Lines: 487

Approximating Multiplicatively Weighted Voronoi Diagrams: Efficient Construction with Linear Size

Joachim Gudmundsson
University of Sydney, Australia

Martin P. Seybold
University of Vienna, Faculty of Computer Science, Theory and Applications of Algorithms, Währinger Straße 29, A-1090 Vienna, Austria

Sampson Wong
University of Copenhagen, Copenhagen

Abstract

Given a set of n sites from $\mathbb{R}^d$, each having some positive weight factor, the Multiplicatively Weighted Voronoi Diagram is a subdivision of space that associates each cell to the site whose weighted Euclidean distance is minimal for all points in the cell.

We give novel approximation algorithms that output a cube-based subdivision such that the weighted distance of a point with respect to the associated site is at most $(1+\varepsilon)$ times the minimum weighted distance, for any fixed parameter $\varepsilon \in (0,1)$. The diagram size is $O_d(n \log(1/\varepsilon)/\varepsilon^{d-1})$ and the construction time is within an $O_D(\log(n)/\varepsilon^{(d+5)/2})$-factor of the size bound. We also prove a matching lower bound for the size, showing that the proposed method is the first to achieve *optimal size*, up to $\Theta(1)^d$-factors. In particular, the obscure $\log(1/\varepsilon)$ factor is unavoidable. As a by-product, we obtain a factor $d^{O(d)}$ improvement in size for the unweighted case and $O(d\log(n) + d^2\log(1/\varepsilon))$ point-location time in the subdivision, improving the known query bound by one d-factor.

The key ingredients of our approximation algorithms are the study of convex regions that we call cores, an adaptive refinement algorithm to obtain optimal size, and a novel notion of *bisector coresets*, which may be of independent interest. In particular, we show that coresets with $O_d(1/\varepsilon^{(d+3)/2})$ worst-case size can be computed in near-linear time.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Multiplicatively Weighted Voronoi Diagram, Compressed QuadTree, Adaptive Refinement, Bisector Coresets, Semi-Separated Pair Decomposition, Lower Bound

Related Version See https://arxiv.org/abs/2112.12350 for the full version of the paper.

Acknowledgements This work was supported under the Australian Research Council Discovery Projects funding scheme (project number DP180102870) and partially supported by Starting Grant 1054-00032B from the Independent Research Fund Denmark under the Sapere Aude research career programme. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101019564) and the Austrian Science Fund (FWF) project Z 422-N, project I 5982-N, and project P 33775-N, with additional funding from the *netidee SCIENCE Stiftung*, 2020–2024.

1 Introduction

Voronoi Diagrams are structures of fundamental importance for many scientific fields. In particular, planar variants with linear worst-case size are very well understood (e.g. [8, 10]). Though closely related to the Nearest-Neigbhor search problem, the *explicit* subdivisions provided by Voronoi Diagrams are a central tool for various problems, including meshing in scientific computing, planning of facility locations, motion planning, or surface reconstruction.

Given a set of sites $\{s_1, \ldots, s_n\} \subset \mathbb{R}^d$, each having a positive weight $w_i > 0$, their Multiplicatively Weighted Voronoi Diagram (MWVD) is the subdivision of $\mathbb{R}^d$ into cells that associates each cell to one site, i.e. the site s_i that minimizes $\|p - s_i\|_2/w_i$ for all points p in the cell. Though all bisectors in an MWVD are either half-spaces ($w_i = w_j$) or Apollonian spheres ($w_i \neq w_j$), the two main difficulties with MWVDs are that Voronoi regions may contain holes, and that the multiplicative weights can violate the triangle inequality.

The MWVD in $\mathbb{R}^1$ has linear size and can be obtained using a Divide & Conquer algorithm in $O(n \log n)$ time [6]. Aurenhammer and Edelsbrunner showed that MWVDs in $\mathbb{R}^2$ can have $\Omega(n^2)$ size and gave a worst-case optimal algorithm [7]. Held and de Lorenzo [17] gave a sweep approach for 2D that runs in $O(n^2 \log n)$ time. In special cases, 2D MWVD size is known to have near-linear, or even linear, bounds [16, 11]. In general, unweighted Voronoi Diagrams, i.e. all $w_i = 1$, are well known to have $\Omega(n^{\lceil d/2 \rceil})$ worst-case size (see e.g. [13]).

Importance of cube-based Approximate Voronoi Diagrams. We limit our discussion on two applications where the simplicity of cube-based AVDs is key for strong bounds.

(i) Axis-Aligned Segment-Queries in 2D.
Using Chazelle's Point-Location & Walk method [9, Sect. 4.2] on an 2D MWVD, it is possible to traverse all k cells of the subdivision that are intersected by an axis-aligned query line-segment in $O(\log(n) + k)$ time, which determines the $\Omega(k)$ distinct nearest-sites for (the sequence of points that are contained in) the query-segment.
Now, an approximate subdivision that consists of canonical squares, or set difference of canonical squares, allows to merge common boundaries of adjacent squares, associated to the same Voronoi site, without increasing the size bound of the subdivision. Thus, allowing to retain the $O(\log(n) + k)$ query bound in the approximate setting.

(ii) Fast Point-Queries when d is large. The 'curse of dimensionality' typically refers to the broad phenomena that either the query-bounds or the space-bounds of known structures for (exact) nearest-neighbor search deteriorate 'quickly' as d increases. In ε-approximate nearest-neighbor search, we are mainly interested in the range $d = 2$ to $d = O(\log(n)/\varepsilon^2)$, due to Johnson-Lindenstrauss dimension reduction (see, e.g., [13, 14]).
Now, cube-based subdivisions allow to use compressed QuadTrees to obtain *very strong* query bounds. For example, in a subdivision of $\mathbb{R}^d$ with $N = O(n/\varepsilon^d)$ cubes, the query time is $O(d \log(n/\varepsilon^d)) = O(d \log(n) + d^2 \log(1/\varepsilon))$.
In contrast, query bounds containing $O(1)^d$-terms are only fast when d is *very small*.

For careful comparison with respect to the dimension, we distinguish between O-notation, O_D-notation that assumes a 'constant-dimension' and hides $d^{O(d)}$-factors, and O_d-notation that assumes a 'small-dimension' and hides $O(1)^d$-factors. E.g. $O((8d)^d) = O_d(d^d) = O_D(1)$. Note that there is a separation between space bounds in the O_D-regime and the O_d-regime. For $d = O(\log \log n)$, any $O(1)^d$ factors in size are $O(\operatorname{polylog} n)$ factors, whereas d^d-factors are $\omega(\operatorname{polylog} n)$. Further, c^d-factors in size are sub-linear $O(n^{1/p})$ for $d \leq \log_c(n)/p$, unlike d^d-factors.

This work studies the problem of computing ε-Approximate MWVDs for prescribed $\varepsilon > 0$. That is, a subdivision of $\mathbb{R}^d$ into cells that are cubes, or set-difference of cubes, that associates each cell with one site that is an ε-approximate weighted nearest-neighbor for all points in the cell. The only known solution til date is to employ the, more general, framework of Har-Peled and Kumar [15], which, e.g., found application in the work [3].

