ELSEVIER

Journal of Computational and Applied Mathematics 102 (1999) 253–277

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Voronoi diagram and medial axis algorithm for planar domains with curved boundaries — II: Detailed algorithm description

Rajesh Ramamurthy, Rida T. Farouki *

Department of Mechanical Engineering and Applied Mechanics, University of Michigan, Ann Arbor, MI 48109, USA

Received 26 May 1998

Abstract

Details of algorithms to construct the Voronoi diagrams and medial axes of planars domain bounded by free-form (polynomial or rational) curve segments are presented, based on theoretical foundations given in the first installment Ramamurthy and Farouki, J. Comput. Appl. Math. (1999) 102 119–141 of this two-part paper. In particular, we focus on key topological and computational issues that arise in these constructions. The topological issues include: (i) the data structures needed to represent various geometrical entities — bisectors, Voronoi regions, etc., and (ii) the Boolean operations (i.e., union, intersection, and difference) on planar sets required by the algorithm. Specifically, representations for the Voronoi polygons of boundary segments, and for individual Voronoi diagram or medial axis edges, are proposed. Since these edges may be segments of (a) nonrational algebraic curves (curve/curve bisectors); (b) rational curves (point/curve bisectors); or (c) straight lines (point/point bisectors), data structures tailored to each of these geometrical entities are introduced. The computational issues addressed include the curve intersection algorithms required in the Boolean operations, and iterative schemes used to precisely locate *bifurcation* or "n-prong" points ($n \geqslant 3$) of the Voronoi diagram and medial axis. A selection of computed Voronoi diagram and medial axis examples is included to illustrate the capabilities of the algorithm.

Keywords: Voronoi diagram; Medial axis; Bisectors; Distance functions; Bifurcation points

1. Introduction

The *Voronoi diagram* of a planar domain D, bounded by N curve segments, is a partition of the plane into N regions — not necessarily disjoint — such that each point within any one region is at least as close to its associated boundary segment as to all the other segments. The *medial axis* or

* Corresponding author. E-mail: farouki@ucdavis.edu.

0377-0427/99/$ – see front matter
PII: S0377-0427(98)00223-4

skeleton of the domain D, on the other hand, is the locus of centers of maximal circles (touching the boundary in at least two points) that may be inscribed within it.[1] The "medial axis transform" (MAT) of D comprises its medial axis with a superposed *radius function*, specifying the radius r of the maximal circle centered at each medial axis point. The boundary of D can, in principle, be recovered from its MAT as the envelope of all such maximal circles.

The a priori construction of Voronoi diagrams and medial axes has proven to be useful in a variety of application contexts — including pattern analysis and shape recognition [2, 3]; image compression [4]; computing offset curves and tool path generation for NC machining [5, 8, 21, 29]; surface fitting [19]; font design [7]; finite element mesh generation [18, 20, 36, 37]; and computing equivalent resistance networks for VLSI circuits [27].

Motivated by these diverse applications, a number of algorithms for the computation of Voronoi diagrams/medial axes have been published — mainly in the computational geometry literature, and for "simple" (i.e., piecewise-linear/circular) boundary curves. Voronoi diagram/medial axis algorithms have been developed by Preparata [30] for convex and nonconvex polygons; by Lee [26] for general (simply connected) polygonal domains; by Srinivasan and Nackman [35] for multiply connected polygonal domains; and by Held [21] for multiply connected domains with piecewise-linear/circular boundaries. Recently, Chou [9] and Choi et al. [6, 7] have considered domains bounded by general (analytic) free-form curve segments.

The Voronoi diagram and medial axis of a planar domain may be regarded as planar graphs, whose edges are portions of point/curve and curve/curve *bisectors* — i.e., loci that are equidistant from certain points or segments of the domain boundary (see Section 2.1) — and whose nodes are intersection points of these bisector segments. Thus, all Voronoi diagram/medial axis construction algorithms must address issues concerned with:

- specifying the *geometrical* information pertaining to the definitions of these edges and nodes;
- specifying the *topological* information that characterizes the connectivity relations of these edges and nodes.

For domains with piecewise-linear/circular boundaries, the bisectors are just linear or conic segments [21, 29, 38]. These can be represented exactly as rational Bézier curves [23], and the computation of their intersections reduces to a quadratic or quartic polynomial root-finding problem. Hence, the edges and the nodes of the Voronoi diagrams and medial axes for such domains can be computed in an essentially exact manner, without resorting to numerical approximations. Consequently, the emphasis in the algorithms of Preparata [30], Lee [26], Srinivasan and Nackman [35], and Held [21] has been primarily on topological issues and questions of computational efficiency.

However, for domains bounded by free-form (polynomial/rational) curves, the bisector constructions are more challenging since curve/curve bisectors do not, in general, admit "simple" exact representations. Correspondingly, one must use iterative numerical methods to locate the nodes of the Voronoi diagram and medial axis (which are the intersections of such bisectors). Note that the free-form boundary context demands the use of specially formulated algorithms — making piecewise-linear approximations to such a boundary, and then invoking a polygonal-domain algorithm, yields results that are not even *qualitatively* (i.e., topologically) correct; see [32].

[1] Note that the medial axis is defined purely by the *geometry* of the boundary, but the Voronoi diagram depends also on its *segmentation* — see [32] for details.

Algorithms to compute the medial axes and Voronoi diagrams of domains bounded by free-form curve segments must devote careful consideration to the geometrical and topological issues noted above. The algorithm by Choi et al. [6] addresses these issues through an ingenious domain decomposition scheme that locates all "special" (i.e., terminal or bifurcation) points of the medial axis through a numerical scheme, and establishes their connectivity (as nodes) in a tree data structure. The edges that connect these points are then amenable to approximation by interpolating discretely sampled bisector point/tangent data. The algorithm also accommodates multiply connected domains by invoking a pre-processing step called *homology killing*.

Our own Voronoi diagram/medial axis algorithm for planar domains with curved boundaries adopts a more traditional approach, based on incremental introduction of the boundary segments and explicit computations of their Voronoi regions at each stage. The theoretical foundations for this algorithm, which draws on results from preparatory studies [12, 13, 15, 16] of point/curve and curve/curve bisectors, are presented in the companion paper [32].

The principles guiding the design of our algorithm are: (i) to capture exact (i.e., rational) parameterizations of the Voronoi diagram/medial axis edges wherever possible; (ii) to provide piecewise-polynomial approximations that satisfy a given geometrical tolerance, for all other edges; and (iii) to remain faithful, within the specified tolerance, to the true topology of the medial axis and Voronoi diagram. In particular, feature (i) is unique to our method among the currently available Voronoi diagram/medial axis algorithms for free-form boundaries — such as those of Choi et al. [7] and Chou [9].

To confine this paper to a reasonable length, we assume that the reader is familiar with the point/curve and curve/curve bisector algorithms described in our earlier papers, cited above. These bisector algorithms capture rational edges exactly, approximate all other edges to within a prescribed tolerance, and explicitly identify all "special" points (e.g., tangent discontinuities) of the bisectors — in both generic and degenerate cases. In this paper, we shall focus on the details of the high-level geometrical and topological procedures employed by our Voronoi diagram/medial axis algorithm, which invokes the bisector computations as a basic tool — see also [33].

Our plan for the remainder of this paper is as follows. We commence in Section 2 with formal definitions for (i) the Voronoi region and polygon of a single boundary segment, and (ii) the Voronoi diagram of a collection of plane curve segments. We then state a key proposition, upon which the Voronoi diagram algorithm is based. The topological and computational issues that arise in implementing this algorithm are discussed in detail in Section 3, while Section 4 gives a complete step-by-step algorithm description. In Section 5 we address the issue of computing the nodes of the Voronoi diagram/medial axis, also called their *bifurcations*. Our attention then turns to the medial axis in Section 6, wherein a precise definition is given, and the Voronoi diagram algorithm is extended to generate the medial axis. Finally, Section 7 presents a selection of computed Voronoi diagram/medial axis examples, and Section 8 offers some concluding remarks.

2. Voronoi diagrams

Our Voronoi diagram algorithm, described in Section 4, proceeds in an incremental manner by the introduction of one boundary segment at a time. Hence, it is necessary that the concepts of

Voronoi region, Voronoi polygon, and Voronoi diagram be defined not only for the boundaries of closed domains, but also for arbitrary collections of planar curve segments.

2.1. Voronoi regions

For brevity, we shall assume that the reader is familiar with the definitions and properties of: (i) the point/curve distance function; (ii) "interior" and "terminal" footpoints of a point on a regular curve; (iii) the bisector of a point and a regular curve; (iv) the bisector of two regular curves; and (v) the self-bisectors of "simple" curves — see [32] for complete details.

Definition 2.1. Let $\{s_1,\ldots,s_M\}$, with $M>1$, be a subset of the N curve segments that comprise the boundary S of a planar domain D, and let $S_M = s_1 \cup \cdots \cup s_M \subseteq S$. Then:
(a) the *Voronoi region* $\mathrm{VR}(s_i)$ of boundary segment s_i, with respect to S_M, is the area defined by

$$\{\boldsymbol{q} \in \boldsymbol{R}^2 \mid \mathrm{dist}(\boldsymbol{q}, s_i) \leqslant \mathrm{dist}(\boldsymbol{q}, s_j) \quad \text{for } 1 \leqslant j \leqslant M,\ j \neq i\}; \tag{1}$$

(b) the *Voronoi polygon*[2] $\mathrm{VP}(s_i)$ of segment s_i, with respect to S_M, is the boundary of $\mathrm{VR}(s_i)$;
(c) the *Voronoi diagram* $\mathrm{VD}(S_M)$ of the segment set S_M is defined by

$$\mathrm{VD}(S_M) = \bigcup_{i=1}^{M} \mathrm{VP}(s_i). \tag{2}$$

Remark 2.2. The Voronoi region $\mathrm{VR}(s_i)$ and polygon $\mathrm{VP}(s_i)$ are dependent on the segment set $S_M = s_1 \cup \cdots \cup s_M$ currently under consideration — they change, in general, upon introducing a new segment s_{M+1} and considering them with respect to $S_{M+1} = s_1 \cup \cdots \cup s_M \cup s_{M+1}$. When $M = 1$, the set S_1 consists of the single segment s_1, and we adopt the convention that $\mathrm{VR}(s_1)$ with respect to S_1 comprises the entire plane.

