Page no.:
Date: / /

UNIT 3:

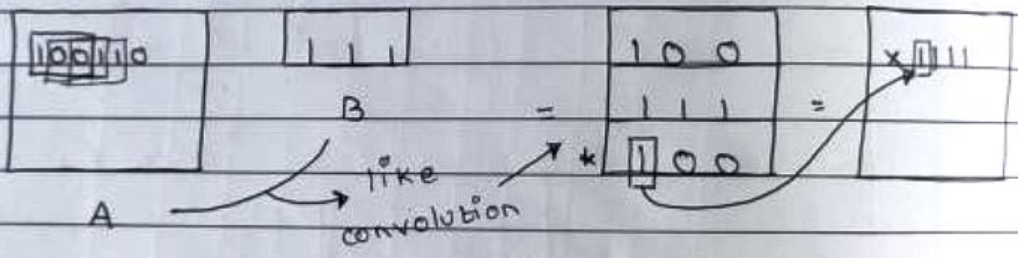
IMAGE MORPHOLOGY

* Dialation

let A(x,y) be a binary image, B as structuring element, then let C(x,y) will be,

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

$$C = A \oplus B$$



* Exosion:

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

$$C = A \ominus B$$

Page no.: ____/
Date: ____/____/____/

Q) Consider 10×10 image. Perform dilation using structuring image given.

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

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

→

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

← Same as input or blank

Page no.:
Date: / /

Dialation adds pixels to the boundary and it helps in enlarging / expanding the boundary. Size of black-box at center has reduced, and white-boundary has increased in size.

Q) Perform erosion on 10x10 image with same structuring element and image.

→ C =

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

Size of white-boundary has reduced and black-box at center has increased in size. Erosion shrinks the boundary and helps in enlarging / expanding the gaps.

Page no.: ____/
Date: ____/____/____/

- Opening:

It is erosion followed by dilation.

Consider A image and B as structuring image. then

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

- Closing:

It is dialation followed by erosion.

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

Q) Perform opening operation on given image

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

and structuring element $B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

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

Open(A, 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

Original image has 2 white blocks which were connected by thin white strip. Opening the image got rid of the strip. Size of white block unchanged. Opening breaks down narrow bridges or isolate

Page no.:
Date: / /

object which are touching one another.

② Perform closing operation

A =

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

B =

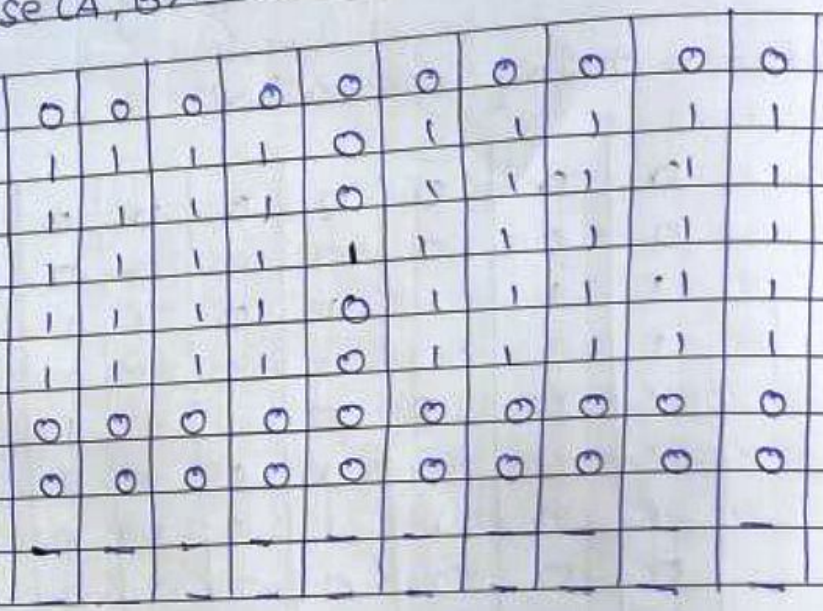
1
1

→ A ⊕ 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	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

