UNIT-5
Data Compression

⊛ Storage Space:

The storage space requirement for the uncompressed data is far huge as compared to the compressed data. Transmission of uncompressed data over digital network is highly complicated in the sense that the uncompressed data requires higher bandwidth for a single point-to-point communication. To eliminate this complication multimedia system uses compressed formats of digital audio and video stream. Compressed formats are cost-effective as well as feasible. The most important compression techniques in use today are JPEG for single pictures, H.263 for video, MPEG for video and audio etc.

⊛ Coding Requirements:

→ Images require higher storage compared to text. Audios and videos have even more demanding properties for data storage along with data rates for communication.

→ Compression/Coding in multimedia is subjected to certain limitations/constraints. While decoding/decompressing the compressed data, it should be as good as possible.

i) Dialogue mode application: Interaction among humans/users via multimedia information.

Requirement for compression and decompression:

↳ End-to-end delay lower than 150ms.
↳ End-to-end delay of 50ms for face-to-face applications.

ii) Retrival mode application: A user retrives information from a multimedia database.

Requirements:

↳ Fast forward and backward data retrival with simultaneous display.
↳ Fast search for information in multimedia databases.
↳ Random access to single images and audio frames with an access time less than 0.5 second.

⊛ Source, Entropy and Hybrid Coding:

no need to remember this table

Entropy Coding	Run-length coding	
	Huffman coding	
	Arithmetic coding	
Source Coding	Prediction	DPCM
		DM
	Transformation	FFT
		DCT
	Layered coding	Bit position
		Subsampling
		Sub-band coding
	Vector Quantization	
Hybrid Coding	JPEG	
	MPEG	
	H.261	
	DVI RTV, DVI PLV	

1) Entropy Coding:

- → Lossless coding, decompression process regenerates the data completely.
- → Used regardless of the media's specific characteristics.
- → Data stream is considered to be a simple digital sequence.
- → Run-length coding, Huffman coding, Arithmetic coding are entropy coding.

2) Source Coding:

- → Lossy coding
- → Semantics of data are considered.
- → Degree of compression depends on data contents.
- → DPCM, DM, DCT etc. are examples of source coding.

3) Hybrid Coding:

- → Used by most multimedia systems.
- → Combination of entropy and source coding.
- → JPEG, MPEG, H.261, DVI etc, are it's examples.

⊛. Huffman Coding:

Huffman coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters. The most frequent character gets the smallest code and the least frequent character gets the largest code.

The variable-length codes assigned to input characters are prefix codes, means the codes (bit sequences) are assigned in such a way that the code assigned to one character is not the prefix of code assigned to any other character. This is how Huffman Coding makes sure that there is no ambiguity when decoding the generated bitstream. There are mainly two major parts in Huffman coding:

1). Build a Huffman Tree from input characters.

2). Traverse the Huffman Tree and assign codes to characters.

Steps to build Huffman Tree:

Input is an array of unique characters along with their frequency of ocurrences and output is Huffman Tree.

Step1: Create a leaf node for each unique character and build a min heap of all leaf nodes. (Min Heap is used as a priority queue. The value of frequency field is used to compare two nodes in min heap. Initially, the least frequent character is at root).

Step2: Extract two nodes with the minimum frequency from the min heap.

Step3: Create a new internal node with a frequency equal to the sum of the two nodes frequencies. Make the first extracted node as its left child and the other extracted node as its right child. Add this node to min heap.

Step4: Repeat steps 2 and 3 until the heap contains only one node. The remaining node is the root node and the tree is complete.

Example:

Character	Frequency
a	05
b	09
c	12
d	13
e	16
f	45

Here, no. of character = 6.
i.e, code must be of length 3.

$\left[\begin{array}{l} 2^2 < 6 \text{ (not suitable to use).} \\ 2^3 < 6 \text{ (suitable to use).} \end{array}\right]$

Step1:

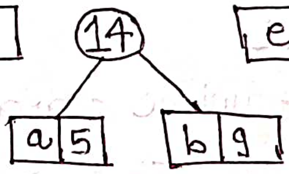
| a | 05 | | b | 09 | | c | 12 | | d | 13 | | e | 16 | | f | 45 |

Step2: Select two leaf containing the least values. (minimum values).

