CAP5400 – Digital Image Processing

Assign 3:
Fourier Transform and Frequency Filtering

Name: Zhenglong Wu
U#: U96342243
Date: Nov. 12th, 2019

1. Introduction

This assignment mainly focuses on the Fourier transformation of the image and implement image smoothing, image edge detection and image enhancement in frequency domain of the image. The Fourier transform is to obtain the distribution of the signal in the frequency domain. The digital image is also a kind of signal, and the Fourier transform is also obtained by its spectral data. For discrete signals such as digital images, the frequency magnitude indicates how severe the signal changes or how fast the signal changes. The higher the frequency, the more severe the signal changes, and the smaller the frequency, the more gradual the signal changes. Corresponding to the image, the high frequency signal is often the edge signal and the noise signal in the image, and the low frequency signal contains the image contour and the background. Therefore, when it is necessary to remove noise in the image, high-frequency noise in the image can be removed by designing a low-pass filter.In this experimental report, the frequency filtering in frequency domain will be mainly discussed, and some image processing functions, such as Fourier transformation function and inversed Fourier transformation function in Matlab, will also be called to process images. In the subsequent experiments, Section 2 describes the interpretation of some algorithms. Section 3 describes the operations and implementation details, such as parameters and all run options. In section 4 and 5, the results of experiment and conclusion are presented.

2. Description of algorithm

In this section, there are some explanations of the new algorithms encountered in the experiment. The description of ROI setting method will be omitted, which is to manually set the location and size of the ROI in the image. The smoothing and edge detection algorithms have already been discussed in the previous assignments and their description will not be repeated here. Low-pass filtering, High-pass filtering and band-stop filtering will be explained together in Frequency Filtering.

2.1. Frequency Filtering

Due to there is a correspondence between filters in space and frequency domain, the most fundamental relationship between the spatial and frequency domains is established by convolution theorem. Giving a filter in frequency domain, we can obtain the corresponding filter in the spatial domain by taking the inverse Fourier transform of the former.

Therefore, in frequency filtering, the first step is to use the Fourier transform to obtain the frequency domain of the image. Then, according to the frequency domain information, a cutoff frequency is defined by user. And the signals above the cutoff frequency are discarded is low-pass filtering. On the contrary, the signals below the cutoff frequency are discarded is high-pass filtering. And there is a special case, when it only needs to retain the frequency information in a certain interval, which is band-stop filtering. After filtering out all the frequency information we don't need, the new frequency domain map can be inversely Fourier transformed to get new result image in spatial domain.

2.2. Combine Filtering

When we process the signal, we sometimes need to analyze the combination of several specific intervals of frequency signals. At this point we need to define a number of frequency thresholds, then obtain the frequency domain map of the image according to the frequency filtering algorithm, only retain the information within the user defined cutoff frequencies in the frequency domain. And finally perform the inverse Fourier transform to get the result image.

2.3. Image Sharpening

Image sharpening can be understood as highlighting the detail information of the image, that is, the enhancement of the edge and details, increasing the magnitude of the high frequency components in the frequency domain.

The first step is to identify high frequency signals in the frequency domain. In this experiment, there are four methods to be discussed: Ideal high pass filter, Butterworth high pass filter, Gaussian high pass filter and Laplacian in frequency domain. All four methods can identify the high-frequency information of the image, then combine the new high-frequency frequency domain with the original frequency domain, and finally perform the inverse Fourier transform to obtain a new image with sharpening enhancement.

3. Description of implementation

The entire codes are developed in the MatLab. In parameter file, there are four common parameters for each image processing, which are “input filename”, “output filename”, “operation name” and “ROI location and size”. In addition, some operations needs more parameters, such as the user-defined operation in edge detection. The code reads the parameter file and run the specified functions to get the output image.

parameter.txt – Common section			
Input filename	Output filename	Operation name	ROI location & size
lena.pgm	lena_histStretch.pgm	histStretch	[28,28…;…;…456]
…	…	…	…

Each ROI is defined by 4 parameters: Rx, Ry, Sx, Sy. Rx and Ry is the location value in both x-axis and y-axis of the top left corner of the ROI, Sx and Sy is the size value in width and height of the ROI. The format of ROI parameters is [28,28,200,200; 28,284,200,200;

284,28,200,456]. It is a Nx4 matrix, which has N rows, and each row represents a ROI in the image. And four columns represent Rx, Ry, Sx, Sy separately.

When the program read the ROI information, it will also run an overlapping test right away to ensure there are no two ROIs overlaps in the same image.

parameter.txt – Optional section				
Operation	Parameter1	Parameter2	Parameter3	…
freqFilter	low-pass	[10;20;30]	…	…
…	…	…	…	…

For example, the parameter1 is one of the parameters required by operation "freqFilter", which represents the operation is 'low-pass', 'high-pass' or 'band-stop'. Parameter2 controls the cutoff frequencies of each ROI. Parameter3 is the display information of image's frequency domain.

In the experiment, there are two third-party image processing functions built in MatLab are used:

- fft2(): Two-dimensional fast Fourier transform.
- ifft2 (): Two-dimensional inverse fast Fourier transform.

