Date distribution of gray values

Histogram Equalisation : for digital images

$$S = CDF = (L-1)\sum_{i=0}^{2^n-1} P(r_i)$$

CDF

* stretches contrast automatically, no need to know values for formula

$$f(x,y) = \begin{array}{|c|c|c|} \hline 10 & 20 & 15 \\ \hline 30 & 15 & 20 \\ \hline 20 & 20 & 30 \\ \hline \end{array}$$

3x3

small pic so only 4 shades

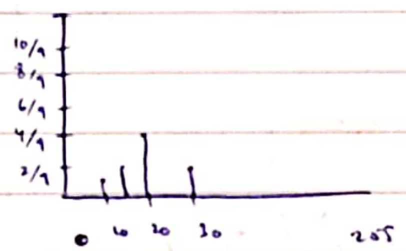
• high Mega Pixel image will have all gray values

gray shades

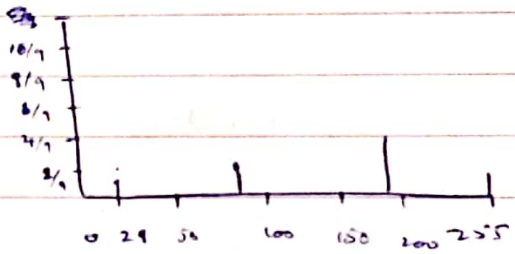
$(L-1)\sum_{i=0}^{2^n-1} P(r_i)$

a_i	$P(a_i)$	CDF	V (round off values of CDF)
10	1/9	$255 \times 1/9 = 28.33$	28
15	2/9	$255(\frac{1}{9}+\frac{2}{9}) = 85$	85
20	4/9	$255 \times (\frac{1}{9}+\frac{2}{9}+\frac{4}{9}) = 198.33$	198
30	2/9	$255 \times (\frac{2}{9}+\frac{1}{9}+\frac{2}{9}+\frac{4}{9}) = 255$	255

$$g(x,y) = \begin{array}{|c|c|c|} \hline 28 & 198 & 85 \\ \hline 255 & 85 & 198 \\ \hline 198 & 198 & 255 \\ \hline \end{array}$$



input



output

a_i	$P(a_i)$	CDF	V
28	1/9	$255 \times \frac{1}{9} = 28.33$	28
85	2/9	$255 \times \frac{2+\ldots}{9} = 85$	85
198	4/9	$255 \times \frac{4+\ldots}{9} = 198.33$	198
255	2/9	$255 \times \frac{2+\ldots}{9} = 255$	255

equalised histogram will remain same

differences

Contrast stretching : semi automatic - have to give points - reversible process

Histogram Equalization: fully automatic - irreversible

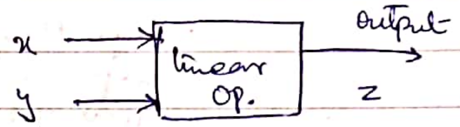
Date

Points Ops ⟶ Neighborhood-based Image Enhancement

- Neighbourhood-based Operations:

(i) Linear vs. Non-Linear Ops.

Non-Linear ⟶ unpredictable behavior



$Z = \alpha x + \beta y$

$S = T(r)$

(i) Homogenity Property

$$f(x) + f(y) = f(x+y)$$

(ii) Multiplicity: Property

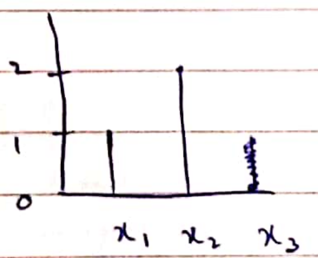
$$\alpha \cdot f(x) = f(\alpha \cdot x).$$

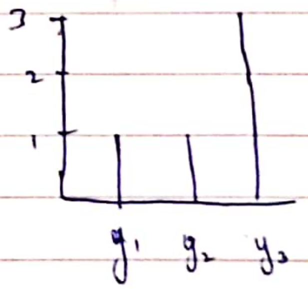
If some operation satisfies both these properties then Linear function, otherwise Non Linear function.

⟶ combined eq.

$$\alpha \cdot T(X+Y) = T(\alpha \cdot x) + T(\alpha \cdot y)$$

(mean will always satisfy this eq.)





