Collars and Intestines:
Practical Conforming Delaunay Refinement

Alexander Rand and Noel Walkington

Carnegie Mellon University

Summary. While several existing Delaunay refinement algorithms allow acute 3D piecewise linear complexes as input, algorithms producing conforming Delaunay tetrahedralizations (as opposed to constrained or weighted Delaunay tetrahedralizations) often involve cumbersome constructions and are rarely implemented. We describe a practical construction for both “collar” and “intestine”-based approaches to this problem. Some of the key ideas are illustrated by the inclusion of the analogous 2D Delaunay refinement algorithms, each of which differs slightly from the standard approach. We have implemented the 3D algorithms and provide some practical examples.

1 Introduction

Acute input angles pose significant challenges to Delaunay refinement algorithms for quality mesh generation in both two and three dimensions. In 2D, the formation of a conforming mesh is relatively simple: acutely adjacent input segments must be split at equal lengths. Research extending Ruppert’s algorithm [17] to accept small input angles [18, 9] focused on finding algorithms which involve simple modifications of Ruppert’s algorithm and produce the “best” output meshes in practice. In 3D, producing a conforming tetrahedralization of an arbitrary piecewise linear complex (PLC) involves a substantial construction [11, 6] and quality refinement algorithms have been developed in the context of this construction [5, 12]. Alternative algorithms involving weighted [4, 3] and constrained [18, 20, 19] Delaunay tetrahedralization have also been developed. Due to these different challenges, the algorithms for 3D Delaunay refinement of acute domains are markedly different than those in 2D.

In this paper, we describe two related strategies for the protection of acute angles during 3D Delaunay refinement: collars and intestines. The collar approach generalizes the construction of Murphy, Mount, and Gable [11] and that of Cohen-Steiner, Colin de Verdière, and Yvinec [6] and produces a quality mesh following the ideas of Pav and Walkington [12]. The intestine approach is closely related to the quality refinement algorithm of Cheng and Poon [5]. Unlike these previous algorithms for 3D Delaunay refinement of acute input, our algorithms are motivated by analogous 2D versions and, more notably, have been implemented.

Algorithm 1 Quality Refinement of Acute Input

```
(PROTECT) Protect acute input angles.
(REFINE) Perform a protected version of Ruppert's algorithm.
```

Algorithm 1 is the template for both the collar and intestine based refinement algorithms. The (PROTECT) step requires information about the $(d-2)$-dimensional features of the input complex. We note that a brute force computation of this information can be avoided using estimates resulting from certain Delaunay refinement algorithms [14, 15] or an exact computation during sparse Voronoi refinement [10].

Section 2 contains necessary preliminaries for our analysis. Section 3 describes both collar and intestine based Delaunay refinement algorithms in two dimensions, and the three-dimensional algorithms are given in Section 4. Finally, some examples and practical issues are discussed in Section 5.

2 Preliminaries

2.1 Definitions

Our algorithms accept an arbitrary PLC as input and involve an intermediate piecewise smooth complex (PSC), defined below.

Definition 1. *In three dimensions [or two dimensions]:*

- *A* **piecewise linear complex** *(PLC), $C = (\mathcal{P}, \mathcal{S}, \mathcal{F})$ $[(\mathcal{P}, \mathcal{S})]$, is a triple [duple] of sets of input vertices $\mathcal{P}$, input segments $\mathcal{S}$, and polygonal input faces $\mathcal{F}$ such that the boundary of any feature or the intersection of any two features is the union of lower-dimensional features in the complex.*
- *A PLC $C' = (\mathcal{P}', \mathcal{S}', \mathcal{F}')$ $[(\mathcal{P}', \mathcal{S}')]$ is a* **refinement** *of the PLC $C = (\mathcal{P}, \mathcal{S}, \mathcal{F})$ if $\mathcal{P} \subset \mathcal{P}'$, each segment in $\mathcal{S}$ is the union of segments in $\mathcal{S}'$, and every face in $\mathcal{F}$ is the union of faces in $\mathcal{F}'$.*
- *A* **piecewise smooth complex** *(PSC), $C = (\mathcal{P}, \mathcal{S}, \mathcal{F})$ $[C = (\mathcal{P}, \mathcal{S})]$, is a triple [duple] of sets of input vertices $\mathcal{P}$, non-self-intersecting smooth input curves $\mathcal{S}$, and non-self-intersecting smooth input faces $\mathcal{F}$ such that the boundary of any feature or the intersection of any two features is the union of lower-dimensional features in the complex.*

Definition 2. *Let C be a PLC.*

- *The* i**-local feature size** *at point x with respect to C,* $\mathrm{lfs}_i(x, C)$*, is the radius of the smallest closed ball centered at x which intersects two* disjoint *features of C of dimension no greater than i.*
- *The* 1**-feature size** *of segment s with respect to C,* $\mathrm{fs}_1(s, C)$*, is the radius of the smallest closed ball centered at a point $x \in s$ intersecting a segment or input vertex of C which does not intersect s.*

If the argument supplied to the local feature size function is a set of points, rather than a single point, then the result is defined to be the infimum of the function over the set, *i.e.* $\mathrm{lfs}_i(s, C) := \inf_{x \in s} \mathrm{lfs}_i(x, C)$. Often the PLC argument supplied to the local feature size is that of the input complex and in this case the argument will be omitted. The subscript will be omitted when it is equal to $(d-1)$ where d is the dimension, i.e. $\mathrm{lfs}(x) := \mathrm{lfs}_2(x)$ in 3D.

For a PSC we will use the same definition of local feature size as for a PLC. Typically the definition of local feature size for a PSC also involves the radius of curvature or the distance to the medial axis. Since we will only use a very restricted class of PSCs the simpler definition is sufficient. In our particular constructions the radius of curvature is proportional to (and often equal to) the local feature size defined above.

2.2 Generic Delaunay Refinement Algorithm

Algorithm 2 Delaunay Refinement

```
Create an initial Delaunay triangulation.
Queue all unacceptable simplices.
while the queue of simplices is nonempty do
    if it is safe to split the front simplex then
        Take an action based on the front simplex.
        Queue additional unacceptable simplices.
    end if
    Remove the front simplex from the queue.
    Dequeue any queued simplices which no longer exist.
end while
```

The Delaunay refinement algorithms which we will consider have the form of Algorithm 2. Additionally, we require that each of the operations involve only local computations in the Delaunay triangulation of the current vertex set. To specify an algorithm from Algorithm 2, it is necessary to carefully describe the following four statements.

Action	Where should a vertex be inserted to "split" a simplex? Should other simplices be added to the queue?
Priority	In what order should the queue be processed?
Unacceptability	Which simplices are unacceptable?
Safety	Which simplices are safe to split?

3 Delaunay Refinement in 2D

Before describing the full 3D algorithms, analogous 2D Delaunay refinement algorithms are given. The resulting algorithms are similar to those typically used for Delaunay refinement in the presence of acute input angles [18, 9], but avoid certain challenges which are difficult to extend to 3D.

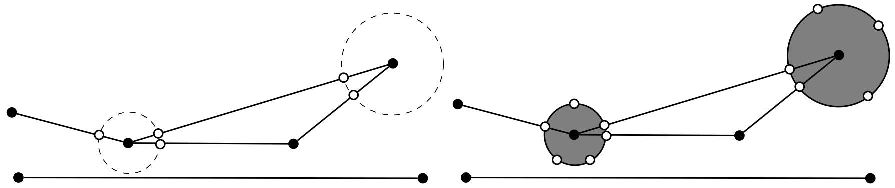
We will describe the two steps in the refinement of an arbitrary PLC given in Algorithm 1: acute input angles are first protected, and then Delaunay refinement is performed. We assume that an appropriate estimate of the local feature size is available at each input vertex. Specifically, we require that for each q_0 which is the vertex of an acute input angle, we are given a distance d_{q_0} which satisfies $b \cdot \text{lfs}(q_0) \leq d_{q_0} \leq \min(c_0 \cdot \text{lfs}_0(q_0), c_1 \cdot \text{lfs}(q_0))$ for some constants $b > 0$, $c_0 \in (0, .5)$ and $c_1 \in (0, 1)$.

3.1 Collar Protection Region

A collar protection region involves forming "collar" segments of equal length around each input vertex so that the Delaunay triangulation conforms to the input near this vertex. The subsequent Delaunay refinement algorithm will then prevent the insertion of any vertices which encroach this collar region.

(PROTECT) Formation of the Protection Region

For each q_0 which is the vertex of an acute input angle, each input segment containing q_0 is split at a distance d_{q_0} away from q_0. Figure 3.1 depicts an example of the points inserted during this step.



(a) Collar approach

(b) Intestine approach

Fig. 1. Two different protection approaches.

Each end segment containing the vertex of an acute input angle will be called a collar simplex and vertices inserted during this step are called collar vertices. First, we observe that the collar simplices are sufficiently far away from disjoint input features of C.

Lemma 1. *For any input point $q_0 \in \mathcal{P}$,*

$$\operatorname{dist}\Big(B(q_0, d_{q_0}), B(q_0', d_{q_0'})\Big) \geq (1 - 2c_0)\,\mathrm{lfs}(q_0) \;\text{ for all } \mathcal{P} \ni q_0' \neq q_0 \text{ and}$$

$$\operatorname{dist}\Big(B(q_0, d_{q_0}), s\Big) \geq (1 - c_1)\,\mathrm{lfs}(q_0) \;\text{ for all segments } s \not\ni q_0.$$

Let α be the smallest angle between adjacent segments in C and let $\bar{C}$ denote the refined PLC obtained after inserting all of the collar vertices. The next lemma quantifies the relationship between the local feature size of $\bar{C}$ and C.

