Unit 5: Data Compression

Why compression?

- To reduce the volume of data to be transmitted (text, fax, images)
- To reduce the bandwidth required for transmission and to reduce storage requirements (speech, audio, video)

Data compression implies sending or storing a smaller number of bits. Although many methods are used for this purpose, in general these methods can be divided into two broad categories: **lossless** and **lossy** methods.

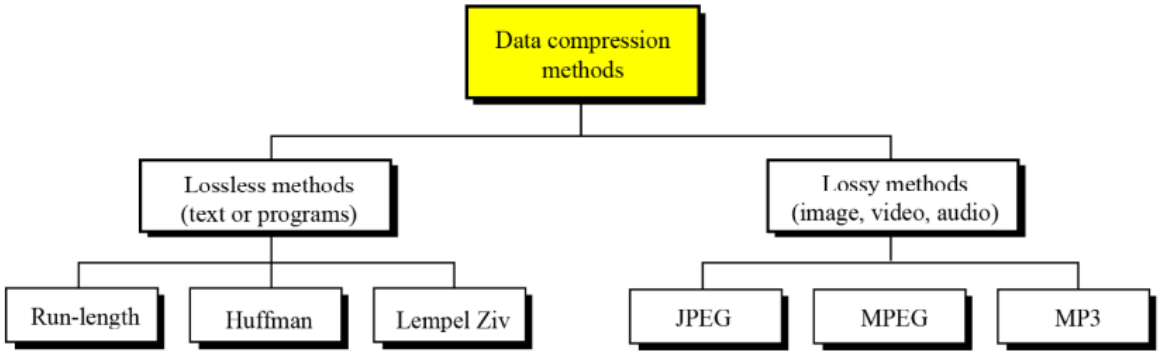


Figure 1: Data Compression Methods

1.1 Storage Space

Uncompressed graphics, audio and video data require considerable storage capacity which in the case of uncompressed video is often not even feasible given today's technology. Data transfer of uncompressed video data over digital networks requires very high bandwidth to be provided for a single point-to-point communication. To provide feasible and cost-effective solutions, most multimedia systems handle compressed digital video and audio data streams.

1.2 Coding Requirements

Images have considerably higher storage requirements than text; audio and video have even more demanding properties for data storage. Not only is a huge amount of storage required, but the data rates for the communication of continuous media are also significant.

How is compression possible?

- Redundancy in digital audio, image, and video data.
- Properties of human perception

Redundancy: Adjacent audio samples are similar (predictive encoding); samples corresponding to silence (silence removal).

In digital image, neighboring samples on a scanning line are normally similar (spatial redundancy)

In digital video, in addition to spatial redundancy, neighboring images in a video sequence may be similar (temporal redundancy)

Human perception: Compressed version of digital audio, image, video need not represent the original information exactly.

Perception sensitivities are different for different signal patterns.

Human eye is less sensitive to the higher spatial frequency components than the lower frequencies (transform coding).

1.3 DIFFERENCE BETWEEN LOSSLESS AND LOSSY DATA COMPRESSION:

S.No	Lossless data compression	Lossy data compression
1.	In Lossless data compression, there is no loss of any data and quality.	In Lossy data compression, there is a loss of quality and data, which is not measurable.
2.	In lossless, the file is restored in its original form.	In Lossy, the file does not restore in its original form.
3.	Lossless data compression algorithms are Run Length Encoding, Huffman encoding, Shannon fano encoding, Arithmetic encoding, Lempel Ziv Welch encoding.	Lossy data compression algorithms are: Transform coding, Discrete Cosine Transform, Discrete Wavelet Transform, fractal compression, etc.
4.	Lossless compression is mainly used to compress text-sound and images.	Lossy compression is mainly used to compress audio, video, and images.
5.	In lossy data compression, lossless data compression holds more data.	As compare to lossless data compression, lossy data compression holds less data.
6.	File quality is high in this compression.	File quality is low in the lossy data compression.
7.	Lossless data compression mainly supports RAW, BMP, PNG, WAV.	Lossy data compression mainly supports JPEG, GIF, MP3, MP4, MKV.

1.4 ENTROPY, SOURCE AND HYBRID CODING

<table>
<tr><td rowspan="3">Entropy Encoding</td><td colspan="2">Run-length Coding</td></tr>
<tr><td colspan="2">Huffman Coding</td></tr>
<tr><td colspan="2">Arithmetic Coding</td></tr>
<tr><td rowspan="7">Source Coding</td><td rowspan="2">Prediction</td><td>DPCM</td></tr>
<tr><td>DM</td></tr>
<tr><td rowspan="2">Transformation</td><td>FFT</td></tr>
<tr><td>DCT</td></tr>
<tr><td rowspan="3">Layered Coding</td><td>Bit Position</td></tr>
<tr><td>Subsampling</td></tr>
<tr><td>Sub-band Coding</td></tr>
<tr><td colspan="2">Vector Quantization</td></tr>
<tr><td rowspan="4">Hybrid Coding</td><td colspan="2">JPEG</td></tr>
<tr><td colspan="2">MPEG</td></tr>
<tr><td colspan="2">H.261</td></tr>
<tr><td colspan="2">DVI RTV, DVI PLV</td></tr>
</table>

1.4.1 Entropy coding

An entropy coding is any lossless data compression method that attempts to approach the lower bound declared by Shannon's source coding theorem, which states that any lossless data compression method must have expected code length greater or equal to the entropy of the source.

