Year: 2016 (Fall)

Date ________
Page ________

Q 1(a) What is a digital image? Explain the relationship between pixels with their adjacency in image processing.

Sol^n → <u>Digital image:</u>

A digital image is a numeric representation (normally binary) of two dimensional images. A two-dimensional function, $f(x,y)$; where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x,y) is called the intensity or grey level of the image at that point. When x, y and the intensity value of f are all finite, discrete quantities, we call the image a digital image.

Thus, the digital image can be seen as matrix,

$$f = \begin{bmatrix} f(1,1) & f(1,2 & \cdots & f(1,M) \\ f(2,1) & f(2,2) & \cdots & f(2,M) \\ \vdots & \vdots & & \vdots \\ f(N,1) & f(N,2) & \cdots & f(N,M) \end{bmatrix}$$

The important relationships between pixels in a digital image are:- Neighbor of a pixel
- Adjacency / connectivity
- Distance measure betn pixels

Date
Page

(i) Neighbor of a pixel:

A pixel p at coordinates (x, y) has four horizontal and vertical neighbors whose coordinates are given by,

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

This set of pixels is called the 4-neighbors of p denoted by $N_4(P)$.

The four diagonal neighbors of p have coordinates $(x+1, y+1)$ $(x+1, y-1)$ $(x-1, y+1)$ $(x-1, y-1)$ are denoted by $N_D(P)$.

These points together with the 4 neighbors are called the 8-neighbors of p denoted by $N_8(P)$.

(ii) Adjacency / connectivity.

Let V be the set of intensity values used to define adjacency. In a binary image, $V = \{1\}$ if we are referring to adjacency of pixel with value 1. In a gray-scale image, the idea is the same, but set V typically contains more elements.

We consider three types of adjacency:

(a) 4-adjacency:

Two pixels p and q with values

from V are 4-adjacent if q is in the set $N_4(P)$.

(b) 8-adjacency:

Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(P)$.

(c) m-adjacency: (mixed adjacency):

Two pixels p and q with values from V are m-adjacent if

(i) q is in $N_4(P)$, or

(ii) q is in $N_D(P)$ and the set ~~$N_4(P)$~~ $N_4(P) \cap N_4(q)$ has no pixels whose value are from V.

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

fig: $N_4(P)$ fig: $N_8(P)$ fig: m-connected

(iii) Distance measures:

Given pixels p, q and z with coordinates (x,y), (s,t), (u,v) respectively, the distance function D has following properties:

(a) $D(P,q) \geq 0$ [$D(P,q) = 0$ if $P=q$]

(b) $D(P,q) = D(q,P)$

(c) $D(P,z) \leq D(P,q) + D(q,z)$

Date ______
Page ______

Q 1(b) Compute the Histogram of Equalization from the given data.

r_k	0	1	2	3	4	5	6	7
n_k	5320	1000	500	525	1236	956	856	128

Soln → Here, step(1):

Gray level (r_k)	Frequency (n_k)	P.D.F $P_r(r_k) = n_k/MN$	CDF $S_k = \sum P_r(r_k)$	$7 * S_k$	Rounding off
0	5320	0.506	0.506	3.542	4
1	1000	0.095	0.601	4.207	4
2	500	0.048	0.649	4.543	5
3	525	0.050	0.699	4.893	5
4	1236	0.117	0.816	5.712	6
5	956	0.091	0.907	6.349	6
6	856	0.081	0.988	6.916	7
7	128	0.012	1	7	7

$\sum F = 10521$

$\therefore MN = 10521$

Step(2): Representing new Gray level

Date
Page

Old Gray level	No. of pixel	New Gray level
0	5320	4
1	1000	4
2	500	5
3	525	5
4	1236	6
5	956	6
6	856	7
7	128	7

Step (3): Modified Histogram

Gray level (S_k)	No. of pixels (n_k)	$P_s(S_k) = n_k/10521$
~~0~~	0	0
~~1~~	0	0
~~2~~	0	0
~~3~~	0	0
4	6320	0.60
5	1025	0.10
6	2192	0.21
7	984	0.09

