Delaunay Meshing of Piecewise Smooth Complexes without Expensive Predicates *

Tamal K. Dey[†] Joshua A. Levine[‡]

Abstract

Recently a Delaunay refinement algorithm has been proposed that can mesh piecewise smooth complexes which include polyhedra, smooth and piecewise smooth surfaces, and non-manifolds. However, this algorithm employs domain dependent numerical predicates, some of which could be computationally expensive and hard to implement. In this paper we develop a refinement strategy that eliminates these complicated domain dependent predicates. As a result we obtain a meshing algorithm that is practical and implementation-friendly.

[*] Research supported by NSF, USA (CCF-0430735 and CCF-0635008).

[†] Department of Computer Science and Engineering, Ohio State University, Ohio, USA. Email: tamaldey@cse.ohio-state.edu

[‡] Department of Computer Science and Engineering, Ohio State University, Ohio, USA. Email: levinej@cse.ohio-state.edu

1 Introduction

Delaunay mesh generation of non-smooth domains such as piecewise smooth surfaces and complexes is a difficult challenge. Aided by recent developments in sampling theory [1, 2] and computational topology [17], Chew's furthest point strategy [11, 12] (Delaunay refinement) has been applied to generate Delaunay meshes for smooth surfaces with provable guarantees [5, 10]. The lack of global smoothness poses two main difficulties in extending these methods to non-smooth domains. First, the sampling theory developed for smooth surfaces [1, 2] breaks down for non-smooth surfaces. Secondly, small input angles possibly present at non-smooth regions pose problems for the termination of Delaunay refinement [21].

Boissonnat and Oudot [4] gave a provable algorithm for a class of non-smooth surfaces called Lipschitz-surfaces. They showed that their algorithm for smooth surface meshing [5] extends to this class only if input angles are sufficiently large. Unfortunately, this approach failed to admit small input angles which limited the input class. Recently Cheng, Dey, and Ramos [9] proposed an approach that completely removed any constraint on input angles. As a result their algorithm could accommodate large class of input domains called piecewise smooth complexes (PSCs). This class includes polyhedral domains, smooth and piecewise smooth surfaces, and even non-manifolds.

The algorithm of [9] uses the idea of protecting non-smooth curves and vertices in the input complex with balls. This idea already gave good results in the polyhedral case [7, 15, 18]. A novelty introduced by Cheng et al. is that they turn the balls into weighted points and then carry out the mesh refinement in the weighted Delaunay triangulation [3, 16]. Notwithstanding its theoretical success, practical relevance of this algorithm remains questionable. It employs some expensive numerical computations that are hard to implement. Unless these computations are removed, one cannot expect a practical solution for Delaunay mesh generation of PSCs. The goal of this paper is to design a refinement strategy without expensive predicates so that it becomes implementation-friendly and practical.

Bottleneck. Consider a smooth surface given by an implicit equation. To generate a meaningful mesh for this surface, one needs to sample it at a scale that captures its smallest local variations [13]. One approach would be to *guess* a scale and sample the surface with it. If the guess is right, the sampling with the furthest point strategy gives a mesh with provable guarantees as shown by Boissonnat and Oudot [5]. A second approach would be to compute a version of the local length scales and mesh the surface with those scales. The algorithm of Cheng et al. [9] works on this latter principle. It computes how a curve or a surface varies normal-wise around a point. It also computes separation distances between different elements (vertices, curves, surfaces) to capture separation feature size (gap size) in the sense of [7, 20]. These assumed powerful numerical primitives allow Cheng et al. to determine the size of the protecting balls with certain desirable properties and allow them to sample surface patches at appropriate length scales. Unfortunately, these computations are expensive and are hard to implement which makes the algorithm impractical.

Solution. To circumvent the problem we follow the guessing approach. We would like to guarantee that even if the guess is incorrect, the algorithm terminates and outputs a mesh which approximates the input complex at a coarse level. A main difficulty in this approach is to formulate a unified refinement strategy that captures the input topology correctly when the guessed scale is right and outputs a mesh with some reasonable properties all the time. To reach this goal we formulate a disk condition that says that the output mesh should have a topological disk formed by triangles around each vertex. This disk condition drives the refinement, that is, we go on refining protecting balls or surface meshes if this disk condition fails. If this refinement terminates, the disk condition necessarily holds for the output mesh. Then, by PL topology, the output Delaunay mesh restricted to each manifold surface patch is a manifold. Furthermore, the input incidence structure among different elements in the PSC is maintained. The output may not be homeomorphic to the input since a small feature such as a small handle may not be detected at the length scale the algorithm is asked to operate. However, a homeomorphic meshing is guaranteed when the supplied scale is sufficiently small. Actually, in practice, the disk condition usually suffices to provide a homeomorphism.

One of our main tasks is to prove that the refinement always terminates. To this end we use the following result. If the protecting balls are sufficiently small and satisfy some *separation* properties (conditions C1-C3 in section 2), then the disk condition holds if the restricted Delaunay triangles are sufficiently small. Therefore, the failure of the disk condition signals either balls do not satisfy separation properties, or are large, or triangles are large. The way we compute the protecting balls, failure of separation properties also implies that they are large. In essence, if the disk condition fails, either a ball or a triangle is too large. The algorithm refines the larger of the two and hence guarantees that neither a protecting ball nor a triangle gets arbitrarily small ensuring termination. One may observe that, this strategy does not allow adaptive mesh sizing. However, one may regulate the input scale to produce meshes at different levels of resolution.

Perhaps the most important ingredient in this approach is to maintain a set of protecting balls with the separation properties. It turns out that it is difficult to ensure one of these properties exactly (condition C3) when the balls are large. Instead we maintain a more relaxed condition which implies the desired property when the balls are sufficiently small.

We have implemented our algorithm in this paper. In an earlier attempt we tried to mesh with the disk condition but pre-computed the balls with a small radius chosen heuristically [8]. The code failed in cases where the pre-selected size of the balls was wrong. The approach in this paper allows the refinement algorithm to determine the balls automatically instead of pre-computing them. We report experimental results for our protection algorithm and meshing in Section A of the Appendix. The code and a video explaining the experimental results have been released [14].

1.1 Domain Throughout this paper, we assume a generic intersection property that a k-manifold $\sigma \subset \mathbb{R}^3$, $0 \leqslant k \leqslant 2$, and a j-manifold $\sigma' \subset \mathbb{R}^3$, $0 \leqslant j \leqslant 2$, intersect (if at all) in a $(k+j-3)$-manifold if $\sigma \not\subset \sigma'$ and $\sigma' \not\subset \sigma$. We will use both geometric and topological versions of closed balls. A *geometric* closed ball centered at point $x \in \mathbb{R}^3$ with radius $r > 0$, is denoted as $B(x,r)$. We use $\operatorname{int} \mathbb{X}$ and $\operatorname{bd} \mathbb{X}$ to denote the interior and boundary of a topological space $\mathbb{X}$, respectively.

The domain $\mathcal{D}$ is a piecewise smooth complex (PSC) where each element is a compact subset of a smooth (C^2) k-manifold, $0 \leqslant k \leqslant 2$. Each element is closed and hence contains its boundaries. For simplicity we assume that each element has a non-empty boundary (used in Lemma 4.1, this restriction can be removed by some added complication in the initialization of the algorithm). We use $\mathcal{D}_k$ to denote the subset of all k-dimensional elements, the kth stratum. $\mathcal{D}_0$ is a set of *vertices*; $\mathcal{D}_1$ is a set of curves called 1-*faces*; $\mathcal{D}_2$ is a set of surface patches called 2-*faces*. For $1 \leqslant k \leqslant 2$, we use $\mathcal{D}_{\leqslant k}$ to denote $\mathcal{D}_0 \cup \ldots \cup \mathcal{D}_k$.

The domain $\mathcal{D}$ satisfies the usual proper requirements for being a complex: (i) interiors of the elements are pairwise disjoint and for any $\sigma \in \mathcal{D}$, $\operatorname{bd} \sigma \subset \mathcal{D}$; (ii) for any $\sigma, \sigma' \in \mathcal{D}$, either $\sigma \cap \sigma' = \emptyset$ or $\sigma \cap \sigma'$ is a union of elements in $\mathcal{D}$. We use $|\mathcal{D}|$ to denote the underlying space of $\mathcal{D}$. For $0 \leqslant k \leqslant 2$, we also use $|\mathcal{D}_k|$ to denote the underlying space of $\mathcal{D}_k$.

1.2 Complexes We will be dealing with weighted points and their Delaunay and Voronoi diagrams. A weighted point p with weight r is represented as a ball $b = B(p,r)$. The squared *weighted distance* of any point $x \in \mathbb{R}^3$ from p is given by $\|x-p\|^2 - r^2$. Under this distance metric, one can define weighted versions of Delaunay and Voronoi diagrams. For a weighted point set $S \subset \mathbb{R}^3$, let $\operatorname{Vor} S$ and $\operatorname{Del} S$ denote the weighted Voronoi and Delaunay diagrams of S respectively. Each diagram is a cell complex where each k-face is a k-polytope in $\operatorname{Vor} S$ and is a k-simplex in $\operatorname{Del} S$. Each k-simplex ξ in $\operatorname{Del} S$ is dual to a $(3-k)$-face V_ξ in $\operatorname{Vor} S$ and vice versa.

