6.300 Signal Processing

Week 13, Lecture A:
Fourier-based Compression

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

Final:
Friday December 19,
1:30-4:30pm Johnson Track

Lecture slides are available on CATSOOP:
https://sigproc.mit.edu/fall25

Check yourself!

If you use your iphone to take a 1-min 4K video, what is the file size if we naively consider 3840x2160 pixels per frame with each pixel represent by 3 bytes (which is 3x8 bits) and 30 frames/second?

Size:

$$3840 \times 2160 \times 3 \times 30 \times 60 = 44{,}789{,}760{,}000 \approx 44\text{GB!}$$

Fourier-based Compression

Compression

Discrete Cosine Transform:

- What is it?
- How it is related to Discrete Fourier Transform (DFT)?
- Why is it better?

JPEG Overview

Effects of Sampling

original: 2048×1536

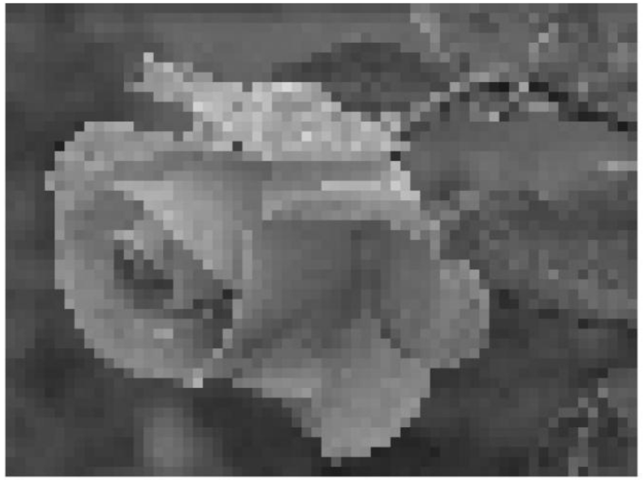
downsampled: 1024 x 768

downsampled: 512 x 384

downsampled: 256 x 192

downsampled: 128 x 96

downsampled: 64 x 48



Quantization of Images

Another way to reduce bit count is to reduce the number of bits used to represent each sample.

Images are 280 × 280 pixels

8 bit image

7 bit image

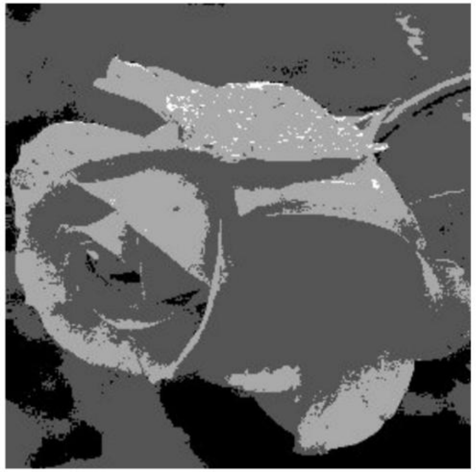
6 bit image

5 bit image

4 bit image

3 bit image

2 bit image



1 bit image

Compression

How can we reduce the number of bits needed to represent the signal without affecting quality?

There are two general types of compression:

- **Lossless**
  - take advantage of statistical dependences in the bitstream
  - enormously successful: RLE, LZW, Huffman, zip ...
  - learn more in 6.1210, ...

- **Lossy**
  - focus on perceptually important information
  - enormously successful: JPEG, MP3, MPEG ...
  - topic of today’s lecture

Lossy Compression

Key idea: throw away the “unimportant” bits (i.e., bits that won’t be noticed). Doing this involves knowing what it means for something to be noticeable.

Many aspects of human perception are frequency based.

→ many lossy formats use frequency-based methods (along with models of human perception).

Lossy Compression: High-level View

To Encode:

- Split signal into “frames”
- Transform each frame into Fourier representation
- Throw away (or attenuate) some coefficients
- Additional lossless compression (LZW, RLE, Huffman, etc.)

To Decode:

- Undo lossless compression
- Transform each frame into time/spatial representation

This is pretty standard! Both JPEG and MP3, for example, work roughly this way.

Given this, one goal is to get the “important” information in a signal into relatively few coefficients in the frequency domain (“energy compaction”).

JPEG

JPEG (Joint Photographic Experts Group) Encoding

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

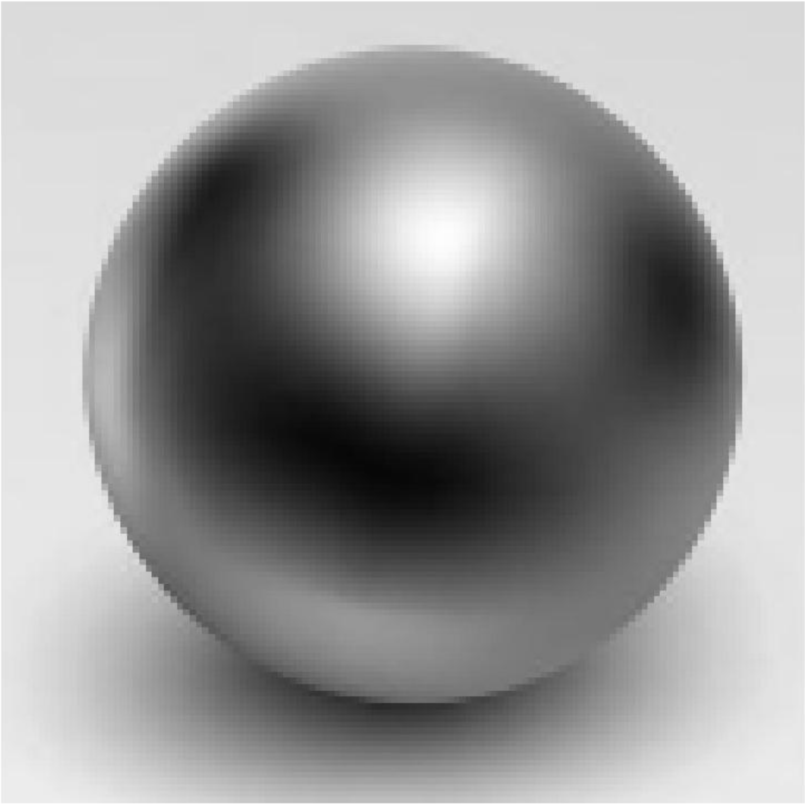
We will focus on the DCT for today's lecture.

