Spring : 2019 :

2) a) i) Digital Image : A digital image is a representation of a two dimentional image as a finite set of digital values called picture elements or pixels.

ii) Resolution : Image resolution describes the image's level of detail. Image resolution is typically described in PPI, which refers to how many pixels are displayed per inch of image.

iii) Sampling : Sampling is a process which converts the continuous analog space into discrete space. Digitizing coordinates is called sampling.

Quantization : Quantization is a process of converting a continuous analog signal into a digital representation of that signal. Digitizing amplitudes (gray scale values) is called quantization.

2) c) Here, $M \times N = 256 \times 256$

$L = 2^k = 32$

$\Rightarrow 2^k = 2^5$

$\therefore k = 5$

$\therefore$ Bit size $= M \times N \times k = 256 \times 256 \times 5 = 327680$ (ans:)

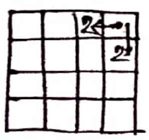
Expression to find no. of bits = image size × bit where

image size = row × column

bit = k

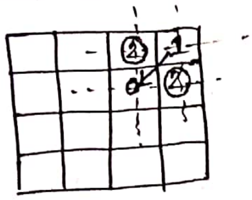
no. of gray level = L $[L = 2^k]$

5) a) i) No. A 4-path doesn't exist between a and b.



From a, we can move towards left and down where we found 2 in both cases but 2 isn't one of the elements of V. So, 4-path do not exist.

ii) step : 01 :

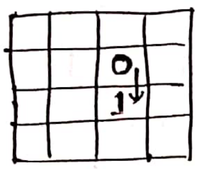


From a, we can move down to 0 which is diagonally adjacent. Here,

$N_4(a) \cap N_4(0)$ has no pixel with value from V.

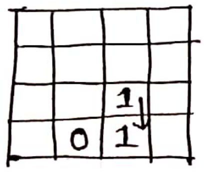
So, they are m-adjacent.

step : 02 :



We can move from 0 to 1 as they are 4 adjacent

step : 03 :

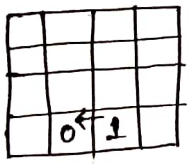


We can't move from 1 to 0 because

$N_4(1) \cap N_4(0)$ has a pixel with value from V.

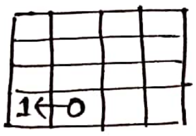
So we moved from 1 to 1 by 4 adjacency.

step : 04 :



We moved from 1 to 0 by 4 adjacency.

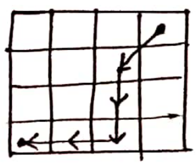
step : 05 :



We moved from 1 to b by 4 adjacency

So, m-path exists between a and b.

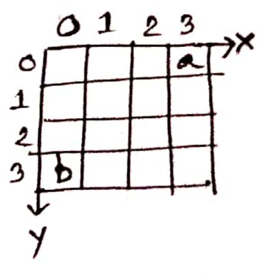
$\therefore$ The path : $a \rightarrow 0 \rightarrow 1 \rightarrow 1 \rightarrow 0 \rightarrow b$



iii) Distance from a to b : $D_4 = |3-0| + |0-3|$

$= 3+3$

$= 6$



$D_8 = \max(|3-0|, |0-3|)$

$= \max(3, 3)$

$= 3$

$D_e = \sqrt{(3-0)^2 + (0-3)^2}$

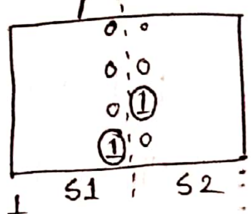
$= \sqrt{9+9}$

$= \sqrt{18}$

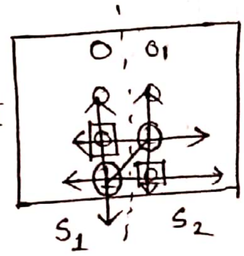
Fall-2019:

5) b) a) The two subsets aren't 4-adjacent.

b) The two subsets are 8-adjacent because there's a pixel value from V in the boundary which is situated in 8-adjacency.



Here, the circled values are in 8-neighbor and the values are from V. S_0, $S1$ and $S2$ are 8-adjacent.



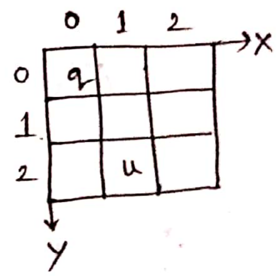
c) Here, the circled values are from V. They are diagonally adjacent and the 4 neighbor adjancents of both S_1 and S_2 intersected where the values are 0 which isn't from V. S_0, S_1 and S_2 are m-adjacent as $N_4(1)$ from $S_1 \cap N_4(1)$ from S_2 aren't from V.

Fall-2020:

4) a) i) No, pixels p and q aren't 4-adjadent.

Yes, pixels p and q are 8-adjacent.

No, pixels p and q aren't m-adjacent



ii) Euclidean distance $= \sqrt{(0-1)^2 + (0-2)^2}$

$= \sqrt{1+4}$

$= \sqrt{5}$

Chessboard distance $= \max(|0-1|, |0-2|)$

$= \max(1, 2)$

$= 2$

City block distance $= |0-1| + |0-2|$

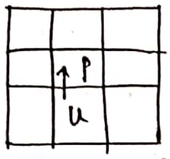
$= 1+2$

$= 3$

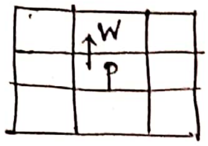
Scanned with CamScanner

iii) Yes, there's a 4-path from pixel u to q.

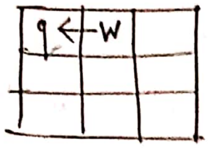
step:01



step:02



step:03



p, q, u, w belongs to V means $V = \{p, q, u, w\}$

$\therefore$ The path is $u \rightarrow p \rightarrow W \rightarrow q$

b) $\text{Ratio}_{row} = \frac{4}{6} = \frac{2}{3} \approx 0.67$

$$\therefore \text{Row}_{positions} = \frac{\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}}{\text{Ratio}_{row}}$$

$$= \begin{bmatrix} 0.67 & 1.33 & 2 & 2.67 & 3.33 & 4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 2 & 3 & 3 & 4 \end{bmatrix}$$

$\text{Ratio}_{column} = \frac{4}{6} = \frac{2}{3} \approx 0.67$

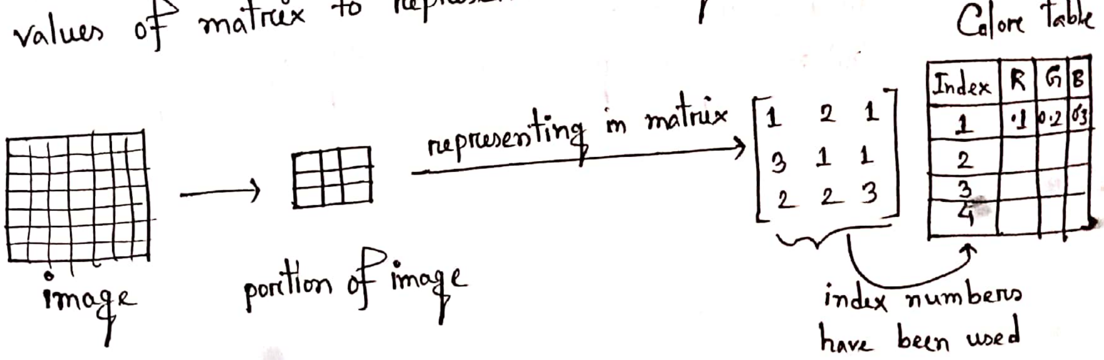
$$\therefore \text{Column}_{positions} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 3 \\ 4 \end{bmatrix}$$

$\therefore$ After zooming, image will be :

180	180	160	160	160	140
180	180	160	160	160	140
110	110	110	120	120	140
110	110	140	120	120	120
110	110	140	120	120	120
120	120	140	160	160	170

Spring - 2020:

7) a) We have to store RGB values in a colour table where each of the RGB values will be stored in an index. then we will use the index numbers to determine the values of matrix to represent an image.



