03 02/08/2023

Image Processing

Unit-1

- Levels of processing
- digital image

02/08/2023

Full stack Web development

Three layers of full stack

- Presentation layer
- Business logic layer
- Data link layer
  - database
  - mysql,

- provides user interface.
- Technologies used html, CSS & javascript.
- front-end.

- Back-end
- dynamically connects Presentation & data link layer.
- languages used php, python, java etc.

Semantic
- developer friendly
- searching optimization

Non-Semantic
- no searching optimization.

	Input		Output
Low level	image	→	image
Mid level	image	→	feature
High level	features	→	classified object

03/08/2023

- Image Processing has 12 steps

- Image Acquisition (acquire image from digital cameras)

- Image Enhancement (• manipulate the image • more subjective)

digitalization has 2 methods

digitalization of amplitude

Quantization

Sampling

digitalization of coordinate

input → fft (input) → Fourier Domain

Fourier Domain

Fourier transformation

fft (img)

- Restoration

- Color Image Processing

- Wavelets processing

Interpolation

- Compression → DCT (compression technique)

- Morphological Operation

IFFT ← Enhanced Image Fourier domain

- Segmentation

- feature extraction

- Image pattern Classification

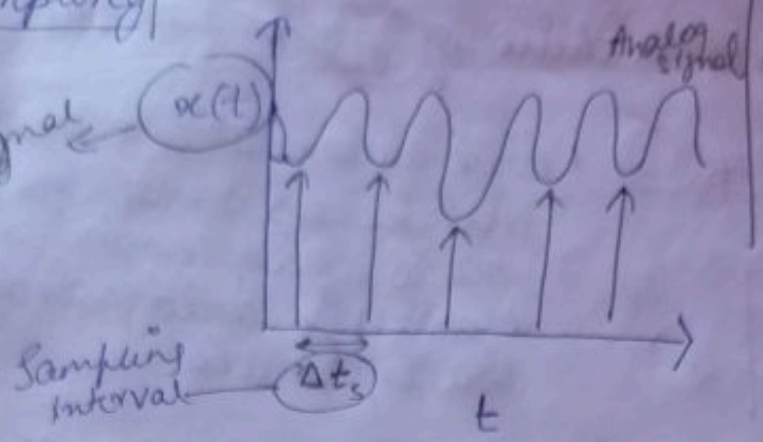
07/08/2023

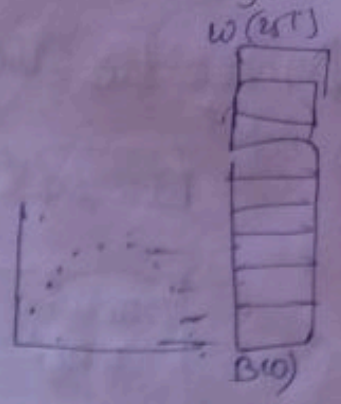
Image Processing

Topic discussed :-

- Components of Image Processing

Sampling





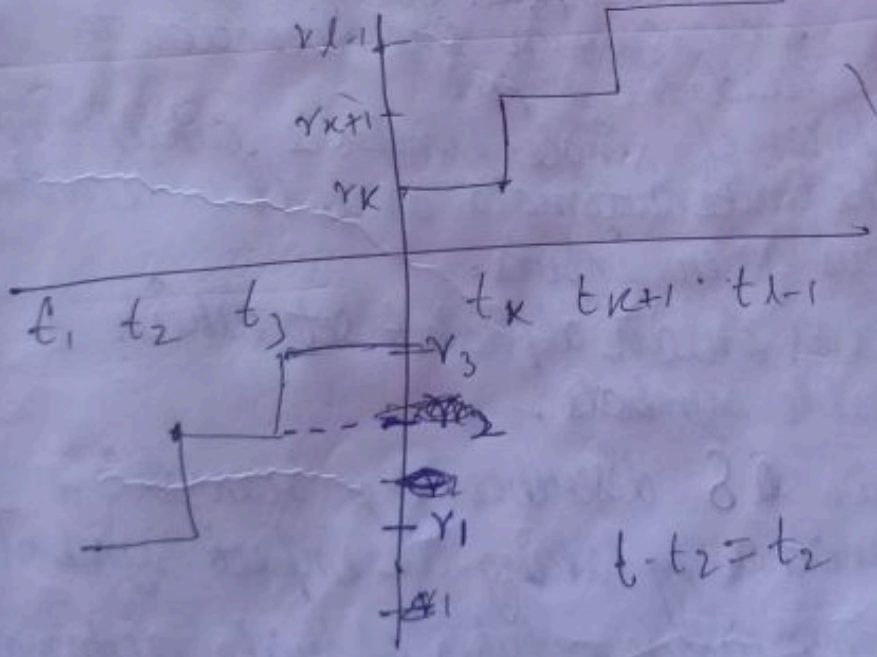
- $x(t)$ at certain discrete intervals of time Δt_s

08/08/2023

$$\text{Sampling Frequency} = \frac{1}{\Delta t_s}$$

Quantization ← uses staircase function.

$$M \times N = \begin{bmatrix} f(0,0), f(0,1) \cdots f(0,n-1) \\ f(1,0) \cdots f(1,n-1) \\ \vdots \\ f(m-1,0) \cdots f(m-1,n-1) \end{bmatrix}$$



$U = \{t_1, t_2, t_3, t_K, t_{K+1}, \ldots t_{L-1}\}$

$U' = \{r_1, r_2, r_3 \ldots r_{K+1}\}$

$t_1 \cdot t_2 = t_2$

09/08/2023 "Image processing"

