



Special Issue on SMI 2017

Hole detection in a planar point set: An empty disk approach

Subhasree Methirumangalath, Shyam Sundar Kannan, Amal Dev Parakkat,
Ramanathan Muthuganapathy*

Advanced Geometric Computing Lab, Department of Engineering Design, Indian Institute of Technology Madras, Chennai 600036, India

ARTICLE INFO**Article history:**

Received 2 April 2017

Revised 20 May 2017

Accepted 25 May 2017

Available online 1 June 2017

Keywords:

Hole detection

Shape reconstruction

Island detection

Delaunay triangulation

Non-parametric

Planar point set

ABSTRACT

Given a planar point set S , outer boundary detection (shape reconstruction) is an extensively studied problem whereas, inner boundary (hole) detection is not a well researched one, probably because detecting the presence of a hole itself is a difficult task. Nevertheless, hole detection has wide applications in areas such as face recognition, model retrieval and pattern recognition. We present a Delaunay triangulation based strategy to detect the presence of holes and an algorithm to reconstruct them. Our algorithm is a unified one which reconstructs holes, both for a boundary sample (points sampled only from the boundary of the object) as well as for a dot pattern (points sampled from the entire object). Our method is a non-parametric one which detects holes irrespective of its shape. Assuming a sampling model, we provide theoretical analysis of the proposed algorithm, which ensures the correctness of the reconstructed holes, for specific structures. We conduct both qualitative and quantitative comparisons with existing methods and demonstrate that our method is better or comparable with them. Experiments with varying point densities and distributions demonstrate that the algorithm is independent of sampling. We also discuss the limitations of the algorithm.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Given a finite set of points, $S \subseteq \mathbb{R}^2$ (Fig. 1(a)), shape reconstruction problem asks for a shape in \mathbb{R}^2 that best approximates S [6]. Most of the existing works in shape reconstruction such as [5,18], focus only on outer boundary detection (Fig. 1(b)). The outer boundary can be considered as a convex/non-convex simple polygon, enclosing all points of S [6].

Visually, Fig. 1(c) captures the features of the shape better than Fig. 1(b), because of the presence of both outer and inner boundaries. An inner boundary (hole) can be considered as an empty convex/non-convex simple polygon which is enclosed within a boundary. Hole detection problem computes a best approximation of inner boundaries of S .

A planar point set can be classified into two types: (i) boundary sample [9] or curve sample [17] and (ii) dot pattern [9] or object sample [17]. If the points are sampled only from the boundary of the object, it is known as a boundary sample (BS), as shown in Fig. 2(a). If the points are sampled from the whole object, it is known as a dot pattern (DP), as shown in Fig. 2(b). Fig. 2(c) and

(d) are the reconstructed shapes for the boundary sample and dot pattern, respectively and we denote the reconstruction of holes for BS and DP as RBS and RDP, respectively.

Given a point set S with its reconstructed outer boundary using Delaunay triangulation based methods such as [5,15,17], hole detection problem computes one of the best approximations of inner boundaries of S . From the set of triangles of the outer boundary reconstructed triangulation (output of [5,15,17]), we propose an algorithm to detect a triangle as the initial hole and expand it to obtain the hole boundary, based on the area and adjacency information of the triangle.

1.1. Motivation

We focus on the hole detection problem due to: (i) the challenges posed by the problem (ii) varied applications for the hole detection (iii) existence of only a few works addressing the problem and (iv) non-existence of a unified method for both reconstruction of boundary sample and reconstruction of dot pattern.

1.1.1. Challenges

The challenges associated with the outer boundary detection (such as ill-posed nature of the problem, dependence of the reconstructed output on density & distribution of the input point set, human cognition and perception [6]) exist for hole detection as well. Apart from them, another major challenge of hole detection is

* Corresponding author.

E-mail addresses: subhasree.rajiv@gmail.com (S. Methirumangalath), shyamsundar33@gmail.com (S.S. Kannan), adp.upasana@gmail.com (A. Dev Parakkat), emry01@gmail.com, mraman@iitm.ac.in (R. Muthuganapathy).

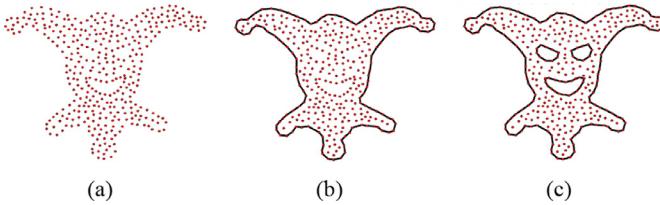


Fig. 1. (a) Input point set (b) Shape with reconstructed outer boundary (c) Shape with reconstructed outer and inner boundaries.

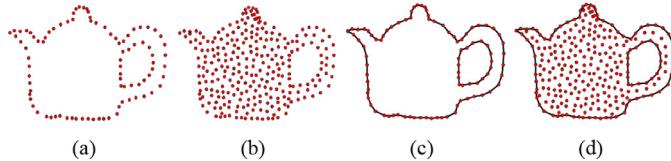


Fig. 2. (a) Boundary sample (b) Dot pattern (c) Reconstructed boundary sample (d) Reconstructed dot pattern.

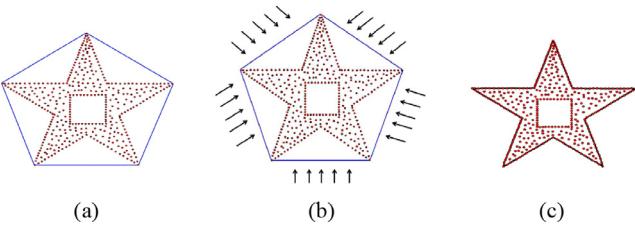


Fig. 3. (a) Convex hull (b) Shrinking of convex hull (c) Reconstructed outer boundary.

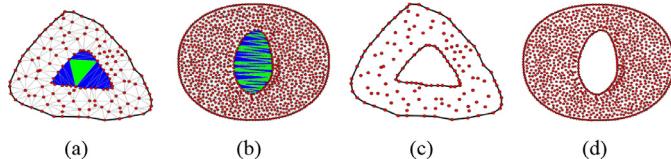


Fig. 4. (a) Hole structure specified in RGG [17]: A fat triangle (green in color) surrounded by a set of thin triangles (blue in color) (b) Hole without the structure specified in RGG (c)–(d) Reconstructed holes of different structures. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to identify the presence of a hole (initial guess of a candidate hole region to start the algorithm) in the given point set. In the case of outer boundary detection, a natural choice for the initial guess of the outer boundary is the convex hull (*CH*) of the point set (blue color boundary shown in Fig. 3(a)), because of the following reasons: (i) *CH* encloses all the points of *S* and (ii) all the vertices of *CH* are part of the reconstructed outer boundary. As an analogy, *CH* can be considered as a rubber band and the rubber band is shrunk (Fig. 3(b)) to compute the outer boundary as shown in Fig. 3(c). An analogous structure to that of the convex hull is not available for the hole detection. Hence, identifying the initial hole region to start with, is a challenging one.

1.1.2. Applications

Hole detection has various applications in fields such as face recognition, model retrieval, pattern recognition etc. Hole detection is used in face detection algorithms, where a hole mapping is used to detect certain facial characteristics such as mouth, nose, eyes and ears [19]. Hole detection has also applications in areas such as three-dimensional (3D) model retrieval system [12] and 2D point set matching. 3D point sets can be visualized by a collection of 2D views and it is easier to obtain the visual similarity between 2D point sets, if both inner and outer boundaries are re-

constructed and thus the 3D model retrieval can be made more efficient. Specifically, Computer Aided Design (CAD) models are characterized by features like holes, tunnels, ribs and helixes [12]. Outer and inner boundary detected 2D point sets of CAD models make the matching more effective and accurate.

Other applications of hole detection are in the areas of Wireless Sensor Networks (WSNs) and power systems. Detecting the holes is a deciding factor for the efficiency of communication in WSNs [10]. Island (hole) formation in the power systems is a causality factor which has to be considered for the study of security analysis and control of power systems [11].

