

Reconstruction of Water-tight Surfaces through Delaunay Sculpting

Submission ID: P000088

Abstract

Given a finite set of points $S \subseteq \mathbb{R}^2$, we define a proximity graph called as shape-hull graph ($SHG(S)$) that contains all Gabriel edges and a few non-Gabriel edges of Delaunay triangulation of S . For any S , $SHG(S)$ is topologically regular with its boundary (referred to as shape-hull (SH)) homeomorphic to a simple closed curve. We introduce the concept of divergent concavity for simple, closed, planar curves based on the alignment of curves in concave portions and discuss various measures to characterize curves having divergent concavity. Under sufficiently dense sampling, we prove that $SH(S)$, where S is sampled from a divergent concave curve Σ_D , represents a piece-wise linear approximation of Σ_D . We extend this result to provide a sculpting algorithm for closed surface reconstruction from a set of raw samples. The surface is constructed through a repeated elimination of Delaunay tetrahedra subjected to circumcenter and topological constraints. Theoretically, we justify our algorithm by establishing a topological guarantee on the 3D shape-hull with the help of topological rules. We demonstrate the effectiveness of our approach with experimental results on models with sharp features and sparsely distributed point clouds. Compared to existing sculpting approaches for surface reconstruction that require either a parameter tuning or several stages, our approach is simple, non-parametric, single stage and reconstructs topologically correct piece-wise linear approximation for divergent concave surfaces.

Keywords: Point sets, Surface reconstruction, Shape reconstruction, Delaunay triangulation, 3D modeling

1. Introduction

Given a finite set of points $S \subset \mathbb{R}^3$, sampled from the surface P of a real world object, the task of reconstructing a model of P from S is referred to as surface reconstruction problem. Recently, the problem has gained a lot of attention both in computer graphics and computational geometry, especially due to the proliferation of laser scanners and their wide applications in areas such as reverse engineering, product design [1], cultural heritage [2], and cartography [3], among others. Apart from laser scanners, input point samples are also obtained from photogrammetry technique or mathematical functions. Different approaches have been taken to solve the problem depending upon the input data type or output reconstructed surface [4]. Input data may be a registered point cloud equipped with or without normals and the reconstructed model may be a parametric or implicit surface or a triangulated surface mesh.

Surface reconstruction is an ill-defined problem [5] as there are several surfaces that might fulfil the topological and geometric properties of the original surface. Therefore, the main challenge for any surface reconstruction approach lies in mathematically defining a surface that best captures the geometric and topological properties of the original surface. It is well established that proximity information and neighborhood relationships of surface samples has a major role in defining the geometric shape of any surface. Aimed at an approach that exploits these two factors, we define a proximity graph called as shape-

hull graph (SHG) which is essentially a restricted Delaunay graph that consists of Gabriel edges [6] and a few non-Gabriel edges. The boundary of SHG , referred to as shape-hull (SH), represents the polygonal reconstruction of a surface from its sample.

In computational geometry, sculpting refers to the process of creating shapes or surfaces through a repeated elimination of tetrahedra from an initial tetrahedral mesh. We propose a sculpting algorithm for closed surface reconstruction based on three dimensional shape-hull graph, taking a registered point cloud without normals (this is referred to as raw point cloud data) as input and producing a triangulated surface mesh as output. The approach combines a local measure (circumradius) and a global information (largest circumradius). As a consequence, no user driven parameter tuning is required for the whole reconstruction process. We observe that non-parametric approach is a reasonable trade-off in many situations especially when considering that the parameter is critical to the output of many established algorithms and these algorithms provide only a limited guidance on selecting the optimal parameter. Moreover, we also provide a guarantee on the topologically correct reconstruction under the theoretical framework based on divergent concave curves, ϵ -sampling [7] and topological rules.

2. Related Work

Most of the current surface reconstruction methods fall into two major categories: one class of surface reconstruc-

tion techniques use implicit functions to represent surfaces and the other methods use Delaunay/Voronoi based triangulated mesh to represent the surface. One of the early work in implicit surface reconstruction is the tangent plane estimation method by Hoppe et.al. [8]. The method uses principal component analysis to estimate the normals and then employ neighborhood graph search to unify the inside/outside directions followed by reconstruction of surface as the zero-level set of an implicit signed distance function with the help of marching cube method. Ohtake et.al [9] present a multilevel partition of unity (MPU) approach to reconstruction which basically tries to locally fit the surface via quadratic functions blended by weighting function (partition of unity). In [10], Kazhdan describes a reconstruction algorithm which uses Stokes theorem to compute an indicator function of the surface. Other related approaches compute indicator function of the underlying shape by solving poisson equations [11], wavelets [12] or generalized eigenvalue problem [13]. Approaches like moving least square methods generally define the surface as an invariant set of projection operator which in turn is computed by numerical optimization on a locally constructed implicit function [14, 15]. Another method to implicit surface reconstruction is the computation of radial basis functions (RBF) [16] or anisotropic basis functions to represent the surface [17]. It is to be noted that most of these techniques use acquired or estimated point normals to facilitate the reconstruction process.

In computational geometry, several Delaunay/Voronoi based surface reconstruction algorithms have been proposed [5, 18], most of which provide provable theoretical guarantees on the quality of reconstructed output under specific sampling model. Underlying intuition is that, when the point sampling is dense enough and free of noises and outliers, the neighboring points on the surface will also be neighbors in the space and hence can be captured with the help of Delaunay triangulations or Voronoi diagrams. Early work in this direction include sculpture algorithm by Boissonnat [19] and three dimensional α -shapes by Edelsbrunner et.al [20]. Thereafter, many researchers approached reconstruction problem from a geometric point of view, focusing on topology reconstruction and providing guarantees on the quality of output under dense sampling model such as ϵ -samples [7]. The model of Delaunay/Voronoi together with ϵ -sampling (or a related sampling model) has been extensively studied and many notable work have been proposed that include crust [7], powercrust [1], r-regular shape reconstruction [21] and cocone family of algorithms [22, 23, 24, 25].

Few other Delaunay based approaches assure a topological equivalence between the reconstructed surface and the original surface S . Gopi et.al. [26] describe a tangent plane based method that is guaranteed to construct a model homeomorphic to the original surface when a locally uniform sampling condition holds. Edelsbrunner’s WRAP [27] algorithm relies on the concepts of flow and stable manifolds where the *flow relation* has been formu-

lated on the set of Delaunay simplices. An extension of WRAP by Ramos et.al [28] uses ϵ -sampling model to provide guarantees such as tubular neighborhood and homotopy equivalences. Gezahegne [29] develop a hybrid approach to surface reconstruction based on sculpture and cocone algorithms and establish a topological guarantee for reconstruction of a piece-wise linear water-tight approximation. Approaches such as constriction [30], greedy algorithm [31], ball pivoting [32], regular interpolant method [33], geometric convection [34], hybrid sculpting [35], Delaunay based region growing [36] do not provide theoretical guarantees but found to perform reasonably well for a range of point sets. Moreover, some of them are rather simplistic in nature and stand out in terms of their efficiency. In general, Delaunay/Voronoi based approaches are found to produce good approximations for dense uniformly sampled point cloud, work on input scans without point normals and often compute output meshes with a complexity in the order of input point set size.

The main contributions of this paper include:

- **Divergent concave curves:** We introduce the concept of divergent concavity for simple, closed and planar curves. We extend the concept of concavity of a closed curve and define pseudo-concavity. Pseudo-concavity along with other measures such as medial balls, outer medial axis and bi-tangents are used to characterize divergent concave curves.
- **Shape-hull graph (SHG):** We define a proximity graph called as shape-hull graph which is capable of capturing the proximity of sample points. Further, we show that, for any set of points S sampled from a divergent concave curve, the boundary of SHG is regular and represents a polygonal reconstruction of the curve.
- **3D Sculpting algorithm:** We extend the concept of SHG to three dimensions and propose a simple and non-parametric sculpting algorithm for closed surface reconstruction. Using the topological constraints, we show that the reconstructed surface is always homeomorphic to a sphere.

