Date
Page

Digital Image Processing

Module 4.

1. Basic Relationship and Distance Matrices:

→ An analog image of size 3X3 is represented in the 1st quadrant of the cartesian system.

Y axis				
2	f(0, 2)	f (1, 2)	f(2,2) end	
1	f(0,1)	f (1,1)	f (2,1)	
0	f(0,0)	f (1,0)	f (2,0)	
	0	1	2	x axis

f(0,0) is the bottom left corner. Since it starts from coordinate position (0,0), it ends at f(2,2) that x=0,1,2 to M-1 and y=0,1,2 -- N-1.

x, y are the dimensions of the image.

Note:

In DIP, discrete form of image is always used. which is represented in the 4th Quadrant. which starts with index (0,0) but in MATLAB it starts with (1,1)

	y=0	y=1	y=2
x=0	f(0,0)	f(0,1)	f(0,2)
x=1	f(1,0)	f(1,1)	f(1,2)
x=2	f(2,0)	f(2,1)	f(2,2)

'Discrete image of 4th Quadrant'

	y=1	y=2	y=3
x=1	(1,1)	(1,2)	(1,3)
x=2	(2,1)	(2,2)	(2,3)
x=3	(3,1)	(3,2)	(3,3)

Analysis in MATLAB software.

Basic Relationship ~~b/w~~ and Distance Matrix (Points):-

Image representation as matrix - Represented as array where each element corresponds to pixel.

Logical pixels - They ~~occur~~ specify a location but don't occupy any physical space. They are represented in the cartesian first coordinate system (where both coordinates are positive and start from the origin)

Physical pixels - Actual dots that occupy the space when displayed on an output device. They are tangible and they occupy a small area. They are represented in the 4th coordinate system (Both coordinates can be -ve or +ve and origin may not be at corner).

Image Topology - Branch of IP that deals ~~that deals~~ with the properties like image neighbourhood, path among pixels boundary and connected components.

4 Neighbourhood - reference pixel $p(x,y)$ at coordinate position (x,y) has 2 horizontal and 2 vertical pixels as neighbours.

$$N4(P) \quad \begin{bmatrix} 0 & X & 0 \\ X & \leftarrow p(x,y) \rightarrow & X \\ 0 & X & 0 \end{bmatrix}$$

Set of pixels $\{(x+1, y), (x-1, y)\ (x, y+1)\ (x, y-1)\}$ are 4 neighbours. $[N4(P)]$

It can be represented as diagonal elements of 4 neighbour

$$ND(X) \quad \begin{bmatrix} X & 0 & X \\ 0 & p(x,y) & 0 \\ X & 0 & X \end{bmatrix}$$

Sets are: $(x-1, y-1)$ $(x-1, y+1)$ $(x+1, y-1)$ $(x+1, y+1)$

8 neighbours

$$N8(P) \quad \begin{bmatrix} X & X & X \\ X & \leftarrow p(x,y) \rightarrow & X \\ X & X & X \end{bmatrix}$$

$\therefore \boxed{N8(X) = N4(P) \cup ND(X)}$

Connectivity – Relationship b/w 2 or more pixels ~~are called~~ is known as pixel connectivity.

The pixels p and q are said to be connected if certain conditions on pixel brightness specified by set V and spatial frequency is satisfied.

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

4 connectivity – If pixels p and q both have the same values specified by the set V and q is said to be in the set of N4(P). This implies any path from P to q on which every other pixel is 4 connected to next pixel.

8 connectivity – 'q' is in the set of N8(P) and p and q pixels share a grey scale.

$$\begin{matrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \qquad V = \{0, 1\}$$

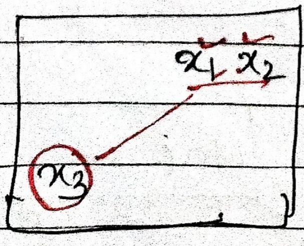
Mixed connectivity – 2 pixels p and q are in m connectivity when:

i) q is in N4(P)

ii) q is in ND(P) and intersection of N4(p) and N4(q) is empty.

$$\left.\begin{matrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}\right\}$$ Here no multiple paths are present.

Relations: A binary relation b/w two pixels a and b are denoted by aRb specifies pair.



x_1, x_2 are in 4 connectivity.
x_3 is ignored since its not connected to any other element.