1.4.1.1 Run length encoding

This method replaces the consecutive occurrences of a given symbol with only one copy of the symbol along with a count of how many times that symbol occurs. Hence the names ‘run length’.

For example, the string AAABBCDDDD would be encoded as 3A2B1C4D.

A real-life example where run-length encoding is quite effective is the fax machine. Most faxes are white sheets with the occasional black text. So, a run-length encoding scheme can take each line and transmit a code for while then the number of pixels, then the code for black and the number of pixels and so on. This method of compression must be used carefully. If there is not a

lot of repetition in the data then it is possible the run length encoding scheme would actually increase the size of a file.

1.4.1.2 Arithmetic coding

Arithmetic coding is a type of entropy encoding utilized in lossless data compression. Ordinarily, a string of characters, for example, the words "hey" is represented for utilizing a fixed number of bits per character.

Arithmetic coding yields a single codeword for each encoded string of characters. The first step is to divide the numeric range from 0 to 1 into a number of different characters present in the message to be sent – including the termination character – and the size of each segment by the probability of the related character.

1.4.2 Source coding

Source coding takes into account the semantics and the characteristics of the data. Thus, the degree of compression that can be achieved depends on the data contents. Source coding is a lossy coding process in which there is some loss of information content. For e.g. in case of speech the speech is transformed from the time domain to frequency domain. In the psychoacoustic the encoder analyzes the incoming audio signals to identify perceptually important information by incorporating several psychoacoustic principles of the human ear. One is the critical-band spectral analysis, which accounts for the ear's poorer discrimination in higher frequency regions than in lower-frequency regions. The encoder performs the psychoacoustic analysis based on either a side-chain FFT analysis or the output of the filter bank.

E.g. Differential Pulse Code Modulation, Delta Modulation, Fast Fourier Transform, Discrete Fourier Transform, Sub-band coding etc

1.4.3 Hybrid Coding

This type of coding mechanism involves the combine use of both the source coding and the entropy coding for enhancing the compression ratio still preserving the quality of information content. The example of Hybrid Coding includes MPEG, JPEG, H.261, DVI techniques.

1.4.4 Major Steps of data compression:

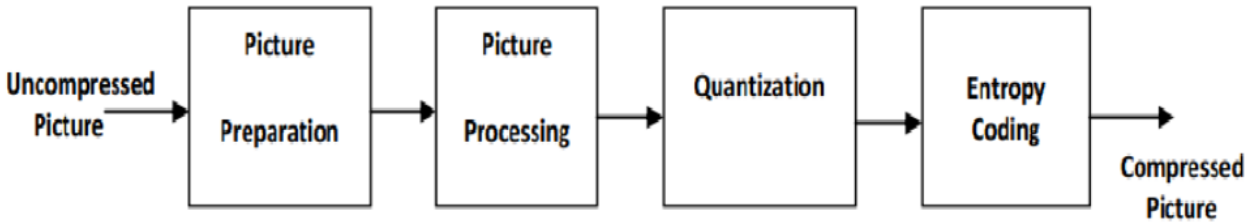


Figure: Major steps of data compression

Preparation: Preparation includes analog to digital conversion and generating an appropriate digital representation of the information. An image is divided into blocks of 4*4 or 8*8 pixels, and represented by a fixed number of bits per pixel.

Processing: This involves the conversion of the information from the time domain to the frequency domain by using DCT (discrete cosine transform). In the case of motion video compression, interframe coding uses a motion vector for each 8*8 block.

Quantization: It defines discrete level or values that the information is allowed to take. This process involves the reduction of precision. The quantization process may be uniform or it may be differential depending upon the characteristics of the picture.

Entropy Encoding: This is the lossless compression method where the semantics of data is ignored but only its characteristics are considered. It may be run length coding or entropy coding. After compression, the compressed video stream contains the specification of the image starting point and an identification of the compression technique may be the part of the data stream. The error correction code may also be added to the stream. Decompression is the inverse process of compression.

1.5 HUFFMAN ENCODING

Huffman coding is a lossless data compression algorithm. It is a **lossless data compression** mechanism. It is also known as **data compression encoding.** It is widely used in image (JPEG or JPG) compression. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters.

Huffman encoding idea are mention in following steps:

- It assigns a variable-length code to all the given characters.
- The code length of a character depends on how frequently it occurs in the given text or string.
- A character gets the smallest code if it frequently occurs.
- A character gets the largest code if it least occurs.

Prefix rule: 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.

The major steps involved in Huffman coding:

- First, construct a **Huffman tree** from the given input string or characters or text.
- Assign, a Huffman code to each character by traversing over the tree.

