

Voronoi Diagrams in Higher Dimensions under Certain Polyhedral Distance Functions*

Jean-Daniel Boissonnat[†] Micha Sharir[‡] Boaz Tagansky[§]
Mariette Yvinec[¶]

March 16, 1996

Abstract

The paper bounds the combinatorial complexity of the Voronoi diagram of a set of points under certain polyhedral distance functions. Specifically, if \mathcal{S} is a set of n points in general position in \mathbb{R}^d , the maximum complexity of its Voronoi diagram under the L_∞ metric, and also under a simplicial distance function, are both shown to be $\Theta(n^{\lceil d/2 \rceil})$. The upper bound for the case of the L_∞ metric follows from a new upper bound, also proved in this paper, on the maximum complexity of the union of n axis-parallel hypercubes in \mathbb{R}^d . This complexity is $\Theta(n^{\lceil d/2 \rceil})$, for $d \geq 1$, and it improves to $\Theta(n^{\lfloor d/2 \rfloor})$, for $d \geq 2$, if all the hypercubes have the same size. Under the L_1 metric, the maximum complexity of the Voronoi diagram of a set of n points in general position in \mathbb{R}^3 is shown to be $\Theta(n^2)$. We also show that the general position assumption is essential, and give examples where the complexity of the diagram increases significantly when the points are in degenerate configurations. Finally, on-line algorithms are proposed for computing the Voronoi diagram of n points in \mathbb{R}^d under a simplicial or L_∞ distance function. Their randomized complexities

*Work on this paper by J.D. Boissonnat and M. Yvinec has been partly supported by the ESPRIT Basic Research Action Nr. 7141 (ALCOM II). Work on this paper by Micha Sharir has been supported by NSF Grants CCR-91-22103 and CCR-93-11127, by a Max-Planck Research Award, and by grants from the U.S.-Israeli Binational Science Foundation, the Israel Science Fund administered by the Israeli Academy of Sciences, and the G.I.F., the German-Israeli Foundation for Scientific Research and Development.

[†]INRIA, 2004 Route des Lucioles, B.P.109, 06561 Valbonne cedex, France; E-mail: Jean-Daniel.Boissonnat@sophia.inria.fr

[‡]School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel and Courant Institute of Mathematical Sciences, New York University, New York NY 10012, USA; E-mail: sharir@math.tau.ac.il

[§]School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel; E-mail: mia@math.tau.ac.il

[¶]INRIA at Sophia Antipolis, and CNRS, URA 1376, Laboratoire I3S, Rue Albert Einstein, 06560 Valbonne, France; E-mail: Mariette.Yvinec@sophia.inria.fr

are $O(n \log n + n^{\lceil d/2 \rceil})$ for simplicial diagrams and $O(n^{\lceil d/2 \rceil} \log^{d-1} n)$ for L_∞ -diagrams.

Keywords: Computational geometry, Voronoi diagram, polyhedral distance function.

1 Introduction

Voronoi diagrams are among the most fundamental constructs in computational geometry, and, as such, have been studied a lot during the past two decades. Most of these studies, however, concentrated on Voronoi diagrams in the plane, with only few studies of diagrams in higher dimensions.

We assume in this paper familiarity of the reader with the standard definition and properties of Voronoi diagrams. They can be found in basic textbooks on computational geometry [7, 14, 15, 16], and in several survey papers [3, 12]. There are many variants of Voronoi diagrams. The three main parameters that can vary are (i) the type of sites defining the diagram (points, lines, etc.), (ii) the metric defining the distance to a site, and (iii) the dimension d . The ‘classical’ case is when the sites are points and the metric is euclidean. In this case, a standard lifting transform into \mathbb{E}^{d+1} implies that the maximum combinatorial complexity of the diagram is $\Theta(n^{\lceil d/2 \rceil})$. However, for other metrics, or for other kinds of sites, this analysis does not apply. In this paper we will only consider Voronoi diagrams for point sites, so the only relevant parameters for us are the metric and the dimension.

As observed in [8], the Voronoi diagram of a set \mathcal{S} of n sites in \mathbb{R}^d can be interpreted as the lower envelope of a set of n d -variate functions, each measuring the distance from an arbitrary point of \mathbb{R}^d to a site of \mathcal{S} . Under reasonable assumptions concerning the shape of the sites and the metric, these functions are (piecewise) algebraic of some fixed degree. Hence, applying the recent results of [18] concerning the complexity of the lower envelope of such a collection of functions, we immediately conclude that the complexity of the Voronoi diagram is $O(n^{d+\varepsilon})$, for any $\varepsilon > 0$, where the constant of proportionality depends on ε , d , and the maximum degree of the relevant functions. Since this is a much weaker bound than the one known for the euclidean case, one might be tempted to conjecture that the actual complexity of the diagram is smaller, perhaps close to $O(n^{\lceil d/2 \rceil})$ for fairly general point sites and metrics. This conjecture has been confirmed at least for $d = 2$, where linear bounds on the complexity of the diagram are known in fairly general settings. Unfortunately, a recent construction due to Aronov [1] shows that, for convex polyhedral sites in $d \geq 3$ dimensions, the Voronoi diagram can have $\Omega(n^{d-1})$ complexity, even under the euclidean metric. However, no such construction is known for point sites. Note also that, for $d = 3$, Aronov’s construction does not violate the above conjecture.

Surprisingly, very little is known about generalized Voronoi diagrams in higher dimensions. Recently, Chew et al. [5] have shown that the complexity of the Voronoi

diagram of a set of n lines in \mathbb{R}^3 under a convex polyhedral distance function (see below for a precise definition), induced by a convex polytope with a constant number of faces, is $O(n^2\alpha(n)\log n)$. Thus the conjecture holds in this case. The simpler case, of point sites under similar distance functions, has not yet been investigated, and this paper initiates the study of such diagrams.

For certain technical reasons, the case of point sites is harder to analyze than the case of lines in 3-space. We have not been able to come up with a sharp bound for point sites and arbitrary polyhedral distance functions, even in \mathbb{R}^3 . Nevertheless, we managed to substantiate the conjecture in the following special cases:

- We show that the maximum complexity of the Voronoi diagram of n points in \mathbb{R}^3 under the L_1 metric is $\Theta(n^2)$.
- We show that the maximum complexity of the Voronoi diagram of n points in \mathbb{R}^d under the L_∞ metric is $\Theta(n^{\lceil d/2 \rceil})$.
- We show that the maximum complexity of the Voronoi diagram of n points in \mathbb{R}^d under a simplicial distance function is also $\Theta(n^{\lceil d/2 \rceil})$.

In these bounds we assume that the given sites are in *general position* with respect to the relevant distance function (see below for a precise definition). It is interesting to note that this requirement is essential for the bounds to hold. We give examples of point sets in degenerate configurations for which the complexity of their L_1 -Voronoi diagrams is much larger.

To obtain the bound concerning L_∞ -Voronoi diagrams, we first derive a related new bound on the complexity of the union of n axis-parallel hypercubes in \mathbb{R}^d . We show that if the hypercubes have arbitrary sizes then the maximum complexity of their union is $\Theta(n^{\lceil d/2 \rceil})$, for $d \geq 1$. If all the hypercubes have the same size then the maximum complexity of their union is $\Theta(n^{\lceil d/2 \rceil})$, for $d \geq 2$. These results were known, and are easy to derive, for $d = 1, 2$. An alternative proof of a linear bound for equal-size cubes in \mathbb{R}^3 has been around for the past several years, but was not published.

The proofs of these bounds borrow ideas from the preceding paper [5]. The main ingredient of most of the proofs is a new technique for obtaining recurrence relationships for the number of vertices of the union, which is a special case of a more general analysis technique recently developed by Tagansky [19]. This technique is obtained by modifying and simplifying the proof technique developed in [10, 18] for the analysis of lower envelopes of multivariate functions. This improved technique has already been used in [2, 5, 19] to obtain improved combinatorial bounds for the complexity of various substructures in arrangements and related problems.

Finally, we propose on-line algorithms to compute the Voronoi diagram of n points in \mathbb{R}^d under a simplicial or L_∞ distance function. Their randomized expected running times are, respectively, $O(n \log n + n^{\lceil d/2 \rceil})$ for simplicial diagrams, and $O(n^{\lceil d/2 \rceil} \log^{d-1} n)$ for L_∞ -diagrams.

2 Preliminaries

Let P be a convex polytope in \mathbb{R}^d with a reference point o in its interior. A homothetic copy of P , having the form $a + \rho P$ for $a \in \mathbb{R}^d$ and $\rho \in \mathbb{R}$, is called a *placement* of P . The placement $a + \rho P$ is said to be centered at a and scaled by factor ρ . We define the distance induced by P from a point a to a point b as the smallest scaling factor ρ such that b belongs to the placement $a + \rho P$. That is,

$$d_P(a, b) = \min\{\rho : b \in a + \rho P\}.$$

We refer to d_P as a (*convex*) *polyhedral distance function* (induced by P). Note that $d_P(a, b)$ is not symmetric, and thus is not a metric, unless P admits a center of symmetry and this center is chosen as the reference point.

Let \mathcal{S} be a set of n points in \mathbb{R}^d and P be a convex polytope with m facets. The Voronoi diagram $\text{Vor}_P(\mathcal{S})$ of \mathcal{S} for the distance d_P is defined as the decomposition of \mathbb{R}^d into Voronoi cells, one for each point of \mathcal{S} , where the Voronoi cell $V(s_i)$ of a point $s_i \in \mathcal{S}$ is the set of points of \mathbb{R}^d which are closer to s_i , under the distance function d_P , than to any other point in \mathcal{S} ; that is,

$$V(s_i) = \{p \in \mathbb{R}^d \mid d_P(p, s_i) \leq d_P(p, s_j) \ \forall s_j \in \mathcal{S}\}$$

Each cell $V(s_i)$ is a star-shaped, generally nonconvex, d -polyhedron. More generally, for $1 \leq k \leq d+1$, consider the locus of points p such that p is equidistant (under d_P) to the points of a subset \mathcal{S}_k of cardinality k of \mathcal{S} , and such that p is strictly closer to the points of \mathcal{S}_k than to any other point in $\mathcal{S} \setminus \mathcal{S}_k$. This locus is a $(d - k + 1)$ -dimensional piecewise linear surface, and each of its faces (of any dimension) is a face of the Voronoi diagram $\text{Vor}_P(\mathcal{S})$. (For this locus to have this dimension, the points of \mathcal{S} must lie in *general position* with respect to P —see below for a precise definition and section 7 for further discussion.) The complexity of the Voronoi diagram $\text{Vor}_P(\mathcal{S})$ is defined as the total number of its faces of all dimensions. If we assume general position, then each face of $\text{Vor}_P(\mathcal{S})$ must have at least one vertex, and each vertex is incident to only a constant number of faces of any dimension. It follows that the complexity of the diagram is proportional to the number of its vertices, so we will concentrate in the foregoing analysis on bounding the number of vertices of the diagram.

We denote placements $a + \rho P$ of P by $\hat{P} = \hat{P}(a, \rho)$. A placement \hat{P} is said to be *free* if it contains no points of \mathcal{S} in its interior. If f is a face of P , \hat{f} refers to the corresponding face of \hat{P} . If a point $p \in \mathcal{S}$ belongs to a facet \hat{f} of \hat{P} , the pair (p, f) is said to be a *contact pair* of the placement \hat{P} . A point p is said to be a *simple contact point* of \hat{P} if it belongs to the relative interior of some facet \hat{f} of \hat{P} . A point p of \mathcal{S} which belongs to the relative interior of a face of \hat{P} of codimension k , is said to be a *contact point with multiplicity k* . Thus, a contact point with multiplicity k is involved in at least k contact pairs (and exactly k contact pairs if the polytope P is simple).

The set \mathcal{P} of all placements of a polytope P is a $(d + 1)$ -dimensional manifold. The set of placements such that a given point p belongs to the hyperplane which is

the affine hull of a facet \hat{f} is a hyperplane in \mathbb{P} and the set of placements \hat{P} such that p belongs to a specific facet \hat{f} is a d -polytope.

In the following, we shall assume that the set \mathcal{S} is in *general position* with respect to the distance d_P . Formally, this means that the following property holds:

Let \hat{P} be any placement of P , which involves contacts with points in some subset $\mathcal{S}' \subseteq \mathcal{S}$. For each $q \in \mathcal{S}'$, let \hat{f}_q be the face of \hat{P} of smallest dimension, say j_q , that q touches. The locus of placements of P at which q touches \hat{f}_q is a portion of a $(j_q + 1)$ -dimensional flat, H_q , in \mathbb{P} . Then the flats $\{h_q\}_{q \in \mathcal{S}'}$ must be linearly independent, in the sense that their intersection has codimension $\sum_{q \in \mathcal{S}'} (d - j_q)$.

This implies that no placement \hat{P} of P has any *redundant* contact point, namely a point whose removal from \mathcal{S}' does not gain new degrees of freedom for placements of P in the vicinity of \hat{P} , at which all other contacts are maintained. For example, if two points touch the relative interior of the same facet of \hat{P} then any of these points is redundant. Similarly, if P has two pairs of parallel facets and there is a placement \hat{P} at which each of these four facets touches a point of \mathcal{S} , then each of those four points is redundant. Indeed, let the four contact points be s_1, s_2, s_3, s_4 , so that s_1 and s_2 touch parallel facets and so do s_3 and s_4 . If we remove s_4 , say, then the contacts of s_1 and s_2 fix the scaling factor of \hat{P} . Hence the contact of s_3 with \hat{P} fixes the plane containing the facet that s_4 touches, so we get the same degrees of freedom regardless of whether s_4 is present or not. Thus none of these configurations can arise when \mathcal{S} is in general position.

A consequence of the general position assumption is that the multiplicities of the contact points of any placement sum up to at most to $d + 1$.

A placement whose contact points multiplicities sum up to $d + 1$ is called a *rigid* placement. The free rigid placements of P are centered at the vertices of the Voronoi diagram $\text{Vor}_P(\mathcal{S})$, and each vertex is the center of such a placement, as follows easily from the definitions. The free rigid placements of P with $d + 1$ distinct contact points are centered at what we call the *regular* vertices of the diagram. The center of such a placement is a point of \mathbb{R}^d which is equidistant (under D_P) to $d + 1$ points of \mathcal{S} and closer to these points than to any other point of \mathcal{S} . Any other vertex of the diagram is called *singular*; it corresponds to a free rigid placement of P at which some points of \mathcal{S} lie on lower-dimensional faces of \hat{P} . More generally, points in a k -face of the Voronoi diagram are centers of maximal free placements whose contact points multiplicities sum up to $d + 1 - k$. The k -face is *regular* if all points in these contacts are distinct, and *singular* otherwise. The general position assumption implies that each (regular or singular) Voronoi vertex is incident to $d + 1$ Voronoi edges, and, more generally, that each k -face, for $0 \leq k \leq d$, of $\text{Vor}_P(\mathcal{S})$ is incident to $d + 1 - k$ $(k + 1)$ -Voronoi faces. Thus the number of faces of the Voronoi diagram incident to each vertex is bounded by a constant depending on d . Hence, as already mentioned above, bounding the complexity of the Voronoi diagram reduces to bounding the number of Voronoi vertices and thus the number of free rigid placements.

