CS 6643 **Spring 2023**
Project 1: Canny's Edge Detector **E. K. Wong**

Total number of points = 100. You can work on the project by yourself or work in a team of two. You can discuss with your classmates on how to do the project but every team is expected to do their own coding and turn in their own report, source code and output image results.

Project Description: Implement the *Canny's Edge Detector* with the following four steps: *Gaussian smoothing, gradient operation, non-maxima suppression* and *thresholding.* The input to your program is a grayscale image of size $N \times M$ (rows × cols.) Use the 7 × 7 Gaussian mask below (page 2) for smoothing the input image. Use the center of the mask as 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. For gradient operation, use the following masks to compute gradients at 0, 45, 90 and 135 degrees[1]:

-1	0	1
-2	0	2
-1	0	1

$g_0: 0^\circ$

0	1	2
-1	0	1
-2	-1	0

$g_1: 45^\circ$

1	2	1
0	0	0
-1	-2	-1

$g_2: 90^\circ$

2	1	0
1	0	-1
0	-1	-2

$g_3: 135^\circ$

The normalized edge magnitude is then taken as the maximum of the absolute values of the responses from the four masks, and then divide by 4; i.e.,

$$M(i,j) = \frac{\max_k \{|h_k(i,j)|, k = 0,1,2 \text{ and } 3\}}{4}$$

where $h_k(i,j)$ is the response generated by using mask g_k in the above. Let the output value be undefined if part of the 3 × 3 mask goes outside of the image border or lies in the undefined region of the image after Gaussian filtering. For the third step, follow the procedure in the lecture slides and perform non-maxima suppression to the normalized edge magnitude image, but let the quantized angle equals to the index of the mask that produces the maximum response; i.e., quantized angle

$$\varsigma(i,j) = \arg\max_k \{|h_k(i,j)|, k = 0,1,2 \text{ and } 3\}$$

At locations with undefined values or if the center pixel has a neighbor (in the direction of the quantized angle) with undefined value when performing non-maxima suppression, let the output be zero (i.e., no edge.) For the fourth step, perform simple thresholding to the magnitude image after non-maxima suppression and produce three binary edge maps by using thresholds chosen at the 25th, 50th and 75th percentiles of the gradient magnitudes after non-maxima suppression (exclude pixels with zero gradient magnitude when determining the percentiles.)

Produce the following image results[2] and histograms in your outputs: (1) image result after Gaussian smoothing, (2) normalized magnitude image, (3) normalized magnitude image after non-maxima

[1] The masks came from the Robinson compass mask for edge detection.

[2] For pixel locations with undefined values as described above, replace with value 0 in the outputs. The output images should be of the same size as the original input image after replacing undefined values with 0's.

suppression, (4) binary edge maps for thresholds chosen at the 25th, 50th and 75th percentiles, and (5) a histogram of the normalized magnitude image after non-maxima suppression.

You can use Python, C++/C, Java or Matlab to implement your program. If you would like to use another 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. You are allowed to use library functions for the *reading* and *writing* of image files, computation of *histograms, matrix* and *vector multiplications*, computation of *percentile values*, and other commonly used math operations.

Testing your program: Three grayscale test images of size $N \times M$ in bitmap (*.bmp*) format will be provided on *Brightspace* for you to test your program.

Hand in the following on Brightspace by the due date:

(a) Your source code file. 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 have comments.

(b) The output image files (in *.bmp* format) generated by your program as described in (1) to (4) above. There should be a total of six output image files for each test image.

(c) A PDF file that contains the following: (i) instructions on how to run your program. If your program requires compilation, instructions on how to compile your program should also be provided; (ii) the histogram for the normalized magnitude image after non-maxima suppression, and (iii) copy-and-paste the output images and your source code onto the PDF file[3]; also, put the threshold values chosen at the 25th, 50th and 75th percentiles below each edge map.

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

[3] This is to make it easier for us to grade your project. It is in addition to the source code file and output image files that you need to submit separately as described in (a) and (b) above.