Beating the Spread: Time-Optimal Point Meshing*

Gary L. Miller[†] Todd Phillips[‡] Donald R. Sheehy[§]

June 6, 2011

Abstract

We present NetMesh, a new algorithm that produces a conforming Delaunay mesh for point sets in any fixed dimension with guaranteed optimal mesh size and quality. Our comparison based algorithm runs in time $O(n \log n + m)$, where n is the input size and m is the output size, and with constants depending only on the dimension and the desired element quality bounds. It can terminate early in $O(n \log n)$ time returning a $O(n)$ size Voronoi diagram of a superset of P with a relaxed quality bound, which again matches the known lower bounds.

The previous best results in the comparison model depended on the log of the **spread** of the input, the ratio of the largest to smallest pairwise distance among input points. We reduce this dependence to $O(\log n)$ by using a sequence of ϵ-nets to determine input insertion order in an incremental Voronoi diagram. We generate a hierarchy of well-spaced meshes and use these to show that the complexity of the Voronoi diagram stays linear in the number of points throughout the construction.

*Partially supported by the National Science Foundation under grant number CCF-0635257.
[†]Department of Computer Science, Carnegie Mellon University, glmiller@cs.cmu.edu
[‡]Department of Computer Science, Carnegie Mellon University, tp517@cs.cmu.edu
[§]Department of Computer Science, Carnegie Mellon University, dsheehy@cs.cmu.edu

1 Introduction

In this paper we present a new algorithm for meshing point sets in fixed dimension. This is the first algorithm we know of that is work-optimal in the comparison-based model in the sense of [Yao81]. Known work-efficient algorithms for meshing are one of two types. The first of these are based on incremental refinement of the Voronoi diagram or Delaunay triangulation. The only work-efficient of these in higher dimension performs a recursive Voronoi refinement where at all times a "quality" Voronoi mesh is maintained. Unfortunately, this leads to work of $O(n \log \Delta + m)$ where n is the input size, m is the output size and Δ is the **spread**, the ratio between the diameter of the bounding box and the distance between the closest pair of features [HMP06, HMP07]. The second type uses a quadtree to generate a mesh. Work-efficient versions use bit manipulation of the coordinates of the points to efficiently help with the point location [MV00, BET99, HPÜ05]. These algorithms are not optimal in the comparison model and possibly more importantly, it is not known how to efficiently handle higher dimensional features (segments, facets) with these methods.

Our algorithm uses range space ϵ-nets to determine the insertion order of the input points to improve the work bound for point sets with large spread. Clarkson used a similar method for doing point location in a Voronoi diagram [Cla88]. In our approach, since we also add some Steiner points, we can guarantee that the total size of the intermediate Voronoi diagrams are only linear size. This insertion order requires us to maintain a Voronoi diagram that need not have good aspect ratio in the usual sense.

Our algorithm will generate a linear-size mesh in fixed constant dimensions. In their 1994 paper Bern, Eppstein, and Gilbert showed how to generate such a linear-size mesh with no large angles [BEG94]. In a later paper we gave a Voronoi refinement algorithm that also generates linear size meshes, [MPS08] but had a running time of only $O(n \log \Delta)$.

Because a standard good aspect ratio mesh is too large, we maintain a weaker but sufficient condition, bounded ply. Throughout the life of the algorithm we maintain a mesh that is of bounded ply which will be used to bound the point location work and the work to determine the insertion order:

Definition 1. *A Voronoi Diagram of a domain Ω is k-**ply** if for every point $x \in \Omega$ at most k circumballs of the Delaunay simplices contain x in their interior.*

Using the bounded-ply property we can afford to maintain a copy of each uninserted point in each Delaunay ball that contains it. We pick an insertion ordering so that the number of uninserted points stored in a Delaunay ball decreases geometrically, which we achieve using ϵ-nets.

Let P be the input points and M be points that have been inserted into the mesh so far including the Steiner points. We say that M is an ϵ-net for P if any ball whose interior is disjoint from M contains at most ϵn points from P. We show, given a mesh M that is an ϵ-net, how to pick at most a constant number of points per Delaunay ball so that after their insertion the new mesh will be a $\epsilon/2$-net. Thus, a **round** consists of adding these new input points plus a constant factor more Steiner points so that we recover a bounded-ply mesh. After $O(\log n)$ rounds the process terminates with a constant ply mesh of size $O(n)$. This output can then be finished to a standard good aspect ratio mesh in output sensitive $O(m)$ time if desired.

In this paper we only consider the case of point set inputs. We feel that the methods proposed should readily be applicable to inputs with higher-dimensional features, such as edges and faces, and with optimal runtime.

2 Beating the Spread

The spread of a point set is the ratio of the largest to smallest interpoint distances, and is denoted as Δ. It is a (geo)metric rather than a combinatorial property; given a set of points P, its cardinality may be n but its spread is not in general bounded by any function of n. It is not uncommon to see a dependence on the spread in the analysis of algorithms in computational geometry and finite metric spaces. Though rarely a problem in practice, it does thwart the most basic principle in the analysis of algorithms, to bound the complexity in terms of the input size.[1]

Consider two classic data structures, the quadtree and the kd-tree. The quadtree partitions space geometrically, breaking squares into 4 pieces of equal **size**. The kd-tree partitions the input points combinatorially into sets of equal **cardinality**. These data structures demonstrate the difference between geometric and combinatorial divide and conquer. The quadtree has depth $\log \Delta$ whereas the kd-tree has depth $\log n$. Unfortunately, many computational problems from nearest neighbor search to network design problems depend on (geo)metric information that is lost when doing a combinatorial divide and conquer. Thus, for many problems, the best known algorithms depend on the spread in either time or space complexity or both.

One approach to dealing with the spread is to restrict the computational model. If coordinates are restricted to be $\log n$-bit integers then the spread is $O(n)$. If we use floating point numbers, the spread is $O(2^n)$. These assumptions about the bit representation of the input also allow for fast computation of logarithms as well as the floor and ceiling functions. These computations are usually omitted from the basic operations of the real RAM model often used in computational geometry to extend the comparison sorting model from the real line to d-dimensional Euclidean space. In their work on metric nets, Har-Peled and Mendel correctly argue that if one can do arithmetic in constant time, it is natural to expect also to perform other operations of size $O(\log \log \Delta)$ in constant time [HPM06]. This is certainly the case for many practical implementations of geometric algorithms. However, it is interesting, both in theory and in practice to explore ways of eliminating the dependence on the spread without resorting to specialized bit operations–in theory because it probes the limits of an important computational model and in practice because it allows one to work with a minimal set of primitives with minimal assumptions about the low-level data representation.[2]

In mesh generation, a dependence on the spread creeps in from two different sources, in the output size and the in the cost of point location. The previous state of the art in comparison based point meshing requires $O(n \log \Delta + m)$ work, where the first term is the cost of point location and the second is the output sensitive term. Even for point set inputs, the lower bounds on quality meshes imply that m may also depend on the spread. Thus, to contest with the spread, we must both optimize point location and also relax the quality condition. This is why our algorithm has two phases, one that produces a linear size Voronoi diagram of a superset in $O(n \log n)$ time and one that refines that mesh to quality in $O(m)$ time.

3 Voronoi Refinement Basics

Voronoi Diagrams. The Voronoi diagram of a finite point set P in $\mathbb{R}^d$, denoted Vor_P, is the polyhedral complex decomposing $\mathbb{R}^d$ into regions based on the nearest neighbor among the points of P. The regions are called Voronoi cells. Since some cells are unbounded, we assume there is a

[1] One can get around this by making assumptions about the bit representations of the inputs. We will address this as well.

[2] Recall that in the popular CGAL library, all primitives are implemented for several different kernels, all use a small, unified interface.

suitably sized bounding ball around the points of P and for a point $p \in P$ we write $\mathrm{Vor}_P(p)$ to denote the intersection of the Voronoi cell with this ball.

The **in-radius** of $\mathrm{Vor}_P(p)$, denoted r_p, is the radius of the largest ball centered at p that is contained in $\mathrm{Vor}_P(p)$. The **out-radius**, R_p is the radius of the smallest ball centered at p that contains $\mathrm{Vor}_P(p)$. The **aspect ratio** of $\mathrm{Vor}_P(p)$ is $\frac{R_p}{r_p}$. When viewed as cell complex, the vertices or 0-faces of $\mathrm{Vor}_P(p)$ are called the **corners** to avoid confusion with the notion of a mesh vertex introduced later. The out-radius is therefore, the distance from p to its farthest corner, while the in-radius is the distance from p to its nearest facet. A set of points is **τ-well-spaced** if every Voronoi cell has aspect ratio at most τ.

The Voronoi diagram is the dual of the Delaunay triangulation, which has a simplex for every subset of points on the boundary of a ball that contains no other points of P. For full-dimensional Delaunay simplices, these circumscribing balls are called **D-balls**. The corners of the Voronoi cells correspond to D-balls[3].

Voronoi Refinement. The goal of Voronoi refinement is to produce a τ-well-spaced set M by adding new vertices called Steiner points to an input set P that is not well-spaced. We want to add as few vertices as possible. The algorithm is simple: starting with Vor_P, iteratively add the farthest corner of any cell with aspect ratio greater than τ. It is perhaps more commonly known in its dual formulation, as Delaunay refinement, where the goal is to improve the Delaunay simplices rather than the Voronoi cells. But the resulting algorithms and their analysis are nearly identical for both the primal and the dual formulation.

It is not immediately obvious that Voronoi refinement should ever terminate. Indeed, for some τ, it will run forever. For a reasonable choice of τ, say $\tau = 3$ for example, not only will the algorithm terminate, it will do so with asymptotically optimal size, both in the number of points added and the total number of faces in the diagram. This latter property results from the aspect ratio condition, and is a major motivation for doing the refinement in the first place.

Sparse Voronoi Refinement. The first obstacle to producing a refined Voronoi diagram in optimal time and space is that the input may have a large Voronoi diagram, $\Omega(n^{\lceil d/2 \rceil})$ faces in the worst case. To overcome this obstacle, the Sparse Voronoi Refinement (SVR) algorithm of Hudson et al. [HMP06] interleaves the addition of input points to the diagram with the addition of Steiner points. In doing so, the algorithm requires two extra pieces. First, input points are only added if they are "close" to the current Voronoi diagram. Second, the Steiner points may not be added "too close" to uninserted input points. The former notion of closeness is what we call **ε-medial**, the ratio of the distances to the nearest and second nearest points must be at most ε. The latter notion of closeness causes the algorithm to **yield** by adding an input point p rather than a Steiner point v if the distance from p to v is less than γ times the radius of the empty ball around v.

By only inserting ε-medial points and yielding when appropriate, Sparse Voronoi refinement maintains a good aspect ratio Voronoi diagram at every stage of the algorithm. Consequently, the total work is output sensitive. This approach has also been generalized to more complex inputs than just point sets, considering also piecewise linear complexes [HMP06].

Point Location, Point Location, Point Location. The bottleneck for the running time of Voronoi refinement is point location. Recall, that in the standard incremental Voronoi (or Delaunay) algorithm, the first step to inserting a new point is to find that point in the current diagram. A

[3] We permit corners on the boundary to represent "degenerate" D-balls, ones corresponding to simplices that are not full-dimensional.

natural and highly effective technique for doing this point location is to eagerly store the uninserted points in the D-balls of each Voronoi diagram as the algorithm progresses. Points are moved whenever an insertion changes a D-ball locally.

In SVR, this approach corresponds to a geometric divide and conquer, similar in spirit to quadtree methods, because after a constant number of moves, the size (radius) of the balls containing any point goes down by a constant factor. Thus, in SVR a single input point may be moved $\Theta(\log \Delta)$ times. In this work, we show how to modify the algorithm so that only $O(\log n)$ moves are necessary. One way to view these results is as a way to achieve similar properties to compressed quadtrees without leaving the comparison model or privileging any fixed set of coordinate axes.

4 Definitions and Notations

Points, Vectors, and Distances. We will treat points of d-dimensional Euclidean space as vectors in $\mathbb{R}^d$. As such, we denote the euclidean distance between two points $x, y \in \mathbb{R}^d$ as $|x - y|$. Moreover, we allow the usual operations of scalar multiplication and addition on points and also on sets of points. So, for example, if S is the unit sphere centered at the origin, c is any point, and r is a non-negative real number, then $rS + c$ is the sphere of radius r centered at c. We will also define the distance from a point x to a set S as $\mathbf{d}(x, S) = \inf_{y \in S} |x - y|$. We write $\text{ball}(c, r)$ to denote the open ball of radius r centered at c and $\text{conv}(X)$ to denote the convex closure of $X \subset \mathbb{R}^d$.

Domains. A **domain** $\Omega \subset \mathbb{R}^d$ is defined by a center c_Ω, a radius r_Ω, and a collection of disjoint open balls $B_1, \ldots, B_k \subset B_\Omega = \text{ball}(c_\Omega, r_\Omega)$ such that

$$\Omega = B_\Omega \setminus \left(\bigcup_{i=1}^{k} B_i \right).$$

The ball B_Ω is called the **bounding ball** of Ω and $S_\Omega = \{x \in \mathbb{R}^d : |x - c_\Omega| = r_\Omega\}$ is the **bounding sphere** of Ω.

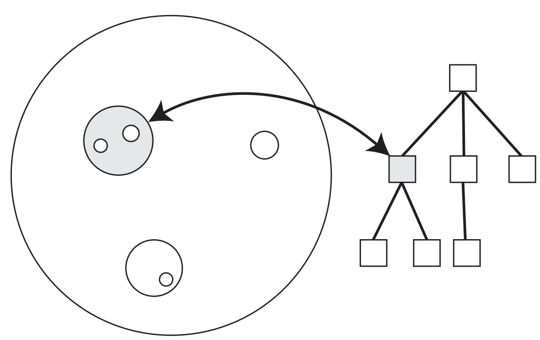

Figure 1: A domain hierarchy as a collection of sets (left) and its tree structure (right).

We get a hierarchy of domains if the balls removed from B_Ω are the bounding balls of other domains. Formally, a **domain hierarchy** is a tree H with disjoint domains as nodes rooted at Ω_{root} such that

1. for any pair $\Omega, \Omega' \in H$, $\text{p}(\Omega') = \Omega$ if and only if $S_{\Omega'} \subset \Omega$, and
2. $\displaystyle\bigcup_{\Omega \in H} \Omega = B_{\Omega_{\text{root}}}$.

Here, $\text{p}(\Omega)$ denotes the parent of Ω in H.

Cages. Given a domain Ω, we want to add vertices near S_Ω to limit the interaction between the inside and the outside of Ω. We will have two parameters, δ determining the density of these points, and γ determining how nearly cospherical they are. We call such a set C_Ω of vertices a **cage** and we require the following three properties, where $\overline{r} = (1-\delta-\gamma)r_\Omega$ and $\overline{S} = (1-\delta-\gamma)S_\Omega$.

1. [**Nearness Property**] For all $v \in C_\Omega$, $\mathbf{d}(v, \overline{S}) \leq \gamma\overline{r}$.
2. [**Covering Property**] For all $x \in \overline{S}$, $\mathbf{d}(x, C_\Omega) \leq (\delta+\gamma)\overline{r}$.
3. [**Packing Property**] For all distinct $u, v \in C_\Omega$, $|u-v| \geq (\delta - 2\gamma)\overline{r}$.

To construct such a set of points, we start with a **cage template** T of points on the unit sphere S. The points of T are a metric space δ-net on S (not to be confused with the range space nets used elsewhere in this paper). That is, for all $x \in S$, $\mathbf{d}(x, T) \leq \delta$ and for each distinct pair $u, v \in T$, $|u - v| \geq \delta$. Such sets are known to exist and can be constructed using a simple greedy algorithm [Gon85, Mat02].

