

AN EFFICIENT ALGORITHM FOR THE CONVEX HULL OF PLANAR SCATTERED POINT SET

Zhongliang Fu, Yuefeng Lu*

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, 430079, P.R. China
- luyuefeng21@gmail.com

Commission II, WG II /3

KEY WORDS: Scattered Point Set, Minimum Convex Hull, Convex Hull Algorithm, Initial Boundary of Convex Hull

ABSTRACT:

Computing the convex hull of a point set is requirement in the GIS applications. This paper studies on the problem of minimum convex hull and presents an improved algorithm for the minimum convex hull of planar scattered point set. It adopts approach that dividing the point set into several sub regions to get an initial convex hull boundary firstly. Then the points on the boundary, which cannot be vertices of the minimum convex hull, are removed one by one. Finally the concave points on the boundary, which cannot be vertices of the minimum convex hull, are withdrew. Experimental analysis shows the efficiency of the algorithm compared with other methods.

1. INTRODUCTION

A convex polygon, with every internal angle less than or equal to 180 degree, is a simple polygon whose interior is a convex set. This means that all the vertices of the polygon will point outwards, away from the interior of the shape (Zhou, 2005). The convex hull of a set Q of points is the smallest convex polygon P for which each point in Q is either on the boundary of P or in its interior (Cormen et al., 2009). Computing the convex hull is a problem in computational geometry. It finds its practical applications in pattern recognition, image processing, statistics and GIS. It also serves as a tool to construct geometric shapes. Especially the minimum convex hull of a point set is requirement in the GIS applications such as area cut, TIN and DTM generation and area dynamic calculation.

At present, many scholars have proposed a number of effective algorithms for computing the convex hull of a set of points in two and three dimensions, such as incremental, Graham's scan, gift-wrapping aka Jarvis march and divide-and-conquer, and so on. Incremental method is to add points one at a time updating the hull as we proceed. Both Graham's scan and gift-wrapping use a technique called "rotational sweep", processing vertices in the order of the polar angles they form with a reference vertex. Divide-and-conquer method firstly divides point set into two subsets, and then recursively computes the convex hulls of the subsets before combining all of them. The time complexity of these algorithms can achieve $O(n\log n)$ or $O(nh)$. However, they are all involved in scanning all the points one by one, which result in much computational overhead. Normally, the minimum convex hull contains only a few points, and most of the points are in the interior of convex hull. There are some improved algorithms, which can enhance performance though excluding non-convex hull vertexes to reduce the analysis points of the minimum convex hull, such as

grouping the set of points (Wang, 2002; Zhang et al., 2009), establishing the auxiliary grid field (Wang, 2010), and obtaining the extreme points (Yu et al., 2005; Wu et al., 2010; Cheng et al., 2009; Liu et al., 2011; Barber et al., 1996). However, these algorithms have some faults including memory or subjective factors, which lead to inefficiency, such grouping and subsets dividing, additional storage space and artificial threshold determined.

Based on above, we present an improved algorithm to rapidly compute the minimum convex hull of a planar scattered point set in this paper. Firstly, the algorithm finds an initial convex hull boundary of the set of points by recursively dividing the set of points into sub regions. The boundary will be updated after each recursive division. Next, the points within the convex hull boundary are removed to avoid judgement. Finally, the concave points on the convex hull boundary are deleted to strike the minimum convex hull of the set of points.

2. PRELIMINARY

Definition 1 For a set of points S in 2-dimensional space, its convex hull is the smallest convex polygon containing S .

Definition 2 (Initial convex hull boundary) The polygon P , which contains some internal points in addition to the smallest convex hull vertices of the point set S , is called the initial convex hull boundary of the point set.

Definition 3 In all vertices of polygon, the one whose interior angle is less than π is convex point, the one whose interior angle is equal to π is called flat point, and the one whose interior angle greater than π is defined as concave point .

* Corresponding author.

