14/03/2023

Page No.
Date

* Image Morphology :-

⇒ Dialation :-

Let $A(x,y)$ be a binary image, B as structuring element, then the resultant element $C(x,y)$ i.e. O/P image.

$$C(x,y) = Max\{A(x,y) * B\}$$

$$C = A \oplus B$$

A | 1 0 0 0 1 0 0 0 |

1	1	1

B

C | 1 0 1 |

A | 16 17 16 18 |

1	1	1

B

C | • 17 18 |

⇒ Erosion :-

$$C = A \ominus B$$

$$C(x,y) = Min\{A(x,y) * B\}$$

Q. Consider 10x10 Image. Perform dialation using structuring element

$$B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = A \oplus B = \max\{A(x,y) * B\}.$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

1's boundary yes, no. of zeros (black part) yes.

last row & column :- copy from i/p image or keep it empty.

* Dialation adds pixels to the boundaries and it helps in enlarging or expanding the boundary.
* Size of black box at the centre has reduced and white boundary has increased in size.

Q. Perform Erosion on 10x10 image with same structuring element i.e. $B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ and same image (i/p).

=> Erosion :- $C = A \ominus B$

$$c(x,y) = \min\{A(x,y) * B\}.$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

* Size of white boundary has decreased. Black box or gap increased in the size.
So, Erosion shrinks the boundaries and enlarges the gaps.

21/03/2023

Page No.
Date

* Opening :-

→ It is erosion followed by dilation.

→ Consider 'A' as image and 'B' as structuring element.

* $Open(A,B) = A \circ B = [A \ominus B] \oplus B$

* Closing :-

→ It is dialation followed by erosion.

* $Close(A,B) = A \bullet B = [A \oplus B] \ominus B$

Q. Perform opening on given image.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Page No.
Date

$A \ominus B =$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	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	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	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

copy last row from i/p image

$[A \ominus B] \oplus B =$

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

* Original image had two white blocks which were connected by thin white strip.
* Opening of this image got rid of this strip.
* Size of white blocks remains unchanged.
* Opening breaks down narrow ridges or isolates objects which are just touching one another.

Q. Perform closing operation.

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

The marked zeroes are creating gap in block of 1's.

$$B = \begin{matrix} 1 \\ 1 \end{matrix}$$

* $$A \oplus B = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Page No.
Date

$[A \oplus B] \ominus B =$

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

Gaps are filled here by '1's.

* Closing operation tends to fuse narrow breaks.
* Boundary extraction can be achieved by $A - \underbrace{(A \ominus B)}_{\text{erosion}}$.

☆ Hit Or Miss Transformation :-

→ Small odd size mask (structuring element) normally 3X3 is scanned over entire image (binary).

→ If binary pattern of structuring element matches the pixels of an image then it is HIT condition. In this case, output pixel in spatial correspondence the centre pixel is set to desired binary state normally '1'.

→ If binary pattern does not match with the pixels, it is MISS condition. Output pixel is set to opposite binary state normally '0'.

Page No.
Date

Q. Given 10x10 image, Apply Hit or Miss Transformation.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B1 = \begin{matrix} X & 1 & X \\ 0 & 1 & 1 \\ 0 & 0 & X \end{matrix}$$

First output I_1 :-

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

$$B_2 : \begin{bmatrix} X & 1 & X \\ 1 & 1 & 0 \\ X & 0 & 0 \end{bmatrix}, \quad B_3 = \begin{bmatrix} X & 0 & 0 \\ 1 & 1 & 0 \\ X & 1 & X \end{bmatrix}, \quad B_4 = \begin{matrix} 0 & 0 & X \\ 0 & 1 & 1 \\ X & 1 & X \end{matrix}$$

Page No.
Date

$$I_2 = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$$I_3 = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$$I_4 = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$I = I_1 + I_2 + I_3 + I_4 \Rightarrow$ ORing.

$$I = \begin{array}{c|cccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

* Another Method.

(I/P image) $\xrightarrow{\text{apply}}$ B1 $\Rightarrow$ ans 1

$(\text{I/P image})^c \xrightarrow{\text{apply}}$ B2 $\Rightarrow$ ans 2

Take AND of ans1 and ans2.

For given 7x7 image, using Hit-or-Miss find an edge.

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

$$B1 = \begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline \end{array} \qquad B2 = \begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline \end{array}$$

* $I_1 = A \cdot B1$

$I_2 = A^c \cdot B2$

$R = I_1 + I_2$.

Page No.
Date

$$I_1 = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$$\cancel{B_2} = A^c = \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

$$B_2 = \begin{matrix} 1 \\ 0 \end{matrix}$$

$$I_2 = \begin{matrix} 0 & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

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

* Image Restoration

Process of removal or reduction of degradation in image. Degradation usually during acquistion of image itself.

Mean Filters : Simple yet efficient filters

- Arithmetic Mean Filter
  - Same as averaging mask
  - Computes avg. value of neighbourhood of corrupted image $g(x,y)$, places average value at center.

$$h_R(x,y) = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\tilde{f}(x,y) = g(x,y) * h_R(x,y)$$

or

$$\tilde{f}(x,y) = \frac{1}{MN} \sum_{m,n} g(m,n)$$: MN = Total pixels

- Geometric Mean Filter
  - Center pixel is replaced by product of pixels within neighbourhood, raised to power $1/MN$

$$\tilde{f}(x,y) = \left[\prod_{(m,n)} g(m,n) \right]^{1/MN}$$

- Achieves smoothing as well as tends to lose less image details.

- Harmonic Mean Filter:

$$\tilde{F}(x,y) = \frac{M \times N}{\sum_{m,n} \frac{1}{g(m,n)}}$$

- Works well with gaussian noise and salt noise (Not pepper noise)

- Order Stastistics Filters : (like Median)
  - Apart from that max-min filter
  - 2 seperate filters work within neighbourhood

Max Filter : $\tilde{F}(x,y) = \max_{(m,n)} \{g(m,n)\}$

Min Filter : $\tilde{F}(x,y) = \min_{(m,n)} \{g(m,n)\}$

- These filters helps to identify brighest, darkest points within image.