- 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

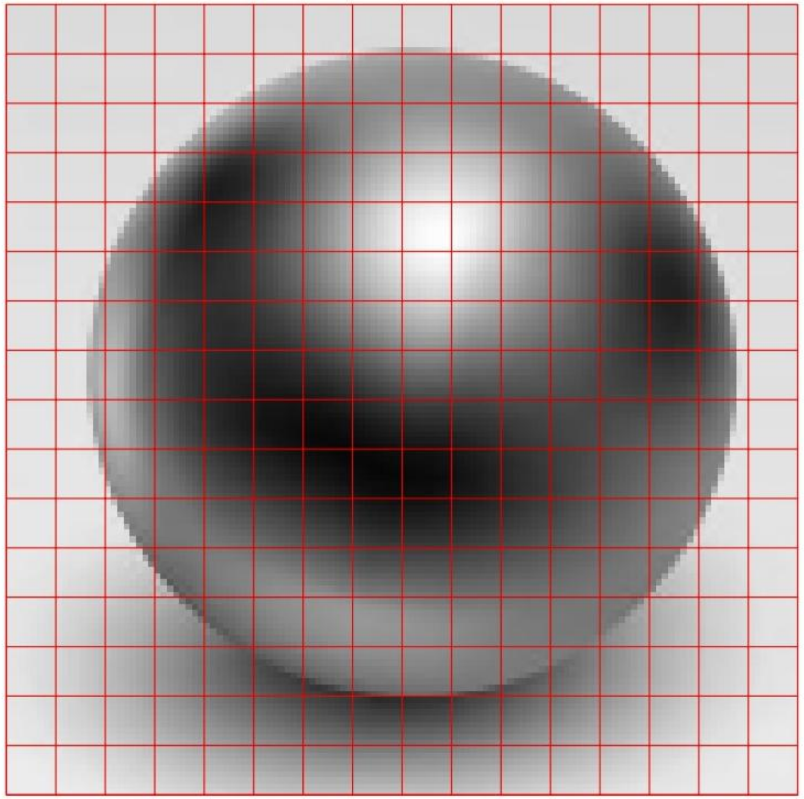
Block Processing

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.



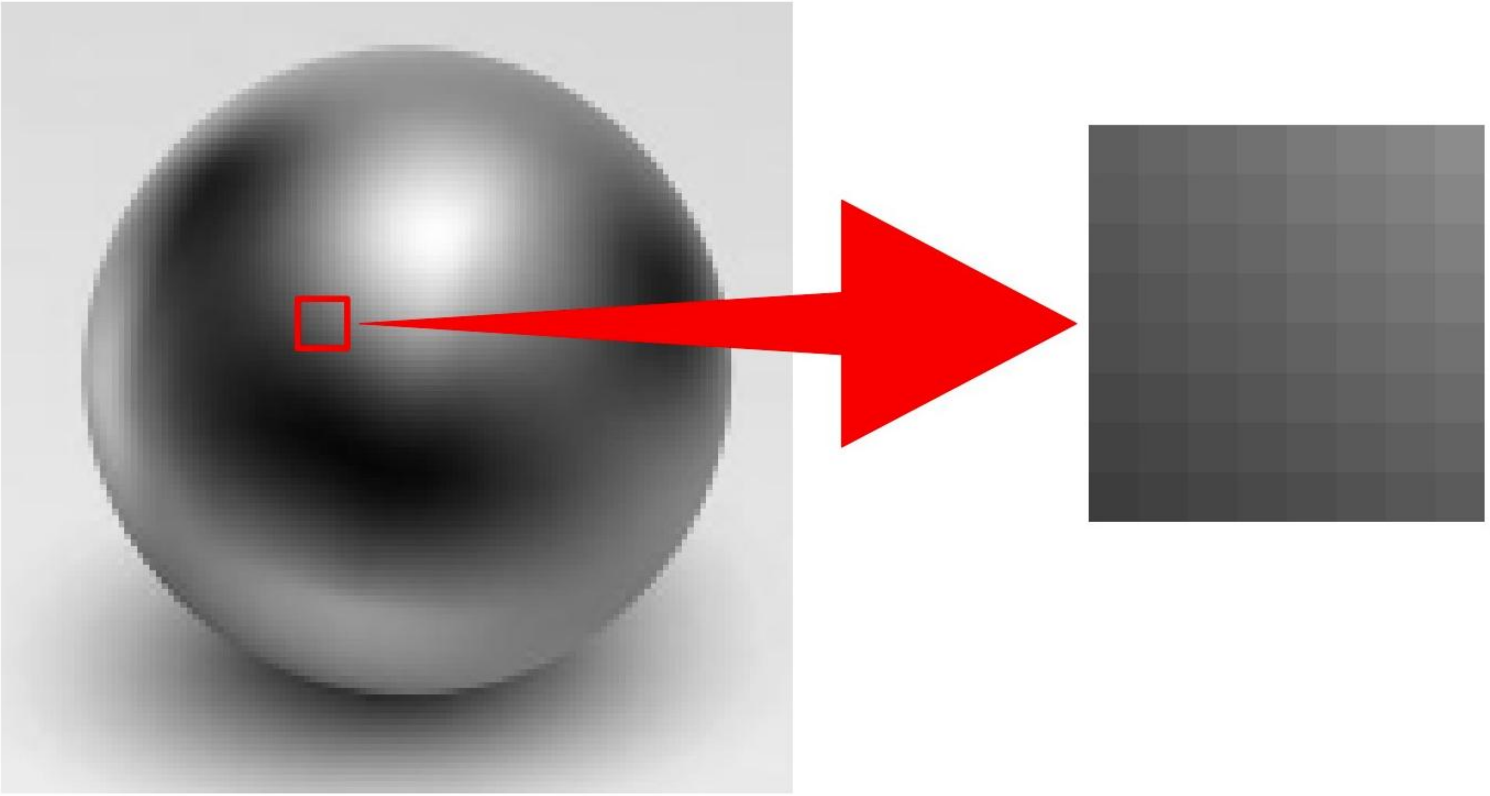
Break the image into blocks.



