p_shifter.cpp
⟨ Audio pitch converter module ⟩
User guide

Kaoru Ashihara

November 4, 2021

‘p_shifter.cpp’ can be used as the audio pitch converter module. By using it along with ‘winmm.lib,’ the pitch of the sound can be converted in semitone steps between -36 semitones and 36 semitones during either the recording or the reproduction of the sound.

‘p_shifter.cpp’ is open source under the MIT license.

1 Header file

To use ‘p_shifter.cpp’ in your C++ project, it is necessary to prepare the header file of the project. Add the lines below to the header file so that these functions can be used in the project.

```
void initialize(HWND hWnd);
void firNoSetter(HWND hWnd, short sNo);
void genFunc(HWND hWnd, short sTap);
void genFir(HWND hwnd, int iParam, short sTap);
DWORD convolve(HWND hWnd, LPSTR lpOrigi, LPSTR lpData, LPSTR lpBuf, DWORD dwOffset, DWORD dwUnt, short sCurrPit, short sTap);
void finalize(HWND hWnd);
```

Then include the header file to ‘p_shifter.cpp’ by modifying the following line in ‘p_shifter.cpp if necessary.’

```
#include “ p_shifter.h ” // Include the header file of your project
```

‘p_shifter.h’ must be replaced by the name of the header file of your project.

2 Functions

‘p_shifter.cpp’ has the functions listed below.

- void initialize(HWND hWnd)
- void firNoSetter(HWND hWnd, short sNo)
- void genFunc(HWND hWnd, short sTap)
- static BOOL fastFt(HWND hwnd, double real[], double image[], short sTap, BOOL isInv)
- static int gcd(HWND hwnd, int x, int y)

- void genFir(HWND hwnd, int iParam, short sTap)
- DWORD convolve(HWND hWnd, LPSTR lpOrigi, LPSTR lpData, LPSTR lpBuf, DWORD dwOffset, DWORD dwUnt, short sCurrPit, short sTap)
- void finalize(HWND hWnd)

The functions ‘initialize,’ ‘genFunc,’ and ‘genFir’ have to be called before starting either the recording or the reproduction and before calling the function ‘convolve.’ The functions ‘fastFt’ and ‘gcd’ are used in ‘genFir.’

In ‘genFir,’ time-varying FIR filters are generated. These filters are iUp[iParam] and iDown[iParam] where ‘iParam’ is an integer between 0 and 36 inclusive. By conolving the FIR filter with the audio data, the pitch-converted data are generated. ‘genFir’ has to be called, therefore, before starting either the recording or the reproduction.

The function ‘firNoSetter’ has to be called whenever the pitch shift amount is changed during the recording or the reproduction. After the recording or after the reproduction, the function ‘finalize’ can be called to delete the FIR filters.

Convolution of the FIR filter with the audio data is done in the function ‘convolve’ and this function will be called whenever the application receives ‘MM_WIM_DATA’ or ‘MM_WOM_DONE.’

3 Variables

3.1 winFunc

‘winFunc[]’ is the window function that is provided by the function ‘genFunc.’

3.2 Channels

‘Channels’ reppresents the number of channels in the audio data. It has to be either 1 or 2.

3.3 iLev, iRelease, iElapsed, and isToRewind

Timing to rewind the time-varying filter can be manipulated by using the variables ‘iLev,’ ‘iRelease,’ ‘iElased,’ and ‘isToRewind.’ ‘iElapsed’ is the elapsed time in sample during which the level of the input signal does not exceed the threshold level represented by the variable ‘iLev.’ When the value of ‘iElapsed’ reaches that of ‘iRelease,’ the BOOL variable ‘isToRewind’ becomes TRUE. When the level of the sound exceeds the threshold level for the first time after ‘isToRewind’ became TRUE, the time-varying filter is forced to be rewound. For example, the variable ‘sFirNo’ in ‘iUp[sCurrPitch][sFirNo]’ is forced to be 0. At the same time, ‘iElapsed’ is reset to 0. Once the filter is rewound, ‘isToRewind’ is FALSE again.

3.4 iUp, and iDown

As mensioned above, iUp[sCurrPitch][sFirNo] and iDown[sCurrPitch][sFirNo] are the FIR filters. The value of ‘sCurrPitch’ represents the pitch shift amount. For instance, iUp[3] is the filter to shift the pitch upward by 3 semitones and iDown[12] is the filter to shift the pitch downward by one octave (one octave equals 12 semitones).

Since they are the time-varying filters, each filter has several subsets. The variable ‘sFirNo’ indicates which subset is in use and it increases by 1 at every sample period.

3.5 Other variables

The variables ‘sFirNo’ and ‘sAdv’ are used in the function ‘convolve’ and their values are automatically updated in this function. ‘sFirNo’ can be reset by the function ‘firNoSetter.’

The values of the variables ‘sNumUp[],’ ‘sNumDown[],’ ‘sShiftUp[],’ and ‘sShiftDown[]’ are provided in the function ‘genFir’ and used in ‘convolve.’

4 License of the program