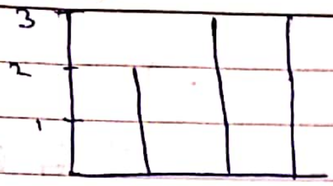
X [1, 2, 0]   Y [1, 1, 3]

Example Operations: (f)

(i) Mean Values

(ii) Median Values

Date



$X+Y$

$[2, 3, 3]$

Median

$5 \cdot \text{Median}(X+Y) = \text{Med}(5 \cdot x) + \text{Med}(5 \cdot y)$

sort first ($x = [0\ 1\ 2]$ $y\ [1\ 1\ 3]$)

$5 \times 3 = 5 + 5$

$15 \neq 10$ ✗

Mean

let $\alpha = 5$

$5 \cdot \text{Mean}(X+Y) = \text{Mean}(5 \cdot x) + \text{Mean}(5 \cdot y)$

$$5 \cdot \left(\frac{8}{3}\right) = \frac{5+10+0}{3} + \frac{5+5+15}{3}$$

$$\frac{40}{3} = \frac{15}{3} + \frac{25}{3}$$

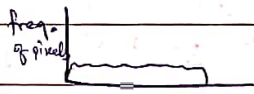
$$\frac{40}{3} = \frac{40}{3}$$ ✓

Non-Linear

linear.

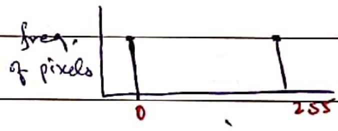
They are good

- linear ops. are more suitable when there's large amount of noise in the input signal with a smaller magnitude.



Noise

- Non-Linear are suitable for small amount of noise with large magnitude



salt + pepper noise

* in convolution & corelation we do sum of products but in convolution, ~~we flip the filter~~ by 180°

Date October 28th, 2021

Neighborhood based Operations

1. Linear vs. Non-linear Ops.
2. Spatial Filtering (i) Spatial Corelation Filter, (ii) Spatial Convolution Filter — for linear operations

Spatial Filtering

1D filter $w(x) = a \quad b \quad c$ — if 1D, odd; if 2D, odd

2D filter $w(x,y) =$

a	b	c
d	e	f
g	h	i

$f(x) =$ | 1 | 2 |

1D corelation filter

1D $g(x) = \sum_{s=-1} \underbrace{w(s)}_{\text{filter value}} \underbrace{f(x+s)}_{\text{signal}}$

1D convolution filter

1D $g(x) = \sum_{s=-1}^{1} w(s) \cdot f(x-s)$

$w(s,t)$ filter

t, y →

s, x ↓

-1,-1	-1,0	-1,1
0,-1	x,y 0,0	0,1
1,-1	1,0	1,1

3 × 3

* output of filter will always be in the center

vals = -1, 0, 1 so $\sum_{-1}^{1}$

2D corelation filter

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} f(x+s, y+t)\, w(s,t)$$

$s=-a$ ↓ -1 to 1; $t=-b$ ↓ -1 to 1

$$\underset{(50,45)}{g(x,y)} = f(50+(-1), 45+(-1)) \cdot w(-1,-1) + f(49,45)\, w(-1,0) + \dots$$

$$\dots f(51,46) \cdot w(1,1)$$

- filter is always odd matrix

Date

pixel 1 pixel 7

$f(x) = 0\ 0\ 0\ 1\ 0\ 0\ 0$

center / response is always at center

$w(s) = 1\ 2\ 3\ 4\ 5$

1D signal -2 -1 → padded values, loc.0, loc.1, 2, 3, 4, 5, 6

0 0 0 0 0 1 0 0 0 apply filter on each pixel

1 2 3 4 5

$g(0) = w(-2)\cdot f(-2) + w(-1)\cdot f(-1) + \dots w(2)\cdot f(2)$

$0\times 1 + 0\times 2 + 0\times 3 + 0\times 4 + 0\times 5 = 0$

now move filter 1 step to right

-2 -1 0 1 2

0 0 0 0 1 0 0 0

1 2 3 4 5

$0\times 1 + \dots 1\times 5 = 5$

now filter moved 1 step to right

0 0 0 1 0 0 0

1 2 3 4 5 = 4

0 0 0 1 0 0 0

1 2 3 4 5 = 3

0 0 0 1 0 0 0 0 0

0 0 ~~1~~ ~~2~~ 1 2 3 4 5 = 0

We get 0 5 4 3 2 1 0 ← reverse of filter

impulse shows exactly the output of filter

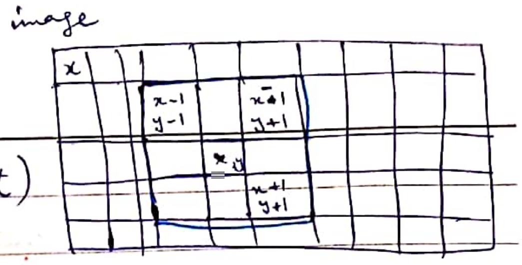
~~filter is flipped 180°~~

$w(s) = 12345$ convolution filter will give the same result as filter

output = 0123450

because we did correlation and simply × filter and

Date



$$g(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} f(x+s, y+t) \cdot w(s,t)$$

pixel value at location (x,y) of the output

-1,-1 -1,0 -1,+ 0,+

$f(x,y) =$

15	15	20	20
15	15	20	20
25	25	30	30
25	25	30	30

g(0,0) center

g(0,1) center

g(1,1) center

g(1,0) center

padding pixel replication

$w(s,t) =$

1	2	3
4	5	6
7	8	9

$g(0,0) = 15 \times 1 + 15 \times 2 + 20 \times 3 + 15 \times 4 + 15 \times 5 + 20 \times 6 + 25 \times 7 + 25 \times 8 + 30 \times 9 = 1,005$

