

Oliver Marlan 311271383 (SID)

Lab Report 1: PHASE VOCODER PITCH-SHIFT

Digital Audio Systems, DESC9115, Semester 1 2012

Graduate Program in Audio and Acoustics

Faculty of Architecture, Design and Planning, The University of
Sydney



THE UNIVERSITY OF
SYDNEY

Introduction

For this Lab Report I will be looking at a Matlab function that performs a pitch shift effect using Phase Vocoder. I have supplied the main function 'pitchshift_oly.m' along with 'fusionFrames.m' and 'createFrames' which are required for certain parts of the function to work.

The audio I have supplied to be used with the function is a short segment of dialogue called 'heroin'.

The script I have supplied to run the function is called 'script_for_pitchshift.m'. This script will import the audio file 'heroin' using the wavread function. It will then perform the function pitchshift_oly, play the pitch-shifted output of 'heroin' and make it into a wave file called 'ShiftedHeroin' using the wavwrite function. Finally, it will plot the output 'ShiftedHeroin' in a simple frequency vs. time plot.

What does the function do?

The simple answer is that this function will alter the pitch of an input audio file by an amount specified by the user.

There are, however, a number of processes that occur within the function that make this happen.

The main steps are:

- The audio signal is separated into frames
- Phase Vocoder method applied
- Resampled to be played at chosen sample rate.

Firstly, Input signal is divided into frames of size determined by input argument 'windowSize' with the frames overlapping each other. In this case, I have set the frame size to 1024 samples, as this is appropriate with a sampling frequency of 44,100 samples per second.

The distance between the beginnings of each frame is called the 'hop' and will determine the number of samples that the frames overlap each other. The size of of overlap is determined by input argument 'hopSize'. By setting the overlap to 256 samples the frames will overlap 75% of each other.

A Hanning window is applied to each frame, and then each window is analyzed in the frequency domain using the Fast Fourier Transform (FFT).

The FFT takes N consecutive samples out of the signal $z(n)$ and performs a mathematical operation to yield N samples $X(k)$ of the spectrum of the signal. [2]

Now that we are in the frequency domain, the phase difference between frames, called phase-shift, is calculated.

The difference between frames is now removed so that they can be shifted in time without too many discontinuities or undesired artifacts in later processing.

The inverse discrete Fourier transform (IDFT) is performed on each frame spectrum. The IDFT allows the transform from the discrete-frequency domain to the discrete-time domain. The IDFT algorithm is given by:

$$x(n) = IDFT[X(k)] = \frac{1}{N} \sum X(k) \frac{e^{j2\pi kn}}{n} \quad n = 0, 1, \dots, N - 1. \quad [2]$$

The result is then windowed with a Hanning window to obtain $q_i[n]$. Windowing is used this time to smooth the signal. This process is shown in this equation. [1]

$$q_i[n] = \left\{ \frac{1}{N} \sum_{k=0}^{N-1} (X_s[k])_i e^{-j\left(\frac{2\pi kn}{N}\right)} \right\} w[n] \quad n = 0, 1, 2, \dots, N - 1$$

(Synthesis stage equation)

Each frame is then overlap-added as shown in the equation below. The variable L stands for the number of frame and $u[n]$ represents the unit step function. [1]

$$y[n] = \sum_{i=0}^{L-1} q_i[n - i \times hop_s] \{u[n - i \times hop_s] - u[n - i \times hop_s - N]\}$$

(Overlap-add of the synthesis frames equation)

Finally, the signal is resampled to the new pitch as specified by the input argument 'step'. One step is equal to one semitone. So to input a value of 4 for 'step', the pitch will be shifted 2 whole tones and input a value of 12 the pitch will be shifted by an octave. The number of steps to be shifted is called the 'scaling factor' and is shown in the function as:

$$\text{alpha} = 2^{(\text{step}/12)};$$

This can be used to find out what the final frequency will be after the pitch shifting has taking place, and can be shown in the mathematical expression:

$$f_{-4} = f_0 \times 2^{\left(\frac{-4}{12}\right)} = 1000 \times 2^{\left(\frac{-4}{12}\right)} = 793.7 \text{ Hz} \quad [1]$$

A procedure called interpolation is used to perform the resampling task. This involves placing a number samples, with a value of zero, between each of the samples obtained from memory. The resulting signal is a digital *impulse train*, containing the desired voice band. [3]

