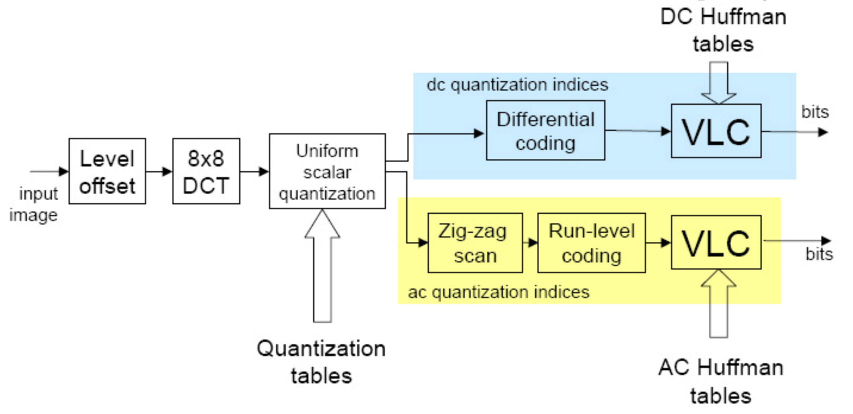
8. JPEG and JPEG 2000 PICTURE CODING

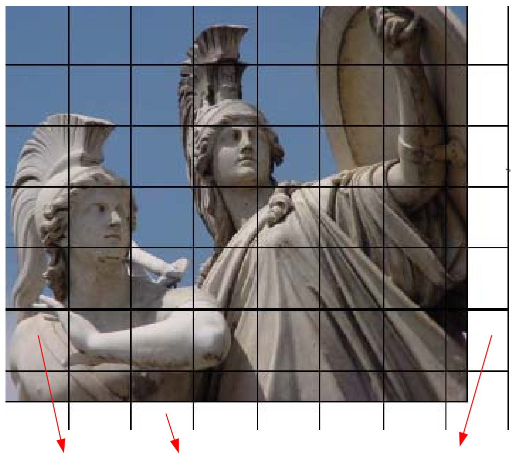
General idea: Joint Photographic Experts Group (JPEG), formally known as ISO/IEC JTC1/SG29/WG1 standard, the work started in 1983 and draft international standard was formulated in 1991. It is widely used for image exchange, storing imagery, www, and digital photography. All processing operations are performed after segmenting images into 8x8 blocks and performing discrete cosine transform. The transformed coefficients are then quantized and entropy coded either by an arithmetic coder (QM-coder with binary decision tree) or by Huffman coding. Motion JPEG (MJPEG) is uniformly used for digital video editing. The performance of the technique is very much dependent on the content of the imagery at hand and the constraints the user sets. With the introduction of JPEG2000, the quality has significantly improved.

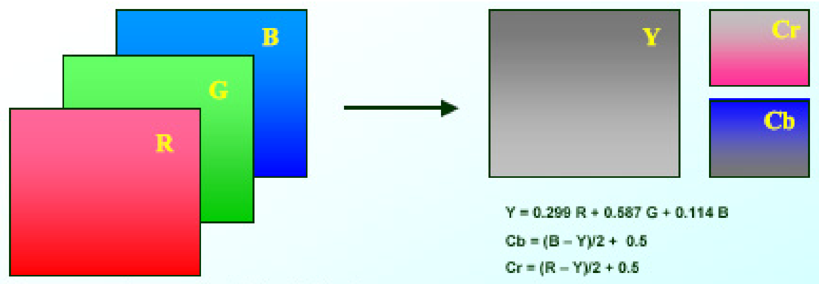


- **Color Transform (RGB → YCbCr);**
- **Image Partition;**
- **Discrete Cosine Transform;**
- **Quantization;**
- **DC Coefficient Encoding;**
- **Zig-zag ordering of AC Coefficients;**
- **Entropy Coding.**

JPEG Baseline Compression and Key Steps

Image arrangement and color transformation:





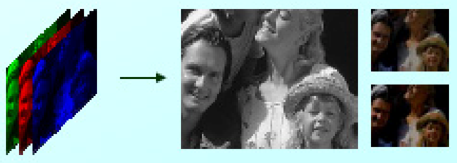
Y = 0.299 R + 0.587 G + 0.114 B

Cb = (B – Y)/2 + 0.5

Cr = (R – Y)/2 + 0.5

The human eye is more sensitive to luminance than to chrominance. Typically JPEG 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.

