EEC201-Project-A-Final-Report

Target

Design a vocoder in Matlab GUI. It should show the waveform of the uploaded voice file,
performing streching, compressing, frequency or pitch shift.

Group member

Qinwe Deng -mrdeng@ucdavis.edu

Zhichao Qiu -zcqiu@ucdavis.edu

Shudi Chen -sdichen@ucdavis.edu

Project tasks

We decompose the design into four parts:

1.design a GUI interface

2.design a function that can cut down a specific pieces of sound from the file

3.design a function that can perform time streching or time compressing on the sound

4.design a function that can perform phase or pitch shifting on the sound

We have read some papers and online videos and found that we can design a phase vocoder using STFT and many overlap-add(OLA) ways such as OLA, SOLA, PSOLA. Therefore, we mainly focus on how to synchronize STFT signals using OLA and other methods.

Descriptions of methodology about each part

1.Use MATLAB to design a GUI interface of phase vocoder

2.We will first compute where should we cut down the signal and then cut it down.

3. For time stretching or time compressing, we SOLA and PSOLA to change the speed of the speech while retain the original pitch. If we have enough time, we will find some frequency domain processing method and compare the result of these methods.

4.For pitch shifting, we use a time domain processing method. First, we use SOLA algorithm to change the speed of the speech. Then we use resampling to change the speed and pitch of the speech at the same time.The speed up ratio is the same with the resampling ratio, so that the pitch will move while retain the original speed.

Division of project tasks

Qinwen Deng: Use OLA and SOLA method to stretch or compress the speech; Pitch shifting

Task describe:

The task of speech stretching or compressing is to change the speed of speech while keeping the pitch of the original speech unmoved. On the other hand, the task of pitch shifting is to shift the pitch of the speech while keeping the speed of the speech unmoved.

Shudi Chen: Use MATLAB to design a GUI interface of phase vocoder, and complete signal cutting part.

Task describe:

This task requires to design a GUI interface to get input parameters for signal processing and to cut down the target signal we desired.

Zhichao Qiu: Use OLA and PSOLA to implement time domain stretching/ compressing, and pitch shifting.

Task describe:

First, we tried to complete this phase vocoder using OLA method, then we found out problems of OLA method and tried to use more advanced OLA method (SOLA & PSOLA) and hope to compare their result. We have successfully completed OLA and SOLA method. As for PSOLA, there are still some problem that I didn't solve.

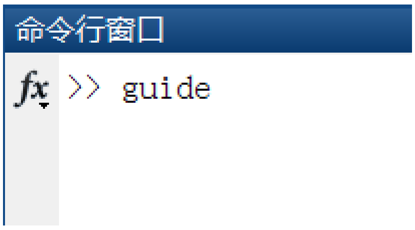
Tabulated Milestone

Time		Milestone
19-Feb	1st Group Meeting	Work Division
26-Feb	2nd Group Meeting	SOLA Program finished
28-Feb	Online discussion	Interim Report finished
5-Mar	3rd Group Meeting	GUI finished
12-Mar	4th Group Meeting	Whole program created
16-Mar	5th Group Meeting	Final Report finished

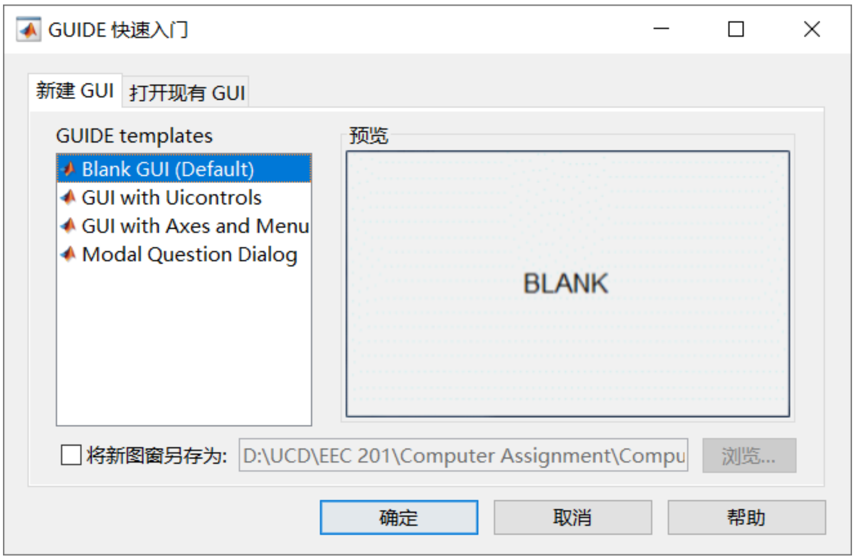
GUI designing guidance and interface descriptions

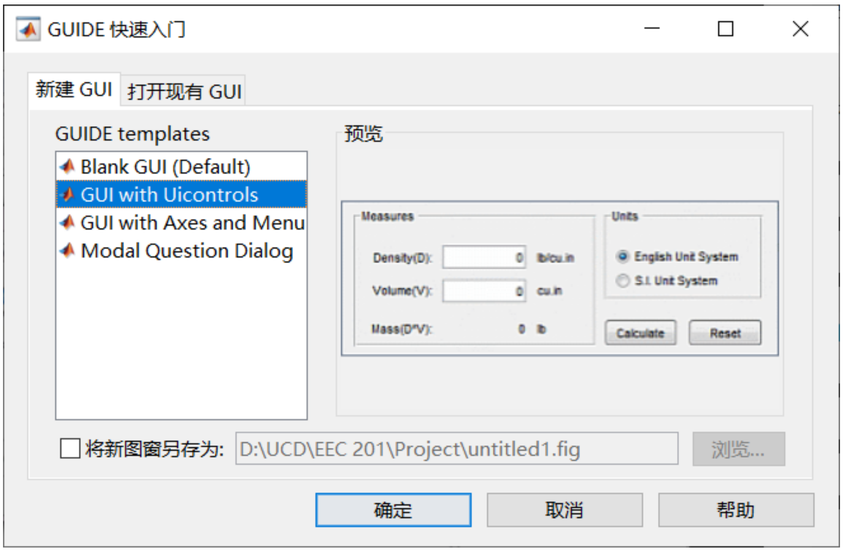
GUI designing guidance

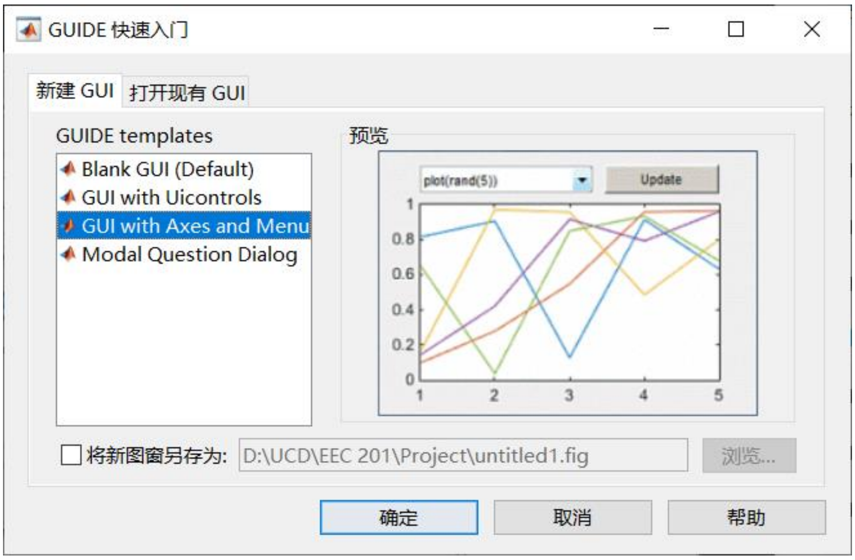
Firstly, using command guide to start building the GUI.

