

27 VORONOI DIAGRAMS AND DELAUNAY TRIANGULATIONS

Steven Fortune

INTRODUCTION

The Voronoi diagram of a set of sites partitions space into regions, one per site; the region for a site s consists of all points closer to s than to any other site. The dual of the Voronoi diagram, the Delaunay triangulation, is the unique triangulation such that the circumsphere of every simplex contains no sites in its interior. Voronoi diagrams and Delaunay triangulations have been rediscovered or applied in many areas of mathematics and the natural sciences; they are central topics in computational geometry, with hundreds of papers discussing algorithms and extensions.

Section 27.1 discusses the definition and basic properties in the usual case of point sites in \mathbb{R}^d with the Euclidean metric, while Section 27.2 gives basic algorithms. Some of the many extensions obtained by varying metric, sites, environment, and constraints are discussed in Section 27.3. Section 27.4 finishes with some interesting and nonobvious structural properties of Voronoi diagrams and Delaunay triangulations.

GLOSSARY

Site: A defining object for a Voronoi diagram or Delaunay triangulation. Also generator, source, Voronoi point.

Voronoi face: The set of points for which a single site is closest (or more generally a set of sites is closest). Also Voronoi region, Voronoi cell.

Voronoi diagram: The set of all Voronoi faces. Also Thiessen diagram, Wigner-Seitz diagram, Blum transform, Dirichlet tessellation.

Delaunay triangulation: The unique triangulation of a set of sites such that the circumsphere of each full-dimensional simplex has no sites in its interior.

27.1 POINT SITES IN THE EUCLIDEAN METRIC

See [Aur91, Ede87, For95] for more details and proofs of material in this section.

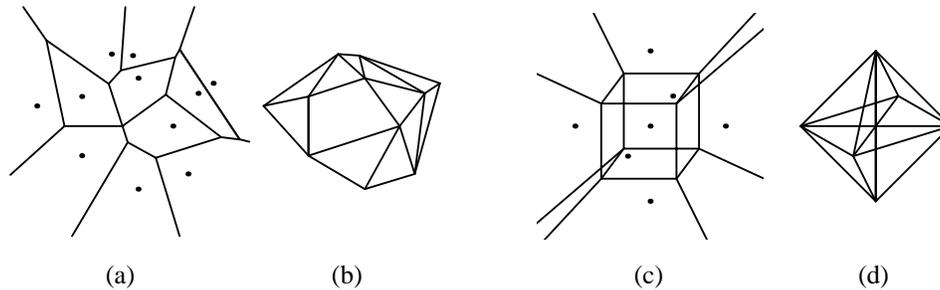
GLOSSARY

Sites: Points in a finite set S in \mathbb{R}^d .

Voronoi face of a site s : The set of all points of \mathbb{R}^d strictly closer to the

FIGURE 27.1.1

Voronoi diagram and Delaunay triangulation of the same set of sites in two dimensions (a,b) and three dimensions (c,d).



site $s \in S$ than to any other site in S . The Voronoi face of a site is always a nonempty, open, convex, full-dimensional subset of \mathbb{R}^d .

Voronoi face $V(T)$ of a subset T : For T a nonempty subset of S , the set of points of \mathbb{R}^d equidistant from all members of T and closer to any member of T than to any member of $S \setminus T$.

Voronoi diagram of S : The collection of all nonempty Voronoi faces $V(T)$, for $T \subseteq S$. The Voronoi diagram forms a cell complex partitioning \mathbb{R}^d . In two dimensions (Figure 27.1.1(a)), the Voronoi face of a site is the interior of a convex, possibly infinite polygon; its boundary consists of **Voronoi edges** (1-dimensional faces) equidistant from two sites and **Voronoi vertices** (0-dimensional faces) equidistant from at least three sites. Figure 27.1.1(c) shows a Voronoi diagram in three dimensions.

Delaunay face $D(T)$ of a subset T : The Delaunay face $D(T)$ is defined for a subset T of S whenever there is a sphere through all the sites of T with all other sites exterior (equivalently, whenever $V(T)$ is not empty). Then $D(T)$ is the (relative) interior of the convex hull of T . For example, in two dimensions (Figure 27.1.1(b)), a **Delaunay triangle** is formed by three sites whose circumcircle is empty and a **Delaunay edge** connects two sites that have an empty circumcircle (in fact, infinitely many empty circumcircles).

Delaunay triangulation of S : The collection of all Delaunay faces. The Delaunay triangulation forms a cell complex partitioning the convex hull of S .

There is an obvious one-one correspondence between the Voronoi diagram and the Delaunay triangulation; it maps the Voronoi face $V(T)$ to the Delaunay face $D(T)$. This correspondence has the property that the sum of the dimensions of $V(T)$ and $D(T)$ is always d . Thus, in two dimensions, $V(T)$ is a Voronoi vertex iff $D(T)$ is an open polygonal region; $V(T)$ is an edge iff $D(T)$ is; $V(T)$ is an open polygonal region iff $D(T)$ is a vertex, i.e., a site. In fact, the 1–1 correspondence is a duality between cell complexes, reversing face ordering: for subsets $T, T' \subseteq S$, $V(T')$ is a face of $V(T)$ iff $D(T)$ is a face of $D(T')$.

The set of sites $S \subset \mathbb{R}^d$ is in **general position** (or is **nondegenerate**) if no $d+2$ points lie on a common d -sphere and no $k+2$ points lie on a common k -flat, for $k < d$. If S is in general position, then the Delaunay triangulation of S is a simplicial complex, and every vertex of the Voronoi diagram is incident to $d+1$ edges in the Delaunay triangulation. If S is not in general position, then Delaunay faces need

not be simplices; for example, the four cocircular sites in Figure 27.1.1(b) form a Delaunay quadrilateral. A *completion* of a Delaunay triangulation is obtained by splitting nonsimplicial faces into simplices without adding new vertices.

