CS 474 Image Processing and Interpretation

Project 4

Filtering in the Frequency Domain

Dennis Brown

University of Nevada Reno

December 11, 2024

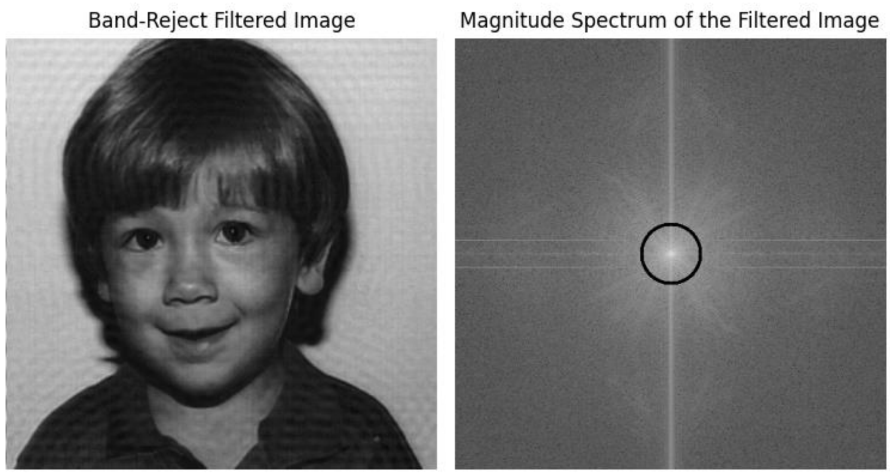
Experiment 1

(1.a)

The noise in the image shown in Figure 1(a). Was generated by adding cosine noise to the original image. The first filter we will apply is a band-reject filter. Band-reject filters are effective for removing periodic noise, meaning we would expect good results after applying the filtering to the noisy image. However, band-reject filters may also remove frequencies that are not related to the noise. We will apply a notch filter later that can help solve this problem. Band-reject filters are applied in the frequency domain; thus, we must first convert the image from its spatial domain to the frequency domain using the Fourier transform. When creating the reject band mask, we characterize it by its radius - the center frequency which the band is defined, and its width – the range of frequencies to attenuate around the radius. Values within the bad are set to zero or attenuated and values outside are left unchanged. In Figure 1(b). We can see the band filter placed over the noisy image. Figure 1(c) is the spectrum of the noisy image. Looking at the spectrum of the noisy image we can see faint spikes away from the origin seen within the red circles, these spikes correspond to periodic noise. When adjusting the parameters of the band-reject filter we should fully include these spikes to remove the periodic noise.

Figure 1(a)(b)(b). (a) Image with noise generated by periodic cosine noise. (b) Band filter placed on top of the noisy image. (c) Spectrum of the noisy image.

After applying the band-reject filter we can see the results in Figure 2(a). Band-reject has done a good job at removing the periodic noise; however, it has also produced a slight ringing effect. We have a slight overshot of image features close to our cutoff frequency which will produce such an effect. We can see the spectrum of the filtered image in Figure 2(b), adjusting the band radius and width parameters, having r = 35 and w = 2, seems to produce the best results capturing the spikes and a small amount of image details.



Band-reject does a good job at removing periodic noise, however as stated it may remove frequencies not related to the noise. To solve this, we can use a notch-reject filter which will precisely target the spikes seen in the spectrum of the noisy image. Unlike band-reject which uses a ring to attenuate frequencies, notch-reject attenuates frequencies in pre-defined areas around the center of the frequency domain. A notch filter is symmetric about the origin, meaning the notch points must occur in symmetric pairs at (u_0, v_0) and $(-u_0, -v_0)$. Figure 2. Is our noisy image with the notch-reject filter applied on top. Seen within the red circles are the areas in which the filter will attenuate.

Figure 2. The noisy image with the notch filter applied on top, highlighted in the red circles.

After adjusting the filter parameters for the radius and position of the notchs. Having r = 2 and the positions placed directly ontop of the spikes, notch-reject produces excellent results. As seen in Figure 3(a). The filtered image has no ring effect since only the frequencies related to the noise are attenuated. Figure 3(b). Shows the spectrum of the filtered image with the notch filter applied over the spikes.

