CS5680/CS6680 – Fall Semester 2024
Assignment 4 – Filter Techniques in the Frequency Domain
Due: 11:59 p.m. Sunday, October 20, 2024
Total Points: 50 points

For all the Fourier transform-related questions, make sure that the center of the image in the frequency domain corresponds to the lowest frequency of (0, 0).

Problem I: Exercises on Low-pass and High-pass Filters in the Frequency Domain [Total: 6 points]
1. [3 points] Design a Gaussian **low-pass** filter with the standard deviation σ_1 of 20 at the *u* direction (row) and the standard deviation σ_2 of 70 at the *v* direction (column) in the frequency domain (Refer to slide 63 of Ch3.2.DIPBasicFreq.pdf for the equation, where $\sigma_1 = \sigma_2$. Obtain the filtered image by filtering the original image ***Sample*** with the designed Gaussian filter (i.e., perform the Gaussian low-pass pixel-wise multiplication operation in the Fourier frequency domain). Display the original image, the Gaussian low-pass filter **(treat it as an image)**, and the filtered image in Figure 1 with appropriate titles.

2. [3 points] Design a Butterworth **high-pass** filter of order 2 with a cutoff frequency of 50 in the frequency domain (Refer to slide 73 of Ch3.2.DIPBasicFreq.pdf for the equation). Obtain the filtered image by filtering the original image ***Sample*** with the designed Butterworth filter (i.e., perform the Butterworth high-pass pixel-wise multiplication operation in the Fourier frequency domain). Display the original image, the Butterworth high-pass filter **(treat it as an image)**, and the filtered image in Figure 2 with appropriate titles.

Problem II: Exercise on Certain Operations in the Frequency Domain [Total: 6 points]
1. [3 points] Apply the Fourier transform on two images ***Sample*** and ***Capital***, respectively. Display the magnitude and the phase of the two Fourier-transformed images in Figure 3 with appropriate titles (Refer to slide 11 of Ch3.2.DIPBasicFreq.pdf for the equations). **Note: Appropriate scaling operations are needed for proper display due to the large dynamic range in the frequency domain. Specifically, the log transformation needs to be applied to the magnitude before scaling operation to ensure proper display.**

2. [3 points] Exchange the phase components of the two Fourier-transformed images and take an inverse Fourier transform. That is, use the phase of ***Capital*** and the magnitude of ***Sample*** to reconstruct the new ***Capital*** image. Use the phase of ***Sample*** and the magnitude of ***Capital*** to reconstruct the new ***Sample*** image. Display the two reconstructed images in Figure 4 with appropriate titles.

Problem III: Remove Additive Cosine Noise [Total: 12 points]
The noisy image ***boy_noisy.gif*** has been generated by adding some noise in the form of a cosine function. The task is to remove the cosine interference to restore the original image. This can be done in the Fourier domain as follows:

1. **[1 point]** Compute the **centered** DFT (Discrete Fourier Transform) of the noisy image.
2. **[3 points]** Compute the magnitude image of the centered DFT image and find eight non-center locations (i.e., frequencies) containing the **four largest distinct magnitudes**. **Please design an appropriate function to get this task done.**
3. **[3 points]** Replace the value at each of the eight locations of the centered DFT image found in step 2 with the average of its 8 neighbors. **Please design an appropriate function to get this task done.**
4. **[1 point]** Take the inverse DFT transform of the modified non-centered DFT image and display the original image and the restored image side-by-side in Figure 5 with appropriate titles.
5. **[2 points]** Display the following restored images side-by-side in Figure 6 with appropriate titles, where each restored image is obtained by taking the inverse DFT transform of the modified non-centered DFT image produced as follows:

1) Replacing the value at each location in the centered DFT image, which contains the two largest distinct magnitudes by excluding the magnitude at the center (it is a very large value), with the average of its 8 neighbors.
2) Replacing the value at each location in the centered DFT image, which contains the three largest distinct magnitudes by excluding the magnitude at the center, with the average of its 8 neighbors.
3) Replacing the value at each location in the centered DFT image, which contains the five largest distinct magnitudes by excluding the magnitude at the center, with the average of its 8 neighbors.
4) Replacing the value at each location in the centered DFT image, which contains the six largest distinct magnitudes by excluding the magnitude at the center, with the average of its 8 neighbors.

6. **[2 points]** On the console, summarize the differences between the five resultant images (i.e., one image shown in Figure 5 and four images shown in Figure 6).

Problem IV: Preliminary Wavelet Transform [Total: 12 points]

1. [2 points] Call a built-in function to compute the maximum decomposition level for image ***Lena***. Apply a maximum-level "db2" wavelet decomposition on ***Lena*** by using the appropriate function(s). Apply the inverse wavelet transform to restore the image. Use the "if-else" statement to compare your restored image with the original image so the appropriate message indicating the equality or inequality between these two images is displayed. Note: The original and the restored images should be the same.

