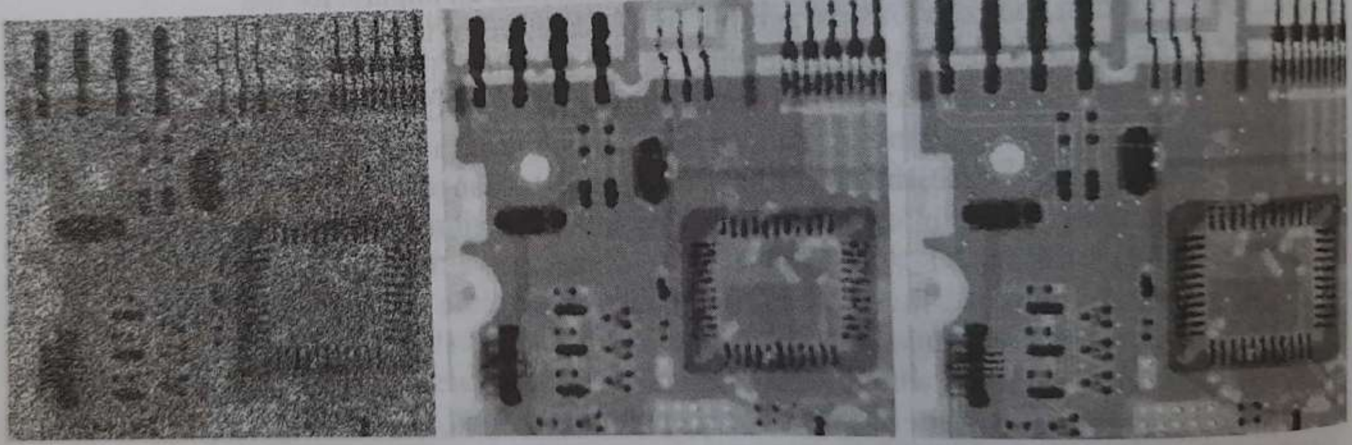


FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{\max} = 7$.

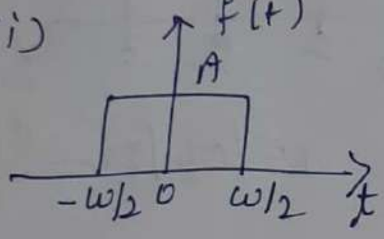
Module-3

Assignment-2

1. Explain the following preliminary concepts
   - (i) complex Numbers
   - (ii) Fourier series
   - (iii) impulses & their shifting property
   - (iv) convolution.

(2) Find the Fourier Transforms of the foll

(i)



draw its FT & spectrum

(ii) unit impulse $\delta(t)$ & (iii) shifted impulse $\delta(t-t_0)$

(iv) train of impulse

$$S_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T)$$

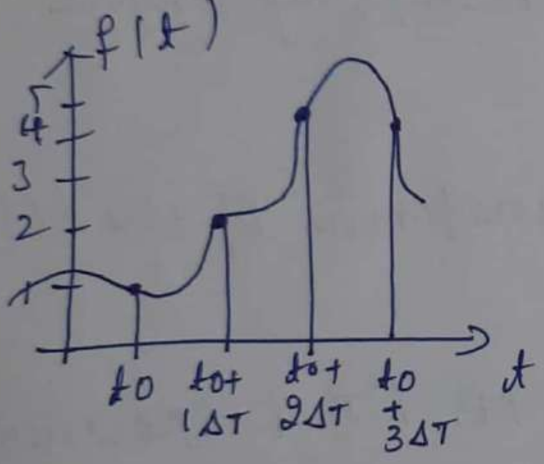
(3) Explain the process of sampling & ~~(4) write the~~ ~~itto~~ delive the FT of sampled fun

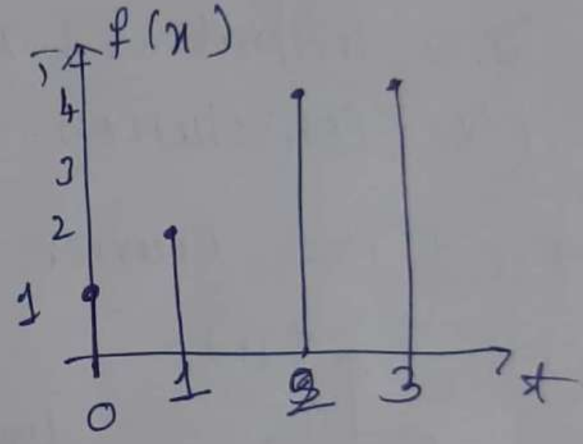
(4) State & explain sampling theorem & highlight on Aliasing & Reconstruction function from sampled data

(5) obtain the DFT from continous Transform of sampled function & write the relationship bet' sampling & freq intervals

⑧ fig ⓐ shows 4 samples of continuous fun f(t) taken ΔT units apart. fig ⓑ shows the sampled values in the x-domain. Note the values of x are 0, 1, 2 & 3. indicating as 4 samples of f(t)

find F(u) & f(x).