RELATION TO CONVEXITY

There is an intimate connection between Delaunay triangulations in \mathbb{R}^d and convex hulls in \mathbb{R}^{d+1} , and between Voronoi diagrams in \mathbb{R}^d and halfspace intersections in \mathbb{R}^{d+1} . To see the connections, consider the special case of $d = 2$. Identify \mathbb{R}^2 with the plane spanned by the first two coordinate axes of \mathbb{R}^3 , and call the third coordinate direction the *vertical* direction.

The *lifting map* $\lambda : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is defined by $\lambda(x_1, x_2) = (x_1, x_2, x_1^2 + x_2^2)$; $\Lambda = \lambda(\mathbb{R}^2)$ is a paraboloid of revolution about the vertical axis. See Figure 27.1.2(a). Let H be the convex hull of the lifted sites $\lambda(S)$.

The Delaunay triangulation of S is exactly the orthogonal projection into \mathbb{R}^2 of the lower faces of H (a face is *lower* if it has a supporting plane with inward normal having positive vertical coordinate). To see this informally, suppose that triangle $\lambda(s)\lambda(t)\lambda(u)$ is a lower facet of H , and that plane P passes through $\lambda(s)\lambda(t)\lambda(u)$. The intersection of P with Λ is an ellipse that projects orthogonally to a circle in \mathbb{R}^2 (Figure 27.1.2(a)). Since all other lifted sites are above the plane, all other unlifted sites are outside the circle, and stu is a Delaunay triangle. The opposite direction, that a Delaunay triangle is a lower facet, is similar.

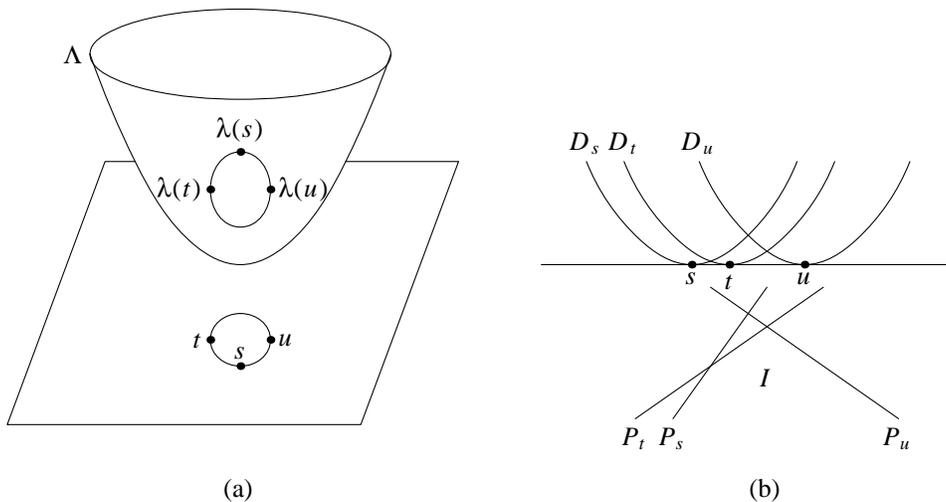
For Voronoi diagrams, assign to each site $s = (s_1, s_2)$ the plane

$$P_s = \{(x_1, x_2, x_3) : x_3 = -2x_1s_1 + s_1^2 - 2x_2s_2 + s_2^2\}.$$

Let I be the intersection of the lower halfspaces of the planes P_s . The Voronoi diagram is exactly the orthogonal projection into \mathbb{R}^2 of the upper faces of I . To

FIGURE 27.1.2

(a) The intersection of a plane with Λ is an ellipse that projects to a circle; (b) on any vertical line, the surfaces $\{D_s\}$ appear in the same order as the planes $\{P_s\}$.



see this informally, consider the surfaces

$$D_s = \{(x_1, x_2, x_3) : x_3 = ((x_1 - s_1)^2 + (x_2 - s_2)^2)\}$$

(see Figure 27.1.2). Viewed as a function from \mathbb{R}^2 into R , D_s gives the squared distance to site s . Furthermore, P_s and D_s differ only by the quadratic term $x_1^2 + x_2^2$, which is independent of s . Hence a point $x \in \mathbb{R}^2$ is in the Voronoi cell of site t iff on the vertical line through x , D_t is lowest among all surfaces $\{D_s\}$. This happens exactly if, on the same line, P_t is lowest among all planes $\{P_s\}$, i.e., x is in the projection of the upper face of I formed by P_t .

COMBINATORIAL COMPLEXITY

In dimension 2, a Voronoi diagram of $n \geq 3$ sites has at most $2n - 5$ vertices and $3n - 6$ edges (and the Delaunay triangulation has at most as many triangles and edges, respectively).

In dimension $d \geq 3$ the Voronoi diagram and Delaunay triangulation can have $\Theta(n^{\lfloor d/2 \rfloor})$ faces. Exact bounds can be given using results from convex polytope theory (Chapter 15). For n sites in d dimensions, the maximum number of Voronoi k -dimensional faces, $k < d$, is $f_{n-k}(C_{d+1}(n)) - \delta_{0k}$, where $C_{d+1}(n)$ is the $d+1$ -dimensional cyclic polytope, f_{n-k} gives the number of $n-k$ dimensional faces (see Section 17.3 and Theorem 17.3.4), and $\delta_{0k} = 1$ if $k = 0$ and 0 otherwise.

For a simple lower bound example in dimension 3, choose $n/2$ distinct point sites on each of two noncoplanar line segments l and l' . Then there is an empty sphere through each quadruple of sites (a, a', b, b') with a, a' adjacent on l and b, b' adjacent on l' . Since there are $\Omega(n^2)$ such quadruples, there are as many Delaunay tetrahedra (and Voronoi vertices).

If point sites are chosen uniformly at random from inside a sphere, then the expected number of faces is linear in the number of sites. In dimension 2, the expected number of Delaunay triangles is $2n$; in dimension 3, the expected number of Delaunay tetrahedra is $\sim 6.77n$; in dimension 4, the expected number of Delaunay 4-simplices is $\sim 31.78n$ [Dwy91]. Similar bounds probably hold for other distributions, but proofs are lacking.

