SYDE 575

Group #22

Lab 3 Report

Prof David A. Clausi

Abdulhameed Naji - *20642623*

Shilpan Shah - *20603389*

Contents

List of Figures

1 Introduction

This lab investigate process for image restoration in frequency domain. This is done by analysing the images using Fourier Transformation to examine or modify their spatial frequency components in the form of sinusoidal gratings. The Fourier transformation generates the amplitude and phase components of the 2D signal, which is an extension of the behavior of one dimensional signals. Afterwards, several noise reduction techniques were used, including various filters such as ideal low pass filter with changing cut-off radius and Gaussian filter. In the later stage of the lab, a Notch filter was designed and implemented to remove grated pattern artifacts in an image by attenuating their associated frequency components; this is in effect completely remedied the image from the artifact - a result which is impossible to achieve by common spatial filtering methods.

2 Fourier Analysis

A test image consisting of white rectangle on a black background was created. The forward Fourier transform is applied to obtain the Fourier spectra of the image. The transformed image is then shifted such that the origin starts at the center of the image – the shift is merely conventional and helps the human perception of the transform. Figure X and Y shows the test image and its Fourier spectra.

The image is then rotated and transformed in order to study the change in the Fourier spectra – the geometric transformation helps understand the required frequency composition to attain our perceived spatial interpretation.

Finally, the Lena image is transformed to frequency domain in an effort to display the magnitude and phase representation in frequency domain. The magnitude and phase components were then separately inverted to identify their respective contribution to the spatial domain image.

2.1 Original Rectangle

2.1.1 General Energy Distribution

The source image and its corresponding transformation are shown in Figure 1 below. Since the Fourier spectra is shifted to the center, the lowest frequency component is in the center. This is the DC gain of the image, and it happens to be the predominant frequency. The alignment of the spectra is indicative to the transitions occurring in spatial domain. We see changes in intensities along the horizontal direction; which in turn corresponds to higher frequencies being prevalent in the horizontal direction of the transform. For a given column in the image, there are no changes in intensities along the vertical direction. As a result, the Fourier spectra shows little to no frequency components along the height of the image.

2.1.2 Fourier Spectra Inferences

The Fourier transform shows the frequency composition of the source image – in the spatial domain, it is the combination of scaled sinusoids in the x and y directions. For instance, an image composed purely by a single sinusoid in the y-direction would appear as horizontally grated patterns whose width is a function of the sinusoid's frequency.

Unlike a grated sinusoid pattern whose frequency response would appear as an impulse, the rectangle forms the equivalent of a 2D step function. This can be visualized if we take a cross section along the width of the image - the cross section would depict a 1-D step function. As a result, the spectra shows a familiar 2D pattern of the *sinc* function, which is the known frequency response of the step function. The frequency at the center, which happens to be the DC-gain, is strongest, with a decaying spread on both sides. Statistically speaking, the DC gain can be correlated to the mean of the image's intensities. As a result, we generally expect the centers in most of our Fourier transformed images to be relatively bright. Very

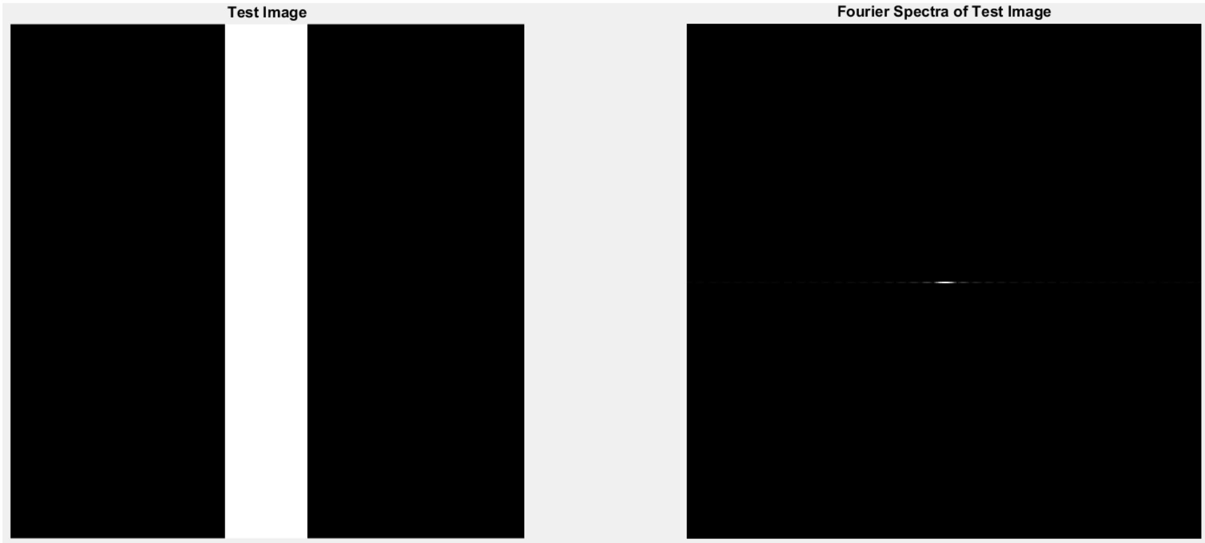

Figure 1: Rectangle image and its associated Fourier spectra

high frequencies are often weaker due to the inherit nature of human perception and imaging – we produce images that do not have frequency values that exceed our perception of a coherent image.

2.2 Rotated Rectangle

2.2.1 Change in Fourier Spectra

The plot of the rotated image and its associated spectra are shown in Figure 2 Visually speaking, the Fourier transform *appears* rotation invariant. The resulting spectra is a simply a rotated version of the original rectangle image. However, in terms of frequency composition, the transform is different - we now see frequency components along a 45° angle which implies there are changes along both the width and height of the image in spatial domain. This is expected from our interpretation of the spatial image - the slanted edge is formed by equal changes in the x and y directions respectively.