Reflexive: For any element 'a' in set A, if relation aRa exists, its reflexive

Symmetric – If aRb implies bRa exists, then its symmetric

Transitive – If the relation aRb and bRc exists, then aRc also exists, this is transitive property.

Distance measures – distance b/w pixels p and q can be given by Euclidian distance, D_4 distance and D_8 distance

$$\begin{array}{c}\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\end{array}\begin{array}{l}\leftarrow (z) \\ \\ (q) \\ \leftarrow (p)\end{array}$$

'sample image'

i) $D(p,q)$ is definite and finite for all p & q
ii) $D(p,q) \geq 0$; if $p=q$, then $D(p,q)=0$
iii) Distance $D(p,q) = D(q,p)$
iv) $D(p,q) + D(q,z) \geq D(p,z)$ (triangular inequality)

Euclidean Distance:

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

(x,y) and (s,t) are coordinates

(p,q) are pixels

(x,y) : (s,t)

D_4 distance / city block distance –

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

D_8 distance / chessboard distance

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

Important image characteristics:

i) **Connected set** – For binary image, connected set is a collection of pixels that are connected to each other based on specific connectivity (N_4 or N_8 or N_D)

Date
Page

@vtupadhai

ii) Digital Path or curve - A digital path from pixel p to q is a sequence of points $p_1, p_2 \cdots p_n$. Length of this path is the number of pixels in the ~~image~~ sequence.
$p = (x_0, y_0)$, $q = (x_n, y_n)$ if both are equal, there exists a closed path.

iii) Region - connected component within the image. Set of pixels grouped together based on connectivity

iv) Connected Component - A subset of pixels within a connected set S, there exists a path between any 2 pixels p and q within S.
p q

v) Adjacent Regions - Two regions R_1 and R_2 are adjacent if their union forms a connected component.
- If region is not adjacent, they are disjoint sets

vi) Boundary - Boundary or contour of image is the set of pixels that outline a region and have one or more neighbours outside the region. If border pixels are within region, it is inner boundary.

vii) Edge - Edges occur when there is abrupt change in intensity of pixels. They can be disjoined or may be joined using edge linking algorithms

Classification of Image processing operations.

Based on neighbourhood:-
i) Point operation
ii) Local operation
iii) Global operation.

Point operations - Where o/p value at a specific coordinate

@vtupadhai

is dependent only on the input value.

Local operation – o/p value of at a specific coordinate is dependant on the input value in the neighbourhood of that pixel.

Global operation – o/p value at a specific coordinate is dependant on all the values of that input image

Another way of categorising them are :–

i) **Linear operations** – follows principles of additivity and homogeniety

ii) **Non linear operations** –

1) **Property of additivity** :–

$$H(a_1 f_1(x,y) + a_2 f_2(x,y)) = H(a_1 f_1(x,y)) + H(a_2 f_2(x,y))$$

$$= a_1 H(f_1(x,y)) + a_2 H(f_2(x,y))$$

$$= a_1 \times g_1(x,y) + a_2 \times g_2(x,y)$$

2) **Property of Homogeniety**

$$H(k f_1(x,y)) = kH(f(x,y)) = k g_1(x,y)$$

They also perform array operations :–

$$F_1 = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad F_2 = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

$$F_1 \times F_2 = \begin{bmatrix} AE & BF \\ CG & HD \end{bmatrix}$$

Arithmatic operations – (Demonstrate with diagrams of your own {self explanatory})

i) **Image Additions** :– $g(x,y) = f_1(x,y) + f_2(x,y)$

o/p pixel. i/p pixel

Rules → Sum should not cross the allowed range. If exceeds the pixel value is set to the maximum allowed value

@vtupadhai

Similarly constant value can be added to a single image

$$g(x,y) = f_1(x,y) + k$$

Applications:-

i) Create double exposure - technique of superimposing an image on another image to produce resultant.

ii) Increase brightness of the image

ii) Image subtraction :- $g(x,y) = f_1(x,y) - f_2(x,y)$

$g(x,y)$ ↓ o/p pixel of image; $f_1(x,y)$, $f_2(x,y)$ — i/p pixel of image

- Use a modulus operation
- For constants:- $g(x,y) = |f_1(x,y) - k|$ which decreases brightness of the image

Applications

i) Background elimination

ii) Brightness reduction

iii) Change Detection

iii) Image Multiplication — $g(x,y) = f_1(x,y) \times f_2(x,y)$

$g(x,y)$ ↓ o/p image; $f_1(x,y)$, $f_2(x,y)$ — i/p image

constant: $g(x,y) = f(x,y) \times k$ ∴ $k > 1$

$g(x,y) = a f(x,y) + k$. $k < 1$

Applications:

i) Increases contrast

ii) used for designing filter masks

iii) Creating a mask to highlight the area of interest

iv) Image Division - $g(x,y) = f_1(x,y) / f_2(x,y)$

- Results in floating point numbers.

