Kritika

Horizontal detection mask

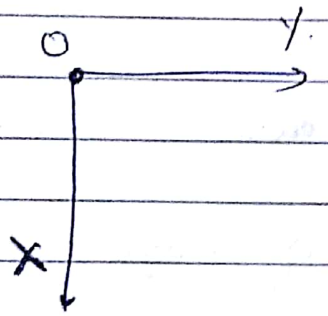
$$\begin{matrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{matrix}$$

Vertical

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

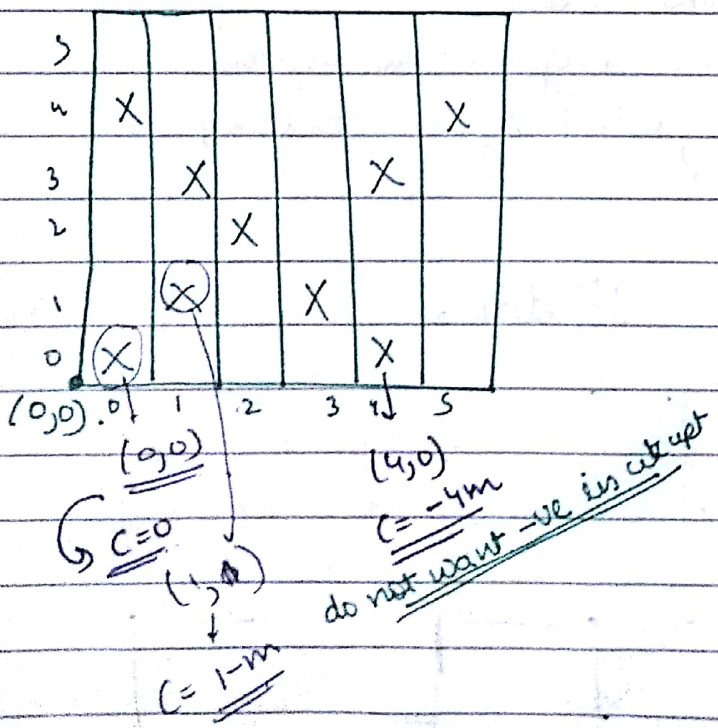
if want to extract all the 4 types of lines apply all the 4 mask

Bond mask → used for checking connections




represent line in $Y = mx + c$ → m-c space.

On mc space get only one point as one slope and one intercept.




$y = mx + c$
$c = y - mx$.

accumulator matrix.
→ accumulator space.

| c \ m | 0 | 0.5 | 1 | 1.5 | 2 |
|---|---|---|---|---|---|
| 0 | 1+1 | 1 | 1+1 | 1 | 1 |
| 1 | 1 | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

hough space

Circle → $(x-a)^2 + (y-b)^2 = R^2$

parameters → $\boxed{a, b, r}$ → hough space

instead of $y = mx + c$ we will use
$\cos\theta$ $\sin\theta$

can find any shape using hough transform.

# edge detection

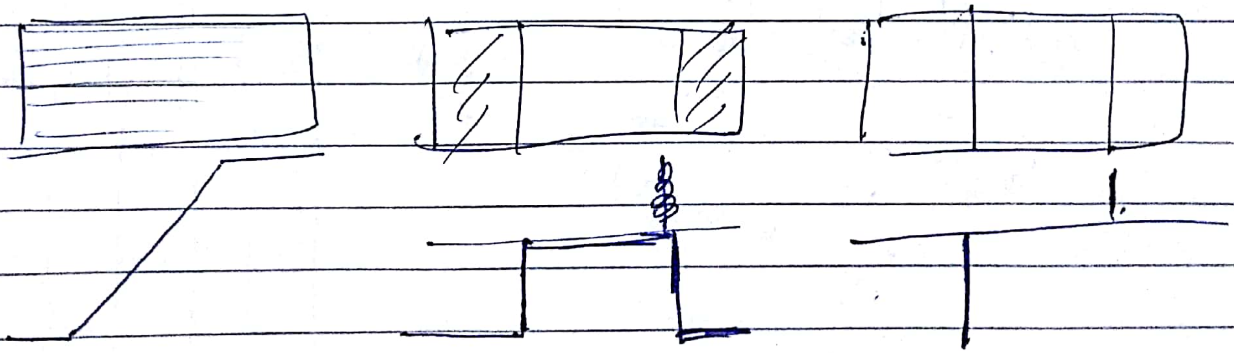
→ edges could be wider also.

first derivative → gives the rate of change.

