

## Research Article

# Grid Partition Variable Step Alpha Shapes Algorithm

Zhenxiu Liao , Jun Liu, Guodong Shi, and Junxia Meng

*School of Civil Engineering, Anhui Jianzhu University, Hefei, Anhui Province 230601, China*

Correspondence should be addressed to Zhenxiu Liao; [liao@ahjzu.edu.cn](mailto:liao@ahjzu.edu.cn)

Received 26 March 2021; Accepted 3 May 2021; Published 18 May 2021

Academic Editor: Sakander Hayat

Copyright © 2021 Zhenxiu Liao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

On the basis of Alpha Shapes boundary extraction algorithm for discrete point set, a grid partition variable step Alpha Shapes algorithm is proposed to deal with the shortcomings of the original Alpha Shapes algorithm in the processing of nonuniform distributed point set and multiconcave point set. Firstly, the grid partition and row-column index table are established for the point set, and the point set of boundary grid partition is quickly extracted. Then, the average distance of the  $k$ -nearest neighbors of the point is calculated as the value of  $\alpha$ . For the point set of boundary grid partition extracted in the previous step, Alpha Shapes algorithm is used to quickly construct the point set boundary. The proposed algorithm is verified by experiments of simulated point set and measured point set, and it has high execution efficiency. Compared with similar algorithms, the larger the number of point sets is, the more obvious the execution efficiency is.

## 1. Introduction

From the traditional electronic total station, handheld satellite positioning collector, to mobile (vehicle-mounted/airborne) three-dimensional laser radar, modern spatial information acquisition technology has entered the era of massive data at the GB and TB level. How to store, process, and express massive data efficiently has become a new challenge to the related fields such as computer information technology (IT), computer-aided design/manufacturing (CAD/CAM), geographic information (GIS) and remote sensing (RS), and even building information modeling (BIM) [1–4]. The boundary information of the discrete point set is composed of discrete points representing the original contour features of the measured object. It is a basic and important technical work in spatial data processing to quickly and efficiently construct the boundary information from the discrete point set. Two-dimensional point set boundary information is the basic data of land area statistics, road earthwork calculation, and other engineering applications [5, 6]. Three-dimensional point set boundary information plays an important role in the process of 3D model reconstruction [7–10].

To construct the boundary information of point set, it is necessary to study the shape of point set composed of two-dimensional or three-dimensional discrete points. From the

literature reading analysis, it is known that, since the 1970s, scholars and experts at home and abroad have successively carried out research work in this field and got good research results. Graham [11] proposed a scanning algorithm for determining the convex hull of a planar set, but the algorithm is only effective for convex hull boundary and cannot deal with the case of concave boundary; Jarvis [12] proposed a numerical method based on two-dimensional convex hull point set to express the contour geometry of two-dimensional point set, but again it can only deal with convex hull boundary; Sampath and Shan [13] improved R. A. Jarvis's algorithm, which can be used to deal with two-dimensional discrete point concave boundary, but the algorithm is not as efficient as the new algorithm proposed by subsequent researchers. In the 1980s, Edelsbrunner et al. [14] gave a rigorous mathematical definition on the “shape of a set of points” based on two-dimensional plane point set and proposed a point set boundary construction algorithm called Alpha Shapes (AS). Based on rigorous mathematical definition, this algorithm can deal with complex discrete point set boundaries including convex hull, concave points, and holes. Ten years later, H Edelsbrunner extended the AS algorithm [15] to be applied to three-dimensional point set surface reconstruction, thus greatly expanding the scope of the application of the algorithm. Jochem et al. [16] used the

AS algorithm to automatically extract the building roof from airborne LiDAR point cloud; Shen et al. [17] and Li et al. [18] used the improved AS algorithm to extract building contour; Wang et al. [19] used the improved algorithm to extract edges from massive point cloud data in mountainous areas; Li and Li [20] used the improved algorithm to reconstruct the 3D surface model from the point cloud data of handicrafts; Sun et al. [5] applied the improved algorithm to extract the plot boundary from the trajectory data points collected by the vehicle-mounted satellite navigation receiver of agricultural machinery and then finely measured the farmland area; Li et al. [21] applied the algorithm to construct the tree crown three-dimensional model; Fu et al. [22] applied the algorithm to construct a three-dimensional model from jujube tree point cloud.

## 2. Alpha Shapes Algorithm

*2.1. Algorithm Content.* In literature [14], Edelsbrunner gave a rigorous mathematical definition of the geometric shape of a two-dimensional plane point set, namely,  $\alpha$ -Shape. Let  $S$  be a two-dimensional planar point set, and give any parameter  $\alpha$ ; the polygon  $\partial S$  extracted from  $S$  by the AS algorithm rule is  $\alpha$ -Shape, which can be used to express the boundary contour of the point set, and its precision is determined by the value of the parameter  $\alpha$ .

Simplified AS algorithm steps are as follows:

Step 1: Input two-dimensional point set  $S = \{P_1, P_2, \dots, P_n\}$ , and calculate the point set average point spacing  $d$  as the value of  $\alpha$ .