Topic discussed :- Sampling & Quantization (staircase function)

09/08/2023 Full stack Web development

- MathML — 1998
  - ↳ based on XML

doesn't run on chrome only support some version of firefox or edge.

```
<html>
<body>
< Math xmlns = "http://www.w3.org/1998/Math/MathML">
```

<mrow> — to combine or bind subexpressions

<mi> — mathematical identifier

<mn> — mathematical number

<mo> — mathematical operator

<msup> → // superscript

<msub> → // subscript

<mfrac> → to write fraction

<mroot> → to write root

<msqrt> → to write square root

Q

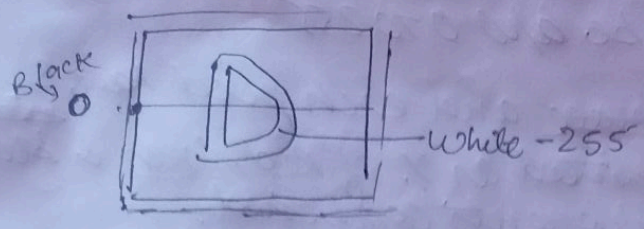
$x^2+y^2=4$

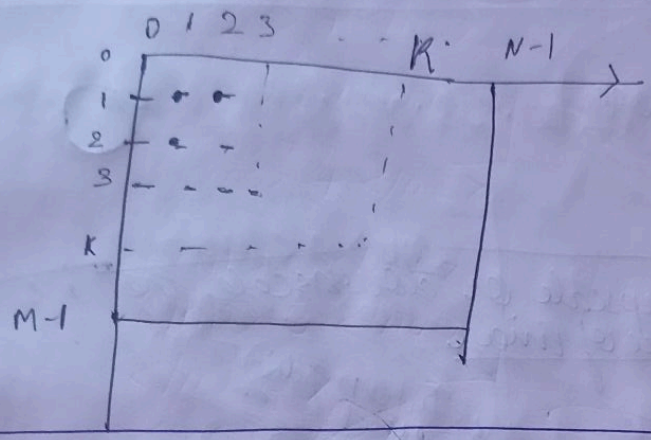
```
<mrow>
<msup>
```

<mo> + </mo>	<mn> =
<msup>	</mrow
<mi> y </mi>	

10/09/2023

Image Processing





Spatial and Intensity Resolution

- Smallest detail of an image
- In pixels per unit distance or dots per inch (dpi) used in printing and publishing industries

→ Smallest change in the intensity.
↳ represented in integer power of 2
↳ commonly 8 bits used.

Types of Images

↳ Binary images
- Each no. associated with the pixel can have only two values 0 and 1.

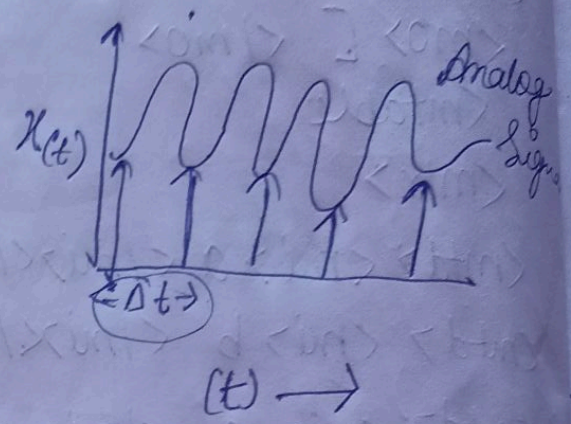
↳ Grayscale image
- All values b/w 0 to 255.
- 0 → B W → 255.

17/08/2023

"Image Processing"

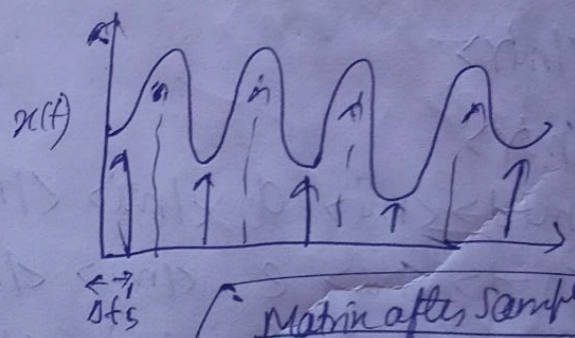
Topic :- Digitalization
↳ Sampling & Quantization

- $f(x, y)$
  - → sampling field (Quantization)
  - → Sampling
- (Δt) → Sampling time interval
- Sampling Frequency $f_s = \frac{1}{\Delta t_s}$



- Case: When we reduce the Sampling time interval Δt_s as $\Delta t's$ where $\Delta t's = \frac{\Delta t_s}{2}$

then we can improve image quality.



$$f's = \frac{1}{\Delta t's} = \frac{1}{\Delta t_s/2}$$

$$\boxed{f's = 2 \times f_s}$$

Matrix after Sampling

$$\begin{bmatrix} f(0,0) & \cdots & f(0,n) \\ f(1,0) & \cdots & f(1,n) \\ \vdots & & \\ f(m-1,0) & \cdots & f(m-1,n) \end{bmatrix}$$

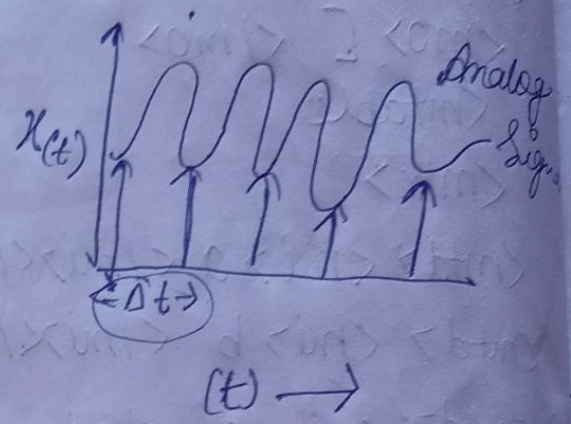
- Quantization :- To get discrete values in matrix after sampling, quantization is performed on the sampled image.
  - It reduces duplicate values.
  - Staircase function is used for quantization.

