An Input-independent single pass algorithm for reconstruction from dot patterns and boundary samples

Safeer Babu Thayyil\textsuperscript{a}, Amal Dev Parakkat\textsuperscript{b}, Ramanathan Muthuganapathy\textsuperscript{a,*}

\textsuperscript{a}Advanced Geometric Computing Lab, Department of Engineering Design, Indian Institute of Technology, Madras, India
\textsuperscript{b}Department of Computer Science \\& Engineering, Indian Institute of Technology Guwahati, India

Abstract

Given a set of points \( S \in \mathbb{R}^2 \), reconstruction is a process of identifying the boundary edges that best approximates the set of points. In general, the set of points can either be derived from only the boundaries of the curves (called as boundary sample) or can be derived from both boundary and interior of the curves (called as dot pattern). Most of the existing algorithms focus towards reconstruction from only boundary samples, termed as curve reconstruction. Unfortunately many of them don’t reconstruct when the input is of dot pattern type (called as shape reconstruction). In this paper, we propose an input-independent non-parametric algorithm for reconstruction that works for both dot patterns as well as boundary samples. The algorithm starts with computing the Delaunay triangulation of the given point set. An edge between a pair of triangles is marked for removal when the circumcenters lie on the same side of the edge. Further, we also propose additional criterion for removing edges based on characterizing a triangle by the distance between its circumcenter and incenter. To maintain a manifold output, a degree constraint is employed. The proposed approach requires only a single pass to capture both inner and outer boundaries irrespective of the number of objects/holes. Moreover, the same criterion has been employed for both inner and outer boundary detection. The experiments show that our approach works well for a variety of inputs such as multiple components, multiple holes etc. Extensive comparisons with state-of-the-art methods for various kinds of point sets including varying the sampling density and distribution show that our algorithm is either better or on par with them. Theoretical discussions on the algorithm have also been presented using \( \epsilon \)-sampling and \( r \)-sampling. Limitations of the algorithm are also discussed.

Keywords: Delaunay Triangulation, Shape reconstruction, Curve reconstruction, Unified algorithm, Dot pattern and boundary samples

1. Introduction

Given a set of points \( S \) lying on a plane, sampled from an object, the reconstruction is a task of embodying the boundary edges (inner and outer boundaries) that best approximates its geometrical identity. The problem has been in existence over the last few decades and has remained as an active research area. The problem has shown to be ill-posed Edelsbrunner (1998). Generally, the reconstruction problem is inherently challenging as its results heavily rely on the human perception which in turn made it very difficult for the mathematical formulation of the problem. However, it has found lots of applications in a wide variety of fields such as pattern recognition, image processing, computer vision, computer graphics etc.

In this paper, the point samples are assumed to be derived from a smooth closed curve(s). When the sample points are derived only from the boundaries of the curve(s) (Figure 1(a)), termed as boundary samples, then the reconstruction is generally called as \textit{curve reconstruction} (Figure 1(b)). On the other hand,
sample points, in addition to boundaries, can be acquired from the interior to the curve(s) (Figure 1(c)). This sampling is termed as dot pattern and the corresponding reconstruction is called *shape reconstruction* (Figure 1(d)). In general, our paper aims to reconstruct from point samples of surfaces embedded in $R^2$ (as opposed to surfaces embedded in $R^3$, see for e.g. Dey et al. (2009)).

![Figure 1: (a) Boundary sample (b) Reconstruction from boundary sample (c) Dot pattern (d) Reconstruction from dot pattern.](image)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Unified</th>
<th>Hole</th>
<th># Par.</th>
<th>Mul. Obj.</th>
<th>Un. Hole</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$-shape Edelsbrunner et al. (1983)</td>
<td>Y</td>
<td>Y</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Crust Amenta et al. (1998)</td>
<td>N</td>
<td>Y</td>
<td>0</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>nn-crust Dey and Kumar (1999)</td>
<td>N</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>NA</td>
</tr>
<tr>
<td>$\chi$-shape Duckham et al. (2008)</td>
<td>Y</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>NA</td>
</tr>
<tr>
<td>simple-shape Gheibi et al. (2011)</td>
<td>Y</td>
<td>N</td>
<td>3</td>
<td>N</td>
<td>NA</td>
</tr>
<tr>
<td>deGoes et. al Goes et al. (2011)</td>
<td>N</td>
<td>Y</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>RGG Peethambaran and Muthuganapathy (2015a)</td>
<td>Y</td>
<td>Y</td>
<td>0</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>WDM-crust Peethambaran et al. (2015)</td>
<td>N</td>
<td>N</td>
<td>0</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>ee-shape Methirumangalath et al. (2017)</td>
<td>Y</td>
<td>Y</td>
<td>0</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>HNN-crust Ohrhallinger et al. (2016)</td>
<td>N</td>
<td>Y</td>
<td>0</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Crawl Parakkat and Muthuganapathy (2016)</td>
<td>N</td>
<td>Y</td>
<td>0</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Peel Parakkat et al. (2018)</td>
<td>N</td>
<td>Y</td>
<td>0</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Our Algo</td>
<td>Y</td>
<td>Y</td>
<td>0</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 1: Strengths and weaknesses of different reconstruction algorithms. # Par. - # Parameters, Mul. Obj. - Multiple Objects, Un. Hole - Unstructured Hole, N - No, Y - Yes, NA - Not Applicable.

1.1. Motivation

1.1.1. Input-independent reconstruction

Even though there are a lot of works in the area of reconstruction, many of the works have been focussed towards one type - curve reconstruction from boundary samples. When an algorithm works for both types of input, then it is termed as *Unified algorithm*. Table 1 summarizes the strengths and weaknesses of a few of the reconstruction algorithms. All the algorithms listed in the table can reconstruct boundary samples. However, as can be seen from Table 1, only a few viz. $\alpha$-shape, RGG, ee-shape, $\chi$-shape and simple-shape are unified algorithms. Among them, $\chi$-shape and simple-shape can reconstruct only a simple closed curve as output and cannot handle holes. Though algorithms such as RGG, ee-shape can capture holes, RGG can work only for restricted hole structures and ee-shape uses different strategies for outer and inner boundaries (holes).

