EE550 - Image and Video Processing - Lab 1 Report

Baris Sevilmis
Lab Date: 25/09/2019
Due Date: 09/10/2019

1 Overview

First lab assignment of EE550 focuses various fundamental notions of Image Processing. Concepts to be implemented are gamma correction, image quantization, filtering, correlation, resampling, phase and magnitude of the 2DFT and Weber law. Details and experimentation of these concepts are clearly explained in the corresponding sections.

2 Image and Color Tables

In this section, our work consists of basic image operations. After displaying images both in RGB and grayscale, we will invert these images. Lastly, gamma correction with different γ ($\in \mathbb{R}$) will be applied on these images.

2.1 Read and Display Images

Easiest and most important part is to read and display images. By this means, we will be able to visualize the effects of our work and therefore, our understanding will have a much more sound basis. Figure 1 & 2 depict images, that we will be processing, in RGB and grayscale. As a last detail, *lena* images has size of $512x512x3$ and trees has size of $258x380$.

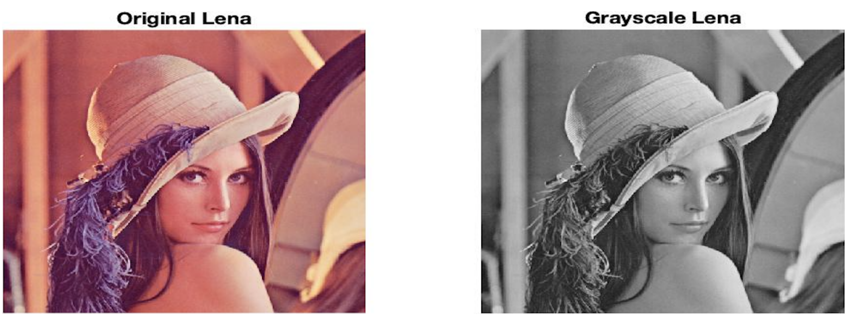

Figure 1: Lena - Original & Grayscale

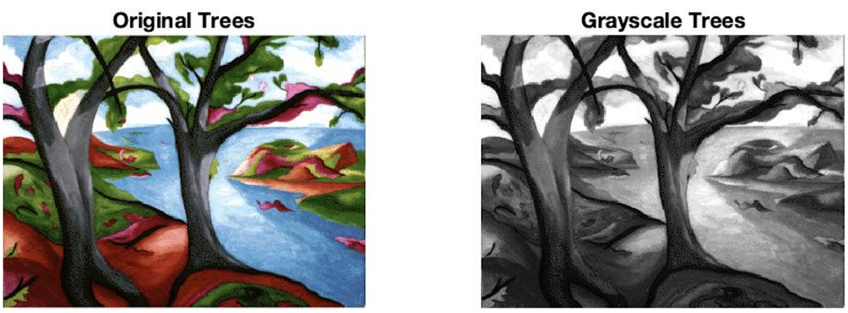

Figure 2: Trees - Original & Grayscale

An important highlight is *trees* and *lena* images have different formats, namely *lena* is RGB directly, and *trees* contains a color map for RGB coloring, but is grayscale on its own. Images are read by **imread()** and displayed by **imshow()** functions in Matlab environment.

2.2 Gray Level & Inversion

In second part, images are inverted. In other words, negative of the images are displayed. grayscale of images are demonstrated in Figure 3 & 4 as well as in Figure 1 & 2, and inverted images are demonstrated in Figure 3 & 4.

Figure 3: Lena - Grayscale & Inverted

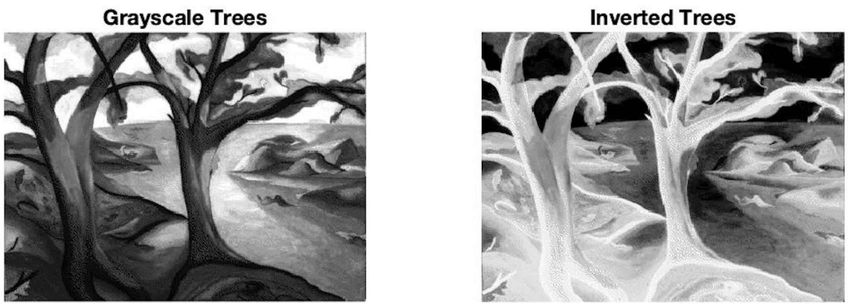

Figure 4: Trees - Grayscale & Inverted

To invert images, 2 different methods are used because of different image formats. *lena* is in RGB scale, and has values in range of $\in [0, 255]$ in each RGB channel. Therefore, $double(255) - lena$ produces inverted *lena* image. On the other hand, *trees* has a color map and color values are represented as double value in range of $\in [0, 1]$. For this reason, $double(1) - color\_map$ produces inverted *trees* image.