Contribution and Paper Organization. Our approach considers convex regions that we call 'cores', which are the intersection of at most $n-1$ Apollonian balls of MWVD bisectors. In Section 3, we introduce an Adaptive Refinement algorithm that ε-approximates each core with

Diagram	Technique	Size	Runtime
ε-AVD	Clustering, PLEB [12]	$O_D\left(n\frac{\log n}{\varepsilon^d}\log\frac{n}{\varepsilon}\right)$	$\times O_D\left(\log\frac{n}{\varepsilon}\right)$
ε-AVD	Clustering, ε-PLSB [18]	$O_D\left(n\frac{\log 1/\varepsilon}{\varepsilon^{d+1}}\right)$	$\times O_D\left(\log\frac{n}{\varepsilon}\right)$
ε-AVD	Triangle ineq., 8-WSPD [4, p148]	$O_d\left(n\left(\frac{d}{\varepsilon}\right)^d\log\frac{1}{\varepsilon}\right)$	$\times O_D\left(\frac{1}{\varepsilon^d}\log\frac{n}{\varepsilon}\right)$
$(1,\varepsilon)$-AVD	Triangle ineq. [5, Cor. 9.10.f]	$O_D\left((n/\varepsilon^{d-1})\log\frac{1}{\varepsilon}\right)$	
ε-AMWVD	Clustering, Sketches [15]	$O_D\left(n\left(\frac{\log^{d+2}(n)}{\varepsilon^{2d+2}}+\frac{1}{\varepsilon^{d(d+1)}}\right)\right)$	
ε-AMWVD	Adaptive Refinement, ε^{-1}-SSPD	$\Theta_d\left(n\frac{\log 1/\varepsilon}{\varepsilon^{d-1}}\right)$	$\times O_D\left(\frac{\log n}{\varepsilon^{(d+5)/2}}\right)$

Table 1 Overview of constructions of ε-AVDs that provide *fast queries* for large d and the proposed method for ε-AMWVDs. Note that ε-AMWVDs are more general than the unweighted ε-AVDs. The time bound of [15] is $O_D\left(n\log^{2d+3}(n)/\varepsilon^{2d+2}+n/\varepsilon^{d(d+1)}\right)$, and the query time $O(d\log(n/\varepsilon^{d(d+1)}))$ is *cubic* in d. All other QuadTree based ε-AVD methods have $O(d\log(n/\varepsilon^d))$ query time.

a set of d-cubes, and show that each core is ε-approximated with $O_d(\log(1/\varepsilon)/\varepsilon^{d-1})$ cubes. In Section 3.1, we show that a top-down propagation in the compressed QuadTree over the set of d-cubes allows to obtain an ε-AMWVD that consists of $O_d(n\log(1/\varepsilon)/\varepsilon^{d-1})$ cells that are d-cubes, or the set difference of d-cubes, each of which associated to *one site* that is weighted nearest-neighbor for all points in the cell, up to a $(1+\varepsilon)$ factor. One by-product of our construction is thus a compressed QuadTree that can report an ε-NN of a query-point in $O(d\log(n)+d^2\log(1/\varepsilon))$ time, thus improving on the query-time of the structure from [15] by one d-factor.

We prove a matching lower bound on the size of the subdivision in Section 4. Specifically, we show that *every* subdivision of $\mathbb{R}^d$, formed by axis-aligned hyper-rectangles, that is an ε-approximation of an Apollonian ball must contain $\Omega_d(\log(1/\varepsilon)/\varepsilon^{d-1})$ hyper-rectangles. Our proposed bound improves on the known $\Omega_d(\varepsilon/(\varepsilon\sqrt{d})^d)$ bound from [4, 5] in two ways. First, the denominator is free of the $\sqrt{d}$-factor and, second, it is the first known lower bound that shows that a $\log(1/\varepsilon)$-factor is *required* in the space. Thus, the proposed construction is the first that computes an ε-AMWVD with worst-case optimal size, up to $\Theta_d(1)$-factors.

In Section 5, we introduce our second approximation algorithm which is the key component to improve the construction time from quadratic to near-linear. We show that cores admit an ε-approximation with low complexity, i.e. with $O_d(1/\varepsilon^{(d+3)/2})$ bisectors, and give an algorithm that outputs such bisector coresets in $O_D(n\log(n)/\varepsilon^{3(d+1)/2})$ time, based on an $O(1/\varepsilon)$-Semi-Separated Pair Decomposition (SSPD) of the site locations. If the sites are a point set with polynomially bounded spread, the construction time improves from an O_D-bound to the respective O_d-bound.

See Table 1 for an overview of the size and runtime of known ε-AVD constructions, and our proposed method. Due to the large amount of previous work, we only include those methods that also compute cube-based Approximate Voronoi Diagrams in the comparison.

2 Preliminaries

We provide a brief overview of canonical d-cubes and QuadTrees. The *canonical cube system* is an hierarchical and infinite tiling of $\mathbb{R}^d$ with canonical cubes. Level zero of the canonical cube system consists of unit cubes with vertices at integer coordinates. For all $\ell \leq -1$, we

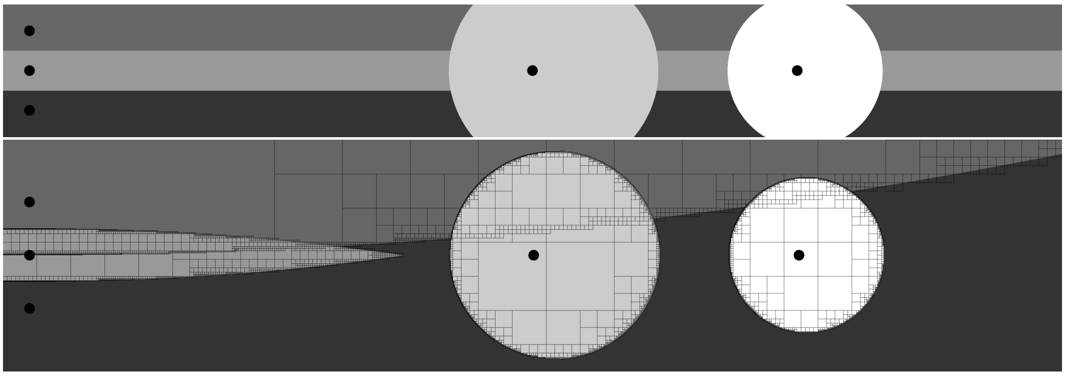

Figure 1 The top shows an example of an exact MWVD of five sites ($\varepsilon_S = 0$). The bottom shows an ε_S-AMWVD of the same instance obtained from cores with $\varepsilon_S = 0.01$. Result squares of the proposed Adaptive Refinement algorithm (Section 3) for all four cores are shown as black overlay.

construct level ℓ by bisecting each cube in level $\ell+1$ along each of the d axes. Therefore, there are 2^d cubes in level ℓ per cube in level $\ell+1$. For all $\ell \geq 1$, we merge 2^d cubes in level $\ell-1$ to obtain a single cube in level ℓ, so that the cubes in level ℓ form a tiling of $\mathbb{R}^d$. For example, a d-cube is a subset of points form $\mathbb{R}^d$ of the form $[2^\ell x_1, 2^\ell(x_1+1)] \times \ldots \times [2^\ell x_d, 2^\ell(x_d+1)]$ for integers $\ell, x_1, \ldots, x_d$. Note that any two d-cubes from the system are either interior disjoint or one cube is a subset of the other.

Given a set of n canonical d-cubes from the system, one can build a QuadTree on the set of cubes, in $O(dn \log \Delta)$ time, where Δ is the ratio between longest and shortest side length of the input set. In this work, we use compressed QuadTrees, which have $O(dn)$ size and can be constructed in $O(dn \log n)$ time. The subdivision of $\mathbb{R}^d$ induced by a QuadTree consists of canonical d-cubes, whereas the subdivision induced by a compressed QuadTree consists of regions that are the set difference of canonical d-cubes.

2.1 Voronoi Maps, Apollonian Balls, and the Core

Mapping $\lambda : \mathbb{R}^d \to \{1, \ldots, n\}$ is called a Voronoi Map for the distance functions $\{d_1, \ldots, d_n\}$, if $d_{\lambda(x)}(x) \leq \min_i d_i(x)$, for all points $x \in \mathbb{R}^d$. The d_i with index $i = \lambda(x)$ is called a nearest-neighbor of point x. In the case of Multiplicatively Weighted Voronoi Diagrams, each site $s_i \in \mathbb{R}^d$ has a positive weight-factor w_i and the distance is $d_i(x) = \|x - s_i\|/w_i$. We denote by $\|\cdot\|$ the Euclidean ℓ_2-norm and indicate other ℓ_p-norms explicitly by $\|\cdot\|_p$. A subdivision of $\mathbb{R}^d$ is called MWVD if every cell in the subdivision is associated to one input site, and if mapping the points in a cell to the associated site is a Voronoi Map. Cell boundaries occur where the weighted distances to two sites are equal, which is along an Apollonian circle for $d = 2$. For general d, we define the Apollonian sphere between s_i and s_j to be $\{x \in \mathbb{R}^d : \|x - s_i\|/w_i = \|x - s_j\|/w_j\}$. A trivial MWVD is to construct the arrangement of the $\binom{n}{2}$ Apollonian spheres, giving a polynomial size bound.

