Computers and Geosciences 109 (2017) 178–193

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

A density based algorithm to detect cavities and holes from planar points

Jie Zhu[a,b], Yizhong Sun[a,b,*], Yueyong Pang[a,b]

[a] Key Laboratory of Virtual Geographic Environment, Nanjing Normal University, Ministry of Education, Nanjing 210023, China
[b] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China

ARTICLE INFO

Keywords:
Cavity and hole
Shape reconstruction
Delaunay triangulation
Planar point
GIS

ABSTRACT

Delaunay-based shape reconstruction algorithms are widely used in approximating the shape from planar points. However, these algorithms cannot ensure the optimality of varied reconstructed cavity boundaries and hole boundaries. This inadequate reconstruction can be primarily attributed to the lack of efficient mathematic formulation for the two structures (hole and cavity). In this paper, we develop an efficient algorithm for generating cavities and holes from planar points. The algorithm yields the final boundary based on an iterative removal of the Delaunay triangulation. Our algorithm is mainly divided into two steps, namely, rough and refined shape reconstructions. The rough shape reconstruction performed by the algorithm is controlled by a relative parameter. Based on the rough result, the refined shape reconstruction mainly aims to detect holes and pure cavities. Cavity and hole are conceptualized as a structure with a low-density region surrounded by the high-density region. With this structure, cavity and hole are characterized by a mathematic formulation called as compactness of point formed by the length variation of the edges incident to point in Delaunay triangulation. The boundaries of cavity and hole are then found by locating a shape gradient change in compactness of point set. The experimental comparison with other shape reconstruction approaches shows that the proposed algorithm is able to accurately yield the boundaries of cavity and hole with varying point set densities and distributions.

1. Introduction

Shape reconstruction from planar points is a significant and fundamental operation in various GIS-related applications, such as cartographic generalization (Galton and Duckham, 2006; Moreira and Santos, 2007), boundary extraction from 2-D point cloud (Zhu et al., 2008; Shen et al., 2011), domain boundaries (Arampatzis et al., 2006; Kolingerová and Žalik, 2006), and geographic information retrieval (Duckham et al., 2008; Bordogna et al., 2012). Several problems emerge from approximating the boundary of a planar point set. First, the use of spatial relation between points is difficult for an optimal approximation model. Second, non-uniformly distributed point set occurs generally to reach an unsatisfactory boundary because of the global measure. Third, the accurate detection of multiple holes and cavities is another difficult task caused by the lack of effective mathematical expression.

With the consideration of Gestalt law (Li et al., 2004), the proximity relationships of points play a significant role in shape reconstruction geometric model. The Delaunay Graph conforms to the Gestalt laws of proximity and continuity, which is widely regarded as a good tool for shape reconstruction. In $\mathbb{R}^2$, several approaches to shape reconstruction through Delaunay filtering have been proposed. However, these approaches cannot define the boundary accurately given one or more of the above approximate problems. For example, Kolingerová and Žalik (2006) adopted a global measure (the length of edges) to remove the longest edges at each iteration; however, the measure was seriously influenced by density variation. Duckham et al. (2008) proposed an algorithm that can characterize the shape of different point densities and distributions; however, this algorithm failed to detect holes. Peethambaran and Muthuganapathy (2015) presented an automatic algorithm for cavities and holes detection; however, this is limited to multiple cavities and holes.

In this paper, we propose a new Delaunay-based approach for shape reconstruction that considers cavity and hole structures. This method has two procedures. The first reaches a rough boundary without holes and pure cavities, and the second achieves the final boundary with precise holes and cavities. Two key points of our approach are worth putting forward. First, we construct an appropriate and practical data structure to facilitate boundary manipulation in the succeeding procedures. Second, cavity and hole are conceptualized as a low-density region surrounded by the high-density region. With such density variation, we derive a

[*] Corresponding author. No.1 of Wen Yuan Road, School of Geography in Nanjing Normal University, Nanjing 210023, Jiangsu, China.
E-mail address: sunyizhong_cz@163.com (Y. Sun).

http://dx.doi.org/10.1016/j.cageo.2017.08.008
Received 28 December 2016; Received in revised form 1 August 2017; Accepted 7 August 2017
Available online 1 September 2017

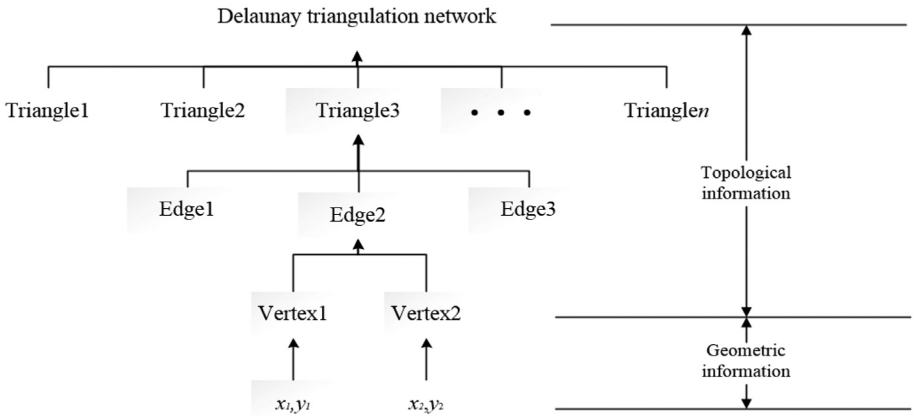


Fig. 1. Data structure used in this study.

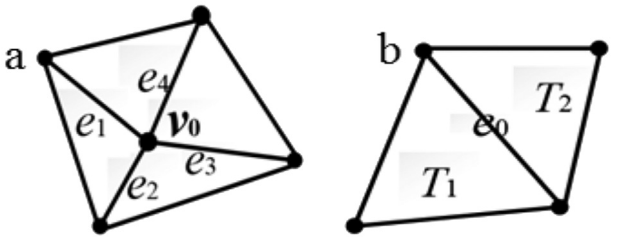


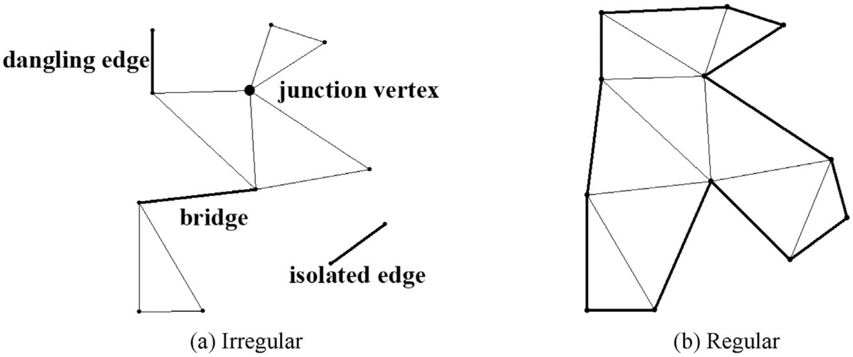
Fig. 2. Illustrations of: (a) Incident edges of a vertex v_0; (b) Triangles of a common edge e_0.

mathematic formulation for the two structures and design a statistical method to detect them.

2. Related work

Several works have been proposed in the literature for shape reconstruction from planar points. We divide these studies into two main approaches, namely, Delaunay-based and curve reconstruction method.

The Delaunay-based method includes two steps: (1) First, it builds a geometric structure from a set of planar points; (2) it then extracts a series of edges with such structure to reach a desirable boundary. The early and well-known work is the Delaunay sculpture by Boissonnat (1984) and the α-shape by Edelsbrunner et al. (1983). The α-shape obtained the boundary by rolling trace with an empty disk of radius α. The α-shape has been successfully applied in uniform distribution, but it suffers from non-uniform density caused by global search. To solve this problem, a family of α-shape has been proposed, including r-shape (Chaudhuri et al., 1997), A-shape (Melkemi and Djebali, 2000), k-orderα-hulls (Krasnoshchekov et al., 2010) and LDA-α-complex (Chevallier and Maillot, 2011). Most of these studies methods are related to certain external parameters and rarely involved the boundary topological examination. The Delaunay sculpture employs an iterative removal process on triangulation network. In this context, the method captures geometric and topological information of points nicely, which is simpler and systematic compared to other methods. Thereafter, many researchers approximated the boundary based on Boissonnat's sculpture, but used different removal strategies. Duckham et al. (2008) characterized the shape by removing exterior edges in order of length until no edge is longer than the length parameter. The algorithm can yield an accurate shape of a wide range of different point densities. Furthermore, the algorithm focuses only on external boundary. Peethambaran and Muthuganapathy (2015) present an automatic algorithm that combines the circumcenter and circumradius of Delaunay triangles to filter Delaunay triangulation. In this context, the algorithm constructs the boundaries for the holes based on the structural arrangement of Delaunay triangles. Although the method is innovative, the attempt is not sufficiently convincing for the thin holes (one is elongated more like a tubular structure) detection. Given that proximity graphs play a vital role in defining the shape and organization of planar points (Jaromczyk and Toussaint, 1992), the Delaunay triangulation subgraphs and the dual graph (Voronoi Diagram) have been also applied to shape reconstruction such as the Gabriel graph (Park et al., 2006), simple-shape (Gheibi et al., 2011) and the minimum spanning tree (Ohrhallinger and Mudur, 2013). The Delaunay-based method can capture geometric information in a set of planar points and several algorithms even consider topological examination to guarantee the topological correctness of boundary. However, these algorithms lack of the stability to approximate the actual boundary in face of complex boundaries, especially for multiple cavities and holes.



(a) Irregular

(b) Regular

Fig. 3. Illustrations of irregular and regular of a simplicial 2-complex.

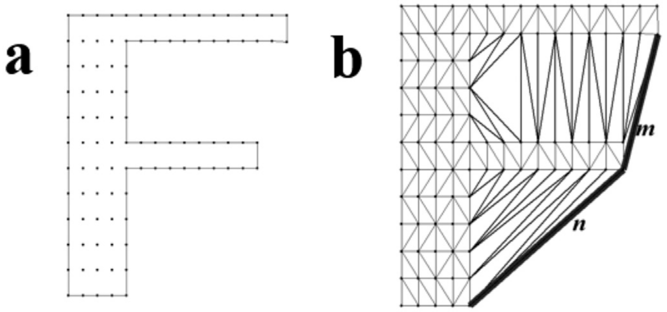


Fig. 4. Illustrations of: (a) Sample points of cavity bounded by a simple polygon; (b) Cavity edges and virtual edges.

In a different method, curve reconstruction takes a set of points on a smooth closed curve, identifies the points of curvature discontinuity, and connects these points to the boundary. Such approaches largely rely on the representation of curve fitting. Related techniques that compute the curvature of the underlying shape have been proposed, which include implicit simplicial model (Fang and Gossard, 1995), quadratic polynomial (Yang and Lee, 1999) and moving least-squares (Lee, 2000). Compared with the Delaunay-based approach, most of these methods are faster, especially for point cloud data. However, these methods cannot guarantee the topological correctness of the boundary because of the lack of topological information, and the computation processes are complicated.

3. Preliminaries

3.1. Data structure

An appropriate and practical data structure is vital to represent geometric information and topological relations for the boundary manipulation, especially for a quick access to boundary edges and topology examination.

In this paper, we adopt a hierarchical data structure (Fig. 1) to represent the topological relation, namely, vertex-edge-triangle. Each vertex records the x, y coordinates, and the links between vertices in the Delaunay triangulation express geometric information. The topological information includes Vertex-edge relation (each edge consists of two vertices) and edge-triangle (each triangle consists of three edges) relation. This study adopted an approach provided by Green and Sibson (1978) to construct the Delaunay triangulation. To efficiently manipulate the boundary, each vertex records its incident edges (see Fig. 2(a)) and each edge records a list of triangles using this edge as the common edge (see Fig. 2(b)) when constructing the Delaunay triangulation. In the past, various data structures have been proposed for boundary representation. One, the winged-edge structure (Weiler, 1985), is familiar to many. The topological information stored in the winged-edge structure for each edge is composed of the adjacencies of the given edge with other edges, vertices, and faces. The winged-edge structure is not sufficient because its information content does not consider the derivation of some important adjacency information (e.g., the adjacency of edges around a vertex). The hierarchical data structure represent the adjacency information by splitting it into two structure (vertex-edge and edge-triangle), which can allow for the derivation of adjacency information. Moreover, information organized hierarchically allows to improve time efficiency.

