Q- Given an 3X3 image, give its bit planes.

1	2	0
4	3	2
7	5	2

grayscale

max no = 7
3 - bit planes will be there

binary ↵

001 010 000
100 011 010
111 101 010

0	0	0
1	0	0
1	1	0

MSB

→ will carry most info

0	1	0
0	1	1
1	0	1

Mid-plane

1	0	0
0	1	0
1	1	0

LSB

steganography - art of hiding information

50 binary.

$2^6 = 64$

$2^8 = 256$

110010

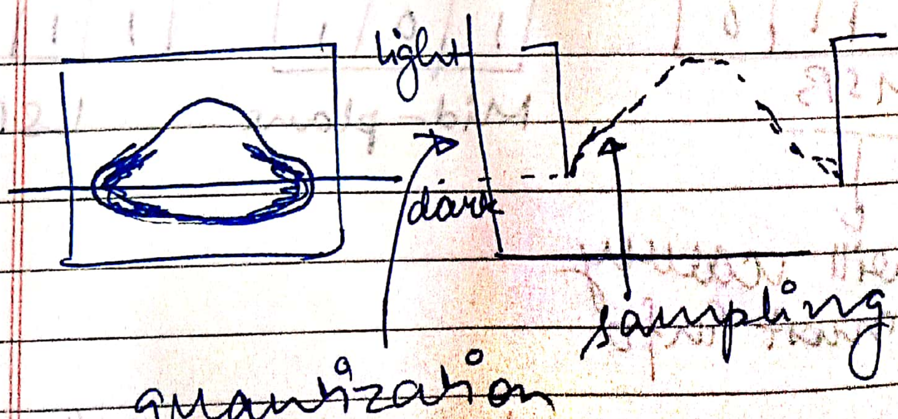
150

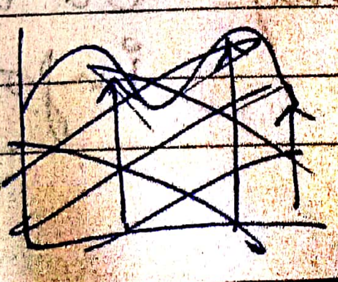
10010110

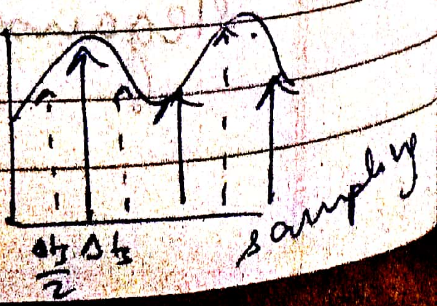
Digitization

picking up the co-ordinates or locations is ~~caused~~ called sampling.

picking up the amplitude or intensity is called quantization







$$f_s = \frac{1}{\Delta s}$$

sampling rate small for reconstruction

$$\frac{2}{\Delta t_s} = 2 \Delta f_s$$

Sampling theorem

A continuous time signal can be completely represented in its samples and recovered back, if the sampling freq. (f_s) $\geq$ 2 (highest freq component of the msg signal f_m)

$$\boxed{f_s \geq 2 f_m}$$

Nyquist rate

$$x(t) = \sin 2\pi t + \sin 3\pi t + \sin 4\pi t$$

since $\omega_1 = 2\pi$

$2\pi f_1 = 2\pi$

$\Rightarrow f_1 = 1$

$\omega_2 = 3\pi$

$2\pi f_2 = 3\pi$

$\Rightarrow f_2 = \frac{3}{2} = 1.5$

$\omega_3 = 4\pi$

$2\pi f_3 = 4\pi$

$\Rightarrow f_3 = 2$

$$f_m = \max(f_1, f_2, f_3) = 2$$

$$\boxed{f_s \geq 4}$$

DIFFERENT IMAGE FORMATS

diff

JPEG
not good for sharp images or text.
lossy Why? - [search]

PNG
lossless

GIF
limited for 256 colors.
CLUT

TIFF
lossless format
compatible to all OS
not copyrighted
huffman coding.