Block Processing

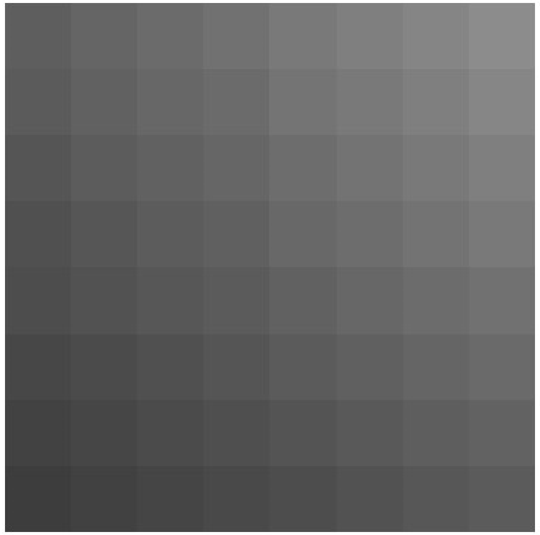
The idea behind block processing is that small patches of an image can be coded efficiently (with just a few bits).

Represent each block with as few bits as possible.



Block Processing

The block has 8×8 = 64 pixels.



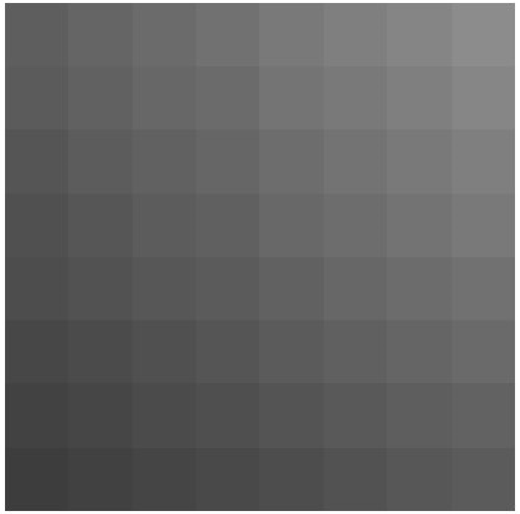
Representing each pixel in a block with an 8-bit number → a total of 64 bytes for this block.

We would like to come up with a way of coding the block with fewer than 64 bytes of data.

Data Compaction

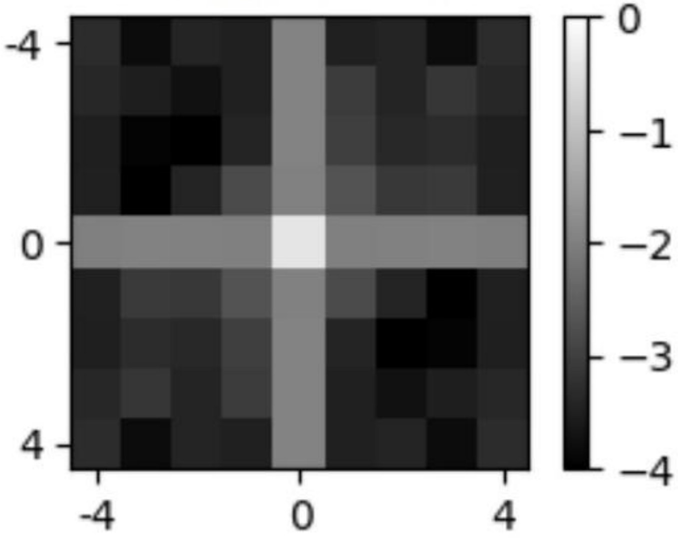
get the “important” information in a signal into relatively few coefficients in the frequency domain

$f[r, c]$



$\overset{\text{DFT}}{\Longrightarrow}$

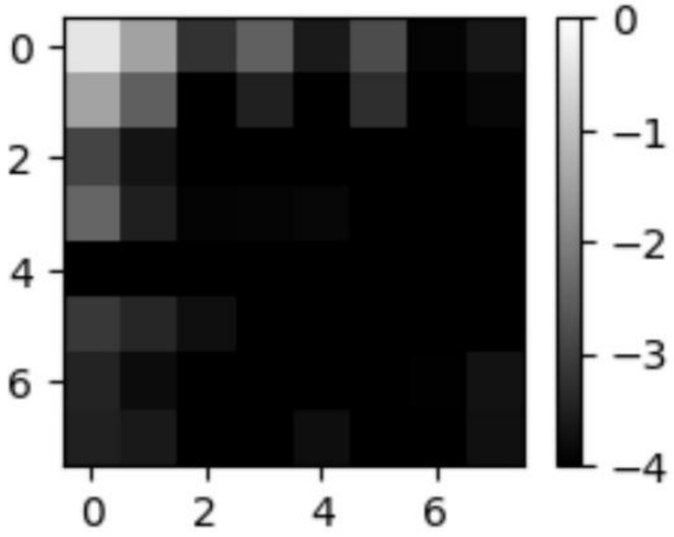
$\log_{10} |F[k_r, k_c]|$



There are 15 discrete frequencies with magnitudes greater than F[0, 0]/100.

$\overset{\text{DCT}}{\Longrightarrow}$

$\log_{10} |F_C[k_r, k_c]|$

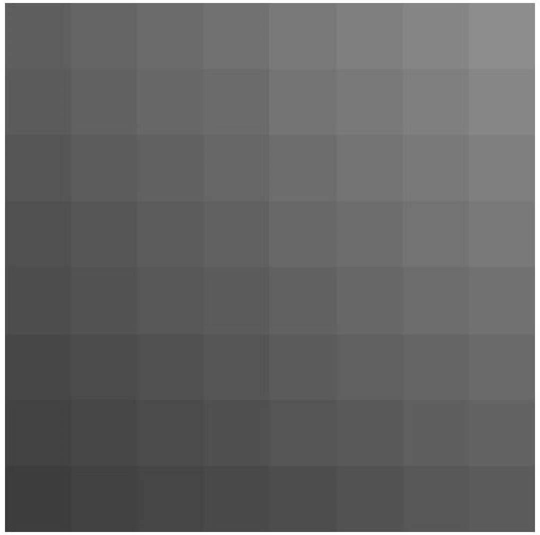


There are only 2 discrete frequencies with magnitudes greater than F[0, 0]/100.

What is the DCT and why is it better than the DFT?

