Chinese Journal of Electronics
Vol.24, No.2, Apr. 2015

A Quick Convex Hull Building Algorithm Based on Grid and Binary Tree*

GAO Yang, CHENG Yuhu and WANG Xuesong

(*School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China*)

Abstract — A quick convex hull building algorithm using grid and binary tree is proposed for the minimum convex buidling of planar point set. Grids are used to assess and eliminate those interior points wihtout any contribution to convex hull building and points are sought in the boundary grid only so as to enhance the efficiency of algorithm. The minimum convex bull is built by taking such advantages of binary tree as quick, convenient and applicable for various point sets with different distributions, so as to resolve the description problem of concave point. The time complexity of the algorithm is low because of grid pretreatment. As the results of comparative expriment of random point and actual picture show, the proposed algorithm can obtain the best profile of 2D planar picture with minimum time, which is applicable for describing the shape of irregular convex-concave objects.

Key words — Convex hull, Gird, Binary tree, Concave point, Time complexity.

I. Introduction

Minimum convex hull of planar point set means the simple polygon that can contain all the points in the set and vertices belonging to the set[1]. As Geographic information system (GIS)[2] is more and more widely applied, the description of spatial object shape becomes increasingly important. Minimum convex hull can be used to describe more properly the shape of spatial object, and thus is widely applied to such areas as GIS, pattern recognition, data mining, artificial intelligence and image processing[3−5].

At present, many researchers have developed various minimum convex hull building algorithms[6−7] and the classical algorithms include Bulk synchronous parallel (BSP) algorithm[8], Grahame method[9] and divide-and-conquer algorithm[10]. BSP algorithm has to repeatedly calculate the angle between point and rotation line and compare the size of the angle so as to determine the boundary vertices. Therefore, its computational efficiency is not high. Prior to convex hull building, Grahame method needs to complete three preparatory tasks: (1) determine the basic point; (2) calculate the horizontal angle between all the elements of point set and basic-point connection line and the distance from basic point to the elements of point set; (3) sort as per size of the angle and distance, before generating the boundaries. Divide-and-conquer algorithm shows higher computational efficiency, but needs to combine the two generated boundaries, as affects the building speed. In view of the deficiencies with the classical algorithms, some researchers have made improvement of them. For instance, Fan *et al.* proposed a Quick convex hull (QCH) algorithm[11], of which the computing speed is faster than many previous algorithms. The time complexity of QCH is $O(N \log N)$, but $O(N^2)$ in the worst circumstance, where N is the number of discrete points. Wang *et al.* proposed a Grid-based convex hull (GCH) algorithm for determining the minimum of planar point set[12], of which the time complexity is $O(N + n \log n)$, where n is the number of discrete points retained after grid pretreatment. Both QCH and GCH can better describe the shape of convex objects, but can only obtain convex points rather than concave points from irregular convex-concave objects[13], thus failing to describe properly the shape of actual objects. For this reason, Sharif *et al.* proposed an algorithm to find Binary tree-based convex hull (BTCH), of which the time complexity is approximately $O(N)$ and $O(N^2)$ in the worst circumstance[14]. BTCH can better resolve the problem of convex point, but still cost relatively much time when the object contains a huge quantity of data points. To resolve the foregoing problems, a Quick gird and binary tree-based convex hull (QGBTCH) building algorithm is proposed, which mainly includes two parts: grid-based pretreatment of data set and BTCH building. On the one hand, the grid-based pretreatment method is used to eliminate the unnecessary points for the purpose of reducing the time spent in operation; on the other hand, binary tree is used to build minimum convex hull so as to resolve the description problem of concave points.

The rest of this paper is organized as follows. Section II describes the quick convex hull building algorithm based on grid and binary tree. Section III provides the experimental data

*Manuscript Received July 2013; Accepted Aug. 2013. This work is supported by National Natural Science Foundation of China (No.61273143, No.61472424), Specialized Research Fund for the Doctoral Program of Higher Education of China (No.20120095110025) and Fundamental Research Funds for the Central Universities (No.2013RC10, No.2013RC12, No.2014YC07).

