Epoch Based Continuous Time and Pitch Scaling

Report on the GUI Developed Using Python

S Vinu Sankar
Indian Institute of Technology Gandhinagar (IITGN)
Palaj, Gandhinagar,
Gujarat-382355
vinu.sankar@iitgn.ac.in

30th June 2017

Advisor: Prof. Chandra Sekhar Seelamantula

Acknowledgement

I am highly thankful to Prof. Chandra Sekhar Seelamantula for providing me this wonderful opportunity and guiding me throughout my internship. I would also like to thank Mr. Sunil R for the great effort he has put in to help me complete my work. I am also thankful to him for providing me with the MATLAB code for the EBCTPS algorithm and explaining it to me.

REPORT

The Algorithm

Epoch Based Continuous Time and Pitch Scaling (EBCTPS) Algorithm is used for scaling speech signals. The algorithm detects **Glottal Closure Instants** (GCIs or epochs) in the input speech signal and processes it to produce the scaled speech signal. For making the speech longer, more signal fragments are overlapped at the GCIs. For making it shorter, some signals are cut at the GCIs. Pitch scaling is performed by resampling time scaled signals. For dynamic scaling, the speech signal is discretized into one second long fragments, scaled individually and then combined together using the Hanning overlap window.

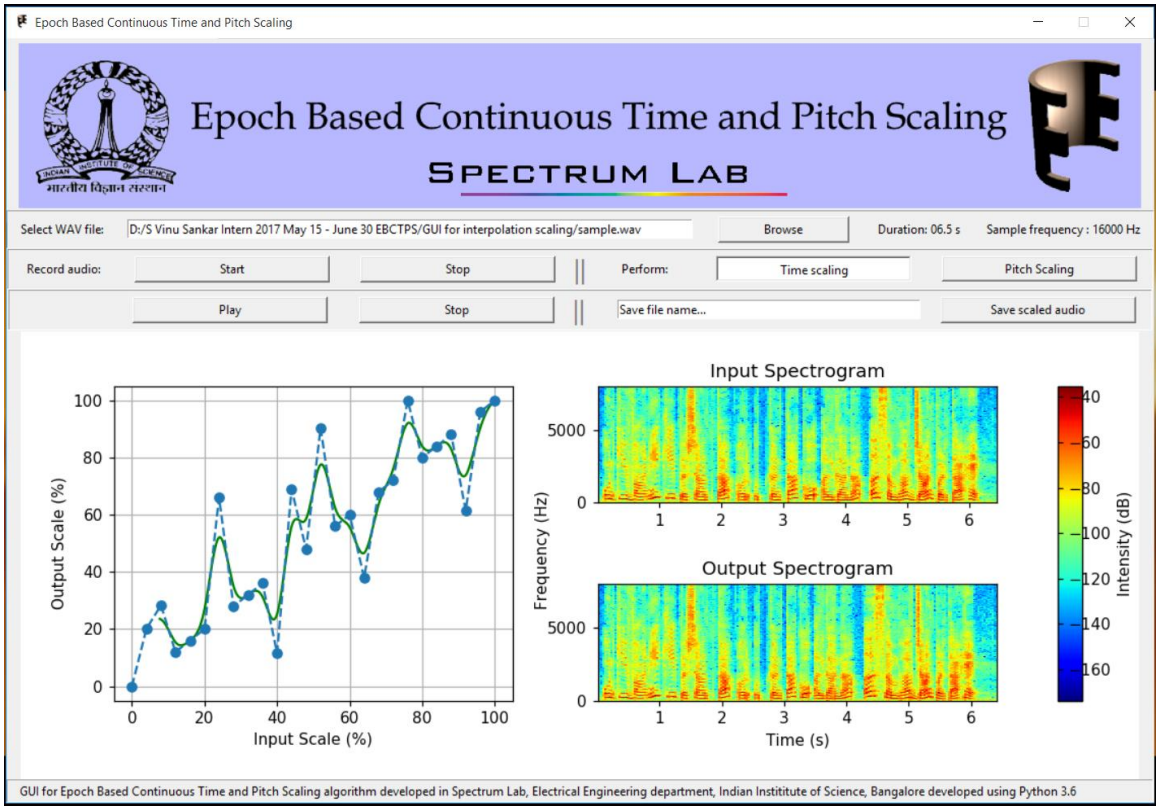
The Program

The program is coded using **Python 3.6** in the Spyder 3.1.2 IDE. The main algorithm is coded in the *EBCTPS.py* script and the Graphical User Interface (GUI) is coded in the *GUI.py* script. The former script has three main functions:

1. *Epoch()*: Finds out GCIs in the input speech signals and returns them in an array.
2. *ETS()*: Performs time scaling for an input speech signal.
3. *EPS*(): Performs pitch scaling for an input speech signal.

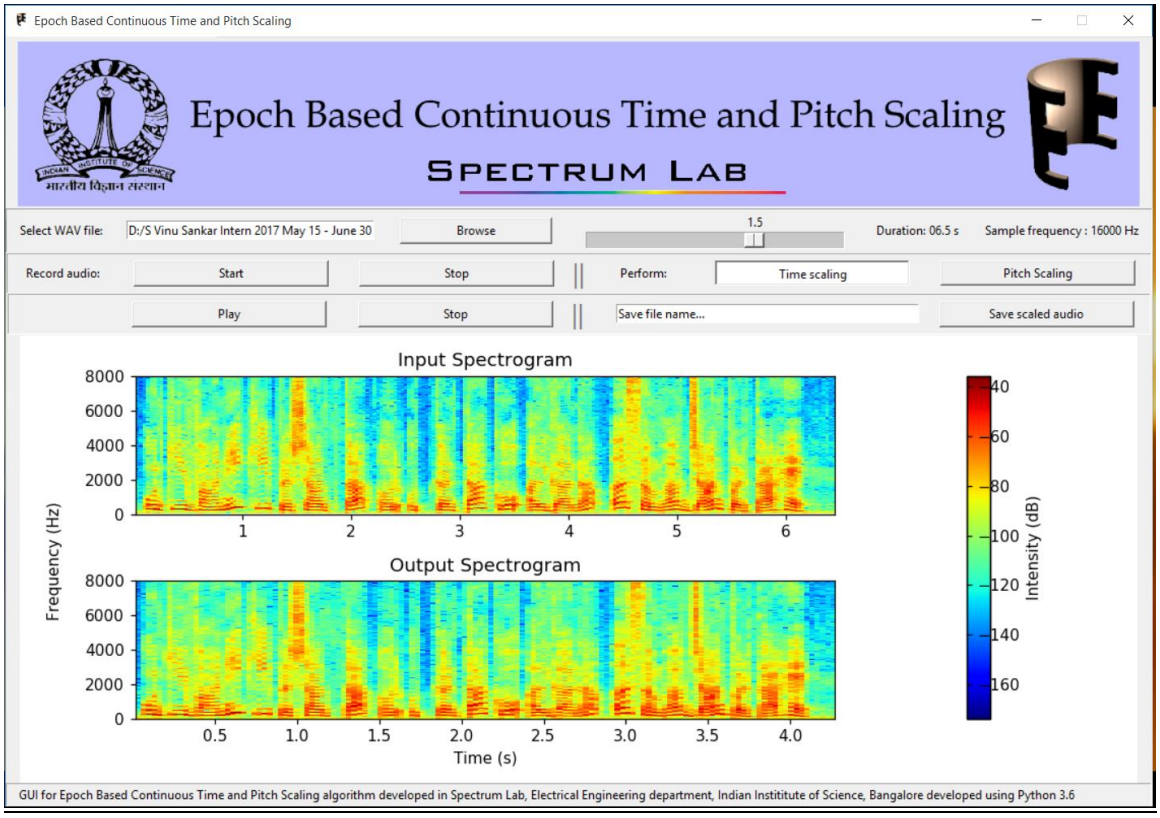
1. GUI for Scaling with Cubic B-Spline Interpolation