3. Preliminaries

We consider a planar simple closed curve Σ . Let S be a set of n points sampled from Σ and $conv(S)$ be its convex hull. Let $int(conv(S))$ and $\partial conv(S)$ denote the interior and boundary of the convex hull of S respectively. Let $d(x, y) = \|x - y\|$, denotes the Euclidean distance between two points $x, y \in S$.

Definition 1. *Voronoi cell (V_p):*

Let $x \in \mathbb{R}^2$. A Voronoi cell of $p \in S$ is the set of all points as close to p as to any other point in S :

$V_p = \{x \in \mathbb{R}^2 \mid d(p, x) \leq d(q, x), \text{ where } p \neq q \text{ and } \forall q \in S\}$ [37].

Definition 2. *Voronoi diagram ($Vor(S)$):*

The set of all points that belong to more than one Voronoi cell (V_p) of S form the Voronoi diagram of S [37].

Definition 3. *Delaunay triangulation ($Del(S)$):*

The straight line dual graph of $Vor(S)$ results in a planar triangulation called as Delaunay triangulation of S , $Del(S)$. [37]

To define shape-hull graph, we use the concept of simplicial complex. A k -simplex is the non-degenerate convex hull of $k + 1$ geometrically distinct points, $v_0, v_1, \dots, v_k \in \mathbb{R}^d$ where $k \leq d$ [38] (Definition 4).

Definition 4. *k -simplex (σ_k):*

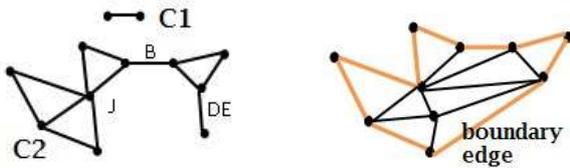
It is the intersection of all convex sets containing (v_0, v_1, \dots, v_k) i.e., $\sigma_k = \{x \in \mathbb{R}^d \mid x = \sum_{i=0}^k \alpha_i v^i \text{ with } \alpha_i > 0 \text{ and } \sum_{i=0}^k \alpha_i = 1\}$

According to the Definition 4, vertex is 0-simplex, edge is 1-simplex, triangle is 2-simplex and tetrahedron is 3-simplex. The convex hull of any non-empty subset of the $(k + 1)$ points that defines a k -simplex is referred to as a **face** of that simplex. Like simplices, vertex is a 0-face, edge is a 1-face and so on. A $(k - 1)$ -faces of a k -simplex is called as a **facet**.

Definition 5. *Simplicial complex [38]:*

A simplicial complex, \mathcal{K} is a set containing finitely many simplices that satisfies the following two restrictions:

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



(a) Non-regular \mathcal{K}_2

(b) Regular \mathcal{K}_2

Figure 1: Examples of simplicial 2-complexes. Figure 1(a) consists of two components C_1 & C_2 . The constructs which violates the regularity of a simplicial 2-complex are shown in Figure 1(a) and denoted by the symbols: J-junction point, DE-Dangling edge and B-Bridge.

A two dimensional Delaunay triangulation is an example of simplicial complex. A simplicial k -complex, \mathcal{K}_k is a simplicial complex where the largest dimension of any simplex is equal to k . Edges which do not belong to any triangle in a \mathcal{K}_2 are either bridges, dangling edges or disconnected line segments (see Figure 1(a)). In a simplicial

2-complex, if one or more triangles are attached to any other k -simplex (where $k = 1$ or 2) through only one of its vertices, then that vertex is termed as a junction point (see Figure 1(a)).

Definition 6. *Regular simplicial 2-complex (\mathcal{RK}_2):* A simplicial 2-complex \mathcal{K}_2 is said to be regular if it satisfies the following conditions:

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

A regular simplicial 2-complex is a full dimensional simplicial complex in which every simplex is incident to a highest dimensional simplex (i.e. triangle in this case). An edge in \mathcal{RK}_2 is a boundary edge if it is incident to a single triangle.

Definition 7. *Boundary triangle:*

A triangle in \mathcal{RK}_2 is a boundary triangle if it is incident to at least one boundary edge. In Figure 1(b), triangles with orange edges represent boundary triangles.

The triangles in \mathcal{RK}_2 , which share all the three edges with other triangles are termed as interior triangles. Triangles with three black edges in Figure 1(b) are examples of an interior triangles.

Definition 8. *Thin triangle:*

A triangle in \mathcal{RK}_2 is said to be a thin triangle if its circumcenter lies strictly external to it. All other triangles are referred to as fat triangles.

A thin triangle is essentially an obtuse triangle and a fat triangle can be either an acute or a right angled triangle.

3.1. Divergent Concavity of Curves

A simple closed curve Σ bounds a region referred to as interior of Σ ($I(\Sigma)$), that lies to the left when travelled in counter clockwise direction along Σ . Jordan curve theorem establishes that a simple closed curve divides the plane into a well-defined interior ($I(\Sigma)$) and exterior ($\overline{I(\Sigma)}$).

Curve Σ is said to be **convex**, if the line segment between any two points on the curve falls in the interior, $I(\Sigma)$. Otherwise it is **concave**. The curvature κ at a point p of Σ is the rate of change of direction of the tangent line at p with respect to arc length s . An inflection point (IP) on the curve is a point where $\kappa = 0$ but $\kappa' \neq 0$ (Figure 2). Since reconstruction in the presence of concave portions is extremely difficult, we restrict our attention to concave curves. Concave portions of a curve is characterized by the sign of the local curvature κ . Concave portions exists between two inflection points and has a negative local curvature sign ($\kappa < 0$).

A bi-tangent (BT) to a curve Σ is a tangent line L that touches Σ at two distinct points. The points where

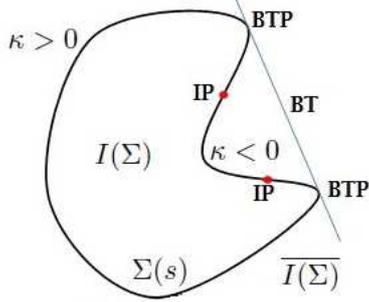


Figure 2: A curve with inflection points (IP), a concave portion (curve portion with $\kappa < 0$), Exterior bi-tangent (BT), and the bi-tangent points (BTP).

BT touches Σ is referred to as bi-tangent points (BTP). We consider only the bi-tangents lying completely in the exterior of the curve ($\overline{I(\Sigma)}$) for our discussion (i.e. BT refers to exterior bi-tangent). With these basic terminology, we introduce the definition of pseudo-concavity of Σ .

Definition 9. Pseudo-concavity:

The portion of Σ lying between two bi-tangent points having at least one sub-portion with $\kappa < 0$ is called as pseudo-concave portion of Σ , denoted by $\mathcal{C}(\Sigma)$.

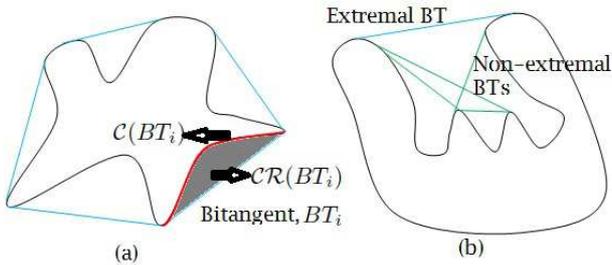


Figure 3: Illustration of pseudo-concave region ($\mathcal{CR}(BT_i)$, grey color region in Figure 3(a)), pseudo-concave portion ($\mathcal{C}(BT_i)$, red color curve portion shown in Figure 3(a)), extremal bi-tangent (blue color) and non-extremal bi-tangents (green color in Figure 3(b)).

Red colored portion of the curve in Figure 3(a) is an example of pseudo-concave portion. Here are some observations on pseudo-concavity of a curve (Figure 3).

