

Conception and Implementation of Spatial Analysis Methods

Homework No.4

Raptakis Giorgos
MN: 374030

Contents

Used Functions.....	1
Graham Scan algorithm	1
Results.....	2

Used Functions

For implementation of Graham Scan algorithm we created firstly the functions.

Function of: `vector<point>random_points(int n, int min, int max)` which create a random number of points in a given interval.

Function of: `int orientation(point , point , point)` which find the orientation of ordered triplet if the relation is clockwise, counterclockwise or collinear.

Function of: `bool compAngle(point , point);` Which compares if the value of angles of two points are the same. We use this function for sorting of our vector of random points.

Function of: `double angle(point);` Which return the value of angle between the point and the point (0,0).

Function of: `void swap(point, point);` Which swap two points

Function of: `point nextToTop(stack<point>);` Which return the point next to last position in the stack.

Function of: `void save_data(string filename, vector<point> , vector<point>);` Which create a file.txt in the assigned position and save the random points that considered in the algorithm and the points of the results.

Function of: `stack<point> grahamScan(vector<point>);` Which is the implementation of our algorithm.

Graham Scan algorithm

In the function of `stack<point> grahamScan(vector<point>);` As input accepted a vector of points and returns a stack of points which shape the convex hull.

Firstly, we find the point with the lower y-value, and place it at the first position of our vector with the help of swap-function.

Then we sort the vector considering the angles of the points to be increased. We check if two consecutive points have the same angle, then keep the farther point from the point in the first position.

Then we create an empty stack and push the first three points in it (always the first two points are stable in the stack).

We make a for- loop for the vector of random points which checks the orientation of the last two points of the stack with the current point of the vector of random points. If the orientation of these three points is clockwise then we just put the current point of the vector to the stack, or else if the orientation of these points is counterclockwise we erase the last point of the stack and put the current point of the for-loop.

At the end, we have the points that form the convex hull of the 2D space of points.

Results

In our example, we create 2D random points in the interval of 0 to 50, which saved with name final.txt file and visualized in QGIS software.

In the following image we can observe the points that used for the convex hull and the formed polygon.