ASSIGNMENT DEADLINE! . 15/04/2020 [wednesday]

①

Morphological Image Processing

Morphology — branch of biology that deals with the form & structure of animals & plants.

Mathematical morphology ⟶ tool for extracting image components that are useful in the representation & description of region. Shape, such as boundaries, skeletons & the convex hull.

Language of Mathematical morphology ⟶ Set theory.

Preliminaries:-

Some basic concepts from set theory

Let A be a set in Z^2. If $a = (a1, a2)$ is an element of A, $\boxed{a \in A}$ otherwise $\boxed{a \notin A.}$

The set with no elements — null or empty set ϕ.

$C = \{w / w = -d \text{ for } d \in D\}$ means C is a set of elements w, & w is formed by multiplying each of the 2 coordinates of all the elements of set D by -1

$A \subseteq B \rightarrow$ A is subset of B.

$C = A \cup B \rightarrow$ union of A & B.

$D = A \cap B \rightarrow$ intersection.

$A \cap B = \phi \longrightarrow$ disjoint / mutually exclusive. no common elements.

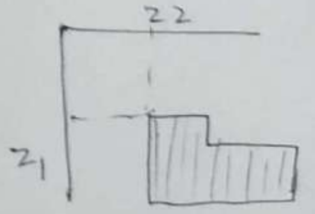
$A^c = \{w / w \notin A\}$ complement of A.

$A - B = \{w / w \in A, w \notin B\} = A \cap B^c$

Reflection of set B → $\hat{B} = \{w / w = -b, \text{ for } b \in B\}$

Translation of set A by point $z = \{z_1, z_2\}$ denoted by $(A)_z$ is defined as

$$(A)_z = \{c / c = a + z, \text{ for } a \in A\}$$



$(A)_z$

B

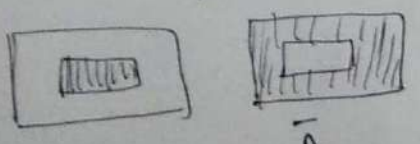
$\hat{B}$

Logic operations involving Binary images

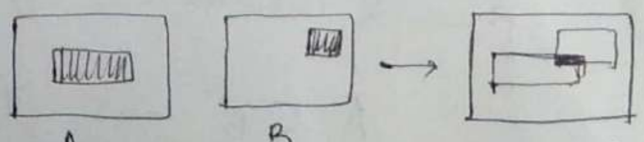
AND, OR & NOT

p	q	p·q	p OR q	NOT ($\bar{P}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

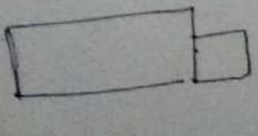
NOT

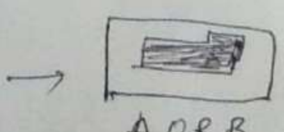


A Ā

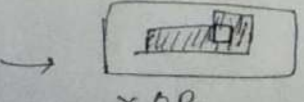


A AND B





A OR B



XOR.

Dilation & Erosion

2 imp morphological operation - Dilation & Erosion.

$Z^2 \rightarrow$ B & W image

$Z^3 \rightarrow$ gray scale image.

Dilation :-

A & B are sets in Z^2, then dilation of A by B denoted $A \oplus B$ is defined as

$$A \oplus B = \{ z / (B)_z \cap A \neq \phi \}.$$

$$= \{ z / [(B)_z \cap A] \subseteq A \}.$$

B → structuring element.

Dilation combines 2 set using vector addition.

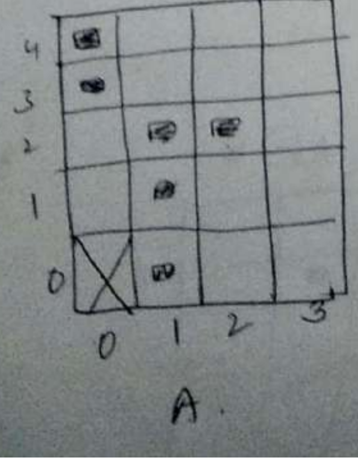
[eg $(a,b) + (c,d) = (a+c, b+d)$]

$$A \oplus B = \{ p \in Z^2 ; p = a + b, a \in A \ \& \ b \in B \}.$$

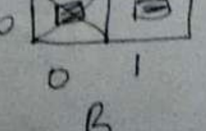
Eg:- $A = \{(1,0), (1,1), (1,2), (2,2), (0,3), (0,4)\}$.

$B = \{(0,0), (1,0)\}$ - structuring element (sub image).