Data Compaction

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

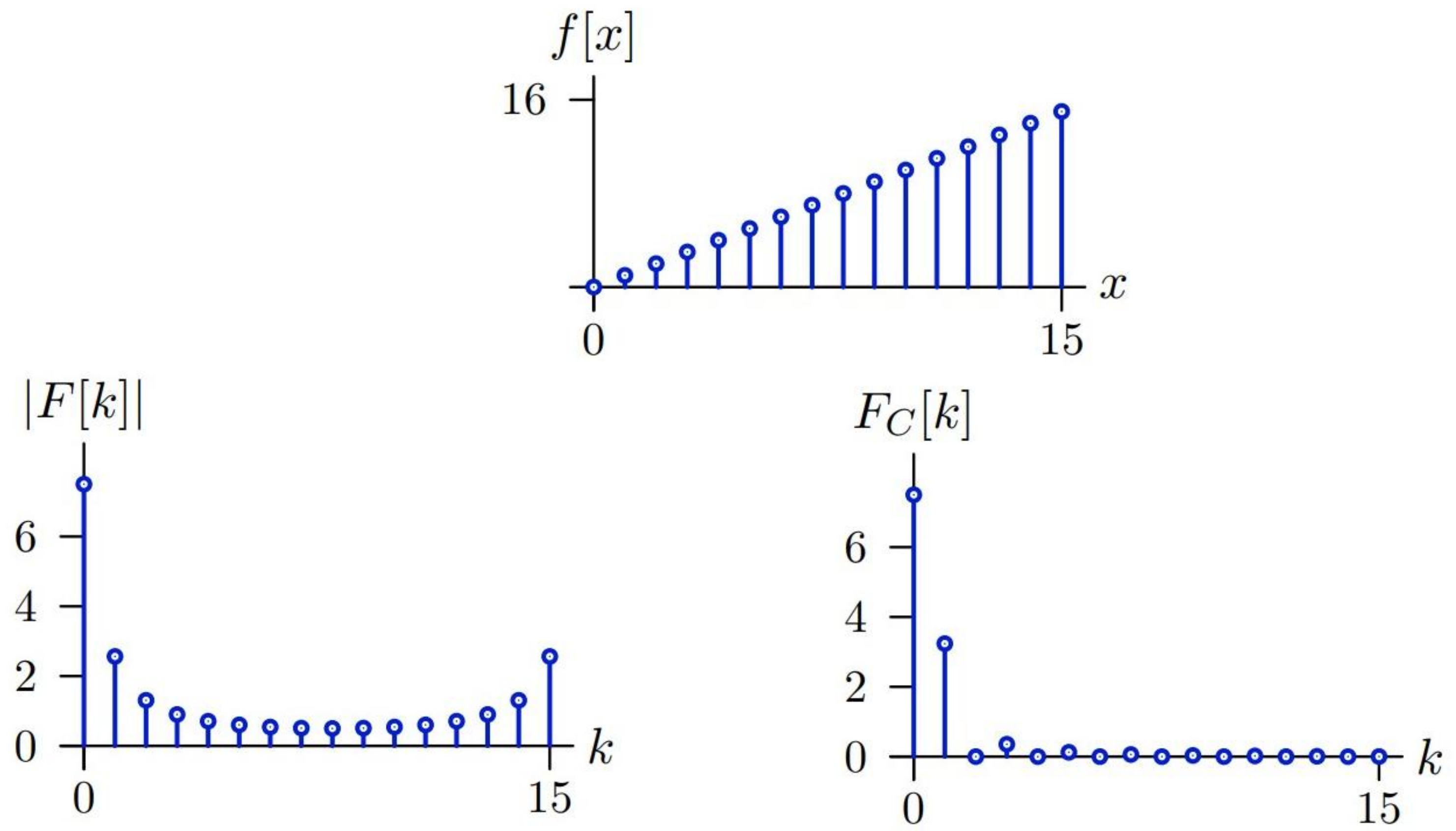


It’s basically a 2D ramp: brighter in the upper right than in the lower left.

Such blocks are common, and not so easy to compress with the DFT.

Compaction of a Ramp

Comparisons of the DFT and DCT of a “ramp.”



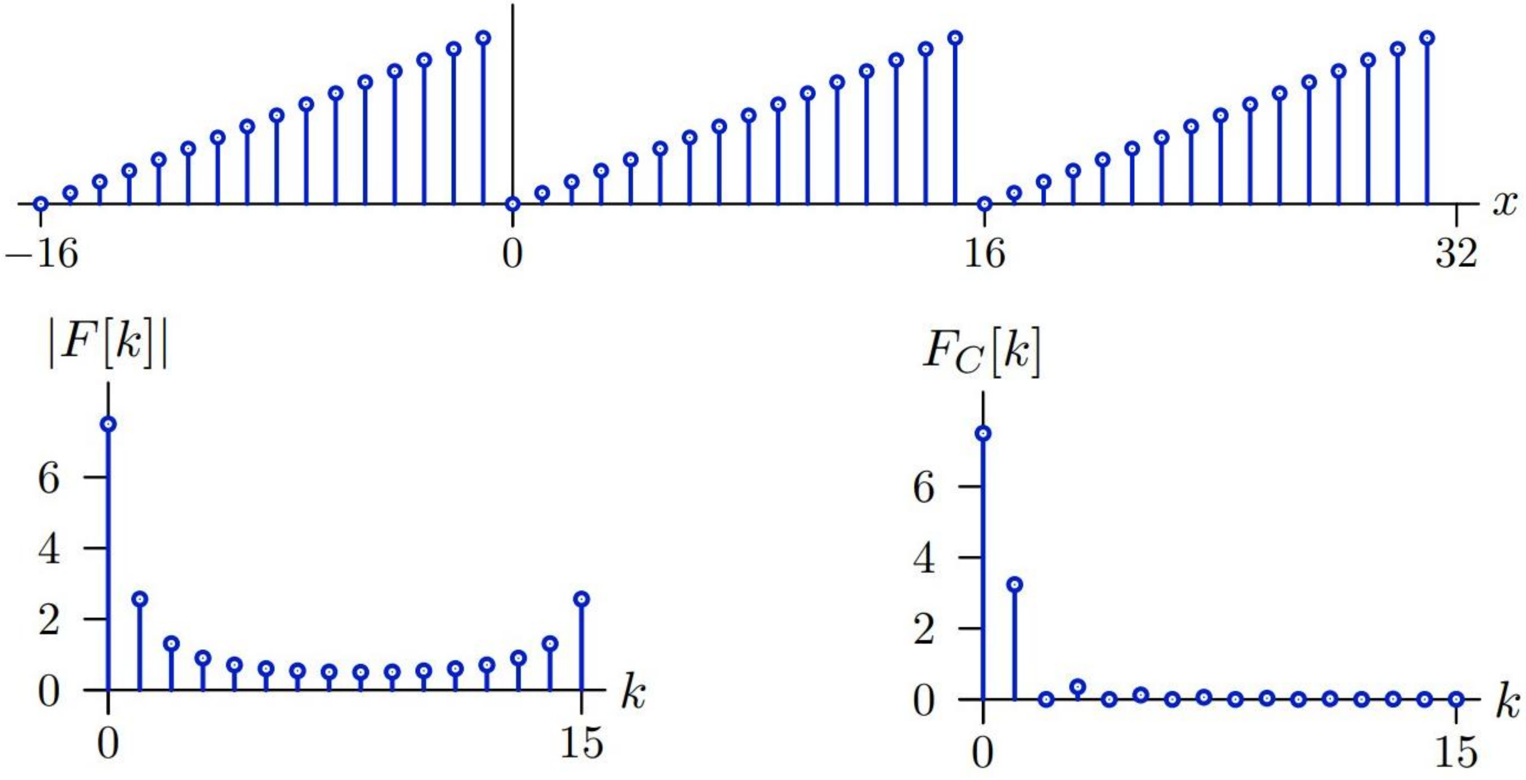
Why are there so many high frequencies in the DFT?

