Local Delaunay-based High Fidelity Surface Reconstruction from 3D Point Sets

Submission Number: 40

Abstract

In this paper, we introduce a feature preserving surface reconstruction algorithm to produce a high fidelity triangulated mesh from an input point set. The concept of local Delaunay triangulation is applied to speed up the reconstruction procedure and to preserve features. The proposed algorithm has running complexity of $O(n \log n)$, where $n$ is the number of points. Additionally, the local Delaunay triangulation improves the memory efficiency of the proposed method compared to global Delaunay-based methods. We introduced the concept of minimum circum-radius triangle to select the prime-triangle in the local Delaunay mesh. Based on the local projection and minimum circum-radius, our triangle selection criteria ensures that the output mesh will be free from non-manifold edges and fold-over triangles. On top of that, we have provided the theoretical correctness of the proposed algorithm with the assumption of $\epsilon$-sampling model. The experimental results show that the proposed method is capable of producing high fidelity meshes from large real-world non-uniform data. We also show the effectiveness of the proposed method compared to the state-of-the-art methods in terms of visual and quantitative analysis.

1. Introduction

Given a set of points, $S \subset \mathbb{R}^3$, sampled from a surface $F$, surface reconstruction is a process of unveiling the geometrical and topological structures of $F$ from $S$ that approximates the surface $F$ [Giesen (2004)]. The problem of surface reconstruction has been in-force over the past several decades. Nevertheless, a comprehensive solution is not yet achieved because of the ill-posed nature of the problem [Edelsbrunner (1998)]. In general, the reconstruction problem is inherently challenging because the mathematical formulation of the concept of an optimal shape is a cumbersome task except for the special categories of shapes, e.g. platonic solids. Plenty of algorithms have been devised for surface reconstruction with different capabilities of handling distinct features. Major drawbacks in the literature are:

- Most of them require multiple parameters, which make the task tedious as the user has to find out the suitable values of the parameters by trial and error.
- Only a few algorithms are able to preserve sharp features and dealing with boundaries.
- Some of them are computationally expensive and not suited for real-time applications and big data.
- Some algorithms require additional information along with the point set e.g. vertex normals.
- Only a few of them provide the theoretical guarantee of the outcomes (e.g. manifold and topology).

In this paper, we propose a local Delaunay-based surface reconstruction algorithm, which produces a high fidelity mesh (a mesh without non-manifold edges and free from fold-over triangles) and is robust against non-uniform sampling. Our method handles surfaces with boundaries and does not destroy sharp features during the reconstruction. Further, the method is memory efficient and fast as we construct only the local Delaunay triangulation (DT). Since we need to compute only the local DT, it is required to store only vertices and faces of the local DT, which is very small compared to global DT and hence it can be computed very fast as discussed in [Gopi et al. (2000); Linsen and Prautzsch (2001)]. The key features of the proposed method include:
• Running complexity of $O(n \log n)$: The proposed method has a running time complexity of $O(n \log n)$, because of the local Delaunay triangulation. Therefore, our method is suitable for large point sets.

• Handling surface boundaries: The proposed method works equally well on an object with and without boundaries.

• Retaining sharp features: The local Delaunay triangulation does not destroy sharp features during the surface reconstruction provided that a set of points are available at sharp features.

• Robust against non-uniform sampling: The proposed method is capable of producing a high quality output against the non-uniform sampling.

• Producing a high fidelity mesh: The selection of non-overlapping prime-triangles ensures the output mesh quality and it is free from fold-over triangles.

• Theoretical correctness: With the assumption of the $\epsilon$-sampling model, the correctness of the proposed algorithm is established.

2. Related work

The literature consists of many state-of-the-art methods for surface reconstruction. A comprehensive review is presented in Berger et al. (2017). Here, we give a short overview of the various methods, which are broadly classified into two categories: implicit and explicit methods.

2.1. Implicit methods

These methods use a distance function to assign each point a signed distance from the surface. Further, a contour algorithm is used to extract the zero-set of the distance function. From the computed zero-set, a polygonal representation of an object is formed. One of the initial works in this direction is the tangent plane estimation method by Hoppe et al. (1992), where the vertex normal computation is done by principal component analysis. Further, the zero iso-surface is extracted using the marching cubes algorithm. Guennebaud and Gross (2007) introduced a point set surface based on moving least-squares fitting of algebraic spheres. The surface can be represented either by a projection procedure or in an implicit form. The method is performing well for low sampling rates and in the presence of high curvature. RIMLS (Oztireli et al. (2009)) is a tool for the mesh-less representation of surfaces by combining the advantages of IMLS (Implicit Moving Least Squares) and LKR (Local Kernel Regression) functional approximation of irregular data. Unlike other implicit methods, RIMLS is capable of capturing sharp features. Poisson surface reconstruction Kazhdan et al. (2006) transforms the problem of surface reconstruction from oriented points as a spatial Poisson problem. This method computes a 3D indicator function to assign a value of one for interior points and a zero for exterior points and it extracts the appropriate iso-surface, which is the reconstructed surface. Screened Poisson surface reconstruction Kazhdan and Hoppe (2013) is an extended version of the Poisson reconstruction, which generates water-tight surfaces from oriented points. Digne et al. (2011) proposed a scale-space strategy for orienting and meshing the raw original surface to ensure the rendering of textures and the detection of scanning artifacts.

2.2. Explicit methods

Explicit methods interpolate the sample points from the point set to generate a triangulated reconstructed surface and are divided into two groups: Delaunay triangulation (DT) / Voronoi diagram (VD) based methods and Region growing methods.

DT/VD based methods: DT / VD methods are based on the idea that the computed triangulated surface is a sub-complex of DT. In general, 3D DT is computed from the input points and tetrahedrons are removed from the convex hull to extract the original shape. Again, DT/VD methods are sub-classified as Global DT/VD and Local DT/VD based on the construction of DT/VD: locally for the neighborhood of each point or globally for all the input points.
Global DT/VD: Sculpting is one of the widely used techniques in DT / VD-based methods. Boissonnat’s
sculpting algorithm [Boissonnat 1984] is one of the early work in this direction, in which tetrahedrons
from DT are eliminated sequentially until the reconstructed shape satisfies the definition of a polyhedron.
From DT are eliminated sequentially until the reconstructed shape satisfies the definition of a polyhedron.
Peethambaran and Muthuganapathy (2015) introduced Shapehull sculpting algorithm, which removes the
tetrahedron of DT subjected to the circum-center and topological constraints. Edelsbrunner and Mücke
(1994) introduced α-shape, which decides a tetrahedron to be retained or removed based on the ball-radius
α. Crust (Amenta et al. 1998) and power-crust (Amenta et al. 2001) reconstruct a provable water-tight
surface from an unorganised 3D point set. The cocone-family algorithms (Tight-cocone by Dey and Goswami
2003) and Robust-cocone by Dey and Goswami (2006) compute a provable water-tight surface from a set
of unorganized points. Dey et al. (2012) proposed a reconstruction algorithm for singular surfaces that
can handle various singularities like boundaries, sharp features, non-manifolds etc. in a unified framework.