17/08/2023

"Image Processing"

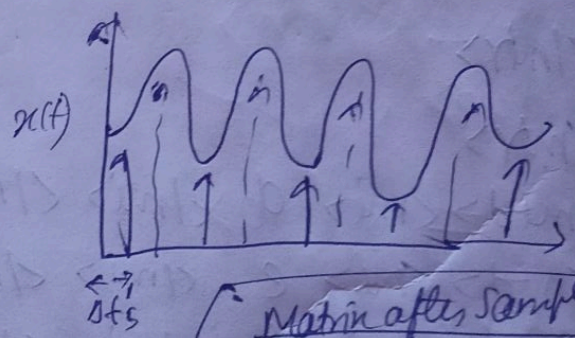
Topic :- Digitalization
↳ Sampling & Quantization

- $f(x, y)$
  - → sampling field (Quantization)
  - → Sampling
- (Δt) → Sampling time interval
- Sampling Frequency $f_s = \frac{1}{\Delta t_s}$



- Case: When we reduce the Sampling time interval Δt_s as $\Delta t's$ where $\Delta t's = \frac{\Delta t_s}{2}$

then we can improve image quality.



$$f's = \frac{1}{\Delta t's} = \frac{1}{\Delta t_s/2}$$

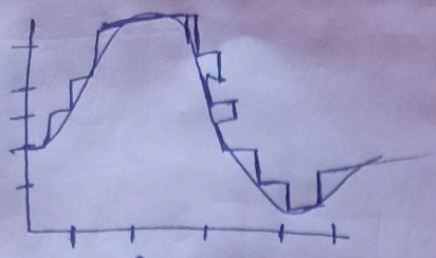
$$\boxed{f's = 2 \times f_s}$$

Matrix after Sampling

$$\begin{bmatrix} f(0,0) & \cdots & f(0,n) \\ f(1,0) & \cdots & f(1,n) \\ \vdots & & \\ f(m-1,0) & \cdots & f(m-1,n) \end{bmatrix}$$

- Quantization :- To get discrete values in matrix after sampling, quantization is performed on the sampled image.
  - It reduces duplicate values.
  - Staircase function is used for quantization.

- Staircase function



- Applying staircase function to sampled image
- Here any values b/w some range ~~same~~ will be taken same point thus forming a figure like staircase i.e like above.

21/08/2023 / Image Processing

Topic discussed :→

- Relationship & Connectivity of Pixels
- Types of Connectivity.

Relationship b/w Pixels

① 4 Neighbours
② diagonal neighbours

discussed in previous page.

③ 8 neighbours

$\hookrightarrow [N_4(P) \cup N_D(P)]$

- 8 neighbours of P are represented here as * in diagram

*	*	*
*	P	*
*	*	*

Connectivity :— used to find property of particular region of image.

If $f(x,y) >$ Threshold; (x,y) belongs to obj.
Else
(x,y) belongs to background

→ Two pixels are said to be connected if they are adjacent in some sense. which are :-

- They are neighbours
- Intensity value must be similar
- For binary image B two points p & q are connected if p belongs to N(q) or q belongs to N(p).

Types of connectivity

↳ 4 connectivity :- Intensity value of p and q must belong to q and q belongs to $N_4(P)$

↳ 8 connectivity :- Intensity value of p and q must belong to q and q belongs to $N_8(P)$.

↳ M connectivity :- Mixed connectivity

↓

Intensity value must be same and p and q doesn't belong.

a 1	b (P)	c (Q)
d 0	e 1	f 0
g 0	h 0	i 1

- $N_4(P) = \{a, c, e\}$

  ∴ q belongs to $N_4(P)$ so they are 4 connected

- $N_8(P) = \{a, c, e\}$

  q belongs to $N_8(P)$ so they are 8 connected

M conn.

- $N_4(P) \cap N_4(Q) = \phi$

No of common elements

Not M connected

$N_4(P) = \{a, c, e\}$

$N_4(q) = \{b\}$

$N_4(P) \cap N_4(q) = \phi$

⇒ They are M connected.

1	1	1
0	1	0
0	0	1

8 connectivity for b and e.

a 1	b 1	c 1
d 0	e 1 q	f 0
g 0	h 0	i 1

- $N_4(P) = \{a, c, e\}$

  ∵ q belongs to $N_4(P)$

  So they are 4 connected.

