|  |  |
| --- | --- |
| ***Programming Language*** |  |
| ***Programming Environment*** | *MATLAB R2021a* |
| ***Reflections*** | *We did not have difficulty in the homework in general. There was reinforcement due to the examples made in the lesson. However, while we were finding the noisy points, we did not know how to do it first and we found it manually, then we found the original points with your help. The points we found ourselves were very close to the original points.* |

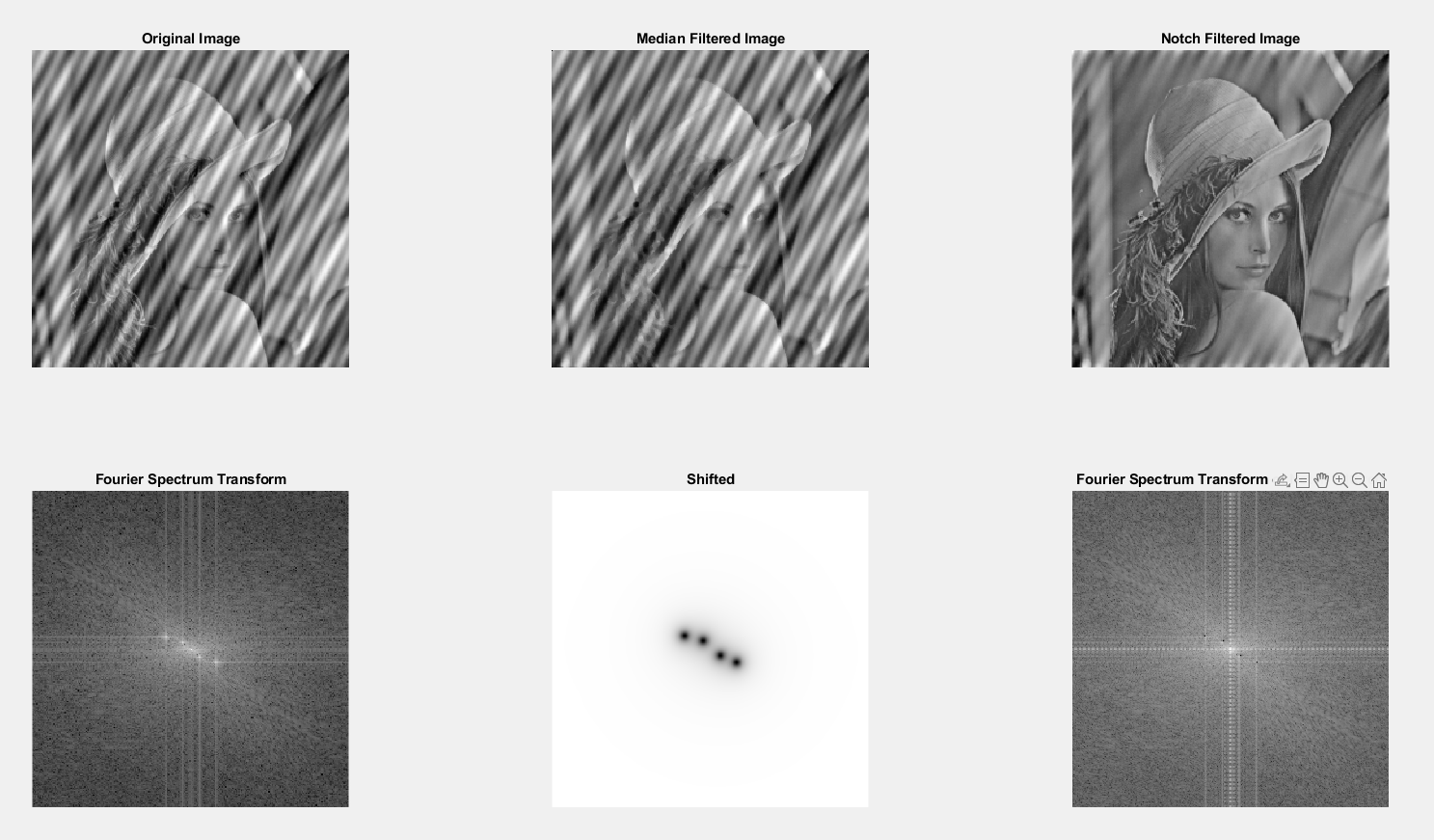
***Results & Discussion***

1. *Paste the output figures that you generated.*

*ekran görüntüsü, fotoğraf kağıdı, siyah beyaz içeren bir resim

Açıklama otomatik olarak oluşturuldu*

**Figure1:** *Output-1*

**

**Figure2:** *Output-2*

1. *Make a discussion about your results around the following questions.* 
   1. *Is the median filter effective at removing this type of noise? Why/why not?*