Q) For given image, what would be centre pixels change to when image is passed through:
(i) AMF (ii) GMF (iii) HMF (iv) Max (v) Min Filters

1	7	5
6	2	3
1	4	2

Date: __/__/__

→ Image:

1	7	5
6	2	3
1	4	2

(i) AMF:

$$\tilde{F}(x,y) = \frac{1}{M \times N}\left[\sum_{(m,n)} g(m,n)\right]$$

$$= \frac{1}{9}\left[1+7+5+6+2+3+1+4+2\right]$$

$$\tilde{F}(2,2) = 3.44 \approx 3$$

(ii) GMF

$$\tilde{F}(x,y) = \left[\prod_{(m,n)} g(m,n)\right]^{1/MN} = \left[1\times7\times5\times6\times2\times3\times1\times4\times2\right]^{1/9}$$

$$\tilde{F}(2,2) = 2.78 \approx 3$$

(iii) HMF

$$\tilde{F}(x,y) = \frac{M \times N}{\sum_{(m,n)} \frac{1}{g(m,n)}} = \frac{9}{\left[1+1/7+1/5+1/6+1/2+1/3+1+1/4+1/2\right]}$$

$$\therefore \tilde{F}(2,2) = 2.19 \approx 2$$

(iv) Max Filter

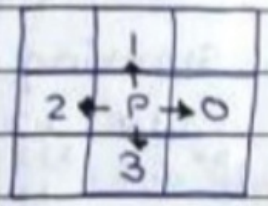
$$\tilde{F}(x,y) = \max_{(m,n)} \{g(m,n)\} = 7$$

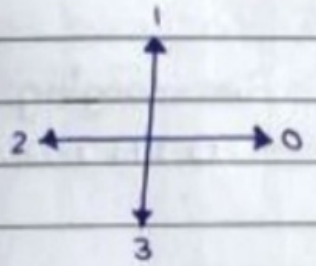
(v) Min Filter

$$\tilde{F}(x,y) = \min_{(m,n)} \{g(m,n)\} = 1$$

* Chain Codes.

- Chain code works best with binary image and is concise way of representing a shape contour.
- Chain code direction convection is.

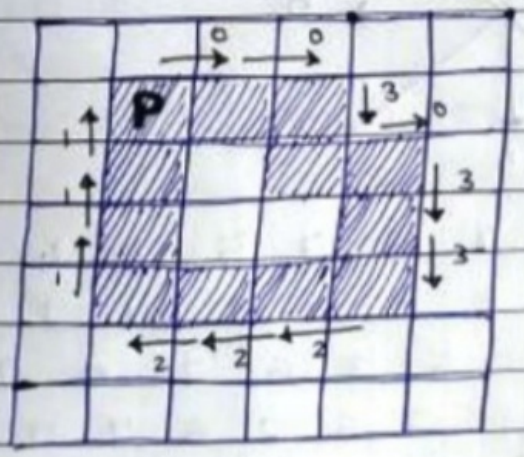




4-chain codes

8-chain codes

Q) For given image, write 4-chain code for the same.

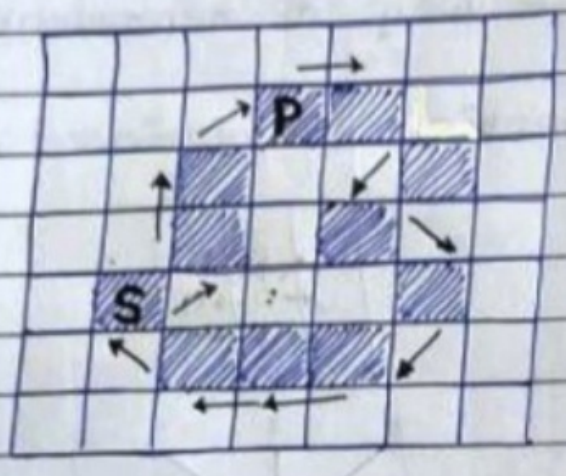


Code: 0 0 3 0 3 3 2 2 2 1 1 1

P : Starting point

Instead of storing co-ordinates of each edge pixel, we can store chain code and starting coordinate.

Q) Given an image, write down 8 chain codes



Starting P:

Code : 0 7 5 7 5 4 4 3 2 2 1

1st diff : 1 0 7 5 7 5 4 4 3 1 2 1

7 7 6 2 6 7 0 7 1 1 7

Starting S:

Code : 1 2 1 0 7 5 7 5 4 4 3

1st diff : 3 1 2 1 0 7 5 7 5 4 4 3

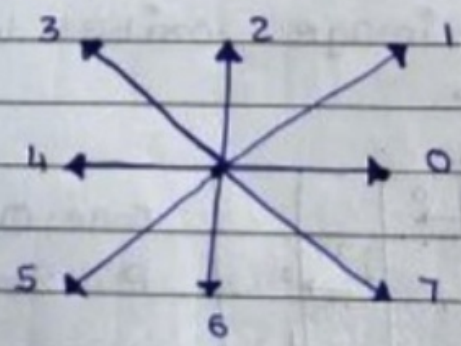
1 1 7 7 7 6 2 6 7 0 7

Rearranging 1st diff:

0 7 1 1 7 7 7 6 2 6 7

- First difference:

It is calculated by taking 2 numbers at time and counting no. of positions required to reach 2nd number from 1st in counter-clockwise direction.



Eg: Code : 0 7 5 7 5 4 4 3 1 2 1

7 6 2 6 7 0 7 6 1 7

1st difference : 7 6 2 6 7 0 7 6 1 7

∴ This first difference of code makes it invariant to rotation.

Date: / /

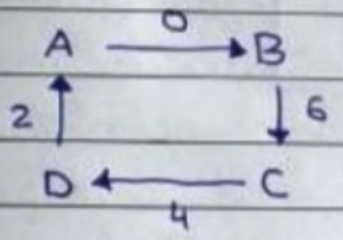
Code can be normalized w.r.t starting position and rotation.

