Convex Hull

Nabin Mulepati
Algorithms
Susquehanna University
Selinsgrove, PA 17870

May 10, 2010

Abstract

Convex hull of a set of points is the region that exclusively includes all of the points in the given set of points. There are different algorithms to find the planar convex hull of a set of points based on different time complexities. They include Graham scan, Jarvis march, Divide-and-Conquer, Optimal output-sensitive algorithms, and Akl-Toussaint heuristic.[1] In this paper we will study the techniques of Graham scan and Jarvis march and their computational speed. Jarvis march, the simpler of the two in terms of implementation, has a running time of O(nh), where n is the number of points in the set and h is the number of points in the hull. However, it has the worst case running time of O(n^2) when all given points form the convex hull. Graham's scan, with a more complex implementation, has a running time of O(nlogn).

Introduction

The convex hull of a finite set of points Q, denoted as CH(Q), is the smallest convex polygon P such that all points in the set Q are either in the interior of P or on the boundary of P.[1] In two dimensions, it is basically a polygon formed by tightly wrapping a thread around a set of needles on a board representing a set of points on a plane. For example, figure 1 shows the planar convex hull of a set of 37 points. Convex hull has a wide range of applications in the field of mathematics and computer science such as in digital image processing, pattern recognition, statistics, geographic information system, etc.[2] Convex hull also serves as a starting point for other computational-geometry problems like the two-dimensional farthest-pair problem.[1]

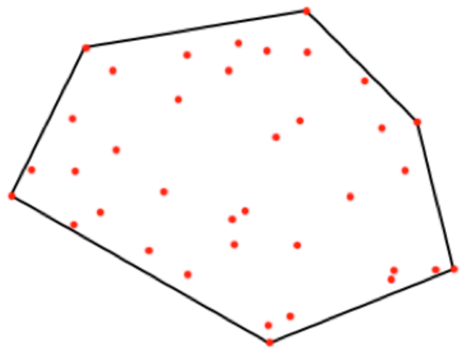

Figure 1: Convex Hull of a set of 37 points

Implementation

Two of the convex hull algorithms, Jarvis march and Graham scan, were implemented in C++. Jarvis march, the simpler of the two, has a complexity of O(nh), where n is the number of given points and h is the number of points in the resulting convex hull. Since h is unknown before the algorithm is run, Jarvis march is output sensitive and hence also known as an output-sensitive algorithm. Jarvis march is generally faster than Graham scan; however, when the size of the hull is proportional to the number of input points, Jarvis march has the worst case complexity of $O(n^2)$. In two dimensions Jarvis march is fairly easy to implement. It is similar to the process of tightly winding wrapping paper around a set of points. For this reason Jarvis march is also known as the Gift-wrapping algorithm.[1]

Jarvis march works by first finding the point with the lowest y-coordinate. In case of a tie, the point with the lowest x-coordinate is selected as the pivot or the starting extreme point. The next point in the hull is the point that forms a line that makes the smallest angle with the direction of the previous convex hull point. The points are selected in counterclockwise order until the next point is the same as the starting extreme point. The algorithm for Jarvis march is shown below:

```
1.      p0 is the extreme point with the minimum y-coordinate, or the leftmost point in
        case of a tie.
2.      Minimum angle = infinity
3.      do
    a.  for i=1 to the number of input points
          1. if the angle A formed by the direction of the previous hull and the next
          point is smaller than the minimum angle, then minimum angle = A
    b.  end for
    c.  push the point that forms the minimum angle A into the convex hull
4.      while next point in the convex hull is not equal to p0
```

For general purposes Graham scan is slower than Jarvis march with a complexity of O(nlogn). The algorithm works by first finding the extreme point with the lowest y-coordinate. In case of a tie, the point with the lowest x-coordinate is selected as the pivot or extreme point. Next, the points are sorted angularly with respect to the extreme point in counterclockwise direction. This is the step that determines the complexity of Graham scan. After this the algorithm uses a stack-based method to find the next points in the convex hull which runs in only O(n).[1] Lines 1 through 5 show the steps of the Graham scan algorithm.[1]

```
1. $p_0$ is the extreme point with the minimum y-coordinate, or the leftmost point in case
   of a tie.
2. $p_1, p_2, \dots, p_m$ are the remaining points in Q sorted by polar angle in counterclockwise
   order around $p_0$. If there is more than one point with the same angle, the point farthest
   from $p_0$ is selected.
3. Push $p_0$, $p_1$ and $p_2$ into an empty stack S.
4. for i = 3 to m
    a. while the angle formed by the point[i], and the top two points in the stack
       makes a right turn.
        i. Pop the point on the top from the stack.
    b. end while
    c. Push $p_i$ into the stack S.
5. return S
```

The algorithm above pushes each point in Q onto the stack but pops each point that is not in the convex hull. When the algorithm terminates, the stack contains all the points of the convex hull.[1]

For testing the program was implemented to populate the set of points in three ways – read points from an external file, generate a given number of random points, and generate a number of points on the circumference of a circle.