3 The Complexity of the Union of Axis-Parallel Hypercubes in \mathbb{R}^d

In this section we obtain a result that will be needed in our analysis of L_∞ -Voronoi diagrams, but which is interesting in its own right.

Let \mathcal{C} be a set of n axis-parallel hypercubes in \mathbb{R}^d . Let $\mathcal{A}(\mathcal{C})$ denote the arrangement of these hypercubes, and let $\mathcal{U}(\mathcal{C})$ denote their union. We may assume, with no loss of generality, that the given hypercubes are in general position, meaning that no two distinct facets of the hypercubes lie in a common hyperplane. Otherwise, we can always perturb them slightly, so as to put them in general position, in such a way that the number of faces of the union does not decrease. (This holds for hypercubes of arbitrary sizes. If all the hypercubes have the same size, and we want to maintain this property under the perturbation, then a more refined argument, which we omit here, shows that there is no loss of generality in assuming general position in this case too.) We want to bound the combinatorial complexity of $\mathcal{U}(\mathcal{C})$, which we measure by the number of vertices of the union (the number of all other faces of the union is clearly proportional to the number of vertices, where the constant of proportionality depends only on d , when the hypercubes are in general position). The main result of this section is:

Theorem 3.1 *The maximum number of vertices of the union of n axis-parallel hypercubes in \mathbb{R}^d is $\Theta(n^{\lfloor d/2 \rfloor})$, for $d \geq 1$. If all the given hypercubes have the same size, then the maximum number of vertices of their union is $\Theta(n^{\lfloor d/2 \rfloor})$, for $d \geq 2$ (it remains $O(n)$ for $d = 1$). The constants of proportionality depend on d .*

3.1 The upper bounds

We first prove the upper bounds by induction on d . The bounds hold for $d = 1, 2$. This is trivial for $d = 1$ and follows for $d = 2$ from the results of [11], or by a simpler and more direct proof, which we omit here. Fix $d \geq 3$, assume that the theorem holds for all $d' \leq d - 1$, and let \mathcal{C} be a collection of n axis-parallel hypercubes in \mathbb{R}^d , as above.

For each hypercube $c \in \mathcal{C}$, define $x_j^+(c)$, $x_j^-(c)$ to be, respectively, the largest and smallest x_j -coordinate of the points in c , for $j = 1, \dots, d$. Any hypercube $c \in \mathcal{C}$ has two facets normal to the x_j -axis, for each $j = 1, \dots, d$, lying on the two respective hyperplanes $x_j = x_j^+(c)$, $x_j = x_j^-(c)$. The facet at $x_j^+(c)$ is said to be *positive* (facing the positive x_j direction as we leave c) and the facet at $x_j^-(c)$ is said to be *negative*.

We use the following notational system for representing vertices of the arrangement of the given hypercubes. For a given ordered d -tuple, (c_1, c_2, \dots, c_d) , of hypercubes in \mathcal{C} , let c_j^* be one of the symbols c_j , \bar{c}_j , for $j = 1, \dots, d$. The tuple $(c_1^*, c_2^*, \dots, c_d^*)$ represents the intersection point p of the facets f_1, \dots, f_d , where f_j is a facet of c_j normal to the x_j -axis; it is the positive facet if $c_j^* = c_j$ and the negative facet if

$c_j^* = \bar{c}_j$. Whenever we use this notation, we assume implicitly that the intersection point p exists (and is then unique). The intersection point p is said to be *positive* if all the intersecting facets are positive.

Such an intersection point (or, rather, a vertex of $\mathcal{A}(\mathcal{C})$) is said to be *outer* if it is contained in a $(d-2)$ -face of some hypercube, and *inner* otherwise. If (c_1^*, \dots, c_d^*) is an inner vertex then the hypercubes c_1, \dots, c_d are distinct.

A vertex of $\mathcal{A}(\mathcal{C})$ is said to be a *k-level vertex* if it is contained in the interiors of exactly k of the hypercubes in \mathcal{C} . The vertices of the (boundary of the) union are 0-level vertices. Let $V_k(\mathcal{C})$ denote the number of inner k -level vertices of $\mathcal{A}(\mathcal{C})$, and let $D_k(\mathcal{C})$ denote the number of outer k -level vertices. We also denote by $V_k(n, d)$ the maximum of $V_k(\mathcal{C})$ over all possible collections of n axis-parallel hypercubes in \mathbb{R}^d , and, similarly, denote by $D_k(n, d)$ the maximum of $D_k(\mathcal{C})$ over all possible such collections of hypercubes.

We first estimate the number of outer vertices of the union $\mathcal{U}(\mathcal{C})$. Such an outer vertex p belongs to at least one $(d-2)$ -face of some hypercube $c \in \mathcal{C}$. Since every hypercube contains only $2d(d-1)$ such $(d-2)$ -faces, we can reduce the problem to $2nd(d-1)$ ‘smaller’ problems, as follows. Fix a $(d-2)$ -face f of some hypercube $c \in \mathcal{C}$, and let K be the affine hull of f . Form the intersections $K \cap c'$, for $c' \in \mathcal{C} - \{c\}$. These are $n-1$ axis-parallel hypercubes in the $(d-2)$ -dimensional space K (and if the hypercubes of \mathcal{C} are of equal size, so are these intersection hypercubes). Any outer vertex of $\mathcal{U}(\mathcal{C})$ that lies on f is clearly an (inner or outer) vertex of the union of these intersection hypercubes. It follows that

$$D_0(n, d) \leq 2nd(d-1) \left(D_0^*(n-1, d-2) + V_0^*(n-1, d-2) \right)$$

where the functions D^* and V^* count, respectively, only outer and inner vertices of the union which lie inside some fixed $(d-2)$ -dimensional hypercube. By the induction hypothesis, we have

$$D_0^*(n-1, d-2) + V_0^*(n-1, d-2) = O(n^{\lceil (d-2)/2 \rceil}).$$

If the hypercubes are of equal size, then we have

$$D_0^*(n-1, d-2) + V_0^*(n-1, d-2) = O(n^{\lfloor (d-2)/2 \rfloor}).$$

Indeed, this holds for $d=3$, because the complexity of the union of equal intervals on a line, intersected with another interval of the same length, is $O(1)$. For $d > 3$, the bound follows by the induction hypothesis. Hence we obtain

$$D_0(n, d) = O(n^{\lfloor d/2 \rfloor}), \tag{1}$$

for hypercubes of arbitrary sizes, and

$$D_0(n, d) = O(n^{\lfloor d/2 \rfloor}), \tag{2}$$

for equal-size hypercubes.

In what follows we will also need a bound on $D_1(n, d)$. This is easy to obtain by a standard application of the Clarkson-Shor probabilistic technique [6] (using a random sample of, say, $n/2$ of the hypercubes). This yields, as is easily verified,

$$D_1(n, d) = O(n^{\lceil d/2 \rceil}), \quad (3)$$

for hypercubes of arbitrary sizes, and

$$D_1(n, d) = O(n^{\lfloor d/2 \rfloor}), \quad (4)$$

for equal-size hypercubes.

We next estimate the number of inner vertices of the union. Let p be a 0-level inner vertex, and assume, without loss of generality, that p is positive and has the representation (c_1, \dots, c_d) . For each coordinate x_j , we will slide from p along an edge e_j in the negative x_j direction. This edge is contained in the intersection of the corresponding $d - 1$ positive facets of the hypercubes c_k , for $k = 1, \dots, d$ and $k \neq j$. As we start tracing e_j from p in the negative x_j -direction, we enter the hypercube c_j . We stop the sliding process as soon as we first encounter one of the following three types of events:

- (i) We meet the negative facet of c_j at the 0-level vertex $(c_1, \dots, c_{j-1}, \bar{c}_j, c_{j+1}, \dots, c_d)$. This can happen only if c_j is smaller than the other $d - 1$ hypercubes. For equal-size hypercubes, this cannot happen.
- (ii) We meet another facet (necessarily the negative facet orthogonal to the x_j -axis) of one of the hypercubes c_k , for some $1 \leq k \leq d$ and $k \neq j$, at the 1-level outer vertex $(c_1, \dots, c_{j-1}, \bar{c}_k, c_{j+1}, \dots, c_d)$, which is contained in the interior of c_j .
- (iii) We meet a new hypercube c' at a (necessarily positive) 1-level inner vertex p' , contained in the interior of c_j and represented by $(c_1, \dots, c_{j-1}, c', c_{j+1}, \dots, c_d)$. We say that p' and p are *neighbors* (in the arrangement $\mathcal{A}(\mathcal{C})$).

If we encounter an event of type (i), we simply ignore this edge, and do not use it in our charging scheme that we are about to describe. As just noted, at most one such edge will be ignored.

If we encounter an event of type (ii), we charge the 1-level outer vertex by one unit. Since we can reach the outer vertex $(c_1, \dots, c_{j-1}, \bar{c}_k, c_{j+1}, \dots, c_d)$ from an inner vertex only along one of the two corresponding facets of c_k (in a direction normal to the other facet), this outer vertex can be charged, by type (ii) events, at most twice, for a total of 2 units (recall that c_k is the unique hypercube appearing more than once in the tuple representing the outer vertex).

If we encounter an event of type (iii), we charge the 1-level inner vertex p' by one unit. The problem is that the vertex p' may be charged in up to d events of type (iii), and we need to account for such multiple charges. Suppose that p' is charged by w of its 0-level inner neighbors. If $w = 1$ (or $w = 0$) then p' pays one unit of charge for its

unique charging neighbor (or does not pay at all). If $w > 1$, we will distribute $w - 1$ of the w units that p' is charged with to other outer vertices, so that p' still has to pay only one unit of charge.

Suppose that p' is positive, has the representation (c_1, \dots, c_d) , and is contained in the interior of c_0 . Suppose that $p_1 = (c_0, c_2, c_3, \dots, c_d)$ and $p_2 = (c_1, c_0, c_3, \dots, c_d)$ are two 0-level inner neighbors of p' . Let h be the 2-dimensional plane $x_i = x_i^+(c_i)$, for $i = 3, \dots, d$, which contains the three vertices p', p_1, p_2 . Let r be the axis-parallel rectangle in h having these points as three of its vertices (see Figure 1). For each hypercube $c \in \mathcal{C}$, let $s(c) = c \cap h$. The collection \mathcal{S} of the nonempty intersections of this form is a set of at most n axis-parallel squares in h . By construction, the two edges p_1p', p_2p' of r do not cross the boundary of any square in \mathcal{S} . Let q be the fourth corner of r . Clearly, q is an outer vertex of $\mathcal{A}(\mathcal{C})$ with the representation $q = (c_0, c_0, c_3, \dots, c_d)$.

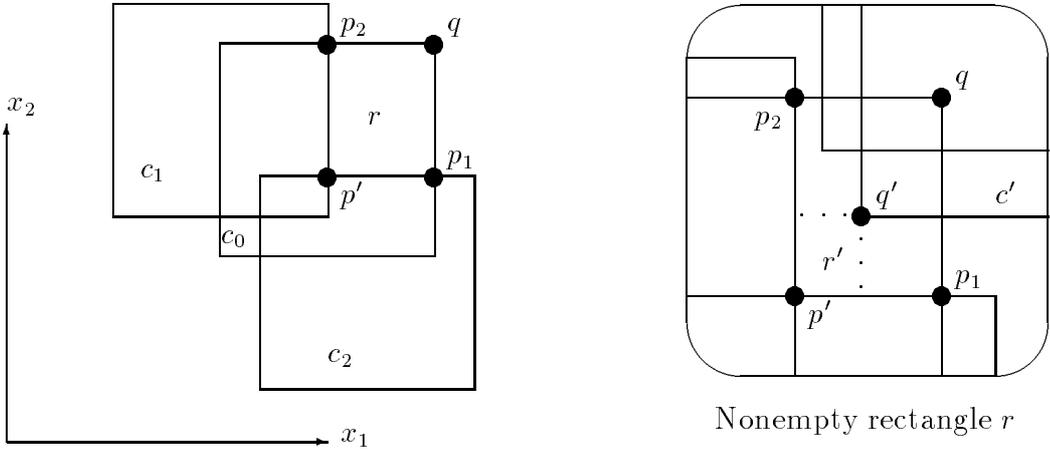


Figure 1: Charging outer vertices within the rectangle p_1p_2

If r does not intersect the interior of any hypercube other than c_0 , then q is a 0-level outer vertex of $\mathcal{A}(\mathcal{C})$, to which we pass 1 unit of charge from p' . The vertex q can be charged in this manner at most once. Indeed, given q , there is only one 2-dimensional plane in which q can be charged: this is the plane passing through q and spanned by the normal directions of the unique pair of facets in the representation of q that belong to the same hypercube (recall that c_0, c_3, \dots, c_d are all distinct, by construction). Moreover, r is the unique maximal rectangle in $c_0 \cap h$ with corner q which is disjoint from (the interior of) any other square in \mathcal{S} . This implies that q can be charged at most once, namely, only by the opposite corner of r .

If the rectangle r meets some other square of \mathcal{S} , let q' be the point in $r \cap \mathcal{U}(\mathcal{C}')$ closest to p' , where $\mathcal{C}' = \mathcal{C} - \{c_0, c_1, \dots, c_d\}$. Note that q' cannot lie on the edges p_1p' or p_2p' , since these edges do not cross any hypercube in \mathcal{C}' , and that q' must be a 1-level outer vertex of $\mathcal{A}(\mathcal{C})$ having the representation $(\bar{c}', \bar{c}', c_3, \dots, c_d)$, for some $c' \in \mathcal{C}'$; see Figure 1. Let r' be the axis-parallel rectangle in h having p' and q' as

opposite corners. Again, (the interior of) r' is contained only in the interior of c_0 , and meets no other hypercube of \mathcal{C} . We pass 1 unit of charge from p' to q' . We claim that, in this case too, q' can be charged in this manner at most once. Indeed, given q' , there is only one 2-dimensional plane h in which q' can be charged, which is shown by the same argument given above (since c', c_3, \dots, c_d are all distinct). Moreover, r' is the unique maximal rectangle in $c_0 \cap h$ with corner q' which is disjoint from (the interior of) any other square in \mathcal{S} and lies in the quadrant of q' opposite to that containing q . This implies, as above, that q' can be charged at most once, namely, only by the opposite corner of r' . Together with the previous charges in the case of type (ii) events, any 1-level outer vertex of $\mathcal{A}(\mathcal{C})$ can be charged a total of 3 units.