Ohrhallinger et al. (2013) proposed a surface reconstruction algorithm, which provides a better interpolation
of sparsely sampled features. Methirumangalath et al. (2017) presented a simple triangle removal approach,
which is retaining solitary triangles and removing triangles anywhere based on the circum-radius of a triangle.

Local DT/VD: Gopi et al. (2000) introduced a fast and memory efficient method for surface reconstruction
based on local DT. This method computes the local neighbours for each reference point and projects on
to the corresponding tangent plane. Further, the local DT is computed and the Delaunay neighbors are
extracted. Finally, the triangles of the reference point are computed from the Delaunay neighbors. Dey et al.
(2010) proposed an octree-based localized Delaunay refinement method for meshing 3D surfaces. Linsen and
Prautzsch (2001) presented a fast local triangulation of the point set and this method works equally well
for the object representation w.r.t. global triangulation. Further, Linsen Li (2003) presented an alternative
to meshes known as fan clouds, which can be easily computed by exploiting the property "localness". Yan
et al. (2014) introduced a localized restricted VD (LRVD) to overcome the problems like re-meshing of
nearby sheet structures and low resolution point sets. Boltcheva and Lévy (2017) discussed a restricted
VD based surface reconstruction algorithm, which computes the restricted VD in parallel. They require a
pre-processing step called smoothing and a post-processing step known as hole-filling. However, most of
these steps can be parallelized which makes the algorithm suitable for a large point set. Recently, Ray et al.
(2018) proposed a parallelizable GPU algorithm to compute 3D VD which returns only the geometry of the
Voronoi cells rather than a combinatorial mesh data structure. For a more detailed review, please refer to

Region growing methods: These approaches are another group under explicit methods. These meth-
ods start with a seed triangle and join new points to the existing region boundary until all points are
considered. One of the well-known approaches is Ball pivoting algorithm [Bernardini et al. 1999], where the
region growing is performed by pivoting a ball of a user-specified radius on an edge of the selected triangle.
Recently, a point set denoising method is used to improve the reconstruction fidelity with Ball pivoting algo-
rithm by Yadav et al. (2018). Kuo and Yau (2005) combined the advantages of DT and region-growing based
approaches to generate an orientable manifold triangulated mesh. Cohen-Steiner and Da (2004) proposed a
greedy surface reconstruction algorithm for models with or without boundaries. Combining region-growing
and Delaunay approaches, Wang et al. (2019) introduced an algorithm for surface reconstruction from an
unoriented point set. The method handles the point set with imperfections.

In this paper, we propose a surface reconstruction algorithm, which takes a set of points, non-uniformly
distributed, sampled from a surface with/without boundaries, with sharp features, and produces a high-
quality mesh with a cost of \( O(n \log n) \). Our method is similar in approach to the method of Gopi et al. (2000)
and to the method of Boltcheva and Lévy (2017), but the triangle selection criterion (triangle formation of
the reference point) and the number of steps required are different.

To select the triangles around a point, Gopi et al. (2000) extracts the Delaunay neighbours after com-
puting 2D local DT around the point and its local neighbours. From the set of neighbouring, the triangles
incident with the point is formed in the 2D domain. This kind of triangle selection criterion leads to the
introduction of false/wrong edges (for instances, in a quadrilateral A, B, C, and D, there can be the cases
where both BD and AC are Delaunay edges) or the rejection of some valid edges (which causes the holes)
especially the degenerate point sets. In contrast to Gopi et al. (2000), we do not require any pre-processing
step like vertex normal computation. Similarly, since we compute the non-overlapping mesh triangles from
the local 3D DT around the reference point unlike the method Gopi et al. (2000), we do not require any post-processing steps like hole-filling.

The method introduced by Boltcheva and Lévy (2017) requires some additional steps like pre-processing (smoothing), post-processing (hole filling), manifold extraction, normal-estimation etc. Most of these steps can be parallelized, except the hole filling step. However, it requires multiple parameters which makes the users try all the possible combinations to generate the optimal result. Contrarily, our method requires only a single parameter and fewer steps.

3. Method overview

Our method falls under the category of advancing front paradigm for surface reconstruction, where we compute the triangle fan (a set of connected triangles for a point) for each point from the local DT of its neighboring points as discussed in Gopi et al. (2000). The method takes a set of unorganized points S, sampled from a surface with or without boundaries, as the input with no additional information like normals. The method returns a set of triangles, which preserves the geometrical and topological properties of the surface by interpolating the given point set. The method progresses through 2 major steps:

- **Local Delaunay triangulation:** In the first step, we compute DT of the local neighbourhood of a point. To delineate the local neighbourhood, we use knn (k-nearest neighbour).

- **Extracting the triangles with respect to a point (triangle fan):** In this step, we select the triangles incident with the point from the local DT as the set of candidate triangles. Further, we project the triangles from the candidate set to a least-square plane defined by their vertices and extract the triangle fan of the point.

The above two steps are performed for each point in the point set. The triangle fans extracted for each point constitutes the required mesh representation of the surface. For each point, the computed triangle fan shares at most two triangles with the triangle fans of the corresponding neighbors. Therefore, when a point is getting processed, only the remaining triangles in the triangle fans are computed. As a result, the stitching of the triangle fans of the points is happening simultaneously.

4. Method

In this section, we discuss the method in detail using the following definitions.

**Definition 1.** Non-overlapping triangles: Let \( t_1 \) & \( t_2 \) be two triangles and \( t_1 \) \& \( t_2 \) are said to be non-overlapping when either they are not intersecting or they are intersecting only at the vertex (Figure 1(a)) or at an edge (Figure 1(b)) after projecting onto a plane \( H \). Figure 1(c) shows the overlapping triangles.