Lemma 2. *There exists $K > 0$ depending only on b and c_0 such that for all x,*

$$\mathrm{lfs}(x, \bar{C}) \leq \mathrm{lfs}(x, C) \leq \frac{K}{\sin(\alpha)}\,\mathrm{lfs}(x, \bar{C}).$$

With the protection region in place, Delaunay refinement can now be performed.

(REFINE) Protected Delaunay Refinement

This step is the Delaunay refinement algorithm described in Algorithm 3 (by specializing Algorithm 2). Each new end segment is "protected" during refinement: no vertices will be inserted in the diametral ball of these segments. To ensure this, circumcenters which encroach these end segments are rejected by the safety criteria. Lemma 1 ensures that no inserted midpoints encroach upon a collar simplex and thus the diametral disk of each collar simplex will be empty throughout the algorithm.

Algorithm 3 2D Delaunay Refinement With Collar

Action	Insert the circumcenter of a simplex unless it causes a lower-dimensional simplex to be unacceptable. In this case, queue the lower-dimensional simplex.
Priority	Segments are given higher priority than triangles.
Unacceptability	A segment with a non-empty diametral disk is unacceptable. A triangle with radius-edge ratio larger than τ is unacceptable.
Safety	Collar simplices are not safe to split.

The termination of the algorithm and properties of the resulting mesh are described in Theorems 1 and 2. The first theorem ensures that the algorithm terminates and the resulting mesh is graded to the local feature size, and the second theorem asserts that the mesh conforms to the input PLC and specifies which triangles near collar simplices may have poor quality.

Theorem 1. *For any $\tau > \sqrt{2}$, there exists $K > 0$ depending only upon τ, b, and c_0 such that for each vertex q inserted by Algorithm 3,*

$$\mathrm{lfs}(q, C) \leq \frac{K}{\sin(\alpha)} r_q.$$

Remark 1. The inequality $\mathrm{lfs}(q, \bar{C}) \leq K r_q$ is shown using an argument identical to the standard analysis of Ruppert's algorithm. Then Lemma 2 yields the desired inequality.

Theorem 2. *Algorithm 3 produces a conforming Delaunay triangulation of C. The circumcenter of any remaining triangle with radius-edge ratio larger than τ lies in the diametral disk of a collar simplex.*

3.2 Intestine Protection Region

The intestine protection region yields the added result that no triangles in the resulting mesh have angles larger than $\pi - 2\kappa$, where $\kappa := \sin^{-1}\left(\frac{1}{2\tau}\right)$ is the minimum angle corresponding to the radius-edge threshold τ.

(PROTECT) Formation of the Protection Region

For each input vertex q_0 at an acute input angle, all input segments containing q_0 are split at a distance d_{q_0} away from q_0. Additionally, vertices are added such that all arcs of the circle centered at q_0 with radius d_{q_0} are no larger than $\frac{\pi}{2}$. This ensures that the diametral ball of each arc of the circle does not contain q_0 and requires at most three additional vertices per input vertex.

We will now consider a PSC $\hat{C}$ defined by the input PLC, vertices inserted on each segment at distance d_{q_0} from each input vertex q_0 (as in the collar protection region), and the boundary arcs of each disk $B(q, d_q)$ as depicted in Figure 3.1. The essential property of the PSC $\hat{C}$ is that all acute angles between features occur between segments of C and are contained in $\bigcup_{q_0} B(q_0, d_{q_0})$. Let α again denote the smallest angle between adjacent segments of C. The local feature sizes with respect to C and $\hat{C}$ are related, as described in the next lemma.

Lemma 3. *There exists $K > 0$ depending only on b, c_0, and c_1 such that for all x,*

$$\mathrm{lfs}(x, \hat{C}) \leq \mathrm{lfs}(x, C) \leq \frac{K}{\sin(\alpha)} \mathrm{lfs}(x, \hat{C}).$$

A suitably sized protection region has been formed and now the subsequent Delaunay refinement algorithm can be described and analyzed.

(REFINE) Protected Delaunay Refinement

Ruppert's algorithm can be performed *outside* of $\bigcup_{q_0} B(q_0, d_{q_0})$ and each of the boundary arcs of any disk $B(q_0, d_{q_0})$ is protected by the diametral disk of its endpoints. This is described completely in Algorithm 4. Refinement of general PSCs in 2D by algorithms similar to Ruppert's has been considered [1, 2, 13] and our analysis follows these developments.

Algorithm 4 2D Delaunay Refinement With Intestine

Action	Insert the circumcenter of a simplex unless it causes a lower-dimensional simplex or arc to be unacceptable. In this case, queue the lower-dimensional object. Insert the midpoint of an arc.
Priority	Segments and arcs are given higher priority than triangles.
Unacceptability	A segment or arc with a non-empty diametral disk is unacceptable. A triangle with radius-edge ratio less than τ is unacceptable.
Safety	All simplices and arcs are safe to split.

Algorithm 4 terminates and produces a conforming graded mesh as described in the following two theorems.

Theorem 3. *For any $\tau > \sqrt{2}$, there exists $K > 0$ depending only upon τ, b, c_0, and c_1 such that for each vertex q inserted by Algorithm 4,*

$$\mathrm{lfs}(q, C) \leq \frac{K}{\sin(\alpha)} r_q.$$

Remark 2. Unlike Theorem 1, the proof of Theorem 3 is substantially more involved than the usual proof for Ruppert's algorithm. This is a result of the smooth input features of $\hat{C}$. Using the techniques of Theorem 1, Theorem 3 can be shown with the strong restriction that $\tau > 2$.

Theorem 4. *Algorithm 4 produces a conforming Delaunay triangulation of C. Any remaining triangle with radius-edge ratio larger than τ is inside $B(q_0, d_{q_0})$ for some input vertex q_0. The resulting triangulation contains no angles larger than $\pi - 2\kappa$.*

4 Delaunay Refinement in 3D

Producing a conforming Delaunay tetrahedralization of a 3D PLC requires a consistent mesh along segments between acutely adjacent features. To initially form this consistent mesh we require the feature size to be known along segments of the input mesh. Given a PLC $C = (\mathcal{P}, \mathcal{S}, \mathcal{F})$, we will assume that we have a refinement $C_1 = (\mathcal{P}_1, \mathcal{S}_1, \mathcal{F})$ such that

(H1) $(\mathcal{P}' \setminus \mathcal{P}) \setminus (\cup_{s\in\mathcal{S}} s) = \emptyset$,
(H2) for any $q_0 \in \mathcal{P}$, all $s_1 \in \mathcal{S}_1$ such that $q_0 \in s_1$ have equal length satisfying $|s_1| \leq c_0 \cdot \mathrm{lfs}_0(q_0)$, and
(H3) for all $s_1 \in \mathcal{S}_1$, $b \cdot \min(\mathrm{fs}_1(s_1), \mathrm{lfs}(s_1)) < |s_1| < c_1 \cdot \min(\mathrm{fs}_1(s_1), \mathrm{lfs}(s_1))$,

where $b > 0$, $c_0 \in (0, .5)$, and $c_1 \in (0, 1)$ are some constants.

4.1 Collar Protection Region

(PROTECT) Formation of the Protection Region

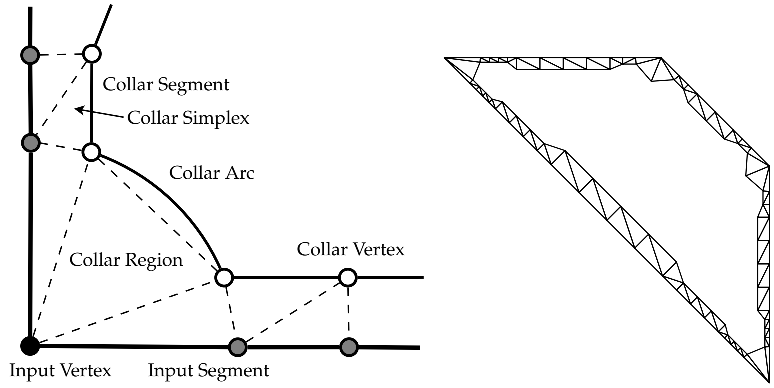
For each input face, the collar is formed by inserting vertices according to the following rules.

1. If s and s' are adjacent non-end segments which meet at vertex q, then a vertex p is inserted at distance $\frac{\max(|s|,|s'|)}{2}$ from q, in any direction into the face perpendicular to s.
2. If s is an end segment and s' is an adjacent non-end segment, both containing vertex q, then insert vertex p at the intersection of any line parallel to s in the face at distance $\frac{|s'|}{2}$ away from s and on the circle of radius $|s|$ around the input point on s.
3. For any input vertex q_0 on a segment s, insert collar vertices such that the sphere of radius $|s|$ around q_0 restricted to the face has no arcs of angle larger than $\frac{\pi}{2}$.

Below is a list of objects defined to describe the collar based on the vertices inserted during this step. These objects are depicted in Figure 2(a).

Collar Vertex	A vertex inserted during the (PROTECT) step or as a midpoint of a collar segment or arc during the (REFINE) step.
Collar Segment	A segment between collar vertices corresponding to adjacent vertices on an input segment.
Collar Arc	An arc between adjacent collar vertices corresponding to the same input vertex.
Collar Region	The region between input segments and collar segments and arcs.
Collar Simplex	A simplex in the Delaunay triangulation of the face which lies inside the collar region.

Following the insertion of the collar vertices, the resulting Delaunay tetrahedralization satisfies a number of properties given in the following lemma.



(a) Collar definitions

(b) Typical collar simplices

Fig. 2. Collar region

Lemma 4. *After inserting collar vertices, the following properties hold.*

(I) All adjacent collar segments and arcs meet at non-acute angles.
(II) The diametral disk of each collar segment contains no vertices in $\mathcal{P}'$.
(III) The circumball of any collar simplex contains no vertices in $\mathcal{P}'$.
(IV) The circumball of any collar simplex does not intersect any disjoint faces or segments.