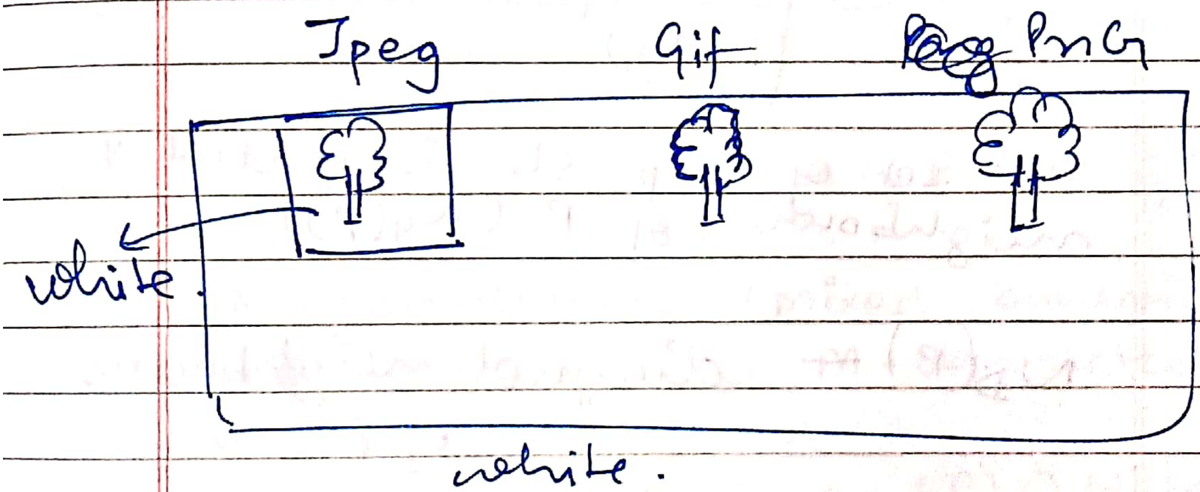
Bitmap
Raster graphics, Microsoft proprietary
no compression Windows Platform

RAW

capture full dynamic range.
pre-processing
negative

EPS

scaling while maintaing quality
scalable to any size.



In Gif — white flicker on our boundaries

PnG — no white proper bckgrnd as it is

JPEG — background will show.

Neighbourhoods of a pixel

$N_4(P)$: A pixel p at location (x, y) has 2 horizontal & 2 vertical neighbours.

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

The set of 4 pixels is called 4 neighbours of P ($N_4(P)$)

$N_D(P)$ — diagonal neighbours of P.

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

$N_8(P)$ — combination of $N_4(P)$ & $N_D(P)$.

Connectivity of pixels

~~Set of~~ Let ~~V~~ be the set of gray levels used to define connectivity for 2 points ~~P, Q~~ $p, q \in V$, 3 types of connectivity are defined:

1. 4-connectivity: $p, q \in V$ & $q \in N_4(P)$.

2. 8-connectivity: $p, q \in V$ & $q \in N_8(P)$

3. M-connectivity: (mixed Connectivity) ~~if~~ $p, q \in V$ are M-connected if 1) $q \in N_4(P)$ OR
   2) $q \in N_D(P)$ & $N_4(P) \cap N_4(q) = \phi$ (empty)

$V = \{1\}$

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

~~Connects~~ show all 4-connected components

$$\begin{array}{ccc} 0 & 1 & 1^{\,q} \\ 0 & {}_{p}1 & 0 \\ 0 & 0 & 1 \end{array}$$

8-connected
$V = \{1\}$

multiple paths b/w p & q.
to remove this, we use
m-connectivity

$$\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

m-connectivity

$V = \{89, 60, 74\}$

$$\begin{array}{cc} 200 & 60^{\,q} \\ 89_{\,p} & 101 \end{array}$$

they are m-connected

$N_4(p) \cap N_4(q) = \underbrace{\{200, 101\}}_{\text{not in } V}$

$\therefore \phi$.

