Computer-Aided Design 62 (2015) 164–175

Contents lists available at ScienceDirect

Computer-Aided Design

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

A non-parametric approach to shape reconstruction from planar point sets through Delaunay filtering☆

Jiju Peethambaran, Ramanathan Muthuganapathy *

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

HIGHLIGHTS

- A fully automatic algorithm using the structural properties of Delaunay triangles.
- External boundary as well as an internal hole detection have been addressed.
- Demonstrate the efficacy with varying point set densities and distributions.
- Theoretical guarantee of the algorithm has been presented.

ARTICLE INFO

Article history:
Received 14 November 2013
Accepted 4 December 2014

Keywords:
Shape reconstruction
Delaunay triangulation
Hole detection
Relaxed Gabriel graph
α-shape
χ-shape

ABSTRACT

In this paper, we present a fully automatic Delaunay based sculpting algorithm for approximating the shape of a finite set of points S in $\mathbb{R}^2$. The algorithm generates a relaxed Gabriel graph (*RGG*) that consists of most of the Gabriel edges and a few non-Gabriel edges induced by the Delaunay triangulation. Holes are characterized through a structural pattern called as **body-arm** formed by the Delaunay triangles in the void regions. *RGG* is constructed through an iterative removal of Delaunay triangles subjected to circumcenter (of triangle) and topological regularity constraints in $O(n \log n)$ time using $O(n)$ space.

We introduce the notion of directed boundary samples which characterizes the two dimensional objects based on the alignment of their boundaries in the cavities. Theoretically, we justify our algorithm by showing that under given sampling conditions, the boundary of *RGG* captures the topological properties of objects having directed boundary samples. Unlike many other approaches, our algorithm does not require tuning of any external parameter to approximate the geometric shape of point set and hence human intervention is completely eliminated. Experimental evaluations of the proposed technique are done using L^2 error norm measure, which is the symmetric difference between the boundaries of reconstructed shape and the original shape. We demonstrate the efficacy of our automatic shape reconstruction technique by showing several examples and experiments with varying point set densities and distributions.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Given a finite set of points, $S \subseteq \mathbb{R}^2$, distributed across a known or unknown object, shape reconstruction deals with the construction of a polygon (could be with holes) that best captures the geometric shape of S (Fig. 1). The problem finds various applications in computer graphics, computer vision, pattern recognition [1], Geographical Information System [2,3], study of physical objects like stars in astronomy [4], molecular shape description in biology [5], and target area monitoring in wireless sensor networks [6].

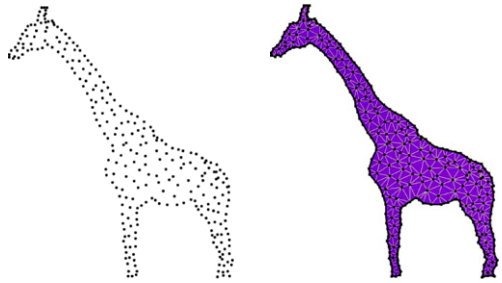
There exist few challenges in approximating the shape of a point set. First, it is unclear what constitutes the best approximation for the geometric shape of a point set mainly due to a lack of precise mathematical definition for its optimal shape. Second, the point set shapes are highly subjective in nature and often depend on a specific application context or other human cognitive factors. As a consequence, the shapes perceived by humans for a majority of point sets vary and reaching a consensus on the optimum shape is an extremely difficult task. The rich variety of shapes available in the nature and the heterogeneity of point sets further weaken a well-defined formulation of the shape approximation problem [7].

Delaunay based shape reconstruction approaches such as α-shape [8], $\mathcal{A}$-shape [9] and χ-shape [10] depend on external parameters to construct the geometric shape of a point set. As a result, a family of shapes is generated for a given point set. For

☆ This paper has been recommended for acceptance by Dr. Vadim Shapiro.

* Corresponding author.
E-mail address: emry01@gmail.com (R. Muthuganapathy).

http://dx.doi.org/10.1016/j.cad.2014.12.002
0010-4485/© 2014 Elsevier Ltd. All rights reserved.



(a) Point set. (b) Shape.

Fig. 1. Shape approximated by the proposed method.

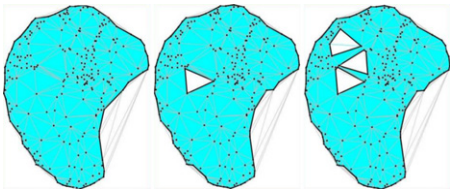


Fig. 2. α-shapes of brain point set for different values of α.
Source: Courtesy: CGAL Library [11].

example, α-shape family of a brain point set consists of a set of different but structurally close shapes for different values of α as demonstrated in Fig. 2. Though it gives the flexibility of choosing the best shape that suits an application, it has few limitations. For instance, choosing the best shape from a set of similar shapes demands a great deal of human involvement which might induce humanly errors thereby affecting the accuracy of the whole system or application. Moreover, since everyone is cognitively different from each other, reaching a consensus on the desired shape by different individuals becomes a challenging task. Considering these facts, it is often desirable to have an automatic algorithm that constructs the geometric shape of a point set.

In general, two approaches have been adopted to evaluate shape reconstruction results. In one approach, the results are experimentally evaluated against a known original shape with respect to certain error parameters. Error is estimated as the area of symmetric difference between the original shape and the reconstructed result [10]. The evaluation of the results are made extensive and authentic by carrying out experiments like varying point density and homogeneity in point distribution. In another approach, algorithms are analyzed theoretically for geometric and topological guarantees on the output shape under some input sampling assumptions [12]. We use both the approaches for evaluating the proposed shape reconstruction algorithm.

2. Related work

A plethora of work has been proposed for geometric shape reconstruction from a planar point set. Though a full review of all the shape reconstruction algorithms is beyond the scope of this paper, we briefly mention some closely related methods based on Delaunay triangulation filtering and proximity graphs.

In a survey on shape reconstruction [7], Edelsbrunner elaborates on how different shape reconstruction methods restrict the Delaunay triangulation to arrive at their respective shapes. The shape is represented by the underlying space of a sub complex obtained from the Delaunay triangulation using a corresponding filtering procedure. In $\mathbb{R}^2$, α-shape of a point set S is generated by connecting the points $p, q \in S$, which are touched by an empty disk of radius α [8]. However, α-shape construction requires an external parameter, α and there exist some point sets for which the α-shape family does not contain the appropriate shape [9]. In another work [13], Edelsbrunner described WRAP algorithm which is based on ideas from Morse theory applied to the flow maps derived from certain distance functions. A variant of WRAP algorithm, proposed by Ramos et al. [14] provides geometric and topological guarantees on reconstructed surfaces within the ϵ-sampling model. Though WRAP algorithm can be extended to any dimension, the commercially available software, Geomagic WRAP based on [13], mainly focuses on three-dimensional surface reconstruction.

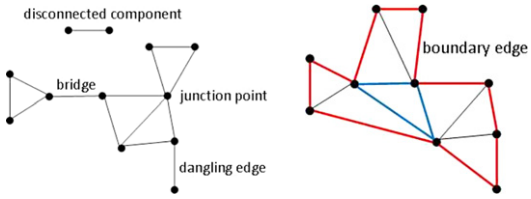
The family of $\mathcal{A}$-shapes proposed by Melkemi et al. [9] contains α-shapes and the crust [12] as special cases. Determination of a proper $\mathcal{A}$ in $\mathcal{A}$-shape computation is controlled by two parameters, α and t. The first parameter, $\alpha \geq 0$ controls the resolution and the second parameter, $t \in [0, 1]$, interpolates between the unweighted case and the case where points are weighted by the local density [7].

A related notion for characterizing a point set distribution is characteristic shape (or χ-shape) [10]. χ-shape algorithm removes external edges of Delaunay triangulation until the largest edge on the boundary is less than a threshold value. Termination and efficiency of the algorithm and the boundary of χ-shape are dependent on the threshold value. Further, it generates only simply connected shapes (holes cannot be detected). Recently, variants of α-shapes such as k-order α-hulls [15] and LDA-α-complex [16] have been proposed for shape reconstruction. LDA-α-complex effectively uses the local density variation found in the point sets for detecting the hollow regions. Relaxing the constraint that the generated shape should possess a closed boundary, proximity graphs such as Relative neighborhood graph (RNG), Gabriel graph, Sphere-of-Influence graph [17] and β-skeletons [18] play a vital role in defining the shape and structure of planar point sets [19].

Methods such as crust, NN-crust and normalized mesh employ Delaunay triangulation along with a sampling criteria of the input curves or surfaces for providing guarantees on the reconstruction [12,20,21]. All these algorithms are extended to $\mathbb{R}^3$, and reconstructs surfaces from point sets with provable guarantees. A detailed description on these algorithms is available in [21]. Recently, the authors [22] proposed a Delaunay sculpting technique for reconstructing curves and surfaces from boundary samples and provide theoretical results on the bound of the sampling parameter ϵ, for which a topologically correct reconstruction is guaranteed. As opposed to this, the proposed algorithm considers points distributed across the entire object as the input and also constructs internal hole boundaries for a wide range of point sets.