1.1.3. Related work

Unlike in the case of outer boundary detection problem, to the best of our knowledge, only a few works such as [2,4,7,8,13,14,16,17] exist for hole detection, perhaps because the latter problem is more challenging than the former. Most of the existing works are Delaunay triangulation based, except the one proposed in [8].

α -shape is the space generated by connecting point pairs that can be touched by an empty disk of radius α [7]. The points of weighted α -shape [14] are the vertices of the Voronoi diagram and the centers of the Delaunay circle having radius greater than the specified threshold value, with weights associated with points in sparse regions. V. Kurlin proposed a method to compute number of holes from a given noisy point cloud, based on topological persistence [13]. The above methods are parametric, in which a parameter was tuned to obtain different outputs for the same input. Even though parameter tuning provides flexibility for the user to select the shape based on their requirements, it is very tedious to tune the parameter to obtain the best perceived shape.

Crawl through neighbors (crawl) [16], reconstruction of RGG (Relaxed Gabriel Graph - output of 2D reconstruction algorithm [17]), which is a collection of most of the Gabriel edges and a few non-Gabriel edges induced by a Delaunay triangulation), crust [2], NN-crust [4] are non-parametric methods, which detect both outer and inner boundaries. Crawl is a Delaunay triangulation based method, designed for boundary sample. RGG is designed for handling dot pattern as input, which detects a hole only if the corresponding region in the point set has a structure in which a fat triangle is surrounded by a set of thin triangles as shown in Fig. 4(a). For a general point set, there is no guarantee that a hole region follows a particular structure, and hence the hole may be of any structure such as shown in Fig. 4(b), whose reconstructed holes are shown in Figs. 4(c)–4(d). Hence, it is challenging to develop a non-parametric algorithm, which in practice, works irrespective of the structure of the hole. Crust and NN-crust are designed to work for boundary samples. An approximate positioning of network nodes near the hole boundaries was done in [8].

1.1.4. Unified method for reconstruction of RBS and RDP

Given an input point set, it is not easy to find out whether it is a boundary sample or dot pattern. Hence, apart from the challenges of reconstruction problem in general, there is a requirement for a unified method for hole detection, which works for both boundary sample and dot pattern. Simple shape [9], a parametric algorithm that works for both BS and DP, can reconstruct only outer boundary. The existing hole detection algorithms are either designed for RBS [2,4] or for RDP [17]. Hence, it is challenging to develop a unified algorithm for hole detection which works for both RBS and RDP. The primary motivation of a unified approach for hole detection is to provide an approach which is independent of the nature of the input.

1.2. Our contributions

In this paper, we propose a Delaunay triangulation based algorithm for hole detection of a planar point set, assuming that the best approximation of the outer boundary of the point set has already been obtained, by an outer boundary detection algorithm such as [5,15,17], where the exterior edges of Delaunay triangulation are removed. (Note that the results of our algorithm is not dependent on the choice of the outer boundary detection algorithm.) The proposed algorithm for hole detection works for both types of inputs - dot pattern and boundary sample unlike a few other algorithms [13,16,17] which are focused on a particular input type. Our algorithm uses disk constraint [15] and regularity constraint [5]. Unlike [5,15] which start from an exterior edge, our algorithm starts from an interior edge that is part of a highest area triangle. Moreover, the algorithm does not assume any structure for a hole as in [17]. Our contributions are listed as follows:

- We propose a method for hole detection with the following characteristics:
 - A Delaunay triangulation based strategy to efficiently detect the presence of holes.
 - Unified method which works for both boundary sample and dot pattern.
 - Non-parametric method, so that tuning of parameter can be avoided.
 - Detects holes irrespective of their structure.
 - Capable of reconstructing multiple holes.
- We perform theoretical analysis of the algorithm, assuming r -sampling.
- We conduct extensive comparative studies and demonstrate that our algorithm works better or comparable with existing methods.
- We perform experiments varying point densities and distributions and they demonstrate that the algorithm works independent of sampling.

2. Basic definitions and notations

Let $S = \{p_1, p_2, \dots, p_n\}$ be the set of n points/sites in a plane, where n is a positive integer greater than 2. An edge between two points p_i and p_j is denoted as e_{ij} . We also use e_i to denote an edge, when the endpoints of the edge are not relevant to explain the context. The length of an edge e_{ij} is denoted as $\|e_{ij}\|$. A triangle formed by three points p_i, p_j and p_k is denoted as \triangle_{ijk} . We also assume that the input point set is without noise (unwanted points), that means the sampling is done exactly from the hole boundaries so that no unwanted points are sampled.

Definition 1. Voronoi diagram of S is a subdivision of the plane to n cells, one for each site in S , with a property that a point q lies in the cell corresponding to a site p_i if and only if distance $(q, p_i) < \text{distance } (q, p_j)$ for each $p_j \in S$ where $i \neq j$, where distance (q, p_i) denotes the Euclidean distance between the points q and p_i .

Definition 2. Delaunay triangulation (DT) of S is a straight line dual graph of Voronoi diagram of S .

A Delaunay triangulation overlaid on top of a Voronoi diagram for a set of points is shown in Fig. 5(a).

Definition 3. Exterior triangle (ET) of a Delaunay Triangulation is a triangle which has at least one edge which is not shared by (part of) any other triangle. Exterior edge of an ET is the edge which is not shared by (part of) any other triangle in DT. An edge which is not an exterior edge is known as an interior edge.

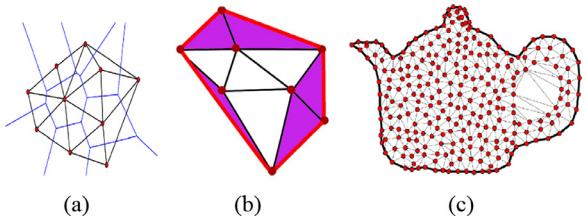


Fig. 5. (a) Delaunay triangulation overlaid on top of a Voronoi diagram (b) Exterior triangles (in purple color), exterior edges (in red color), interior edges (in black color) (c) Outer boundary reconstructed triangulation (G). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

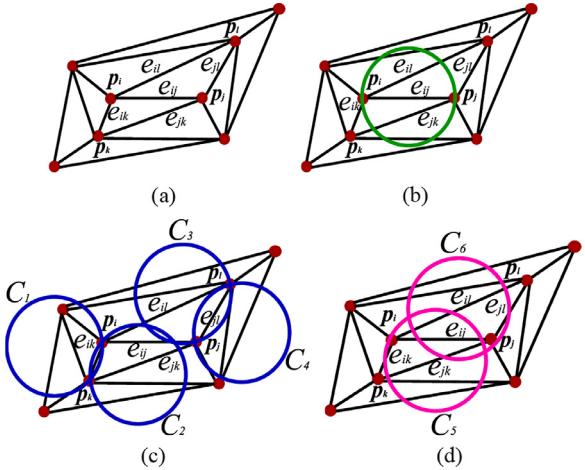


Fig. 6. (a) Interior edge e_{ij} shared by \triangle_{ijk} and \triangle_{ijl} (b) Diametric disk on interior edge e_{ij} with diameter $\|e_{ij}\|$ (c) Chord disks C_1 & C_2 on e_{ik} , C_3 & C_4 on e_{jl} with same diameter $\|e_{ij}\|$ (d) Midpoint disks C_5 & C_6 on e_{ll} with same diameter $\|e_{ij}\|$.

The purple colored triangles, the red colored edges and the black colored edges in Fig. 5(b) are the exterior triangles, exterior edges and interior edges, respectively.

Definition 4. Outer boundary reconstructed triangulation (OBRT) is the output of an outer boundary detection algorithm such as [5,15,17], where the exterior edges are removed from the Delaunay triangulation of the input point set.

OBRT is computed by an outer boundary detection algorithm [5,15,17]. OBRT for a teapot point set is shown in Fig. 5(c), where the solid black line indicates the reconstructed outer boundary. It can be noticed that OBRT is a simplicial complex, where a simplicial complex is defined as a collection of simplices. (A k -simplex is defined as the nondegenerate convex hull of $k + 1$ geometrically distinct points, $v_0, v_1, \dots, v_k \in \mathbb{R}^d$ where $k \leq d$ [3]. Points, edges, triangles and tetrahedra are all examples of simplices.) For easy reference, we denote OBRT as G for further discussions.