- $N_8(P) = \{a, c, e\}$

  ∵ q belongs to $N_8(P)$

  So they are 8 connected

- For M connectivity –

  $N_4(P) = \{a, c, e\}$

  $N_4(q) = \{b\}$

  $N_4(P) \cap N_4(q) = \phi$

  They are M connected

---

8 connectivity for c and e

a 0	b 1	c 1 p
d 0	e 1 q	f 0
g 0	h 0	i 1

- $N_4(P) = \{b\}$

  ∵ q doesn't belong to $N_4(P)$

  So they aren't 4 connected

- $N_8(P) = \{b, e\}$

  ∵ q ∈ $N_4(P)$

  they are 8 connected

- for M connected

  $N_4(P) = \{b\}$

  $N_4(q) = \{b\}$

  $N_4(P) \cap N_4(q) = \{b\}$

  So they are not m connected

Adjacency

If two pixels are connected, then they are said to be adjacent

Types of Adjacency

4-adjacency

8 "

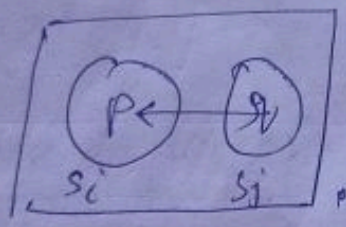
M "

23/08/2023

"Image Processing"

Adjacency :-

- If p and q are connected, then they are said to be adjacent.
- Types of Adjacency
  - ↳ 4 adjacency
  - ↳ 8 adjacency
  - ↳ M adjacency



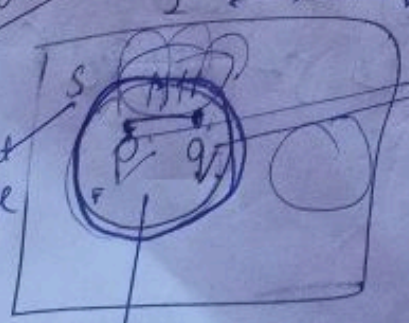
- $P \in S_i$
- $q \in S_j$
- Such that p and q are adjacent.

Digital Path

↳ A digital Path from $P(x,y)$ and $q(s,t)$ is a sequence of distinct pixels.

Connected Components

I → Image



Subset Image

pixels of Subregion of S.

- If there is only one component that component is connected, it is called a connected Set.

We can say p & q are connected if path from p & q must be connected with pixels of S.

- Region

- R → subset of pixels on an image
  R is region of image if R is connected set.
- 4 and 8 adjacency are considered when region were referred.
- Not connected regions are called disjoint regions.
-

1	1	0	R_i
1	0	0	
0	1 (p)	0	
0	0	1 (q)	R_j
1	1	1	
1	1	1	

- Union of these two regions are not possible as R_i and R_j p and q are not 4 adjacent.

Distance measures :-

- Suppose p(x,y), q(s,t) and z(m,n) are three pixels.
- Distance function (D) is valid if
  - $D(p, q) >= 0$
    ∀ any two points
  - $D(p, q) = 0$ iff $(p = q)$
  - $D(p, q) = D(q, p)$, symmetric
  - $D(p, z) <= D(p, q) + D(q, z)$

Consider two points p, q

① Euclidean distance

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

② City block Distance

$$D_4(p,q) = |x-s| + |y-t|$$

$D_4 = 1$ four neighbours of P

③ Chessboard distance

$$D_8(p,q) = \max(|x-s|, |y-t|) = 2$$

$D_8 = 1$ Eight neighbours of P.

|2|, |4|

(1,2) (3,4)

$\sqrt{(1-3)^2 + (2-4)^2}$

$|1-3| + |2-4|$

$|-2| + |-2|$

$2 + 2 = 4$

|2| |3|

Q

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Array/ Dot Product.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Matrix Product

$$\begin{bmatrix} 3 & 3 \\ 7 & 7 \end{bmatrix}$$

Linear V/s Non-Linear Operations

- Consider we have image $f(x,y)$ and operator H is applied on it. (Input image)
- Imagine we get resultant image as $g(x,y)$

$$\boxed{H(f(x,y)) = g(x,y)}$$

(Output image)

- H is linear if

$$H[a_i \cdot f(x,y) + a_j \cdot f(x,y)]$$
$$= a_i \cdot H(f(x,y)) + a_j \cdot H(f(x,y))$$
$$= a_i\, g(x,y) + a_j\, g(x,y)$$

$[\because H f(x,y) = g(x,y)]$

If LHS = R.H.S

then H is a linear operation

$f_i(x,y)$ ↓ $\quad f_j(x,y)$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$\boxed{\Pi = \Sigma}$

$a_i = 2$

$a_j = 2$

Substitute the value in LHS and RHS

LHS

$$\Sigma\left[2\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 2\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right]$$

$$\cancel{= 2\Sigma\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 2\Sigma\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}$$

$$= \Sigma\left\{\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix} + \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right\}$$

$$= \Sigma\begin{bmatrix} 4 & 6 \\ 8 & 10 \end{bmatrix}$$

$$= 28$$

RHS

$$a_i g_i(x,y) + a_j g(x,y)$$

$$2\Sigma\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 2\Sigma\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$2 \times 10 + 2 \times 4$$

$$20 + 8$$

$$= 28$$

LHS = RHS

$\Rightarrow$ This is a Linear ~~function~~ operator.

23/08/2023 "Image Processing"

$f_1(x,y)$ $\quad$ $f_2(x,y)$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$a_i = 2$
$a_j = 2$

$H \rightarrow$ Max.

