• Lossless vs. lossy compression
• Perceptual models
• Selecting info to eliminate
• Quantization and entropy encoding

**Perceptual Coding**

• Start by evaluating input response of bitstream consumer (eg, human ears or eyes), i.e., how consumer will perceive the input.
  – Frequency range, amplitude sensitivity, color response, ...
  – Masking effects
• Identify information that can be removed from bit stream without perceived effect, e.g.,
  – Sounds outside frequency range, or masked sounds
  – Visual detail below resolution limit (color, spatial detail)
  – Info beyond maximum allowed output bit rate
• Encode remaining information efficiently
  – Use DCT-based transformations
  – Quantize DCT coefficients
  – Entropy code (eg, Huffman encoding) results

**Perceptual Coding Example: Images**

• Characteristics of our visual system ⇒ opportunities to remove information from the bit stream
  – More sensitive to changes in luminance than color ⇒ spend more bits on luminance than color (encode separately)
  – More sensitive to large changes in intensity (edges) than small changes ⇒ quantize intensity values
  – Less sensitive to changes in intensity at higher spatial frequencies ⇒ use larger quanta at higher spatial frequencies
• So to perceptually encode image, we would need:
  – Intensity at different spatial frequencies
  – Luminance (grey scale intensity) separate from color intensity

**Lossless vs. Lossy Compression**

- Huffman and LZW encodings are *lossless*, i.e., we can reconstruct the original bit stream exactly: $\text{bits}_{\text{OUT}} = \text{bits}_{\text{IN}}$.
  – What we want for “naturally digital” bit streams (documents, messages, datasets, …)
- Any use for *lossy* encodings: $\text{bits}_{\text{OUT}} \neq \text{bits}_{\text{IN}}$?
  – “Essential” information preserved
  – Appropriate for sampled bit streams (audio, video) intended for human consumption via imperfect sensors (ears, eyes).
JPEG Image Compression

JPEG = Joint Photographic Experts Group

JPEG-YCbCr (601) from "digital 8-bit RGB"

\[ Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \]
\[ Cb = 128 - 0.168736 \times R - 0.331264 \times G + 0.5 \times B \]
\[ Cr = 128 + 0.5 \times R - 0.418688 \times G - 0.081312 \times B \]

All values are in the range 0 to 255

2D Discrete Cosine Transform (DCT2)

\[ X_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{mn} \cos \left( \frac{\pi}{M} (m + \frac{1}{2}) p \right) \cos \left( \frac{\pi}{N} (n + \frac{1}{2}) q \right) \]

\[ \alpha_p = \begin{cases} 1/\sqrt{M} & p = 0 \\ \sqrt{2/M} & 1 \leq p \leq M - 1 \end{cases} \]
\[ \alpha_q = \begin{cases} 1/\sqrt{N} & q = 0 \\ \sqrt{2/N} & 1 \leq q \leq N - 1 \end{cases} \]

\[ X_k = \sum_{n=0}^{N-1} x_n \cos \left( \frac{\pi}{N} (n + \frac{1}{2}) k \right) \]

1D DCT (Type 2)

DC Component

2D DCT Basis Functions

http://en.wikipedia.org/wiki/YCbCr

http://www.cs.cf.ac.uk/Dave/Multimedia/node231.html
Quantization (the "lossy" part)

Divide each of the 64 DCT coefficients by the appropriate quantizer value ($Q_{\text{lum}}$ for Y, $Q_{\text{chr}}$ for Cb and Cr) and round to nearest integer $\Rightarrow$ many 0 values, many of the rest are small integers.

Note fewer quantization levels in $Q_{\text{chr}}$ and at higher spatial frequencies. Change "quality" by choosing different quantization matrices.

Quantization Example

<table>
<thead>
<tr>
<th>239</th>
<th>32</th>
<th>27</th>
<th>-12</th>
<th>3</th>
<th>-5</th>
<th>3</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>-3</td>
<td>-19</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>-70</td>
<td>2</td>
<td>8</td>
<td>23</td>
<td>9</td>
<td>6</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>5</td>
<td>-6</td>
<td>11</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-17</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>-3</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

DCT Coefficients

Visit coeffs in order of increasing spatial frequency $\Rightarrow$ tends to create long runs of 0s towards end of list:

15 3 3 5 0 3 -1 -1 0 0 -1 0 0 0 0 0 1 0...

Entropy Encoding

Use differential encoding for first coefficient (DC value) $\Rightarrow$ encode difference from DC coeff of previous block.
Encode DC coeff as (N), coeff

N is Huffman encoded, differential coeff is an N-bit string

<table>
<thead>
<tr>
<th>DC Coef Difference</th>
<th>Size</th>
<th>Typical Huffman codes for Size</th>
<th>Additional Bits (in binary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>00</td>
<td>-</td>
</tr>
<tr>
<td>-1,1</td>
<td>1</td>
<td>010</td>
<td>0,1</td>
</tr>
<tr>
<td>-2, -2,2,3</td>
<td>2</td>
<td>011</td>
<td>00,01,10,11</td>
</tr>
<tr>
<td>-7, -8, -8, -7, -7</td>
<td>3</td>
<td>100</td>
<td>000, 001, 100, 110</td>
</tr>
<tr>
<td>-15, -16, -16, -16, -15</td>
<td>4</td>
<td>101</td>
<td>0000, 0011, 1000, 1100, 1111</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(run,N)</td>
<td>(0,0) is EOB meaning remaining coeffs are 0</td>
</tr>
</tbody>
</table>

Encode AC coeffs as (run,N),coeff

Run = length of run of zeros preceding coefficient
N = number of bits of coefficient
Coeff = N-bit representation for coefficient

(run,N) pair is Huffman coded
(0,0) is EOB meaning remaining coeffs are 0
(15,0) is ZRL meaning run of 16 zeros

Entropy Encoding Example

Quantized coeffs:

15 3 3 5 0 3 -1 -1 0 0 -1 0 0 0 0 0 0 0 1 0...

DC: (N),coeff, all the rest: (run,N),coeff

(4) 15 (0,2) 3 (0,2) 3 (0,3) 5 (1,2) 3 (0,1) -1
(0,1) -1 (2,1) -1 (6,1) 1 EOB

Encode using Huffman codes for N and (run,N):

To read more see "The JPEG Still Picture Compression Standard" by Gregory K. Wallace
http://white.stanford.edu/~brian/psy221/reader/Wallace.JPEG.pdf

JPEG Results

The source image (left) was converted to JPEG (q=50) and then compared, pixel-by-pixel. The error is shown in the right-hand image (darker = larger error).