Image Compression

By

Kiran TALELE

talelesir@gmail.com

99 870 30 881

Kiran TALELE

@ Bharatiya Vidya Bhavans' Sardar Patel Institute of Technology Andheri(w) Mumbai

- **Associate Professor,** Electronics Engineering Department (1997)
- **Dean,** Students, Alumni & External Relations (2022)

@ Sardar Patel Technology Business Incubator(SP-TBI), Funded by Department of Science & Technology(DST), Govt. of India

- **Head**, Academic Relations (2015)

@ IEEE Bombay Section

•**Treasurer** (2020)
•Executive Committee Member (2015)

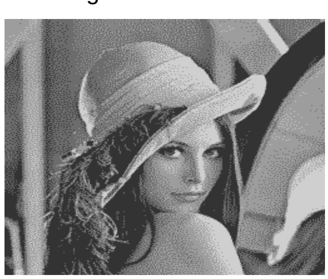
Image Compression

- Reduces the number of bits required for storage
- Size of the image on Hard Disk is reduced

◆ **Coding is essentially representing the information in a more efficient and robust manner**

Original BPP = 8

BPP = 4

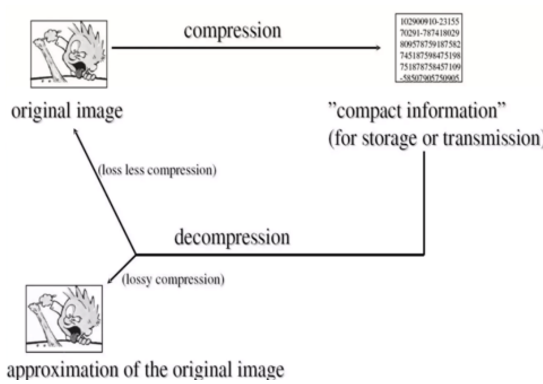


BPP = 2

BPP = 1

Types of Compression Strategies :

1. **Lossless Compression Techniques**
2. **Lossy Compression Techniques**



Types of Compression Strategies :

1. **Lossless Compression Techniques :**

   1. Arithmetic coding
   2. Huffman Coding
   3. Run-Length Coding
   4. Bit plane Coding : CAC
   5. Lossless Predictive coding : DPCM

Removes Redundancies

e.g. Medical Images, Satellite images etc

2. Lossy Compression Techniques :

Useful as very high compression ratios can be achieved with insignificant amount of data loss.

1. Improved Gray Scale Quantization
2. Lossy Predictive Coding : Delta Modulation
3. Transform Coding :
   - JPEG using DCT
   - DWT based Image Coding

Removes irrelevancies
Quantization
Greater Compression

Data Redundancy

Data Redundancy is defined as,

$$R_D = 1 - \frac{1}{C_R} \quad where, \quad C_R = \frac{n_1}{n_2}$$

where n1 and n2 denote the number of information-carrying units in the two data sets that represent the same information.

Data Redundancy & Compression

DATA = Information + Redundancy

- For data reduction or compression we try to reduce three kinds of redundancies present in data

Three approaches :

- Reduce **Coding Redundancy** - some pixel values more common than others.
- Reduce **Interpixel Redundancy** - neighboring pixels have similar values.
- Reduce **Psychovisual Redundancy** - some color differences are imperceptible.

1. Interpixel Redundancy

- Interpixel redundancy is due to the correlation between the neighboring pixels in an image. That means neighboring pixels are not statistically independent.
- The gray levels are not equally probable. The value of any given pixel can be predicated from the value of its neighbors that is they are highly correlated.
- The information carried by individual pixel is relatively small.
- To reduce the interpixel redundancy the difference between adjacent pixels can be used to represent an image.

2. Psychovisual Redundancy

- Psychovisual redundancies exist because human perception of the information in an image normally does not involve quantitative analysis of every pixel value in the image.
- In general, an observer searches for distinguishing features such as edges or textural regions and mentally combines them into recognizable groupings.
- The brain then correlates these groupings with prior knowledge in order to complete the image interpretation process.

Psychovisual Redundancy . . . .

- Psychovisual redundancy is associated with real or quantifiable visual information. Its elimination is possible only because the information itself is not essential for normal visual processing
- Since the elimination of psychovisually redundant data results in a loss of quantitative information, it is commonly referred to as quantization

3. Coding Redundancy

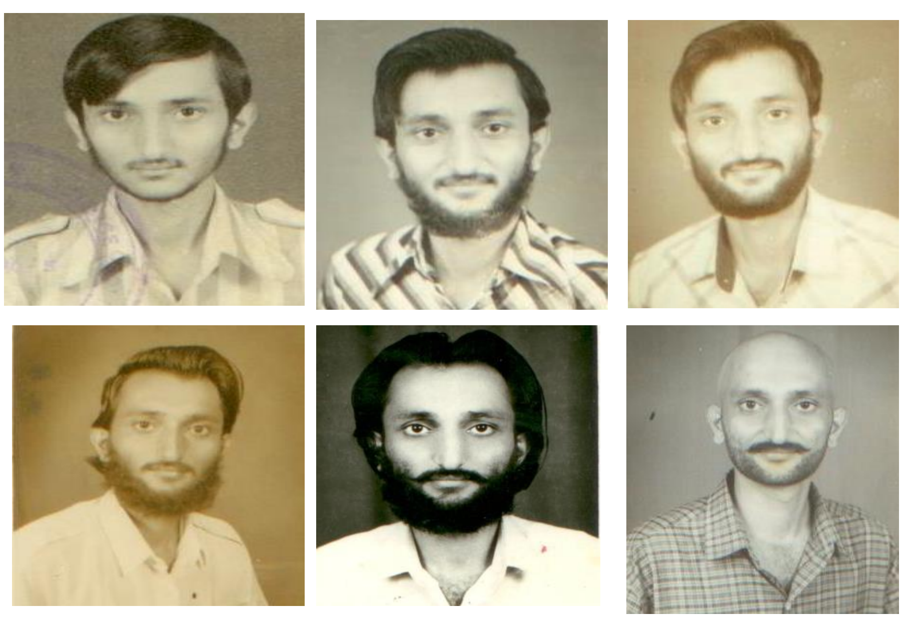
- Coding redundancy is associated with the representation of information.
- The information is represented in the form of codes.
- If the gray levels of an image are coded in a way that uses more code symbols than absolutely necessary to represent each gray level then the resulting image is said to contain coding redundancy

4. Interpixel Temporal Redundancy

- Interpixel temporal redundancy is the statistical correlation between pixels from successive frames in video sequence
- Temporal redundancy is also called interframe redundancy. Temporal redundancy can be exploited using motion compensated predictive coding
- Removing a large amount of redundancy leads to efficient video compression.