Example:



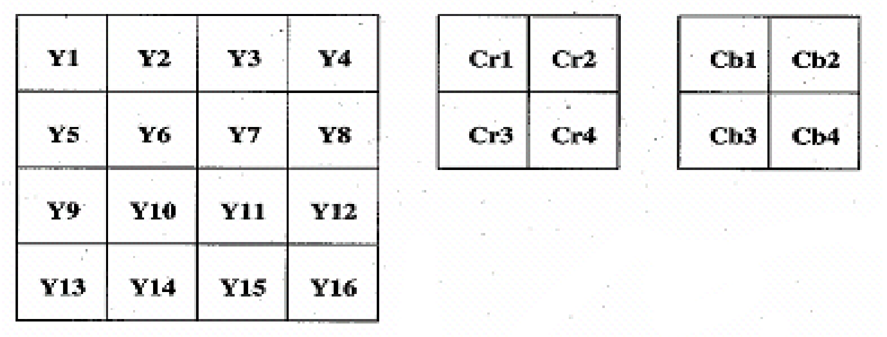
Baseline Algorithm: Images are padded with extra pixels to conform nice 8x8 blocks. Color transformation for better mimicking the visual perception is handled through the following mapping:

$$Y = 0.3 * R + 0.6 * G + 0.1 * B; \quad C_r = \frac{B - Y}{2} + 0.5; \quad C_b = \frac{R - Y}{1.6} + 0.5 \qquad (8.1)$$

1. **DCT Computation:** Over 8x8 blocks, a constant intensity-level of 128 is subtracted from each pixel of 8 bits resolution. Then 2-D DCT is performed.
2. **Quantization Matrix:** DCT coefficients are threshold quantized using a quantization matrix (QM), and then reordered using a zig-zag scanning.
3. **Variable-rate Code (VLC) assignment:** DC coefficient is coded by a DPCM coder relative to the DC of the previous block. Non-zero AC terms are Huffman coded.
4. Chrominance channels are sub-sampled by a factor of 2 in both directions depending on the selection of the case:

- Subsampling

4:4:4	(no subsampling)
4:2:2	(Cr, Cb horizontal subsampling)
4:2:0	(Cr, Cb horizontal + vertical subsampling)



5. Pixels or color images are either non-interleaved (three scans) or interleaved to require a single scan.
6. Non-interleaved and interleaved orderings:

   Scan 1: Y1, Y2, Y3,…,Y16
   Scan 2: Cr1, Cr2, Cr3, Cr4
   Scan 3: Cb1, Cb2, Cb3, Cb4

whereas the interleaved ordering is structured as:

Y1, Y2, Y3, Y4, Cr1, Cb1, Y5, Y6, Y7, Y8, Cr2, Cb2,…

7. Quantization Tables for Luminance and chrominance are given by the matrices QM:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	36	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Example 8. 1: Perform JPEG baseline coding for the following 8x8 DCT blocks:

Level shifted 8 × 8 original image block (shifted by 128)

-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

Forward DCT Values

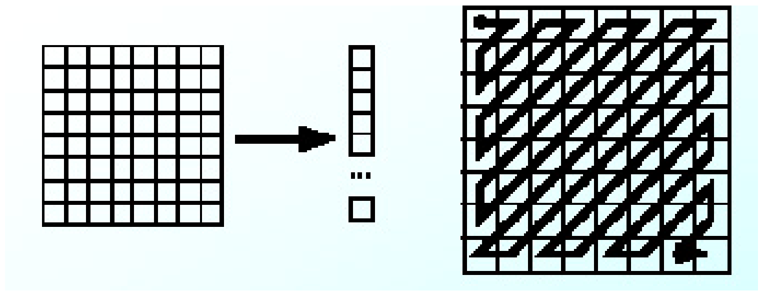
-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

JPEG recommended quantization matrix from step 7

-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

Quantized coefficient array

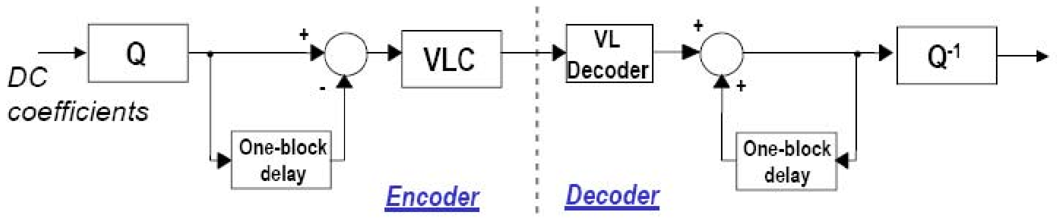


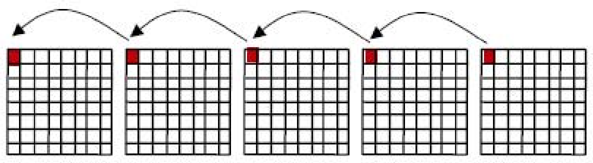
- 1-D coefficient sequence after zig-zag scanning:

-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

where EOB denotes the end of the block.

- Coding the DC coefficient: Encode the difference between DC coefficients of the current and previous blocks.