Step 2: Traverse  $S$ , take  $P_i$  ( $i \leq n$ ) as the center of the circle, and take the length not greater than  $2\alpha$  as the radius  $R$  to construct the search circle, and search the point set  $S_i = \{P_1^i, P_2^i, \dots, P_m^i\}$  ( $m \leq n$ ),  $S_i \subseteq S$ .

Step 3: Traverse  $S_i$  and construct  $\alpha$ -shape criterion to determine whether line segment  $P_i P_j$  ( $j \leq m$ ) is a boundary edge. If so, add it to  $\alpha$ -shape set and return to Step 2. If not, proceed to the next point until the traversal is complete and return to Step 2.

Step 4: After traversing  $S$  is complete, output  $\alpha$ -Shape collection.

The simplified flow chart of Alpha Shapes algorithm is shown in Figure 1.

Among these steps, the third step, “construct  $\alpha$ -shape criterion,” is the core step of the algorithm, and the rules are as follows.

Draw a circle with radius  $R = \alpha$  through two points  $P_i, P_j$  (as  $P_1$  and  $P_2$  in Figure 2). If no point in point set  $S$  falls into the circle, the line segment  $P_i P_j$  is the boundary line; otherwise, it is the nonboundary line. The judgment method of whether a point in the point set  $S$  falls into the circle is shown in Figure 2. According to formula (1), the center point  $P_c$  is calculated, the point set  $S$  is traversed, and the distance  $d_k$  between each point (as  $P_k$  in Figure 2) and the center point  $P_c$  is calculated. If  $d_k < \alpha$ , it indicates that a point falls into the circle.

The formula for calculating the coordinates of the center point  $P_c$  can be obtained via the “distance intersection algorithm” in Geomatics [23]:

$$\begin{cases} x_c = x_1 + 0.5(x_2 - x_1) + H(y_2 - y_1) \\ y_c = y_1 + 0.5(y_2 - y_1) + H(x_1 - x_2) \end{cases} \quad (1)$$

In the formula,  $H = \pm \sqrt{(\alpha/d)^2 - (1/4)}$ ,  $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ .

The process of constructing point set boundary by AS algorithm can be understood as a circle with radius  $R = \alpha$  rolling outside the edge of the point set  $S$ . When the value of  $\alpha$  is appropriate, the trajectory that the circle rolls through is the boundary of point set  $S$ , as shown in Figure 3. At the range of  $\alpha \in (0, \infty)$ , when  $\alpha \rightarrow 0$ , all points in  $S$  are boundary points; when  $\alpha \rightarrow \infty$ , the convex hull of  $S$  is the boundary; when the value of  $\alpha$  is reasonable and the distribution of points in  $S$  is uniform, the AS algorithm can construct the boundary of point set  $S$  in an ideal way. If the value of  $\alpha$  is too large, the turning angle of the lines connecting the boundary points would be replaced by a larger blunt angle, resulting in a blunted effect at the corner, as shown in the shaded area in Figure 3.

*2.2. Algorithm Shortcomings.* Compared with the previous similar algorithms, AS algorithm has the advantages of rigorous mathematical definition, the ability to deal with complex two-dimensional point set boundary and three-dimensional point set surface, and a wide range of applications, but there are also shortcomings. The only parameter  $\alpha$  in the algorithm determines the fineness of the point set shape, which needs to be manually input and adjusted according to different scenarios. At the same time, the unicity of parameter  $\alpha$  determines that the algorithm is very suitable for processing convex hull point sets with uniform distribution density but cannot ideally deal with the two following scenarios:

- (1) For point sets with nonuniform density distribution per unit area, such as the discrete points at the farmland boundary collected by handheld or vehicle-mounted satellite navigation receiver, and the three-dimensional laser point cloud of bare tree branches, the processing effect is not very ideal.
- (2) For the point set composed of many concaves, such as complex buildings, roads, water flow, and other linear features, the processing effect in the concave corner area is not perfect. Therefore, it is necessary to improve and perfect the algorithm.

When applying AS algorithm to extracting building contour, it is found in [17] that if the value of  $\alpha$  is too small, the building contour point set will be very fragmented; if it is too large, the concave corner area of the building will be distorted due to being excessively blunted (Figure 3). According to different application scenarios, literatures [5, 18] put forward an improved algorithm called “dual threshold Alpha Shapes,” aiming at the specific problems caused by the excessively single value of  $\alpha$ . Based on the AS

