FULLY GENERALIZED 2D CONSTRAINED DELAUNAY MESH REFINEMENT: REVISED

PANAGIOTIS A. FOTEINOS*, ANDREY N. CHERNIKOV†, AND NIKOS P. CHRISOCHOIDES‡

Abstract. Traditional refinement algorithms insert a Steiner point from a few possible choices at each step. Our algorithm, on the contrary, defines regions from where a Steiner point can be selected and thus inserts a Steiner point among an infinite number of choices. Our algorithm significantly extends existing generalized algorithms by increasing the number and the size of these regions. The lower bound for newly created angles can be as high as 26.56° with grading guarantees. This corrects a rather optimistic bound we had in a previous publication [12].

Key words. Delaunay triangulation, mesh generation

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

1. Introduction. Delaunay refinement algorithms can be categorized into two families of algorithms: truly Delaunay and constrained Delaunay refinement algorithms. The former produce meshes which are truly Delaunay by repeatedly splitting the constrained segments until they appear in the mesh. The latter produce meshes which are as Delaunay as possible, i.e., they preserve most (but not all) of the nice properties of truly Delaunay triangulations [9, 22]. The advantage of constrained Delaunay refinement algorithms is that they produce meshes with fewer elements and guarantee better bounds on the minimum angles and grading. For these reasons, we chose to develop a constrained Delaunay refinement algorithm.

Traditional Delaunay refinement algorithms [8, 9, 14, 17, 21, 23] improve the quality of the mesh by inserting additional points into the mesh: the so-called *Steiner* points. Specifically, they insert the circumcenter of a bad triangle and the midpoint of an encroached segment.

There is no universal rule, however, for where the Steiner points should be inserted. In the literature, there are methods that insert points other than circumcenters and midpoints. For example, in [25], a bad triangle is split with its offcenter instead of its circumcenter producing a smaller mesh in practice. Similarly, in [11], a bad triangle is split by inserting a point chosen among a total number of four candidate points. Our goal is to develop an algorithm that allows for customizable point insertion strategies. This flexibility could also help to remove slivers in three dimensions deterministically, as opposed to randomized algorithms [10, 16]. In this paper, we show that there is an infinite number of Steiner points that can be chosen to split a triangle or a segment. The points that can be selected as Steiner points form *selection regions*. When a bad triangle is considered for splitting, the selection region is a two-dimensional region called *selection disk*. (Sometimes, we refer to selection disks as *selection circles*). Similarly, when a segment is considered for splitting, the selection region is a one-dimensional region called *selection interval*.

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

†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).

Our group has already devised three generalized Delaunay refinement algorithms [5, 7, 12]. The algorithm in [5] (called *Semi-Generalized Refinement* algorithm) deploys only selection disks, i.e., selection intervals are not used. Rather, each encroached segment is traditionally split at its midpoint. The algorithm in [7] (called *Generalized Refinement* algorithm) extends the Semi-Generalized Refinement algorithm by introducing the selection intervals. When the encroached segment to be split, however, forms an acute input angle, the selection interval is not defined; the Generalized Refinement algorithm traditionally splits such segments at their midpoint. In this paper, we extend the Generalized Refinement algorithm by introducing the selection interval of segments forming acute input angles.

The generalized Delaunay refinement method we introduced in [12] already offers significant extensions to the selection regions. However, it contains an imprecision: the achieved bound is not 30° as we claimed. In this paper, we correct that mistake and we show that the algorithm produces graded meshes with a lower bound of 26.56°.

Some of the applications of selection regions are discussed in our previous work. Specifically, our work in [4, 7] describes how the selection circles can be utilized in order to decrease the size of the mesh. In addition, our work in [6] shows how selection regions can incorporate many different point placement strategies performed in parallel.

As proved theoretically in [24] and shown practically in [13], mesh quality strongly affects the convergence speed and the solution accuracy of the finite element solver: the larger the minimum angle of the mesh is, the lower the condition number of the linear system becomes which yields a faster and a more robust solution.

Both the Semi-Generalized [5] and the Generalized Refinement algorithm [7] are proved to terminate with a lower angle bound as high as 20.7°. In this paper, we improve the quality of the mesh proving termination with a lower angle bound as high as 26.56°.

Hudson [15] developed a theoretical framework for both two and three dimensions which shows that one can split elements at points other than their circumcenters. He, however, derives a rather weak angle bound in two dimensions: his algorithm guarantees that the lower bound for the angles is 20.7° (we guarantee that the lower bound for the angles is 26.56°). Furthermore, his framework suffers from a severe restriction: the input cannot form acute angles. As it will become obvious in Section 3, the input of our algorithm can have acute angles as small as 60°.

To our knowledge, among the truly Delaunay refinement algorithms guaranteeing good grading, the algorithm presented by Miller, Pav, and Walkington [18] comes with the highest lower angle bound (26.45°). As far as constrained Delaunay refinement techniques are concerned, Chew [9] describes a constrained Delaunay refinement algorithm with a 30° lower angle bound, but with no proof of good grading. Shewchuk [23] shows that Chew's algorithm [9] offers good grading for a worse angle bound (i.e., 26.56° instead of 30°) and with input angles more than 60°. Rand [20] shows that Chew's algorithm generates well-graded meshes with an angle bound as high as 28.60°, but without emphasis on small (smaller than 90°) input angles. In this paper, we show that Chew's algorithm produces well-graded elements with a 26.56° angle bound and with input angles larger than 60°.

In summary, the contributions of this paper are the following:

- It extends the flexibility offered by the existing generalized algorithms [5, 7, 15]: our algorithm presented here increases the number and size of regions from where Steiner points can be chosen, and

- It corrects an imprecision of our previous publication [12].

The rest of the paper is organized as follows. In Section 2, we shortly describe the traditional Delaunay refinement algorithms. In Section 3, we define the selection regions and give the pseudocode of our algorithm. In Section 4, we prove important lemmas and theorems needed for the proof of termination and good grading. The proof of good grading is presented in Section 5 and the proof of termination in Section 6. Finally, Section 7 experimentally evaluates our algorithm, and Section 8 concludes the paper.

2. Delaunay Refinement Background. The input domain Ω to be meshed is usually described as a Planar Straight Line Graph (PSLG) [21, 23]. A PSLG $\mathcal{X}$ is a set of input vertices and segments. The input segments are *constrained*, i.e., they have to appear in the final mesh, possibly as a union of smaller subsegments. We shall refer to an input segment as simply a *segment*. The input vertices should also be preserved in the final mesh. For brevity, we call both the segments and the input vertices *features*.

Let p_i, p_j be two vertices in the mesh. We denote the mesh edge that connects p_i with p_j as $e(p_i, p_j)$ regardless of whether it is an input segment or not. The Euclidean distance between these points is denoted as $\|p_i - p_j\|$.

We use the circumradius-to-shortest-edge ratio ρ of a triangle to measure its quality. If the circumradius-to-shortest-edge ratio of a triangle t is equal to or larger than a specified upper bound $\bar{\rho}$, then t is said to be a poor or skinny triangle. Mesh refinement algorithms split poor triangles, until the circumradius-to-shortest-edge ratio of all the triangles in the mesh is less than $\bar{\rho}$. This upper bound sets a lower bound for the angles in the mesh, since the circumradius-to-shortest-edge ratio of a triangle with shortest edge l, circumradius r, and smallest angle A is $\rho = \frac{r}{l} = \frac{1}{2 \sin A}$ [19, 23]. Therefore, when the refinement terminates, it is guaranteed that all the angles in the mesh are larger than $\arcsin \frac{1}{2\bar{\rho}}$. For brevity, we denote this angle lower bound as θ: $\theta = \arcsin \frac{1}{2\bar{\rho}}$. Clearly, θ can only be an acute angle.

A triangle t is said to satisfy the *constrained Delaunay property*, if there is no vertex that lies strictly inside t's circumscribed circle (*circumcircle*) and is visible from the interior of t [9, 23]. Two vertices p_i, p_j are visible to each other, if the line connecting p_i with p_j does not intersect (at exactly one point) the interior of any constrained segment. See Figure 2.1(left) and Figure 2.1(middle). At any time during the refinement process, all the triangles in the mesh have to satisfy the constrained Delaunay property.

Cavity [14] of a point p is defined to be the set $\mathcal{C}(p)$ of triangles t_i in the mesh, such that the circumcircle of every t_i in $\mathcal{C}(p)$ includes the point p, and p is visible from the interior of t_i (see Figure 2.1(right)). We denote $\partial\mathcal{C}(p)$ to be the set of boundary edges of the cavity, i.e., the edges which are incident upon only one triangle in $\mathcal{C}(p)$. For our analysis, we use the Bowyer-Watson (B-W) point insertion algorithm [3, 26], which can be written shortly as in Algorithm 1. Note that this definition of a cavity implies that if p is inserted into the mesh, then all the triangles in $\mathcal{C}(p)$ do not satisfy the constrained Delaunay property and must be deleted. After the deletion of these triangles, the cavity has to be re-triangulated, such that all the newly formed triangles respect the constrained Delaunay property.

As mentioned above, refinement algorithms repeatedly split skinny triangles, until the ratio of all the triangles in the mesh is less than the upper bound $\bar{\rho}$. Traditionally, a skinny triangle t is deleted by inserting its circumcenter p (using the Bower-Watson point insertion algorithm). We will show, however, that there is a

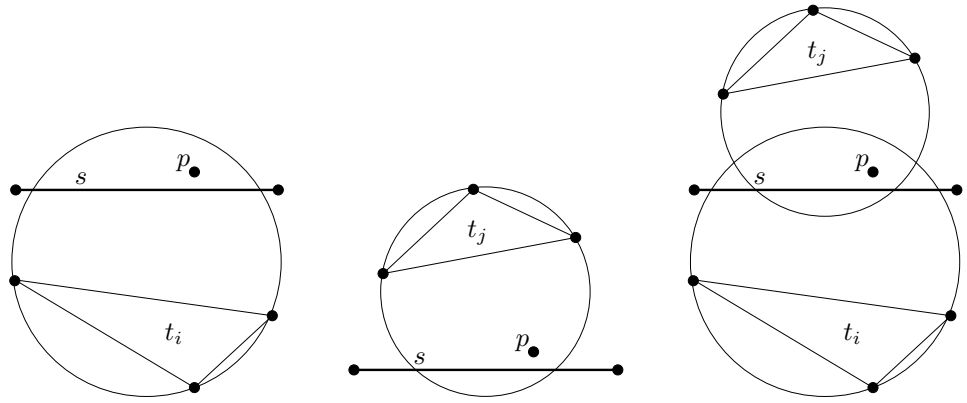

FIG. 2.1. ***(Left)*** *Triangle t_i satisfies the constrained Delaunay property: although p is inside the circumcircle, it is not visible from the interior of t_i because of the constrained segment s.* ***(Middle)*** *Triangle t_j does not respect the constrained Delaunay property: p lies inside the circumcircle and is visible from the interior of t_j.* ***(Right)*** *The cavity $\mathcal{C}(p)$ of p includes only the triangle t_j. Although p lies inside t_i's circumcircle, t_i does not belong to $\mathcal{C}(p)$, since p is not visible from the interior of t_i. Notice that t_i respects the constrained Delaunay property, but t_j does not; therefore, t_j must be deleted.*

Algorithm 1: The Bowyer-Watson point insertion procedure.

1 **Algorithm:** `BowyerWatson(`V, T, p`)`

Input : V is the set of vertices.
T is the set of triangles.
p is the Steiner point to be inserted.

Output: V and T after the insertion of p.

2 $V \leftarrow V \cup \{p\}$;

3 $T \leftarrow T \setminus \mathcal{C}(p) \cup \{(p\xi) \mid \xi \in \partial\mathcal{C}(p)\}$;

whole two-dimensional space inside the circumcircle of t where a Steiner point p can be chosen from.

