MCBU

CSE3113/CSE3214 Introduction to Digital Image Processing

Homework 2 Report

Ekin ŞUATAMAN – 170315022

13.6.2020

1. Tools

MATLAB R2018a Version 1.0.0.1

2. Problems

My image is second image and total I observe 4 problems for my ımage (2.tif).

2ND Image Problems:

- **Periodic noise problem**
- **Salt-pepper noise problem**
- **Low brightness**
- **Sharpening**

3. Solutions

Code-Part1

```
close all;
clc;
clear;

%Loading Original and noise Image
a=imread('images/original.tif');
subplot(1,2,1);
imshow(a);
title('Original Image');

b=imread('images/2.tif');
subplot(1,2,2);
imshow(b);
title('This is Image 2');

%applying 2 3 matricss and remove the salt pepper noise with median filter
b=medfilt2(b,[2 3]);
figure, imshow(b)
title('Median Filter');
```

First I read the ımages (original.tif and 2.tif) after that I apply median filter to 2.tif image which I called b. I apply median filter cause my image has a salt pepper noise and median filter remove this problem. I also use matrix by 2 3 because default matrix value was 3 3 and I want to apply this filter more bigger squares so I choose 2 3 matrix.

Code-Part2

```
22      %Paddedsize doubles the original image in two sizes
23 -    PQ=paddedsize(size(b));
24
25      %Notch filters for extra peaks in the Fourier Transform
26 -    H1=notch('btw',PQ(1),PQ(2),7,81,24);
27 -    H2=notch('btw',PQ(1),PQ(2),7,923,727);
28
29      %Calculate the Fourier Transform of the image
30 -    F=fft2(double(b),PQ(1),PQ(2));
31
32      %Apply array multiplication for each pixel
33 -    FS_b=F.*H1.*H2;
34
35      %Coverted result to spacial domain
36 -    F_b=real(ifft2(FS_b));
37
38      %For return to original size crop the image to undo padding
39 -    F_b = F_b(1:size(b,1),1:size(b,2));
40 -    figure,imshow(F_b,[])
```

I deal with Periodic Noise in this part by using notch filter. Firstly I use paddedsize function because this function works to enlarge the picture in both dimensions. We must enlarge because if we apply Fourier Transform to the original size of the picture, the picture may be distorted because we get something periodic. I used paddedsize to avoid this problem.

First, I converted it to the frequency domain and detected the brightness that corresponds to the noise, and applied the filter to destroy them. In notch function firstly I apply ideal filter but ringing effect occurred as a problem so I applied butterworth because it gradually jumps from white to black. Notch filter reduced the periodic noise in the picture very much as it made the black around the center white. The noise in the periodic noise is generally symmetrical so I used a 2 notch filter to destroy them.

- PQ(1) and PQ(2) represented symbolizes doubled dimensions on the X and Y axis.
- 7,8,24 values are I see brightness in these values (by using data cursor).

I did array multiplication when multiplying my image in the frequency domain with H1 and

H2(.* means multiple pixel by pixel).Then apply inverse Fourier Transform and convert it to

real number then I cropped back to original size.

Code-Part3

```
%Sharpining
sharp1=imsharpen(F_b);
figure, imshow(sharp1,[])
title('Sharpening Filter');

%where increasing brightness
g = sharp1;

% Display the Fourier Spectrum
% Move the origin of the transform to the center of the frequency rectangle.
Fc=F;
Fcf=fftshift(FS_b);

% use abs to compute the magnitude and use log to brighten display
S1=log(1+abs(Fc));
S2=log(1+abs(Fcf));
%figure, imshow(S1,[])
%figure, imshow(S2,[])
```

For sharpening problem I use imsharpen filter and also notch filter blurring my ımage too. I do Fftshift to bring the image to the center, I convert abs to get rid of complex numbers because if I don't get rid of complex numbers the value range will be very large. Also used log transformation for when I do linear mapping, for most places don’t turn black.

Code-Part4

```
61
62      %where increasing brightness with gama
63 -    im = g/255; %normalize
64
65 -    g1 = 0.4;
66 -    out1 = 255*(g.^g1);
67 -    figure,imshow(out1, []);
68 -    title('After increased brightness with gama');
69
70
```

For brightness I use gama and I divided 255 because to change the colour range from 0 to 1. Also I choose 0.4 because the best result came at this value and lastly I multiple every pixel with 255.

4. Conclusions and Observations

It was difficult to assess the outcomes and decide whether it was good enough or could have been better It was also difficult to decide which method to use with which variables.

5. References

I benefited from the lecture videos published on teams and lecture slides.