$$\begin{array}{cc} 74 & {}^{q}60 \\ 89_{\,p} & 101 \end{array}$$

not m-connected

$V = \{2, 3, 4\}$

2	8	7
2	3	6
2	4	3

4-connected

Q-

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

$V = \{1\}$

① 4-connectivity
② 8-connectivity
③ M-connectivity

①

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

②

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

③

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

4	1	3	2
2	3p	4	5
5	2	2q	4
2	2	4	3

$V = \{2, 3, 4\}$

whether p & q are m-connected. Why

not m-connected

$\because N_4(3) \cap N_4(2) = \{4, 2\} \neq \phi$

Path: Path from $P(x,y)$ to $q(s,t)$ is defined as a sequence of distinct pixels.

$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$

where

$(x_0, y_0) = (x, y)$, $(x_n, y_n) = (s, t)$

(x_i, y_i) is adjacent to (x_{i-1}, y_{i-1}) for $1 \leq i \leq n$

n = length of path.

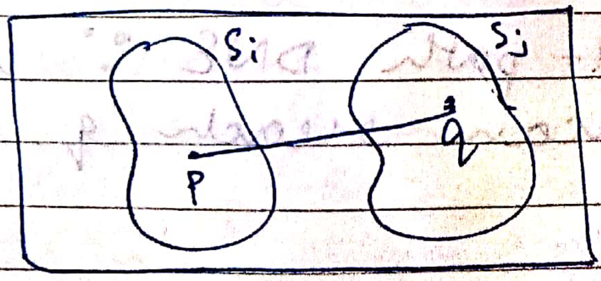
Adjacency: Two pixels p & q are adjacent if they are connected depending upon connectivity used.

Scanned by CamScanner

Let V be the set of grey levels used to define adjacency

- 4 - adjacency: 2 pixels p & q with values from V are 4 - adjacent if q is in the set of $N_4(p)$
- 8 - adjacency: if q is in the set of $N_8(p)$
- m - adjacency: if q is in the set of $N_m(p)$

Eg:- 2 unit subsets S_i & S_j are adjacent if there exists a $p \in S_i$ & there exists a $q \in S_j$ s.t p & q are adjacent.



Q– Let $V = \{0, 1\}$ & compute the length of the shortest 4, 8 & m-path b/w p & q. If a particular path does not exist b/w these pts explain why?

$$
\begin{matrix}
3 & 1 & 2 & 1\,(q) \\
2 & 2 & 0 & 2 \\
1 & 2 & 1 & 1 \\
(p)\,1 & 0 & 2 & 2
\end{matrix}
$$

4 – path

$$
\begin{matrix}
3 & 1 & 2 & 1\,(q) \\
2 & 2 & 0 & 2 \\
1 & 2 & 1 & 1 \\
(p)\,1 & 0 & 2 & 2
\end{matrix}
$$

4 – path DNE $\because$ no $N_4(p)$ can reach q.

Gonsalvez

8-path

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

Length = 4

m-path

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

Length = 4

- Connected Component:

  Let S be the subset of I

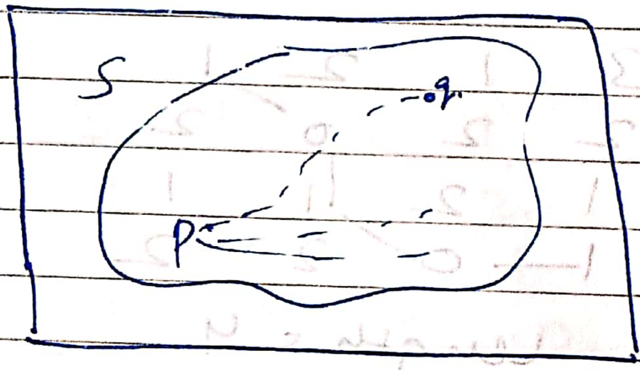
  $S \subseteq I$ (Image)

  & $p, q \in S$. Then p is connected to q in S if there is a path from p to q consisting entirely the pixels in S.