1.5.1 Huffman Tree

Step 1: For each character of the node, create a leaf node. The leaf node of a character contains the frequency of that character.

Step 2: Set all the nodes in sorted order according to their frequency.

Step 3: There may exist a condition in which two nodes may have the same frequency. In such a case, do the following:

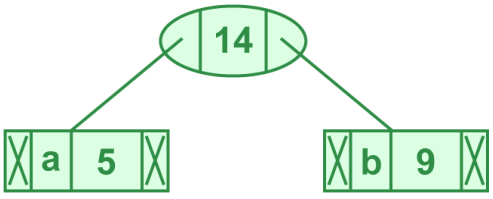
1. Create a new internal node.
2. The frequency of the node will be the sum of the frequency of those two nodes that have the same frequency.
3. Mark the first node as the left child and another node as the right child of the newly created internal node.

Step 4: Repeat step 2 and 3 until all the node forms a single tree. Thus, we get a Huffman tree

For example:

Character	Frequency
A	5
B	9
C	12
D	13
E	16
F	45

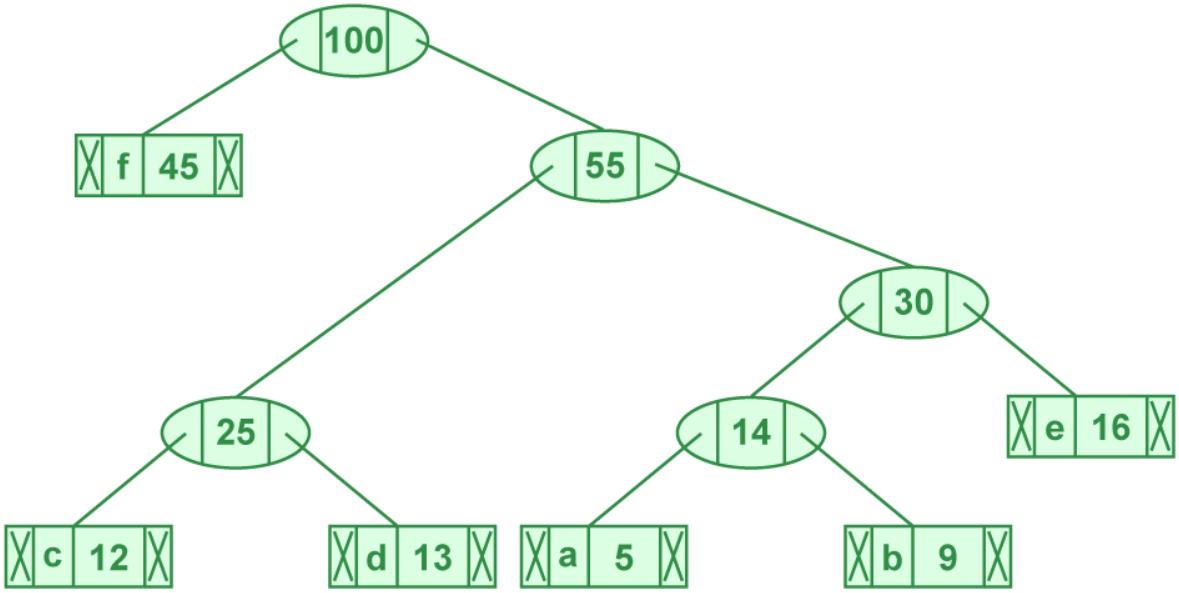
1. **Step 1:** Make pairs of characters and their frequencies. i.e. (a,5), (b,9), (c,12), (d,13), (e,16), (f,45)
2. **Step 2**: Sort pairs with respect to frequency, in this case given data are already sorted.
3. **Step 3:** Pick the first two characters and join them under a parent node. Add a new internal node with frequency 5 + 9 = 14.



Character	Frequency
C	12
D	13
New node	14
E	16
F	45

4. **Step 4:** Repeat Steps 2 and 3 until, we get a single tree.

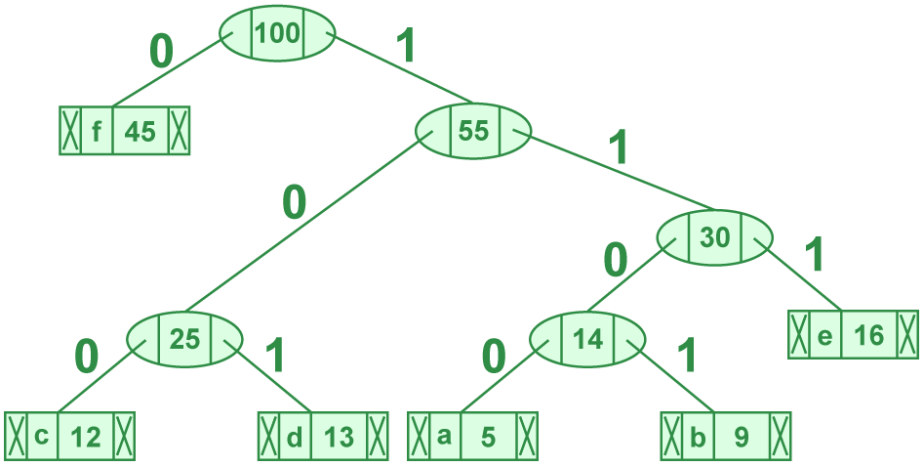
The Huffman tree will be



Character	Frequency					
A	5	12	14	25	45	
B	9	13	16			100
C	12	14	25	30	55	
D	13					
E	16	16	45	45		
F	45	45				