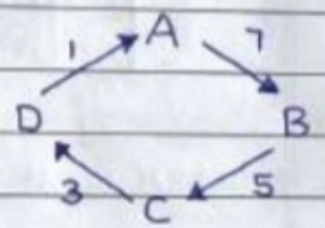
- Redefine starting point so that resulting sequence of numbers forms integer of min. magnitude. (Translation invariant)

It can be normalized for rotation by steps:

- Write chain code
- Redefine starting point: Take last position number at first.
- Find first difference
- Take smallest magnitude

Eg.





- Chaincode:

  0 6 4 2

- Performing 1st diff.

  2 0 6 4 2

  6 6 6 6

- Result:

  6 6 6 6

- Chain code:

  7 5 3 1

- Performing 1st diff.

  1 7 5 3 1

  6 6 6 6

- Result:

  6 6 6 6

06/04/2023

Page No.
Date

☆ Image Segmentation :-

Discontinuities.

Similarities

1). ~~Based on Discontinuities :-~~

* Image Segmentation segments image into small regions or objects. It is based on similarities and discontinuities. In similarities, we partition image into regions having similar areas based on predefined condition.

→ In image segmentation using discontinuities, we divide image based on abrupt changes in intensity.

* Image Segmentation based on Discontinuities :-

→ point, line, edges are considered as three basic discontinuities, observed in image.
Simple Method is to get spatial mask having addition of all the coefficients equal to zero as point, line, edges are high frequency component.

→ Point can be detected using simple mask.

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

Page No.
Date

* Line Detection :-

Image =

0	0	0	10	0	0	0	0
0	0	0	10	0	0	0	0
0	0	0	10	0	0	0	0
0	0	0	10	0	0	0	0
0	10	10	10	10	10	10	0
0	0	0	10	0	0	0	0
0	0	0	10	0	0	0	0
0	0	0	10	0	0	0	0

* Result 1 (R1) = Image * Mask 1

Mask 1 =

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

R1 =

copy from i/p image

0	0	0	10	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	-20	-20	-20	-20	-30	-20	0
0	40	40	40	40	60	40	0
0	-20	-20	-20	-20	-30	-20	0
0	0	0	0	0	0	0	0
0	0	0	10	0	0	0	0

R2 = Image * Mask 2

mask 2 =

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

R2 =

0	-30	-60	-30	0	0	
0					0	
10					10	
10					10	
10					10	
0	-30	-60	-30	0	0	

* Image has two lines one horizontal and one vertical. By applying mask 1 i.e. horizontal mask, we observe in output that it detects lines only in horizontal direction and cleans up vertical lines.

* To detect vertical lines, mask 2 can be used. ~~and suppose~~

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

mask 3

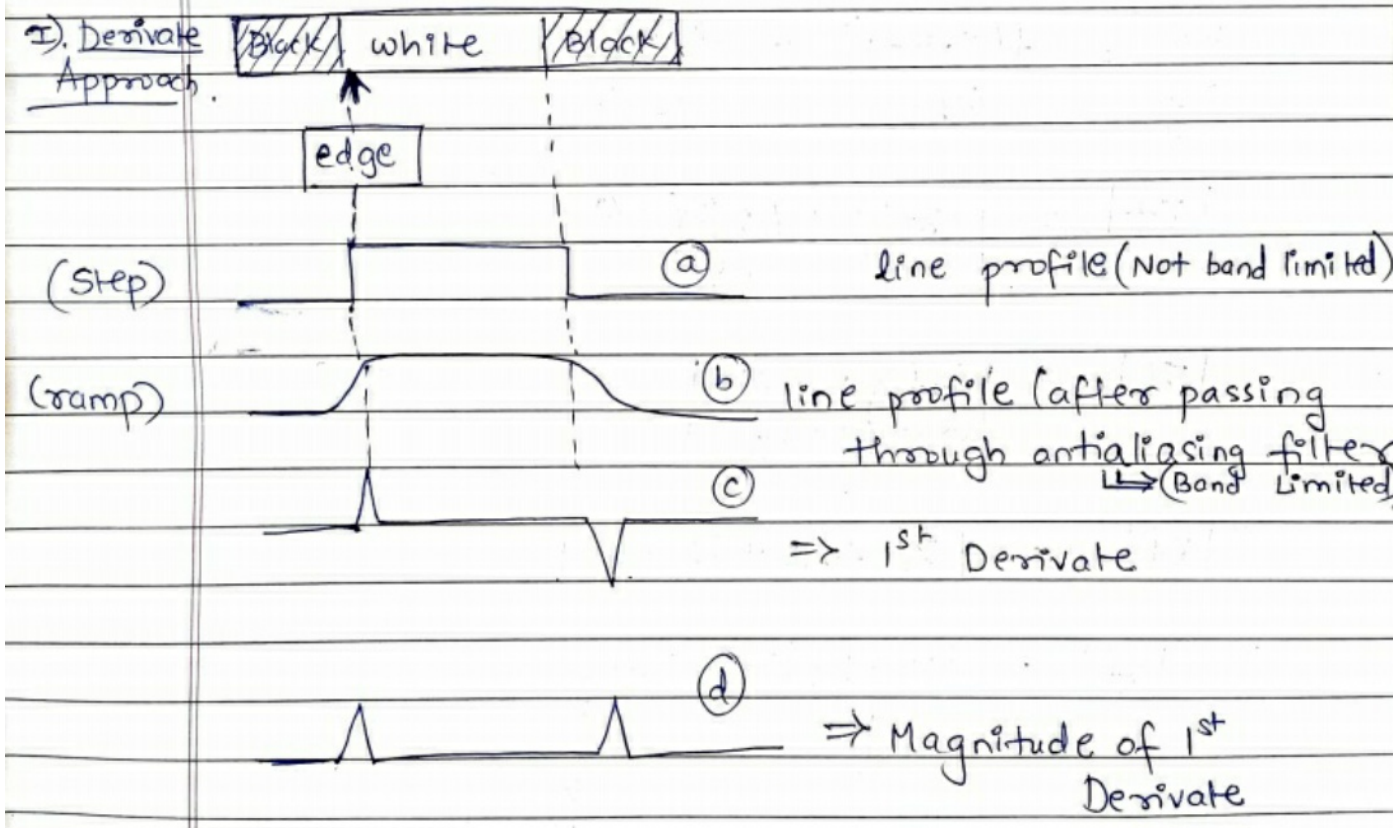
-1	-1	2
-1	2	-1
2	-1	-1

mask 4

⇒ Mask 3 and 4 can be used to detect diagonal lines.