For any $p \in S$, the set of pixels in S that are connected to p is called connected component of S.

Any 2 pixels of a connected

component are connected to each other. Distinct ~~comm~~ connected components are disjoint. By using the concept of connected component, I can identify a region in a image.



Q- ~~prev~~ Repeat for $V = \{1, 2\}$

$$\begin{array}{cccc} 3 & 1 & 2 & 1\ (q) \\ 2 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ 1 & 0 & 2 & 2 \\ (p) & & & \end{array}$$

4-connected

Length = 6

$$\begin{array}{cccc} 3 & 1 & 2 & 1\ (q) \\ 2 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ 1 & 0 & 2 & 2 \\ (p) & & & \end{array}$$

8-conn

length = 4

m-conn

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

length = 6

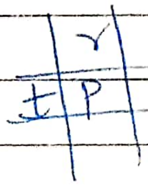
3	1	2	1
2	2	0	2
1	2	1	T
1	0	2	2

• Left to right scanning

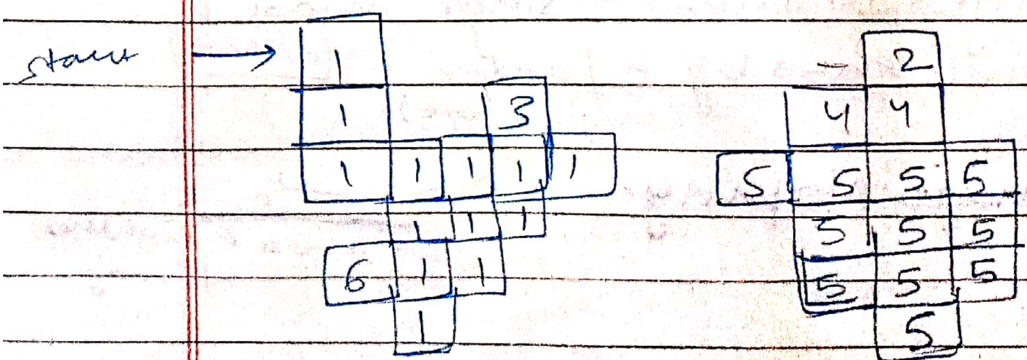
Component Labelling Algorithm

1st Pass: $l(P)$ = Pixel value at location P.
$L(P)$ = Label assigned to pixel location P.

- if $l(P)=0$, move to the next scanning location.
- if $l(P)$ is in V & $l(r) = l(t) = 0$, Assign new label to $l(P)$
- if $l(P)$ is in V, & $L(r) \neq L(t)$, Assign one of the label to $l(P)$
  $L(r) = L(t)$, then $L(P) = L(r)$

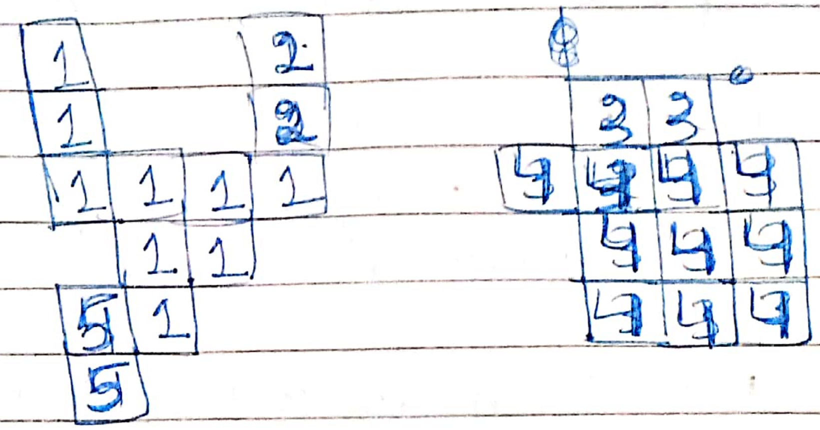


Second pass: process equivalent pairs to form equivalence classes. Assign a different label to each class.