Page no.:
Date: / /

Close (A, B) -



closing operation tends to fuse narrow ~~bridges~~ bricks

Boundary extraction can be obtained by,

$$A - [A \ominus B]$$

Page no.:
Date:

* Hit or Miss Transformation:
Small old sizes mass (normally 3x3) is scanned over entire large image (binary).
If binary pattern of structuring element matches pixels of an image, then it is "Hit" condition.
In this case, o/p pixels in spatial correspondence, the center pixel is set to desired binary state normally one(1).
If binary pattern does not match with pixels, it is "Miss" condition. In this case, o/p pixels is set to opposite binary state normally 0.

Q) Given 10x10 image, apply Hit or Miss Transformation.

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	0	0	0	0	0
0	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

Page no.: ____

Date: ____/____/____

B_1 =

X	1	X
0	(1)	1
0	0	X

→ I_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	_
	_	1	0	0	0	0	0	0	0	_
	_	0	0	0	0	0	0	0	0	_
	_									_

Q) B_2 =

X	1	X
1	(1)	0
X	0	0

B_4 =

0	0	X
0	(1)	1
X	1	X

B_3 =

X	0	0
1	(1)	0
X	1	X

Page no.: ____/

Date: __/__/____/

I_2 =

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	1	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	

I_3 =

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	

I_4 =

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	

Page no.:
Date: / /

Now, ORing I_1, I_2, I_3, I_4

$I =$

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

Page no.: ____/
Date: __/__/____/

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

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

$B_1 =$ | ① | 1 | , $B_2 =$ | 1 | ⓪ |

→ $I_1 = A \ominus B_1$

$I_2 = A^c \ominus B_2$

$R = I_1 \& I_2$

$I_1 =$

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

Page no.:
Date: / /

A^C =

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

I_2 =

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

$R = I_1 \& I_2$

=

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

Page no.: ____/
Date: ____/____/____/

* Revision *

Q) Perform histogram equalization.

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

→ Input Frequency Table:

n	0	1	2	3	4	5	6	7
nk	0	8	8	13	15	12	11	0

n	n_k	PDF	CDF	7 x CDF	n'	n_k'
0	0	0	0	0	0	0
1	8	0.125	0.125	0.875	1	8
2	8	0.125	0.250	1.875	2	8
3	14	0.219	0.469	3.283	3	14
4	15	0.234	0.703	4.921	5	15
5	12	0.188	0.891	6.237	6	12
6	7	0.110	1	7	7	7
7	0	0	1	7	7	7
	64					

Page no.:
Date: / /

Output Frequency Table

n'	0	1	2	3	4	5	6	7
n'_k	0	8	8	14	0	15	12	7

Page no.: ____/
Date: ____/____/____/

Q) Compute Hadamard Transform.

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

6	8	12	10
2	0	0	-2
0	-2	-2	-4
0	2	2	0

Q) Find 2D-DFT.

16	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

-36	0	-8	4
0	4	4	0
-8	4	4	0
4	0	0	-4

→ $F = T.f.T'$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} T'$$

$$= \begin{bmatrix} 16 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \\ 16 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$\therefore F = \begin{bmatrix} 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \end{bmatrix}$$

Page no.:
Date: / /

* 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.

Page no.: ____/
Date: ___/___/____/

- 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 Statistics 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

Page no.:
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

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

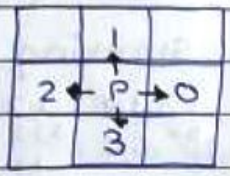
(v) Min Filter

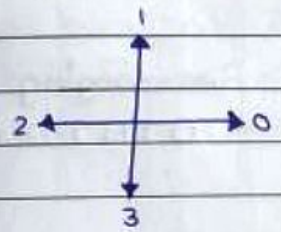
$$\tilde{F}(x,y) = \min_{(m,n)}\{g(m,n)\} = 1$$