Subquadratic bounds on the complexity of the Delaunay triangulation of point sites in \mathbb{R}^3 can be obtained in a few cases. The *spread (of points)* of a set of points is the ratio between largest and smallest interpoint distances. A point set in \mathbb{R}^3 of size n with spread Δ can have at most $O(\Delta^3)$ Delaunay tetrahedra, for all $\Delta = O(\sqrt{n})$ [Eri05]. Thus if the point set is dense, i.e., has spread $O(n^{1/3})$, there are only $O(n)$ tetrahedra. Points chosen on a surface can also have subquadratic complexity. For example, if the surface is sufficiently continuous and satisfies mild genericity conditions, then any (ϵ, κ) -sample of n points on the surface has complexity $O(n \log n)$ [ABL03]. A set of points is an (ϵ, κ) -sample if for any point in the set, there is at least one and at most κ other points in the set within geodesic distance ϵ measured on the surface.

27.2 BASIC ALGORITHMS

Table 27.2.1 lists basic algorithms that compute the Delaunay triangulation of n point sites in \mathbb{R}^d using the Euclidean metric. Using the connection with con-

vexity, any $(d+1)$ -dimensional convex hull algorithm can be used to compute a d -dimensional Delaunay triangulation; in fact the divide-and-conquer, incremental, and gift-wrapping algorithms are specialized convex hull algorithms. Running times are given both for worst-case inputs, and for inputs chosen uniformly at random inside a sphere, with expectation taken over input distribution. The Voronoi diagram can be obtained in linear time from the Delaunay triangulation, using the one-one correspondence between their faces. See [Aur91, Ede87, For95, AK00, AKL13] for more references. Chapter 67 lists available implementations of Voronoi diagram algorithms.

TABLE 27.2.1 Delaunay Triangulation algorithms in the Euclidean metric for point sites.

ALGORITHM	DIM	WORST CASE	UNIFORM
Flipping	2	$O(n^2)$	
Plane sweep	2	$O(n \log n)$	
Divide-and-conquer	2	$O(n \log n)$	$O(n)$
Randomized incremental	2	$O(n \log n)$	
Randomized incremental	≥ 3	$O(n^{\lceil d/2 \rceil})$	$O(n \log n)$
Gift-wrapping	≥ 2	$O(n^{\lceil d/2 \rceil + 1})$	$O(n)$

THE RANDOMIZED INCREMENTAL ALGORITHM

The incremental algorithm adds sites one by one, updating the Delaunay triangulation after each addition. The update consists of discovering all Delaunay faces whose circumspheres contain the new site. These faces are deleted and the empty region is partitioned into new faces, each of which has the new site as a vertex. See Figure 27.2.1. An efficient algorithm requires a good data structure for finding the faces to be deleted. Then the running time is determined by the total number of face updates, which depends upon site insertion order. The bounds given in Table 27.2.1 are the expected running time of an algorithm that makes a random choice of insertion order, with each insertion permutation equally likely; the bounds for the worst-case insertion order are about a factor of n worse. For uniform data there is a double expectation, over both insertion order and input distribution. With additional algorithmic complexity, it is possible to obtain deterministic algorithms with the same worst-case running times [Cha93].

THE PLANE SWEEP ALGORITHM

The plane sweep algorithm computes a planar Delaunay triangulation using a horizontal line that sweeps upward across the plane. The algorithm discovers a Delaunay triangle when the sweepline passes through the topmost point of its circumcircle; in Figure 27.2.2, the Delaunay triangles shown have already been discovered. A sweepline data structure stores an ordered list of sites; the entry for site s corresponds to an interval I_s on the sweepline where each maximal empty circle with

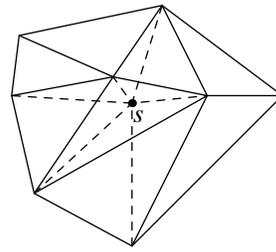


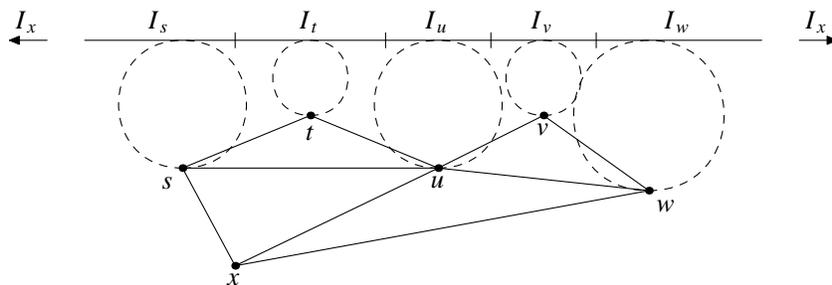
FIGURE 27.2.1

The addition of site s deletes four triangles and adds six (shown dashed).

topmost point in I_s touches site s . The sweepline moves in discrete steps only when the ordered list changes. This happens when a new site is encountered or when a new Delaunay triangle is discovered (at the topmost point of the circumcircle of three sites that are consecutive on the sweepline list). A priority queue is needed to determine the next sweepline move. The running time of the algorithm is $O(n \log n)$ since the sweepline moves $O(n)$ times—once per site and once per triangle—and it costs time $O(\log n)$ per move to maintain the priority queue and sweepline data structure.

FIGURE 27.2.2

The sweepline list is x, s, t, u, v, w, x . The next Delaunay triangle is tuv .



OTHER ALGORITHMS

The planar divide-and-conquer algorithm uses a splitting line to partition the point set into two equal halves, recursively computes the Delaunay triangulation of each half, and then merges the two subtriangulations in linear time. If the sites form the vertices of a convex polygon, then the Voronoi diagram can be computed in linear time [AGSS89]. In any dimension, the gift-wrapping algorithm is a specialization of the convex-hull gift-wrapping algorithm (Chapter 26) to Delaunay triangulations. There is an approximately output-sensitive algorithm, with running time $O(f \log n \log \Delta)$ where f is the number of output simplices, n is the number of points and Δ is the spread of the point set [MS14].

