12. STILL IMAGE COMPRESSION TECHNIQUES

Performance criteria in image compression

The aim of image compression is to transform an image into compressed form so that the information content is preserved as much as possible. Two main criteria of measuring the performance of an image compression algorithm are:

- *Compression Efficiency*,
- *Distortion* caused by the compression algorithm.
- *Speed* of the compression and decompression process.

The standard way to measure them is to fix a certain bit rate and then compare the distortion caused by different methods. In on-line applications the waiting times of the user are often critical factors. In the extreme case, a compression algorithm is useless if its processing time causes an intolerable delay in the image processing application. In an image archiving system one can tolerate longer compression times if the compression can be done as a background task. However, fast decompression is usually desired.

Among other interesting features of the compression techniques we may mention the *robustness* against *transmission errors*, and *memory requirements* of the algorithm. The compressed image file is normally an object of a data transmission operation. The transmission is in the simplest form between internal memory and secondary storage but it can as well be between two remote sites via transmission lines. The data transmission systems commonly contain fault tolerant internal data formats so that this property is not always obligatory. The memory requirements are often of secondary importance, however, they may be a crucial factor in hardware implementations. From the practical point of view the last but often not the least feature is *complexity of the algorithm itself*, i.e. *the ease of implementation*. Reliability of the software often highly depends on the complexity of the algorithm. Let us next examine how these criteria can be measured.

Compression efficiency: Most obvious measure of the compression efficiency is the *bit rate*, which gives the average number of bits per stored pixel of the image:

$$bit\ rate = \frac{\text{size of the compressed file}}{\text{pixels in the image}} = \frac{C}{N} \quad \text{(bits per pixel)}$$

where C is the number of bits in the compressed file, and N ($=X \cdot Y$) is the number of pixels in the image. If the bit rate is very low, *compression ratio* might be a more practical measure:

$$Compression\ Ratio = \frac{\text{size of the original file}}{\text{size of the compressed file}} = \frac{N \cdot k}{C}$$

where k is the number of bits per pixel in the original image. The overhead information (header) of the files is ignored here.

Distortion: Distortion measures can be divided into two categories: subjective and objective measures. A distortion measure is said to be *subjective*, if the quality is evaluated by humans. The use of human analysts, however, is quite impractical and therefore rarely used. The weakest point of this method is the subjectivity at the first place. It is impossible to establish a single group of humans (preferably experts in the field) that everyone could consult to get a quality evaluation of their pictures. Moreover, the definition of distortion highly depends on the application, i.e. the best quality evaluation is not always made by people at all.

For objective measures distortion is calculated as the *difference* between the original and the reconstructed image by a predefined function. It is assumed that original image is perfect. All changes are considered as occurrences of distortion, no matter how they appear to a human observer. The quantitative distortion of the reconstructed image is commonly measured by the ***mean absolute error* (MAE), *mean square error* (MSE), and *peak-to-peak signal to noise ratio* (PSNR):**

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - x_i\right|$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - x_i\right)^2$$

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left[255^2/\mathrm{MSE}\right], \qquad \text{assuming } k=8.$$

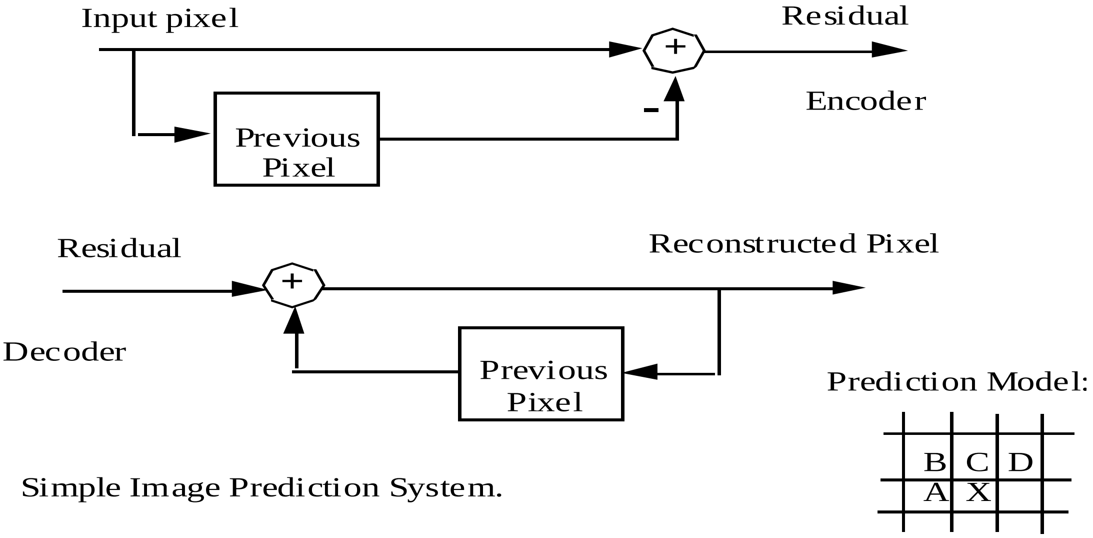
These measures are widely used in the literature. Unfortunately these measures do not always coincide with the evaluations of a human expert. The human eye, for example, does not observe small changes of intensity between individual pixels, but is sensitive to the changes in the average value and contrast in larger regions. Thus, one approach would be to calculate the *mean values* and *variances* of some small regions in the image, and then compare them between the original and the reconstructed image. Another deficiency of these distortion functions is that they measure only local, pixel-by-pixel differences, and do not consider global artifacts, like *blockiness, blurring,* or the *jaggedness of the edges*.

Image Compression Techniques: Image compression techniques can be classified into two broad classes:

- Techniques using 1-D compression methods repeatedly, which include majority of classical image coding systems: lossless and lossy, non-adaptive and adaptive PCM, DPCM, DM, APC, TC.

Examples: Lena 256x256, original, PCM Encoding with 1, 4, 7 bits/pixel without channel errors and subject to 0.5% bit errors.

Simple Lossles Predictive Systems:



Simple Image Prediction System.

- Pixel value at X can be estimated in terms of its causal neighbors in many ways. Simplest is to use the following averaging rule:

$$\hat{X} = Int\{\frac{a+b+c+d}{4})\}$$

- Difference between the pixel and its estimate is coded by one of many coders.