For constants: $g(x,y) = f(x,y) / k$

Application: change detection, seperation of luminescence and reflectance components & contrast reduction.

Date
Page

Applications of Arithmatic operations: -

- Image averaging process can be used to remove noise.
  Noise is a random fluctuation of pixel values, which affects the quality of the image.

$$g(x,y) = f(x,y) + \eta(x,y)$$

o/p image   i/p image

Several instance of noise images can be averaged a

$$\bar{g}(x,y) = \frac{1}{M} \sum_{i=1}^{M} g_i(x,y)$$

M is no of noisy images. As M increases, the average process reduces the intensity of the noise and it becomes so low, that it automatically gets removed. As, M becomes large $E\{\bar{g}(x,y)\} = f(x,y)$.

Logical operations:- Basic operations include AND/NAND, OR/NOR, EXOR/EXNOR and Invert/Logical NOT

1) Truth Table: (AND/NAND)

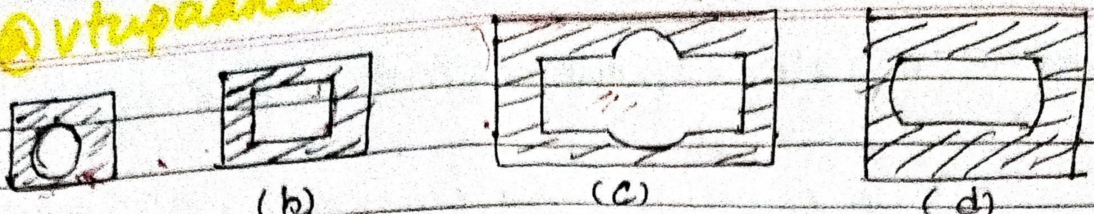
A	B	C(AND)	N(AND)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

- Operators AND and NAND takes 2 images as i/p and produce 1 o/p image which is the output of logical AND/NAND of individual pixels.

Applications:

i) Computation of intersection of images.
ii) Design of filter masks.
iii) Slicing of Grey Scale images

@vtupadhai


(a) (b) (c) (d)

a, b → Images 1 and 2
c → Result of image 1 OR image 2.
d → Result of image 1 AND image 2

2) OR/NOR

Truth Table:

A	B	C(OR)	C(NOR)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

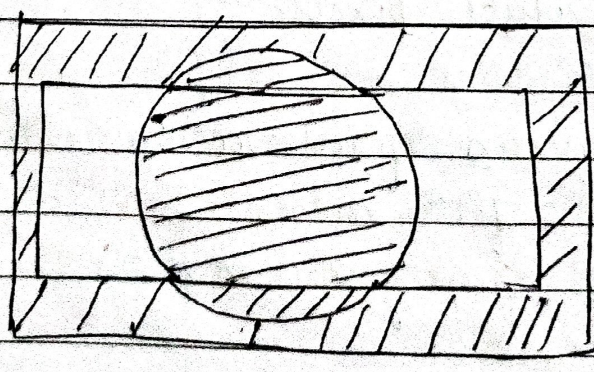
Applications: – OR is used for union operator of 2 images
– OR can be used as merging operator.

3) XOR/NOR

Truth Table

A	B	C(XOR)	C(XNOR)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Application – • Change Detection
• Used as subcomponent of complex image operation



Result of XOR operation.