Graphics hardware, in particular Z-buffers, allow efficient practical computation of fixed-resolution approximate Voronoi diagrams for quite general sites and distance functions [HCK⁺99].

27.3 EXTENSIONS

GLOSSARY

Order- k Voronoi diagram: The order- k Voronoi diagram partitions \mathbb{R}^d on the basis of the first k closest sites (without distinguishing order among them).

Farthest site Voronoi diagram: The farthest site Voronoi diagram partitions \mathbb{R}^d on the basis of the farthest site, or equivalently, the closest $n-1$ of n sites.

Constrained Delaunay triangulation: Constrained Delaunay triangulations are defined relative to a set of **constraint facets** that restrict visibility. The constrained Delaunay triangulation of a set of sites has the property that for every simplex, the interior of the simplex circumsphere contains no site visible from the interior of the simplex.

Conforming Delaunay triangulation: Fix a set of noncrossing **constraint facets** E and a set of point sites S . A conforming Delaunay triangulation is the Delaunay triangulation of a set of sites $S' \supseteq S$ so that every facet in E is the union of Delaunay faces of S' .

Power or Laguerre diagram: A Voronoi diagram for sites s_i with weights w_i where the distance from a point x is measured along a tangent to the sphere of radius $\sqrt{w_i}$ centered on s_i .

HIGHER-ORDER VORONOI DIAGRAMS

The order- k Voronoi diagram can be obtained as an appropriate projection of the k -level of an arrangement of hyperplanes (see [Ede87, For93] and Section 28.2 of this Handbook); it can also be obtained as the orthogonal projection of an intersection polytope [AS92]. In dimension 2, the order- k Voronoi diagram has $O(k(n-k))$ faces. In dimensions $d \geq 3$, the sum of the number of faces of the order- j diagrams, $j \leq k$, is $O(n^{\lceil d/2 \rceil} k^{\lfloor d/2 \rfloor + 1})$ [CS89]; finding good bounds for fixed k remains an open problem. See Table 27.3.1 for running time bounds.

TABLE 27.3.1 Algorithms for order- k Voronoi diagrams of point sites in the Euclidean metric.

PROBLEM	DIM	TIME
Farthest site	2	$O(n \log n)$
Farthest site	≥ 3	$O(n^{\lceil d/2 \rceil})$
Order- k	2	$O(k(n-k) \log n + n \log^3 n)$
Order- j , $1 \leq j \leq k$	≥ 3	$O(n^{\lceil d/2 \rceil} k^{\lfloor d/2 \rfloor + 1})$

CONSTRAINED DELAUNAY TRIANGULATIONS

Let S be a set of n point sites in \mathbb{R}^2 and E a set of noncrossing *constraint* edges with endpoints in S . A point $p \in \mathbb{R}^2$ is *visible* from a site s if the open segment ps does not intersect any edge of E . The *constrained Delaunay triangulation* (of S with respect to E) is a triangulation of S extending the edges in E so that the circumcircle of every triangle contains no site that is visible from the interior of the triangle. In \mathbb{R}^2 the constrained Delaunay triangulation always exists; it is as close as possible to the true Delaunay triangulation, subject to the constraint that the edges in E must be used. See also Section 28.2.

The *bounded distance* from a site to a point is Euclidean distance if the point is visible, and infinite otherwise; the *bounded Voronoi diagram* of S using E is defined using bounded distance. The bounded Voronoi diagram is dual to a subgraph of the constrained Delaunay triangulation.

Both the constrained Delaunay triangulation and the bounded Voronoi diagram can be computed in time $O(n \log n)$ using either divide-and-conquer or the sweepline paradigm. If the sites and constraint edges are the vertices and edges of a simple polygon, respectively, then the constrained Delaunay triangulation can be computed in linear time [KL96].

The constrained Delaunay triangulation can be generalized to dimension $d > 2$. Let S be a set of point sites in \mathbb{R}^d and E a set of $d-1$ -dimensional closed simplicial *constraint facets* with vertices in S that are noncrossing (the intersection of two constraint simplices is either empty or a face of both). A *constrained Delaunay triangulation* (of S with respect to E) is a triangulation such that constraint facets are triangulation facets and such that, for every d -simplex, the interior of the circumsphere of the simplex contains no site visible from the interior of the simplex. In dimension $d > 2$, a constrained Delaunay triangulation does not always exist; for example, the Schönhardt polyhedron in dimension 3 cannot be triangulated without extra Steiner points (see Section 29.5). Shewchuk [She08] gives a sufficient condition for the existence of constrained Delaunay triangulations: the constraint simplices must be *ridge-protected*, that is, each j -face of a constraint simplex, $j \leq d-2$, must have a closed circumsphere not containing any sites. He also gives two algorithms that construct the constrained Delaunay triangulation when it exists, one a sweep algorithm [She00] and the second a flipping algorithm [She03].

CONFORMING DELAUNAY TRIANGULATIONS

Let S be a set of point sites in \mathbb{R}^d and E a set of noncrossing j -dimensional *constraint simplices*, $j < d$. A *conforming Delaunay triangulation* of E is the Delaunay triangulation of a set of sites $S' \supseteq S$ so that every simplex in E is the union of faces of Delaunay simplices of S' . In \mathbb{R}^2 , Edelsbrunner and Tan [ET93] give an algorithm for conforming Delaunay triangulations, where the cardinality of S' is $O(n^3)$, n the cardinality of S . See [CP06] for an approach in \mathbb{R}^3 .

OTHER DISTANCE MEASURES

Table 27.3.2 lists Voronoi diagram algorithms where “distance” is altered. The distance from a site s_i to a point x can be a function of the Euclidean distance $e(s_i, x)$ and a site-specific real weight w_i .

TABLE 27.3.2 Algorithms for point sites in \mathbb{R}^2 , other distance measures.