Let S be a point set sampled from $|\mathcal{D}|$. For any sub-collection $\mathbb{X} \subset \mathcal{D}$ we define $\operatorname{Del} S|_{\mathbb{X}}$ to be the Delaunay subcomplex restricted to $\mathbb{X}$, i.e., each simplex $\xi \in \operatorname{Del} S|_{\mathbb{X}}$, called a restricted simplex, is the dual of a Voronoi face V_ξ where $V_\xi|_{\mathbb{X}} = V_\xi \cap |\mathbb{X}| \neq \emptyset$. By this definition, for any $\sigma \in \mathcal{D}$, $\operatorname{Del} S|_\sigma$ denotes the Delaunay subcomplex restricted to σ, and $\operatorname{Del} S|_{\mathcal{D}_i} = \bigcup_{\sigma \in \mathcal{D}_i} \operatorname{Del} S|_\sigma$ and $\operatorname{Del} S|_{\mathcal{D}} = \bigcup_{\sigma \in \mathcal{D}} \operatorname{Del} S|_\sigma$. An i-face $\sigma \in \mathcal{D}_i$ should be meshed with i-simplices. However, $\operatorname{Del} S|_\sigma$ may have lower dimensional simplices not incident to any restricted i-simplex. Therefore, we compute special sub-complexes of restricted complexes. Define the following i-dimensional subcomplexes (Figure 5 in Appendix):

$$\operatorname{Skl}^i S|_\sigma = \mathit{closure}\{t \mid t \in \operatorname{Del} S|_\sigma \text{ is an } i\text{-simplex}\} \text{ and } \operatorname{Skl}^i S|_{\mathcal{D}_i} = \bigcup_{\sigma \in \mathcal{D}_i} \operatorname{Skl}^i S|_\sigma.$$

2 Protection and refinement

The meshing algorithm computes a set of balls protecting 1-faces. Unlike [9], the protecting balls are adjusted on the fly as refinement proceeds. It inserts points in 2-faces to refine the triangulation. The protecting balls and the triangulation are refined simultaneously either to satisfy a disk condition or to achieve a refinement level dictated by an input scale parameter. In what follows all skipped proofs appear in the Appendix.

2.1 Covering 1-faces Let $\sigma(x,y)$ denote the curve segment oriented from x to y on any 1-face σ. In this notation $\sigma = \sigma(u,v)$ where σ is oriented from the end point u to the other end point v. Let $b = B(c,r)$ be a ball with $c \in \sigma(x,y)$. The intersection $b \cap \sigma$ is a set of curve segments. Among them the curve segment containing c is called the *segment of* b *in* σ, $\operatorname{seg}_\sigma(b)$, see Figure 1. We call two balls b and b' and their corresponding weighted vertices *adjacent* if their centers are adjacent on a 1-face σ. We use $d(x,y)$ to denote the Euclidean distance between two points x, y and use $d_\sigma(x,y)$ to denote the length of a curve segment $\sigma(x,y)$. Let $b_0, b_1, ..., b_k$ be a set of balls that protect σ where $b_i = B(c_i, r_i)$ with $c_i \in \sigma$. We require that the balls satisfy the following conditions:

(C1) b_0 and b_k are centered at u and v respectively. These will be called the *vertex balls*.

(C2) σ is covered by the balls, that is, $\sigma \subseteq \cup_i \operatorname{seg}_\sigma(b_i)$ and any two adjacent balls $b = B(c,r)$ and $b' = B(c',r')$ intersect deeply, that is, $d(c,c') \leqslant r + \frac{6r'}{7}$ where $r' \leqslant r$.

(C3) No point in $\operatorname{seg}_\sigma(b_i)$ is contained in a ball non-adjacent to b_i.

Notice that the choice of the constant $\frac{6}{7}$ in C2 is a little arbitrary. We need only a factor of r' in the expression and a follow-up analysis with other constants are also possible.

We will maintain a point set $S \subset \mathcal{D}$ with the following properties throughout the algorithm: all points in S except those in $\mathcal{D}_{\leqslant 1}$ are unweighted and no unweighted point has a negative weighted distance to any other point. This means each unweighted point $p \in S$ has V_p non-empty. We call such a point set *admissible*.

It is worthwhile to note that one of the consequences of conditions C1-C3 would be the following result which would imply R2 in Lemma 2.2.

Lemma 2.1. *Let S be an admissible point set satisfying conditions C1-C3. Let p and q be adjacent weighted vertices on a 1-face σ. V_{pq} is the only Voronoi facet in* $\operatorname{Vor} S$ *that intersects $\sigma(p,q)$.*

Proof. Let z be any point in $\sigma(p,q)$. Let b_p and b_q be the balls centered at p and q respectively. The point z is contained in $b_p \cup b_q$. Since z cannot be contained in any ball other than b_p and b_q (property

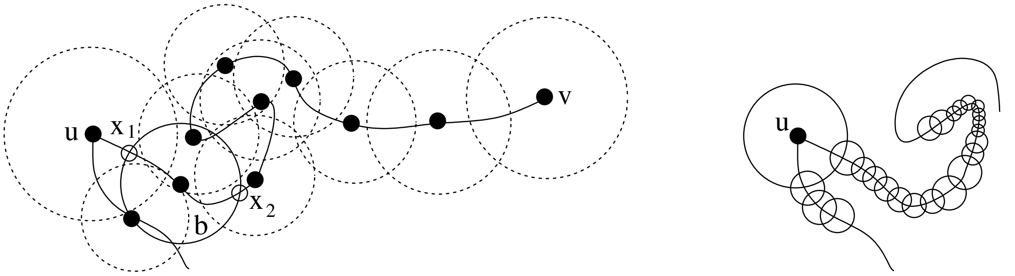

Figure 1: Covering 1-faces : (left) A 1-face σ between u and v is being protected. The ball b shown with solid boundary has $\mathrm{seg}_\sigma(b)$ as the curve segment between x_1 and x_2. The balls satisfy C1 and C2 but intersect arbitrarily. (right) Balls are refined and they start satisfying separation properties C1-C3.

C3), it cannot lie on any Voronoi facet partly defined by a point other than p and q. However, $\sigma(p,q)$ has to intersect at least one Voronoi facet since p and q lie in two different Voronoi cells. Therefore, the only Voronoi facet which intersects $\sigma(p,q)$ is V_{pq}. $\square$

For any triangle $t \in \mathrm{Skl}^2\, S|_\sigma$, define $\mathrm{size}(t,\sigma)$ to be the maximum weighted distance between the vertices of t and points in $V_t|_\sigma$. This is the maximum weighted distance between vertices of t and the points where the dual Voronoi edge of t intersects σ.

λ-property: We say S has the λ-*weight* property if each point in S has a weight at most $\lambda \geqslant 0$. We say S has the λ-*size* property if $\mathrm{size}(t,\sigma) \leqslant \lambda$ for each triangle $t \in \mathrm{Skl}^2\, S|_\sigma$ and S has the λ-weight property.

The following observation is at the heart of our refinement algorithm (see section C in Appendix for a proof sketch).

LEMMA 2.2. *Let $S \subset \mathcal{D}$ be an admissible point set and $p \in S$ be a point on a 2-face σ. Let $\sigma_p \subset \sigma$ be the set of all connected components in $V_p|_\sigma$ that intersect a Voronoi edge. There exists a constant $\lambda > 0$ so that hypotheses (H1)-(H3) imply results R1 and R2 where*

(H1) σ_p is not empty.

(H2) S satisfies λ-size property.

(H3) weighted points in S satisfy C1-C3.

(R1) σ_p is a 2-disk where any edge of V_p intersects σ_p at most once and any facet of V_p intersects σ_p in an empty set or an open curve.

(R2) If $p \in \mathrm{bd}\,\sigma$, at least two Voronoi facets of V_p intersect $\mathrm{bd}\,\sigma$, each intersecting one of the curve segments between p and its adjacent weighted points (possibly two) in $\mathrm{bd}\,\sigma$.

Interpreted in terms of the Delaunay triangulation, the conclusion of the above lemma implies that the triangles incident to p and restricted with respect to σ form a topological disk around p. This disk has p at the boundary if and only if p is in $\mathrm{bd}\,\sigma$. Furthermore, if p is in $\mathrm{bd}\,\sigma$, it is connected to its two adjacent weighted points in $\mathrm{bd}\,\sigma$ on this disk. We will formulate a *disk condition* with these properties. Our refinement algorithm is primarily driven by this disk condition. We will see that (H1) holds if (H2) holds. Therefore, the conclusion of Lemma 2.2 fails only if either there is a protecting ball with radius more than λ, or there is a triangle $t \in \mathrm{Skl}^2\, S|_\sigma$ for which $\mathrm{size}(t, \sigma) > \lambda$ for some $\lambda > 0$. However, since we do not know which of the above two cases has happened, we take a conservative approach. We compute the maximum radius $r_{\max}$ of all protecting balls and also compute the maximum $d_{\max} = \mathrm{size}(t, \sigma)$ over all t and σ. Let x be the point of intersection of a Voronoi edge with $\mathcal{D}$ which realizes $d_{\max}$. If $r_{\max} > d_{max}$ we refine the largest protecting ball. Otherwise, we insert x. In the first case we are ensured that we are refining balls of size larger than a fixed positive constant. In the second case, we are inserting a point in a compact domain with a positive lower bound on its distances to every other points. Termination by packing argument follows.

