Audio Processing Joren Six

25 March 2011

Goal

Cover the basic principles of doing stuff with audio.

Outline

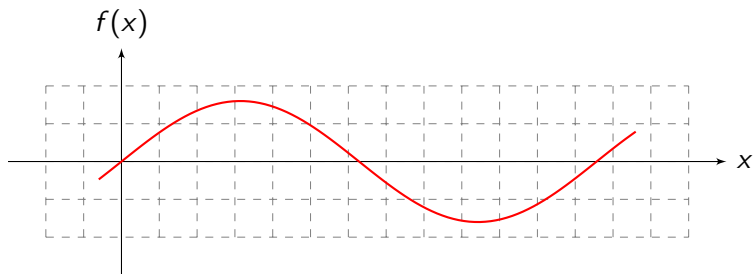# Basics - Analog Audio

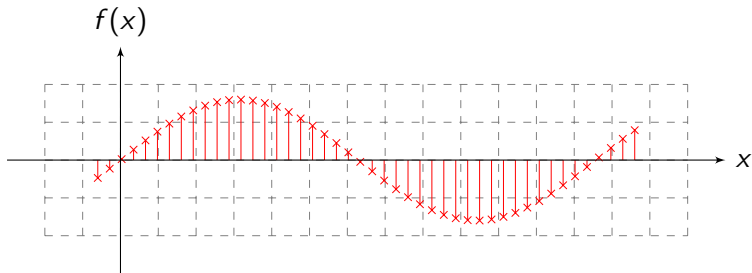**Figure:** Continuous wave

Basics - Digital Audio

Figure: Sampled wave

Basics - Digital Audio - Samplerate

Listing 1: A sampled sine wave buffer

```
1 double sampleRate = 44100.0;
  double frequency = 440.0;
  double seconds = 2.0;
  float[] b = new float[seconds * sampleRate];
  for (int sample = 0; sample < b.length; sample++) {
6     double time = sample / sampleRate;
      b[sample] = 0.8 * Math.sin(twoPiF0 * time);
  }
```

Basics- Digital Audio - Bit depth

Listing 2: A sampled sine wave buffer

```
  final byte [] byteBuffer = new byte [b.length * 2];
2 int bIndex = 0;
  for (int i = 0; i < byteBuffer.length; i++) {
      final int x = (int) (b[bIndex++] * 32767.0);
      byteBuffer [i] = (byte) x;
      i++;
7     //unsigned right shift
      byteBuffer [i] = (byte) (x >>> 8);
  }
```

float in $[-1.0, 1.0]$ to 16bit signed little endian PCM.
Multiply each sample with $\lfloor (2^{16} - 1)/2 \rfloor = 32767$

TarsosDSP - What

TarsosDSP is a collection of JAVA classes to do simple audio
processing. Bascially chainable operations on float or byte buffers.

TarsosDSP - Contents

- ▶ Filters: low pass, high pass
- ▶ Pitch detectors: YIN and MPM
- ▶ FFT
- ▶ WAV file writer
- ▶ . . .

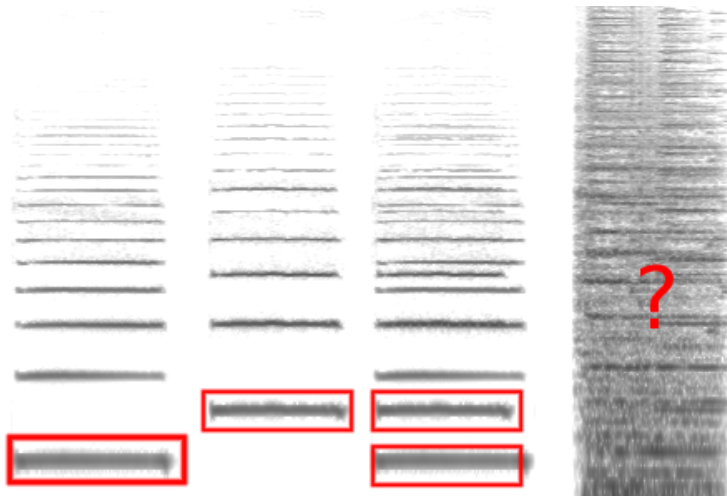# TarsosDSP - Sound Detection

Demo

TarsosDSP - Pitch Detection

Figure: Pitch detection
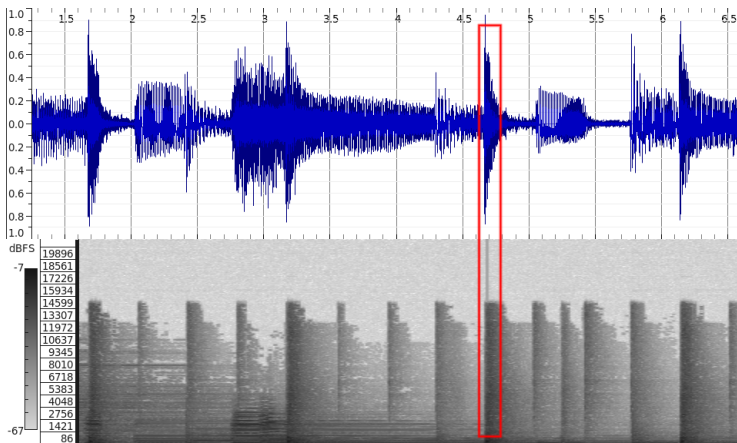
TarsosDSP - Percussion Detection

Figure: Percussion onset detection

TarsosDSP - FFT to MIDI

Demo

TarsosDSP - Contact

Figure: https://github.com/JorenSix/TarsosDSP

Conclusion and Questions

joren.six@hogent.be