4. Description of results

4.1. Low-pass filtering and regular smoothing

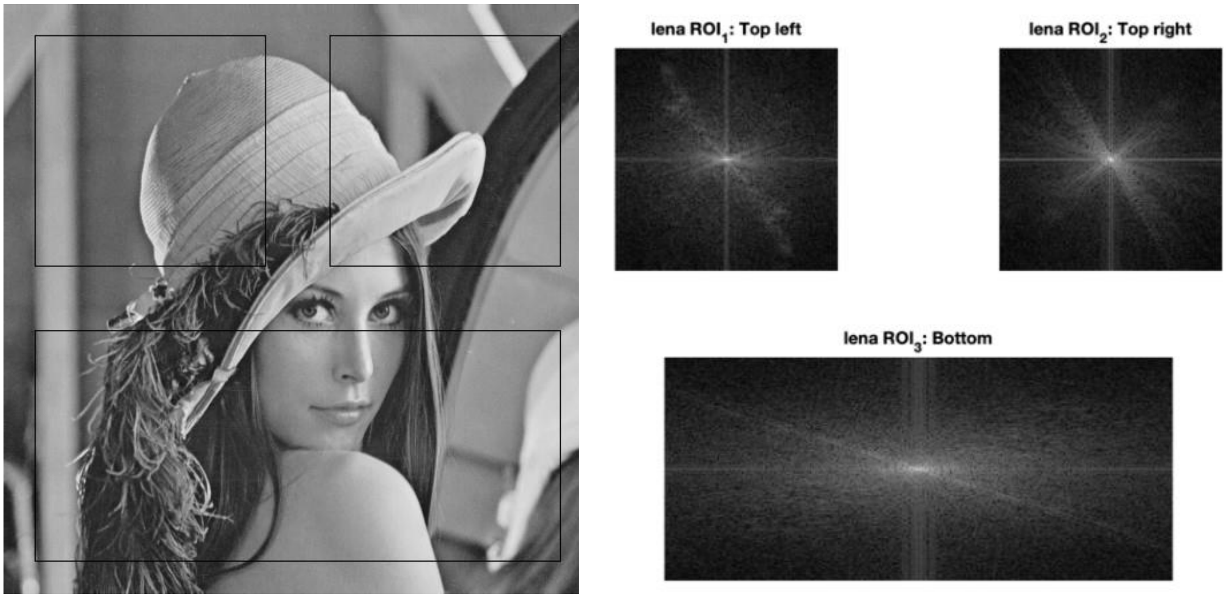

Figure 1.1 Original image and Fourier transform of each ROI

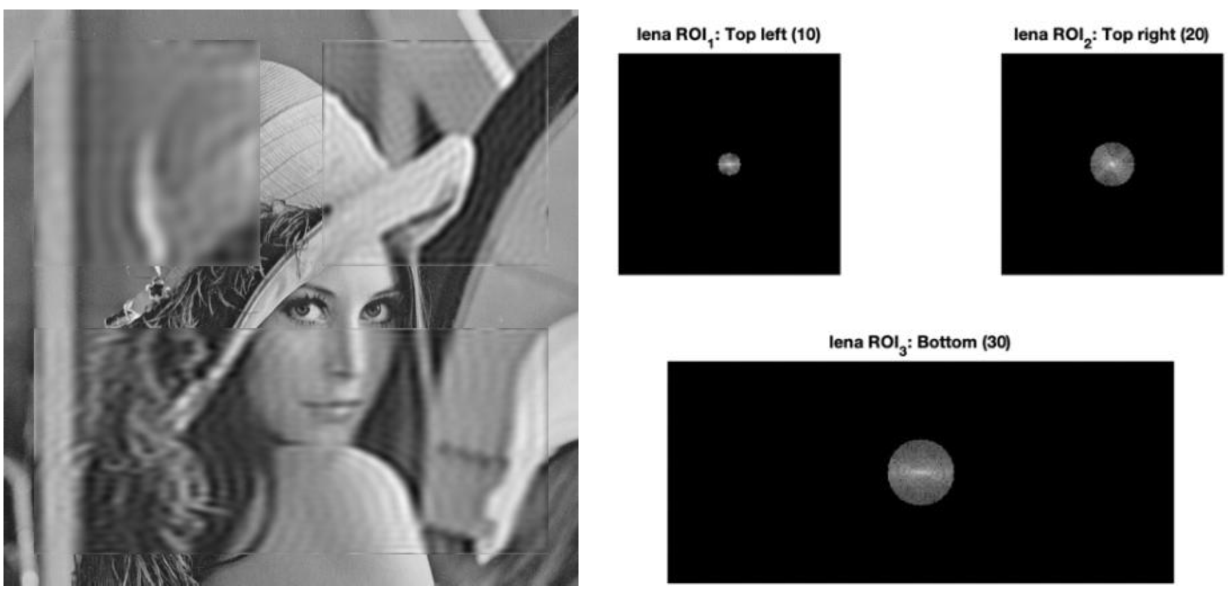

Figure 1.2 Image with low-pass filtering and new Fourier transform of each ROI
ROI_1 cutoff frequency: 10
ROI_2 cutoff frequency: 20
ROI_3 cutoff frequency: 30