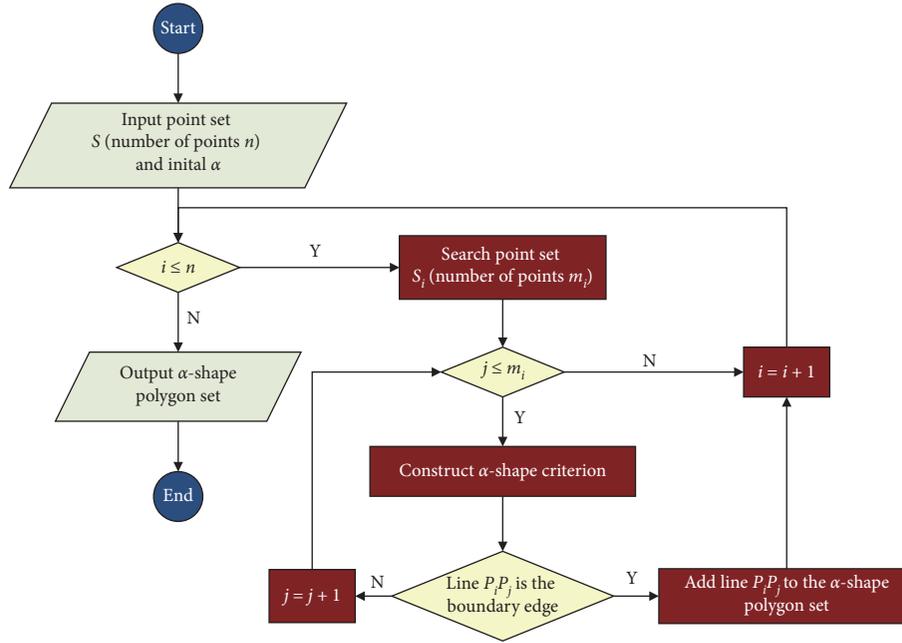


FIGURE 1: Simplified flow chart of Alpha Shapes algorithm.

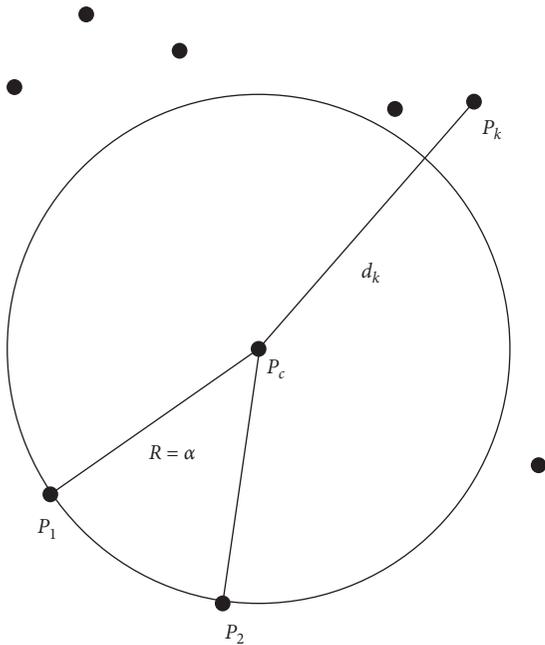


FIGURE 2: Sketch map of  $\alpha$ -Shapes criterion.

algorithm, literature [19] used the grid detection method to quickly filter nonboundary points and proposed an algorithm for fast extracting edges from massive point clouds. However, the  $\alpha$  value of the algorithm is fixed, which still cannot overcome the specific problems caused by the single value of  $\alpha$ . Literature [20] proposed a surface reconstruction algorithm using “self-adaptive step Alpha Shapes algorithm,” which can automatically calculate different values according to the point density of different regions of the point set and better deal with the problem caused by

nonuniform point distribution. However, this algorithm requires that every point in the point set has the same status to participate in the search calculation, and there is still room for the improvement of the execution efficiency. Based on the algorithms proposed in literatures [14, 15], this paper puts forward a grid partition variable step Alpha Shapes (GPVAS) algorithm, which has higher computational efficiency while solving the problems caused by nonuniform distribution of point sets.

### 3. Grid Partition Variable Step Alpha Shapes Algorithm

#### 3.1. Algorithm Overview

The main improvements of the GPVAS algorithm are as follows:

- (1) The point set  $S$  is partitioned, the nonboundary grid area is removed by fast filtering, and the boundary grid point set  $S_G$  is extracted.
- (2) For points in point set  $S_G$ , within the range of point set  $S$ , variable step Alpha Shapes (VAS) algorithm [20] is applied to construct the boundary shape of point set  $S$ . Figure 4 shows the simplified flow chart of GPVAS algorithm.

3.2. *Extracting Boundary Grid Partition Point Set.* This step consists of two steps: “grid partition” and “extracting point set of boundary grid partition.” The steps are as follows.

3.2.1. *Grid Partitions.* The envelop rectangle  $G$  of point set  $S$  is constructed, and its inflexion points are  $G_1(X_{\min}, Y_{\min})$ ,  $G_2(X_{\min}, Y_{\max})$ ,  $G_3(X_{\max}, Y_{\max})$  and  $G_4(X_{\max}, Y_{\min})$ , respectively.

