CMPEN 455: Digital Image Processing

Project 3 - Filtering Using Frequency Domain Analysis

Jonathan Lausch, Qiong Li, Zekai Liu

10/25/2019

A. Objectives

Part one of the project was applying an ideal Low Pass Filter (LPF) to an image, graphing the resulting image in the frequency domain, and also viewing the result in space. The goal of part two of the project was to design a notch filter that could correct an image with modulation introduced.

B. Methods

To run the code for this project, please run 'mainproj3.m' with the included 'notch.m' file.

Question 1a was done by setting the direct current to zero. To find this value, we needed to have the image's frequency components. To get this, we used the fft2() function, and as shown in class, the DC offset is at the origin of the image. However, in MATLAB, the position of the DC offset is (1,1). We set this value to zero to remove the offset. The resulting image is shown in Figure 2.

Question 1b (i): In order to have a centered frequency domain image, we modulated g by $(-1)^{(x+y)}$ and then took the Fourier transform of that, which is plotted in the mesh seen in Figure 5.

Question 1b (ii) was done by using the provided lpfilter function to filter the checkerboard image. To use this filter, we took the Fourier transform of the image, and then multiplied the image in the frequency domain by the filter, since this is the equivalent of convolution in the space domain. The result is shown in Figure 3.

Question 1c: We used zero padding to compute a more accurate filtered image (Figure 1 shows the zero-padded original). In this case, the zero-padded image was 512x512 (Figure 1). Before we filter the image, we also needed to zero-pad the filter. Similar to the previous questions, we multiplied the image's frequency components and the filter together since this equates to convolution in the space domain. The result is seen in Figure 6.

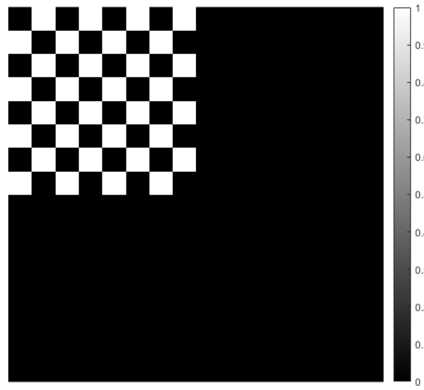

Figure 1: Zero-padded image.

Question 1d is discussed in the Results section below.

Question 2a: The filter designed to remove the modulation of the original image can be found in the ‘notch.m’ file. It is an array of one’s, with positions (1,33) and (1,481) set to zero. The postions set to zero represent the function that the image was modulated by. These positions were found by using the method in Lecture notes L15-17 to L15-20. While the calculations are very similar, the main difference is the coefficient inside of the cosine function being 32. To find the two positions set to zero, the first is just the mentioned coefficient, and the second is the size of one dimension of the image minus this coefficient. In Matlab, the origin is at (1,1), which is why the found numbers have 1 added to them in the file. The filter can be seen in Figure 12. It appears mostly white, since only two values were set two zero – these show up as tiny dots to the left and right of the origin of the image.

Question 2b: Using the notch filter discussed above, we were able to remove the majority of the modulation as seen in Figure 13. The filter was designed in the frequency domain so a simple multiplication operation could be performed with the Fourier Transform of the modulated image. In essence, Figure 11 was multiplied with Figure 12, which resulted in Figure 14. The inverse Fourier Transform of Figure 14 returned Figure 13, which is almost identical to the original image. Some of the modulating lines can still be seen, but are hardly noticeable. *Note: all Fourier Transforms shown are the magnitude of the centered, log-scaled version created with an identical process as Question 1b above.*

Question 2c is discussed in the Results section below.

C. Results

The result of Question1_a is below. This is the ‘checkerboard’ image with the DC offset removed. Compared to the original image, the white portions are now gray, and the darkest pixels remain unchanged. Mathematically, this makes sense because any pixels greater than zero in the original must be that value minus the DC offset, so it is impossible for it to look the same if the DC offset is greater than 0.

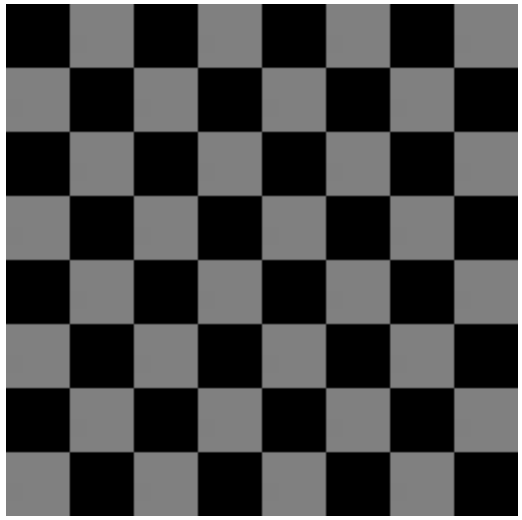

Figure 2 – Image without DC offset.