❖ Redundancy

Inter pixel Redundancy-- neighboring pixels have similar values.








Inter pixel Redundancy-- neighboring pixels have similar values

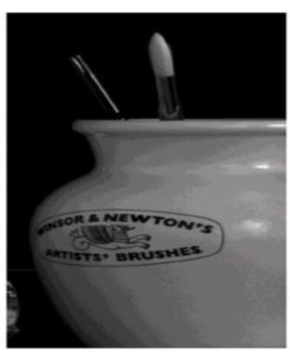





DPCM Coding Technique exploits Interpixel Redundancy

Psycho Visual Redundancy

Input Image
[256] Levels



Input Image
Uniformly Quantized
to [16] Levels

Input Image
IGS Quantized
to [16] Levels

- IGS Coding Technique exploits

Psycho visual Redundancy

Coding Redundancy

- It is possible to reduce the number bits required for storage using variable length coding technique that depends on statistical parameters of pixels.

Huffman Coding and RLE Techniques exploits Coding Redundancy

Fidelity Criteria

Objective Criteria :

- Bits Per Pixel [BPP]
- Entropy, Average Length, Efficiency
- Mean Square Error [MSE]
- Signal to Noise Ratio [SNR]
- Peak Signal to Noise Ratio [PSNR]

Subjective Criteria :

: Images are viewed by human beings

Fidelity Criteria…Objective Criteria :

1. Mean Square Error [MSE]

$$MSE = \frac{1}{MxN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - \hat{f}(x,y)]^2$$

2. Signal to Noise Ratio [SNR]

$$SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y)]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - \hat{f}(x,y)]^2}$$

3. Peak Signal to Noise Ratio [PSNR]

$$PSNR = \frac{(Peak\ Signal\ Value)^2}{MSE}$$

Fidelity Criteria…Subjective Criteria :

- Subjective Criteria : Images are viewed by human beings.

Evaluations can be done with the following scale.

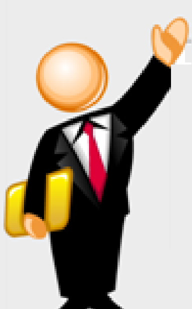
1 : Excellent, 2 : Fine,
3 : Passable, 4 : Marginal,
5: Inferior, 6 : Unusable

Fidelity Criteria…Subjective Criteria :

Value	Rating	Description
1	Excellent	An image of Extremely High Quality, as Good as you could desire.
2	Fine	An Image of High Quality, providing enjoyable viewing. Inferior is not objectionable.
3	Passable	An image of acceptable quality.
4	Marginal	An image with poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor quality image, but you could watch it.
6	Unusable	Image is so bad that you can not watch it.

Lossless Compression

1. Arithmetic Coding
2. Huffman Coding
3. Run-Length Coding
4. DPCM Coding



[redacted]

[1] Arithmetic Coding

Arithmetic coding was developed by Jorma Rissanen and Glen Langdon at IBM/Arc in 1977-78

Jorma Rissanen

Ref :
U.S. Patent 4,122,440 – (IBM) Filed 4 March 1977, granted 24 October 1978 – Glen George Langdon, Jorma Johannes Rissanen – Method and means for arithmetic string coding

Glen Langdon

[1] Arithmetic Coding

- Example : Message " I N D I A"

Size of Message = 5 Characters
= 5 Bytes
= 40 bits

Symbol	Probability
I	0.4
N	0.2
D	0.2
A	0.2

Step-1 : To find interval for first character “ I”

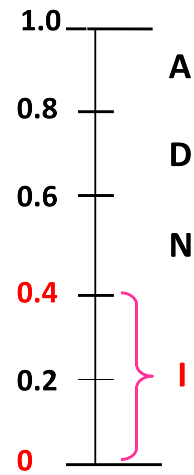
Lower range value = 0
Upper range value = 1

No of sub-intervals = 5

$$\text{Step Value} = \frac{\text{Upper - Lower}}{\text{No of Sub intervals}}$$

$$\text{Step Value} = \frac{1 - 0}{5} \qquad \text{Step Value} = 0.2$$

Interval for Character I is [0.0 to 0.4]



Step-2 : To find interval for 2nd character “ N”

Range : [0.0 to 0.4]

Lower value = 0
Upper value = 0.4

No of sub-intervals = 5

Step Value = 0.08

Interval for Character N is [0.16 to 0.24]

Step-3 : To find interval for 3rd character " D"

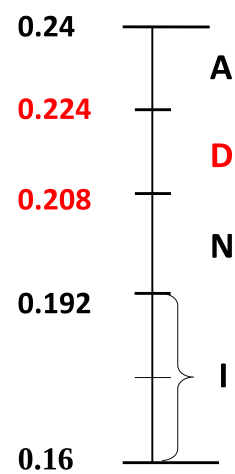
Range : [0.16 to 0.24]

Lower value = 0.16
Upper value = 0.24

$$\text{Step Value} = \frac{0.24 - 0.16}{5}$$

Step Value = 0.08

Interval for Character D is [0.208 to 0.224]



Step-4 : To find interval for 4th character " I"

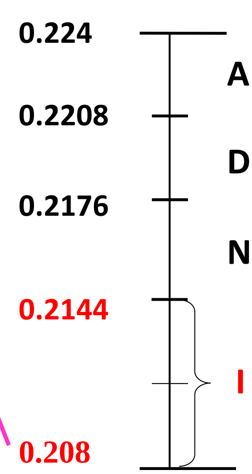
Range : [0.208 to 0.224]

Lower value = 0.208

Upper value = 0.224

Step Value = 0.032

Interval for Character I is [0.208 to 0.2144]

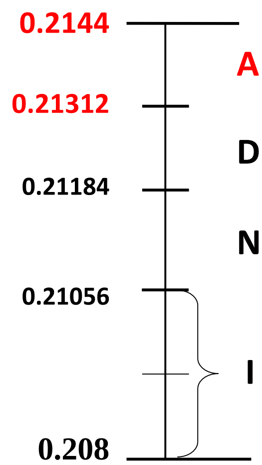


Step-5 : To find interval for 5th character “ A”

Range : [0.208 to 0.2144]

No of sub-intervals = 5

Step Value = 0.00128



Interval for Character A is [0.21312 to 0.2144]