2.2.2 Observations & Conclusions

Based on rotations, we can conclude that the Fourier spectra retains it's overall shape. The spectra is rotated by the same amount of its corresponding image. Consequently, the magnitude in the frequency domain changes in such a way to reflect the intensity transitions due to the slanted edge. In effect, this verifies the ability of the transform to represent any image by adjusting the magnitude and phase of sinusoid components in frequency domain.

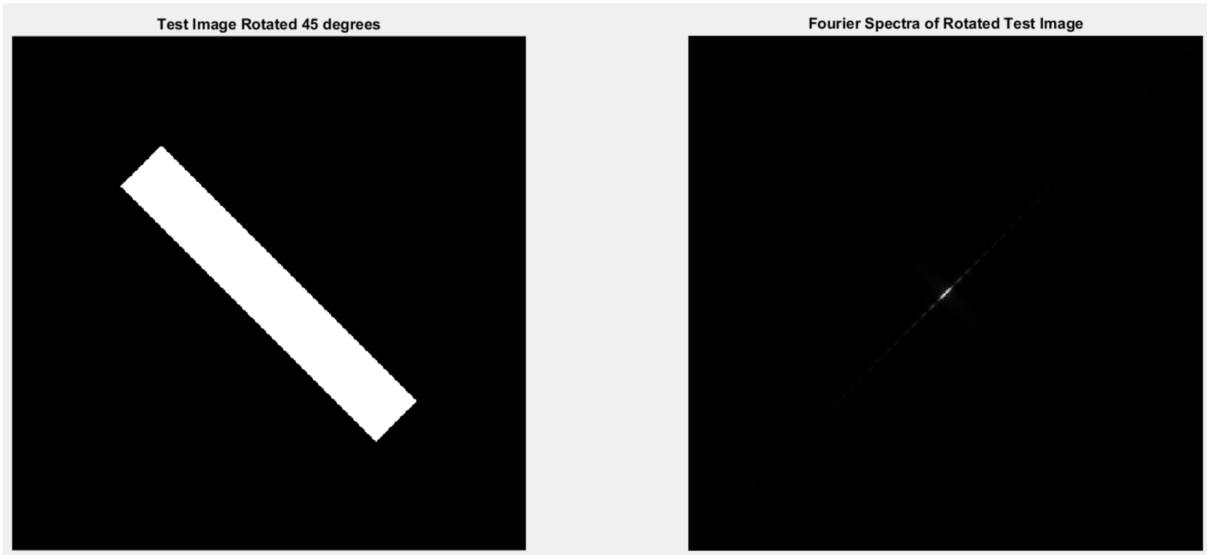

Figure 2: Rotated rectangle image and its associated Fourier spectra

2.3 Magnitude & Phase Decomposition

The Fourier transform was performed on the Lena image; the amplitude and phase are extracted and plotted as shown in Figure 3. From there, the inverse transform is taken for both the amplitude and phase separately. The resulting image reconstruction is shown in Figure 4.

