



HACETTEPE UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

FALL 2018

**BBM 415
Image Processing Laboratory
Problem Set-2**

Supervisors:

Dr. Erkut Erdem

TA Ozge Yalcinkaya

Ege Berke Balseven

21590776

Author:

Number:

Subject:

Motion Deblurring in
Frequency Domain

Due Date:

23.11.2018

Contents

1	Introduction	2
2	Implementation Details	2
3	Experimental Results	3
3.1	Part 1: Exploring Deconvolution	3
3.2	Part 2: Finding the Blur Kernel	6
3.3	Part 3-A: Deconvolution	7
3.4	Part 3-B: Deconvolution with Regularization	8
3.5	Part 3-C: Bonus	9
3.6	Part 4: Richardson-Lucy	13
4	Conclusion	15

1 Introduction

This assignment includes four parts. The main purpose of this assignment, image deblurring with deconvolution method in frequency domain. Part1 is the process of discovering how deconvolution works. In this part since i use the original picture, i can obtain the transformation kernel by dividing that with blur image. After that i get the restored image by diving blur image with transformation kernel, that is how deconvolution works. In part2 i have only blur images, so that i have to guess kernels which is called blind deconvolution. I examined the magnitude spectrum of each image to guess the applied kernels. In part3-A, i have blurred images and their kernels. I did deconvolution for restoring but i obtain noisy because of i didn't consider the noise. After in part3-B, to get rid of noise, i am doing threshold. In part 3-C, as a different method, i use Wiener deconvolution for deblurring. I can restore the picture real time with track bar. At part4, i implement Richardson-Lucy's iterative formula by using the given kernel and the blurred image.

2 Implementation Details

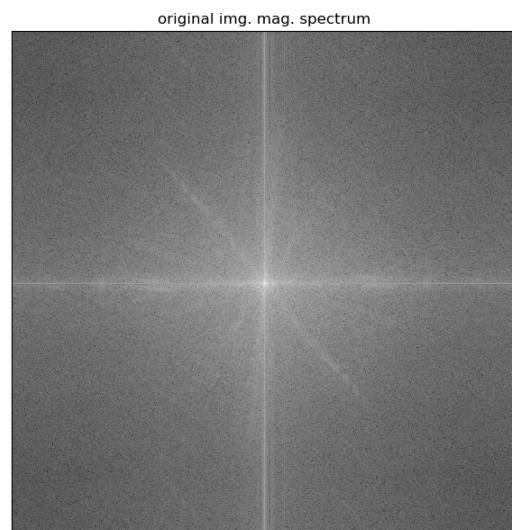
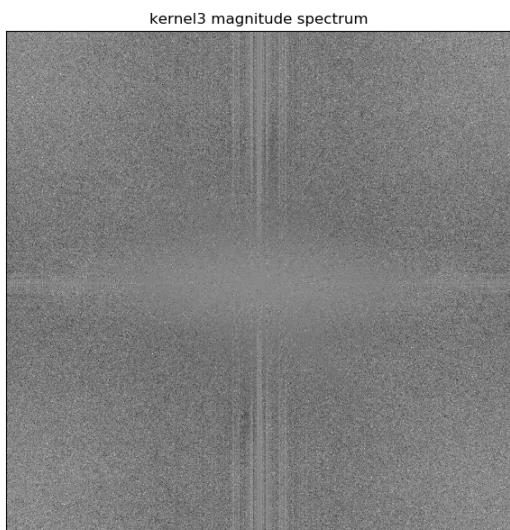
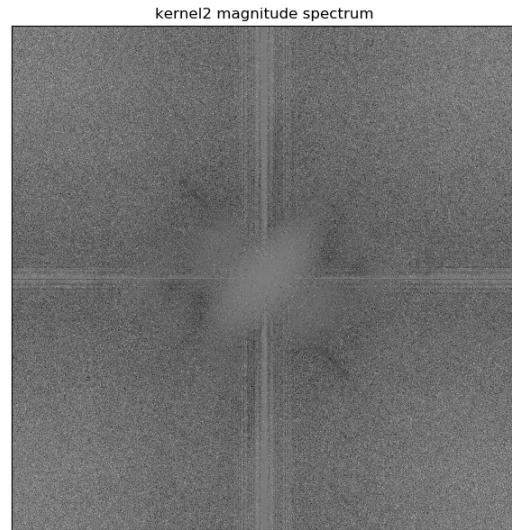
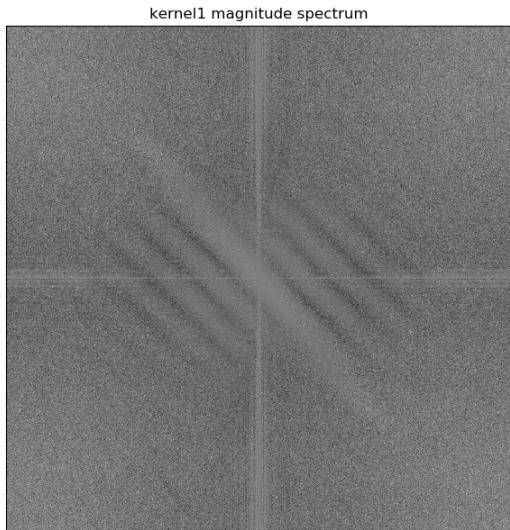
Part1 was not difficult. It taught me deconvolution operation and fourier transforms. But it was too hard to get to guess the right kernel in part2. In Part3, the result looked like be correct if the correct kernels were obtained. But inverse fourier transform and threshold operation took long time. I obtained all kinds of different results. It was understood that there were not hard mistakes after realizing it, but it was very difficult to find. In part4, there was only the implementation of the formula so I have not encountered much trouble.

