DIP Assignment #3: Filtering in the Frequency Domain

Due: Wednesday, November 6, 2019

INTRODUCTION Welcome to your first DIP assignment! This assignment consists of two parts. In the first part are some simple exercises, while the second part requires certain programming works. Please submit a report (in PDF format) and all relevant codes as the solutions. Warning: We encourage discussions among students, but the assignment should be finished independently, without copying existed answers. **Plagiarism = Fail**. Besides, there may be at least 30% penalty for late submission.

1 EXERCISES

Answer the following questions in your report.

1.1 Rotation (10 Points)

Figure. 1(b) was generated by:

1. Multiplying Fig. 1(a) by $(-1)^{x+y}$;
2. Computing the discrete Fourier transform;
3. Taking the complex conjugate of the transform;
4. Computing the inverse discrete Fourier transform;
5. Multiplying the real part of the result by $(-1)^{x+y}$.

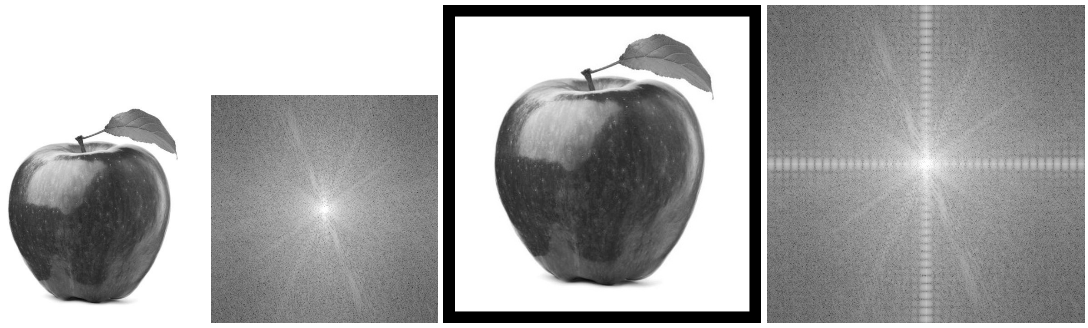
Explain (mathematically) why Fig. 1(b) appears as it does.

1.2 Fourier Spectrum (10 Points)

The Fourier spectrum in Figure. 2(b) corresponds to the original image Figure. 2(a). Figure. 2(c) is Figure. 2(a) padded with zeros, and Figure. 2(d) is the Fourier spectrum of Figure. 2(c). Explain the significant increase in signal strength along the vertical and horizontal axes of Figure. 2(d) compared with Figure. 2(b).

(a) (b)

Figure 1: Rotation



(a) (b) (c) (d)

Figure 2: Fourier Spectrum

1.3 Lowpass and Highpass (10 Points)

1). Find the equivalent Filter $H(u,v)$ in the frequency domain for the following spatial filter:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

2). Is $H(u,v)$ a low-pass filter or a high-pass filter? Prove it mathematically.

2 Programming Tasks

Write programs to finish the following two tasks, and answer questions in your report. Do not forget to submit all relevant codes and a README.txt to demonstrate the details of your development environment (e.g., language, version, third-party packages).

Pre-requirement

1. **Input**: Please download the archive "hw3.zip", unzip it and choose the image according to the last two digits of your student ID. This image is the initial input of your programming tasks in this assignment. For example, if your student ID is "14110563", then you should take "63.png" as your input. You can convert the image format (to BMP, JPEG, ...) if necessary. Make sure that you have selected the correct image. Misusing images may result in zero scores.

2. **Language**: Any language is allowed (python is recommended), but you can only use element-wise indexing and scalar operations. Advanced indexing (including but not limited to row-wise/col-wise/range indexing) and vectorization are forbidden. As an example, given a matrix A, if you want to divide all its elements by 2, you should use loops over matrix elements, sequentially performing "A[i][j] /= 2", instead of directly writing "A /= 2" in python with numpy. This "annoying" requirement will be removed in future homework. **Particularly, if non-scripting language is chosen, please submit the executable file and define the corresponding input in README.txt additionally.**
3. **Others**: There remain some issues that you should pay attention to:
    a) You can use third-party packages for loading/saving images (e.g., opencv or PIL).
    b) Good UX (User Experience) is encouraged, but will only bring you negligible bonuses. Please don't spend too much time on it, since this is not an HCI course.
    c) Keep your codes clean and well-documented. Bad coding styles will result in 20% penalty at most.

2.1 Fourier Transform (40 Points)

Write a function to perform 2-D Discrete Fourier Transform (DFT) or Inverse Discrete Fourier Transform (IDFT). The function prototype is "dft2d(input_img, flags) → output_img", returning the DFT / IDFT result of the given input. "flags" is a parameter to specify whether DFT or IDFT is required.
For the report, please load your input image and use your program to:

1. Perform DFT and manually paste the (centered) Fourier spectrum on your report. (10 Points)
2. Perform IDFT on the result of the last question, and paste the real part on your report. Note: the real part should be very similar to your input image. (Why? Think about it.) (10 Points)
3. Detailed discuss how you implement DFT / IDFT in less than 2 pages. Please focus on the algorithm part. Don't widely copy/paste your codes in the report, since your codes are also submitted. (20 Points)

2.2 Bonus: Fast Fourier Transform (50 Points)

This is an optional task. You are required to manually implement the Fast Fourier Transform (FFT). The function prototype is "fft2d(input_img, flags)→ output_img", returning the FFT / IFFT result of the given input. "flags" is a parameter to specify whether FFT or IFFT is required. "fft2d" should produce very similar results in comparison to "dft2d". However, your implementation may be limited to images whose sizes are integer powers of 2. We recommend you to handle this problem by simply padding the given input so as to obtain a proper size.
For the report, please load your input image and use your program to:

1. Perform FFT and manually paste the (centered) Fourier spectrum on your report. (10 Points)
2. Perform IFFT on the result of the last question, and paste the real part on your report. (10 Points)
3. Explain why does FFT hove a lower time complexity than DFT. (15 Points)
4. Detailed discuss how you implement FFT / IFFT in less than 2 pages. (15 Points)

2.3 Filtering in the Frequency Domain (30 Points)

Write a function that performs filtering in the frequency domain. The function prototype is "filter2d_freq(input_img, filter) → output_img", where "filter" is the given filter. You can modify the prototype if necessary. According to the convolution theorem, filtering in the frequency domain requires you to apply DFT / FFT to the given image and filter, and then multiply them, followed by IDFT / IFFT to get the filtered result. Hence, it should be easy to implement "filter2d_freq" based on "dft2d" (or "fft2d").

For the report, please load your input image and use your "filter2d_freq" function to:

1. Smooth your input image with an 5×5 averaging filter. Paste your result on the report. (8 Points)
2. Sharpen your input image with the following 3×3 Laplacian filter and then paste the result. (8 Points)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

3. Detailed discuss how you implement the filtering operation function in less than 2 pages. (14 Points)