Remark 3. The circumball of a simplex refers to the *smallest* open sphere such that all vertices of the simplex lie on the boundary of the sphere.

Since the circumball of each collar simplex is empty, the collar simplices conform to the input. Collar segments meet non-acutely and thus the complement of the collar region in each face is well-suited for Ruppert's algorithm. The final property is needed to guarantee that subsequent vertices inserted for conformity will not encroach upon disjoint collar simplices.

The collar divides each face into two regions: the collar region and the non-collar region. Let $\bar{C}$ be the PSC including each face divided into its collar and non-collar regions and all collar segments and arcs. Let α_1 be the smallest angle between an input segment and another adjacent input feature in the mesh and let α_2 be the smallest angle between adjacent input faces. The next lemma asserts that this augmented complex $\bar{C}$ preserves the initial local feature size, up to a factor depending on α_1 and α_2.

Lemma 5. *There exists a constant $K > 0$ depending only upon b, c_0, and c_1 such that*

$$\operatorname{lfs}(x, \bar{C}) \leq \operatorname{lfs}(x, C) \leq \frac{K}{\sin\alpha_1 \sin\alpha_2} \operatorname{lfs}(x, \bar{C}).$$

As usual, the protection procedure is followed by a Delaunay refinement algorithm.

(REFINE) Protected Delaunay Refinement

The PSC $\bar{C}$ is now refined based on both quality and conformity criteria using a modified version of Ruppert's algorithm. Similarly to the non-acute case, any maximum radius-edge threshold $\tau > 2$ can be selected for determining poor quality tetrahedra. The Delaunay refinement algorithm is specified in Algorithm 5.

Algorithm 5 3D Delaunay Refinement With Collar

Action	Insert the circumcenter of a simplex unless it causes a lower-dimensional simplex, collar segment, or collar arc to be unacceptable. In this case, queue the lower-dimensional object. Insert the midpoint of a collar arc.
Priority	Collar segments and arcs are given the highest priority. Other simplices are prioritized by dimension with lower-dimensional simplices processed first.
Unacceptability	A simplex, collar segment, or collar arc is unacceptable if it has a nonempty circumball. A tetrahedron is unacceptable if its radius-edge ratio is larger than τ.
Safety	It is not safe to split any collar simplex (this includes both triangles in input faces and subsegments of input segments).

The key difference between Algorithm 5 and the 3D version of Ruppert's algorithm is the safety criteria. This prevents the cascading encroachment associated with acutely adjacent segments and faces. Since collar arcs must be protected, analysis of the 3D refinement with the *collar* protection scheme is closely related to the 2D refinement with the *intestine* protection scheme.

During the algorithm, it is important to ensure that the properties of the collar in Lemma 4 continue to hold while allowing refinement of the non-collar region of each face to create a conforming mesh. In the 2D collar protection procedure, the collar simplices (i.e. the end segments) never change during Algorithm 3. In 3D however, the set of collar simplices does change. This occurs when the standard Delaunay refinement algorithm seeks to insert a vertex in a face that encroaches upon a collar segment or collar arc. Instead of adding this encroaching vertex, this collar segment or arc is split. This new vertex is a collar vertex and the collar segment or arc is replaced with two new collar segments or arcs. The collar *region* has not changed but the set of collar *simplices* has changed. Further, this new vertex may encroach upon the circumball of another collar simplex in an adjacent face. In this face, the collar segment associated with this encroached circumball is also split so that the collar simplices on adjacent faces again "line up." So conformity of the mesh is maintained by only splitting the collar segments and thus the algorithm never attempts to insert the circumcenter of an encroached collar simplex.

Several key properties hold throughout the algorithm whenever there are no collar segments or collar arcs on the queue.

Lemma 6. *If the queue of unacceptable simplices does not contain any collar segments or collar arcs, the following properties hold.*

(I) Adjacent collar segments and arcs meet at non-acute angles.
(II) The circumball of any collar element contains no vertices in $\mathcal{P}'$.

The first property is important to guarantee the termination of the algorithm, while the second property is important for ensuring the resulting tetrahedralization conforms to the input. These two facts are stated precisely in the next two theorems. Recall that α_1 is the smallest angle between an input segment and an adjacent feature while α_2 is the smallest angle between adjacent input faces.

Theorem 5. *For any $\tau > 2$, there exists $K > 0$ depending only upon τ, b, c_0, and c_1 such that for each vertex q inserted by Algorithm 5,*

$$\text{lfs}(q, C) \leq \frac{K}{\sin \alpha_1 \sin \alpha_2} r_q.$$

Remark 4. Since $\bar{C}$ includes smooth arcs, the proof of Theorem 5 involves many of the techniques used in Theorem 3.

Theorem 6. *Algorithm 5 produces a conforming Delaunay tetrahedralization of C. The circumcenter of any remaining tetrahedra with radius-edge ratio larger than τ lies in the circumball of a collar simplex.*

4.2 Intestine Protection Region

The intestine approach for protecting acute input angles mirrors that in 2D described in Section 3.2. Smooth features will be added to the input to isolate all input segments and vertices (or at least those contained in acutely adjacent features) from the region to be refined for tetrahedron quality.

(PROTECT) Formation of the Protection Region

The vertices and features which are added to the mesh in this step are a superset of those added during the (PROTECT) step of the collar approach (which created the PSC $\bar{C}$). In addition to features of $\bar{C}$, the following objects are included to form a new PSC $\hat{C}$.

- For each input vertex q_0 which belongs to some segment let d_{q_0} be the length of all segments containing q_0. Then $\hat{C}$ includes $\partial B(q_0, d_{q_0})$.

- For each collar segment s let c be the surface of revolution produced by revolving segment s about its associated input segment. The features c and ∂c are included in $\hat{C}$.

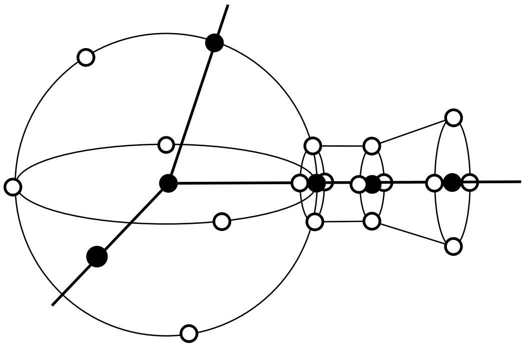


Fig. 3. Intestine Protection Region

The region inside each sphere and cylindrical surface added to the mesh will be called the intestine region and the remaining volume is called the non-intestine region. This is depicted in Figure 3. This construction is designed to ensure the following fact.

Lemma 7. *The non-intestine region of the PSC $\hat{C}$ contains no acute angles between features.*

This lemma is necessary to ensure that the usual proof of termination and grading will apply to Delaunay refinement in the non-intestine region. Let α_1 and α_2 denote the smallest angles in the input as discussed previously.

Lemma 8. *There exists $K > 0$ depending only on b, c_0 and c_1 such that for all x,*

$$\mathrm{lfs}(x, C) \leq \mathrm{lfs}(x, \hat{C}) \leq \frac{K}{\sin\alpha_1 \sin\alpha_2} \mathrm{lfs}(x, C).$$

Remark 5. Recall that $\bar{C}$ is the PSC containing the input and the collar construction. Lemma 8 is shown by first showing

$$\mathrm{lfs}(x, \hat{C}) \leq \mathrm{lfs}(x, \bar{C}) \leq K\,\mathrm{lfs}(x, \hat{C}),$$

and then applying Lemma 5.

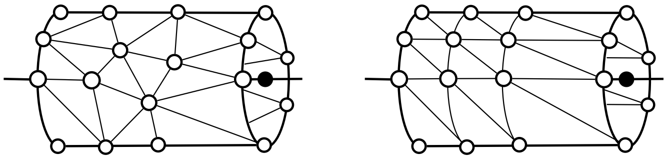
(REFINE) Protected Delaunay Refinement

In a similar fashion to the Delaunay refinement algorithm of Cheng and Poon [5], the PSC $\hat{C}$ has been constructed without any acute angles in the non-intestine region so that Delaunay refinement can be performed. The analysis of this approach involves an understanding of the Delaunay refinement of smooth surfaces in 3D. The intermediate PSC $\bar{C}$ including the collar region is much simpler from this perspective as all 2D faces in the complex are affine. While the collar approach involved elements of the analysis for 2D PSCs, the analysis of the intestine approach more closely resembles the much less complete theory of the refinement of 3D PSCs [5, 3, 16, 7].

We now consider two different approaches to performing a quality refinement of the non-intestine region. The first is to perform the usual Delaunay refinement and split smooth surfaces by projecting the circumcenter of any Delaunay triangle in the face to the surface. This is described in Algorithm 6. This approach suffers from one minor drawback: the Delaunay tetrahedralization inside the cylindrical regions of the intestine may not conform to the input. To eliminate this issue, the second approach is to impose more structure on the refinement of these cylindrical regions. This algorithm is given in Algorithm 7. Figure 4 shows the difference between the refinement around required cylindrical surfaces of the two algorithms.

Algorithm 6 3D Delaunay Refinement With Intestine - Unstructured

Action	Project the circumcenter of a simplex to its associated surface or curve and insert this vertex, unless it causes a lower-dimensional simplex to be unacceptable. In this case, queue the lower-dimensional object.
Priority	Simplices are prioritized by dimension, with lower-dimensional items processed first.
Unacceptability	A simplex in the non-intestine region is unacceptable if it has a nonempty circumball. A tetrahedron is unacceptable if its radius-edge ratio is larger than τ.
Safety	All simplices are safe to split.