For a domain Ω we construct its cage by adding for each $x \in c_\Omega + \overline{r}T$, a new point x' such that $|x - x'| \leq \gamma\overline{r}$. It is easy to check that this set of points will satisfy the three properties of a cage.

Definition 2. *A cage C_Ω centered at c with radius r is ε-**encroached** or simply **encroached** by a point $p \notin C_\Omega$ if either*

1. *p is an input point in* $\text{annulus}(c, \varepsilon r, r)$, *(**inner-encroachment**), or*
2. *p is an input or Steiner point in* $\text{annulus}(c, r, \frac{2r}{\varepsilon})$, *(**outer-encroachment**).*

Roughly speaking, non-encroached cages have room on the inside (w.r.t. input points) and room on the outside (w.r.t. all mesh vertices).

Hierarchical Meshes. A **mesh** is a set of points M and its Voronoi diagram. The points of M are called the **vertices** of the mesh.

Definition 3. *A **hierarchical mesh** is a mesh M along with a domain hierarchy H_M such that:*

1. *M has a vertex at the center of every domain, i.e. $c_\Omega \in M$ for all $\Omega \in H_M$*
2. *No domain is ε-encroached.*

Given a hierarchical mesh M and $\Omega \in H_M$, we define M_Ω to be the points of M contained in Ω plus the centers of the children of Ω in H_M. Formally,

$$M_\Omega = (M \cap \Omega) \cup \bigcup_{\Omega' \in \text{children}(\Omega)} \{c_{\Omega'}\}.$$

We call this the set M restricted to the domain Ω, and it is well defined for any domain Ω and any set M that contains the centers of the children of Ω. In particular, for a subset $P \subset M$, $P_\Omega = P \cap M_\Omega$.

In a hierarchical mesh, we can also define the Voronoi cell of a cage C_Ω as

$$\text{Vor}_M(C_\Omega) = \bigcup_{u \in M \cap B_\Omega} \text{Vor}_M(u).$$

Definition 4. *We say that a hierarchical mesh M is τ-**quality** if the following conditions are met:*

1. *For every non-cage vertex $v \in M$,* $\text{Vor}_M(v)$ *has aspect ratio at most τ.*
2. *For every $\Omega \in H_M$,* $\text{Vor}_M(C_\Omega)$ *has aspect ratio at most τ.*

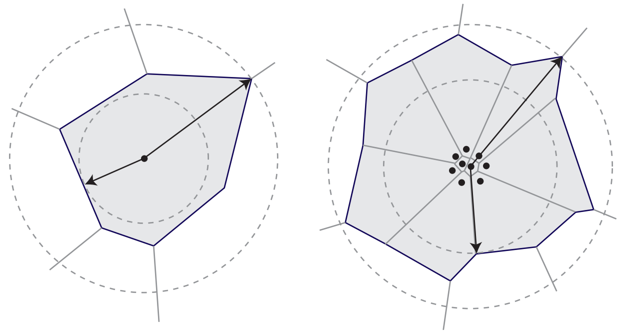

Figure 2: Quality cells of a vertex (left) and a cage (right).

3. No domain in H_M is ε-encroached.

The four constants γ, δ, ε, and τ are called the **meshing parameters**. Throughout, they are assumed to be fixed constants independent of the dimension.

Definition 5. *For a set S and a domain Ω, the* ***feature size*** *is a function $\mathbf{f}_S^\Omega : \mathbb{R}^d \to \mathbb{R}$ that maps a point x to the distance to its second nearest neighbor among the points of S_Ω.*

We are mainly interested in the feature size of the input and of the mesh, $\mathbf{f}_P^\Omega$ and $\mathbf{f}_M^\Omega$ respectively, over the domains of M_H.

5 Additively-Weighted Voronoi Diagrams

There is a natural generalization of Voronoi diagrams in which the points are permitted to have weights that affect the distance additively. For a point v, let r_v be the weight of v. The distance between two weighted points is defined as

$$\mathbf{d}(u, v) = |u - v| - r_u - r_v.$$

An equivalent formulation simply measures the distance between the spheres of radius r_u and r_v centered at u and v respectively. For a point set M, we define the **additively-weighted Voronoi cell** of a point $v \in M$ to be

$$\text{Vor}(v) = \{x \in \mathbb{R}^d : \min_{u \in M} \mathbf{d}(u, x) = \mathbf{d}(v, x)\}.$$

The **additively-weighted Voronoi diagram** is the cell complex decomposing $\mathbb{R}^d$ obtained by taking all of the additively-weighted Voronoi cells. This definition generalizes the standard Voronoi diagram, which may be viewed as an additively-weighted Voronoi diagram for points of 0 weight.

The additively-weighted Voronoi diagram is different from the more common notion of weighted Voronoi diagrams obtained by replacing the distance function with the power distance. The cells of the additively-weighted diagram are not necessarily polyhedra nor are they necessarily convex. Still, it is possible to extend basic properties of Voronoi diagrams to the case of additive weights. The **in-radius** of a Voronoi cell $\text{Vor}(v)$ is defined as

$$\textbf{in-radius}(\text{Vor}(v)) := \max\{r : \text{ball}(v, r) \subset \text{Vor}(v)\},$$

and similarly, the **out-radius** is defined as

$$\textbf{out-radius}(\text{Vor}(v)) := \min\{r : \text{Vor}(v) \subset \text{ball}(v, r)\}.$$

We say that u and v are neighbors if $\mathrm{Vor}(u) \cap \mathrm{Vor}(v) \neq \emptyset$. The in-radius of $\mathrm{Vor}(v)$ may also be defined as $\frac{1}{2}\mathbf{d}(u,v) + r_v$, where u is the nearest among the neighbors of v in additive distance. The **aspect ratio** of $\mathrm{Vor}(v)$ is the ratio of the out-radius to the in-radius.

Approximation by Cages. The additively-weighted Voronoi diagram can be approximated by a regular Voronoi diagram by replacing the weighted points with a small cage of new vertices at distance r_v from each weighted point v. The approximate cells are the union of the Voronoi cells of the cage vertices. These approximate cells can also be used to get a good approximation of the in-radius and out-radius of the weighted Voronoi cell. Let v be a vertex with cage vertices C. The neighbors of C are those vertices v that share a Voronoi face with a vertex in C but are not in $C \cup \{v\}$. The in-radius of the approximate Voronoi cell is $\frac{1}{2}\mathbf{d}(u,v) + r_v$, where u is the nearest among the neighbors of C in additive distance. Since the neighbors necessarily have weight 0, this reduces to $\frac{|v-u|+r_v}{2}$.

For unweighted points M, the feature size function $\mathbf{f}_M : \mathbb{R}^d \to \mathbb{R}$ is the distance to the second nearest point of M. So, for points v in M, $\mathbf{f}_M(v)$ is the distance to the nearest neighbor of v in $M \setminus \{v\}$. Thus, in the absence of weights, the in-radius of $\mathrm{Vor}(v)$ is $\frac{1}{2}\mathbf{f}_M(v)$ and if the aspect ratio is τ and the out-radius is R then $\mathbf{f}_M(v) = \frac{2R}{\tau}$. If the points have weights then the definition of $\mathbf{f}_M$ is the same as if the points have no weights. The following lemma shows how this feature size relates to the out-radius and aspect ratio of the weighted Voronoi cells.

Lemma 1. *Let v be a weighted point among a set M. Let r be the in-radius of* $\mathrm{Vor}(v)$. *If $r_a \leq \varepsilon(|a-b| - r_b)$ for all $a, b \in M$, then*

$$\frac{2r(1-\varepsilon)}{1+\varepsilon} \leq \mathbf{f}_M(v) \leq \frac{2r}{1-\varepsilon}$$

Proof. Let u and w be the nearest points to v in Euclidean and weighted distance respectively (it could be that $u = w$). So, $2r = |v-w| + r_v - r_w$ and $\mathbf{f}_M(v) = |u-v|$. By assumption, r_v and r_w are both less than $\varepsilon|v-w|$. So, it follows that

$$(1-\varepsilon)|v-w| \leq |v-w| + r_v - r_w \leq (1+\varepsilon)|v-w|. \tag{1}$$

This assumption and the definitions of u and w also imply that

$$(1-\varepsilon)|v-w| \leq |u-v| \leq |v-w|. \tag{2}$$

So, the result follows from (1) and (2). □

When we choose sufficiently dense cages, nearly the same bounds apply for the approximate weighted Voronoi cells:

Lemma 2. *Let v be a vertex or a cage in a hierarchical mesh M. Let r be the in-radius of* $\mathrm{Vor}(v)$ *and let c be the center of v. If no cages are ε-encroached for ε sufficiently small, then*

$$r \leq \mathbf{f}_M^{\Omega}(c) \leq 3r,$$

where Ω is the domain containing the boundary of $\mathrm{Vor}(v)$.

6 The Algorithm

6.1 Overview of the Algorithm

Like Sparse Voronoi Refinement, the core of the NetMesh algorithm is an incremental construction of a Voronoi diagram with the refinement steps to maintain mesh quality. There are five main concerns. The algorithm must **(1) order the input points**. These points are added one at a time in an **(2) incremental construction**. After each insertion, Steiner points are added in a **(3) refinement** phase that recovers the quality invariant. All the while, uninserted points are organized in a **(4) point location** data structure. Once all of the inputs have been added, an optional **(5) finishing** procedure turns the linear-size hierarchical mesh into a standard well-spaced mesh. Each of these concerns will be addressed in more detail below, but first we will describe the main ideas used and how they fit together.

Point Location. The point location data structure associates each point with each D-ball that contains it. So, it is easy to report the set of D-balls containing an input point and similarly, to report the set of points in a D-ball. These associations are updated locally every time a new point changes the underlying Delaunay triangulation. We will prove that no point is ever in more than a constant number of D-balls and thus the size of this structure will not exceed $O(n)$.

Incremental updates. In Sparse Voronoi Refinement, every insertion is medial. This is critical to maintain quality in the mesh throughout the algorithm. In the NetMesh algorithm, we change the domain hierarchy before inserting each point to guarantee that it is medial in whatever domain contains it. We show that this is sufficient to get the same guarantees as in SVR. Thus, we can insert the points in any order.

Ordering the input with ϵ-nets. The theory of range space ϵ-nets is used to choose the input insertion order. One round of the algorithm consists of the union of a collection of ϵ-nets for the input points for each D-ball, where the ranges are open balls. It is known that such sets exist, are small, and can be found quickly and deterministically [Cha00]. The points in any round may be inserted in any order, after which, the next round is computed. In each round, the maximum number of points stored in any D-ball goes down by a constant factor, so the total number of rounds is $O(\log n)$.

Refinement. The refinement, or cleaning phase of the algorithm is a standard Voronoi refinement in that it adds Steiner points at the farthest corner of any cell with bad aspect ratio. As in SVR, if the Steiner point is sufficiently close to an uninserted input point p, then p is added instead. One slight change is that we maintain the aspect ratio of the Voronoi cells of cages, but do not require the cage vertices themselves to have good aspect ratio Voronoi cells.

Finishing the mesh. The algorithm produces a quality hierarchical mesh of linear size. If one wants to extend this mesh to a standard well-spaced mesh, it is a straightforward procedure to do this in $O(m)$ time, where m is the number of vertices in an optimal-size, well-spaced superset of P. This finishing process can run quickly because it need not do any point location (all of the input points have already been inserted).

6.2 Point Location Operations

Each uninserted input point stores a list of D-balls that contain it as well as a list of cages that it encroaches. Similarly, the D-balls have lists of uninserted vertices that they contain. With each change in the Voronoi diagram, these lists are updated. We say that the points are "stored *in* the balls" to simplify the description of this list upkeep. A point will generally be contained in several D-balls. The uninserted points are moved out of D-balls that have been destroyed and into newly created D-balls. This shuffling of points between D-balls is the work of point location. A point is touched in this process if it is moved into a new ball or even if it is considered for moving into a new ball. We count the point location work from the perspective of the uninserted input points.

There are four main point location operations needed.

1. Find the D-balls containing a point to insert it into the Voronoi diagram.

2. Find the nearest and second nearest neighbor of a point in its domain in order to compute its mediality.
3. Find any cages encroached by a given point.
4. Find a nearby input point to yield to, when inserting a Steiner point.

The first operation is trivial.

For cage vertices v in a domain Ω, let $center(v)$ be the vertex at the center of Ω. Let $B(x)$ be the set of D-balls containing x. Let $V(B)$ be the $d+1$ vertices of the Delaunay simplex corresponding to the D-ball B. Let $U(x) = \{V(B) : B \in B(x)\}$. If Ω is the domain containing x, then the nearest and second nearest neighbors of x in M_Ω are in $U(x)$ or $\{center(v) : v \in U(x)\}$, so it is easy to identify them. Call these vertices n_x and s_x respectively. Thus, $\textsc{Mediality}(x) = \frac{|x-n_x|}{|x-s_x|}$ can be computed in time $O(|B(x)|)$.

To check encroachment of input points is easy because this information is stored with the points. At the time a cage is created, any encroaching input points must be relocated, so the encroachment is discovered at that time. To check encroachment of Steiner points, it suffices to observe that if a Steiner point x encroaches a cage C, then some vertex of c must appear in $U(x)$. So, there are only $O(|U(x)|)$ cages to check and each check takes constant time.

To find a point to yield to, we simply need to check for input points in a small empty ball around the proposed input point. This is trivial for Steiner points added during refinement because the Steiner point is the center of a D-ball B and thus we only need to check $\textsc{Uninserted}(B)$. For cage vertices v, the search requires us also to check the points in $\{\textsc{Uninserted}(B) : B \in B(v)\}$. In both cases, the points checked in this process also need to be checked for relocation when the new vertex is inserted. Thus, the cost for this search is dominated by the cost of relocating points, which we analyze in detail later.

6.3 Incremental Updates to Hierarchical Meshes

The basic operation in incremental Voronoi diagrams is $\textsc{Insert}(v)$, which adds the vertex v to the Voronoi diagram and updates the point location data structures. To keep this operation constant time (not counting the cost of point location), we must guarantee that $|B(v)|$ is a constant because every D-ball in $B(v)$ is destroyed by the insertion. This is done by making sure that every new insertion is medial. Before the new point is inserted, we update the domain hierarchy. If the point was not medial, then it must be significantly closer to its nearest neighbor than it second nearest neighbor, and thus we add or expand cage around the nearest neighbor. We must also update the domain hierarchy of the new point encroaches on an existing cage.

$\textsc{Insert}(x)$
 for each C in $\textsc{OutEncroach}(x)$: $\textsc{ReleaseCage}(C)$
 Add x to Vor_M and update the point location structure.

$\textsc{YieldingInsert}(x)$
 let v be the nearest neighbor of x in the current mesh.
 if there is an input point p in $\mathrm{ball}(x, \gamma|x-v|)$
  then $\textsc{Insert}$(p) **else** $\textsc{Insert}$(x)

There are three basic cage operations:

NewCage(p, r)
  Initialize a new cage.
  for each $x \in T$, YieldingInsert($rx + p$).

ReleaseCage(C)
  for each cage vertex v in C push v to the RefineList.
  Delete the cage C

GrowCage(C)
  Let x be the center of C and let r be its radius.
  if in-radius(Vor(C)) $\geq \frac{r}{\varepsilon^2}$ **then** NewCage($x, \frac{r}{\varepsilon}$).
  ReleaseCage(C).

Equipped with the cage operations, we define the following routine. Its purpose is to rearrange the domain hierarchy by creating or growing new cages so that a new vertex v can be added to a domain in which it is medial.