1. Multiple pseudo-concave portions are possible for Σ .
2. A pseudo-concave portion $\mathcal{C}(\Sigma)$ always contains portions with $\kappa > 0$.
3. Every BT induces a $\mathcal{C}(\Sigma)$. The region bounded by BT and the corresponding $\mathcal{C}(\Sigma)$ constitutes the pseudo-concave region of BT , denoted by $\mathcal{CR}(BT)$ (grey colored region in Figure 3(a)).
4. There may exist some BT_i in $\mathcal{CR}(BT_j)$. Here, BT_i is referred to as non-extremal BT (green color bi-tangents in Figure 3(b)).

Medial axis of Σ is closure of the set of points in the plane which have two or more closest points in Σ [7]. Medial

axis also contains the centers of all osculating disks (empty disks tangent to Σ). A medial ball $B(c, r)$, centered at $c \in$ medial axis of Σ with radius r , is a maximal ball whose interior contains no points of Σ . For any Σ , there exists inner and outer medial axis. We restrict our attention to outer medial axis and the corresponding medial balls for defining divergent pseudo-concavity.

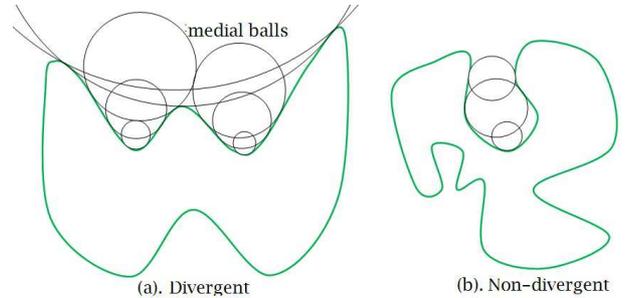


Figure 4: Illustration of divergent and non-divergent pseudo-concavities.

Definition 10. Divergent pseudo-concavity:

A $\mathcal{C}(\mathcal{BT})$ of Σ is said to be divergent, if the radii of medial balls, $B(c, r)$, r_i monotonically increases as it goes along the outer medial axis of $\mathcal{C}(\mathcal{BT})$ from one end to the extremal BT end.

An example of divergent pseudo-concavity is illustrated in Figure 4(a). For $\mathcal{C}(\mathcal{BT})$ having non-extremal BT s, medial axis may have branches that go separately to different $\mathcal{C}(\mathcal{BT})$ s of non-extremal BT s. The Definition 11 is valid in this case as well as the medial ball rolls only towards extremal BT end (Figure 4(a)).

Definition 11. Divergent concave curve(Σ_D):

A simple, closed planar curve Σ is said to be divergent concave if all its pseudo-concave portions ($\mathcal{C}(\mathcal{BT})$) are divergent.

Figures 4(a) & 4(b) illustrate examples of divergent and non-divergent curves respectively. In Figure 4(b), radii of medial balls continuously increase for some time, then decrease for a smaller interval of time and then again start increasing as it approaches the corresponding extremal bi-tangent. There are other type of non-divergent pseudo-concavity where the radii of medial balls monotonously decreases as it approaches the extremal bi-tangent.

In $Del(S)$ of a finite set of points S sampled from a Σ_D , an **external Delaunay triangle** is a triangle, $\Delta_x \in Del(S)$ which is partially or fully exposed to the exterior of original curve Σ_D . i.e. $\{\Delta_x\} \cap \{I(\Sigma)\} \subseteq \{\Delta_x\} \setminus \{\}$, where $\{\}$ is the null set. External Delaunay triangles exist only in the pseudo-concave regions.

We use ϵ -sampling [7] where the density depends on the local feature size. Local feature size at a point p on Σ , $LFS(p)$ is the distance from p to the closest point on the medial axis of Σ . This sampling is capable of quantifying

the local level of detail at each point on a smooth curve. Formally, ϵ -sampling is defined as follows:

Definition 12. ϵ -sampling [7]:

For a constant $\epsilon > 0$, Σ is said to be ϵ -sampled by a finite set of samples S , if

$$\forall p \in \Sigma, \exists s \in S \text{ such that } \|p - s\| \leq \epsilon LFS(p).$$

In ϵ -sampling, curve portion which encapsulates more details are sampled densely. Hence the concave portions of Σ_D are densely sampled. Dense sampling implies that, external Delaunay triangles are very thin triangles and their Voronoi vertices tends to approximate outer medial axis of $\mathcal{C}(BT_j)$. Consequently, circumcircles of these triangles approximate the medial balls. Under such a dense sampling assumption, we prove Lemma 3.1.

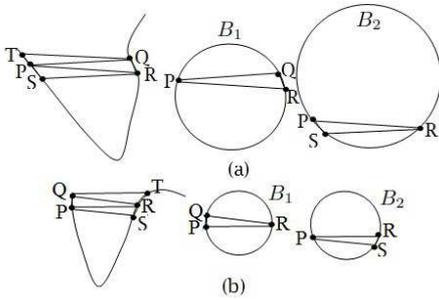


Figure 5: Illustration of two cases for Lemma 3.1.

Lemma 3.1. In $Del(S)$, where S is ϵ -sampled from a divergent concave curve, Σ_D , external Delaunay triangles in each $\mathcal{CR}(BT_j)$ are obtuse with their longest edge facing BT_j

Proof Assume the contrary. Two cases occur, i.e. first, the existence of obtuse external Delaunay triangles whose longest edge not facing BT_j and the second, existence of acute external Delaunay triangle. Under dense sampling assumption, second case contains isosceles triangles. We use Figure 5 to show our claim in both the cases. In both the cases, more than half portion of the ball B_1 lies below the edge (P, R) and the major portion of the ball, B_2 lies above (P, R) . If the medial ball, B_1 has a larger radius than B_2 , it cuts the curve somewhere below the edge (P, R) . As per the definition of medial ball, this is never possible and hence the radius of B_1 must be less than the radius of B_2 . Radius of $B_1 \leq$ radius of B_2 violates the divergent concavity of Σ_D and hence the assumed cases are never possible. Hence the lemma.

4. Shape-hull Graph

Armed with the preliminaries and definitions, now we formally define shape-hull graph. We denote the ball having radius r passing through two distinct points x, y by $B(r, x, y)$. Boundary of a simplicial complex, denoted by ∂ , consists of only boundary edges (boundary edge is defined in Section 3).

Definition 13. Shape-hull graph (SHG):

Shape-hull graph of S , $SHG(S)$ is a simplicial complex that consists of Delaunay edges (p, q) which satisfy either of the following properties:

1. $B(\frac{d(p,q)}{2}, p, q)$ does not contain any other point $l \in S \setminus \{p, q\}$.
2. If $B(\frac{d(p,q)}{2}, p, q)$ contains a point $l \in S \setminus \{p, q\}$ such that $\Delta pql \in Del(S)$, then either of the following is true.
 - (a) Circumcenter of Delaunay triangle, Δpql lies on or interior to $\partial SHG(S)$.
 - (b) All three vertices of Delaunay triangle, Δpql lie on $\partial SHG(S)$.

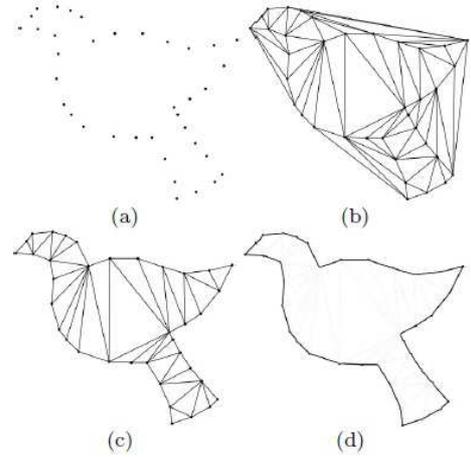


Figure 6: (a). Sample (b). Delaunay (c). Shape-hull graph and (d). Shape-hull.

Definition 13 implies that $SHG(S)$ contains all fat Delaunay triangles and a few thin Delaunay triangles. Consequently, $SHG(S) \subseteq Del(S)$. To represent polygonal curve of a given sample points, we extract the boundary of $SHG(S)$, which is referred to as shape-hull ($SH(S)$). Figures 6(c) & 6(d) show an example of SHG and its boundary (SH) respectively.

