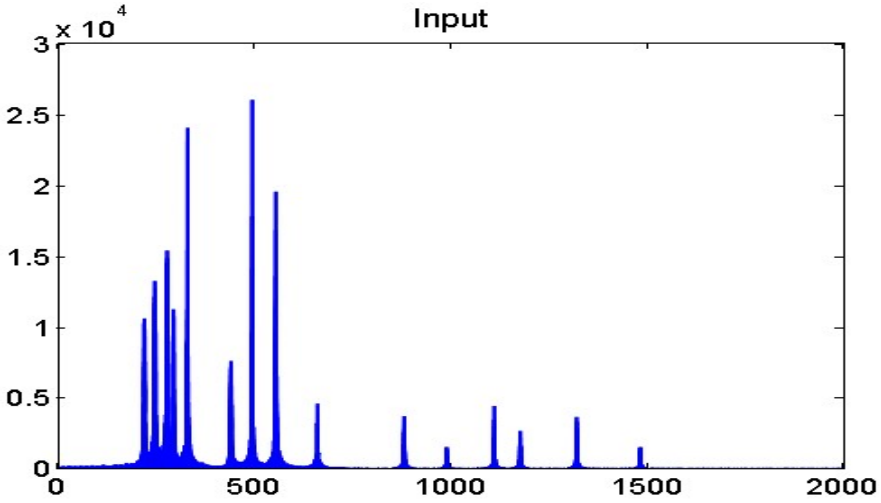
Multimedia Assignment2 103070028 莊筑鈞

1.

ⅰ. Determine : There are three songs in the input signal, and I know one pitch is low, one is moderate, the other is high after I listened. So, I guess that I can distinguish the low pitch song by a "Low-pass Filter", cut the moderate pitch one by a "Bandpass Filter", and find the high pitch one by a "High-pass Filter".



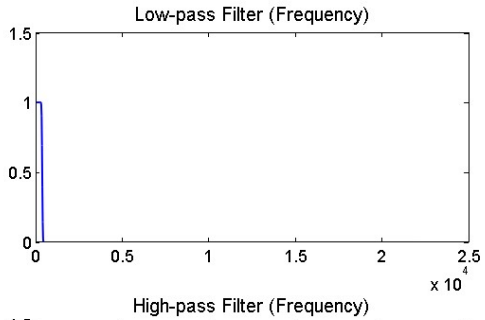
From the figure of input signal, we can see several high point and the signal seems like three parts, and I then guess that 0 to 380 hz is the low pitch song, and so as I use low-pass filter to distinguish the song. Then, I cut 381 to 700 hz as the moderate pitch song with bandpass filter, and 701 hz as the high pitch song with high-pass filter.

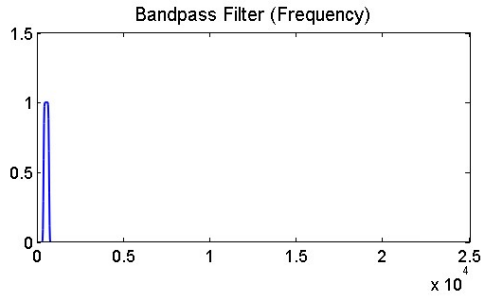
ⅱ. Implement : I choose N=1501 to implement the function, and call the function "my_filter" to create the filter and do the convolution respectively.

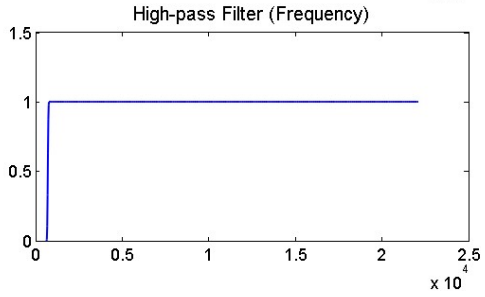
```
[outputSignal, outputFilter] = my_filter(y_input, fs, 1501,'Blackmann', 'low-pass', 380);
%%%bandpass
[outputSignal_pass, outputFilter_pass] = my_filter(y_input, fs, 1501,'Blackmann', 'bandpass', [381 700]);
%%%high-pass
[outputSignal_high, outputFilter_high] = my_filter(y_input, fs, 1501,'Blackmann', 'high-pass', 701);
```

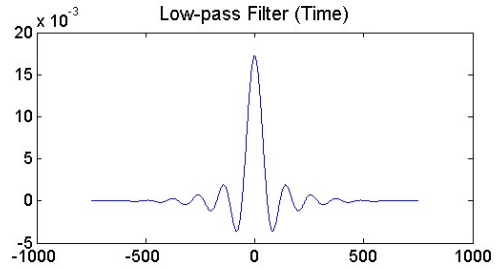
In "my_filter", normalize the "fcutoff" and initialize the "middle" (referenced by the slide p.80) at first, and then I determine which kind of filter is by "strcmp" and create the according filter refers to formulas in the slide. Next, do windowing function (Blackmann) according to the folmulas in the slide,too. Last, do the convolution with the inputsignal and the outputfilter, and save the output into the "outputsignal".