3 Experimental Results

3.1 Part 1: Exploring Deconvolution



Input images: blur1, blur2, blur3, original image



magnitude spectrums:

blur1's kernel has 135 degree angle and high frequency

blur2's kernel has 45 degree angle and low frequency

blur3' kernel has 0 degree angle and very low frequency

original image's magnitude spectrum is different from kernels

deconvolution result1



deconvolution result2



deconvolution result3



original image



deconvolution results:

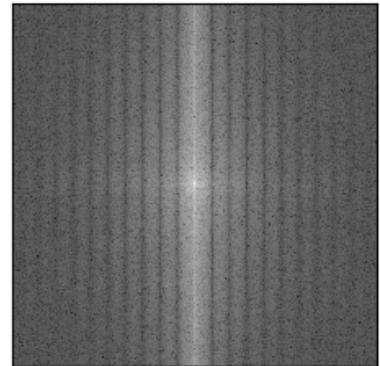
all results are the same as the original image

3.2 Part 2: Finding the Blur Kernel

angle 0 kernel magnitude spectrum



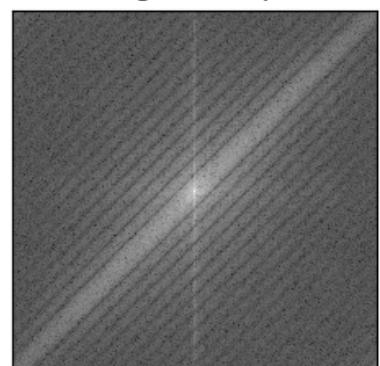
blur1 magnitude spectrum



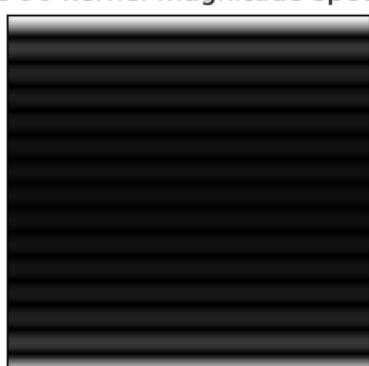
angle 135 kernel magnitude spectrum



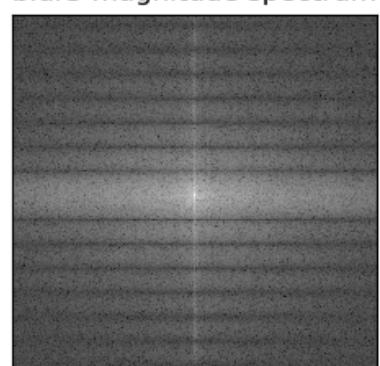
blur2 magnitude spectrum



angle 90 kernel magnitude spectrum

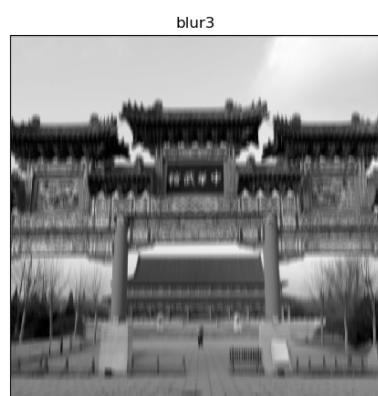
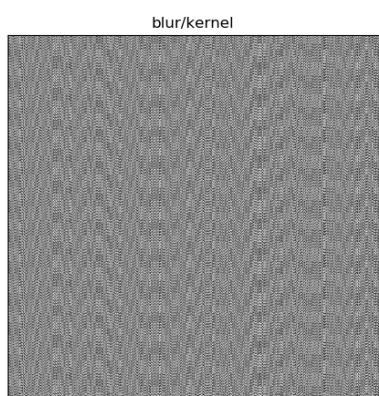
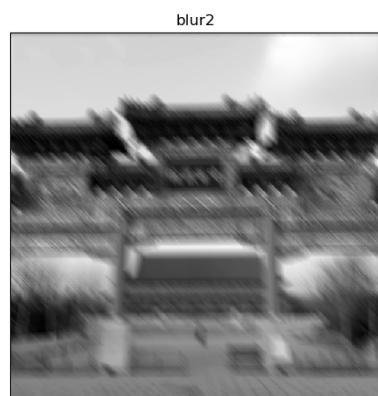
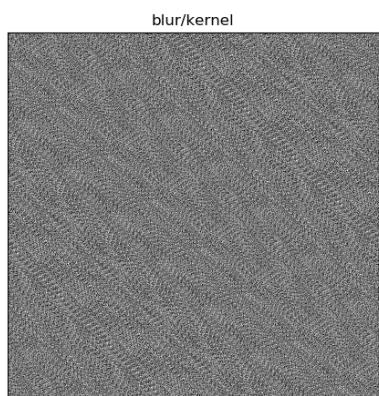
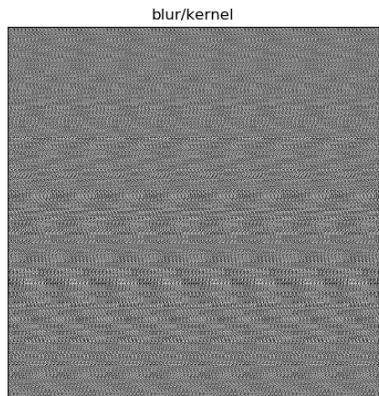


blur3 magnitude spectrum



The kernels magnitude spectrums which i created are similar to blur images magnitude spectrums. I obtain the restore image by dividing these two values together and i am correcting their frequencies. Because corrupted pictures have such frequencies.