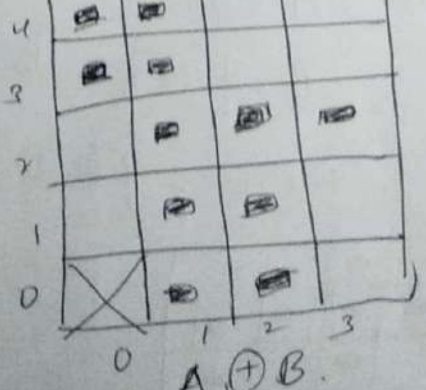
$A \oplus B = \{ (1,0),(1,1),(1,2),(2,2),(0,3),(0,4),(2,0)(2,1),(2,2)$
$(3,2),(1,3)(1,4)\}$.



A.



B



A ⊕ B.

Properties of dilation :-

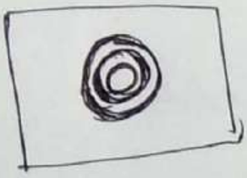
① commutative $A \oplus B = B \oplus A$

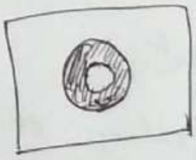
② Associative $A \oplus [B \oplus D] = [A \oplus B] \oplus D$.

③ $A \oplus B = \bigcup_{b \in B} A_b$

④ Invariant to translation $A_z \oplus B = (A \oplus B)_z$.

⑤ If $A \subseteq D$ then $A \oplus B \subseteq D \oplus B$.





(Expansion of region, fill or grow)

Appⁿ → Bridging gaps.

Erosion :-

for sets A & B in Z^2 the erosion of A by B, denoted $A \ominus B$, is defined as

$$A \ominus B = \{ z \,/\, (B)_z \subseteq A \}.$$

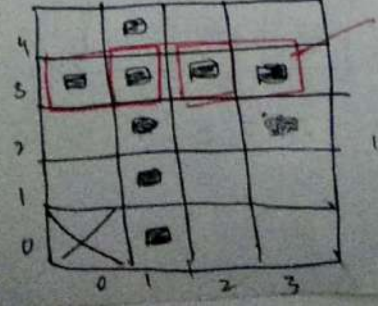
or $A \ominus B = \{ p \in Z^2 : p = a + b \in A \text{ for every } b \in B \}$

Eg:- $A = \{(1,0), (1,1), (1,2), (0,3), (1,3), (2,3), (3,3), (1,4)\}$

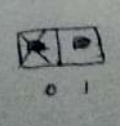
$B = \{(0,0), (1,0)\}$

$A \ominus B = \{(0,3), (1,3), (2,3)\}$

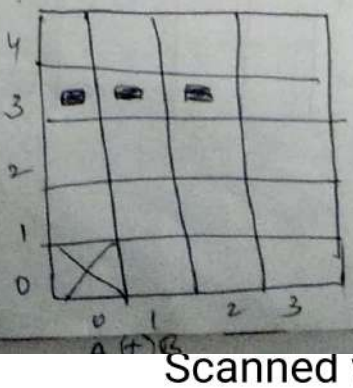
A

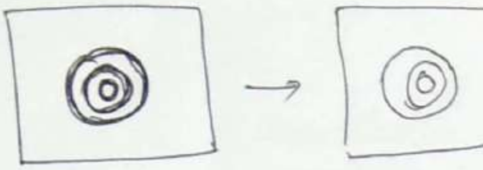


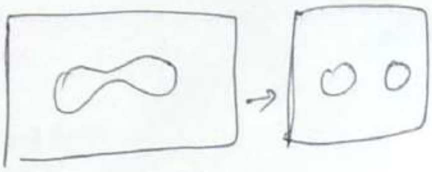
B



$A \ominus B$







Shink & reduce.

Dilation & Erosion are duals of each other

$$(A\ominus B)^{C} = A^{C}\oplus \hat{B}$$

Opening and closing - Operations

Opening generally smoothes the contour of an object, breaks narrow isthmuses & eliminates thin protrusions.

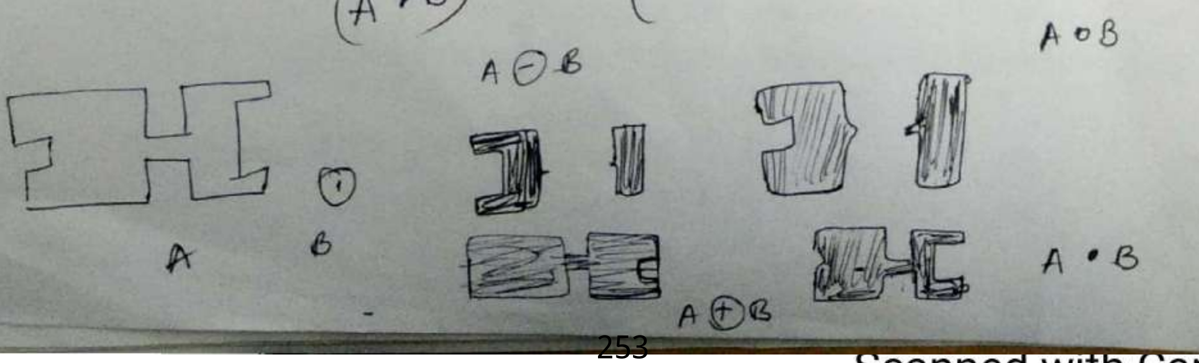
closing → fills gaps in the contour.