2.1. Our contributions

In this paper, we present an automatic algorithm for shape reconstruction from planar point sets which completely eliminates the hassles of human intervention. The proposed method constructs the boundaries for point sets sampled from both simply connected (objects without holes) and multiply connected objects (objects with holes). The novelty of our approach lies in judiciously using the information embedded in the geometry of the Delaunay triangles to avoid parametrization of the algorithm and thereby relieving the end user of the tedious task of tuning an external parameter. We use a combination of a local measure (circumcenter of Delaunay triangles) and a global measure (size of the Delaunay triangles quantified by the circumradius) for the filtration of the Delaunay triangulation. Moreover, our algorithm also constructs boundaries for a wide variety of holes based on the structural arrangements of the Delaunay triangles. To the best of our knowledge, the holes conceptualized as a set of acute Delaunay triangles surrounded by several sets of obtuse Delaunay triangles connected in a particular way is totally new in the arena of hole detection.



(a) Non-regular. (b) Regular.

Fig. 3. Illustration of regular and non-regular simplicial complexes. The constructs which violate the regularity of a graph are shown in Fig. 3(a). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

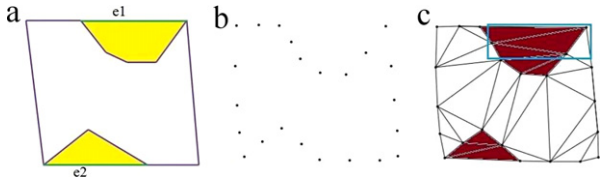


Fig. 4. Illustration of a directed boundary sample of a polygonal object: (a) Polygonal object O along with the cavities and the virtual edges shown in yellow and green colors respectively, (b) Points sampled from ∂O, S, (c) Delaunay triangulation of S with external Delaunay triangles shown in dark red color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The hole concept described in this paper effectively captures the low density regions (regions where the holes are most likely to exist) in a wide range of point sets. However, we lay no claim on the optimality of the reconstructed hole boundaries due to a lack of mathematical definition for a hole in the point sets.

3. Preliminaries

3.1. Geometrical definitions and notations

Let S be a set of n points in the Euclidean plane and $conv(S)$ be its convex hull. $Vor(S)$, $Del(S)$, $int(conv(S))$ and $\partial conv(S)$ denote, respectively, the Voronoi diagram of S, the Delaunay triangulation of S, the interior of $conv(S)$ and the boundary of $conv(S)$. Let $d(p, q) = \|p - q\|$, denote the Euclidean distance between two points $p, q \in S$.

In geometry, a k-simplex is defined as the non-degenerate convex hull of $k+1$ geometrically distinct points, $v_0, v_1, \ldots, v_k \in \mathbb{R}^d$ where $k \leq d$ [23]. Points, edges, triangles and tetrahedra are all examples of simplices. In particular, a point is 0-simplex, an edge is 1-simplex, a triangle is 2-simplex and a tetrahedron is 3-simplex. A collection of simplices forms a simplicial complex, which is formally defined as follows:

Definition 1. Simplicial complex [24]:
A simplicial complex, $\mathcal{K}$ is a set containing finitely many simplices that satisfies the following two restrictions:

- $\mathcal{K}$ contains every face of every simplex in $\mathcal{K}$;
- For any two simplices, $\sigma, \tau \in \mathcal{K}$, their intersection $\sigma \cap \tau$ is either empty or a common face of σ and τ.

A simplicial k-complex, $\mathcal{K}_k$ is a simplicial complex where the largest dimension of any simplex is equal to k. A two dimensional Delaunay triangulation is an example of simplicial 2-complex.

Line segments which do not belong to any triangle in a $\mathcal{K}_2$ are either bridges, dangling edges or disconnected line segments (see Fig. 3(a)). In a simplicial 2-complex, if one or more triangles are attached to any other k-simplex (where $k = 1$ or 2) through only one of its vertices, then that vertex is termed as a junction point, i.e., the triangle(s) is(are) free to oscillate about its junction point (see Fig. 3(a)).

Definition 2. Regular simplicial 2-complex ($\mathcal{RK}_2$):
A simplicial 2-complex $\mathcal{K}_2$ is said to be regular if it satisfies the following conditions:

- All the points in $\mathcal{K}_2$ are pairwise connected by a path on the edges.
- It does not contain any junction points, dangling edges or bridges.

A detailed explanation on simplices and simplicial complexes can be found in [24]. An edge in $\mathcal{RK}_2$ is a boundary edge (red colored edges in Fig. 3(b)) if it is incident to a single triangle.

Definition 3. Boundary triangle:
A triangle in $\mathcal{RK}_2$ is a boundary triangle if it is incident to at least one boundary edge. In Fig. 3(b), all triangles having red edges are boundary triangles.

The triangles in $\mathcal{RK}_2$, which share all the three edges with other triangles are termed as interior triangles. The triangle with blue edges in Fig. 3(b) is an example of an interior triangle. An obtuse triangle in $\mathcal{RK}_2$ is referred to as a ***thin triangle***. It is to be noted that a thin triangle has always its circumcenter lying strictly external to it. All other triangles are referred to as ***fat triangles***. The longest edge of a thin triangle is called as a ***characteristic edge***.

Let $B(p, q, R)$ be a ball passing through the points p and q with a radius R. An edge (p, q) where $p, q \in S$ is referred to as a Gabriel edge iff $B(p, q, \frac{d(p,q)}{2})$ is empty. All possible edges of S except Gabriel edges are called as non-Gabriel edges. A relaxed Gabriel graph ($RGG(S)$), is a regular simplicial 2-complex that consists of most of the Gabriel edges and a few non-Gabriel edges inherited from the $Del(S)$. $RGG(S)$ retains a non-Gabriel edge $(p, q) \in Del(S)$ if it satisfies either of the following:

- Circumcenter of the Delaunay triangle $\triangle pqr$ for which (p, q) is the characteristic edge, lies internal to $\partial RGG(S)$.
- Removal of (p, q) violates regularity in $RGG(S)$.

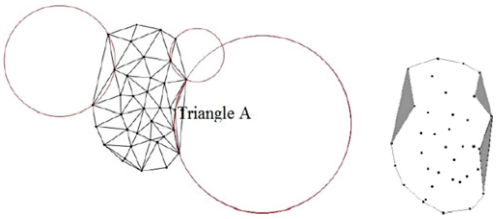
3.2. Sampling criteria

In this section, we assume that the point set S has been sampled from a polygonal object O and we call S, an object sample. All the subsequent terminology introduced are with respect to the original object O. We characterize point sets based on a sampling condition and the boundary alignment of O. Let C be the set of all open connected regions of $conv(O) \setminus \partial O$. Each polygon given by the closure $\overline{C}$, is defined as a cavity [25]. The edges of ∂O in each cavity are called cavity edges, and the edges of $conv(O)$ in the cavities are called ***virtual edges***. Every cavity has exactly one virtual edge covering it (Fig. 4(a)) and a polygonal object can have multiple cavities.

Let B be a set of points sampled from ∂O and $Del(B)$ denotes its Delaunay Graph. An **external Delaunay triangle** is a triangle, $\triangle_x \in Del(B)$ which is partially or fully exposed to the exterior of ∂O as shown in Fig. 4(c), and it exists only in the cavities.

Definition 4. Directed boundary sample:
A boundary sample B is said to be directed, if every external Delaunay triangle in $Del(B)$ is obtuse and its longest edge faces towards the virtual edge of the corresponding cavity.

$(r, \uparrow)$-***sample***. In Definition 4, we considered only the boundary sample. Now we extend the definition to object samples with some additional sampling criteria. Sampling such as ϵ-sampling [12], defined in terms of the medial axis of the object, enables providing theoretical guarantees for curve or surface reconstruction. ϵ-sampling to object samples seems to be inappropriate, as the



(a) Delaunay triangulation. (b) Convex hull with irrelevant region in gray color.

Fig. 5. An example showing the irrelevant space in defining the shape of a set of points. In Fig. 5(a), triangle A constitutes the irrelevant region as quantified by its circumcircle.

points are distributed all over the object. So, we adopt a model similar to the one used in [20] with few additional constraints (Definition 5).

Definition 5. $(r, \uparrow)$-sample:

A point set S sampled from a geometric object O is said to be $(r, \uparrow)$-sample if it satisfies the following properties.

1. Object O possesses a directed boundary sample.
2. Each pair of adjacent boundary samples, $p, q \in \partial S$ lies at a distance of at most $2r$, i.e. $d(p, q) \leq 2r$.
3. Each pair of samples p, q where $p \in int(S)$ and $q \in S$ lies at a minimum distance of $2r$, i.e. $d(p, q) \geq 2r$.