Equ. class

Equ pairs (1,3) (1,6) ⇒ 1, 3, 6 = 1 new label
(2,4) (4,5) ⇒ 2, 4, 5 = 2 new label



equivalence pairs

(1,2) (1,5) ⇒ (1,2,5)

⇒ 1 new label

(3,4) ⇒ (3,4)

⇒ 3 new label

Image Algebra

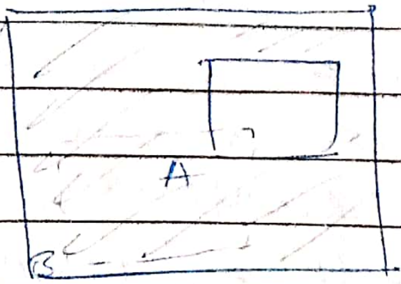
Arithmetic
- Addition
- Subtraction – identification of moving objects
- Multiplication – single image by a factor
- Division → by a factor (single image)

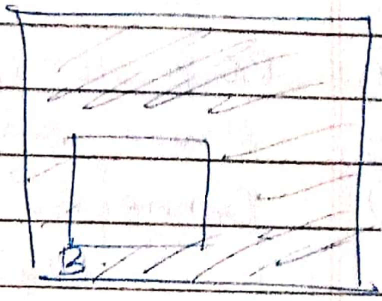
scalar

~~[illegible]~~ darkness & lightness of image

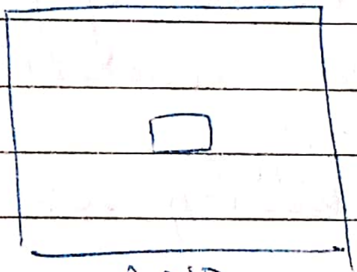
→ **Image Morphing**

logical operators
- AND
- OR
- NOT
- XOR

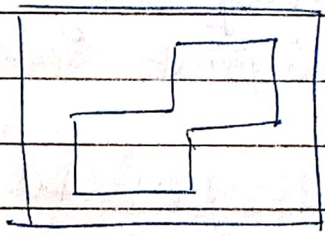




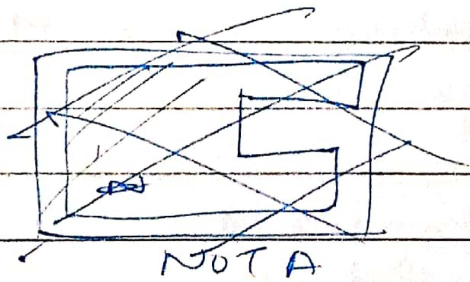
NOT — ~~inverse~~ ~~XOR~~
AND — overlap
OR — both



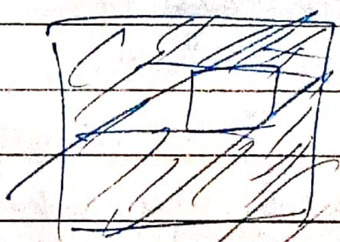
AND



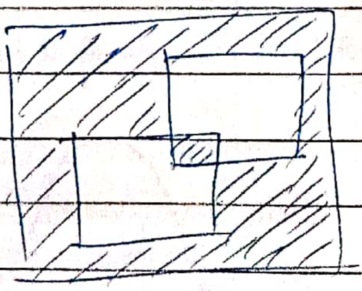
OR.