$\alpha$-shape is independent of a hole structure but requires a parameter $\alpha$ to be tuned. Other approaches such as $\chi$-shape, simple-shape are also parametric algorithms whereas RGG, ee-shape are non-parametric algorithms. It is quite a tedious task to tune the parameter(s) to get the desired output. RGG, ee-shape, and $\chi$-shape also cannot handle multiple objects (components).
Given an input point set, it is not easy to find out whether it is a boundary sample or dot pattern. To the best of our knowledge, no algorithm exists that can identify the type of the input point set. Considering that, there is a requirement for a unified algorithm (that works independent of the input type) that also uses the same strategy for the detection of inner boundary (hole) as well as outer boundary. Devising such an algorithm that is also non-parametric is an additional challenge.

1.1.2. Applications

An input-independent (unified) reconstruction algorithm can have wide variety of applications in different fields. With the advent of technologies such as GPS and applications in geographical information systems (GIS), sampling in the form of dot pattern has gained prominence. In GIS, reasonable outer boundary of the imprecise region consisting of the given cities are retrieved Arampatzis et al. (2006). Aggregating the set of points representing the separate buildings to form a single polygon is an important task in map generalization Galton and Duckham (2006). One of the challenging issues of 2D parametrization is how to do a boundary mapping. Parametrization is used in texture mapping which in turn is used in computer graphics and image processing Yu (2012). In the field of geometric modeling, there are many problems in 3D involving surfaces, where surfaces are represented as $S(u,v)$, with parameters $u$ and $v$. These problems are analyzed in the underlying parametric $(uv)$-space, which require reconstructed boundaries (for example, please see Sundar et al. (2014)). Harish et al. Ganapathy et al. (2015) showed that reconstruction techniques can be employed to better approximate the regions of interest in a crash optimization problem.

In the field of image processing, particularly in biomedical image processing (with the advent of MRI and CT imaging systems), reconstruction from boundary samples is an important task. Face detection algorithms that recognize facial features such as mouth, nose etc can use reconstruction. In the case of matching of shapes represented as point samples, the reconstructed outer and inner boundaries can be used to define similarity measure.

1.1.3. Theoretical Challenges

Only a very few works in the area of reconstruction address the theoretical challenges such as conditions for exact reconstruction (i.e. up to its polygonal approximation), topological correctness (i.e., correct number of holes are identified) etc. When an algorithm works for reconstruction of both inputs, there is no common sampling strategy that can be used for proving the theoretical challenges. For the problem of curve reconstruction, $\epsilon$-sampling, introduced in Amenta et al. (1998), based on local-feature size (LFS) and medial axis has been used, albeit only in a few research works. For the dot pattern, unfortunately, $\epsilon$-sampling does not work because LFS is not well-defined (because of the presence of interior points) and hence a different sampling has to be used. To address the theoretical challenges for dot pattern, we resort to $r$-sampling used in Methirumangalath et al. (2017).

1.2. Our Contributions:

In this paper, we propose a unified algorithm for the reconstruction of outer boundaries as well as inner boundaries without any user intervention. The following are our major contributions:

- A unified algorithm that is also non-parametric.
- The algorithm uses the same strategy for capturing both hole boundary as well as outer boundary.
- Our algorithm needs only a single pass irrespective of the number of holes/objects.
- We also show that our algorithm is topologically correct.

2. Related Work

Reconstruction algorithms are taxonomised in different ways. It can be based on Delaunay triangulation or non-Delaunay triangulation or curve/shape/unified reconstruction. Some algorithms are designed only
for outer boundary detection while others are designed for both outer boundary as well as inner boundary.

One of the initial works in this field is the introduction of $\alpha$-shape Edelsbrunner et al. (1983). $\alpha$-shape, a Delaunay-based approach, was motivated to generalize convex hull for characterizing a set of points in the plane. M. Melkemi et al. Melkemi and Djebali (2000) introduced the concept of A-shape which is a refined version of the $\alpha$-shape. $\gamma$-shape Duckham et al. (2008) is a non-convex hull to characterize the set of points distributed in the plane. This is a sculpting algorithm which removes the longest exterior edge (greater than a $\text{len}$ parameter) from the triangulation of the input point set subject to the regularity constraint. Gheibi et al. Gheibi et al. (2011) devised a non-Delaunay based unified algorithm known as simple-shape. It starts with an initial shape, the convex hull of the input points, and in subsequent steps, the shape is changed to concave based on some hybrid selection criterion which includes closeness, edge length and angle.

Crust group (crust Amenta et al. (1998), nn-crust Dey and Kumar (1999), cc-crust Dey et al. (2000), and gathanG Dey and Wenger (2001)) algorithms are proximity-based approaches used for curve reconstruction. nn-crust is designed for reconstruction of a closed curve and it can be extended to general $d$ dimension. cc-crust is an improved version of nn-crust to capture the boundary of open curves. gathanG is the next one in this series which modified the sampling condition to handle sharp corners. Stefan Ohrhallinger et al. Ohrhallinger et al. (2016) proposed a variation of nn-crust known as HNN-crust which can reconstruct a curve with a lesser number of samples. To reduce the sampling density, they introduced a new sampling scheme, $p$-sampling, which regulates the gap between geodesically consecutive points.