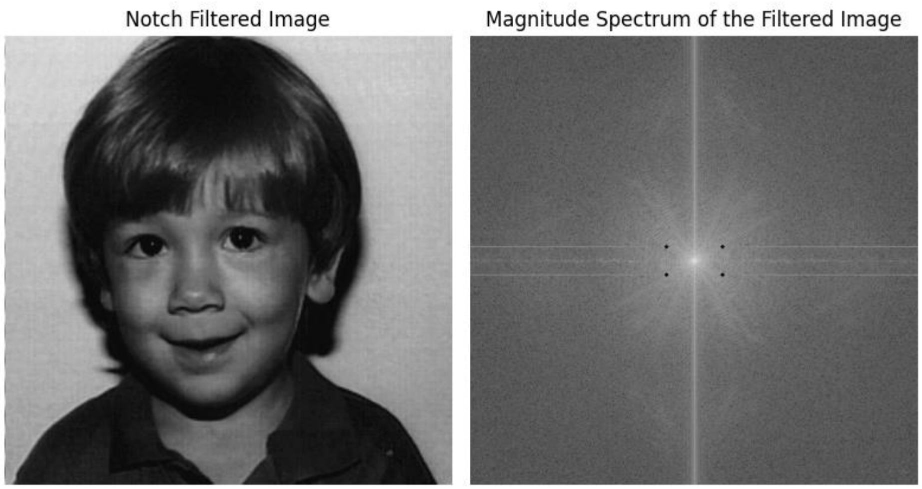


Figure 3(a)(b). (a) The image after applying the notch filter is excellent at removing noise. (b) The spectrum of the filtered image with notches placed at the spikes.

Comparing band-reject and notch-reject to gaussian filtering in the spatial domain with 7x7 and 15x15 sizes. We can see in Figure 4(b) and Figure 4(c) that this does not remove the noise but rather blurs the entire image.

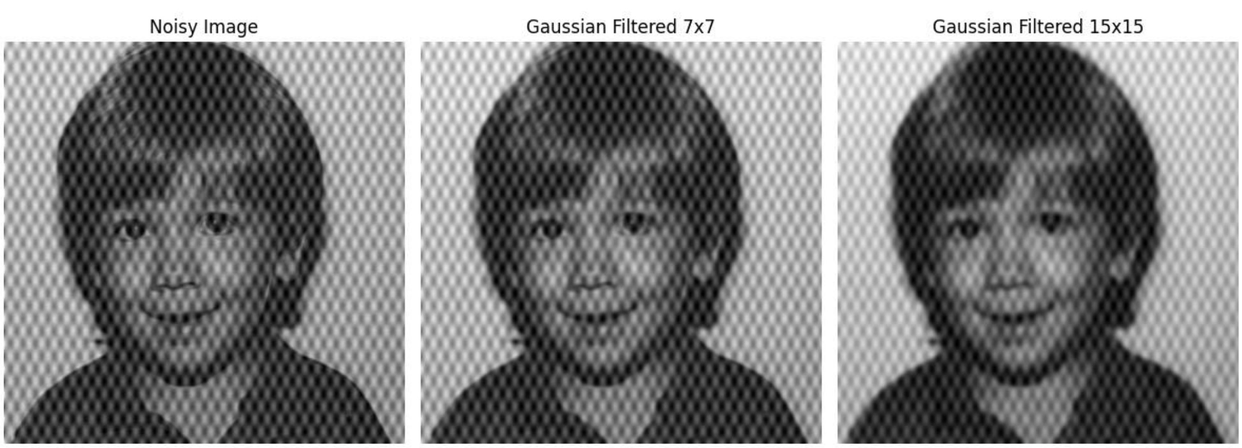


Figure 4(a)(b)(c). (a) The original noisy image. (b) Noisy image with a 7x7 gaussian filter applied smooths the entire image. (c) Noisy image with a 15x15 gaussian filter applied, applies even more smoothing to the image and does not remove any noise.