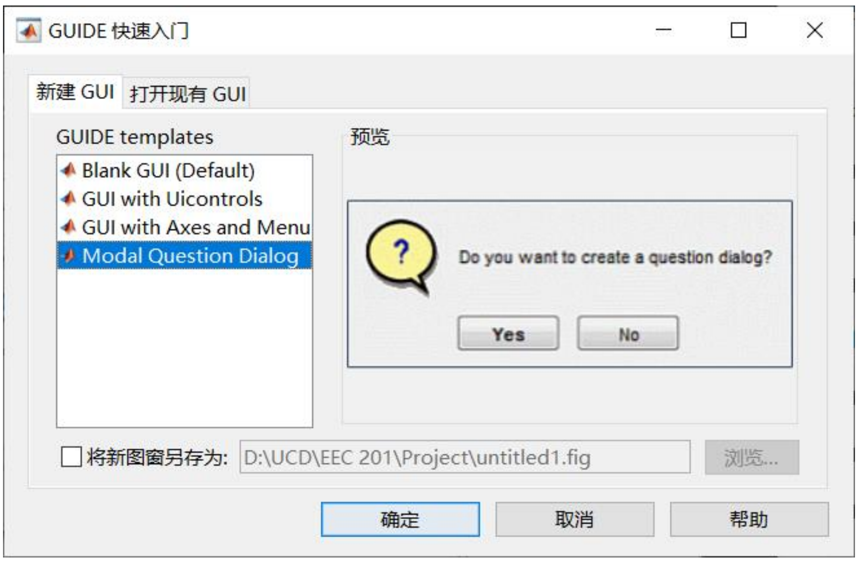


Then, you can choose which model of GUI you want to use. There are four types of GUI models. The first one is Blank GUI, you will design and build the GUI all by yourself. The second model is GUI with uicontrols. It provides some basic control function. The third model is GUI with Axes and Menu, which has menu function and can plot figures. The last model is used to create a question dialog. In this Project, we choose the first model, Blank GUI.Four models are shown below.