And why are there fewer in the DCT?

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

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

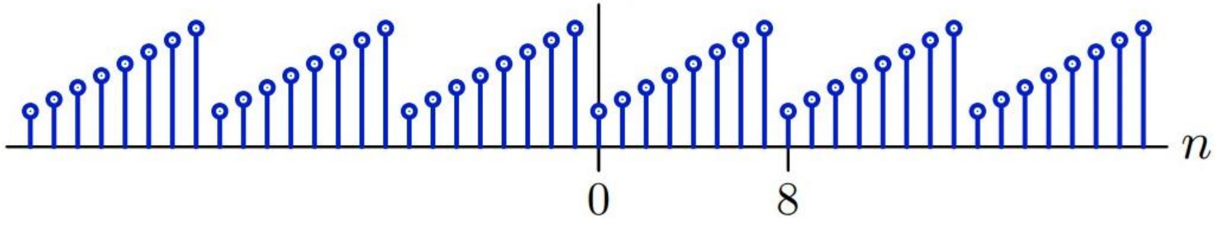
$$f[n] = F_C[0] + 2\sum_{k=1}^{N-1} F_C[k]cos\left(\frac{\pi k}{N}(n+\frac{1}{2})\right) \qquad \text{(synthesis)}$$

How was DCT derived and why is it being used?

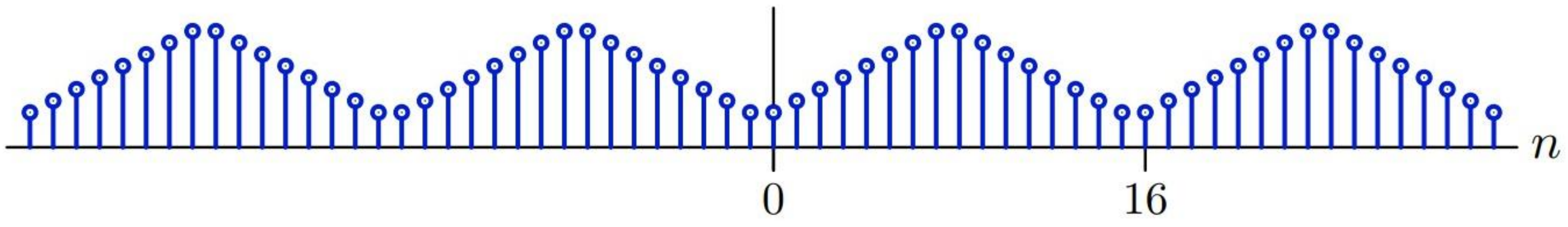
Discrete Cosine Transform (DCT)

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

$$f[n] = f[n+8]$$



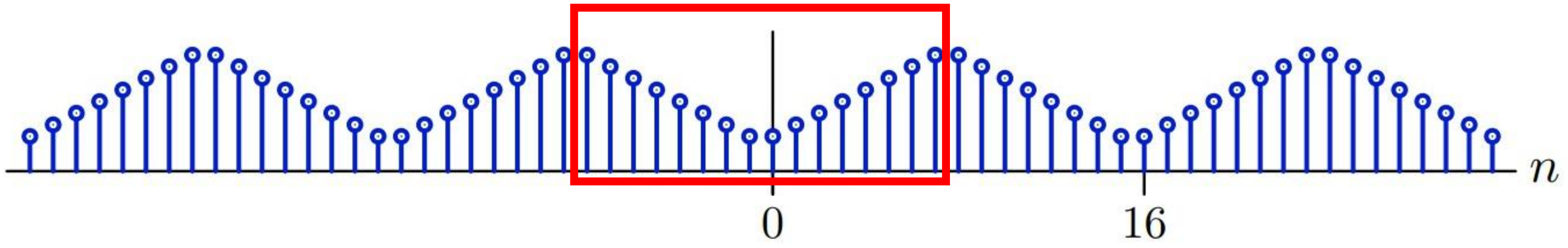
by first replicating one period in reverse order.



The resulting “folded” function does not have a step discontinuity.

