

GENERALIZED 2D DELAUNAY MESH REFINEMENT

ANDREY N. CHERNIKOV* AND NIKOS P. CHRISOCHOIDES†

Abstract. Delaunay refinement is a popular mesh generation method which makes it possible to derive mathematical guarantees with respect to the quality of the elements. Traditional Delaunay refinement algorithms insert Steiner points in a small enumerable number (one or two) of specific positions inside circumscribed circles of poor quality triangles and on encroached segments. In this paper we prove that there exist entire two-dimensional and one-dimensional regions that can be used for the insertion of Steiner points (innumerable number of choices) while the guarantees on mesh quality can be preserved. This result opens up the possibility to use multiple point placement strategies, all covered by a single proof. In addition, the parallelization of this generalized algorithm immediately implies the parallelization for each individual point placement method.

Key words. Delaunay triangulation, mesh generation

AMS subject classifications. 65D18, 68W05, 68W10, 68N19

1. Introduction. In this paper we build the theoretical foundation for the development of sequential Delaunay mesh generation algorithms and software that satisfy the following requirements:

1. make it possible to use custom point placement strategies;
2. guarantee well-shaped elements with bounded minimal angles;
3. offer proofs of termination and good grading.

Our ongoing research includes the extension of the parallel mesh generation frameworks [3, 4, 14, 15] for use with the generalized sequential analysis presented here, so that multiple custom point placement strategies can be used for parallel guaranteed quality Delaunay refinement.

Delaunay refinement is a popular technique for generating triangular and tetrahedral meshes for use in the finite element method, the finite volume method, and interpolation in various numeric computing areas. Among the reasons for its popularity is the amenability of the method to rigorous mathematical analysis, which makes it possible to derive guarantees on the quality of the elements by proving the termination of the algorithm and the good grading of the mesh.

The field of guaranteed quality Delaunay refinement has been extensively studied, see [6, 7, 11, 16, 18, 19] and the references therein. However, new ideas and improvements keep being introduced. One of the basic questions is where to insert additional (so-called Steiner) points into an existing mesh in order to improve the quality of the elements and/or satisfy other algorithm-specific goals, see [7, 8, 10, 12, 13, 20] for a number of approaches. The goal of this paper is to develop a general framework which allows to use custom point insertion strategies, all covered by a single proof.

In [5] we listed two-dimensional and three-dimensional point insertion methods and suggested the use of two-dimensional and three-dimensional regions, respectively, which we called selection disks. These regions were defined for the highest dimension only, and, hence, we termed the approach semi-generalized. In this paper we develop a fully generalized approach for two dimensions, i.e., we define the selection regions for both one-dimensional elements (segments) and two-dimensional elements (triangles).

*Department of Computer Science, College of William and Mary, Williamsburg, Virginia 23185 (ancher@cs.wm.edu).

†Department of Computer Science, College of William and Mary, Williamsburg, Virginia 23185 (nikos@cs.wm.edu).

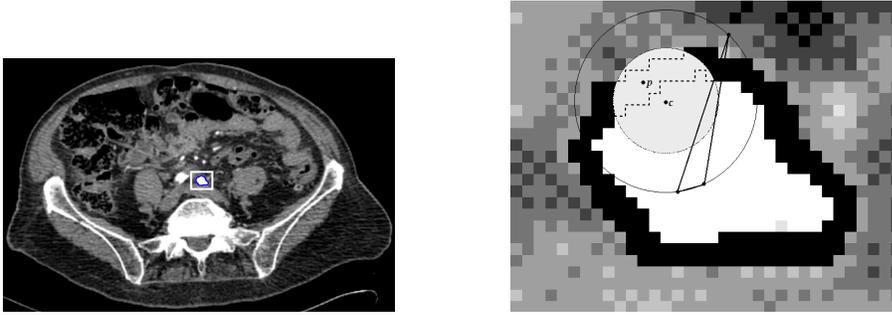


FIG. 1.1. **(Left)** An MRI scan showing a cross-section of a body. **(Right)** A zoom-in of the selected area containing an artery: the inside is white, the outside has different shades of gray, and the black zone is an approximate boundary between these regions. The standard Delaunay refinement algorithm would insert the circumcenter c . However, in order to construct a mesh which conforms to the boundary, another point (p) would be a better choice.

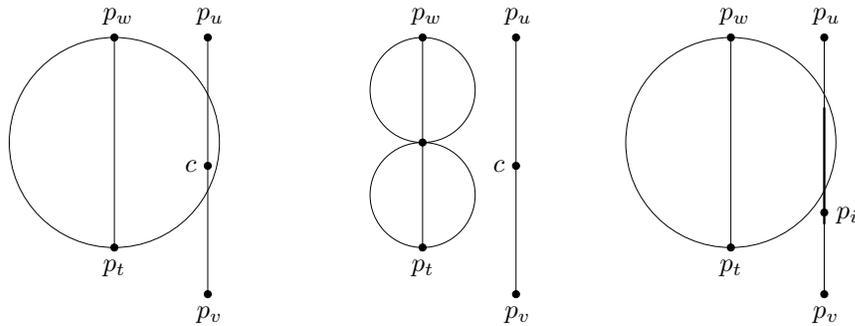


FIG. 1.2. Two constrained segments $e(p_w p_t)$ and $e(p_u p_v)$ are located close to each other and segment $e(p_u p_v)$ is split by a Delaunay refinement algorithm. **(Left)** If the center c of $e(p_u p_v)$ is inserted, c will encroach upon $e(p_w p_t)$, and **(Middle)** $e(p_w p_t)$ will need to be split. **(Right)** However, a point p_i in the selection region of $e(p_u p_v)$ can be chosen instead of c , such that it does not encroach upon $e(p_w p_t)$.

Our future work includes the development of three-dimensional and N -dimensional fully generalized algorithms.

The use of the selection regions can find numerous applications in mesh optimization algorithms for a diverse range of applications. In particular, we are interested in conforming the mesh to the boundary between different materials specified by medical images, see Figure 1.1. In this case, one of the goals of the mesh generation step is to avoid creating edges that would intersect the boundary, which can be achieved by inserting Steiner points inside the boundary zone. One-dimensional selection regions can be used for inserting points along the boundaries according to general distribution requirements. Another mesh optimization example involving one-dimensional selection regions is shown in Figure 1.2.