The plane \( H \) is the least square fitting of vertices of \( t_1 \) & \( t_2 \).

**Definition 2.** Non-manifold edge: Let \( e_i \) be an edge in a mesh. If more than two triangles are incident on \( e_i \), then \( e_i \) is said to be a Non-manifold edge.

**Definition 3.** Manifold constraint: A triangle is said to satisfy the manifold constraint when the insertion of the triangle to a mesh does not introduce any non-manifold elements (non-manifold edges or non-manifold vertices) in the mesh.

Here we restrict our manifold constraint definition to the edge manifold and we evaluate the quality of the output mesh based on the edge manifold criterion.

![Figure 1: The above figure shows overlapping and non-overlapping triangles after projecting the triangles onto a least-square plane defined by their vertices. (a) & (b) are non-overlapping triangles (c) overlapping triangles.](image-url)
Definition 4. Prime-triangle: Let $T(p_i)$ be a set of triangles incident with a vertex $p_i$. Then $t \in T(p_i)$ is said to be a prime-triangle with respect to $p_i$ if $t$ is the smallest circum-radius triangle.

Let us consider an input point set $S = \{p_0, p_1, \cdots, p_{n-1}\} \subset \mathbb{R}^3$ sampled from a surface with $n \in \mathbb{N}$ denoting the number of points. The following sections explain the pipeline of the algorithm in detail.

4.1. Local Delaunay triangulation

In the first stage of the proposed method, local neighbourhood selection is performed to obtain the local DT. For a given point $p_i$, the set of indices of local neighbours is represented as $\Omega_k^i$, where $k$ is the number of neighbour points.

Under $\epsilon$-sampling, it has been proven that the 3D DT of a sufficiently dense set of samples contains a piece-wise-linear surface homeomorphic to the surface $F$ (see Theorem 2 in Amenta and Bern (1999)). Therefore, we construct the local DT $(DT(\Omega_k^i))$ at every point $p_i$, and the set of triangles in $DT(\Omega_k^i)$ is the candidate set for the subsequent steps in the pipeline. In section 4.3.1, we establish the relation between the mesh $M$, $DT(\Omega_k^i)$, and $DT(S)$.

4.2. Fetching the candidate triangles for a point $p_i$

The mesh $M$ is composed of the non-overlapping triangles, which are incident with each point $p_i$ collected from $DT(\Omega_k^i)$. Most of the triangles in $DT(\Omega_k^i)$ do not contribute to the final mesh. Therefore, a set of triangles incident to the point $p_i$ is extracted from $DT(\Omega_k^i)$ which is termed as a candidate set $T(p_i)$. In other words, the set of triangles formed by the 1-ring neighbourhood of $p_i$, where $p_i$ is the central vertex in $DT(\Omega_k^i)$, is the candidate set $T(p_i)$ for further processing.

4.3. Extracting the triangles with respect to a point $p_i$

For each reference point $p_i$, a set of connected non-overlapping prime-triangles incident with the point is extracted from $T(p_i)$ and stored in a set $t_{\text{prime}}(p_i)$. Initially, $T(p_i)$ is sorted in the ascending order of their circum-radii. Subsequently, a minimum circum-radius triangle is selected from the set $T(p_i) \setminus t_{\text{prime}}(p_i)$ and designated as the prime-triangle. Further, the prime-triangle is added to the set $t_{\text{prime}}(p_i)$ only if it does not overlap with the existing prime-triangle(s) in $t_{\text{prime}}(p_i)$. This process is repeated until there are no more non-overlapping prime-triangles with respect to the triangles in $t_{\text{prime}}(p_i)$ for the point $p_i$. During the selection of each prime-triangle, manifold constraint is evaluated. This process is repeated for all points $p_i \in S$.

To check whether or not the candidate prime-triangle overlaps with any of the already selected prime-triangles, we project the prime-triangles along with the candidate prime-triangle on to a least-square plane defined by the vertices of those triangles. After the projection, the intersections of the candidate prime-triangle with the other prime-triangles are computed. If the candidate prime-triangle is a non-overlapping (i.e., they do not intersect) for all prime-triangles, then we add it to the list of the corresponding point. If it overlaps, we proceed to the next triangle from $T(p_i)$ and repeat the process.

4.3.1. Theoretical validation of prime-triangle selection

In this section, we set up our theoretical framework based on the $\epsilon$-sampling model (Amenta et al. 2001) and use the following definitions to build up the lemmas.

Definition 5. Medial axis/Medial surface (MA): Let $F$ be a smooth surface without boundaries. The medial axis of $F$ is the locus of centers of medial balls that are tangent to $F$ in two or more points (Amenta et al. 2001).
Definition 6. Local Feature Size (lfs): Let \( x \) be a point on the surface \( F \). \( \text{lfs}(x) \) is the minimum Euclidean distance from the point \( x \) to any point on the medial axis \cite{Amenta2001}.

Definition 7. \( \epsilon \)-sampling: Let \( F \) be a smooth surface without boundaries. \( S \) is an \( \epsilon \)-sampled point set from \( F \) when the distance from any point \( x \in F \) to the nearest sample is at most \( \epsilon \times \text{lfs}(x) \), where \( 0 < \epsilon < 1 \) \cite{Amenta2001}.

For the theoretical justification that the output mesh is the best approximation of the original surface, we use the following assumptions.

Let \( F \) be a smooth surface without boundaries, \( M \) be a mesh representation of \( F \), and \( S \) be the \( \epsilon \)-sampled point set (the points are in general position) from \( F \). We assume that for every point \( p_i \in S \), the local neighbourhood includes all the points from the 1-ring neighbourhood of \( p_i \in D T(S) \). In other words, the 1-ring neighbourhood of \( p_i \in D T(\Omega^1_i) \) is the same as the 1-ring neighbourhood of \( p_i \in D T(S) \). Hence, all the prime-triangles for a point \( p_i \) in \( D T(\Omega^1_i) \) are also present in \( D T(S) \) (by the nearest neighborhood property of \( D T \) by \cite{O'Rourke1994}). Therefore, we use the fact \( M \subset \bigcup_{i=0}^{n-1} D T(\Omega^1_i) \) to establish the theory.