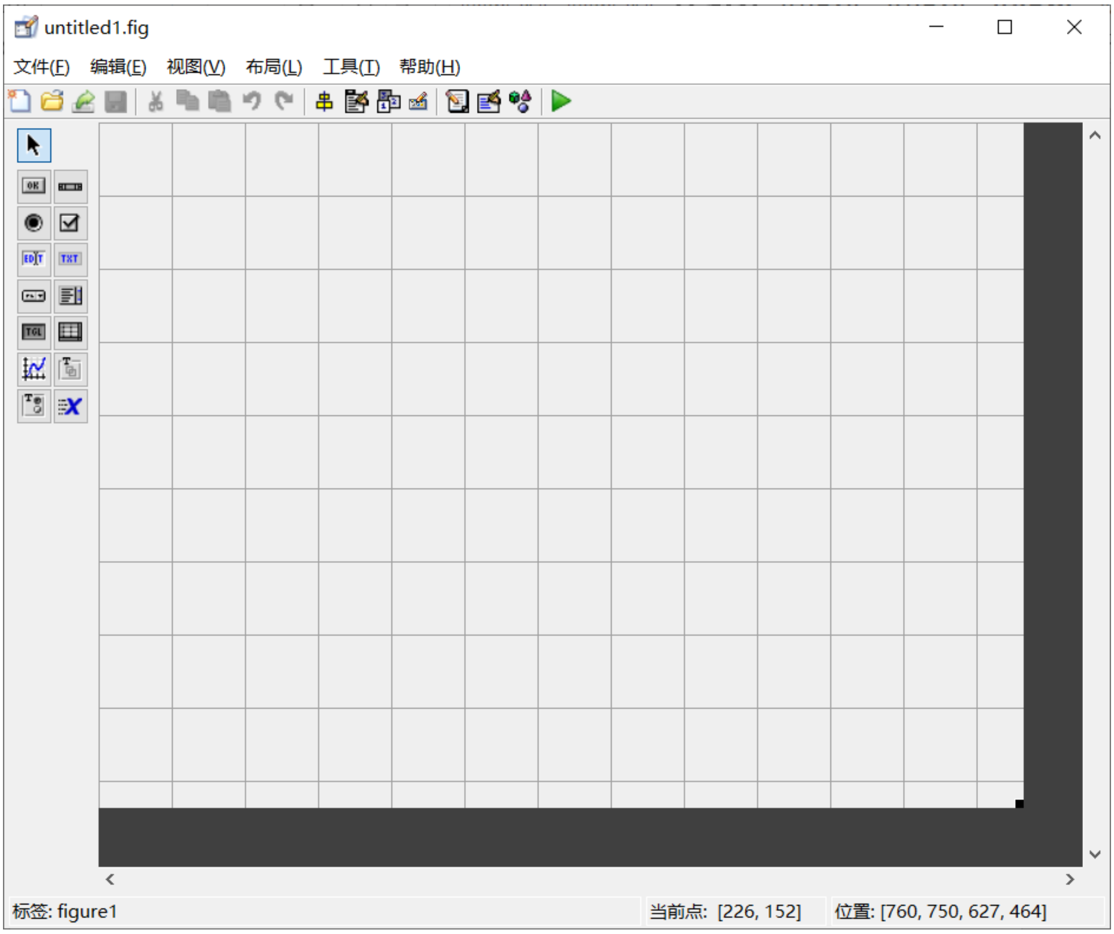




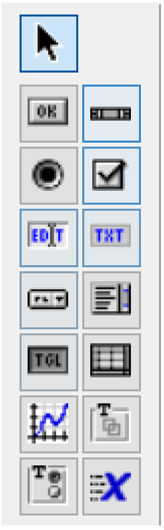


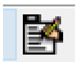


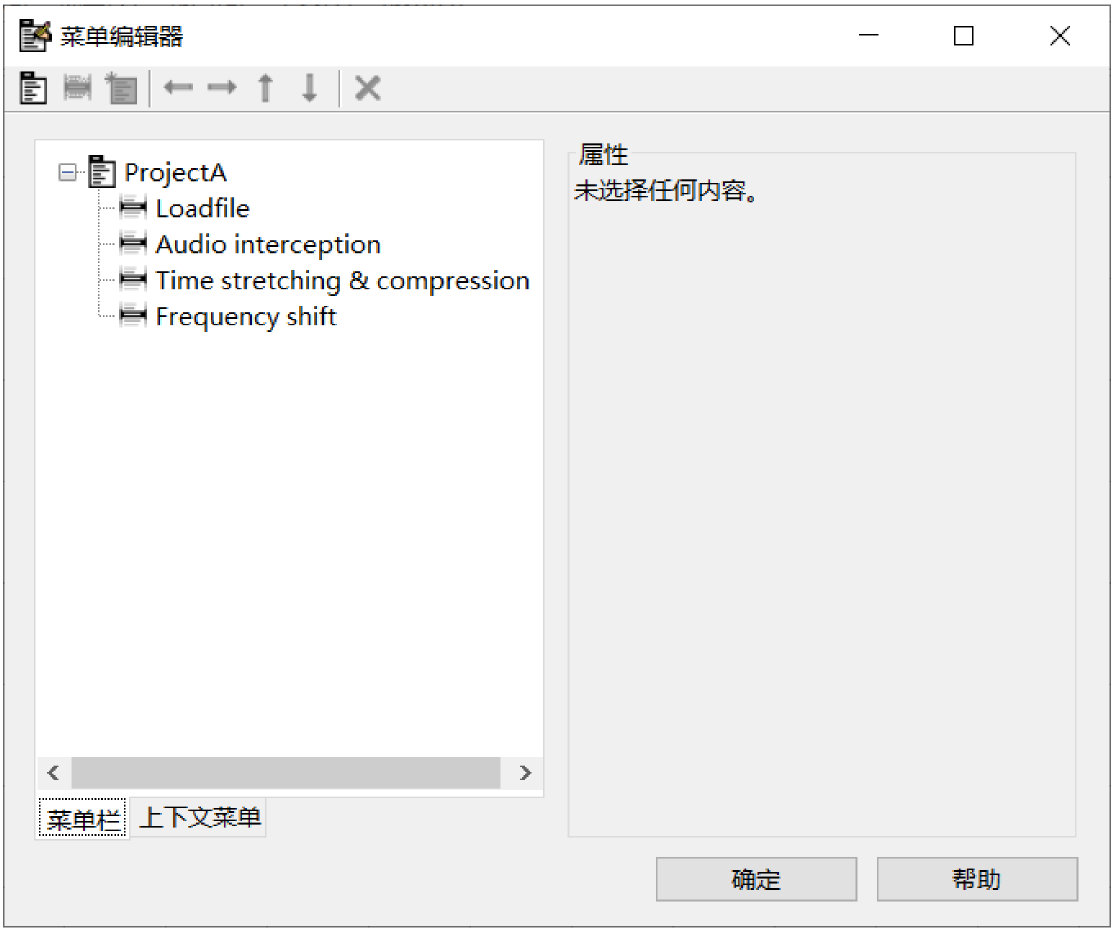
After choosing our desired models, we will get a interface shown below.



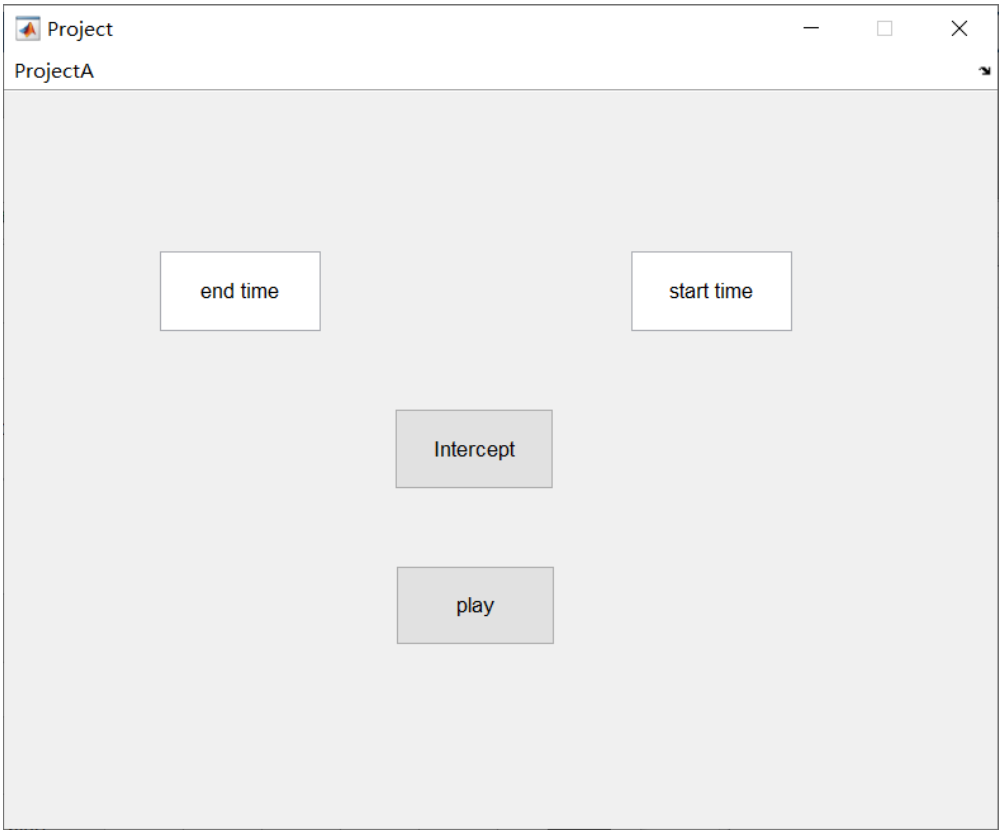
The menu on the left side is the basic tool menu. By using these tools, we can achieve the function we want.