4) Invert / Logical NOT –

Truth table:

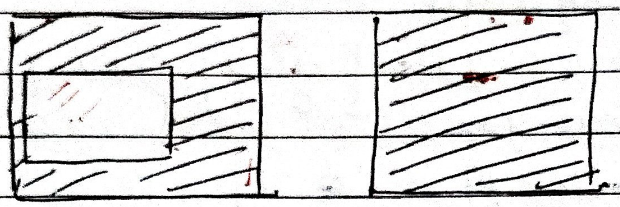
A	C (NOT)
0	1
1	0

For grey scale, inversion operation is given by

$$g(x,y) = 255 - f(x,y)$$

Applications: –

i) Obtaining negative of image
ii) Making features clear to the observer
iii) Morphological processing



Original   Invert image

Similarly images can be compared using
$=, >, >=, <, <=, \neq$

Geometric operations: (2D operations)

Translation – Movement of an image to a new position

$$x' = x + \delta x$$

x, y

$$y' = y + \delta y$$

$$F' = F + T \quad \text{(vector representation)}$$

give explanation same as Q1

$$[x' \; y' \; 1] = \begin{bmatrix} 1 & 0 & \delta x \\ 0 & 1 & \delta y \\ 0 & 0 & 1 \end{bmatrix} [x, y, 1]^T$$

Scaling – Enlarging or shrinking of an image

$$x' = x \times S_x$$

$$y' = y \times S_y$$

$$[x', y'] = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} [x, y]$$

If scaling factors are equal, then its isotropic scaling or else it is a differential scaling

$$\therefore [x', y', 1] = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} [x, y, 1]^T$$

Mirror / Reflection operation – ~~create~~ Returns the image where pixels are reversed.

→ Reflection along X axis –

$$F'' = [-x, y] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times [x, y]^T$$

→ Reflection along Y axis

$$F' = [x, -y] = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \times [x, y]^T$$

→ Reflection about $y = x$

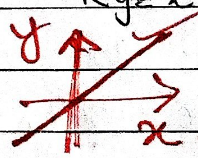
$$F' = [x, -y] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times [x, y]^T$$

→ Reflection about $y = -x$

$$F' = [x, -y] = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \times [x, y]^T$$

In homogenous coordinate system:–

$$R_{yaxis} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$R_{y=x} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{xaxis} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{y=-x} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$R_{origin} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Shearing – Transformation that produces distortion in shape. It can be applied in x direction and y direction

Date
Page

Xshear:

$x' = x + ay$

$y' = y$

$$x\,shear = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{where } (a = Sh_x)$$

Yshear:

$x' = x$

$y' = y + bx$

$$\begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (b = Sh_y)$$

Rotation : Rotate an image by an angle.

$$[x'\ y'] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} [x\ y]^T$$

$$\boxed{F' = R \cdot A}$$

rotation

+ve angle → clockwise rotation -ve angle → anticlockwise

$$[x'\ y'\ 1] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Affine Transforms: – The transformation that maps the pixel at the coordinates (x, y) to a new coordinate position given as a pair of transformation equation.

$x' = T_x(x, y)$

$y' = T_y(x, y)$

T_x and T_y are polynomials. This linear equation gives affine transform.

$x' = a_0x + a_1y + a_2$

$y' = b_0x + b_1y + b_2$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Inverse Transformations: – Restore the transformed object to its original form

Inverse translation :
$$\begin{bmatrix} 1 & 0 & -tx \\ 0 & 1 & -ty \\ 0 & 0 & 1 \end{bmatrix} \quad \{\text{default}\}$$

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

Inverse scaling
$$\begin{bmatrix} -Sx & 0 & 0 \\ 0 & -Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3D Transforms – Used in CT scans and MRI.

Translation =
$$\begin{bmatrix} 0 & 0 & 0 & \delta x \\ 0 & 1 & 0 & \delta y \\ 0 & 0 & 1 & \delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling =
$$\begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation $R_{x,\theta} =$
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_{y,\theta} =$
$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_{Z(\theta)} =$
$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflection transform: –

Reflection$_{xy}$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_{xz} =$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflection yz plane = $\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Shear Quantities a and b.

shear = $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $\quad$ $shear_x$ = $\begin{bmatrix} 1 & a & b & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$shear_y$ = $\begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Image Interpolation Techniques –
- Nearest neighbour
- Bilinear
- Bicubic

Forward mapping – Process of applying transformation iteratively to every pixel in the image, yielding a new coordinate position and copying the values of pixel to a new position

Backward mapping – Process of checking the pixels of the output image to determine the position of the pixels in the input image in order to guarantee that all the pixels of image are processed.

Sometimes may not fitted into new coordinate.