Opening: $A \circ B = (A\ominus B)\oplus B$. erosion of A by B followed by dilation of the result by B.

Closing $A \bullet B = (A\oplus B)\ominus B$ Dilation of A by B followed by erosion of the result by B.

opening & closing are duals of each other.

$$(A\bullet B)^{C} = (A^{C}\circ \hat{B})$$



Properties of opening:—

- → $A \circ B$ is a subset of A
- → If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$
- → $(A \circ B) \circ B = A \circ B$.

properties of closing:—

- → A is a subset of $A \bullet B$
- → If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$
- → $(A \bullet B) \bullet B = A \bullet B$.

Hit & Miss transformation

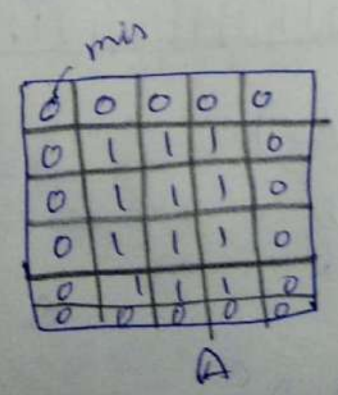
- → for finding local patterns of pixels.
- → basic tool for shape detection.

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

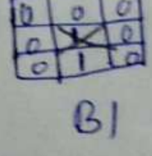
Set $A \circledast B$ contains all the points at which simultaneously B_1 found a match (hit) in A & B_2 found a match in A^c.

& $A \circledast B = (A \ominus B_1) - (A \oplus \hat{B}_2)$

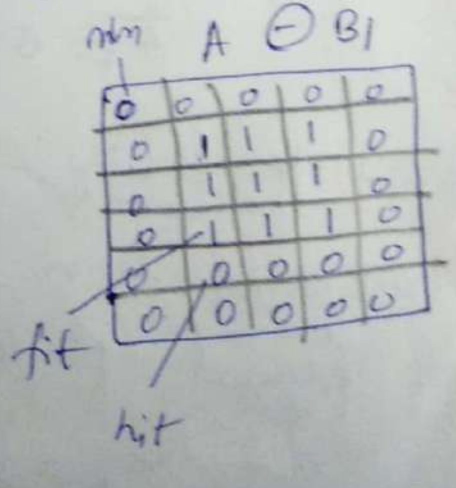
Eg!-



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



0	0	0
0	×	0
0	1	0



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

A^C

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

0	1	0
0	1	0
0	1	0

B_2

$A^C \ominus B_2$

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

$$(A \ominus B_1) \cap (A^C \ominus B_2) = \phi.$$

Some basic Morphological Algorithms

① Boundary Extraction:

$\beta(A) \rightarrow$ Boundary of set A.

$$\beta(A) = A - (A \ominus \underbrace{B)}_{\text{suitable Structuring element.}}$$

Eg:-

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

A

1	1	1
1	1	1
1	1	1

B

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

$A \ominus B$

$A - (A \ominus B)$

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

② Hole filling :- (Region filling)

Background region surrounded by a connected border of foreground pixels.

$X_0 = P.$

$$X_k = (X_{k-1} \oplus \underbrace{B}_{\text{structuring element (symmetric)}}) \cap A^C \qquad k = 1, 2, \ldots$$

③ Extraction of Connected Components

$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1, 2, 3 \cdots$$

$$x_0 = p$$

④ **Convex Hull** :- A set A is said to be 'convex' if the st. line segment joining any 2 points in A lies entirely within 'A'. The convex Hull H of an arbitrary set 'S' is the smallest convex set containing S.

Difference $(H - S) \rightarrow$ Convex deficiency of S.

Convex Hull & Convex Deficiency are useful for object description.

⑤ **Thinning** :- The thinning of a set A by a structuring element B, denoted $A \otimes B$

$$A \otimes B = A - (A \circledast B)$$
$$= A \cap (A \circledast B)^c.$$

A more useful expression for thinning A symmetrically is based on a seq of structuring elements

$\{B\} = \{B^1, B^2, B^3 \cdots B^n\}$ where B^i is a rotated version of B^{i-1}.

$$A \otimes \{B\} = ((\cdots((A \otimes B^1) \otimes B^2) \cdots \otimes B^n).$$

⑥ **Thickening** :
Morphological dual of thinning & is defined by

$$A \odot B = A \cup (A \circledast B)$$

transform.