Plots and Graphs

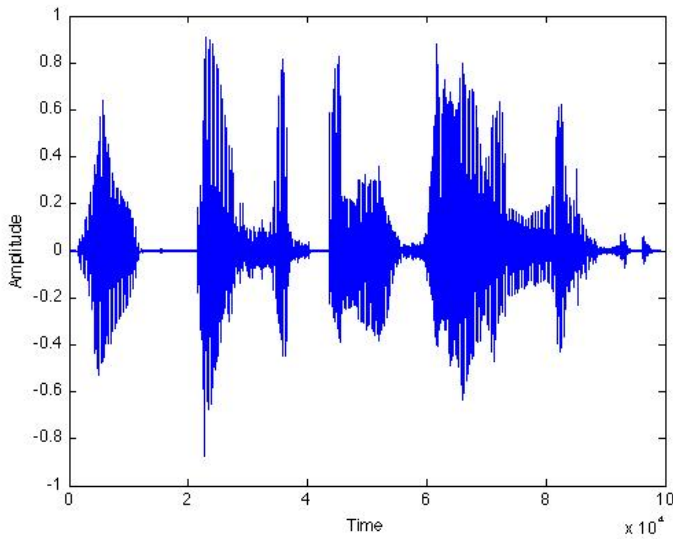


FIG 1. Original 'heroin' waveform

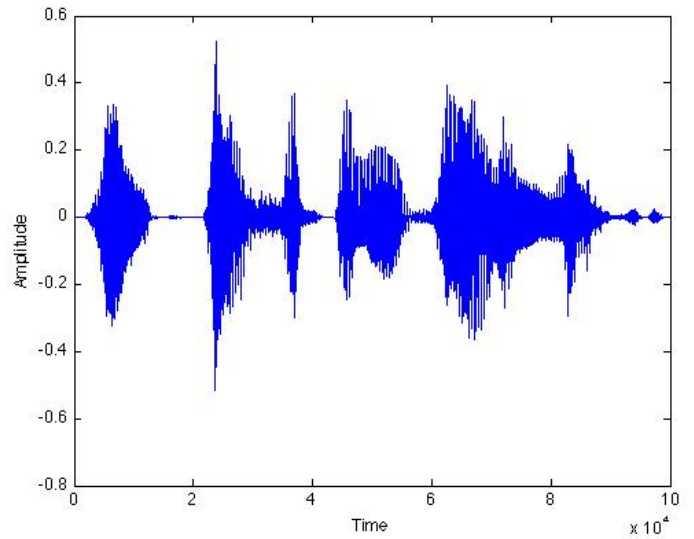


FIG 2. Output 'ShiftedHeroin' waveform

Figures 1 and 2 clearly show that some amplitude content has been altered, while the duration of the signal has not changed at all, which is the desired affect.

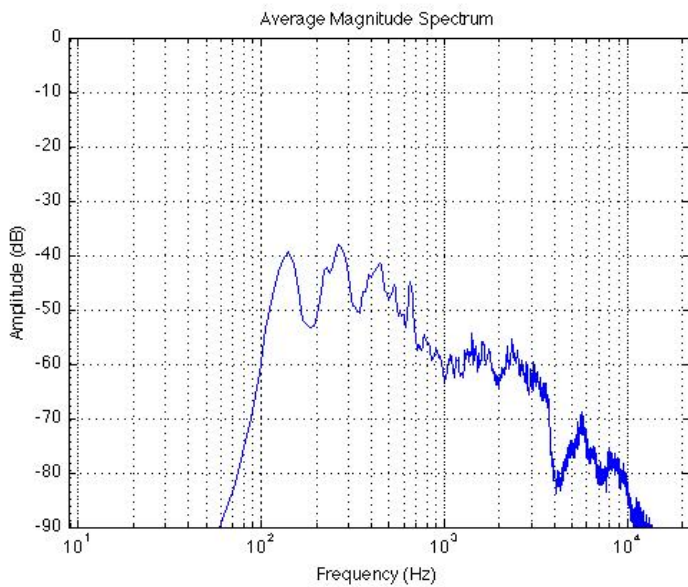


FIG 3. Original 'heroin' Freq. Response response

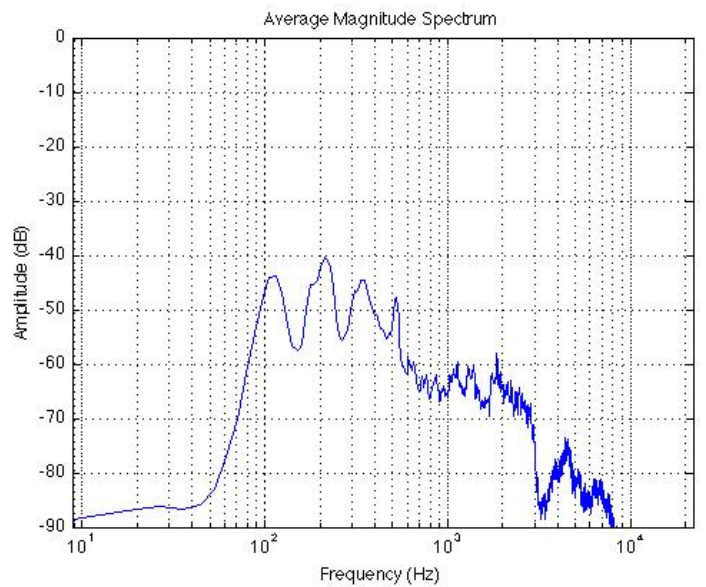


FIG 4. Output 'ShiftedHeroin' freq.

Figures 3 and 4 show the frequency response before and after the pitch shift effect. We can clearly see that the pitch of the signal has been lowered as desired, while the harmonic structure of the signal has been maintained as desired.

Bibliography

- [1] Grondin François, “Guitar Pitch Shifter”, Matlab code and Mathematic Equations <http://www.guitarpitchshifter.com/matlab.html>, accessed 1/4/2012
- [2] *DAFX: Digital Audio Effects*, Chapter 1, West Sussex: John Wiley & Sons, 2002.
- [3] *The Scientist and Engineer's Guide to DSP*, Chapter 3, Stephen Smith, Digital Book.