RGG Peethambaran and Muthuganapathy (2015a) is a non-parametric Delaunay-based sculpting algorithm which produces relaxed Gabriel graph having Gabriel edges and a few non-Gabriel edges. In cc-shape Methirumangalath et al. (2015), the authors proposed a unified algorithm, where they used circle and regularity constraints to decide which edge to be removed. This algorithm, with modification in the starting condition, was introduced in Methirumangalath et al. (2017) for hole detection. Crawl Parakkat and Muthuganapathy (2016), another non-parametric Delaunay-based approach, starts with a seed edge to build a connected subgraph from Delaunay triangulation which is the best approximation of the curve. In Peel Parakkat et al. (2018), the longest edge from every Delaunay triangle is removed until the degree constraint of every vertex is satisfied. WDM-crust Peethambaran et al. (2015) is a Voronoi based labeling approach for curve reconstruction along with medial axis approximation. A few works such as deGoes et al. Goes et al. (2011) focus on the reconstruction from a noisy dataset.

3. Preliminaries

**DEFINITION 1.** Neighboring triangles: Two triangles are said to be neighboring triangles if they share an edge (Figure 2(a)).

**DEFINITION 2.** Coordinated triangles: Neighboring triangles are termed as coordinated triangles if their circumcenters lie on the same side of the shared edge (Figure 2(b)).

Figure 2: (a) Neighboring triangles $\triangle_{abc} \& \triangle_{bdc}$ with shared edge $bc$. (b) Coordinated triangles $\triangle_{abc} \& \triangle_{bde}$ with circumcenters $c_1 \& c_2$ lying on the same side of the shared edge $bc$. (c) Skinny triangle (d) $r$-sampling: $(v,p), (v,s)$ are adjacent points and $(v,u), (v,q)$ are non-adjacent points.
DEFINITION 3. Medial axis: Let C be a curve and S be the point set, the medial axis of C is the locus of centers of all circles which are tangent to C in two or more points Amenta et al. (1998).

DEFINITION 4. Local Feature Size (LFS): Local feature size LFS (s) of a point s ∈ S is the Euclidean distance from s to the closest point m on the medial axis Amenta et al. (1998).

DEFINITION 5. ε-sampling: Let C be a curve and Sbs its boundary sample. Sbs is said to be an ε-sampled representation of C, if for every point c ∈ C, there exists at least a sample point s ∈ Sbs such that the Euclidean distance between c and s is atmost ε * LFS(c) where ε ≤ 1 Amenta et al. (1998).

DEFINITION 6. Skinny triangle: A skinny triangle is a thin non-obtuse triangle whose base is smaller than the distance between the circumcenter and the incenter of the triangle (Figure 2(c)).

The smallest angle in such triangles can be shown to be less than 19.5° (see Appendix A for more details). Hereafter, a skinny triangle is a non-obtuse triangle in the Delaunay triangulation whose smallest angle is less than 19.5° and the longest sides ranging from 2.996 to 2.824 times the smallest side as shown in Figure 3.

DEFINITION 7. r-sampling (Figure 2 (d)): Let Sdp be a dot pattern and ∂Sdp be the boundaries of Sdp. r-sampling is a sampling method where the distance between any pair of adjacent samples, p,q ∈ ∂Sdp is at most 2r and between any pair of non-adjacent samples where p ∈ ∂Sdp and q is an interior point is at least 2r Methirumangalath et al. (2017)

DEFINITION 8. Degree constraint: When there are more than two edges at a vertex (point), ‘degree constraint’ implies that only two shorter edges are retained (and all other edges are removed) from that vertex (point).

DEFINITION 9. Pseudo hole: A pseudo hole is a false hole present in the shape produced by the algorithm which is not present in the original shape.

4. Theoretical background

Let DT(S) (or simply DT) be the Delaunay triangulation of a given point set S and CT be Coordinated triangles. Let Sbs denote boundary sample and Sdp denote dot pattern. In this section, we show that a shared edge of coordinated triangles can be removed from DT (i.e. it won’t form part of reconstructed boundary). Also, we show that certain edges can be removed from skinny triangles of DT as well as by applying degree constraint.

4.1. Boundary sample

Let Cbs be a simple closed curve and Pbs be the polygonal approximation of Cbs generated from ε-sampled Sbs. We build our theory based on the observation that Pbs ⊂ DT(Sbs) by Amenta et al. Amenta et al. (1998). To be consistent with the smallest angle (19.5°) of skinny triangle, the maximum value of ε is to be less than 0.256. Hence, we choose ε = 0.25 for our theoretical analysis.

LEMMA 4.1. Let ebc be an edge shared between coordinated triangles in DT(Sbs), then ebc ∉ Pbs.

Figure 3: Illustration of the length of the longest sides of a skinny triangle bound between 2.996 to 2.824 times the smallest side.
Proof. Let \( \triangle abc \) and \( \triangle bdc \) be coordinated triangles (CT). Figures 4(a) and 4(b) show two possible configurations of CT. Then, \( \angle bdc > 90^\circ \) or \( \angle bac > 90^\circ \). For instance, assume \( \angle bac > 90^\circ \), then, by the definition of obtuse triangles, \( ||e_{bc}|| > ||e_{ab}|| \) and \( ||e_{bc}|| > ||e_{ab}|| \) (\( \|e_{bc}\| > \|e_{ab}\| \) and \( \|e_{bc}\| > \|e_{cd}\| \) if \( \angle bdc > 90^\circ \)). Also, when the circumcenters are on the same side of the shared edge, it is clear that the shared edge will be the longest in at least one of the two triangles. Under \( \epsilon \)-sampling of a simple closed curve, it has been proven that the longest edge of any triangle in the DT \( \notin P_{\text{bs}} \) Parakkat et al. (2018) which implies \( e_{bc} \notin \) boundary edge. 

\[ \text{LEMMA 4.2. Let } \triangle lmn \text{ is a skinny triangle in } DT(S_{bs}) \text{ and } e_{ln} \text{ is an edge such that } ||e_{ln}|| < ||e_{lm}|| \text{ and } ||e_{ln}|| < ||e_{mn}||. \text{ Then, } e_{ln} \notin P_{bs} \text{ and } e_{mn} \notin P_{bs}. \]

Proof. For an \( \epsilon \)-sampled curve in the plane, \( \epsilon < 1 \), the angle spanned by three adjacent samples is at least \( \pi - 4 \cdot \text{arcsin}(\epsilon/2) \) Amenta et al. (1998). For \( \epsilon = 0.25 \), the angle constituted by three adjacent points is at least \( 151^\circ \).