LHS

$$H[a_i f_1(x,y) + a_j f_2(x,y)]$$

$$\text{Max}\left[2\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 2\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right]$$

$$\text{Max}\left[\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix} + \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right]$$

$$\text{Max}\begin{bmatrix} 4 & 6 \\ 8 & 10 \end{bmatrix}$$

$$= 10.$$

RHS

$$a_i\, g(x,y) + a_j\, g(x,y)$$

$$2\,\text{Max}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 2\,\text{Max}\begin{bmatrix} 1 & \\ 1 & \end{bmatrix}$$

$$2 \times 4 + 2 \times 1$$

$$8 + 2$$

$$= 10$$

$\therefore$ LHS = RHS

$\Rightarrow$ Max is a Linear operator

for $n = \text{Min}$

$f_1(x,y)$ $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $f_2(x,y)$ $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ $a_i = 2$, $a_j = 2$

LHS	RHS
$n[a_i f_1(x,y) + a_j f_2(x,y)]$	$a_i(g_i(x,y)) + a_j(g_j(x,y))$
$\text{Min}\left[2\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 2\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right]$	$2\,\text{Min}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 2\,\text{Min}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$
$\text{Min}\left[\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix} + \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right]$	$2 \times 1 + 2 \times 1$
$\text{Min}\begin{bmatrix} 4 & 6 \\ 8 & 10 \end{bmatrix}$	$2 + 2$
$= 4$	$= 4$

$\therefore$ LHS = RHS

$\Rightarrow$ Min is a Linear Operator

Operations on pixels

↳ **Arithmetic operations** [+, −, *, /]

↳ all operations performed among pixels and images are of same size.

e.g.:- $f(x,y) = \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$ $g(x,y) = \begin{bmatrix} 10 & 10 \\ 10 & 10 \end{bmatrix}$

• Images are grayscale

- $s(x,y) = \begin{bmatrix} 20 & 30 \\ 40 & 50 \end{bmatrix}$
- $d(x,y) = \begin{bmatrix} 0 & 10 \\ 20 & 30 \end{bmatrix}$
- $p(x,y) = \begin{bmatrix} 100 & 200 \\ 300 & 400 \end{bmatrix} = \begin{bmatrix} 100 & 200 \\ 255 & 255 \end{bmatrix}$
- $V(x,y) = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

↓

Normalize if values are not in given range

★ Addition: e.g. Image Averaging
↳ used for image enhancement;

- $g(x,y) = f(x,y) + N(x,y)$

$g(x,y)$ ↓ corrupted image after adding noise; $f(x,y)$ ↓ original image; $N(x,y)$ ↓ noisy mask.

- $g'(x,y) = 1/K \sum_{i=1}^{K} g_i(x,y)$ where K is diff noisy images

$g'(x,y)$ ↓ average of all noisy images.

★ Image Subtraction

- $g(x,y) = F(x,y) - h(x,y)$

  $g(x,y)$ → Resultant Image; $F(x,y)$ → Original Image I; $h(x,y)$ → Original Image II

- Used in medical imaging.

★ Image Multiplication/Division

- $\boxed{g(x,y) = F(x,y) * h(x,y)}$
- Used in shading correction
- $f(x,y) = g(x,y) / h(x,y)$

  $f(x,y)$ → Perfect image after applying correction; $h(x,y)$ → correction

→ **Set Operations** ← logical operators are used.

Image "Processing"

28/08/2023

↳ Spatial Operations

↳ Single pixel operations.

- $S = T(z)$
  - ↳ Transformation function

↳ Intensity transformation of pixels.

Neighborhood operations

- ↳ perform local blurring of image.

↳ $g(x,y) = \frac{1}{\ldots} \sum_{x,y \in S(x,y)} f(x,z)$

$g \in S(x,y)$

Affine Transformations

- Identity
- Scaling
- Rotation
- Translation

$$\begin{array}{cc} & \begin{array}{cc} 0 & 1 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\ & \begin{array}{cc} (0,0) & (0,1) \\ (1,0) & (1,1) \end{array} \end{array}$$

$\boxed{x' = C_x x}$ for (0,0) $x' = 2 \times 0 = 0$ $\boxed{C_x = 2}$

$\boxed{y' = C_y y}$ $y' = C_y = 2$ $C_y = 2$

Scaling

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

for (0,0)

$x' = 2 \times 0 = 0$; $y' = 2 \times 0 = 0$.

for (0,1)

$x' = 2 \times 0 = 0$ $y' = 1 \times 2 = 2$

for (1,0)

$x' = 2 \times 1 = 2$ $y' = 2 \times 0 = 0$

for (1,1)

$x' = 2 \times 1 = 2$ $y' = 2 \times 1 = 2$.

New matrix

$$\begin{array}{cc} \begin{array}{cc} 1 & 2 \end{array} & \\ \begin{bmatrix} \boxed{(0,0)} & \boxed{(0,2)} \\ 3 & 4 \\ (2,0) & (2,2) \end{bmatrix} & \end{array}$$

$x' = 0\cos 45 - 0\sin$

Rotation

$x' = x\cos\theta - y\sin\theta$

$y' = x\sin\theta + y\cos\theta$

for (0,0)

$x' = 0$ $y' = 0$

for (0,2)

$x' = 0\cos 45 - 2\sin 45$

$= 0 \times \frac{1}{\sqrt{2}} - 2 \times \frac{1}{\sqrt{2}}$

$= -\sqrt{2}$

$y' = 0\sin 45 + 2\cos 45$

$= 0 \times \frac{1}{\sqrt{2}} + 2 \times \frac{1}{\sqrt{2}}$

$= \sqrt{2}$

for (2,0)

$x' = 2\cos 45 - 0\sin 45$

$= 2 \times \frac{1}{\sqrt{2}}$

$= \sqrt{2}$

$y' = 2\sin\theta + 0\cos\theta$

$= 2 \times \frac{1}{\sqrt{2}} + 0 \times \frac{1}{\sqrt{2}}$

$= \sqrt{2}$

for (2,2)

