Maharaja Agrasen Institute of Technology
ETCS 211
Computer Graphics & Multimedia
UNIT 4

LOSSY COMPRESSION METHODS
JPEG

LOSSY COMPRESSION METHODS

Our eyes and ears cannot distinguish subtle changes. In such cases, we can use a lossy data compression method. These methods are cheaper—they take less time and space when it comes to sending millions of bits per second for images and video. Several methods have been developed using lossy compression techniques. JPEG (Joint Photographic Experts Group) encoding is used to compress pictures and graphics, MPEG (Moving Picture Experts Group) encoding is used to compress video, and MP3 (MPEG audio layer 3) for audio compression.

As discussed, an image can be represented by a two-dimensional array (table) of picture elements (pixels).
A grayscale picture of 307,200 pixels is represented by 2,457,600 bits, and a color picture is represented by 7,372,800 bits.

In JPEG, a grayscale picture is divided into blocks of 8 × 8 pixel blocks to decrease the number of calculations because, as we will see shortly, the number of mathematical operations for each picture is the square of the number of units.

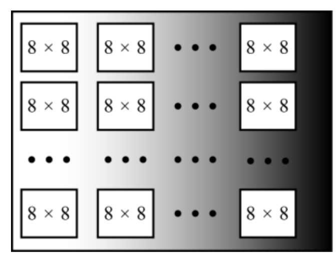


Figure JPEG grayscale example, 640 × 480 pixels

The whole idea of JPEG is to change the picture into a linear (vector) set of numbers that reveals the redundancies. The redundancies (lack of changes) can then be removed using one of the lossless compression methods we studied previously. A simplified version of the process is shown in Figure 15.11.

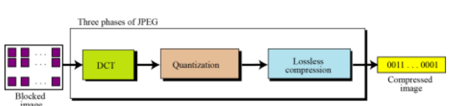


Figure The JPEG compression process

Discrete cosine transform (DCT)

In this step, each block of 64 pixels goes through a transformation called the discrete cosine transform (DCT). The transformation changes the 64 values so that the relative relationships between pixels are kept but the redundancies are revealed. The formula is given in Appendix G. $P(x, y)$ defines one value in the block, while $T(m, n)$ defines the value in the transformed block.

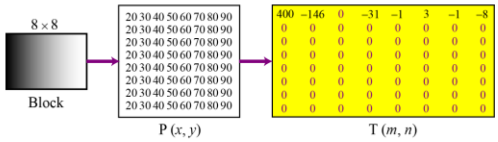


Figure Case 3: gradient grayscale

Quantization

After the T table is created, the values are quantized to reduce the number of bits needed for encoding. Quantization divides the number of bits by a constant and then drops the fraction. This reduces the required number of bits even more. In most implementations, a quantizing table (8 by 8) defines how to quantize each value. The divisor depends on the position of the value in the T table. This is done to optimize the number of bits and the number of 0s for each particular application.

Compression

After quantization the values are read from the table, and redundant 0s are removed. However, to cluster the 0s together, the process reads the table diagonally in a zigzag fashion rather than row by row or column by column. The reason is that if the picture does not have fine changes, the bottom right corner of the T table is all 0s.

JPEG usually uses run-length encoding at the compression phase to compress the bit pattern resulting from the zigzag linearization.

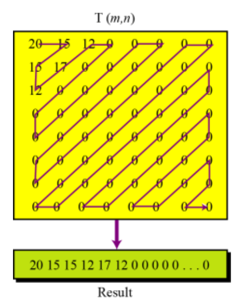


Figure Reading the table

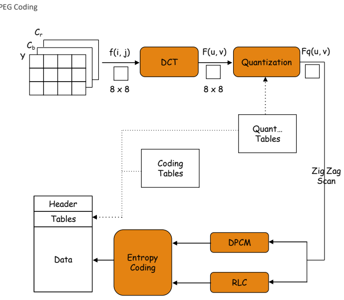
Why JPEG

The *compression ratio* of lossless methods (e.g., Huffman, Arithmetic, LZW) is not high enough for image and video compression.