Selecting | a | 5 | | b | 9 |

Step3:

| c | 12 | | d | 13 |



| e | 16 | | f | 45 |

sum selected ones freq. make smaller value as left child and greater as right.

i.e, repeat step 2

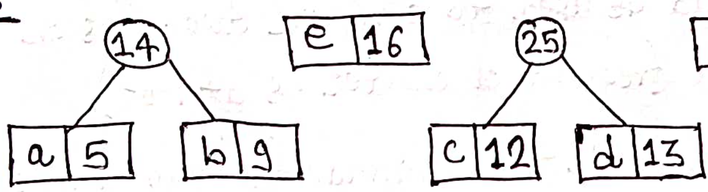
Step4: Select two leaf with minimum output:

Selecting | c | 12 | | d | 13 |

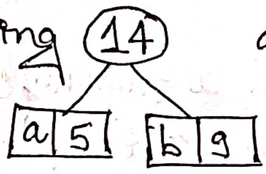
now steps 2 and 3 repeated until we get single node

i.e, repeat step3

Step5:

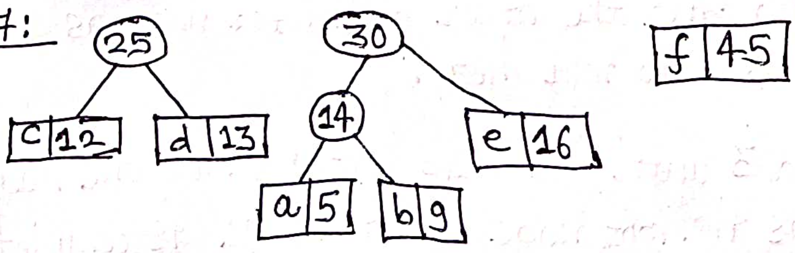


Step6: Selecting



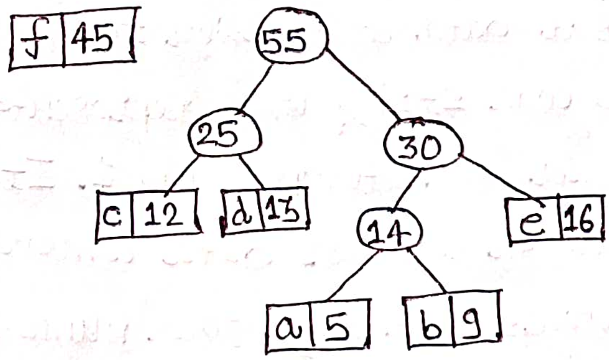
and | e | 16 |

Step7:



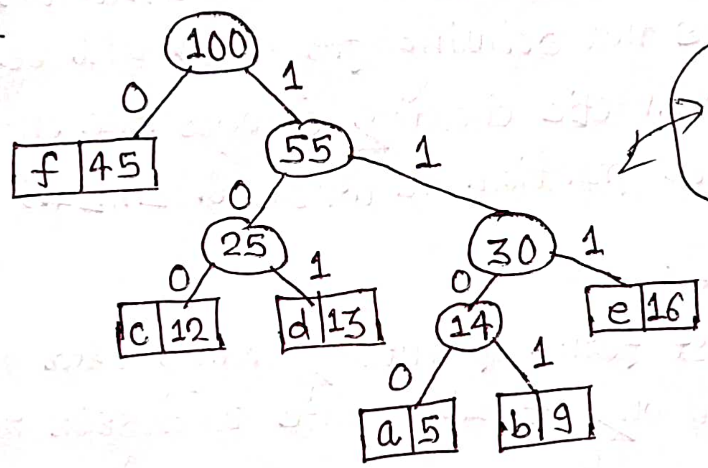
Step 8: Selecting nodes 25 and 30.

Step 9:



Step 10: Selecting nodes f | 45 and 55

Step 11:



Since only one node so, left denoted by 0 right by 1 (Codes are assigned).

given in question

The codes are as follows:

Character	code-word	frequency
f	0	45
c	100	12
d	101	13
a	1100	5
b	1101	9
e	111	16

Bits used before coding: $5\times3+9\times3+12\times3+13\times3+16\times3+45\times3$
$= 300$

Bits used after coding: $5\times4+9\times4+12\times3+13\times3+16\times3+45\times0$
$= 179$

$\therefore \frac{300-179}{300}\times 100 = 40.3\%$ of space is saved or compressed after using Huffman coding.

Arithmetic encoding

⊛. Arithmetic coding:

Arithmetic coding is a data compression technique that encodes data by creating a code string which represents a fractional value on the number line between 0 and 1. It is a form of entropy coding used in lossless data compression. A string of characters is represented using fixed number of bits per character. When a string is converted to arithmetic encoding, frequently used characters will be stored with fewer bits and characters not occuring frequently will be stored with more bits. Arithmetic encoding encodes the entire message into single number, a fraction n where $(0.0 \le n \le 1.0)$.