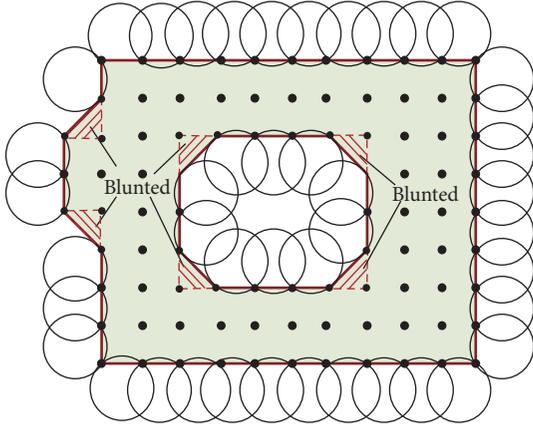


FIGURE 3: Building boundary of point set by Alpha Shapes algorithm.

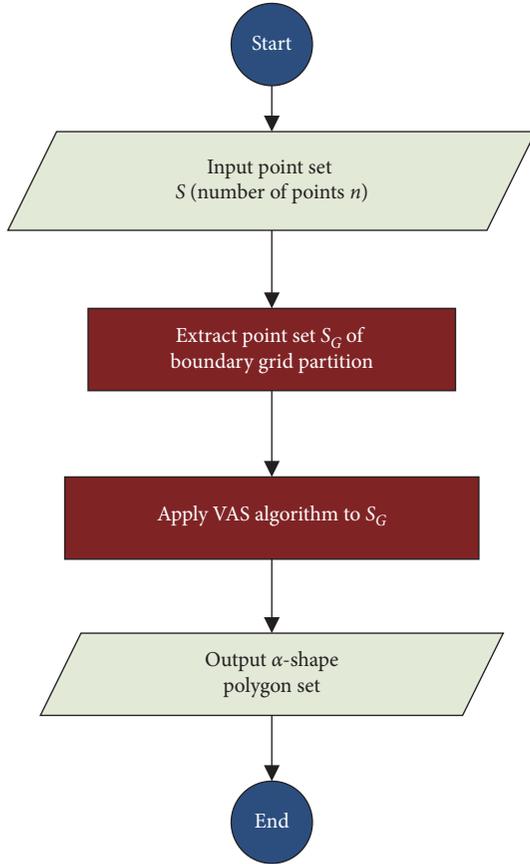


FIGURE 4: Simplified flow chart of GPVAS algorithm.

The envelop rectangle  $G$  is divided into  $M \times N$  grid partition set  $G_0$  by the square grid with side length of  $d$  (usually 2-3 times the average point spacing of point set  $S$ ), where

$$\begin{cases} M = \left\lceil \frac{Y_{\max} - Y_{\min}}{a} \right\rceil + 1, \\ N = \left\lceil \frac{X_{\max} - X_{\min}}{a} \right\rceil + 1, \end{cases} \quad (2)$$

where  $M$  is the number of rows,  $N$  is the number of columns, and the symbol  $\lceil \cdot \rceil$  is the integer of real numbers.

3.2.2. *Extracting Point Set of Boundary Grid Partition.* A row-column index table is established for the point set  $S$ , and the formula for calculating the row-column index value of point  $P_i$  is as follows:

$$\begin{cases} r_i = \left\lceil \frac{Y_i - Y_{\min}}{a} \right\rceil, \\ c_i = \left\lceil \frac{X_i - X_{\min}}{a} \right\rceil, \end{cases} \quad (3)$$

where  $r_i$  is the row number in the grid partition where point  $P_i$  is located and  $c_i$  is the column number in the grid partition where point  $P_i$  is located. Thus, the mapping relationship between point and grid partition is established, as shown in Figure 5.

Traverse the grid, and quickly determine whether the grid contains a point from the row-column index table of point set  $S$ . If it contains a point, it is 1; otherwise, it is 0.

Traverse the grid, extract the point set  $G_1$  of boundary grid partition (the diagonal filling grid shown in Figure 5), and then quickly obtain the boundary grid point set  $S_G$  through the row-column index table of the point set  $S$ . The judgment rule of the boundary grid is that the boundary grid contains points, and at least one of its eight adjacent grids does not contain any point.

3.3. *Extracting Boundary Grid Partition Point Set.* Once the value of parameter  $\alpha$  in AS algorithm (i.e., the search step in constructing  $\alpha$ -shape) is set, it is constant throughout the boundary construction process, which is the reason for the unsatisfactory effect of the algorithm in dealing with the nonuniform distribution point set or point set containing concave points. Literature [20] proposed VAS algorithm and used kd-tree to calculate the average distance of  $k$ -nearest neighbors of the point as  $\alpha$  value to participate in the construction of  $\alpha$ -shape. The average distance of  $k$ -nearest neighbors of a point is the average distance between the nearest  $k$  number of points and the point in a point set. At this point, the value of  $\alpha$  is variable; when the point density is large, the  $\alpha$  value is small, which ensures the continuity of the boundary and the high  $\alpha$ -shape construction efficiency into account; when the point density is small, the  $\alpha$  value becomes larger, which can prevent boundary fragmentation caused by the excessively small  $\alpha$  value. However, in the process of calculating  $\alpha$  value and applying  $\alpha$  value to construct  $\alpha$ -shape, the VAS algorithm needs to search the entire  $S$  point set under the worst condition. The time complexity is  $O(n^3)$ , and the efficiency of the algorithm is not ideal.

