Divide-and-Conquer for Voronoi Diagrams Revisited*

Oswin Aichholzer† Wolfgang Aigner* Franz Aurenhammer‡ Thomas Hackl* Bert Juettler §
Elisabeth Pilgerstorfer‡ Margot Rabl‡

Abstract

We propose a simple and practical divide-and-conquer algorithm for constructing planar Voronoi diagrams. The novel aspect of the algorithm is its emphasis on the top-down phase, which makes it applicable to sites of general shape.

1 Introduction

The divide-and-conquer paradigm gave the first optimal solution for constructing the closest-site Voronoi diagram in the plane [13]. Though being a classical example for applying a powerful algorithmic method in computational geometry, the resulting algorithm became no favorite for implementation, not even in the case of point sites.

Literature tells us that divide-and-conquer is involved if emphasis is on the bottom-up phase, even if the sites are of relatively simple shape; see [10, 15, 5, 11]. The crux is the missing separability condition for the sites, which would prevent the merge curve from breaking into several components. Many alternative strategies for computing generalized Voronoi diagrams have been tried, including incremental insertion [3] and the plane-sweep technique [7].

In all these algorithms the bisector curves take part in the computation. Bisectors are usually composed of several curve pieces, and may even be two-dimensional if not defined carefully in the case of shared endpoints (which arise naturally when decomposing complex sites into simpler ones). Consequently, the algorithms are involved and also suffer from numerical imprecison. Difficulties may be partially eluded when working in the dual environment: Instead of intersecting two bisectors, the center of a circle tangent to the three defining sites is calculated. This bears the advantage of working on the sites directly. For general sites, however, tangent circles may not be unique, and are usually difficult to calculate.

The algorithm we propose works directly on the sites, too, but its atomic operation is much simpler, namely, an inclusion test of a site in a *fixed* circle. We first extract the combinatorial structure of the Voronoi diagram, and fill in the bisector curves later on. In contrast to existing Voronoi/Delaunay algorithms, no constructed object is ever discarded. Our setting is very general, including polygonal sites, circular disks, and even spline curves. Boundaries of curved planar objects with holes can be modeled.

Applications are manifold. One is motion planning in piecewise-circular environments [16] which, compared to piecewise-linear environments, leads to shorter and 'smoother' robot paths. Another is shape offsetting where, compared to similar approaches [9, 8, 4], our method is simpler because we compute only a combinatorial representation of the diagram for this application.

2 Dividing the Voronoi diagram

Our sites are pairwise disjoint topological disks of dimension two, one, or zero in the Euclidean plane $\mathbb{R}^2$. That is, a site is either homeomorphic to a disk or to a line segment, or is simply a point. This includes polygons, circular disks, and open spline curves as sites. Here and throughout this note, let S denote the given set of sites. The distance of a point x to a site $s \in S$ is $d(x, s) = \min_{y \in s} \delta(x, y)$, where δ denotes the Euclidean distance function. As done e.g. in [3, 15], we define the Voronoi diagram, $V(S)$, of S via its *edge graph*, $\mathcal{G}_S$, which is the set of all points having more than one closest point on the union of all sites. An edge of $\mathcal{G}_S$ containing points equidistant from two or more different points on the same site s is called a *self-edge* for s. The *regions* of $V(S)$ are the maximal connected subsets of the complement of $\mathcal{G}_S$ in $\mathbb{R}^2$. They are topologically open sets.

Observation 1 *The regions of $V(S)$ bijectively correspond to the sites in S. Each site is contained in its region, and regions are simply connected.*

We thus can talk of the region *of a site*, s, which we will denote with $R(s)$ in the sequel. The differences to a bisector-based definition of the Voronoi diagram should be noticed. Self-edges are ignored in such a definition unless the sites are split into suitable pieces. Such pieces, however, share boundaries—a fact that,

*Supported by FWF NRN 'Industrial Geometry' S9205-N12

†Institute for Software Technology, Graz University of Technology, Austria, {oaich,waigner,thackl}@ist.tugraz.at

‡Institute for Theoretical Computer Science, Graz University of Technology, Austria, auren@igi.tugraz.at

§Institute of Applied Geometry, Johannes Kepler University Linz, Austria, {Bert.Juettler,Margot.Rabl}@jku.at

if not treated with care, may give rise to unpleasant phenomena like two-dimensional bisectors.

