**Example: Digitizing Analog Video**

- Baseband signal has 5 MHz maximum frequency
  - Remember: starting point is a lousy analog signal
  - Nyquist sampling rate is 10 Msamples/sec
- Let’s assume 8-level quantization
  - Requires 8 bits/sample
  - Visible SNR of 48 dB – pretty good picture
- Requires uncompressed bit rate of 80 Mbits/sec
  - Way too fast for many wired connections
  - Signal is still poor analog video *plus* quantization noise
**Lossless Data Compression**

- Representing a common and/or recurring combination of bits with reduced alphabet
- Usually reduces a digital data set to a smaller number of bits
  - No loss of real data (perfect reconstruction)
  - Likely combinations of bits get short bit sequences
  - Unlikely combinations get long bit sequences
- Examples of lossless image compression include
  - Huffman encoding, Lempel-Ziv
  - PCX, GIF, LZW, ZIP, PNG

**Example of Lossless Compression**

- Test Sentence (101 characters)
  - I am really excited that the Georgia Tech Yellow Jackets have a chance to beat the dogs in November.
- Swap the following representations
  - “zz” = “z”; “za” = “the_”; “zb” = “that_”
  - “zc” = “ed_”; “zd” = “_in_”; “ze” = “have_”;
  - “zf” = “_am_”; “zg” = “ed.”; “zh” = “er.”;
- New Sentence (82 characters)
  - zfreally excitzczbzaGeorgia Tech Yellow Jackets zea chance to beat zadogszdNovembzh
Challenge of Lossless Data Compression

- Most video and images have patterns that can be exploited for lossless data compression
- Different images have different "pattern sets"
- How do we identify the patterns
  - Option 1: Assume a priori pattern structure
  - Option 2: Ad-hoc patterns, build-as-you-go

Run Length Encoding

- Compression used in PCX, Fax transmissions
- Works well on "cartoonish" images
- Basic Idea: Assign repetitive data sequences with low-bit possibilities in the alphabet
- PCX Image
  - Simple form of Run-Length Encoding
  - Repetitive colors (3 or more) are recorded as "[2 flag bits + 6 repetition bits] [color value]"
  - Possible to have "compressed" file larger than original
  - Easy to compute; early adoption in computer image use
**Huffman Coding**

<table>
<thead>
<tr>
<th>Char</th>
<th>Freq</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>space</td>
<td>7</td>
<td>111</td>
</tr>
<tr>
<td>a</td>
<td>4</td>
<td>010</td>
</tr>
<tr>
<td>e</td>
<td>4</td>
<td>000</td>
</tr>
<tr>
<td>f</td>
<td>3</td>
<td>1101</td>
</tr>
<tr>
<td>h</td>
<td>2</td>
<td>1010</td>
</tr>
<tr>
<td>i</td>
<td>2</td>
<td>1000</td>
</tr>
<tr>
<td>m</td>
<td>2</td>
<td>011</td>
</tr>
<tr>
<td>n</td>
<td>2</td>
<td>0010</td>
</tr>
<tr>
<td>s</td>
<td>2</td>
<td>1011</td>
</tr>
<tr>
<td>t</td>
<td>2</td>
<td>0110</td>
</tr>
<tr>
<td>l</td>
<td>1</td>
<td>11001</td>
</tr>
<tr>
<td>o</td>
<td>1</td>
<td>00110</td>
</tr>
<tr>
<td>p</td>
<td>1</td>
<td>10011</td>
</tr>
<tr>
<td>r</td>
<td>1</td>
<td>11000</td>
</tr>
<tr>
<td>u</td>
<td>1</td>
<td>00111</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
<td>10010</td>
</tr>
</tbody>
</table>

**Lempel-Ziv Algorithm (LZ77 and LZ78)**

- Build a pattern alphabet as you go
  - Basic adaptive algorithm
  - Very easy to implement with minimal memory
  - Incorporated into types of image & video data
- Used in famous DEFLATE compression algorithm (i.e. ZIP and other archival tools)
- Lossless algorithm
- Near-optimum for very long data streams
Lempel-Ziv Algorithm

- Starting data sequence
  - “those lame dogs lose the games”
- Parse data into the smallest non-repeatable chunks
  - (1) t, (2) h, (3) o, (4) s, (5) e, (6) _, (7) l, (8) a, (9) m, (10) e_, (11) d, (12) og, (13) s_, (14) lo, (15) se, (16) _t, (17) he, (18) _g, (19) am, (20) es
- Final coding: each chunk is written as previous portion reference # plus its last character:
  - [0t][0h][0o][0s][0e][0_][0l][0a][0m][5_][0d][3g][4_][7o][4e][6t][2e][6g][8m][5s]

Lempel-Ziv Algorithm

- Starting data sequence (73 bits)
  - 0101010101010101010101010101010101010101010101010101010101010100
- Parse data into the smallest non-repeatable chunks
  - (1) 0, (2) 1, (3) 01, (4) 010, (5) 10, (6) 101, (7) 0101, (8) 01010, (9) 1010, (10) 10101, (11) 010101, (12) 0101010, (13) 101010, (14) 1010101, (15) 01010101, (16) 010101010
- Final bit sequence (64 bits)
  - [00][01][11][10][100][1101][1001][1110][01100][10011][01111] [1][10110][10100][11011][11001][11110]
Final Comment