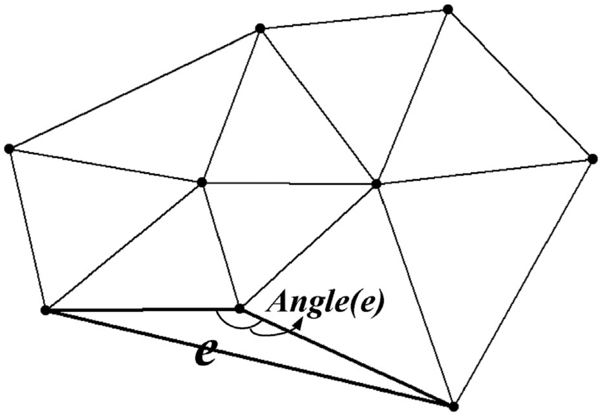


Fig. 7. An example of boundary angle.

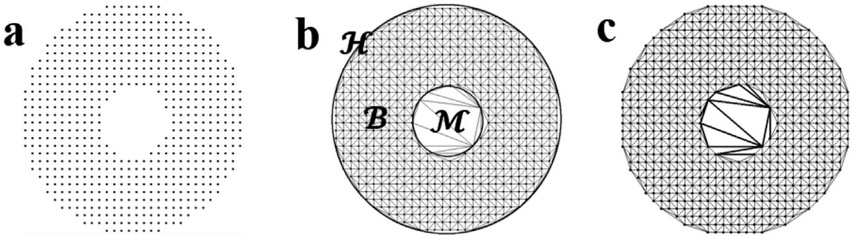


Fig. 5. Illustrations of: (a) Sample points of hole; (b) Hole polygon; (c) Hole edges.

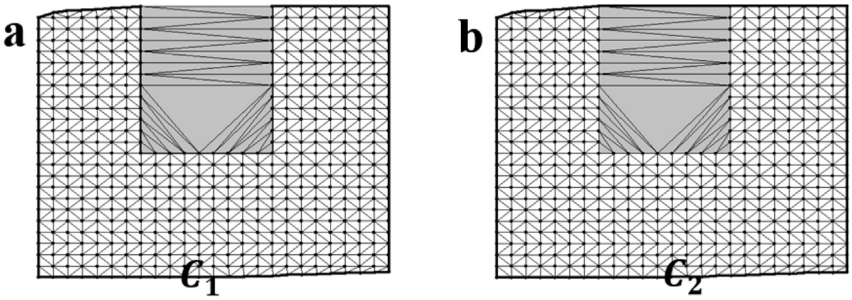


Fig. 6. Illustrations of: (a) Line segments in cavity; (b) Line segments in hole.

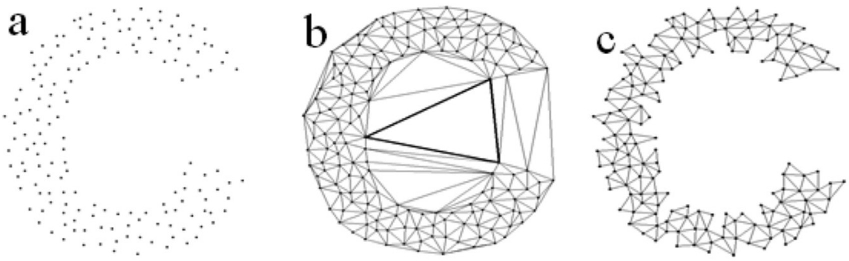


Fig. 8. Example of cavity with satisfactory triangle: (a) Point set; (b) Cavity with satisfactory triangle; (c) Boundary with sharp corners.

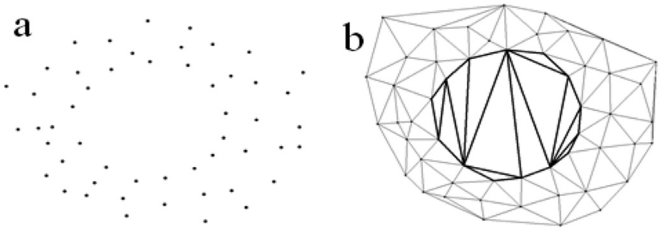


Fig. 9. An example of a hole: (a) Point set; (b) Hole.

3.2. Related geometrical definition

Definition 1. Delaunay triangulation:

Let S be a set of unorganized points in$\mathbb{R}^2$. The Voronoi cell of a point $p \in S$, denoted as $\mathrm{V}_p(S)$, is the set of points$x \in \mathbb{R}^2$ that are close to p than any other point in S:

$$\mathrm{V}_p(S) = \left\{x \in \mathbb{R}^2 | \|x - p\| \leq \|x - q\|, q \in S,\ q \neq p\right\}$$

The union of the Voronoi cells of all points $s \in S$ forms the Voronoi diagram of$S$, denoted as

$$VD(S) = \cup \mathrm{V}p(S), p \in S$$

The dual of the Voronoi diagram obtained by connecting the points S, which associated polygons are adjacent, is called Delaunay triangulation of S, $DT(S)$ (Boissonnat and Oudot, 2005).

To define the topological space of Delaunay triangulation graph, the concept of simplicial complex is adopted in this paper. Vertices, edges and triangles are all examples of simplices. An n-dimensional simplex is called an n –simplex. A 0-simplex is a point, a 1-simplex is a line segment and a 2- simplex is a triangle. A collection of simplices forms a simplicial complex (Definition 2).

Definition 2. Simplicial complex:

Let E be a linear space. A simplicial complex K is a finite collection of simplices in E if K satisfies the following two conditions:

(1) Any face of a simple from K is also in K;
(2) The intersection of any two simplices $\mathrm{C}, D \in K$ is either empty or a face of both C and D

A two dimensional Delaunay triangulation is an example of simplicial 2-complex. In a simplicial 2-complex, if line segments do not belong to any triangle, line segments are considered as either dangling edges, bridges or isolated edges; if more than two line segments which are attached to a single triangle are linked through one vertex, the vertex is called junction vertex (See Fig. 3).

Definition 3. Irregularity of a simplicial 2-complex:

A simplicial 2-complex is irregular if it suffers from dangling edges, bridges or junction vertices (Peethambaran and Muthuganapathy, 2015). We can use the proposed data structure to carry on the irregularity readily for topological examination.

Definition 4. Boundary edge:

An edge is a boundary edge of $DT(S)$ if it only belongs to one triangle, namely, the edge is not the common edge.

The boundary of Delaunay triangulation is represented by a collection of boundary edges.

Definition 5. Boundary vertex:

A vertex is a boundary vertex of $DT(S)$ if it is incident with a boundary edge. All other vertices are referred to as interior vertices.

Definition 6. Boundary triangle:

A triangle is a boundary triangle of $DT(S)$ if it consists of at least one boundary edge.

According to Gestalt law of proximity, the boundary is formed by connecting neighboring points. Thus, two types of boundary triangles exist in $DT(S)$. An obtuse triangle (the circumradius of such triangle that lies external to it) always shows too big angle, making two boundary vertices non-neighbors on the boundary. Such triangle is considered to be unsatisfactory triangle (Kolingerová and Žalik, 2006), whereas all other triangles are satisfactory triangles.

Definition 7. Cavity:

Let $Q \subset \mathbb{R}^2$ be a simple polygon bounding all sample points P (see Fig. 4(a)), $conv(P)$ be the convex hull of P, and C be the set of all open

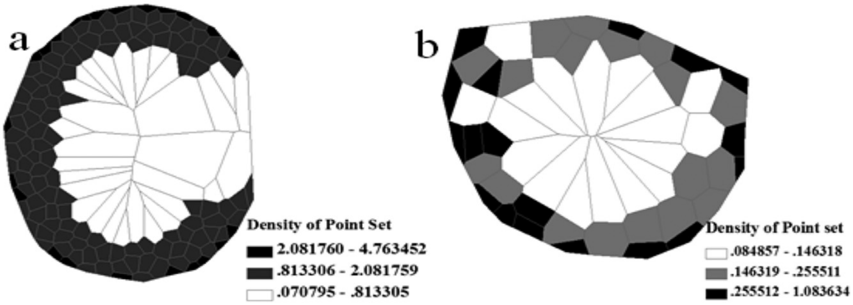


Fig. 10. Density variation of cavity and hole by Voronoi: (a) Cavity; (b) Hole.

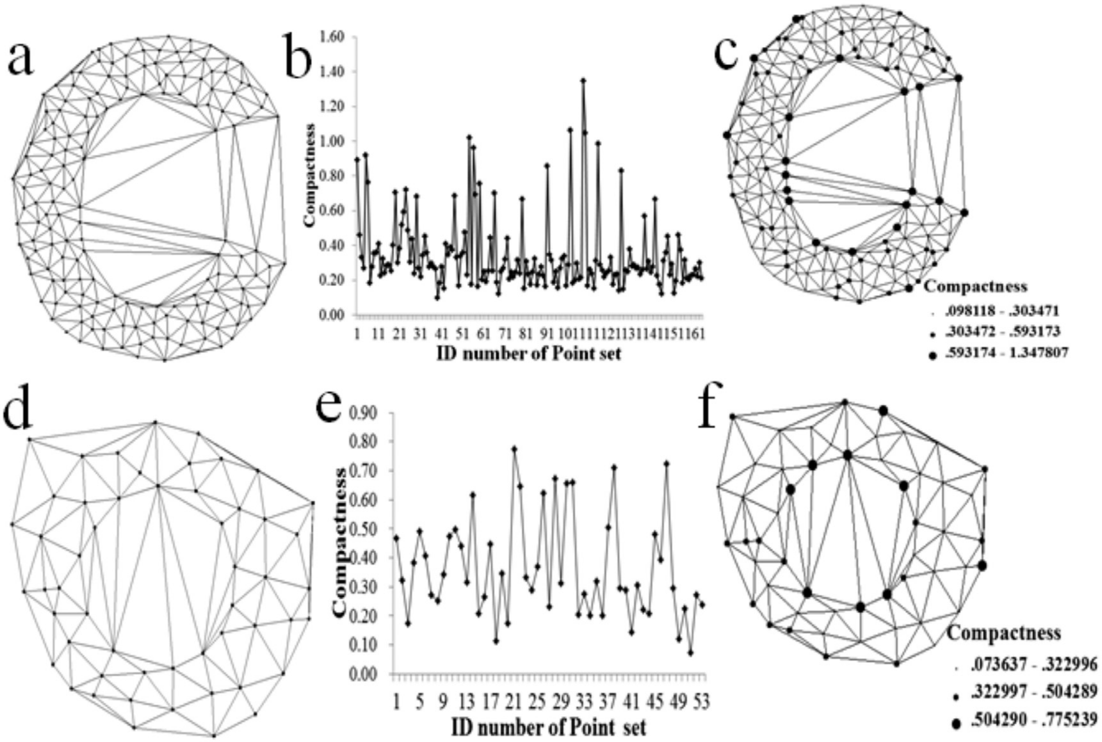


Fig. 11. Compactness of two structures:(a) Delaunay triangulation of a cavity; (b) Compactness of point set in cavity; (c) Compactness variation of cavity; (d) Delaunay triangulation of a hole; (e) Compactness of point set in hole; (f) Compactness variation of hole.

connected regions of $conv(P)\setminus Q$. Each polygon given by closure $\overline{C}$ is referred as a cavity (Mistry et al., 2014). The edges of Q in each cavity are called cavity edges (bold black lines except $\boldsymbol{m}$ and $\boldsymbol{n}$ in Fig. 4(b)), and the edges of $conv(P)$ in the cavities are called virtual edges (line segment $\boldsymbol{m}$ and $\boldsymbol{n}$ in Fig. 4(b)). Each cavity is composed of several cavity edges and one virtual edge.