Definition 5. Let e_{ij} be an edge under consideration which is shared between two triangles \triangle_{ijk} and \triangle_{ijl} . Diametric disk of an edge e_{ij} is a disk with center as the midpoint of the edge and with diameter $\|e_{ij}\|$. Chord disk of an edge e_{ik} or e_{jk} or e_{il} or e_{jl} is a disk with the edge as its chord and with diameter $\|e_{ij}\|$. Midpoint disk of an edge e_{ik} or e_{jk} or e_{il} or e_{jl} is a disk whose center is the midpoint of the edge and with diameter $\|e_{ij}\|$.

Note that, in this paper, the term disk means an open two-dimensional disk excluding its boundary circle. Consider an interior edge e_{ij} on DT, as shown in Fig. 6(a). It can be observed that an interior edge is shared by (part of) two triangles. For example, the edge e_{ij} is shared by \triangle_{ijk} and \triangle_{ijl} , as shown in Fig. 6(a). Fig. 6(b)

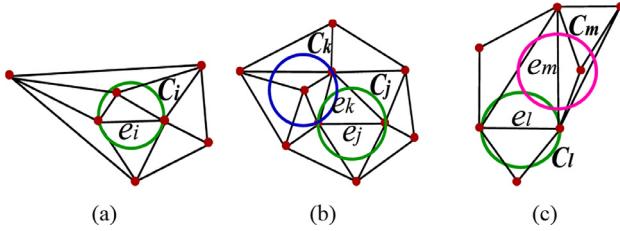


Fig. 7. (a) Non-empty diametric disk C_i on e_i (b) Empty diametric disk C_j on e_j and non-empty chord disk C_k on e_k (c) Empty diametric disk C_l on e_l and non-empty midpoint disk C_m on e_m .

shows a diametric disk on the edge e_{ij} . It is to be noted that both chord disk and midpoint disks are associated with the adjacent edge(s) of e_{ij} on both the triangles (shared by e_{ij}), with diameter $\|e_{ij}\|$. Chord disk of e_{ik} exists only when $\|e_{ij}\| > \|e_{ik}\|$ and there are two chord disks associated with one edge. Chord disks C_1 & C_2 on e_{ik} , C_3 & C_4 on e_{jl} with same diameter $\|e_{ij}\|$ are shown in Fig. 6(c). Midpoint disk of e_{jk} exists only when $\|e_{ij}\| \leq \|e_{jk}\|$. Fig. 6(d) shows midpoint disks C_5 on e_{jk} and C_6 on e_{il} .

2.1. Disk and regularity constraints

Let e_{ij} in Δ_{ijk} of G be the edge under consideration, where G is outer boundary reconstructed triangulation of the point set. Our algorithm checks two constraints before removing an edge e_{ij} from G : (i) disk constraint and (ii) regularity constraint. Disk constraint checks whether a disk on an edge e_{ij} contains any point other than p_i and p_j , or in other words if the disk is non-empty. Non-emptiness of the disk on e_{ij} implies, e_{ij} is longer in the local neighborhood and so it can be considered for removal.

Definition 6. Disk constraint is satisfied if any of the disks is non-empty. For that:

- Check diametric disk on e_{ij} with diameter $\|e_{ij}\|$ is non-empty
- If diametric disk is empty:
 - Check any of the chord disks with the same diameter $\|e_{ij}\|$ (on the adjacent edge(s) of e_{ij}) is non-empty
 - If all the chord disks are empty:
 - * Check any of the midpoint disk(s) with the same diameter $\|e_{ij}\|$ (on the adjacent edge(s) of e_{ij}) is non-empty

Fig. 7(a) shows a non-empty diametric disk C_i on e_i . An empty diametric disk C_j on e_j and a non-empty chord disk C_k on e_k are shown in Fig. 7(b). Fig. 7(c) shows an empty diametric disk C_l on e_l and a non-empty midpoint disk C_m on e_m .

When e_i is considered for removal (after satisfying disk constraint), the regularity constraint is checked on $G - e_i$, where $G - e_i$ is G without e_i .

Regularity constraint is based on a structure viz. cut vertex on the simplicial complex G . A vertex in a simplicial complex is a cut vertex, upon whose removal results in an increased number of components.

Definition 7. A simplicial complex satisfies regularity constraint if it does not have any cut vertex.

3. Hole detection algorithm

In this section, we discuss the intuition, observation and the algorithm for detection of holes.

3.1. Intuition

As pointed out in Section 1.1.1, the major challenge of the hole detection problem is identifying the presence of a hole. We assume

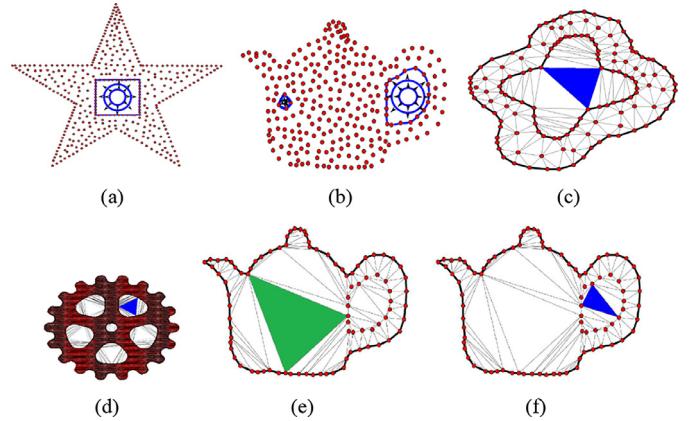


Fig. 8. (a)–(b) Rubber band analogy for hole boundary detection (c)–(d) Highest area triangle (Δ) in blue as initial hole (e) Non-valid highest area triangle (in green) (f) Valid highest area triangle (in blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that a small rubber band (Fig. 8(a), and (b)) is placed in an arbitrary position. We can then grow the rubber band (unlike shrinking the rubber band in the case of convex hull for the outer boundary) to obtain the boundary of the hole. As shown in Fig. 8(b), the rubber band can be placed in more than one arbitrary positions.

3.2. Observation

From the observations of holes of different shapes in Fig. 8(c) and (d), the highest area triangle Δ (shown in blue color) can be considered as an initial candidate (rubber band) for hole region for the algorithm due to the following reasons: (i) Δ is the sparsest region, ie. the region which has no points interior to it (ii) When the initial hole region is grown, the adjacent triangles of Δ contribute to expand the region (iii) It expands towards the lesser area triangles which are closer towards the boundary of the hole and (iv) All three points on Δ are part of the reconstructed hole (analogous to the case of convex hull in Section 1.1.1).

Definition 8. A valid highest area triangle in G is the highest area triangle whose none of the vertices are on the reconstructed outer or inner boundaries.

Even if the triangle colored green in Fig. 8(e) is of highest area in G , it is not a valid highest area triangle. On the other hand, the triangle colored blue in Fig. 8(f) is a valid one.

3.3. The algorithm

From the given point set, DT is constructed and its outer boundary is reconstructed using an outer boundary reconstruction algorithm [5,15,17]. Detect a valid highest area triangle Δ_1 from the outer boundary reconstructed triangulation G . This is considered as the initial hole. The three neighboring triangles of Δ_1 are pushed into priority queue (PQ), in the descending order of the area of the triangles. Pop the head of PQ to obtain Δ_2 . If disk constraints are satisfied on e_{ij} which is the edge shared between Δ_1 and Δ_2 , and if $G - e_{ij}$ satisfies regularity, then e_{ij} and in turn Δ_2 is removed from G and those become part of the hole. Disk constraint on e_{ij} of Δ_{ijk} is satisfied if one of the disks associated with e_{ij} is non-empty. Regularity constraint on e_{ij} of Δ_{ijk} is satisfied, if p_k is not part of any of the outer or hole boundaries. The edge e_{ij} is retained in G , if either of the constraints is not satisfied. This process is continued until PQ becomes empty. Procedure for single hole detection (Algorithm 1) is called repetitively from Algorithm 2 to obtain multiple holes.

Algorithm 1: HOLE_DETECTION(G).

Input: G (OBRT).
Output: Single hole P .

