ABSTRACT

Title of dissertation: ADAPTIVE SAMPLING FOR GEOMETRIC APPROXIMATION

Ahmed Abdelkader Abdelrazek, Doctor of Philosophy, 2020

Dissertation directed by: Professor David M. Mount, Computer Science

Geometric approximation of multi-dimensional data sets is an essential algorithmic component for applications in machine learning, computer graphics, and scientific computing. This dissertation promotes an algorithmic sampling methodology for a number of fundamental approximation problems in computational geometry. For each problem, the proposed sampling technique is carefully adapted to the geometry of the input data and the functions to be approximated. In particular, we study proximity queries in spaces of constant dimension and mesh generation in 3D.

We start with polytope membership queries, where query points are tested for inclusion in a convex polytope. Trading-off accuracy for efficiency, we tolerate one-sided errors for points within an ε-expansion of the polytope. We propose a sampling strategy for the placement of covering ellipsoids sensitive to the local shape of the polytope. The key insight is to realize the samples as Delone sets in the intrinsic Hilbert metric. Using this intrinsic formulation, we considerably simplify state-of-the-art techniques yielding an intuitive and optimal data structure.

Next, we study nearest-neighbor queries which retrieve the most similar data point to a given query point. To accommodate more general measures of similarity, we consider non-Euclidean distances including convex distance functions and Bregman divergences. Again, we tolerate multiplicative errors retrieving any point no farther than $(1 + \varepsilon)$ times the distance to the nearest neighbor. We propose a sampling strategy sensitive to the local distribution of points and the gradient of the distance functions. Combined with a careful regularization of the distance minimizers, we obtain a generalized data structure that essentially matches state-of-the-art results specific to the Euclidean distance.

Finally, we investigate the generation of Voronoi meshes, where a given domain is decomposed into Voronoi cells as desired for a number of important solvers in computational fluid dynamics. The challenge is to arrange the cells near the boundary to yield an accurate surface approximation without sacrificing quality. We propose a sampling algorithm for the placement of seeds to induce a boundary-conforming Voronoi mesh of the correct topology, with a careful treatment of sharp and non-manifold features. The proposed algorithm achieves significant quality improvements over state-of-the-art polyhedral meshing based on clipped Voronoi cells.

ADAPTIVE SAMPLING FOR GEOMETRIC APPROXIMATION

by

Ahmed Abdelkader Abdelrazek

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2020

Advisory Committee:
Professor David M. Mount: *Advisor, Chair*
Professor William Goldman: *Dean's Representative*
Professor Samir Khuller
Professor Ramani Duraiswami
Professor MohammadTaghi Hajiaghayi
Professor Nina Amenta: *External Member*

To my parents: Nashwa and Abdelkader.

Acknowledgments

I am very grateful to the many wonderful people I met through the journey of my PhD, and all the experiences and conversations we shared together.

I am fortunate to have had David Mount as my advisor. As an advisor, Dave always supported me and let me do my thing. As a colleague, Dave has been available to listen providing his insights and good-natured humor. Beyond our long technical discussions, I learned a lot from Dave's dedication to teaching and service. (Unfortunately though, after all this time as Dave's advisee, I still have not learned much about tennis.)

Even before I met Dave, I was fortunate to have Mohamed Ebeida as mentor. Mohamed encouraged me to pursue my interest in geometry and provided me with unique opportunities to work on some very challenging problems. By working with Mohamed, I learned about many exciting works in geometric modeling and computer graphics which had a major influence on the way I do geometry. Mohamed was particularly patient and always provided me with invaluable advice.

Through my connection to Mohamed, I was fortunate to go on multiple research visits at Sandia National Labs. I am grateful to Scott Mitchell for welcoming me to Albuquerque upon my first visit back in 2015. I greatly enjoyed working closely with Scott, especially as we put together what became my first SoCG paper which is now part of this dissertation. I am also grateful to Cynthia Phillips, Randy Smith, Daniel Turner, James Stewart, and Michael Parks for helping with my visits and supporting me as I juggled multiple projects.

I am grateful to my committee members for all their support and encouragement. I am grateful to Samir Khuller and Ramani Duraiswami for serving on my preliminary examination committee back in 2018, and I am grateful to William Goldman, Nina Amenta, and MohammadTaghi Hajiaghayi for joining the examination committee.

I am fortunate to have had a chance to work with a large group of collaborators and co-authors. I am grateful to Chandrajit Bajaj for supporting me as I started learning about surface reconstruction and for inviting me to visit his lab at UT Austin. I am grateful to John Owens for always being available to provide valuable feedback and insightful remarks. I learned a lot from John about paper writing and the review process, which helped me greatly with the papers we wrote together and continues to influence my writing. I am grateful to Ahmed Mahmoud for his critical help with the VoroCrust project. I am grateful to Philip Dasler and Aditya Acharya for sharing the joys of the work that became my first hardness result. I am grateful to Ping-yeh Chiang for mentoring me on how to become a better mentor. I am also grateful to Sunil Arya, Guilherme da Fonseca, Khaled Harras, Drew Kouri, Amr Mohamed, Ojas Parekh, Ahmed Rushdi, and Nickolas Winovich.

I am grateful to the professors and staff members at the department. William Gasarch was particularly patient with me as a troublesome student in his computational complexity course, and gave me some critical advice that I only came to accept years later. I also enjoyed taking classes with Hal Daumé, David Jacobs, and Aravind Srinivasan, and I am grateful for their continued support years later. I am grateful to Tom Goldstein, Matthias Zwicker, and Ming Lin for their critical support over the past year and for providing valuable advice as I navigated the job

market. I am grateful to Yiannis Aloimonos and John Dickerson for being warm and supportive colleagues. Last but not least, I am grateful to the wonderful staff members for their patience and support helping with all the administrative work: Jennifer Story, Sharron McElroy, Jodie Gray, Adelaide Findlay, Tom Hurst, and Regis Boykin.

I am fortunate to have been welcomed to the computational geometry community early on in my PhD. I am grateful to the support and encouragement I received from Joe Mitchell, John Hershberger, Jack Snoeyink, Kenneth Clarkson, Jeff Erickson, Bettina Speckmann, Sándor Fekete, Boris Aronov, and Xavier Goaoc. On the computational topology side, I am grateful to Robert Ghrist, Don Sheehy, Ulrich Bauer, Michael Lesnick, Tamal Dey, Yusu Wang, Herbert Edelsbrunner, Michael Kerber, and Steve Oudot. I am particularly grateful to Brittany Fasy and David Millman for their continued support and encouragement. I am grateful to all the funding, mostly through NSF, that supported my trips to the conferences and workshops where I had a chance to meet and get to know those wonderful people.

Outside the technical realm, I am grateful to my landlady Charlotte Pappas for hosting me at her lovely home for the better part of my PhD. I am also grateful to the wonderful friends I made during my visits to Albuquerque: Tatiana Minot, Vicky and Jeff Kauffman, Susi and Charlie Knoblauch, and Leila Salim.

Finally, I am grateful to the friends who kept me company through this journey: Sameh Eisa, Mohamed Elsabagh, Hamdy Elgammal, Pauline Ezan, Federico Iuricich, Gregory Kramida, Ahmed Mamdouh, Ahmed Saeed, Alejandro Flores Velazco, and all the good Pineway people.

Table of Contents

Chapter 1: Introduction

A predominant theme in geometric computing is the decomposition of geometric domains into a discrete set of simple pieces that are easy to process. At a high level, this can be seen as a multi-dimensional analogue to the use of finite-precision arithmetic to approximate computations over the reals. Indeed, it is often the case that such discrete decompositions may only approximate the original geometry. It is then imperative to trade-off acceptable degradations in accuracy against a computational budget. Using the analogy of digital arithmetic, single-precision floating points may suffice for a range of calculations, while others require double or even arbitrary precision.

Depending on the context, the required decompositions can take on different forms. For example, the indexing of multi-dimensional data typically utilizes a decomposition of space, whereas the digital representation of a 3D model typically takes the form of a surface mesh. In order to achieve efficiency, it is often necessary to adapt the decomposition to the instance at hand, that is, to the distribution of data points or the shape of the model.

Over the past few decades, different research communities have developed a variety of decomposition and approximation techniques. While these techniques utilize different mathematical formulations and prioritize different objectives, they actually have a lot in common.

This dissertation offers a reconciliation of a number of related themes in geometric approximation. This is based on employing *adaptive sampling* as the unifying paradigm. In particular, we develop sampling methods that capture the relevant features of the underlying geometry while providing a suitable trade-off in accuracy against processing cost.

Through a combination of sampling techniques from geometry processing and analysis techniques from algorithm theory, we obtain a number of results demonstrating the benefits of the proposed algorithmic sampling methodology. We apply our sampling methodology to the following problems: (1) proximity search with point sets and polytopes in multi-dimensional spaces, and (2) mesh generation in 3D. For each problem, the proposed sampling technique is carefully adapted to the geometry of the input data and the functions to be approximated.

In the remainder of this introduction, we briefly overview the problems we study and summarize the contributions of the dissertation. In doing so, we further elaborate on the different aspects of the proposed algorithmic sampling methodology to be developed in the remainder of the dissertation.

1.1 Polytope Membership Queries

Convex bodies are ubiquitous in computational geometry and optimization theory. Specifically, we consider polytopes represented as the intersection of n

half-spaces in $\mathbb{R}^d$. The high combinatorial complexity of multidimensional convex polytopes has motivated the development of algorithms and data structures for approximate representations.

In Chapter 3, we demonstrate an intriguing connection between convex approximation and the classical concept of Delone sets from the theory of metric spaces. We show that with the help of a classical structure from convexity theory, called the Macbeath region, it is possible to construct an ε-approximation of any convex body as the union of $O(1/\varepsilon^{(d-1)/2})$ ellipsoids, where the center points of these ellipsoids form a Delone set in the Hilbert metric associated with the convex body.

Using the proposed approximation based on ellipsoid covers, we design a data structure that answers ε-approximate polytope membership queries in $O(\log(1/\varepsilon))$ time. This matches the best asymptotic results for this problem, by a data structure that both is simpler and arguably more elegant.

This first application clearly demonstrates the main ingredients of the proposed sampling methodology. By working in the Hilbert metric intrinsic to the polytope, we obtain a sufficient sampling criteria as a Delone set with local approximations provided by shape-sensitive ellipsoids. Compared to state-of-the-art results that also utilized Macbeath regions, the intrinsic formulation greatly simplifies the analysis of the resulting data structure.

1.2 Nearest-Neighbor Search Queries

Nearest-neighbor searching involves indexing a set of n points from a metric space into a data structure such that the nearest neighbor to a given query point can be retrieved efficiently. In order to achieve efficiency in terms of storage and query time, we consider the problem in an approximate setting, where we retrieve any point whose distance is no farther than $(1 + \varepsilon)$ times the distance to the true nearest neighbor.

In Chapter 4, we present a new approach to ε-approximate nearest-neighbor queries in fixed dimension d under a variety of non-Euclidean distances. In particular, we consider two families of distance functions: (a) convex scaling distance functions including the Mahalanobis distance, the Minkowski metric and multiplicative weights, and (b) Bregman divergences including the Kullback-Leibler divergence and the Itakura-Saito distance.

Under mild assumptions on the distance functions, we propose a sampling strategy that adapts the sampling density to their growth rates in addition to the local distribution of data points. This enables a generalized data structure that answers queries in logarithmic time using $O(n \log(1/\varepsilon)/\varepsilon^{d/2})$ space, which nearly matches the best known results for the Euclidean metric.

A crucial ingredient to the efficiency of the proposed data structure is a careful application of *convexification*, which appears to be relatively new to computational geometry. The proposed convexification successfully circumvents the reliance on the *lifting transform*, which has been essential in the fastest state-of-the-art data structures.

This second application demonstrates the treatment of both shape and function constraints within our sampling methodology. This is a recurring scenario in geometry processing applications that deal with different types of differential equations, e.g., fluid flows and elasticity. In contrast, the consideration of of non-Euclidean distances and their differential properties has not received much attention in the computational geometry community. This further underscores the potential benefits of exploiting these connections as facilitated by the proposed unification through sampling.

1.3 Voronoi Mesh Generation

The computational modeling of physical phenomena requires robust numerical algorithms and compatible high-quality domain discretizations. Finite element methods traditionally use simplicial meshes, where well-known angle conditions prohibit skinny elements. The limited degrees of freedom of linear tetrahedral elements often lead to excessive refinement when modeling complex geometries or domains undergoing large deformations. This motivated generalizations to general polyhedral elements, which enjoy larger degrees of freedom and have recently been in increasing demand.

In the second half of this dissertation, we study the problem of decomposing a volume bounded by a piecewise-smooth surface into a collection of Voronoi cells, a particularly attractive class of polyhedral cells. The proposed scheme, called *VoroCrust*, leverages ideas from α-shapes and the power crust algorithm to produce unweighted Voronoi cells conforming to the surface. The scheme is based on a suitable sampling of the surface, which is used to define a union balls of balls with radii proportional to the feature size. The corners of this union of balls are the Voronoi sites, on both sides of the surface, and the facets common to cells on opposite sides reconstruct the surface.

In Chapter 5, we start by assuming the surface is a smooth manifold with a known local feature size. We derive sufficient conditions on the sampling to guarantee an isotopic surface reconstruction. In addition, we describe a simple approach to further decompose the enclosed volume into a volumetric mesh of fat Voronoi cells with a suitable bound on the number of cells.

Then, Chapter 6 presents the design and analysis of a robust implementation of VoroCrust that can handle realistic 3D models. The crux of the algorithm is a refinement process that estimates a suitable sizing function to guide the placement of Voronoi seeds. This enables VoroCrust to protect all sharp features, and mesh the surface and interior into quality elements. The algorithm carefully handles non-manifold features and successfully eliminates undesired slivers on the surface. The quality of the produced meshes is demonstrated through a variety of challenging models, establishing clear advantages over state-of-the-art polyhedral meshing methods based on clipped Voronoi cells.

In this third application, we demonstrate a two-fold approach to designing geometric algorithms, which is both robust and practical within our sampling methodology. In particular, sliver elimination is widely recognized as a challenging problem, and known analyses are rather intricate with pessimistically-weak guarantees of marginal value in practice. Our two-fold approach is as follows. We start by proving termination with a relaxed sampling criterion that tolerates a limited deterioration in quality. Then, we provide a novel probabilistic analysis of termination with the strict sampling criterion by borrowing ideas from the analysis of randomized algorithms. The proposed implementation combines the two criteria to guarantee termination in practice, while ensuring a strong guarantee on quality. The novel use of probabilistic reasoning in this context underscores the potential benefits of a sampling methodology with strong algorithmic aspects.

Chapter 2: Literature Review

We review the most relevant related work on each of the problems we consider in the dissertation.

2.1 Approximate Polytope Representations

We review the related work on the efficient representation of convex polytopes as pertains to membership testing. Let $K \subseteq \mathbb{R}^d$ denote a convex polytope given as the intersection of n halfspaces. Throughout, we assume that the dimension d is a fixed constant and that K is full dimensional and bounded.

The *polytope membership problem* is that of preprocessing K so that it is possible to determine efficiently whether a given query point $q \in \mathbb{R}^d$ lies within K. Polytope membership queries, both exact and approximate, arise in many application areas, such as linear programming and ray-shooting queries [1–4], nearest-neighbor searching and the computation of extreme points [5–7], collision detection [8], and machine learning [9].

We summarize prior work on polytope membership as follows. In Section 2.1.1, we motivate the study of approximate representations by reviewing classical results from *exact range queries*. Then, we review related work on *approximating polytopes* in Section 2.1.2, as may be used for membership testing. Finally, we review state-of-the-art results on approximate membership queries in Section 2.1.3. Later in Chapter 3, we apply our sampling methodology to obtain a simplified data structure matching state-of-the-art results.

2.1.1 Exact Membership Queries

To gain insight into the membership testing problem, we consider an equivalent problem in the dual setting. It turns out that polytope membership is equivalent to answering halfspace emptiness queries for a set of n points in $\mathbb{R}^d$. When the dimension d is small, i.e., $d \leq 3$, it is possible to build a data structure of linear size to answer such queries in logarithmic time [10, 11]. For higher values of d, however, the fastest data structures with near-linear space have a query time of roughly $O\big(n^{1-1/\lfloor d/2 \rfloor}\big)$ [12], which can be prohibitively expensive in practice.

Another closely related problem is polytope intersection queries [11, 13, 14], which can be considered as a general version of polytope membership queries. Barba and Langerman [14] showed how to preprocess polytopes in $\mathbb{R}^d$, treating d as a constant, so that given two such polytopes, it can be determined whether they intersect each other. As expected, the preprocessing time and space required are rather high, growing as the combinatorial complexity of the polytopes (which can be as high as $\Theta(n^{\lfloor d/2 \rfloor})$) raised to the power $\lfloor d/2 \rfloor$.

2.1.2 Approximating Polytopes

The study of general convex sets motivated the following interesting problem. It asks to compute a convex polytope P to approximate a given closed convex set $K \subseteq \mathbb{R}^d$. Assuming K is normalized to have unit diameter, it is required that the *Hausdorff distance* between P and K is at most a given error threshold $\varepsilon > 0$. In addition, the polytope P is required to have *low combinatorial complexity*, which is the total number of faces of all dimension. We call such a polytope an *ε-approximating polytope.*

Known bounds on the complexity of ε-approximating polytope are of two types. *Non-uniform bounds* there is an ε_0, depending on K (for example, its maximum curvature), allowing a bound on the complexity of ε-approximating polytopes with $\varepsilon \leq \varepsilon_0$. Such bounds often hold in the limit as ε tends to 0, or equivalently as the complexity of the approximating polytope tends to infinity [15–18]. The other types of *uniform bounds* are usually stated for an ε_0 that does not depend on K. For subsequent algorithmic applications of ε-approximating polytopes, it is convenient to apply the approximation as a black-box without further dependencies on the properties of the inputs. As such, we focus on uniform bounds.

Dudley [19] showed that, for any convex body K in $\mathbb{R}^d$, it is possible to construct an ε-approximating polytope P with $O(1/\varepsilon^{(d-1)/2})$ facets. This bound is asymptotically tight in the worst case, even when K is a Euclidean ball. This construction implies a (trivial) data structure for approximate polytope membership problem with space and query time $O(1/\varepsilon^{(d-1)/2})$. In this connection, Bronshteyn and Ivanov obtained the same bound for the number of vertices, which is also the best possible [20].

Despite these bounds on the number of facets or the number of vertices, this falls short of bounding the total combinatorial complexity. The upper-bound theorem by McMullen [21,22] bounds the complexity of a polytope with n facets or vertices by $O(n^{\lfloor d/2 \rfloor})$. Known classes of pathological polytopes, e.g., the cyclic polytope, realize this upper bound [23]. As such, a direct application of the upper-bound theorem to the polytopes constructed by Dudley or Bronshteyn-Ivanov yields a weak upper bound of roughly $O(1/\varepsilon^{(d^2-d)/4})$ on the complexity of ε-approximating polytopes. However, given the special structure of the pathological polytopes achieving the worst-case bounds from the upper-bound theorem, it is plausible to expect ε-approximating polytopes to achieve lower complexities by exploiting the extra tolerance available.

In a series of papers, Arya et al. [24–29] were finally able to present a construction of an *ε-approximating polytope* matching the bounds Dudley and Bronshteyn-Ivanov. Their construction makes use of a width-based variant of *economic cap covers* [30] to approximate the boundary of the polytope in layers. Then, they bound the total combinatorial complexity of the facets using the witness-collector technique [31].

2.1.3 Approximate Membership Queries

The review above demonstrates a large gap between the high computational overhead of exact membership testing and the succinct representations available through approximating polytopes. This has motivated the study of approximate membership queries.

To quantify the approximation errors, we introduce the real parameter $\varepsilon > 0$, where errors are measured relative to the diameter of K, denoted by $\mathrm{diam}(K)$. Given a query point $q \in \mathbb{R}^d$, an *ε-approximate polytope membership query* returns `True` if $q \in K$, `False` if the distance from q to its closest point in K is greater than $\varepsilon \cdot \mathrm{diam}(K)$, and it may return either result otherwise.

A simple approximation scheme was proposed by Bentley *et al.* [32]. First, a d-dimensional grid with cells of diameter $\Theta(\varepsilon \cdot \mathrm{diam}(K))$ is constructed. Then, for every column along the x_d-axis, the two extreme x_d values where the column intersects K are stored. Given a query point q, it is easy to determine if $q \in P$. The storage required by the approach is $O(1/\varepsilon^{d-1})$.

In follow up work, the grid employed by Bentley *et al.* [32] was replaced with an adaptive subdivision as in the *SplitReduce* data structure of Arya et al. [33]. Given a parameter t, space is subdivided hierarchically using a quadtree until each cell either (1) lies completely inside K, (2) completely outside K, or (3) intersects K's boundary such that it is possible to approximate the portion of the boundary within the cell by at most t halfspaces, against which query points lying in such a cell can be tested. In [33] it is shown that the quadtree height is $O(\log \frac{1}{\varepsilon})$, allowing an overall query time is $O(\log \frac{1}{\varepsilon} + t)$.

While the SplitReduce data structure is conceptually simple, it leaves open the possibility of achieving a query time of $O(\log \frac{1}{\varepsilon})$ with a minimum storage of $O(1/\varepsilon^{(d-1)/2})$. This improved performance was recently achieved by Arya et al. [34], where the novel ingredient was to abandon the quadtree-based approach of [33] and [24] in favor of a hierarchy of ellipsoids. The ellipsoids are chosen through a sampling process inspired by a classical construct from the theory of convexity, called *Macbeath regions* [35]. The main result of [34] is the following.

Theorem 1. *Given a convex polytope K in $\mathbb{R}^d$ and an approximation parameter $0 < \varepsilon \leq 1$, there is a data structure that can answer ε-approximate polytope membership queries with*

$$\textit{Query time: } O\left(\log \frac{1}{\varepsilon}\right) \textit{ and Space: } O\left(\frac{1}{\varepsilon^{(d-1)/2}}\right).$$

The contributions of [34] hint that a more “shape-sensitive” approach potentially achieves dramatic improvements over the space requirements of the data structure. In Chapter 3, we further expand on this idea by working in the intrinsic Hilbert metric, which elucidates the role of the Macbeath regions and enables an intuitive data structure matching the results of [34].

2.2 Nearest-Neighbor Searching

A fundamental computational problem that arises countless times throughout science and engineering is searching a data set for objects which are similar to a given query object. This type of query arises in numerous areas, such as data compression, pattern recognition, clustering, large data analytics, information retrieval and visualization, similarity search in image and video databases, machine learning, geometric network design, and signal processing. These problems are typically handled by modeling objects as points in a metric space and applying nearest-neighbor searching.

The most widely studied metric space is real d-dimensional space, $\mathbb{R}^d$, under the Euclidean metric. While many applications of nearest-neighbor searching involve spaces of high dimension, there are also many applications that reside in relatively low dimensions (say, smaller than 20), and theoretical computer science has played a key role in the development of many of the most widely used data structures today.

We summarize prior work on nearest-neighbor searching as follows. In Section 2.2.1, we motivate the study of approximate representations by reviewing classical results on *exact* nearest-neighbor search. Then, we review approximate nearest-neighbor search under the Euclidean metric in Section 2.2.2, which is most related to our work. Finally, we review related work on nearest-neighbor search under more general metrics in Section 2.2.3. Later in Chapter 4, we apply our sampling methodology to obtain a data structure for nearest-neighbor search under more general metrics with performance matching state-of-the-art results for the Euclidean metric. For related work on nearest-neighbor searching in high dimensions, please refer to the recent survey [36].

2.2.1 Exact Search

Without any data structures, it is straightforward to answer nearest-neighbor queries exactly by simply considering all data points. Clearly, this only takes $O(n)$ time and $O(n)$ storage. In very low dimensions with $d \leq 2$, this can be improved to $O(\log n)$ time still with linear storage using simple techniques like binary search trees and point-location. Unfortunately, for $d > 2$, the computational overhead seem to grow extremely rapidly either in terms of the query time or the storage requirements. Namely, the best solution achieving *logarithmic* query time uses roughly $O(n^{d/2})$ storage space [37], which is too high for many applications. On the other hand, it is possible to keep the storage linear and achieve a barely sublinear query time of $O(n^{f(d)})$, where $f(d) = \frac{1}{d}\left(\log(2^d - 1)\right)$ [38]. However, such limited asymptotic improvements have no real impact in practice.

2.2.2 Approximate Search

This prohibitive computational overhead of exact nearest-neighbor searching motivated the study of approximations. In particular, we aim to achieve logarithmic query times using only linear storage. Given an approximation parameter $\varepsilon > 0$, *ε-approximate nearest-neighbor searching* (ε-ANN) returns any site whose distance from q is within a factor of $1+\varepsilon$ of the distance to the true nearest neighbor. Throughout, we focus on $\mathbb{R}^d$ for fixed d and on data structures that achieve logarithmic query time of $O(\log \frac{n}{\varepsilon})$.

Approximate nearest neighbor searching in spaces of fixed dimension has been widely studied. Data structures with $O(n)$ storage and query times no better than $O(\log n + 1/\varepsilon^{d-1})$ have been proposed by several authors [39–42]. In subsequent papers, it was shown that query times could be reduced at the expense of greater storage [5, 43–45]. Har-Peled introduced the AVD (approximate Voronoi diagram) data structure and showed that $O(\log \frac{n}{\varepsilon})$ query time could be achieved using $\widetilde{O}(n/\varepsilon^d)$ space [44].

Space-time trade-offs were established for the AVD in a series of papers [46–49]. At one end of the spectrum, it was shown that with $O(n)$ storage, queries could be answered in time $O(\log n + 1/\varepsilon^{(d-1)/2})$. At the other end, queries could be answered in time $O(\log \frac{n}{\varepsilon})$ with space $\widetilde{O}(n/\varepsilon^d)$. In [33], the Arya et al. presented a reduction from Euclidean approximate nearest neighbor searching to polytope membership. They established significant improvements to the best trade-offs throughout the middle of the spectrum, but the extremes were essentially unchanged [24, 33]. While the AVD is simple and practical, in [47] lower bounds were presented that imply that significant improvements at the extreme ends of the spectrum are not possible in this model.

Recently, Arya et al. [34, 50] succeeded in reducing the storage to $O(n/\varepsilon^{d/2})$ by building upon recent developments on approximate polytope membership queries. Their main result achieves the following improved trade-off.

Theorem 2. *Given a set X of n points in $\mathbb{R}^d$, an approximation parameter $0 < \varepsilon \leq 1$, and m such that $\log \frac{1}{\varepsilon} \leq m \leq 1/(\varepsilon^{d/2} \log \frac{1}{\varepsilon})$, there is a data structure that can answer Euclidean ε-approximate nearest neighbor queries with*

$$\textit{Query time: } O\left(\log n + \frac{1}{m \cdot \varepsilon^{d/2}}\right) \quad \textit{and} \quad \textit{Space: } O(nm)\,.$$

By setting m to its upper limit it is possible to achieve logarithmic query time while roughly *halving* the exponent in the ε-dependency of the previous best bound.

2.2.3 Non-Euclidean Distances

Unlike the simpler data structure of [44], which can be applied to a variety of metrics, the recent results of Arya et al. [34, 50] exploit properties that are specific to Euclidean space, which significantly limits its applicability. [1] In particular, it applies a reduction to approximate polytope membership [27] based on the well-known *lifting transformation* [10]. However, this transformation applies only for the Euclidean distance. Furthermore, all the aforementioned data structures rely on the triangle inequality. Therefore, they cannot generally be applied to situations where each site is associated with its own distance function as arises, for example, with multiplicatively weighted sites.

Har-Peled and Kumar introduced a powerful technique to overcome this limitation through the use of *minimization diagrams* [52]. For each site p_i, let $f_i : \mathbb{R}^d \to \mathbb{R}^+$ be the associated distance function. Let $\mathcal{F}_{\min}$ denote the pointwise minimum of these functions, that is, the *lower-envelope function*. Clearly, approximating the value of $\mathcal{F}_{\min}$ at a query point q is equivalent to approximating the distance to q's nearest neighbor.[2] Har-Peled and Kumar proved that ε-ANN searching over a wide variety of distance functions (including additively and multiplicatively weighted sites) could

[1] Chan [51] presented a similar result by a very different approach, and it generalizes to some other distance functions, however the query time is not logarithmic.

[2] The idea of using envelopes of functions for the purpose of nearest-neighbor searching has a long history, and it is central to the well-known relationship between the Euclidean Voronoi diagram of a set of points in $\mathbb{R}^d$ and the lower envelope of a collection of hyperplanes in $\mathbb{R}^{d+1}$ through the lifting transformation [10].

be cast in this manner [52]. While this technique is very general, the complexity bounds are much worse than for the corresponding concrete versions. For example, in the case of Euclidean distance with multiplicative weights, in order to achieve logarithmic query time, the storage used is $O((n \log^{d+2} n)/\varepsilon^{2d+2} + n/\varepsilon^{d^2+d})$. Similar results are achieved for a number of other distance functions that are considered in [52].

This motivates the question of whether it is possible to answer ANN queries for non-Euclidean distance functions while matching the best bounds for Euclidean ANN queries. In Chapter 4, we apply our sampling methodology to obtain such data structures. We achieve this by adapting the sampling to *both* the local distribution of points and the growth rates of the distance functions. In addition, we circumvent the reliance on the lifting transform by a careful application of *convexification* from the optimization of non-convex functions.

2.3 Mesh Generation

The computational modeling of physical phenomena requires robust numerical algorithms and compatible high-quality domain discretizations so-called *meshes*. In this section, we review the most relevant related work on mesh generation. As we deal with pieceswise-smooth surfaces with arbitrarily small angles, we review prior work on this challenging problem through the development of Delaunay meshing algorithms in Section 2.3.1. Next, we motivate the relatively new interest in polyhedral meshing in Section 2.3.2. Then, Section 2.3.3 we further motivate the study of Voronoi meshes which will be the focus of Chapters 5 and 6.

2.3.1 Delaunay Mesh Generation

Delaunay refinement (DR) is a very successful algorithm for the generation of quality unstructured tetrahedral meshes [53]. Since the presence of small angles in the input domain may threaten the termination of DR, a lower bound on input angles may be necessary. A series of works extended DR to more general classes of domains starting with polyhedral domains with no input angles less than 90° [54], and then polyhedral domains with arbitrarily small angles [55]. Motivated by scientific applications dealing with realistic physical domains and engineering designs, the class of inputs with curved boundaries is particularly relevant as treated in [56, 57] and implemented in the CGAL library [58]; albeit with assumed lower bounds on the smallest angle in the input.

The challenging treatment of arbitrarily small input angles was finally resolved by Cheng et al. [59] for a large class of inputs called piecewise-smooth complexes. Cheng et al. [59] achieved that by deriving a feature size that blends the definitions used for smooth and polyhedral domains, ensuring the protection of sharp features. However, their algorithm is largely impractical as it relies on expensive predicates evaluated using the equations of the underlying surface. To obtain a practical variant as implemented in the DelPSC software, Dey and Levin [60] relied on an input threshold to guide refinement, where topological correctness can only be guaranteed if it is sufficiently small.

2.3.2 Polyhedral Mesh Generation

The limited degrees of freedom of linear tetrahedral as well as hexahedral elements often require excessive refinement when modeling complex geometries or domains undergoing large deformations, e.g., cutting, merging, fracturing, or adaptive refinement [61–64]. This motivated generalizations to general polyhedral elements, which enjoy larger degrees of freedom.

While the generation of tetrahedral meshes based on Delaunay refinement [53] or variational optimization [65] is well established, research on polyhedral mesh generation is less mature. State-of-the-art approaches often rely on *clipping*, i.e., truncating cells of an initial mesh to fit the domain boundaries [66]. Such an initial mesh can be obtained as a Voronoi mesh, e.g., with seeds randomly generated inside the domain [67] or optimized by centroidal Voronoi tessellations (CVT) [66], possibly taking anisotropy into account [68]. Alternatively, an initial Voronoi mesh can be obtained by dualizing a conforming tetrahedral mesh [69]. Although no clipping is needed if the tetrahedralization is *well-centered*, generating such meshes is very challenging and only heuristic solutions are known [70]. A weaker *Gabriel property* ensures all tetrahedra have circumcenters inside the domain and can be guaranteed for polyhedral domains with bounded minimum angles [71]; however, the dual Voronoi cells still need to be clipped.

2.3.3 Orthogonal Primal-Dual Meshing

Voronoi meshes, along with their dual Delaunay triangulations, are a prime example of primal-dual mesh pairs. In particular, the Voronoi facets are *orthogonal* to their dual Delaunay facets. More generally, orthogonal primal-dual mesh pairs are unstructured staggered meshes [72] with desirable conservation properties [73], enabling discretizations that closely mimic the continuum equations being modeled [74, 75]. The power of orthogonal duals [76] was recognized in early works on structural design [77, 78] and numerical methods [79], and has recently been demonstrated on a range of applications in computer graphics [80], self-supporting structures [81], mesh parameterization [82], and computational physics [83]. In particular, Voronoi-Delaunay meshes are the default geometric realization of many formulations in numerical methods [84], fluid animation [85], fracture modeling [86], and computational cell biology [87].

Chapter 3: Polytope Membership Queries

Polytope membership queries, both exact and approximate, arise in many application areas, such as linear programming and ray-shooting queries [1–4], nearest neighbor searching and the computation of extreme points [5–7], collision detection [8], and machine learning [9]. Please refer to Section 2.1 for a review of related work.

In this chapter, we demonstrate an intriguing connection between convex approximation and the classical concept of Delone sets from the theory of metric spaces. We show that with the help of a classical structure from convexity theory, called the Macbeath region, we design a data structure that answers ε-approximate polytope membership queries in $O(\log(1/\varepsilon))$ time. This matches the best asymptotic results for this problem, by a data structure that both is simpler and arguably more elegant.

3.1 Introduction

We consider the following fundamental query problem. Let K denote a bounded convex polytope in $\mathbb{R}^d$, presented as the intersection of n halfspaces. The objective is to preprocess K so that, given any query point $q \in \mathbb{R}^d$, it is possible to determine efficiently whether q lies in K. Throughout, we assume that d is a fixed constant and K is full-dimensional.

Let ε be a positive real parameter, and let $\operatorname{diam}(K)$ denote K's diameter. Given a query point $q \in \mathbb{R}^d$, an *ε-approximate polytope membership query* returns a positive result if $q \in K$, a negative result if the distance from q to its closest point in K is greater than $\varepsilon \cdot \operatorname{diam}(K)$, and it may return either result otherwise.

A space-optimal solution for the case of polylogarithmic query time was presented in [34]. It achieves query time $O(\log \frac{1}{\varepsilon})$ with storage $O(1/\varepsilon^{(d-1)/2})$. This paper achieves its efficiency by abandoning the grid- and quadtree-based approaches in favor of an approach based on ellipsoids and a classical structure from convexity theory called a *Macbeath region* [35].

The approach presented in [34] is based on constructing a collection of nested eroded bodies within K and covering the boundaries of these eroded bodies with ellipsoids that are based on Macbeath regions. Queries are answered by shooting rays from a central point in the polytope towards the boundary of K, and tracking an ellipsoid at each level that is intersected by the ray. While it is asymptotically optimal, the data structure and its analysis are complicated by various elements that are artifacts of this ray shooting approach.

In this chapter, we present a simpler and more intuitive approach with the same asymptotic complexity as the one in [34]. The key idea is to place the Macbeath regions based on *Delone sets*. A Delone set is a concept from the study of metric spaces. It consists of a set of points that have nice packing and covering properties with respect to the metric balls. Our main result is that any maximal set of disjoint shrunken Macbeath regions defines a Delone set with respect to the Hilbert metric induced on a suitable expansion of the convex body. This observation leads to a simple DAG structure for membership queries. The DAG structure arises from a hierarchy of Delone sets obtained by layering a sequence of expansions of the body. Our results uncover a natural connection between the classical concepts of Delone sets from the theory of metric spaces and Macbeath regions and the Hilbert geometry from the theory of convexity.

3.2 Preliminaries

In this section we present a number of basic definitions and results, which will be used throughout the chapter. We consider the real d-dimensional space, $\mathbb{R}^d$, where d is a fixed constant. Let O denote the origin of $\mathbb{R}^d$. Given a vector $v \in \mathbb{R}^d$, let $\|v\|$ denote its Euclidean length, and let $\langle \cdot, \cdot \rangle$ denote the standard inner product. Given two points $p, q \in \mathbb{R}^d$, the Euclidean distance between them is $\|p - q\|$. For $q \in \mathbb{R}^d$ and $r > 0$, let $B(q, r)$ denote the Euclidean ball of radius r centered at q, and let $B(r) = B(O, r)$.

3.2.1 Polytope Representation

Let K be a convex body in $\mathbb{R}^d$, represented as the intersection of m closed halfspaces $H_i = \{x \in \mathbb{R}^d : \langle x, v_i \rangle \leq a_i\}$, where a_i is a nonnegative real and $v_i \in \mathbb{R}^d$. The bounding hyperplane for H_i is orthogonal to v_i and lies at distance $a_i/\|v_i\|$ from the origin. The boundary of K will be denoted by ∂K. For $0 < \kappa \leq 1$, we say that K is in *κ-canonical form* if $B(\kappa/2) \subseteq K \subseteq B(1/2)$. Clearly, such a body has a diameter between κ and 1.

It is well known that in $O(m)$ time it is possible to compute a non-singular affine transformation T such that $T(K)$ is in $(1/d)$-canonical form [44, 88]. Further, if a convex body P is within Hausdorff distance ε of $T(K)$, then $T^{-1}(P)$ is within Hausdorff distance at most $d\varepsilon$ of K. (Indeed, this transformation is useful, since the resulting approximation is directionally sensitive, being more accurate along directions where K is skinnier.) Therefore, for the sake of approximation with respect to Hausdorff distance, we may assume that K has been mapped to canonical form, and ε is scaled by a factor of $1/d$. Because we assume that d is a constant, this transformation will only affect the constant factors in our analysis.

3.2.2 Polytope Expansion

A number of our constructions involve perturbing the body K by means of *expansion*, but the exact nature of the expansion is flexible in the following sense. Given $\delta > 0$, let K_δ denote any convex body containing K such that the Hausdorff distance between ∂K and ∂K_δ is $\Theta(\delta \cdot \mathrm{diam}(K))$. For example, if K is in canonical form, K_δ could result as the Minkowski sum of K with another convex body of diameter δ or from a uniform scaling about the origin by δ. Because reducing the approximation parameter by a constant factor affects only the constant factors in our complexity bounds, the use of an appropriate K_δ instead of closely related notions of approximation, like the two just mentioned, will not affect our asymptotic bounds.

Given $\delta > 0$, we perturb each H_i to obtain

$$H_{i,\delta} \;=\; \{x \in \mathbb{R}^d : \langle x, \vec{v}_i \rangle \leq a_i + \delta)\}.$$

The associated bounding hyperplane is parallel to that of H_i and translated away from the origin by a distance of $\delta/\|v_i\|$. With that, we define K_δ as the convex polytope $\bigcap_{i=1}^n H_{i,\delta}$. To ensure the required bound on the Hausdorff error, we require that $c_1\delta \leq \|v_i\| \leq c_2$ for all i, where c_1 and c_2 are nonnegative reals. The following argument shows that this condition suffices. If $c_1\delta \leq \|v_i\| \leq c_2$, then each bounding halfspace of K is translated away from the origin by a distance of $\delta/\|v_i\| \geq \delta/c_2$, which establishes the lower bound on the Hausdorff distance. Also, each bounding halfspace is translated by a distance of $\delta/\|v_i\| \leq 1/c_1$. Since K, being in canonical form, is nested between balls of radius $\kappa/2$ and $1/2$, this translation of the halfspace is equivalent to a scaling about the origin by a factor of at most $2/c_1\kappa$, which maps each point of K away from the origin by a distance of at most $(2/c_1\kappa)/2 = 1/c_1\kappa$. This establishes the upper bound on the Hausdorff distance.

3.2.3 Macbeath Regions

Our algorithms and data structures will involve packings and coverings by ellipsoids, which will possess the essential properties of Delone sets. These ellipsoids are based on a classical concept from convexity theory, called *Macbeath regions*, which were described first by A. M. Macbeath in a paper on the existence of certain lattice points in a convex body [35]. They have found uses in diverse areas (see, e.g., Bárány's survey [30]).

Given a convex body K, a point $x \in K$, and a real parameter $\lambda \geq 0$, the λ-*scaled Macbeath region* at x, denoted $M_K^\lambda(x)$, is defined to be

$$x + \lambda((K - x) \cap (x - K)).$$

When $\lambda = 1$, it is easy to verify that $M_K^1(x)$ is the intersection of K and the reflection of K around x (see Fig. 3.1a), and hence it is centrally symmetric about x. $M_K^\lambda(x)$ is a scaled copy of $M_K^1(x)$ by the factor λ about x. We refer to x and λ as the *center* and *scaling factor* of $M_K^\lambda(x)$, respectively. To simplify the notation, when K is clear from the context, we often omit explicit reference in the subscript and use $M^\lambda(x)$ in place of $M_K^\lambda(x)$. When $\lambda < 1$, we say $M^\lambda(x)$ is *shrunken*. When $\lambda = 1$, $M^1(x)$ is *unscaled* and we drop the superscript. Recall that if C^λ is a uniform λ-factor scaling of any bounded, full-dimensional set $C \subset \mathbb{R}^d$, then $\mathrm{vol}(C^\lambda) = \lambda^d \cdot \mathrm{vol}(C)$.

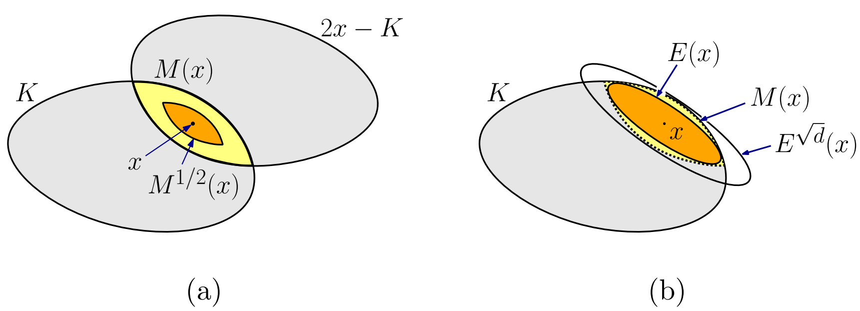


Figure 3.1: (a) Macbeath regions and (b) Macbeath ellipsoids.

An important property of Macbeath regions, which we call *expansion-containment*, is that if two shrunken Macbeath regions overlap, then an appropriate expansion of one contains the other (see Fig. 3.2a). The following is a generalization of results of Ewald, Rogers and Larman [89] and Brönnimann, Chazelle, and Pach [90]. Our generalization allows the shrinking factor λ to be adjusted, and shows how to adjust the expansion factor β of the first body to cover an α-scaling of the second body, e.g., the center point only (see Fig. 3.2b).

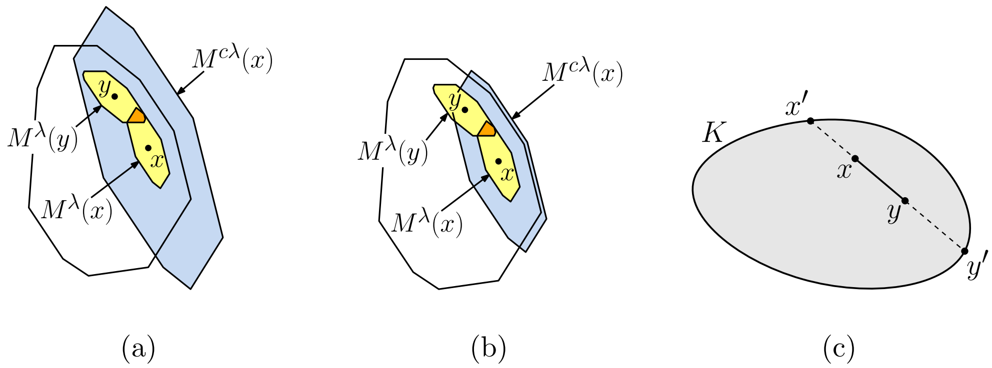


Figure 3.2: (a)-(b) Expansion-containment per Lemma 1. (c) The Hilbert metric.

Lemma 1. *Let $K \subset \mathbb{R}^d$ be a convex body and let $0 < \lambda < 1$. If $x, y \in K$ such that $M^{\lambda}(x) \cap M^{\lambda}(y) \neq \emptyset$, then for any $\alpha \geq 0$ and $\beta = \frac{2+\alpha(1+\lambda)}{1-\lambda}$, $M^{\alpha\lambda}(y) \subseteq M^{\beta\lambda}(x)$ (see Fig. 3.2).*

3.2.4 Delone Sets and the Hilbert Metric

An important concept in the context of metric spaces involves coverings and packings by metric balls [17]. Given a metric f over $\mathbb{X}$, a point $x \in \mathbb{X}$, and real $r > 0$, define the ball $B_f(x, r) = \{y \in \mathbb{X} : f(x, y) \leq r\}$. For $\varepsilon, \varepsilon_p, \varepsilon_c > 0$, a set $X \subseteq \mathbb{X}$ is an:

ε-packing: If the balls of radius $\varepsilon/2$ centered at every point of X do not intersect.

ε-covering: If every point of $\mathbb{X}$ is within distance ε of some point of X.

$(\varepsilon_p, \varepsilon_c)$-Delone Set: If X is an ε_p-packing and an ε_c-covering.

Delone sets have been used in the design of data structures for answering geometric proximity queries in metric spaces through the use of hierarchies of nets, such as navigating nets [91], net trees [92], and cover trees [93].

In order to view a collection of Macbeath regions as a Delone set, it will be useful to introduce an underlying metric. The Hilbert metric [94] was introduced over a century ago by David Hilbert as a generalization of the Cayley-Klein model of hyperbolic geometry. A *Hilbert geometry* (K, f_K) consists of a convex domain K in $\mathbb{R}^d$ with the Hilbert distance f_K. For any pair of distinct points $x, y \in K$, the line passing through them meets ∂K at two points x' and y'. We label these points so that they appear in the order $\langle x', x, y, y' \rangle$ along this line (see Fig. 3.2c). The Hilbert distance f_K is defined as

$$f_K(x,y) \;=\; \frac{1}{2} \ln \left(\frac{\|x'-y\|}{\|x'-x\|} \frac{\|x-y'\|}{\|y-y'\|} \right).$$

When K is not bounded and either x' or y' is at infinity, the corresponding ratio is taken to be 1. To get some intuition, observe that if x is fixed and y moves along a ray starting at x towards ∂K, $f_K(x, y)$ varies from 0 to ∞.

Hilbert geometries have a number of interesting properties; see the survey by Papadopoulos and Troyanov [95] and the multimedia contribution by Nielsen and Shao [96]. First, f_K can be shown to be a metric. Second, it is invariant under projective transformations.[1] Finally, when K is a unit ball in $\mathbb{R}^d$, the Hilbert distance is equal (up to a constant factor) to the distance between points in the Cayley-Klein model of hyperbolic geometry.

[1]This follows from the fact that the argument to the logarithm function is the *cross ratio* of the points (x', x, y, y'), and it is well known that cross ratios are preserved under projective transformations.

Given a point $x \in K$ and $r > 0$, let $B_H(x, r)$ denote the ball of radius r about x in the Hilbert metric. The following lemma shows that a shrunken Macbeath region is nested between two Hilbert balls whose radii differ by a constant factor (depending on the scaling factor). Thus, up to constant factors in scaling, Macbeath regions and their associated ellipsoids can act as proxies to metric balls in Hilbert space. This nesting was observed by Vernicos and Walsh [97] (for the conventional case of $\lambda = 1/5$), and we present the straightforward generalization to other scale factors. For example, with $\lambda = 1/5$, we have $B_H(x, 0.09) \subseteq M^{1/5}(x) \subseteq B_H(x, 0.21)$ for all $x \in K$.