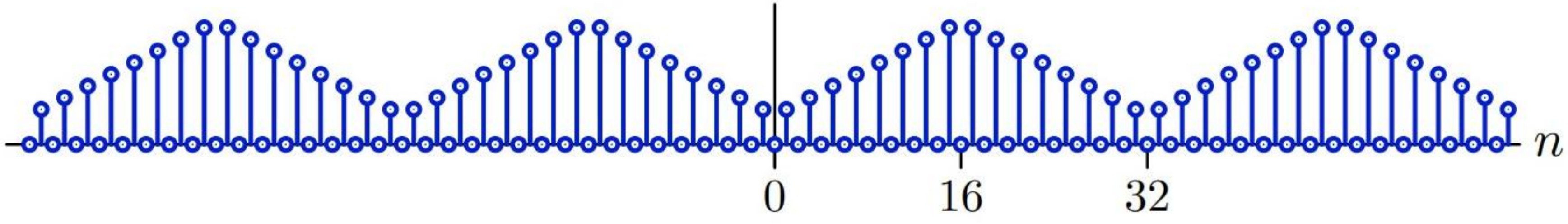
Discrete Cosine Transform (DCT)

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



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

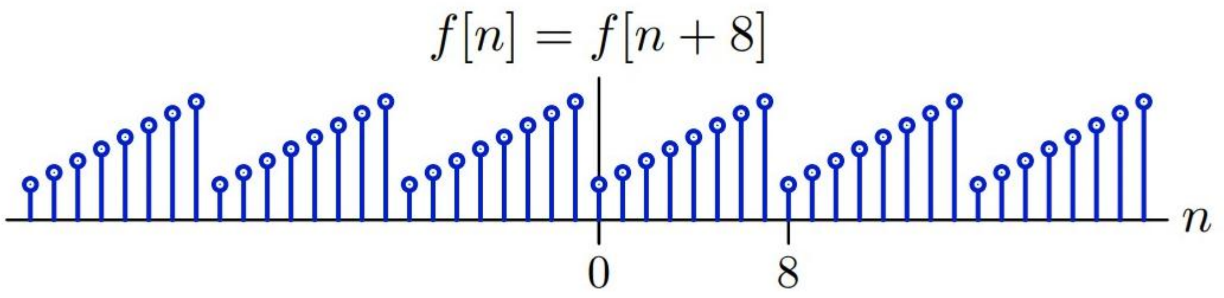
$$g[n] = g[n + 4N]$$

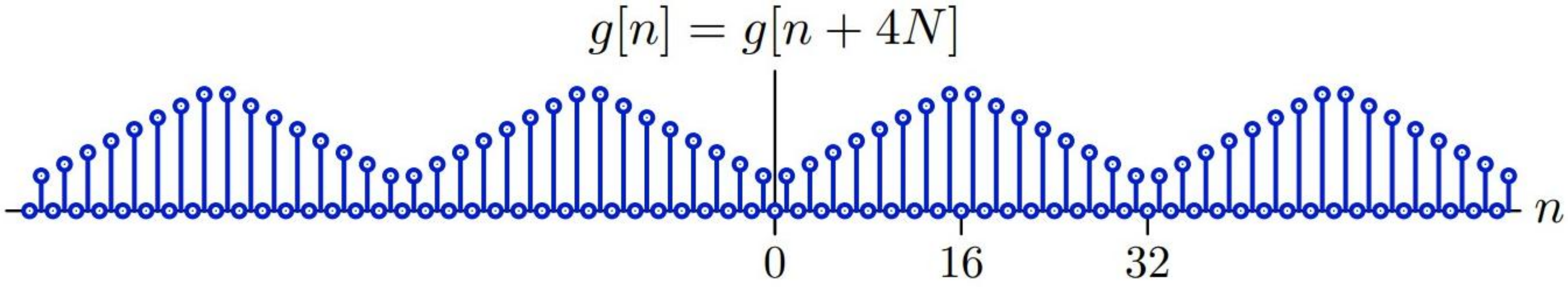


The resulting signal is symmetric about n = 0, periodic in 4N, and contains only odd numbered samples.

Discrete Cosine Transform (DCT)

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



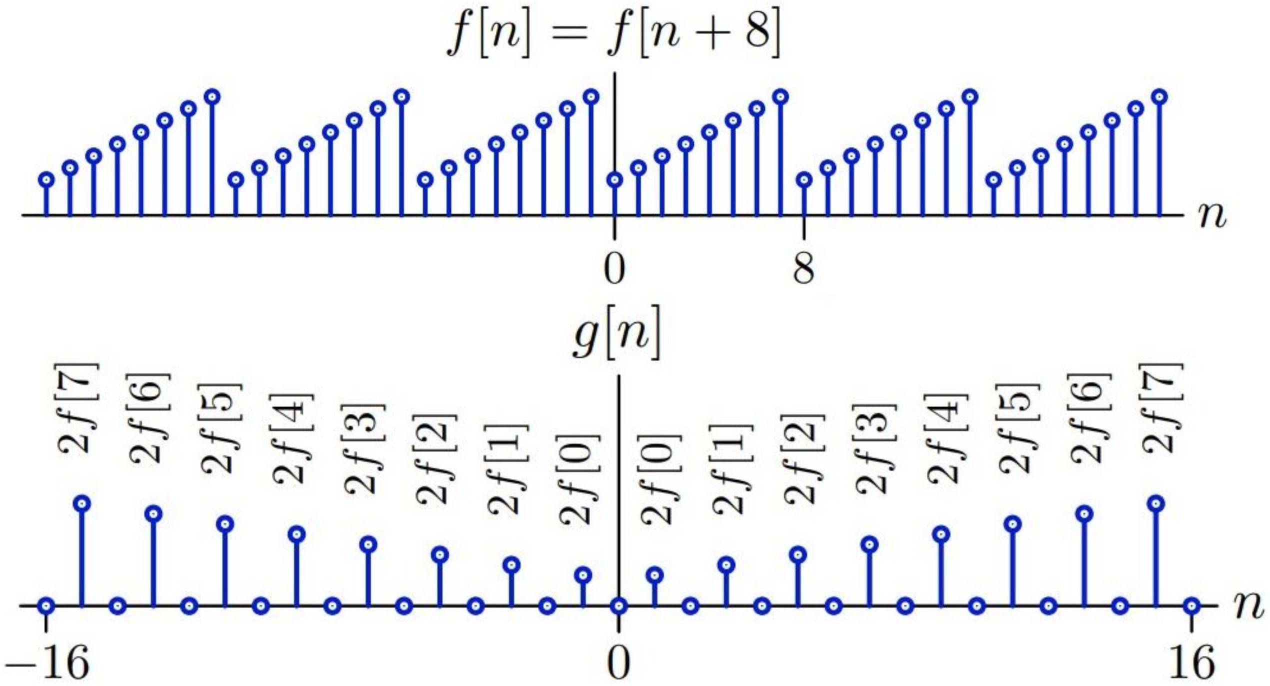


$$g[n] \overset{DFT}{\Longleftrightarrow} G[k]$$

$$f[n] \overset{DCT}{\Longleftrightarrow} F_C[k] = G[k]$$

Discrete Cosine Transform (DCT)