(1.b)

Instead of removing the noise, we will employ a procedure to extract the noise. This procedure involves using the notch filter but inverting the mask and then applying. This will extract the noise that was added to the image shown in Figure 5(a). To verify we will also find the magnitude spectrum of the extracted noise shown in Figure 5(b).

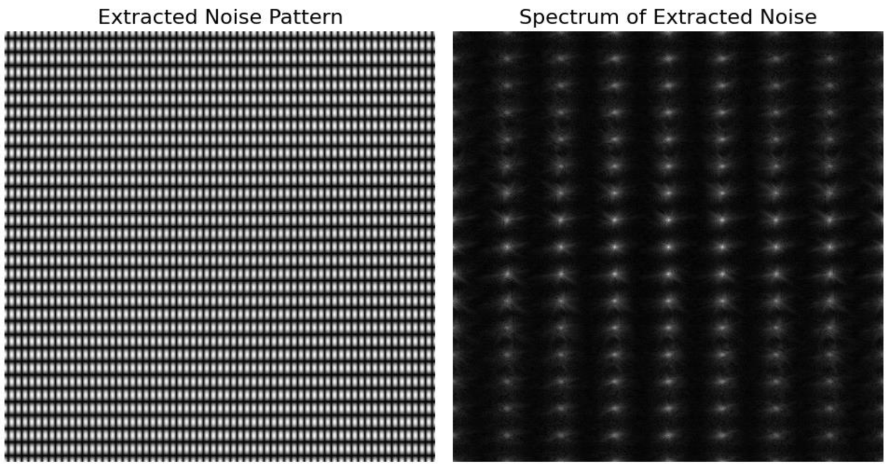


Figure 5(a)(b). (a) The extracted noise pattern resembles cosine noise. (b) Spectrum of the noise pattern verifies the noise was produced from adding cosine.

We can see a pattern in Figure 5(a) that resembles cosine waves giving us a clue that the noise was indeed generated from cosine waves. In Figure 5(b) we can see the spectrum of the noise pattern which verifies it was indeed generated from cosine waves. We can clearly see periodic spike bright spots corresponding to the cosine noise.

(1.c)

Facial recognition technology is a relatively new and controversial topic. Image processing plays a big role in ensuring that noise is removed from images and ensuring proper identification. Failure to perform facial recognition properly can lead to consequences such as law enforcement wrongly identifying a suspect to financial fraud. Such errors can further lead to erosion of public trust in such systems and even financial loss.

Experiment 2

In this experiment we will generate a filter transfer function in the frequency domain, this will tell us how a given filter modifies the frequency components (u, v). We will then compare the filtering results in both the frequency and spatial domains. We will begin with a 3x3 vertical Sobel Kernel $h(x, y)$ and an input image $f(x, y)$ with a size of 256 x 256. To avoid wrap around in the frequency domain both f and h must be padded with zeros to a size of 258 x 258. Currently our Sobel kernel has odd symmetry, however the first element is not 0, therefore we will pad the kernel with a leading row and column of 0's. After padding the kernel its size is 4 x 4, to have a kernel size of 256 x 256 we will embed the 4 x 4 kernel into a larger array. The embedded kernel must be placed in the center of the larger array. Once embedded the spectrum must be manually shifted using $h(x, y) \cdot (-1)^{x+y}$. After centering, the FFT $H(u, v)$ can be computed. When computing the FFT we set the real part to 0 since only the imaginary part is non-zero. Finally, we undo the centering we performed given by $H(u, v) \cdot (-1)^{u+v}$. Once the input image has been padded with 0's we will also center it using the equation above and compute its FFT $F(u, v)$. With both FFT's computed we can now apply the filter and compute $G(u, v) = F(u, v)H(u, v)$. With $G(u, v)$ calculated we can convert back to the spatial domain by computing the inverse FFT and only keep the real part. Once back in the spatial domain we must undo the last centering by applying $h(x, y) \cdot (-1)^{x+y}$ once more, finally we trim the padding on the final image.

The filter transfer function is displayed as an image in Figure 6. The input image is shown in Figure 7(a). The results of applying the Sobel filter in the frequency domain can be seen in Figure 7(b). Applying the Sobel filter in the spatial domain can be seen in Figure 7(c). As expected, both procedures produce near identical results. The spectrum of the input image is shown in Figure 7(d). The spectrum of the Sobel filter in the frequency domain is shown in Figure 7(d), and the spectrum of the Sobel filter in the spatial is shown in Figure 7(e).