To get rid of the unbounded components of the diagram, we include a surrounding circle, Γ, (or any other desired curve) into the set S of sites. We can always choose Γ in a way such that each vertex of $V(S \setminus \{\Gamma\})$ is also a vertex of $V(S)$.

For later purposes, we seek a (minimal) set of points whose removal from the edge graph $\mathcal{G}_S$ breaks all its cycles. Finding such a set is nontrivial, in view of the possible presence of self-edges. For a site $s \neq \Gamma$, let $p(s)$ be a point on s with smallest ordinate, and denote with $q(s)$ the closest point on $\mathcal{G}_S$ vertically below $p(s)$. By the boundedness of $R(s)$, the point $q(s)$ always exists, and we define a new geometric graph as

$$\mathcal{T}_S = \mathcal{G}_S \setminus \{q(s) \mid s \in S \setminus \{\Gamma\}\}. \tag{1}$$

Lemma 1 *The graph $\mathcal{T}_S$ is a tree.*

Proof. Omitted in this version. □

3 Augmented domains

Our next aim is to interpret the tree $\mathcal{T}_S$ in Lemma 1 as the medial axis of a generalized planar domain. In this way, we will be able to construct the Voronoi diagram $V(S)$ by means of a medial axis algorithm, as if a *simply connected* domain was the input. Usually, the similarity between these two structures is exploited the other way round: Medial axes are constructed as special cases of Voronoi diagrams.

Define $\mathcal{B} = B_0 \setminus \{s \in S \mid s \neq \Gamma\}$, where B_0 denotes the disk bounded by Γ. Then the medial axis, $\mathrm{MA}(\mathcal{B})$, of $\mathcal{B}$ is just the closure of the edge graph $\mathcal{G}_S$ of $V(S)$. We want to combinatorially disconnect the shape $\mathcal{B}$ at appropriate positions, such that the medial axis of the resulting domain corresponds to the tree decomposition $\mathcal{T}_S$ of $V(S)$.

As observed in [6], a maximal inscribed disk can be used to split the medial axis of a simply connected shape into two components which share a point at the disk's center. In order to extend this result to general shapes, we introduce the notion of an *augmented domain*, which is a set $\mathcal{A}$ together with a projection $\pi_{\mathcal{A}} : \mathcal{A} \to \mathbb{R}^2$. Initially, $\mathcal{A}$ is the original shape $\mathcal{B}$, and the associated projection $\pi_{\mathcal{B}}$ is the identity. Now, consider a maximal inscribed disk D of an augmented domain $\mathcal{A}$, which touches the boundary $\partial\mathcal{A}$ in exactly two points u and v. Denote with $\overset{\frown}{uv}$ and $\overset{\frown}{vu}$ the two circular arcs which the boundary of D is split into. The new augmented shape, $\mathcal{A}'$, which is obtained from $\mathcal{A}$ by splitting it with D, is defined as

$$\mathcal{A}' = \mathcal{A}^0 \cup D^1 \cup D^2$$

where $\mathcal{A}^0 = \{(x,0) \mid x \in A \setminus D\}$, $D^1 = \{(x,1) \mid x \in D\}$, and $D^2 = \{(x,2) \mid x \in D\}$. The associated projection is

$$\pi_{\mathcal{A}'} : \mathcal{A}' \to \mathbb{R}^2,\ (x,i) \mapsto \pi_A(x).$$

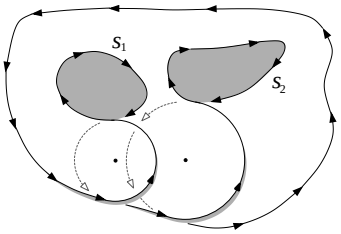

Figure 1: Boundary of an augmented domain.

We say that the line segment in $\mathcal{A}$ between points (x,i) and (y,j) is *contained* in $\mathcal{A}'$ if one of the following conditions is satisfied:

1. $i = j$ and the line segment $\overline{xy}$ avoids ∂D,
2. $\{i,j\} = \{0,1\}$ and $\overline{xy}$ intersects the arc $\overset{\frown}{uv}$, or
3. $\{i,j\} = \{0,2\}$ and $\overline{xy}$ intersects the arc $\overset{\frown}{vu}$.

