Mini Project 2 - Spectrogram

1. Objectives

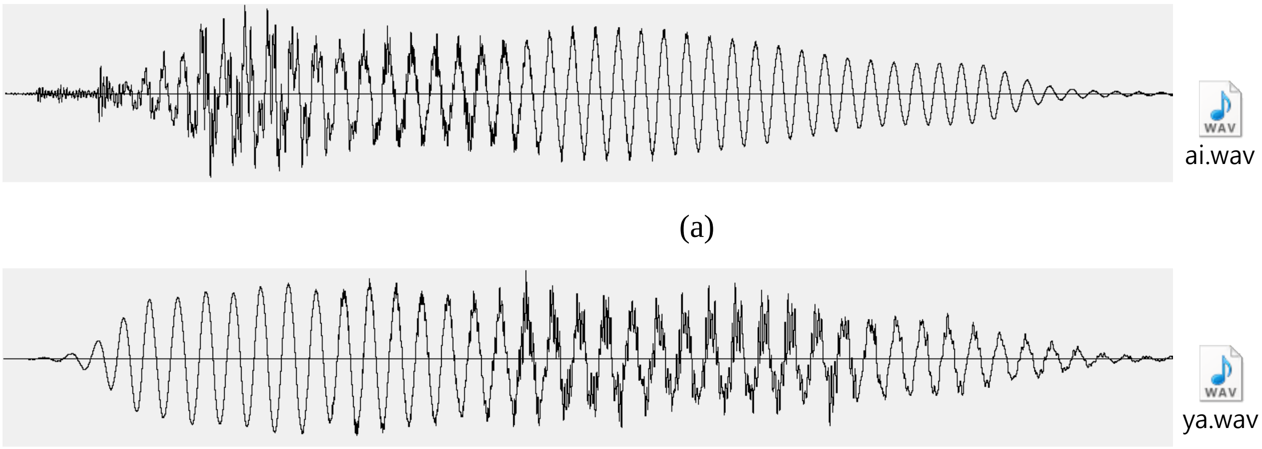
(a) Be familiar with Discrete Fourier Transform (DFT).

(b) Understand how a spectrogram is generated.

(c) Mastery over the C language.

2. Background

Fig.1 shows waveforms of a compound vowel ‘ai’ and a syllable ‘ya’. The compound vowel ‘ai’ and the syllable ‘ya’ have very similar sound components ‘a’ and ‘i’, but are in different order over time. It can be obviously observed that the waveform in the beginning of ‘ai’ and the one in the end of ‘ya’ are very similar, that is, they may have similar harmonic component, since they all are the sound ‘i’ (vice versa for the case of the end of ‘ai’ vs. the beginning of ‘ya’. These two waveforms are different speech segment. But, if you take Fourier transform for ***whole segments of ‘ai’ and ‘ya’***, they may have very similar frequency responses in Discrete-Time Fourier Transforms (DTFTs), indicating that we cannot distinguish the two sound ‘ai’ and ‘ya’ if the analysis window is too long so as to make Fourier analysis lose ***time resolution*** due to the fact that speech signals change over time.



(a)

(b)

Fig.1: (a) waveform of a compound vowel ‘ai’ and (b) waveform of a syllable ‘ya’. Note that ‘ai’ consists of two sounds ‘a’ and ‘i’ in sequence and ‘ya’ consists of two sounds ‘i’ and ‘a’ in sequence.

To catch the change in frequency components over time, as shown in Fig. 2, ***Short-Time Fourier Analysis*** or ***Short-Time Fourier Transform (STFT)*** is adopted to analyze a speech signal by decomposing the speech signal into a series of short segments, referred to as ***analysis frames***, and analyze each one independently. Apparently, the waveform in each short segment (analysis frame) could be more stationary since waveforms in a shorter window look more periodic and have more similar harmonic components. Also, each analysis frame could catch different waveform so as to observe signal changing over time. We, therefore, can analyze phonemes and their transition by spectrogram.