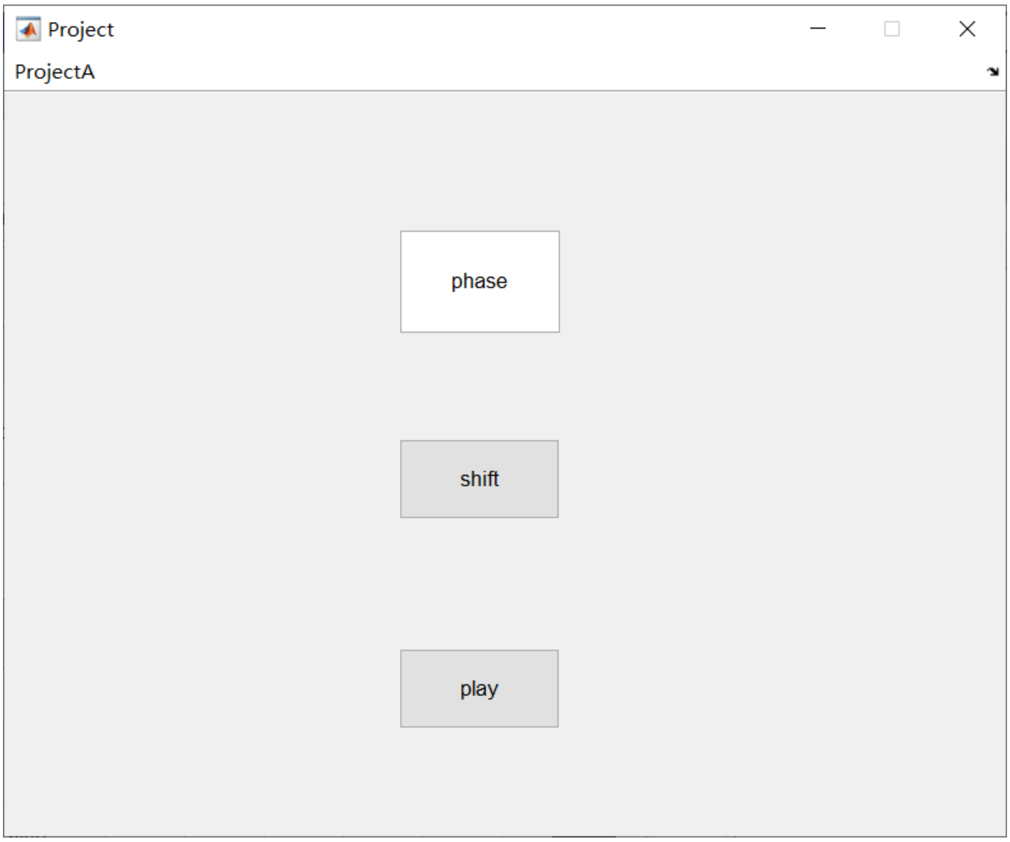


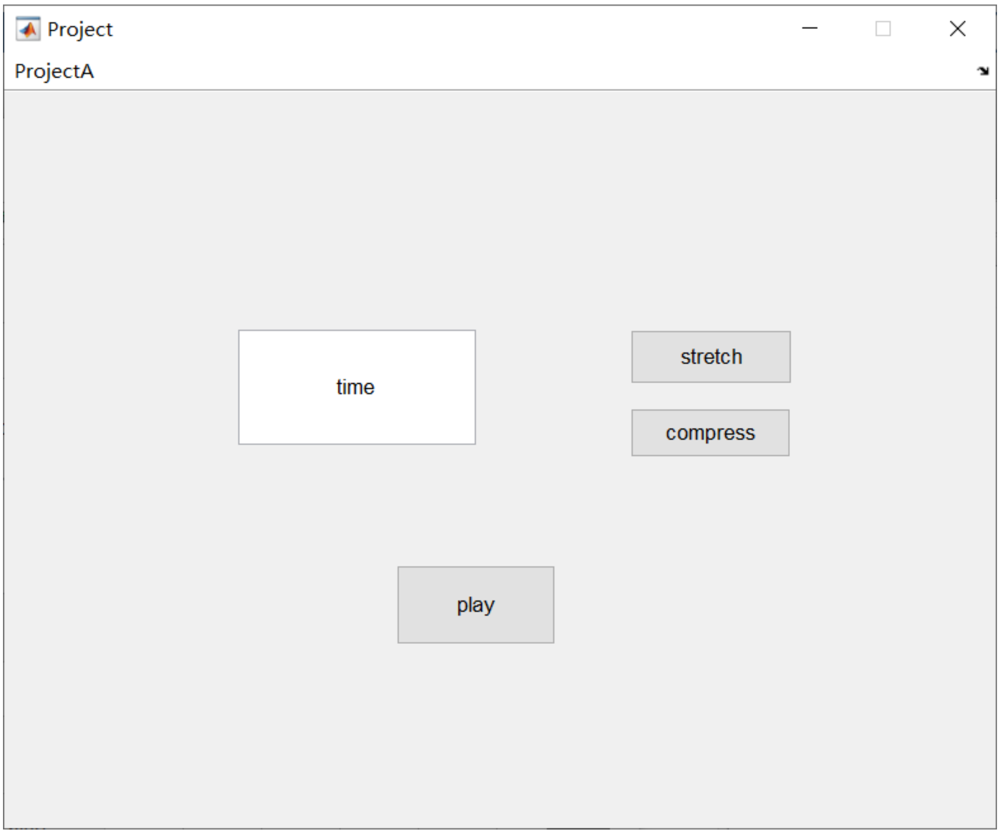
There is a button  on the top of the window. It is the menu function model. We use this button to create a menu. This is the menu we designed.



The next step is to design the interface we want by using the tools mentioned above. The designed interfaces are shown below.


Project
ProjectA
end time
start time
Intercept
play


Project
ProjectA
phase
shift
play



As we have designed all the interfaces we need, the last step is to edit the Callback function of each button, which is the task we will do in the rest of time.

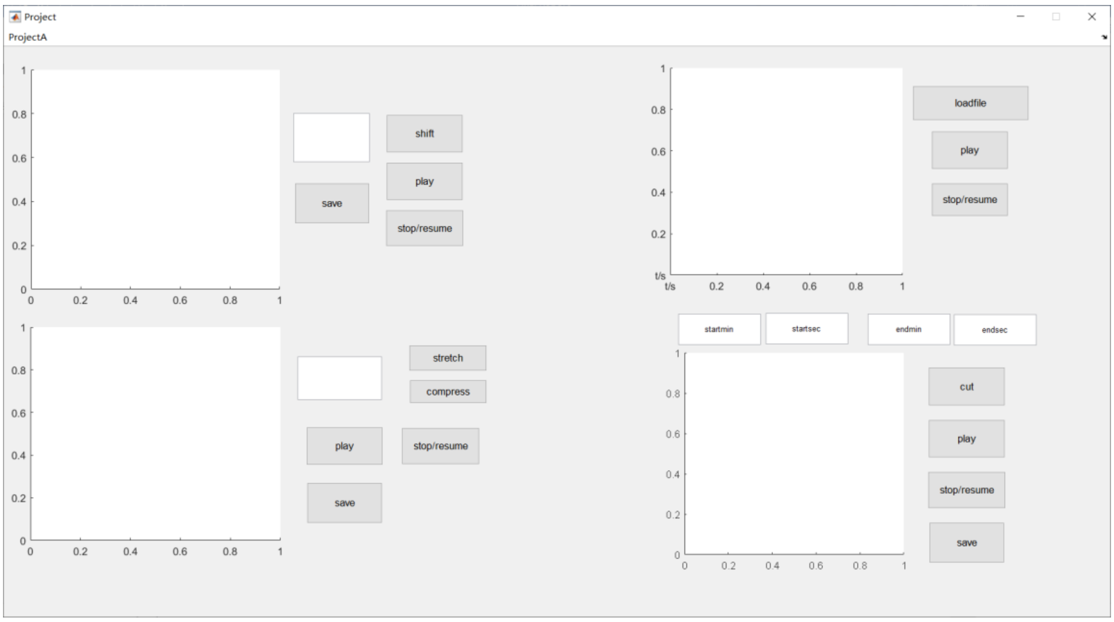


Figure.1 Interface

By using MATLAB, we designed an interface which is shown in Figure.1 This interface is designed to achieved four main tasks of this project:

1) Loading an audio file from the computer

2) Cut down a piece of sound from the file

3) Perform time stretching or time compressing on the sound

4) Perform phase or pitch shifting on the sound

GUI interface introduce

First part: Loading file.