Approximate Voronoi Maps of Apollonian Spheres and cube-based ε-AVDs A mapping $\lambda : \mathbb{R}^d \to \{1, \ldots, n\}$ is called an ε-approximate Voronoi Map for the functions $\{d_1, \ldots, d_n\}$, if $d_{\lambda(x)}(x) \leq (1+\varepsilon) \min_i d_i(x)$, for all points $x \in \mathbb{R}^d$.

Recall that the MWVD bisector of s_j and s_i is a $(d-1)$-dimensional hyper-plane, if $w_j = w_i$. We introduce a parameter $\varepsilon_S \in [0, \varepsilon)$, that we calibrate in Section 5.2, and use it to ε_S-approximate hyper-planes with hyper-spheres. (This will turn out advantageous for obtaining optimal size.) Let the sites be sorted by weight, so that $w_1 \leq \ldots \leq w_n$, breaking ties arbitrary but *fixed*. We define for all indices $i < j$ the Apollonian balls

$$ball(i,j) = ball(s_i, s_j, \gamma_{ij}) = \left\{x \in \mathbb{R}^d : \|x - s_i\|\gamma_{ij} \leq \|x - s_j\|\right\} \ , \tag{1}$$

where $\gamma_{ij} := \max(w_j/w_i, 1+\varepsilon_S)$. We call γ_{ij} the ***effective weight*** of $ball(i,j)$. For $\varepsilon_S > 0$, $\gamma_{ij} \geq 1 + \varepsilon_S$ and it follows that $ball(i,j)$ is not a half-space. Note that the arrangement of the surfaces of all $\{ball(i,j)\}$ yields an ε_S-approximate Voronoi Map. See Figure 1.

To enable *fast* point location with Compressed Quad-Trees, an ε-Approximate Voronoi Diagram (ε-AVD) is a subdivision of $\mathbb{R}^d$ into d-cubes, and set-difference of d-cubes, that is an ε-approximate Voronoi Map. That is, each cube in the subdivision of $\mathbb{R}^d$ is associated to *one* input site that is an ε-Nearest-Neighbor for all points in the cube.

Closest, Furthest, and the Core of Apollonian Balls We further define $t^*(s_i, s_j, \gamma_{ij})$ to be the *closest distance* from s_i to a point on the surface of $ball(i,j)$, and $t^\dagger(s_i, s_j, \gamma_{ij})$ to be the *furthest distance* from s_i to a point on the surface of $ball(i,j)$. Note that these points are on the line through s_i and s_j, and their distances are

$$\gamma_{ij} = \max(w_j/w_i, 1+\varepsilon_S) \tag{2}$$

$$t^*_{ij} = t^*(s_i, s_j, \gamma_{ij}) = \|s_j - s_i\|/(\gamma_{ij}+1) \tag{3}$$

$$t^\dagger_{ij} = t^\dagger(s_i, s_j, \gamma_{ij}) = \|s_j - s_i\|/(\gamma_{ij}-1) \ . \tag{4}$$

For example, $ball(i,j)$ has diameter $t^*_{ij} + t^\dagger_{ij}$.

Let the set of balls of site s_i be $B_i := \{(i,j) \ : \ i < j\ \}$. For every subset $A_i \subseteq B_i$, define the convex region $core(A_i) := \bigcap_{(i,j)\in A_i} ball(i,j)$. By definition, the point $s_i \in core(A_i)$ for all non-empty $A_i \subseteq B_i$.

3 Small Approximate Voronoi Diagrams using $\binom{n}{2}$ Bisectors

The exact Voronoi region of site s_j in an MWVD is $core(B_j) \setminus \bigcup_{i<j} core(B_i)$ and a simple construction of the Voronoi Map may process the regions $core(B_j)$ by descending index j and assign all points in $core(B_j)$ to the index j. We introduce a suitable discretization for this idea next.

▶ **Lemma 1.** *There exist two balls centered at s_i, one with radius R containing $core(B_i)$, and one with radius r contained in $core(B_i)$, so that $R/r \leq 3/\varepsilon_S$. I.e. $core(B_i)$ is $3/\varepsilon_S$-fat.*

Proof. Since any bisector has $t^\dagger_{ij}/t^*_{ij} = \frac{\gamma_{ij}+1}{\gamma_{ij}-1} \leq 1 + 2/\varepsilon_S$ and the intersection of bisectors retains the maximum over those ratios, $core(B_i)$ is $3/\varepsilon_S$-fat with $r := \min_j\{t^*_{ij}\}$. ◀

To discretize a $\frac{R}{r}$-fat region for some $\varepsilon_A \in (0, \varepsilon_S)$, we consider the coarsest level where the canonical cubes have diameter at most $diam(C) \leq r\varepsilon_A$, i.e. side-length $len(C) \leq r\varepsilon_A/\sqrt{d}$. Within distance at most R from s_i, there are $O_d((\frac{2R}{r} \cdot \frac{\sqrt{d}}{\varepsilon_A})^d) = O_d((\sqrt{d}/\varepsilon_A^2)^d)$ such cubes. Checking each of the k bisectors that define the fat region, we can determine with $O(k)$ distance computations if the centroid point of a cube is in $core(B_i)$. Since any one cube is entirely inside, is entirely outside, or contains a point of the boundary, we have that only the latter case is potentially incorrect when deciding membership by the cube's centroid point. Since any point x on the boundary has $\|x - s_i\| \geq r$ and any point q with erroneous membership decision has $\|q-x\| \leq \varepsilon_A r$ from a point x on the boundary (i.e. $d_i(x) = d_j(x)$), the discretization of the core approximates within a factor

$$\frac{d_i(q)}{d_j(q)} = \frac{d_i(q)}{d_i(x)} \cdot \frac{d_i(x)}{d_j(q)} \leq \left(1 + \frac{\|x-q\|}{\|x-s_i\|}\right) \frac{d_j(x)}{d_j(q)} \leq \left(1 + \frac{\|x-q\|}{\|x-s_i\|}\right)\left(1 + \frac{\|x-q\|}{\|q-s_j\|}\right)$$

$$\leq (1+\varepsilon_A)(1+\frac{\varepsilon_A r}{\|q-s_j\|}) \leq (1+\varepsilon_A)(1+\frac{\varepsilon_A r}{\|x-s_j\| - \|x-q\|})$$

$$\leq (1+\varepsilon_A)(1+\frac{\varepsilon_A r}{\|x-s_i\| - \|x-q\|}) \leq (1+\varepsilon_A)(1+\frac{\varepsilon_A}{1-\varepsilon_A}) = 1 + O(\varepsilon_A) \ .$$

▶ **Observation 2.** *$O(1/\varepsilon)$-fat cores allow a discretization of $O_d(n(\sqrt{d}/\varepsilon^2)^d)$ total size that ε-approximates each core. Construction time is at most a factor $d \cdot n$ over the size bound.*

Note that the argument for cubes that intersect the boundary in our approximation bound already holds if the maximum distance of two points in a cube (diameter) is sufficiently small with respect to the distance to s_i, and not just if the diameter is at most $r\varepsilon_A$. Next, we discuss our, more space efficient, top-down search method that exploits this fact. (Note that $O(\log(1/\varepsilon_S))$ levels of the canonical cube system are relevant for any given core.)