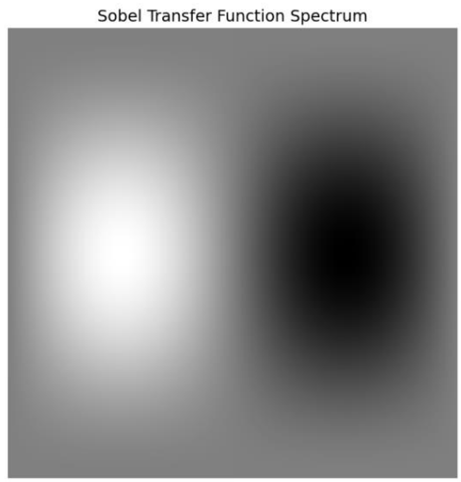


Figure 6. Filter transfer function image.

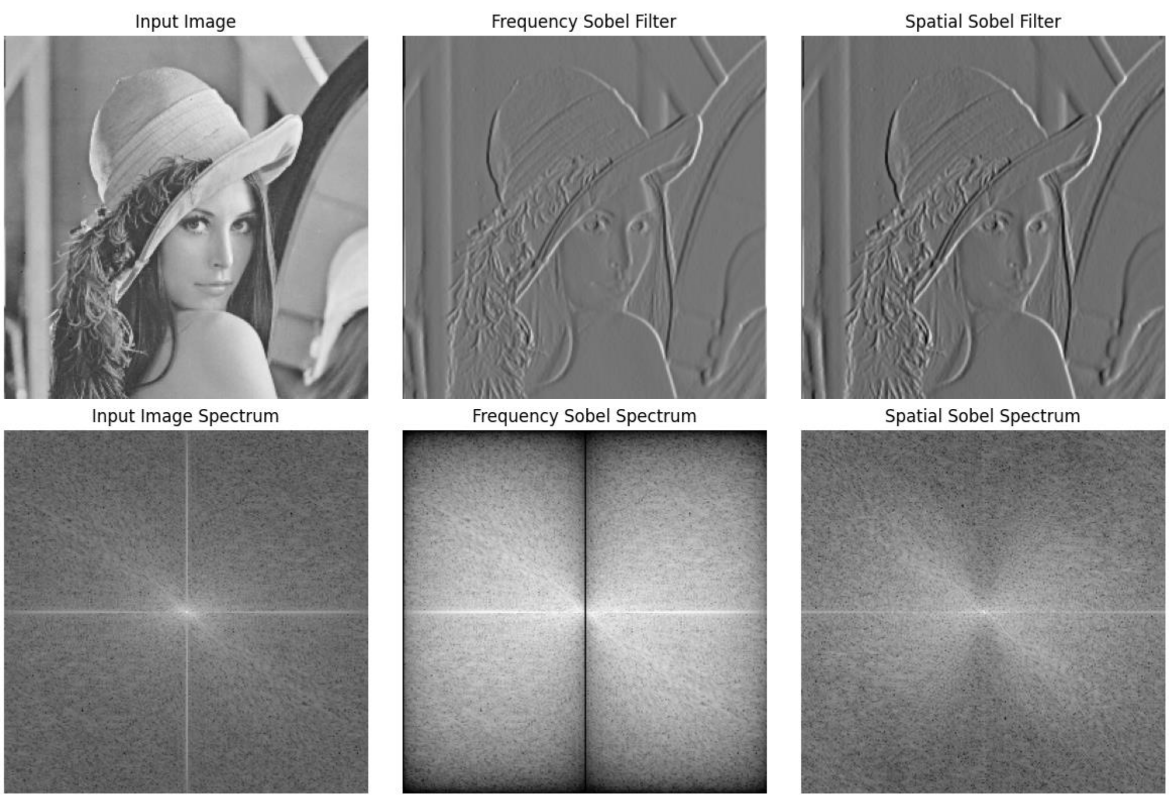


Figure 7(a)(b)(c)(d)(e)(f). (a) Input image with size of 256x256. (b) Sobel filter applied in the frequency domain. (c) Sobel filter applied in the spatial domain. (d) Spectrum of the input image. (e) Spectrum of the image after applying Sobel filter in the frequency domain. (f) Spectrum of the image after applying Sobel filter in the spatial domain.

