**Bilkent University**

**Department of Computer Engineering**

CS 478/564 COMPUTATIONAL GEOMETRY

PROGRESS REPORT

**Zeynep Nur Öztürk**

**21501472**

# Implementation of Two-dimensional Convex Hull Algorithms and Comparing Their Performances

# 1.Introduction

## a. What is Convex Hull?

Convex hull is the smallest convex set which contains every point in a given point set. Figure 1 shows the convex hull of 1000 randomly created points.

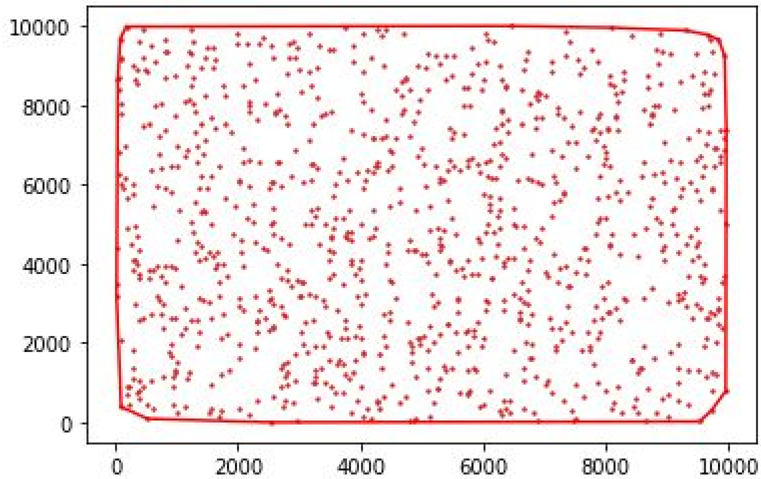



Figure 1 : A picture of convex hull which is output of the implemented code which will be discussed in section 2.

Calculating the convex hull has various algorithms. Based on the algorithm, the time complexity or memory usage differs. Preference of the algorithm depends on the fundamental aim of the project.

## b. Usage of Convex Hulls

Convex hull algorithms have been used in many field. Mathematics, statistics, economics, geometric modeling are some of the areas that uses convex hulls[6]. For example,  asymptotic behaviour of polynomials are calculating by convex hulls in mathematics[6].In robot motions,  the obstacles and paths that robot will move are determined by convex hulls[7]. Also, in a research about tracking disease epidemic on animals, the epidemic area has been found by convex hulls as in figure 2.

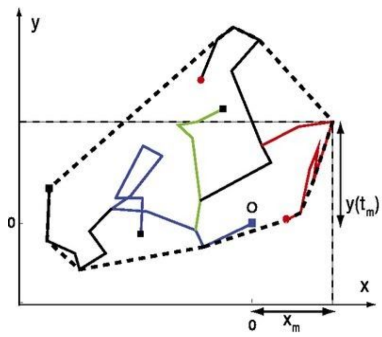



Figure 2 : A random walk of five individuals of animals. The graph is taken from article [8].

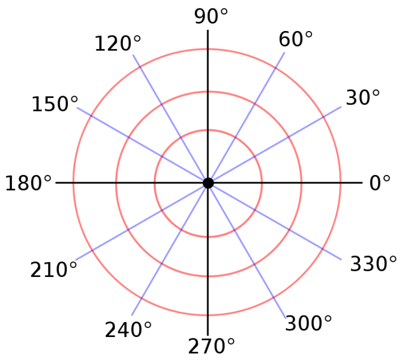
# 2. Background

In general most of the convex hull algorithms have O(nlogn) running time in 2D or 3D. Graham Scan, Gift Wrapping which is known as Jarvis March, Quickhull, Merge Hull which is known as Divide and Conquer Algorithm are investigated in this project.

## a. Graham Scan

This algorithm has been found by Ronald Graham in 1972 [9]. This has O(nlogn) running time. The steps of Graham scan has been explained based on the algorithm that is written for this project.

i. **Preprocess**: The random points received as an array S. The point which has the minimum y axis has been chosen as pivot. All of the points has been sorted by their polar angle from smallest the larger. If two vertices has the same polar angle, then the distance is calculated. The ordered array is given




to the function.

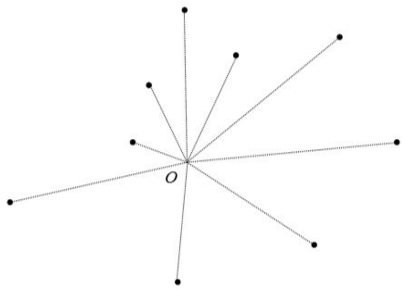



Figure 3: Polar grid [10]                Figure 4: polar ordered points[3]

ii. **Scan:** The pivot which is chosen and the first element in the sorted list are pushed in to the convex array, and in each loop we get another point from the sorted list and apply the left test for last 2 element in the convex array and the one in the sorted list. If left test fails the points are pop out from list until left test gives positive result. Then, return the convex hull array.