The rest of the paper is organized as follows. In Section 2 we summarize the essential background required for the understanding of Delaunay refinement algorithms in general. In Section 3 we define the selection circle and the selection interval, as well as our Generalized Delaunay Refinement (GDR) algorithm. In Section 4 we prove a number of lemmas and a point spacing theorem which make it possible to bound the lengths of the edges created by the GDR algorithm in terms of the local feature size

BOWYERWATSON(V, T, p)
Input: V is the set of vertices
 T is the set of triangles
 p is the Steiner point
Output: V and T after the insertion of p
1 $V \leftarrow V \cup \{p\}$
2 $T \leftarrow T \setminus \mathcal{C}(p) \cup \{(p\xi) \mid \xi \in \partial\mathcal{C}(p)\}$

FIG. 2.1. *The Bowyer-Watson point insertion procedure.*

function determined by the input. These results allow us to show that the algorithm cannot produce a sequence of edges of ever decreasing length and therefore, since the area of the domain is bounded, it terminates. In Section 5 we prove the termination of the algorithm, and in Section 6 we prove the good grading of the meshes it produces. Section 8 concludes the paper.

2. Delaunay Refinement Background. Let the input domain Ω be described by a Planar Straight Line Graph (PSLG) [18, 19]. A PSLG \mathcal{X} consists of a set of vertices and a set of straight line segments. Each segment of \mathcal{X} is considered *constrained* and must be preserved during the construction of the mesh, although it can be subdivided into smaller subsegments. The vertices of \mathcal{X} must be a subset of the final set of vertices in the mesh.

Let the mesh $\mathcal{M}_{\mathcal{X}}$ for the given PSLG \mathcal{X} consist of a set $V = \{p_i\}$ of vertices and a set $T = \{t_i\}$ of triangles which connect vertices from V . We denote the triangle with vertices p_u, p_v , and p_w as $\triangle p_u p_v p_w$. We use the symbol $e(p_i p_j)$ to represent the edge of the mesh which connects points p_i and p_j .

As a measure of the quality of elements we use the circumradius-to-shortest edge length ratio specified by an upper bound $\bar{\rho}$. The value of this bound has to be greater than or equal to $\sqrt{2}$ which comes from the ratio of the maximum distance of a point within a circle to either of the end points of the diameter over the radius of the circle, see the proof of Theorem 4.4. This bound in two dimensions is equivalent to a lower bound on a minimal angle [17, 19] since for a triangle with the circumradius-to-shortest edge length ratio ρ and minimal angle A , $\rho = 1/(2 \sin A)$. A mesh is said to satisfy the *Delaunay property* if the circumscribed circle (*circumcircle*) of every triangle does not contain any of the mesh vertices [9].

We use the notion of *cavity* [11] which is the set of elements in the mesh whose circumcircles include a given point p . We denote $\mathcal{C}_{\mathcal{M}}(p)$ to be the cavity of p with respect to mesh \mathcal{M} and $\partial\mathcal{C}_{\mathcal{M}}(p)$ to be the set of boundary edges of the cavity, i.e., the edges which are incident upon only one triangle in $\mathcal{C}_{\mathcal{M}}(p)$. When \mathcal{M} is clear from the context, we will omit the subscript. For our analysis, we use the Bowyer-Watson (B-W) point insertion algorithm [2, 21], which can be written shortly as in Figure 2.1.

In order not to create skinny triangles close to constrained segments and to prevent the insertion of Steiner points outside the domain, Delaunay refinement algorithms observe special *encroachment* rules [18, 19]. In particular, if a Steiner point p is considered for insertion but it lies within the diametral circle of a constrained subsegment s , then a point on s is inserted instead. Traditionally, the midpoint of s is inserted, however, as we show below, there is a whole interval in s from which any point can be selected.

DEFINITION 2.1 (Local feature size [18, 19]). *The local feature size function $\text{lfs}(p)$ for a given point p is equal to the radius of the smallest disk centered at p that*

intersects two non-incident elements of the PSLG.

$\text{lfs}(p)$ satisfies the Lipschitz condition:

LEMMA 2.2 (Lemma 1 in [18], Lemma 2 in [19]). *Given any PSLG and any two points p_i and p_j , the following inequality holds:*

$$\text{lfs}(p_i) \leq \text{lfs}(p_j) + \|p_i - p_j\|. \quad (2.1)$$

Here and in the rest of the paper we use the standard Euclidean norm $\|\cdot\|$.

The following definitions of the insertion radius and of the parent of a Steiner point play a central role in the analysis in [18, 19] and we use them for our analysis in the generalized form, too.

DEFINITION 2.3 (Insertion radius [19]). *The insertion radius $R(p)$ of point p is the length of the shortest edge which would be connected to p if p is inserted into the mesh, immediately after it is inserted. If p is an input vertex, then $R(p)$ is the Euclidean distance between p and the nearest input vertex visible from p .*

Here a vertex is called visible from another vertex if the straight line segment connecting both vertices does not intersect any of the constrained segments.

REMARK 1. *If $e(p_l p_m)$ is an edge in the mesh and p_l was inserted after p_m (or both p_l and p_m are input vertices) then $R(p_l) \leq \|p_l - p_m\|$. Indeed, if $e(p_l p_m)$ was the shortest edge among the edges incident upon p_l at the time p_l was inserted into the mesh (in the case of input vertices, assume that they were inserted simultaneously), then $R(p_l) = \|p_l - p_m\|$ by the definition of the insertion radius; otherwise, $R(p_l) < \|p_l - p_m\|$.*

REMARK 2. *As shown in [19], if p is an input vertex, then $R(p) \geq \text{lfs}(p)$. Indeed, from the definition of $\text{lfs}(p)$, the second feature (in addition to p) which intersects the disk centered at p is either a constrained segment or the nearest input vertex visible from p .*

The following definition of a parent vertex generalizes the corresponding definition in [19]. In our analysis, even though the position of the child is not fixed, the parent is still defined to be the same vertex as in the traditional approach.

DEFINITION 2.4 (Parent of a Steiner point). *The parent \hat{p} of point p is the vertex which is defined as follows: (i) If p is an input vertex, it has no parent. (ii) If p lies on an encroached segment, then \hat{p} is the encroaching point (possibly rejected for insertion). (iii) If p is inserted inside the circumcircle of a poor quality triangle, \hat{p} is the most recently inserted vertex of the shortest edge of this triangle.*

The quantity $D(p)$ is defined as the ratio of $\text{lfs}(p)$ over $R(p)$ [19]:

$$D(p) = \frac{\text{lfs}(p)}{R(p)}. \quad (2.2)$$

It reflects the density of vertices near p at the time p is inserted, weighted by the local feature size. To achieve good mesh grading we would like this density to be as small as possible. If the density is bounded from above by a constant, the mesh is said to have a *good grading* property.

