ENGI 4559 Digital Signal and Image Processing – Fall 2015
Project 2: Filtering in the Frequency and Z Domains (20%)

Part 1: Lowpass Filters and Highpass Filters (10%)

Download the image “lenna.png” from the website. It is an 8-bit grayscale, 512x512 image of lenna. Open the “lenna” image in Matlab, compute its DFT, and separate the Fourier spectrum and the phase angle using the functions seen in class. Then, design two frequency-domain filters for this image, a blurring filter and a sharpening filter. These should be filters that do not cause ring effects we saw in the lecture slides. Moreover, for the sharpening filter, do not simply extract the edges but actually sharpen the image; in other words, do not completely discard the grayscale intensity information of the image as in the lecture slides.

Discuss your work. What shape or equation do your filters have, and why?

Part 2: Notch Filters (5%)

So far, the frequency-domain filters we’ve used operate over the entire frequency spectrum of the image. There are times when we would prefer to affect small regions of the frequency spectrum, while leaving the rest of the spectrum unchanged. Such filters are called *notch filters*, and you will now design one.

Download the image “car.png” from the website. It is the 8-bit grayscale, 428x544 image of a car scanned from a newspaper and showing a Moiré pattern, which we saw in class. Open the image in Matlab, compute its Fourier Transform, and separate the Fourier Spectrum and the Phase Angle using the functions seen in class. Display the spectrum image. Do you notice anything unusual? Design a custom filter to correct this problem by blocking the minimal amount of the spectrum.

Discuss your work. Which regions of the frequency spectrum did you design your filter to block, and why? Could you further improve the quality of the image by filtering out other regions? Why or why not?

Part 3: Running-Sum Filters (5%)

As we saw in class, a running sum can be expressed in the spatial domain as the difference of two step functions:

$$s(n) = u(n) - u(n-L)$$

We also saw early in the semester what a 2D step function was like. An image version of this will simply be a white rectangle going from (0,0) to (L_x,L_y) in an otherwise black image. Create such a running sum filter, and display its Fourier spectrum. Create several of them with different values, and consider the different spectrums.

Compare the running sum's spectrum to the car's spectrum, and pick values of the running sum that correspond as best you can to the frequencies you want to eliminate. Then apply the filter, and recover the car image to confirm that the Moiré pattern has been removed. Discuss your work: what value of the running sum did you pick, did it work better than the notch filter used in Part 2, and why?

Important Note:

For parts 2 and 3 of the assignment, you do not need to write general algorithms to correct Moiré patterns in any image. Simply correct the problem for this one specific image.

Submission and evaluation:

Print out a report in which you present your results and your discussion on the questions asked in this project, and hand it to the professor in class. Do not hand in your Matlab code. This project must be done individually. Do not forget to write your name on your report.

For each part, half the marks will be given for demonstrating you did the work correctly, and half the marks will be given for a discussion demonstrating that you understand the work you did. Bonus marks (up to a maximum final grade of 100%) will be given for innovative and justified additional experiments and analysis. Penalty marks (up to 10% of the project mark) will be deducted for poor writing, spelling, or formatting of the report.