Key Concepts

- Lossless vs. lossy compression
- Kolmogorov complexity
- Predictive coding / Huffman coding
- JPEG / Discrete cosine transform (DCT)
- JPEG2000 / Wavelets
Image and Video Data Rates

Image
- $640 \times 480 \times 24b = \sim 3/4$ MB

Full screen Image
- $1024 \times 768 \times 24b = \sim 2.5$MB

DVD
- $720 \times 480 \times 24b \times 30f/s = \sim 30$ MB/s

High Definition DVD
- $1920 \times 1080 \times 24b \times 30f/s = \sim 178$MB/s

Film
- $4000 \times 3000 \times 36b \times 30f/s = \sim 1.5$GB/s
- 8 TB for one 90 minute movie!

Lossless vs. Lossy Compression

Lossless
- All information stored
- Exact original can be reconstructed

Lossy
- Some information discarded
- Goal: discard information humans won’t notice
- Much higher compression ratios possible
Kolmogorov Complexity

What is the shortest program that can generate the data?

17 KB JPEG

\[ x + iy, \text{ } \text{ xmin, xmax, ymin, ymax } \] (24B)

Run Length Encoding

\[ BWBBB\ldots BW \]

\[ \downarrow \]

\[ BW\{12\}B\{6\}W\{3\}BW \]
Example

Alphabet: A, B, C, D
Frequencies: $\frac{1}{4}$, $\frac{1}{4}$, $\frac{1}{4}$, $\frac{1}{4}$
Code: 00, 01, 10, 11

ABCDBDAC = 8 characters
0 0 1 0 1 1 0 1 1 1 0 1 1 0 0 1 1 = 16 bits

$\frac{1}{4} \cdot 2 + \frac{1}{4} \cdot 2 + \frac{1}{4} \cdot 2 + \frac{1}{4} \cdot 2 = 2$ bits/ch on average

Example

Alphabet: A, B, C, D
Frequencies: $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{8}$
Code: 0, 10, 110, 111

ABACADAB = 8 characters
0 1 0 0 1 1 0 0 1 1 1 0 1 0 = 14 bits

$\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1 \frac{3}{4}$ bits/ch on average
Entropy

\[ H(X) = \sum_{i=1}^{n} p(x_i) \log_2 \left( \frac{1}{p(x_i)} \right) \]

\[ = - \sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \]

Huffman Coding

A (.10)
B (.15)
C (.30)
D (.16)
E (.29)
Huffman Coding

A
(.10)

B
(.15)

C
(.30)

D
(.16)

E
(.29)

AB
(.25)

0
1

0
1

ABD
(.41)

AB
(.25)

A
(.10)

B
(.15)

C
(.30)

D
(.16)

E
(.29)
Huffman Coding

Huffman Coding

Huffman Coding
Lossy Compression

Chroma subsampling

Transform Coding
- Fourier / DCT (JPEG)
- Wavelets (JPEG2000)

Chroma Subsampling

[Diagram showing chroma subsampling]
Transform Coding

\[ X = 2 2 2 2 3 4 5 6 6 6 \quad H(X) = 2.0049 \]

\[ D(X) = 2 0 0 0 1 1 1 0 0 0 \quad H(D(X)) = 1.3222 \]

Bases

Basis vectors \( b_0, b_1, \ldots, b_n \)

Express any vector as

\[ a_0 \times b_0 + a_1 \times b_1 + \ldots + a_n \times b_n \]

where the coefficients \( a_i \) are scalars.
Pixel Basis

Another Basis
Two Observations on Images

Characterize images by the frequencies that are present

The frequencies in natural images fall off as $1/f$
  That is, high frequencies are less common

The human visual system is less sensitive to high frequencies
  That is, it is more important to preserve low frequencies than high frequencies

Discrete Cosine Transform (DCT)

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

262,144 pixels

43384 largest terms, 16%
(dropped 218,760 terms)
DCT

262,144 pixels
8353 largest terms, 3.2%
(dropped 253,791 terms)

Quantization

\[
\begin{bmatrix}
-415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\
4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\
-47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\
-49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\
12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\
-8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\
-1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\
0 & 0 & -1 & -4 & -1 & 0 & 1 & 2
\end{bmatrix}
\]

\[
\begin{bmatrix}
16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\
14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\
14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\
18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\
24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\
49 & 64 & 73 & 87 & 103 & 121 & 120 & 101 \\
72 & 92 & 95 & 98 & 112 & 100 & 103 & 99
\end{bmatrix}
\]

\[
DCT \text{ Image} \div \begin{bmatrix}
-26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\
0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\
-3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\
-4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

= Quantized DCT Image
Storage Order

Error

error = abs(original – compressed) * 8
Wavelets

### Haar Wavelets

Scaling function = average

Transform:
- \( S = (A + B)/2 \)
- \( D = (A - B)/2 \)

Invert:
- \( A = S + D \)
- \( B = S - D \)
Haar Wavelets

6 8 5 9 5 5 6 6

1 Transform Step

7 7 5 6 -1 -2 0 0

Averages
• Smoothed version of signal
• Lower resolution image

Differences
• Local, high frequencies
• Details missing from low resolution part
Haar Wavelets

6 8 5 9 5 5 6 6

7 7 5 6 -1 -2 0 0

7 5.5 -1 -2 0 0
Haar Wavelets

6 8 5 9 5 5 6 6

7 7 5 6 -1 -2 0 0

7 5.5 0 -0.5 -1 -2 0 0

6.25 75 0 -0.5 -1 -2 0 0

Haar Wavelets

6 8 5 9 5 5 6 6

Full Transform

6.25 75 0 -0.5 -1 -2 0 0

High Resolution Details
Medium Resolution
Details
Low Resolution Details
Average Value
2D Wavelet Transform

Standard
- Apply full transform horizontally, then full transform vertically
- Creates long, thin basis functions – bad for image compression

Non-standard
- Repeatedly apply 1 step horizontally, then 1 step vertically
- Creates square basis functions – good for image compression

What else needs compression?

Video
Textures (different requirements than plain images)
Geometry (including animations)
Anything else?
Things to Remember

Smallest representation is the shortest program
Entropy and average number of bits
Huffman coding
Transform to basis that represents the image well
Remove parts that are harder to perceive
Quantization – allocate bits to bases
Frequency space – JPEQ
Wavelets are pyramidal functions – JPEG2000
Wavelets localize features better than sin and cos