**In our opinion, the median filter alone is not sufficient to remove the noisy type. The median filter tries to focus by clearing the pixel jumps in the picture, but it is not enough for our picture, it did not give the desired result.**

* 1. *What is the effect of changing the kernel size of the median filter?*

**As the kernel size increases in the median filter, the noisy noises on the picture disappear, but the picture also loses its clarity.**

* 1. *Can you remove this type of noise effectively using some other spatial domain filters?*

**Spatial domain filters are a type of image filtering technique that operate directly on the pixels of an image in the spatial domain.** **They are commonly used for noise reduction, image enhancement, and other image processing tasks.** **While the median filter is commonly used for reducing impulse noise, there are other spatial domain filters that can effectively remove different types of noise as well. As alternative filter variants, the Gaussian Filter works by combining the image with a Gaussian kernel that highlights nearby pixels and reduces high-frequency noise. The mean filter, on the other hand, is effective at reducing random noise, but can blur edges and fine details in an image.**

***Source Code***

***A-Code1:***

clc; clearvars; close all;

img1 = imread('image20.tif');

subplot(2, 3, 1);

imshow(img1);

title("Original Image");

img2 = medfilt2(img1);

subplot(2, 3, 2);

imshow(img2);

title("Median Filtered Image");

PQ = paddedsize(size(img1));

H1 = notch('btw', PQ(1), PQ(2), 5, 26, 11);

H2 = notch('btw', PQ(1), PQ(2), 5, 82, 9);

H3 = notch('btw', PQ(1), PQ(2), 6, 432, 504);

H4 = notch('btw', PQ(1), PQ(2), 6, 488, 502);

%H1 = notch('btw', PQ(1), PQ(2), 5, 27, 8);

%H2 = notch('btw', PQ(1), PQ(2), 5, 83, 8);

%H3 = notch('btw', PQ(1), PQ(2), 6, 429, 506);

%H4 = notch('btw', PQ(1), PQ(2), 6, 485, 496);

%H1 = notch('btw', PQ(1), PQ(2), 5, -81, -7);

%H2 = notch('btw', PQ(1), PQ(2), 5, -26, -8);

%H3 = notch('btw', PQ(1), PQ(2), 6, 28, 11);

%H4 = notch('btw', PQ(1), PQ(2), 6, 82, 12);

subplot(2, 3, 5);

H\_combined = H1 .\* H2 .\* H3 .\* H4;

imshow(fftshift(H\_combined));

title("Shifted");

F = fft2(double(img1), PQ(1), PQ(2));

image20 = F .\* H\_combined;

img20 = real(ifft2(image20));

img20 = img20(1:size(img1, 1), 1:size(img1, 2));

subplot(2, 3, 3);

imshow(img20, []);

title("Notch Filtered Image");

img3 = img1;

Ff = fft2(img3);

Fsh = fftshift(Ff);

S = log(1 + abs(Fsh));

subplot(2, 3, 4);

imshow(S, []);

title("Fourier Spectrum Transform");

Fcf = fftshift(image20);

Ss = log(1 + abs(Fcf));

subplot(2, 3, 6);

imshow(Ss, []);

title("Fourier Spectrum Transform of Notch Filtered");

***B-Code2:***

clc; clearvars; close all;

img1 = imread('image21.tif');

subplot(2, 3, 1);

imshow(img1);

title("Original Image");

img2 = medfilt2(img1);

subplot(2, 3, 2);

imshow(img2);

title("Median Filtered Image");

PQ = paddedsize(size(img1));

H1 = notch('btw', PQ(1), PQ(2), 7,17,11);

H2 = notch('btw', PQ(1), PQ(2), 7, 43,22);

H3 = notch('btw', PQ(1), PQ(2), 7, 471,491);

H4 = notch('btw', PQ(1), PQ(2), 7, 501,499);

subplot(2, 3, 5);

H\_combined = H1 .\* H2 .\* H3 .\* H4;

imshow(fftshift(H\_combined));

title("Shifted");

F = fft2(double(img1), PQ(1), PQ(2));

image20 = F .\* H\_combined;

img20 = real(ifft2(image20));

img20 = img20(1:size(img1, 1), 1:size(img1, 2));

subplot(2, 3, 3);

imshow(img20, []);

title("Notch Filtered Image");

img3 = img1;

Ff = fft2(img3);

Fsh = fftshift(Ff);

S = log(1 + abs(Fsh));

subplot(2, 3, 4);

imshow(S, []);

title("Fourier Spectrum Transform");

Fcf = fftshift(image20);

Ss = log(1 + abs(Fcf));

subplot(2, 3, 6);

imshow(Ss, []);

title("Fourier Spectrum Transform of Notch Filtered");