Lab 1 (Weeks 3,4): Canny Edge Detection

Each task in the lab exercise is worth 2% of your final unit grade (total 10%). *A task is only considered complete if you can demonstrate a working program and show understanding of the underlying concepts. Note that later tasks should reuse code from earlier tasks.*

In this laboratory exercise, you will implement the Canny Edge detector (without hysteresis thresholding) using matlab. Make sure you run your code on the test images provided to see if the results are as expected before attempting to get marked.

References:
Canny edge detector overview: http://en.wikipedia.org/wiki/Canny_edge_detector

Resources:
Test images are located in the lab1 folder. These should be used to test your code.

Task 1: Implement Gaussian blur

Write a program that performs Gaussian blur on an input image using the following 5x5 kernel, where B is the blurred version of input image A. You code should not use the conv function in matlab. Show your result on screen.

$$\mathbf{B} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

Task 2: Calculate image gradients

Write a program that does the following in order. You should reuse your Task 1 code

1. Load an image from the hard drive as a grayscale (single channel) image.
2. Blur the image using a 5x5 Gaussian filter
3. Calculate the gradient of the blurred image in the x and y directions using a 3x3 Sobel filter
4. Check the x and y gradient values to make sure they are correct

Task 3: Calculate gradient magnitude

Extend your program to calculate the gradient magnitude. Gradient magnitude G is a function of the gradients in the x and y directions (Gx, Gy)

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

Show the gradient magnitude on screen as a grayscale image

Task 4: Calculate gradient orientation

Extend your program to calculate the gradient orientation as follows

$$\Theta = \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right).$$

Note that gradient orientations should be rounded to the nearest 45-degrees (Compass directions: North, North-East, East etc). Rounding can be performed using a combination of integer arithmetic and the floor() function. Display the gradient orientations on screen to check that they are correct. You may want to use colours to illustrate the different orientations.

Task 5: Non-maxima Suppression and thresholding

Finally, extend your code to perform non-maxima suppression in order to “thin” the edges. A detailed summary is provided here. The idea is to zero any pixel that is not greater in terms of gradient magnitude than both pixels on either side of its gradient orientation.
http://en.wikipedia.org/wiki/Canny_edge_detector#Non-maximum_suppression

Threshold the non-maxima suppression results. The final output should be a black and white (binary) image showing the edge pixels. Test your code against the test images.