2.3 Gamma Correction

In third part of the exercise 2.1, focus is on gamma correction. Following formulas are applied to color channels of both images, namely RGB channels of *lena* & color map of *trees*:

$$r^{'} = r^{\gamma}$$
$$g^{'} = g^{\gamma}$$
$$b^{'} = b^{\gamma}$$

As we will be using γ values greater than 1, images in range of $\in [0, 255]$ have a very high chance of getting out of the range. Therefore, normalization becomes compulsory. Then, various values of γ can be experimented. For the sake of experiments, γ values are chosen from $\in [0, 0.5, 1, 1.5, 2.0]$. Figure 5 & 6 depicts gamma correlated images with their corresponding γ values. Noticeably, change in γ causes detectable contrast change within the images.

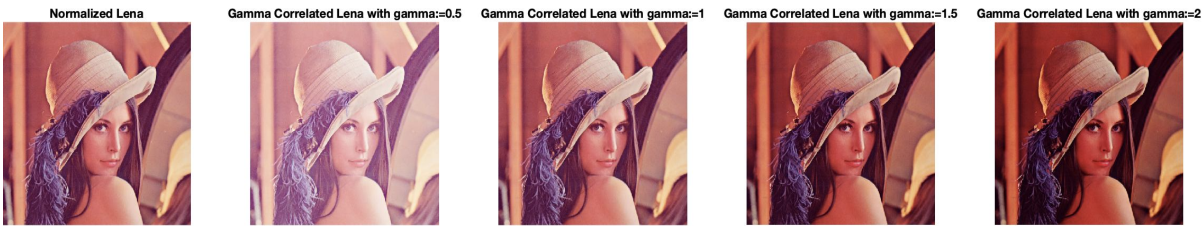

Figure 5: Lena - Gamma Correction: $\in [0, 0.5, 1, 1.5, 2]$

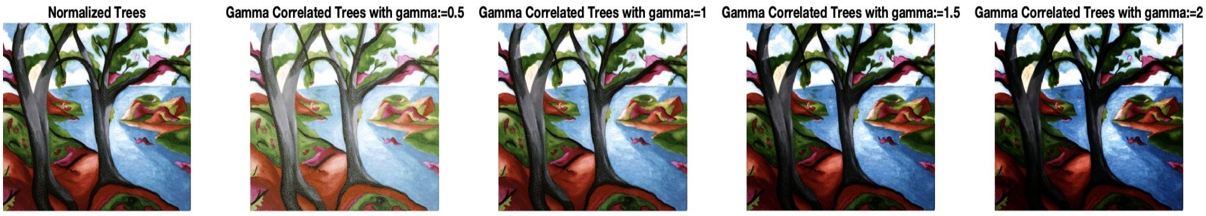

Figure 6: Trees - Gamma Correction: $\in [0, 0.5, 1, 1.5, 2]$

2.4 Chessboard, *Indexed* & *Truecolor*

In the last part of the exercise 2.1, main task is to build 2 chessboards based on different methods, namely RGB and colormap. RGB chessboard has size of $8x8x3$, and color map chessboard has size of $8x8$ with color map itself has size $256x3$. In color map, for specific pixel values, 3 channels corresponding to RGB values are provided. These RGB values are applied for those specific pixels containing those intensity values. Since we are working in range of $\in [0, 255]$ for simplicity, following formulas are applied for specific pixels:

$$RGB : k = 0 \rightarrow 8 : r = 255, g = 255, b = 0 (k := k + 2)$$
$$RGB : k = 1 \rightarrow 8 : r = 0, g = 0, b = 255 (k := k + 2)$$
$$Color\_Map : Value(0) := CM(1) : r = 255, g = 255, b = 0$$
$$Color\_Map : Value(255) := CM(255) : r = 0, g = 0, b = 255$$

Resulting images are illustrated within Figure 7. These images are additionally saved in to *TIFF* format as requested.

