Programming Assignment 4

4
CS 474

Steven Hernandez and Adam Lychuk

Date due: 12/09/2020
Date handed in: 12/09/2020

Division of work:
Steven Hernandez wrote the theory, implementation and results and discussions for experiment 1a and experiment 1b. He also implemented the sobel frequency mask with the lenna image. Adam Lychuk wrote the theory of experiment 2 and 3 and the implementation of the sobel spatial mask along with experiment 3. Results and discussion for experiment 3 was made by Adam.

Theory

Experiment 1

Band-reject filters are used in order to remove periodic noise which can arise from interferences such as electrical or electromechanical interferences during an image acquisition. These reject bands can be made by using either an ideal, butterworth or a gaussian filter as shown below in Table 1. The main idea of this method is that the band will help reject whatever frequencies happen to be within it. Therefore, the undesired frequencies within it will not appear on the image. In order to better visualize the periodic noise in an image, it is best to look at it in the frequency domain where the spectrum can clearly show the impulses caused by the interferences. Depending on the location of these impulses, the band may be adjusted to be wider to cover the errors. The rest of the desired spectrum is left mostly intact. While these band reject methods help eliminate the noise within an image, the opposite can also be done with the help of a band-pass filter. This can be done by subtracting the values for the band reject from 1 as seen in Figure 1. Hence, a band-pass filter will be made where only the frequencies from the interferences are accepted, but rejects all other values. The results are the isolated specific frequencies in the image rather than the actual picture.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

Table 1: Ideal, Butterworth and Gaussian frequency domain equations.

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

Figure 1: operation of the band pass filter

Experiment 2

Spatial filters can be modified for multiplication in the frequency domain, providing equivalent results to convolution in the spatial domain as well as often providing a computational speed up. In addition if we know our spatial filter exhibits odd or even symmetry we can save even more computing power as well as use theory to improve our overall results. Take for example the sobel filter in Figure 2 a. If we pad with zeros as seen in Figure 2 b, the sobel filter exhibits odd symmetry as defined below in Figure 2 c. When padding with zeros we can vary the size of the filter to match the image we are convolving with because odd symmetry is maintained when an even number of zeros is added. In other words if we insert our even dimensioned filter exhibiting odd symmetry into another even dimensioned array we will maintain that odd symmetry as shown in Figure 3.

a. $\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$ b. $\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & -2 & 0 & 2 \\ 0 & -1 & 0 & 1 \end{matrix}$

c. $h(x,y) = -h(N-x, M-y)$

Figure 2: a. Sobel Filter b. Sobel Filter Padded c. Definition of odd symmetry.

This is important as we know that a function with odd symmetry is real and odd in the spatial domain and therefore imaginary and odd in frequency domain. Therefore, we do not need to calculate the real values in the frequency domain during our Fourier transform, saving computational cost, or if they are calculated anyways, recognize computational errors from the discrete nature of the calculations. This holds in the even case as well. If a filter exhibited even symmetry we could instead ignore the imaginary values. These are powerful debugging tools in the discrete case as well since for example, the spectrum of the odd filter will be visually asymmetric.

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -2 & 0 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Figure 3: Padded sobel filter inserted in an even dimensioned array.

Experiment 3

Homomorphic filtering can improve the overall visual fidelity of an image that has sensor errors in illumination and reflectivity capture. Reflectance values in images often vary abruptly at the junction between objects, and illumination values vary slowly over the rest of the image. As we know high frequency values are edges in images so emphasizing them increases reflectance and de-emphasizing low frequencies reduces the washing out of an overly illuminated image. If we model illumination as i and reflectance as r, we get i multiplied by r in the spatial domain which is i convolved with r in the frequency domain as shown below Figure 4 a. In this case illumination and reflectance would be intrinsically linked due to their convolution in the frequency domain. However if we instead take the log of f(x,y) in the spatial domain we can separate illumination and reflectance in the frequency domain as shown in Figure 4 b. Now separated we can multiply in the frequency domain H(u,v) the filter as shown in Figure 4 c.

$$f(x, y) = i(x, y)\, r(x, y)$$

a.

b. $$F(ln(f(x,y)))=F(ln(i(x,y)))+F(ln(r(x,y)))$$

c. $$Z(u,v)H(u,v) = Illum(u,v)H(u,v) + Refl(u,v)H(u,v)$$

Figure 4: a. Function describing illumination and reflectance in the spatial domain b. The Fourier of the log of f(x,y) c. Filter transfer function in the frequency domain.