Lemma 2. *Given a convex body $K \subset \mathbb{R}^d$, for all $x \in K$ and any $0 \leq \lambda < 1$,*

$$B_H\big(x, \frac{1}{2}\ln(1+\lambda)\big) \;\subseteq\; M^{\lambda}(x) \;\subseteq\; B_H\left(x, \frac{1}{2}\ln\frac{1+\lambda}{1-\lambda}\right).$$

3.3 Macbeath Regions as Delone Sets

Lemma 2 justifies using Macbeath regions as Delone sets. Given a point $x \in K$ and $\delta > 0$, define $M_\delta(x)$ to be the (unscaled) Macbeath region with respect to K_δ, that is, $M_\delta(x) = M_{K_\delta}(x)$. Towards our goal of using Delone sets for approximating convex bodies, we study the behavior of overlapping Macbeath regions at different scales of approximation and establish a bound on the size of such Delone sets. In particular, we consider maximal sets of disjoint shrunken Macbeath regions $M_\delta^\lambda(x)$ defined with respect to K_δ, such that the centers x lie within K; let X_δ denote such a set of centers. The two scale factors used to define the Delone set will be denoted by (λ_p, λ_c), where we assume $0 < \lambda_p < \lambda_c < 1$ are constants. Define $M'_\delta(x) = M_\delta^{\lambda_c}(x)$

and $M''_\delta(x) = M_\delta^{\lambda_p}(x)$.

3.3.1 Varying the Scale

A crucial property of metric balls is how they adapt to changing the resolution at which the domain in question is being modeled. We show that Macbeath regions enjoy a similar property.

Lemma 3. *Given a convex body $K \subset \mathbb{R}^d$ and $\lambda, \delta, \varepsilon \geq 0$, for all $x \in K$,*

$$M_{K_\delta}^{\lambda}(x) \;\subseteq\; M_{K_{(1+\varepsilon)\delta}}^{\lambda}(x) \;\subseteq\; M_{K_\delta}^{(1+\varepsilon)\lambda}(x).$$

Proof. The first inclusion is a simple consequence of the fact that enlarging the body can only enlarge the Macbeath regions. To see the second inclusion, it will simplify the notation to translate space by $-x$ so that x now coincides with the origin. Thus, $M_K(x) = K \cap -K$. Recalling our representation from Section 3.2, we can express K as the intersection of a set of halfspaces $H_i = \{y : \langle y, v_i \rangle \leq a_i\}$. (The translation affects the value of a_i, but not the approximation, because $x \in K$, $a_i \geq 0$.) We can express $M_K(x)$ as the intersection of a set of slabs $\Sigma_i = H_i \cap -H_i$, where each slab is centered about the origin. $M_{K_\delta}(x)$ can be similarly expressed as the intersection of slabs $\Sigma_{i,\delta} = H_{i,\delta} \cap -H_{i,\delta}$, where the defining inequality is $\langle y, v_i \rangle \leq a_i + \delta$. This applies analogously to $M_{K_{(1+\varepsilon)\delta}}(x)$, where the defining inequality is $\langle y, v_i \rangle \leq a_i + (1+\varepsilon)\delta$. Since $a_i \geq 0$, we have $a_i + (1+\varepsilon)\delta \leq (1+\varepsilon)(a_i + \delta)$, which implies that $\Sigma_{i,(1+\varepsilon)\delta} \subseteq (1+\varepsilon)\Sigma_{i,\delta}$. Thus, we have

$$M_{K_{(1+\varepsilon)\delta}}(x) \;=\; \bigcap\nolimits_{i=1}^{m} \Sigma_{i,(1+\varepsilon)\delta} \;\subseteq\; \bigcap\nolimits_{i=1}^{m} (1+\varepsilon)\Sigma_{i,\delta} \;=\; M_{K_\delta}^{(1+\varepsilon)}(x).$$

The lem now follows by applying a scaling factor of λ to both sides. $\square$

As we refine the approximation by using smaller values of δ, it is important to bound the number of Macbeath regions at higher resolution that overlap any given Macbeath region at a lower resolution. Our bound is based on a simple packing argument. We will show that the shrunken Macbeath regions $M''_\delta(y)$ that overlap a fixed shrunken Macbeath region at a coarser level of approximation $M'_{s\delta}(x)$, with $s \geq 1$, lie within a suitable constant-factor expansion of $M'_{s\delta}(x)$. Let $Y_{\delta,s}(x)$ denote the set of points y such that $M''_\delta(y)$ are pairwise disjoint and overlap $M'_{s\delta}(x)$. Since these shrunken Macbeath regions are pairwise disjoint, we can bound their number by bounding the ratio of volumes of $M'_{s\delta}(x)$ and $M''_\delta(y)$.

As an immediate corollary of the second inclusion of Lemma 3 we have $\mathrm{vol}(M^\lambda_\delta(x)) \geq \mathrm{vol}(M^\lambda_{s\delta}(x))/s^d$. This allows us to establish an upper bound on the growth rate in the number of Macbeath regions when refining to smaller scales.

Lemma 4. *Given a convex body $K \subset \mathbb{R}^d$ and $x \in K$. Then, for constants $\delta \geq 0$, $s \geq 1$ and $Y_{\delta,s}(x)$ as defined above, $|Y_{\delta,s}(x)| = O(1)$.*

Proof. By the first inclusion of Lemma 3, $M'_\delta(y) \subseteq M'_{s\delta}(y)$, and we have $M'_{s\delta}(x) \cap M'_{s\delta}(y) \neq \emptyset$. Next, by applying Lemma 1 (with the roles of x and y swapped) we obtain $M'_{s\delta}(x) = M^{\lambda_c}_{s\delta}(x) \subseteq M^{\beta\lambda_c}_{s\delta}(y)$, with $\alpha = 1$ and $\beta = (3+\lambda_c)/(1-\lambda_c)$.

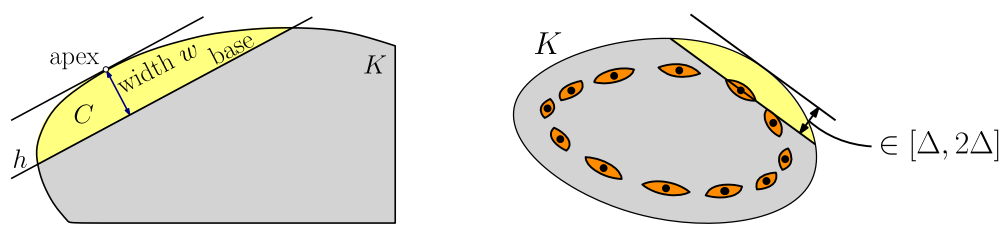
By definition of X_δ the shrunken Macbeath regions $M''_\delta(y)$ are pairwise disjoint, and so it suffices to bound their volumes with respect to that of $M'_{s\delta}(x)$ to obtain a bound on $|Y_{\delta,s}(x)|$. Applying the corollary to Lemma 3 and scaling, we obtain

$$\mathrm{vol}(M''_\delta(y)) \;\geq\; \frac{1}{s^d}\mathrm{vol}(M''_{s\delta}(y)) \;=\; \left(\frac{\lambda_p}{\beta\lambda_c s}\right)^d \mathrm{vol}(M^{\beta\lambda_c}_{s\delta}(y)) \;\geq\; \left(\frac{\lambda_p}{\beta\lambda_c s}\right)^d \mathrm{vol}(M'_{s\delta}(x)).$$

Thus, by a packing argument the number of children is at most $\left(\frac{\beta\lambda_c s}{\lambda_p}\right)^d = O(1)$. $\square$

3.3.2 Size Bound

We bound the cardinality of a maximal set of disjoint shrunken Macbeath regions $M_\delta^\lambda(x)$ defined with respect to K_δ, such that the centers x lie within K; let X_δ denote such a set of centers. This is facilitated by associating each center x with a *cap* of K, where a cap C is defined as the nonempty intersection of the convex body K with a halfspace (see Fig. 3.3a). Letting h denote the hyperplane bounding this halfspace, the *base* of C is defined as $h \cap K$. The *apex* of C is any point in the cap such that the supporting hyperplane of K at this point is parallel to h. The *width* of C is the distance between h and this supporting hyperplane. Of particular interest is a cap of minimum volume that contains x, which may not be unique. A simple variational argument shows that x is the centroid of the base of this cap [89].



(a) (b)

Figure 3.3: (a) Cap concepts and (b) the economical cap cover.

As each Macbeath region is associated with a cap, we can obtain the desired bound by bounding the number of associated caps. We achieve this by appealing to the so-called *economical cap covers* [98]. The following lem is a straightforward adaptation of the width-based economical cap cover per Lemma 3.2 of [88].

Lemma 5. *Let $K \subset \mathbb{R}^d$ be a convex body in κ-canonical form. Let $0 < \lambda \leq 1/5$ be any fixed constant, and let $\Delta \leq \kappa/12$ be a real parameter. Let $\mathcal{C}$ be a set of caps, whose widths lie between Δ and 2Δ, such that the Macbeath regions $M_K^\lambda(x)$ centered at the centroids x of the bases of these caps are disjoint. Then $|\mathcal{C}| = O(1/\Delta^{(d-1)/2})$ (see Fig. 3.3a(b)).*

This leads to the following bound on the number of points in X_δ.

Lemma 6. *Let $K \subset \mathbb{R}^d$ be a convex body in κ-canonical form, and let X_δ as defined above for some $\delta > 0$ and $0 < \lambda \leq 1/5$. Then, $|X_\delta| = O(1/\delta^{(d-1)/2})$.*

Proof. In order to apply Lemma 5 we will partition the points of X_δ according to the widths of their minimum-volume caps. For $i \geq 0$, define $\Delta_i = c_2 2^i \delta_i$, where c_2 depends on the nature of the the expansion process that yields K_δ. Define $X_{\delta,i}$ to be the subset of points $x \in X_\delta$ such that width of x's minimum cap with respect to K_δ lies within $[\Delta_i, 2\Delta_i]$. By choosing c_2 properly, the Hausdorff distance between K and K_δ is at least $c_2\delta = \Delta_0$, and therefore any cap whose base passes through a point of X_δ has width at least Δ_0. This implies that every point of X_δ lies in some subset $X_{\delta,i}$ for $i \geq 0$.

If a convex body is in κ-canonical form, it follows from a simple geometric argument that for any point x in this body whose minimal cap is of width at least Δ, the body contains a ball of radius $c\Delta$ centered at x, for some constant c (depending on κ and d). If $\Delta_i > \kappa/12$, then $B(x, c\kappa/12) \subseteq K_\delta$ for all $x \in X_{\delta,i}$. It follows that $B(x, c\kappa/12) \subseteq M_\delta(x)$ implying that $\mathrm{vol}(M_\delta^\lambda(x)) \geq \lambda^d \cdot \mathrm{vol}(B(c\kappa/12))$ which is $\Omega(1)$

as c, κ and λ are all constants. By a simple packing argument $|X_{i,j}| = O(1)$. There are at most a constant number of levels for which $\Delta_j > \kappa/12$, and so the overall contribution of these subsets is $O(1)$.

Henceforth, we may assume that $\Delta_j \leq \kappa/12$. Since $\lambda \leq 1/5$, we apply Lemma 5 to obtain the bound $|X_{\delta,i}| = O(1/\Delta_i^{(d-1)/2})$. (There is a minor technicality here. If δ becomes sufficiently large, K_δ may not be in κ-canonical form because its diameter is too large. Because $\delta = O(1)$ and hence $\mathrm{diam}(K_\delta) = O(1)$, we may scale it back into canonical form at the expense of increasing the constant factors hidden in the asymptotic bound.) Thus, up to constant factors, we have

$$|X_\delta| \;=\; \sum_{i \geq 0} |X_{\delta,i}| \;=\; \sum_{i \geq 0} O\left(\frac{1}{\Delta_i}\right)^{\frac{d-1}{2}} \;=\; \sum_{i \geq 0} O\left(\frac{1}{c_2 2^i \delta}\right)^{\frac{d-1}{2}} \;=\; O\left(\left(\frac{1}{\delta}\right)^{\frac{d-1}{2}}\right).$$

□

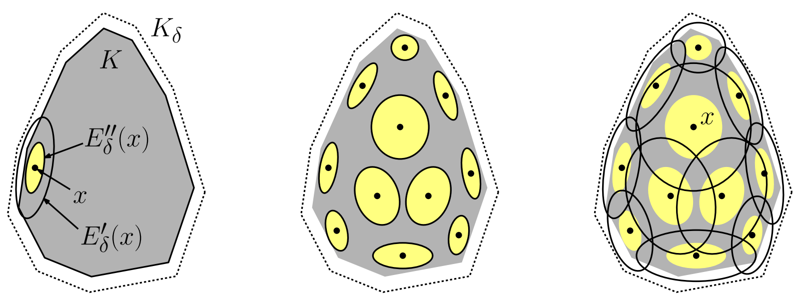
3.3.3 Macbeath Ellipsoids

For the sake of efficient computation, it will be useful to approximate Macbeath regions by shapes of constant combinatorial complexity. We have opted to use ellipsoids. (Note that bounding boxes [7] could be used instead, and may be preferred in contexts where polytopes are preferred.)

Given a Macbeath region, define its associated *Macbeath ellipsoid* $E_K^\lambda(x)$ to be the maximum-volume ellipsoid contained within $M_K^\lambda(x)$ (see Fig. 3.1b). Clearly, this ellipsoid is centered at x and $E_K^\lambda(x)$ is an λ-factor scaling of $E_K^1(x)$ about x. It is well known that the maximum-volume ellipsoid contained within a convex body is unique, and Chazelle and Matoušek showed that it can be computed for a convex polytope in time linear in the number of its bounding halfspaces [99]. By John's Theorem (applied in the context of centrally symmetric bodies) it follows that $E_K^{\lambda}(x) \subseteq M_K^{\lambda}(x) \subseteq E_K^{\lambda\sqrt{d}}(x)$ [100].

Given a point $x \in K$ and $\delta > 0$, define $M_\delta(x)$ to be the (unscaled) Macbeath region with respect to K_δ (as defined in Section 3.2), that is, $M_\delta(x) = M_{K_\delta}(x)$. Let $E_\delta(x)$ denote the maximum volume ellipsoid contained within $M_\delta(x)$. As $M_\delta(x)$ is symmetric about x, $E_\delta(x)$ is centered at x. For any $\lambda > 0$, define $M_\delta^{\lambda}(x)$ and $E_\delta^{\lambda}(x)$ to be the uniform scalings of $M_\delta(x)$ and $E_\delta(x)$, respectively, about x by a factor of λ. By John's Theorem, we have

$$E_\delta^{\lambda}(x) \;\subseteq\; M_\delta^{\lambda}(x) \;\subseteq\; E_\delta^{\lambda\sqrt{d}}(x). \tag{3.1}$$



(a) (b) (c)

Figure 3.4: A Delone set for a convex body. (Not drawn to scale.)

Two particular scale factors will be of interest to us. Define $M'_\delta(x) = M_\delta^{1/2}(x)$ and $M''_\delta(x) = M_\delta^{\lambda_0}(x)$, where $\lambda_0 = 1/(4\sqrt{d} + 1)$. Similarly, define $E'_\delta(x) = E_\delta^{1/2}(x)$ and $E''_\delta(x) = E_\delta^{\lambda_0}(x)$ (see Fig. 3.4(a)). Given a fixed δ, let X_δ be any maximal set of points, all lying within K, such that the ellipsoids $E''_\delta(x)$ are pairwise disjoint for all $x \in X_\delta$.

These ellipsoids form a packing of K_δ (see Fig. 3.4(b)). The following lem shows that their suitable expansions cover K while being contained within K_δ (see Fig. 3.4(c)).

Lemma 7. *Given a convex body K in $\mathbb{R}^d$ and a set X_δ as defined above for $\delta > 0$,*

$$K \ \subseteq \ \bigcup_{x \in X_\delta} E'_\delta(x) \ \subseteq \ K_\delta.$$

Proof. To establish the first inclusion, consider any point $y \in K$. Because X_δ is maximal, there exists $x \in X_\delta$ such that $E''_\delta(x) \cap E''_\delta(y)$ is nonempty. By containment, $M''_\delta(x) \cap M''_\delta(y)$ is also nonempty. By Lemma 1 (with $\alpha = 0$), it follows that $y \in M^\lambda_\delta(x)$, where

$$\lambda \ = \ \frac{2\lambda_0}{1-\lambda_0} \ = \ \frac{2/(4\sqrt{d}+1)}{1-1/(4\sqrt{d}+1)} \ = \ \frac{2}{4\sqrt{d}} \ = \ \frac{1}{2\sqrt{d}}.$$

By applying Eq. (3.1) (with $\lambda = 1/(2\sqrt{d})$), we have $M_\delta^{1/(2\sqrt{d})}(x) \subseteq E_\delta^{1/2}(x) = E'_\delta(x)$, and therefore $y \in E'_\delta(x)$. Thus, we have shown that an arbitrary point $y \in K$ is contained in the ellipsoid $E'_\delta(x)$ for some $x \in X_\delta$, implying that the union of these ellipsoids covers K. The second inclusion follows from $E'_\delta(x) \subseteq M'_\delta(x) \subseteq M_\delta(x) \subseteq K_\delta$ for any $x \in X_\delta \subseteq K$. $\square$

In conclusion, if we treat the scaling factor λ in $E^\lambda(x)$ as a proxy for the radius of a metric ball, we have shown that X_δ is a $(2\lambda_0, 1/2)$-Delone set for K. By Lemma 2 this is also true in the Hilbert metric over K_δ up to a constant factor adjustment in the radii. (Note that the scale of the Hilbert balls does not vary with δ. What varies is the choice of the expanded body K_δ defining the metric.)

By John's Theorem, Macbeath regions and Macbeath ellipsoids differ by a constant scaling factor, both with respect to enclosure and containment. We remark that all the results of the previous two sections hold equally for Macbeath ellipsoids. We omit the straightforward, but tedious, details.

Remark 1. *All results from previous subsection on scaled Macbeath regions apply to scaled Macbeath ellipsoids subject to appropriate modifications of the constant factors.*

3.4 Approximate Polytope Membership

The Macbeath-based Delone sets developed above yield a simple data structure for answering ε-APM queries for a convex body K. We assume that K is represented as the intersection of m halfspaces. We may assume that in $O(m)$ time it has been transformed into κ-canonical form, for $\kappa = 1/d$. Throughout, we will assume that Delone sets are based on the Macbeath ellipsoids $E''_\delta(x)$ for packing and $E'_\delta(x)$ for coverage (defined in Section 3.3.3).

3.4.1 The Data Structure

Our data structure is based on a hierarchy of Delone sets of exponentially increasing accuracy. Define $\delta_0 = \varepsilon$, and for any integer $i \geq 0$, define $\delta_i = 2^i \delta_0$. Let X_i denote a Delone set for K_{δ_i}. By Lemma 7, we may take X_i to be any maximal set of points within K such that the packing ellipsoids $E''_\delta(x)$ are pairwise disjoint. Let $\ell = \ell_\varepsilon$ be the smallest integer such that $|X_\ell| = 1$. We will show below that

$\ell = O(\log 1/\varepsilon)$.

Given the sets $\langle X_0, \ldots, X_\ell \rangle$, we build a rooted, layered DAG structure as follows. The nodes of level i correspond 1–1 with the points of X_i. The leaves reside at level 0 and the root at level ℓ. Each node $x \in X_i$ is associated with two things. The first is its *cell*, denoted $\mathrm{cell}(x)$, which is the covering ellipsoid $E'_\delta(x)$ (the larger hollow ellipsoids shown in Fig. 3.5). The second, if $i > 0$, is a set of *children*, denoted $\mathrm{ch}(x)$, which consists of the points $y \in X_{i-1}$ such that $\mathrm{cell}(x) \cap \mathrm{cell}(y) \neq \emptyset$.

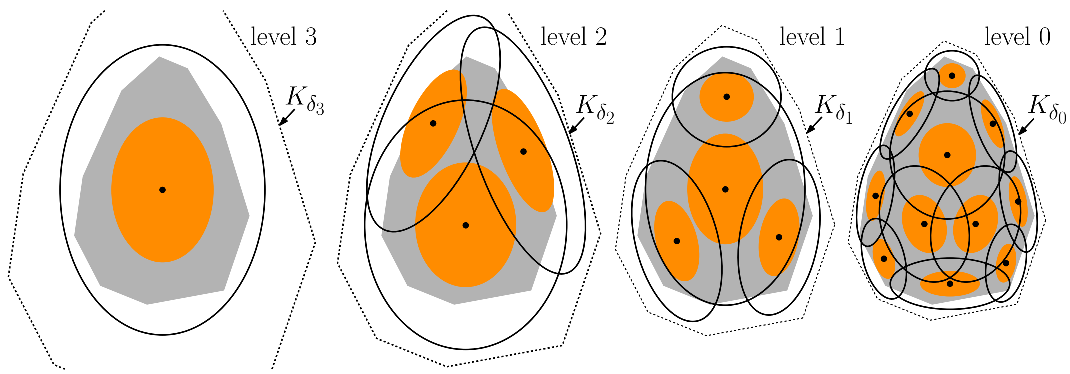


Figure 3.5: Hierarchy of ellipsoids for answering APM queries.

To answer a query q, we start at the root and iteratively visit any one node $x \in X_i$ at each level of the DAG, such that $q \in \mathrm{cell}(x)$. We know that if q lies within K, such an x must exist by the covering properties of Delone sets, and further at least one of x's children contains q. If q does not lie within any of the children of the current node, the query algorithm terminates and reports (without error) that $q \notin K$. Otherwise the search eventually reaches a node $x \in X_0$ at the leaf level whose cell contains q. Since $\mathrm{cell}(x) \subseteq K_{\delta_0} = K_\varepsilon$, this cell serves as a witness to q's approximate membership within K.

3.4.2 Performance Analysis

In order to bound the space and query time, we need to bound the total space used by the data structure and the time to process each node in the search, which is proportional to the number of its children. Building upon Lemmas 4 and 6, we have our main result.

Theorem 3. *Given a convex body K and $\varepsilon > 0$, there exists a data structure of space $O(1/\varepsilon^{(d-1)/2})$ that answers ε-approximate polytope membership queries in time $O(\log 1/\varepsilon)$.*

Since the expansion factors δ_i grow exponentially from ε to a suitably large constant, it follows that the height of the tree is logarithmic in $1/\varepsilon$, which is made formal below.

Lemma 8. *The DAG structure described above has height $O(\log 1/\varepsilon)$.*

Proof. Let c_2 be an appropriate constant, and let $\ell = \lceil \log_2(2/c_2\varepsilon) \rceil = O(\log 1/\varepsilon)$. Depending the nature of the expanded body K_δ, the constant c_2 can be chosen so the Hausdorff distance between K and K_{δ_ℓ} is at least $c_2\delta_\ell = c_2 2^\ell \varepsilon \geq 2$. Because K is in κ-canonical form, it is contained within a unit ball centered at the origin. Therefore, K_{δ_ℓ} contains a ball of radius two centered at the origin, which implies that the Macbeath ellipsoid $E'_{\delta_\ell}(O)$ (which is scaled by $1/2$) contains the unit ball and so contains K. Thus, (assuming that the origin is added first to the Delone set) level ℓ of the DAG contains a single node. $\square$

By Lemma 4, each node has $O(1)$ children and $\delta_i = 2^i\delta_0 = 2^i\varepsilon$, we obtain the following space bound by summing $|X_i|$ for $0 \leq i \leq \ell$.

Lemma 9. *The storage required by the DAG structure described above is* $O(1/\varepsilon^{(d-1)/2})$.

As mentioned above, by combining Lemmas 4 with 6, it follows that the query time is $O(\log 1/\varepsilon)$ and by Lemma 9 the total space is $O(1/\varepsilon^{(d-1)/2})$, which establish Theorem 3.

3.4.3 Construction

While our focus has been on demonstrating the existence of a simple data structure derived from Delone sets, we note that it can be constructed by well-established techniques. While obtaining the best dependencies on ε in the construction time will likely involve fairly sophisticated methods, as seen in the paper of Arya et al. [50], the following shows that there is a straightforward construction.

Lemma 10. *Given a convex body $K \subset \mathbb{R}^d$ represented as the intersection of m halfspaces and $\varepsilon > 0$, the above DAG structure for answering ε-APM queries can be computed in time $O(m + 1/\varepsilon^{O(d)})$, where the constant in the exponent does not depend on ε or d.*

Proof. First, we transform K into canonical form, and replace it with an $\frac{\varepsilon}{2}$-approximation K' of itself. This can be done in $O(m+1/\varepsilon^{O(d)})$, so that K' is bounded by $O(1/\varepsilon^{(d-1)/2})$ halfspaces (see, e.g., [101]). We then build the data structure to solve APM queries to an accuracy of $(\varepsilon/2)$, so that the total error is ε.

Because the number of nodes increases exponentially as we descend to the leaf level, the most computationally intensive aspect of the remainder of the construction is computing the set X_0, a maximal subset of K whose packing ellipsoids $E''_{\delta_0}(x)$ are pairwise disjoint. To discretize the construction of X_0, we observe that by our remarks at the start of Section 3.2, the Hausdorff distance between K and K_{δ_0} is $\Omega(\delta_0) = \Omega(\varepsilon)$. It follows that each of the ellipsoids $E''_{\delta_0}(x)$ contains a ball of radius $\Omega(\lambda_0 \varepsilon) = \Omega(\varepsilon)$. We restrict the points of X_0 to come from the vertices of a square grid whose side length is half this radius. Since K is in canonical form, it suffices to generate $O(1/\varepsilon^{O(d)})$ grid points. By decreasing the value of ε slightly (by a constant factor), it is straightforward to show that any Delone set can be perturbed so that its centers lie on this grid.

Each Macbeath ellipsoid can be computed in time linear in the number of halfspaces bounding K', which is $O(1/\varepsilon^{O(d)})$ [99]. The maximal set is computed by brute force, repeatedly selecting a point x from the grid, computing $E''_{\delta_0}(x)$, and marking the points of the grid that it covers until all points interior to K are covered. The overall running time is dominated by the product of the number of grid points and the $O(1/\varepsilon^{O(d)})$ time to compute each Macbeath ellipsoid. $\square$

Chapter 4: Non-Euclidean Nearest-Neighbor Searching

Nearest-neighbor searching is a fundamental retrieval problem with numerous applications in fields such as machine learning, data mining, data compression, and pattern recognition. A set of n points, called *sites*, is preprocessed into a data structure such that, given any query point q, it is possible to report the site that is closest to q. The most common formulation involves points in $\mathbb{R}^d$ under the Euclidean metric. For classical pointer-based data structures, the objective is to achieve $O(n)$ storage and $O(\log n)$ query time. When approximation is involved, an important issue is the dependence of the storage and query time on ε, and particularly how rapidly these processing requirements grow with the dimension.

In this chapter, we present a general approach for designing data structures for ANN queries for non-Euclidean distance functions while matching the best bounds for Euclidean ANN queries. In particular, the proposed data structures achieve $O(\log \frac{n}{\varepsilon})$ query time and $O((n/\varepsilon^{d/2}) \log \frac{1}{\varepsilon})$ storage. Thus, we suffer only an extra $\log \frac{1}{\varepsilon}$ factor in the space bounds compared to the best results for Euclidean ε-ANN searching.

4.1 Introduction

Given a set P of n points in $\mathbb{R}^d$, a nearest-neighbor query is given a point $q \in \mathbb{R}^d$, and the objective is to return the closest point to P. It is well known that exact nearest neighbor searching in multi-dimensional spaces is quite inefficient, and so much effort has been devoted to developing efficient approximation algorithms. Given an approximation parameter $\varepsilon > 0$, an *ε-approximate nearest-neighbor query* (or ε-ANN) returns any point whose distance is within a factor of $(1 + \varepsilon)$ of that of the actual nearest neighbor. Throughout, we assume that d is fixed, and we treat n and ε as asymptotic quantities.

The most relevant related work on nearest-neighbor searching with non-Euclidean distances is due to Har-Peled and Kumar. In their paper [52], they proved that ε-ANN searching over a wide variety of distance functions (including additively and multiplicatively weighted sites) could be cast in terms of *minimization diagrams*. They formulated this problem in a very abstract setting, where no explicit reference is made to sites. Instead the input is expressed in terms of abstract properties of the distance functions, such as their growth rates and "sketchability." While this technique is very general, the complexity bounds are much worse than for the corresponding concrete versions. For example, in the case of Euclidean distance with multiplicative weights, in order to achieve logarithmic query time, the storage used is $O((n \log^{d+2} n)/\varepsilon^{2d+2} + n/\varepsilon^{d^2+d})$. Similar results are achieved for a number of other distance functions that are considered in [52].

In this chapter, we present a general approach for designing such data structures

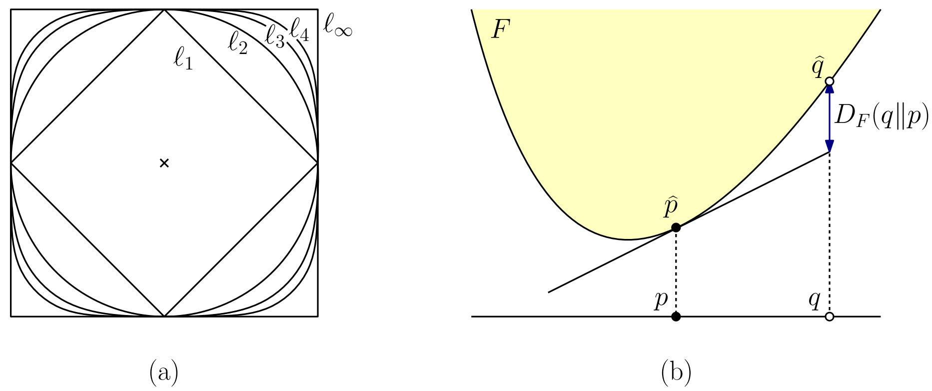


Figure 4.1: (a) Unit balls in different Minkowski norms. (b) Geometric interpretation of the Bregman divergence.

achieving $O(\log \frac{n}{\varepsilon})$ query time and $O((n/\varepsilon^{d/2}) \log \frac{1}{\varepsilon})$ storage. Thus, we suffer only an extra $\log \frac{1}{\varepsilon}$ factor in the space bounds compared to the best results for Euclidean ε-ANN searching. We demonstrate the power of our approach by applying it to a number of natural problems:

Minkowski Distance: The ℓ_k distance (see Figure 4.1(a)) between two points p and q is defined as $\|q - p\|_k = (\sum_{i=1}^{d} |p_i - q_i|^k)^{\frac{1}{k}}$. Our results apply for any real constant $k > 1$.

Multiplicative Weights: Each site p is associated with weight $w_p > 0$ and $f_p(q) = w_p\|q - p\|$. The generalization of the Voronoi diagram to this distance function is known as the *Möbius diagram* [102]. Our results generalize from ℓ_2 to any Minkowski ℓ_k distance, for constant $k > 1$.

Mahalanobis Distance: Each site p is associated with a $d \times d$ positive-definite matrix M_p and $f_p(q) = \sqrt{(p - q)^\intercal M_p (p - q)}$. Mahalanobis distances are widely used in machine learning and statistics. Our results hold under the assumption that for each point p, the ratio between the maximum and minimum eigenvalues of M_p is bounded.

Scaling Distance Functions: Each site p is associated with a closed convex body K_p whose interior contains the origin, and $f_p(q)$ is the smallest r such that $(q - p)/r \in K_p$ (or zero if $q = p$). (These are also known as *convex distance functions* [103].) These generalize and customize normed metric spaces by allowing metric balls that are not centrally symmetric and allowing each site to have its own distance function.

Scaling distance functions generalize the Minkowski distance, multiplicative weights, and the Mahalanobis distance. Our results hold under the assumption that the convex body K_p inducing the distance function satisfies certain assumptions. First, it needs to be *fat* in the sense that it can be sandwiched between two Euclidean balls centered at the origin whose radii differ by a constant factor. Second, it needs to be *smooth* in the sense that the radius of curvature for every point on K_p's boundary is within a constant factor of its diameter. (Formal definitions will be given in Section 4.4.2.)

Theorem 4 (ANN for Scaling Distances)**.** *Given an approximation parameter* $0 < \varepsilon \leq 1$ *and a set* S *of* n *sites in* $\mathbb{R}^d$ *where each site* $p \in S$ *is associated with a fat, smooth convex body* $K_p \subset \mathbb{R}^d$ *(as defined above), there exists a data structure that can answer* ε*-approximate nearest-neighbor queries with respect to the respective scaling distance functions defined by* K_p *with*

$$\textit{Query time: } O\left(\log \frac{n}{\varepsilon}\right) \quad \textit{and} \quad \textit{Space: } O\left(\frac{n \log \frac{1}{\varepsilon}}{\varepsilon^{d/2}}\right).$$

Another important application that we consider is the Bregman divergence. Bregman divergences generalize the squared Euclidean distance [104], the Kullback-Leibler divergence (also known as relative entropy) [105], and the Itakura-Saito distance [106] among others. They have numerous applications in machine learning and computer vision [107, 108].

Bregman Divergence: Given an open convex domain $\mathbb{X} \subseteq \mathbb{R}^d$, a strictly convex and differentiable real-valued function F on $\mathbb{X}$, and $q, p \in \mathbb{X}$, the *Bregman divergence* of q from p is

$$D_F(q,p) \;=\; F(q) - (F(p) + \nabla F(p) \cdot (q-p)),$$

where ∇F denotes the gradient of F and "$\cdot$" is the standard dot product.

The Bregman divergence has the following geometric interpretation (see Figure 4.1(b)). Let $\widehat{p}$ denote the vertical projection of p onto the graph of F, that is, $(p, F(p))$, and define $\widehat{q}$ similarly. $D_F(q,p)$ is the vertical distance between $\widehat{q}$ and the hyperplane tangent to F at the point $\widehat{p}$. Equivalently, $D_F(q,p)$ is just the error that results by estimating $F(q)$ by a linear model at p.

The Bregman divergence possibly lacks many of the properties of typical distance functions. It is generally not symmetric, that is, $D_F(q,p) \neq D_F(p,q)$, and it generally does not satisfy the triangle inequality, but it is a convex function in its first argument. Throughout, we treat the first argument q as the query point and the second argument p as the site, but it is possible to reverse these through dualization [104].

Data structures have been presented for answering exact nearest-neighbor queries in the Bregman divergence by Cayton [109] and Nielson *et al.* [110], but no complexity analysis was given. Worst-case bounds have been achieved by imposing restrictions on the function F. Various different complexity measures have been proposed, including the following. Given a parameter $\mu \geq 1$, and letting $\|p - q\|$ denote the Euclidean distance between p and q:

- D_F is *μ-asymmetric* if for all $p, q \in \mathbb{X}$, $D_F(q, p) \leq \mu D_F(p, q)$.
- D_F is *μ-similar*[1] if for all $p, q \in \mathbb{X}$, $\|q - p\|^2 \leq D_F(q, p) \leq \mu \|q - p\|^2$.

Abdullah *et al.* [112] presented data structures for answering ε-ANN queries for decomposable[2] Bregman divergences in spaces of constant dimension under the assumption of bounded similarity. Later, Abdullah and Venkatasubramanian [113] established lower bounds on the complexity of Bregman ANN searching under the assumption of bounded asymmetry.

Our results for ANN searching in the Bregman divergence are stated below. They hold under a related measure of complexity, called *τ-admissibility*, which is more inclusive (that is, weaker) than μ-similarity, but seems to be more restrictive than μ-asymmetry. It is defined in Section 4.5.1, where we also explore the relationships

[1]Our definition of μ-similarity differs from that of [111]. First, we have replaced $1/\mu$ with μ for compatibility with asymmetry. Second, their definition allows for any Mahalanobis distance, not just Euclidean. This is a trivial distinction in the context of nearest-neighbor searching, since it is possible to transform between such distances by applying an appropriate positive-definite linear transformation to the query space.

[2]The sum of one-dimensional Bregman divergences.

between these measures.

Theorem 5 (ANN for Bregman Divergences)**.** *Given a τ-admissible Bregman divergence D_F for a constant τ defined over an open convex domain $\mathbb{X} \subseteq \mathbb{R}^d$, a set S of n sites in $\mathbb{R}^d$, and an approximation parameter $0 < \varepsilon \leq 1$, there exists a data structure that can answer ε-approximate nearest-neighbor queries with respect to D_F with*

$$\textit{Query time: } O\left(\log \frac{n}{\varepsilon}\right) \quad \textit{and} \quad \textit{Space: } O\left(\frac{n \log \frac{1}{\varepsilon}}{\varepsilon^{d/2}}\right).$$

Note that our results are focused on the *existence* of these data structures, and construction is not discussed. While we see no significant impediments to their efficient construction by modifying the constructions of related data structures, a number of technical results would need to be developed. We therefore leave the question of efficient construction as a rather technical but nonetheless important open problem.

4.1.1 Methods

Our solutions are all based on the application of a technique, called *convexification*. Recently, Arya et al. showed how to efficiently answer several approximation queries with respect to convex polytopes [28, 34, 50, 114], including polytope membership, ray shooting, directional width, and polytope intersection. As mentioned above, the linearization technique using the lifting transformation can be used to produce convex polyhedra for the sake of answering ANN queries, but it is applicable only to the Euclidean distance (or more accurately the squared Euclidean distance and the related power distance [115]). In the context of approximation, polytopes are not required. The convex approximation methods described above can be adapted to work on any convex body, even one with curved boundaries. This provides us with an additional degree of flexibility. Rather than applying a transformation to linearize the various distance functions, we can go a bit overboard and "convexify" them.

Convexification techniques have been used in non-linear optimization for decades [116], for example the αBB optimization method locally convexifies constraint functions to produce constraints that are easier to process [117]. However, we are unaware of prior applications of this technique in computational geometry in the manner that we use it. (For an alternate use, see [17].)

The general idea involves the following two steps. First, we apply a quadtree-like approach to partition the query space (that is, $\mathbb{R}^d$) into cells so that the restriction of each distance function within each cell has certain "nice" properties, which make it possible to establish upper bounds on the gradients and the eigenvalues of their Hessians. We then add to each function a common "convexifying" function whose Hessian has sufficiently small (in fact negative) eigenvalues, so that all the functions become concave (see Figure 4.3 in Section 4.3 below). We then exploit the fact that the lower envelope of concave functions is concave. The region lying under this lower envelope can be approximated by standard techniques, such as the ray-shooting data structure of [34]. We show that if the distance functions satisfy some *admissibility conditions*, this can be achieved while preserving the approximation errors.

The rest of this chapter is organized as follows. In the next section we present definitions and preliminary results. Section 4.3 discusses the concept of convexification, and how it is applied to vertical ray shooting on the minimization diagram of sufficiently well-behaved functions. In Section 4.4, we present our solution to ANN searching for scaling distance functions, proving Theorem 4. In Section 4.5, we do the same for the case of Bregman divergence, proving Theorem 5.

4.2 Preliminaries

In this section we present a number of definitions and results that will be useful throughout this chapter.

4.2.1 Notation and Assumptions

Given a function $f : \mathbb{R}^d \to \mathbb{R}$, its *graph* is the set of $(d+1)$-dimensional points $(x, f(x))$, its *epigraph* is the set of points on or above the graph, and its *hypograph* is the set of points on or below the graph (where the $(d+1)$-st axis is directed upwards). The *level set* (also called *level surface* if $d \geq 3$) of f is the set of points $x \in \mathbb{R}^d$ for which f has the same value.

The gradient and Hessian of a function generalize the concepts of the first and second derivative to a multidimensional setting. The *gradient* of f, denoted ∇f, is defined as the vector field $\left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_d}\right)^\intercal$. The gradient vector points in a direction in which the function grows most rapidly, and it is orthogonal to the level surface. For any point x and any unit vector v, the rate of change of f along v is given by the dot product $\nabla f(x) \cdot v$. The *Hessian* of f at x, denoted $\nabla^2 f(x)$, is a $d \times d$ matrix of second-order partial derivatives at x. For twice continuously differentiable functions, $\nabla^2 f(x)$ is symmetric, implying that it has d (not necessarily distinct) real eigenvalues.

Given a d-vector v, let $\|v\|$ denote its length under the *Euclidean norm*, and the *Euclidean distance* between points p and q is $\|q - p\|$. Given a $d \times d$ matrix A, its *spectral norm* is $\|A\| = \sup\{\|Ax\| \,/\, \|x\| \,:\, x \in \mathbb{R}^d \text{ and } x \neq 0\}$. Since the Hessian is a symmetric matrix, it follows that $\|\nabla^2 f(x)\|$ is the largest absolute value attained by the eigenvalues of $\nabla^2 f(x)$.

A real-valued function f defined on a nonempty subset $\mathbb{X}$ of $\mathbb{R}^d$ is *convex* if the domain $\mathbb{X}$ is convex and for any $x, y \in \mathbb{X}$ and $\alpha \in [0, 1]$, $f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$, and it is *concave* if $-f$ is convex. A twice continuously differentiable function on a convex domain is convex if and only if its Hessian matrix is positive semidefinite in the interior of the domain. It follows that all the eigenvalues of the Hessian of a convex function are nonnegative.

Given a function $f : \mathbb{R}^d \to \mathbb{R}$ and a closed Euclidean ball B (or generally any closed bounded region), let $f^+(B)$ and $f^-(B)$ denote the maximum and minimum values, respectively, attained by $f(x)$ for $x \in B$. Similarly, define $\|\nabla f^+(B)\|$ and $\|\nabla^2 f^+(B)\|$ to be the maximum values of the norms of the gradient and Hessian, respectively, for any point in B.

4.2.2 Minimization Diagrams and Ray Shooting

Consider a convex domain $\mathbb{X} \subseteq \mathbb{R}^d$ and a set of functions $\mathcal{F} = \{f_1, \ldots, f_m\}$, where $f_i : \mathbb{X} \to \mathbb{R}^+$. Let $\mathcal{F}_{\min}$ denote the associated *lower-envelope function*, that is $\mathcal{F}_{\min}(x) = \min_{1 \leq i \leq m} f_i(x)$. As Har-Peled and Kumar [52] observed, for any

$\varepsilon > 0$, we can answer ε-ANN queries on any set S by letting f_i denote the distance function to the ith site, and computing any index i (called a *witness*) such that $f_i(q) \leq (1+\varepsilon)\mathcal{F}_{\min}(q)$.

We can pose this as a geometric approximation problem in one higher dimension. Consider the hypograph in $\mathbb{R}^{d+1}$ of $\mathcal{F}_{\min}$, and let us think of the $(d+1)$st axis as indicating the *vertical* direction. Answering ε-ANN queries in the above sense can be thought of as approximating the result of a vertical ray shot upwards from the point $(q, 0) \in \mathbb{R}^{d+1}$ until it hits the lower envelope, where the allowed approximation error is $\varepsilon\mathcal{F}_{\min}(q)$. Because the error is relative to the value of $\mathcal{F}_{\min}(q)$, this is called a *relative ε-AVR query*. It is also useful to consider a variant in which the error is absolute. An *absolute ε-AVR query* returns any witness i such that $f_i(q) \leq \varepsilon + \mathcal{F}_{\min}(q)$ (see Fig. 4.2).

The hypograph of a general minimization diagram can be unwieldy. Our approach to answer AVR queries efficiently will involve subdividing space into regions such that within each region it is possible to transform the hypograph into a convex shape. In the next section, we will describe this transformation. Given this, our principal utility for answering ε-AVR queries efficiently is encapsulated in the following lemma (see Figure 4.2). The proof presented below is based on the constructions in [114].

Lemma 11. (Answering ε-AVR Queries) *Consider a unit ball $B \subseteq \mathbb{R}^d$ and a family of concave functions $\mathcal{F} = \{f_1, \ldots, f_m\}$ defined over B such that for all $1 \leq i \leq m$ and $x \in B$, $f_i(x) \in [0, 1]$ and $\|\nabla f_i(x)\| \leq 1$. Then, for any $0 < \varepsilon \leq 1$, there is a*

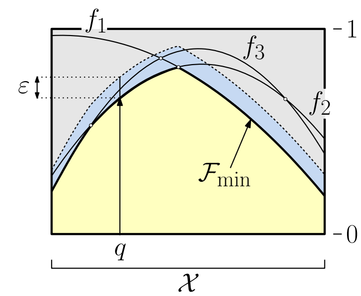


Figure 4.2: Approximate AVR query assuming absolute errors. For the query q, the exact answer is f_2, but f_3 would be acceptable.

data structure that can answer absolute ε-AVR queries in time $O(\log \frac{1}{\varepsilon})$ and storage $O((\frac{1}{\varepsilon})^{d/2})$.

Proof. We adapt an approach for ray-shooting described in [114], which reduces ray-shooting to walking the ray through a collection of ellipsoids. In order to apply this approach, we will define two convex bodies K^- and K^+, where $K^- \subset K^+$. The aforementioned ellipsoids will be contained withing K^+ and will cover K^-. The number of ellipsoids will be $O(1/\varepsilon^{(d-1)/2}$ and each vertical ray will pass through $O(\log \frac{1}{\varepsilon})$ ellipsoids of this collection. Knowing the last ellipsoid of this collection that is hit by an upward ray will provide the answer to an ε-AVR query.

In order to apply this approach, let us translate space so that B is centered at the origin, and let us translate the functions of $\mathcal{F}$ up by one unit, so that the function values lie in $[1, 2]$. Let C denote a semi-infinite convex cylinder in $\mathbb{R}^{d+1}$ whose central axis is vertical, whose cross section is B, and which is bounded below by the horizontal hyperplane $f(x) = -\frac{1}{2}$. Let K^- be the convex body formed by intersecting C with epigraph of the lower envelope function $F_{\min}$. To define

Next, to apply the method given in [114] we enclose K within an expanded body K^+ as follows.

We will follow the strategy presented in [27] for answering ε-ANN queries. It combines (1) a data structure for answering approximate central ray-shooting queries, in which the rays originate from a common point and (2) an approximation-preserving reduction from vertical to central ray-shooting queries [34].

Let K denote a closed convex body that is represented as the intersection of a finite set of halfspaces. We assume that K is centrally γ-fat for some constant γ (recall the definition from Section 4.4.2). An *ε-approximate central ray-shooting query* (ε-ACR query) is given a query ray that emanates from the origin and returns the index of one of K's bounding hyperplanes h whose intersection with the ray is within distance $\varepsilon \cdot \mathrm{diam}(K)$ of the true contact point with K's boundary. We will make use of the following result, which is paraphrased from [34].

Approximate Central Ray-Shooting: Given a convex polytope K in $\mathbb{R}^d$ that is centrally γ-fat for some constant γ and an approximation parameter $0 < \varepsilon \leq 1$, there is a data structure that can answer ε-ACR queries in time $O(\log \frac{1}{\varepsilon})$ and storage $O(1/\varepsilon^{(d-1)/2})$.

As in Section 4 of [34], we can employ a projective transformation that converts vertical ray shooting into central ray shooting. While the specific transformation presented there was tailored to work for a set of hyperplanes that are tangent to a paraboloid, a closer inspection reveals that the reduction can be generalized (with a change in the constant factors) provided that the following quantities are all bounded above by a constant: (1) the diameter of the domain of interest, (2) the difference between the maximum and minimum function values throughout this domain, and (3) the absolute values of the slopes of the hyperplanes (or equivalently, the norms of the gradients of the functions defined by these hyperplanes). This projective transformation produces a convex body in $\mathbb{R}^{d+1}$ that is centrally γ-fat for some constant γ, and it preserves relative errors up to a constant factor.

Therefore, by applying this projective transformation, we can reduce the problem of answering ε-AVR queries in dimension d for the lower envelope of a set of linear functions to the aforementioned ACR data structure in dimension $d+1$. The only remaining issue is that the functions of $\mathcal{F}$ are concave, not necessarily linear. Thus, the output of the reduction is a convex body bounded by curved patches, not a polytope. We address this by applying Dudley's Theorem [19] to produce a polytope that approximates this convex body to an absolute Hausdorff error of $\varepsilon/2$. (In particular, Dudley's construction samples $O(1/\varepsilon^{d/2})$ points on the boundary of the convex body, and forms the approximation by intersecting the supporting hyperplanes at each of these points.) We then apply the ACR data structure to this approximating polytope, but with the allowed error parameter set to $\varepsilon/2$. The combination of the two errors, results in a total allowed error of ε.