Figure 7: Truecolor and indexed chessboards ($8x8$)

3 Image Quantization

Second exercise of the lab assignment consists of a technique called image quantization. Main idea is to determine reconstruct images by adjusting quantization step size. For the sake of experimentation, and additionally given constraints, quantization are applied on grayscale images. Reconstructed images need to have specific gray levels due to reconstruction phase. Uniform quantization is used for quantization. Uniform quantization is defined as following:

$$Q(x) = \left\lfloor \frac{x}{\Delta} \right\rfloor$$

Δ stands for quantization step size, which is used to determine gray level of the reconstructed image. $\hat{x}$ happens to be reconstructed value as below:

$$\hat{x} = Q(x)\Delta + \frac{\Delta}{2}$$

These formulas are applied to provided *lena-y* image in the interest of reconstructing the same image with specific gray levels, namely $gray\_level \in [128, 64, 32, 16, 8, 4, 2]$. Figure 8 demonstrates original *lena-y* and grayscale *lena-y* because of the demonstration convention used in the report, although they are exactly same images. Figure 9 illustrates *lena-y* image itself, and reconstructed versions of this image with corresponding quantization steps to generate requested gray levels.

Figure 8: Original and Greyscale *lena-y*

Figure 9: Reconstructed *lena-y* images according to requested gray levels

4 Filtering

In this part of lab exercise, 2.3, main aim is to convolve provided *gold-text* image with given filters, extract frequency responses of filters, interpret resulting images after convolution operations. Convolution operation in 2D is given by following mathematical formula:

$$x(k,l) **g(k,l) = \sum_{k'=-\infty}^{\infty} \sum_{l'=-\infty}^{\infty} x(k',l')g(k-k',l-l')$$

First filter we will use is a $5x5$ separable 2D filter, to be more specific $filt\_1(2D)$ as in above matrix:

$$filt\_1(2D) = \begin{bmatrix} 0.001274490 & 0.008607270 & 0.015936480 & 0.008607270 & 0.001274490 \\ 0.008607270 & 0.058129210 & 0.107627040 & 0.058129210 & 0.008607270 \\ 0.015936480 & 0.107627040 & 0.199272960 & 0.107627040 & 0.015936480 \\ 0.008607270 & 0.058129210 & 0.107627040 & 0.058129210 & 0.008607270 \\ 0.001274490 & 0.008607270 & 0.015936480 & 0.008607270 & 0.001274490 \end{bmatrix}$$

This filter can be obtained by the operation: $filt\_1(1D) * filt\_1(1D).T$ where $.T$ stands for the *transpose* operation. This operation can be shown as below:

$$filt\_1(1D) = \begin{bmatrix} 0.0357 \\ 0.2411 \\ 0.4464 \\ 0.2411 \\ 0.0357 \end{bmatrix}$$

$$filt\_1(2D) = \begin{bmatrix} 0.0357 \\ 0.2411 \\ 0.4464 \\ 0.2411 \\ 0.0357 \end{bmatrix} * \begin{bmatrix} 0.0357 & 0.2411 & 0.4464 & 0.2411 & 0.0357 \end{bmatrix}$$

Frequency response of $filt\_1(2D)$ is demonstrated in Figure 10 as above:

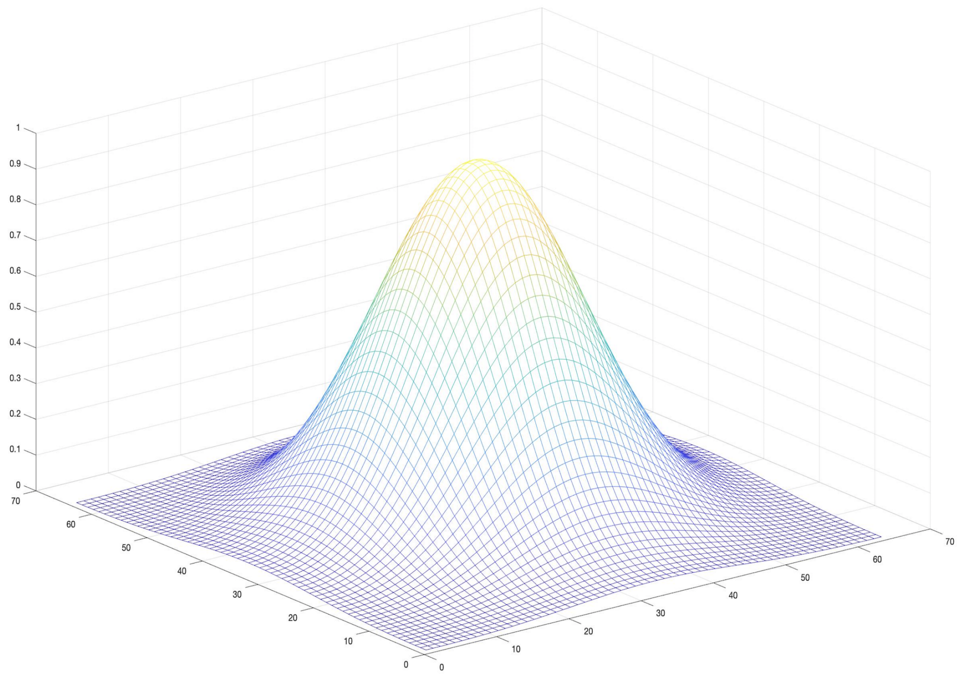

Figure 10: Frequency Response of $filt\_1(2D)$

After convolving *gold-text* with $filt\_1(2D)$, second task is to convolve resulting image with another provided filter ($filt\_2(2D)$)with size($3x3$), which is the following filter:

$$filt\_2(2D) = \begin{bmatrix} -0.1667 & -0.6667 & -0.1667 \\ -0.6667 & 4.3333 & -0.6667 \\ -0.1667 & -0.6667 & -0.1667 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} -1 & -4 & -1 \\ -4 & 26 & -4 \\ -1 & -4 & -1 \end{bmatrix}$$

Frequency response of $filt\_2(2D)$ is depicted above in Figure 11:

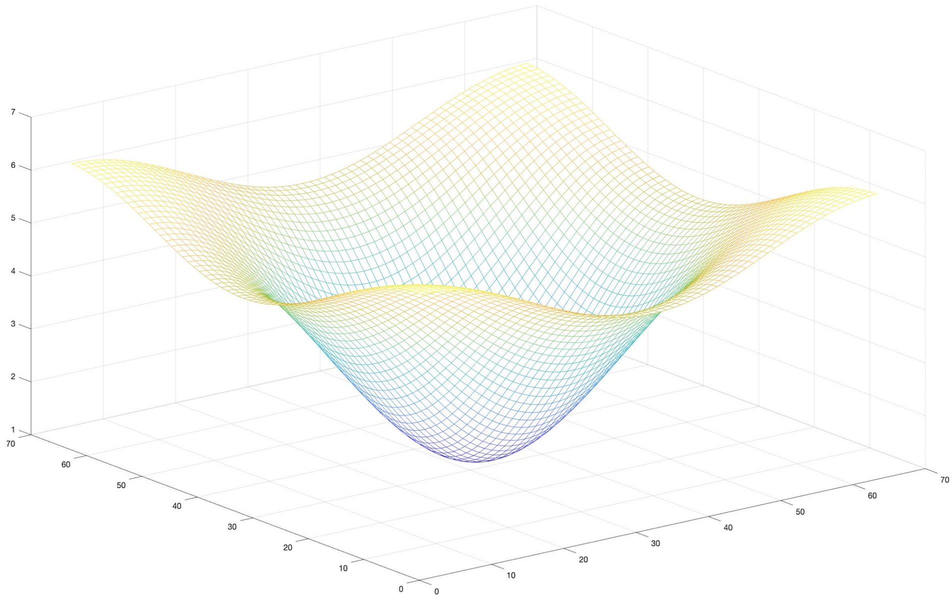

Figure 11: Frequency Response of $filt\_2(2D)$

Although results of convolution after both filters is relatively easy to understand, it would be better to discuss results after visually seeing them. Therefore, Figure 12, provides resulting *gold-text* images after convolution operations and Figure 13 demonstrates intensity plots of these resulting images:

Figure 12: Convolution of *gold-text* with $filt\_1(2D)$ & $filt\_2(2D)$

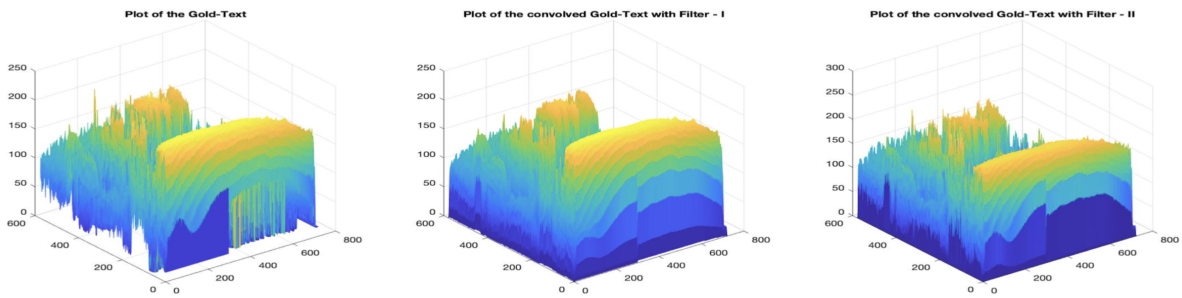

Figure 13: Intensity map of *gold-text* after convolution with $filt\_1(2D)$ & $filt\_2(2D)$