JPEG uses *transform coding*, it is largely based on the following observations:

- Observation 1: A large majority of useful image contents change relatively slowly across images, i.e., it is unusual for intensity values to alter up and down several times in a small area, for example, within an 8 x 8 image block.
  A translation of this fact into the spatial frequency domain, implies, generally, lower spatial frequency components contain **more information** than the high frequency components which often correspond to less useful details and noises.
- Observation 2: Experiments suggest that humans are more immune to loss of higher spatial frequency components than loss of lower frequency components.

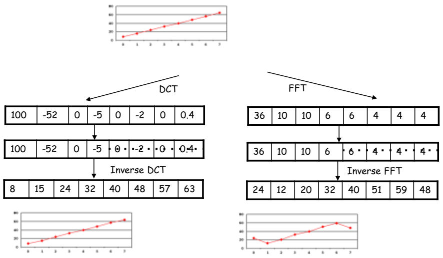
JPEG Coding



Steps Involved:

1. Discrete Cosine Transform of each 8x8 pixel array f(x,y) □ $_T$ F(u,v)
2. Quantization using a table or using a constant
3. Zig-Zag scan to exploit redundancy
4. Differential Pulse Code Modulation(DPCM) on the DC component and Run length Coding of the AC components
5. Entropy coding (Huffman) of the final output

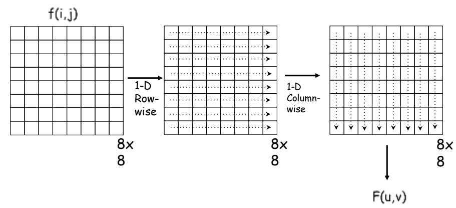
Example and Comparison



Example Description:
- f(n) is given from n = 0 to 7; (N=8)
- Using DCT(FFT) we compute F(ω) for ω = 0 to 7
- We truncate and use Inverse Transform to compute f'(n)

2-D DCT

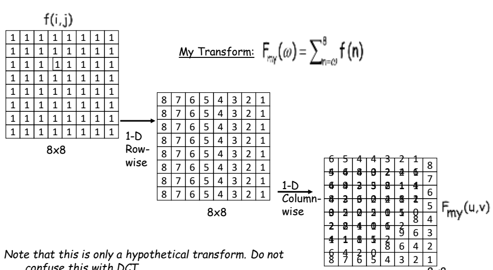
Images are two-dimensional; How do you perform 2-D DCT?
- Two series of 1-D transforms result in a 2-D transform as demonstrated in the figure below



- F(0,0) is called the DC component and the rest of F(i,j) are called AC components

2-D Transform Example

The following example will demonstrate the idea behind a 2-D transform by using our own cooked up transform: The transform computes a running cumulative sum.



My Transform: $F_{my}(\omega) = \sum_{n=\omega}^{8} f(n)$

Note that this is only a hypothetical transform. Do not confuse this with DCT

Quantization

Why? -- To reduce number of bits per sample
F'(u,v) = round(F(u,v)/q(u,v))

Example: 101101 = 45 (6 bits).
Truncate to 4 bits: 1011 = 11. (Compare 11 x 4 =44 against 45)
Truncate to 3 bits: 101 = 5. (Compare 8 x 5 =40 against 45)
Note, that the more bits we truncate the more precision we lose

Quantization error is the main source of the Lossy Compression.

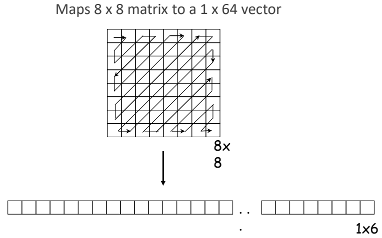
Uniform Quantization:
- q(u,v) is a constant.

Non-uniform Quantization -- Quantization Tables
- Eye is most sensitive to low frequencies (upper left corner in frequency matrix), less sensitive to high frequencies (lower right corner)
- Custom quantization tables can be put in image/scan header.
- JPEG Standard defines two default quantization tables, one each for luminance and chrominance.