1.5.2 Coding

Therefore, we get a single tree. Then, we will find the code for each character with the help of the above tree. Assign a weight to each edge. Note that each **left edge-weighted is 0** and the **right edge-weighted is 1.**



Character	Frequency	code-word
A	5	1100
B	9	1101
C	12	100
D	13	101
E	16	111
F	45	0

Here, no of character = 6

No of bits required to represent character = 2^3 i.e. code length will be 3.

Bits used before coding: 5*3 +9*3 + 12*3 + 13*3 +16*3 + 45*3 = 300

Bits used after coding: 5*4 + 9*4 + 12*3 + 13*3 + 16*3 + 45*0 = 179

Therefore, (300-179)/300 = 40.3% of space is saved after Huffman coding.

Q. Find the ***Average Code Length for the String*** and ***Length of the Encoded String*** of the given string "**abcdrcbd**" Use the Huffman coding technique to find the representation bits of the character.

Character	Frequency
A	5
B	2
C	1
D	1
R	2

Hints:

Average Code Length = ∑ (frequency × code length) / ∑ (frequency)

length= Total number of characters in the text x Average code length per character

2 JPEG

The JPEG standard for compressing continuous –tone still pictures (e.g. photographs) was developed by joint photographic experts group, JPEG became an ISO standard. JPEG is one algorithm, to satisfy the requirements of a broad range of still-image compression applications.

Why we need image compression?

Let us take an example, A typical digital image has 512x480 pixels. In 24-bit colour (one byte for each of the red, green and blue components), the image requires 737,280 bytes of storage space. It would take about 1.5 minutes to transmit the uncompressed image over a 64kb/second link. The JPEG algorithms offer compression rates of most images at ratios of about 24:1. Effectively, every 24 bits of data is stuffed into 1 bit, giving a compressed file size (for the above image dimensions) of 30,720 bytes, and a corresponding transmission time of 3.8 seconds.

There are some requirements of JPEG standard and they are:

- The JPEG implementation should be independent of image size.
- The JPEG implementation should be applicable to any image and pixel aspect ratio.
- Color representation itself should be independent of the special implementation.
- Image content may be of any complexity, with any statistical characteristics.
- The JPEG standard specification should be state of art (or near) regarding the compression factor and achieved image quality.
- Processing complexity must permit a software solution to run on as many available standard processors as possible. Additionally, the use of specialization hardware should substantially enhance image quality.

2.1 STEPS OF JPEG COMPRESSION PROCESS

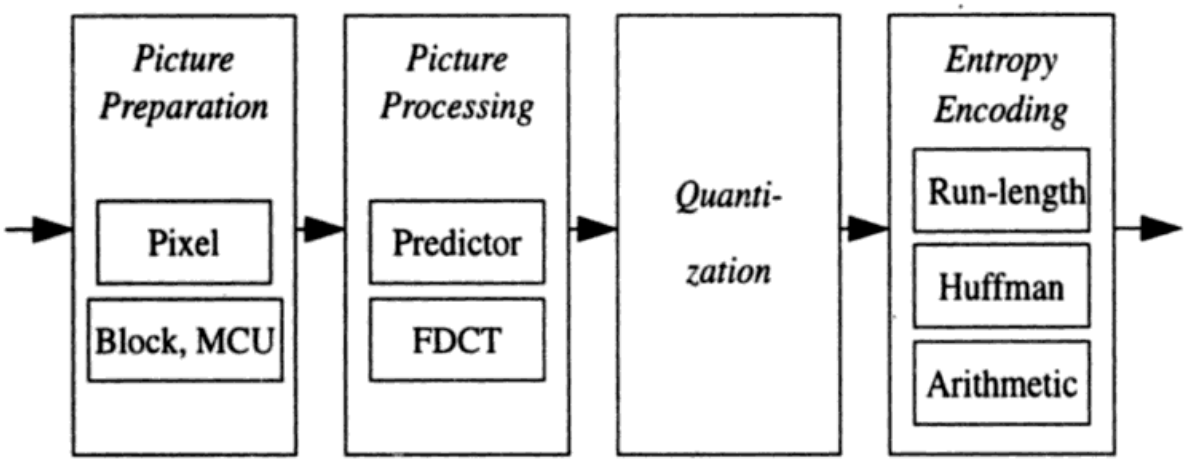


Figure 2: Steps of JPEG compression process

Step 1: (Block/Image Preparation)

This step involves the block preparation. For e.g. let us assume the input to be 640*480 RGB image with 24 bits/pixel. The luminance and chrominance component of the image is calculated using the YIQ model for NTSC system.

