Challenge 3 Report

Juan Reyes
University of California, Santa Barbara

Marco Chavez
University of California, Santa Barbara

***Abstract*—This report focuses on the implementation of a phase vocoder in MATLAB. The purpose of this phase vocoder is to modify the length of an audio file while preserving the integrity of the frequency content and pitch. After testing on various forms of audio, such as speech and music, it seems to work really well.**

I. INTRODUCTION

Fourier transforms open up many possibilities for signal processing and analysis. In the case of this report, a specialized form of fourier transforms was utilized. It is known as a STFT (Short time fourier transform). Using this type of fourier transform proved to be useful to implement the phase vocoder. The phase vocoder in essence changes the frequency and phase content of the signal in a consistent manner and then using the iSTFT (inverse STFT) the signal can be reconstructed into the time domain using an OLA(overlap-add) method to achieve the modification of the playback rate of the audio signal.

II. METHODS

First using the STFT a spectrogram was generated to contain the frequency information of sequential frames. The number frames are dependent on the window size and hop size, in this implementation the window size and hop size used are 1024 and 256, respectively. The window used when calculating the the STFT were hamming windows, to create a precise representation of each frame in the frequency domain.

After using the STFT method, the phases were interpolated to fit the new length of the audio signal, which was calculated using the rate variable. The interpolation was done using the magnitudes and phases of each frame.

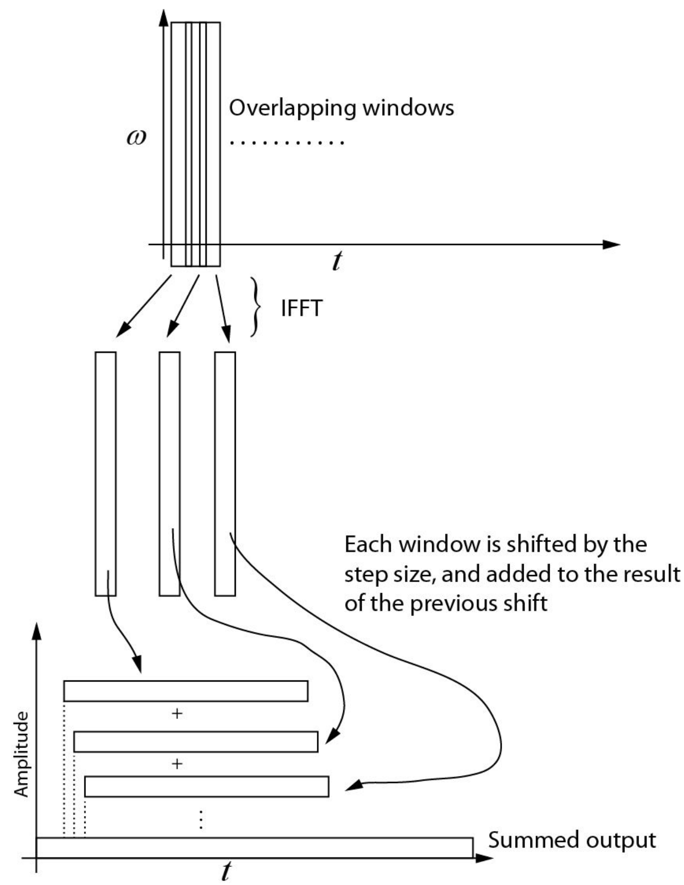

Figure 1. Visualization of OLA method.
Source: dsp.stackexchange.com via Nicholas Kinar

The phase shift accounts for the modification of frequency. To maintain the integrity of the audio file, in particular to avoid pitch distortion the phase angle of consecutive frames must be shifted. To achieve this for every two frames that are used to calculate a new frequency, the phase difference must be calculated and added to the phases of all subsequent frames.

The sum of the magnitudes of the two frames and the phase difference are combined and used to create the new interpolated spectrogram.

With the new spectrogram that was stretched or compressed an iSTFT was used to convert it back into the time domain. The method employed for this is known as OLA. This was done by using one frame at a time from the interpolated spectrogram. An iFFT is performed on each frame and then multiplied to a hamming window. This is then added to the final signal. Again this is repeated for the next frame and added to the final signal with an offset of one hop size. This is continuously done until all frames have been added. Figure 1 shows this visually.