0.21312 < Arithmetic Codeword < 0.2144

Arithmetic Codeword For The Message I N D I A is

0.21313

(b) Find BPP and Compression Ratio

- (1) To find **BPP** of Compressed Message **:**

$$BPP = \frac{\text{Size of Compressed Message in Bits}}{\text{Total No of Characters}}$$

BPP = 6.2

- (2) To find **Compression Ratio :**

$$Compression\ Ratio = \left(\frac{\begin{matrix}\text{Original} & - & \text{Compressed} \\ \text{Size} & & \text{Size}\end{matrix}}{\text{Original Size}} \right) \text{X } 100\ \%$$

Compression Ratio = 20 %

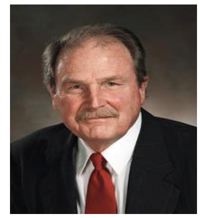
Home Work

- Encode the following Message using Arithmetic Coding :

HELLO, JAPAN

[2] Huffman Coding

Huffman encoding was invented by David Huffman in 1952

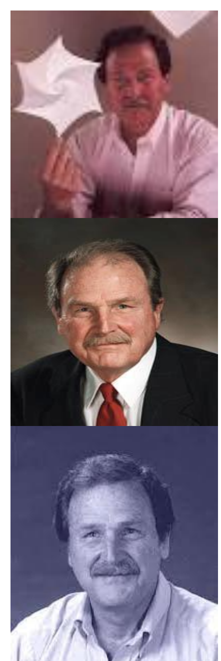


August 9, 1925
To
October 7, 1999

- **Ref :**

Huffman, D. (1952). "A Method for the Construction of Minimum-Redundancy Codes" (PDF). Proceedings of the IRE. 40 (9): doi:10.1109/JRPROC.1952.273898.

The Story of Huffman Coding



The story of the invention of Huffman codes is a great story that demonstrates that students can also do better than other professionals

David Huffman (1925-1999) was a research student in an Electrical Engineering course in 1951 in MIT USA. His professor, Robert Fano, assigned his class a project to develop an optimal compression scheme (without mentioning this was a difficult and unsolved problem).

David's solution, published in his paper A Method for the Construction of Minimum-Redundancy Codes is one of the most-cited papers in all of information theory.

[2] Huffman Coding

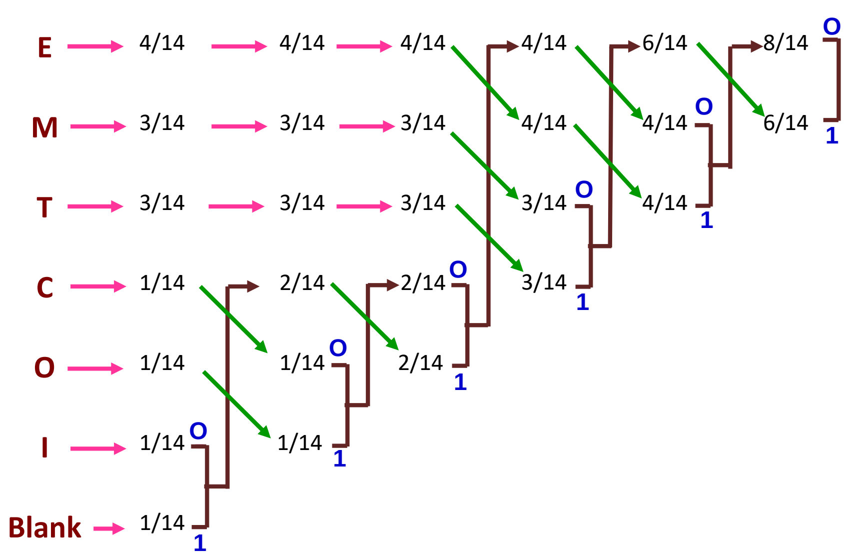
- Example-1 :

  Consider a Message :

  COMMITTEE MEET

Symbol	Probability
C	$^{1}/_{14}$
O	$^{1}/_{14}$
M	$^{3}/_{14}$
I	$^{1}/_{14}$
T	$^{3}/_{14}$
E	$^{4}/_{14}$
Blank	$^{1}/_{14}$

- **Find Huffman Tree :**



- **Find Symbol Codewords :**

Symbol	Probability P_K	Code-word	Length of Codeword L_k
E	4/14	01	2
M	3/14	10	2
T	3/14	11	2
C	1/14	0000	4
O	1/14	0001	4
I	1/14	0010	4
Blank	1/14	0011	4

*** Find Code Word of the Message:**

Message and Codeword :

C O M M I T T E E ~~B~~ M E E T

0000 0001 10 10 1110 11 11 01 01 0011 10 01 01 11

Encoded Message is then given by :
000000011010111011110101001110010111

Size of Encoded Message : 36 bits

(b) Find BPP and Compression Ratio

- (1) To find **BPP** of Compressed Message :

$$BPP = \frac{\text{Size of Compressed Message in Bits}}{\text{Total No of Characters}}$$

BPP = 36/14 == 2.57

- (2) To find **Compression Ratio :**

$$\textit{Compression Ratio} = \left(\frac{\begin{array}{ll}\text{Original} & - \text{Compressed} \\ \text{Size} & \quad\text{Size}\end{array}}{\text{Original Size}} \right) \text{X } 100\ \%$$

Compression Ratio = 67.85 %

Example -2 :

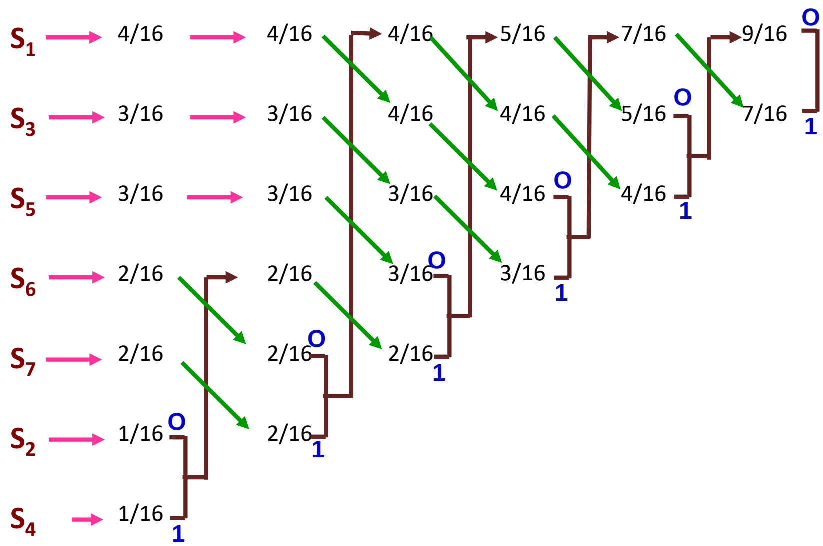
F =

10	10	65	20
10	75	20	30
20	30	44	50
50	44	30	10

Symbol	Probability
$S_1 = 10$	4/16
$S_3 = 20$	3/16
$S_5 = 30$	3/16
$S_6 = 44$	2/16
$S_7 = 50$	2/16
$S_2 = 65$	1/16
$S_4 = 75$	1/16