4. The algorithm

In computational geometry, sculpting is the process of creating shapes through repeated elimination of triangles from an initial triangular mesh. The proposed algorithm is analogous to Boissonat's sculpture [26] method but uses a different sculpting strategy. As opposed to a selection criteria based on the maximum distance in sculpture, we use the combination of circumcenter and circumradius of Delaunay triangle to capture the structure of triangles and consequently the geometric proximity of input points.

In most of the cases, an exterior circumcenter (the circumcenter of a Delaunay triangle that lies outside the boundary of the triangulation) produces a wider angle at the vertex opposite to the boundary edge of the triangle, which in turn pushes the other two boundary vertices further away from each other and makes them non-neighbors on the boundary. In Fig. 5, the triangle A is removable as it is irrelevant with respect to the shape of the input points. Apart from the wider angle property, there are two factors that establish triangle A as removable: First, its emptiness due to the circumcircle property of Delaunay triangulation and second, its existence in the proximity to the boundary. In our approach, we basically search for such irrelevant boundary Delaunay triangles and remove them to construct the geometric shape of the input points.

4.1. Triangle filtration

The main algorithm starts by constructing the Delaunay triangulation of the input point set S and uses the filtration procedure along with a hole detection method to reach the final shape. Input to the filtration algorithm (Algorithm 1) is a regular simplicial complex, $\mathcal{RK}_2(S)$ and it generates a filtered regular simplicial complex, denoted by $F(S)$. $F(S)$ is initialized to $\mathcal{RK}_2(S)$. Then, the filtration algorithm proceeds by iteratively removing all thin boundary triangles, $\triangle T$, if it satisfies the CIRCUMCENTER and REGULARITY constraints.

Algorithm 1: TRIANGLE_FILTER($\mathcal{RK}_2(S)$)

Input: Regular simplicial complex $\mathcal{RK}_2(S)$
Output: Filtered regular simplicial complex $F(S)$
Initialize $F(S)$=$\mathcal{RK}_2(S)$;
Construct the heap PQ, containing **deletable** boundary triangles of $F(S)$, sorted in the descending order of their circumradii;
while *PQ is not empty* **do**
  T=root(PQ), delete T from PQ;
  if T *is* ***deletable*** *&* *circumcenter*(T) *lies outside* $\partial_i F(S)$ *&* *characteristic edge of* $T \in \partial_i F(S)$ **then**
    Delete T from $F(S)$;
    Add the **deletable** neighbors of T to PQ;
  end
end
return *Filtered regular simplicial complex,* $F(S)$

- CIRCUMCENTER PROPERTY:- Circumcenter of $\triangle T$ lies external to the intermediate boundary of $F(S)$, denoted by $\partial_i F(S)$ and the characteristic edge of $\triangle T$ is a boundary edge of $\partial_i F(S)$.
- REGULARITY PROPERTY:- $\partial_i F(S)$ is regular even if the $\triangle T$ is removed from it. A triangle $\triangle T$ is considered as **deletable** if it satisfies the REGULARITY constraint.

Regularity of the intermediate $F(S)$ is ensured by checking for dangling edges or junction points on thin triangle removal in constant time.

There may arise situations where two thin boundary triangles with circumcenter lying exterior to ∂RGG_i share a common edge as highlighted in blue color in Fig. 6(b). If one triangle gets removed, the other one is retained in the intermediate RGG_i due to the regularity constraint. Consequently, the triangle, that is selected first is removed, while the other one is retained in the final shape. Hence, the final shape is completely dependent on the order in which the algorithm selects these triangles. In our approach, the triangles are selected for removal according to their circumradius. Deletable boundary thin triangle with the largest circumradius gets removed first. To accomplish this, we use a heap based priority queue as described in Algorithm 1. The reason for choosing circumradius as the selection criterion is that, when combined with the location of the circumcenter, circumradius effectively captures the geometric proximity of points.

4.2. Structure of holes

Broadly, we classify the possible holes in a point set as fat and thin. A fat hole (Fig. 7(b)) is more compact and consists of a set of fat triangles surrounded by thin triangles whereas a thin hole (Fig. 7(d)) is elongated, more like a tubular structure that consists of a series of thin and fat triangles. From our observation, for the fat holes, we could draw a common structure in terms of Delaunay triangles. For example, a closer look at the structure of Delaunay triangulation around the hole in the teapot (Fig. 8(a)) shows a fat triangle surrounded by thin triangles. Similar structure of Delaunay triangulation can be observed around the hole region in other objects (Fig. 8(b) and (c)) as well. However, it is extremely difficult to characterize thin holes by a common Delaunay structure (see Fig. 7(d)). Hence, we account for fat holes in this paper and hereafter we use the term hole to refer to fat holes.

In our approach, we conceptualize a hole as a structure with a body surrounded by a set of arms. A **body** is a set of connected fat triangles and an **arm** is a set of connected thin triangles attached to a fat triangle in the body. Holes can be categorized based on the number of arms it possesses.

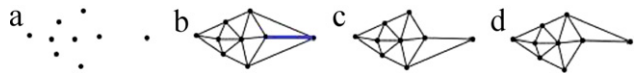


Fig. 6. Illustration on how the order of triangle removal leads to different shapes for certain point sets. (a) Point set, S, (b) $Del(S)$, (c) $RGG1$, (d) $RGG2$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

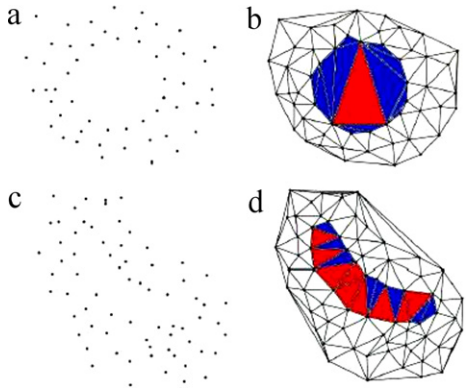


Fig. 7. Illustration of fat and thin hole structures: (a) Point set, S having fat hole, (b) $Del(S)$ showing fat triangles(red shaded) surrounded by thin triangles (blue shaded), (c) Point set, P having elongated hole, (d) $Del(P)$ showing Delaunay structure in the elongated hole region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

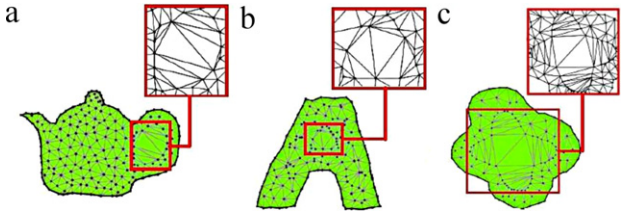


Fig. 8. Structure of Delaunay triangulation in the hole regions of various point sets. External boundary of each point set has been shown to give a feel of the final shape.

Definition 6. Fat hole:

A fat hole is defined as a set of $m-2$ connected fat triangles (called body, B) surrounded by m arms ($A_{1..m}$) where $m \geq 3$, such that:

- Fat triangles in the **body** are connected in a linear fashion, i.e. each fat triangle in a body is connected to at least one **arm**, A_i.
- An **arm** consists of a set of thin triangles.
- An **arm**, A_i is attached to the **body** B via the characteristic edge of a thin triangle, $T \in A_i$.
- Each internal edge of an **arm** A_i is a characteristic edge of a thin triangle $T_j \in A_i$ further away from the **body**, B.

Intuitively, our characterization of the fat hole works because of the fact that, each arm can be imagined to create a cavity from somewhere 'inside' the point set. As the hole boundary has to be a connected one, the virtual edges of each cavities are also connected. As each virtual edge is connected to a triangle, the set of connected virtual edges happen to form fat triangles in the void region. The boundary of the hole is then detected from the body consisting of fat triangles and further expanded by surrounding arms, thus creating a relatively larger empty region in the point set forming a hole. The concept of a hole is illustrated in Fig. 10(a). All the **arms** (Fig. 10(a)) are boundary aligned towards the **body**. Once the body is deleted and its surrounding triangles are labeled as boundary triangles, arm boundaries can be constructed through the *TRIANGLE_FILTER*() (Algorithm 1). In Fig. 10(b), the structure consists of a set of m fat triangles surrounded by m thin triangles. This violates Definition 6 and as a result, a point gets trapped inside the structure thereby invalidating it as a hole. In Fig. 10(c), fat triangles are not connected in a linear fashion and hence violates Definition 6.