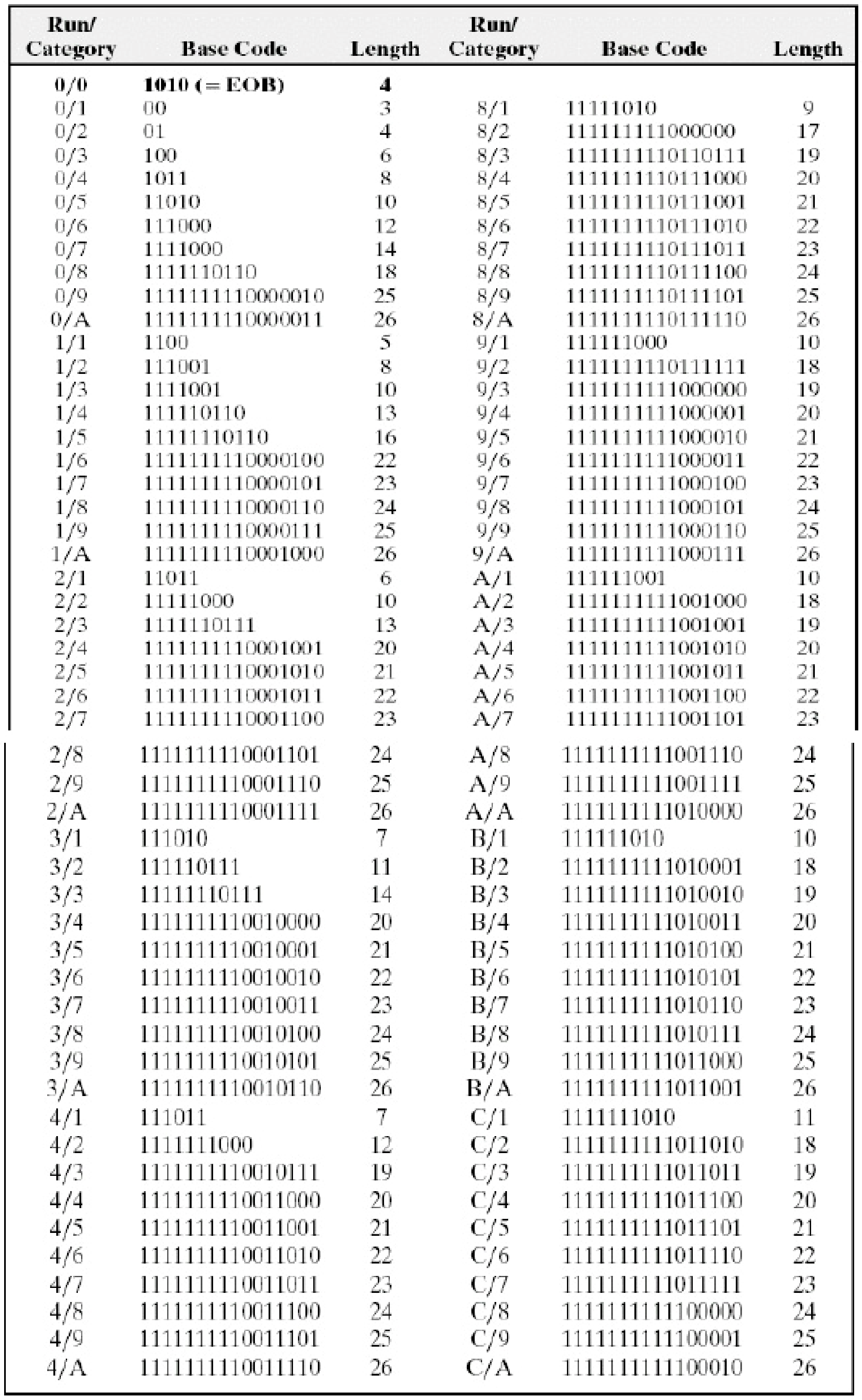
- Actual bit assignment is done according to categories for both DC and AC coefficients:

Range	DC Difference Category	AC Category
0	0	N/A
−1, 1	1	1
−3, −2, 2, 3	2	2
−7, . . . , −4, 4, . . . , 7	3	3
−15, . . . , −8, 8, . . . , 15	4	4
−31, . . . , −16, 16, . . . , 31	5	5
−63, . . . , −32, 32, . . . , 63	6	6
−127, . . . , −64, 64, . . . , 127	7	7
−255, . . . , −128, 128, . . . , 255	8	8
−511, . . . , −256, 256, . . . , 511	9	9
−1023, . . . , −512, 512, . . . , 1023	A	A
−2047, . . . , −1024, 1024, . . . , 2047	B	B
−4095, . . . , −2048, 2048, . . . , 4095	C	C
−8191, . . . , −4096, 4096, . . . , 8191	D	D
−16383, . . . , −8192, 8192, . . . , 16383	E	E
−32767, . . . , −16384, 16384, . . . , 32767	F	N/A

- Default DC codes:

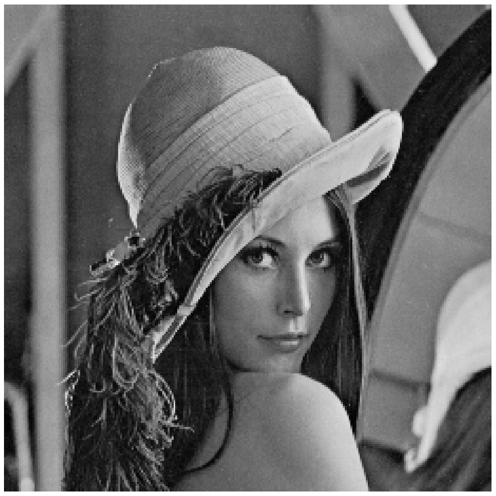
Category	Base Code	Length	Category	Base Code	Length
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