InsertInput(v)
  let u be the nearest neighbor of v in M_Ω
  if Mediality(v) $\leq \varepsilon$ **then** NewCage($u, |u - p|/\varepsilon$)
  for each C in InEncroach(v): GrowCage(C)
  Insert(v)

6.4 Refinement

The algorithm maintains a list of cells with bad aspect ratio called RefineList. The cleaning procedure goes through this list and refines these cells until none are left. The RefineList is updated every time a Voronoi cell changes. The structure of the Voronoi diagram makes it easy to check the aspect ratio of a cell and Theorem 4 implies that this can be done in constant time. If a cell's aspect ratio was good but goes bad, it is added to the list. If its aspect ratio was bad but becomes good, it is removed from the list.

Clean(M: Mesh)
  while RefineList is not empty
    let $v \in$ RefineList
    let x be the far corner of Vor(v)
    YieldingInsert(x)

6.5 Input Ordering with ϵ-Nets

We employ the theory of range space ϵ-nets to order the inputs for insertion. The following is a special case of Theorem 4.6 from [Cha00] when the range space is defined by open balls.

Theorem 3. *Let $P \subset \mathbb{R}^d$ be a set of n points and let $\epsilon \in (0, 1)$. There exists an algorithm* Net(ϵ, P) *that runs in $O(\frac{1}{\epsilon^2}(\log \frac{1}{\epsilon})^{d+1} n)$ time and returns a subset $N \subseteq P$ such that $|N| = O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ and any open ball that contains ϵn points of P also contains a point of N. In particular, for constant ϵ the running time is linear and the net is of constant size.*

Using the NET algorithm as a black box, we select the next round of points to insert as follows.

SELECTROUND(M: mesh)
    $N \leftarrow \emptyset$
    for each $B \in$ DBALLS(M)
        $N \leftarrow N \cup$ NET($\frac{1}{2d}$, UNINSERTED(B))
    return N

If the maximum number of uninserted points in a D-ball of some mesh is k, then after adding the points chosen by SELECTROUND, this maximum is at most $\frac{k}{2}$. This follows from the fact that every new D-ball is covered by at most d of the old D-balls (see Theorem 10). So, the total number of rounds is at most $\lceil \log n \rceil$. We can now give the main loop of the algorithm.

NETMESH(P: points)
    Initialize an empty mesh M
    UNINSERTED $\leftarrow P$
    let c, r be such that $P \in$ ball(c, r)
    OUTERCAGE = NEWCAGE($c, \frac{r}{\varepsilon}$)
    while UNINSERTED is not empty
        V = SELECTROUND(M)
        for each $v \in V$
            INSERTINPUT(v)
            CLEAN(M)
    return M

6.6 Finishing the Mesh

The output of NETMESH is a quality hierarchical mesh. If the desired output is a well-spaced mesh according to the traditional definition, i.e. quality with a single domain, then some finishing procedure is required. Fortunately, it is trivial given the cage operations defined above:

FINISHMESH(M: Mesh)
    while there exists a cage C other than OUTERCAGE
        GROWCAGE(C)
        CLEAN(M)

Since the cages are not encroached, they have some space around them. The FINISHMESH procedure simply grows the cages until this space is filled. No new cages are formed and no point location work on input points is required.

Note that finishing the hierarchical mesh in this way may result in a mesh with more than a linear number of points because well-spaced meshes are subject to potentially superlinear (or even superpolynomial!) lowerbounds. This is why we consider the finishing operation to be optional.

7 Overview of the Analysis

An intermediate mesh, M_i, is the mesh after i vertices or cages have been inserted in during the incremental construction. To analyze the NETMESH algorithm, we will prove that two invariants are maintained for each intermediate mesh: **the feature size invariant** and **the quality invariant**.

Definition 6. *A hierarchical mesh M satisfies the* ***quality invariant*** *if each intermediate mesh M_i is τ'-quality for some constant τ' depending only on the meshing parameters.*

Definition 7. *A hierarchical mesh M of an input set P satisfies the* ***feature size invariant*** *if for all domains $\Omega \in H_M$ and all vertices $v \in M_\Omega$*

$$\mathbf{f}_P^\Omega(v) \leq K f_M^\Omega(v),$$

where K is a constant that depend only on the mesh parameters.

The quality invariant is useful because of several properties of quality meshes.

Theorem 4. *If M is a τ-quality mesh, then*

1. *no point of $\mathbb{R}^d$ is contained in more than $O(1)$ D-balls,*
2. *no D-ball intersects more than $O(1)$ other D-balls, and*
3. *no vertex of M has more than $O(1)$ Delaunay neighbors.*

These structural results about quality meshes are known for the case of a single domain [MTTW99, HMP06]. To extend them to the case of a quality hierarchical meshes follows the same methods as in previous work. The three conclusions are proven in Theorems 16, 19, and 17 respectively.

Over a single domain, standard results in mesh size analysis imply that the feature size invariant suffices to prove that the number of vertices is bounded (up to constants) by the feature size integral:

$$\int_\Omega \frac{dx}{\mathbf{f}_P^\Omega(x)^d}.$$

In previous work [MPS08], we showed that the feature size integral is $O(n)$ when the input points satisfy a certain spacing condition. We prove that in each domain Ω of the hierarchy, the points of P_Ω satisfy this spacing condition (Lemma 8), allowing us to prove that the total output size is $O(n)$ (Theorem 9).

Theorem 4 and the quality invariant imply that the cost to update the Voronoi diagram for a single insertion is constant. That is, the number of combinatorial changes to the Voronoi diagram is constant for each each insertion. Thus, since the total number of points added is $O(n)$, the total work is $O(n)$, not counting the cost of point location.

To bound the cost of point location, we first show that at most a constant number of vertices are added to any D-ball in the course of a round (Lemma 31). This is then used to show that the total amount of point location work is $O(n)$ per round. Since there are only $O(\log n)$ rounds, the total work is $O(n \log n)$ as desired.

Finally, in Section 13, we show that the FinishMeshprocedure runs in $O(m)$ time. This allows us to conclude the following theorem about the overall running time.

Theorem 5. *Given n points $P \subset \mathbb{R}^d$, the* NetMesh *algorithm produces a hierarchical quality mesh of size $O(n)$ in $O(n \log n)$ time. If this is followed by the* FinishMesh *procedure, the output is a well-spaced mesh of size $O(m)$ in $O(n \log n + m)$ time.*

8 Size Bounds

In this section we will show that the output of NetMesh has linear size. The analysis will follow a straightforward strategy. We will argue that the algorithm never inserts a vertex too close to an existing vertex. This is known as the **insertion radius invariant**, and it allows us to prove that the **feature size invariant** holds for all intermediate meshes. We use this to prove that for all domains Ω, M_Ω has size linear in $|P_\Omega|$ from which the overall bound follows. This strategy is not new; it parallels closely the approach of Ruppert [Rup95] for Delaunay refinement and its sparse version introduced by Hudson, Miller, and Phillips [HMP06]. We have adapted it to the case of hierarchical meshes.

We say that a hierarchical mesh M is constructed **incrementally** if the vertices are added one at a time and the domains are adjusted before every insertion so that no domain is encroached. In particular, the algorithm given is such an incremental construction. The intermediate mesh after i points and cages have been added is denoted by M_i, its domain hierarchy H_{M_i} is denoted H_i, and $P_i = P \cap M_i$ is the set of inputs inserted thus far. Define the **insertion radius** of the ith vertex added v as $\lambda_v = \mathbf{f}^{\Omega}_{M_i}(v)$, where $\Omega \in H_i$ is the domain into which v was inserted.

Definition 8. *A hierarchical mesh M of an input set P constructed incrementally satisfies the **insertion radius invariant** if for all domains $\Omega \in H_i$ for all i and all vertices $v \in M_{i\Omega}$*

$$\mathbf{f}^{\Omega}_{P_i}(v) \leq \begin{cases} K'_C \lambda_v & \text{if } v \text{ is inserted as a cage vertex,} \\ K'_S \lambda_v & \text{if } v \text{ is inserted as a circumcenter, and} \\ K'_I \lambda_v & \text{if } v \text{ is inserted as an input vertex} \end{cases}$$

where K'_C,K'_S, and K'_I are constants that depend only on the mesh parameters.

The following lemma states that as long as the insertion radius of every vertex is not too small then the distance to its nearest neighbor is also not too small. Its proof is straightforward and reserved for the appendix.

Lemma 6. *If M is a hierarchical mesh constructed incrementally that satisfies the insertion radius invariant, then M also satisfies the feature size invariant.*

Lemma 6 implies that in order to prove that the spacing of the points in the final mesh is good, it will suffice to show that the algorithm maintains the insertion radius invariant throughout. This is proven in the following lemma.

Lemma 7. *The hierarchical mesh M constructed by the* NetMesh *algorithm satisfies the insertion radius invariant.*

Proof. We proceed by induction on the total number of vertices added. Let v be the ith vertex added and let Ω be the domain it is inserted into.
Case 1: v is a cage vertex. Since P_Ω contains at least the center of Ω, the feature size is bounded as $\mathbf{f}^{\Omega}_{P}(v) \leq r_\Omega$. By construction, adjacent cage vertices are at least αr_Ω apart, where $\alpha = (\delta - 2\gamma)(1 - \delta - \gamma)$. So, $\lambda_v \geq \alpha r_\Omega$. Combining these two facts and choosing $K'_C \geq \frac{1+\varepsilon}{\alpha}$ yields $\mathbf{f}^{\Omega}_{P_i}(v) \leq K'_C \lambda_v$ as desired.
Case 2: v is a clean move. Steiner points are added when some vertex (or cage) $u \in M_{i-1\Omega}$ has aspect ratio greater than τ. Let V_u denote this poor aspect ratio cell. Let w be the nearest neighbor of u in M_Ω, so $\mathbf{f}^{\Omega}_{M}(u) = |u - w|$. In case we yielded in order to insert v, let v' be the true circumcenter that we tried to insert. The yielding condition guarantees that

$$|v - v'| \leq \gamma |u - v'|. \tag{3}$$

Since u or w or both can be the center of a child domain of Ω, we need to also consider vertices u', w' of M that define the insertion radius of v and the in-radius of V_u respectively. Since w does not encroach a domain at u and $|u-w| \leq |u-v'|$, it follows that

$$|u-u'| \leq \varepsilon |u-v'|. \tag{4}$$

The D-ball centered at v' has radius $|u'-v'|$ and is empty of vertices, so $\lambda_v \geq |u'-v'| - |v-v'|$. Using the triangle inequality, (3), and (4), we can bound the insertion radius as follows.

$$|u-v'| \leq \beta \lambda_v. \tag{5}$$

where, $\beta = \frac{1}{1-\varepsilon-\gamma}$. Since w is closer to u than v, $\mathbf{f}^{\Omega}_{M_{i-1}}(u) = \mathbf{f}^{\Omega}_{M_i}(u)$ and $\mathbf{f}^{\Omega}_{P_{i-1}}(u) = \mathbf{f}^{\Omega}_{P_i}(u)$. So, we can use induction and Lemma 6 to get that

$$\mathbf{f}^{\Omega}_{P_i}(u) \leq K'_I |u-w|. \tag{6}$$

We use K'_I because it is the largest of the K' constants.

We may now derive a bound on $\mathbf{f}^{\Omega}_{P_i}(v)$ as follows.

$$\begin{aligned}
\mathbf{f}^{\Omega}_{P_i}(v) &\leq \mathbf{f}^{\Omega}_{P_i}(u) + |u-v| && [\mathbf{f}^{\Omega}_{P_i} \text{ is 1-Lipschitz}] \\
&\leq \mathbf{f}^{\Omega}_{P_i}(u) + |u-v'| + |v'-v| && [\text{triangle inequality}] \\
&\leq \mathbf{f}^{\Omega}_{P_i}(u) + (1+\gamma)|u-v'| && [\text{by (3)}] \\
&\leq K'_I|u-w| + (1+\gamma)|u-v'| && [\text{by (6)}] \\
&\leq \left(\frac{3K'_I}{\tau} + 1 + \gamma\right)|u-v'| && [V_u \text{ aspect ratio} > \tau] \\
&\leq \left(\frac{3K'_I}{\tau} + 1 + \gamma\right)\beta\lambda_v && [\text{by (5)}]
\end{aligned}$$

So, setting $K'_S \geq \left(\frac{3K'_I}{\tau} + 1 + \gamma\right)\beta$ yields the desired bound.

Case 3: v is an input. Choose u such that $\lambda_v = |u-v|$ and let j and Ω_j be the time that u was inserted and the domain it was inserted into respectively. If $u \in C_\Omega$, then v encroaches on Ω, which is impossible. If u is an input vertex then $\lambda_v = f^{\Omega}_P(v)$ so we are done. So, we may assume that u is a Steiner point, inserted either as either a circumcenter or as a cage vertex that was later released.

We define $K'_u = K'_S$ in the former case and $K'_u = K'_C$ in the latter. By choosing $K'_I \geq \frac{K'_u}{\gamma} + 1$, we can now derive the following bound.

$$\begin{aligned}
\mathbf{f}^{\Omega}_{P_i}(v) &\leq \mathbf{f}^{\Omega}_{P_i}(u) + |u-v| && [\mathbf{f}^{\Omega}_{P_i} \text{ is 1-Lipschitz}] \\
&\leq \mathbf{f}^{\Omega_j}_{P_j}(u) + |u-v| && [\text{by Lemma 44}] \\
&\leq K'_u\lambda_u + |u-v| && [\text{by induction}] \\
&\leq \left(\frac{K'_u}{\gamma} + 1\right)|u-v| && [\text{because } u \text{ did not yield to } v] \\
&\leq K'_I|u-v| && \left[\frac{K'_u}{\gamma} + 1 \leq K\right] \\
&= K'_I\lambda_v. && [\lambda_v = |u-v|]
\end{aligned}$$

□

Lemma 8. *Let q and q' be any two input points and let r be the distance between them. If $A = \text{annulus}(q, 2r, \frac{6r}{\varepsilon^3})$ contains no input points, then q and q' are inside some cage contained in A for all intermediate meshes after each has been inserted.*

Sketch. Let $p_1, \ldots, p_k$ be all input points in $\text{ball}(p, 2r)$ ordered by the order in which they were inserted. Clearly q and q' are among the p_i's. The proof is a straightforward induction on k, requiring us only to show that each insertion leaves the desired cage around the previous set. The constant $\frac{6}{\varepsilon^3}$ was carefully chosen to make this work. The full proof is Lemma 47 in the appendix. □

We can now prove that the output mesh has size linear in the input size.

Theorem 9. *If M is the output of the* NetMesh *algorithm for an input set P, then $|M| = O(|P|)$.*

Sketch. Let Ω be any domain in the output. Let $p_1 \ldots, p_j$ be the vertices of P_Ω ordered such that for each $i = 3 \ldots j$, $\mathbf{f}_{P_i^\Omega}(p_i)/\mathbf{f}_{P_{i-1}^\Omega}(p_i) \geq \frac{12}{\varepsilon^3} + 1$, where $P_i = \{p_1, \ldots, p_i\}$. Lemma 8 guarantees that such an ordering can be found by a trivial greedy algorithm (see Lemma 46 for details of the construction).