(a) Unstructured Approach of Algorithm 6

(b) Structured Approach of Algorithm 7

Fig. 4. Refinement of cylindrical surfaces around the intestine.

These algorithms terminate and produce meshes which are graded to the local feature size. This is summarized in the following theorem.

Algorithm 7 3D Delaunay Refinement With Intestine - Structured

Action	Project the circumcenter of a simplex to its associated surface or curve and insert this vertex, unless it causes a lower-dimensional simplex to be unacceptable. In this case, queue the lower-dimensional object. EXCEPTION: when handling a triangle associated with a cylindrical region which did not yield to another simplex, divide this cylindrical region into two cylinders of equal length and include the new boundary circle in the PSC. Moreover, insert vertices on this circle in the same fashion as in the construction of the intestine region.
Priority	Simplices are prioritized by dimension, with lower-dimensional items processed first.
Unacceptability	A simplex in the non-intestine region is unacceptable if it has a nonempty circumball. A tetrahedron is unacceptable if its radius-edge ratio is larger than τ.
Safety	All simplices are safe to split.

Theorem 7. *For any $\tau > 4$, there exists $K > 0$ depending only upon τ, b, c_0, and c_1 such that for each vertex q inserted by Algorithm 6 or Algorithm 7,*

$$\mathrm{lfs}(q, C) \leq \frac{K}{\sin\alpha_1 \sin\alpha_2} r_q.$$

Remark 6. The restriction $\tau > 4$ is stronger than the restriction $\tau > 2$ seen in Theorem 5. The techniques of Theorem 3 have not yet been extended to the case of curved surfaces, and without these techniques, the stronger condition on τ is necessary. Extending this result to admit all $\tau > 2$ is a topic of ongoing research.

Algorithm 7 produces a conforming Delaunay tetrahedralization of the input. This is shown in the next theorem.

Theorem 8. *Algorithm 7 produces a conforming Delaunay tetrahedralization of C. All tetrahedra with radius-edge ratio larger than τ lie in the intestine region.*

The previous result does not hold for Algorithm 6, as the resulting mesh may not conform to the input. This may occur when a vertex on the boundary of the cylindrical region encroaches upon a triangle in a required face inside the intestine region.

However, a simple conforming (but not Delaunay) tetrahedralization of the intestine region does exist. The spheres around input vertices are tetrahedralized using the Delaunay tetrahedra. For the cylindrical sections, let p_1 and p_2 be the endpoints of the corresponding input segment. The tetrahedralization is produced with two types of tetrahedra.

- For any Delaunay triangle t on the boundary of the cylinder, include the tetrahedron with base t and vertex at p_1.

- For any arc s on the circle around p_2, include the tetrahedra with vertices p_1, p_2 and the endpoints of s.

These tetrahedra are depicted in Figure 5. This construction yields a mesh which conforms to the input. This is summarized in the following theorem.

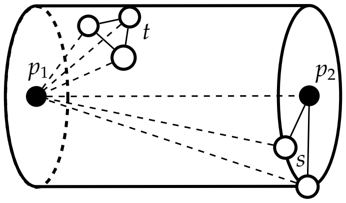


Fig. 5. Two types of tetrahedra are used to produce a conforming tetrahedralization of the intestine region following Algorithm 6.

Theorem 9. *Algorithm 6 produces a conforming Delaunay tetrahedralization of the non-intestine region of $\hat{C}$. The previous construction yields a conforming tetrahedralization of the intestine region of $\hat{C}$ which matches the Delaunay tetrahedralization on the boundary of the intestine region. All tetrahedra with radius-edge ratio larger than τ lie in the intestine region.*

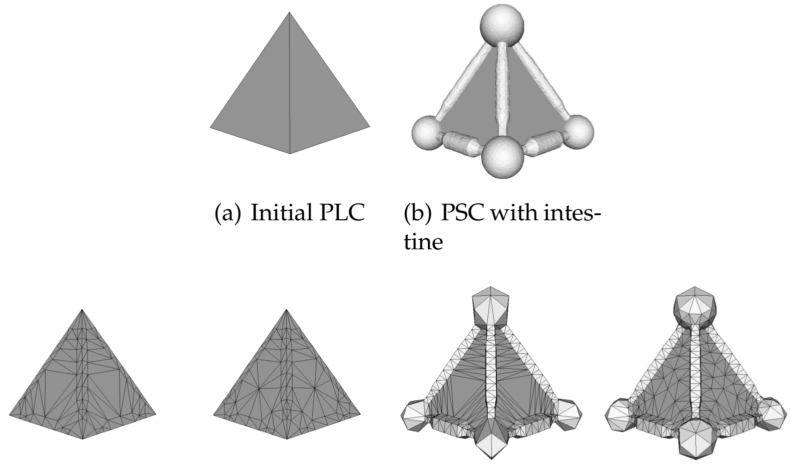
5 Implementation Details and Examples

In 3D, we have implemented both collar and intestine based protection schemes. Our implementation relies on estimates of the local feature size given by a different Delaunay refinement algorithm [14, 15]. Algorithm 6 (rather than Algorithm 7) has been implemented and will be referred to as the intestine approach in the examples below. In the future, we hope to implement both algorithms and do a thorough comparison.

Figure 6 demonstrates both protection strategies on a very simple PLC: a single tetrahedra. Figure 7 shows the refinement of a single face of the pyramid during this refinement using the collar. The result looks very similar when using the intestine approach.

An essential method for reducing the number of vertices in the final mesh is to protect only input segments and vertices which are part of acute input angles. This yields a substantial improvement in the output mesh size. Figure 8 shows an input PLC, the resulting mesh when all segments are protected, and the resulting mesh when only acute input segments are protected. The resulting mesh with full protection contains 18079 vertices while the mesh with partial protection only contains 3216 vertices.

Finally, Figure 9 contains six examples produced by Algorithm 5. Data on the input and output sizes of the meshes produced for each of these examples



(a) Initial PLC (b) PSC with intestine

(c) Initial collar (d) Final collar (e) Initial intestine (f) Final intestine

Fig. 6. Refinement of a simple pyramid.

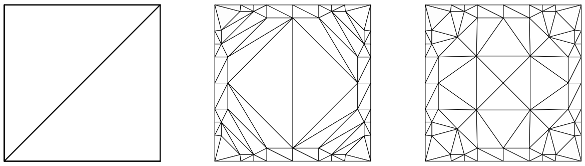
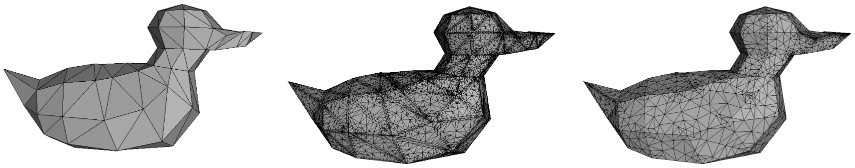


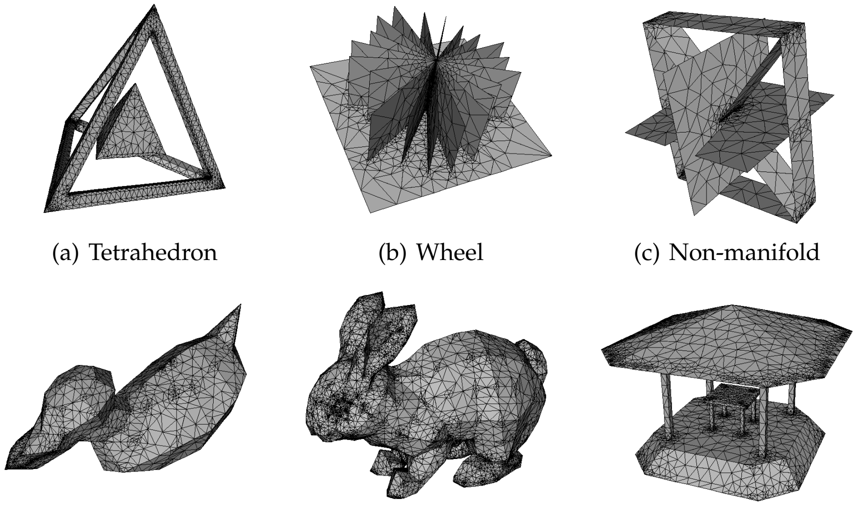
Fig. 7. Refinement of the base of the pyramid.



(a) Input PLC (b) Full protection (c) Partial protection

Fig. 8. Comparison of full and partial collar protection.

is contained in Table 1. Each of the meshes produced only uses the partial collar described above. While the refinement is performed in a bounding box, this bounding box was removed for the PLCs which enclose a volume. This is indicated in the "Box" column of Table 1.



(a) Tetrahedron (b) Wheel (c) Non-manifold

(d) Duck (e) Rabbit (f) Gazebo

Fig. 9. Examples meshes produced by Algorithm 5.

Table 1. Results of Algorithm 5 on six PLCs with acute angles.

	Input			Output		
Name	Vertices	Segments	Faces	Vertices	Tetrahedra	Box
Tetrahedron	24	54	28	3700	11476	No
Wheel	46	65	21	4397	27182	Yes
Non-manifold	22	35	10	2498	15142	Yes
Duck	93	273	182	3216	11001	No
Rabbit	453	1353	902	18968	69001	No
Gazebo	97	148	57	4868	15318	No

References

1. C. Boivin and C. Ollivier-Gooch. Guaranteed-quality triangular mesh generation for domains with curved boundaries. *International Journal for Numerical Methods*

in Engineering, 55(10):1185–1213, 2002.