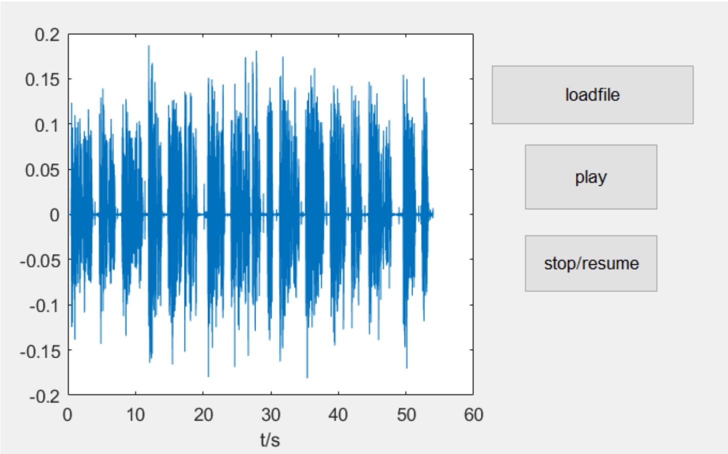


Figure.2 Load file

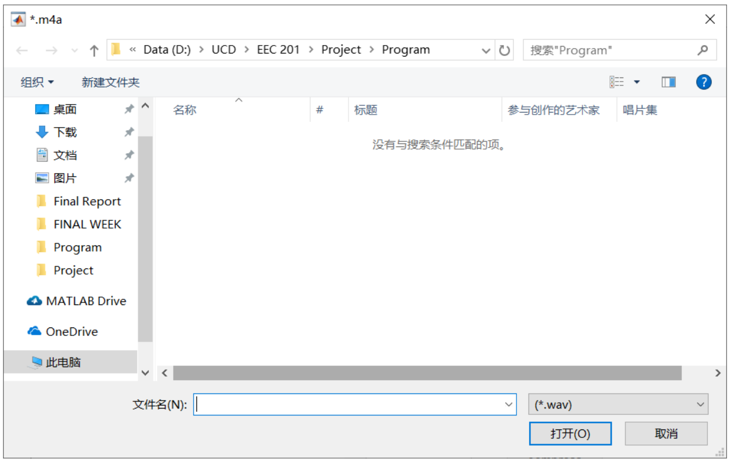


Figure.3 File choosing window

This is the “Load file” part, which can load an audio file from the computer. Once you click the “loadfile” button, there will be a window for you to choose the file you want to load from computer. After you choose the file, the waveform of the audio you chose will be shown on the waveform on axes1. The “play” button is designed to play the audio and the “stop/resume” button is to pause and restart the audio.

Second part: Cut down audio.

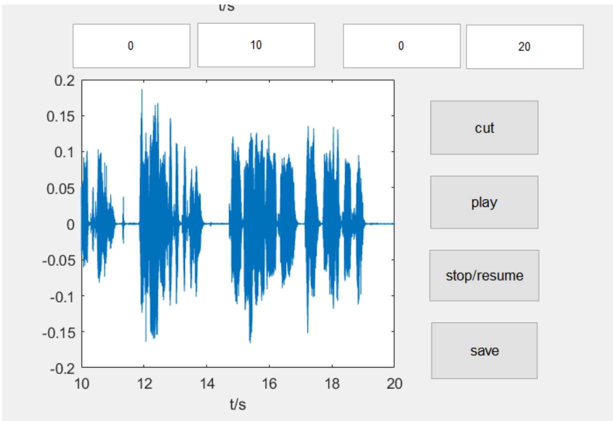


Figure.4 Cut down audio part

Figure.5 File saving

This part is to grab a piece of audio. Four edit texts “startmin”, “startsec”, “endmin” and “endsec” are used to decided which part of audio you want to grab. For example, if you want to cut the audio into 10 seconds, from 0:10 to 0:20. Then the “startmin” is 0, “startsec” is 10, “endmin” is 0 and “endsec” is 20. Then you click “cut” button, the audio will be shortened into a 10 seconds audio and its waveform will be plotted on the axes2. The result is shown in the Figure.4. The “play” and “stop/resume” button still function as play, pause and resume the audio. The “save” button can save the file. As shown in the Figure.5, all audio files are saved by using “save” button.