The boundaries of the Voronoi regions are evidently loci of points that are equidistant from distinct segments, s_i and s_j, of the boundary. Thus, edges of the Voronoi polygons must be portions of curve/curve bisectors (subsets of which may actually be point/curve bisectors) for distinct boundary segments. The construction of the Voronoi polygons thus requires robust methods for curve/curve bisector computations. The algorithms developed in our earlier studies [12, 13, 15, 16] satisfy this need.

2.2. Voronoi diagram algorithm — theory

To keep the discussion reasonably self-contained, we now briefly review the theoretical basis of our Voronoi diagram algorithm (see [32] for details).

The algorithm commences with a boundary segment set comprising a single segment, whose Voronoi region with respect to itself is the entire plane (from Remark 2.2). We then incrementally augment the boundary segment set, introducing one additional boundary segment at a time, and

[2] It is customary to call $\mathrm{VP}(s_i)$ a "polygon" although, in general, it has curved edges.

re-construct the Voronoi regions of each segment with respect to this augmented set. Note that introducing a single segment may alter any or all of the current Voronoi regions (with respect to the augmented set). When all the boundary segments have been incorporated in this manner, with their Voronoi regions/polygons updated at each step, the Voronoi diagram of the entire boundary is simply the union of the final Voronoi polygons.

The following proposition indicates how the Voronoi regions of the existing segments are updated upon introducing a new segment, and also how the Voronoi region of the newly introduced segment is determined:

Proposition 2.3. *Let $S_M = s_1 \cup \cdots \cup s_M$ be a subset of the curve segments comprising the boundary S of a planar domain D, and let $\mathrm{VR}(s_i)$ denote the Voronoi region of segment s_i with respect to S_M, for $1 \leqslant i \leqslant M$. Introducing a new segment s_{M+1}, and setting $S_{M+1} = s_1 \cup \cdots \cup s_{M+1}$, the bisector of s_i and s_{M+1} partitions $\mathrm{VR}(s_i)$ into three disjoint subsets such that*

1. *$V_<(s_i)$ is the subset of points in $\mathrm{VR}(s_i)$ closer to s_i than to s_{M+1};*
2. *$V_=(s_i)$ is the subset of points in $\mathrm{VR}(s_i)$ equidistant from s_i and s_{M+1};*
3. *$V_>(s_i)$ is the subset of points in $\mathrm{VR}(s_i)$ closer to s_{M+1} than to s_i.*

Then, for $1 \leqslant i \leqslant M$, we have

$$\mathrm{VR}(s_i)\,\text{w.r.t. } S_{M+1} = \mathrm{VR}(s_i)\,\text{w.r.t. } S_M - V_>(s_i), \tag{3}$$

while the Voronoi region of the newly introduced segment is given by

$$\mathrm{VR}(s_{M+1})\,\text{w.r.t. } S_{M+1} = \bigcup_{i=1}^{M} \mathrm{VR}(s_i)\,\text{w.r.t. } S_M - V_<(s_i). \tag{4}$$

Proof. See [32]. □

To compute $V_<(s_i)$ and $V_>(s_i)$, we introduce [32] the regions defined by

$$\begin{aligned} \overline{V}_<(s_i, s_{M+1}) &= \{\boldsymbol{q} \in \boldsymbol{R}^2 \mid \mathrm{dist}(\boldsymbol{q}, s_i) < \mathrm{dist}(\boldsymbol{q}, s_{M+1})\}, \\ \overline{V}_>(s_i, s_{M+1}) &= \{\boldsymbol{q} \in \boldsymbol{R}^2 \mid \mathrm{dist}(\boldsymbol{q}, s_i) > \mathrm{dist}(\boldsymbol{q}, s_{M+1})\}, \end{aligned} \tag{5}$$

and we then have

$$\begin{aligned} V_<(s_i) &= \mathrm{VR}(s_i)\,\text{w.r.t. } S_M \cap \overline{V}_<(s_i, s_{M+1}), \\ V_>(s_i) &= \mathrm{VR}(s_i)\,\text{w.r.t. } S_M \cap \overline{V}_>(s_i, s_{M+1}). \end{aligned} \tag{6}$$

It was shown in [32] that the boundaries of $\overline{V}_<(s_i, s_{M+1})$ and $\overline{V}_>(s_i, s_{M+1})$ are certain subsets of the "boundary" of the bisector[3] of s_i and s_{M+1}.

Thus, using (3), (4), and (6), we can update the existing Voronoi regions, and compute the Voronoi region of the newly introduced segment. However, as the following corollary shows, the Boolean intersection operations in (6) can be avoided, and we may use (7) and (8) below in lieu of (3) and (4).

[3] We speak of the *boundary* of the bisector, since this bisector is of mixed dimension when s_i and s_{M+1} share a common endpoint with C^0 continuity — see [16].

Corollary 2.4. *The updated Voronoi regions can be expressed as*

$$\mathrm{VR}(s_i)\,\text{w.r.t. } S_{M+1} = \mathrm{VR}(s_i)\,\text{w.r.t. } S_M - \overline{V}_>(s_i, s_{M+1}) \tag{7}$$

for $1 \leqslant i \leqslant M$, *while for the newly introduced segment we have*

$$\mathrm{VR}(s_{M+1})\,\text{w.r.t. } S_{M+1} = \bigcup_{i=1}^{M} \mathrm{VR}(s_i)\,\text{w.r.t. } S_M - \overline{V}_<(s_i, s_{M+1}). \tag{8}$$

Proof. We show that, for $1 \leqslant i \leqslant M$, each point $\boldsymbol{q}$ of $\mathrm{VR}(s_i)$ w.r.t. $S_M - V_>(s_i)$ belongs to $\mathrm{VR}(s_i)$ w.r.t. $S_M - \overline{V}_>(s_i, s_{M+1})$, and vice versa, so that these two sets are identical, and hence expression (7) follows from (3).

First, suppose $\boldsymbol{q} \in \mathrm{VR}(s_i)$ w.r.t. $S_M - V_>(s_i)$. Then, $\boldsymbol{q} \in \mathrm{VR}(s_i)$ w.r.t. S_M and $\boldsymbol{q} \notin V_>(s_i)$. Thus, by the definition of $V_>(s_i)$, $\mathrm{dist}(\boldsymbol{q}, s_i) \not> \mathrm{dist}(\boldsymbol{q}, s_{M+1})$ and so $\boldsymbol{q} \notin \overline{V}_>(s_i, s_{M+1})$. Hence, $\boldsymbol{q} \in \mathrm{VR}(s_i)$ w.r.t. $S_M - \overline{V}_>(s_i, s_{M+1})$.

Conversely, if $\boldsymbol{q} \in \mathrm{VR}(s_i)$ w.r.t. $S_M - \overline{V}_>(s_i, s_{M+1})$, then $\boldsymbol{q} \in \mathrm{VR}(s_i)$ w.r.t. S_M and $\boldsymbol{q} \notin \overline{V}_>(s_i, s_{M+1})$. However, since $V_>(s_i) \subseteq \overline{V}_>(s_i, s_{M+1})$ by (6), $\boldsymbol{q} \notin V_>(s_i)$ and so $\boldsymbol{q} \in \mathrm{VR}(s_i)$ w.r.t. $S_M - V_>(s_i)$. This completes the proof of (7). In a similar fashion, it can be shown that, for $1 \leqslant i \leqslant M$, $\mathrm{VR}(s_i)$ w.r.t. $S_M - V_<(s_i)$ and $\mathrm{VR}(s_i)$ w.r.t. $S_M - \overline{V}_<(s_i, s_{M+1})$ are equal so that, from (4), expression (8) is true. □

3. Implementation issues

Although Proposition 2.3 and Corollary 2.4 furnish the theoretical basis for our Voronoi diagram algorithm, a number of issues remain to be addressed to facilitate a practical implementation. These include:

(1) restricting the computations to a finite subset of the plane;
(2) developing a data structure to represent the boundaries of $\mathrm{VR}(s_i)$, $\overline{V}_<(s_i, s_{M+1})$, $\overline{V}_>(s_i, s_{M+1})$;
(3) developing data structures for the three types of curve segment that the define edges of the Voronoi diagram or medial axis – namely: (a) nonrational algebraic curve/curve bisectors; (b) rational point/curve bisectors; and (c) linear point/point bisectors;
(4) methods for computing the intersections between similar or dissimilar boundary segment types on $\mathrm{VR}(s_i)$, $\overline{V}_<(s_i, s_{M+1})$, $\overline{V}_>(s_i, s_{M+1})$;
(5) and algorithms for the Boolean set operations (difference and union) required in (7) and (8).

Note that the inputs and outputs of our Voronoi diagram algorithm are sets of polynomial or rational Bézier curves. A degree-n polynomial Bézier curve is defined by the coordinates (x_j, y_j), $0 \leqslant j \leqslant n$, of its control points:

```
BEZIER_POLY
{
    int n;
    double x[n+1];
    double y[n+1];
}
```

while a degree-n rational Bézier curve is defined by control points *and* "weights" $w_0, \ldots, w_n$ [10]:

```
BEZIER_RATIONAL
{
     int n;
     double x[n+1];
     double y[n+1];
     double w[n+1];
}
```

3.1. Computational domain

In principle, an algorithm can be based on Proposition 2.3 and Corollary 2.4 to construct the unbounded Voronoi diagram (both affine and semi-infinite Voronoi edges) of the boundary S of a domain D. From a practical standpoint, however, implementing such an algorithm incurs several difficulties:

(i) It is difficult to construct piecewise-rational approximations, satisfying a given geometrical tolerance, to semi-infinite (non-rational) algebraic segments of the Voronoi diagram, since the error analysis described in [15] can only accommodate *finite* bisector segments.

(ii) The intersection calculations [34] between rational curve segments that arise when evaluating the Boolean expressions (7) and (8) are likewise incompatible with semi-infinite segments; most intersection algorithms rely on affine curve subdivision methods.

(iii) Finally, in evaluating (7) and (8), it is necessary to determine if certain points lie inside or outside of the regions $\mathrm{VR}(s_i)$, $\overline{V}_<(s_i, s_{M+1})$, and $\overline{V}_>(s_i, s_{M+1})$. Such inclusion tests are much simpler for bounded sets.