To show that the prime-triangle selected by the method is always on the mesh \( M \), we construct Lemma 4.3. Lemmas 4.1 & 4.2 are supporting Lemma 4.3.

Lemma 4.1. Let \( e_{p_i p_k} \) be an edge of \( \triangle p_i p_j p_k \subset D T(\Omega^1_i) \) intersects the MA. Then \( \|e_{p_i p_k}\| \geq 2 \times \text{lfs}(p_j) \).

Proof. Let us assume \( e_{p_i p_k} \) be an edge of \( \triangle p_i p_j p_k \) intersects the MA at a point \( m \). Then \( \|e_{p_i p_j}\| \geq \text{lfs}(p_j) \) since \( \text{lfs}(p_j) \) is the shortest distance from a sample \( p_j \) to the MA. The edge \( e_{p_j p_k} \) intersects the MA by bisecting its dual Voronoi edge. Therefore \( \|e_{p_i p_k}\| \geq 2 \times \text{lfs}(p_j) \).

Lemma 4.2. Let \( p_i \) be the reference point and \( \triangle p_i p_j p_k \) be the current prime-triangle selected by the proposed method. Then \( \triangle p_i p_j p_k \) (any of the edges of \( \triangle p_i p_j p_k \)) does not intersect the MA (In 3D, MA will be a surface \cite{Ramanathan2010} unlike Figure 2 see definition 2).

Proof. For a contradiction, let’s assume that the above lemma is false. Then at least one edge of \( \triangle p_i p_j p_k \) intersects the MA. Let the edge \( e_{p_i p_k} \) intersect the MA (Figure 2(a)). Then \( \|e_{p_i p_k}\| \geq 2 \times \text{lfs}(p_j) \), by Lemma 4.1. Let \( x \) be a point on the surface located at the midpoint of the edge \( e_{p_i p_k} \) (presented in green color in Figure 2(a)). To comply with \( \epsilon \)-sampling, there will be a point \( y \), within the circle of radius \( \text{lfs}(x) \) as shown in magenta color in Figure 2(a). Then by the nearest-neighbor-property of \( D T \) by \cite{O'Rourke1994}, the points \( p_i, p_j, y \) or \( p_i, y, p_k \) form a triangle with smaller circum-radius and that leads to a contradiction.

Therefore, the above lemma ensures that the prime-triangles are possibly lying on the surface as none of its edges intersect the MA. Furthermore, \cite{Peethambaran2015} concluded that the triangles with smaller circum-radius are capable of capturing the geometrical proximity of the surface.

Lemma 4.3. Let \( t_{\text{prime}}(p_i) \) be the prime-triangle \( (\triangle p_i p_j p_k \text{ selected by the proposed method with } \angle p_i p_j p_k = \theta) \) incident with the reference point \( p_i \). Then \( t \in M \).

Proof. There are two cases to be considered:

- Case A (special case): Let us suppose that \( t \) is the part of a sliver tetrahedron (we use the same notion of sliver tetrahedron as discussed in \cite{Cohen-Steiner2004}). Then any pair of triangles with the dihedral angle approximately equal to \( 180^\circ \) would lead to equally good reconstruction, both topologically and geometrically \cite{Cohen-Steiner2004}. Therefore, there always exists a triangle paired with \( t \) (paired by a common edge) and their dihedral angle approximately equal to \( 180^\circ \). Therefore \( t \in M \).

- Case B (general case): For a contradiction, let’s assume that the above lemma is false. Then there exists at least one triangle with larger circum-radius, which replaces the prime-triangle \( t \). Then there are two cases to be considered.


Let us suppose that there is only one triangle. Let \( \triangle p_ixy \in M \) be the triangle with \( \angle xpy = \alpha \) \((\alpha > \theta)\), which replaces \( t \) as shown in Figure 2(b). Case I - \( \triangle p_ixy \) is an invalid triangle (i.e. \( \triangle p_ixy \notin DT(\Omega^k_i) \)) since there is no Voronoi edge/face between the vertices \( x \) & \( y \). Case II - If \( \triangle p_ixy \in DT(\Omega^k_i) \), then the edge \( e_{xy} \) intersects the MA. Because \( \triangle p_ip_jp_k \) is a prime-triangle and none of the edges intersects the MA by Lemma 4.2. Therefore \( \triangle p_ixy \notin M \), which is a contradiction.

Let us suppose that there are more than one triangles. Let \( \triangle p_ip_jx \ & \triangle p_ip_kx \) be the triangles with angles \( \angle xp_j=\gamma \ & \angle xp_k=\beta \) \((\gamma + \beta > \theta)\) respectively, which replace \( t \) as shown in Figure 2(c). Case I - \( \triangle p_ip_jx \) is an invalid triangle as there is no Voronoi edge/face between the vertices \( p_i \ & x \). Case II - Otherwise, the edge \( e_{pi,x} \) intersects the MA (by Lemma 4.2 none of the edges of \( t \) intersect the MA). This is a contradiction.

4.4. Method summary

The diagram of the proposed method is shown in Figure 3. Let \( p_i \) be the reference point. To explain the algorithm visually, we have considered five points as the local neighborhood of \( p_i \) (i.e. \( k = 5 \)). The algorithm starts with computing the local neighborhood of \( p_i \) as shown in Figure 3(a). Subsequently, \( DT(\Omega^k_i) \) (DT of the neighborhood of \( p_i \)) is constructed in the second stage (Figure 3(b)). To extract the candidate set of triangles \( T(p_i) \in DT(\Omega^k_i) \), we select the triangles, which are incident with the reference point \( p_i \) as shown in Figure 3(c) (a set of triangles formed by the 1-ring neighborhood of \( p_i \in DT(\Omega^k_i) \) such that \( p_i \) is the central vertex). Figure 3(d) shows the rejected triangles from \( DT(\Omega^k_i) \). The rejected triangles are a set of triangles that do not incident with the reference point \( p_i \). Therefore, this set of triangles do not be part of the final mesh as a contribution of the reference point \( p_i \). In the next step, we sort \( T(p_i) \) in the ascending order of
their circum-radii. Subsequently, we select the prime-triangles iteratively from $T(p_i) \setminus t_{\text{prime}}(p_i)$ (Figure 3(e)) and added to the set $t_{\text{prime}}(p_i)$ if it does not violate the manifold and non-overlapping constraints. Figure 3(f) depicts only four triangles that are marked and the remaining triangles overlap. In the end, the set $t_{\text{prime}}(p_i)$ contains the triangles, which contribute to the final mesh for the reference point $p_i$.