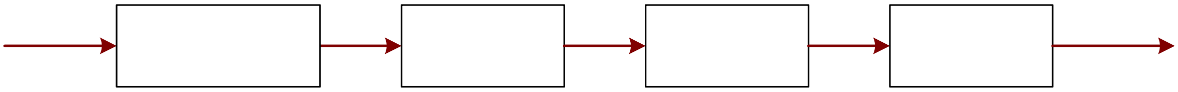
Examples: Lena 256x256, DPCM Encoding with 1, and 4 bits/pixel without channel errors, prediction error (difference image), and subject to 0.5% bit errors.

- Techniques are inherently 2-D: Vector Quantization (VQ), Transform Coded Quantization (TCQ), Sub-band Coding (SBC), JPEG, JPEG2000, MPEG, etc.

Examples: Lena 256x256, VQ Encoding with Coded Bitrate: 0.25 (bpp); PSNR: 24.4 (dB); Coded Bitrate: 0.50 (bpp); PSNR: 28.8 (dB) and Coded Bitrate : 0.75 (bpp); PSNR: 34.8 (dB)

Examples: Lena 256x256, original, Subband Coding with 16-tap filters, 1.0 bpp, PSNR: 33 (dB)

DCT-Based Encoding of Still Images:



Discrete Cosine Transform (DCT) Systems: 1-D *Discrete Cosine transform* (DCT) and its inverse IDCT are defined by:

$$C(u) = \alpha(u)\sum_{x=0}^{N-1} f(x)\cdot\cos\left[\frac{(2x+1)u\pi}{2N}\right] \qquad f(x) = \sum_{u=0}^{N-1}\alpha(u)C(u)\cdot\cos\left[\frac{(2x+1)u\pi}{2N}\right]$$

where

$$\alpha(u) = \begin{cases} \sqrt{1/N} & \text{for} \quad u = 0 \\ \sqrt{2/N} & \text{for} \quad u = 1,2,\ldots,N-1 \end{cases}$$

The corresponding 2-D DCT, and the inverse DCT are defined as

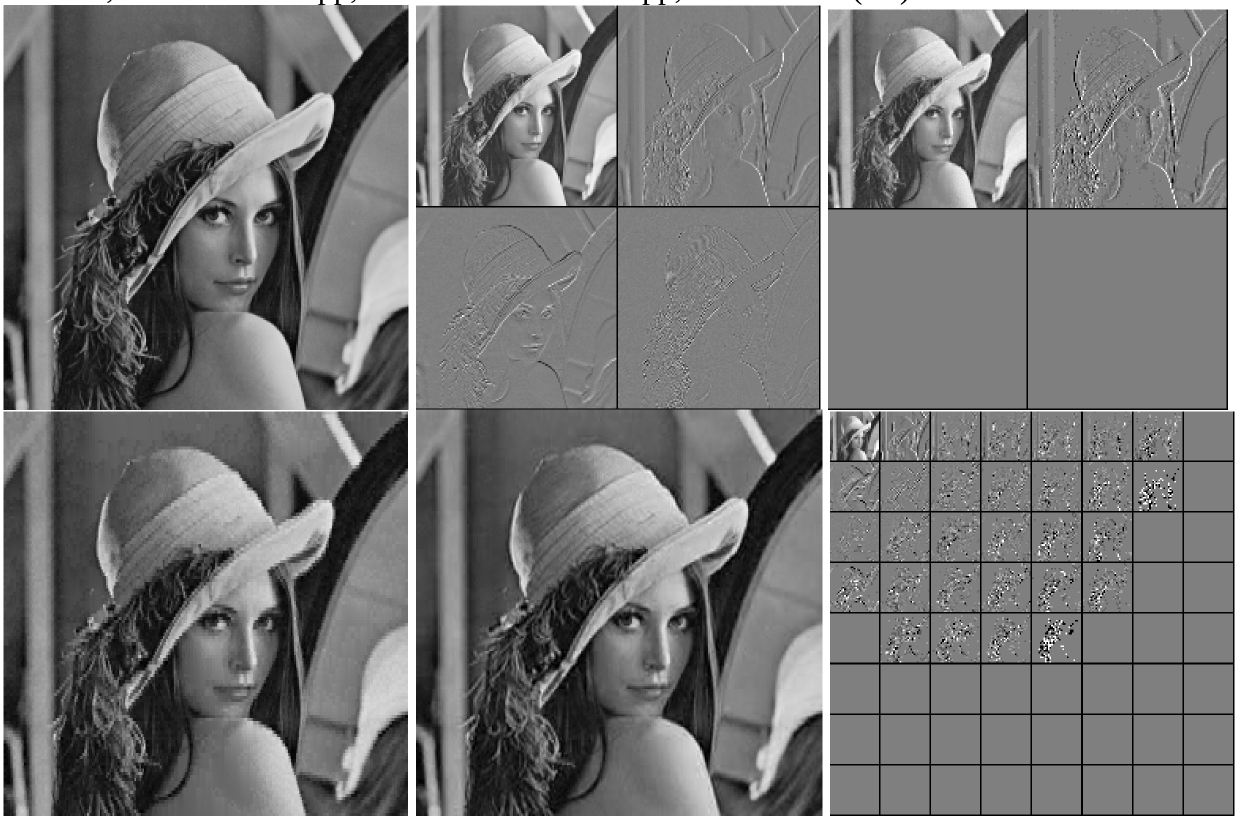
$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cdot\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cdot\cos\left[\frac{(2x+1)v\pi}{2N}\right]$$

and

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)C(u,v)\cdot\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cdot\cos\left[\frac{(2x+1)v\pi}{2N}\right]$$

The advantage of DCT is that it can be expressed without complex numbers. 2-D DCT is also separable (like 2-D Fourier transform), i.e. it can be obtained by two subsequent 1-D DCT in the same way than Fourier transform.

Example: Lena 256x256, 2x2, Set bit rate: 1 bpp; Coded Bitrate: 0.95 (bpp) PSNR: 29.4 (dB); and 8x8, Set bit rate: 1 bpp; Coded Bitrate: 0.98 bpp; PSNR: 32.5 (dB)



Baseline JPEG:

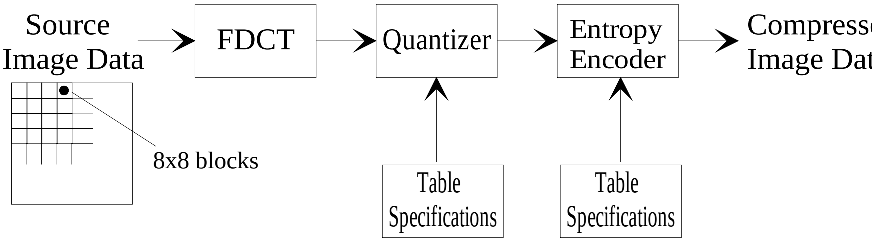
The image is first segmented into 8×8 blocks of pixels, which are then separately coded. Each block is transformed to frequency domain by fast discrete cosine transform (FDCT). The transformed coefficients are quantized and then entropy coded either by arithmetic coder (QM-coder with binary decision tree) or by Huffman coding.