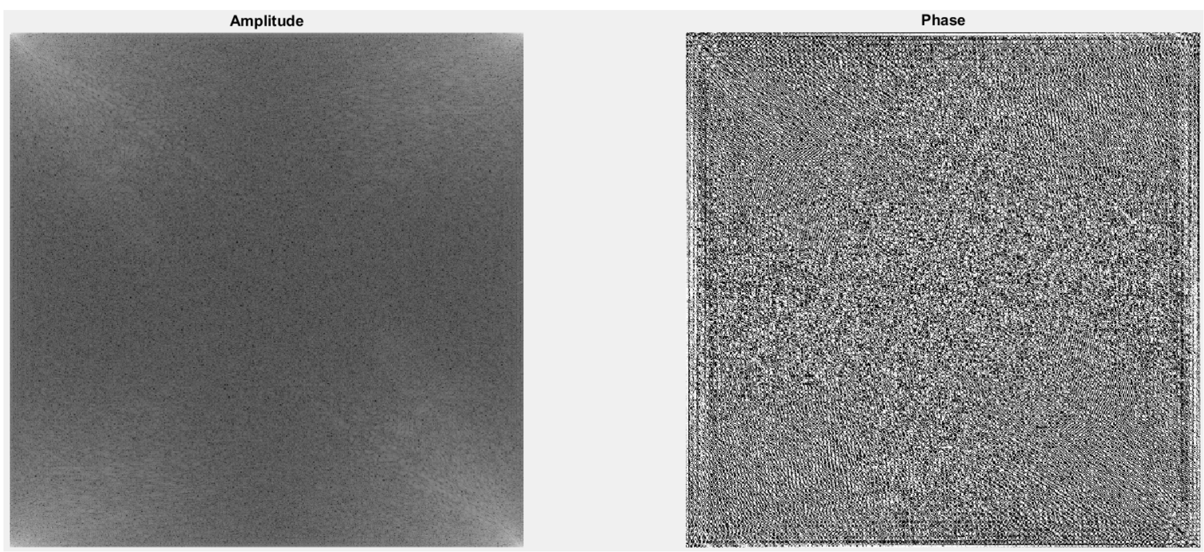

Figure 3: Lena image decomposed to its phase and amplitude components

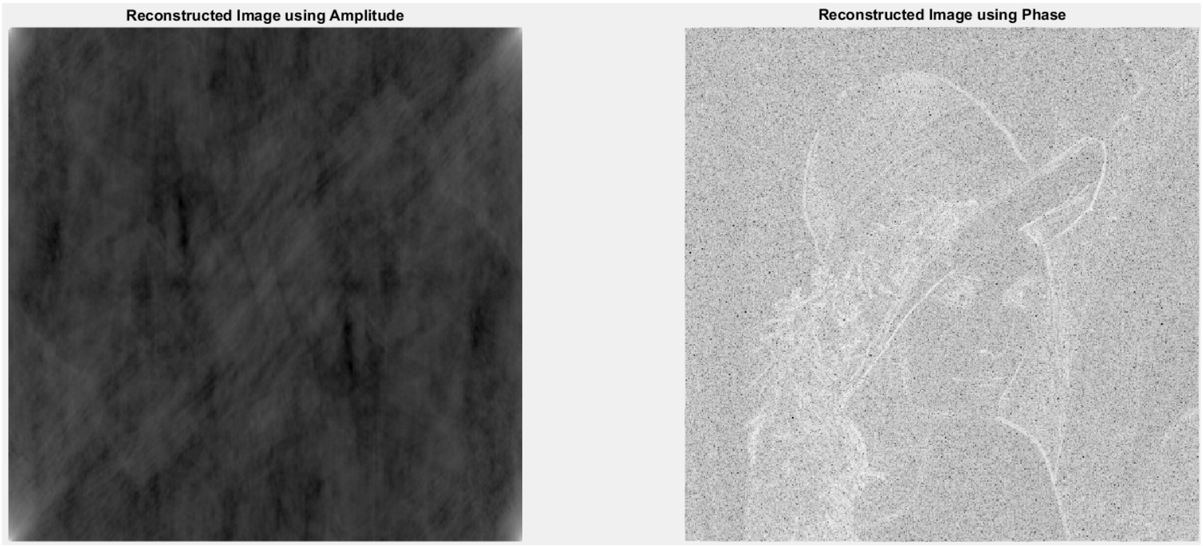

Figure 4: Lena image reconstructed using the amplitude and phase separately

2.3.1 Amplitude Reconstruction

As shown in Figure 4 the reconstructed image using the amplitude does not resemble the original image. Overall, the image looks like a generic smoke cloud. The amplitude dictates the strength of the sinusoid components - as a result, we expect the information in the magnitude plot to be sufficient to govern the shape of image details, but not their actual position, alignment, or pattern.

2.3.2 Phase Reconstruction

The phase component captures the overall outline or features in the image; albeit in a very noisy manner. For a given frequency, the phase plot dictates how are the sinusoids offset - which in turn controls the interference or superposition of the sinusoids with each other. Components of similar phase will have higher outputs than components out of sync. Intuitively, this provides us with the sense of directional information. Consequently, the phase of the transformed image is capable of capturing the location, alignment, and pattern of shapes and features in the image.

3 Noise Reduction In Frequency Domain

In this experiment, the Lena image is contaminated with Gaussian noise with variance of 0.005. The source and noisy image are then compared in frequency domain. From there, a filter is designed in frequency domain - those include two ideal low pass filters and a Gaussian filter. In frequency domain, the filters simply mask off regions of the frequency plot - depending on the intensity of the mask, the filter excludes or includes frequencies in the masked region. The resulting noisy image is shown in Figure 5.

Figure 5: Source image (left) versus the noise contaminated image (right)

For the upcoming experiments, the PSNR of each method is recorded. The results are depicted in the chart shown in Figure 6

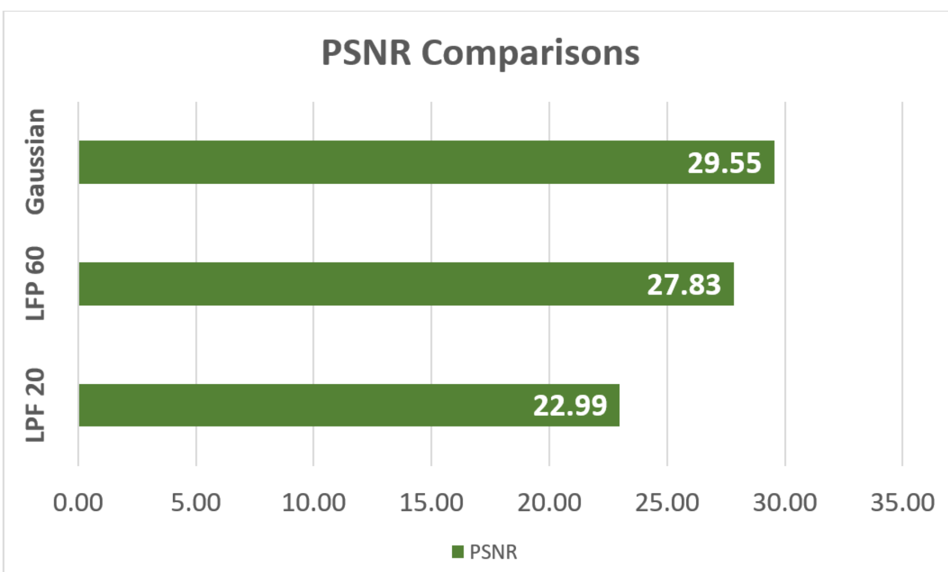

Figure 6: PSNR of each three frequency filtering methods

3.1 Noise Fourier Spectra

Figure 7 shows the Fourier spectra of the source and noisy image. The immediately notable difference is the loss of low frequency components. The original image had brighter streaks and regions localized near the origin – these magnitudes represent the presence of low frequency components in the image. Low frequency components generally make up most general features of the image whereas high frequencies are reserved to forming edges and fine details. After applying the transformation, we notice the streaks and central region have their magnitudes attenuated.

On the other hand, the overall magnitude of the spectra has been biased towards higher values. This is due to the small variance of the selected Gaussian noise - the grains are smaller and stronger, which in turn introduces a positive bias towards high frequency components.

