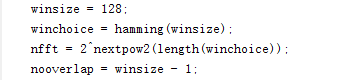
Audio Signal Processing project 1

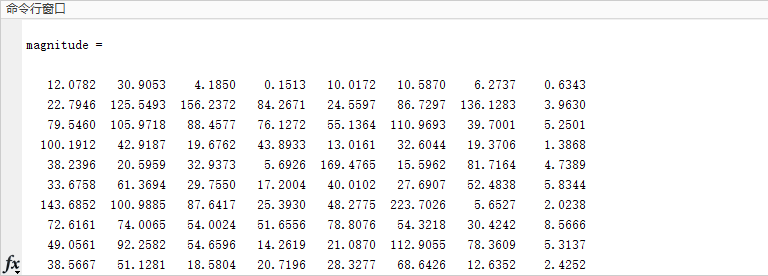
Qihua Gong

Part 1

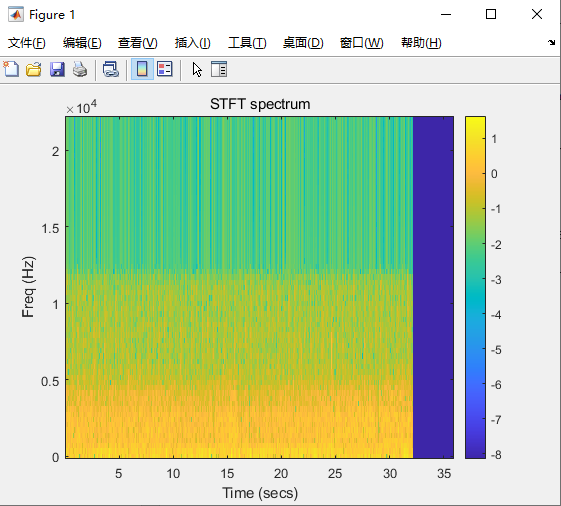
1. Compute the magnitude short-term spectrum of each signal
   1. This is the first part of the whole project, which we have 12 audio files in MP3 format. What we have to do is to import them into MATLAB and calculate their magnitude of short-term spectrum.
   2. The short-time Fourier transform (STFT) is to first multiply a function and a window function, and then perform a one-dimensional Fourier transform. A series of Fourier transform results are obtained through the sliding of the window function, and a two-dimensional representation is obtained by arranging these results. Specifically, it can be divided into the following steps:
      1. Read audio files. (Call audioread, process to get an array to save audio data, and a sampling frequency)
      2. Determine the relevant parameters. (Such as window function, window length, number of overlap points, overlap length, number of Fourier points, etc.)
      3. Do short-time Fourier transform. (S-Two-dimensional array data containing time and frequency sequence obtained after STFT processing the input signal)
      4. Draw a spectrogram based on the processed time-frequency matrix.
   3. The code part work on this part is project1mag.m. The use of this function is to calculate the magnitude spectrum of STFT for the given audio. I first read the audio signal data and Fs audio sampling rate from the audio. Let's set the basic parameters of the calculation formula, such as window size, window choice and FFTLength and OverlapLength. The base setting of these parameters in the project1mag.m, we will change and compare the parameters later in this paper.



* 1. The calculated magnitude is a matrix of numbers.



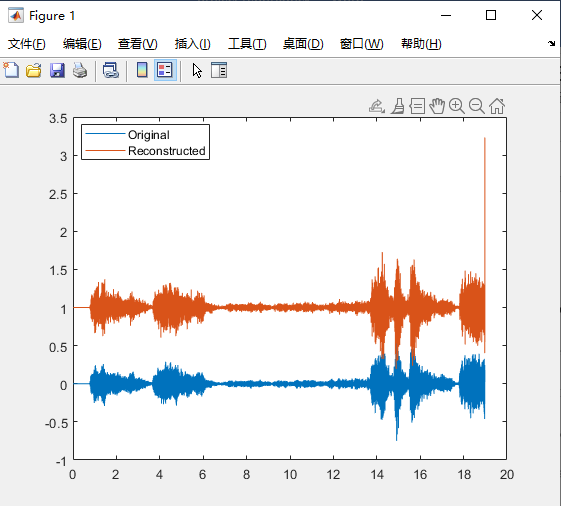
In order to show the magnitude results more clearly, I did a drawing through MATLAB. Set x-label to Times and y-label to Frequency, and color the drawing. Taking audio1 as an example, we will get the following graph:



It can already clearly show the audio level. All the other results saved in the file magnitude.