Even though the Steiner point p of a skinny triangle t_i is always inserted inside t_i's circumcircle, t_i may not belong to $\mathcal{C}(p)$. This can happen when p is not visible from the interior of t_i; that is, when t_i and p lie on opposite sides of a constrained segment. See Figure 2.1(right) for an illustration: in such a case, the insertion of p fails to remove the skinny triangle t_i from the mesh. To deal with these circumstances and to prevent the insertion of Steiner points outside the domain, Delaunay refinement algorithms obey special *encroachment* rules [9, 23]. We define encroachment as follows:

DEFINITION 2.1. *(Encroachment of a constrained segment) A segment s is said to be encroached upon by a skinny triangle t, if p_i is not visible from the interior of t due to s, i.e., the line connecting p_i and any point in the interior of t intersects the interior of s. If more than one segment lies between t and p_i, encroached is the segment closest to t.*

See Figure 2.2 for an illustration.

If a skinny triangle t encroaches upon a segment s, then its Steiner point is rejected from the mesh. In addition, all the *free* vertices (i.e., vertices that neither are input vertices nor lie on segments) which lie inside the diametral circle of s and are visible

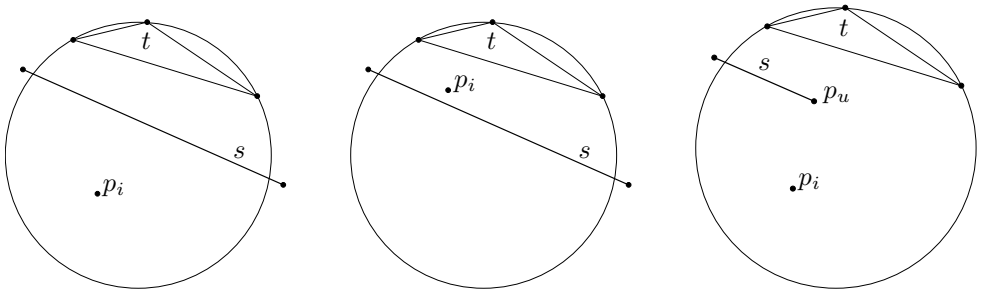

FIG. 2.2. *A skinny triangle t and its Steiner point p_i when* ***(Left)*** *t encroaches upon segment s, and when* ***(Middle)*** *it does not.* ***(Right)*** *Does t encroach upon s or not? No matter where p_i is, this case illustrated in this subfigure cannot happen, since t did not satisfy the constrained Delaunay property before the insertion of p_i. Indeed, p_u lies inside t's circumcircle and is visible from the interior of t, because the line connecting p_u with any point in the interior of t does not intersect the interior of s but one of its endpoints.*

from the interior of s are deleted. (The diametral circle of a segment is the circle with diameter that segment.) Then, a new Steiner point is inserted on s. Traditionally, the midpoint of s is inserted, but we will show that there is a whole one-dimensional space inside s where a Steiner point can be chosen from.

The following definitions of the *local feature size*, *insertion radius*, and *parent* play a central role in the analysis in [21, 23], and we use them for our analysis in the generalized form, too.

DEFINITION 2.2 (Local feature size [21, 23]). *The* local feature size function $\mathrm{lfs}(p)$ *for a given point p is equal to the radius of the smallest disk centered at p that intersects two non-incident features of the PSLG.*

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

LEMMA 2.3 (Lemma 1 in [21], Lemma 2 in [23]). *Given any PSLG and any two points p_i and p_j, the following inequality holds:*

$$\mathrm{lfs}(p_i) \leq \mathrm{lfs}(p_j) + \|p_i - p_j\|. \tag{2.1}$$

DEFINITION 2.4 (Insertion radius). *The* insertion radius $R(p)$ *of point p is the distance from p to its nearest visible vertex, immediately after p 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.*

REMARK 1. *Assume that p_l and p_m are mutually visible vertices inserted into the mesh, and that 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 p_m was the closest visible vertex from 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 [23], if p is an input vertex, then $R(p) \geq \mathrm{lfs}(p)$. Indeed, the (closed) disk with center p and radius $R(p)$ intersects two non-incident features: the input vertex p and p's closest visible vertex.*

Below, we define the parent of a Steiner point:

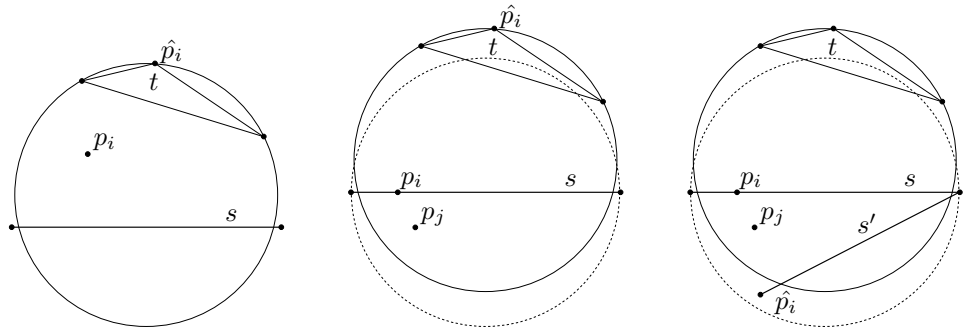

FIG. 2.3. *Different cases of parenthood as defined in Definition 2.5. t is a skinny triangle, p_i, p_j are Steiner points, and s, s' are (constrained) segments. Point p_i is inserted after p_j.* ***(Left)*** *t does not encroach upon s, and therefore its Steiner point p_i is not rejected. The parent $\hat{p}_i$ of p_i is the most recently inserted vertex of t's shortest edge.* ***(Middle)*** *t encroaches upon s and its Steiner point p_j is rejected. Instead, the Steiner point p_i is inserted on s. Since the diametral circle of s is empty of non-free vertices, the parent of p_i is the most recently inserted vertex of t's shortest edge.* ***(Right)*** *t encroaches upon s, and therefore its Steiner point p_j is rejected. Instead, the Steiner point p_i is inserted on s. Since the diametral circle of s contains a non-free vertex, lying on the constrained segment s', the parent $\hat{p}_i$ of p_i is this non-free vertex.*

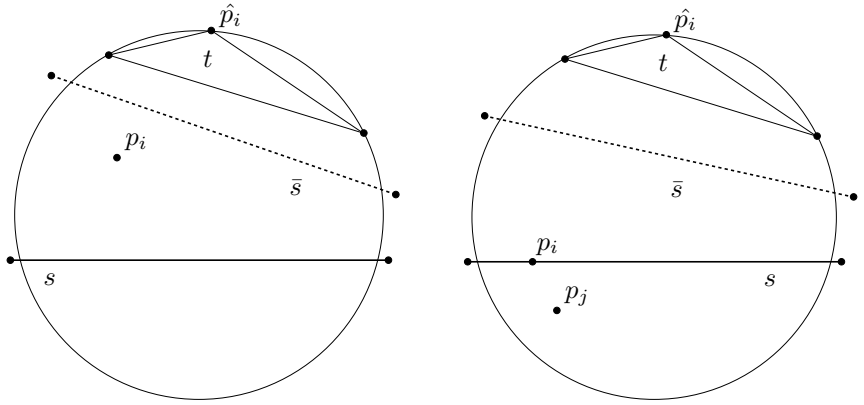

FIG. 2.4. *The constrained segment $\bar{s}$ is shown with the dashed line.* ***(Left)*** *Rule-2 applies (i.e., p_i is not rejected), as in Figure 2.3(left), but now $\bar{s}$ obstructs visibility between p_i and $\hat{p}_i$.* ***(Right)*** *Rule-3 applies (i.e., t encroaches upon s), as in Figure 2.3(middle), but now $\bar{s}$ obstructs visibility between p_i and $\hat{p}_i$.*

DEFINITION 2.5 (Parent of a Steiner point). *The* parent $\hat{p}_i$ *of point p_i is the vertex defined by the following four rules:*

Rule-1. *If p_i is either an input vertex or a rejected vertex, then it has no parent.*

Rule-2. *If p_i is inserted inside the circumcircle of a poor quality triangle t, $\hat{p}_i$ is the most recently inserted vertex of the shortest edge of t. See Figure 2.3(left).*

Rule-3. *If p_i is inserted on a segment s, encroached upon by a skinny triangle t, and no non-free vertex visible from p_i lies inside the diametral circle of s, then $\hat{p}_i$ is the most recently inserted vertex of the shortest edge of the encroaching triangle. See Figure 2.3(middle).*

Rule-4. *If p_i is inserted on a segment s, encroached upon by a skinny triangle t, and at least one non-free vertex visible from p_i lies inside the diametral circle of s, then $\hat{p}_i$ is the closest-to-p_i such a non-free vertex. See Figure 2.3(right).*

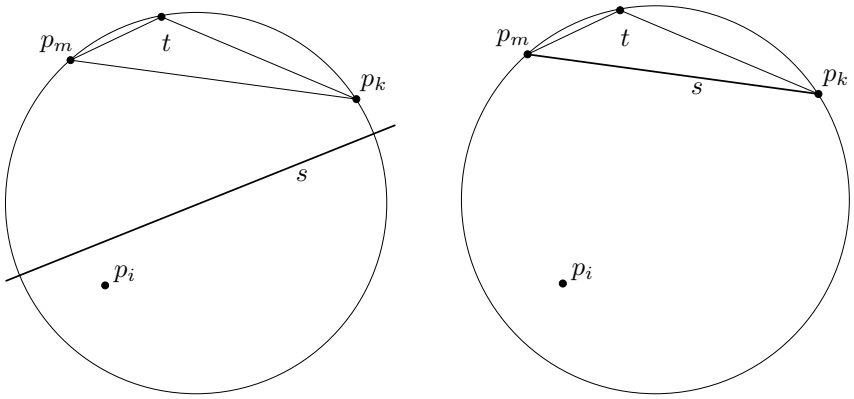

FIG. 2.5. *The separator of an encroached (constrained) segment s. A skinny triangle t encroaches upon the constrained segment s (because t and its Steiner point p_i lie on opposite sides of s). The edge $e\,(p_m p_k)$ is the separator of s. Intuitively, the separator of s is the edge of t which s would fall on, if we moved s towards t. The right figure illustrates the special case where the encroached segment coincides with its separator.*

LEMMA 2.6. *The parent $\hat{p}_i$ (if one exists) of a point p_i is visible from p_i.*

Proof. Let p_i be a vertex and $\hat{p}_i$ be its parent. For the sake of contradiction, assume that $\hat{p}_i$ is not visible from p_i due to a segment $\bar{s}$, i.e., the line connecting p_i and $\hat{p}_i$ intersects the interior of $\bar{s}$. We will investigate what that means for each rule of Definition 2.5.

Rule-1 does not apply, since we have assumed that p_i does have a parent in this lemma. This also implies that p_i is neither an input nor a rejected vertex.

If Rule-2 applies, then consider Figure 2.4(left): the skinny triangle t must encroach upon $\bar{s}$. But that means that p_i should have been rejected from the mesh: a contradiction.

If Rule-3 applies, then consider Figure 2.4(right): the skinny triangle t does not encroach upon s but upon $\bar{s}$: a contradiction.

Finally, if Rule-4 applies, $\hat{p}_i$ and p_i are visible to each other by definition: a contradiction. □

Notice that a rejected point does not have a parent according to our definition (Rule-1), while it does in the traditional approaches. This change is not necessary, but it considerably simplifies the proofs in the next sections.