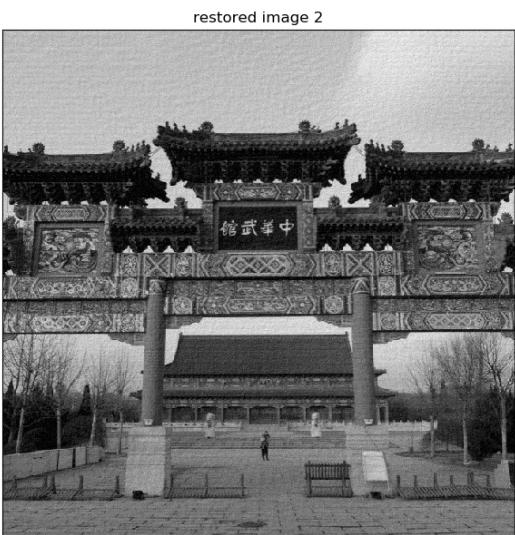
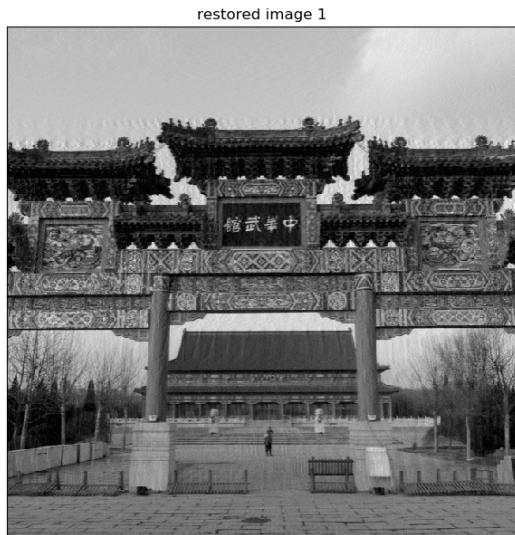
3.3 Part 3-A: Deconvolution

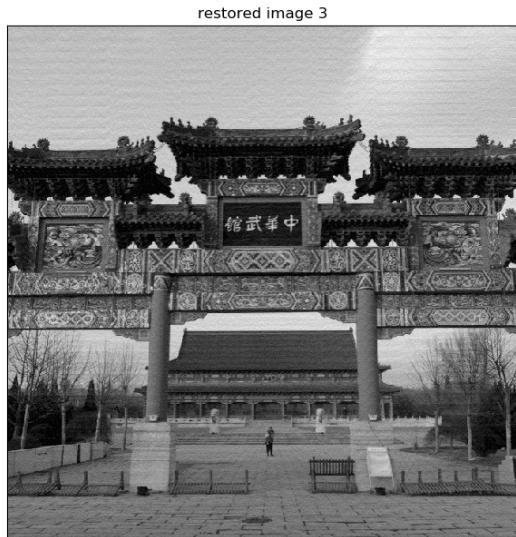


The reason of the get such noisy images, the kernels on the part1 are includes noise, but the kernels i found have not noise, so i get the noisy pictures when the kernels are not includes the noises.

3.4 Part 3-B: Deconvolution with Regularization

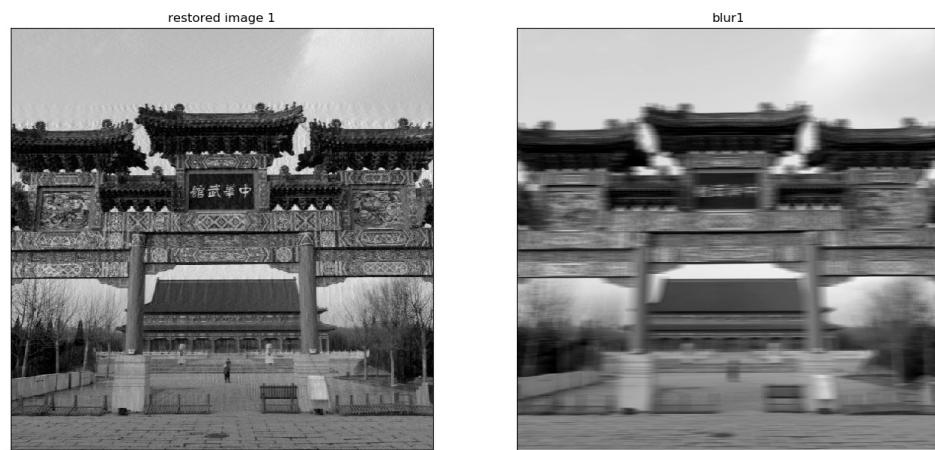
In this part, i did threshold operation to the kernels, so i removed the noises in the images and obtain restored images. Best threshold values are near the zero of this images (0.04, 0.01, 0.01). The best kernel sizes are: 21, 21, 15. Here are the restored images:





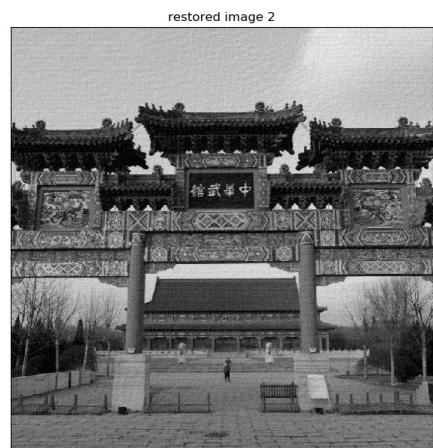
3.5 Part 3-C: Bonus

In this bonus part, as a different method, i use Wiener deconvolution for deblurring. Also i can restore the picture real time with track bar. The results are similar to part3-B. Upper ones are Wiener deconvolution results, below ones are part3-B results. d = diameter, angle = degrees, snr= signal/noise ratio. Angles are same as part3-b, diameters also similar as part3-b kernel sizes.



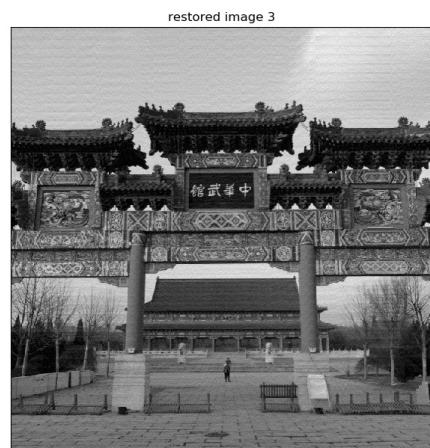
blur1 results

Picture edges slightly blurred, nearly same with part3-b.



blur2 results

Picture edges slightly blurred, part3-b obviously better.

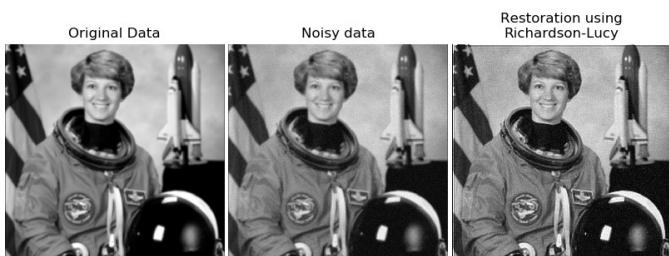
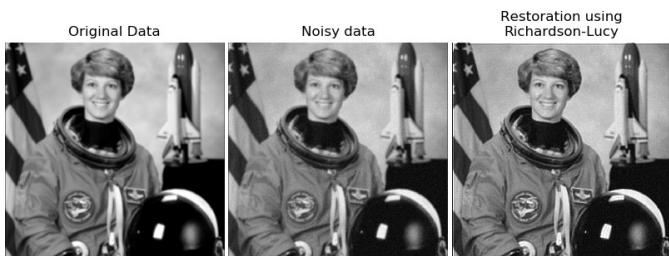
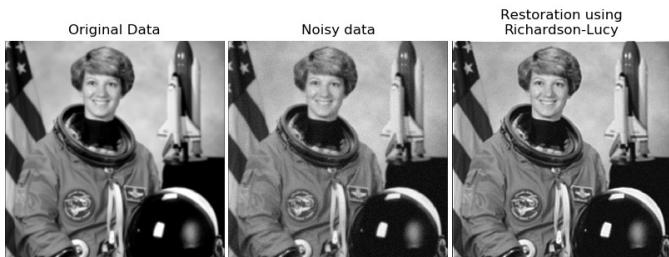