SHG captures interesting properties of S , such as proximity information of points and geometric shape of S for a range of point sets. A study on other proximity graphs such as Gabriel graph (GG), relative neighborhood graph (RNG) and minimal spanning tree (MST) along with $SHG(S)$, we found the relationship among them as $MST \subseteq RNG \subseteq GG \subseteq SHG(S) \subseteq Del(S)$. In Euclidean plane, this is a well established result [39] except the relation between $GG(S)$ and $SHG(S)$. We state the relation between GG and PSH in Lemma 4.1.

Lemma 4.1. For a finite set of points $S \subseteq R^2$, $GG(S) \subseteq SHG(S)$.

Proof By definition, Gabriel graph contains an edge (p, q) ($p, q \in S$), if and only if the circle passing through the

points p and q centered at the edge (p, q) is empty. Using the empty circle property, it has been shown that, $GG(S) \subseteq Del(S)$ [39]. This directly implies that $GG(S)$ contains all edges of $Del(S)$ except the longest edges of thin Delaunay triangles. On the other hand, from Definition 13, it directly follows that $SHG(S)$ consists of all Gabriel-edges (condition 1) and few non-Gabriel edges (condition 2). Further, $SHG(S) \subseteq Del(S)$ (Definition 13) and hence $GG(S) \subseteq SHG(S)$.

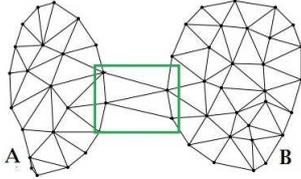


Figure 7: Illustration of Lemma 4.2.

Lemma 4.2. [REGULARITY LEMMA] *For a finite set of points $S \subseteq \mathbb{R}^2$, $SHG(S)$ is a regular simplicial complex.*

Proof We need to show two points. Each pair of points in $SHG(S)$ is connected by a path and $SHG(S)$ is free of junction points, dangling edges and bridges. To prove these points, we make use of a case (SHG) shown in Figure 7 where two groups of points, denoted by A and B are connected by two adjacent triangles shown in green box. For the sake of argument, we assume that the triangles in the green box are obtuse. In such a case, despite the two triangles in the green box contains non-Gabriel edges, the condition $ii(a)$ of Definition 13 (all the three vertices of both triangles are boundary vertices) makes sure that these triangles are never deleted from SHG . So from an extreme case like this, we can see that any pair of points are connected by a path in SHG . A similar argument and case hold good for the absence of bridges in SHG . Dangling edges are possible if two thin boundary triangles (whose circumcenter lie outside ∂SHG) share a common edge (Figure 8(d)). However, in this case, if one triangle gets removed, then all the three vertices of the other triangle becomes boundary vertices and hence will be retained in SHG (again due to condition $ii(a)$ of Definition 13). Similarly, due to the same condition of SHG definition, junction points (Figure 8 (a)) are also not possible in SHG . Hence the lemma.

Corollary 4.3. *For a finite set of points $S \subseteq \mathbb{R}^2$, $SH(S)$ is homeomorphic to a simple closed curve.*

Proof Assume the contrary, i.e. $SH(S)$ either contains more than one disconnected components or is an open curve. If it contains two (or more) disconnected components, it must have come from two disconnected components of its corresponding $SHG(S)$ and contradicts the regularity of $SHG(S)$ (Lemma 4.2). A similar argument

suffice for non-simple curves violating the regularity of $SHG(S)$ due to junction points. If $SH(S)$ is an open curve, then it contains two dangling edges on either end of the curve. According to the definition of SHG , such dangling edges are never considered as boundary edges. Further, dangling edges are never present in $SHG(S)$ (due to Lemma 4.2). Hence $SH(S)$ is always a simple closed curve. Hence the corollary.

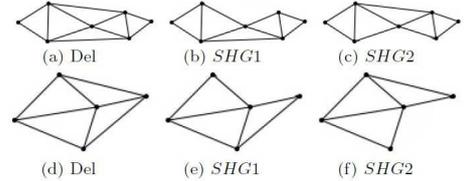


Figure 8: Illustration of different SHG s for certain point sets. Figures 8(a)-(c) show the case where two thin boundary triangles share a common interior point and Figures 8(d)-(f) show the case where two thin boundary triangles share a common edge.

A point set can have different shape-hull graphs as illustrated in Figure 8. This arises when two thin boundary triangles with circumcenter lying exterior to $\partial SHG(S)$, share a common edge or a common interior point in $Del(S)$. We call such triangles as candidate triangles. If one triangle gets removed, the other one will be retained in the shape-hull graph due to the regularity constraint. So, the total number of SHG for a point set can be expressed in terms of number of candidate triangles (t^*) it contains and it's trivial to observe that this number is $\Theta(2^{\frac{t^*}{2}})$.

It can be noted that, for divergent concave curves, the only region where the boundary to be detected from $Del(S)$ are pseudo-concave regions.

Lemma 4.4. *SH of an ϵ -sampled smooth divergent concave curve Σ_D contains only the edges between every pair of adjacent samples for $\epsilon < 1$.*

Proof For S , ϵ -sampled from a smooth curve and $\epsilon < 1$, $Del(S)$ contains an edge between every pair of adjacent samples (proved in Theorem 12, [7]). We already know that $SHG(S) \subseteq Del(S)$. For an ϵ -sample S of Σ_D , $SHG(S)$ does not contain an edge in any of its pseudo-concave regions (implied by Lemma 3.1) and hence SH constitutes the boundary of S . Further we need to show that an edge between adjacent samples never gets deleted in $SHG(S)$ and consequently in $SH(S)$. This statement is an immediate consequence of regularity lemma for $SHG(S)$ (Lemma 4.2) and the second condition of Definition 13. Since $SH(S)$ consists only of boundary edges, all remaining edges between non-adjacent samples (edges lying interior to Σ_D) will be omitted in $SH(S)$. Hence $SH(S)$ contains only the edges between every pair of adjacent samples.

As far as geometric guarantees of $SH(S)$ is concerned, Theorem 4.5 put forward by Amenta et.al. [7] is equally applicable to $SH(S)$ under the theoretical frame work based on ϵ -sampling, divergent concavity of curves and Delaunay triangulation.

THEOREM 4.5. [7] *The distance from a point p on an ϵ -sampled smooth curve F to some point on the polygonal reconstruction of the samples is at most $(\frac{\epsilon^2}{2})LFS(p)$.*

5. Surface Reconstruction

In this section, we present a simple sculpting algorithm to compute shape-hull of surface samples in \mathbb{R}^3 . In 3D, $SHG(S)$ consists of Gabriel tetrahedra and few non-Gabriel tetrahedra whose circumcenter (CC) lies inside $\partial SHG(S)$ or whose deletion violates the topological regularity. The proposed algorithm is inspired from Boissonat’s sculpture [19] method but adopts a different sculpting strategy. We use the combination of circumcenter and circumradius of Delaunay tetrahedron to capture the structure of Delaunay tetrahedra and consequently the geometric proximity of surface samples.

Intuitively, a tetrahedron with one boundary face and its circumcenter lying exterior to the boundary of the tetrahedral mesh tends to have a wider solid angle at the vertex opposite to the boundary face as illustrated in Figure 9. A wider solid angle at the opposite vertex D pushes the boundary vertices further away from each other by making them likely non-neighbors on the surface to be reconstructed. In Figure 9(a), boundary face $\triangle ABC$ has longer edges compared to the interior edges of the tetrahedron. On the contrary, $\triangle ABC$ has shorter edges compared to the interior edges in Figure 9(b). Intuitively, the face $\triangle ABC$ in Figure 9(a) and eventually the tetrahedron itself does not play a role in defining the surface and may be deleted.

