Data Compression

- Block Processing
- Discrete Cosine Transform (DCT)
- JPEG

HW 12 (last one) is posted.
- 2 problems + lab
- no check-in
- due December 3 at 5pm

Practice Final Exam is posted.
- Solutions will be posted December 3

No office hours November 25 or 26 (Thanksgiving)

November 23, 2021

Data Compression

Fourier methods underlie many common data compression schemes, including JPEG (for pictures), MP3 (for audio) and MPEG (for video).

Example: JPEG (Joint Photographic Experts Group) Encoding
1. color encoding: RGB → YCrCb
2. 2D DCT (discrete cosine transform): a kind of Fourier series
3. quantization to achieve perceptual compression (lossy)
4. run-length and Huffman encoding (lossless)

We will focus on steps 2 & 3: the DCT and quantization of its components.
- the image is broken into 8×8 pixel blocks
- each block is represented by its 8×8 DCT coefficients
- each DCT coefficient is quantized, using higher resolutions for coefficients with greater perceptual importance

Block Processing

The idea behind block processing is that small patches of an image can be coded efficiently (with just a few bits).

Start with an image, such as this ball.

Break the image into blocks.

Represent each block with as few bits as possible.

Energy Compaction

The block has 8×8 = 64 pixels.

Representing each pixel in a block with an 8-bit number → a total of 64 bytes for this block.
Energy Compaction
Try coding the 2D DFT instead. Here is the magnitude of the 2D DFT.

\[
f[r, c] \quad \log_{10} |F[k_r, k_c]|\]

This looks promising. There are only 14 discrete frequencies with magnitudes greater than \(F[0, 0]/100\).

Retaining just the DC term and these 14 other components introduces little error and reduces the byte count by a factor of \(64/15 = 4.27\).

Energy Compaction
Try coding the 2D DCT instead. Here is the magnitude of the 2D DCT.

\[
f'[r, c] \quad \log_{10} |F'_C[k_r, k_c]|\]

This looks even more promising. Now there are only 2 discrete frequencies with magnitudes greater than \(F[0, 0]/100\).

Retaining just these 3 terms reduces the byte count by a factor of \(64/3 = 21.3\), which is five times fewer bytes than that for the DFT.

Energy Compaction
Consider the structure of the patch that we have been examining.

It’s basically a 2D ramp: brighter in the upper right than in the lower left. Such blocks are common, and not so easy to compress with the DFT.

Compaction of a Ramp
Compare the DFT and DCT of a “ramp.”

Why are there so many high frequencies in the DFT? And why are there fewer in the DCT?
**Discrete Cosine Transform (DCT)**

The DFT of a Ramp

The DFT is the Fourier series of a periodically extended version of a signal.

\[
f_p(x) = \sum_{m=-\infty}^{\infty} f(x + mN)
\]

Periodic extension of a ramp results in a sawtooth wave. Step discontinuities at the window edges produce high-frequency content.

**Discrete Cosine Transform (DCT)**

The idea in the Discrete Cosine Transform (DCT) is to avoid introducing step discontinuities in periodic extension:

\[
f[n] = f[n + 8]
\]

by first replicating one period in reverse order.

The resulting “folded” function does not have a step discontinuity in value (although there is a discontinuity in slope).

**Discrete Cosine Transform (DCT)**

The DFT of the folded, stretched, and doubled signal is the DCT of the original function.

\[
g[n] = g[n + 4N]
\]

\[
f[n] \Leftrightarrow F_c[k] = G[k]
\]

**Discrete Cosine Transform (DCT)**

The DFT of the folded, stretched, and doubled signal is the DCT of the original function.

\[
f[n] = f[n + 8]
\]

Next, stretch the folded signal in time by inserting zeros between successive samples, and double the signal’s values (to keep the same DC value).

\[
g[n] = g[n + 4N]
\]

\[
G[k] = \frac{1}{4N} \sum_{n=(4N)} g[n] e^{-j \frac{2\pi}{4N} n} = \frac{1}{4N} \sum_{n=0}^{N-1} 2f[m] \left( e^{-j \frac{2\pi}{4N} (2m+1)} + e^{j \frac{2\pi}{4N} (2m+1)} \right)
\]

**Discrete Cosine Transform (DCT)**

The idea in the Discrete Cosine Transform (DCT) is to avoid introducing step discontinuities in periodic extension:

\[
f[n] = f[n + 8]
\]

by first replicating one period in reverse order.