 DOI:10.1049/cje.2015.04.015

and results analysis. Finally, Section IV provides our concluding remarks.

II. Quick Convex Hull Building Algorithm Based on Grid and Binary Tree

Mainly with the help of grids, QGBTCH eliminates the interior points which are not useful for convex hull building and reduces the counts traversed by binary tree-based convex hull building algorithm to enhance the overall performance of the algorithm. QGBTCH includes mainly four steps: 1) establish the grid field and calculate the points in the grid according to the input data set X; 2) eliminate the interior points, retaining only the point set X' included in the grid with points around; 3) establish the binary tree according to the x-coordinate and y-coordinate of the data set X' respectively; 4) sort the obtained data clockwise so as to obtain the final convex hull.

1. Grid-based pretreatment of data set

The real world is observed with massive data and various and complex information. In the process of obtaining the convex hull of the data set, most of its interior points are redundant. To enhance the efficiency of eliminating interior points, with reference to the grid method that is common used in GIS, the grid method is used for pretreatment of data set. The basic concept is as follows: establish a grid field that can cover the point set and find out the location (line number and column number) of points-contained grid in the boundary grid. Then, retain the points in the boundary grids (leftmost and rightmost point-contained grids of each line and uppermost and lowermost point-contained grids of each column) and eliminate the rest of points in the point set. As shown in Fig. 1, the graph includes 100 random points, with the size of the grid field as 10 lines × 10 columns, where '+' represents the point set retained eventually. The steps for grid-based pretreatment are as follows:

Step 1 establish a grid field that can contain data set, find out the boundary grids with points and determine the points contained in such grids.

Step 2 retain the points determined in Step 1 and eliminate the rest of data points.

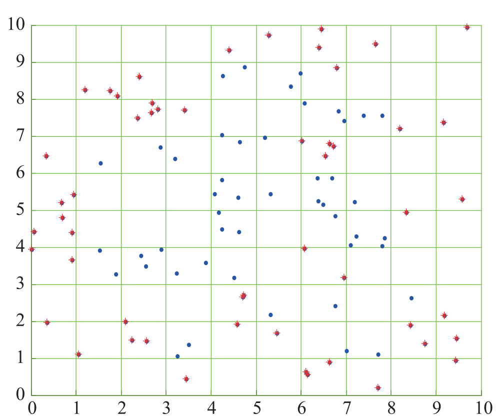


Fig. 1. Grid-based pretreatment for data set

For the optimal effect of eliminating redundant points, the lines and columns of the grid field are set as $L = C = \mathrm{int}(\sqrt{N})$, where L is the number of lines for the grid field, C is the number of columns for the grid field, N is the number of points in the data set and int represents the rounding operation[12]. As for GCH method in Ref.[12], upon establishment of a grid field, it is necessary to seek the location of each point in the grid field, but with our grid-based method used, it is necessary to seek the points in the boundary grids only, with no need to seek the internal grid and points therein.

2. BTCH building

Binary tree is a finite set of n ($n << N$) nodes, which may be an empty set, or composed of one root node and two sub-trees, known as the left sub-tree and right sub-tree of the root, respectively, which are not intersected with each other. To build the convex hull, first create a binary tree for x-coordinate of points in the point set, with x-coordinate of the first point as the root of the binary tree. According to its x-coordinate, the new node is added into the binary tree: if the x-coordinate of the point is greater than the root node, it will be added into the right sub-tree; otherwise into the left sub-tree. For each node, the minimum y-coordinate $y_{\min}$ and the maximum y-coordinate $y_{\max}$ corresponding to its x-coordinate will be saved. If the x-coordinate of the new node is equal to the previous one, the x-coordinate is given up and the y-coordinate y is compared. When the y-coordinate y is greater than $y_{\max}$ or smaller than $y_{\min}$, the maximum y-coordinate or minimum y-coordinate will be updated; otherwise the y-coordinate y will be given up. The algorithm can better obtain the boundary point and eliminate all the interior points. The algorithm steps for binary tree-based convex hull building are as follows:

Step 1 Input x-coordinate x and y-coordinate y of n number of nodes.