As such, our Adaptive Refinement algorithm first determines $r = \min\{t^*_{ij}\}$ from the given set of k bisectors of site s_i, and then starts on the smallest canonical cube that contains the ball of radius $3r/\varepsilon_S$ around the site s_i. Recursively, we check if the current cube C is entirely inside or entirely outside, i.e. $\|centr(C) - centr(ball(i,j))\| + diam(C)/2 \leq rad(ball(i,j))$ for all $j > i$ or $\|centr(C) - centr(ball(i,j))\| - diam(C)/2 > rad(ball(i,j))$ for a $j > i$. If so, the search stops and includes the current cube C in the result set, or respectively excludes it. Otherwise, we check if the cube's diameter is sufficiently small for the centroid-test, i.e. $diam(C) \leq \varepsilon_A(\|s_i - centr(C)\| - diam(C)/2)$. If not, then all 2^d children of the cube are searched recursively. If it is, then we stop the search and include the cube in the output set based on the result of its centroid-test, i.e. cube C is included if and only if the centroid point of C is inside *each* of the k bisectors that define $core(B_i)$.

Note that the search stops descending on a cube C if one of the two criteria holds. Termination and correctness follow immediately from the above discussion. To improve on the above size bound, we bound the total number of canonical cubes that the search visits, each of which taking $O(d \cdot k)$ time.

▶ **Definition 3** (Distance Classes)**.** *Let $ball_s(x) = \{p : \|s-p\| \leq x\}$ be the ball of radius x around site s. Let L be the set of canonical cubes that our top-down search,* Adaptive Refinement*, visits. We partition $L =: \bigcup_j L_j$ in distance classes, such that L_j contains those cubes $C \in L$ where $C \subseteq ball_s(2^j r)$ and $C \nsubseteq ball_s(2^{j-1} r)$.*

Note that $L_j = \emptyset$ for $j \leq -2$, since such a cube C would be contained in $ball_s(r/4)$. Consequently, its parent C' would be contained in $ball_s(r/2)$, satisfying the inclusion-test criteria that stops the search. Thus, there are $O(\log(1/\varepsilon_S))$ non-empty distance classes.

We use Stirling's formula to bound the volume of the Euclidean d-ball of radius 1

$$\kappa_d = \mathrm{Vol}(ball(1)) \in \left[\frac{\pi^{d/2}}{\lceil d/2 \rceil !}, \frac{\pi^{d/2}}{\lfloor d/2 \rfloor !}\right] = \Theta_d(d^{-(d+1)/2}) \ . \tag{5}$$

▶ **Lemma 4** (Simple Bound)**.** *There are $O_d(1/\varepsilon_A^d)$ canonical cubes in class L_j.*

Proof. All cubes of distance class L_j are contained in the d-ball around s with radius $2^j r$, which has the volume $\mathrm{Vol}(ball_s(2^j r)) = \kappa_d \cdot (2^j r)^d = O_d((2^j r)^d / d^{(d+1)/2})$. Thus, it suffices to show that any one cube has side-length at least $\varepsilon_A 2^j r/(8\sqrt{d})$.

From the distance class partition, we have that a cube with diameter δ in class j has that all of its points have distance $\geq 2^j r/2 - \delta$ from the center s.

Now, having the top-down search visit a cube C with diameter δ would require the search did not terminate at its parent C', which has diameter 2δ. Thus, 2δ was not sufficiently small for stopping, i.e. $2\delta > \varepsilon_A(\|s_i - \mathit{centr}(C')\| - 2\delta/2)$. Since $\mathit{centr}(C') \in C$, its distance from s_i is at least $2^j r/2 - \delta$. Hence, $2\delta > \varepsilon_A(2^j r/2 - 2\delta)$, which implies that $\delta > \frac{\varepsilon_A}{1+\varepsilon_A} \cdot 2^j r/4$.

Thus, any cube in L_j must have diameter $\geq \varepsilon_A \cdot 2^j r/8$, and consequently side-length $\geq \varepsilon_A \cdot 2^j r/(8\sqrt{d})$. ◀

Thus, the lemma yields a running time bound and, consequently, a result size bound. In the full paper, we show that this bound can be improved by one $(1/\varepsilon_A)$-factor.

We summarize our results thus far before discussing how to assemble the Approximate Voronoi Diagram from the ε_A-approximations of the cores.

▶ **Theorem 5.** *Let $\mathcal{R} \subseteq \mathbb{R}^d$ be a region that is the intersection of k bisectors of $O(1/\varepsilon_S)$-fatness, s its center, and $\varepsilon_A \in (0, \varepsilon_S)$. One can compute a set L of $O_d\big(\log(1/\varepsilon_S)/\varepsilon_A^{d-1}\big)$ canonical cubes that ε_A-approximates $(\mathcal{R}, s)$. Time is an $O(d \cdot k)$-factor over the size bound.*

Our lower bound in Theorem 7 will show that $\Omega_d\left(\frac{\log(1/\varepsilon)}{\varepsilon^{d-1}}\right)$ cubes are required, if $\varepsilon \ll 1/d^3$.

3.1 Assembling the Approximate Diagram from Cubes

In this section, we combine the ε_A-approximations of each of the regions $\mathit{core}(B_i)$ to construct an ε-AMWVD, where $\varepsilon = (1 + \varepsilon_S)(1 + \varepsilon_A) - 1$. For each $1 \leq i < n$, we construct the ε_A-approximate cubes for $(\mathit{core}(B_i), s_i)$ using Theorem 5. Each cube in the ε_A-approximation of $(\mathit{core}(B_i), s_i)$ is given the label i. We collect all cubes for all labels $1 \leq i < n$ in a list. For $i = n$, we construct a canonical cube that contains all other canonical cubes for $1 \leq i < n$, and give this canonical cube the label n and also add it to the list. (This cube will be at the root of the compressed QuadTree.)

Sort the list of canonical cubes by their z-order. To remove duplicate cubes, iterate over the sorted list and keep only the cube with the minimum label (from those that are identical cubes). Construct a compressed QuadTree from this set of canonical cubes using, say, the Divide&Conquer approach (see Lemma 2.11 in [13]). The leaves of the compressed QuadTree induces a subdivision of $\mathbb{R}^d$, where each cell in the subdivision is either a canonical cube, or the set difference of at most 2^d canonical cubes.

Finally, we label all cells in the compressed QuadTree as follows. The cubes that are from the the sorted list have their initial label, and the root has initial label n. Starting at the root, if a child is unlabeled, or the child has larger label than its parent, then the child replaces its label with its parent's label. We repeat this process for all nodes in the compressed QuadTree in top-down fashion, say in a DFS traversal. This completes the construction of the approximate Voronoi Diagram.

To answer approximate (weighted) nearest-neighbor queries, given a query point $q \in \mathbb{R}^d$, we search our QuadTree for the smallest canonical cube containing q. The weighted nearest-neighbor of q is the site with index equal to the label stored at this node. Recall that point-location time in a compressed QuadTree is $O(d \log N)$ where N is the number of cubes in the tree.

Next, we prove the correctness of our proposed construction. When querying with a point q, we have two cases: Either the label returned is n, or it is less than n. If the label is n, then by construction, q is in none of the ε_A-approximations of $(\mathit{core}(B_i), s_i)$, for any $1 \leq i < n$. Therefore, q is outside the ε_A-approximation of $\mathit{core}(B_i)$ for all $1 \leq i < n$, so s_n is indeed the site with the smallest weighted distance to q, up to a factor of $(1+\varepsilon)$. Otherwise, let the label be i, for some $1 \le i < n$. Due to the top-down propagation, we know that there is no canonical cube in the sorted list that both, contains q and has label less than i. Therefore, q is outside the ε_A-approximations of $core(B_j)$ for all $j < i$. So q has smaller (weighted) distance to s_i than any of $\{s_1, \ldots, s_i\}$, up to a factor of $(1+\varepsilon)$. Moreover, we know that q is in the ε_A-approximation of $core(B_i)$. Therefore, up to a factor of $(1+\varepsilon)$, q has smaller weighted distance to s_i than any of $\{s_i, \ldots, s_n\}$.

