Discrete Fourier transform

Gilbert Strang (1994): "FFT is the most important numerical algorithm of our lifetime"

Included in Top 10 Algorithms of 20th Century by the IEEE journal Computing in Science & Engineering

https://en.wikipedia.org/wiki/Discrete_Fourier_transform
https://en.wikipedia.org/wiki/Fast_Fourier_transform

Discrete Cosine Transform (DCT)

- Real version of Fast Fourier Transform
- Expansion into a cosine Fourier series
- More possible definitions

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

- Inverse transform (up to a scale factor)

$$X_k = \frac{x_0}{2} + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(k + \frac{1}{2}\right)n\right], \quad k = 0, 1, \ldots, N-1.$$

DCT in JPEG

- Encoding of a JPEG image: color transformation, splitting into 8x8 blocks
- Each block is an 8x8 matrix of integers in $[0, 255]$
- Subtract 128 - values in $[-128, 127]$
- Twodimensional DCT:

$$G_{u,v} = \frac{1}{4}\alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y} \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

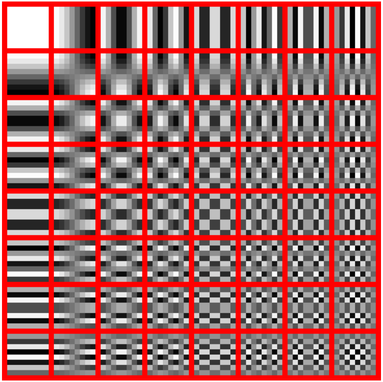
- Normalization factors (for orthonormal transformation)

$$\alpha(t) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } t = 0 \\ 1 & \text{otherwise} \end{cases}$$

- Rounding, other technical steps, ...

DCT in JPEG

We obtain the original 8x8 image as a linear combination of the following basis:



DCT in JPEG

29993 bytes vs. 5872 bytes

DFT in MP3

- Psychoacoustic model - identification of sound components, which are important for human perception of sound/music
- (Windowed) DFT is used to obtain the frequency spectrum
- Subband decomposition
- First song used by Karlheinz Brandenburg to develop the MP3: "Tom's Diner" by Suzanne Vega

Signal Processing

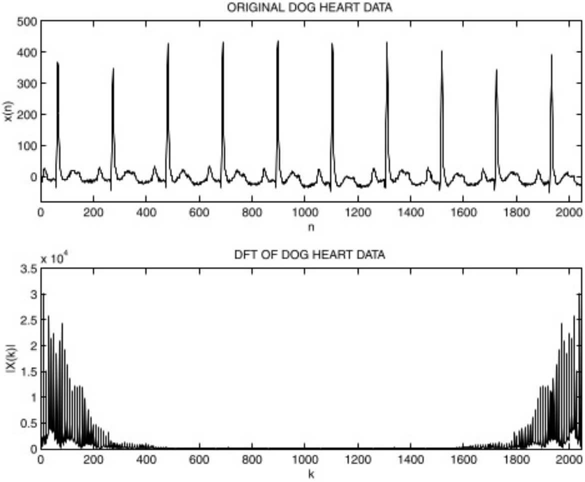


FIGURE 2.2

2048 samples recorded of a dog heart and its DFT coefficients. The magnitudes of the DFT coefficients are shown (see property 1 in Section 2.5.1).

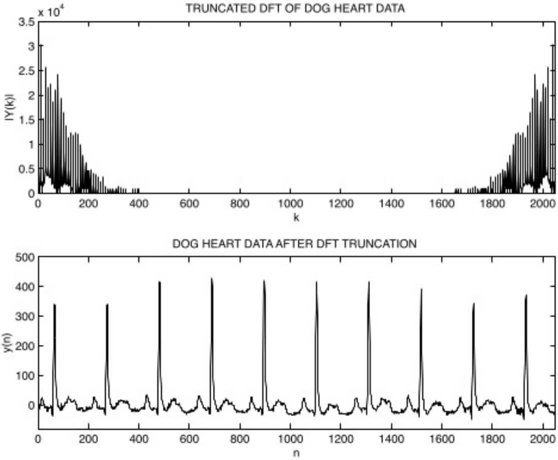


FIGURE 2.3
The truncated DFT coefficients and the time signal reconstructed from the truncated DFT.

Data Compression

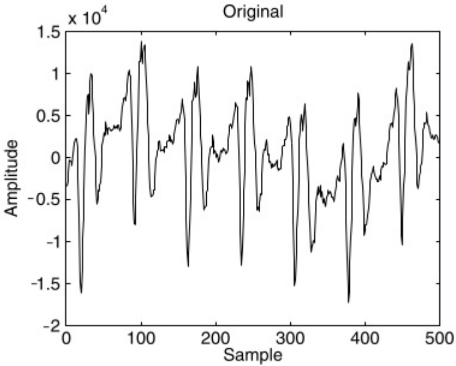


FIGURE 2.25
A piece of an example audio signal, sampled at 32 khz. Shown is the left channel of the stereo signal.