$g(0,1) =$

$g(1,0) =$

$g(1,1) =$

Average Filter:

$f(x,y)$

a	b	c
d	e	f
g	h	i

$g(x,y) = \frac{1}{9}a + \frac{1}{9}b + \frac{1}{9}c + \frac{1}{9}d + \frac{1}{9}e + \frac{1}{9}f + \frac{1}{9}g + \frac{1}{9}h + \frac{1}{9}i$

Linear Filter represented in sum of Product form.

$w(s,t) =$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

symmetric filter

Date

$$f(x,y) = \begin{array}{|c|c|c|c|} \hline 15 & 15 & 20 & 20 \\ \hline 15 & 15 & 20 & 20 \\ \hline 25 & 25 & 30 & 30 \\ \hline 25 & 25 & 30 & 30 \\ \hline \end{array} \qquad w(s,t) = \begin{array}{|c|c|c|} \hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline \end{array}$$

* if symmetric, corelation and convolution will give same results

$$g(0,0) = 15 \times \frac{1}{9} + 15 \times \frac{1}{9} + 20 \times \frac{1}{9} + 15 \times \frac{1}{9} + 15 \times \frac{1}{9} + 20 \times \frac{1}{9} + 25 \times \frac{1}{9} + 25 \times \frac{1}{9} + 30 \times \frac{1}{9}$$

$$= 20$$

$$g(0,1) = 15 \times \frac{1}{9} + 20 \times \frac{1}{9} + 20 \times \frac{1}{9} + 15 \times \frac{1}{9} + 20 \times \frac{1}{9} + 20 \times \frac{1}{9} + 25 \times \frac{1}{9} + 30 \times \frac{1}{9} + 30 \times \frac{1}{9}$$

$$= 22$$

$g(1,0) = 23$

$g(1,1) = 25$.

$g(x,y)$

20	22
23	25

↓ less contrast

$f(x,y)$

15	20
25	30

↓ more contrast

same entropy

* when pixels' difference ↓ contrast ↓ blurring ↑

1/25	1/25	1/25	1/25	1/25
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
1/25	.	.	.	1/25

5x5 averaging filter

$$\text{padding} = \frac{n-1}{2}$$

n → width of filter

padding → $\boxed{\text{0 or replicate values}}$

if 7x7 $\;1/49 \;\; 1/49 \; \dots$

* avg filter width ↑ blurring ↑

Date November 2nd, 2021

r	s	t
u	v	w
x	y	z

x, y

a	b	c
d	e	f
g	h	i

w filter
linear
mean, median

corelation $g(x,y) = r \times a + s \times b + t \times c \ldots \ldots + v \times e + \ldots z \times i$

convolution $g(x,y) = r \times i + s \times h + t \times g + u \times f + v \times e + w \times d + \ldots + z \times a$

180° flip
so output is
same. by default
it gives inverse of input which is
same as corelation

Linear Filters: does smoothing

$\boxed{\text{Mean Filter}}$

1/9	1/9	"
"	"	"
"	"	1/9

- operation that can be represented as sum of products

- take equal portions

0	255	255
255	240	255
255	255	0

Mean Filter is taking equal portion of all pixels but we dont want equal part of noise i.e. 0.

x effected most

$\boxed{\text{Weighted mean filter}}$

use when distribution of noise is minute

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

- better than Mean Filter when there's less noise

WMF giving less importance to noise i.e. 0

• smoothing brings values closer
contrast ↓ blur ↑

Date

0.025	0.1	0.025
0.1	0.5	0.1
0.025	0.1	0.025

× if noise, lesser problem, lesser portion/effect

better WMF
max weight so it has more effect.

Unsharp Masking

→ sharpen $f(x,y)$ input image
increase distance in neighborhood values.

Steps: (1) Generate a smooth image $f'(x,y)$ from the input image $f(x,y)$ by using smoothing filter.
e.g. Mean Filter, Gaussian Filter, WMF.

