UNIT-4 - Image Compression

Applications

1. Tele video conference
2. Remote Sensing
3. Document and medical Imaging
4. facts & fax
5. space and Waste Management Applications

Advantages

1. Reduces transmission Time which Increases transmission speed
2. Bandwidth require for transmission of data can be reduced.
3. Band required for storage is less

Image compression fundamentals

Data

The Data are the means by which the information is conveyed.

→ Data and Information are not Same because Various amounts of data can be used to represent the same information.

Data compression

It refers to the process of reducing the amount of data required to represent the information.

Redundent data

Representations that contain irrelevant (or) repeated information are said to contain redundant data.

If n_1, n_2 denote the number of bits two representations of the same information then the relative data redundancy

$R_D = 1 - \frac{1}{C_R}$ where C_R is compression ratio

$$CR = \frac{n_1}{n_2}$$

If n_1 is equal to n_2 then there is no data redundancy

$R_D = 0$ indicating that there is no data redudancy in 2 representation of the information.

→ If n_2 is very much Less than n_1 then ideally

$$CR = \frac{n_1}{n_2} = \frac{n_1}{0} = \infty$$

(i) $R_D = 1$ which implies highly redundant data is present in first data set

(ii) if $n_2 >> n_1$ then idealy

$$CR = \frac{n_1}{n_2} = \frac{n_1}{\infty} = 0$$

$$R_D = 1 - \frac{1}{CR} = 1 - \frac{1}{0}$$

$$= 1 - \infty = -\infty$$

which implies the second data is having more redandant data.

C_R lies in the range $(0, \infty)$ and R_D lies in the range $(-\infty, 1)$

Types of redundancies

There are four types of data redundancy

1. coding redundancy
2. Inter pixel (or) spatial and Temporal
3. psycho visual redundancy
4. Fidelity criteria

Coding redundancy

a code is a system of symbols (Letters, numbers (or) bits) used to represent the information

code word

Each piece of information is assigned a sequence of code symbols called a code word.

The no.of symbols used in each code word is called it's code Length

Assume that discrete random variable r_K in the interval lies in interval $[0, L-1]$ is used to represent the intensity's of an $M \times N$ image and each r_K occurs with the probability $P(r_K) = \frac{n_K}{MN}$ where no. of pixels in the image (i)

(ii)

n_K is no. of pixels in

If the no. of bits used to represent each r_K is $l(r_K)$ than the avg no. of bits is required to represent each pixel is

$$L_{avg} = \sum_{K=0}^{L-1} l(r_K)\, p(r_K)$$

The total no. of bits required

$$b = MN L_{avg}$$

r_K	$P(r_K)$	Fixed code Length code 1	$l_1(r_K)$	Variable Length code code 2	$l_2(r_K)$
r_1	0.25	01010111	8	10	2
r_2	0.47	1000 0000	8	1	1
r_3	0.25	1100 0010	8	000	3
r_4	0.03	1111 1111	8	001	3

Code 1

$$L_{avg} = \sum_{K=0}^{L-1} l(r_K)\, p(r_K)$$

$= 8(0.25) + 8(0.47) + 8(0.25) + 8(0.03)$

$\Rightarrow$ ~~1.80 bits~~ = 8 bits

code 2

$$L_{avg} = 2(0.25) + 1(0.47) + 3(0.25) + 3(0.03)$$
$$= 1.81 \text{ bits}$$

$$(n_1 >> n_2)\, C_R = \frac{n_1}{n_2} = \frac{8}{1.81} = 4.42$$

$$R_D = 1 - \frac{1}{C_R} = 1 - \frac{1}{4.42} = 77\%$$

Interpixel (or) Spatial and Temporal redundancy

Incase of spatial filtering techniques the response of every pixel depends on its neighbourhood pixels which indicates that the pixel and it's neighbourhood are correlated.

due to this redundancy occurs when these pixels ~~correlated~~ represented with code words, this redundancy type is called inter pixel (or) Spatial redundancy

→ Incase of video Sequence each video is represented with Multiple frames which are correlated in time

→ To reduce interpixel redundancy Associated with spatial and Temporal correlated pixels, a 2-Dimensional intensity array Must be transformed into a non-visual representation ~~into~~

→ for Ex:

Run Length coding

The difference b/w adjascent pixels can be transformed called as Mapping.

Mapping is said to be reversable if the pixels of original 2-D intensity array can be reconstructed without error from the transformed data set.

One-Dimensional Run Length coding

It represents strings of symbols in image matrix, which is used for fax Machines and records only areas that belongs to the object in an image.

2) Image row is described by a sub ist.

a) First element = Row number

b) Subsequent terms are ordinate pairs

c) First element of pair is begining of a run

3) It can have several sequences in each row

psycho visual redundancy

The human eye does not respond with equally Senstivity to all the visual Information, Certain information has less relative importance than other information in visual processing. This information is Said to be psycho visually redundant.

The Elimination of psycho visually redundant data results in the Loss of important information called as quantization.

Quantization is the process of mapping broad range of input Values to the limited values of the output.

As it is irreversible operation quantization is lossy data compression, so to reduce psycho visual redundancy we use IGS quantization (Improved Gray Scale)

IGS quantization procedure

1. The Sum is initially Said to zero
2. The sum is formed from the Current Eight bit gray Level and the four Least Significant bits of previously generated Sum.
3. If the four most significant bits of current Values are 1111 then 0000 is added instead.

4. The four most significant bits of resulting sum are called as IGS coded pixel value.

pixel	Gray level	sum	IGS code
i-1	N/A	0000 0000	N/A
i	0110 1100	0110 1100	0110
i+1	0111 0101 +1100	1000 (MSB) 0001 (LSB)	1000
i+2	1011 0111 0001 1011 1000	1011 1000	1011
i+3	1111 0100 0000 1111 0100	1111 0100	1111

Fidelity Criteria

Removal of irrelevant visual information results loss of important information because the information is lost a mean of quantifying the nature of Loss is required

Two types of Criteria are used for the measurement of loss of information

1. objective fidelity criteria
2. Subjective Fidelity criteria

when the information loss is expressed as mathical function as a input/o/p compression process it is said to be objective fedility criteria.

If $f(x,y)$ is an input Image $\hat{f}(x,y)$ is the approximation of $f(x,y)$ after compressing and subsequent decompressing image

for any (x,y) the Error between the approximated image and the original image is given by $e(x,y)$.

$$e(x,y) = \hat{f}(x,y) - f(x,y)$$

The root mean square error

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y) \right]^2 \right]^{1/2}$$

The rms value of signal to noise ratio is given by:

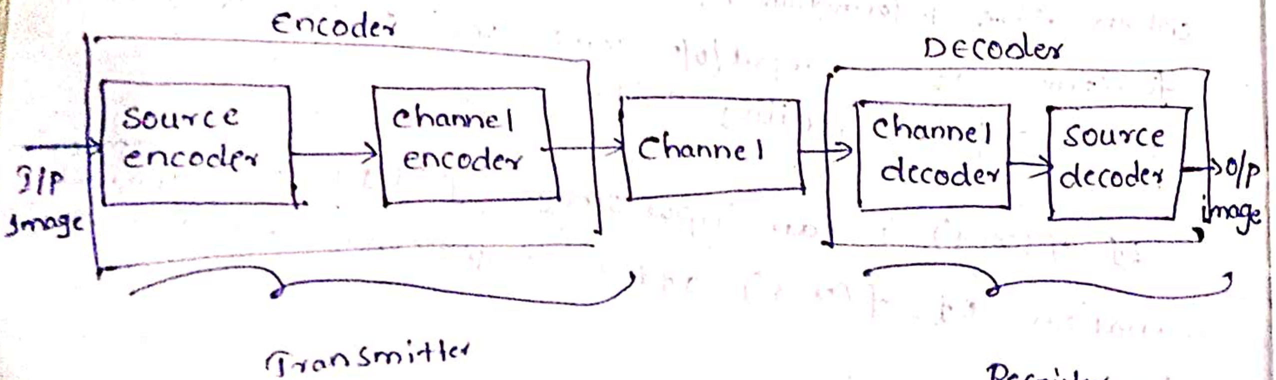
$$SNR_{rms} = \sqrt{\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(\hat{f}(x,y)\right)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y)\right]^2}}$$

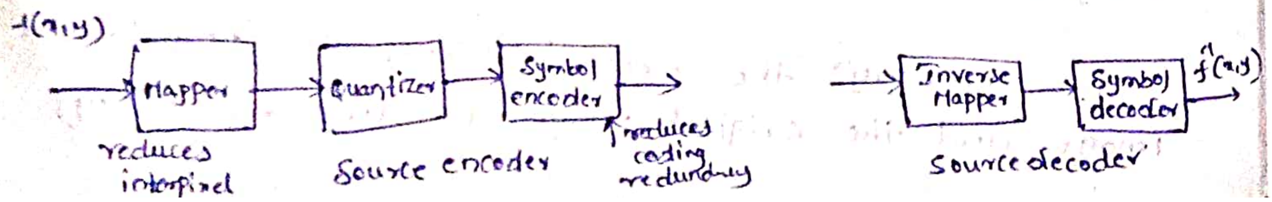
Based on visual perception the evaluation of Image quality can be assed, this assessment is called Subjective fedility criteria, This may be done by means of side by side comparision's of $f(x,y)$ and $\hat{f}(x,y)$

Side by side comparisions can be done with a scale such as $\{-3, -2, -1, 0, 1, 2, 3\}$ to represent the subjective Evaluations {much worse, slightly worse, same, slightly better, better, much better} respectively

2. Imp**

Image compression Model





→ The comp…
an encoder …
→ An Input Im…
set of Sym…
→ The channel
in which …
→ If the c…
decoder …
all types …
Increases
→ The decode…
decoder,
• Source E…

Source En…
redundancie…

Mapper

It transfor…
to reduce

→ The Mappe…
of coeffi…
Ex: …

→ This opera…
→ quantise…
according …
→ This bloc…
image …
→ symbol …
code …
Ex: …

→ This blo…

→ The compression system consisting of two structural blocks an encoder and a Decoder.

→ An Input Image $f(x,y)$ is fed into the Encoder which creates set of symbols from the input data, after tranmission over the channel, the encoded represntation is fed into decoder in which the reconstructed output Image $\hat{f}(x,y)$ is generated

→ If the channel is Error free, the channel Encoder and decoder may be eliminated. Source Encoder removes all types of data redundancy where as channel Encoder Increases noise immunity Source encoders output.

→ The decoder includes channel decoder followed by source decoder,

- Source Encoder

Source Encoder is used for eliminating all types of redundancies from the input image

Mapper

It transforms the input data into non-visual format to reduce interpixel redundancies in the input image.

→ The Mapper transforms the input image into an array of coefficents

Ex: Runlength coding

→ This operation is reversible

→ quantiser reduces the accuracy of Mappers output according to the predefined fedility criteria.

→ This block removes Psycho visual redundancy from input image this operation is irreversible

→ symbol Encoder creates a fixed (or) Variable length code to represent quantizer output

Ex: Variable length code can be used to Encode the quantizer output to assign shortest code word

→ This block reduces coding redundancy.

Huffman coding

1. Arranging the gray level values in decreasing order of their probabilities
2. Combine least two probabilities and generate single probability
3. sort the resulting probabilities again in decreasing (or) deciding order.
4. Repeat from step 2 until only two probabilities remain
5. Assign symbols from reduced probabilities to the original probabilities such that in each assignment upper part is assigned to symbol 0 and Lower part is assigned to symbol 1

gray Levels (r_k)	$P(r_k)$	Step 1				
a_1	0.1	a_2	0.4 (1)	0.4	0.4	0.4
a_2	0.4	a_6	0.3 (00)	0.3	0.3	0.3
a_3	0.06	a_1	0.1 (011)	0.1	0.1	0.2 (01)
a_4	0.1	a_4	0.1 (0100)	0.1	0.2	0.1
a_5	0.04	a_3	0.06 (01010)	0.1		
a_6	0.3	a_5	0.04 (01011)			

Graylevels (r_k)	$P(r_k)$	Step1		Step2 & 3	Repeat Step2	step3	Step2		Step3&4
a_1	0.1	a_2	0.4 (1)	0.4	0.4	0.4 (1)	0.4 (1)	0.4 (1)	0.6 (0)
a_2	0.4	a_6	0.3 (00)	0.3 (00)	0.3 (00)	0.3 (00)	0.3 (00)	0.6 (0)	0.4 (1)
a_3	0.06	a_1	0.1 (011)	0.1 (011)	0.1 (011)	0.2 (010)	0.3 (01)		
a_4	0.1	a_4	0.1 (0100)	0.1 (0100)	0.2 (010)	0.1 (011)			
a_5	0.04	a_3	0.06 (01010)	0.1 (0101)					
a_6	0.3	a_5	0.04 (01011)						

r_k	$P(r_k)$	Codeword	length (L_k)
a_1	0.1	011	3
a_2	0.4	1	1
a_3	0.06	01010	5
a_4	0.1	0100	4
a_5	0.04	01011	5
a_6	0.3	00	2

$L_{avg} = \sum L_k p(r_k)$

$= 3(0.1) + 1(0.4) + 5(0.06) + 4(0.1) + 5(0.04) + 2(0.3)$

$= 0.3 + 0.4 + 0.3 + 0.4 + 0.2 + 0.6$

$= 2.2$ bit/Symbol

$H = -\sum p(r_k) \log p(r_k)$

$= -[0.1 \log(0.1) + 0.4 \log(0.4) + 0.06 \log(0.06) +$
$0.1 \log(0.1) + 0.04 \log(0.04) + 0.3 \log(0.3)]$

$= 0.65$ bits/Symbols

$\% \eta = \frac{H}{L_{avg}} \times 100 = 29.31\%$

Arithmetic Coding

→ It doesn't generate code words for individual Symbols of a message instead it generates a single code for entire message that is to be transmitted hence it is more efficient then Huffman coding and Shanon coding