Consider Figure 4(c). Let \( \triangle lmn \) is a skinny triangle with edges \( e_{ln}, e_{lm}, e_{mn} \) where \( ||e_{ln}|| < ||e_{lm}|| \) and \( ||e_{ln}|| < ||e_{mn}|| \). By definition of skinny triangle (where the smallest angle has shown to be \( < 19.5^\circ \)), \( ||e_{mn}|| \approx \|e_{lm}\| > 2.83\|e_{ln}\| \) and all the interior angles are less than or equal to \( 90^\circ \). Therefore no three adjacent points constitute an angle greater than \( 151^\circ \). i.e. none of the pairs of edges be part of the boundary.

On the other hand, at most one side of a skinny triangle could be the part of the reconstructed boundary. The non-boundary edges cross the medial axis (otherwise it makes an interior angle greater than \( 90^\circ \)). Therefore the length of the non-boundary edges is at least \( 2 \cdot \text{LFS}(\epsilon) \). For \( \epsilon = 0.25 \), the length of a boundary is less than \( \text{LFS}(\epsilon) \). Then, only the smallest side, \( e_{ln} \), can be the candidate for the boundary edge. Therefore \( e_{ln} \notin P_{bs} \) and \( e_{mn} \notin P_{bs} \) as they both have length greater than \( \text{LFS}(\epsilon) \).

\[ \text{LEMMA 4.3. Let } e_{qt} \text{ be an edge removed by the degree constraint, then } e_{qt} \notin P_{bs}. \]

Proof. Let \( e_{ap} \in P_{bs} \) and \( e_{qs} \in P_{bs} \) (a sample configuration is shown in Figure 4(d)). Since \( S_{bs} \) follows \( \epsilon \)-sampling, \( ||e_{qt}|| > ||e_{ap}|| \) and \( ||e_{qt}|| > ||e_{qs}|| \). Under \( \epsilon \)-sampling of a simple closed curve, it has been proven that only the two smallest edges connected to a point are retained if the degree of that point is greater than 2 Parakkat et al. (2018). Hence \( e_{qt} \) is removed and \( e_{qt} \notin P_{bs} \).

\[ 4.2. \text{ Dot pattern} \]

Let \( P_{dp} \) represents the polygonal approximation of a simple closed curve \( C_{dp} \), generated from a \( r \)-sampled point set \( S_{dp} \). The Voronoi diagram of the \( r \)-sampled point set \( S_{dp} \) consists of an edge between every pair of adjacent points. i.e. there is an edge between every pair of adjacent points in the corresponding Delaunay triangulation (Delaunay triangulation is the dual of Voronoi diagram O’Rourke (1994)). Therefore \( P_{dp} \subset DT(S_{dp}) \).

\[ \text{LEMMA 4.4. Let } e_{bc} \text{ be an edge shared between coordinated triangles in } DT(S_{dp}), \text{ then } e_{bc} \notin P_{dp}. \]

Proof. Let \( \triangle abc \) and \( \triangle bdc \) be coordinated triangles. Figures 4(a) and 4(b) show two possible configurations of CT. Then, \( \angle bac > 90^\circ \) or \( \angle bdc > 90^\circ \). By the definition of obtuse triangles, if \( \angle bac > 90^\circ \), then
e or equal to three. Therefore $e \in$ boundary of the sample. Based on this finding, for each triangle $T$ given point set (Figure 5(b)).

5. Algorithm and Topological Correctness

LEMMA 4.6. Let $e_{bc}$ be an edge removed by the degree constraint, then $e_{ad} \notin P_{dp}$.

Proof. For a contradiction, let’s assume that $e_{ad}$ be an edge removed by the degree constraint and $e_{ad} \in P_{dp}$. Since $e_{ad}$ is an edge removed by the degree constraint, the vertices $a$ and $d$ are having a degree three or more. Consider the vertex $a$ and $c$ & $b$ are the other two vertices to which $a$ is connected to form edges $e_{ab}, e_{ac} \in P_{dp}$. Then $||e_{ad}|| > 2r$ and $||e_{ab}|| \leq 2r$ & $||e_{ac}|| \leq 2r$ by the property of $r$-sampling. Since $P_{dp}$ is the polygonal approximation of a simple closed curve $C_{dp}$ no vertex is allowed to have a degree greater than or equal to three. Therefore $e_{ad} \notin P_{dp}$ which is a contradiction.

5.1. Marking a shared edge in CT

Lemmas 4.1 and 4.4 indicate that the edges shared between coordinated triangles are not part of any boundary of the sample. Based on this finding, for each triangle $T \in DT$, the algorithm checks for CT with respect to $T$. If CT exist, the shared edge between those two triangles is marked (cyan edges in (Figure 5(c))).
5.2. Marking edges from a skinny triangle

Lemmas 4.2 & 4.5 indicate that two longer edges of a skinny triangle are not part of the boundary of the curve-shape. Hence, those edges in skinny triangles (shown in red in Figure 5(d)), are marked. Figure 5(e) shows all the marked edges so far, in cyan and magenta.

5.3. Applying degree constraint

A graph \( G \) is formed from the set of unmarked edges from DT (Figure 5(f)). Since the edges in DT are removed arbitrarily based on CT and skinny triangles, there are possibilities of the presence of non-manifold edges. In order to maintain the output as manifold (for e.g., if there exists only outer boundary, then it should be topologically equivalent to a circle), we impose a degree constraint (Definition 8) on each vertex.

Figure 5(g) shows the graph, which is the final reconstructed boundary (in blue) after checking for degree constraint for the point set shown in Figure 5(a).

The pseudo-code for the algorithm for reconstruction, given a set of points \( S \) is delineated in Algorithm 1. The running time complexity of the algorithm can be shown to be \( O(n \log n) \) where \( n \) is the number of points in \( S \).