Also, observe that the parent of a Steiner point p_i inserted on an encroached, constrained segment s might lie outside the diametral circle of s (see Figure 2.3(middle) for an illustration). This is different from the traditional approaches where the parent of p_i is always inside or on the diametral circle of the encroached segment. This modification of the parent definition is crucial, since the diametral circle of an encroached segment might not contain any traditional parent candidates at all in our generalized algorithm. For our analysis, it is useful to know whether the parent of p_i is strictly inside the diametral circle of s or not; we, therefore, classify the parent as either an *external* or a *non-external* parent according to the following definition:

DEFINITION 2.7 (External parent). *Assume that p_i is a Steiner point inserted on an encroached segment s. If its parent $\hat{p}_i$ lies on or outside the diametral circle of s, then we say that $\hat{p}_i$ is an external parent. Conversely, if $\hat{p}_i$ lies strictly inside the diametral circle of s, then we say that $\hat{p}_i$ is a non-external parent.*

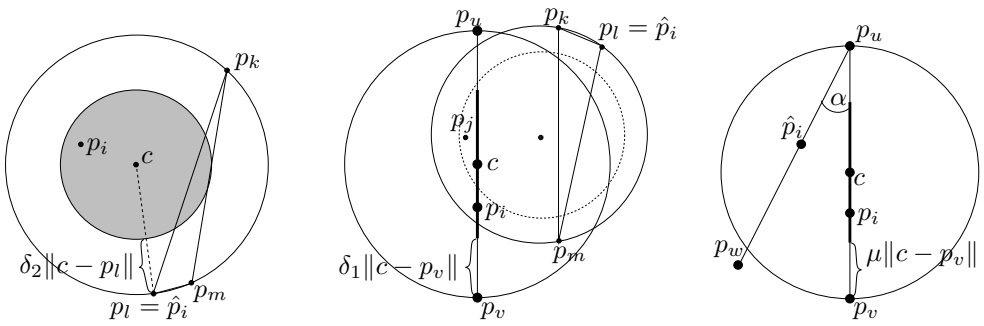

FIG. 3.1. *The three kinds of selection regions. Big dots denote non-free vertices lying on (constrained) segments, while small dots denote free vertices. Point c is not a part of the mesh; it is rather an auxiliary point that denotes the center of the selection regions. Point p_i is a Steiner point inserted into the mesh. Point p_j is a Steiner point rejected from the mesh.* ***(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.* ***(Middle)*** *Type-B selection interval (bold) for an encroached segment $e(p_u p_v)$. Also illustrates case (2) from Table 4.1. The constrained segment $e(p_u p_v)$ is encroached since the skinny triangle $\triangle p_k p_l p_m$ and its Steiner point p_j lie on opposite sides. The Steiner point p_j is rejected, and the diametral circle is emptied of vertices which are free and visible from the interior of $e(p_u p_v)$; in this example, only the free vertex p_m is deleted. Another Steiner point p_i in the Type-B selection interval of $e(p_u p_v)$ is inserted instead. The parent $\hat{p}_i$ of the vertex p_i may be an external vertex (as depicted), i.e., a vertex outside the diametral circle of $e(p_u p_v)$.* ***(Right)*** *Type-C selection interval (bold) for an encroached segment $e(p_u p_v)$. Also illustrates case (4) from Table 4.1. The Steiner point p_i has been inserted in the Type-C selection interval. The parent $\hat{p}_i$ is a non-external vertex which lies on a segment incident to $e(p_u p_v)$. The input angle formed by the segments $e(p_u p_v), e(p_u p_w)$ is denoted with α.*

For example, the point $\hat{p}_i$ is an external parent in Figure 2.3(middle), but it is a non-external parent in Figure 2.3(right).

REMARK 3. *If p is inserted on a segment s encroached upon by a skinny triangle t, and $\hat{p}$ is an external parent, then $\hat{p}$ is the most recently inserted vertex of t's shortest edge. Indeed, if Rule-3 did not apply, then Rule-4 would, i.e., $\hat{p}$ would lie inside the diametral circle of s: a contradiction.*

Next, we define the *separator* of an encroached segment:

DEFINITION 2.8 (Separator of an encroached, constrained segment). *Let t encroach upon a segment s. The separator of s is the unique edge of t that lies between s and the interior of t.*

Notice that the separator of an encroached segment always exists and is unique. See Figure 2.5 for a couple of examples.

The density [23] of a vertex p, denoted as $D(p)$, is defined as:

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

Refinement algorithms are proved to produce well-graded triangles by showing that the density of all mesh vertices is less than a constant.

3. Generalized Delaunay Refinement Algorithm. In the following sections, we will show that our Generalized Constrained Delaunay Refinement (GCDR) algorithm guarantees termination and good grading for a circumradius-to-shortest-edge ratio upper bound $\bar{\rho}$ arbitrarily close to $\frac{\sqrt{5}}{2}$. This value for $\bar{\rho}$ corresponds to an angle lower bound of $\theta = \arcsin\frac{1}{\sqrt{5}} = 26.565$ degrees.

DEFINITION 3.1 (Selection circle). *For a skinny triangle with circumcenter c, shortest edge length l, circumradius r, and circumradius-to-shortest-edge ratio $\rho = r/l \geq \bar{\rho} \geq \frac{\sqrt{5}}{2}$, 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*

$$\frac{\sqrt{5}}{2\bar{\rho}} \leq \delta_2 \leq 1. \tag{3.1}$$

See Figure 3.1(left) for an illustration.

REMARK 4. *If $\delta_2 = 1$, then the selection circle shrinks to the circumcenter point.*

DEFINITION 3.2 (Type-B selection interval). *If s is an encroached segment with center c, then the* Type-B 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*

$$\frac{\sqrt{5}}{2\bar{\rho}\delta_2} \leq \delta_1 \leq 1, \tag{3.2}$$

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

REMARK 5. *If $\delta_1 = 1$, then the Type-B selection interval shrinks to the center point.*

REMARK 6. *If $\bar{\rho} = \frac{\sqrt{5}}{2}$, then both δ_2 and δ_1 can only be equal to 1; therefore, both the selection circles of skinny triangles and the Type-B selection intervals of encroached segments shrink to the respective center points.*

REMARK 7. *If $\delta_2 = \frac{\sqrt{5}}{2\bar{\rho}}$, then δ_1 can only be equal to 1; therefore, the Type-B selection intervals of encroached segments shrink to the center points.*

REMARK 8. *If $\delta_1 = \frac{\sqrt{5}}{2\bar{\rho}}$, then δ_2 can only be equal to 1; therefore, the selection circles shrink to the circumcenters.*

DEFINITION 3.3 (Type-C selection interval). *If s is a segment with center c, then the* Type-C selection interval *of s is the subsegment of s with center c and length $|s|(1-\mu)$, where μ is a constant parameter chosen such that*

$$2\cos\alpha_{\min} \leq \mu \leq 1, \tag{3.3}$$

and $\alpha_{\min}$ is the minimum input angle in the PSLG. Clearly, $\alpha_{\min}$ cannot be smaller than $60°$. See Figure 3.1(right) for an illustration.

REMARK 9. *If $\mu = 1$, then the Type-C selection interval shrinks to the center point. In our previous Generalized algorithms [5, 7], the Type-C selection interval is always the center point.*

REMARK 10. *If $\alpha_{\min}$ is equal to $60°$, then μ can only be equal to 1, and therefore the Type-C selection interval of the corresponding segment shrinks to its center point.*

REMARK 11. *If $\delta_1 = \delta_2 = \mu = 1$, then our GCDR algorithm is identical to Chew's second algorithm, as described in [23].*

Algorithm 2 presents the GCDR algorithm. For brevity, let us classify the non-rejected Steiner points, inserted by the GCDR algorithm, into three categories:

Algorithm 2: The Generalized Constrained Delaunay Refinement algorithm.

Algorithm: `GeneralizedDelaunayRefinement(`$\mathcal{X}$, $\bar{\rho}$, δ_2, δ_1, μ, F_A`()`, F_B`()`, F_C`()`, $\mathcal{M}$`)`

Input : $\mathcal{X}$ is the input PSLG.
$\bar{\rho}$ is the upper bound on circumradius-to-shortest-edge ratio, $\bar{\rho} \geq \frac{\sqrt{5}}{2}$.
δ_2 is the parameter which defines selection circles for skinny triangles, $\frac{\sqrt{5}}{2\bar{\rho}} \leq \delta_2 \leq 1$.
δ_1 is the parameter which defines Type-B selection intervals of encroached segments, $\frac{\sqrt{5}}{2\bar{\rho}\delta_2} \leq \delta_1 \leq 1$.
μ is the parameter which defines Type-C selection intervals of encroached segments, $2\cos\alpha_{\min} \leq \mu \leq 1$, where $\alpha_{\min}$ is the minimum input angle present in $\mathcal{X}$.
F_A`()`, F_B`()`, and F_C`()` are user-defined functions which return specific Steiner points within selection circles and selection intervals, respectively.
$\mathcal{M} = (V, T)$ is an initial constrained Delaunay triangulation of $\mathcal{X}$, where V is the set of vertices and T is the set of triangles.

Output: A constrained Delaunay mesh $\mathcal{M}$ whose triangles have circumradius-to-shortest-edge ratio less than $\bar{\rho}$.

Let SkinnyTriangles be the set of triangles in T whose circumradius-to-shortest-edge ratio is larger than or equal to $\bar{\rho}$;
while SkinnyTriangles $\neq \emptyset$ **do**
    Pick $t \in$ SkinnyTriangles;
    $p \leftarrow F_A(\mathcal{M}, \delta_2, t)$; `/* `$p$` is of Type-A */`
    if *t encroaches upon a segment s* **then** `/* `p` is rejected */`
        `Delete` the free vertices inside the diametral circle of s;
        if *there is a non-free vertex which lies strictly inside the diametral circle of s and is visible from the interior of s* **then**
            $p \leftarrow F_C(\mathcal{M}, \mu, s)$; `/* `$p$` is of Type-C */`
        else
            $p \leftarrow F_B(\mathcal{M}, \delta_1, s)$; `/* `$p$` is of Type-B */`
        end
    end
    `BowyerWatson(`V, T, p`)` ; `/* insert `p` into the mesh */`
    `Update` SkinnyTriangles;
end

- If a point p_i is inserted inside the selection circle of skinny triangle, then p_i is called a *Type-A* point.
- If a point p_i is inserted on an encroached segment s, and there is no non-free vertex strictly inside the diametral circle of s and visible from p_i, then p_i is called a *Type-B* point. Type-B points are only inserted in the Type-B selection interval of encroached segments.
- If a point p_i is inserted on an encroached segment s, and there is at least one non-free vertex strictly inside the diametral circle of s and visible from p_i, then p_i is called a *Type-C* point. Type-C points are only inserted in the

Type-C selection interval of encroached segments.

For example, the vertex p_i in Figure 3.1(left), Figure 3.1(middle), and Figure 3.1(right) is a Type-A, Type-B, and a Type-C point respectively.

Notice that the parent of Type-B and Type-C points is not the same as in our previous Generalized Refinement algorithm presented in [7]: the parent $\hat{p}$ of a Type-C point p is now a non-free vertex strictly inside the diametral circle of an encroached segment s, regardless of whether $\hat{p}$ lies on a segment incident to s or not. This change is not necessary, but it further simplifies the proofs of the sections followed.

The analysis below assumes that all angles in the input PSLG are not less than 60°. (Input angles less than 60 degrees can be removed via postprocessing techniques [9] or via concentric circular shell splitting [21, 23], but without guarantee of good grading.)

4. Point Spacing Theorem. The main result of this section is Theorem 4.6 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 GCDR 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.

First, we prove Lemmas 4.1, 4.2, 4.3, 4.4, and 4.5 that establish important relations used in the proof of Theorem 4.6 as well as in the proof of good grading in Section 5. Lemmas 4.1, 4.2, and 4.3 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 4.4 relates the length of the encroached segment with the length of the circumradius of the encroaching triangle. Finally, Lemma 4.5 determines which of the edges of a skinny triangle can be the shortest.

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

$$R(p_i) \geq \delta_2 r, \tag{4.1}$$

where r is the circumradius of the corresponding skinny triangle.

