Deniz Yüksel 28.12.2018

21600880

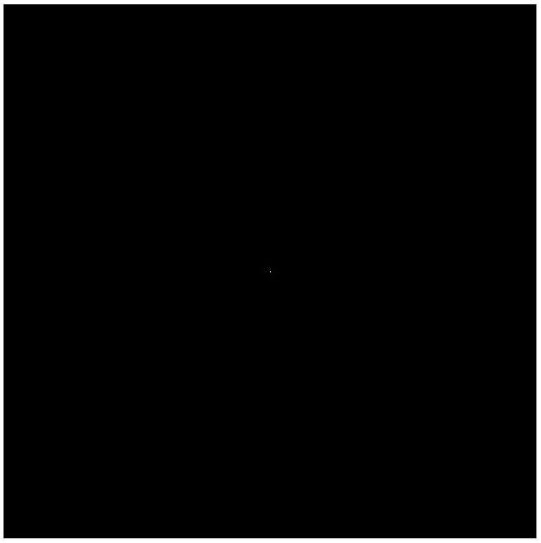
Section 1

EEE 391 MATLAB ASSIGNMENT 2

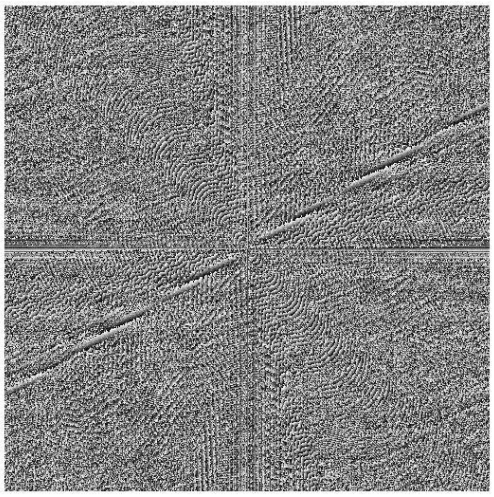
The initial image that is the output of the given Matlab code is given below:

In order to calculate the m,n pair, I drew a horizontal line that ranges from –pi to pi. Then, I figured out that Matlab generates an array of 512x512. In this case, for the horizontal axis, values range from 1 to 512. I divided the horizontal line to 8 sections each with increments of pi/4. Then I calculated that 192:320 is the range from –pi/4 to pi/4.

Magnitude of FT_Image: (there is a dot in the middle)



Phase of FT_Image:



My code to generate low pass and high pass filters:

```
dimension = 512;
lowerInterval = dimension * 3 / 8;
upperInterval = dimension - ( dimension * 3 / 8);
LP = zeros(dimension, dimension);

for m = lowerInterval:upperInterval
    for n = lowerInterval:upperInterval
        LP(m,n) = 1;
    end
end
m = 0;
n = 0;

Filtered_Image = InverseFourierTransform(LP.*FT_image);
%imshow(real(Filtered_Image),[])
%this effect blurs the image!

HP = ones(dimension, dimension);    % 1 in every cell.
HP = HP - LP;
% subtract LP to achieve 0 in middle cells ranging from (192,
320).

Filtered_Image = InverseFourierTransform(HP.*FT_image);
% imshow(real(Filtered_Image2),[])
% this effect shows the edges of fruits. Other parts of the
image are more
% like gone.
```

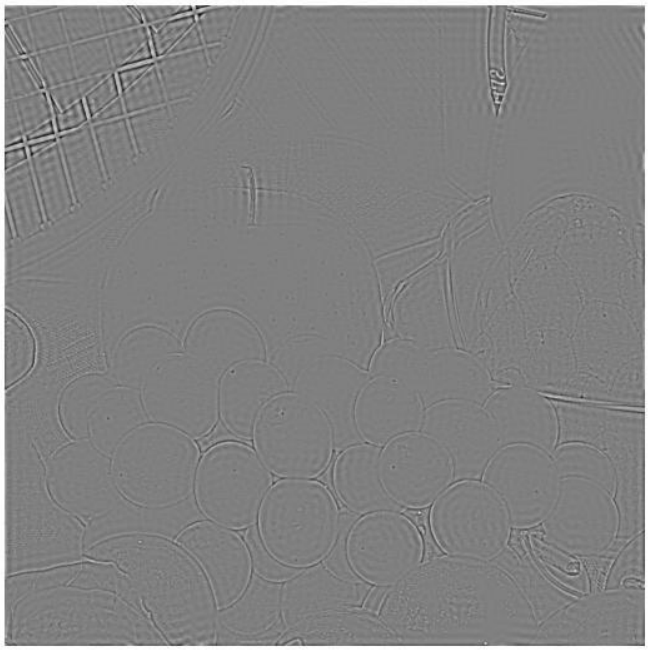
After applying low pass filter:

It is reasonable since low pass filters mute the high frequencies. In this image's case, the

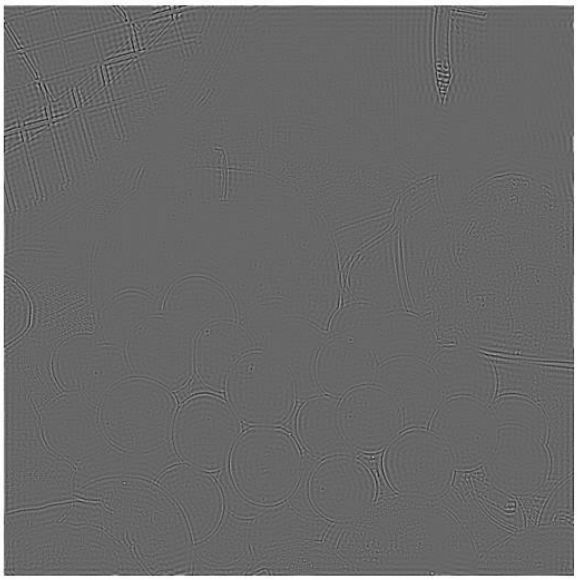
edges of the fruits are the high frequencies. So, the image is blurred!

After applying high pass filter:

When applied, this filter passes the high frequencies, in this case the edges of fruits and the nets of the tennis-racket like object. These sharp edges are more apparent in the picture whereas the low passes are gone.



When I change the bandwidth to an interval that is (160, 352), which was originally (192, 320), I get softer edges in high pass filter:



On the other hand, my new low pass filter shows the image in a less blurry way:

Code for my delta functions that would be used in the second question:

```
%size of the image is 512x512.
[x1,y1,z1] = size(image_rgb);

deltaOne = zeros(dimension, dimension);
for i = 1:x1
    for j = 1:y1
        if( i == j - 1)
            deltaOne(i,j) = -1;
        end
    end
end

deltaTwo = zeros(dimension, dimension);
for i = 1:x1
    for j = 1:y1
        if( i == j + 1)
            deltaOne(i,j) = 1;
        end
    end
end

deltaThree = zeros(dimension, dimension);
for i = 1:x1
    for j = 1:y1
        if( i - 1 == j)
            deltaOne(i,j) = -1;
        end
    end
end

hx = deltaOne + deltaTwo;
hy = deltaThree + deltaFour;
deltaFour = zeros(dimension, dimension);
for i = 1:x1
    for j = 1:y1
        if( i + 1 == j)
            deltaOne(i,j) = 1;
        end
    end
end

Filtered_Image = InverseFourierTransform(hx.*FT_image);
%imshow(real(Filtered_Image),[])
```

Since I could not realize and implement conv2d, I could not use my delta and h functions.. ☹