Image Processing HW2

Shang-Ying,He
Communication Engineering
National Tsing Hua University
Hsinchu,Taiwan. R.O.C.
sunnyhe1122@gmail.com

I. Introduction

Program about the image restoration and homomorphic filtering in frequency domain to perform the convolution and didn't use any filter API. In other words, convolution in spatial domain equals to multiply in frequency domain. And generating the filter using function by myself.

II. METHOD

A. Image Restoration

First task for image restoration is about using wiener filtering to deblur the images. And first step is to estimate the degradation function by observation, experimentation and mathematical modeling. In my observation for the image in figure 1. first row, which is about fish, it's a RGB image and Fourier transform is to convert image's intensity to frequency domain, so I converted it from RGB to HSI (HSV) and transform it's third dimension to frequency domain. We can see the result in figure 1. This image's low frequency's elements are larger than the others, so what I want to do is to inhibit the low frequency, and the function (1) is degradation function I estimated for this task. And then using (2) to generate the wiener filter. Result about generating filter is in figure 1. and multiplying filter with image which is in frequency domain to perform that when convolution in spatial domain also means multiplying in frequency domain.

$$H(u,v) = e^{-k(u^2+v^2)^{\frac{5}{6}}} \tag{1}$$

$$\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+K} \tag{2}$$

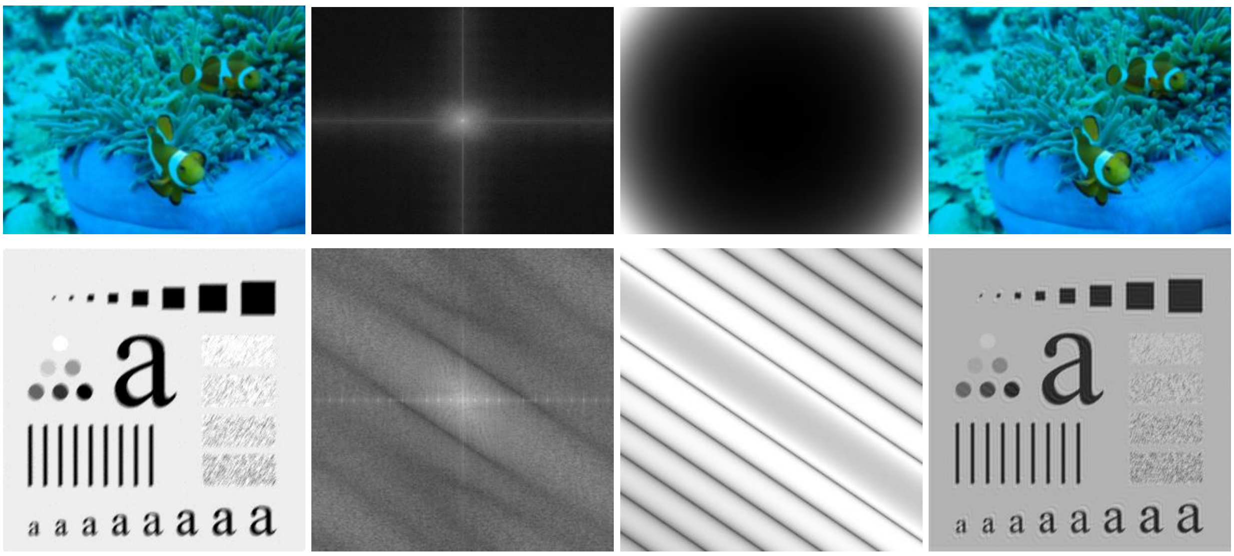
For the other image in figure1. about some patterns. We can observe that it has some oblique lines in frequency domain. That is this image has motion blur about x-direction and y-direction, so I use (3) to be the degradation function for this image's wiener filter, actually (3) is a sinc function with offset, so we can see the filter in figure 1. , it's wave just like a sinc. And the parameter 'a' and 'b' in (3) are the how oblique the x-direction and y-direction are. If we set them all equal to 1, then the result for filter will be symmetric. And I choose the parameter for a=0.0155, b=-0.0105,T=0.5 and K in wiener filter is 0.01.

$$\mathrm{H(u,v)} = \frac{T}{\pi(ua+vb)}\sin\pi(ua+vb)\, e^{-j\pi(ua+vb)} \tag{3}$$

Second task for image restoration is about using notch filter to denoise the image. See image in figure 2. which is about flower. For it's Fourier transform, we can see that there are two spot around the center and that's the target we needed to remove. So first step is to find it's coordinate, I use np.amax() function in numpy in rough interval to find the maximum value, then use np.argwhere() function to find the maximum value's coordinate. Next step is to remove the spot we founded by notch filter. Function about notch filter I use is (4). D1,D2 is Euclidean distance between two spots and the other pixels around them, D0 is the thresh hold to decide whether to remove. Coordinates I founded are (322,325) and (702,699), and I choose D0=30. We can see the filter and result in figure 2.