2nd derivative → can show edge exist

apply 1st derivative mask
- if no edge → no response
- if gradual edge → some response

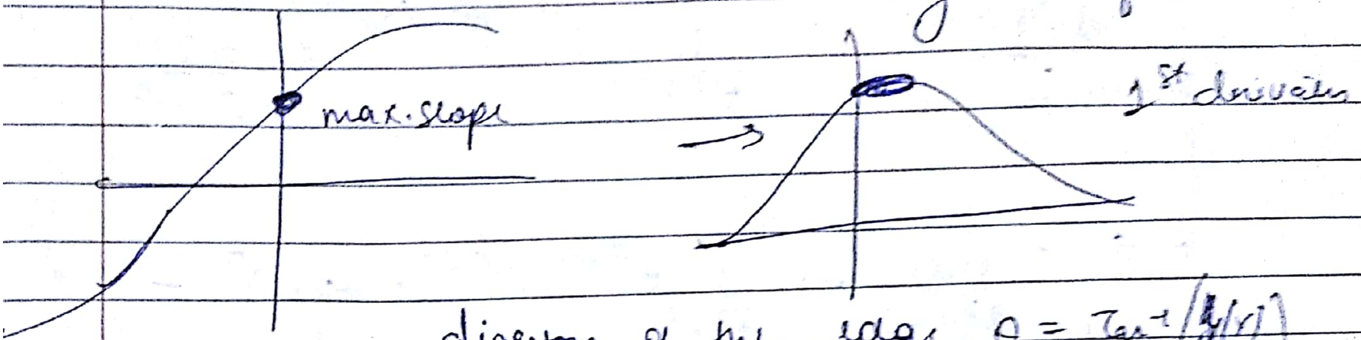
actual | 1st derivative | 2nd order



2nd order derivative is very sensitive for noise
∴ prefer 1st order derivative for edge detection

$\frac{dy}{dx} = f(\delta x + x, y) - f(x, y)$

Before applying 2nd order → reduce the noise
i.e., apply the averaging filter
→ gaussian filter



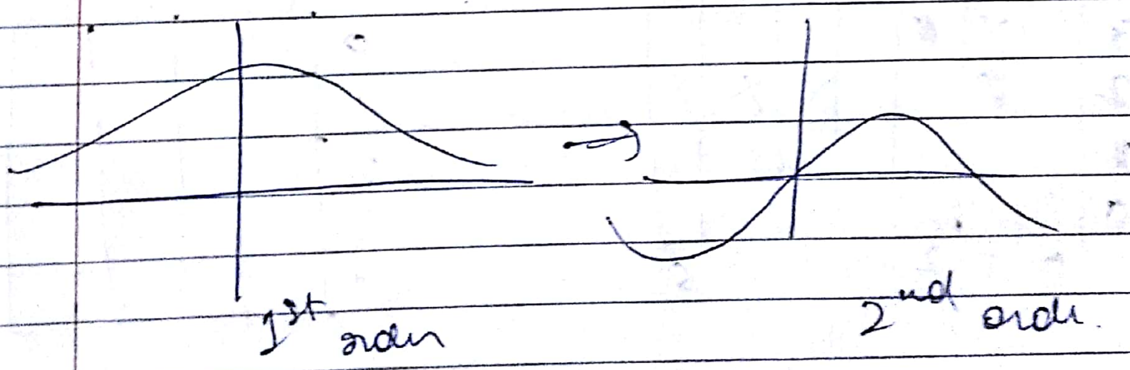

direction of the edge $\theta = \tan^{-1}\left(\frac{g_y(x)}{g(x)}\right)$

apply Robert on Im.