For the above algorithm to work, it is important that the balls satisfy C1-C3 when balls are sufficiently small. It turns out that it is difficult to maintain the condition C3 at early phases when the balls are relatively large. We replace C3 by the following two conditions that are maintained by the ball refinement algorithm. These conditions imply C3 when the balls are small enough. For a 1-face $\sigma(u, v)$ covered by balls $b_0, b_1, .., b_k$ these conditions are:

(C3.a) Let $b = B(c, r)$ be any ball in $\{b_0, b_1, ..., b_k\}$. If c' is contained in $\mathrm{seg}_\sigma(b)$ for an adjacent ball $b' = B(c', r')$, then $d_\sigma(c, c') \geqslant \frac{8}{7} r$.

(C3.b) Any two balls centered in different 1-faces do not intersect.

Lemma 2.3. *There exists a $\lambda > 0$ so that, if all protecting balls are smaller than λ, C3.a and C3.b imply C3.*

Proof. First consider any two non-adjacent balls b_1 and b_2 with centers c_1 and c_2 respectively on a curve σ. It follows from the differentiability of σ that there exists a $\lambda > 0$ so that any ball of size smaller than λ intersects σ in a single segment. Assuming that each b_i has a radius smaller than λ, we have $b_i \cap \sigma = \mathrm{seg}_\sigma(b_i)$. We claim that no point in $\mathrm{seg}_\sigma(b_2)$ can lie in $\mathrm{seg}_\sigma(b_1)$ when λ is sufficiently small. If there were such a point there would exist a ball b_3 adjacent to b_1 whose center c_3 would lie in $\mathrm{seg}_\sigma(b_1)$. But we know that the length $d_\sigma(c_1, c_3)$ is at least $\frac{8}{7} r_1$ by property C3.a. Making λ sufficiently small, the length of $\mathrm{seg}_\sigma(b_1)$ can be made arbitrarily close to r_1 reaching a contradiction. Therefore, we can claim that the curve segments of two non-adjacent balls cannot intersect if they are centered on a same 1-face σ (notice that the balls may intersect otherwise). The lemma follows since balls centered on different 1-faces are not allowed to intersect by C3.b. ⊔⊓

2.2 Ball refinement

The ball refinement routine simply removes a ball and covers the curve segment between the centers of its adjacent balls with balls of smaller radii. Therefore, we encounter the generic situation where the curve segment $\sigma(x, y)$ needs to be covered by protecting balls whose radii are determined by a given parameter $\alpha > 0$. The points x and y are the right and left end points of some segments, say $\mathrm{seg}_\sigma(b_0)$ and $\mathrm{seg}_\sigma(b_k)$ respectively, see Figure 2. We call this routine Cover(x, y, α).

We proceed from x toward y along the curve while computing the balls that satisfy conditions C1,C2, and C3.a. Condition C3.b will be taken care of by another routine called Separate. Suppose that $b_i = B(c_i, r_i)$ has already been computed. Let $\sigma(x_i, y_i) = \mathrm{seg}_\sigma(b_i)$. We compute a small ball $\beta_{i+1} = B(y_i, \alpha/3)$ that *aids* the computation of b_{i+1}, see Figure 2.5. The segment $\mathrm{seg}_\sigma(\beta_{i+1})$ is of interest to us as we would like to put the next ball b_{i+1} at the right end point, say z_{i+1}, of this segment where r_{i+1} is taken as α. However, there could potentially be problems if $z_{i+1} \notin \sigma(x, y)$ as in Figure 2.5(left) or $\mathrm{seg}_\sigma(b_{i+1})$ contains y as in Figure 2(right). In the first case b_k may violate C3.a and in the second case b_{i+1} may not intersect b_k deeply violating C2. In these cases we conduct an *end game.* If $z_{i+1} \in \mathrm{seg}_\sigma(b_k)$, we throw away z_{i+1} and take b_{i+1} as β_{i+1} enlarged concentrically to a radius

$\frac{2\alpha}{3}$. In the other case when $\mathrm{seg}_\sigma(b_{i+1})$ contains y, we enlarge b_{i+1} to a radius of $\frac{7\alpha}{6}$. Cover terminates after the end game.

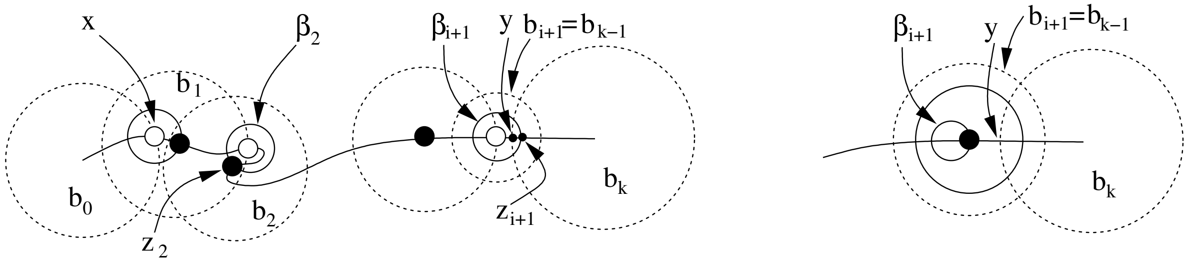

Figure 2: Curve segment between x and y is being covered. Aiding balls are shown with solid boundaries. Notice how the centers of b_1 and b_2 are placed with the aiding balls. The end game with enlarged aiding ball is shown on left, the other case is shown on right.

Lemma 2.4. Cover(x, y, α) *maintains* C1*,* C2*, and* C3.a.

Cover does not necessarily satisfy C3.b. We use the routine Separate to enforce C3.b on a set of balls $\mathcal{B}$. This routine calls RefineBall(b) which removes the ball b and replaces it with smaller balls.

```
Separate($\mathcal{B}$)

  1. while ($\exists$ intersecting balls $b, b' \in \mathcal{B}$ with $c$, $c'$ in different 1-faces)
       If $r \geqslant r'$, $\mathcal{B} := (\mathcal{B} \setminus b) \cup$ RefineBall($b$) else $\mathcal{B} := (\mathcal{B} \setminus b') \cup$ RefineBall($b'$).
     endwhile.
  2. return $\mathcal{B}$.
```

RefineBall(b)
If b covers $\sigma \in \mathcal{D}_1$, let r_σ be the minimum of the radii of balls adjacent to b on σ and b itself.

1. If $b = B(v, r)$ is a vertex ball, shrink b to $b' = B(v, r/2)$. For each σ covered by b let b_σ be the adjacent ball to b on σ. Compute $\mathcal{B}_\sigma :=$ RefineBall(b_σ). Return Separate$(\cup_\sigma \mathcal{B}_\sigma)$.
2. If b is not a vertex ball, let $\sigma(x, y)$ be the segment between $\mathrm{seg}_\sigma(b_1)$ and $\mathrm{seg}_\sigma(b_2)$ where b_1 and b_2 are adjacent to b. Remove b and return Cover(x, y, α) where $\alpha = r_\sigma/4$.

Lemma 2.5. *(i)* RefineBall(b) *terminates, and (ii) maintains* C1,C2,C3.a,C3.b.

Proof. (i) : Observe that RefineBall(b) makes recursive calls to itself and through Separate. Consider the tree of ball refinements rooted at b made by these recursive calls. An internal node b' in the tree represents a ball b' that is refined into smaller balls (children).

Notice that RefineBall may refine at most one vertex ball which is necessarily the root b. Since all vertex balls are kept fixed after that, two balls centered at two different 1-faces intersect only if the larger ball has a radius more than a fixed positive constant $\lambda > 0$. Hence an internal node is refined only if it has a radius more than $\lambda > 0$.

The children of a node b' are created by Cover which, by construction, creates only finitely many balls with radius at most $(1/4 \times 7/6) = 7/24$th the radius of b'. The height of the refinement tree is finite since any path from the root to a leaf has internal nodes with radius larger than $\lambda > 0$ and each level decreases the radius by a factor $7/24$ or less. Also, this tree has finitely many children for all nodes. Therefore, the tree is finite implying that RefineBall terminates.