Neither the DCT nor the entropy coding lose any information from the image. DCT only transforms the image into frequency space so that it is easier to compress. The only phase resulting to distortion is the quantization phase. The pixels in the original block are represented by 8-bit integers, but the resulting transform coefficients are 16 bit real numbers, thus the DCT itself would result extension in the file size, if no quantization were performed.

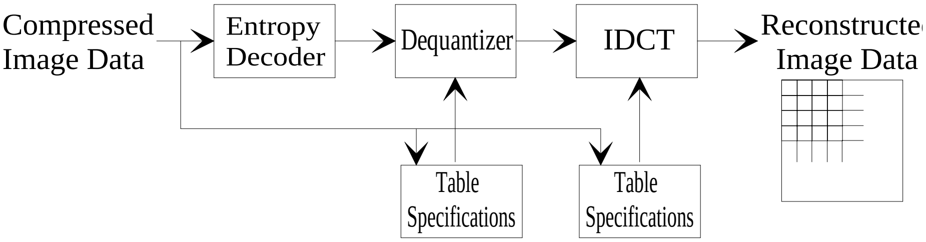
Quantization in JPEG is done by dividing the transform coefficients c_i (real number) by the so-called quantization factor q_i (integer between 1,…,255):

$$\hat{c}_i = round\left(c_i / q_i\right)$$

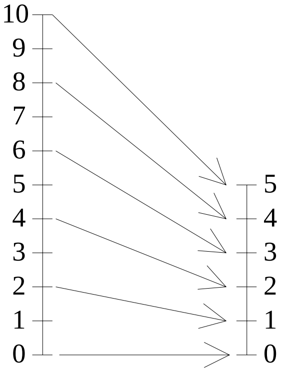
The higher the quantization factor, the less accurate is the representation of the value. Even the lowest quantization factor (q=1) results to a small amount of distortion, since the original coefficients are real number, but the quantized values are integers. The de-quantization is defined by: $r_i = \hat{c}_i \cdot q_i$



Main structure of JPEG encoder.



Main structure of JPEG decoder.



Example of quantization by factor of 2.

In JPEG, the quantization factor is not uniform within the block. Instead, the quantization is performed so that more bits are allocated to the low frequency components (consisting the most important information) than to the high frequency components. See the Table below for an example of possible quantization matrices.

The basic quantization tables of JPEG are also shown below, where the first one is applied both in the gray-scale image compression, and for the Y component in color image compression (assuming YUV, or YIQ color space). The second quantization table is for the chrominance components (U and V in YUV color space). The bit rate of JPEG can be adjusted by scaling the basic quantization tables up (to achieve lower bit rates), or down (to achieve higher bit rates). The relative differences between the q_i factors are retained.

Table: Possible quantization tables

Uniform quantization

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

More accurate quantization

1	2	2	4	4	8	16	16
2	4	8	8	8	16	16	32
4	4	8	16	16	16	32	32
4	8	16	16	16	32	32	32
8	16	16	32	32	32	32	32
8	16	16	32	32	32	64	64
16	16	32	32	32	32	64	64
16	16	32	32	32	64	64	64

Less accurate quantization

8	64	64	128	256	256	256	256
64	128	128	128	256	256	256	256
128	256	256	256	256	256	256	256
256	256	256	256	256	256	256	256
256	256	256	256	256	256	256	256
256	256	256	256	256	256	256	256
256	256	256	256	256	256	256	256
256	256	256	256	256	256	256	256

Table : JPEG quantization tables

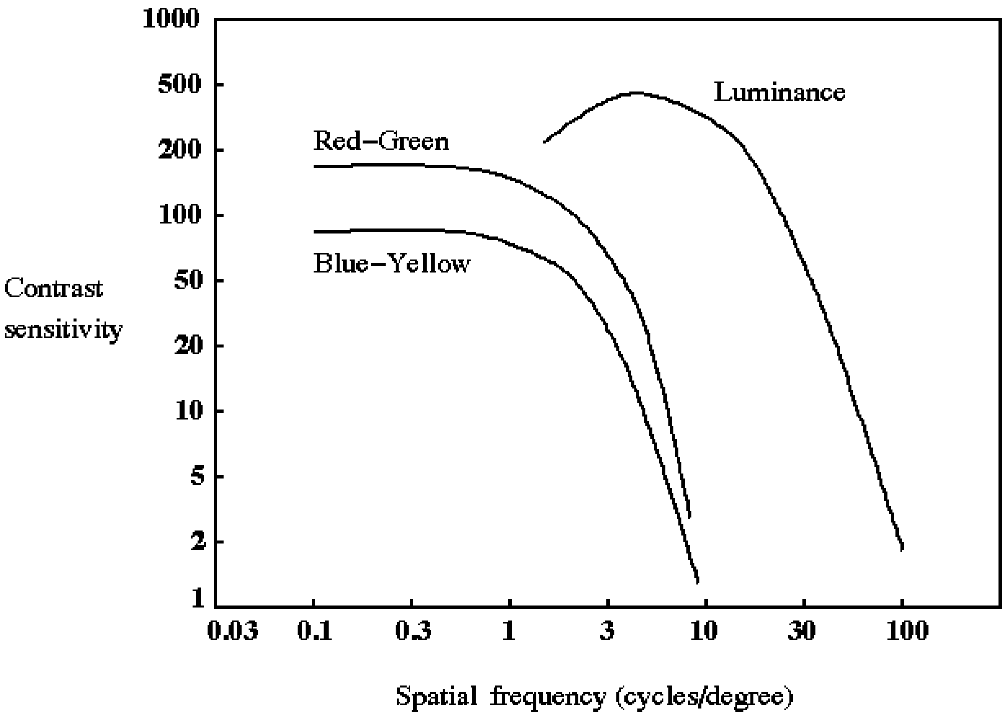
Luminance

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

Chrominance

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

Sensitivity of the Eye to Luminance and Chrominance Variations:



Example: Energy compaction property of DCT demonstrated on 7th frame of Mobile and Calendar Sequence: (anim.gif) Consider one of the 8x8 blocks of this frame:

$$s(n_1, n_2) = \begin{bmatrix} 183 & 160 & 94 & 153 & 194 & 163 & 132 & 165 \\ 183 & 153 & 116 & 176 & 187 & 166 & 130 & 169 \\ 179 & 168 & 171 & 182 & 179 & 170 & 131 & 167 \\ 177 & 177 & 179 & 177 & 179 & 165 & 131 & 167 \\ 178 & 178 & 179 & 176 & 182 & 164 & 130 & 171 \\ 179 & 180 & 180 & 179 & 183 & 169 & 132 & 169 \\ 179 & 179 & 180 & 182 & 183 & 170 & 129 & 173 \\ 180 & 179 & 181 & 179 & 181 & 170 & 130 & 169 \end{bmatrix}$$

DCT coefficients of this frame after subtracting 128 to minimize DC bias.

$$\mathrm{NINT}[S(k_1, k_2)] = \begin{bmatrix} 313 & 56 & -27 & 18 & 78 & -60 & 27 & -27 \\ -38 & -27 & 13 & 44 & 32 & -1 & -24 & -10 \\ -20 & -17 & 10 & 33 & 21 & -6 & -16 & -9 \\ -10 & -8 & 9 & 17 & 9 & -10 & -13 & 1 \\ -6 & 1 & 6 & 4 & -3 & -7 & -5 & 5 \\ 2 & 3 & 0 & -3 & -7 & -4 & 0 & 3 \\ 4 & 4 & -1 & -2 & -9 & 0 & 2 & 4 \\ 3 & 1 & 0 & -4 & -2 & -1 & 3 & 1 \end{bmatrix}$$

where "NINT" denotes nearest integer truncation. It is worth noting that most of the h.f. components (lower right region) are much smaller than those around the (0,0) (upper left corner.)

Bit Allocation / Quantization: It represents which coefficients must be retained and how coarsely each retained coefficient has to quantized via ***zonal*** or ***threshold*** coding.

- **Zonal coding:** Locations of coefficients with the K largest variances are indicated by a zonal mask. Same for all blocks.
- **Bit Assignment:** Suppose a total of B bits/frame is the rate and there are M retained coefficients with variances: $\sigma_i^2; i = 1, \cdots, M$. The number of bits allocated for each coefficient is given by:

$$b_i = \frac{B}{M} + \frac{1}{2}\log_2 \sigma_i^2 - \frac{1}{2M}\sum_{i=1}^{M}\log_2 \sigma_i^2$$

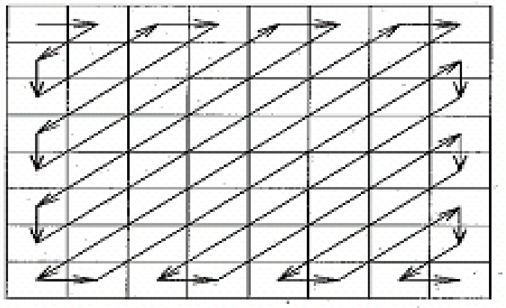
- **Threshold Coding:** It is an adaptive method, where only those coefficients whose magnitudes are above a threshold are retained within each block. This operation is expressed in terms of a Quantization Matrix (QM).

$$\hat{S}(k_1, k_2) = NINT[\frac{S(k_1, k_2)}{T(k_1, k_2)}]$$

where $\hat{S}(k_1, k_2)$ is a limited by a threshold mechanism and quantized approximation of $S(k_1, k_2)$ and $T(k_1, k_2)$ is the corresponding element of the quantization matrix (QM). A coefficient value at location (k_1, k_2) is retained if $\hat{S}(k_1, k_2) \neq 0$. The elements of QM are 8-bit integers which determine the quantization step size for each location. The choice of the QM depends on the source noise level and the viewing conditions. In general, coarser the quantization, larger the weights used for higher-frequency coefficients. A typical QM for luminance components of color imagery is given by:

$$T(k_1,k_2) = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

- Since relative locations of transmitted coefficients vary from block to block, the following zig-zag scan of transform coefficients (except DC) is used to order them in decreasing order. The symbols are then entropy or Huffman coded for transmission or storage.

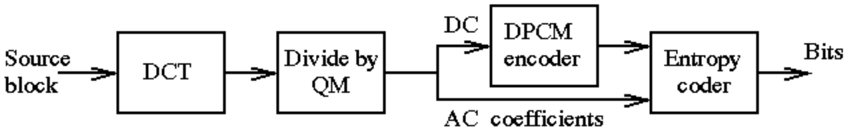


- Application of this quantization matrix $T(k_1,k_2)$ to the above 8x8 DCT coefficients results in:

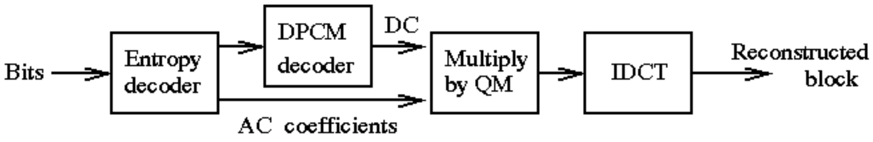
$$\hat{S}(k_1,k_2) = \begin{bmatrix} 20 & 5 & -3 & 1 & 3 & -2 & 1 & 0 \\ -3 & -2 & 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where 45 out of 64 coefficients are truncated to zero.

- DC Coefficients of each block are then encoded by a DPCM coder among themselves.

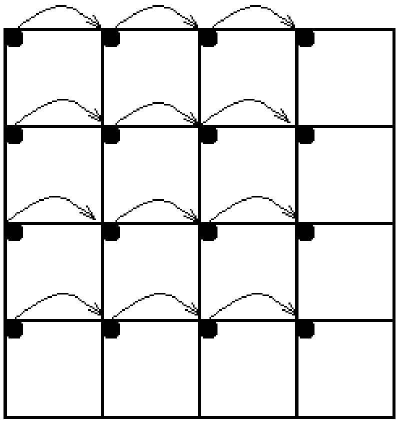


(a)



(b)

DPCM coding of DC coefficients (also referred as DC Prediction): Encoder (a) and decoder (b) of the DCT-based compression system.