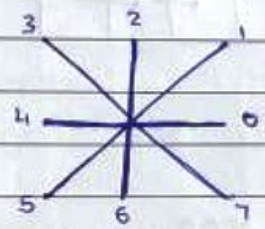
Page no.: ____/
Date: ____/____/____/

* Chain Codes:

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



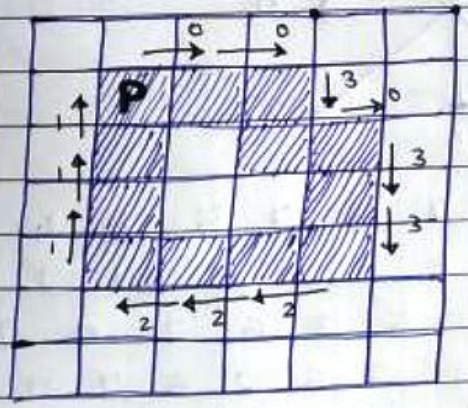




4-chain codes

8-chain codes

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



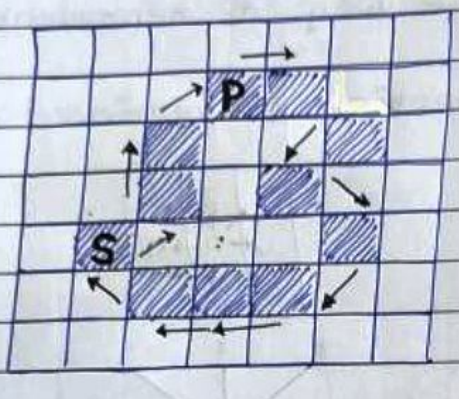
Code: 0 0 3 0 3 3 2 2 2 1 1 1

P: Starting point

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

Page no.: ____/
Date: ____/____/____

Q) Given an image, write down 8 chain codes

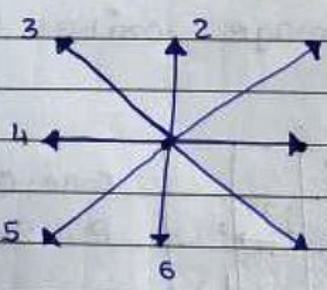


Starting P:
Code: 0 7 5 7 5 4 4 3 2 2 1
1st diff: 1 0 7 5 7 5 4 4 3 1 2 1
7 7 6 2 6 7 0 7 6 1 7

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.

Page no.: ____/
Date: ____/____/____

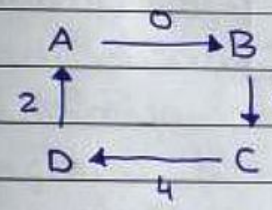
Code can be normalized w.r.t starting position and rotation.

- 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

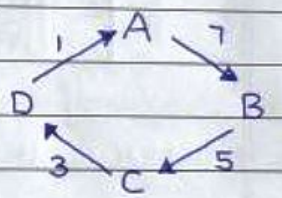
Eg.



- 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

Page no.:
Date:

IMAGE SEGMENTATION

Image Segmentation
| on basis of
↓ ↓
Similarities Discontinuties
↓
Eg: Point, line, edges
(High Freq. Component)
↓
To apply: HPF

Image segmentation segments images into small regions. It is based on similarities and discontinuties.
In similarities, we partition image into regions having similar areas based on pre-defined
In discontinuties, we divide image based on abrupt changes in intensity

* Segmentation on basis of

Points, lines, edges are basic discontinuities in image. Simple method is to get mask with coeff. sum = 0 as all are HF components.

Page no.: ____/
Date: ___/___/_____/

Point can detected using simple mask.