- 1: Identify valid highest area triangle Δ_{ijk} from G .
- 2: **if** no valid highest area triangle is available **then**
- 3: **return** \emptyset .
- 4: **end if**
- 5: Let P be a hole initialized as Δ_{ijk} .
- 6: Initialize a Priority Queue (PQ) with the adjacent triangles of Δ_{ijk} in the descending order of area.
- 7: **repeat**
- 8: Pop the head of priority queue to Δ_{ijl} .
- 9: Remove e_{ij} and in turn Δ_{ijl} from G , if it satisfies disk constraint and $G - e_{ij}$ satisfies regularity constraint, where e_{ij} is the edge shared with P .
- 10: **if** e_{ij} is removed from G **then**
- 11: Add adjacent triangles of Δ_{ijl} (if they are not already present in PQ and in P) and update PQ appropriately.
- 12: $P = P \cup \Delta_{ijl}$.
- 13: **end if**
- 14: **until** PQ is non empty.
- 15: **return** P .

Algorithm 2: MULTIPLE_HOLE_DETECTION(S).

Input: Input point set, S .
Output: Hole(s), H .

- 1: Reconstruct outer boundary using an outer boundary reconstruction algorithm [5,15,17].
- 2: G = Outer boundary reconstructed triangulation of S .
- 3: $H = \emptyset$.
- 4: **repeat**
- 5: $P = \text{HOLE_DETECTION}(G)$.
- 6: $H = H \cup P$.
- 7: $G = G - P$.
- 8: **until** $P = \emptyset$
- 9: **return** H .

3.3.1. Illustration of the algorithm

Illustration of the working of our algorithm using a teapot shape is given in Fig. 9. The point set and its Delaunay triangulation are shown in Fig. 9(a) and (b), respectively. The outer boundary reconstructed triangulation denoted as G (Fig. 9(c)) is computed using an outer boundary reconstruction algorithm [5,15,17] and is used as input to our algorithm. Fig. 9(d) shows the valid highest area triangle Δ_1 (blue in color) and it is considered as the initial hole. The three adjacent triangles of Δ_1 are shown in green color in Fig. 9(e) and are pushed into priority queue (PQ), in the descending order of their areas, because the hole is likely to grow towards the neighboring highest area triangle. The triangle Δ_2 from the head of PQ (green color triangle in Fig. 9(f)) is popped out and disk constraint is checked for the edge e_i which is shared between Δ_1 and Δ_2 (Fig. 9(g)). The diametric disk on e_i is non-empty as shown in Fig. 9(g), hence there is no need to check non-emptiness of other disks. $G - e_i$ is checked for regularity and since it satisfies regularity, e_i (in turn Δ_2) is removed from G and the new edges (as shown in black color in Fig. 9(h)), are added to the intermediate hole. The intermediate hole is updated as shown in Fig. 9(i) and the adjacent triangles of Δ_2 which are not part of the intermediate hole is pushed into PQ and PQ is updated in the descending order of the area of triangles. Among the triangles which

are there currently in PQ (green colored triangles, adjacent to the intermediate hole in Fig. 9(j)), the highest area triangle is popped out and the disk constraint is checked. The diametric disk on e_j is non-empty as shown in Fig. 9(k) and the regularity constraint on $G - e_j$ is also satisfied. The edges of G (Fig. 9(l)) and the intermediate hole (Fig. 9(m)) are updated. After repeating the process further, another intermediate hole obtained is as shown in Fig. 9(n). The adjacent triangles of the intermediate hole which are in PQ currently are shown in green color (Fig. 9(o)). The highest area triangle among them is popped from PQ and consider the edge e_k on that triangle (Fig. 9(p)). The diametric disk on the edge e_k of the triangle is empty as shown in Fig. 9(q) and there is no chord disk existing on any of the adjacent edges of e_k , so the non-emptiness of midpoint disk is checked. Midpoint disks on both the adjacent edges of e_k are empty as shown in Fig. 9(r) and (s). Hence, e_k is retained as an edge on the boundary of the hole. This process is repeated until PQ is empty. The updated G and the reconstructed hole are as shown in Fig. 9(t). Only one hole is detected in this example (Fig. 9(t)). For multiple hole detection, Algorithm 1 is called repetitively from Algorithm 2 until all the holes are detected.

3.4. Complexity analysis

Algorithm 1 for single hole detection has $O(n \log n)$ and $O(n)$ as time and space complexity, respectively, where n is the number of points in S . The time complexity is $O(n \log n)$ because the major steps of Algorithm 1 and their time complexities are as follows:

- Identifying a valid highest area triangle (Step 1 of Algorithm 1) takes time complexity of $O(n)$, because checking whether all three points of a triangle are marked with a flag, is performed only once and the overall number of comparisons for finding out the valid highest area triangle is the number of triangles, in the worst case. Note that the number of triangles in a DT is $O(n)$.
- The time complexity for construction of priority queue and its updation (Steps 6 & 11 of Algorithm 1) is $O(n \log n)$.
- Checking disk constraint on an edge (Step 9 of Algorithm 1) takes constant time because, for one edge removal, only a constant number of adjacent triangles has to be checked.
- Checking regularity constraint on $G - e_{ij}$ (Step 9 of Algorithm 1) takes constant time because it can be done by checking whether p_k of Δ_{ijk} is already marked with a flag, before removing e_{ij} .

Time complexity of Algorithm 2 for multiple hole detection is $O(h^*n \log n)$, where h is the number of holes detected. No extra space (memory) than $O(n)$ is required for Algorithm 2, implying that the space complexity is $O(n)$.

4. Theoretical analysis for hole detection

Consider an original polygonal object O with zero or more holes. We provide a theoretical guarantee by showing that only the holes present in O are reconstructed. Due to the regularity constraint, our algorithm guarantees that any hole detected is a polygon. The correctness of each reconstructed hole is guaranteed by proving that each hole has edges between every pair of adjacent samples of corresponding hole boundary of O . Note that the adjacent boundary samples are those points which are consecutive on the hole boundary of the polygonal object. Those points which are not adjacent are non-adjacent.

For reconstruction of boundary sample and reconstruction of dot pattern, we assume that an input point set S is sampled from an original polygonal object O using r -sampling (see Definition 9) and a modified version of r -sampling [15], respectively. Let e_{ij} be an edge between the points p_i and p_j on a triangle Δ_{ijk} in G . Let B