* Edge Detection :-

I). Derivate Approach.



* Figure (c), shows I derivate of line profile which is positive at leading edge and negative at trailing edge and zero for rest of the areas of gray levels.
* We get spike only at edge and hence it helps to detect edges.

* **Finding Gradient :-**

	$y-1$	y	$y+1$
$x-1$	z_1	z_2	z_3
x	z_4	z_5	z_6
$x+1$	z_7	z_8	z_9

$$\frac{\partial f}{\partial x} = \underset{(z_8)}{f(x+1, y)} - \underset{(z_5)}{f(x, y)}$$

$$\frac{\partial f}{\partial y} = \underset{(z_6)}{f(x, y+1)} - \underset{(z_5)}{f(x, y)}$$

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

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

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

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

Mask 1 →

1	0
-1	0

1	-1
0	0

→ Mask 2

Ordinary Mask

Page No.
Date

* Steps :-

1). Image * mask 1 = R1
2). R2 = Image * mask 2
3). R1 + R2 = (R)

OR.

1). Add both the mask to get resultant mask.
2). Apply Resultant mask on Image. i.e. Image * Mask.

* Robert's ~~L.G.~~ Mask :

Robert L.G.
- It claimed that better results could be obtained if we take cross difference instead of straight differences. &
- Robert Mask can be written as

$|z_5 - z_9| \rightarrow \begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$

$|z_6 - z_8| \rightarrow \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$

Sum of both the mask $\rightarrow \begin{matrix} 1 & 1 \\ -1 & -1 \end{matrix}$

* I derivate operator → Robert, Ordinary, Prewitt, Sobel.

* Prewitt

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

Fx $\qquad$ Fy

$F_x \rightarrow z_7 + z_8 + z_9 \rightarrow -(z_1 + z_2 + z_3)$

$F_y \rightarrow z_3 + z_7 + z_9 \rightarrow -(z_1 + z_4 + z_7)$

$F_x + F_y \rightarrow$

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

* **Sobel :-**

-1	2	-1
0	0	0
1	2	1

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

$F_x + F_y \rightarrow$

-2	2	0
-2	0	2
0	2	2

* Prewitt & Sobel Mask gives better result even if image is noisy.
- Derivate filters are sensitive to noise.
- Prewitt and Sobel are high pass filter but advantages of these 2 filters ~~are~~ of edge detection is due to the operator providing smoothing effect along with differentiation.
- They perform smoothing in one direction using low pass component and edge detection in perpendicular direction using high pass component.
- Hence, they perform well even on noisy images.

F_x

-1
0
1

HP Mask

1	1	1

LP Mask

-1	0	1

HP Mask

F_y.

1
1
1

LP Mask

Page No.
Date

* Image segmentation using 2nd Derivate (Laplacian)

- 2nd Derivate is given by -

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

$$\frac{\partial^2 f}{\partial x^2} = \underset{(z_8)}{f(x+1, y)} + \underset{(z_2)}{f(x-1, y)} - \underset{(z_5)}{2f(x, y)}$$

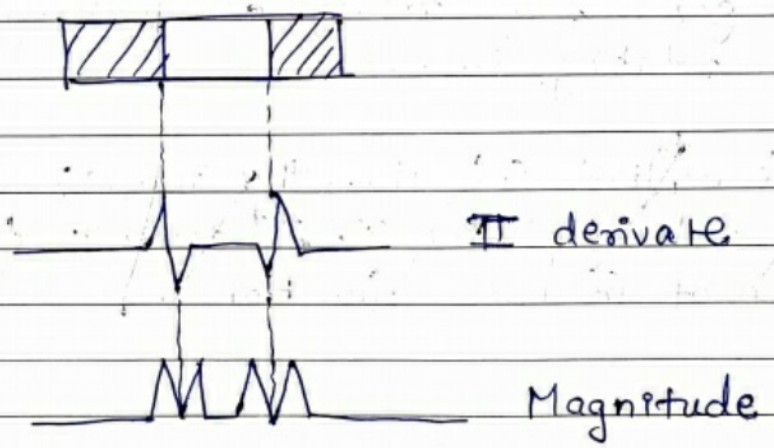
$$\frac{\partial^2 f}{\partial y^2} = \underset{(z_6)}{f(x, y+1)} + \underset{(z_4)}{f(x, y-1)} - \underset{(z_5)}{2f(x, y)}$$

- Laplacian Operator

0	1	0
1	-4	1
0	1	0

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

*



Q. Can we apply 2nd order derivate filter direct on image?

- Laplacian is not directly applied in its original form as an edge detector because:-

1). Since, Laplacian is 2nd Derivate filter, it is sensitive to noise much more than compared to 1st derivate if noise.

is present in an image, Laplacian gives very high
- ~~Lapt~~ value & ruins entire image.
2). Magnitude of Laplacian ~~produces~~ produces double edges which is undesirable effects.

* Some kind of noise cleaning is done prior to application of Laplacian operator so as to get better results.
Smoothing can be done by preceeding smoothing operator like Gaussian smoothing. The resultant algorithm is known as Laplacian of Gaussian (LoG).

* Advantages of Secondary Derivate Filter :-
- Unlike Sobel and Prewitt, they are isotropic filters i.e. their responses are independent of direction of discontinuities. ~~These~~ isotropic filters are rotation invariant i.e. rotating image and then applying mask gives same results as applying mask and then rotating image.

* ~~We have to first smoothen the image.~~
- ~~Then take Laplacian i.e. 2nd derivate to combine action of Gaussian function.~~
- And

11/04/2023

Page No.
Date

* Laplacian of Gaussian (LOG) :-

→ We have to first smoothen the image and then take Laplacian i.e. 2nd derivate to combine action of Gaussian function and 2nd derivate.

