Instructions:

You can see the code for doing some basic MATLAB operations with images in the Implementation Notes section.

Each student is required to submit an individual report of the project.

Problem 3: (30 points, 3 pages max)

In this exercise, you will learn about different noise removal techniques in spatial and frequency domains

You will need sound understanding of probability and random variables. You can revise these concepts by going over Chapters 4-6, 'Probability, Random Variables and Stochastic Processes' by Papoulis.

In MATLAB, you can use `randn` to generate samples from a uniformly distributed random variable(or noise). Using the command `X1=randn(1,5)` you can draw 5 samples from a uniformly distributed random variable with mean 0 and variance 1 (denoted as $\mathcal{N}(0,1)$).
Similarly, you can generate a 256 x 256 noisy image by `N=randn(256,256)`. We can think of this noise as modeling all kinds of noisy/erroneous steps in image acquisition(such that randomness of photon count, noise of camera electronics, CCD etc).

P3.1: Load the original noiseless cameraman image again(denoted f). Normalize it so that its range is between 0-1. (`img=double(img)./256.0`). Generate a 256 x 256 noisy image N_1 with variance $\sigma = 0.2$ and 0 mean using `randn`. Add this noise to the cameraman image and display the noisy image f_{n1}. Verify that the variance of the noisy image

$$\sigma^2 = \frac{1}{N}\sum_x\sum_y(f_{n1}(x,y) - \overline{f_{n1}(x,y)})^2$$

is indeed close to 0.2.

Now generate another noisy image $f_n 2$ using the same procedure. Imagine that this is the second image captured using the same system, only its noise is different. We can reduce noise by averaging the images f_{n1} and f_{n2}.

(a) Provide a Mathematical derivation of how you can reduce noise using images f_{n1} and f_{n2}.

(b) Apply this procedure to your two images to come up with $\hat{f}$ with reduced error. Write code to calculate σ for $\hat{f}$ and show that it is less than the individual images' noise.

P3.2:
Now assume that you have a noisy image f_n, with a histogram in the range $T1$ and $T2$.

(a): Write equation to stretch the histogram of f_n such that the stretched image, g_n, has histogram in the range 0-255.

(b): How is the noise in g_n, i.e σ_g related to σ_f? Provide a derivation.

(c): Take the image you used in contrast stretching section of problem 2. Add gaussian noise with $\sigma = 0.2$ and $\mu = 10$ as in P3.1. Now apply the equation derived in P3.2(a) to get image g_n with histogram stretched. Confirm that the noise in g_n and f_n is indeed related by your relationship derived in part (b).

P3.3: Take the noisy image f_n1 used in P3.1 and de-noise it using an edge-preserving filter for gaussian noise. Please note that you will get no credit for a simple mean filter.

Best of luck!

Implementation Notes

MATLAB has an extensive help available online. If some MATLAB command is missing here, or for more details on any of these commands, you can use MATLAB's help at www.mathworks.com

```
f=imread('MyImageFileIn.tif');% Read an image file

imwrite(f,'MyImageFileOut.tif');% Write an image file

imagesc(f);colormap(gray);colorbar;% Display an image
imshow(f);% Display an image

Filt=[0,1,0;1,0,1;0,1,0];
g=imfilter(f,Filt);% Filter an image f to produce g

F=fft2(f);
F=fftshift(F);
imagesc(abs(F))% Find 2D FFT F(u,v) of an image f(x,y)
 % Sometimes the dynamic range of FFT is too large, so you may
see one or two impulses only while actually there is more.  To
see such a FFT, use imagesc(log(abs(F))) to squeeze the dynamic
range.

fr=ifft2(ifftshift(F));
imagesc(abs(fr))% Find Inverse 2D FFT fr(x,y) of a 2D FFT
F(u,v)
```

surf(X,Y,Z)% Plot a surface $Z = f(X,Y)$

```
imhist(f) or hist(f(:))% Plot histogram of image f
```
[1]

[1]

Assistant Professor, FAST-NU
sadiq@purdue.edu