After applying $filt\_1(2D)$ to image, it is clear that edge contrasts within the image are reduced. In other words, intensity values of edges and corners are reduced, which can also be seen by intensity maps as some of the spikes disappeared. Reason of this because, we are using a low pass filter that causes blurring effect. As we convolve the resulting image with $filt\_2(2D)$, this effect is reversed upto a certain level, and therefore edges are enhanced nearly into their original values. It is also illustrated in the frequency response of filters, in which they have their frequency responses in reversed direction. Therefore, it is obvious that $filt\_2(2D)$ is undo-ing effect of $filt\_1(2D)$.

5 Correlation

In this section of the lab assignment, our focus is on the correlation operator, which is considered as similarity comparator between two signals. Correlation in 2D has a very similar to convolution in 2D, such that it can be written in terms of convolution:

$$\varphi(k,l) = \sum_{k'=-\infty}^{\infty} \sum_{l'=-\infty}^{\infty} x(k',l')g(k+k',l+l')$$

$$\varphi(k,l) = x(-k,-l) * *g(k,l)$$

Our task is to find correlation between images *gold-text* & *g-letter*. By finding the maximum value in correlation, we obtain point with highest similarity between given images. In terms of the given images, we are searching for *g-letter* image within *gold-text*. Combining these two aspects, maximum point of correlation corresponds to *g-letter* within *gold-text* image.

Second task is to do correlation using the Fourier(frequency) domain, instead of the spatial domain. If we take 2DFT of both images and take element-wise multiplication of *gold-text* and *conjugate(g-letter)*, taking Inverse-2DFT of the resulting 2D signal and finding maximum of this 2D signal should provide us the same point, given that translation factor is taken into consideration and we are working with zero nominal average images with a symmetric dynamic range with respect to zero($[-128, 127]$ for image range $[0, 255]$). Zero nominal average simply corresponds to $image - 128$, which takes us to dyanmic range mentioned above since experimentation is done in range of $[-128, 127]$.

Lastly, we have to do same set operations on same image but with different noise levels, namely add a noise to standard images with a normal distribution with standard deviations of $[5, 10, 25, 40, 50]$. Since noise addition happens randomly, correlation results differ. Although frequency domain and spatial domain correlation results show in same point, there is a high probability of correlation result to be at a random location. Because of this randomness, different experimentation results are shown in Figure 14, 15 & 16:

Figure 14: Correlation Results - I

Figure 15: Correlation Results - II

Figure 16: Correlation Results - III

Some results are not illustrated within images, as they are outside given image boundaries. Actual correlation result(std. deviation of noise = 0.0) are demonstrated in every image, where these points are directly located right at the bottom *g-letter* location, namely correlation operation succeeds both in spatial domain and frequency domain. However, other detected points are due to noise added to image. Every time new experimentation are done, these points seem to be at random pixel coordinates due to the randomness of noise addition.

6 Resampling

In this section of lab assignment, our focus will be on downsampling, in which are trying to represent the same 2D signal/image with a subset of our original data/pixels. Therefore, downsampling corresponds to choosing subset of data for reconstructing image. In other words, we will remove some of the pixels, and resample image from the remaining pixels. Sampling of a 2D signal in spatial domain is defined as:

$$s_e(x, y) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f(x, y)\delta(x - k\Delta x, y - l\Delta y)$$

In frequency domain:

$$S_e(u, v) = \frac{1}{\Delta x \Delta y} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} F(u - \frac{k}{\Delta x}, v - \frac{l}{\Delta y})$$