For any two points (x,i) and (y,j) in $\mathcal{A}'$, their distance now can be defined. It equals the distance of $\pi_{\mathcal{A}}(x)$ and $\pi_{\mathcal{A}}(y)$ in $\mathbb{R}^2$, provided the connecting line segment is contained in $\mathcal{A}'$, and is ∞, otherwise. An (open) disk in $\mathcal{A}'$ with center (m,i) and radius ϱ is the set of all points in $\mathcal{A}'$ whose distance to (m,i) is less than ϱ. Such a disk is said to be *inscribed* in $\mathcal{A}'$ if its projection into $\mathbb{R}^2$ is again an open disk.

Having specified inscribed disks for $\mathcal{A}'$, the boundary of $\mathcal{A}'$ and the medial axis (transform) of $\mathcal{A}'$ can be defined as in the case of planar shapes. In particular, $\partial\mathcal{A}'$ derives from $\partial\mathcal{A}$ by disconnecting the latter boundary at the contact points u and v of the splitting disk D, and reconnecting it with the circular arcs $\overset{\frown}{uv}$ and $\overset{\frown}{vu}$. This process is depicted in Figure 1.

Concerning the medial axis, every maximal inscribed disk in $\mathcal{A}$ different from D corresponds to exactly one maximal inscribed disk in $\mathcal{A}'$, hence there is a bijection between $\mathrm{MAT}(\mathcal{A}) \setminus \{D\}$ and $\mathrm{MAT}(\mathcal{A}') \setminus \{D^1, D^2\}$. The medial axis of $\mathcal{A}'$ therefore is the same geometric graph as $\mathrm{MA}(\mathcal{A})$, except that the edge of $\mathrm{MA}(\mathcal{A})$ containing the center of D is split into two disconnected edges in $\mathrm{MA}(\mathcal{A}')$ which both have the center of D as one of their endpoints.

To draw the connection to the edge graph $\mathcal{G}_S$ of $V(S)$, the initial shape $\mathcal{B}$ is augmented with $|S| - 1$ maximal inscribed disks, namely, the ones centered at the points $q(s) \in \mathcal{G}_S$, where $q(s)$ was the vertical projection onto $\mathcal{G}_S$ of a point with smallest ordinate on the site s. Denote with $\mathcal{A}_S$ the resulting domain after these $|S| - 1$ augmentation steps. We may conclude the main finding of this section as follows.

Lemma 2 *The tree $\mathcal{T}_S$ in (1) is the medial axis of the augmented domain $\mathcal{A}_S$.*

4 The algorithm

Using Lemma 2, the Voronoi diagram $V(S)$ can be obtained by computing the medial axis of the augmented domain $\mathcal{A}_S$.

The construction of $\partial\mathcal{A}_S$ is easy once the augmenting disks are available. Their centers lie on the edge graph $\mathcal{G}_S$ of $V(S)$ but, of course, the disks need to be found without knowledge of $\mathcal{G}_S$. A simple and efficient plane-sweep can be applied; we omit the details here. The construction can be implemented in $O(n \log n)$ time if the sites in S are described by a total of n objects, each being managable in constant time. Note that $\partial\mathcal{A}_S$ then consist of $\Theta(n)$ pieces.

Given the fact that $\partial\mathcal{A}_S$ is highly self-crossing, it may seem complicated to compute the medial axis of $\mathcal{A}_S$. However, $\mathcal{A}_S$ has a connected boundary, and therefore can be split into subdomains with the same property using maximal inscribed disks. This suggests a divide-and-conquer algorithm for computing $\mathrm{MA}(\mathcal{A}_S)$. The domain and its medial axis tree are split recursively, until directly solvable base cases remain. For simply connected shapes, a similar approach has been applied in [2, 1].

Recall that $\partial\mathcal{A}_S$ consists of pieces that bound inscribed disks (called *artificial arcs*) and pieces that stem from site boundaries (called *site segments*). To calculate a splitting disk, we fix some point p on a site segment and compute a maximal inscribed disk D for $\mathcal{A}_S$ that touches $\partial\mathcal{A}_S$ at p. Starting with an (appropriately oriented) disk of large radius, $\partial\mathcal{A}_S$ is scanned and the disk is shrunk accordingly whenever an intersection with a site segment occurs. Intersections with artificial arcs are, however, ignored. This works correctly because the set of maximal inscribed disks is the same for $\mathcal{A}_S$ and for the original shape $\mathcal{B}$ (except for finitely many augmenting disks). Computing a splitting disk takes $O(n)$ time, if each object describing the sites can be handled in $O(1)$ time.