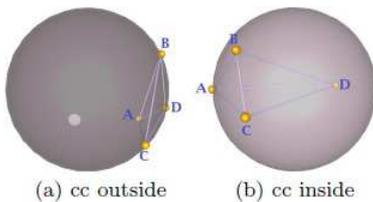


Figure 9: Tetrahedra with one boundary face, $\triangle ABC$ (in both Figures 9(a) & 9(b)). Grey color ball is the circumsphere of tetrahedra. In Figures 9(a) & 9(b), vertex D is opposite to $\triangle ABC$ and lies interior to the Delaunay mesh. cc indicates the circumcenter of the tetrahedron.

Similarly, a tetrahedron having two boundary faces with its circumcenter lying exterior to the boundary of the tetrahedral mesh is found to have a wider planar angle between its interior faces as shown in Figure 10(a). A wider planar angle pushes the vertices opposite to the interior edge to lie at a relatively larger distance and makes

them non-neighbors on the surface. So the tetrahedron (in effect two boundary face and the edge shared by them) may be deleted. In our approach, we remove all boundary tetrahedra of the type shown in Figures 9(a) & 10(b) without relaxing the topological regularity to reconstruct the required surface.

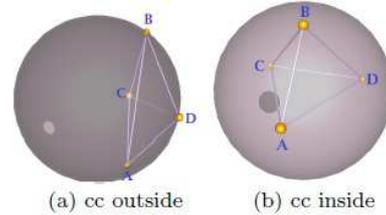


Figure 10: Tetrahedra with two boundary faces, $\triangle ABC$ and $\triangle ABD$ (in both Figures 10(a) & 10(b)). Circumsphere is shown in grey color. Observe the planar angle at edge CD. It’s wider for the tetrahedron in Figure 10(a) than the one in Figure 10(b). Consequently, points A and B lie relatively at a larger distance in Figure 10(a). cc indicates the circumcenter of the tetrahedron.

5.1. Algorithm

The algorithm starts by constructing the Delaunay mesh, $Del(S)$ of the given point cloud S . Then it iteratively removes all **deletable** tetrahedra whose circumcenter lies outside the intermediate boundary ∂D_i (∂D_i is the boundary of restricted Delaunay mesh D_i obtained after each iteration). A tetrahedron T_i in D_i is **deletable** if it satisfies either of the following rules:

Definition 14. *Tetrahedra removal rules:*

1. T_i has only one face (f) on ∂D_i and the vertex opposite to f is not on the boundary of D_i .
2. T_i has exactly two faces (f_1 and f_2) on the boundary of D_i and the edge of T_i , opposite to the common edge of f_1 and f_2 is not on the boundary.

Basically these rules are designed to ensure the topological equivalence of the reconstructed surface to a sphere. Corollary 5.1 establishes this claim. Deleting a tetrahedron with three boundary faces will disconnect a point from D_i and hence we restrict the removal process only to tetrahedra with one or two boundary faces. The algorithm terminates when no more tetrahedra can be removed from D_i and the boundary of non-eliminated tetrahedra constitutes the surface approximation of S . Algorithm 1 presents the pseudo code of the surface reconstruction.

Corollary 5.1. *∂D_i of Delaunay sub mesh D_i obtained in each iteration i , is topologically equivalent to a sphere.*

Proof This is the 3D equivalent to Corollary 4.3. Initially, $D_0 = Del(S)$ and ∂D_0 is the convex hull of S which is topologically equivalent to a sphere. We need to show

Algorithm 1: SHAPE-HULL(S)

Input: Point set S

Output: Triangulated surface mesh B

```
1 Construct  $D_0 = Del(S)$ ;  
2 Construct the heap priority queue,  $PQ$  containing  
   deletable boundary tetrahedra of  $D_0$ , sorted in the  
   descending order of their circumradii;  
3 while  $PQ$  is not empty do  
4    $T = \text{root}(PQ)$ , delete  $T$  from  $PQ$ ;  
5   if  $T$  is deletable & circumcenter( $T$ ) lies outside  
    $\partial D_i$  then  
6     Delete  $T$  from  $D_i$ ;  
7     Add the deletable neighbors of  $T$  to  $PQ$ ;  
8   end  
9 end  
10 return Triangulated surface mesh,  $D_k$ 
```

that the deletion of T_i does not affect the topology of D_i . To establish this claim, we argue that each edge in ∂D_i is incident to exactly two boundary triangles and each vertex in ∂D_i is incident to a set of neighboring triangles homeomorphic to a disk. This is true for ∂D_0 . It can be easily verified that the claim is true for D_i due to tetrahedral removal rules given in Definition 14.

5.2. Selection Criteria

In sculpting algorithms, selection criteria, i.e. selection of tetrahedra for removal plays a crucial role in defining the geometrical correctness of the reconstructed surface. It is unknown what criterion will work well for a wide range of point set and difficult to find one as the requirement may vary depending on the algorithms. Different selection criteria have been employed in various sculpting algorithms. Sculpture [19] algorithm uses a value, $V(T_i)$ defined by the maximum distance between the faces of a boundary tetrahedron, T_i and the associated parts of the circumsphere of T_i as the selection criterion. This removal order may produce deadlock tetrahedra on the reconstructed surface. Deadlock tetrahedra are non-removable tetrahedra (due to tetrahedra removal rules) which are clearly external to the polyhedral representation of the point set. Veltkamp [30] employs γ -indicator (ratio between the radius of the circumsphere of Delaunay tetrahedron and the radius of the circumcircle of the boundary face) which may generate long and skinny triangles on the surface. Hybrid sculpting [35] selects the tetrahedra based on the magnitude of the longest edge of boundary tetrahedron. However, longest edge does not capture the geometric property of a tetrahedron in its entirety.

We use circumradius of a tetrahedron as the selection criterion. Tetrahedron with largest circumradius gets removed first. The reason for choosing circumradius as criterion is that when combined with the location of circumcenter, circumradius serve as an effective tool in capturing

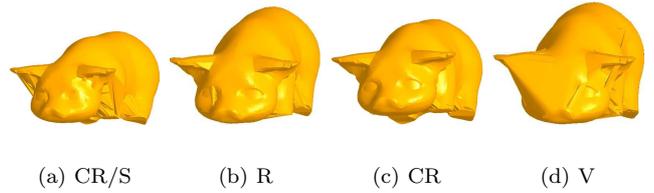


Figure 11: Experiment on selection criteria. It is apparent from the Figure 11(c) that selection based on circumradius generates comparatively better result. In Figure, CR refers to circumradius, S-shortest edge and V-volume of tetrahedron. R refers to random removal.

the geometric proximity of surface samples as already described and illustrated in Figures 9 and 10. Moreover, we experimented with few selection criteria such as volume, ratio of circumradius to the shortest edge, side length and circumradius of tetrahedra. Tetrahedra with the largest quantity was deleted first. From the experiment, selection criterion that found to perform well for different data sets is circumradius (Refer Figure 11 for the results of pig point cloud using different selection criteria). For ordering the boundary tetrahedra according to their circumradius, we employ a priority queue using heap.

5.3. Complexity

In 3D, construction of Delaunay triangulation takes $O(n^2 \log n)$ in the worst case [19]. Let us denote the number of Delaunay tetrahedra in $\text{conv}(S)$ by t_c and the number of tetrahedra in the reconstructed surface by t_d . Let $k = t_c - t_d$. While loop iterates for k times and each iteration costs $O(\log n)$ for push and pop operations of priority queue. Topology checking and deletion of tetrahedra from Delaunay triangulation data structure can be done in constant time. So given $Del(S)$, algorithm takes $O(k \log n)$ time to reconstruct the surface. The preceding complexity implies a direct relation between the time taken for sculpting and the concaveness on the surface. For surfaces having large concave portion, k tends to be large and as a result, the reconstruction time will be substantially increased. However, the worst case time complexity of shape-hull algorithm is $O(n^2 \log n)$, dominated by $Del(S)$ construction.