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



$$g[n] \overset{DFT}{\Longleftrightarrow} G[k]$$

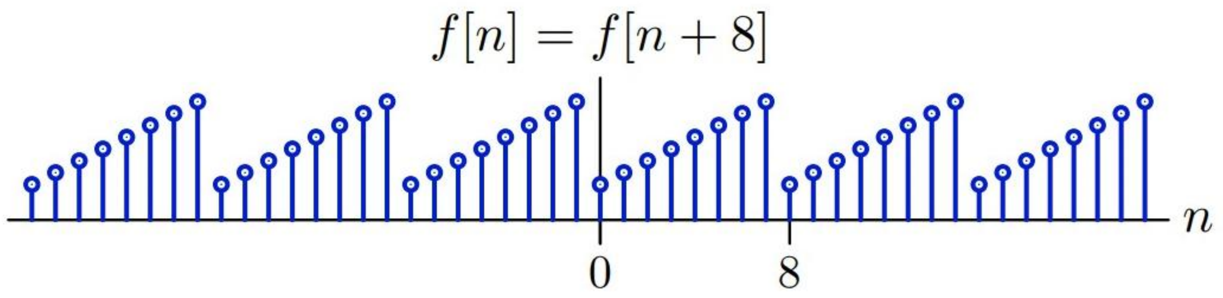
$$f[n] \overset{DCT}{\Longleftrightarrow} F_C[k] = G[k]$$

$$G[k] = \frac{1}{4N} \sum_{n=<4N>} g[n] e^{-j\frac{2\pi}{4N}kn} \quad = \frac{1}{4N} \sum_{m=0}^{N-1} 2f[m](e^{-j\frac{2\pi}{4N}k(2m+1)} + e^{j\frac{2\pi}{4N}k(2m+1)})$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} f[m] cos\left(\frac{2\pi k}{4N}(2m+1)\right) \boxed{= \frac{1}{N} \sum_{m=0}^{N-1} f[m] cos\left(\frac{\pi k}{N}(m+\frac{1}{2})\right) = F_C\ [k]}$$

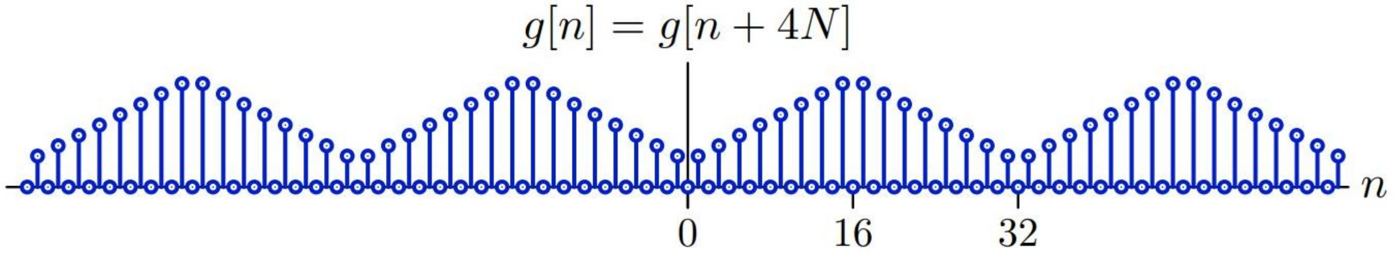
Discrete Cosine Transform (DCT)

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

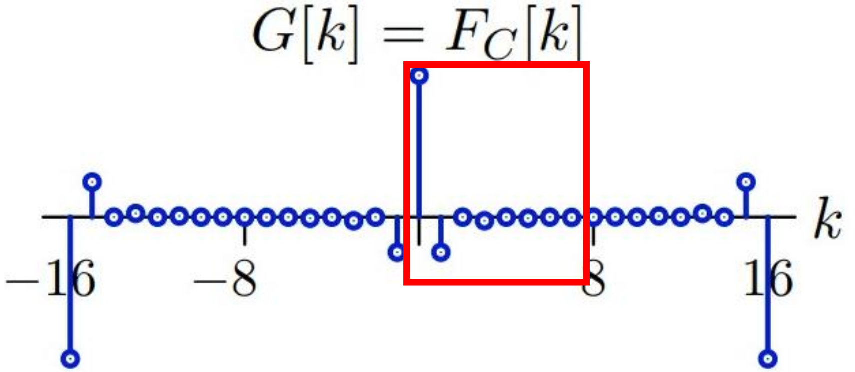


$$g[n] \overset{DFT}{\Longleftrightarrow} G[k]$$

$$f[n] \overset{DCT}{\Longleftrightarrow} F_C[k] = G[k]$$



The DFT of $g[n]$ is real, symmetric about *k* = 0 and with anti-symmetry within a period. It is completely characterized by *N(=8)* unique numbers in $G[k]$: $G[0]$ to $G[7]$.



Discrete Cosine Transform (DCT)