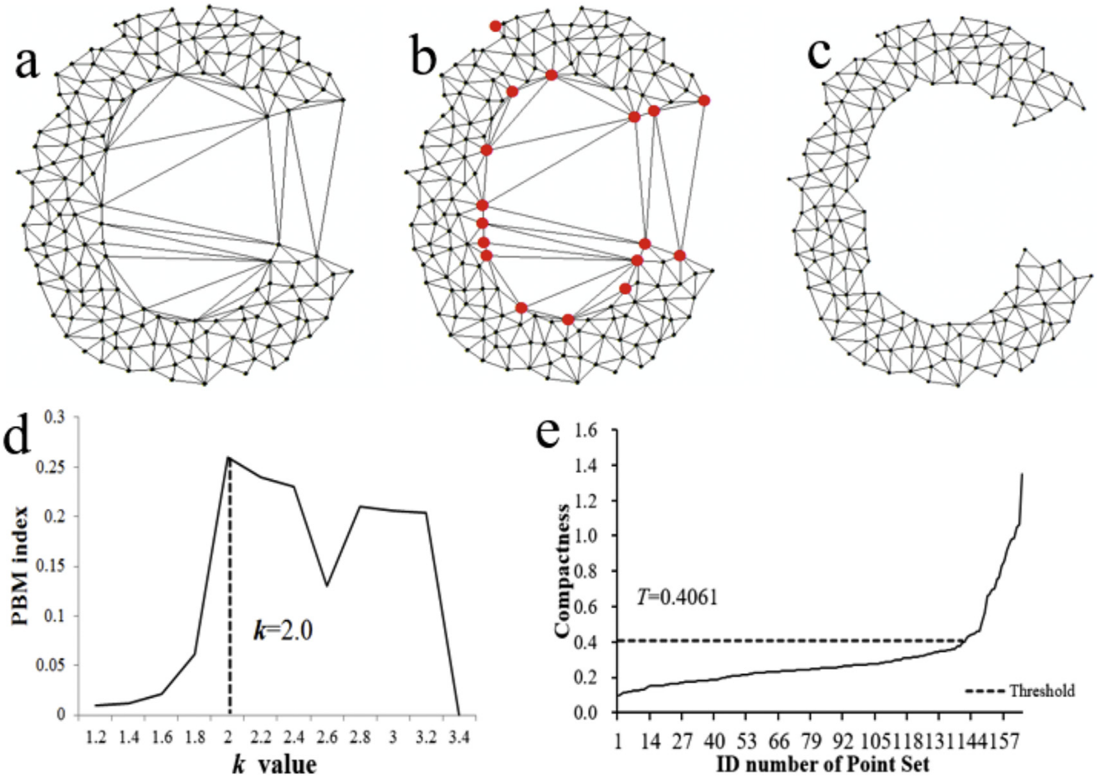


Fig. 12. Different stages of the proposed algorithm for cavity detection: (a) Result with Algorithm 1 ($\mathscr{R}$=120); (b) Point set where $F(p) > T$; (c) Final boundary with Algorithm 2; (d) PBM value with the change of k value; (e) Determination of compactness threshold.

Definition 8. Hole:

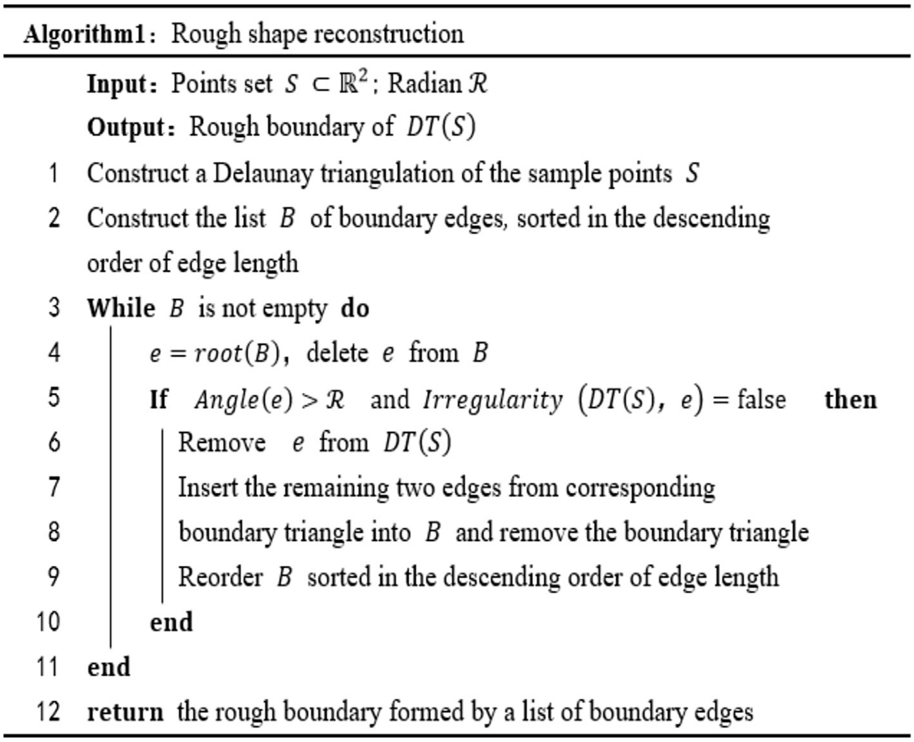
Armed with Jordan curve theorem (Hales, 2007), let $\mathscr{H}$ be a simple closed curve bounding all sample points in the plane $\mathbb{R}^2$(see Fig. 5(b)). Then its complement, $\mathbb{R}^2\backslash\mathscr{H}$, consists of exactly two connected components. One is bounded (the interior) and the other is unbounded (the exterior). Let $\mathscr{B}$ be the "interior" region (see Fig. 5(b)) bounded by the curve $\mathscr{H}$ and $\mathscr{M}$ be the set of all connected regions of $\mathscr{M}\subset\mathscr{B}$ (see Fig. 5(b)). Each connected polygon is referred as a hole and the closed polygonal curves are hole boundaries. In $DT(S)$, the edges of $\mathscr{M}$ in each hole are called hole edges (see Fig. 5(c)).

Observation 1: Hole can be regarded as the extension of cavity.

A simple closed curve C_1 bounds all sample points P, the line segments are said to be cavity edges (see Fig. 6 (a)), if they lie to the exterior of C_1 according to the Jordan curve theorem. Let $conv(P)$ be the convex hull of P and C_2 be a simple closed curve bounding the convex hull. In this case, the previous cavity edges can be said to be hole edges (see Fig. 6 (b)) as they lie to the interior of C_2.

(see Fig. 7). If $Angle(e) > \mathscr{R}$, the boundary edge is removed, otherwise, it is maintained. The boundary edges of unsatisfactory triangles are evidently more likely to be deleted both in high and low density regions.

The rough shape reconstruction (Algorithm 1) proceeds by the iterative removal of boundary edges satisfying$Angle(e) > \mathscr{R}$and *Irregularity* $(DT(S), e)$=false (see Definition 3). In each iteration, one boundary edge e may be removed from $DT(S)$, we measure the list of boundary edges B in descending order of edge length to avoid the multiplicity of reconstructed results. Various criteria (e.g., length and circumradius) have been used for ordering the boundary triangles. We have conducted some validation experiments, which showed that no criteria can ensure the best shape and have a major effect on the reconstructed results. The reason for choosing length is that, when compared with other criteria (e.g., circumradius), length can get a quick access from the list B. When a boundary edge e is removed, two new boundary edges will be inserted into B, reordering the new B. In our approach, we use a heap based priority queue (denoted by $root(B)$ in Algorithm 1) to make sure that the longest boundary edge is selected first at each iteration. The algorithm terminates when B is empty.



Algorithm1: Rough shape reconstruction

```
Input: Points set $S \subset \mathbb{R}^2$; Radian $\mathcal{R}$
Output: Rough boundary of $DT(S)$
Construct a Delaunay triangulation of the sample points $S$
Construct the list $B$ of boundary edges, sorted in the descending
order of edge length
While $B$ is not empty do
    $e = root(B)$, delete $e$ from $B$
    If $Angle(e) > \mathcal{R}$ and $Irregularity\ (DT(S), e) =$ false then
        Remove $e$ from $DT(S)$
        Insert the remaining two edges from corresponding
        boundary triangle into $B$ and remove the boundary triangle
        Reorder $B$ sorted in the descending order of edge length
    end
end
return the rough boundary formed by a list of boundary edges
```

4. Proposed algorithm

Our algorithm can be divided into two main parts, which are rough shape reconstruction and refined shape reconstruction.

4.1. Rough shape reconstruction

Various parameters have been used for shape reconstruction, including length, area, and perimeter. However, these measures are global constants, which directly lead to unsatisfactory results when point set $S\subset\mathbb{R}^2$ is distributed in different densities. According to Definition 6, the unsatisfactory triangles of $DT(S)$ push two boundary vertices further away from each other in high and low density regions. Hence, the boundary edges corresponding to unsatisfactory triangles should be removed. Based on the above, a relative parameter–angle$\mathscr{R}$is adopted in the rough shape reconstruction to evaluate the criterion for boundary edge. Let a boundary edge be e and the angle opposite to e be $Angle(e)$

4.2. Formulation of cavity and hole

In this section, we provide a mathematical definition for **Observation 1** and propose a statistical approach to detect them.

Algorithm 1 is based on the removal of boundary edges where satisfactory triangles should be retained. However, it cannot satisfy the following structures:

- **Cavity with satisfactory triangles**

Broadly, we can obtain a desirable result from cavities with unsatisfactory triangles by setting a suitable threshold in Algorithm 1. However, in case of satisfactory triangulates (see Fig. 8(b)) during the iteration, Algorithm 1 becomes invalid and precludes further cavity detection. Moreover, if we relax the threshold to detect the structure, the final boundary becomes undesirable, as indicated by many shape corners (see Fig. 8(c)).

- **Hole**

According to the definition of boundary edge, the edges inside the hole do not belong to the boundary edge. Hence, the holes (see Fig. 9) cannot be detected in Algorithm 1.

As shown in Fig. 10, we adopt the Voronoi diagram to express the density variation of the two structures. The density of the point set is defined as the reciprocal of Voronoi area occupied by each point. Cavity and hole structures are then conceptualized as a structure with the low-density region surrounded by high-density region.

