Middle East Technical University

Electrical and Electronics Engineering

EE430 Project

Preliminary Report

Spectrogram

by

Aykut DEMİREL

1813948

**1. INTRODUCTION**

As a preliminary work for the main project, a spectrogram function was written. Actually, MATLAB has its own spectrogram command but a new one with variable window type, window length and shift parameters was defined. This report includes the spectrogram script code, the differences observed on the screen when different type of windows are used with different window lengths and shift parameters. Comparison of the original spectrogram and the new one is also discussed.

**2. EFFECT OF WINDOW LENGTH**

When a signal is analyzed with different window lengths without changing the window type and amount of shift, the output changes significantly. Outputs for a sample sound file are provided below as an example. Window type is Hamming, amount of shift is 30 and window lengths are 500, 2000 and 5000, in this case.

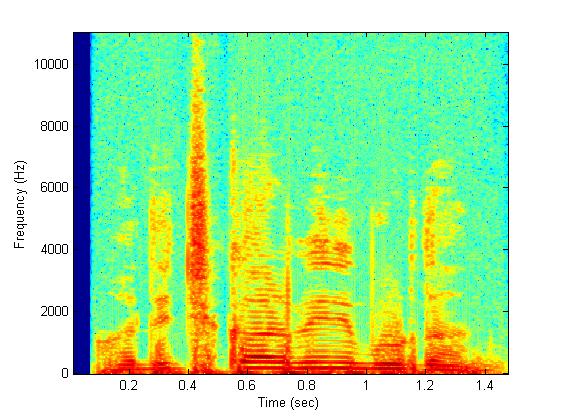


Figure 2.1. Window length = 500

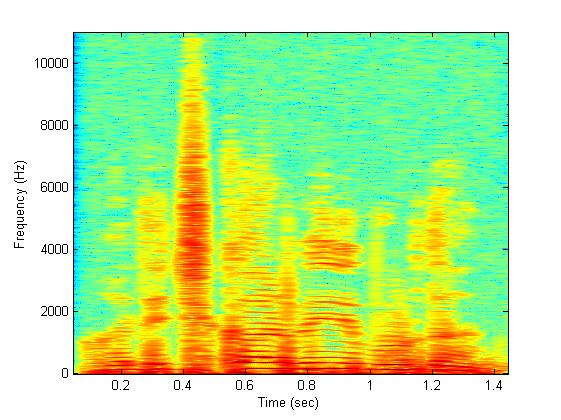


Figure 2.2. Window length = 2000

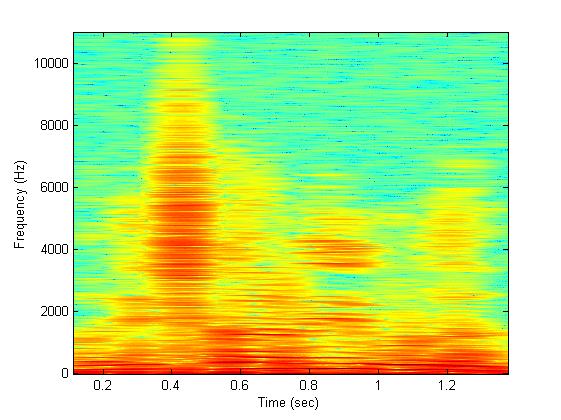


Figure 2.3. Window length = 5000

As seen from above figure, when the window length is increased, the resolution on the time axis decreases significantly. If we equate the window length to the length of the input signal, spectrogram gives the FFT of the whole input signal. When window length gets smaller, the input signal becomes more fractioned and accuracy of the instantaneous frequency analysis gets better.

**3. EFFECT OF THE AMOUNT OF SHIFT**

The amount of the shift is another important parameter that determines the performance of the spectrogram. Again, 3 different outputs for the same signal are given below. In this case, window length is 500 and shifts are 1, 30, 100.