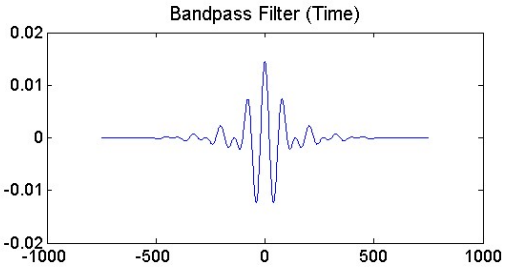
ⅲ. compare of spectrum and shape of filters :

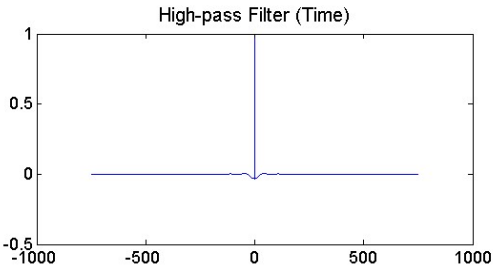












＊Spectrum :

Low-pass filter is to let frequency lower than the filter pass, so only the frequency lower than the filter (I set 380 hz) is 1, frequency which is higher than the filter is all 0.
Bandpass filter is to catch the frequency between an upper bound and a lower bound(here is 381 to 700 hz), and only frequency between this range can pass. So, only the frequency between the range is 1, for the frequency out of the range is all 0.
High-pass filter is to let frequency higher than the filter pass, so only the frequency higher than the filter is 1, other frequency is all 0.

In addition, if the filter size is larger, the effect of filter is better, but it'll run for a very long time if the filter size is too large. So, choosing a filter size is very related to the signal after filtered.

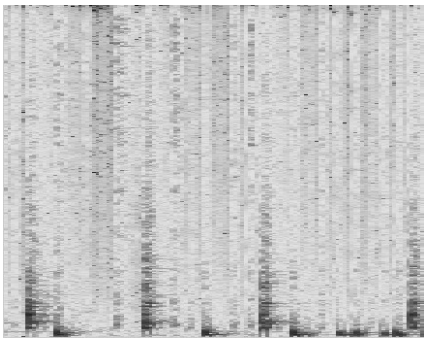
＊Time domain :

In the time domain, the low-pass filter is a sinc function, and there are two functions subtracted in the bandpass filter, and high-pass filter is a negative sinc function. So, that's why the figures in time domain look like.

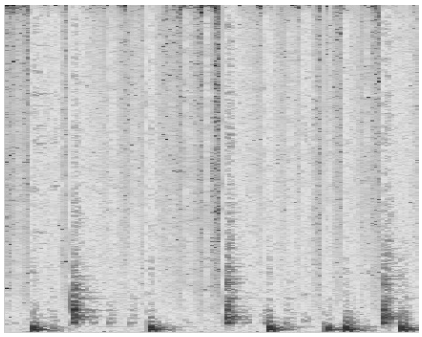
2.

ⅰ. The observation of classes of different instruments :

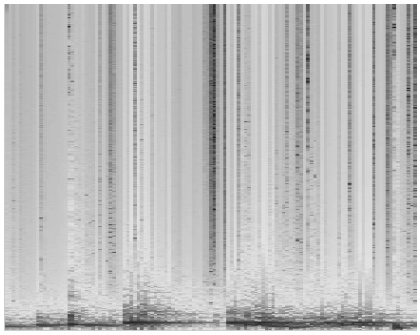
The spectrogram of drum is the most obviously different of the four instruments, and it has many intensive vertical line in the figure.