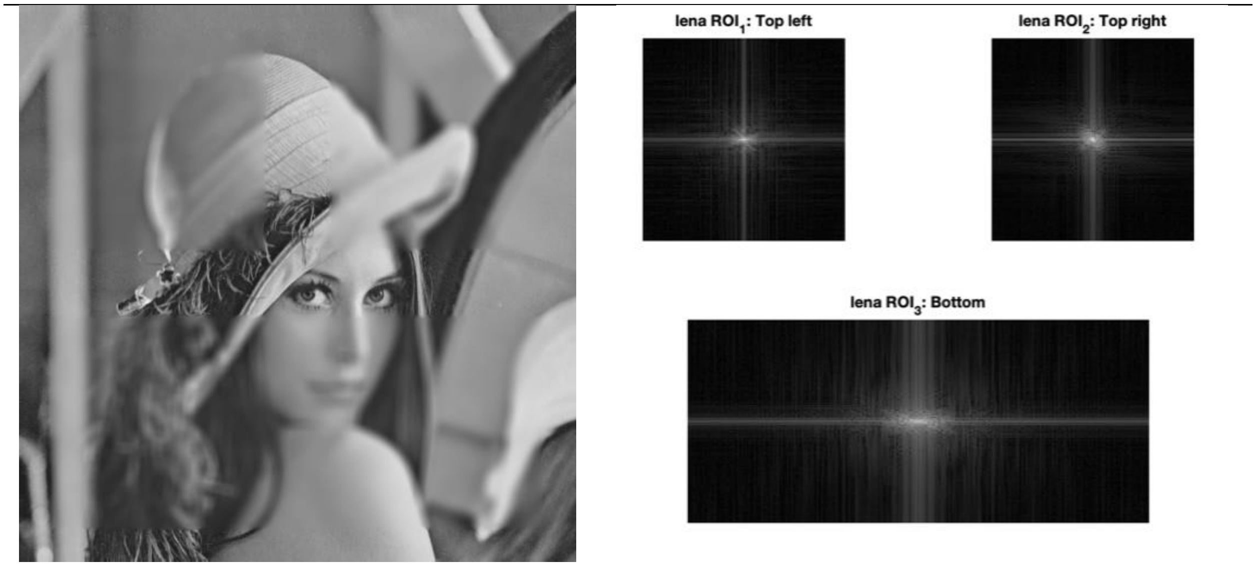

Figure 1.3 Image with adaptive smoothing and Fourier transform of each ROI

The figure 1.1 shows the three ROIs that we choose in the experiment, and the corresponding Fourier transform graph of each ROI. The figure 1.2 shows the image after the smoothing operation. The figure 1.2 shows the image after the low-pass filtering, which based on the formula:

$$G(u,v) = H(u,v) * F(u,v)$$

$$H(u,v) = \begin{cases} 0\ (High\ frequency) & if\ \ D(u,v) \geq D_0 \\ 1\ (Low\ frequency) & if\ \ D(u,v) < D_0 \end{cases}$$

$$D(u,v)\ =\ \sqrt{\left(u-\frac{M}{2}\right)^2+\left(v-\frac{N}{2}\right)^2}$$

F(u, v) is the Fourier transform of the original image, which is implemented by Matlab built-in function fft2(). H(u, v) is the filter mask. And G(u, v) is the Fourier transform new images.

The filter H(u, v) is determined by the distance D(u, v). For centered spectrum, D(u, v) is the Euclidian distance from the point(u, v) to the origin of frequency rectangle. M and N are the length and width of the frequency rectangle. D_0 is the cutoff frequency.

Compare Figure 1.1 with Figure 1.2, we can notice that low pass filtering not only eliminates the noise but also blurs the image, which is very similar to the smoothing. And Compare Figure 1.2 with Figure 1.3, we can notice the difference obviously between both Fourier transform. Low pass filtering only retains the high-energy region in the center of the frequency domain and smoothing only retains the horizontal and vertical lines passing through the center of the frequency domain. And Figure 1.2 with low pass filtering shows many wavy lines in the image.

4.2. High-pass filtering and edge detection

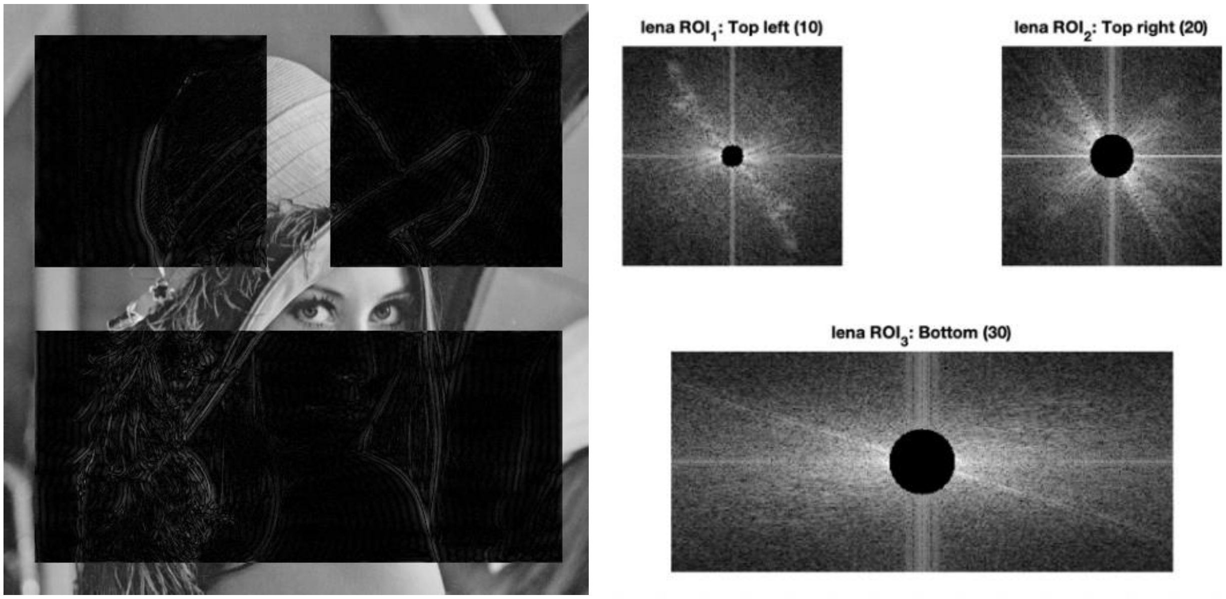

Figure 2.1 Image with high-pass filtering and new Fourier transform of each ROI
ROI_1 cutoff frequency: 10
ROI_2 cutoff frequency: 20
ROI_3 cutoff frequency: 30