Thus, it is convenient to restrict the Voronoi diagram construction to some user-defined finite subset $\mathscr{A}$ of $\boldsymbol{R}^2$. Typically, one chooses a sufficiently large bounding box for S as the computational domain[4] $\mathscr{A}$, and we shall follow this practice for all the examples illustrated in this paper.

3.2. Data structure for the boundaries of $\mathrm{VR}(s_i)$, $\overline{V}_>(s_i, s_{M+1})$, and $\overline{V}_<(s_i, s_{M+1})$

The Voronoi region $\mathrm{VR}(s_i)$ can be specified by its oriented[5] boundary, $\mathrm{VP}(s_i)$. As noted in Section 2.1, the edges of $\mathrm{VP}(s_i)$ are portions of the bisectors of s_i with other boundary segments s_j $(j \neq i)$. Hence, each edge of the Voronoi diagram of S belongs to the Voronoi polygon of (at least) two boundary segments, and directly storing the geometric information for each Voronoi polygon edge is redundant. Furthermore, since these edges can be nonrational algebraic curve segments, rational curve segments, or line segments, it is natural to store them in distinct data structures tailored to their individual types.

[4] Introducing the computational domain $\mathscr{A}$ does not alter the geometry or topology of the Voronoi diagram in its interior — i.e., the portion of the *unbounded* Voronoi diagram that lies within $\mathscr{A}$ is identical to the restricted Voronoi diagram computed by an algorithm that operates in the interior of $\mathscr{A}$ only.

[5] $\mathrm{VP}(s_i)$ is oriented such that the interior of $\mathrm{VR}(s_i)$ lies to the *left* as $\mathrm{VP}(s_i)$ is traversed in the sense of its parameterization.

Consider now the following data structure for a single edge E among the N edges of the Voronoi polygon VP(s_i):

```
VORONOI_EDGE
{
     char type;
     int position;
     char orient;
     int start_point;
     int end_point;
}
```

Here, `type` $\in$ { `a`,`r`,`l`} specifies whether E is an algebraic, rational, or linear segment, and the integer `position` gives the location of edge E (depending on `type`) in arrays that contain the geometrical definitions of all algebraic, rational, or linear segments[6] in the Voronoi diagram. The character `orient` takes the value `f` or `r` according to whether the orientation of E is *identical* or *opposite* to that of its parameterization.

It is convenient to represent VP(s_i) by an ordered set of N elements of type `VORONOI_EDGE`, the end point of segment j coinciding with the start point of segment $j+1$ mod N, for $0 \leqslant j < N$. This ordering defines the sense of parameterization of VP(s_i). For this purpose, we need to store: (a) in each `VORONOI_EDGE` element, pointers to its start and end points; and (b) a separate array for storing the coordinates of all such terminal points. The former information is stored in the two integers `start_point` and `end_point`, while the latter is stored in an array of points associated with VP(s_i).

Finally, note that arrays of `VORONOI_EDGE` elements may also be used to represent the oriented boundaries of $\overline{V}_<(s_i, s_{M+1})$ and $\overline{V}_>(s_i, s_{M+1})$, provided their associated algebraic, rational, and linear boundary segments and their terminal points are stored in separate arrays for each type.

3.3. Data structures for bisector segments

Since curve/curve bisectors are generically (subsets of) nonrational algebraic curves [13], they must be approximated by polynomial or rational segments [15]. Using the error analysis developed in [15], these approximations can be guaranteed to satisfy any user-specified geometrical tolerance.

The approximate curve/curve bisector $\mathscr{B}$ for boundary segments s_1 and s_2 may be constructed as follows: (i) an ordered sequence of n points, including all of the "special" points — i.e., *critical* or *transition* points [15] — lying precisely on $\mathscr{B}$, is computed;[7] and (ii) parabolic or cubic interpolants to position/tangent/curvature data, satisfying the desired tolerance, are then fitted between consecutive pairs of points. Thus, the data structure for the approximate representation of $\mathscr{B}$ should store the Bézier control points for each of the $n-1$ parabolic/cubic interpolants. It must also store the identities of the two curves s_1 and s_2 that $\mathscr{B}$ bisects.

[6] Data structures for these segments are explained in the following section.

[7] Note that these points have an induced parameterization [15] that can be associated with either of the boundary segments.

Now suppose that we need to revise the approximate representation of $\mathscr{B}$ by introducing an additional bisector point[8] to the existing sequence of points on $\mathscr{B}$. This need arises when (i) it is necessary to refine the approximation to a tighter geometrical tolerance; and (ii) when a newly identified bifurcation point[9] of the Voronoi diagram/medial axis, lying on $\mathscr{B}$, must be introduced. The appropriate position of the new point in the sequence may be determined by comparing the parameter value of its footpoint on s_1 (say) with those of existing points. Thus, there is also a need to store footpoint parameter values on s_1 and s_2 for each of the interpolated bisector points.

Based on the above considerations, we now introduce the following data structure for the approximate representation of curve/curve bisectors:

```
ALGEBRAIC_CURVE
{
     int number_of_exact_points;
     BEZIER_POLY segment[number_of_exact_points-1];
     int curve1;
     int curve2;
     double foot1[number_of_exact_points];
     double foot2[number_of_exact_points];
}
```

As previously noted, point/curve bisectors in the Voronoi diagram/medial axis are generically rational loci, and can be exactly represented by rational Bézier curves of appropriate degree. To aid in the intersection calculations required by the Boolean operations, the identities of the point and curve that define such bisectors are also stored (this information is required as input to the numerical scheme for computing exact coordinates of bifurcation points in the Voronoi diagram/medial axis — see Section 3.5). Thus, the data structure that represents rational point/curve bisectors has the form

```
RATIONAL_CURVE
{
     BEZIER_RATIONAL segment;
     int end_point;
     int curve;
}
```

Finally, to justify the data structure used to represent linear segments of the Voronoi diagram/medial axis, we note that such segments can be of two types: (i) perpendicular bisectors of pairs of distinct endpoints of boundary segments; and (ii) normal lines to boundary segments at their endpoints. Thus, we represent a line segment as a polynomial Bézier curve (since this is convenient for intersection calculations) together with the identities of two points. If the points are distinct, they identify the curve endpoints for a type (i) segment, and if they are identical a type (ii) segment is

[8] Clearly, this approach can be used to introduce several new points, one at a time.

[9] Bifurcation points are systematically identified as the algorithm proceeds through the sequential introduction of additional boundary segments.

indicated:

```
STRAIGHT_LINE
{
     BEZIER_POLY segment;
     int end_point1;
     int end_point2;
}
```

3.4. Implementation of the Boolean operations

To perform the Boolean operations in expressions (7) and (8), descriptions for the oriented boundaries of $\mathrm{VR}(s_i)$ w.r.t. S_M, $\overline{V}_<(s_i, s_{M+1})$, and $\overline{V}_>(s_i, s_{M+1})$, for $1 \leqslant i \leqslant M$, are necessary and sufficient.

The oriented boundary of $\mathrm{VR}(s_i)$ w.r.t. S_M is already known from prior computations, as the Voronoi polygon of s_i w.r.t. S_M. Hence, it suffices to construct the oriented boundaries of $\overline{V}_<(s_i, s_{M+1})$ and $\overline{V}_>(s_i, s_{M+1})$. The theory underlying the construction of these sets, for cases where the segments s_i and s_{M+1} are either disjoint or share a common endpoint, was described at length in [32], and is therefore omitted here.

Now suppose A and B are planar domains, with oriented boundaries ∂A and ∂B. Then the boundaries $\partial(A \cup B)$, $\partial(A - B)$, and $\partial(A \cap B)$ of the union, difference, and intersection of A and B are all subsets of $\partial A \cup \partial B$. To determine these subsets, we classify the segments of ∂A as follows:

(a) those whose points lie entirely inside of B, outside of B, or on ∂B;
(b) those whose points lie partly inside and partly outside of B;
(c) those whose points lie partly inside of B, and partly on ∂B;
(d) those whose points lie partly outside of B, and partly on ∂B.

The segments of ∂B may be likewise classified with respect to set A. Now since the "inclusion status" of points on the segments of types (b)–(d) on ∂A changes, these segments must intersect[10] ∂B, and vice versa. Hence, if we split segments of these types on both ∂A and ∂B at the intersection points, each resulting subsegment of both ∂A and ∂B will be of type (a).

Given this preparatory subdivision, we may identify the segments of ∂A that belong to $\partial(A \cup B)$, $\partial(A - B)$, and $\partial(A \cap B)$ from Table 1, once we determine their status as being either (i) inside B; (ii) outside B; (iii) on ∂B with the same orientation; or (iv) on ∂B with the opposite orientation.

Finally, we need to orient the segments of the boundaries $\partial(A \cup B)$, $\partial(A - B)$, and $\partial(A \cap B)$ such that the interior of the sets $A \cup B$, $A - B$, and $A \cap B$ lies locally to the left as these boundaries are traversed in the sense of their parameterizations. This description of the computation of Booleans applies directly to the sets in Eqs. (7) and (8).

3.5. Intersection computations

The need to compute curve intersections arises in the initial subdivision of boundary segments described above for the Boolean operations in (7) and (8). The boundaries of $\mathrm{VR}(s_i)$ w.r.t. S_M, $\overline{V}_<(s_i, s_{M+1})$, and $\overline{V}_>(s_i, s_{M+1})$ are, in general, composed of portions of nonrational algebraic curves,

[10] If a segment lies partly on the boundary, and partly not on it, we also regard its point of departure from the boundary to be an intersection.

Table 1
The boundaries ∂A and ∂B have segments $a_1,\ldots,a_m$ and $b_1,\ldots,b_n$. According to whether a_i lies inside or outside B, or on its boundary ∂B with identical ("+on") or opposite ("−on") orientation, (a) indicates whether or not segment a_i belongs to the boundaries of $\partial(A\cup B)$, $\partial(A\cap B)$, and $\partial(A-B)$. Likewise for b_j with respect to A and ∂A in (b)

	a_i inside B	a_i outside B	a_i +on ∂B	a_i −on ∂B
(a) *Status* of $a_i\in\partial A$ *with respect to* B *and* ∂B				
$A\cup B$	no	yes	yes	no
$A\cap B$	yes	no	yes	no
$A-B$	no	yes	no	yes
	b_j inside A	b_j outside A	b_j +on ∂A	b_j −on ∂A
(b) *Status of* $b_j\in\partial B$ *with respect to* A *and* ∂A				
$A\cup B$	no	yes	no	no
$A\cap B$	yes	no	no	no
$A-B$	yes	no	no	no

rational curves, and line segments. Thus, intersections may arise between (i) two lines; (ii) a line and a rational curve; (iii) two rational curves; (iv) a rational and a nonrational algebraic curve; and (v) two nonrational algebraic curves.