3. Generalized Delaunay Refinement Algorithm.

DEFINITION 3.1 (Selection circle). *For a skinny triangle with circumcenter c , shortest edge length l , circumradius r , and circumradius-to-shortest edge length ratio*

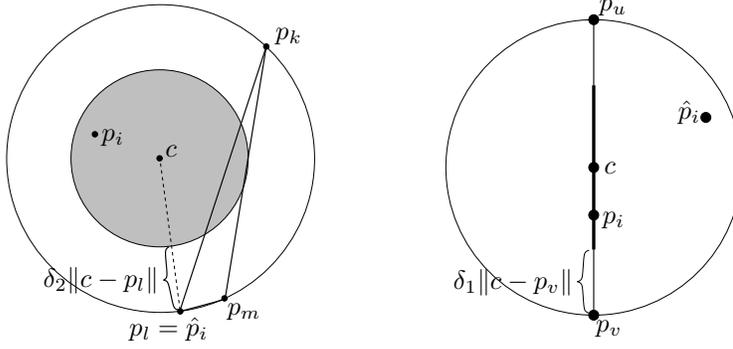


FIG. 3.1. **(Left)** Selection circle (shaded) for the skinny triangle $\triangle p_k p_l p_m$ with the shortest edge $e(p_l p_m)$. Also illustrates case (1) from Table 4.1. **(Right)** Selection interval (bold) for an encroached segment $e(p_u p_v)$. Also illustrates case (2) from Table 4.1.

$\rho = r/l \geq \bar{\rho} \geq \sqrt{2}$, the selection circle is the circle with center c and radius $r(1 - \delta_2)$, where δ_2 is a constant parameter chosen such that

$$\frac{\sqrt{2}}{\bar{\rho}} \leq \delta_2 \leq 1. \quad (3.1)$$

See Figure 3.1(left) for an illustration.

REMARK 3. If $\delta_2 = 1$ then the selection circle shrinks to the circumcenter point.

DEFINITION 3.2 (Selection interval). If s is an encroached segment with center c then the selection interval of s is the subsegment of s with center c and length $|s|(1 - \delta_1)$, where δ_1 is a constant parameter chosen such that

$$\frac{\sqrt{2}}{\bar{\rho}\delta_2} \leq \delta_1 \leq 1 \quad (3.2)$$

and $|s|$ is the length of s . See Figure 3.1(right) for an illustration.

REMARK 4. If $\delta_1 = 1$, then the selection interval shrinks to the center point.

REMARK 5. If $\bar{\rho} = \sqrt{2}$ then both δ_2 and δ_1 can only be equal to 1; therefore, both the selection circles of skinny triangles and the selection intervals of encroached segments shrink to the center points.

REMARK 6. If $\delta_2 = \frac{\sqrt{2}}{\bar{\rho}}$ then δ_1 can only be equal to 1; therefore, the selection intervals of encroached segments shrink to the center points.

If $\delta_1 = 1$ then our Generalized Delaunay Refinement (GDR) algorithm is equivalent to the semi-generalized algorithm which was presented in [5] for both two and three dimensions. This paper lays the foundation for the development of the three-dimensional fully generalized algorithm.

Figure 3.2 presents the GDR algorithm. For brevity, let us abbreviate the types of Steiner points inserted by the GDR algorithm as follows:

- *Type-A*: points within selection circles of skinny triangles,
- *Type-B*: points within selection intervals of encroached constrained segments that do not create input angles less than 90° ,

GENERALIZEDDELAUNAYREFINEMENT(\mathcal{X} , $\bar{\rho}$, δ_2 , δ_1 , $f_2()$, $f_1()$, \mathcal{M})

Input: \mathcal{X} is the PSLG which defines the domain Ω

$\bar{\rho}$ is the upper bound on circumradius-to-shortest edge length ratio,

$$\bar{\rho} \geq \sqrt{2}$$

δ_2 is the parameter which defines selection circles for skinny triangles,

$$\frac{\sqrt{2}}{\bar{\rho}} \leq \delta_2 \leq 1$$

δ_1 is the parameter which defines selection intervals for encroached segments,

$$\frac{\sqrt{2}}{\bar{\rho}\delta_2} \leq \delta_1 \leq 1$$

$f_2()$ and $f_1()$ are user-defined functions which return specific positions

for Steiner points within selection circles and selection intervals, respectively

$\mathcal{M} = (V, T)$ is an initial triangulation of \mathcal{X} , where V is the set of vertices

and T is the set of triangles

Output: A refined Delaunay mesh \mathcal{M} which respects the bound $\bar{\rho}$

```

1  Let SkinnyTriangles be the set of skinny triangles in  $T$ 
2  while SkinnyTriangles  $\neq \emptyset$ 
3    Pick  $t \in$  SkinnyTriangles
4     $p \leftarrow f_2(\delta_2, t)$  //Type-A
5    Let EncroachedSegments be the set of encroached segments
6    if EncroachedSegments  $= \emptyset$ 
7      BOWYERWATSON( $V, T, p$ )
8      Update SkinnyTriangles
9    endif
10   while EncroachedSegments  $\neq \emptyset$ 
11     Pick  $s \in$  EncroachedSegments
12     if  $s$  creates an input angle between  $60^\circ$  and  $90^\circ$ 
13        $p \leftarrow$  midpoint of  $s$  //Type-C
14     else
15        $p \leftarrow f_1(\delta_1, s)$  //Type-B
16     endif
17     BOWYERWATSON( $V, T, p$ )
18     Update EncroachedSegments
19   endwhile
20 endwhile

```

FIG. 3.2. *The Generalized Delaunay Refinement algorithm.*

- *Type-C*: center points of encroached constrained segments that create at least one input angle α , $60^\circ \leq \alpha < 90^\circ$.

For example, in Figure 3.3, segments $e(p_r p_w)$ and $e(p_t p_w)$ create input angles between 60° and 90° , and $e(p_u p_v)$ and $e(p_s p_w)$ do not.

The analysis below assumes that all angles in the input PSLG are not less than 60° . In practice such geometries are rare; however, a modification of the algorithm with concentric circular shell splitting [18, 19] allows to guarantee the termination of the algorithm (but not the good grading) even though the small angles created by the segments of the input PSLG cannot be improved.

4. Point Spacing Theorem. The main result of section is Theorem 4.4 which establishes the relation between the insertion radius of a point and that of its parent or the local feature size. In particular, in both cases the insertion radius is bounded from below and, therefore, the lengths of the edges created by the GDR algorithm are bounded from below. This result allows us to prove in the following sections the termination of the algorithm and the good grading of the meshes it produces.

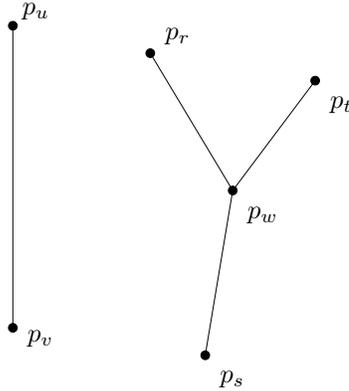


FIG. 3.3. Examples of segments which can be split by the specified types of Steiner points. Segments $e(p_r p_w)$ and $e(p_t p_w)$ can only be split by center points (Type-C), and segments $e(p_u p_v)$ and $e(p_s p_w)$ can be split by arbitrary points within their selection intervals (Type-B).