Compression artifacts:

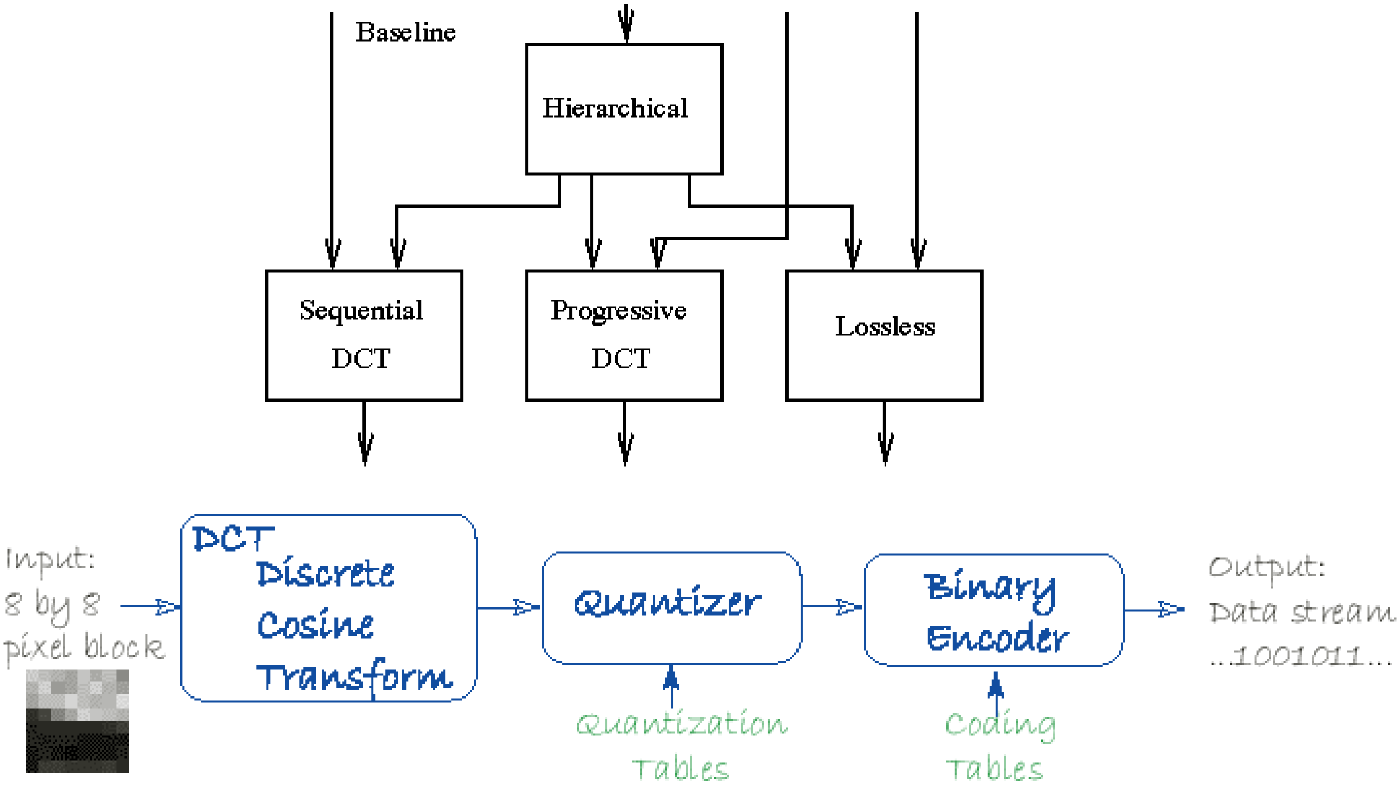
1. Blurring due to truncation of hf terms.
2. Graininess due to coarse coding of some terms, and
3. Blocking artifacts. Overlapping blocks at the encoder (LOT) does the quality.

JPEG Image Compression Standards:

ITU-T G3/G4	Binary images (non-adaptive) - for fax transmission of documents (1980)
ISO JBIG	Binary and bit-plane encoding - to handle progressive transmission & halftones (1994)
ISO JPEG	***Still frame gray scale and color image*** - Block based DCT (1993)
ISO JPEG-LS	New Lossless Coding Standard -Nonlinear prediction, context-based, Rice-Goulomb coding (1997)
ISO JPEG 2000	Wavelet Coding Standard (finalized 2002)

- It is a family of image compression techniques for continuous-tone (gray-scale or color) imagery. It is a lossy technique based on DCT.
- JPEG (Joint Picture Experts Group) provides four modes of operation:
  1. Sequential (baseline).
  2. Hierarchical.
  3. Progressive, and
  4. Lossless.

These options are illustrated with the following flow-diagram.



Standard features:

- **Resolution independence:** Arbitrary image sizes and resolution are ok.
- **Precision:** DCT modes are restricted to 8 & 12 bits/sample; for lossless coding precision could be 2-16 bits/sample.
- **Luminance-Crominance separation:** Handled through the following mapping:

$$Y = 0.3 * R + 0.6 * G + 0.1 * B$$

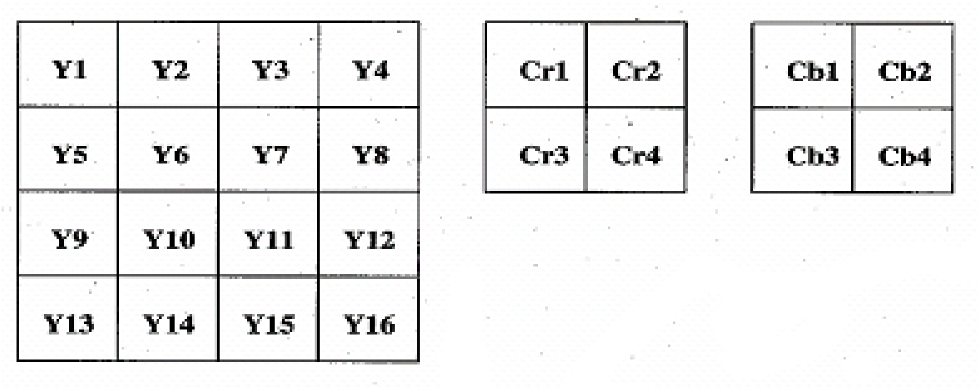
$$C_r = \frac{B - Y}{2} + 0.5$$

$$C_b = \frac{R - Y}{1.6} + 0.5$$

- **Extensible**: No bounds on the number progressive stages, or lower-resolution stages in the case of hierarchical mode.