$-Z_5 + 0 + 0 + Z_9$

In all the mask sum of the coefficients is 0
because → area of smooth intensity will not show up

# 2nd order derivative




laplacian of gaussian
→ first gaussian then laplacian.
$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$

Date

$$\frac{d(G(x))}{\partial} \rightarrow \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \times \frac{-2x}{2\sigma^2}$$

$$\frac{d(g(y))}{\partial} \rightarrow \frac{1}{2\pi\sigma^2} \cdot e^{-(x^2+y^2)/2\sigma^2} \times \frac{-2y}{2\sigma^2}$$

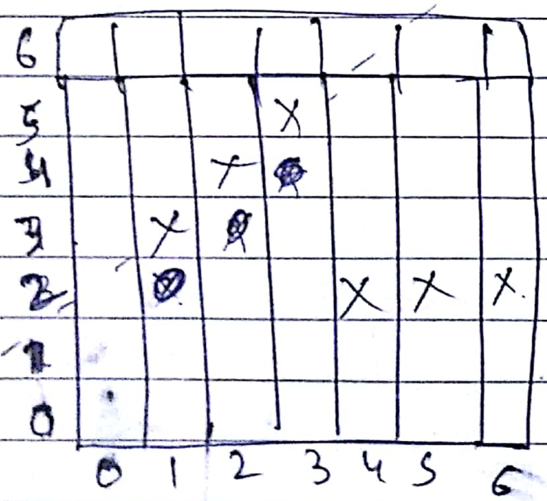
$$\frac{\partial^2 G(x)}{\partial^2} \rightarrow \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \cdot \left(\frac{-2x}{2\sigma^2}\right)^2 - \frac{1}{2\pi\sigma^4} e^{-(x^2+y^2)/2\sigma^2}$$

$+$

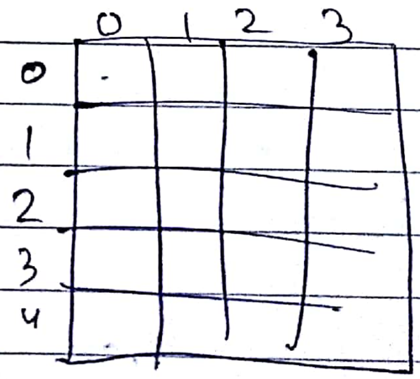
$$\frac{\partial^2 G(y)}{\partial^2} \rightarrow \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \cdot \left(\frac{-2y}{2\sigma^2}\right)^2 - \frac{1}{2\pi\sigma^4} e^{-(x^2+y^2)/2\sigma^2}$$

$$\Rightarrow (x^2+y^2)\left[\frac{1}{2\pi\sigma^4} e^{-(x^2+y^2)/2\sigma^2}\right] - \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

4)



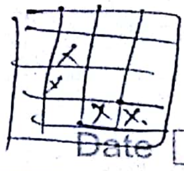
accumulator.



intercept $\rightarrow$ $c=2$

$m=0, c=2$

$m=1, c=2$



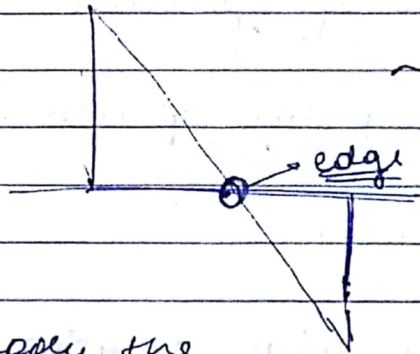
Cannot directly apply laplacian as 2nd order derivative is very sensitive to noise

$y = mx + c$

and

there are 2 responses one at positive side and other at negative side.

2nd order derivative.