The GPVAS algorithm proposed in this paper still adopts the calculation steps of VAS algorithm, first calculating the  $\alpha$  value and then applying the  $\alpha$  value to construct the  $\alpha$ -shape. The biggest improvement of GPVAS algorithm lies in the grid partition and point set row-column index table

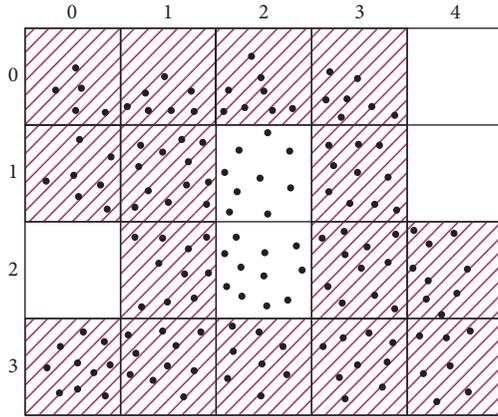


FIGURE 5: Extracting point set of boundary grid partition by GPVAS algorithm.

constructed based on the above steps. No matter whether it is calculating  $\alpha$  value or constructing  $\alpha$ -shape, it is convenient to take the grid where the target point or center point is located as the center and search layer by layer from small to large distance (as shown in Figure 6), which is equivalent to arranging the point set  $S$  in ascending sort order according to the distance value from the target point or center point. The experimental results show that the efficiency of the algorithm is significantly improved.

#### 4. Comparative Experiments

To intuitively verify the time efficiency of the algorithm, two kinds of data are designed for experimental verification based on AS algorithm, VAS algorithm and GPVAS algorithm: one is the data point set of computer numerical simulation (hereinafter referred to as the simulated point set), and the other is the data point set of engineering measurement (hereinafter referred to as the measured point set).

**4.1. Comparison of Simulated Point Sets.** The data of random point set generated randomly by circular analytic formula (4) according to the density of the upper semicircle point is 3 times that of the lower semicircle point, and the inner point of the upper and lower semicircle is generated randomly according to the uniform distribution, as shown in Figure 7.

$$S = \{(x, y) | (x - 1)^2 + (y - 1)^2 \leq 1\}. \quad (4)$$

The effective range of parameter  $k$  in the calculation of the average distance of  $k$ -nearest neighbors in VAS algorithm and GPVAS algorithm is [9, 24]. According to the experimental statistics in literature [20],  $k = 20$  is considered to be more ideal. In this paper,  $k = 20$  is also used to calculate the nearest neighbor average distance as  $\alpha$ , and  $2\alpha$  is taken as the grid edge length. The experimental laptop is configured as Intel(R) Core(TM) i7-10750H CPU@2.60 GHz, 16 G memory, and 64-bit Windows 10 operating system; experimental algorithm program is based on Microsoft Visual Studio 2010 IDE, C# language; the simulation point set is

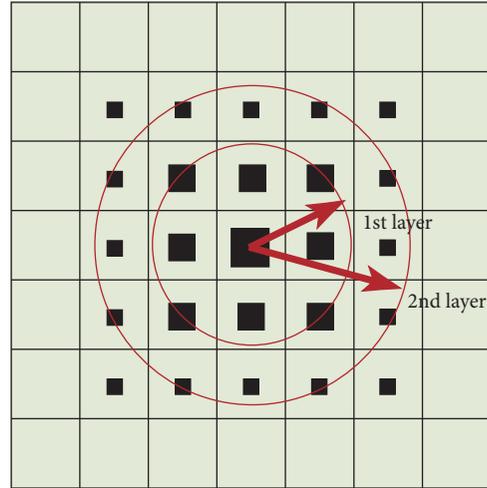


FIGURE 6: Sketch map of layer-by-layer search by GPVAS algorithm.

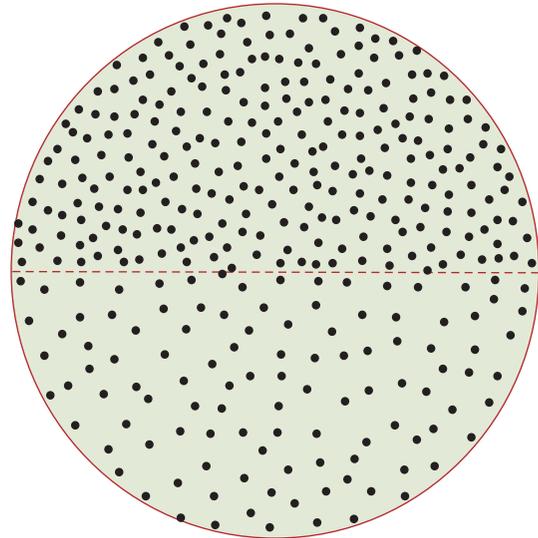


FIGURE 7: Circular nonuniform simulation point set.

divided into four control groups (1000, 5000, 10000, and 20000) according to the number of points. The experimental results are shown in Table 1.