Computing the intersection of two line segments is a trivial matter. Given a line segment and a rational curve, computing their intersections can be cast as a polynomial root-finding problem by expressing the line in implicit form. Standard algorithms can then be invoked to compute the real roots [25], using the numerically stable Bernstein form [14]. The implicitization approach is also useful when intersecting rational curves of degree $\leqslant 4$, but geometrical subdivision methods are faster and more reliable [34] for higher-order rational curves.[11] Further details on intersecting rational curves using subdivision methods may be found in [34]. A method similar to that described in [24] has been used in our Voronoi diagram/medial axis computations.

Nonrational curve/curve bisectors in the Voronoi diagram/medial axis are approximated [15] by piecewise-polynomial curves. Standard intersection methods may be applied to them, but the resulting intersection points are then necessarily approximate. Such intersection points correspond to certain special points, called *bifurcation* points of the Voronoi diagram/medial axis. To ensure an accurate topology for the Voronoi diagram and medial axis, it is necessary to compute the coordinates of bifurcation points to within the given tolerance. Since this problem is rather complex, we defer a detailed treatment of it to Section 5, and consider next the overall Voronoi diagram algorithm.

4. Voronoi diagram algorithm

We now present pseudo-code for our Voronoi diagram algorithm, based on Proposition 2.1, and taking into account the various issues addressed in the preceding section. In the sequel, we shall

[11] The bisector of a point $\boldsymbol{p}$ and a curve $\boldsymbol{r}(t)$ is a rational curve of degree $3n-1$ or $4n-2$, according to whether $\boldsymbol{r}(t)$ is a polynomial or rational curve of degree n [12].

denote the oriented boundary of a domain D by ∂D, and the complete bisector of the boundary segments s_i and s_{M+1} of D by $B(s_i, s_{M+1})$. Also, $\mathscr{A}$ is the given computational domain, containing the domain D whose Voronoi diagram we wish to compute.

input: boundary $S = \partial D = s_1 \cup \cdots \cup s_N$ of domain D
1. set $S_1 = s_1$ and $\mathrm{VP}(s_1)$ w.r.t. $S_1 = \partial\mathscr{A}$;
2. for $M = 1, \ldots, N-1$ do
   (a) set $S_{M+1} = S_M \cup s_{M+1}$ and $\mathrm{VP}(s_{M+1})$ w.r.t. $S_{M+1} = \mathrm{NULL}$;
   (b) for $i = 1, \ldots, M$ do
      1. if (s_i & s_{M+1} share a common end point)
         compute $B(s_i, s_{M+1})$ with degenerate bisector algorithm;
         else
         compute $B(s_i, s_{M+1})$ with generic bisector algorithm;
      2. construct $\partial \overline{V}_<(s_i, s_{M+1})$ and $\partial \overline{V}_>(s_i, s_{M+1})$;
      3. set $A = \mathrm{VR}(s_i)$ w.r.t. S_M and $B = \overline{V}_>(s_i)$;
         (i) compute the intersection points of ∂A and ∂B, and split their segments at these points;
         (ii) determine the status of each segment of ∂A as lying either interior or exterior to B, or lying on ∂B with the same or opposite orientation;
         (iii) repeat (ii) swapping A with B and ∂A with ∂B;
      4. compute $\partial V_>(s_i) = \partial(A \cap B)$;
      5. set $A = \mathrm{VR}(s_i)$ w.r.t. S_M and $B = V_>(s_i)$, and compute $\mathrm{VP}(s_i)$ w.r.t. $S_{M+1} = \partial(A - B)$;
      6. set $A = \mathrm{VR}(s_i)$ w.r.t. S_M and $B = \overline{V}_<(s_i, s_{M+1})$, repeat steps 3(i)–(iii), and compute $\partial V_<(s_i) = \partial(A \cap B)$;
      7. set $A = \mathrm{VR}(s_i)$ w.r.t. S_M and $B = V_<(s_i)$, and compute $\partial C = \partial(A - B)$;
      8. set $A = \mathrm{VR}(s_{M+1})$ w.r.t. S_{M+1} and $B = C$, repeat steps 3(i)–(iii), and compute $\mathrm{VP}(s_{M+1})$ w.r.t. $S_{M+1} = \partial(A \cup B)$;
   end do
end do
3. define $\mathrm{VD}(S) = \bigcup_{i=1}^{N} \mathrm{VP}(s_i)$ w.r.t. S_N;

output: Voronoi diagram $\mathrm{VD}(S)$ of boundary.

5. Identification of bifurcation points

Bifurcations of the Voronoi diagram/medial axis are special points with *three or more* footpoints on the domain boundary. It is difficult to devise a single algorithm that computes the exact coordinates of all the different types of bifurcations that may occur. Instead, depending on the number and nature of the footpoints on the boundary S, specially formulated algorithms are needed for each circumstance. For this purpose, we classify bifurcations into the following categories: (i) those with an infinite number of footpoints on S; (ii) those with (at least) two coincident footpoints on S; and (iii) those with a finite number ($\geqslant 3$) of distinct footpoints on S. Methods for computing the exact coordinates of bifurcations in these categories are discussed below.

5.1. Bifurcations with infinitely many footpoints

Bifurcations with an infinite number of footpoints on S are centers of circular segments of the domain boundary, as is shown by the following lemma.

Lemma 5.1. *Let $\boldsymbol{b}_* = (x_*, y_*)$ be a bifurcation point in the domain D with an infinite number of footpoints on its boundary $S = s_1 \cup \cdots \cup s_N$. Then $\boldsymbol{b}_*$ is the center of some circular segment s_i of the boundary.*

Proof. Since $\boldsymbol{b}_*$ has at most two terminal footpoints on each of the boundary segments, it has at most $2N$ terminal footpoints on S. Thus, $\boldsymbol{b}_*$ has infinitely many interior footpoints on S, and consequently on some segment s_i of S. Now $\boldsymbol{b}_*$ has an interior footpoint on s_i at $u = u_*$ only if u_* is a root of

$$P_\perp(u) = [x_* - X(u)]X'(u) + [y_* - Y(u)]Y'(u) \tag{9}$$

when s_i is the polynomial curve $\boldsymbol{r}(u) = (X(u), Y(u))$, or if u_* is a root of

$$\begin{aligned} P_\perp(u) = {} & [x_* W(u) - X(u)][W(u)X'(u) - W'(u)X(u)] \\ & + [y_* W(u) - Y(u)][W(u)Y'(u) - W'(u)Y(u)] \end{aligned} \tag{10}$$

when s_i is the rational curve $\boldsymbol{r}(u) = (X(u)/W(u),\ Y(u)/W(u))$. Now both (9) and (10) are polynomial equations with (at most) $2n - 1$ and $3n - 2$ real roots, respectively. Thus, $\boldsymbol{b}_*$ can have an infinite number of interior footpoints only if (9) or (10) vanishes identically and $|\boldsymbol{b}_* - \boldsymbol{r}(u)|$ is constant. The latter condition represents the equation of a circular arc centered at $\boldsymbol{b}_*$. □

Thus, to identify bifurcations with an infinite number of footpoints on S, we compare the radius r_i of each circular arc s_i of S with the distance of its center $\boldsymbol{c}_i$ from all the other segments s_j, $1 \leqslant j \neq i \leqslant N$, of S. If r_i equals the distance of $\boldsymbol{c}_i$ to some segment s_j, then $\boldsymbol{c}_i$ is a bifurcation point of the Voronoi diagram/medial axis with infinitely many footpoints on S. Moreover, $\boldsymbol{c}_i$ is a *transition point* [15] of the curve/curve bisector for s_i and s_j. Since such points are explicitly captured [15] in the construction of curve/curve bisectors, no special algorithms are required to include such bifurcations in the Voronoi diagram/medial axis of the domain.

5.2. Bifurcations with coincident footpoints

Next, we identify bifurcation points that have two *coincident* footpoints on a single segment s_i of S. As the following lemma shows, such points are centers of curvature for *vertices* (i.e., points of extremum curvature) on s_i.

Lemma 5.2. *Let $\boldsymbol{b}_* = (x_*, y_*)$ be a bifurcation point in the domain D with two coincident footpoints on segment s_i of the boundary $S = s_1 \cup \cdots \cup s_N$. Then $\boldsymbol{b}_*$ coincides with the center of curvature of a vertex on s_i.*

Proof. Suppose s_i is the polynomial curve $\boldsymbol{r}(u)=(X(u),Y(u))$ and $\boldsymbol{b}_*$ has two identical footpoints on s_i at $u=u_*$. Then $u=u_*$ is a double root of the polynomial $P_\perp(u)$ defined by (9). Hence, solving $P_\perp(u_*)=\partial P_\perp(u_*)/\partial u=0$ as a system of linear equations in (x_*,y_*), we find that

$$(x_*,y_*)=\left[(X,Y)+\frac{X'^2+Y'^2}{X'Y''-X''Y'}\,(-Y',X')\right]_{u=u_*},$$

and since $\boldsymbol{n}=(-Y',X')/(X'^2+Y'^2)^{1/2}$ and $\kappa=(X'Y''-X''Y')/(X'^2+Y'^2)^{3/2}$ are the normal and (signed) curvature of $\boldsymbol{r}(u)$, we see that $\boldsymbol{b}_*$ coincides with the center of curvature [12] or evolute point $\boldsymbol{r}(u_*)-\boldsymbol{n}(u_*)/\kappa(u_*)$ of the curve $\boldsymbol{r}(u)$ at $u=u_*$. Further, since $|\boldsymbol{b}_*-\boldsymbol{r}(u_*)|=1/\kappa(u_*)$, the "maximal" circle of D centered at $\boldsymbol{b}_*$ coincides with the osculating circle to $\boldsymbol{r}(u)$ at $u=u_*$.