		\hat{p}_i				
		Type-A	Type-B		Type-C or input	
p_i	Type-A	(1)				
	Type-B	(2)	n/a	(5)	n/a	(5)
	Type-C	(3)	n/a	(6)	(4)	(6)
		adjacent	non-adjacent	adjacent	non-adjacent	

TABLE 4.1

All possible type combinations of p_i and \hat{p}_i . The cells above the labels “adjacent” and “non-adjacent” correspond to the cases when p_i and \hat{p}_i lie on adjacent and non-adjacent segments, respectively. Each of the cases (n) is analyzed separately.

First, we prove Lemmas 4.1, 4.2, and 4.3 that establish important relations used in the proof of Theorem 4.4 as well as in the proof of good grading in Section 6. Lemmas 4.1 and 4.2 bound the insertion radius of a Steiner point from below in terms of the “size” of the two-dimensional and one-dimensional selection region, respectively. Lemma 4.3 deals with the case when the position of the point is fixed to the center of the segment.

LEMMA 4.1. *If Steiner point p_i is of Type-A then*

$$R(p_i) \geq \delta_2 r, \quad (4.1)$$

where r is the circumradius of the corresponding skinny triangle.

Proof. Consider Figure 3.1(left). By the Delaunay property, the circumcircle of the skinny triangle $\triangle p_k p_l p_m$ does not contain any of the mesh vertices in its interior. Therefore, the donut between the boundary of the circumcircle and the boundary of the selection circle is also empty. Thus, the distance from p_i to the closest mesh vertex has to be greater than or equal to the width of the donut. Hence, the insertion radius of p_i has to be greater than or equal to the width of the donut which is equal to $\delta_2 r$. \square

LEMMA 4.2. *If Steiner point p_i is of Type-B, it lies on segment s , and the vertex closest to p_i is one of the endpoints of s , then*

$$R(p_i) \geq \delta_1 \frac{|s|}{2}. \quad (4.2)$$

Proof. By the encroachment rule, the diametral circle of s is empty except for possibly \hat{p}_i , see Figure 3.1(right). Suppose p_v is the end point of s which is the mesh vertex closest to p_i . Then $R(p_i) = \|p_i - p_v\|$ and from the definition of the selection interval, $R(p_i) \geq \delta_1 \frac{|s|}{2}$. \square

LEMMA 4.3. *If p_i is of Type-C and \hat{p}_i is either of Type-C or an input vertex, and, furthermore, p_i and \hat{p}_i lie on adjacent segments, then*

$$R(p_i) = \|p_i - \hat{p}_i\|. \quad (4.3)$$

Proof. Consider Figure 4.1(left). Since p_i is the center of the diametral circle of $e(p_u p_v)$, and by the encroachment rule this circle is empty of mesh vertices except for the encroaching vertex \hat{p}_i , the vertex closest to p_i is \hat{p}_i and (4.3) holds. \square

THEOREM 4.4 (Point Spacing Theorem). *With the use of the GDR algorithm either*

$$R(p_i) \geq C_n \cdot R(\hat{p}_i), \quad n = 1, 2, 3, 4, \quad (4.4)$$

or

$$R(p_i) \geq C_n \cdot \text{lfs}(p_i), \quad n = 5, 6, \quad (4.5)$$

where C_n are defined separately for each of the cases (n) from Table 4.1 as follows: $C_1 = \bar{\rho}\delta_2$, $C_2 = \frac{\delta_1}{\sqrt{2}}$, $C_3 = \frac{1}{\sqrt{2}}$, $C_4 = \frac{1}{2 \cos \alpha_{\min}}$, $C_5 = \frac{\delta_1}{2-\delta_1}$, $C_6 = 1$, where α_{\min} is the minimum input angle.

Proof.

Case (1) By the definition of the parent vertex, \hat{p}_i is the most recently inserted endpoint of the shortest edge of the triangle. Consider Figure 3.1(left). Without loss of generality let $\hat{p}_i = p_l$ and $e(p_l p_m)$ be the shortest edge of the skinny triangle $\triangle p_k p_l p_m$ with circumradius r . Then

$$\begin{aligned} R(p_i) &\geq \delta_2 r && \text{(from Lemma 4.1)} \\ &= \delta_2 \frac{r}{\|p_l - p_m\|} \|p_l - p_m\| \\ &= \delta_2 \rho \|p_l - p_m\| \\ &\geq \delta_2 \bar{\rho} \|p_l - p_m\| && \text{(since } \rho \geq \bar{\rho} \text{)} \\ &\geq \delta_2 \bar{\rho} R(p_l) && \text{(from Remark 1)} \\ &= \delta_2 \bar{\rho} R(\hat{p}_i); \end{aligned}$$

therefore, (4.4) holds with $C_1 = \bar{\rho}\delta_2$.

The argument above holds for all types of \hat{p}_i because it does not involve the properties of \hat{p}_i specific for a particular type.

Case (2) Let \hat{p}_i encroach upon the constrained segment $e(p_u p_v)$, see Figure 3.1(right). Then

$$\begin{aligned} R(\hat{p}_i) &\leq \min\{\|\hat{p}_i - p_u\|, \|\hat{p}_i - p_v\|\} && \text{(from Remark 1)} \\ &\leq \sqrt{2} \frac{\|p_u - p_v\|}{2} && \text{(because } \hat{p}_i \text{ encroaches upon } e(p_u p_v)) \\ &\leq \sqrt{2} \frac{R(p_i)}{\delta_1} && \text{(from Lemma 4.2, since } \hat{p}_i \text{ is rejected);} \end{aligned}$$

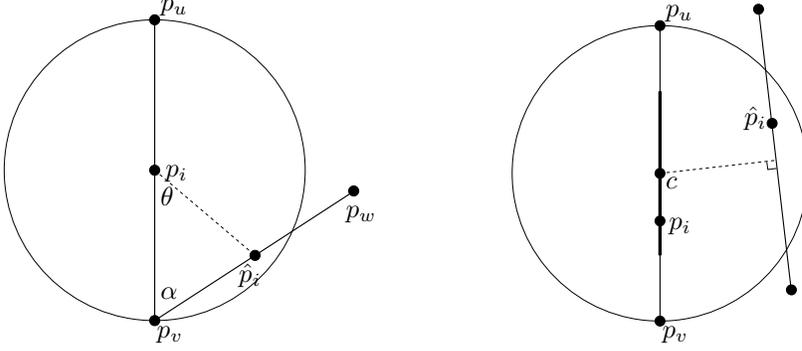


FIG. 4.1. (Left) p_i and \hat{p}_i lie on adjacent segments separated by angle α , $60^\circ \leq \alpha < 90^\circ$ (case (4) from Table 4.1). (Right) p_i and \hat{p}_i lie on non-adjacent segments (case (5) from Table 4.1).

therefore, (4.4) holds with $C_2 = \frac{\delta_1}{\sqrt{2}}$.