Proof. Consider Figure 3.1(left). By the way we defined Type-A Steiner points (see section 3), p_i is actually inserted into the mesh; therefore, there is no segment that lies between p_i and the skinny triangle $t = \triangle p_k p_l p_m$ (i.e., t does not encroach upon any segment). By the constrained Delaunay property, t's circumcircle does not contain vertices visible from t's interior. Since p_i and t lie on the same side of any constrained segment, there is no vertex inside the circumcircle which is visible from p_i. Therefore, the donut between the boundary of the circumcircle and the boundary of the selection circle cannot contain points visible from p_i. Thus, the distance from p_i to the closest mesh vertex visible from p_i has to be greater than or equal to the width of the donut. This implies that the insertion radius of p_i has to be greater than or equal to the width of the donut which is equal to $\delta_2 r$. □

LEMMA 4.2. *If a Steiner point p_i, inserted on an encroached segment s, is of Type-B, and $\hat{p}_i$ is either a Type-A point or a non-free external parent, then*

- *there are no non-free vertices inside the diametral circle of s that are visible from p_i, and*
- *the following inequality holds:*

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

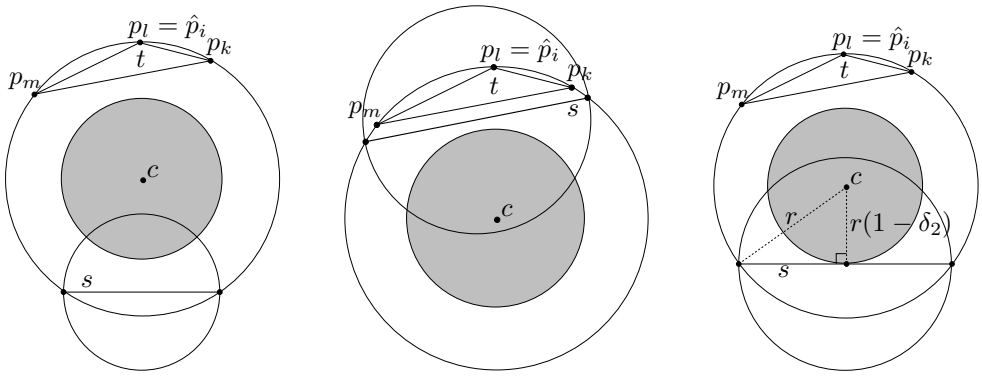

FIG. 4.1. *The shaded circle represents the selection circle of the skinny triangle t. Point c is the center of the selection circles.* ***(Left)*** *Segment s does not intersect the selection circle, but t cannot encroach upon s.* ***(Middle)*** *Segment s does not intersect the selection circle, t encroaches upon s, but $\hat{p}_i$ is not an external parent, since it lies strictly inside the diametral circle of s.* ***(Right)*** *If s intersects the selection circle, then its length is minimized when it is tangent to the selection circle. The radius of t's circumcircle is denoted as r.*

Proof. For the first part, for the sake of contradiction, assume that the diametral circle of s is not empty of non-free vertices visible from p_i. Let p_j be the closest-to-p_i, non-free vertex which is inside the diametral circle of s and visible from p_i. By Definition 2.5 (Rule-4), p_j is the parent of p_i. However, p_j is neither a Type-A point (since it is a non-free vertex) nor an external parent (since it lies inside the diametral circle of s): a contradiction.

For the second part, recall that all the free vertices which are inside the diametral circle of $s = e\,(p_u p_v)$ (see Figure 3.1(middle)) and visible from the interior of s are deleted. Also, from the first part above, there are no non-free vertices which lie inside the diametral circle of s and are visible from p_i. Therefore, one of the endpoints of s —say the endpoint p_v —is the vertex closest to p_i among the vertices that are visible from p_i. This means that $R(p_i) = \|p_i - p_v\|$, and from the definition of the Type-B selection interval, $R\,(p_i) \geq \delta_1 \frac{|s|}{2}$. □

LEMMA 4.3. *If p_i, inserted on an encroached segment s, is of Type-C, then*

- *if $\hat{p}_i$ is the closest vertex to p_i, the following equality holds:*

$$R\,(p_i) = \|p_i - \hat{p}_i\|, \tag{4.3}$$

or

- *if one of the endpoints of s is closest to p_i, the following inequality holds:*

$$R\,(p_i) \geq \mu \frac{|s|}{2}. \tag{4.4}$$

Proof. Consider Figure 3.1(right). By the way we defined a Type-C point, there is at least one non-free vertex strictly inside the diametral circle of $s = e\,(p_u p_v)$ and visible from the interior of s (i.e., visible from p_i). Therefore, by Definition 2.5 (Rule-4), $\hat{p}_i$ is the closest-to-p_i, non-free vertex visible from the interior of s. Recall, however, that all the free vertices which are inside the diametral circle of s and visible from the interior of s are deleted. Therefore, there are two scenarios: either $\hat{p}_i$ or one of the endpoints of s is the closest vertex to p_i. In the first case, equation (4.3) holds

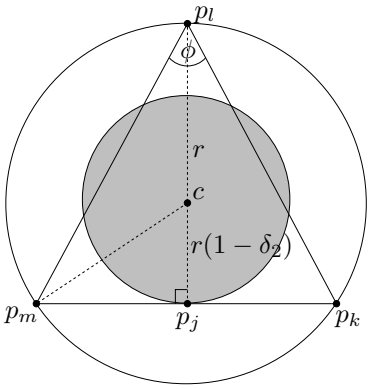

FIG. 4.2. *The skinny triangle $t = \triangle p_k p_l p_m$ where $e(p_m p_k)$ is the separator of the encroached segment. Since $e(p_m p_k)$ is tangent to the selection circle at the point p_j, angle ϕ takes its minimum value. Without loss of generality, t is an isosceles triangle with $e(p_m p_k)$ being the base.*

by definition of the insertion radius (Definition 2.4). In the second case, since p_i is inserted in the Type-C selection interval of s, p_i is separated from the endpoints of s by a distance at least $\mu\frac{|s|}{2}$, and inequality (4.4) holds. □

LEMMA 4.4. *Let p_i be a Steiner point inserted on a segment s encroached upon by a skinny triangle t with circumradius equal to r. If the parent of p_i is an external parent, then*

- *s intersects t's selection circle, and*
- *the following inequality holds:*

$$\frac{|s|}{2} \geq r\sqrt{2\delta_2 - \delta_2^2}. \tag{4.5}$$

Proof. From Remark 3, we obtain that the parent $\hat{p_i}$ of p_i is a vertex of t (in fact, $\hat{p_i}$ is the most recently inserted vertex of t's shortest edge). We claim that s has to intersect t's selection circle. Indeed, if it does not, either s cannot be encroached upon by t (see Figure 4.1(left)), or p_i cannot be an external parent (see Figure 4.1(middle)): a contradiction.

For the second part, observe that the length of s reaches its smallest value when s is tangent to t's selection circle, and its endpoints lie precisely on t's circumcircle. See Figure 4.1(right) for an illustration. From the right triangle formed, we obtain that $\frac{|s|}{2}$ is at least $\sqrt{r^2 - r^2(1-\delta_2)^2}$, and equation (4.5) holds. □

LEMMA 4.5. *Let triangle t encroach upon the constrained segment s. The separator of s cannot be the shortest edge of t.*

Proof. See Figure 3.1(middle). From Definition 2.8, the separator of $s = e(p_u p_v)$ is the edge $e(p_m p_k)$. We will prove the Lemma by showing that the angle $\angle p_k p_l p_m$ cannot be the smallest angle of the encroaching triangle $t = \triangle p_k p_l p_m$.

For the sake of contradiction, assume that $\phi = \angle p_k p_l p_m$ is the smallest angle of t. It is well-known that all inscribed angles subtended by the same arc of a circle are equal. Therefore, we can turn t into an isosceles triangle, without changing the value of ϕ, by moving appropriately the point p_l on t's circumcircle. Now, observe that the value of ϕ decreases as the endpoints of $e(p_m p_k)$ move on t's circumcircle and

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

TABLE 4.1

All possible type combinations of p_i and $\hat{p}_i$. The cells above labels "external" and "non-external" correspond to the cases when $\hat{p}_i$ is an external and a non-external parent respectively. Also, the cells above labels "incident" and "non-incident" correspond to the cases when p_i and $\hat{p}_i$ lie on incident and non-incident constrained segments respectively. Each of the cases (n) is analyzed separately.

away from the point p_l. Thus, the smallest value that ϕ can take is when $e(p_m p_k)$ is tangent to the selection circle of t, as depicted in Figure 4.2. Note that the edge $e(p_m p_k)$ cannot move further away from p_l, since otherwise t could not encroach upon s any more.

See Figure 4.2. From the right triangle $\triangle p_m p_j c$, we obtain that $\|p_m - p_j\| = r\sqrt{\delta_2(2-\delta_2)}$. Similarly, from the right triangle $\triangle p_m p_j p_l$, we obtain that $\|p_m - p_l\| = r\sqrt{2(2-\delta_2)}$. Therefore, we have that

$$\begin{aligned}
\sin\tfrac{\phi}{2} &= \tfrac{\|p_m-p_j\|}{\|p_m-p_l\|} && \text{(from the right triangle } \triangle p_m p_j p_l\text{)} \\
&= \tfrac{r\sqrt{\delta_2(2-\delta_2)}}{r\sqrt{2(2-\delta_2)}} \\
&= \sqrt{\tfrac{\delta_2}{2}} \\
&\geq \sqrt{\tfrac{1}{2\bar{\rho}}} && \text{(from Equation (3.1))} \\
&= \sqrt{\sin\theta} \\
&\geq \sin\theta && (0 < \sin\theta < 1\text{; therefore } \sqrt{\sin\theta} > \sin\theta)
\end{aligned}$$

yielding that $\phi > 2\theta$.

However, since t is a skinny triangle, its smallest angle must be no larger than θ: $\phi \leq \theta$, a contradiction. □

THEOREM 4.6 (Point Spacing Theorem). *With the use of the GCDR algorithm either*

$$R(p_i) \geq C_n \cdot R(\hat{p}_i), \quad n = 1, 2, 3, 4 \tag{4.6}$$

or

$$R(p_i) \geq C_n \cdot \mathrm{lfs}(p_i), \quad n = 5, \tag{4.7}$$

where C_n are defined separately for each of the cases (n) from Table 4.1 as follows: $C_1 = \bar{\rho}\delta_2$, $C_2 = \frac{2\delta_1}{\sqrt{5}}$, $C_3 = \bar{\rho}\delta_1\sqrt{2\delta_2 - \delta_2^2}$, $C_4 = \frac{\mu}{2\cos\alpha_{\min}}$, $C_5 = \frac{\mu}{2-\mu}$, where $\alpha_{\min}$ is the minimum input angle of the PSLG.

Proof.

Case (1) By the definition of the parent vertex, Definition 2.5 (Rule-2), $\hat{p}_i$ is the most recently inserted endpoint of the shortest edge of the triangle. Consider

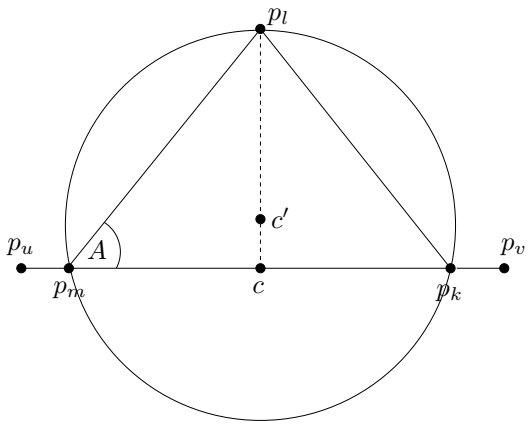

FIG. 4.3. *Points c and c' are the centers of the encroached segment $s = e(p_u p_v)$ and the circumcircle of the encroaching triangle $t = \triangle p_m p_l p_k$ respectively. The endpoints of the separator of s (points p_k and p_m) have moved exactly on s, while t is an isosceles triangle (with $e(p_m p_k)$ being the base) maximizing in this way the upper bound of $R(\hat{p_i})$.*

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

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

therefore, (4.6) 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) See Figure 3.1(middle). In this case, p_i is a Type-B point inserted on an encroached, constrained segment $s = e(p_u p_v)$, and its parent $\hat{p_i}$ is of Type-A. Therefore, Lemma 4.2 holds.

