CE7491 Graded Lab 2, AY2008-09 Semester 1

1. Spatial Noise and Noise-Filtering

Write a script that takes an images '`lena.tif`' and '`flower.tif`' and corrupts it with two types of noise, zero mean-Gaussian noise, and 'salt-and-pepper' noise (see figure below). You can choose the amount of noise you want. Now, filter the image using both a median filter and Gaussian filter as a way to reduce the two types of noise. The parameter of the filter and the window size can be decided by you, try to pick "good" ones. Compare your filtered images with the original images. Which of the two filters produces the best result in terms of MSE and which produces better results subjectively (i.e. looks better)?

(a) (b)

Figure 4a. Noise examples (a) Gaussian noise (b) salt-and-pepper noise.

Useful matlab function: `imnoise`

2. High-Boost Spatial Filtering

Implement a *function* that performs high-boost filtering. Your function should take an image, `I`, and a 3rd parameter, `A`, to control the amount of boosting to be performed as discussed in the lecture notes.

3. Low-Pass and High-Pass Butterworth Filter

Write a Matlab function that applies either the low-pass or high-pass Butterworth filter in the frequency domain to an image and returns the resulting filtered image. The function should accept four parameters: the image *I*, the "cut-off" of the filter *Do*, the exponent *n*, and a parameter to specify whether lowpass or high-pass filtering is to be performed.

In addition to generating the filtered image, plot your constructed Butterworth filter, *H*.

Useful matlab commands for plotting *H*: `mesh, colormap`

4. Noise Removal

Build a filter to restore the '`pepper_corrupt.tif`' image shown below. Note that you may not be able to restore it completely, but you can significantly improve the quality of the image.

'peppers_corrupt.tif'

5. Wiener Filter Restoration

Part a) Write a function that implements the "Wiener" filter in matlab (do not use `deconvwnr`). Note that the transpose operator, `H'`, takes the complex conjugate of a matrix. You can assume a constant power spectrum of the noise and the original image, and therefore your filter can use the Signal-to-Noise Ratio (SNR) as discussed in the notes.

Part b) The '`coins_blurred.tif`' image shown below has been blurred with a 'disk' filter of size 2. Try to recover this image using inverse filtering and *your* Wiener filter. You should see that Wiener filter is significantly better. Note that your return values may be complex and outside the range [0 255]. You should use only the real component and clip (not scale) your results to be between [0 255].

'coins_blur.tif'