→ to remove noise, we apply the gaussian & LoG

(1) apply the gaussian mask on the image. → $\frac{1}{9}$

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 2 | 4 | 2 |
| 1 | 2 | 1 |

then apply the laplacian mask

(or)

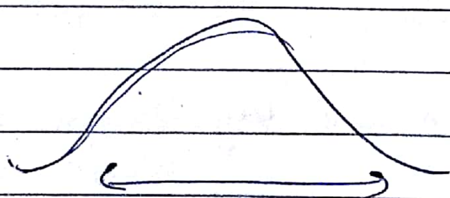
| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

(2) first apply laplacian on gaussian mask, then apply this mask on the image

of the equivalent mask

if the center pixel is (-ve) subtract from the image

if (+ve) add to the image



$\sigma$ → greater it is more smoothening will happen

Date

Sobel → 1st order derivative.

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

→ laplacian

# Difference of gaussian (DoG)

# optimal edge operator

localization. → min distance

# Canny edge operator

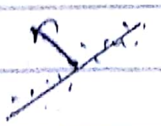
Gradient values → Roberts, Sobel, Prewitt.

Strength of the edge → Magnitude of the edge.

$\tan^{-1} \frac{G_Y}{G_X}$ } orientation.

normal of the edge gives the orientation.

non maxima Suppression.

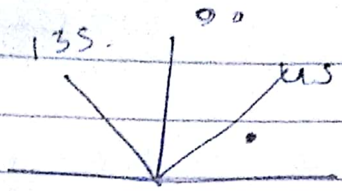


many things will be having this direction

Some of them are edge points and some of them are not.

if magnitude of any point is larger → edge.
having non larger magnitude are suppressed
∴ we will have the clear edges.

Date




(find the closest direction).

if less than its neighbours → suppress it.

(O)

maintain the 2 thresholds                Strong edge point

Stricter threshold — definitely belong to edge ($g_H$) { group

$T_H$   $T_1$ → if magnitude is above this threshold, belong to the

$T_L$   $T_2$ — if mag is $< T_L$ → very weak point and need to be suppressed.

→ ($g_L$)

Point b/w these group

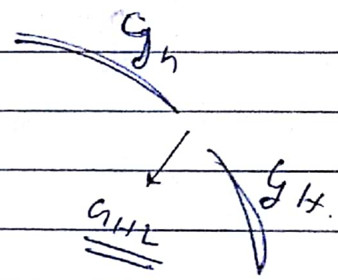
$T_H < \; < T_L$

→ might belong to the edge.

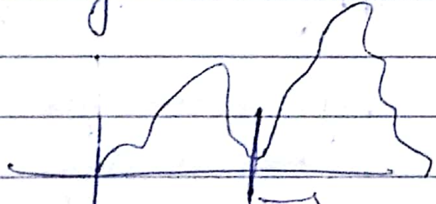
first we check all the pixels in $G_H$

if any pixel in $G_{HL}$ has any connectivity with pixel in $G_H$

we add it to the $G_H$

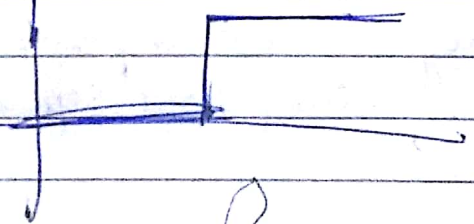


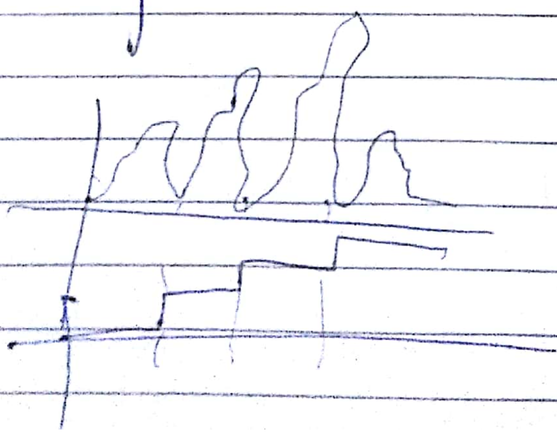

# Histogram thresholding.



→ histogram.