* 1. Finally I do the Magnitude compare with different parameters. The compare results all save in the file compare. I mainly change the Win size and the Window choice, which affect the result most.

1. Now let’s move forward to the part 1 section 2: reconstruct the signal back.
   1. To reconstruct the signal, a more common method is the OLA (Overlap-and-Add) method. In simple terms, when the original signal is framed, the adjacent two frames have a part of the overlap area. For the voice signal Processing, under normal circumstances, this overlap area is 50% or 75%, then when the signal is reconstructed, after each frame is inversely transformed, the corresponding superposition is also required. Therefore, the enhanced part of the overlapping part is related to the selected window function.
   2. The main code for the reconstruction part is project1rec.m, and I saved the reconstructed file in the folder ReconstructAudio, you can listen to it. Although the restoration is not good, the content of the sound can be heard roughly. I think I nearly reconstruct the audio. I make a signal compare graph which shows the result more clearly:

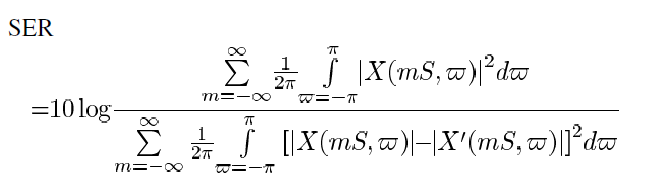


* 1. After my experiment, I found that the main influence on sound restoration is the choice of these two lines of parameters (beside: different window can cause different result):

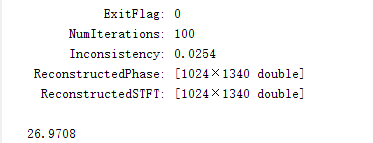


* 1. The function hamming(n) is a window function, where n can be described as dividing the waveform into n blocks and sampling each block. Therefore, the larger the n, the more sampling times (sampling frequency), and the more restored the reconstructed signal. And this Overlap length is also required. According to the transformation characteristics of STFT, the more sampling times (sampling frequency), the easier it is for adjacent waveforms to overlap, and the overlapping part will be distorted, or because adjacent waveforms are in Fourier It is easy to be changed into a similar frequency during the change, causing distortion of the original signal. If you want to restore better, you can also increase the window size of hamming. Note that the overlap length should also be increased at the same time, and you can adjust the amount of increase yourself. My experiment is that when winsize is 128 and ovlength is 64, the restoration sound is too blurry. When the current parameters 1024 and 400 are selected, a relatively satisfactory result is obtained. Adjusting further up my MATLAB stuck and loses response.

1. Part 1 section 3: SER calculate
   1. Just follow the function below and create the code, then calculate:



* 1. My calculate result example audio 3 G&L algorithms with SER:



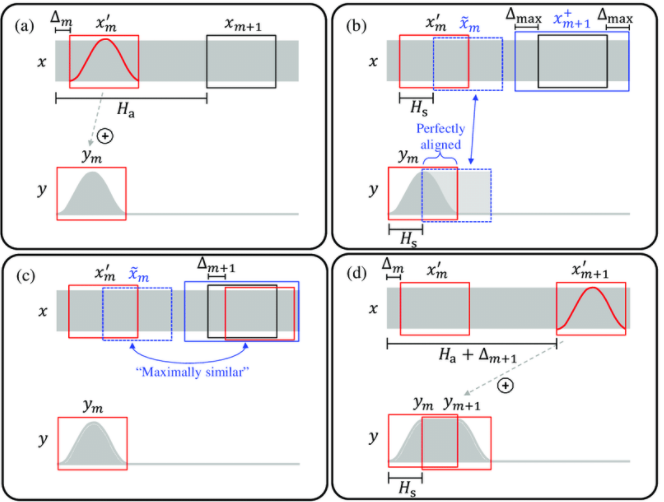
A bit higher with the result in paper.