The resulting “folded” function does not have a step discontinuity in value (although there is a discontinuity in slope).

**Discrete Cosine Transform (DCT)**

The DFT of the folded, stretched, and doubled signal is the DCT of the original function.

\[
g[n] = g[n + 4N]
\]

\[
f[n] \Leftrightarrow F_c[k] = G[k]
\]

**Discrete Cosine Transform (DCT)**

The DFT of the folded, stretched, and doubled signal is the DCT of the original function.

\[
f[n] = f[n + 8]
\]

Next, stretch the folded signal in time by inserting zeros between successive samples, and double the signal’s values (to keep the same DC value).

\[
g[n] = g[n + 4N]
\]

\[
G[k] = \frac{1}{4N} \sum_{n=(4N)} g[n] e^{-j \frac{2\pi}{4N} n} = \frac{1}{4N} \sum_{n=0}^{N-1} 2f[m] \left( e^{-j \frac{2\pi}{4N} (2m+1)} + e^{j \frac{2\pi}{4N} (2m+1)} \right)
\]

**Discrete Cosine Transform (DCT)**

The idea in the Discrete Cosine Transform (DCT) is to avoid introducing step discontinuities in periodic extension:

\[
f[n] = f[n + 8]
\]

by first replicating one period in reverse order.

The resulting “folded” function does not have a step discontinuity in value (although there is a discontinuity in slope).

**Discrete Cosine Transform (DCT)**

The DFT of the folded, stretched, and doubled signal is the DCT of the original function.

\[
g[n] = g[n + 4N]
\]

\[
f[n] \Leftrightarrow F_c[k] = G[k]
\]

**Discrete Cosine Transform (DCT)**

The DFT of the folded, stretched, and doubled signal is the DCT of the original function.

\[
f[n] = f[n + 8]
\]

Next, stretch the folded signal in time by inserting zeros between successive samples, and double the signal’s values (to keep the same DC value).

\[
g[n] = g[n + 4N]
\]

\[
G[k] = \frac{1}{4N} \sum_{n=(4N)} g[n] e^{-j \frac{2\pi}{4N} n} = \frac{1}{4N} \sum_{n=0}^{N-1} 2f[m] \left( e^{-j \frac{2\pi}{4N} (2m+1)} + e^{j \frac{2\pi}{4N} (2m+1)} \right)
\]

**Discrete Cosine Transform (DCT)**

The idea in the Discrete Cosine Transform (DCT) is to avoid introducing step discontinuities in periodic extension:

\[
f[n] = f[n + 8]
\]

by first replicating one period in reverse order.

The resulting “folded” function does not have a step discontinuity in value (although there is a discontinuity in slope).

**Discrete Cosine Transform (DCT)**

The DFT of the folded, stretched, and doubled signal is the DCT of the original function.

\[
g[n] = g[n + 4N]
\]

\[
f[n] \Leftrightarrow F_c[k] = G[k]
\]

**Discrete Cosine Transform (DCT)**

The DFT of the folded, stretched, and doubled signal is the DCT of the original function.

\[
f[n] = f[n + 8]
\]

Next, stretch the folded signal in time by inserting zeros between successive samples, and double the signal’s values (to keep the same DC value).

\[
g[n] = g[n + 4N]
\]

\[
G[k] = \frac{1}{4N} \sum_{n=(4N)} g[n] e^{-j \frac{2\pi}{4N} n} = \frac{1}{4N} \sum_{n=0}^{N-1} 2f[m] \left( e^{-j \frac{2\pi}{4N} (2m+1)} + e^{j \frac{2\pi}{4N} (2m+1)} \right)
\]
Comparison of DFT and DCT Basis Functions

DFT (real and imaginary parts) versus DCT.