In previous work [MPS08], we showed if P_Ω can be ordered this way then any well-spaced superset satisfying the bound in Lemma 7 has size $O(|P_\Omega|)$. So, in particular $|M_\Omega| = O(|P_\Omega|)$. Now, we observe that because every domain contains at least 2 input points, $\sum_\Omega |P_\Omega| < 2|P|$. So, the total mesh size can be bounded as $|M| \leq \sum_\Omega |M_\Omega| = O(\sum_\Omega |P_\Omega|) = O(|P|)$. □

9 Range Spaces and ϵ-Nets

In this section we discuss ideas and definitions from hypergraph and range space theory that we will need in our meshing algorithm. We will also give a distance measure derived from a range space that is useful for our analysis. A **range space** or **hypergraph** is a pair $(X, \mathcal{R})$ where X is a set and $\mathcal{R}$ is a collection of sets called **ranges**. A **range space ϵ-net** for $(X, \mathcal{R})$ is a subset N of X such that $N \cap R \neq \emptyset$ for all $R \in \mathcal{R}$ such that $|R \cap X| \geq \epsilon|X|$.

Throughout this discussion the ranges will be open balls in $\mathbb{R}^d$ including those with infinite radius, i.e. halfspaces. For a subset $M \subset \mathbb{R}^d$, define:

$$\mathcal{B}_M = \{B \ : \ B \text{ is a ball and } B \cap M = \emptyset\}.$$

A useful subset of $\mathcal{B}_M$ is the set of D-balls of M:

$$\mathcal{D}_M = \{B \in \mathcal{B}_M \ : \ B \text{ is a D-ball of } M\}.$$

The following geometric lemma is useful for translating between statements about D-balls and statements about arbitrary empty balls in the space.

Theorem 10. *If $M \subset \mathbb{R}^d$ and $B \in \mathcal{B}_M$ then B is covered by at most d D-balls of $\mathcal{D}_M$ and these d balls all share a common point.*

The proof is in Appendix A.

Let $\mathcal{G}_{\mathcal{B}_M}$ be the graph with vertex set $\mathcal{B}_M$ and edges for each pair of balls that intersect. For any $x, y \in R_d \setminus M$, let $\mathbf{d}_{\mathcal{B}_M}(x, y)$ be the length of the shortest path in $\mathcal{G}_{\mathcal{B}_M}$ between a ball containing x to a ball containing y. Define $\mathcal{G}_{\mathcal{D}_M}$ and $\mathbf{d}_{\mathcal{D}_M}$ similarly. These distances are related by the following lemma.

Lemma 11. *If $M \subset \mathbb{R}^d$ is finite then* $\mathbf{d}_{\mathcal{D}_M} \leq 2\mathbf{d}_{\mathcal{B}_M}$

Proof. Let $x, y \in \mathbb{R}^d$ be any pair of points and let $s = \mathbf{d}_{\mathcal{B}_M}(x, y)$. It will suffice to find D-balls $E_1, \ldots, E_{2s} \in \mathcal{D}_M$ such that $x \in E_1$, $y \in E_{2s}$, and each $E_i \cap E_{i+1}$ is nonempty. By the definition of $\mathbf{d}_{\mathcal{B}_M}$, there exists balls $B_1, \ldots, B_s \in \mathcal{B}_M$ such that $x \in B_1$, $y \in B_s$, and each $B_i \cap B_{i+1}$ is nonempty. Let z_i be a point in $B_i \cap B_{i+1}$ for $i = 1 \ldots s-1$ and define $z_0 := x$ and $z_s := y$. Now, by Theorem 10, there are d D-balls covering each B_i and they all have a common intersection. So, letting E_{2i-1} and E_{2i} be the D-balls among these that contain z_{i-1} and z_i gives the desired path of length at most $2s$ in $\mathcal{G}_{\mathcal{D}_M}$. □

10 Bounding the ply

In this section, we prove that the D-balls have constant ply, a fact that is useful for many parts of the analysis. In particular this is important for showing that the cost of a single insertion is constant. The constant ply can then be used to show that the degree of the intersection graph of the D-balls is bounded by a constant, and moreover, that the degree of every vertex of the Delaunay 1-skeleton is bounded by a constant. The latter bound implies that the total complexity of the Delaunay triangulation is linear in the number of vertices.

These results are known in the case of Voronoi diagrams with bounded aspect ratio [MTTW99], but we extend them to hold when there is a domain hierarchy rather than a single domain. To begin, we give some lemmas about the limited interaction between the different domains in the hierarchy.

Lemma 12. *Let (M, H) be a hierarchical mesh with cages. For any domain Ω, every D-ball B that intersects $B'_\Omega = \mathrm{ball}(c_\Omega, (1 - 2\delta - 2\gamma)r_\Omega)$ is contained in B_Ω.*

Proof. Suppose for contradiction that there exists a D-ball $B = \mathrm{ball}(c, r)$ such that both $B \cap B'_\Omega$ and $B \setminus B_\Omega$ are nonempty. Let z be the projection of c onto the sphere $\{x : |x - c| = (1 - \delta - \gamma)r_\Omega\}$. By the cage spacing properties, there is some vertex u in C_Ω such that $|u - z| < (\delta + \gamma)r_\Omega$. Next, $|c - z| \leq r - (\delta - \gamma)r_\Omega$ by our supposition and the triangle inequality. So, it follows that $|c - u| \leq |u - z| + |z - c| < r$. However, this implies that $u \in B$, contradicting the assumption that B is a D-ball. □

Lemma 13. *Let M be a τ-quality hierarchical mesh with parameters such that $\varepsilon + \varepsilon^2 < 1 - 2\delta - 2\gamma$. If Ω_1, Ω_2, Ω_3 are three nested domains in H_M, i.e. $\Omega_1 = \mathrm{p}(\Omega_2)$ and $\Omega_2 = \mathrm{p}(\Omega_3)$ then no D-ball intersects both Ω_1 and Ω_3.*

Proof. Suppose for contradiction that some ball B intersects both Ω_1 and Ω_3. Let c_2 and c_3 be the centers of Ω_2 and Ω_3 respectively. Since c_2 does not encroach the outside of Ω_3 and c_3 does not encroach the inside of Ω_3 we have that

$$\frac{1}{\varepsilon} r_{\Omega_3} \leq |c_2 - c_3| \leq \varepsilon r_{\Omega_2}. \tag{7}$$

Let x be a point of $B \cap \Omega_3$. By (7) and the triangle inequality, $|x - c_2| \leq (\varepsilon + \varepsilon^2) r_{\Omega_2}$. Since $\varepsilon + \varepsilon^2 < 1 - 2\delta - 2\gamma$, Lemma 12 implies that B is contained in B_{Ω_2} and therefore is disjoint from Ω_1, a contradiction. □

Lemma 14. *Let M be a τ-quality hierarchical mesh and let Ω be any domain in H_M. If B is a ball of radius r centered in B_Ω empty of points in M and $x \in B$, then*

$$\mathbf{f}_M^\Omega(x) \geq c_{14} r,$$

where $c_{14} = \frac{1}{24\tau^2}$

Proof. The proof of Lemma 6.1 from [HMP06] may be repeated verbatim here even though the Voronoi cells are defined differently because the proof only uses the the in-radius and out-radius conditions which are well-defined. □

Lemma 15. *Let M be a τ-quality mesh with $\varepsilon < \frac{1}{3}$. Let $B = \mathrm{ball}(c, r)$ be a D-ball corresponding to a simplex $\sigma \subset M$ and let x be a point of B. If Ω is a domain such that the ancestor of some vertex of σ in M_Ω is not the nearest neighbor of x in M_Ω, then*

$$\mathbf{f}_M^\Omega(x) \leq 3r.$$

Proof. Let u' be the ancestor of u in M_Ω assumed to exist, and thus $f_M^\Omega(x) \leq |x - u'|$. Since no domains are ε-encroached, $|u' - u| \leq \varepsilon|u' - v|$, and thus, by the triangle inequality $|u' - u| \leq \frac{\varepsilon|u-v|}{1-\varepsilon}$. Since u and v are on the boundary of B, $|u - v| \leq 2r$ and so, using the assumption that $\varepsilon < \frac{1}{3}$, $|u - u'| \leq r$. Because $x \in B$, $|x - u| \leq 2r$. Using the triangle inequality and the preceding inequalities, we get $f_M^\Omega(x) \leq |x - u| + |u - u'| \leq 3r$. □

Theorem 16. *A τ-quality hierarchical mesh has ply at most c_{16} where c_{16} depends only on the meshing parameters.*

Proof. Let x be any point and let Ω be the domain containing x. Let S be the set of D-balls containing x. For any $\sigma \in \mathrm{Del}(M)$, let B_σ be its D-ball and let Ω_σ be the least common ancestor domain of its vertices. Let n_x be the nearest neighbor of x in M_Ω and let Ω' be the child domain of Ω centered at n_x (if it exists).

There are two corner cases that we need to eliminate first: these are balls B_σ in S such that

1. $\Omega' = \Omega_\sigma$, or
2. $\Omega = \Omega_\sigma$ and B_σ is not centered in Ω.

In both of these cases, B_σ spans the bounding sphere of either Ω or Ω'. So, Lemmas 12 and 18 imply that there are only a constant number of such balls. Let S' be the subset of S formed by removing this constant sized set of balls.

Consider the following two subsets of S'.

$$S_1 = \{B_\sigma \in S' : \Omega = \Omega_\sigma \text{ or } \Omega = \mathrm{p}(\Omega_\sigma)\}$$
$$S_2 = \{B_\sigma \in S' : \mathrm{p}(\Omega) = \Omega_\sigma \text{ or } \mathrm{p}(\Omega) = \mathrm{p}(\Omega_\sigma)\}$$

By Lemma 12, S_1 and S_2 cover all of S'. Thus it will suffice to prove these sets have constant size. Ignoring the overlap, these two cases correspond to counting the balls that contain x that come from its own domain or children and those that come from a parent or sibling domain. In fact, the two cases are completely symmetric. The rest of the proof will show that $|S_1|$ is a constant. It can then be repeated for the second case by inserting S_2 and $\mathrm{p}(\Omega)$ in place of S_1 and Ω.

The set of vertices on simplices whose D-balls are in S_1 is $V = \{v \in \sigma : B_\sigma \in S_1\}$. Let v' denote the ancestor in M_Ω of any vertex $v \in V$, and define $V' = \{v' : v \in V\}$. We will first show that $|V| \leq c_{18}|V'|$. Then we will show that $|V'| \leq \alpha$, where $\alpha = \left(\frac{24\tau}{c_{14}}\right)^d$. Every D-ball in S_1 corresponds to some subset of $d+1$ points of V so will conclude that $|S_1| \leq \binom{|V|}{d+1} \leq (\alpha c_{18})^{d+1}$.

Claim: $|V| \leq c_{18}|V'|$.
Fix some $v \in V'$. Let $U = \{u \in V : u' = v'\}$ It will suffice to show that $|U| \leq c_{18}$. If $v' \in \Omega$, then $v = v'$ and so $|U| = 1$. So, we may assume that v' is the center of some domain $\Omega_{v'}$ whose parent is

Ω. Note that $B_\sigma \cap \Omega$ is nonempty for all $B_\sigma \in S_1$. So, any $u \in U$ came from a ball B_σ that spans the bounding sphere of $\Omega_{v'}$. Thus, Lemmas 12 and 18 implies that $|U| \le c_{18}$.

Claim: $|V'| \le \alpha$**, where** $\alpha = \left(\frac{24\tau}{c_{14}}\right)^d$**.**
Let $r_{\min}$ and $r_{\max}$ be the minimum and maximum radii among the balls of S_1. By Lemma 14, $\mathbf{f}_M^\Omega(x) \ge c_{14} r_{\max}$. Lemma 15 implies that $\mathbf{f}_M^\Omega(x) \le 3r_{\min}$. Combining these two facts, we see that all of the balls have radii that differ by at most a constant:

$$r_{\max} \le \frac{3r_{\min}}{c_{14}}. \tag{8}$$

Consider any ball $B_\sigma \in S_1$ with a vertex $v \in \sigma$. The bounded aspect ratio condition implies that

$$f_M^\Omega(v') \ge \frac{r_{\min}}{\tau}. \tag{9}$$

The vertices $v' \in V'$ are not too far from x compared to the radii of the balls:

$$\begin{aligned}
|v' - x| &\le |v - v'| + |v - x| && \text{[by the triangle inequality]}\\
&\le |v - v'| + 2r_{\max} && [v, x \in B \in S_1]\\
&\le \varepsilon \mathbf{f}_M^\Omega(v') + 2r_{\max} && \text{[non-encroachment]}\\
&\le 3r_{\max}. && \left[\varepsilon \le \frac{1}{3} \text{ and Lemma 15}\right]
\end{aligned} \tag{10}$$

We can now show that $|V'| \le \alpha$ by a volume packing argument. Specifically, let $U = \{\text{ball}(v', \frac{r_{\min}}{2\tau}) : v' \in V'\}$. Note that $|U| = |V'|$. By (9), these balls are disjoint. By (10), these balls are contained in a ball of radius $4r_{\max}$. Applying (8), we conclude that $|U| \le \left(\frac{24\tau}{c_{14}}\right)^d$. ☐

Corollary 1. *Every vertex v in a τ-quality hierarchical mesh M is in at most $(d+1)c_{16}$ Delaunay simplices.*

Proof. Let U be a set of $d+1$ points in $\text{ball}(c, r)$ such that $v \in \text{conv}(U)$ and r is sufficiently small so that every D-ball with v on its boundary intersects U. By Theorem 16, there are only c_{16} D-balls intersecting any point in U, so the total number of Delaunay simplices containing v is at most $(d+1)c_{16}$. ☐

Theorem 17. *Let M be a τ-quality mesh and let $\mathcal{G}_{\text{Del}}$ be the graph formed by the 1-skeleton of $\text{Del}(M)$. The maximum degree of any node of $\mathcal{G}_{\text{Del}}$ is bounded by a constant that depends only on the meshing parameters.*

Proof. Let $v \in M$ be any vertex. Note that every simplex containing v in $\text{Del}(M)$ has a corresponding D-ball with v on its boundary. Corollary 1 implies that there are only $(d+1)c_{16}$ such D-balls. Each such ball contributes at most d edges, so the total is at most $d(d+1)c_{16}$. ☐

Lemma 18. *Let (M, H) be a hierarchical mesh with no encroached cages. Let $\Omega \in H$ be a domain such that $\mathbf{f}_P^\Omega(z) \le K\mathbf{f}_M^\Omega(z)$ for all $z \in \Omega$. The number of vertices of M contained in $A = \text{annulus}(c_\Omega, 2\varepsilon r_\Omega, r_\Omega)$ is at most some constant c_{18} depending only on the meshing parameters.*

Proof. For points z in A, we know that $\mathbf{f}_P^\Omega(z) \ge \varepsilon r_\Omega$ because C_Ω is not encroached. It follows that for all $z \in M \cap A$, $\mathbf{f}_M^\Omega(z) \ge \frac{\varepsilon}{K} r_\Omega$. So, there must be disjoint balls of radius at least $\frac{\varepsilon}{2K} r_\Omega$ around each such z. Therefore, a simple packing completes the proof. ☐

Theorem 19. *Let M be a τ-quality mesh and let $\mathcal{G}_\mathcal{D}$ be the intersection graph of the D-balls of* $\mathrm{Del}(M)$*. The maximum degree of any node of $\mathcal{G}_\mathcal{D}$ is bounded by a constant c_{19} that depends only on the meshing parameters.*

Proof. Let $B = \mathrm{ball}(c, r)$ be any D-ball and let Ω be the domain containing its center. Let S be the set of D-balls intersecting B. We will show that $|S| \le c_{19}$ by bounding separately the number of such balls with radius at least βr and those with radius less than βr, where β is a constant independent of the dimension.