(ii): Assume that C1,C2, and C3.a hold before calling RefineBall. It creates new balls by calling Cover which satisfies C1,C2, and C3.a (Lemma 2.4). If a ball does not satisfy C3.b (may happen only after a vertex ball is refined), it refines it by calling Separate. The claim follows. ◻

3 Meshing algorithm

The algorithm for meshing $\mathcal{D}$ first protects the 1-faces with Protect($\mathcal{D}$,λ) where the input scale parameter λ is the upper limit for the radii of the protecting balls.

```
Protect($\mathcal{D}$,$\lambda$)
  1. Protect each vertex $v \in \mathcal{D}_0$ with a ball $B_v = B(v, r_v)$ where $r_v$ is 1/3rd the distance of
     $v$ to any other vertex in $\mathcal{D}_0$. Let $\mathcal{B}$ be the set of vertex balls.
  2. For each $\sigma \in \mathcal{D}_1$ do the following. Let $u$ and $v$ be the end points of $\sigma$ where
     $\mathrm{seg}_\sigma(B_u) = ux$ and $\mathrm{seg}_\sigma(B_v) = yv$. Let $\alpha = \min\{r_u, r_v\}$. $\mathcal{B} := \mathcal{B} \cup$ Cover$(x, y, \alpha)$.
  3. If there is any ball $b \in \mathcal{B}$ with radius larger than $\lambda$, compute $\mathcal{B} := \{\mathcal{B} \setminus b\} \cup$
     RefineBall$(b)$.
  4. return Separate($\mathcal{B}$).
```

Lemma 3.1. Protect *terminates with balls satisfying C1,C2,C3.a, and C3.b.*

Proof. Observe that at the end of step 2, Protect creates a set $\mathcal{B}$ of finitely many balls (most likely quite large). After that, we represent the refinements in step 3 and 4 with a refinement tree for each ball $b \in \mathcal{B}$. Each node in these trees represent a call to RefineBall and its children represent the finitely many balls created as output of the call (Lemma 2.5).

In step 4 if a vertex ball, say $b' = B(u, r)$, is refined, its radius cannot be smaller than half of the distance of u from all 1-faces that do not contain u. This is because b' has to intersect a smaller ball centered on such a 1-face (constraint by Separate). It follows that r is larger than a fixed positive constant $\lambda_1 > 0$. If the radii of all vertex balls are larger than λ_1, two balls centered at two different 1-faces intersect only if the larger ball has a radius more than a fixed positive constant $\lambda_2 > 0$. In step 3, a ball is refined only if its radius is more than $\lambda > 0$. Then, the argument in the proof of Lemma 2.5 applies to argue that the tree of refinement for each ball $b \in \mathcal{B}$ is finite. Hence Protect terminates. At termination it must satisfy C1,C2,C3.a, and C3.b since it refines balls with Cover and calls Separate to enforce C3.b. ◻

After the initial protection of $\mathcal{D}_1$, refinement of $\mathcal{D}_2$ begins. In this phase Delaunay refinement is run with a *disk condition* formulated as follows. See Figure 6 in the Appendix for more explanations. Let p be a point on a 2-face σ and let $\mathrm{Umb}_\sigma(p)$ be the set of triangles in $\mathrm{Skl}^2\, S|_\sigma$ that are incident to p.

Disk_Condition(p) : (D1) For each $\sigma \in \mathcal{D}_2$ containing p, the underlying space of $\mathrm{Umb}_\sigma(p)$ is a 2-disk, (D2) point p is in the interior of this 2-disk if and only if $p \in \mathrm{int}\,\sigma$, (D3) in $\mathrm{Umb}_\sigma(p)$, p is not connected to any other point on $\mathcal{D}_1$ which is not adjacent to it, (D4) all vertices of $\mathrm{Umb}_\sigma(p)$ are in σ.

Once the restricted Delaunay triangles are collected, the above checks are only combinatorial. One may notice that D1 and D2 are dual to R1 and R2. We assume that as we insert points, weighted or unweighted, $\mathrm{Vor}\, S$ and $\mathrm{Del}\, S$ get updated appropriately.

```
DelPSC ($\mathcal{D}, \lambda$)
  1. Protection. Let $\mathcal{B}$:= PROTECT($\mathcal{D}, \lambda$). Let $S$ be the current weighted point set.
  2. Mesh2Complex.
     (a) Let $(p, \sigma)$ be any tuple where $p \in \sigma$. Let $t \in \mathrm{Skl}^2\, S|_\sigma$ be the triangle that maximizes size$(t, \sigma)$
         over all $t$ and $\sigma$. Let $x \in V_t|_\sigma$ realize this maximum, say $d_1$. Let $d_2$ be the maximum radius
         of all vertices realized by ball, say $b$.
         If conditions D1 or D2 in Disk_Condition($p$) is violated,
           if $d_1 \geqslant d_2$ insert $x$ into $S$ else compute $\mathcal{B} := \{\mathcal{B} \setminus b\} \cup$ REFINEBALL($b$)
         else if D3 or D4 is violated insert $x$ into $S$.
         Go to step 2(c).
     (b) If size$(t, \sigma) > \lambda$ for some tuple $(t, \sigma)$, where $t \in \mathrm{Skl}^2\, S|_\sigma$, insert $x \in V_t|_{\mathcal{D}}$ that realizes size$(t, \sigma)$
         into $S$.
     (c) If $S$ has grown in the last execution of step 2, repeat step 2.
  3. Return $\bigcup_i \mathrm{Skl}^i\, S|_{\mathcal{D}}$.
```

Notice that in step 2(a) we refine either a ball or a triangle if any of D1, D2 is violated. However, for D3 or D4 violations, we only refine a triangle. This is important because D3 and D4 are not covered by Lemma 2.2 and may be violated no matter how small the balls are. We argue separately for D3,D4 in the termination proof. In step 2(b) we refine triangles to reach the refinement level of the input scale.

We observe that DelPSC never inserts unweighted points inside any protecting ball. If the inserted point x in step 2(a) lies in a protecting ball $b = B(q, r)$, its weighted distance to q would be non-positive. Its weighted distance to its nearest point in S would even be smaller. Since the largest ball has a positive radius, x would not be inserted reaching a contradiction. If x is inserted in step 2(b), the point p is connected to a point q where $q \notin \sigma$. Then, p and q have a positive weighted distance since no two balls from two different 2-faces intersect. Therefore, point x has a positive weighted distance from p and hence cannot be inside a ball.

Guarantees: The analysis of the algorithm establishes two main facts: (i) the algorithm terminates, (ii) at termination the output mesh has guarantees G1 and G2:

(G1) For each $\sigma \in \mathcal{D}_2$, $\mathrm{Skl}^2\, S|_\sigma$ is a 2-manifold with vertices only in σ. Further, $\mathrm{bd}\,\mathrm{Skl}^2\, S|_\sigma$ is homeomorphic to $\mathrm{bd}\,\sigma$ with vertices only in $\mathrm{bd}\,\sigma$.

(G2) There exists a $\lambda > 0$ so that the output mesh of DelPSC($\mathcal{D}$,λ) is homeomorphic to $|\mathcal{D}|$. Further, this homeomorphism respects stratification with vertex restrictions, that is, for $0 \leqslant i \leqslant 2$, $\mathrm{Skl}^i\, S|_\sigma$ is homeomorphic to $\sigma \in \mathcal{D}_i$ where $\mathrm{bd}\,\mathrm{Skl}^i\, S|_\sigma = \mathrm{Skl}^{i-1}\, S|_{\mathrm{bd}\,\sigma}$ and vertices of $\mathrm{Skl}^i\, S|_\sigma$ lie in σ.

We need the following result which says that when a sufficiently small scale is reached, each Voronoi cell V_p contains at least one Voronoi edge intersecting σ if $p \in \sigma \in \mathcal{D}_2$.

LEMMA 3.2. *There exists a $\lambda > 0$ so that σ_p is non-empty for any $p \in \sigma$ if S satisfies the λ-size property.*

THEOREM 3.1. DelPSC *terminates.*

Proof. Consider a vertex p on a 2-face σ. The conclusion of Lemma 2.2 implies disk conditions D1 and D2. Therefore, if it does not hold for p, at least one of the premises of Lemma 2.2 does not hold.

(H1) If σ_p is empty, then according to Lemma 3.2 S does not satisfy the λ_1-size property for some $\lambda_1 > 0$. (H2) S does not satisfy the λ_2-property for some $\lambda_2 > 0$. (H3) Protecting balls do not satisfy C1-C3. But, when the disk condition is checked, the protecting balls satisfy condition C1,C2,C3.a, and C3.b due to Lemma 2.5(ii). Therefore, if H3 has failed, at least one ball has a radius more than λ_3 where λ_3 satisfies Lemma 2.3.

Let d_1 be the maximum size(t, σ) over all t and σ and x be the point realizing this distance. Let d_2 be the maximum radius of all protecting balls. The argument in the previous paragraph implies $\max\{d_1, d_2\} \geqslant \delta = \min\{\lambda_1, \lambda_2, \lambda_3\}$. In DelPSC the point x is inserted if $d_1 \geqslant d_2$. Otherwise, the ball with radius $d_2 \geqslant \delta$ is refined.