Now suppose that $\kappa(u_*)$ is *not* a local maximum. Then, it is known from classical differential geometry [6, 22] that the osculating circle at $\boldsymbol{r}(u_*)$ cuts $\boldsymbol{r}(u)$ at $u=u_*$ so that parts of $\boldsymbol{r}(u)$ lie (locally) inside the osculating circle. But since, in this instance, the osculating circle is also the maximal circle, this is impossible. Hence, $\kappa(u_*)$ must be a local extremum — and the osculating circle then lies (locally) on one side of $\boldsymbol{r}(u)$ at $u=u_*$ [6, 22]. Hence, $\boldsymbol{b}_*$ is the center of curvature of a vertex on s_i. Analogous arguments hold in the case that s_i is a rational curve. □

Geometrically, the center of curvature at $u=u_*$ can be regarded as the limiting intersection point of "neighboring" normal lines to the curve $\boldsymbol{r}(u)$, at u_* and $u_*+\Delta u$, as $\Delta u\to 0$. Thus, a bifurcation point $\boldsymbol{b}_*$ coinciding with the center of curvature at $u=u_*$ is considered to have a "double" footpoint (or two coincident footpoints) at that point on the domain boundary S.

Since bifurcation points of this type have two footpoints on one boundary segment, s_i say, and (at least) one footpoint on another segment, s_j say, they are actually *critical points* [15] on the bisector of s_i and s_j. Such points are, in fact, computed while constructing the curve/curve bisector by locating those centers of curvature of the vertices of s_i and s_j that are equidistant from both curves. Hence, no separate algorithm is required to ensure inclusion of these bifurcations in the Voronoi diagram/medial axis of the domain.

5.3. *Bifurcations with distinct footpoints*

Bifurcations with precisely three distinct footpoints on S are the generic case, and are thus considered first below. According to the nature of their three footpoints on S, we may distinguish between three types of these generic bifurcation points:

(1) those that have three footpoints on a single boundary segment s_i;
(2) those that have two footpoints on a single boundary segment s_i, and the third footpoint on a different segment s_j;
(3) those that have each of their three footpoints on three distinct boundary segments, s_i, s_j, s_k.

Bifurcations of type (1) must have (at least) one terminal [13] footpoint on s_i. Now suppose s_i is represented by the parametric curve $\boldsymbol{r}(u)$ for $u\in[0,1]$. Then, depending on whether the bifurcation

[12] We adopt a sign convention in which the curvature is *positive* when the normal points *away* from the center of curvature.

[13] We assume the boundary is composed of "simple" segments (see [16, Definition 4.2]) and hence no point may have three "interior" footpoints on a *single* segment.

point has a terminal footpoint at either $\boldsymbol{r}(0)$ or $\boldsymbol{r}(1)$, it is a self-intersection [12] of the point/curve bisector of $\boldsymbol{r}(0)$ or $\boldsymbol{r}(1)$ with $\boldsymbol{r}(u)$. Since the point/curve bisector of $\boldsymbol{r}(0)$ and s_i is constructed when the bisector of s_{i-1} and s_i is computed,[14] all bifurcation points of type (1) lying on this point/curve bisector are identified at this time. Similarly, all bifurcation points of type (1) lying on the bisector of $\boldsymbol{r}(1)$ and s_i are located while constructing the bisector of s_i and s_{i+1}. In this manner, all type (1) bifurcations of the Voronoi diagram are located precisely, and no further refinement of their coordinates is necessary.

Bifurcations of type (2) are "exceptional" (critical or transition) points — see [15] — on the curve/curve bisector of segments s_i and s_j. Such points are explicitly located by a bisection method while computing this curve/curve bisector [15]. The bifurcations of type (3), however, must be explicitly located during the construction of the Voronoi diagram. Based upon the nature of their footpoints on s_i, s_j, s_k, we can further differentiate among the type (3) bifurcation points as follows:

(3a) those with *terminal* footpoints on all three of the segments s_i, s_j, s_k;

(3b) those with *terminal* footpoints on two of the segments s_i, s_j, s_k and an *interior* footpoint on the third;

(3c) those with *interior* footpoints on two of the segments s_i, s_j, s_k and a *terminal* footpoint on the third;

(3d) those with *interior* footpoints on all three of the segments s_i, s_j, s_k.

Bifurcations of type (3a) are points of concurrency of three linear segments in the Voronoi diagram, while those of type (3b) correspond to the common intersection of a linear segment and two rational point/curve bisectors. The locations of such bifurcation points computed by standard curve-intersection algorithms are essentially exact, and require no further refinement.

Bifurcations of type (3c) in the Voronoi diagram arise where two rational (point/curve) and one nonrational (curve/curve) bisector segments meet, while those of type (3d) are the intersections of three nonrational segments. Since the nonrational bisector segments must be approximated by Hermite interpolants to discrete data, the bifurcation-point coordinates computed as their intersections are inherently approximate. We describe below how these approximate coordinates can be used as input to a Newton–Raphson scheme that gives essentially *exact* bifurcation-point locations, based on the original boundary segments.[15] We focus on type (3d) bifurcations below, and briefly mention the necessary adaptations for type (3c) bifurcations.

Let $\boldsymbol{b}_0$ be a starting approximation to a type (3d) bifurcation point $\boldsymbol{b}_*$ in the Voronoi diagram, which is equidistant from the three boundary segments $\boldsymbol{q}(t), \boldsymbol{r}(u), \boldsymbol{s}(v)$ with $t, u, v \in [0, 1]$. Here, $\boldsymbol{b}_0$ is obtained as the intersection point of approximants to the curve/curve bisectors of any two pairs of these segments — $\boldsymbol{q}(t), \boldsymbol{r}(u)$ and $\boldsymbol{q}(t), \boldsymbol{s}(v)$ say. Now, by definition, $\boldsymbol{b}_*$ has exactly one interior footpoint on each of the curves $\boldsymbol{q}(t), \boldsymbol{r}(u), \boldsymbol{s}(v)$, and all points in some neighborhood $\mathcal{N}(\boldsymbol{b}_*)$ of it must also have this property.[16] We assume the approximations to the curve/curve bisectors of $\boldsymbol{q}(t), \boldsymbol{r}(u)$ and $\boldsymbol{q}(t), \boldsymbol{s}(v)$ are "sufficiently close" to the *exact* bisectors that $\boldsymbol{b}_0 \in \mathcal{N}(\boldsymbol{b}_*)$.

[14] We suppose that $\boldsymbol{r}(0)$ is the common endpoint of s_{i-1} and s_i, while $\boldsymbol{r}(1)$ is the common endpoint of s_i and s_{i+1}.

[15] See Choi et al. [6] for an alternative approach to this bifurcation-point problem.

[16] For, if this were not true, $\boldsymbol{b}_*$ must have *two* footpoints on (at least) one of the curves $\boldsymbol{q}(t), \boldsymbol{r}(u), \boldsymbol{s}(v)$ — contradicting our supposition that it is a type (3d) bifurcation.

Now let t_*, u_*, v_* be the footpoint parameter values of $\boldsymbol{b}_* = (x_*, y_*)$ on the three curves.[17] The variables $\boldsymbol{x}_* = (x_*, y_*, t_*, u_*, v_*)^{\mathrm{T}}$ must then satisfy the system of equations

$$\begin{aligned}
F(\boldsymbol{x}) &= [x - X(t)]^2 + [y - Y(t)]^2 - [x - X(u)]^2 - [y - Y(u)]^2 = 0,\\
G(\boldsymbol{x}) &= [x - X(t)]^2 + [y - Y(t)]^2 - [x - X(v)]^2 - [y - Y(v)]^2 = 0,\\
Q(\boldsymbol{x}) &= [x - X(t)]\,X'(t) + [y - Y(t)]\,Y'(t) = 0,\\
R(\boldsymbol{x}) &= [x - X(u)]\,X'(u) + [y - Y(u)]\,Y'(u) = 0,\\
S(\boldsymbol{x}) &= [x - X(v)]\,X'(v) + [y - Y(v)]\,Y'(v) = 0,
\end{aligned} \tag{11}$$

or, in vector form, $\boldsymbol{F}(\boldsymbol{x}) = (F(\boldsymbol{x}), G(\boldsymbol{x}), Q(\boldsymbol{x}), R(\boldsymbol{x}), S(\boldsymbol{x}))^{\mathrm{T}} = \boldsymbol{0}$. Since each component of $\boldsymbol{F}$ is differentiable with respect to each component of $\boldsymbol{x}$, we may expand it in a Taylor series about $\boldsymbol{x}_i$ and write

$$\boldsymbol{F}(\boldsymbol{x}_{i+1}) = \boldsymbol{F}(\boldsymbol{x}_i) + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_i} \cdot \Delta \boldsymbol{x}_i + \cdots, \tag{12}$$

where $\Delta \boldsymbol{x}_i = (x_{i+1} - x_i, y_{i+1} - y_i, t_{i+1} - t_i, u_{i+1} - u_i, v_{i+1} - v_i)$. Here $\partial \boldsymbol{F}/\partial \boldsymbol{x}$ denotes the 5×5 Jacobian matrix

$$\begin{bmatrix}
2X(u) - 2X(t) & 2Y(u) - 2Y(t) & -2Q & 2R & 0\\
2X(v) - 2X(t) & 2Y(v) - 2Y(t) & -2Q & 0 & 2S\\
X'(t) & Y'(t) & \partial Q/\partial t & 0 & 0\\
X'(u) & Y'(u) & 0 & \partial R/\partial u & 0\\
X'(v) & Y'(v) & 0 & 0 & \partial S/\partial v
\end{bmatrix} \tag{13}$$

for the system (11), where

$$\begin{aligned}
\partial Q/\partial t &= [x - X(t)]X''(t) + [y - Y(t)]Y''(t) - X'^2(t) - Y'^2(t),\\
\partial R/\partial u &= [x - X(u)]X''(u) + [y - Y(u)]Y''(u) - X'^2(u) - Y'^2(u),\\
\partial S/\partial v &= [x - X(v)]X''(v) + [y - Y(v)]Y''(v) - X'^2(v) - Y'^2(v).
\end{aligned} \tag{14}$$

Ignoring higher-order terms, we obtain from $\boldsymbol{x}_i$ a closer approximation $\boldsymbol{x}_{i+1}$ to the solution $\boldsymbol{x}_*$ of (11) by setting $\boldsymbol{F}(\boldsymbol{x}_{i+1}) = \boldsymbol{0}$ in (12). This yields a system of linear equations for the increments $\Delta \boldsymbol{x}_i$, which define the Newton–Raphson iteration for the exact bifurcation point and its footpoint parameter values. This commences with the initial approximation $\boldsymbol{x}_0 = (x_0, y_0, t_0, u_0, v_0)$, where $\boldsymbol{b}_0 = (x_0, y_0)$ has footpoint parameter values t_0, u_0, v_0.