→ can use this valley for threshold





Date

in case we find the mean pixel

(1) global thresholding. → dependent upon the intensity value of all the pixel

(2) adaptive thresholding → depends upon the intensity as well as location.

global

(1) take a global mean, and then threshold.

now find the mean of foreground pixels

~~i.e. no of p~~

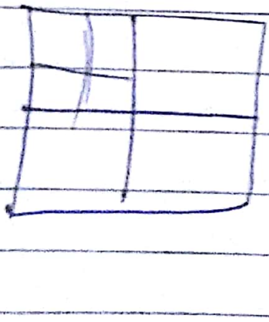
$\mu_1$, $\mu_2$

$T_2 \rightarrow \frac{\mu_1 + \mu_2}{2}$ | new threshold.

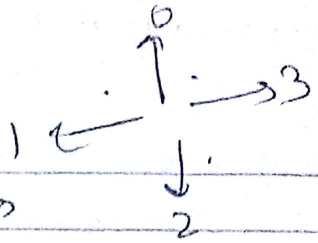
now apply it on the original image

find the thresholds again and again and stop when diff b/w thresholds is very less.

divide the image into 4 parts
do thresholding of every part separately

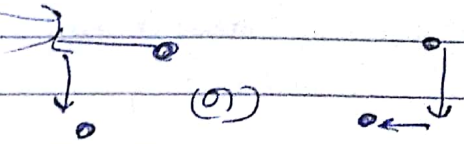


# Chain code




either 4 connectedness or 8 connectedness

going up → 0 } code.
going right → 3 }




choose any one

2121666006465 44124 → can be different for every one

→ normalize it

find the difference b/w 2 and 1 → 3. going in
1 and 2 → 1 anticlockwise
in 4 connected wise dir

2121666 00 64 65 44124.
7 1 7 5 0 0 2 0 6 → 1st difference

etc

have to make it rotation invariant. → when you put your 0, 1, 2, 3 — invariant.

rotation 7175002066 2 — —

and find the smallest one

0001225 6677 . —

→ i.e. invariant of from where do you start your chain.

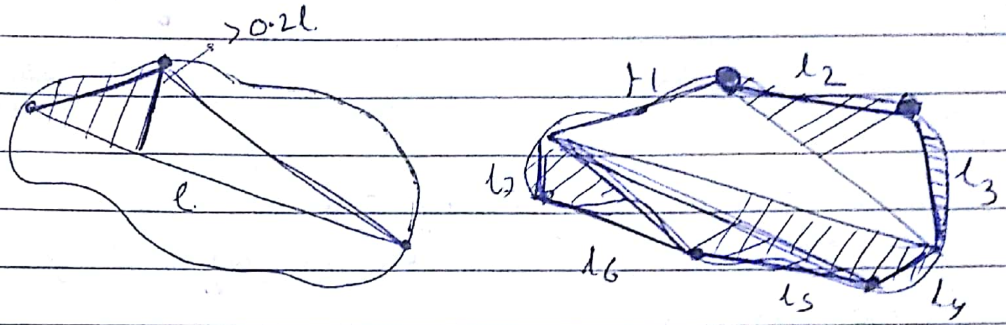
Signature
→ 2D boundary ⇒ ~~two~~ representative one D function

Date

# Polygonal Approximations : splitting

length of line → l.
Threshold → 0.2 l.

any length that is greater than threshold, is also the extreme point.




Polygon → [l1, l2, l3, l4, l5, l7]
→ feature vector

(or)

Can store the changes in the slopes

(or) all the pixels
(or) m, c and extreme points

# Signature → invariant to location, but depend on rotation and scaling

① to decrease dependence on rotation → using the major axis of the regions.

