Sound 3

Sound Files and Formats

Sound compression (I)

Compression of sound data requires different techniques from those for graphical data. Requirements are less stringent than for video

- data rate for CD-quality audio is much less than for video, but still exceeds the capacity of typical broadband connections
  - Data rate is 44100*2*2 bytes/sec=176,400bytes/s=1.41Mbits/sec
- 3 minute song recorded in stereo occupies 31Mbytes

Sound is difficult to compress using lossless methods

- complex and unpredictable nature of sound waveforms

Different requirements depending on the nature of the sound

- speech, music, natural sounds
- …and on nature of the application

Sound compression (II)

A simple lossless compression method is to record the length of a period of silence

- no need to record 44,100 samples of value zero for each second of silence – effectively a form of run-length encoding
- in reality this is not lossless, as "silence" rarely corresponds to sample values of exactly zero; rather some threshold value is applied

Difference between how we perceive sounds and images results in different lossy compression techniques for the two media

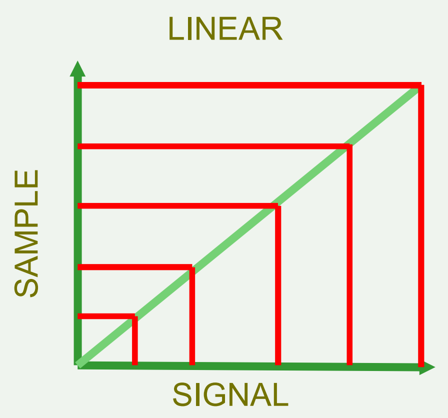
- high spatial frequencies can be discarded in images
- high sound frequencies, however, are highly significant

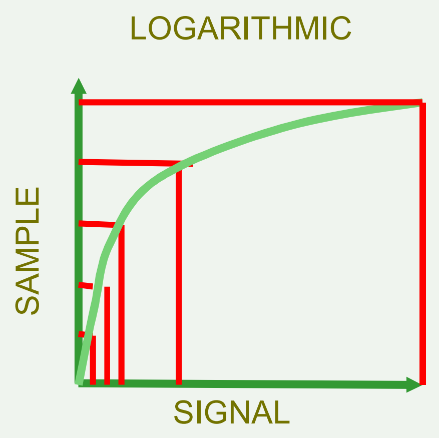
So what can we discard from sound data?

Sound compression (III)

Our perception of loudness is essentially logarithmic in the amplitude of a sound. Nonlinear quantization techniques provide compression by requiring a smaller sample size to cover the full range of input

Remember...





Adaptive Differential Pulse Code Modulation

This is based on storing the difference between samples

- related to inter-frame compression of video BUT less straightforward

Audio waveforms change rapidly so no reason to assume that differences between samples are small

- unlike video, where two consecutive frames may be very similar

Adaptive Differential Pulse Code Modulation

Differential Pulse Code Modulation (DPCM) computes a predicted value for a sample based on preceding samples

- stores difference between predicted and actual sample
- difference will be small if prediction is good

ADPCM extends this by using an adaptive step (sample) size to obtain further compression

- large differences quantized using large steps, while small differences are quantized using small steps
- efficient use of bits, taking account of rate of change of signal

Linear Predictive Coding

A radical approach to compression of speech – it uses a mathematical model of the vocal tract. Instead of transmitting speech as audio samples, the parameters describing the state of the vocal tract are sent.

- At the receiving end these parameters are used to reconstruct the speech by applying them to the same model
- Achieves very low data rates: 2.4kbps (usable on a poor quality telephone line)
- Speech has a machine-like quality
  - suitable for accurate transmission of content, but not faithful rendition of a particular voice
- Similar in concept (but rather more complicated!) to vector-coding of 2D and 3D graphics

Perceptually based compression (I)

Is there data corresponding to sounds we do not perceive in a sound sample? If so, then we can discard it, thereby achieving compression

- Sound may be too quiet to be heard
- One sound may be obscured by another sound
- Threshold of hearing
  - varies nonlinearly with frequency
  - very low or high frequency sounds must be much louder than mid-range sounds to be heard
  - we are most sensitive to sounds in the frequency range corresponding to human speech
- Sounds below the threshold can be discarded
  - compression algorithm uses a psychoacoustical model that describes how the threshold varies with frequency

Perceptually based compression (II)

Loud tones can obscure softer tones that occur at the same time

- not just a function of relative loudness, but also of the relative frequencies of the two tones

This is known as *masking*

- modification of the threshold of hearing curve in the region of a loud tone
- sounds normally above the unmodified threshold are no longer heard

Masking can also hide noise as well as other tones

- courser quantization (smaller sample size = fewer bits) can be used in regions of loud sounds

Perceptually based compression (III)

Actually making use of masking in a compression algorithm is very complicated

- MPEG standards use this sort of compression for the audio tracks of video
- MP3, which is MPEG-1 Layer 3 audio, achieves 10:1 compression
- AAC, from MPEG 4, is even better: approx 128Kbit/second
  - Used by Itunes and QuickTime 6.