blur3 results

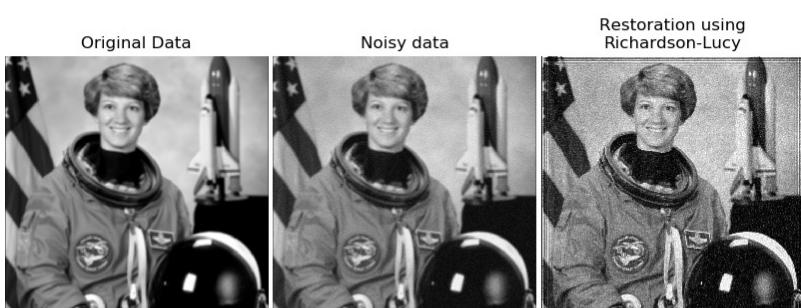
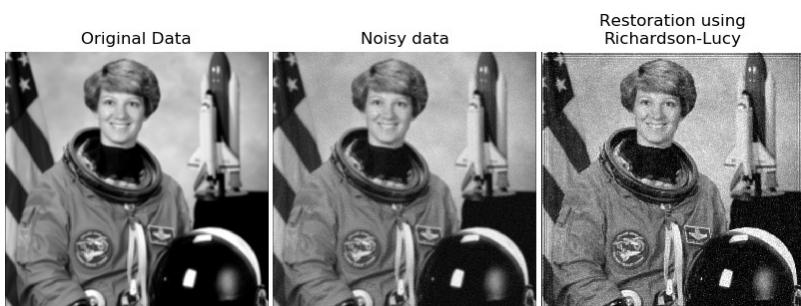
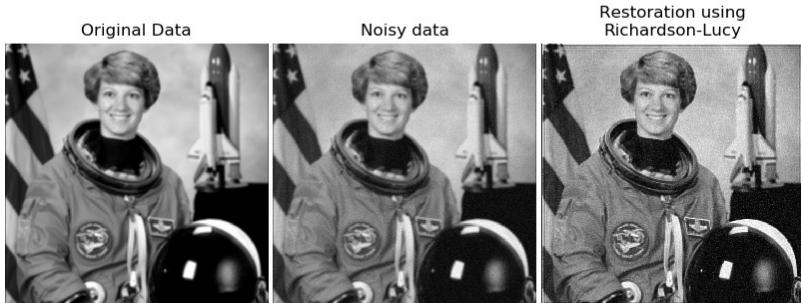
Picture edges slightly blurred, part3-b slightly better.

3.6 Part 4: Richardson-Lucy

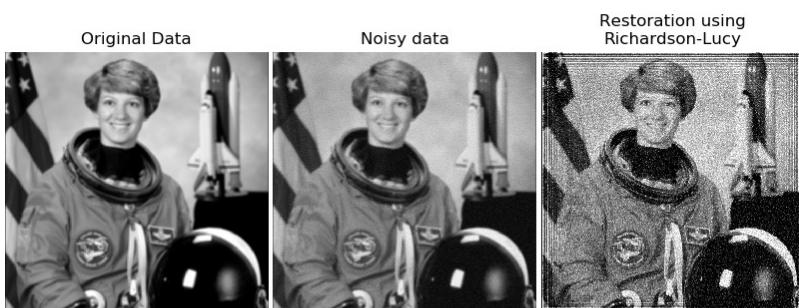
This part shows us deconvolution with Richardson-Lucy's. Richardson-Lucy is a iterative algorithm that recovers an original image from its noisy and blurred version. Here are the different results according to the number of iterations:



iterations in order of = 5, 10, 20
almost unchanged



iterations in order of = 30, 50, 75. ideal result is 30. 50, 75 are not bad



iterations in order of = 150, 300
results are so noisy but a bit sharp

4 Conclusion

This assignment is quite tutorial on fourier transform and deconvolution methods. Image deblurring is one of the most important subject in image processing. Furthermore this assignment requires a lot of research. So i learn so many things. Part2 and part3 were quite difficult. But in the end, results i got for each part are successfully.