⊛. Run Length Encoding:

It is the simplest form of entropy coding used in lossless compression of data allowing 4-25g byte compression into 3 bytes. In this encoding, frequently (at least 4 times) repeating byte are replaced depending on their occurance into three bytes, where first byte represent the repeating byte, second is the exclamation and third byte represents the no. of of occurance.

Syntax: x!x

- x → Repeating character.
- ! → Exclamation (flag)
- x → no. of repeatition

Example: ABCCCCCDDDDF = 12

RLE: ABC!5D!4F = 9 ← Compressed.

Procedure of RLE:

→ A byte should occur at least 4 consecutive times.

→ Compressed data contains following format.

x!x

- x → repeating character (byte).
- ! → exclamation (flag)
- x → no. of repeatition.

Example : ABBBBBBBBBBBCCDDDDDDDDE

↓ RLE

A01 B11 C02 D08 E01

↓ OR

AB!11C!2 D!8 E

Drawback of RLE:

When no. repeating value then RLE generates more output value (bigger size) than input value.
i.e, Compression of a data will increase the size compared to original data.

Example: AABCCD
RLE: A02B01C02D01

DCT stands for Discrete Cosine Transformation.

⊛. Lossy Sequential DCT-based mode:-

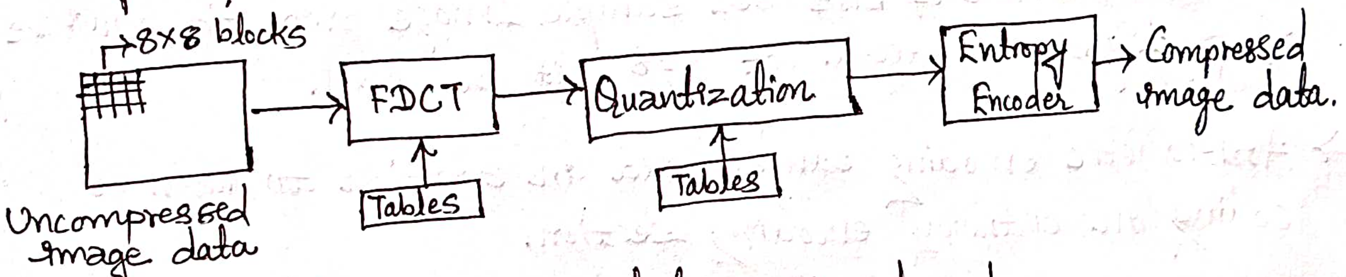


Fig: Lossy sequential DCT-based mode.

$$T(u,v) = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} \cdot \cos \frac{(2x+1)u\pi}{16} \cdot \cos \frac{(2y+1)}{v\pi}$$

where, $C_u, C_v = 1/\sqrt{2}$ for $u,v=0$

otherwise $C_u, C_v = 1$.

steps of DCT

i) Image Processing:

→ After image prepration, the uncompressed image data are grouped into units of 8x8 pixels.
→ Each sample is encoded using $p=8$ bit.
→ Each pixel is an integer between 0 to 255.

Image Processing is carried as follows:-

↳ DCT based transformation coding is carried out.
↳ A forward DCT (FDCT) is applied.
↳ For later reconstruction, the decorder uses IDCT.

ii> Quantization:

→ The entire lossy process occurs in quantization.
→ Specific frequencies are given more importance than others.
→ Tables are used for quantization and dequantization.
→ Image quality may decrease due to quantization.

iii> Entropy Encoding:

→ DC coefficients are encoded by subtracting the DC coefficients of previous unit.
→ Huffman coding is preferred as it is free.
→ Coding tables for each DC and AC coefficients must be provided.
→ AC coefficients are processed using zig-zag sequence.

(B). Expanded lossy DCT-based mode:

→ Image processing in this mode differs from DCT mode in terms of the number of bits per sample. Image prepration must be of p=12 bit instead of p=8 bit per pixel.

→ Arithmetic encoding can be used in addition to Huffman coding in entropy encoding section.

Note: Only above two points are different rest all is same as DCT-based mode.

⊛ JPEG Compression Process:

JPEG stands for Joint Photographic Experts Group. We perform JPEG compression to reduce the size of the file without damaging it's quality. Reducing the size of image will also improve the efficiency of system as it will give less load on it.

