CENG466 – Fundamentals of Image Processing Assignment 2

Ertuğrul Aypek
Computer Engineering
Middle East Technical University
2171270
ertugrul.aypek@metu.edu.tr

Kemal Baş
Computer Engineering
Middle East Technical University
2237048
kemal.bas@metu.edu.tr

I. INTRODUCTION

In this assingment, we will apply frequency domain transforming and perform operations in the frequency domain for image enhancement and image compression. First operation that we will perform is "noise reduction" and Second one is "edge detection" in frequency domain. We did not complete the third part.

II. FIRST TASK – NOISE REDUCTION USING FREQUENCY DOMAIN FILTERING

Noise reduction in frequency domain is straightforward, and it is case dependent. We take fourrier transform of the image and shift the resulted frequency domain image. Then for better observation we take logarithm of that image. This process makes transformed image namely "zoomed" so we can see more details. After these ooperations we observe the image, determine the noise in the transformed image, pick a noise filter in frequency domain and apply that filter to image. Latsly we take inverse log of the transformed image and take it back to spatial domain by inverse transform.

In the first image, after we transformed, shifted and took the logarithm of the image We saw that there are two major noise lines which are very bright. One on y axis, one on x axis. We filtered these noises by setting their value to 0, then took inverse logarithm and inverse transform and got the final image. The final image was not very different than the original image but the noise in the image was small anyways.

In second image, after we transformed, shifted and took the logarithm of the image We saw that there are many circular noises as two different size circles In the frequncy domain image they were forming two cocentric circles, one the outer formed by bigger noise circles the inner one formed by smaller noise circles. The first thing that we thought was creating two different sized highpass filters and apply them to corresponding noises one by one, but that would take too many time. Instead we realized something. The intensity values of these noises in the frequency domain image were very close so we determined a range for these noises from minimum value of the noises to maximum value of the noises.

After that we applied a threshold filter to transformed image. That is if the pixel value of the image is in this range we set its value to 0. When we saw the result of the filtering there were small white pixels in the noise that are negligible (because they were out of range extreme values) but the problem was that: the middle of the transforme image was having many different pixel values getting brighter. At one point these brightness values fell into our range and this made image blurred. So we added a condition to filter: If the coordinate of the pixel that fell into range is closer than first noise cicle to center of the image, we did not set its value to 0 because it cannot be a noise. After that our denoised image was not blurred.

Figure 1. Denoised image A2

In the third image which is an RGB image, we first separated the RGB elements of the image. After, we transformed, shifted and took the logarithm of the seperated images and we saw that there are stripes of noises in transformed image. Again we constructed coordinate specific line filters to eliminate these stripe noises. There were some noise on the diagonals of the transformed image, again we filtered these noises and set their value to 0 in transformed image. After filtering part, we inverse transformed these 3 images and concatenate these images to one RGB image. The result was much better than the input but still there was some visible noise in the result image. We knew that it may not be possible to eliminate all noise in the image and we filtered all the noises that we saw in frequency domain so we leaved the result as it is.

Figure 2. Denoised image of A3

III. SECOND TASK – EDGE DETECTION

Edge detection task in frequency domain can be achieved by using high-pass filters. A high-pass filter is a filter that passes the values of high frequencies while eliminating low frequency values.

Detailed content of an image forms low frequency value in frequency domain. However, core contents like edges form high frequency values. Thus, using a high-pass filter will serve for the purpose of edge detection.

In order to reduce the complexity, we worked on grayscaled version of the images.

We observed that using ideal high-pass filter does not give nice results as the outputs contain some noise. Thus, we used a smooth high-pass filter and the results get better.

Figure 3. 1200x630 smooth high-pass filter used for B1

Figure 2. 480x720 smooth high-pass filter used in B2

We observed that the radius of the filter is important to find more accurate edge maps. As an image gets more detailed, bigger circle regions are required to eliminate details. For it, we used radius of 40 for B1 and radius of 55 for B2. This numbers were chosen by trial and error method. Below are final results of our method.

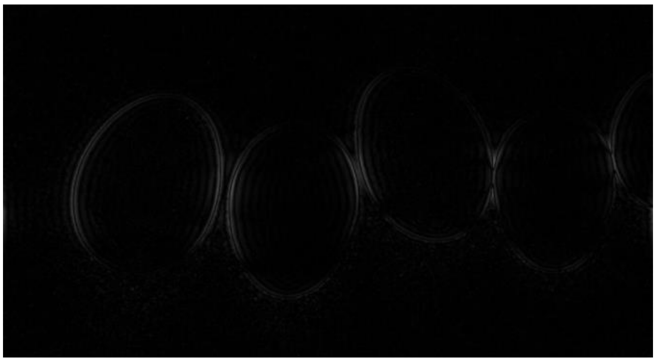

Figure 4. Result of our method for B1

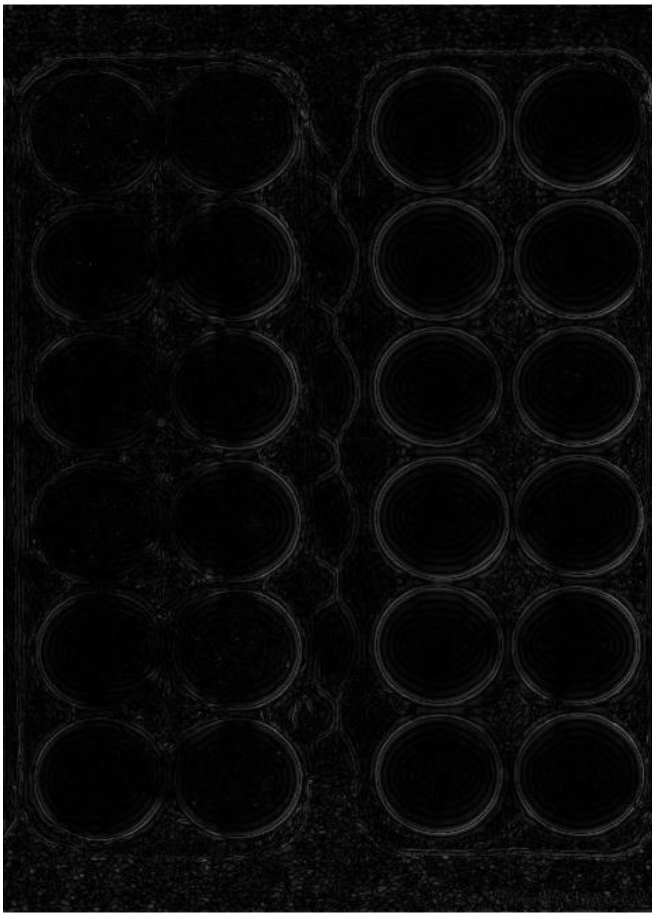

Figure 5. Extracted edge maps of B2

IV. THIRD TASK – IMAGE COMPRESSION USING WAVELET DECOMPOSITION

As we said in the introduction we did not perform this part.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published