Algorithm 1: Complete_Reconstruct(\( S \))

**Input:** Input point set, \( S \).
**Output:** Reconstructed Output \( R \).

1: Construct Delaunay triangulation, \( DT(S) \).
2: for each triangle \( T \) do
3: Take all three neighboring triangles and check whether they constitute coordinated triangles.
4: Mark the shared edges for all coordinated triangles.
5: Identify the skinny triangles and mark the two longest edges.
6: end for
7: Create a graph \( G \) with all unmarked edges of \( DT \) (if all three edges are unmarked, they are not considered).
8: Apply degree constraint on all vertices of \( G \).
9: return \( G \) as CT-shape

5.4. Topological Correctness

In this section, we show that the reconstruction using Algorithm 1 does not result in more number of holes than present in the object. We only show the proofs for a boundary sample (as they can derived in a similar manner for a dot pattern). Let \( S_{bs} \) be a \( \epsilon \)-sampled points from an object \( O \).
**Lemma 5.1.** The Delaunay triangles formed by the sample points on a hole boundary are always opened (the largest angle opened) towards the direction where the radii of the medial balls increases.

*Proof.* Suppose not. There exist a triangle $\triangle abc$ which is opened in the opposite direction of the rise of the radii of the medial ball. Then the triangle needs a vertex inside the hole (not on the boundary). This a contradiction. Please refer to Figure 6.

**Lemma 5.2.** Under $\epsilon$-sampling, the Algorithm 1 captures the inner-boundary (hole boundary).

*Proof.* Let us consider the types of Delaunay triangles formed in the holes. Type 0 (blue color in Figure 7): None of the sides is part of the hole boundary. Type 1 (green color in Figure 7): Only one side is part of the hole boundary. Type 2 (red color in Figure 7): Two sides are part of the hole boundary.

- Case 1-The triangle is a Type 2: Since the two sides of the triangle are part of the boundary, the angle spanned is greater than $151^\circ$. Consequently, Type 2 triangles are obtuse triangles. By Lemma 5.1 a Type 2 triangle and the subsequent Type 1 triangles are opened in the same direction as long as the radii of the medial ball increases. Therefore they are removed by applying CT constraint as they form a coordinated pair with one of its neighboring triangles.

- Case 2-The triangle is a Type 1: Type 1 triangles can be obtuse triangles or non-obtuse triangles. The obtuse triangles are handled by CT constraint as shown in Case 1. Otherwise, let it be a non-obtuse triangle. Let $a$ is the boundary edge and $b$&$c$ are the non-boundary edge. The distance between any two adjacent points($\|x - y\|$) is less than or equal to $\frac{2\epsilon}{1-\epsilon}LFS(x)$ (Dey (2006)). Then $\|a\|\leq\frac{2\epsilon}{1-\epsilon}LFS() \leq 0.666*\text{LFS}()$. The non-boundary edges cross the medial axis (Otherwise it makes an interior angle greater than $90^\circ$). Since $b$&$c$ are crossing the medial axis, $\|b\| \approx \|c\| \geq 2*\text{LFS}()$. Let $\|a\| = 0.666*\text{LFS}()$. Then $\|b\| \approx \|c\| > 3*\|a\|$. Hence, the angle opposite to the edge $a$ is less than $19.5^\circ$ which makes the triangle a skinny triangle. The two longer non-boundary edges of the skinny triangle are removed (Lemmas 4.2 & 4.5).

- Case 3-The triangle is a Type 0: These types of triangles either form a CT pair with adjacent triangles which in turn captured by CT constraint or it must be captured by degree constraint since none of the edges is part of the boundary.

All the remaining edges which are not part of the boundary are removed by degree constraint.

**Lemma 5.3.** The reconstruction using Algorithm 1 does not have any pseudo holes.

*Proof.* Suppose the Algorithm 1 captures a pseudo hole. Then the set of boundary points that constitute the pseudo hole is disjoint with the set of boundary points that constitute actual outer/inner boundary.
Figure 8: Results of our algorithm (CT-shape) for various features like concavity, multiple holes (deer has two holes), multiple components etc. Input DP = Input dot pattern, Input BS = Input boundary sample, Output = Output of our algorithm.

Otherwise, the degree of the shared vertex is greater than two which in turn captured by the degree constraint. If there is such a disjoint set of boundary points that constitute a pseudo hole, then the hole region is inhabited with three kinds of Delaunay triangles, namely Type 0, Type 1, Type 2 as mentioned in Lemma 5.2. By the Lemma 5.2, it is proven that any hole is populated with three kinds Delaunay triangles and our Algorithm 1 captures such kind of triangles to generate correct hole boundary. Then those triangles are no longer constitute a pseudo hole but it is an actual hole. Hence a contradiction.

**LEMMA 5.4.** Let \( S_{bs} \) be a \( \epsilon \)-sampled boundary sample from an object \( O \). Algorithm 1 guarantees topological correctness.

**Proof.** The proof follows from Lemma 5.2 (which ensures that hole boundaries are captured when present in the object) and Lemma 5.3 (which ensures there are no false positive holes).

6. Results & Discussions

Our algorithm (Algorithm 1) is implemented in C++ with CGAL (Version: 4.6) libraries and visualized in OpenGL and tested in MacOS 10.12.3. The input point sets (dot patterns and boundary samples) consist of points from simple objects, objects with multiple holes, objects with multiple components, objects with non-divergent concavities etc. The algorithm has also been tested with different sampling densities and distributions. Most of the inputs are generated from the images of MPEG database with the help of WebPlotDigitizer. The point sets are generated with no particular sampling as the algorithm has to be tested on generic inputs to assess its performance (only for theoretical analysis, sampling has been assumed as in Section 4). Figure 8 shows some of the results of our algorithm for various dot patterns and boundary samples. The figure shows that our algorithm can generate good results for both kinds of inputs with divergent features.
Figure 9: Effect of tuning the parameter in $\alpha$-shape (first four figures) and $\chi$-shape (next four figures).