drum_01

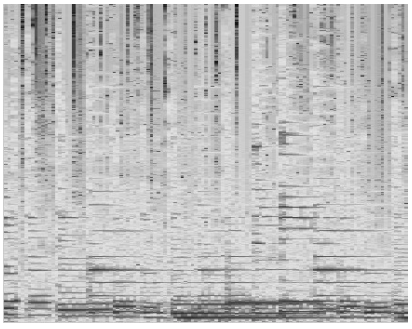


drum_02

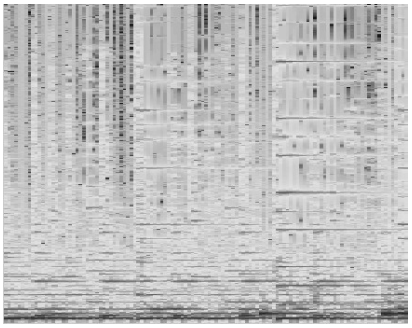


drum_03

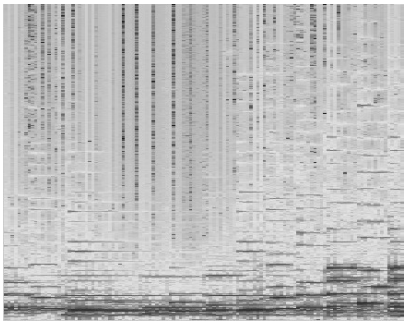
The spectrograms of guitar and piano seem a little like, they are all full of criss-crossed vertical and horizontal lines, which cut the figure into many little blocks with different size. The difference is that the horizontal lines of the figure of guitar are everywhere in the picture, but the horizontal lines of the piano is mostly distribute in the lower of the picture.



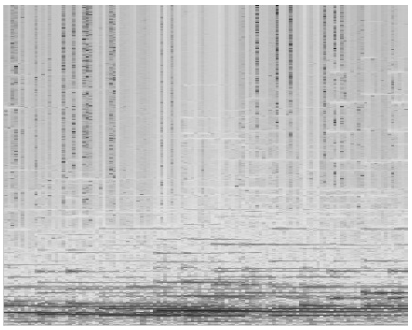
guitar_01



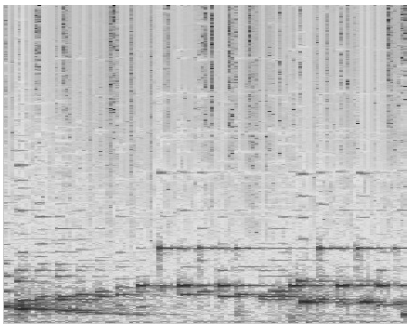
guitar_02



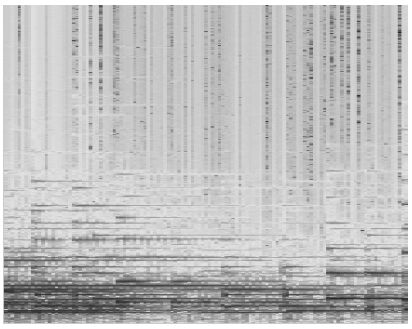
guitar_03



piano_01

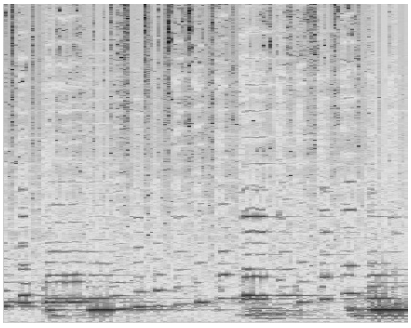


piano_02

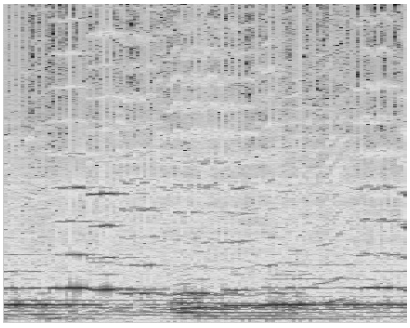


piano_03

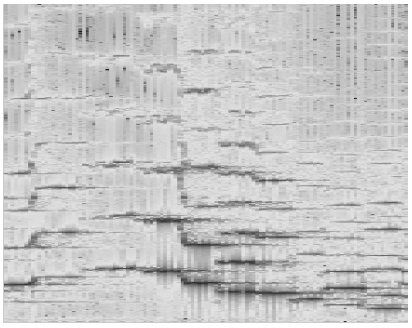
The spectrogram of violin also have many horizontal lines in everywhere, but the horizontal lines are not straight, they are a little wavy and longer than the horizontal lines I observed in the figure of guitar. The horizontal lines look like many long waves.