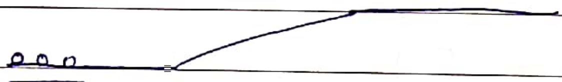
(2) Generate an unsharping mask $M(x,y)$ by subtracting $f'(x,y)$ from $f(x,y)$

(3) Add the mask $M(x,y)$ in the original image to get the sharp image $g(x,y)$.

$$g(x,y) = f(x,y) + \alpha \cdot M(x,y) \qquad \alpha \geq 0 \; ; \; \alpha = 1$$

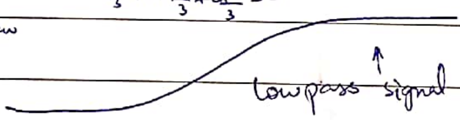
$\boxed{1D}$

$f(x) =$



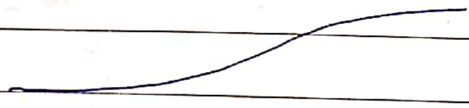
$\rightarrow 0 \times \frac{1}{3} + 0 \times \frac{1}{3} + 0 \times \frac{1}{3} = 0$

(1) $f'(x) =$



lowpass signal $= 3$

increasing filter will straighten the line/slope



$= 5$

(2) $f(x,y) - f'(x,y)$

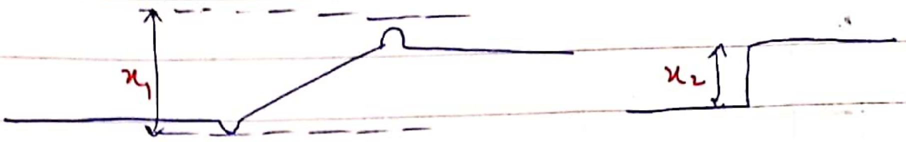
$M(x) =$

~~lowpass~~ highpass signal

Date

* increase in max + min distance

③ $g(x) =$



$x_1 > x_2$

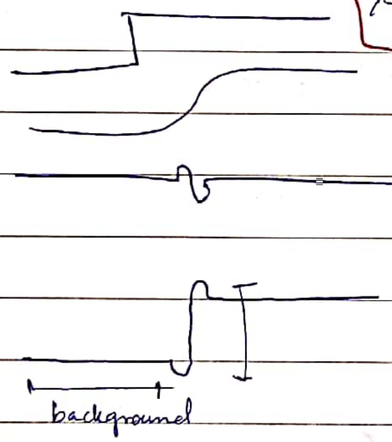
Example :

$f(x,y) = \begin{bmatrix} 15 & 20 \\ 25 & 30 \end{bmatrix}$

① $f'(x,y) = \begin{bmatrix} 20 & 22 \\ 23 & 25 \end{bmatrix}$

② $M(x,y) = \begin{bmatrix} -5 & -2 \\ 2 & 5 \end{bmatrix}$

③ $g(x,y) = \cancel{\begin{bmatrix} 20 & 22 \\ 23 & 25 \end{bmatrix}} \quad \begin{bmatrix} 10 & 18 \\ 27 & 35 \end{bmatrix}$



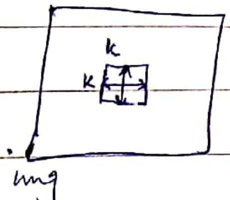
0 - 256

Slide 19

$k = \frac{(m-1)}{2}$

width size of object in img.

if this filter size then affect will be on m pixels



objects of k size or below will be removed

if size of obj. is 50

$m = 2k+1$

$m = 2(50)+1$

$m = 101$

101×101 filter size

Date

Non-Linear Filters:

① Median Filter

matrix 1

255	255	255
200	0	255
255	240	210

→ noise

sort 0, 200, 210, 240, 255, 255, 255, 255, 255

Median = 255

- will give output pixel from the image
- sharpens
- ↑ distance b/w pixel values

Mean = 213.8 ≈ 214

- will give output not found in the pixels of image
- smoothes
- retains original info of image

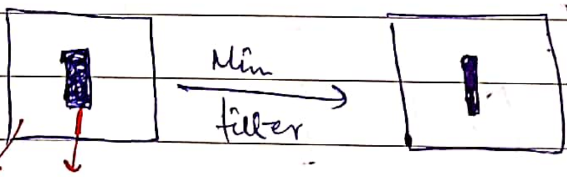
$f(x,y)$ =

15	20
25	30

$f_{median}(x,y)$ =

② Min Filter

- output min in matrix 1. i.e. 0
- if image has pepper noise (randomly 0's), min filter will not remove that noise
- it will do erosion



3x3 = erode 1 px
5x5 = erode 2 px

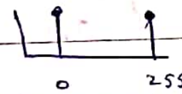
③ Max Filter

- output max in matrix 1 i.e. 255.
- does dilation

* if bg light, fg dark (opposite) then min will do dilation and max will do erosion.

* salt + pepper

salt → random white pixels; pepper → random black pixels



Date: November 4th, 2021.