→ Consider Gaussian function :-

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

where, σ determines degree of blurring, x and y are spatial co-ordinates

→ Let $x^2+y^2 = r^2$

$$\therefore \; h(r) = e^{-(r^2/2\sigma^2)}$$

→ Taking I derivate wrt 'r' :-

$$\frac{\partial}{\partial r} h(r) = \frac{\partial}{\partial r} e^{-(r^2/2\sigma^2)}$$

$$= e^{-\left(\frac{r^2}{2\sigma^2}\right)} \cdot \frac{\partial}{\partial r}\left(\frac{-r^2}{2\sigma^2}\right)$$

$$= e^{-\left(\frac{r^2}{2\sigma^2}\right)} \cdot \frac{-2r}{2\sigma^2}$$

$$= \underset{(u)}{e^{-\left(\frac{r^2}{2\sigma^2}\right)}} \cdot \underset{(v)}{\left(\frac{-r}{\sigma^2}\right)}.$$

→ Taking II derivate :-

$$\frac{\partial^2}{\partial r^2} h(r) = \frac{\partial}{\partial r}\left[\frac{-r}{\sigma^2} e^{-r^2/2\sigma^2}\right]$$

$$= \frac{-1}{\sigma^2} \frac{\partial}{\partial r} \left[r \cdot e^{-r^2/2\sigma^2} \right]$$

$$= \frac{-1}{\sigma^2} \left[r \frac{\partial}{\partial r} e^{-r^2/2\sigma^2} + e^{-r^2/2\sigma^2} \frac{\partial}{\partial r} r \right]$$

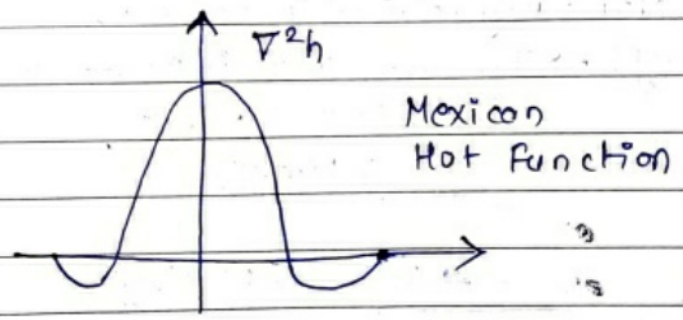
$$= -\frac{1}{\sigma^2} \left[r\, e^{-r^2/2\sigma^2} \left(\frac{-2r}{2\sigma^2} \right) + e^{-r^2/2\sigma^2} (1) \right]$$

$$= \frac{1}{\sigma^2} \left[\frac{r^2}{\sigma^2} e^{-r^2/2\sigma^2} - e^{-r^2/2\sigma^2} \right]$$

$$= \frac{1}{\sigma^2} e^{-r^2/2\sigma^2} \left[\frac{r^2}{\sigma^2} - 1 \right]$$

$$= \frac{(r^2 - \sigma^2)}{\sigma^4} e^{-r^2/2\sigma^2}$$

→ Laplacian of Gaussian ($\nabla^2 h$) can be approximated using a matrix which has large positive value at the center and negative values at peripheries as $\nabla^2 h$ function looks like Mexican Hat Function.



→ <u>Mask</u> :-

$$\begin{matrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{matrix}$$

* Edge Linking

Image → Gradient operator → Edge Linking → Good Edge Image.

⇒ Techniques for Edge Linking :-

I). Local Processing :
We analyse pixels in neighbourhood. If pixels share some common properties, they are linked together.
Two properties that we consider to link pixels are:

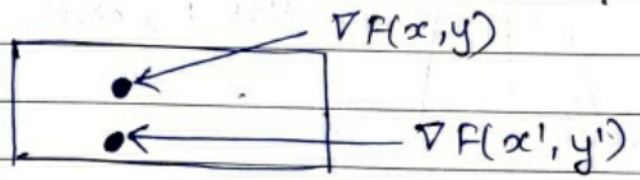
(i). Strength of the Gradient :-

$$|\nabla F| = [F_x^2 + F_y^2]^{1/2}$$

→ F_x gradient in x-direction, F_y gradient in y-direction, pixel in neighbourhood of pixel $\nabla F(x,y)$ is linked to pixel $\nabla F(x',y')$ if

$$|\nabla F(x,y) - \nabla F(x',y')| \leq T$$

T → Non-negative Threshold.
$\nabla F(x,y)$ → Image obtained after using Sobel/Prewitts Mask.



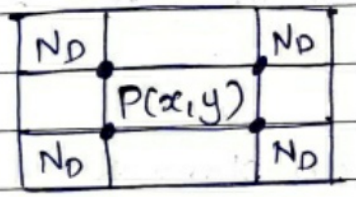
(ii). Direction of Gradient :-

$$\theta = \tan^{-1}\left[\frac{F_y}{F_x}\right]$$

$\theta(x, y)$
$\theta(x', y')$
$\theta(x, y) - \theta(x', y') \leq \theta$.

→ Only if both these conditions are satisfied, pixels are linked together. For every pixels, all its 8 neighbours are checked for these two conditions.

* Connectivity of pixels :-



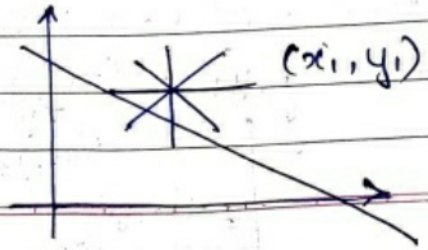
N_D :- Neighbourhood art diagonal.
→ Shares a common point.

N_4 : Four connectivity → share common side.

N_8 : Eight connectivity

* Hough Transform :-

- In set of discrete pixel, hough transform checks if these points lie on a straight line if yes, it draws a line joining all these points.



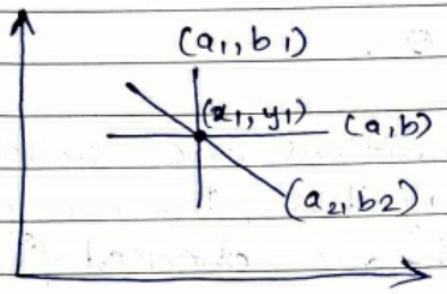
$y = mx + c$
$y_1 = mx_1 + c$
$y_1 = ax_1 + b$

- Consider point (x_1, y_1).
- Line passing through this point is,

$y_1 = mx_1 + c$ which can be written as

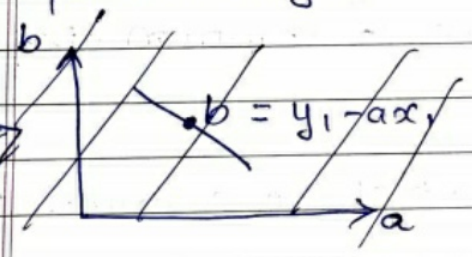
$y_1 = ax_1 + b$; considering $a \rightarrow$ slope, $b \rightarrow$ intercept.