5 Practical aspects

In view of keeping the algorithm efficient, disks that split the domain $\mathcal{A}_S$ in a balanced way are desired. Unfortunately, computing such a disk with simple means turns out to be hard. We can, however, choose a disk D randomly, by taking a random site segment on $\partial\mathcal{A}_S$ as its basis. Objects on $\partial\mathcal{A}_S$ and edges of $\mathrm{MA}(\mathcal{A}_S)$ correspond to each other in an (almost) bijective way, which suffices to convey randomness from boundary objects to medial axis edges. For the analysis, we thus may suppose that the center c of D lies on every edge of $\mathrm{MA}(\mathcal{A}_S)$ with the same probability. Under the assumption that the graph diameter of $\mathrm{MA}(\mathcal{A}_S)$ is linear in n, the point c lies on the diameter with constant probability, and $\mathrm{MA}(\mathcal{A}_S)$ is split at c into two parts of expected size $\Theta(n)$. A randomized runtime of $O(n \log n)$ results.

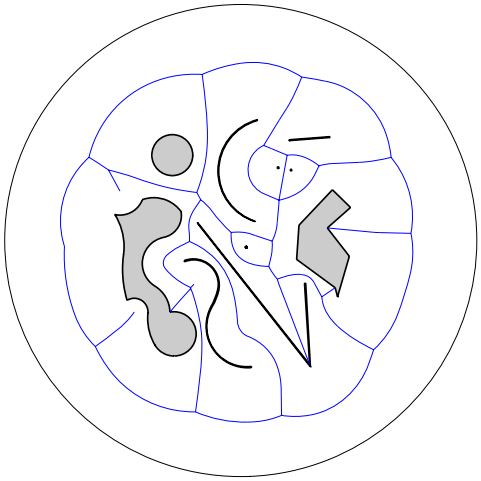

Figure 2: A mixed set of sites

n	atomic steps	ratio $n \log_2 n$	ratio $n(\log_2 n)^2$
507	6620	1.45	0.16
2070	32892	1.44	0.13
5196	91649	1.43	0.12
10474	199001	1.42	0.11
20488	417839	1.42	0.10
172198	4223178	1.41	0.09

Table 1: Count for complex sites bounded by n arcs

The assumption above is realistic in scenarios where a small number of sites is represented by a large number of individual objects. For example, if biarcs [12, 14] are used for approximation, then the number of leaves (hence also the number of vertices) of $\mathrm{MA}(\mathcal{A}_S)$ is determined by the original sites and not by the number of biarcs used. See Table 1, where step counts for our algorithm are averaged (and rounded) over 40 different equal-sized inputs.

The other extreme is the case of n point sites. Here, by the way how $\mathcal{A}_S$ is constructed, the diameter of $\mathrm{MA}(\mathcal{A}_S)$ will be typically much smaller, because many long 'vertical' branches will emanate from the surrounding circle Γ. As a simple heuristic, we may choose a small number of splitting disks tangent to Γ first, and continue with randomly splitting the result-

n	atomic steps	ratio $n \log_2 n$	ratio $n(\log_2 n)^2$
400	7591	2.20	0.25
2000	54662	2.49	0.23
4000	143391	3.00	0.25
20000	1015149	3.55	0.25
40000	2659149	4.35	0.28
200000	19820012	5.63	0.32

Table 2: Count for uniformly distributed point sites

ing augmented subdomains. See Table 2 for a sketch of the obtained results. We implemented the algorithm to accept circular arc input in its current version, including (though not optimizing) the handling of line segments and points. The Voronoi diagram in Figure 2 has been produced by this code.