In order to obtain a witness, each sample point from Dudley's construction is associated with the function(s) that are incident to that point. We make the general position assumption that no more than $d+1$ functions can coincide at any point on the lower envelope of $\mathcal{F}$, and hence each sample point is associated with a constant number of witnesses. The witness produced by the ACR data structure will be one of the bounding hyperplanes. We check each of the functions associated with the sample point that generated this hyperplane, and return the index of the function having the smallest function value. □

4.3 Convexification

In this section we discuss the key technique underlying many of our results. As mentioned above, our objective is to answer ε-AVR queries with respect to the minimization diagram, but this is complicated by the fact that it does not bound a convex set.

In order to overcome this issue, let us make two assumptions. First, we restrict the functions to a bounded convex domain, which for our purposes may be taken to be a closed Euclidean ball B in $\mathbb{R}^d$. Second, let us assume that the functions are smooth, implying in particular that each function f_i has a well defined gradient ∇f_i and Hessian $\nabla^2 f_i$ for every point of B. As mentioned above a function f_i is convex (resp., concave) over B if and only if all the eigenvalues of $\nabla^2 f_i(x)$ are nonnegative (resp., nonpositive). Intuitively, if the functions f_i are sufficiently well-behaved it is possible to compute upper bounds on the norms of the gradients and Hessians throughout B. Given $\mathcal{F}$ and B, let Λ^+ denote an upper bound on the largest eigenvalue of $\nabla^2 f_i(x)$ for any function $f_i \in \mathcal{F}$ and for any point $x \in B$.

We will apply a technique called *convexification* from the field of nonconvex optimization [116, 117]. If we add to f_i any function whose Hessian has a maximum eigenvalue at most $-\Lambda^+$, we will effectively "overpower" all the upward curving terms, resulting in a function having only nonpositive eigenvalues, that is, a concave function.[3] The lower envelope of concave functions is concave, and so techniques for convex approximation (such as Lemma 11) can be applied to the hypograph of the resulting lower-envelope function.

To make this more formal, let $p \in \mathbb{R}^d$ and $r \in \mathbb{R}$ denote the center point and radius of B, respectively. Define a function ϕ (which depends on B and Λ^+) to be

$$\phi(x) \;=\; \frac{\Lambda^+}{2}\left(r^2 - \sum_{j=1}^{d}(x_j - p_j)^2\right) \;=\; \frac{\Lambda^+}{2}(r^2 - \|x-p\|^2).$$

It is easy to verify that ϕ evaluates to zero along B's boundary and is positive within B's interior. Also, for any $x \in \mathbb{R}^d$, the Hessian of $\|x-p\|^2$ (as a function of x) is a $d \times d$ diagonal matrix $2I$, and therefore $\nabla^2\phi(x) = -\Lambda^+ I$. Now, for $1 \le i \le m$, define $\widehat{f}_i(x) \;=\; f_i(x) + \phi(x)$ and

$$\widehat{F}_{\min}(x) \;=\; \min_{1 \le i \le m} \widehat{f}_i(x) \;=\; \mathcal{F}_{\min}(x) + \phi(x).$$

Because all the functions are subject to the same offset at each point x, $\widehat{F}_{\min}$ preserves the relevant combinatorial structure of $\mathcal{F}_{\min}$, and in particular f_i yields the minimum value to $\mathcal{F}_{\min}(x)$ at some point x if and only if $\widehat{f}_i$ yields the minimum value to $\widehat{F}_{\min}(x)$. Absolute vertical errors are preserved as well. Observe that $\widehat{F}_{\min}(x)$ matches the value of $\mathcal{F}_{\min}$ along B's boundary and is larger within its interior. Also, since $\nabla^2\phi(x) = -\Lambda^+ I$, it follows from elementary linear algebra that each eigenvalue of $\nabla^2\widehat{f}_i(x)$ is smaller than the corresponding eigenvalue of $\nabla^2 f_i(x)$ by Λ^+. Thus, all the eigenvalues of $\widehat{f}_i(x)$ are nonpositive, and so $\widehat{f}_i$ is concave over B. In turn, this implies that $\widehat{F}_{\min}$ is concave, as desired. We will show that, when properly applied,

[3] While this intuition is best understood for convex functions, it can be applied whenever there is an upper bound on the maximum eigenvalue.

relative errors are nearly preserved, and hence approximating the convexified lower envelope yields an approximation to the original lower envelope.

4.3.1 A Short Example

As a simple application of this technique, consider the following problem. Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be a collection of m multivariate polynomial functions over $\mathbb{R}^d$ each of constant degree and having coefficients whose absolute values are $O(1)$ (see Figure 4.3(a)). It is known that the worst-case combinatorial complexity of the lower envelope of algebraic functions of fixed degree in $\mathbb{R}^d$ lies between $\Omega(n^d)$ and $O(n^{d+\alpha})$ for any $\alpha > 0$ [118], which suggests that any exact solution to computing a point on the lower envelope $\mathcal{F}_{\min}$ will either involve high space or high query time.

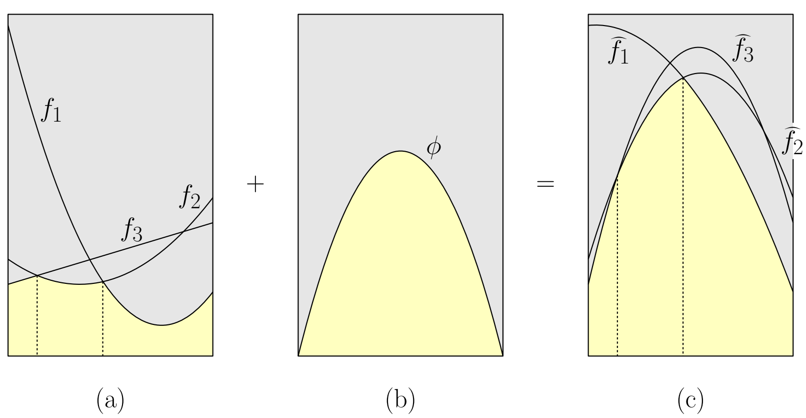


Figure 4.3: Convexification.

Let us consider a simple approximate formulation by restricting $\mathcal{F}$ to a unit d-dimensional Euclidean ball B centered at the origin. Given a parameter $\varepsilon > 0$, the objective is to compute for any query point $q \in \mathbb{R}^d$ an *absolute ε-approximation* by returning the index of a function f_i such that $f_i(q) \leq \mathcal{F}_{\min}(q) + \varepsilon$. (While relative errors are usually desired, this simpler formulation is sufficient to illustrate how convexification works.) Since the degrees and coefficients are bounded, it follows that for each $x \in B$, the norms of the gradients and Hessians for each function f_i are bounded. A simple naive solution would be to overlay B with a grid with cells of diameter $\Theta(\varepsilon)$ and compute the answer for a query point centered within each grid cell. Because the gradients are bounded, the answer to the query for the center point is an absolute ε-approximation for any point in the cell. This produces a data structure with space $O((\frac{1}{\varepsilon})^d)$.

To produce a more space-efficient solution, we apply convexification. Because the eigenvalues of the Hessians are bounded for all $x \in B$ and all functions f_i, it follows that there exists an upper bound $\Lambda^+ = O(1)$ on all the Hessian eigenvalues. Therefore, by computing the convexifying function ϕ described above (see Figure 4.3(b)) to produce the new function $\widehat{F}_{\min}$ (see Figure 4.3(c)) we obtain a concave function. It is easy to see that ϕ has bounded gradients and therefore so does $\widehat{F}_{\min}$. The hypograph of the resulting function when suitably trimmed is a convex body of constant diameter residing in $\mathbb{R}^{d+1}$. After a suitable scaling (which will be described later in Lemma 13), the functions can be transformed so that we may apply Lemma 11 to answer approximate vertical ray-shooting queries in time $O(\log \frac{1}{\varepsilon})$ with storage $O((\frac{1}{\varepsilon})^{d/2})$. This *halves* the exponential dependence in the dimension over the simple approach.

4.3.2 Admissible Distance Functions

A key issue in the convexification process is how approximation errors are affected. We will show that if the functions satisfy certain admissibility properties, then this will be the case. We are given a domain $\mathbb{X} \subseteq \mathbb{R}^d$, and we assume that each distance function is associated with a defining site $p \in \mathbb{X}$. Consider a distance function $f_p : \mathbb{X} \to \mathbb{R}^+$ with a well-defined gradient and Hessian for each point of $\mathbb{X}$.[4] Given $\tau > 0$, we say that f_p is τ-*admissible* if for all $x \in \mathbb{X}$:

(*i*) $\|\nabla f_p(x)\| \, \|x - p\| \;\leq\; \tau f_p(x)$, and

(*ii*) $\|\nabla^2 f_p(x)\| \, \|x - p\|^2 \;\leq\; \tau^2 f_p(x)$.

Intuitively, an admissible function exhibits growth rates about the site that are polynomially upper bounded. For example, it is easy to prove that $f_p(x) = \|x - p\|^c$ is $O(c)$-admissible, for any $c \geq 1$.

Admissibility implies bounds on the magnitudes of the function values, gradients, and Hessians. Given a Euclidean ball B and site p, we say that B and p are β-*separated* if $\mathbf{d}(p, B)/\mathrm{diam}(B) \geq \beta$ (where $\mathbf{d}(p, B)$ is the minimum Euclidean distance between p and B and $\mathrm{diam}(B)$ B's diameter). The following lemma presents upper bounds on $f^+(B)$, $\|\nabla f^+(B)\|$, and $\|\nabla^2 f^+(B)\|$ in terms of these quantities. (Recall the definitions from Section 4.2.1.)

[4] This assumption is really too strong, since distance functions often have undefined gradients or Hessians at certain locations (e.g., the sites themselves). For our purposes it suffices that the gradient and Hessian are well defined at any point within the region where convexification will be applied.

Lemma 12. *Consider an open convex domain $\mathbb{X}$, a site $p \in \mathbb{X}$, a τ-admissible distance function f_p, and a Euclidean ball $B \subset \mathbb{X}$. If B and p are $(\tau\kappa)$-separated for $\kappa > 1$, then:*

(i) $f_p^+(B) \;\le\; f_p^-(B)\kappa/(\kappa-1)$,

(ii) $\|\nabla f_p^+(B)\| \;\le\; f_p^+(B)/(\kappa \mathrm{diam}(B))$, *and*

(iii) $\|\nabla^2 f_p^+(B)\| \;\le\; f_p^+(B)/(\kappa \mathrm{diam}(B))^2$.

Proof. To prove (i), let x^+ and x^- denote the points of B that realize the values of $f_p^+(B)$ and $f_p^-(B)$, respectively. By applying the mean value theorem, there exists a point $s \in \overline{x^- x^+}$ such that $f_p^+(B) - f_p^-(B) = \nabla f_p(s) \cdot (x^+ - x^-)$. By the Cauchy-Schwarz inequality

$$f_p^+(B) - f_p^-(B) \;=\; \nabla f_p(s) \cdot (x^+ - x^-) \;\le\; \|\nabla f_p(s)\| \, \|x^+ - x^-\|.$$

By τ-admissibility, $\|\nabla f_p(s)\| \le \tau f_p(s)/\|s-p\|$, and since $x^+, x^-, s \in B$, we have $\|x^+ - x^-\|/\|s-p\| \le \mathrm{diam}(B)/\mathbf{d}(p,B) \le 1/(\tau\kappa)$. Thus,

$$f_p^+(B) - f_p^-(B) \;\le\; \frac{\tau f_p(s)}{\|s-p\|}\|x^+ - x^-\| \;\le\; \frac{\tau f_p(s)}{\tau\gamma} \;\le\; \frac{f_p^+(B)}{\kappa}.$$

This implies that $f_p^+(B) \le f_p^-(B)\kappa/(\kappa-1)$, establishing (i).

To prove (ii), consider any $x \in B$. By separation, $\mathbf{d}(p,B) \ge \tau\kappa \mathrm{diam}(B)$. Combining this with τ-admissibility and (i), we have

$$\|\nabla f_p(x)\| \;\le\; \frac{\tau f_p(x)}{\|x-p\|} \;\le\; \frac{\tau f_p^+(B)}{\mathbf{d}(p,B)} \;\le\; \frac{\tau f_p^+(B)}{\tau\kappa \mathrm{diam}(B)} \;=\; \frac{f_p^+(B)}{\kappa \mathrm{diam}(B)}.$$

This applies to any $x \in B$, thus establishing (ii).

To prove (iii), again consider any $x \in B$. By separation and admissibility, we have

$$\|\nabla^2 f_p(x)\| \;\leq\; \frac{\tau^2 f_p(x)}{\|x-p\|^2} \;\leq\; \frac{\tau^2 f_p^+(B)}{\mathbf{d}^2(p,B)} \;\leq\; \frac{f_p^+(B)}{(\kappa\,\mathrm{diam}(B))^2}.$$

This applies to any $x \in B$, thus establishing (iii). □

4.3.3 Convexification and Ray Shooting

A set $\mathcal{F} = \{f_1, \ldots, f_m\}$ of τ-admissible functions is called a *τ-admissible family of functions*. Let $\mathcal{F}_{\min}$ denote the associated lower-envelope function. In Lemma 11 we showed that absolute ε-AVR queries could be answered efficiently in a very restricted context. This will need to be generalized the purposes of answering ANN queries, however.

The main result of this section states that if the sites defining the distance functions are sufficiently well separated from a Euclidean ball, then (through convexification) ε-AVR queries can be efficiently answered. The key idea is to map the ball and functions into the special structure required by Lemma 11, and to analyze how the mapping process affects the gradients and Hessians of the functions.

Lemma 13. (Convexification & Ray-Shooting) *Consider a Euclidean ball $B \in \mathbb{R}^d$ and a family of τ-admissible distance functions $\mathcal{F} = \{f_1, \ldots, f_m\}$ over B such that each associated site is (2τ)-separated from B. Given any $\varepsilon > 0$, there exists a data structure that can answer relative ε-AVR queries with respect to $\mathcal{F}_{\min}$ in time $O(\log \frac{1}{\varepsilon})$ with storage $O((\frac{1}{\varepsilon})^{d/2})$.*

Proof. We will answer approximate vertical ray-shooting queries by a reduction to the data structure given in Lemma 11 for answering approximate central ray-shooting queries. In order to apply this lemma, we need to transform the problem into the canonical form prescribed by that lemma.

We may assume without loss of generality that f_1 is the function that minimizes $f_1^-(B)$ among all the functions in $\mathcal{F}$. By Lemma 12(i) (with $\kappa = 2$), $f_1^+(B) \leq 2f_1^-(B)$. For all i, we may assume that $f_i^-(B) \leq 2f_1^-(B)$ for otherwise this function is greater than f_1 throughout B, and hence it does not contribute to $\mathcal{F}_{\min}$. Under this assumption, it follows that $f_i^+(B) \leq 4f_1^-(B)$.

In order to convert these functions into the desired form, define $h = 5f_1^-(B)$, $r = \text{radius}(B)$, and let $c \in \mathbb{R}^d$ denote the center of B. Let B_0 be a unit ball centered at the origin, and for any $x \in B_0$, let $x' = rx + c$. Observe that $x \in B_0$ if and only if $x' \in B$. For each i, define the normalized distance function

$$g_i(x) \;=\; \frac{f_i(x')}{h}.$$

We assert that these functions satisfy the following properties. They are straightforward consequences of admissibility and separation, but for the sake of completeness, we present the derivations below.

Lemma 14. *Each of the normalized distance functions $g(x) = f(x')/h$ defined in the proof of Lemma 13 satisfy the following properties:*

(a) $g^+(B_0) \leq 4/5$ *and* $g^-(B_0) \geq 1/5$,

(b) $\|\nabla g^+(B_0)\| \leq 1/2$, *and*

(c) $\|\nabla^2 g^+(B_0)\| \leq 1/4$.

Proof. For any $x \in B_0$, we have

$$g(x) \;\leq\; \frac{f^+(B)}{h} \;\leq\; \frac{2f^-(B)}{h} \;\leq\; \frac{4f_1^-(B)}{h} \;=\; \frac{4}{5},$$

and

$$g(x) \;\geq\; \frac{f^-(B)}{h} \;\geq\; \frac{f_1^-(B)}{h} \;=\; \frac{1}{5},$$

which establishes (a).

Before establishing (b) and (c), observe that by the chain rule in differential calculus, $\nabla g(x) = (r/h)\nabla f(x')$ and $\nabla^2 g(x) = (r^2/h)\nabla^2 f(x')$. (Recall that x and x' are corresponding points in B_0 and B, respectively.) Since B_0 is a unit ball, $\mathrm{diam}(B_0) = 2$. Thus, by Lemma 12(ii) (with $\kappa = 2$), we have

$$\|\nabla g(x)\| \;=\; \frac{r}{h}\|\nabla f(x')\| \;\leq\; \frac{r}{h}\frac{f^+(B)}{2(2r)} \;\leq\; \frac{1}{4},$$

which establishes (b). By Lemma 12(iii),

$$\|\nabla^2 g(x)\| \;=\; \frac{r^2}{h}\|\nabla f(x')\| \;\leq\; \frac{r^2}{h}\frac{f^+(B)}{(2(2r))^2} \;\leq\; \frac{1}{16},$$

which establishes (c). $\square$

Next, we convexify these functions. To do this, define $\phi(x) = (1 - \|x\|^2)/8$. Observe that for any $x \in B_0$, $\phi(x) \in [0, 1/8]$ and $\|\nabla\phi(x)\| = \|x\|/4$ and $\nabla^2\phi(x)$ is the diagonal matrix $-(1/4)I$. Define

$$\widehat{g}_i(x) \;=\; g_i(x) + \phi(x).$$

It is easily verified that these functions satisfy the following properties.

(a′) $\widehat{g}_i^+(B_0) \leq 1$ and $\widehat{g}_i^-(B_0) \geq 1/5$

(b′) $\|\nabla\widehat{g}_i^+(B_0)\| \leq \|\nabla g_i^+(B_0)\| + \|\nabla\phi^+(B_0)\| < 1$

(c′) $\|\nabla^2\widehat{g}_i^+(B_0)\| \leq \|\nabla^2 g_i^+(B_0)\| - (1/4) \leq 0$

By property (c′), these functions are concave over B_0. Given that $\widehat{g}_i^-(B_0) \geq 1/5$, in order to answer AVR queries to a relative error of ε, it suffices to answer AVR queries to an absolute error of $\varepsilon' = \varepsilon/5$. Therefore, we can apply Lemma 11 (using ε' in place of ε) to obtain a data structure that answers relative ε-AVR queries with respect to $\mathcal{F}_{\min}$ in time $O(\log \frac{1}{\varepsilon})$ with storage $O((\frac{1}{\varepsilon})^{d/2})$, as desired. $\square$

Armed with this tool, we are now in a position to present the data structures for answering ε-ANN queries for each of our applications, which we do in the subsequent sections.

4.4 Search Queries with Convex Distance Functions

Recall that in a scaling distance we are given a convex body K that contains the origin in its interior, and the distance from a query point q to a site p is defined to be zero if $p = q$ and otherwise it is the smallest r such that $(q - p)/r \in K$.[5] The body K plays the role of a unit ball in a normed metric, but we do not require that the body be centrally symmetric. In this section we establish Theorem 4 by demonstrating a data structure for answering ε-ANN queries given a set S of n

[5]This can be readily generalized to squared distances, that is, the smallest r such that $(q-p)/\sqrt{r} \in K$. A relative error of $1+\varepsilon$ in the squared distance, reduces to computing a $\sqrt{1+\varepsilon}$ relative error in the original distance. Since $\sqrt{1+\varepsilon} \approx (1+\varepsilon/2)$ for small ε, our approach can be applied but with a slightly smaller value of ε. This generalizes to any constant power.

sites, where each site p_i is associated with a scaling distance whose unit ball is a fat, smooth convex body.

Before presenting the data structure, we present two preliminary results. The first, given in Section 4.4.1, explains how to subdivide space into a number of regions, called *cells*, that possess nice separation properties with respect to the sites. The second, given in Section 4.4.2, presents key technical properties of scaling functions whose unit balls are fat and smooth.

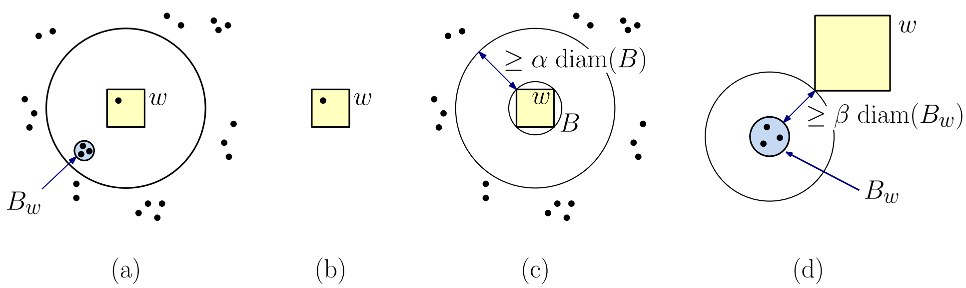


Figure 4.4: Basic separation properties for Lemma 15.

4.4.1 Separation Properties

In order to apply the convexification process, we will first subdivide space into regions, each of which satisfies certain separation properties with respect to the sites S. This subdivision results from a height-balanced variant of a quadtree, called a *balanced box decomposition tree* (or BBD tree) [119]. Each cell of this decomposition is either a quadtree box or the set-theoretic difference of two such boxes. Each leaf cell is associated with an auxiliary ANN data structure for the query points in the cell, and together the leaf cells subdivide all of $\mathbb{R}^d$.

The separation properties are essentially the same as those of the AVD data structure of [47]. For any leaf cell w of the decomposition, the sites can be partitioned into three subsets, any of which may be empty (see Figure 4.4(a)). First, a single site may lie within w. Second, a subset of sites, called the *outer cluster*, is well-separated from the cell. Finally, there may be a dense cluster of points, called the *inner cluster*, that lie within a ball B_w that is well-separated from the cell. After locating the leaf cell containing the query point, the approximate nearest neighbor is computed independently for each of these subsets (by a method to be described later), and the overall closest is returned. The next lemma formalizes these separation properties. It follows easily from Lemma 6.1 in [120]. Given a BBD-tree cell w and a point $p \in \mathbb{R}^d$, let $\mathbf{d}(p, w)$ denote the minimum Euclidean distance from p to any point in w.

Lemma 15 (Basic Separation Properties)**.** *Given a set S of n points in $\mathbb{R}^d$ and real parameters $\alpha, \beta \geq 2$. It is possible to construct a BBD tree T with $O(\alpha^d n \log \beta)$ nodes, whose leaf cells cover $\mathbb{R}^d$ and for every site $p \in S$, either*

(*i*) *it lies within w, but there can be at most one site for which this holds (see Figure 4.4(b)),*

(*ii*) (outer cluster) *letting B denote the smallest Euclidean ball enclosing w, $\boldsymbol{d}(p, B) \geq \alpha \cdot \mathrm{diam}(B)$ (see Figure 4.4(c)), or*

(*iii*) (inner cluster) *there exists a ball B_w associated with w such that $\boldsymbol{d}(B_w, w) \geq \beta \cdot \mathrm{diam}(B_w)$ and $p \in B_w$ (see Figure 4.4(d)).*

Furthermore, it is possible to compute the tree T in total time $O(\alpha^d n \log n \log \beta)$, and the leaf cell containing a query point can be located in time $O(\log(\alpha n) + \log\log \beta)$.

4.4.2 Admissibility

In this section we explore how properties of the unit ball affect the effectiveness of convexification. Recall from Section 4.3 that convexification relies on the admissibility of the distance function, and we show here that this will be guaranteed if unit balls are fat, well centered, and smooth.

Given a convex body K and a parameter $0 < \gamma \leq 1$, we say that K is *centrally γ-fat* if there exist Euclidean balls B and B' centered at the origin, such that $B \subseteq K \subseteq B'$, and $\text{radius}(B)/\text{radius}(B') \geq \gamma$. Given a parameter $0 < \sigma \leq 1$, we say that K is *σ-smooth* if for every point x on the boundary of K, there exists a closed Euclidean ball of diameter $\sigma \cdot \text{diam}(K)$ that lies within K and has x on its boundary. We say that a scaling distance function is a *(γ, σ)-distance* if its associated unit ball B is both centrally γ-fat and σ-smooth.

In order to employ convexification for scaling distances, it will be useful to show that smoothness and fatness imply that the associated distance functions are admissible. This is encapsulated in the following lemma. It follows from a straightforward but rather technical exercise in multivariate differential calculus.

Lemma 16. *Given positive reals γ and σ, let f_p be a (γ, σ)-distance over $\mathbb{R}^d$ scaled about some point $p \in \mathbb{R}^d$. There exists τ (a function of γ and σ) such that f_p is τ-admissible.*

Proof. For any point $x \in \mathbb{R}^d$, we will show that (i) $\|\nabla f_p(x)\| \cdot \|x - p\| \leq f_p(x)/\gamma$ and (ii) $\|\nabla^2 f_p(x)\| \cdot \|x - p\|^2 \leq 2f_p(x)/(\sigma\gamma^3)$. It will follow that f_p is τ-admissible for $\tau = \sqrt{2/(\sigma\gamma^3)}$.

Let K denote the unit metric ball associated with f_p and let K' denote the scaled copy of K that just touches the point x. Let r be the unit vector in the direction px (we refer to this as the *radial direction*), and let n be the outward unit normal vector to the boundary of K' at x. (Throughout the proof, unit length vectors are defined in the Euclidean sense.) As K' is centrally γ-fat, it is easy to see that the cosine of the angle between r and n, that is, $r \cdot n$, is at least γ. As the boundary of K' is the level surface of f_p, it follows that $\nabla f_p(x)$ is directed along n. To compute the norm of the gradient, note that

$$\nabla f_p(x) \cdot r \;=\; \lim_{\delta \to 0} \frac{f_p(x + \delta r) - f_p(x)}{\delta}.$$

As f_p is a scaling distance function, it follows that

$$f_p(x + \delta r) - f_p(x) \;=\; \frac{\delta}{\|x - p\|} f_p(x).$$

Thus

$$\nabla f_p(x) \cdot r \;=\; \frac{f_p(x)}{\|x - p\|}.$$

Recalling that $r \cdot n \geq \gamma$, we obtain

$$\|\nabla f_p(x)\| \;\leq\; \frac{f_p(x)}{\gamma \|x - p\|}.$$

Thus $\|\nabla f_p(x)\| \cdot \|x - p\| \leq f_p(x)/\gamma$, as desired.

We next bound the norm of the Hessian $\nabla^2 f_p(x)$. As the Hessian matrix is positive semidefinite, recall that it has a full set of independent eigenvectors that are mutually orthogonal, and its norm equals its largest eigenvalue. Because f_p is a scaling distance function, it changes linearly along the radial direction. Therefore, one of the eigenvectors of $\nabla^2 f_p(x)$ is in direction r, and the associated eigenvalue is 0 (see Figure 4.5). It follows that the remaining eigenvectors all lie in a subspace that is orthogonal to r. In particular, the eigenvector associated with its largest eigenvalue must lie in this subspace. Let u denote such an eigenvector of unit length, and let λ denote the associated eigenvalue.

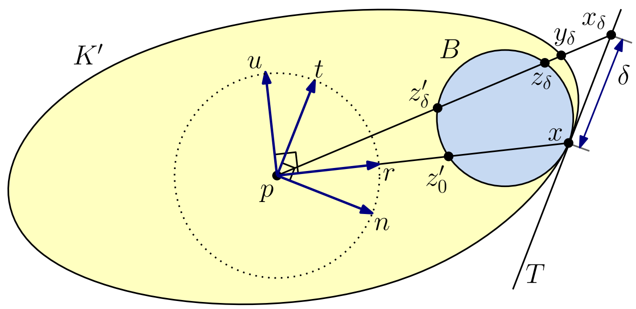


Figure 4.5: Proof of Lemma 16.

Note that λ is the second directional derivative of f_p in the direction u. In order to bound λ, we find it convenient to first bound the second directional derivative of f_p in a slightly different direction. Let T denote the hyperplane tangent to K' at point x. We project u onto T and let t denote the resulting vector scaled to have unit length. We will compute the second directional derivative of f_p in the direction t. Let λ_t denote this quantity. In order to relate λ_t with λ, we write t as $(t \cdot r)r + (t \cdot u)u$. Since r and u are mutually orthogonal eigenvectors of $\nabla^2 f_p(x)$, by elementary linear algebra, it follows that $\lambda_t = (t \cdot r)^2 \lambda_r + (t \cdot u)^2 \lambda_u$, where λ_r and λ_u are the eigenvalues associated with r and u, respectively. Since $\lambda_r = 0$, $\lambda_u = \lambda$, and $t \cdot u = r \cdot n \geq \gamma$, we have $\lambda_t \geq \gamma^2 \lambda$, or equivalently, $\lambda \leq \lambda_t / \gamma^2$. In the remainder of the proof, we will bound λ_t, which will yield the desired bound on λ.

Let $x_\delta = x + \delta t$ and $\psi(\delta) = f_p(x_\delta)$. Clearly $\lambda_t = \psi''(0)$. Using the Taylor series and the fact that $\psi'(0) = \nabla f_p(x) \cdot t = 0$, it is easy to see that

$$\psi''(0) = 2 \cdot \lim_{\delta \to 0} \frac{\psi(\delta) - \psi(0)}{\delta^2}.$$

Letting y_δ denote the intersection point of the segment $\overline{px_\delta}$ with the boundary of K', and observing that both x and y_δ lie on $\partial K'$ (implying that $f_p(x) = f_p(y_\delta)$), we have

$$\psi(\delta) \;=\; f_p(x_\delta) \;=\; \frac{\|x_\delta - p\|}{\|y_\delta - p\|} f_p(x),$$

and thus

$$\psi(\delta) - \psi(0) \;=\; \frac{\|x_\delta - p\| - \|y_\delta - p\|}{\|y_\delta - p\|} f_p(x) \;=\; \frac{\|x_\delta - y_\delta\|}{\|y_\delta - p\|} f_p(x).$$

It follows that

$$\psi''(0) \;=\; 2 \cdot \lim_{\delta \to 0} \frac{1}{\delta^2} \frac{\|x_\delta - y_\delta\|}{\|y_\delta - p\|} f_p(x) \;=\; \frac{2 f_p(x)}{\|x - p\|} \cdot \lim_{\delta \to 0} \frac{\|x_\delta - y_\delta\|}{\delta^2}.$$

We next compute this limit. Let $B \subset K'$ denote the maximal ball tangent to K' at x and let R denote its radius. As K' is σ-smooth, we have that

$$R \;\geq\; \frac{\sigma}{2} \cdot \mathrm{diam}(K') \;\geq\; \frac{\sigma}{2} \cdot \|x - p\|.$$

Consider the line passing through p and x_δ. For sufficiently small δ, it is clear that this line must intersect the boundary of the ball B at two points. Let z_δ denote the intersection point closer to x_δ and z'_δ denote the other intersection point. Clearly, $\|x_\delta - y_\delta\| \leq \|x_\delta - z_\delta\|$ and, by the power of the point theorem, we have

$$\delta^2 \;=\; \|x_\delta - x\|^2 \;=\; \|x_\delta - z_\delta\| \cdot \|x_\delta - z'_\delta\|.$$

It follows that

$$\frac{\|x_\delta - y_\delta\|}{\delta^2} \;\leq\; \frac{\|x_\delta - z_\delta\|}{\delta^2} \;=\; \frac{1}{\|x_\delta - z'_\delta\|}.$$

Thus

$$\lim_{\delta \to 0} \frac{\|x_\delta - y_\delta\|}{\delta^2} \;\leq\; \lim_{\delta \to 0} \frac{1}{\|x_\delta - z'_\delta\|} \;=\; \frac{1}{\|x - z'_0\|},$$

where z'_0 denotes the point of intersection of the line passing through p and x with the boundary of B. Since the cosine of the angle between this line and the diameter of ball B at x equals $r \cdot n$, which is at least γ, we have $\|x - z'_0\| \geq 2\gamma R$. It follows that

$$\lim_{\delta \to 0} \frac{\|x_\delta - y_\delta\|}{\delta^2} \;\leq\; \frac{1}{2\gamma R} \;\leq\; \frac{1}{\sigma\gamma\|x - p\|}.$$

Substituting this bound into the expression found above for λ_t, we obtain

$$\lambda_t \;=\; \psi''(0) \;\leq\; \frac{2 f_p(x)}{\sigma\gamma\|x - p\|^2}.$$

Recalling that $\lambda \leq \lambda_t/\gamma^2$, we have

$$\lambda \;\leq\; \frac{2 f_p(x)}{\sigma\gamma^3\|x - p\|^2},$$

which implies that $\|\nabla^2 f_p(x)\| \cdot \|x - p\|^2 \leq 2 f_p(x)/(\sigma\gamma^3)$. This completes the proof. $\square$

Our results on ε-ANN queries for scaling distances will be proved for any set of sites whose associated distance functions (which may be individual to each site) are all (γ, σ)-distances for fixed γ and σ. Our results on the Minkowski and Mahalanobis distances thus arise as direct consequences of the following easy observations.

Lemma 17.

(*i*) *For any positive real $k > 1$, the Minkowski distance ℓ_k is a (γ, σ)-distance, where γ and σ are functions of k and d.*

This applies to multiplicatively weighted Minkowski distances as well.

(*ii*) *The Mahalanobis distance defined by a matrix M_p is a (γ, σ)-distance, where γ and σ are functions of M_p's minimum and maximum eigenvalues.*

4.4.3 The Data Structure

Let us return to the discussion of how to answer ε-ANN queries for a family of (γ, σ)-distance functions. By Lemma 16, such functions are τ-admissible, where τ depends only on γ and σ.

We begin by building an (α, β)-AVD over $\mathbb{R}^d$ by invoking Lemma 15 for $\alpha = 2\tau$ and $\beta = 10\tau/\varepsilon$. (These choices will be justified below.) For each leaf cell w, the nearest neighbor of any query point $q \in w$ can arise from one of the three cases in the lemma. Case (i) is trivial since there is just one point.

Case (ii) (the *outer cluster*) can be solved easily by reduction to Lemma 13. Recall that we have a BBD-tree leaf cell w, and the objective is to compute an ε-ANN from among the points of the outer cluster, that is, a set whose sites are at Euclidean distance at least $\alpha \cdot \mathrm{diam}(w)$ from w. Let B denote the smallest Euclidean ball enclosing w, and let $\mathcal{F}$ be the family of distance functions associated with the sites of the outer cluster. Since $\alpha = 2\tau$, B is (2τ)-separated from the points of the outer cluster. By Lemma 13, we can answer ε-AVR queries with respect to $\mathcal{F}_{\min}$, and this is equivalent to answering ε-ANN queries with respect to the outer cluster. The query time is $O(\log \frac{1}{\varepsilon})$ and the storage is $O((\frac{1}{\varepsilon})^{d/2})$.

All that remains is case (iii), the *inner cluster*. Recall that these sites lie within a ball B_w such that $\mathbf{d}(B_w, w) \geq \beta \cdot \mathrm{diam}(B_w)$. In approximate Euclidean nearest-neighbor searching, a separation as large as β would allow us to replace all the points of B_w with a single representative site, but this is not applicable when different sites are associated with different scaling distance functions. We will show instead that queries can be answered by partitioning the query space into a small number of regions such that Lemma 13 can be applied to each region. Let $\{p_1, \ldots, p_m\}$ denote the sites lying within B_w, and let $\mathcal{F} = \{f_1, \ldots, f_m\}$ denote the associated family of (γ, σ)-distance functions.

Let p' be the center of B_w, and for $1 \leq i \leq m$, define the *perturbed distance function* $f_i'(x) = f_i(x + p_i - p')$ to be the function that results by moving p_i to p' without altering the unit metric ball. Let $\mathcal{F}'$ denote the associated family of distance functions. Our next lemma shows that this perturbation does not significantly alter the relative function values.

Lemma 18. *Let $p \in \mathbb{R}^d$ be the site of a τ-admissible distance function f. Let B be a ball containing p and let x be a point that is β-separated from B for $\beta \geq 2\tau$. Letting p' denote B's center, define $f'(x) = f(x + p - p')$. Then*

$$\frac{|f'(x) - f(x)|}{f(x)} \;\leq\; \frac{2\tau}{\beta}.$$

Proof. Define B_x to be the translate of B whose center coincides with x. Since p and p' both lie within B, x and $x + p - p'$ both lie within B_x. Let $\kappa = \beta/\tau$. Since x and B are β-separated, p' and B_x are also β-separated. Equivalently, they are $(\tau\kappa)$-separated. Because $\kappa \geq 2$, $\kappa/(\kappa - 1) \leq (1 + 2/\kappa)$. Because f' has the same unit metric ball as f, it is also τ-admissible, and so by Lemma 12

$$f'^{+}(B_x) \;\leq\; \frac{\kappa}{\kappa - 1} f'^{-}(B_x) \;\leq\; \left(1 + \frac{2}{\kappa}\right) f'^{-}(B_x) \;=\; \left(1 + \frac{2\tau}{\beta}\right) f'^{-}(B_x).$$

Letting $x' = x - (p - p')$, we have $f(x) = f'(x')$. Clearly $x' \in B_x$. Let us assume that $f'(x) \geq f(x)$. (The other case is similar.) We have

$$\begin{aligned} f'(x) - f(x) &= f'(x) - f'(x') \leq f'^{+}(B_x) - f'^{-}(B_x) \\ &\leq \frac{2\tau}{\beta} f'^{-}(B_x) \leq \frac{2\tau}{\beta} f'(x') = \frac{2\tau}{\beta} f(x), \end{aligned}$$

which implies the desired inequality. □

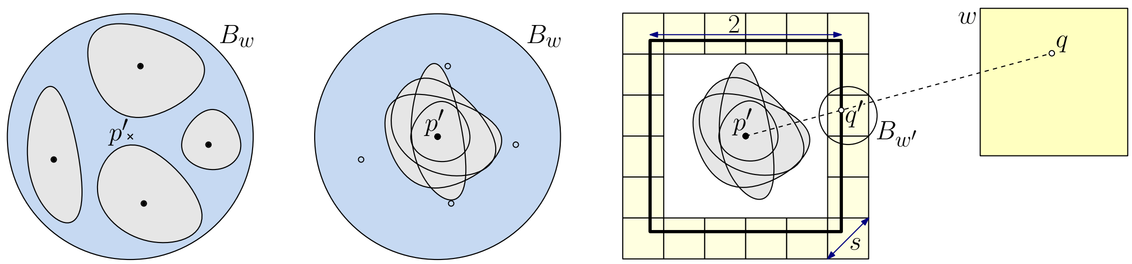
Since every point $x \in w$ is β-separated from B_w, by applying this perturbation to every function in $\mathcal{F}$, we alter relative errors by at most $2\tau/\beta$. By selecting β so that $(1 + 2\tau/\beta)^2 \leq 1 + \varepsilon/2$, we assert that the total error is at most $\varepsilon/2$. To see this, consider any query point x, and let f_i be the function that achieves the minimum value for $\mathcal{F}_{\min}(x)$, and let f'_j be the perturbed function that achieves the minimum value for $\mathcal{F}'_{\min}(x)$. Then

$$\begin{aligned} f_j(x) &\leq \left(1 + \frac{2\tau}{\beta}\right) f'_j(x) \leq \left(1 + \frac{2\tau}{\beta}\right) f'_i(x) \\ &\leq \left(1 + \frac{2\tau}{\beta}\right)^2 f_i(x) \leq \left(1 + \frac{\varepsilon}{2}\right) f_i(x). \end{aligned}$$

It is easy to verify that for all sufficiently small ε, our choice of $\beta = 10\tau/\varepsilon$ satisfies this condition (and it is also at least 2τ as required by the lemma).

We can now explain how to answer ε-ANN queries for the inner cluster. Consider the sites of the inner cluster, which all lie within B_w (see Figure 4.6(a)). We apply Lemma 18 to produce the perturbed family $\mathcal{F}'$ of τ-admissible functions (see Figure 4.6(b)).

Since these are all scaling distance functions, the nearest neighbor of any query point $q \in \mathbb{R}^d$ (irrespective of whether it lies within w) is the same for every point



(a) (b) (c)

Figure 4.6: (a) Inner-cluster sites with their respective distance functions, (b) their perturbation to a common site p', and (c) the reduction to Lemma 13.

on the ray from p' through q. Therefore, it suffices to evaluate the answer to the query for any single query point q' on this ray. In particular, let us fix a hypercube of side length 2 centered at p' (see Figure 4.6(c)). We will show how to answer $(\varepsilon/3)$-AVR queries for points on the boundary of this hypercube with respect to $\mathcal{F}'$. A general query will then be answered by computing the point where the ray from p' to the query point intersects the hypercube's boundary and returning the result of this query. The total error with respect to the original functions will be at most $(1+\varepsilon/2)(1+\varepsilon/3)$, and for all sufficiently small ε, this is at most $1+\varepsilon$, as desired.

All that remains is to show how to answer $(\varepsilon/3)$-AVR queries for points on the boundary of the hypercube. Let $s = 1/(2\tau+1)$, and let W be a set of hypercubes of diameter s that cover the boundary of the hypercube of side length 2 centered at p' (see Figure 4.6(c)). The number of such boxes is $O(\tau^{d-1})$. For each $w' \in W$, let $B_{w'}$ be the smallest ball enclosing w'. Each point on the hypercube is at distance at least 1 from p'. For each $w' \in W$, we have $\mathbf{d}(p', B_{w'}) \geq 1 - s = 2\tau \cdot \mathrm{diam}(B_{w'})$, implying that p' and $B_{w'}$ are (2τ)-separated. Therefore, by Lemma 13 there is a data structure that can answer $(\varepsilon/3)$-AVR queries with respect to the perturbed distance functions $\mathcal{F}'_{\min}$ in time $O(\log \frac{1}{\varepsilon})$ with storage $O((\frac{1}{\varepsilon})^{d/2})$.

In summary, a query is answered by computing the ray from p' through q, and determining the unique point q' on the boundary of the hypercube that is hit by this ray. We then determine the hypercube w' containing q' in constant time and invoke the associated data structure for answering $(\varepsilon/3)$-AVR queries with respect to $\mathcal{F}'$. The total storage needed for all these structures is $O(\tau^{d-1}/\varepsilon^{d/2})$. For any query point, we can determine which of these data structures to access in $O(1)$ time. Relative to the case of the outer cluster, we suffer only an additional factor of $O(\tau^{d-1})$ to store these data structures.

Under our assumption that γ and σ are constants, it follows that both τ and α are constants and β is $O(1/\varepsilon)$. By Lemma 15, the total number of leaf nodes in the (α, β)-AVD is $O(n \log \frac{1}{\varepsilon})$. Combining this with the $O(1/\varepsilon^{d/2})$ space for the data structure to answer queries with respect to the outer cluster and $O(\tau^{d-1}/\varepsilon^{d/2})$ overall space for the inner cluster, we obtain a total space of $O((n \log \frac{1}{\varepsilon})/\varepsilon^{d/2})$. The query time is simply the combination of the $O(\log(\alpha n) + \log\log \beta) = O(\log n + \log\log \frac{1}{\varepsilon})$ time to locate the leaf cell (by Lemma 15), and the $O(\log \frac{1}{\varepsilon})$ time to answer $O(\varepsilon)$-AVR queries. The total query time is therefore $O(\log \frac{n}{\varepsilon})$, as desired. This establishes Theorem 4.

4.5 Search Queries with Bregman Divergences

In this section we demonstrate how to answer ε-ANN queries for a set of n sites over a Bregman divergence. We assume that the Bregman divergence is defined by a strictly convex, twice-differentiable function F over an open convex domain $\mathbb{X} \subseteq \mathbb{R}^d$. As mentioned in the introduction, given a site p, we interpret the divergence $D_F(x, p)$ as a distance function of x about p, that is, analogous to $f_p(x)$ for scaling distances. Thus, gradients and Hessians are defined with respect to the variable x. Our results will be based on the assumption that the divergence is τ-admissible for a constant τ. This will be defined formally in the following section.

4.5.1 Measures of Bregman Complexity

In Section 4.1 we introduced the concepts of similarity and asymmetry for Bregman divergences. We can extend the notion of admissibility to Bregman divergences by defining a Bregman divergence D_F to be τ-*admissible* if the associated distance function $f_p(\cdot) = D_F(\cdot, p)$ is τ-admissible.

It is natural to ask how the various criteria of Bregman complexity (asymmetry, similarity, and admissibility) relate to each other. For the sake of relating admissibility with asymmetry, it will be helpful to introduce a directionally-sensitive variant of admissibility. Given f_p and τ as above, we say that f_p is *directionally* τ-*admissible* if for all $x \in \mathbb{X}$, $\nabla f_p(x) \cdot (x - p) \leq \tau f_p(x)$. (Note that only the gradient condition is used in this definition.)

To facilitate the analysis below, we start with establishing a number of basic properties of Bregman divergences. Throughout, we assume that a Bregman divergence is defined by a strictly convex, twice-differentiable function F over an open convex domain $\mathbb{X} \subseteq \mathbb{R}^d$. Given a site p, we interpret the divergence $D_F(x, p)$ as a distance function of x about p, and so gradients and Hessians are defined with respect to the variable x. The following lemma provides a few useful observations regarding the Bregman divergence. We omit the proof since these all follow directly from the definition of Bregman divergence. Observation (i) is related to the *symmetrized Bregman divergence* [112]. Observation (ii), known as the *three-point property* [104], generalizes the law of cosines when the Bregman divergence is the Euclidean squared distance.

Lemma 19. *Given any Bregman divergence D_F defined over an open convex domain $\mathbb{X}$, and points $q, p, p' \in \mathbb{X}$:*

(i) $D_F(q, p) + D_F(p, q) = (\nabla F(q) - \nabla F(p)) \cdot (q - p)$

(ii) $D_F(q, p') + D_F(p', p) = D_F(q, p) + (q - p') \cdot (\nabla F(p) - \nabla F(p'))$

(iii) $\nabla D_F(q, p) = \nabla F(q) - \nabla F(p)$

(iv) $\nabla^2 D_F(q, p) = \nabla^2 F(q)$.

In parts (iii) and (iv), derivatives involving $D_F(q, p)$ are taken with respect to q.

The above result allows us to establish the following upper and lower bounds on the value, gradient, and Hessian of a Bregman divergence based on the maximum and minimum eigenvalues of the function's Hessian.

Lemma 20. *Let F be a strictly convex function defined over some domain $\mathbb{X} \subseteq \mathbb{R}^d$, and let D_F denote the associated Bregman divergence. For each $x \in \mathbb{X}$, let $\lambda_{\min}(x)$ and $\lambda_{\max}(x)$ denote the minimum and maximum eigenvalues of $\nabla^2 F(x)$, respectively. Then, for all $p, q \in \mathbb{X}$, there exist points r_1, r_2, and r_3 on the open line segment $\overline{pq}$ such that*

$$\begin{array}{rcccl} \frac{1}{2}\lambda_{\min}(r_1)\|q-p\|^2 & \leq & D_F(q,p) & \leq & \frac{1}{2}\lambda_{\max}(r_1)\|q-p\|^2 \\ \lambda_{\min}(r_2)\|q-p\| & \leq & \|\nabla D_F(q,p)\| & \leq & \lambda_{\max}(r_3)\|q-p\| \\ \lambda_{\min}(q) & \leq & \|\nabla^2 D_F(q,p)\| & \leq & \lambda_{\max}(q). \end{array}$$

Proof. To establish the first inequality, we apply Taylor's theorem with the Lagrange form of the remainder to obtain

$$F(q) \;=\; F(p) + \nabla F(p) \cdot (q-p) + \frac{1}{2}(q-p)^{\intercal}\nabla^2 F(r_1)(q-p),$$

for some r_1 on the open line segment $\overline{pq}$. By substituting the above expression for $F(q)$ into the definition of $D_F(q,p)$ we obtain

$$D_F(q,p) = \; F(q) - F(p) - \nabla F(p) \cdot (q-p) = \; \frac{1}{2}(q-p)^{\intercal}\nabla^2 F(r_1)(q-p).$$

By basic linear algebra, we have

$$\lambda_{\min}(r_1)\|q-p\|^2 \;\leq\; (q-p)^{\intercal}\nabla^2 F(r_1)(q-p) \;\leq\; \lambda_{\max}(r_1)\|q-p\|^2.$$

Therefore,

$$\frac{\lambda_{\min}(r_1)}{2}\|q-p\|^2 \;\leq\; D_F(q,p) \;\leq\; \frac{\lambda_{\max}(r_1)}{2}\|q-p\|^2,$$

which establishes the first assertion.