Start at the centroid.
then suppose, we go at ~~d~~ increment of 30°.
for square → incrementing the ~~Square~~ angle by 45°

② scale invariance → scaling signature function to fixed amplitude by dividing fun^n values by the standard deviation of the function

# (or) Can also represent any polygon by its convex hull. (Points which are part of object as well as of convex hull).

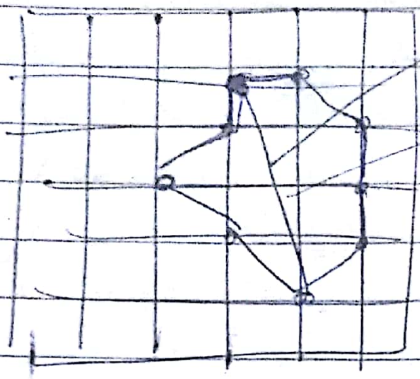
# Boundary Descriptors
(very rough representation).

Scanned by CamScanner

No. of vertical component + No. of horizontal comp + $\sqrt{2}$ diagonal component = length of Boundary

diameter (distance)

Major axis



v → 3

H → 1

d → 5

$$\boxed{3 + 1 + 5\sqrt{2}}$$

Diameter of Boundary (B).

$$Diam(B) = \max [D(P_i, P_j)].$$

max euclidean distance

line → Major axis

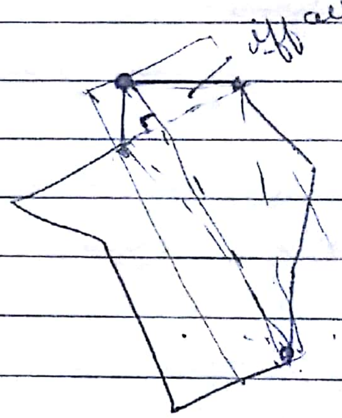
distance → diameter.

line ⊥ to major axis

minor

i.e, all Boundary point should lie in this rectangle → Basic Rectangle.

joining the extremeties of Major and minor axis



iff all Boundary points come into this rectangle then minor axis.

eccentricity → $\frac{\text{Major}}{\text{minor}}$

Order 8 then no. is shape no → 8.

images we take fourier Transform

not fourier Series

Date

high frequency → where there are many changes

take amplitude   subtract mean
if 2nd order moment n=2

Skewed towards left or Right (shifted)

Perimeter → Boundary descriptor but used as area descriptor because it gives the Boundary.

Euler No. → Connected component − no. of holes

~~Black~~ Foreground pixel.

A → 0   (1 − 1)  CC, hole
B → −1   (1 − 2)  CC, hole.

Picture elements → Pixels
Texture elements → Texel

Checkboard → high frequency image

# GLCM gray level ~~correlation~~ Matrix
co-occurrence.

| | | |
|---|---|---|
| 0,0 4 | 0,1 4 | 0,2 1 |
| 1,0 4 | 1,1 | 1,2 |
| 2,0 | 2,1 | 2,2 |

0, 1, 2 → 3 gray level

(0 0) → 4

one pixel to the grey

0 0 0

0 0

0 0

Date

0 0 0 1 2      0,0 → 4
1 1 0 1 1      0,1 → 4
2 2 1 0 0      0,2 → 1
1 1 0 2 0      1,0 → 4.
0 0 1 0 1

Histogram same ] for Block
GLCM → different ] check Board and Strips

1 Pixel Right    1 pixel Below.

0 0 0
1 1 0

| 00 | 01 | 02 |
|---|---|---|
| 4 | 2 | 0 |
| 10 | 11 | 12 |
| 20<br>1 | 21 | 22 |

| | | |
|---|---|---|
| 00 | 10 | 20 |
| 01 | 11 | 21 |
| 02 | 12 | 22 |

# Quadtree

4

Splitting and merging

Break any Image into 4.
∴ 4 Sub Images
if Sub Image is Smooth → do nothing
if even 1 pixel is not Smooth then Break Sub Image even into 4. Sub Images

$\boxed{4,5}$ → have Same intensity Values
∴ all merge them.