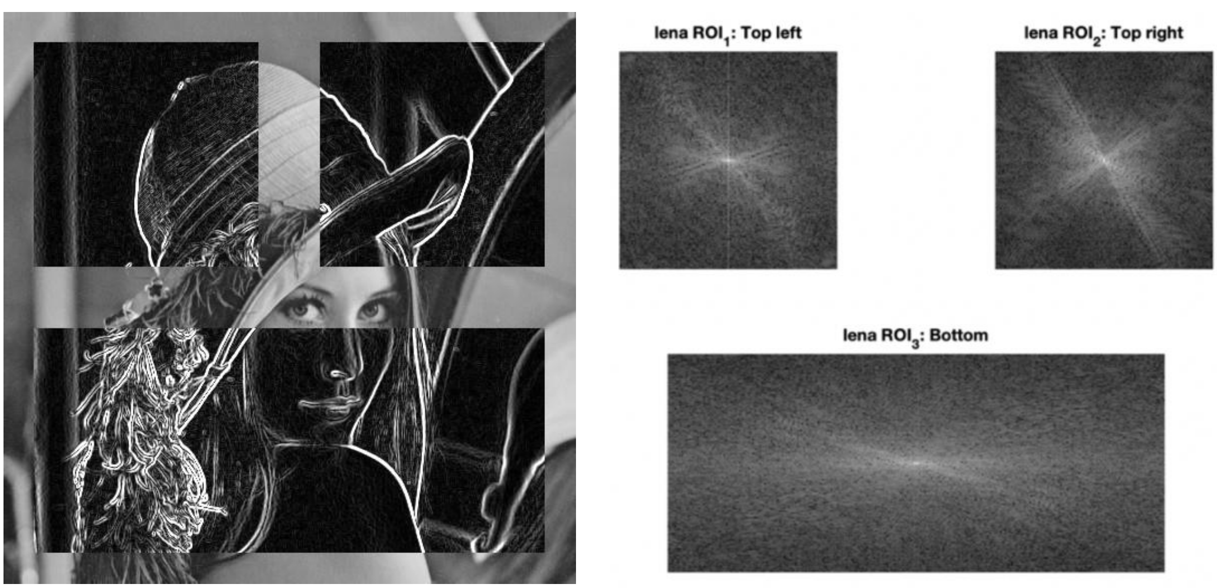

Figure 2.2 Image with 3x3 sobel edge detection and new Fourier transform of each ROI

The figure 2.2 shows the three ROIs that we choose in the experiment with 3x3 sobel edge detection and new Fourier transform of each ROI. The figure 2.1 shows the three ROIs with high-pass filtering and new Fourier transform of each ROI, which based on the formulas similar to the above low-pass frequency filtering.

$$G(u,v) = H(u,v) * F(u,v)$$

$$H(u,v) = \begin{cases} 1 \ (High\ frequency) & if \ \ D(u,v) \geq D_0 \\ 0 \ (Low\ frequency) & if \ \ D(u,v) < D_0 \end{cases}$$

$$D(u,v)\ calculation\ formula\ is\ the\ same\ as\ in\ the\ previous\ experiment$$

Firstly, the Fourier transform of the image F is calculated by fft2() function in the Matlab. Secondly, the H(u, v) will only keep region above the cutoff frequency in the frequency domain. Finally, ifft2() function in the Matlab is used to display the result image.

Compare Figure 1.1 with Figure 2.1, we can notice that high pass filtering removes the high energy region of the image and only left the edges in the image, which is very similar to the edge detection.

Then compare Figure 2.1 with Figure 2.2, we can notice the difference obviously between both Fourier transform. High-pass filtering causes a circular hole in the center of the frequency domain, which is the removed region. However, the adaptive smoothing is more like to distribute the energy that concentrated in the low frequency region into all high frequency regions. And Figure 2.2 with sobel edge detection is clearer than Figure 2.1 with High-pass filtering.

4.3. Band-stop filtering

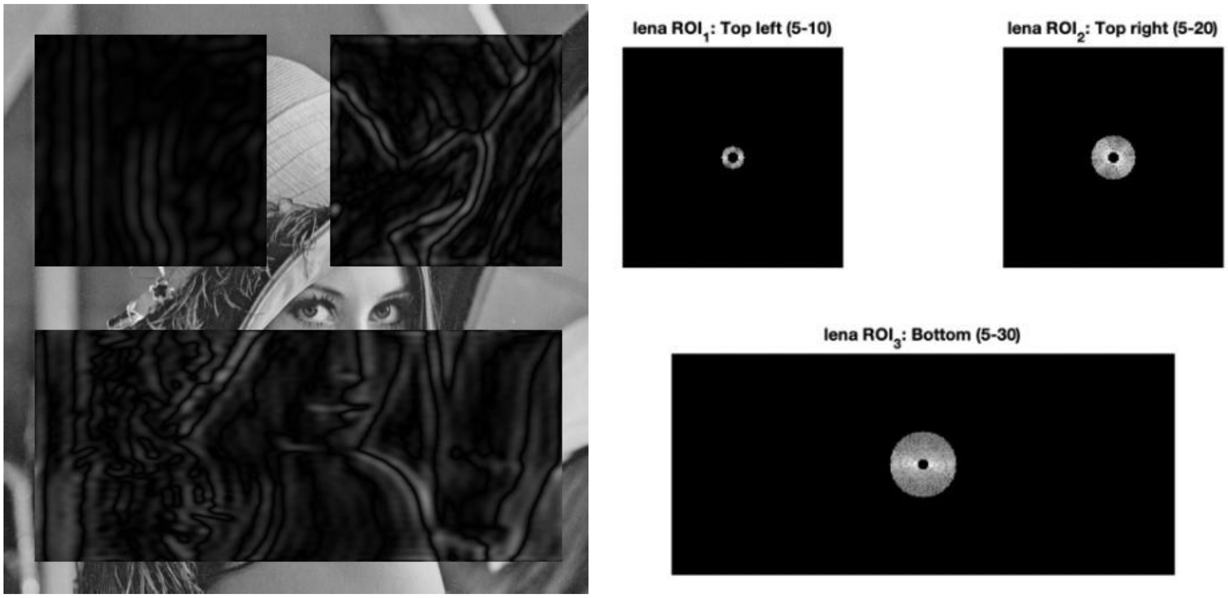

Figure 3.1 Image with band-stop filtering and new Fourier transform of each ROI
ROI_1 cutoff frequencies: 5 and 10
ROI_2 cutoff frequencies: 5 and 20
ROI_3 cutoff frequencies: 5 and 30

Figure 3.1 is the three ROIs that we choose in the experiment with band-stop filtering and new Fourier transform of each ROI, which is implemented based on the formulas similar to the low pass and high pass filtering.

$$G(u,v) = H(u,v) * F(u,v)$$

$$H(u,v) = \begin{cases} 1 & if \;\; D(u,v) \geq D_1 \; and \; D(u,v) < D_2 \\ 0 & else \end{cases}$$

$$D(u,v) \; calculation \; formula \; is \; the \; same \; as \; in \; the \; previous \; experiment$$

D_1 and D_2 are two cutoff frequencies that are defined by the user. Once we get the result of the image Fourier transform, we can use H(u, v) to retain only the frequency region of a certain interval we want, and display the result image by inversed 2D Fourier transform.

