Prof. A. Zakhor   Fall 2012
EE225b – Digital Image Processing
Assignment #5 – Image Restoration

Overview:

In this assignment, you will apply various image restoration techniques to restore an image corrupted by noise.

Assignment specifics:

Part A: On the website, there is a file `NoisyImg.bmp`, which is a 256-level gray image of a truck in a desert which has been corrupted by noise. First, apply median filtering to the given image and save the restored image as `Median.bmp`. Next, apply adaptive Wiener filtering to the given image and save the restored image as `Wiener.bmp`. Implement adaptive median filtering as described in class and apply it to the given image, and save the restored image as `AdaptiveMedian.bmp`. Devise your own technique by combining two or more of the previous techniques to improve the quality of the restoration. Compare and contrast your result with the previous three, and save the restored image as `ResultA.bmp`.

Part B: On the website, there is a second file `NoisyBlur.bmp`, which is a 256-level gray image of some gingo nuts, corrupted by blur and additive noise. The blurring filter is given by the `fspecial` command shown below. Use a combination of noise-removal and the iterative deconvolution algorithm described in class to restore the image. As the number of iterations increases, the image gets sharper but the amount of ringing will increase. Save three sample images, a blurry one with no ringing (`ResultB.bmp`), a sharper one with little ringing (`ResultC.bmp`), and even sharper one with a lot of ringing (`ResultD.bmp`). Be sure to indicate the number of iterations and the choice of β.

Please submit all your `.m` and `.bmp` files via email to the grader. Email submissions must be received before class on the due date. There should be an executable Matlab script `Lab8.m` which generates all your results.

Here are some helpful Matlab commands:

`wiener2`	Adaptive Wiener filtering
`medfilt2`	2D median filtering
`ordfilt2`	2D order-statistic filtering
`fspecial('disk', 3)`	Blurring filter
`imfilter, conv2, filter2`	Apply a 2D-filter to an image