From the first part of Lemma 4.2, we have that there are no non-free vertices that are inside the diametral circle of s and visible from p_i. Therefore, by Definition 2.5 (Rule-3), the parent $\hat{p_i}$ of p_i is the most recently inserted vertex of the shortest edge of the encroaching triangle $t = \triangle p_k p_l p_m$. By Lemma 4.5, $e(p_m p_k)$ cannot be the shortest edge of t, since it is the separator of s. Therefore, the shortest edge of t is either $e(p_l p_m)$ or $e(p_l p_k)$. If $e(p_l p_m)$ is shorter than $e(p_l p_k)$, then the parent is the vertex p_l or p_m. In either case, $R(\hat{p_i}) \leq \|p_l - p_m\|$ from Remark 1; otherwise, $e(p_l p_k)$ is the shortest edge and thus the parent is the vertex p_l or p_k. In either case, $R(\hat{p_i}) \leq \|p_l - p_k\|$ from Remark 1. Hence, no matter which exact vertex is the parent, we obtain that $R(\hat{p_i}) \leq \min\{\|p_l - p_m\|, \|p_l - p_k\|\}$.

Furthermore, from the second part of Lemma 4.2, we obtain that $R(p_i) \geq \delta_1 \frac{\|p_u - p_v\|}{2}$ (inequality (4.2)).

We next try to find an upper bound for the ratio $\frac{R(\hat{p_i})}{R(p_i)}$. Without loss of generality, assume that segment s has been rotated around its midpoint c in such a way that

the separator of s (i.e., $e(p_m p_k)$) is parallel to s, as depicted in Figure 3.1(middle). Note that this rotation does not change either the upper bound of $R(\hat{p_i})$ or the lower bound of $R(p_i)$ because both the length of s and t's vertices remain intact.

Keeping the lower bound of $R(p_i)$ fixed (i.e., keeping the position of s's endpoints fixed), we will first try to calculate what is the maximum value $R(\hat{p_i})$ can reach. See Figure 3.1(middle): by moving the endpoints of the edge $e(p_m p_k)$ (the separator of the encroached segment) on t's circumcircle and towards the encroached segment $e(p_u p_v)$, $R(\hat{p_i})$ does not decrease, because the length of the edges $e(p_l p_m)$ and $e(p_l p_k)$ increases. Note that $e(p_m p_k)$ can at most fall on $e(p_u p_v)$, since otherwise $e(p_u p_v)$ would intersect the interior of t. Also, by moving appropriately the vertex p_l on the circle between the endpoints of $e(p_m p_k)$, we can turn t into an isosceles triangle, and therefore we can further increase the quantity $\min\{\|p_l - p_m\|, \|p_l - p_k\|\}$. See Figure 4.3 for an illustration.

$$\begin{aligned}
R(\hat{p_i}) &\leq \min\{\|p_l - p_m\|, \|p_l - p_k\|\} \\
&= \|p_l - p_m\| && \text{(since } t \text{ is turned into an isosceles triangle)} \\
&= \tfrac{\|p_m - c\|}{\cos A} && \text{(from the right triangle } \triangle p_m c p_l \text{ of Figure 4.3)} \\
&\leq \tfrac{\|p_u - p_v\|}{2\cos A} \\
&\leq \tfrac{R(p)}{\delta_1 \cos A} && \text{(from inequality (4.2)).}
\end{aligned}$$

However, Lemma 4.5 also implies that the minimum angle of t is A and since t is a skinny triangle, we have that $A \leq \theta$. Therefore, we finally get that

$$R(\hat{p_i}) \leq \frac{R(p)}{\delta_1 \cos\theta}. \tag{4.8}$$

Since θ is an acute angle, we obtain that:

$$\begin{aligned}
\cos\theta &= \sqrt{1 - \sin^2\theta} \\
&= \sqrt{1 - \tfrac{1}{4\bar{\rho}^2}} && \text{(since } \sin\theta = \tfrac{1}{2\bar{\rho}} \text{ by definition)} \\
&\geq \sqrt{1 - \tfrac{1}{5}} && \text{(since } \bar{\rho} \geq \tfrac{\sqrt{5}}{2} \text{ from Definition 3.1)} \\
&= \tfrac{2}{\sqrt{5}},
\end{aligned}$$

and inequality (4.8) becomes

$$R(\hat{p_i}) \leq \frac{\sqrt{5} R(p)}{2\delta_1}.$$

Therefore, (4.6) holds with $C_2 = \frac{2\delta_1}{\sqrt{5}}$.

Case (3) Let t be the triangle encroaching upon s, l be its shortest edge, and r be its circumradius. From Remark 3, we have that $\hat{p_i}$ is the most recently inserted vertex of l. Since t is a poor triangle, $|l|$ can at most be equal to $\frac{r}{\bar{\rho}}$. Therefore, from Remark 1, we obtain that :

$$R(\hat{p_i}) \leq \frac{r}{\bar{\rho}}. \tag{4.9}$$

Also, notice that:

$$\begin{aligned}
R(p_i) &\geq \delta_1 \tfrac{|s|}{2} && \text{(from inequality (4.2))} \\
&\geq \delta_1 r \sqrt{2\delta_2 - \delta_2^2} && \text{(from inequality (4.5))} \\
&\geq \delta_1 \bar{\rho} R(\hat{p_i}) \sqrt{2\delta_2 - \delta_2^2} && \text{(from inequality (4.9)).}
\end{aligned}$$

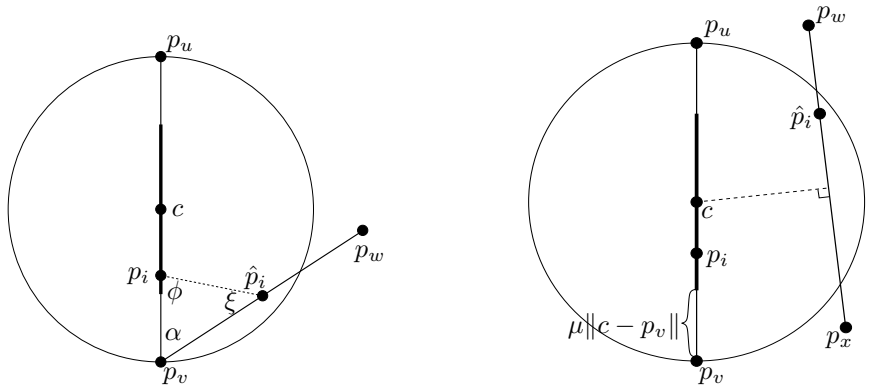

FIG. 4.4. *Point p_i is a Type-C point inserted on the encroached segment s. Point $\hat{p}_i$ is a non-external parent. Point c is not part of the mesh; it is an auxiliary point that denotes the center of the segments.* ***(Left)*** *Points p_i and $\hat{p}_i$ lie on incident constrained segments separated 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-incident constrained segments (case (5) from Table 4.1).*

Therefore, in this case, equation (4.6) holds with $C_3 = \bar{\rho}\delta_1\sqrt{2\delta_2 - \delta_2^2}$.

The argument above holds for each type of $\hat{p}_i$ shown in Table 4.1, because it does not involve the properties of $\hat{p}_i$ specific for a particular type.

Case (4) Since p_i is a Type-C point, Lemma 4.3 holds. Consider Figure 4.4(left). Based on Lemma 4.3, we separate 2 possibilities:

(a) If the parent $\hat{p}_i$ is the closest point to p_i, then

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

We wish to determine what values angle ϕ can take. Since $\|p_i - \hat{p}_i\| \leq \|p_i - p_v\|$ by our assumption, we get that $\alpha \leq \xi$ from $\triangle p_i p_v \hat{p}_i$. Therefore, $\phi \leq 180^\circ - 2\alpha$. This implies that

$$\begin{array}{rl} \frac{R(p_i)}{R(\hat{p}_i)} & \geq \frac{\sin\alpha}{\sin(180-2\alpha)} \\ & = \frac{\sin\alpha}{\sin(2\alpha)} \\ & = \frac{\sin\alpha}{2\sin\alpha\cos\alpha} \\ & = \frac{1}{2\cos\alpha}, \end{array}$$

yielding that

$$R(p_i) \geq \frac{1}{2\cos\alpha_{\min}} R(\hat{p}_i). \tag{4.10}$$

(b) If an endpoint of the constrained encroached segment $e(p_u p_v)$ is the closest point to p_i, then

$$\begin{array}{rcll} \frac{R(p_i)}{R(\hat{p}_i)} & \geq & \frac{\mu\frac{\|p_u - p_v\|}{2}}{R(\hat{p}_i)} & \text{(from inequality (4.4))} \\ & \geq & \frac{\mu\|p_u - p_v\|}{2\|\hat{p}_i - p_v\|} & \text{(from Remark 1).} \end{array}$$

An upper bound for the distance $\|\hat{p}_i - p_v\|$ is obtained when $\hat{p}_i$ lies on the diametral circle of the segment $e(p_u p_v)$ (in fact, $\hat{p}_i$ cannot lie exactly on the

diametral circle, since it is a non-external vertex). From the isosceles triangle $\triangle \hat{p}_i c p_v$ (see Figure 4.4(left)), we get that $\|\hat{p}_i - p_v\| < \|p_u - p_v\| \cos\alpha$, and therefore

$$\begin{array}{rcl} \frac{R(p_i)}{R(\hat{p}_i)} & > & \frac{\mu\|p_u - p_v\|}{2\|p_u - p_v\| cos\alpha} \\ & = & \frac{\mu}{2\cos\alpha}, \end{array}$$

obtaining in this case that

$$R(p_i) > \frac{\mu}{2\cos\alpha_{\min}} R(\hat{p}_i). \tag{4.11}$$

In both cases, $C_4 = \frac{\mu}{2\cos\alpha_{\min}}$ satisfies both equation (4.10) and (4.11), and thus (4.6) holds.

The argument above holds for each type of $\hat{p}_i$ shown in Table 4.1, because it does not involve the properties of $\hat{p}_i$ specific for a particular type.

Case (5) See Figure 4.4(right). Based on Lemma 4.3, we, again, separate two possibilities:

(a) If $\hat{p}_i$ is the vertex closest to p_i, then, from equation (4.3), $R(p_i) = \|p_i - \hat{p}_i\|$. Since, however, p_i and $\hat{p}_i$ lie on non-incident features, we obtain that $\text{lfs}(p_i) \leq \|p_i - \hat{p}_i\|$ (see Definition 2.2), yielding that $R(p_i) \geq \text{lfs}(p_i)$.

(b) Otherwise, an endpoint of the constrained encroached segment $e(p_u p_v)$ is the closest point to p_i. Then, if c is the center of s, by Definition 2.2 of the lfs () function,

$$\text{lfs}(c) \leq \|c - \hat{p}_i\|. \tag{4.12}$$

Therefore,

$$\begin{array}{rcll} \text{lfs}(p_i) & \leq & \text{lfs}(c) + \|p_i - c\| & \text{(from Lemma 2.3)} \\ & \leq & \|c - \hat{p}_i\| + \|p_i - c\| & \text{(from (4.12))} \\ & \leq & \frac{|s|}{2} + \|p_i - c\| & \text{(because } \hat{p}_i \text{ is inside the diametral circle of } s) \\ & \leq & \frac{|s|}{2} + (1-\mu)\frac{|s|}{2} & \text{(since } p_i \text{ lies in the Type-C selection interval of } s) \\ & = & (2-\mu)\frac{|s|}{2} & \\ & \leq & (2-\mu)\frac{R(p_i)}{\mu} & \text{(from inequality (4.4)).} \end{array}$$

In both cases, $C_5 = \frac{\mu}{2-\mu}$ satisfies the inequality (4.7).