Figure 3 below is the non-zero padded image with an ideal low pass filter applied. This filter appears to blur the image by removing any “sharp edge" content.

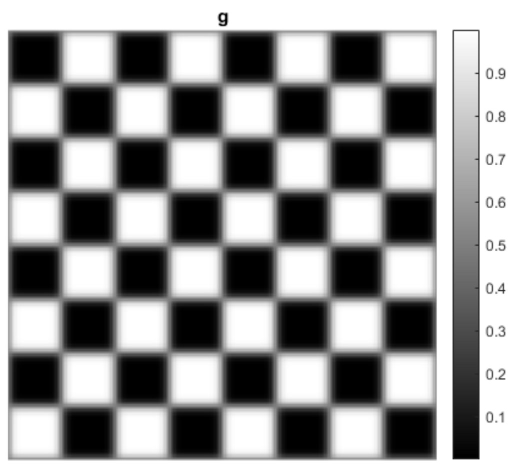

Figure 3 – Non-zero padded image with LPF applied.

(g(x,y) in section 1(b)-ii)

The image below is the Fourier transform of Figure 3. Normally, since a LPF was applied, the data would be in each corner of the graph, but this was centered by multiplying the inverse Fourier Transform of the filter by (-1)^(x+y), to have the uniform display below.

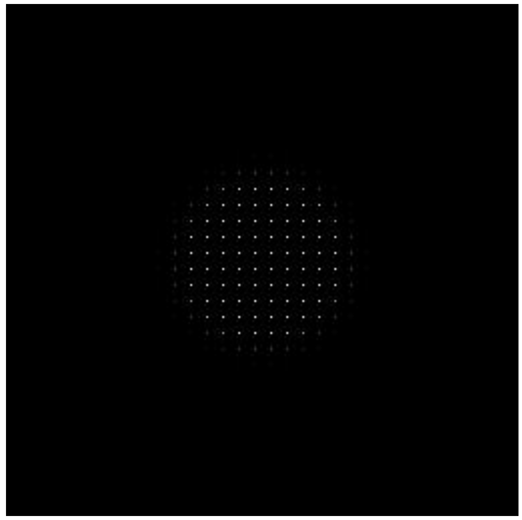

Figure 4 – Fourier Transform of non-zero-padded filtered image.

(|G(x,y)| in section 1(b)-ii)

The image below is a 3D visualization of the low pass filter we used for question 1. Again, this is the centered version of the filter.

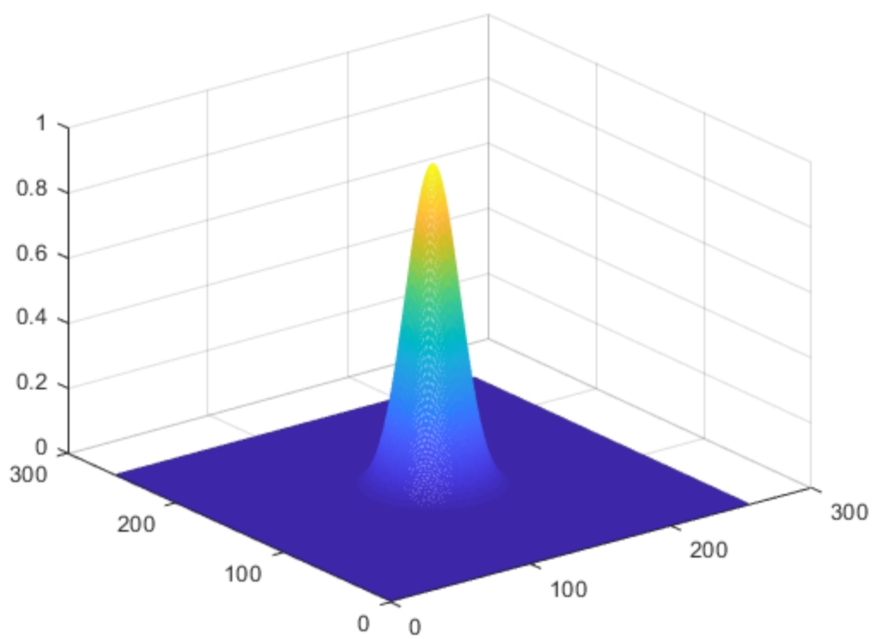

Figure 5 – Mesh of LPF used.

Seen below is the resulting image from a filter applied with zero-padding. The most noticeable difference from Figure 3 is the deep shade of black the pixels retain from the image, unlike Figure 3.

