



# Periodic Noise Reduction by Frequency Domain Filtering

# Periodic Noise Reduction by Frequency Domain Filtering

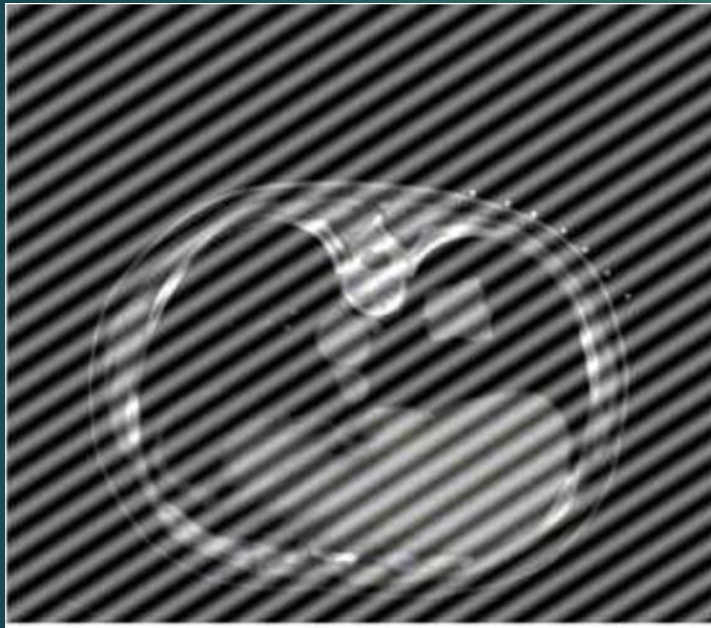
**Periodic Noise:** is the noise which arises from electrical or electromechanical interference during image acquisition.

An image affected by periodic noise will look like a repeating pattern has been added on top of the original image.

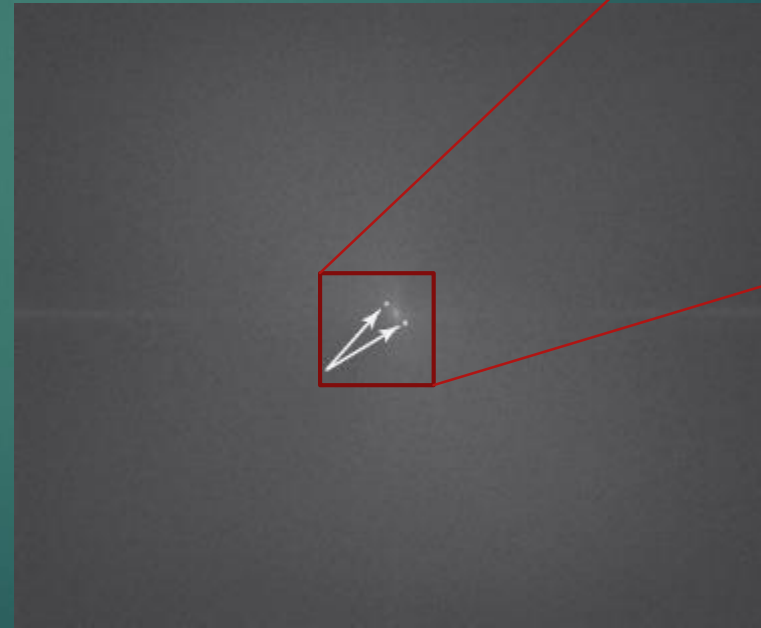


# Periodic Noise Reduction by Frequency Domain Filtering

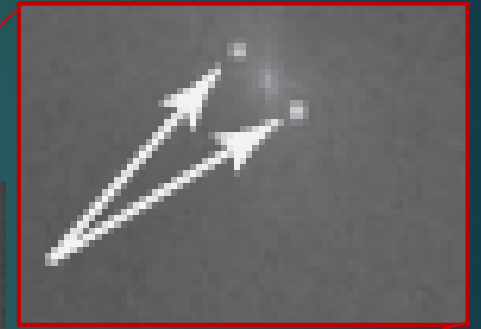
In the frequency domain periodic noise can be seen as discrete spikes.