2. D. E. Cardoze, G. L. Miller, M. Olah, and T. Phillips. A Bezier-based moving mesh framework for simulation with elastic membranes. In *Proceedings of the 13th International Meshing Roundtable*, pages 71–80, 2004.
3. S.-W. Cheng, T. K. Dey, and J. A. Levine. A practical Delaunay meshing algorithm for a large class of domains. In *Proceedings of the 16th International Meshing Roundtable*, pages 477–494, 2007.
4. S.-W. Cheng, T. K. Dey, and E. A. Ramos. Delaunay refinement for piecewise smooth complexes. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1096–1105, 2007.
5. S.-W. Cheng and S.-H. Poon. Three-dimensional Delaunay mesh generation. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 295–304, 2003.
6. D. Cohen-Steiner, E. Colin de Verdière, and M. Yvinec. Conforming Delaunay triangulations in 3D. *Computational Geometry: Theory and Applications*, 28(2-3):217–233, 2004.
7. B. Hudson. Safe Steiner points for delaunay refinement. Research Notes of the 17th International Meshing Roundtable, 2008.
8. G. L. Miller, S. E. Pav, and N. J. Walkington. An incremental Delaunay meshing algorithm. Technical Report 02-CNA-023, Center for Nonlinear Analysis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2002.
9. G. L. Miller, S. E. Pav, and N. J. Walkington. When and why Ruppert's algorithm works. In *Proceedings of the 12th International Meshing Roundtable*, pages 91–102, 2003.
10. G. L. Miller, T. Phillips, and D. Sheehy. Fast sizing calculations for meshing. In *18th Fall Workshop on Computational Geometry*, 2008.
11. M. Murphy, D. M. Mount, and C. W. Gable. A point-placement strategy for conforming Delaunay tetrahedralization. *International Journal of Computational Geometry and Applications*, 11(6):669–682, 2001.
12. S. E. Pav and N. J. Walkington. Robust three dimensional Delaunay refinement. In *Proceedings of the 13th International Meshing Roundtable*, pages 145–156, 2004.
13. S. E. Pav and N. J. Walkington. Delaunay refinement by corner lopping. In *Proceedings of the 14th International Meshing Roundtable*, pages 165–181, 2005.
14. A. Rand and N. Walkington. 3D Delaunay refinement of sharp domains without a local feature size oracle. In *Proceedings of the 17th International Meshing Roundtable*, 2008.
15. A. Rand and N. Walkington. Delaunay refinement algorithms for estimating local feature size, 2009. Submitted.
16. L. Rineau and M. Yvinec. Meshing 3D domains bounded by piecewise smooth surfaces. In *Proceedings of the 16th International Meshing Roundtable*, pages 443–460, 2007.
17. J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18(3):548–585, 1995.
18. J. R. Shewchuk. Mesh generation for domains with small angles. In *Proceedings of the 16th Annual Symposium on Computational Geometry*, pages 1–10, 2000.
19. H. Si. On refinement of constrained Delaunay tetrahedralizations. In *Proceedings of the 15th International Meshing Roundtable*, pages 510–528, 2006.
20. H. Si and K. Gartner. Meshing piecewise linear complexes by constrained Delaunay tetrahedralizations. In *Proceedings of the 14th International Meshing Roundtable*, pages 147–163, 2005.

A Details and Proofs

Below is additional information and details which may be of use to the reviewer. Due to the page limit, only the first 18 pages are to be submitted for inclusion in the IMR proceedings.

A.1 Properties of Local Feature Size

Definition 3. *A PLC* $C^* = (\mathcal{P}^*, \mathcal{S}^*, \mathcal{F}^*)$ *[*$(\mathcal{P}^*, \mathcal{S}^*)$*] is a* **subcomplex** *of the PLC* $C = (\mathcal{P}, \mathcal{S}, \mathcal{F})$ *if either* $C^* = C$ *or there is a feature* $t \in \mathcal{P} \cup \mathcal{S} \cup \mathcal{F}$ *[*$t \in \mathcal{P} \cup \mathcal{S}$*] such that*

$$\begin{aligned} \mathcal{P}^* &= \{p \in \mathcal{P} \,|\, p \subset t\}, \\ \mathcal{S}^* &= \{s \in \mathcal{S} \,|\, s \subset t\}, \text{ and} \\ \mathcal{F}^* &= \{f \in \mathcal{F} \,|\, f \subset t\}. \end{aligned}$$

To protect acute input features, additional vertices, segments, and faces will be added input PLC before performing the Delaunay refinement. The propositions and lemmas below are useful for comparing the local feature size of the input PLC to the local feature size of the protected PLC. The first two propositions consider the refinement and augmentation of PLCs (or PSCs) respectively.

Proposition 1. *Let* $C = (\mathcal{P}, \mathcal{S}, \mathcal{F})$ *[or* $(\mathcal{P}, \mathcal{S})$*] be a PLC and let* $\bar{C} = (\bar{\mathcal{P}}, \bar{\mathcal{S}}, \bar{\mathcal{F}})$ *[or* $(\bar{\mathcal{P}}, \bar{\mathcal{S}})$*] be a refinement of* C*. Let* $\bar{C}_s$ *be some (possibly trivial) subcomplex of* $\bar{C}$*. If*

$$\operatorname{lfs}(y, C) \leq K \operatorname{lfs}(y, \bar{C}_s)$$

for all points y *which belong to some feature of* $\bar{C}_s$ *of dimension at most* $\dim(\bar{C}_s) - 1$*, then*

$$\operatorname{lfs}(x, C) \leq (2K + 1) \operatorname{lfs}(x, \bar{C}_s)$$

holds for all x*.*

Proof. Let x be any point. Let y be the nearest point on a feature of $\bar{C}_s$ to x. Then

$$\begin{aligned} \operatorname{lfs}(x, C) &\leq \operatorname{lfs}(y, C) + |x - y| \\ &\leq K \operatorname{lfs}(y, \bar{C}_s) + |x - y| \\ &\leq K \operatorname{lfs}(x, \bar{C}_s) + (K + 1)|x - y| \\ &\leq (2K + 1) \operatorname{lfs}(x, \bar{C}_s) \end{aligned}$$

Above, we have used the fact that $|x - y| \leq \operatorname{lfs}(x, \bar{C}_s)$ since y is the nearest point to x on any feature of $\bar{C}_s$.

Proposition 2. *Let $C = (\mathcal{P}, \mathcal{S}, \mathcal{F})$ [or $(\mathcal{P}, \mathcal{S})$] and $\hat{C} = (\mathcal{P} \cup \hat{\mathcal{P}}, \mathcal{S} \cup \hat{\mathcal{S}}, \mathcal{F} \cup \hat{\mathcal{F}})$ [or $(\mathcal{P} \cup \hat{\mathcal{P}}, \mathcal{S} \cup \hat{\mathcal{S}})$] be PSCs. Suppose that for all $y \in \hat{\mathcal{P}}$, $y \in s \in \hat{\mathcal{S}}$, or $y \in f \in \hat{\mathcal{F}}$,*

$$\operatorname{lfs}(y, C) \leq K \operatorname{lfs}(y, \hat{C})$$

then for all x

$$\operatorname{lfs}(x, C) \leq (4K + 3) \operatorname{lfs}(x, \hat{C}).$$

Proof. Consider any point y contained in a feature of C. Suppose $\operatorname{lfs}(y, \hat{C}) < \operatorname{lfs}(y, C)$. This implies the existence of a point z either in $\hat{\mathcal{P}}$ or on a segment of $\mathcal{S}$ which is disjoint from the feature containing y such that $|y - z| \leq \operatorname{lfs}(y, \hat{C})$. Hence,

$$\begin{aligned} \operatorname{lfs}(y, C) &\leq \operatorname{lfs}(z, C) + |y - z| \\ &\leq K \operatorname{lfs}(z, \hat{C}) + |y - z| \\ &\leq K \operatorname{lfs}(y, \hat{C}) + (K + 1)|y - z| \\ &\leq (2K + 1) \operatorname{lfs}(y, \hat{C}). \end{aligned}$$

Finally, we can apply Proposition 1 to extend this bound from points on features of $\hat{C}$ to all points and get the desired estimate.

The next lemmas estimate the change in the local feature size which occurs when performing certain types of refinements to PLCs in the presence of acute input angles.

Lemma 9. *Let $C = (\mathcal{P}, \mathcal{S}, \mathcal{F})$ [or $(\mathcal{P}, \mathcal{S})$] be a PLC with refinement $\bar{C} = (\mathcal{P} \cup \bar{\mathcal{P}}, \bar{\mathcal{S}}, \mathcal{F})$ [or $(\mathcal{P} \cup \bar{\mathcal{P}}, \bar{\mathcal{S}})$] such that*

$$\bar{\mathcal{P}} \subset \bigcup_{s \in \mathcal{S}} s^\circ \tag{1}$$

where s° is the relative interior of segment s. For a segment $s \in \mathcal{S}$, let $\bar{C}_s$ be the PLC containing all features of $\bar{C}$ which are contained in s. If for any segment $s \in \mathcal{S}$

$$\operatorname{lfs}(q, C) \leq K \operatorname{lfs}(q, \bar{C}_s)$$

holds for any vertex $q \in s \cap \bar{\mathcal{P}}$, then

$$\operatorname{lfs}(x, C) \leq \left(\frac{4K + 4}{\sin \alpha_1} + 4K + 3 \right) \operatorname{lfs}(x, \bar{C})$$

holds for all x.