Figure 3.1 shows a ring shape in the Fourier frequency domain. And the resulting image can only show the unclear contours.

4.4. Combined filtering

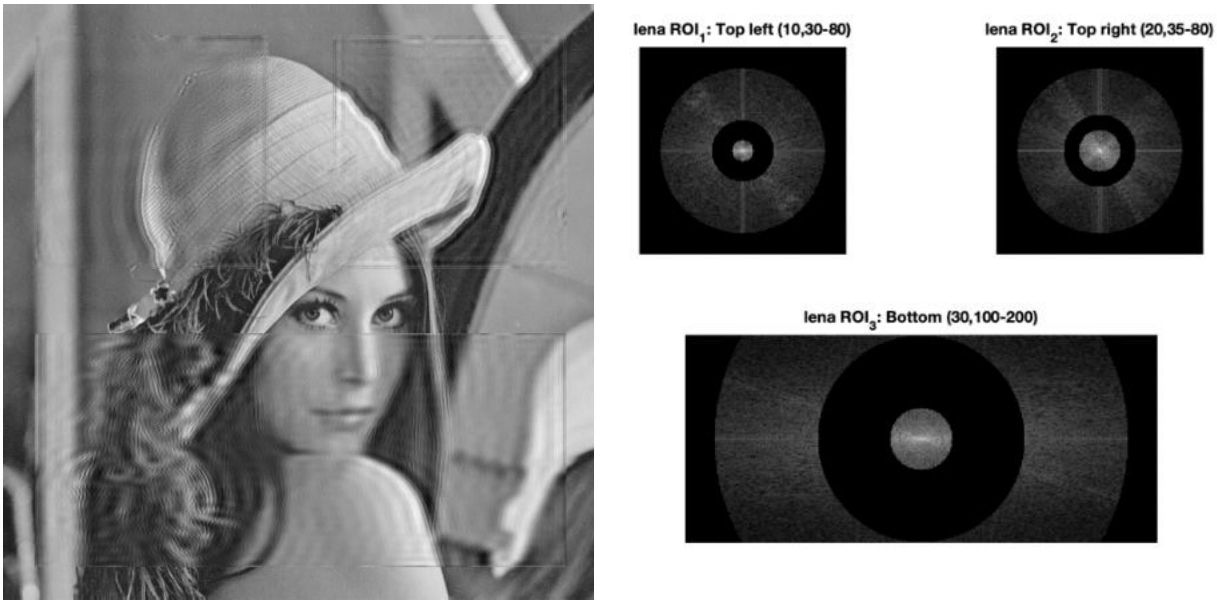

Figure 4.1 Image with low-pass combined filtering and new Fourier transform of each ROI
ROI_1 cutoff frequencies: <10 and (30, 80)
ROI_2 cutoff frequencies: <20 and (35, 80)
ROI_3 cutoff frequencies: <30 and (100, 200)

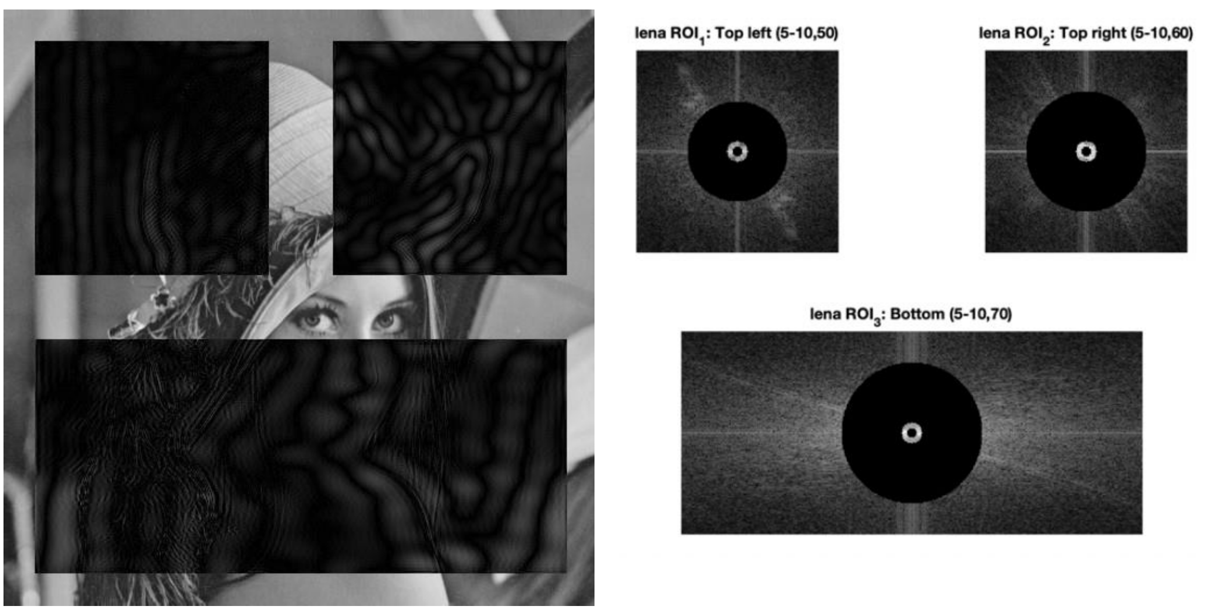

Figure 4.2 Image with high-pass combined filtering and new Fourier transform of each ROI
ROI_1 cutoff frequencies: (5, 10) and >50
ROI_2 cutoff frequencies: (5, 10) and >60
ROI_3 cutoff frequencies: (5, 10) and >70

Figure 4.1 shows the three ROIs we selected in the experiment combining band-stop filtering with low-pass filtering, and the new Fourier transform for each ROI. This is implemented by combining the filter H(u, v) of both band-stop and low-pass.