Step 2 Establish the binary tree according to x: make x_1 as the root node of the binary tree *root*. If $x_2 < root$, save x_2 in the right sub-tree; otherwise in the left sub-tree. If $x_i = x_j$ where $i = 2, \cdots, n$ and $j = 1, \cdots, i-1$, give up x_i.

Step 3 Along with Step 2, save or update the maximum value and minimum value of x-coordinate in the corresponding node. If $y < D$ (where D is a constant, being generally the central point of the y-coordinate), $y_{\min} = y$; otherwise $y_{\max} = y$. When $x_i = x_j$, amend $y_{\min}$ and $y_{\max}$ according to the value y_i.

Step 4 Obtain the binary tree of nodes, as shown in Fig. 2.

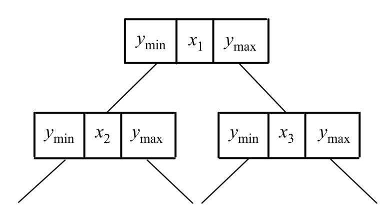


Fig. 2. Schematic diagram of binary tree

Step 5 Repeat Steps 2–4 to establish the binary tree for the y-coordinate y.

Step 6 Take out all the nodes from the binary tree and sort clockwise to obtain the convex hull of data set.

3. Time complexity analysis

In the grid-based pretreatment of data set, the time complexity is approximately $O(\sqrt{N})$ and the worst time complexity is $O(N)$. Because the time complexity of binary tree-based convex hull building algorithm is $O(2n)$, the total time complexity of QGBTCH is $O(2n+\sqrt{N})$ and the worst time complexity is $O(2n+N)$.

III. Experiment and Analysis

Random data points and actual picture are selected, respectively, for the experimental analysis and the algorithms for comparison include QCH, GCH and BTCH. To eliminate the influence of random factors, each experiment is conducted 30 times, from which a mean value is taken. The hardware of computer is configured as: Pentium CPU frequency: 2.20GHz; memory: 2GB; operating system: Windows XP Pro SP3; programming environment: MATLAB 7.8.0.

1. Experiment of random data points

To assess the computational efficiency, generate 7 groups of discrete data points at random 1000, 5000, 10000, 20000, 50000, 80000 and 100000; the value range of data points is the integer within $[0,N]$ interval. Table 1 shows the computer time consumption for 4 algorithms to determine the minimum convex hull.

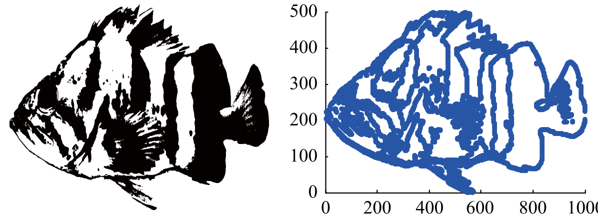
Table 1. Comparison of computer time consumption on random data point set

Number of data points	QGBTCH (s)	QCH (s)	GCH (s)	BTCH (s)
1000	0.00497	0.17620	0.08023	0.036868
5000	0.00784	0.18032	0.08143	0.343803
10000	0.01180	0.18128	0.08311	1.365853
20000	0.01637	0.18342	0.08796	7.778962
50000	0.03251	0.18615	0.09199	78.39020
80000	0.04873	0.18270	0.09652	217.47710
100000	0.05491	0.18440	0.09975	343.16240

From Table 1, it may be observed: 1) because the time complexity of all the algorithms is related to the number of data points, with the scale of data points expanding, the computer time consumption of all the algorithms shows a trend of gradual increase; 2) the computer time consumption is relatively low for QGBTCH and GCH, for the grid-based pretreatment has eliminated most of the non-related points in advance, while only a limited number of points are involved in convex hull operation; 3) comparing with GCH, QGBTCH consumes less time, for GCH needs to seek the location of all the points in the grid field, while QGBTCH only needs to seek the points in the boundary grids with a limited number; 4) when the scale of data points is relatively small, the advantage of the grid-based algorithm is not obvious: *e.g.*, when the number of random points is 1000, the computer time consumption of BTCH is much lower than that of GCH.