It can be observed from the Delaunay Diagram (the dual graph of Voronoi) that the length variations of the edges incident to points in the border of cavity and hole tend to have relatively larger, as both short and long edges exist (see Figs. 8 and 9). To exploit this characteristic, we adopt a statistical variable-compactness of point, which is expressed as follows:

Definition 9. Compactness of point:

In $DT(S)$, for a point $p \in S$, the neighborhood $N(p)$ is the set of edges incident to point p. The compactness of p is defined as

$$F(p) = Local\_SD(p)/Local\_Mean\_Length(p) \tag{1}$$

where $Local\_Mean\_Length(p)$
is the mean length of edges in $N(p)$, and $Local_SD(p)$is the standard deviation of the lengths of edges in $N(p)$. Here,

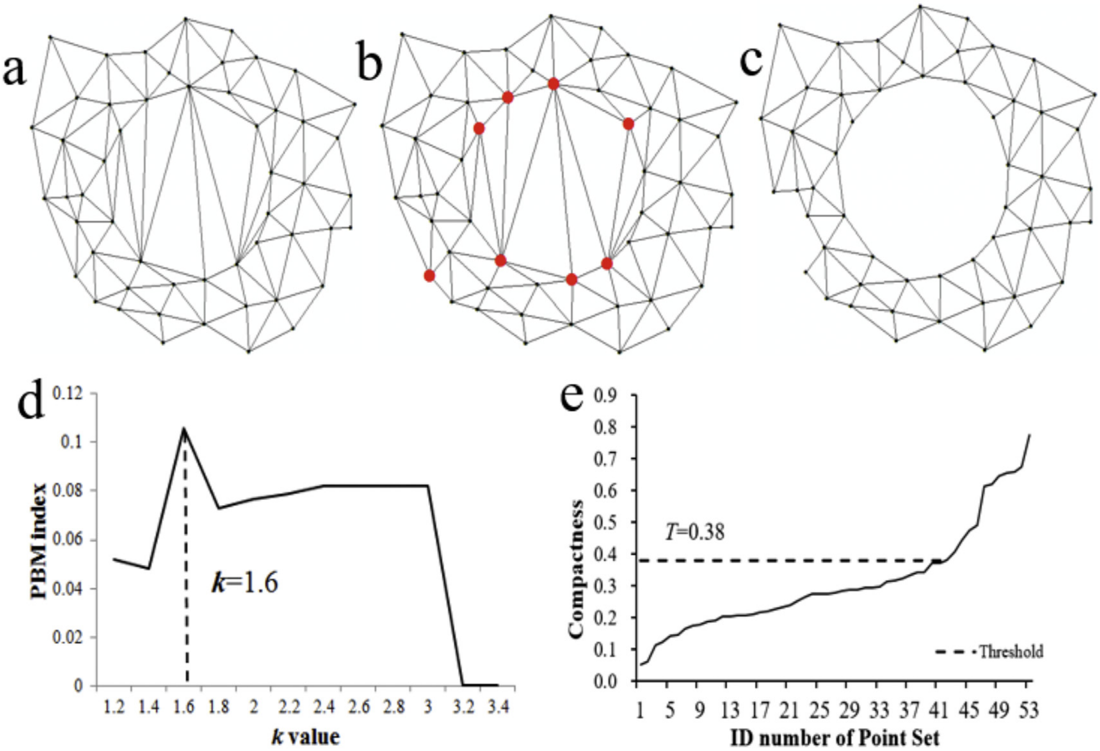
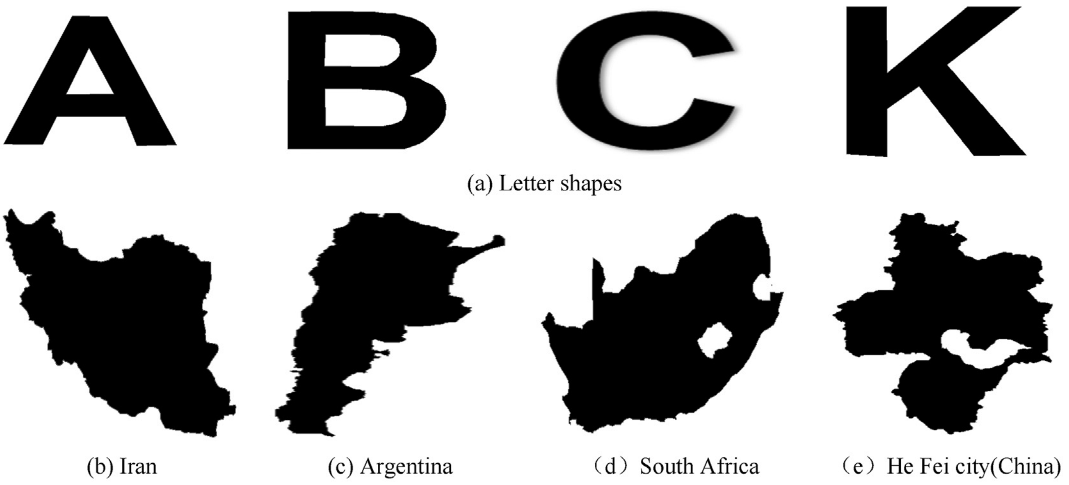


Fig. 13. Different stages of the proposed algorithm for cavity detection: (a) Result with Algorithm 1 ($\mathscr{R}$=120); (b) Point set where $F(p) > T$; (c) Final boundary with Algorithm 2; (d) PBM value with the change of k value; (e) Determination of compactness threshold.


(a) Letter shapes

(b) Iran (c) Argentina （d）South Africa （e）He Fei city(China)

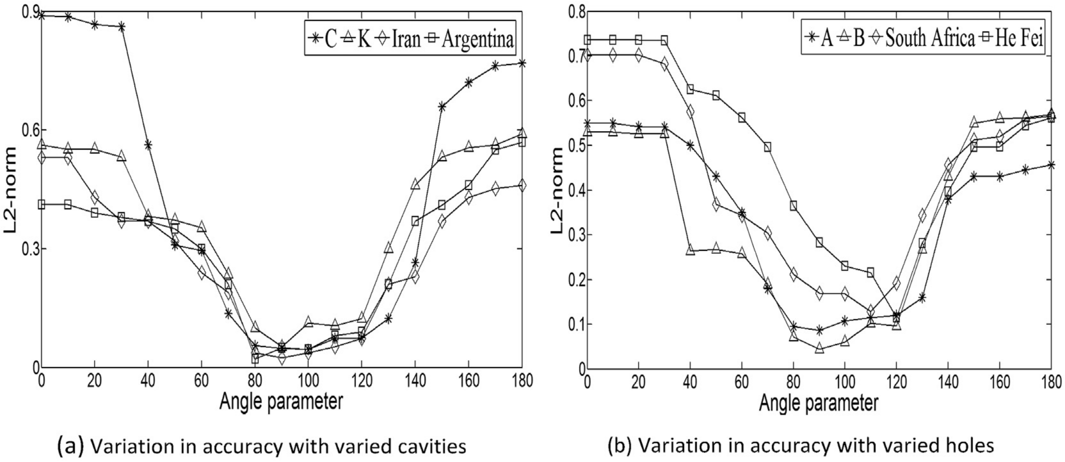
Fig. 14. Experimental data.

$$Local\_Mean\_Length(p) = \frac{1}{d(p)}\sum_{i=1}^{d(p)}|e_i| \tag{2}$$

where $d(p)$
denotes the number of Delaunay edges incident to p, and $|e_i|$ is the length of edges in $N(p)$.

$$Local\_SD(p) = \sqrt{\frac{1}{d(P)}\sum_{i=1}^{d(P)}(Local\_Mean\_Length(p) - |e_i|)^2} \tag{3}$$

Instead of standard deviation, $F(p)$ is a relative parameter, which can compare the discrete degree of two data sets that have different numbers or have a great difference in mean value. Evidently, $F(p)$ value of each point inside boundary will be small given that the lengths of their



(a) Variation in accuracy with varied cavities

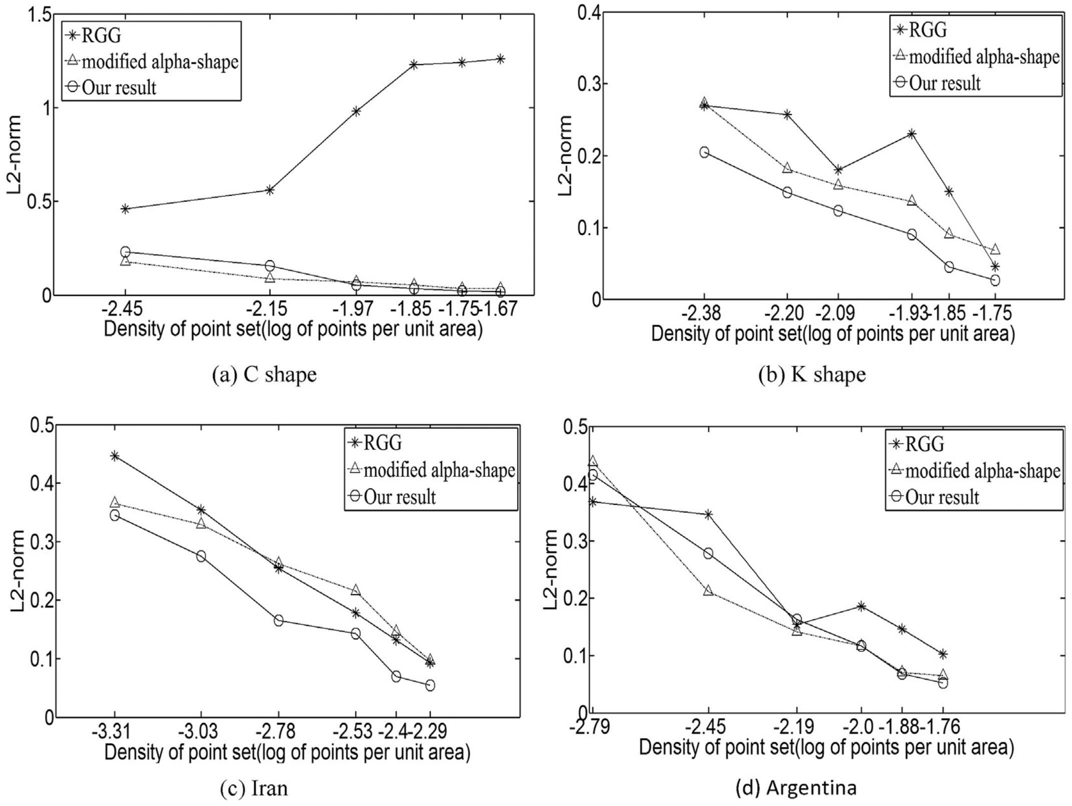
(b) Variation in accuracy with varied holes

Fig. 15. Effects of parameter.$\mathscr{R}$.

Table 1
Experimental results of statistical test of the Kruskal-Wallis test.

	Our result vs modified **α**-shape	our result vs *RGG*		Our result vs modified **α**-shape	our result vs *RGG*
Experiment 2					
C shape			K shape		
Density =-2.45	P > 0.05	P < 0.05	Density=−2.38	P < 0.05	P < 0.05
Density=−2.15	P < 0.05	P < 0.05	Density=−2.20	P < 0.05	P < 0.05
Density =-1.97	P > 0.05	P < 0.05	Density=−2.09	P < 0.05	P < 0.05
Density =-1.85	P > 0.05	P < 0.05	Density=−1.93	P < 0.05	P < 0.05
Density =-1.75	P > 0.05	P < 0.05	Density=−1.85	P < 0.05	P < 0.05
Density =-1.67	P > 0.05	P < 0.05	Density=−1.75	P<0.05	P>0.05
Iran			Argentina		
Density =-3.31	P > 0.05	P < 0.05	Density=−2.79	P > 0.05	P < 0.05
Density =-3.03	P < 0.05	P < 0.05	Density=−2.45	P < 0.05	P < 0.05
Density =-2.78	P < 0.05	P < 0.05	Density=−2.19	P > 0.05	P > 0.05
Density =-2.53	P<0.05	P>0.05	Density =-2.0	P > 0.05	P < 0.05
Density =-2.4	P < 0.05	P < 0.05	Density=−1.88	P > 0.05	P < 0.05
Density =-2.29	P<0.05	P>0.05	Density=−1.76	P > 0.05	P < 0.05
A shape			B shape		
Density =-2.45	P < 0.05	P < 0.05	Density=−2.39	P<0.05	P>0.05
Density =-2.14	P > 0.05	P < 0.05	Density=−2.22	P<0.05	P>0.05
Density =-1.96	P > 0.05	P < 0.05	Density=−2.10	P < 0.05	P < 0.05
Density =-1.84	P > 0.05	P < 0.05	Density=−1.95	P < 0.05	P < 0.05
Density =-1.74	P > 0.05	P < 0.05	Density=−1.86	P > 0.05	P < 0.05
Density =-1.65	P > 0.05	P < 0.05	Density=−1.77	P > 0.05	P > 0.05
South Africa			He Fei		
Density =-2.28	P < 0.05	P < 0.05	Density=−1.96	P < 0.05	P < 0.05
Density =-2.01	P<0.05	P>0.05	Density=−1.90	P < 0.05	P < 0.05
Density =-1.92	P < 0.05	P < 0.05	Density=−1.82	P < 0.05	P < 0.05
Density =-1.81	P > 0.05	P > 0.05	Density=−1.75	P < 0.05	P < 0.05
Density =-1.72	P<0.05	P>0.05	Density=−1.64	P < 0.05	P < 0.05
Density =-1.63	P < 0.05	P < 0.05	Density=−1.58	P < 0.05	P < 0.05
Experiment 3					
K shape			B shape		
r=0	P < 0.05	P < 0.05	r=0	P < 0.05	P < 0.05
r=0.5 × 10^{-3}	P < 0.05	P < 0.05	r=0.5 × 10^{-3}	P < 0.05	P < 0.05
r=1.0 × 10^{-3}	P < 0.05	P < 0.05	r=1.0 × 10^{-3}	P < 0.05	P < 0.05
r=1.5 × 10^{-3}	P < 0.05	P < 0.05	r=1.5 × 10^{-3}	P < 0.05	P < 0.05
r=2.0 × 10^{-3}	P > 0.05	P < 0.05	r=2.0 × 10^{-3}	P > 0.05	P > 0.05
r=2.5 × 10^{-3}	P > 0.05	P > 0.05	r=2.5 × 10^{-3}	P > 0.05	P > 0.05
r=3.0 × 10^{-3}	P > 0.05	P > 0.05			

*The result is convincing when P < 0.05.