⇒ Thinning of A is based on a sequence of structuring elements.

$$\{B\} = \{B^1, B^2, B^3, \dots B^n\}$$

where $B^i \rightarrow$ rotated version of B^{i-1}

⇒ Thinning by a sequence of structuring elements is

$$A \otimes \{B\} = ((\dots((A \otimes B^1) \otimes B^2) \cdot \dots) \otimes B^n)$$

⇒ This process is to thin A by one pass with B^1, then thin the result with one pass of B^2 & so on.. until A is thinned with one pass of B^n.

⇒ The entire process is repeated until no further changes occur.

Eg:

center pixel

0	0	0
X	1	X
1	1	1

B^1 → rotate clock wise once → B^2

X	0	0
1	1	0
1	1	X

B^2

1	X	
1	1	
1	X	

B^3

1	1	X
1	1	
X		

B^4

1	1	1
X	1	X

B^5

X	1	1
	1	1
		X

B^6

	X	1
	1	1
	X	1

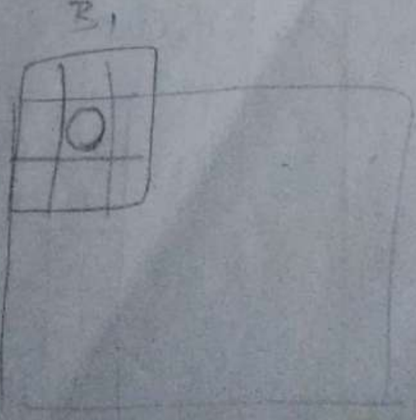
B^7

		X
	1	1
X	1	1

B^8

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

A



Note : thining = Center pixel = $\begin{cases} 0 & \text{completely match with B (0's 1's} \\ \text{retain} & \text{not " " "} \end{cases}$

Step 1 : $A_1 = A \otimes B^1$

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

$A_2 = A_1 \otimes B^2$

x	0	0
1	1	0
1	1	x

B^2

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

No. change

$A_3 = A_2 \otimes B^3$

1	x	0
1	1	0
1	x	0

B^3

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

$A_4 = A_3 \otimes B^4$

1	1	x
1	1	0
x	0	0

B^4

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

No change

1	1	1
x	1	x
0	0	0

15

B^5

$A_5 = A_4 \otimes B^5$

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

x	1	1
0	1	1
0	0	x

B^6

$A_6 = A_5 \otimes B^6$

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

0	x	1
0	1	1
0	x	1

B^7

$A_7 = A_6 \otimes B^7$

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

0	0	x
0	1	1
x	1	1

B^8

$A_8 = A_7 \otimes B^8$

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

$A_{8,4} = A_8 \otimes B^{1,2,3,4}$

$A_{8,5} = A_{8,4} \otimes B^5$; $A_{8,6} = A_{8,5} \otimes B^6$ & so on.... Can be performed

Thickening :

=> Dual of thinning

$$A \odot B = A \cup (A \circledast B)$$

B → structuring element suitable for thickening.

Also, $A \odot \{B\} = ((\ldots(A \odot B^1) \odot B^2) \ldots) \odot B^n)$

expressed as a sequential operation

=> usually, for thickening we follow 2 steps in practise

(a) thin the background (i.e $C = A^c$ thin C)

(b) complement the result (i.e form C^c

=> This method is followed to remove the disconnected points

Eg:

	1				1			1	
		1	1	1	1	1	1	1	
	1					1	1		

Set A

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

complement of A

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	

X = thinning the complement of A

			1						
	1				1	1	1	1	
	1	1	1	1	1	1	1	1	
	1	1	1	1	1	1	1		
									1

X^c (complement of X) thickened set

	1				1	1	1	1	
	1	1	1	1	1	1	1	1	
	1	1	1	1	1	1	1		

Final result with no disconnected points

Skeleton :

⇒ Given a point set A, the skeleton of A can be found by

$$S(A) = \bigcup_{K=0}^{M} S_K(A)$$

$$S_n = (A \ominus KB) - \left[(A \ominus KB) \underset{\substack{\uparrow \\ \text{opening}}}{\circ} B\right]$$

⇒ This eqn gives us a particular number of sub skeletons & the union of sub skeletons gives us the skeleton of the final of the given set A which is represented by S(A)

⇒ M → the last iterative step before A erodes to empty set

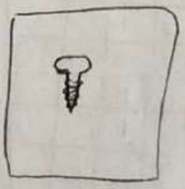
$(A \ominus KB)$ → A is ~~dilated~~ eroded with the structuring element B for successive K no. of times

$$M = \max\{K \mid (A \ominus KB) \neq \Phi\}$$

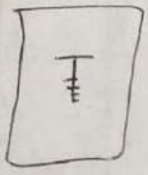
Eg:

1	1	1
1	①	1
1	1	1

→ center pixel



image



Skeleton

n = 0

A⊖KB

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

n = 1

		1		
		1		
	1	1	1	
	1	1	1	
	1	1	1	

n = 2

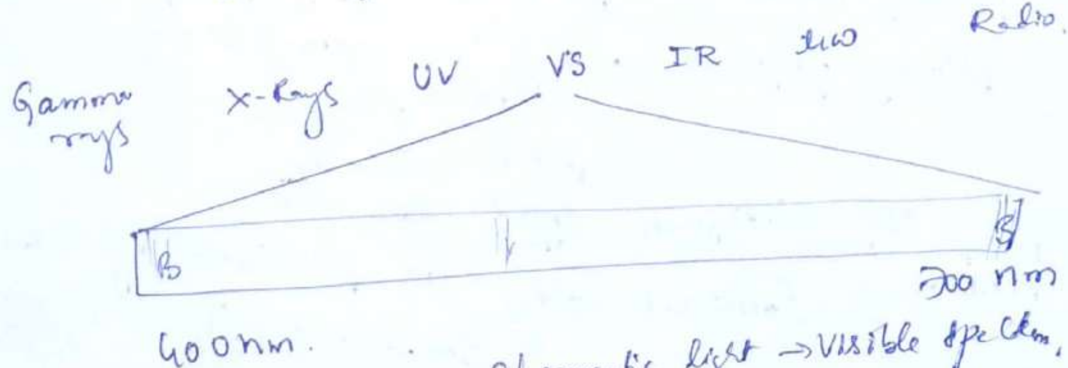
		1		

Color Fundamentals

①

1666 → Isac Newton → light passes thru' prism emerges cont. spectrum of lights (colors) VIBGYOR

Color → humans perceive in an object → nature of light reflected from the object.



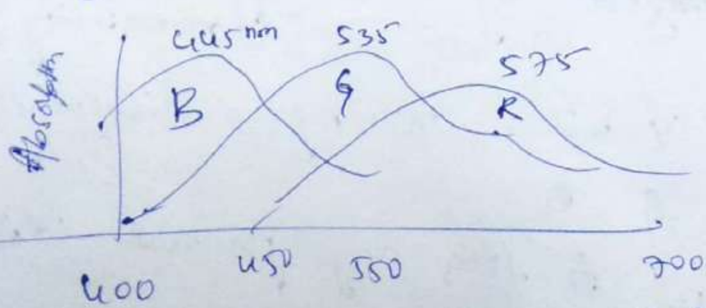
Chromatic light → visible spectrum.

quality of light → 3 things
- Radiance
- Luminance.
- Brightness

Radiance → total amount of energy flows from the light source (watts) W

Luminance → — " — perceived by observer (lumens) lm

Brightness → cannot be measured → subjective descriptor (intensity).



6-7 million cones are there in human eye

3 principal categories sensitive to RGB.

65% cones → R

33% → G

2% → B

primary colors → are seen due to absorption of human eye.

CIE → standard 1931 → specific w/L.

Commision Internationale de I'E clairage

435.8 mm – Blue

546.1 nm → Green

700nm → Red

Primary → added → Secondary colors of light. ②

Magenta → (R + B)
Cyan → (G + B)
Yellow → (R + G)

Mixing primary → white
Mixing secondary → Black.

Primary colors of light → absorbs one color & reflects the other 2.
RGB - primary
CYM → secondary

primary colors of pigments → CYM → primary
RGB → secondary.

To distinguish different colors

Hue → dominant color as perceived by an observer;
(dominant w/L)

Saturation → amount of white light mixed with hue.
(purity)

eg:- pink (red + white) is less saturated.
lavender (violet + white)

Brightness → chromatic notion of intensity.

Hue + Saturation — together is called 'Chromaticity'

Color may be characterized by its Brightness & Chromaticity.

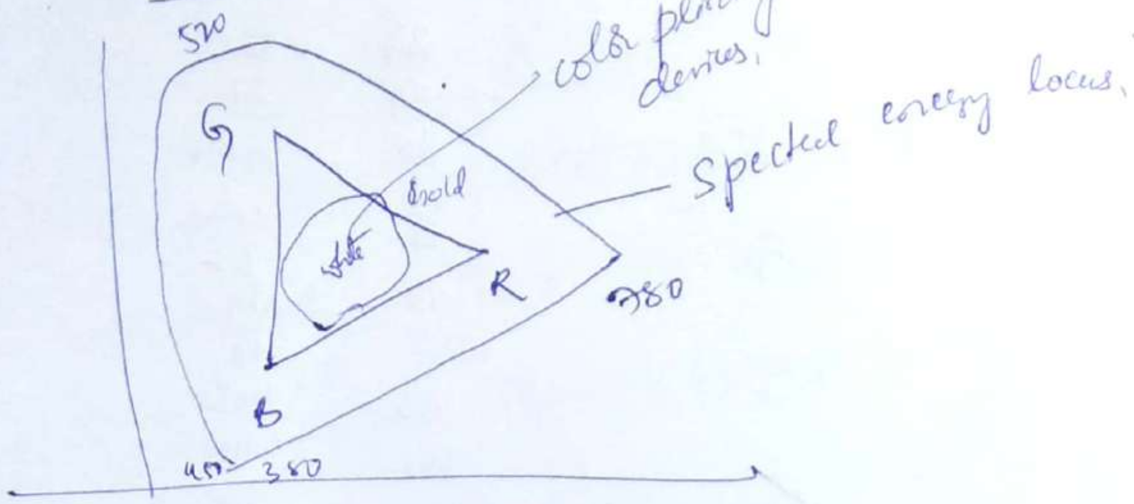
Tristimulus Values X, Y & Z. x, y & z → trichromatic coeffs.
R G B