If the vertex p' has $w > 1$ 0-level inner neighbors, the number of pairs of these neighbors is always at least $w - 1$, so there is no problem in distributing $w - 1$ units of charge from p' to nearby outer vertices, in the manner described above.

Summing up the charges, each 0-level inner vertex p receives at least $d - 1$ units, by sliding in all directions parallel to the coordinate axes, with the possible exception of one direction in which we encounter a type (i) event (for equal-size hypercubes, p always receives d units). Each 0-level outer vertex pays at most 1 unit, each 1-level outer vertex pays at most 3 units, and each 1-level inner vertex pays at most 1 unit. We can thus conclude that

$$(d - 1)V_0(\mathcal{C}) \leq V_1(\mathcal{C}) + 3D_1(\mathcal{C}) + D_0(\mathcal{C}), \quad (5)$$

for hypercubes of arbitrary sizes, and

$$dV_0(\mathcal{C}) \leq V_1(\mathcal{C}) + 3D_1(\mathcal{C}) + D_0(\mathcal{C}), \quad (6)$$

for equal-size hypercubes. We can now apply the following probabilistic argument, similar to that used in [5, 19]. In the case of hypercubes of arbitrary sizes, we have

$$\begin{aligned} \frac{n-1}{n}V_0(\mathcal{C}) &= \frac{n-d}{n}V_0(\mathcal{C}) + \frac{d-1}{n}V_0(\mathcal{C}) \leq \\ &\frac{n-d}{n}V_0(\mathcal{C}) + \frac{1}{n}V_1(\mathcal{C}) + \frac{3}{n}D_1(\mathcal{C}) + \frac{1}{n}D_0(\mathcal{C}) \\ &= \mathbf{E}(V_0(\mathcal{R})) + O(n^{\lfloor d/2 \rfloor - 1}), \end{aligned}$$

where \mathcal{R} is a random sample of $n - 1$ hypercubes of \mathcal{C} , and where \mathbf{E} denotes expectation with respect to the choice of \mathcal{R} (see (1), (3)). For the case of equal-size hypercubes, we obtain, in much the same way, the improved recurrence (see (2), (4))

$$V_0(\mathcal{C}) \leq \mathbf{E}(V_0(\mathcal{R})) + O(n^{\lfloor d/2 \rfloor - 1}).$$

We can thus write, for the case of hypercubes of arbitrary sizes, the recurrence

$$\frac{n-1}{n}V_0(n, d) \leq V_0(n-1, d) + O(n^{\lfloor d/2 \rfloor - 1}),$$

whose solution, for $d \geq 3$, is easily seen to be

$$V_0(n, d) = O(n^{\lceil d/2 \rceil}).$$

For the case of equal-size hypercubes, we obtain the recurrence

$$V_0(n, d) \leq V_0(n-1, d) + O(n^{\lfloor d/2 \rfloor - 1}),$$

whose solution, for $d \geq 2$, is easily seen to be

$$V_0(n, d) = O(n^{\lfloor d/2 \rfloor}).$$

This completes the proof of the upper bounds.

3.2 The lower bounds

We next prove the lower bound for equal-size hypercubes, by constructing the following set \mathcal{C} of $m\Delta$ hypercubes $c_{k,i}$, for $k = 1, \dots, \Delta$ and $i = 1, \dots, m$, in $\mathbb{R}^{2\Delta}$, for integer parameters m, Δ . Set $M > m$ and, for $j = 1, \dots, 2\Delta$, let the x_j -coordinate of the center of the hypercube $c_{k,i}$ be

$$\begin{cases} \frac{i}{M} & j = 2k - 1 \text{ or } j = 2k \\ \frac{1}{2} & j \text{ is odd and } j \neq 2k - 1 \\ 0 & j \text{ is even and } j \neq 2k. \end{cases}$$

The common size of all these hypercubes is 2.

Let V be the set of the following $m\Delta$ points in $\mathbb{R}^{2\Delta}$:

$$v_{i_1, \dots, i_\Delta} = \left(\frac{i_1}{M} + 1, \frac{i_1}{M} - 1, \dots, \frac{i_\Delta}{M} + 1, \frac{i_\Delta}{M} - 1 \right),$$

where $i_k \in \{1, \dots, m\}$, for $k = 1, \dots, \Delta$. For each $r \in \{1, 2, \dots, \Delta\}$ we have, as is easily verified,

$$x_j^-(c_{r, i_r}) \leq x_j(v_{i_1, \dots, i_\Delta}) \leq x_j^+(c_{r, i_r}) \quad \text{for } j = 1, \dots, 2\Delta,$$

with two of the inequalities being equalities (for $x_{2r-1}^+(c_{r, i_r})$ and $x_{2r}^-(c_{r, i_r})$). Thus each point $v \in V$ lies on a $(d-2)$ -face of each of the Δ hypercubes c_{r, i_r} and is thus an outer vertex of $\mathcal{A}(\mathcal{C})$, which is represented, in the above notation, as $(c_{1, i_1}, \bar{c}_{1, i_1}, \dots, c_{\Delta, i_\Delta}, \bar{c}_{\Delta, i_\Delta})$. Next we note that, for $k = 1, \dots, \Delta$ and for $q > i_k$, we have

$$x_{2k}(v_{i_1, \dots, i_\Delta}) = \frac{i_k}{M} - 1 < \frac{q}{M} - 1 = x_{2k}^-(c_{k, q}),$$

and for $q < i_k$ we have

$$x_{2k-1}(v_{i_1, \dots, i_\Delta}) = \frac{i_k}{M} + 1 > \frac{q}{M} + 1 = x_{2k-1}^+(c_{k, q}).$$

Thus no point $v \in V$ lies in the interior of any hypercube, so they are all outer vertices of $\bigcup \mathcal{C}$. This is easily seen to prove the lower bound for equal-size axis-parallel hypercubes, in any dimension $d \geq 2$.

To prove the lower bound for axis-parallel hypercubes of arbitrary sizes, it suffices to consider the case where d is odd, say $d = 2\Delta + 1$. Take the number M above to be $(m + 1)^2$. All the points in $V \subset \mathbb{R}^{2\Delta}$ now lie in the interior of the hypercube b , whose center is at $(1, -1, \dots, 1, -1)$ and whose size is $\frac{2}{m+1}$. Let s' be the segment in $\mathbb{R}^{2\Delta+1}$ connecting the origin with the point $(0, \dots, 0, 2)$, and let $s'_i \subseteq s'$, for $i = 1, \dots, m$ be the segment in $\mathbb{R}^{2\Delta+1}$ connecting $(0, \dots, 0, \frac{2i-1}{m} - \frac{1}{m+1})$ and $(0, \dots, 0, \frac{2i-1}{m} + \frac{1}{m+1})$. Define the $m\Delta$ hypercubes $c_{k,i}$ in $\mathbb{R}^{2\Delta}$, as above, and embed them in the hyperplane $x_{2\Delta+1} = 0$. Now define, for each i and k , a new $(2\Delta + 1)$ -hypercube $c'_{k,i}$ as the Minkowski sum $c_{k,i} \oplus s'$. Define another collection $\{b'_1, \dots, b'_m\}$ of m smaller hypercubes in $\mathbb{R}^{2\Delta+1}$, where $b'_i = b \oplus s'_i$. We thus obtain a collection \mathcal{C} of $m(\Delta + 1)$ hypercubes in $\mathbb{R}^{2\Delta+1}$. Associate with each vertex $v \in V$ the vertical edge $v \oplus s'$, which intersects the boundary of each of the m pairwise-disjoint hypercubes b'_i at points that are clearly vertices of the union of \mathcal{C} . This shows that $\bigcup \mathcal{C}$ has at least $2m^{\Delta+1}$ vertices, thus establishing the lower bound for axis-parallel hypercubes of arbitrary sizes (in odd dimensions). Note that we only used two different sizes in this construction. This completes the proof of Theorem 3.1. \square

4 The L_∞ -Voronoi Diagram of Points in \mathbb{R}^d

In this section, we study the complexity of the L_∞ -Voronoi diagram of a set of n points in \mathbb{R}^d . The L_∞ -distance function is the distance function associated with an axis parallel hypercube in \mathbb{R}^d whose side length is 2, where the reference point is the center of the hypercube. We show the following result:

Theorem 4.1 *The maximum complexity of the L_∞ -Voronoi diagram of a set of n points in \mathbb{R}^d is $\Theta(n^{\lceil d/2 \rceil})$, provided that the set is in general position with respect to the L_∞ -distance function.*

4.1 The upper bound

Let \mathcal{S} be a set of n points in general position in \mathbb{R}^d , with respect to an axis-parallel hypercube C . We denote by $\text{Vor}_\infty(\mathcal{S})$ the Voronoi diagram of \mathcal{S} under the L_∞ -distance. Since C is a simple polytope, the discussion in Section 2 implies that a vertex of $\text{Vor}_\infty(\mathcal{S})$ corresponds to a free rigid placement of C with exactly $d + 1$ contact pairs. The vertex is regular if all the contact points are distinct, and singular otherwise. By the general position assumption, no facet of any placement \hat{C} of C can contain more than one point of \mathcal{S} , and the $d + 1$ contact pairs involve $d + 1$ facets of \hat{C} . Since \hat{C} has d pairs of parallel facets, a free rigid placement has at least two parallel contact pairs, namely, contact pairs involving parallel facets. Moreover, by

the general position assumption, there can be only one pair of parallel contacts pairs, as already noted. It follows that at this placement there is a vertex v of \hat{C} incident to d (mutually orthogonal) facets of \hat{C} , each containing a point of \mathcal{S} . We can represent \hat{C} as

$$\hat{C} = \{\mathbf{x} \mid x_j^-(\hat{C}) \leq x_j \leq x_j^+(\hat{C}), \text{ for } j = 1, \dots, d\},$$

where $x_j^+(\hat{C}) - x_j^-(\hat{C}) = 2\rho(\hat{C})$ for all j , where $\rho(\hat{C})$ is the scaling factor of \hat{C} . With no loss of generality, assume that v is incident to the facets $x_j = x_j^-(\hat{C})$, for $j = 1, \dots, d$, and that the facet $x_j = x_j^-(\hat{C})$ touches a point $p_j \in \mathcal{S}$, for $j = 1, \dots, d$ (so v is the vertex of \hat{C} all of whose coordinates are the smallest possible). As remarked above, these points do not have to be distinct: if k of the p_j 's are equal (to some $p \in \mathcal{S}$), then p lies on a $(d - k)$ -face of \hat{C} incident to v .

We now shrink \hat{C} towards v , keeping v fixed. We lose one contact of \hat{C} with a point, but retain the d remaining contact pairs (between the points p_1, \dots, p_d and the corresponding facets of \hat{C} incident to v). We stop the shrinking when one of these points comes to lie on another facet of \hat{C} . With no loss of generality, assume that this is the point p_1 , and that the new facet it lies on is $x_2 = x_2^+(\hat{C})$ (because each negative facet already has a contact, the new facet has to be a positive facet). The new placement that we have reached is free and rigid but singular. Let v' be the vertex of \hat{C} incident to the facets $x_1 = x_1^-(\hat{C})$, $x_2 = x_2^+(\hat{C})$, and $x_j = x_j^-(\hat{C})$, for all $j = 3, \dots, d$. These facets are incident to the points p_1, p_3, \dots, p_d . We now shrink \hat{C} towards v' , losing the contact between p_2 and the facet $x_2 = x_2^-(\hat{C})$, but retaining the other d contact pairs, and stop when one of the contacting points comes to lie on another facet of \hat{C} .

We keep iterating this process. In the general step, just before starting a shrinking process, we have some number, k , of remaining points, call them q_1, \dots, q_k , such that each q_i lies in the relative interior of some face f_i of codimension t_i , where $\sum_{i=1}^k t_i = d + 1$. By the general position assumption, there is exactly one parallel pair of facets among the $d + 1$ facets of \hat{C} that are incident to the faces f_i .

Suppose first that in the present placement of \hat{C} there is a face f_i of codimension 1 (that is, f_i is a facet), and that f_i is one of the pair of parallel facets. Then the remaining $k - 1$ faces f_j , for $j \neq i$, have a common vertex w , and we can keep shrinking \hat{C} towards w , losing only the one contact pair involving q_i and maintaining the other d contact pairs. We stop the shrinking, as above, when one of the other $k - 1$ points comes to lie on another facet of \hat{C} .

Suppose next that the preceding subcase does not occur. Let q_1 and q_2 be the two (distinct) points incident to the (unique) pair of parallel facets. By assumption, the corresponding faces f_1, f_2 have each codimension at least 2. Hence, by the general position assumption, there are at least 3 coordinates, say x_1, x_2, x_3 , such that q_1 is incident to a pair of facets orthogonal to the x_1 and x_3 axes, and q_2 is incident to a pair of facets orthogonal to the x_2 and x_3 axes. Let us fix the points q_1 and q_2 (there are $O(n^2)$ choices for such a pair), and also fix the $\beta \geq 3$ coordinates such that the facets incident to q_1 and q_2 are orthogonal to these coordinates (there is a constant

number of such choices). Note that β is equal to the sum of the multiplicities of the two points involved in the parallel contact pairs minus 1. Then the scaling factor $\rho_0 = \rho(\hat{C})$ of \hat{C} is fixed (under the above assumptions, it is equal to $\frac{1}{2}|x_3(q_1) - x_3(q_2)|$), and β coordinates of its center are also fixed. Hence the center of \hat{C} must lie on an appropriate $(d - \beta)$ -th dimensional flat K . For each point $p \in \mathcal{S} \setminus \{q_1, q_2\}$, let \hat{C}_p be the intersection of K with the cube $p + \rho_0 C$. It is easily checked that the center of \hat{C} in placements under consideration must be a vertex of the union of the equal-size axis-parallel hypercubes \hat{C}_p . By Theorem 3.1, the number of such vertices is $O(n^{\lfloor (d-\beta)/2 \rfloor})$, so the number of placements under consideration is

$$O(n^2) \cdot O(n^{\lfloor (d-\beta)/2 \rfloor}) = O(n^{\lceil d/2 \rceil}),$$

since $\beta \geq 3$.