Spatial domain

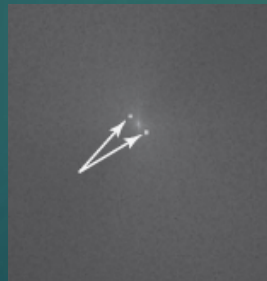
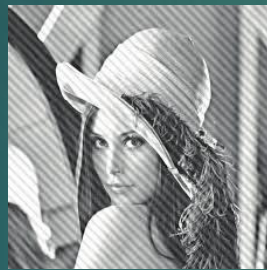


Frequency domain

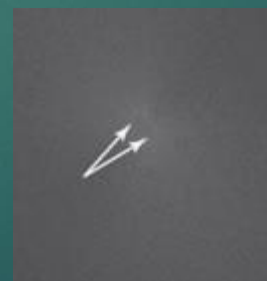
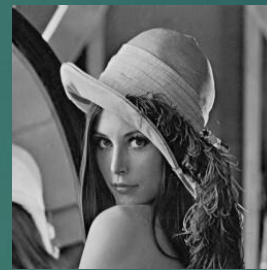


# Periodic Noise Reduction by Frequency Domain Filtering

Eliminating or reducing these impulses in the frequency domain will eliminate or reduce the sinusoidal noise in the spatial domain.



Frequency domain filter



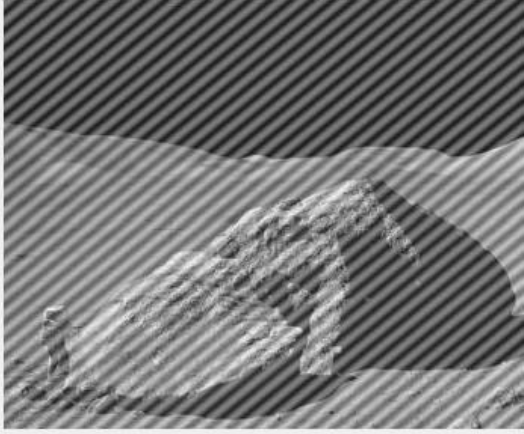
# Project idea

The idea of this project is to convert the image to the frequency domain using Fourier transform, and then allow the user to determine the places where he/she finds frequency spikes and then the program covers these points with black spots, and thus when converting the image to the spatial domain, the image will be almost free from periodic noise.