Case (3) This is a special case of Case (2), with $\delta_1 = 1$. Thus, (4.4) holds with $C_3 = \frac{1}{\sqrt{2}}$.

Case (4) This case was proved in [19]. We present the proof using our notation for the completeness of the paper.

Consider Figure 4.1(left). Let the adjacent constrained segments $e(p_u p_v)$ and $e(p_v p_w)$ be separated by an angle α , $60^\circ < \alpha < 90^\circ$. Then

$$\begin{aligned} \frac{R(p_i)}{R(\hat{p}_i)} &= \frac{\|p_i - \hat{p}_i\|}{R(\hat{p}_i)} && \text{(from Lemma 4.3)} \\ &\geq \frac{\|p_i - \hat{p}_i\|}{\|\hat{p}_i - p_v\|} && \text{(from Remark 1)} \\ &= \frac{\sin \alpha}{\sin \theta} && \text{(considering } \triangle p_i p_v \hat{p}_i \text{).} \end{aligned}$$

Considering α fixed and the position of \hat{p}_i variable, the fraction $\frac{\sin \alpha}{\sin \theta}$ is minimized when θ is maximized (since $0 < \theta < 90^\circ$) or, equivalently, when \hat{p}_i lies on the boundary of the diametral circle of $e(p_u p_v)$. In that case

$$R(p_i) = \|p_i - \hat{p}_i\| = \|p_i - p_v\| = r, \quad (4.6)$$

where r is the radius of the diametral circle of $e(p_u p_v)$, and from the isosceles triangle $\triangle \hat{p}_i p_i p_v$,

$$\|\hat{p}_i - p_v\| = 2r \cos \alpha. \quad (4.7)$$

Therefore, noting (4.6) and (4.7), the fraction $\frac{\|p_i - \hat{p}_i\|}{\|\hat{p}_i - p_v\|}$ has a lower bound of $\frac{1}{2 \cos \alpha}$, i.e.,

$$R(p_i) \geq \frac{1}{2 \cos \alpha} R(\hat{p}_i).$$

Therefore, (4.4) holds with $C_4 = \frac{1}{2 \cos \alpha_{\min}}$, where α_{\min} is the minimum input angle.

Case (5) Let \hat{p}_i lie on a constrained segment (including the case when it is an end point of this segment) and encroach upon a non-adjacent constrained segment $s = e(p_u p_v)$, see Figure 4.1(right), and p_i lie in the selection interval of s . Then there are two possibilities:

- (a) If \hat{p}_i is the vertex closest to p_i then $R(p_i) = \|p_i - \hat{p}_i\| \geq \text{lfs}(p_i)$ by Definition 2.1 of the $\text{lfs}(\cdot)$ function.

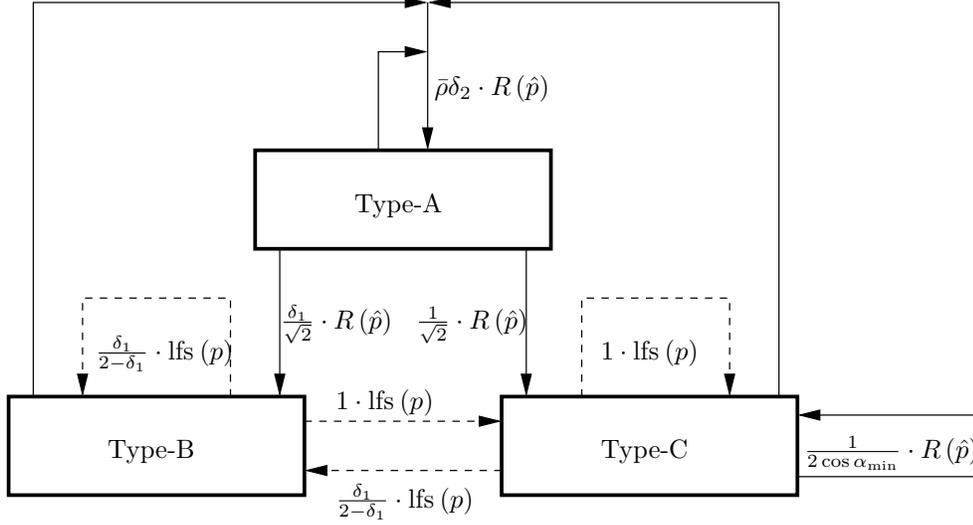


FIG. 5.1. A diagram illustrating the relationship between the insertion radius $R(p)$ of a vertex p and the insertion radius $R(\hat{p})$ of its parent \hat{p} . The head of each arrow points to the box marked with the type of p and the tail leaves from the box marked with the type of \hat{p} . Solid arrows are labeled with the minimum value of $R(p)$ in terms of $R(\hat{p})$. Dashed arrows are labeled with the minimum possible value of $R(p)$. The GDR algorithm terminates because no solid cycle has a product of constant multipliers smaller than one. Input vertices are not shown since they do not participate in cycles.

- (b) Otherwise, because the diametral circle of s is empty except for \hat{p}_i , one of the endpoints of s must be the vertex closest to p_i . Then, if c is the center of s , by Definition 2.1 of the $\text{lfs}(\cdot)$ function,

$$\text{lfs}(c) \leq \|c - \hat{p}_i\|. \quad (4.8)$$

Therefore,

$$\begin{aligned} \text{lfs}(p_i) &\leq \text{lfs}(c) + \|p_i - c\| && \text{(from Lemma 2.2)} \\ &\leq \|c - \hat{p}_i\| + \|p_i - c\| && \text{(from (4.8))} \\ &\leq \frac{|s|}{2} + \|p_i - c\| && \text{(because } \hat{p}_i \text{ encroaches upon } s) \\ &\leq \frac{|s|}{2} + (1 - \delta_1) \frac{|s|}{2} && \text{(since } p_i \text{ lies in the selection interval of } s) \\ &= (2 - \delta_1) \frac{|s|}{2} \\ &\leq (2 - \delta_1) \frac{R(p_i)}{\delta_1} && \text{(from Lemma 4.2).} \end{aligned}$$

In both cases, $C_5 = \frac{\delta_1}{2 - \delta_1}$ satisfies the inequality (4.5).

The argument above holds for each type of \hat{p}_i (Type-B, Type-C, or input) because it does not involve the properties of \hat{p}_i specific for a particular type.

Case (6) This is a special case of Case (5), with $\delta_1 = 1$. Thus, the inequality (4.5) can be satisfied with $C_6 = 1$. \square

5. Proof of Termination.

THEOREM 5.1. *The GDR algorithm terminates.*

Proof. To prove the termination, we need to prove that the GDR algorithm does not create edges of ever decreasing length. Then, from the fact that in a Delaunay

mesh each vertex is always connected by an edge to its nearest neighbor [1], it follows that there exists an empty circle around each vertex of radius equal to the distance to its nearest neighbor. Since one can pack only a finite number of such circles in the bounded domain, the algorithm will eventually terminate because it will run out of space to insert new vertices.