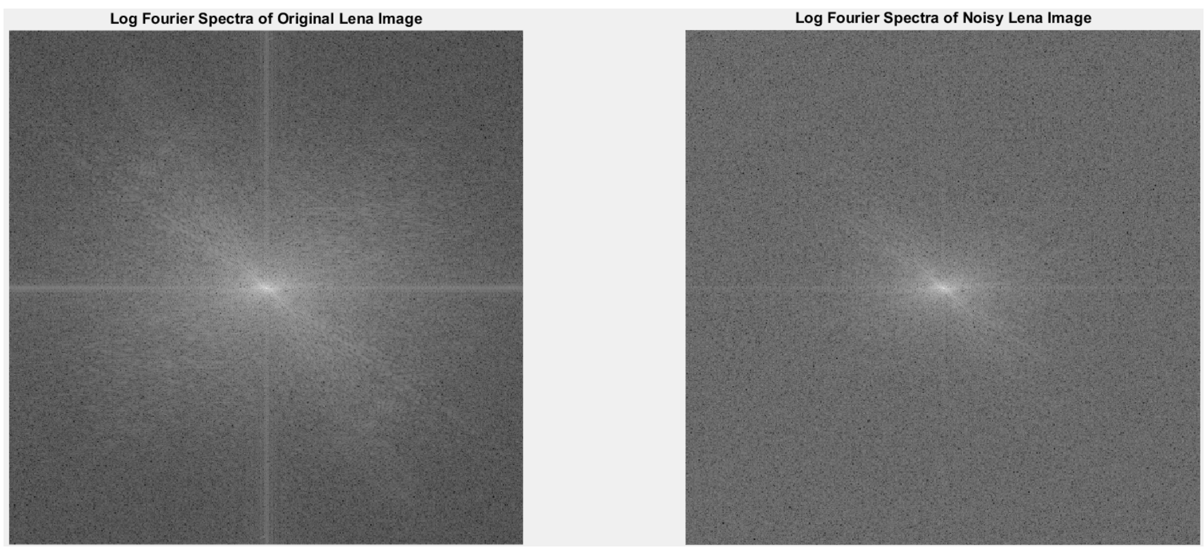

Figure 7: Fourier spectra of the source image (left) versus the noisy image (right)

3.2 Ideal Filter - High Cut-Off

An ideal filter is created strictly in the frequency domain by forming a circular, white mask. The mask overlaps frequencies of interest and multiplies their values. Since the mask is white with a value of 1, frequencies that lie inside the circle are preserved whereas frequencies outside are completely attenuated.

The filter of radius 60 is shown in Figure 8. The filter effectively masks a relatively large portion of the frequencies. The higher frequencies outside the circle are attenuated. As a result, this corresponds to a low pass filter that blurs the image by rejecting some of the high frequency components that form outer edges and fine details.

Aside from the spotty noise that results from residual Gaussian noise, the image suffers from what is known as the Gibbs effect, or ringing artifact. The effect is subtle and is noticeable around the outline of the hat and overall edge features. The artifact is a result of naive frequency filtering by which the spatial domain response exhibits oscillations around edges. This can be supported by the duality property of the Fourier transform - an ideal box filter in frequency domain has a time domain representation of a sinc function. Conversely, had the frequency domain filter been a sinc function - the spatial domain filtering result would be the expected ideal box filter blur.

The PSNR of this filter is **27.83**.

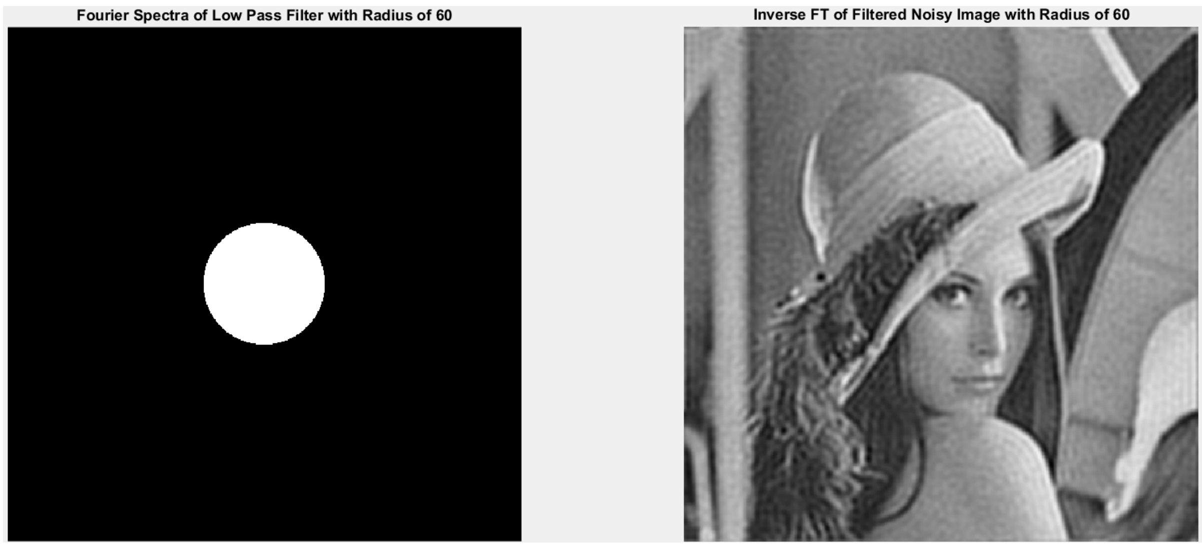

Figure 8: Ideal filter of radius 60 and the associated filtered image

3.3 Ideal Filter - Low Cut-Off

Similar to the former example, an ideal filter is recreated, but with a smaller radius of 20. The smaller radius imply that an even smaller set of frequencies will be included in the end result. More frequencies are being attenuated – some of which are relatively low frequencies that form the basis of our image. As a result, we expect the intensity of the blur to increase significantly. Figure 9 shows the result of the frequency filtering – the image has been blurred to unrecognizable extents.