Figure 10: Simple closed curve (dot pattern): Results of (a) $\alpha$-shape (b) $\chi$-shape (c) simple-shape (d) RGG (e) ec-shape (f) our result (CT-shape).

Figure 11: Simple closed curve (boundary sample): Results of (a) $\alpha$-shape (b) $\chi$-shape (c) simple-shape (d) RGG (e) ec-shape (f) Crawl (g) HNN-crust (h) Peel (i) WDM-crust (j) our result (CT-shape).

In the implementation, it is assumed that all the infinite edges incident to the infinite vertex is marked.

Also, to remove the artifact created by the irregular sampling (please refer to Figure 12), when $\triangle abc$, $\triangle arb$ and $\triangle cbq$ are neighbors of a $\triangle abc$ and $E$ be the set of corresponding edges, we use the following conditions:

- Let $e \in \triangle abc$, if $e$ is marked and all edges in $E \setminus \{e\}$ are unmarked, then $e$ is unmarked.
- Let $e_1, e_2 \in \triangle abc$, if $e_1$ and $e_2$ are marked and all edges in $E \setminus \{e_1, e_2\}$ are unmarked, then $e_1$ and $e_2$ are unmarked.
- Let $e_1, e_2, e_3 \in \triangle abc$, if $e_1, e_2$ and $e_3$ are marked and all edges in $E \setminus \{e_1, e_2, e_3\}$ are unmarked, then $e_1, e_2$ and $e_3$ are unmarked.
Figure 12: Illustration of artefacts handling: (i) point set (ii) Delaunay triangulation (DT) (iii) our result after removing the artefacts (iv) consider the set of Delaunay triangles highlighted in the blue-dotted circle (v) Let $\triangle abc$ be a triangle and $\triangle acp$, $\triangle arb$ and $\triangle bqc$ are neighboring triangles in DT (enlarged view)(vi) $\triangle acp$ and $\triangle abc$ form a CT pair and the shared edge $(e_{ac})$ is marked. (vii) $\triangle arb$ and $\triangle abc$ form a CT pair and the shared edge $(e_{ab})$ is marked. (viii) $\triangle bqc$ and $\triangle abc$ form a CT pair and the shared edge $(e_{bc})$ is marked. (ix) pseudo hole created after removing all the marked edges. (x) our result without removing the artefacts.

6.1. Comparison with similar methods

Here we considered five algorithms ($\alpha$-shape, $\chi$-shape, simple-shape, RGG, ec-shape) for dot pattern and nine algorithms ($\alpha$-shape, $\chi$-shape, simple-shape, RGG, ec-shape, Crawl, HNN-crust, Peel, WDM-crust) for boundary samples for the sake of comparison. The comparison has been done extensively for both qualitative and quantitative analysis. It may be noted that HNN-crust, WDM-crust, Crawl and Peel do not work for dot patterns and hence they have been included only for the comparison of results for boundary samples as input. For qualitative comparison, the point sets generated from the images created by ourselves and then sampled using WebPlotDigitizer. For quantitative comparison, the maps of various countries are obtained using $\chi$-shape software. In all the comparison results, we use circles to denote regions where boundaries are not well-approximated.

6.1.1. Qualitative Comparison

In this section, we compare our results with various state-of-the-art methods. In the case of algorithms that require parameter tuning, we chose the visually best result after tuning the parameter(s). The first four sets in Figure 9 show sample results of $\alpha$-shape for different values of $\alpha$. It can be seen that the result has a few holes until $\alpha$ becomes 88% and concavities between handle and body start filling for $\alpha$ values greater than 89%. For the $\alpha$ value of 89%, the cup does not have holes and have also captured the concavities between handle and body, we would be choosing such a result for comparison. The next four sets of figures in Figure 9 show the tuning for $\chi$-shape. The parameters that we used to generate the result is noted on the top left corner of the corresponding results in Figure 9.

Figures 10 and 11 show some of the results generated by various algorithms along with our results. Though all the algorithms were able to result in a good approximation of the shape for a simple closed curve, the results of various algorithms degrades for different types of features. For qualitative comparison, we visually compared the ability of our algorithm to capture various features like non-divergent concavity, multiple objects and unstructured holes with various other related methods.

- Non-Divergent concavities

A non-divergent concavity is the concavity in the shape in which, the radii of the outer medial ball is not monotonically decreasing towards the boundary Peethambaran and Muthuganapathy (2015b).
Figures 13 and 14 show the results of various algorithms on points sampled from an object with non-divergent concavity. It can be seen that all the algorithms except RGG (Figures 13(d) and 14(d)) and ec-shape (Figures 13(e) and 14(e)) work for objects with non-divergent concavities. RGG is also a Delaunay-based sculpting algorithm whose edge removal rule prevents removing edges from where divergent concavity ends. As a consequence, the boundaries are not well approximated. For dot patterns, though ec-shape captures the feature very well, it could not capture the outer boundary of the object perfectly (as shown in Figure 13(e)). It is because, for an edge in the beginning of the non-divergent concavity, all three circles (diameter, chord and mid-point) are becoming empty and hence retained. Though α-shape captures the non-divergent concavity for dot pattern (Figure 13(a)), it failed to capture the same for boundary sample (Figure 14(a)). A few other algorithms also failed to capture this feature for boundary samples (see the circled regions in Figure 14).

- **Multiple Objects**

To ensure that the final output to be a simple polygon, Delaunay-based sculpting strategies make use of a graph constraint that the triangle is sculpted only when one of its vertex is not already part of the detected boundary. This simple graph constraint works very well if the input has only one object and fails in the case of multiple objects. Since χ-shape, simple-shape, RGG and ec-shape follows Delaunay sculpting strategy with graph constraint, as shown in Figures 15 and 16, their results are always a single connected component. Though α-shape captures the shape well in the case of dot patterns, it failed to capture shape from boundary samples. It can be seen that our algorithm is able to capture multiple objects irrespective of the input type.