To recap, the number of terminal placements of \hat{C} that we can reach by our iterated shrinking process is $O(n^{\lceil d/2 \rceil})$. This also includes the case where the iterated shrinking process can continue all the way through, until the hypercube shrinks to a point; the number of such terminal placements is clearly only $O(n)$.

We claim that any such terminal placement \hat{C} can be reached from only a constant number of initial placements of C . To see this, suppose first that the shrinking process has not terminated at a singleton hypercube. Pick a terminal placement \hat{C} , and reverse the shrinking process: choose a vertex v of \hat{C} incident to all but one of the $d+1$ facets touched by points of \mathcal{S} . By construction, there is always at least one such vertex at the end of a shrinking step, and the discarded facet is necessarily one of the pair of parallel contact facets. When we expand \hat{C} from v , none of the points touching \hat{C} can enter into the interior of \hat{C} (the point touching the discarded facet also touches another facet incident to v , so it remains on the boundary of \hat{C} while we expand). We stop the expanding process when \hat{C} hits another point, and then continue to expand from some (possibly different) vertex of \hat{C} . There are at most d expanding steps, and in each of them we have a constant number of choices for the vertex from which we expand, implying that only a constant number of initial placements (where the constant depends on d) can reach the same terminal placement \hat{C} . A similar (and actually simpler) argument also applies to the case where the terminal placement is a singleton hypercube.

So far we have only counted vertices of the diagram, but the arguments in Section 2 imply that the overall complexity of the diagram is proportional to the number of its vertices, which thus completes the proof of the upper bound in Theorem 4.1.

4.2 The lower bound

We next prove the lower bound in Theorem 4.1. We first give a sufficient condition for $d+1$ points p_1, p_2, \dots, p_{d+1} in \mathbb{R}^d to lie on the boundary of some axis-parallel

hypercube. The condition is:

$$\begin{aligned}
x_i(p_i) &= \min \{x_i(p_j) \mid j = 1, \dots, d+1\}, \quad \text{for } i = 1, \dots, d \\
x_1(p_{d+1}) &= \max \{x_1(p_j) \mid j = 1, \dots, d+1\} \\
x_1(p_{d+1}) - x_1(p_1) &= \|p_{d+1} - p_1\|_\infty = \max \{\|p_i - p_j\|_\infty \mid i, j = 1, \dots, d+1\}.
\end{aligned} \tag{7}$$

Indeed, under this condition, p_1, p_2, \dots, p_{d+1} are on the boundary of the hypercube c of size $a(c) = \|p_{d+1} - p_1\|_\infty$, and whose smallest x_i -coordinate is $x_i^-(c) = x_i(p_i)$, for $i = 1, \dots, d$. (Notice that any ordered $(d+1)$ -tuple of points that lie on the boundary of a hypercube fulfills the above condition, up to a permutation of the axes or of the points and up to inversion of the orientations of some of the axes.)

Let us assume that the dimension d is odd. The idea of the construction is to take $l = (d+1)/2$ lines in \mathbb{R}^d and n points on each line, such that any appropriately ordered $(d+1)$ -tuple of points, formed by choosing a pair of consecutive points on each of those lines, satisfies the above condition. Then, since any line intersects the boundary of a hypercube in at most two points, the hypercube passing through these $d+1$ points is a free rigid placement for the whole set, which implies that the complexity of the L_∞ -Voronoi diagram of this set is $\Omega(n^{\lceil d/2 \rceil})$. To implement this idea, choose a real α such that $0 < \alpha < \frac{1}{2^n}$, and define the following lines:

- For $r = 1, \dots, l-1$, line δ_r is directed along $v_r = -\mathbf{e}_{2r} + \sum_{i \neq 2r} \mathbf{e}_i$ (where \mathbf{e}_i denotes the unit vector directed along the positive x_i -axis) and passes through the point p_r whose coordinates are all equal to 2 except that $x_{2r-1}(p_r) = x_{2r}(p_r) = 0$.
- the last line δ_l is directed along $v_l = \sum_{i=1}^d \mathbf{e}_i$ and passes through the point p_l whose coordinates are all equal to 2 except that $x_1(p_l) = 4$ and $x_d(p_l) = 0$.
- for $r = 1, \dots, l$, the n points on the line δ_r are the points $p_r(k_r) = p_r + k_r \alpha v_r$ for $k_r = 0, \dots, n-1$

It is now easy to verify that, for any choice of k_1, k_2, \dots, k_l in $\{0, \dots, n-1\}^l$, the $(d+1)$ -tuple $\{p_1(k_1), p_1(k_1+1), \dots, p_l(k_l), p_l(k_l+1)\}$ fulfills the condition (7). Table 1 shows the coordinates of the points in such a tuple for $d = 5$.

Thus, if d is odd, the L_∞ -Voronoi diagram of a set of n points in \mathbb{R}^d can have $\Omega(n^{\lceil d/2 \rceil})$ complexity in the worst case. The result obviously also holds for any even dimension d , by using the above construction in dimension $d-1$.

5 Voronoi Diagrams for Simplicial Distance Functions

In this section, we consider the Voronoi diagram $\text{Vor}_\sigma(\mathcal{S})$ of a point set \mathcal{S} in \mathbb{R}^d for a distance function d_σ induced by a d -simplex σ , and prove the following:

Theorem 5.1 *The maximum complexity of the Voronoi diagram of a set of n points in \mathbb{R}^d , under the distance function induced by a d -simplex, is $\Theta(n^{\lceil d/2 \rceil})$, provided that the points are in general position with respect to the simplex.*

$p_1(k_1)$	$p_1(k_1 + 1)$	$p_2(k_2)$	$p_2(k_2 + 1)$	$p_3(k_3)$	$p_3(k_3 + 1)$
$k_1\alpha$	$(k_1 + 1)\alpha$	$2 + k_2\alpha$	$2 + (k_2 + 1)\alpha$	$4 + k_3\alpha$	$4 + (k_3 + 1)\alpha$
$-k_1\alpha$	$-(k_1 + 1)\alpha$	$2 + k_2\alpha$	$2 + (k_2 + 1)\alpha$	$2 + k_3\alpha$	$2 + (k_3 + 1)\alpha$
$2 + k_1\alpha$	$2 + (k_1 + 1)\alpha$	$k_2\alpha$	$(k_2 + 1)\alpha$	$2 + k_3\alpha$	$2 + (k_3 + 1)\alpha$
$2 + k_1\alpha$	$2 + (k_1 + 1)\alpha$	$-k_2\alpha$	$-(k_2 + 1)\alpha$	$2 + k_3\alpha$	$2 + (k_3 + 1)\alpha$
$2 + k_1\alpha$	$2 + (k_1 + 1)\alpha$	$2 + k_2\alpha$	$2 + (k_2 + 1)\alpha$	$k_3\alpha$	$(k_3 + 1)\alpha$

Table 1: A 6-tuple of points in \mathbb{R}^5 used in the lower bound construction for L_∞ -Voronoi diagrams

5.1 The upper bound

Our goal is to bound the number of free rigid (homothetic) placements of a d -simplex σ among a set \mathcal{S} of n points in general position with respect to σ . Each free rigid placement $\hat{\sigma}$ has $d + 1$ contact pairs involving $2 \leq k \leq d + 1$ distinct contact points s_1, s_2, \dots, s_k . Let $\mu(s_i)$ be the multiplicity of the contact point s_i at this placement. Then, by the general position assumption, $\sum_{i=1, \dots, k} \mu(s_i) = d + 1$.

The number of free rigid placements involving at most $\lfloor (d + 1)/2 \rfloor = \lceil d/2 \rceil$ distinct contact points is obviously $O(n^{\lceil d/2 \rceil})$. In particular, this also bounds the number of free rigid placements with all contact points having multiplicity ≥ 2 . The number of free rigid placements with two contact points of respective multiplicities 1 and d is $O(n)$ (a contact point with multiplicity d arises when a vertex of $\hat{\sigma}$ touches a point of \mathcal{S}).

Let us consider a free rigid placement $\hat{\sigma}$ with at least one simple contact point (i.e., with multiplicity 1) and with no contact point with multiplicity d . Let \hat{v} be a vertex of $\hat{\sigma}$ opposite to a facet \hat{f} touching a simple contact point. We shrink the simplex towards \hat{v} , dropping at once the contact pair involving \hat{f} but keeping the other d contact pairs. The shrinking process stops as soon as one of the contact points p reaches a new facet of $\hat{\sigma}$, thus augmenting the multiplicity of p by one.

This shrinking scheme can be repeated as long as the free rigid placement has a simple contact point and no contact point with multiplicity d (in the latter case, the shrinking will collapse σ to a single point). At the end, we reach either a free rigid placement of σ such that each contact point has multiplicity at least two, or a free rigid placement with two contact points of respective multiplicities 1 and d .

Each such terminal placement can be reached from only a constant number, depending on d , of initial free rigid placements. To show this, we consider, as above, the reverse of the shrinking process. Each step of the reverse process expands $\hat{\sigma}$ from a vertex \hat{v} , such that the facet \hat{f} opposite to \hat{v} does not touch any point with contact multiplicity 1, and stop as soon as \hat{f} hits a new point of \mathcal{S} . There are at most d

expanding steps (actually, only about half as many steps), and at each step we have a constant number of choices, showing that the reverse process can reach at most a constant number (depending on d) of free rigid placements. Thus, the number of free rigid placements is proportional to the number of terminal placements in the shrinking process, which, as argued above, is $O(n^{\lceil d/2 \rceil})$, thus proving the asserted upper bound.

5.2 The lower bound

We prove the lower bound in a manner similar to that used for the L_∞ -distance. Without loss of generality, we may assume that the simplex σ defining the distance is the simplex whose vertices are the origin and one point at positive unit abscissa on each coordinate axis. Indeed, any simplex can be transformed into this simplex by an affine transformation, which does not change the combinatorial structure and the complexity of the Voronoi diagram.

A sufficient condition for $d+1$ points in \mathbb{R}^d to be contact points of a rigid placement of σ is the following:

$$\begin{aligned} x_i(p_i) &= \min \{x_i(p_j) \mid j = 1, \dots, d+1\} \quad \text{for } i = 1, \dots, d \\ \sum_{i=1}^d x_i(p_{d+1}) &= \max \{\sum_{i=1}^d x_i(p_j) \mid j = 1, \dots, d+1\}. \end{aligned} \quad (8)$$

As in the case of the L_∞ -distance, we assume first that the dimension d is odd, and we choose $(d+1)/2$ lines in \mathbb{R}^d and n points on each line, such that any set of $d+1$ points, formed by choosing a pair of consecutive points on each of those lines, satisfies the above condition, for an appropriate permutation of the points. In fact, it is easily seen that the lines constructed in subsection 4.2 fulfill this requirement, provided that the parameter α is chosen so that $0 < \alpha < \frac{1}{2nd}$. This proves that, if d is odd, the complexity of a simplicial Voronoi diagram of n points in \mathbb{R}^d can be $\Omega(n^{\lceil d/2 \rceil})$, a result which holds a fortiori for even dimensions too.

6 The L_1 -Voronoi Diagram of Points in \mathbb{R}^3

This section analyzes the complexity of Voronoi diagrams of point sets under the L_1 -norm. The L_1 -distance between two points p and q of \mathbb{R}^d is

$$d_{L_1}(p, q) = \sum_{i=1}^d |p_i - q_i|.$$

This distance function is polyhedral, and is induced by the d -polytope which is the dual of the d -cube. This polytope is the convex hull of the $2d$ unit vectors $\pm \mathbf{e}_i$, where \mathbf{e}_i is the unit vector in the positive x_i -direction, for $i = 1, \dots, d$. We will call this polytope a d -co-cube. In the case of the L_1 -distance function, we have only been able to prove tight bounds for $d = 3$:

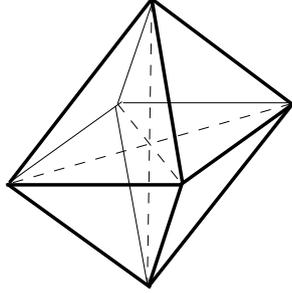


Figure 2: The regular octahedron

Theorem 6.1 *If \mathcal{S} is a set of n points in \mathbb{R}^3 in general position with respect to the L_1 -distance, then the maximum complexity of the Voronoi diagram of \mathcal{S} under the L_1 -distance is $\Theta(n^2)$.*

6.1 The upper bound

In the three-dimensional case, the co-cube is just the regular octahedron O which is the convex hull of the six vertices $u_1(+1, 0, 0)$, $\overline{u}_1(-1, 0, 0)$, $u_2(0, 1, 0)$, $\overline{u}_2(0, -1, 0)$, $u_3(0, 0, 1)$ and $\overline{u}_3(0, 0, -1)$. The octahedron has twelve edges and eight faces and is shown in Figure 2.

Let \mathcal{S} be a set of n points in \mathbb{R}^3 in general position with respect to the octahedron O . Our goal is to bound the number of free rigid placements of O among the points of \mathcal{S} . For this, we bound, in succession, the number of

- (P1) free rigid placements with at least one contact point of multiplicity at least two (which we call hereafter a *double contact point*);
- (P2) free rigid placements with three contact pairs involving three faces of O sharing a common vertex; and
- (P3) all other free rigid placements.