Figure 4 shows the merging of the sets of triangles computed for different points. Let us suppose that the algorithm finishes the processing of $i^{th}$ reference point. Figure 4(a) shows the set of triangles selected for $i^{th}$ reference point. At $(i+1)^{th}$ reference point (shown in green color in Figure 4(b)), there are two triangles already selected (shown in red color in Figure 4(b)). These triangles are selected by one of its neighbors (the point above the reference point in Figure 4(b)). Hence, for the current reference point, only the remaining triangles are computed as shown in Figure 4(c)). This process continues until all the points are processed. Therefore, the merging process is not an extra computation rather it is happening automatically when the neighboring points are processed. The pseudo-code of the proposed method is presented in Algorithm 1 which is termed as Prime-Triangle based Surface Reconstructor (PTSR).

4.4.1. Optimal value of $k$ (Optimal local neighbourhood)

The topology and geometry of the final mesh rely on the selection of the local neighbourhood of the point $p_i$. For a water-tight surface, the optimal result is obtained when the 1-ring neighborhood of $p_i \in DT(\Omega_k^i)$ is the same as the 1-ring neighborhood of $p_i \in DT(S)$. Hence, we increase the value of $k$ until the 1-ring neighborhood of $p_i \in DT(\Omega_k^i)$ becomes the same as that of $p_i \in DT(S)$ and it will be invariant for the further increase of $k$ because of the nearest-neighbor-property of DT O’Rourke (1994). Therefore, the optimal 1-ring neighbourhood will not change with the further increase of $k$.

For surfaces with boundaries, a small value of $k$ results pseudo-holes, as well as a very large value of $k$ covers actual holes in the final mesh and destroys the topological information of the object. In general, the value of $k$ depends on the distribution of the point set. i.e. a small value of $k$ is sufficient for a uniformly distributed point set and the value of $k$ increases as the degree of non-uniformity increases. In Results and discussions section, we have shown the output meshes of the same input point set with different values of $k$.

Lemma 4.4. For any point set $S$, there exists always a value of $k$ for which the local neighbourhood of a point $p_i \in DT(\Omega_k^i)$ includes the local neighbourhood of the point $p_i \in DT(S)$.

Proof. Let $N_l(p_i) \& N_g(p_i)$ be the sets of points in the local neighborhood of a point $p_i \in DT(\Omega_k^i) \& p_i \in DT(S)$ respectively. Let us suppose for the current value of $k$, $N_l(p_i) \not\supset N_g(p_i)$. i.e., there are some points $p_j \in N_g(p_i)$ such that $\|p_j p_i\|$ greater than the $k^{th}$ minimum distance in $S$. Then increase the value of $k$ by $\delta (k = k + \delta)$. Since the position of the points are fixed in $S$, $\exists \delta$ for which $N_l(p_i) \supset N_g(p_i)$.

4.4.2. Assurance of the mesh quality

Our method selects non-overlapping prime-triangles for every point and these triangles constitute the required mesh representation of the target surface. The selection of triangles with smaller circum-radius captures the geometrical proximity of the surface Peethambaran and Muthuganapathy (2015). Therefore, the prime-triangles always results best possible triangles, which preserve the geometrical and topological properties of the surface. Furthermore, the non-overlapping constraint ensures that the mesh is free from fold-over triangles. As a result, the selection of non-overlapping prime-triangles provides the best-possible triangle quality from the available set of triangles.

Every prime-triangle is examined to verify whether it is complying with the manifold constraint. This constraint is enforced at two levels. The first level screening ensures the inclusion of the current prime-triangle is not introducing non-manifold edges in the set $t_{\text{prime}}(p_i)$ while the second level ensures the same in the sets $t_{\text{prime}}(p_j)$ and $t_{\text{prime}}(p_k)$, where $p_j$ and $p_k$ are the neighboring vertices of the prime-triangle incident with the reference point $p_i$. If it complies at both levels of manifold constraint, then the triangle is added to the sets $t_{\text{prime}}(p_i)$, $t_{\text{prime}}(p_j)$, and $t_{\text{prime}}(p_k)$. When the algorithm terminates, the collection of triangles from the sets associated with all the points forms the required triangular mesh. For instance, let $\Delta_{abc}$ is the current prime-triangle under consideration for a reference point $a$. The first level test ensures
that the inclusion of the $\triangle_{abc}$ is not introducing non-manifold edges in the set $t_{prime}(a)$. Similarly, the second level test ensures the same in the sets $t_{prime}(b)$ and $t_{prime}(c)$. If both conditions are satisfied, then the current prime-triangle is added to the sets $t_{prime}(a)$, $t_{prime}(b)$, and $t_{prime}(c)$.

The final mesh $M$ generated by our method from an $\epsilon$-sampled point set is always a 2-manifold surface. The following lemma substantiate the claim.

**Lemma 4.5.** The proposed method guarantees a 2-manifold surface.

**Proof.** Lemma 4.3 guarantees that the selected prime-triangles are always on the surface. Lemma 4.4 ensures that for a point set $S$, there is a $k$, which defines the local neighbourhood of a point in such a way that the triangle-fans for each point $p_i$ is subset of $DT(\Omega_i^k)$. For each point (reference point), the method selects all the valid prime-triangles which constitute a triangle fan concerning the reference point. As a consequence, the method does not yield a pseudo-hole (artificial hole) in the final mesh. Furthermore, the manifold and non-overlapping triangle constraints make sure that the final mesh is free from non-manifold edges. Because of the scenarios where the occurrence of non-manifold edges always have some triangles which are overlapping with the already selected prime-triangles. Hence the mesh returned by the method is always a 2-manifold surface.

Further, the following lemma guarantees that the mesh generated by our algorithm does not depend on the starting point. i.e. the proposed algorithm always returns the same output irrespective of the starting point.

**Lemma 4.6.** Let $p_i$ be a reference point and $\triangle_{p_jp_k}$ be the selected prime-triangle with $\angle p_jp_kp_i = \theta \leq \angle p_jp_kp_i = \beta$. Then $\triangle_{p_jp_kp_i}$ will be also selected for the reference points $p_j$ & $p_k$.

