

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 JJR 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:

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

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

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

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

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

Note that the scaling factor would normally be omitted, but let's include it so that we can do a fair comparison with the Sobel operator. **Although each partial neighborhood has an odd shape whose centroid is between pixels, the median value will be one of the actual pixel values at a pixel center, so we should arguably just divide by a pixel displacement of 2.**

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.

Also implement the Sobel edge detector for comparison; be sure to include the 1/8 scaling factor in the Sobel gradient magnitude calculation, so that we can use the same threshold for both edge detectors.

Do not perform non-maximum suppression.

Run the program on the `horse.png` image, using `thresh = 60` **for both the JJR edge detector and the Sobel edge detector. (A better way to compare would be to adjust the threshold for each detector so that the total number of edge pixels is equal, but let's keep it simple and just use the same fixed threshold.)**

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 the region defined by $349 \leq r \leq 353$, $349 \leq c \leq 353$** (assuming r and c start at 0), and the Sobel gradient magnitude and edge map for same region

Notes

The horizontal derivative involves subtracting the medians of the following two partial neighborhoods:

A	B
D	
G	H

B	C
	F
H	I

The vertical derivative involves subtracting the medians of the following two partial neighborhoods:

A	B	C
D		F

D		F
G	H	I

Output Values

JJR gradient magnitude values:

1.58	?	?	?	74.95
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?

JJR edge map:

-	?	?	?	edge
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?

Sobel gradient magnitude values:

0.73	?	?	?	78.51
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?

Sobel edge map:

-	?	?	?	edge
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?