4.3. RGG generation

Algorithm 2: RGG_GENERATE(S)

```
   Input: S
   Output: Relaxed Gabriel graph of S, RGG(S)
1  Initialize F(S)=Del(S);
2  F(S)=TRIANGLE_FILTER(F(S)) ;
3  Create a list L of fat triangles sorted in the descending order
   of their circumradius;
4  foreach fat triangle, △T ∈ L do
5      B=BODY_CHECK(△T, 0, -1, id(△T)) (refer Appendix 7) ;
6      if B does not contains a NULL value & deletion of B leaves
       F_i(S) regular then
7          Delete body B from F_i(S);
8          Update the neighboring triangles of B as boundary
           triangles;
9          Delete from L, all the fat triangles present in B;
10     end
11     else
12         Delete from L, all the fat triangles present in B;
13     end
14 end
15 RGG(S)=TRIANGLE_FILTER(F_i(S));
16 return Relaxed Gabriel graph, RGG(S)
```

External boundary of the point set is extracted through the filtration algorithm applied to the $Del(S)$ (line 2 of Algorithm 2). The process of external boundary extraction is illustrated in Fig. 9(a)–(e).

The hole detection part of the algorithm (Lines 3–15 of Algorithm 2), first creates a sorted list L of all fat triangles present in $F(S)$. Fat triangles in L are non-increasingly sorted based on their circumradius. Then, for each fat triangle $\triangle T$ in L, it checks whether $\triangle T$ is a member of any hole structure (body and arms) or not. This check is done based on the presence of a NULL value in B returned by BODY_CHECK() algorithm (described in Appendix 7). If NULL value is not present, then $\triangle T$ is a member of a hole in the intermediate $F(S)$ and hence the entire hole body B (all fat triangles in B) is deleted from L. Further, B is also deleted from the $F_i(S)$ subjected to the REGULARITY property. Neighboring thin triangles of each deleted fat triangle are updated as boundary triangles. Further, it calls TRIANGLE_FILTER() algorithm which deletes all thin triangles present in the arms and completes the hole boundary construction.

Deletion of fat triangles in B from L ensures a linear time operation for the whole process of hole body identification and deletion (for loop of RGG_GENERATE() algorithm). Fig. 9(f)–(g) shows various stages of hole detection for teapot shape. Deleted body of the shape is shown in Fig. 11(a) and (c) and its respective hole after deleting arms is shown in Fig. 11(b) and (d). RGG_GENERATE() algorithm incurs a time complexity of $O(n \log n)$ mainly contributed by the Delaunay triangulation construction, TRIANGLE_FILTER() sub routine and fat triangle sorting.

4.4. Theoretical analysis

In this section, we analyze the topological correctness of $RGG(S)$ under a few assumptions on the input objects and point sets. In

Fig. 9. Different stages of the RGG_GENERATE() algorithm. (a) Point set, (b) Delaunay triangulation, (c) and (d) removing thin triangles iteratively, (e) RGG_i after the body removal and (f) Final *RGG*.

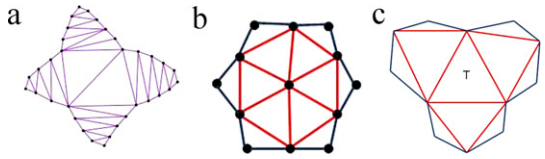


Fig. 10. Illustration of hole and non-hole structures. Fig. 10(a) shows a 4-arm hole. In Fig. 10(b) and (c), triangles with red colored edges indicate fat triangles and triangles with blue color edges constitute thin triangles. Fig. 10(b) is an example of non-hole where m fat triangles are surrounded by m arms. Fig. 10(c) shows another example of non-hole structure where the fat triangles do not follow a linear connection. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

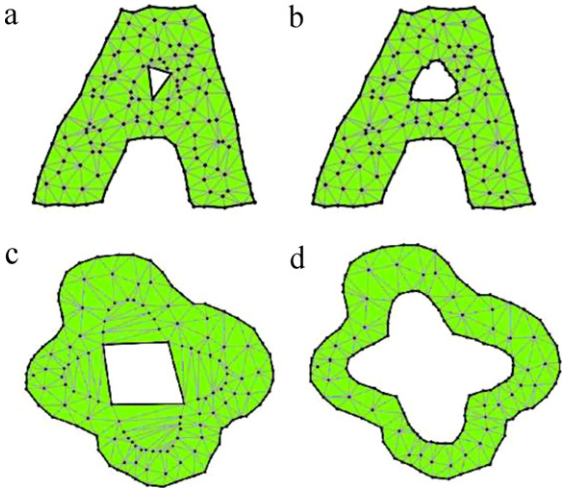


Fig. 11. Illustration of **body** and **arm** removal.

particular, we discuss about the false alarms (false positive holes), external boundary (∂_x) and internal boundaries (∂_h). The object boundary ∂O is the union of external and internal boundaries, i.e. $\partial O = \partial_x \cup \partial_h$.

Guarantee on the external boundary.

Lemma 4.1. *Let S be a $(r, \uparrow)$-sample of an object O, $\partial RGG(S)$ contains an edge between every pair of adjacent samples of $\partial_x O$.*

Proof. We need to show two points here.

1. RGG_GENERATE() algorithm eliminates all the external Delaunay triangles in $Del(S)$ to reach $\partial_x O$.
2. Edges between every pair of the adjacent samples of $\partial_x O$ are never deleted.

First one directly follows from Definition 4, as Line 2 of Algorithm 2 eliminates all triangles satisfying Definition 4 iteratively and gradually approaches towards the original boundary $\partial_x O$.

To show the second point, consider two adjacent boundary samples $p, q \in \partial_x O$. Since S is $(r, \uparrow)$-sampled, the distance between p and q is less than or equal to $2r$, i.e. $d(p, q) \leq 2r$. The distance from both, p and q to another interior point s, in r-sampling is $\geq 2r$. So the triangle formed by p,q and s is either a fat triangle (equilateral triangle if $d(p, s) == d(q, s) == 2r$) or a thin triangle whose circumcenter lies inside $\partial RGG(S)$. Hence, once the RGG_GENERATE() hits the actual boundary samples, then it never deletes any boundary triangles further and hence the *RGG* contains an edge between every pair of adjacent boundary samples of O.

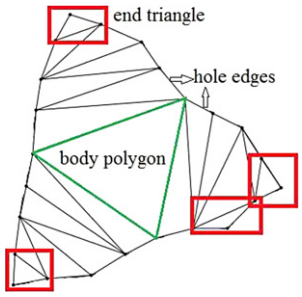


Fig. 12. A 3-arm hole showing end triangles (in red boxes), hole edges and the body polygon (green colored). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Correctness of hole detection. In m-arm holes, the boundary of the **body** represents a polygon (referred to as **body polygon**) having m vertices and two adjacent **arms** are joined at one of the m vertices of the **body polygon** (see Fig. 12). An **end triangle** of an **arm** is its member triangle which has two edges lying on the hole boundary. It is to be noted that an **arm** can have more than one **end triangles**. Except the **end triangles**, all other member triangles of an **arm** of a hole contribute only a single edge, referred to as a **hole edge**.

Lemma 4.2. *m-arm fat hole structure induces $(r, \uparrow)$-sampling along the $\partial_i O$.*

Proof. Without loss of generality, we consider a single fat hole, H where $\partial H \in \partial_h O$. Since, in the Delaunay triangulation of S, **body polygon** exists only in the middle of the hole, it does not have a role in characterizing the sampling of ∂H. Hence, ∂H sampling solely depends on its **arms** structure. An arm consisting of member triangles with the **hole edges** having a length of r satisfies the conditions 2 and 3 in Definition 5 along a sub portion of ∂H. If all the arms are of this kind, then the whole ∂H becomes $(r, \uparrow)$-sampled.

Lemma 4.3. *Let S be a $(r, \uparrow)$-sample of an object O, $\partial RGG(S)$ contains an edge between every pair of adjacent samples of $\partial_h O$.*

Proof. A reverse argument of Lemma 4.2 holds good here due to the assumption that the object O contains only fat holes. This implies that body-arm structures are generated in the hole regions of $Del(S)$. Consequently, RGG_GENERATE() constructs the hole boundaries for all the body-arm structures and hence the proof.

False alarms. A false positive hole arises when the Delaunay triangles in an empty region $\mathfrak{E}$ of S, where S sampled from O, conforms to the hole definition (Definition 6), when $\mathfrak{E}$ is actually not a hole in the object O. We consider a dense $(r, \uparrow)$-sampling where the closest neighboring points of each point lie at a distance of exactly $2r$. Due to the sampling distance constraint, all Delaunay triangles of $Del(S)$ lying in the interior of the object O are equilateral. Since the $Del(S)$ in the interior of O never contains thin Delaunay triangles, any possibility of body and arm structures in the interior of the object is eliminated.

We summarize the results of Lemmas 4.1–4.3, in Lemma 4.4.

