Purpose: Become more proficient with frequency domain image representations and image enhancement techniques, as presented in Chapter 4 of the textbook.

Procedure: For *each* MATLAB m-file you write, be sure to comment the code sufficiently, and include initial comments as a crude help system. That is, if a user types "help" followed by your m-file name, comments will print on the screen which tell the user what the program does, what the proper syntax is to use the program, and any other useful tidbits. No Image Processing Toolbox (or *any other* Toolbox) programs should be used. If you're unsure, use the help files to see if the function you want to use is part of MATLAB or is listed as part of one of the Toolboxes.

For spatial domain images, the origin should appear in the upper left corner, and the image axes should be such that the x and y dimensions of an individual pixel should appear to be square. The x-axis is for the rows and the y-axis is for the columns. The gray levels should map 255 to white and 0 to black, and all final pixel values should be integers.

For frequency domain "images" (i.e., 2-D magnitude spectra), the origin should appear in the center, and the axes should be such that the u and v dimensions of an individual "pixel" should appear to be square (so that the aspect ratio of the original image is the same as the aspect ratio of the spectrum). The u-axis and the v-axis should be aligned properly with the orientation of the x-axis and y-axis of the image in the spatial domain. To display a spectrum as a 2-D image (as opposed to a 3-D mesh plot), the gray levels should be scaled if necessary (not just truncated) to map 255 to white and 0 to black, and all pixel values should be integers. Note that you may want to use the logarithm[1] of the spectral magnitudes (i.e., dB) for display purposes, depending upon the situation. Note also that a 3-D mesh plot of a spectrum (using either a linear or a logarithmic scale) can sometimes provide more insight as to which frequency components are actually present, as opposed to inspecting just the 2-D "image" of the spectrum. Use all the tools at your disposal!

Due to time constraints, only this first main part of the project is mandatory. The other parts are left included for you to see what the rest of the project would have included, if we had more time.

➪ You will be performing basic frequency domain filtering of images as part of this project. For the filters you will apply, you can choose either Butterworth or Gaussian LPF and HPF as covered in your text. You can also choose Butterworth or Gaussian variations of a notch filter, BPF, or BRF. You *don't* have to use all these types of filters; just a few will be necessary to complete the project! But the choice of which to use is up to you (yes, this is how I can assess your engineering judgment a bit). Any filters you design must have circularly symmetric filter functions (except for a notch filter, for which the notch pairs must only be symmetric with regard to the center point of the spectrum). It will be up to you to choose the best type of filter for the intended application.

- Whichever filter(s) you use, you must verify to me the proper operation of your frequency domain filtering program(s) with a test image **before** you proceed to a more complex image. To standardize this testing and make it easier for you and for me, we'll once again use the MAT file `iptest_im.mat` in which five test images reside; it is loaded on the "Images" section of the course website. Use the test image `iptest01` to *qualitatively* but *carefully* compare your filtering

[1] If you don't remember, reacquaint yourself with the difference in MATLAB between the `log` and the `log10` commands.

results to one or more of Figures 4.44, 4.46, and 4.53 in your text, and the similar figures in the lesson slides.[2] Which figure(s) you use for comparison depends upon which filter(s) you choose to implement. You *don't* have to implement and compare them all, and doing so will not earn any extra credit.

- You must implement *at least* one type of LPF and one type of HPF on test image `iptest01` and compare your results to the appropriate figure[3] in the text. You must show your filtered test images as figures in your report, and it should be clear upon visual inspection and by the text of your report that your filter is providing reasonably equivalent results to those shown in the appropriate textbook figure. You do not have to reproduce any of the figures directly from the text in your report, only your own filtered images. Your frequency domain filter program *must* take into account proper zero padding to avoid circular convolution, and it must output a filtered result that is a spatial domain image of the same

This is the end of the mandatory part of the project. The rest is NOT to be turned in, but you are free to do it on your own if you wish.

➪ Load the "mat" file `proj_04.mat` from the "Images" section of the course website. This file contains two MATLAB arrays:

- The array `p4a` is the **spectrum** of a 2196×2041 "mystery" image. The image may or may not have been zero padded and the spectrum origin may or may not have been shifted to the center by pre-multiplying the original image by $(-1)^{x+y}$ before the FFT was taken; you should be able to determine that by just looking at the spectrum.
- The array `p4b` is a spatial domain image of Lena (of size 512×512) that has been severely corrupted with undesirable noise. The corrupting signal is periodic in nature, and its frequency content does not have much overlap with the essential frequency content of the original Lena image, so the potential for image enhancement/restoration is very favorable.

An image of the magnitude of `p4a` is shown in Figure 1 and image `p4b` is shown in Figure 2, at the end of these instructions.

➪ Identify the "mystery" image in `p4a` and show (as a figure in your report) the spatial domain image from which the spectrum came. No filtering or enhancement to this image is required. This simple step just checks your ability to deal with spectral data, and it's fun.

➪ Using one or more frequency domain filters, try to remove as much of the corrupting noise in `p4b` as possible, while causing as little degradation to the Lena image as is practical. You may apply other enhancement techniques (such as gray scale manipulations in the spatial domain) after frequency domain filtering to improve the appearance of the final result, but all filtering for noise mitigation must be performed in the frequency domain. For this part of the report, include as a minimum these figures:

- the spatial domain corrupted Lena image provided to you,
- a 2-D "image" of the magnitude spectrum of the corrupted Lena image (in dB),

[2]Image `iptest01` is very similar to the 688×688 image shown in Figure 4.40(a) of your textbook.