Instead of inverting the 5×5 Jacobian matrix to compute $\Delta \boldsymbol{x}_i$, note that the last three equations[18] can be used to write $\Delta t_i, \Delta u_i, \Delta v_i$ in terms of $\Delta x_i, \Delta y_i$. These expressions can then be substituted into the first two equations to obtain a 2×2 system for $\Delta x_i, \Delta y_i$. Solving this system, and computing

[17] For brevity, we assume that $\boldsymbol{q}(t), \boldsymbol{r}(u), \boldsymbol{s}(v)$ are polynomial curves — the extension to rational curves is straightforward — and we employ the same symbols for their coordinate components (the indicated parameters identify the curves under consideration).

[18] Proposition 5.3 below shows that $\partial Q/\partial t, \partial R/\partial u, \partial S/\partial v$ are nonzero at $\boldsymbol{x}_i$.

the corresponding $\Delta t_i, \Delta u_i, \Delta v_i$ increments, gives the Newton–Raphson step $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \Delta \boldsymbol{x}_i$. A termination criterion can be based on the magnitudes of the individual components of $\Delta \boldsymbol{x}_i$, or a suitable combination of them.

We now verify convergence of the Newton–Raphson iterations by showing that the Jacobian matrix (13) is nonsingular in a neighborhood of $\boldsymbol{b}_*$.

Proposition 5.3. *Let $\boldsymbol{b}_* = (x_*, y_*)$ be a bifurcation in the Voronoi diagram of the boundary S of a planar domain D, with distinct interior footpoints on three different boundary segments $\boldsymbol{q}(t) = (X(t), Y(t))$, $\boldsymbol{r}(u) = (X(u), Y(u))$, $\boldsymbol{s}(v) = (X(v), Y(v))$. If t_*, u_*, v_* are the footpoint parameter values of $\boldsymbol{b}_*$ on these curves, the Jacobian* (13) *is non-singular at $\boldsymbol{x}_* = (x_*, y_*, t_*, u_*, v_*)$.*

Proof. Since $Q = R = S = 0$ at $\boldsymbol{x}_*$, the Jacobian (13) has determinant

$$4\{[X(u_*) - X(t_*)][Y(v_*) - Y(t_*)] - [X(v_*) - X(t_*)][Y(u_*) - Y(t_*)]\} \frac{\partial Q}{\partial t} \frac{\partial R}{\partial u} \frac{\partial S}{\partial v} \tag{15}$$

there, the partial derivatives (14) being evaluated at $\boldsymbol{x}_*$. Now the first term of (15), in parentheses, cannot vanish if t_*, u_*, v_* identify the footpoints of the bifurcation $\boldsymbol{b}_*$, since its vanishing implies that these footpoints are collinear, and it is impossible for $\boldsymbol{b}_*$ to be equidistant from three distinct points on a line. Thus, one of the partial derivatives (14) must vanish if (13) is singular.

Now if the first derivative in (15) vanishes at $\boldsymbol{x} = \boldsymbol{x}_*$, we may solve $Q = \partial Q / \partial t = 0$ as a system of linear equations for the coordinates (x_*, y_*) of $\boldsymbol{b}_*$. But this is the same system of linear equations as in Lemma 5.2. Thus, $\boldsymbol{b}_*$ must coincide with the center of curvature or evolute point $\boldsymbol{q}(t_*) - \boldsymbol{n}(t_*)/\kappa(t_*)$ of the curve $\boldsymbol{q}(t)$. Moreover, $\boldsymbol{b}_*$ has two identical footpoints on $\boldsymbol{q}(t)$ at $t = t_*$ which is not possible by hypothesis. Similarly, $\boldsymbol{b}_*$ must coincide with the center of curvature of the curve points $\boldsymbol{r}(u_*)$ or $\boldsymbol{s}(v_*)$ if the second or third derivative in (15) vanishes at $\boldsymbol{x}_*$ which is again impossible by hypothesis. □

By the inverse function theorem, $\boldsymbol{F}(\boldsymbol{x})$ is one-to-one and onto in some neighborhood of $\boldsymbol{x}_*$ if the Jacobian (13) is non-singular at $\boldsymbol{x}_*$, and $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ then has a unique solution in that neighborhood. Furthermore, since $\boldsymbol{F}(\boldsymbol{x})$ is C^∞, the Jacobian (13) must be non-singular in a neighborhood of $\boldsymbol{x}_*$. Thus, other than in exceptional circumstances, the Newton–Raphson scheme will converge (quadratically) to the exact bifurcation point. If nonconvergence is observed, however, we may appeal to a simple bisection method to refine the bifurcation point, similar to that used [15] to locate bifurcation points of type (2). The bisection method is slower but essentially foolproof. Once the bifurcation points have been refined, the Hermite approximants [15] to curve/curve bisectors terminating at these points are adjusted to incorporate their exact coordinates and tangent/curvature data.

Consider next the bifurcations of type (3c). Such points have two interior footpoints, on $\boldsymbol{q}(t)$ and $\boldsymbol{r}(u)$ say, and one terminal footpoint on $\boldsymbol{s}(v)$. If the latter has coordinates (x_0, y_0), we replace Eqs. (11) by the system

$$\begin{aligned}
F(\boldsymbol{x}) &= [x - X(t)]^2 + [y - Y(t)]^2 - [x - X(u)]^2 - [y - Y(u)]^2 = 0,\\
G(\boldsymbol{x}) &= [x - X(t)]^2 + [y - Y(t)]^2 - (x - x_0)^2 - (y - y_0)^2 = 0,\\
Q(\boldsymbol{x}) &= [x - X(t)]X'(t) + [y - Y(t)]Y'(t) = 0,\\
R(\boldsymbol{x}) &= [x - X(u)]X'(u) + [y - Y(u)]Y'(u) = 0.
\end{aligned}$$

A Newton–Raphson iteration scheme, analogous to that described above, can be developed from these equations for refinement of type (3c) bifurcations.

Finally, consider the case of a bifurcation $\boldsymbol{b}_*$ with *more than three* distinct footpoints on the domain boundary S. Such bifurcations are "exceptional", but do not require any special treatment — the methods discussed above suffice for computing their exact coordinates. To see this, suppose that $\boldsymbol{b}_*$ has distinct footpoints on four segments s_1, s_2, s_3, s_4 of S. Now when three of these boundary segments, s_1, s_2, s_3 say, have been included in the boundary segment set, $\boldsymbol{b}_*$ will be identified as a "generic" bifurcation point of type (3). At this stage, depending on the type of $\boldsymbol{b}_*$, the appropriate algorithm discussed in this section may be employed to locate the exact coordinates of $\boldsymbol{b}_*$. The subsequent inclusion of s_4 does not alter the coordinates of $\boldsymbol{b}_*$ in any way. Thus, due to our strategy of including one boundary segment of S at a time, and because of the fact that each boundary segment is "simple", we are guaranteed to capture all exceptional bifurcation points having more than three disticnt footpoints on the domain boundary.

6. Medial axes

We now consider the construction of the medial axis of a planar domain D from the Voronoi diagram of the domain boundary S. The medial axis may be formally defined as follows [6, 32]:

Definition 6.1. The *medial axis* of a planar domain D with boundary S is the closure of the set of points in D that have (at least) *two distinct footpoints on* S — i.e., it is the closure[19] of the point set

$$\{\boldsymbol{q} \in D \mid \exists\, \boldsymbol{q}_1, \boldsymbol{q}_2 \in S \text{ such that } \operatorname{dist}(\boldsymbol{q}, S) = |\boldsymbol{q} - \boldsymbol{q}_1| = |\boldsymbol{q} - \boldsymbol{q}_2| \text{ with } \boldsymbol{q}_1 \neq \boldsymbol{q}_2\}. \tag{16}$$

We shall denote the medial axis of S by $\mathrm{MA}(S)$. Note here that $\operatorname{dist}(\boldsymbol{q}, S)$ refers to the distance of $\boldsymbol{q}$ from the *entire* boundary S — and not just a particular boundary segment; see [32] for further details.

A *maximal circle* is associated with each point $\boldsymbol{q}$ of the medial axis — this circle is contained within D and touches its boundary S at the footpoints of $\boldsymbol{q}$ on S. If each edge of $\mathrm{MA}(S)$ admits a parametrization $(x(t), y(t))$, we can superpose the corresponding *radius functions* $r(t)$ on these edges, describing the size of the maximal circles. The set of three-dimensional loci $(x(t), y(t), r(t))$ then define the *medial axis transform* (MAT) of the domain D; we can recover the boundary S from its MAT as the envelope of the one-parameter family of circles of radii $r(t)$ centered on each medial axis point $(x(t), y(t))$.

A domain boundary S described by "simple" (polynomial/rational) curves does not, in general, admit a "simple" closed-form parameterization for its MAT. Conversely, given an MAT of simple functional form, the corresponding boundary S cannot be exactly described by simple (polynomial/rational) curve segments. The introduction of Pythagorean-hodograph (PH) curves in the Minkowski metric by Moon [28] to define MAT elements $(x(t), y(t), r(t))$ is an important development in

[19] By taking the *closure* of the set (16) we include its limit points — at which two formerly distinct footpoints on S coalesce into one. Such points correspond to centers of curvature for the vertices (i.e., points of extremum curvature) on S.