Quiz Sol.

Q filter $m \times m$ img. $f(x,y)$. eq. for convolution?

$$g(x,y) = \sum_{s=-\frac{(m-1)}{2}}^{\frac{(m-1)}{2}} \sum_{t=-\frac{(m-1)}{2}}^{\frac{(m-1)}{2}} f(x-s,\, y-t) \cdot w(s,t)$$

Median Filter

- Before applying Neighborhood, decide size of it. 3x3 or 5x5 . . . ?

if Sir doesn't give then assume size.

- 3x3 Neighborhood filter

a	b	c
d	e	f
g	h	i

3x3

$$f(x,y) = \begin{bmatrix} 15 & 20 \\ 25 & 30 \end{bmatrix}$$

↓

do padding

0 padding not good so replicate

$$\begin{bmatrix} 15 & 15 & 20 & 20 \\ 15 & 15 & 20 & 20 \\ 25 & 25 & 30 & 30 \\ 25 & 25 & 30 & 30 \end{bmatrix}$$

$$g(x,y) = \begin{bmatrix} 20 & 20 \\ 25 & 25 \end{bmatrix}$$

↓

2 extremes

2 ranges observed

this helps in sharpening

(e.g. in slides)

sort then pick

$15, 15, \cancel{20}\ 15, 15, \underbrace{20}_{\text{median}}, 20, 25, 25, 30$

Date

Min Filter: • erosion when bg black, fg white

• e.g. lungs overlap with heart, do this filter, overlapping will be eroded.

3 x 3 — 1 pixel eroded

5 x 5 — 2 pixel eroded

7 x 7 ⋮ — ⋮ 3

Max filter: • dilation when bg black, fg white

3 x 3 — 1 pixel dilated

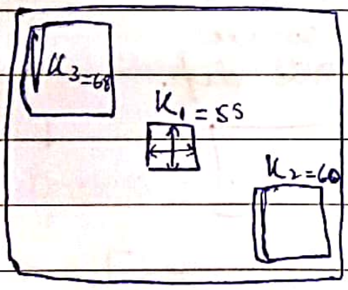
5 x 5 — 2

⋮ ⋮

0 — black

1 — light

$$f(x,y) = \begin{matrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&1&0&0\\ 0&0&1&1&1&0\\ 0&0&1&1&1&0\\ 0&0&1&1&1&0\\ 0&0&0&0&0&0 \end{matrix} \xrightarrow[\text{Filter} \cdot \text{erosion}]{\text{Min}} \begin{matrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0 \end{matrix}$$



we want to smooth this image

$m = 2k + 1$

$m = 2(55) + 1$

$m = 111$ → this will remove K_1 and erode K_2 and K_3.

new $K_2 = 60 - 55 = 5$.

new $K_3 = 68 - 55 = 13$.

[Yes] → [No] →

Neighborhood Ops.
- sharpening
- edge detection
- differentiation
- derivation

Date November 8th, 2021.

Image Sharpening / Edge Detection using Derivatives :

* Smoothing brings values closer
contrast ↓ blur ↑

1. First order derivative
2. Second order derivative

Raw profile of a 3-bit image:

First Order Derivative

Case 1. gives a zero output at smooth intensity regions. → 7 7 7 7 7
Case 2. gives a non zero value along Ramps → 7, 6, 5, 4, 3, 2, 1
Case 3. gives a non zero value at the onset of ~~along~~ steps or isolated points → 0 0 0 7 7 7

ramp

ramp

1D, change in row only

use this for 1st o.d.

$$\boxed{f'(x) = f(x+1) - f(x)}$$

centered at x so we take this def.

Case 1

7 7 7 7 7

$f'(x) = f(x+1) - f(x)$

$= 7 - 7 = 0$.

Case 3:

0 0 0 7 7 7

$= 7 - 0 = 7$.

0 0 0 7 7 7

0 0 7 0 0 0

Case 2:

7 6 5 4 3 2 1

$= f(x+1) - f(x)$

$= 5 - 6 = -1$

Date more derivatives ...

$f'(x) = f(x) - f(x-1)$ ← centered at $x-1$

$f'(x) = f(x+2) - f(x+1)$ ← centered at $x+1$

Second Order Derivative

Case 1 : gives 0 output at smooth intensity regions → 7 7 7 7 7

Case 2 : gives a 0 value along ramps.