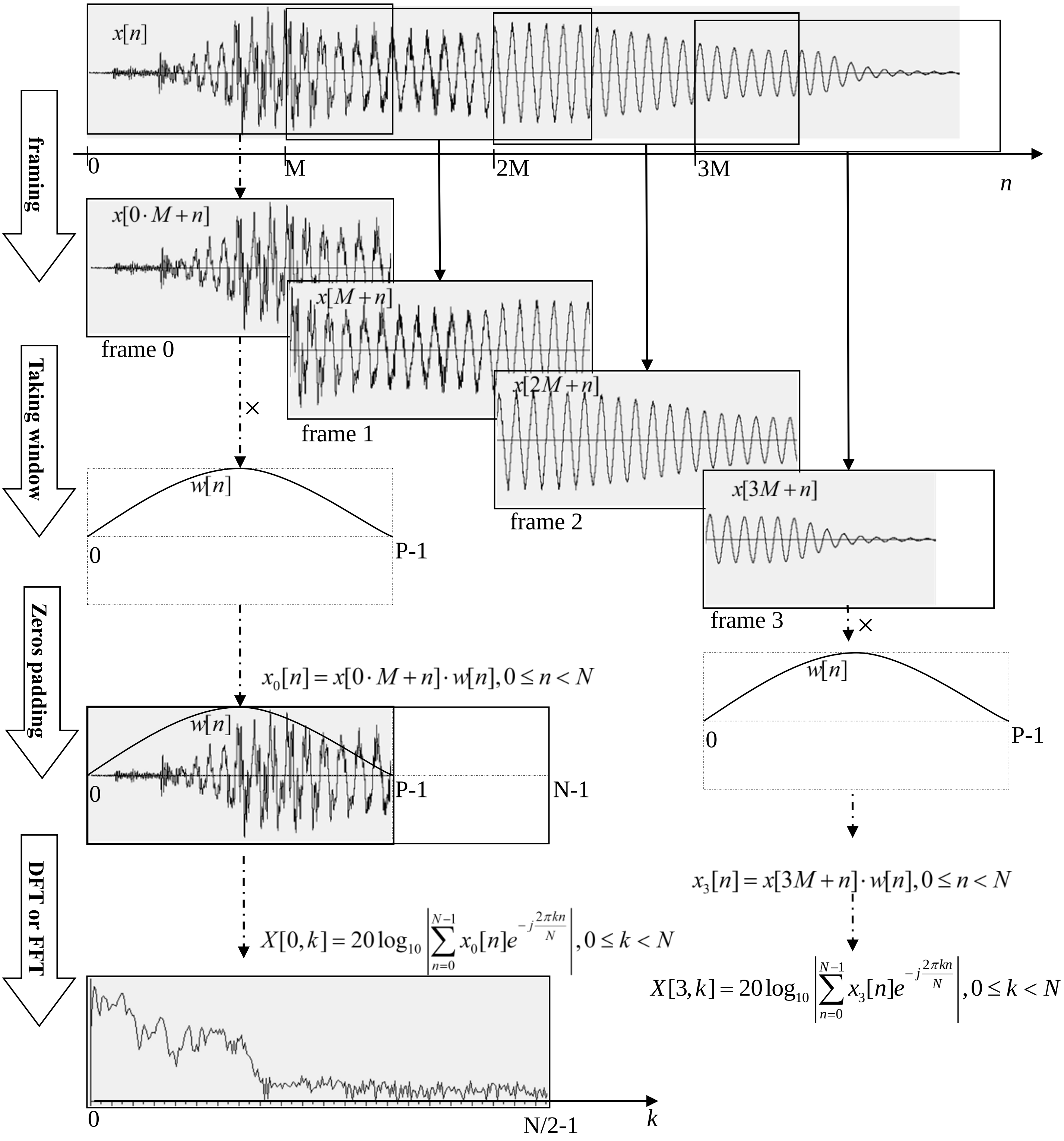
A spectrogram is a visual representation of a ***spectrum sequence***, capturing spectrum changes with time. Spectrograms are sometimes called sonogram, spectral waterfalls, voiceprints, or voicegrams. In Digital Signal Processing (DSP), we can regard spectrogram, $X[m,k]$, as a function of time (frame index) and frequency bin, i.e.