Date
Page

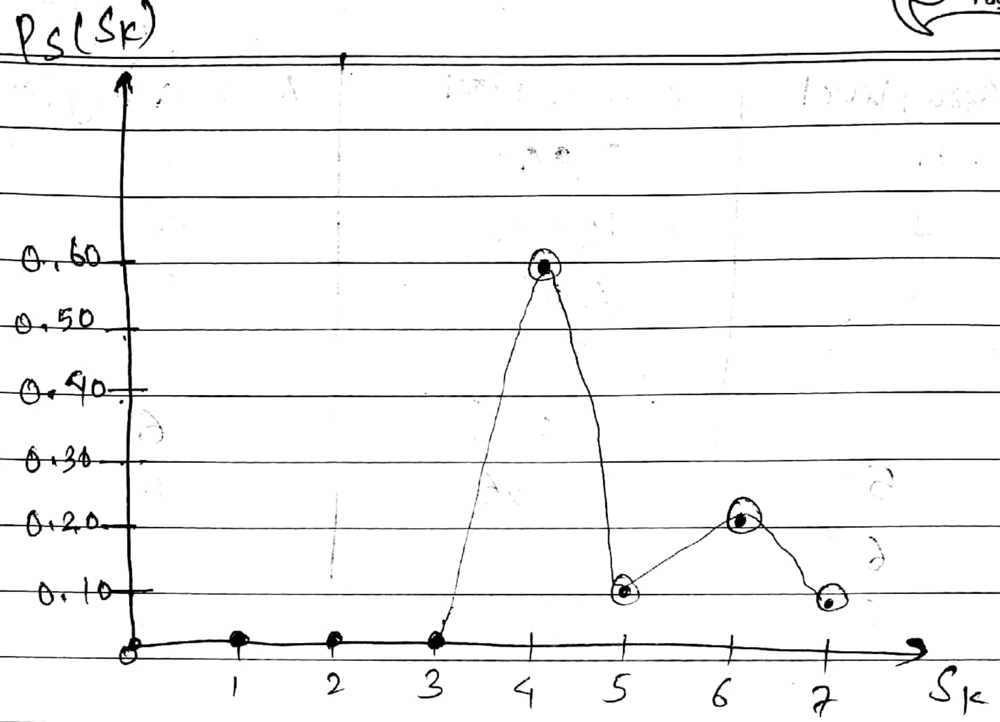


Fig: Equalized Histogram

Q 2(a) What do you mean by Image sharpening? Explain Median filter with suitable example.

Soln → Sharpening an image increases the contrast between bright and dark regions to bring out features. The sharpening process is basically the application of a high pass filter to an image.

The following array is a kernel for a common high pass filter used to sharpen an image.

$$\begin{bmatrix} -1/9 & -1/9 & -1/9 \\ -1/9 & 1 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{bmatrix}$$

Note: Median filter (solved in 2014 Fall 3(a)(i))

Q.2 (b) Calculate the Haar transform T from the given image matrix F.

F =

1	0	0	1
1	1	0	1
1	0	1	0
1	0	1	-1

And then reconstruct the original F by performing inverse Haar transform on T.

Sol^n →

Setp 1 :

Given image matrix, F =

$$F = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

We have, the Haar transformation matrix for N = 4

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

For computing Haar Transform,

$$T = HFH^T$$

$$= \left(\frac{1}{2}\right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \left(\frac{1}{2}\right) \begin{bmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{bmatrix}$$