As stated previously, the radius of the circle controls the cut-off frequency; hence, a larger circle encompasses more frequencies, which leads to less blurring. A smaller circle cuts off even lower frequencies, which in turn may help alleviate high frequency noise patterns, but also destroys the image integrity by removing frequency components that build up the spatial features of our image.

The PSNR of the low cut-off filter is **22.98** and is understandably lower than the former filter due to its excessive filtering of needed high frequency components.

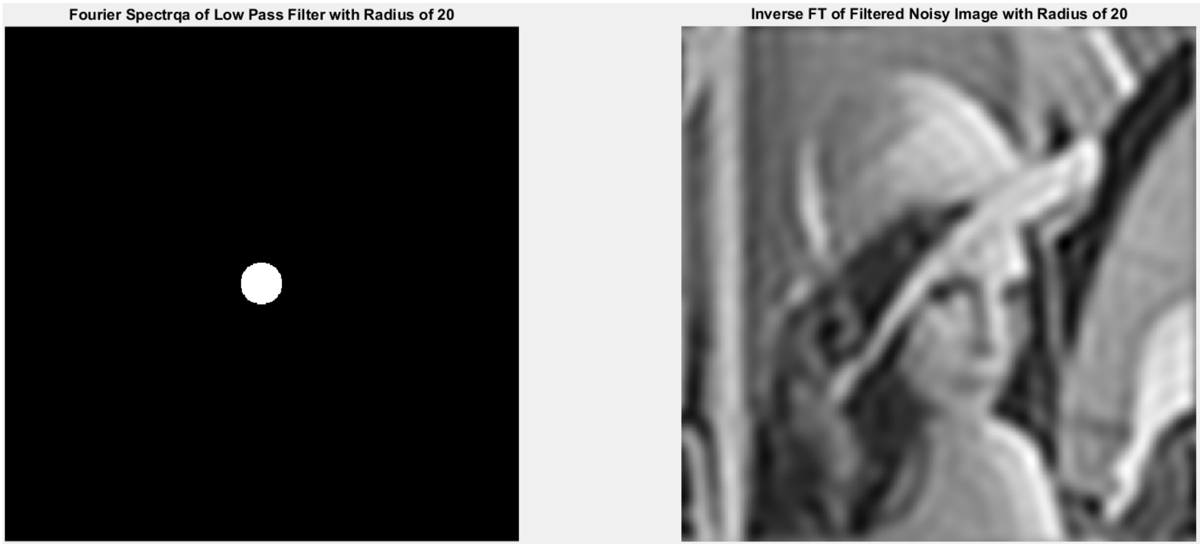

Figure 9: Ideal filter of radius 20 and the associated filtered image

3.4 Gaussian Filter

A Gaussian filter with standard deviation of 60 is created in frequency domain. The filter does not have a sharp cut-off, instead, frequencies are attenuated along a radial gradient. Figure 10 shows the result of applying the filter.

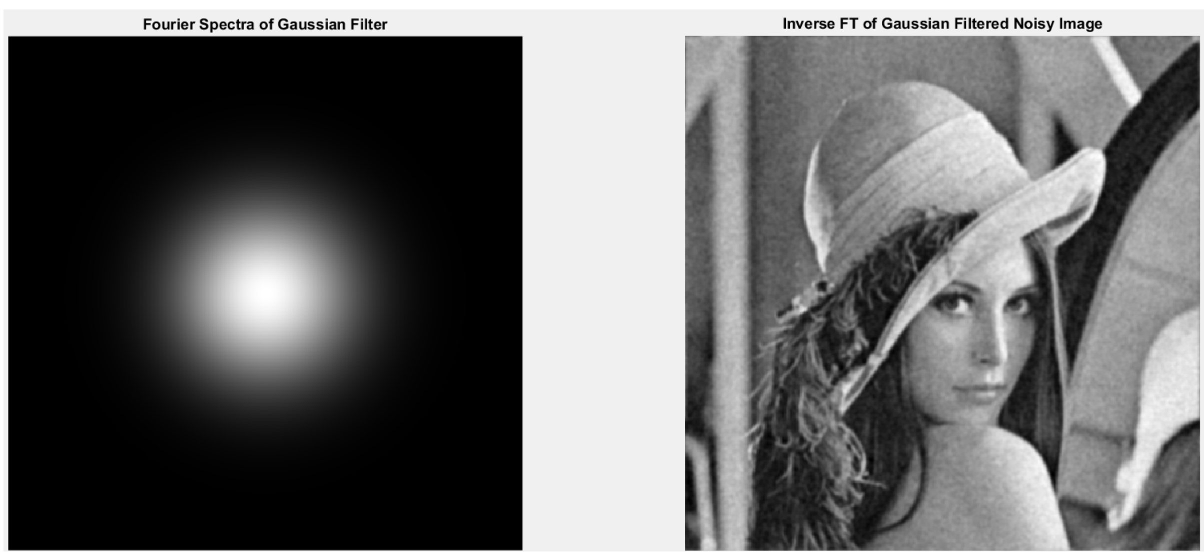

Figure 10: Gaussian filter with standard deviation of 60 and the associated filtered image

The resulting image contains better color contrast. The spotted noise pattern and ringing artifacts are non-existent. The Gaussian filter is very effective at removing Gaussian noise since it is sampled from the same distribution. Recalling that an ideal filter's sharp cut-off corresponds to a sinc function in spatial domain, one can conclude that the smooth fall-off of the Gaussian filter helps produce a spatial domain signal that does not include notable oscillations. As a result, the ringing artifact is completely negated in the filtered image.