Baseline Algorithm:

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.



Y1	Y2	Y3	Y4
Y5	Y6	Y7	Y8
Y9	Y10	Y11	Y12
Y13	Y14	Y15	Y16

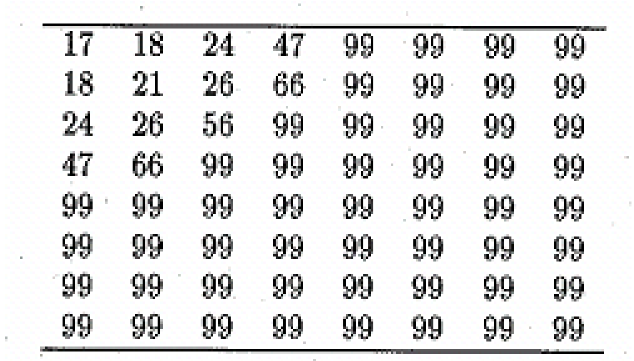
Cr1	Cr2
Cr3	Cr4

Cb1	Cb2
Cb3	Cb4

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 Table for Luminance Channel is given by the matrix QM:



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

Case Study: 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

-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.
- Coding the AC coefficients: Define (RUN,LEVEL) as symbols; e.g., (0,-3); (0,1); (0,-3); ...
- These symbols are VLC (Huffman or arithmetic) coded.
- Decoder 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)

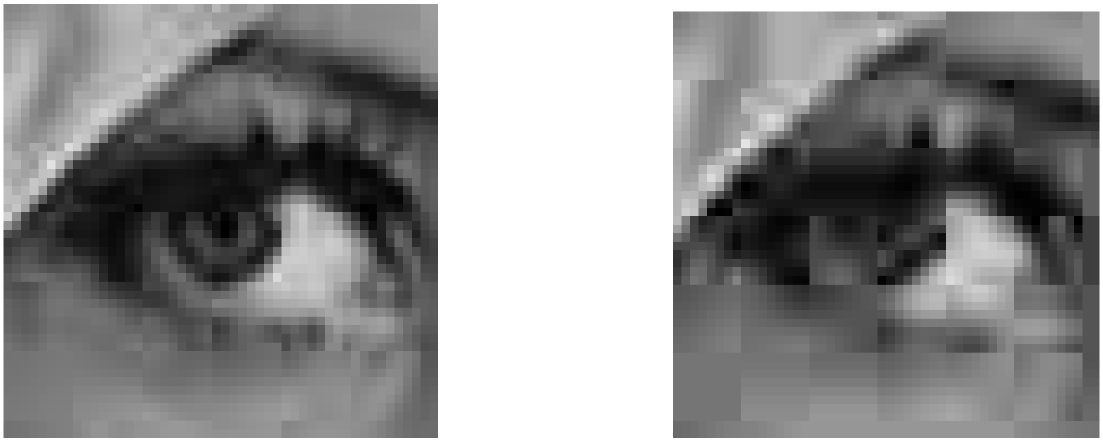
JPEG (bpp = 1.00; mse = 17.26)

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.

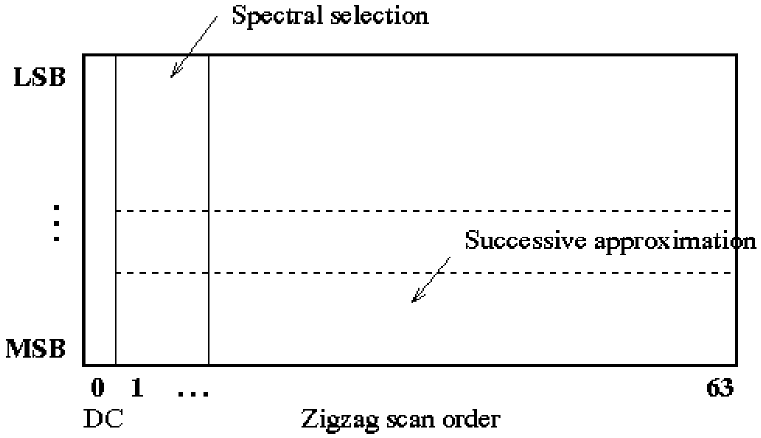
1. VCDemo from Delft University: ..\Chapter12\VcDemo_Software\VCDemo.exe

http://www-ict.its.tudelft.nl/vcdemo.

2. JPEG Color Imagery Compression Demo from Simon Fraser University in Burnaby, B.C., Canada:

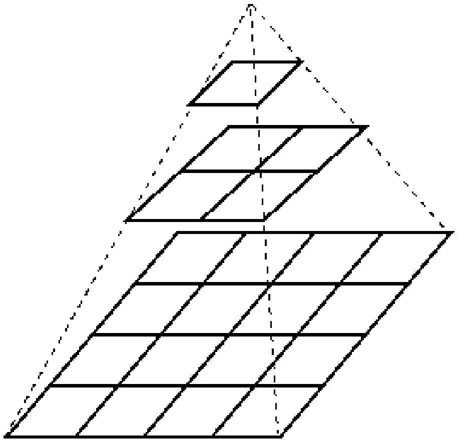
http://www.cs.sfu.ca/CC/365/mark/material/cgi-bin/whichjpeg.cgi

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.

Wavelet Transform and Compression: Basic idea of (discrete) wavelet transform is to decompose the image into:

- Smooth components representing the average color information and obtained by a low-pass filter.
- Detail components represent variations in neighboring pixels is the output of an high pass filter.