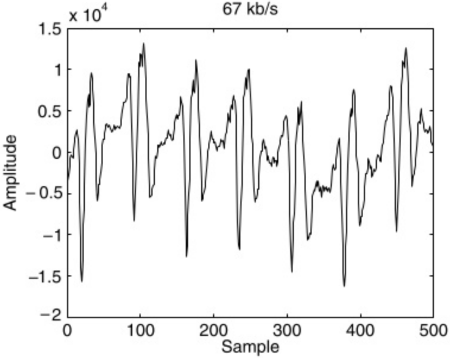


FIGURE 2.26
The stereo audio signal, coded and decoded with 67 kb/s. The left channel is shown.

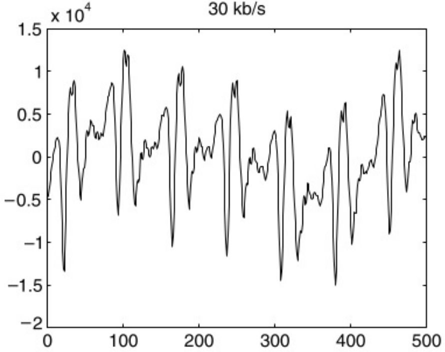
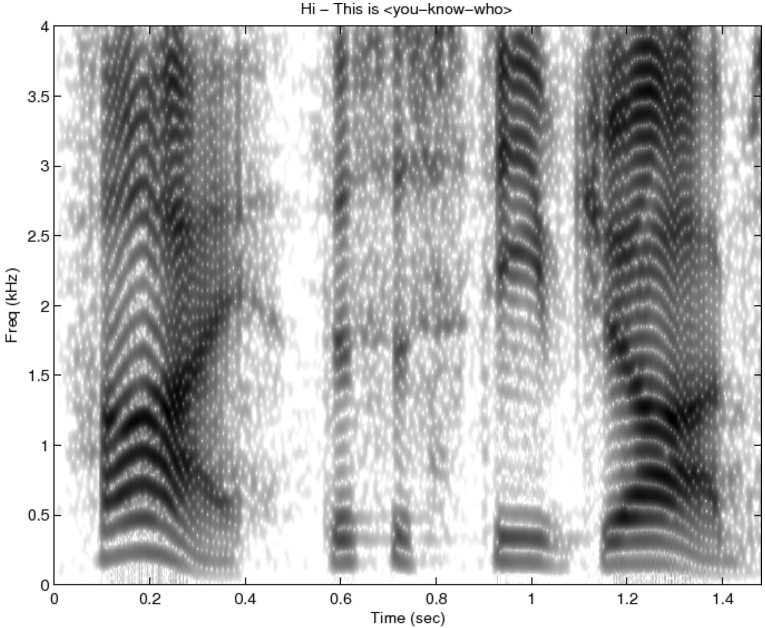


FIGURE 2.27
The left channel of the stereo audio signal, coded and decoded, but with 30 kb/s.

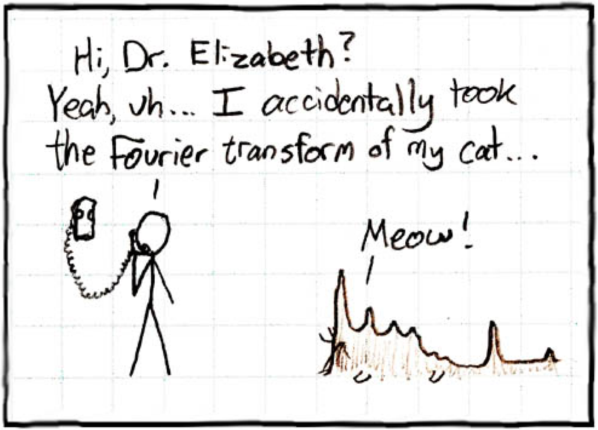


Classic spectrogram of a speech sample

```
[y,fs,bits] = wavread('SpeechSample.wav');
soundsc(y,fs); % Let's hear it
% for classic look:
colormap('gray'); map = colormap; imap = flipud(map);
M = round(0.02*fs); % 20 ms window is typical
N = 2ñextpow2(4*M); % zero padding for interpolation
w = 0.54 - 0.46 * cos(2*pi*[0:M-1]/(M-1));
colormap(imap); % Octave wants it here
spectrogram(y,N,fs,w,-M/8,1,60);
colormap(imap); % Matlab wants it here
title('Hi - This is <you-know-who> ');
ylim([0,(fs/2)/1000]); % don't plot neg.  frequencies
```

Figure 2.3: Time and Frequency represented in a musical score. " . . . zum Raum wird hier die Zeit" (Richard Wagner, "Parsifal"). Reprinted with kind permission of Schott Musik International, Mainz.

Karlheinz Gröchenig (a.k.a. Charlie):
"Foundations of Time-Frequency Analysis"


Hi, Dr. Elizabeth?
Yeah, uh... I accidentally took the Fourier transform of my cat...
Meow!