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

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

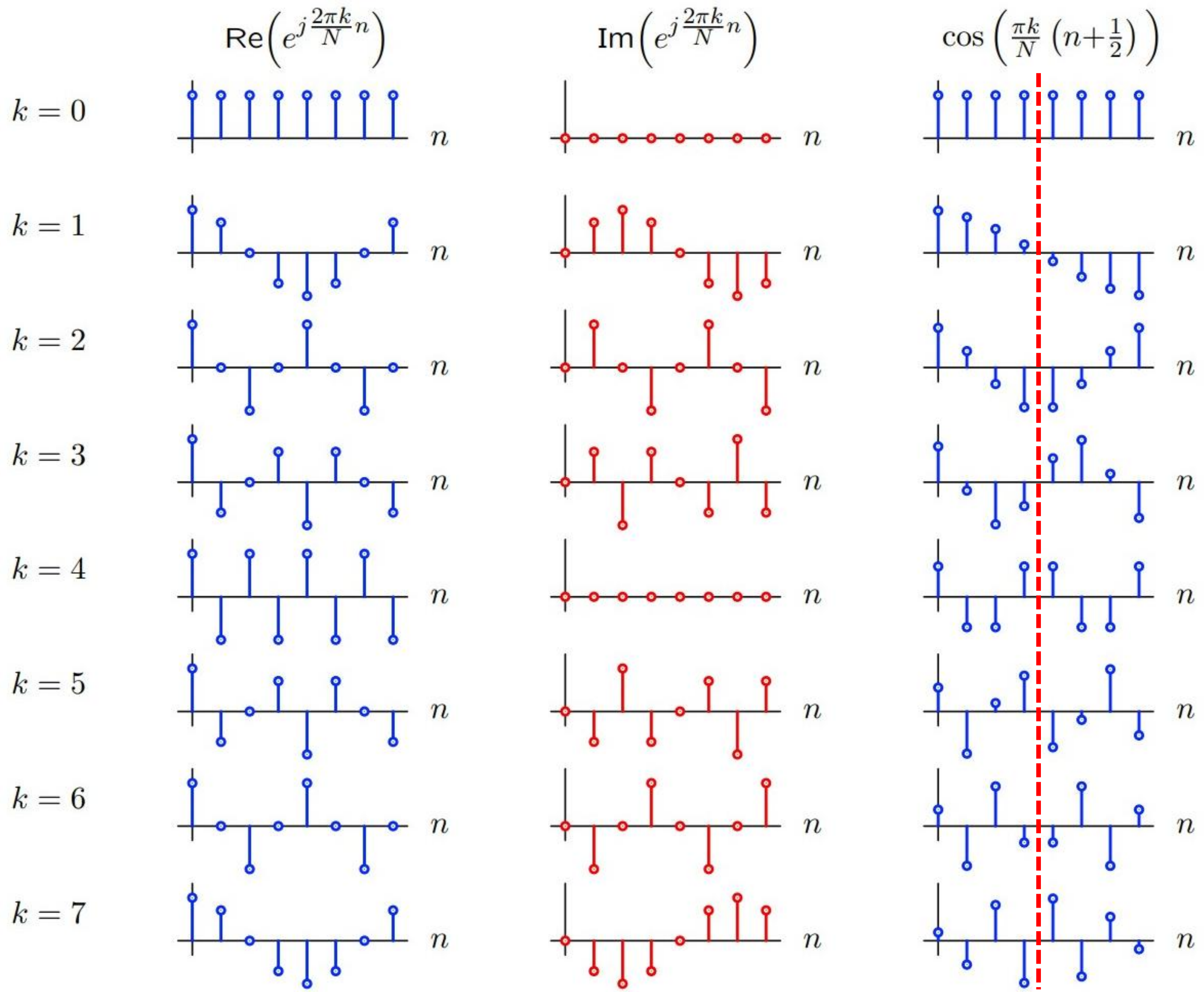
$$f[n] = F_C[0] + 2\sum_{k=1}^{N-1} F_C[k]cos\left(\frac{\pi k}{N}(n+\frac{1}{2})\right) \qquad \text{(synthesis)}$$

If $f[n] = \delta[n]$ what is its corresponding $F_C[k]$? **Participation question for lecture**

How was the synthesis equation derived?

Comparison of DFT and DCT Basis Functions

DFT (real and imaginary parts) versus DCT.



The k[th] DCT basis function of order N is given by:

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

- $\phi_k[n]$ is not periodic in N!
- Special symmetric (or anti-symmetric) properties depending on k

Orthogonality of DCT Basis Functions

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

$$\boxed{\sum_{n=0}^{N-1} \phi_k[n]\phi_l[n] = \begin{cases} N & if\ k = l = 0 \\ \frac{N}{2} & if\ k = l \neq 0 \\ 0 & otherwise \end{cases}}$$ (orthogonality)

Find DCT Synthesis Equation (using Orthogonality)

We first express $f[n]$ as a weighted sum of DCT basis functions:

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

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

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

Compare with analysis equation:

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

Left-hand side is $\mathrm{N}F_C[l]$. Swaps order of summation on the right-hand side.

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

$$\boxed{\sum_{n=0}^{N-1} \phi_k[n]\phi_l[n] = \begin{cases} N & if\ k=l=0 \\ \frac{N}{2} & if\ k=l\neq 0 \\ 0 & otherwise \end{cases}}$$

Evaluate the right-hand side using orthogonality.

$$\mathrm{N}F_C[l] = \begin{cases} Na_0 & if\ l=0 \\ \frac{1}{2}Na_l & otherwise \end{cases}$$

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

Discrete Cosine Transform (DCT)

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

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

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

The DCT has a number of useful properties:

- It maps spatial domain to frequency domain (like DFT)
- If input has length N, then the output has length N.
- It is purely real-valued (unlike DFT) .
- It reduces 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.

JPEG

JPEG (Joint Photographic Experts Group) Encoding

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

We will focus on 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

Quantization

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

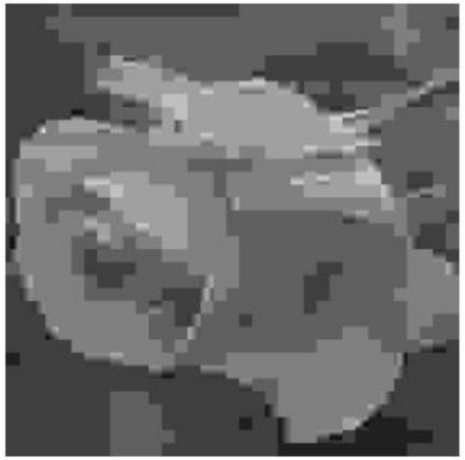
$$
\begin{array}{c|cccccccc}
q[k_r,k_c] & & & & k_c \rightarrow & & & & \\
\hline
 & 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
 & 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\
 & 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\
k_r & 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\
\downarrow & 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\
 & 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\
 & 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\
 & 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99
\end{array}
$$

These values were chosen to represent human sensitivities.

High frequencies are more coarsely quantized than low frequencies.

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

JPEG: Results



1%: 1666 bytes

10%: 2550 bytes

20%: 3595 bytes

40%: 5318 bytes

80%: 10994 bytes

100%: 47k bytes

Summary

The number of bits used to represent a signal is of critical importance, especially when considering transmitting data for communications.

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