Size of Input Image : 128 bits

- **Find Huffman Tree :**



- **Symbol Codewords :**

Symbol S_K	Probability P_K	Code-word C_K	Length of Codeword L_k
S_1 = 10	4/16	1 0	2
S_3 = 20	3/16	1 1	2
S_5 = 30	3/16	0 0 0	3
S_6 = 44	2/16	0 1 0	3
S_7 = 50	2/16	0 1 1	3
S_2 = 65	1/16	0 0 1 0	4
S_4 = 75	1/16	0 0 1 1	4

- **Symbol Codewords Table :**

Symbol	Code-word
$S_1 = 10$	1 0
$S_3 = 20$	1 1
$S_5 = 30$	0 0 0
$S_6 = 44$	0 1 0
$S_7 = 50$	0 1 1
$S_2 = 65$	0 0 1 0
$S_4 = 75$	0 0 1 1

F =

10	10	65	20
10	75	20	30
20	30	44	50
50	44	30	10

F =

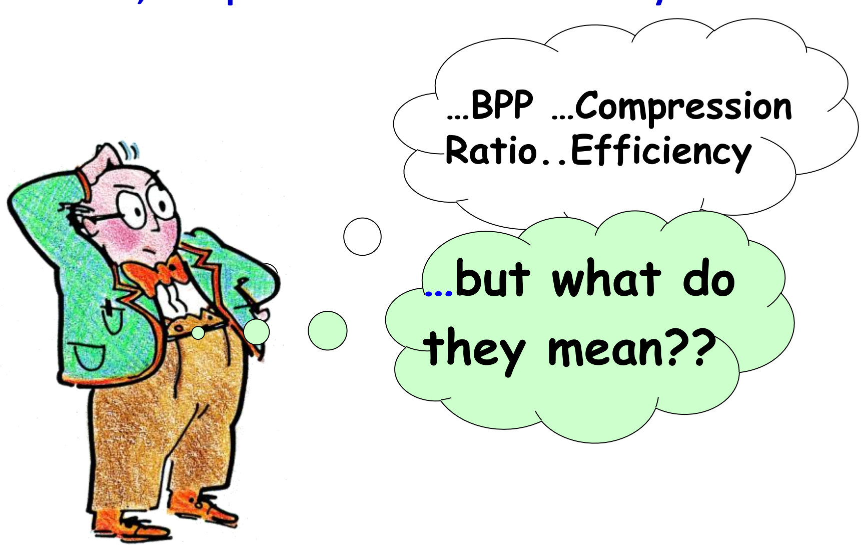
10	10	0010	11
10	0011	11	000
11	000	010	011
011	010	000	10

Encoded Image is then given by :

1010001011100011110001100001001101101000010

Size of Encoded Image : 43 bits

(b) **Find BPP, Compression Ratio and Efficiency.**



(b) **Find Bits Per Pixel, Compression Ratio and Efficiency.**

- (1) To find **BPP** of Compressed Image **:**

$$\text{BPP} = \frac{\text{Size of Compressed Image in Bits}}{\text{Total No of Pixels}}$$

BPP = 43/16

BPP = 2.687

- **(2) To find Compression Ratio :**

$$\text{Compression Ratio} = \left(\frac{\text{Original Size} - \text{Compressed Size}}{\text{Original Size}} \right) \text{X } 100\ \%$$

Compression Ratio = 66.41 %

(3) To find Entropy:

$$\text{Entropy} = \sum_{k=1}^{N} \text{P}_{\text{k}} \log_2 \left(\frac{1}{P_k} \right)$$

Entropy = 2.6537

(4) To find Average Length :

$$\text{Average Length} = \sum_{k=1}^{N} \mathrm{P_k}\ L_k$$

Average Length = 2.6875

(5) To find Efficiency :

$$\text{Efficiency} = \left(\frac{\text{Entropy}}{\text{Average Length}}\right) \text{X } 100\ \%$$

$$= \left(\frac{2.6537}{2.6875}\right) \mathbf{X\ 100\ \%}$$

$$= 98.75$$

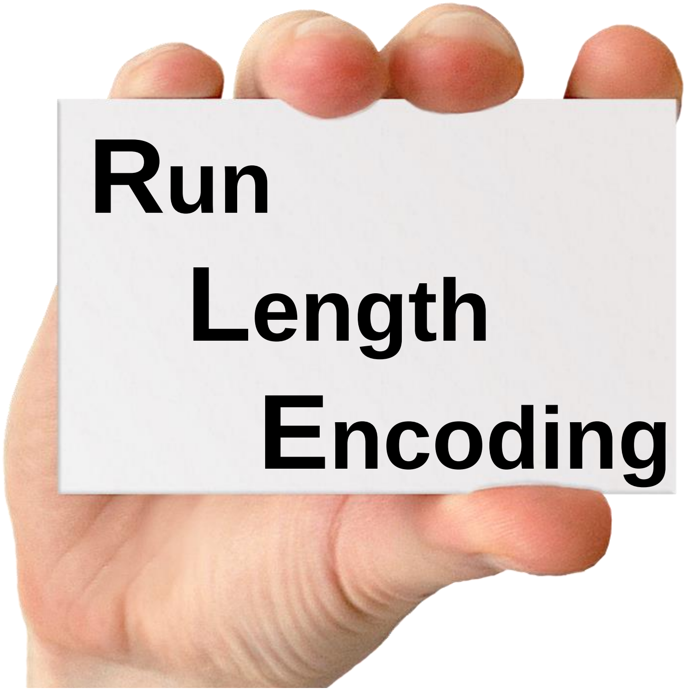
Home Work

- Encode the following Message using Huffman Coding :

M I C R O P R O C E S S O R

C O M M U N I C A T I O N

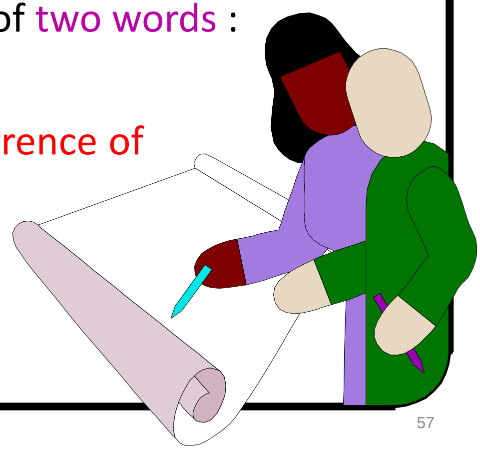
COMMITTEE MEET TOMORROW


Run
Length
Encoding


[3] Run Length Coding
Run Length Coding was patented by Hitachi in 1983
Hitachi, Ltd. is a Japanese multinational electronics company headquartered in Chiyoda, Tokyo.

