

CS5680/CS6680 – Fall Semester 2021
Assignment 4 – Filter Techniques in the Frequency Domain
Due: 11:59 p.m. Saturday, October 23, 2021
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: 8 points]

1. [4 points] Design a Gaussian **low-pass** filter with the standard deviation σ_1 of 75 at u direction (row) and the standard deviation σ_2 of 25 at v direction (column) in the frequency domain (Refer to slide 63 of Ch3.2.DIPBasicFreq.pdf for the equation with $\sigma_1=\sigma_2$). 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. [4 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: 8 points]

1. [4 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. [4 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 corresponding 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. Your goal is to remove the cosine interference. This can be done as follows:

1. [1 point] Compute the centered DFT (Discrete Fourier Transform) of the noisy image.
2. [3 points] Compute the magnitude and find the frequencies (e.g., locations) corresponding to the **four largest distinct magnitudes** by excluding the magnitude at the center (a very large value). **Please design an appropriate function to get this task done.**
3. [3 points] Replace the value of each of the four frequencies (e.g., locations), which correspond to the four largest distinct magnitudes found in step 2), by the average of its 8 neighbors. Note: Make sure you will perform this change on their symmetric points (i.e., their complex conjugate counterpart) to ensure the proper reconstruction. **Please design an appropriate function to get this task done.**
4. [1 point] Take the inverse DFT transform and display the original image and the resultant image side-by-side in figure 5 with appropriate titles.
5. [2 points] Instead of finding the frequencies corresponding to four largest distinct magnitudes as mentioned in Step 2 and replacing the value of each of the four frequencies as mentioned in Step 3, generate four resultant images by respectively finding the frequencies corresponding to two, three, five,

and six largest distinct magnitudes and replacing the value of each of the two, three, five, and six largest distinct magnitudes. Display the resultant images side-by-side in figure 6 with appropriate titles.

6. [2 points] On Matlab console, summarize the differences of 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: 10 points]

1. [2 points] Call a Matlab built-in function to compute the maximum decomposition level for image *Lena*. Apply a maximum-level “db2” wavelet decomposition on *Lena* by using appropriate Matlab function(s). Apply the inverse wavelet transform to restore the image. Use “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. [6 points] Apply a 3-level “db2” wavelet decomposition on *Lena* by using appropriate Matlab function(s).

Independently perform the inverse wavelet transform after each of the following operations:

- a) Set the four values of each 2×2 non-overlapping block in the approximation subband as its average.
- b) Set the first level vertical detail coefficients (e.g., LH₁) as 0’s.
- c) Set the third level diagonal detail coefficients (e.g., HH₃) as 0’s.

Display the three reconstructed images in figures 7, 8, and 9, respectively.

For the above three “set” operations, please do not call Matlab built-in functions.

3. [2 points] On Matlab console, summarize the differences between the original image and each of the three resultant images obtained from a), b), and c) and explain why the operations mentioned in a), b), and c) caused these differences.

Problem V: A Simple Solution to Remove Gaussian White Noise [12 points]

1. [1 point] Load in *Lena* and call `imnoise` function to add Gaussian white noise of 0 mean and 0.01 variance.
2. [1 point] Apply a 3-level “db2” wavelet decomposition on the noisy *Lena*.
3. [2 points] Estimate the noise variance at the 1st-level wavelet subband (e.g., LH₁, HL₁, and HH₁) using $\sigma^2 = [\text{median}(|f_{ij}|)]/0.6745$, where f_{ij} represents all wavelet coefficients in LH₁, HL₁, and HH₁ subbands.
4. [1 point] Compute the adaptive threshold t of the 1st-level wavelet subband by $t = \sigma\sqrt{2\log M}$, where M is the number of elements in the 1st-level wavelet subband (LH₁, HL₁, and HH₁).
5. [4 points] Modify the wavelet coefficient f_{ij} in the LH₁, HL₁, and HH₁ subband using the soft thresholding:

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

6. [1 point] Apply steps 3, 4, and 5 on the 2nd-level wavelet subband (e.g., LH₂, HL₂, and HH₂) by using information in the 2nd-level subband (i.e., apply similar ideas explained in steps 3, 4, and 5 on the 2nd-level wavelet subband).
7. [1 point] Apply steps 3, 4, and 5 on the 3rd-level wavelet subband (e.g., LH₃, HL₃, and HH₃) by using information in the 3rd-level subband (i.e., apply similar ideas explained in steps 3, 4, and 5 on the 3rd-level wavelet subband).
8. [1 point] Take the inverse wavelet transform to get the denoised image.
9. Display the noisy image and the denoised image side-by-side in figure 10.

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