Y=0.30R + 0.59G + 0.11B

I= 0.60R – 0.28G – 0.32B

Q= 0.21R – 0.52G + 0.31B

Separate matrices are constructed for Y, I and Q each elements in the range of 0 and 255. The square blocks of four pixels are averaged in the I and Q matrices to reduce them to 320*240. Thus the data is compressed by a factor of two. Now, 128 is subtracted from each element of all three matrices to put 0 in the middle of the range. Each image is divided up into 8*8 blocks. The Y matrix has 4800 blocks; the other two have 1200 blocks.

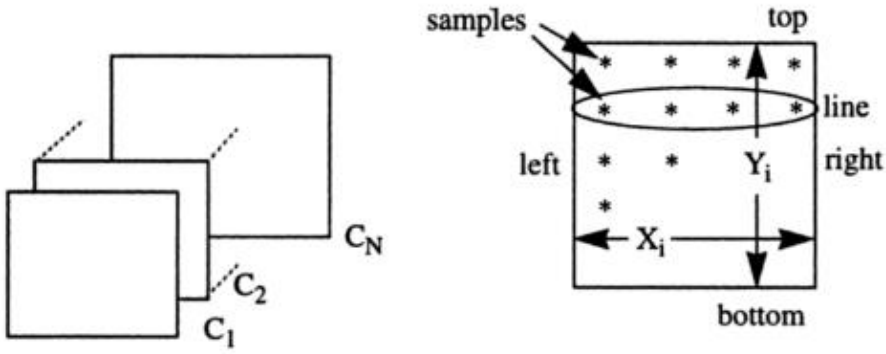


Figure 6.4: *Digital uncompressed still image with the definition of the respective image components according to the JPEG standard.*

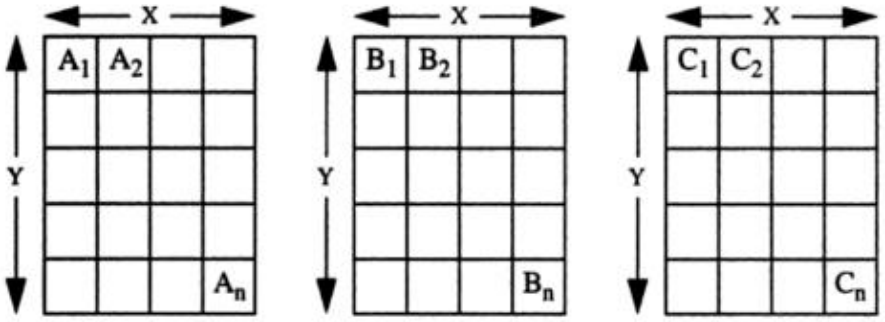


Figure 6.5: *Example of JPEG image preparation with three components having the same resolution.*

Step 2: (Discrete Cosine Transformation)

Discrete Cosine Transformation is applied to each 7200 blocks separately. The output of each DCT is an 8*8 matrix of DCT coefficients. DCT element (0,0)is the average value of the block. The other element tells how much spectral power is present at each spatial frequency.

Step 3: (Quantization) In this step the less important DCT coefficients are wiped out. This transformation is done by dividing each of the coefficients in the 8*8 DCT matrix by a weight taken from a table. If all the weights are 1 the transformation does nothing however, if the weights increase sharply from the origin, higher spatial frequencies are dropped quickly.

Step 4: (Differential Quantization) This step reduces the (0,0) value of each block by replacing it with the amount it differs from the corresponding element in the previous block. Since these elements are the averages of their respective blocks, they should change slowly, so taking the differential values should reduce most of them to small values. The (0,0) values are referred to as the DC components; the other values are the AC components.

Step 5: (Run length Encoding) This step linearizes the 64 elements and applies run-length encoding to the list. In order to concentrate zeros together, a zigzag scanning pattern is used. Finally run length coding is used to compress the elements.

Step 6: (Statistical Encoding) Huffman encodes the numbers for storage or transmission, assigning common numbers shorter codes than uncommon ones. JPEG produces a 20:1 or even better compression ratio. Decoding a JPEG image requires running the algorithm backward and thus it is roughly symmetric: decoding takes as long as encoding.

Although JPEG is one algorithm, to satisfy the requirements of a broad range of still-image compression applications, it has 4 modes of operation.

2.2 SEQUENTIAL DCT-BASED

In this mode, 8x8 blocks of the image input are formatted for compression by scanning the image left to right and top to bottom. A block consists of 64 samples of one component that make up the image. Each block of samples is transformed to a block of coefficients by the forward discrete cosine transform (FDCT). The coefficients are then quantized and entropy-encoded.