Nonetheless, the PSNR of the Gaussian filtered image is **29.53** and is significantly better than the ideal filter of similar radius. This aligns with expectation - since the applied noise is Gaussian, a Gaussian filter is more effective at inverting it.

4 Filter Design for Pattern Rejection

The image in this problem suffers from patterns contaminating the image. The patterns are consistent and pronounced, which leads to the conclusion that they may be caused by pure sinusoids of relatively high magnitude. Such sinusoids should appear as outliers as their relative magnitude and phase is not coherent with the rest of the image. This is verified by performing the Fourier transform and studying the image in frequency domain. The resulting Fourier spectra is shown in Figure 11.

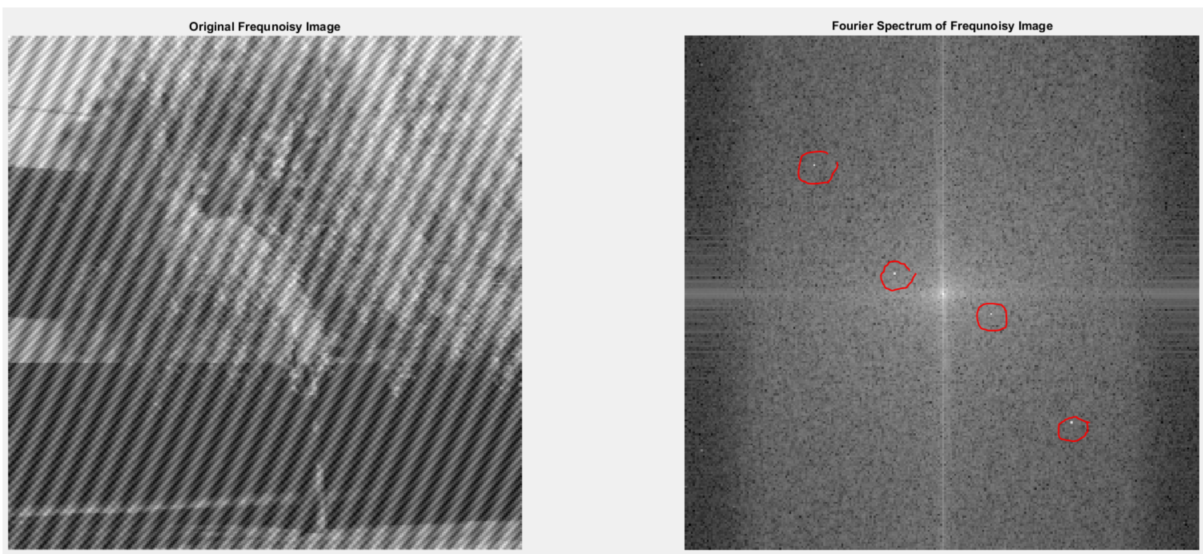

Figure 11: Source image and its corresponding Fourier spectra; the white dots are circled in red for visibility

The spectra depicts several discretely pronounced white dots that represent undesirable components of varying frequencies that are not abiding by the overall spectra. In order to rectify the image, the white dots need to be suppressed – in other words, the violating frequency components must be attenuated to get rid of their superimposing grated pattern.

A typical low pass filter does not suffice as it will accidentally filter desirable frequencies. Instead, what is sought is a 2D Notch filter. A Notch filter selectively attenuates a small band of frequencies. Using the low pass filter design earlier, a Notch filter is formed by creating separate ideal low pass filters that are offset to overlap with the dots. The filter is ideally sized small such that surrounding frequencies are left unharmed. After producing the localized low pass filters, a cumulative filter is formed by adding the resulting low pass filters – this yields one filter that masks several frequency bands. Finally, an inverse histogram transformation is performed to invert the intensities of the mask. This turns the white, localized filters into black circles; this in turns makes it so that only the frequencies within the localized filters are suppressed. The resulting filter is shown in Figure 12.

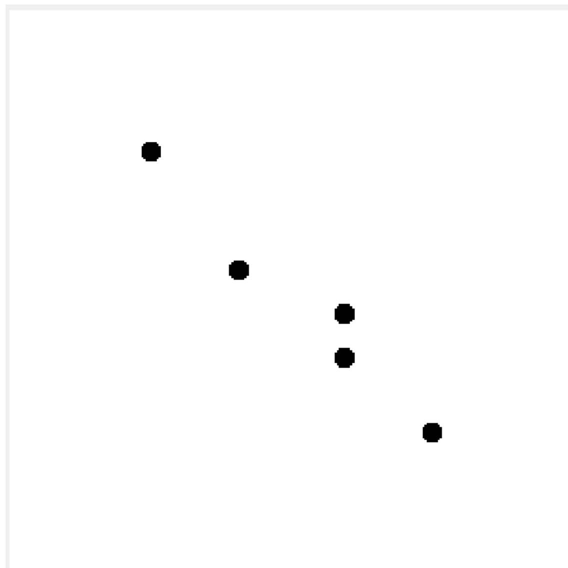

Figure 12: Resulting Notch filter formed by the manipulation of several localized ideal low pass filters

The resulting Notch filter is applied directly to the image via element-wise multiplication to produce the Fourier spectra shown in Figure 13 – note how the white dots are now eliminated by the 0 energy bands. By taking the inverse Fourier transform, we arrive at an image free of undesirable grating patterns as seen in Figure 14. The size of the Notch filter can be easily modified to reduce impact on surrounding frequencies. The filter is sized reasonably large for display purposes.

