[[1]](#footnote-1)

Lab 6

Chengpin Luo, 11510880, SUSTech

*Abstract*—In this lab, we are going to implement filtering in frequency domain further, including homomorphic filtering and band-reject filter. Additionally, template matching would be performed using correlations of the images.

*Index Terms*— Homomorphic filter, band-reject filter, correlation

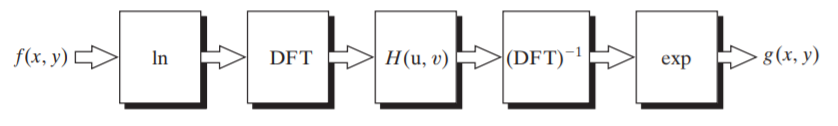
# INTRODUCTION

I

N order to improve the appearance of an image by simultaneous intensity range compression and contrast enhancement, we need to perform a frequency-domain filtering called “Homomorphic filtering”. Sometimes, an image was with periodic noise. To filter this noise out, we need band-reject filter, which can filter a range of the unwanted frequencies. Sometimes, we need to identify if a template is matched with an object in another image and we could do image correlations to figure this out. In this lab, these three techniques would be implemented in part II, III and IV respectively.

# Homomorphic filtering

Homomorphic filtering is a procedure that could compress the intensity range and enhance the contrast simultaneously, performed in frequency domain. A summary of this procedure is shown in figure 1. As the figure shown, the figure would firstly be log-transformed and then transformed into the frequency domain. After multiplied by the filter, the image would be inverse-transformed. Finally, we take the exponential of the filtered result to get the output image.



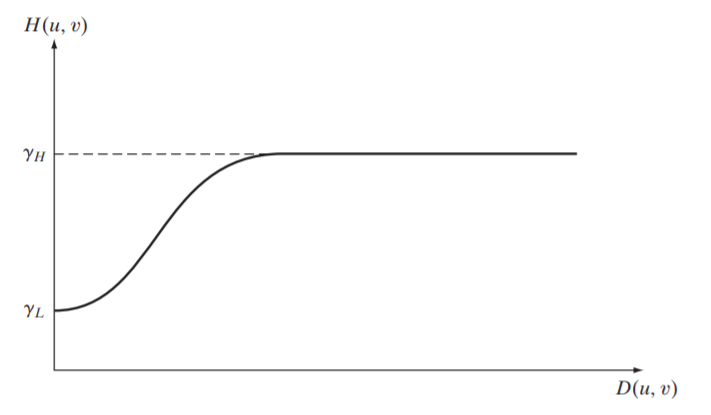
**Fig. 1.** Steps of Homomorphic filtering

Basically, an image could be expressed as the product of its illumination component, i(x,y), and reflectance components, r(x,y) as shown in formula (1). The log operation could then divide these two components apart and therefore the homomorphic filter function could process these two components respectively.

 (1)

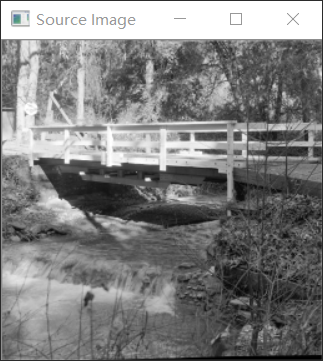
Generally, a homomorphic filter function could be described in formula (2) and figure 2.

 (2)

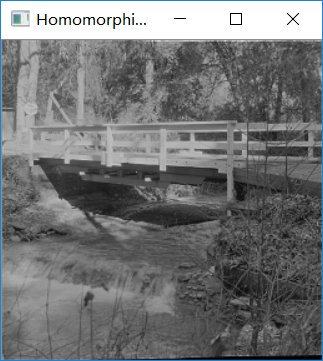


**Fig. 2.** Homomorphic filter function

Now, we are going to implement the homomorphic filtering. Two images in figure 3 was used. When we followed the instructions in figure 1 strictly, we could get result of goldhill.pgm. However, the result of bridge.pgm turned to be totally dark. I checked the range of the pixel values of bridge.pgm and found that the range is started from 0, which would result in –inf when doing log transformation. Therefore, we add the image values by 1 before log transformation and subtract the values by 1 after taking the exponential. And then we got the results in figure 4, where D0=100, c=1.0, gammaH=1.25 and gammaL=0.75. From the results, we can see that the dynamic range decreased in both images and the low intensities are more visible. And because the high frequencies are enhanced by homomorphic filtering, the reflectance components of the image (edge information) were more sharpened



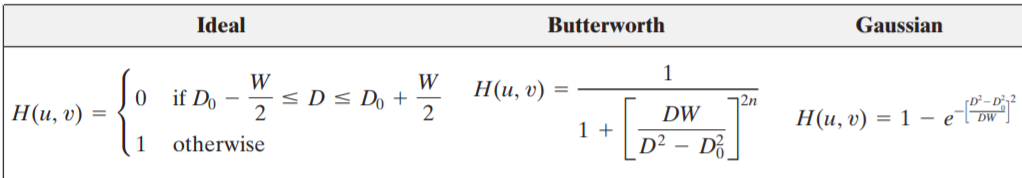
**Fig. 3.** Input image (L: bridge, R: Goldhill)

