Electronics and Computer Science
University of Southampton

CM214-COMP2008
Data Communications and Networks

Lossy Data Compression

Karl R. Wilcox
krw@ecs.soton.ac.uk

Objectives

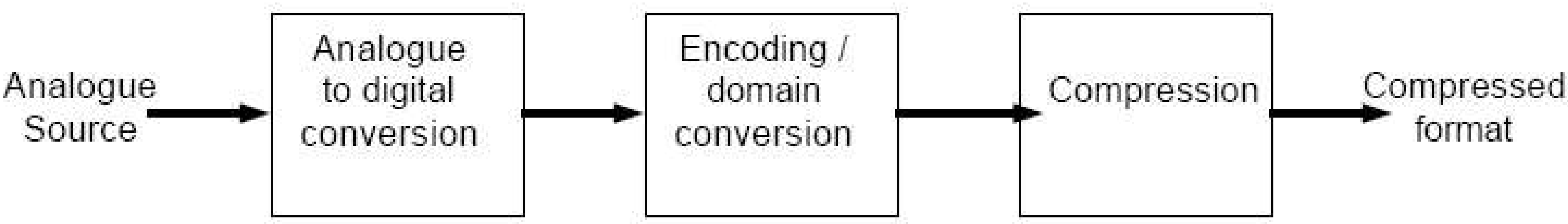
- To understand how “lossy” data compression works
  - Where it is appropriate for use
  - Where it should not be used

- (Peterson & Davie, Section 7.2)

Lossy Compression

- Is only appropriate for data that is, in some way “analogue”
  - Not a general purpose compression algorithm like Lempel-Ziv
- Takes advantage of imperfections in human perception
  - Removes information we don’t notice

Analogue Compression



- We will look at A/D conversion process
- And MP3 / JPEG / MPEG encoding & compression

A/D Conversion

1. Sampling
   - Take “snapshot” of continuously varying signal at discrete times
2. Quantisation
   - Convert snapshot to a discrete value
3. Coding
   - Convert value into ***b***-bit number

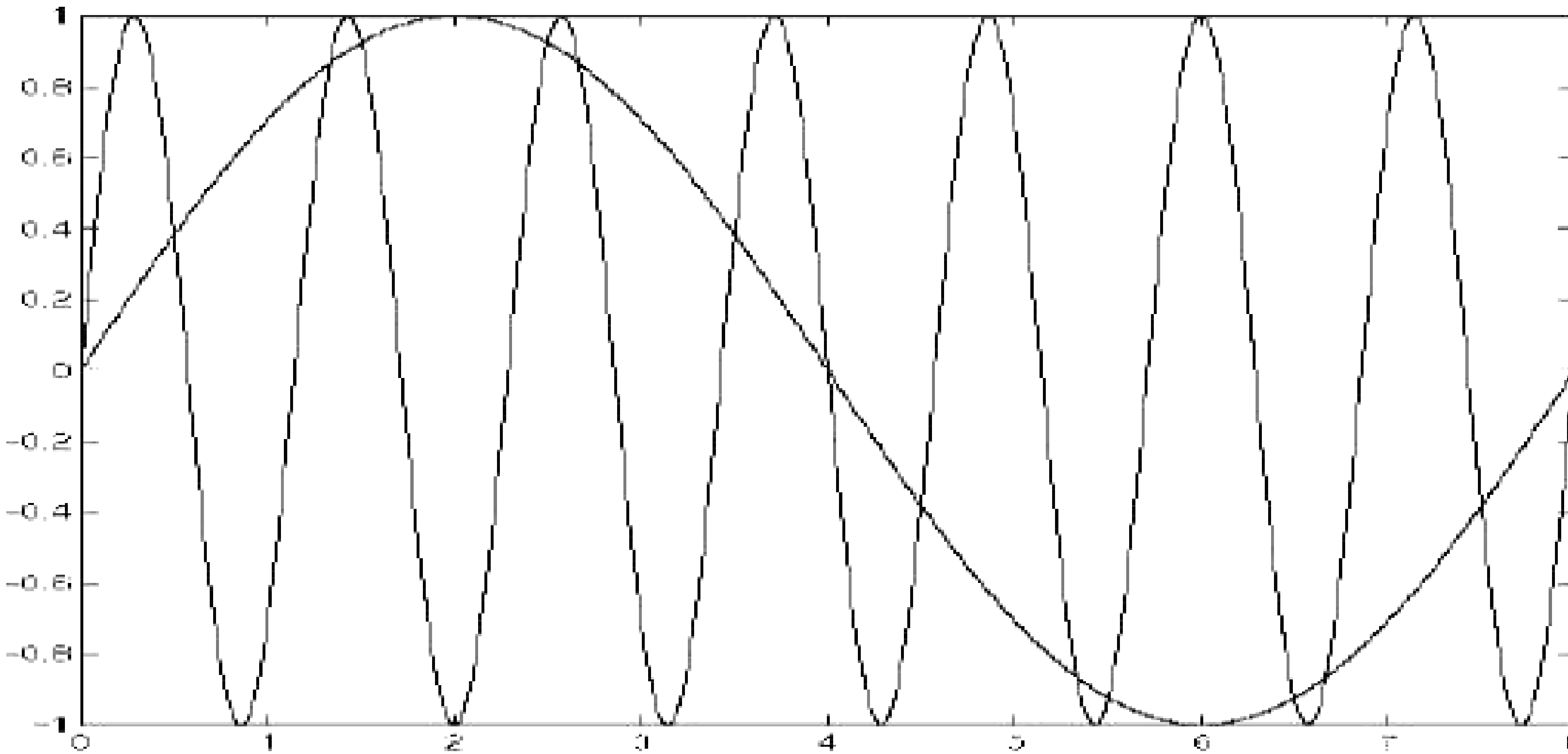
How often? How many bits?

