JPEG (Joint Photographic Expert Group)

- Baseline Sequential Codec
  - Basis for all of the DCT-based codecs.
  - For images with 8 bits per sample and uses Huffman coding only, source samples in the range $[-2^7, 2^7-1] = [0-2^7, 255-2^7]$.
  - Other DCT sequential codecs were defined to accommodate the two different sample precisions (8 and 12 bits) and two different types of entropy coding (Huffman and arithmetic).
  - The baseline sequential codec is inherently a rich and sophisticated compression method which will be sufficient for many applications.
  - Baseline Version addresses lossy compression.
JPEG

• Procedure:
  1. Segment image into blocks, usually 8×8.
  2. Take DCT of a block.
  3. Divide by a quantization/perceptual weighting matrix.
  4. Round coefficients to nearest integer (this is where quantization takes place).
  5. Code DC coefficients separately using DPCM. The remaining DCT coefficients are referred to as AC coefficients.
  6. Unwrap the block of all quantized DCT coefficients in zig-zag order to form a length 64 (for 8×8 blocks) vector. This helps to facilitate run-length and entropy coding by placing low-frequency coefficients which are more likely to be nonzero before high-frequency coefficients.
7. Form an intermediate representation through run-length coding the unwrapped zig-zag sequence where each non-zero AC coefficients is represented in combination with the run-length (consecutive number) of zero-valued AC coefficients preceding it.

8. Entropy encode the result using Huffman coding.

9. Repeat steps 2 to 8 for each block.
JPEG

**Coding**

\[ x(n_1,n_2) \]

\[ 128 \]

\[ \text{DCT} \rightarrow \text{Weighting matrix} \rightarrow \text{DPCM for DC} \rightarrow \text{Zig-Zag unwrapping} \rightarrow \text{Entropy coder} \rightarrow \text{Transmit} \]

**Decoding**

\[ \text{Compressed bit stream} \]

\[ \text{Inverse Entropy coding} \rightarrow \text{Zig-Zag wrapping} \rightarrow \text{DPCM}^{-1} \text{ for DC} \rightarrow \text{Inverse Weighting} \rightarrow \text{IDCT} \rightarrow x_r(n_1,n_2) \]

128
JPEG

Illustration

Note: DC coefficients do not need to be run-length coded since no preceding zero coefficients.
JPEG

1. An $8 \times 8$ block from Lenna

$$x(n_1, n_2) = \begin{bmatrix}
139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\
144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\
150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\
159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\
159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\
161 & 161 & 161 & 160 & 160 & 157 & 157 & 157 \\
162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\
162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 
\end{bmatrix}$$

2. The JPEG DCT

$$X(K_1, K_2) = \frac{C(K_1)C(K_2)}{4} \sum_{n_1=0}^{7} \sum_{n_2=0}^{7} x(n_1, n_2) \cos\left(\frac{(2n_1+1)K_1\pi}{16}\right) \cos\left(\frac{(2n_2+1)K_2\pi}{16}\right)$$

DC component

<table>
<thead>
<tr>
<th>1260</th>
<th>-1</th>
<th>-12</th>
<th>-5</th>
<th>2</th>
<th>-2</th>
<th>-3</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-23</td>
<td>-17</td>
<td>-6</td>
<td>-3</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>-11</td>
<td>-9</td>
<td>-2</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>-7</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>-3</td>
<td>2</td>
<td>-4</td>
<td>-2</td>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>

High frequency

Only shown is integer part
3. **Psychovisual weighting function**

\[
W(K_1, K_2) = \begin{bmatrix}
16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\
14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\
14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\
18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\
24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\
49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\
72 & 92 & 95 & 98 & 112 & 100 & 103 & 99
\end{bmatrix}
\]

4. **Round to integers**

\[
Y(K_1, K_2) = \text{round}\left( \frac{X(K_1, K_2)}{W(K_1, K_2)} \right) = \begin{bmatrix}
79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

5. **DC coefficient is DPCM coded; AC coefficients are zig-zag unwrapped and coded.**
JPEG

