EE7150 Computer Project IV
Frequency Domain Filtering (Chapter 4)
Due Date: Oct 7, 2021

1. **Two-Dimensional Fast Fourier Transform**

   The purpose of this part of the project is to develop a 2-D FFT matlab function that implements the steps in page-313 of Section 4.7 [Section 4.7.3 of 3rd Edition]. The function will be used in several projects below. Your function must have the capabilities to perform the following steps with an input image:

   a) Multiply the input image by $(-1)^{x+y}$ to center the transform for filtering.
   b) Compute the 2-D Fourier transform.
   c) Multiply the resulting (complex) array by a real filter function (in the sense that the real coefficients multiply both the real and imaginary parts of the transforms). Recall that multiplication of two images is done on pairs of corresponding elements.
   d) Compute the inverse 2-D Fourier transform.
   e) Multiply the result by $(-1)^{x+y}$. Take the real part of the IDFT and output the filtered image.

   Use this function to implement the following frequency domain filtering problems.

2. **Fourier Spectrum and Average Value**
   a) Download Fig. 4.40(a) from Pilot and compute its (centered) Fourier spectrum. [Note for students with 3rd Edition: Fig. 4.41(a) has been replaced with Fig. 4.40(a) in 4th Ed.]
   b) Display the spectrum.
   c) Use your result in part-(a) to compute the average value of the image.

3. **Lowpass Filtering**
   a) Implement the Gaussian lowpass filter in Eq. (4-116) [Eq. (4.8-7) of 3rd Ed]. You must be able to specify the size, $M \times N$, of the resulting 2D function. In addition, you must be able to specify the location of the center of the Gaussian function.
   b) Use Fig. 4.40(a) from Pilot and lowpass filter it to duplicate the results in Fig. 4.44.

4. **Highpass Filtering**
   a) Implement the Gaussian highpass filter from Table 4.6 [Eq. (4.9-4) of 3rd Ed.]. (Note: You can use the same program as in Part-3 to generate highpass filters)
   b) Use Fig. 4.40(a) from Pilot and highpass filter it to duplicate the results in Fig. 4.53.

5. **Highpass Filtering Combined with Thresholding**
   a) Download Fig. 4.55(a) [Fig. 4.57(a) of 3rd Ed.] from Pilot and use your program from Part-4 to approximate the results in Fig. 4.55 (Note that in this case, you will use a Gaussian, instead of a Butterworth filter).

Project Report Guidelines

Page 1. Cover Page. Typed or printed neatly.

- Project title
- Project number
- Course number
- Student's name
- Date due
- Date handed in

Page 2: Abstract (not to exceed half a page) – A gist of the objectives of the project.

Page 3: Technical discussion - One to two pages (max). This section should include the techniques used and the principal equations (if any) implemented.

Page 4 (or 5): Results - 4-5 pages, or as needed. Include all the images generated in the project. Number images individually so they can be referenced in the preceding discussions. Include titles, axis labels, legends and captions in the figures, as appropriate.

Discussion of Results: One to two pages (max). A discussion of results should include major findings in terms of meeting the project objectives, and make clear reference to any images generated.

Figures: Number the Figures sequentially as Figure 1, Figure 2, etc. and add captions/titles/axis markers to the figures, as appropriate.

Appendix: Program listings - Include listings of all program codes written by the student. The steps used in the codes should be well documented. Standard routines and other material obtained from other sources should be acknowledged by name, but their listings need not be included.

A note on program code implementation: The objective of the computer programs and codes used in the projects is for the students to learn how to manipulate images to obtain and display desirable results. There are numerous packages and tools that perform some of the functions required to implement the projects. However, the use of "canned" routines or GUIs as the only method to implement an entire project is discouraged. For example, if a student is using MATLAB and the Image Processing Toolbox, a balanced approach would be to use MATLAB's programming environment to write m-functions to implement the projects, using some of MATLAB's own in-built functions in the process. A good example is the implementation of the 2-D Fast Fourier Transform (FFT). The student should use the MATLAB in-built function that computes the 2-D FFT directly, but write code or functions for operations such as centering the transform, multiplying it by a filter function, and obtaining the spectrum.

A Note on Submission:

- **On-line students:** Upload the project report and matlab codes on Pilot. If multiple matlab functions are used, then create and upload a zip-folder.
- **In-class students:** Same as On-line students. In addition, submit a printout of the project report and all matlab codes on the due date in class.