Nyquist Sampling Theorem

"The minimum sampling rate F_s needed to reconstruct an analogue signal from a sampled signal is the Nyquist rate, $2F_{max}$"

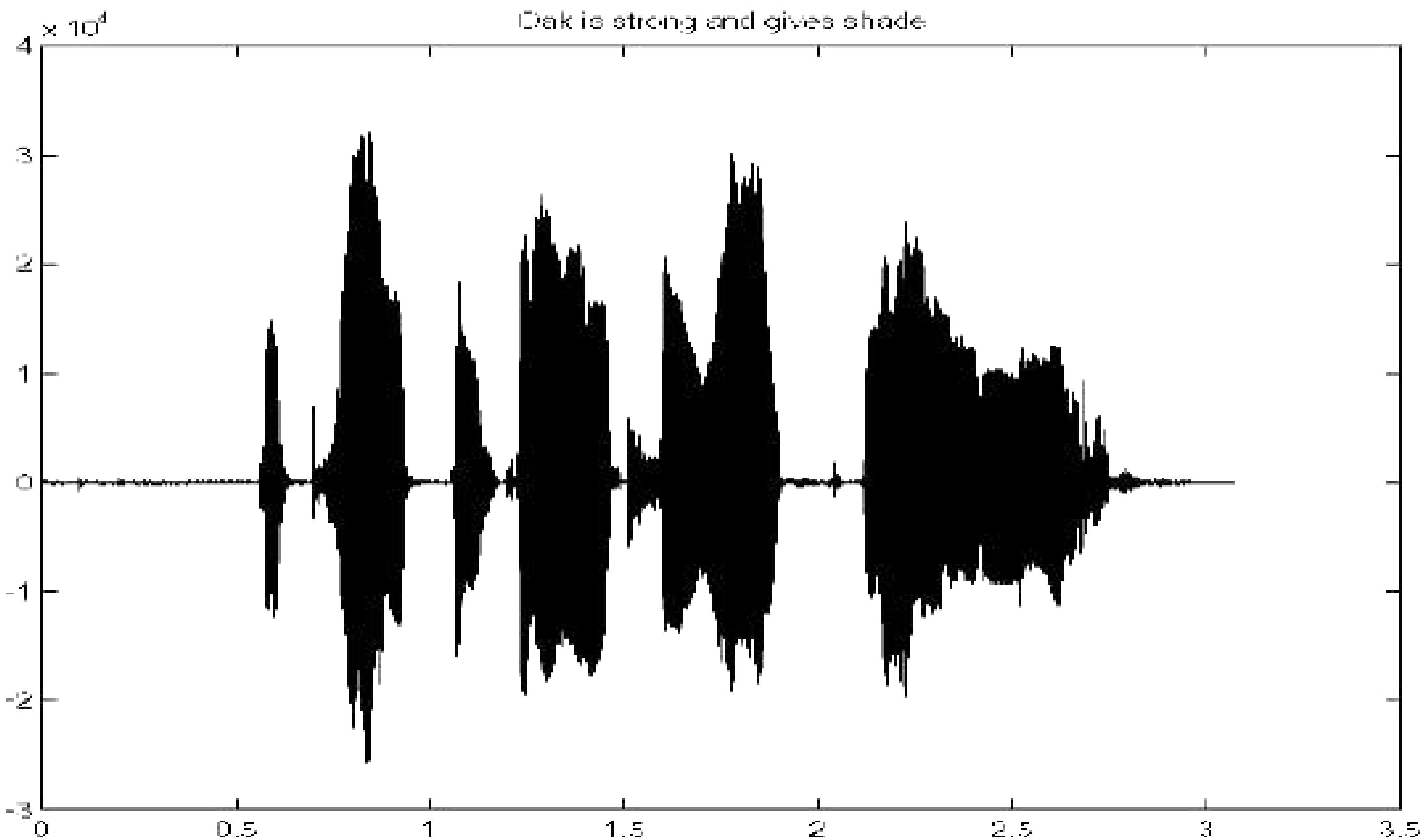
i.e. You must sample twice as fast as the most rapidly varying signal