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

- Line detection

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

Mask 1 :

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

0	0	0	0	0	0
0	0	0	0	0	0
-30	-30	-30	-30	-30	-30
20	20	20	20	60	20
-30	-30	-30	-30	-30	-30

Page no.: ____/
Date: __/__/____/

Mask 2 :

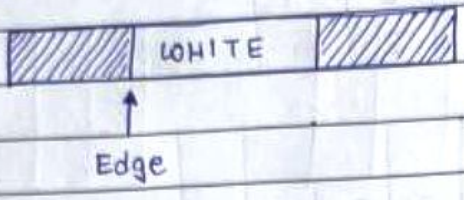
-1	2	-1
-1	2	-1
-1	2	-1

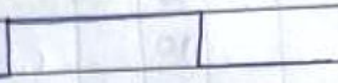
0	-30	60	-30	0	0	
0					0	
10					10	
10					10	
10					10	
0	-30	60	-30	0	0	

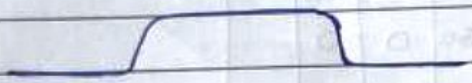
Image has 2 lines .. 1 vertical and horizontal.
By applying mask 1, we observe that it detects line only in horizontal direction and cleans up vertical lines.
To detect vertical lines, mask 2 is used.

Page no.:
Date: / /

• Edge Detection



(a)  Line Profile

(b)  Applying Anti-aliasing Filter

(c)  1st derivate

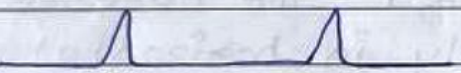
(d)  Mod of 1st derivate

Fig. (c) which is positive at leading edge and -ve at trailing edge, and 0 for rest of the areas. We get spike only at edge and hence it helps to detect edges

Page no.: ____/
Date: ____/____/____/

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

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

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

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

$$\therefore |\nabla f| \approx \left|\left(\frac{\delta f}{\delta x}\right) + \left(\frac{\delta f}{\delta y}\right)\right|$$

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

$$= \begin{array}{|c|c|}\hline 1 & 0 \\ \hline -1 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 1 & -1 \\ \hline 0 & 0 \\ \hline \end{array}$$ (Ordinary masks)

Robert's Mask:

Robert L.G. claimed that better results would be obtained, if we take cross differences instead of straight diff. Thus, Robert's mask will be return as:

$$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline -1 & -1 \\ \hline \end{array}$$

$|z_5 - z_9|$ $|z_6 - z_8|$ Sum

Page no.:
Date: / /

- Prewitts Mask

HP Mask (edge detection)

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

LP Mask (Smoothing)

F_x

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

F_y

→

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

$F_x + F_y$

$(z_1 + z_8 + z_9) - (z_1 + z_2 + z_3)$

$(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$

- Sobel Mask

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

F_x

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

F_y

→

-2	-2	0
-2	0	2
0	2	2

$F_x + F_y$

Prewitt and Sobel work (give better result) even if image is noise.

→ Derivative filters are sensitive to noise. Prewitt and Sobel filters are HPF but advantage of these two filter with edge detection is due to, these operators provided smoothing effect along with differentiation. They perform smoothing in one direction using LP component and edge detection in ⊥ direction using HP component. Hence, they perform well even on noisy images.

Page no.: ____/
Date: ____/____/____/

* Image Segmentation using 2nd Derivative (laplacian)

	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

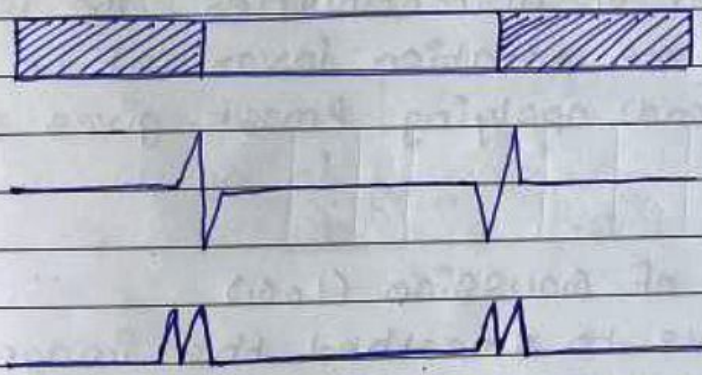
2nd derivative is given by,

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

where,

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

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



Q) Can we apply 2nd order derivative filter directly on image ?