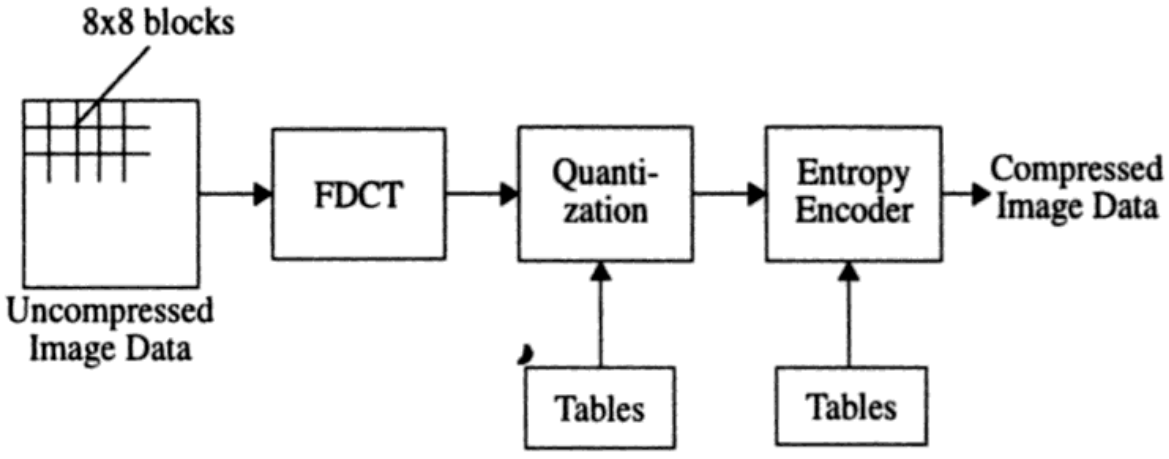


Figure 3: Steps of the lossy sequential DCT-based coding mode

2.2.1 Image Processing

It basically involves the block preparation where the image samples are grouped into 8*8 pixels and passed to the encoder. Then Discrete Cosine Transformation is applied to the blocks where the pixel values are shifted into the range [-128,127] with zero as the center. Each of these values is then transformed using Forward DCT (FDCT). DCT is similar to Discrete Fourier Transformation as it maps the values from the time to the frequency domain.

$$S_{vu} = \frac{1}{4} c_u c_v \sum_{x=0}^{7} \sum_{y=0}^{7} S_{yx} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)}{v\pi}$$

$$\text{where } c_u,\ c_v = \frac{1}{\sqrt{2}} \text{ for u, v} = 0; \text{ otherwise } c_u,\ c_v = 1$$

2.2.2 Quantization

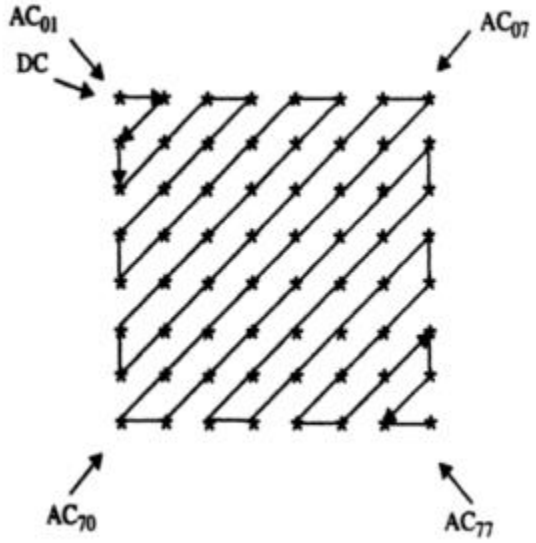
The JPEG application provides a table of 64 entries. Each entry will be used for the quantization of one of the 64 DCT-coefficients. Each of the 64 coefficient can be adjusted separately.

Each table entry is an 8-bit integer value (coefficient Q_{vu}). The quantization process become less accurate as the size of the table entries increase. Quantization and de-quantization must use the same tables.

2.2.3 Entropy Encoding

During the initial step of entropy encoding, the quantized DC-coefficients are treated separately from the quantized AC-coefficients.

- The DC-coefficient determines the basic color of the data units.
- The DCT processing order of the AC coefficients involves the zigzag sequence to concentrate the number of zeros.



AC coefficient are DCT coefficients for which the frequency in one or both dimensions is non zero.

DC coefficient are DCT coefficients for which the frequency in one or both dimensions is zero

JPEG specifies Huffman and arithmetic encoding as entropy encoding methods. However, as this is lossy sequential DCT-based mode, only Huffman encoding is allowed. In lossy sequential mode the framework of the whole picture is not formed but parts of it are drawn i.e. sequentially done.

Figure 4: Sequential picture presentation used in lossy DCT-based mode

2.3 EXPANDED/ PROGRESSIVE DCT-BASED

This method produces a quick low-resolution version of the image, which is gradually (progressively) refined to higher resolutions. This is particularly useful if the medium separating the coder and decoder has a low bandwidth (e.g., a 14.4K modem connection to the Internet, in turn providing a slow connection to a remote image database). The user can stop the download at any time. This is similar to the sequential DCT-based algorithm, but the image is encoded in multiple scans.

Figure 5: Progressive picture presentation used in the expanded lossy DCT-based mode

For the expanded lossy DCT-based mode, JPEG specifies progressive encoding in addition to sequential encoding. At first, a very rough representation of the image appears which is progressively refined until the whole image is formed. This progressive coding is achieved by layered coding. Progressiveness is achieved in two different ways:

- By using a spectral selection in the first run only, the quantized DCT coefficients of low frequencies of each data unit are passed in the entropy encoding. In successive runs, the coefficients of higher frequencies are processed.