- Re(e^{i\pi kn/N})
- Im(e^{i\pi kn/N})
- \cos(\pi (n+1/2)/N)

The DCT basis functions are symmetric or antisymmetric about n = 3.5 at half-integer multiples of the fundamental frequency.

Find DCT Synthesis Equation (using Orthogonality)

We would like to express f[n] as a weighted sum of DCT basis functions.

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

Multiply both sides by \phi_k[n] and sum over n.

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

Left-hand side is F_C[l]. Swap order of summation on the right-hand side.

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

Evaluate the right-hand side using orthogonality.

N_{F_C}[l] = \begin{cases} N_0 & \text{if } k = l = 0 \\ \frac{N_0}{k} & \text{if } k = l \neq 0 \\ 0 & \text{otherwise} \end{cases}

f[n] = \sum_{k=0}^{N-1} \hat{a}_k \cos \left( \frac{\pi k}{N} \left( n + \frac{1}{2} \right) \right) = F_C[0] + 2 \sum_{k=1}^{N-1} F_C[k] \cos \left( \frac{\pi k}{N} \left( n + \frac{1}{2} \right) \right)

Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is described by analysis and synthesis equations that are analogous to those of the DFT.

F_C[k] = \frac{1}{N} \sum_{n=0}^{N-1} f[n] \cos \left( \frac{\pi k}{N} \left( n + \frac{1}{2} \right) \right) \quad \text{(analysis)}

f[n] = F_C[0] + 2 \sum_{k=1}^{N-1} F_C[k] \cos \left( \frac{\pi k}{N} \left( n + \frac{1}{2} \right) \right) \quad \text{(synthesis)}

DCT Basis Functions

As with the DFT, the DCT basis functions are orthogonal to each other.

\phi_k[n] = \cos \left( \frac{\pi k}{N} \left( n + \frac{1}{2} \right) \right)

\phi_0[n] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \phi_k[n]\phi_0[n] = \frac{1}{N} \sum_{n=0}^{N-1} \cos \left( \frac{\pi k}{N} \left( n + \frac{1}{2} \right) \right) \cos \left( \frac{\pi l}{N} \left( n + \frac{1}{2} \right) \right)

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

= \begin{cases} 1 & \text{if } k = l = 0 \\ \frac{1}{2} & \text{if } k = l \neq 0 \\ 0 & \text{otherwise} \end{cases}

The sum across time of the product of two different basis functions is zero.

2D DCT Basis Functions

Grid of 8 x 8 basis functions organized in rows (k_r) and columns (k_c).

Each basis function has 8 x 8 elements organized by row r and column c.

The DCT has a number of useful properties:
- It maps spatial domain to frequency domain (much like DFT).
- If input has length N, then the output has length N.
- It is purely real-valued (unlike DFT).
- It eliminates discontinuities caused by periodic extension of DFT.

However:
- It does not have a “filtering” property.

Discrete Cosine Transform

The DCT has a number of useful properties:
- It maps spatial domain to frequency domain (much like DFT).
- If input has length N, then the output has length N.
- It is purely real-valued (unlike DFT).
- It eliminates discontinuities caused by periodic extension of DFT.

However:
- It does not have a “filtering” property.
The filtering property of Fourier transforms results from the eigenfunction property of the Fourier basis functions.