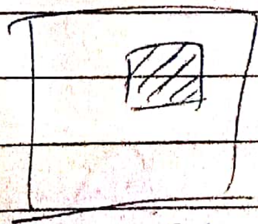
NOT A



NOT A



XOR



NOT A

Implementation

1. RGB to GRAY.
   Show its 3 planes & convert to GRAY
2. Connectivity - identify 4 neighbours $N_4(P)$; $N_8(P)$ & $N_D(P)$

3. Distance ~~Metho~~ Measure
   Adjacency. (b/w any p & q)

4. Connected component of p for disjoint images.

5. Component labelling

~~6~~ ~~[illegible]~~

filtering

Mask convolve on image ⇒ modified Img

Mask

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Image

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

O/P ⇒

	z_5'	

$$z_5' = w_1 z_1 + w_2 z_2 + w_3 z_3 + \cdots + w_9 z_9$$

Linear: when O/P is weighted sum of i/P pixels.

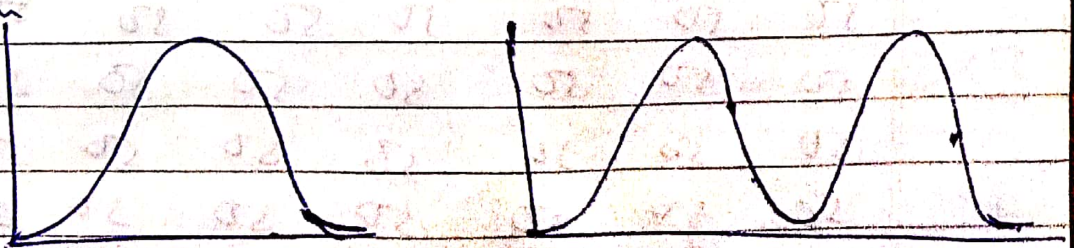
Non-linear: $z_5' = \text{Max/min}(z_k,\ k=1,2,\cdots 9)$

Noise: Impulse noise ↳ random occurrence of white intensity pixels

Salt & pepper noise: random occurrences of black & white intensity pixels.

Gaussian noise

If histogram is generated as gaussian distribution, then gaussian noise is corrupted in sample.



* Other noise study on your own.

low pass filters

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

Averaging Mask

smooth regions of image - low frequency component

immediate change - high freq component

low pass filters retain low freq components & blur the high freq component

$\frac{1}{(9)}$ → to retain original pixel values

Apply

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	23	23	23	23	23	23	10
50	36	36	36	36	36	36	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

O/P img.

$z_5' = \frac{1}{9}[10 \times 9] = \underline{10}$

$3' = \cancel{\frac{1}{6} \times 60 + \frac{1}{3} \times 50}$ $\quad \frac{60+150}{9}$

$= \frac{210}{9}$

$= 23.3$

~~10 + 10 + 30 + 20 + 150 + 23~~

110

10 10 10
50 50 50 $\Rightarrow$ $\frac{30 + 300}{9} = \frac{330}{9}$
50 50 50

$= 36.6$

if image $\rightarrow$ 6×6 & mask 3×3

O/P img. 4×4 to solve it, we replicate the ~~last~~ border of I/P img so that after O/P we get original size.

after padding $\rightarrow$ 8×8 (I/P) $\longrightarrow$ 6×6 O/P.

6x6 $n \times n \rightarrow$ original Img

3x3 $f \times f \rightarrow$ mask

padding
p

$n+2p-f+1 \times n+2p-f+1$

$6+2-3+1 \times 6+2-3+1$

$\boxed{6 \times 6}$

$\cancel{n}+2p-f+1=\cancel{n}$

$\boxed{p = \frac{f-1}{2}}$

padding

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	220	10	140	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	220	50	190	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

to remove impulse noise,
non-linear filtering used.
mask

take

$z_5' = \text{median}(z_k, k=1,2 \ldots 9)$

10 10 10 10 10 50 50 50 220

O/P img

10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10
50	50	50	50	50	50
50	50	50	50	50	50
50	50	50	50	50	50

High Pass filter

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

- useful for highlighting fine details. The elements of the mask contains both the +ve & −ve wts

Sum of the mask elements is 0

$\frac{1}{9n}$ scaling factor

−1	−1	−1
−1	8	−1
−1	−1	−1

$-50+80$
-30

after high pass filter

10	10	10	10	10	10	10	10	10	10
10	0	0	0	0	0	0	0	0	10
10	0	0	0	0	0	0	0	0	10
10	-270	-270	-270	-270	-270	-270	-270	-270	10
100	270	270	270	270	270	270	270	270	100
100	0	0	0	0	0	0	0	0	100
100	0	0	0	0	0	0	0	0	100
100	100	100	100	100	100	100	100	100	100

↳ padding.