Since the time for top-down label propagation is linear in the tree size, our construction time bound is one logarithmic factor over the size bound:

▶ **Theorem 6.** *Given $\varepsilon_S > \varepsilon_A > 0$ and a set of balls B_i for each $i < n$, one can compute an ε-approximate Voronoi Diagram, where $\varepsilon = (1+\varepsilon_S)(1+\varepsilon_A) - 1$, with total size $O_d(n \log(1/\varepsilon_S)/\varepsilon_A^{d-1})$. The construction time is $O_d\Big(\log \frac{n}{\varepsilon_A} + n^{-1} \sum_i |B_i|\Big)$ times the size bound. Moreover, time to locate a query-point is $O(d \log(n) + d^2 \log(1/\varepsilon_A))$.*

This theorem will be used as a tool in Section 5, where we improve the construction time to near-linear, using our efficient construction of a bisector coreset for the $\{B_i\}$. Note that the construction time is already quadratic in n, since $|B_i| < n$ for all i. Next, we show that the result size is optimal, up to $\Theta_d(1)$ factors.

4 A Matching Lower Bound for Diagram Size

In this section, we show our matching lower bound for the size of ε-AMWVDs. That is, any subdivision comprised of axis-aligned hyper-rectangles requires $\Omega_d(n \cdot \log(1/\varepsilon)/\varepsilon^{d-1})$ cells. Our MWVD instances consist of n copies of a two-site instance that are placed sufficiently far from each other. The main idea for the two-site instance is that there are $\Omega(\log 1/\varepsilon)$ distinct regions of space, each of which having a 'large' total volume but having a geometric shape that only allows to cover a relative 'small' volume with any one cell. Though the basic approach is similar to the $\Omega_d(n \cdot \varepsilon/(\sqrt{d}\varepsilon)^d)$ lower bound in [4, Section 5], the difference is that that our argument addresses various sections of two Apollonian balls with curvatures $\Theta(\varepsilon)$, instead of one hyper-cylinder that is bounded by two parallel hyper-planes. This results in a bound that is stronger by a $(d^{(d-1)/2} \log \frac{1}{\varepsilon})$-factor than the known bound for unit-weight ε-AVDs, and matches our upper bound in Theorem 5 up to $\Theta_d(1)$-factors.

Though it is an intriguing problem to also settle the question of optimal complexity for unit-weight ε-AVDs, it is, unfortunately, quite unclear if one can obtain such a bound without curved MWVD bisectors. (Cf. last two paragraphs of Section 8 in [5].)

▶ **Theorem 7.** *Let $\varepsilon \in (0, 1/16d^3]$, $w_I = 1$, $w_O = (1+\varepsilon)^2$, and $B := ball(s_I, s_O, w_O/w_I)$ be the Apollonian ball of $s_I = (-1/\sqrt{d}, \ldots, -1/\sqrt{d})$ and $s_O = ((1+\varepsilon)^2/\sqrt{d}, \ldots, (1+\varepsilon)^2/\sqrt{d})$. Any subdivision of $\mathbb{R}^d$ in axis-aligned hyper-rectangles that is an ε-approximation of the MWVD bisector B must contain $\Omega_d\big(\log(1/\varepsilon)/\varepsilon^{d-1}\big)$ cells.*

Proof. Any ε-approximation of the MWVD of B must assign the points inside $B_I := ball(s_I, s_O, (1+\varepsilon)^3)$ to site s_I and outside $B_O := ball(s_I, s_O, 1+\varepsilon)$ to site s_O, i.e. only the points in $B_O \setminus B_I$ may be labeled with either site. Thus, any one cell c in an ε-approximation must not intersect both, B_I and $\mathbb{R}^d \setminus B_O$. Note that $B_I \subset B \subset B_O$ and the sites, as well as the centers m_I and m_O, are co-linear, i.e. on the main diagonal. From (3) and (4), we have that $t^* = 1$ and that t^* and $t^\dagger$ have the relations

$$t_I^\dagger(1+\varepsilon) = t^\dagger = t_O^\dagger/(1+\varepsilon)$$

$$t_I^*(1+\varepsilon) = t^* = t_O^*/(1+\varepsilon) \ ,$$

which shows that their radii, i.e. $r = (t^* + t^\dagger)/2$, have relation $r_I(1+\varepsilon) = r = r_O/(1+\varepsilon)$. The radii are $\Theta(1/\varepsilon)$.

Let w.l.o.g. the t_I^* point on B_I be at the origin. Let A contain the points from the upper half-space of $B_O \setminus B_I$, where upper/lower is due to a fixed hyper-plane that contains the main diagonal. Define partition $A =: \cup_i A_i$ such that the points in A_i have a norm in range $(2^i, 2^{i+1}]$, and let A_{-1} have the points with norm ≤ 1. We prove the following three claims in the full version of the paper.

▷ Claim 8. Let $A = B_O \setminus B_I$. The i-th section $A_i = \{x \in A : \|x\| \in (2^i, 2^{i+1}]\}$ has volume at least $\mathrm{Vol}(A_i) \geq \varepsilon 2^i \cdot \kappa_{d-1} 2^{(i+1)(d-1)-1} = \Omega_d(\varepsilon 2^{di}/d^{d/2})$.

▷ Claim 9. Any axis-aligned hyper-rectangle c, which does not contain points from B_I, can cover a volume of at most $\mathrm{Vol}(c \cap A_i) = O_d((\varepsilon 2^i)^d/d^{(d+1)/2})$.

▷ Claim 10. Let $\varepsilon \in (0, 1/d^3]$. Any axis-aligned hyper-rectangle c, which does not contain points from $\mathbb{R}^d \setminus B_O$, can cover a volume of at most $\mathrm{Vol}(c \cap A_i) = O_d((\varepsilon 2^i)^d/d^{(d+1)/2})$, provided index $i \leq \frac{5}{4}\log_2(1/\varepsilon)$.

Thus, $\Omega_d\left(\frac{\varepsilon 2^{di}/d^{d/2}}{(\varepsilon 2^i)^d/d^{(d+1)/2}}\right) = \Omega_d(\sqrt{d}/\varepsilon^{d-1})$ hyper-rectangles are necessary to cover any of the $\Omega(\log 1/\varepsilon)$ many sections from A. ◀

5 Approximate Cores: Computing Bisector Coresets Efficiently

Next, we define the notion ε-approximation that we use for the proof (Section 5.2) of the quality guarantee for the algorithm in Section 5.1. It extends the intuitive idea that 'large balls' in the set B_i may not be relevant for the intersection that defines $core(B_i)$.

Let α-$ball(i,j)$ denote the enlarged ball that is obtained by setting the effective weight to $w_j/\alpha w_i$ in the bisector, i.e. $\alpha\text{-}ball(i,j) = ball(s_i, s_j, \gamma_{ij}/\alpha)$. For $\alpha \geq 1$, we define a relation between any two subsets $X, Y \subseteq B_i$ from the bisectors of s_i as

$$X \prec_\alpha Y \quad \Longleftrightarrow \quad \forall\ (i,k) \in Y \ : \ core(X) \subseteq \alpha\text{-}ball(i,k) \ ,$$

and say for such a pair that X is an α-cover of Y. Given a subset $X \subseteq B_i$, we call the largest subset $Y \subseteq B_i$ with $X \prec_\alpha Y$ the set of balls that are α-covered by X. Further, X is called an α-cover if it covers *all balls* in B_i, i.e. $X \prec_\alpha B_i$, and we have

$$core(B_i) \quad \subseteq \quad core(X) \quad \subseteq \quad \alpha\text{-}core(B_i) := \bigcap_{(i,j)\in B_i} \alpha\text{-}ball(i,j) \quad . \tag{6}$$

For example, the set of balls that are 1-covered by a singleton set $\{(i,j)\}$ contains all balls $(i,k) \in B_i$ with $ball(i,j) \subseteq ball(i,k)$. Note that $X \prec_\alpha Y$ and $Y \prec_{\alpha'} Z$ implies $X \prec_{\alpha\cdot\alpha'} Z$. Clearly, using α-covers $\{A_1, \ldots, A_{n-1}\}$ of the bisectors (i.e. $A_i \prec_\alpha B_i$ for all sites s_i) turns the ε-approximation algorithm of Section 3.1 into one that computes an ε'-approximate Voronoi Diagram, with $\varepsilon' = (1+\varepsilon)\alpha - 1$, whose running time is sensitive to $|A_i|$.