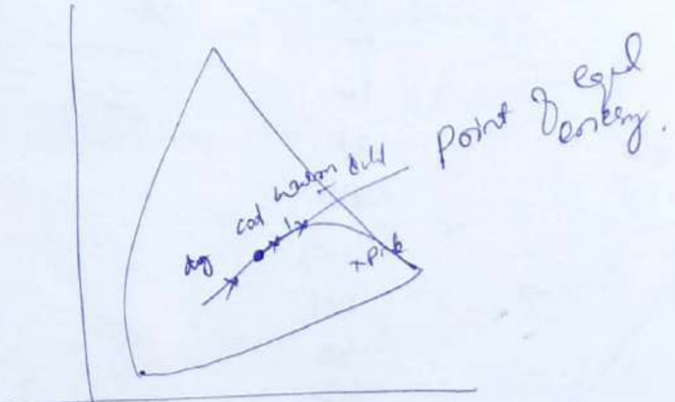
Amount of RGB → required to form any particular color.

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad z = \frac{Z}{X+Y+Z}$$

$$x + y + z = 1.$$

· Chromaticity Diagram.





Color models: RGB, HSI, CMY & CMYK

This chromatic diagram is useful for color mixing: a st. line joining any 2 points in the diagram defines all the different color variations that can be obtained by combining these 2 colors additively.

Color Models

Color model is a specification of a coordinate system & a subspace within that system where each color is represented by a single point.

Now a days most color models are oriented towards hardware (such as for color monitors & printers) or towards applications where color manipulation is a goal.

(such as in the creation of color graphics for animation).

There are 3 models.—

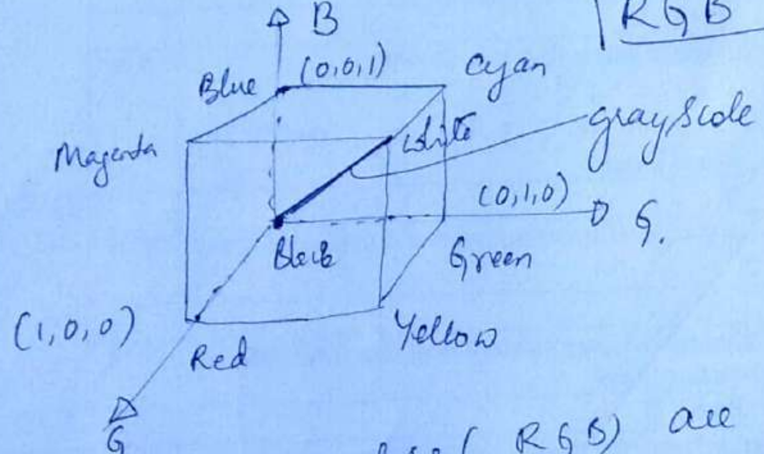
① RGB (Red, Green, Blue) model for color monitors & a broad class of color video cameras.

② CMY (Cyan, Magenta, Yellow) model - color printing.

CMYK ⟶ including black.

③ HSI (Hue, Saturation & Intensity) model ⟶ mainly for grayscale techniques.

RGB color Model - RGB→ primary colors of light.



The model is based on a Cartesian coordinate system.

Primary colors (RGB) are at 3 corners.

Secondary colors (CMY) are 3 other corners.

Black is at the origin & white is at the corner farthest from the origin.

The gray scale (points of equal RGB values) extends from black to white along the line joining 2 points.

CMY & CMYK color models

Primary colors of pigments and secondary colors of light.

Eg:- When a surface is coated with cyan pigment → illuminated with white light & no red light is reflected from the surface.

i.e cyan subtracts Red light from reflected white light

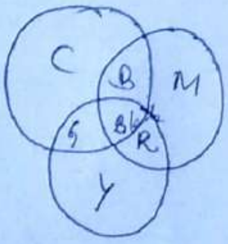
RGB to CMY conversion

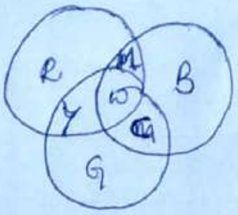
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$C = 1 - R = G + B$

$M = 1 - G = R + B$

$Y = 1 - B = R + G.$





HSI color model

Hue → is a color attribute that describes a pure color (like pure yellow / orange / red) where as

Saturation gives a measure of the degree to which a pure color is diluted by white light.