Process:

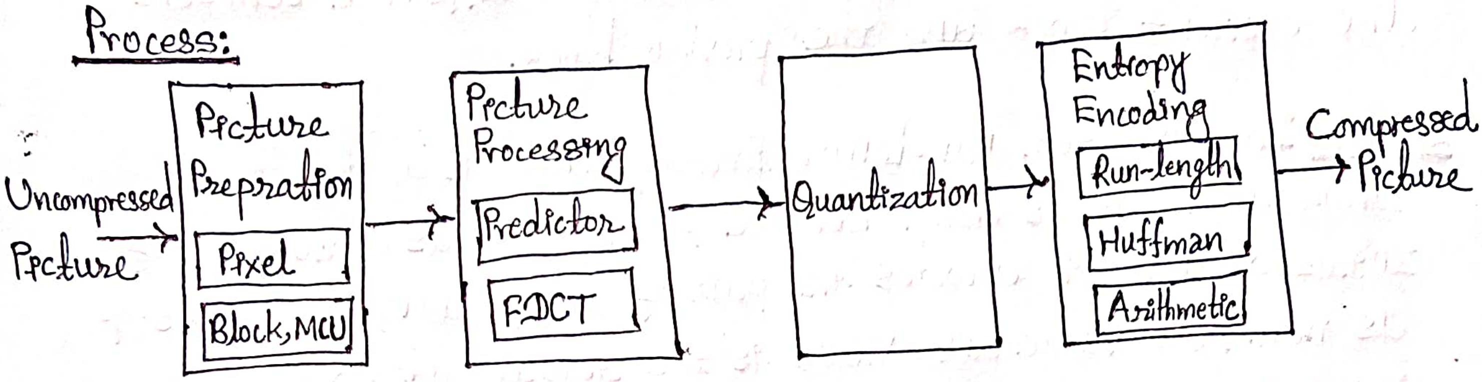


Fig: Steps of the JPEG compression process.

Step1: The input image is divided into a small block which is having 8x8 dimensions. The dimension is sum up to 64 units. Each unit of image is called pixel.

Step2: JPEG uses [Y, Cb, Cr] model instead of using [R, G, B] model. So in the 2nd step, RGB is converted into YCbCr.

Step3: After the conversion of colors, it is forwarded to DCT. DCT uses a cosine function and does not use complex numbers. It converts informations which are in a block of pixels from the spatial domain to the frequency domain.

Step4: Humans are unable to see important aspects of the image because they are having high frequencies. The matrix after DCT conversion can only preserve values at the lowest frequency to certain point. Quantization is used to reduce the number of bits per sample.

Step5: The zigzag scan is used to map the 8x8 matrix to a 1x64 vector. Zigzag scanning is used to group low-frequency coefficients to the top level of the vector and the high coefficient to the bottom. To remove the large number of zero in the quantized matrix, the zigzag matrix is used.

Step6: Next step is vectoring, DPCM is applied to DC component. DC components are large and vary but they are usually close to the previous value. DPCM encodes the difference between the current block and the previous block.

Step7: In this step, Run-Length Encoding (RLE) is applied to AC components. This is done because AC components have a lot of zeros in it. It encodes in pair of (skip, value) in which skip is non zero value and value is the actual coded value of the non zero components.

Step8: In this step, DC components are coded into Huffman.

MPEG Compression Process:

MPEG stands for Moving Picture Experts Group. It develops audio video file formats which are known as MPEG. It has released various versions of file namely MPEG-1, MPEG-2, MPEG-3 etc. MPEG Compresses the data to small bits. The compression rate of MPEG is very high as it stores only the changes that occur from one frame to another frame.

Process: → only written using points but almost similar to JPEG compression.

1) **Reduction of resolution:** MPEG uses YUV model instead of RGB so, first step is to convert RGB into YUV model.

2) **Motion Estimation:** Motion estimation calculates motion vector for finding the matching blocks, between the future frame corresponding to the present frame. It detects temporal redundancy.

3) **Motion compensation and image substitution:** Motion vector produced during motion estimation is used here to reduce file size and bandwidth.

4) **DCT:** DCT uses a cosine function and does not use complex numbers. It converts informations which are in a block of pixels from the spatial domain to the frequency domain. DCT decreases frequencies so that humans are able to see important aspects.

5) **Quantization:** Quantization is the process of reducing the number of bits needed to store an integer value by reducing the precision of integer.

6) **Run Length Encoding:** RLE is applied to AC components because AC components have a lot of zeros in it. It encodes in pair of (skip, value) in which skip is non zero value and value is the actual coded value of the non zero components.

7) **Encoding Entropy:** Encoding entropy is mainly done by Huffman coding. It reduces file size in huge amount and finally we get compressed image.