Let \hat{O} be a placement of type (P1), with a double contact point. That is, there is a point $p \in \mathcal{S}$ that lies on an edge \hat{e} of \hat{O} ; we will denote this double contact by the pair (p, e) . For each pair (p, e) of a point p of \mathcal{S} and an edge e of O , the subset of placements attaining the double contact pair (p, e) is contained in a two-dimensional linear subspace $\mathbb{P}(p, e)$ of the set \mathbb{P} of placements. The subspace $\mathbb{P}(p, e)$ can be parametrized by the position t of one of the endpoints of \hat{e} on the line parallel to e through p , and by the scaling factor ρ . In this subspace, any other contact pair (p', f) of a placement \hat{O} appears as a (possibly empty) segment $s(p', f)$. Then a rigid placement with the double contact pair (p, e) corresponds to a vertex of the planar arrangement of those segments, and a free placement with the double contact

pair (p, e) corresponds to a point which lies on or below the lower envelope of those segments, relative to the ρ -direction. This follows from the observation that if we fix t and increase ρ then \hat{O} expands, so that, once a point enters the expanding octahedron, it will never leave it again (see also [5] for a similar argument). Hence, the number of free rigid placements with the double contact pair (p, e) is at most the number of vertices of the lower envelope of the at most $6(n - 1)$ segments representing the contact pairs (p', f) in $\mathcal{IP}(p, e)$. (Note that (a) not every vertex of the lower envelope necessarily represents a free rigid placement of type (P1), because the corresponding scaling factor may be too small for the point p to actually lie on e , (b) the vertices formed by the intersection of two segments represent placements with three contact points (one of which is p), whereas segment endpoints represent placements with two contact points (one of which is p), each being a double-contact point, and (c) by the general position assumption, the two facets of \hat{O} incident to e are not involved in those contact pairs.) Moreover, it is easy to verify that the segments that represent contacts with a fixed face f of O are all parallel, thus we have six families of at most $n - 1$ parallel segments each, so the complexity of their overall lower envelope is linear. Summing over the $12n$ possible pairs (p, e) , we conclude:

Lemma 6.2 *Given a set \mathcal{S} of n points in \mathbb{R}^3 in general position with respect to the regular octahedron O , the number of type (P1) free rigid placements of O amidst the points of \mathcal{S} is $O(n^2)$.*

Let us next bound the number of free rigid placements \hat{O} of type (P2), that is, with contact pairs involving three facets sharing a vertex. If such a placement has no double contact point, we apply a shrinking process to \hat{O} , in which the vertex of \hat{O} incident to three contact faces is fixed. This process maintains at least three contact pairs and does not encounter any new contact point, since at any time during the shrinking the octahedron is contained in the initial placement. The shrinking process stops as soon as one of the contact points reaches an edge of the octahedron. Then the reached placement is a free rigid placement with a double contact point. Moreover, each free rigid placement with a double contact point can be reached in this way from at most two rigid free placements without double contact points. Indeed, from a terminal placement with a double contact point on edge e we can recover an initial free rigid placement without a double contact point by expanding the octahedron from one of the two vertices opposite to edge e in the two facets incident to e . Thus the number of type (P2) placements without double contact point is no more than twice the number of free rigid placements with a double contact point, which proves the following lemma.

Lemma 6.3 *Given a point set \mathcal{S} as above, the number of free rigid placements of O with contact pairs involving three facets sharing a common vertex is $O(n^2)$.*

Finally, we bound the number of all other rigid free placements, that is, placements of type (P3). Let us consider a rigid placement \hat{O} with no double contact point and

with no vertex common to three facets involved in contact pairs. Only two cases are then possible:

- (a) The 4 contact facets f_1, f_2, f'_1, f'_2 of O form two pairs $(f_1, f_2), (f'_1, f'_2)$ of adjacent facets (i.e., with a common edge) and two complementary pairs $(f_1, f'_1), (f_2, f'_2)$ of parallel facets.
- (b) The 4 contact facets have no pair of adjacent facets. This case can be realized only by one of the two following complementary subsets of four facets of O : the first set is $\{u_1u_2u_3, \overline{u_1u_2u_3}, \overline{u_1u_2u_3}, u_1\overline{u_2u_3}\}$, and the second set is $\{\overline{u_1u_2u_3}, u_1\overline{u_2u_3}, u_1u_2\overline{u_3}, \overline{u_1u_2u_3}\}$; see Figure 2.

The first case does not occur for sets of points in general position with respect to O , as already noted. Thus it remains to bound the number of free rigid placements in case (b). For this we apply the following scheme. Let \hat{O} be a rigid free placement as in case (b) with the four contact pairs $(p_1, f_1), (p_2, f_2), (p_3, f_3)$ and (p_4, f_4) . We choose three of these four contact pairs, say $(p_1, f_1), (p_2, f_2), (p_3, f_3)$, and slide \hat{O} while maintaining these three contact pairs, and having the fourth point p_4 penetrate the octahedron. The three contact pairs $(p_1, f_1), (p_2, f_2)$ and (p_3, f_3) determine a line in the space \mathbb{P} of placements, and we just have to follow this line in the (unique) direction where p_4 penetrates into the octahedron. Let us add to O the three internal square facets $u_1u_2\overline{u_1u_2}, u_1u_3\overline{u_1u_3}$, and $u_2u_3\overline{u_2u_3}$ (see Figure 2); these are the intersections of the octahedron with its three symmetry planes, each containing four vertices of O . In the following, we refer to the octahedron augmented with these three internal facets as the *augmented octahedron*. The sliding process is stopped as soon as one of the following events occurs:

1. Point p_4 reaches one of the three internal facets.
2. One of the points p_1, p_2, p_3 reaches an edge of O and thus becomes a double contact point.
3. A contact with a new point is encountered on a face other than f_4 .
4. A contact with a new point on face f_4 is encountered.

In the first case, we reach a rigid placement of the *augmented octahedron*. This rigid placement will be called quasi-free because it has no point of \mathcal{S} inside the octahedron, except for one point on an internal facet. Consider the number of pairs (v, f) , where f is one of the contact facets and v is a vertex of O incident to f . Since we have three triangular contact facets and one quadrangular contact facet, the number of these pairs is 13, which implies that one of the six vertices of O has to be shared by three of those contact facets. Since f_1, f_2 and f_3 do not share a common vertex, one of the contact facets sharing the common vertex has to be the internal facet reached by p_4 . We can thus apply to this placement the shrinking scheme used in the proof of Lemma 6.3, retaining the three contact pairs whose facets share the common vertex, and stopping when we reach a quasi-free rigid placement of O with a double contact point. As argued above, such a terminal placement can be reached by at most two initial quasi-free rigid placements of O .

To bound the number of these terminal placements, we proceed as above. The placements that achieve the double contact (p, ϵ) belong to a 2-dimensional subspace $\mathbb{P}(p, \epsilon)$ of \mathbb{P} . In this subspace, the locus of all placements with an additional contact pair (with an external or an internal facet) is a line segment. The quasi-free placements that we are interested in appear as vertices of the arrangement of these segments lying at level at most 4 (i.e., with at most 4 segments lying below the vertex in the ρ direction). Indeed, if we fix the position of one endpoint of \hat{e} on the line parallel to ϵ and passing through p , and increase the scale factor ρ from zero until we reach a terminal placement, the point on the internal facet could have crossed at most two external facets of O (when it gets into the octahedron through an edge; this point cannot get into the octahedron through a vertex because the set of points would not be then in general position) and two internal facets (because it cannot cross the internal facet incident to edge ϵ). Using standard arguments (based on the Clarkson-Shor analysis technique [6]), it is easily seen that the number of such vertices is $O(n)$. Hence the number of stopping events of the first type is $O(n^2)$.

In the second stopping case, the reached placement is a rigid placement of the (non-augmented) octahedron with a double contact pair and at most one point inside the octahedron. Arguing as in the preceding paragraph, it is easily seen that such a placement corresponds to a vertex of level at most 2 in the planar arrangement of segments representing the contact pairs (of the non-augmented octahedron) in the two-dimensional subspace of \mathbb{P} associated with the reached double contact pair. Thus the number of placements reached in this case is also bounded by $O(n^2)$.

In the third case, the reached placement is a rigid placement of the octahedron with three contact facets sharing a vertex and at most one point inside the octahedron. Again, arguing as above and applying the Clarkson-Shor technique, it follows that the number of terminal placements that we reach in this case is proportional to the number of free rigid placements of type (P2). Thus the number of placements reached in the third case is also $O(n^2)$.

In the last case, the reached placement is a rigid placement of the octahedron with four contact pairs involving four nonadjacent facets and one point inside the octahedron. In the following, we denote by $c_j(\mathcal{S})$ the number of rigid placements \hat{O} with four contact pairs involving four nonadjacent facets and with j points of \mathcal{S} inside the octahedron.

Before continuing, it is important to observe that each terminal placement reached in case 4 of the above sliding process is reached from a unique initial rigid free placement. Indeed, since the single point p_4 inside the octahedron did not cross during the sliding process any internal facet of the augmented octahedron, it lies in one of the eight octants into which the three internal facets partition O , and the external facet bounding that octant must be the contact facet f_4 , so the initial placement \hat{O} is uniquely determined.

On the other hand, we have four choices of the triple of the contact pairs that are preserved in the sliding process. If in one of these choices we reach a terminal placement of one of the first three types, we charge the initial free rigid placement to

this terminal placement, observe that any such terminal placement can be charged in this manner only a constant number of times, and thus conclude that the number of initial free rigid placements of this kind is $O(n^2)$. If each of the four sliding processes terminate in a placement of type 4, then the initial free rigid placement can be charged to four terminal placements. Moreover, every initial and terminal placement in this case involves four contact pairs with four nonadjacent contact facets, except that the initial placements are free and the terminal placements contain a point inside the octahedron.

Thus, the preceding case analysis leads to the following recurrence relationship:

$$4c_0(\mathcal{S}) \leq c_1(\mathcal{S}) + O(n^2),$$

from which we obtain

$$c_0(\mathcal{S}) \leq \frac{n-4}{n}c_0(\mathcal{S}) + \frac{1}{n}c_1(\mathcal{S}) + O(n).$$

Now, $\frac{n-4}{n}c_0(\mathcal{S}) + \frac{1}{n}c_1(\mathcal{S})$ is just the expected number of free rigid placements with four contact pairs involving four nonadjacent facets, for a random sample \mathcal{R} of $n-1$ points of \mathcal{S} ; see [5, 19], and the analysis in Section 4 for similar arguments. Thus, if we denote by $c_0(n)$ the maximum of $c_0(\mathcal{S})$ over all sets \mathcal{S} of n points in general position with respect to O , we obtain the recurrence

$$c_0(n) \leq c_0(n-1) + O(n),$$

whose solution is

$$c_0(n) = O(n^2).$$

Thus, the number of free rigid placements with four contact pairs involving four nonadjacent facets is also bounded by $O(n^2)$, which thus completes the proof of the upper bound in Theorem 6.1.

Remark: An obvious open problem is to extend this result to higher dimensions. Informally, the reason we have failed in doing so is that, for $d \geq 4$, the d -co-cube has a large number of facets (that is, 2^d facets). Consequently, there are too many combinatorially-different types of free rigid placements of the d -co-cube, which so far impeded a successful analysis of their number. A first goal in this direction would be to obtain a sharp bound on the complexity of the L_1 -Voronoi diagram in four dimensions. Note also that our analysis relies on the special structure of the regular octahedron, e.g., in excluding free rigid placements of type (a), so, at present, we do not have a quadratic bound for the case of a non-regular octahedron in 3-space.

6.2 The lower bound

As in subsections 4.2 and 5.2, a configuration that attains the lower bound in Theorem 6.1 can be built by choosing points on two lines in 3-space such that, for any

subset of four points, consisting of a pair of consecutive points on each line, there is a free placement of O in contact with these points.

The first line δ_1 is the line $y = 0, z = 1$, parallel to the x axis; the set of points on δ_1 is taken to be $\mathcal{S}_1 = \{p_h = (\frac{h}{n}, 0, 1) : h = 0, \dots, n-1\}$. The second line δ_2 is the line $x = 0, z = -1$, parallel to the y axis; the set of points on δ_2 is taken to be $\mathcal{S}_2 = \{q_k = (0, \frac{k}{n}, -1) : k = 0, \dots, n-1\}$. See the left part of Figure 3. Let $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$. First consider the two-dimensional L_1 -Voronoi diagram of each subset $\mathcal{S}_1, \mathcal{S}_2$, within the respective planes $z = 1$ and $z = -1$. Let O_1 (resp. O_{-1}) denote the 2-co-cube within the plane $z = 1$ (resp. $z = -1$). In the plane $z = 1$, the line $x = \frac{h+1/2}{n}, z = 1$, for each $0 \leq h \leq n-2$, is the bisecting line of the pair (p_h, p_{h+1}) and the locus of the centers of free placements of O_1 with p_h and p_{h+1} as contact points. Similarly, in the plane $z = -1$, the line $y = \frac{k+1/2}{n}, z = -1$, for each $0 \leq k \leq n-2$, is the bisector of the pair (q_k, q_{k+1}) and the locus of the centers of free placements of O_{-1} with q_k and q_{k+1} as contact points. Thus, for each pair (k, h) , with $0 \leq k, h \leq n-2$, there is a free placement $\hat{O}_1(h, k)$ of O_1 , which is centered at the point $(\frac{h+1/2}{n}, \frac{k+1/2}{n}, 1)$ and touches the points p_h and p_{h+1} , and there is a free placement $\hat{O}_{-1}(h, k)$ of O_{-1} , which is centered at the point $(\frac{h+1/2}{n}, \frac{k+1/2}{n}, -1)$ and touches the points q_k and q_{k+1} . Such placements $\hat{O}_1(h, k)$ and $\hat{O}_{-1}(h, k)$ are drawn in dotted lines in the left part of Figure 3. The scale factors of these placements are, respectively, $\rho_1 = \frac{k+1}{n}$ and $\rho_{-1} = \frac{h+1}{n}$. Since $|\rho_1 - \rho_{-1}| \leq 2$, it is easily verified that the two placements $\hat{O}_1(h, k)$ and $\hat{O}_{-1}(h, k)$ are cross-sections of a placement \hat{O} of the three-dimensional octahedron, centered at the point $(\frac{h+1/2}{n}, \frac{k+1/2}{n}, \frac{k-h}{2n})$, and scaled by the factor $1 + \frac{\rho_1 + \rho_{-1}}{2} = 1 + \frac{2+h+k}{2n}$. The right part of Figure 3 shows the cross-section of the placement \hat{O} by the plane $y = \frac{k+1/2}{n}$.

Thus, for each of the $(n-1)^2$ pair (k, h) with $0 \leq k, h \leq n-2$, there is a free rigid placement of the octahedron O among the set \mathcal{S} of $2n$ points, which touches the points p_h, p_{h+1}, q_k , and q_{k+1} ; this proves the lower bound of Theorem 6.1.

7 Degenerate Configurations

Next, we show that the general position assumption is essential for the upper bounds of Theorems 4.1, 5.1, and 6.1 to hold. Specifically, we show:

Theorem 7.1 *For any polyhedral convex distance function d_P and any dimension $d \geq 2$, there exist sets \mathcal{S} of n points in \mathbb{R}^d , not in general position with respect to the distance d_P , whose Voronoi diagrams $\text{Vor}_P(\mathcal{S})$ have complexity $\Omega(n^d)$.*

Figure 4 shows the L_1 -Voronoi diagram of a degenerate set of points in \mathbb{R}^2 .