The atomic step needed in the algorithm is an intersection test of a site-describing object and a given disk. This is among the simplest imaginable tests when a closest-site Voronoi diagram is to be computed by means of distance calculations. The structure and variety of the base cases depend on the type of sites. For point sites, there are only two of them, if the surrounding circle Γ is handled symbolically. They are of the simple form shown in Figure 3. (Artificial arcs are drawn dashed.) For circular arc splines, we get four generic base cases for C^1 continuity and nine additional cases for C^0 continuity; see [1].

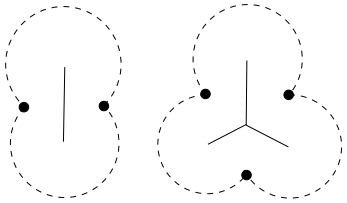

Figure 3: The two base cases for point sites.

6 Applications

We put particular emphasis on circular arcs as sites, because no practical algorithm for constructing their Voronoi diagram is available, and our algorithm naturally offers the ability to handle them. Moreover, biarcs [12, 14] enable a data-inexpensive and Voronoi diagram preserving approximation of general polynomial spline curves; we omit the details here.

The Voronoi diagram $V(S)$ of a set S of circularly approximated sites can be used as a tool for planning a robot motion in a piecewise-circular (PC)-environment [16]. Compared to piecewise-linear (PL)-environments, this offers several advantages. Not only can an approximation of $V(S)$ be computed more quickly now, but it also will consist of significantly fewer edges, namely, $\Theta(n^{\frac{2}{3}})$ instead of n. This leads to a more compact description of the paths the robot is supposed to move on. Another feature not shared by PL-environments is that the paths are locally C^1 between any two sites with C^1 boundaries, except for junctions with self-edges.

Several algorithms for shape offsetting are based on the Voronoi diagram or the medial axis [9, 8, 4]. Once more, a PC-representation of the input shape is advantageous, because the class of such shapes is closed under offsetting operations. Our Voronoi diagram algorithm is particularly well suited to the offsetting task, because it delivers the combinatorial structure without computing the edge graph explicitly.

References

[1] O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, and M. Rabl. Medial axis computation for planar free-form shapes. *Computer-Aided Design*. To appear.

[2] O. Aichholzer, F. Aurenhammer, T. Hackl, B. Jüttler, M.Oberneder, and Z. Sír. Computational and structural advantages of circular boundary representation. In *Springer Lecture Notes in Computer Science*, volume 4619, pages 374–385, 2007.

[3] H. Alt, O. Cheong, and A. Vigneron. The Voronoi diagram of curved objects. *Discrete & Computational Geometry*, 34:439–453, 2005.

[4] L. Cao and J. Liu. Computation of medial axis and offset curves of curved boundaries in planar domain. *Comput. Aided Des.*, 40(4):465–475, 2008.

[5] L. Chew and R. Drysdale. Voronoi diagrams based on convex distance functions. In *Proc. 1st Ann. ACM Symposium on Computational Geometry*, pages 235–244, 1985.

[6] H. Choi, S. Choi, and H. Moon. Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 181:57–88, 1997.

[7] S. Fortune. A sweep line algorithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987.

[8] M. Held. Voronoi diagrams and offset curves of curvilinear polygons. *Computer-Aided Design*, 30(4):287–300, 1998.

[9] M. Held, G. Lukacs, and L. Andor. Pocket machining based on contour-parallel tool paths generated by means of proximity maps. *Computer-Aided Design*, pages 189–203, 1994.

[10] D. Kirkpatrick. Efficient computation of continuous skeletons. In *Proc. 20th Ann. Symp. Foundations of Computer Science*, pages 18–27, 1979.

[11] R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Computational Geometry: Theory and Applications*, 3:157–184, 1993.

[12] D. S. Meek and D. J. Walton. Spiral arc spline approximation to a planar spiral. *J. Comput. Appl. Math.*, 107:21–30, 1999.

[13] M. Shamos and D. Hoey. Closest-point problems. In *Proc. 16th Ann. Symp. Foundations of Computer Science*, pages 151–162, 1975.

[14] Z. Šír, R. Feichtinger, and B. Jüttler. Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics. *Comp.–Aided Design*, 38:608–618, 2006.

[15] C.-K. Yap. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete & Computational Geometry*, 2:365–393, 1987.

[16] C.-K. Yap and H. Alt. Motion planning in the CL-environment. In *Lecture Notes In Computer Science 382*, pages 373–380, 1989.