Theorem 1 (Decision theory of the position relationship between point and straight line) Let p, q, r be the coordinates of any three distinct points on the plane, so $p=(p_x, p_y)$, $q=(q_x, q_y)$, $r=(r_x, r_y)$, \overrightarrow{pq} denotes a directed line segment from p to q , expressed as follow:

$$D = \begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix}$$

Further, $D = (q_x - p_x) \times (r_y - p_y) - (q_y - p_y) \times (r_x - p_x)$.

Positional relationship of r and \overrightarrow{pq} can be judged by the size of D , shown in Figure 1:

- (1) If $D > 0$, the point r is located to the left side of \overrightarrow{pq} ;
- (2) If $D < 0$, the point r is located to the right side of \overrightarrow{pq} ;
- (3) If $D = 0$, three points (p, q, r) are collinear.

Proof: Omitted. \square

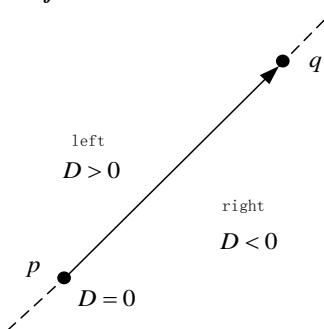


Figure 1. Judgement of the positional relation between point and straight line

Theorem 2 (Decision theory of the convex and concave of point) Let p_{i-1}, p_i, p_{i+1} be three consecutive vertices of the polygon in a counter clockwise direction. For the common endpoint p_i of $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_ip_{i+1}}$, the convex and concave will be determined using the vector cross product method, in other words, by the sign of $(p_i - p_{i-1}) \times (p_{i+1} - p_i)$ which is cross product of $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_ip_{i+1}}$. Let $p_{i-1} = (x_{i-1}, y_{i-1})$, $p_i = (x_i, y_i)$, $p_{i+1} = (x_{i+1}, y_{i+1})$. $S(p_{i-1}, p_i, p_{i+1})$ is convex and concave judgments function, and can be derived from following equation.

$$S(p_{i-1}, p_i, p_{i+1}) = (x_i - x_{i-1}) \times (y_{i+1} - y_i) - (x_{i+1} - x_i) \times (y_i - y_{i-1}).$$

As shown in Figure 2:

- (1) If $S(p_{i-1}, p_i, p_{i+1}) > 0$, p_i is convex point;
- (2) If $S(p_{i-1}, p_i, p_{i+1}) = 0$, p_i is flat point;
- (3) If $S(p_{i-1}, p_i, p_{i+1}) < 0$, p_i is concave point.

Proof: Omitted. \square

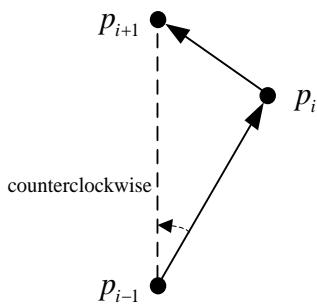


Figure 2. Judgement of the convex property or concave property of a point

3. ALGORITHM DESCRIPTION

3.1 Principle of the Algorithm

Firstly, find the points with maximum and minimum x coordinates, as well as the point with maximum and minimum y coordinates in planar scattered point set. The four extreme points can be used to generate an initial boundary of the convex hull. After removing points within the initial boundary of the convex hull, the point set will be divided into two parts by the two extreme points at x coordinate direction. And then the point set is divided into three vertical regions by the x coordinate of two extreme points at y coordinate direction. Then all the three regions are recursively divided again according to the point with maximum and minimum y coordinate in the region under the principle that is to determine whether the extreme point is bump or not by connecting the point and its two neighbour points on the initial boundary. If it is bump, we can add it onto the initial boundary, and then delete points within the initial boundary of the convex hull to get an initial boundary which contains all the vertices of the minimum convex hull. Finally, process the initial boundary again to get the final convex hull.

