Save Fluffy

Author: Han Song

University of Washington

Department of Applied Mathematics

Seattle, WA

Abstract: We examined 20 signal data set which belongs to Fluffly by using Fast Fourier Transform and Gaussian filter to extract marble path inside Fluffy. The marble's last location was found and Fluffy is saved.

GitHub Repo: SaveFluffy.git

I. Introduction and Overview:

Fluff accidentally swallowed a marble and felt discomfort. We took him to the doctor and got ultrasound performed on him. During the ultrasound, Fluffy could not stay still and resulted in noisy data. However, we had to act quickly before the marble became life threatening to Fluffy. Luckily, we can apply Fast Fourier Transform technique to the ultrasound data to filter and retrieve the last coordinate where the marble was.

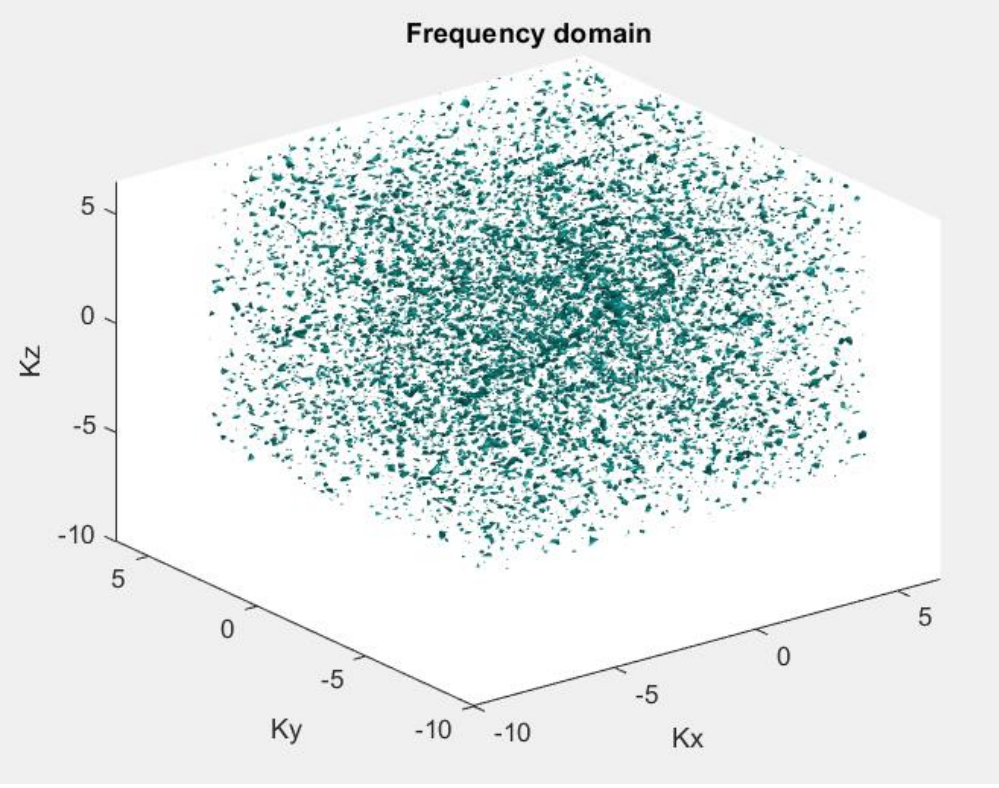

Figure 1: Fluffy's ultrasound data in frequency domain

II. Theoretical Background:

The noise does not happen in a consistent frequency, so it makes sense for us to translate data from time domain to frequency domain via Fast Fourier Transform (FFT). FFT computes the discrete Fourier transform with the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi kn}{N}}, k = 0,1, \dots, N-1$$

From the definition, there is N terms k and N term N, by choosing N as a power of 2, we can reduce processing time from $O(N^2)$ to $O(NlogN)$. We can also inverse FFT and go back to time domain. The purpose of going back and forth is to eliminate any undesired frequency. Taking average of data set in frequency domain, we will obtain the major frequency. Gaussian filter can be used to denoise completely.