Procedure

1. Initially start with the interval (0,1).

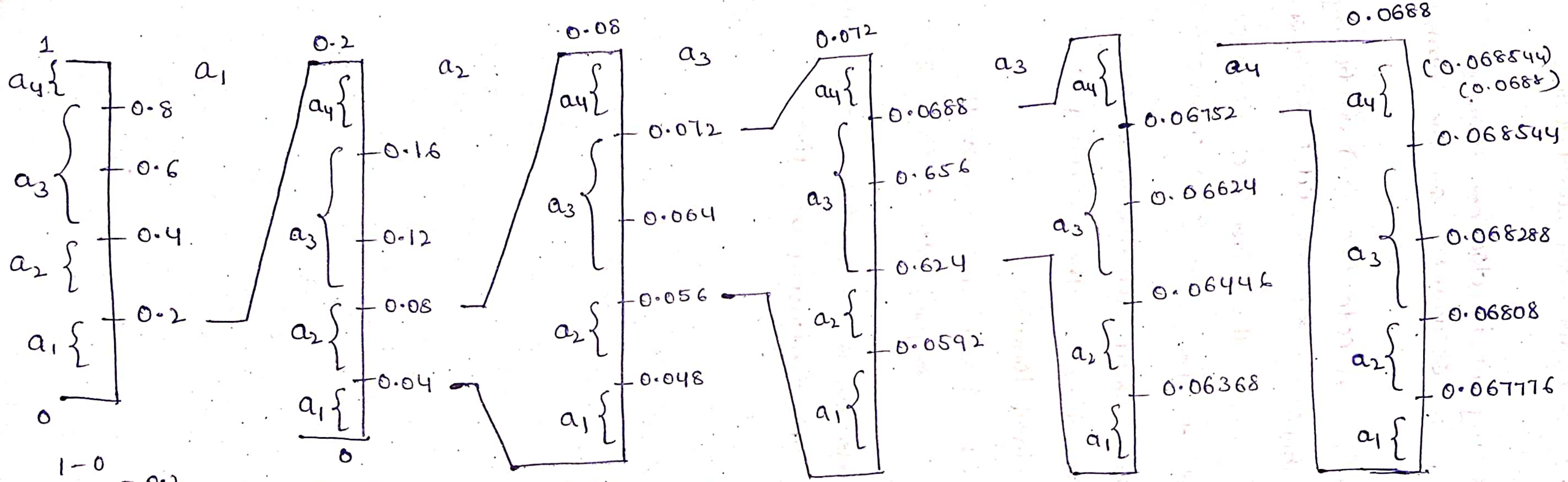
2. Divide the Interval into parts equal to the no. of symbols in the message.

3. Assign the range for each partition based on the probabilities of symbols in the message.

4. Take the sub-interval of current symbol and enlarge it to the full length.

5. Repeat from step 2 until all the symbols in the message are encoded.

6. Once all the symbols are encoded the final sub-interval of last symbol can be used for representing the given message.



$\frac{1-0}{5} = 0.2$

$\frac{0.2-0}{5} = 0.04$

0.04

$\frac{0.08-0.04}{5}$

$= 0.008$

0.056

$\frac{0.0032-0.056}{5}$

$= 0.0032$

0.0624

$\frac{0.0688-0.0624}{5}$

$= 0.00128$

0.06752

$\frac{0.0688-0.0698}{5}$

$= 0.000256$

Lempel-Ziv-Welch coding (LZW) (or) Dictionary Based coding

The key feauture of LZW coding is it does not require prior knowledge of probability of occurance of the symbols to be encoded.

→ At the start of coding process, A code book (or) dictionary containg the source symbols to be coded is constructed

→ For a 8-bit monochrome image the first 256 words of the dictonary are assigned to the intensity's from 0-to 255

→ As the encoder examines sequentially all the image pixels the intensity sequences that are not in the dictionary are placed in algorithmically determined locations

→ If the first two pixels of the image are white, the sequence 255-255 might to be assigned to location 256

☆ → consider a 4x4 eight bit image of a mono-chrome image having the intensity as given

$$\begin{bmatrix} 39 & 39 & 126 & 126 \\ 39 & 39 & 126 & 126 \\ 39 & 39 & 126 & 126 \\ 39 & 39 & 126 & 126 \end{bmatrix}$$

Currently recognized sequence	Pixel being processed	Encoded O/P	Dictionary location (Codeword)	Dictionary entry
	39 (1st pixel)			
39	39 (2nd pixel)	39	256	39-39
39	126 (3rd)	39	257	39-126
126	126 (4th)	126	258	126-126
126	39 (5th)	126	259	126-39
39	39 (6th)	39	256	39-39
39-39	126 (7th)	256	260	39-39-126
126	126 (8th)	126	258	126-126
126-126	39 (9th)	258	261	126-126-39
39	39 (10th)	39	256	39-39
39-39	126 (11th)	256	260	39-39-126
39-39-126	126 (12th)	260	262	39-39-126-126
126	39 (13th)	126	259	126-39
126-39	39 (14th)	259	263	126-39-39
39	126 (15th)	39	257	39-126
39-126	126 (16th)	257	264	39-126-126
126		126		

Bit-Plane coding :

The technique Called Bit-plane coding is based on the concept of decomposing an image into Series of Binary Images and compressing each binary image using any one of the image Compression methods

Bit-plane decomposition

The Gray levels of an m-bit gray scale image can be represented in the form of base 2 polynomial as $a^{m-1}2^{m-1} + \dots + a^2 2^2 + a^1 2^1 + a^0 2^0$
based on this property a simple method of decomposing the image into a connection of binary images is to Separate the 'm'-1-bit planes the 'm' coefficients of the polynomial into 'm'-1-bit planes the Lowest order bit plane is generated by collecting a_0 bits of each pixel and the higher order bit plane is generated by collecting a $m-1$ bits of each pixel

Disadvantage

1. Small change in gray Level can have significant impact on the Complexity of bit planes

   Ex:- If the pixel with the intensity 127 is represented by 127 ⇒ (0.1111111) when intensity value is changed to 128 ⇒ (10000000) bit plane 7 is changed from 0 to 1 and bit-plane 6 to 0 will transition 1 to 0 ⊕

An alternative decomposition is to represent the image by m-bit graycode

The m-bit gray code

$$g_{m-1}2^{m-1} + g_{m-2}2^{m-2} + \dots + g_1 2^1 + g_0 2^0$$

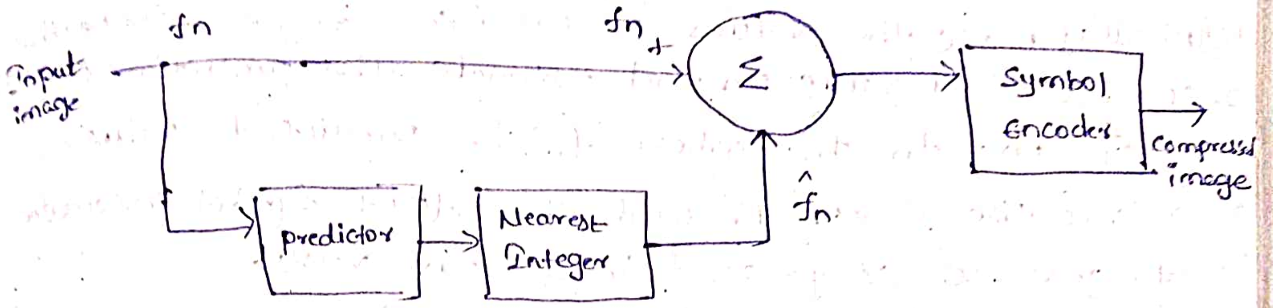
$$g_{m-1} = a_{m-1}$$

$$g_{m-2} = a_{m-1} \oplus a_{m-2}$$

$$g_1 = a_2 \oplus a_1 \quad g_0 = a_1 \oplus a_0$$

The gray code has unique property successive code words different only one bit position small change in gray level have effect

Loss-Less predictive coding



Encoder

→ It is based on eliminating the interpixel redundancies of closely spaced pixels by extracting and coding only the new information in each pixel.