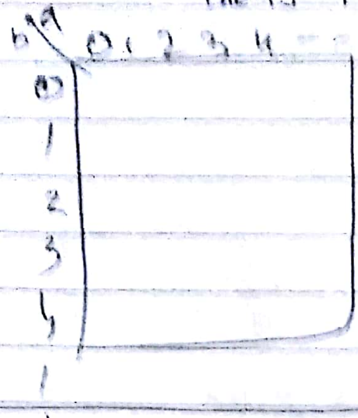
merge 8,9,10 → having Same Values

Hough

Eq$^n$ of circle

$$(x-a)^2+(y-b)^2 = r^2$$

$$(x-1)^2+(y-0)^2 = 25$$




(at the end we count the value).

changing the pixel value

frequency transform

1-D.

$$F(u) = \boxed{\frac{1}{N}\sum_{x=0}^{N-1} f(x)\, e^{-j2\pi\left(\frac{xu}{N}\right)}}$$

$x(n) = [2\ 2\ 1\ 1]$

$\boxed{e^{j\theta} = \cos\theta + j\sin\theta}$

$$F(u) = \frac{1}{4}\sum_{x=0}^{3} f(x)\, e^{-j\frac{\pi}{2}(xu)}$$

$\cos(-\theta) = \cos\theta$

$\sin(-\theta) = -\sin\theta$

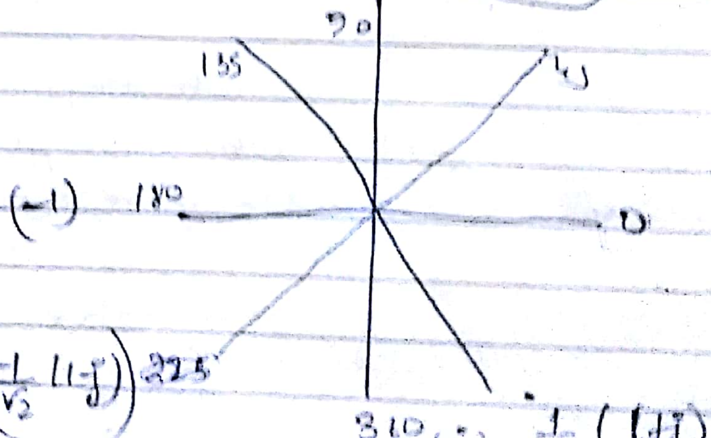
$\boxed{e^{-j\theta} = \cos\theta - j\sin\theta}$

If frequency component is 0

i.e., $u=0$.

$$F(u) = \frac{1}{4}\sum_{x=0}^{3} f(x)$$

$=$ average



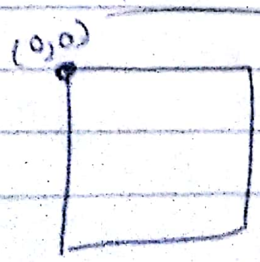

$\frac{1}{\sqrt{2}}(1+j)$

$135 \to -\frac{1}{\sqrt{2}}(1+j)$

If the frequency is 0 then we get the average

↓

$\boxed{\text{DC component}}$




$0 \to 1$

$45 \to \frac{1}{\sqrt{2}}(1-j)$

$90 \to -j$

Scanned by CamScanner

$$F(0) = \frac{1}{4}\sum_{n=0}^{3} f(n) = \frac{1}{4}(6) = \frac{3}{2}$$

$$F(1) = \frac{1}{4}\sum_{n=0}^{3} f(n)\, e^{-j\frac{\pi}{2}n}$$

$$= \frac{1}{4}\left[f(0) + f(1)e^{-j\pi/2} + f(2)e^{-j\pi} + f(3)e^{-j3\pi/2}\right]$$

$$= \frac{1}{4}\left[2 + 2\cdot(-j) + 1(-1) + 1\cdot j\right]$$

$$= \frac{1}{4}\left[2 - 2j - 1 + j\right]$$

instead of Normalizing here we can also normalize when we take inverse transform.