Let $S_1 = \{b \in S : \mathrm{radius}(b) \ge \beta r\}$. Observe that for any $b \in S_1$, $\mathrm{vol}(b \cap \mathrm{ball}(c, 1 + 2\beta)) \ge \beta^d \mathbb{V}_d$. Theorem 16 implies that $\mathrm{ball}(c, r + 2\beta r)$ is covered at most t times by the D-balls of S_1 and thus by volume packing,

$$|S_1| \le t(2 + \frac{1}{\beta})^d.$$

We will now bound the size of $S_2 = \{b \in S : \mathrm{radius}(b) < \beta r\}$. Let $B' = \mathrm{ball}(c', r')$ be a D-ball of S_2 and let V be the vertices of the Delaunay simplex corresponding to B'. Let $h : \mathcal{D}_M \to M$ be a map that takes a D-ball B' to an arbitrary vertex its corresponding Delaunay simplex. Let $g : S \to M_\Omega$ be a map defined as $g(B') = \mathrm{lca}_{M_\Omega}(h(B'))$. As a shorthand, we write $g(S)$ to denote $\bigcup_{B' \in S}\{g(B')\}$. The map g allows us to charge the balls of S to nearby vertices in M_Ω. In Lemma 20, we prove that

$$|g(S_2)| \le c_{20}.$$

Then, in Lemma 21, we prove that

$$|g^{-1}(v)| \le c_{21},$$

for all $v \in M_\Omega$. Together these allow us to conclude that

$$|S_2| = \sum_{v \in g(S_2)} |g^{-1}(v)| \le c_{20}c_{21}.$$

So, setting $c_{19} = t(2 + \frac{1}{\beta})^d + c_{20}c_{21}$, we conclude that

$$|S| = |S_1| + |S_2| \le c_{19}.$$

□

Lemma 20. *If S_2 is a collection of D-balls of radius at most βr intersecting a D-ball $B = \mathrm{ball}(c, r)$, then $|g(S_2)| \le c_{20}$, where $\beta = \frac{c_{14}}{4} r$ and c_{20} is a constant that depends only the meshing parameters.*

Proof. Fix some $B' \in S_2$ and let r' be its radius. Let Ω be the domain containing c. Let $u = h(B')$ and $v = g(B')$. If $u, v \in \Omega$ then $u = v$ and $|u - v| = 0$. The cage construction guarantees that if $u, v \notin \Omega$ then $|u - v| \le (1 - \delta + \gamma)|c - v|$. Using the triangle inequality, we know that $|c - v| \le r + 2r' + |u - v|$. Combining these inequalities, we get that

$$|c - v| \le \alpha r, \tag{11}$$

where $\alpha = \frac{1+2\beta}{\delta-\gamma}$.

We want to prove that the vertices of $g(S_2)$ are not too close together. To do this, we will bound the feature size at $v \in g(S_2)$. There are two cases to consider. First, if $|u - v| > \beta r$ then $\mathbf{f}_M^\Omega(v) \ge \beta r$ because the cage centered at v that contains u cannot intersect any point of M_Ω (other than v itself). Second, if $|u - v| \le \beta r$ then $\mathbf{f}_M^\Omega(v) \ge \mathbf{f}_M^\Omega(z) - |z - u| - |u - v|$, where $z \in B \cap B'$.

Since $z \in B$ and B is centered in Ω, Lemma 14 implies that $\mathbf{f}_M^\Omega \geq c_{14} r = 4\beta r$. Since z and u are in B', $|z-u| \leq 2r' \leq 2\beta r$. So in this case as well, we conclude that

$$\mathbf{f}_M^\Omega(v) \geq \beta r. \tag{12}$$

We can now complete the proof with a volume packing argument. For each $v \in g(S_2)$, we consider the ball $b_v = \text{ball}(v, \frac{\beta}{2} r)$. Inequality (12) implies that these balls are disjoint. Moreover, (11) implies that these balls are all contained in $\text{ball}(c, (\alpha + \frac{\beta}{2})r)$. It follows that the number of balls b_v can be at most $c_{20} = \left(\frac{2\alpha}{\beta} + 1\right)^d$. □

Lemma 21. *For all $v \in M_\Omega$, $|g^{-1}(v)| \leq c_{21}$, where c_{21} is a constant that depends only on the meshing parameters.*

Proof. Let $U = \{h(B') : B' \in g^{-1}(v)\}$. Corollary 1 implies that $|g^{-1}(v)| \leq (d+1)c_{16}|U|$. So, it will suffice to prove that $|U| \leq \frac{c_{21}}{(d+1)c_{16}}$.

Fix $B' = \text{ball}(c', r) \in g^{-1}(v)$ and let $u = h(B')$. If $u \in M_\Omega$ then $u = v$ and $|U| = 1$. So, we may assume that there is some domain $\Omega'' \in \text{children}(\Omega)$ centered at v. Let r'' be the radius of Ω''. There are two cases to consider.

Case 1: $u \in U_1 = \{u \in U : |u - v| \leq 2\varepsilon r''\}$. Lemma 12 implies that $B' \subset B_{\Omega''}$. Lemma 13 implies that $r' \geq (1 - 2\delta - 2\gamma - 2\varepsilon)r''$. Theorem 16 says that $B_{\Omega''}$ can only be covered c_{16} times by the balls B'. So, by volume packing $|U_1| \leq \alpha$, where $\alpha = (1 - 2\delta - 2\gamma - 2\varepsilon)^{-d}$.

Case 2: $u \in U_2 = \{u \in U : |u - v| > 2\varepsilon r''\}$. Lemma 18 implies that $|U_2| \leq c_{18}$.

We now conclude that $|U| = |U_1| + |U_2| \leq \alpha + c_{18}$. Choosing $c_{21} = (d+1)c_{16}(\alpha + c_{18})$ completes the proof. □

11 The Quality Invariant

In this section, we will prove that NetMesh maintains the quality invariant throughout the course of the algorithm. As shown in Section 10, this is an important property to have. Our goal will be to prove the following.

Theorem 22. *For any input, the intermediate meshes of the* NetMesh *algorithm are τ''-quality, where τ'' depends only on the mesh parameters.*

The proof will follow directly from Lemmas 23 and 29 below. The former guarantees that the quality is bounded after every call to InsertInput. The latter guarantees that the quality is bounded throughout the Clean operation.

11.1 Quality during input insertion.

In this section, we will show that starting with a τ-quality mesh and inserting an input point results in a τ'-quality mesh where τ' is a constant that depends only on the meshing parameters. Throughout this section, let M be the τ-quality starting mesh and let M' be the mesh after executing InsertInput(v) for some $v \in P$. Whenever we refer to a cell $\text{Vor}_M(u)$ or $\text{Vor}_{M'}(u)$, r_u and r'_u denote the respective in-radii and R_u, R'_u denote the out-radii. We say that a cell $\text{Vor}_M(u)$ is **caged** during InsertInput if a new cage is added that is contained in $\text{Vor}_M(u)$.

Lemma 23. *There is a constant τ' depending only on the meshing parameters such that the mesh after every call to* InsertInput *is τ'-quality.*

Proof. The CLEAN procedure explicitly guarantees that the starting mesh M is τ-quality. We need to prove that all cells $\mathrm{Vor}_{M'}(u)$ (excepting cage vertices) have aspect ratio at most τ', i.e. $\frac{R'_u}{r'_u} \leq \tau'$. Fix one such u. There are four different cases to consider:

1. The Voronoi cell in M, $\mathrm{Vor}_M(u)$, had aspect ratio at most τ.
2. $u = C_\Omega$ is a newly created cage.
3. u was a cage vertex in M that got released.
4. $u = v$ is the newly inserted input vertex.

Case 1: $\mathrm{Vor}_M(u)$ had aspect ratio at most τ. If $\mathrm{Vor}_M(u)$ was caged during INSERTINPUT then Lemma 25 implies that $\mathrm{Vor}_{M'}(u)$ has aspect ratio at most c_{25}. Otherwise, Lemma 24 implies that $r_u \leq c_{24} r'_u$. In this case, the out-radius cannot go up with the addition of more points so $R'_u \leq R_u$. Thus, since $R_u \leq \tau r_u$, we get that $R'_u \leq c_{24}\tau r'_u$ as desired.
Case 2: $u = C_\Omega$ is a newly created cage. For this case, Lemma 28 implies that $R'_u \leq c_{28} r'_u$.
Case 3: u was a cage vertex in M that got released. If u is caged then Lemma 25 implies that its aspect ratio is at most c_{25}. Otherwise, Lemma 24 implies that $r_u \leq c_{24} r'_u$. Let C_Ω be the released cage and let c be its center. The cage spacing guarantees that $r_\Omega \leq s r_u$, where $s = (\delta - \gamma)(1 - \delta - \gamma)$. So, it follows that

$$r_\Omega \leq c_{24} s r'_u. \tag{13}$$

Now, we must consider the cases where v encroached the inside or the outside of Ω. For an outer encroachment, Lemma 26 implies that $R_c \leq c_{26} r_\Omega$, where R_c is the out-radius of $\mathrm{Vor}_M(C_\Omega)$. The out-radius of a cage is strictly greater than the out-radius of its cage vertices, so $R'_u < R_c$ and thus by (13), $R'_u < c_{26} c_{24} s r'_u$.

For an inner encroachment, we call GROWCAGE, which conditionally adds a larger cage around the existing cage before releasing it. Lemma 27 implies that $R'_u \leq c_{27} r_\Omega$. So, (13) implies $R'_u \leq c_{27} c_{24} s r'_u$.
Case 4: $u = v$ is the newly inserted input vertex. Note that in all of the preceding cases, inserting u only increased the aspect ratio bound. Consequently, by the time INSERTINPUT actually adds u to the mesh, it is added to a quality mesh M'' with the same domain hierarchy as M'. Let Ω be the domain u is inserted into. Let w be the nearest vertex to u in M''_Ω. Note that u is ε-medial in M'', for otherwise we would have created a new domain or yielded when inserting w, and so $\mathbf{f}^\Omega_{M''}(u) \leq \frac{1}{\varepsilon}|u - w| \leq \frac{3}{\varepsilon} r'_u$. Since M'' is quality and the D-ball centered at the farthest corner of $\mathrm{Vor}_{M'}(u)$ is empty in M'', Lemma 14 implies that $R'_u \leq c_{14} \mathbf{f}^\Omega_{M''}(u)$. Thus, $R'_u \leq \frac{3c_{14}}{\varepsilon} r'_u$. □

Lemma 24. *If $\mathrm{Vor}_{M'}(u)$ is cell that is not caged during* INSERTINPUT(v)*, then $r_u \leq c_{24} r'_u$.*

Proof. We show the nearest neighbor of $\mathrm{Vor}_{M'}(u)$ cannot be too close. If it has a new nearest neighbor, it can only be from a neighboring cage recently added or the new input vertex. Since u does not encroach any new cages, they can only decrease the in-radius by a $1 - \varepsilon$ factor. Since it was not caged, v must have been medial and therefore r'_u can only go down by a $\frac{\varepsilon}{2}$ factor. So, choosing $c_{24} = \frac{2}{\varepsilon(1-e)}$ suffices to yield $r_u \leq c_{24} r'_u$ as desired. □

Lemma 25. *If $\mathrm{Vor}_{M'}(u)$ is caged during* INSERTINPUT(v)*, then $R'_u \leq c_{25} r'_u$, where c_{25} depends only on the meshing parameters.*

Proof. Let C_Ω be the cage. The new Voronoi cell is contained in the newly formed domain, so $R'_u \leq r_\Omega$. If any other point was added to Ω, then it must have been v and so by construction, $r'_u \geq \frac{\varepsilon}{2} r_\Omega$. So, it suffices to choose $c_{25} = \frac{2}{\varepsilon}$. □

The following two lemmas show that cage vertices released in the algorithm have out-radii bounded by a constant times the radius of the cage they belonged to.

Lemma 26. *Let C_Ω be a cage in a hierarchical mesh M and let $\Omega' = \mathrm{p}(\Omega)$. Suppose there exists a point v that is ε-medial in Ω' and outer encroaches C_Ω. If $\mathrm{Vor}_M(C_\Omega)$ has aspect ratio at most τ then the out-radius of $\mathrm{Vor}_M(C_\Omega)$ is at most $c_{26} r_\Omega$, where c_{26} depends only on the meshing parameters.*

Proof. Let R denote the out-radius of $\mathrm{Vor}_M(C_\Omega)$ and let c be the center of C_Ω. Since the aspect ratio is at most τ, it follows that $R \leq \tau \mathbf{f}_M^{\Omega'}(c)$. Let w be the second nearest neighbor of v in $M_{\Omega'}$. Then, we can bound $\mathbf{f}_M^{\Omega'}(c)$ as follows.

$$\begin{aligned}
\mathbf{f}_M^{\Omega'}(c) &\leq \mathbf{f}_M^{\Omega'}(v) + |c - v| && \left[\mathbf{f}_M^{\Omega'} \text{ is 1-Lipschitz}\right] && (14)\\
&\leq |v - w| + |c - v| && [\text{by the choice of } w] && (15)\\
&\leq (1 + \frac{1}{\varepsilon})|c - v| && [v \text{ is } \varepsilon\text{-medial}] && (16)\\
&\leq \frac{1+\varepsilon}{\varepsilon^2} r_\Omega. && [v \text{ encroaches } \Omega] && (17)
\end{aligned}$$

So, it suffices to choose $c_{26} = \frac{\tau(1+\varepsilon)}{\varepsilon^2}$. □

Lemma 27. *Let C_Ω be a cage in a hierarchical mesh M. Let M' be the resulting mesh after* GrowCage*(C_Ω). If $\mathrm{Vor}_M(C_\Omega)$ has aspect ratio at most τ then the out-radius of $\mathrm{Vor}_{M'}(u)$ is at most $c_{27} r_\Omega$ for all $u \in C_\Omega$, where c_{27} depends only on the meshing parameters.*

Proof. Let R_u denote the out-radius of $\mathrm{Vor}_{M'}(u)$. In the GrowCage routine, either a new cage is added or it is not. In the former case, the new cage has radius $\frac{r_\Omega}{\varepsilon}$ so $R_u \leq \frac{r_\Omega}{\varepsilon}$ in this case. If the new cage is not added, it is because, $r_C \leq \frac{r_\Omega}{\varepsilon}$. Because we assumed that the Voronoi cell of C_Ω had aspect ratio at most τ, it follows that $R_C \leq \tau r_C \leq \frac{\tau r_\Omega}{\varepsilon}$. By the definition of R_C, we have that $R_u \leq R_C$ and thus $R_u \leq \frac{\tau r_\Omega}{\varepsilon}$ as desired. □

Lemma 28. *If $u = C_\Omega$ is a cage added during* InsertInput*, then the $R'_u \leq c_{28} r'_u$, where c_{28} depends only on the meshing parameters.*

Proof. If u is caged, then Lemma 25 implies $\mathbb{R}'_u \leq c_{25} r'_u$. So, we may assume that u is not caged. Let c be the center of Ω and Ω' is the previous domain that C_Ω as inserted into. So, $\mathbf{f}_M^{\Omega'}(c) = 3r_u$ and thus Lemma 24 implies that

$$\mathbf{f}_M^{\Omega'}(c) \leq 3c_{24} r'_u. \tag{18}$$

Let B be the D-ball centered at the far corner x of $\mathrm{Vor}_{M'}(u)$ and let r be its radius. So, $R'_u \leq r + r_\Omega$. Since C_Ω is not encroached, we have that $r_\Omega \leq r$ and thus

$$R'_u \leq 2r. \tag{19}$$

Since B was empty in M, Lemma 14 implies that there is a $y \in \Omega$ such that

$$\mathbf{f}_M^{\Omega'}(y) \geq c_{14} r. \tag{20}$$

So, $\mathbf{f}_M^{\Omega'}(y) \leq \mathbf{f}_M^{\Omega'}(c) + r_\Omega$ by the Lipschitz property of $\mathbf{f}_M^{\Omega'}$. Next, $r_\Omega < r'_u$, so (18) implies that $f_M^{\Omega'}(y) \leq (3c_{24} + 1)r'_u$. So, by (20),

$$r \leq c_{14}(3c_{24} + 1)r'_u. \tag{21}$$

Therefore, (19) and (21) imply that $R'_u \leq 2c_{14}(3c_{24} + 1)r'_u$. Choosing $c_{28} = 2c_{14}(3c_{24} + 1)$ completes the proof. □

11.2 Quality during the refinement process.

Lemma 29 (Clean preserves quality)**.** *Let M' be any intermediate mesh in the course of running* Clean(M) *on a τ'-quality mesh M. Then, M' is τ''-quality, where τ'' depends only on the meshing parameters.*

Proof. Let $V = \mathrm{Vor}^{\Omega}_{M'}(v)$ for some $v \in M'_{\Omega}$ and $\Omega \in H_{M'}$. We need to prove that the aspect ratio $\frac{R_V}{r_V}$ is at most τ''. There are two cases: V is the Voronoi cell of a vertex or V is the Voronoi cell of a cage.