For the second assertion, we recall from Lemma 19(iii) that $\nabla D_F(q,p) = \nabla F(q) - \nabla F(p)$. Let v be any unit vector. By applying the mean value theorem to the function $\psi(t) = v^{\intercal}\nabla F(p + t(q - p))$ for $0 \leq t \leq 1$, there exists a point $r_2 \in \overline{pq}$ (which depends on v) such that $v^{\intercal}(\nabla F(q) - \nabla F(p)) = v^{\intercal}\nabla^2 F(r_2)(q - p)$. Taking v to be the unit vector in the direction of $q - p$, and applying the Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}\|D_F(q,p)\| \;&=\; \|\nabla F(q) - \nabla F(p)\| \;\geq\; |v^{\intercal}(\nabla F(q) - \nabla F(p))| \\ &=\; |v^{\intercal}\nabla^2 F(r_2)(q - p)| \;\geq\; \lambda_{\min}(r_2)\|q - p\|.\end{aligned}$$

For the upper bound, we apply the same approach, but take v to be the unit vector in the direction of $\nabla F(q) - \nabla F(p)$. There exists $r_3 \in \overline{pq}$ such that

$$\begin{aligned}\|D_F(q,p)\| \;&=\; \|\nabla F(q) - \nabla F(p)\| \;=\; |v^{\intercal}(\nabla F(q) - \nabla F(p))| \;=\; |v^{\intercal}\nabla^2 F(r_3)(q - p)| \\ &\leq\; \|\nabla^2 F(r_3)(q - p)\| \;\leq\; \lambda_{\max}(r_3)\|q - p\|.\end{aligned}$$

This establishes the second assertion.

The final assertion follows from the fact that $\nabla^2 D_F(q,p) = \nabla^2 F(q)$ (Lemma 19(iv)) and the definition of the spectral norm. $\square$

With the help of this lemma, we can now relate the various measures of complexity for Bregman divergences.

Lemma 21. *Given an open convex domain $\mathbb{X} \subseteq \mathbb{R}^d$:*

(i) Any μ-similar Bregman divergence over $\mathbb{X}$ is 2μ-admissible.

(ii) Any μ-admissible Bregman divergence over $\mathbb{X}$ is directionally μ-admissible.

(iii) A Bregman divergence over $\mathbb{X}$ is μ-asymmetric if and only if it is directionally $(1 + \mu)$-admissible.

Proof. For each $x \in \mathbb{X}$, let $\lambda_{\min}(x)$ and $\lambda_{\max}(x)$ denote the minimum and maximum eigenvalues of $\nabla^2 F(x)$, respectively. We first show that for all $x \in \mathbb{X}$, $2 \leq \lambda_{\min}(x)$ and $\lambda_{\max}(x) \leq 2\mu$. We will prove only the second inequality, since the first follows by a symmetrical argument. Suppose to the contrary that there was a point $x \in \mathbb{X}$ such that $\lambda_{\max}(x) > 2\mu$. By continuity and the fact that $\mathbb{X}$ is convex and open, there exists a point $q \in \mathbb{X}$ distinct from x such that for any r on the open line segment $\overline{qx}$,

$$(q-x)^\intercal \nabla^2 F(r)(q-x) \;>\; 2\mu\|q-x\|^2. \tag{4.1}$$

Specifically, we may take q to lie sufficiently close to x along $x+v$, where v is the eigenvector associated with $\lambda_{\max}(x)$. As in the proof of Lemma 20, we apply Taylor's theorem with the Lagrange form of the remainder to obtain

$$\begin{aligned} D_F(q,x) &= F(q) - F(x) - \nabla F(x) \cdot (q-x) \\ &= \frac{1}{2}(q-x)^\intercal \nabla^2 F(r)(q-x) \;=\; \left(\frac{1}{t}\right)^2 \frac{1}{2}(r-x)^\intercal \nabla^2 F(r)(r-x). \end{aligned}$$

By Eq. (4.1), we have $D_F(q,x) > \mu\|q-x\|^2$. Therefore, D_F is not μ-similar. This yields the desired contradiction.

Because $2 \leq \lambda_{\min}(x) \leq \lambda_{\max}(x) \leq 2\mu$ for all $x \in \mathbb{X}$, by Lemma 20, we have

$$\|q-p\|^2 \;\leq\; D_F(q,p), \qquad \|\nabla D_F(q,p)\| \;\leq\; 2\mu\|q-p\|, \quad \text{and} \quad \|\nabla^2 D_F(q,p)\| \;\leq\; 2\mu,$$

which imply

$$\|\nabla D_F(q,p)\|\,\|q-p\| \;\leq\; 2\mu D_F(q,p) \qquad \text{and} \qquad \|\nabla^2 D_F(q,p)\|\,\|q-p\|^2 \;\leq\; 2\mu D_F(q,p),$$

which together imply that D is 2μ-admissible, as desired.

To prove (ii), observe that by the Cauchy-Schwarz inequality $\nabla D_F(q,p) \cdot (q - p) \leq \|\nabla D_F(q,p)\| \cdot \|q-p\|$, and therefore, any divergence that satisfies the condition for μ-admissibility immediately satisfies the condition for directional μ-admissibility.

To show (iii), consider any points $p, q \in \mathbb{X}$. Recall the facts regarding the Bregman divergence presented in Lemma 19. By combining observations (i) and (iii) from that lemma, we have $D_F(q,p) + D_F(p,q) = \nabla D_F(q,p) \cdot (q-p)$. Observe that if D is directionally $(1+\mu)$-admissible, then

$$D_F(q,p) + D_F(p,q) \;=\; \nabla D_F(q,p) \cdot (q-p) \;\leq\; (1+\mu) D_F(q,p),$$

which implies that $D_F(p,q) \leq \mu(D_F(q,p)$, and hence D is μ-asymmetric. Conversely, if D is μ-asymmetric, then

$$\nabla D_F(q,p) \cdot (q-p) \;=\; D_F(q,p) + D_F(p,q) \;\leq\; D_F(q,p) + \mu D_F(q,p) \;=\; (1+\mu) D_F(q,p),$$

implying that D_F is directionally $(1+\mu)$-admissible. (Recall that directional admissibility requires only that the gradient condition be satisfied.) $\square$

Remark 2. *Claim (i) is strict since the Bregman divergence D_F defined by $F(x) = x^4$ over $\mathbb{X} = \mathbb{R}$ is not μ-similar for any μ, but it is 4-admissible. We do not know whether claim (ii) is strict, but we conjecture that it is.*

4.5.2 The Data Structure

Let us return to the discussion of how to answer ε-ANN queries for a τ-admissible Bregman divergence over a domain $\mathbb{X}$. Because any distance function that is τ-admissible is τ'-admissible for any $\tau' \geq \tau$, we may assume that $\tau \geq 1$.[6] We begin by building an (α, β)-AVD over $\mathbb{R}^d$ by invoking Lemma 15 for $\alpha = 2\tau$ and $\beta = 4\tau^2/\varepsilon$. (These choices will be justified below.) For each leaf cell w, the nearest neighbor of any query point $q \in w$ can arise from one of the three cases in the lemma. Cases (i) and (ii) are handled in exactly the same manner as in Section 4.4.3. (Case (i) is trivial, and case (ii) applies for any τ-admissible family of functions.)

It remains to handle case (iii), the *inner cluster*. Recall that these sites lie within a ball B_w such that $\mathbf{d}(B_w, w) \geq \beta \cdot \mathrm{diam}(B_w)$. We show that as a result of choosing β sufficiently large, for any query point in w the distance from all the sites within B_w are sufficiently close that we may select any of these sites as the approximate nearest neighbor. This is a direct consequence of the following lemma.

Lemma 22. *Let D be a τ-admissible Bregman divergence and let $0 < \varepsilon \leq 1$. Consider any leaf cell w of the (α, β)-AVD, where $\beta \geq 4\tau^2/\varepsilon$. Then, for any $q \in w$ and points $p, p' \in B_w$*

$$\frac{|D(q,p) - D(q,p')|}{D(q,p)} \;\leq\; \varepsilon.$$

Proof. Without loss of generality, we may assume that $D(q,p) \geq D(q,p')$. By adding $D(p,p')$ to the left side of Lemma 19(ii) and rearranging terms, we have

$$\begin{aligned} D(q,p) - D(q,p') \leq\; & (D(q,p) - D(q,p')) + D(p,p') \\ =\; & (D(p',p) + (\nabla F(p') - \nabla F(p)) \cdot (q - p')) + D(p,p') \\ =\; & (\nabla F(p') - \nabla F(p)) \cdot (q - p')) + (D(p',p) + D(p,p')). \end{aligned}$$

[6]Indeed, it can be shown that any distance function that is convex, as Bregman divergences are, cannot be τ-admissible for $\tau < 1$.

By Lemma 19(i) we have

$$\begin{aligned} D(q,p) - D(q,p') &\leq (\nabla F(p') - \nabla F(p)) \cdot (q - p') + (\nabla F(p') - \nabla F(p)) \cdot (p' - p) \\ &= (\nabla F(p') - \nabla F(p)) \cdot (q - p). \end{aligned}$$

Let v be any unit vector. Applying the mean value theorem to the function $\psi(t) = v^\intercal \nabla F(p + t(p' - p))$ for $0 \leq t \leq 1$, implies that there exists a point $r \in \overline{pp'}$ (which depends on v) such that $v^\intercal(\nabla F(p') - \nabla F(p)) = v^\intercal \nabla^2 F(r)(p' - p)$. Taking v to be the unit vector in the direction of $q - p$, and applying the Cauchy-Schwarz inequality, we obtain

$$D(q,p) - D(q,p') \leq (\nabla^2 F(r)(p' - p)) \cdot (q - p) \leq \|\nabla^2 F(r)\| \, \|p' - p\| \, \|q - p\|.$$

By Lemma 19(iv) and τ-admissibility, $\|\nabla^2 F(r)\| = \|\nabla^2 D(r,q)\| \leq \tau D(r,q)/\|r - q\|^2$, which implies

$$D(q,p) - D(q,p') \leq \frac{\tau D(r,q)}{\|r - q\|^2} \|p' - p\| \, \|q - p\|. \tag{4.2}$$

Since r lies on the segment between p' and p, it follows that $r \in B_w$. Letting $\delta = \mathrm{diam}(B_w)$, we have $\max(\|p' - p\|, \|r - p\|) \leq \delta$ and $\|r - q\| \geq \beta\delta$. By the triangle inequality, $\|q - p\| \leq \|q - r\| + \|r - p\|$. Therefore,

$$\frac{\|q - p\|}{\|r - q\|} \leq \frac{\|q - r\| + \|r - p\|}{\|r - q\|} = 1 + \frac{\|r - p\|}{\|r - q\|} \leq 1 + \frac{1}{\beta},$$

and since clearly $\beta \geq 1$,

$$\frac{\|p' - p\| \, \|q - p\|}{\|r - q\|^2} \leq \frac{1}{\beta}\left(1 + \frac{1}{\beta}\right) \leq \frac{2}{\beta}. \tag{4.3}$$

We would like to express the right-hand side of Eq. (4.2) in terms of p rather than r. By the τ-admissibility of D and the fact that $r, p \in B_w$, we can apply

Lemma 12(i) (with the distance function $f_q(\cdot) = D(\cdot, q)$ and $\kappa = \beta/\tau$) to obtain $D(r, q) \le D(p, q)/(1 - \tau/\beta)$. Combining Eq. (4.3) with this, we obtain

$$D(q, p) - D(q, p') \;\le\; \frac{2\tau}{\beta} D(r, q) \;\le\; \frac{2\tau}{\beta(1 - \tau/\beta)} D(p, q).$$

In Lemma 21(iii) we showed that any $(1 + \mu)$-admissible Bregman divergence is μ-asymmetric, and by setting $\mu = \tau - 1$ it follows that $D(p, q) \le (\tau - 1)D(q, p)$. Putting this all together, we obtain

$$D(q, p) - D(q, p') \;\le\; \frac{2\tau(\tau - 1)}{\beta(1 - \tau/\beta)} D(q, p).$$

All that remains is to set β sufficiently large to obtain the desired result. Since $\tau \ge 1$ and $\varepsilon \le 1$, it is easily verified that setting $\beta = 4\tau^2/\varepsilon$ suffices to produce the desired conclusion. □

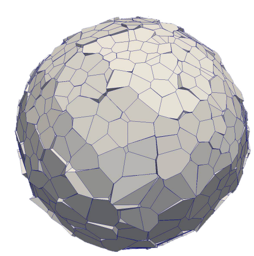
Under our assumption that τ is a constant, α is a constant and β is $O(1/\varepsilon)$. The analysis proceeds much like the case for scaling distances. By Lemma 15, the total number of leaf nodes in the (α, β)-AVD is $O(n \log \frac{1}{\varepsilon})$. We require only one representative for cases (i) and (iii), and as in Section 4.4.3, we need space $O(1/\varepsilon^{d/2})$ to handle case (ii). The query time is simply the combination of the $O(\log(\alpha n) + \log\log \beta) = O(\log n + \log\log \frac{1}{\varepsilon})$ time to locate the leaf cell (by Lemma 15), and the $O(\log \frac{1}{\varepsilon})$ time to answer $O(\varepsilon)$-AVR queries for case (ii). The total query time is therefore $O(\log \frac{n}{\varepsilon})$, as desired. This establishes Theorem 5.

Chapter 5: Sampling Conditions for Voronoi Meshing

Mesh generation is a fundamental problem in computational geometry, geometric modeling, computer graphics, scientific computing and engineering simulations. There has been a growing interest in polyhedral meshes as an alternative to tetrahedral or hex-dominant meshes [121].

In this chapter, we initiate our study the Voronoi meshing problem that asks to decompose a volume bounded by a piecewise-smooth surface into a collection of Voronoi cells. We start by assuming the surface is a smooth manifold with a known local feature size, and derive sufficient conditions on the sampling to guarantee an isotopic surface reconstruction.

5.1 Introduction



An intuitive approach to surface approximation is to place pairs of Voronoi seeds *mirrored* across the surface such that their shared Voronoi facets approximate the surface. However, a naive implementation of this idea results in a rough surface with spurious misaligned facets; see the inset. Nonetheless, a more principled mirroring approach provided the first provably-correct surface reconstruction algorithm [122]. Given an ϵ-sample from an unknown smooth surface, the PowerCrust algorithm [123] places weighted Voronoi seeds at a subset of the vertices in the Voronoi diagram of the input samples.

The proposed scheme, called VoroCrust, can be viewed as a principled mirroring technique, which shares a number of key features with the power crust algorithm [123]. The power crust literature [122–126] developed a rich theory for surface approximation, namely the ε-sampling paradigm. Recall that the power crust algorithm uses an ε-sample of unweighted points to place weighted sites, so-called *poles*, near the medial axis of the underlying surface. The surface reconstruction is the collection of facets separating power cells of poles on the inside and outside of the enclosed volume.

Regarding samples and poles as primal-dual constructs, power crust performs a *primal-dual-dual-primal dance*. VoroCrust makes a similar dance where weights are introduced differently; the samples are weighted to define unweighted sites tightly hugging the surface, with the reconstruction arising from their unweighted Voronoi diagram. The key advantage is the freedom to place more sites within the enclosed volume without disrupting the surface reconstruction. This added freedom is essential to the generation of graded meshes; a primary virtue of the proposed algorithm. Another virtue of the algorithm is that all samples appear as vertices in the resulting mesh. While the power crust algorithm does not guarantee that, some variations do so by means of filtering, at the price of the reconstruction no longer being the boundary of power cells [122, 127, 128].

The main construction underlying VoroCrust is a suitable union of balls centered on the bounding surface, as studied in the context of non-uniform approximations [129]. Unions of balls enjoy a wealth of results [130–132], which enable a variety of algorithms [123, 133, 134].

Similar constructions have been proposed for meshing problems in the applied sciences with heuristic extensions to 3D settings; see [135] and the references therein for a recent example. Aichholzer et al. [136] adopt closely related ideas to construct a union of surface balls using power crust poles for sizing estimation. However, their goal was to produce a coarse homeomorphic surface reconstruction. As in [136], the use of balls and α-shapes for surface reconstruction was explored earlier, e.g., ball-pivoting [137, 138], but the connection to Voronoi meshing has been absent. In contrast, VoroCrust aims at a decomposition of the enclosed volume into fat Voronoi cells conforming to an isotopic surface reconstruction with quality guarantees.

In this chapter, we present a theoretical analysis of an abstract version of the VoroCrust algorithm. This establishes the quality and approximation guarantees of its output for volumes bounded by smooth surfaces. A description of the algorithm we analyze is given next; see Figure 5.1 for an illustration in 2D.

The abstract VoroCrust algorithm

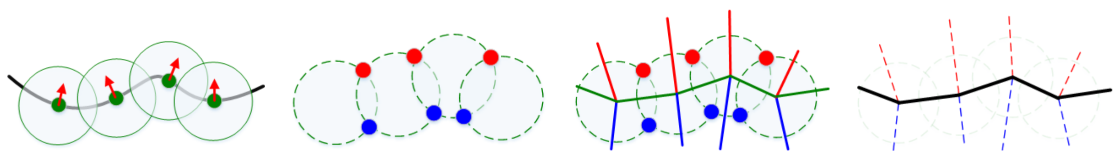
1. Take as input a sample $\mathcal{P}$ on the surface $\mathcal{M}$ bounding the volume $\mathcal{O}$.

2. Define a ball B_i centered at each sample p_i, with a suitable radius r_i, and let $\mathcal{U} = \cup_i B_i$.

3. Initialize the set of sites $\mathcal{S}$ with the corner points of $\partial\mathcal{U}$, $\mathcal{S}^\uparrow$ and $\mathcal{S}^\downarrow$, on both sides of $\mathcal{M}$.

4. Optionally, generate additional sites $\mathcal{S}^{\downarrow\downarrow}$ in the interior of $\mathcal{O}$, and include $\mathcal{S}^{\downarrow\downarrow}$ into $\mathcal{S}$.

5. Compute the Voronoi diagram $\mathrm{Vor}(\mathcal{S})$ and retain the cells with sites in $\mathcal{S}^{\downarrow} \cup \mathcal{S}^{\downarrow\downarrow}$ as the volume mesh $\hat{\mathcal{O}}$, where the facets between $\mathcal{S}^{\uparrow}$ and $\mathcal{S}^{\downarrow}$ yield a surface approximation $\hat{\mathcal{M}}$.



(a) Surface balls. (b) Labeled corners. (c) Voronoi cells. (d) Reconstruction.

Figure 5.1: VoroCrust reconstruction, demonstrated on a planar curve.

In this chapter, we assume $\mathcal{O}$ is a bounded open subset of $\mathbb{R}^3$, whose boundary $\mathcal{M}$ is a closed, bounded and smooth surface. We further assume that $\mathcal{P}$ is an ε-sample, with a weak σ-sparsity condition, and r_i is set to δ times the local feature size at p_i. For appropriate values of ε, σ and δ, we prove that $\hat{\mathcal{O}}$ and $\hat{\mathcal{M}}$ are isotopic to $\mathcal{O}$ and $\mathcal{M}$, respectively. We also show that simple techniques for sampling within $\mathcal{O}$, e.g., octree refinement, guarantee an upper bound on the fatness of all cells in $\hat{\mathcal{O}}$, as well as the number of samples.

The rest of this chapter is organized as follows. [Dave: Fix section numbers.] Section 2 introduces the key definitions and notation used throughout the paper. Section 3 describes the placement of Voronoi seeds and basic properties of our construction assuming the union of surface balls satisfies a structural property. Section 4 proves this property holds and establishes the desired approximation guarantees under certain conditions on the input sample. Section 5 considers the generation of interior samples and bounds the fatness of all cells in the output mesh. Section 6 concludes the paper with pointers for future work. A number of proofs are deferred to the appendices.

5.2 Preliminaries

Throughout this chapter, standard general position assumptions [139] are made implicitly to simplify the presentation. We use $\mathbf{d}(p, q)$ to denote the Euclidean distance between two points $p, q \in \mathbb{R}^3$, and $\mathbb{B}(c, r)$ to denote the Euclidean ball centered at $c \in \mathbb{R}^3$ with radius r. We proceed to introduce the notation and recall the key definitions used throughout, following those in [123, 129, 130].

5.2.1 Sampling and Approximation

We take as input a set of sample points $\mathcal{P} \subset \mathcal{M}$. A local scale or *sizing* is used to vary the sample density. Recall that the *medial axis* [123] of $\mathcal{M}$, denoted by $\mathcal{A}$, is the closure of the set of points in $\mathbb{R}^3$ with more than one closest point on $\mathcal{M}$. Hence, $\mathcal{A}$ has one component inside $\mathcal{O}$ and another outside. Each point of $\mathcal{A}$ is the center of a *medial ball* tangent to $\mathcal{M}$ at multiple points. Likewise, each point on $\mathcal{M}$ has two tangent medial balls, not necessarily of the same size. The *local feature size* at $x \in \mathcal{M}$ is defined as $\text{lfs}(x) = \inf_{a \in \mathcal{A}} \mathbf{d}(x, a)$. The set $\mathcal{P}$ is an ε*-sample* [140] if for all $x \in \mathcal{M}$ there exists $p \in \mathcal{P}$ such that $\mathbf{d}(x, p) \leq \varepsilon \cdot \text{lfs}(x)$.

We desire an approximation of $\mathcal{O}$ by a Voronoi mesh $\hat{\mathcal{O}}$, where the boundary $\hat{\mathcal{M}}$ of $\hat{\mathcal{O}}$ approximates $\mathcal{M}$. Recall that two topological spaces are *homotopy-equivalent* [129] if they have the same topology type. A stronger notion of topological equivalence is *homeomorphism*, which holds when there exists a continuous bijection with a continuous inverse from $\mathcal{M}$ to $\hat{\mathcal{M}}$. The notion of isotopy captures an even stronger type of equivalence for surfaces *embedded* in Euclidean space. Two surfaces $\mathcal{M}, \hat{\mathcal{M}} \subset \mathbb{R}^3$ are *isotopic* [141, 142] if there is a continuous mapping $F : \mathcal{M} \times [0, 1] \to \mathbb{R}^3$ such that for each $t \in [0, 1]$, $F(\cdot, t)$ is a homeomorphism from $\mathcal{M}$ to $\hat{\mathcal{M}}$, where $F(\cdot, 0)$ is the identity of $\mathcal{M}$ and $F(\mathcal{M}, 1) = \hat{\mathcal{M}}$. To establish that two surfaces are *geometrically close*, the distance between each point on one surface and its closest point on the other surface is required. Such a bound is usually obtained in the course of proving isotopy.

5.2.2 Diagrams and Triangulations

The set of points defining a Voronoi diagram are traditionally referred to as *sites* or *seeds*. When approximating a manifold by a set of sample points of varying density, it is helpful to assign weights to the points reflective of their density. In particular, a point p_i with weight w_i, can be regarded as a ball B_i with center p_i and radius $r_i = \sqrt{w_i}$.

Recall that the *power distance* [130] between two points p_i, p_j with weights w_i, w_j is $\pi(p_i, p_j) = \mathbf{d}(p_i, p_j)^2 - w_i - w_j$. Unless otherwise noted, points are *unweighted*, having weight equal to zero. There is a natural geometric interpretation of the weight: all points q on the boundary of B_i have $\pi(p_i, q) = 0$, inside $\pi(p_i, q) < 0$ and outside

$\pi(p_i, q) > 0$. Given a set of weighted points $\mathcal{P}$, this metric gives rise to a natural decomposition of $\mathbb{R}^3$ into the *power cells* $V_i = \{q \in \mathbb{R}^3 \mid \pi(p_i, q) \leq \pi(p_j, q) \, \forall p_j \in \mathcal{P}\}$. The *power diagram* $\mathrm{wVor}(\mathcal{P})$ is the cell complex defined by collection of cells V_i for all $p_i \in \mathcal{P}$.

The nerve [130] of a collection $\mathcal{C}$ of sets is defined as $\mathcal{N}(\mathcal{C}) = \{X \subseteq \mathcal{C} \mid \cap T \neq \emptyset\}$. Observe that $\mathcal{N}(\mathcal{C})$ is an abstract simplicial complex because $X \in \mathcal{N}(\mathcal{C})$ and $Y \subseteq X$ imply $Y \in \mathcal{N}(\mathcal{C})$. With that, we obtain the *weighted Delaunay triangulation*, or *regular triangulation*, as $\mathrm{wDel}(\mathcal{P}) = \mathcal{N}(\mathrm{wVor}(\mathcal{P}))$. Alternatively, $\mathrm{wDel}(\mathcal{P})$ can be defined directly as follows. A subset $T \subset \mathbb{R}^d$, with $d \leq 3$ and $|T| \leq d+1$ defines a d-simplex σ_T. Recall that the *orthocenter* [143] of σ_T, denoted by z_T, is the unique point $q \in \mathbb{R}^d$ such that $\pi(p_i, z_T) = \pi(p_j, z_T)$ for all $p_i, p_j \in T$; the *orthoradius* of σ_T is equal to $\pi(p, z_T)$ for any $p \in T$. The *Delaunay condition* defines $\mathrm{wDel}(\mathcal{P})$ as the set of tetrahedra σ_T with an *empty orthosphere*, meaning $\pi(p_i, z_T) \leq \pi(p_j, z_T)$ for all $p_i \in T$ and $p_j \in \mathcal{P} \setminus T$, where $\mathrm{wDel}(\mathcal{P})$ includes all faces of σ_T.

There is a natural duality between $\mathrm{wDel}(\mathcal{P})$ and $\mathrm{wVor}(\mathcal{P})$. For a tetrahedron σ_T, the definition of z_T immediately implies z_T is a *power vertex* in $\mathrm{wVor}(\mathcal{P})$. Similarly, for each k-face σ_S of $\sigma_T \in \mathrm{wDel}(\mathcal{P})$ with $S \subseteq T$ and $k+1 = |S|$, there exists a dual $(3-k)$-face σ'_S in $\mathrm{wVor}(\mathcal{P})$ realized as $\cap_{p \in S} V_p$. When $\mathcal{P}$ is unweighted, the same definitions yield the standard (unweighted) Voronoi diagram $\mathrm{Vor}(\mathcal{P})$ and its dual Delaunay triangulation $\mathrm{Del}(\mathcal{P})$.

5.2.3 Unions of Balls

Let $\mathcal{B}$ denote the set of balls corresponding to a set of weighted points $\mathcal{P}$ and define the *union of balls* $\mathcal{U}$ as $\cup\mathcal{B}$. It is quite useful to capture the structure of $\mathcal{U}$ using a combinatorial representation like a simplicial complex [130, 144]. Let f_i denote $V_i \cap \partial B_i$ and $\mathcal{F}$ the collection of all such f_i. Observing that $V_i \cap B_j \subseteq V_i \cap B_i \forall B_i, B_j \in \mathcal{B}$, f_i is equivalently defined as the spherical part of $\partial(V_i \cap B_i)$. Consider also the decomposition of $\mathcal{U}$ by the cells of wVor($\mathcal{P}$) into $\mathcal{C}(\mathcal{B}) = \{V_i \cap B_i \mid B_i \in \mathcal{B}\}$. The *weighted α-complex* $\mathcal{W}(\mathcal{P})$ is defined as the *geometric realization* of $\mathcal{N}(\mathcal{C}(\mathcal{B}))$ [130], i.e., $\sigma_T \in \mathcal{W}$ if $\{V_i \cap B_i \mid p_i \in T\} \in \mathcal{N}(\mathcal{C}(\mathcal{B}))$. It is not hard to see that $\mathcal{W}$ is a subcomplex of wDel($\mathcal{P}$).

To see why $\mathcal{W}$ is relevant, consider its *underlying space*; we create a collection containing the convex hull of each simplex in $\mathcal{W}$ and define the *weighted α-shape* $\mathcal{J}(\mathcal{P})$ as the union of this collection. It turns out that the simplices $\sigma_T \in \mathcal{W}$ contained in $\partial\mathcal{J}$ are dual to the faces of $\partial\mathcal{U}$ defined as $\cap_{i \in T} f_i$. Every point $q \in \partial\mathcal{U}$ defined by $\cap_{i \in T_q} f_i$, for $T_q \in \mathcal{B}$ and $k + 1 = |T_q|$, witnesses the existence of σ_{T_q} in $\mathcal{W}$; the k-simplex σ_{T_q} is said to be *exposed* and $\partial\mathcal{J}$ can be defined directly as the collection of all exposed simplices [144]. In particular, the *corners* of $\partial\mathcal{U}$ correspond to the facets of $\partial\mathcal{J}$. Moreover, $\mathcal{J}$ is homotopy-equivalent to $\mathcal{U}$ [130].

The union of balls defined using an ε-sampling guarantees the approximation of the manifold under suitable conditions on the sampling. Following earlier results on uniform sampling [145], an extension to non-uniform sampling establishes sampling conditions for the isotopic approximation of hypersurfaces and medial axis reconstruction [129].

5.3 Seeds Placement and Surface Reconstruction

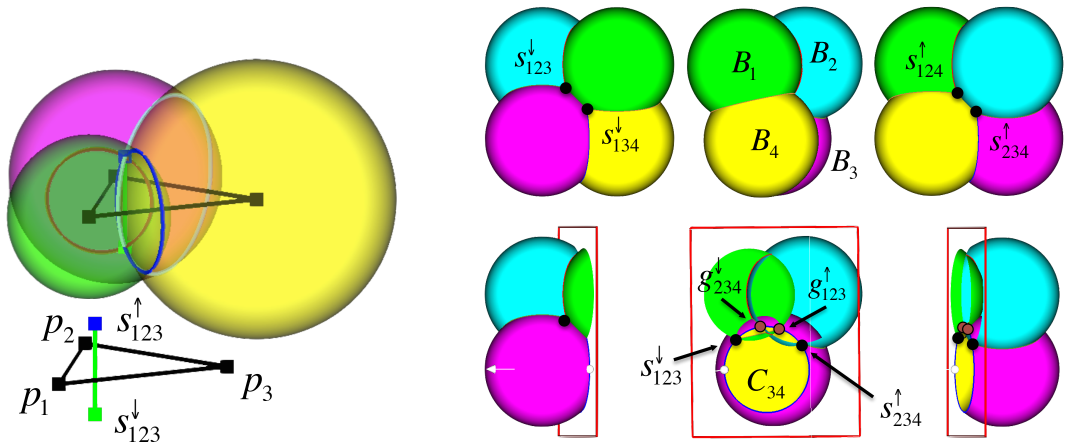
We determine the location of Voronoi seeds using the union of balls $\mathcal{U}$. The correctness of our reconstruction depends crucially on how sample balls $\mathcal{B}$ overlap. Assuming a certain structural property on $\mathcal{U}$, the surface reconstruction is embedded in the dual shape $\mathcal{J}$.

5.3.1 Seeds and Guides

Central to the method and analysis are triplets of sample spheres, i.e., boundaries of sample balls, corresponding to a *guide triangle* in wDel($\mathcal{P}$). The sample spheres associated with the vertices of a guide triangle intersect contributing a pair of *guide points*. The reconstruction consists of Voronoi facets, most of which are guide triangles.

When a triplet of spheres $\partial B_i, \partial B_j, \partial B_k$ intersect at exactly two points, the intersection points are denoted by $g^{\updownarrow}_{ijk} = \{g^{\uparrow}_{ijk}, g^{\downarrow}_{ijk}\}$ and called a pair of *guide points* or *guides*; see Figure 5.2a. The associated *guide triangle* t_{ijk} is *dual* to $g^{\updownarrow}_{ijk}$. We use arrows to distinguish guides on different sides of the manifold with the *upper* guide $g^{\uparrow}$ lying outside $\mathcal{O}$ and the *lower* guide $g^{\downarrow}$ lying inside. We refer to the edges of guide triangles as *guide edges* $e_{ij} = \overline{p_i p_j}$. A guide edge e_{ij} is associated with a dual *guide circle* $C_{ij} = \partial B_i \cap \partial B_j$, as in Figure 5.2a.

The Voronoi seeds in $\mathcal{S}^{\uparrow} \cup \mathcal{S}^{\downarrow}$ are chosen as the subset of guide points that lie



(a) Overlapping balls and guide circles.

(b) Arrangement of half-covered seed pairs.

Figure 5.2: (a) Guide triangle and its dual seed pair. (b) Cutaway view in the plane of circle C_{34}.

on $\partial\mathcal{U}$. A guide point g which is not interior to any sample ball is *uncovered* and included as a *seed* s into $\mathcal{S}$; covered guides are not. We denote *uncovered guides* by s and *covered guides* by g, whenever coverage is known and important. If only one guide point in a pair is covered, then we say the guide pair is *half-covered*. If both guides in a pair are covered, they are ignored. Let $\mathcal{S}_i = \mathcal{S} \cap \partial B_i$ denote the seeds on sample sphere ∂B_i.

As each guide triangle t_{ijk} is associated with at least one dual seed s_{ijk}, the seed witnesses its inclusion in $\mathcal{W}$ and t_{ijk} is exposed. Hence, t_{ijk} belongs to $\partial\mathcal{J}$ as well. When such t_{ijk} is dual to a single seeds s_{ijk} it bounds the interior of $\mathcal{J}$, i.e., it is a face of a *regular component* of $\mathcal{J}$; in the simplest and most common case, t_{ijk} is a facet of a tetrahedron as shown in Figure 5.3b. When t_{ijk} is dual to a pair of seeds $s^{\updownarrow}_{ijk}$, it does not bound the interior of $\mathcal{J}$ and is called a *singular face* of $\partial\mathcal{J}$. All singular faces of $\partial\mathcal{J}$ appear in the reconstructed surface.

5.3.2 Disk Caps

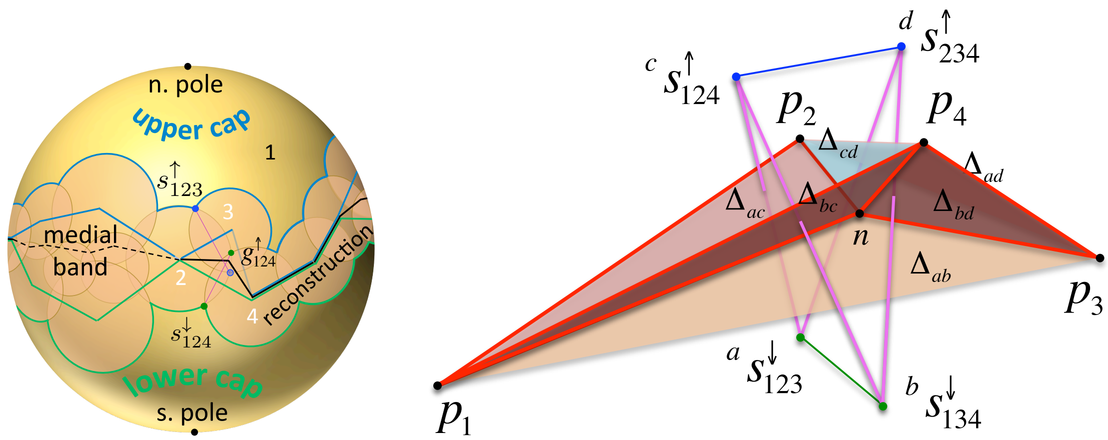
We describe the structural property required on $\mathcal{U}$ along with the consequences exploited by VoroCrust for surface reconstruction. This is partially motivated by the requirement that all sample points on the surface appear as vertices in the output Voronoi mesh.

We define the subset of ∂B_i inside other balls as the *medial band* and say it is *covered*. Let the caps $K_i^{\uparrow}$ and $K_i^{\downarrow}$ be the complement of the medial band in the interior and exterior of $\mathcal{O}$, respectively. Letting n_{p_i} be the normal line through p_i perpendicular to $\mathcal{M}$, the two intersection points $n_{p_i} \cap \partial B_i$ are called the *poles* of B_i. See Figure 5.3a.

We require that $\mathcal{U}$ satisfies the following structural property: each ∂B_i has *disk caps*, meaning the medial band is a *topological annulus* and the two caps contain the poles and are *topological disks*. In other words, each B_i contributes one connected component to each side of $\partial\mathcal{U}$. As shown in Figure 5.3a, all seeds in $\mathcal{S}_i^{\uparrow}$ and $\mathcal{S}_i^{\downarrow}$ lie on $\partial K_i^{\uparrow}$ and $\partial K_i^{\downarrow}$, respectively, along the arcs where other sample balls intersect ∂B_i. In Section 5.4, we establish sufficient sampling conditions to ensure $\mathcal{U}$ satisfies this property. In particular, we will show that both poles of each B_i lie on $\partial\mathcal{U}$.

The importance of disk caps is made clear by the following observation. The requirement that all sample points appear as Voronoi vertices in $\hat{\mathcal{M}}$ follows as a corollary.

Proposition 1 (Three upper/lower seeds)**.** *If ∂B_i has* disk caps*, then each of $\partial K_i^{\uparrow}$ and $\partial K_i^{\downarrow}$ has at least three seeds and the seeds on ∂B_i are not all coplanar.*



(a) Caps and medial band.

(b) Sliver and half-covered seeds.

Figure 5.3: (a) Decomposing the sample sphere ∂B_1. (b) Uncovered seeds and reconstruction facets. Let $\tau_p \in \mathcal{W}(\mathcal{P}) \subseteq \mathrm{wDel}(\mathcal{P})$ and $\tau_s \in \mathrm{Del}(\mathcal{S})$ denote the tetrahedra connecting the four samples and the four seeds shown, respectively. $s^{\downarrow}_{123}$ and $s^{\downarrow}_{134}$ are the uncovered lower guide seeds, with $g^{\uparrow}_{123}$ and $g^{\uparrow}_{134}$ covered. The uncovered upper guide seeds are $s^{\uparrow}_{124}$ and $s^{\uparrow}_{234}$, with $g^{\downarrow}_{124}$ and $g^{\downarrow}_{234}$ covered. $\triangle_{ac}$ is the Voronoi facet dual to the Delaunay edge between ${}^{a}s^{\downarrow}_{123}$ and ${}^{c}s^{\uparrow}_{124}$, etc. Voronoi facets dual to magenta edges are in the reconstructed surface; those dual to green and blue edges are not. n is the circumcenter of τ_s and appears as a Voronoi vertex in $\mathrm{Vor}(\mathcal{S})$ and a *Steiner vertex* in the surface reconstruction. In general, n is not the orthocenter of the sliver τ_p.

Proof. Every sphere $S_{j\neq i}$ covers strictly less than one hemisphere of ∂B_i because the poles are uncovered. Hence, each cap is composed of at least three arcs connecting at least three upper seeds $\mathcal{S}^{\uparrow}_i \subset \partial K^{\uparrow}_i$ and three lower seeds $\mathcal{S}^{\downarrow}_i \subset \partial K^{\downarrow}_i$. Further, any hemisphere through the poles contains at least one upper and one lower seed. It follows that the set of seeds $\mathcal{S}_i = \mathcal{S}^{\uparrow}_i \cup \mathcal{S}^{\downarrow}_i$ is not coplanar. $\square$

Corollary 1 (Sample reconstruction)**.** *If ∂B_i has disk caps, then p_i is a vertex in $\hat{\mathcal{M}}$.*

Proof. By Proposition 1, the sample is equidistant to at least four seeds which are not all coplanar. It follows that the sample appears as a vertex in the Voronoi diagram and not in the relative interior of a facet or an edge. Being a common vertex to at least one interior and one exterior Voronoi seed, VoroCrust retains this vertex in its output reconstruction. □

5.3.3 Sandwiching in the Dual Shape

Triangulations of smooth surfaces embedded in $\mathbb{R}^3$ can have half-covered guides pairs, with one guide covered by the ball of a fourth sample not in the guide triangle dual to the guide pair. The tetrahedron formed by the three samples of the guide triangle plus the fourth covering sample is a *sliver*, i.e., the four samples lie almost uniformly around the equator of a sphere. In this case we do not reconstruct the guide triangle, and also do not reconstruct some guide edges. We show that the reconstructed surface $\hat{\mathcal{M}}$ lies entirely within the region of space bounded by guide triangles, i.e., the α-shape of $\mathcal{P}$, as stated in the following theorem.

Theorem 6. *If all sample balls have disk caps, then $\hat{\mathcal{M}} \subseteq \mathcal{J}(\mathcal{P})$.*

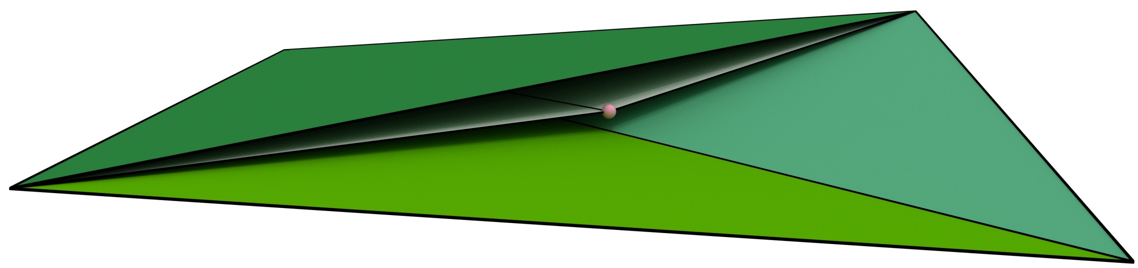


Figure 5.4: Cutaway view of a sliver tetrahedron $\tau_p \in \mathcal{W}(\mathcal{P}) \subseteq \mathrm{wDel}(\mathcal{P})$, drawn to scale. Half-covered guides give rise to the Steiner vertex (pink), which results in a surface reconstruction using four facets (only two are shown) sandwiched within τ_p. In contrast, filtering $\mathrm{wDel}(\mathcal{P})$ chooses two of the four facets of τ_p, either the bottom two, or the top two (only one is shown).

The simple case of a single isolated sliver tetrahedron is illustrated in Figures 5.3b, 5.4 and 5.2b. A sliver has a pair of lower guide triangles and a pair of upper guide triangles. For instance, t_{124} and t_{234} are the pair of upper triangles in Figure 5.3b. In such a tetrahedron, there is an edge between each pair of samples corresponding to a non-empty circle of intersection between sample balls, like the circles in Figure 5.2a. For this circle, the arcs covered by the two other sample balls of the sliver overlap, so each of these balls contributes exactly one uncovered seed, rather than two. In this way the upper guides for the upper triangles are uncovered, but their lower guides are covered; also only the lower guides of the lower triangles are uncovered. Theorem 6 follows directly from Theorem 2 in [131].

5.4 Sampling Conditions and Approximation Guarantees

We take as input a set of points $\mathcal{P}$ sampled from the bounding surface $\mathcal{M}$ such that $\mathcal{P}$ is an ε-sample, with $\varepsilon \leq 1/500$. We require that $\mathcal{P}$ satisfies the following sparsity condition: for any two points $p_i, p_j \in P$, $\mathrm{lfs}(p_i) \geq \mathrm{lfs}(p_j) \implies \mathbf{d}(p_i, p_j) \geq \sigma\varepsilon\mathrm{lfs}(p_j)$, with $\sigma \geq 3/4$. [DAVE: THIS IS A TYPESETTING NITPICK, BUT I WOULD PREFER THAT EXPRESSIONS LIKE εLFS(p) BE WRITTEN AS $\varepsilon \cdot$ LFS(p) OR WITH A BIT OF SPACE, AS IN $\varepsilon\,$LFS(p).]

Such a sampling $\mathcal{P}$ can be obtained by known algorithms. Given a suitable representation of $\mathcal{M}$, the algorithm in [146] computes a loose ε'-sample E which is a $\varepsilon'(1 + 8.5\varepsilon')$-sample. More specifically, whenever the algorithm inserts a new sample p into the set E, $\mathbf{d}(p, E) \geq \varepsilon'\mathrm{lfs}(p)$. To obtain E as an ε-sample, we set $\varepsilon'(\varepsilon) = (\sqrt{34\varepsilon + 1} - 1)/17$. Observing that $3\varepsilon/4 \leq \varepsilon'(\varepsilon)$ for $\varepsilon \leq 1/500$, the returned ε-sample satisfies our required sparsity condition with $\sigma \geq 3/4$.

5.4.1 The Medial Band

We start by adapting Theorem 6.2 and Lemma 6.4 from [129] to the setting just described. For $x \in \mathbb{R}^3 \setminus M$, let $\Gamma(x) = \mathbf{d}(x, \tilde{x})/\mathrm{lfs}(\tilde{x})$, where $\tilde{x}$ is the closest point to x on $\mathcal{M}$.

Corollary 2. *For an ε-sample $\mathcal{P}$, with $\varepsilon \leq 1/20$, the union of balls $\mathcal{U}$ with $\delta = 2\varepsilon$ satisfies:*

1. *$\mathcal{M}$ is a deformation retract of $\mathcal{U}$,*

2. *$\partial\mathcal{U}$ contains two connected components, each isotopic to $\mathcal{M}$,*

3. *$\Gamma^{-1}([0,a']) \subset U \subset \Gamma^{-1}([0,b'])$, where $a' = \varepsilon - 2\varepsilon^2$ and $b' \leq 2.5\varepsilon$.*

Proof. Theorem 6.2 from [129] is stated for balls with radii within $[a,b]$ times the lfs. We set $a = b = \delta$ and use $\varepsilon \leq 1/20$ to simplify fractions. This yields the above expressions for $a' = (1-\varepsilon)\delta - \varepsilon$ and $b' = \delta/(1-2\delta)$. The general condition requires $(1-a')^2 + \big(b' - a' + \delta(1+2b'-a')/(1-\delta)\big)^2 < 1$, as we assume no noise. Plugging in the values of a' and b', we verify that the inequality holds for the chosen range of ε. $\square$

Furthermore, we require that each ball $B_i \in \mathcal{B}$ contributes one facet to each side of $\partial\mathcal{U}$. Our sampling conditions ensure that both poles are outside any ball $B_j \in \mathcal{B}$.

Lemma 23 (Disk caps)**.** *All balls in $\mathcal{B}$ have disk caps for $\varepsilon \leq 0.066$, $\delta = 2\varepsilon$ and $\sigma \geq 3/2$.*

Proof. Fix a sample p_i and let x be one of the poles of B_i and $B_x = \mathbb{B}(c, \mathrm{lfs}(p_i))$ the tangent ball at p_i with $x \in B_x$. Letting p_j be the closest sample to x in $P \setminus \{p_i\}$, we assume the worst case where $\mathrm{lfs}(p_j) \geq \mathrm{lfs}(p_i)$ and p_j lies on ∂B_x. To simplify the calculations, take $\mathrm{lfs}(p_i) = 1$ and let ℓ denote $\mathbf{d}(p_i, p_j)$. As lfs is 1-Lipschitz, we get $\mathrm{lfs}(p_j) \leq 1 + \ell$. By the law of cosines, $\mathbf{d}(p_j, x)^2 = \mathbf{d}(p_i, p_j)^2 + \mathbf{d}(p_i, x)^2 - 2\mathbf{d}(p_i, p_j)\mathbf{d}(p_i, x)\cos(\phi)$, where $\phi = \angle p_j p_i c$. Letting $\theta = \angle p_i c p_j$, observe that $\cos(\phi) = \sin(\theta/2) = \ell/2$. To enforce $x \notin B_j$, we require $\mathbf{d}(p_j, x) > \delta\mathrm{lfs}(p_j)$, which is equivalent to $\ell^2 + \delta^2 - \delta\ell^2 > \delta^2(1+\ell)^2$. Simplifying, we get $\ell > 2\delta^2/(1-$

$\delta - \delta^2)$ where sparsity guarantees $\ell > \sigma\varepsilon$. Setting $\sigma\varepsilon > 2\delta^2/(1-\delta-\delta^2)$ we obtain $4\sigma\varepsilon^2 + (8+2\sigma)\varepsilon - \sigma < 0$, which requires $\varepsilon < 0.066$ when $\sigma \geq 3/4$. $\square$

Corollary 2 together with Lemma 23 imply that each ∂B_i is decomposed into a covered region $\partial B_i \cap \cup_{j \neq i} B_j$, the *medial band*, and two uncovered caps $\partial B_i \setminus \cup_{j \neq i} B_j$, each containing one pole. Recalling that seeds arise as pairs of intersection points between the boundaries of such balls, we show that seeds can be classified correctly as either inside or outside $\mathcal{M}$.