Undersampling



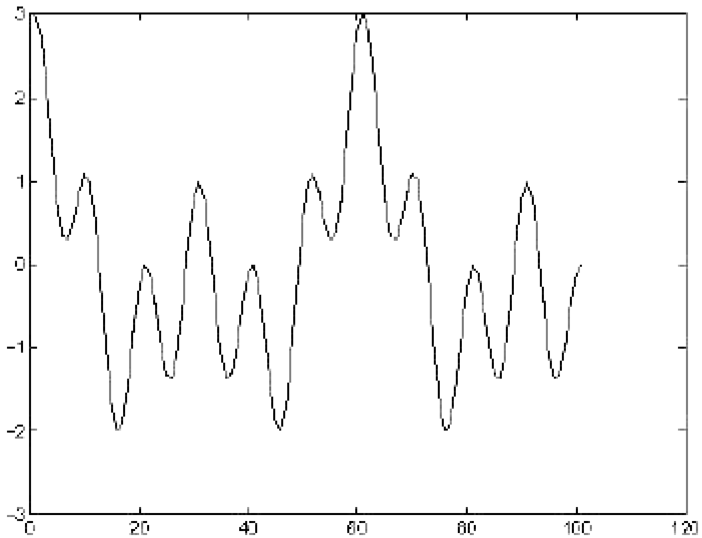
Sample taken every second (1Hz) of signal varying at 7/8Hz appears as 1/8Hz due to undersampling or “aliasing”

Real Speech

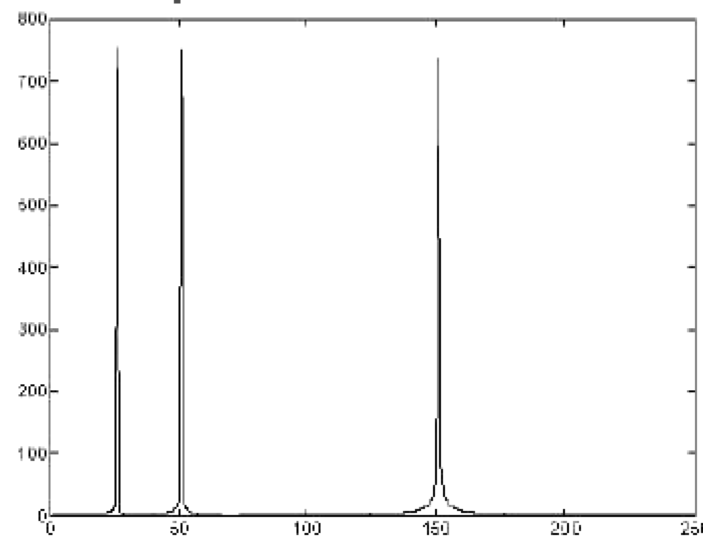


Domain Conversion - 1

- For compression we convert from time domain to frequency domain
  - From set of time varying signals to set of sine waves of various frequencies