Chapter : 03 [28 marks]

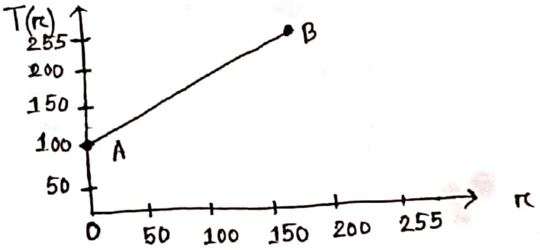
Spring - 2019:

2) d) Three major goals of image enhancement in spatial domain:

i) highlighting interesting detail
ii) remove noise
iii) making images more visually appealing

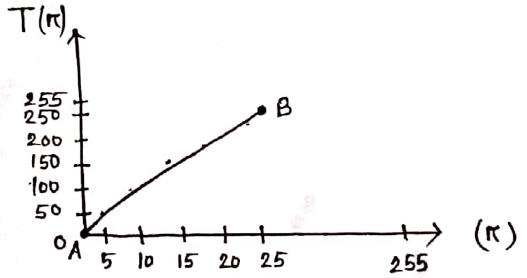
Spring - 2020:

3) a) i)



A (r, T(r)) = (0, 100)
B (r, T(r)) = (155, 255)

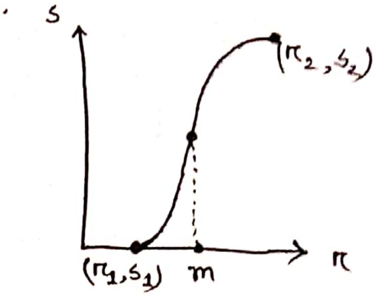
ii)



A (r, T(r)) = 0, 0
B r, T(r) = 25.5, 255

b) i) A simple image enhancement technique that improves the contrast in an image by stretching the range of intensity values is 'contrast stretching'. Contrast stretching means that the bright pixels in the image will become brighter and the dark pixels will become darker which means higher contrast image.

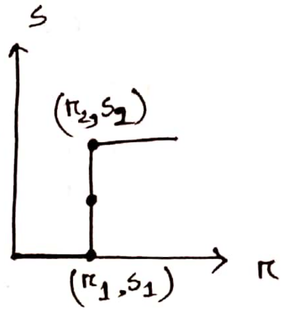
First of all, we have to set a value 'm'. If a pixel is greater than 'm', then we will increase pixel value so that it becomes brighter. On the other hand, if a pixel is less than 'm', then we will decrease the image value so that it becomes darker.



ii) In contrast stretching, when the output image gives the values 0 or 255 only, it act's like thresholding. The only output will be 0 or 255.

For thresholding,

$r_1 = r_2$, $s_1 = 0$, $s_2 = L-1$



c) i) I =

01111000	00111100
00111100	11111010

plane : 8

0	0
0	1

plane : 7

1	0
0	1

plane : 6

1	1
1	1

plane : 5

1	1
1	1

plane : 4

1	1
1	1

plane : 3

0	1
1	0

plane : 2

0	0
0	1

plane : 1

0	0
0	0

ii) if $n = 8$, $2^{8-1} = 2^7 = 128$

0	0
0	1

$\longrightarrow$

0	0
0	128

if $n = 7$, $2^{7-1} = 2^6 = 64$

1	0
0	1

$\longrightarrow$

64	0
0	64

$\therefore$ Reconstructe image =

64	0
0	192

2) a) Median filter : a median filter operates over a window by selecting the median intensity in the window.

In mean filtering we find the average of all the pixel values. But in median filtering, we sort all the pixel values and pick mid value. That's why no new pixel value is introduced and never replace with largest or smallest value. That's how all the noises are removed. That's why median filtering is much better suited than averaging for the removal of salt-and-pepper noise.

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

ii) $156\times1 + 158\times1 + 156\times1 + 158\times1 - 154\times8 + 156\times1 + 157\times1 + 158\times1 + 160\times1$

$= 27$

iii) $0\times156 - 1\times158 + 0\times156 - 1\times158 + 4\times154 - 1\times156 + 0\times157 - 1\times158 + 0\times160$

$= -14$

iv) $\dfrac{156\times1 + 158\times1 + 156\times1 + 158\times1 + 154\times1 + 156\times1 + 157\times1 + 158\times1 + 160\times1}{9}$

$= 157$

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

3x3 mean filter

v) 154 156 156 156 $\boxed{157}$ 158 158 158 160

median

156	158	156
158	157	156
157	158	160

} after applying median filter

Scanned with CamScanner

c) Equalizing original histogram :

Gray Level r_k	No. of pixels, n_k	$P_r(r_k) = \frac{n_k}{N}$	CDF	$(L-1) \times CDF$ $255 \times CDF$	(Equalized Gray Level) Rounding off H_k	No of pixel (output)
2	43	0.1075	0.1075	27.4125	27	43
18	11	0.0275	0.135	34.425	34	11
33	47	0.1175	0.2525	64.3875	64	47
58	31	0.0775	0.33	84.15	84	31
67	27	0.0675	0.3975	101.3625	101	27
96	49	0.1225	0.52	132.6	133	49
114	71	0.1775	0.6975	177.8625	178	71
152	21	0.0525	0.75	191.25	191	21
184	14	0.035	0.785	200.175	200	14
206	52	0.13	0.915	233.325	233	52
220	24	0.06	0.975	248.625	249	24
245	10	0.025	1	255	255	10
	N=400					

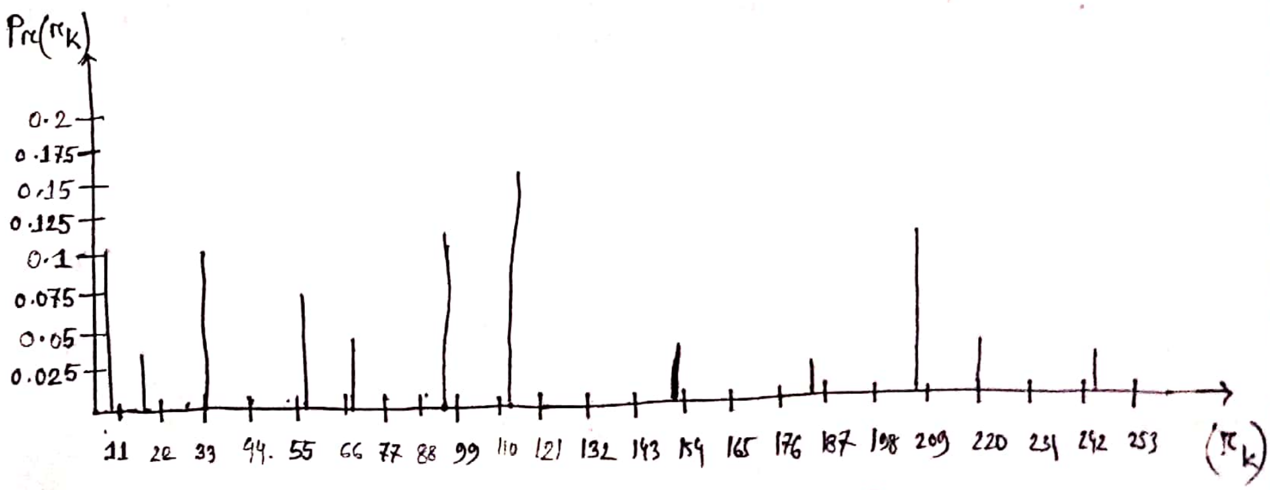
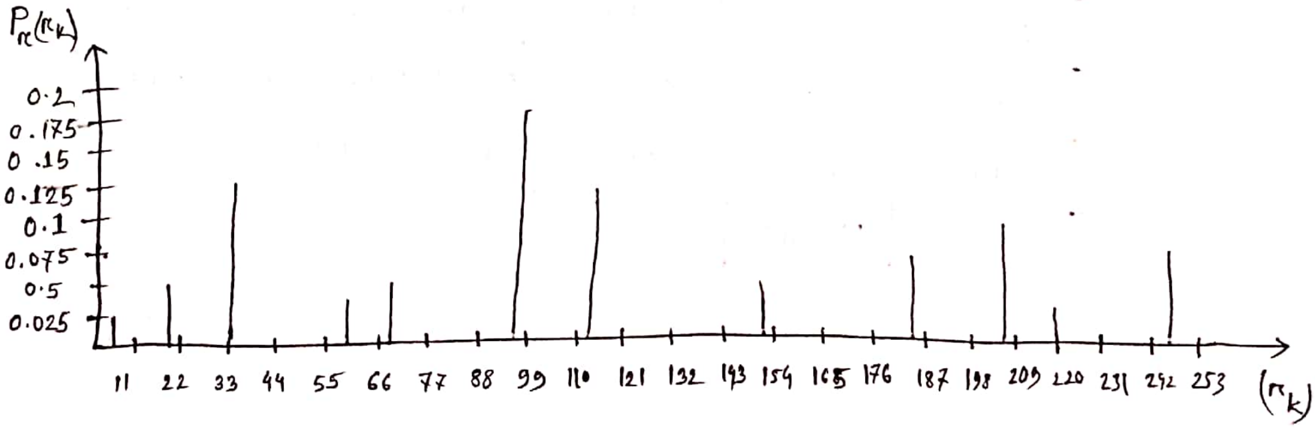