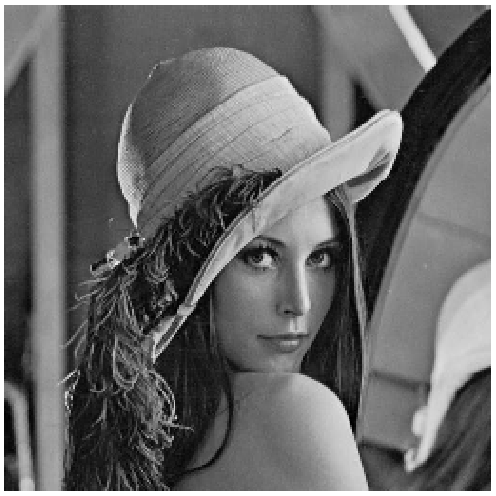
- Coding the AC coefficients: Define (RUN,LEVEL) as symbols: (1,-3); (1,1); (1,-3); ...,(2,-1)
- These symbols are VLC (Huffman or arithmetic) coded according to the run/category.

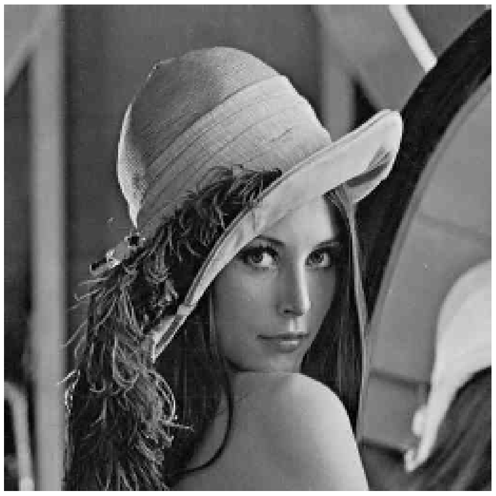


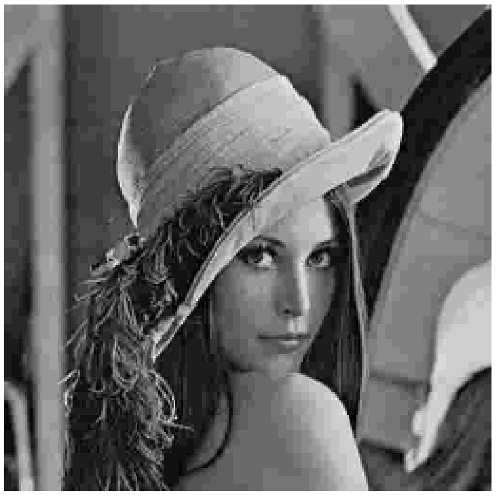
Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
0/0	**1010 (= EOB)**	**4**			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	111111111000000	17
0/3	100	6	8/3	1111111110110111	19
0/4	1011	8	8/4	1111111110111000	20
0/5	11010	10	8/5	1111111110111001	21
0/6	111000	12	8/6	1111111110111010	22
0/7	1111000	14	8/7	1111111110111011	23
0/8	1111110110	18	8/8	1111111110111100	24
0/9	1111111110000010	25	8/9	1111111110111101	25
0/A	1111111110000011	26	8/A	1111111110111110	26
1/1	1100	5	9/1	111111000	10
1/2	111001	8	9/2	1111111110111111	18
1/3	1111001	10	9/3	1111111111000000	19
1/4	111110110	13	9/4	1111111111000001	20
1/5	11111110110	16	9/5	1111111111000010	21
1/6	1111111110000100	22	9/6	1111111111000011	22
1/7	1111111110000101	23	9/7	1111111111000100	23
1/8	1111111110000110	24	9/8	1111111111000101	24
1/9	1111111110000111	25	9/9	1111111111000110	25
1/A	1111111110001000	26	9/A	1111111111000111	26
2/1	11011	6	A/1	111111001	10
2/2	11111000	10	A/2	1111111111001000	18
2/3	1111110111	13	A/3	1111111111001001	19
2/4	1111111110001001	20	A/4	1111111111001010	20
2/5	1111111110001010	21	A/5	1111111111001011	21
2/6	1111111110001011	22	A/6	1111111111001100	22
2/7	1111111110001100	23	A/7	1111111111001101	23
2/8	1111111110001101	24	A/8	1111111111001110	24
2/9	1111111110001110	25	A/9	1111111111001111	25
2/A	1111111110001111	26	A/A	1111111111010000	26
3/1	111010	7	B/1	111111010	10
3/2	111110111	11	B/2	1111111111010001	18
3/3	11111110111	14	B/3	1111111111010010	19
3/4	1111111110010000	20	B/4	1111111111010011	20
3/5	1111111110010001	21	B/5	1111111111010100	21
3/6	1111111110010010	22	B/6	1111111111010101	22
3/7	1111111110010011	23	B/7	1111111111010110	23
3/8	1111111110010100	24	B/8	1111111111010111	24
3/9	1111111110010101	25	B/9	1111111111011000	25
3/A	1111111110010110	26	B/A	1111111111011001	26
4/1	111011	7	C/1	1111111010	11
4/2	1111111000	12	C/2	1111111111011010	18
4/3	1111111110010111	19	C/3	1111111111011011	19
4/4	1111111110011000	20	C/4	1111111111011100	20
4/5	1111111110011001	21	C/5	1111111111011101	21
4/6	1111111110011010	22	C/6	1111111111011110	22
4/7	1111111110011011	23	C/7	1111111111011111	23
4/8	1111111110011100	24	C/8	1111111111100000	24
4/9	1111111110011101	25	C/9	1111111111100001	25
4/A	1111111110011110	26	C/A	1111111111100010	26