$$F(2) = \frac{1}{4}\sum_{n=0}^{3} f(n)\, e^{-j\pi n}$$

$$= \frac{1}{4}\left[f(0) + f(1)e^{-j\pi} + f(2)e^{-j2\pi} + f(3)e^{-3\pi}\right]$$

$$= \frac{1}{4}\left[2 + 2\cdot(-1) + 1(1) + (1)(-1)\right]$$

$$\Rightarrow \frac{1}{4}(0) \to 0.$$

$$F(3) = \frac{1}{4}\sum_{n=0}^{3} f(n)\, e^{-j\frac{\pi 3}{2}n}$$

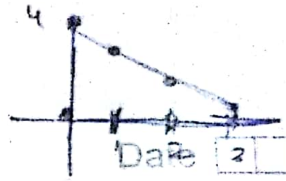
$$= \frac{1}{4}\left[f(0) + f(1)e^{-j3\pi/2} + f(2)e^{-3\pi j} + f(3)\cdot e^{-\frac{9\pi}{2}}\right]$$

$$= \frac{1}{4}\left[2 + 2(j) + 1(-1) + 1(\quad\right]$$

$$= \frac{1}{4}(1 + j)$$

$$\Rightarrow F(n) = \frac{1}{4}\left[6 \quad 1-j \quad 0 \quad 1+j\right]$$

discrete signal which is given




Date 2

eg $x(n) = [4 \quad 3 \quad 2 \quad 1]$

$$F(0) = \frac{1}{4} \sum_{n=0}^{3} f(n) \cdot e^{-j \pi/2 (nu)}$$

$$= \frac{1}{4} \sum \quad \boxed{10}$$

$F(1) = 2-2j$

$F(2) = 2$

$F(3) = 2+2j$

$$F(n) = \frac{1}{4} [10 \quad 2-2j \quad 2 \quad 2+2j].$$

# Now image is 2D  $M \times N \rightarrow$ image size

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \, e^{-j 2\pi \left(\frac{xu}{M} + \frac{yv}{N}\right)}$$

eg Image

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

now find the 2D transform of this image.

one D → ~~fourier~~ frequency transfer

first find the DFT of all the rows

then take DFT of all the columns

Date

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \\ 1 \\ -3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 8 \\ -2 \\ 0 \\ -2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 3 \\ 3 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 11 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

intermediat Matrix

$$\therefore \begin{bmatrix} 0 & 1 & 3 & 1 \\ 1 & 2 & 3 & 2 \\ 3 & 3 & 2 & 3 \\ 1 & 2 & 3 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 5 & -3 & 1 & -3 \\ 8 & -2 & 0 & -2 \\ 11 & 1 & -1 & 1 \\ 8 & -2 & 0 & -2 \end{bmatrix}$$

one D DFT of rows of original matrix

now calculate one D DFT of columns of intermediat Matrix

Date

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 8 \\ 8 \\ 11 \\ 8 \end{bmatrix} \Rightarrow \begin{bmatrix} 32 \\ -6 \\ 0 \\ -6 \end{bmatrix}$$

$$\begin{bmatrix} 5 & -3 & 1 & -3 \\ 8 & -2 & 0 & -2 \\ 11 & 1 & -1 & 1 \\ 8 & -2 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 32 & -6 & 0 & -6 \\ -6 & -4 & 2 & -4 \\ 0 & 2 & 0 & 2 \\ -6 & -4 & 2 & -4 \end{bmatrix}$$

$\Downarrow$

nomalizing it

$$\frac{1}{16} \begin{bmatrix} 32 & -6 & 0 & -6 \\ -6 & -4 & 2 & -4 \\ 0 & 2 & 0 & 2 \\ -6 & -4 & 2 & -4 \end{bmatrix}$$