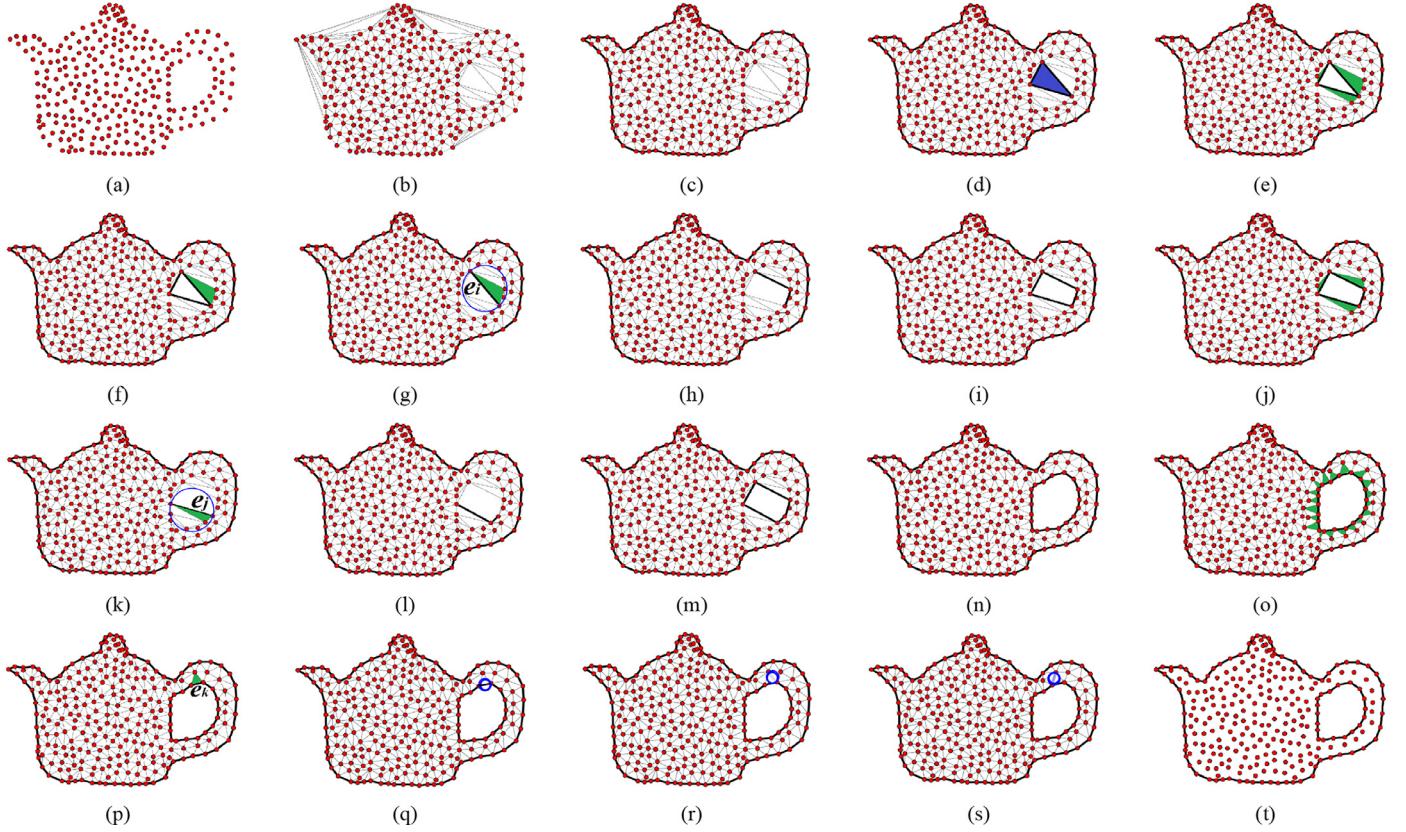


Fig. 9. Illustration of hole detection algorithm: detecting the presence of a hole and expanding it to obtain the hole boundary.

and NB denote the set of boundary and non-boundary edges of a hole, respectively. ROB denotes the reconstructed outer boundary.

Definition 9. In RBS, an input point set S is sampled from a polygonal object O under r -sampling if it satisfies the following constraints:

- $\|p_i - p_j\| < 2r$, for all pairs of adjacent hole boundary samples $p_i, p_j \in S$.
- $\|p_i - p_j\| \geq 2r$, for all pairs of non-adjacent hole boundary samples $p_i, p_j \in S$.

Lemma 4.1. Assuming r -sampling, the circumcenter of a valid highest area triangle \triangle_{fgh} in G lies inside a hole region iff \triangle_{fgh} lies inside the hole boundary.

Proof. In order to prove the forward direction of Lemma 4.1, it is enough to prove its contrapositive: if \triangle_{fgh} does not lie inside the hole boundary, then the circumcenter of \triangle_{fgh} does not lie inside the hole region. For all the other types of triangles except obtuse triangle, circumcenter lies on or inside the triangle itself. Hence, it is enough to prove that there does not exist an obtuse triangle \triangle_{fgh} outside the hole region whose circumcenter is inside the hole region as shown in Fig. 10(a). If there exists \triangle_{fgh} outside the hole region, whose circumcenter is inside the hole region, then there exists an edge on the hole boundary which lies between the circumcenter and e_{fh} . Due to r -sampling, there are edges on the hole boundary within the distance $2r$, which implies there is a Delaunay triangle with at least one point in the circumscribing circle of \triangle_{fgh} (Fig. 10(a)). This contradicts the circumcircle property of Delaunay triangulation and thus the existence of the obtuse triangle \triangle_{fgh} .

Reverse direction of Lemma 4.1 can be proved using a similar argument as that of the forward direction. It can be done by proving that there does not exist \triangle_{fgh} inside the hole boundary, whose circumcenter lies outside the hole region. \square

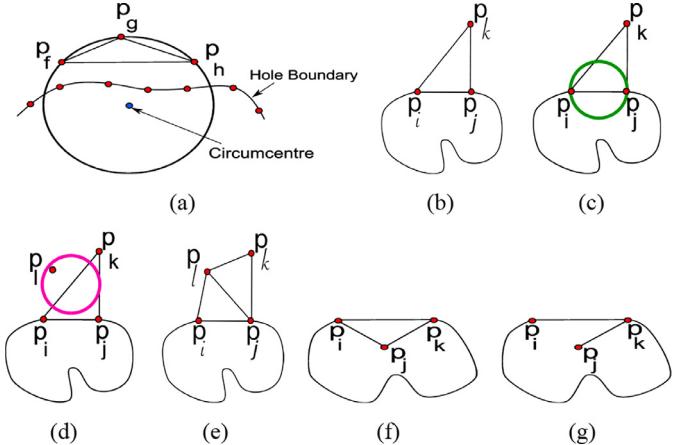


Fig. 10. (a) Circumcircle of \triangle_{fgh} (b) \triangle_{ijk} with $\|p_i - p_j\| < 2r$, $\|p_i - p_k\| > 2r$, $\|p_j - p_k\| > 2r$ (c) Non-empty midpoint disk on e_{ik} (e) Valid Delaunay triangles, Curved part in (b)-(g): intermediate hole (f) \triangle_{ijk} with $\|p_i - p_j\| < 2r$, $\|p_i - p_k\| > 2r$, $\|p_j - p_k\| < 2r$ and e_{jk} is already detected as an edge on the hole boundary (g) Graph which does not satisfy regularity constraint after removal of e_{ij} .

If the circumcenter of a valid highest area triangle in G lies inside a hole region, then it is said to be a potential triangle.

Lemma 4.2. In RBS, starting from a potential triangle in G , Algorithm 1 retains all e_{ij} 's in B .

Proof. Let e_{ij} be a boundary edge of a potential triangle \triangle_{ijk} in Fig. 10(b). The curved part shown in Fig. 10(b)-(g) represents the reconstructed intermediate hole. Three cases are considered: (i) Case-1 : $\|p_i - p_j\| < 2r$, $\|p_i - p_k\| > 2r$, $\|p_j - p_k\| > 2r$. The

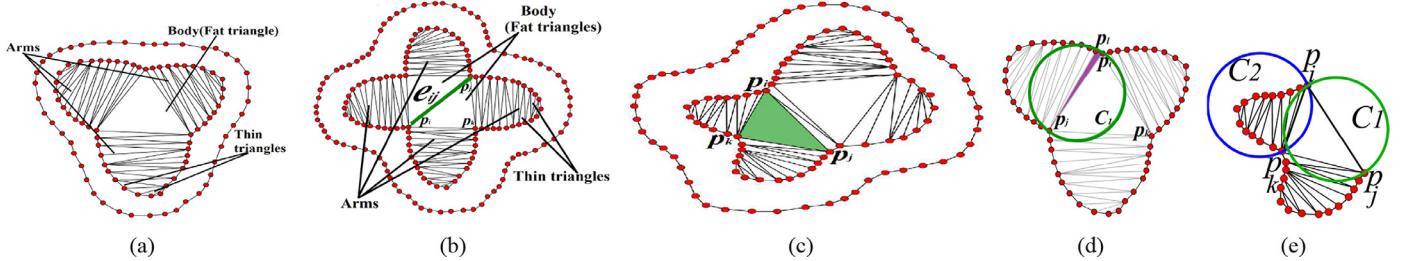


Fig. 11. (a) Hole with body-arm structure: 3-Arm with one fat triangle (b) 4-Arm with two fat triangles and a candidate edge e_{ij} (c) Distorted body-arm structure with distorting edge e_{ij} (d) Non-empty diametric disk C_1 (e) Empty diametric disk C_1 & Non-empty chord disk C_2 .

diametric disk is empty (Fig. 10(c)), because $\|p_i - p_j\|$ is less than both $\|p_i - p_k\|$ and $\|p_j - p_k\|$. $\|p_i - p_j\| > \|p_i - p_k\|$ and $\|p_j - p_k\|$ ensures only midpoint disks exist. If $\exists p_l \in$ any of the midpoint disk (Fig. 10(d)) implies \triangle_{ijk} is an invalid Delaunay triangle whose valid Delaunay triangles are shown in Fig. 10(e).