As follows for a analog signal to be constructed perfectly:

$$\frac{1}{\Delta x} \geq 2U_{max}$$
$$\frac{1}{\Delta y} \geq 2V_{max}$$

In our case, we are downsampling *sub4* image by a factor of 2, to be more specific one out of 2 neighboring pixels are collected in horizontal and vertical direction. Original image is visualized in leftmost part of Figure 17. Mentioned downsampling result is illustrated as the middle image in Figure 17. It is compulsory to state that reconstructed image is still representative of the original image. There are still some extra frequency components on the reconstructed image. These extra "contours" are called as spectral overlaps or Moire patterns due to the lack of precaution during downsampling and they are seen mostly in high frequency areas of time domain. Nevertheless, reconstructed signal has enough frequency components to represent original *sub4* image. Rightmost image in Figure 17 corresponds to another downsampling on the resulting image, to be specific image in the middle. However, downsampling factor is 4 instead of 2 both in horizontal and vertical direction. Result of this reconstruction is not nearly as successful, as it was in the first downsampling. There are not some Moire patterns, or basic spectral overlaps, but frequency components of the resulting image are not representing the downsampled image at all. Therefore, resulting image has only slight similarities with the downsampled image due to the resampling factor.

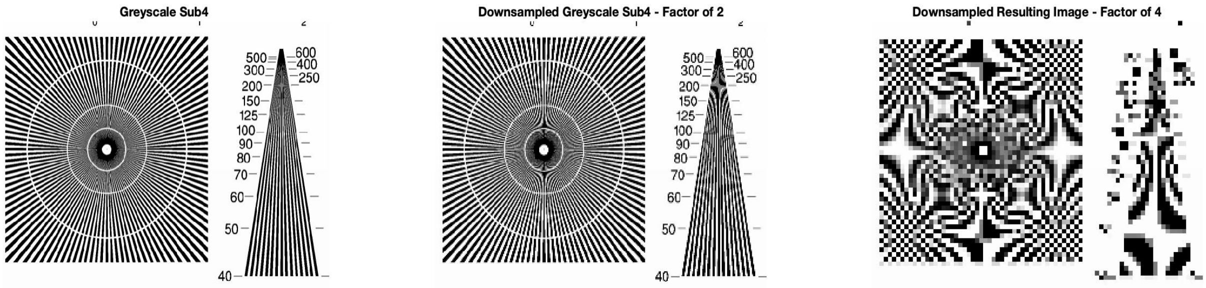

Figure 17: Resampling: Moire Patterns

7 Phase and Magnitude of 2DFT

7.1 Real and Imaginary Part of 2DFT

In first subsection of the exercise 2.6, our focus will be over understanding $\mathbb{R}$&$\mathbb{C}$ components of given image by taking 2D Discrete Fourier Transform of *lena-y* & taking the Inverse 2D Discrete Fourier Transform by removing either $\mathbb{C}$ or $\mathbb{R}$ components of images. Consequently, demonstrating visualizations of these new images, will improve our understanding over these $\mathbb{R}$&$\mathbb{C}$ further. 2DFT of a 2D signal/image is considered as the following formula:

$$2DFT(x(k,l)), (k,l) \in (K,L)$$

$$X(m,n) = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} x(k,l)e^{-2j\pi(\frac{mk}{K}\frac{nl}{L})} = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} x(k,l)e^{-2j\pi\frac{mk}{K}}e^{-2j\pi\frac{nl}{L}} = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1} x(k,l)W_K^{mk}W_L^{nl}$$

$$(m,n) \in ([0,..,K),[0,..,L))$$

$$Inverse - 2DFT(X(m,n)), (m,n) \in (M,N)$$

$$x(k,l) = \frac{1}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X(m,n)e^{2j\pi(\frac{mk}{M}\frac{nl}{N})} = \frac{1}{\sqrt{MN}}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} X(m,n)e^{2j\pi\frac{mk}{M}}e^{2j\pi\frac{nl}{N}}$$

