

Assignment 3

The purpose of this homework is to experiment with edge detection and openCV

Your program should be able to do the following:

1. Edge Detection [5 points]
 - Add edge detection for grey level images to your choice of options.
 - Use Sobel operator (3by3) and (5x5) to compute dx and dy, compute gradient amplitude, compute edge direction
 - Your program should
 - operate within specified ROIs (up to three ROIs as in previous assignments)
 - generate display of the amplitude of the gradient operator as intensity image
 - generate binary edge image derived from amplitude of the gradient operator by thresholding.
 - generate binary edge image by further thresholding the above output using direction information (e.g. display only horizontal (+/- 10 degree) edges or 45 degree +/- 10 degree edges etc)
 - Test your program on some grey level images
 2. Utilization of OpenCV [5 points]
 - replicate first part of this assignment using OpenCV calls and compare performance
 - utilize Canny module of OpenCV and compare to Sobel results
 - implement histogram equalization using OpenCV and compare to you histogram stretching implementation
 - Implement Otsu algorithm using OpenCV and compare to your optimal thresholding algorithm
 - Combine operations by applying histogram equalization only to foreground as determined by Otsu.
- Make sure that you have complete report for this assignment (not just few comments).
 - Include input and output images (use several gray level images as appropriate).
 - Discuss performance of edge detection on grey level images.
 - Discuss performance and utility of OpenCV
 - Discuss performance of Canny edge detector (vs Sobel) as edge detection.
 - Discuss performance of combined operations

How to submit

- Submit paper report in class on the due date
- See TA help desk for instruction on program submission and testing.

Note one Sobel 5x5 mask (Dx). Dy is the transpose

-5	-4	0	4	5
-8	-10	0	10	8
-10	-20	0	20	10
-8	-10	0	10	8
-5	-4	0	4	5