

p_shifter.cpp 1.0.6
〈 Audio pitch converter module 〉
User guide

Kaoru Ashihara

July 28, 2022

The C++ source file ‘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 and the external variables can be used in the project.

```
void initialize(HWND hWnd, int iLvl, int iRel, int iMrk, unsigned
ch, BOOL flg);
void genFunc(HWND hWnd);
BOOL prepFir(HWND hwnd,short sPitch);
BOOL prepFir(HWND hwnd);
void genFir(HWND hwnd, short sParam);
DWORD convolve(HWND hWnd, LPSTR lpOrig, LPSTR lp-
Data, LPSTR lpBuf, DWORD dwOffset, DWORD dwUnt, short
sCurrPit);
void finalize(HWND hWnd);

extern short sNumTaps;
extern BOOL bFlg;
extern BOOL isAllowed;
```

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

In ‘p_shifter.cpp,’ there are the functions listed below.

- void initialize((HWND hWnd, int iThr, int iRel, int iMrk, unsigned ch, BOOL flg))

- `BOOL genFunc(HWND hWnd)`
- `static BOOL fastFt(HWND hwnd, double real[], double image[], BOOL isInv)`
- `static int gcd(HWND hwnd, int x, int y)`
- `BOOL prepFir(HWND hwnd, short sPitch)`
- `BOOL prepFir(HWND hwnd)`
- `void genFir(HWND hwnd, short sParam)`
- `DWORD convolve(HWND hWnd, LPSTR lpOrig, LPSTR lpData, LPSTR lpBuf, DWORD dwOffset, DWORD dwUnt, short sCurrPitch)`
- `static BOOL malocfir(HWND hWnd, LPSTR *lpFilter, DWORD dwSize)`
- `void finalize(HWND hWnd)`

The functions ‘initialize,’ ‘genFunc,’ ‘prepFir,’ 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’ and the function ‘malocfir’ is called by the function ‘prepFir.’

In ‘genFir,’ the array of time-varying FIR filters is generated and stored to the LPSTR variable ‘lpEven.’ This array can consist of the subsets of the FIR filter. By convolving 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.

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

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

3.2 Channels

‘Channels’ represents 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,’ ‘iElapsed,’ 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. At the same time, ‘iElapsed’ is reset to 0. Once the filter is rewound, ‘isToRewind’ is FALSE again.

3.4 iQ, iO, and iE

Values of the variables ‘iQ,’ ‘iO,’ and ‘iE’ are given in the function ‘genFunc’ and are used in the function ‘convolve.’

3.5 External variables

The variable ‘sNumTaps’ represents the FIR filter size. It must be initialized before calling the functions ‘genFunc,’ ‘prepFir,’ and ‘genFir.’ This variable must be a power of 2, otherwise the BOOLEAN function ‘genFunc’ returns FALSE.

‘bFlg’ is a BOOLEAN variable to check if the pitch shift amount has been changed.

When another BOOLEAN variable ‘isAllowed’ is TRUE, the pitch can be manipulated during either the recording or the reproduction of the sound. When it is FALSE, less memory is consumed by the application.

3.6 Other variables

The DWORD variable ‘dwA’ represents the address of the current FIR filter coefficient. DWORD variables ‘dwAdv’ and ‘dwEnd’ are the addresses of the first and the last coefficients in the filter. The DWORD ‘dwPosit’ is used in the function ‘genFir’ as the index of the filter coefficient.

DWORD arrays 'dwHead[]' and 'dwTail[]' are used as the address of the first and the last coefficient in each subset of the filters, respectively. The subscript of these array identifies the subset of the filters. When the value of the pitch shift amount 'n' is smaller than 0, the subscript of the array must be 36 - n, otherwise it is n. For example, if the pitch shift amount is -3 semitones, dwHead[39] and dwTail[39] are used.

4 License of the program

MIT license

Copyright©2022, AIST

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.