Zig-Zag Scan

Why? -- to group low frequency coefficients in top of vector and high frequency coefficients at the bottom

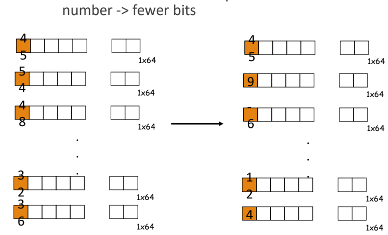
Maps 8 x 8 matrix to a 1 x 64 vector



DPCM on DC Components

The DC component value in each 8x8 block is large and varies across blocks, but is often close to that in the previous block.

Differential Pulse Code Modulation (DPCM): Encode the difference between the current and previous 8x8 block. Remember, smaller number -> fewer bits



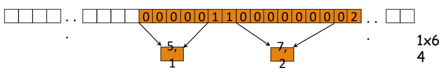
RLE on AC Components

The 1x64 vectors have a lot of zeros in them, more so towards the end of the vector.

- Higher up entries in the vector capture higher frequency (DCT) components which tend to be capture less of the content.
- Could have been as a result of using a quantization table

Encode a series of 0s as a (*skip*,*value*) pair, where *skip* is the number of zeros and *value* is the next non-zero component.

- Send (0,0) as end-of-block sentinel value.



Entropy Coding: DC Components

DC components are differentially coded as (**SIZE**,**Value**)

- The code for a **Value** is derived from the following table

SIZE	Value	Code
0	0	---
1	-1,1	0,1
2	-3, -2, 2,3	00,01,10,11
3	-7,…, -4, 4,…, 7	000,…, 011, 100,…111
4	-15,…, -8, 8,…, 15	0000,…, 0111, 1000,…, 1111
.		.
.		.
11	-2047,…, -1024, 1024,… 2047	…

Size_and_Value Table

Entropy Coding: DC Components (Contd..)

DC components are differentially coded as (**SIZE**,**Value**)

- The code for a **SIZE** is derived from the following table

SIZE	Code Length	Code
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Huffman Table for DC component SIZE field

Example: If a DC component is 40 and the previous DC component is 48. The difference is -8. Therefore it is coded as:
1010111
0111: The value for representing –8 (see Size_and_Value table)
101: The size from the same table reads 4. The corresponding code from the table at left is 101.

Entropy Coding: AC Components

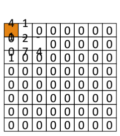
AC components (range –1023..1023) are coded as (S1,S2 pairs):

- **S1: (RunLength/SIZE)**
  - **RunLength:** The length of the consecutive zero values [0..15]
  - **SIZE:** The number of bits needed to code the *next* nonzero AC component's value. [0-A]
  - (0,0) is the End_Of_Block for the 8x8 block.
  - **S1** is Huffman coded (see AC code table below)
- **S2: (Value)**
  - **Value:** Is the value of the AC component.(refer to size_and_value table)

Run/ SIZE	Code Length	Code
0/0	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011

Run/ SIZE	Code Length	Code
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
… 15/A	More	Such rows

Entropy Coding: Example



Example: Consider encoding the AC components by arranging them in a zig-zag order -> 12,10, 1, -7 2 0s, -4, 56 zeros

12: read as zero 0s,12: (0/4)12 10111100
1011: The code for (0/4 from AC code table)
1100: The code for 12 from the Size_and_Value table.
10: (0/4)10 10111010
1: (0/1)1 001
-7: (0/3)-7 100000
2 0s, -4: (2/3)-4 1111110111011
1111110111: The 10-bit code for 2/3
011: representation of –4 from Size_and_Value table.
56 0s: (0,0) 1010 (Rest of the components are zeros therefore we simply put the EOB to signify this fact)

JPEG Modes

Sequential Mode:

Each image is encoded in a single left-to-right, top-to-bottom scan.

- The technique we have been discussing so far is an example of such a mode, also referred to as the **Baseline Sequential Mode.**
- It supports only 8-bit images as opposed to 12-bit images as described before.

