CS 558: Homework Assignment 1- Edge Detection

Enrique Dunn
Department of Computer Science
Stevens Institute of Technology
edunn@stevens.edu

Collaboration Policy.   Homeworks will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems. Use of the Internet is allowed, but should not include searching for previous solutions or answers to the specific questions of the assignment. I will assume that, as participants in a graduate course, you will be taking the responsibility of making sure that you personally understand the solution to any work arising from collaboration.

Late Policy.   No late submissions will be allowed without consent from the instructor. If urgent or unusual circumstances prohibit you from submitting a homework assignment in time, please e-mail me explaining the situation.

Submission Format.   Electronic submission on Canvas is mandatory. Submit in a zip file, a pdf file:

- source code (excluding libraries),

- resulting images,

- at most one page of text explaining anything that is not obvious.

Also include the code and output images separately.

Problem 1.   Download the images provided on the course website. You must implement an edge detector comprising the following steps:

1. Gaussian filtering of the input image. Allow the user to specify the σ of the Gaussian function (do not forget the normalizing constant). Make sure that the filter size is large enough and that the filter coefficients sum to 1. Try values of σ ranging from 1 to 10 and submit results using two different values.

2. Gradient computation using the Sobel filters. Pick a threshold for the gradient magnitude so that the number of remaining edge pixels is similar to the examples in the slides. Use your best judgement. Submit an image of the gradient magnitude for some value of σ for each input image.

sigma = smoothing filter

3. Perform non-maximum suppression. For simplicity, do NOT use interpolation for this step. Instead, assign the edge orientation to one of four possible cases: horizontal, vertical and the two diagonals and compare the gradient magnitude of the current pixel with the two neighbors orthogonal to that orientation. For example, if the edge is approximately vertical, compare the gradient magnitude of the pixel to its left and right neighbor. If it is larger, the pixel should be kept as an edge pixel. (See Slide 43 for how to compute the orientation of an edge. It is not necessary to follow Slide 60 for non-maximum suppression. You can get partial credit by classifying edges as horizontal or vertical to simplify the requirement.) Submit the output of this process using the edge image of the previous step as input.

Requirements and notes.

- You can use any programming language. You may have to explain the homework to me in person, if I am not familiar with your choice.

- You are allowed to use image reading and writing functions, but not filtering, edge detection or other image processing functions. You can use such functions to verify that yours are correct, but they cannot be included in your submission.

- If you are having trouble reading the images in your programming environment, send me an email.

- Your code must include a main function that takes as arguments the filenames and parameters without having to modify the code, a filtering function that applies a filter to an image and returns an output image of the same size as the input, and a function that implements non-maximum suppression.

- Your filtering operations should replicate boundary pixels when the filter window falls out of bounds. For example, if the leftmost pixel has intensity 34, then 34 should be used for any intensity value to its left.

- Submit your code parameterized to run for one image with fixed parameters.