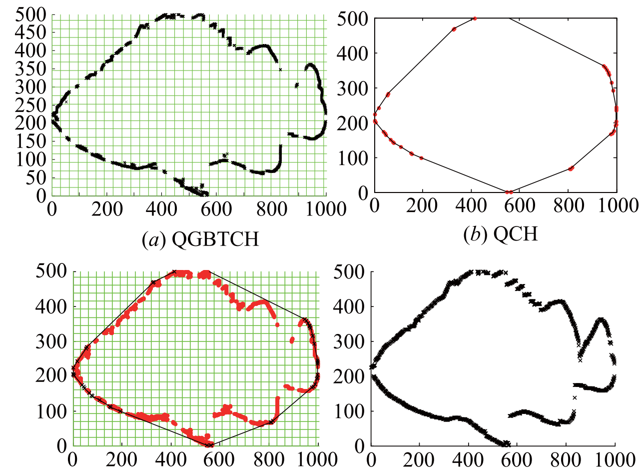
2. Experiment of actual picture

Substitute the random point set with one actual picture of fish as shown in Fig. 3(a), while Fig. 3(b) is the boundary points of fish. Using QGBTCH, QCH, GCH and BTCH respectively, seek the convex hull of the boundary points of fish, as shown in Fig. 4, where the points in Figs. 4(a) and (d) represent the vertices of convex hull, the lines in Figs. 4(b) and (c) represent the convex hull, and the points in Fig. 4(c) represent the points in the grid or the vertices of convex hull.



(a) Actual picture of fish (b) Boundary points of fish

Fig. 3. Actual picture and boundary points of fish



(a) QGBTCH (b) QCH (c) GCH (d) BTCH

Fig. 4. Convex hull building result of actual picture

From Figs. 4(a) and (d), it may be observed that the profile of fish can be obtained and the problem of concave points can be resolved by seeking the convex hull with the binary tree method. Figs. 4(b) and (c) show that with the quick convex hull method, only a few finite points (the points of line connection in the figure), without distinguishing what graphs are obtained. As Figs. 4(a) and (d) show, the horizontal and vertical concave-convex profiles of fish can be obtained with QGBTCH, but only the horizontal concave-convex profile and vertical convex profile of fish with BTCH. With QGBTCH and GCH, the convex hull can be built only using the discrete points within grid in Figs. 4(a) and (c), but QCH and BTCH both need to seek the convex hull according to all the boundary points of fish in Fig. 4(b). Therefore, both QGBTCH and GCH cost less time, as is also proven by the results of Table 2. All in all, QGBTCH can obtain the full profile of 2D actual picture with a minimum cost of time and can well identify objects.

Table 2. Comparison of computer time consumption on actual picture

QGBTCH (s)	QCH (s)	GCH (s)	BTCH (s)
0.0524	0.1827	0.0952	0.6350

IV. Conclusions

The convex hull problem of points in the plane is one of the oldest and most-studied problems. The task is to determine the smallest convex polygon that contains all the given points. Minimum convex hull is an important measure for describing the shape of spatial objects. When the shape of a target object is of irregular concave-convex or the data point set is in a relatively big scale, the existing convex hull building algorithms are observed with such problems as unclear profile description and low computational efficiency. For this reason, a quick algorithm for determining the minimum convex hull of planar point set is proposed with reference to the advantages of grid and binary tree. Its main concept is: pretreat data through a grid field and eliminate a large number of redundant interior points which are not actually involved in convex hull building; establish the binary tree based on the data points retained after grid pretreatment and sort all the nodes of the binary tree clockwise so as to obtain the minimum convex hull of planar point set. As the results of comparative experiment of discrete data point set generated at random and actual picture show, comparing with the currently representative algorithms of minimum convex hull building, the proposed algorithm can better describe the profile of irregular objects and its time complexity is relatively low.