violin_01



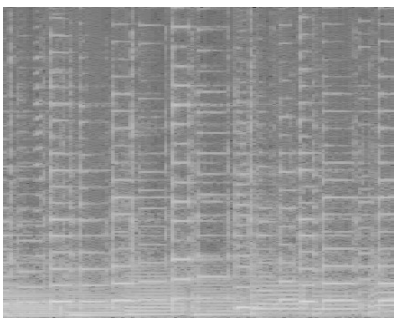
violin_02



violin_03

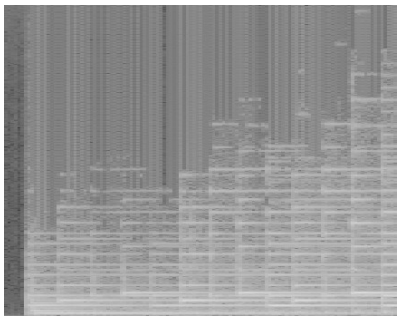
ⅱ.distinguish the test spectrogram :

test_spectrogram_01



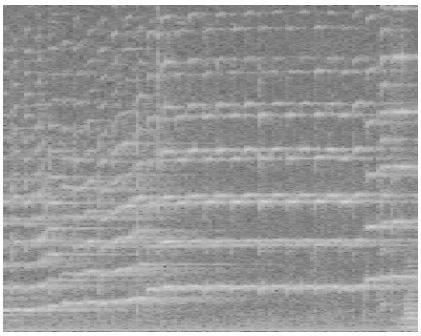
From the observation of (i)(horizontal lines everythere in figure), we can find that the "test_spectrogram_01" as "guitar".

test_spectrogram_02



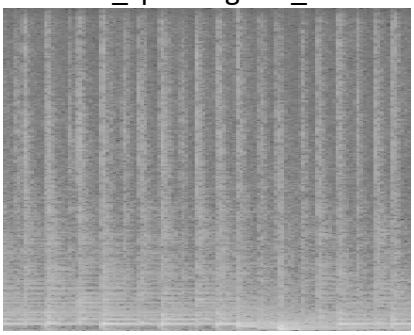
From the observation of (i)(horizontal lines in the lower part of figure), we can find that the "test_spectrogram_02" as "piano".

test_spectrogram_03



From the observation of (i)(waves), we can find that the "test_spectrogram_03" as "violin".

test_spectrogram_04



From the observation of (i)(all vertical lines), we can find that the "test_spectrogram_04" as "drum".

ⅲ. Implement of STFT :

First, I use a loop to cut total signal into segments with the length of "segment_duration-segment_overlap"(這次切到下次切的距離) and use am array "w" to temporarily save the segment we cut. Then, I use "Hanning windowing function" to do windowing with the formula in the slide, and multiply with "w" array, and then save into

"w" array again. Then, call "fft" function to do FFT with "w" and saved into an array "f". Then, put "f" into the all the rows with according column of the "S" matrix. After the loop runs out, they'll be merged into a completed matrix "S" and accomplish STFT.

```
for i=0:segment_duration-segment_overlap:length(x)-segment_overlap-1
    if(i==0)
        w=x(i+1:i+segment_duration);
    else
        w=x(i+1:i+segment_duration);
    end
    %Hanning windowing function
    for j=0:segment_duration-1
        w(j+1)=w(j+1)*(0.5+0.5*cos(2*pi*j)/segment_duration);
    end
    %FFT
    f=fft(w, m);
    S(:, count) = f(1:s_r);
    count=count+1;
end
```

ⅳ. Reference :

參考

1.http://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_Ch16.pdf

2.http://www.mathworks.com/matlabcentral/fileexchange/45197-short-time-fourier-transformation--stft--with-matlab-implementation?focused=3891027&tab=function 網站中的

```
% form the stft matrix
rown = ceil((1+nfft)/2);            % calculate the total number of rows
coln = 1+fix((xlen-wlen)/h);        % calculate the total number of columns
stft = zeros(rown, coln);           % form the stft matrix
```

以及

```
% calculate the time and frequency vectors
t = (wlen/2:h:wlen/2+(coln-1)*h)/fs;
f = (0:rown-1)*fs/nfft;
```

3. http://www.360doc.com/content/11/0215/10/5521449_93148675.shtml 網站中的 FFT 原理