- Sampling, Quantization, and Compression are not independent designs in communication links
  - Optimum solution is to do these together
  - Example: Vocoder on cell phone
    - Compression & quantization optimized together for speech
    - Sounds terrible if you try to listen to a musical song
  - Audio signal requires
    - 40 kSample/sec sampling (20 kHz max frequency)
    - 60 dB of SNR for fidelity (10 bits/sample for uniform quant.)
    - Total of 400 kbit/sec for high-quality, uncompressed voice
- Typical cellular vocoder works at 8 kbit/sec

JPEG -- Lossy Image Compression

- Joint Photographic Experts Group (1992)
- Loses original image information without the possibility of reconstruction
- Converts image from RGB to YCbCr.
  - Chrominance is downsampled
  - Each channel is converted to 2D freq domain
  - Only keep the most significant freq components
  - Add run length encoding (RLE) to reduce size
- JPEG 2000 Standard uses wavelet-based compression instead of discrete cosine transform
**Image Discrete Cosine Transform**

- Each block of 64 pixels is expressed as a linear combination of the 64 tiles shown on the right.
- Compression level is based on which coefficients are thrown away (from lower-right to upper-left).
- Explains JPEG/MPEG errors result in "blockish" errors in image frames.

**Variable-Rate JPEG Compression**

- 42 kB
- 23 kB
- 13 kB
- 9 kB
- all pics
### Example Compression of Two Pictures

![Simpsons](image1.png) ![Kauai](image2.png)

### Comparison of Compression Sizes

<table>
<thead>
<tr>
<th>Storage</th>
<th>Compression</th>
<th>Simpsons</th>
<th>Kauai</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP</td>
<td>None</td>
<td>341 KB</td>
<td>3,073 KB</td>
</tr>
<tr>
<td>PCX</td>
<td>RLE</td>
<td>171 KB</td>
<td>2,732 KB</td>
</tr>
<tr>
<td>GIF</td>
<td>RLE + LZ</td>
<td>142 KB</td>
<td>374 KB</td>
</tr>
<tr>
<td>JPG</td>
<td>Lossy</td>
<td>75 KB</td>
<td>142 KB</td>
</tr>
</tbody>
</table>

- RLE works well on “cartoonish” figures, not on photos
- LZ algorithms dramatically improve photos
- JPG is always best (but lossy)
- Interesting that both photos reduce to similar order-of-magnitude sizes in JPG (they start off vastly different)
Digital Video Compression

- Redundancy in moving pictures from frame to frame
- Best video compression algorithms are “3D”, taking advantage of
  - Patterns in single images
  - Frame-to-frame patterns
  - “Motion Vectors” within a scene
- Trade-off: the more redundancy you remove, the more catastrophic bit errors become

So Let’s Turn Analog Video To Digital

- Specs for encoding analog video and audio
  - Included “MPEG Layer 3 Audio” spec or MP3
  - Video based on digitizing analog TV signal
- Applications that use MPEG1
  - Archiving analog video footage on digital media
  - Video CD and MP3 video recordings
  - Unsuitable for high-definition digital television
**MPEG1 Video Standard Facts**

- MPEG-1 supports resolutions up to 4095×4095 (12-bits), and bitrates up to 100 Mbit/s
- Most commonly used at 352x240, 352x288, or 320x240 with bitrates less than 1.5 Mbit/s
- Splits image into YCbCr streams and subsamples the color signals (just like JPEG)
- Applies “3D” compression (2D image and temporal changes)

**MPEG2**

- Meant for higher-definition transmissions
- Capable of HDTV-type transmissions
- Applications
  - Video stream on DVDs
    - Peak data rate of 10.08 Mbit/s
    - Maximum resolution of 720 × 480 pixels
  - Digital Video Broadcast (DVB) in Europe
  - Advanced Television Systems Committee (ATSC) in North America
MPEG2 Applications

- Video stream on DVDs
  - Peak data rate of 10.08 Mbit/s
  - Maximum resolution of 720 × 480 pixels
  - YUV video information with subsampling
- Digital Video Broadcast (DVB) in Europe
- Advanced Television Systems Committee (ATSC) in North America
  - Broadcast television data rate 19.4 Mbit/s
  - YCbCr video information with subsampling
- Digital Satellite Broadcasts (most common)

YUV Color Scheme

Similar to NTSC analog YIQ scheme (image from Wikipedia commons)
MPEG 4 Specifications

- Introduced in 1998
- Based on MPEG1 and MPEG2 standards
- Added copyrighting, 3D features, interactivity
- Improved coding efficiency for video, audio, speech
- Improved error resilience

Summary

- Three baseband ops for digital representation
  - Sampling
  - Quantizing
  - Compression (lossless and lossy)
- In video, MPEG-X standards specify how to perform these operations (audio specs as well).
- Still have not transmitted the data across a radio channel