[3]The HPF figures in the text have been "tweaked" by the publisher to look better, so your results will only come close to the textbook figures.

- a 3-D "mesh plot" of the magnitude spectrum of the corrupted Lena image (in linear units),
- a magnitude spectrum of your chosen filter function(s) (as a 2-D image or as a 3-D mesh plot—your choice),
- a 2-D "image" of the magnitude spectrum of the filtered image (in dB), and
- the spatial domain image result of the corrupted Lena image *after* filtering.

As always, provide clear and unambiguous numbered figure captions. That means a figure number **and** a helpful description of the figure.

Questions/Discussion: The write-up for this project report should be concise, using wording and formatting suitable for an IEEE technical journal, following the format referenced on the course web site. Be sure to include sufficient figures in your report of any images you create, modify, or that otherwise show results of your code. If you aren't sure if you should include it as a figure, then you probably should include it! Also, **don't make image figures too small** just to fit more on a page, and ensure that the font size used for figure titles and axis labels is large enough (similar in size to the font size in the main text of your report). The figures need to be large enough so that a printed version of your report would show the important details of the image.

As a minimum, address these points in your project report. These points *should be woven into the main text of your report*, not "tacked on" at the end as if they were an afterthought.

❶ Briefly discuss each image processing step you took to verify your filters, to identify the "mystery" image, and to filter the Lena image.

- Why did you perform each step?
- How well did each step achieve the intended purpose?

❷ For all filtering operations, be sure to *fully* specify any filter parameters you used for each figure shown.

- For the corrupted Lena image, why did you choose the particular filter *type* that you used?
- For the corrupted Lena image, why did you choose the specific filter *parameters* that you used?

If you used any other enhancement techniques, be sure to list them as well (and defend why you used them).

❸ Now that you've used both spatial domain and frequency domain image processing filters, how would you describe the advantages and disadvantages of each?

Turn in: For this Project, turn in (as one or more e-mail attachments sent to me as part of a single e-mail message):

➪ An electronic project report (as a PDF file regardless of the program used to create the report). Name your PDF file "`Last1_Last2_proj04.pdf`" please, where "Last1" is the last name of team member 1, and "Last2" is the last name of team member 2. Be sure to follow the format given on the course web site (see the `Admin` section of the course web site).

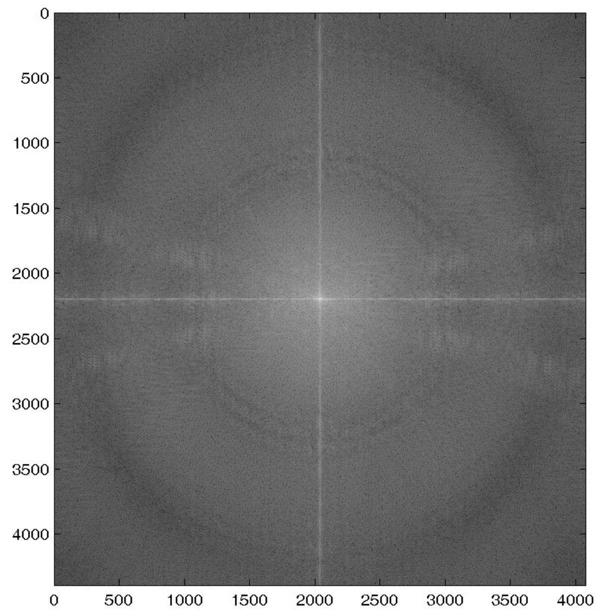

Figure 1: The magnitude of spectrum `p4a`, displayed in dB.

- ➪ Any MATLAB m-files you created for this project. There is no need to send me any data or image files that you generated. I'll run your m-files and generate your images myself. If for some reason I need additional files from a particular student, I'll request them separately.

Get an early start on this assignment if you want to do a good job.

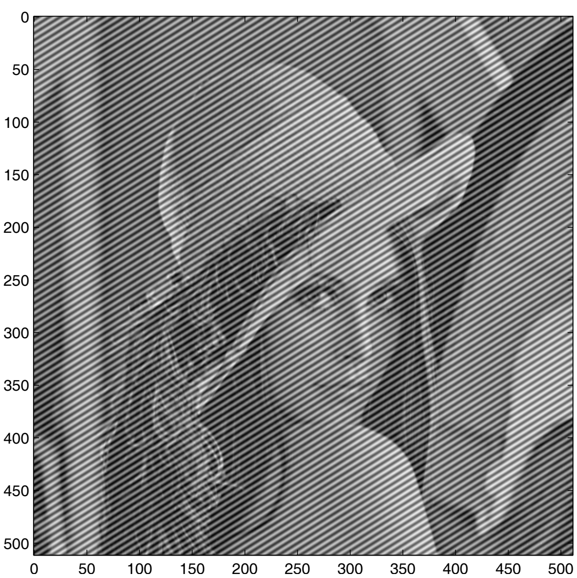

Figure 2: The noisy image `p4b`.