→ The new information of a pixel is defined as the difference between actual and predicted values of that pixel.

→ The system consist of encoder and a decoder with identical predictors.

→ As the each successive pixel of the input image denoted by f_n is introduced into the encoder, predictor generates approximate value of the pixel based on same number of past inputs.

→ The output of integer is rounded off nearest integer is denoted by $\hat{f}_n$ and is use to find difference (or) prediction error, denoted by e_n

$$e_n = f_n - \hat{f}_n$$

which is coded by the symbol encoder using one of the variable length coding.

→ ~~The decoder reconstructs~~ Symbold-

→ The symbol decoder reconstructs e_n from the compressed image and performs inverse operation.

$$f_n = e_n + \hat{f}_n$$

→ The predictor is formed by the Linear Combination of Previous pixels.

where α_i is called prediction coefficient

procedure.

1. Initially make the Values of predictors as 'zero', Take the
2. Current pixel value f_n and estimate the prediction error using $e_n = f_n - \hat{f}_n$ where $\hat{f}_n$ is predicted value.
3. → code the Error e_n with the help of symbol encoder to generate compressed image pixel value.
4. Transmit generated code word to the decoder
5. decode the codeword to retrieve the Error value e_n
6. Add prediction Error with predicted value $\hat{f}_n$ to generate the decompressed image pixel value.
7. update the predictors with previously generated pixel value.
8. Repeat from Step 2 until all the pixels are encoded (or) decoded.

Ex- {16, 48, 120, 238}

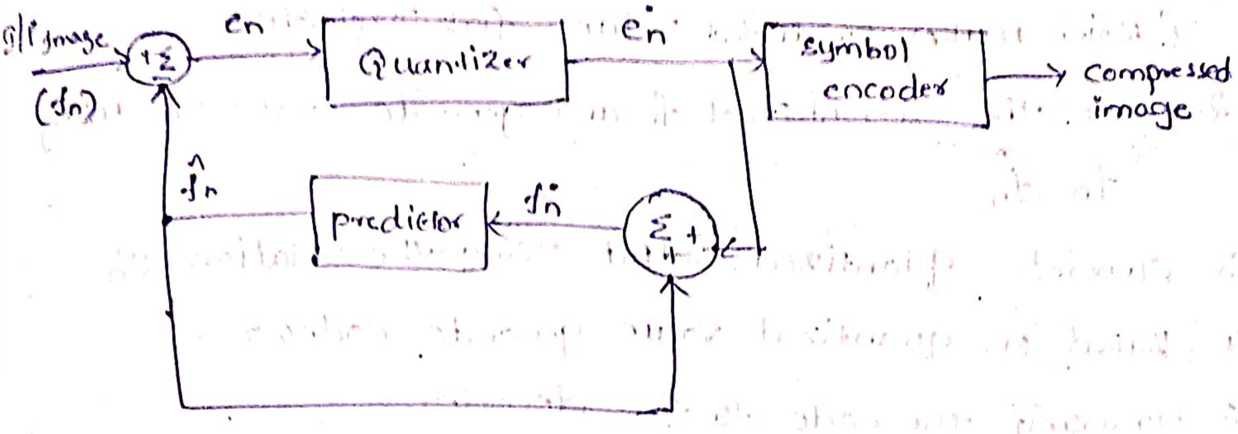
Encoder

f_n	$\hat{f}_n$	$e_n = f_n - \hat{f}_n$	encoder O/P codeword
16	0	16	codeword of 16
48	16	32	Codeword of 32
120	48	72	code word of 72
238	120	118	Codeword of 118

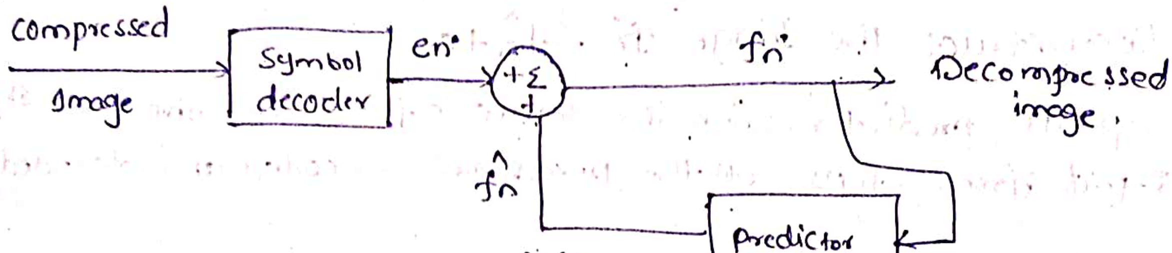
Decoder

encoder o/p	e_n	$\hat{f}_n$	f_n
codeword 16	16	0	16
codeword 32	32	16	48
codeword 72	72	48	120
codeword 118	118	120	238

Lossy predictive coding / Lossy compression / Error Compression



Encoder (a)



(b)

→ Lossy compression is based on comprising the accuracy of reconstructed image in exchange for increased compression

→ The quantizer works as nearest integer function of the Error free Encoder, which is inserted between symbol Encoder and the point at which prediction Error is formed.

→ It maps the prediction Error into a limited range of outputs denoted by $\dot{e}_n$ which establish the amount of compression and destortion associated with Lossy predictive coding.

From the block diagram where $\hat{f_n}$ is given by

$$\hat{f_n} = \text{round}\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right]$$

The closed loop ~~formatio~~ configuration prevents Error at decoder output.

$$e_n = \begin{cases} +6.5, & e_n \geq 0 \\ -6.5, & e_n < 0 \end{cases}$$

Assumptions

1. Let the quantizer is operated as follows.
1. Initially Make Error free Compression for the first pixel (which makes predicted Value = first pixel value
2. Take the current pixel f_n and generate error e_n using $f_n - \hat{f_n}$
3. Generate quantized output using the relation $\dot{e_n}$
4. Based on quantized value generate codeword
5. Transmit the code to the decoder.
6. Retrieve $\dot{e_n}$ from the codeword with the help of symbol decoder.
7. Reconstruct the Image $\dot{f_n} = \hat{f_n} + \dot{e_n}$
8. update predictor with the value of $\dot{f_n}$ for the next step.
9. Repeat from step2 all the pixels are encoded and decoded.