Experiment 3

Often an image has poor shading due to uneven lighting. To balance out shading we use a homomorphic filter. This filter enhances high frequencies and attenuates low frequencies, but will preserver find detail. Given a model of image formation $f(x,y) = i(x,y)r(x,y)$, where $i(x,y)$ is the illumination component and $r(x,y)$ is the reflectance component. Illumination varies slowly and affects low frequencies; reflectance varies faster and affects high frequencies. These low and high frequencies are intermixed together, the goal of homomorphic filtering is to separate these frequencies. Using a high-frequency emphasis filter shown below can achieve these results.

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

Here, D_0 is the cutoff frequency of the filter, γ_L and γ_H control the gains for low and high frequencies and C is a constant the controls slope of the function. The shape of the function is shown in Figure 8.

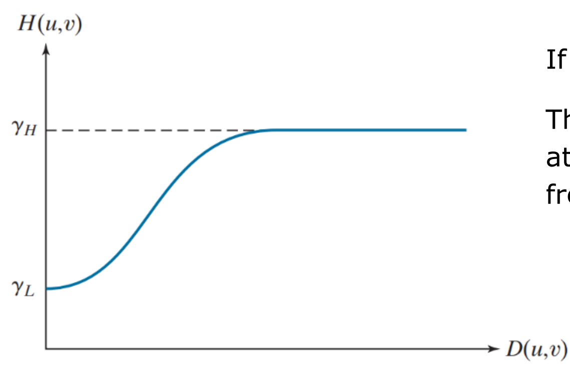


Figure 8. Approximate function shape of high-frequency filter.

If $\gamma_L < 1$ and $\gamma_H > 1$

Then the low frequencies are attenuated, and the high frequencies are amplified.

Figure 9(a) is the input image for which we will filter, notice the image has uneven shading notably brighter in the center and dark around the edges. Figure 9(b) is the generated high-pass filter that will be applied to the image. After applying the filter, the results can be seen in Figure 9(c), setting the parameters $D_0 = 1.8$, $\gamma_L = 0.3$, $\gamma_H = 1.1$, and $C = 1$ seems to produce the best results. Shading has been greatly improved, details around the edges are visible and the image center brightness has been reduced. Figure 10(a) displays the input image spectrum, and Figure 10(b) displays the filtered image spectrum.

Figure 9(a)(b)(c). (a) The input image with poor shading. (b) The high-pass filter to be applied. (c) Final filtered image with improved shading.

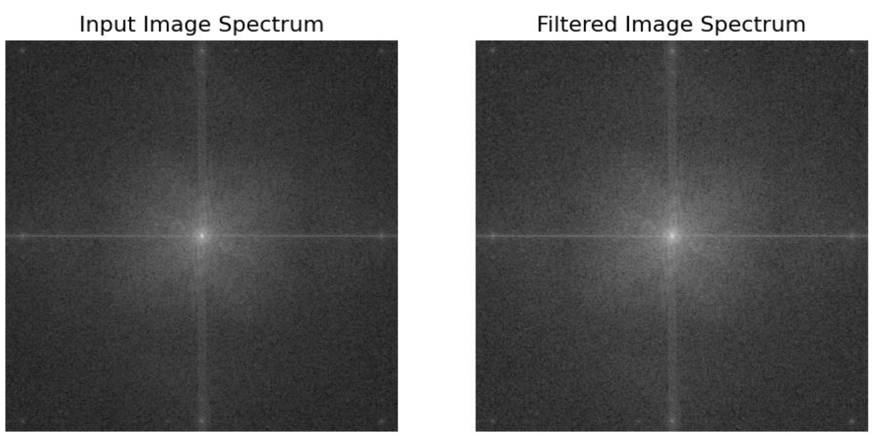


Figure 10(a)(b). (a) Input image spectrum. (b) Filtered image spectrum.