Proof: Let d_P be a convex distance function, and let \mathcal{S} be a set of n points contained in a hyperplane $H \subset \mathbb{R}^d$ parallel to a facet f of P , such that \mathcal{S} is in general position with respect to the distance function induced by f in H . For each $x \in \mathcal{S}$, the locus

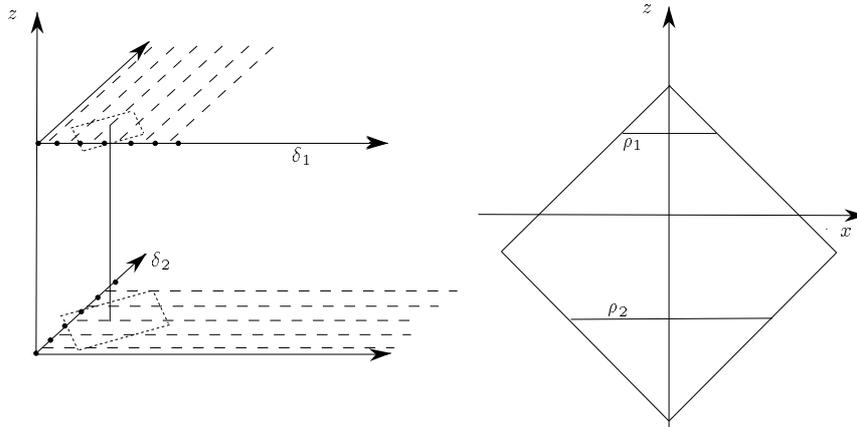


Figure 3: The $\Omega(n^2)$ lower bound construction for L_1 -Voronoi diagrams in \mathbb{R}^3

of the centers of the placements \hat{P} of P for which x lies on \hat{f} is a polyhedral cone C_x with apex at x . All the cones C_x are translates of each other, and, because of the general position of \mathcal{S} with respect to f , the complexity of the arrangement \mathcal{A} of these cones is $\Omega(n^d)$. Each cell c of \mathcal{A} has the property that all maximal free placements of P centered at points of c touch the same subset of points of \mathcal{S} . Moreover, it is easily verified that there are $\Omega(n^d)$ distinct subsets of this kind. This clearly implies the assertion of the theorem. \square

8 Algorithms

In this section we present an efficient algorithm for constructing the L_∞ -Voronoi diagram of a set \mathcal{S} of point sites in \mathbb{R}^d . The algorithm is incremental and on-line, that is, it adds the sites one by one, and maintains the Voronoi diagram of the set of all the already inserted sites; it does not require previous knowledge of the whole set \mathcal{S} . The algorithm uses the method of the *history graph* described in [4] (see also [9]). We show that if the sites are inserted in random order then the expected running time of the algorithm is $O(n^{\lceil d/2 \rceil} \log^{d-1} n)$. A simple modification of the technique yields a randomized algorithm for constructing Voronoi diagrams under simplicial distance functions, whose expected running time is $O(n^{\lceil d/2 \rceil} + n \log n)$.

8.1 Algorithm for L_∞ -Voronoi diagrams

We subdivide the Voronoi cell of each site x into $2d$ subcells, one for each of the $2d$ facets of the hypercube, where the subcell corresponding to the facet f consists of the centers of all maximal free placements having the contact pair (x, f) . For example,

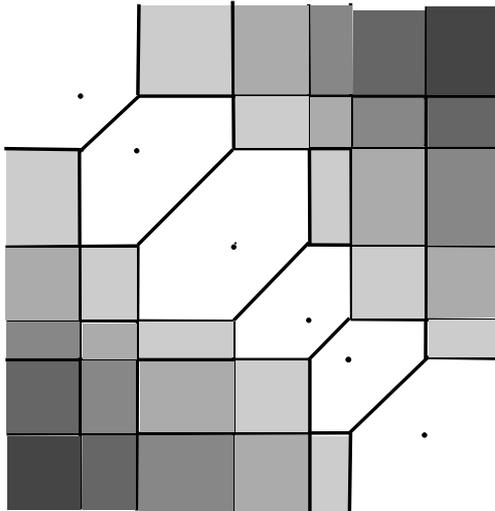


Figure 4: A degenerate configuration for the L_1 metric

the Voronoi diagram of a single site x subdivides the whole space into $2d$ polyhedral cones with apex at x . This subdivision increases the overall complexity of the diagram by only a constant factor (depending on d).

For technical reasons, and for simplicity of the presentation, we prefer not to treat explicitly the unbounded faces of the diagram. This will be done by surrounding \mathcal{S} with additional *sentinel sites*, so that all Voronoi cells of the sites in \mathcal{S} become bounded, and no original Voronoi vertex is lost. This technical issue will be discussed in detail in the description of the initial phase of the algorithm, given below.

Before describing the algorithm itself, it is worth observing some facts about the faces of the diagram. Each k -face Φ of the diagram can be described as a connected component of the locus of the centers of those maximal free placements that realize a given set of $d + 1 - k$ contact pairs. We refer to these contact pairs as the contact pairs of Φ . We distinguish between two types of Voronoi faces (of dimension ≥ 1): (i) *sliding faces*, whose sets of contact pairs include two parallel contact pairs, and (ii) *shrinking faces*, whose sets of contact pairs involve facets of the cube which are all incident to some common vertex of the cube.

Lemma 8.1 *Each subface of a sliding face is a sliding face, and all the maximal free placements centered on a sliding face have the same scaling factor. A sliding edge is parallel to some coordinate axis, and, more generally, higher-dimensional sliding faces are axis-parallel polyhedra.*

Proof: The set of contact pairs of a subface Φ' of a Voronoi face Φ is a superset of the set of contact pairs of Φ . Moreover, the scaling factor of a maximal placement centered on a sliding face is determined by the two parallel contact pairs. This proves the first claim. The second claim follows from the fact that, as in the analysis of

the complexity of the diagram, a sliding edge e can be identified with an edge of the union of axis-parallel hypercubes (whose size is equal to the fixed scaling factor associated with e), and thus must be parallel to some coordinate axis. A similar argument applies to higher-dimensional sliding faces. \square

It is possible that several faces of the diagram have the same set of contact pairs. However, Lemma 8.3 below will show that this is not true for edges (1-faces). For higher-dimensional faces, one can show that there is only a constant number of Voronoi faces with a given set of contact pairs. We omit the proof of this property since it is not used in the analysis of the algorithm.

The fact that there is a unique Voronoi edge for each given set of d contact pairs is a consequence of the following lemma, which considers sliding faces of any dimension.

Let \mathcal{Q} be a set of k contact pairs, including two parallel contact pairs, and let $L_{\mathcal{Q}}$ denote the locus of the centers of the free maximal placements that realize the contact pairs of \mathcal{Q} . Note that \mathcal{Q} determines the scaling factor $\rho_{\mathcal{Q}}$ of these placements and $k-1$ coordinates of their centers. Let $H_{\mathcal{Q}}$ denote the $(d-k+1)$ -flat that contains the centers of these placements. Let $\mathcal{S}_{\mathcal{Q}}$ denote the subset of points of \mathcal{S} appearing in the contact pairs of \mathcal{Q} , let $I_{\mathcal{Q}}$ denote the intersection of the placements with size $\rho_{\mathcal{Q}}$ centered at the points of $\mathcal{S}_{\mathcal{Q}}$, and let $U_{\mathcal{Q}}$ denote the union of the placements with size $\rho_{\mathcal{Q}}$ centered at the points of $\mathcal{S} \setminus \mathcal{S}_{\mathcal{Q}}$.

Lemma 8.2 *The locus $L_{\mathcal{Q}}$ of the centers of the maximal free placements that realize a set \mathcal{Q} of contact pairs (which include two parallel contact pairs) is, in the above notation, $H_{\mathcal{Q}} \cap (I_{\mathcal{Q}} \setminus U_{\mathcal{Q}})$.*

Proof: Let $\hat{C} = \hat{C}(c, \rho_{\mathcal{Q}})$ be a maximal free placement whose center c is in $L_{\mathcal{Q}}$. Then clearly $c \in H_{\mathcal{Q}}$, and \hat{C} contains all points of $\mathcal{S}_{\mathcal{Q}}$ on its boundary, and does not meet any point in $\mathcal{S} \setminus \mathcal{S}_{\mathcal{Q}}$. This implies that c must lie on the boundary of all cubes $\hat{C}(s, \rho_{\mathcal{Q}})$, for $s \in \mathcal{S}_{\mathcal{Q}}$, and outside all cubes $\hat{C}(s, \rho_{\mathcal{Q}})$, for $s \in \mathcal{S} \setminus \mathcal{S}_{\mathcal{Q}}$. Hence we have $L_{\mathcal{Q}} \subseteq H_{\mathcal{Q}} \cap (I_{\mathcal{Q}} \setminus U_{\mathcal{Q}})$. The converse containment is proved in much the same manner, observing that if $c \in H_{\mathcal{Q}} \cap I_{\mathcal{Q}}$ then c must lie on the boundary of all cubes $\hat{C}(s, \rho_{\mathcal{Q}})$, for $s \in \mathcal{S}_{\mathcal{Q}}$, which, together with the fact that $c \notin U_{\mathcal{Q}}$, implies that $\hat{C}(c, \rho_{\mathcal{Q}})$ is a maximal free placement that realizes the set \mathcal{Q} of contact pairs. \square

Lemma 8.3 *The locus of centers of all maximal free placements that realize a given set of d contact pairs is a line segment.*

Proof: If the set \mathcal{Q} of d contact pairs contains two parallel contact pairs, then the result follows from the previous lemma. Indeed, in that case, $H_{\mathcal{Q}}$ is an axis-parallel line, $H_{\mathcal{Q}} \cap I_{\mathcal{Q}}$ is the intersection of some line segments of length $\rho_{\mathcal{Q}}$, and $H_{\mathcal{Q}} \cap U_{\mathcal{Q}}$ is the union of some other line segments with the same length $\rho_{\mathcal{Q}}$. This is easily seen to imply that $L_{\mathcal{Q}}$ cannot have more than one connected component.

If \mathcal{Q} does not contain two parallel contact pairs, the faces of the cube involved in the contact pairs of \mathcal{Q} share a common vertex v . Moreover, the location \hat{v} of that vertex must be the same for all maximal free placements whose centers lie in $L_{\mathcal{Q}}$, and all these placements can be obtained from one another by a homothety whose center is \hat{v} . Hence, for any pair of such placements, the larger placement contains the smaller one, and the locus $L_{\mathcal{Q}}$ of their centers is contained in a line (passing through \hat{v} and parallel to some vector of the form $(\pm 1, \pm 1, \dots, \pm 1)$). Moreover, the above nesting property is easily seen to imply that $L_{\mathcal{Q}}$ cannot contain more than one connected component, and is thus a line segment. \square

It also follows from Lemma 8.2 and from the proof of Lemma 8.3 that a line parallel to a coordinate axis and contained in the affine hull of a sliding face f , either misses f or intersects f along a single line segment. The maximal free placements centered on such a segment (or on a sliding edge) are obtained from any one of them by an axis-parallel translation and have a nonempty intersection. Moreover, these observations also imply:

Lemma 8.4 *Each of the maximal free placements centered on a sliding edge e or, more generally, on any axis-parallel line segment e contained in a sliding face, is contained in the union of the maximal free placements centered at the endpoints of e .*

The following lemma is a consequence of the proof of Lemma 8.3:

Lemma 8.5 *Each of the maximal free placements centered on a shrinking edge e is contained in the placement centered at one of the endpoints v of e . The set \mathcal{Q}_v of contact pairs defining v involves one more site than the set \mathcal{Q}_e of the contact pairs defining e . All such placements contain the maximal free placement centered at the other endpoint v' of e . The set $\mathcal{Q}_{v'}$ of the contact pairs defining v' involves the same sites as \mathcal{Q}_e but the multiplicity of one of the sites in $\mathcal{Q}_{v'}$ is one more than its multiplicity in \mathcal{Q}_e .*

The algorithm builds incrementally the 1-skeleton of the Voronoi diagram, i.e., the set of vertices and edges of the diagram, together with their incidence relations. In addition, each vertex and edge is represented with its set of contact pairs. Lemma 8.2 is easily seen to imply that each sliding face is homeomorphic to a ball of the appropriate dimension. A similar result can be proved for shrinking faces. Thus it is easy to construct, in time linear in the output size, the full set of Voronoi faces, together with their incidence structure, from the final 1-skeleton. We omit details of this final construction step.

The algorithm maintains the following data structures:

- a history graph;
- a dictionary D , containing one entry for each set of contact pairs associated with a sliding face constructed by the algorithm;

- a multi-level dynamic segment tree data structure associated with each entry in D , which supports efficient ray-shooting queries along some fixed axis-parallel direction within each sliding face.

The history graph is a directed acyclic graph, each node of which represents a rigid placement of an axis-parallel hypercube, so that, during some stage of the incremental algorithm, this placement has been maximally free and thus centered at a Voronoi vertex. The graph has the property that the placement associated with a node is contained in the union of the placements associated with its parents, a property that we refer to hereafter as the *inclusion property*. Furthermore, each vertex of the current Voronoi diagram is linked via a double pointer to the node of the history graph corresponding to the same placement. See [4, 9] for earlier uses of similar history graphs.

When a new site s is inserted, it may be contained in some of the maximal free placements centered at the current Voronoi vertices. These placements are no longer free, and the corresponding Voronoi vertices and history graph nodes are said to be *killed* by s . The insertion of s generates new Voronoi vertices. A new node is created in the history graph for each new Voronoi vertex; this node is made a child of some of the older nodes in the graph, which represent free placements in the current diagram, whose associated placements overlap the placement associated with the new node, in a manner that ensures the inclusion property. The precise manner in which this is accomplished is described in detail below. A parent node may or may not have been killed by the insertion of s .

A node of the history graph, the corresponding Voronoi vertex, and the corresponding placement, are said to be *in conflict* with a site s if this placement of the hypercube contains s in its interior.

Each entry in the dictionary D corresponds to a set \mathcal{Q} of contact pairs, including two parallel contact pairs, that appeared, at some stage of the algorithm, as the set of contact pairs of some sliding faces of the current diagram.

Let $F_{\mathcal{Q}}$ be the union of the sliding faces whose set of contact pairs is \mathcal{Q} . Each entry in D points to a dynamic ray-shooting structure $\mathcal{R}_{\mathcal{Q}}$ that supports fast ray-shooting queries in some fixed axis-parallel direction within the current version of $F_{\mathcal{Q}}$. The ray-shooting structures are based on standard (dynamic) multi-level segment trees [13], and are described in more detail later.