**Proof.** Let us suppose that the above lemma is not true. Let $p_i$ be the current reference point and one or more triangles are covering the area spanned by $\beta$ with circum-radii less than that of $\triangle_{p_jp_k}$. Then at least one point must be in the circumcircle of $\triangle_{p_jp_kp_i}$ as shown in Figure 5(a). Then the points $p_i$, $p_j$, $y$ or $p_i$, $y$, $p_k$ will form a triangle with smaller circum-radius with respect to $p_i$ and it is introducing a new point in the 1-ring neighborhood of $p_i \in DT(\Omega_i^k)$. This is a contradiction. The same argument holds for the point $p_j$. 

### 4.5. Running time complexity

Algorithm starts with constructing a $Kd$-tree from the input point set. It requires a cost of $O(n \log n)$. Subsequently, the local neighbourhood of a fixed size $k$ for each point is computed with a cost of $O(n \log n)$. For each point, we compute the local $DT$ of its neighbourhood. From the computed $DT$, we select the non-overlapping prime-triangles after projecting on to a plane. Subsequently, the non-overlapping prime-triangles undergo the manifold test. All these steps can be done in a constant time ($O(1)$). As a result, the total complexity of the algorithm is upper bounded by $O(n \log n)$. Figure 5(b) depicts that the running time of algorithm grows 'Linearithmic' with respect to the size of the input. This growth rate portrays that our algorithm is efficient and suitable for the large point sets compared to the global Delaunay-based methods.
Algorithm 1 PTSR(S)

Input: Input point set $S \in \mathbb{R}^3$.
Output: ReconstructedSurface $R$.
1: for each point $p_i \in S$ do
2:    Compute the local neighbourhood of $p_i$, $\Omega^k_i$.
3:    Construct Delaunay triangulation, $DT(\Omega^k_i)$.
4:    Collect the triangles incident with $p_i$, $T(p_i)$ (1-ring neighborhood of $p_i \in DT(\Omega^k_i)$).
5:    Sort the triangles, $T(p_i)$ in the ascending order of their circum-radii.
6: for each triangle $t_j \in T(p_i)$ do
7:    if $t_j$ is a prime-triangle complying with the non-overlapping triangle and manifold constraints
8:        $R = R \cup t_j$.
9:    end if
10: end for
11: end for
12: return $R$

5. Results & discussions

Our algorithm (Algorithm 1) is implemented in C++ with CGAL [Consortium et al. (1996)] (Version: 4.6) libraries and visualized in JavaView. To see the capability of the proposed algorithm, we have experimented with several different kinds of models, which are collected from Aim@Shape and Stanford 3D scans repositories. The input models have varying attributes like fine level details, sharp features, non-uniform sampling densities, large point sets. In this section, we show the effects of different neighbourhood sizes on the proposed method and measure the algorithm’s robustness against down-sampling. Furthermore, the performance of our method is evaluated on large point sets. We also show the effectiveness of our method compared to the state-of-the-art methods.

On top of that, we have included the quantitative analysis to show the comparison between the state-of-the-art methods and the proposed method in terms of the number of non-manifold edges (3rd column in Table 1) and mesh quality (4th column in Table 1) in the reconstructed mesh. Furthermore, the Hausdorff distance (5th column in Table 1) is computed between original model and corresponding reconstructed surface to show that the proposed algorithm produces a triangulated mesh that is quite close to the original model.

![Figure 6: The above figure shows the outcomes of the proposed method with different values of $k$. Figure (a) shows a point set ($n = 24K$). Figure (b)-(f) show the reconstructed surfaces corresponding to the different values of $k$. With $k = 6$, the proposed method produces too many holes. The optimal result is produced with $k = 24$. As it can be seen from Figure (e) and (f), $k = 256$ and $k = 512$ are filling all holes.](image-url)

5.1. Effects of different neighbourhood

The sampling density of the input point set and the value of $k$ have a vital role in the proposed algorithm. In this experiment, we have considered the scanned angel model ($n = 24K$) with a moderately non-uniform sampling as the input point set and different meshes are reconstructed with different values of $k$. As it
can be seen from Figure 6, the small value of $k$ produces several pseudo-holes (artificial holes) and bigger values of $k$ lead to the filling of holes. With a small value of $k$, the local DT does not find the points in the neighbour disk and creates holes. Moreover, a large value of $k$ manages to get more data in the local DT and closes holes.

5.2. Robustness against down-sampling

The robustness of our method against the down-sampling is displayed in Figure 7. For this experiment, we have chosen a large 3D scanned Hand model ($n = 920K$). As shown in Figure 7, we have five different meshes with different numbers of vertices of the same model. These meshes are reconstructed using the proposed method with down-sampled versions of original Hand model using a fixed neighbourhood size ($k = 32$). We used blue curves to mark boundaries (holes) and the area around the thumb is highlighted in a rectangular box for the comparison purpose.

As it can be seen that density of blue curves are decreasing with the density of the point set because the distance between the neighboring points is increasing. Consequently, the distance between two points directly opposite to each other on a hole boundary is less than or equal to the distance between a point and its farthest neighbour under the given value of $k$. As it can be seen from Figure 7, the holes are reducing around the fingers as the sampling density is decreasing.

5.3. Robustness against large input point sets

The concept of local DT helps the algorithm to deal with large point sets effectively. Figure 8 show the outcome of the proposed method with large and highly non-uniform input point sets. As it can be seen from the figure, our method can easily handle millions of points and is able to reconstruct and high fidelity surface with fine levels of features. Figure 8(b) shows the outcome of the proposed method from a highly non-uniform input point set, which is captured by a 3D scanner. Figure 8(e) shows that the proposed method is not only capable of handling the large number of vertices but also able to reconstruct fine details (texts on the wall and small features) in the geometry.

Additionally, Figure 7 and 11 also depict the robustness of our method against large point set as these models have 0.92M, and 0.62M vertices respectively.

5.4. Comparison with the state-of-the-art methods

To evaluate the competency of our algorithm, we perform both quantitative and visual comparisons with the state-of-the-art methods. For quantitative analysis, we use three different metrics, $e_{nm}$, $Q$, and Hausdorff distance (HD) to show the efficiency of the different algorithms (please refer to Table 1). The metric $e_{nm}$ denote the number of non-manifold edges in a mesh. The second metric demonstrate the measure of quality of the mesh denoted by $Q$ to evaluate the shape of triangles produced by a surface reconstruction algorithm. In a triangulated mesh, the presence of skinny triangles (a triangle with the area close to zero) makes the mesh processing difficult in many engineering applications. Nevertheless, this is nearly impossible in many
cases due to the imperfection of point sets like noise, non-uniformity, sparsity, missing points etc. Hence, it is very important to quantify the close similarity of a triangle with the equilateral triangle.