figure : normalization histogram of original image

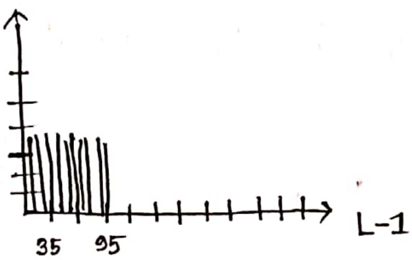
Equalizing Desired histogram

Gray Level r_k	No of pixels, n_k	PDF $P_r(r_k)$ $\frac{n_k}{N}$	CDF	(L-1) CDF 255xCDF	(Equalized Gray Level) Rounding off S_k
2	10	0.025	0.025	6.375	6
18	24	0.06	0.085	21.675	22
33	52	0.13	0.215	54.825	55
58	14	0.035	0.25	63.75	64
67	21	0.0525	0.3025	77.1375	77
96	71	0.1775	0.48	122.4	122
114	49	0.1225	0.6025	153.6375	154
152	27	0.0675	0.67	170.85	171
184	31	0.0775	0.7475	190.6125	191
206	47	0.1175	0.865	220.575	221
220	11	0.0275	0.8925	227.5875	228
245	43	0.1075	1	255	255
	N=400				



Fall-2020:

6) a) i)



After image negative,

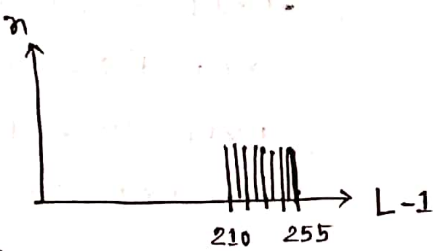
$s = (L-1) - r$ [transformation function]

For $r = 160$, $s = 255 - 160 = 95$

For $r = 220$, $s = 255 - 220 = 35$

After the transformation, all the pixels moved towards left. That means the image will become darker.

ii)



After adding 50 to all pixels,

$s = 50 + r$ [transformation function]

For $r = 160$, $s = 50 + 160 = 210$

For $r = 220$, $s = 50 + 220 = 270$

After the transformation, all the pixels will move towards right. It's called histogram sliding. That means the image will become brighter.

iii)

$s = T(r)$

$$s = \begin{cases} 1 & \text{if } r > 128 \\ 0 & \text{if } r \leq 128 \end{cases}$$

In this transformation,

3) b) ii) The objectives of sharpening spatial filters are :

i) Highlighting the fine detail in an image

ii) Enhancing details that has been blurred

Here,

$$\frac{\partial f}{\partial y} = G_y = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

$$\frac{\partial f}{\partial x} = G_x = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

$\therefore$ Gradient magnitude $= |G| = |G_x| + |G_y|$

$$= |(-1)\times 1 + 0\times 6 + 1\times 4 + (-2)\times 5 + 0\times 2 + 2\times 7 + (-1)\times 2 + 0\times 6 + 1\times 0| +$$

$$|(-1)\times 1 + (-2)\times 6 + (-1)\times 4 + 0\times 5 + 0\times 2 + 0\times 7 + 1\times 2 + 2\times 6 + 1\times 0|$$

$$= |-1+0+4-10+0+14-2+0+0| + |-1-12-4+0+0+0+2+12+0|$$

$$= |5| + |-3|$$

$$= 5+3$$

$$= 8 \quad \text{(ans:)}$$

$\therefore$ Angle, $\theta = \tan^{-1} \frac{G_y}{G_x}$

$$= \tan^{-1}\left(\frac{-3}{5}\right) \quad [G_y = -3 \,;\, G_x = 5]$$

$$\approx -30.964^\circ \quad \text{(ans:)}$$

Chapter : 08 [14 marks]

Spring : 2020 :

5) a) Entropy = 5.3 bits/pixel

We know,

compression, $C = \frac{n_1}{n_2}$ [n_1 = original image bit size, n_2 = bit size after compression]

$= \frac{8}{5.3}$ [For maximum, n_2 = entropy]

$= 1.509433962$

≈ 1.51 pixel (ans:)

b) i)

r_k	n_k	$P_k(r_k) = \frac{n_k}{N} = \frac{n_k}{32}$
21	12	0.375
95	4	0.125
169	4	0.125
243	12	0.375

$\therefore$ entropy $= -(0.375 \log_2 0.375 + 0.125 \log_2 0.125 + 0.125 \log_2 0.125 + 0.375 \log_2 0.375)$

$= 1.811278124$

≈ 1.81 (ans:)

ii) After sorting:

21	0.375	0.375	0.375	0.625
243	0.375	0.375	0.625	0.375
95	0.125	0.25		
169	0.125			

applying huffman code:

21	0.375 [1]	0.375 [1]	0.625 [0]
243	0.375 [00]	0.375 [00]	0.375 [1]
95	0.125 [010]	0.25 [01]	
169	0.125 [011]		

	huffman code
∴ 21 =	1
∴ 243 =	00
∴ 95 =	010
∴ 169 =	011

(ans:)

iii) $L_{avg} = (0.375 \times 1) + (0.375 \times 2) + (0.125 \times 3) + (0.125 \times 3)$

$= 0.375 + 0.75 + 0.375 + 0.375$

$= 1.875$

∴ Compression ratio, $C = \frac{8}{1.875} = 4.266666667 \approx 4.3$ pixels (ans:)

iv) $R_D = 1 - \frac{1}{C} = 1 - \frac{1}{4.266666667} = 0.765625 \approx 0.766$

v) After applying huffman coding, we can see that 76.6% data was redundant.

∴ Huffman effectiveness $= \frac{\text{entropy}}{L_{avg}} = \frac{1.81}{1.875} = 0.9653333333$

≈ 0.9653

∴ Effectiveness = 96.53% (ans:)

c) ii)

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code word)	Dictionary Entry
	51			
51	4	0	8	51-4
4	3	4	9	4-3
3	7	3	10	3-7
7	51	7	11	7-51
51	4			
51-4	3	8	12	51-4-3
3	7			
3-7	51	10	13	3-7-51
51	4			
51-4	3			
51-4-3	7	12	14	51-4-3-7
7	51			
7-51	4	11	15	7-51-4
4	3			
4-3	7	9	1	4-3-7
7		7		

Dictionary Location	Entry
0	51
1	4-3-7
2	
3	3
4	4
5	
6	
7	7
8	51-4
9	4-3
10	3-7
11	7-51
12	51-4-3
13	3-7-51
14	51-4-3-7
15	7-51-4

After encoding :

0 4 3 7 8

10 12 11 9 7

$$\text{Compression ratio} = \frac{16\times 3}{10\times 4} = \frac{48}{40}$$

Compressed image size < Original image size

$\therefore$ Compression is achieved

Scanned with CamScanner

Spring, 2019 :

3) c) The Golomb code of n with respect to m, denoted $G_m(n)$, is a combination of the unary code of quotient, floor $\lfloor n/m \rfloor$ and the binary representation of remainder (n mod m).
$G_m(n)$ is constructed as follows:

Step:01: Form the unary code of quotient $\lfloor n/m \rfloor$

Step:02: Let $k = \lceil \log_2 m \rceil$, $c = 2^k - m$, $r = n \bmod m$ and compute truncated remainder r' such that

$$r' = \begin{cases} r \text{ truncated to } k\text{-}1 \text{ bits} & 0 \le r < c \\ r+c \text{ truncated to } k \text{ bits} & \text{otherwise} \end{cases}$$

Step:03: Concatenate the results of steps 1 & 2

Calculation for $G_4(9)$:

Step:01: $\lfloor 9/4 \rfloor = 2$ $\quad \therefore 110$

Step:02: $k = \lceil \log_2 4 \rceil = 2$ $\quad c = 2^k - 4 = 2^2 - 4 = 0$

$r = 9 \bmod 4 = 1$ $\quad \therefore r' = 01$

Step:03: 11001

(ans:)

Extra: $G_4(7)$

Step:01: $\lfloor 7/4 \rfloor = 1$ $\quad \therefore 10$

Step:02: $k = \log_2 4 = 2$ $\quad c = 2^2 - 4 = 0$ $\quad r = 7 \bmod 4 = 3$

$\therefore r' = 11$

Step:03: 1011 (ans:)

Chapter : 09 [14 marks]

Spring : 2020 :

4) a) To obtain image (b) from image (a), we can apply 'opening method'. Opening of image A by structuring element B, denoted by $A \circ B$ is simply an erosion followed by a dilation.

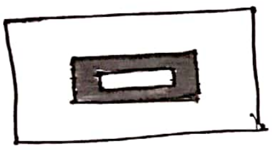
$$A \circ B = (A \ominus B) \oplus B$$

For this transformation, structuring element will be :

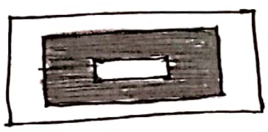
1	1	1
1	1	1
1	1	1

At first the extra boxes will be removed from the outer portion by applying erosion. Then there will increase a layer to outer portion by applying dilation.

step : 01



step : 02



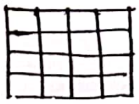
b) Dilation: Dilation of image f by structuring element s is given by $f \oplus s$. The structuring element s is positioned with it's origin at (x,y) and the new pixel value is determined using the rule :

$$g(x,y) = \begin{cases} 1 & \text{if } s \text{ hits } f \\ 0 & \text{otherwe} \end{cases}$$

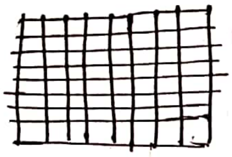
Erosion: Erosion of image f by structuring element s is given by $f \ominus s$. The structuring element s is positioned with it's origin at (x,y) and the new pixel value is determined using the rule :

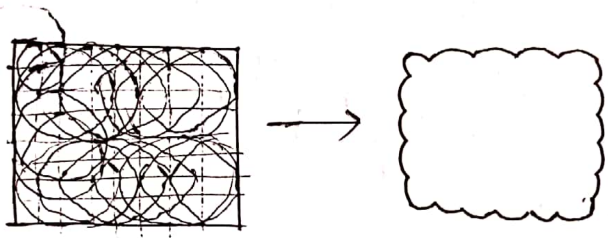
$$g(x,y) = \begin{cases} 1 & \text{if } s \text{ fits } f \\ 0 & \text{otherwise} \end{cases}$$

i) Suppose,

$A =$  $B =$ 

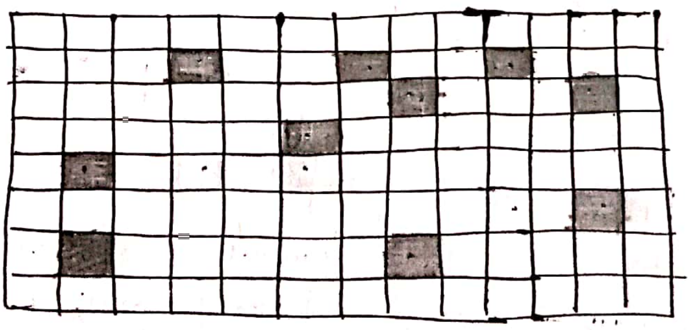
if $A = L \times L$ and $B = \frac{L}{2} \times \frac{L}{4}$ then $A \oplus B$ will make A increased by it's four sides by $\frac{L}{2}$.

$A \oplus B =$ 

ii) $A \ominus c =$ 

c) i) SE's for 4connected end points:

ii)



iii) Yes, it's possible to achieve edge detection using morphological operations. If we do erosion on an image and then subtract the erosioned image from the original image, we can obtain the edge. Similarly, if we operate dilation on an image and then subtract the real image from the dilated image, we can achieve edge.

$$A - (A \ominus B) \quad \text{or} \quad (A \oplus B) - A$$

Fall : 2020 :

3) a) ii) a)

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

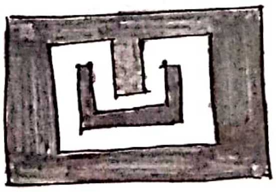
b)

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

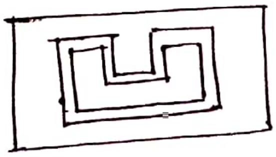
c)

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

b) i) $B = (A \oplus S) \cup A^c =$



$B = (A \oplus S) \cap A^c =$



Spring-2019:

6) a) Erosion: Erosion of a set A by structuring element B:

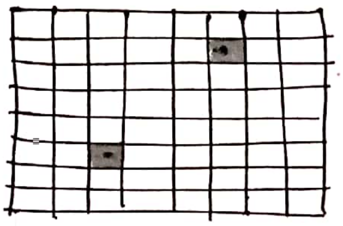
- set of all points z, such that B translated by z is contained in A.

$$A \ominus B = \{ z \mid (B)_z \subseteq A \}$$

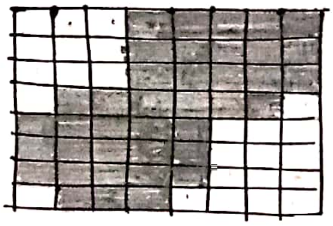
Dilation: Dilation of a set A by structuring element B:

- all z in a such that B hits A when origin of $B = z$
- such that overlap A by at least one element

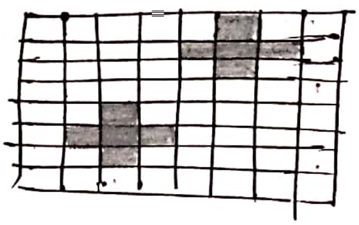
d) i)



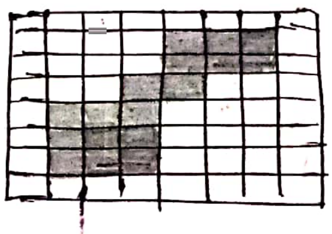
ii)



iii)



iv)



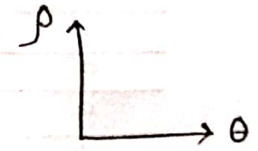
Chapter : 10

Spring : 2020

1) a) Equation of the curve : $\rho = x\cos\theta + y\sin\theta$ where

θ = angle between line

ρ = diameter



Hough space

b) For each black square given in picture, we get (x, y) coordinates. We get $\rho = x\cos\theta + y\sin\theta$ by putting (x, y) values:
Assume, horizontal axis $= x$, vertical axis $= y$:

x	y	$\theta = 0^\circ$	$\theta = \frac{\pi}{2}$
1	1	1	1
1	2	1	2
1	3	1	3
1	4	1	4
1	5	1	5
2	1	2	1
2	5	2	5
2	7	2	7
2	8	2	8
2	9	2	9
3	1	3	1
3	5	3	5
3	7	3	7
3	9	3	9
4	1	4	1
4	5	4	5
4	7	4	7
4	8	4	8
4	9	4	9
5	1	5	1
5	2	5	2
5	3	5	3
5	4	5	4
5	5	5	5
6	0	6	0
6	10	6	10
7	3	7	3
7	4	7	4
7	5	7	5

x	y	$\theta = 0$	$\theta = \frac{\pi}{2}$
8	3	8	3
8	5	8	5
8	9	8	9
8	10	8	10
9	3	9	3
9	4	9	4
9	5	9	5
9	10	9	10
10	0	10	0
11	11	11	11

The accumulator cell in (ρ, θ) space is given below:

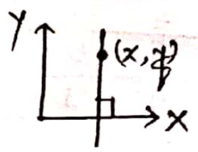
θ \ ρ	2	5
0	5	5
$\frac{\pi}{2}$	2	8

$\therefore$ Maximum value $= 8$ $\left[(\rho, \theta) = \left(5, \frac{\pi}{2}\right)\right]$

Fall: 2020:

2) a) i) The limitation in parameter space representation in Hough transformation is it doesn't work for vertical lines i.e. lines those are parallel to Y-axis. Because they have undefined slope.