**Eigenfunction property:** If the input to an LTI system is an eigenfunction, then the output is a scaled version of that same eigenfunction.

\[
\phi(t) \xrightarrow{\text{LTI}} \lambda \phi(t)
\]

Complex exponentials are eigenfunctions of linear, time-invariant systems.

\[
e^{j\omega t} \xrightarrow{\text{LTI}} \lambda e^{j\omega t}
\]

- scaling the amplitude of a complex exponential does not change \(\omega\)
- shifting a complex exponential in time does not change \(\omega\)

**Filter property:** If we express an input signal as a sum of eigenfunctions, then the output signal is a weighted sum of those same eigenfunctions.

\[
\text{Input: } \sum_{n} a_n e^{j\lambda n f} \xrightarrow{\text{LTI}} \sum_{n} a_n e^{j\lambda n f} = \text{Output}
\]

The DCT cannot be used for filtering because circular convolution preserves the eigenfunction property of complex exponentials.

Circular convolution of an input \(f[n]\) with unit-sample response \(h[n]\) is equivalent to conventional convolution of \(f_p[n]\), a periodically extended version of \(f[n]\), with the unit-sample response \(h[n]\).

\[
f[n] \xrightarrow{\text{Circular convolution}} f_p[n] 
\]

A similar argument holds for all of the DFT basis functions, since they are all periodic in \(N\).

The DCT cannot be used for filtering because circular convolution does not preserve the eigenfunction property of its basis functions.

\[
\phi[n] = \cos \left( \frac{\pi k}{N} \left( n + \frac{1}{2} \right) \right)
\]

Shifting the half-integer multiplies of the fundamental frequency does not simply scale the basis function.

\[
f[n] \xrightarrow{\text{Circular convolution}} f_p[n] 
\]

\[
f_p[n] \neq \lambda f[n] \text{ for } n \in [0, N-1].
\]

Therefore the DCT does not have a filtering property.

The DCT has a number of useful properties:

- It maps spatial domain to frequency domain (much like DFT).
- It is purely real-valued (unlike DFT).
- If input has length \(N\), then the output has length \(N\).
- It eliminates discontinuities caused by periodic extension of DFT.

However:

- It does not have a “filtering” property.

But the DCT represents patches of a smooth image very efficiently. For that reason, it is widely used in audio and image compression.

**Data Compression**

Fourier methods underlie many common data compression schemes, including JPEG (for pictures), MP3 (for audio) and MPEG (for video).

Example: JPEG (Joint Photographic Experts Group) Encoding

1. color encoding: RGB \(\rightarrow\) YCrCb
2. 2D DCT (discrete cosine transform): a kind of Fourier series
3. quantization to achieve perceptual compression (lossy)
4. run-length and Huffman encoding (lossless)

We will focus on steps 2 & 3: the DCT and quantization of its components.

- the image is broken into \(8 \times 8\) pixel blocks
- each block is represented by its \(8 \times 8\) DCT coefficients
- each DCT coefficient is quantized, using higher resolutions for coefficients with greater perceptual importance
Quantization

DCT amplitudes are quantized by dividing by a frequency-dependent number \( q[k_r, k_c] \) and then rounding to the nearest integer.

\[
\begin{array}{cccccccc}
16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\
14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\
\end{array}
\]

These values were chosen to represent human sensitivities. High frequencies are more coarsely quantized than middle frequencies.

Different tables of this form are used to implement different “qualities.”

Summary

The number of bits used to represent a signal is of critical importance in modern communication systems.

Modern compression systems combine lossless compression techniques (such as LZW, Huffman, and zip) with perceptual (lossy) compression based on Fourier representations.

The Discrete Cosine Transform (DCT) is a close relative of the DFT that is more easily compressed using block coding methods.

The DCT is not useful for filtering because its basis functions are not eigenfunctions of LTI systems.

The DCT does provide significantly improved data compaction and is widely used in both audio and video signal processing.

JPEG: Results

1%: 1666 bytes 10%: 2550 bytes 20%: 3595 bytes

40%: 5318 bytes 80%: 10994 bytes 100%: 47k bytes