The argument above holds for each type of $\hat{p}_i$ shown in Table 4.1, because it does not involve the properties of $\hat{p}_i$ specific for a particular type. □

5. Proof of Good Grading. The main result of this section is Theorem 5.5 which proves that GCDR produces well-graded triangles. Theorem 5.5 will also allow us to prove termination, since it is possible to bound from below the closest distance of any two visible vertices. Notice that we first prove good grading and then we prove termination.

First, we prove Lemmas 5.1, 5.2, and 5.3 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 5.4 which shows that the vertex density in a point is bounded from above by a linear function of the density in its parent. Lemma 5.4 is proved only for cases (1)–(4), since for case (5) the relation of the insertion radius to the local feature size

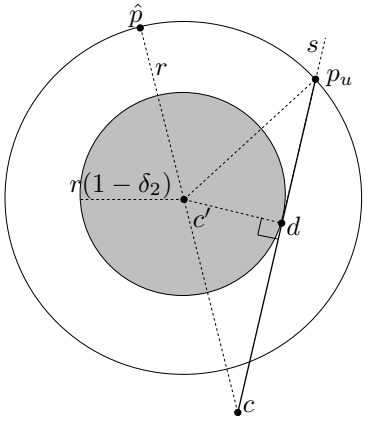

FIG. 5.1. *The circumcircle and selection circle of the triangle t which encroaches upon the constrained segment s. The point c is the center of s. The parent $\hat{p}$ has been moved on the position that maximizes the distance $\|c-\hat{p}\|$, while s remained fixed. Next, keeping the position of c fixed, we are moving the endpoints of s such that $|s|$ is minimized; in this case, s is tangent to the selection circle, and at least one endpoint of s lies precisely on the circumcircle.*

proved by Theorem 5.5 follows directly from the Spacing Theorem. Finally, we prove Theorem 5.5 by enumerating all possible type combinations of a point and its parent.

LEMMA 5.1. *If p is of Type-A, then*

$$\|p - \hat{p}\| \leq (2 - \delta_2)r, \tag{5.1}$$

where r is the circumradius of the skinny triangle t.

Proof. If c is the circumcenter of t, then

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

□

LEMMA 5.2. *If p, lying on an encroached, constrained segment s, is of Type-B, and its parent $\hat{p}$ is an external parent, then*

$$\|p - \hat{p}\| \leq \left(1 + \sqrt{\frac{2-\delta_2}{\delta_2}} - \delta_1\right) \frac{|s|}{2}. \tag{5.2}$$

Proof. By Definition 2.5 (Rule-2), $\hat{p}$ is a vertex of the skinny triangle t encroaching upon s.

First, we will try to find a tight upper bound for the ratio $\frac{\|c-\hat{p}\|}{|s|}$, where c is the center of s.

Keeping the position of c and the position of s's endpoints fixed, we are trying to increase the distance $\|c-\hat{p}\|$ by moving only the parent $\hat{p}$ on t's circumcircle. See Figure 5.1: the distance $\|c-\hat{p}\|$ reaches its largest value when $\hat{p}$ lies on the extension of the straight line connecting c and t's circumcenter c'.

Conversely, keeping the position of c fixed, we are trying to decrease $|s|$ by moving only the endpoints of s: the length of s reaches its smallest value when s is tangent to the selection circle, and one of its endpoints—say the point p_u—lies precisely on the circumcircle. See Figure 5.1 for an illustration. Note that if s did not intersect the selection circle at all, then the first part of Lemma 4.4 would be violated: a contradiction.

From the right triangle $\triangle cdc'$ (see Figure 5.1), we obtain that:

$$\begin{aligned}
\|c-c'\|^2 &= \|d-c'\|^2 + \|c-d\|^2 \\
&= r^2(1-\delta_2)^2 + \|c-d\|^2 \\
&= r^2(1-\delta_2)^2 + \left(\tfrac{|s|}{2} - \|p_u-d\|\right)^2 \\
&= r^2(1-\delta_2)^2 + \left(\tfrac{|s|}{2} - r\sqrt{2\delta_2-\delta_2^2}\right)^2 \qquad \text{(from the right triangle } \triangle kdp_u\text{)} \\
&= r^2 + \tfrac{|s|^2}{4} - r|s|\sqrt{2\delta_2-\delta_2^2},
\end{aligned}$$

yielding that:

$$\|c-c'\| = \sqrt{r^2 + \frac{|s|^2}{4} - r|s|\sqrt{2\delta_2-\delta_2^2}}. \tag{5.3}$$

We also have that:

$$\begin{aligned}
\tfrac{\|c-\hat{p}\|}{|s|} &= \tfrac{r+\|c-c'\|}{|s|} \qquad \text{(since } c, \hat{p} \text{ and } c' \text{ are collinear)} \\
&= \tfrac{r+\sqrt{r^2+\frac{|s|^2}{4}-rs\sqrt{2\delta_2-\delta_2^2}}}{|s|} \qquad \text{(from Equation (5.3)),}
\end{aligned}$$

and after the simplification, we finally get that:

$$\frac{\|c-\hat{p}\|}{|s|} = \left(\frac{r}{|s|}\right) + \sqrt{\left(\frac{r}{|s|}\right)^2 + \frac{1}{4} - \left(\frac{r}{|s|}\right)\sqrt{2\delta_2-\delta_2^2}}. \tag{5.4}$$

Basic calculus, however, reveals that the right-hand part of Equation (5.4) is an always increasing function with respect to the "variable" $\frac{r}{|s|}$. Also, from inequality (4.5), we already know that $\frac{r}{|s|} \leq \frac{1}{2\sqrt{2\delta_2-\delta_2^2}}$. Thus, by replacing $\frac{r}{|s|}$ in the right-hand part of Equation (5.4) with its largest value and simplifying the result, we get that:

$$\frac{\|c-\hat{p}\|}{|s|} \leq \frac{1}{2}\sqrt{\frac{2-\delta_2}{\delta_2}}. \tag{5.5}$$

Lastly, the desired outcome follows:

$$\begin{aligned}
\|p-\hat{p}\| &\leq \|p-c\| + \|c-\hat{p}\| && \text{(from the triangle inequality)} \\
&\leq (1-\delta_1)\tfrac{|s|}{2} + \|c-\hat{p}\| && \text{(since } p \text{ is in the Type-B selection interval)} \\
&\leq (1-\delta_1)\tfrac{|s|}{2} + \tfrac{|s|}{2}\sqrt{\tfrac{2-\delta_2}{\delta_2}} && \text{(from Equation (5.5))} \\
&= \left(1+\sqrt{\tfrac{2-\delta_2}{\delta_2}} - \delta_1\right)\tfrac{|s|}{2}.
\end{aligned}$$

□

Lemma 5.3. *If p_i is of Type-C, then*

$$\|p_i - \hat{p}_i\| \leq (2-\mu)\frac{|s|}{2}, \tag{5.6}$$

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

Proof. If c is the center of s, then

$$\begin{array}{rcll} \|p_i - \hat{p}_i\| & \leq & \|p_i - c\| + \|c - \hat{p}_i\| & \text{(from the triangle inequality)} \\ & \leq & (1-\mu)\frac{|s|}{2} + \|c - \hat{p}_i\| & \text{(since } p_i \text{ is in the Type-C selection interval)} \\ & \leq & (1-\mu)\frac{|s|}{2} + \frac{|s|}{2} & \text{(since } \hat{p}_i \text{ is a non-external parent)} \\ & = & (2-\mu)\frac{|s|}{2}. & \end{array}$$

□

LEMMA 5.4. *If p is a vertex of the mesh inserted by the GCDR algorithm, and C_n ($n = 1, 2, 3, 4$) are the constants specified by Theorem 4.6 for the corresponding cases listed in Table 4.1, then the following inequality holds:*

$$D(p) \leq B_n + \frac{D(\hat{p})}{C_n}, \quad n = 1, 2, 3, 4 \tag{5.7}$$

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

Proof.

First, we prove the inequality

$$\|p - \hat{p}\| \leq B_n \cdot R(p) \tag{5.8}$$

for each of the cases below.

Case (1):

$$\begin{array}{rcll} \|p - \hat{p}\| & \leq & (2-\delta_2)r & \text{(from Lemma 5.1)} \\ & = & \frac{2-\delta_2}{\delta_2}\delta_2 r & \\ & \leq & \frac{2-\delta_2}{\delta_2}R(p) & \text{(from Lemma 4.1);} \end{array}$$

therefore, inequality (5.8) can be satisfied with $B_1 = \frac{2-\delta_2}{\delta_2}$.

Case (2): In this case, p_i is of Type-B and its parent $\hat{p}_i$ is a Type-A point. We assume that p_i is an external parent, since if it was a non-external parent, the distance between p_i and $\hat{p}_i$ would be smaller.

$$\begin{array}{rcll} \|p - \hat{p}\| & \leq & \left(1 + \sqrt{\frac{2-\delta_2}{\delta_2}} - \delta_1\right)\frac{|s|}{2} & \text{(from Lemma 5.2)} \\ & = & \frac{1+\sqrt{\frac{2-\delta_2}{\delta_2}}-\delta_1}{\delta_1}\delta_1\frac{|s|}{2} & \\ & \leq & \frac{1+\sqrt{\frac{2-\delta_2}{\delta_2}}-\delta_1}{\delta_1}R(p) & \text{(from inequality (4.2));} \end{array}$$

therefore, inequality (5.8) can be satisfied with $B_2 = \frac{1+\sqrt{\frac{2-\delta_2}{\delta_2}}-\delta_1}{\delta_1}$.

Case (3): The analysis is exactly the same as in case (2), since p_i is a Type-B point, and $\hat{p}_i$ is an external parent. Therefore, inequality (5.8) can be satisfied with $B_3 = B_2$.

Case (4): Based on Lemma 4.3, we separate two possibilities:

(a) If $\hat{p}_i$ is closest to p_i, then $\|p_i - \hat{p}_i\| = R(p_i)$ from equality (4.3).

(b) If an endpoint of s is closest to p_i, then

$$\begin{array}{rcll} \|p_i - \hat{p}_i\| & \leq & (2-\mu)\frac{|s|}{2} & \text{(from Lemma 5.3)} \\ & = & \frac{2-\mu}{\mu}\mu\frac{|s|}{2} & \\ & \leq & \frac{2-\mu}{\mu}R(p_i) & \text{(from inequality (4.3));} \end{array}$$

therefore, inequality (5.8) can be satisfied with $B_4 = \frac{2-\mu}{\mu}$ in both cases.