Proof. First, let $q_0 \in \mathcal{P}$ and let s be a segment containing q_0. Let q be the nearest vertex to q_0 on s. If $q \in \mathcal{P}$, then

$$\operatorname{lfs}(q_0, C) \leq \operatorname{lfs}(q_0, C_s) = \operatorname{lfs}(q_0, \bar{C}_s).$$

Otherwise, $q \in \bar{\mathcal{P}}$ and it follows that

$$\begin{aligned}\mathrm{lfs}(q_0, C) &\leq \mathrm{lfs}(q, C) + |q - q_0| \\ &\leq K\,\mathrm{lfs}(q, \bar{C}_s) + |q - q_0| \\ &\leq K\,\mathrm{lfs}(q_0, \bar{C}_s) + (K+1)|q - q_0| \\ &\leq (2K+1)\,\mathrm{lfs}(q_0, \bar{C}_s). \end{aligned} \tag{2}$$

Next, let u be a point (not necessarily a vertex in the mesh) contained in some segment of $\mathcal{S}$. The local feature size can be realized several ways. Let w be the point on a disjoint feature of $\bar{C}$ which realizes the local feature size at u with respect to $\bar{C}$.

Case 1. u and w lie on disjoint features of C.

Then $\mathrm{lfs}(u, C) = \mathrm{lfs}(u, \bar{C})$.

Case 2. u and w are contained in the same segment of $s \in \mathcal{S}$.

In this case, apply Proposition 1 using (2) to get the desired inequality:

$$\mathrm{lfs}(u, C) \leq (4K+3)\,\mathrm{lfs}(u, \bar{C}_s) = (4K+3)\,\mathrm{lfs}(u, \bar{C}).$$

Case 3. u and w belong to adjacent segments in C, denoted s_u and s_w.

Let q_0 denote the input point which lies at the intersection of s_u and s_w. Then,

$$\begin{aligned}\mathrm{lfs}(u, C) &\leq \mathrm{lfs}(q_0, C) + |u - q_0| \\ &\leq (2K+1)\,\mathrm{lfs}(q_0, \bar{C}) + |u - q_0| \\ &\leq (2K+1)\,\mathrm{lfs}(u, \bar{C}) + (2K+2)|u - q_0| \\ &\leq \left(2K + 1 + \frac{2K+2}{\sin\alpha}\right)\mathrm{lfs}(u, \bar{C}). \end{aligned}$$

As seen in Figure 10, we have used the fact that

$$\mathrm{lfs}(u, \bar{C}) \geq \mathrm{dist}(u, s_w) \geq \frac{|u - q_0|}{\sin\alpha}.$$

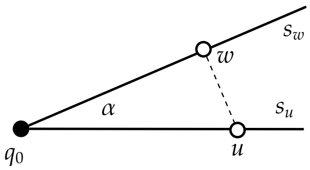


Fig. 10. In Case 3 of Lemma 1, $\mathrm{dist}(p, s_2) \geq \frac{|p - q_0|}{\sin(\alpha)}$.

Finally, applying Proposition 1 on the entire complex $\bar{C}$ with constant K yields the desired result.

Lemma 10. *Let $C = (\mathcal{P}, \mathcal{S}, \mathcal{F})$ be a PLC with refinement $\bar{C} = (\mathcal{P} \cup \bar{\mathcal{P}}, \mathcal{S} \cup \bar{\mathcal{S}}, \bar{\mathcal{F}})$ such that*

$$\bar{\mathcal{P}} \subset \bigcup_{f \in \mathcal{F}} f^\circ$$

and for all $\bar{s} \in \bar{\mathcal{S}}$,

$$\bar{s} \subset \bigcup_{f \in \mathcal{F}} f^\circ$$

where f° is the relative interior of face f. For a face $f \in \mathcal{F}$, let $\bar{C}_f$ be the PLC containing all features of $\bar{C}$ which are contained in f. If for any face $f \in \mathcal{F}$

$$\mathrm{lfs}(y, C) \le K \,\mathrm{lfs}(y, \bar{C}_f)$$

holds for any $y \in f$ which belongs to $\bar{\mathcal{P}}$ or a segment of $\bar{\mathcal{S}}$, then

$$\mathrm{lfs}(x, C) \le \left(\frac{4K+4}{\sin \alpha_2} + 4K + 3 \right) \mathrm{lfs}(x, \bar{C})$$

holds for all x.

Proof. The first step is to show that

$$\mathrm{lfs}(y, C) \le (2K+1)\,\mathrm{lfs}(y, \bar{C}_f)$$

for any $y \in \mathcal{P}$ and $y \in s \in \mathcal{S}$. This is a result of the same argument seen previously. Following the proof of Lemma 9, consider any point u contained a face of $\mathcal{F}$ and let w be the point on a feature of $\bar{C}$ which is disjoint from the feature containing u which realizes the local feature size. This gives three cases:

1. u and w lie on disjoint features of C.
2. u and w are contained in a single face of C.
3. u and w belong to adjacent faces if C.

The first three cases have identical proofs to those of Lemma 9 with one exception: the minimum angle between faces α_2 is used in the third case.

A.2 Section 3 Details

Proof (Lemma 1). This follows directly from the restrictions on c_0 and c_1.

Proof (Lemma 2). Let s be an input segment, and let q be a collar vertex in s. Observe the following inequality:

$$\mathrm{lfs}(q, C) \le \max \left\{ \frac{1-b}{b}, \frac{1-b}{1-2c_0} \right\} \mathrm{lfs}(q, \bar{C}_s).$$

The desired inequality then follows from Lemma 9.

Proof (Theorem 1). The standard analysis of Ruppert's algorithm applies with respect to the protected complex $\bar{C}$ since end segments are never split (which follows from Proposition 1) and thus no segment is encroached by a vertex on an adjacent segment of $\bar{C}$. This asserts the existence of C depending only upon τ such that

$$\text{lfs}(q, \bar{C}) \leq C r_q.$$

The proof is then completed by applying the estimate in Lemma 2 to convert the estimate on the local feature size with respect to $\bar{C}$ into an estimate on the local feature size with respect to C.

Remark 7. While the constant in Theorem 1 does not depend on c_1, the restriction that $c_1 < 1$ is important to ensure the proof is valid: otherwise, it would be possible for vertices to be inserted on segments which encroach upon collar simplices and the proof would not hold.

Proof (Theorem 2). No vertex is inserted which encroaches a collar simplex so all collar simplices conform in the resulting mesh. All unacceptable (and thus non-collar) segments are queued for splitting and none are rejected by the safety rule, thus in the final mesh all segments conform to the input.

Proof (Lemma 3). The first inequality is immediate from the definition of local feature size: adding additional features and refining existing ones only decreases the local feature size. The second inequality will be shown by applying Proposition 2 to $\bar{C}$, the complex with just the initial collar vertices inserted. Let $y \in \partial B(q, d_q)$. Then,

$$\text{lfs}(y, \bar{C}) \leq \frac{2}{1 - c_1} \text{lfs}(y, \bar{C}).$$

Combining the conclusion of Proposition 2 with Lemma 2 yields the result.

Before proving Theorem 3, it is important to understand the application of Ruppert's algorithm to smooth input complexes. Below is a brief discussion and an essential lemma used in this analysis.

Ruppert's algorithm can be generalized to PSCs [1, 2, 13]. The analysis of Pav and Walkington applies to Algorithm 4, however that theory involves a trade-off between the maximum total variation in orientation of the curves in the input and the smallest allowable radius-edge threshold of the refinement. This application leads to two choices:

- Use the input as specified and select a sufficiently large τ to ensure termination.
- Pick any $\tau > \sqrt{2}$. First split smooth input features to have a sufficiently small total variation in orientation based on the τ value selected and then perform Algorithm 4.

However, we will show that this trade-off is unnecessary. Algorithm 4 will terminate and produce a well-graded, conforming Delaunay triangulation for any input complex previously described (such that all input arcs have total variation in orientation of at most $\frac{\pi}{2}$) and any radius-edge threshold $\tau > \sqrt{2}$. This will require a more careful analysis, which is centered around the following technical lemma.

Proposition 3. *Let $\theta \in (0, \frac{\pi}{2}]$ and let a_θ be the arc of a circle $\partial B(q_0, R)$ which subtends an angle θ. Let $x \in \partial B(a_\theta) \setminus B(q_0, R)$, let p be the nearest endpoint of a_θ to x and let q be the projection of x onto $\partial B(a_\theta)$. For any $\tau > \sqrt{2}$ there exists $\beta^*_\tau > 0$, independent of θ, such that if*

$$\frac{|x-q|}{R} \leq \beta^*_\tau,$$

then

$$\frac{|x-p|}{|x-q|} \leq \tau.$$

Remark 8. The most important feature of this lemma is that β^*_τ is independent of θ. This is essential in the proof of termination of the Delaunay refinement algorithm for any $\tau > \sqrt{2}$ without a coupled restriction based on the total variation in orientation of the input curves.

Proof (Proposition 3). Let c be the center of $B(a_\theta)$ and let ϕ denote the angle between segments $\overline{pc}$ and $\overline{xc}$ as shown in Figure 11. Let r denote the radius of $B(a_\theta)$.

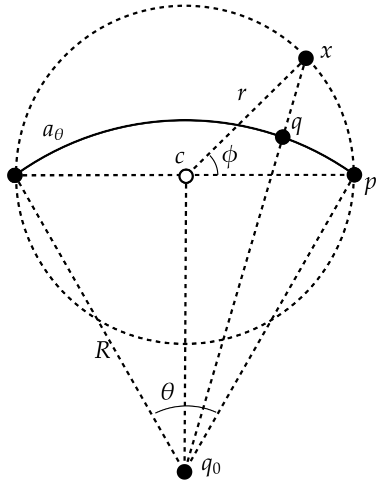


Fig. 11. Configuration in Proposition 3.