$\boxed{\text{Case 1}}$

	7	7	7	7	7
$f'(x)$	0	0	0	0	
$f''(x)$	0	0	0	0	

$\boxed{\text{Case 2}}$

	7	6	5	4	3	2	1
$f'(x)$	-1	-1	-1	-1	-1	-1	
$f''(x)$	0	0	0	0	0	0	0

Case 3 : gives a non zero value at onset of a step or isolated points

$\boxed{\text{Case 3}}$

	0	0	0	7	7	7
$f'(x)$	0	0	7	0	0	0
$f''(x)$	0	7	-7	0	0	0

$$f'(x) = f(x+1) - f(x)$$

$$f''(x) = f'(x+1) - f'(x)$$

$$f''(x) = f(x+2) - f(x+1) - [f(x+1) - f(x)]$$

$$f''(x) = f(x+2) - 2f(x+1) + f(x)$$

centered at $x+1$

$$\boxed{f''(x) = f(x+1) - 2f(x) + f(x-1)}$$

← put x in place of $x+1$ to center the derivation at loc. x

use this eq. of 2nd o.d.

- edges in images are ramps.

Date

	Ramp								Step					Thin line	
Example:	5	5	4	3	2	1	0	0	7	7	7	0	0	5 7 5	

$f'(x)$: 0 −1 −1 −1 −1 −1 0 7 0 0 −7 0 5 2 −2 0

(smooth edges) (step) (thin line)

$f''(x)$: −1 0 0 0 0 1 7 −7 0 −7 7 5 −3 −4 2 0

(smooth edges) (step) (thin line)

1st O.D. edges thick
2nd O.D. 0 on edges, nonzero on start and end of edges

Property ① 1st O.D. gives you thick edges (non-zero values), while 2nd O.D. gives you thin edges. (0 value)

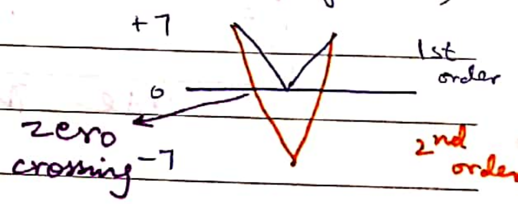
P ② 2nd O.D. has double response to gray level steps than 1st O.D.

P ③ For isolated points, the 2nd O.D. shows double response (zero crossing) while 1st O.D. shows single response.

	isolated						
	0	0	0	7	0	0	0
$f'(x)$	0	0	7	−7	0	0	0
$f''(x)$	0	7	−14	7	0	0	0

- 2nd O.D. is sensitive to noise.

responses at isolated point (e.g. noise)



* when you have noisy images, don't apply 2nd O.D. directly. First remove noise then apply 2nd O.D. because otherwise too much fluctuations b/w +ve and −ve sides.

- 2nd shows sharp responses than 1st.

Date

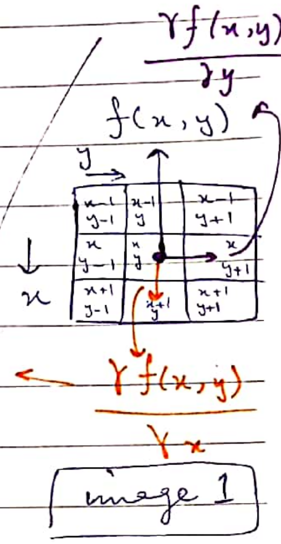
First O.D. on image (2D):

$\cancel{f(x,y) = f(x+1, y+1) - f(x,y)}$

$$f'(x,y) = \underbrace{\frac{\partial f(x,y)}{\partial x}} + \underbrace{\frac{\partial f(x,y)}{\partial y}}$$

$\rightarrow = f(x+1, y) - f(x,y) +$

$\rightarrow \quad f(x, y+1) - f(x,y)$



$$f'(x,y) = f(x+1, y) + f(x, y+1) - 2f(x,y)$$

check coefficient in this eq.

Q. make a filter for $f'(x,y)$

$$f_{mean}(x,y) = \frac{1}{9} f(x-1, y-1) + \frac{1}{9} f(x-1, y) + \dots \frac{1}{9} f(x+1, y+1)$$

$$f'(x,y) = 0 \times f(x-1, y-1) \dots + 1 \times f(x+1, y) + -2 f(x,y)$$

0	0	0
0	-2	1
0	1	0

1st order derivative

* when you apply this filter on image 1, 3 cases of 1st O.D. will be followed.