The entire ball refinement can be represented with a trees as in the proof of Lemma 3.1 where a ball is refined only if its radius is at least a fixed positive constant. The argument for Lemma 3.1 still holds to claim that only finitely many balls are refined altogether. Therefore, the algorithm cannot refine balls forever. This also implies that the minimum size of the balls remains larger than a fixed positive constant, say $\xi > 0$.

Now we argue that each point inserted by the algorithm maintains a lower bound on its distance to all other points. Then, a standard packing argument implies termination. In step 2(a), each inserted point x maintains a weighted distance at least $\delta > 0$ if it is inserted because of violation of any of D1,D2. If D3 or D4 is violated, the weighted distance of x from p is at least half the weighted distance between p and a point q where either p and q are non-adjacent points in $\mathcal{D}_1$ or q lies on a different 2-face. Since protecting balls have a minimum size $\xi > 0$ and they intersect deeply (C2), the weighted distance between p and q is larger than a fixed positive constant. Hence, x has a distance more than a fixed positive constant from all other points. The only remaining case is step 2(b) where a point is inserted only if its weighted distance is at least $\lambda > 0$ from all other points.

□

4 Proof of guarantees

Let M denote the output mesh of DelPSC.

THEOREM 4.1. *M has guarantee G1.*

Proof. At the end of Mesh2Complex the disk condition ensures that $\mathrm{Skl}^2\, S|_\sigma$ is a simplicial complex where each vertex v belongs to σ and has a 2-disk as its star. It follows from a result in PL topology that $\mathrm{Skl}^2\, S|_\sigma$ is a 2-manifold when DelPSC terminates.

The boundary of $\mathrm{Skl}^2\, S|_\sigma$ has all weighted vertices in $\mathrm{bd}\,\sigma$. Each such point p is connected to its adjacent vertices in $\mathrm{bd}\,\sigma$ by the disk condition. Therefore, the boundary of $\mathrm{Skl}^2\, S|_\sigma$ consists of edges that connect adjacent vertices in $\mathrm{bd}\,\sigma$ and hence this boundary is homeomorphic to $\mathrm{bd}\,\sigma$.

□

To prove G2 we use a result of Edelsbrunner and Shah [17] about the extended topological ball property (TBP). It can be shown that the following two properties P1 and P2 imply the extended TBP [9]. Therefore, according to the Edelsbrunner-Shah [17] result, the underlying space of $\mathrm{Del}\, S|_{\mathcal{D}}$ is homeomorphic to the $|\mathcal{D}|$ if P1 and P2 hold. Let F be a k-face of $\mathrm{Vor}\, S$ where S is the output vertex set.

(P1) If F intersects an element $\sigma \in \mathcal{D}_j \subseteq \mathcal{D}$, the intersection is a closed $(k + j - 3)$-ball.

(P2) There is a unique lowest dimensional element $\sigma_F \in \mathcal{D}$ so that F intersects σ_F and only elements that are incident to σ_F.

Lemma 2.2 almost provides condition P1 except for the case that V_p may intersect a patch τ where $p \notin \tau$ (Figure 7(middle,right)). Lemma 4.1 establishes that this is not possible. Lemma 4.2 gives P2.

Lemma 4.1. *There exists a constant $\lambda > 0$ so that if S has the λ-size property, then for each point p, $V_p|_{\mathcal{D}} = \bigcup_{\sigma \ni p} \sigma_p$.*

Lemma 4.2. *There exists a $\lambda > 0$ so that if S has the λ-size property, then the following holds. Let F be a k-face in $\operatorname{Vor} S$. There is an element $\sigma_F \in \mathcal{D}$ so that F intersects σ_F and only elements in $\mathcal{D}$ that have σ_F on their boundary.*

Theorem 4.2. *M satisfies $G2$.*

Proof. For a sufficiently small $\lambda > 0$, the triangles in $\operatorname{Skl}^2 S|_{\mathcal{D}}$ satisfy the conditions for Lemma 4.1 and Lemma 4.2. This means that properties P1 and P2 are satisfied when λ is sufficiently small. Also when P1 and P2 are satisfied $\bigcup_i \operatorname{Skl}^i S|_{\mathcal{D}} = \operatorname{Del} S|_{\mathcal{D}}$. It follows that the Edelsbrunner-Shah conditions are satisfied for the output M of DelPSC. Thus, M has an underlying space homeomorphic to $|\mathcal{D}|$. The homeomorphism constructed by Edelsbrunner and Shah actually respects the stratification, that is, for each $\sigma \in \mathcal{D}_i$, $\operatorname{Skl}^i S|_\sigma$ is homeomorphic to σ. Also, $\operatorname{Skl}^1 S|_\sigma$ consists of only edges that connect adjacent vertices on σ. Furthermore, property G1 holds for any output of DelPSC. This means, $\operatorname{bd}(\operatorname{Skl}^2 S|_\sigma) = \operatorname{Skl}^1 S|_{\operatorname{bd}\sigma}$. Because of the vertex balls, we also have $\operatorname{bd}(\operatorname{Skl}^1 S|_\sigma) = \operatorname{Skl}^0 S|_{\operatorname{bd}\sigma}$ trivially. Therefore, for $0 \leqslant i \leqslant 2$, $\operatorname{bd}(\operatorname{Skl}^i S|_\sigma) = \operatorname{Skl}^{i-1} S|_{\operatorname{bd}\sigma}$ and $\operatorname{Skl}^i S|_\sigma$ has vertices only in σ. □

5 Conclusions

We have presented a new practical algorithm to mesh a wide variety of geometric domains with the Delaunay refinement technique. Unlike previous approaches, this algorithm computes the protecting balls on the fly and thus gets rid of expensive computations to fix them a priori. The only domain dependent numerical computations are: (i) computing intersection points of the input curves with spheres (to determine the end points of $\operatorname{seg}_\sigma(b)$) and (ii) computing intersection of Voronoi edges with surface patches (to determine restricted triangles and their sizes; these computations are always necessary for the restricted Delaunay mesh [5, 9]). These computations are much easier than normal variation and gap size computations as proposed in [9].

The output mesh maintains a manifold property and with increasing level of refinement captures the topology of the input. In practice, most of the time this level is achieved fast with only the disk condition. An interesting aspect of the algorithm is that the input 'non-smooth features' are preserved in the output. Our experimental results, implemented in CGAL [6], validate our claims (Section A of the appendix). We also showed additional implementation details in a recent video presentation [14].

In applications, sometimes it is desired that the mesh elements have good aspect ratios and their size adapts to the input feature size. Our algorithm can be easily extended to guarantee bounded aspect ratio for most triangles except the ones near non-smooth regions. However, it cannot produce meshes with adaptive size. We note that, since adaptive meshes require an estimate of the input feature size, it may not be possible to produce such meshes without expensive computations. We also note that it should be possible to extend our algorithm to volumes using an approach similar to [19].

Acknowledgment

We acknowledge discussions with Siu-Wing Cheng which helped carrying out this research.

References

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discr. Comput. Geom.* **22** (1999), 481–504.

[2] N. Amenta, S. Choi, T. K. Dey and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *Internat. J. Comput. Geom. Applications* **12** (2002), 125–141.

[3] F. Aurenhammer. Power diagrams: properties, algorithms, and applications. *SIAM J. Computing* **16** (1987),78–96.

[4] J.-D. Boissonnat and S. Oudot. Provably good sampling and meshing of Lipschitz surfaces. *Proc. 22nd Ann. Sympos. Comput. Geom.* (2006), 337–346.

[5] J.-D. Boissonnat and S. Oudot. Provably good surface sampling and meshing of surfaces. *Graphical Models* **67** (2005), 405–451.

[6] CGAL, Computational Geometry Algorithms Library, http://www.cgal.org.

[7] S.-W. Cheng and S.-H. Poon. Three-dimensional Delaunay mesh generation. *Discrete Comput. Geom.* **36** (2006), 419–456.

[8] S.-W. Cheng, T. K. Dey, and J. A. Levine. A practical Delaunay meshing algorithm for a large class of domains. *Proc. 16th Internat. Meshing Roundtable* (2007), 477–494.

[9] S.-W. Cheng, T. K. Dey, and E. A. Ramos. Delaunay refinement for piecewise smooth complexes. *Proc. 18th Ann. ACM-SIAM Sympos. Discrete Algorithms* (2007), 1096–1105.

[10] S.-W. Cheng, T. K. Dey, E. A. Ramos and T. Ray. Sampling and meshing a surface with guaranteed topology and geometry. *Proc. 20th Ann. Sympos. Comput. Geom.* (2004), 280–289.

[11] L. P. Chew. Guaranteed-quality triangular meshes. Report TR-98-983, Comput. Sci. Dept., Cornell Univ., Ithaca, New York, 1989.

[12] L. P. Chew. Guaranteed-quality mesh generation for curved surfaces. *Proc. 9th Ann. Sympos. Comput. Geom.* (1993), 274–280.

[13] K. Clarkson. Building triangulations using ε-nets. *Proc. 38th Sympos. Theory Comput.* (2006).