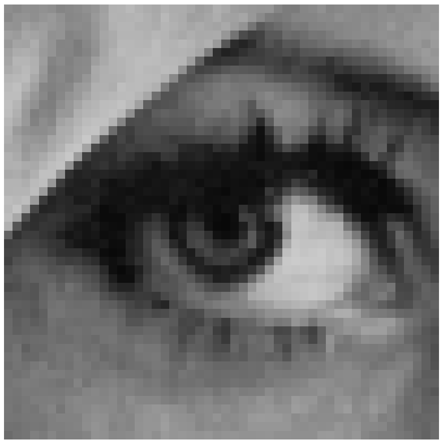
Example 8. 2: Performance of JPEG decoder, which implements the inverse operations and the following is obtained: The reconstruction errors varies in (-25,+25). This is considered a reasonable level of JPEG compression.

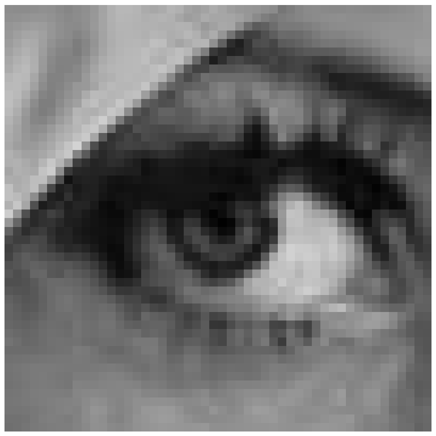

Lena: Original (bpp = 8.0; mse = 0.00)

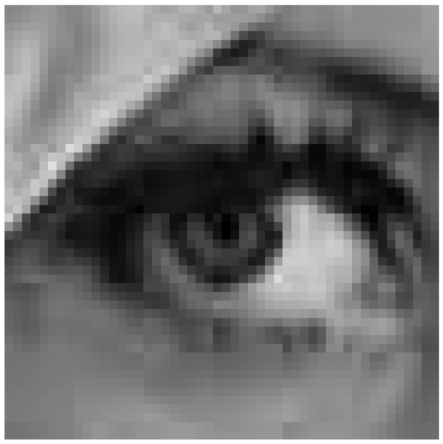

JPEG (bpp = 1.00; mse = 17.26)

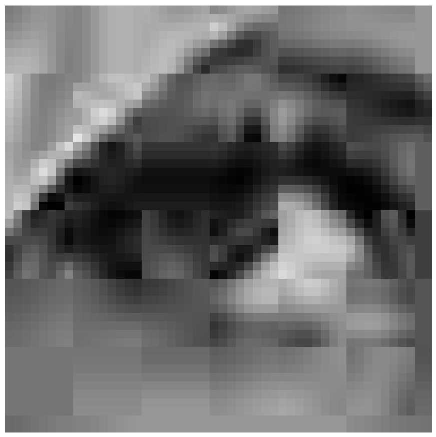

JPEG: (bpp = 0.50; mse = 33.08)


JPEG: (bpp = 0.25; mse = 79.11)


Original: (bpp = 8.00; mse = 0.00)


JPEG: (bpp = 1.00; mse = 17.26)


JPEG: (bpp = 0.50; mse = 33.08)

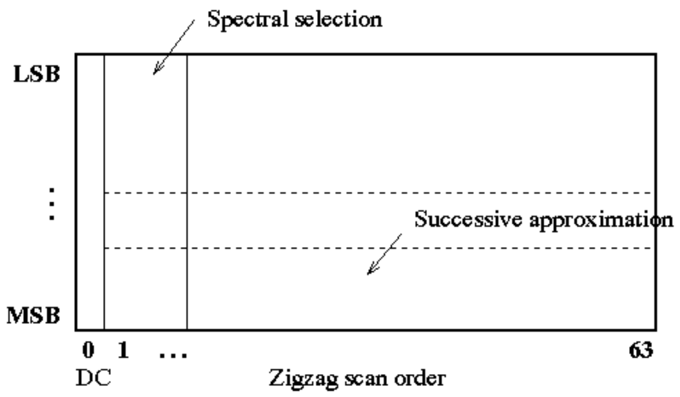

JPEG: (bpp = 0.25; mse = 79.11)

Magnifications of Lena compressed by JPEG.

