iSing Voice Harmonizer

Cyril Lan, Jessie Li, Darren Yin
Overview

1. Read in voice signal through microphone
2. Read in harmonics through keys on keyboard
3. FFT to detect pitch of voice signal
4. Pitch shift N copies of voice signal for each of the N keys pressed on keyboard
5. Inverse FFT to get back pitch shifted signals, blend, and output!
Overall Block Diagram
Fast Fourier Transform Module

Diagram of Fast Fourier Transform Module:
- Buffer
- BRAM (32 x 512) x 2
- Butterfly
- Addresser
Fast Fourier Transform Butterfly
## Fast Fourier Transform - Bit shifting

<table>
<thead>
<tr>
<th>Stage 1</th>
<th>Stage 2</th>
<th>Stage 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LCIRC 2</strong>&lt;br&gt;CBA ACB</td>
<td><strong>LCIRC 1</strong>&lt;br&gt;CBA BAC</td>
<td><strong>LCIRC 0</strong>&lt;br&gt;CBA CBA</td>
</tr>
<tr>
<td>000 000 = 0</td>
<td>000 000 = 0</td>
<td>000 000 = 0</td>
</tr>
<tr>
<td>001 100 = 4</td>
<td>001 010 = 2</td>
<td>001 001 = 1</td>
</tr>
<tr>
<td>010 001 = 1</td>
<td>010 100 = 4</td>
<td>010 010 = 2</td>
</tr>
<tr>
<td>011 101 = 5</td>
<td>011 110 = 6</td>
<td>011 011 = 3</td>
</tr>
<tr>
<td>100 010 = 2</td>
<td>100 001 = 1</td>
<td>100 100 = 4</td>
</tr>
<tr>
<td>101 110 = 6</td>
<td>101 011 = 3</td>
<td>101 101 = 5</td>
</tr>
<tr>
<td>110 011 = 3</td>
<td>110 101 = 5</td>
<td>110 110 = 6</td>
</tr>
<tr>
<td>111 111 = 7</td>
<td>111 111 = 7</td>
<td>111 111 = 7</td>
</tr>
</tbody>
</table>
Butterfly Module - Optimizing for speed
CPU Description
Pitch Shifting

Time domain

Initial signal

Frequency Domain

Amplitude

-f_2 \rightarrow -f_1 \rightarrow -f_0 \rightarrow f_0 \rightarrow f_1 \rightarrow f_2

Frequency

Pitch shifted signal

Amplitude

-f_2 \rightarrow -f_1 \rightarrow -f_0 \rightarrow f_0 \rightarrow f_1 \rightarrow f_2

Frequency

\[ 1.0594 \times L \]
Pitch Shifting Module

- Module takes in target frequency played on keyboard and shifts sung note frequency to match keyboard frequency.
- Apply Phase Vocoder Algorithm to produce pitch shifted signal
Phase Vocoder Algorithm

- 3 Stages: Analysis, Processing, Synthesis

- Analysis: Apply Hanning window to extract a small frame of time domain signal with most of the energy focused around DC component.

- Processing: Apply a DFT to divide up the frequency spectrum into a series of discrete bins each with magnitude and phase information.

- Since the frequency components of the signal may not coincide exactly with the bin frequencies, we need to calculate the true frequencies associated with the bins.
Phase Vocoder (cont.)

- After calculating the true bin frequencies using phase offsets, a new spectrum is obtained.
- Synthesis: Apply Inverse FFT to obtain time domain signal for a particular frame and multiply with window to smooth out the signal.
- Add the windowed signals together to reconstruct the entire time domain signal.
Timeline

- 11/25: Finalize implementation details and implement midi controller module. Have skeleton code for other modules.
- 12/4: Finish implementing FFT, Pitchshifter, and CPU modules
- Week of 12/5 – 12/9: Integration testing