- **Unstructured holes**

13
Delaunay sculpting strategies stop sculpting once it reaches the shape boundary. Hence $\chi$-shape, simple-shape, RGG and ec-shape result only in one outer boundary. After detecting the shape boundaries, RGG Peethambaran and Muthuganapathy (2015a) and extension of ec-shape Methirumangalath et al. (2017) make use of extra criteria to identify holes. These holes are identified and captured based on the existence of particular structures in the Delaunay triangulation (for e.g. body-arm structure in Peethambaran and Muthuganapathy (2015a)). Figures 17 and 18 show the result of various algorithms for a point set sampled from an object with multiple holes. With the presence of a pseudo-hole inside the body, $\alpha$-shape captured the holes perfectly. Due to the unavailability of any hole-detection strategies, $\chi$-shape and simple-shape resulted in only outer boundaries. Absence of a body-arm structure on the entire hole resulted in capturing only a partial hole information by RGG (Figure 17(d)).
Figure 18: Object with holes (boundary sample): Results of (a) $\alpha$-shape (b) $\chi$-shape (c) simple-shape (d) RGG (e) ec-shape (f) Crawl (g) HNN-crust (h) Peel (i) WDM-crust (j) our result (CT-shape).

ec-shape has captured all the holes but overdigged all of them and also captured a few pseudo-holes. It can be seen that our algorithm has captured the boundaries properly along with all holes (Figure 17(f)). A similar trend can be observed for boundary samples (Figure 18) but ec-shape has captured the details well in this case.

It can be seen from the comparative results that CT-shape has captured the details of the boundaries quite well under different features present in the object irrespective of the input type. Also, it should be noted that HNN-crust, Crawl and Peel have worked very well for boundary sample across objects having different features but unfortunately, those algorithms do not work for dot patterns.

6.1.2. Quantitative Comparison

This section discusses how similar algorithms work on input point sets with varying densities in comparison with our algorithm. The results of various unified algorithms are compared with the ground-truth for different inputs generated using the $\chi$-shape interface. For quantitative comparisons, we only needed the outer boundary as each of the maps have only one boundary (we ignored all holes). As a measure, we made use of $L^2$-error norm Duckham et al. (2008) to compare the results, which is defined as:

$$L^2 = \frac{\text{area}((C - P) \cup (P - C))}{\text{area}(C)}$$

(1)

where $C$ and $P$ are the ground truth and reconstructed result respectively. Figure 19 shows the results of various unified reconstruction algorithms on varying point densities sampled from the boundary of one country (Spain). Figure 20 shows the point density versus $L^2$ error plots of our experimentation on various point sets extracted from the boundary of the different country shapes (Paraguay, Slovakia, Spain and Zimbabwe) for both dot patterns and boundary samples. From Figure 20, it is clear that our algorithm works better or on par with various unified reconstruction algorithms.

6.1.3. Performance on Varying Point distribution

We have also conducted experiments on varying point distributions. Figure 21 shows the comparison of our results with other unified algorithms for various point distributions. The four instances of point
Figure 19: A qualitative comparison of the shape reconstruction algorithms for Spain point set with varying densities. All the point distributions used for the experiment are randomly distributed. Area of original shape is computed using $\chi$-shape software (Duckham et al. (2008)).

Figure 20: Exemplification of performance comparison on different point densities of different country shapes.
Our algorithm has captured the details quite well except in the case of SPDI. \( \alpha \)-shape, \( \chi \)-shape and simple-shape seem to have better results but it should be noted that they require tuning of parameter(s). As there exists a flipped triangle at the beginning of concavity in the U-shape, RGG could not capture any of the concavities irrespective of the distribution. The presence of a body-arm structure resulted in a pseudo-hole in SBDI of RGG. ec-shape captured the outer boundaries for various distributions along with few pseudo-holes.

6.1.4. Removing skinny triangles from \( \alpha \)-shape

In general, \( \alpha \)-shape has not performed well for boundary samples. It might be that the presence of skinny triangles are blocking some edges to be removed. However, as can be observed from Figure 22, removing the two longest edges from skinny triangles will not improve the result. This demonstrates that we cannot use an existing approach and improve upon its result. Our approach of using CT, skinny triangles and degree constraint has resulted in better outputs.
6.2. Effect of non-$\epsilon$-sampling

In this section, we demonstrate the efficiency of our algorithm to handle the features represented by insufficient points (not comply with $\epsilon$-sampling). We start with the epsilon sampled point set ($\epsilon < \frac{1}{3}$) of the model BUNNY Ohrhallinger et al. (2016). The experiment is conducted on the insufficiently sampled convex and concave region as shown in Figure 23. Figure 23(a) is the epsilon-sampled point set and Figure 23(b) is the corresponding result produced by our algorithm. Figure 23(c)-(g) shows the effect of the under-sampled convex region to the result. It is clear from Figure 23(c)-(f) that our algorithm can produce a decent output even for the point set not complying with epsilon sampling criteria. In Figure 23(g) the samples on the right side of the left ear are closer to the samples on the left side of the right ear. Therefore the curve is opened at a point on the left side of the left ear as it is wrongly connected to a point left side of the right ear. Subsequently, the degree constraint removes the open curve. Figure 23(h)-(l) depicts the effect of the sparsity of samples on the concave region. When the region is highly sparse-sampled the algorithm can not reproduce the feature decently as shown in Figure 23(l).

6.3. Running time

The algorithm starts with computing the Delaunay triangulation (DT) of the point set. It can done with a cost of $O(n \log n)$. Further, every triangle in DT is processed for possible CT (Coordinated triangles) pair and subsequent marking of its shared edges. This process requires only a cost of $O(n)$. Therefore the total complexity is bounded by $O(n \log n)$.