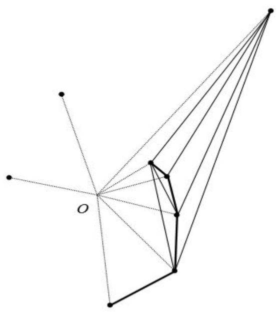



Figure 5: Scanning Process [3]

The preprocess dominated running complexity is O(nlogn), the " scanning process' complexity is O(nlogn).

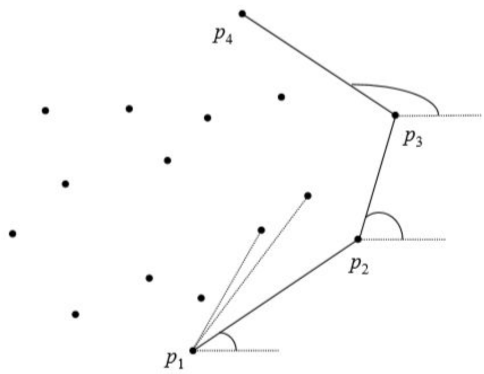
## b. Jarvis' March

This algorithm has been found by R.A. Jarvis in 1973 [11]. This algorithm has order O(nh) running time complexity. The steps of Jarvis March has been explained based on the algorithm that is written in the course slides [3].

i. **Preprocess:** preprocess of this algorithm is to find smallest y coordinated point as bottom pivot and largest as top pivot.

ii. **Scan:** In this part we order points based on the bottom pivot. Bottom pivot is pushed to the convex array, then choose the smallest polar angled point for the array's last element. Then continue until the top pivot and repeat the same step from top to bottom. Figure 6 represents this step.

This algorithm calculates only the vertices on the convex hull. In other words, only the points of the convex hull changes the running time. Preprocess has O(n) running time while the scan part has O(nh).This algorithm has O(nh) time complexity while h is the number of points in the convex hull.




. Figure 6: Jarvis March scanning [3]

## c. Quickhull

N dimensional Quickhull is invented by C. B. Barber, D.P. Dobkin and H.Huhdanpaa in 1996 [12]. The fundamental logic in this algorithm dividing the points into to 2 subarrays. The steps of Quickhull has been explained based on the algorithm that is written in the course slides [3].

i. **Preprocess:** Finding the smallest and largest point in the x axis and create another point which is slightly less then the smallest x point.

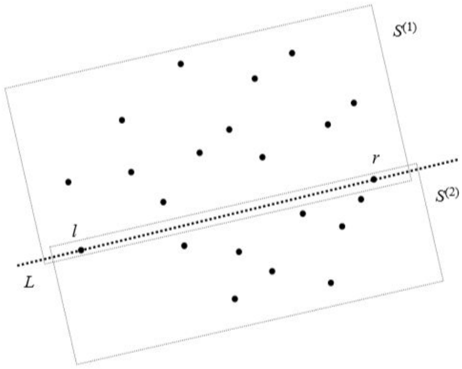



Figure 7: preprocess part of the quickhull[3]

ii. **Recursive method:** In this part two points and sub array S are given to the function. The farthest point is found on the S by calculating the triangle area which is O(1). The points in the triangle are ignored and the recursive called for the two subarrays for left out and right out part of the triangle. This continue until one point is left in the subarray.

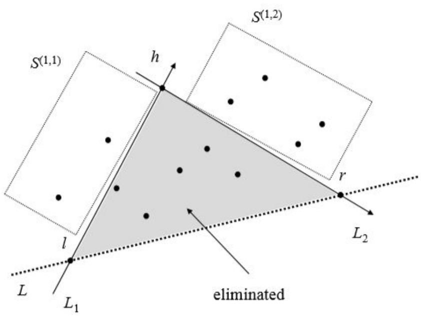



Figure 8: finding the farthest point and subarray the left and right part [3].

In the worst case, quickhull has O(n*n) running time which in case it is not equally divided in each recursive. However, it has O(nlogn) in the average running time. Preprocess part has O(n) running time while the recursive has O(n*n) in the worst case or O(nlogn) in the average case.

## d. Merge Hull

Merge hull has two different part. The fist part is calling quickhull recursively then merge them. Because the quickhull is explained in the section 2.c only the merge hull will be discussed based on the course slides [3].

i. Find p internal to P1.
ii. Check whether or not p is internal to P2. If internal to P2 go to the step iii else go to the step iv. To check it scan the P2 and apply left test even if one left test return negative it means it is not internal to the P2. This requires O(n) at most because of scan all the points in P2.