Example 8. 3: Explore JPEG encoding using the following two demos:
1. VCDemo from Delft University: http://www-ict.its.tudelft.nl/vcdemo.
2. JPEG Color Imagery Compression Demo from Simon Fraser University in Burnaby, B.C., Canada:http://www.cs.sfu.ca/CC/365/mark/interactive-jpeg/Ijpeg.html

JPEG Modes:

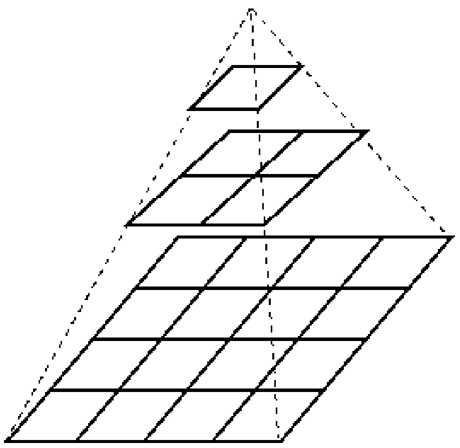
- Sequential/Baseline mode
- Lossless mode
- Progressive mode
- Hierarchical mode
- Motion JPEG: Sequential JPEG is applied to each image frame in a video.



JPEG – Progressive Level:

- The progressive mode consists of a sequence of ``scans" each of which codes a part of the quantized DCT coefficients.
- Spectral selection: The DCT coefficients are grouped into spectral bands. The lower frequency bands are usually coded (sent) first.
- Successive approximation: The information is first sent with lower precision, and then refined in later scans. Two processes may be combined to provide a graceful progression.

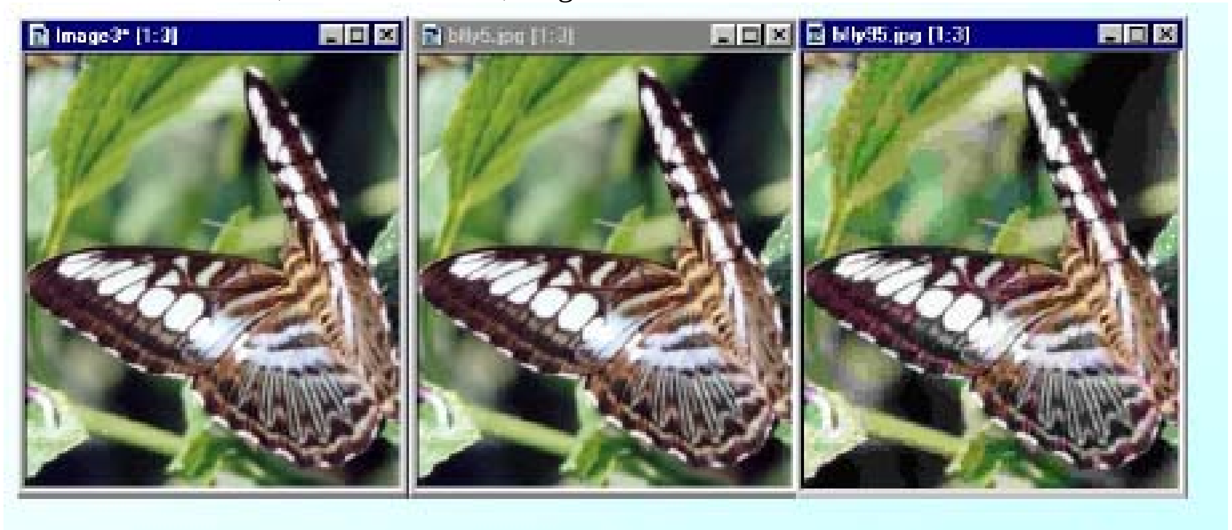
JPEG – Hierarchical Level:



- The first stage (lowest resolution) is coded using one of the sequential or progressive JPEG modes. The output of each hierarchical stage is then up-sampled (interpolated) and used as the prediction for the next stage.
- The image quality at extremely low bit rates surpasses any of the other JPEG modes, but this is achieved at the expense of a higher bit rate at the completion of the progression.

JPEG - Adaptive Quantization

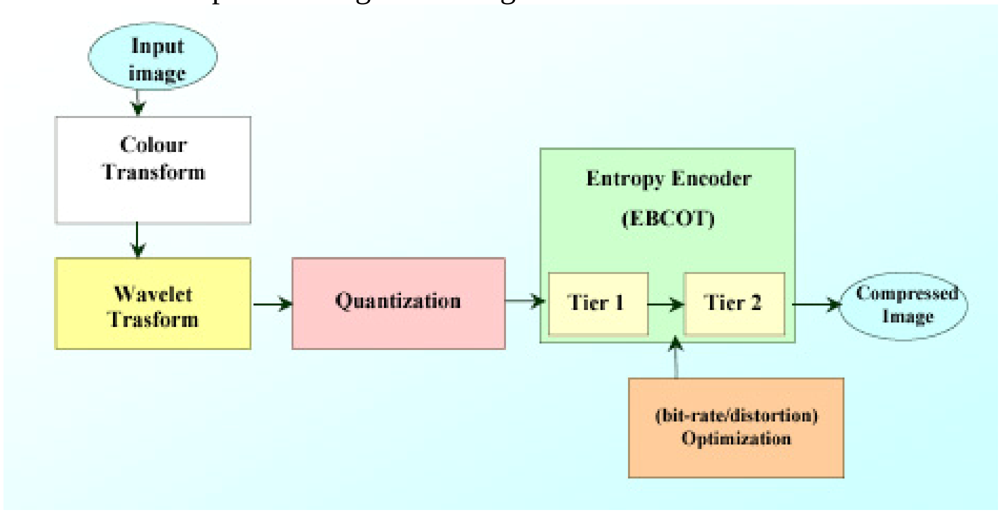
- Allows spatially adaptive quantization, where the quantization matrix can be scaled block-to-block.
  - e.g., separate between high-activity (edges), medium activity (texture) and uniform blocks based on a measure of the intensity variance.
  - provides up to 30% better performance as compared to non-adaptive quantization.
- ISO DIS 10918-3, JPEG Extension, August 1994.

