Homework #2
Due Tu. 3/04

You must turn in your code as well as output files. Please generate a report that contains the code and ouput in a single readable format.

Getting Started

- You may want to download Irfanview image viewing software. It handles pretty much any image type, lets you convert, and provides batch processing.

  `http://www.irfanview.com/`

- Download the sample images from the class website.

  `http://www.ee.unlv.edu/~b1morris/ecg782/hw/hw02`

1. Spatial Domain Filtering

   The following question operates on the `city.jpg` image.

   (a) Perform image smoothing using a 7×7 averaging filter and a Gaussian filter with $\sigma = 0.5$ and 3. Compare the outputs.

   (b) Perform edge enhancement using the Sobel operator (Matlab's default parameters). Repeat using the Laplacian and Laplacian of Guassian operators. Compare the outputs

2. Frequency Domain Filtering

   The following question operates on the `city.jpg` image.

   (a) Find the Fourier transform of the image. Be sure to center the frequencies.

   (b) Perform image smoothing in the frequency domain using the filters defined in the previous problem. Compare the output images from the two methods (spatial and frequency) and the time for operation.

   (c) Perform edge enhancement using the filters defined in the previous problem.

   (d) Define a lowpass filter in the frequency domain with radius of 1/4 the height. Show the result. Repeat with a similar sized Guassian and compare the results. Give the σ parameter you used and show the output transform image.

   (e) Repeat with a rectangular filter with the same dimension as the ideal lowpass. Compare the results between the ideal filter and the rectangular approximation.

3. Canny Edge Detection

   (a) Give the convolution kernels for determining the gradient. You may examine the function `gradient.m` to help with the explanation. (It may be easiest to apply the `gradient` to an impulse and inspect the results.

   (b) Implement the simplified version of the Canny edge detector (single scale). The syntax of the function should be

   $$\texttt{[E,M,A]=canny(I,sig,}\tau\texttt{)},$$

   where `E` contains the detected edges, `M` the smoothed gradient magnitude, `A` contains the gradient angle, `I` is the input image, `sig` is the σ parameter for the smoothing filter, and `tau`$= [\tau_h, \tau_l]$ is the two element vector containing the hysteresis thresholds.

   See Algorithm 6.4 for non-maximal suppression and Algorithm 6.5 for hysteresis thresholding. (It may be more efficient to implement the hysteresis as edge tracking)

(c) Apply your Canny detector on `wirebond_mask.tif` using $\tau = [0.8, 0.6]$ with the following values for $\sigma^2 = [0.5, 1, 3]$. Show your results in a (1,3) subplot. Invert the color, white for 0 and black for 1, to save ink. Discuss how the choice of σ affects the results.

(d) Apply your Canny detector on `city.jpg`. Adjust the σ and τ parameters as you see fit. Display the resulting edges and the parameter settings used.

4. Corner Detection

(a) Consider the symmetric 2×2 matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

By finding the roots of the characteristic equation,

$$\det(\lambda I - A) = 0,$$

show that the eigenvalues of A are given by

$$\lambda = \frac{\mathrm{tr}(A) \pm \sqrt{\mathrm{tr}(A)^2 - 4\det(A)}}{2}.$$

(b) Compute the feature detection autocorrelation matrix A for the checkerboard image. Use a simple 3×3 box filter for the window function. Show the image with an overlay of the keypoint locations, defined as those points with $\lambda_{min} > \tau$ with τ 80% of the maximum λ_{min} value over the whole image. Also, draw a vector indicating the keypoint orientation (scaled by magnitude λ_{min}).

The angle of the principle eigenvector of A is given by

$$\phi = \frac{1}{2}\arctan\left(\frac{2b}{a-c}\right).$$

(c) Repeat for the fingerprint image this time using τ as 20% of maximum λ_{min}.