Corollary 3. *If a seed pair lies on the same side of $\mathcal{M}$, then at least one seed is covered.*

Proof. Fix such a seed pair $\partial B_i \cap \partial B_j \cap \partial B_k$ and recall that $\mathcal{M} \cap \partial B_i$ is contained in the medial band on ∂B_i. Now, assume for contradiction that both seeds are uncovered and lie on the same side of $\mathcal{M}$. It follows that $B_j \cap B_k$ intersects B_i away from its medial band, a contradiction to Corollary 2. $\square$

Corollary 2 guarantees that the medial band of B_i is a superset of $\Gamma^{-1}([0, a']) \cap \partial B_i$, which means that all seeds s_{ijk} are at least $a' \mathrm{lfs}(\tilde{s}_{ijk})$ away from $\mathcal{M}$.

5.4.2 Seeds and Guide Triangles

In addition to the topological properties of the medial band, we examine the geometry of the seeds and the guide triangles giving rise to the VoroCrust surface reconstruction. We start by bounding the elevation of such seeds above T_{p_i}, the *tangent plane* to $\mathcal{M}$ at p_i.

Lemma 24. *For a seed $s \in \partial B_i$, $\theta_s = \angle s p_i s' \geq 29.34°$ and $\theta_s > \frac{1}{2} - 5\varepsilon$, where s' is the projection of s on T_{p_i}, implying $\mathbf{d}(s, s') \geq h_s^{\perp} \delta lfs(p_i)$, with $h_s^{\perp} > 0.46$ and $h_s^{\perp} > \frac{1}{2} - 5\varepsilon$.*

Proof. Let $\text{lfs}(p_i) = 1$ and $B_s = \mathbb{B}(c, 1)$ be the tangent ball at p_i with $s \notin B_s$; see Figure 5.5a. Observe that $\mathbf{d}(s, \mathcal{M}) \leq \mathbf{d}(s, x)$, where $x = \overline{sc} \cap \partial B_s$. By the law of cosines, $\mathbf{d}(s, c)^2 = \mathbf{d}(p_i, c)^2 + \mathbf{d}(p_i, s)^2 - 2\mathbf{d}(p_i, c)\mathbf{d}(p_i, s)\cos(\pi/2 + \theta_s) = 1 + \delta^2 + 2\delta \sin(\theta_s)$. We may write[1] $\mathbf{d}(s, c) \leq 1 + \delta^2/2 + \delta \sin(\theta_s)$. It follows that $\mathbf{d}(s, x) \leq \delta^2/2 + \delta \sin(\theta_s)$. As lfs is 1-Lipschitz and $\mathbf{d}(p_i, x) \leq \delta$, we get $1 - \delta \leq \text{lfs}(x) \leq 1 + \delta$. There must exist a sample p_j such that $\mathbf{d}(x, p_j) \leq \varepsilon \text{lfs}(x) \leq \varepsilon(1 + \delta)$. Similarly, $\text{lfs}(p_j) \geq (1 - \varepsilon(1 + \delta))(1 - \delta)$. By the triangle inequality, $\mathbf{d}(s, p_j) \leq \mathbf{d}(s, x) + \mathbf{d}(x, p_j) \leq \delta^2/2 + \delta \sin(\theta_s) + \varepsilon(1 + \delta)$. Setting $\mathbf{d}(s, p_j) < \delta(1 - \delta)(1 - \varepsilon(1 + \delta))$ implies $\mathbf{d}(s, p_j) < \delta \text{lfs}(p_j)$, which shows that for small values of θ_s, s cannot be a seed and $p_j \neq p_i$. Substituting $\delta = 2\varepsilon$, we get $\theta_s \geq \sin^{-1}(2\varepsilon^3 - 5\varepsilon + 1/2) \geq 29.34°$ and $\theta_s > 1/2 - 5\varepsilon$. $\square$

We make frequent use of the following bound on the distance between related samples.

Proposition 2. *If $B_i \cap B_j \neq \emptyset$, then $\mathbf{d}(p_i, p_j) \in [\kappa_\varepsilon, \kappa\delta] \cdot lfs(p_i)$, with $\kappa = 2/(1 - \delta)$ and $\kappa_\varepsilon = \sigma\varepsilon/(1 + \sigma\varepsilon)$.*

Proof. The upper bound comes from $\mathbf{d}(p_i, p_j) \leq r_i + r_j$ and $\text{lfs}(p_j) \leq \text{lfs}(p_i) +$

[1] Define $f(u, v) = \sqrt{1 + u^2 + 2uv} - (1 + u^2/2 + uv)$ and observe that $f(u, -u/2) = 0$ is the only critical value of $f(u, .)$. As $\partial^2 f/\partial v^2 \leq 0$ for $(u, v) \in \mathbb{R} \times [-1, 1]$, we get that $f(u, v) \leq 0$ in this range.

$\mathbf{d}(p_i, d_j)$ by 1-Lipschitz, and the lower bound from $\text{lfs}(p_i) - \mathbf{d}(p_i, d_j) \leq \text{lfs}(p_j)$ and the sparsity. $\square$

Bounding the circumradii is the culprit behind why we need such small values of ε.

Lemma 25. *The circumradius of a guide triangle t_{ijk} is at most $\varrho_f \cdot \delta lfs(p_i)$, where $\varrho_f < 1.38$, and at most $\overline{\varrho}_f \cdot \boldsymbol{d}(p_i, p_j)$ where $\overline{\varrho}_f < 3.68$.*

Proof. Let p_i and p_j be the triangle vertices with the smallest and largest lfs values, respectively. From Claim 2, we get $\mathbf{d}(p_i, p_j) \leq \kappa\delta\text{lfs}(p_i)$. It follows that $\text{lfs}(p_j) \leq (1 + \kappa\delta)\text{lfs}(p_i)$. As t_{ijk} is a guide triangle, we know that it has a pair of intersection points $\partial B_i \cap \partial B_j \cap \partial B_k$. Clearly, the seed is no farther than $\delta\text{lfs}(p_j)$ from any vertex of t_{ijk} and the orthoradius of t_{ijk} cannot be bigger than this distance.

Recall that the weight w_i associated with p_i is $\delta^2\text{lfs}(p_i)^2$. We shift the weights of all the vertices of t_{ijk} by the lowest weight w_i, which does not change the orthocenter. With that $w_j - w_i = \delta^2(\text{lfs}(p_j)^2 - \text{lfs}(p_i)^2) \leq \delta^2\text{lfs}(p_i)^2((1+\kappa\delta)^2 - 1) = \kappa\delta^3\text{lfs}(p_i)^2(\kappa\delta + 2)$. On the other hand, sparsity ensures that the closest vertex in t_{ijk} to p_j is at distance at least $N(p_j) \geq \sigma\varepsilon\text{lfs}(p_j) \geq \sigma\varepsilon(1 - \kappa\delta)\text{lfs}(p_i)$. Ensuring $\alpha^2 \leq (w_j - w_i)/N(p_i)^2 \leq \kappa\delta^3(2 + \kappa\delta)/(\sigma^2\varepsilon^2(1 - \kappa\delta)^2) \leq 1/4$ suffices to bound the circumradius of t_{ijk} by $c_{rad} = 1/\sqrt{1 - 4\alpha^2}$ times its orthoradius, as required by Claim 4 in [143]. Substituting $\delta = 2\varepsilon$ and $\sigma \geq 3/4$ we get $\alpha^2 \leq 78.97\varepsilon$, which corresponds to $c_{rad} < 1.37$. It follows that the circumradius is at most $c_{rad}\delta\text{lfs}(p_j) \leq c_{rad}(1 + \kappa\delta)\delta\text{lfs}(p_i) < 1.38\delta\text{lfs}(p_i)$.

For the second statement, observe that $\text{lfs}(p_i) \geq (1 - \kappa\delta)\text{lfs}(p_j)$ and the sparsity condition ensures that the shortest edge length is at least $\sigma\varepsilon \mathrm{lfs}(p_i) \geq \sigma\varepsilon(1-\kappa\delta)\mathrm{lfs}(p_j)$. It follows that the circumradius is at most $\frac{\delta c_{rad}}{\sigma\varepsilon(1-\kappa\delta)} < 3.68$ times the length of any edge of t_{ijk}. $\square$

Given the bound on the circumradii, we are able to bound the deviation of normals.

Lemma 26. *If t_{ijk} is a guide triangle, then (1) $\angle_a(n_{p_i}, n_{p_j}) \leq \eta_s\delta < 0.47°$, with $\eta_s < 2.03$, and (2) $\angle_a(n_t, n_{p_i}) \leq \eta_t\delta < 1.52°$, with $\eta_t < 6.6$, where n_{p_i} is the line normal to $\mathcal{M}$ at p_i and n_t is the normal to t_{ijk}. In particular, t_{ijk} makes an angle at most $\eta_t\delta$ with T_{p_i}.*

Proof. Proposition 2 implies $\mathbf{d}(p_i, p_j) \leq \kappa\delta \mathrm{lfs}(p_i)$ and (1) follows from the Normal Variation Lemma [147] with $\rho = \kappa\delta < 1/3$ yielding $\angle_a(n_{p_i}, n_{p_j}) \leq \kappa\delta/(1-\kappa\delta)$. Letting R_t denote the circumradius of t, Lemma 25 implies that the $R_t \leq \varrho_f \cdot \delta \mathrm{lfs}(p_i) \leq \mathrm{lfs}(p_i)/\sqrt{2}$ and the Triangle Normal Lemma [148] implies $\angle_a(n_{p^*}, n_t) < 4.57\delta < 1.05°$, where p^* is the vertex of t subtending a maximal angle in t. Hence, $\angle_a(n_{p_i}, n_t) \leq \angle_a(n_{p_i}, n_{p^*}) + \angle_a(n_{p^*}, n_t)$. $\square$

5.4.3 Approximation Guarantees

Towards establishing homeomorphism, the next lemma on the monotonicity of distance to the nearest seed is critical. First, we show that the nearest seeds to any surface point $x \in \mathcal{M}$ are generated by nearby samples.

Lemma 27. *The nearest seed to $x \in \mathcal{M}$ lies on some ∂B_i where $\boldsymbol{d}(x, p_i) \leq 5.03 \cdot \varepsilon lfs(x)$. Consequently, $\boldsymbol{d}(x, p_i) \leq 5.08 \cdot \varepsilon lfs(p_i)$.*

Proof. In an ε-sampling, there exists a p_a such that $\mathbf{d}(x, p_a) \leq \varepsilon \mathrm{lfs}(x)$, where $\mathrm{lfs}(p_a) \leq (1+\varepsilon)\mathrm{lfs}(x)$. The sampling conditions also guarantee that there exists at least one seed s_a on ∂B_a. By the triangle inequality, we get that $\mathbf{d}(x, s_a) \leq \mathbf{d}(x, p_a) + \mathbf{d}(p_a, s_a) \leq \varepsilon \mathrm{lfs}(x) + \delta \mathrm{lfs}(p_a) \leq \varepsilon(1 + 2(1+\varepsilon))\mathrm{lfs}(x) = \varepsilon(2\varepsilon + 3)\mathrm{lfs}(x)$.

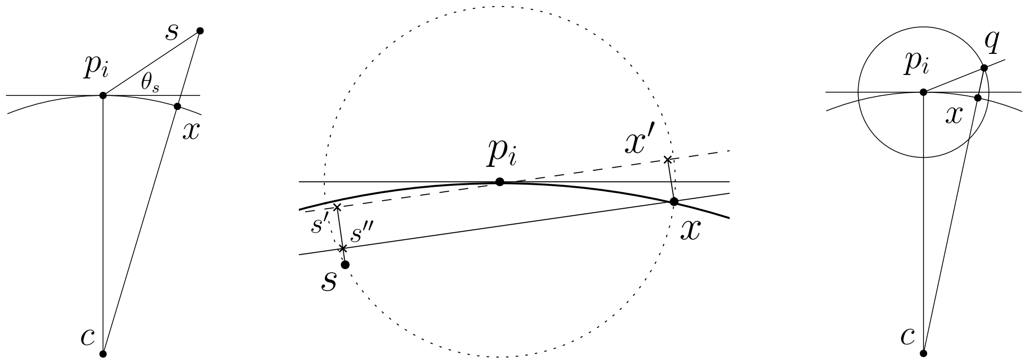
We aim to bound ℓ to ensure $\forall p_i$ s.t. $\mathbf{d}(x, p_i) = \ell \cdot \varepsilon \mathrm{lfs}(x)$, the nearest seed to x cannot lie on B_i. Note that in this case, $(1 - \ell\varepsilon)\mathrm{lfs}(x) \leq \mathrm{lfs}(p_i) \leq (1+\ell\varepsilon)\mathrm{lfs}(x)$. Let s_i be any seed on B_i. It follows that $\mathbf{d}(x, s_i) \geq \mathbf{d}(x, p_i) - \mathbf{d}(p_i, s_i) \geq \ell \cdot \varepsilon \mathrm{lfs}(x) - 2\varepsilon \mathrm{lfs}(p_i) \geq \varepsilon\big((1-2\varepsilon)\ell - 2\big)\mathrm{lfs}(x)$.

Setting $\varepsilon\big((1-2\varepsilon)\ell - 2\big)\mathrm{lfs}(x) \geq \varepsilon(2\varepsilon+3)\mathrm{lfs}(x)$ suffices to ensure $\mathbf{d}(x, s_i) \geq \mathbf{d}(x, s_a)$, and we get $\ell \geq (2\varepsilon+5)/(1-2\varepsilon)$. Conversely, if the nearest seed to x lies on B_i, it must be the case that $\mathbf{d}(x, p_i) \leq \ell\varepsilon \mathrm{lfs}(x)$. We verify that $\ell\varepsilon = \varepsilon(2\varepsilon+5)/(1-2\varepsilon) < 1$ for any $\varepsilon < 0.13$. It follows that $\mathbf{d}(x, p_j) \leq \ell\varepsilon/(1-\ell\varepsilon)\mathrm{lfs}(p_i)$. $\square$

Lemma 28. *For any normal segment N_x issued from $x \in \mathcal{M}$, the distance to $\mathcal{S}^{\uparrow}$ is either strictly increasing or strictly decreasing along $\Gamma^{-1}([0, 0.96\varepsilon]) \cap N_x$. The same holds for $\mathcal{S}^{\downarrow}$.*

Proof. Let n_x be the outward normal and T_x be the tangent plane to $\mathcal{M}$ at x. By Lemma 27, the nearest seeds to x are generated by nearby samples. Fix one such nearby sample p_i. For all possible locations of a seed $s \in \mathcal{S}^{\uparrow} \cap \partial B_i$, we will show a sufficiently large lower bound on $\langle s - s'', n_x \rangle$, where s'' the projection of s onto T_x.

Take $\mathrm{lfs}(p_i) = 1$ and let $B_s = \mathbb{B}(c, 1)$ be the tangent ball to $\mathcal{M}$ at p_i with $s \in B_s$. Let A be the plane containing $\{p_i, s, x\}$. Assume in the worst case that $A \perp T_{p_i}$ and x is as far as possible from p_i on $\partial B_s \cap T_{p_i}$. By Lemma 27, $\mathbf{d}(p_i, x) \leq 5.08\varepsilon$ and



(a) Seed elevation θ_s. (b) Bounding seed height above T_x. (c) Bounding $\mathbf{d}(q, \mathcal{M})$.

Figure 5.5: Constructions used for (a) Lemma 24, (b) Lemma 28 and (c) Theorem 7.

it follows that $\theta_x = \angle(n_x, n_{p_i}) \leq 5.08\varepsilon/(1 - 5.08\varepsilon) \leq 5.14\varepsilon$. This means that T_x is confined within a $(\pi/2 - \theta_x)$-cocone centered at x. Assume in the worst case that n_x is parallel to A and T_x is tilted to minimize $\mathbf{d}(s, s'')$; see Figure 5.5b.

Let T'_x be a translation of T_x such that $p_i \in T'_x$ and denote by x' and s' the projections of x and s, respectively, onto T'_x. Observe that T'_x makes an angle θ_x with T_{p_i}. From the isosceles triangle $\triangle p_i c x$, we get that $\theta'_x \leq 1/2\angle p_i c x = \sin^{-1} 5.08\varepsilon/2 \leq 2.54\varepsilon$. Now, consider $\triangle p_i x x'$ and let $\phi = \angle x p_i x'$. We have that $\phi = \theta_x + \theta'_x \leq 2.54\varepsilon + \delta/(1 - \delta) \leq 4.55\varepsilon$. Hence, $\sin(\phi) \leq 4.55\varepsilon$ and $\mathbf{d}(x, x') \leq 5.08\varepsilon \sin(\phi) \leq 0.05\varepsilon$. On the other hand, we have that $\angle s p_i s' = \psi \geq \theta_s - \theta_x$ and $\mathbf{d}(s, s') \geq \delta \sin\psi$, where $\theta_s \geq 1/2 - 5\varepsilon$ by Lemma 24. Simplifying we get $\sin(\psi) \geq 1/2 - 10.08\varepsilon$. The proof follows by evaluating $\mathbf{d}(s, s'') = \mathbf{d}(s, s') - \mathbf{d}(x, x')$. $\square$

Theorem 7. *For every $x \in \mathcal{M}$ with closest point $q \in \hat{\mathcal{M}}$, and for every $q \in \hat{\mathcal{M}}$ with closest point $x \in \mathcal{M}$, we have $\|xq\| < h_t \cdot \varepsilon^2 lfs(x)$, where $h_t < 30.52$. For $\varepsilon < 1/500$, $h_t \cdot \varepsilon^2 < 0.0002$. Moreover, the restriction of the mapping π to $\hat{\mathcal{M}}$ is a* *homeomorphism and $\hat{\mathcal{M}}$ and $\mathcal{M}$ are ambient isotopic. Consequently, $\hat{\mathcal{O}}$ is ambient isotopic to $\mathcal{O}$ as well.*

Proof. Fix a sample $p_i \in \mathcal{P}$ and a surface point $x \in \mathcal{M} \cap B_i$. We consider two cocones centered at x: a p-cocone contains all nearby surface points and a q-cocone contains all guide triangles incident at p_i. By Theorem 6, all reconstruction facets generated by seeds on B_i are sandwiched in the q-cocone.

Lemma 26 readily provides a bound on the q-cocone angle as $\gamma \leq \eta_t \delta$. In addition, since $\mathbf{d}(p_i, x) \leq \delta \mathrm{lfs}(p_i)$, we can bound the p-cocone angle as $\theta \leq 2\sin^{-1}(\delta/2)$ by Lemma 2 in [122]. We utilize a mixed pq-cocone with angle $\omega = \gamma/2 + \theta/2$, obtained by gluing the lower half of the p-cocone with the upper half of the q-cocone.

Let $q \in \hat{\mathcal{M}}$ and consider its closest point $x \in \mathcal{M}$. Again, fix $p_i \in \mathcal{P}$ such that $x \in B_i$; see Figure 5.5c. By sandwiching, we know that any ray through q intersects at least one guide triangle, in some point y, after passing through x. Let us assume the worst case that y lies on the upper boundary of the pq-cocone. Then, $\mathbf{d}(q, x) \leq \mathbf{d}(y, y') = h = \delta \sin(\omega) \mathrm{lfs}(p_i)$, where y' is the closest point on the lower boundary of the pq-cocone point to q. We also have that, $\mathbf{d}(p_i, x) \leq \cos(\omega)\delta \mathrm{lfs}(p_i) \leq \delta \mathrm{lfs}(p_i)$, and since lfs is 1-Lipschitz, $\mathrm{lfs}(p_i) \leq \mathrm{lfs}(x)/(1-\delta)$. Simplifying, we write $\mathbf{d}(q, x) < \delta\omega/(1-\delta) \cdot \mathrm{lfs}(x) < h_t \varepsilon^2 \mathrm{lfs}(x)$.

With $\mathbf{d}(q, x) \leq 0.55\varepsilon \mathrm{lfs}(x)$, Lemma 28 shows that the normal line from any $p \in \mathcal{M}$ intersects $\hat{\mathcal{M}}$ exactly once close to the surface. It follows that for every point $x \in \mathcal{M}$ with closest point $q \in \hat{\mathcal{M}}$, we have $\mathbf{d}(x, q) \leq \mathbf{d}(x, q')$ where $q' \in \hat{\mathcal{M}}$ with x its closest point in $\mathcal{M}$. Hence, $\mathbf{d}(x, q) \leq h_t \varepsilon^2 \mathrm{lfs}(x)$ as well.

Building upon Lemma 28, as a point moves along the normal line at x, it is either the case that the distance to $\mathcal{S}^{\uparrow}$ is decreasing while the distance to $\mathcal{S}^{\downarrow}$ is increasing or the other way around. It follows that these two distances become equal at exactly one point on the Voronoi facet above or below x separating some seed $s^{\uparrow} \in \mathcal{S}^{\uparrow}$ from another seed $s^{\downarrow} \in \mathcal{S}^{\downarrow}$. Hence, the restriction of the mapping π to $\hat{\mathcal{M}}$ is a homeomorphism.

This shows that $\hat{\mathcal{M}}$ and $\mathcal{M}$ homeomorphic. Recall that Corollary 2(3) implies $\mathcal{U}$ is a *topological thickening* [142] of $\mathcal{M}$. In addition, Theorem 6 guarantees that $\hat{\mathcal{M}}$ is embedded in the interior of $\mathcal{U}$, such that it separates the two surfaces comprising $\partial\mathcal{U}$. These three properties imply $\hat{\mathcal{M}}$ is isotopic to $\mathcal{M}$ in $\mathcal{U}$ by virtue of Theorem 2.1 in [142]. Finally, as $\hat{\mathcal{M}}$ is the boundary of $\hat{\mathcal{O}}$ by definition, it follows that $\hat{\mathcal{O}}$ is isotopic to $\mathcal{O}$ as well. □

5.5 Quality Guarantees and Output Size

Building upon the analysis in Section 5.4, we establish a number of quality guarantees on the output mesh. The main result is an upper bound on the *fatness* of all Voronoi cell, i.e., the outradius to inradius ratio where the outradius is the radius of the smallest enclosing ball, and the inradius is the radius of the largest enclosed ball.

5.5.1 Surface Elements

Recall that fatness is the outradius to inradius ratio, where the outradius is the radius of the smallest enclosing ball, and the inradius is the radius of the largest enclosed ball. The good quality of guide triangles allows us to bound the inradius of Voronoi cells.

Lemma 29. *For all guide triangles t_{ijk}: (1) Edge length ratios are bounded: $\ell_k/\ell_j \leq \kappa_\ell = \frac{2\delta}{1-\delta}\frac{\sigma\varepsilon}{1+\sigma\varepsilon}$. (2) Angles are bounded: $\sin(\theta_i) \geq 1/(2\overline{\varrho}_f)$ implying $\theta_i \in (7.8^\circ, 165^\circ)$. (3) Altitudes are bounded: the altitude above e_{ij} is at least $\alpha_t|e_{ij}|$, where $\alpha_t = 1/4\overline{\varrho}_f > 0.067$.*

Proof. The edge ratio bound is basically a restatement of Proposition 2. Denote by ℓ_i and θ_i the length of the triangle edge opposite to p_i and the angle at vertex p_i, respectively. Proposition 2 implies $\ell_k \leq \kappa\delta\text{lfs}(p_i)$ and the sparsity condition guarantees that $\ell_j \geq \kappa_\varepsilon\text{lfs}(p_i)$, hence $\ell_i/\ell_k \leq \kappa_\ell$ for any pair of edges.

Let R_{ijk} denote t_{ijk}'s circumradius. By the Central Angle Theorem, $\sin(\theta_i) = \ell_i/(2R_{ijk})$, and we also have $R_{ijk} \leq \overline{\varrho}_f\ell_i$ from Lemma 25. Hence $\sin(\theta_i) \geq 1/(2\overline{\varrho}_f)$.

For the worst case altitude, let the edge under consideration be the longest, $e = \ell_k$, and the second longest edge ℓ_j, so $\ell_j \geq \ell_k/2$. The altitude is then $\sin(\theta_i)\ell_j \geq \ell_k/(4\overline{\varrho}_f)$. □

The following technical lemma bounds the inradius of Voronoi cells with seeds in $\mathcal{S}^\uparrow \cup \mathcal{S}^\downarrow$.

Corollary 4. *If t_{ijk} is a guide triangle with associated seed s, then $\angle sp_is'' \geq \frac{1}{2} - \eta_t'\varepsilon$,* *where s'' is the projection of s on the plane of t_{ijk} and $\eta'_t \leq 5 + 2\eta_t < 18.18$, implying $\boldsymbol{d}(s, s'') \geq \hat{h}_s \delta lfs(p_i)$ with $\hat{h}_s \geq \frac{1}{2} - \eta'_t \varepsilon$.*

Proof. Combining Lemma 24 with Lemma 26, we have $\angle s p_i s'' \geq \angle s p_i s' - \angle_a(n_{t_{ijk}}, n_{p_i})$. $\square$

Observe that a guide triangle is contained in the Voronoi cell of its seed, even when one of the guides is covered. Hence, the tetrahedron formed by the triangle together with its seed lies inside the cell, and the cell inradius is at least the tetrahedron inradius.

Lemma 30. *For seeds $s_{ijk} \in \mathcal{S}^{\uparrow} \cup \mathcal{S}^{\downarrow}$, the inradius of the Voronoi cell is at least $\varrho_v \delta \cdot lfs(p_i)$ with $\varrho_v = \hat{h}_s / (1 + \frac{3}{2\sigma \varrho_f}) > 0.3$ and $\hat{h}_s \geq \frac{1}{2} - (5 + 2\eta_t)\varepsilon$.*

Proof. Fix a seed s_{ijk} and observe that $\{p_i, p_j, p_k\}$ belong to its Voronoi cell. By the convexity of the cell, it follows that the tetrahedron $T = p_i p_j p_k s_{ijk}$ is contained inside it. We establish a lower bound on the cell's inradius by bounding the inradius of T. Let f_i denote the facet of T opposite to p_i and f_0 denote t_{ijk}. Let A_i be the area of f_i.

Observe that the incenter c_T divides T into four smaller tetrahedra, one for each facet of T, where the distance from c_T to the plane of each facet is equal to the inradius r. This allows us to express the volume of T as $V = \sum_{i=0}^{3} r A_i / 3$. Hence, we have that $r = 3V / \sum_i A_i$. We may also express V as $H A_0 / 3$, where H is the distance from s_{ijk} to the plane of t_{ijk}. Substituting for V and factoring out A_0, we get that $r = H / (1 + \sum_{i>0}^{3} A_i / A_0)$.

Triangle area ratios A_i/A_0 are bounded because triangle angles are bounded, and edge lengths are bounded by the local feature size. Consider the edge $e_i = \overline{p_j p_k}$ common to f_i and t_{ijk} and let α_s and α_p be the altitudes of e_i in f_i and t_{ijk}, respectively. It follows that $A_i/A_0 = \alpha_s/\alpha_p$. Note α_s is less than the length of the longest edge of f_i.

Hence, assuming that $\text{lfs}(p_j) \geq \text{lfs}(p_k)$, we get that $\alpha_s \leq \delta\text{lfs}(p_j)$. On the other hand, the sparsity condition guarantees $\mathbf{d}(p_j, p_k) \geq \sigma\varepsilon\text{lfs}(p_j)$, allowing us to rewrite $\alpha_s \leq \frac{\delta}{\sigma\varepsilon}\mathbf{d}(p_j, p_k)$. From Lemma 29, we have that $\alpha_p \geq \mathbf{d}(p_j, p_k)/(4\overline{\varrho}_f)$. It follows that $A_i/A_0 \leq \frac{1}{2\sigma\overline{\varrho}_f}$. The proof follows by invoking Corollary 4 to bound $H \geq \hat{h}_s\delta\text{lfs}(p_i)$. □

5.5.2 Meshing the Interior

To get an upper bound on cell outradii, we must first generate seeds interior to $\mathcal{O}$. We consider a simple algorithm for generating $\mathcal{S}^{\downarrow\downarrow}$ based on a standard octree over $\mathcal{O}$. For sizing, we extend lfs beyond $\mathcal{M}$, using the point-wise maximal 1-Lipschitz extension $\text{lfs}(x) = \inf_{p\in\mathcal{M}}(\text{lfs}(p) + \mathbf{d}(x, p))$ [149]. An octree box $\square$ is refined if the length of its diagonal is greater than $2\delta \cdot \text{lfs}(c)$, where c is the center of $\square$. After refinement terminates, we add an interior seed at the center of each empty box, and do nothing with boxes containing one or more guide seeds.

Given an octree box $\square_i$, denote by c_i its center and r_i its radius (half its diagonal length). Assume that the input P has been scaled and shifted to fit into the unit cube $[0, 1]^3$. Starting with the unit cube as the box associated with the root node of the octree, the refinement process terminates with $r_i \leq \delta \mathrm{lfs}(c_i)$ for all leaf boxes $\square_i$. Note that refinement depends only on lfs and is independent of the number of points in P, and the distances between them. We establish the following Lipschitz-like properties for the size of leaf boxes.

Proposition 3. *If $\square_i$ is a leaf box, then $\frac{\delta}{2+\delta} lfs(c_i) \leq r_i \leq \delta lfs(c_i)$.*

Proof. By definition the leaf box was not split, so $r_i \leq \delta \mathrm{lfs}(c_i)$. Letting $\square_j$ be the parent of $\square_i$, it is clear that $\square_j$ had to be split. Hence, $r_j = 2r_i > \delta \mathrm{lfs}(c_j)$. By Lipschitzness, $\mathrm{lfs}(c_i) \leq \mathrm{lfs}(c_j) + r_i \leq r_i(1 + 2/\delta)$. $\square$

Proposition 4. *For any $p \in \square_i$, where $\square_i$ is a leaf box, $\frac{\delta}{2(1+\delta)} \leq r_i \leq \frac{\delta}{1-\delta} lfs(p)$.*

Proof. Observe that $\mathbf{d}(p, c_i) \leq r_i$, so $\mathrm{lfs}(p)$ is bounded in terms of $\mathrm{lfs}(c_i)$. Conveniently, Proposition 3 bounds $\mathrm{lfs}(c_i)$ in terms of r_i. To get the lower bound, we write $\mathrm{lfs}(p) \leq \mathrm{lfs}(c_i) + r_i \leq (\frac{2+\delta}{\delta} + 1) r_i$. For the upper bound, we write $\mathrm{lfs}(p) \geq \mathrm{lfs}(c_i) - r_i \geq (1/\delta - 1) r_i$. $\square$

Lemma 31. *If $\square_i$ and $\square_j$ are two leaf boxes sharing a corner, then $r_i/r_j \in [1/2, 2]$.*

Proof. Assume that $r_j \leq r_i$. From Proposition 3 we have $r_i \leq \delta \mathrm{lfs}(c_i)$ and $r_j \geq \frac{\delta}{2+\delta} \mathrm{lfs}(c_j)$. Together with lfs being 1-Lipschitz, this gives $r_j \geq \frac{\delta}{2+\delta} \big(\mathrm{lfs}(c_i) - (r_i + r_j)\big) \geq \frac{\delta}{2+\delta} (r_i/\delta - r_i - r_j)$. Simplifying, we get $r_j \geq \frac{r_i}{2} \frac{1-\delta}{1+\delta}$. For $\delta < 1/3$, we obtain $r_j > r_i/4$. As the ratio of box radii is a power of two, $r_j \in \{r_i/2, r_i\}$. $\square$

These propoerties of the octree may be used to bound the outradius of Voronoi cells.

Lemma 32. *The Voronoi cell of $s \in \mathcal{S}$ has outradius at most $\frac{2\delta}{1-3\delta}lfs(s) \leq \frac{4(1+\delta)}{1-3\delta} r_i$, where $\square_i$ is the leaf box containing s.*

Proof. Let v be a vertex on the Voronoi cell of s. The octree construction guarantees $v \in \square_j$, for some leaf box $\square_j$. Proposition 4 gives $r_j \leq \delta/(1-\delta)\mathrm{lfs}(v)$. Fixing some $s' \in \square_j \cap \mathcal{S} \neq \emptyset$, it follows that $\mathbf{d}(v,s) \leq \mathbf{d}(v,s') \leq 2r_j$. Hence, $\mathrm{lfs}(v) \geq \frac{1-\delta}{2\delta}\mathbf{d}(v,s)$. By Lipschitzness, $\mathrm{lfs}(s) \geq \mathrm{lfs}(v) - \mathbf{d}(v,s) \geq \frac{1-3\delta}{2\delta}\mathbf{d}(v,s)$. As $s \in \square_i$, Proposition 4 gives $\mathrm{lfs}(s) \leq \frac{2(1+\delta)}{\delta} r_i$. It follows that $\mathbf{d}(v,s) \leq \frac{2\delta}{1-3\delta}\mathrm{lfs}(s) \leq \frac{4(1+\delta)}{1-3\delta} r_i$. $\square$

5.5.3 Volumetric Cells

Any Voronoi vertex is in some box, and every box has at least one seed. This provides an upper bound on the distance between a Voronoi vertex and its closest seed, and an upper bound on the cell outradius, for both interior and guide seeds. Interior seeds are at the center of a box containing no other seeds, so interior cell inradius is at least a constant factor times r. Combining the outradius and inradius bounds provides the following results.

Lemma 33. *The fatness of interior cells is at most $\frac{8\sqrt{3}(1+\delta)}{1-3\delta} < 14.1$.*

Proof. Let $s \in \mathcal{S}$ be an interior seed and recall that s was inserted at the center of some empty leaf box $\square_i$. By construction, s is the only seed in $\square_i$. It follows that the inradius of $\mathrm{Vor}(s)$ is at least $\frac{1}{2\sqrt{3}} r_i$, which is half the distance from c_i to any of its sides. The proof follows from the bound on the outradius in terms of r_i as provided by Lemma 32. $\square$

Lemma 34. *The fatness of boundary cells is at most $\frac{4(1+\delta)}{(1-3\delta)(1-\delta)^2 \varrho_v} < 13.65$.*

Proof. Let $s \equiv s_{ijk} \in \mathcal{S}$ be a boundary seed and recall the lower bound of $\varrho_v \varepsilon \mathrm{lfs}(p_i)$ on the inradius of $\mathrm{Vor}(s)$ from Lemma 30. By Lipschitzness, we may express this as $\varrho_v \delta(1-\delta)\mathrm{lfs}(s)$. On the other hand, an upper bound of $\frac{4(1+\delta)}{1-3\delta} r_a$ on the circumradius of $\mathrm{Vor}(s)$ is provided by Lemma 32, where $\square_a$ is the leaf box containing s. From Proposition 4, we have that $r_a \leq \frac{\delta}{1-\delta}\mathrm{lfs}(s)$. With both bounds expressed in terms of $\mathrm{lfs}(s)$, we evaluate their ratio. $\square$

5.5.4 Size Bound

To bound the number of cells, we bound the integral of lfs^{-3} over the domain $\mathcal{O}$. As the integral is bounded over a single cell, it effectively counts the seeds.

Lemma 35. $|\mathcal{S}^{\downarrow\downarrow}| \leq 18\sqrt{3}/\pi \cdot \varepsilon^{-3} \int_{\mathcal{O}} \mathit{lfs}^{-3}$.

Proof. Let $\mathcal{I} = \mathcal{S}^{\downarrow} \cup \mathcal{S}^{\downarrow\downarrow}$ and $V(s)$ denote the Voronoi cell of seed s. Since the Voronoi cells of interior seeds in $\mathcal{I}$ partition the volume $\mathcal{O}$, $\int_{\mathcal{O}} \mathrm{lfs}^{-3} = \sum_{s\in\mathcal{I}} \int_{V(s)} \mathrm{lfs}^{-3}$. Bounded outradii and inradii will bound each integral by as follows.

Fix a seed s and let R_s and r_s be the circumradius and inradius of $V(s)$, respectively. From Lemma 32, we have $R \leq \frac{2\delta}{1-3\delta}\mathrm{lfs}(s)$. By Lipschitzness, for any $x \in \mathrm{Vor}(s)$, $\mathrm{lfs}(x) \geq \frac{1-5\delta}{1-3\delta}\mathrm{lfs}(s)$. Thus, $\int_{\mathrm{Vor}(s)} \mathrm{lfs}^{-3} \geq f_1(\delta)\mathrm{lfs}^{-3}(s)\mathrm{vol}(\mathrm{Vor}(s))$, where $f_1(\delta) = \left(\frac{1-3\delta}{1-5\delta}\right)^3$.

If $s \in \mathcal{S}^{\downarrow\downarrow}$, Proposition 4 yields $r_s \geq \frac{\delta}{4\sqrt{3}(1+\delta)}\mathrm{lfs}(s)$. Hence, $\mathrm{vol}(\mathrm{Vor}(s)) \geq f_2(\delta)\mathrm{lfs}^3(s)$, where $f_2(\delta) = \frac{4\pi}{3}\left(\frac{\delta}{4\sqrt{3}(1+\delta)}\right)^3$. If $s = s_{ijk} \in \mathcal{S}^{\downarrow}$, Lemma 30 gives $r_s \geq \varrho_v \varepsilon \mathrm{lfs}(p_i)$. Recalling $\mathbf{d}(p_i, s_{ijk}) = \delta \mathrm{lfs}(p_i)$ and the extension of lfs to the interior of $\mathcal{O}$, we get $\mathrm{lfs}(s) \leq (1+\delta)\mathrm{lfs}(p_i)$. It follows that $r_s \geq \frac{\varrho_v \delta}{1+\delta}\mathrm{lfs}(s)$ and

$\mathrm{vol}(\mathrm{Vor}(s)) \geq f_3(\delta)\mathrm{lfs}^3(s)$, where $f_3(\delta) = \frac{4\pi}{3}\left(\frac{\varrho_v \delta}{1+\delta}\right)^3$.

Letting $f_4(\delta) = f_1(\delta) \cdot \min(f_2(\delta), f_3(\delta))$, we established that $\mathrm{vol}(\mathrm{Vor}(s)) \geq f_4(\delta)\mathrm{lfs}^3(s)$. Plugging that into the above bound, we get $\int_{\mathrm{Vor}(s)} \mathrm{lfs}^{-3} \geq f_4(\delta)$. Hence, $\int_{\mathcal{O}} \mathrm{lfs}^{-3} \geq f_4(\delta)|\mathcal{I}| \geq f_4(\delta)|\mathcal{S}^{\downarrow\downarrow}|$. The proof follows by observing that $\frac{1}{f_4(\delta)} \leq 18\sqrt{3}/\pi \cdot \varepsilon^{-3}$. $\square$

Chapter 6: Robust Sampling for Voronoi Meshing

Finite element methods traditionally use simplicial meshes, where well-known angle conditions prohibit skinny elements [150]. The limited degrees of freedom of linear tetrahedral as well as hexahedral elements often require excessive refinement when modeling complex geometries or domains undergoing large deformations, e.g., cutting, merging, fracturing, or adaptive refinement [61–64].

This motivated generalizations to general polyhedral elements, which enjoy larger degrees of freedom and have recently been in increasing demand in computer graphics [151], physically-based simulations [152], applied mathematics [153], computational mechanics [154] and computational physics [155]. A key advantage of general polyhedral elements is their superior ability to adjust to deformation [151, 156] and topological changes [157], while being less biased to principal directions compared to regular tessellations [158]. In addition, polyhedral elements typically have more neighbors, even at corners and boundaries, enabling better approximation of gradients and possibly higher accuracy using the same number of conventional elements [121].

To further ensure the fidelity of the discrete model, the fundamental properties of continuum equations have to be preserved [75]. A well-principled framework is enabled through the combined use of primal meshes and their orthogonal duals [76].

The power of orthogonal duals, exemplified by Voronoi-Delaunay meshes, has recently been demonstrated on a range of applications in computer graphics [80] and computational physics [83]. It is therefore imperative to develop new algorithms for primal-dual polyhedral meshing.

In this chapter, we present the design and implementation of VoroCrust: the first algorithm for meshing non-convex, non-smooth, and even non-manifold domains by conforming polyhedral Voronoi meshes. The implicit output mesh, compactly encoded by a set of Voronoi seeds, comes with an orthogonal dual defined by the corresponding Delaunay tetrahedralization. This makes VoroCrust one of the first robust and efficient algorithms for primal-dual polyhedral meshing. The crux of the algorithm is a robust refinement process that estimates a suitable sizing function to guide the placement of Voronoi seeds. This enables VoroCrust to protect all sharp features, and mesh the surface and interior into quality elements. We demonstrate the performance of the algorithm through a variety of challenging models, see Figure 6.5, and compare against state-of-the-art polyhedral meshing methods based on clipped Voronoi cells; see Figures 6.1 and 6.2.

6.1 Introduction

Despite many attempts to design a robust Voronoi meshing algorithm, a general solution to the problem remained elusive. In particular, a number of widely used numerical simulators for flow and transport models, e.g., TOUGH2 [159] and PFLOTRAN [160], compute gradients along nodal lines connecting neighboring cells, and

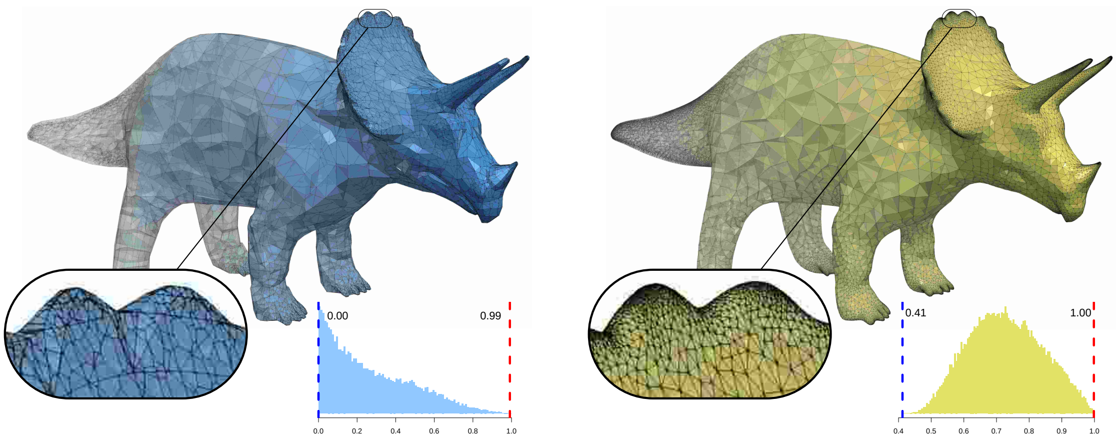


Figure 6.1: State-of-the-art methods for conforming Voronoi meshing clip Voronoi cells at the bounding surface. The Restricted Voronoi Diagram [66] (left) is sensitive to the input tessellation and produces surface elements of very low quality, per the shortest-to-longest edge ratio distribution shown in the inset. In contrast, VoroCrust (right) generates an unclipped Voronoi mesh conforming to a high-quality surface mesh.

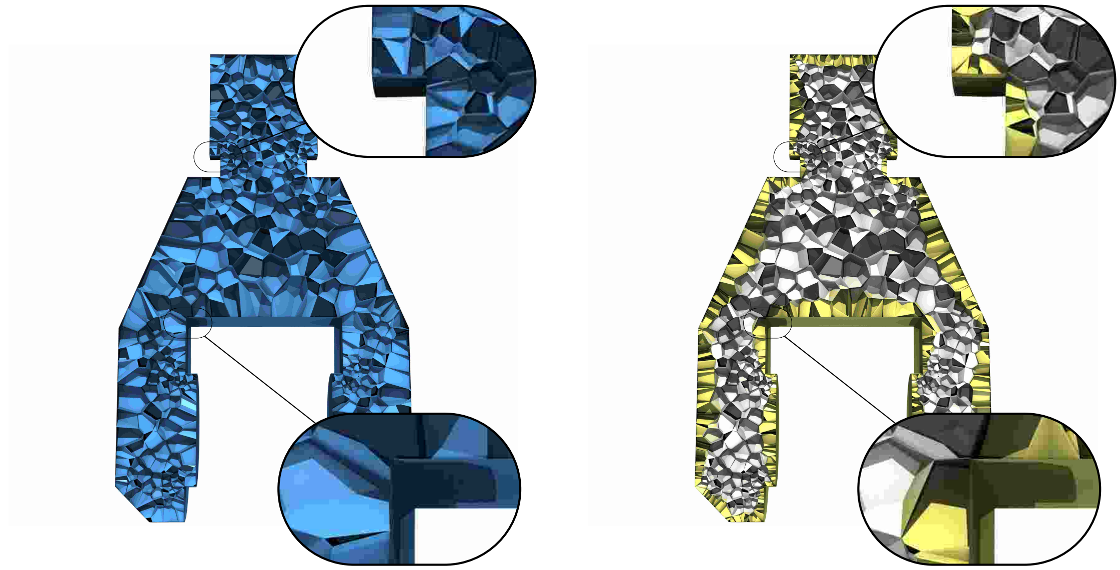
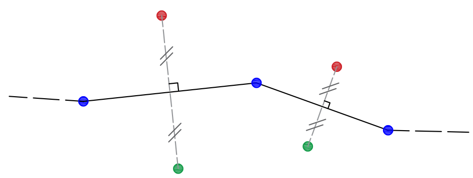


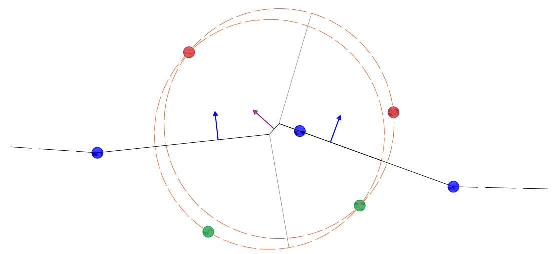
Figure 6.2: State-of-the-art clipping [66] may create non-convex cells (left); anywhere from 3% up to 96%. In contrast, VoroCrust always produces true Voronoi cells conforming to the boundary (right).

hence require that these dual edges are orthogonal to the common primal facets [161]. Several heuristic approaches to the generation of Voronoi meshes for such simulators were developed [135,162–165]. The situation is further complicated for multi-material domains, where the difficulty of generating conforming meshes necessitates dealing with mixed elements straddling the interface between multiple materials [166–168]. In contrast, VoroCrust is a well-principled algorithm for conforming Voronoi meshing that can handle a large class of domains having as boundary either a manifold or non-manifold surface with arbitrarily sharp features.

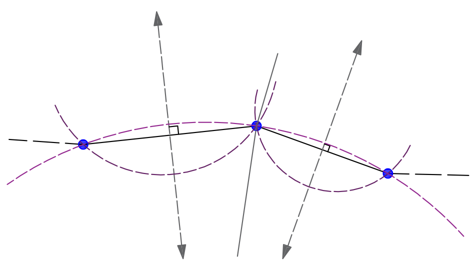
While PowerCrust successfully avoids misaligned facets, the placement of seeds as described is restricted to lie close to the medial axis resulting in very skinny Voronoi cells extending perpendicularly to the surface; see Figure 6.3(c). For the



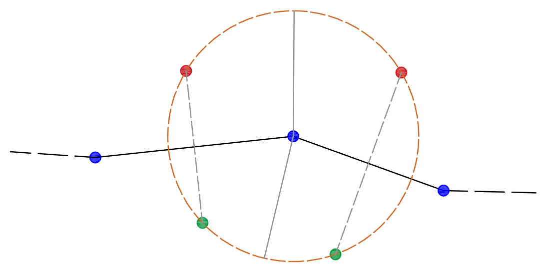
(a) Naive mirroring of seeds.