$$X[m,k] = 20\log_{10}\left|\sum_{n=0}^{N-1} x_m[n]e^{-j\frac{2\pi kn}{N}}\right|, \qquad 0 \le k < N \tag{1}$$

where *m* represents frame index; *k* represents frequency bin index. Note that $\sum_{n=0}^{N-1} x_m[n]e^{-j\frac{2\pi kn}{N}}$ in Eq. 1 is in the same form with DFT. $x_m[n]$ is a short-time signal with a length of *N* samples, indicating taking DFT of a discrete time signal of length *N*. We usually call the parameter *N* '*FFT window length*'. Note that FFT stands for Fast Fourier Transform which is an efficient implantation of DFT. In mathematics, $x_m[n]$ is expressed by

$$x_m[n] = x[m\cdot M + n]\cdot w[n], \qquad 0 \le n < N \tag{2}$$

where $x[m\cdot M + n]$ denotes a time-shifted signal of original input speech, $x[n]$; *M* represents *frame interval* in sample; $w[n]$ is a window function, a mathematical function with zero-valued outside of some chosen interval. Therefore, we can regard $x_m[n]$ as *m*-th frame signal of length *N*. There are many types of window functions, such as rectangular window (Eq. 3), Hamming window (Eq. 4), and Hanning window (Eq. 5).



$x[n]$
0
M
2M
3M
n
framing
$x[0\cdot M+n]$
$x[M+n]$
frame 0
frame 1
$x[2M+n]$
frame 2
$x[3M+n]$
frame 3
Taking window
×
$w[n]$
0
P-1
×
$w[n]$
0
P-1
Zeros padding
$x_0[n]=x[0\cdot M+n]\cdot w[n], 0\le n<N$
$w[n]$
0
P-1
N-1
$x_3[n]=x[3M+n]\cdot w[n], 0\le n<N$
DFT or FFT
$X[0,k]=20\log_{10}\left|\sum_{n=0}^{N-1}x_0[n]e^{-j\frac{2\pi kn}{N}}\right|, 0\le k<N$
$X[3,k]=20\log_{10}\left|\sum_{n=0}^{N-1}x_3[n]e^{-j\frac{2\pi kn}{N}}\right|, 0\le k<N$
0
N/2-1
k

Fig.2: Overview of obtaining spectrogram. The process in sequence, includes framing, taking window, zero padding, and DFT or FFT.

$$\text{Rectangular:}\quad w[n]=\begin{cases}1, & 0\le n<P\\ 0, & \text{otherwise}\end{cases}\tag{3}$$

$$\text{Hamming:}\quad w[n]=\begin{cases}0.54-0.46\cos(\frac{2\pi n}{P-1}), & 0\le n<P\\ 0, & \text{otherwise}\end{cases}\tag{4}$$

$$\text{Hanning:}\quad w[n]=\begin{cases}0.5-0.5\cos(\frac{2\pi n}{P-1}), & 0\le n<P\\ 0, & \text{otherwise}\end{cases}\tag{5}$$

For analysis of speech, we usually choose Hamming window as the window function in Eq. 2. Note that in Eqs 3-5, a window function has zero-value outside of an interval $0\le n<P$. We therefore define *P* as ***analysis window size***, regarding how many sample points are excerpted from the analyzed signal $x[n]$ for an analysis window. Thus, the value *P* must be equal or smaller than the FFT window length *N*. A analyzed signal is said to be zero-padded as *P*<*N* because $x_m[n]=x[m\cdot M+n]\cdot w[n]$ is zero for $P\le n<N$. Since we only have non-zero samples for $0\le n<P$, only *P* points excerpted from $x[n]$ are informative for frequency analysis. As FFT window length *N* is set to be greater than analysis window size *P*, the *N*-*P* zero-padded samples in $x_m[n]=x[m\cdot M+n]\cdot w[n]$ just let the spectrum of *m*-th frame $X[m,k]$ has a smoother outline extrapolating more frequency bins.