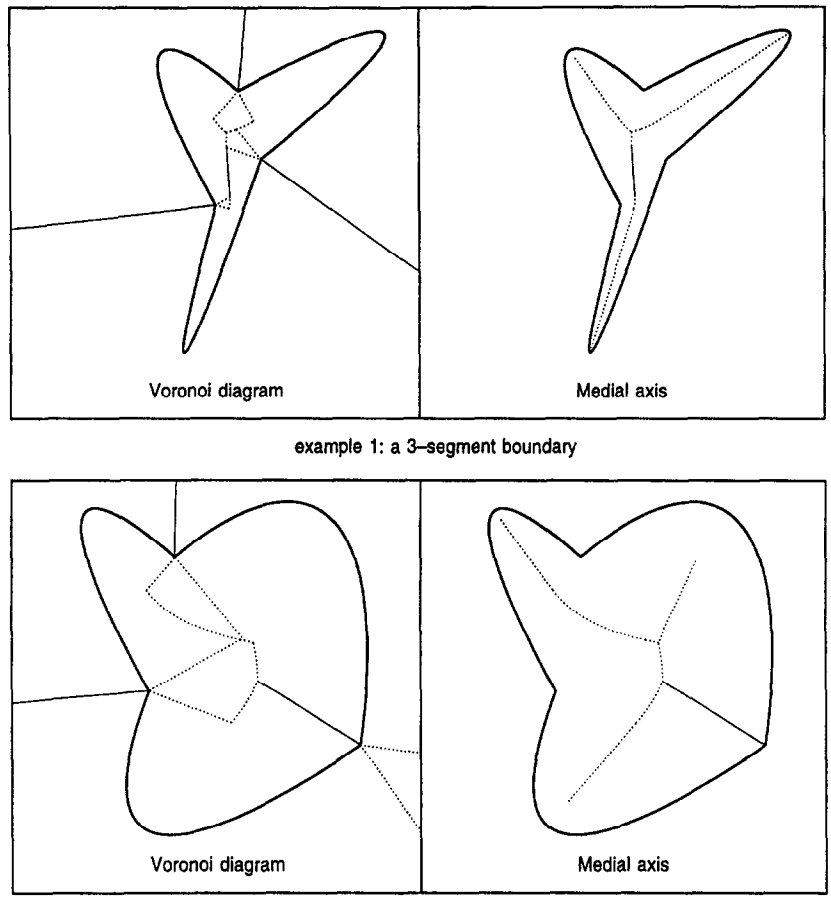


Fig. 1. Voronoi diagrams and medial axes for two 3-segment boundaries. Dotted loci indicate edges having *exact* representations, while solid loci are those that have been approximated to the prescribed tolerance. Note that, in both cases, all but one of the medial axis edges are represented exactly.

this respect. PH curves are characterized by the property [11, 17] that their *parametric speed* (the rate of change of arc length s with the curve parameter t) is a *polynomial* function of t, and in the Minkowski metric the arc-length element is given by $ds^2 = dx^2 + dy^2 - dr^2$ rather than the Euclidean form $dx^2 + dy^2 + dr^2$. An MAT whose elements are defined by Minkowski-metric PH curves admits the recovery of a domain boundary S that is *exactly* describable by rational curve segments.

It was shown in [32] that, although VD(S) and MA(S) typically have a number of edges in common, neither is (in general) a subset of the other. Our medial axis construction algorithm is based on this observation, and the following four properties [32] of VD(S) and MA(S):

(a) all segments of VD(S) lying outside S do not belong to MA(S);

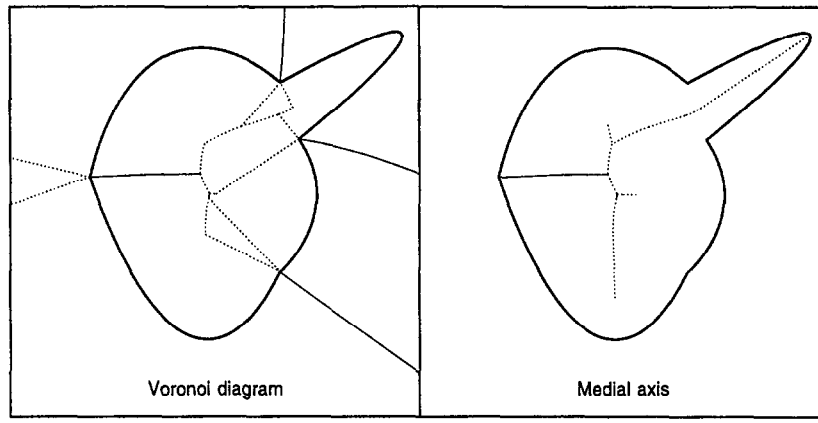
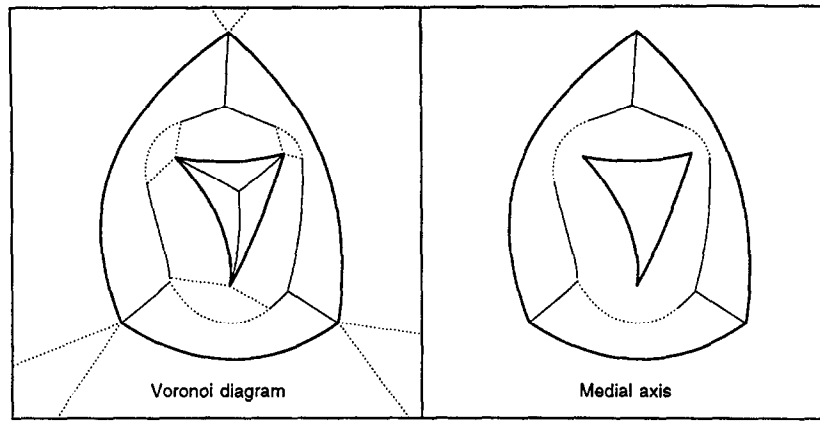


Fig. 2. Further examples of Voronoi diagrams and medial axes, for domains with 4- and 5-segment boundaries. Again, the exact Voronoi diagram and medial axis edges are indicated by dotted loci, while approximate edges are shown as solid loci.

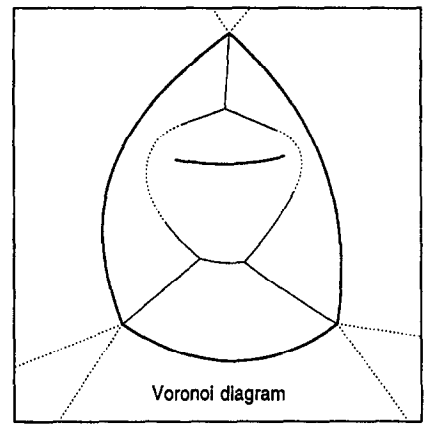
(b) any line segments[20] in VD(S) whose points have a unique footpoint on S do not, by Definition 6.1, belong to MA(S);
(c) all other segments of VD(S) do belong to MA(S);
(d) all edges of MA(S) whose points have *both* their footpoints on the *same* boundary segment do not, by Definition 2.1, belong to VD(S) — such edges are portions of interior *self-bisectors* of boundary segments.

Thus, to construct MA(S) from VD(S), we first remove all the segments of VD(S) lying outside S. Then, line segments in VD(S) whose points have a unique footpoint on S are removed. Finally, we construct the interior self-bisectors (if any) of each of the boundary segments s_i, using the algorithm

[20] Points of the rational and nonrational algebraic edges of the Voronoi diagram must have (at least) two distinct footpoints on the boundary. This is easily seen by noting that such edges are point/curve and curve/curve bisectors, respectively.



example 5: a 6-segment boundary (multiply-connected domain)



example 6: a collection of 4 planar segments

Fig. 3. Example 5 shows the ability of the algorithm to compute Voronoi diagrams/medial axes for *multiply connected* domains (here, a 6-segment boundary with a single hole). Example 6 shows that the algorithm can also compute Voronoi diagrams for arbitrary sets of curve segments (which do not necessarily bound a domain — in such cases, the medial axis is undefined).

described in [16], and incorporate the portions of these self-bisectors that lie inside *both* $\mathrm{VP}(s_i)$ and S, for $i = 1, \ldots, N$.

The process of computing interior self-bisectors of individual boundary segments can be a rather complicated matter in its own right. Fortunately, for a certain class of boundary segments, called "simple" segments,[21] this construction is fairly straightforward [16]. Thus, to make the self-bisector calculations tractable, a preprocessing step in which boundary segments are split into "simple" segments is necessary. In [16] we have shown that conic segments are always simple, while (polynomial)

[21] A *simple* segment is defined such that each point of its self-bisector has (at most) two interior footpoints on it — i.e., its self-bisector cannot exhibit bifurcations.

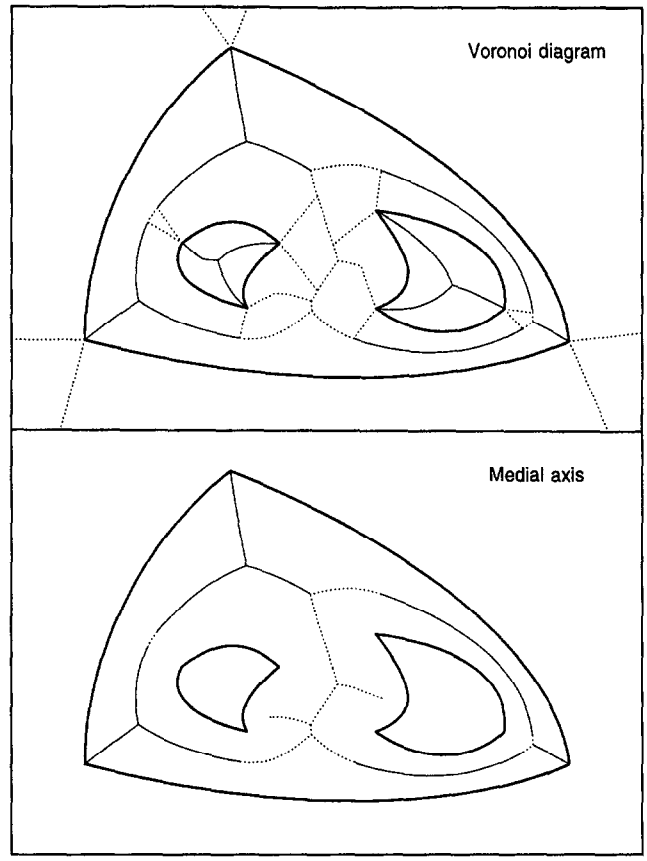


Fig. 4. Voronoi diagram/medial axis of a 9-segment boundary with two holes; note the intricate structure of *exact* Voronoi edges between these holes.

cubic segments can be easily split into at most three simple subsegments. Thus, in most cases of practical interest, this step poses no significant computational difficulty.