However since our result was initially separated by taking the log in the spatial domain, after taking the inverse dft we must take the exponential of the result to reverse the logarithm as shown in Figure 4 a and b resulting finally in our g(x,y) Figure 5 c.

a. $$s(x,y) = i'(x,y) + r'(x,y)$$ b. $$e^{s(x,y)} = e^{i'(x,y)}e^{r'(x,y)}$$

c. $$g(x,y) = i_0(x,y)r_0(x,y)$$

Figure 5: a. Inverse Fourier Transform of filtered illumination and reflectance components b. Exponential of result c. Final homomorphically filtered image.

As mentioned above H(x) needs to reduce illumination and increase reflectance. By simultaneously squeezing the intensity range and sharpening the image we de-emphasize lower frequency values associated with illumination, and emphasize higher frequency values associated with reflectance. The Gaussian High Pass Filter (GHPF), shown in Figure 6, approximates the aforementioned transformation well where y_h and y_L linearly scale the GHPF to fit a specific frequency range, c controls sharpness, and D_0 describes the range of frequency values in the center of the function to discard. This entire process can be described succinctly in the chart in Figure 7.

$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c\left[(u^2+v^2)/D_0^2\right]}\right] + \gamma_L$$

Figure 6: Gaussian High Pass Filter function

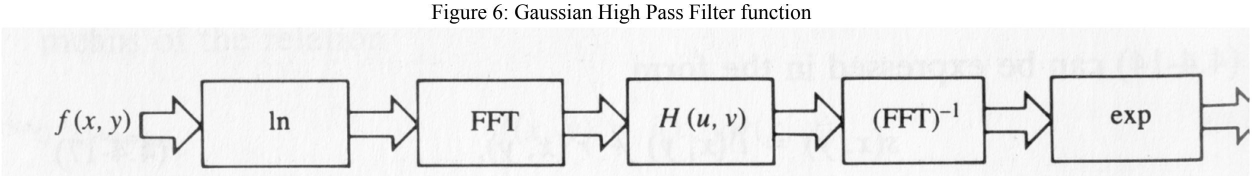

Figure 7: Steps for homomorphic filtering

Implementation

Experiment 1

This part of the experiment was split into smaller separate programs to help smooth out the process of filtering the periodic noise. The first step was to pad the original image which has the additive noise with zeros so the image can be left in the upper left corner. Once the image has been padded and multiplied by $(-1)^{x+y}$ to center the spectrum, it is then computed with the 2D FFT function. The next step was to make the mask H(u,v) of size PxQ which was made using the butterworth equation seen in Table 1; therefore, there was no need to worry about padding, centering and processing with DFT as it was implemented directly into the frequency domain instead of the spatial. Once F(u,v) and H(u,v) were made, the product G(u,v) was computed using element-wise complex multiplication. The inverse 2DFFT was called when the computation was complete and instead of using both the real and imaginary parts of the image, only the real values were used in the image after undoing the centering $(-1)^{x+y}$. The image is still padded at this point; hence, the image is then extracted and placed into an image that has the respective measurements MxN. The second part of the experiment is to extract the image and leave the noise. In order to do this, a band-pass mask was used by subtracting the mask values from 1 and then computing the element-wise complex multiplication.

Experiment 2

The process of experiment 2 is similar to the one of experiment 1; however, the main issue was making the sobel mask since it required more manipulation in order to get it to work correctly since it was being made first in the spatial domain. The original image goes through the same process of being padded, centered and then transformed. The sobel mask on the other hand first gets a leading row and column of zeros. Next it gets padded so it is size PxQ, but the sobel mask must be on the center. It is then multiplied by $(-1)^{x+y}$ to center the spectrum and transformed with the 2D DFT. The real values should be 0, but just in case they are not, they are set equal to zero and the imaginary values are uncentered. Once this is done, the rest of the process is done like experiment 1.