2nd O.D. on image (2.D):

$$f''(x,y) = \underbrace{\nabla^2}_{\text{Laplacian operator}} f(x,y) = \frac{\partial^2 f(x,y)}{\partial x} + \frac{\partial^2 f(x,y)}{\partial y}$$

$$f''(x,y) = f(x+1, y) - 2f(x,y) + f(x-1, y) + f(x, y+1) - 2f(x,y) + f(x, y-1)$$

Scanned with CamScanner

corelation = sum of products
convolution = difference of products

Date

$$f''(x,y) = f(x+1,y) - 4f(x,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)$$

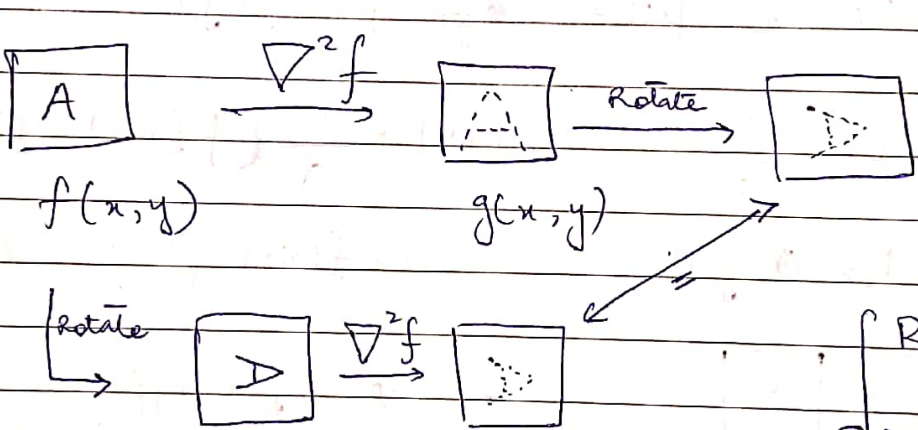
• Filter

$$\nabla^2 f = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

2nd O.D.

image 1.

• Laplacian is ~~Isotropic Filter~~ → ~~variation in rotation~~ rotation invariant filters



Rotate → $\nabla^2 f$
or
$\nabla^2 f$ → Rotate

Same result because rotate is isotropic filter

$$\nabla^2 f = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

improved def. of laplacian / 2nd O.D.

in laplacian, we dont ~~only consider 4 neighbours~~ but 8 neighbors.

Date **Digital Image Processing II**

November 11^{th}, 2021.

- thick edges
- single response

∇f 1st O.D. =

0	0	0
0	-2	1
0	1	0

$\nabla^2 f$ 2nd O.D. =

- thin edges
- double response

0	1	0
1	-4	1
0	1	0

Image Sharpening: using Laplacian

obj: 7 7 7 7 7 7

bg

$f(x)$ 0 0 0 0 0

$f'(x)$ 0 0 0 0 7 0 0 0 0 0 0

smooth edges so 0

Sharpening: increase ~~values~~ difference b/w values

$f''(x) = 0\ 0\ 0\ 7\ \ -7\ 0\ 0\ 0\ 0\ 0\ 0$

-ve cuz center is -ve

1D signal $\boxed{f_{sharp}(x) = f(x) - \nabla^2 f(x)}$

$f_{sharp}(x) = 0\ 0\ 0\ -7\ \ \cancel{14}\ 7\ \ 7\ 7\ 7\ 7\ 7\ 7$

$f' = f - f_{min}$

$f_N = \dfrac{f'}{f'_{max}} \times 255$ — if 8 bit

~~xx~~ normalize like in past. cuz $f_{sharp}(x)$ has -7 value, can't be in -ve.

normalize 7 7 7 0 14 14 14

$\nabla^2 f$ =

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

1st O.D. can be used for sharpening as well

1D signal $\boxed{f_{Sharp}(x) = f(x) + \nabla^2 f(x)}$

- add to increase distance b/w values
- cuz center is +ve

November 18th, 2021

Date

* 1^{st} O.D. we look at 0° and 270°

1st order derivative variants

Q Write a filter of 1st O.D. that gives horizontal discontinuities

$$f'(x) = f(x+1) - f(x) \quad 1D$$

$$f(x+1, y) - 2f(x,y) + f(x, y+1) \quad 2D$$

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & -1 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad \xrightarrow{y} \qquad \frac{\partial f(x,y)}{\partial y}$$

$$x \downarrow \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & -1 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \qquad \frac{\partial f(x,y)}{\partial x}$$

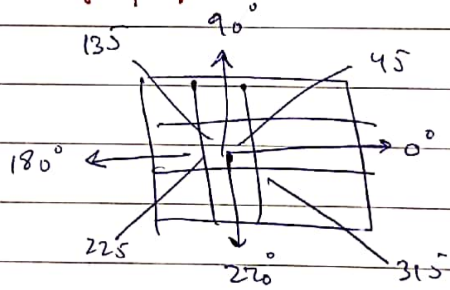
$$f'(x) = f(x+1) - f(x)$$

$$f'(x,y) = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}$$

$$= \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & -2 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$ → apply this on image, you'll get 'edge profile'; Thick edges

Extensions of 1^{st} O.D.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9



(1) Normal: $f'(x,y) = (Z_6 - Z_5) + (Z_8 - Z_5)$

$$\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & -2 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

(2) Roberts': depends on cross differences, Roberts proved that using this method contrast increases.