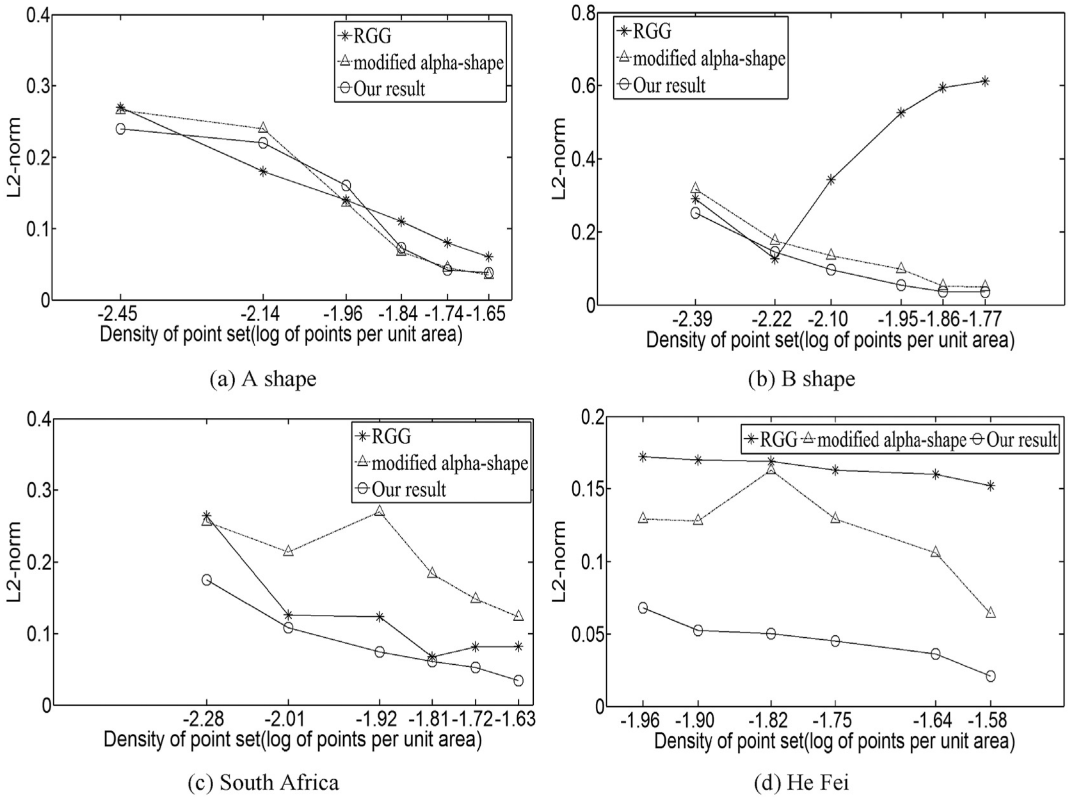
(a) C shape

(b) K shape

(c) Iran

(d) Argentina

Fig. 16. Variation in accuracy of varied cavities with changing density of point sets.



(a) A shape

(b) B shape

(c) South Africa

(d) He Fei

Fig. 17. Variation in accuracy of varied holes with changing density of point sets.

incident edges change little. In contrast, $F(p)$ will be large for the points on the border of cavity and holes with a larger length variation than those inside.

As shown in Fig. 11, the points on the border of the cavity and hole have greater $F(p)$, as denoted by $Sets = \{F(p)|F(p) > T\}$. $F(p)$ value can effectively capture the gradient changes from the boundary to the inside of $DT(S)$. We pose this variation as the problem that one requires an effective means of determining the point by locating a sharp gradient change in the $F(p)$ values. A clustering method is employed to determine T in this paper. The first step sorts the $F(p)$ values of all points in ascending order, and the values are then detected in order. When the first point where $F(p)$ value cannot satisfy the rule of k standard deviations occurs, T is assigned the $F(p+1)$

value. The rule is defined as follows:

$$|x_i - \overline{x}| \le k\sigma \quad (4)$$

where $\overline{x}$ is the mean value of the $F(p)$ values and σ is the standard deviation of the $F(p)$ values. Evidently, T becomes large with the increased k value.

The rule splits all the values into two clusters. Hence, values in the same cluster should be ensured as similar to each other and dissimilar to those in different clusters for obtaining a suitable k. PBM index (Pakhira

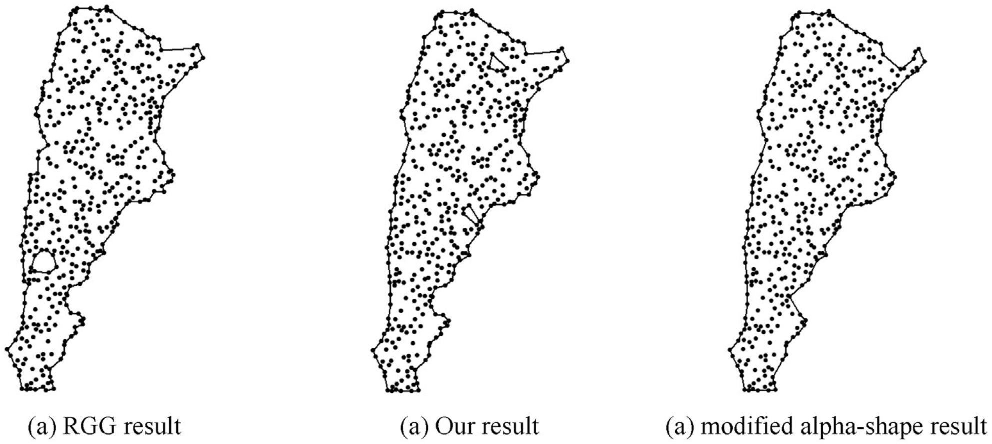


Fig. 18. Shapes generated for Argentina shape point set with density −2.45 (log of points per unit area).

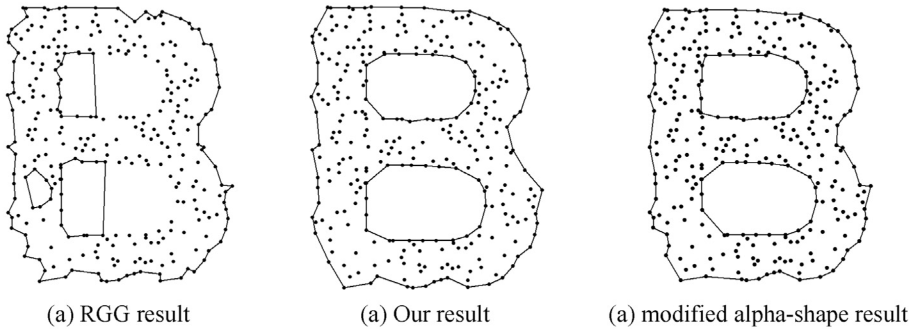


Fig. 19. Shapes generated for B shape point set with density −2.22 (log of points per unit area).

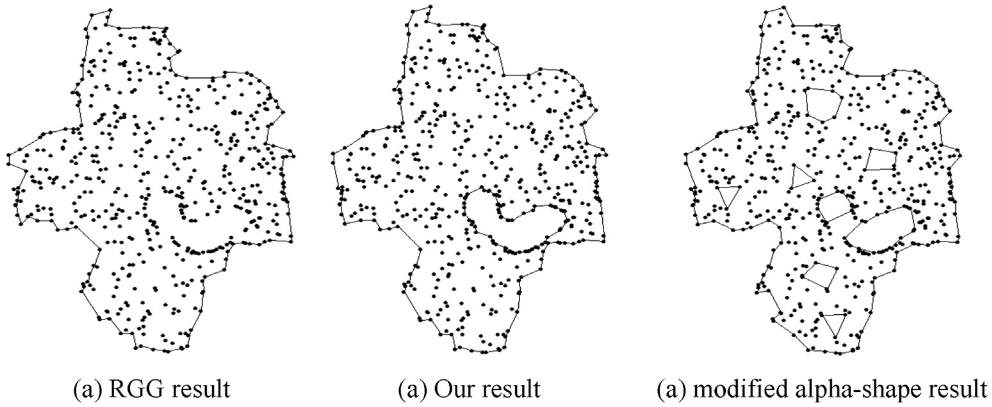


Fig. 20. Shapes generated for He Fei point set with density −1.96 (log of points per unit area).

et al., 2004) provides a measure of how "neatly split" the two clusters are. PBM index is expressed as:

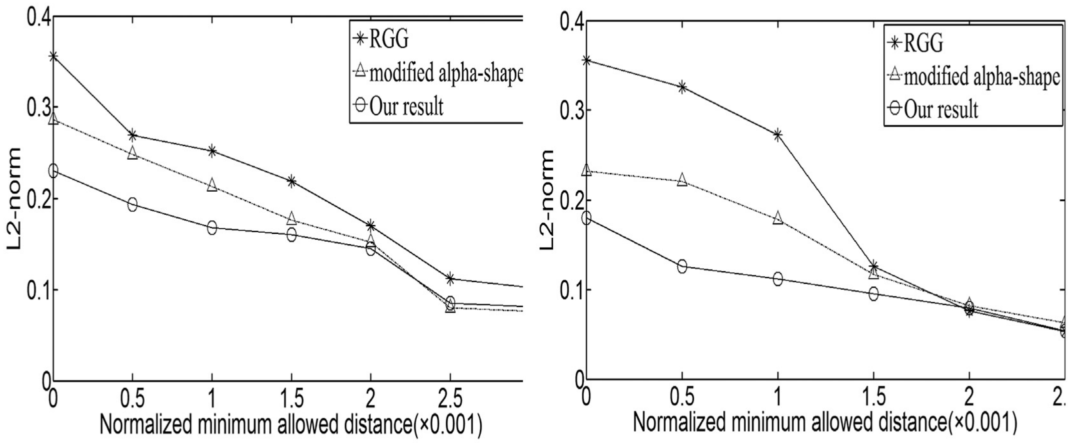
$$\mathrm{PBM} = \left(\frac{1}{N_c}\frac{E_1}{E_{N_c}}D_{N_c}\right)^2 \tag{5}$$

$$E_{N_c} = \sum\nolimits_{i=1}^{N_c} E_i \ , \ E_i = \sum\nolimits_{j=1}^{N_i} \left\|x_j - v_i\right\|, \ D_{N_c} = \max\nolimits_{i,j=1}^{N_c}\left(\left\|v_i - v_j\right\|\right) \tag{6}$$

where N_c is the number of clusters, N_i is the total number of points in C_i cluster, v_i is the center of the C_i cluster, and D_{N_c}

is the measurement of the maximum separation between a pair of clusters. When this index is large, the rule of *k* standard deviations generally gains better result.

When both short and long edges exist in the *Sets*, the next step involves filtering these edges to obtain the final boundary. From clustering analysis perspective, the compactness of the points in the final boundary does not vary too much, and evidently, the long edges incident to points on the border of cavity and hole should be removed. Furthermore, Algorithm1 can further act on the retained edges. In this paper, we use the AUTOCLUST algorithm (Estivill-Castro and Lee, 2002a) to delete the long edges of the Delaunay triangulation. The algorithm combined hierarchical exploration with graph and density information to detect the spatial cluster of different hierarchy (refer to the inside AUTOCLUST for more details). The specific expressions are as follows:



(a) K shape (b) A shape

Fig. 21. Effects of inhomogeneity.