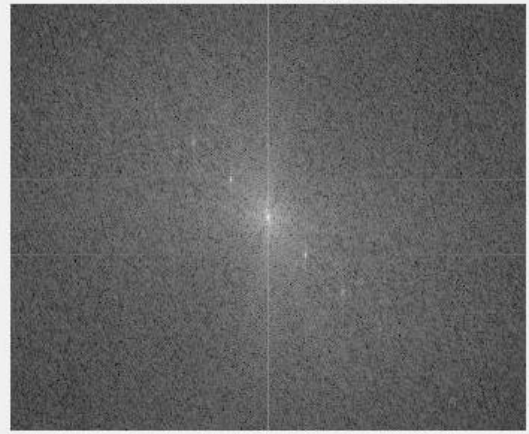


# Filtering Example

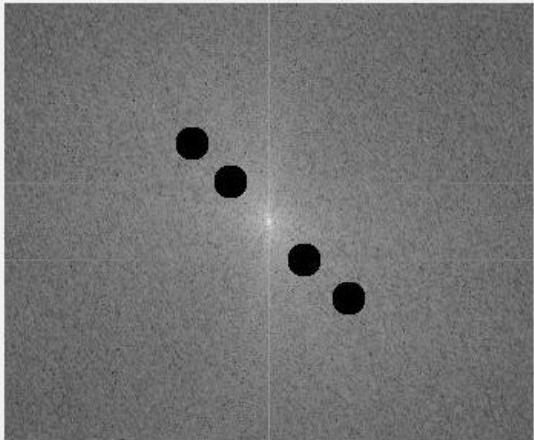
Original Image



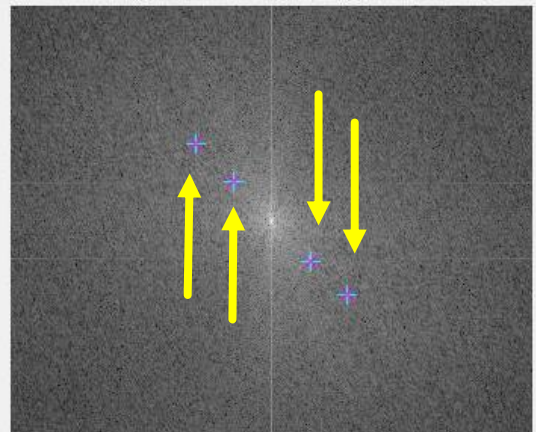
Original Fourier Spectrum



Filtered Fourier Spectrum



Original Fourier Spectrum



Filtered Image

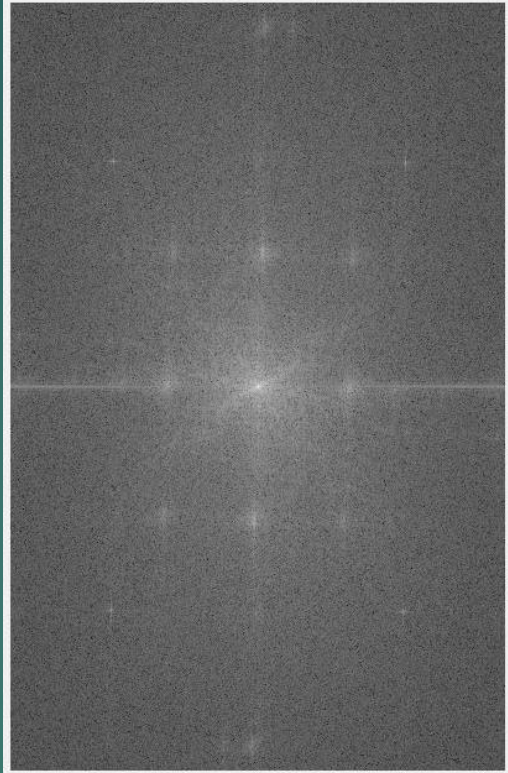


# Filtering Example

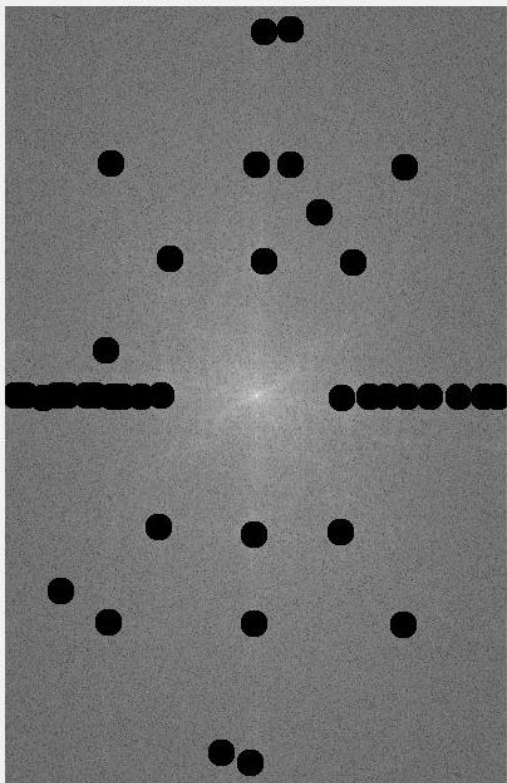
Original Image



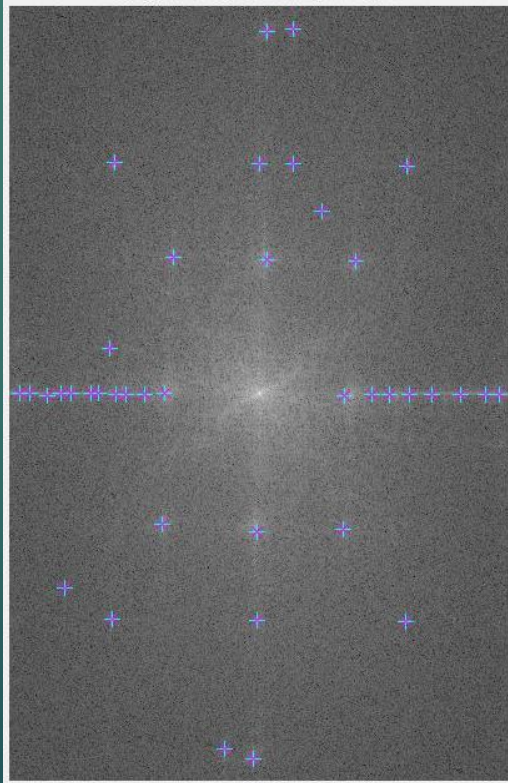
Original Fourier Spectrum



Filtered Fourier Spectrum



Original Fourier Spectrum





# Filtering Example

**Filtered Image**

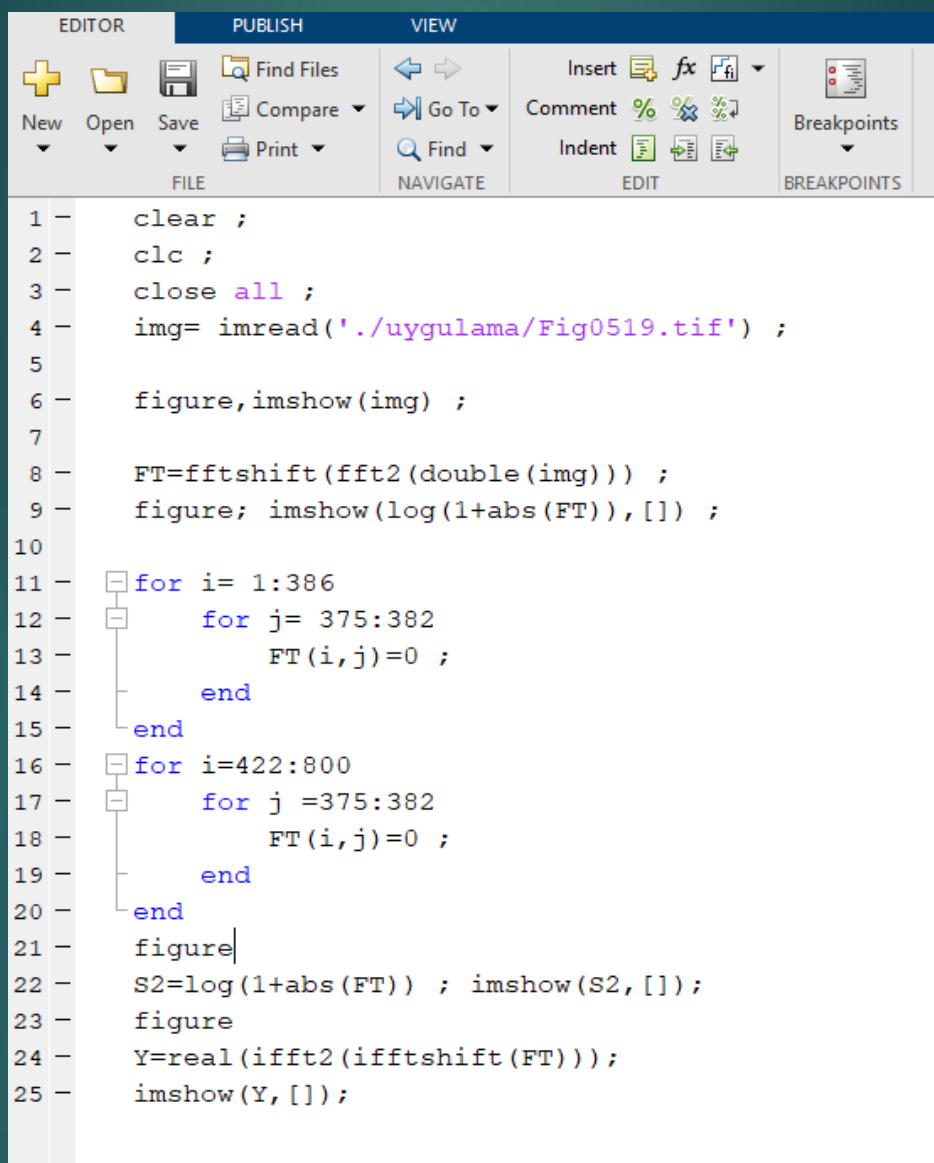




# MATLAB Code

```
EDITOR PUBLISH VIEW
1 - clear ;
2 - clc ;
3 - close all ;
4 - img= imread('./uygulama/Fig0519.tif') ;
5 - %img=rgb2gray(img) ;
6 - figure,imshow(img) ;
7
8 - FT=fftshift(fft2(img)) ;
9 - figure; imshow(log(1+abs(FT)),[]) ;
10 - [x0,y0] = getpts();
11
12 - [r ,c]=size(FT);
13
14 - for i=1:r
15 -     for j=1:c
16 -         for n=1:length(x0)
17 -             x=x0(n);
18 -             y=y0(n);
19 -             if (((x-j)^2)+((y-i)^2)<120) % 60 is the radius , default is 121;
20 -                 FT(i,j)=0;