The goal of our next algorithm is to compute a subsets $A_i \subseteq B_i$, so that A_i is an α-cover of B_i, and A_i has *constant size*. Then, we apply Theorem 6 to those bisector sets $\{A_i\}$.

Recap: σ-Semi-Separated Pair Decompositions with Low Weight

Let $S \subseteq \mathbb{R}^d$ be a set of n points. A list of subset pairs $\mathcal{P} = \{(X_i, Y_i) : X_i, Y_i \subseteq S,\ X_i \cap Y_i = \emptyset\ \}$ is called a pair pecomposition if there is, for every $\{s, s'\} \in \binom{S}{2}$, a pair $(X_i, Y_i) \in \mathcal{P}$ with $|\{s, s'\} \cap X_i| = 1 = |Y_i \cap \{s, s'\}|$. The quantity $\sum_i (|X_i| + |Y_i|)$ is called the *weight* of the pair decomposition $\mathcal{P}$. It is well known that a pair decomposition of n points has weight $\Omega(n \log n)$. (See [13, Lemma 3.31].)

A pair decomposition $\mathcal{P}$ of S is called a σ-SSPD with respect to constant $\sigma > 1$, if every point set pair $(X, Y) \in \mathcal{P}$ has the separation property

$$\min \left\{ \max_{x,x' \in X} \|x - x'\|_2\ ,\ \max_{y,y' \in Y} \|y - y'\|_2 \right\} \cdot \sigma \leq \min_{x \in X, y \in Y} \|x - y\|_2\ . \tag{7}$$

That is, the two sets have a closest-pair distance of at least σ times the small diameter.

Given a set of n points from $\mathbb{R}^d$, a σ-SSPD with weight $w(n, d, \sigma) = O_d(d^{7d/2}\sigma^d n \log n) = O_D(\sigma^d n \log n)$ can be computed in deterministic $O_D(\sigma^d n + n \log n)$ time [1, Theorem 5]. For point sets with polynomially bounded spread, it is possible to improve both (deterministic) O_D-bounds to O_d-bounds with a QuadTree based pair decomposition, using [2, Lemma 2.8].

The efficiency of our coreset construction stems from low weight SSPDs. We use the SSPD separation in terms of the radius of the two sets, which increases σ by a factor of two.

5.1 Computing Approximate Cores: SSPDs and Conic Space Partitions

Let $\beta, \varepsilon_C > 0$ and $\sigma \geq 2$ be constants, which we calibrate in Section 5.2. A β-cone around s_i is an angular domain of the spherical coordinate system around s_i. Each of its $(d-1)$ angular dimensions is partitioned into intervals of at most 2β radians. For each s_i, we assign each β-cone a unique array index j, where $j = O_d(1/\beta^{d-1})$. E.g. a rotation of at most β radians suffices to rotate any point in the cone onto the cone's central ray.

Let $\mathcal{P}$ be a σ-SSPD of the input sites S. For a pair $(L, H) \in \mathcal{P}$, we call L the 'light set' and H the 'heavy set' if s_ℓ is the site with maximum index in L, s_h is the site with the maximum index in H, and $\ell < h$.

Our algorithm maintains the following structure: For each site $s_i \in S$, and for each β-cone around s_i with array index j, the data structure stores a set of partner sites A_{ij}. Our algorithm populates the structure in three passes. In our first pass, for each $(L, H) \in \mathcal{P}$, we reduce the size of H to a subset H'. In our second pass, we iterate over $\mathcal{P}$ to initialize each of the sets A_{ij}. Finally, the sets are populated in the third pass.

In our first pass, for each $(L, H) \in \mathcal{P}$, we construct a subset H' of H. If the diameter of H is at most the diameter of L, we set $H' := \{\}$. If the diameter of H is larger than the diameter of L, we construct H' as follows. Let $s_\ell \in L$ with ℓ maximal. For the j^{th} cone around s_ℓ, we let the sites of H contained in this cone be $C_{\ell j}$. We use the following function:

```
Scan-Cone-Sites(i, C, ε_C):
    Let C' := ∅, a = min{t*_ij : s_j ∈ C}, and b = min{t†_ij : s_j ∈ C}.
    Let I_k = (x_k, x_{k+1}], with length aε_C/2 and x_1 = a, cover [a,b].
    Every interval I_k holds one pointer.
    FOR s_j ∈ C DO
        Compute the index k with t*_ij ∈ I_k.
        If diameter (t*_ij + t†_ij) is smaller than that of I_k's reference,
            then set I_k's pointer on s_j.
    FOR interval I_k DO
        Add the kept bisector to result set C'.
    return C'
```

We select for the j^{th} cone a subset by setting $C'_{\ell j} :=$ Scan-Cone-Sites$(\ell, C_{\ell j}, \varepsilon_C)$ and define $H' = \cup_j C'_{\ell j}$. This completes the construction of H'.

In our second pass, we initialize each cone of each site in our structure to store an interval $[a, b]$. We iterate over all pairs $(L, H) \in \mathcal{P}$ and all $s_i \in L \cup H$, and store for j^{th} cone of s_i, a variable a equal to the minimum value of a t^*_{ik}, and a variable b equal to the minimum value of a $t^\dagger_{ik}$. This minimum is taken over all sites $s_k \in H' \cup \{s_\ell, s_h\}$ that are in the j^{th} cone of s_i and have $k > i$. This gives us the interval $[a, b]$. After the pass over $\mathcal{P}$ is completed, we iterate over each cone of each site and partition the interval $[a, b]$ into disjoint intervals $I_k = (x_k, x_{k+1}]$ of length $a\varepsilon_C/2$ that cover $[a, b]$, i.e. $x_{k+1} - x_k = a\varepsilon_C/2$ and $x_1 = a$.

In our third pass, we populate the sets A_{ij} based on the intervals $\{I_k\}$ of the j^{th} cone of s_i. We iterate over all pairs $(L, H) \in \mathcal{P}$ and maintain a reference from I_k to the site that realized a minimum diameter. For $s_i \in L \cup H$, and for the j^{th} cone around s_i, we let the sites $s_m \in H' \cup \{s_\ell, s_h\}$ with $m > i$ that are contained in this cone be C_{ij}. For each $s_m \in C_{ij}$, we locate the interval I_k of the cone that contains t^*_{im} and compare the diameter of $ball(i, m)$ with the smallest diameter of I_k that we have encountered so far. If the diameter of $ball(i, m)$ is smaller, we set s_m to be the site of I_k realizing the minimum diameter. After the pass over all pairs is completed, for the j^{th} cone of site s_i, and for all intervals I_k, we add the site that realized the minimum diameter for I_k into the set A_{ij}. This completes our three passes that construct the cone sets. Finally, we set $A_i = \cup_j A_{ij}$, and then apply Theorem 6 to the set of balls A_i.

In the next section, we show that A_i is an α-cover of B_i. The algorithm's runtime bound $O_d(w(n, d, \sigma) \cdot m/\beta^{d-1})$ will follow from weight $w(n, d, \sigma)$ of a σ-SSPD, the number of β-cones in the partitions of $\mathbb{R}^d$, and the maximum number m of sites in the sets A_{ij}.

5.2 Correctness: Choosing Sufficient β, σ, and ε_C

Our $(1+\varepsilon)$ bound consists of seven components for each of the convex cores. The components use the target approximation ε_A for the Adaptive Refinement in Section 3, an ε_S that scales half-space bisectors to sufficiently large balls (see Section 2.1), an ε_C that is the tolerance for selecting a small set of sites per β-cone, an ε_T that virtually translates sites along a ray from another site, and ε_R that virtually rotate a site's partner (cf. Figure 2).