References

[1] H.C. Thomas, E.L. Charles, L. Ronald and S. Clifford, *Introduction to Algorithms, Second Edition*, MIT Press and McGraw-Hill, pp.955-956, 2001.

[2] V.D. Obade and R. Lal, "Assessing land cover and soil quality by remote sensing and geographical information systems (GIS)", *Catena*, Vol.104, No.1, pp.77-92, 2013.

[3] J. Bai and X.C. Feng, "Image denoising and decomposition using non-convex functional", *Chinese Journal of Electronics*, Vol.21, No.1, pp.102-106, 2012.

[4] M.I. Karavelas, R. Seidel and E. Tzanaki, "Convex hulls of spheres and convex hulls of disjoint convex polytopes", *Computational Geometry-theory and Applications*, Vol.46, No.6, pp.615-630, 2013.

[5] L.Y. Ma and J. Yu, "A convex approach for local statistics based region segmentation", *Chinese Journal of Electronics*, Vol.21, No.4, pp.623-626, 2012.

[6] W.M. Cao and L.J. Ding, "Cancer subtypes recognition and its gene expression profiles analysis based on geometrical learning", *Chinese Journal of Electronics*, Vol.15, No.4, pp.891-894, 2006.

[7] Y. Davydov and C. Dombry, "Asymptotic behavior of the convex hull of a stationary Gaussian process", *Lithuanian Mathematical Journal*, Vol.52, No.4, pp.363-368, 2012.

[8] L. Cinque and M.C. Di, "A BSP realization of Jarvis' algorithm", *Pattern Recognition Letters*, Vol.22, No.2, pp.147-155, 2001.

[9] H. Ratschek and J. Rokne, "Exact and optimal convex hulls in 2D", *International Journal of Computational Geometry & Applications*, Vol.10, No.2, pp.109-129, 2000.

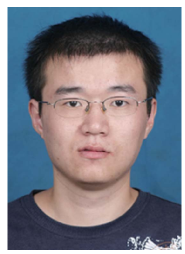
[10] C.D. Castanho, W. Chen and K. Wada, "A parallel algorithm for constructing strongly convex superhulls of points", *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, Vol.E83A, No.4, pp.722-732, 2000.

[11] J.S. Fan, S.W. Xiong and J.Z. Wang, "The multi-objective differential evolution algorithm based on quick convex hull algorithms", *Proc. of the 5th International Conference on Natural Computation*, Tianjin, China, pp.469-473, 2009.

[12] J.H. Wang and Y.M. Chen, "A gird-aided algorithm for determining the minimum convex hull of planar points set", *Geomatics and Information Science of Wuhan University*, Vol.35, No.4, pp.403-406, 2010. (in Chinese)

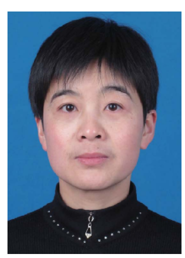
[13] S.D. Li, D.S. Wang and Y.Q. Dai, "Efficient secure multiparty computational geometry", *Chinese Journal of Electronics*, Vol.19, No.2, pp.324-328, 2010.

[14] M. Sharif, S. Khan, S.J. Khan and M. Raza, "An algorithm to find convex hull based on binary tree", *Proc. of the IEEE 13th International Multitopic Conference*, Islamabad, Pakistan, pp.1-6, 2009.



GAO Yang received the B.S. degree from China University of Mining and Technology in 2008. He is currently a doctor candidate in the School of Information and Electrical Engineering, China University of Mining and Technology. His main research interest is remote sensing image processing. (Email: gaojianrui88@163.com)

CHENG Yuhu received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences in 2005. He is currently a professor in the School of Information and Electrical Engineering, China University of Mining and Technology. His main research interests include machine learning and intelligent system. (Email: chengyuhu@163.com)



WANG Xuesong (corresponding author) received the Ph.D. degree from China University of Mining and Technology in 2002. She is currently a professor in the School of Information and Electrical Engineering, China University of Mining and Technology. Her main research interests include machine learning and artificial intelligence. (Email: wangxuesongcumt@163.com)