$$(k,l) \in ([0,..,M),[0,..,N))$$

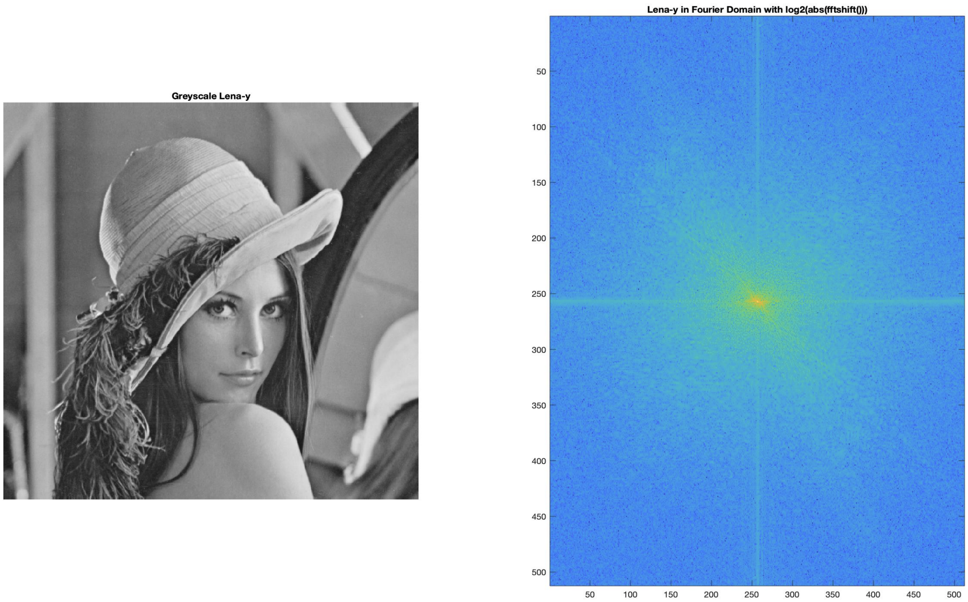

Figure 18: Grayscale Lena-y & Fourier Domain of Lena-y

Figure 18 depicts original *lena-y* image and its frequency distribution in log(absolute) form for visualization purposes.

Figure 19: Real and Imaginary Part of Image(Non Dynamic Range)

Figure 19 illustrates visualization when we take only real or imaginary part of the image after 2D Discrete Fourier Transform and turn back into the spatial domain by Inverse 2D Discrete Fourier Transform. Left image corresponds to *only real* part after the transform and right image corresponds to *only imaginary* part of the image. Right image seems darker when visualizing, because it has negative values and during visualization all of these values are converted to 0, although both of the results should be same. Reason of negative values existence is us working with complex part, and we have to multiply this complex part with i before display to understand data on complex part of the image.

Figure 20: Real and Imaginary Part of Image(Dynamic Range)

As a consequence, we change value range of imaginary part of image. Final results of real and imaginary part are depicted in Figure 20. Both of the results are same, but in different value ranges naturally. It is obvious that in both cases there is Lena and 180 degrees rotated Lena conjugated in the same image. Without the real and complex part of the image, half of the data is for the image missing, and therefore, this kind of result occurs.

7.2 Variations of Phase and Magnitude of the 2DFT

In second part of the exercise 2.6, our main task is to set phase of image during Fourier transform before taking the inverse, and setting magnitude of image to 1 during Fourier transform before taking inverse. These are two separate tasks to comprehend effects of both phase being 0 and magnitude being 1 individually.

Figure 21 demonstrates setting phase to 0. To set phase to 0, we have to compute $absolute(image)$ within the frequency domain(as requested). As we turn back into the spatial domain using inverse Fourier transform, we take the real part of the image and display the result. Resulting values are not in a reasonable range for visualizing the result, for this reason three different methods are tried. In leftmost image, minimum value of image is being subtracted from every pixel, but since range is not statically changed relative to $[0, 255]$ range, result is as same as before. Second method is to take log of image, and since logarithm is a monotonic operator produced image should be more comprehensible. As it can be seen, middle image produced by logarithmic operator is better for display purposes. Lastly, we combine both of the methods, in which we first subtract minimum value of the image from the image itself and then take the logarithm. As it can be seen, rightmost image is same as in middle image(*logarithm only*), but using both linear and monotonic operator gives better visual results. To understand effects of phase on images, it would be better to see effect of magnitude first and than interpret both magnitude and phase effect together.