(ii) Case-2: $\|p_i - p_j\| < 2r$, $\|p_i - p_k\| > 2r$, $\|p_j - p_k\| < 2r$. In this case, if e_{jk} is already detected as an edge on the hole boundary (as shown in Fig. 10(f)) then, even if one of the disks is non-empty, e_{ij} cannot be removed because of regularity constraint (Fig. 10(g)) shows the graph which does not satisfy regularity constraint after removal of e_{ij} and hence e_{ij} is retained. On the other hand, if e_{jk} is not already detected as a hole boundary edge: diametric disk on e_{ij} is empty because $\|p_i - p_k\| > 2r$. Two cases arise here: both disks on p_{ik} and p_{jk} are midpoint disks, disk on p_{ik} , p_{jk} are midpoint and chord disks, respectively. In both the cases, as the corresponding disks are empty, $\exists p_m \in$ midpoint disk on p_{ik} contradicts our assumption that \triangle_{ijk} is a valid Delaunay triangle. Suppose $\exists p_l \in$ chord or midpoint disk on p_{jk} then it implies that \triangle_{ijl} is the hole because of r -sampling.

(iii) Case-3: $\|p_i - p_j\| < 2r$, $\|p_i - p_k\| < 2r$, $\|p_j - p_k\| < 2r$. In this case, all the three edges e_{ij} , e_{ik} and e_{jk} are retained and \triangle_{ijk} is reconstructed as a hole. \square

For the proof for removal of non-boundary edges from G , we borrow the concept of a body-arm structure from [17]. In [17], a hole is visualized as a body-arm structure with a body surrounded by a set of arms as shown in Fig. 11(a), where a body is a set of connected fat triangles (acute triangles) and an arm is a set of thin triangles (obtuse triangles) attached to a fat triangle in the body. Fat triangles in a body are connected in a linear fashion ie. each fat triangle in a body is connected to at least one arm. Fig. 11(a) and (b) show a 3-Arm structure and a 4-Arm structure, respectively. For further details on body-arm structure, please refer [17].

Definition 10. A candidate edge is an edge common to a pair of fat triangles in a body-arm structure.

For example in a body-arm structure with 4-arms, there exists a candidate edge e_{ij} as shown in Fig. 11(b), whereas in the case of 3-arms, there is only one fat triangle and hence there is no candidate edge (Fig. 11(a)).

Definition 11. If a fat triangle occurs in between the thin triangles of an arm of a body-arm structure, then it is known as a distorted body-arm structure. The edge which distorts the body-arm structure is known as a distorting edge.

An example of a distorted body-arm structure is shown in Fig. 11(c), where \triangle_{ijk} is the fat triangle in between the thin triangles and e_{ij} is the distorting edge.

Lemma 4.3. In RBS, starting from a potential triangle of a hole in G , Algorithm 1 removes all $e_{ij} \in NB$ (where e_{ij} is an edge of \triangle_{ijk}) if it satisfies either of the following conditions:

1. The hole is a body-arm structure with the length of the candidate edge longer than either of its adjacent edges.

2. The hole is a distorted body-arm structure with the length of the distorting edge and the candidate edge(s) longer than their adjacent edges.

Proof. Case-1 : If the hole is a body-arm structure with only one fat triangle: Assume that the diametric disk on e_{ij} is empty. Due to the body-arm structure, there exists a thin triangle \triangle_{iji} (purple colored triangle in Fig. 11(d)). Hence, p_l is present in diametric disk C_1 , which leads to a contradiction to the assumption. The proof holds for remaining non-boundary edges in the arms. Hence e_{ij} 's are removed.

Case-2 : If the hole has a body-arm structure with more than one fat triangle and the length of the candidate edge is longer than either of its adjacent edges: The diametric disk C_1 on the candidate edge e_{ij} is empty because of the fat triangle, as shown in Fig. 11(e). Due to the body-arm structure, there exists a thin triangle that has e_{ik} as one of its edges. The chord disk C_2 on e_{ik} is non-empty because of the presence of the thin triangle (Fig. 11(e)). Hence e_{ij} is removed. As in case-1, all other non-boundary edges of the body-arm structure are removed.

Case-3: The hole is a distorted body-arm structure with the length of the distorting edge & the candidate edge(s) longer than their adjacent edges: All the non-boundary edges of the body-arm structure up to the distorting edge is removed as in case-1 or case-2. The removal of distorting edge reduces to the proof of case-2 (Fig. 11(e)). \square

To summarize, the theoretical analysis ensures the correctness of the reconstructed holes, assuming the input point set is sampled from an original polygonal object under r -sampling. This was done by showing that all the boundary edges of a hole are retained and all non-boundary edges are removed (under specific conditions), by our hole detection algorithm.

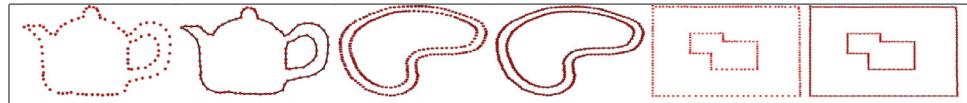
For RDP, as there are non-boundary points (unlike RBS), its guarantee can be proved in similar lines as that of RBS, assuming a slight modification in r -sampling [15].

5. Results

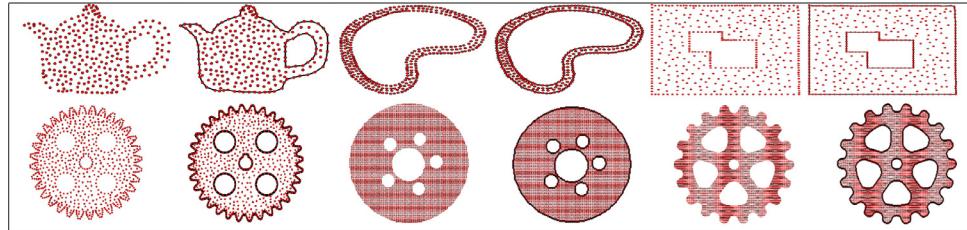
We have used CGAL 4.3 [1] for implementation. To the best of our knowledge, there is no repository for 2D point sets with holes. Hence, the input point sets we used for generating results and comparisons are either generated from 3D models of Engineering Shape Benchmark [12] by converting them to 2D shapes or created by ourselves. Even though a sampling model is assumed for theoretical proofs of our algorithm (reconstruction algorithms in general are theoretically proved assuming a sampling model [2]), the algorithm has been tested on point sets not conforming to any sampling model. For example, in the case of quantitative comparison of our results with other algorithms, we have used input point sets with different point densities and distributions.

Table 1

1st, 3rd & 5th columns: Boundary sample, 2nd, 4th & 6th columns: Output of our algorithm. Concavities (in all the outputs) and sharp corners (last column of 1st row) are captured well.

**Table 2**

1st, 3rd & 5th columns: Dot pattern, 2nd, 4th & 6th columns: Output of our algorithm. Concavities (in all the outputs) and sharp corners (last column of 1st row) and multiple holes (outputs in 2nd row) are captured well.



Input	α -Shape	RGG	Our Result	Input	α -Shape	RGG	Our Result

Fig. 12. Comparative results of RDP with input point sets. Left bottom in each box characterizes the output: G - Good reconstruction of holes, N - Holes not detected. Some of the defects of reconstruction have been circled.

A few results of our algorithm for the boundary sample and dot pattern are shown in [Tables 1](#) and [2](#) respectively. It can be noticed that the holes with various shapes for both BS and DP are captured. Concavities, sharp corners and multiple holes are captured well. It should be emphasized that our algorithm detects holes irrespective of the structure of the hole, without parameter tuning.

5.1. Comparison with existing methods

We have compared our algorithm with six existing methods viz. α -shape [7], RGG [17], crust [2], NN-crust [4], topologically persistent hole detection (TPHD) [13] & crawl [16], for which the codes were available. We have performed both qualitative and quantitative comparisons. For qualitative comparison, we have used point sets generated from 3D models of Engineering Shape Benchmark [12] and those created by ourselves. For quantitative comparison, we have created point sets of standard shapes like square, rectangle and polygons because any error measure (in this paper, L^2 -error norm [5]), compares the reconstructed shape only if it is from a standard shape, whose exact area can be computed.