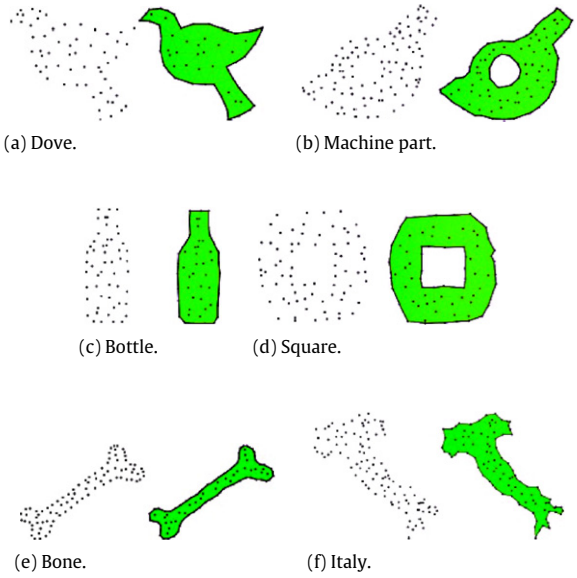


Fig. 13. Shapes generated for point sets with low density. All are randomized point distributions.

Lemma 4.4 (*Topological Correctness*)**.** *Let S be a $(r, \uparrow)$-sample of an object O (possibly with fat holes), $\partial RGG(S)$ contains an edge between every pair of adjacent samples of ∂O and $RGG(S)$ is free of false positive holes.*

5. Experimental results and discussion

The proposed algorithm is implemented in C++ using computational geometry algorithms library (CGAL) [11]. A user-friendly GUI that inherits the Delaunay triangulation GUI of the Triangulation_2 package [11] has also been developed.

We analyzed the effectiveness of our shape approximation algorithm by testing on a variety of 2D point sets of different sampling densities and distributions. Since there does not exist a mathematical definition for what constitutes the optimal shape of a set of points, we use original shapes [10] to compare the reconstructed results. Some of the test data are sampled from the images taken from AIM@SHAPE repository [27]. For the experiments, we used semi-randomized and randomized point distributions which are generated with the help of χ-shape software [10]. In a semi-randomly distributed point set, each point is randomly positioned at a distance greater than a threshold distance from any other point. We experimented with different point sets belonging to different categories which consist of samples from both simply connected and multiply connected objects.

5.1. Uniform and non-uniform point sets

Fig. 13 shows the results obtained for various randomly distributed point (non-uniform) sets with low densities. Fig. 14 shows the results generated by our algorithm for highly dense and evenly distributed point sets sampled from different images. Figs. 13 and 14 reassure our claim that RGG_GENERATE() algorithm can be used to construct boundaries for a wide range of hole structures including multiple hole boundaries of arbitrarily different shapes and sizes, as well as symmetric holes like circle and square. Considering the fact that no optimal algorithm is known till date which account for all possible shapes available, our attempt for characterizing the hole boundaries in shape reconstruction is quite relevant.

5.2. Comparative study with other methods

In this section, we present a comparative study of our shape reconstruction algorithm with two other prominent shape reconstruction approaches. The first approach is α-shape [7] and the second one is χ-shape [10]. We would like to remark that, though α-shape is perceived as the generalization of the convex hull and general theory about the shape of point sets, it has also been employed for shape reconstruction in [28]. All the three approaches make use of Delaunay triangulation filtering for the shape characterization. While α-shape and χ-shape remove the edges of the Delaunay triangulation, our approach removes thin boundary triangles satisfying CIRCUMCENTER and REGULARITY properties mentioned in Section 4.1.

Fig. 15 illustrates a qualitative comparison of our algorithm with α-shape and χ-shape algorithms. At the end of this section, we also briefly compare our approach with two variants of α-shape-conformal and LDA-α-shapes. We compare our method to the above mentioned approaches based on the following criteria.

External parameter and theoretical guarantee. Both α-shape and χ-shape are dependent on an external parameters (α for α-shape and *normalized length* for χ-shape) and hence generates a set of shapes for a given point set. It is extremely difficult to select the best α (correspondingly, *normalized length*) for a non-uniform point set. Moreover, neither, α-shape nor χ-shape provides a theoretical guarantee on the reconstructed output. In contrast, our approach does not rely on any external parameter and hence generates only one shape for a given point set. Theoretically, it has been shown that, for point sets $(r, \uparrow)$-samples, ∂RGG captures a piece-wise linear approximation of the original object boundary (both internal and external boundaries).

Hole detection. α-shape and our method detect holes in the point set whereas χ-shape lacks this ability. Our method found to perform better than α-shape for fat holes as illustrated in Fig. 16(a)–(c) (also see A-shape point set of Fig. 15). However, as already mentioned in Section 4.2, ∂RGG (Fig. 17(b)) cannot detect the thin holes (such as Fig. 17(a)) whereas α-shape detects the hole (Fig. 17(c)).

Noise and disconnected regions. All the three approaches include the noise as a part of the final shape, unless the point set is preprocessed for them. If the input point set consists of disconnected regions, only α-shape has the ability to generate multiple regions for a well chosen value of α. A preprocessing step [10] such as employing a clustering algorithm like k-means can be used in other

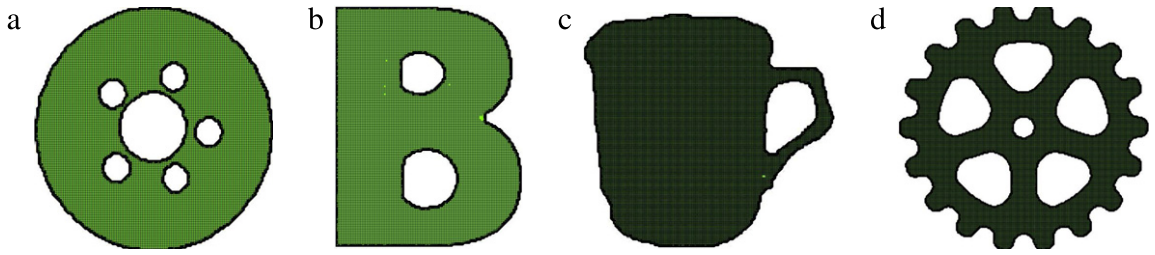


Fig. 14. Reconstruction results for point sets with high densities and uniform distribution. (a) Disc brake rotor, (b) Letter B, (c) Cup, (d) Gear.

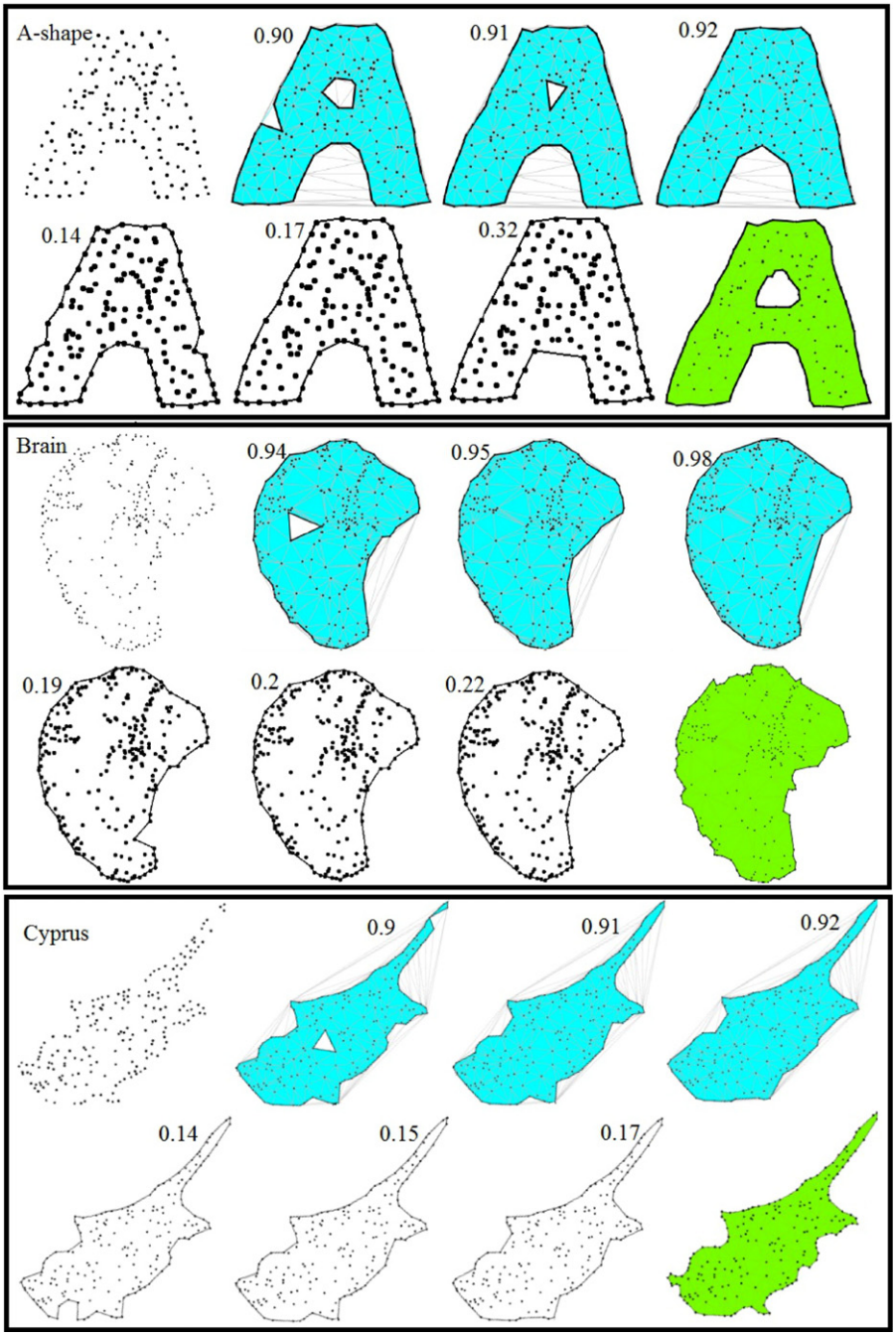


Fig. 15. Qualitative comparison of α-shapes (in cyan) and χ-shapes (in gray boundary) (with their parameters in the top left in each figure) with *RGG* (in green) for various non-uniform point sets. It can be observed that the middle shapes for each family is visually optimal. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

