

Computer Graphics (CSCI-GA 2270-001) - Assignment 1

Avadesh Meduri (N10537558)

September 24, 2020

1 Common Information

The code was written in Ubuntu 18.04, with a cmake version 3.10.2 and C++ version 7.5.0. I have implemented both the algorithms in this assignment using the template functions provided in the github repo and have only used standard C++ libraries. I have commented the code so as to explain what I am trying to do in each block of the function.

2 Answers to Exercises

1. Answer to Exercise 1

The convex hull obtained using the graham scan algorithm for the given data set is shown in figure 1. The blue points represent the points in the point cloud that the algorithm identified to be on the convex hull. The red points are the remaining points in the point cloud.

A short description of what each function does is written below :

- (a) salient angle function : a cross product between the three points is computed to check if a right turn is made (if the cross product is negative a right turn is made).
- (b) compare function : computes the cross product between vectors formed between p0 (left most point) and the two points that are compared. If the cross product is positive then the first point (p1) forms a smaller polar angle than the second point (p2).
- (c) convex hull : computes the convex hull using the graham scan algorithm. (function is commented in the code)
- (d) load_xyz : The function reads each integer one after the other. The first two integers in the line are read and stored, while the third integer is ignored. The function assumes that each line contains only one point coordinate (x,y,z), similar to the format of the file provided in the assignment.
- (e) det : I was not sure how to use this function in the exercise as I was able to implement the algorithm without this function.

2. Answer to Exercise 2

The result obtained by running the point in polygon file for the given data files is shown in figure 2.

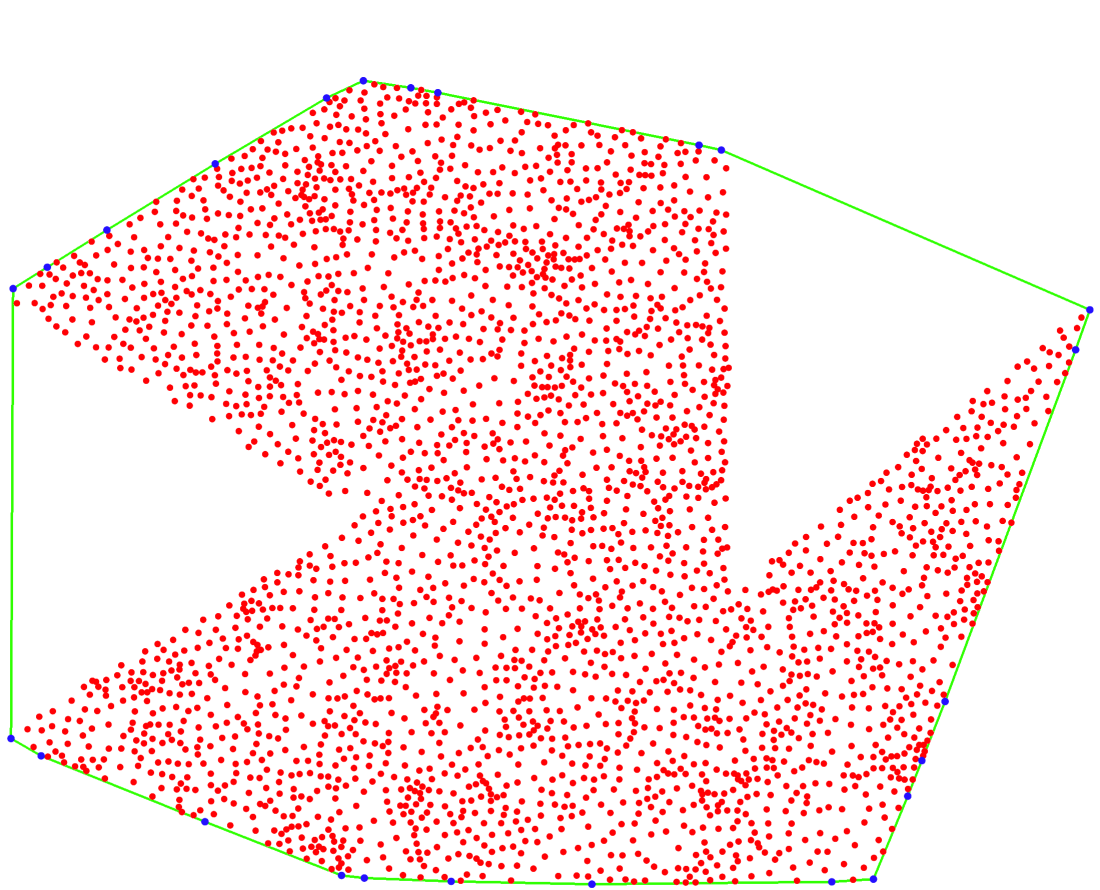


Figure 1: Obtained convex hull for the given set of points using the implemented graham scan algorithm

The green points are the points inside the polygon and the red are those outside the polygon.