10	10	10
10	10	10
100	100	100

$-50+80-300$

$= -300+30 = -270 = \frac{-270}{9} = -30$ → pagan

10	10	10
100	100	100
100	100	100

⇒ $-30+800-500$

⇒ $300-30 = \frac{270}{9} = 30$

$-270 \rightarrow 0$ & $270 \rightarrow 255$

0 0 0 0 → 30

0 255 255 255

Mask →

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

scaling factor

~~10~~ 0 - 100 -

10	10	10	10	10	10	10	10	10	10
10	0	0	0	0	0	0	0	0	10
10	0	0	0	0	0	0	0	0	10
10	-90	-90	-90	-90	-90	-90	-90	-90	10
100	90	90	90	90	90	90	90	90	100
100	0	0	0	0	0	0	0	0	
100	0	0	0	0	0	0	0	0	
100									

$-90 \rightarrow 0$

10	10	10
10	10	10
100	100	100

$-10+40-10-10-100$

10-100

10	10	10
100	100	100
100	100	100

$-10-300+400$

100-10

~~10~~

High Boost Filter

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & x & -1 \\ \hline -1 & -1 & -1 \\ \hline\end{array}$$

Here

$x = 9A - 1$

Multiplicative constant

if $A = 1.1$

$x = 8.9$

10	10	10	10	10	10	10	10	10	10	
10	1	1	1	1	1	1	1	1	10	
10	1	1	1	1	1	1	1	1	10	
10	~~261~~ -29	~~261~~ -29	~~261~~ -29	-29	-29	-29	-29	-29	10	
10										
100	40	40	40	40	40	40	40	40		
100										
100										
100	10	10	10	10	10	10	10	10		
100										

10 10 10
10 10 10
10 10 10

$-80 + 89 = 9 \times \frac{1}{9}$

e.g. 9×10

10 10 10
10 10 10
100 100 100

$-50 - 300 + 89$

-350
89
$\frac{-261}{9} = 29$

10 10 10
100 100 100
100 100 100

$890 - 30 - 500$

390
-30
$\frac{360}{9} = 40$

~~10+~~

$-800 + 890 = \frac{90}{9} = 10$

1.5

$\frac{~~-80~~ + 98 = 18}{9} = 2$

$-80 + 125$

125
80
$\frac{45}{9} = 5$

$-50 - 300 + 125$

$1250 - 30 - 500$

1250
-530
$\frac{720}{9}$ 80

$-800 + 1250$
800

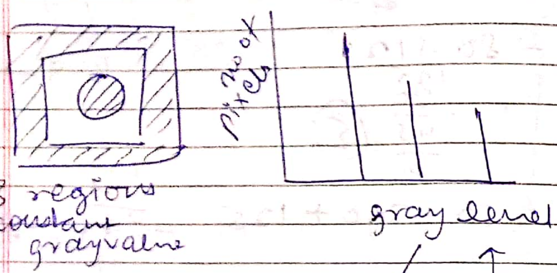
$\frac{450}{9}$ 50

1.5
×9
$13.5 - 1 = 12.5$

1.2
×9
$10.8 - 1 = 9.8$

350
125
$\frac{-225}{9} = -25$

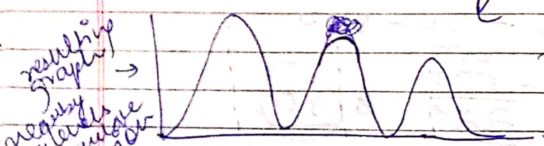
Gaussian filter / smoothing filter

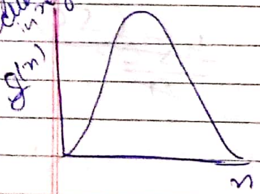


3 regions of constant grayvalue

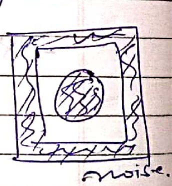
gray level

Gaussian noise introduced.