- Successive approximation transfers all of the quantized coefficients in each run, but single bits are differentiated according to their significance. The most-significant bits are encoded first, then the less-significant bits.

2.4 LOSSLESS

The decoder renders an exact reproduction of the original digital image.

This mode is used when it is necessary to decode a compressed image identical to the original. Compression ratios are typically only 2:1. Rather than grouping the pixels into 8x8 blocks, data units are equivalent to single pixels. Image processing and quantization use a predictive technique, rather than a transformation encoding one. For a pixel X in the image, one of 8 possible predictors is selected (see table below). The prediction selected will be the one which gives the best result from the *a priori* known values of the pixel's neighbours, A, B, and C. The number of the predictor as well as the difference of the prediction to the actual value is passed to the subsequent entropy encoding.

Selection Value	Prediction
0	No prediction
1	X=A
2	X=B
3	X=C
4	X=A+B-C
5	X=A+(B-C)/2
6	X=B+(A-C)/2
7	X=(A+B)/2

2.5 HIERARCHICAL

The input image is coded as a sequence of increasingly higher resolution frames. The client application will stop decoding the image when the appropriate resolution image has been reproduced.

This mode uses either the lossy DCT-based algorithms or the lossless compression technique. The main feature of this mode is the encoding of the image at different resolutions. The prepared image is initially sampled at a lower resolution (reduced by the factor 2n). Subsequently, the resolution is reduced by a factor 2n-1 vertically and horizontally. This compressed image is then

subtracted from the previous result. The process is repeated until the full resolution of the image is compressed.

Hierarchical encoding requires considerably more storage capacity, but the compressed image is immediately available at the desired resolution. Therefore, applications working at lower resolutions do not have to decode the whole image and then subsequently reduce the resolution.

2.6 MPEG COMPRESSION PROCESS

MPEG stands for Moving Picture Coding Exports Group. MPEG algorithms compress data to form small bits that can be easily transmitted and then decompressed. MPEG achieves its high compression rate by storing only the changes from one frame to another, instead of each entire frame. The video information is then encoded using a technique called *Discrete Cosine Transform (DCT)*. MPEG uses a type of lossy compression, since some data is removed. But the diminishment of data is generally imperceptible to the human eye.

MPEG compression removes two types of redundancies:

Spatial redundancy:

- Pixel values are not independent, but are correlated with their neighbour both within the same frame and across frames. So, to some extent, the value of a pixel is predictable given the values of neighbouring pixels.
- It is removed with the help of DCT compression.

Temporal redundancy:

- Pixels in two video frames that have the same values in the same location (some objects repeated again and again in every frame).
- It is removed with the help of Motion compensation technique Macroblock
- It is the basic hierarchical component used achieving high level of compression.
- The key to achieving a high rate of compression is to remove much redundant information as possible.
- Entropy encoding and Huffman coding are two schemes used for encoding video information.
- MPEG takes the advantage of the fact that there exists a correlation between successive frames of moving pictures.

2.6.1 MPEG Compression

The MPEG compression algorithm encodes the data in 5 steps.

First a ***reduction of the resolution*** is done, which is followed by a motion compensation in order to reduce temporal redundancy. The next steps are the Discrete Cosine Transformation (DCT) and a quantization as it is used for the JPEG compression; this reduces the spatial redundancy

(referring to human visual perception). The final step is an entropy coding using the Run Length Encoding and the Huffman coding algorithm.