Time Domain



Frequency Domain

Domain Conversion – 2

- Conversion assumes signals look like lots of superimposed sine waves
  - All "natural" signals do!
  - (corollary –only works for natural signals)
- From Nyquist, we need to sample at twice the rate of the highest frequency sine wave

Fourier Transform

- Time to frequency domain conversion is achieved using the Fourier Transform
  - If signal is: Discrete Fourier Transform is:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{-j2\Pi kn/N} \qquad X_k = \sum_{k=0}^{N-1} x_n e^{j2\Pi kn/N}$$

- In reality, use FFT algorithm from library routine or hardware implementation (e.g. DSPs in cameras & phones)

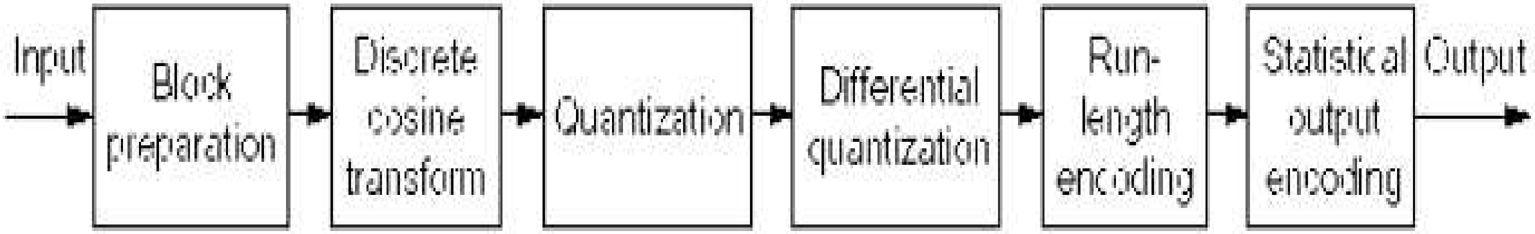
Why Frequency Domain?

- Because easier to identify features humans cannot perceive
  - E.g. For MP3 audio we can remove
    - All frequencies outside human range
    - “time domain” masked sounds (quiet sound at same time as a loud one)
    - “frequency domain” masked sounds
  - Based on empirical perception tests of humans **NOT** on mathematical algorithms

JPEG Compression

- Similar to MP3, uses empirical & physiological knowledge to remove information that humans cannot perceive
  - More conventional techniques then compress resulting data
- Can control amount of data “loss”
- Too much results in no tonal variation