2. [8 points] Apply a 3-level "db2" wavelet decomposition on ***Lena*** by using appropriate function(s). **Independently** perform the inverse wavelet transform after each of the following operations:

a) Set the 16 values of each 4×4 non-overlapping block in the approximation subband as its average.
b) Set the first level vertical detail coefficients as 0's.
c) Set the second level horizontal detail coefficients as 0's.
d) Set the third level diagonal detail coefficients as 0's.

Display the four reconstructed images in Figures 7, 8, 9, and 10, respectively.
For the above four "set" operations, please do not call built-in functions.

3. [2 points] On the console, summarize the differences between the original image and each of the four reconstructed images obtained from a), b), c), and d) and explain why the aforementioned operations caused these differences.

Problem V: A Simple Solution to Remove Gaussian White Noise [14 points]

1. **[1 point]** Load in ***Lena*** and call a built-in function to add Gaussian white noise of 0 mean and 0.01 variance and save this noisy image as "NoisyLena.bmp".

Denoising Method 1: **[8 points]**

1. Load in "NoisyLena.bmp".
2. **[1 point]** Apply a 3-level "db2" wavelet decomposition on the noisy *Lena*.
3. **[1 point]** Estimate the noise standard deviation at the 1st-level diagonal wavelet subband (e.g., HH_1) using $\sigma \;\; = [median(|f_{ij}\,|)/0.6745]$, where f_{ij} represents all wavelet coefficients in the HH_1 subband.
4. **[1 point]** Compute the adaptive threshold t of the 1st-level wavelet subband by $t = \sigma\sqrt{2\ln M}$, where M is the number of coefficients in the 1st-level wavelet subbands (LH_1, HL_1, and HH_1).
5. **[3 points]** Modify the wavelet coefficients f_{ij} in LH_1, HL_1, and HH_1 subbands using the soft thresholding:

$$f'_{ij} = \begin{cases} f_{ij} - t & \text{if } f_{ij} \geq t \\ f_{ij} + t & \text{if } f_{ij} \leq -t \\ 0 & \text{if } |f_{ij}| < t \end{cases} \qquad \text{(Eq. 1)}$$

6. **[0.5 points]** Apply steps 3, 4, and 5 on the 2nd-level wavelet subbands (e.g., LH_2, HL_2, and HH_2) by using the information in the 2nd-level subbands (i.e., apply similar ideas explained in steps 3, 4, and 5 on the 2nd-level wavelet subbands).
7. **[0.5 points]** Apply steps 3, 4, and 5 on the 3rd-level wavelet subbands (e.g., LH_3, HL_3, and HH_3) by using the information in the 3rd-level subbands (i.e., apply similar ideas explained in steps 3, 4, and 5 on the 3rd-level wavelet subbands).
8. **[1 point]** Take the inverse wavelet transform to get the denoised image.

Denoising Method 2: **[3 points]**

1. Load in "NoisyLena.bmp".
2. Apply a 3-level "db2" wavelet decomposition on the noisy *Lena*.
3. Estimate the noise standard deviation at the 1st-level wavelet subband (e.g., LH_1, HL_1, and HH_1) using $\sigma = [median(|f_{ij}|)/0.6745]$, where f_{ij} represents all wavelet coefficients in LH_1, HL_1, and HH_1 subbands.
4. Compute the adaptive threshold t of the 1st-level wavelet subbands by $t = \sigma\sqrt{2 \ln M}$, where M is the number of coefficients in the 1st-level wavelet subbands (LH_1, HL_1, and HH_1).
5. Modify the wavelet coefficients f_{ij} in LH_1, HL_1, and HH_1 subbands using the same soft thresholding technique summarized in (Eq. 1)
6. Apply steps 3, 4, and 5 on the 2nd-level wavelet subbands (e.g., LH_2, HL_2, and HH_2) by using the information in the 2nd-level subbands (i.e., apply similar ideas explained in steps 3, 4, and 5 on the 2nd-level wavelet subbands).
7. Apply steps 3, 4, and 5 on the 3rd-level wavelet subbands (e.g., LH_3, HL_3, and HH_3) by using the information in the 3rd-level subbands (i.e., apply similar ideas explained in steps 3, 4, and 5 on the 3rd-level wavelet subbands).
8. Take the inverse wavelet transform to get the denoised image.

You must design appropriate functions for steps 3, 4, and 5 and call these functions to solve this problem.

Note: The only difference between the Denoising Method 1 and the Denoising Method 2 is the estimation of the noise standard deviation. Please try to modularize your codes via appropriate functions to provide efficient solutions.

[2 points] Compute the PSNR (Peak Signal to Noise Ratio) value between the original image ***Lena*** and the noisy image ***NoisyLena***, the PSNR value between the original image ***Lena*** and the denoised image obtained by the Denoising Method 1, and the PSRN value between the original image ***Lena*** and the denoised image obtained by the Denoising Method 2. Display the noisy image and the two denoised images obtained by the Desnoising Method 1 and the Denoising Method 2 side-by-side in Figure 10 and output the major visual differences between the two denoised images and the differences in terms of the four PSNR values on the console.