$$g(n) = e^{-\frac{n^2}{2\sigma^2}}$$



Discrete Gaussian distribution

$$g[x,y] = c\, e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

$$\frac{g[x,y]}{c} = e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

then we calculate mask

size of mask $n=7$
& let $\sigma^2 = 2$

	-3	-2	-1	0	1	2	3
-3	0.011	0.038	0.082	0.105	0.082	0.038	0.011
-2	0.038	0.135	0.286	0.368	0.286	0.135	0.038
-1	0.082	0.286	0.606	0.778	0.606	0.286	0.082
0	0.105	0.368	0.778	1.0	0.778	0.388	0.105
1	0.082	0.286	0.606	0.778	0.606	0.286	0.082
2	0.038	0.135	0.286	0.368	0.286	0.135	0.038
3	0.011	0.038	0.082	0.105	0.082	0.038	0.011

$$g[3,3] = \frac{g[-3,-3]}{c} = e^{-\frac{(9+9)}{2(2)}} = e^{-\frac{9}{2}} = 0.011$$

$$g[-2,-2] = g[2,2] = e^{-\frac{4}{2}} = 0.135$$

$$g[1,1] = e^{-\frac{2}{2(2)}} = 0.606$$

$$g[0,0] = e^{-0} = 1$$

$$g[3,2] = e^{-\frac{(9+4)}{2(2)}} = e^{-\frac{13}{4}}$$

$$g[1,3] = e^{-\frac{(9+1)}{4}} = e^{-10/4} = e^{-5/2} = 0.082$$

$$g[1,2] = e^{-\frac{(4+1)}{4}} = e^{-5/4} = 0.286$$

$$g[0,1] = e^{-\frac{(0+1)}{4}} = e^{-1/4} = 0.778$$

$$g[0,2] = e^{-4/4} = e^{-1} = 0.368$$

$$g[0,3] = e^{-9/4}$$

Scanned by CamScanner

Now I want integers instead of real nos.

so $\frac{g[3,3]}{c} = 0.011$

we want $g[3,3] = 1$

so $\boxed{C = \frac{1}{0.011}}$ Corners should be 1

	-3	-2	-1	0	1	2	3
-3	1	3	7	9	7	3	1
-2	3	12	26	33	26	12	3
-1	7	26	55	70	55	26	7
0	9	33	70	90	70	33	9
1	7	26	55	70	55	26	7
2	3	12	26	33	26	12	3
3	1	3	7	9	7	3	1

divide by a constant

n = sum of all elements of mask then it will retain characters of filter.

$\frac{1}{1098}$

Averaging filter gives better result but blur imgs

v/s

smoothing filter gives more smooth transition b/w pixel values.

if no gaussian noise, then it can be used just as smoothing filter.

$\sigma^2 = 1, \; n = 5$

	-2	-1	0	1	2
-2	0.018	0.082	0.135	0.082	0.018
-1	0.082	0.368	0.606	0.368	0.082
0	0.135	0.606	1.0	0.606	0.135
1	0.082	0.368	0.606	0.368	0.082
2	0.018	0.082	0.135	0.082	0.018

$e^{\frac{[0+0]}{2}} = e^0 \qquad e^{-\frac{[1+1]}{2}} = e^{-\frac{2}{2}} = 0.368$

$c = \frac{1}{0.018}$

	-2	-1	0	1	2
-2	1	4	7	4	1
-1	4	20	33	20	4
0	7	33	55	33	7
1	4	20	33	20	4
2	1	4	7	4	1