The algorithm is still the same as before. After obtaining the Fourier transform of the image, the user defined cutoff frequencies region is filtered, and finally a new image is generated by using the inverse Fourier transform.

In the Fourier transform, the center region represents high energy, which is the majority of the information in the image. The four corner regions represent low energy, they usually store the edge information of the image, sometimes representing the key information of the image.

Therefore, comparing Figure 4.1 with Figure 1.2, which only has low-pass filter, the Figure 4.1 is like adding some partial edge information to the blurred image. Similarly, comparing Figure 4.2 with Figure 2.1, which only has high-pass filter, the Figure 4.1 is like adding blurred information to the edge signal of the image.

4.5. Image Sharpening

a. Butterworth High-pass Filter

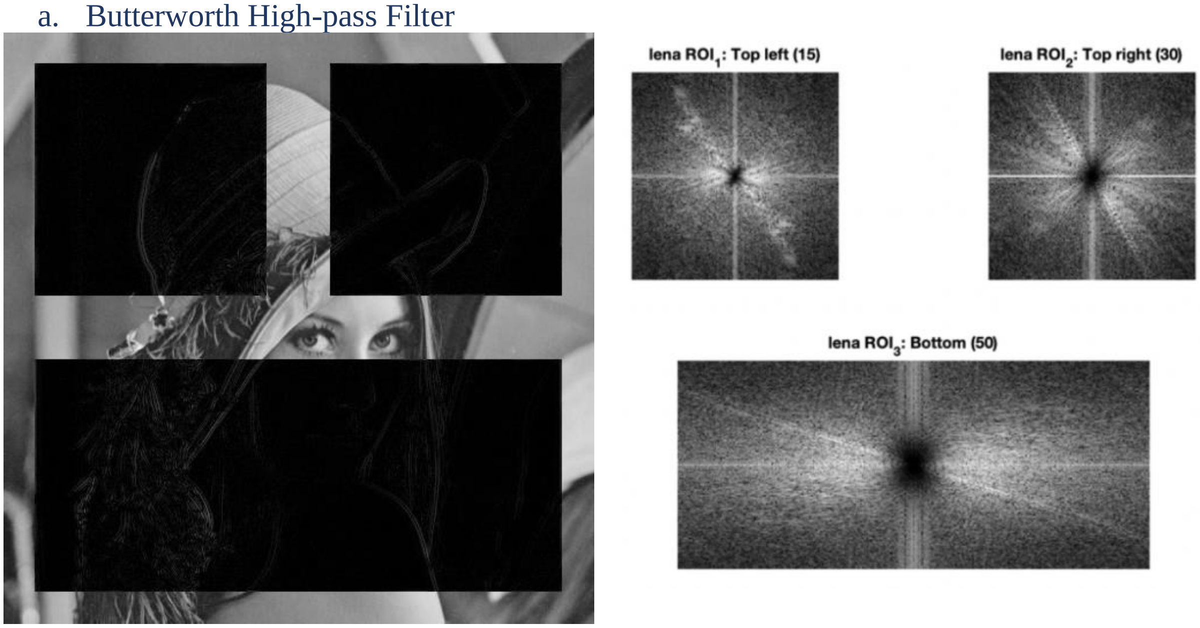

Figure 5a.1 Butterworth high-pass filtering in frequency domain
ROI_1 cutoff frequency: 15
ROI_2 cutoff frequency: 30
ROI_3 cutoff frequency: 50

Figure 5a.2 Combination of Butterworth high-pass filtering and original image

Figure 5a.1 shows the three ROIs we selected in the experiment with Butterworth high-pass filtering in frequency domain and the new Fourier transform for each ROI. This is implemented by the formulas:

$$G(u,v) = H(u,v) * F(u,v)$$

$$H(u,v) = \frac{1}{1 + [D_0/D(u,v)]^{2n}}$$

$$D(u,v)\ calculation\ formula\ is\ the\ same\ as\ in\ the\ previous\ experiment$$

Parameter D_0 is the user defined cutoff frequency. And n is the order of the Butterworth high pass filter. In this experiment, n is equal to 2.

Figure 5a.1 screens out the high frequency signal in the original image, that is, the edge information of the image. Comparing the Fourier transform of Figure 5a.1 with Figure 2.1, which is the ideal high pass filter, butterworth also creates a hole in the center of the frequency domain, but it is not a regular circle.

Figure 5a.2 shows the combination of Butterworth high-pass filtering and original image. We can notice, the image does enhance the edges and details and has a sharpening effect. This is implemented by the formula:

$$G(u,v) = H(u,v) * F(u,v) + F(u,v)$$

b. Gaussian High-pass Filter

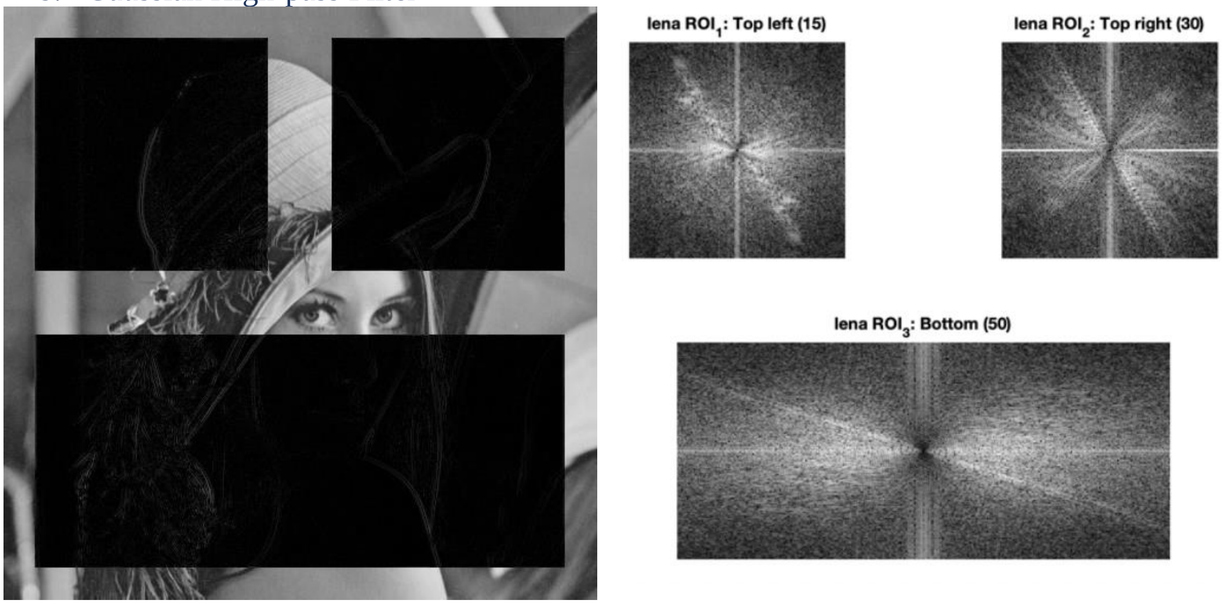

Figure 5b.1 Gaussian high-pass filtering in frequency domain
ROI_1 cutoff frequency: 15
ROI_2 cutoff frequency: 30
ROI_3 cutoff frequency: 50

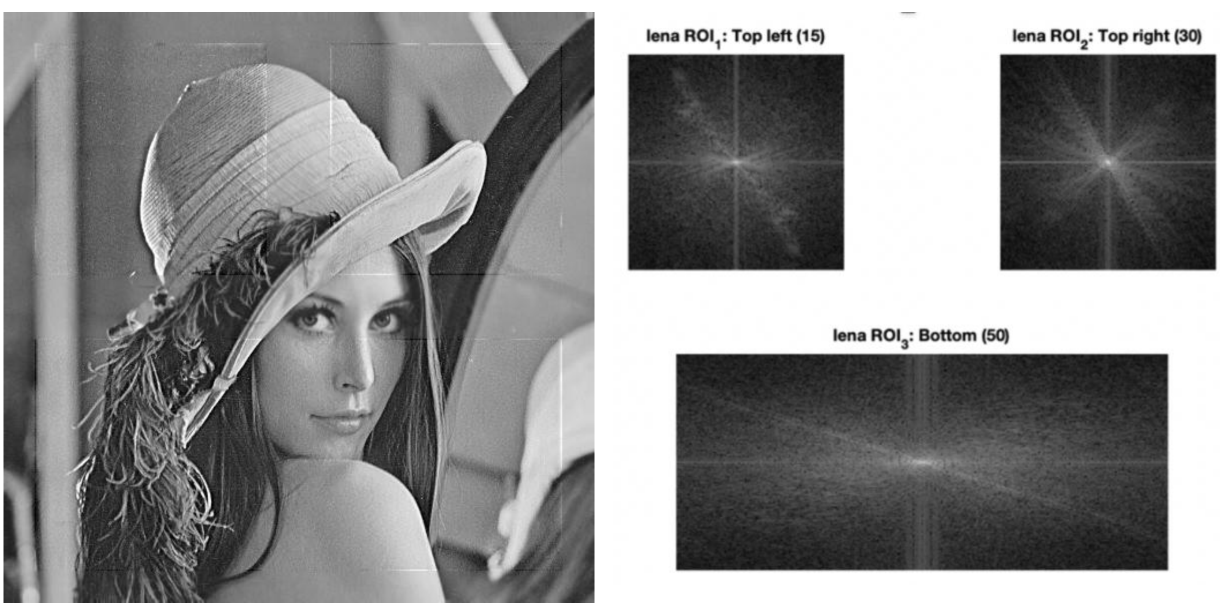

Figure 5b.2 Combination of Gaussian high-pass filtering and original image

Figure 5b.1 shows the three ROIs we selected in the experiment with Gaussian high-pass filtering in frequency domain and the new Fourier transform for each ROI. This is implemented by the formulas:

$$G(u,v) = H(u,v) * F(u,v)$$

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

$$D(u,v)\ calculation\ formula\ is\ the\ same\ as\ in\ the\ previous\ experiment$$

Parameter D_0 is the user defined cutoff frequency.

Figure 5b.1 also successfully screens out the high frequency signal in the original image, that is, the edge information of the image. Comparing the Fourier transform of Figure 5b.1 with Figure 2.1 with ideal high pass filter and Figure 5a.1 with Butterworth high-pass filtering, gaussian high pass filter only creates a very small hole in the center of the frequency domain.

Figure 5b.2 shows the combination of Gaussian high-pass filtering and original image. We can notice, the image does enhance the edges and details and has a sharpening effect. This is also implemented by the formula below:

$$G(u,v) = H(u,v) * F(u,v) + F(u,v)$$

c. Laplacian in frequency domain

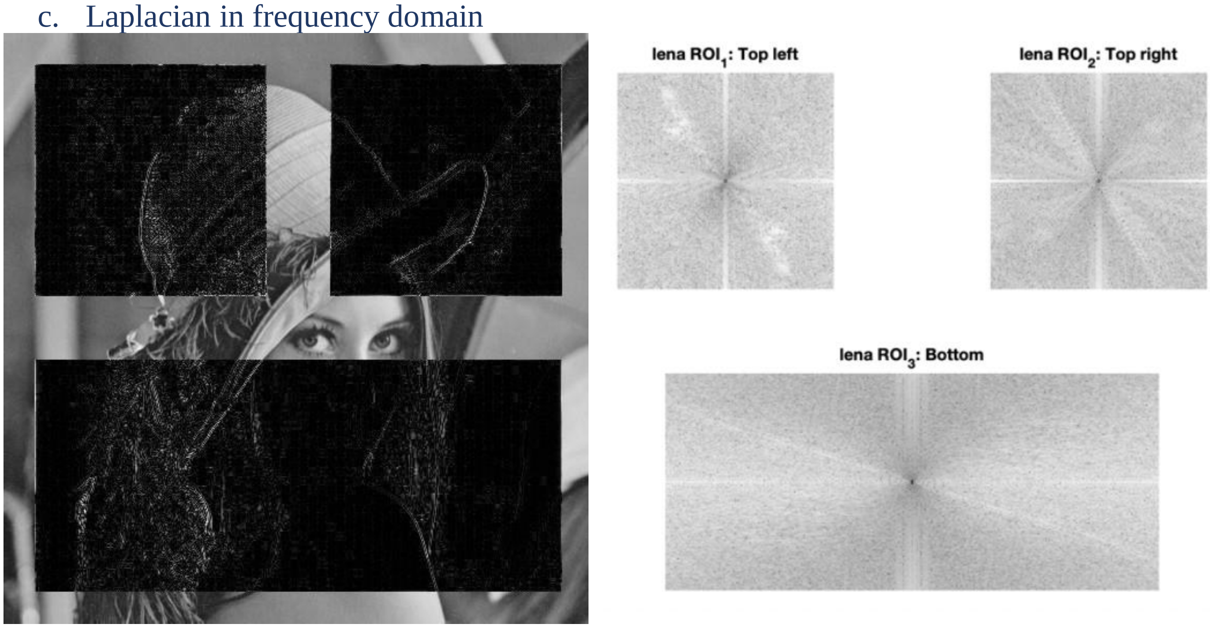

Figure 5c.1 Laplacian in frequency domain

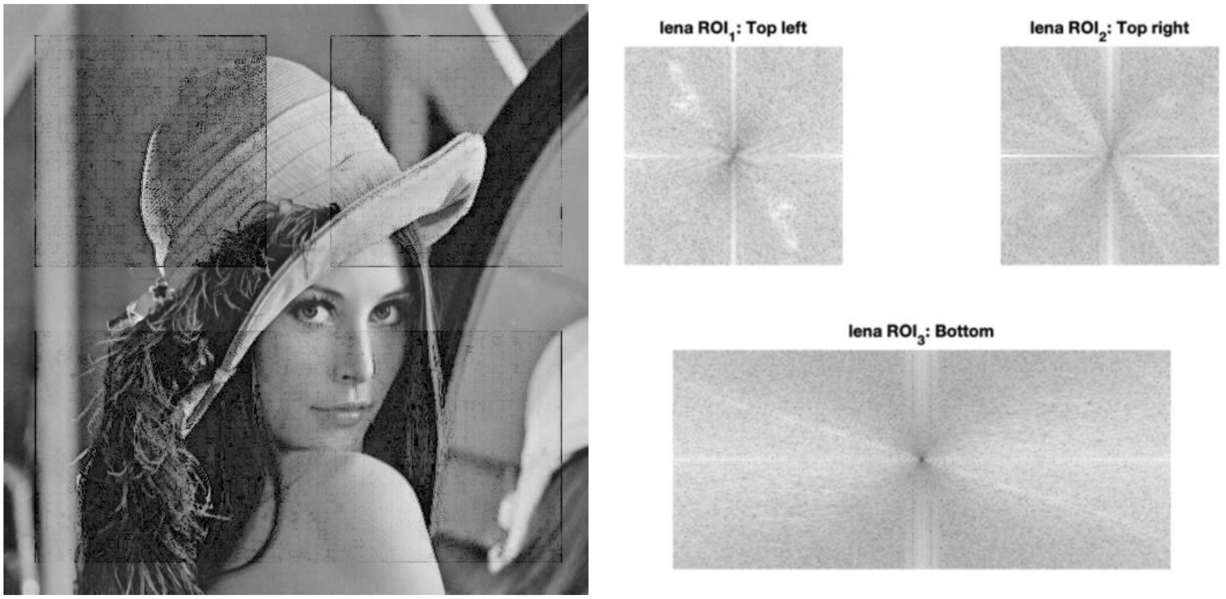

Figure 5c.2 Combination of Laplacian in frequency domain and original image

Figure 5c.1 shows the three ROIs we selected in the experiment with Gaussian high-pass filtering in frequency domain and the new Fourier transform for each ROI. This is implemented by the formulas:

$$G(u,v) = H(u,v) * F(u,v)$$

$$H(u,v) = 1 - \left[\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right]$$

Figure 5c.1 successfully screens out the high frequency signal in the original image, that is, the edge information of the image. And it shows more clearly than ideal high pass filter, Butterworth high pass filter and Gaussian high pass filter.

The enhance image can be obtained by substracting the Laplacian from the original image. Therefore, the high-frequency signal energy is strengthened, and the low-frequency signal energy is weakened. Its frequency domain map has obvious high energy.

Figure 5c.2 shows the combination of Gaussian high-pass filtering and original image. We can notice, the image does enhance the edges and details and has a sharpening effect.

5. Conclusion

According to the experiment 4.1, we can notice the low-pass filtering in the frequency domain is similar to the smoothing filtering in the spatial domain. They all have the effect of eliminating noise and blurring the image. But the ideal low-pass filtering in the Figure 1.2 is not as smooth as adaptive smooth in Figure 1.3.

According to the experiment 4.2, we can notice that high-pass filtering in the frequency domain is similar to the edge detection in the spatial domain. Because in the frequency domain, low frequencies are responsible for the general gray-level appearance of an image over smooth area, while high frequencies are responsible for detail, such as edges and noise. However, comparing with Figure 2.2 with sobel edge detection, Figure 2.1 with ideal high-pass filtering does not detect significant edge information and is not suitable for human eyes reading. Butterworth high-pass filtering, Gaussian high-pass filtering and Laplacian transform in experiment 4.5 all have similar shortcomings in edge detection. But a new sharpened image can be obtained by combining the new frequency domain with the original frequency domain.

In this experiment, the focus is on the transformation and correspondence between the frequency domain and the spatial domain of the image. Some task that is impossible to be implemented in the spatial domain can be done easily by operating on the image frequency domain. And frequency filtering is more efficient for big filter mask, spatial filtering is more efficient for small filter mask.