Therefore, the measure of the quality of a mesh ($Q$) is defined as the ratio of the circum-radius ($r_i$) to minimum-edge-length ($e_i$) of a triangle Yadav et al. (2019):

$$Q = \frac{1}{n_t} \sum_{i=0}^{n_t-1} q_i, \text{ where } q_i = \frac{r_i}{e_i},$$

where $n_t$ is the number of triangles in the reconstructed mesh. The summary of the quantitative analysis is given in Table 1. The third metric, Hausdorff distance (HD) measures the closeness between the original model and reconstructed mesh. The small value of Hausdorff distance indicates the high fidelity surface reconstruction. By using the Hausdorff distance, we show that the concept of the prime-triangle and local DT help to reconstruct a high fidelity surface.

We use the following state-of-the-art methods, PC (Amenta et al. (2001)), RC (Dey and Goswami (2006)), SRST (Methirumangalath et al. (2017)), SP (Kazhdan and Hoppe (2013)), APSS (Guennebaud and Gross (2007)), and RIMLS (Öztireli et al. (2009)) for the comparison exercise. PC, RC and SRST are explicit methods, while RIMLS, APSS and SP are implicit methods. PC, RC and SP are designed for water-tight surfaces whereas SRST, APSS and RIMLS can handle surfaces with or without boundaries. Furthermore, all these methods require human interaction in the form of the parameter(s) tuning to produce the optimal result. For every parametric method, we use the optimal results by tuning the corresponding parameter(s).

For the experimentation, we use several large and non-uniform 3D scanned data with boundaries and these data are used for comparison with RIMLS, APSS, and SRST methods because of their abilities to handle boundaries.

Figure 8: Outcomes with big data: The above experiment shows the effectiveness of the proposed method with big real data. The number of vertices are mentioned below each model, where M stands for million. Figure (b) shows the outcome with a highly non-uniform 3D scanned data. Figure (e) shows the effectiveness of PTSR in terms of the big data handling and small levels of features reconstruction.

Figure 9 shows that the proposed method reconstructs a high fidelity surface without changing the geometrical information. As it can be seen from Figure 9, SP and RIMLS are modifying small details. Simultaneously, these methods are changing the number of vertices in the corresponding reconstructed surfaces, which is good in certain applications e.g. animation but not recommended for several CAD/engineering related applications. On the contrary, our method reconstructs the mesh without changing the number of vertices with better mesh quality compared to RIMLS, and SP. Furthermore, the surface captured by our method is very close to the original surface (please refer to Table 1).
Figure 9: The above figure illustrates that the proposed method is capable of reconstructing the small levels of details (e.g., the veins of the leaf near the base) without modifying the geometrical information (number of vertices in the reconstructed surface is same as that in input point set ($n = 202K$)). The methods RC, SP [Kazhdan and Hoppe 2013], and RIMLS [Öztireli et al. 2009] are modifying the number of vertices ($n = 190K$, $n = 260K$, and $n = 828K$ respectively) and also losing the small levels of details during reconstruction.

Figure 10: Sharp features and mesh quality: Fandisk model ($n = 7k$) is free from noise and has sharp and cylindrical features. The black curve shows the sharp edge information in the smooth geometry and is detected using the dihedral angle $\theta = 20^\circ$. The magnified views are showing the performance of the proposed method against state-of-the-art methods in terms of mesh quality and feature preservation. As it can be seen, the proposed method not only better preserves features but also creates better triangles compared to the mentioned methods.

Figure 11: Non-uniform data with boundaries: Figure (a) shows a highly non-uniform disk brake scanned model ($n = 624K$). SRST is producing several non-manifold edges (see Table 1) along with several triangles inside the holes. From Figure 11(c) and Table 1, it is evident that our method is producing a high fidelity meshes without non-manifold edges and the value of $Q$ suggests an optimal mesh quality. On top of that, our method is capable to reconstruct holes and sharp features effectively. Table 1 shows the reconstructed mesh is close to the original model compared to the output produced by SRST.

Figure 12 shows the effectiveness of the proposed method in terms of topological and sharp features preservation during the reconstruction. The input point set is taken from a mesh by removing the connectivity information. Therefore, we are aware about the topology of the input data. As it can be seen from the figure, PC, RC, and SRST are generating additional triangles, which change the topology. The sharp edges have several artefacts and are blurred in case of RIMLS and SP. On the contrary, our method is reconstructing the sharp features without changing the topological and geometrical information. Table 1 shows that our method captures the properties of the model with good quality mesh and the reconstructed
sharp edges along with several artefacts. RIMLS and SP introduce blurring across the topological information. Figure 12: An example to show the effectiveness of the proposed method w.r.t. sharp features and topological information compared to the state-of-the-art methods. Figure (a) shows input point set (n = 34K). In Figure (b) - (d), PC, RC, and SRST produce additional triangles to change the topological information. Furthermore, RIMLS and SP introduce blurring across the sharp edges along with several artefacts.

The Hausdorff distance (RMS error) is computed for quantitative comparison between the above methods. In visual comparison, it is clear that SRST handles the features effectively (Figure 13(c)). However, it introduced additional triangles inside the holes which makes the reconstructed mesh away from the original model compared to our result (please refer to the fifth row in Table 1). It is also yielding many non-manifold edges in the mesh. Though RIMLS generating a manifold mesh (Table 1), the quality of the mesh produced by RIMLS (Figure 13(b)) is not as good as the output by PTRS. On the other hand, the quality of the mesh generated by PTSR is better compared to SRST & RIMLS and PTSR is able to reconstruct holes and features as shown in Figure 13(d). On top of that, Figure 13(e-h) shows the performance of these methods with the laser scanned models with boundaries. In visual comparison, it is clear that SRST (Figure 13(g)), RIMLS (Figure 13(f)), and PTSR (Figure 13(h)) handle the features decently. RIMLS generating a manifold mesh. However the quality of the mesh produced by the RIMLS is not as good as the output by PTSR and the output of SRST is non-manifold.