$f_n = \{16, 16, 16, 40, 120, 74, 74\}$

Encoder

f_n	$\hat{f_n}$	$e_n = f_n - \hat{f_n}$	$\dot{e_n}$	$\dot{f_n}$
16	16	0	6.5	22.5
16	22.5	-6.5	-6.5	16
16	16	0	+6.5	+22.5
40	22.5	+17.5	+6.5	29
120	29	91	+6.5	35.5
74	35.5	38.5	+6.5	42
74	42	32	+6.5	48.5

$$\dot{e_n} = \begin{cases} +6.5, & e_n \geq 0 \\ -6.5, & e_n < 0 \end{cases}$$

$$\dot{f_n} = \dot{e_n} + \hat{f_n}$$

Decoder

f_n

codeword	e_n'	$\hat{f_n}$	$f_n' = e_n' + \hat{f_n}$	error
1	+6.5	16	+22.5	-6.5
0	-6.5	+22.5	+16	0
1	+6.5	16	+22.5	-6.5
1	+6.5	+22.5	29	11
1	+6.5	+29	35.5	84.5
1	+6.5	+35.5	42	32
1	+6.5	+42	48.5	25.5

UNIT-5 Morphology and compression

a) Image Segmentation

- Based on Discontinuities 14M
  - point detection
  - Line detection
  - Edge detection
- Based on Similarities
  - Thresholding 11M
  - Region - Based 14M

b) Morphological Image processing

- Dilation & Erosin } 14M
- Some morphological algorithms (14m
- Opening & Closing } 14M
- Hit (or) Miss transform } 8M

Image Segmentation

→ It is a process of dividing the Entire Image into Subparts based on discontinutes and similarities

Image Segmentation based on Segmentation

Dis Segmentation Of the Image is done based on abrupt change in the intensity values of the pixels in an image this technique involves defining a mask of 3x3 having the coefficients $\omega_1 \omega_2 ---- \omega_9$ and applying the mask on every pixel of an image and calculate the response of each pixel as the mask is applied.

for a 3x3 image having the pixel values $z_1, z_2 --- z_9$

will have the response of the mask on z_5 can be return as

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

3x3 image

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

3x3 masks

$$R = w_1 z_1 + w_2 z_2 + w_3 z_3 + \cdots + w_9 z_9$$
$$= \sum_{i=1}^{9} w_i z_i$$

a. point detection

The mask that can be applied for the detection of isolated points in an image is given by

-1	-1	-1
-1	8	-1
-1	-1	-1

1. Project the above mask on each and every pixel in the of image and in each case calculate the response r_i.
2. Define a thresh hold limit 'T' for the detection of a point
3. Identify the responses from step1 which satisfy the condition $R_i \geq T$
4. A point is said to be located for the pixels whose responses are ≥ 3
5. The mask used for point detection is same as image sharping.

Line detection

The masks ~~whi~~ that are defined for the detection of lines for the image are given by.

R_H

-1	-1	-1
2	2	2
-1	-1	-1

a) Horizontal Line

R_V

-1	2	-1
-1	2	-1
-1	2	-1

(b) Vertical Line.

R_{+45}

-1	-1	2
-1	2	-1
2	-1	-1

c) $+45^0$

R_{-45}

2	-1	-1
-1	2	-1
-1	-1	2

d) -45^0

Case-i

detection of orientation of image in a Line

1. project each of four masks on the image independantly and calculate the responses.
2. Let the responses are Labelled as:

   R_H R_V $R(45)$ $R(-45)$
3. Identify the maximum response out of above 4 responsey based on the max response orientation of Line detected corresponding to that mask.

(Case-ii)

detection of Number of Lines in a particular direction

1. Take a specific mask from the above four based on the given direction
2. Run the mask over the Entire image
3. Calculate the response on each run and identify the no. of positive responses.
4. The no. of positive responses = The no. of Lines

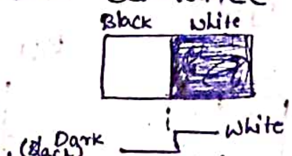
Edge detection

Edges in an image can be detected with the help of first order and second order derivatives.

There are two types of edge models based on their intensity profiles.

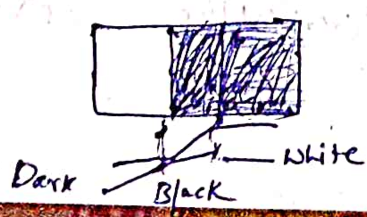
1. Step edge

It Involves the transition between two intensity levels occuring Ideally over the distance of one pixel



2. Ramp Edge

In this model we don't have a thin path instead an edge point any point contained in the Ramp and a edge segment is set of such points that are connected.



Edge detection using first order derivative

gradient

first order derivative on an image provides '0' output in the areas of constant gray Level and produces Non-Zero values at the points, starting and. Ending from the Ramp pages and also produces constant Value along the ramp

For Finding edge strength and direction at co-ordinates (x,y) of an image f the gradient of 'F' is denoted by

This gradient points in the direction of greatest rate of change of F at Location (x,y)

The Magnitude of Gradient

$$mag(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

The first order derivative of an image can be calculated as follows

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

$$g_x = f(x+1,y) - f(x,y)$$

$$g_y = f(x,y+1) - f(x,y)$$

$$\text{Gradient}(f) = g_x + g_y$$

$$\text{gradient}(f) = \underbrace{[f(x+1,y) - f(x,y)]}_{g_x} + \underbrace{[f(x,y+1) - f(x,y)]}_{g_y}$$

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

0	0	0
0	-2	1
0	1	0

Gradient Operator

1. Roberts Cross Gradient Operator

-1	0
0	1

0	-1
1	0

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

3x3 image

$g_x = z_9 - z_5$

$g_y = z_8 - z_6$

Considering a 3x3 image region derivatives can be obtained by filtering the image with the mask given

2. Prewitt Gradient operator

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

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

$$g_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

3. Sobel gradient

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

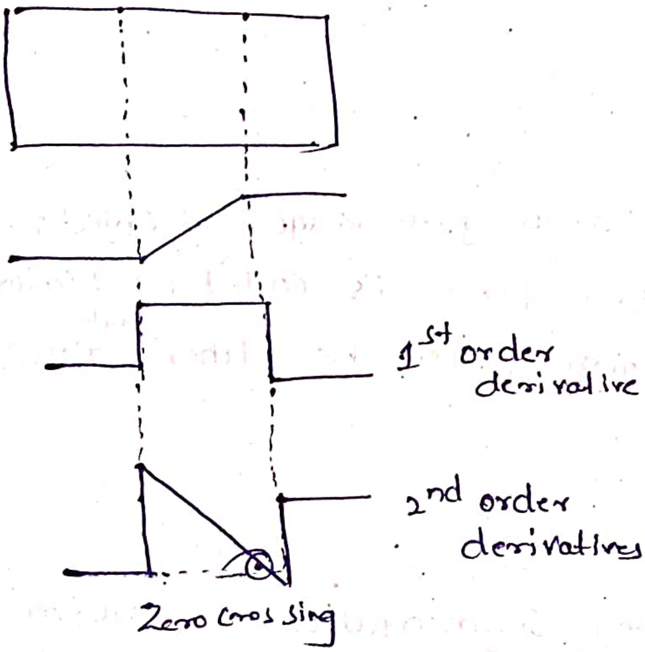
-1	0	1
-2	0	2
-1	0	1

$$g_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Edge detection using second order derivative.

The second order derivative of an image produces 0 output along the areas of constant gray Level and also along the ramp, but it provides a positive spike at the onset of dark to bright transition and a negative spike at the onset of bright to drak transition



The Second order derivative produces two values for every Edge

Positive Spike

Negative Spike

The Straight Line joining these two spikes will cross zero gray Level at certain point is called ZeroCrossed point and it gives the Location of edge in the Image

Drawback

1. It produces two pixel thick Edge
2. It is Sensitive to Noise
3. The Lapalician of the input Image Can be Calculated as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

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

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

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

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x,y).$$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1,y) + f(x-1,y) - 2f(x,y) + f(x,y+1) + f(x,y-1) - 2f(x,y)$$

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

The process of partioning the image into objects based on Similarity between pixels is called as similarity based image Segmention. Similarity can be either grey level values, texture (or) color.

Thresholding

This method of segmentation is prefreed whenever Separating the objects from background is required.

Thresholding is divided into 2 types based on no. of thresholds used for segmenting the image.