$$\mathrm{H(u,v)} = \begin{cases} 0, & \text{if } D_1(u,v) \le D_0 \;\; or \;\; D_2(u,v) \le D_0 \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

Final task for image restoration is about using band reject filter to denoise the image. See image in figure 2. which is about sunset. For it's Fourier transform, we can see that there are some spot in the image. They are very small, we may



original frequency domain filter result

Figure 1. result of using wiener filter to deblur images and their filter

zoom in the image to see them. So first step is to find their coordinate also, and the method I use is the same as previous. But the next step is different. Function I use is (5) D(u,v) is the Euclidean distance between origin and the other pixels, if it is in the interval we choose, we set it to zero. D0 and W are parameters about interval we choose. I choose D0=300 and W=60. Result of filter and filtered image is in figure 2.

$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)<D_0-W/2\\ 0 & \text{if } D_0-W/2\le D(u,v)\le D_0+W/2\\ 1 & \text{if } D(u,v)>D_0+W/2\end{cases} \tag{5}$$

B. *Homomorphic Filtering*

This part is about image enhancing, using ideal, Butterworth and Gaussian high pass filter and we called this method as homomorphic filtering. I choose the cut off frequency as 30 to extract high frequency elements in image. And then added them through a ratio to original image to get the enhanced image. We can see the result in figure 3. For ideal filter, there are some waves in the image but the others are not. For Butterworth filter, we can see the dark texture at the edge of objects but for the Gaussian filter, there is no this situation. In conclusion, from ideal, Butterworth to Gaussian filter, we can observe that the results are smooth gradually.

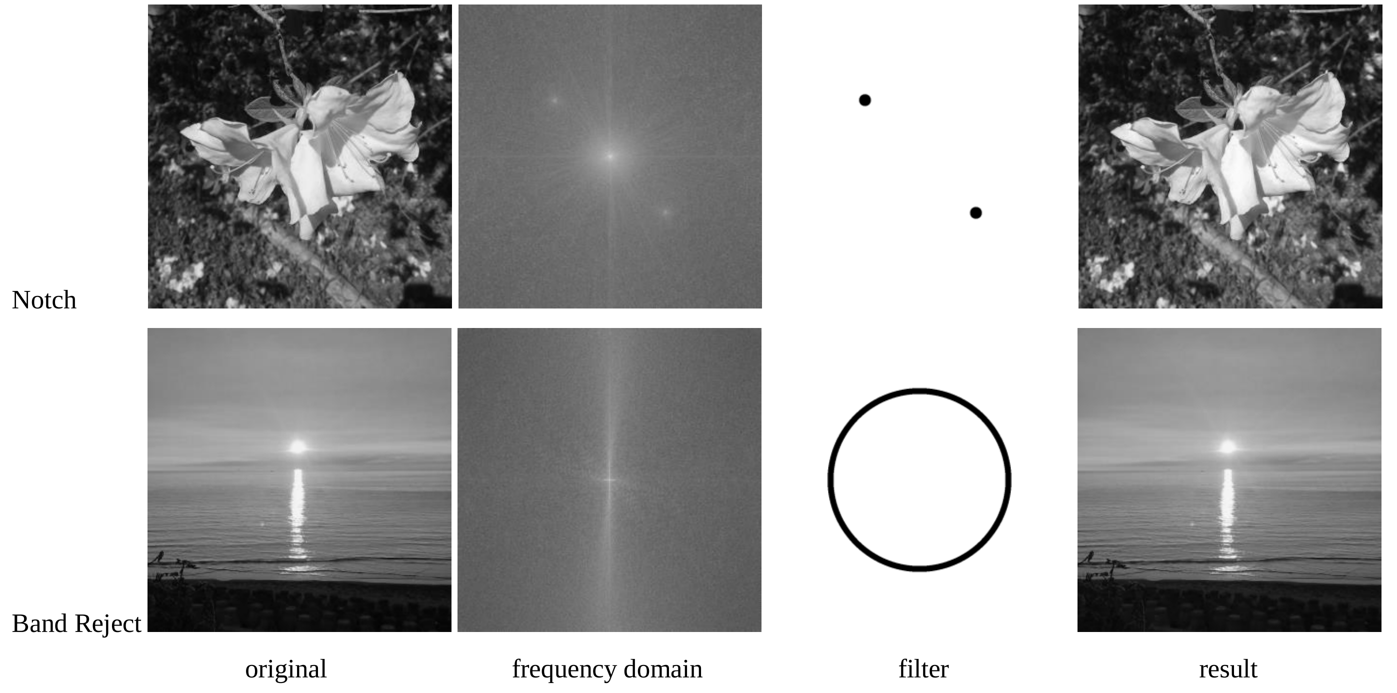


Figure 2. result of using notch filter and band reject filter to denoise images and their filter