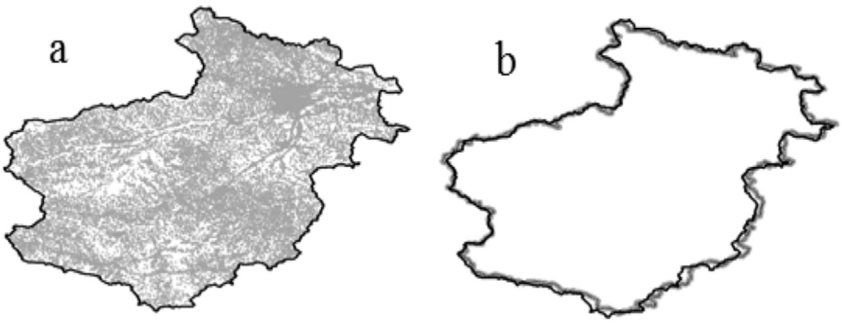


Fig. 22. Illustration of: (a) POI data set of Luoyang (gray point set) and urban contour (black line) generated by our algorithm; (b) Boundary generated by our method (black line) versus comparative boundary (gray line).

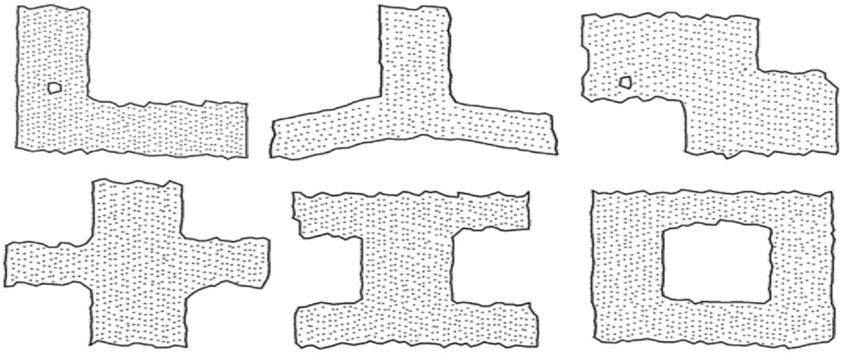
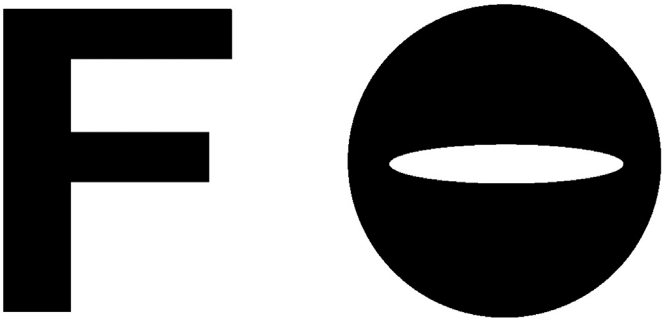


Fig. 23. Extracting the results of different building boundaries using our algorithm.



(a) Cavity shape (b) Hole shape

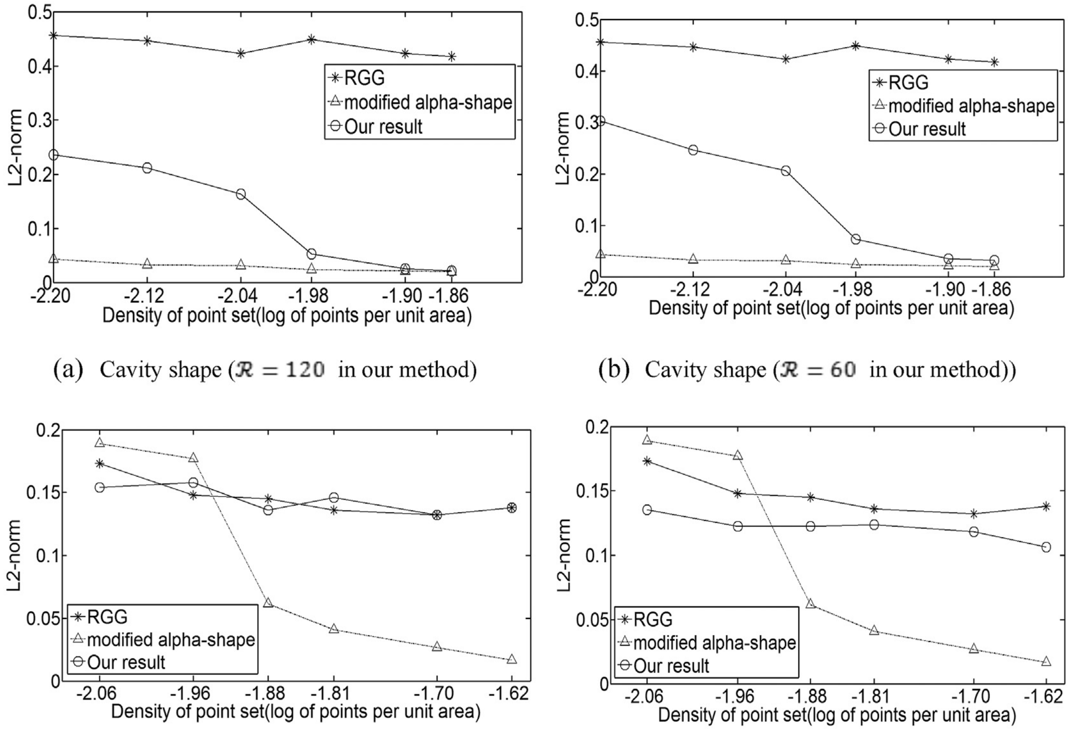
Fig. 24. Experimental data.

$$T(p) = \{e_i | |e_i| > Local\_Mean\_Length(p) + Global\_SD\} \tag{7}$$

$$Global\_SD = \frac{1}{N}\sum_{i=1}^{N} Local\_SD(p) \tag{8}$$

where *Global_SD* means the average of *Local_SD*(p) values in $DT(S)$. If an edge incident to point p in *Sets* belongs to $T(p)$, the edge is removed from $DT(S)$.

Refined shape reconstruction (Algorithm 2) aims to detect pure cavities and holes. First, we construct the list of points A, where $F(p)$ values are more than T from the Delaunay triangulation of rough boundary. The list A is then divided into boundary vertices A_B and interior vertices A_I . In Algorithm 2, pure cavity (line. **5–16**) detection has priority over hole detection (line. **17–28**) because the proposed method is based on



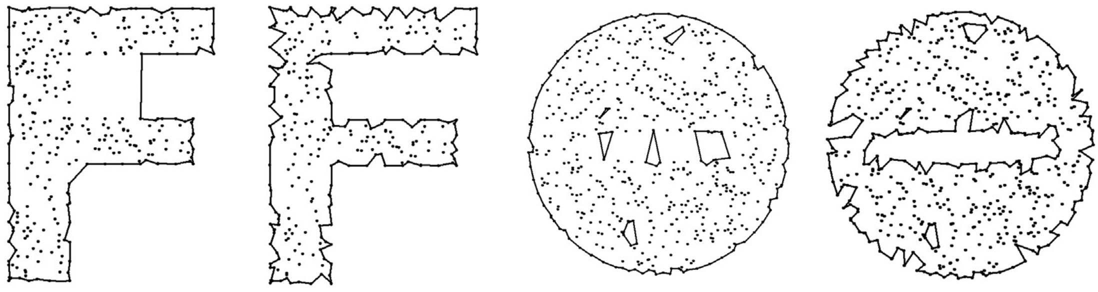
(a) Cavity shape ($\mathcal{R} = 120$ in our method)

(b) Cavity shape ($\mathcal{R} = 60$ in our method))

(c) Hole shape ($\mathcal{R} = 120$ in our method)

(d) Hole shape ($\mathcal{R} = 60$ in our method)

Fig. 25. Variation in accuracy of cavity and hole with changing density of point sets.



(a)Our resul($\mathcal{R} = 120$) in density -2.12

(b) Ourresult($\mathcal{R} = 60$) in density -2.12

(c) Our result ($\mathcal{R} = 120$) in density -1.88

(d) Our result ($\mathcal{R} = 60$) in density -1.88

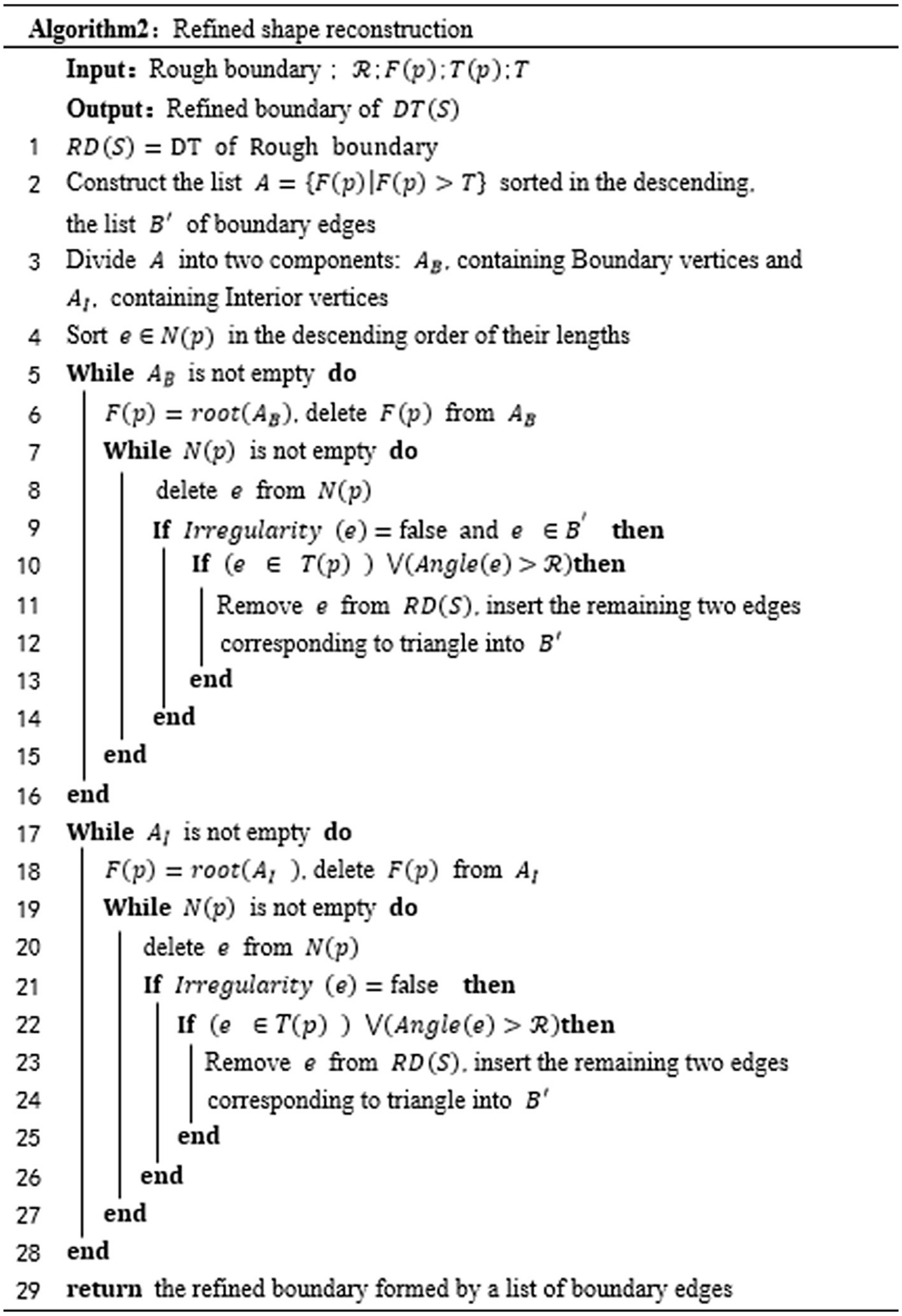
Fig. 26. Examples of a limitation of our algorithm.

filtering virtual boundary edges. Afterwards, we sort the $F(p)$ of A and e of $N(p)$ in descending order to yield the unique boundary. For cavities, the virtual boundary edge is first determined if it is a removal edge on the *long edges* and *Irregularity* constraints. The *long edges* constraint is simple that e should belong to $T(p)$. When one edge can be deleted, the remaining two edges corresponding to the triangle are added into list $B^{'}$. Compared with cavities, the holes located inside Delaunay triangulation have no boundary edges. Hence, the edges in holes are removed simply in the descending order of A and $N(p)$ on the- *long edges* and *Irregularity* constraints. Furthermore, several edges in list $B^{'}$may not belong to $T(p)$after $B^{'}$ is updated. To solve this problem, we employ Algorithm 1 to detect these edges.