Now, for all cases (1)–(4),

$$\begin{array}{rcll} \text{lfs}\,(p) & \leq & \text{lfs}\,(\hat{p}) + \|p - \hat{p}\| & \text{(from Lemma 2.3)} \\ & \leq & \text{lfs}\,(\hat{p}) + B_n R\,(p) & \text{(from (5.8))} \\ & = & D\,(\hat{p})\, R\,(\hat{p}) + B_n R\,(p) & \text{(from (2.2))} \\ & \leq & D\,(\hat{p})\, \frac{R(p)}{C_n} + B_n R\,(p) & \text{(from Theorem 4.6).} \end{array}$$

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

THEOREM 5.5. *Suppose that the following three inequalities hold:*

I-1. $\bar{\rho} > \frac{\sqrt{5}}{2}$ *(implying a lower angle bound of* 26.56°*), and*

I-2. $\alpha_{\min} > 60^\circ$ *(recall that* $\alpha_{\min}$ *is the smallest input angle).*

Then, there exist fixed constants $D_A \geq 1$*,* $D_B \geq 1$*, and* $D_C \geq 1$ *such that, for any vertex* p *inserted by the GCDR algorithm, the following inequalities hold:*

$$D\,(p) \leq \begin{cases} D_A & \textit{if } p \textit{ is of Type-A}, \\ D_B & \textit{if } p \textit{ is of Type-B}, \\ D_C & \textit{if } p \textit{ is of Type-C}. \end{cases} \tag{5.9}$$

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 [23]. 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.,

$$D\,(\hat{p}) \leq \begin{cases} D_A & \text{if } \hat{p} \text{ is of Type-A}, \\ D_B & \text{if } \hat{p} \text{ is of Type-B}, \\ D_C & \text{if } \hat{p} \text{ is of Type-C}. \end{cases} \tag{5.10}$$

Inductive step: For each of the cases (n), $n = 1, 2, 3, 4$, we start with (5.7) 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 (5.9) can be satisfied if D_A, D_B, and D_C are chosen such that the following inequalities (5.11), (5.12), (5.13), (5.14), (5.15), (5.16), (5.17), and (5.18) hold:

Case (1):

$$B_1 + \frac{D_A}{C_1} \leq D_A, \tag{5.11}$$

$$B_1 + \frac{D_B}{C_1} \leq D_A, \tag{5.12}$$

$$B_1 + \frac{D_C}{C_1} \leq D_A, \tag{5.13}$$

Case (2):

$$B_2 + \frac{D_A}{C_2} \leq D_B, \tag{5.14}$$

Case (3):

$$B_3 + \frac{D_B}{C_3} \leq D_B, \tag{5.15}$$

$$B_3 + \frac{D_C}{C_3} \leq D_B, \tag{5.16}$$

Case (4):

$$B_4 + \frac{D_B}{C_4} \leq D_C, \tag{5.17}$$

$$B_4 + \frac{D_C}{C_4} \leq D_C, \tag{5.18}$$

For case (5), from Theorem 4.6, we have: $D(p) = \text{lfs}(p)/R(p) \leq 1/C_5$, i.e., the inequalities in (5.9) can be satisfied if D_C is chosen such that the following inequality holds:

Case (5):

$$\frac{1}{C_5} = B_4 \leq D_C, \tag{5.19}$$

Notice that since $B_n \geq 1$ for every $n = 1, 2, 3, 4$, the solution of the system above guarantees that D_A, D_B, D_C are larger than or equal to 1. Also notice that inequality (5.14) implies that $D_B > D_A$, because $\frac{1}{C_2} < 1$. Therefore, (5.11) can be ignored, since it follows from (5.12) and (5.14).

Inequalities (5.12), (5.14), (5.17), and (5.19) are satisfied when:

$$\begin{array}{rcl} D_A & = & \frac{B_2C_2+B_1C_1C_2}{C_1C_2-1}, \\ D_B & = & B_2 + \frac{B_2C_2+B_1C_1C_2}{(C_1C_2-1)C_2}, \\ D_C & = & \max\left\{B_4 + \frac{B_2}{C_4} + \frac{B_2C_2+B_1C_1C_2}{(C_1C_2-1)C_2C_4}, \frac{1}{C_5}\right\} \end{array} \tag{5.20}$$

Inequalities (5.12), (5.15), and (5.17) are satisfied when:

$$\begin{array}{rcl} D_A & = & B_1 + \frac{B_3C_3}{(C_3-1)C_1}, \\ D_B & = & \frac{B_3C_3}{C_3-1}, \\ D_C & = & B_4 + \frac{B_3C_3}{(C_3-1)C_4} \end{array} \tag{5.21}$$

Inequalities (5.13), (5.14), (5.15), and (5.18) are satisfied when:

$$\begin{array}{rcl} D_A & = & B_1 + \frac{B_4C_4}{(C_4-1)C_1}, \\ D_B & = & \max\left\{B_2 + \frac{B_1}{C_2} + \frac{B_4C_4}{(C_4-1)C_1C_2}, B_3 + \frac{B_4C_4}{(C_4-1)C_3}\right\}, \\ D_C & = & \frac{B_4C_4}{C_4-1} \end{array} \tag{5.22}$$

Combining the equations (5.20), (5.21), and (5.22), we finally get that:

$$\begin{array}{rcl} D_A & = & \max\Big\{\frac{B_2C_2+B_1C_1C_2}{C_1C_2-1}, B_1+\frac{B_3C_3}{(C_3-1)C_1}, B_1+\frac{B_4C_4}{(C_4-1)C_1}\Big\}, \\ D_B & = & \max\Big\{B_2+\frac{B_2C_2+B_1C_1C_2}{(C_1C_2-1)C_2}, \frac{B_3C_3}{C_3-1}, B_2+\frac{B_1}{C_2}+\frac{B_4C_4}{(C_4-1)C_1C_2}, B_3+\frac{B_4C_4}{(C_4-1)C_3}\Big\}, \\ D_C & = & \max\Big\{B_4+\frac{B_2}{C_4}+\frac{B_2C_2+B_1C_1C_2}{(C_1C_2-1)C_2C_4}, \frac{1}{C_5}, B_4+\frac{B_3C_3}{(C_3-1)C_4}, \frac{B_4C_4}{C_4-1}\Big\} \end{array} \tag{5.23}$$

Note that the inequalities I-1 and I-2 guarantee that the grading constants D_A, D_B, and D_C are well defined, i.e., they are neither infinite nor negative. □

The area (size) of the selection regions affects the grading of our algorithm, theoretically at least. Indeed, if either of the parameters δ_1, δ_2, μ decreases (thus increasing the area of the selection regions), then the lower bounds for the constants D_A, D_B, D_C of Theorem 5.5 increase making the grading worse. In practice, when we increase the area of the selection regions, the deterioration of grading is less severe (see Section 7).

6. Proof of Termination. The good grading of GCDR implies termination as well. For reasons of completeness, however, we present a concrete proof.

THEOREM 6.1. *GCDR terminates producing triangles with angles arbitrarily close to* $30°$.

Proof. To prove termination, it suffices to prove that no two vertices, visible to each other, are closer than a real constant $C' > 0$, since the insertion of an infinite number of vertices would necessarily introduce at some point two visible vertices closer than C'. From Remark 1, however, the distance of any two visible vertices is bounded from below by the insertion radius of one of these vertices. Therefore, it is adequate to prove that the insertion radius of any point is not less than the positive, real constant C'.

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 can occur in the following two cases:

1. A Type-B point p_i lies on an encroached segment s, and its parent is a non-free non-external vertex, i.e., $\hat{p_i}$ is a non-free point that lies strictly inside the diametral circle of s. Recall that the parent of any point p_i is visible from p_i (see Lemma 2.6). By the way we defined Type-B points, however, there is no non-free vertex strictly inside the diametral circle of s and visible from p_i: a contradiction.
2. A Type-C point p_i lies on an encroached segment s, and its parent is either a free vertex or a non-free external parent. By the way we defined Type-C points, the diametral circle of s contains at least one non-free vertex visible from p_i. Therefore, from Definition 2.5 (Rule-4), the parent of p_i is the closest-to-p_i, non-free vertex that lies inside the diametral circle of s and is visible from p_i. This means that p_i can be neither a free vertex nor an external parent: a contradiction.

All the remaining parent-child combinations (marked with numbers) have been analyzed in Theorem 5.5. From that theorem it follows that with the use of the GCDR algorithm the insertion radius of any vertex is no less than $C' = \frac{\text{lfs}_{\min}}{\max\{D_A, D_B, D_C\}}$, where $\text{lfs}_{\min} = \min_{p\in\Omega} \text{lfs}\,(p) > 0$. Note that inequalities I-1 and I-2 of Theorem 5.5 imply that the positive constants D_A, D_B, D_C cannot approach infinity, and therefore C' is a positive real number. □

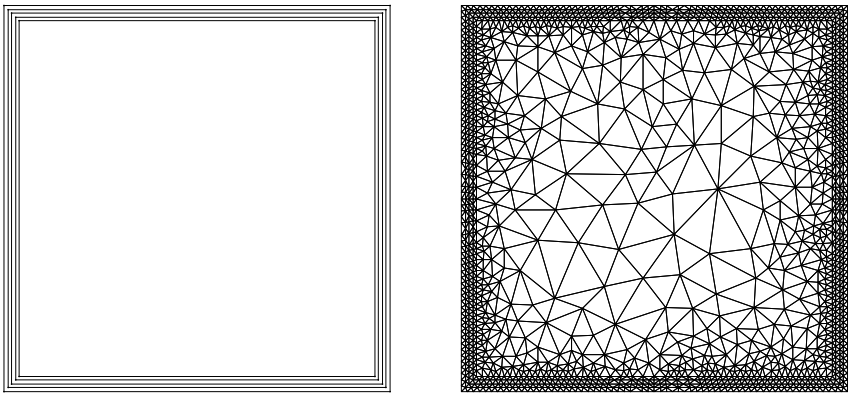

FIG. 7.1. ***(Left)*** *The input PSLG used for the experiments. It consists of 5 concentric squares. The horizontal and vertical distance between two adjacent squares is one hundredth of the side length of the outer-most square.* ***(Right)*** *The output mesh obtained by GCDR on the input PSLG with* $\bar{\rho}$ *being set to* 1*. GCDR terminates producing* 3200 *triangles whose angles are more than* 30°.

7. Experimental Evaluation. In this section, we experimentally evaluate the grading achieved by GCDR. We have implemented GCDR on top of the Computational Geometry Algorithms Library (CGAL) [2]. For the visualization of the mesh, we used the C++ Visualization Toolkit Library (VTK) [1].

The input PSLG used for the experiments is depicted in Figure 7.1. Other input PSLGs yielded similar results.

We conducted two experiments denoted as Experiment-1 and Experiment-2 respectively. In the Experiment-1, we executed GCDR 36 times. At every execution, we changed the size of the selection regions by altering the values of the parameters δ_2, δ_1, and μ. Also, for each execution, all the Steiner points inserted (or considered for insertion) by GCDR lied exactly on the boundary of selection circles or selection intervals. More precisely, every Type-A Steiner point is inserted as close to an arbitrary vertex of the corresponding skinny triangle as possible, and every Type-B/Type-C Steiner point is inserted as close to an arbitrary endpoint of the corresponding encroached segment as possible. In this way, we test our algorithm when the inserted Steiner points lie in extreme positions.

Table 7.1 summarizes the results for the first experiment under the label "Experiment-1". The quality upper bound is set to $\bar{\rho} = 2$ which means that $\delta_2 \geq \frac{\sqrt{5}}{2 \cdot 2} \approx 0.56$ and that $\delta_1 \geq \frac{0.56}{\delta_2}$. Since $\alpha_{\min} = 90^\circ$, we also have that $\mu \geq 0$. Each row of the table corresponds to a specific execution where the values for the parameters are shown in the second, third, and forth cell of each row. The table reports the largest density observed in practice per point type: the largest density observed among Type-A points (8^{th} column), among Type-B (9^{th} column), and among Type-C points (10^{th} column). For comparison, the table also depicts (at the 5^{th}, 6^{th}, and 7^{th} column) the theoretical upper bound of the density per point type for the respective configuration, as calculated in inequality (5.23) of Theorem 5.5.

The observed largest densities of Experiment-1 should be less than their theoretical counterparts. Indeed, the grading achieved in practice is much smaller than the theoretical bound of Theorem 5.5. This fact verifies the theory and also implies that GCDR behaves much better than theory suggests.

Also, notice that μ seems to strongly affect the size of the output mesh. Indeed,

TABLE 7.1

The 36 configurations used by Experiment-1 and Experiment-2. The results are shown in the last 7 columns.