[14] T. K. Dey and J. A. Levine. DelPSC: A Delaunay mesher for piecewise smooth complexes (multimedia submission). *Proc. 24th Ann. Sympos. Comput. Geom.* (2008), 220–221.

[15] D. Cohen-Steiner, E. C. de Verdière and M. Yvinec. Conforming Delaunay triangulations in 3D. *Proc. Ann. Sympos. Comput. Geom.*, 2002, 199–208.

[16] H. Edelsbrunner. *Geometry and Topology for Mesh Generation.* Cambridge Univ. Press, England, 2001.

[17] H. Edelsbrunner and N. Shah. Triangulating topological spaces. *Internat. J. Comput. Geom. Appl.* **7** (1997), 365–378.

[18] M. Murphy, D. M. Mount and C. W. Gable. A point placement strategy for conforming Delaunay tetrahedralization. *Intl. J. Comput. Geom. Appl.*, 11 (2001), 669–682.

[19] S. Oudot, L. Rineau, and M. Yvinec. Meshing volumes bounded by smooth surfaces. *Proc. 14th Internat. Meshing Roundtable* (2005), 203–219.

[20] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, **18** (1995), 548–585.

[21] J. R. Shewchuk. Mesh generation for domains with small angles. *Proc. 16th Ann. Sympos. Comput. Geom.* (2000), 1–10.

Appendix

A Experimental results

The input to our software is a polygonal model which we assume approximates a PSC. A user specified threshold for dihedral angles is used to select edges of the input as sharp features (elements of $\mathcal{D}_1$) which we protect. Non-manifold and boundary edges are also included as elements of $\mathcal{D}_1$. In Figure 3 we show how our protection algorithm works on four different sets of curves.

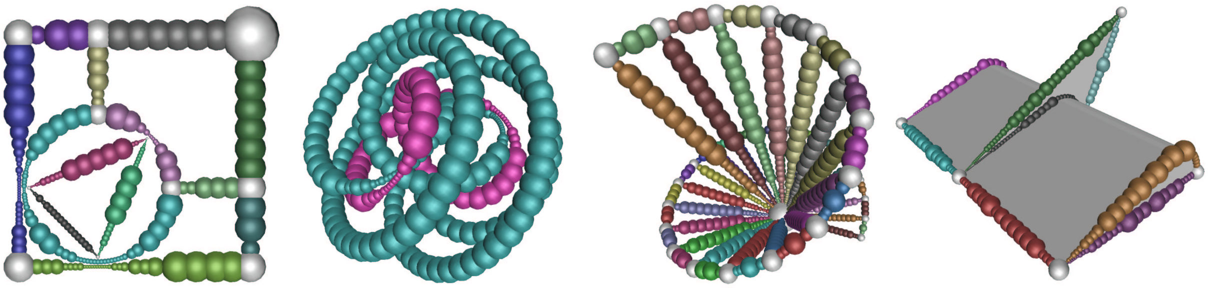

Figure 3: Protection: (left) A 2d example of curve protection. (middle) 2 different 3d examples of curve complexes where we have run PROTECT. (right) The final set of protecting balls returned by Mesh2Complex on the WEDGE model.

In Table 1 we show both the time to protect curves as well as the time to generate the surface mesh for twelve different datasets. All experiments were run on a PC with a 2.8GHz CPU and 2GB RAM. Those datasets which took no time for protection had no sharp features in their input; the PSC they approximate was assumed to be a single smooth patch. The majority of the datasets were meshed in under one minute, only those with complicated topologies took longer.

Dataset	Protection Time	Meshing Time	# of vertices
9 HOLES	0.0 s	105.6 s	8725
ARM	52.6 s	339.7 s	20692
COG	1.4 s	56.2 s	7697
GUIDE	2.6 s	22.9 s	4414
HORN	0.0 s	21.5 s	3192
LOCK	10.6 s	26.6 s	5314
OCTO	0.0 s	7.3 s	1410
PART	0.2 s	23.2 s	4261
PLATE	17.9 s	83.9 s	9773
PUMP	19.1 s	319.6 s	20301
SWIRL	0.0 s	61.1 s	6880
WEDGE	0.2 s	13.2 s	3080

Table 1: Protection and Meshing times for our datasets.

In Figure 4 we show output meshes for various input shapes. These meshes include four smooth shapes (9 HOLES, HORN, OCTO, and SWIRL), two non-manifolds (HORN and WEDGE), and eight piecewise-smooth shapes (PART, GUIDE, WEDGE, COG, ARM, LOCK, PUMP, and PLATE). DelPSC has been used to mesh dozens of other models; further experimentation was presented in a recent multimedia presentation [14].

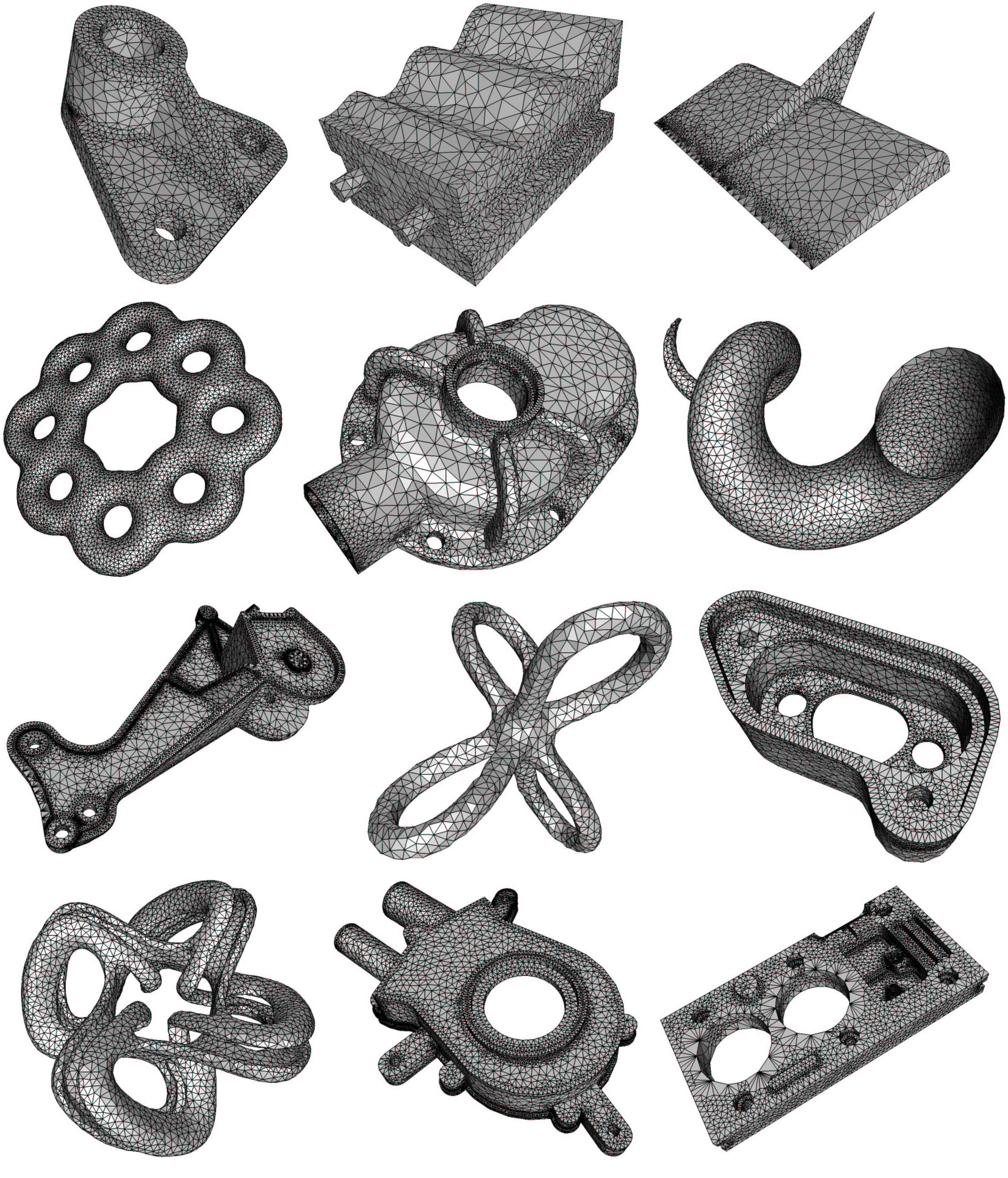

Figure 4: DelPSC output meshes. Left to right, top to bottom: Part, Guide, Wedge, 9 Holes, Cog, Horn, Arm, Octo, Lock, Swirl, Pump, and Plate, models.

B Explanatory Figures

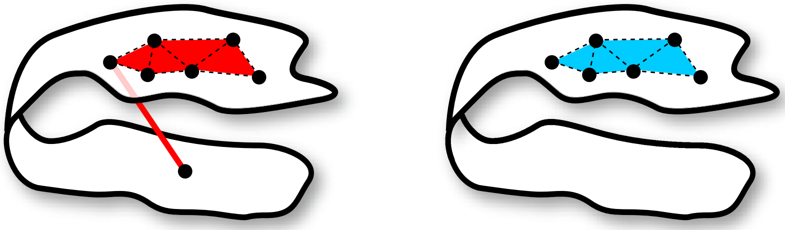

Figure 5: (left) $\mathrm{Del}\, S|_\sigma$ and (right) $\mathrm{Skl}^2\, S|_\sigma$.