Case 1: V is the Voronoi cell of the vertex v. First, we observe that there is D-ball $B = \mathrm{ball}(x, R_V)$ centered on the farthest corner of V. By Lemma 14, $R_V \leq 24{\tau'}^2 \mathbf{f}^{\Omega}_M(v)$. Now, we observe that $r_V \geq \frac{1}{3}\mathbf{f}^{\Omega}_{M'}(v)$ by Lemma 2. We apply Lemma 48 to get that $r_V \geq \frac{1}{3K'}\mathbf{f}^{\Omega}_M(v)$. So, we get that $\frac{R_V}{r_V} \leq 72K'{\tau'}^2$.

Case 2: V is the Voronoi cell of a cage centered at v. Since no new cages are created during Clean, it must be that $v \in M$. Let c be the center of the cage C that contains v. The cage spacing guarantees that $r_V \geq \frac{\delta - 2\gamma}{2}|c - v|$. Let s be the Steiner point whose insertion caused C to be released. Let M'' be the mesh it was inserted into and let Ω' be the domain it encroached. Now, R_V cannot be larger than the out-radius of $\mathrm{Vor}(C)$ in M'', which, in turn, is at most R_C, the out-radius of $\mathrm{Vor}(C)$ in M because Voronoi cells can only shrink during cleaning. We now bound R_C as follows:

$$
\begin{aligned}
R_C &\leq \tau' r_C && \big[\text{because } M \text{ is } \tau'\text{-quality}\big] && (22)\\
&\leq \tau' \mathbf{f}^{\Omega'}_M(c) && [\text{by Lemma 2}] && (23)\\
&\leq \tau' K' \mathbf{f}^{\Omega'}_{M''}(c). && [\text{by Lemma 48}] && (24)
\end{aligned}
$$

Now, since s was inserted by a Clean operation into Ω', $\mathbf{f}^{\Omega'}_{M''}(c) \leq 3|c - s|$. Moreover, since s encroaches, we have that $|c - s| \leq \frac{|c-v|}{\varepsilon}$. So, we conclude that

$$R_V \leq \frac{3\tau' K'(1+\varepsilon)}{\varepsilon(1-\varepsilon)(\delta - 2\gamma)} r_v.$$

□

12 Point Location Analysis

Definition 9. *A vertex $v \in M$* ***touches*** *an uninserted point $u \in P \setminus M$ if when v was inserted into M there were intersecting D-balls B_u and B_v containing u and v respectively.*

The quality invariant and Theorem 4 guarantee that only a constant number of balls are created or destroyed during an insertion, so the total amount of point location work done on any input point is $O(t)$, where t is the number of times it was touched.

Theorem 30. *The total cost of point location in the* NetMesh *algorithm is $O(n \log n)$.*

Proof. As noted before, it suffices to count the number of touches on uninserted input points throughout the algorithm. Since there are only $O(\log n)$ rounds, it will suffice to show that no input point can be touched more than a constant number of times in a single round.

Let M be the mesh at the start of a round. Consider any point $p \in P$. We will show that p cannot be touched more than a constant number of times in this round. By definition, a point x touches p

if $\mathbf{d}_{\mathcal{D}_{M'}}(p,x) \le 1$ in the mesh M' just prior to inserting x. So, it follows that $\mathbf{d}_{\mathcal{B}_M}(p,x) \le 1$ because D-balls in M' are empty of points of M. Moreover, by Lemma 11, $\mathbf{d}_{\mathcal{D}_M}(p,x) \le 2$. Therefore, the set of points that can touch p this round are all contained in one of the constant number of D-balls that are within 2 hops of p in $\mathcal{G}_{\mathcal{D}_M}$. In Lemma 31 below, we show that only a constant number of points are added to any D-ball in a single round. Thus, the total number of points that can touch p in a round is at most a constant. □

Lemma 31. *In any round starting with a mesh M, at most a constant number of points are added to any D-ball of M.*

Proof. Fix a particular round and let B be a D-ball of M. Let M' denote the mesh at the end of the round. Let P' denote the input points of M'. We wish to upper bound the number of points of M' in B. By standard mesh size analysis,

$$\begin{aligned} |M' \cap B| &= \sum_{\Omega \in H_{M'}} O\left(\int_{B_\Omega \cap B} \frac{dx}{\mathbf{f}^{\Omega}_{P'_\Omega}(x)^d} \right) \\ &\le \sum_{\Omega \in H_{M'}} O\left(\int_{B_\Omega \cap B} \frac{dx}{\mathbf{f}^{\Omega}_{P'_\Omega \cup M_\Omega}(x)^d} \right). \end{aligned} \tag{25}$$

Lemma 32 shows that there are only c_{32} terms in this summation and Lemma 35 shows each term is at most c_{35}. Thus, we conclude that $|M' \cap B| \le c_{32}c_{35}$. □

Lemma 32. *Let M and M' be the meshes before and after a round of the* NetMesh *algorithm. For any D-ball B in M', at most a constant number of domains of $H_{M'}$ intersect B.*

Proof. Let x be the center of B. There are only a constant number of domains of H_M intersecting B, because each contains a D-ball intersecting B and Theorem 4 implies there can only be a constant number of such balls. Any newly created domains must have been caused by the insertion of an input point $y \in M' \setminus M$. However, if the new domain intersects B then either y caused a cage from M to grow or $\mathbf{d}_{\mathcal{B}_M}(x,y) \le 5$. In either case, there are only a constant number of new domains intersecting B. □

In the following lemmas, we fix a particular domain Ω and use the following simplified notations. We number the k input points added this round as $P'_\Omega \setminus P_\Omega = \{p_1, \ldots, p_k\}$ where the ordering is the one given in Lemma 33 below. The part of the ball of Ω contained in B is defined as $A = B_\Omega \cap B$. The **near input points** are denoted Q and are formally defined as

$$Q = \{q \in P'_\Omega \setminus P_\Omega : \mathbf{d}_{\mathcal{B}_M}(q,x) \le 5 \text{ for some } x \in B\}.$$

The index set of the **far input points** is $I = \{i : p_i \notin Q\}$ and $I_0 = I \cup \{0\}$. Let $S_i = M_\Omega \cup \{p_1, \ldots, p_i\}$ and set $S_0 = M$. The function f_i is equal to $\mathbf{f}^{\Omega}_{S_i}$.

We partition the set A into pieces based on which far point was the last to affect the feature size:

$$U_j = \{x \in A : \max\{i \in I : f_i(x) \ne f_{i-1}(x)\} = j\}$$

and

$$U_0 = A \setminus \bigcup_{j \in I} U_j.$$

Define h_i and h_{ij} as

$$h_i = \int_A \frac{dx}{f_i(x)^d}$$

and

$$h_{ij} = \int_{U_j} \frac{dx}{f_i(x)^d}.$$

Since A is the disjoint union of $\{U_j : j \in I_0\}$,

$$h_i = \sum_{j \in I_0} h_{ij}.$$

Lemma 33. *There exists an ordering $\{p_1, \dots, p_k\}$ of $P'_\Omega \setminus P_\Omega$, such that $h_i - h_{i-1} \le c_{33}$, for all $i = 1 \dots k$.*

Proof. The desired ordering is a so-called **well-paced** ordering. It is one for which $\mathbf{f}^\Omega_{S_i}(p_i) \ge \alpha \mathbf{f}^\Omega_{S_{i-1}}(p_i)$ for all i. In previous work [MPS08], we showed that the change in the feature size integral over any domain is at most a constant after inserting a well-paced point. Calling this constant c_{33}, it will suffice to show that $P'_\Omega \setminus P_\Omega$ is well-paced with respect to M_Ω. This requires the same case analysis as used in Lemma 47, though it is easier in this case because we only require the inputs to be well-paced with respect to $M - \Omega$ rather than the stronger condition that they be well-paced with respect to the bounding cage. Alternatively, one could get an explicit ordering directly from the algorithm by keeping a list for each domain, appending new input points to the list corresponding to the domain that contains them, and appending the list for a released domain to the list of its parent. That the resulting list satisfies the well-paced condition is immediate from the algorithm and the feature size invariant. □

Lemma 34. $|Q| \le c_{34}$, *where c_{34} is a constant that depends only on d and the meshing parameters.*

Proof. By definition, $q \in Q$ implies that $\mathbf{d}_{\mathcal{B}_M}(q, x) \le 5$ for some $x \in B$. It is easily checked that Theorem 19 and Lemma 11 implies that at most $10c_{19}$ D-balls can contain points of Q. Each round is defined by selecting only a constant size net from each D-ball. So each D-ball only contributes at most a constant number of points to Q and thus the total size of Q is at most a constant. □

Lemma 35. $\int_{B_\Omega \cap B} \frac{dx}{\mathbf{f}^\Omega_{P'_\Omega \cup M_\Omega}(x)^d} \le c_{35}$, *where c_{35} is a constant that depends only on d and the meshing parameters.*

Proof. In our simplified notation, the statement reduces to proving that $h_k \le c_{35}$. Writing h_k as a telescoping sum, we get

$$\begin{aligned} h_k &= h_0 + \sum_{i=1}^{k} (h_i - h_{i-1}) \\ &= h_0 + \sum_{p_i \in Q} (h_i - h_{i-1}) + \sum_{i \in I} (h_i - h_{i-1}) \\ &\le h_0 + c_{33} c_{34} + \sum_{i \in I} (h_i - h_{i-1}) && \text{[by Lemmas 33 and 34]} \end{aligned}$$

Far input points cannot change the feature size by very much. This is formalized in Lemma 51, where it is proven that

$$\sum_{i \in I} (h_i - h_{i-1}) \le (c_{51} - 1) h_0.$$

The feature size in an empty ball cannot be too small compared to its radius. Specifically, Lemma 52 shows that $h_0 \leq c_{52}$. So setting $c_{35} = c_{33}c_{34} + c_{51}c_{52}$ suffices to complete the proof. See Appendix D for Lemmas 51 and 52 and their proofs. □

13 Finishing the mesh

The finishing process takes a hierarchical quality mesh and returns a well-spaced mesh.

Theorem 36. *Given a hierarchical quality mesh, the* FinishMesh *procedure runs in $O(m)$ time, where m is the size of the output mesh.*

Proof. The GrowCage and Clean procedures preserve the quality of the mesh, so each insertion takes constant time. There is no point location work to be done, so the total running time is linear in the number of points added. □

14 Conclusion and Future Work

In this paper, we have given an algorithm for generating quality hierarchical meshes of point sets with size $O(n)$ in $O(n \log n)$ time. We also showed how to extend these hierarchical meshes to traditional well-spaced meshes in optimal output-sensitive time $O(n \log n + m)$. The algorithm and its analysis introduce novel uses of ϵ-nets and the linear-size meshing theory introduced in [MPS08].

Future Work We have restricted our discussion to the rarefied case of point set inputs. We expect it should now be possible to design a work efficient algorithm for inputs with higher dimensional features such as segments and faces. The algorithm presented is basically a work efficient parallel algorithm. It should be possible to show the present algorithm runs in polylog parallel time with no increase in work and thus beating the time and work bounds in parallel SVR [HMP07].

Yet another issue is integrating ideas from the NetMesh algorithm into the already relatively fast SVR code [AHMP07]. Future experiments in this direction are in order. The algorithm removes the spread term in the run time for the mesh based persistent homology algorithms [HMOS10]. It may also have applications for efficient surface reconstruction especially in the higher dimensional cases[Dey07].

References

[AHMP07] Umut A. Acar, Benoît Hudson, Gary L. Miller, and Todd Phillips. SVR: Practical engineering of a fast 3D meshing algorithm. In *Proc. 16th International Meshing Roundtable*, pages 45–62, 2007. 14

[BEG94] Marshall Bern, David Eppstein, and John R. Gilbert. Provably Good Mesh Generation. *Journal of Computer and System Sciences*, 48(3):384–409, June 1994. 1

[BET99] Marshall W. Bern, David Eppstein, and Shang-Hua Teng. Parallel construction of quadtrees and quality triangulations. *International Journal of Computational Geometry and Applications*, 9(6):517–532, 1999. 1

[Cha00] Bernard Chazelle. *The Discrepancy Method*. Cambridge University Press, 2000. 6.1, 6.5

[Cla88] Kenneth L. Clarkson. A Randomized Algorithm for Closest-Point Queries. *SIAM Journal on Computing*, 17(4):830–847, August 1988. 1

[Dey07] T. K. Dey. *Curve and Surface Reconstruction : Algorithms with Mathematical Analysis*. Cambridge University Press, 2007. 14

[Gon85] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985. 4

[HMOS10] Benoît Hudson, Gary L. Miller, Steve Y. Oudot, and Donald R. Sheehy. Topological inference via meshing. In *Symposium on Computational Geometry*, 2010. 14

[HMP06] Benoît Hudson, Gary Miller, and Todd Phillips. Sparse Voronoi Refinement. In *Proceedings of the 15th International Meshing Roundtable*, pages 339–356, Birmingham, Alabama, 2006. Long version available as Carnegie Mellon University Technical Report CMU-CS-06-132. 1, 3, 7, 8, 10

[HMP07] Benoît Hudson, Gary L. Miller, and Todd Phillips. Sparse Parallel Delaunay Refinement. In *19th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 339–347, San Diego, June 2007. 1, 14

[HPM06] Sariel Har-Peled and Manor Mendel. Fast construction of nets in low dimensional metrics, and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006. 2

[HPÜ05] Sariel Har-Peled and Alper Üngör. A Time-Optimal Delaunay Refinement Algorithm in Two Dimensions. In *Symposium on Computational Geometry*, 2005. 1

[Mat02] Jiří Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, 2002. 4

[MPS08] Gary L. Miller, Todd Phillips, and Donald R. Sheehy. Linear-size meshes. In *CCCG: Canadian Conference in Computational Geometry*, 2008. 1, 7, 8, 12, 14

[MTTW99] Gary L. Miller, Dafna Talmor, Shang-Hua Teng, and Noel Walkington. On the radius-edge condition in the control volume method. *SIAM J. on Numerical Analysis*, 36(6):1690–1708, 1999. 7, 10

[MV00] Scott A. Mitchell and Stephen A. Vavasis. Quality mesh generation in higher dimensions. *SIAM J. Comput.*, 29(4):1334–1370 (electronic), 2000. 1

[Rup95] Jim Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, 18(3):548–585, 1995. Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) (Austin, TX, 1993). 8

[Yao81] Andrew Chi-Chih Yao. A lower bound to finding convex hulls. *J. ACM*, 28:780–787, October 1981. 1

A The Ball Cover Theorem

A.1 Carathéodory's Theorem

Carathéodory's Theorem is a classic result on convex sets that is critical to our proof of the D-ball Cover Theorem.