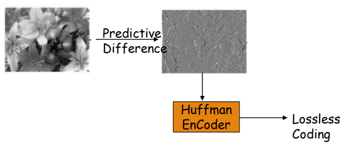
JPEG Modes

Lossless Mode:

Truly lossless

It is a predictive coding mechanism as opposed to the baseline mechanism which is based on DCT and quantization(the source of the loss).

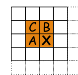
Here is the simple block diagram of the technique:



Lossless Mode (Contd..)

Predictive Difference:

- For each pixel a predictor (one of 7 possible) is used that *best* predicts the value contained in the pixel as a combination of up to 3 neighboring pixels.
- The difference between the predicted value and the actual value (**X**)contained in the pixel is used as the *predictive difference* to represent the pixel.
- The predictor along with the predictive difference are encoded as the pixel's content.
- The series of pixel values are encoded using huffman coding



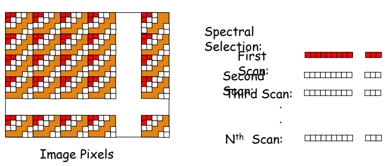
Predictor	Prediction
P1	A
P2	B
P3	C
P4	A+B-C
P5	A + (B-C)/2
P6	B + (A-C)/2
P7	(A+B)/2

Notes:
- The very first pixel in location (0, 0) will always use itself.
- Pixels at the first row always use P1,
- Pixels at the first column always use P2.
- The best (of the 7) predictions is always chosen for any pixel.

JPEG Modes

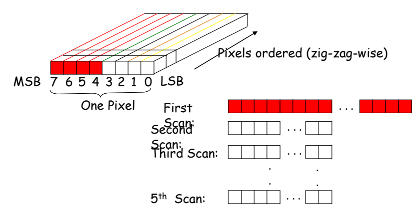
Progressive Mode: It allows a coarse version of an image to be transmitted at a low rate, which is then progressively improved over subsequent transmissions.

- *Spectral Selection* : Send DC component and first few AC coefficients first, then gradually some more ACs.



Progressive Mode

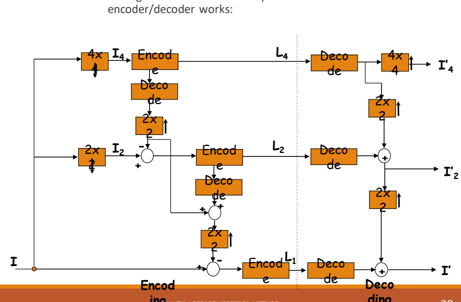
Successive Approximation : All the DCT components are sent few bits at a time: For example, send n1 (say,4) bits (starting with MSB) of all pixels in the first scan, the next n2(say 1) bits of all pixels in the second and so on.



Hierarchical Mode

Used primarily to support multiple resolutions of the same image which can be chosen from depending on the target's capabilities.

The figure here shows a description of how a 3-level hierarchical encoder/decoder works:



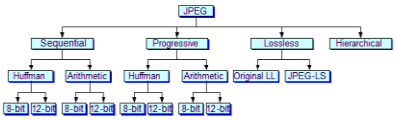
JPEG STANDARDS FOR COMPRESSING STILL IMAGES

The lossy JPEG

JPEG is an acronym for Joint Photographic Experts Group and JFIF is an acronym JPEG File Interchange Format. Normally, the files with the .JPG extension are JFIF files. If a JPEG file is loaded into a text editor such as Window's NotePad , the string JFIF is started in the first line starting at the 7th character. JPEG compression is mainly discards much of the original information so the exact original image cannot be reconstructed from a JPEG file. This is called lossy compression. While this sounds bad, a photograph actually contains considerable information that the human eye cannot detect so this can be safely discarded.

JPEG Compression Modes

The JPEG standard defined four compression modes: Hierarchical, Progressive, Sequential and lossless. Figure 0 shows the relationship of major JPEG compression modes and encoding processes.