$x' = 2\cos 45 - 2\sin 45$

$= 0$

$y' = 2\cos 45 + 2\sin 45$

$= 2 \times \frac{1}{\sqrt{2}} + 2 \times \frac{1}{\sqrt{2}}$

$= \sqrt{2} + \sqrt{2}$

$= 2\sqrt{2}$

1 $(0,0)$	2 $(-\sqrt{2}, \sqrt{2})$
3 $(\sqrt{2}, \sqrt{2})$	4 $(0, 2\sqrt{2})$

Rotation

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

04/09/2023

"Image Processing"

Topic: Image Enhancement

Enhancement

- Spatial Domain
  - works directly on pixels.
  - $g(x) = T[f(x)]$
    - Transformation function
  - enhanced image.
- frequency Domain
  - fourier's transformation.

Intensity transformation

- Linear (-ve and identity transformation) — $S = L - 1 - r$ [where $L-1$ is max value]
- logarithmic (log & inverse log) — $S = c\log(1+r)$ & $r \geq 0$ (c → constant)
- Power-law (nth power & nth root transformation) — $S = cr^{\gamma}$ (c → positive constant)

- Contrast Stretching — used for enhancing low contrast image.
  - $255 \times \left(\frac{r - r_{min}}{r_{max} - r_{min}}\right)$ — normalization
  - for 8bit image
    - r_1 r_2 S_1 S_2
      - r_1 → min, r_2 → max, S_1 → 0, S_2 → 255
    - transformation b/w range is done and rest value are darkened.

$$\begin{bmatrix} 100 & 120 & 220 \\ 150 & 200 & 100 \\ 100 & 150 & 200 \end{bmatrix}$$

0 - 255

$r_{min} = 100$ $\quad$ $r_{max} = 220$

$s_1 = 0$ $\quad$ $s_2 = 255$

- $255 \times \left(\frac{100-100}{220-100}\right) = 0$
- $255 \times \frac{120-100}{220-100} = \frac{255 \times 20}{120} = 42.5$
- $255 \times \frac{220-100}{220-100} = 255$ $\quad = 255$
- $255 \times \frac{50}{120} = \frac{1275}{12} = 106.25$
- $255 \times \frac{100}{120} = \frac{2550}{12} = 212.5$
- $255 \times \frac{0}{220-100} = 0$
- $255 \times \frac{0}{220-100} = 0$
- $255 \times \frac{50}{120} = 106.25$
- $255 \times \frac{100}{120} = 212.5$

$$\begin{bmatrix} 0 & 42.5 & 255 \\ 106.25 & 212.5 & 0 \\ 0 & 106.25 & 212.5 \end{bmatrix}$$

its application in satellite images

- Intensity level slicing operation

Here, only intensity value of interest is brighten or darken and rest values remains same.

Image Processing — 06/09/2023

Unit 2

- **Bit Plane slicing :→**

Bit plane slicing

Pixel	Value	After slicing
10	1010	0000 (0)
5	0101	0001 (1)
8	1000	0000 (0)
3	0011	0001 (1)

↳ • Higher level order bit
↳ for most significant visual info.

↳ • Lower level – for more subtle intensity i.e minute details.

Advantage :- • computation & storage of image is less.
• used in compression and reduces cost.

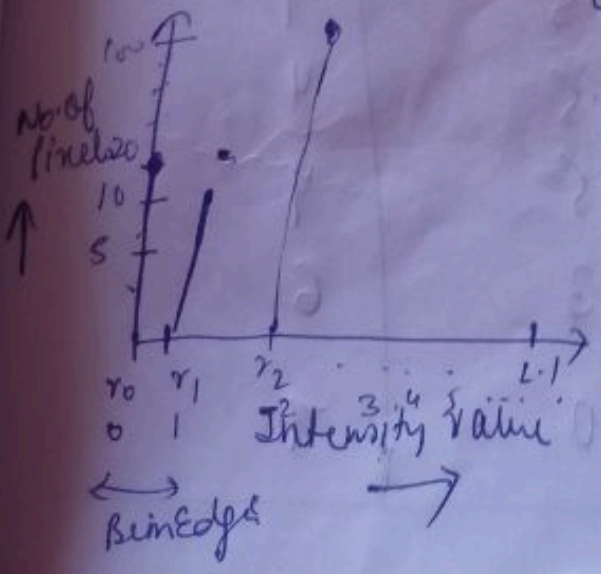
- **Histogram :→** • Unnormalized Histogram $\boxed{h(r_K) = n_K}$ • Intensity value range (0 to L-1). $K = 0, 1, 2 \ldots L-1$

• ↳ discrete function for digital image.

histogram ↳ Original pixel value ↳ no. of pixels with that particular intensity value.

• $\boxed{\text{Normalized Histogram} = \frac{h(r_K)}{N} = \frac{n_K}{N}}$

~~No. of Rows~~ Total no. of Pixels.



$r_0 \to 20$
$r_1 \to 10$
$r_2 \to 100$

advantage of it :-

• for checking for the similarity in two images.

Disadvantage

• Images considered as similar according to histogram ~~might only be same~~ but semantically not correct ~~but not~~

Histogram Equalization :- Used for image enhancement like contrast stretching.