Third part: Time stretching and compression

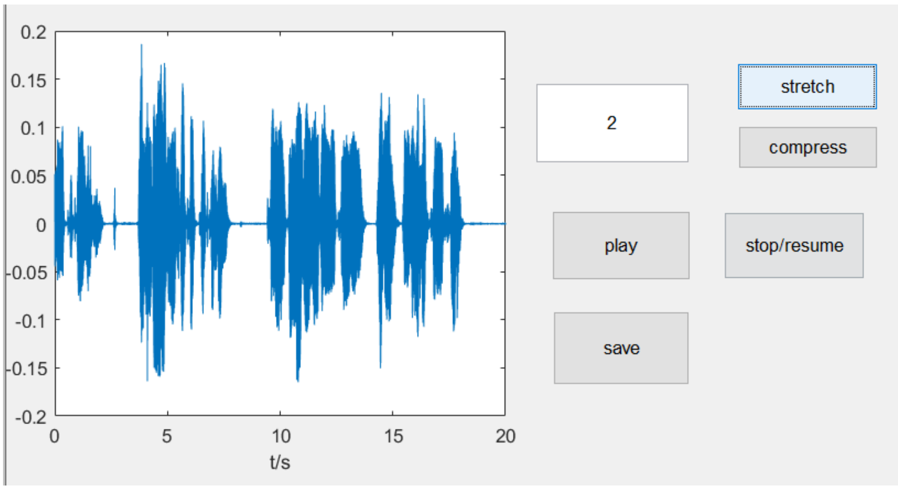


Figure.6 Time stretching

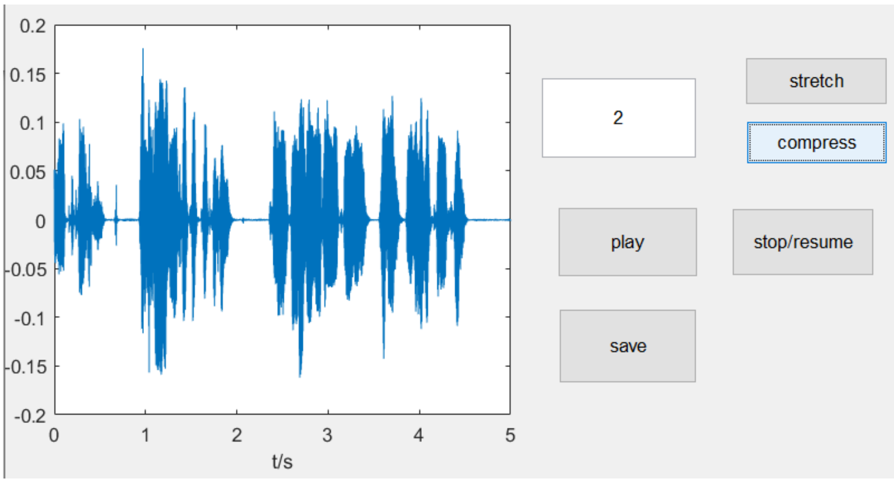


Figure.7 Time compression

This part is to stretch and compress the audio. Once you enter the parameter in the edit text frame, the audio can be stretched or compressed when you click the “stretch" or “compress” button. The waveform of the audio will be plotted on the axes3. The result of stretching and compression are shown in Figure.6 and Fiugre.7. The function of “play”, “stop/resume” and “save” buttons remain unchanged.

Forth part: Pitching shifting.

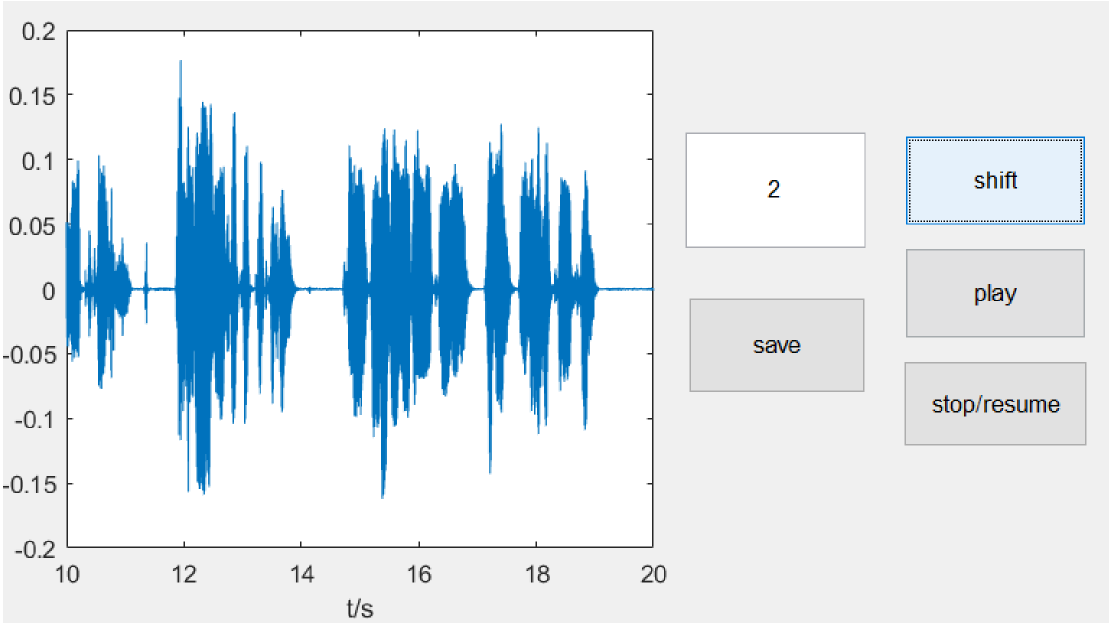


Figure.8 Pitch shifting

This part is used to shift the pitch of the audio. Once you enter the parameter in the edit text frame then click the "shift" button, the audio's pitch will be shifted. The waveform of the audio after pitch-shifting will be plotted on the axes4, which is shown in the Figure.4. The function of "play", "stop/resume" and "save" buttons remain unchanged.

General description of OLA, SOLA and PSOLA

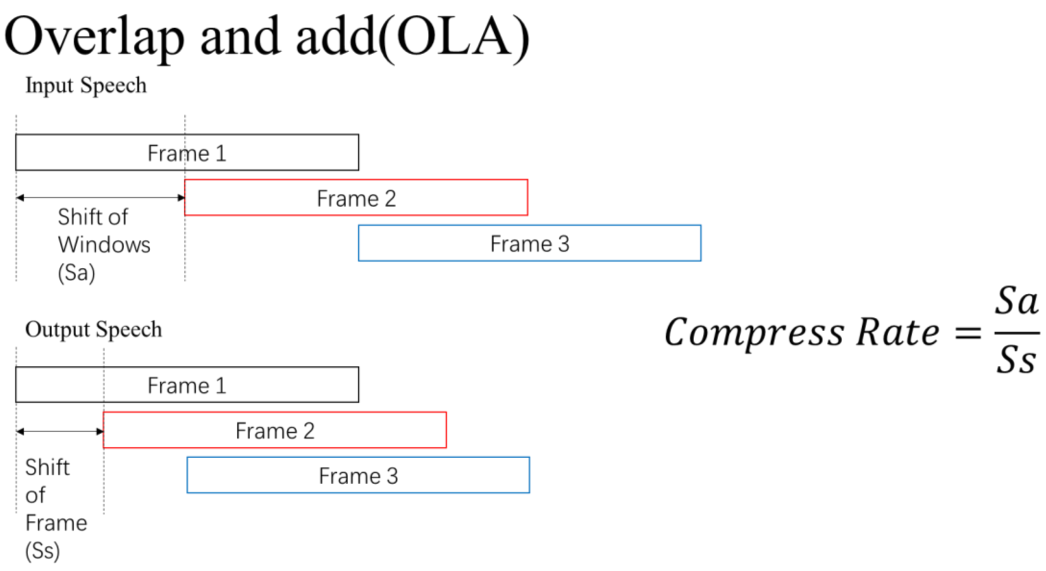
Method:

In this project, we use overlap and add (OLA) method, which is a time domain approach, to stretch or compress the speech. The idea of OLA is:

1. use window to cut the original speech into overlapped frames.