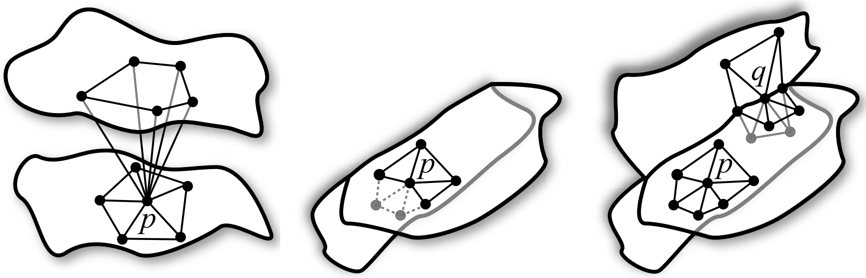

Figure 6: Disk condition: (left) triangles incident to point $p \in \sigma$ and restricted to σ do not form a disk since they form two disks pinched at p violating condition D1, (middle) point $p \in \sigma$ has a topological disk but some of its vertices (lightly shaded) belong to τ violating condition D4, (right) Points p and q satisfy the disk condition. Point p, an interior point in σ, lies in the interior of its disk in σ. Point q, a boundary point, has three disks for each of the three 2-faces.

C Proofs of lemmas in section 2

For proving Lemma 2.2 we need some results from sampling theory [1, 2]. Let $\Sigma \subset \mathbb{R}^3$ be a smooth (C^2-smooth) closed surface. The local feature size $f(x)$ at a point $x \in \Sigma$ is its distance to the medial axis of Σ.

Lemma C.1. ([1, 2]) *Let $\varepsilon \in (0, 1/3)$ be some constant.*

(i) *For any two points x and y in Σ such that $\|x - y\| \leqslant \varepsilon f(x)$,*

  (a) *the angle between the surface normals at x and y is at most $\varepsilon/(1 - 3\varepsilon)$;*

  (b) *the angle between xy and the surface normal at x is at least $\arccos(\varepsilon/2)$.*

(ii) *Let pqr be a triangle with vertices on Σ and circumradius no more than $\varepsilon f(p)$. The angle between the normal of pqr and the surface normal at p is less than 7ε.*

We use Lemma C.1 to prove Lemma 2.2. This result says that if restricted triangles incident to a point in a 2-face σ have small sizes, V_p intersects σ nicely. Figure 7 explains more about its implications.

For any point x on a 2-face σ, we use $n_\sigma(x)$ to denote a unit outward normal to the surface of σ at x.

Proof. [Proof of Lemma 2.2.] Proof of R1: Consider that p is unweighted. For any facet F of V_p, we use H_F to denote the plane of F. Assume that λ is less than 1/32 the local feature size of the surface of σ at p. Let $B = B(p, 2\lambda)$. It is known that $B \cap \sigma$ is a 2-disk.

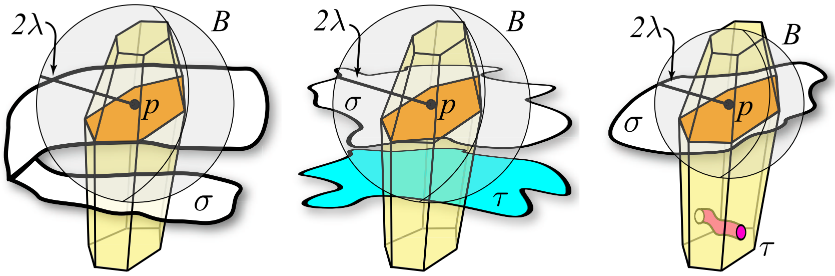

Figure 7: (left) A 2-face σ has intersected V_p where σ_p is not topological disk, this is not possible by R1. (middle) σ_p is a disk though another face τ where $p \notin \tau$ intersects some edge of V_p. This is possible. (right) Within B, σ intersects V_p in a topological disk. It is possible that there is a different component (τ) which does not intersect any Voronoi edge and hence does not contribute any dual restricted triangle incident to p.

First, we claim that for any facet F of V_p, if H_F intersects $B \cap \sigma$, then $\angle n_\sigma(p), H_F \leqslant \arcsin(1/16)$ and both $H_F \cap B \cap \sigma$ and $F \cap B \cap \sigma$ contain no closed curve. The dual Delaunay edge pq of F has length at most 4λ, which is less than 1/8 the local feature size. By Lemma C.1(ib), $\angle n_\sigma(p), H_F = \pi/2 - \angle n_\sigma(p), pq \leqslant \arcsin(1/16)$. There is no closed curve in $H_F \cap B \cap \sigma$ because such a closed curve would bound a 2-disk in $B \cap \sigma$, which would contain a point x such that $\angle n_\sigma(x), H_F = \pi/2$. This is a contradiction because $\angle n_\sigma(x), H_F \leqslant \angle n_\sigma(x), n_\sigma(p) + \angle n_\sigma(p), H_F \leqslant 1/13 + \arcsin(1/16) < \pi/2$ by Lemma C.1(ia). Since $H_F \cap B \cap \sigma$ contains no closed curve, neither does $F \cap B \cap \sigma$. This proves the claim.

Second, we claim that for any facet F of V_p, if H_F is within a distance of λ from p, $H_F \cap B \cap \sigma$ is a single open curve. Consider the disk $H_F \cap B$. Let $\vec{d}$ be the projection of $n_\sigma(p)$ onto H_F. Let

$L \subset H_F$ be the line through the center of $H_F \cap B$ orthogonal to $\vec{d}$. Let x be any point in $H_F \cap B \cap \sigma$. The angle between px and the tangent plane at p is at most $\arcsin(1/32)$ by Lemma C.1(ib). We already proved that $\angle n_\sigma(p), H_F \leqslant \arcsin(1/16)$. So the distance between x and L is less than $\|p - x\| \sin(2 \arcsin(1/16)) \leqslant 2\lambda \sin(2 \arcsin(1/16)) < 0.25\lambda$. Let $L^* \subset H_F$ be the strip of points at distance 0.25λ or less from L. Since the radius of B is 2λ and H_F is at most λ distance from p, the radius of $H_F \cap B$ is at least $\sqrt{3}\lambda$. It follows that the boundary of $H_F \cap B$ intersects L^* in two disjoint circular arcs. We already proved that there is no closed curve in $H_F \cap B \cap \sigma$. It can be shown that if $H_F \cap B \cap \sigma$ contains two open curves, one of the curves, say C, must have both endpoints on the same arc in $H_F \cap B \cap L^*$. The radius of $H_F \cap B$ is at least $\sqrt{3}\lambda$. So some tangent to C must make an angle at most $\arcsin(0.25/\sqrt{3}) < 0.15$ with $\vec{d}$. But this implies that the angle between the surface normal at some point on C and $n_\sigma(p)$ is at least $\pi/2 - 0.15 - \angle n_\sigma(p), H_F \geqslant \pi/2 - 0.15 - \arcsin(1/16) > 1$, contradicting Lemma C.1(ia). This proves the claim.

Third, we claim that for any facet F of V_p, if F intersects $B \cap \sigma$, $F \cap B \cap \sigma$ is a single open curve with endpoints in $\operatorname{bd} F$. We already proved that there is no closed curve in $F \cap B \cap \sigma$. Since F does not have any tangential contact with σ, $F \cap B \cap \sigma$ is a set of open curves and the endpoints of any open curve in $F \cap \sigma$ thus lie in $\operatorname{bd} F$. Assume to the contrary that $F \cap B \cap \sigma$ contains two open curves, say ξ and ξ'. By our assumption, H_F is within a distance of λ from p. We have shown before that $H_F \cap B \cap \sigma$ is a single open curve. Follow $H_F \cap B \cap \sigma$ from ξ to ξ'. When we leave ξ, we must leave F at a Voronoi edge $e \subset \operatorname{bd} F$. Afterward, we stay in the plane H_F and we must cross the support line of e again in order to reach ξ'. Therefore, some tangent to $H_F \cap B \cap \sigma$ is parallel to e. However, the angle between the surface normal at some point on $H_F \cap B \cap \sigma$ and $n_\sigma(p)$ would then be at least $\pi/2 - \angle n_\sigma(p), e \geqslant \pi/2 - 7/32$ by Lemma C.1(ii). This contradicts Lemma C.1(ia). This proves the claim.