• Baseline JPEG codec uses modified Huffman coding method.
• Huffman coding requires that one or more sets of Huffman code tables be specified by the application.
• The description of Baseline sequential entropy coding is given in two steps:
  1. Conversion of the quantized DCT coefficients into an intermediate sequence of symbols.
  2. Assignment of variable length codes to symbols.
IEEE 508 - JPEG

JPEG

• Intermediate entropy coding representation
  ➢ Each non-zero AC coefficient is represented in combination with the run-length of preceding zero-valued AC coefficients.
  ➢ Each run-length/non-zero coefficient combination is represented by a pair of symbols (symbol 1, symbol 2):
    ✓ symbol 1 represents (RUN-LENGTH, SIZE)
      – RUN-LENGTH represents zero-runs of length 0 to 15. If actual zero-runs greater than 15, value (15,0) used as an extension symbol. Value (0,0) used as End Of Block (EOB) meaning that the remaining AC coefficients are zero.
      – SIZE represents number of bits used to encode AMPLITUDE. Up to 10 bits used to encode AMPLITUDE of quantized AC coefficients.
    ✓ symbol 2 represents the AMPLITUDE of the non-zero coefficient. Quantized AC coefficients are in the range [-2^{10}, 2^{10}-1].
The differential DC coefficient is also represented by a pair of symbols \((\text{symbol 1}, \text{symbol 2})\)

- \text{symbol 1} represents SIZE of the differential DC coefficient. Up to 11 bits used to encode AMPLITUDE of quantized DC coefficients.
- \text{symbol 2} represents AMPLITUDE of the differential DC coefficient. Differential DC coefficients are in the range \([-2^{11}, 2^{11}-1]\).
JPEG

- Size and range of symbol 2 for AC coefficients:

<table>
<thead>
<tr>
<th>SIZE</th>
<th>AMPLITUDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>2</td>
<td>-3..-2, 2..3</td>
</tr>
<tr>
<td>3</td>
<td>-7..-4, 4..7</td>
</tr>
<tr>
<td>4</td>
<td>-15..-8, 8..15</td>
</tr>
<tr>
<td>5</td>
<td>-31..-16, 16..31</td>
</tr>
<tr>
<td>6</td>
<td>-63..-32, 32..63</td>
</tr>
<tr>
<td>7</td>
<td>-127..-64, 64..127</td>
</tr>
<tr>
<td>8</td>
<td>-255..-128, 128..255</td>
</tr>
<tr>
<td>9</td>
<td>-511..-256, 256..511</td>
</tr>
<tr>
<td>10</td>
<td>-1023..-512, 512..1023</td>
</tr>
</tbody>
</table>

Note:
Positive values represented as unsigned magnitude binary numbers.
Negative values represented as the 1’s complement of the corresponding positive values.
Example:
1 is 1 and -1 is 0
3 is 11 and -3 is 00
4 is 100 and -4 is 011

- For DC coefficients one additional level is added to the bottom
JPEG

• Variable-length entropy coding
  - Variable-length codes are assigned to each symbol in \((symbol \ 1, symbol \ 2)\) representation.
  - DC coefficient’s symbols are coded and output first.
  - For both DC and AC coefficients:
    ✓ \(symbol \ 1\) is encoded with a variable-length code (VLC) from Huffman table.
    ✓ \(symbol \ 2\) is encoded with variable-length integer (VLI) code determined by SIZE (not Huffman).
  - The Huffman codes (VLCs) are specified externally as an input to JPEG encoders. They are application-specific. The JPEG standard includes examples, not required ones.
  - The VLI codes are “hardwired” into the standard.
## JPEG