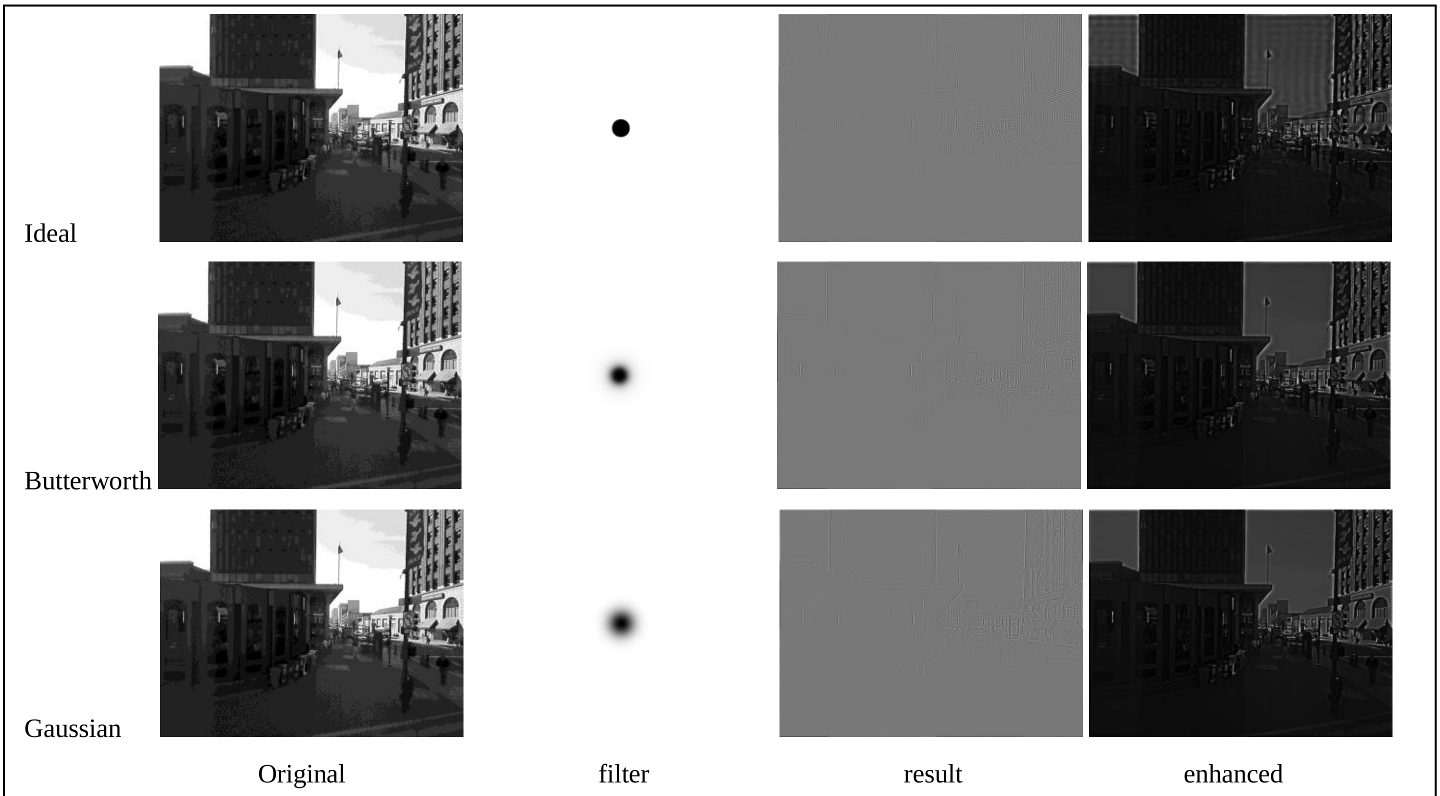


Figure 3. result of homomorphic filtering

III. EXPERIMENTS

A. Image Resoration

Result of wiener filtering in figure 1. we can observe that after filtering the image, result of images have been deblurred. But result about fish is not obvious as the other. I think the reason about it maybe the estimating of degradation function which I define is not so well. But I still try to find the better result on it.

Result of notch filtering in figure 2. we can observe that the original image has noise, if you zoom in the image. After filtering from frequency domain to remove the spot, we can observe from result that the noise of image has been removed, if you zoom in also.

Result of band reject filtering in figure 2. it's concept is the same as notch filter, remove the spot from frequency domain also means remove the noise in image. We can observe that the noise in the image has been removed from original image, if you zoom in also.

B. Homomorphic Filtering

Result of homomorphic filtering in figure 3. we can observe that the filter from mode of ideal, Butterworth to Gaussian are smooth gradually. About this, it's also reflect on the result of enhanced image result of Gaussian is much smoother than the others. After enhancing the image we can observe the edge of image easily.

IV. CONCLUSION

This project let me know more about the image restoration and image enhancing by homomorphic filter in frequency domain. About this, a problem cost some of my time is when we transform image from frequency domain to spatial domain, it's value will be the complex and much larger than origin, so I use np.log1p() to scale the values in the interval between 0 and 255. And why np.log1p() rather than np.log(), because zero can't put into log. When I complete this project, I have benefited a great deal.