Figure 6 – Zero-padded filtered image.

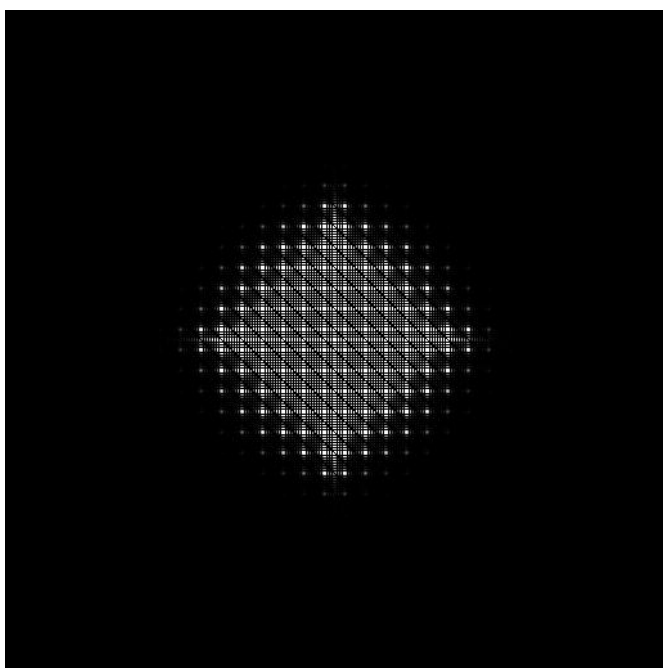

Figure 7 – Fourier Transform of zero-padded filtered image.

Answer to question 1d: The filter still blurs the image, but there is no observable wraparound error because of the zero-padding. In Figure 7, much more of the low-frequency content than Figure 4, where no zero-padding was used. The modulation operation centers the content in the Frequency domain. While the zero-padding retains more of the qualities of the original image, it is double the size, which leads to a less-efficient computation. This is a tradeoff – a 'better' result will require a longer computation, with the possibility of having to resize the image after the operation.

The results from Question 2 are below, beginning with Figure 8 – the original image. Referred to as f(x,y).

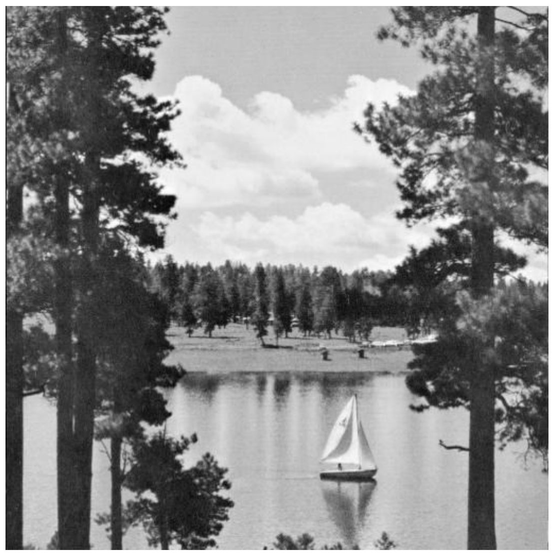

Figure 8 – Original image.

The image below is the Fourier transform of Figure 8, or |F(u,v)| in Question 2. This shows the frequency content of the image before modulation.

Figure 9 – Magnitude of Fourier Transform of Figure 9.

The next image below is referred to as c(x,y) in Question 2, and is the original image modulated by a cosine function, listed in the description.

Figure 10 – Figure 8 modulated by 32cos(2*pi*32y/N).

Figure 11 is the Fourier transform of Figure 10. This is very similar to that of the original image in Figure 9, but appears slightly darker than the original Fourier Transform. Referred to in Question 2 as |C(u,v)|.

Figure 11 – Magnitude of Fourier Transform of modulated image.

The image below is the notch filter used to remove the cosine function of the original image. The black lines around the edge were added specifically for this document to show exactly where the image content is, as it is all ones except for two very small dots to the left and right of the origin. Referred to as |H(u,v)|.