Table 4.1 presents an exhaustive enumeration of all possible parent-child combinations. First, we prove by contradiction that the combinations marked as “n/a” cannot arise. These combinations have the following three common properties:

1. the child point p_i is either of Type-B or of Type-C,
2. the parent \hat{p}_i lies on the segment adjacent to the segment containing p_i ,
3. at least one of the points p_i and \hat{p}_i is of Type-B.

Since a point of Type-B or of Type-C can only be inserted as a result of encroachment by its parent, the angle between the segments containing p_i and \hat{p}_i has to be less than 90° . Therefore, both p_i and \hat{p}_i must be of Type-C or input which contradicts item 3 above.

All the remaining parent-child combinations (marked with numbers) have been analyzed in Theorem 4.4. From this theorem it follows that with the use of the GDR algorithm no new edge will be created whose length is less than $C \cdot \text{lfs}_{\min}$, where $C > 0$ is a constant, and $\text{lfs}_{\min} = \min_{p \in \Omega} \text{lfs}(p)$, see Figure 5.1. \square

6. Proof of Good Grading. The main result of this section is Theorem 6.4 which establishes that the insertion radius of a vertex has a lower bound proportional to its local feature size. This is a stronger result than the bound on the insertion radius required for the termination of the algorithm that is proportional to the global minimum of the local feature size. First, we prove Lemmas 6.1 and 6.2 that bound from above the distance from a point to its parent in terms of the “size” of the corresponding two-dimensional and one-dimensional selection region, respectively. These results are used to prove Lemma 6.3 which establishes that the vertex density in a point is bounded from above by a linear function of the density in its parent. Lemma 6.3 is proved only for cases (1)–(4), since for cases (5) and (6) the relation of the insertion radius to the local feature size proved by Theorem 6.4 follows directly from the Spacing Theorem. Finally, we prove Theorem 6.4 by enumerating all possible type combinations of a point and its parent.

LEMMA 6.1. *If p is of Type-A then*

$$\|p - \hat{p}\| \leq (2 - \delta_2)r, \quad (6.1)$$

where r is the circumradius of the skinny triangle t .

Proof. If c is the circumcenter of t , then

$$\begin{aligned} \|p - \hat{p}\| &\leq \|p - c\| + \|c - \hat{p}\| && \text{(from the triangle inequality)} \\ &\leq (1 - \delta_2)r + \|c - \hat{p}\| && \text{(since } p \text{ is in the selection circle)} \\ &= (1 - \delta_2)r + r && \text{(since } \hat{p} \text{ is a vertex of } t\text{)} \\ &= (2 - \delta_2)r. \end{aligned}$$

\square

LEMMA 6.2. *If p is of Type-B then*

$$\|p - \hat{p}\| \leq (2 - \delta_1) \frac{|s|}{2}, \quad (6.2)$$

where $|s|$ is the length of the encroached segment s .

Proof. If c is the center of s , then

$$\begin{aligned}
\|p - \hat{p}\| &\leq \|p - c\| + \|c - \hat{p}\| && \text{(from the triangle inequality)} \\
&\leq (1 - \delta_1) \frac{|s|}{2} + \|c - \hat{p}\| && \text{(since } p \text{ is in the selection interval)} \\
&\leq (1 - \delta_1) \frac{|s|}{2} + \frac{|s|}{2} && \text{(since } \hat{p} \text{ encroaches upon } s) \\
&= (2 - \delta_1) \frac{|s|}{2}.
\end{aligned}$$

□

LEMMA 6.3. *If p is a vertex of the mesh inserted by the GDR algorithm, and C_n ($n = 1, 2, 3, 4$) are the constants specified by Theorem 4.4 for the corresponding cases listed in Table 4.1, then the following inequality holds:*

$$D(p) \leq B_n + \frac{D(\hat{p})}{C_n}, \quad n = 1, 2, 3, 4 \quad (6.3)$$

where $B_1 = \frac{2-\delta_2}{\delta_2}$, $B_2 = \frac{2-\delta_1}{\delta_1}$, $B_3 = 1$, $B_4 = 1$.

Proof.

First, we prove the inequality

$$\|p - \hat{p}\| \leq B_n \cdot R(p) \quad (6.4)$$

for each of the cases below.

Case (1):

$$\begin{aligned}
\|p - \hat{p}\| &\leq (2 - \delta_2)r && \text{(from Lemma 6.1)} \\
&= \frac{2-\delta_2}{\delta_2} \delta_2 r \\
&\leq \frac{2-\delta_2}{\delta_2} R(p) && \text{(from Lemma 4.1);}
\end{aligned}$$

therefore, inequality (6.4) can be satisfied with $B_1 = \frac{2-\delta_2}{\delta_2}$.

Case (2):

$$\begin{aligned}
\|p - \hat{p}\| &\leq (2 - \delta_1) \frac{|s|}{2} && \text{(from Lemma 6.2)} \\
&= \frac{2-\delta_1}{\delta_1} \delta_1 \frac{|s|}{2} \\
&\leq \frac{2-\delta_1}{\delta_1} R(p) && \text{(from Lemma 4.2, since } \hat{p} \text{ is rejected);}
\end{aligned}$$

therefore, inequality (6.4) can be satisfied with $B_2 = \frac{2-\delta_1}{\delta_1}$.

Case (3): The argument is a special case of Case (2) with $\delta_1 = 1$; therefore, inequality (6.4) can be satisfied with $B_3 = 1$.

Case (4): From Lemma 4.3, $\|p - \hat{p}\| = R(p)$; therefore, inequality (6.4) can be satisfied with $B_4 = 1$.

Now, for all cases (1)–(4),

$$\begin{aligned}
\text{lfs}(p) &\leq \text{lfs}(\hat{p}) + \|p - \hat{p}\| && \text{(from Lemma 2.2)} \\
&\leq \text{lfs}(\hat{p}) + B_n R(p) && \text{(from (6.4))} \\
&= D(\hat{p}) R(\hat{p}) + B_n R(p) && \text{(from (2.2))} \\
&\leq D(\hat{p}) \frac{R(p)}{C_n} + B_n R(p) && \text{(from Theorem 4.4).}
\end{aligned}$$

The result follows from the division of both sides by $R(p)$. □

THEOREM 6.4 (Extension of Lemma 7 in [19] and Lemma 2 in [18]). *Suppose that $\bar{\rho} > \sqrt{2}$ and the smallest angle in the input PSLG is strictly greater than 60° .*

There exist fixed constants $D_A > 0$, $D_B > 0$, and $D_C > 0$ such that, for any vertex p inserted (or considered for insertion and rejected) by the GDR algorithm, the following inequalities hold:

$$D(p) \leq \begin{cases} D_A & \text{if } p \text{ is of Type-A,} \\ D_B & \text{if } p \text{ is of Type-B,} \\ D_C & \text{if } p \text{ is of Type-C.} \end{cases} \quad (6.5)$$

Therefore, the insertion radius of p has a lower bound proportional to its local feature size.