Experiment 3

In implementing experiment 3 we first created the high-frequency emphasis filter which is the Gaussian high pass filter described in the theory section. We implement this as a function where we can input D0, c, yl, and yh. This function returns the (GHPF) filter with the inputted parameters. We then pad the image as in experiment 1 and take the natural log of the result in the spatial domain. Next we compute the forward 2d fft on the result. To apply the filter to the image we do element wise complex multiplication in the frequency domain and then call the inverse 2d fft on the result. Then as described in the theory section we apply the exponential function to both imaginary and real parts. We then take the magnitude of the result and apply min max normalization to get the final image.

Results and Discussion

Experiment 1

Part A

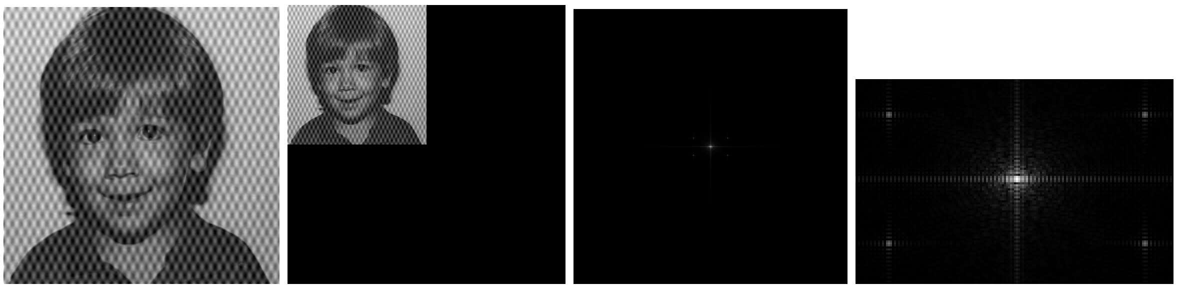

Figure 8: (a) boy_noisy.pgm (b) padded PxQ boy_noisy (c) The spectrum of the padded image (d) closer view of the spectrum

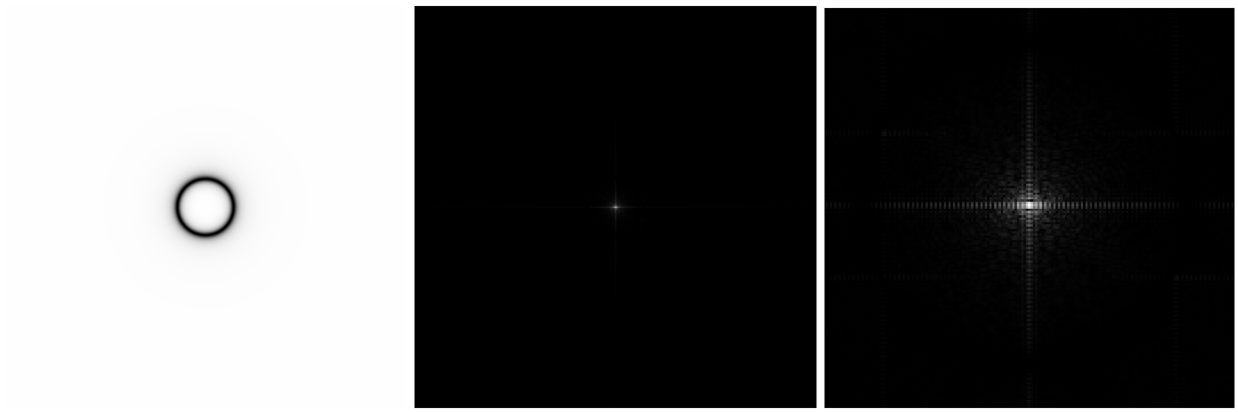

Figure 9: (a) Band-Reject Butterworth (b) Spectrum after filter is applied (c) closer view of the spectrum