There are two thresholding techniques

1. single thresholding
2. Multiple thresholding

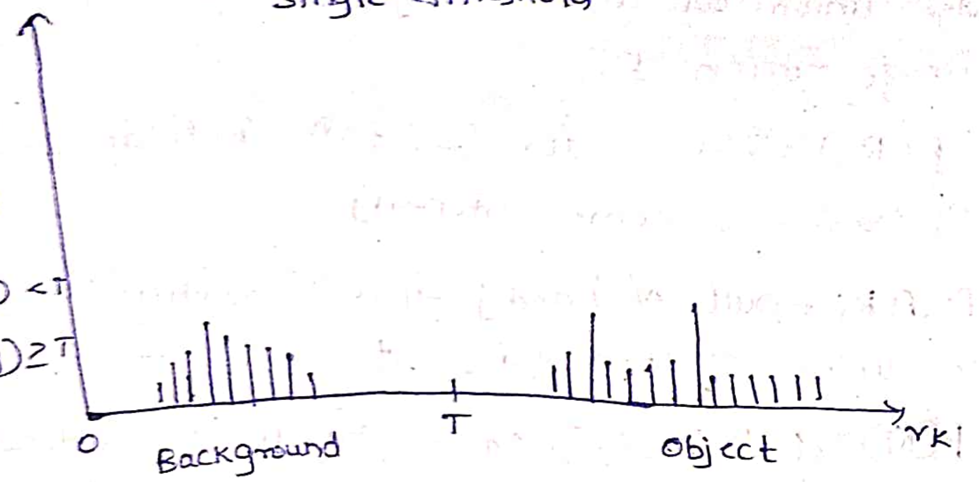
Single thresholding

Single threshold

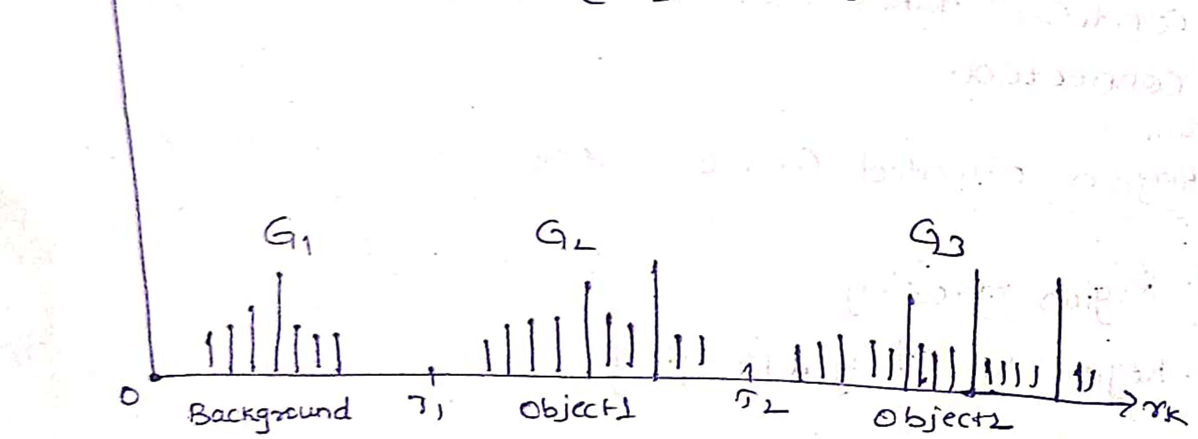
Let $f(x,y)$ is i/p image

$g(x,y)$ is o/p image

$$g(x,y) = \begin{cases} 0, & f(x,y) < T \\ 1, & f(x,y) \geq T \end{cases}$$



$$g(x,y) = \begin{cases} G_1 & \text{if } f(x,y) \leq T_1 \\ G_2 & \text{if } T_1 < f(x,y) < T_2 \\ G_3 & \text{if, if } (x,y) \geq T_2 \end{cases}$$



This method is preferred when a single class of objects mustly partioned on the background of an Image $f(x,y)$

Multiple

This method is preferred when multiple ~~single~~ class of objects must be partioned on the background of an Image $g(x,y)$

Region orientation Segmentation

Basics

The process of Segmenting the image into Subregions based on specific Criteria is called region oriented Segmentation.

The following are Necessary conditions for segmenting the Image into Subregions.

1. The union of all Subregions must form the original image region 'R'.

2. $P(R_i) = TRUE$ for $i = 1$ to n indicates that if all pixels in R_i have the same intensity

3. $R_i \cap R_j = null$ $\forall$ i and j this condition indicates that the conditions must be disjoint

4. $P(R_i)$ ~~Y~~ $P(R_i \cup R_j) = False$ for all $i \neq j$ indicates that region R_i and R_j are different in the sense of their probability.

5. R_i is a connected region for all $i = 1, 2, 3 \ldots n$ this condition takes that all the pixels in the region R_i must be connected.

*imm

Region oriented can be done in two methods

1. Region growing
2. Region spilitting & Merging

Region growing

1. Consider a seed pixel
2. Start appending the neighbor pixels that satisfy homogenity criterion with the seed pixel.
3. Stop appending when further appending is not possible.
4. Choose a new seed pixel for the remaining pixel that are not appended.
5. Repeat steps 2, 3, 4 until No pixel is left without appending.

Ex:

R ⇒

0	1	7	5	3
1	0	5	3	7
7	3	0	1	5
5	7	1	0	3
7	5	3	1	0

consider seed pixel as 7

Now append the pixels based on a condition that the difference wrt seed pixel must be Lessthan (<3)

Region 1

Region 2

3	1	0	5	7
0	0	0	5	7
0	1	3	5	7
1	1	3	5	7
1	3	3	5	7

consider new seed pixel as 3 with the condition that the difference wrt seed pixel must be Lessthan (or) equal to (≤ 2)

Region3

Region 1

Region2

0	0	3	5	7
0	0	3	5	7
0	3	3	5	7
1	3	1	5	7
1	1	1	5	7

4) Seed pixel 1

Diff wrt seed pixel ≤ 1

0	0	3	5	7
0	0	3	5	7
0	3	3	5	7
1	3	1	5	7
1	1	1	5	7

5) Seed pixel = 5

diff wrt seed pixel ≤ 1

0	0	3	5	7
0	0	3	5	7
0	3	3	5	7
1	3	1	5	7
1	1	1	5	7

1. The techinique is Very simple
2. It clearly defines the region boundaries

disadvantege

If seed pixel is choosen wrong the segmentation gives improper result

Region spliting and Merging

1. Blind the split the image into four regions (or) four sub regions once splitting into rows and once splitting along columns.
2. Identity the regions with Homogenity
3. split non-homogenious into four sub regions
4. Continue splitting
5. Identify the Images with homogonity property and merge them.
6. Stop merging until further merging is not possible.

Morphological Image processing :

Morphology

It is a Mathematical tool which deals with the Shape and Structure of a region such as boundaries, Skeletons

→ The Language of Mathematical Morphology is the Set theory.

Sets represents objects in an image, these sets are members of twodimensional integer space Z^2 where each element of a set is a two dimensional Vector whose coordinates are (x,y) of pixels in the Image

Set

Let 'A' be a Set in twodimensional integer Z^2 and if $A = \{a_1, a_2\}$ then $a_1 \in A$ and $a_2 \in A$ if a_1 is not an Element of A then it is denoted by $a_1 \notin A$

Subset

If every element of set 'A' is also an element in set B then $A \subseteq B$ then A is Said to be Subset of B.