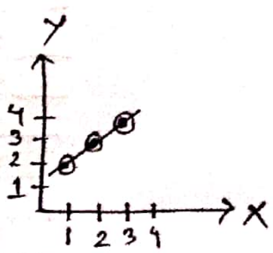
$$m = \tan\theta$$
$$= \tan 90°$$
$$= \infty$$



Therefore, lines must be converted into polar coordinate.

ii) Straight line equation: $y = mx + c$

$$\Rightarrow y - mx = c$$
$$\therefore c = -mx + y \quad \ldots (1)$$



Scanned with CamScanner

Putting (1,2) in equation ① : $c = -m+2$ ... ... (II)

m	0	2	1
c	2	0	1

Putting (2,3) in equation ① : $c = -2m+3$ ... ... (III)

m	0	1
c	3	1

Putting (3,4) in equation ① : $c = -3m+4$ ... ... (IV)

m	0	1
c	4	1

Plotting the points for equation (II), (III), (IV) in 5x5 accumulator array:

c \ m	0	1	2	3	4
4	1	0	0	0	0
3	1	0	0	0	0
2	1	0	0	0	0
1	0	3	0	0	0
0	0	0	1	0	0

Here, maximum value = 3 at point $(m, c) = (1, 1)$.
Which means line equation (II), (III), (IV) intersected at a specific point. So, (1, 2), (2, 3), (3, 4) are collinear.

[Showed]

$\therefore$ Equation of line : $y = 1 \times x + 1$ $[(m, c) = (1, 1)]$

$\therefore y = x + 1$ (ans:)

Chapter : 11

Spring : 2020 :

6) a) i) Chain code, CC = 001232446567

Circular first difference, CFD = 011172027111

Normalized CFD, NCFD = 011172027111 (ans:)

ii)

Chain code, CC = 012234546607

Circular first different, CFD = 110111720271

Normalized CFD, NCFD = 011172027111 (ans:)

Chapter wise easy to hard order এ সাজানো :

Chapter : 1-2 : at least 5 marks (সোজা chapter)
→ definition
→ adjacency, connectivity, path
→ distance measure

Chapter : 11 : at least 3 marks
→ chain code [এটা যে না পারবে সে ছাগল -_-]

Chapter : 6 : 3/4 marks আসতে পারে (ছোট video, সোজা জিনিস)
→ additive + subtractive color model
→ color subtraction
→ RGB vs CMY

Chapter : 8 : 14 marks (previous থেকে repeat হয় বেশি)
→ Entropy, Redundancy, Compression
→ Huffman coding, Golomb Coding, Arithmetic Coding
→ LZW coding, Run Length Encoding

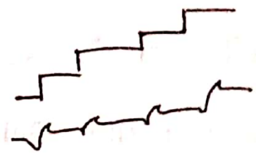
Chapter : 9 : at least 7 marks (risky chapter, পড়েও 0 পাইতে পারি)
→ Erosion, Dilation, Opening, Closing
→ Hit-miss, thinning, thickening

Chapter : 3 : 28 marks (ধৈর্যের পরীক্ষার জন্য এই chapter) :)
→ Transformation
→ Histogram
→ Filtering

Chapter : 10 : at least 14 marks (আল্লাহর হাতে ছেড়ে দাও)
→ hough transform
→ canny edge detection
→ segmentation

Low intensity = black (dark color)
High intensity = white (light color)

Mach Band Effect : actual intensity :
perceived intensity :



not continuous data = digital image
(discrete image)

human brain 10^{10} সংখ্যক intensity level detect করতে পারে

amplitude + time discrete } digital signal

amplitude + space discrete } digital image ← { pixel value discrete + ২টা pixel এর মধ্যবর্তী ফাঁকা space এর value discrete

Image acquisition : i) illumination source
ii) scene
iii) sensor

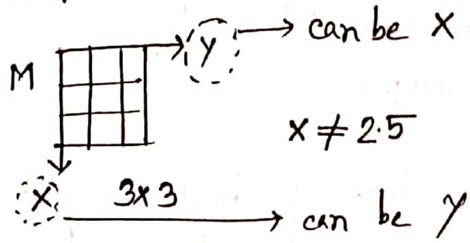
low wavelength, high frequency } gamma ray

high wavelength, low frequency } radio frequency

absorption : whole light photon কে object absorb করে, no reflection
diffuse reflection : scattered reflection (কয়েকটা reflection ১টা point এ)
specular reflection : mirror এ যেভাবে হয় (১টা reflection ১টা point এ)
transparency : যেদিকে ray পড়বে ওদিক ঐ দিকেই direct reflection :
refraction : density অনুযায়ী ray আগাবে
fluorescence : কিছুটা absorb করে, কিছুটা reflect করে
subsurface scattering : surface এ scattered
phosphorescence : time এর ১টা ব্যাপার, else fluorescence এর মতই
interreflection : optical fible :

Incoming energy কোন object এ পতিত হয়ে reflect হলে কোন sensor তাকে capture করে corresponding একটা voltage generate করে। এরকম কয়েকটা sensor বসিয়ে একটা image capture করি। বর্তমান digital camera, mobile এ যে sensor use করা হয় সেটা হচ্ছে CCD (Charge-Coupled Device)। এটা আবিষ্কারের পরই digital image capture করা সম্ভব হয়। continuous image কে discrete image বানায় CCD।

M = no. of row
N = no. of column



Simple image formation model : $f(x,y) = i(x,y) \times r(x,y)$

Example :

0   1   0 ⟶ intensity

•   •   • ⟶ points

(0,0)   (0,1)   (0,2) ⟶ স্থানাঙ্ক/coordinate

$f(0,1) = 1$ means (0,1) coordinate এর intensity 1.

x, y = coordinate
$f(x,y)$ = intensity

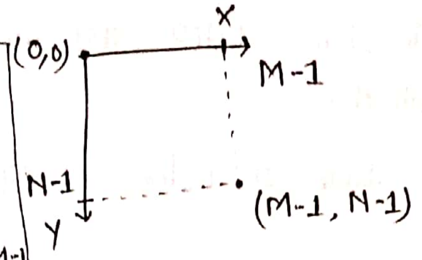
$0 < i(x,y) < \infty$: যে light কোন object এ পড়ছে তার illumination/intensity $0-\infty$ হতে পারে। 0 means কোন light পড়েনি, ∞ means max which is 10^{10} for human brain

$0 < r(x,y) < 1$: $r(x,y) = 0.5$ means কোন object এ যে পরিমাণ light আসছে তার 50% reflect করছে। reflect হয়ে যে intensity আসে সেটাই sensor capture করে।

কতটুকু photon আসলো আর কতটুকু reflect হলো তার multiplication-ই হচ্ছে image formation।

space এর সাথে related হওয়ায় spatial coordinate

$$A_{M,N} = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,M-1} \\ a_{1,0} & a_{1,1} & a_{1,M-1} \\ & & \\ a_{N-1,0} & a_{N-1,1} & a_{N-1,M-1} \end{bmatrix}$$



steps of image digitization

1. sampling
2. digitization

Illumination এর unit = lumen

Sampling :

→ digitizing coordinates

→ a process which converts the continuous analog space into discrete space

Quantization:

→ digitizing amplitudes (gray scale values)

→ a process of converting a continuous analog signal into a digital representation of that signal

whole image space কে ছোট ছোট box এ ভাগ করাটাই sampling। Intensity অনুযায়ী ভাগ ভাগ করাকে Quantization বলে। কোন image কে save করলে তা depend করে sampling rate (১টা image কে কত টুকরা করা হলো) এবং কত bit দিয়ে color represent করছি ($2^0 = 1$ (১টা color), $2^2 = 4$ (৪টা color), $2^3 = 8$ (8 টা color)) সেটার উপর। এভাবে image টা save করতে কত bit দরকার জানা যাবে। Gray Level eq^n : $L = 2^k$ → bit

L = colors (no. of gray levels) (gray level resolution)

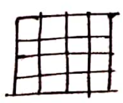
$k = 1;\ L = 2$

$k = 2;\ L = 4$

$k = 3;\ L = 8$

$k = 4;\ L = 16$

spatial resolution : $M \times N$



$k = 4$

$L = 16$

image size = 4×4 pixel $[M \times N]$

image bit size = $4 \times 4 \times 4$ bits $[M \times N \times k]$

Gray Level resolution + Spatial resolution ছাড়া image এর size বের করা অসম্ভব।

Scanned with CamScanner

Sampling বাড়ালে সবসময় ভালো result পাওয়া যায়না। কেননা এতে image volume বেড়ে যায় যা store করতে বেশি space লাগে + processing এ বেশি সময় লাগে।

Index image : ১টা color table এ index set থাকে যা matrix এ use হয়।

যদি sampling rate কমানো হয় (14×12 থেকে 8×8 নেয়া হয়) তাহলে

i) larger cell size
ii) lower spatial resolution
iii) lower features spatial accuracy
iv) lower file size, faster display and faster processing time

subsampled from 1024×1024 up to 512×512 তে alternate row column delete করা হয়েছে কিন্তু intensity একই আছে। আবার 512×512 কে 1024×1024 এ নিতে চাইলে row column duplicate করব।

বড় image থেকে ছোট image এ গেলে, আবার ঐ ছোট image থেকে বড় image এ back করলে আগের অবস্থায় ফেরা যায় না। Nearest neighbor, bilinear interpolation use করেও image ছোট করা যায়। আগের image পাওয়া যায়না বলে down sampling is an irreversible process। Spatial resolution unchanged রেখে gray level resolution কমালে আস্তে আস্তে black & white image চলে আসে। কতটুকু gray level কমাতে হবে তা depend করে image content টা কি এবং image টা নিয়ে কি করতে চাচ্ছি / image application কি।

zooming can be achieved by the following techniques:

i) nearest neighbor interpolation
ii) pixel replication
iii) bilinear interpolation
iv) bicubic interpolation

3×3 কে 6×6 এ নিতে চাইলে,

[ ] : rounding
⌊ ⌋ : floor
⌈ ⌉ : ceil

$$Ratio_{row} = \frac{3}{6} = 0.5 \qquad Ratio_{column} = \frac{3}{6} = 0.5$$

$$Row_{positions} = \frac{[1 \;\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 6]}{Ratio_{row}} = [0.5 \;\; 1 \;\; 1.5 \;\; 2 \;\; 2.5 \;\; 3]$$

↓ after round off

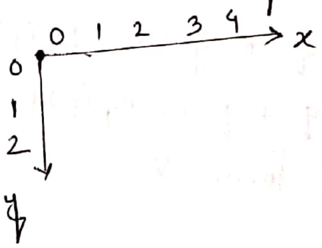
$[1 \;\; 1 \;\; 2 \;\; 2 \;\; 3 \;\; 3]$

1^{st} row এর value, 2^{nd} row এর value, 3^{rd} row এর value

same ভাবে column বের করে merge করতে হবে। RGB তে R G B ৩টা matrix নিয়ে same process করতে হয়। Shrinking self study।

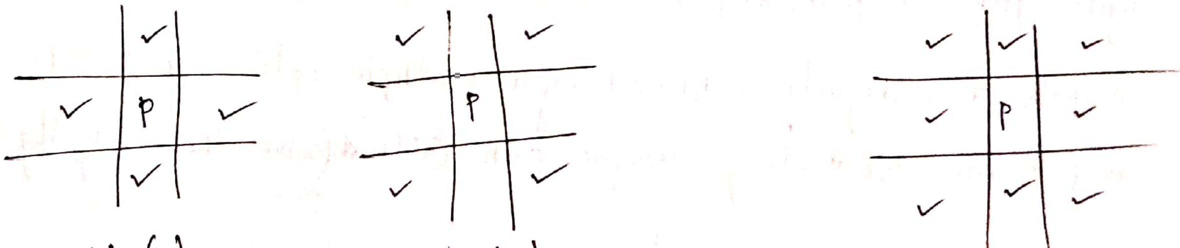
MATLAB এ origin point = (1,1) [not (0,0)]

conventional indexing method of pixels :



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

২টা pixel adjacent হলে তাকে neighbor বলে। 3 types of neighbor :
$N_4(p)$: 4 neighbor of p ; $N_D(P)$: diagonal neighbor of P ; $N_8(P)$: 8 neighbor of P



$N_4(p)$ $N_D(P)$

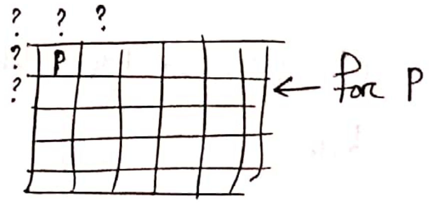
unit distance (1)

Euclidean distance

$$d = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2} = 1.414$$

Scanned with CamScanner

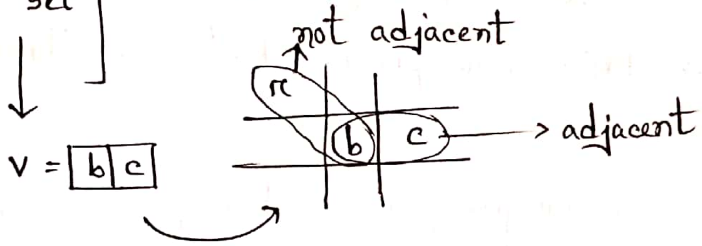
border pixel এর neighbor missing : handle এর ৩টা technique :



① discard ② zero padding ③ pixel replication

২টা pixel adjacent মানে তারা অবশ্যই neighbor এবং তাদের gray level value same set of values থেকে আসে। মানে তাদের gray level intensity value এর সাথে connection আছে + neighboring এর সাথে communication আছে। 3 types of adjacency : 4, 8, m

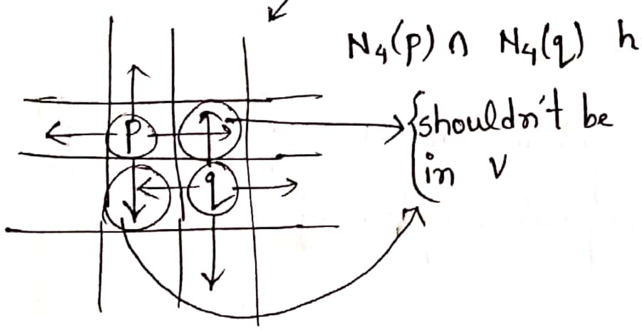
যখন $\left[\begin{matrix} N_4(p) \\ N_D(p) \\ N_8(p) \end{matrix} + V \text{ set} \right]$ একসাথে consider করি তখন এটা adjacency



adjacent হলে connected, adjacent না হলে connected না।

m-adjacency : ① V set এর মধ্যে থাকবে p, q + q in $N_4(p)$

② V set এর মধ্যে থাকবে p, q + q in $N_D(p)$ + $N_4(p) \cap N_4(q)$ has no pixel from V



digital path : 4-path, 8-path, m-path

8-path এ ambiguity তৈরি হয় মানে multiple option create হয়।

m-path use করলে always ~~unique~~ unit path দেয়। (solves the ambiguity)

connectivity : chapter (02) : 1st video : 46 min

connected region

disjoint region

Distance metrics : For pixel p, q, z : $\begin{cases} D(p,q) \geq 0 \\ D(p,q) = D(q,p) \\ D(p,z) \leq D(p,q) + D(q+z) \end{cases}$