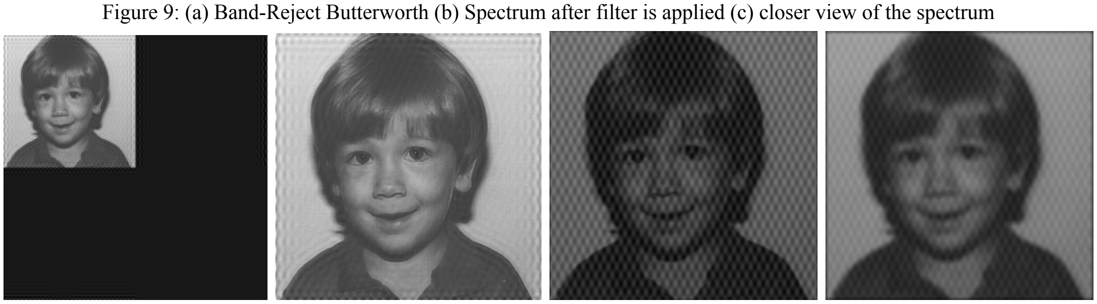

Figure 10: (a) Inverse 2D (b) boy_noisy with removed padding (c) spatial Gaussian 7x7 (d) spatial Gaussian 15x15

The experiment had us start with an image with a cosine noise which was then padded and transformed into the spectrum seen in Figure 8(c). At first, there seems like there is not much to see, but upon closer inspection the impulses from the cosine noise can clearly be seen surrounding the spectrum of the image, Figure 8(d). After going back to the chapter discussing this situation, it was decided that a band reject filter would be the best option to go with. In this case, the butterworth filter was used. The butterworth filter has a couple variables where the user can choose in order to adjust the size of the band. After multiple trials, D_0 set at 71, W set at 15 and n at 1 gave the optimal filter seen in Figure 9(a) and it successfully covered most of the spectrum given from the impulses. Figure 9(c) shows that the impulses are no longer there; however, while it is hard to see, the filter also affected the spectrum as some of the spectrum seems to be missing some values. After the image has gone through the inverse 2D DFT and has its padding removed, the result shows that most of the noise has been successfully removed; nevertheless, there is still some noticeable noise near the edges of the image, Figure 10(b), which I believe are the remnants of the impulses that were further away from their centers and so the band did not fully covered the whole impulses. But if the band was wider, then it could have probably affected the spectrum of the image even more. Comparing Figure 10(b) with Figure 10(c) and 10(d), the difference between them is huge. The noise in Figure 10(c) is still very noticeable and Figure 10(d) seems to have done a bit better than its counterpart, but noise is still clearly visible.

Part B

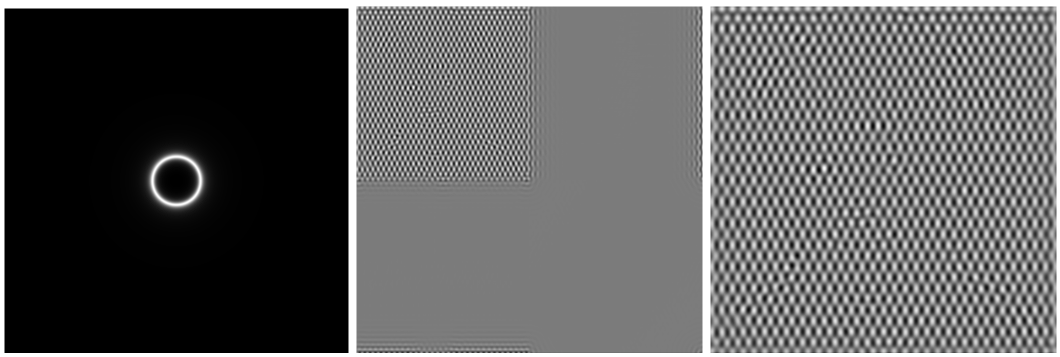

Figure 11: (a) band-pass filter (b) padded noise (c) Noise extracted from boy_noisy

The second part of the experiment had us do the opposite of part A where we removed the noise. In this case the picture was removed and the result was the noise being shown in the image. This was done by using a band-pass filter as seen in Figure 9(a) which was made by subtracting the values of the mask seen in Figure 11(a) from 1. This gave us the opposite of what band-reject did and here the impulses made from the noise were saved while the spectrum of the child was eliminated as seen in Figure 11(c).