III. Algorithm Implementation and Development:

First, I discretized our signal in time domain with L = 15 and n = 64 from -L to L, then we translate that into Fourier domain from $-2\pi \; to \; 2\pi$ and use fftshift to shift the main into the right order and created a grid for Fourier domain

```
L=15; % spatial domain
n=64; % Fourier modes
x2=linspace(-L,L,n+1); x=x2(1:n); y=x; z=x;
k=(2*pi/(2*L))*[0:(n/2-1) -n/2:-1]; ks=fftshift(k);
[Kx,Ky,Kz]=meshgrid(ks,ks,ks);
```

Next, I extracted the major frequency of 20 data signal by averaging all FFT and find the coordinate of the maximum value of average FFT and locate that peak in Fourier grid

```
Uave = zeros(n,n,n);
for j = 1:n_measurement
   Utn(:,:,:) = fftn(reshape(Undata(j,:,:),n,n,n));
   Uave = Uave + Utn;
end
Uave = fftshift(Uave)./20;
[value,index] = max(Uave(:));
[a,b,c] = ind2sub(size(Uave),index); %extract coordinates of the marble
isosurface(Kx,Ky,Kz,abs(Uave)./max(abs(Uave(:))),0.4)
xlabel('Kx')
ylabel('Ky')
zlabel('Kz')
title('Frequency domain')
xx = Kx(a,b,c);
yy = Ky(a,b,c);
zz = Kz(a,b,c);
```

Lastly, we use the coordinates of the peak frequency in a Gaussian filter to determine the path of the marble.

```
A = zeros(1,n_measurement);
B = A;
```

```
C = A;
filter=exp(-0.2*(((Kx-xx).^2)+((Ky-yy).^2)+((Kz-zz).^2)));
for i = 1:20
   data(:,:,:) = reshape(Undata(i,:),n,n,n);
   Un = fftn(data);
   Unft = filter.*fftshift(Un);
   Unf = ifftn(Unft);
   [value,index] = max(Unf(:));
   [b,a,c] = ind2sub(size(Unf),index);
   A(i) = a;
   B(i) = b;
   C(i) = c;
         End
```

IV. Computational Results:

The central frequency in Fourier Domain is (1.8,-1.0,0.0) and the last coordinates of the marble is (-5.6, 4.2, -6.0) . From the Gaussian filtered result, I was able to plot the trajectory of this marble and it appeared to be in a downward spiral shape.