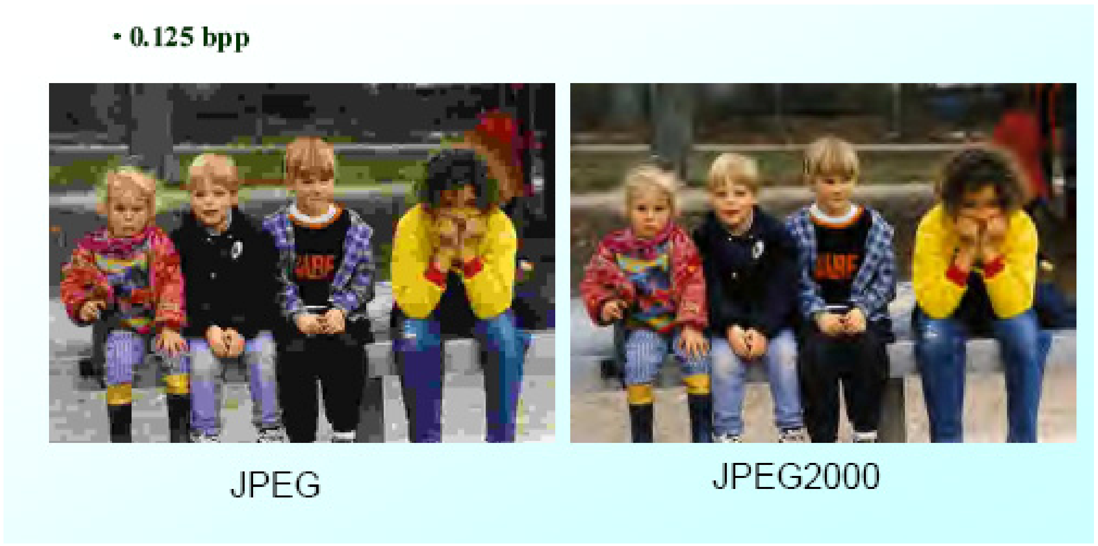


Original Uncompressed (3.2MB) JPEG low level (179 KB) JPEG High level (15 KB)

JPEG2000

Block diagram of the JPEG2000 standard and its impact on compression can be clearly seen from a small demo at 0.125 bits/pixel coding of an image.





JPEG2000 is a Wavelet-based compression and as in the JPEG and other DCT-based compression systems it consists of the following stages:

- Filtering
- Quantizer
- Entropy coding
- Arithmetic coding
- Bit allocation

1. **Aim** of the JPEG2000 was to develop a new still image coding standard for different types of still images (bi-level, gray-level, color, multi-component, hyper-component), with different characteristics (natural, scientific, medical, remote sensing, text, rendered graphics, compound, etc.), allowing different imaging models (client/server, real-time transmission, image library archival, limited buffer and bandwidth resources, etc.) preferably within a unified and integrated system.

2. Objectives

- Superior low bit-rate performance
- Continuous-tone and bi-level compression
- Lossless and lossy compression
- Progressive transmission by pixel accuracy and resolution
- Robustness to bit-errors
- Open architecture
- Sequential build-up capability (real time coding)
- Backward compatibility with JPEG
- Content-based description
- Protective image security

3. JPEG2000 Markets and Applications

· Internet, Mobile, and E-Commerce Applications
· Printing, Scanning
· Digital Photography
· Remote Sensing
· Facsimile
· Medical
· Digital Libraries

4. JPEG2000 Features

- High compression efficiency
- Lossless color transformations
- Lossy and lossless coding in one algorithm
- Progressive by resolution and quality
- Static and dynamic Region-of-Interest
- Error resilience
- Multiple component images
- Block and line based transforms
- Compressed image manipulation methods

5. JPEG2000 Requirements:

- Higher compression efficiency than current JPEG
- Backward compatibility with current JPEG
- Progressive coding (by accuracy and by resolution)
- ROI coding (static and dynamic)
- Error resilience capabilities
- Object oriented functionalities (coding, information embedding, …)

Embedded Block Coding with Optimized Truncation (EBCOT):

- Each sub-band is partitioned into a set of blocks
- All blocks within a sub-band have the same size (possible exception for the blocks at the image boundaries)
- Blocks are encoded independently
- Post-processing operation determines the extent to which each block's bit stream should be truncated
- Final bit stream is composed of a collection of "layers"

Why block coding?