- By varying values of a and b, we get infinite number of lines passing through x_1, y_1.



- If we write $b = y_1 - ax_1$ (1), consider ab plane instead of x, y.

We will get a single line in ab plane due to point (x_1, y_1).



Line in ab plane is due to single point (x, y) with different value points of a & b.

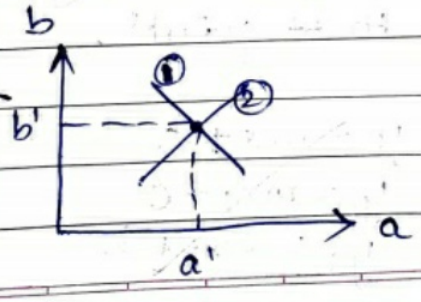
- Consider 2nd point (x_2, y_2) such that

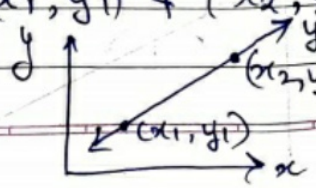
$y_2 = a x_2 + b$

$\rightarrow b = y_2 - ax_2$ —— (2)

- This line in ab plane will intersect 1st line if (x_1, y_1), (x_2, y_2) are part of same line in (x, y) plane.
- Intersect^n is given by (a', b') which will be given by equation $y = a'x + b'$.

This line will pass through point (x_1, y_1) & (x_2, y_2).



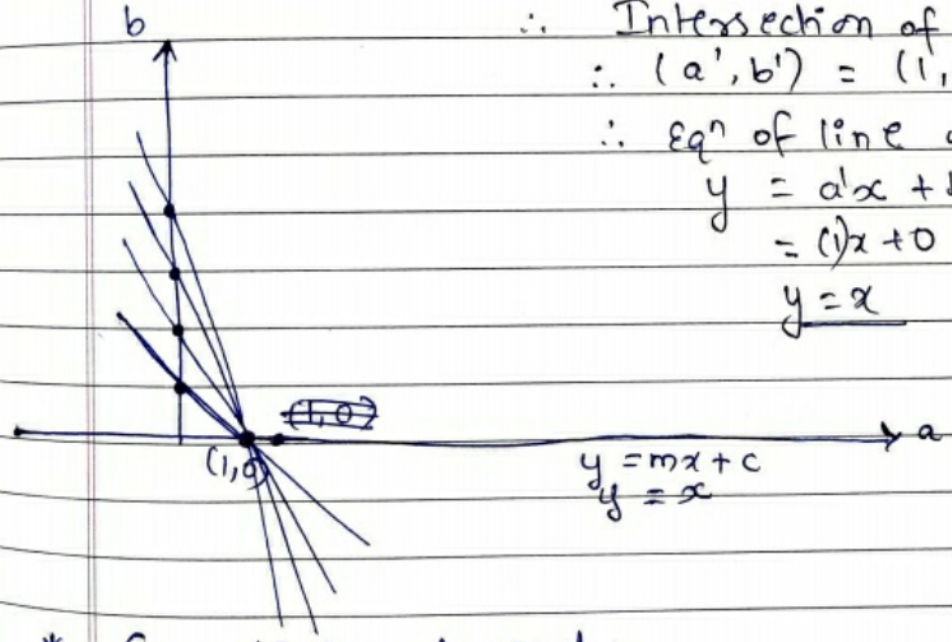


- If we have 'n' point lying on straight line, we get 'n' lines in a-b plane,
  × Each line representing each point.
- These lines would intersect at a single point (a', b') in ab plane using equation $y = a'x + b'$.
- We would get a line passing through all the points.

Q. Given 4 points in x-y plane with co-ordinates $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$.
Use Hough transform to join these points.

Given :- $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$, $y = ax + b$ $\therefore b = y - ax$

x	y	$b = y - ax$
1	1	$b = 1 - a$
2	2	$b = 2 - 2a$
3	3	$b = 3 - 3a$
4	4	$b = 4 - 4a$



$\therefore$ Intersection of all lines : $(1,0)$

$\therefore (a', b') = (1,0)$

$\therefore$ Eqn of line will be

$y = a'x + b'$

$= (1)x + 0$

$y = x$

$y = mx + c$

$y = x$

* Connectivity of pixel :-

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

$N_4(P)$ → Sides of pixels

N_D → Diagonal $[N_D(P)]$ → points at diagonal

$N_8(P) = N_4(P)$ OR $N_D(P)$

18/04/2023

Page No.
Date

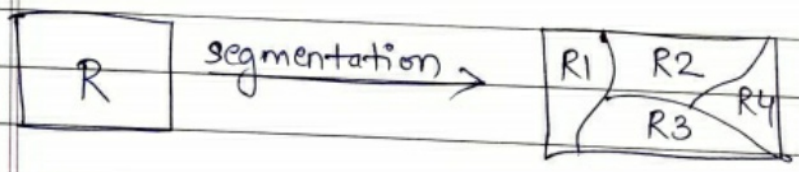
* Region Based Segmentation

→ Regions are created by grouping together pixels which share common features based on some criteria.

→ 'R' represents entire image which is divided into sub regions R1, R2, R3, etc.

→ Five conditions for region based segmentations:-

1). Addition of all the regions equals entire image.
2). Each region is connected to other either by 4 connectivity or 8 connectivity.
3). Intersection of any pair of adjacent regions should be an empty set.
4). For each region, uniformity predicate (condition) has to be true.
5). For any two adjacent regions, predicate has to be false.



1). Region Growing
2). Region Splitting
3). Region merging
4). Split and merge.

