Revised Due Date: 11:59pm Monday Oct 22nd, 2018.

General Guidelines:
For this assignment you will submit via Moodle a carefully formatted assignment report. You do not have to submit your MATLAB code. Your report must be submitted as an Adobe Acrobat (.pdf) file. Your report should be well organized and address all questions below. (The last question is a bonus question for both 4422 and 5323.)

Points will be awarded in part for the clarity and formatting of your report. All plots must be clearly labelled and follow standard scientific conventions. Ensure that x and y axes are clearly labelled and units are specified. Please ensure that all plots are imported as *vector graphics*, not raster images.

1. (50 marks) For this question, use a photo taken under low light conditions, using a tripod. Convert the colour photo to grayscale using MATLAB's rgb2gray function.

   Assume that the power spectral density of natural images is isotropic and falls as $1/\omega^2$ and the noise spectral density is white. Then, as we showed in class, the frequency domain representation of the Wiener filter is

   $$H(\omega) = \frac{1}{1 + (\omega/\beta)^2} \tag{1}$$

   where $\omega^2 = \omega_x^2 + \omega_y^2$.

   (a) Show a picture of your image.

   Some useful MATLAB functions: figure, imagesc, axis, colormap.

   (b) Answer the following questions first for $\beta = 0.1$ and then for $\beta = 10$.

   i. Construct the optimal Wiener filter $H(\omega_x, \omega_y)$ in the frequency domain and then take the inverse Fourier transform to obtain the corresponding spatial filter $h(x, y)$. Truncate it at 0.1% of its maximum value. Show an image of your filter and a plot of its 1D cross-section.

   Some useful MATLAB functions: size, ind2sub, find, floor, ceil, abs, min, max, sum, sqrt, meshgrid, ifft2, fftshift, ifftshift, real, imagesc, axis, colormap, title, set, plot.

   ii. Apply the filter to your image in the space domain, using MATLAB function conv2 with 'same' boundary conditions. Record and report the execution time for convolution using MATLAB functions tic and toc. Show a picture of your filtered image.

   Some useful MATLAB functions: imagesc, axis, colormap, title, set, sprintf.

iii. Now apply the filter in the frequency domain. Record and report the execution time using MATLAB functions tic and toc. Include the time required to compute the Fourier transform and inverse Fourier transform of the image, but do not include the time required to construct the filter. Show a picture of your filtered image. Does it look the same as your image filtered in the space domain? (Hint: it should, except at the boundaries.)

   Some useful MATLAB functions: size, meshgrid, sqrt, fft2, ifft2, real, imagesc, axis, colormap, title, set, sprintf.

(c) For optimal efficiency, which method would you use for $\beta = 0.1$? Which method would you use for $\beta = 10$?

(d) Empirically estimate the crossover value for β for your image, i.e., at what value you would switch from filtering in the space domain to filtering in the Fourier domain.

   Some useful MATLAB functions: log, exp, semilogx, legend, xlabel, ylabel, set, interp1, fminsearch, sqrt

(e) What value of β produces the best perceptual results for your image? Show the image after filtering with this value of β.

   Some useful MATLAB functions: imagesc, axis, colormap, title, set, sprintf.

2. (50 marks) For this question, use a photo taken under normal lighting conditions. Convert the colour photo to grayscale using MATLAB's rgb2gray function.

   (a) Show a picture of your image

   Some useful MATLAB functions: figure, imagesc, axis, colormap

   (b) Measure and report the standard deviation of intensity values in your image.

   Some useful MATLAB functions: std.

   (c) Now create a noisy version of your image by adding zero-mean IID Gaussian noise with standard deviation matching your image. Show a picture of the noisy image.

   Some useful MATLAB functions: normrnd, size, figure, imagesc, axis, colormap.

   (d) Using the Wiener filter described in Q1, empirically estimate the value of β that minimizes the mean squared error (MSE). Show a plot of the MSE as a function of β. Report the execution time when β is set to its optimal value. Show a picture of the image after filtering with the optimal value of β.

Some useful MATLAB functions: log, exp, ceil, meshgrid, sqrt, sum, conv2, mean, semilogx, xlabel, ylabel, set, min, figure, imagesc, axis, colormap.

(e) Now repeat this experiment using an efficient implementation of an isotropic Gaussian filter, truncated at 0.1% of its maximum value and normalized to unit volume. Empirically estimate the value of the space constant σ that minimizes the mean squared error (MSE). Show a plot of the MSE as a function of σ. Report the execution time when σ is set to its optimal value. Show a picture of the image after filtering with the optimal value of σ.

Some useful MATLAB functions: double, rgb2gray, log, exp, ceil, sqrt, sum, conv2, mean, min, tic, toc, sprintf, normrnd, size, figure, semilogx, xlabel, ylabel, set, imagesc, axis, colormap.

(f) Compare your results and discuss the relative advantages and disadvantages of the Wiener and Gaussian filters for denoising.

3. (20 bonus marks for all). Using your noisy image from Q2, explore the use of bilateral filtering to improve results. Can you obtain superior MSE? How about perceptual quality?

   Some useful MATLAB functions: double, rgb2gray, log, exp, imbilatfilt, mean, min, tic, toc, sprintf, figure, semilogx, xlabel, ylabel, set, imagesc, axis, colormap, title.