- exploit local variations in the statistics of the image from block to block
- provide support for applications requiring random access to the image
- reduce memory consumption in hardware implementations of the compression or decompression engine
- Allow for parallel implementation

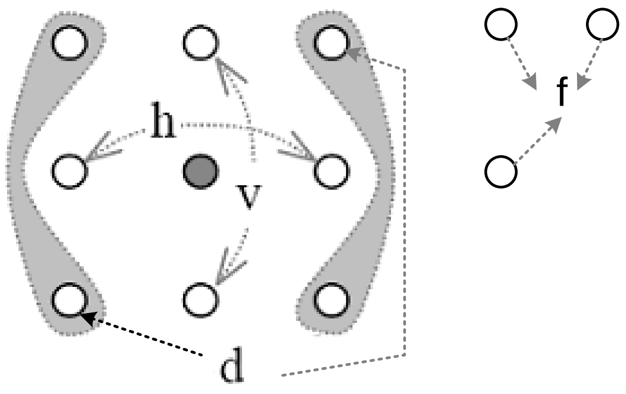
Types of Coding Operations:

- Zero coding (ZC)
- Run-Length coding (RLC)
- Sign coding (SC)
- Magnitude refinement (MR)
  - Arithmetic coding is used

– Reduced complexity in “lazy coding mode”

Zero Coding (ZC):

- Use of 1 of 9 different context states to code the value of the symbol, depending upon the significance state variables of:
  - Immediate horizontal neighbors (h)
  - Immediate vertical neighbors (v)
  - Immediate diagonal neighbors (d)
  - Non-immediate neighbors (f)



Run-Length Coding (RLC):

. used in conjunction with the ZC primitive, in order to reduce the average number of binary symbols which must be encoded using the arithmetic coding engine

. Sign coding (SC)

. used at most once for each sample in the block immediately a previously insignificant symbol is found to be significant during a Zero Coding or Run-Length Coding operation

. Magnitude Refinement (MR)

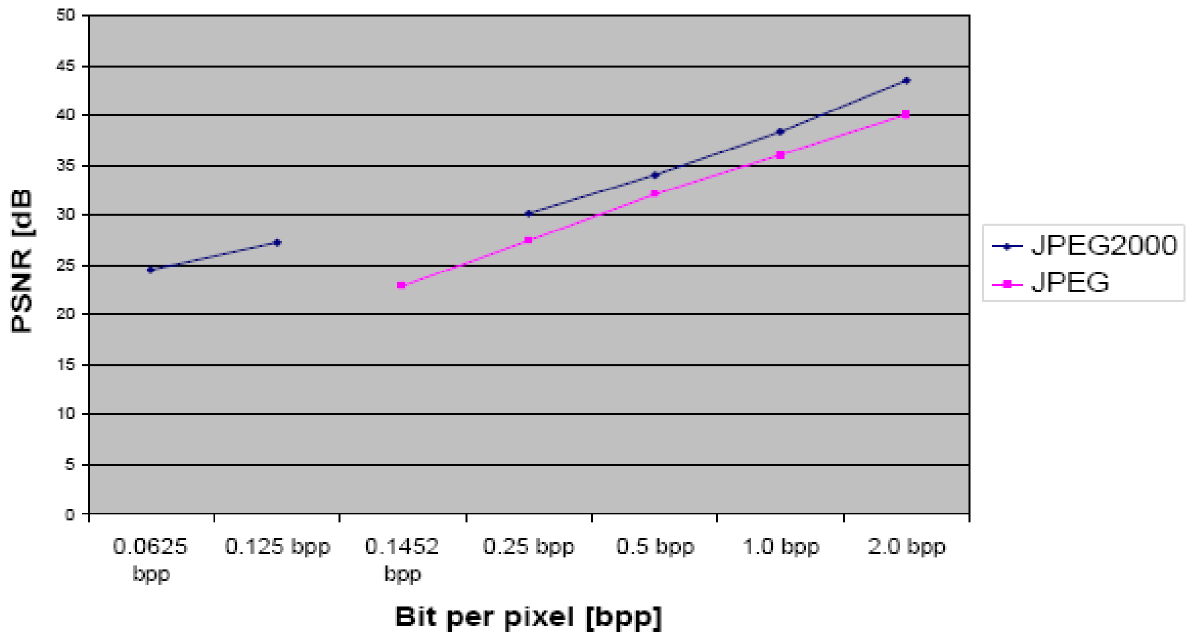
. used to encode an already significant sample

If the sample is non yet significant, a combination of the "Zero Coding" (ZC) and "Run-Length Coding" (RLC) primitives is used to encode whether or not the symbol is significant in the current bit-plane

If so, the "Sign Coding" (SC) primitive must also be invoked to send the sign

• If the sample is already significant, the "Magnitude Refinement" primitive is used to encode the new bit position

JPEG2000 vs JPEG Performance:



Example 8. 4: Explore JPEG2000 compression using VcDemo.