2. Move the frames closer (for compressing) or further (for stretching), as what is shown in the following figure.

3. Add the moved frames to generate a new speech

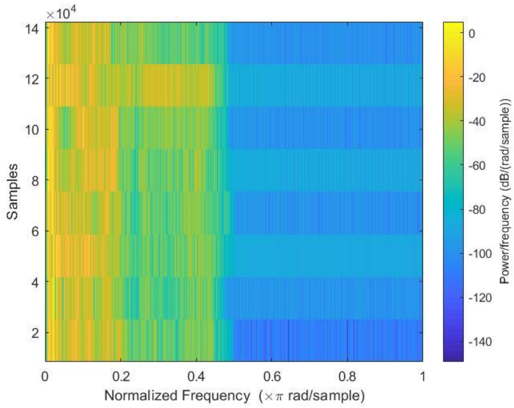


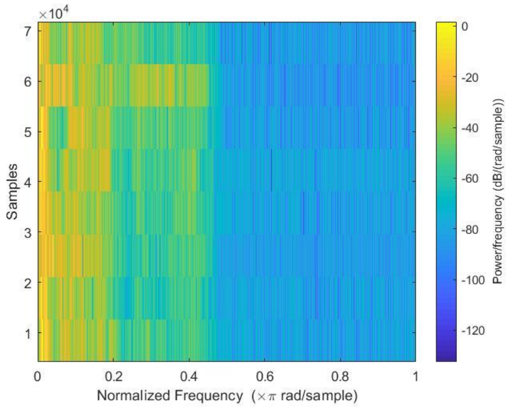
The problem of OLA method is that because the shift between two overlapped frames is fixed, the waveform of overlapping part between nearby frames can not match perfectly, which may product unexcepted impulse in the new speech and reduce the quality of speech. This problem can be solved be Synchronized overlap and add (SOLA) method. The overall idea of SOLA method is that the shift between two overlapped frames, Ss, is not fixed. By adding an offset number, Ss can be smaller or bigger to find the most similar overlapping part between two nearby frames. In our program, the offset number is less than 1/10 of the frame length, and the similarity of two overlapping part is measure by cross correlation.

For pitch shifting, first, we use SOLA to change the speed of speech. Then resample the speech to change the speed and shift the pitch at the same time. For example, if we want to shift the pitch up by the factor of 2, we can firstly use SOLA to stretch the speech by the factor of 2, then use downsampling by the factor of 2 to compress the speech by the factor of 2 and shift the pitch up by the factor of 2 at the same time.

Test results

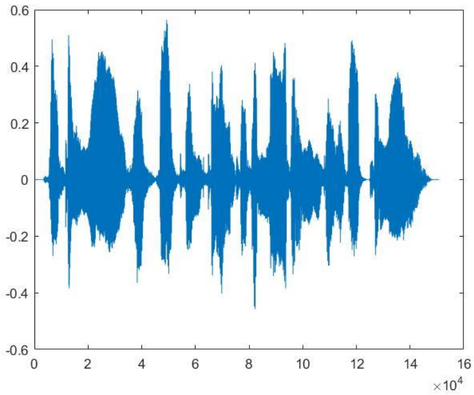
OLA part (to obtain the result, please run OLAtest.m)

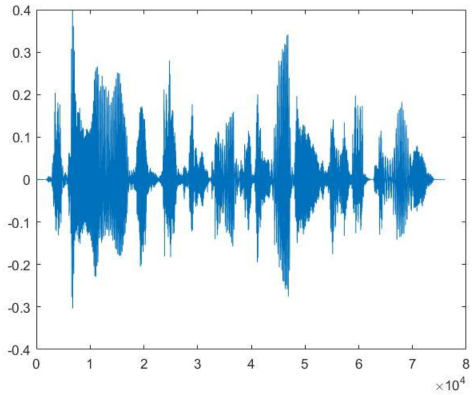




The figure on the left is spectrum of the original signal. The other one is signal compressed by 0.5.

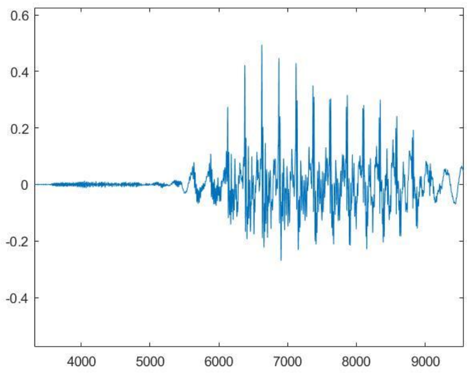
Comparing the above two figure, they are similar in frequency but the time axis have been compressed to 0.5.

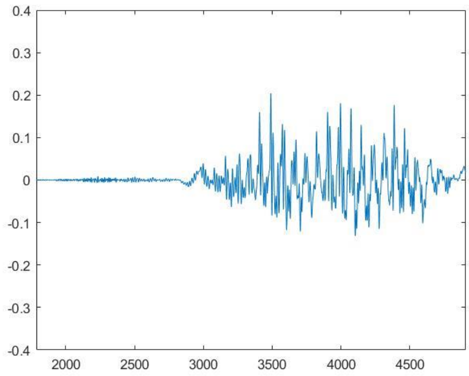




The waveform on the left is the original signal. The other one was compressed by 0.5.

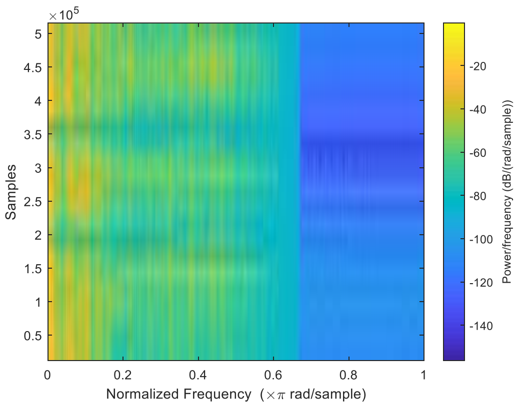
The above waveform doesn't give a clear comparison between the two wave form, so we zoom out in the starting part of each waveform

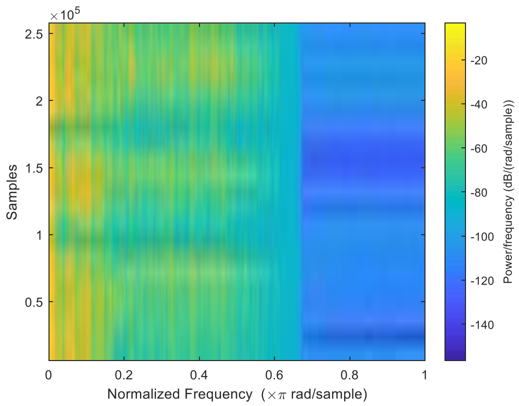




You can see that there are still obvious difference between the two signal, so if you compare the two signal, you can still hear the differences.