**Fig. 4.** Results of homomorphic filtering

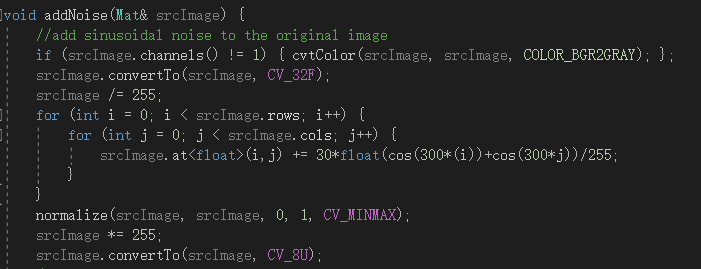
# Band-reject filtering

In this part, we are going to use bandreject filter to filter periodic noise. Three types of bandreject filter is shown in table 1. In this lab, we use only Butterworth bandreject filter.

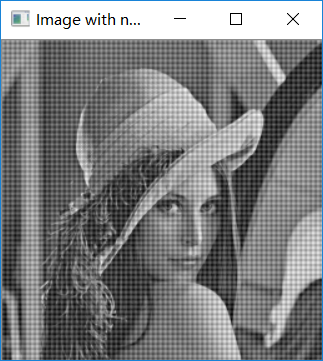
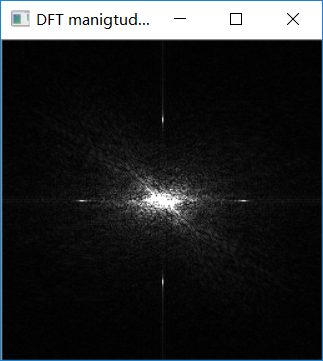


**Table 1.** Bandreject filter

Firstly, we added some sinusoidal noise to the image. The code is shown in figure 5. We add the cosine noise in the two directions respectively. Using lena.pgm as input image, the result image and its DFT spectrum is shown in figure 6. We can see that the DFT spectrum has four bright points surrounding the centre, which represents the sinusoidal noise.

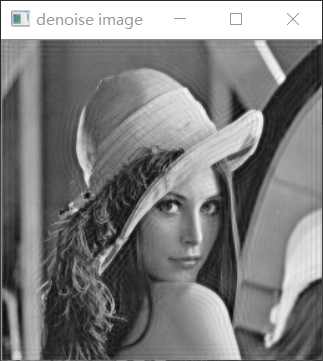
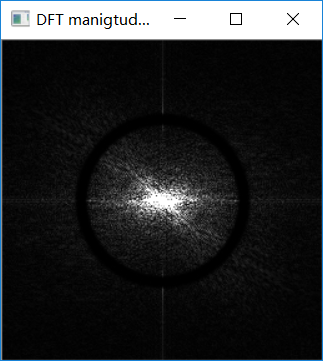


**Fig 5.** Code for adding sinusoidal noise

**Fig. 6.**Noisy image and its DFT

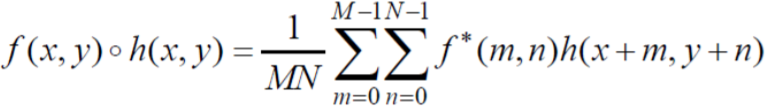
Then we implmented the bandreject filtering using the Butterworth filter, the result is shown in figure 7. It could be seen clearly from the result image that almost all of the noises were filtered, with some ring effects exist. And when we look at the DFT spectrum, we can see a dark band, which represented that the frequencies in the band has been filtered by the bandreject filter.



**Fig. 7.** Filtered image and its DFT

# Template matching

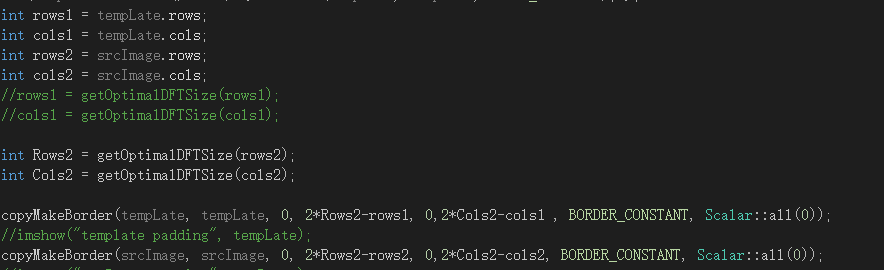
In this part, we are going to implement template mathing using image correlation. Suppose we have a template h(x,y) and an image f(x,y). If there is a match, the correlation of two functions h(x,y) and f(x,y) will reach the maximum at the location where h(x,y) will be most similar inside f(x,y). Image correlation is described in formula (3):

 (3)

In the frequency domain, we have:

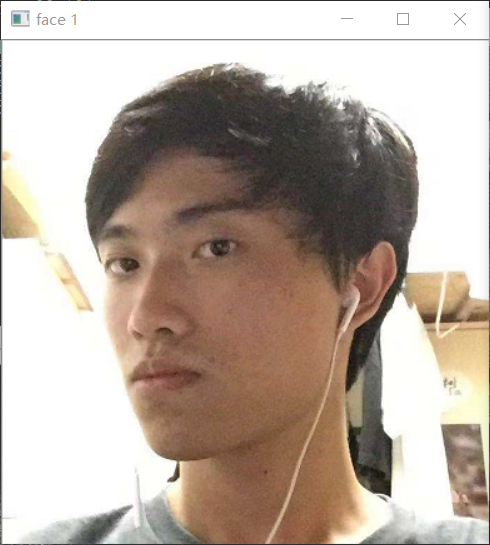
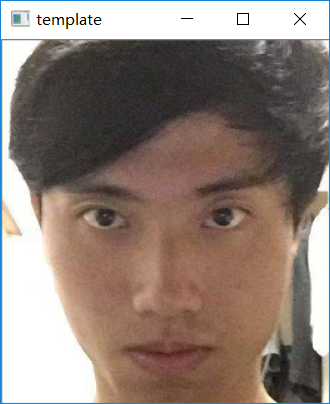
 (4)

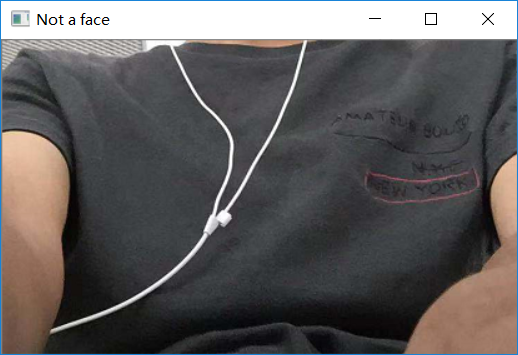
Therefore, when we implement correlation in the frequency domain, we just multiply the complex conjugate of one DFT with another one, and then inverse the DFT to the time domain. Additionally, in order to eliminate the wraparound error, we need to pad the images before taking the DFT. In this lab, the code for padding is shown in figure 8. After the inverse DFT, we need to crop the image for the final result.

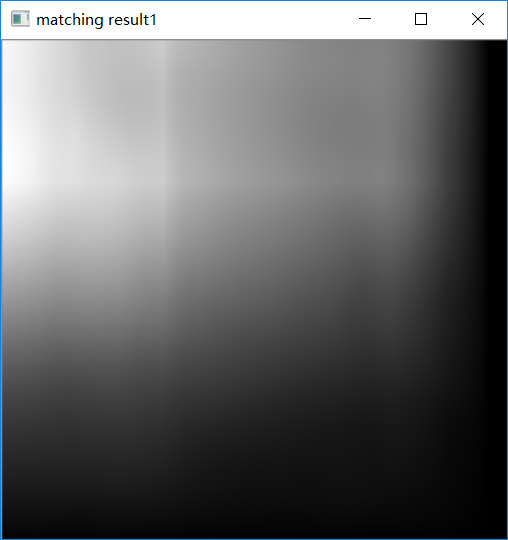
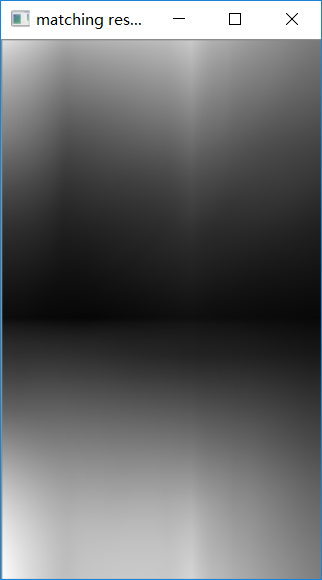


**Fig. 8.** Padding for eliminating wraparound errors

In our template matching task, I use one image of my face as template, one larger image with my face, and another large image without my face. Then I did the correlation between the template and the other two images respectively. The input image is shown in figure 9 and the correlation result is shown in figure 10.



 **Fig. 9.** Template, face and not-face

**Fig. 10.**Matching result(L:face,R:not a face)

In figure 10, we can see that the matching result 1 shows a maximum point at the top right- near the centre, there is a bright crossing point. However, in result 2 we could not find such a point. Therefore, it could be determined that figure 1 contains the face while figure 2 not.

# Conclusion

In this lab, we implement homomorphic filtering, band-reject filtering and image correlation. Using homomorphic could increase the contrast and compress the dynamic range at the same time while using band-reject filtering could filter the periodic noise as long as we could know the frequency of the noise using DFT. Finally, doing image correlation could help us figure out whether the template is in the matching image or not.

# Reference

[1] *Digital Image Processing* Gonzalez 4th edition

[2]<https://docs.opencv.org/2.4/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html>.

1. Chengpin Luo is with Electrical and Electronic Engineering Department, Southern University of Science and Technology, 518055, CHINA (Student Number: 11510880, e-mail: 11510880@mail.sustc.edu.cn). [↑](#footnote-ref-1)