3. Parameters of spectrogram by DFT

- Analysis Window Types

Why not using rectangular window for spectrogram? Assume we have a pure sinusoidal signal to be analyzed by spectrogram, e.g.

$$x[n]=\cos(\omega_0 n)\tag{6}$$

and the corresponding Fourier transform is

$$X(e^{j\omega}) = \sum_{k=-\infty}^{\infty} \pi\delta(\omega - \omega_0 + 2\pi k) + \pi\delta(\omega + \omega_0 + 2\pi k) \tag{7}$$

But that spectrogram analyzes the windowed signal $x_m[n] = x[m \cdot M + n] \cdot w[n]$ and this windowed signal is a distorted version of original pure sinusoidal signal $x[n]$. Also, let's recall the convolution theorem, i.e.

$$x_0[n] = x[n] \cdot w[n] \xrightarrow{FT} X_0(e^{j\omega}) = \frac{1}{2\pi} X(e^{j\omega}) * W(e^{j\omega}) = \frac{1}{2\pi}\int_{\omega=-\pi}^{\pi} X(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta \tag{8}$$

From Eq. (9), we know that the DTFT $X_0(e^{j\omega})$ depends on the DTFT $W(e^{j\omega})$ (DTFT of the window function $w[n]$). So, let's check the $W(e^{j\omega})$ if $w[n]$ is a rectangular function of length *P*:

$$\begin{aligned} W(e^{j\omega}) &= \sum_{n=0}^{P-1} w[n]e^{-j\omega} = \sum_{n=0}^{P-1} 1 \cdot e^{-j\omega} = \frac{1 - e^{-j\omega P}}{1 - e^{-j\omega}} \\ &= \frac{e^{-j0.5\omega P}(e^{+0.5\omega P} - e^{-0.5\omega P})}{e^{-j0.5\omega}(e^{+j0.5\omega} - e^{-j0.5\omega})} = e^{-j0.5\omega(P-1)} \frac{\sin(0.5\omega P)}{\sin(0.5\omega)} \end{aligned} \tag{9}$$

Apparently, $W(e^{j\omega})$ is in a form of a sinc function. **So….Think about the rest by yourself. I talked too much!**

- Window Size *N*, frequency bin index *k*, and analysis bandwidth

The parameter *N* is the number of points for DFT analysis (sometime called '***window size***'). Since each frequency bin index *k* in $X[m,k]$ corresponds to a frequency component of $\frac{2\pi k}{N} \times f_s$ Hz of continuous-time signal (f_s : sample rate), larger *N* means more points analyzed and more sophisticated frequency resolution. We usually define the term $bw = \frac{2\pi k}{N} \times f_s$ (Hz) as ***analysis bandwidth***.

- Frame Interval *M*

The parameter *M* is the ***frame interval*** regarding rate of DFT analysis on time axis. Smaller *M* means more detailed time resolution for spectrogram. Conventionally, *M* is set to be equal or smaller than *N*, indicating each analysis frame $x_m[n]$ could have some overlapping samples to the neighboring analysis frames (e.g. $x_{m-?}[n]\ldots x_{m-1}[n]$, $x_{m+1}[n],\ldots x_{m+?}[n]$). Typically, ***window size*** is set to be 20ms~30ms, that is *N*=160 points~240points for an 8kHZ sample-rate speech signal, or *N*=320 points~480points for a 16kHz sample rate signal. ***Frame interval*** is usually set to be 5ms~20ms, that is M=40points~160points for an 8kHZ sample-rate speech signal, or *M*=80 points~320points for a 16kHz sample rate signal.