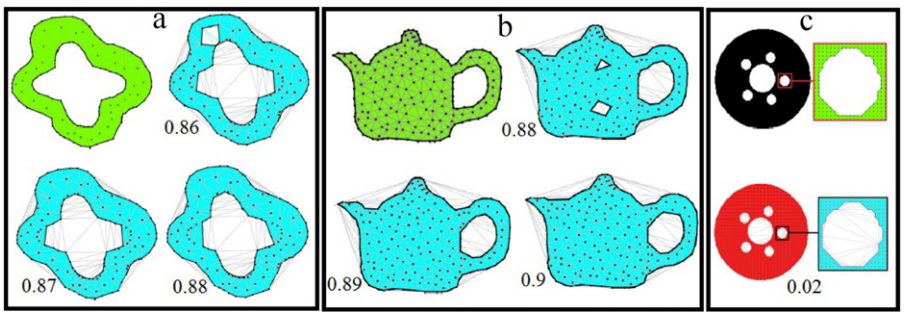
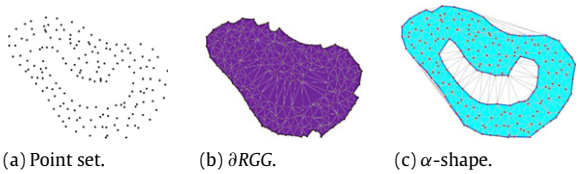


Fig. 16. Qualitative comparison of hole boundaries of *RGG* (in green) and α-shape (in cyan, with parameter value indicated in the left bottom of each figure). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Point set. (b) ∂RGG. (c) α-shape.

Fig. 17. Limitation in hole detection.

approaches to cluster the point set first and then approximate the shape of each region.

Regularity. α-shape can be irregular as it may contain bridges, dangling edges or junction points. χ-shape and ∂RGG are always regular as both the algorithms check for the regularity constraint before filtering the Delaunay triangulation.

Time complexity. All the three algorithms run in $O(n \log n)$ as $Del(S)$ computation takes $O(n \log n)$ time.

Since α-shape relies on a global scale parameter, shape reconstruction based on α-shape is successful only for uniform sampling [29]. To cure this limitation, variants such as LDA-α-shape [16] and conformal α-shape [29] have been proposed. These algorithms utilize the local density and geometric information available in the point set quantified by a maximally finite ball (LDA-α-ball) and local scale parameter (conformal α-shape) for the filtration of Delaunay complex. LDA-α-complex requires high density sampling in the critical regions of the shape as the algorithm completely relies on the information of local density variation. As opposed to this, we use a criterion, which effectively combines a local measure (circumcenter of Delaunay triangles) and a global measure (removal based on circumradius of Delaunay triangles) to filter Delaunay complex. For detecting holes, we use the structural arrangements of Delaunay triangles and hence our approach does not rely on any external parameter. Conformal α-shape uses a local scale parameter $\widehat{\alpha}$ and also provide geometric and topological guarantees using ϵ-sampling model. Hence it mainly focuses on curve and surface reconstruction. LDA-α shape uses a different sampling called as (ϵ, δ)-sample to provide theoretical guarantees. In our approach, we use a semi-random sampling called as $(r, \uparrow)$-sample to establish that ∂RGG of an object O represents a piecewise linear approximation of ∂O.

5.3. Effects of point density

To validate our algorithm, we conducted experiments on point distributions of well defined shapes and computed to what extent ∂RGGs of these point sets conform to the respective original shapes. We used L^2*error norm* (used for the evaluation of χ-shape algorithm in [10]) for the performance evaluation of our shape reconstruction algorithm. It is defined as the area of the symmetric difference between the original shape O and the $\partial RGG(S)$ as a proportion of the total area of the original shape O. Let $Area(O)$ and $Area(S)$ denote the area of the original shape and the area of ∂RGG respectively, then L^2*error norm* is computed as follows:

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

An L^2*error norm* of zero implies that the two shapes are equal in area and also their boundaries are structurally alike. The units of the area in all the cases are square pixels.

We used randomly distributed points of Germany, Vietnam, F and K shapes for the point density experiment because they exhibit a range of sinuosity and non-convexities. For the Germany point set, it is apparent from Fig. 18 that all the three shapes tend to the original shape by capturing fine details of the boundary as the point set density increases.

Fig. 19 exhibits the variation in the accuracy of all the three shapes (∂RGG, χ-shape and α-shape) with the varying point set densities as quantified by L^2*error norm*. From Fig. 19(a)–(b), it can be observed that as the point set density increases, accuracy of each shape also increases. Our algorithm performs relatively better as the point sets density increases for Germany and Vietnam shapes.

It is to be noted that, F and K shape boundaries do not possess different levels of sinuosity. Instead, they have non-convex regions on their boundaries. A direct proportionality relation between shape accuracy and point set density is not much evident from

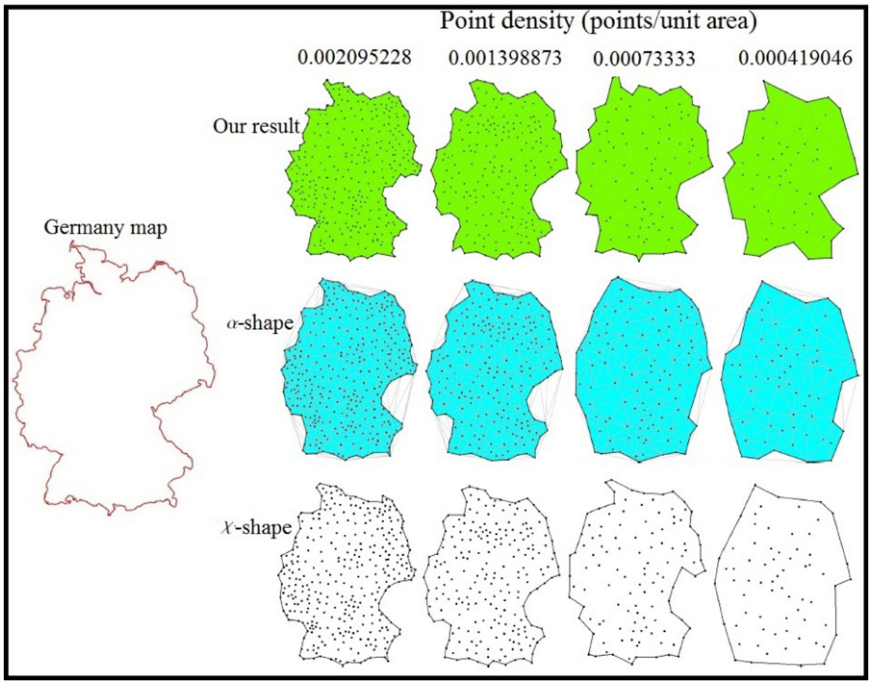
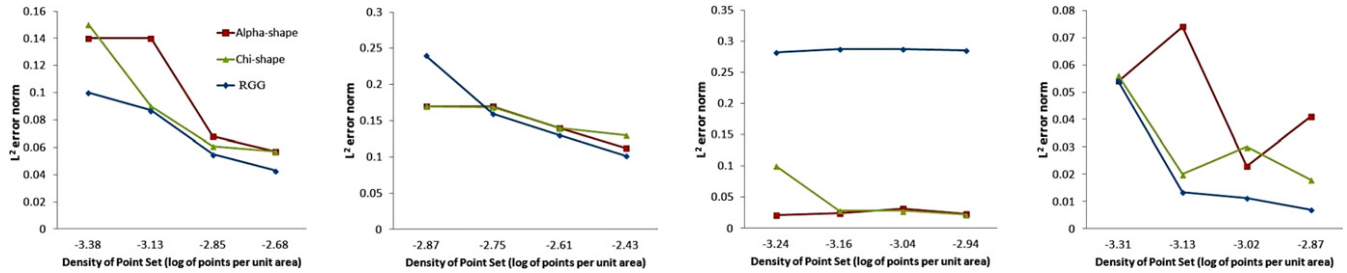
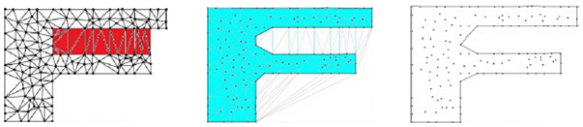


Fig. 18. A qualitative comparison of the shape reconstruction algorithms for Germany point set with varying densities. All the point distributions used for the experiment are randomly distributed. Area of original shape is computed using χ-shape software [10].