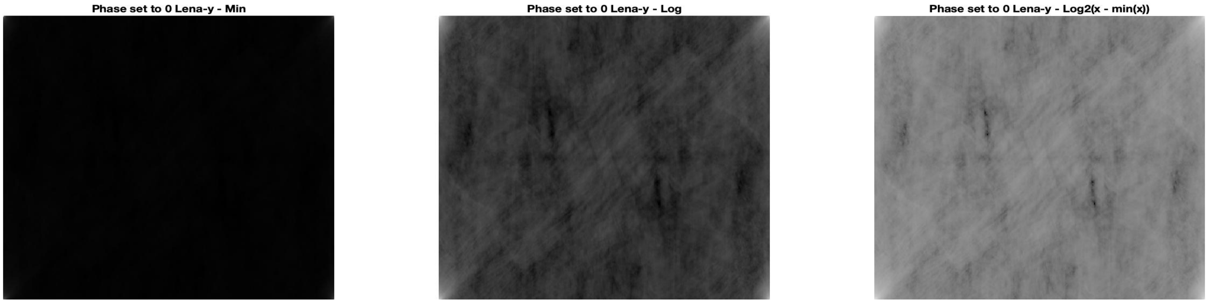

Figure 21: Phase of Image set to 0

Figure 22 illustrates, setting magnitude of image to 0, by dividing image by its $absolute(image)$ in frequency domain(as requested). We turn back into spatial domain by taking inverse Fourier transform and take the real part of the image. Right image is the result of the computation done on the *lena-y* image, namely when its magnitude is set to 0. Edge and corner points of the image are still visible. It would be sensible to conclude that, magnitude of image mainly takes care of high frequency components such as edges and corners. On the other hand, phase of the image is focused on variations in images such as contrast, saturation, color migration and similar features.

Figure 22: Magnitude of Image set to 1

8 Weber Law

Last section of the lab assignment is centered upon Weber law. Weber law is to detect sensitivity level of human visual system based on darkness/brightness level. Result of the experiment indicate human eye is more sensitive to dark gray than light gray. Our objective is to experiment and obtain same result. We will have two different experimentation environment(as suggested in exercise 2.7), in which we will both do tests in light and dark background. We will try to detect changes by varying α values, which is the following:

$$\frac{L_2 - L_1}{L_1} = \frac{\Delta L}{L_1}$$

.

By varying L_1 & L_2 according to two different L_b value $\in 10, 200$, we run two different test cases. Results shown on figures below represent the exact point that a specific human visual system can detect the difference. However, this value can differ from one human to another, therefore different values can be tested for L_1 & L_2 in the provided lab assignment code. Moreover, these tests should be done to exactly understand these change levels. Figure 23 depicts a dark background with $L_b = 10$. L_1 & L_2 values are chosen as really similar values to L_b at first, and then L_2 is changed slightly, and its intensity value is increased. For the sake of testing, detecting change between L_1 & L_2 starts between sub figure 3 & 4($\alpha \in [0.8, 1.2]$). With same conditions but $L_b = 220$, test is recreated in Figure 24. According to the testing human visual system, change starts around sub figure 2 & 3($\alpha \in [0.4, 0.8]$). Results of this test affiliate with the results done under Weber's Law. To be more specific, it is harder for human visual system to divert gray tones in a dark background compared a light background or it is hard to detect changes in gray levels at low intensity environments. Of course, this is a test with many constraints such as starting & finishing test at the same optical illumination conditions and results may differ from an individual to another due to uniqueness of human visual system. However, in our case Weber's Law did hold.

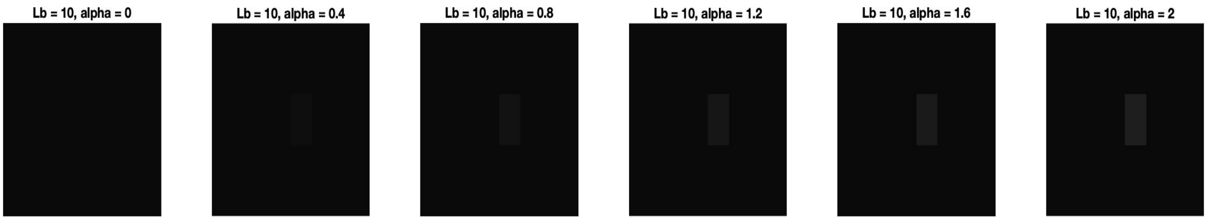

Figure 23: $L_b = 10$,$\alpha \in [0, 2]$

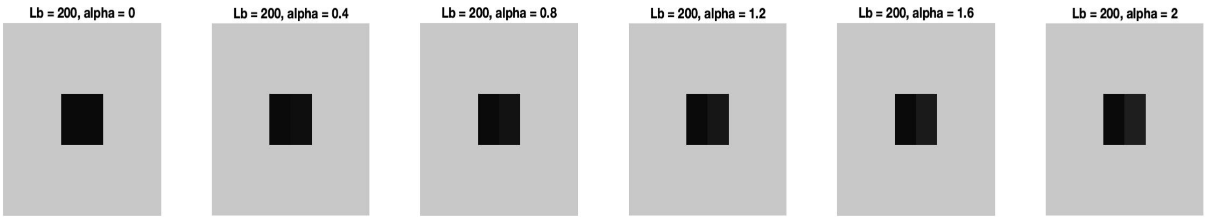

Figure 24: $L_b = 220$,$\alpha \in [0, 2]$