ECE/OPTI 532, Fall 2020

Homework 2 Assignment

Due Tue. Sept. 22 at 5:00 p.m. MST

Write a computer program to implement the Rodriguez edge detector, which is a modified 3 x 3 gradient operator that computes an edge map $E(r,c)$ from a grayscale input image $I(r,c)$ as follows:

$$I_y(r,c) = \text{median}\{I(r,c-1), I(r,c+1), I(r-1,c-1), I(r-1,c), I(r-1,c+1)\} - \text{median}\{I(r,c-1), I(r,c+1), I(r+1,c-1), I(r+1,c), I(r+1,c+1)\}$$

$$I_x(r,c) = \text{median}\{I(r-1,c), I(r-1,c+1), I(r,c+1), I(r+1,c), I(r+1,c+1)\} - \text{median}\{I(r-1,c-1), I(r-1,c), I(r,c-1), I(r+1,c-1), I(r+1,c)\}$$

$$\text{median}\{S\} = \text{middle value after sorting the elements of S in numerical order}$$

$$G = \left|\vec{\nabla} I\right| = \sqrt{I_x^2 + I_y^2} \,/\, 1.2$$

$$E(r,c) = \begin{cases} \text{edge}, & \text{if } G(r,c) \geq \text{ thresh} \\ \text{non-edge}, & \text{otherwise} \end{cases}$$

Note that the 1.2 scaling factor would normally be omitted, but let's include it so that we can do a fair comparison with the Sobel operator.

For the median calculation, you may use a built-in function or library function, or you may implement your own bubble sort or other sorting algorithm. For example, if you're using C, then you may want to use the `qsort()` function.

Do not perform non-maximum suppression.

Run the program on the `horse.png` image, using $\text{thresh} = 60$.

Turn in the following:

1. the source code in a format that the grader can compile and run it
2. the output edge map as an image or PDF file
3. the numerical values of the JJR gradient magnitude and edge map for Region 1, defined by $205 \leq r \leq 209, 182 \leq c \leq 186$ (assuming r and c start at 0), and the Sobel gradient magnitude and edge map for same region
4. the numerical values of the JJR gradient magnitude and edge map for Region 2, defined by $347 \leq r \leq 351, 350 \leq c \leq 354$ (assuming r and c start at 0), and the Sobel gradient magnitude and edge map for the same region