PROBLEM	DISTANCE TO x	TIME
Additive weights	$w_i + e(s_i, x)$	$O(n \log n)$
Multiplicative weights	$w_i e(s_i, x)$	$O(n^2)$
Laguerre or power	$\sqrt{e(s_i, x)^2 - w_i}$	$O(n \log n)$
L_p	$\ s_i - x\ _p$	$O(n \log n)$
Skew	$e(s_i, x) + \kappa \Delta_y(s_i, x)$	$O(n \log n)$
Convex distance function		$O(n \log n)$
Abstract	axiomatic	$O(n \log n)$
Simple polygon	geodesic	$O(n \log^2 n)$
Crystal growth	$w_i \cdot SP(s_i, x)$	$O(n^3 + nS \log S)$
Anisotropic	local metric tensor	$O(n^{2+\epsilon})$

The seemingly peculiar **power distance** [Aur87] is the distance from x to the sphere of radius $\sqrt{w_i}$ about s_i along a line tangent to the sphere. Many of the basic Voronoi diagram algorithms extend immediately to the power distance, even in higher dimension.

A (**polygonal convex distance function**) [CD85] is defined by a convex polygon C with the origin in its interior. The distance from x to y is the real $r \geq 0$ so that the boundary of $rC + x$ contains y . Polygonal convex distance functions generalize the L_1 and L_∞ metrics (C is a diamond or square, respectively); a polygonal convex distance function is a metric exactly if C is symmetric about the origin.

The (**skew distance**) [AAC⁺99] between two points is the Euclidean distance plus a constant times the difference in y -coordinate. It can be viewed as a measure of the difficulty of motion on a plane that has been rotated in three dimensions about the x -axis.

An **abstract** Voronoi diagram [KMM93] is defined by the “bisectors” between pairs of sites, which must satisfy special properties.

The **geodesic** distance inside an environment of polygonal obstacles is the length of the shortest path that avoids obstacle interiors. Some progress using the geodesic metric appears in [HS99].

The **crystal growth** Voronoi diagram [SD91] models crystal growth where each crystal has a different growth rate. The distance from a site s_i to a point x in the Voronoi face of s_i is $w_i \cdot SP(s_i, x)$, where w_i is a weight and $SP(s_i, x)$ is the shortest path distance lying entirely within the Voronoi face of s_i . The parameter S in the running time measures the time to approximate bisectors numerically.

An **anisotropic** Voronoi diagram [LS03] requires a metric tensor at each site to specify how distance is measured from that site. The anisotropic Voronoi diagram generalizes the multiplicatively weighted diagram; both have the property that the region of a site may be disconnected or not simply-connected.

The **Bregman divergence** from a point p to a point q is defined relative to a convex function F and roughly measures how $F(q)$ differs from the first-order Taylor approximation obtained from F at p . The Bregman divergence generalizes various functions from statistics and machine learning, e.g., the Kullback-Leibler and Itakura-Saito divergences. Boissonnat *et al.* [BNN10] define various versions and give algorithms for the Voronoi diagram and Delaunay triangulation defined using Bregman divergence.

OTHER SITES

Many classes of sites besides points have been used to define Voronoi diagram and Voronoi-diagram-like objects. For example, the Voronoi diagram of a set of disjoint circles in the plane is just an additively-weighted point-site Voronoi diagram.

The Voronoi diagram of a set of n line segment sites in \mathbb{R}^2 can be computed in time $O(n \log n)$ using the sweepline method or the divide-and-conquer method. The well-known medial axis of a polygon or polygonal region can be obtained from the Voronoi diagram of its constituent line segments. The medial axis of a simple polygon can be found in linear time, using the linear-time triangulation algorithm [AGSS89].

Aichholzer *et al.* [AAA⁺10] describe a randomized algorithm that handles segments and circular arcs; more general free-form curves can be handled by approximation using polygonal and curved arcs. The algorithm uses an unusual divide-and-conquer paradigm and runs in expected time $O(n \log n)$ under some assumptions.

The *straight skeleton* of a simple polygon [AAAG95] is structurally similar to the medial axis, though it is not strictly a Voronoi diagram. It is defined as the trace of the vertices of the polygon, as the polygon is shrunk by translating each edge inward at a constant rate. Unlike the medial axis, it has only polygonal edges. Several algorithms achieve time and space bounds of roughly $O(nr)$, r the number of reflex vertices; a subquadratic worst-case bound is known [EE99]. Some progress on extension to three-dimensional polyhedra is available [BEGV08].

The worst-case combinatorial and algorithmic complexity of Voronoi diagrams of general sites in three dimensions is not well understood [KS03, KS04]. For many sites and metrics in \mathbb{R}^3 , roughly cubic upper bounds on the combinatorial complexity of the Voronoi diagram can be obtained using the general theory of lower envelopes of trivariate functions (see Chapter 28 on arrangements). Known lower bounds are roughly quadratic, and upper bounds are conjectured to be quadratic.

A specific long-standing open problem is to give tight bounds on the combinatorial complexity of the Voronoi diagram of a set of n lines in \mathbb{R}^3 using the Euclidean metric. Roughly quadratic upper bounds are known if the lines have only a constant number of orientations [KS03]. The boundary of the union of infinite cylinders of fixed radius is also known to have roughly quadratic complexity [AS00]; this boundary can be viewed as the level set of the line Voronoi diagram at fixed distance. Everett *et al.* [EGL⁺09, ELLD09] give a complete combinatorial analysis of all possible Voronoi diagrams of three lines; Hemmer *et al.* [HSH10] describe a nearly-cubic exact-arithmetic algorithm that computes the Voronoi diagram of an arbitrary set of lines in \mathbb{R}^3 .

Dwyer [Dwy97] shows that the expected complexity of the Euclidean Voronoi diagram of n k -flats in \mathbb{R}^d is $\theta(n^{d/(d-k)})$, as long as $d \geq 3$ and $0 \leq k < d$. The flats are assumed to be drawn independently from the uniform distribution on k -flats intersecting the unit ball. Thus the expected complexity of the Voronoi diagram of a set of n uniformly random lines in \mathbb{R}^3 is $O(n^{3/2})$.