* 1. I think the biggest problem may be that the audio source you gave me is in MP3 format and has multiple channels. In order to be able to implement the previous process, I turned them all into a single track. The loss of the process in the middle caused the SER to be too large.

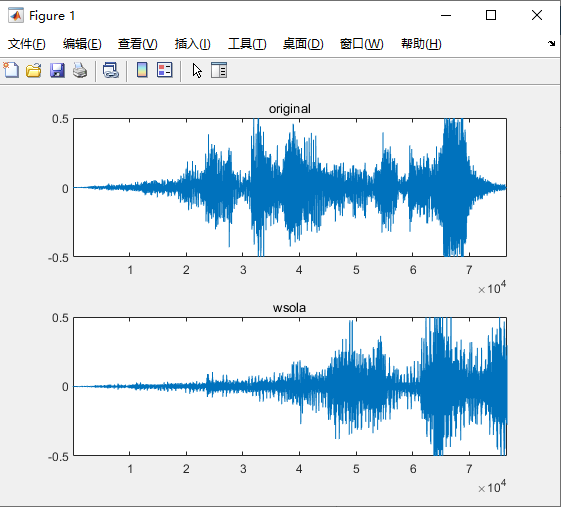
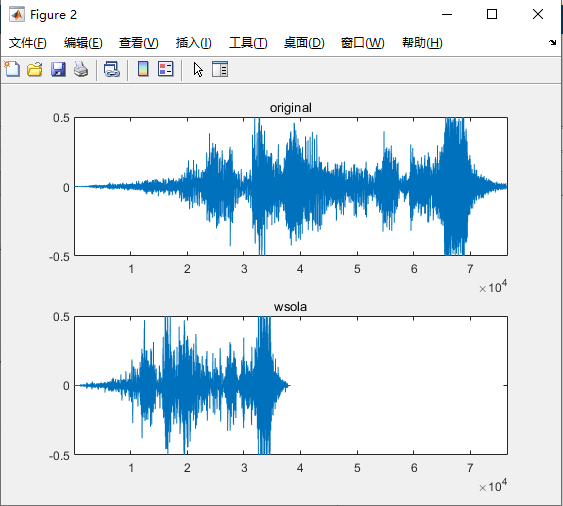
1. Part 2: TSM apply
   1. I chose Wsola algorithm for variable speed. The operation steps are:
      1. Intercept a frame of signal x’m from the original signal (as shown in figure a above);
      2. Find a frame of signal x~m at a distance of Hs from x’m, and find a frame of signal xm+1 at a distance of Ha from x’m (as shown in figure a above);
      3. In the blue area x+m+1 with xm+1 as the center and extending Δ \DeltaΔmax on both sides (as shown in figure b above), find a frame of x'm+1 that is most similar to x~m, which is the figure above. The red frame in the blue box in c is the most similar to the blue dashed frame;
      4. Superimpose the found x’m+1 at a distance Hs from x’m.

rate = Ha / Hs Lout = Lin / rate

Hs is fixed, usually half of the frame length. The rate is used to control whether the output signal becomes faster or slower. When rate <1, the output signal becomes slower, otherwise it becomes faster.



1. The way that introduced on Zhu’s paper is more like the OLA algorithm. To compare with the OLA algorithm, it is often impossible to maintain the periodic structure in the original signal. In the process of frame cropping, it is impossible to guarantee that each frame can cover the complete cycle and ensure its phase alignment, which will cause phase skipping distortion. In order to ensure that the fundamental frequency remains unchanged, the WSOLA algorithm must be used. WSOLA does not directly find a frame at Ha for direct splicing, but finds the closest frame to the ideal frame in an interval for superimposition.
2. Here is the result graph of my Wsola code result: (Also saved in file TSMsignal.)



Any rate factors you want, just change the parameters Rate in file TSM.m