- $S = T(r)$ → Transformation function

  S → Enhanced Pixel Value

  r → Original Pixel Value

↳ Two conditions for transformation function to satisfy:

- $T(r)$ → should be single value and monotonically increasing in range $0 <= r <= 1$
- $0 <= T(r) <= 1$ for $0 <= r <= 1$ → ensures that Transformed image will not have pixel value greater than allowed maximum value

ensures there is no change in ordering of pixels i.e. brighter remains brighter & darker remains darker.

- $\boxed{r = T^{-1}(S)}$

- $$\boxed{S_K = T(r_K) = (L-1)\sum_{j=0}^{K} Pr(r_j) \quad K = 0, 1, 2 \ldots L-1}$$

r_K	$Pr(r_K) = \frac{n_K}{MN}$	cf $\sum_{j=0}^{K} Pr(r_j)$	$(L-1) * \sum_{j=0}^{K} Pr(r_j)$	$S_K = T(r_K)$
0	0	0	0	0
1	0.1	0.1	0.7	1
2	0.1	0.2	1.4	1
3	0.3	0.5	3.5	4
4	0	0.5	3.5	4
5	0	0.5	3.5	4
6	0.4	0.9	6.3	6
7	0.1	1.0	7.0	7

P_K	n_K	$Pr(r_K) = \frac{n_K}{MN}$	CF $\sum_{j=0}^{K} Pr(r_j)$	$(L-1)\sum_{j=0}^{K} Pr(r_j)$	$S_K = T(r_K)$
0	790	0.192	0.19	1.33	1
1	1023	0.25	0.44	3.08	3
2	850	0.21	0.65	4.55	5
3	656	0.16	0.81	5.67	6
4	329	0.08	0.89	6.23	6
5	245	0.06	0.95	6.85	7
6	122	0.03	0.98	6.86	7
7	81	0.02	1.00	7.00	7
Σn_K	4096				

	0	1	2	3	4	5	6	7
	3	2						

$2^8 = 256$

$(L-1)$ 7

$\begin{bmatrix} 0 & 5 & 7 \\ 3 & 2 & \end{bmatrix}$

2^4

2^8

08/09/202

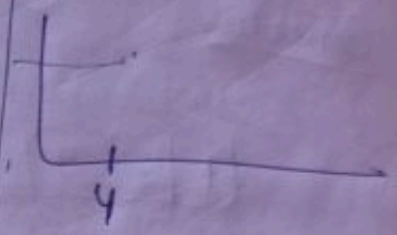
"Image Processing"

↳ Histogram matching / specification.

Intensity Values	No. of pixels	$P_r = \frac{P_i}{MN}$	CF	(L-1)CF	~~Original~~
0	8	0.125	0.12	0.84	1
1	10	0.15	0.27	1.89	2
2	10	0.15	0.42	2.94	3
3	2	0.03	0.45	3.15	3
4	12	0.18	0.63	4.41	4
5	16	0.25	0.88	6.16	6
6	4	0.06	0.94	6.58	7
7	2	0.03	0.97	6.79	7
	MN = 64				

Intensity Values	No. of Pixels	$P_r = \frac{P_i}{MN}$	CF	(L-1)×CF	~~Target Value~~
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	20	0.31	0.31	2.17	2
5	20	0.31	0.62	4.34	4
6	16	0.25	0.87	6.09	6
7	8	0.12	0.99	6.73	7
	ΣP_r 64				

VR	Original	Target	Matching
0	1	0	4
1	2	0	4
2	3	0	5
3	3	0	5
4	5	2	6
5	6	4	6
6	7	6	7
7	7	7	7



Histogram specification

Unit-2 Topic:- **Spatial filtering** → enhancement done in spatial domain

↳ applied directly on image.

↳ neighbourhood is considered.

↳ filter can be linear or non linear.

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

• Correlation

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

• filter is directly applied on the image.

Convolution

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

↳ first filter is reversed then it is applied.

Q. Perform correlation & convolution on 1D signal 00010000 with the mask 1 2 3 2 8.

Correlation

0	0	0	0	0	0	0	0	0

1	2	3	2	8

0	0	0	8	2	3	2	1	0	0	0

Convolution

0	0	0	1	0	0	0	0

8	2	3	2	1

↳

0	0	0	1	2	3	2	8	0	0	0

Spatial filtering technique

- ↳ Linear smoothening filter
- ↳ Median filter (non linear)
- ↳ Sharpening filter.

Image Processing

13/09/2023

Topic:-

Smoothening & sharpening

Linear Smoothening filter.

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

- Box filter
  - ↳ coefficients are equal
  - ↳
- Weighted average.
  - ↳ pixels are multiplied by different value
  - ↳ diagonal element weight less.
  - ↳ pixel at centre given more weight

Adv
↳ used for noise reduction.

dis ↳ Blur the edges.

- Mean filter
  - ↳ Same as Box filter.
- geometric mean filter
  - ↳ Less image details lost

$$f'(x,y) = \frac{1}{mn} \sum_{(r,c) \in S_{x,y}} g(r,c)$$

$$f'(x,y) = \prod_{(r,c) \in S_{x,y}} g(r,c)^{\frac{1}{mn}}$$

product

- Harmonic mean filter

$$f'(x,y) = \frac{mn}{\sum_{(r,c) \in S_{xy}} \frac{1}{g(r,c)}}$$

works well on salt noise fails for pepper noise
works well on gaussian noise also

- Contra harmonic mean filter.

$$\frac{\sum_{(r,c) \in S_{xy}} g(r,c)^{Q+1}}{\sum_{(r,c) \in S_{xy}} g(r,c)^{Q}}$$

→ Order of filter.