(b) Naive mirroring reconstruction.



(c) PowerCrust reconstruction.



(d) VoroCrust reconstruction.

Figure 6.3: Voronoi-based reconstruction interpolates boundary samples (blue) using the Voronoi facets generated by seeds on different sides of the boundary, e.g., inside (green) and outside (red). Naive mirroring (a) results in large normal deviations (b) due to Voronoi facets between non-paired seeds. PowerCrust reduces normal deviations by placing weighted seeds on the medial axis away from the boundary (c). VoroCrust eliminates misaligned facets (d) using unweighted seeds.

purposes of conforming Voronoi meshing, it is necessary to avoid such skinny cells. In contrast, VoroCrust is able to capture the surface using pairs of unweighted seeds placed close to the surface, enabling further decomposition of the interior using additional seeds; see Figure 6.3(d). A visual summary of the VoroCrust algorithm is provided in Figure 6.4.

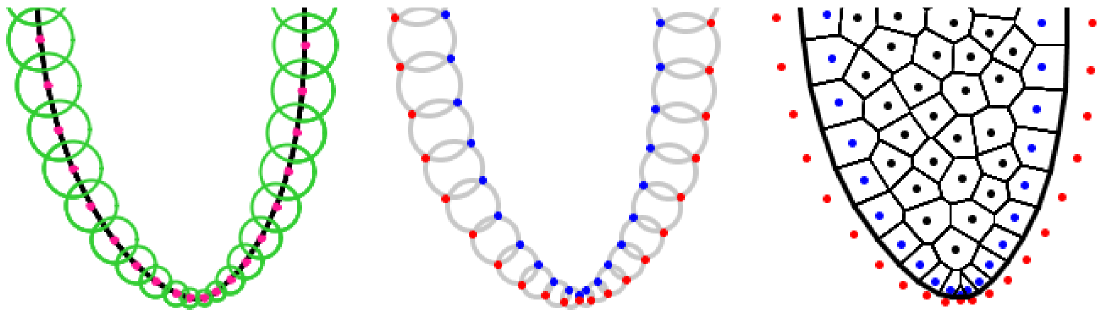
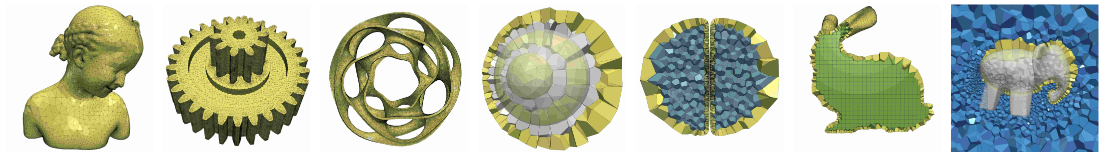


Figure 6.4: VoroCrust summary: (left) Cover the boundary by a union of balls, (middle) place pairs of Voronoi seeds where balls intersect to capture and isolate the boundary, and finally (right) seed the interior.

The issue of arbitrarily small input angles was finally resolved by Cheng et al. [59] for a large class of inputs called piecewise-smooth complexes. Cheng et al. [59] achieved that by deriving a feature size that blends the definitions used for smooth and polyhedral domains, ensuring the protection of sharp features. However, their algorithm is largely impractical as it relies on expensive predicates evaluated using the equations of the underlying surface. To obtain a practical variant as implemented in the DelPSC software, Dey and Levin [60] relied on an input threshold to guide refinement, where topological correctness can only be guaranteed if it is sufficiently small. Another issue with using such a threshold is the uniform sizing of the output mesh, since adaptive sizing requires better sensitivity to the underlying surface. In



(a) (b) (c) (d) (e) (f) (g)

Figure 6.5: VoroCrust can handle inputs having both smooth (a) and sharp (b) features as well as complex topology (c), multi-layers interfacing different types of materials (d), and multiple components (e). The enclosed volume is decomposed into convex unclipped Voronoi cells which can be optimized by CVT (e), controlled to exhibit dominant lattices structures (f), or generated by randomly-sampled seeds (g).

contrast, the proposed VoroCrust refinement leverages the quality of the input mesh to automatically estimate a sizing similar to the one defined by Cheng et al. [59, 169]; this enables VoroCrust to retain the superior guarantees they established while being practical as shown in our results.

The rest of this chapter is organized as follows. We describe all steps of the algorithm in Section 6.2. Then, we provide additional implementation details in Section 6.3. Finally, we present the evaluation and comparisons in Section 6.4.

6.2 The VoroCrust Algorithm

Given a representation of a domain vol, the algorithm produces a boundary-conforming Voronoi decomposition. The crux of the algorithm is the generation of a set of weighted surface samples corresponding to a set of balls $\mathcal{B}$ whose union $\mathcal{U} = \cup\mathcal{B}$ approximates the boundary $\mathcal{M} = \partial\text{vol}$. Specifically, $\mathcal{U}$ covers $\mathcal{M}$ and has the same topology. In addition, $\mathcal{U}$ captures the sharp features of $\mathcal{M}$. To further guarantee the quality of surface approximation, the radii of surface balls vary smoothly and are sufficiently small w.r.t. the local curvature of $\mathcal{M}$. In other words, the radii of balls in $\mathcal{B}$ mimic a local feature size for $\mathcal{M}$. Finally, certain configurations of balls are perturbed to eliminate undesirable artifacts in the output surface mesh. These requirements are used to design a refinement process that converges to a suitable union of balls. The conforming surface mesh is obtained by essentially dualizing $\mathcal{U}$ to obtain a set of Voronoi seeds $\mathcal{S}^{\updownarrow}$. Once $\mathcal{U}$ is obtained, the interior is easily meshed by sampling additional seeds $\mathcal{S}^{\downarrow\downarrow}$ outside $\mathcal{U}$. The output mesh can then be computed as a subset of the Voronoi diagram of the seeds in $\mathcal{S}^{\updownarrow} \cup \mathcal{S}^{\downarrow\downarrow}$ without any clipping. In the remainder of this section, we elaborate on these steps per the high-level pseudocode in Algorithm 1 and Figure 6.4.

6.2.1 Input Specification

VoroCrust can handle a domain vol having as boundary a piecewise-smooth complex (PSC) $\mathcal{M}$ that can be either manifold or non-manifold. The boundary PSC $\mathcal{M}$ possibly contains *sharp features* where the normal to the surface does not vary smoothly. We make no assumption on how small the input angles might be at such sharp features. VoroCrust guarantees the preservation of all sharp features; sharp corners appear exactly as vertices, while sharp creases are approximated by a set of edges.

Input Mesh. The algorithm takes as input a watertight piecewise-linear complex

(PLC) $\mathcal{T}$ approximating the boundary $\mathcal{M}$. As in [170], we assume that $\mathcal{T}$ approximates $\mathcal{M}$ in terms of both the Hausdorff error and the surface normals; this enables various predicates to be evaluated using the input PLC rather than the equations describing the underlying PSC [169]. In particular, we assume that all dihedral angles in the input mesh, except at sharp features, are at least $\pi - \theta^\flat$, where the *smoothness threshold* $\theta^\flat > 0$ is an implicit design parameter. For the current implementation, we assume $\mathcal{T}$ is a triangle mesh with no self-intersection. Well-established methods can be used to obtain such a mesh given a suitable representation of the domain vol [57, 60, 171].

Parameters. The algorithm also takes the following inputs:

- sz: a sizing field indicating the largest allowed *size* of mesh elements, and defaults to the diameter of $\mathcal{T}$ or ∞.
- $\theta^\sharp < \frac{\pi}{2}$: an angle threshold used to identify the *sharp features* in the PLC $\mathcal{T}$ and bound approximation errors.
- $L < 1$: a *Lipschitz* parameter that bounds the variation of radii in $\mathcal{B}$ and helps speed-up proximity queries.

We distinguish the angle parameters θ by the superscripts inspired from musical notation: $\sharp$ for sharp and $\flat$ for flat.

Algorithm 1: High-level VoroCrust algorithm

Input: PLC $\mathcal{T}$ approximating the domain vol, sizing field sz,

and parameters $\theta^\sharp$ and L (Section 2.1)

$\mathcal{F} \leftarrow$ the set of sharp features w.r.t. $\theta^\sharp$ (Section 2.2)

$\mathcal{B} \leftarrow$ a set of balls protecting all features in $\mathcal{F}$ (Section 2.3)

while $\mathcal{U} = \cup\mathcal{B}$ *does not cover* $\mathcal{T}$ **do**

Add balls to recover the protection of $\mathcal{F}$ and cover $\mathcal{T}$

Shrink balls violating any ball conditions (Section 2.3)

or forming half-covered seeds (Section 2.4)

end

$\mathcal{S}^\updownarrow \leftarrow$ pairs of seeds from triplets of balls in $\mathcal{B}$ (Section 2.4)

$\mathcal{S}^{\downarrow\downarrow} \leftarrow$ seeds sampled from the interior of vol $\setminus \mathcal{U}$ (Section 2.5)

return $\mathcal{S}^\updownarrow \cup \mathcal{S}^{\downarrow\downarrow}$

6.2.2 Preprocessing Steps

Before refinement, VoroCrust indexes the elements of the input PLC $\mathcal{T}$ and enforces the smoothness condition per the parameter $\theta^\flat$. Then, the algorithm constructs a number of data structures for proximity queries against $\mathcal{T}$ and $\mathcal{B}$.

Feature Detection. We define a *sharp edge* as an edge of $\mathcal{T}$ subtending a dihedral angle less than $\pi - \theta^\sharp$, or any non-manifold edge incident to exactly one or more than two facets. These sharp edges partition the set of facets incident to any fixed vertex into *sectors*. We define a *sharp corner* as a vertex of $\mathcal{T}$ incident to more than two sharp edges, or two sharp edges whose supporting lines make an angle less than $\pi - \theta^\sharp$, or two facets in the same sector whose normals differ by at least $\theta^\sharp$. A polyline arising from a chain of connected sharp edges is called a *crease*, and either forms a cycle or connects two sharp corners. The connected components of the boundary containing no sharp features, denoted $\mathcal{T}_S$, are called *surface patches*. The collection of sharp corners, creases and surface patches are collectively referred to as the *strata* of $\mathcal{T}$.

The algorithm uses $\theta^\sharp$ to test each edge in $\mathcal{T}$, and collects all sharp edges in a set E. Then, each vertex is tested using $\theta^\sharp$ and E, and the sharp corners are collected into the set $\mathcal{F}_C$. From E and $\mathcal{F}_C$, connected chains of sharp edges are collected into the set $\mathcal{F}_E$ by flooding through common vertices except for sharp corners. As a byproduct, each crease is given an index and an orientation, applied consistently to all its sharp edges. Similarly, the facets of $\mathcal{T}$ are indexed, oriented and collected into the set of surface patches $\mathcal{T}_S$ by flooding across non-sharp edges. Finally, we set $\mathcal{F} = \mathcal{F}_C \cup \mathcal{F}_E$.

Patch Smoothing. If the input mesh $\mathcal{T}$ does not satisfy the required bound on dihedral angles in terms of $\theta^\flat$, VoroCrust starts by applying adaptive loop subdivision [172] to ensure all dihedral angles between neighboring facets in the same surface patch in $\mathcal{T}_S$ are sufficiently large. In our implementation, we run 6 iterations of loop subdivision, applying subdivision adaptively such that facets with all associated dihedral angles larger than 175° are not subdivided. Typical values of $\theta^\flat$ resulting from this step range from 10° to 15°.

Proximity Queries. Upon generating a new sample point $p \in \mathcal{T}$, VoroCrust needs to find the balls in $\mathcal{B}$ covering p, and estimate its distance to the elements of $\mathcal{T}$ satisfying certain conditions w.r.t. $\theta^\sharp$. To speed up such queries, the algorithm constructs three *boundary k-d trees* to index the elements in $\mathcal{F}_C$, $\mathcal{F}_E$ and $\mathcal{T}_S$. The k-d trees for $\mathcal{F}_E$ and $\mathcal{T}_S$ are populated by supersampling the respective elements with a large number of samples proportional to their sizes. Similarly, the balls in $\mathcal{B}$ are indexed into three *ball k-d trees*. When querying the ball k-d trees for balls in the neighborhood of a given point, the L-Lipschitzness of ball radii helps to bound the range and overhead of such queries; see the appendix for more details.

6.2.3 Ball Refinement

At a high level, the desired union of balls $\mathcal{U}$ has to (1) protect the sharp features of $\mathcal{T}$ as in [169], and (2) cover $\mathcal{T}$ while matching its topology as in [173]. VoroCrust achieves this through a set of *ball conditions* imposed on the balls in $\mathcal{B}$. Violations of these conditions drive a refinement process which converges to a suitable union of balls. Before describing this process, we introduce a number of definitions and subroutines.

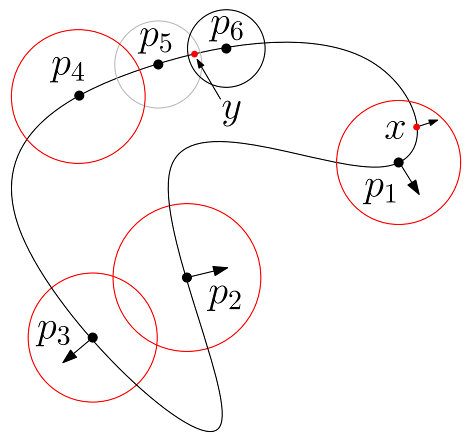


Figure 6.6: Ball conditions. C1 is violated at x by b_{p_1}. C2 is violated by b_{p_2} and b_{p_3}. C3 is violated by b_{p_4} and b_{p_5}. C4 is violated at y.

Smooth Neighborhoods. As in [169], we appeal to the curvature of the surface to infer a suitable notion of sizing. Fix a point $x \in \mathcal{T}$ and let σ be a face of $\mathcal{T}$ containing x. If σ is a sharp edge, define $v_{x,\sigma}$ as a unit vector parallel to σ. If σ is a surface patch, define $v_{x,\sigma}$ as a unit vector normal to σ. $v_{x,\sigma}$ inherits the orientation of the stratum, i.e., the crease or surface patch, containing σ. A path γ lying entirely in a unique stratum Σ is called a *smooth path* iff for all $x, y \in \gamma$ we have that $\angle v_{x,\sigma}, v_{y,\tau} \leq \theta^\sharp$, where σ and τ are the two top-dimensional faces of Σ containing x and y, respectively. Two points $x, y \in \mathcal{T}$ are called *co-smooth* iff they can be connected by a smooth path. For example, for the curve shown in Figure 6.6, if $\theta^\sharp = \pi/4$, then p_1 is not co-smooth with x while p_5 is co-smooth with p_6.

Ball Conditions. For a sample point $p \in \mathcal{T}$, let $b_p \in \mathcal{B}$ denote the ball centered at p and let r_p denote its radius. The following conditions drive the refinement process and are ensured for $\mathcal{B}$ upon termination; see Figure 6.6.

(C1) Smooth Coverage. For any $b_p \in \mathcal{B}$ and all $x \in b_p \cap \mathcal{T}$, we require that p and x are co-smooth.

(C2) Smooth Overlaps. For any $b_p, b_q \in \mathcal{B}$ s.t. $b_p \cap b_q \neq \emptyset$, we require that $b_p \cup b_q$ contains a smooth path from p to q.

(C3) Local L-Lipschitzness. For any two balls $b_p, b_q \in \mathcal{B}$ such that $p, q \in \mathcal{F}_C$, or $p, q \in \mathcal{F}_E$, or $p, q \in \mathcal{T}_S$, we require that $r_p \leq r_q + L \cdot \|p - q\|$.

(C4) Deep Coverage. Fix a constant $\alpha \in (0, 1)$. For all $x \in \mathcal{T}$, we require that $\|x - p\| \leq (1 - \alpha) \cdot r_p$ for some ball $b_p \in \mathcal{B}$. In addition, we require that $\|p - q\| \geq (1 - \alpha) \cdot \max(r_p, r_q)$ for all balls $b_p, b_q \in \mathcal{B}$.

Sizing Estimation. A *sizing* assigns to each new sample p a radius r_p. We seek a sizing at most sz that satisfies all ball conditions. VoroCrust computes such a sizing by dynamically evolving the assignments r_p for each ball $b_p \in \mathcal{B}$ in the course of the refinement process. To speed up convergence, a newly generated ball b_p is initialized with a conservative estimate that is more likely to satisfy all ball conditions. To help avoid C1 and C2 violations, the boundary k-d trees are queried using p to obtain a surrogate point q^* for the nearest non-co-smooth point on $\mathcal{T}$. To help avoid C3 violations, the ball k-d trees are queried to find the ball b_q whose center is nearest to p. With that, we set $r_p = \min(sz(p), 0.49 \cdot \|p - q^*\|, r_q + L \cdot \|p - q\|)$.

Termination. Since VoroCrust uses the PLC $\mathcal{T}$, which only provides a discrete approximation to the PSC $\mathcal{M}$, and approximates various distance queries, the sizing estimates as defined above may later be found to violate some ball conditions. By similar arguments to those in [60], refinement terminates satisfying all ball conditions. The intuition is that for each region on a crease or surface patch, there exists a positive lower bound on ball radii below which neither of the first two conditions can be violated. The refinement process resolves violations by *shrinking* some balls, effectively adjusting all sizing estimates, before recursing to restore protection and coverage. As demonstrated through a variety of challenging models, our algorithm is tuned to avoid excessive refinement; see Section 3.

6.2.4 Sampling Basics

The refinement process uses Maximal Poisson-Disk Sampling (MPS) [174–176] to generate the balls needed to protect the creases and cover the surface patches. The MPS procedure maintains an *active pool*, initialized by all faces on the stratum at hand. To generate a new sample, MPS starts by sampling a face σ from the active pool with a probability proportional to its measure, defined as the length for edges and the area for facets. Then, a point p is sampled from σ uniformly at random. If p is not covered by the balls in $\mathcal{B}$, it is assigned a radius r_p and the ball b_p is added into $\mathcal{B}$. Otherwise, p is discarded and a *miss counter* is incremented. Upon counting 100 successive misses, all faces in the active pool are *subdivided* into *subfaces* and the miss counter is reset; edges are split in half and facets are evenly split into four by connecting edge midpoints. Any subface whose points are all deeply covered is discarded, and the remaining subfaces become the new active pool.

Deep Coverage. For any point $x \in \mathcal{T}$, condition C4 dictates a stronger form of coverage by the balls in $\mathcal{B}$. We say that $x \in \mathcal{T}$ is α-*deeply covered* by a ball $b_p \in \mathcal{B}$ if $\|p - x\| \leq (1 - \alpha) \cdot r_p$; see Figure 6.6. We set $\alpha = 1 - \sqrt{3}/2 \approx 0.13$ in our implementation. Equivalently, we require adjacent balls to intersect *deeply*. The reason for that is twofold. First, any point x in the proximity of a crease Σ must be closer to the weighted samples on Σ than the samples on any other stratum of $\mathcal{T}$ [60]. Second, a sufficient distance between pairs of seeds is needed to bound the aspect ratio of Voronoi cells [173]. The refinement process ensures C4 by modifying the coverage test for MPS as follows. First, a new sample is only accepted if it is *not* deeply covered. Second, upon subdividing a face in the active pool, a subface is discarded only if it is completely deeply covered by a single ball with a co-smooth center. Third, the requirements of protecting sharp features prohibit deep overlaps between balls of different types; we elaborate on this further below following the description of our MPS implementation.

Detecting Violations. Before MPS discards a subface σ, the algorithm checks for violations of C1 or C2, and shrinks encroaching balls as follows. The algorithm starts by finding the nearest sample to σ on each stratum using the respective ball k-d tree. Then, the algorithm queries the trees for neighboring balls and checks whether σ is deeply covered by any of these balls. For each such ball b_p, the algorithm also checks whether p is co-smooth with the points of σ. If not, the algorithm finds the point $q^* \in \sigma$ minimizing the distance to p and shrinks b_p if necessary to ensure $r_p \leq 0.49 \cdot \|p - q^*\|$. By ensuring such b_p does not overlap σ, C1 violations are avoided. In addition, letting τ denote the subface containing p, any ball b_q with $q \in \sigma$ cannot overlap b_p. This effectively avoids C2 violations as the algorithm ensures $\max(r_p, r_q) \leq 0.49 \cdot \|p - q\|$ before σ and τ are both discarded. Finally, whenever the algorithm shrinks a ball, it needs to check for violations of C3 and possibly shrink more balls; the algorithm in [57] is similar in that regard. However, violations of C3 are not checked during the MPS procedure, which possibly terminates with such violations. As we describe below, enforcing C3 is interleaved with a later step to speed up convergence.

Testing Co-smoothness. Given two subfaces σ, τ on a stratum Σ and a point $p \in \tau$, our implementation uses a more practical test rather than computing smooth paths on Σ. This test is based on the observation that smooth paths starting at a subface σ are confined to small (co)cones of aperture $2\theta^\sharp$ emanating from the boundary of σ. In particular, the smooth neighborhood is nearly collinear or coplanar with σ if Σ is a crease or surface patch, respectively.

The algorithm starts by finding the point $q^* \in \sigma$ minimizing the distance to p, and sets $v_{pq^*} = p - q^*$. Then, the co-smoothness test is relaxed to only require that (1) $\angle v_{\sigma,q^*}, v_{\tau,p} \leq \theta^\sharp$ and (2) $\angle v_{\sigma,q^*}, v_{pq^*} \leq \theta^\sharp$ if Σ is a crease, or $\angle v_{\sigma,q^*}, v_{pq^*} \leq \frac{\pi}{2} - \theta^\sharp$ if Σ is a surface patch. We argue that this relaxed test suffices for the refinement process to eventually guarantee both C1 and C2. Let $\gamma \in \Sigma$ be any path from p to σ. If γ is a smooth path, then the test passes on all subfaces along γ. Otherwise, the test fails for some subface $\sigma' \in \gamma$. Hence, if no smooth path exists from p to σ, then every such path γ encounters a subface σ' for which the test fails before reaching

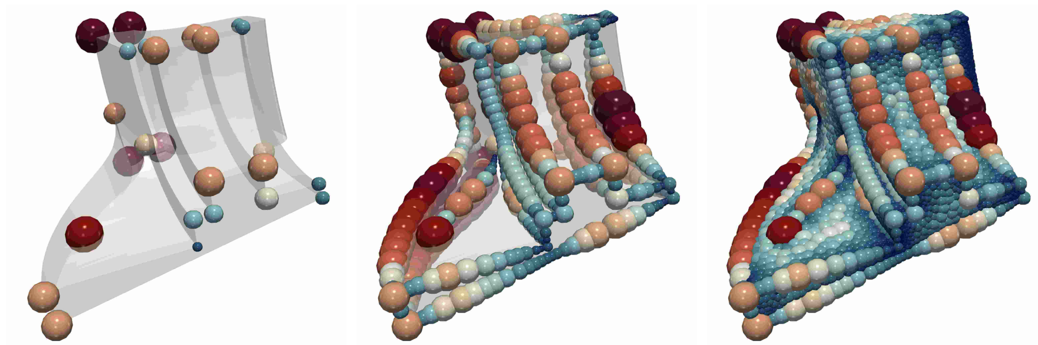


Figure 6.7: The three phases of VoroCrust refinement demonstrated on the Fandisk model: protection by corner balls (left) followed by edge balls (center), and finally coverage by surface balls (right).

σ. By applying the relaxed test to every subface σ and each ball in a sufficiently large neighborhood around σ, any remaining violations of C1 or C2 can be detected before MPS terminates. To further validate this claim, we implemented the strict test and verified that both C1 and C2 are always satisfied when MPS terminates.

6.2.5 Protection and Coverage

The refinement process is realized as a recursive MPS procedure (RMPS) that goes through three phases, ordered by the dimension of the underlying stratum, starting with the protection of sharp corners to the protection of creases and finally the coverage of surface patches; see Figure 6.7. At each phase, if refinement shrinks any of the balls belonging to a previous phase, the algorithm recurses by rerunning RMPS on the affected lower-dimensional strata before proceeding. The process starts by initializing the set of balls with one *corner ball* centered at each sharp corner. As the base case of RMPS, the algorithm enforces C3 among corner balls, shrinking balls as needed. Then, each crease Σ is protected by a set of *edge balls* by running RMPS on Σ. If any corner ball had to be shrunk, RMPS immediately recurses to adjust the corner balls. Whenever RMPS terminates on all creases, the algorithm enforces C3 on all edge balls and reruns RMPS as needed to restore protection. After successfully protecting all sharp corners and creases, the algorithm proceeds to cover each surface patch Σ by a set of *surface balls* by running RMPS on Σ. Similarly, if any corner or edge ball had to be shrunk, RMPS immediately recurses to the respective phase. Finally, the algorithm enforces C3 on surface balls. Before rerunning RMPS as needed to restore protection and coverage, the algorithm perturbs slivers, as we describe in Section 6.2.7; this helps refinement converge in fewer iterations.

We now turn back to the restrictions on overlaps between balls of different type. Whenever a subface encountered by RMPS is completely contained in a corner ball, it is excluded from RMPS in higher phases on neighboring strata. Similarly, whenever a subface is completely contained in an edge ball, it is excluded from RMPS on neighboring surface patches. This is necessary to ensure the protection of sharp features. As a consequence, the deep coverage condition C4 may be violated in the vicinity of sharp features. This contributes to the deterioration of element quality in these neighborhoods but otherwise does not threaten the termination of the algorithm; see Section 6.2.7.

6.2.6 Density Regulation

Extra care is needed to avoid the well-known clustering phenomenon resulting from the greedy generation of samples. This can be mitigated by biasing the sampling to avoid introducing new sample points near the boundaries of existing balls. In particular, whenever the radius assigned to a new sample p results in the ball b_p violating C4 by containing an existing sample, p is rejected with a small constant probability; we set this constant to 0.1 in our implementation. If p is not rejected, b_p is shrunk to ensure it satisfies C4. As demonstrated in Section 3, VoroCrust successfully avoids unnecessarily dense clusters of samples.

6.2.7 Surface Meshing

VoroCrust populates the set of *surface seeds* $\mathcal{S}^{\updownarrow}$ using triplets of overlapping balls in $\mathcal{B}$. The bounding spheres of each such triplet intersect in exactly two points on either side of the boundary. The algorithm places one labeled Voronoi seed at each such point as long as it does not lie in the interior of any fourth ball in $\mathcal{B}$. Then, the Voronoi facets common to two Voronoi seeds on different sides of the boundary constitute the resulting VoroCrust surface mesh which coincides with the weighted α-shape of the samples $\mathcal{W}$ inheriting the topology of $\mathcal{U}$ [131]. The deep coverage condition C4 guarantees that all samples p appear as vertices in the Voronoi diagram of $\mathcal{S}^{\updownarrow}$, with at least 4 seeds lying on ∂b_p. We point out that VoroCrust effectively remeshes the surface on-the-fly to reduce the complexity of the output within the tolerance specified by the input parameters. The quality of surface elements follows from L-Lipschitzness [173], with the exception of elements formed by corner or edge balls in the vicinity of sharp features.

Sliver Elimination. VoroCrust applies further refinement to the set of balls $\mathcal{B}$ to eliminate undesirable artifacts in the output. When a triplet of overlapping balls yield only one Voronoi seed, we have a *half-covered seed pair*. The four samples yielding the problematic configuration of balls are typically the vertices of a nearly flat tetrahedron appearing as a *regular component* in $\mathcal{W}$ [173]; we refer to such regular components as *slivers*. These slivers result in extra *Steiner vertices*, besides the samples, appearing in the Voronoi diagram of the seeds and consequently on the output surface mesh. As these Steiner vertices may not lie on the input surface, their incident Voronoi facets may not be aligned with the surface possibly yielding large deviations in surface normals; see Figure 6.8. To eliminate such slivers, the algorithm determines one ball to shrink for each half-covered seed.

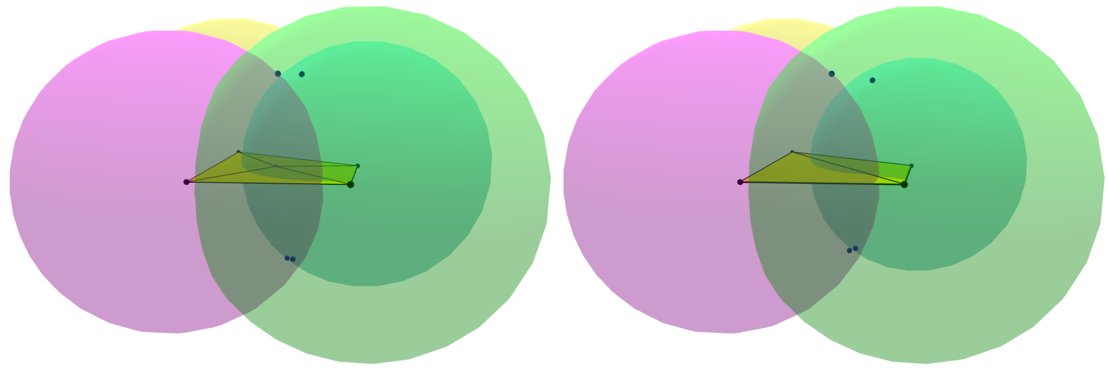


Figure 6.8: Sliver elimination: (left) A quartet of balls centered at four samples (black) with four half-covered seeds (blue) yielding a Steiner vertex (pink) with four incident facets. (right) Shrinking one ball resolves half-covered seeds eliminating the Steiner vertex to yield only two facets; see the supplemental materials for the numerical values.

For every ball $b_p \in \mathcal{B}$, the algorithm queries the ball k-d trees for neighboring balls and collects those overlapping b_p into the set $\mathcal{B}_p$. The algorithm iterates over $\mathcal{B}_p$ to form triplets of overlapping balls including b_p. For each such triplet t, the algorithm computes the pair of intersection points on their bounding spheres and tests whether the pair is half-covered by any fourth ball in $\mathcal{B}_p$; all candidate fourth balls along with the triplet in t are collected into a secondary set $\mathcal{B}_t$. Then, every quartet of balls in $\binom{\mathcal{B}_t}{4}$ defining a half-covered seed pair is considered in isolation. For each such quartet, the algorithm determines the ball requiring the least shrinkage to uncover all seeds. Over all quartets in $\binom{\mathcal{B}_t}{4}$, the ball requiring the least shrinkage is assigned a smaller radius. For each ball b, the algorithm records the smallest radius assigned to b over all quartets it is part of. Once all balls are processed, the algorithm shrinks every ball assigned a smaller radius. Recalling that L-Lipschitzness is satisfied for $\mathcal{B}$, $|\mathcal{B}_p|$ is kept small and the running time of this procedure is linear in $|\mathcal{B}|$. The procedure just described eliminates a subset of existing slivers but potentially violates some ball conditions and creates new slivers. The algorithm reruns RMPS to resolve such violations before repeating to eliminate any remaining slivers.

Each execution of the above procedure, followed by rerunning RMPS, counts as a single iteration of sliver elimination. The termination of the algorithm requires a finite bound on the number of such iterations, which can be established by bounding the shrinkage that may be applied to any ball through subsequent iterations. The intuition behind this bound is the well-known relationship between increasing the density of sampling and the increased local flatness of the surface approximation. Specifically, shrinkage decreases as the density increases. As it turns out, violations of the deep coverage condition C4 are the main cause for refinement after shrinking to eliminate slivers. The termination of the algorithm can be guaranteed by accepting a set of balls with no half-covered seeds as long as all boundary points are only α'-deeply covered, for some $\alpha' < \alpha$.

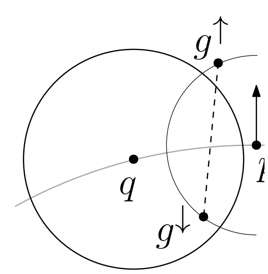


Figure 6.9: Bounding Δ.

Shrinkage Ratio. Fix a triplet t and let $g^{\uparrow}$ and $g^{\downarrow}$ denote the intersection points of its bounding spheres, such that t has a half-covered seed due to a fourth ball b_q. Assume w.l.o.g. that $g^{\downarrow} \in b_q$ while $g^{\uparrow} \notin b_q$, i.e., $\|q - g^{\downarrow}\| < r_q$ while $\|q - g^{\uparrow}\| \geq r_q$; see Figure 6.9. To resolve the half-covered seed, the algorithm shrinks b_q by setting its radius to $\|q - g^{\downarrow}\|$. Hence, the shrinkage is $r_q - \|q - g^{\downarrow}\| > 0$. As violations of α-deep coverage after shrinking are the main cause for further refinement, we consider shrinkage as a ratio of the original radius which we denote by Δ. The above inequalities imply the following bound: $\Delta = \frac{r_q - \|q-g^{\downarrow}\|}{r_q} \leq \frac{\|q-g^{\uparrow}\| - \|q-g^{\downarrow}\|}{\|q-g^{\downarrow}\|} = \frac{\|q-g^{\uparrow}\|}{\|q-g^{\downarrow}\|} - 1$. In particular, as $\frac{\|q-g^{\uparrow}\|}{\|q-g^{\downarrow}\|}$ approaches 1, α-deep coverage is less likely to be violated after shrinking. Specifically, if $\Delta \leq \frac{\alpha}{\alpha-2}$, then $\frac{\alpha}{2}$-deep coverage holds. Assuming the input $\mathcal{T}$ is sufficiently smooth per $\theta^{\flat}$, this observation guarantees the termination of the algorithm if $\frac{\alpha}{2}$-deep coverage is accepted.

6.2.8 Termination without Slivers

In this section, we formalize our claim of the termination of the proposed refinement process with additional iterations triggered by the shrinking performed in the course of sliver elimination as described in Section 2.4 in the paper. In particular, whenever a ball $b_q \in \mathcal{B}$ of radius r_q encroaches on a pair of seed locations $\{g^{\uparrow}, g^{\downarrow}\}$ such that it covers exactly one, w.l.o.g. $g^{\downarrow}$, the radius of this ball is reduced to $\|q - g^{\downarrow}\|$; see Figure 6.9. The main result of this section establishes that as the density of sampling increases, the maximum shrinkage $\Delta = \frac{r_q - \|p-g^{\downarrow}\|}{r_q}$ can be upper bounded in terms of the deviation of surface normals at the centers of overlapping balls in the current $\mathcal{B}$. Theorem 8 guarantees the termination of the algorithm by requiring that the dihedral angles of the input surface mesh $\mathcal{T}$ are at least $\pi - \theta^{\flat}$, except at sharp features.

Recall that the algorithm generates a set of balls $\mathcal{B}$ whose union covers the input surface $\mathcal{T}$. In particular, $\mathcal{B}$ is required to satisfy an α-deep coverage condition such that every surface point $x \in \mathcal{T}$ is contained in a ball $b_p \in \mathcal{B}$ of radius r_p such that $\|x - p\| \leq (1 - \alpha) \cdot r_p$. The main result of this section is then the guaranteed finite termination of one variant of the algorithm, where refinement stops if sliver elimination leaves all surface points $\frac{\alpha}{2}$-deeply covered, rather than α-deeply covered.

In what follows, we recall a few definitions from Section 2.3 in the paper. The parameter L bounds the variation in ball radii per the L-Lipschitzness condition dictating that for any two balls $b_p, b_q \in \mathcal{B}$ with p, q lying on the same surface patch Σ, we have that $r_p \leq r_q + L \cdot \|p - q\|$, i.e., the radii of balls covering Σ are L-Lipschitz. In addition, for any point $p \in \mathcal{T}$ and a facet σ, we denote by $v_{\sigma,p}$ a unit normal vector to σ at p.

Theorem 8. *Consider any ball $b_p \in \mathcal{B}$ with p lying on a facet σ on the surface patch Σ, and a pair of potential seed locations $g^{\uparrow}$ and $g^{\downarrow}$ on the boundary of b_p. Let $b_q \in \mathcal{B}$, with $q \in \Sigma$, be an encroaching ball containing exactly one of the seed locations. Assume in addition that the segment $g^{\uparrow}g^{\downarrow}$ makes an angle at most θ with $v_{\sigma,p}$, and the segment pq makes an angle at least $\frac{\pi}{2} - \theta$ with $v_{\sigma,p}$. If $\theta \leq \theta^{\flat}$, with $\theta^{\flat}$ depending on α and L, then the shrinkage Δ applied to b_q to resolve the encroachment maintains a relaxed $\frac{\alpha}{2}$-deep coverage condition. In particular, Δ can be bounded as*

$$\Delta = \max\left(\frac{\|q - g^{\uparrow}\|}{\|q - g^{\downarrow}\|}, \frac{\|q - g^{\downarrow}\|}{\|q - g^{\uparrow}\|}\right) - 1 < \frac{\alpha}{2 - \alpha}.$$

For example, using the default values of $\alpha = 1 - \frac{\sqrt{3}}{2}$ and $L = \frac{1}{4}$, the bound on the *ratio Δ holds as long as $\theta^\flat < 0.049°$. Fixing $\alpha = 1 - \frac{\sqrt{3}}{2}$, a simplified bound can be expressed as $\theta^\flat < \tan^{-1}\left(\frac{1}{1000}(1-L)^2\right)$.*

Before presenting the proof of Theorem 8, we start with a number of technical results. Observe that if the algorithm terminates earlier, then there is nothing to prove. Hence, we assume throughout that refinement eventually ensures all balls are sufficiently small such that any two balls b_p, b_q in $\mathcal{B}$ may only overlap if $\angle v_{\sigma,p}, v_{\tau,q} \leq \theta^\flat$, where σ, τ are the two faces containing p, q on some surface patch Σ.

The first proposition justifies the choice of the right hand side in Lemma 8. In particular, if the radius of the ball b_q is reduced from r_q to $(1-\Delta)\cdot r_q$, then $\frac{\alpha}{2}$-deep coverage holds.

Proposition 5. *Consider any ball $b_q \in \mathcal{B}$ and a point $x \in \mathcal{T}$ such that x is α-deeply covered by b_q. If b_q is shrunk to be of radius $r'_q \geq \left(1 - \frac{\alpha}{2-\alpha}\right)\cdot r_q$, then x is still $\frac{\alpha}{2}$-deeply covered by b_q.*

Proof. By the definition of deep coverage, $\|x-q\| \leq (1-\alpha)\cdot r_q$. Observing that $(1-\frac{\alpha}{2})\cdot(1-\frac{\alpha}{2-\alpha}) = 1-\alpha$, we can rewrite the bound as $\|x-q\| \leq (1-\frac{\alpha}{2})\cdot(1-\frac{\alpha}{2-\alpha})\cdot r_q \leq (1-\frac{\alpha}{2})\cdot r'_q$. $\square$

The next proposition shows that any seed is far from the surface, as coming closer puts it inside some ball.

Proposition 6. *Consider any ball $b_p \in \mathcal{B}$, and let g be a potential seed location on the boundary of b_p, and $g^\perp$ its projection on the plane T_p supporting any facet $\sigma \ni p$. Then, $\angle gpg^\perp \geq \phi - \theta^\flat$, where $\phi = \sin^{-1}\left(\alpha \cdot \frac{1-(1-\alpha)\cdot L}{1+(1-\alpha)\cdot L}\right)$.*

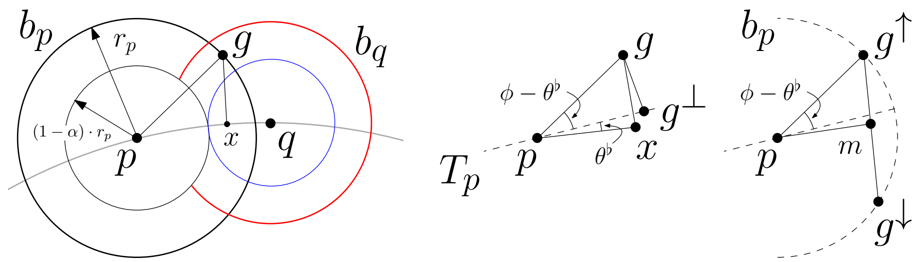


Figure 6.10: (left) Proposition 6: distance of a seed g to the tangent plane T_p. (right) Corollary 5: the midpoint m.

Proof. Assume for contradiction that $\angle gpg^\perp < \phi - \theta^\flat$. Letting x denote the closest point to g on $\mathcal{T}$, refinement ensures that $\|g - x\| < r_p \cdot \sin(\phi)$. Since $x \in \mathcal{T}$, deep coverage implies the existence of a sample q with $\|x - q\| \leq (1-\alpha) \cdot r_q$ such that $\|p - q\| \leq (1-\alpha) \cdot (r_p + r_q)$. In addition, by the L-Lipschitzness condition,

$$r_p \leq r_q + L \cdot \|p - q\| \leq r_q + L \cdot (1-\alpha) \cdot (r_p + r_q) \implies r_p \leq \frac{1 + (1-\alpha) \cdot L}{1 - (1-\alpha) \cdot L} \cdot r_q.$$

Figure 6.10 (left) depicts this situation by two tangent balls. Then, by the triangle inequality we get

$$\begin{aligned} \|q - g\| &\leq \|q - x\| + \|x - g\| < (1-\alpha) \cdot r_q + r_p \cdot \sin(\phi) \\ &\leq (1-\alpha) \cdot r_q + \frac{1 + (1-\alpha) \cdot L}{1 - (1-\alpha) \cdot L} \cdot r_q \cdot \sin(\phi) = r_q, \end{aligned}$$

which is a contradiction as g cannot be contained in b_q, as shown in Figure 6.10 (left). $\square$

Henceforth, ϕ is as defined in Proposition 6. As a corollary, we obtain bounds on the distance between any two seeds in a pair.

Corollary 5. *Let $g^{\uparrow}$ and $g^{\downarrow}$ be a pair of potential seed locations on a ball $b_p \in \mathcal{B}$, and m be the midpoint of the segment $g^{\uparrow}g^{\downarrow}$. Then $\|m - g^{\uparrow}\| = \|m - g^{\downarrow}\| \geq r_p \cdot \sin(\phi - \theta^{\flat})$ and $\|p - m\| \leq r_p \cdot \cos(\phi - \theta^{\flat})$.*

Proof. The first statement follows directly from the definition of ϕ as a lower bound on the angle $\angle gpm$. Observing that $g^{\uparrow}g^{\downarrow}$ is a chord of b_p, it is perpendicular to pm; see Figure 6.10 (right). By Proposition 6, we can write $\|p - m\| = r_p \cdot \cos(\angle gpm) = r_p\sqrt{1 - \sin^2(\angle gpm)} \leq r_p\sqrt{1 - \sin^2(\phi - \theta^{\flat})} = r_p \cdot \cos(\phi - \theta^{\flat})$. □

The point where $g^{\uparrow}g^{\downarrow}$ intersects the tangent plane T_p, denoted by x, is particularly useful in our proof. The next proposition bounds the distance from that point to the midpoint of the segment $g^{\uparrow}g^{\downarrow}$.

Proposition 7. *Consider any ball $b_p \in \mathcal{B}$ and a facet σ on a surface patch Σ such that $p \in \sigma$. Let $g^{\uparrow}$ and $g^{\downarrow}$ be a pair of potential seed locations on the boundary of b_p and let m be the midpoint of the segment $g^{\uparrow}g^{\downarrow}$. We further assume that $g^{\uparrow}g^{\downarrow}$ makes an angle at most $\theta^{\flat}$ with $v_{\sigma,p}$; see Figure 6.11. Let T_p denote the plane supporting σ and let x denote the point of intersection between T_p and the segment $g^{\uparrow}g^{\downarrow}$. Then, $\|m - x\| \leq r_p \cdot \cos(\phi - \theta^{\flat}) \cdot \tan(\theta^{\flat})$.*

Proof. Letting $m^{\perp}$ denote the projection of m on the plane T_p, we have that $\|m - m^{\perp}\| = \|p - m\| \cdot \sin(\theta^{\flat})$. By Corollary 5, we can write $\|p - m\| \leq r_p \cdot \cos(\phi - \theta^{\flat})$. Observing that $\|m - m^{\perp}\| = \|m - x\| \cdot \sin(\frac{\pi}{2} - \theta^{\flat}) = \|m - x\| \cdot \cos(\theta^{\flat})$, we get $\|m - x\| = \frac{\|m - m^{\perp}\|}{\cos(\theta^{\flat})} \leq \frac{\|p - m\| \cdot \sin(\theta^{\flat})}{\cos(\theta^{\flat})} \leq r_p \cdot \cos(\phi - \theta^{\flat}) \cdot \tan(\theta^{\flat})$. □

The main technical argument is encapsulated in the following lemma which bounds the shrinkage in terms of the angle $\theta^\flat$ defining the smoothness of the input mesh $\mathcal{T}$.

Lemma 36. *Consider any ball $b_p \in \mathcal{B}$ with p lying on a facet σ on the surface patch Σ, and a pair of potential seed locations $g^\uparrow$ and $g^\downarrow$ on the boundary of b_p. Let $b_q \in \mathcal{B}$, with $q \in \Sigma$, be an encroaching ball containing exactly one of the seed locations. Assume in addition that the segment $g^\uparrow g^\downarrow$ makes an angle at most $\theta^\flat$ with $v_{\sigma,p}$, and the segment pq makes an angle at least $\frac{\pi}{2} - \theta^\flat$ with $v_{\sigma,p}$. Then the shrinkage Δ applied to b_q to resolve the encroachment is bounded as*

$$\Delta = \max\big(\frac{\|q-g^\uparrow\|}{\|q-g^\downarrow\|}, \frac{\|q-g^\downarrow\|}{\|q-g^\uparrow\|}\big) - 1 < \zeta \cdot (1 + \frac{2\delta}{\lambda - \delta}) - 1, \tag{6.1}$$

where $\zeta \leq \sqrt{\dfrac{1+\sin(\theta^\flat)}{1-\sin(\theta^\flat)}}$, $\delta \leq \tan(\theta^\flat) \cdot \big(\frac{2}{1-L} + \cos(\phi - \theta^\flat)\big)$, *and* $\lambda \geq sin(\phi - \theta^\flat)$.

Proof. Since $b_p \cap b_q \neq \emptyset$, it follows that $\|p-q\| \leq r_p + r_q$. By the L-Lipschitzness condition, we have that $r_p \leq r_q + L \cdot \|p - q\|$. Substituting into the first inequality, we get that

$$\|p-q\| \leq r_p + (r_p + L \cdot \|p-q\|) \implies \|p-q\| \leq \frac{2}{1-L} \cdot r_p.$$

Let T_p denote the plane supporting σ and $q^\perp$ the projection of q onto T_p. By the assumption that pq makes an angle at least $\frac{\pi}{2} - \theta^\flat$, we have

$$\|q - q^\perp\| \leq \|p-q\| \cdot \sin(\theta^\flat) \leq \frac{2}{1-L} \cdot r_p \cdot \sin(\theta^\flat).$$

Letting $g \in \{g^\uparrow, g^\downarrow\}$ and $g^\perp$ denote the projection of g onto T_p, observe that $\|g - g^\perp\| \geq r_p \cdot \sin(\phi)$. Assuming $\theta^\flat$ is sufficiently small, we have that both the seed locations are farther from T_p than q.

Assume without loss of generality that $\|q - g^{\uparrow}\| \geq \|q - g^{\downarrow}\|$, and let H_q denote a plane parallel to T_p and passing through q. To simplify the analysis, we work instead with the point q' where the segment $g^{\uparrow}g^{\downarrow}$ intersects H_q; see Figure 6.11. Hence, we seek a bound on the ratio $\frac{\|q'-g^{\uparrow}\|}{\|q'-g^{\downarrow}\|}$. As we show later, we can use that to bound $\frac{\|q-g^{\uparrow}\|}{\|q-g^{\downarrow}\|}$ as desired, while suffering only a small multiplicative factor. We point out that while the points in question are not necessarily coplanar, it is easy to see that the worst-case is achieved when both seeds lie in a common plane with p and q.

