Speech Processing Project 3 Writeup

Matt Ruffner

April 5, 2019

1 Overview

In this project, the Pitch Synchronus Overlap and Add (PSOLA) algorithm was implemented in MATLAB in order to achieve simultaneous pitch shifting and time scaling. By using the Audio Toolbox (only available for R2019a), realime pitch shifting and time scaling was also implemented. Two example inputs and system outputs are included: `jordan.wav`, `dig.wav`, and `jordan-out.wav`,`dig-out.wav`. Both of these files were processed with parameters `ts=1.0` and `ps=[2.0 0.5]`.

2 Approach

At first, I just implemented time scaling with overlap and add by itself. Then I experimented with pitch modification and achieved pitch scaling along a linear pitch axis. However, it did not duplicate frames as needed so a pitch shifted sound clip changed duration as a result. After some online research, I found pseudocode outlining the PSOLA algorithm [1]. This was then implemented in MATLAB and turned into a function. This function is outlined in Section 3.

3 Usage

The time scaling and pitch modification is done by the function `sigout=tpss(sigin,fs,ts,ps)`. This function can be found in the file `tpss.m`. Other source files included in the submission are the main runner script `p3.m`, and the real-time modification script `p3realtime.m`. This project has dependencies on both the MATLAB Audio Toolbox and the sap-voicebox toolkit.

In order to run the `tpss` function on your own input `.wav` file, edit the `audioread` command on line 5 of `p3.m` to point to your input file. It should be noted that the `tpss` function asserts that the input is mono. If you load a stereo file, the assertion will fail and the function will not run. Pitch and timescale modification parameters are clearly defined on lines 9 and 11 of `p3.m`.

In order to run the real-time script, no modifications should need to be made to the script. In order to edit the parameters of the time scaling and pitch shifting, you need to edit the arguments passed to the call to `psola` function handle on line 34 of `p3realtime.m`. This simple example of real-time audio processing in MATLAB was adapted from an Audio Toolbox example script.

4 Limitations

Detecting unvoiced speech is implemented on lines 26-41 in `tpss.m`, however it is left commented out currently because the real-time demonstration does not work when it is enabled. If you are just doing static tests you can un-comment these lines and notice and improvement in quality. As stated earlier, this code does not implement the extra LPC layer onto the PSOLA algorithm.

References

[1] *DAFX; digital audio effects, 2d ed.* John Wiley & Sons Ltd, 2011.