							Experiment-1				Experiment-2		
Configuration Id	δ_2	δ_1	μ	D_A	D_B	D_C	$\max_{p \text{ is Type-A}} D(p)$	$\max_{p \text{ is Type-B}} D(p)$	$\max_{p \text{ is Type-C}} D(p)$	Number of triangles	$\max_{p \text{ is Type-A}} D(p)$	$\max_{p \text{ is Type-B}} D(p)$	$\max_{p \text{ is Type-C}} D(p)$
1	1.00	1.00	1.00	3.40	4.80	1.00	1.14	0.35	1.00	972	1.14	0.35	1.00
2	1.00	1.00	0.80	3.40	4.80	1.50	1.21	0.38	1.00	1164	1.43	0.50	1.00
3	1.00	1.00	0.60	3.40	4.80	2.33	1.29	0.49	1.00	1362	1.55	0.51	1.00
4	1.00	1.00	0.40	3.40	4.80	4.00	1.52	0.51	1.27	1661	1.58	0.51	1.21
5	1.00	0.90	1.00	4.25	6.50	1.00	1.14	0.36	1.00	980	1.15	0.39	1.00
6	1.00	0.90	0.80	4.25	6.50	1.50	1.21	0.42	1.00	1164	1.50	0.55	1.00
7	1.00	0.90	0.60	4.25	6.50	2.33	1.55	0.51	1.00	1360	1.57	0.56	1.00
8	1.00	0.90	0.40	4.25	6.50	4.00	1.51	0.54	1.27	1667	1.56	0.55	1.21
9	1.00	0.80	1.00	5.81	9.62	1.00	1.13	0.40	1.00	964	1.16	0.43	1.00
10	1.00	0.80	0.80	5.81	9.62	1.50	1.21	0.47	1.00	1164	1.53	0.60	1.00
11	1.00	0.80	0.60	5.81	9.62	2.33	1.29	0.58	1.00	1378	1.54	0.63	1.00
12	1.00	0.80	0.40	5.81	9.62	4.00	1.51	0.62	1.22	1663	1.54	0.62	1.33
13	0.90	1.00	1.00	4.85	6.52	1.00	1.36	0.35	1.00	1007	1.57	0.35	1.00
14	0.90	1.00	0.80	4.85	6.52	1.50	1.46	0.38	1.00	1161	1.58	0.51	1.00
15	0.90	1.00	0.60	4.85	6.52	2.33	1.54	0.46	1.00	1357	1.70	0.52	1.00
16	0.90	1.00	0.40	4.85	6.52	4.00	1.71	0.51	1.27	1675	1.74	0.51	1.19
17	0.90	0.90	1.00	6.35	9.22	1.00	1.33	0.43	1.00	957	1.58	0.42	1.00
18	0.90	0.90	0.80	6.35	9.22	1.50	1.46	0.42	1.00	1163	1.50	0.55	1.00
19	0.90	0.90	0.60	6.35	9.22	2.33	1.59	0.51	1.00	1359	1.61	0.54	1.00
20	0.90	0.90	0.40	6.35	9.22	4.00	1.71	0.56	1.22	1653	1.75	0.56	1.19
21	0.90	0.80	1.00	9.52	14.94	1.00	1.32	0.43	1.00	975	1.67	0.52	1.00
22	0.90	0.80	0.80	9.52	14.94	1.50	1.46	0.47	1.00	1165	1.49	0.61	1.00
23	0.90	0.80	0.60	9.52	14.94	2.33	1.54	0.59	1.00	1383	1.56	0.61	1.00
24	0.90	0.80	0.40	9.52	14.94	4.00	1.71	0.62	1.22	1661	1.68	0.61	1.27
25	0.80	1.00	1.00	7.52	9.63	1.00	1.66	0.35	1.00	1011	1.58	0.35	1.00
26	0.80	1.00	0.80	7.52	9.63	1.50	1.91	0.38	1.00	1161	1.91	0.52	1.00
27	0.80	1.00	0.60	7.52	9.63	2.33	1.81	0.47	1.00	1345	2.04	0.52	1.00
28	0.80	1.00	0.40	7.52	9.63	4.00	2.34	0.52	1.27	1651	2.01	0.52	1.25
29	0.80	0.90	1.00	10.82	14.92	1.00	1.68	0.43	1.00	991	1.94	0.42	1.00
30	0.80	0.90	0.80	10.82	14.92	1.50	1.91	0.42	1.00	1161	2.26	0.57	1.00
31	0.80	0.90	0.60	10.82	14.92	2.33	1.81	0.52	1.00	1361	1.86	0.56	1.00
32	0.80	0.90	0.40	10.82	14.92	4.00	2.21	0.56	1.22	1654	2.02	0.57	1.23
33	0.80	0.80	1.00	20.65	30.64	1.00	1.86	0.54	1.00	996	1.81	0.49	1.00
34	0.80	0.80	0.80	20.65	30.64	1.50	1.91	0.47	1.00	1163	1.73	0.62	1.00
35	0.80	0.80	0.60	20.65	30.64	2.33	1.88	0.63	1.00	1385	1.94	0.64	1.00
36	0.80	0.80	0.40	20.65	30.64	4.00	2.26	0.61	1.24	1664	2.36	0.64	1.27

no matter how large the Type-A and Type-B regions are, when the size of Type-C selection intervals increases (i.e., low values for μ), the number of triangles is considerably higher. On the contrary, for high values of μ, low values for δ_2 and δ_1 do not result in a high size output mesh. This fact is attributable to the high number of Type-C points: in all the executions, the number of Type-C points (inserted into the mesh) is 3 to 4 times more than the combined number of Type-A and Type-B points.

Although the Steiner points in Experiment-1 are inserted in extreme positions, it

is not certain whether the measured maximum densities are the worst. For example, see the 2nd configuration under the label Experiment-1: Type-C Steiner points lie farther from the center of the encroached segments than they do in the 1st configuration, but the maximum practical densities observed do not increase.

Therefore, we conducted another experiment which could potentially generate the highest (i.e., worst) practical densities. The results are illustrated in Table 7.1 under the label "Experiment-2". Here, each configuration of the previous experiment is repeated for 100 times (and not just once as before). Each time, the Steiner points inserted were randomly chosen from within their corresponding selection region (i.e., they do not always lie on the boundary of the selection regions as before). We report the highest density per point type among the densities observed during the 100 repetitions.

The maximum densities measured in Experiment-2 are slightly higher than before, but still less than the theoretical counterparts. These observations further verify the theory and suggest that GCDR behaves much better in practice.

8. Conclusions and Future Work. We have presented and implemented a constrained Delaunay refinement algorithm in two dimensions. Our algorithm is more flexible than the traditional approaches, since the user can choose which Steiner point to insert into the mesh among an infinite, enumerable number of choices. Our algorithm significantly increases the number of these choices over our previous Generalized algorithms [4–7]. The flexibility offered by Generalized algorithms can benefit sliver removal in three dimensions [10, 16], since the area from which Steiner points are selected now increases potentially allowing to achieve even better dihedral angles. These customizable point insertion strategies offered by our algorithm can also help in boundary recovery by inserting, for instance, points on the boundary while refining the mesh. Other applications of Generalized algorithms can be found in [7].

The parallelization of our Generalized algorithm is left as future work. A parallel Generalized algorithm would automatically imply the parallelization of any point insertion strategy as long as the Steiner points lie in the selection regions.

Lastly, we wish to implement our algorithm in three dimensions and identify point insertion strategies suitable for sliver removal.

9. Acknowledgments. We thank the anonymous reviewers from the SISC journal whose insightful comments on our paper [7] motivated some of the research presented in this manuscript. We also thank Dr. Alex Rand who pointed out the imprecision of our earlier work in [12]. This work is supported in part by NSF grants: CCF-1139864, CCF-1136538, and CSI-1136536 and by the John Simon Guggenheim Foundation and the Richard T. Cheng Endowment.

References.

[1] VTK, *Visualization Toolkit*. `http://www.vtk.org`.

[2] CGAL, *Computational Geometry Algorithms Library*. `http://www.cgal.org`, v4.0.

[3] ADRIAN BOWYER, *Computing Dirichlet tesselations*, Computer Journal, 24 (1981), pp. 162–166.

[4] ANDREY N. CHERNIKOV AND NIKOS P. CHRISOCHOIDES, *Generalized Delaunay Mesh Refinement: From Scalar to Parallel*, in Proceedings of the 15thInternational Meshing Roundtable, Birmingham, AL, September 2006, Elsevier.

[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] ——, *Three-Dimensional Delaunay Refinement for Multi-Core Processors*, in Proceedings of the 22nd ACM International Conference on Supercomputing, Island of Kos, Greece, June 2008.

[7] ——, *Generalized Two-Dimensional Delaunay Mesh Refinement*, SIAM Journal on Scientific Computing, 31 (2009), pp. 3387–3403.

[8] L. Paul Chew, *Guaranteed-quality triangular meshes*, Tech. Report TR89983, Cornell University, Computer Science Department, 1989.

[9] ——, *Guaranteed quality mesh generation for curved surfaces*, in Proceedings of the 9th ACM Symposium on Computational Geometry, San Diego, CA, 1993, pp. 274–280.

[10] ——, *Guaranteed-quality Delaunay meshing in 3D*, in Proceedings of the 13th ACM Symposium on Computational Geometry, Nice, France, 1997, pp. 391–393.

[11] Hale Erten and Alper Üngör, *Triangulations with locally optimal steiner points*, in SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing, Aire-la-Ville, Switzerland, Switzerland, 2007, Eurographics Association, pp. 143–152.

[12] Panagiotis Foteinos, Andrey Chernikov, and Nikos Chrisochoides, *Fully generalized 2d constrained delaunay mesh refinement*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2659–2686.

[13] Lori A. Freitag and Carl Ollivier-Gooch, *A Cost/Benefit Analysis of Simplicial Mesh Improvement Techniques as Measured by Solution Efficiency*, Preprint ANL/MCS-P722-0598, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill, 10 (2000).

[14] Paul-Louis George and Houman Borouchaki, *Delaunay Triangulation and Meshing. Application to Finite Elements*, HERMES, 1998.

[15] Benoît Hudson, *Safe Steiner Points for Delaunay Refinement*, in International Meshing Rountable Research Notes, 2008.

[16] Xiang-Yang Li, *Generating well-shaped d-dimensional Delaunay meshes*, Theoretical Computer Science, 296 (2003), pp. 145–165.

[17] 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.

[18] Gary L. Miller, Steven E. Pav, and Noel Walkington, *When and Why Ruppert's Algorithm Works*, in IMR, 2003, pp. 91–102.

[19] 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.

[20] Alexander Rand, *Where and how chew's second delaunay refinement algorithm works*, in Proceedings of the 23rd Canadian Conference on Computational Geometry, August 2011, pp. 157–162.

[21] Jim Ruppert, *A Delaunay refinement algorithm for quality 2-dimensional mesh generation*, Journal of Algorithms, 18(3) (1995), pp. 548–585.

[22] Jonathan Richard Shewchuk, *Constrained Delaunay tetrahedralizations and provably good boundary recovery*, in Eleventh International Meshing Roundtable, Ithaca, NY, Sept. 2002, pp. 193–204.

[23] ——, *Delaunay refinement algorithms for triangular mesh generation*, Computational Geometry: Theory and Applications, 22 (2002), pp. 21–74.

[24] ——, *What is a Good Linear Element? - Interpolation, Conditioning, and Quality Measures*, in Proceedings of the 11th International Meshing Roundtable, Sandia National Laboratories, September 2002, pp. 115–126.

[25] 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.

[26] David F. Watson, *Computing the n-dimensional Delaunay tesselation with application to Voronoi polytopes*, Computer Journal, 24 (1981), pp. 167–172.