Union

The union of two sets A and B is the Set of Elements which are taken from both sets A and B

Intersection

It is the Set of all Elements belongs the Sets A and B

Disjoint Sets

Two Sets A and B is disjoint where $(A \cap B) = \phi$

Complement

The Complement of Set 'A' is the Set of Elements not Contained in 'A' $A^c = \{ w / w \notin A \}$

Difference

The difference of two Sets A and B denoted by $A - B$ is a set of Elements that belongs to A but not to B

$(A-B) = \{ w / w \in A, w \notin B \}$

Reflection

The Reflection of Set B, denoted by $\hat{B}$ is Set of Elements of vector 'w' such that $W = -B$ for all $b \in B$

$$\hat{B} = \{ w / w = -b \text{ for all } b \in B \}$$

Translation

The Translation of Set A by a point $z = (z_1, z_2)$ denoted by. $(A)_z = \{ c / c = a+z, \text{ for all } a \in A \}$

Morphological Transformations

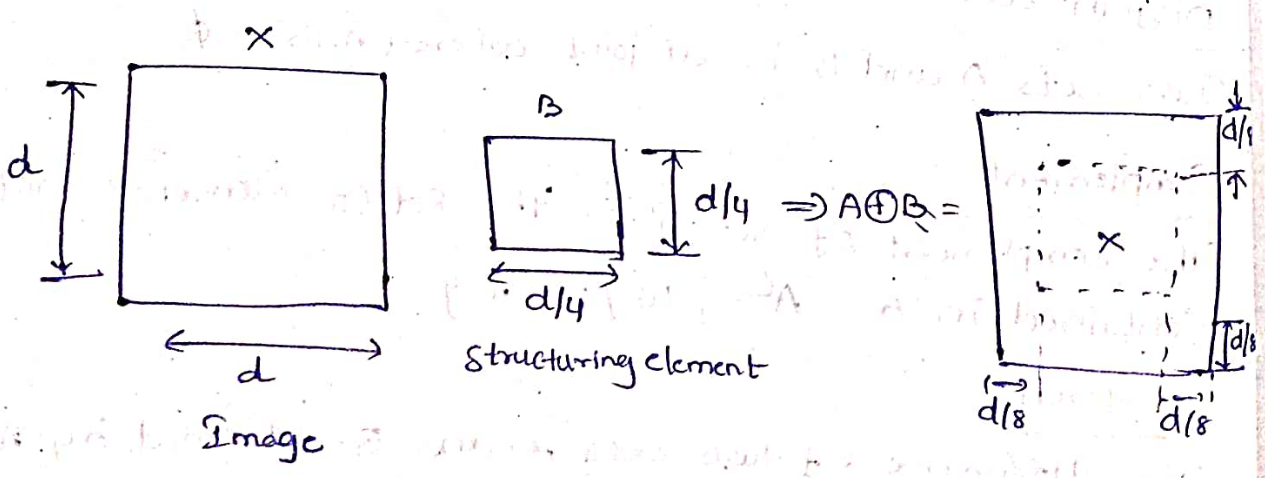
It gives the relation of an Image X with another small point 'B' called as Structuring Element.

Some Morphological operations

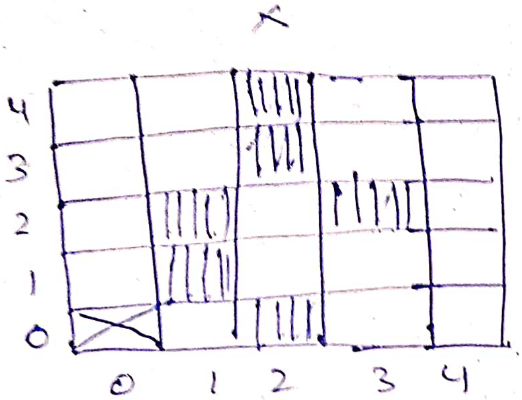
(i) Dilation

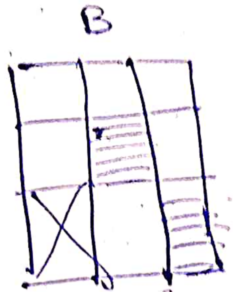
If two Sets X and B are in two dimensional integer Space Z^2 then the direction of X by B is denoted by

$X \oplus B = \{ P / P = x+b, x \in X, b \in B \}$ i.e. every element of X and B are added results a Set of Elements called as dilated Set.



From the above example we can conclude that dilation of an Image results Expansion of an Image with Improved thickness





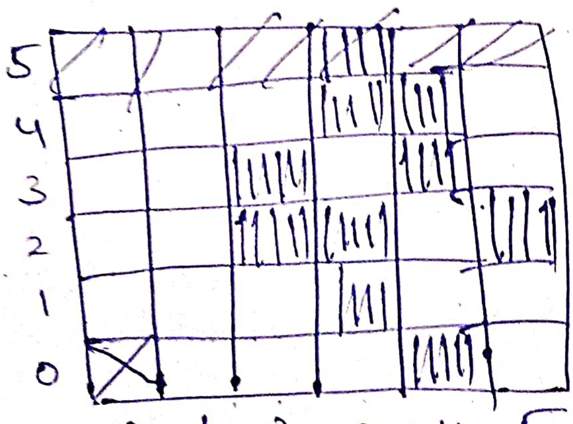
$B = \{(1,1), (0,2)\}$

~~$X = \{(0,2), (1,1), (2,1), (2,3), (3,2), (4,2)\}$~~

$X \oplus B = \{$

$X = \{(1,1)\ (1,2)\ (2,0)\ (2,3)\ (2,4)\ (3,2)\}$

$X \oplus B = \{(2,2)\ (2,3)\ (3,1)\ (3,4)\ (3,5)\ (4,3)\ (3,1)$
$(3,2)\ (4,0)\ (4,3)\ (4,4)\ (5,2)\}$



BEE :

Ground water

1. Surface Water from upland and mountain Streams and from Large Lakes Comparitively unpolluted.
2. Groundwater free from any pollution but containing Iron and Mangenese are other dissolved gases like CO_2.
3. Surface water from lakes and Strings Some what turbit may not be polluted bacteria.
4. Surface water from turbid Streams generally polluted Sometimes very heavy.
5. Ground water Containg disloved Salts of Calcium and Megnisum in Excessive limits

input Level rk to the output level

and draw ble.