5.4. Comparison with Sculpting Algorithms

As pointed out in Section 5.2, shape-hull algorithm differs from sculpture [19], constriction [30] and hybrid sculpting [35] algorithms in terms of the selection criteria. Constriction algorithm, starts with $Del(P)$ ($\gamma([-1, 1], [0, 1])$) and uses the fundamental rules of tetrahedra removal (Definition 14) to construct a pruned γ -graph that interpolates the points. In each iteration, the boundary tetrahedra are eliminated based on γ -indicator which is a value based on the ratio between the radius of the circumsphere (R) and the radius of the circumcircle (r) of the boundary

face ($c_0 = 1 - \frac{r}{R}$) (Figure 12). If there are two boundary faces, then sum of the γ -indicators of each one is considered. The boundary tetrahedron with the smallest sum of γ -indicators is removed in each iteration.

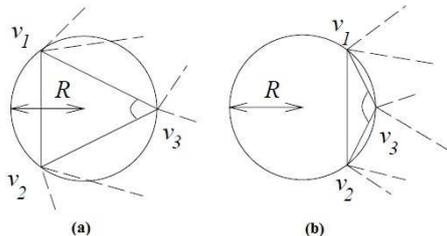


Figure 12: γ -indicator functions in 2D. Figure 12(a) has a γ -indicator > 0 whereas Figure 12(b) has γ -indicator < 0 . [Image courtesy: Veltkamp [30]].

Both, the constriction and shape-hull algorithms depends on a closely related but different selection criterion which uses circumcenter and circumradius. So it is worthwhile and necessary to analyze and study the relationship between both the methods and the corresponding selection criteria. Essentially, both the selection criteria capture the angle at the vertex opposite to the boundary face of a tetrahedron. In addition to the angle, our selection criteria quantifies the size of the tetrahedron, which can not be achieved through γ -indicator. In certain instances, the boundary triangle with the smallest γ -indicator may not be a good choice as illustrated in Figures 13(c) & 13(d). We observe that wider solid angle at interior vertex together with tetrahedron constitutes a better local measure to facilitate the reconstruction. Moreover, γ -indicator incurs an additional computational overhead of division ($\frac{r}{R}$) which definitely makes an impact on the running time.

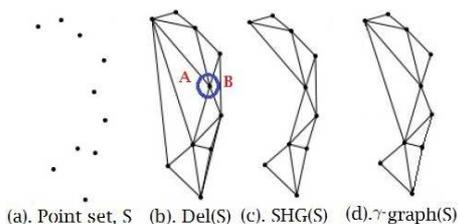


Figure 13: Comparison of $SHG(S)$ with pruned γ -graph(S) for a 2D point set. Two different sized obtuse triangles (A & B) are attached to the vertex encircled in blue color such that the vertex act as an interior vertex to them in some iterations of both the algorithms. Since the small sized triangle (B) has a wider solid angle at the interior vertex and hence a smaller γ -indicator value, constriction algorithm removes it in an earlier iteration. However, since we use circumradius to order the removal, the large sized triangle (A) gets removed first giving a reasonably good shape of S .

In constriction, boundary tetrahedron with an interior circumcenter may be removed if it respects the tetrahedra

removal rules. As opposed to this, no tetrahedra with circumcenter lying interior to the boundary are removed in shape-hull. If the constriction process get stuck with deadlocked tetrahedra then more overlapping 3-simplices are created by fine tuning the value of parameter c_1^0 in the interval $[-1, 0)$. However, for a particular input sample, it's impossible to tell in advance which value of c_1^0 will guarantee a hamiltonian polyhedron and hence completely relies on trial and error method. On the contrary, shape-hull does not rely on any external parameter and for a densely sampled divergent concave surfaces, it guarantees to interpolate the points.

Though, constrained sculpting [29] employs circumradius as the selection criterion, it uses a different removal strategy which depends on the goodness measure of boundary faces inferred through cocone [22] test. One advantage of hybrid and constraint sculpting algorithms is that compared to the other three, it can construct surfaces with genrae. Both the algorithms use the concept of pseudo-prism to construct the genus boundary. However, these algorithms alternates between three stages until the termination criteria is met and hence may take longer computational time. To the best of our knowledge, apart from topological guarantees based on the tetrahedra removal rules, the previous sculpting algorithm gives no bound on the specific sampling distance for which an exact reconstruction is guaranteed. As opposed to this, 3D shape-hull algorithms relies on the theoretical framework designed for 2D shape-hull (Section 4).

6. Experimental Results

Our algorithm was implemented in C++ using Delaunay triangulation and geometric predicates available in Computational Geometry Algorithms Library (CGAL). The algorithm demands fast construction of Delaunay triangulation in order to efficiently scale for large models and exact computations of circumcenters and circumradii of Delaunay tetrahedra for ordering and eliminating tetrahedra from the intermediate boundary. Fortunately, CGAL offers a kernel for exact predicates that computes geometric predicates such as circumcenter and circumradii with numerical certainty and allows fast point location queries to construct optimal Delaunay triangulation. Having compared with the sculpting algorithms in Section 5.4, we now analyze various properties of the reconstructed models and qualitatively compare our results with the results generated by various state of the art algorithms such as powercrust [1], robustcocone [23] and screened Poisson [11]. It is to be noted that all the three algorithms (powercrust, robustcocone and poisson) depend on multiple external parameters and hence we had to try out several parameters (in fact, different combinations of parameters where each parameter was picked from the range of values provided along with the algorithm) to obtain the best possible results in our experiment.

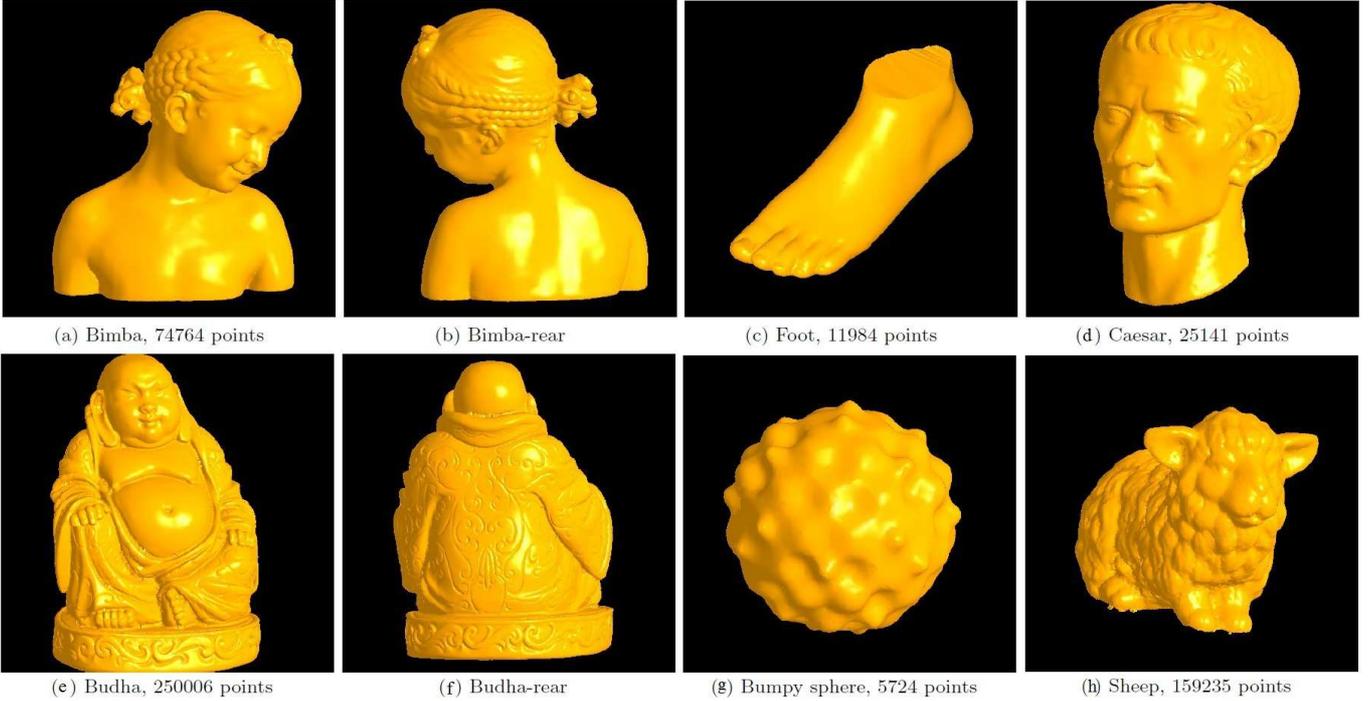


Figure 14: Models reconstructed by our algorithm.

