Practice in Data Compression

Boris Kudryashov

UT (Estonia),
ITMO University (St.-Petersburg, Russia)
Color image compression:

- RGB to YUV conversion
- Decimate chrominance components U,V.
- Compress grayscale images Y, and decimated U,V
- Pack compressed data into the bitstream.
The components are processed by 2D DCT of blocks of size $8 \times 8$ (column-wise 1D transform is followed by row-wise transform). Forward one-dimensional DCT:

$$X(k) = \frac{c(k)}{2} \sum_{n=0}^{7} x(n) \cos \left( \frac{(2n + 1)k\pi}{16} \right), \quad k = 0, 1, \ldots, 7$$

Inverse 1D DCT transform:

$$x(n) = \sum_{k=0}^{7} X(k) \frac{c(k)}{2} \cos \left( \frac{(2n + 1)k\pi}{16} \right), \quad n = 0, 1, \ldots, 7$$

where

$$c(k) = \begin{cases} 
\frac{1}{\sqrt{2}}, & k = 0 \\
1, & k \neq 0 
\end{cases}$$
Quantization examples

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

\[
Q_{UV} = \begin{pmatrix}
17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\
18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\
24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\
\end{pmatrix}.
\]
Let $X$ be a $8 \times 8$ block of grayscale image. 
Quantization:

$$I = X./Q;$$

Reconstruction:

$$\hat{X} = I.*Q;$$

$X(1, 1)$ is called **DC coefficient** all other elements are called **AC coefficients**.

Modern implementations of the JPEG standard allow to quantize all AC coefficients by the same step $\Delta$, say, and recommend to use quantization step equal to 8 for the DC coefficient.
DC coding

Original Y

DC coefficients
Properties of DC coefficients

- DC coefficients are highly correlated
- Large dynamic range: for non-quantized DC
  \[ 0 \leq DC_i \leq 256 \times 64/2/2/(\sqrt{2})^2 = 2048 \]

To exploit correlation differential coding is used (DPCM).

\[ \Delta_i = DC_i - DC_{i-1} \]

under some ordering of DC coefficients. Range (without quantization)

\[ -4096 \leq \Delta_i \leq 4096 \]

If quantization step \( q = 8 \) then the range is still very large.

Coding approach: \((\text{category, amplitude})\). Category \( k \) is encoded by non-uniform (Huffman) code, amplitude is encoded by uniform code of length \( k \). See Homework-3 assignment for details.
### Categories of integer numbers

<table>
<thead>
<tr>
<th>Category</th>
<th>Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<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>
<tr>
<td>11</td>
<td>-2047, ..., -1024, 1024, ..., 2047</td>
</tr>
<tr>
<td>12</td>
<td>-4095, ..., -2048, 2048, ..., 4095</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Each DC coefficient is described by a pair (category, amplitude).
If $\Delta_i \geq 0$ then the amplitude is the binary representation of this difference of length equal to the category.
Otherwise the amplitude is the codeword of the complement binary code for the absolute value of the difference, which has the length equal to the category. The category is then coded by the Huffman coder.
Consider 4 blocks:

\[
\begin{array}{c|c}
B & C \\
\hline
A & X \\
\end{array}
\]

Let DC of A, B, and C, \(DC_A, DC_B, DC_C\) be already quantized to \(QDC_A, QDC_B, QDC_C\) and encoded. Then prediction \(\hat{DC}_X\) for \(DC_X\) is

\[
\hat{DC}_X = \begin{cases} 
QDC_C & \text{if } |QDC_A - QDC_B| \leq |QDC_B - QDC_C| \\
QDC_A & \text{otherwise}
\end{cases}
\]

No information about prediction mode has to be transmitted.
Properties of AC coefficients:

- Range $[-1023, 1023]$ before quantization.
- After quantization most of AC coefficients became zeros.
- The further away from left-up corner, the smaller typical values of AC coefficients.
The *run-length* coder generates a codeword 
\((\text{run-length}, \text{category}), \text{amplitude}\), where *run-length* is the length of zero run followed by the given non-zero coefficient, *amplitude* is the value of this non-zero coefficient and *category* is the number of bits needed to represent the *amplitude*. The pair \((\text{run-length}, \text{category})\) is coded by the 2-D Huffman code and the *amplitude* is coded as in the case of DC coefficients and is added to the codeword.
Two special cases:

- After a non-zero coefficient all other AC coefficients are zero. In this case the special symbol (EOB) is transmitted which codes end-of-block condition.
- A pair (run-length, category) appeared which is not included in the table of the Huffman code. In this case a special codeword called escape-code followed by the uniform codes for run-length and category is transmitted.
Wavelet-based image compression

Motivation:

• Cosine transform provides uniform frequency resolution. It would be better to pay more attention to lower frequencies. Logarithmic frequency scale could be preferable

• Compression without splitting the frame into small blocks could allow avoiding blocking artifacts
Linear transforms and linear filters