Sound files and formats

There are many sound file formats

- Windows PCM waveform (.wav)
  - a form of RIFF specification; basically uncompressed data
- Windows ADPCM waveform (.wav)
  - another form of RIFF file, but compressed to 4 bits/channel
- NeXT/SUN file format (.snd, or .au)
  - actually many different varieties: header followed by data
  - data may be in many forms.
- MPEG format (includes MP3, AAC)
  - has various different forms of compression
- OGG open source format (video too)
  - Wav, mp3 and ogg most likely supported by web browsers
- QuickTime, AVI and Shockwave Flash
  - can include audio as well as video

Examples of sizes

A 2.739 second stretch of sound, digitised at 22050 samples/second, 16 bits, mono takes

- 120,856 bytes as a .snd
- 120,876 bytes as an uncompressed .wav
- 30,658 bytes as a compressed .wav
- 5,860 bytes as RealAudio
- 321,736 bytes as ASCII text

…and if you listen hard you can hear the difference!

Another example: speech

Original (64000 bps) This is the original speech signal sampled at 8000 samples/second and u-law quantized at 8 bits/sample.

ADPCM (32000 bps) This is speech compressed using the Adaptive Differential Pulse Coded Modulation (ADPCM) scheme. The bit rate is 4 bits/sample (compression ratio of 2:1).

LPC10 (2400 bps) This is speech compressed using the Linear Predictive Coding (LPC10) scheme. The bit rate is 0.3 bits/sample (compression ratio of 26.6:1).

Music (I)

So far we have considered digitizing sounds recorded from the real world. To store and transmit natural sounds it is generally necessary to use digitized recordings of the real sounds. However, we have seen that speech can be specified as states of the vocal tract

- Music can also be specified: musical scores
- We can send a piece of music to someone as either
  - a recording of an actual performance
  - some notation of the score and the instrument to play it, provided the receiver has some means of recreating the music from the score e.g. they can play it on a piano etc
- This is akin to bitmapped versus vector-based graphics

Music (II)

Automated music production

- pianolas: player pianos
- barrel organs

Synthesizers

- electronic instruments for producing many different sounds, usually controlled via a keyboard

Many sound cards in PCs can synthesize sounds

Automation requires an appropriate language for specifying sounds

- musical instrument
- note and its duration etc

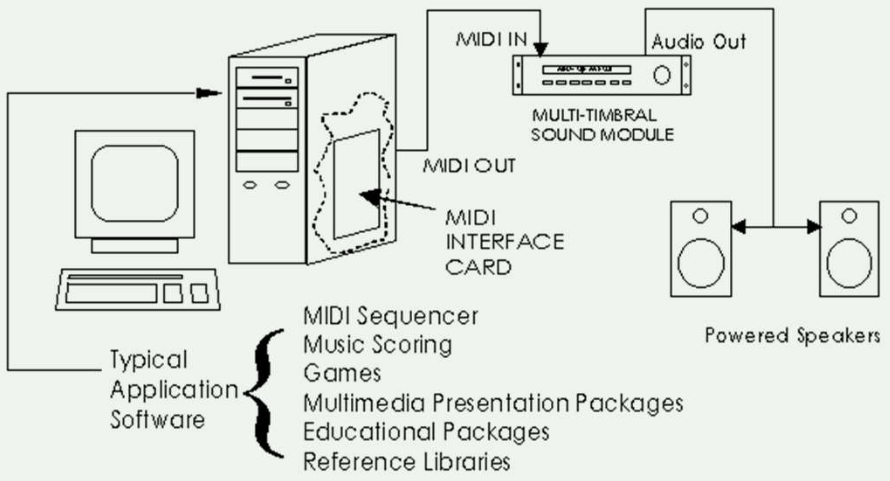
MIDI does this and more...

MIDI (I)

Musical Instruments Digital Interface

Resynthesis instead of reproduction

- like a player piano, instead of a CD player
- consists of commands which result in notes being played (synthesised)



MIDI (II)

Messages sent can define

- note on/off, pitch of note, pitch bend, control change, timbre

Synthesisers which use FM or wavetables can also be controlled

For more information see http://www.midi.org

Data rate is typically 0.1% of high quality digitised sound

- but no guarantees that the music produced is exactly as the composer intended
- instrument voices will vary with quality of sound hardware
- different instruments may actually be used

Not useful for vocal music

End of Lecture