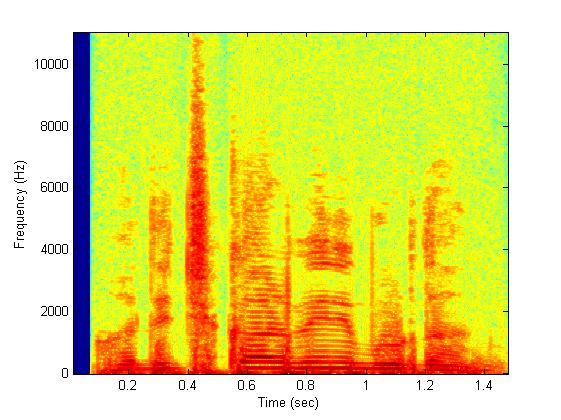


Figure 3.1. Shift = 1

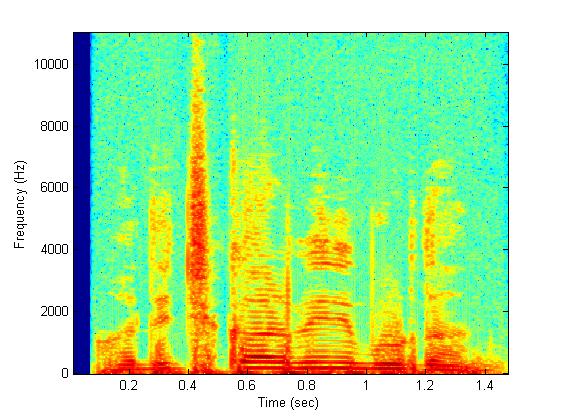


Figure 3.2. Shift = 30

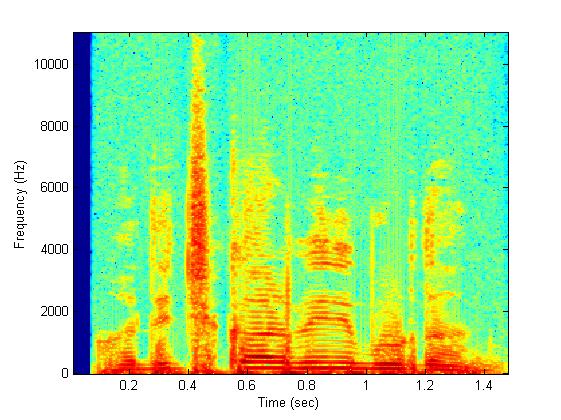


Figure 3.3. Shift = 100

When the amount of the shift is small, high frequency components at the ends of the windows whose FFTs are taken becomes more visible and when it is large, these high frequency components are suppressed by low frequency components. This is the reason why high frequencies in Figure 3.1 are shown by yellow while those in Figure 3.3 are shown by blue.

**4. EFFECT OF THE WINDOW TYPE**

Window type is another parameter that can be changed and it may be crucial to choose the correct window type for some special cases. For example, a Gaussian window is used when we want to reduce the effect of the high frequencies in the signal, i.e., when we want to smooth the signal.

**5. COMPARISON OF THE ORIGINAL SPECTROGRAM WITH THE NEW ONE**

The differences between the results when the original spectrogram command of MATLAB is used and when the new one is used are due to differences between their window lengths, shifts and window types. The original spectrogram command uses Hamming window and divides the input into 8 segments and take FFTs by 50% overlays, by default. When we use the same parameters with the original one, the results are the same. An example is shown below.

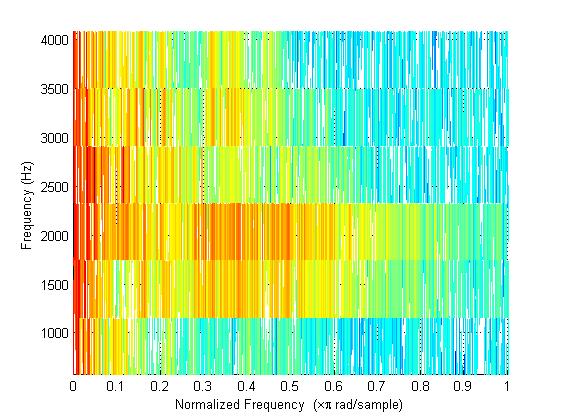


Figure 5.1. The MATLAB Spectrogram

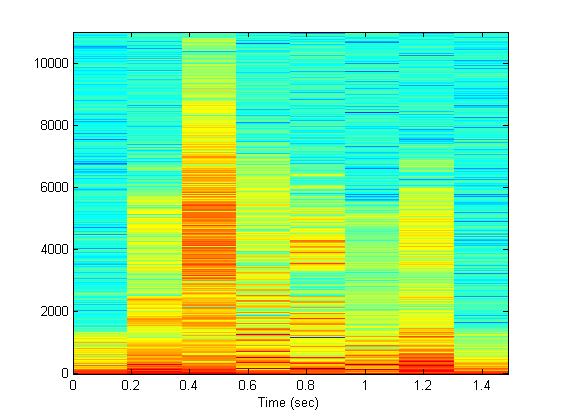


Figure 5.2. The new spectrogram

**6. APPENDIX - SCRIPT OF THE NEW SPECTROGRAM**

clear all;

close all;

%%

%determine the window type and lentgh and shift

wlength=500;

shift=100;

window = hamming(wlength); % can be gausswin,hamming,hann,tukeywin,blackman

%%

%load a sound file

[mywave,fs]=wavread('hos1.wav');

%%

%form the STFT matrix

row=ceil((1+wlength)/2);

col=1+fix((length(mywave)-wlength)/shift);

stft=zeros(row,col);

%%

%fill the STFT matrix by calculating fft for all windows

i=0;

k=1;

while(i+wlength<=length(mywave))

win = mywave(i+1:i+wlength).\*window;

W = fft(win, wlength);

stft(:,k) = W(1:row);

i=i+shift;

k=k+1;

end

%%

%form the time and frequency vectors

t = (wlength/2:shift:length(mywave)-wlength/2-1)/fs;

f = (0:row-1)\*fs/wlength;

%%

%figure the spectrogram

imagesc(t,f,20\*log10(abs(stft)));

axis xy

xlabel('Time (sec)');

ylabel('Frequency (Hz)');