(a) Germany. (b) Vietnam. (c) F shape. (d) K shape.

Fig. 19. Variation in the accuracy of shapes with the changing density of various point sets.



(a) Our result. (b) α-shape. (c) χ-shape.

Fig. 20. Shapes generated for F point set with density −2.94 (log of points per unit area). The high lighted red triangles are the obstructing triangles in ∂RGG. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 19(c)–(d). For F-shape, χ-shape and α-shape algorithms show a better performance than the ∂RGG (please refer Figs. 19(c) and 20). In addition to that, ∂RGG results do not improve much for F shape (Fig. 19(c)). This is mainly due to the presence of obstructing triangles (either fat triangles or thin triangles whose characteristic edges are not on the boundary, triangles high lighted in red color in Fig. 20) in the upper non-convex portion of the F shape. RGG_GENERATE() (Algorithm 2) can never filter such triangles and the cavity remains undetected (please refer Fig. 20) irrespective of an increase in the point set density. The boundary of K shape never gave rise to obstructing Delaunay triangles and hence our algorithm generates a more accurate K shape compared to the other two algorithms (refer Fig. 19(d)).

5.4. Effects of point distribution

To investigate the effects of point distribution on the shape reconstruction methods such as α-shape, χ-shape and the proposed algorithm, we conducted an experiment on the shape of the upper case letter K generated using sanserif font (Arial). We considered four different point distributions of letter K—a semi-randomly distributed points all over the shape (SR-semi-random), a semi-randomly distributed point set where points are concentrated in the interior of the shape (SRIB-semi-random internally biased), a semi-randomly distributed points where points are concentrated towards boundary (SRBB-semi-random boundary biased) and a randomly distributed point set (TR-truly random). We conducted experiment on all the three shape reconstruction algorithms (α-shape, χ-shape and ∂RGG) using an instance of each point distribution category. Fig. 21 illustrates the results generated by each algorithm for different distributions. From Fig. 21, it is evident that as the inhomogeneity in the point distribution increases, the deviation between the results by each algorithm and the original shape also increases. To be more precise, ∂RGG (similarly, α-shape and χ-shape) of semi-random distribution is structurally closer to the original shape than the ∂RGG of a truly random distribution. This was an expected behavior because a truly random distribution of points may lack several features and information of the original shape thereby making the reconstruction process extremely difficult.

Fig. 22 captures a more accurate and precise summary of the point distribution experiment as quantified by L^2*error norm*. We

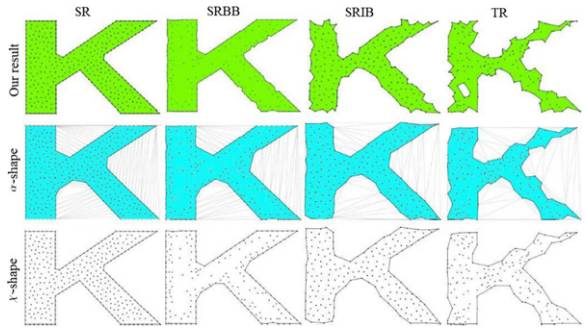


Fig. 21. Results generated by shape reconstruction algorithms for various types of point distributions of K-shape. In this Figure, SR refers to semi-random, SRBB refers to semi-random internally biased, SRIB refers to semi-random boundary biased and TR refers to truly random distributions.

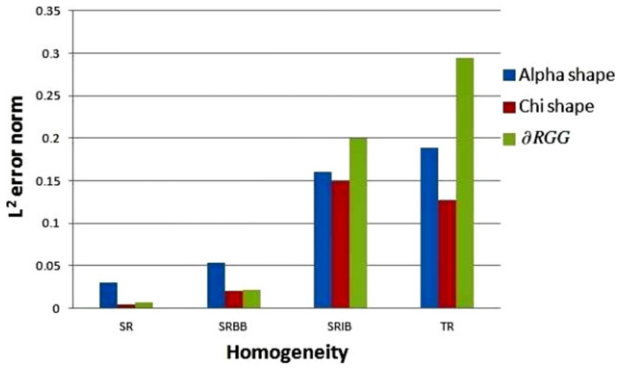


Fig. 22. Variation in the accuracy of shapes with the changing homogeneity in the K-shape point sets.

would like to point out that letter K is a representative shape for a wide variety of shapes having non-convexities on their boundary and hence we believe that, the experiment has the potential to capture the behavior of all the three algorithms on different distributions of such shapes. For SR distribution, all the three shape reconstruction algorithms construct boundaries structurally very close to the original boundary as indicated by the histograms in Fig. 22. When comparing results for SRBB and SRIB distributions, it can be observed that all the three algorithms perform better in the case of SRBB distribution. This is mainly due to the presence of more boundary points capable of defining the boundary of the shape in the case of SRBB.

For a TR distribution, α-shape and χ-shape performed better than ∂RGG (Fig. 22). In particular, χ-shape was found to be more tolerant to the random distribution than the other two. Further we noticed that randomly distributed points can have large gaps which may constitute undesired holes (or false positives) in the ∂RGG as shown in the last entry of the first row of Fig. 21. Some extremely random point distributions may possess void regions

having Delaunay triangulation conforming to the body and arms structures and hence get detected by our algorithm. However, even for an extremely random point distribution carrying much less information about the geometry of the original shape (such as K in Fig. 21), our algorithm is able to generate a prototype structure representing the original shape. Further refinement schemes can be applied on this structure to generate a better quality approximation. From our experiments, we found that the chances of having such unwanted holes in semi-random distributions like the ones shown in the first three columns of Fig. 21 are very low.

6. Conclusion

In this paper, we have proposed an automatic shape reconstruction algorithm for planar point sets. Such a non-parametric algorithm is highly desirable as it relieves the end user of the tedious task of tuning an external parameter to construct the geometric shapes of point sets. The approach is more relevant when considering the fact that most of the established shape reconstruction techniques depend on external parameters to define the shapes but provide limited information on how to pick the right parameter to get an accurate shape.

The proposed algorithm is simple, easy to implement and handles object samples from simply and multiply connected objects. We summarize the key results of our empirical study on the proposed shape reconstruction approach.

1. Like many other shape reconstruction methods, the proposed algorithm is also dependent on the homogeneity in the point distributions. It performs better for semi-random distributions than truly random distributions.
2. Both qualitative and quantitative comparisons with α-shape and χ-shape show that this approach yields boundaries with reasonable accuracy. The quantitative analysis was carried out based on L^2*error norm*.
3. It has been shown that, fat holes that have body and arms structure can be automatically detected. There is no restriction on the number of holes as well.

For genus deduction in three dimensions using the proposed approach, it requires a redesign and modification on the concept of hole and the corresponding algorithms. A direct extension may end up in detecting interior hollow regions such as the cavity inside a coconut shell. However, it is unclear at this point of time, where exactly such cavity detection is applied and hence one of the future work lies in this direction. Apart from this, we are also looking at exploring thin hole formations using Voronoi diagram.

Appendix. Body identification

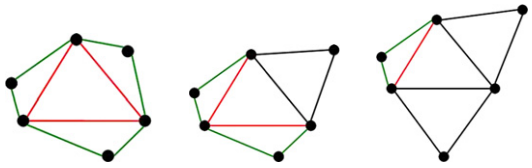
Here we describe the identification of potential bodies in a regular simplicial complex $F(S)$. To accomplish the task, we classify fat triangles in $F(S)$ based on the number of thin neighboring triangles as follows (Fig. 23):

- 3-*thin neighbored*-A fat triangle surrounded by three thin triangles such that the thin triangles share their characteristic edges with the fat triangle.
- 2-*thin neighbored*-A fat triangle whose two edges are also the characteristic edges of two thin triangles and has another fat triangle as its neighbor.
- 1-*thin neighbored*-A fat triangle whose one edge is also the characteristic edge of a thin triangle and has two other fat triangles as neighbors.
- 0-*thin neighbored*-All other fat triangles except 3-*thin neighbored*, 2-*thin neighbored* and 1-*thin neighbored* in PSH are called 0-*thin neighbored* fat triangles.