[3] Run-Length Encoding

- RLE consists of repetitive sequence of Tokens
- Size of Token is Two Bytes.
- Each Token consists of two words :
  1. Pixel Value
  2. Consecutive occurrence of Pixel value.



57

Example -1 :

$$F = \begin{array}{|ccccc|} \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 8 & 8 & 0 \\ 0 & 8 & 8 & 8 & 2 \\ 2 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

RLE coded image is :

0, 6, 8, 3, 0, 2, 8, 3, 2, 5, 0, 6

Size of RLE Coded Image: 12 bytes

Size of input Image is : 25 bytes

Image is compressed

Compression Ratio = 52 %

Example -2 :

F =

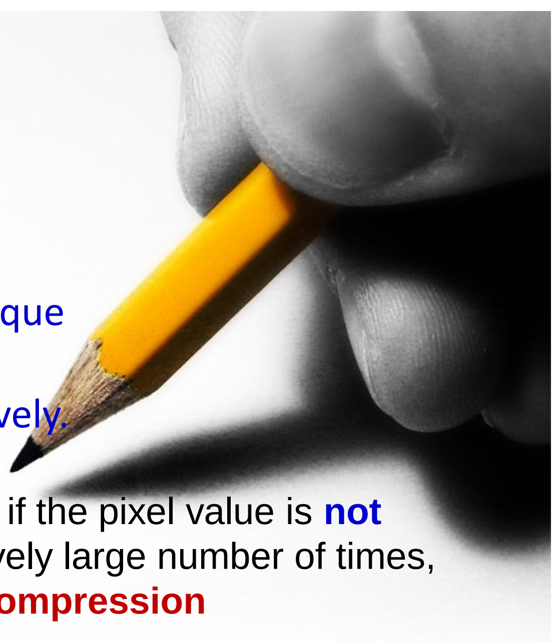
0	0	0	0	0
1	8	8	8	2
2	8	8	8	0
0	1	2	2	1
5	0	2	1	4

RLE coded image is =
0, 5, 1, 1, 8, 3, 2, 2, **8, 3**
0, 2, 1, 1, 2, 2, 1, 1, 5, 1
3, 1, 2, 1, 1, 1, 4, 1

RLE coded image Size : 30 Bytes
Size of input Image is : 25 Bytes

Image is NOT Compressed

Here, size of the RLE coded image is larger than size of input image. That means every time RLE does not give compression.

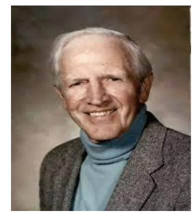


- **RLE does not give compression every time.**
- RLE is effective compression technique when pixel value is repeated consecutively.
- In the overall image if the pixel value is **not** repeated consecutively large number of times, **RLE cannot give compression**

[4] Differential PCOM (DPCM)

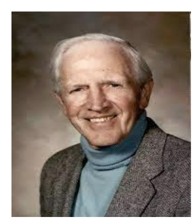
DPCM was invented by
C. Chapin Cutler at
Bell Labs in 1950

Reference:
U.S. patent 2605361, C. Chapin Cutler, "Differential Quantization of Communication Signals", filed June 29, 1950, issued July 29, 1952



Dec 16, 1914
To
Dec 1, 2002

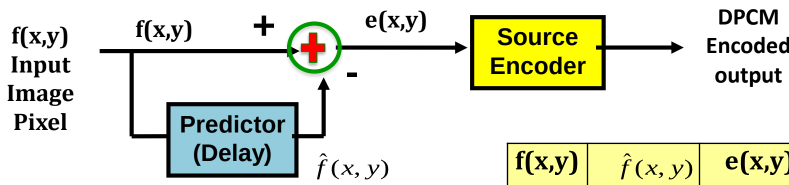
- **Cassius Chapin Cutler** (December 16, 1914 – December 1, 2002) was an American electrical engineer at Bell Labs.
- His notable achievements include the invention of the corrugated waveguide and differential pulse-code modulation (DPCM).



- He was born on December 16, 1914 in Springfield, Massachusetts to Paul A. Cutler and Myra Chapin. He received the B.S. degree in electrical engineering from Worcester Polytechnic Institute in 1937. On September 27, 1941 he married Virginia Tyler in Waterford, Maine.
- In 1979 Cutler left Bell Labs to become a professor of applied physics at Stanford University.

[4] Lossless Predictive Coding : Differencial PCM

(a) DPCM Encoder



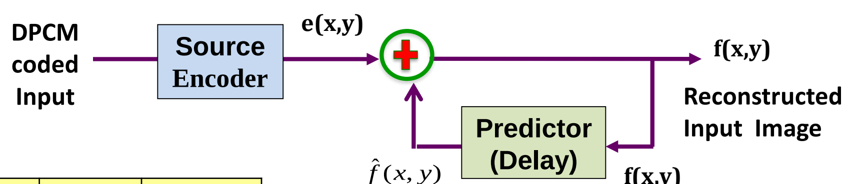
f(x,y)	$\hat{f}(x,y)$	e(x,y)
102	0	102
100	102	-2
104	100	4
103	104	-1
99	103	-4
98	99	-1
104	98	6
100	104	-4
102	100	2

Input Image :

$$A = \begin{bmatrix} 102 & 100 & 104 \\ 103 & 99 & 98 \\ 104 & 100 & 102 \end{bmatrix}$$

Lossless Predictive Coding : Differencial PCM

(b) DPCM Decoder



e(x,y)	$\hat{f}(x,y)$	f(x,y)
102	0	102
-2	102	100
4	100	104
-1	104	103
-4	103	99
-1	99	98
6	98	104
-4	104	100
2	100	102

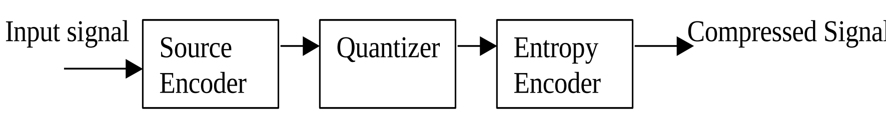
Decoded Image :

$$B = \begin{bmatrix} 102 & 100 & 104 \\ 103 & 99 & 98 \\ 104 & 100 & 102 \end{bmatrix}$$

Lossy Compression

1. Improved Gray Scale Quantization
2. Lossy Predictive Coding : Delta Modulation
3. Transform Coding :
   - JPEG using DCT
   - DWT based Image Coding

Lossy Image Compression System



Source Encoder

An Encoder is for Image Compression. A variety of linear transforms are available like :

- Discrete Fourier Transform (DFT),
- Discrete Cosine Transform (DCT),
- Discrete Wavelet Transform (DWT).