→ Laplacian is not directly applied in its original form, as an edge detector because

- Since it is a 2nd derivative filter, it is sensitive to noise much more than 1st deriva
- If noise is present, Laplacian gives very

Page no.: ____/
Date: ____/____/____/

high value and ruins entire image.
- Magnitude of laplacian 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 preceding smoothing operator like Gaussian smoothing. The resultant algo. is called Laplacian of Gaussian or LoG.

- Advantages of 2nd derivative
  - Unlike Sobel and Prewitts, they are isotropic filters i.e. their responses independent of direction of discontinuities. The isotropic filters are rotation invariant i.e. rotating image and applying mask gives same result.

- Laplacian of Gaussian (LoG)
  - We have to smoothed the image and take laplacian to combine action of Gaussian function and 2nd derivative.
  - Consider gaussian function,

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

where,

σ = Degree of blurring
x, y = Spatial coordinate

Page no.: ____/
Date: ____/____/____/

- Let $x^2 + y^2 = r^2$

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

- 1st derivative w.r.t r.

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

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

- Now, 2nd derivative will by

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

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

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

$$\nabla^2 h = \frac{-h(r)[r^2 - \sigma^2]}{\sigma^4}$$

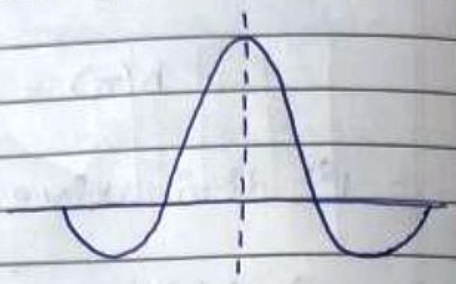
LoG can be approximated using a matrix which has large positive value at the centre and negative at periferis, as $\nabla^2 h$ function looks like Mexican Hat function.

Page no.:
Date:

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

Mexican Hat Function:



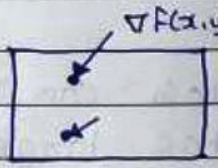
* Edge Linking:

Image → Gradient operator → Edge Linking → Good Edge Image

Techniques

- Local Processing:
  - We analyze pixels in neighbourhood. If pixels share some common property, they are linked together. Two common properties required.
    i) Strength of the Gradient

  - Strength of Gradient:
    $|\nabla f| = [F_x^2 + F_y^2]^{1/2}$



$\nabla f(x,y)$ is linked to $\nabla f(x',y')$ pixel if
$|\nabla F(x,y) - \nabla F(x',y')| \leq T$
$T \rightarrow$ Non-negative threshold
$\nabla f(x,y) \rightarrow$ Image obtained after using Sobel/Prewitt mask

Page no.: ____/
Date: __/__/____/

- Direction of Gradient

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

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

- Only if both conditions are satisfied, pixels are linked together. For every pixels in it's 8 neighbour are checked for these 2 condition.

★	★	★
★	(x,y)	★
★	★	★

N_D : Share a common point (★)
N_4 : Share a common side (★)
+↓
N_8

• Hough Transform
- In set of discrete pixels, it checks if the points lie on straight line. If yes, it draws a line joining all the points.
- Consider (x_1, y_1) as point. Line passing through those points is given by:

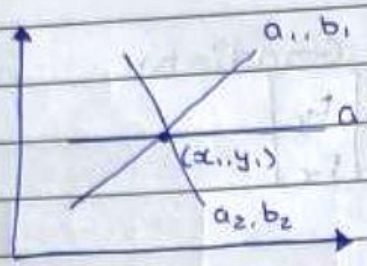
$$y_1 = mx_1 + c$$

which can be written as

$$y_1 = ax_1 + b$$

considering a: slope, b: intercept.
- By varying values of a and b, we get infinite no. of lines, passing through x_1, y_1