Date ______
Page ______

$$= \left(\frac{1}{4}\right) \begin{bmatrix} 4 & 1 & 2 & 3 \\ 0 & 1 & -2 & 1 \\ 0 & -\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & -\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{bmatrix}$$

$$= \left(\frac{1}{4}\right) \begin{bmatrix} 10 & 0 & 3\sqrt{2} & -\sqrt{2} \\ 0 & 2 & -\sqrt{2} & -3\sqrt{2} \\ -\sqrt{2} & -\sqrt{2} & 2 & 0 \\ -\sqrt{2} & \sqrt{2} & 0 & 2 \end{bmatrix}$$

$$T = \begin{bmatrix} 5/2 & 0 & \frac{3}{2\sqrt{2}} & -1/2\sqrt{2} \\ 0 & 1/2 & -1/2\sqrt{2} & -3/2\sqrt{2} \\ -1/2\sqrt{2} & -1/2\sqrt{2} & 1/2 & 0 \\ -1/2\sqrt{2} & \frac{1}{2\sqrt{2}} & 0 & 1/2 \end{bmatrix} //$$

Hence, the Haar Transform T is,

$$\begin{bmatrix} 5/2 & 0 & \frac{3}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ 0 & 1/2 & -1/2\sqrt{2} & -\frac{3}{2\sqrt{2}} \\ -1/2\sqrt{2} & -1/2\sqrt{2} & 1/2 & 0 \\ -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & 0 & 1/2 \end{bmatrix}$$

Date ______
Page ______

Q 3 (a) Explain basic steps of filtering in Fourier domain with necessary figures.

Soln →

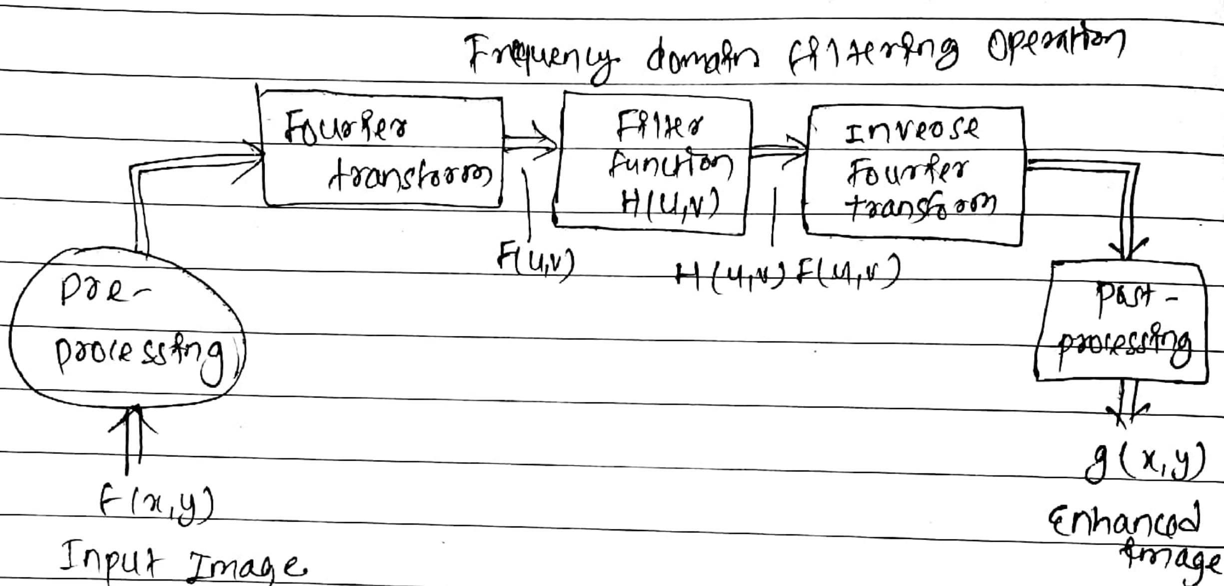


Figure: Basic steps for filtering in Fourier (Frequency) domain.

- To filter an image in Fourier (frequency) domain:

(i) Multiply the input image by $(-1)^{x+y}$ to center the transform.

(ii) Compute $F(u,v)$ the DFT of the image.