Brightness is a subjective descriptor that is practically impossible to measure.

HSI model → ideal for developing image processing algorithms.

RGB to HSI

then $H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G. \end{cases}$

$$\text{with } \theta = \cos^{-1}\left\{ \frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2 + (R-B)(G-B)\right]^{\frac{1}{2}}} \right\}$$

Saturation: $S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right]$

Intensity: $I = \frac{1}{3}(R+G+B)$

Assume the RGB values have been normalized to the range [0,1] & the angle θ is measured wrt red axis of HSI space.

HSI to RGB

There are 3 sectors of interest, corresponding to 120° intervals in the separation of primaries.

① **RG sector:** $(0^\circ \le H \le 120^\circ)$

RGB components are given by $B = I(1-S)$

$$R = I\left[1 + \frac{S \cos H}{\cos(60^\circ - H)}\right]$$

& $G = 3I - (R+B)$

② **GB sector:** $(120^\circ \le H < 240^\circ)$

new $\boxed{H = H - 120^\circ}$

RGB components are: ⑦

$$R = I(1-S)$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$$B = 3I - (R+G)$$

③ **BR sector** $(240^\circ \le H \le 360^\circ)$

then new $H = H - 240^\circ$

the RGB components are

$$G = I(1-S)$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

& $R = 3I - (G+B)$.

Pseudo color Image Processing (also called false color)

→ assigning colors to gray values based on a specified criterion.

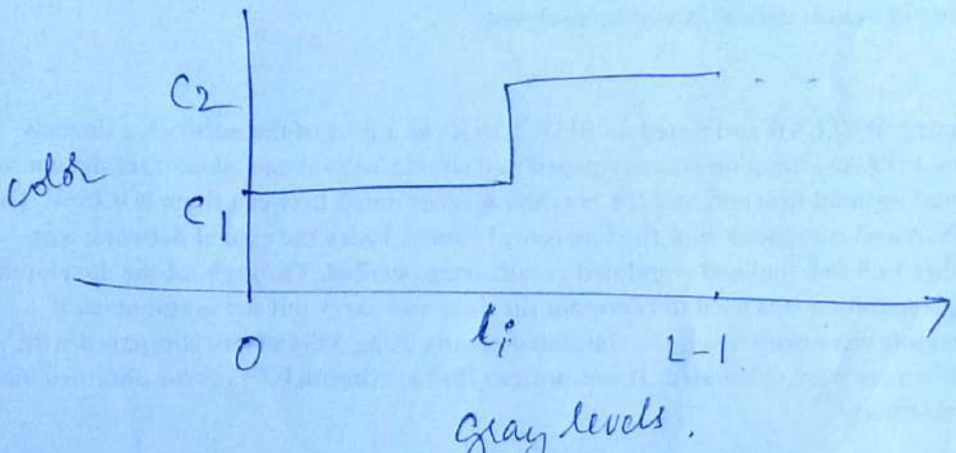
→ Principal use of pseudocolor is for human visualization & interpretation of gray-scale events in an image or seq. of images.

→ One of the principal motivation for using color is the fact that humans can discern thousands of color shades & intensities, compared to only 2 dozens or so shades of gray.

Intensity slicing

⑧

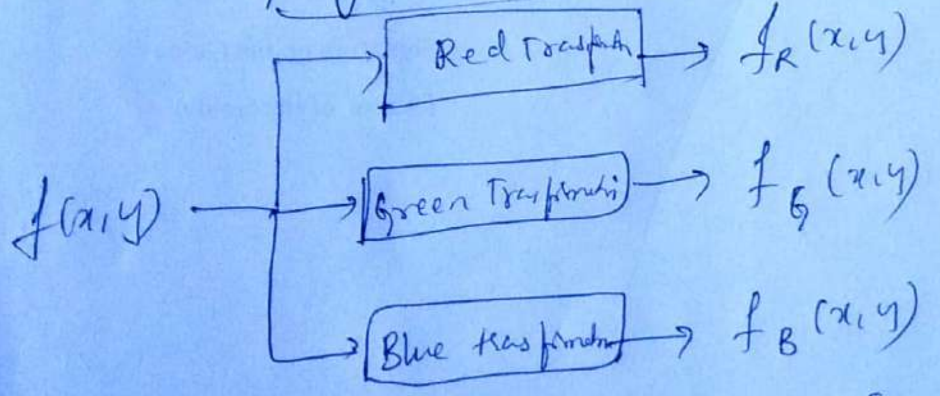
The technique of intensity slicing (density slicing) & color coding is one of the simplest examples of pseudo color image processing.



$l_0 \rightarrow$ Black
$l_{L-1} \rightarrow$ white.

When more levels are used, the mapping function takes on a staircase form.

Gray-level to color transformation



Functional block diagram for pseudo color image processing, f_R, f_G & f_B are fed into the corresponding Red, Green & Blue i/ps of an RGB color monitor.