Quantizer :

- It is used to approximate the input coefficient value so that it can be compressed efficiently.
- Quantization: a process of representing a large – possible infinite – set of values with a much smaller set.
- Scalar quantization: a mapping of an input value x into a finite number of output values, y: $Q: x \rightarrow y$

Improved Gray Scale [IGS]Quantization

Example :

- Consider a Digital Image

A =

12	26	41
13	10	25
57	54	06

(a) Find 3 bit IGS coded Image.

Pixel	Pixel in 6 bit Binary	Sum 000 000	IGS Code
12	0 0 1 1 0 0	0 0 1 1 0 0	0 0 1
26	0 1 1 0 1 0	0 1 1 1 1 0	0 1 1
41	1 0 1 0 0 1	1 0 1 1 1 1	1 0 1
13	0 0 1 1 0 1	0 1 0 1 0 0	0 1 0
10	0 0 1 0 1 0	0 0 1 1 1 0	0 0 1
25	0 1 1 0 0 1	0 1 1 1 1 1	0 1 1
57	1 1 1 0 0 1	1 1 1 0 0 1	1 1 1
54	1 1 0 1 1 0	1 1 0 1 1 1	1 1 0
06	0 0 0 1 1 0	1 1 1 1 0 1	1 1 1

(b) Find Decoded Image.

IGS Coded pixel value	Decoded Pixel in 6 bit Binary	Output Pixel Value
0 0 1	0 0 1 0 0 0	8
0 1 1	0 1 1 0 0 0	24
1 0 1	1 0 1 0 0 0	40
0 1 0	0 1 0 0 0 0	16
0 0 1	0 0 1 0 0 0	8
0 1 1	0 1 1 0 0 0	24
1 1 1	1 1 1 0 0 0	56
1 1 0	1 1 0 0 0 0	48
1 1 1	1 1 1 0 0 0	56

Input Image.

$$A = \begin{bmatrix} 12 & 26 & 41 \\ 13 & 10 & 25 \\ 57 & 54 & 06 \end{bmatrix}$$

Encoded Image.

001 011 101
010 001 011
111 110 111

Decoded Image.

$$B = \begin{bmatrix} 8 & 24 & 40 \\ 16 & 8 & 24 \\ 56 & 48 & 56 \end{bmatrix}$$

(b) Find BPP, Compression Ratio, MSE, SNR and PSNR

- (1) To find **BPP** of Compressed Image **:**

$$BPP = \frac{\text{Size of Compressed Image in Bits}}{\text{Total No of Pixels}}$$

BPP = 27/9

BPP = 3

- (2) To find **Compression Ratio :**

$$\text{Compression Ratio} = \left(\frac{\begin{matrix}\text{Original} & - & \text{Compressed} \\ \text{Size} & & \text{Size}\end{matrix}}{\text{Original Size}} \right) \text{X } 100 \ \%$$

Compression Ratio = 50 %

(3) To find Mean Square Error [MSE]

$$MSE = \frac{1}{MxN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\,f(x,y) - \hat{f}(x,y)]^2$$

MSE =

(4) **To find Signal to Noise Ratio** [SNR]

$$SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\,f(x,y)]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\,f(x,y) - \hat{f}(x,y)]^2}$$

SNR =

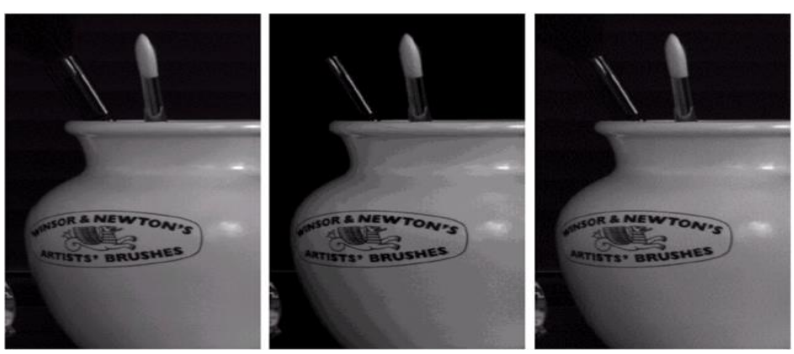
(5). **To find Peak Signal to Noise Ratio** [PSNR]

$$PSNR = \frac{(Peak\ Signal\ Value)^2}{MSE}$$

PSNR =

Psycho visual Redundancy

- Brightness of a region is perceived by the ye that does not depend on the absolute grey level values. This is because the human eye does not respond with equal sensitivity to all visual information. Some information is visually more important than others. Information which is not visually important is called Psychovisual Redundancy.
- Psychovisual redundancies exist in all images and are eliminated without hampering the subjective quality of image.
- Simply reducing the quantization i.e. number of bits of representation, compress the image but also produces false contouring.
- I.G.S. coding reduces the quantization but also reduces false contouring.