(iii) Multiply $F(u,v)$ by a filter function $H(u,v)$

(iv) Compute the inverse DFT of the result in (iii)

(v) Obtain the real part of the result in (iv)

(vi) Multiply the result in (v) by $(-1)^{x+y}$.

Date
Page

Q. 3(b) What is Noise in Image? explain the Adaptive Median filter in detail.

Sol^n Image noise is a random variation of brightness or color information in images. Noises are unwanted information in digital images. Noise produces undesirable effects such as artifacts, unrealistic edges, unseen lines, corners, blurred objects and distrubs background scenes.

Adaptive median filter:

The behavior of adaptive filters changes depending on the characteristics of image inside the filter region. The median filter performs relatively well on impulse noise as long as the spatial density of impulse noise is not large.

- The adaptive median filter can handle much more spatially dense impulse noise, and also performs some smoothing for non-impulse noise.
- The key insight in the adaptive median filter is that the filter size changes depending on the characteristics of the image.
- consider the following notation:

Z_{min} → Minimum gray level value in S_{xy}.
Z_{max} → Maximum gray level value in S_{xy}.
Z_{med} → Median of gray level value in S_{xy}.
Z_{xy} → Gray level of coordinate (x, y)
S_{max} → Maximum allowed size of S_{xy}.

Date
Page

The adaptive media filtering algorithm works in two levels denoted by Level A and level B as follows:

Level A:

$A_1 = Z_{med} - Z_{min}$

$A_2 = Z_{med} - Z_{max}$

- If $A_1 > 0$ and $A_2 > 0$, Go to level B
- Else increase window size.
- If window size $\leq S_{max}$ repeat level A
- Else output Z_{med}

Level B:

$B_1 = Z_{xy} - Z_{min}$

$B_2 = Z_{xy} - Z_{max}$

- If $B_1 > 0$ and $B_2 < 0$, output Z_{xy}
- Else output Z_{med}.

The main purpose of this algorithm are:

(a) To remove salt and paper (impulse noise).
(b) To provide smoothing of other noise that may not be impulse.
(c) To reduce distortion such as exessive thining or thickening of object boundaries.

Date
Page

The values of Z_{min} and Z_{max} are considered to be "impulse like" noise components statistically.

Q 4 (a) Why is image compression necessary? Explain lossy predictive coding with examples.

Sol → Image compression is the art and science of reducing the amount of data required to represent an image.

The objective of image compression is to reduce the redundancy of the image and to store or transmit data in an efficient form.

- Image compression is needed save storage space for high-resolution digital camera photos.
- To reduce the bandwidth while transferring images, videos through the network.
- To reduce the transmission time.
- Images can be loded faster in web pages.

Note: For Lossy predictive coding (solved in 2014 (Fall) 3(b)).

Date ____
Page ____

Q 4 (b) What is coding Redundancy? Construct Huffman code for each gray level and calculate average bit length from the given data.

r	0	1	2	3	4	5	6	7
N(r)	20	30	60	120	180	160	204	300

Where, r = Gray level
N(r) = pixels having r^{th} gray level

Soln

→ <u>Coding Redundancy:</u>

Image = Information + redundant data

A code is a system of symbols (letters, numbers, bits, and the like) used to represent a body of information or set of events. Each piece of information or event is assigned a sequence of code symbols, called a code word. The number of symbols in each code word is its length. The 8-bit codes that are used to represent the intensities in most 2-D intensity arrays contain more bits than are needed to represent the intensities.

Therefore, coding redundancy is the way to reduce the amount of data representing the same information.