Results and Discussion

These two algorithms were tested on two test data sets. The first data set consisted of 50 to 200000 randomly generated points. The results obtained from the first data set are shown in table 1 below. For each number of points shown in table 1, each algorithm was run five times to get the mean running time with the standard deviation. The plot of the results in table 1 is shown in figure 1. The data in table 1 and the plot in figure 2 show that the running time for Jarvis march and Graham scan is almost the same for a small number of data points. However, as the number of input points increases, Jarvis march is faster than Graham scan.

# of Points	Size of Convex Hull	Mean running time of Jarvis march (ms)	Std. deviation for Jarvis march (ms)	Mean running time of Graham scan (ms)	Std. deviation for Graham scan (ms)
50	12 ± 2	**1**	0	**6**	5
5000	23 ± 2	**235**	38	**297**	97
10000	21 ± 1	**419**	59	**522**	134
20000	28 ± 3	**643**	135	**975**	332
40000	32 ± 3	**1336**	450	**1667**	554
60000	27 ± 5	**1602**	523	**2226**	551
80000	30 ± 3	**2110**	228	**3043**	639
100000	31 ± 4	**2578**	284	**3583**	404
150000	31 ± 2	**4572**	1335	**6478**	1425
200000	33 ± 2	**5798**	1352	**8116**	1088

Table 1: Results from the first dataset of random points.

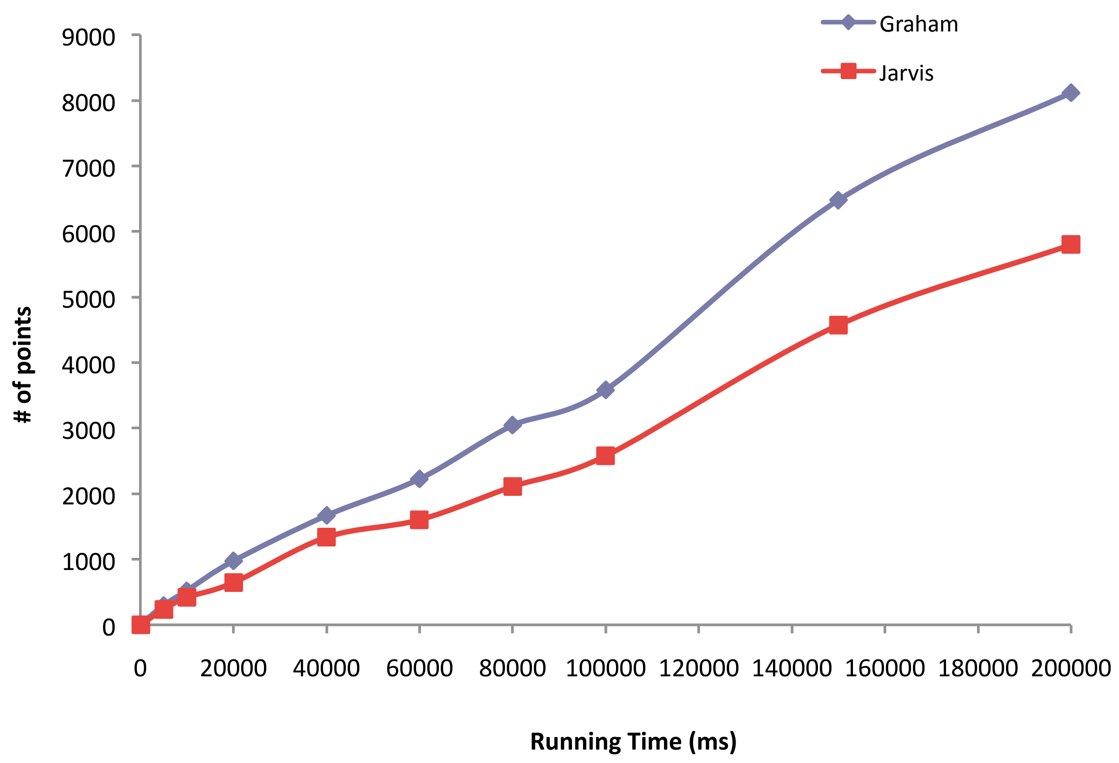

Figure 2: Plot showing # of input points vs. running time

It should be noted that the running time of Jarvis march is much lower in this dataset because the size of the convex hull is very small compared to the number of input points. In this dataset the running time of Jarvis march is O(nh) whereas the running time of Graham scan is O(nlogn) with the value of h being much smaller than that of log(n).