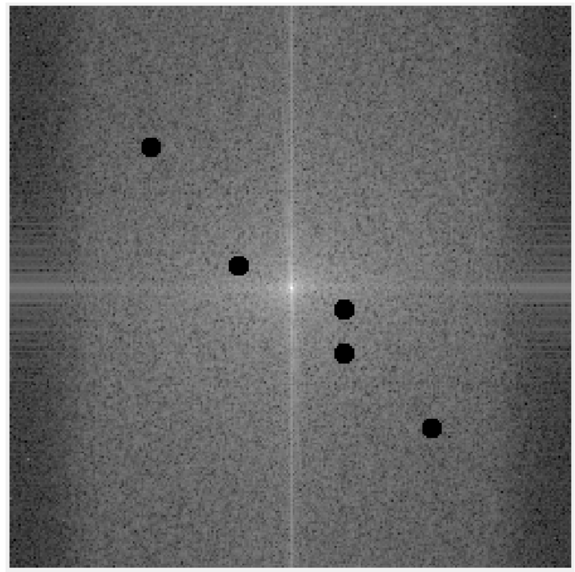

Figure 13: Resulting spectra after applying the notch filter

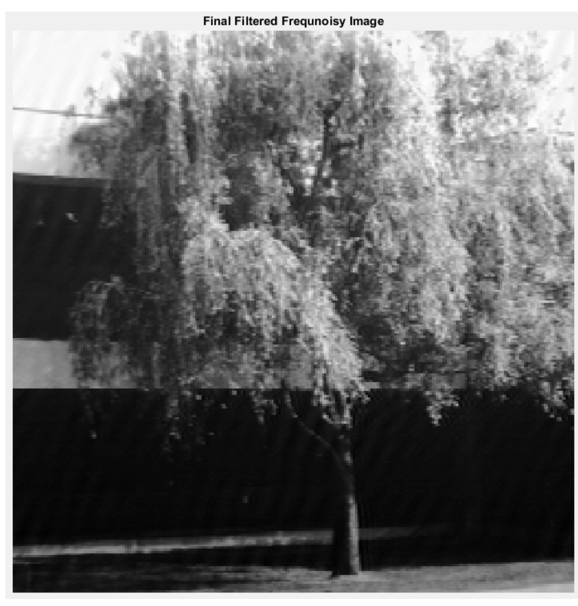

Figure 14: Final image after applying the frequency domain filter

5 Conclusion

In this lab, we performed image restoration and comprehension by applying various filters in frequency domain and examining the effects of magnitude and phase plots. The magnitude plot provided useful insight when identifying compositions of interest within an image - this is typically in the form of pure sinusoidal noise, DC gain, and presence of high frequency components such as ones introduced by Gaussian noise.

Conversely, performing the inverse transformation solely on the magnitude component yields an incoherent spatial image. The phase plot however gives a directional indication as a result of the offset of sinusoidal components from each other - as a result, taking the inverse transform produced an image representative of the shape or outline of the source image.

The ideal low pass filter with varying degree of cut-off radii, and low pass Gaussian filter were also applied in frequency domain and the effects were analyzed. The ringing artifact was discovered when using ideal filtering - a consequence that arises from the sinc function representation of the filter in time domain. In order to alleviate the artifact, the frequency domain filter must be selected to deviate from a shape that would produce a sinc function – this is effectively achieved by smoothing out the edges of the filter as seen in the Gaussian filter case. The cut-off radius correlates to cut-off frequency; increasing the radius has the effect of preserving more frequencies within the filter's masking region.

Lastly, a specialized Notch filter was implemented for the purpose of rejecting sinusoidal components contributing to grating patterns in an image. Directional gratings correspond to a sinusoidal component or a combination thereof. As a result, the frequency domain representation exposed white dots indicative of the outlier harmonics that contribute to the undesirable artifacts. The Notch filter masks the dots in an effort to fully attenuate them. Taking the resulting inverse transform yielded an artifact-free image which would have been near impossible to achieve using typical spatial filters.