Date ______
Page ______

Now,

The total number of pixels = 20 + 30 + 60 + 120 + 130 + 160 + 204 + 300

= 1024

now, lets conver the each gray level into probabilities and arranging them in decreasing order:

Original source | Source reduction

symbol	probability	1	2	3	4	5	6
7	0.29 00	00 0.29	00 0.29	00 0.29	00 0.29	1 0.43	0.58 0
6	0.20 11	11 0.20	11 0.20	10 0.23	01 0.29	0.29 00	0.43 1
5	0.16 010	010 0.16	010 0.16	0.20 11	0.23 10	0.29 01	
4	0.13 011	011 0.13	011 0.13	0.16 010	0.20 11		
3	0.12 100	100 0.12	0.12 100	0.13 011			
2	0.06 1010	0.06 1010	0.11 101				
1	0.03 10110	0.05 1011					
0	0.02 10111						

Figure: Huffman code assignment procedure

The average length of this code is:

$L_{avg} = 0.29 * 2 + 0.20 * 2 + 0.16 * 3 + 0.13 * 3 + 0.12 * 3 + 0.06 * 4 + 0.03 * 5 + 0.02 * 5$

$= 2.70$ bits / pixel

Date ______
Page ______

Q 5(b) Define morphological operation of an image. Explain dialation and erosion with suitable example.

Sol? Morphological operation :

Morphology is a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and the convex hull. Morphological operation in image processing is a collection of non-linear operations related to the shape or morphology of features in an image. Morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images.

Morphological technique probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. Some operations test whether the element "fits" within the neighbourhood, while others test whether it "hits" or intersects the neighbourhood.

A Morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image.

The structuring element is a small binary image i.e. a small matrix of pixels, each with a value of zero or one. The "origin" of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

The fundamental operations:

(i) Dilation:

Dilation is an operation that grows or thickens objects in an image. The specific manner and extent of this thickening is controlled by a set referred to as a structuring element.

The origin of structuring element must be clearly identified by using the both representation (i.e. 0 or 1).

With A and B as sets in Z^2, the dilation of A by B, denoted $A \oplus B$, is defined as

$$\boxed{A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \emptyset \}}$$

The dilation of A by B then is the set of all displacements, z such that B and A overlap by at least one element.

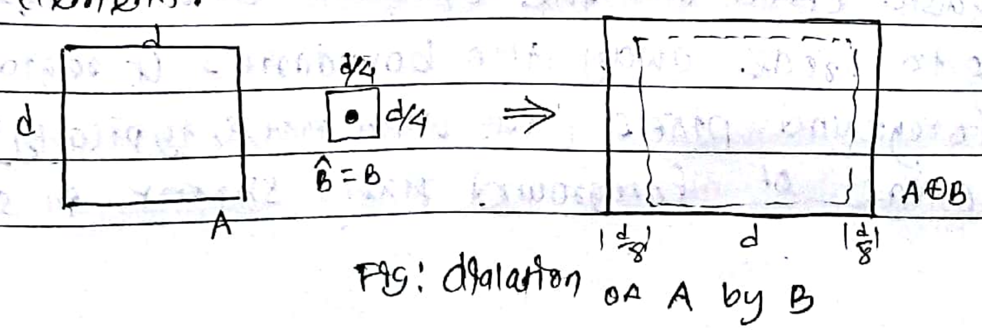


Fig: dilation of A by B

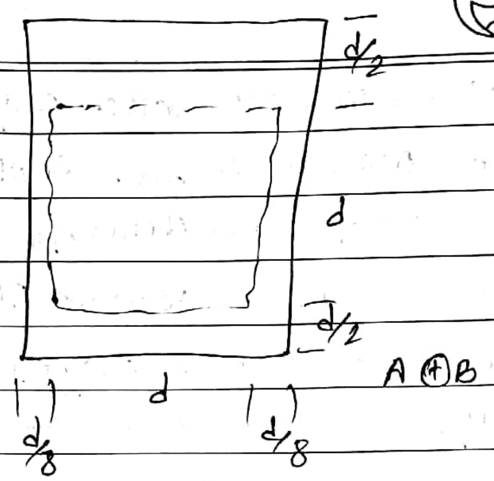


Fig: elongated structuring element

Effect of Dilation:

- Expand the object,
- Both inside and outside borders of the object.
- Dilation fills holes in the object.
- Dilation smooths out the object contour.
- Depends on the structure element.
  - Bigger structure element gives greater effect.

