1st Assignment: The Speech and Audio Signal

Speech Technologies

2016

The objective of this assignment is to have a first view of the speech signal. You should observe how different sounds have different time and frequency features. You should learn concepts as: voiced/unvoiced, pitch, formants. You should be able to use and develop tools to record, play and analyze speech.

Please, report the results of the assignments using a 4-pages paper format. You can use for instance the templates that you can find in `http://www.icassp2016.com/papers/PaperKit.html#Templates`. Include in your final report relevant screen shots and the evaluation results of your pitch detector. Please, upload a **pdf** file, original *wav* file and label files, and the source code of the pitch detector.

Sampling rate

1. Download and execute **wavesurfer** or a similar program.[1]
2. Record a complete sentence (around 3 seconds), mono, 16 bits, 44kHz. We suggest using a close-talk microphone to avoid noise. You can select the recording parameters using the *properties* option, right mouse-button. Save it as a `.wav` file.
3. Look at the spectrum of several speech sounds: what is the *speech bandwidth*?
4. Downsample the file to rates 16kHz, 12kHz, 8KHz and 4KHz. Listen to the five files and compare the quality. For downsampling the file you can use **SoX**: it allows changing the sampling rate and the sample format, demultiplexing stereo files, filtering, and many other effects. For instance, to change the rate to 8KHz, just enter:

   ```
   sox myfile.wav -r 8k myfile_8k.wav
   ```

   You can also use **matlab** or **octave**, or even wavesurfer, using the *convert* command in the *transform* menu.

Sounds: voicing and formants

Analyze the diferent sounds

1. Use the *transcription* pane to label at least 10 phones (properties, extend boundaries)
2. Which sounds are approximately stationary and which ones have different parts? Look at both waveform and spectrogram.

[1] In some Linux distributions you should execute `padsp wavesurfer`; **padsp** redirects the I/O of programs that use OSS, and outdated sound interface.

3. Look at the waveform: which sounds have higher energy in the sentence?

4. Identify at least three voiced and three unvoiced sounds in your sentence.

5. Identify at least 5 different vowels. Which are the values of the first 2 formants? For doing this, add the *formants* pane that plots spectrogram and formants. Observe in which part of the spectrogram are the formants. For the 5 vowels, plot the spectrum and look how the formants values correspond to the *resonances* in the spectrum. (Include the plots in your report)

Pitch (F0)

1. Choose an instant where signal is clearly voiced and compute the pitch from time domain and also from the frequency domain.

2. Add the *pitch contour* pane and report the lowest and highest pitch in the sentence. (You should disregard low and high values that are *wavesurfer* errors).

3. Implement pitch detection method (ex: correlation, mdf, cepstrum) and a voiced/unvoiced detector in any language (C++, python or MATLAB). You can also include pre- and post-processing to improve the results, or a combination of systems. Use the following database to test the performance of your algorithms: `http://www.cstr.ed.ac.uk/research/projects/fda`, and report v/uv errors, uv/v errors, gross pitch errors (>20 %) and MSE (Mean Squared Error) of fine pitch errors. The database is also available in Atenea in wav format as well as an evaluation program in C++.