① City Block Distance - $D_4(p,q) = |(x-s)| + |y-t|$
② Chess Board Distance - $D_8(p,q) = \max(|(x-s)|, |(y-t)|)$
③ Euclidean Distance - $D_E(p,q) = ((x-s)^2 + (y-t)^2)^{1/2}$

$p(x,y)$, $q(s,t)$

neighborhood based arithmetic/logic operation :

p = sum of product $= w_1a + w_2b + w_3c + w_4d + w_5e + w_6f + w_7g + w_8h + w_9i$

$= \sum_i w_i f_i$

= e point এ বসবে মানে center এ

f

	a	b	c		
	d	e	f		
	g	h	i		

w

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

neighborhood processing : ① smoothing/averaging
② noise removal/filtering
③ edge detection
④ contrast enhancement

কোন dark ছবিকে process করে bright করাই enhancement। or brightness কমানো for specific application is also enhancement।

Process or manipulate an image so that the result is more suitable than the original image for a specific application is called image enhancement. 3 types : spatial domain, frequency domain, combination method. why enhancement? ans : highlight interesting detail, remove noise, making images more visually appealing

spatial domain process : $g(x,y) = T[f(x,y)]$ ← input

output → $g(x,y)$

T → some operator defined over some neighborhood of (x,y)

প্রতিটা pixel individually process করা → point to point operation 

neighbor নিয়ে operation এর পর ঐ specific pixel এর জন্য output বের করা } → local operation 

whole image এর উপর কাজ করে একটা specific pixel এর output বের করা } → global operation 

$s = T(r)$ } r input image এ T transformation operation perform করে s output বের করব

standard image : $L = 2^8 = 256$ (0-255)

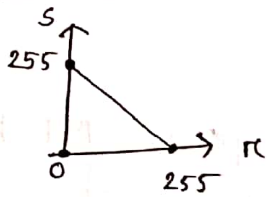
T = gray level/intensity transformation/mapping function

00000000 = 0
⋮
11111111 = 255 } এজন্য 0-255 not 1-256 } → gray level range $[0 : L-1]$

<u>Linear function</u> : negative transformation :

$r = 0$ হলে $s = 255$

$r = 255$ হলে $s = 0$

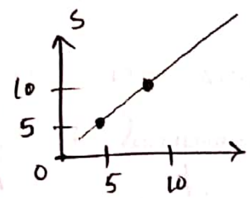


$s = \underbrace{(L-1)}_{\text{max intensity}} - r$

identity transformation :

$r = 5$ হলে $s = 5$

$r = 10$ হলে $s = 10$



$s = r$

input intensity ↑ output intensity ↓
input intensity ↓ output intensity ↑ } negative transformation

শুধুমাত্র 0/1 দিয়ে represent করলে binary transformation

intensity scaling : $s = T(r) = a(r)$

a → কোন কিছু দিয়ে each pixel এর value increase করা

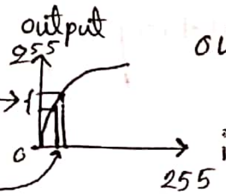
contrast enhance $\rightarrow$ thresholding : $s = T(r)$

$$s = \begin{cases} 1 & r > \text{threshold} \\ 0 & r \leq \text{threshold} \end{cases}$$

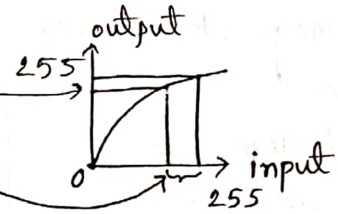
<u>Logarithmic Function</u>: Log transformation : $s = c \times \log(1+r)$

s → output, c → constant, r → input

output এ expanded range
input এ lower amplitude



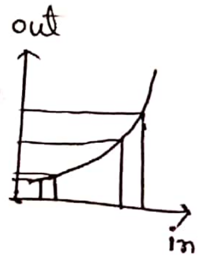
output compressed
higher amplitude of input



dynamic range লাগলে log transform করতে পারি

<u>Inverse</u> Logarithm Transformation :

lower amplitude input এ compressed output
higher amplitude input এ expanded output



Log function কে change করে বিভিন্ন দিকে নিতে চাইলে,

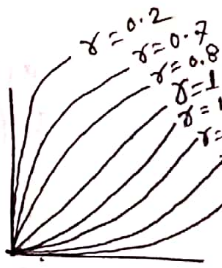
<u>Power Law function</u> : Gamma transformation

$\gamma > 1$ = inverse log এর মতো

$\gamma < 1$ = log এর মতো

$\gamma = 1$ = Linear

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



different size device এ same image দেখি gamma transformation এর মাধ্যমে। Gamma correction এ যদি inverse log = 1.09 আগেই দেই, then display এর log = 0.9 এর সাথে কাটাকাটি হয়ে $\gamma = 1$ হবে। ফলে same image টাই দেখাব।

Piecewise Linear Transformation function :

advantage : অন্য কিছু image আছে যা baki transformation process এ possible না but piecewise এ possible

disadvantage : user input বেশি লাগে যেমন piece দিয়ে কোন ২টা point বোঝানো হচ্ছে। কয়টা piece করা লাগবে

3 types : contrast stretching, grey/intensity level slicing, bit plane slicing

contrast stretching

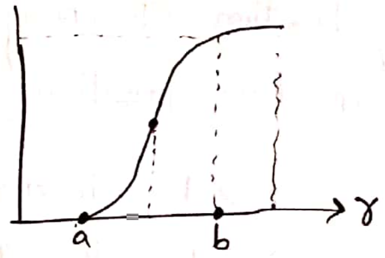
low contrast image কে enhance করে

poor illumination + wrong setting of lens aperture during image acquisition } reason of low contrast

m set করে m থেকে ছোট হলে pixel কে dark করবে

m set করে m থেকে বড় হলে pixel কে bright করবে

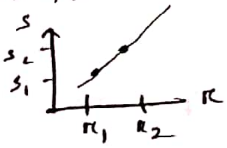
darkness আর brightness এর difference যত বেশি, contrast তত high or better



$a(r_1, s_1)$, $b(r_2, s_2)$ } piecewise linear function

point ২টা দিয়ে কতটুকু dark কতটুকু bright করব তা control হয়

if $r_1 = s_1$ & $r_2 = s_2$ then



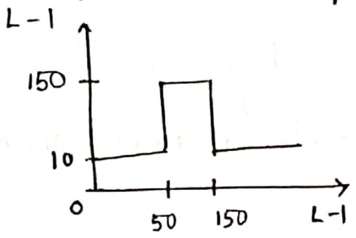
} linear curve

$$s = T(r) = \frac{1}{1 + \left(\frac{m}{r}\right)^E}$$

} E controls the slope of function

$r_1 = r_2$ & $s_1 = 0$ & $s_2 = L-1$ then acts like threshold (binary)

Grey Level Slicing / Intensity Level Slicing



50	100
200	0

in

→

150	150
10	10

out

} slicing এর বাইরের অংশকে threshold করা হয়েছে। (reduces others to lower level except A to B)

50	100
200	0

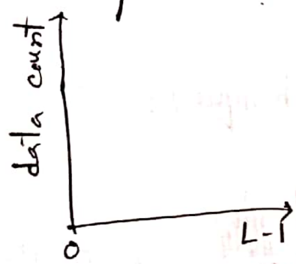
→

150	150
200	0

} acts as identical slice কে আগের মত রাখা হয়েছে। (preserves all levels except A to B)

Histogram processing:

কয়টা pixel আছে ঐ level এ



intensity level representation

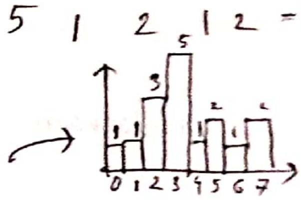
} image histogram 2 types:
1. intensity
2. individual color channel (3 histograms: R, G, B)