For prescribed $\varepsilon > 0$, we set the components such that

$$(1+\varepsilon_A)(1+\varepsilon_S)(1+\varepsilon_T)(1+\varepsilon_R)^2(1+\varepsilon_C)^2 \;\le\; 1+\varepsilon \tag{8}$$

$$\max\{\varepsilon_R, \varepsilon_T, \varepsilon_C\} \;<\; \varepsilon_S \;, \tag{9}$$

where the last inequality is *strict* to accommodate Lemma 13. For example, we can set $\varepsilon_S = \varepsilon/8$ and $\varepsilon_A = \varepsilon_C = \varepsilon_R = \varepsilon_T = \varepsilon/16$.

This section shows $core(A_i) \subseteq \left(\frac{1+\varepsilon}{1+\varepsilon_A}\right)$-$core(B_i)$ and consequently the approximation bound of our approach. Recall from Section 2.1 that all bisectors in B_i have $w_j/w_i \ge 1+\varepsilon_S$.

To show inclusion properties, we will use the following parametrization of balls in B_i: Consider a fixed ray q, say the x-axis, that emanates from the origin s_i, having $w_i = 1$. Ignoring the input instance S briefly, any pair (s, w) of a point s on q and a real $w > 1$ defines a ball, with respective two points on the x-axis of distance $t^*, t^\dagger > 0$. It is convenient to use parametrization $(t^*, t^\dagger)$, instead of (s, w), to describe this ball: If input sites s_j and s_k are on the same ray q, then $ball(i, j) \subseteq ball(i, k) \;\Leftrightarrow\; t^*_{ij} \le t^*_{ik} \wedge t^\dagger_{ij} \le t^\dagger_{ik}$. It is noteworthy that both inequalities can be decided without square-root computations (cf. Eq. (3) and (4)).

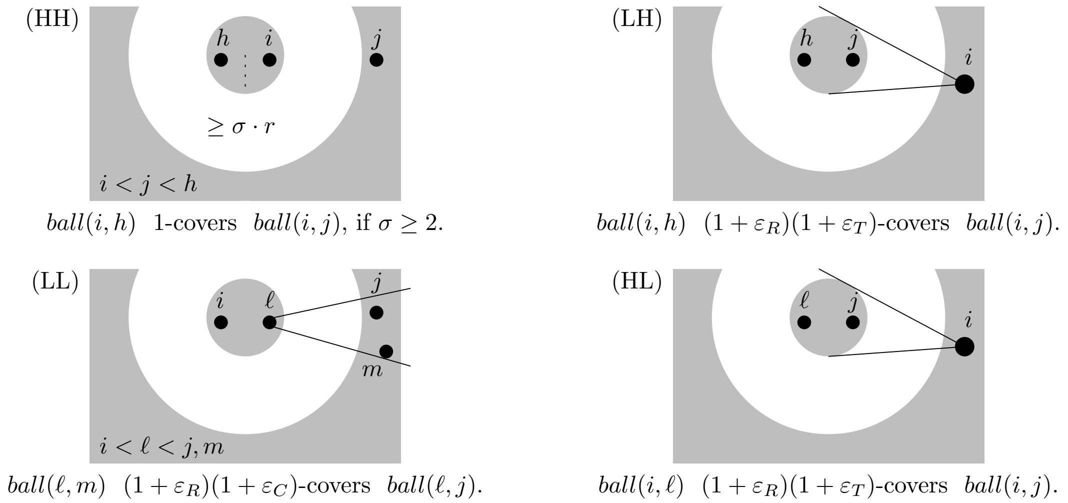

Figure 2 Cases (HH), (LH), (HL), and (LL), for covering an absent ball $(i, j) \in B_i \setminus A_i$.

To show that every $(i, j) \in B_i \setminus A_i$ is α-covered, the main idea is to consider the pair $(L, H) \in \mathcal{P}$ that separates it to observe that at least one bisector that α-covers (i, j) is contained in A_i. There are four cases for an absent bisector (i, j): (LL) $s_i \in L$ and L has smaller diameter, (LH) $s_i \in L$ and H has smaller diameter, (HL) $s_i \in H$ and L has smaller diameter, and (HH) $s_i \in H$ and H has smaller diameter. We use at most three affine transformations to bound each case. See Figure 2. The bound for (LL) is $\alpha = (1 + \varepsilon_R)^2(1 + \varepsilon_C)^2(1 + \varepsilon_T)$, the bound for (LH) is $\alpha = (1 + \varepsilon_R)^2(1 + \varepsilon_T)(1 + \varepsilon_C)$, the bound for (HL) is $\alpha = (1+\varepsilon_R)^2(1+\varepsilon_T)(1+\varepsilon_C)$, and the bound for (HH) is $\alpha = (1+\varepsilon_R)(1+\varepsilon_C)$.

We start by showing an observation about pair decompositions. A cluster of points H that is, relative to its diameter, far from a given point s_i can be rotated with a small angle onto a common ray q, from s_i through an arbitrary point s_h from the cluster.

▶ **Observation 11** (Distant clusters). *Let angle $\beta \in (0, 1]$, $s_i \in S$, c and r be the center and radius of the minimum enclosing ball of $H \subseteq S \setminus \{s_i\}$, $\sigma := \|s_i - c\|/r > 0$, and $s_h \in H$. If $\sigma \geq 2/\beta$, then $\angle s' s_i s_h \in [0, \beta]$ for all $s' \in H$.*

Proof. Since $\frac{r}{r\sigma} = \tan\frac{\beta}{2} = \frac{\sin\beta}{1+\cos\beta} \leq \frac{\beta}{1+(1-\beta^2)} = \frac{1}{2/\beta-\beta}$ and $\beta \geq 0$, any $\sigma \geq 2/\beta$ suffices. ◀

This observation motivates our main lemma that analyzes the enlargement of a ball from B_i that is required to contain the ball that is obtained from a small rotation around s_i.

▶ **Lemma 12** (Rotations at s_i). *If $\beta = \angle s_j s_i s_k \in [0, 1]$ and $[t^*_{ij}, t^\dagger_{ij}] = [t^*_{ik}, t^\dagger_{ik}]$, then $ball(i, j) \subseteq \alpha\text{-}ball(i, k)$ for all $\alpha \geq 1 + \beta^2/2$.*

Note that this bound also applies to rotations of s_i on s_j around s_k for $k > i, j$, i.e. if $[t^*_{ik}, t^\dagger_{ik}] = [t^*_{jk}, t^\dagger_{jk}]$ and $\beta = \angle s_i s_k s_j$ is small, then $B \subseteq \alpha\text{-}ball(j, k)$, where B is the translation of $ball(i, k)$ with the vector $\overrightarrow{s_i s_j}$.

So far we showed that choosing a cone angle $\beta = \sqrt{2\varepsilon_R}$ and $\sigma \geq 2/\sqrt{2\varepsilon_R}$ satisfies the $(1 + \varepsilon_R)$-factors for rotations in all cases (i.e. LL, LH, HL, and HH). Next we show that translations of sites in the low diameter set have a $(1 + \varepsilon_T)$-bound, for sufficiently large σ.

▶ **Lemma 13** (Translations). *Let p and q be on a common ray from s, $\|s-p\| < \|s-q\|$, $\varepsilon_T \in (0, \varepsilon_S)$, point $m := (p+q)\frac{1}{2}$, $r := \|m-p\|$. If $1+\varepsilon_T < \gamma$, then we have that $\|s-m\| \geq \sigma r$ implies that $t^*(s,q,\gamma) \leq t^*\left(s,p,\frac{\gamma}{1+\varepsilon_T}\right)$ and $t^\dagger(s,q,\gamma) \leq t^\dagger\left(s,p,\frac{\gamma}{1+\varepsilon_T}\right)$, for all $\sigma \geq 1 + 2/\varepsilon_T$. This also implies that $t^*(q,s,\gamma) \leq t^*\left(p,s,\frac{\gamma}{1+\varepsilon_T}\right)$ and $t^\dagger(q,s,\gamma) \leq t^\dagger\left(p,s,\frac{\gamma}{1+\varepsilon_T}\right)$.*

The first translation property is used for the cases where H has smaller diameter and the second for the cases where L has smaller diameter. One may think of the above discussion as a means to virtually place all sites in the low diameter set at the same spatial point with two transformations. We now show that partners of s_i with lower weight than other partners, transformed to the same location, can be ignored in an α-cover (e.g. Figure 2 LH and HL).