For technical reasons, and for simplicity of the presentation, we prefer not to treat explicitly the faces at infinity of the diagram. Therefore, we introduce a set \mathcal{S}_0 of additional sites, called *sentinel sites*, and first compute the Voronoi diagram of $\mathcal{S} \cup \mathcal{S}_0$, from which the diagram of \mathcal{S} is easy to derive. The sentinel sites are chosen in such a way that each site of \mathcal{S} has a bounded cell in $\text{Vor}_{\infty}(\mathcal{S} \cup \mathcal{S}_0)$, and each vertex of $\text{Vor}_{\infty}(\mathcal{S})$ is a vertex of $\text{Vor}_{\infty}(\mathcal{S} \cup \mathcal{S}_0)$. We initialize the algorithm with the Voronoi diagram of \mathcal{S}_0 , and then insert the sites of \mathcal{S} . In this way, no unbounded faces will be created during the incremental insertion stages. In what follows, we assume that the sites of \mathcal{S} lie within a large axis-parallel hypercube \hat{C}_0

centered at the origin of the coordinate system. The sentinel sites are chosen to be the 2^d vertices of \hat{C}_0 ; see Figure 5 for an illustration. Any placement of the hypercube C that intersects \hat{C}_0 and has a sufficiently large scaling factor must contain one of the sentinel sites. It follows that each site of \mathcal{S} has a bounded cell in the diagram $\text{Vor}_\infty(\mathcal{S} \cup \mathcal{S}_0)$. Moreover, it is easily checked that any free rigid placement whose contact pairs involve only sites of \mathcal{S} does not contain any sentinel site. Hence, each vertex of $\text{Vor}_\infty(\mathcal{S})$ is also a vertex of $\text{Vor}_\infty(\mathcal{S} \cup \mathcal{S}_0)$, so the chosen set of sentinel sites has the required properties. Furthermore, it is easy to check that the union of the maximal free placements centered at the vertices of $\text{Vor}_\infty(\mathcal{S}_0)$ contains all the free rigid placements of the hypercube among any subset of $\mathcal{S} \cup \mathcal{S}_0$ that contains \mathcal{S}_0 . Thus it will always be possible to ensure the inclusion property.

Initial Step

The data structures are initialized with the Voronoi diagram (shown in Figure 5) of the 2^d sentinel points. Each vertex of this diagram is associated with a node of the history graph which is a child of the root node.

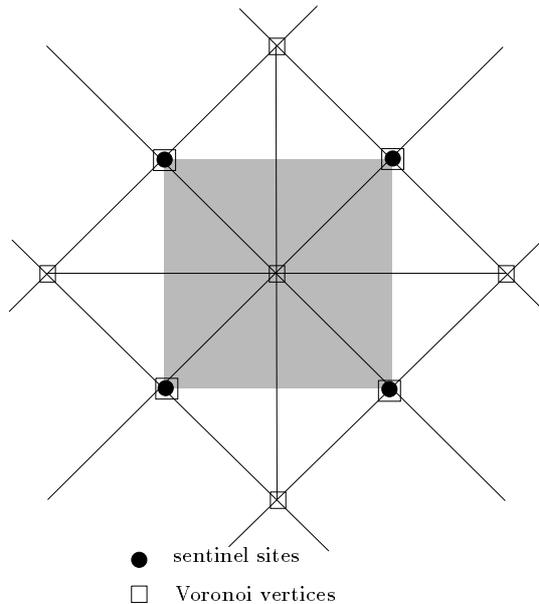


Figure 5: Initialization of the L_∞ -Voronoi diagram; if the sites of \mathcal{S} all lie in the shaded region, their Voronoi cells are all bounded.

Incremental Updating of the Diagram

Each subsequent step inserts a new site $x \in \mathcal{S}$ into the diagram. Hereafter, the Voronoi diagram before the insertion of x will be called the *current* diagram, while

the diagram after the insertion of x will be called the *updated* diagram. Each vertex or edge of the current Voronoi diagram that is not a feature of the updated diagram is said to be *killed* (as above), while each vertex or edge of the updated diagram that is not a feature of the current diagram is called *new*. The following substeps are performed:

The **first substep** identifies the killed vertices and edges. The killed vertices are identified by a traversal of the history graph. This traversal starts at the root node of the graph, and then visits all the nodes conflicting with x and their children, backtracking at each node that is not in conflict with x .

Next, we scan the 1-skeleton of the current diagram to identify all the killed edges. Observe that each killed edge e must be incident to at least one killed vertex, because, by Lemmas 8.4 and 8.5, each maximal free placement centered on an edge e is contained in the union of the maximal free placements centered at the vertices of e . Hence, if some maximal free placement, centered at some point on e , is in conflict with x , then at least one endpoint of e must be a killed vertex. The killed edges and vertices are discarded from the 1-skeleton.

The next two substeps create the new vertices and edges of the Voronoi diagram. For each new vertex, a new node is added to the history graph in such a way that its incoming arcs satisfy the inclusion property. The last substep updates the dictionary D by creating new entries for the sets of contact pairs of the new sliding faces that do not correspond to already existing entries of D . In addition, this substep creates the ray-shooting structures associated with these new entries, and updates the structures for the old entries.

In accordance with the definition in Section 2, we say that a placement or some Voronoi face has *x -multiplicity k* if it has k contact pairs involving x .

The **second substep** creates the new vertices with a simple contact at x , and the new edges with no contact at x . To do this, we consider in turn each killed edge e of the current diagram that is incident to only one killed vertex. Then x is contained in only some of the maximal free placements centered on e (whose centers form a connected portion of e , by Lemma 8.3). The edge e is replaced by a new and shorter edge e' that joins the non-discarded vertex of e with a new vertex v whose placement has a simple contact at x . A new node, corresponding to v , is appended to the history graph. This node becomes a child of the discarded vertex of e if e is a shrinking edge, or a child of both vertices of e if e is a sliding edge. This guarantees the inclusion property for v by Lemmas 8.4 and 8.5. The above process provides all the new vertices of the updated Voronoi diagram with a simple contact at x , as well as all the new edges with no contact at x . Indeed, any new vertex with a simple contact at x and any point on a new edge with no contact at x is the center of a maximal free placement with d contact pairs involving previously inserted sites, and thus belongs to a Voronoi edge of the current diagram, so it will be found by the above procedure.

The **third substep** proceeds by induction on the x -multiplicity of vertices and edges, to create new Voronoi vertices and edges with higher x -multiplicity. Assume that all

new vertices and edges with x -multiplicity up to k and $k - 1$, respectively, are known. Then the algorithm creates the new Voronoi vertices and edges with x -multiplicities $k + 1$ and k , respectively, as follows. Each Voronoi vertex v with x -multiplicity k is incident to $d + 1 - k$ Voronoi edges with x -multiplicity k , whose sets of contact pairs can be obtained, in constant time, from the contact pairs of v , by relaxing each of the $d + 1 - k$ contact pairs not involving x . Moreover, we have:

Lemma 8.6 *If e is a Voronoi edge with x -multiplicity k , then at least one endpoint of e is a vertex with x -multiplicity k .*

Proof: Assume to the contrary that there exists an edge e that does not satisfy this property; that is, both endpoints of e have x -multiplicity $k + 1$. Suppose first that e is a sliding edge, parallel to some coordinate axis x_j . Then the two new contact pairs at the endpoints v_1, v_2 of e must be (x, f_1) and (x, f_2) , where f_1 and f_2 are the two facets of the hypercube orthogonal to the x_j -axis. If ρ_e is the common scaling factor of the maximal free placements of the hypercube centered on e , then the length of e must be $2\rho_e$. Since e is a sliding edge, it must involve some contact pair (y, f) , for $y \neq x$. But then, by the general position assumption, the x_j -coordinates of x and y are different, which implies that the length of e must be smaller than $2\rho_e$, a contradiction that establishes the asserted property.

If e is a shrinking edge, then all contact pairs of e involving x are of the form (x, f) , where all those facets f have a common vertex w of the hypercube, and all maximal free placements centered on e are obtained by shrinking or expanding the hypercube with respect to w . This is easily seen to imply that we can gain a new contact pair involving x only when the hypercube is shrinking, but not when it is expanding. Again, this implies the asserted property. \square

Lemma 8.6 implies that all Voronoi edges with x -multiplicity k can be obtained from the Voronoi vertices with x -multiplicity k , by relaxing one contact pair not involving x . More precisely, this procedure generates each such edge e either once (if it has one endpoint with x -multiplicity k and one endpoint with x -multiplicity $k + 1$) or twice (if both endpoints of e have x -multiplicity k). We detect edges that are generated twice using a dictionary data structure. For each edge e that is generated only once, from a vertex v with x -multiplicity k , we compute the other endpoint of e (the one with x -multiplicity $k + 1$) in constant time. For this, we iterate over all possible contact pairs that involve x and are not present in the set \mathcal{Q}_e of the contact pairs of e . We add in turn each such contact pair to \mathcal{Q}_e , compute the position of the center of the corresponding free rigid placement, if such a placement exists, and choose the placement whose center is closest to v . Since each vertex with x -multiplicity $k + 1$ is incident to $k + 1$ edges with x -multiplicity k , the above procedure will produce all Voronoi vertices with x -multiplicity $k + 1$.

It remains to create a new node in the history graph for each new vertex, and to link it to earlier nodes so that the inclusion property still holds. This is done as follows. If the new vertex v' with x -multiplicity $k + 1$ is linked to a vertex v with

x -multiplicity k via a shrinking edge, the node for v' becomes a child of v (cf. Lemma 8.5). Otherwise, by Lemma 8.6, all the edges with x -multiplicity k incident to v' are sliding edges. This means that if we relax any of the contacts involving x in the set of contact pairs defining v' then we preserve the unique pair of parallel contact pairs. This clearly implies that the $d - k$ contact pairs of v' that do not involve x contain the two parallel contact pairs.

It follows that v' is located in a sliding $(k + 1)$ -face f of the current diagram. We use the ray-shooting data structures to find parents for the node associated with v' , as follows. Let \mathcal{Q} be the set of contact pairs of f . The ray-shooting structure associated with the entry for \mathcal{Q} in the dictionary D supports fast ray-shooting queries in some fixed axis-parallel direction within f . We first shoot from v' in the fixed given direction within f , both forward and backwards. Let h and h' be the two subfaces of f hit by these rays, and let w and w' denote the corresponding impact points. We know, by Lemma 8.4, that the maximal free placement centered at v' is contained in the union of the maximal free placements centered at w and at w' . Then we perform similar ray-shootings from w in h and from w' in h' , forward and backwards, along the corresponding shooting directions within those subfaces (note that these new directions are orthogonal to the first shooting direction), using the data structures available for h and h' (recall that a subface of a sliding face is a sliding face). This yields four new impact points on lower-dimensional subfaces, and we keep iterating these shootings until we reach edges of f . By taking the set of endpoints of these edges, we obtain 2^{k+1} Voronoi vertices (of the current diagram), and the node of the history graph associated with v' becomes a child of each of the 2^{k+1} nodes corresponding to these vertices, provided that the corresponding placements overlap. Clearly, this implies that the inclusion property holds for v' .

Finally, the **fourth substep** updates the dictionary D and the ray-shooting structures. First, we have to create an entry in D and a ray-shooting structure for each set of contact pairs of any new sliding face that involves x . These sets of contact pairs are found inductively, by decreasing the cardinality of these sets (i.e., by increasing the dimension of the corresponding faces). That is, we obtain the sets of contact pairs of the new sliding k -faces from the sets of contact pairs of the new sliding $(k - 1)$ -faces, by relaxing one contact pair, as long as the resulting subset of the contact pairs still involves x and still contains two parallel contact pairs. For each such new set \mathcal{Q} of contact pairs, we associate an entry in D and form a list $\mathcal{L}_{\mathcal{Q}}$ of all the contact pairs that have been relaxed to obtain the set \mathcal{Q} from sets of contact pairs of new sliding $(k - 1)$ -faces (that is, this list represents all $(k - 1)$ -subfaces of the new sliding k -face). We then use this list to build the ray-shooting structure $\mathcal{R}_{\mathcal{Q}}$, which facilitates fast ray-shooting queries in some fixed axis-parallel direction within the union of the faces of the updated diagram sharing the contact set \mathcal{Q} . This is done as follows.

Let \mathcal{Q} be the set of contact pairs of some new sliding face, let $\rho_{\mathcal{Q}}$ be the corresponding scaling factor, and let $\mathcal{S}_{\mathcal{Q}}$ be the subset of the sites that are involved in the pairs of \mathcal{Q} . The union $L_{\mathcal{Q}}$ of the faces of the updated diagram, whose common set of contact pairs is \mathcal{Q} , is contained in a k -dimensional flat $H_{\mathcal{Q}}$. We know from Lemma 8.2

that L_Q is the set difference $(I_Q \cap H_Q) \setminus (U_Q \cap H_Q)$, where I_Q is the intersection of the axis-parallel hypercubes of size ρ_Q that are centered at the sites of \mathcal{S}_Q , and U_Q is the union of the axis-parallel hypercubes of size ρ_Q that are centered at the sites of $\mathcal{S} \setminus \mathcal{S}_Q$. (In fact, it is not necessary to consider all the sites of \mathcal{S}_Q and $\mathcal{S} \setminus \mathcal{S}_Q$: Let I'_Q be the intersection of the hypercubes of size ρ_Q centered at the sites of \mathcal{S}_Q that are involved in the contact pairs in the list \mathcal{L}_Q , and let U'_Q be the union of the hypercubes of size ρ_Q centered at the sites of $\mathcal{S} \setminus \mathcal{S}_Q$ that are involved in the contact pairs of \mathcal{L}_Q . Clearly, $L_Q = (I'_Q \cap H_Q) \setminus (U'_Q \cap H_Q)$.)

The k -flat H_Q is parallel to k coordinate axes, say x_1, \dots, x_k . To build \mathcal{R}_Q , we choose a fixed shooting direction parallel to one of those axes, say x_k . The ray-shooting structure is a multi-level data structure that stores the intersections of H_Q with the facets orthogonal to the x_k -axis of the hypercubes of size ρ_Q centered at the sites involved in \mathcal{L}_Q .

In the following description, a *cube* will denote a $(k-1)$ -dimensional hypercube which is the intersection with H_Q of a facet orthogonal to the x_k -axis of one of the hypercubes of size ρ_Q centered at the sites involved in \mathcal{L}_Q . The ray-shooting structure has k levels. The first $k-1$ levels of the structure constitute a multi-dimensional segment tree representing the projections of the cubes onto the $(k-1)$ -subspace spanned by x_1, \dots, x_{k-1} . The last level is a balanced binary tree storing, in the order of the x_k coordinates of their centers, the ‘canonical’ collection of cubes assigned to each node of the $(k-1)$ -st level of the segment tree. Clearly, this structure allows us to compute efficiently the first cube hit by any query ray parallel to the x_k -axis, from which we immediately obtain the contact set of the sliding subface hit by the ray. The cost of a ray-shooting query in this structure is $O(\log^k n)$: Querying the multi-level segment tree takes $O(\log^{k-1} n)$ time [13], and the output of this query is a collection of $O(\log^{k-1} n)$ nodes of the segment tree at the $(k-1)$ -st level. We then have to locate the x_k coordinate of the origin of the query ray in each corresponding binary tree, which takes logarithmic time per node. A cube can be inserted into such a structure also in $O(\log^k n)$ time, as described in [13], and the cost of building \mathcal{R}_Q is thus at most $O(\log^k n)$ times the number of $(k-1)$ -subfaces of the Voronoi faces of the current diagram with the same set Q of contact pairs.