3.2 Steps of the Algorithm

Algorithm is as follows:

(1) All points of the planar scattered point set are sorted in ascending order of x -coordinates to find the minimum point p_{L0} and the maximum point p_{R0} of the point set based on their x coordinates, and thus the x coordinates of two extreme points are identified as the division border, which means the borders of the point set as shown in Figure 3 (1). At the meantime, the set of points is divided into two parts by $\overrightarrow{p_{L0}p_{R0}}$.

(2) Find out the point p_{T0} with maximum y coordinate and p_{B0} with minimum y coordinate respectively, then divide the entire set of points into three vertical regions (left, centre, right) based on the x coordinates of these two points (if x -coordinates of p_{T0} and p_{B0} are the same, the entire set of points will be divided into two vertical regions (left, right)). As shown in Figure 3 (2), the points p_{L0}, p_{R0}, p_{T0} and p_{B0} , are the vertexes of the final convex hull. In this case, points located in this initial convex hull interior will be excluded from the vertexes of the final convex hull.

(3) Recursively divide the left region: in the left region, repeat step 2 to find two adjacent vertices before and after the points p_{T1} and p_{B1} from the points set located in the same side of p_{T1} and p_{B1} . For example: if p_{T1} locates in left side of $\overrightarrow{p_{L0}p_{R0}}$, get the two adjacent vertices of p_{T1} from the vertexes which locates in left side of $\overrightarrow{p_{L0}p_{R0}}$ on the initial convex hull boundary, respectively. Based on Theorem 2, we can determine convexity of the point p_{T1} and the point p_{B1} . If it is bump, add this point to the initial boundary, delete the points within the initial boundary, and then divide the regions where the point locates again by the x -coordinate of this point. Otherwise, the

boundary is no longer being divided, as shown in Figure 3 (3)-(4). If the sub-region with lower level exists, recursively do the above operation until there is no any bumps or points in the sub-region.

(4) If the sub-region contains only one point, obtain the two adjacent vertices in the same side of this point, and then determine the nature of its convexity according to the location of $\overline{p_{L0}p_{R0}}$. If the bump, add this point to the initial boundary of convex hull and delete the points within the initial boundary of the convex hull, as shown in Figure 3(5).

(5) Choose the point with the maximum y-coordinate and the point with the minimum y-coordinate in the sub-region. In case of a tie, if these points locate in the upper part of the point set,

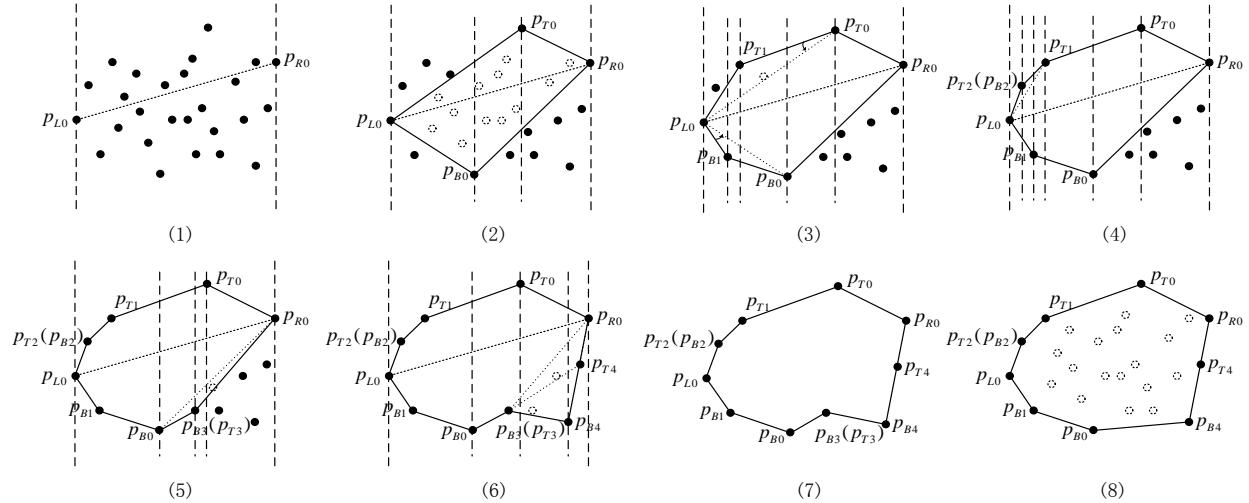


Figure 3. Generation process of the initial boundary

3.3 Algorithm Analysis

3.3.1 Conclusion: The initial boundary of the convex hull of the point set which is derived by the algorithm presented in this paper contains all points of the point set, that is to say, there is no any point outside of the boundary.