3

4

3

HSI to RGB

(i) RG Sector

$0^\circ \le H < 120^\circ$

$H = H - 0^\circ$

$B = I(1-S)$

$$R = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$G = 3I - (R+B)$

(ii) GB Sector

$120^\circ \le H < 240^\circ$

$H = H - 120^\circ$

$R = I(1-S)$

$$G = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$B = 3I - (R+G)$

(iii) BR Sector

$240^\circ \le H \le 360^\circ$

$H = H - 240^\circ$

$G = I(1-S)$

$$B = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$R = 3I - (G+B)$

RGB to HSI

$$R = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$Z = \frac{Z}{X+Y+Z}$$

$X+Y+Z = 1$

$$I = \frac{1}{3}(R+G+B)$$

$$S = 1 - \frac{3}{(R+G+B)}$$

$$H = \begin{cases} \theta & B < G \\ 360 - \theta & B \ge G \end{cases}$$

$$\theta = \cos^{-1}\left[\frac{\frac{1}{2}(R-G) + (R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}}\right]$$

RGB $\rightarrow$ HSI

(29, 104, 215)

X Y Z

$x = R$

$y = G$

$z = B$

$$R = \frac{29}{29+104+215} = \frac{29}{348} = 0.083$$

$$G = \frac{104}{29+104+215} = \frac{104}{348} = 0.299$$

$$B = \frac{215}{29+104+215} = \frac{215}{348} = 0.618$$

Histogram Equilization

1. Find the Histogram of gray Levels of the Input Image by making the table of r_k (gray levels) and No. of Pixels with that gray level. n_k

2. For Each input gray level Calculate the cumulative sum $\sum_{i=0}^{k} n_i$ by adding no. of pixels for Each gray Level

3. Divide the Cumulative Sum by the total no. of Pixels in the Image. $\dfrac{\sum_{i=0}^{k} n_i}{n}$

4. Scale the output by Multiplying the Value obtained in Step3 with the Maximum grey Level Value.

5. Round off the output of Step-4 to the nearst integer Value.

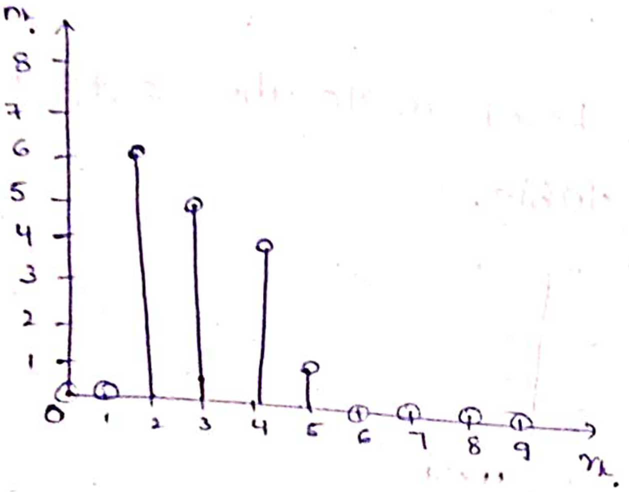
6. Map the input gray Level r_k to the output level s_k and draw the table.

$$\begin{bmatrix} 2 & 3 & 3 & 2 \\ 4 & 2 & 4 & 3 \\ 3 & 2 & 3 & 5 \\ 2 & 4 & 2 & 4 \end{bmatrix}_{4\times4}$$

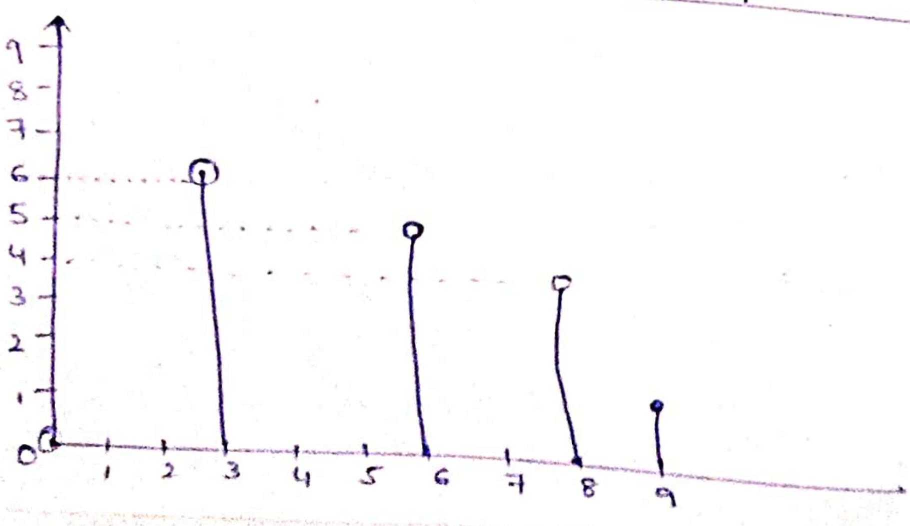
For the given 4x4 Image having the gray levels between [0,9] plot histogram of the Equalized Image.

r_k	n_k	$\sum_{i=0}^{k} n_i$	$\sum_{i=0}^{k} \frac{n_i}{n}$	$\left[\sum_{i=0}^{k} \frac{n_i}{n}\right] * r_{k(max)}$	
0	0	0	$\frac{0}{16}=0$	0	0
1	0	0	$\frac{0}{16}=0$	0	0
2	6	6	$\frac{6}{16}=0.37$	$0.37\times 9=3.33$	3
3	5	11	$\frac{11}{16}=0.68$	$0.68\times 9=6.12$	6
4	4	15	$\frac{15}{16}=0.93$	$0.93\times 9=8.37$	8
5	1	16	$\frac{16}{16}=1$	$1\times 9=9$	9
6	0	16	$\frac{16}{16}=1$	$1\times 9=9$	9
7	0	16	$\frac{16}{16}=1$	$1\times 9=9$	9
8	0	16	$\frac{16}{16}=1$	$1\times 9=9$	9
9	0	16	$\frac{16}{16}=1$	$1\times 9=9$	9

Step-1



r_k	0	1	2	3	4	5	6	7	8	9
S_k	0	0	3	6	8	9	9	9	9	9



②

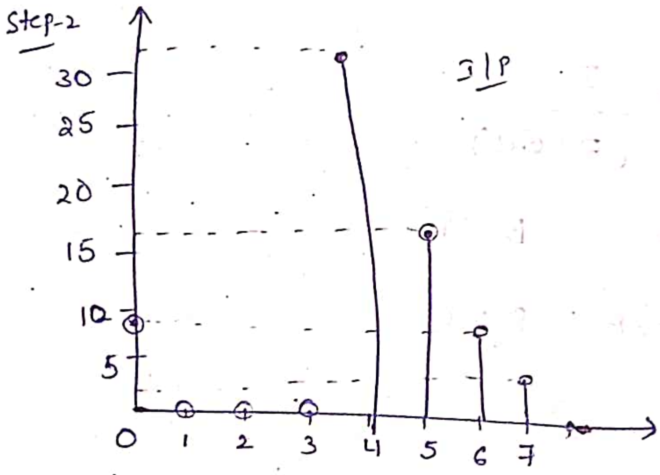
r_k	0	1	2	3	4	5	6	7
n_k	8	0	0	0	31	16	8	1

(0,7)

$\Rightarrow n = 64$

r_k	0	1	2	3	4	5	6	7
n_k	8	0	0	0	31	16	8	1
$\sum_{i=0}^{k} n_i$	8	8	8	8	39	55	63	64
$\sum_{i=0}^{k} \frac{n_i}{n}$	$\frac{8}{64}=0.11$	$\frac{8}{64}=0.11$	$\frac{8}{64}=0.11$	$\frac{8}{64}=0.11$	$\frac{39}{64}=0.61$	$\frac{59}{64}=0.86$	$\frac{63}{64}=0.98$	$\frac{64}{64}=1$
$\sum_{i=0}^{k} \frac{n_i}{n} \times r_k(max)$	0.87	0.87	0.87	0.87	4.27	6.02	6.86	7
rounded off S_k	1	1	1	1	4	6	7	7

Step-2



r_k	0	1	2	3	4	5	6	7
S_k	1	1	1	1	4	6	7	7