The output sequence $y(n)$ of the discrete-time filter with the pulse response $h(n)$ is the convolution of the input sequence $x(n)$ with $h(n)$, that is,

$$y(n) = \sum_{k=0}^{n} h(k)x(n-k) = \sum_{k=0}^{n} h(n-k)x(k).$$

This convolution can be rewritten in matrix form as $y = Tx$ where

$$T = \begin{pmatrix}
h(0) & 0 & 0 & 0 & 0 & \ldots \\
h(1) & h(0) & 0 & 0 & \ldots \\
h(2) & h(1) & h(0) & 0 & \ldots \\
h(3) & h(2) & h(1) & h(0) & \ldots \\
\vdots & h(3) & h(2) & h(1) & \ldots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{pmatrix}.$$
Thus, transform of long sequences can be replaced by filtering. We can use moving average as low-pass filter:

$$y_L(n) = \frac{1}{2}x(n) + \frac{1}{2}x(n - 1).$$

Equivalent transform matrix

$$T = \begin{pmatrix}
\frac{1}{2} & 0 & 0 & 0 & \ldots \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 & \ldots \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & \ldots \\
0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\end{pmatrix}.$$
Moving difference filter

High-pass filter

$$y_H(n) = \frac{1}{2} x(n) - \frac{1}{2} x(n - 1)$$

The corresponding transform matrix has the form

$$T = \begin{pmatrix}
\frac{1}{2} & 0 & 0 & \ldots & 0 \\
-\frac{1}{2} & \frac{1}{2} & 0 & 0 & \ldots \\
0 & -\frac{1}{2} & \frac{1}{2} & 0 & \ldots \\
& \ddots & \ddots & \ddots & \ddots \\
& & & & & \\
& & & & & 
\end{pmatrix}.$$
Notice that entire sequence $x(n)$, $n = 1, 2, ...$ can be reconstructed from decimated sequences $y_L(i)$, $i = 2, 4, ...$ and $y_H(i)$, $i = 2, 4, ...$. In general the following scheme can be used:
Single step of decomposition

with analysis filters

\[ h_0(n) = \frac{x(n) + x(n-1)}{\sqrt{2}}; \quad h_1(n) = \frac{x(n) - x(n-1)}{\sqrt{2}}; \]

and synthesis filters

\[ g_0(n) = \frac{r(n) + r(n-1)}{\sqrt{2}}; \quad g_1(n) = \frac{d(n) - d(n-1)}{\sqrt{2}}; \]

This is example of Haar wavelet decomposition.
2-step decomposition

\[ x(n) \rightarrow h_0(n) \rightarrow \downarrow 2 \rightarrow h_1(n) \rightarrow \downarrow 2 \rightarrow d_1(n) \]

\[ r_1(n) \rightarrow h_0(n) \rightarrow \downarrow 2 \rightarrow r_2(n) \]

\[ h_1(n) \rightarrow \downarrow 2 \rightarrow d_2(n) \]
Synthesis filters

\[ h_0(n) = (-1, 2, 6, 2, -1)\sqrt{32}, \]
\[ h_1(n) = (-2, 4, -2)/\sqrt{32}. \]

Analysis filters

\[ g_0(n) = (2, 4, 2)/\sqrt{32}, \]
\[ g_1(n) = (-1, -2, 6, -2, -1)/\sqrt{32}. \]

For providing perfect-reconstruction finite length transform special “cyclic extension” procedure is applied.
2D wavelet decomposition

\[
\begin{array}{c|c|c|c}
  & LL3 & HL3 &  \\
\hline
LH3 &  &  & HL2 \\
\hline
LH2 & LH3 & HH3 & \\
\hline
LH1 & LH2 & HH2 & HH1 \\
\end{array}
\]
2D wavelet decomposition example
Filtering is interpreted as decomposition over set of shifted pulse responses of two filters. The same filters are applied to large image and smaller images (subbands). Therefore, been expanded to original size, we have a set of embedded bases with different orders of filters, i.e. with different (space) resolutions. This makes discrete wavelets used for image compression analogous to continuous-time wavelet filterbanks.
The quantized highpass subbands usually contain many zeros. They can be efficiently compressed using zero run lengths coding followed by the Huffman coding of pairs (run length, amplitude) or arithmetic coding. (Similarly to AC in JPEG). The lowpass subbands usually do not contain zeros at all or contain small number of zeros and can be encoded by the Huffman code or by the arithmetic code. (Similarly to DC in JPEG).

More advanced coding procedures used, for example, in MPEG-4 standard try to take into account dependencies between subbands. One of such methods is called zero-tree coding.
For each pixel in H subbands there are 4 successors at next level of decomposition. High amplitudes typically related to edges which are repeated at all levels. “Significant” pixels are used as context for next levels.
Applications

- JPEG-2000
- DjVu
- Other applications where high compression ratios and scalability are important.

Limitations:
For JPEG under high compression a blocking artifact can be observed.
There are known such artifacts for wavelets but contours can be blurred.
Summary

- 2D-Wavelet filtering provides non-uniform resolution paying more attention to low-frequency range
- Higher compression ratios than achieved by JPEG-like schemes
- Components (subbands) remain highly correlated. Hierarchical schemes based on using such correlation were developed.