Feature preserving reconstruction: We tested our method on several publicly available data sets from either Stanford 3D scanning repository (bunny) or Aim@shape repository (bimba, sheep, fandisk, foot etc.). Figure 14 shows the models reconstructed by the proposed algorithm for various point clouds of different sizes. The algorithm is able to reconstruct very fine features of several models such as intertwined hair of bimba (Figures 14(a) & (b)), imprints on the base of budha model (Figure 14(e) & (f)), wrinkles on the forehead of caesar (Figure 14(d)), protuberances on the bumpy sphere (Figure 14(g)) and furs of sheep (Figure 14(h)).

Robustness to non-uniform sampling: In surface reconstruction, the reconstructed model is expected to match the original surface in terms of topological and geometric properties [5]. The major challenge of the problem lies in the reconstruction of a model from a sparse, non-uniform and raw point samples, that approximates the original surface S with reasonable accuracy. Figure 15 illustrates the adaptivity of our algorithm to the successive down sampling of raw point cloud. For down-sampling bunny point cloud, we used Poisson disc sampling [40] that refines the existing samples and generates a non-uniform (specifically semi-random) point cloud. In Figure 15, one can observe that screened Poisson progressively shrinks, left ear of the bunny distorts in the second stage and subsequently merges with the right ear in the third stage of under-sampling. Similarly, robust cocone starts degrading from the second stage, important features fades away in the last stage, especially the left ear gets disconnected into pieces. In contrast, powercrust is found to be more resilient to down-sampling. In the third stage, one can observe few deadlocked tetrahedra in shape-hull. One obvious reason for the presence of such undesirable tetrahedra is the violation of divergent concavity between the left ear and the body of bunny due to heavy under-sampling.

Figure 16 shows the evaluation of the under-sampling results performed using root mean square (RMS) of Hausdorff distances computed with METRO [41]. The results indicate that, our method performs well in all the three stages of the sampling, especially for lower samples (722 points) where our method, despite having few unwanted tetrahedra on the reconstructed model, outperforms the

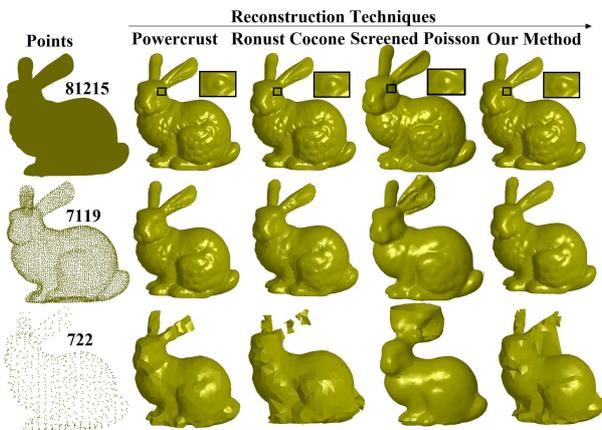


Figure 15: Results on down-sampling experiment on bunny model. Insets show the left eye of each reconstructed bunny model for 81215 point cloud.

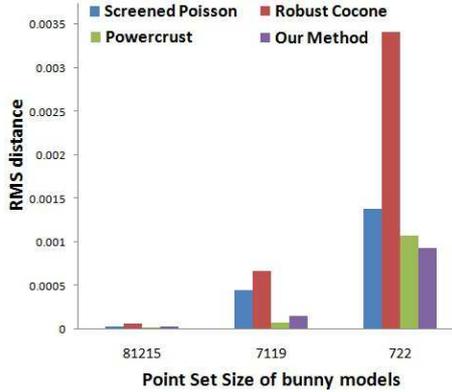


Figure 16: Bar charts of the root mean square (RMS) distances between the reconstructed bunny models (under-sampling experiment) and the reference model.



Figure 17: Models reconstructed by our method for sparse point clouds of star and knob.

other three algorithms. In addition to this, our approach is also capable of reconstructing models from very sparse, non-uniform point clouds as shown in Figure 17.

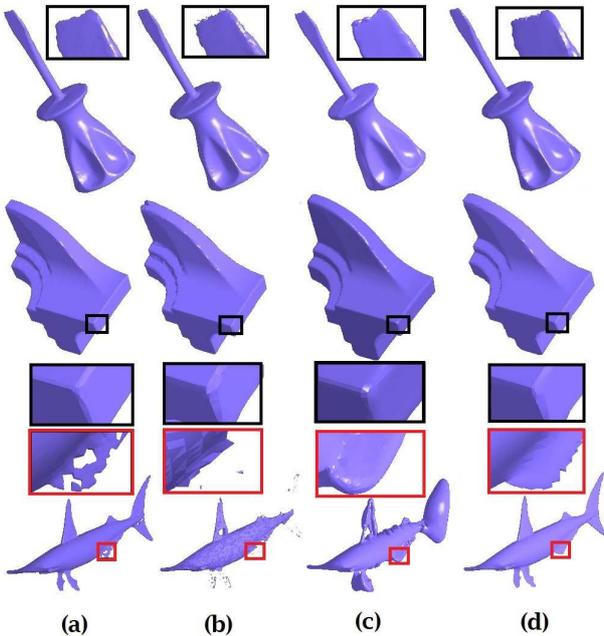


Figure 18: Reconstruction of the screwdriver (first row), fandisc (second row) and shark (third row) data sets by (a). Powercrust (b). Robust cocone (c). Screened Poisson and (d). the proposed algorithm. Close-ups show the sharp head of screw driver, a corner of fandisc and pelvic fin of shark.

Sharp edges and corners: Figure 18 shows how various algorithms reconstruct models with sharp edges and corners. We consider raw point clouds of three models for the experiment: screw driver containing 27152 points, fandisc containing 11984 points and shark containing 10054 points, all of which possess several sharp features. An edge (a curve of non-smooth points on the surface) is considered to be sharp, if the dihedral angle around the edge is bounded away from π degrees. Correspondingly, a sharp corner is a non-smooth point on the surface whose exterior solid angle is bounded away from 2π degrees. Figure 18 broadly indicates that the Delaunay based methods capture the sharp features better than the Screened Poisson method. This is evident from the close-ups of screened Poisson models in Figure 18 (c), where the edges leading from the fandisc corner are round and hence does not capture the sharpness property. Further, dorsal and pelvic fins of the screened Poisson shark model are distorted and it's tail is round and bulgy. Among the Delaunay based algorithms, our method seems to have achieved a better reconstruction, especially when comparing the way the three edges meet at the corner of fandisc in the close-ups of the second row in Figure 18. Similarly, for shark model which has sharp fins and tail, our method generates a better result as compared to the other three. All tetrahedra having either sharp edges or sharp corners are found to have their circumcenter inside the final reconstructed surface due to the dihedral and solid angle constraints and hence our algorithm clearly has a superiority in dealing with such models.

7. Conclusion and Future Work

In this paper, the concept of divergent concavity for simple and closed planar curves has been introduced. A new proximity graph called as shape-hull graph has been defined and its properties have been studied. It has been shown that under ϵ -sampling model, the boundary of shape-hull graph represents the polygonal reconstruction of a divergent concave curve. Extending these ideas to three dimensions, we have presented a simple and non-parametric sculpting algorithm for closed surface reconstruction from raw point clouds. We have done some experiments on under-sampled data and models with sharp features. Results show that our approach is capable of capturing very fine details of the surfaces and reconstructs the models from under-sampled and sparse data. Compared to other prominent surface reconstruction algorithms, our approach found to perform well in the case of reconstruction of models with sharp features. There are some avenues for future work, especially the reconstruction of surfaces having genus. An extension of the proposed algorithm for addressing non-divergent concave surfaces is also under our consideration.

