Compression

Key Concepts

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

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

DVD
- $720 \times 480 \times 24bx30f/s = \sim 30 MB/s$

High Definition DVD
- $1920 \times 1080 \times 24bx30f/s = \sim 178MB/s$

Film
- $4000 \times 3000 \times 36bx30f/s = \sim 1.5GB/s$
- 8 TB for a 90 minute movie!

Lossless vs. Lossy Compression

Lossless
- All information stored
- Original can be reconstructed exactly

Lossy
- Some information discarded
- Much higher compression ratios possible
- Strategies
  - Discard information humans won’t notice
  - Discard information that rarely occurs
Kolmogorov Complexity

What is the shortest program that can generate this fractal image?

17 KB JPEG

\[ z_0 = x + i \cdot y \]
\[ \text{repeat 1000} \]
\[ z_{n+1} = z_n^2 + c \]
\[ \text{setpixel}(x, y, |z_{n+1} - z_0|) \]

Re(c), Im(c), xmin, xmax, ymin, ymax (24B)

Image Entropy

An image has low entropy if its pixel values are predictable
An image has high entropy if its pixels values are random (unpredictable)

0 entropy  Some entropy  High entropy

0 entropy  Some entropy  High entropy
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
00 01 10 11 01 11 00 11 = 16 bits

$\frac{1}{4} \times 2 + \frac{1}{4} \times 2 + \frac{1}{4} \times 2 + \frac{1}{4} \times 2 = 2$ 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 10 0 110 0 111 0 10 = 14 bits

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

Given a random variable X with possible event outcomes $x_1, x_2, \ldots, x_n$, and event probabilities $p(x_1), p(x_2), \ldots, p(x_n)$, the entropy is given as:

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

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

$\log 1/p(x_i)$ is the number of bits associated with $x_i$

H is the average number of bits

Huffman Coding

Input
- Probability of each possible value

Output
- Assigns a binary code to every value

Rare values receive longer codes than common values
Huffman Coding

A (.10)

B (.15)

C (.30)

D (.16)

E (.29)

AB (.25)

ABD (.41)

0

1

0

1

CE (.59)

0

1

A (.10)

B (.15)

C (.30)

E (.29)
Huffman Coding

Algorithm

- Start with all events in their own block
- Repeat until there is only one block left:
  - Merge the two blocks with lowest probabilities into a single block whose probability is the sum of its children
  - Single block becomes a parent of two merged blocks
- Results in a binary tree
- Path from the final block to the leaf node is the code for each event
Transform Coding

If you transform the input data into another space, you can make it much more predictable.

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

Difference Operator

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

Transform Coding

All colors equally probable

D(Image) has 0 entropy
Two Observations on Images

In natural images, low frequencies are more common than high frequencies
That is, more bits should be allocated to low frequencies

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 Basis

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

Image space

Frequency space

Lossy Compression

262,144 pixels

43384 largest terms, 16%
(dropped 218,760 terms)
Lossy Compression

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 Image

Quantization Matrix

= 

Quantized DCT Image
Storage Order

\[
\begin{bmatrix}
-36 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\
0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\
-3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\
-4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

Error

\[\text{error} = \text{abs}(\text{original} - \text{compressed}) \times 8\]
Peak Signal-to-Noise Ratio

Original Image (I) and new image (I')

\[ MSE = \frac{1}{N} \sum_{i=0}^{N} \| I(i) - I'(i) \|^2 \]

\[ RMSE = \sqrt{MSE} \]

\[ PSNR = 10 \log_{10} \left( \frac{I_{max}^2}{MSE} \right) \]

\[ = 20 \log_{10} \left( \frac{I_{max}}{RMSE} \right) \]

Wavelets

Cosines

Wavelets
Haar Wavelets

Scaling function = average

Wavelet = difference

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

7 7 5 6 -1 -2 0 0

1 Transform Step

Averages
- Smoothed version of signal
- Lower resolution image

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

\[
\begin{array}{cccccccc}
6 & 8 & 5 & 9 & 5 & 5 & 6 & 6 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
7 & 7 & 5 & 6 & -1 & -2 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
& & & & -1 & -2 & 0 & 0 \\
\end{array}
\]
Haar Wavelets

\[ \begin{array}{cccccccc}
6 & 8 & 5 & 9 & 5 & 5 & 6 & 6 \\
\end{array} \]

\[ \begin{array}{cccccccc}
7 & 7 & 5 & 6 & -1 & -2 & 0 & 0 \\
\end{array} \]

\[ \begin{array}{cccccccc}
7 & 5.5 & 0 & -0.5 & -1 & -2 & 0 & 0 \\
\end{array} \]
Haar Wavelets

\[
\begin{array}{ccccccc}
6 & 8 & 5 & 9 & 5 & 5 & 6 & 6 \\
\downarrow & & & & & & \\
7 & 7 & 5 & 6 & -1 & -2 & 0 & 0 \\
\downarrow & & & & & & \\
7 & 5.5 & 0 & -0.5 & -1 & -2 & 0 & 0 \\
\end{array}
\]

CS148 Lecture 16  Pat Hanrahan, Fall 2009
Haar Wavelets

6 8 5 9 5 5 6 6

Full Transform

6.25 75 0 -5 -1 -2 0 0

→ High Resolution Details
→ Medium Res. Details
→ Low Resolution Details
→ Average Value

2D Wavelet Transform

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

Non-standard (better way)
- Repeatedly apply 1 step horizontally, then 1 step vertically
- Creates square basis functions – good for image compression
Haar Wavelets

What else needs compression?

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

Lossless
- No error
- Compression depends on the entropy

Lossy
- Error, but more compression
- Human visual system less sensitive to high freq.
- Distribute error to higher frequencies
- Different error metrics (PSNR)

Different basis functions
- JPEG / Discrete cosine transform (DCT) - good
- JPEG2000 / Wavelets - better