A short description of what each function does is written below :

- (a) `intersect_segment` : The function first computes the co-efficients of the two lines formed by the 4 points (a, b form the first line, c, d form the second) so that they satisfy the equations $ax + by + c = 0$. After which, a simple check is performed to see if the lines are co linear or parallel by comparing the slopes. Subsequently, the point of intersection is computed by solving the two equations. Finally, a check is performed to see if the point of intersection lies within the two line segments, in which case true is returned.
- (b) `is_inside` : The function computes the bounding box by identifying the maximum and minimum x and y co-ordinates from the set of points that form the polygon. Using this, the top right point of the bounding box, (max x, max y) is multiplied by 10 times to be used as the point that is definitely outside the polygon. Using this point, a ray is cast from the point to be evaluated to the outside point and the `intersect_segment` function is used to see how many times the ray intersects with the polygon. This function assumes that the successive points in the cloud form the edges of the polygon and hence it takes the first two points, followed by the second two points (2nd and 3rd etc ...) as the edges of the polygon.
- (c) `load_xyz` : same as the previous exercise
- (d) `load_obj` : follows a similar pattern as the `load_xyz` function. It ignores the first and the last elements in each line and stores the second and third. The function terminates with it sees the letter 'f'.
- (e) `save_xyz` : stores the points that are identified to be inside the polygon, in the same format described in the github page.

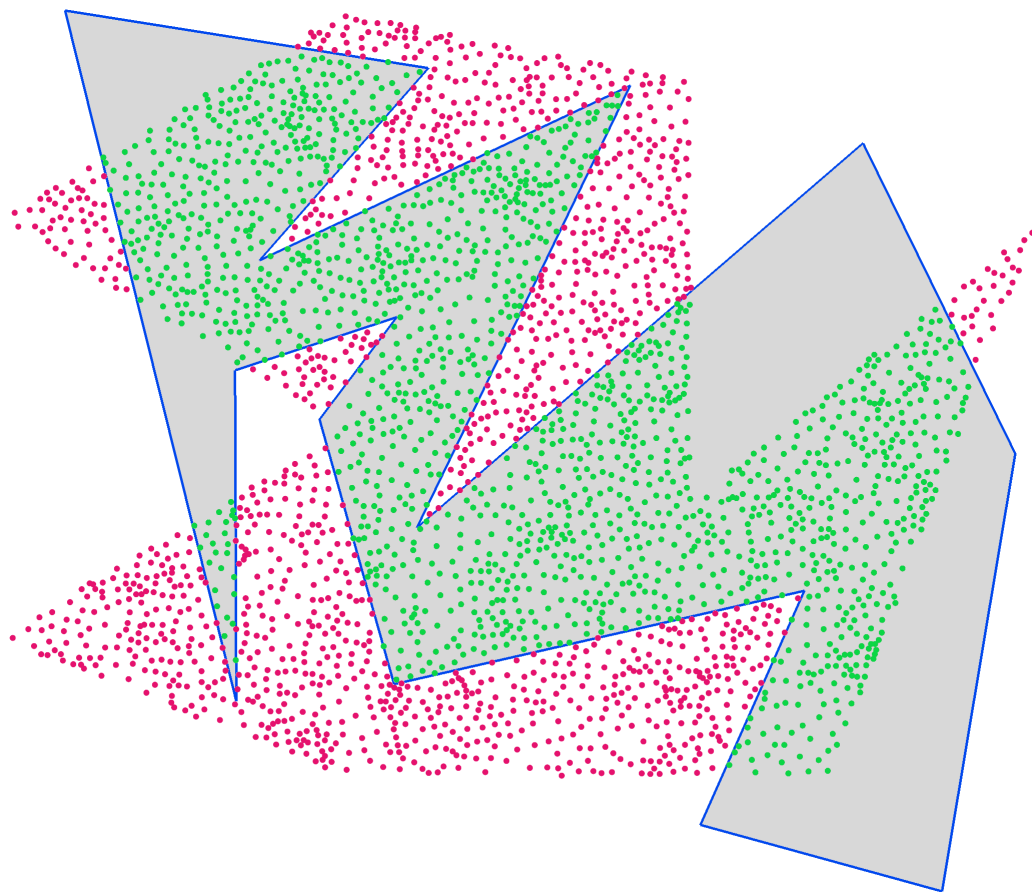


Figure 2: The points that are identified to be within the polygon and outside the polygon