1). Region Growing:-

we as a user setting it.

Threshold.

* Condition :- $\text{Max } g(x,y) - \text{Min } g(x,y) \leq 3$ } Check region wise and not neighbour wise.

↳ Growing region should satisfy this condition.

Consider region pixel to start growing

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

→ R1

R2

*

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

→ R3.

R2 R1.

How to check whether within region condition is satisfied or not?

→ Consider max ~~of~~ min. within region.

for R1, max = 5, min = 2 ; 5 - 3 = 2

↓

condit^n satisfied.

* Disadvantage :-

- If we change the seed pixel, segmented image will be different. In this method, at a time only one pixel is assigned as a member of current region. Hence, method is slow.

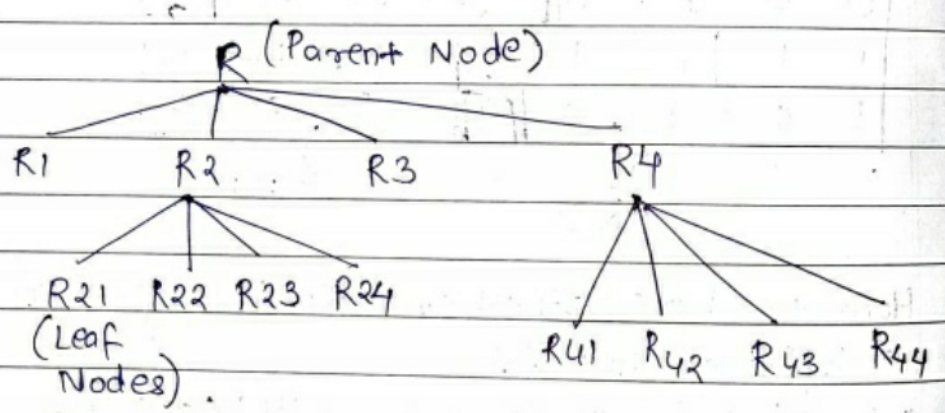
- Segmented images are different, if directions in which we try to grow the regions are different.

* Better approach can be considering small region instead of a single pixel.

* <u>Region Splitting</u> :-

Use same condition :- $\max g(x,y) - \min g(x,y) \leq 3$.

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



- ~~For~~ If gray levels in the region do not satisfy the condition, divide regions into four quadrants.
- If condition is satisfied, leave the region as it is.
- Do this recursively until all regions satisfy the condition.
- Method is applicable only to images whose no. of

rows and no. of columns are integer power of 2.

3). Region Merging.

* Use same condition.

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

- For merging, it is opposite of region splitting.
- Like ~~merging~~ splitting, this method is applicable only to images whose no. of rows and columns are integer power of '2'.
- Start from pixel level, keep on merging four adjacent regions arranged in 2H2 fashion satisfying the condition.
- Merge to form a bigger region if condition is satisfied otherwise keep regions as it is.

4). Split & Merge :-

- Region splitting starts with whole image and splits the region into four quadrants.
- Region merging starts with a pixel and merge it into larger region if condition is satisfied.

- If Homogeneous regions are small, region merging is better.
- If most of the regions in an image are homogeneous then splitting is better than merging.
- In most applications, split and merge works better.

* Steps:-

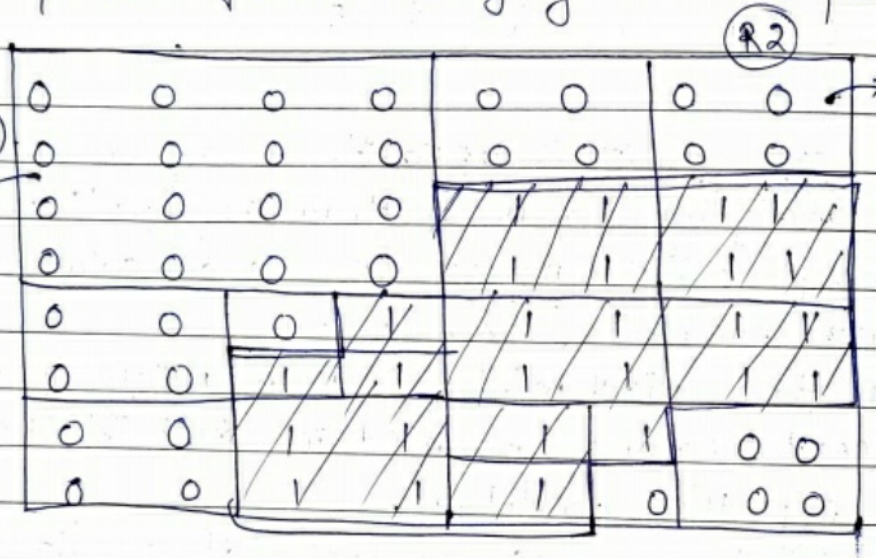
1). Split the region into four quadrants if condition is not true.

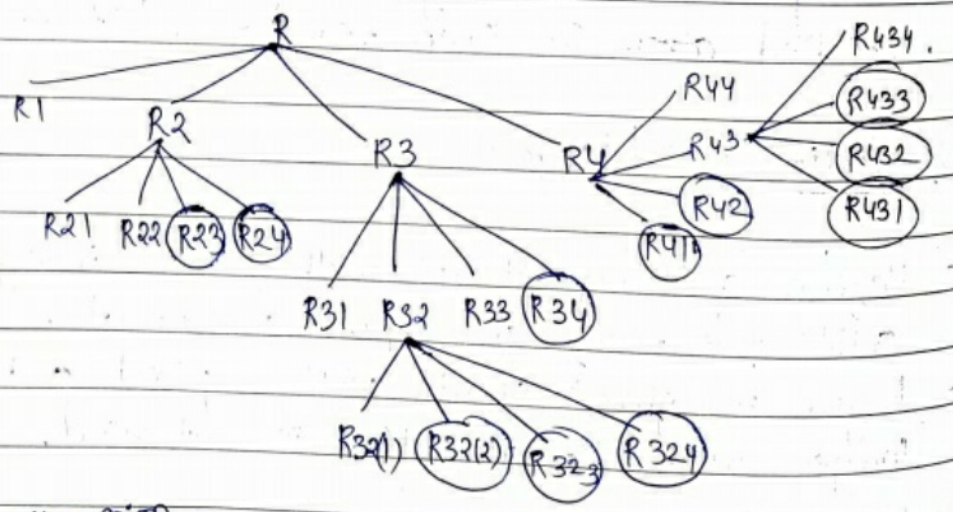
2). Continue subdivision for all new sub-images till stoping criteria is reached.