Within each family of wavelets (such as the Daubechies family) there are wavelet subclasses distinguished by:

1. Number of coefficients and by
2. Level of iteration.

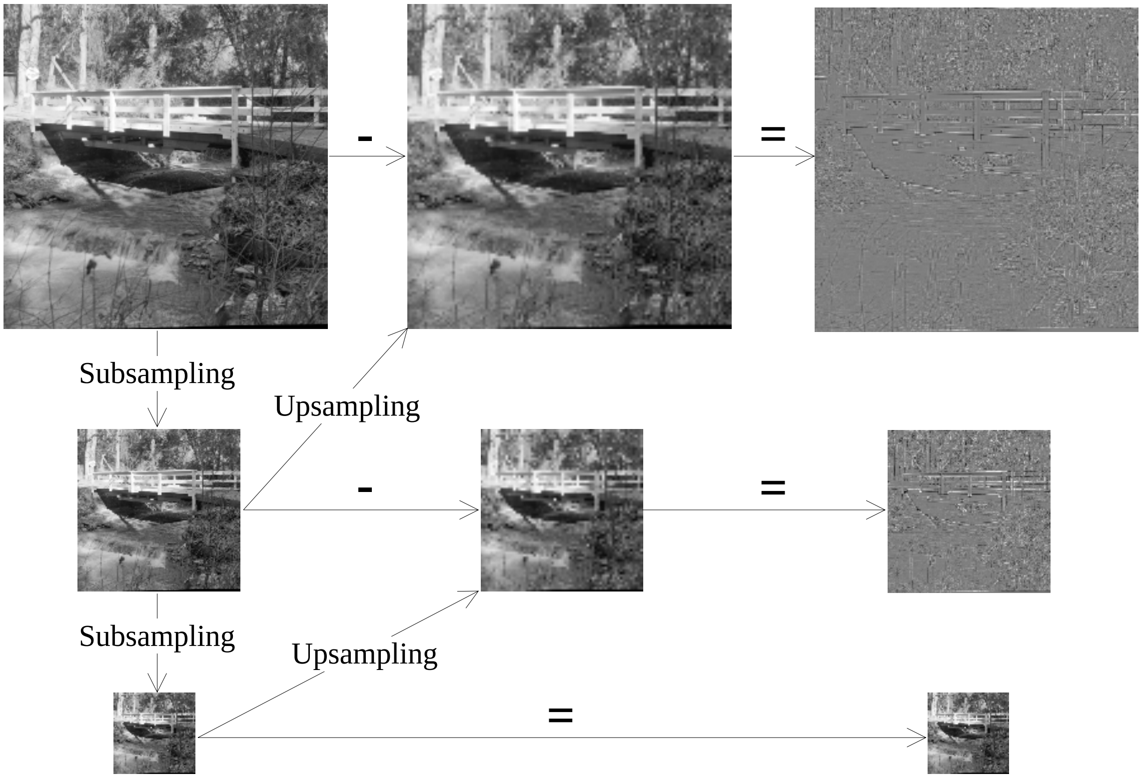
Wavelets are classified within a family most often by the number of vanishing moments. This is an extra set of mathematical relationships for the coefficients that must be satisfied, and is directly related to the number of coefficients. For example, within the Coiflet wavelet family there are Coiflets with two vanishing moments, and Coiflets with three vanishing moments. In the figure below, several different wavelet families are illustrated.

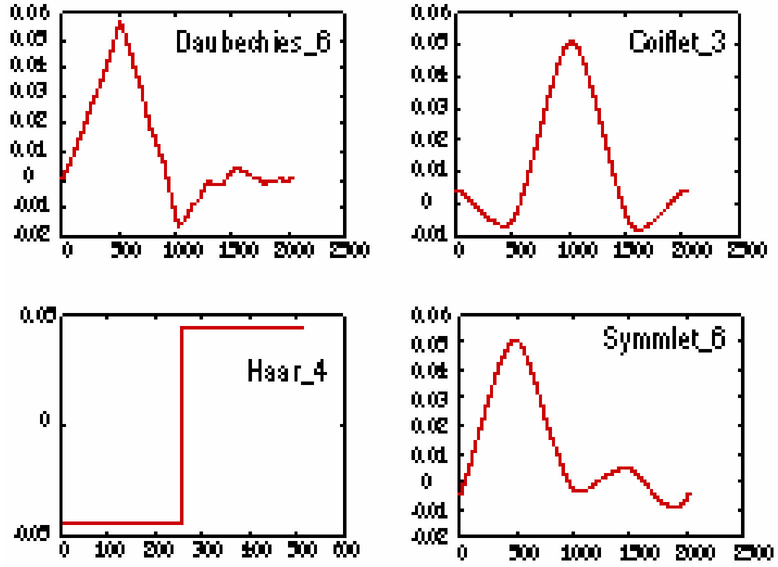
Wavelet decomposition is applied in a hierarchical algorithm, which called a pyramidal algorithm. The wavelet coefficients are arranged so that:

- Odd rows contain an ordering of wavelet coefficients that act as the smoothing filter, and
- Even rows contain an ordering of wavelet coefficient with different signs that act to bring out the data's detail.

The decomposition matrix is first applied to the original, full-length vector. Then the vector is smoothed and decimated by half and the matrix is applied again. Then the smoothed, halved vector is smoothed, and halved again, and the matrix applied once more. This process continues until a trivial number of "smooth-smooth-smooth..." data remain. That is, each matrix application brings out a higher resolution of the data while at the same time smoothing the remaining data.

The output of the DWT consists of the remaining "smooth (etc.)" components, and all of the accumulated "detail" components.





Different families of wavelet functions.

Example of vertical and horizontal sub band decomposition.

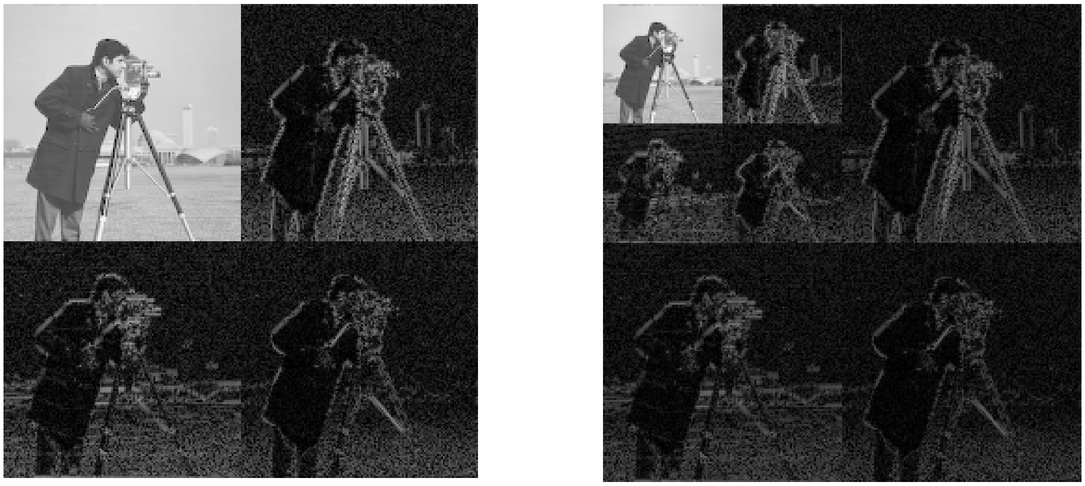


Illustration of the first and second iterations of wavelet decomposition.