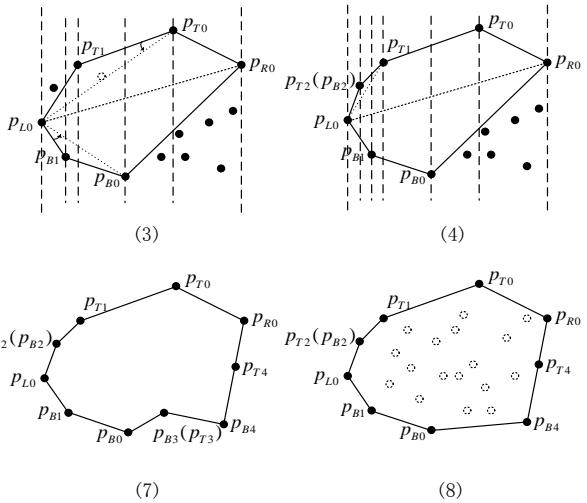
Proof: Assumption that there is any point outside of the initial boundary of the convex hull, which is denoted as p_w . Let the sub-region where it is located be R_{p_w} , and let the sub-set of points on the initial boundary of the convex hull in this sub-region be $S_R = \{p_i(x_i, y_i) | 0 \leq i \leq 2\}$. Because p_w is a point which is outside of the initial boundary of the convex hull ($p_w \notin S_R$), y coordinate of p_w must be greater than y coordinate of the maximum point or less than the y coordinate of the minimum point in S_R , where p_w is an extreme point of the sub-region R_{p_w} in the y coordinate. Whereas the vertices on the initial boundary, which derived in our algorithm, are all the extreme points in a sub-region at y coordinate direction (such as the points in S_R are the extreme points in the sub-region R_{p_w} at y coordinate direction). This situation is contradicted with assumption, so the initial boundary derived by our method is the boundary which contains all points of the point set. \square

choose the point which is farther away from p_{T0} in the x-coordinate. Otherwise, take the point that is farther away from p_{B0} in the x-coordinate.

(6) Repeat steps (3) - (5) operation in the second and third (if the third exist) vertical regions until to find the final initial boundary of the convex hull as shown in Figure 3 (7).

(7) According to Theorem 2, the convex and concave properties of vertices on the initial boundary will be judged.

(8) Delete the concave points on the initial boundary, the remaining points of the boundary are the all vertices of the final convex hull as shown in Figure 3(8).



3.3.2 Time Complexity: Currently, the lower limit of time complexity of the convex hull is $O(n \log n)$ (Yao, 1981). The time complexity of our algorithm for the extreme points is $O(1)$. The key step is to divide the point set. The worst case is that all points are concentrated in the vertices and the horizontal boundary of a triangle or in the vertices and the horizontal diagonal of a diamond. In this case, the point set can be divided into $n-1$ sub-regions and the extreme points are stroked in the sub-regions respectively with the time complexity of $O(n)$. The time complexity of determining the convex or concave property of the points and processing the adjacent point by sorting is $O(n \log n)$. Since each update needs to delete the points within this initial boundary of the convex hull, the number of judge points should be less than n . Therefore the total time complexity of the algorithm is less than $O(n \log n)$.