3). Merge adjacent regions for which condition is true.

4). Stop when further merging is not possible.

Q.





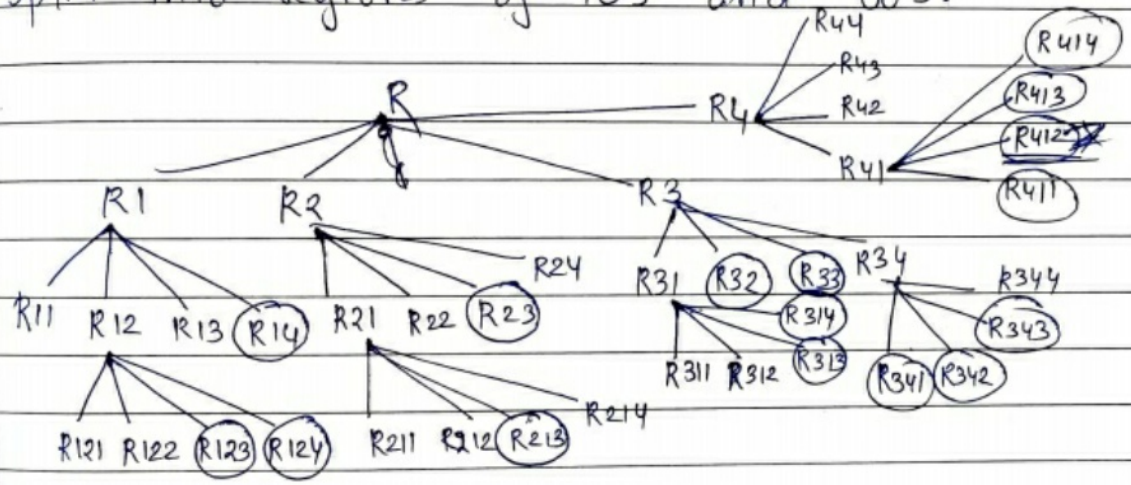
* Mark the region ~~all~~ ~~#s~~ having all 1's in tree diagram and those regions can be merged further.

Q. Using split and merge, segment the given image.

All around 10's

13	12	13	12	11	12	11	12
13	13	63	63	61	11	12	11
11	12	63	62	61	62	12	12
13	13	62	63	62	61	13	13
12	11	62	63	62	11	12	11
62	62	63	61	61	62	13	13
62	61	61	62	13	12	13	11
61	62	63	11	12	11	12	12

Split into regions of 10's and 60's.



- Mark the region around 60's.

* Thresholding

- **Otsu Method:**

Automatically find the optimal threshold intensity works by searching for threshold intensity that optimally separates image into two classes foreground and background. It does this by maximizing a metric called between class variance. This variance is given by eqn:

$$\sigma_B^2 = \omega_b \cdot \omega_f (\mu_b - \mu_f)^2$$

where,

$\omega_b = \dfrac{\text{No of pixel in bg}}{\text{Total pixels}}$

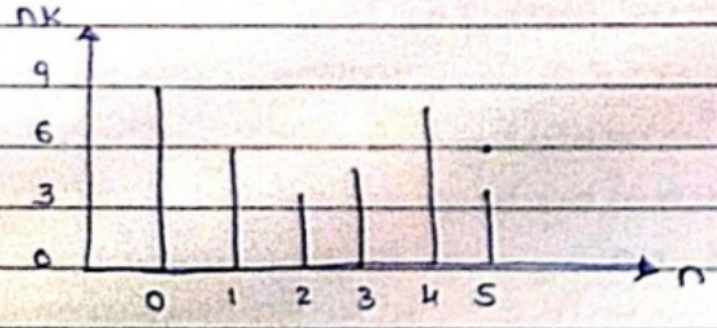
$\omega_f = \dfrac{\text{No. of pixels in fg}}{\text{Total pixels}}$

μ_b = Mean intensity of bg

μ_f = Mean intensity of fg

Eg:

0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0



- Procedure for global thresholding to obtain
  - Select initial 'T' estimate (T_0)
  - Segment image using T will produce 2 group of pixels. G_1, G_2
  - Compute avg grey value, μ_1, μ_2 for regions G1, G2
  - $T = \frac{1}{2}(\mu_1 + \mu_2)$
  - Repeat till difference in 'T' in iterations smaller than predefined parameter T_0.

Eg: $\begin{vmatrix} 5 & 3 & 9 \\ 2 & 1 & 7 \\ 3 & 4 & 2 \end{vmatrix}$ $T_0 = 4.55 \approx 5$

If $T > T_0$, consider T as T_0. Repeat calculation for G1, G2, μ_1, μ_2 and new T and check for condition.

Date: / /

Now, value < 2 : BG
value ≥ 2 : FG

$\therefore \omega_b = \frac{9+6}{36} = 0.42$, $\omega_f = \frac{21}{36} = 0.53$

$\therefore \mu_b = 0.4$, $\mu_f = 3.57$

$\therefore \sigma_B^2 = \omega_b \cdot \omega_f (\mu_b - \mu_f)^2$
$= 2.44$

Now, value < 3 : BG

$\therefore \omega_b = \frac{19}{36} = 0.53$, $\omega_f = \frac{17}{36} = 0.47$

$\therefore \mu_b = 0.74$, $\mu_f = 3.94$

$\therefore \sigma_B^2 = \omega_b \cdot \omega_f (\mu_b - \mu_f)^2$
$\therefore \sigma_B^2 = 2.55$

- Limitation:
  - Works well if histogram has bimodal dist.
  - Doesn't work well if,
    - → Object area is smaller compared to bg
    - → Object intensity is very dim (close to bg. intensity).
    - → Image is noisy
    - → Image contains objects (regions) with different discrete intensities