Sequential: Sequential-mode images are encoded from top to bottom. Sequential mode supports sample data with 8 and 12 bits of precision. In the sequential JPEG, each color component is completely encoded in single scan. Within sequential mode, two alternate entropy encoding processes are defined by the JPEG standard: one uses Huffman encoding; the other uses arithmetic coding.
Progressive: In progressive JPEG images, components are encoded in multiple scans. The compressed data for each component is placed in a minimum of 2 and as many as 896 scans. The initial scans create a rough version of the image, while subsequent scans refine it.
Lossless: preserves exact, original image, small compression ration, less use
Hierarchical: JPEG is a super-progressive mode in which the image Is broken down into a number of subimages called frames. A frame is a collection of one or more scans. In hierarchical mode, the first frame creates a low-resolution version of image. The remaining frames refine the image by increasing the solution.

Baseline JPEG

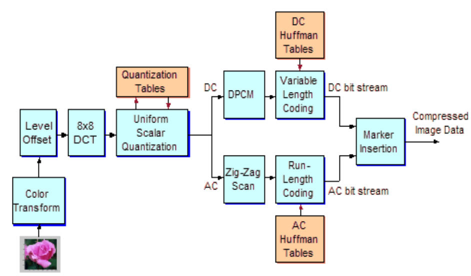
Baseline JPEG decompressor supports a minimal set of features. It must be able to decompress image using sequential DCT-based mode. For baseline compression the bit depth must be *B*=8; however, it is convenient to describe a more general situation. The image samples are assumed to be unsigned quantities in the range [0, 2^B-1]. The level offset subtract 2^(B-1) from every sample value so as to produce signed quantities in the range [-2^(B-1), 2^(B-1)-1]. The purpose of this is to ensure that all the DCT coefficients will be signed quantities with a similar dynamic range. The image is partitioned into blocks of size 8×8. Each block is then independently transformed using the 8×8 DCT. If the image dimensions are not exact multiples of 8, the blocks on the lower and right hand boundaries may be only partially occupied. These boundary blocks must be padded to the full 8×8 block size and processed in an identical fashion to every other block. The compressor is free to select the value used to pad partial boundary blocks.

Color Space

The VGA card displays colors by setting the intensity of the three colors RED, BLUE and GREEN. The three colors form the axis of a cartesian coordinate system. Any color is then a point in this color space. A grayscale image is formed by only using points in color space where RED, BLUE and GREEN intensities are all equal. The alternate set of axis so called luminance and chrominance. The luminance axis is labelled Y, the blue chrominance axis is labelled U and the red chrominance axis is labelled V. The three new axes we have created form the three components used in JPEG image files. The following formulas will convert between the two coordinate systems.

Y = 0.299 R + 0.587 G + 0.114 B U = -0.1687 R - 0.3313 G + 0.5 B + 128 V = 0.5 R - 0.4187 G - 0.0813 B + 128 R = Y + 1.402 (V-128) G = Y - 0.34414(U-128) - 0.71414(V-128) B = Y + 1.772 (U-128)

here is a reason for using the YUV color space. The human eye is more sensitive to luminance than to chrominance. Typically JPEGs throw out 3/4 of the chrominance information before any other compression takes place. This reduces the amount of information to be stored about the image by 1/2. With all three components fully stored, 4 pixels needs 3 x 4 = 12 component values. If 3/4 of two components are discarded we need 1 x 4 + 2 x 1 = 6 values.
The largest horizontal and vertical sample factors determine the height and width of the Minimum Coded Unit (MCU) respectively. For the case of figure 4 the MCU would be two 8×8 blocks high and two 8×8 blocks wide for a total of four 8×8 blocks. The blocks are stored in the file all Ys first then all Us then all Vs. For this case one MCU would contain four Y 8×8 blocks followed by one U 8×8 block and one V 8×8



MPEG STANDARD FOR COMPRESSING VIDEO

The name MPEG is an acronym for Moving Pictures Experts Group. MPEG is a method for video compression, which involves the compression of digital images and sound, as well as synchronization of the two.

There currently are several MPEG standards.