Size of Convex Hull	**Running time of Jarvis march (ms)**	**Running time of Graham scan (ms)**
63	17	1
419	313	64
1257	2029	157
2095	4119	360
3142	9548	922
4189	13729	1545
6284	25788	3246

Table 2: Results from the second dataset of points on the circumference of a circle.

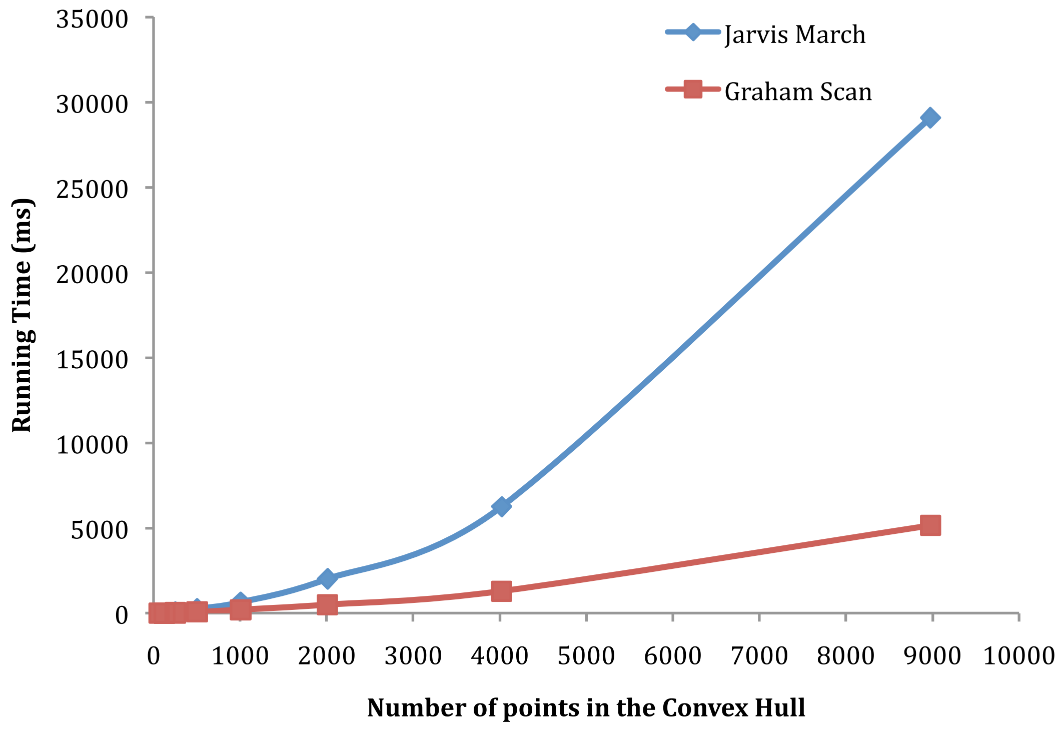

Figure 3: Plot showing running time vs. size of the Convex hull (worst case).

The points in the second data set were generated by picking a step size for the angle and a very large radius to find the points on the circumference of a circle using polar coordinates. This means that every input point is in the convex hull, which is the worst-case situation for Jarvis march. The results obtained from this data set are shown in table 2 and the plot of the running time vs. the size of the convex hull is shown in figure 3. The result shows that as the number of input points, all of which are in the convex hull, goes up the running time of Jarvis march goes up much faster in the order of $O(n^2)$ whereas the running time of Graham scan stays at O(nlogn). Clearly, Graham scan is significantly faster in this case.

Conclusion

The results obtained above show that Jarvis march algorithm with a running time complexity of O(nh) is generally faster than Graham scan algorithm with a running time complexity of O(nlogn). It was observed that Jarvis march is significantly faster than Graham scan if the size of the convex hull is known to be small. However, if the size of the convex hull is close to the number of input points, i.e. if the input points are distributed on the convex hull, Jarvis march is much slower with a running time of $O(n^2)$. From the experiments, it can be concluded that the results obtained above agree with the running time complexities of both of the algorithms.

References

1. Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. Third. Cambridge: The MIT Press, 2009. 1029-46.

2. Fleischer, Rudolf. "Convex Hull Applications." N.P., 2007. Web. 2 May 2010. <http://fleischer.selfip.com/Courses/Algorithms/Alg_ss_07w/Webprojects/Chen_hull/applications.htm>