Combining with Algorithms 1 and 2, Figs. 12 and 13, respectively, shows the steps of cavity and hole detection. First, we employ Algorithm 1 to produce rough boundaries (see Figs. 12(a) and 13(a)), and then use Algorithm 2 to locate and detect the two structures based on the rough result. The PBM index is adopted to determine where the cavity and hole are (i.e. $F(p) > T$), as shown in Fig. 12(d)–(e) and 13(d)–(e). The final boundaries (see Fig. 12(c) and 13(c))) are generated based on the constraints of Algorithms 1 and 2.

4.3. Time complexity

The proposed algorithm has a computational complexity is $O(N^2)$



```
Algorithm2: Refined shape reconstruction
    Input: Rough boundary ; $\mathcal{R}$; $F(p)$; $T(p)$; $T$
    Output: Refined boundary of $DT(S)$
    $RD(S)$ = DT of Rough boundary
    Construct the list $A = \{F(p) | F(p) > T\}$ sorted in the descending,
    the list $B'$ of boundary edges
    Divide $A$ into two components: $A_B$, containing Boundary vertices and
    $A_I$, containing Interior vertices
    Sort $e \in N(p)$ in the descending order of their lengths
    While $A_B$ is not empty do
        $F(p) = root(A_B)$, delete $F(p)$ from $A_B$
        While $N(p)$ is not empty do
            delete $e$ from $N(p)$
            If $Irregularity$ $(e)$ = false and $e \in B^{'}$ then
                If $(e \in T(p)) \vee (Angle(e) > \mathcal{R})$then
                    Remove $e$ from $RD(S)$, insert the remaining two edges
                    corresponding to triangle into $B'$
                end
            end
        end
    end
    While $A_I$ is not empty do
        $F(p) = root(A_I)$, delete $F(p)$ from $A_I$
        While $N(p)$ is not empty do
            delete $e$ from $N(p)$
            If $Irregularity$ $(e)$ = false then
                If $(e \in T(p)) \vee (Angle(e) > \mathcal{R})$then
                    Remove $e$ from $RD(S)$, insert the remaining two edges
                    corresponding to triangle into $B'$
                end
            end
        end
    end
    return the refined boundary formed by a list of boundary edges
```

where N is the cardinality of input points. In Algorithm 1, the number of boundary edges in the list B is linearly proportional to the number of vertices by Euler's formula. Hence the time complexity of sorting step is $O(N\log N)$ (line 2). Next, the main filtration loop (line 3–10) requires $O(N)$ time. In Algorithm 2, finding the cavity/hole requires two steps: computation of $F(p)$ and determination of T, which can be done in $O(N)$ and $O(N\log N)$ time. Three preprocessing steps need to be executed before proceeding the main filtration procedures of cavity and hole: (1) sorting the list of $F(p)$ of border points (line 2) takes $O(N\log N)$ time; (2) splitting the sorted result into two components (line 3) takes $O(N)$ time; (3) sorting the list of edges incident to one border point (line 4) takes $O(N\log N)$ time. Finally, the main filtration loops of cavity (line 5–16) and hole (line 17–28) each require $O(N^2)$ time.

5. Experimentation

The proposed algorithm is implemented in C# using the ArcGIS 10.0 software. We conducted several experiments to test the algorithm's abilities to reconstruct shapes derived from point patterns with well-defined shapes. We also present a comparative study with two other representative approaches considering both cavity and hole detection. One is alpha-shape (Edelsbrunner et al., 1983) and the other is ***RGG*** (Peethambaran and Muthuganapathy, 2015). To ensure fair comparison with our method and ***RGG***, the alpha-shape output was modified by considering ***Irregularity of a simplicial 2-complex***.

Three experiments were designed to evaluate the performance of our method. First, the experiments examined the effects of varying parameters on our shape reconstruction method. Second, the effects of varying point densities on the performance of the three approaches were analyzed. Finally, the effects of increasing inhomogeneity on the performance of the three approaches were studied.

5.1. Experimental setup

We chose four different uppercase letters for the tests: “A,” ”B,” ”C,” and ”K” (see Fig. 14(a)). The letter shapes were generated using a sans serif font (Arial). Another type of shape was borders of three countries and one city (see Fig. 14(b)-(e)).

We used the L^2 error norm (Duckham et al., 2008) for the performance evaluation of the three approaches quantitatively. The L^2 error norm provides a good measure of closeness between true shapes and the reconstructed result of input points. This term is defined as the area of the symmetric difference between an original region O and a reconstructed result S as a proportion of the total area of$S$. That is,

$$L^2 \text{ error norm} = \frac{Area((O-S)\cup(S-O))}{Area(O)} \qquad (9)$$

The L^2 error norm of zero means that the two shapes are equal in area and that their boundaries are similar.

To ensure randomized not by chance, for each shape, 50 replications of internal points were generated, and the average accuracy of these 50 replications for each shape at each parameter was set as the final accuracy. All the external parameters that led to the lowest L^2 error norm were used as the optimal parameterization for the following experimentations.

5.2. Experiment 1: effects of parameter $\mathscr{R}$

In this experiment, each shape was filled with a number of evenly polygon vertices and semi-random distribution of internal points. The semi-random distribution of internal points were generated based on the x-outline algorithm (Zhong and Duckham, 2016). Each point must be greater than a certain threshold distance d from any other points, where $d=\sqrt{rA/\pi}$; A is the total area of true shape; r is the normalized minimum allowed pair-wise distance which is the ratio between πd^2 and A. Semi-random distribution was chosen because the truly random points are highly inhomogeneous, producing clusters and unwanted holes. In Experiment 1, we defined r=2.5×10^{-3} for the shapes.

Fig. 15 (a) and (b) exhibit the variations in the L^2 error norm of varied cavities and holes, respectively, as reconstructed using our method. In general, two line charts of all shapes reveal three major phases. First, the accuracy of all shapes exhibit progressive improvement with decreasing $\mathscr{R}$ parameter from 180 (i.e., the convex hull), characterized by decreasing L^2error norm. Second, below a certain parameter, the algorithm starts to filter the body of shapes, indicated by a rapid decrease in L^2error norm. All the shapes in Fig. 15 (a) and (b) have response curves that lead to that rapid change as the parameter decreases from value around 120. All the curves also represent a relatively low L^2 error norm and stay relatively small changes within a certain range. The L^2 error norm of these shapes can often reach its minimum at around these parameter values. Third, when the parameter is smaller than a certain value, our algorithm begins to erode the interior of these shapes, characterized from a sharp increasing L^2 error norm, however, L^2 error norm shows a steady state with decreasing parameter value when considering the semi-random point set and our Irregularity of a simplicial 2-complex.

The accuracy curves in two graphs demonstrate that the performance of our algorithm for all shapes can provide good reconstructed shapes at parameter 120, since the L^2 error norm of parameter 120 can often exhibit a relatively low curve. Furthermore, the accuracy results suggest that these shapes can reach the minimum L^2 error norm at the parameters of around 80–120.

5.3. Experiment 2: effects of varying point densities

In this experiment, we aim to investigate the potential relationship between the accuracy of three approaches and the point density. Point density is defined that the point set filled shapes using on average one point occupying a region of $n \times n$ pixels (Duckham et al., 2008). Each shape was filled with polygon vertices and a truly-random distribution of internal points. We used the nonparametric Kruskal-Wallis test to examine the null hypothesis that the difference in accuracy between our method and RGG or modified alpha-shape may have occurred by chance. The null hypothesis is rejected as the coefficient of significant difference is at 0.05 level (i.e., P < 0.05). All the statistical test results are presented in Table 1.

Figs. 16 and 17 exhibit the variations in the L^2 error norm of varied cavities and holes generated by three methods with varying densities. For cavity structure, the accuracy of the modified alpha-shape and our method increase as the point set density increases. Conversely, the ***RGG*** results exhibited large variations with changing densities in several cases (especially in “C” shape). This is mainly due to the problem of *Cavity with satisfactory triangles*. Our algorithm performs relatively better than the other two approaches as the point sets density increases for K and Iran shapes. For Argentina shape, the boundary of Argentina is elongated with different levels of sinuosity, which makes the chance of reconstruted shape errors more than the shapes with relatively large area to perimeter ratio (e.g., Iran). Our method did not significantly outperform the other two approaches on this kind of shape (see Fig. 18).

For hole detection, the accuracy of the modified alpha-shape and our method increase as the point set density increases. For B shape, RGG fails to detect the holes (see Fig. 19(a)) at the point density of −2.22, leading to an increase in L^2 error norm. Our method can generate a more accurate B shape than modified alpha-shape (see Fig. 19(b) and (c)). It can be observed from the letter shapes that our algorithm exceeds the accuracy of RGG at higher point density and is more accurate than modified alpha-shape at lower density.

Unlike letter shapes, South Africa and He Fei shapes possess different levels of cavities and holes, our method significantly outperforms the other two approaches. In addition to that, RGG is lack of the robustness to the varied holes, especially for the thin holes (see Fig. 20(a)). α-shape relies on a global parameter, which may generate undesirable holes with truly-random distribution of internal points (see Fig. 20(c)).

5.4. Experiment 3: effects of inhomogeneity

Compared with country or city border, the letter shapes can capture a more precise summary of the reconstructed shapes, we focused on the letter shapes in this experiment. The letters K and A is a representative shape for cavities detection (Peethambaran and Muthuganapathy, 2015) and hole detection (Edelsbrunner et al., 1983). We set the parameter *r* from o (truly random point distributions) to 3 × 10^{-3} (semi-random point distributions) in cavity detection and set parameter *r* from o to 2.5 × 10^{-3} in hole detection. Each shape was only filled with varied distribution of internal points (without polygon vertices). The density of points for K and A was fixed approximately at −1.75 and −1.84 respectively where our method did not significantly outperform the other two alternatives in Experiment 2.

Fig. 21(a) and (b) display the variations in the L^2 error norm of K and A shapes generated by the three methods with decreasing homogeneity distributions. As inhomogeneity in point distributions decreases, the accuracy of all shapes by three approaches increases. To be more precise, a truly random distribution of points may lack several features of the original shape, making the deviation between the final boundary results and the original shape become great. The semi-random distribution is structurally closer to the original shape thereby making the reconstruction more accurate. The magnitude of L^2 error norm from truly-random to semi-random distribution indicated that our method was more tolerant than the other two approaches, characterized by minimal magnitude of L^2 error norm (approximately 64% for K shape and 70% for A shape). The modified alpha-shape was in a moderate effect with magnitude of 73% and 74%. RGG was particularly sensitive to the increasing homogeneity with magnitude of 78% and 85%. This is mainly due to the absence of boundary points capable of defining shapes. For K and A shape, the nonparametric Kruskal-Wallis test shows that the proposed algorithm significantly outperformed modified alpha-shape and RGG with parameter *r* from 0 to 1.5 × 10^{-3}. It becomes clear that the proposed algorithm outperformed the other two algorithms as the point set is close to truly-random distribution.

6. Applications

In this section, two applications validate the practicability of our algorithm, as well as illustrate its wide range of potential applications. In the first application, we employ the algorithm to obtain urban contour using Point of Interest (POI) data, an important part of urban morphology. Fig. 22(a) displays the actual POI data (geographic entity data) of Luoyang city, China, and shows 77 497 records in the POI dataset. As urban contour is a simple polygon that contains all entities, we simply adopted Algorithm 1 and cavity detection to extract it. The administrative boundary of Luoyang is used for comparison to validate the boundary produced by our approach. It can be observed from Fig. 22(b) that our method is a good choice for this task.