Voronoi diagrams in \mathbb{R}^3 can be defined by convex distance functions, as in the plane. If the distance function is determined by a convex polytope with a constant number of facets, then the Voronoi diagram of a set of disjoint polyhedra has combinatorial complexity roughly quadratic in the total number of vertices of all polytopes [KS04].

KINETIC VORONOI DIAGRAMS

Consider a set of n moving point sites in \mathbb{R}^d , where the position of each site is a continuous function of a real parameter t , representing time. In general the Voronoi diagram of the points will vary continuously with t , without any change to its combinatorial structure; however at certain discrete values t_i , $i = 1, \dots$, the combinatorial structure will change. A *kinetic Voronoi diagram algorithm* determines times that the structure changes and at each change updates a data structure representation of the Voronoi diagram. See a survey in [DC08] and also Chapter 53.

Rubin [Rub15] recently showed a nearly-quadratic upper bound on the number of combinatorial changes to a planar Delaunay triangulation, provided that the points are each moving at unit speed along a straight line.

OTHER SURFACES

The Delaunay triangulation of a set of points on the surface of a sphere S^d has the same combinatorial structure as the convex hull of the set of points, viewed as sitting in \mathbb{R}^{d+1} . On a closed Riemannian manifold, the Delaunay triangulation of a set of sites exists and has properties similar to the Euclidean case, as long as the set of sites is sufficiently dense [LL00, GM01].

MOTION PLANNING

The motion planning problem is to find a collision-free path for a robot in an environment filled with obstacles. The Voronoi diagram of the obstacles is quite useful, since it gives a lower-dimensional skeleton of maximal clearance from the obstacles. In many cases the shape of the robot can be used to define an appropriate metric for the Voronoi diagram. See Section 50.2 for more on the use of Voronoi diagrams in motion planning.

SURFACE RECONSTRUCTION

The *surface reconstruction problem* is to construct an approximation to a two-dimensional surface embedded in \mathbb{R}^3 , given a set of points sampled from the surface. A whole class of surface reconstruction methods are based on the computation of Voronoi diagrams [AB99, Dey07] (see Chapters 35 on surface reconstruction).

DELAUNAY REFINEMENT

Delaunay refinement builds a mesh of “fat” triangles, i.e., with no small or large angles, by maintaining a Delaunay triangulation while judiciously adding new sites. The original ideas come from Chew [Che89], who showed that skinny triangles can be eliminated by repeatedly adding as new site the circumcenter of any skinny triangle, and Ruppert [Rup95], who with an additional update rule gave an algorithm that produces meshes that have the minimum possible number of triangles, up to

a constant factor. See [CDS13] for the now-extensive theory including extensions to higher dimensions.

IMPLEMENTATIONS

There are a number of available high-quality implementations of algorithms that compute Delaunay triangulations and Voronoi diagrams of point sites in the Euclidean metric. These can be obtained from the web and from the algorithms libraries CGAL and LEDA (see Chapters 67 and 68 on implementations). It is typically challenging to implement algorithms for sites other than points or metrics other than the Euclidean metric, largely because of issues of numerical robustness. See [Bur96, Hel01, SIII00] for approaches for line segment sites in the plane.

27.4 IMPORTANT PROPERTIES

ROUNDNESS

The Delaunay triangulation is “round,” that is, skinny simplices are avoided. This can be formalized in two dimensions by Lawson’s classic result: over all possible triangulations, the Delaunay triangulation maximizes the minimum angle of any triangle. No generalization using angles is known in higher dimension. However, define the *enclosing radius* of a simplex as the minimum radius of an enclosing sphere. In any dimension and over all possible triangulations of a point set, the Delaunay triangulation minimizes the maximum enclosing radius of any simplex [Raj94]. Also see Section 29.4 on mesh generation.

OPTIMALITY

Fix a set S of sites in \mathbb{R}^d . For a triangulation T of S with simplices t_1, \dots, t_n , define

$$\begin{aligned}v_i &= \text{sum of squared vertex norms of } t_i \\c_i &= \text{squared norm of barycenter of } t_i \\a_i &= \text{volume of } t_i \\s_i &= \text{sum of squared edge lengths of } t_i \\r_i &= \text{circumradius of } t_i.\end{aligned}$$

Over all triangulations T of S , the Delaunay triangulation attains the unique min-

imum of the following functions, where κ is any positive real [Mus97]:

$$\begin{aligned} V(T) &= \sum_i v_i a_i \\ C(T) &= \sum_i c_i a_i \\ H(T) &= \sum_i s_i / a_i && d = 2 \text{ only} \\ R(T, \kappa) &= \sum_i r_i^\kappa && d = 2 \text{ only.} \end{aligned}$$

VISIBILITY DEPTH ORDERING

Choose a viewpoint v and a family of disjoint convex objects in \mathbb{R}^d . Object A is *in front of* object B from v if there is a ray starting at v that intersects A and then B in that order. Though an arbitrary family can have cycles in the “in front of” relation, the relation is acyclic for the faces of the Delaunay triangulation, for any viewpoint and any dimension [Ede90].

An application comes from computer graphics. The *painter’s algorithm* renders 3D objects in back to front order, with later objects simply overpainting the image space occupied by earlier objects. A valid rendering order always exists if the “in front of” relation is acyclic, as is the case if the objects are Delaunay tetrahedra, or a subset of a set of Delaunay tetrahedra.

SUBGRAPH RELATIONSHIPS

The edges of a Delaunay triangulation form a graph DT whose vertices are the sites. In any dimension, the following subgraph relations hold:

$$NNG \subseteq EMST \subseteq RNG \subseteq GG \subseteq DT$$

where $EMST$ is the Euclidean minimum spanning tree, RNG is the relative neighborhood graph, GG is the Gabriel graph, and NNG is the nearest neighbor graph with edges viewed as undirected; the containment $NNG \subseteq EMST$ requires that the sites be in general position. See Section 32.1 on proximity graphs.