The Hausdorff distance (RMS error) is computed for quantitative comparison between the above mentioned state-of-the-art methods and the proposed method with the ground-truth (original model) and reconstructed surface (please refer to Table 1). We used Meshlab v1.3.2 to compute the Hausdorff distance (the one-sided version). Hausdorff distance is a measure of the difference between two different descriptions of the same 3D object. The zero value indicates that the two objects are the same and the value increases as the differences between the objects increase. In these experiments, we use the models Fandisk (Figure 10), Block (Figure 12), Disk (Figure 11) and David (Figure 13(a-d)). Additionally, the term n_b from Table 1 represents the density of boundaries in the reconstructed surfaces.

Table 1: The quantitative comparison. e_{nm} - # non-manifold edges, n_b - # vertices on boundaries, Q - quality of the mesh, HD - Hausdorff distance.

<table>
<thead>
<tr>
<th>Models</th>
<th>Methods</th>
<th>e_{nm}</th>
<th>n_b</th>
<th>Q</th>
<th>HD</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box</td>
<td>RIMLS</td>
<td>271</td>
<td>207</td>
<td>1.82</td>
<td>19</td>
<td>(25.0, 0.001, 15, 0.75, 3, 400)</td>
</tr>
<tr>
<td>Block</td>
<td>RC</td>
<td>300</td>
<td>0</td>
<td>1.41</td>
<td>45</td>
<td>(0.4, 0.3, 0.6)</td>
</tr>
<tr>
<td>SRST</td>
<td>SP</td>
<td>142</td>
<td>0</td>
<td>3.06</td>
<td>20.9</td>
<td>(8.1, 0.4)</td>
</tr>
<tr>
<td>Disk</td>
<td>SRST</td>
<td>52</td>
<td>0</td>
<td>1.85</td>
<td>11.6</td>
<td>(0.4, 0.3, 0.6)</td>
</tr>
<tr>
<td>David</td>
<td>RIMLS</td>
<td>1091</td>
<td>0</td>
<td>2.06</td>
<td>5.47</td>
<td>(8.1, 0.4)</td>
</tr>
<tr>
<td>SRST</td>
<td>SP</td>
<td>128</td>
<td>0</td>
<td>4.99</td>
<td>9.9</td>
<td>(0.4, 0.3, 0.6)</td>
</tr>
<tr>
<td>Figure 13(a-d)</td>
<td>PTSR</td>
<td>423</td>
<td>0</td>
<td>1.5</td>
<td>1.75</td>
<td>(25.0, 0.001, 15, 0.75, 3, 400)</td>
</tr>
<tr>
<td>Figure 13(b)</td>
<td>SRST</td>
<td>126</td>
<td>0</td>
<td>0.64</td>
<td>0.09</td>
<td>(0.4, 0.3, 0.6)</td>
</tr>
<tr>
<td>Figure 13(c)</td>
<td>SRST</td>
<td>176</td>
<td>0</td>
<td>8.45</td>
<td>4.44</td>
<td>(8.1, 0.4)</td>
</tr>
<tr>
<td>Figure 13(d)</td>
<td>SRST</td>
<td>11.6</td>
<td>0</td>
<td>1.85</td>
<td>11.6</td>
<td>(0.4, 0.3, 0.6)</td>
</tr>
<tr>
<td>Figure 13(e)</td>
<td>SRST</td>
<td>1.93</td>
<td>0</td>
<td>1.85</td>
<td>11.6</td>
<td>(0.4, 0.3, 0.6)</td>
</tr>
</tbody>
</table>

As a summary of comparison, we have shown that the proposed method is capable of producing an
Figure 13: Kinect data: The above figure shows that our method is able to reconstruct the topological information accurately (3 holes) from Kinect data (n = 52K) similar to SRST and better compared to RIMLS. Pierrot model (Figure (e)-(g)) is captured using a 3D laser scanner and has several boundaries along with noise components. The proposed method is able to reconstruct the mesh with better mesh quality along with boundaries and features.

accurate mesh in terms of boundaries/holes from a non-uniform point set. The quantitative analysis shows that our method produces a high fidelity and better mesh quality, along with zero non-manifold edges compared to the state-of-the-art methods.

6. Conclusion, limitations and future work

In this paper, we presented a simple and single-parametric surface reconstruction algorithm, which takes a point set as input and produces a high fidelity triangulated mesh. The proposed method employed the concept of local Delaunay triangulation, which makes the method memory efficient and helps to reduce the running time complexity of the algorithm to $O(n \log n)$ for a fixed $k$. As the local DT is fast and memory efficient, our method is capable of handling large point sets effectively compared to the most of the state-of-the-art methods. On top of that, the concept of local Delaunay helps the proposed algorithm to retain sharp features during surface reconstruction. We use the projection on a locally least-square plane to select the non-overlapping prime-triangles for a point that make sure the manifold structure as well as the non-existence of fold-over triangles in the output mesh. From the usability point of view, our method is easy to use compared to several other methods as it needs only one input parameter from the user side. We have also provided the theoretical guarantee of the algorithm with the assumption of $\epsilon$-sampling model. The $\epsilon$-sampling assumption is only used for the theoretical proof and our method performs equally good on point sets, which are not following $\epsilon$-sampling criterion. Additionally, this method is equally effective in geometries with boundaries unlike some of the state-of-the-art methods. To prove our claims, we have experimented with several benchmark models and results are shown in the experimentation section.

Even though our algorithm captures the geometrical and topological properties of a model from a real scanned data (without smoothing), it fails in the presence of outliers. Since it is a DT based algorithm which interpolates every point in the point set. Therefore, the algorithm considers the outliers as surface points, which are actually away from the surface, and is not able to capture the topology of the model. Furthermore, the proposed method does not introduce any non-manifold elements (edges & vertices) or fold-over triangles in the final mesh $M$ with various kinds of input point sets (as shown in the results section). However, for a highly irregular surface or in the presence of outliers and a high level noise, there is a chance of creating a few non-manifold vertices.

As a future work, we would like to extend the algorithm to handle point sets with noise and outliers. Also, we did not discuss the relation between $k$ and the distribution of point set. In the future work, we would like derive a relation between the local neighbourhood size and the corresponding sampling distribution of the point set. In the current method, we enforced only the edge manifold constraint. We would like to add a vertex manifold constraint in the future work.

References