Theorem 37 (Carathéodory's Theorem)**.** *Let $A \subset \mathbb{R}^d$. If $x \in \mathrm{cone}(A)$ then $x \in \mathrm{cone}(A')$ for some $A' \subseteq A$ such that $|A'| \leq d$. If $x \in \mathrm{conv}(A)$ then $x \in \mathrm{conv}(A')$ for some $A' \subseteq A$ such that $|A'| \leq d + 1$.*

We will need the following extended form of Carathéodory's Theorem for $\mathcal{V}$-polyhedra, i.e. those formed by the Minkowski sum of a polytope and a cone.

Corollary 2. *If $x \in \mathrm{conv}(A) + \mathrm{cone}(B)$, then there exist subsets $A' \subset A$ and $B' \subset B$ such that*

$$x \in \mathrm{conv}(A') + \mathrm{cone}(B') \text{ and } |A'| + |B'| \leq d + 1.$$

Moreover, if $|A'| = 0$ then $|B'| \leq d$.

Proof. Using the cone form of Carathéodory's Theorem, it suffices to observe that

$$x \in \mathrm{conv}(A) + \mathrm{cone}(B) \text{ if and only if } \begin{bmatrix} x \\ 1 \end{bmatrix} \in \mathrm{cone} \begin{bmatrix} A & B \\ 1 & 0 \end{bmatrix}.$$

□

We will also make use of the following technical lemma related to $\mathcal{V}$-polyhedra.

Lemma 38. *If $\mathrm{cone}(v) \subseteq \mathrm{conv}(A) + \mathrm{cone}(B)$ for some $v \in \mathbb{R}^d$ then $v \in \mathrm{cone}(B)$.*

Proof. We will prove the contrapositive. Suppose that $v \notin \mathrm{cone}(B)$. Then for a sufficiently large t, $\mathbf{d}(tv, \mathrm{cone}(B)) > \max_{a \in A} |a|$. Let $z = a + b$ be the nearest points of $\mathrm{conv}(A) + \mathrm{cone}(B)$ to tv, where $a \in \mathrm{conv}(A)$ and $b \in \mathrm{conv}(B)$.

$$\begin{aligned} |tv - z| &\geq |tv - b| - |z - b| \\ &= |tv - b| - |a| \\ &> 0. \end{aligned}$$

Thus, $tv \notin \mathrm{conv}(A) + \mathrm{cone}(B)$ and therefore $\mathrm{cone}(v) \not\subseteq \mathrm{conv}(A) + \mathrm{cone}(B)$. □

Lemma 39. *If c is a point in the $\mathcal{V}$-polyhedron $\mathrm{conv}(Q) + \mathrm{cone}(H)$, where $c \neq 0$ and $Q \neq \emptyset$ then for all $x \in \mathbb{R}^d$, either or both of the following hold:*

1. $x^T(q-c) \geq 0$ *for some* $q \in Q$*, or*

2. $x^T h > 0$ *for some* $h \in H$*.*

Proof. Let the coefficients α_i and β_j be such that

$$c = \sum_{i=1}^{|Q|} \alpha_i q_i + \sum_{j=1}^{|H|} \beta_j h_j,$$

where $\alpha_i, \beta_j \geq 0$ and $\sum \alpha_i = 1$. Suppose for contradiction that $x^T(q_i - c) < 0$ for all $q_i \in Q$, and $x^T h_j \leq 0$ for all $h_j \in H$. So, it follows that

$$\sum_{i=1}^{|Q|} \alpha_i x^T(q_i - c) + \sum_{j=1}^{|H|} \beta_j x^T h_j < 0.$$

Factoring this expression implies that

$$x^T \left(\sum_{i=1}^{|Q|} \alpha_i q_i + \sum_{j=1}^{|H|} \beta_j h_j - \sum_{i=1}^{|Q|} \alpha_i c \right) < 0.$$

However, the left side of the above inequality simplifies to $x^T(c-c) = 0$, a contradiction. □

Lemma 40. *If* $c \in \text{cone}(H)$ *for some* $H \subset \mathbb{R}^d$ *and* B_c *is a ball centered at* c *that does not contain the origin, then for all* $x \in B_c$*, there exists* $h \in H$ *such that* $x^T h > 0$*.*

Proof. Since $c \in \text{cone}(H)$, there are nonnegative coefficients $\{\beta_h\}_{h \in H}$ such that $c = \sum_{h \in H} \alpha_h h$. Fix any $x \in B_c$. Since $0 \notin B_c$, $|c - x| < |c - 0|$ and therefore $x^T c > \frac{|x|}{2} > 0$. Suppose for contradiction that $x^T h \leq 0$ for all $h \in H$. Then $x^T c = \sum_{h \in H} \alpha_h x^T h \leq 0$, a contradiction. □

A.2 A simpler version of the D-ball Cover Theorem

We warm up with a simpler version of the main result. It deals with the special case of balls centered in bounded Voronoi cells. It only proves the weaker bound of $d+1$ rather d balls to cover and it does not prove that the covering balls are pairwise intersecting.

We start with a lemma that gives a sufficient condition for a point x to be in a bounded D-ball B_q.

Lemma 41. *Let* c*,* q*, and* v *be three points such that* $v^T q = v^T c = 0$*. Let* B_q *be an open ball centered at* q *with* v *on its boundary. Let* B_c *be an open ball centered at* c *that does not contain* v*. If* $x \in B_c$ *and* $x^T(q-c) \geq 0$ *then* $x \in B_q$*.*

Proof. Let $x \in B_c$ and $x^T(q-c) \geq 0$ according to the hypothesis. Since $x^T(q-c) \geq 0$, we have that

$$x^T q \leq x^T c. \tag{26}$$

Since $x \in B_c$ and $v \notin B_c$, $|c-x| < |c-v|$ and therefore because $v^T c = 0$,

$$x^T x - 2x^T c < v^T v. \tag{27}$$

It will suffice to prove that $|q - x| < |q - v|$. This follows from the following inequalities.

$$\begin{aligned}
|q - x|^2 &= q^T q - 2x^T q + q^T q \\
&\le q^T q - 2x^T c + q^T q && \text{[by (26)]} \\
&\le v^T v + q^T q && \text{[by (27)]} \\
&\le |q - v|^2. && \left[\text{because } v^T q = 0\right]
\end{aligned}$$

□

Theorem 42. *Let M be a finite set. For any ball $B_c \in \mathcal{B}_M$ centered in a bounded Voronoi cell, there is a collection of at most $d + 1$ D-balls that cover B_c.*

Proof. Let c be the center of B_c and let v be the nearest neighbor of c in M. By Carathéodory's Theorem, there exists a subset Q of the corners of $\mathrm{Vor}(v)$ such that $c \in \mathrm{conv}(Q)$ and $|Q| \le d + 1$. So, for any $x \in \mathbb{R}^d$, there is a $q \in Q$ such that $x^T(q - c) \ge 0$. Each $q \in Q$ corresponds to a D-ball B_q of radius $|q - v|$. Without loss of generality, we may assume v is the origin. Therefore, Lemma 41 implies that $x \in B_q$. □

A.3 The D-ball Cover Theorem

To prove the more general ball cover theorem, we need to be more careful to deal with infinite D-balls. The infinite D-balls are those corresponding to the facets of the convex hull of M. Thus, they are in correspondence with the unbounded 1-faces of Vor_M. The Voronoi cells of Vor_M are $\mathcal{V}$-polyhedra. That is, they can be written as the Minkowski sum of a convex polytope and a polytopal cone. So, there exists finite sets A, B such that $\mathrm{Vor}_M(v) = \mathrm{conv}(A) + \mathrm{cone}(B)$. Moreover, the set A is a subset of the Voronoi corners and B is a subset of the normals of the facets of $\mathrm{conv}(M)$. Thus, the points of A and the vectors of B are all in correspondence with the D-balls of $\mathcal{D}_M$.

Theorem 43 (The D-ball Cover Theorem)**.** *For all $B \in \mathcal{B}_M$, there exist d D-balls $b_1, \ldots, b_d \in \mathcal{D}_M$ such that $B \subseteq \bigcup_{i=1}^d b_i$. Moreover, these D-balls have a nonempty, common intersection.*

Proof. Let c be the center of B and let v be the nearest neighbor of c in M. Consider the ray starting from v that passes through c parameterized as $r(t) = v + t(c - v)$ for $t \ge 0$. We must distinguish between the cases where $r(t) \in \mathrm{Vor}(v)$ for all $t \ge 0$ and where there is some t for which $r(t) \notin \mathrm{Vor}(v)$.
Case 1: $r(t) \in \mathrm{Vor}(v)$ for all $t \ge 0$. Voronoi cells are $\mathcal{V}$-polyhedra and can be written as $\mathrm{Vor}(v) = \mathrm{conv}(Q) + \mathrm{cone}(H)$ where Q is a subset of D-ball centers and H is a set of normals of unbounded D-balls. Without loss of generality, we may assume that $v = 0$. By Carathéodory's Theorem, there is a subset $H' \subseteq H$ of size d such that $c \in \mathrm{cone}(H')$. So, Lemma 40 implies that for all $x \in B$ there is an $h \in H'$ for which $x^T h > 0$. So, B is covered by the d unbounded D-balls corresponding to the vectors in H'. Moreover, $c^T h > 0$ and therefore, the chosen D-balls have a common intersection at c.
Case 2: $r(t) \notin \mathrm{Vor}(v)$ for some $t \ge 0$. In this case, the ray must leave the Voronoi cell and so for some t', the point $c' = r(t')$ is in $\mathrm{Vor}(u) \cap \mathrm{Vor}(v)$ for some u in M. The set $\mathrm{Vor}(u) \cap \mathrm{Vor}(v)$ is a $d - 1$-dimension $\mathcal{V}$-polyhedron, and thus can be written as $\mathrm{conv}(Q) + \mathrm{cone}(H)$ where Q is a subset of D-ball centers and H is a set of normals of unbounded D-balls. So, by Carathéodory's Theorem, there are subsets $Q' \subset Q$ and $H' \subseteq H$ such that $c \in \mathrm{conv}(Q') + \mathrm{cone}(H')$ and $|Q'| + |H'| \le d$. Without loss of generality, we may assume $\frac{v+u}{2}$ is the origin. For $q \in Q'$ or $h \in H'$ let B_q and B_h be the bounded and unbounded D-balls corresponding to q and h respectively. Fix any $x \in B$.

Lemma 39 implies that $x^T(q-c) \geq 0$ for some $q \in Q'$ or $x^T h > 0$ for some $h \in H'$. In the former case, Lemma 41 implies that $x \in B_q$. In the latter case, $x \in B_h$, because 0 is on the boundary of B_h for all $h \in H$.

We now observe that $0 \in B_q$ for all $q \in Q'$, because $|q-v|^2 = |q|^2 + |v|^2 > |q-0|^2$. So, the bounded D-balls all have a common intersection at 0 and in fact at a sufficiently small open ball U centered at 0. So the intersection of U with the relative interior of cone(H') is contained in the common intersection of the chosen D-balls. □

B Technical Lemmas for Size Bounds

The following useful lemma guarantees that the feature size of any mesh vertex is maximized at the time it is inserted. Such a fact would be trivial if not for the possibility to change the underlying domains.

Lemma 44. *If M is a hierarchical mesh constructed by the* NetMesh *algorithm, then for all $v \in M$,*

$$\max_i \max_{\substack{\Omega \in H_i: \\ v \in M_{i\Omega}}} \mathbf{f}_{P_i}^{\Omega}(v) = \mathbf{f}_{P_j}^{\Omega_j}(v),$$

where j is the insertion time of v and $\Omega_j \in H_j$ is the domain it is inserted into.

Proof. If $v \notin P$ then Ω_i is the unique domain such that $v \in M_{\Omega_i}$. However, $P_{j\Omega_j} \subseteq P_{i\Omega_i}$. Thus, $\mathbf{f}_{P_i}^{\Omega_i}(v) \leq \mathbf{f}_{P_j}^{\Omega_j}(v)$ for non-input points.

If $v \in P$, then it is possible for v to be in more than one M_Ω. Clearly, the input feature size of v will be maximized at the highest domain in the hierarchy that contains it, i.e. at the largest scale. At time j, this domain is Ω_j Just as with the Steiner point case, any changes to this highest level domain do not eliminate any input points and therefore $\mathbf{f}_{P_i}^{\Omega_i}(v) \leq \mathbf{f}_{P_j}^{\Omega_j}(v)$ in this case as well. □

Lemma 45. *Let M be a τ-quality hierarchical mesh constructed incrementally. Given two vertices $u, v \in M$, if u was inserted before v then $\lambda_v \leq \frac{|u-v|}{1-\varepsilon}$.*

Proof. Let i be the insertion time of v and let $\Omega \in H_i$ be the domain it was inserted into. So, $\lambda_v = \mathbf{f}_{M_i}^{\Omega}(v)$. If either $u \in M_{i\Omega}$ or $u \notin B_\Omega$ then $\lambda_v \leq |u-v|$. So we may assume $u \in B_\Omega \setminus M'_\Omega$ and thus for some $\Omega' \in$ children(Ω), $u \in B_{\Omega'}$. Since v does not encroach on Ω', we have that $|c_{\Omega'} - v| \leq \frac{|u-v|}{1-\varepsilon}$. Moreover, $c_{\Omega'} \in M_\Omega$, so $\lambda_v \leq |c_{\Omega'} - v| \leq \frac{|u-v|}{1-\varepsilon}$ as desired. □

Lemma (Lemma 6)**.** *If M is a hierarchical mesh constructed incrementally that satisfies the insertion radius invariant, then M also satisfies the feature size invariant.*

Proof. Let Ω be any domain and let v be any vertex of M_Ω. Let u be the nearest neighbor of v in M_Ω. This implies that $\mathbf{f}_M^{\Omega}(v) = |u-v|$ and so it will suffice to prove $\mathbf{f}_P^{\Omega}(v) \leq K_v|u-v|$. Let i (respectively j) be the insertion time of v (respectively u) and let Ω_i (Ω_j respectively) be the domain it was inserted into. Let $K'_v \in \{K'_C, K'_S, K'_I\}$ be the appropriate constant depending on how v was inserted and similarly for K'_u. We choose K so that

$$K \geq \frac{\max\{K'_C, K'_S, K'_I\}}{1-\varepsilon} + 1 \tag{28}$$

There are two cases to consider.