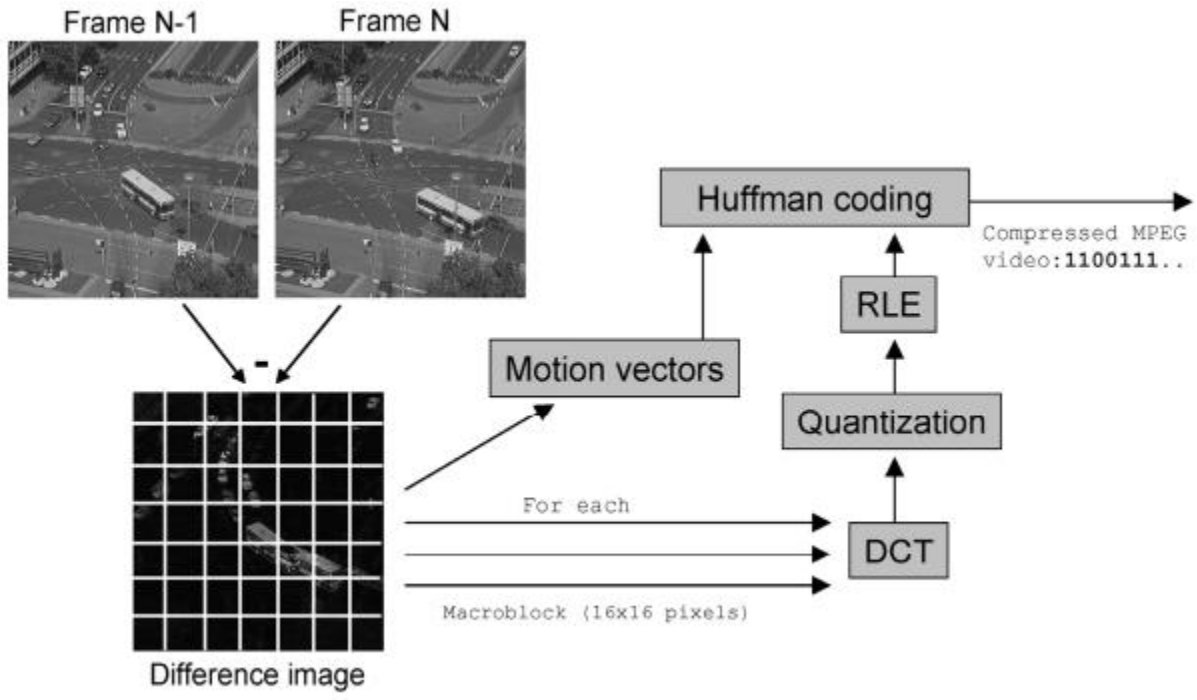


Figure 6: Five steps for a standard MPEG compression

Step 1: Reduction of the Resolution

The human eye has a lower sensibility to colour information than to dark-bright contrasts. A conversion from RGB-colour-space into YUV colour components help to use this effect for compression. The chrominance components U and V can be reduced (subsampling) to half of the pixels in horizontal direction (4:2:2), or a half of the pixels in both the horizontal and vertical (4:2:0).

Step 2: Motion Estimation

An MPEG video can be understood as a sequence of frames. Because two successive frames of a video sequence often have small differences (except in scene changes), the MPEG-standard offers a way of reducing this temporal redundancy. It uses three types of frames:

I-frames (intra), P-frames (predicted) and B-frames (bidirectional)

The ***I-frames*** are "key-frames", which have no reference to other frames and their compression is not that high. The ***P-frames*** can be predicted from an earlier I-frame or P-frame. P-frames cannot be reconstructed without their referencing frame, but they need less space than the I-frames, because only the differences are stored. The ***B-frames*** are a two directional version of the P-frame, referring to both directions (one forward frame and one backward frame). B-frames cannot be referenced by other P- or B-frames, because they are interpolated from forward and

backward frames. P-frames and B-frames are called inter coded frames, whereas I-frames are known as intra coded frames.

The references between the different types of frames are realized by a process called ***motion estimation*** or motion compensation. The correlation between two frames in terms of motion is represented by a motion vector. The resulting frame correlation, and therefore the pixel arithmetic difference, strongly depends on how good the motion estimation algorithm is implemented. The steps involved in motion estimation:

- **Frame Segmentation** - The Actual frame is divided into nonoverlapping blocks (macro blocks) usually 8x8 or 16x16 pixels. The smaller the block sizes are chosen, the more vectors need to be calculated.
- **Search Threshold** - In order to minimize the number of expensive motion estimation calculations, they are only calculated if the difference between two blocks at the same position is higher than a threshold, otherwise the whole block is transmitted.
- **Block Matching** - In general block matching tries to "stitch together" an actual predicted frame by using snippets (blocks) from previous frames.
- **Prediction Error Coding** - Video motions are often more complex, and a simple "shifting in 2D" is not a perfectly suitable description of the motion in the actual scene, causing so called prediction errors.
- **Vector Coding** - After determining the motion vectors and evaluating the correction, these can be compressed. Large parts of MPEG videos consist of B- and P-frames as seen before, and most of them have mainly stored motion vectors.

Step 3: Discrete Cosine Transform (DCT)

DCT allows, similar to the Fast Fourier Transform (FFT), a representation of image data in terms of frequency components. So the frame-blocks (8x8 or 16x16 pixels) can be represented as frequency components. The transformation into the frequency domain is described by the following formula:

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cdot\cos\frac{(2x+1)u\pi}{2N}\cdot\cos\frac{(2y+1)v\pi}{2N}$$

$$C(u), C(v) = 1/\sqrt{2} \ \ for\ u,v = 0$$
$$C(u), C(v) = 1, \ else$$
$$N = block\ size$$

The inverse DCT is defined as:

$$f(x,y) = \frac{1}{4}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)F(u,v)\cos\frac{(2y+1)u\pi}{16}\cos\frac{(2x+1)v\pi}{16}$$

Step 4: Quantization

During quantization, which is the primary source of data loss, the DCT terms are divided by a quantization matrix, which takes into account human visual perception. The human eyes are more reactive to low frequencies than to high ones. Higher frequencies end up with a zero entry after quantization and the domain was reduced significantly.

Step 5: Entropy Coding

The entropy coding takes two steps: Run Length Encoding (RLE) [2] and Huffman coding [1]. These are well known lossless compression methods, which can compress data, depending on its redundancy, by an additional factor of 3 to 4.