6 Appendix

6.1 lab3.m

```
close all
clear
clc

%Section 1
cameraman = (imread('cameraman.tiff'))./255;
lena = (imread('lena.tiff'));
frequnoisy = (imread('frequnoisy.tif'));

%Section 2
f = zeros(256,256);
f(:,108:148)=1;

%Plotting the test image
figure;
subplot(1,2,1);
imshow(f);
title('Test Image', 'fontsize', 14);

%Fourier Spectra of the image
ft = fft2(f);
shift_ft = fftshift(ft);

subplot(1,2,2);
%imshow(5 * log(1 + abs(shift_ft)), colormap(gray));
imshow((abs(shift_ft)),[]);
title('Fourier Spectra of Test Image', 'fontsize', 14);

%% Rotate 45 degrees
rotate_45 = imrotate(f, 45);
figure
subplot(1,2,1);
imshow(rotate_45);
title('Test Image Rotated 45 degrees', 'fontsize', 14);

%Rotate 45 degrees for fourier
ft_45 = fft2(rotate_45);
shift_ft_45 = fftshift(ft_45);

subplot(1,2,2);
imshow(abs(shift_ft_45), []);
```

```
title('Fourier Spectra of Rotated Test Image', 'fontsize', 14);

%% Amplitude and Phase

glena = rgb2gray(lena);

ft_glena = fft2(glena);
ft_shift_glena = fftshift(ft_glena);
glena_amplitude = abs(ft_glena);
glena_phase = ft_glena./glena_amplitude;

%Inverse Fourier
inv_amplitude = ifft2(glena_amplitude);
inv_phase = ifft2(glena_phase);

%Plot the original image, reconstructed image using amplitude and phase
figure;
imshow(glena);
title('Grayscale Lena', 'fontsize', 14);

figure;
subplot(1,2,1);
imshow(log10(glena_amplitude),[]);
title('Amplitude', 'fontsize', 14);

subplot(1,2,2);
imshow(glena_phase,[]);
title('Phase', 'fontsize', 14);

figure;
subplot(1,2,1);
imshow(log10(inv_amplitude),[]);
title('Reconstructed Image using Amplitude', 'fontsize', 14);

subplot(1,2,2);
imshow(log10(inv_phase),[]);
title('Reconstructed Image using Phase', 'fontsize', 14);


%% Section 3
close all

intensities = im2double(glena);
noisy_lena = imnoise(intensities, 'gaussian' ,0,0.005);
```

```
figure;
subplot(1,2,1);
imshow(intensities, []);
title('Original Lena Image', 'fontsize', 14);
subplot(1,2,2);
imshow(noisy_lena, []);
title('Noisy Lena Image', 'fontsize', 14);


%Fourier Spectra
ft_original_lena = fft2(intensities);
ft_noisy_lena = fft2(noisy_lena);

figure;
subplot(1,2,1);
imshow( log10(abs( fftshift(ft_original_lena) )), []);
title('Log Fourier Spectra of Original Lena Image', 'fontsize', 14);

subplot(1,2,2);
imshow(log10(abs(fftshift(ft_noisy_lena) )), []);
title('Log Fourier Spectra of Noisy Lena Image', 'fontsize', 14);
%%
%Low Pass Filter ...Lena image has 512x512 dimensions
r = 60;
h = fspecial('disk',r); h(h>0)=1;
hfreq = zeros([512],[512]);
hfreq([[512]/2-r:[512]/2+r],[[512]/2-r:[512]/2+r])=h;

%Fourier Spectra of the Low Pass Filter
ft_lpf_shift_lena = hfreq; % h_freq is in frequency domin
figure;
subplot(1,2,1);
imshow(ft_lpf_shift_lena, []);
title('Fourier Spectra of Low Pass Filter with Radius of 60', 'fontsize', 14);

%LPF on Noisy Image
lena_lpf = fftshift(ft_noisy_lena).* ft_lpf_shift_lena;

%Inverse FT of LPF noise image
ift_lena = ifft2(ifftshift(lena_lpf));

%plot
subplot(1,2,2);
imshow(ift_lena);
title('Inverse FT of Filtered Noisy Image with Radius of 60', 'fontsize', 14);
```

```
%PSNR
lena_psnr_60 = psnr(lena, intensities);
lena_psnr_60

%%
%Low Pass Filter .. r =20
r = 20;
h = fspecial('disk',r); h(h>0)=1;
hfreq = zeros([512],[512]);
hfreq([[512]/2-r:[512]/2+r],[[512]/2-r:[512]/2+r])=h;


%Fourier Spectra of the Low Pass Filter 20
ft_lpf_shift_lena20 = hfreq; % h_freq is in frequency domin
figure;
subplot(1,2,1);
imshow(ft_lpf_shift_lena20, []);
title('Fourier Spectrqa of Low Pass Filter with Radius of 20', 'fontsize', 14);

%LPF on Noisy Image 20
lena_lpf20 = fftshift(ft_noisy_lena).* ft_lpf_shift_lena20;

%Inverse FT of LPF noise image 20
ift_lena20 = ifft2(ifftshift(lena_lpf20));

%plot 20
subplot(1,2,2);
imshow(ift_lena20, []);
title('Inverse FT of Filtered Noisy Image with Radius of 20', 'fontsize', 14);

%PSNR 20
lena_psnr_20 = 1psnr(ift_lena20, intensities);
lena_psnr_20

%%
%Gaussian Filter
f_gaussian_filter = fspecial('gaussian',512,60);
f_gaussian_filter = f_gaussian_filter./(max(max(f_gaussian_filter)));

figure;
imshow(f_gaussian_filter);
title('Gaussian Filter', 'fontsize', 14);

figure;
```

```
subplot(1,2,1);
imshow(f_gaussian_filter, []);
title('Fourier Spectra of Gaussian Filter', 'fontsize', 14);

% Apply filter onto noisy image
lena_gaussian_filter = fftshift(ft_noisy_lena).*f_gaussian_filter;

% Inverse FT on filtered noisy image
ift_lena_gaus = ifft2(ifftshift(lena_gaussian_filter));

% Plot the images
subplot(1,2,2);
imshow(ift_lena_gaus,[]);
title('Inverse FT of Gaussian Filtered Noisy Image', 'fontsize', 14);

% PSNR
lena_psnr_gaussian = psnr(ift_lena_gaus, intensities);
lena_psnr_gaussian

%% Section 4 Design your own Filter
close all
frequnoisy = (imread('frequnoisy.tif'));

figure;
subplot(1,2,1);
imshow(frequnoisy);
title('Original Frequnoisy Image');

ft_image_noise = fft2(frequnoisy);
ft_shift = fftshift(ft_image_noise);
subplot(1,2,2);
imshow(log(abs(ft_shift)), []);
title('Fourier Spectrum of Frequnoisy Image');


% delete points at (65,65), (105,119), (153,159), (193, 193)
% Naaah, we need a notch filter - create those same circles on the area of the
% dots and apply them successively.

% ft_image_noise(65,65)=ft_image_noise(65,66);
% ft_image_noise(105,119)=ft_image_noise(105,120);
% ft_image_noise(153,159)=ft_image_noise(153,160);
% ft_image_noise(193,193)=ft_image_noise(193,194);

% Create a combination of notch filters.
```

```
f_notch = MakeNotchFilter(frequnoisy, [65,65,105,119,153,159,193,193,153,139],
    4);
figure;
imshow(f_notch, [])

% Apply the filter
ft_notch_frequnoisy = ft_shift.*f_notch; % Both of these images are shifted to
    center
figure;
imshow(log10(abs(ft_notch_frequnoisy)), []);

% This figure is redundant?
figure;
imshow(log(abs(ft_image_noise)), [])

final = ifft2(ifftshift((ft_notch_frequnoisy)));
figure;
imshow(final, []);
title('Final Filtered Frequnoisy Image');
```