5.2. Qualitative comparison

[Fig. 12](#) shows the comparison of our results for dot pattern with α -shapes and RGG (Crust, NN crust, TPHD & crawl do not work for dot patterns and the algorithm in [8] does not detect the hole

boundary but detects only the approximate positions to place sensor nodes). G & N in left bottom in each box denote good reconstruction of hole & hole is not detected at all, respectively.

It can be observed that our results are as good as or better than others. Summarizing the comparison of RDP with other methods and ours: (i) our results capture small features like key hole better than α -shape and RGG (ii) concavities are captured by our results either better than or equal to the other two methods.

A few of the results for the qualitative comparison for boundary sample are shown in [Fig. 13](#), which shows α -shape, RGG, crust, NN-crust, TPHD, crawl & our result in that order. Summarizing the comparison of RBS with other methods and ours: (i) our results capture small features like key hole better than α -shape and RGG, and equal to that of crust, NN-crust, TPHD & crawl (ii) sharp features are captured better by our method than the other four methods and as good as TPHD & crawl (iii) concavities are captured by our results either better than or equal to other methods. It is to be emphasized that our approach works independent of the type of the input point set.

5.3. Quantitative comparison varying point densities & distributions

For quantitative comparison, we have considered only the error(s) caused by the reconstructed hole boundaries and not the ones by the outer boundary, since we are comparing the perfor-

Input	α - Shape	RGG	Crust	NN-Crust	TPHD	Crawl	Our Result

Fig. 13. Comparative results of RBS with input point sets. Left bottom in each box characterizes the output: G – Good reconstruction of holes, N – Holes not detected. Some of the defects of reconstruction have been circled.

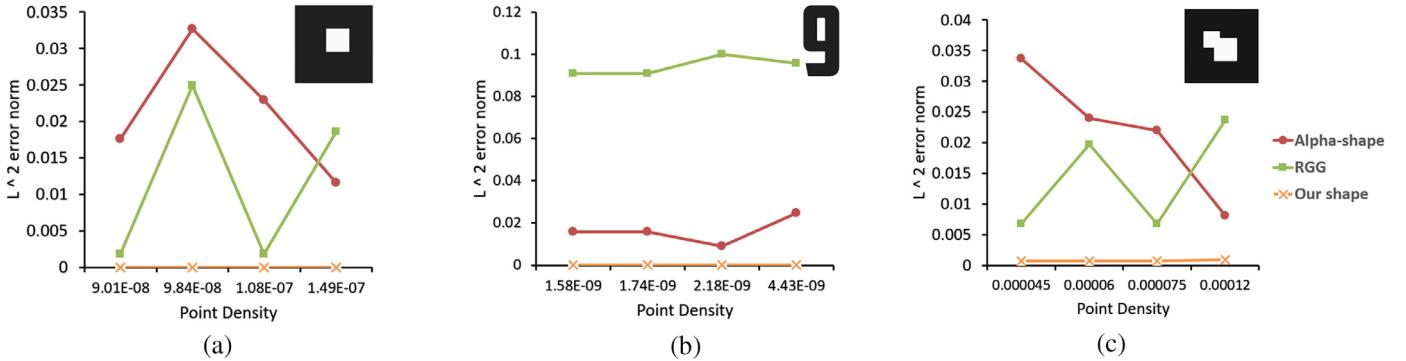


Fig. 14. Plots for point density vs L^2 error norm along with original shape in the inset which show that L^2 -error is less in the case of our result compared to other methods.

mance of the hole detection algorithms. The error measure we used for quantitative comparison is L^2 error norm [5], which provides the symmetric difference between the areas of both the set of holes, which provides a measure on how much the reconstructed hole(s) differs from the original. If the value of L^2 error is zero, that means there is no difference between the original and reconstructed hole(s).

We analyzed how L^2 error norm varies with point sets with different point density, where point density is the number of points per unit area. From the plots shown in Fig. 14, it can be observed that our results have lesser L^2 error than the other two methods.

We also analysed how L^2 error norm varies with heterogeneity in point distribution. We considered three types of distributions : (i) Random Dense Boundary (RDB)- where the points on the hole boundary are dense and rest of the points are randomly distributed (ii) Random Sparse Boundary - (RSB) - where the points on the hole boundary are sparse and the rest of the points are randomly distributed and (iii) Random (R) - where all the points are

randomly distributed. Note that the term “boundary” in RDB and RSB refers to hole boundary and we compare only the hole boundaries and not the outer boundaries. Hence, for RDB, RSB and R, the points on the outer boundary are not taken into account, and only the points on the hole boundary and those between outer and hole boundary are considered. The plots in Fig. 15 show that in RDB and RSB our results are better than the other two methods. However, in random (R) distribution, our results have more L^2 error than either of the methods. A few results used for obtaining the plots for varying point distribution (RDB & RSB) and the plots for varying the point density are shown in Fig. 16.

6. Discussion

Crawl, Crust and NN-crust work only for boundary samples, unlike our unified method. The outputs of crawl, crust and NN-crust can be open curves (hence it is difficult to find out whether they are holes), whereas the holes detected by our

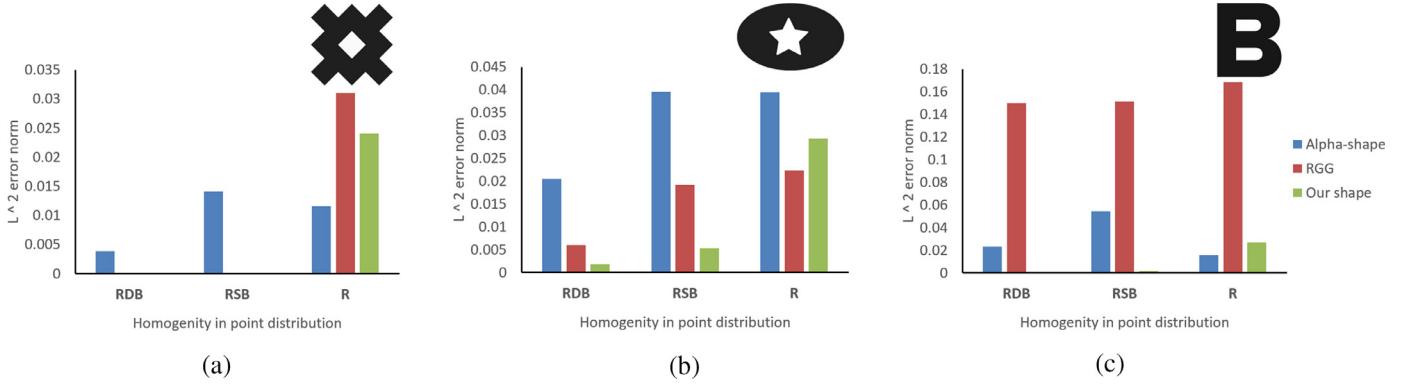


Fig. 15. Plots for point distribution vs L^2 error norm along with original shape in the inset which show that in RDB & RSB, L^2 -error is less in the case of our result compared to other methods. But, in Random (R) distribution, our result degrades its performance than α -shape and outperforms RGG (Fig. (a) & (c)) whereas our result is not as good as RGG but better than α -shape (Fig. (b)).

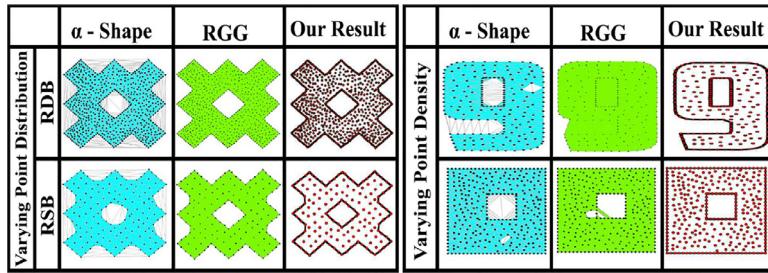


Fig. 16. A few results used for obtaining Point distribution vs L^2 error norm plots & Point density vs L^2 error norm plots.