$f'(x,y) = (Z_9 - Z_5) + (Z_8 - Z_6)$

(3) Prewitt's operator:

$$\begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline & & \\ \hline -1 & -1 & -1 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 1 & & -1 \\ \hline 1 & & -1 \\ \hline 1 & & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 2 & 1 & 0 \\ \hline 1 & 0 & -1 \\ \hline 0 & -1 & -2 \\ \hline \end{array} = 2Z_1 + Z_2 + Z_4 - Z_6 - Z_8 - 2Z_9$$

3 directions contributing so better

it's better because it involves more neighbours, gives good edge profile.

* repitition of event in a ~~specified~~ unit time is freq.

* operators differ in direction of neighbors

Date

(4) Sobel : give more weight to 4-neighbors while calculating x and y, less weight to neighbors in diagonal.

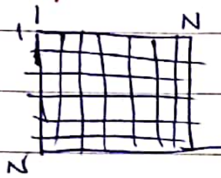
Then added together.

* Roberts' filter ?

0	0	0
0	-1	-1
0	1	1

Spatial Filtering in Frequency Domain :

pixels values being used as it is



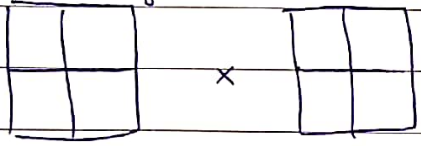
N^2 complexity

But after padding : 4 loops $O(N^4)$

spatial domain f

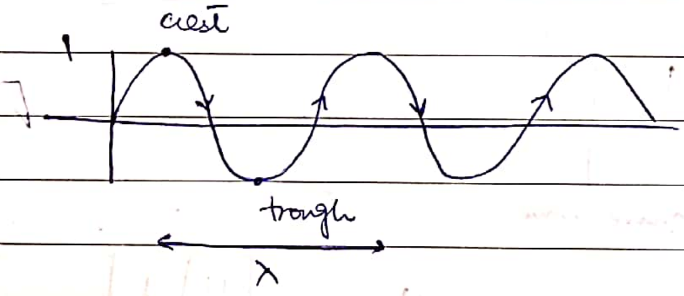
freq. domain F

$F(x,y)$ $\times$ $w(x,y)$ $\rightarrow$ $F'(x,y)$ $\rightarrow$ $g(x,y)$



N×N   M×M / N×N

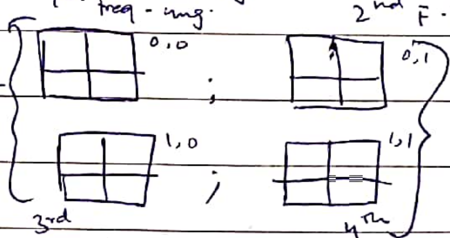
• change in values represent freq.



• varying sum can be represented as weighted sum.

• varying f wave will have many sines + cosines

$$I(r,c) = \begin{bmatrix} 5 & 3 \\ 1 & 2 \end{bmatrix} \; ; \; B(r,c,u,v) =$$



if 10×10 then 100 freq. images

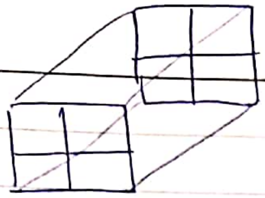
Frequencies / Basis Images

2D images that are orthogonal + orthonormal

1^{st} F.I. $B(r, c, 0, 0)$

Date

$I(r,c)$

$\underset{u\ v}{T(0,0)} =$ 

when multiplied with coefficients we get back original image

All pixels in input img. contribute to each value in output img. for freq. transforms

coefficient types

(i) Orthogonal (ii) Orthonormal

$$\boxed{T(u,v)} = \sum_{r=0}^{N-1} \sum_{c=0}^{M-1} I(r,c) \cdot B(r, c, u, v)$$

Forward transform equation for square waves

unique coefficient in Fourier Domain

Example: (0,0) $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ 0 freq.

original img. $\begin{bmatrix} 5 & 3 \\ 1 & 2 \end{bmatrix}$

(0,1) $\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$ square wave

(1,0) $\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$ vertical/inverted wave ?

(1,1) $\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ square wave

$T(0,0) = 5 \times 1 + 3 \times 1 + 1 \times 1 + 2 \times 1 = 11$

$T(0,1) = 5 - 3 + 1 - 2 = 1$

$T(1,0) = 5 + 3 - 1 - 2 = 5$

$T(1,1) = 5 - 3 - 1 + 2 = 3$

$T(u,v)$ $\begin{bmatrix} 11 & 1 \\ 5 & 3 \end{bmatrix}$

signal decomposed to get coefficients

Scanned with CamScanner