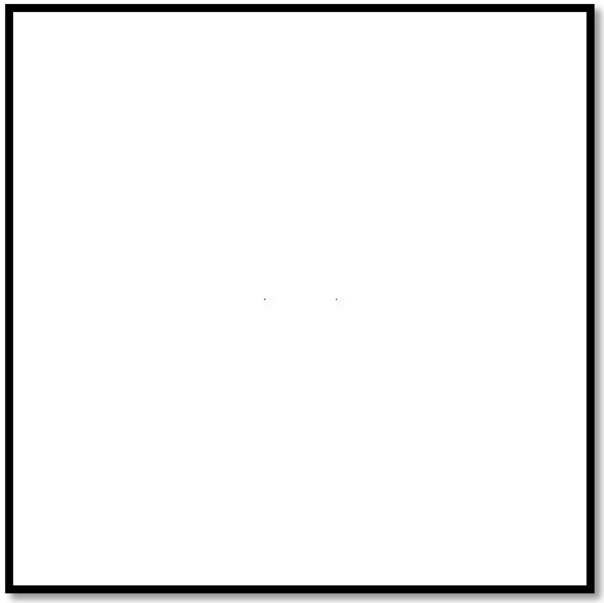

Figure 12 – Magnitude of notch filter used (inside frame).

Figure 13 is the result of applying the notch filter to the modulated image (Figure 10). Some modulating lines are still visible in this corrected image. Referred to as g(x,y).

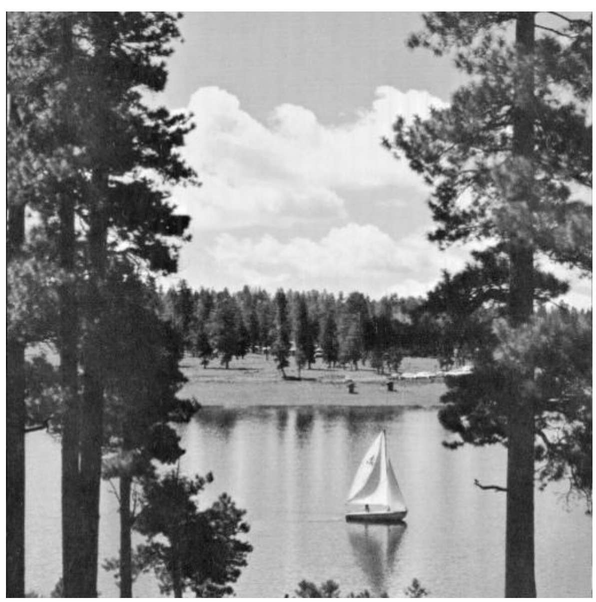

Figure 13 – Corrected image.

The Fourier transform of the corrected image. While almost identical to the original, when placed side by side there is a very slight difference in the shade of gray. Referred to as |G(u,v)|.

Figure 14 – Fourier Transform of corrected image (Figure 12).

Figure 15 is the difference of the original image and the corrected image. Notably, this is not zero, so the modulation was not completely removed with our notch filter.

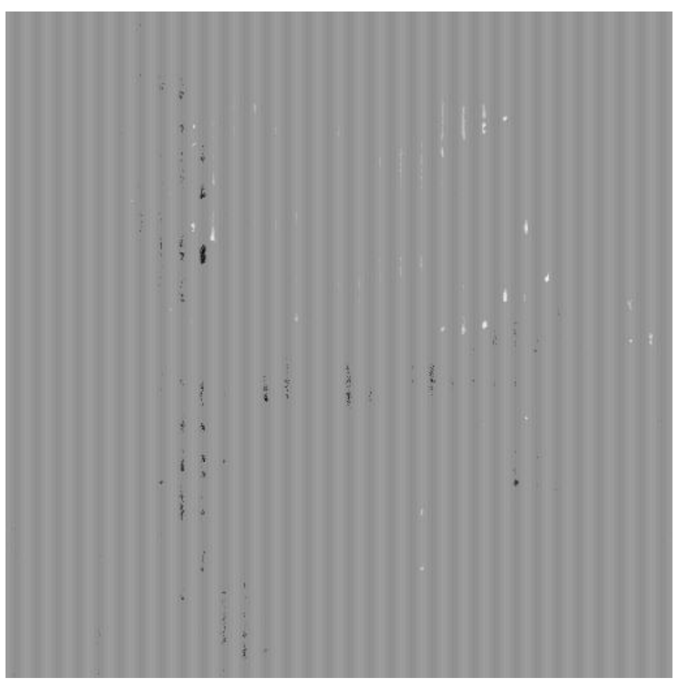

Figure 15 – Difference of original image and corrected image.

Figure 16 – Difference of original and corrected Fourier Transforms.

Question 2c: It is not possible to completely recover f from c due to the periodic nature of the function the image was modulated by. It most likely affects all the frequencies in the image; however, the corrected image is strikingly similar to the original, and is of decent quality.

D. Conclusion

The biggest outcome from Question 1 is that zero-padding an image that is about to be filtered will yield higher quality results. With this method, more low frequency content will be retained since there is little to no wraparound error. From Question 2, it was realized that a corrupted image cannot be fully recovered by filtering out the specific frequencies it was modulated by. However, the process to remove those frequencies does result in an image that is almost indistinguishable from the original.