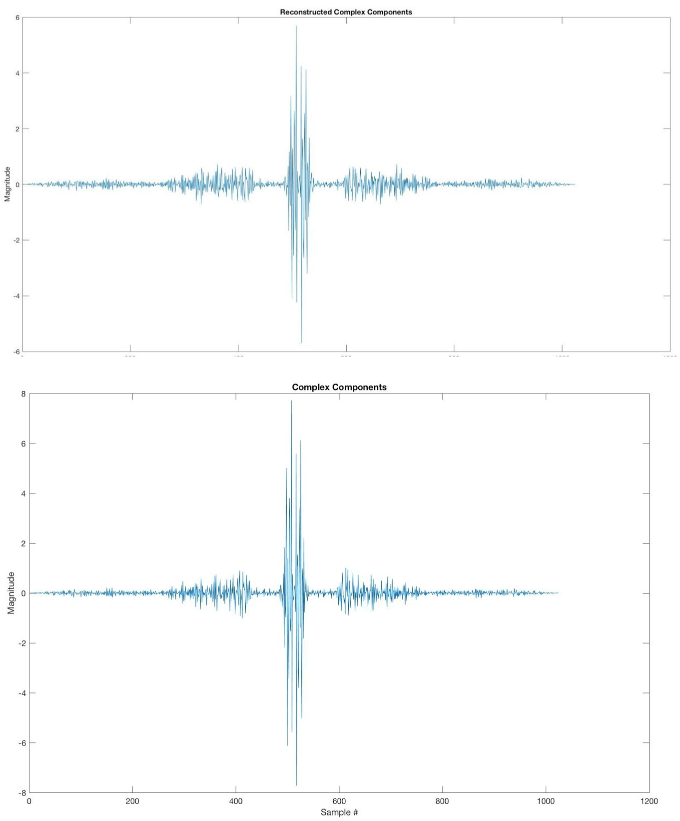

Figure 2. Top: After
Bottom: Before

III. RESULTS

Applying this method to various audio signals, such as speech and music, seem to yield desirable results. From a sensory perspective, the final audio file sounds stretched or compressed without change in the pitch. It does not add the chipmunk effect when the original file is compressed. Similarly when stretching the file it does not create a deeper sounding pitch. From data perspectives, the complex components before and after are preserved as shown in Figure 2. The time signal is shown in Figure 3 when it is stretched by a factor of 2 and Figure 4 when it is compressed by a factor of 2. In both plots, the original audio is the purple waveform and the modified audio is the cyan waveform.

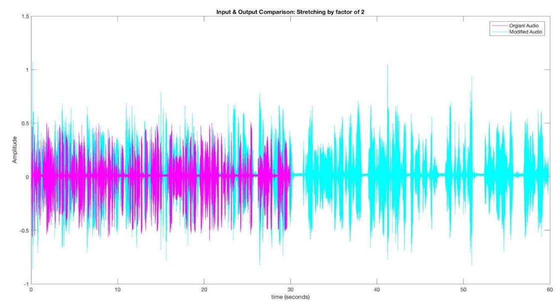

Figure 3. Stretched waveform
y-axis: amplitude, x-axis: time

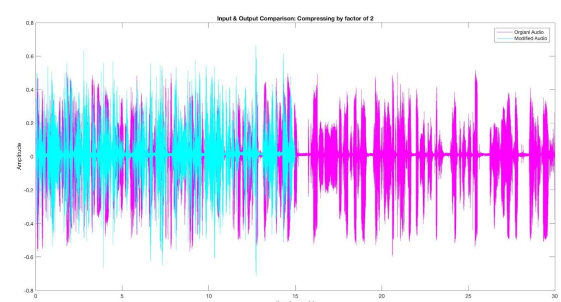

Figure 4. Compressed waveform
y-axis: amplitude, x-axis: time

IV. DISCUSSION

The method seems to work well for many types of audio content. The quality of the results are largely determined by the window size and hop size. Sizes of 1024 and 256 worked really well in the audio tested. Smaller hop sizes may create better results for high frequency signals but can be costly in terms of performance. Since the spectrogram was interpolated it can also raise a concern of phase accuracy, although it was not thoroughly tested it may create artifacts for certain frequency content.

V. CONCLUSION

In conclusion the phase vocoder method achieves the desired outcome. It utilizes the STFT and iSTFT to create a precise resynthesis of the interpolated spectrogram. As well as the preservation of the frequency content and pitch. It also seems to have a reasonable time complexity.

REFERENCES

- J. Laroche and M. Dolson, "New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects," *Applications of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on*, New Paltz, NY, 1999, pp. 91-94.

Juan Reyes

- Wrote phase interpolation code
- Helped debug code

Marco Chavez

- Wrote stft and istft code
- Helped debug code