Each facet of V_p that intersects $B \cap \sigma$ intersects it in a single open curve. Every curve endpoint is dual to some triangle incident to p. Thus, our assumption (H2) about sizes of restricted triangles incident to p ensures that every curve endpoint lies strictly inside B. This and assumption H1 imply that the facets of V_p intersect $B \cap \sigma$ in a set of simple closed curves. This situation has been analyzed in [10] which shows that exactly one face in this arrangement of closed curves lies inside V_p and it is a 2-disk. Of course, this face is σ_p. This proves that $\sigma_p = V_p \cap B \cap \sigma$ and it is a 2-disk. Take an edge e of V_p that intersects $B \cap \sigma$. By Lemma C.1(ii), $\angle n_\sigma(p), e \leqslant 7/32$. Then, by Lemma C.1(ia), $B \cap \sigma$ is monotone in the direction of e. Therefore, e intersects $B \cap \sigma$ exactly once. It follows that any edge of V_p intersects σ_p at most once. The fact that σ_p intersects a Voronoi facet in a single open curve follows from the third claim.

The above proves the lemma for the case that p is unweighted. The case of p being weighted can be handled similarly by setting B to be $B(p, 2w_p + 2\lambda)$. Recall that the weight w_p is assumed at most λ as S satisfies the λ-size property. Therefore, B would be small enough so that essential arguments of the unweighted case go through here too.

Proof of R2: Since conditions C1-C3 hold (H3), Lemma 2.1 can be applied. The conclusion of this lemma implies R2. ⊒

Proof. [Proof of Lemma 2.4.] COVER affects only the sequence of balls $b_0, b_1, ..., b_k$ as described. By construction the new balls cover entire curve segment $\sigma(x, y)$ and vertex balls are not changed. So, C1 and the first part of C2 are satisfied. To prove the second part consider a ball $b_i = B(c_i, r_i)$. We have following cases:

Case 1 ($i = 0$): We need to examine the effect of the new ball b_1 onto b_0. First notice that $\operatorname{seg}_\sigma(b_0)$ cannot contain c_1 by construction. Therefore, C3.a holds trivially. If b_1 is not created with the end game, the radius of b_1 is α and $d(c_0, c_1) < r_0 + \alpha/3 < r_0 + r_1/3$ where $r_1 = \alpha < r_0$ by construction. This satisfies C2. If b_1 is created with the end game, then two cases occur. If b_1 is the enlarged aiding ball

β_1, its radius is $2\alpha/3$. Then, $d(c_0, c_1) = r_0 < r_0 + r_1/3$. This satisfies C2. In the case where b_1 is not the enlarged aiding ball, its radius is $\frac{7}{6}\alpha$. We can similarly argue that $d(c_0, c_1) < r_0 + \alpha/3 = r_0 + \frac{3}{7}r_1$. So, b_0 satisfies C2.

Case 2 ($i = k$): In this case the center c_{k-1} of b_{k-1} cannot lie in $\mathrm{seg}_\sigma(b_k)$ by construction. So, C3.a holds trivially. The ball b_{k-1} is necessarily created with the end game. If b_{k-1} is enlarged β_{k-1}, $d(c_{k-1}, c_k) < r_k + \alpha/3 = r_k + r_{k-1}/2$ satisfying C2. In the case where b_{k-1} is not the enlarged aiding ball, its radius is $\frac{7}{6}\alpha$. Then, $d(c_k, c_{k-1}) < r_k + \alpha = r_k + \frac{6}{7}r_{k-1}$ satisfying C2.

Case 3 ($i \neq 0$ and $i \neq k$): If b_i is adjacent to b_0 and b_k, it satisfies C2 by arguments in previous two cases. Also, b_i is not large enough to contain c_0 or c_k. Hence C3.a also holds. Now consider the case where b_i is adjacent to another ball b where $b \neq b_0$ and $b \neq b_k$. If neither b_i nor b is created by end game, we have $d(c, c_i) \leqslant r_i + \alpha/3 = r_i + r_i/3$ satisfying C2. Also, in this case $d_\sigma(c, c_i) \geqslant \frac{4}{3}\alpha = \frac{4}{3}r_i$ which satisfies C3.a. We are left with the case when either of b and b_i is created with the end game. With similar arguments, one can check that C2 and C3.a hold in these cases too. ⊓⊔

D Proof of Lemma 3.2

Proof. Suppose that σ_p is empty, that is, no Voronoi edge of V_p intersects σ. By our assumption $\mathrm{bd}\,\sigma$ is non-empty and let $q \in \sigma$ be a weighted point in $\mathrm{bd}\,\sigma$. Because of Lemma 2.1 an edge of V_q has to intersect σ. Consider walking on a path in σ from p to q. Let $p = p_0, p_1, .., p_k = q$ be sequence of vertices whose Voronoi cells are encountered along this walk. Since no edge of V_p intersects σ and some edge of V_q intersects σ, there exists two consecutive vertices p_i and p_{i+1} in this sequence so that no edge of V_{p_i} intersects σ whereas some edge of $V_{p_{i+1}}$ does intersect σ. By Lemma 2.2 we can claim that $\sigma_{p_{i+1}}$ is a disk. A boundary cycle of σ_{p_i} overlaps with the boundary of $\sigma_{p_{i+1}}$. This is impossible as the curves on the boundary of $\sigma_{p_{i+1}}$ intersect Voronoi edges whereas those on the boundary of σ_{p_i} do not. ⊓⊔

E Proofs of lemmas in section 4

Proof. [Proof of Lemma 4.1.] Recall that σ_p denotes the set of connected components in $V_p|_\sigma$ intersecting a Voronoi edge. We are required to prove that $V_p|_{\mathcal{D}} = \bigcup_{\sigma \ni p} \sigma_p$.

Assume to the contrary that $\bigcup \sigma_p \subset V_p|_{\mathcal{D}}$. For any connected component $C \in V_p|_{\mathcal{D}} \setminus \bigcup \sigma_p$, $\mathrm{bd}\,C$ lies inside facets of V_p. Also, V_p can intersect a 1-face only if it contains p (by Lemma 2.1). Therefore, $\mathrm{bd}\,C$ cannot have endpoints implying that $\mathrm{bd}\,C$ is a set of closed curves not intersecting any Voronoi edge.

Let $C \subset \sigma'$. Voronoi cells partition σ'. A path on σ' from $\mathrm{bd}\,C$ to a sample point $q \in \sigma'$ passes through the connected components of this partition. We must encounter two adjacent components along this path, say C' and C'', where $C' \neq \sigma'_s$ for any s and $C'' = \sigma'_r$ for some $r \in \sigma'$. This is because the first and last components satisfy this property. Then, we reach a contradiction since $\mathrm{bd}\,C'' = \mathrm{bd}\,\sigma'_r$ intersects Voronoi edges by Lemma 2.2 whereas $\mathrm{bd}\,C'$ does not. ⊓⊔

Proof. [Proof of Lemma 4.2.] Case 1: F is a Voronoi cell V_p. Let $\sigma_F \in \mathcal{D}$ be the lowest dimensional element containing p. We claim all elements in $\mathcal{D}$ intersecting F have σ_F in their boundaries and thus σ_F is unique. If not, let there be another $\sigma' \in \mathcal{D}$ where $\sigma_F \not\subset \mathrm{bd}\,\sigma'$. Notice that $p \notin \sigma_F \cap \sigma'$ since otherwise $\sigma_F \cap \sigma'$ is either an element of $\mathcal{D}$ whose dimension is lower than σ_F or $\sigma_F \cap \sigma' = \sigma_F$ both of which are impossible. It follows that $p \notin \sigma'$. But we already argued above that V_p intersects only elements in $\mathcal{D}$ that contain p.

Case 2: F is a Voronoi facet V_{pq}. Let σ_F be a lowest dimensional element that F intersects. Assume there is another σ' intersecting F where $\sigma_F \not\subset \mathrm{bd}\,\sigma'$. We go over different dimensions of σ' each time reaching a contradiction.

Assume $\sigma' \in \mathcal{D}_{\leqslant 2}$; σ' intersects F and does not contain σ_F on its boundary. Two cases can arise. Either (i) σ_F and σ' are disjoint within V_p or V_q, or (ii) σ_F and σ' have a common boundary in V_p and V_q. Case (i) cannot happen due to the claim in Case 1. For (ii) to happen both p and q have to be on the common boundaries of σ_F and σ'. This means that p and q have to be on some element in $\mathcal{D}_{\leqslant 1}$. Observe that p and q are non-adjacent since otherwise V_{pq} has to intersect the common boundary of σ_F and σ' whose dimension is lower than that of σ_F. But this would contradict the disk condition that no two non-adjacent vertices in $\mathcal{D}_1$ are connected by a restricted edge.

The above argument implies that all elements intersecting F have σ_F as a subset.

Case 3: F is a Voronoi edge. Certainly F cannot intersect a 2-face σ_F more than once due to Lemma 4.1. The other possibility is that $F = V_t$, $t \in \mathrm{Skl}^2\, S|_{\sigma_F}$, and F intersects $\sigma' \neq \sigma_F$. But then a Voronoi cell adjacent to F would intersect two 2-faces σ_F and σ' and t is in both $\mathrm{Skl}^2\, S|_{\sigma_F}$ and $\mathrm{Skl}^2\, S|_{\sigma'}$ violating the disk condition. □