Applying the fact that $r = R\sin\left(\frac{\theta}{2}\right)$ gives

$$|x - p| = 2r \sin\left(\frac{\phi}{2}\right) = 2R \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right). \tag{3}$$

Using $|c - q_0| = R \cos\left(\frac{\theta}{2}\right)$ and the law of cosines gives the next sequence of equalities:

$$\begin{aligned}(|x - q| + R)^2 &= r^2 + \left(R \cos\left(\frac{\theta}{2}\right)\right)^2 - 2rR \cos\left(\frac{\theta}{2}\right) \cos\left(\phi + \frac{\pi}{2}\right) \\ &= \left(R \sin\left(\frac{\theta}{2}\right)\right)^2 + \left(R \cos\left(\frac{\theta}{2}\right)\right)^2 \\ &\qquad - 2R^2 \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\phi + \frac{\pi}{2}\right) \\ &= R^2 + R^2 2 \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\phi \\ &= R^2 \left(1 + \sin\theta \sin\phi\right).\end{aligned}$$

Rearranging leads to an expression for $|x - q|$:

$$|x - q| = R\left(\sqrt{1 + \sin(\theta)\sin(\phi)} - 1\right). \tag{4}$$

Combining (3) and (4) gives

$$\frac{|x - p|}{|x - q|} = \frac{2 \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right)}{\sqrt{1 + \sin(\theta)\sin(\phi)} - 1} = \frac{\sqrt{1 + \sin(\theta)\sin(\phi)} + 1}{2 \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right)}. \tag{5}$$

Let $\beta := \frac{|x-q|}{R}$ and $\gamma := \sqrt{2\beta + \beta^2}$. Rearranging terms in (4) yields

$$\gamma^2 = 2\beta + \beta^2 = \sin(\theta)\sin(\phi). \tag{6}$$

This implies that $\gamma \geq \sin\phi$ or $\gamma \geq \sin\theta$. These two possibilities are handled in two cases. However, since (5) is symmetric in the variables ϕ and θ, the argument is identical in each case. Thus, we consider only the case $\gamma \geq \sin\phi$. Then,

$$\sqrt{1 + \sin(\theta)\sin(\phi)} = \sqrt{1 + \gamma^2} \leq 1 + \frac{\gamma^2}{2},$$

and

$$\begin{aligned}\cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right) \geq \cos\left(\frac{\pi}{4}\right) \cos\left(\frac{\sin^{-1}\gamma}{2}\right) &\geq \left(\frac{1}{\sqrt{2}}\right)\left(1 - \frac{\left(\sin^{-1}\gamma\right)^2}{8}\right) \\ &\geq \left(\frac{1}{\sqrt{2}}\right)\left(1 - \frac{\pi^2\gamma^2}{32}\right).\end{aligned}$$

The final inequality results from $\sin^{-1}\gamma \leq \frac{\pi}{2}\gamma$ which requires that $\gamma \leq 1$. To ensure this, we will seek $\beta^*_\tau \leq \frac{1}{3}$. Substituting these estimates into (5) gives

$$\frac{|x-p|}{|x-q|} \leq \sqrt{2}\left(\frac{1+\gamma^2/4}{1-\gamma^2\pi^2/32}\right).$$

Let

$$\gamma^* = 4\sqrt{\frac{2(\tau - \sqrt{2})}{8\sqrt{2}+\pi^2\tau}},$$

and select

$$\beta^*_\tau = \frac{1}{3}\min\left(1,(\gamma^*)^2\right).$$

Substitution yields that if $\frac{|x-q|}{R} \leq \beta^*_\tau$, then

$$\frac{|x-p|}{|x-q|} \leq \tau.$$

Proof (Theorem 3). Recall that $\hat{C}$ denotes the PSC which includes the original input with the protecting circles around each input vertex which are split into arcs subtending at most $\frac{\pi}{2}$. Lemma 3 ensures that it is sufficient to prove the theorem considering local feature size with respect to $\hat{C}$ rather than C. We will show inductively that

$$\text{lfs}(q,\hat{C}) \leq \begin{cases} r_q & \text{if } q \text{ is an input point,} \\ C_1 r_q & \text{if } q \text{ is a segment or arc midpoint,} \\ C_2 r_q & \text{if } q \text{ is a circumcenter.} \end{cases}$$

This inequality is shown in six cases. These cases are distinguished by the type of simplex which is processed to cause q to be inserted and the type of point x which is the nearest neighbor to q when q is inserted.

Case 1. Let q be a midpoint of some subsegment or arc and suppose that x lies on a disjoint feature of the PSC. Then, $\text{lfs}(q) \leq r_q$, so $C_1 \geq 1$ must be required.

Case 2. Let q be the circumcenter of poor quality triangle t which is proposed for insertion. Let p be the newer vertex on the shortest edge of t. Then r_p is at most the length of this short edge which is less than $\frac{|p-q|}{\tau}$, since t is a poor quality triangle. Now, the Lipschitz property of the local feature size can be applied.

$$\begin{aligned} \text{lfs}(q) &\leq \text{lfs}(p) + |p-q| \\ &\leq C_1 r_q + |p-q| \\ &\leq \left(\frac{C_1}{\tau}+1\right)|p-q| \\ &\leq \left(\frac{C_1}{\tau}+1\right)r_q. \end{aligned}$$

This gives the requirement

$$C_2 \geq \frac{C_1}{\tau} + 1. \tag{7}$$

Case 3. Let q be a midpoint of a subsegment and that x is an endpoint of this segment. Then q must have been inserted due to an encroaching (but rejected) circumcenter, c. Local feature size at q is estimated through this vertex c.

$$\begin{aligned} \mathrm{lfs}(q) &\leq \ \mathrm{lfs}(c) + |c - q| \\ &\leq C_2 r_c + |c - q| \\ &\leq \left(C_2 \sqrt{2} + 1\right) r_q. \end{aligned}$$

This gives the requirement

$$C_1 \geq C_2 \sqrt{2} + 1. \tag{8}$$

Case 4. Vertex q is the midpoint of some arc and the nearest neighbor to q lies on an input feature (in $\hat{C}$) which is disjoint from a.

Then $\mathrm{lfs}(q, \hat{C}) \leq |q - x| = r_q$ since this vertex must be on a disjoint feature (with respect to $\hat{C}$ because $\hat{C}$ is non-acute). This yields the (previously required) condition that $C_1 \geq 1$.

Case 5. Vertex q is the midpoint of some arc and the nearest neighbor to q is an endpoint of a.

Let p be an endpoint of the arc a and let y be a point encroaching a which caused a to be split. Let c be the circumcenter of $B(a)$.

If y lies on an input feature, then

$$\mathrm{lfs}(q, \hat{C}) \leq |q - y| \leq |q - c| + |c - y| \leq 2|p - c| \leq 2|p - q| \leq 2r_q. \tag{9}$$

Thus it is sufficient to require that

$$C_1 \geq 2. \tag{10}$$

Otherwise, y is a (possibly rejected) circumcenter of a poor quality triangle.

$$\begin{aligned} \mathrm{lfs}(q) &\leq \ \mathrm{lfs}(y) + |q - y| \\ &\leq C_2 r_y + |q - y| \\ &\leq \ \sqrt{2} C_2 |p - c| + |q - y| \\ &\leq \ \sqrt{2} C_2 |p - q| + |q - y| \\ &\leq (\ \sqrt{2} C_2 + 2) r_q \end{aligned}$$

The fact that $|q - y| \leq 2|q - p|$ follows from the same reasoning as in (9). This gives a restriction that

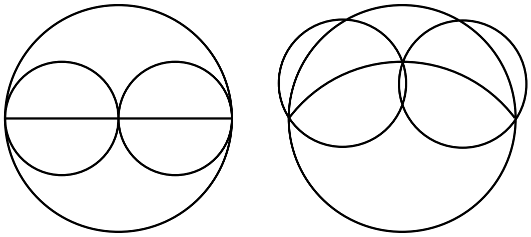


Fig. 12. Diametral balls nest properly when segments are split, while chordal balls do not.

$$C_1 \geq \sqrt{2}C_2 + 2 \tag{11}$$

which is strictly stronger than (10).

Case 6. Vertex q is the midpoint of some arc and the nearest neighbor to q is a circumcenter x in the mesh.

Let arc a lie on a circle centered at input point q_0. Unlike the case of straight line input, an arc may be encroached by a circumcenter in the mesh which was inserted and did not yield to the (larger) arc that was protected at the time. This can occur because the protected chordal balls do not nest when splitting arcs as seen in Figure 12. Let a_θ be the arc containing a in the mesh **when x was inserted** into the mesh. So, $x \in B(q, |q-p|) \setminus B(a_\theta)$ where p is an endpoint of a. Let a_θ be the nearest endpoint of a_θ to x and let θ be the angle subtended by a_θ.

Let $\beta = \frac{|x-q|}{|q-q_0|}$. Now consider two cases depending on the size of β. Let $\beta^*_{\frac{\sqrt{2}+\tau}{2}}$ be the constant given in Proposition 3.