The script GUI.py performs the *discretization and reconstruction* of the input and processed signals respectively. The GUI can be used for browsing the audio .wav file from the system or *recording* an audio .wav file using an inbuilt microphone for scaling. One can choose to perform time scaling or pitch scaling. The dynamic scaling is performed based on the scale the user inputs using the live *cubic B-spline graph* plotted on the GUI. Further, it also plots the *spectrogram* for both the input and the scaled audio. The scaled audio signal, by default, will be saved in the same location as that of the input signal on pressing the 'Save' button. The user also has the option to name and save the scaled audio signal and it will be saved into the 'Saved_audios' folder in the root location of the scripts.



2. GUI for Scaling with Constant Scaling

The script GUI.py constructs the GUI. The GUI can be used for browsing the audio .wav file from the system or *recording* an audio .wav file using an inbuilt microphone for scaling. One can choose to perform time scaling or pitch scaling. The dynamic scaling is performed based on the scale the user inputs using the slide bar on the GUI. Further, it also plots the *spectrogram* for both the input and the scaled audio. As the scaling slid bar is modified, the scale of the audio playing keeps on changing dynamically. The scaled audio signal, by default, will be saved in the same location as that of the input signal on pressing the 'Save' button. The user also has the option to name and save the scaled audio signal and it will be saved into the 'Saved_audios' folder in the root location of the scripts.



Features:

- Dynamic scaling can be implemented efficiently
- Easy-to-use and attractive GUI
- Gives details about speech signals
- Input and output signals can be compared using spectrograms
- Audio can be recorded within the GUI
- Scale input through continuous cubic B-spline

Modules Required

For making this program in Python, some extra modules has to be downloaded and installed separately. These modules (latest versions as of May 2017) are listed below.

- Numpy

  For easy scientific operation on lists

  http://www.numpy.org/

  import numpy

- Scipy

  For filtering, interpolation and resampling

  https://docs.scipy.org/doc/

  import scipy

- Matplotlib

  For plotting graphs

  https://matplotlib.org/

  import matplotlib

- PIL

  For loading title image in the GUI

  http://www.pythonware.com/products/pil/

  import PIL

- Sounddevice

  For playing and recording audio

  https://python-sounddevice.readthedocs.io/en/0.3.7/

  import sounddevice

- Soundfile

  For opening .wav audio files

  https://pypi.python.org/pypi/SoundFile/0.8.1

  import soundfile

Other modules used are Tkinter, Time, Math and OS.

import tkinter - for making GUI

import time - for recording using sounddevice

import math - for mathematical operations

import os - for controlling the system os

Python Module for the Code:

For code citation, please use

S Vinu Sankar, GUI for Epoch Based Continuous Time and Pitch Scaling, (2017), GitHub repository, https://github.com/vinusankars/Time-and-Pitch-Scaling-with-Python/

For using the code,

Run

$: git clone https://github.com/vinusankars/Time-and-Pitch-Scaling-with-Python.git

Now the code will stored in a folder “Time-and-Pitch-Scaling-with-Python” in your system. Copy paste the scripts to Python directory and run

$: python GUI.py

to run the Graphical User Interface

To use the algorithm script, in your Python program import it using

import EBCTPS.py

References:

- "Auto Time-Scale Modification" PhD Thesis by David Dorran 2005

- MATLAB code for EBCTPS algorithm by Mr. Sunil R, Spectrum Lab, Electrical Engineering Department, Indian Institute of Science, Bangalore

- B-spline points function was adapted from a piece of code in https://stackoverflow.com/questions/24612626/b-spline-interpolation-with-python