DILATION

A geometrically embedded graph G has *dilation* c if for any two vertices, the shortest path distance along the edges of G is at most c times the Euclidean distance between the vertices. In \mathbb{R}^2 , the Delaunay triangulation has dilation at most ~ 1.998 [Xia13] and examples exist with dilation at least ~ 1.5932 . With an equilateral-triangle convex distance function, the dilation is at most 2.

INTERPOLATION

Suppose each point site $s_i \in S \subset \mathbb{R}^d$ has an associated function value f_i . For $p \in \mathbb{R}^d$ define $\lambda_i(p)$ as the proportion of the area of s_i 's Voronoi cell that would be removed if p were added as a site. Then the *natural neighbor* interpolant $f(p) = \sum \lambda_i(p) f_i$ is C^0 , and C^1 except at sites. This construction can be generalized to give a C^k interpolant for any fixed k [HS02].

Alternatively, for a triangulation of S in \mathbb{R}^2 , consider the piecewise linear surface defined by linear interpolation over each triangle. Over all possible triangulations, the Delaunay triangulation minimizes the roughness of the resulting surface, where *roughness* is the square of the L_2 norm of the gradient of the surface, integrated over the triangulation [Rip90].

27.5 SOURCES AND RELATED MATERIAL

FURTHER READING

[Aur91, For95, AK00]: Survey papers that cover many aspects of Delaunay triangulations and Voronoi diagrams.

[AKL13, OBSC00]: General reference books on Voronoi diagrams and Delaunay triangulations; the latter has an extensive discussion of applications.

[Ede87, PS85, BKOS97]: Basic references for geometric algorithms.

[CDS13]: Extensive discussion of Delaunay mesh generation.

www.voronoi.com, www.ics.uci.edu/~eppstein/gina/voronoi.html: Web sites devoted to Voronoi diagrams.

RELATED CHAPTERS

- Chapter 26: Convex hull computations
- Chapter 28: Arrangements
- Chapter 29: Triangulations and mesh generation
- Chapter 32: Proximity algorithms
- Chapter 36: Computational convexity
- Chapter 50: Algorithmic motion planning

REFERENCES

- [AS00] P.K. Agarwal and M. Sharir. Pipes, cigars, and kreplach: the union of Minkowski sums in three dimensions. *Discrete Comput. Geom.*, 24:645–685, 2000.
- [AGSS89] A. Aggarwal, L.J. Guibas, J.B. Saxe, and P.W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete Comput. Geom.*, 4:591–604, 1989.

- [AAA⁺10] O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, E. Pilgerstorfer, and M. Rabl. Divide-and-conquer for Voronoi diagrams revisited. *Comput. Geom.*, 43:688–699, 2010.
- [AAAG95] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner. A novel type of skeleton for polygons. *J. Univer. Comput. Sci.*, 1:752–761, 1995.
- [AAC⁺99] O. Aichholzer, F. Aurenhammer, D.Z. Chen, D.T. Lee, and E. Papadopoulou. Skew Voronoi diagrams. *Internat. J. Comput. Geom. Appl.*, 9:235–247, 1999.
- [AB99] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.*, 22:481–504, 1999.
- [ABL03] D. Attali, J.-D. Boissonnat, and A. Lieutier. Complexity of the Delaunay triangulation of points on surfaces: The smooth case. *Proc. 19th. Sympos. Comput. Geom.*, pages 201–210, ACM Press, 2003.
- [AS92] F. Aurenhammer and O. Schwarzkopf. A simple randomized incremental algorithm for computing higher order Voronoi diagrams. *Internat. J. Comput. Geom. Appl.*, 2:363–381, 1992.
- [Aur87] F. Aurenhammer. Power diagrams: properties, algorithms, and applications. *SIAM J. Comput.*, 16:78–96, 1987.
- [Aur91] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, 1991.
- [AK00] F. Aurenhammer and R. Klein. Voronoi diagrams. In J. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 201–290, Elsevier, Amsterdam, 2000.
- [AKL13] F. Aurenhammer, R. Klein, and D.T. Lee. *Voronoi Diagrams and Delaunay Triangulations* World Scientific, Singapore, 2013.
- [BEGV08] G. Barequet, D. Eppstein, M.T. Goodrich, and A. Vaxman. Straight skeletons of three-dimensional polyhedra. *Proc. 16th Europ. Sympos. Algorithms.*, vol. 5193 of *LNCS*, pages 148–160, Springer, Berlin, 2008.
- [BKOS97] M. de Berg, M. van Kreveld, M.H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, 1997; 3rd edition, 2008.
- [BNN10] J.-D. Boissonnat, F. Nielsen, and R. Nock. Bregman Voronoi diagrams. *Discrete Comput. Geom.*, 44:281–307, 2010.
- [Bur96] C. Burnikel. *Exact computation of Voronoi diagrams and line segment intersections*. Ph.D. Thesis, Universität des Saarlandes, 1996.
- [CD85] L.P. Chew and R.L. Drysdale. Voronoi diagrams based on convex distance functions. In *Proc. 1st Sympos. Comput. Geom.*, pages 234–244, ACM Press, 1985.
- [CDS13] S.-W. Cheng, T.K. Dey, and J.R. Shewchuk. *Delaunay Mesh Generation*, CRC Press, Boca Raton, 2013.
- [Cha93] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.
- [Che89] L.P. Chew. Guaranteed quality triangular meshes. Technical Report TR-89-983, Dept. Comp. Sci., Cornell University, 1989.
- [CP06] S.-W. Cheng and S.-H. Poon. Three-dimensional Delaunay mesh generation. *Discrete Comput. Geom.* 36:419–456, 2006.
- [CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- [DC08] O. Devillers and P.M.M. de Castro. State of the Art: Updating Delaunay triangulations for moving points. Research Report RR-6665, INRIA, 2008.