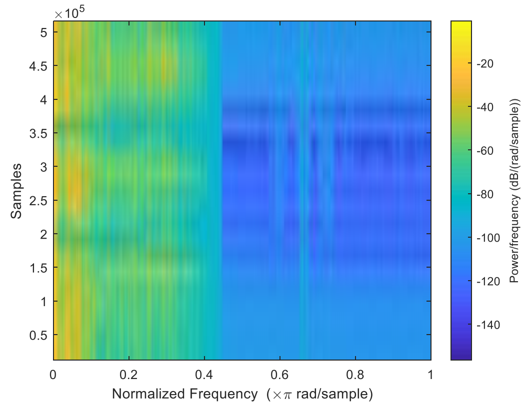
SOLA part (to obtain the result, please run SOLA.m and Pitchshift.m)



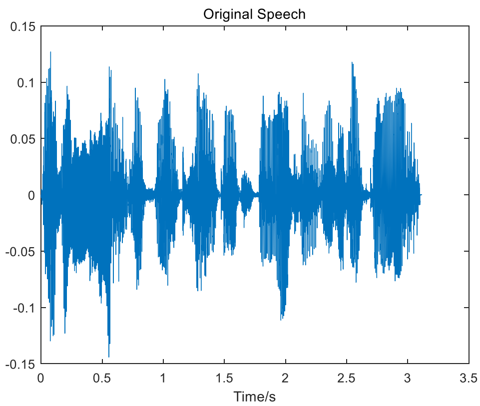


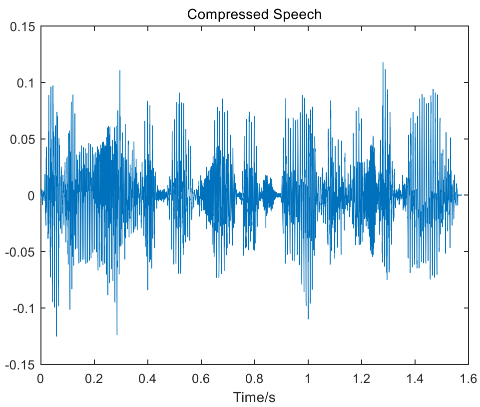
Those two spectrums are similar, which means the pitch of compressed speech is not moved.

The spectrum of pitch shifted speech is



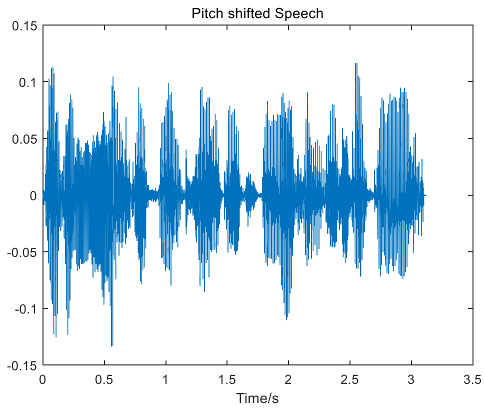
Compared with the spectrum of original speech, the bandwidth of pitch shifted speech is compressed, but the overall form is similar to the form of the spectrum of original speech.



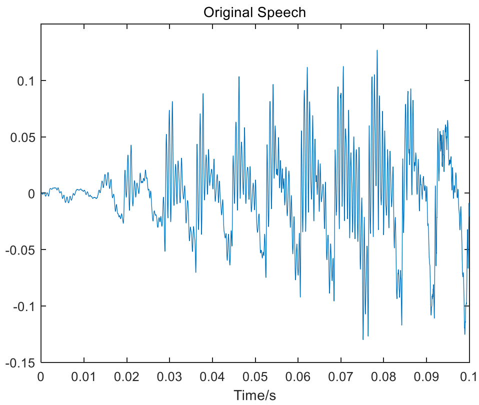


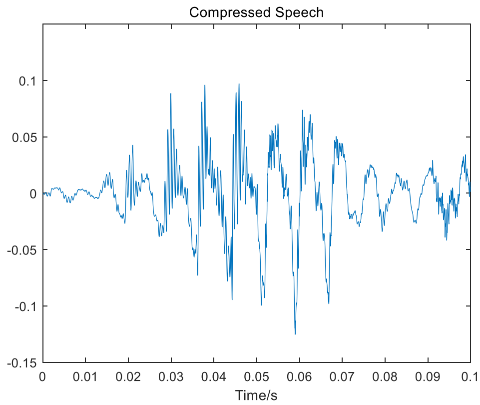
The waveform on the left is the original waveform and the other is the compressed one.

The waveform of a segment of the pitch shifted speech is



Focusing on the first 0.1s of each speech, the waveform of the original speech is

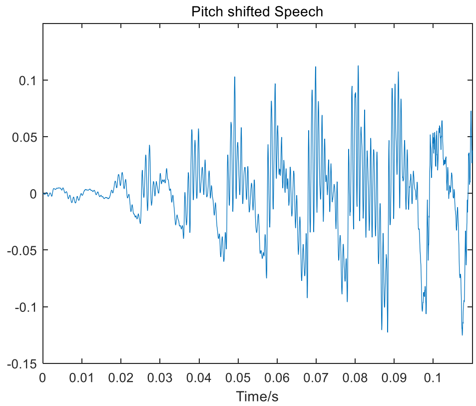




While the waveform of the compressed speech is

Compared with the waveform of the original speech, the length of each pitch in compressed speech is not changed, while the number of pitches is less than that in original speech. Thus, the speed of compressed speech is higher while the pitch is not changed.

The waveform of the pitch shifted speech is

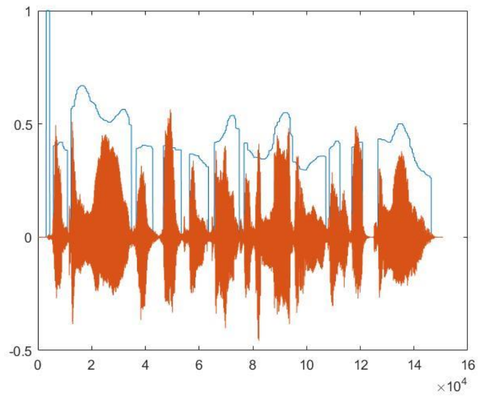


Compared with the waveform of the original speech, the length of each pitch in compressed speech is wider, while the total length of a bunch of similar pitches is the same. Thus, the pitch of shifted speech is lower while the speed is not changed.

PSOLA part (to obtain the result, please run TDPSOLAvocoder.m)

We searched some data online about PSOLA and found out that PSOLA method requires two important steps: concatenation of syllabus and PSOLA. So far, we have completed pitch estimation, which is one important part of pre-processing of PSOLA. It's very convenient to implement estimation using auto correlation function method. Still there's another important pre-processing step called pitch marking. In that part, we seek to compare the candidates in a specific search region to obtain the optimal candidate for reconstructing the original signal. Based on the data we have searched online, most of the engineers use Dynamic Programming to implement the above process. Unfortunately, we don't have that much time to complete it. All we get now is computing the pitch contour the target signal.

The following figure is the target signal and its corresponding pitch contour.



Reference

1. Eric Moulines, Francis Charpentier

   Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones

   Speech Communication, Volume 9,Issues 5-6, Dec 1990, Page 453-467

2. TAMU lecturing slides-http://research.cs.tamu.edu/prism/lectures/sp/l19.pdf

3. GUI learning video (Chinese)- www.bilibili.com/video/av14841464

4. Olli Parviainen

   Time and pitch scaling in audio processing

   Software Developer's Journal 4/2006

5. Removal of spectral mismatch using PSOLA for speech improvement-
http://shodhganga.inflibnet.ac.in/bitstream/10603/125263/15/15_chapter%207.pdf