Proof. The proof is by induction and is similar to the proof of Lemma 7 in [19]. The base case covers the input vertices, and the inductive step covers the other three types of vertices.

Base case: The theorem is true if p is an input vertex, because in this case, by Remark 2, $D(p) = \text{lfs}(p)/R(p) \leq 1$.

Inductive hypothesis: Assume that the theorem is true for \hat{p} , i.e.,

$$D(\hat{p}) \leq \begin{cases} D_A & \text{if } \hat{p} \text{ is of Type-A,} \\ D_B & \text{if } \hat{p} \text{ is of Type-B,} \\ D_C & \text{if } \hat{p} \text{ is of Type-C.} \end{cases} \quad (6.6)$$

Inductive step: For each of the cases (n), $n = 1, 2, 3, 4$, we start with (6.3) and apply the inductive hypothesis considering the possible type combinations of p and \hat{p} from Table 4.1. As a result, the inequalities in (6.5) can be satisfied if D_A , D_B , and D_C are chosen such that the following inequalities (6.7), (6.8), (6.9), (6.10), (6.11), and (6.12) hold:

Case (1):

$$B_1 + \frac{D_A}{C_1} \leq D_A, \quad (6.7)$$

$$B_1 + \frac{D_B}{C_1} \leq D_A, \quad (6.8)$$

$$B_1 + \frac{D_C}{C_1} \leq D_A, \quad (6.9)$$

Case (2):

$$B_2 + \frac{D_A}{C_2} \leq D_B, \quad (6.10)$$

Case (3):

$$B_3 + \frac{D_A}{C_3} \leq D_C, \quad (6.11)$$

Case (4):

$$B_4 + \frac{D_C}{C_4} \leq D_C. \quad (6.12)$$

For cases (5) and (6), from Theorem 4.4 we have: $D(p) = \text{lfs}(p)/R(p) \leq 1/C_n$, i.e., the inequalities in (6.5) can be satisfied if D_B , and D_C are chosen such that the following inequalities (6.13) and (6.14) hold:

Case (5):

$$\frac{1}{C_5} \leq D_B, \quad (6.13)$$

Case (6):

$$\frac{1}{C_6} \leq D_C. \quad (6.14)$$

From (6.7),

$$D_A \geq \frac{B_1 C_1}{C_1 - 1}. \quad (6.15)$$

We solve (6.8) together with (6.10) and obtain

$$D_A \geq \frac{(B_1 C_1 + B_2) C_2}{C_1 C_2 - 1}. \quad (6.16)$$

We solve (6.9) together with (6.11) and obtain

$$D_A \geq \frac{(B_1 C_1 + B_3) C_3}{C_1 C_3 - 1}. \quad (6.17)$$

From (6.12),

$$D_C \geq \frac{B_4 C_4}{C_4 - 1}. \quad (6.18)$$

Finally, from (6.15), (6.16), (6.17), (6.18), (6.10), (6.11), (6.13), and (6.14) we obtain the solution:

$$\begin{aligned} D_A &\geq \max \left\{ \frac{B_1 C_1}{C_1 - 1}, \frac{(B_1 C_1 + B_2) C_2}{C_1 C_2 - 1}, \frac{(B_1 C_1 + B_3) C_3}{C_1 C_3 - 1} \right\}, \\ D_B &\geq \max \left\{ \frac{D_A}{C_2} + B_2, \frac{1}{C_5} \right\}, \\ D_C &\geq \max \left\{ \frac{B_4 C_4}{C_4 - 1}, \frac{D_A}{C_3} + B_3, \frac{1}{C_6} \right\}. \end{aligned}$$

If we plug in the values for B_n and C_n we have:

$$\begin{aligned} D_A &\geq \max \left\{ \frac{(2-\delta_2)\bar{\rho}}{\bar{\rho}\delta_2-1}, \frac{(2-\delta_2)\bar{\rho}\delta_1+2-\delta_1}{\bar{\rho}\delta_1\delta_2-\sqrt{2}}, \frac{(2-\delta_2)\bar{\rho}+1}{\bar{\rho}\delta_2-\sqrt{2}} \right\}, \\ D_B &\geq \frac{\sqrt{2}D_A+2-\delta_1}{\delta_1}, \\ D_C &\geq \max \left\{ \frac{1}{1-2\cos\alpha_{\min}}, \sqrt{2}D_A + 1 \right\}. \end{aligned}$$

□

REMARK 7. D_A , D_B , and D_C are undefined if any of the following conditions hold:

$$\bar{\rho} = \sqrt{2}, \quad \delta_2 = \frac{\sqrt{2}}{\bar{\rho}}, \quad \delta_1 = \frac{\sqrt{2}}{\bar{\rho}\delta_2}, \quad \alpha_{\min} = 60^\circ. \quad (6.19)$$

These values of δ_2 and δ_1 correspond to the boundaries of the selection circles and intervals, respectively.

us the same bound

7. An Example of a Point Placement Approach. As an example of a practical application of selection regions we developed an approach which in many cases allows us to decrease the number of inserted Steiner points. We emphasize that this is just a starting point in the development of mesh optimization techniques based on the use of selection regions. The motivating idea is shown in Figure 8.1(left): with one point insertion we try to eliminate as many skinny triangles as possible. In addition, we try to preserve as many good quality triangles as we can. For more details on this approach see our earlier description in [3]. Table 7.1 shows the number of triangles obtained for the simplified brain outline (see Figure 7.1) for three point placement methods: circumcenter, off-center and our suggested technique. For all angle bounds we can see a slight improvement in the number of inserted points, and, as a result, in the final number of triangles.

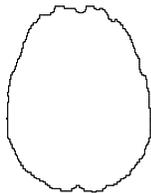


FIG. 7.1. A simplified outline of a human brain.

TABLE 7.1

The comparison of the number of triangles generated with the use of circumcenter, off-center, and an optimization-based point insertion strategies.