4. Implementations

A. Tools needed: Wavesurfer and gcc compiler

B. Spectrogram by ***DFT***

Steps:

i. Recording

Please record your ISOLATED five basic vowels, i.e. [ɑ], [i], [u], [ɛ], and [ɔ], one time for each sample rate of 16,000 Hz or 8,000 Hz. Hence, there are two WAVE files recorded. Sample encodings are all Lin16, i.e. linear quantized level represented by 16bits (per sample) and channels are all set to be Mono. Note that ISOLATED means you have to pronounce each vowel separately. You can find apparent silence (or short pause) between vowels. After recording, two isolated-vowel WAVE files are obtained. Please name the two files vowel-16k.wav and vowel-8k.wav.

ii. Generate the 8 sine waves (*.wav) that satisfy the following specifications:

1. Sample rates: 16,000 Hz or 8,000 Hz
2. The corresponding 8 continuous time signals are $x(t) = 10000\cos(2\pi ft)w(t)$; where *f* = 50, 200, 55, 220 Hz; $w(t) = \begin{cases} 1.0, & t = 0.0\text{-}1.0 \text{ sec} \\ 0.0, & \text{otherwise} \end{cases}$.

3. Please name the 8 wave files: cos_050Hz-16k.wav, cos_220Hz-16k.wav…cos_050Hz-8k.wav…cos_220Hz-8k.wav

iii. Save the spectrograms $X[m,k]$ for the WAVE files vowel-16k.wav, vowel-8k.wav, cos_050Hz-16k.wav, cos_220Hz-16k.wav…cos_050Hz-8k.wav…, and cos_220Hz-8k.wav with the following settings, to files in ascii:

Setting 1:

Analysis window size = 5ms

Analysis window type = rectangular

DFT/FFT window size = 8ms

Frame interval = 5ms

Setting 2:

Analysis window size = 5ms

Analysis window type = hamming

DFT/FFT window size = 8ms

Frame interval = 5ms

Setting 3:

Analysis window size = 20ms

Analysis window type = rectangular

DFT/FFT window size = 32ms

Frame interval = 10ms

Setting 4:

Analysis window size = 20ms

Analysis window type = hamming

DFT/FFT window size = 32ms

Frame interval = 10ms

Therefore, 40 ascii files are saved: vowel-16k.{Set1~Set4}.txt, vowel-8k.{Set1~Set4}.txt, cos_050Hz-16k.{Set1~Set4}.txt…cos_220Hz-16k.{Set1~Set4}.txt… cos_050Hz-8k.{Set1~Set4}.txt, …and cos_220Hz-8k.{Set1~Set4}.txt

Note that you should use DFT to compute Eq. 1.

iv. Show the spectrogram datum in vowel-16k.{Set1~Set4}.txt, vowel-8k.{Set1~Set4}.txt, cos_050Hz-16k.{Set1~Set4}.txt …and cos_220Hz-8k.{Set1~Set4}.txt by Matlab Image plot (imagesc). The results would look like the ones shown in the spectrogram by the wavesurfer. Please show the frequency in Hz on y-axis and frame index on x-axis.

v. Compare the results by Settings 1-4 and discuss the differences and their significances.

vi. Calculate how many multiplications and additions are executed for Settings 1-4.

C. Requirements

i. Please write a report to record every step you make in the above-mentioned implementation steps with word descriptions and figures excerpted from your screen.

ii. Upload your report with your C programs processing the implementations. Note that adding comments on your C codes is necessary.

References:

[1] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon (2001). *Spoken Language Processing: a guide to theory, algorithm, and system development*, page 274-281. Prentice Hall

[2] https://www.speech.kth.se/wavesurfer/