Lab 7: Computer Vision

The task for this assignment is to implement a set of shape detection algorithms. Your functions will receive a 3D structure of size 640x480x3 representing the RGB channels of a 640x480 pixel image and it should return information about the content of if.

Problem 1: slope-intercept equation line detection

You have to develop a program to detect lines in images. Luck is in your side and in this exercise all lines will be black with a white background. The biggest problem you will find are some random figures have the same color as the line.

For this task you are to complete one function:

- For C:
    - Line detect_slope_intercept(unsigned char ***image)
- For python:
    - def detect_slope_intercept(image):

The function receives:

- A tridimensional structure called image containing the values of the RGB channels of the image.

The function must return:

- a Line structure with the m and b fields containing the parameters of the line function (mx+b=y).

Considerations:

- The voting space should be size exactly 2000x2000
- m is bounded to -10 and +10
- b is bounded to -1000 and +1000
- The margin for correctness will be +/-.1 for m and +/-3 for b during grading

Problem 2: normal equation line detection

Everything worked fine until you received a few images with almost vertical lines, this caused the function to start failing. Now you need to implement a function to include these special cases.

For this task you are to complete one function:

- For C:
    - Line detect_normal(unsigned char ***image)
- For python:
    - def detect_normal(image):

The function receives:

- A tridimensional structure called image containing the values of the RGB channels of the image.

The function must return:

- a Line structure with the theta (in radians) and r fields containing the parameters of the line function (r = x * cos(theta) + y * sin(theta)).

Considerations:

- The voting space should be size 1800x1800
- Theta is bounded to 0 and Pi
- r is bounded to -900 and +900
- The margin for correctness will be +/-.1 for theta m and +/-3 for r during grading

Problem 3: counting quarters

For this problem you have to create a program to count the number of circles in an image. All circles have the same radius (in pixels) and they will be complete (no overlapping between circles or partially outside the image)

For this task you are to complete one function:

- For C:
  - Line detect_circles(unsigned char ***image)
- For python:
  - def detect_circles(image):

The function receives:

- A tridimensional structure called image containing the values of the RGB channels of the image.

The function must return:

- an integer with the number of circles detected in the image.

Considerations:

- The voting space should be size 640x480 (size of the image)
- The radius is always 30 (all circles have the exact same footprint)

Instructions:

1. Login to your server
2. Go to the EECS348_Labs directory
   ```
   $ cd EECS348_Labs
   ```
3. Download the Lab code:
   ```
   $ ./getLab Lab6 <Language>
   Fetching Lab6 code
   Unpacking code
   done!
   ```
4. Edit student_code file:
   ```
   $ nano Labs/Lab6/src/student_code.c
   ```

5. Test the Lab:
```
$ ./testLab Lab6
<All Test Results Successful>
```
6. Submit your code:
```
$ ./submitLab Lab6
Creating submission file
Uploading file
Submission finished
```