21 -             end
22 -         end
23 -     end
24 - end
25
26 - S2=log(1+abs(FT)) ; imshow(S2,[]);
27 - figure
28 - Y=real(ifft2(ifftshift(FT)));
29 - imshow(Y,[]);
```

# MATLAB Code



The image shows the MATLAB Editor window with a script for image processing. The script reads an image, computes its 2D FFT, applies a mask to zero out the central region, and then displays the magnitude of the original and masked FFTs. The interface includes a menu bar with EDITOR, PUBLISH, and VIEW tabs, and a toolbar with icons for file operations, navigation, editing, and breakpoints. The script is as follows:

```
1 - clear ;
2 - clc ;
3 - close all ;
4 - img= imread('./uygulama/Fig0519.tif') ;
5
6 - figure,imshow(img) ;
7
8 - FT=fftshift(fft2(double(img))) ;
9 - figure; imshow(log(1+abs(FT)),[]) ;
10
11 - for i= 1:386
12 -     for j= 375:382
13 -         FT(i,j)=0 ;
14 -     end
15 - end
16 - for i=422:800
17 -     for j =375:382
18 -         FT(i,j)=0 ;
19 -     end
20 - end
21 - figure
22 - S2=log(1+abs(FT)) ; imshow(S2,[]);
23 - figure
24 - Y=real(ifft2(ifftshift(FT)));
25 - imshow(Y,[]);
```