According to the comparison of experimental results in Table 1, when the number of point sets is small, the efficiency of VAS algorithm and GPVAS algorithm is lower than that of AS algorithm. The order of efficiency is as follows: AS > VAS > GPVAS. This is because the VAS algorithm and GPVAS algorithm need to spend some time to dynamically calculate the average distance of  $k$ -nearest neighbors as  $\alpha$  value before each execution of the  $\alpha$ -shape criterion to filter the boundary points. In addition, GPVAS also needs to establish a grid partition and row-column index table for the point set. As the number of points increases, the execution efficiency of VAS algorithm and GPVAS algorithm begins to improve. Compared with AS algorithm, the execution efficiency of VAS algorithm can be improved by 20%~30%. When the number of points increases, the efficiency of

TABLE 1: Comparison table of experimental results of different algorithms (simulated point set).

Algorithm	Number of points	$\alpha$ value (min/max)	Execution time (ms)
AS algorithm	1000	0.360	245
	5000	0.072	5128
	10,000	0.036	26124
	20,000	0.018	128356
VAS algorithm	1000	0.240/0.781	326
	5000	0.048/0.156	4289
	10,000	0.024/0.075	19356
	20,000	0.012/0.039	96786
GPVAS algorithm	1000	0.240/0.781	861
	5000	0.048/0.156	3243
	10,000	0.024/0.075	6934
	20,000	0.012/0.039	10189



FIGURE 8: Boundary of strip terrain points on mountain highway.

TABLE 2: Comparison table of experimental results of different algorithms (measured point set).

Algorithm	$\alpha$ value (min/max)	Execution time (ms)
AS algorithm	10 m	16287
AS algorithm	30 m	21458
VAS algorithm	8.6 m/41.2 m	11356
GPVAS algorithm	8.6 m/41.2 m	7127

GPVAS algorithm increases rapidly. When the number of points is 20,000, the execution efficiency increases by more than 10 times. The order of the execution efficiency of the algorithm is  $GPVAS > VAS > AS$ . This is because the time-consuming distance budget time complexity of the AS algorithm is  $O(n^3)$ , and the GPVAS algorithm is able to construct the grid partition and row-column index table of the point set with negligible extra storage space before the  $\alpha$ -shape criterion filtering the boundary point set with the time complexity of  $O(n^2)$ , so as to quickly extract the boundary point set. Compared with the increase in the total number of point sets, the increase in the number of boundary point sets constructed by grid partitioning is very limited.

#### 4.2. Comparative Study of Strip Terrain Points on Mountain Highway.

In order to verify the effectiveness of the

algorithm applied to the measured point set, this paper selects the highway strip terrain data (Figure 8), which is composed of more than 8000 GPS measuring points with a total length of about 17 km in the mountainous area of southern Anhui, and the average distance between the measuring points is 28.513 m. There are many curves in the strip terrain data in mountainous areas, dense collection points in undulating sections, and sparse collection points in flat sections. The density distribution of point set is not uniform, and there are many concave areas, so it is an ideal experimental data to verify the algorithm. The experimental results are shown in Table 2; Figures 8 and 9(a) to Figure 9(c) (Figures 9(a)–9(c)) are the enlarged images of the boundaries extracted by different algorithms at highway curve in Figure 8.

The experimental results show that all the three algorithms can be used to construct the boundary of experimental data, and the construction efficiency is