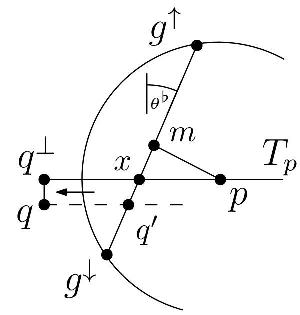


Figure 6.11: Setup for Lemma 36.

Letting x denote the intersection of T_p and $g^{\uparrow}g^{\downarrow}$, we start by bounding the distance between q' and x. Observing that both q' and $x \in g^{\uparrow}g^{\downarrow}$ while $q' \in H_q$, we get that $\|q - q^{\perp}\| = \|q' - x\| \cdot \cos(\theta^{\flat})$. It follows that

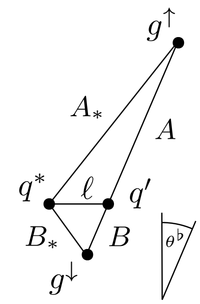
$$\|q' - x\| = \frac{\|q - q^{\perp}\|}{\cos(\theta^{\flat})} \leq \frac{2}{1-L} \cdot r_p \cdot \frac{\sin(\theta^{\flat})}{\cos(\theta^{\flat})} \leq \frac{2}{1-L} \cdot r_p \cdot \tan(\theta^{\flat}).$$

By Proposition 7, we get that

$$\|q' - m\| = \|q' - x\| + \|x - m\| \leq r_p \cdot \tan(\theta^{\flat}) \cdot \big(\frac{2}{1-L} + \cos(\phi - \theta^{\flat})\big). \qquad (6.2)$$

Letting $\lambda = \frac{\|m-g^{\uparrow}\|}{r_p}$, and $\delta = \frac{\|q'-m\|}{r_p}$, we can bound the ratio as $\frac{\|q'-g^{\uparrow}\|}{\|q'-g^{\downarrow}\|} = \frac{\lambda+\delta}{\lambda-\delta} = 1 + \frac{2\delta}{\lambda-\delta}$.

We need to account for using the proxy q' instead of the point realizing the actual worst-case; see the inset. Observe that the angle $\angle g^{\uparrow} q' q^* = \frac{\pi}{2} + \theta^\flat$ while $\angle g^{\downarrow} q' q^* = \frac{\pi}{2} - \theta^\flat$. Using the simplified notation in the figure, we apply the cosine rule to express the ratio realized by an arbitrary point q^* on H_q and at distance ℓ from q' as:



$$\frac{A_*^2}{B_*^2} = \frac{A^2 + \ell^2 + 2 \cdot A \cdot \ell \cdot \cos(\frac{\pi}{2} + \theta^\flat)}{B^2 + \ell^2 - 2 \cdot B \cdot \ell \cdot \cos(\frac{\pi}{2} - \theta^\flat)} = \frac{A^2 + \ell^2 + 2 \cdot A \cdot \ell \cdot \sin(\theta^\flat)}{B^2 + \ell^2 - 2 \cdot B \cdot \ell \cdot \sin(\theta^\flat)}.$$

For a fixed $\theta^\flat$, this ratio is maximized when $A = B = \ell$. Namely,

$$\frac{A_*^2}{B_*^2} \leq \frac{A^2 + A^2 + 2 \cdot A \cdot A \cdot \sin(\theta^\flat)}{A^2 + A^2 - 2 \cdot A \cdot A \cdot \sin(\theta^\flat)} = \frac{1 + \sin(\theta^\flat)}{1 - \sin(\theta^\flat)} \cdot \frac{A^2}{B^2}.$$

Hence, we apply the following correction, denoted ζ, when deriving the bound on $\theta^\flat$, using the ratio $\frac{A}{B}$.

$$\frac{\|q^* - g^{\uparrow}\|}{\|q^* - g^{\downarrow}\|} \leq \sqrt{\frac{1 + \sin(\theta^\flat)}{1 - \sin(\theta^\flat)}} \cdot \frac{\|q' - g^{\uparrow}\|}{\|q' - g^{\downarrow}\|} = \zeta \cdot \big(1 + \frac{2\delta}{\lambda - \delta}\big).$$

This completes the proof. $\square$

Lemma 36 confirms the intuition that the shrinkage ratio Δ decreases as the density of sampling increases, which in turn decreases the deviation of surface normals $\theta^\flat$. Figure 6.12 shows the bounds on shrinkage suggested by the lemma for the default value of $\alpha = 1 - \frac{\sqrt{3}}{2}$ and a range of values for L around the default value of $\frac{1}{4}$.

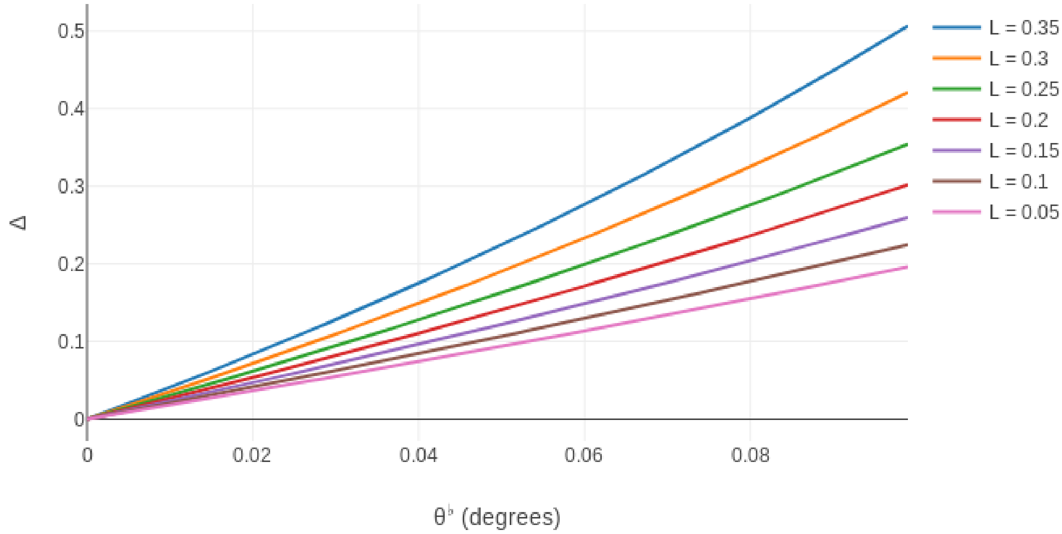


Figure 6.12: The shrinkage ratio Δ decreases as the angle $\theta^\flat$ decreases.

We are now ready to prove Theorem 8 by directly invoking Lemma 36. In particular, the theorem guarantees termination for a variant of the algorithm that leave the surface $\frac{\alpha}{2}$-deeply covered rather than α-deeply covered. Referring to Figure 6.12, we seek a specific bound to ensure that shrinking is sufficiently small to satisfy the relaxed deep coverage condition for termination.

Proof. The angle $\theta^\flat$ is chosen to ensure that $\frac{\|q^*-g^\uparrow\|}{\|q^*-g^\downarrow\|}$ is sufficiently small, i.e., less than $1+\frac{\alpha}{2-\alpha}$. The range of validity for $\theta^\flat$ is established by invoking the bound from Lemma 36 per Equation 36. Enforcing the desired bound, we get

$$\Delta < 1+\frac{\alpha}{2-\alpha} \implies \zeta\cdot\big(1+\frac{2\delta}{\lambda-\delta}\big) < 1+\frac{\alpha}{2-\alpha} \implies \frac{2\delta}{\lambda-\delta} < \frac{1}{\zeta}\big(1+\frac{\alpha}{2-\alpha}\big)-1$$
$$\implies \big(3-\frac{1}{\zeta}\big(1+\frac{\alpha}{2-\alpha}\big)\big)\cdot\delta+\big(1-\frac{1}{\zeta}\big(1+\frac{\alpha}{2-\alpha}\big)\big)\cdot\lambda<0.$$

As define in Lemma 36, we make the substitutions $\zeta \leq \sqrt{\frac{1+\sin(\theta^\flat)}{1-\sin(\theta^\flat)}}$, $\delta \leq r_p \cdot \tan(\theta^\flat)\cdot\big(\frac{2}{1-L}+\cos(\phi-\theta^\flat)\big)$ from Equation 6.2, and $\lambda \geq r_p\cdot\sin(\phi-\theta^\flat)$ from Corollary 5.

Simplifying, we obtain the following characterization of $\theta^\flat$ in terms of α and L, where $\phi = \sin^{-1}\left(\alpha \cdot \frac{1-(1-\alpha)\cdot L}{1+(1-\alpha)\cdot L}\right)$.

$$\begin{aligned}&\big(3 - \sqrt{\frac{1-\sin(\theta^\flat)}{1+\sin(\theta^\flat)}} \cdot \big(1 + \frac{\alpha}{2-\alpha})\big) \cdot \tan(\theta^\flat) \cdot \big(\frac{2}{1-L} + \cos(\phi - \theta^\flat)\big) \\ +&\big(1 - \sqrt{\frac{1-\sin(\theta^\flat)}{1+\sin(\theta^\flat)}} \cdot \big(1 + \frac{\alpha}{2-\alpha})\big) \cdot \sin(\phi - \theta^\flat) < 0. \end{aligned} \tag{6.3}$$

Setting $\theta^\flat = 0$ trivially satisfies the inequality, as the first term vanishes while the second term is negative. Hence, an upper bound may be determined by a simple bisection search over the interval $[0, \frac{\pi}{2}]$.

Using the default parameters $\alpha = 1 - \frac{\sqrt{3}}{2}$ and $L = \frac{1}{4}$ yields the upper bound $\theta^\flat < 0.049°$ per the following:

$$\begin{aligned}&\big(3 - 4 \cdot (2 - \sqrt{3}) \cdot \sqrt{\frac{1-\sin(\theta^\flat)}{1+\sin(\theta^\flat)}}\big) \cdot \tan(\theta^\flat) \cdot \big(\frac{8}{3} + \cos(4.95° - \theta^\flat)\big) \\ +&\big(1 - 4 \cdot (2 - \sqrt{3}) \cdot \sqrt{\frac{1-\sin(\theta^\flat)}{1+\sin(\theta^\flat)}}\big) \cdot \sin(4.95° - \theta^\flat) < 0, \end{aligned} \tag{6.4}$$

To gain more intuition about the general formula in Equation 6.3, we derive a simpler one with a strictly smaller upper bound on $\theta^\flat$ in terms of L, where α is fixed at $1 - \frac{\sqrt{3}}{2}$ and $\theta^\flat$ assumed to be sufficiently small. This can be achieved by making the $\tan(\theta^\flat)$ term larger and the $\sin(\phi - \theta^\flat)$ term, which is in fact negative, smaller in magnitude. First, the $\frac{1}{\varsigma}$ factor is very close to 1 for small values of $\theta^\flat$ and can be replaced by $\frac{1}{2}$ for the $\tan(\theta^\flat)$ term and a constant value very close to one for the $\sin(\phi - \theta^\flat)$ term. Similarly, we replace $\cos(\phi - \theta^\flat)$ by 1. Finally, we replace $\sin(\phi - \theta^\flat)$ with $\frac{1-L}{20}$. By relaxing the coefficients, we obtain the simplified formula:

$$\tan(\theta^\flat) < \frac{1}{1000}(1 - L)^2, \tag{6.5}$$

which for $L = \frac{1}{4}$ implies $\theta^\flat < 0.032°$. Figure 6.13 shows the degradation incurred by the simplification. $\square$

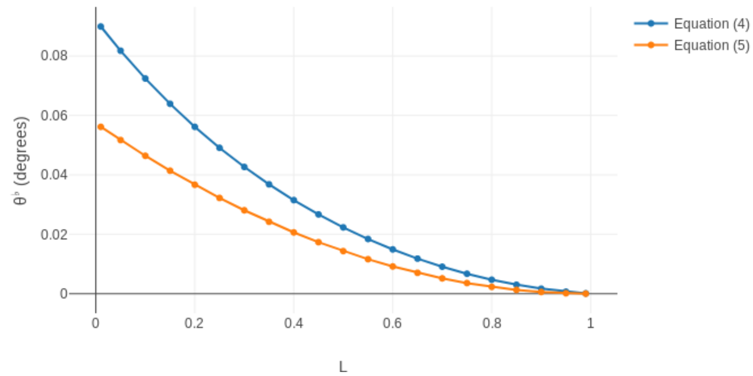


Figure 6.13: Equation 6.5 simplifies the general bound in Equation 6.3 for the default value of $\alpha = 1 - \frac{\sqrt{3}}{2}$.

In Figure 6.14, we provide additional values to justify fixing the value of α as a design parameter, and to further validate the utility of the formula derived in the proof of Lemma 36. The upper-bounds corresponding to the relevant range of parameter settings are summarized in the figure below with $L \in [0.05, 0.95]$.

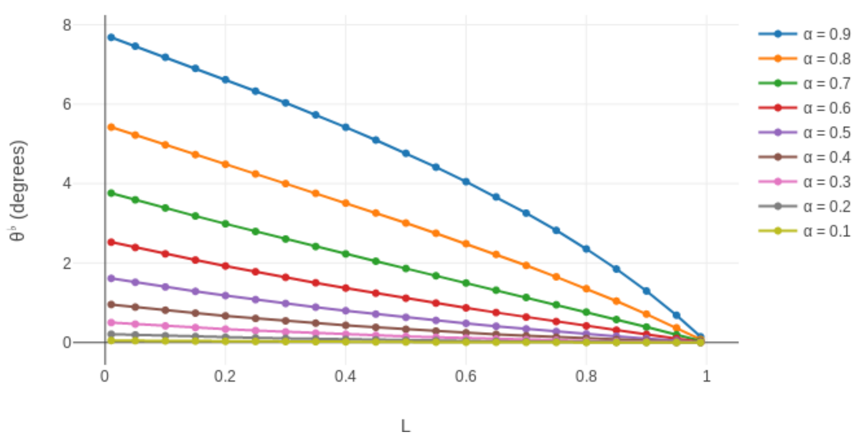


Figure 6.14: Upper-bounds on $\theta^\flat$ for different values of α and L per Equation 6.3.

In conclusion, ensuring the input surface mesh $\mathcal{T}$ is sufficiently smooth, with respect to the chosen parameters α and L, implies a suitable bound on the shrinkage ratio Δ to guarantee the termination of the algorithm. The smoothness of the input mesh is defined in terms of the dihedral angles subtended by adjacent facets away from the sharp features per the parameter $\theta^\flat$. As the derived formula exhibits no singularities for $L < 1$, the bound degrades smoothly as shown in Figure 6.14.

To further validate our claim, within machine precision, we use $\alpha = 0.05$ to obtain a strictly positive lower envelop for all settings of the input parameter L defining the L-Lipschitzness condition, as well as all relevant settings of the design parameter α for deep coverage; see Figure 6.15 where we used $\log_{10}$ scale to better distinguish small positive values. This guarantees the termination the algorithm regardless of the parameters used, assuming the surface is sufficiently smooth.

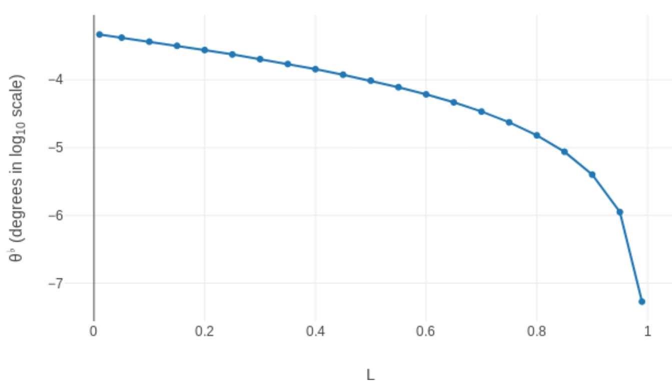


Figure 6.15: Setting $\alpha = 0.05$ still yields a strictly positive upper-bound on $\theta^\flat$ satisfying Equation 6.3.

Finally, we point out that to enable the derivations above, various inequalities had to be relaxed such that they no longer correspond to any situation that may be encountered by the algorithm. Hence, the derived bounds on $\theta^\flat$ are rather conservative and only serve to establish the existence of strictly positive upper bounds.

6.2.9 Practical Sliver Elimination

Our implementation always reruns RMPS to recover α-deep coverage. We argue that this variant terminates with high probability by combining the bounds on shrinkage with the stability of deep coverage as a distribution. In our experiments, VoroCrust always terminates with all slivers eliminated successfully while avoiding excessive refinement; see Section 3. In the unlikely event that sliver elimination fails to terminate in a constant number of iterations, set to 100, we restart in a *safe mode* accepting $\frac{\alpha}{2}$-deep coverage to guarantee termination; we never encountered such cases.

Decaying Shrinkage and Violations. Subsequent invocations of RMPS in the course of sliver elimination increase the density of sampling. A consequence of the ball conditions maintained by RMPS is that the radii of overlapping balls get smaller. In particular, the deviation in normals at the centers of overlapping balls gets smaller, which is equivalent to enforcing the smooth overlap condition C2 with a smaller angle threshold. Intuitively, the neighborhood of each sample becomes *nearly flat*. This flatness increases the ratio $\frac{\|q-g^\uparrow\|}{\|q-g^\downarrow\|}$ for all nearby samples q, which reduces the shrinkage ratio Δ and restricts the potential locations of new samples that create new slivers. It follows that the percentage of triplets with half-covered seed pairs

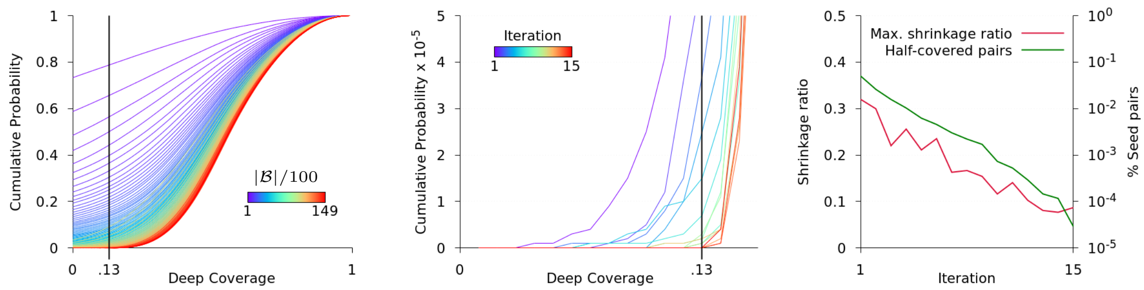


Figure 6.16: Empirical analysis of sliver elimination using the Bimba model: (left) evolution of the deep-coverage distribution through the first invocation of RMPS as $\mathcal{B}$ grows in increments of 100 balls, (middle) sliver elimination executes 15 iterations where shrinking eventually ceases to violate α-deep coverage, (right) the refinement incurred by sliver elimination decreases the maximum shrinkage ratio applied in subsequent iterations. As a result, the number of newly created slivers, measured by the percentage of triplets with half-covered seed pairs, decays rapidly.

decays rapidly; see Figure 6.16(right).

Deep-coverage Distribution. Let f_i be a function that maps each $x \in \mathcal{T}$ to $\max\{1 - \frac{\|x-p\|}{r_p} \mid b_p \in \mathcal{B}_{i,x}\}$ where $\mathcal{B}_{i,x}$ is the subset of balls containing x at iteration i. We use the family of functions $\{f_i\}$ to define the deep-coverage distribution as $F_i(\alpha) = \Pr[f_i(x) \leq \alpha \mid x \in \mathcal{T}]$ with $\alpha \in [0, 1]$. We estimate F_i by the empirical distribution function over 100 bins using independent random samples of 10^6 points. Figure 6.16(left) shows the evolution of the deep-coverage distribution through the first invocation of RMPS until convergence. Every subsequent invocation of RMPS, following shrinking for sliver elimination, converges to a nearly identical distribution. Related aspects of the distributions of MPS samplings were analyzed [177], which are consistent with our experiments[1]. As seen in Figure 6.16(middle), shrinking for sliver elimination initially violates α-deep coverage, per C4 requiring a fixed $\alpha \approx 0.13$, but causes no such violations over the last few iterations. The combination of decaying shrinkage and the stability of deep coverage as a distribution bounds the probability of such violations. It follows that subsequent invocations of RMPS are less likely to introduce new balls to recover α-deep coverage. As a result, the number of newly created slivers per iteration decays rapidly; see Figure 6.16(right). Hence, the total number of slivers encountered by the algorithm is bounded in expectation, which implies termination in a finite number of steps with high probability.

6.2.10 Volume Meshing

Once the refinement process terminates, the set of balls $\mathcal{B}$ is fixed and a conforming surface mesh can be generated. To further decompose the interior into a set of graded Voronoi cells, additional weighted samples $\mathcal{S}^{\downarrow\downarrow}$ are generated in the interior of the domain. Similar to $\mathcal{B}$, the balls corresponding to interior samples are required to satisfy the L-Lipschitzness condition. Standard MPS may be used for sampling the interior. However, to reduce the memory footprint of this step, the spoke-darts algorithm [179] is used instead following a lightweight initialization phase using standard dart-throwing; see the appendix for more details. Alternatively, the interior samples may be chosen as the vertices of a structured lattice. This can be used to output a hex-dominant mesh conforming to the surface; see Figure 6.5(f). The

[1] The total variation distance [178] between the empirical distributions obtained through all subsequent iterations is at most 0.02.

quality of the volume mesh can be further improved by applying CVT optimization to the set of interior seeds; see Figure 6.5(d).

6.2.11 Meshing 2D Domains

The proposed VoroCrust algorithm can readily be applied to the decomposition of 2D domains into conforming Voronoi meshes. As illustrated in Figure 6.4, the seed placement strategy can be applied in 2D given a suitable union of balls. The refinement strategy described in this section can easily be applied to generate such a union of balls by regarding the 2D boundary as a set of creases embedded in 3D. In particular, assuming the 2D boundary is available as a set of line segments or a planar straight-line graph (PSLG) as common in 2D meshing, the input segments can be mapped to 3D by adding a third coordinate, e.g., $z = 0$, to all end points. The ball conditions and refinement process for the protection of sharp features, as defined in Section 6.2.3, guarantee a union of balls that approximates the embedded 2D boundary.

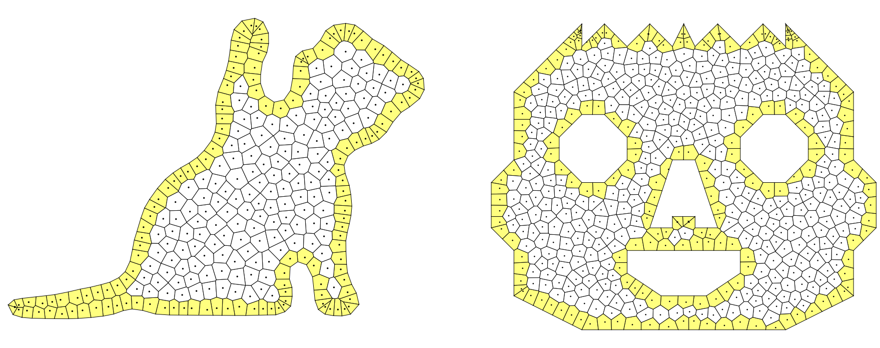


Figure 6.17: The VoroCrust algorithm readily handles 2D domains.

Such a union of balls can be used to place Voronoi seeds in 2D as follows. First, all balls are projected onto the 2D plane as circles centered along the boundary. Then, the pairs of intersection points between consecutive circles are computed. Recalling that the edge balls protecting any given crease may only overlap consecutive balls along the same crease, these pairs of intersection points are well-defined. Once the intersection pairs are obtained, the algorithm places Voronoi seeds across the 2D boundary and proceeds to sample additional seeds to mesh the 2D interior. Figure 6.17 shows a number of conforming 2D Voronoi meshes, with uniform sizing in the interior, obtained by a 2D implementation of VoroCrust.

6.3 Implementation Details

This section provides additional details to better explain some of the subroutines we use in our prototype implementation of the VoroCrust algorithm. We start by describing the speed-ups for proximity queries against the input PLC $\mathcal{T}$ and the set of balls $\mathcal{B}$. Then, we describe the generation of interior samples. Finally, we instrument the code to detect performance bottlenecks and help improve the algorithm in future iterations.

6.3.1 Supersampling the Boundary

The algorithm constructs one k-d tree for each type of strata to speed up proximity queries against $\mathcal{T}$. The k-d tree indexing the sharp corners is simply populated using the set of sharp corners. In order to populate the k-d tree indexing the creases, the algorithm generates a set of 10^5 points sampled uniformly at random from all sharp edges. Similarly, the k-d tree indexing the surface patches is populated using a set of 10^6 points sampled uniformly from all facets. Each generated sample q stores a vector $v_{\sigma,q}$ for each edge or facet $\sigma \ni q$.

6.3.2 Querying the Boundary k-d trees

Given a point p on a face σ, the algorithm estimates the distance to the nearest non-co-smooth point on the input mesh $\mathcal{T}$ by querying the three boundary k-d trees indexing the sharp corners, creases and surface patches. Let K denote any of the boundary k-d trees. As the query aims to determine the nearest non-co-smooth point, the co-smoothness test described in Section 2.3 can be used to filter the set of points indexed by K. We implemented a custom k-d tree that performs this filtration on-the-fly. As in the standard k-d tree, the query maintains an estimate of the distance to the nearest point which can be initialized to any sufficiently large value, e.g., the diameter of $\mathcal{T}$ or ∞. By comparing the current estimate against the distance from p to the splitting plane associated with the current node, the query discards an entire subtree if it cannot improve the estimate. The only difference is that due to the filtration defined by the co-smoothness test, a node associated with a point which is co-smooth with p does not provide a distance to update the estimate.

6.3.3 Ball Neighborhood

To find the set of balls overlapping a given ball b_p, a naive search would be costly. Instead, we find an upper bound on the distance between p and any sample q

such that b_q may overlap b_p. Then, we use this bound to query the k-d trees.

Consider two overlapping balls b_p and b_q generated by the MPS procedure, with radii r_p and r_q. W.l.o.g., assume $r_q \geq r_p > 0$. The L-Lipschitzness condition implies that $r_q \leq r_p + L \cdot \|p - q\|$. Since the two ball overlap: $\|p - q\| < r_p + r_q$. Combining the two inequalities, it follows that: $\|p - q\| < r_p + r_q + L \cdot \|p - q\|$. We conclude that $\|p - q\| \leq \frac{2}{1-L} \cdot r_p$. Hence, we query the k-d trees for all balls whose centers are within that distance from p and check if they overlap b_p.

6.3.4 Point Neighborhood

The deep coverage condition is checked for each new sample p. To speed up this check, we derive an upper bound on the distance between p and the center of any ball that may cover it, and use this to query the k-d trees.

Let q denote the center of the closest ball to p, which we find by a standard nearest-neighbor query to the k-d tree in question. The radius of a ball placed at p respecting L-Lipschitzness can be estimated as $r_p \leq r_q + L \cdot \|p - q\|$.

Consider a ball b_s that barely covers p. It follows that $r_s \leq r_p + L \cdot \|p - s\|$, where $\|p - s\| \leq r_s$. Combining the two inequalities, it follows that $r_s \leq r_q + L \cdot \|p - q\| + L \cdot r_s$, implying $r_s \leq \frac{r_q + L \cdot \|p - q\|}{1-L}$. Hence, we query the k-d tree for all balls whose centers are within that distance from p and check if they contain p.

6.3.5 Sampling the interior

The algorithm starts by computing a bounding box bb enclosing the input mesh $\mathcal{T}$; we expand bb to the box $3\times$ larger with the same center. This box is used to initialize the set of interior seeds $\mathcal{S}^{\downarrow\downarrow}$ using a lightweight dart-throwing phase. Additional samples are added as needed using the more efficient spoke-darts algorithm [179]. To guide interior sampling, and ensure a sufficient distance between interior seeds and surface seeds, each surface seed $s \in \mathcal{S}^{\updownarrow}$ is assigned a radius r_s by averaging the radii of the three balls in $\mathcal{B}$ defining it. As was done for the set of surface balls $\mathcal{B}$, we maintain two k-d trees $K^{\updownarrow}$ and $K^{\downarrow\downarrow}$ for all balls centered at seeds in $\mathcal{S}^{\updownarrow}$ or $\mathcal{S}^{\downarrow\downarrow}$, respectively.

To initialize $\mathcal{S}^{\downarrow\downarrow}$, a new sample z is generated uniformly at random from bb. Then, the closest seed $s \in \mathcal{S}^{\updownarrow}$ to z is found by a nearest-neighbor query to $K^{\updownarrow}$. If $\|z - s\| < r_s$, z is rejected. Otherwise, z gets the label of s and a radius $r_z = r_s + L \cdot \|z - s\|$, which extends the estimated sizing function to the interior of the domain [149]. Similarly, the closest interior seed $z^* \in \mathcal{S}^{\downarrow\downarrow}$ to z is found by querying $K^{\downarrow\downarrow}$ and z is rejected if $\|z - z^*\| < r_{z^*}$. Whenever a new sample is rejected, we increment a *miss counter* and otherwise reset it back to 0 if the sample was successfully added into $\mathcal{S}^{\downarrow\downarrow}$. Initialization terminates when the miss counter reaches 100.

Then, we continue to add seeds into $\mathcal{S}^{\downarrow\downarrow}$ using the spoke-darts algorithm [179] as follows. We populate a queue Q with all seeds generated by dart-throwing. While the queue is not empty, we pop the next sample z and do the following. Letting b_z

be the ball centered at z with radius r_z, we choose a random direction δ and shoot a *spoke* (ray) starting at z in that direction to obtain a new point z_δ at distance $2 \cdot r_z$ from z. Then, we query the k-d trees to find all balls potentially containing z_δ. For each such ball, we trim the line segment ℓ_δ between z and z_δ by pushing z_δ to lie on the boundary of that ball. Once we are done, if z_δ was pushed all the way into the ball b_z, we increment the miss counter. Otherwise, we sample a point z^+ uniformly at random on ℓ_δ, add it as a seed, and reset the miss counter to 0. As before, z^+ is assigned a label and a radius before pushing it into Q. When the miss counter reaches 100, we discard the current point and pop a new point from Q. This process terminates when Q is empty. Finally, we enforce L-Lipschitzness on all interior samples, shrinking balls as necessary, before repopulating Q with all seeds and repeating until no ball gets shrunk.

6.3.6 Code Profiling and Bottlenecks

We instrument our code to collect more detailed timing statistics for the main procedures of the algorithm; see Section 6.2. As would be expected, the most time consuming component of the algorithm is surface coverage, with related MPS iterations as described under "Protection and Coverage," and to a lesser extent volume sampling per Section 6.3.5; other procedures including preprocessing, sharp feature protection, and sliver elimination are not as demanding. In particular, each surface sample requires a sizing estimate by querying the boundary k-d trees which store a dense sampling of surface elements; see Section 6.3.1. In addition, whenever we shrink a surface ball, checking for uncovered surface patches requires restarting the surface MPS procedure. For example, Table 6.1 summarizes the running time on two sample models.

Procedure	Smooth	Sharp Features
Corner protection	0	0.213
Edge protection	0	4.157
Surface coverage	671.165	180.986
Fixing C3 violations	17.255	2.962
Sliver elimination	14.127	3.216
Interior sampling	13.981	36.395

Table 6.1: Timing breakdown for the smooth model shown in Figure 6.1 and the model with sharp features shown in Figure 6.2.

Per the table above, C3 violations and sliver elimination incur higher overhead for the smooth model with higher surface curvature compared to the model with sharp features and otherwise flat regions.

6.4 Evaluation

We demonstrate the capabilities of the VoroCrust algorithm and study the impact of input parameters. Then, we compare against the work of Yan et al. [66] as a representative of state-of-the-art clipping-based methods. All experiments were conducted on a Mac Pro machine with a 3.5 GHz 6-Core Intel Xeon E5 processor and 32 GB of RAM.

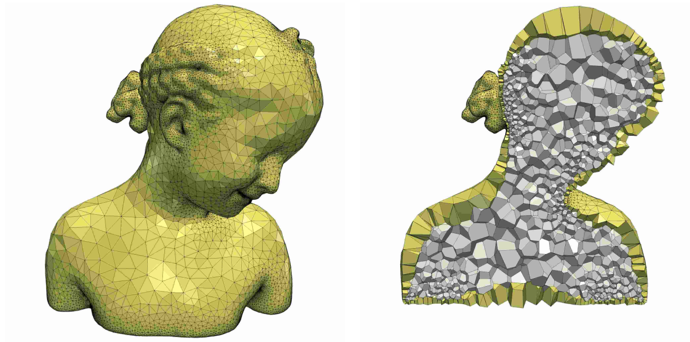
6.4.1 Sample Results

We test VoroCrust on a variety of models exhibiting different challenges ranging from smooth models with detailed features and narrow regions as in Figure 6.18, to sharp features with curvature and holes as in Figure 6.19, and even non-manifold boundaries as in Figure 6.20. The quality of the surface mesh is measured by the percentage of triangles with angles less than 30° or greater than 90°, as well as the minimum triangle quality[2] Q_{min}. The quality of the volume mesh is measured by the maximum aspect ratio[3] ρ_{max}, which is often realized by cells incident to the surface. We also report the approximation error in terms of the Hausdorff error d_H (normalized by the diameter of the bounding box). The number of seeds in $\mathcal{S}^{\updownarrow}$ and $\mathcal{S}^{\downarrow\downarrow}$ are reported along with the time in seconds taken to generate each, denoted $T^{\updownarrow}$ and $T^{\downarrow\downarrow}$, respectively. Meshes were generated from VoroCrust seeds using Voro++ [180].

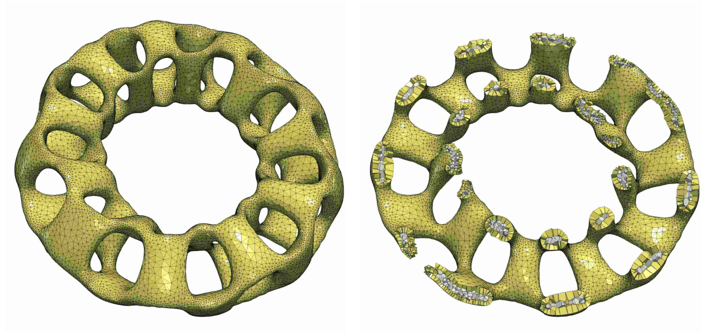
Non-manifold models are particularly important in physical simulations with multiple materials of different properties. VoroCrust detects non-manifold features in the input mesh, as described in Section 6.2.2, and the ball conditions described in

[2]Triangle quality is defined as $\frac{6S}{\sqrt{3}hP}$, where S is the area, h is the longest edge length, and P is half the perimeter.

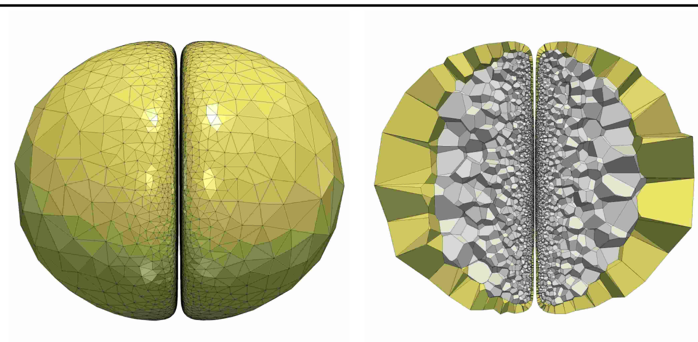
[3]Aspect ratio is defined as the ratio between the radius of the smallest circumscribing sphere to the radius of the largest inscribed sphere.



$\theta_{<30}\%$	$\theta_{>90}\%$	Q_{min}	ρ_{max}	$d_H(\times 10^{-2})$	$\mathcal{S}^{\updownarrow}$	$\mathcal{S}^{\downdownarrows}$	$T^{\updownarrow}$	$T^{\downdownarrows}$
2	16	0.373	5.345	0.614	68472	17035	935	587

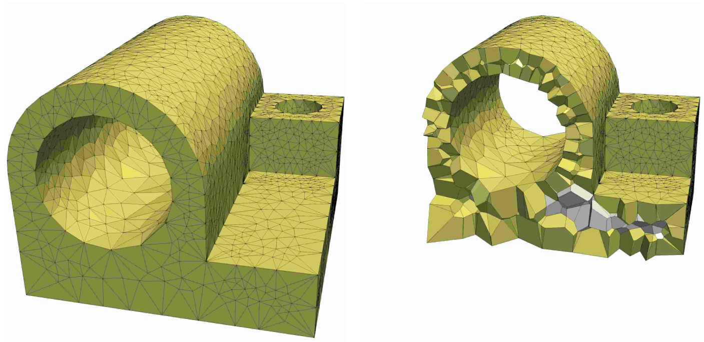


$\theta_{<30}\%$	$\theta_{>90}\%$	Q_{min}	ρ_{max}	$d_H(\times 10^{-2})$	$\mathcal{S}^{\updownarrow}$	$\mathcal{S}^{\downdownarrows}$	$T^{\updownarrow}$	$T^{\downdownarrows}$
2	16	0.383	5.407	0.171	114472	6726	1581	1363

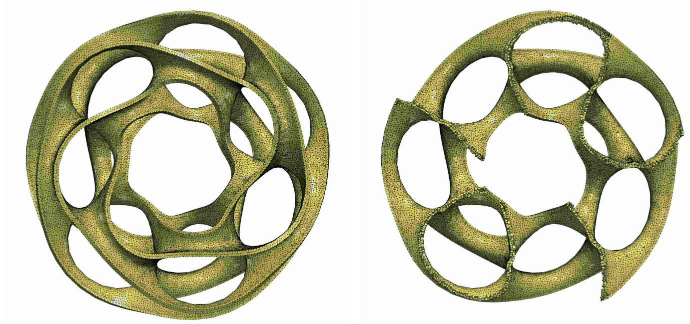


$\theta_{<30}\%$	$\theta_{>90}\%$	Q_{min}	ρ_{max}	$d_H(\times 10^{-2})$	$\mathcal{S}^{\updownarrow}$	$\mathcal{S}^{\downdownarrows}$	$T^{\updownarrow}$	$T^{\downdownarrows}$
0.07	15	0.4	4.863	0.851	497536	113837	4582	7007

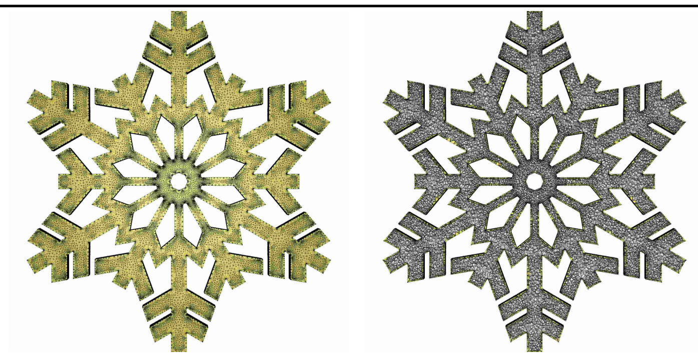
Figure 6.18: Sample results on smooth models.



$\theta_{<30}\%$	$\theta_{>90}\%$	Q_{min}	ρ_{max}	$d_H(\times 10^{-2})$	$\mathcal{S}^{\updownarrow}$	$\mathcal{S}^{\downdownarrows}$	$T^{\updownarrow}$	$T^{\downdownarrows}$
19	23	0.149	12.495	0.569	11480	868	32	34



$\theta_{<30}\%$	$\theta_{>90}\%$	Q_{min}	ρ_{max}	$d_H(\times 10^{-2})$	$\mathcal{S}^{\updownarrow}$	$\mathcal{S}^{\downdownarrows}$	$T^{\updownarrow}$	$T^{\downdownarrows}$
11	19	0.273	377.029	0.087	258010	0	1464	3432



$\theta_{<30}\%$	$\theta_{>90}\%$	Q_{min}	ρ_{max}	$d_H(\times 10^{-2})$	$\mathcal{S}^{\updownarrow}$	$\mathcal{S}^{\downdownarrows}$	$T^{\updownarrow}$	$T^{\downdownarrows}$
21	25	0.086	63	0.058	85380	57474	2146	9497

Figure 6.19: Sample results on models with sharp features.

Section 6.2.3 guide the refinement to protect those features, ensuring their correct recovery in the output mesh. Figure 6.20 shows VoroCrust output for a collection of non-manifold models. In addition, Figure 6.21 shows VoroCrust output for a complex mechanical model.

We encountered no issues with any of the models, which demonstrates the robustness of the algorithm and its implementation. We set $\theta^\sharp$ to $60°$ for smooth models, and choose an appropriate value of $\theta^\sharp$ for models with sharp features. The value of L was fixed at 0.25 for all inputs. We note that the output surface meshes are of high quality per the minimum triangle quality and angle bounds, while achieving small approximation errors. The demonstrated quality of VoroCrust output, with no skinny elements, is in agreement with the theoretical guarantees established in Chapter 5.

6.4.2 Parameter Tuning

We start by studying the impact of L on the complexity of the output surface mesh and the running time of the algorithm. Figure 6.22 demonstrates this impact on the Joint model. The results of this experiment demonstrate the impact of L on the level of refinement per the number of balls in $\mathcal{B}$ generated by the algorithm. In particular, smaller values of L lead to higher refinement. On the other hand, larger values of L slow down the algorithm due to the increased size of ball neighborhoods resulting in processing a larger number of balls for various tasks; see Section 6.3.3. This behavior of the algorithm in terms of L is consistent for different values of $\theta^\sharp$ as

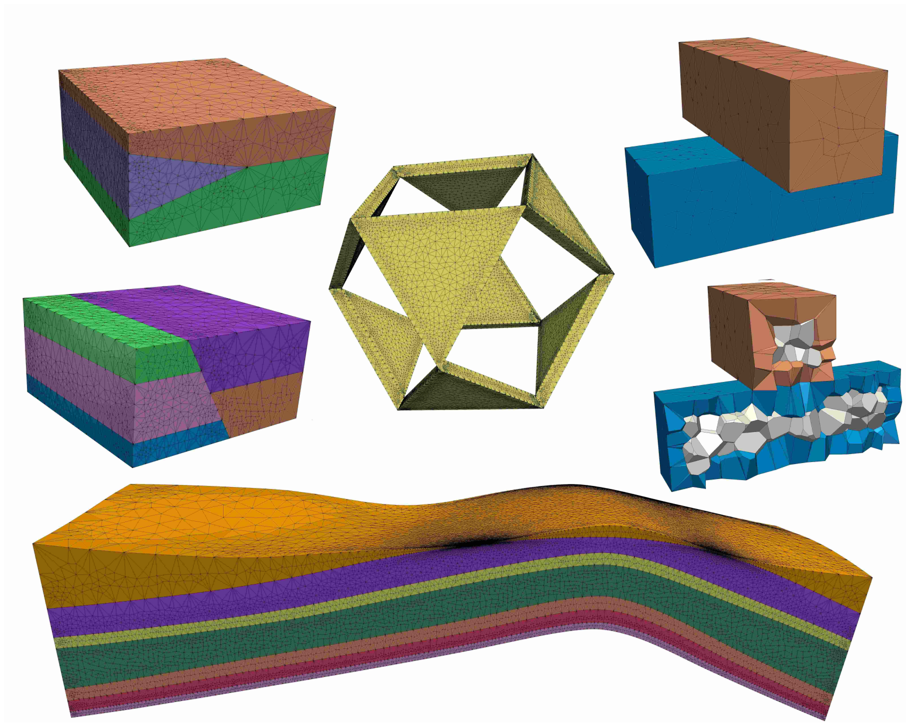


Figure 6.20: Sample outputs for non-manifold domains consisting of multiple materials depicted in different colors. VoroCrust automatically detects the non-manifold interfaces between the materials (top left) and decomposes each subdomain into Voronoi cells that conform to those interfaces while preserving all sharp features (top right). More challenging cases involve contact at sharp features (top center), or multiple layers tapering into narrow regions towards contact (bottom).

can be seen in Figure 6.22.

Next, we study the impact of varying both L and $\theta^\sharp$. We chose a relatively simple smooth model to better assess the degradation in surface approximation. Figure 6.24 illustrates VoroCrust output on the Goat model for 5×5 combinations of parameter settings. As shown earlier, smaller values of L result in more regular meshes with superior element quality per the minimum triangle angle. On the other hand, the parameter $\theta^\sharp$ controls the surface approximation. Namely, higher values of $\theta^\sharp$ result in higher Hausdorff errors.

Finally, we study the impact of the input sizing field sz on the multi-layered nested spheres models. Figure 6.23 shows how sz can be used to directly control ball radii to enforce further refinement. The default setting of $sz = \infty$ incurs the minimum level of refinement required by the geometry of the domain according to the quality requirements indicated by the parameters L and $\theta^\sharp$. We note that sz can be specified as a spatially varying sizing field.

In summary, this study demonstrates the flexibility of the VoroCrust algorithm to accommodate a wide range of parameter settings that cater to the requirements of different applications. In particular, the set of parameters provided allows the user to trade-off the quality of the surface mesh, approximation error, output complexity, and running time.

6.4.3 Comparison

We compare against the restricted Voronoi diagram (RVD) [66] as a representative of state-of-the-art polyhedral meshing algorithms based on clipped Voronoi cells. While RVD is typically used within CVT-based algorithms to speed up energy calculations, we are only interested in its robust clipping capabilities which provide a suitable baseline for comparison. For all models, we use the interior VoroCrust seeds $\mathcal{S}^{\downarrow\downarrow}$ as input to RVD clipping. As shown in Figure 6.1, VoroCrust achieves superior quality in terms of the surface mesh, where RVD clipping produces an imprint of the input mesh with many small facets. In particular, by examining the ratio of the shortest to longest edge length per surface facet, it is clear that RVD clipping results in many skinny facets which can be problematic for many applications. Moreover, RVD clipping possibly results in non-convex cells for non-convex models, e.g., Figure 6.2. In our experiments, the ratio of non-convex cells in RVD output varies between 3% and 96%, depending on the curvature of the input surface and the chosen set of Voronoi seeds. In contrast, VoroCrust output conforms to the boundary with true Voronoi cells, which are guaranteed to be convex, while achieving much better quality of surface elements. We note that clipping the Voronoi cells of a given set of seeds can be performed much faster, as in the parallel RVD implementation of [66], compared to the multiple iterations and non-trivial steps of VoroCrust refinement; see Section 6.2.