First, suppose that $\beta > \beta^*_{\frac{\sqrt{2}+\tau}{2}}$. Now, we estimate the local feature size at q using the Lipschitz property and the associated input point q_0:

$$\begin{aligned}
\text{lfs}(q, \hat{C}) &\leq |q - q_0| + \text{lfs}(q_0, \hat{C}) \\
&\leq 2|q - q_0| \\
&\leq \frac{2}{\beta^*_{\frac{\sqrt{2}+\tau}{2}}}|x - q| \\
&\leq \frac{2}{\beta^*_{\frac{\sqrt{2}+\tau}{2}}} r_q.
\end{aligned}$$

We must require that

$$C_1 \geq \frac{2}{\beta^*_{\frac{\sqrt{2}+\tau}{2}}}. \tag{12}$$

Next, suppose that $\beta \leq \beta^*_{\frac{\sqrt{2}+\tau}{2}}$. We seek to apply Proposition 3 and assert that

$$\frac{|x-p|}{|x-q|} \leq \frac{\sqrt{2}+\tau}{2}.$$

However, x is not necessarily on the boundary of $B(a)$ and q is not necessarily the projection of x onto a. If $\frac{|x-p|}{|x-q|} \leq 1$ the desired estimate holds. Otherwise, let q' be the projection of x onto a_θ, and let $x' = \overline{q'x} \cap \partial B(a_\theta)$ as shown in Figure 13. Then,

$$\frac{|x-p|}{|x-q|} \leq \frac{|x-p|}{|x-q'|} \leq \frac{|x'-p|}{|x'-q'|}.$$

Moreover,

$$\frac{|x'-q'|}{|q'-q_0|} \leq \frac{|x-q|}{|q-q_0|} \leq \beta^*_{\frac{\sqrt{2}+\tau}{2}}$$

and thus applying Proposition 3 (using x' and q') implies that

$$\frac{|x-p|}{|x-q|} \leq \frac{|x'-p|}{|x'-q'|} \leq \frac{\sqrt{2}+\tau}{2}.$$

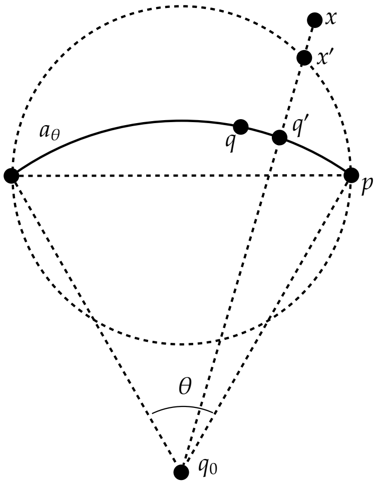


Fig. 13. Diagram for Theorem 3.

Now the local feature size can be estimated:

$$\begin{aligned}
\mathrm{lfs}(q,\hat{C}) &= |x-q| + \mathrm{lfs}(x,\hat{C}) \\
&\le |x-q| + C_2 r_x \\
&\le |x-q| + C_2 |x-p| \\
&\le |x-q| + C_2 \frac{\sqrt{2}+\tau}{2} |x-q| \\
&= \left(1 + C_2 \frac{\sqrt{2}+\tau}{2}\right) r_q.
\end{aligned}$$

Then, we require $C_1 \ge 1 + C_2 \frac{\sqrt{2}+\tau}{2}$. This restriction and several previous ones, (8), (10), and (11), can all be satisfied if

$$C_1 \ge 2 + C_2 \frac{\sqrt{2}+\tau}{2}. \tag{13}$$

Thus, it remains to find constants satisfying the three remaining conditions: (7), (12), and (13). A valid choice of C_1 and C_2 is given below:

$$\begin{aligned}
C_2 &:= \max\left(\frac{4+2\tau}{\tau-\sqrt{2}}, \frac{4-2\beta^*_{\frac{\sqrt{2}+\tau}{2}}}{\beta^*_{\frac{\sqrt{2}+\tau}{2}}(\sqrt{2}+\tau)}\right) \\
C_1 &:= 1 + C_2 \frac{\sqrt{2}+\tau}{2}.
\end{aligned}$$

This completes the proof.

Proof (Theorem 3). These properties are immediate from the definition of the Delaunay refinement algorithm. Note that no triangles contain large angles since all Delaunay triangles inside the protected disks are acute although they can be arbitrarily small if the input contains small angles.

A.3 Section 4 Details

Proof (Lemma 4). Property I follows immediately from the construction. Properties II and IV result from the local feature size bound which we assume the input satisfies. Finally, Property III results from the fact the collar in a face is formed based only upon lengths of subsegments in the associated input segment. Property IV follows from the assumption (H2) on the input complex.

Proof (Lemma 5). The first inequality is immediate for any refinement. For the second inequality, consider an additional intermediate PLC, $\mathcal{C}_0$. Let $\mathcal{C}_0 = (\mathcal{P}_0, \mathcal{S}_0, \mathcal{F})$ where $\mathcal{P}_0$ contains $\mathcal{P}$ and all vertices of $\mathcal{P}_1$ which are adjacent to some input vertex in $\mathcal{P}$ on a segment of $\mathcal{S}_1$. Recall that $\mathcal{P}_1$ is the refined PLC which satisfies the assumptions H1-3. Then $\mathcal{S}_0$ is the refined segments of $\mathcal{S}$ based on the vertices in $\mathcal{P}_0$. Figure 14 depicts the various intermediate PLCs which have been defined. The result will follow by arguing the following list of inequalities.

$$\operatorname{lfs}(x, C_0) \geq k_0 \sin\alpha_1 \operatorname{lfs}(x, C) \tag{14}$$

$$\operatorname{lfs}(x, C_1) \geq k_1 \operatorname{lfs}(x, C_0) \tag{15}$$

$$\operatorname{lfs}(x, \bar{C}) \geq k_2 \sin\alpha_2 \operatorname{lfs}(x, C_1) \tag{16}$$

Lemma 9 implies (14) and Lemma 10 implies (16). Finally (15) follows from the fact that no acutely adjacent features are refined between C_0 and C_1.

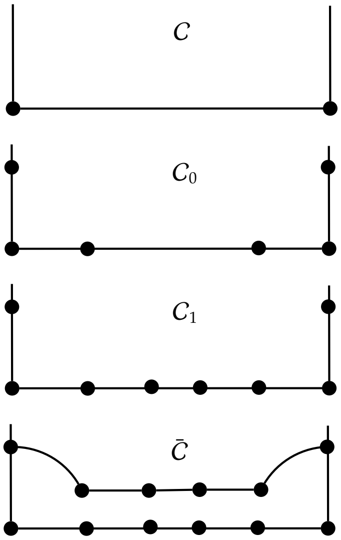


Fig. 14. Intermediate PLCs described in Lemma 5 for a partial example face.

We highlight two propositions which are necessary to ensure the correctness of the algorithm. Both follow from the Delaunay property. The first is given in [8] (which gave a *complete* description of the 3D Delaunay refinement algorithm for non-acute input) and provides the natural analogy to the fact that when splitting a segment, the diametral balls of the new subsegments are contained in the diametral ball of the old segment.

Proposition 4. *[8, Lemma 4.5] Let $\mathcal{T}$ be the Delaunay triangulation of a planar face F such that the circumball of each bounding segment is empty. Let $\mathcal{B}_1$ be the union of the circumballs of all bounding segments of F and $\mathcal{B}_2$ be the union of the circumballs of all triangles in F.*

Let $p \in F \setminus \mathcal{B}_1$. If $\mathcal{T}'$ is the Delaunay triangulation of the face resulting from the addition of p, define $\mathcal{B}_2'$ to be the union of the circumballs of triangles in the resulting Delaunay triangulation. Then,

$$\mathcal{B}_2' \subset \mathcal{B}_1 \cup \mathcal{B}_2.$$

This fact is important in ensuring that when additional collar vertices are inserted the resulting collar simplices are not encroached by other vertices added as circumcenters of poor quality triangles. Collar simplices can be encroached by collar vertices in adjacent faces but these encroachments are removed once encroachment of collar segments in adjacent faces are identical (and thus termination can be ensured).

The next proposition motivates the idea behind protecting the collar segments. This ensures that whenever a triangle in a face is processed on the refinement queue, either the circumcenter is valid to be inserted into the mesh or it encroaches a collar segment, which will be queued for splitting. This is necessary for ensuring conformity of the mesh upon termination of the algorithm.

Proposition 5. *Let A be the set of vertices on the boundary of some face F and let A' be a set of vertices inside F. Suppose that for each boundary segment of F there is a circle through the end points of the segment which does not contain any vertices of $A \cup A'$ in its interior. Then the Delaunay triangulation of $A \cup A'$ conforms to F. Moreover, for any Delaunay triangle t in the interior of F, the circumcenter of t either lies inside F or inside the empty disk associated with a boundary segment.*

Proof (Lemma 6). By applying Lemma 4 to the collar region of each face, conclude that no circumcenter can encroach a collar simplex. Property I then follows because a collar simplex can only be encroached by a vertex on an adjacent collar segment which also encroaches the corresponding collar segment in the face with the original simplex.

Property II results from the fact that nearly formed collar arcs meet at an angle of at least $\frac{\pi}{2}$ since the original arcs were split to subtend angles of at most $\frac{\pi}{2}$.

Proof (Theorem 5). Lemma 5 ensures that it is sufficient to prove the inequality

$$\mathrm{lfs}(q, \bar{C}) \leq C r_q.$$

This standard proof of termination of Ruppert's algorithm is used in conjunction with the techniques in the proof of Theorem 3 to handle the collar arcs around input points.

Acute angles usually cause the the standard proof to fail. This failure occurs when a simplex (a Delaunay triangle in a face or a subsegment of an input segment) is encroached by a vertex on an adjacent input feature. Lemma 6 ensures that this does not occur: collar simplices are never split. If a collar segment is split due to a vertex on another collar segment, this vertex may lie on an adjacent input feature but these collar segments are disjoint features in the PLC $\bar{C}$ ensuring that the original proof of termination for Delaunay refinement holds.

Proof (Theorem 6). These properties are immediate from the description of the algorithm (which ensures remaining poor quality tetrahedra failed the safety criteria) and Lemma 6 (which ensures that collar simplices conform to the input).

Proof (Lemma 5). This follows from Proposition 2.

Proof (Theorem 8). The important task is to verify that the Delaunay tetrahedralization conforms to the input inside the intestine region. Because each triangle on the boundary of the collar is protected, no vertex outside of the intestine region can lie in the diametral ball of a required face. By constructing all vertices on cylinders around concentric rings, no vertex on the boundary of the cylinder can encroach upon the diametral ball of a required face either.

Proof (Theorem 9). This follows by construction.