▶ **Observation 14** (Weight Monotonicity). *If $1 < \gamma \leq \gamma'$, then $ball(p,q,\gamma') \subseteq ball(p,q,\gamma)$.*

Proof. We give the, slightly more technical, argument for $t^\dagger(p,q,\gamma') \leq t^\dagger(p,q,\gamma)$. This holds iff $\frac{\|p-q\|}{\gamma'-1} \leq \frac{\|p-q\|}{\gamma-1}$, which holds iff $\gamma' - 1 \geq \gamma - 1$, since $\gamma - 1 \neq 0 \neq \gamma' - 1$. ◀

Thus, for case (LH) and (HL) it suffices that s_i scans s_h and s_ℓ, respectively. (They are member of $H' \cup \{s_\ell\} \cup \{s_h\}$ and checked by algorithm when pair (L, H) is considered.) It remains to prove the $(1 + \varepsilon_C)$ factor in the approximations of Scan-Cone-Sites.

▶ **Lemma 15** (Constant per cone). *Let $\{s_2, \ldots, s_n\}$ be on a common ray from s_1, $w_i/w_1 \geq 1 + \varepsilon_S$, and $\varepsilon_S > \varepsilon_C > 0$. Computing a $C_1 \subseteq B_1$ of size $O(1/\varepsilon_C\varepsilon_S)$, with $C_1 \prec_{(1+\varepsilon_C)} B_1$, takes $O(n)$ time.*

Thus, selecting at most $m = O(1/\varepsilon_C^2)$ sites per cone introduces only a factor of $(1 + \varepsilon_C)$. This completes the argument for all four cases, and we have $core(A_i) \subseteq \frac{1+\varepsilon}{1+\varepsilon_A}$-$core(B_i)$. Taking $\sigma = 1 + 2/\varepsilon_T$, $\beta = \sqrt{2\varepsilon_R}$, and $m = O(\varepsilon_C^{-2})$, the coreset construction time $O_d(w(n,d,\sigma) \cdot m/\beta^{d-1}) = O_D((\varepsilon^{-d} n \log n) \cdot \varepsilon^{-2}\varepsilon^{-(d-1)/2}) = O_D(n \log(n)/\varepsilon^{3(d+1)/2})$. We summarize:

▶ **Theorem 16.** *The approximation algorithm computes, for each $1 \leq i < n$, a subset $A_i \subseteq B_i$ with $core(A_i) \subseteq \frac{1+\varepsilon}{1+\varepsilon_A}$-$core(B_i)$ and $|A_i| = O_d(1/\varepsilon^{(d+3)/2})$ in $O_D(n \log(n)/\varepsilon^{3(d+1)/2})$ time.*

We are now ready to show our main result.

▶ **Corollary 17.** *For any $\varepsilon > 0$, one can compute an ε-AMWVD of size $O_d(n \log(1/\varepsilon)/\varepsilon^{d-1})$. The construction time is $O_D(\log(n)/\varepsilon^{(d+5)/2})$ times the output size.*

The query time of the search structure is $O(d \log(n) + d^2 \log(1/\varepsilon))$.

Proof. Applying Theorem 6 on the bisector coresets that are obtained from Theorem 16, the construction time of the ε-AMWVD is a factor $O_d\left(|A_i| + \log(n/\varepsilon)\right) = O_d\left(\log(n/\varepsilon)/\varepsilon^{(d+3)/2}\right)$ over the output size bound. Hence, construction time is dominated by computing the bisector coreset, taking a factor $O_D\left(\frac{n\log(n)/\varepsilon^{3(d+1)/2}}{n\log(1/\varepsilon)/\varepsilon^{d-1}}\right) = O_D(\varepsilon^{-(d+5)/2}\log(n)/\log(1/\varepsilon))$ over the output size bound. ◀

References

1. Mohammad Ali Abam, Mark de Berg, Mohammad Farshi, Joachim Gudmundsson, and Michiel H. M. Smid. Geometric spanners for weighted point sets. *Algorithmica*, 61(1):207–225, 2011. doi:10.1007/s00453-010-9465-2.
2. Mohammad Ali Abam and Sariel Har-Peled. New constructions of SSPDs and their applications. *Comput. Geom.*, 45(5-6):200–214, 2012. doi:10.1016/j.comgeo.2011.12.003.
3. Boris Aronov, Gali Bar-On, and Matthew J. Katz. Resolving SINR queries in a dynamic setting. *SIAM J. Comput.*, 49(6):1271–1290, 2020. doi:10.1137/19M128733X.
4. Sunil Arya and Theocharis Malamatos. Linear-size approximate Voronoi diagrams. In *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms (SODA'02)*, pages 147–155, 2002. URL: http://dl.acm.org/citation.cfm?id=545381.545400.
5. Sunil Arya, Theocharis Malamatos, and David M. Mount. Space-time tradeoffs for approximate nearest neighbor searching. *J. ACM*, 57(1):1:1–1:54, 2009. doi:10.1145/1613676.1613677.
6. Franz Aurenhammer. The one-dimensional weighted Voronoi diagram. *Information Processing Letters*, 22(3):119–123, 1986. URL: https://www.sciencedirect.com/science/article/pii/0020019086900554, doi:https://doi.org/10.1016/0020-0190(86)90055-4.
7. Franz Aurenhammer and Herbert Edelsbrunner. An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recognit.*, 17(2):251–257, 1984. doi:10.1016/0031-3203(84)90064-5.
8. Franz Aurenhammer, Rolf Klein, and Der-Tsai Lee. *Voronoi Diagrams and Delaunay Triangulations*. World Scientific, 2013. doi:10.1142/8685.
9. Bernard Chazelle. Filtering search: A new approach to query-answering. *SIAM J. Comput.*, 15(3):703–724, 1986. doi:10.1137/0215051.
10. Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational geometry: algorithms and applications, 3rd Edition*. Springer, 2008. URL: https://www.worldcat.org/oclc/227584184.
11. Chenglin Fan and Benjamin Raichel. Linear expected complexity for directional and multiplicative Voronoi diagrams. In *Proc. 28th European Symposium on Algorithms (ESA'20)*, pages 45:1–45:18, 2020. doi:10.4230/LIPIcs.ESA.2020.45.
12. Sariel Har-Peled. A replacement for Voronoi diagrams of near linear size. In *Proc. 42nd Symposium on Foundations of Computer Science (FOCS'01)*, pages 94–103, 2001. doi:10.1109/SFCS.2001.959884.
13. Sariel Har-Peled. *Geometric approximation algorithms*. Number 173 in Mathematical Surveys and Monographs. American Mathematical Society, 2011.
14. Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory Comput.*, 8(1):321–350, 2012. doi:10.4086/toc.2012.v008a014.
15. Sariel Har-Peled and Nirman Kumar. Approximating minimization diagrams and generalized proximity search. *SIAM J. Comput.*, 44(4):944–974, 2015. doi:10.1137/140959067.
16. Sariel Har-Peled and Benjamin Raichel. On the complexity of randomly weighted multiplicative Voronoi diagrams. *Discret. Comput. Geom.*, 53(3):547–568, 2015. doi:10.1007/s00454-015-9675-0.
17. Martin Held and Stefan de Lorenzo. An efficient, practical algorithm and implementation for computing multiplicatively weighted Voronoi diagrams. In *Proc. 28th European Symposium on Algorithms (ESA'20)*, pages 56:1–56:15, 2020. doi:10.4230/LIPIcs.ESA.2020.56.
18. Yogish Sabharwal, Nishant Sharma, and Sandeep Sen. Nearest neighbors search using point location in balls with applications to approximate Voronoi decompositions. *J. Comput. Syst. Sci.*, 72(6):955–977, 2006. doi:10.1016/j.jcss.2006.01.007.