Wavelet-based compression: As in the JPEG and other compression systems it consists of the following stages:

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

JPEG2000 The next generation still image compression standard:

- Aim of the JPEG2000 is 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.

- JPEG2000 coding system is intended for low bit-rate applications, exhibiting rate-distortion and subjective image quality performance superior to existing standards.

JPEG2000 Objectives

- Superior low bit-rate performance
- Continuous-tone and bi-level compression
- Lossless and lossy compression
- Progressive transmission by pixel accuracy and resolution
- Fixed-rate, fixed-size, limited workspace memory
- Random codestream access and processing
- Robustness to bit-errors
- Open architecture
- Sequential build-up capability (real time coding)
- Backward compatibility with JPEG
- Content-based description
- Protective image security
- Side channel spatial information (transparency)

JPEG2000 Markets and Applications

- Internet
- Mobile
- Printing
- Scanning
- Digital Photography
- Remote Sensing
- Facsimile
- Medical
- Digital Libraries
- E-Commerce

JPEG2000 Features

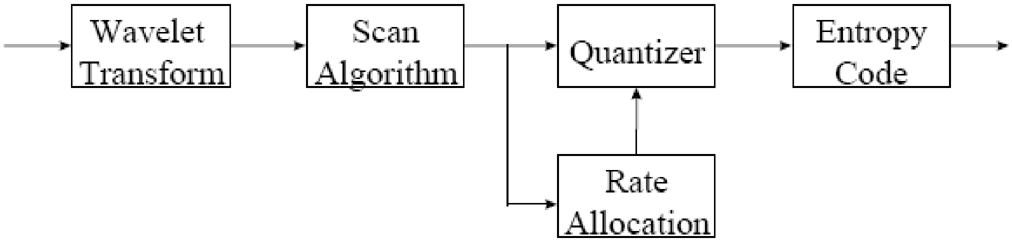
- High compression efficiency
- Lossless colour transformations
- Lossy and lossless coding in one algorithm
- Embedded lossy to lossless coding
- Progressive by resolution and quality
- Static and dynamic Region-of-Interest
- Error resilience
- Visual (fixed and progressive) coding
- Multiple component images
- Block and line based transforms
- Compressed image manipulation methods

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, …)

JPEG2000: Basic encoding scheme:



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?

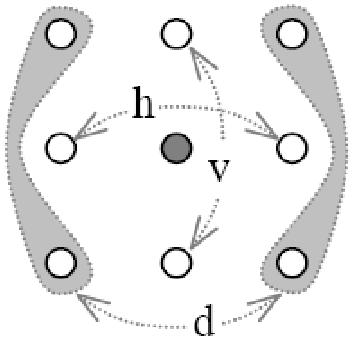
- exploit local variations in the statistics of the image from block to block
- provide support for applications requiring random access to the image
- reduce memory consumption in hardware implementations of the compression or decompression engine
- Allow for parallel implementation

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)

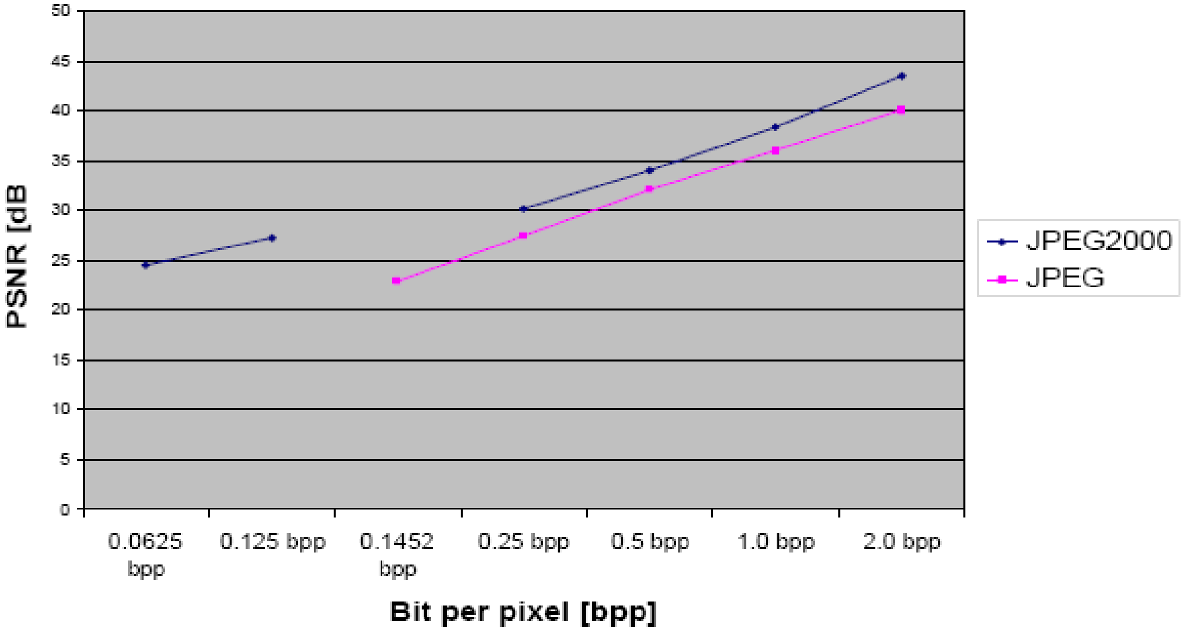
. used to encode an already significant sample

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:



Coding Samples: Village at 0.125 bpp (JPEG, JPEG2000) and 0.25 bpp (JPEG, JPEG2000)

Coding Samples: Hotel at 0.125 bpp (JPEG, JPEG2000) and 0.25 bpp (JPEG, JPEG2000)

ISO / IEC Terminology:

- ISO: International Standardization Organization
- IEC: International Electro-echnical Committee
- ISO/IEC JTC1: Joint Technical Committee
  - SC29: Information Technologies
    - WG1: still images, JPEG and JBIG
  - Joint Photographic Experts Group and Joint Bilevel Image Group
    - WG11: video, MPEG
  - Motion Picture Experts Group
    - WG12: multimedia, MHEG
  - Multimedia Hypermedia Experts Group