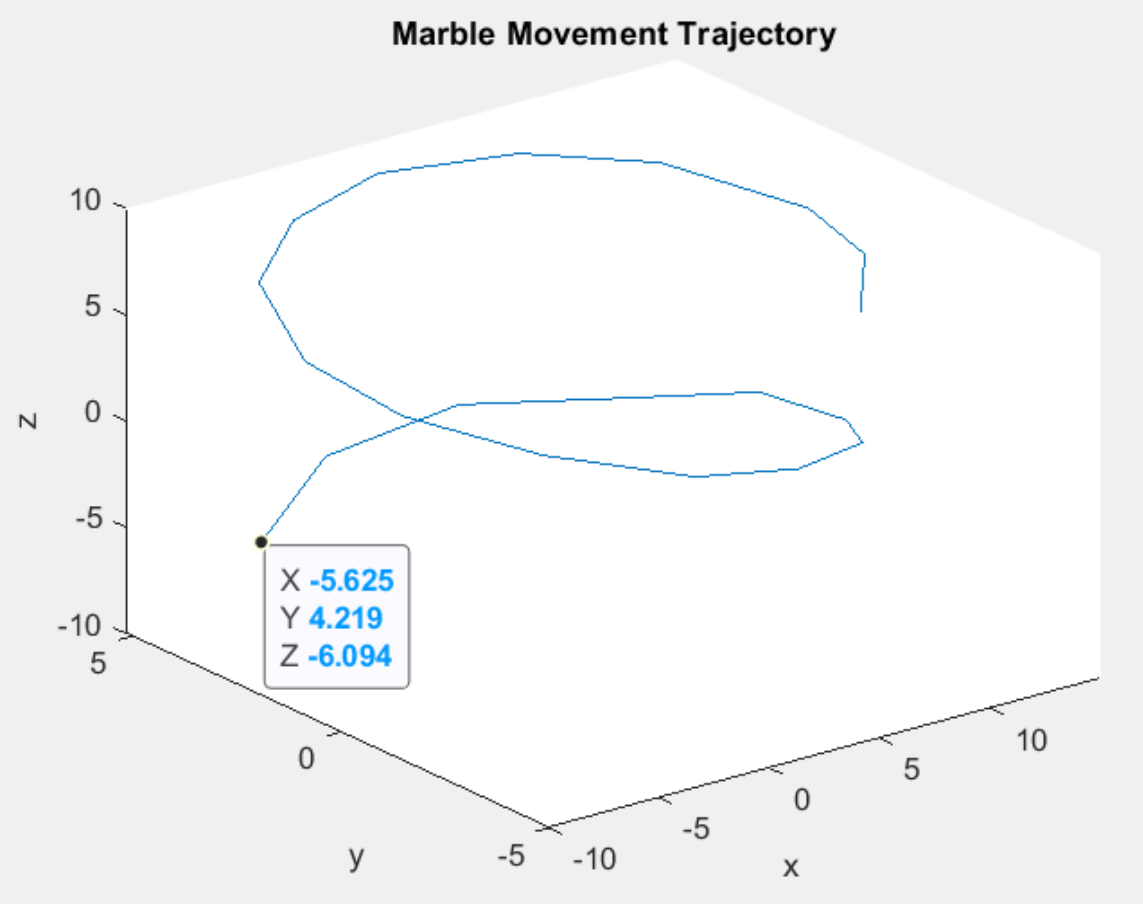

Figure 2: Marble trajectory in time domain

Appendix A: MATLAB functions

Size(matrix): to obtain matrix size. In this assignment, the video data is a 4D-uint8 type, so size(videos) will yield [row, column, channel, frame].

max(matrix): to obtain the largest value in the interested matrix

Linspace(x0,xend, step) : axis (xend-x0) long and step number of point between x0 and xend

Fft: Fourier transform

Ifft: inverse Fourier transform

Fftshift: fft will switch the left and right side, this function will rearrange them

Meshgrid: create a Cartesian join of two or multiple defined axis

Fftn: Fourier transform in n dimension

Appendix B: MATLAB Codes

```
%%%%%%%%%%%%%%%%%%%%%%% Save Fluffy %%%%%%%%%%%%%%%%%%%%%%%
% Name: Han Song
% Class: AMATH 582
% Due date: way past due

clear all; close all; clc;
load Testdata
[n_measurement, ~] = size(Undata);
L=15; % spatial domain
n=64; % Fourier modes
x2=linspace(-L,L,n+1); x=x2(1:n); y=x; z=x;
k=(2*pi/(2*L))*[0:(n/2-1) -n/2:-1]; ks=fftshift(k);
[X,Y,Z]=meshgrid(x,y,z);
[Kx,Ky,Kz]=meshgrid(ks,ks,ks);

Uave = zeros(n,n,n);
for j = 1:n_measurement
    Utn(:,:,:) = fftn(reshape(Undata(j,:,:),n,n,n));
    Uave = Uave + Utn;
end
Uave = fftshift(Uave)./20;
[value,index] = max(Uave(:));
[a,b,c] = ind2sub(size(Uave),index); %extract coordinates
of the marble
isosurface(Kx,Ky,Kz,abs(Uave)./max(abs(Uave(:))),0.4)
xlabel('Kx')
ylabel('Ky')
zlabel('Kz')
title('Frequency domain')
xx = Kx(a,b,c);
yy = Ky(a,b,c);
zz = Kz(a,b,c);
fprintf('Center frequency: %.3f %.3f %.3f\n',xx,yy,zz);

A = zeros(1,n_measurement);
B = A;
C = A;
filter=exp(-0.2*(((Kx-xx).^2)+((Ky-yy).^2)+((Kz-zz).^2)));
for i = 1:20
    data(:,:,:) = reshape(Undata(i,:),n,n,n);
    Un = fftn(data);
    Unft = filter.*fftshift(Un);
    Unf = ifftn(Unft);
    [value,index] = max(Unf(:));
```

```
    [b,a,c] = ind2sub(size(Unf),index);
    A(i) = a;
    B(i) = b;
    C(i) = c;
end
figure(2)
plot3(x(A),y(B),z(C))
xlabel('x')
ylabel('y')
zlabel('z')
title('Marble Movement Trajectory')
fprintf('Intense acoustic wave should be focused at %.3f
%.3f %.3f\n',x(A(end)),y(B(end)),z(C(end)));
```