CENG466 Digital Image Processing Take Home Exam 2 Solutions

Yekta Demirci
Electrical and Electronics Department
Middle East Technical University
Ankara, Turkey
e209360@metu.edu.tr

Zumrud Shukurlu
Computer Engineering Department
Middle East Technical University
Ankara, Turkey
e2174761@ceng.metu.edu.tr

***Abstract*—In this document, the solutions proposed for Take Home Exam 2 are explained and the results are discussed.**

***Index Terms*—Discrete Wavelet Transformation, Haar, Edge Detection, Discrete Fourier Transform, Bandreject filters, Notch filters, Image Processing**

I. Introduction

The steps followed to do Take Home Exam 2 are explained clearly. The results are shown and discussed at the end of each section. All the codes are implemented in MATLAB language. For the **first question**, 3 level discrete wavelet is taken and the mixed images are found. In the **second part**, types of noise have been detected in the images and the images have been restored by bandreject and notch filters. Finally in the **last question**, the image and the kernel are multiplied in the frequency domain rather than convolution in spatial domain to detect edges.

II. Part 1: Wavelet Transformation

In this question, two gray scale images are given and it is told that they are reconstructed from 3 level wavelet coefficients of the original images by using haar function. However during the reconstruction some of the coefficients are mixed up, therefore the resulted reconstructed images are not clean and have characteristics of the other image. It is required to take discrete wavelet of these 2 reconstructed images to find mixed coefficients.

For this purpose `dwt2(I,'haar')` function of MATLAB is used. In order to get 3 level decomposition, approximation coefficients are applied to this given function to obtain finer details.

At each dwt2 use, 4 elements are returned; an approximation, horizontal detail, vertical detail and diagonal detail. Then applying dwt2 function to the approximation again the second level details and approximation can be obtaibed. By this way, up to 3 level details are obtained. Since 4 coefficients are returned and it is applied 3 times, 12 coefficients are obtained at the end per image, where an approximation is equal to the next level transformation coefficients. At each level, each size of the image is halved. For instance, A1.png has an size of 640x422, at each level, details and approximations will be halved like 320x211, 160x106, 80x53. By using these the approximation, details of a level and the inverse dwt2 function, images can be constructed back. Basically 4 elements obtained per level from dwt2 are needed to fully construct the images back. Approximation has the information left from the image after details are taken apart. Vertical details show the changes on the vertical axis, horizontal shows the changes on the horizontal axis and diagonal shows the changes both on vertical and horizontal combined. Also there is difference between levels, as we go to deeper levels we obtain finer details and approximation starts to disappear since details are taken apart from that at each level.

At level 3(deepest level) nothing is mixed, so the same coefficients are used back. Then at level 2, two details are mixed between the images. It is the diagonal one for the A1 image(ccH2 matrix in the code) and the vertical one for the A2 image(cD2 matrix in the code). At level 1, four details are mixed up. They are vertical and diagonal details of the A1 image(ccV1 and ccD1 matrixes in the code), vertical and diagonal details of A2 image(cH1 and cV1 matrixes in the code)

After putting back the details into the correct places, images are reconstructed by using idwt2() and saved.

III. Part 2: Noise Elimination

The principal aim of this task is the recognition of different noise types in the images and amending them with suitable filtering techniques.

A. B1.png

In the first image, to ease the analysis of the image, we applied histogram equalization to the image and observed that there are patterns of **sinusiodal noise** in the image. Furthermore, we applied Fast Fourier Transformation (`fft2()` function in MATLAB) to the image and observed two sets of dots with different distances from the center of the transformed image.

We adjusted the circles to zero with a **bandreject filter** that set values to zero in the specified two intervals. The intervals have been determined by testing different values after the general algorithm for the filter had been determined. Alternatively, intervals could be found by thresholding a copy of the frequency domain image so that all intensities below a particular value would be set to zero and looping through that image to find the first nonzero index and setting the found indices to zero in the filter as the last step.

B. B2.png

This image has been fixed with the same principle as the first image, since we noticed from its histogram equalization and the Fourier domain image that there are similar **sinusoidal noise** patterns in the image, only there are three instead of two circles in its Fourier transformation. The same routine of finding inner and outer radii of three circles and constructing the appropriate **bandreject filter** for eliminating them has been applied.

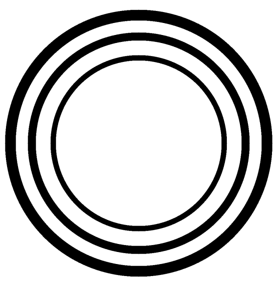

Fig. 1. Bandreject filter applied to the Fourier transformation of the second image

Discussion: After applying the filter and reverse transformation to return to the spatial domain, the moire-like wavy pattern that was visible in the corners of the image has disappeared for both images.

Notes:

- Bandreject filters in the book had only one circle for the desired interval, but as it would be repetitive in the code, we merged the filters for all intervals in one filter and applied all at once for each image.
- Though the output images had nonuniform distribution of intensity values, we didn't apply histogram equalization to the output image, since it was not the requirement of this homework.

C. B3.png

It is easy to see frmo the original image that there are vertical lines that disrupts the uniformity. In its Fourier transformation, there are bursts of light in the horizontal axis additional to the light of the ideal Fourier spectrum. To remove those spikes, we applied a **notch filter** to the transformed image, eliminated the sparks by setting the square proximity of them to zero. The coordinates of spikes was found by analysing the image with naked eye and determining the approximate value by manual testing on the magnitude of transformed image for better visualisation.

Discussion: After filtering, the vertical lines of noise in the original image was reduced, but not completely, since we could not filter all values that was spread in the transformed image out.

Note: Notch filters in the book had only one pair of values for elimination, but as it would be repetitive in the code, we merged the filters for two pairs in one filter and applied all at once for this image.

IV. PART 3: EDGE DETECTION

In this question, edge detection is required to be done by relating the convolution in spatial domain with multiplication in frequency domain. Basically convolution in spatial domain corresponds to multiplication in frequency domain and vice versa. By using this, discrete Fourier transform of the image and the filters are taken. The filters are 2x2 sized Roberts Cross kernels given in the HW1. One of them is for detecting the edges on horizontal and the other one is for vertical axis. Then the results will be combiner. 2x2 kernels give better resolution therefore Roberts kernels used instead of Sobel kernels. However Sobel kernel could be used in a similar way, as well. The important point is matching kernel size with the image size. Padding of fft2() function is used for this purpose. By this way, we obtained the same sized matrices in fourier domain and element wise multiplication is done. Then inverse fourier transformation is taken. The edges on X and Y directions are combined and finally overall edges are found. Thanks to the duality property of the fourier transform this filtering is almost the same as the spatial domain filtering. Some small difference occurs between the detected edges because of sampling and quantization through the process.

Fig. 2. Resulted C1_output.png

REFERENCES

[1] Gonzalez, R. C., & Woods, R. E. (2002). Digital image processing. Upper Saddle River, N.J: Prentice Hall. Chicago.