BOĞAZİÇİ UNIVERSITY

CMPE 362: SIGNAL PROCESSING

Assignment 3

Huriye Özdemir - 2016400318

May 26, 2019

Contents

1 Advanced Peak Finder

1.1 Explanation

To improve our peak detection algorithm that we developed in the first homework I designed a low pass filter by changing the limit frequency of the low pass filter between 1000Hz, 2000Hz, 3000Hz and 4000Hz. I also applied these four different low pass filters with cut off frequencies (1k,2k,3k,4k) and plotted number of peaks versus changing cut off frequencies.

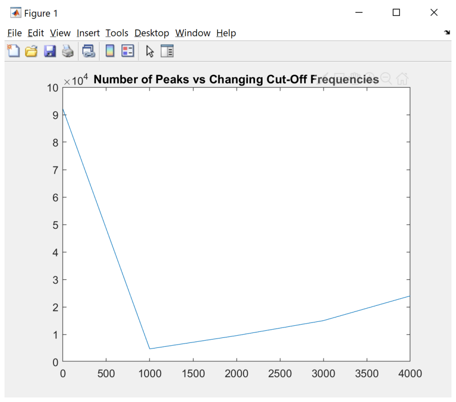

Figure 1: Number of peaks vs changing cut off frequencies

1.2 Code

You can find the code for question 1 below:
Details are explained in the comments.

```
clear; clc;

data = audioread('PinkPanther30.wav');
nofilter = findpeaks(data);

cutoff_Freq = [0, 1000, 2000, 3000, 4000];
cutoff_N = length(cutoff_Freq);
peaks_LowPass = zeros(1,cutoff_N);

% Add no filter result to first index
peaks_LowPass(1) = length(nofilter);

for i = 2:cutoff_N
    % Design low pass filter
    LP_filter = designfilt('lowpassiir', ...
                'FilterOrder',8, ...
                'PassbandFrequency', cutoff_Freq(i), ...
                'PassbandRipple',0.3, ...
                'SampleRate', 100e3);

    % Apply low pass filter
    result = filter(LP_filter,data);

    % Find peaks
    peaks = findpeaks(result);

    % Save number of peaks
    peaks_LowPass(i) = length(peaks);
end

% Plot result of part 1
figure;
plot(cutoff_Freq, peaks_LowPass); title('Number of Peaks vs
    Changing Cut-Off Frequencies');
```

2 Converting an Image Into Space Sound

2.1 Explanation

To convert an image into space sound, first, I checked the pixel values by dividing columns into 10 parts. It changes amplitude values in each index of pixel. After checking whether the pixel black or not, I created a sin wave with 10 amplitude for non-black values, zero amplitude for black values. I used index of pixel to use as a frequency in wave function. To convert each 1024 spectra that has 1 second duration into time domain, I used ”linspace” function that creates a time until 1024 incrementally. Then I added these waves in the ”wavs” matrix to combine with ”transpose” function later.

2.2 Code

```
img=imread("Hubble-Massive-Panorama.png");
I = rgb2gray(img);

% create binary image to set all values 0 and 1
binary_img = imbinarize(I);

% create zero array to combine generated wavs
wavs = zeros(1024,1024);

for i = 1 : 1024
    for j=1:900
        if binary_img(j,i)
            % Set amplitude ranks from 10 to 1
            amp = 10 -(j/90);
            % Fs number rank from 0 to 1
            t = linspace(0, 1, 1024);
            y= amp*sin(2*pi*j*t);
            % Add waves to wavs matrix
            wavs(i,:)=wavs(i,:) + y;
        else
            amp=0;
            t = linspace(0, 1, 1024);
            y= amp*sin(2*pi*j*t);
            wavs(i,:)=wavs(i,:) + y;
        end
    end
end

% combine waves and create a sound
C = transpose(wavs);
sound(C(:),1024)
```