- +ve value of Q filter eliminates salt noise
- -ve values of Q, it eliminates pepper noise.
  - Cannot do both simultaneously

Non-Linear filters

- **Median filter** :— • Pixel values are replaced by median values of its neighbours.
  - $\frac{1}{2}$ values above median & $\frac{1}{2}$ below median
  - Reduces noise (especially salt & pepper).
  - Forces the points with distinct intensity values to be more like their neighbours.

0 0 0 0 0 1 4 0 2 2

0 0 0 0 0 1 2 2 4

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

→

0	1	1	1	1	0
1	1	1	1	1	1
1	1	1	1	2	1
1	1	1	1	1	1
1	2	2	2	2	2
0	1	1	2	1	0

Apply following filters on image below calculate the pixel value at (1,1) smoothening is done in 3x3 neighbourhood

• Box filter

7	9	5
4	6	8
2	0	1

7	9	5
4	0.67	8
2	0	1

• Weighted average

7	9	5
4	6	8
2	0	1

7	9	5
4	$\frac{1}{7}$=0.14	8
2	0	1

• Median filter

7	9	5
4	6	8
2	0	1

7	9	5
4	5	8
2	0	1

• Max filter

7	9	5
4	6	8
2	0	1

7	9	5
4	9	8
2	0	1

• Min filter

7	9	5
4	6	8
2	0	1

7	9	5
4	0	8
2	0	1

$\frac{9+0}{2} = 4.5$

Mid point

7	9	5
4	6	8
2	0	1

7	9	5
4	4.5	8
2	0	1

• Geometric

7	9	5
4	6	8
2	0	1

• Harmonic

7	9	5
4	6	8
2	0	1

Image Processing

18/09/2023

Sobel operations

→ • first order derivatives for image Sharpening
→ implemented using magnitude of gradient

• $\text{grad}(f) = \nabla f = \begin{pmatrix} \frac{df}{dx} \\ df/dy \end{pmatrix}$

• de magnitude $= \sqrt{\frac{d^2f}{d^2x} + \frac{d^2f}{d^2y}}$

$$\text{mag}(\nabla f) = \left|\frac{df}{dx}\right| + \left|\frac{df}{dy}\right|$$

• $G_x = \frac{\partial f}{dx} = [f(x+1, y-1) + f(x+1, y+1) + 2f(x+1, y)] - [f(x-1, y-1) + f(x-1, y+1) + 2f(x-1, y)]$

• $G_y = \frac{df}{dy} = [f(x-1, y+1) + f(x+1, y+1) + 2f(x, y+1)] - [f(x-1, y-1) + f(x+1, y-1) + 2f(x, y-1)]$

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

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

10×-1 + 10×

0	0	-10	-10	-10

$G_x = 1$

$G_y = 0$

Frequency domain filtering

spatial domain $f(x) \longrightarrow \boxed{FT} \longrightarrow f(u)$ frequency domain

Fourier Tranform

and after applying filter, we do inverse operation

$f(x) \longleftarrow \boxed{FT} \longleftarrow f(u)$

Advantage

- There is no information loss in forward or backward direction

Fourier Transform

$$\boxed{f(u) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi ux}\, dx}$$

operation. Space

frequency of value

Sqrt(-1)

- Inverse fourier Transform

$$\boxed{f(x) = \int_{-\infty}^{\infty} f(u)\, e^{i2\pi ux}\, du}$$

Convolution and fourier Transform

↳ Convolution in spatial domain:

$$g(x) = f(x)\,(conv)\,h(x) = \int_{-\infty}^{\infty} f(x)\,h(x-x)\,dx$$

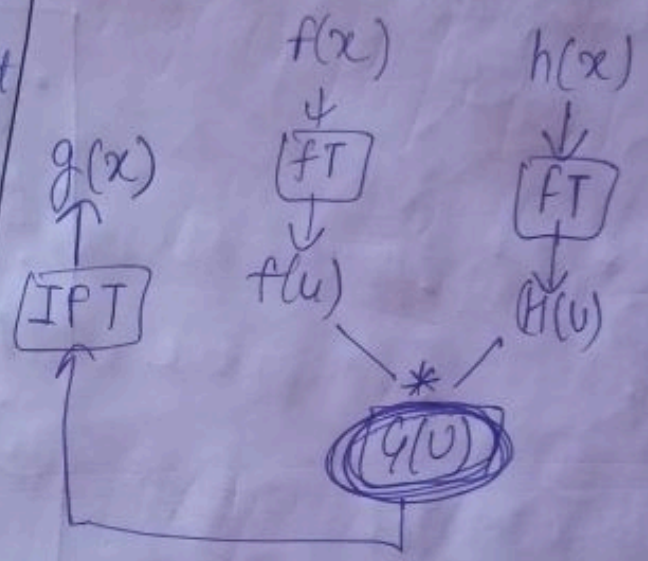
↳ In frequency Domain convolution is equivalent to

$$g(x) = f(x)\,(conv)\,h(x) \longleftrightarrow G(U) = F(U) * H(U)$$

Convolution in spatial domain ⟷ Multiplication in frequency.

Adv

- Algorithm for doing FFT and IFFT are very fast
- ~~whereas~~ In spatial filtering is expensive and time consuming for large Kernels.



- **Low frequency** – slow variation in intensity value e.g sky.
- **High frequency** – sharp transition in Intensity value e.g edges or noise

Low pass filter → allows only low frequency components

↳ used for smoothening

High pass filter → allows only high frequency components

↳ used for sharpening