$x' = x\cos\theta - y\sin\theta$ $\quad$ (Point = 10,5) $\theta = 45^\circ$

$= 10\cos 45 - 5\sin 45$

$= 10(0.707) - 5(0.707)$

$= 3.535$

$y' = x\sin\theta + y\cos\theta$ $\quad$ (3.535, 10.605)

$= 10\sin(45^\circ) + 5\cos 45^\circ$ $\quad$ rounded to

$= 10.605$ $\quad$ (4, 11)

i) Nearest Neighbour – It determines the closest pixel and assigns it to every pixel in the new image matrix i.e brightness of the image pixels is equal to the closest neighbour. Sometimes, it may lead to pixel blocking and distort the resulting image. This is called aliasing.

ii) Bilinear Technique – It is the 1st order interpolation. Here, 4 neighbours of the transformed original pixel that surround the new pixels are obtained and are used to calculate the new pixel value.

$$g(x,y) = (1-a)(1-b)f(x'y') + (1-a)b f(x'y'+1) + a(1-b)f(x'+1, y') + ab f(x'+1, y'+1)$$

$g(x,y)$ ↓ o/p image.

If desired pixel is very close to one of the neighbours, it's weight will be higher, which may lead to blurring of edges. It reduces aliasing.

iii) Bicubic Technique – second order Interpolation. It's neighbourhood has 16 pixels. It fits two polynomials to the 16 pixels of the transformed original matrix and the new image pixel.

Set operation – Eg

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

A = {(0,0), (0,2) (2,2)} represent value 1 which can be used for image analysis.

$A^c = \{c / c \notin A\}$

$\hat{A} = \{c = -a, a \in A\}$

$A \cup B = \{c / (c \in a) \vee (c \in B)\}$

@vrupalini

$A \cap B = \{c \mid (c \in a) \wedge (c \in b)\}$

$A - B = \{c \mid (c \in A) \wedge (c \notin B)\}$

Dilation: Increases boundary of a region while the small holes present in the image becomes smaller

E.g

Q. Consider the image:

Apply Dilation & erosion

S = [1, 1]

with coordinates

$\{(0,0), (0,1)\}$

	0	1	2
0	0	0	1
F = 1	0	0	1
2	0	1	1

Ans:- Image 'F' can be written as: $F = \{(0,2)\ (1,2)\ (2,1)\ (2,2)\}$

Dilation :→ $S = \{(0,0), (0,1)\}$

Step1: Add coordinates of S to all coordinates of F

F dilation S = $\{(0,2)\ (1,2)\ (2,1)\ (2,2)\ (0,3)\ (1,3)\ (2,2)\ (2,3)\}$

Step2: Eliminate all the repeated pairs

Step3: Union of these sets results in dilation

1	0	1
0	0	0
0	0	1

$\boxed{1\ \ 1}$

Dilation :→

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

Erosion :→

Step1: Substract the coordinates of S from all coordinates of F

F Erosion S = $\{(0,2)\ (1,2)\ (2,1)\ (2,2)$
$(0,1)\ (1,1)\ (2,0)\ (2,1)\}$

Step2: Find the intersection element. S = (2,1)

Erosion

1	0	1
0	0	0
0	0	1

$\boxed{1\ \ 1}$

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

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

Note: for coordinates (x, y), and (s, t)

dilation $(x+s,\ y+t)$

erosion $(x-s,\ y-t)$

@vtupadhai

Date
Page

Statistical operations:

• Mean = $\bar{x} = \frac{x_1 + x_2 + \cdots x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$

• Median

• Mode

• Percentile

• Standard deviation and variance

• Entropy =

$$H = -\sum_{i=1}^{n} p_i \log_2 p$$ ∴ p_i is prior probability

Calculate entropy of the image:

Q.

1	2	3	5
3	4	4	6
9	9	8	7
1	3	5	6

Entropy = $-\sum_{i=1}^{n} p_i \log_2 p_i$

symbol in image	probability of occurance
1	2/16
2	1/16
3	3/16
4	2/16
5	2/16
6	2/16
7	1/16
8	1/16
9.	2/16

$$\text{Entropy} = -\Big\{\frac{2}{16}\log_2\frac{2}{16} + \frac{1}{16}\log_2\frac{1}{16} + \frac{3}{16}\log_2\frac{3}{16} + \frac{2}{16}\log_2\frac{2}{16} + \frac{2}{16}\log_2\frac{2}{16} + \frac{2}{16}\log_2\frac{2}{16} + \frac{1}{16}\log_2\frac{1}{16} + \frac{1}{16}\log_2\frac{1}{16} + \frac{2}{16}\log_2\frac{2}{16}$$

$$= 3.075$$