Homework #5

Due Date: Mar. 11, 2003

1. [10 each] Lim, Problem 7.6, 8.5, 8.15, 8.18 (be sure to look at the frequency domain data).

2. [50] In this problem, you will implement and perform histogram equalization on an image. Please down load hw5_image.mat from the course web site. Please hand in copies of all images and of your Matlab code.
    a. Use Matlab's `hist` function to determine the histogram of the input image using 256 bins (you will have to use `head(:)` to convert image into a column vector). Plot the histogram function and the cumulative histogram function.
    b. Let the desired histogram function be triangle function – find and plot the desired cumulative histogram function.
    c. Derive a transformation $g = T[f]$ to convert pixel values of the input image (f) to the equalized image (g).
    d. Plot the histograms of g and f in a single graph. Do the same for the cumulative historgrams.
    e. Display the images g and f using `imagesc`. Was performing equalization a good thing to do for this image?

3. [60] In this problem, you will implement and perform gradient-based edge detection. Use the same image as in Problem 2. Please hand in copies of all images and of your Matlab code.
    a. Using the Sobel function, determine and display (using `imagesc`) the x and y gradients of the image.
    b. Determine $|\nabla f|$ and threshold at a level of 5% of the maximum gradient value to produce a binary map of edges. Display using `imagesc` and use `colormap(1-gray)` to invert the intensity scale (saves toner).
    c. Write an edge thinning procedure as described in class (keep only edges that are larger than the surround pixels in the x direction or in the y direction). Apply this to the edge image and display.
    d. Add noise to the original image using `0.1*randn(size(head))`.
    e. Repeat steps b. and c.
    f. Now smooth noisy image with a Gaussian filter with $\sigma = 1$ (see Eqn. 8.17). This filtering can be done in the Fourier domain or image domain.
    g. Repeat steps b. and c.

Just for fun: Try different edge detection methods using Matlab's `edge` command on both the original and noisy images. Both LOG and Canny methods allow the user to select filtering values (σ).