Coding Assignment – CSE 409

Submission Deadline: September 2, 2022, Friday, 11:55 PM (Platform – Moodle)

Submission Instructions:

1. Create a folder with your student id, ex. 1705xxx.
2. Put the source files, an input file and three corresponding output images in the folder.
3. Zip the folder and upload it on Moodle at the appropriate assignment submission link.

Allowed Programming Language: Any. For c/cpp you can use bitmap header file shared with you for CSE 410 Offline 2. But in case of other languages, you need to find and use the appropriate bitmap image generation library.

Tasks:

Given the two end points of some lines, you have to draw them using the basic Midpoint Line Algorithm, using Unweighted Area Sampling Antialiasing technique and using Weighted Area Sampling Antialiasing technique. You can assume that the background color is white and the color of the lines is black.

For the Unweighted Area Sampling technique, use the overlapping box count approach to approximate the overlap in area.

In the Weighted Area Sampling technique, make sure that the intensity of a pixel varies based on the distance between its center and the center of a line. In this regard, you can experiment with different functions and use any one that shows this property.

Note that, your program should be able to handle lines with any slope. You can exchange the start and end points and use the idea of reflection wrt different lines (e.g. x-axis, y-axis, y=x line etc.) as appropriate. You do not need to handle colors, drawing grayscale images will suffice.

Marks Distribution:

1. Draw lines using Midpoint Line Algorithm (6 marks)
2. Apply Anti-Aliasing using
    a. Unweighted Area Sampling. (7 marks)
    b. Weighted Area Sampling – Gupta-Sproull Method (7 marks)
3. Bonus: Handling colored lines (3 marks)

Input Format:

Read inputs from a file titled "input.txt". The first line of the input file will contain two integers W, H indicating the dimension of the bitmap images to be generated. The next line will contain an integer N denoting the number of lines to be drawn. Each of the next N lines will contain four integers denoting the (x_{start}, y_{start}), (x_{end}, y_{end}) points of each line.

Limits:

- $0 <= N <= 25$
- $1 <= W <= 800$; $1 <= H <= 600$
- For any x, y: $0 <= x < W$, $1 <= y < H$.

Sample Input:
200 300 // image dimension will have 200 pixel width, 300 pixel height
1 // Only 1 line is to be drawn
4 5 150 200 // Two end points of the line are (4, 5), (150, 200)

If you choose to do the bonus task, after the coordinates of each line, add an additional line denoting its color. For example,
4 5 150 200
255 0 0 // the line has red color

Output Format:
Generate three bmp images "1_R.bmp", "2_RUA.bmp", and "3_RWA.bmp". The bottom left pixel of each image maps to (0, 0) coordinate and the top right pixel maps to (W-1, H-1) coordinate. So, the whole image should capture a part of the first quadrant only, which conforms to the constraint specified in the Input Format.

Sample Output: Not available, sorry. But you can see some visible differences while you implement the techniques.

* You will find lots of codes online (e.g. you can have a look here), you can use them but modify as applicable. Don't copy the codes, write yourself after understanding how it works. Your codes will be checked for plagiarism.

* Performance in this assignment can only improve the marks of CT-3, CT-4. The total CT marks will be evaluated as follows:
Sum of (Best 3 of (CT-1, CT-2, Best 2 of (CT-3, CT-4, Coding Assignment)))
Anything you get over 20 in the assignment will be distributed to CT-3 and/or CT-4 as appropriate.