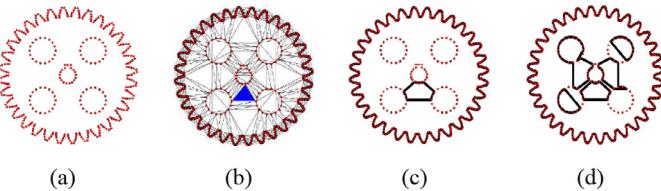


Fig. 17. Our method: (a) BS (b) Valid highest area \triangle in non-hole region (c) Initially detected hole (d) Result for RBS .

method are polygons. RGG detects holes, only if the holes satisfy a specified structure, unlike our method which is irrespective of the structure. Summarizing the characteristics of the proposed hole detection algorithm, our algorithm is: (i) simple, efficient and easy to implement. (ii) non-parametric, which avoids parametric tuning. (iii) unified one for both boundary sample and dot pattern. (iv) detects holes irrespective of their structure. (iv) capable of detecting sharp corners (if it is not acute angled), concavities and small features like holes.

6.1. Limitations

Even if our method has many advantages, one of the limitations of our method is that if the valid highest area triangle is not in the hole region, our algorithm does not perform well (Fig. 17). The boundary sample and G are shown in Fig. 17(a) & (b), respectively. It can be seen that the valid highest area triangle \triangle_1 (shown in blue color in Fig. 17(b)) is not in the hole region. (In this case \triangle_1 does not have its circumcenter inside the hole boundary and hence it is not a potential triangle as we pointed out in Lemma 4.1.) Initial hole reconstructed & four more holes detected further are shown in Fig. 17(c) & (d), respectively. Even if our algorithm does not work for boundary sample in this example, it detects the holes

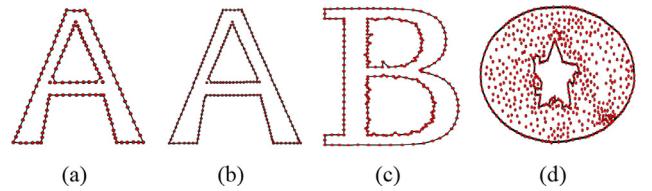


Fig. 18. (a)–(b) Hole with acute sharp angles is not captured even after increasing the point density. (c) Noisy input point set creates over-digging (d) Result of random distribution which over-digs.

including the key hole in the case of dot pattern (2nd row, 2nd column of Table 2).

Our algorithm is not able to capture a hole which has acute sharp angles, even after increasing the point density, as shown in Fig. 18(a) & (b). The reason is, the hole with an acute sharp angle does not satisfy the body-arm condition specified in Lemma 4.3. If a hole has non-acute sharp angles, our algorithm captures the sharp corners very well (Last figure on the first row of Tables 1 & 2). If the input point set is noisy, the hole boundary is not captured well as shown in Fig. 18(c). When the input point set is of random distribution, our algorithm degrades its performance by over-digging (Fig. 18(d)).

7. Conclusion

We have designed a strategy to detect the presence of a hole and developed a Delaunay triangulation based method for hole detection of a planar point set which works both for boundary sample and dot pattern. Our algorithm detects holes irrespective of the structure of the hole. We evaluated our algorithm both theoretically and experimentally. Theoretical analysis is provided under r -sampling for specific hole structures. We have conducted extensive comparative studies with the existing methods and demonstrated

that our algorithm works better or comparable with other methods. We have also tested the algorithm with varying point densities and distributions, showing a better performance under L^2 -error norm. We also observed that it does not perform well in random point distributions. Our method has been tested on input point sets independent of sampling, with comparable or better performance with other methods. If the highest area triangle detected is not in the hole region, performance of our algorithm degrades. One of the directions of future work is to detect holes in a noisy point set and another one is an extension of this work to 3D.

Acknowledgements

The first author would like to thank National Institute of Technology (NIT) Calicut for granting permission to conduct her research at IIT Madras under QIP scheme. Other authors would like to thank IIT Madras for its support.

References

- [1] The CGAL Project, *CGAL User and Reference Manual*, CGAL Editorial Board, 4.3 edition, 2013.
- [2] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th annual conference on computer graphics and interactive techniques. SIGGRAPH '98. New York, NY, USA: ACM; 1998. p. 415–21. ISBN 0-89791-999-8.
- [3] Desbrun M, Kanso E, Tong Y. Discrete differential forms for computational modeling. Basel: Birkhäuser Basel; 2008. p. 287–324. ISBN 978-3-7643-8621-4. doi:10.1007/978-3-7643-8621-4_16. http://dx.doi.org/10.1007/978-3-7643-8621-4_16.
- [4] Dey TK, Kumar P. A simple provable algorithm for curve reconstruction. In: Proceedings of the symposium on discrete algorithms SODA'99; 1999. p. 893–4.
- [5] Duckham M, Kulik L, Worboys MF, Galton A. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. Pattern Recogn 2008;41(10):3224–36.
- [6] Edelsbrunner H. Shape reconstruction with delaunay complex. In: Lucchesi CL, Moura AV, editors. Latin. Lecture Notes in Computer Science, 1380. Springer; 1998. p. 119–32. ISBN 3-540-64275-7.
- [7] Edelsbrunner H, Kirkpatrick DG, Seidel R. On the shape of a set of points in the plane. IEEE Trans Inf Theory 1983;29(4):551–8.
- [8] Funke S, Klein C. Hole detection or: “how much geometry hides in connectivity?”. In: Proceedings of the twenty-second annual symposium on computational geometry. SCG '06. New York, NY, USA: ACM; 2006. p. 377–85. ISBN 1-59593-340-9. <http://doi.acm.org/10.1145/1137856.1137911>.
- [9] Gheibi A, Davoodi M, Javad A, Panahi F, Aghdam MM, Asgaripour M, et al. Polygonal shape reconstruction in the plane. IET Comput Vis 2011;5(2):97–106.
- [10] Ghosh P, Gao J, Gasparri A, Krishnamachari B. Distributed hole detection algorithms for wireless sensor networks. In: Proceedings of the IEEE 11th international conference on mobile ad hoc and sensor systems (MASS); 2014. p. 257–61. doi:10.1109/MASS.2014.25.
- [11] Guler T, Gross G. Detection of island formation and identification of causal factors under multiple line outages. IEEE Trans Power Syst 2007;22(2):505–13. doi:10.1109/tpwrs.2006.888985. <http://dx.doi.org/10.1109/tpwrs.2006.888985>.
- [12] Jayanti S, Kalyanaraman Y, Iyer N, Ramani K. Developing an engineering shape benchmark for CAD models. Comput-Aided Des 2006;38(9):939–53. doi:10.1016/j.cad.2006.06.007. <http://dx.doi.org/10.1016/j.cad.2006.06.007>.
- [13] Kurlin V. A fast and robust algorithm to count topologically persistent holes in noisy clouds. In: Proceedings of the computer vision and pattern recognition CVPR 2014;; 2014. p. 1458–63.
- [14] Melkemi M, Djebali M. Weighted \mathcal{A} -shape: a descriptor of the shape of a point set. Pattern Recogn 2001;34(6):1159–70.
- [15] Methirumangalath S, Parakkat AD, Muthuganapathy R. A unified approach towards reconstruction of a planar point set. Comput Graph 2015;51:90–7. International Conference Shape Modeling International.
- [16] Parakkat AD, Muthuganapathy R. Crawl through neighbors: a simple curve reconstruction algorithm. Comput Graph Forum 2016;35(5):177–86. doi:10.1111/cgf.12974. <https://doi.org/10.1111/cgf.12974>.
- [17] Peethambaran J, Muthuganapathy R. A non-parametric approach to shape reconstruction from planar point sets through Delaunay filtering. Comput-Aided Des 2015;62:164–75. <http://dx.doi.org/10.1016/j.cad.2014.12.002>.
- [18] Seref O, Zobel CW. Recursive voids for identifying a nonconvex boundary of a set of points in the plane. Pattern Recogn 2013;46(12):3288–99.
- [19] Udo Ahlers UZ, R Rajagopalan. Model-free face detection and head tracking with morphological hole mapping. In: Proceedings of the 13th european signal processing conference (EUSIPCO'05), Antalya, Turkey, September 4–8, 2005; 2005.