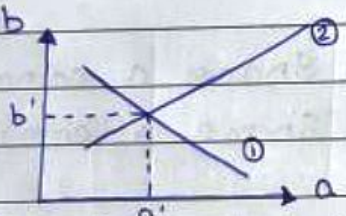
Page no.: ___/
Date: ___/___/___



- If we write,

$$b = y_1 - ax, \quad \text{①}$$

consider a, b plane instead of x, y, we will get a single line in a, b due to point (x_1, y_1).



- Line in a, b plane is due to single point (x, y) with points of a, b.
- consider another point (x_2, y_2) such that

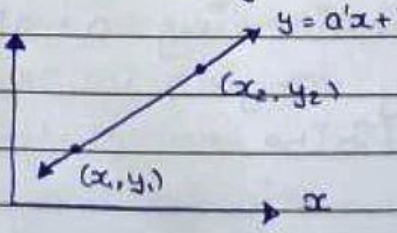
$$y_2 = ax_2 + b$$

$$b = y_2 - ax_2 \quad \text{②}$$

- This line in a-b plane will intersect first line, if (x_1, y_1) and (x_2, y_2) are part of same line in x-y plane. Intersection is given by (a', b'), will give eqn.

$$y = a'x + b'$$

- This line will pass through points (x_1, y_1) and (x_2, y_2)



Page no.: ____/
Date: ____/____/____/

- If we have n points lying on straight line, we get n lines in a-b plane, each line representing each point. These lines would intersect at single point a', b' in a-b plane. Using eqn.

$$y = a'x + b'$$

we would a line passing through all points.

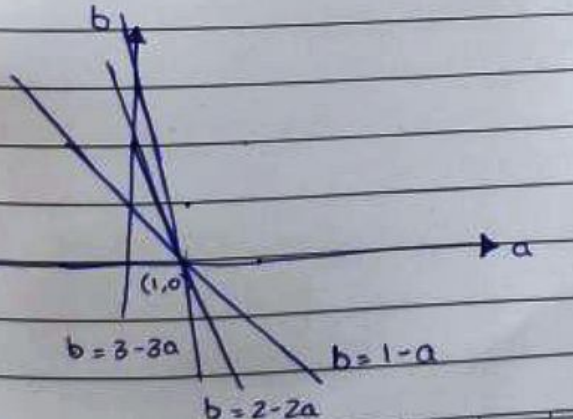
Q) Given 4 points in x-y plane with coordinates (1,1), (2,2), (3,3), (4,4). Use Hough transform to join.

→ 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	$b = 4 - 4a$



∴ Intersection of all lines: (1,0)

∴ (a', b) = (1,0)

∴ Eqn. of line will be,

$y = a'x + b'$

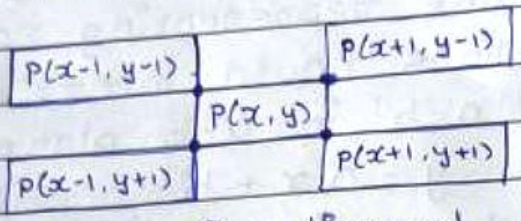
$\therefore y = 1x + 0$

$\therefore y = x$

Page no.:
Date:

* Connectivity

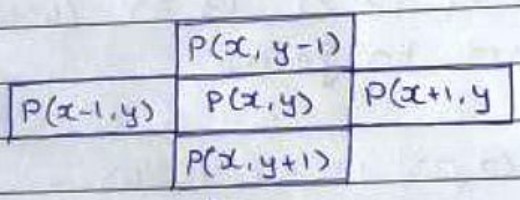
• Consider pixel P(x, y).



$N_D(P)$

- Points at ~~diga~~ diagonal

• Consider



$N_4(P)$

- Sides of pixels

• $N_8(P) = N_D(P) + N_4(P)$