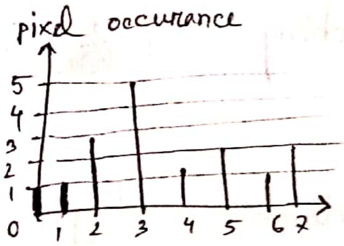
3 bit 4x4 image
↓
$2^3 = 8 \rightarrow$ intensity level

5	5	3	3
4	3	3	3
0	6	7	2
1	7	2	2

Bin pos : 0 1 2 3 4 5 6 7

h : 1 1 3 5 1 2 1 2 = 16





$h(r_k) = n_k$

k তম no. এর pixel কয়টা = n

$h(0) = 1$ $h(4) = 1$

$h(1) = 1$ $h(5) = 2$

$h(2) = 3$ $h(6) = 1$

$h(3) = 5$ $h(7) = 2$

can't construct an image from histogram because we don't know the location of pixels

কোন location এ pixel distributed তা বলা যায় না। image এ dark pixel বেশি না কম এসব বলা যাবে।

different images but } histogram same

Histogram of :

dark image : 

concentrated on low side

bright image : 

concentrated on high side

Low contrast image : 

narrow and centered toward the middle

High contrast image : 

broad range

image এর histogram change করলে image টাও change হয়ে যাবে।

<u>Histogram processing</u> : normalization $p(r_k) = n_k / n$

$\therefore p(1) = \frac{2}{8}$

$p(0) = \frac{1}{8}$

$p(2) = \frac{1}{8}$

$p(3) = \frac{4}{8}$

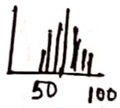
$\Big\} \; \underline{1}$

sum of all components of a normalized histogram $= \underline{1}$

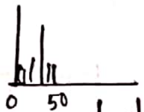
Transformation function (processing technique) :

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

histogram sliding :

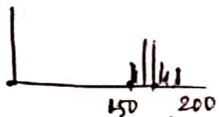


50 কে 0 + 100 কে 50

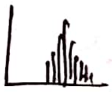


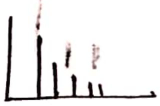
darker

50 কে 150 + 100 কে 200



brighter

histogram stretching :  $\rightarrow$  : $g = \frac{255\,(f - MIN)}{MAX - MIN}$

histogram equalization :  

better

stretching এ dynamic range

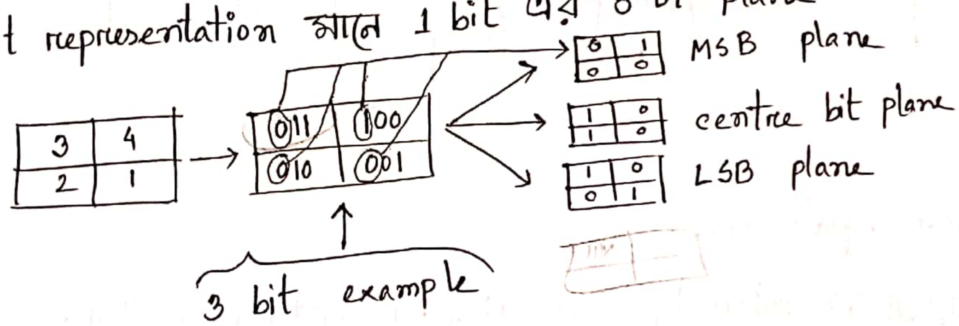
equalization এ dynamic range + equal number of pixel for various intensity level

Bit plane slicing:

একটা image হচ্ছে একটা matrix। matrix এর সবচেয়ে ছোট box হলো pixel। pixel value গুলা এক একটা digital number যেটায় intensity value থাকে। যদি 8 bit image হয় তাহলে তাকে 00000000-11111111 দ্বারা represent করা হয়। এখানে প্রতিটা bit এর contribution different যাকে highlight করা হয়।

11111111
MSB LSB

8 bit representation মানে 1 bit এর 8 টা plane।



01000100

0100010$\boxed{0}$: যদি 0 নিই then ১টা bit এর info মাত্র
010001$\boxed{00}$: " 00 " " ২টা " " " "
01000$\boxed{100}$: " 100 " " ৩টা " " " "
0100$\boxed{0100}$: " 0100 " " ৪টা " " " "
010$\boxed{00100}$: " 00100 " " ৫টা " " " "
01$\boxed{000100}$: " 000100 " " ৬টা " " " "
0$\boxed{1000100}$: " 1000100 " " ৭টা " " " "
$\boxed{01000100}$: " 01000100 " " ৮টা " " "

মানে lower bit smaller info carry করে। যদি MSB নেই তাহলে whole info carry করা possible। MSB নিলে 8 bit এর size কমে যাবে। last 7 bit বাদ হয়ে যায়।

bit plane slicing হচ্ছে image কে different plane এ ভাগ করব। Then high order এর ২/৩/৪ টা plane combine করে একটা image create করি তাহলে size অনেক কমে যায়। image compression + enhancement ২টা ক্ষেত্রেই কাজ করে।

Image reconstruction using n bit planes:

n^{th} plane in the pixels are multiplied by constant 2^{n-1}.

if $n=8$, $2^{8-1} = 2^7 = 128$ $\quad \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 128 & 128 \\ 128 & 0 \end{bmatrix}$

if $n=7$, $2^{7-1} = 2^6 = 64$ $\quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 64 \\ 64 & 0 \end{bmatrix}$

(+)

using 7^{th} & 8^{th} plane $\longrightarrow$ reconstructed image. $\longrightarrow \begin{bmatrix} 128 & 192 \\ 192 & 0 \end{bmatrix}$

Histogram equalization:

Image এর ~~কী~~ যত intensity আছে সবগুলোর সংখ্যা যেন সমান হয়। এটাই equalization এর motive। Digital image = discrete image। Coordinate + intensity value = discrete। Continuous intensity থাকলে perfect equalization possible হতো। তাই real এ approximate histogram equalization পাওয়া যায় যেটা perfect না।

algorithm: input image নিতে হবে যার L সংখ্যক level/color থাকবে। image কে 3 bit দিয়ে represent করলে $L = 2^3 = 8$ এমন।

① histogram $h(r_k)$ or normalized histogram $p(r_k)$ calculation (PMF/PDF)

PMF = $\underbrace{\text{Probability mass function}}_{\text{for discrete}}$; PDF = $\underbrace{\text{Probability distributed function}}_{\text{for continuous}}$

② CDF of $h(r_k)$ or $p(r_k)$ calculation (CDF_{hf}); CDF = Cumulative distribution function

③ If $p(r_k)$ is taken, $s = T(r) = (L-1)\sum_{q=0}^{r} h(q) = (L-1)\, CDF_{hf}(r)$

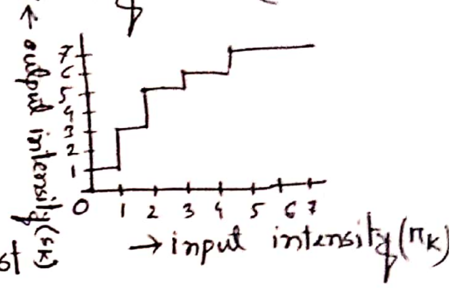
If $h(r_k)$ is taken, $s = T(r) = (L-1)\sum_{q=0}^{r} h(q) = \dfrac{(L-1)\, CDF_{hf}(r)}{M \times N}$

$\Big\}$ CDF normalized

④ apply transformation on each pixel of the input

PMF = কোন pixel value এর image এ থাকার probability কত সেটা বুঝাতে help করে

histogram equalization transformation function :



histogram equalization শুধুমাত্র image এর contrast enhance করে তা না, intensity level কেও dynamic করে, expand করে, range কে বাড়ায়।

output image কে specify করে দিই আগে থেকে এবং histogram equalization এর পর ঐ output image পেতে চাই সেটা সম্ভব না। সেজন্য equalization এর সাথে সাথে histogram specification/matching ও করা লাগে।

desired output image এর equalization করাই histogram specification/matching

histogram specification algorithm :

1. equalization of input image (equalize the histogram of input image)
2. equalization of output image (equalize the specified histogram)
3. map two of the images / relate the two equalized histograms