The second application comes from building boundary extraction based on LIDAR data. We conducted experiments with various geometrical shapes of building boundaries from Kuala Lumpur City Center District in Malaysia (provided by Shen et al., 2008) to validate our algorithm. The point density was estimated at about 6 points per square meter, the vertical accuracy of LIDAR data was 15 cm, and the horizontal resolution was 30 cm. The resulting building boundaries are shown in Fig. 23. We can see that the proposed method can describe the concaves or holes of different building boundaries effectively.

7. Discussion

In our algorithm, the points on the border of the cavity and hole have greater $F(p)$ value. The circumstances where $F(p)$ of points inside the Delaunay triangulation are greater than or similar to $F(p)$ of points on the border of the cavity/hole reveals a limitation of our algorithm. Fig. 24 provides two defined shapes by example to illustrate the problem of our method. Each shape was filled with even polygon vertices and a truly-random distribution of internal points.

For cavity detection, the proposed algorithm was outperformed by modified alpha-shape with density of points from −2.20 to −2.04 (see Fig. 25(a)). Within this range, $F(p)$ of points inside the Delaunay triangulation are similar to $F(p)$ of points on the border of the cavity. Hence, a very small part border points are selected out, precluding further cavity detection (see Fig. 26(a)). To obtain the pure cavity, the parameter $\mathscr{R}$ can be relaxed. As shown in Fig. 26(b), the performance of our method is still worse than modified alpha-shape. When the point density increases at −1.98, $F(p)$ of points inside the Delaunay triangulation are significantly lower than $F(p)$ of points on the border. The cavity shape can get a desirable result (see Fig. 25(a)). Fig. 25(c)-(d) and 26(c)-(d) show the same experiment as cavity experiments. Two results for hole detection represent a relatively high L^2 error norm with varying point densities (see Fig. 25(c)-(d)). The experiment in this case yielded several undesirable holes (e.g., Fig. 26(c)) as $F(p)$ of points inside the Delaunay triangulation are greater than $F(p)$ of points on the border. This limitation of our algorithm may due to two issues. One is the inhomogeneous distribution (especially in truly random distribution) of internal points, which can cause extremely uneven edge lengths incident to points inside the reconstructed shape. The other is the shapes (e.g., thin hole in Fig. 24 and Argentina shape in Fig. 18). This disadvantage may produce the

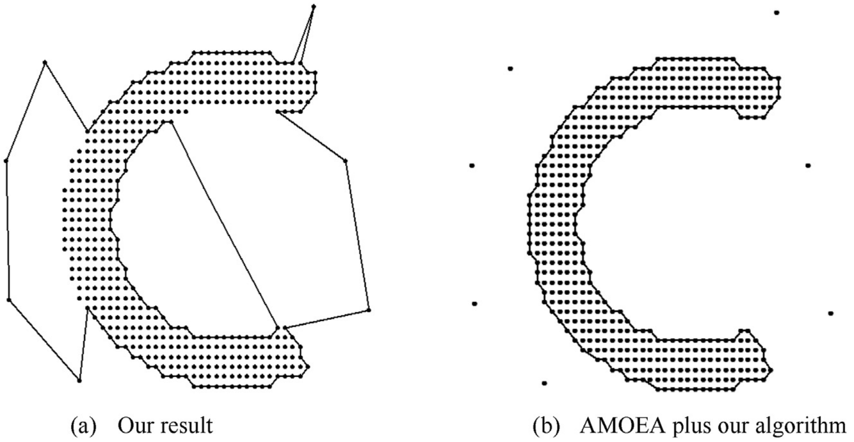

(a) Our result
(b) AMOEA plus our algorithm

Fig. 27. Example our algorithm and AMOEA plus our algorithm for a cavity shape point set containing noise data.

negative influence on the detection of hole/cavity boundaries (See Figs. 25 and 26) and generate undesirable holes (See Figs. 23 and 26(c)). Moreover, each method generating holes is different from each other, which include edge-based (e.g., our algorithm) and triangle-based (e.g., alpha-shape) filtration and the experiment results illustrate that our method did not outperform the modified alpha-shape in certain circumstances (e.g., Fig. 25(c)), hence it becomes a challenging task to handle the undesirable holes for reaching the desirable shape.

The proposed algorithm generates boundaries containing all the points even noise. Hence, the algorithm does not deal with noise well. However, noise can be pre-processed using a spatial clustering algorithm (e.g., AMOEA method by Estivill-Castro and Lee (2002b)). Fig. 27 shows a cavity shape point set with noise data. Our algorithm can separate sample points from noise data using an appropriate clustering in a preprocessing step.

8. Conclusions and future work

To validate our algorithm objectively, we conducted a series of experiments and compared with other Delaunay-based shape reconstruction approaches: alpha-shape and ***RGG***, some significant results are summarized as follows:

- The accuracy of shape reconstruction by the proposed algorithm increases as the point set density increases. For cavity structure, in many cases our method significantly outperforms the comparative methods at less dense regions. For hole structure, our method has the ability to deal with different hole structure with varying densities, and there is no restriction on the number of holes as well.
- The accuracy of shape reconstruction by our method increases as the inhomogeneity in the point distribution decreases. For both cavity and hole structure, the proposed algorithm outperformed the other two alternatives as the point set is close to truly-random distribution (see **Experiment 3**).
- The total time complexity of our algorithm is $O(N^2)$ time, however, RGG and alpha-shape require $O(N\log N)$ time.

In cases where the input points tend to be inhomogeneous (especially in truly random distribution), our algorithm revealed some limitations, so it needs to be solved in future work. Apart from this context, the extensions of the algorithm to higher dimensions are possible based on the proposed data structure, which requires both redesign and modification on the corresponding algorithms.

Acknowledgements

This study was supported by the National Natural Science Foundation of China (Grant Nos. 41671392 and 41631175).

References

Arampatzis, A., Van Kreveld, M., Reinbacher, I., Jones, C.B., Vaid, S., Clough, P., Joho, H., Sanderson, M., 2006. Web-based delineation of imprecise regions. Comput. Environ. Urban Syst. 30 (4), 436–459.

Boissonnat, J.D., 1984. Geometric structures for three-dimensional shape representation. ACM Trans. Graph. (TOG) 3 (4), 266–286.

Boissonnat, J.D., Oudot, S., 2005. Provably good sampling and meshing of surfaces. Graph. Models 67 (5), 405–451.

Bordogna, G., Ghisalberti, G., Psaila, G., 2012. Geographic information retrieval: modeling uncertainty of user's context. Fuzzy Sets Syst. 196, 105–124.

Chaudhuri, A.R., Chaudhuri, B.B., Parui, S.K., 1997. A novel approach to computation of the shape of a dot pattern and extraction of its perceptual border. Comput. Vis. Image Underst. 68 (3), 257–275.

Chevallier, N., Maillot, Y., 2011. Boundary of a non-uniform point cloud for reconstruction. In: Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry. ACM, pp. 510–518.

Duckham, M., Kulik, L., Worboys, M., Galton, A., 2008. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. Pattern Recognit. 41 (10), 3224–3236.

Edelsbrunner, H., Kirkpatrick, D.G., Seidel, R., 1983. On the shape of a set of points in the plane: information Theory. IEEE Trans. 29 (4), 551–559.

Estivill-Castro, V., Lee, I., 2002a. Argument free clustering for large spatial point-data sets via boundary extraction from Delaunay Diagram. Comput. Environ. Urban Syst. 26 (4), 315–334.

Estivill-Castro, V., Lee, I., 2002b. Multi-level clustering and its visualization for exploratory spatial analysis. GeoInformatica 6(2) (pp), 123–152.

Fang, L., Gossard, D.C., 1995. Multidimensional curve fitting to unorganized data points by nonlinear minimization. Comput. Aided Des. 27 (1), 48–58.

Galton, A., Duckham, M., 2006. What Is the Region Occupied by a Set of Points? Geographic Information Science. Springer, pp. 81–98.

Gheibi, A., Davoodi, M., Javad, A., Panahi, F., Aghdam, M., Asgaripour, M., Mohades, A., 2011. Polygonal shape reconstruction in the plane. Iet Comput. Vis. 5 (2), 97–106.

Green, P.J., Sibson, R., 1978. Computing Dirichlet tessellations in the plane. Comput. J. 21 (2), 168–173.

Hales, T.C., 2007. The Jordan curve theorem, formally and informally. Am. Math. Mon. 114 (10), 882–894.

Jaromczyk, J.W., Toussaint, G.T., 1992. Relative neighborhood graphs and their relatives. Proc. IEEE 80 (9), 1502–1517.

Kolingerová, I., Žalik, B., 2006. Reconstructing domain boundaries within a given set of points, using Delaunay triangulation. Comput. Geosci. 32 (9), 1310–1319.

Krasnoshchekov, D.N., Polishchuk, V., Vihavainen, A., 2010. Shape approximation using k-order alpha-hulls. In: Proceedings Proceedings of the Twenty-sixth Annual Symposium on Computational Geometry. ACM, pp. 109–110.

Lee, I.K., 2000. Curve reconstruction from unorganized points. Comput. Aided Des. 17 (2), 161–177.

Li, Z., Yan, H., Ai, T., Chen, J., 2004. Automated building generalization based on urban morphology and Gestalt theory. Int. J. Geogr. Inf. Sci. 18 (5), 513–534.

Melkemi, M., Djebali, M., 2000. Computing the shape of a planar points set. Pattern Recognit. 33 (9), 1423–1436.

Mistry, S., Niranjan, U., Gopi, M., 2014. Puzzhull: cavity and protrusion hierarchy to fit conformal polygons. Comput. Aided Des. 46, 233–238.

Moreira, A.J.C., Santos, M.Y., 2007. Concave hull: a k-nearest neighbours approach for the computation of the region occupied by a set of points. GRAPP 2007. In: Proceedings of the Second International Conference on Graphics Theory and Applications, Barcelona, Spain, March 8-11, Volume GM/R.

Ohrhallinger, S., Mudur, S., 2013. An efficient algorithm for determining an aesthetic shape connecting unorganized 2d points. Comput. Graph. Forum 32 (8), 72–88.

Pakhira, M.K., Bandyopadhyay, S., Maulik, U., 2004. Validity index for crisp and fuzzy clusters. Pattern Recognit. 37 (3), 487–501.

Park, J.C., Shin, H., Choi, B.K., 2006. Elliptic Gabriel graph for finding neighbors in a point set and its application to normal vector estimation. Comput. Aided Des. 38 (6), 619–626.

Peethambaran, J., Muthuganapathy, R., 2015. A non-parametric approach to shape reconstruction from planar point sets through Delaunay filtering. Comput. Aided Des. 62, 164–175.

Shen, W., Jing, L.I., Chen, Y.H., Deng, L., Peng, G.X., 2008. Algorithms study of building boundary extraction and normalization based on LIDAR data. J. Remote Sens. 12 (5), 692–698.

Shen, W., Zhang, J., Yuan, F., 2011. A new algorithm of building boundary extraction based on LIDAR data. In: Proceedings of IEEE Conference on Geoinformatics. IEEE Press, pp. 1–4.

Weiler, K., 1985. Edge-based data structures for solid modeling in curved-surface environments. IEEE Comput. Graph. Appl. 5 (1), 21–40.

Yang, M., Lee, E., 1999. Segmentation of measured point data using a parametric quadric surface approximation. Comput. Aided Des. 31 (7), 449–457.

Zhong, X., Duckham, M., 2016. Characterizing the shapes of noisy, non-uniform, and disconnected point clusters in the plane. Comput. Environ. Urban Syst. 57, 48–58.

Zhu, C., Zhang, X., Hu, B., Jaeger, M., 2008. Reconstruction of tree crown shape from scanned data. In: International Conference on Technologies for E-learning and Digital Entertainment. Springer Berlin Heidelberg, pp. 745–756.