CS 6643 **Fall 2019**
Project 1: Canny's Edge Detector **E. K. Wong**

Total number of points = 100. This is a one-person project. No teams allowed. You can discuss with your classmates how to do the project but everyone is expected to implement and submit their own project.

Project Description: Implement the *Canny's Edge Detector* as described in the lecture slides. The Canny's Edge Detector consists of four steps: *Gaussian smoothing, gradient operation, non-maxima suppression* and *thresholding*. The input to your program is a grayscale image of size *N* X *M*. Use the 7 x 7 Gaussian mask as shown below (on page 2) for smoothing the input image. Use the center of the mask as the reference center. If part of the Gaussian mask goes outside of the image border, let the output image be undefined at the location of the reference center. Note that the entries in the Gaussian mask do not sum to 1. After performing convolution (or cross-correlation), you need to perform normalization by dividing the results by the sum of the entries (= 140 for the given mask) at each pixel location. Instead of using the Robert's operator, use the Sobel's operator for gradient operation. If part of the 3 x 3 mask of the operator goes outside of the image border or lies in the undefined region of the image after Gaussian filtering, let the output value be undefined. For the third step, follow the procedure in the lecture slides for non-maxima suppression. At locations with undefined gradient values and at locations where the center pixel has a neighbor with undefined gradient value, let the output be zero (i.e., no edge.) For the fourth step, use double thresholding to threshold the gradient magnitude $N(i,j)$ after non-maxima suppression into a binary edge map $E(i,j)$. Set up a low threshold T_1 and a high threshold T_2 so that $T_2 = 2T_1$.

- If $N(i,j) < T_1$ let $E(i,j) = 0$
- If $N(i,j) > T_2$ let $E(i,j) = 255$
- If $T_1 \leq N(i,j) \leq T_2$ let $E(i,j) = 255$ if pixel (i,j) has an 8-connected neighbor (i',j') with gradient magnitude $N(i',j') > T_2$ AND the gradient angles of (i,j) and (i',j') differ by 45° or less (i.e., $|\theta(i,j) - \theta(i',j')| \leq 45°$); otherwise, let $E(i,j) = 0$.

Design your program in a modular fashion and create separate program functions for the four steps: *Gaussian smoothing, gradient operator, non-maxima suppression* and *thresholding.* Generate the following *normalized* image results with an output range of [0,255]. For image locations with undefined values as described above, replace with value 0. The output images should be of the same size as the original input image. (1) Normalized image result after Gaussian smoothing. (2) Normalized horizontal and vertical gradient responses (two separate images.) To generate normalized gradient responses, take the absolute value of the results first and then normalize. (3) Normalized gradient magnitude image. (4) Normalized gradient magnitude image after non-maxima suppression. (5) Binary edge map using double thresholding.

You can use Python, C++/C, Java or Matlab to implement your program. If you would like to use a different language, send me an email first. You are not allowed to use any built-in library functions to perform any of the steps that you are required to implement, including the convolution (or cross-correlation) operation. The only library functions you are allowed to use are: *reading* and *writing* image files, *matrix* and *vector multiplications* and other *commonly used math operations*.

Testing your program: Two grayscale test images of size *N* X *M* in bitmap (*.bmp*) format will be provided on NYU Classes for you to test your program. A 15-min time slot will be set up during class time for you to run your program on a different test image that will be provided in class. (Details to be announced later.)

Hand in the following on NYU Classes by the project due date:

(a) A text file that contains the source code. Put comments in your source code to make it easier for someone else to read your program. Points will be taken off if you do not put in comments.
(b) Output image results (1) to (5) as described above for each test image provided.

(c) A PDF file that contains: (i) File name of your source code. (ii) Instructions on how to run your program and instructions on how to compile your program if your program requires compilation. (iii) Output image results (1) to (5) for all test images. (v) The source code of your program.

Note that you need to hand in the source code and image results as separate files (as generated by your program) and in the PDF report. You can just copy-and-paste the image results and source code onto a Words document first and then convert into a PDF file.

7×7 Gaussian mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1