Case: u inserted before v:

$$\begin{aligned}
\mathbf{f}_P^{\Omega}(v) &\le \mathbf{f}_{P_i}^{\Omega_i}(v) && \text{[by Lemma 44]}\\
&\le K'_v \lambda_v && \text{[by assumption]}\\
&\le K'_v \frac{|u-v|}{1-\varepsilon} && \text{[by Lemma 45]}\\
&< K|u-v|. && \text{[by (28)]}
\end{aligned}$$

Case: v inserted before u:

$$\begin{aligned}
\mathbf{f}_P^{\Omega}(v) &\le \mathbf{f}_P^{\Omega}(u) + |u-v| && [\mathbf{f}_P^{\Omega} \text{ is 1-Lipschitz}]\\
&\le \mathbf{f}_{P_j}^{\Omega_j}(u) + |u-v| && \text{[by Lemma 44]}\\
&\le K'_u \lambda_u + |u-v| && \text{[by assumption]}\\
&\le K'_u \frac{|u-v|}{1-\varepsilon} + |u-v| && \text{[by Lemma 45]}\\
&\le K|u-v|. && \text{[by (28)]}
\end{aligned}$$

□

Lemma 46. *Let Ω be any domain in the output. There exists an ordering $p_1 \dots, p_j$ of the vertices of P_Ω such that for each $i = 3 \dots j$, $\mathbf{f}_{P_i}^{\Omega}(p_i)/\mathbf{f}_{P_{i-1}}^{\Omega}(p_i) \ge \frac{12}{\varepsilon^3} + 1$, where $P_i = \{p_1, \dots, p_i\}$.*

Proof. The desired ordering can be found by starting with the two farthest points of P_Ω as p_1 and p_2 followed by greedily adding any point that satisfies the desired property. Suppose this process gets stuck after adding i points and some $j - i$ points are leftover. Let $p \in P_i$ be such that some leftover point lies in $\mathrm{Vor}_{P_i}(p)$ and let q be the farthest such point from p. Let p' be the nearest neighbor of p in P_i. Let $r = \mathbf{f}_{P_i^{\Omega}}(q) = |p - q|$. Since the ordering was stuck, it must be that $|q - p'|/|q - p| \ge \frac{12}{\varepsilon^3}+$. By the triangle inequality, $|p' - p| \ge |p' - q| - |p - q|$. Combining the previous two statements give that $|p' - p| \ge \frac{12r}{\varepsilon^3}$. By our choice of p' as the nearest neighbor of p, we get that $\mathrm{annulus}(p, 2r, \frac{6r}{\varepsilon^3})$ is contained in $\mathrm{Vor}_{P_i}(p)$. Moreover, by our choice of q, this annulus is empty of points from P_Ω, contradicting Lemma 8. □

Lemma 47. *Let q and q' be any two input points and let r be the distance between them. If $A = \mathrm{annulus}(q, 2r, \frac{6r}{\varepsilon^3})$ contains no input points, then q and q' are inside some cage contained in A for all intermediate meshes after each has been inserted.*

Proof. Let $p_1, \dots, p_k$ be all input points in $\mathrm{ball}(p, 2r)$ ordered by the order in which they were inserted. Clearly q and q' are among the p_i's. We will proceed by induction on k. In the base case, there are only two points, p_1 and p_2. Let Ω be the domain into which p_2 was inserted and P' and M' be the input and mesh vertices in the domain just after insertion. Since A contains no input points and $\mathbf{f}_P^{\Omega}$ is Lipschitz, $\mathbf{f}_P^{\Omega}(p_1) \ge \frac{6r}{\varepsilon^3} - 2r \ge \frac{4r}{\varepsilon^3}$. So, by Lemma 7, $\mathbf{f}_M^{\Omega}(p_1) \ge \frac{4r}{K\varepsilon^3}$. Since the algorithm chooses $\varepsilon < \frac{1}{K}$, we have that $\mathbf{f}_M^{\Omega}(p_1) \ge \frac{4r}{\varepsilon^2}$ and therefore, using the Lipschitz property, $\mathbf{f}_M^{\Omega}(p_2) \ge \mathbf{f}_M^{\Omega}(p_1) - |p_1 - p_2| \ge 4r(\frac{1}{\varepsilon^2} - 1)$. After insertion, we have that $\mathbf{f}_{M'}^{\Omega}(p_2) \le 4r$, because $p_1 \in M'_\Omega$. So, the ratio of nearest to second nearest neighbor distances for p_2 is bounded as

$$\frac{\mathbf{f}_{M'}^{\Omega}(p_2)}{\mathbf{f}_M^{\Omega}(p_2)} \le \frac{4r}{4r(\frac{1}{\varepsilon^2} - 1)} < \varepsilon.$$

Thus, the algorithm adds a cage around p_1 and p_2 as desired.

The inductive step has two cases. Either the ith point is added inside or outside the cage surrounding the first $i-1$ points (guaranteed to exist by induction). The latter case is identical to the base case, so it only remains to consider the case where p_i lies inside the cage from the previous round. If p_i does not encroach this cage, then it remains and we are done. If p_i does encroach this cage, then it will grow by a factor of $\frac{1}{\varepsilon}$. However, it's total size cannot exceed $\frac{1}{\varepsilon^2}$ times the distance from p_i to the center because otherwise it would not encroach. This distance is at most $4r$, so the cage is in A. Thus, the grown cage also satisfies the induction hypothesis and we are done. ☐

C Technical lemmas regarding the feature size function

The following lemma extends the feature size invariant to mesh vertices in $B_\Omega \setminus M_\Omega$.

Lemma 48. *If (M, H) is a hierarchical mesh of an input set P such that the feature size invariant holds and no domain is ε-encroached then for all domains $\Omega \in H$ then for all $u \in M \cap B_\Omega$,*

$$\mathbf{f}_P^\Omega(u) \leq \alpha \mathbf{f}_M^\Omega(u),$$

where α is a constant depending only on the meshing parameters.

Proof. Fix a domain Ω and a vertex $u \in M \cap B_\Omega$. If $u \in M_\Omega$, then the feature size invariant implies the desired result. So, we may assume that $u \notin M_\Omega$. Let v be the nearest neighbor of u in M_Ω. Since u does not ε-encroach on any domains, we have that

$$|u - v| \leq \varepsilon \mathbf{f}_M^\Omega(v). \tag{29}$$

So,

$$\mathbf{f}_M^\Omega(v) \leq \frac{1}{1-\varepsilon} f_M^\Omega(u), \tag{30}$$

because $\mathbf{f}_M^\Omega$ is Lipschitz.

$$\begin{aligned}
\mathbf{f}_P^\Omega(u) &\leq \mathbf{f}_P^\Omega(v) + |u - v| && \left[\mathbf{f}_P^\Omega \text{ is 1-Lipschitz}\right] \\
&\leq K\mathbf{f}_M^\Omega(v) + |u - v| && \text{[by the feature size invariant]} \\
&\leq (K + \varepsilon)\mathbf{f}_M^\Omega(v) && \text{[by (29)]} \\
&\leq \frac{K+\varepsilon}{1-\varepsilon}\mathbf{f}_M^\Omega(u). && \text{[by (30)]}
\end{aligned}$$

☐

For quality meshes, it is possible to extend the feature size invariant to all points in the plane.

Lemma 49. *If M is a τ'-quality mesh of an input set P such that the feature size invariant holds and no domain is ε-encroached then for all domains $\Omega \in H_M$ then for all $x \in B_\Omega$,*

$$\mathbf{f}_P^\Omega(u) \leq \beta \mathbf{f}_M^\Omega(u),$$

where β is a constant depending only on the meshing parameters.

Proof. Fix a domain Ω and a point $x \in B_\Omega$. Let u be the nearest neighbor of x in M. Observe that $u \in B_\Omega$. So by the Lipschitz property of $\mathbf{f}_P^\Omega$ and Lemma 48,

$$\mathbf{f}_P^\Omega(x) = \alpha \mathbf{f}_M^\Omega(u) + |u - x|.$$

Because $\mathbf{f}_M^\Omega$ is also 1-Lipschitz,

$$\mathbf{f}_P^\Omega(x) = \alpha \mathbf{f}_M^\Omega(x) + (1+\alpha)|u - x|.$$

Because we chose u to be the nearest neighbor, the diametral ball of u and x is empty and centered in B_Ω. Thus, Lemma 14 implies that

$$|u - x| \leq 24\tau'^2 \mathbf{f}_M^\Omega(x).$$

So, choosing $\beta = \alpha + (1+\alpha)24\tau'^2$ we have that $\mathbf{f}_P^\Omega(x) \leq \beta \mathbf{f}_M^\Omega(x)$ as desired. □

D Technical Lemmas for the Point Location Analysis

Lemma 50. *For all $i \in I$ and $x \in U_i$, $f_0(x) \leq c_{50} f_i(x)$, where $c_{50} = \frac{1}{1-e}$.*

Proof. Fix an index $i \in I$ and a point $x \in U_i$. Suppose for contradiction that $f_i(x) < (1-e)f_0(x)$. We will show that $\mathbf{d}_{\mathcal{B}_M}(x, p_i) \leq 5$, contradicting the hypothesis that $i \in I$.

Recall that for $x \in U_i$, $f_i(x) \neq f_{i-1}(x)$. Combined with our supposition that $f_i(x) < (1-e)f_0(x)$, this implies that there can be at most one point of M_Ω in $\mathrm{ball}(x, \frac{1}{1-e}|x - p_i|)$. Call this point z and let Z denote the set of points of M whose ancestor in M_Ω is z.

We will construct a chain of balls $B_1, \ldots, B_6$ from p_i to x. To do this, the following claim is useful.

If $y \in \mathrm{ball}(x, |x - p_i|)$ does not encroach any domain of $H_{M'}$ then $\mathrm{ball}\left(\frac{x+y}{2}, \frac{|x-y|}{2}\right) \cap M \subseteq Z$.

We will give the construction using this claim and then give its proof.

Let B_1 be the maximal ball of $\mathcal{B}_M$ tangent to p_i centered on $\overline{xp_i}$. If $x \in B_1$, then $\mathbf{d}_{\mathcal{B}_M}(x, p_1) \leq 1$, so we may assume that $x \notin B_1$ and therefore $B_1 \subset \mathrm{ball}(x, |x - p_i|)$. So the claim implies that some vertex of Z is on the boundary of B_1.

Let B_4 be the maximal ball of $\mathcal{B}_M$ tangent to x centered on $\overline{xz}$. As with B_1, we may assume that $p_i \notin B_4$ and thus the claim implies some vertex of Z is on the boundary of B_4.

If $Z = \{z\}$ then $z \in \partial(B_1) \cap \partial(B_4)$ and thus $\mathbf{d}_{\mathcal{B}_M}(x, p_i) \leq 1$. So, we may assume that there is some domain $\Omega' \in H_{M'}$ centered at z whose parent is Ω. Let r' be the radius of Ω'. Since Ω' is not encroached and $\varepsilon < \frac{1}{3}$, $\mathrm{annulus}(z, r', 3r')$ is empty of points of M. Any two points a, b of such an annulus have $\mathbf{d}_{\mathcal{B}_M}(a, b) \leq 3$. This is easily seen by considering the plane through a, b, and z and packing the annulus with balls of radius r' as in Figure 3. Since both B_1 and B_6 intersect the annulus, we may choose $a \in B_1$ and $b \in B_6$ and let $B_2, \ldots, B_5$ be the balls packing the annulus on the shortest path from a to b. The balls $B_1, \ldots, B_6$ witness that $\mathbf{d}_{\mathcal{B}_M}(x, p_i) \leq 5$ as desired.

To conclude the proof, we need only prove the claim. Let $B' = \mathrm{ball}(c, r)$ be the ball under consideration where $c = \frac{x+y}{2}$ and $r = \frac{|x-y|}{2}$. Suppose for contradiction that there exists $w \in B' \cap (M \setminus Z)$. Then, $|c - w| < r = |x - c|$. Let v be the ancestor of w in M_Ω and note that $v \neq z$. Observe that $|x - y| \leq (1-\varepsilon)|x - v|$ for otherwise $f_0(x) \leq |x - v| \leq \frac{1}{1-\varepsilon} f_i(x)$ contrary to our supposition. It follows that

$$|x - v| \geq \frac{2\varepsilon r}{1 - \varepsilon}. \tag{31}$$

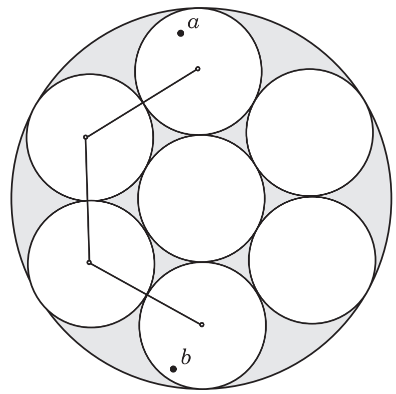

Figure 3: Any two points $a, b \in \text{annulus}(z, r', 3r')$ have $\mathbf{d}_{\mathcal{B}_M}(a,b) \leq 3$.

We can now bound $|v - w|$ in terms of r as follows.

$$\begin{aligned}
|v-w| &\leq \varepsilon|y-v| && [y \text{ does not encroach}] \\
&\leq \varepsilon(|y-c| + |c-w| + |w-v|) && [\text{by the triangle inequality}] \\
&< \frac{2\varepsilon r}{1-\varepsilon}. && [|y-c| = r \text{ and } |c-w| < r] \qquad (32)
\end{aligned}$$

This allows us to derive the following contradiction.

$$\begin{aligned}
|c-w| &\geq |x-v| + |v-w| - |c-x| && [\text{by the triangle inequality}] \\
&> \left(\frac{2r}{1-\varepsilon} - \frac{2\varepsilon}{1-\varepsilon} - 1\right) r && [\text{by (31) and (32)}] \\
&= r.
\end{aligned}$$

□

Lemma 51. $\sum_{i\in I}(h_i - h_{i-1}) \leq (c_{51} - 1)h_0$.

Proof. First, we bound h_{jj} using Lemma 50 as follows.

$$h_{jj} = \int_{U_j} \frac{dx}{f_j(x)^d} \leq c_{50}^d \int_{U_j} \frac{dx}{f_0(x)^d} = c_{50}^d h_0. \qquad (33)$$

For any $i \in I$, define i^* to be the largest element of I_0 less than i. If $i, j \in I$ and $i > j$, then $h_{ij} - h_{(i-1)j} = 0$ as guaranteed by the definition of U_j. Because $f_i \leq f_{i'}$ for all $i \leq i'$, $h_{(i-1)j} \geq h_{i^*j}$ for all i. The desired bound is now proven as follows.

$$\begin{aligned}
\sum_{i\in I}(h_i - h_{i-1}) &\leq \sum_{j\in I_0}\sum_{i\in I: i\leq j}(h_{ij} - h_{(i-1)j}) \\
&\leq \sum_{j\in I_0}\sum_{i\in I: i\leq j}(h_{ij} - h_{i^*j}) && [i^* \leq i-1] \\
&= \sum_{j\in I_0}(h_{jj} - h_{0j}) \\
&\leq \sum_{j\in I_0}(c_{50}^d - 1)h_{0j} && [\text{by (33)}] \\
&= (c_{50}^d - 1)h_0.
\end{aligned}$$

Choosing $c_{51} = c_{50}^d$ completes the proof. □

Lemma 52. $h_0 \leq c_{52}$.

Proof. There are four types of domains to consider: the smallest domain Ω such that $B \subset B_\Omega$, domains Ω such that $|M_\Omega| = 0$, domains Ω such that $|M_\Omega| = 1$, and domains Ω such that M_Ω contains an entire cage $C_{\Omega'}$ of some domain $\Omega' \in H_M$. In the first case, the result follows easily from Lemma 14. In the second case, $\mathbf{f}_M^\Omega = \infty$, and thus, the integral evaluates to 0. In the third case, it is easy to evaluate the integral directly using polar coordinates to find that it is constant.

The last case is the interesting one. We use the coarse bounds that $\mathbf{f}_M^\Omega(x) \geq \delta r_{\Omega'}$ for $(1 - \delta - \gamma) r_{\Omega'} \leq |x - c_\Omega| \leq 2r_{\Omega'}$ and $\mathbf{f}_M^\Omega(x) \geq \frac{1}{2}|x - c_\Omega|$ for $|x - c_\Omega| > 2r_{\Omega'}$. Integrating with polar coordinates centered at c_Ω yields an answer $O(\log \frac{r_\Omega}{r_{\Omega'}})$. Only a constant number of points in a round may cause Ω' to grow because all but one must lie in the Voronoi cell of C_Ω and thus they are all within a constant D-ball distance of one another. So, $\frac{r_\Omega}{r_{\Omega'}} = O(1)$ and thus the integral also evaluates to $O(1)$. ∎