Name offile l061.cpp

Your main should call only the folllowing method: part1(...) (with or without parameters.

Create part1(...) method that does the following:

1) read the file image.ppm (image will contain an image of coins)

2) create the file imagef.ppm by applying the complete canny edge detection from previous lab (optimize the values to work best for the easy file I provided) Optional: you may also create the imageg.ppm. image1.ppm and image2.ppm for debugging purposes.

3) use the edges you found and gradient direction (not rounded!!!) to implement voting and create the imagev.ppm that displays the result of the phase 1 voting for centers.

4) use the results of voting and a threshold value to pick good candidates for circle centers and create imageCC.ppm file that will display the original image and on the original image will do a filled circle of radius 4 of color red for each candidate for circle center. (a filled circle of radius 4 can be obtained by drawing 4 circles with that center of radius 1,2,3,4)

5) **allow the tester to change the lower and upper threshold by using command line arguments as follows:**

Ex1: ./l061 -L 110 -H 200 -F myimage.ppm -TC 300

- the line above will use 110 and 200 for low and high threshold for th canny edge and the 300 will be the threshold value used for deciding if a pixel is a

candidate for a center or not (so if a pixel received >=300 votes it means is a high chance that the pixel is a center)

Ex2: ./l061

The line above has no parameters which means it will use the values for lower and higher threshold that you optimized for your image, also will use the threshold value you optimized for the first image and it will apply it to the file image.ppm.

****
****

Hints: use besenhamalgorithm to vote on the direction of the radiant. Also a good improvement would be to calculate the intersection points on the extremes so the line i more close to reality compared to using 2 very close points in which case the linewill be off.

Submit the following document after your code runs on the easy image I provided (which you will transform in p3 ppm file using teh convert tool on gnu linux:

Project 6 Coin Detection Part 1.docx

Your code will be tested against any piece of the easy image provided not just against the entire image.