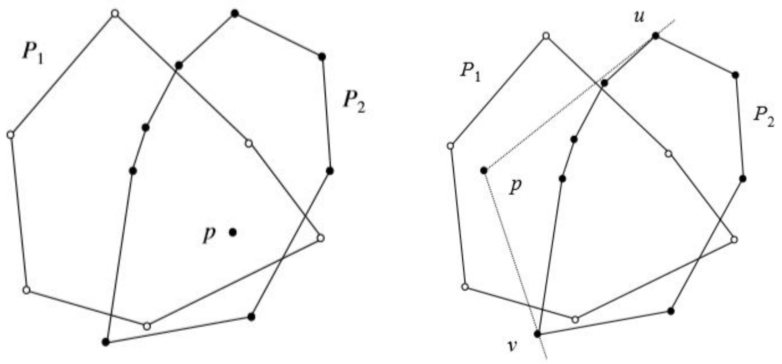



Figure 9: showing p inside P1 and P2 on the left and showing p inside P1 but not in side P2 on the right[3]

iii. Sort all points of P1 and P2 based on p with their polar angle. Sorting all points requires O(N) times
iv. Find u and v vertices on P2 to eliminate the middle chain. To find them, the supporting vectors are used by scanning all the points of P2 which has O(n) time complexity. Merge P2's outside chain and P1 and sort them by polar angle is O(N) times.
v. Use graham scan for points left. Because the list is sorted the graham scan's scan part will be used. Therefore, the time complexity is equal to O(n).

# 3. Progress

## a. Comparison

In the “An Associative Implementation Of Classical Convex Hull Algorithms” article, the authors published comparisons between Graham scan, Jarvis March, Quickhull. The table in below(figure 10) shows the seconds that spend. The points in that table were obtained by randomly generating x and y[5]. In this project the points will be created randomly from 1,000 up to 10,000,000. The clustered and non clustered methods will be created.

| Input Size | Graham Scan | Jarvis March | Quick Hull |
|---|---|---|---|
| 100 | .21 | .14 | .17 |
| 500 | .33 | .15 | .23 |
| 1000 | .37 | .21 | .26 |
| 3000 | .39 | .20 | .29 |
| 5000 | .42 | .22 | .31 |
| 10000 | .44 | .25 | .35 |
| 15000 | .45 | .27 | .30 |
| 16000 | .42 | .28 | .34 |
| 20000 | .32 | .35 | .36 |

Table 2: Time in Seconds to compute the Convex Hull for $n$ points.

Figure 11[5]

## b. Language & Frameworks

Python will be used for this project. For GUI implementation the tkinter library will be used.

# 4.References


**[**1] X. Kong, H. Everett, and G. Toussaint, “The Graham scan triangulates simple polygons,” *Pattern Recognition Letters*, vol. 11, no. 11, pp. 713–716, 1990.

[2] S.-L. Chen, J.-Y. Zhang, X.-G. Lu, K.-C. Chou, and Y. A. Chang, “Application of Graham Scan Algorithm in Binary Phase Diagram Calculation,” *Journal of Phase Equilibria & Diffusion*, vol. 27, no. 2, pp. 121–125, Jan. 2006.

[3]Computational Geometry: An Introduction, F. P. Preparata and M.I. Shamos, SpringerVerlag, 1985.

[5]Maher M. Atwah and Johnnie W. Baker and Selim Akl, “An Associative Implementation Of Classical Convex Hull Algorithms ” , http://www.cs.kent.edu/~jbaker/PDC-Sp12/references/atwah96.pdf Accessed 30.03.2020.

[6] Convex hull. (2020, April 18). Retrieved May 25, 2020, from https://en.wikipedia.org/wiki/Convex_hull

[7] Tao, Tan, & Juan. (2018, October 04). Path Planning with Obstacle Avoidance Based on Normalized R-Functions. Retrieved May 25, 2020, from https://www.hindawi.com/journals/jr/2018/5868915/

[8] Dumonteil, E., Majumdar, S., Rosso, A., & Zoia, A. (2013, March 12). Spatial extent of an outbreak in animal epidemics. Retrieved May 25, 2020, from https://www.pnas.org/content/110/11/4239.full

[9] Graham scan. (2020, April 21). Retrieved May 25, 2020, from https://en.wikipedia.org/wiki/Graham_scan

[10] Dumonteil, E., Majumdar, S., Rosso, A., & Zoia, A. (2013, March 12). Spatial extent of an outbreak in animal epidemics. Retrieved May 25, 2020, from https://www.pnas.org/content/110/11/4239.full

[11] Dumonteil, E., Majumdar, S., Rosso, A., & Zoia, A. (2013, March 12). Spatial extent of an outbreak in animal epidemics. Retrieved May 25, 2020, from https://www.pnas.org/content/110/11/4239.full

[12]Dumonteil, E., Majumdar, S., Rosso, A., & Zoia, A. (2013, March 12). Spatial extent of an outbreak in animal epidemics. Retrieved May 25, 2020, from https://www.pnas.org/content/110/11/4239.full