Next, the older ray-shooting structures need also be updated with the appropriate new subfaces induced by x , so the above data structures need to be maintained dynamically. The modification of the structure corresponding to an (old) entry Q amounts to inserting a new cube with scaling factor ρ_Q centered at x . As just argued, this can be done in time $O(\log^k n)$ per update, where k is the dimension of the relevant sliding face.

This completes the description of the algorithm. Putting everything together, we obtain:

Theorem 8.7 *The L_∞ -Voronoi diagram of a set of n points in \mathbb{R}^d in general position can be constructed on-line in randomized expected time $O(n^{\lceil d/2 \rceil} \log^{d-1} n)$.*

Proof: The randomized analysis of this algorithm uses the formalism of objects,

regions, and conflicts, introduced by Clarkson and Shor [6]. The objects are the sites; the regions, each defined by a set of objects of \mathcal{S} , are the rigid placements of a hypercube among the points of \mathcal{S} ; a site conflicts with a region when it belongs to its interior. Each region is fully described by its $d + 1$ contact pairs and is thus triggered by a set of at most $d + 1$ sites. Thus, for any subset $\mathcal{S}' \subseteq \mathcal{S}$, a region defined by a subset of \mathcal{S}' of size $\leq d + 1$ and without any conflict with the sites of \mathcal{S}' is a free rigid placement for \mathcal{S}' . Then, by standard analysis (see, e.g., [4]), the expected number of new Voronoi vertices created at step r (i.e., when inserting the r -th object) is $O(\frac{1}{r}f_0(r))$, where $f_0(r)$ is the expected number of free rigid placements for a random subset of size r of \mathcal{S} . By Theorem 4.1, $f_0(r)$ is $O(r^{\lceil d/2 \rceil})$, and thus the expected total number of Voronoi vertices (and thus also of Voronoi faces of all dimensions) created by the algorithm is $\sum_{r=1}^n O(r^{\lceil d/2 \rceil - 1}) = O(n^{\lceil d/2 \rceil})$.

Let us first ignore the cost of traversing the history graph in the first substep, the cost of ray-shooting queries, the cost of building and maintaining the ray-shooting structures, and the cost of searching and updating the dictionary D . The remaining cost at each step is clearly proportional to the number of Voronoi faces killed or created at that step. (This also applies to the cost of updating the history graph, which follows from the fact that each node in the history graph has a bounded number of parents, so the expected number of arcs in the history graph is proportional to the expected number of nodes.) Hence, except for the items just excluded, the overall expected running time of the rest of the algorithm is proportional to the total number of faces ever created, which is $O(n^{\lceil d/2 \rceil})$.

Let us now analyze the cost entailed in ray shooting. In dimension 2, there is no need for ray shooting data structures, because we only need to shoot along sliding edges, which is trivial. If $d \geq 3$, the number of ray shooting queries is at most proportional to the number of vertices that the algorithm creates, and the total number of subfaces inserted into the ray-shooting structures is at most proportional to the total number of subfaces of the new sliding faces. Each structure has at most $d - 1$ levels, which implies that the cost for each ray-shooting query is $O(\log^{d-1} n)$, and the cost for the construction and updating of the structures is $O(\log^{d-1} n)$ times the number of faces. Hence, the total expected cost entailed in ray shooting is $O(n^{\lceil d/2 \rceil} \log^{d-1} n)$.

Consider next the cost of handling the dictionary. In dimension 2, there is no need for a dictionary. If $d \geq 3$, the number of operations (insertions and queries) performed in the dictionary D storing the sets of contact pairs of sliding faces is proportional to the number of sliding faces (of all dimensions) that have been created. It follows that the total expected cost entailed in handling the dictionary is $O(n^{\lceil d/2 \rceil} \log n)$.

Finally, we estimate the expected cost of traversing the history graph. This can be done using the notion of *biregions*, as in [4]. A biregion is defined as a pair of regions that can appear as a parent-child pair in the history graph. More precisely, a biregion is a pair (\hat{C}_1, \hat{C}_2) , where \hat{C}_1, \hat{C}_2 are two placements of the hypercube, such that there exists a subset \mathcal{S}^* of \mathcal{S} and a site $x \in \mathcal{S} \setminus \mathcal{S}^*$ such that (i) \hat{C}_2 is a free rigid placement in the set $\mathcal{S}^* \cup \{x\}$ and contains x on its boundary, (ii) \hat{C}_1 is a free rigid placement in the set \mathcal{S}^* , which overlaps \hat{C}_2 and may or may not contain x in its interior, and either

(iii.a) \hat{C}_1 and \hat{C}_2 share a common vertex and \hat{C}_2 is contained in \hat{C}_1 , or (iii.b) \hat{C}_1 and \hat{C}_2 have the same size ρ , and there exists a sequence $\hat{C}_2 = \hat{C}^{(1)}, \hat{C}^{(2)}, \dots, \hat{C}^{(j)} = \hat{C}_1$ of hypercube placements of size ρ , such that all of them are free in the set \mathcal{S}^* , each placement $\hat{C}^{(i+1)}$ is obtained from the preceding placement $\hat{C}^{(i)}$ by translation along some axis-parallel direction (during which the hypercube remains free), and $\hat{C}^{(i+1)}$ has one more contact pair than $\hat{C}^{(i)}$ (relative to the set \mathcal{S}^*).

A site is said to be in conflict with a biregion if it conflicts with one of the two regions forming the biregion (except for the site x that may have killed the parent region and lies on the boundary of the child region). Then the total cost of the graph traversals is at most proportional to the sum of the *weights* of the biregions appearing in the history graph, where the weight of a biregion is the number of sites in conflict with it (recall that an edge of the history graph is traversed only when the new site is in conflict with the parent region of the biregion corresponding to the edge). Using standard analysis, as in [4], one can show that the expected value of this sum of weights is $O(\sum_{r=1}^n \frac{n-r}{r^2} f_0^*(r))$, where $f_0^*(r)$ is the expected number of biregions with no conflicting site in a random sample of r sites. (This bound holds because a biregion is defined in a purely local manner from a bounded number of ‘trigger sites’, which are the at most $2(d+1)$ sites involved in the contact pairs of the two regions. Moreover, any such set of trigger sites determines only a constant number (depending on d) of biregions, as follows from the above definition. This locality condition implies the above bound, as follows easily from the analysis in [4] (see also [6]).) We have $f_0^*(r) = O(r^{\lceil d/2 \rceil})$, since the expected number of regions appearing in the history graph of the sample is $O(r^{\lceil d/2 \rceil})$, and since each such region has $O(1)$ parents. It therefore follows that the above cost is $O(n \log n + n^{\lceil d/2 \rceil})$. \square

8.2 Algorithm for simplicial Voronoi diagrams

The above algorithm can be adapted to yield a similar algorithm for simplicial Voronoi diagrams in any fixed dimension d . This algorithm is in fact much simpler, since simplicial Voronoi diagrams have only shrinking edges, and no sliding faces. Thus there is no need for the dictionary and ray-shooting structures.

As in the case of L_∞ -Voronoi diagrams, we wish to avoid having to deal with unbounded Voronoi edges and faces. To do so, we initialize the construction with the diagram of some set \mathcal{S}_0 of a constant number of *sentinel sites*, construct the diagram of $\mathcal{S}' = \mathcal{S} \cup \mathcal{S}_0$, and then remove the sentinels of \mathcal{S}_0 . The sentinels are the $d+1$ vertices of a simplex $\hat{\sigma}'_0$ obtained as follows: Consider a placement $\hat{\sigma}_0$ of the simplex defining the distance function, which contains the origin of the coordinate system, and let $\hat{\sigma}'_0$ be a homothetic copy of $\hat{\sigma}$ with a negative scaling factor, whose absolute value is chosen to be sufficiently large, so that the simplex $\hat{\sigma}'_0$ contains all the sites of \mathcal{S} . (The σ -Voronoi diagram of the vertices of $\hat{\sigma}'_0$ is illustrated in Figure 6 in the 2-dimensional case.) Each placement $\hat{\sigma}$, which intersects $\hat{\sigma}'_0$ and has a sufficiently large scaling factor, contains one of the sentinels, which implies that each site of \mathcal{S} has a bounded cell in $\text{Vor}_\sigma(\mathcal{S}')$. Each maximal free placement whose set of contact pairs includes only

sites of \mathcal{S} contains no sentinel, because it is contained in the placement $\hat{\sigma}_1$, obtained from $\hat{\sigma}'_0$ through a homothety with scale factor -2 , which has one contact pair with each sentinel. Thus it is easy to recover the Voronoi diagram $\text{Vor}_\sigma(\mathcal{S})$ from $\text{Vor}_\sigma(\mathcal{S}')$.

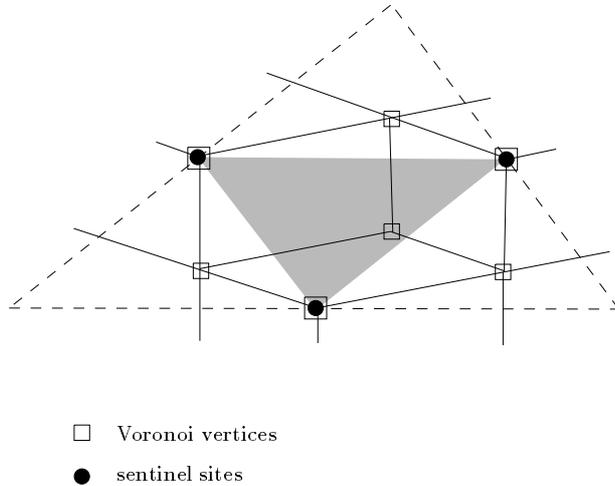


Figure 6: Initialization of a simplicial Voronoi diagram

The randomized analysis of this algorithm is the same as given above, except for the cost of handling the dictionary and the ray-shooting structures, which is simply ignored. Hence we obtain:

Theorem 8.8 *Any simplicial Voronoi diagram of a set of n points in \mathbb{R}^d in general position can be constructed on line in randomized expected time $O(n \log n + n^{\lceil d/2 \rceil})$.*

9 Conclusion

In this paper we have studied the complexity of Voronoi diagrams of point sets in higher dimensions under certain special polyhedral convex distance functions, including simplicial distance functions and the L_1 and L_∞ norms. We have obtained tight worst-case bounds for all the cases that we studied. Some of these bounds match the known maximum complexity of euclidean Voronoi diagrams, namely $\Theta(n^{\lceil d/2 \rceil})$, lending support to the conjecture that this bound holds for fairly general cases of Voronoi diagrams of point sites in higher dimensions, even though it is known not to hold for more general sites [1].

For the simplicial and the L_∞ distance functions, we have presented efficient on-line randomized algorithms, whose expected running times are, respectively, $O(n \log n + n^{\lceil d/2 \rceil})$ and $O(n^{\lceil d/2 \rceil} \log^{d-1} n)$. The first algorithm is thus worst-case optimal, and the second is very close to being worst-case optimal.

There are quite a few open problems that this paper raises. The first problem is to extend the bound obtained for L_1 -Voronoi diagrams to four and higher dimensions. Another, more challenging problem is to extend our analysis to Voronoi diagrams under arbitrary polyhedral convex distance functions. An even more challenging problem is to extend our analysis further to cases where the sites are general convex polytopes, rather than points. (Here, based on [1], the goal is to obtain bounds close to $O(n^{d-1})$.) This is open even in three dimensions (where the goal is to obtain near-quadratic bounds). For this, one would probably need an appropriate combination of the techniques used here and in the previous paper [5].

References

- [1] B. Aronov, personal communication.
- [2] B. Aronov, M. Sharir and B. Tagansky, The union of convex polyhedra in three dimensions, submitted to *SIAM J. Computing*.
- [3] F. Aurenhammer, Voronoi diagrams: a survey of a fundamental geometric data structure, *ACM Comput. Surv.*, 23 (1991), 345–405.
- [4] J.-D. Boissonnat, O. Devillers, R. Schott, M. Teillaud and M. Yvinec, Applications of random sampling to on-line algorithms in computational geometry, *Discrete Comput. Geom.* 8 (1992), 51–71.
- [5] L.P. Chew, K. Kedem, M. Sharir, B. Tagansky and E. Welzl, Voronoi diagrams of lines in three dimensions under a polyhedral convex distance function, *Proc. 6th ACM-SIAM Symp. on Discrete Algorithms*, 1995, 197–204.
- [6] K. Clarkson and P. Shor, Applications of random sampling in computational geometry II, *Discrete Comput. Geom.* 4 (1989), 387–421.
- [7] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.
- [8] H. Edelsbrunner and R. Seidel, Voronoi diagrams and arrangements, *Discrete Comput. Geom.* 1 (1986), 25–44.
- [9] L. Guibas, D.E. Knuth and M. Sharir, Randomized incremental construction of Voronoi and Delaunay diagrams, *Algorithmica* 7 (1992), 381–413.
- [10] D. Halperin and M. Sharir, New bounds for lower envelopes in three dimensions with applications to visibility of terrains, *Discrete Comput. Geom.* 12 (1994), 313–326.
- [11] K. Kedem, R. Livne, J. Pach and M. Sharir, On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles, *Discrete Comput. Geom.* 1 (1986), 59–71.
- [12] D. Leven and M. Sharir, Intersection and proximity problems and Voronoi diagrams, in *Advances in Robotics*, Vol. I, (J. Schwartz and C. Yap, Eds.), 1987, 187–228.
- [13] K. Mehlhorn, *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*, Springer-Verlag, Heidelberg, 1984.
- [14] K. Mulmuley, *Computational Geometry: An Introduction through Randomized Algorithms*, Prentice Hall, New York, 1993.

- [15] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, Cambridge, UK, 1994.
- [16] F. Preparata and M. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.
- [17] R. Seidel, A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons, *Comput. Geom. Theory Appl.* 1 (1991), 51–64.
- [18] M. Sharir, Almost tight upper bounds for lower envelopes in higher dimensions, *Discrete Comput. Geom.* 12 (1994), 327–345.
- [19] B. Tagansky, A new technique for analyzing substructures in arrangements, *Proc. 11th ACM Symp. on Computational Geometry*, 1995, 200–210.