6.4. Limitations

When the input has noise around the boundaries, the algorithm cannot handle such data. This is because, the boundary sample with noise then becomes a dot pattern with a very less number of points interior to the boundary (Figure 24(a)). The result of the algorithm is shown in Figure 24(b), which is not satisfactory. However, if we can apply a noise simplification (Figure 24(c)) such as one proposed in Parakkat et al. (2018), then our algorithm produces a reconstruction result (Figure 24(d)). Features like open curves and self-intersections Parakkat et al. (2018) are not applicable for dot-pattern and hence our algorithm does not handle them.

6.5. Conclusion and Future works

In this paper, we devised a unified reconstruction algorithm and showed it works irrespective of the type of point set (dot pattern or boundary sample). The algorithm is easy to implement and proven to give good results under various point densities and distributions. We have conducted an extensive comparative study (both quantitatively and qualitatively) and as a non-parametric unified approach, it captures different
features (non-divergent concavity, multiple components, unstructured holes etc) from the point set which
was shown as limitations in the current literature. In contrast to other non-parametric unified algorithms,
our algorithm needs only a single pass to detect the boundaries (both inner and outer) irrespective of the
number of holes/objects. As a future work, we would like to extend the algorithm to look into various other
challenging tasks like handling point sets with noise and outliers. We are also working on the extension of
the proposed algorithm to higher dimensions.

References


Dey, T.K., 2006. Curve and Surface Reconstruction: Algorithms with Mathematical Analysis (Cambridge Monographs on


28, 1371–1382.

doi.org/10.1016/S0925-7721(99)00051-6.

S0925-7721(01)00015-3.

of a set of points in the plane. Pattern Recognition 41, 3224–3236.

Edelsbrunner, H., 1998. Shape reconstruction with delaunay complex., in: Lucchesi, C.L., Moura, A.V. (Eds.), LATIN,

Information Theory 29, 551–558.

Ganapathy, H., Ramu, P., Muthuganapathy, R., 2015. Alpha shape based design space decomposition for island failure regions
in reliability based design. Structural and Multidisciplinary Optimization 52, 121–136. URL: https://doi.org/10.1007/s00158-014-1224-6.

reconstruction in the plane. IET computer vision 5, 97–106.


Methirumangalath, S., Kannan, S.S., Parakkat, A.D., Muthuganapathy, R., 2017. Hole detection in a planar point set: An

Methirumangalath, S., Parakkat, A.D., Muthuganapathy, R., 2015. A unified approach towards reconstruction of a planar
A. Characterisation of a skinny triangle

Consider the configuration as shown in Figure A.25. Let $C$ be the unit circle with center $o$ and $\Delta_{i,j,k}$ is a triangle inscribed in $C$. Let $cc$ be the circumcentre (coincide with the origin $(0,0)$) and $ic$ be the incenter of the $\Delta_{i,j,k}$. Let $d$ is the distance between $cc$ & $ic$ and $b$ is the base of the $\Delta_{i,j,k}$. From the $\Delta_{i,n,o}$, we get $x_1 = - \sin \theta$ & $y_1 = - \cos \theta$. Therefore $b = 2 \sin \theta$ (by the symmetry of the base with respect to the $y$-axis).

Consider the $\Delta_{i,n,k}$ & $\Delta_{i,n,o}$. Let $\angle nki = \gamma/2$, $\angle noi = \theta$, $\|ini\| = \sin \theta$, $\|ok\| = 1$, and $\|no\| = \cos \theta$.

Then,

$$\tan \gamma/2 = \frac{\sin \theta}{1 + \cos \theta} = \tan \theta/2. \quad (A.1)$$

Therefore $\theta = \gamma$ Let $ic(x,y)$ be the coordinates of the incenter of $\Delta_{i,j,k}$ and $I, J, K$ are the sides of the $\Delta_{i,j,k}$.
opposite to the coordinates $i, j, k$ respectively.

\[
ic(x) = \left( \frac{x_1 I + x_2 J + x_3 K}{I + J + K} \right) \quad (2)
\]
\[
ic(y) = \left( \frac{y_1 I + y_2 J + y_3 K}{I + J + K} \right) \quad (3)
\]

From the $\triangle_{k,n,j}$,

\[
I = \sqrt{\sin^2 \theta + (1 + \cos \theta)^2}
\]
\[
= \sqrt{\sin^2 \theta + 1 + \cos^2 \theta + 2 \cos \theta}
\]
\[
= \sqrt{\sin^2 \theta + \cos^2 \theta + 2 \cos \theta + 1}
\]
\[
= \sqrt{2 + 2 \cos \theta}
\]
\[
= 2 \cos \frac{\theta}{2}
\]

Similarly, $J = 2 \cos \frac{\theta}{2}$ (by symmetry with respect to $y$-axis). By substituting values for $I, J, K$ in Equation (3),

\[
ic(y) = \frac{2 \cos \frac{\theta}{2} \ast - \cos \theta + 2 \cos \frac{\theta}{2} \ast - \cos \theta + 2 \sin \theta}{2 \cos \frac{\theta}{2} + 2 \cos \frac{\theta}{2} + 2 \sin \theta} \quad (4)
\]

When $cc - ic = b$, the Equation 4 becomes

\[
-2 \sin \theta = \frac{2 \cos \frac{\theta}{2} \ast - \cos \theta + 2 \cos \frac{\theta}{2} \ast - \cos \theta + 2 \sin \theta}{2 \cos \frac{\theta}{2} + 2 \cos \frac{\theta}{2} + 2 \sin \theta}
\]
\[
-2 \sin \theta = \frac{-4 \cos \frac{\theta}{2} \cos \theta + 2 \sin \theta}{4 \cos \frac{\theta}{2} + 2 \sin \theta}
\]
\[
-2 \sin \theta = \frac{-2 \cos \frac{\theta}{2} \cos \theta + \sin \theta}{2 \cos \frac{\theta}{2} + \sin \theta} \quad (5)
\]

By solving the Equation (5), we get $\theta \approx 19.5^\circ$