- [Dey07] T.K. Dey *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*. Cambridge Monog. Appl. Comput. Math., Cambridge University Press, 2006.
- [Dwy91] R.A. Dwyer. Higher-dimensional Voronoi diagrams in linear expected time. *Discrete Comput. Geom.*, 6:343–367, 1991.
- [Dwy97] R. Dwyer. Voronoi diagrams of random lines and flats. *Discrete Comput. Geom.*, 17:123–136, 1997.
- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer, Berlin, 1987.
- [Ede90] H. Edelsbrunner. An acyclicity theorem for cell complexes in d dimensions. *Combinatorica*, 10:251–260, 1990.
- [EE99] D. Eppstein and J. Erickson. Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions. *Discrete Comput. Geom.*, 22:569–592, 1999.
- [EGL⁺09] H. Everett, C. Gillot, D. Lazard, S. Lazard, and M. Pouget. The Voronoi diagram of three arbitrary lines in \mathbb{R}^3 . *Abstracts of 25th Europ. Workshop. Comput. Geom.*, 2009.
- [ELLD09] H. Everett, D. Lazard, S. Lazard, M.S. El Din. The Voronoi diagram of three lines. *Discrete Comput. Geom.*, 42:94–130, 2009.
- [Eri05] J. Erickson. Dense point sets have sparse Delaunay triangulations. *Discrete Comput. Geom.* 33:83–15, 2005.
- [ET93] H. Edelsbrunner and T.-S. Tan. An upper bound for conforming Delaunay triangulations. *Discrete Comput. Geom.*, 10:197–213, 1993.
- [For93] S.J. Fortune. Progress in computational geometry. In R. Martin, editor, *Directions in Geometric Computing*, pages 81–128, Information Geometers, Winchester, 1993.
- [For95] S.J. Fortune. Voronoi diagrams and Delaunay triangulations. In F. Hwang and D.Z. Du, editors, *Computing in Euclidean Geometry*, 2nd edition, pages 225–265, World Scientific, Singapore, 1995.
- [GM01] C.I. Grima and A. Márquez. *Computational Geometry on Surfaces*. Kluwer, Dordrecht, 2001.
- [HCK⁺99] K.E. Hoff III, T. Culver, J. Keyser, M.C. Lin, and D. Manocha. Fast computation of generalized Voronoi diagrams using graphics hardware. *Proc. ACM Conf. SIGGRAPH*, pages 277–286, 1999.
- [Hel01] M. Held. VRONI: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. *Comput. Geom.*, 18:95–123, 2001.
- [HS99] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 26:2215–2256, 1999.
- [HS02] H. Hiyoshi and K. Sugihara. Improving continuity of Voronoi-based interpolation over Delaunay spheres. *Comput. Geom.*, 22:167–183, 2002.
- [HSH10] M. Hemmer, O. Setter, and D. Halperin. Constructing the exact Voronoi diagram of arbitrary lines in space. In *Proc. 18th Europ. Sympos. Algorithms*, vol. 6346 of *LNCIS*, pages 398–409, Springer, Berlin, 2010.
- [KL96] R. Klein and A. Lingas. A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon. *Internat. J. Comput. Geom. Appl.*, 6:263–278, 1996.
- [KMM93] R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Comput. Geom.*, 3:157–184, 1993.
- [KS03] V. Koltun and M. Sharir. Three dimensional Euclidean Voronoi diagrams of lines with a fixed number of orientations. *SIAM J. Comput.*, 32:616–642, 2003.

- [KS04] V. Koltun and M. Sharir. Polyhedral Voronoi diagrams of polyhedra in three dimensions. *Discrete Comput. Geom.*, 31:83–124, 2004.
- [LL00] G. Leibon and D. Letscher. Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. *Proc. 16th Sympos. Comput. Geom.*, pages 341–349, ACM Press, 2000.
- [LS03] F. Labelle and J.R. Shewchuk. Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation. *Proc. 19th Sympos. Comput. Geom.*, pages 191–200, ACM Press, 2003.
- [MS14] G.L. Miller and D.R. Sheehy. A new approach to output-sensitive Voronoi diagrams and Delaunay triangulations. *Discrete Comput. Geom.*, 52:476–491, 2014.
- [Mus97] O.R. Musin. Properties of the Delaunay triangulation. *Proc. 13th Sympos. Comput. Geom.*, pages 424–426, ACM Press, 1997.
- [OBSC00] A. Okabe, B. Boots, K. Sugihara, and S.N. Chio. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, second edition. Wiley, Chichester, 2000.
- [PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry*. Springer-Verlag, New York, 1985.
- [Raj94] V.T. Rajan. Optimality of the Delaunay triangulation in \mathbb{R}^d . *Discrete Comput. Geom.*, 12:189–202, 1994.
- [Rip90] S. Rippa. Minimal roughness property of the Delaunay triangulation. *Comput. Aided Design*, 7:489–497, 1990.
- [Rub15] N. Rubin. On kinetic Delaunay triangulations: a near-quadratic bound for unit speed motions. *J. ACM*, 62:25, 2015.
- [Rup95] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, 18:548–585, 1995.
- [SD91] B. Schaudt and R.L. Drysdale. Multiplicatively weighted crystal growth Voronoi diagram. In *Proc. 7th Sympos. Comput. Geom.*, pages 214–223, ACM Press, 1991.
- [She00] J.R. Shewchuk. Sweep algorithms for constructing higher-dimensional constrained Delaunay triangulations. In *Proc. 16th Sympos. Comput. Geom.*, pages 350–359, ACM Press, 2000.
- [She03] J.R. Shewchuk. Updating and constructing constrained Delaunay and constrained regular triangulations by flips. In *Proc. 19th Sympos. Comput. Geom.*, pages 181–190, ACM Press, 2003.
- [She08] J.R. Shewchuk. General-dimensional constrained Delaunay and constrained regular triangulations, I: Combinatorial Properties. *Discrete Comput. Geom.*, 39:580–637, 2008.
- [SII00] K. Sugihara, M. Iri, H. Inagaki, and T. Imai. Topology-oriented implementation—An approach to robust geometric algorithms. *Algorithmica*, 27:5–20, 2000.
- [Xia13] G. Xia. The stretch factor of the Delaunay triangulation is less than 1.998. *SIAM J. Comput.*, 42:1620–1659, 2013.