MPEG-1 is intended for intermediate data rates, on the order of 1.5 Mbit/sec.

MPEG-2 is intended for high data rates of at least 10 Mbit/sec.

MPEG-3 was intended for HDTV compression but was found to be redundant and was merged with MPEG-2.

MPEG-4 is intended for very low data rates of less than 64 Kbit/sec.

i. In principle, a motion picture is a rapid flow of a set of frames, where each frame is an image. In other words, a frame is a spatial combination of pixels, and a video is a temporal combination of frames that are sent one after another.

ii. Compressing video, then, means spatially compressing each frame and temporally compressing a set off names.

iii. Spatial Compression: The spatial compression of each frame is done with JPEG (ora modification of it). Each frame is a picture that can be independently compressed.

iv. Temporal Compression: In temporal compression, redundant frames are removed.

v. To temporally compress data, the MPEG method first divides frames into three categories:

vi. I-frames, P-frames, and B-frames. Figure1 shows a sample sequence off names.

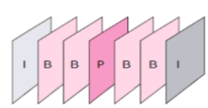
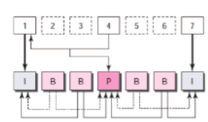


Fig1: MPEG frames

vii. Figure2 shows how I-, P-, and B-frames are constructed from a series of seven frames.



I-frames: An intracoded frame (I-frame) is an independent frame that is not related to any other frame.

They are present at regular intervals. An I-frame must appear periodically to handle some sudden change in the frame that the previous and following frames cannot show. I-frames are independent of other frames and cannot be constructed from other frames.

P-frames: A predicted frame (P-frame) is related to the preceding I-frame or P-frame. In other words, each P-frame contains only the changes from the preceding frame. P-frames can be constructed only from previous I- or P-frames. P-frames carry much less information than other frame types and carry even fewer bits after compression.

B-frames: A bidirectional frame (B-frame) is related to the preceding and following I-frame or P-frame. In other words, each B-frame is relative to the past and the future. Note that a B-frame is never related to another B-frame.

The luminance component contains the gray scale picture & the chrominance components provide the color, hue & saturation.

The MPEG decoder has three parts, audio layer, video layer, system layer.

The system layer reads and interprets the various headers in the source data and transmits this data to either audio or video layer.

The basic building block of an MPEG picture is the macro block as shown:

The macro block consist of 16×16 block of luminance gray scale samples divided into four 8×8 blocks of chrominance samples.

The MPEG compression of a macro block consist of passing each of the °6 blocks their DCT quantization and entropy encoding similar to JPEG.

A picture in MPEG is made up of slices where each slice is continuous set of macro blocks having a similar gray scale component.

The concept of slice is important when a picture contains uniform areas.

The MPEG standard defines a quantization stage having values (1, 31). Quantization for intra coding is:

QDCT =(16×DCT)+sign(DCT)×quantizer scale)2×quantizer - scale×θ

Where

DCT = Discrete cosine transform of the coefficienting encoded

Q = Quantization coefficient from quantization table

Quantization rule for encoding,

QDCT =16×DCT2×quantizer - scale×θQDCT =16×DCT2×quantizer - scale×θ

The quantized numbers Q_(DCT)are encoded using non adaptive Haffman method and the standard defines specific Haffman code tables which are calculated by collecting statistics.

VIVA VOICE

1. What is the difference between lossless and lossy compression techniques?

2. What are the advantages of huffman encoding?

3. Why are compression techniques used?

4. Define compression & decompression?

5. What is the need for compression?

6 What are the different types of compression available? List the difference between them

7. Write down the difference between jpeg2 and jpeg4

8. List the various compression techniques

9. List the benefits of DCT

10 Define quantization

11. Differentiate MPEP2 and MPEG 4

12. Write short notes on jpeg
13. Discuss the various levels of RAID technologies
14. Write short notes on mpeg
15. Write notes on file format standard
16. Explain the multimedia I/O technologies
17. With a neat diagram explain the architecture of MPEG standard
18. What is MIDI? Explain how it is used for music recording