- Huffman table for AC luminance coefficients

<table>
<thead>
<tr>
<th>RUN/SIZE</th>
<th>Length</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/0 (EOB)</td>
<td>4</td>
<td>1010</td>
</tr>
<tr>
<td>0/1</td>
<td>2</td>
<td>00</td>
</tr>
<tr>
<td>0/2</td>
<td>2</td>
<td>01</td>
</tr>
<tr>
<td>0/3</td>
<td>3</td>
<td>100</td>
</tr>
<tr>
<td>0/4</td>
<td>4</td>
<td>1011</td>
</tr>
<tr>
<td>0/5</td>
<td>5</td>
<td>11010</td>
</tr>
<tr>
<td>0/6</td>
<td>7</td>
<td>1111000</td>
</tr>
<tr>
<td>0/7</td>
<td>8</td>
<td>11111000</td>
</tr>
<tr>
<td>0/8</td>
<td>10</td>
<td>111110110</td>
</tr>
<tr>
<td>0/9</td>
<td>16</td>
<td>1111111110000010</td>
</tr>
<tr>
<td>0/A</td>
<td>16</td>
<td>1111111110000011</td>
</tr>
<tr>
<td>1/1</td>
<td>4</td>
<td>1100</td>
</tr>
<tr>
<td>1/2</td>
<td>5</td>
<td>11011</td>
</tr>
<tr>
<td>1/3</td>
<td>7</td>
<td>1111001</td>
</tr>
<tr>
<td>1/4</td>
<td>9</td>
<td>111110110</td>
</tr>
<tr>
<td>1/5</td>
<td>11</td>
<td>1111110110</td>
</tr>
<tr>
<td>1/6</td>
<td>16</td>
<td>111111100000100</td>
</tr>
<tr>
<td>1/7</td>
<td>16</td>
<td>1111111100001100</td>
</tr>
<tr>
<td>1/8</td>
<td>16</td>
<td>1111111110001110</td>
</tr>
<tr>
<td>1/9</td>
<td>16</td>
<td>1111111110001111</td>
</tr>
<tr>
<td>1/A</td>
<td>16</td>
<td>1111111110001000</td>
</tr>
</tbody>
</table>

...
JPEG

- Extended JPEG
  - Arithmetic coding (can adapt to image statistics), typically 5-10% improvement in compression over Huffman.
  - Predictive lossless mode.
  - Progressive mode.
  - Hierarchical mode.
JPEG

- Extended JPEG: DPCM (Predictive) lossless mode:
  - Predictor combines values of three neighboring samples (A, B, C) to form a prediction of the sample X.
  - The difference is encoded losslessly by Huffman or Arithmetic coding algorithm.
  - There are eight different predictors:

<table>
<thead>
<tr>
<th>Selection value</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no prediction</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>A+B-C</td>
</tr>
<tr>
<td>5</td>
<td>A+((B-C)/2)</td>
</tr>
<tr>
<td>6</td>
<td>B+((A-C)/2)</td>
</tr>
<tr>
<td>7</td>
<td>(A+B)/2</td>
</tr>
</tbody>
</table>
JPEG

- Extended JPEG: Progressive Mode
  - Each image component is encoded in multiple scans.
    - The first scan encodes a rough but recognizable version.
    - The succeeding scans refine the image until reaching the level of picture quality established by the quantization tables.
  - Two progressive methods:
    - Spectral selection: Only specified “band” of coefficients from the zig-zag sequence are coded within a given scan.
    - Successive approximation: Upon first encoding, the N most significant coefficients are encoded, where N is specifiable. In subsequent scans the less significant bits are encoded.
JPEG

- Progressive mode

Spectral selection

Successive approximation
JPEG

• Extended JPEG: Hierarchical mode:
  à Provides “pyramidal” encoding of an image at multiple resolutions, each differing in resolution from its adjacent encoding by a factor of two in any or both dimensions.
  à Procedure:
    ✓ Filter and down-sample the original image by multiples of 2.
    ✓ Encode the reduced-size image.
    ✓ Decode the reduced-size image and then interpolate and up-sample it by 2, using the identical interpolation filter which will be used in the receiver.
    ✓ Use the up-sampled image as a prediction of the original, and encode the difference image.
    ✓ Repeat the previous two steps until the full resolution of the image has been encoded.
Three-level hierarchical coder

Encoder

Decoder

JPEG
JPEG

• Color image coding
  ➢ Code each component separately.
  ➢ Use YIQ (NTSC), YUV (PAL/SECAM) or YC_bC_r (CIR60) color spaces.

Model-based image coding

- Not standard
- The image or portion of the image is modeled and the model parameters are used for image synthesis.

**Example:** wireframe model for face

- At Transmitter, model parameters are estimated from analyzing the image and coded

- At Receiver, image estimated from the recovered and decoded model parameters