Algorithm 3: BODY_CHECK($\triangle T$, *depth*, *predecessor_id*, *root_id*)

Input: A fat triangle $\triangle T$, recursion depth *depth*, *predecessor_id* and *root_id*
Output: A list of fat triangles constituting a body, B

```
1  if depth==0 then
2      if △T==3-thin neighbored then
3          B={△T}
4      end
5      else if △T==2-thin neighbored then
6          B={△T, BODY_CHECK(fat_neighbor(△T), 1, id(△T),
           id(△T))}
7      end
8      else if △T==1-thin neighbored then
9          B={△T, BODY_CHECK(fat_neighbor1(△T), 1, id(△T),
           id(△T)), BODY_CHECK(fat_neighbor2(△T), 1, id(△T),
           id(△T))}
10     end
11     else
12         B=[△T, NULL]
13     end
14     return B
15 end
16 else
17     if △T==2-thin neighbored then
18         return △T
19     end
20     else if △T==1-thin neighbored & id(△T) ≠ root_id then
21         Let the neighbor_k is the fat_neighbor of △T whose
           id≠predecessor_id B={△T, BODY_CHECK(neighbor_k,
           depth + 1, id(△T), root_id)} return B
22     end
23     else
24         R = {△T, NULL} return R
25     end
26 end
```



(a) 3-thin neighbored. (b) 2-thin neighbored. (c) 1-thin neighbored.

Fig. 23. Classification of fat triangles based on the number of thin triangles attached to them via their characteristic edges. The characteristic edges are shown in red color. The triangles with green edges constitute thin triangles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The bodies in $F(S)$ are identified using BODY_CHECK() procedure given in Algorithm 3. BODY_CHECK() receives a fat triangle, $\triangle T$ from the RGG_GENERATE() algorithm (Algorithm 2) and finds the body which consists of $\triangle T$. The parameter *depth* indicates the level of recursion and is set to 0 at the start. *Predecessor_id* is the identifier of the predecessor triangle which called the current BODY_CHECK() in the execution hierarchy. Initially, it is set to -1 as the fat triangle provided by the RGG_GENERATE() algorithm does not have any predecessor. The *Predecessor_id* is used to avoid redundant recursive calls in the case of 1-*thin neighbored* triangles. *root_id* is the identifier of the first fat triangle that initiated the BODY_CHECK() procedure and is used to eliminate non-hole structures of the form shown in Fig. 10(b). At any level of the recursion, if the fat triangle encountered is the one which initiated

the body check procedure, it clearly indicates that some points are trapped inside the hole structure. Consequently, the algorithm invalidates such structures as body of any hole.

If $\triangle T$ is a 3-*thin neighbored* triangle, then it must be a body of a 3-arm type hole (Definition 6) and it is returned immediately from Algorithm 3 as *B*. On the other hand, if $\triangle T$ is either 1-*thin neighbored* or 2-*thin neighbored*, it requires further check to confirm that $\triangle T$ is indeed a part of any body in the triangulation. We adopt a recursive strategy to confirm this and the recursion advances in the direction of neighboring fat triangles of $\triangle T$ updating the list of fat triangles encountered during the recursion. At any point of recursion, if a 0-*thin neighbored* triangle is found, it implies that the body definition has been violated and hence the triangle along with a NULL value is returned.

References

[1] Ying-Lie O, Toet A, Foster D, Heijmans HJAM, Meer P, editors. Shape in picture: mathematical description of shape in gray level images. NATO ASI series. Series F: computer and system sciences, vol. 126. Driebergen, The Netherlands: Springer; 1994.

[2] Galton A, Duckham M. What is the region occupied by a set of points? In: Proceedings of the 4th international conference on geographic information science, GIScience'06. Berlin (Heidelberg): Springer-Verlag; 2006. p. 81–98.

[3] Moreira AJC, Santos MY. Concave hull: a k-nearest neighbours approach for the computation of the region occupied by a set of points. In: J. Braz, P.-P. Vázquez, J. M. Pereira (Eds.), GRAPP (GM/R), INSTICC—Institute for Systems and Technologies of Information, Control and Communication. 2007, p. 61–8.

[4] Ogawa H. Labeled point pattern matching by Delaunay triangulation and maximal cliques. Pattern Recognit 1986;19(1):35–40.

[5] Wilson JA, Bender A, Kaya T, Clemons PA. Alpha shapes applied to molecular shape characterization exhibit novel properties compared to established shape descriptors. J Chem Inf Model 2009;49(10):2231–41.

[6] 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.

[7] Edelsbrunner H. Shape reconstruction with Delaunay complex. In: Proceedings of the third Latin American symposium on theoretical informatics, LATIN'98. London (UK): Springer-Verlag; 1998. p. 119–32.

[8] Edelsbrunner H, Kirkpatrick D, Seidel R. On the shape of a set of points in the plane. IEEE Trans Inform Theory 1983;29(4):551–9. http://dx.doi.org/10.1109/TIT.1983.1056714.

[9] Melkemi M, Djebali M. Computing the shape of a planar points set. Pattern Recognit 2000;33(9):1423–36.

[10] Duckham M, Kulik L, Worboys M, Galton A. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. Pattern Recognit 2008;41:3224–36.

[11] CGAL, Computational geometry algorithms, http://www.cgal.org/ (December 2012). URL: http://www.cgal.org/.

[12] Amenta N, Bern M, Eppstein D. The crust and the beta-skeleton: combinatorial curve reconstruction. Graph Models Image Process 1998;60(2):125–35.

[13] Edelsbrunner H. Surface reconstruction by wrapping finite sets in space. In: Discrete and computational geometry: the Goodman–Pollack festschrift. 2003. p. 379–404.

[14] Ramos EA, Sadri B. Geometric and topological guarantees for the wrap reconstruction algorithm. In: Proceedings of the eighteenth annual ACM–SIAM symposium on discrete algorithms, SODA'07. Philadelphia (PA, USA): Society for Industrial and Applied Mathematics; 2007. p. 1086–95.

[15] Krasnoshchekov DN, Polishchuk V, Vihavainen A. Shape approximation using k-order alpha-hulls. In: Proceedings of the 2010 annual symposium on computational geometry, SoCG'10. New York (NY, USA): ACM; 2010. p. 109–10.

[16] Chevallier N, Maillot Y. Boundary of a non-uniform point cloud for reconstruction: extended abstract. In: Proceedings of the 27th annual ACM symposium on computational geometry, SoCG'11. New York (NY, USA): ACM; 2011. p. 510–8.

[17] Toussaint G. A graph-theoretical primal sketch. In: Computational morphology. 1988. p. 229–60.

[18] Kirkpatrick DG, Radke JR. A framework for computational morphology. In: Toussaint GT, editor. Computational geometry. 1985. p. 217–48.

[19] Jaromczyk J, Toussaint G. Relative neighborhood graphs and their relatives. Proc IEEE 1992;80(9):1502–17. http://dx.doi.org/10.1109/5.163414.

[20] Attali D. r-Regular shape reconstruction from unorganized points. Comput Geom 1998;10(4):239–47. Special Issue on Applied Computational Geometry.

[21] Dey TK. Curve and surface reconstruction: algorithms with mathematical analysis. Cambridge monographs on applied and computational mathematics, New York (NY, USA): Cambridge University Press; 2006.

[22] Peethambaran J, Muthuganapathy R. Reconstruction of water-tight surfaces through Delaunay sculpting. Comput-Aided Des January 2015;58:62–72. http://dx.doi.org/10.1016/j.cad.2014.08.021. Solid and Physical Modeling 2014. URL: http://www.sciencedirect.com/science/article/pii/S0010448514001900.

[23] Desbrun M, Kanso E, Tong Y. Discrete differential forms for computational modeling. In: ACM SIGGRAPH 2006 courses, SIGGRAPH'06. New York (NY, USA): ACM; 2006. p. 39–54.

[24] Cheng S-W, Dey TK, Shewchuk JR. Delaunay mesh generation. Chapman and Hall/CRC computer and information science series, CRC Press; 2013.

[25] Mistry S, Niranjan U, Gopi M. Puzzhull: cavity and protrusion hierarchy to fit conformal polygons. Comput-Aided Des 2014;46(0):233–8. 2013 {SIAM} Conference on Geometric and Physical Modeling.

[26] Boissonnat J-D. Geometric structures for three-dimensional shape representation. ACM Trans Graph 1984;3(4):266–86.

[27] Aim@shape, Aim@shape repository, 2013. http://shapes.aim-at-shape.net/ [Online; last accessed 21-03-2013].

[28] Edelsbrunner H, Mücke EP. Three-dimensional alpha shapes. ACM Trans Graph 1994;13(1):43–72.

[29] Cazals F, Giesen J, Pauly M, Zomorodian A. Conformal alpha shapes. In: Proceedings of second symposium on point based graphics, 2005, pp. 55–61.