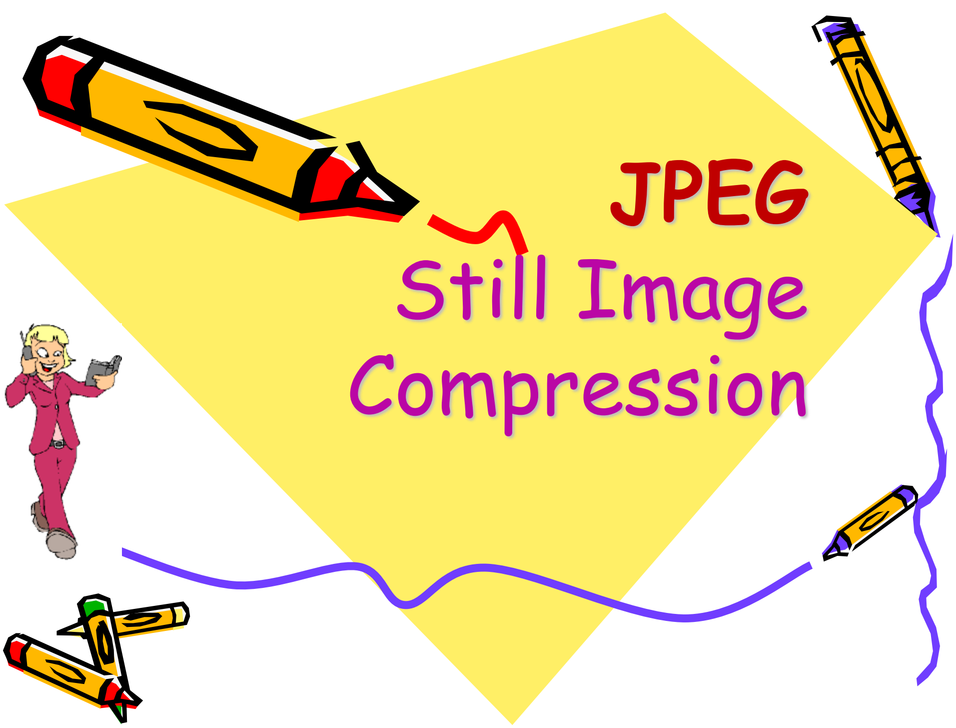


Input Image [256] Levels	Input Image Uniformly Quantized to [16] Levels	Input Image IGS Quantized to [16] Levels

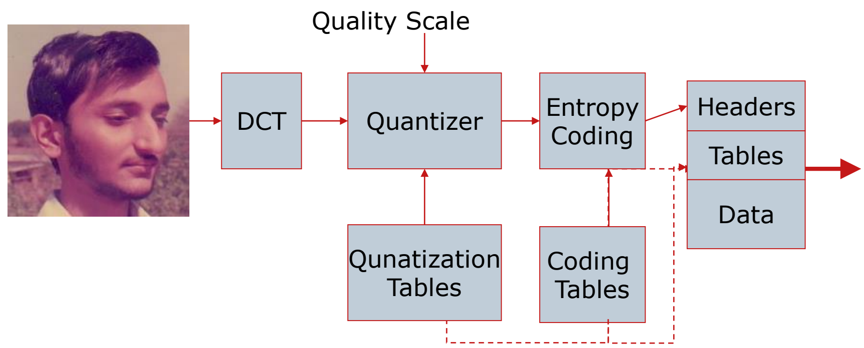
Lossy Compression

1. Improved Gray Scale Quantization
2. ~~Lossy Predictive Coding : Delta Modulation~~
3. Transform Coding :
   – JPEG using DCT
   – DWT based Image Coding

Kiran TALELE 99870 30 881 talelesir@gmail.com



JPEG Encoder



- An Input image is first divided into pixel blocks of size 8 x 8 which are processed left to right, top to bottom.
- Each block of 8 x 8 pixel size is level shifted by subtracting the quantity 2^{n-1} where 2^n is the highest gray value (ie for n = 8 , 2^n = 256).

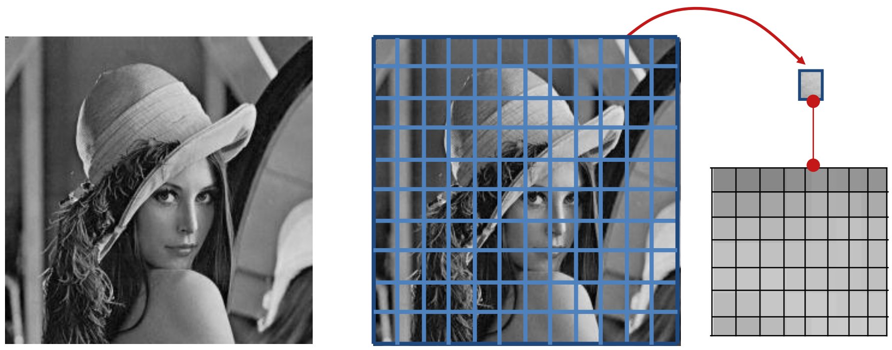
- The 2D DCT of the block is then computed and quantized.

$$C'(u,v)\left[\frac{C(u,v)}{Q(u,v)}\right]$$

Quantized DCT coefficients are reordered using the zigzag pattern to form a 1D sequence of quantized coefficients.

- Non-zero AC coefficients are coded using a variable length code that defines the coefficients value and the number of preceding zeros.
-
- DC coefficients are the difference coded relative to the DC coefficient of the previous sub-image.

JPEG Coding Example Contd



- The **picture** is divided in 8x8 blocks

JPEG Coding Example …

Input Image : (8 x 8) Size Block

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

JPEG Coding Example ...

After 128 Deduction :

-76	- 73	- 67	- 62	- 58	- 67	- 64	- 55
-65	- 69	- 62	- 38	- 19	- 43	- 59	- 56
-66	- 69	- 60	- 15	16	- 24	- 62	- 55
-65	- 70	- 57	- 6	26	- 22	- 58	- 59
-61	- 67	- 60	- 24	- 2	- 40	- 60	- 58
-49	- 63	- 68	- 58	- 51	- 65	- 70	- 53
-43	- 57	- 64	- 69	- 73	- 67	- 63	- 45
-41	- 49	- 59	- 60	- 63	- 52	- 50	- 34

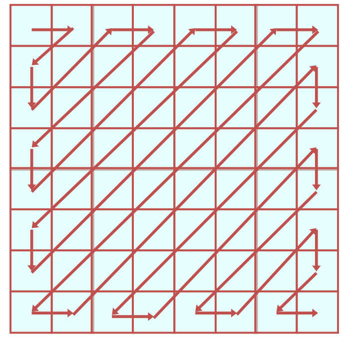
After DCT :

-415	- 29	- 62	25	55	- 20	-1	3
7	- 21	- 62	9	11	-7	- 6	6
- 46	8	77	-25	-30	10	7	-5
- 50	13	35	-15	- 9	6	0	3
11	- 8	-13	- 2	- 1	1	- 4	1
- 10	1	3	- 3	- 1	0	2	-1
-4	- 1	2	- 1	2	- 3	1	- 2
- 1	- 1	- 1	- 2	- 1	- 1	0	- 1

Zigzag Scaning of Quantized DCT Coeffs.

