Audio Processing

An Introduction for Students of Computer Science

Catania, 5–9 July 2004

Jörn Loviscach
jlovisca@informatik.hs-bremen.de
Hochschule Bremen (University of Applied Sciences)
Germany

Agenda for the Week

Today: foundations of sound, digital audio, and real-time audio processing

Tuesday: dynamics

Wednesday: delay, time modulation, reverb

Thursday: filters, Fourier and z transform

Friday: generators, sound synthesis

Special wishes? Just ask!

Agenda for Today

- What Is It All About?
- What Is Sound?
- Digital Audio, PCM Data Formats
- Waveform Display and Level Measurement
- Generation and Output of Waves
- Quantization Issues of Voltage and Time
- Real-Time Issues: Blocks and Threads
- Metrics of Loudness

What Is It All About?

- recording and mastering:
  demo of Steinberg Cubase
- sound synthesis:
  demo of Native Instruments Pro-53
  and Arturia Moog Modular V
- audio data compression (MPEG etc.)
- audio restoration (denoiser etc.)
- suppressing echo and room noise
  (mobile phones etc.)
- 3D sound
- ...

What Is Sound?

- pressure waves, mostly in the air
- audible frequency range 20...20,000 Hz
- audible sound level at 1 kHz in air:
  20 µPa (threshold) ... 200 Pa (painful);
  atmospheric pressure: 100,000 Pa
- both level and frequency are perceived on logarithmic scales
- standard frequencies 440 Hz and 1 KHz; musical intervals; demo with PD
- decibel: 10 dB = power ratio of 10:1,
  20 dB = pressure or voltage ratio of 10:1; demo with PD

Digital Audio, PCM Data Formats

- sample voltage at fixed rate (CD: 44,100 Hz); store every sample as a number (CD: 16 bit integer value per stereo channel)
- 24 bit / 96 kHz gets increasingly popular; from 32 bit on a floating point representation will be more efficient; but in future we may be using bitstreams (1 bit / x MHz) anyway.
- Accessing PCM audio data streams with javax.sound.sampled: signed/unsigned, Little/Big Endian

Waveform Display and Level Measurement

- real-time sound input with javax.sound.sampled
- displaying a waveform (JFrame, paint, repaint, double buffering)
- RMS (root mean square) level: DC voltage to generate the same power in the average
- Peak level: maximum magnitude of voltage over some period
- Lab task: Implement a Peak/RMS VU meter in Java

Generation and Output of Waves

- clips in javax.sound.sampled
- sine waves with given frequency and amplitude by Math.sin
- sawtooth, square, and pulse waves
- generating broad-band noise with random numbers
- Note: All of this is neither efficient nor does it sound good. More on that later.
- reading sound files into clips

Quantization Issues of Voltage and Time

- reducing the number of bits per sample leads to noise (strongly colored noise if the resulting number of bits is small)
- using only every n-th sample to reduce the sampling frequency leads to aliasing: mirrored frequencies; demo: spreadsheet
- Perfect reconstruction of a wave is possible and is only possible if it does not contain any frequencies $\geq f_s/2$. Shannon-Nyquist-Kotel'nikov
- Lab task: Experiment with reducing the number of bits and samples

Real-Time Issues: Blocks and Threads

- DSP-based or fully hardwired audio processing: compute sample per sample PC: lots of other things going on, always process a block (e.g., 1000) of samples and use ring buffers to keep things streaming
- latency: The larger the block size, the larger the delay between input and output.
- spawn a thread to separate audio processing from GUI etc.
- demo with thread, block-orientation in

Metrics of Loudness

- Lab task: Write a Java program that interactively registers your hearing threshold at different frequencies.
- Sound Pressure Level: 20 µPa is 0 dB SPL, regardless of frequency (i.e., SPL is **no** metric of loudness)
- Phon: the dB SPL number of a tone of 1 kHz that is perceived as equally loud (But what is 2 Phon compared to 1 Phon?)
- Sone: 40 dB SPL @ 1 kHz := 1 Sone; a sound that appears twice as loud has 2 Sone, etc. (No logarithmic scale!)