4. EXPERIMENTAL COMPARISON OF ALGORITHMS

We implement the classical Graham algorithm, divide-and-conquer algorithm and the algorithm proposed in this paper in C# on an Intel Core i5-M450 2.40 GHz PC with a 2GB main memory and an ATI Mobility Radeon HD 545v GPU. We generate random points with number from 500 to 10,000 in the same range, and design three groups of test to evaluate the performance of above three algorithms. They are striking the minimum convex hull of random points in a round, a

rectangular and an irregular shape, respectively. Figure 4 shows the convex hull of 1000 random points in three graphics respectively by using our algorithm.

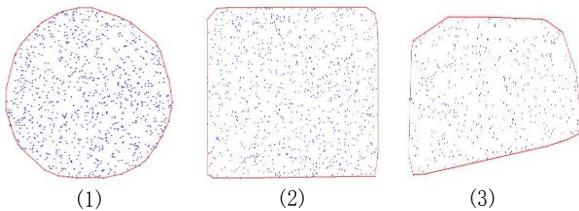


Figure 4. The minimum convex hull of 1000 random points

The time cost is calculated from the average value of overhead during striking a convex hull using the same points by 10 times. The results are shown in Table 1.

As can be seen from Table 1, our algorithm beats classic algorithms in generating the minimum convex hull of the same point set.

points	Graham's scan algorithm(ms)			divide-and-conquer algorithm(ms)			this paper's algorithm(ms)		
	round	rectangular	irregular	round	rectangular	irregular	round	rectangular	irregular
500	22	21	10	18	17	11	18	15	10
1000	69	56	25	39	38	20	36	32	18
5000	945	973	318	516	530	186	354	356	139
10000	3616	3668	1243	1836	1882	631	1245	1664	441

Table 1. Comparison of three algorithms

5. CONCLUSION

Convex hull problem is the classic problem in computational geometry. Corresponding convex hull algorithms have a wide range of applications in GIS, computer graphics and other fields. The proposed algorithm, first get the initial boundary of convex hull by dividing regions recursively, determines the vast majority of points within the boundary in the process of obtaining the initial boundary of the convex hull, and greatly improves efficiency for the final minimum convex hull by excluding these points. Experiments show that the algorithm is more efficiency than existing algorithms.

REFERENCES

ZHOU Peide, 2005. Computational Geometry: *Algorithms Analysis and Design (2nd Edition)*. Tsinghua University Press, Beijing, pp.44.

Thomas H.Cormen, Charles E.Leiserson, Ronald L.Rivest, et al., 2009. *Introduction to Algorithms (Third Edition)*. The MIT Press, Cambridge, pp.1014-1047.

WANG Jiechen, 2002. Study of optimizing method for algorithm of minimum convex closure building for 2D spatial data. *Acta Geodaetica et Cartographica Sinica*, 31(01), pp.82-86.

ZHANG Zhongwu, WU Xincui, 2009. Algorithm for convex hull of planar massive scattered point set. *Computer Engineering*, 35(09), pp.43-45, 48.

WANG Jiechen, CHEN Yanming, 2010. A gird-aided algorithm for determining the minimum convex hull of planar points set. *Geomatics and Information Science of Wuhan University*, 35(04), pp.403- 406.

YU Xiangyu, SUN Hong, YU Zhixiong, 2005. An improved algorithm to determine the convex hull of 2D points set. *Journal of Wuhan University of Technology*, 27(10), pp.81-83, 92.

WU Wenzhou, LI Lifan, WANG Jiechen, 2010. An improved Graham algorithm for determining the convex hull of planar points set. *Science of Surveying and Mapping*, 35(06), pp.123-125.

CHENG Sanyou, LI Yingjie, 2009. A new algorithm of minimum convex hull and its Application. *Geography and Geo-Information Science*, 25(05), pp.43-45.

LIU Renwu, YANG Dehong, LI Yan, et al, 2011. An improved algorithm for producing minimum convex hull. *Journal of Geodesy and Geodynamics*, 31(03), pp.130-133.

BARBER C B, DOBKIN D P, HUHDANPAA H. 1996. The Quick hull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4), pp.469-83.

YAO A C C, 1981. A lower bound to finding convex hulls. *Journal of the ACM (JACM)*, 28(4), pp.780-787.