Part C

In both face verification and face recognition, models rely on high quality datasets of face images. A machine learning model will be feature engineered to, in the case of face verification, recognize if that person is who they say they are within some level of error. Or, in the case of face recognition, match directly to a face in the data set. The features the model tracks vary between face recognition and face verification, and among machine learning methodologies such as eigenfaces or convolutional neural networks. Often however, in all cases, those features are minor perturbations in the face. Color of eyes, size of ears, shape of nose, etc. These features are mathematically described and as such noise can significantly change the values of the image to make verification or recognition impossible. If the noisy images were used in the training dataset, the model might even learn the noise pattern reducing accuracy as well as making matches without the noise virtually impossible. Algorithmic errors can also cause similar problems. If all the images are low contrast for example, the model may find it difficult to find distinct features. Performing histogram equalization, or sharpening the images, could result in a better training set with the caveat that the images at inference should be similarly modified.

Experiment 2

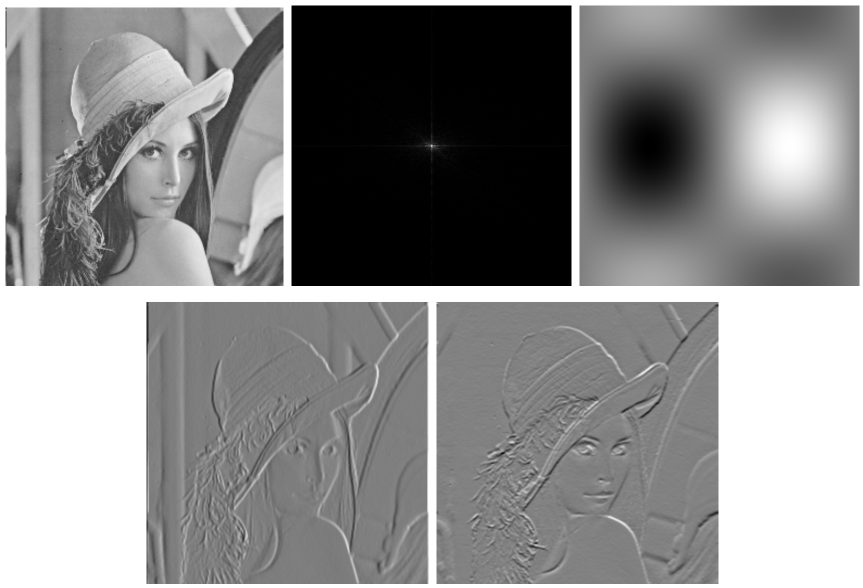

Figure 12: (a) lenna f(x,y) (b) Padded lenna F(u,v) (c) sobel H(u,v) (d) lenna g(x,y) (e) lenna spatial

The experiment had us transfer the sobel filter to the frequency domain and then compare the resulting multiplication in the frequency domain to the convolution in spatial domain. The lenna image and its centered spectrum are shown in Figure 12 a and b. Figure 12 c is the resulting spectrum of the sobel mask. It is noticeably symmetric due to its odd symmetry confirming we correctly transferred the spatial filter to the frequency domain. Figure 12 d is the lenna image with the sobel mask applied in the frequency domain and Figure 12 e is the sobel mask applied in the spatial domain. Slight differences in contrast are shown, but the two results are almost identical emphasizing edges in the x direction. This is the expected result due to the equivalence of convolution in the spatial domain to multiplication in the frequency domain. These slight differences are most likely minor calculator errors in either the spatial domain or frequency domain.

Experiment 3

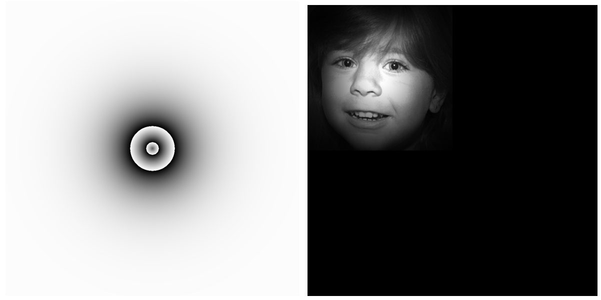

Figure 13: (a) the homomorphic (b) padded girl image

This experiment had us create a homomorphic filter in an attempt to improve the girl image. We were able to successfully create the homomorphic filter as shown in Figure 13 a, but applying it to the padded image resulted in a black image. During our steps we took the log of the girl image and took the forward fft, and created the homomorphic filter as shown above directly in the frequency domain. We believe the issue arises in the multiplication of the filter and fft of the girl image, however it may arise much later in the implementation of steps.