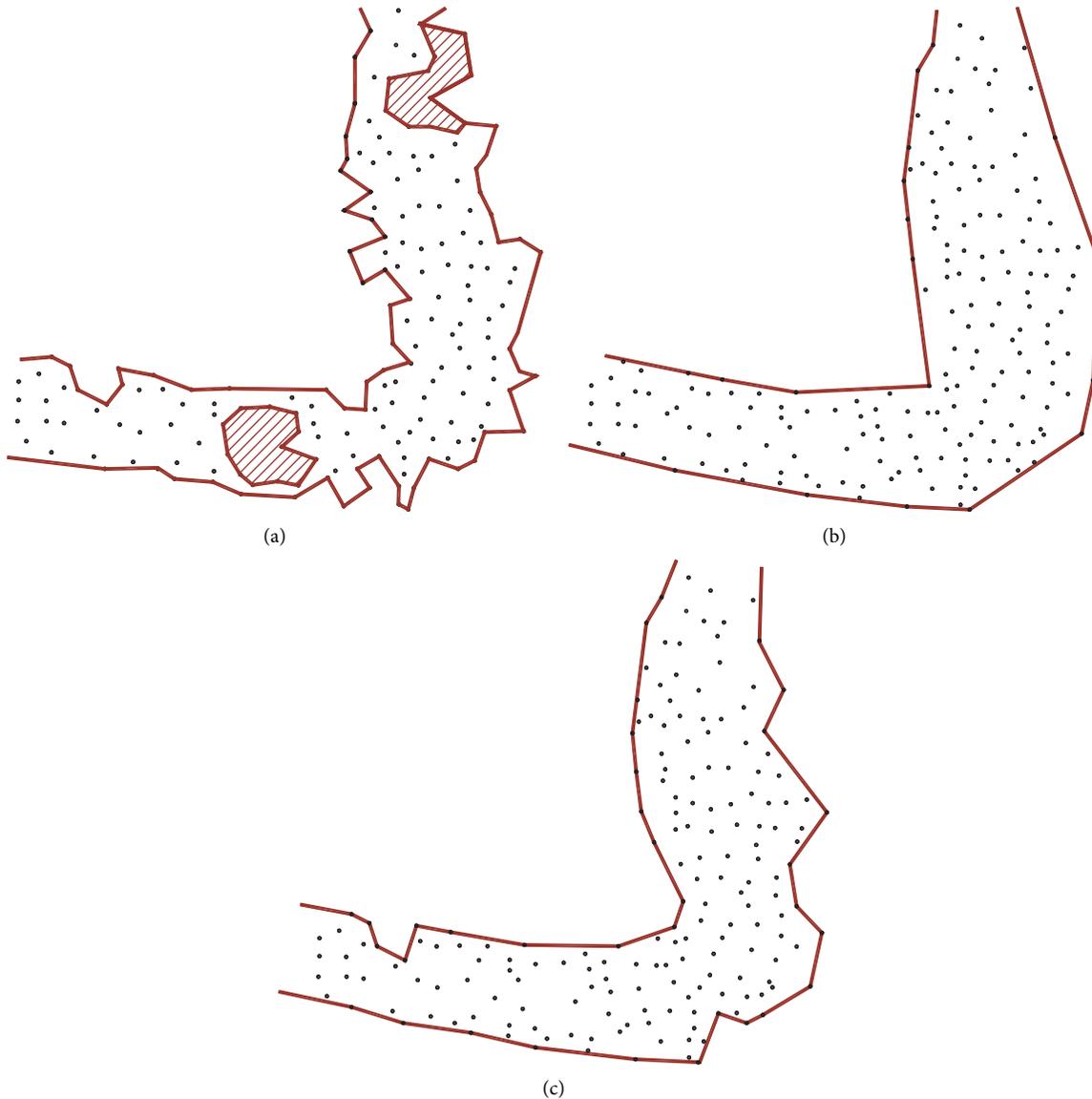


FIGURE 9: Boundaries extracted by different algorithms at highway curve. (a) AS algorithm,  $\alpha = 10$  m. (b) AS algorithm,  $\alpha = 30$  m. (c) VAS, GPVAS algorithm,  $\alpha_{\min} = 8.6$  m,  $\alpha_{\max} = 41.2$  m.

GPVAS > VAS > AS. Regarding the construction results, GPVAS algorithm and VAS algorithm are almost the same with good effect (Figure 9(c)), while the AS algorithm is greatly affected by the set value of  $\alpha$ . When the value of  $\alpha$  is small (10 m in the experiment), the boundary is relatively fine, and it is easy to form the boundary of holes in the area of sparse points (Figure 9(a), where the diagonal filling area is determined as the hole). When the value of  $\alpha$  is large (30 m in the experiment), the boundary is relatively crude (Figure 9(b)), and the algorithm execution efficiency is lower than that when the value of  $\alpha$  is small.

## 5. Conclusions

Based on the Alpha Shapes algorithm for extracting the boundary of discrete point sets, this paper analyzes and

summarizes the previous research work. In view of the shortcomings of Alpha Shapes algorithm in processing non-uniform distributed point sets and multiconcave point sets, this paper proposes the grid partition variable step Alpha Shapes algorithm, which is used to quickly construct the boundary of point sets. This algorithm has two main advantages:

- (1) Establish grid partition and row-column index table for point set, quickly filter nonboundary point partition, and extract boundary grid partition point set involved in subsequent  $\alpha$ -shape construction; compared with the increase in the total number of point sets, the increase in the number of point sets of boundary grid partition constructed by grid partition is very limited, which is the main reason why GPVAS algorithm can effectively deal with a large number of point sets.

- (2) The average distance of  $k$ -nearest neighbors of the point calculated by kd-tree is used as the  $\alpha$  value. In the region with dense point distribution, the  $\alpha$  value is small, and, in the region with sparse point distribution, the  $\alpha$  value is large, so that the algorithm can well deal with the regional boundary with nonuniform point distribution.

The algorithm is verified by simulated point set and measured point set, and the execution efficiency of the algorithm is very high. Compared with similar algorithms, the larger the number of point sets is, the more obvious the efficiency improvement is. As an alternative algorithm, this algorithm has been effectively verified in engineering scenarios such as land area statistics and road earthwork calculation. In the field of 3D point cloud surface reconstruction with broader application scenarios, this algorithm has not been verified, which is also the follow-up research direction of this paper.

### Data Availability

The data used to support the findings of this research were generated from experiments.

### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

### Acknowledgments

This research was funded by the National Natural Science Foundation of China (41906168), Natural Science Research Project of Anhui Education Department (KJ2018JD04), and AHJZU-Anhui Huali Construction Co., Ltd., Joint Research Project (HYB20190152).