Point position	Minimum angle bound $\theta = \arcsin \frac{1}{2\bar{\rho}}$					
	5°	10°	15°	20°	25°	30°
Circumcenter	714	874	1018	1344	1754	3186
Off-center	702	842	1002	1300	1584	2412
Our example of an optimization-based method	700	828	964	1274	1564	2370

8. Conclusions. We developed a fully generalized two-dimensional sequential Delaunay refinement algorithm. It makes it possible to develop custom point insertion methods for a variety of mesh optimization goals, for example,

- for minimizing the number of inserted points, see [3] and Figure 1.2 here;
- for splitting multiple poor quality triangles simultaneously, see Fig. 8.1(left);
- for eliminating slivers, see [8, 12, 13];

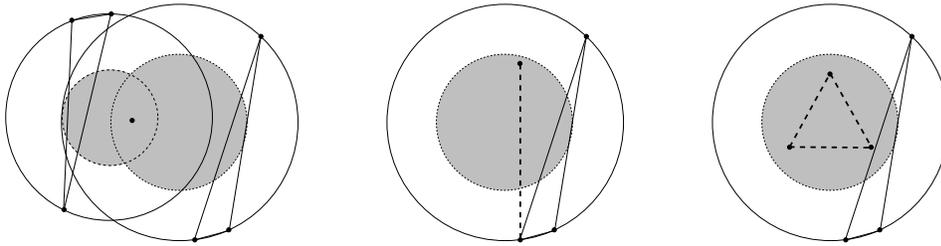


FIG. 8.1. *Examples of the approaches for choosing Steiner points within selection disks of skinny triangles.*

- for creating elongated edges in required directions, see Fig. 8.1(center);
- for inserting more than one point, e.g., to create elements with specific shapes, see Fig. 8.1(right);
- for satisfying other application-specific requirements, for example, conformity to a boundary zone, see Fig. 1.1.

The presented algorithm is based on the two main simplifying decisions for the definition of the selection regions: (1) the selection regions are balls (circles or intervals) centered in the geometric centers of the corresponding circumcircles or segments, and (2) the size of a selection region is a function of the local circumradius of the triangle or segment and of the global quality bound (disregarding the constant parameters). There are indications that locally larger (more complex) selection regions can be defined; therefore, we do not claim the maximality in the proposed approach. For example, the off-center point in the Üngör's approach [20] is determined based on the local quality of the triangle, and it may or may not be inside the selection region as we defined it here. Our ongoing work also includes the improvement of the quality bound and the investigation of the relation between the input angles and the sizes of the selection intervals.

Currently we are also working on combining the sequential Generalized Delaunay Refinement algorithm with the parallel Delaunay meshing analysis [3, 4, 14, 15]. The result will make it possible to automatically parallelize any custom point insertion method with the only restriction that it inserts points within selection regions. Our future work includes the extension of the fully generalized algorithm to higher dimensions.

9. Acknowledgments. We thank Gary Miller for helpful references and Andriy Fedorov for insightful discussions on medical imaging. We also thank Panagiotis Foteinos for a deeply technical exploration of generalized Delaunay refinement. This work was supported (in part) by the NSF grants CCS-0750901 and CCF-0833081, and by the John Simon Guggenheim Foundation. We thank the anonymous referees for their helpful comments.

REFERENCES

- [1] MARSHALL WAYNE BERN AND DAVID EPPSTEIN, *Mesh generation and optimal triangulation*, in *Computing in Euclidean Geometry*, Ding-Zhu Du and Frank Hwang, eds., Lecture Notes Series on Computing, World Scientific, 1992, pp. 23–90.
- [2] ADRIAN BOWYER, *Computing Dirichlet tessellations*, *Computer Journal*, 24 (1981), pp. 162–166.

- [3] ANDREY N. CHERNIKOV AND NIKOS P. CHRISOCHOIDES, *Generalized Delaunay mesh refinement: From scalar to parallel*, in Proceedings of the 15th International Meshing Roundtable, Birmingham, AL, Sept. 2006, Springer, pp. 563–580.
- [4] ———, *Parallel guaranteed quality Delaunay uniform mesh refinement*, SIAM Journal on Scientific Computing, 28 (2006), pp. 1907–1926.
- [5] ———, *Three-dimensional semi-generalized point placement method for Delaunay mesh refinement*, in Proceedings of the 16th International Meshing Roundtable, Seattle, WA, Oct. 2007, Springer, pp. 25–44.
- [6] L. PAUL CHEW, *Guaranteed-quality triangular meshes*, Tech. Report TR89983, Cornell University, Computer Science Department, 1989.
- [7] ———, *Guaranteed quality mesh generation for curved surfaces*, in Proceedings of the 9th ACM Symposium on Computational Geometry, San Diego, CA, 1993, pp. 274–280.
- [8] ———, *Guaranteed-quality Delaunay meshing in 3D*, in Proceedings of the 13th ACM Symposium on Computational Geometry, Nice, France, 1997, pp. 391–393.
- [9] BORIS N. DELAUNAY, *Sur la sphere vide*, Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematika i Estestvennyka Nauk, 7 (1934), pp. 793–800.
- [10] WILLIAM H. FREY, *Selective refinement: A new strategy for automatic node placement in graded triangular meshes*, International Journal for Numerical Methods in Engineering, 24 (1987), pp. 2183–2200.
- [11] PAUL-LOUIS GEORGE AND HOUMAN BOROUCHEKI, *Delaunay Triangulation and Meshing. Application to Finite Elements*, HERMES, 1998.
- [12] XIANG-YANG LI, *Generating well-shaped d-dimensional Delaunay meshes*, Theoretical Computer Science, 296 (2003), pp. 145–165.
- [13] XIANG-YANG LI AND SHANG-HUA TENG, *Generating well-shaped Delaunay meshes in 3D*, in Proceedings of the 12th annual ACM-SIAM symposium on Discrete algorithms, Washington, D.C., 2001, pp. 28–37.
- [14] LEONIDAS LINARDAKIS AND NIKOS CHRISOCHOIDES, *Delaunay decoupling method for parallel guaranteed quality planar mesh refinement*, SIAM Journal on Scientific Computing, 27 (2006), pp. 1394–1423.
- [15] ———, *Graded Delaunay decoupling method for parallel guaranteed quality planar mesh generation*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1875–1891.
- [16] GARY L. MILLER, *A time efficient Delaunay refinement algorithm*, in Proceedings of the 15th annual ACM-SIAM symposium on Discrete algorithms, New Orleans, LA, 2004, pp. 400–409.
- [17] GARY L. MILLER, DAFNA TALMOR, SHANG-HUA TENG, AND NOEL WALKINGTON, *A Delaunay based numerical method for three dimensions: Generation, formulation, and partition*, in Proceedings of the 27th Annual ACM Symposium on Theory of Computing, Las Vegas, NV, May 1995, pp. 683–692.
- [18] JIM RUPPERT, *A Delaunay refinement algorithm for quality 2-dimensional mesh generation*, Journal of Algorithms, 18(3) (1995), pp. 548–585.
- [19] JONATHAN RICHARD SHEWCHUK, *Delaunay refinement algorithms for triangular mesh generation*, Computational Geometry: Theory and Applications, 22 (2002), pp. 21–74.
- [20] ALPER ÜNGÖR, *Off-centers: A new type of Steiner points for computing size-optimal guaranteed-quality Delaunay triangulations*, in Proceedings of LATIN, Buenos Aires, Argentina, Apr. 2004, pp. 152–161.
- [21] DAVID F. WATSON, *Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes*, Computer Journal, 24 (1981), pp. 167–172.