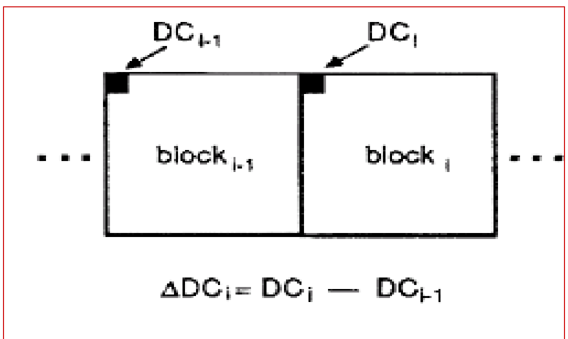
After quantization:

-26	-3	–6	2	2	0	0	0
1	-2	–4	0	0	0	0	0
–3	1	5	–1	–1	0	0	0
–4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



- After zigzag scan:
  [- 26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2
  0 0 -1 2 0 0 0 0 0 -1 -1 EOB]
- Represent Each number with its Value, Run and Size

DPCM of DC Coefficient



Differential DC encoding

JPEG Coding Example Contd.

After zigzag scan:

[- 26 - 3 1 - 3 - 2 - 6 2 - 4 1 - 4 1 1 5
0 2 0 0 - 1 2 0 0 0 0 0 -1 -1 EOB]

After DPCM of DC coefficient (assuming previous DC value of -17):

[- 9 - 3 1 - 3 - 2 - 6 2 - 4 1 - 4 1 1 5
0 2 0 0 - 1 2 0 0 0 0 0 -1 -1 EOB]

After RLC:

[-9 (0,-3) (0,1) (0,-3) (0,-2) (0,-6) (0,2) (0,-4) (0,1) (0,-4) (0,1)
(0,1) (0,5) (1,2) (2,-1) (0,2) (5,-1) (0,-1) EOB]

After VLC:

1010110 0100 001 0100 0101 100001 0110 100011 001
100011 001 001 100101 11100110 110110 0110 11110100 0001010

Total Bits : 92 **Total Pixels : 64** **Bits/Pixel : 1.43** **Compression Ratio : 5.6**

JPEG Coding Example Contd.

After inverse quantization:

-416	- 33	- 60	32	48	0	0	0
12	- 24	- 56	0	0	0	0	0
- 42	13	80	- 24	- 40	0	0	0
- 56	17	44	- 29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

JPEG Coding Example Contd.

After IDCT:

- 70	- 64	- 61	- 64	- 69	- 66	- 58	- 50
- 72	- 73	- 61	- 39	- 30	- 40	- 54	- 59
- 68	- 78	- 58	- 9	13	- 12	- 48	- 64
- 59	- 77	- 57	0	22	- 13	- 51	- 60
- 54	- 75	- 64	- 23	- 13	- 44	- 63	- 56
- 52	- 71	- 72	- 54	- 54	- 71	- 71	- 54
45	- 59	- 70	- 68	- 67	- 67	- 61	- 50
- 35	- 47	- 61	- 66	- 60	- 48	- 44	- 44

JPEG Coding Example Contd.

After level shift by +128:

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
4	53	64	105	115	84	65	72
6	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

JPEG Coding Example Contd.

Error:

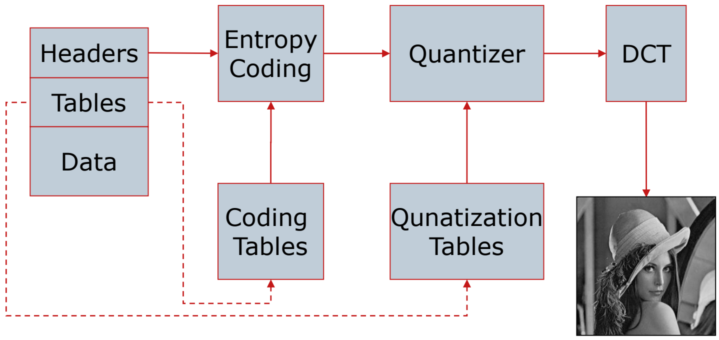
- 6	- 9	- 6	2	11	- 1	- 6	- 5
7	4	- 1	1	11	- 3	- 5	3
2	9	- 2	- 6	- 3	- 12	- 14	9
- 6	7	0	- 4	- 5	- 9	- 7	1
- 7	8	4	- 1	11	4	3	- 2
3	8	4	- 4	2	11	1	1
2	2	5	- 1	- 6	0	- 2	5
- 6	- 2	2	6	- 4	- 4	- 6	10

Kiran TALELE 99870 30 881 talelesir@gmail.com

JPEG Performance

Bits/pixel	Quality	Compression Ratio
>= 2	Indistinguishable	8-to-1
1.5	Excellent	10.7-to-1
0.75	Very Good	21.4-to-1
0.5	Good	32-to-1
0.25	Fair	64-to-1

JPEG Decoder



Dr. Kiran TALELE

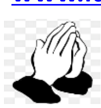
- Academic : PhD
- Professional :
  - Dean-Students, Alumni & External Relations @ Bharatiya Vidya Bhavans' Sardar Patel Institute of Technology (SP-IT), Mumbai
  - Head-Academic Relation @ Sardar Patel Technology Business Incubator (SP-TBI), Mumbai
  - Treasurer-IEEE Bombay Section

091-9987030881

kiran.talele@spit.ac.in / ktvtalele@gmail.com

https://www.linkedin.com/in/k-t-v-talele/

www.facebook.com/Kiran-Talele-1711929555720263



Stay Connected

SARDAR PATEL TECHNOLOGY BUSINESS INCUBATOR

- **Dr. Kiran TALELE** is an Associate Professor in Electronics & Telecommunication Engineering Department of Bharatiya Vidya Bhavans' Sardar Patel Institute of Technology, Mumbai with 33+ years experience in Academics.

- He is a Dean of Students, Alumni and External Relations at Sardar Patel Institute of Technology, Andheri Mumbai.
  He is also a Head of Sardar Patel Technology Business Incubator, Mumbai.

- His area of research is Digital Signal & Image Processing, Computer Vision, Machine Learning and Multimedia System Design.

- **He has published 85+ research papers at various national & international refereed conferences and journals. He has filed published 12+ patents at Indian Patent Office. One patent is granted in 2021.**

- He is a Treasurer of IEEE Bombay Section and Mentor for Startup Incubation & Intellectual Asset Creation.

- He received incentives for excellent performance in academics and research from Management of S.P.I.T. in 2008-09. He is a recipient of P.R. Bapat IEEE Bombay Section Outstanding Volunteer Award 2019.