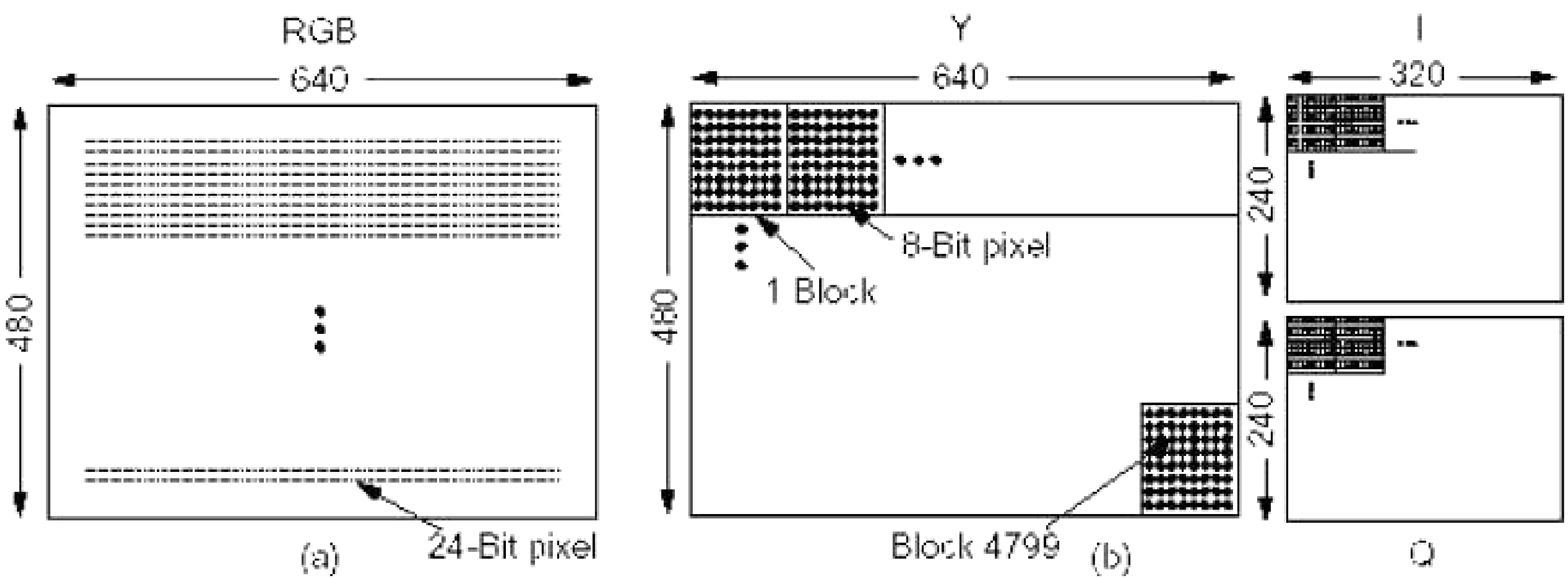
JPEG Process



- We will look at each stage in turn

Block Preparation

- Take a 24 bit RGB image & convert to “luminance” & “chrominance”
  - Latter loses information through averaging



Conversion to Luminance & Chrominance

- From the RGB pixel values we calculate Y, I & Q values
  - (see previous slide)
- $Y = 0.30R + 0.59G + 0.11B$
- $I = 0.60R – 0.28G – 0.32B$
- $Q = 0.21R – 0.52G – 0.31B$
  - Why these values?
    - Physiological characteristics of *human* eyes

Block Conversion

- Each 8 x 8 block goes through Discrete Cosine Transform (DCT)
  - Similar to Fourier Transform
  - (See P&D for the maths)
- The calculated DCT coefficients then quantised according to a quantisation table

Compression

DCT Coefficients

150	80	40	14	4	2	1	0
92	75	36	10	6	1	0	0
52	38	26	8	7	4	0	0
12	8	6	4	2	1	0	0
4	3	2	0	0	0	0	0
2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized coefficients

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	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

Quantization table

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

Quantisation

- The quantisation tables can be adjusted by the user
- The table is then run-length encoded in a zig-zag pattern
- These results then Huffman encoded
  - This is how JPEG provides variable compression rates
  - And why we can't specify required compression ratio at outset

MPEG Compression

- MPEG (Motion Picture Experts Group) includes video & audio streams
  - (Audio is MP3)
  - Synchronised to embedded time code
- Non-symmetrical encoding / decoding times
  - Need to identify scene changes
  - Need to identify moving features

MPEG Frame Types

- Video has four different frame types
  - Intracoded (I)
    - Full frame encoded in a JPEG like form
  - Predicitive (I)
    - Difference from previous frame
  - Bidirectional (B)
    - Difference from previous & next frames
  - DC coded
    - Block average for fast forward (optional)

MPEG Transmission

- Frames are sent such that dependent frames are transmitted after the frame they depend on
  - Original frames – I B B P B B I
  - Transmit as – I P B B I B B
- Difficult in real time
  - Video conferencing often just frame by frame JPEG compression (M-JPEG)

Summary

- Lossy compression applies to the data transmitted over the network at application level
- Appropriate only for “analogue” data
- Can require high processing overhead
- Can result in very high compression ratios