One of the simplest applications of dilation is for bridging gaps in "text with the poor resolution with broken characters.

(ii) Erosion:

Erosion is an operation that thins or shrinks objects in a binary images. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus areas of foreground pixels shrink in size, and holes within those areas become larger.

With A and B as sets in Z^2, the erosion of A by B, denoted $A \ominus B$, is defined as,

$$A \ominus B = \{ z \mid (B)_z \subseteq A \}$$

In words, this equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A.

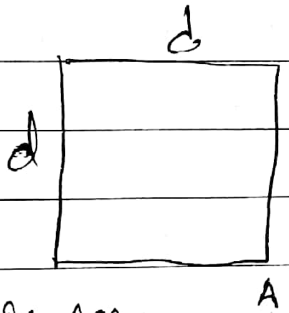


Fig (a)

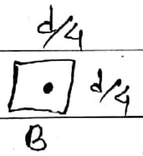


Fig (b)

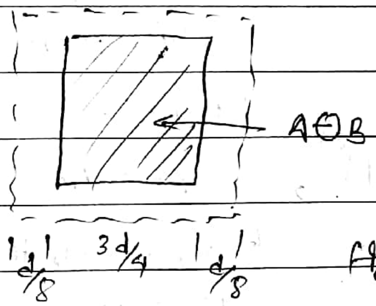


Fig (c)

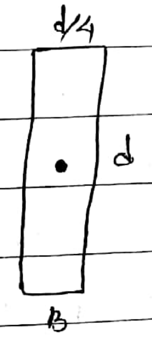


Fig (d)

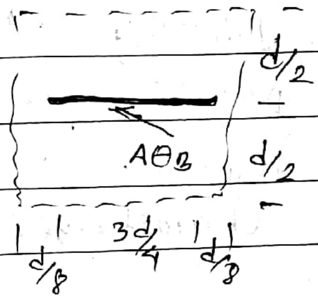


Fig (e)

Figure: (a) set A
(b) square structuring element B
(c) Erosion of A by B, shown shaded
(d) Elongated structuring element
(e) Erosion of A by B using this element.

Q 6(a) What is Image segmentation? Compare prewitt filter with sobel filter with their equations, matrics and characteristics.

Soln →

previously solved in solution 2014 (Fall) 5(a)

Q.6 (b) What is the role of Edge linking in boundary detection? Explain the Hough Transform for detecting lines with suitable figures.

Soln → **Role of Edge Linking in boundary detection:**

- Edge detection is always followed by edge linking.
- Ideally, edge detection techniques yield pixels lying only on the boundaries between regions. In practice, this pixel set seldom characterizes a boundary completely because of
  - → Noise
  - → Breaks in the boundary due to non-uniform illumination.
  - → Other effects that introduces spurious discontinuities.

Thus, edge detection algorithms are usually followed by linking and other boundary detection procedures designed to assemble edge pixels into meaningful boundaries.

Note: For solution of Hough Transform, already solved in solution set 2014 (Fall): 5(b)

Q.7 Write Short notes on: (Any two)

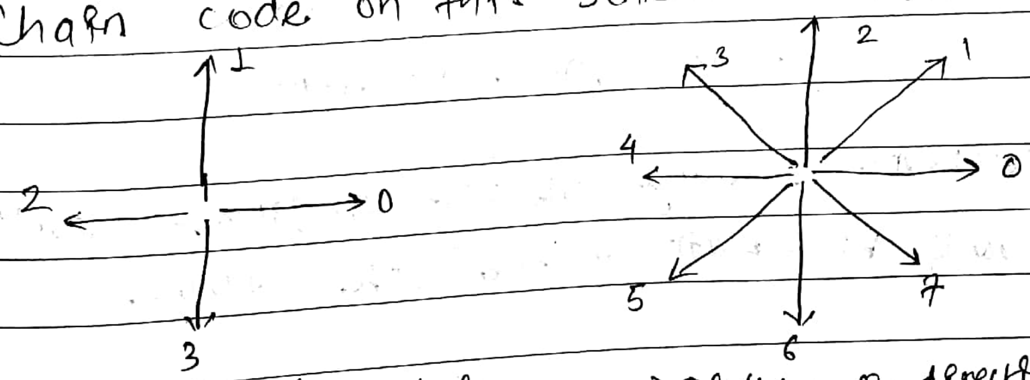
(a) Chain codes:

Chain codes are used to represent a boundary by a connected sequence of straight line segments of specified length and direction.

Typically, this representation is based on 4- or 8- connectivity of the segments.

The direction of each segment is coded by using a number. A boundary code formed as a sequence of such directional numbers is referred to as a Freeman chain code.

→ Chain code on this scheme are referred as:



Direction numbers for → Fig (a): 4-directional chain code

→ Fig(b): 8-directional chain code

The chain code of a boundary depends on the starting point. However, the code can be normalised with respect to the starting point by treating it as a circular sequence of direction numbers and redefining the starting point so that the resulting sequence of numbers is a integer magnitude.

We can normalize by using rotation, i.e. 90° or 45° for the codes using the different chain code.

Q 7(b) <u>perception training algorithm:</u>

- The perceptron is a single layer feed-forward neural network.

-

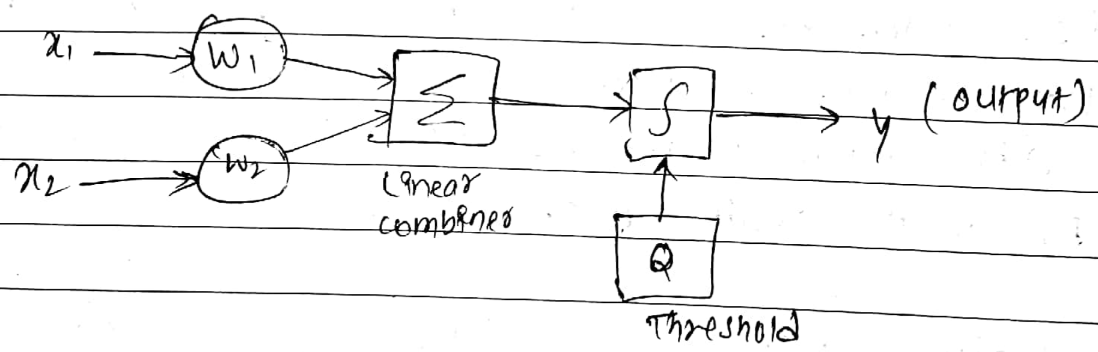


Figure: simple perception

Perceptron training rule was originally developed by Frank Rosenblatt in the late 1950s.

The training patterns are presented to the network's inputs; the output is computed. Then the connection weights W_j are modified by an amount that is proportional to the product of

- the difference between the actual output, y and the desired output, d, and
- the input pattern x.

The algorithm is as follow:

1. Initialize the weights and threshold to small random numbers.
2. present a vector x to the neuron inputs and Calculate the output.
3. update the weights according to:

$$\boxed{W_j(t+1) = W_j(t) + \eta(d-y)x}$$

where, d is the desired output,
t is the iteration number and
eta is the gain or step size, $0.0 < \eta < 1.0$

4. Repeat steps 2 and 3 until:
   - the iteration error is less than a user-specified error threshold or
   - a predetermined number of iterations have been completed.

Date
Page

During training, it is often useful to measure the performance of the network as it attempts to find the optimal weight set.

A common error measure or cost function used is sum squared error. It is computed over all of the input vector / output vector pairs in the training set and is given by the equation below!

$$E = \frac{1}{2} \sum_{i=1}^{p} \| y^{(i)} - d^{(i)} \|^2$$

Where, 'p' is the number of input / output vector pairs in the training set.