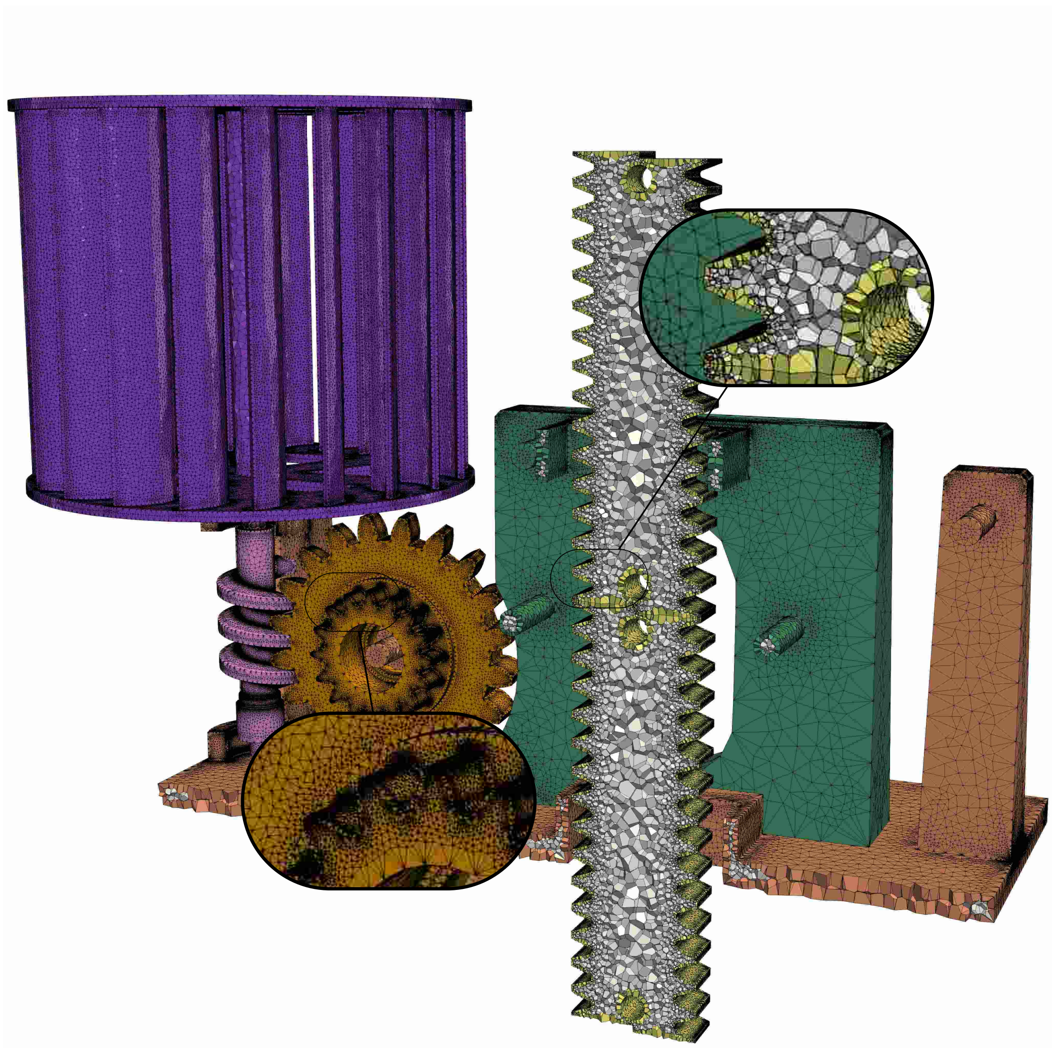


Figure 6.21: VoroCrust output for complex mechanical parts sharing non-manifold contact interfaces with detailed sharp features.

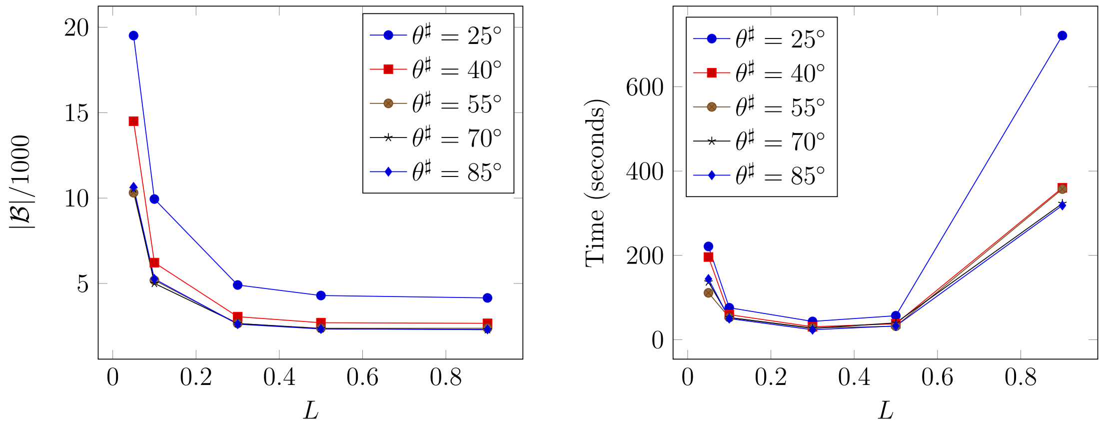


Figure 6.22: Impact of the parameter L on the Joint model for varying values of $\theta^\sharp$. While the level of refinement is inversely proportional to L, increasing L slows down the algorithm due to larger ball neighborhoods.

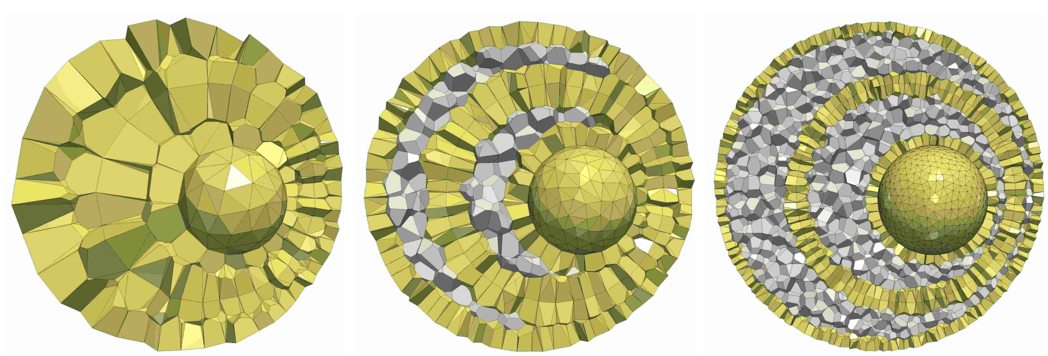


Figure 6.23: Impact of the sizing field parameter sz on the nested spheres model. From left to right: $sz = \infty$ (default), $sz = 1$, and $sz = 0.5$.

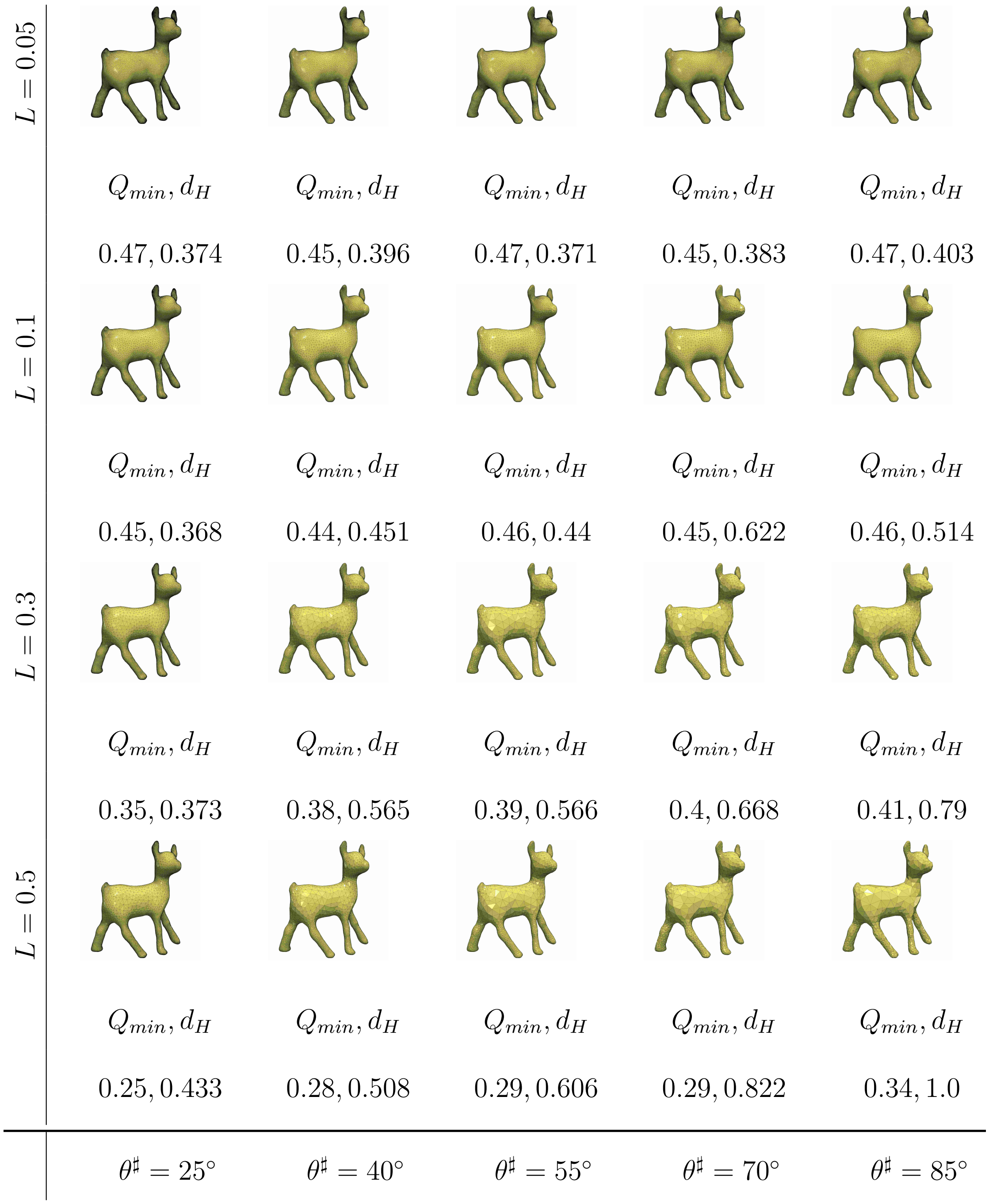


Figure 6.24: Impact of input parameters on surface quality and approximation error.

Chapter 7: Conclusions and Future Directions

In this dissertation, we applied a sampling methodology to a number of fundamental problems in computational geometry. Our work emphasizes the potential benefits of a sampling approach that adapts to both the shape or distribution of the data, as in Chapter 3, as well as the functions defined on this data, as in Chapter 4, for the design of approximation algorithms and data structures. While this sampling approach is heavily inspired by related sampling techniques in geometry processing, we also demonstrate the benefits of applying advanced techniques from algorithm theory to the design and analysis of new algorithms in geometry processing, as in Chapter 6.

The work presented here opens many potential directions for future research to further develop the different aspects of our algorithmic sampling methodology. In the sections below, we outline ongoing work and a number of follow-up questions to the work we did on each problem.

7.1 Polytope Approximation

In Chapter 3, we demonstrated a simplified application of Macbeath regions for convex approximations by appealing to the intrinsic Hilbert metric. One important consideration that we did not satisfactorily address is the efficient construction of the proposed data structure, or the practical implementation of the such constructions. While the boostrapping algorithm presented in [50] makes some progress in this direction, it is not particularly well-suited for implementation.

The Delone set formulation encourages the investigation of practical construction algorithms based on sampling techniques similar to those from geometry processing, e.g., Poisson-disk sampling [174–176]. This may be combined with recent developments in convex optimization to implement the lower-level steps. In particular, the explicit computation of Macbeath regions can be avoided by directly computing their John ellipsoids using the algorithm in [181]. Then, the generation of random samples may benefit from efficient random walks from sampling in polytopes as in [182].

7.2 Nearest-Neighbor Searching

In Chapter 4, we developed generalized data structures for nearest-neighbor searching under non-Euclidean distances. An essential ingredient to the efficiency of the proposed data structures is to retain the reduction to *approximate ray-shooting queries* against a convex envelope of distance minimizers. As this reduction previously relied on the *lifting transform*, its application was limited to the Euclidean distance. By applying *convexification*, we circumvent reliance on the lifting transform.

As we have seen in Chapter 4, the efficient implementation of approximate ray-shooting relies on the approximation of derived polytope that arise from the envelopes of distance functions at local neighborhoods. Those polytopes in turn are approximated using Macbeath regions, similar to the work presented in Chapter 3.

In ongoing work, we avoid the reduction to ray-shooting queries by defining the Macbeath regions directly in the original space without any lifting. We achieve that by extending the Delone set criteria to derive a succinct approximation of the Voronoi diagram using a hierarchy of ellipsoids. The proposed approach works for Bregman divergences with well-behaved generators, and allows space-time trade-offs similar to what the AVD data structure offers Euclidean nearest-neighbor search [34, 50], as stated in Theorem 2.

7.3 Distance Approximation

By further elaborating on the proposed ellipsoidal covers for nearest-neighbor searching, we consider the approximation of the distance function itself rather than searching for an approximate nearest-neighbor.

Observe that the ellipsoids approximating the Voronoi diagram cover the entire space using primitive elements which are sensitive to the distance functions. The resulting cover bears similarity to the anisotropic meshes studied in approximation theory. In ongoing work, we use the ellipsoidal cover to propose the *first* continuous approximation of the distance function to the set of points. The approximation can be evaluated in the same asymptotic time of standard nearest-neighbor search queries, and exhibits bounded gradients whose magnitudes are proportional to the reciprocal of the approximation parameter ε.

Another application of the distance-sensitive ellipsoidal cover is to approximate the *level sets* of the distance functions to a set of n points, as used in topological data analysis. In particular, the recent work of Choudhary et al. [183] uses adaptive grids, or *pixels*, to approximate the level sets to derive a *sparse filtration* of size $O(n/\varepsilon^d)$. It is plausible to expect the distance-sensitive ellipsoidal cover to enable a a filtration of size only $O(n/\varepsilon^{d/2})$, similar to the recent improvements in the storage requirements of nearest-neighbor search data structures.

7.4 Voronoi Meshing

The VoroCrust algorithm described in Chapter 6 has been successfully implemented and verified over numerous challenging inputs. However, there are a number of drawbacks and feature requests that require further research.

The main limitation of the presented algorithm is the possible presence of short Voronoi edges in the interior of the output mesh, which can lead to small time steps in numerical simulations significantly increasing their cost. To eliminate such short edges, mesh improvement techniques may be applied as postprocessing [184, 185].

Another limitation is the requirement that the input triangulation is a faithful approximation of the domain. This inhibits the application of this approach to implicit forms [186], noisy inputs [187], or unclean geometries [188]. In particular, the algorithm does not fill holes or undesirable cracks in non-watertight inputs [189]. Nonetheless, VoroCrust readily handles surfaces with boundary as shown in Figure 7.1.

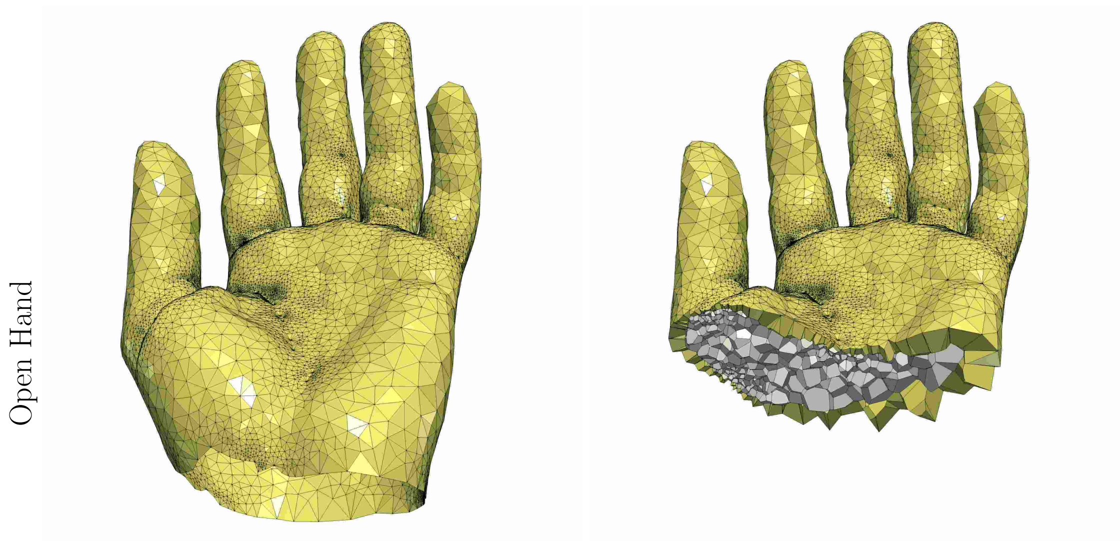


Figure 7.1: VoroCrust can handle surfaces with boundary. Volume samples within a suitable bounding box can be filtered, e.g., manually, as shown.

Finally, the isotropic nature of the proposed sampling process may result in an unnecessarily large number of cells in narrow regions. For such geometries, boundary layers of elongated cells enable higher fidelity near the boundary [190, 191]. In cases of strong anisotropy, aligning the cells, e.g., to the eigenvectors of a Hessian [68, 192], better captures the variation of physical quantities.

Bibliography

[1] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14(3):432–448, 1993.

[2] J. Matoušek and O. Schwarzkopf. On ray shooting in convex polytopes. *Discrete Comput. Geom.*, 10:215–232, 1993.

[3] T. M. Chan. Fixed-dimensional linear programming queries made easy. In *Proc. 12th Annu. Sympos. Comput. Geom.*, pages 284–290, 1996.

[4] E. A. Ramos. Linear programming queries revisited. In *Proc. 16th Annu. Sympos. Comput. Geom.*, pages 176–181, 2000.

[5] K. L. Clarkson. An algorithm for approximate closest-point queries. In *Proc. Tenth Annu. Sympos. Comput. Geom.*, pages 160–164, 1994.

[6] T. M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. *Discrete Comput. Geom.*, 16:369–387, 1996.

[7] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. Assoc. Comput. Mach.*, 51:606–635, 2004.

[8] J. Erickson, L. J. Guibas, J. Stolfi, and L. Zhang. Separation-sensitive collision detection for convex objects. In *Proc. Tenth Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 327–336, 1999.

[9] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[10] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2010.

[11] D. P. Dobkin and D. G. Kirkpatrick. Fast detection of polyhedral intersection. *Theo. Comp. Sci.*, 27(3):241–253, 1983.

[12] J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Appl.*, 2:169–186, 1992.

[13] B. Chazelle and D. P. Dobkin. Intersection of convex objects in two and three dimensions. *J. Assoc. Comput. Mach.*, 34:1–27, 1987.

[14] L. Barba and S. Langerman. Optimal detection of intersections between convex polyhedra. In *Proc. 26th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 1641–1654, 2015.

[15] P. M. Gruber. Asymptotic estimates for best and stepwise approximation of convex bodies I. *Forum Math.*, 5:521–537, 1993.

[16] R. Schneider. Polyhedral approximation of smooth convex bodies. *J. Math. Anal. Appl.*, 128:470–474, 1987.

[17] K. L. Clarkson. Building triangulations using ε-nets. In *Proc. 38th Annu. ACM Sympos. Theory Comput.*, pages 326–335, 2006.

[18] K. Böröczky, Jr. Approximation of general smooth convex bodies. *Adv. Math.*, 153:325–341, 2000.

[19] R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *J. Approx. Theory*, 10(3):227–236, 1974.

[20] E. M. Bronshteyn and L. D. Ivanov. The approximation of convex sets by polyhedra. *Siberian Math. J.*, 16:852–853, 1976.

[21] P. McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.

[22] R. Seidel. The upper bound theorem for polytopes: an easy proof of its asymptotic version. *Computational Geometry*, 5(2):115 – 116, 1995.

[23] D. Gale. Neighborly and cyclic polytopes. In *Proc. Sympos. Pure Math*, volume 7, pages 225–232, 1963.

[24] S. Arya, G. D. da Fonseca, and D. M. Mount. Polytope approximation and the Mahler volume. In *Proc. 23rd Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 29–42, 2012.

[25] S. Arya, G. D. da Fonseca, and D. M. Mount. Optimal area-sensitive bounds for polytope approximation. In *Proc. 28th Annu. Sympos. Comput. Geom.*, pages 363–372, 2012.

[26] S. Arya, G. D. da Fonseca, and D. M. Mount. On the combinatorial complexity of approximating polytopes. *Discrete & Computational Geometry*, 58(4):849–870, 2017.

[27] S. Arya, G. D. da Fonseca, and D. M. Mount. Approximate polytope membership queries. *SIAM J. Comput.*, 47(1):1–51, 2018.

[28] S. Arya, G. D. da Fonseca, and D. M. Mount. Approximate convex intersection detection with applications to width and Minkowski sums. In *Proc. 26th Annu. European Sympos. Algorithms*, pages 3:1–14, 2018.

[29] R. Arya, S. Arya, G. D. da Fonseca, and D. M. Mount. *Optimal Bound on the Combinatorial Complexity of Approximating Polytopes*, pages 786–805.

[30] I. Bárány. The technique of M-regions and cap-coverings: A survey. *Rend. Circ. Mat. Palermo*, 65:21–38, 2000.

[31] O. Devillers, M. Glisse, and X. Goaoc. Complexity analysis of random geometric structures made simpler. In *Proc. 29th Annu. Sympos. Comput. Geom.*, pages 167–176, 2013.

[32] J. L. Bentley, M. G. Faust, and F. P. Preparata. Approximation algorithms for convex hulls. *Commun. ACM*, 25(1):64–68, 1982.

[33] S. Arya, G. D. da Fonseca, and D. M. Mount. Approximate polytope membership queries. In *Proc. 43rd Annu. ACM Sympos. Theory Comput.*, pages 579–586, 2011.

[34] S. Arya, G. D. da Fonseca, and D. M. Mount. Optimal approximate polytope membership. In *Proc. 28th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 270–288, 2017.

[35] A. M. Macbeath. A theorem on non-homogeneous lattices. *Ann. of Math.*, 56:269–293, 1952.

[36] A. Andoni, P. Indyk, and I. Razenshteyn. Approximate nearest neighbor search in high dimensions. In *Proceedings of the International Congress of Mathematicians*, pages 3287–3318, 2018.

[37] K. L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, 17(4):830–847, 1988.

[38] A. C. Yao and F. F. Yao. A general approach to d-dimensional geometric queries. In *Proc. 17th Annu. ACM Sympos. Theory Comput.*, page 163–168, 1985.

[39] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *J. Assoc. Comput. Mach.*, 45:891–923, 1998.

[40] S. N. Bespamyatnikh. Dynamic algorithms for approximate neighbor searching. In *Proc. Eighth Canad. Conf. Comput. Geom.*, pages 252–257, 1996.

[41] T. M. Chan. Closest-point problems simplified on the RAM. In *Proc. 13th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 472–473, 2002.

[42] C. A. Duncan, M. T. Goodrich, and S. Kobourov. Balanced aspect ratio trees: Combining the advantages of k-d trees and octrees. *J. Algorithms*, 38:303–333, 2001.

[43] T. M. Chan. Approximate nearest neighbor queries revisited. *Discrete Comput. Geom.*, 20:359–373, 1998.

[44] S. Har-Peled. A replacement for Voronoi diagrams of near linear size. In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 94–103, 2001.

[45] Y. Sabharwal, S. Sen, and N. Sharma. Nearest neighbors search using point location in balls with applications to approximate Voronoi decompositions. *J. Comput. Sys. Sci.*, 72:955–977, 2006.

[46] S. Arya, T. Malamatos, and D. M. Mount. Space-efficient approximate Voronoi diagrams. In *Proc. 34th Annu. ACM Sympos. Theory Comput.*, pages 721–730, 2002.

[47] S. Arya, T. Malamatos, and D. M. Mount. Space-time tradeoffs for approximate nearest neighbor searching. *J. Assoc. Comput. Mach.*, 57:1–54, 2009.

[48] S. Arya and T. Malamatos. Linear-size approximate Voronoi diagrams. In *Proc. 13th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 147–155, 2002.

[49] S. Arya, G. D. da Fonseca, and D. M. Mount. A unified approach to approximate proximity searching. In *Proc. 18th Annu. European Sympos. Algorithms*, pages 374–385, 2010.

[50] S. Arya, G. D. da Fonseca, and D. M. Mount. Near-optimal ε-kernel construction and related problems. In *Proc. 33rd Internat. Sympos. Comput. Geom.*, pages 10:1—10:15, 2017.

[51] T. M. Chan. Applications of Chebyshev polynomials to low-dimensional computational geometry. *J. Comput. Geom.*, 9(2):3–20, 2017.

[52] S. Har-Peled and N. Kumar. Approximating minimization diagrams and generalized proximity search. *SIAM J. Comput.*, 44:944–974, 2015.

[53] S.-W. Cheng, T. Dey, and J. Shewchuk. *Delaunay Mesh Generation.* CRC Press, 2012.

[54] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. In *Proceedings of the fourteenth annual symposium on Computational geometry*, pages 86–95. ACM, 1998.

[55] S.-W. Cheng and S.-H. Poon. Three-dimensional Delaunay mesh generation. *Discrete & Computational Geometry*, 36(3):419–456, 2006.

[56] S. Oudot, L. Rineau, and M. Yvinec. Meshing volumes with curved boundaries. *Eng. with Comput.*, 26(3):265–279, June 2010.

[57] J. Tournois, C. Wormser, P. Alliez, and M. Desbrun. Interleaving Delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. *ACM Trans. Graph.*, 28(3):75:1–75:9, 2009.

[58] The CGAL Project. *CGAL User and Reference Manual.* CGAL Editorial Board, 4.14.1 edition, 2019.

[59] S.-W. Cheng, T. K. Dey, and E. A. Ramos. Delaunay refinement for piecewise smooth complexes. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1096–1105, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[60] T. K. Dey and J. A. Levine. Delaunay meshing of piecewise smooth complexes without expensive predicates. *Algorithms*, 2(4):1327–1349, 2009.

[61] C. Wojtan, N. Thürey, M. Gross, and G. Turk. Deforming meshes that split and merge. *ACM Trans. Graph.*, 28(3):76:1–76:10, 2009.

[62] M. Wicke, D. Ritchie, B. M. Klingner, S. Burke, J. R. Shewchuk, and J. F. O'Brien. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph.*, 29(4):49:1–49:11, 2010.

[63] P. Clausen, M. Wicke, J. R. Shewchuk, and J. F. O'Brien. Simulating liquids and solid-liquid interactions with Lagrangian meshes. *ACM Trans. Graph.*, 32(2):17:1–17:15, 2013.

[64] Z. Chen, M. Yao, R. Feng, and H. Wang. Physics-inspired adaptive fracture refinement. *ACM Trans. Graph.*, 33(4):113:1–113:7, 2014.

[65] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun. Variational tetrahedral meshing. *ACM Trans. Graph.*, 24(3):617–625, 2005.

[66] D.-M. Yan, W. Wang, B. Lévy, and Y. Liu. Efficient computation of 3D clipped Voronoi diagram. volume 45, pages 843 – 852. 2013. Geometric Modeling and Processing 2010.

[67] M. S. Ebeida and S. A. Mitchell. Uniform random Voronoi meshes. In *Proceedings of the 20th International Meshing Roundtable*, pages 273–290, 2012.

[68] M. Budninskiy, B. Liu, F. de Goes, Y. Tong, P. Alliez, and M. Desbrun. Optimal Voronoi tessellations with Hessian-based anisotropy. *ACM Trans. Graph.*, 35(6):242:1–242:12, 2016.

[69] R. V. Garimella, J. Kim, and M. Berndt. Polyhedral mesh generation and optimization for non-manifold domains. In *Proceedings of the 22nd International Meshing Roundtable*, pages 313–330, 2014.

[70] E. VanderZee, A. N. Hirani, D. Guoy, and E. A. Ramos. Well-centered triangulation. *SIAM Journal on Scientific Computing*, 31(6):4497–4523, 2010.

[71] H. Si, K. Gärtner, and J. Fuhrmann. Boundary conforming Delaunay mesh generation. *Computational Mathematics and Mathematical Physics*, 50(1):38–53, 2010.

[72] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12):2182–2189, 1965.

[73] B. Perot. Conservation properties of unstructured staggered mesh schemes. *Journal of Computational Physics*, 159(1):58 – 89, 2000.

[74] P. B. Bochev and J. M. Hyman. Principles of mimetic discretizations of differential operators. In *Compatible Spatial Discretizations*, pages 89–119, 2006.

[75] M. Desbrun, E. Kanso, and Y. Tong. Discrete differential forms for computational modeling. In *Discrete Differential Geometry*, volume 38 of *Oberwolfach Seminars*, pages 287–324. Birkhäuser Basel, 2008.

[76] P. Mullen, P. Memari, F. de Goes, and M. Desbrun. HOT: Hodge-optimized triangulations. *ACM Trans. Graph.*, 30(4):103:1–103:12, 2011.

[77] W. J. M. Rankine. Principle of the equilibrium of polyhedral frames. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 27(180):92–, 1864.

[78] J. C. Maxwell. On reciprocal figures, frames, and diagrams of forces. *Transactions of the Royal Society of Edinburgh*, 26(1):1–40, 1870.

[79] R. H. Macneal. An asymmetrical finite difference network. *Quarterly of Applied Mathematics*, 11(3):295–310, 1953.

[80] F. de Goes, P. Memari, P. Mullen, and M. Desbrun. Weighted triangulations for geometry processing. *ACM Trans. Graph.*, 33(3):28:1–28:13, 2014.

[81] M. Akbarzadeh, T. Van Mele, and P. Block. On the equilibrium of funicular polyhedral frames and convex polyhedral force diagrams. *Computer-Aided Design*, 63:118 – 128, 2015.

[82] C. Mercat. Discrete Riemann surfaces and the Ising model. *Communications in Mathematical Physics*, 218(1):177–216, 2001.

[83] Darren E. Generalised primal-dual grids for unstructured co-volume schemes. *Journal of Computational Physics*, 375:155 – 176, 2018.

[84] R. A. Nicolaides and X. Wu. Covolume solutions of three-dimensional div-curl equations. *SIAM Journal on Numerical Analysis*, 34(6):2195–2203, 1997.

[85] S. Elcott, Y. Tong, E. Kanso, P. Schröder, and M. Desbrun. Stable, circulation-preserving, simplicial fluids. *ACM Trans. Graph.*, 26(1), 2007.

[86] N Sukumar and J. E. Bolander. Voronoi-based interpolants for fracture modelling. In *Tessellations in the Sciences: Virtues, Techniques and Applications of Geometric Tilings*. 2009.

[87] I. L. Novak, F. Gao, Y.-S. Choi, D. Resasco, J. C. Schaff, and B. M. Slepchenko. Diffusion on a curved surface coupled to diffusion in the volume: Application to cell biology. *Journal of Computational Physics*, 226(2):1271 – 1290, 2007.

[88] S. Arya, G. D. da Fonseca, and D. M. Mount. On the combinatorial complexity of approximating polytopes. *Discrete & Computational Geometry*, 58(4):849–870, 2017.

[89] G. Ewald, D. G. Larman, and C. A. Rogers. The directions of the line segments and of the r-dimensional balls on the boundary of a convex body in Euclidean space. *Mathematika*, 17:1–20, 1970.

[90] H. Brönnimann, B. Chazelle, and J. Pach. How hard is halfspace range searching. *Discrete Comput. Geom.*, 10:143–155, 1993.

[91] R. Krauthgamer and J. R. Lee. Navigating nets: Simple algorithms for proximity search. In *Proc. 15th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 798–807, 2004.

[92] S. Har-Peled and M. Mendel. Fast construction of nets in low dimensional metrics, and their applications. *SIAM J. Comput.*, 35:1148–1184, 2006.

[93] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proc. 23rd Internat. Conf. on Machine Learning*, pages 97–104, 2006.

[94] D. Hilbert. Ueber die gerade linie als kürzeste verbindung zweier punkte. *Mathematische Annalen*, 46:91–96, 1895.

[95] A. Papadopoulos and M. Troyanov. *Handbook of Hilbert Geometry*. European Mathematical Society, 2014.

[96] F. Nielsen and L. Shao. On Balls in a Hilbert Polygonal Geometry (Multimedia Contribution). In *Proc. 33rd Internat. Sympos. Comput. Geom.*, pages 67:1–67:4, 2017.

[97] C. Vernicos and C. Walsh. Flag-approximability of convex bodies and volume growth of Hilbert geometries. HAL Archive (hal-01423693i), 2016.

[98] I. Bárány and D. G. Larman. Convex bodies, economic cap coverings, random polytopes. *Mathematika*, 35:274–291, 1988.

[99] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, 21:579–597, 1996.

[100] K. Ball. An elementary introduction to modern convex geometry. In S. Levy, editor, *Flavors of Geometry*, pages 1–58. Cambridge University Press, 1997. (MSRI Publications, Vol. 31).

[101] S. Arya, G. D. da Fonseca, and D. M. Mount. Approximate polytope membership queries. *SIAM J. Comput.*, 2017. To appear.

[102] J.-D. Boissonnat and M. Karavelas. On the combinatorial complexity of Euclidean Voronoi cells and convex hulls of d-dimensional spheres. In *Proc. 14th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 305–312, 2003.

[103] L. P. Chew and R. L. Drysdale III. Voronoi diagrams based on convex distance functions. In *Proc. First Annu. Sympos. Comput. Geom.*, pages 235–244, 1985.

[104] J.-D. Boissonnat, F. Nielsen, and R. Nock. Bregman Voronoi diagrams. *Discrete Comput. Geom.*, 44:281–307, 2010.

[105] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22(1):79–86, 1951.

[106] F. Itakura and S. Saito. Analysis synthesis telephony based on the maximum likelihood method. In *Proc. Sixth Internat. Congress Accoustics*, volume 17, pages C17–C20, 1968.

[107] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *J. of Machine Learning Research*, 6:1705–1749, 2005.

[108] S. Si, D. Tao, and B. Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE Trans. Knowl. and Data Eng.*, 22(7):929–942, 2010.

[109] L. Cayton. Fast nearest neighbor retrieval for Bregman divergences. In *Proc. 25th Internat. Conf. Machine Learning*, pages 112–119, 2008.

[110] F. Nielsen, P. Piro, and M. Barlaud. Bregman vantage point trees for efficient nearest neighbor queries. *2009 IEEE Internat. Conf. on Multimedia and Expo*, pages 878–881, 2009.

[111] M. R. Ackermann and J. Blömer. Coresets and approximate clustering for Bregman divergences. In *Proc. 20th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 1088–1097, 2009.

[112] A. Abdullah, J. Moeller, and S. Venkatasubramanian. Approximate Bregman near neighbors in sublinear time: Beyond the triangle inequality. In *Proc. 28th Annu. Sympos. Comput. Geom.*, pages 31–40, 2012.

[113] A. Abdullah and S. Venkatasubramanian. A directed isoperimetric inequality with application to Bregman near neighbor lower bounds. In *Proc. 47th Annu. ACM Sympos. Theory Comput.*, pages 509–518, 2015.

[114] A. Abdelkader and D. M. Mount. Economical Delone sets for approximating convex bodies. In *Proc. 16th Scand. Workshop Algorithm Theory*, pages 4:1–4:12, 2018.

[115] F. Aurenhammer. Power diagrams: Properties, algorithms and applications. *SIAM J. Comput.*, 16:78–96, 1987.

[116] D. P. Bertsekas. Convexification procedures and decomposition methods for nonconvex optimization problems 1. *J. Optim. Theory Appl.*, 29:169–197, 1979.

[117] I. P. Androulakis, C. D. Maranas, and C. A. Floudas. αBB: A global optimization method for general constrained nonconvex problems. *J. Global Optim.*, 7:337–363, 1995.

[118] M. Sharir. Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete Comput. Geom.*, 12:327–345, 1994.

[119] S. Arya and D. M. Mount. Approximate range searching. *Comput. Geom. Theory Appl.*, 17:135–163, 2000.

[120] S. Arya, T. Malamatos, and D. M. Mount. The effect of corners on the complexity of approximate range searching. *Discrete Comput. Geom.*, 41:398–443, 2009.

[121] M. Peric and S. Ferguson. The advantage of polyhedral meshes. *Dynamics - Issue 24*, pages 4–5, Spring 2005. The customer magazine of the CD-adapco Group, currently maintained by Siemens at http://siemens.com/mdx. The issue is available at http://mdx2.plm.automation.siemens.com/magazine/dynamics-24. See also: https://pdfs.semanticscholar.org/51ae/90047ab44f53849196878bfec4232b291d1c.pdf (accessed November 11, 2019).

[122] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete & Computational Geometry*, 22(4):481–504, December 1999.

[123] N. Amenta, S. Choi, and R.-K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(2):127–153, July 2001.

[124] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical models and image processing*, 60(2):125–135, 1998.

[125] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 415–421, 1998.

[126] N. Amenta, S. Choi, and R.-K. Kolluri. The power crust. In *Proceedings of the Sixth ACM Symp. on Solid Modeling and Applications*, pages 249–266, 2001.

[127] N. Amenta, S. Choi, T. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *16th Annual Symposium on Computational Geometry*, pages 213–222, 2000.

[128] T. Dey, K. Li, E. Ramos, and R. Wenger. Isotopic reconstruction of surfaces with boundaries. In *Computer Graphics Forum*, volume 28:5, pages 1371–1382, 2009.

[129] F. Chazal and A. Lieutier. Smooth manifold reconstruction from noisy and non-uniform approximation with guarantees. *Computational Geometry*, 40(2):156 – 170, 2008.

[130] H. Edelsbrunner. The union of balls and its dual shape. *Discrete & Computational Geometry*, 13(3):415–440, Jun 1995.

[131] N. Amenta and R.-K. Kolluri. The medial axis of a union of balls. *Computational Geometry*, 20(1):25 – 37, 2001. Selected papers from the 12th Annual Canadian Conference.

[132] F. Cazals, H. Kanhere, and S. Loriot. Computing the volume of a union of balls: A certified algorithm. *ACM Trans. Math. Softw.*, 38(1):3:1–3:20, 2011.

[133] F. Cazals, T. Dreyfus, S. Sachdeva, and N. Shah. Greedy geometric algorithms for collection of balls, with applications to geometric approximation and molecular coarse-graining. *Computer Graphics Forum*, 33(6):1–17, 2014.

[134] K. Sykes D. Letscher. On the stability of medial axis of a union of disks in the plane. In *28th Canadian Conference on Computational Geometry, CCCG 2016*, pages 29–33, 2016.

[135] Ø. Klemetsdal, R. Berge, K.-A. Lie, H. Nilsen, and O. Møyner. *SPE-182666-MS*, chapter Unstructured Gridding and Consistent Discretizations for Reservoirs with Faults and Complex Wells. Society of Petroleum Engineers, 2017.

[136] O. Aichholzer, F. Aurenhammer, B. Kornberger, S. Plantinga, G. Rote, A. Sturm, and G. Vegter. Recovering structure from r-sampled objects. *Computer Graphics Forum*, 28(5):1349–1360, 2009.

[137] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, Oct 1999.

[138] P. Stelldinger. Topologically correct surface reconstruction using alpha shapes and relations to ball-pivoting. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[139] H. Edelsbrunner and E.-P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9(1):66–104, January 1990.

[140] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. *Journal of Algorithms*, 30(2):302–322, 1999.

[141] N. Amenta, T. Peters, and A. Russell. Computational topology: ambient isotopic approximation of 2-manifolds. *Theoretical Computer Science*, 305(1):3 – 15, 2003. Topology in Computer Science.

[142] F. Chazal and D. Cohen-Steiner. A condition for isotopic approximation. *Graphical Models*, 67(5):390 – 404, 2005. Solid Modeling and Applications.

[143] S.-W. Cheng, T. Dey, H. Edelsbrunner, M. Facello, and S.-H. Teng. Silver exudation. *J. ACM*, 47(5):883–904, September 2000.

[144] H. Edelsbrunner. *Weighted alpha shapes*. University of Illinois at Urbana-Champaign, Department of Computer Science, 1992.

[145] P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

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

[147] N. Amenta and T. Dey. Normal variation for adaptive feature size. *CoRR*, abs/1408.0314, 2014.

[148] T. Dey. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*. Cambridge University Press, New York, NY, USA, 2006.

[149] G. Miller, D. Talmor, and S.-H. Teng. Data generation for geometric algorithms on non-uniform distributions. *International Journal of Computational Geometry and Applications*, 09(06):577–597, 1999.

[150] J. Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures. preprint, University of California at Berkeley, 2002.

[151] S. Martin, P. Kaufmann, M. Botsch, M. Wicke, and M. Gross. Polyhedral finite elements using harmonic basis functions. *Computer Graphics Forum*, 27(5):1521–1529, 2008.

[152] J.E. Bishop. A displacement-based finite element formulation for general polyhedra using harmonic shape functions. *International Journal for Numerical Methods in Engineering*, 97(1):1–31, 2014.

[153] G. Manzini, A. Russo, and N. Sukumar. New perspectives on polygonal and polyhedral finite element methods. *Mathematical Models and Methods in Applied Sciences*, 24(08):1665–1699, 2014.

[154] A. L. Gain, C. Talischi, and G. H. Paulino. On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 282:132 – 160, 2014.

[155] K. Lipnikov, G. Manzini, and M. Shashkov. Mimetic finite difference method. *Journal of Computational Physics*, 257:1163 – 1227, 2014. Physics-compatible numerical methods.

[156] A. L. Gain, C. Talischi, and G. H. Paulino. On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 282:132 – 160, 2014.

[157] J. Wu, R. Westermann, and C. Dick. A survey of physically based simulation of cuts in deformable bodies. *Computer Graphics Forum*, 34(6):161–187, 2015.

[158] C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes. Polygonal finite elements for topology optimization: A unifying paradigm. *International Journal for Numerical Methods in Engineering*, 82(6):671–698, 2013.

[159] K. Pruess. TOUGH2: A general-purpose numerical simulator for multiphase fluid and heat flow. *NASA STI/Recon Technical Report N*, 92, 1991.

[160] P. C. Lichtner, G. E. Hammond, C. Lu, S. Karra, G. Bisht, B. Andre, R. Mills, and J. Kumar. PFLOTRAN user manual: A massively parallel reactive flow and transport model for describing surface and subsurface processes. *Technical report*, LA-UR-15-20403, 2015. Los Alamos National Lab; Sandia National Lab; Lawrence Berkeley National Lab; Oak Ridge National Lab; OFM Research.

[161] K. Pruess. The TOUGH codes — a family of simulation tools for multiphase flow and transport processes in permeable media. *Vadose Zone Journal*, 3, 2004.

[162] C.M. Freeman, K. L. Boyle, M. Reagan, J. Johnson, C. Rycroft, and G.J. Moridis. MeshVoro: A three-dimensional Voronoi mesh building tool for the TOUGH family of codes. *Computers & Geosciences*, 70:26 – 34, 2014.

[163] L. Hu, K. Zhang, X. Cao, Y. Li, and C. Guo. IGMESH: A convenient irregular-grid-based pre- and post-processing tool for TOUGH2 simulator. *Computers & Geosciences*, 95:11 – 17, 2016.

[164] S. Bonduà, V. Bortolotti, P. Macini, E. Mesini, and E. M. Vasini. 3D Voronoi pre- and post- processing tools for using the TOUGH2 family of numerical simulator for hydrocarbon gas migration. In *Offshore Mediterranean Conference*, 2017.

[165] S.-K. Kim, G.-O. Bae, and K.-K. Lee. Improving accuracy and flexibility of numerical simulation of geothermal heat pump systems using Voronoi grid refinement approach. *Geosciences Journal*, 19(3):527–535, 2015.

[166] R. V. Garimella and K. Lipnikov. Solution of the diffusion equation in multi-material domains by sub-division of elements along reconstructed interfaces. *International Journal for Numerical Methods in Fluids*, 65(11-12):1423–1437, 2011.

[167] E. Kikinzon, Y. Kuznetsov, K. Lipnikov, and M. Shashkov. Approximate static condensation algorithm for solving multi-material diffusion problems on meshes non-aligned with material interfaces. *Journal of Computational Physics*, 347:416 – 436, 2017.

[168] A. S. Dawes. Three-dimensional multi-material polyhedral method for diffusion. *Computers & Fluids*, 156:485 – 495, 2017.

[169] S.-W. Cheng, T. Dey, and E. Ramos. Delaunay refinement for piecewise smooth complexes. *Discrete & Computational Geometry*, 43(1):121–166, Jan 2010.

[170] T. K. Dey and T. Ray. Polygonal surface remeshing with Delaunay refinement. *Engineering with Computers*, 26(3):289–301, 2010.

[171] Y. Hu, Q. Zhou, X. Gao, A. Jacobson, D. Zorin, and D. Panozzo. Tetrahedral meshing in the wild. *ACM Trans. Graph.*, 37(4):60:1–60:14, 2018.

[172] C. T. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, Department of Mathematics, The University of Utah, August 1987.

[173] A. Abdelkader, C. L. Bajaj, M. S. Ebeida, A. H. Mahmoud, S. A. Mitchell, J. D. Owens, and A. A. Rushdi. Sampling conditions for conforming Voronoi meshing by the VoroCrust algorithm. In *34th International Symposium on Computational Geometry (SoCG 2018)*, volume 99 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:16, 2018.

[174] M. S. Ebeida, S. A. Mitchell, A. Patney, A. A. Davidson, and J. D. Owens. A simple algorithm for maximal Poisson-disk sampling in high dimensions. *Computer Graphics Forum*, 31(2):785–794, 2012.

[175] D.-M. Yan and P. Wonka. Gap processing for adaptive maximal Poisson-disk sampling. *ACM Transactions on Graphics*, 32(5):148:1–148:15, 2013.

[176] J. Guo, D.-M. Yan, X. Jia, and X. Zhang. Efficient maximal Poisson-disk sampling and remeshing on surfaces. *Computers & Graphics*, 46:72 – 79, 2015.

[177] S. A. Mitchell, A. Rand, M. S. Ebeida, and C. Bajaj. Variable radii Poisson-disk sampling. In *Proceedings of the 24th Canadian Conference on Computational Geometry (CCCG)*, 2012.

[178] A. DasGupta. Metrics, information theory, convergence, and Poisson approximations. In *Asymptotic Theory of Statistics and Probability*, Springer Texts in Statistics, chapter 2, pages 19–34. Springer, 2008.

[179] S. A. Mitchell, M. S. Ebeida, M. A. Awad, C. Park, A. Patney, A. A. Rushdi, L. P. Swiler, D. Manocha, and L.-Y. Wei. Spoke-darts for high-dimensional blue-noise sampling. *ACM Trans. Graph.*, 37(2):22:1–22:20, 2018.

[180] C. Rycroft. Voro++: A three-dimensional Voronoi cell library in C++. *Chaos*, 19(4):–, 2009. Software available online at http://math.lbl.gov/voro++/.

[181] M. B. Cohen, B. Cousins, Y. T. Lee, and X. Yang. A near-optimal algorithm for approximating the john ellipsoid. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 849–873. PMLR, 2019.

[182] A. Laddha, Y. T. Lee, and S. Vempala. Strong self-concordance and sampling. page 1212–1222, 2020.

[183] A. Choudhary, M. Kerber, and S. Raghvendra. *Improved Topological Approximations by Digitization*, pages 2675–2688.

[184] A. Abdelkader, A. Mahmoud, A. Rushdi, S. Mitchell, J. Owens, and M. Ebeida. A constrained resampling strategy for mesh improvement. *Computer Graphics Forum*, 36(5):189–201, 2017.

[185] D. Sieger, P. Alliez, and M. Botsch. Optimizing Voronoi diagrams for polygonal finite element computations. In *International Meshing Roundtable (IMR)*, pages 335–350. Springer, 2010.

[186] L. Wang, F. Hétroy-Wheeler, and E. Boyer. On volumetric shape reconstruction from implicit forms. In *Computer Vision – ECCV 2016*, pages 173–188, 2016.

[187] B. Mederos, N. Amenta, L. Velho, and L. H. de Figueiredo. Surface reconstruction from noisy point clouds. In *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP '05, 2005.

[188] M. Attene, M. Campen, and L. Kobbelt. Polygon mesh repairing: an application perspective. *ACM Comput. Surv.*, 45(2):15:1–15:33, 2013.

[189] A. Kumar, A. Shih, Y. Ito, D. Ross, and B. Soni. A hole-filling algorithm using non-uniform rational B-splines. In *Proceedings of the 16th International Meshing Roundtable*, pages 169–182, 2008.

[190] R. V. Garimella and M. S. Shephard. Boundary layer mesh generation for viscous flow simulations. *International Journal for Numerical Methods in Engineering*, 49(1-2):193–218, 2000.

[191] A. Loseille and R. Löhner. Robust boundary layer mesh generation. In *Proceedings of the 21st International Meshing Roundtable*, pages 493–511, 2013.

[192] X.-M. Fu, Y. Liu, J. Snyder, and B. Guo. Anisotropic simplicial meshing using local convex functions. *ACM Trans. Graph.*, 33(6):182:1–182:11, 2014.