### References

- [1] B. Xiong, M. Jancosek, S. O. Elberink, and G. Vosselman, "Flexible building primitives for 3D building modeling," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 101, pp. 275–290, 2015.
- [2] B. Yang, R. Huang, Z. Dong, Y. Zang, and J. Li, "Two-step adaptive extraction method for ground points and breaklines from lidar point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 119, pp. 373–389, 2016.
- [3] B. S. Yang, W. X. Dai, Z. Dong, and Y. Liu, "Automatic forest mapping at individual tree levels from terrestrial laser scanning point clouds with a hierarchical minimum cut method," *Remote Sensing*, vol. 8, no. 5, 372 pages, 2016.
- [4] B. S. Yang, F. X. Liang, and R. G. Huang, "Progress, challenges and perspectives of 3D lidar point cloud processing," *Acta Geodaetica et Cartographica Sinica*, vol. 46, no. 10, pp. 1509–1516, 2017.
- [5] Y. Z. Sun, J. Li, B. Liu et al., "Measurement of agricultural machinery operation area based on improved alpha shapes algorithm," *Journal of Chinese Agricultural Mechanization*, vol. 40, no. 8, pp. 144–148, 2019.
- [6] Z. X. Liao and D. L. Wang, "Research on DTM TIN generation algorithm considering border restriction," *Journal of Hefei University of Technology (Natural Science Edition)*, vol. 34, no. 9, pp. 1381–1384, 2011.
- [7] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.
- [8] X. Xu and K. Harada, "Automatic surface reconstruction with alpha-shape method," *The Visual Computer*, vol. 19, no. 7–8, pp. 431–443, 2003.
- [9] S. N. Karl and Q. N. Truong, "Image super-resolution using support vector regression," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1596–1610, 2007.
- [10] H. Ganapathy, P. Ramu, and R. Muthuganapathy, "Alpha shape based design space decomposition for island failure regions in reliability based design," *Structural and Multidisciplinary Optimization*, vol. 52, no. 1, pp. 121–136, 2015.
- [11] R. Graham, "An efficient algorithm for determining the convex hull of a planar set," *Information Processing Letters*, vol. 1, no. 1, pp. 132–133, 1972.
- [12] R. A. Jarvis, "Computing the Shape Hull of Points in the Plane," in *Proceedings of the IEEE Computing Society Conference on Pattern Recognition and Image Processing*, pp. 231–241, IEEE, Troy, NY, USA, 1977.
- [13] A. Sampath and J. Shan, "Building boundary tracing and regularization from airborne LiDAR point clouds," *Photogrammetric Engineering & Remote Sensing*, vol. 73, no. 7, pp. 805–812, 2007.
- [14] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 551–559, 1983.
- [15] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Transactions on Graphics*, vol. 13, no. 1, pp. 43–72, 1994.
- [16] A. Jochem, B. Höfle, M. Rutzinger, and N. Pfeifer, "automatic roof plane detection and analysis in airborne lidar point clouds for solar potential assessment," *Sensors*, vol. 9, no. 7, pp. 5241–5262, 2009.
- [17] W. Shen, J. Li, Y. H. Chen, D. Lei, and G. X. Peng, "Algorithms study of building boundary extraction and normalization based on lidar data," *Journal of Remote Sensing*, vol. 12, no. 5, pp. 692–698, 2008.
- [18] Y. F. Li, D. B. Tan, G. Gao, and L. Rui, "Extraction of building contour from point clouds using dual threshold alpha shapes algorithm," *Journal of Yangtze River Scientific Research Institute*, vol. 33, no. 11, pp. 1–4, 2016.
- [19] Z. Y. Wang, H. C. Ma, H. G. Xu, and Y. Zhiwei, "Novel algorithm for fast extracting edges from massive point clouds," *Computer Engineering and Applications*, vol. 46, no. 36, pp. 213–215, 2010.
- [20] S. L. Li and H. J. Li, "Surface reconstruction algorithm using self-adaptive step alpha-shape," *Journal of Data Acquisition and Processing*, vol. 34, no. 3, pp. 491–499, 2019.
- [21] Q. Li, X. W. Gao, X. Y. Fei et al., "Construction of tree crown three-dimensional model of using alpha-shape algorithm," *Bulletin of Surveying and Mapping*, vol. 2018, no. 12, pp. 91–95, 2018.
- [22] Y. X. Fu, C. M. Li, J. Zhu, W. Baolong, Z. Bin, and F. Wei, "Three-dimensional model construction method and experiment of jujube tree point cloud using alpha-shape algorithm," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 36, no. 22, pp. 214–221, 2020.
- [23] H. Zhang, Y. N. Wen, A. L. Liu et al., *GIS Algorithm Basis*, Science Press, Beijing, China, 2006.
- [24] H. J. Li, X. Y. Liu, X. P. Zhang, and D. Yan, "A semi-automatic 3D point cloud classification method based on the probability mixture of local shape features," *Journal of Zhejiang University: Science Edition*, vol. 44, no. 1, pp. 1–9, 2017.