References

- [1] N. Amenta, S. Choi, R. K. Kolluri, The power crust, in: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, SMA '01, ACM, New York, NY, USA, 2001, pp. 249–266.
- [2] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk, The digital michelangelo project: 3d scanning of large statues, in: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000, pp. 131–144.
- [3] N. Amenta, M. Bern, Surface reconstruction by voronoi filtering, in: Proceedings of the Fourteenth Annual Symposium on Computational Geometry, SCG '98, ACM, New York, NY, USA, 1998, pp. 39–48.
- [4] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, C. T. Silva, A benchmark for surface reconstruction, ACM Trans. Graph. 32 (2) (2013) 20:1–20:17.
- [5] F. Cazals, J. Giesen, Delaunay triangulation based surface reconstruction: Ideas and algorithms, in: Effective Computational Geometry for Curves and Surfaces, Springer, 2006, pp. 231–273.
- [6] K. R. Gabriel, R. R. Sokal, A new statistical approach to geographic variation analysis, Systematic Zoology 18 (3) (1969) pp. 259–278.
- [7] N. Amenta, M. Bern, D. Eppstein, The crust and the beta-skeleton: Combinatorial curve reconstruction, in: Graphical Models and Image Processing, 1998, pp. 125–135.
- [8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, SIGGRAPH Comput. Graph. 26 (2) (1992) 71–78.
- [9] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, H.-P. Seidel, Multi-level partition of unity implicits, in: ACM SIGGRAPH 2003 Papers, SIGGRAPH '03, ACM, New York, NY, USA, 2003, pp. 463–470.
- [10] M. Kazhdan, Reconstruction of solid models from oriented point sets, in: Proceedings of the Third Eurographics Symposium on Geometry Processing, SGP '05, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2005.
- [11] M. Kazhdan, H. Hoppe, Screened poisson surface reconstruction, ACM Trans. Graph. 32 (3) (2013) 29:1–29:13.
- [12] J. Manson, G. Petrova, S. Schaefer, Streaming surface reconstruction using wavelets, Computer Graphics Forum (Proceedings of the Symposium on Geometry Processing) 27 (5) (2008) 1411–1420.
- [13] P. Alliez, D. Cohen-Steiner, Y. Tong, M. Desbrun, Voronoi-based variational reconstruction of unoriented point sets, in: Proceedings of the Fifth Eurographics Symposium on Geometry Processing, SGP '07, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2007, pp. 39–48.
- [14] M. Alexa, J. Behr, D. Cohen-or, S. Fleishman, D. Levin, C. T. Silva, Computing and rendering point set surfaces, IEEE Transactions on Visualization and Computer Graphics 9 (2003) 3–15.
- [15] S. Fleishman, D. Cohen-Or, C. T. Silva, Robust moving least-squares fitting with sharp features, in: ACM SIGGRAPH 2005 Papers, SIGGRAPH '05, ACM, New York, NY, USA, 2005, pp. 544–552.
- [16] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, T. R. Evans, Reconstruction and representation of 3d objects with radial basis functions, in: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01, ACM, New York, NY, USA, 2001, pp. 67–76.
- [17] H. Q. Dinh, G. Turk, Reconstructing surfaces using anisotropic basis functions, in: In International Conference on Computer Vision (ICCV) 2001, 2001, pp. 606–613.
- [18] T. K. Dey, Curve and Surface Reconstruction: Algorithms with Mathematical Analysis (Cambridge Monographs on Applied and Computational Mathematics), Cambridge University Press, New York, NY, USA, 2006.
- [19] J.-D. Boissonnat, Geometric structures for three-dimensional shape representation, ACM Trans. Graph. 3 (4) (1984) 266–286.
- [20] H. Edelsbrunner, E. P. Mücke, Three-dimensional alpha shapes, ACM Trans. Graph. 13 (1) (1994) 43–72.
- [21] D. Attali, r-regular shape reconstruction from unorganized points., Comput. Geom. 10 (4) (1998) 239–247.
- [22] N. Amenta, S. Choi, T. K. Dey, N. Leekha, A simple algorithm for homeomorphic surface reconstruction, in: Proceedings of the Sixteenth Annual Symposium on Computational Geometry, SCG '00, ACM, New York, NY, USA, 2000, pp. 213–222.
- [23] T. K. Dey, S. Goswami, Provable surface reconstruction from noisy samples, Comput. Geom. Theory Appl. 35 (1) (2006) 124–141.
- [24] T. K. Dey, S. Goswami, Tight cocone: A water-tight surface reconstructor, in: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications, SM '03, ACM, New York, NY, USA, 2003, pp. 127–134.
- [25] T. Dey, J. Giesen, J. Hudson, Delaunay based shape reconstruction from large data, in: Parallel and Large-Data Visualization and Graphics, 2001. Proceedings. IEEE 2001 Symposium on, 2001, pp. 19–146.
- [26] M. Gopi, S. Krishnan, C. T. Silva, Surface reconstruction based on lower dimensional localized delaunay triangulation., Comput. Graph. Forum 19 (3) (2000) 467–478.
- [27] H. Edelsbrunner, Surface reconstruction by wrapping finite sets in space, Discrete and Computational Geometry 5 (2003) 379–404.
- [28] E. A. Ramos, B. Sadri, Geometric and topological guarantees for the wrap reconstruction algorithm., in: N. Bansal, K. Pruhs, C. Stein (Eds.), SODA, SIAM, 2007, pp. 1086–1095.
- [29] A. Gezahegne, Surface Reconstruction with Constrained Sculpting, University of California, Davis, 2005.
- [30] R. C. Veltkamp, Closed Object Boundaries from Scattered Points, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1994.
- [31] D. Cohen-Steiner, F. Da, A greedy delaunay-based surface reconstruction algorithm., The Visual Computer 20 (1) (2004) 4–16.
- [32] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, S. Member, The ball-pivoting algorithm for surface reconstruction, IEEE Transactions on Visualization and Computer Graphics 5 (1999) 349–359.
- [33] S. Petitjean, E. Boyer, Regular and non-regular point sets: Properties and reconstruction, Computational Geometry 19 (23) (2001) 101 – 126.
- [34] R. Chaine, A geometric-based convection approach of 3-d reconstruction., in: L. Kobbelt, P. Schröder, H. Hoppe (Eds.), Symposium on Geometry Processing, Vol. 43, Eurographics Association, 2003, pp. 218–229.
- [35] M. Attene, M. Spagnuolo, Automatic surface reconstruction from point sets in space, Computer Graphics Forum 19 (2000) 457–465.
- [36] C.-C. Kuo, H.-T. Yau, A delaunay-based region-growing approach to surface reconstruction from unorganized points, Computer-Aided Design 37 (8) (2005) 825 – 835.
- [37] J. O'Rourke, Computational Geometry in C, 2nd Edition, Cambridge University Press, New York, NY, USA, 1998.
- [38] S.-W. Cheng, T. K. Dey, J. R. Shewchuk, Delaunay Mesh Generation, Chapman and Hall / CRC computer and information science series, CRC Press, 2013.
- [39] J. Jaromczyk, G. Toussaint, Relative neighborhood graphs and their relatives, Proceedings of the IEEE 80 (9) (1992) 1502 – 1517. doi:10.1109/5.163414.
- [40] M. Corsini, P. Cignoni, R. Scopigno, Efficient and flexible sampling with blue noise properties of triangular meshes, IEEE Transactions on Visualization and Computer Graphics 18 (6) (2012) 914–924.
- [41] P. Cignoni, C. Rocchini, R. Scopigno, Metro: Measuring error on simplified surfaces, Computer Graphics Forum 17 (2) (1998) 167–174. doi:10.1111/1467-8659.00236.