7. Computed examples

Figs. 1–5 show examples of Voronoi diagrams and medial axes computed by our algorithm. All the implementation issues discussed above, including the data structures for the various geometrical entities, computation of the Boolean operations, curve/curve intersection algorithms, and the refinement of type (3c) and (3d) bifurcation points were incorporated into the C code that generated these examples. In addition, robust implementations of the basic point/curve and curve/curve bisector algorithms [12, 13, 15, 16] — for both generic and degenerate cases — were required.

The boundary segments in these examples are all conics, and hence their self-bisectors can be represented exactly as portions of their symmetry axes. In all the illustrations, the Voronoi

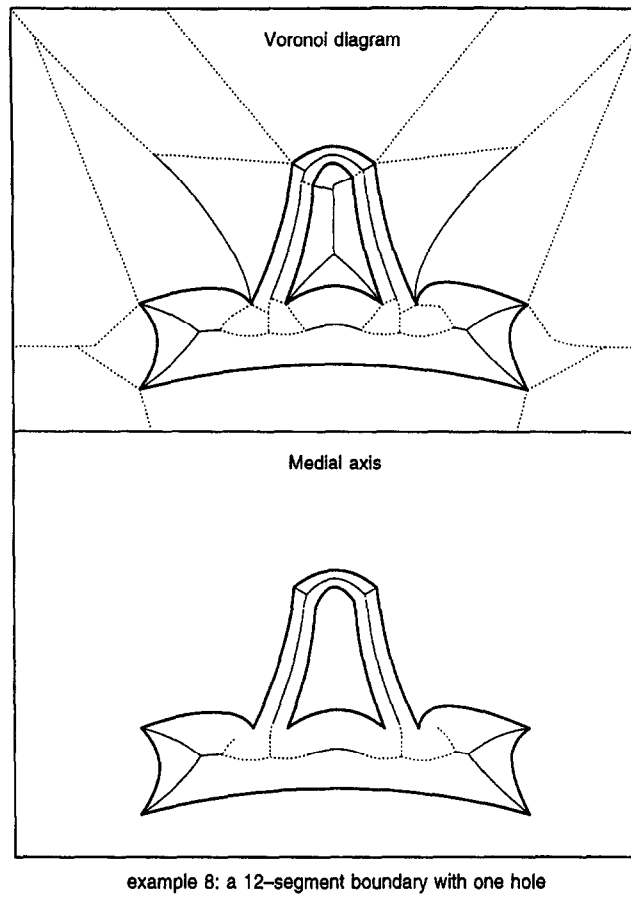


Fig. 5. Voronoi diagram and medial axis for a 12-segment boundary.

diagram/medial axis edges that have *exact* representations — namely, linear point/point bisectors, rational point/curve bisectors, and the self-bisectors of conic segments — are shown as dotted loci, while edges that have been approximated (namely, nonrational algebraic curve/curve bisectors) are shown as solid loci.

In the examples shown in Figs. 1–5, a geometrical tolerance of 10^{-3} was employed for the nonrational curve/curve bisector approximations, while the bifurcation-point coordinates were refined to machine precision (in each case the domain boundaries have dimensions of order unity). The run times ranged from a few cpu seconds for simple cases (examples 1 and 2) to 90 cpu seconds for example 8, on a Sun workstation. No special effort has been made to optimize the code. The algorithm obviously has $O(N^2)$ complexity in the number N of boundary segments, because of the sequential manner in which segments are introduced and the need to modify the Voronoi regions of all existing segments upon the introduction of each new one. We expect that more-efficient algorithms, employing a divide-and-conquer strategy, could be developed using the basic geometrical utilities described herein.

8. Closure

The Voronoi diagram/medial axis algorithm presented above applies to the boundaries of both simply and multiply connected domains since, during the merge process, no stipulation that the domain be simply connected was necessary. The algorithm has been implemented in C, and accepts domains bounded by linear, conic, and cubic segments, a user-specified geometrical tolerance, and the dimensions of the computational domain $\mathscr{A}$ as input. The output is a set of polynomial and rational Bézier curves that represent the Voronoi diagram/medial axis edges exactly wherever possible, and otherwise approximate them within the given tolerance. The results reported here, and in the companion paper [32], form the basis for the first author's Ph.D. thesis — see [31] for further details.

Acknowledgements

We gratefully acknowledge support of the National Science Foundation (grant CCR–9530741) and the Office of Naval Research (grant N00014-95-1-0767).

References

[1] H. Alt, O. Schwarzkopf, The Voronoi diagram of curved objects, Proc., 11th ACM Comput. Geometry Symp., Vancouver, BC, 1995, pp. 89–97.
[2] H. Blum, R.N. Nagel, Shape description using weighted symmetric axis features, Pattern Recog. 10 (1978) 167–180.
[3] F.L. Bookstein, The line-skeleton, Comput. Graph. Image Process 11 (1979) 123–137.
[4] J.W. Brandt, A.K. Jain, V.R. Algazi, Medial axis representation and encoding of scanned documents, J. Visual Commun. Image Representation 2 (1991) 151–165.
[5] H.I. Choi, S.W. Choi, C.Y. Han, H.P. Moon, K.H. Roh, N.S. Wee, New algorithm for offest of plane domain via MAT, in: Differential and Topological Techniques in Geometric Modeling and Processing '98, Bookplus Press, 1998, pp. 161–185.
[6] H.I. Choi, S.W. Choi, H.P. Moon, Mathematical theory of medial axis transform, Pacific J. Math. 181 (1997) 57–88.
[7] H.I. Choi, S.W. Choi, H.P. Moon, N.S. Wee, New algorithm for medial axis transform of plane domain, Graphical Models Image Process. 59 (1997) 463–483.
[8] J.J. Chou, Numerical control milling machine toolpath generation for regions bounded by free form curves. Ph.D. Thesis, University of Utah, 1989.
[9] J.J. Chou, Voronoi diagrams for planar shapes, IEEE Comput Graph. Appli. 15 (3) (1995) 52–59.
[10] G. Farin, Curves and Surfaces for Computer Aided Geometric Design, 3rd ed., Academic Press, Boston, 1993.
[11] R.T. Farouki, Pythagorean-hodograph curves in practical use, in: R.E. Barnhill (Ed.), Geometry Processing for Design and Manufacturing, SIAM, Philadelphia, 1992, pp. 3–33.
[12] R.T. Farouki, J.K. Johnstone, The bisector of a point and a plane parametric curve, Comput. Aided Geom. Design 11 (1994) 117–151.
[13] R.T. Farouki, J.K. Johnstone, Computing point/curve and curve/curve bisectors, in: R.B. Fisher (Ed.), Design and Application of Curves and Surfaces, The Mathematics of Surfaces, vol. V. Oxford University Press, Oxford, 1994, pp. 327–354.
[14] R.T. Farouki, V.T. Rajan, On the numerical condition of polynomials in Bernstein form, Comput. Aided Geom. Design 4 (1987) 191–216.
[15] R.T. Farouki, R. Ramamurthy, Specified-precision computation of curve/curve bisectors, Int. J. Comput. Geom. Appli. 8 (1998) 599–617.

[16] R.T. Farouki, R. Ramamurthy, Degenerate point/curve and curve/curve bisectors arising in medial axis computations for planar domains with curved boundaries, Comput. Aided Geom. Design 15 (1998) 615–635.
[17] R.T. Farouki, T. Sakkalis, Pythagorean hodographs, IBM J. Res. Develop. 34 (1990) 736–752.
[18] P.L. George, Automatic method (2): Voronoi type mesh generation, in: P.G. Ciarlet, J.L. Lions (Eds.), Handbook of Numerical Analysis, vol. IV. Finite Element Methods, Part 2, and Numerical Methods for Solids, Part 2, Elsevier, Amsterdam, 1996, pp. 149–187.
[19] L.M. Gross, Transfinite surface interpolation over Voronoi diagrams, Ph.D. Thesis, Arizona State University, 1995.
[20] H.N. Gursoy, N.M. Patrikalakis, An automatic coarse and fine surface mesh generation scheme based on the medial axis transform, I. algorithms. Eng. Computers 8 (1992) 121–137.
[21] M. Held, On the Computational Geometry of Pocket Machining, Springer, Berlin, 1991.
[22] D. Hilbert, S. Cohn-Vossen, Geometry and the Imagination, Chelsea, New York, 1952.
[23] D.-S. Kim, I.-K. Hwang, B.-J. Park, Representing the Voronoi diagram of a simple polygon using rational quadratic Bézier curves, Comput. Aided Design 27 (1995) 605–614.
[24] P.A. Koparkar, S.P. Mudur, A new class of algorithms for the processing of parametric curves, Comput. Aided Design 15 (1983) 41–45.
[25] J.M. Lane, R.F. Riesenfeld, Bounds on a polynomial, BIT 21 (1981) 112–117.
[26] D.T. Lee, Medial axis transformation of a planar shape, IEEE Trans. Pattern Anal. Machine Intell. PAMI-4 (1982) 363–369.
[27] S.N. Meshkat, C.M. Sakkas, Voronoi diagram for multiply-connected polygonal domains II: implementation and application, IBM J. Res. Develop. 31 (1987) 373–381.
[28] H.P. Moon, Computing rational offsets via medial axis transform using polynomial speed curves in $\boldsymbol{R}^{2,1}$, in: Differential and Topological Techniques in Geometric Modeling and Processing '98, Bookplus Press, 1998, pp. 187–203.
[29] H. Persson, NC machining of arbitrarily shaped pockets, Comput. Aided Design 10 (1978) 169–174.
[30] F.P. Preparata, The medial axis of a simple polygon, Proc. 6th Symp. Math. Foundations of Comput. Sci., 1977, pp. 443–450.
[31] R. Ramamurthy, Voronoi diagrams and medial axes of planar domains with curved boundaries, Ph.D. Thesis, University of Michigan, 1998.
[32] R. Ramamurthy, R.T. Farouki, Voronoi diagram and medial axis algorithm for planar domains with curved boundaries I. Theoretical foundations, J. Comput. Appl. Math. 102 (1999) 119–141.
[33] R. Ramamurthy, R.T. Farouki, Topological and computational issues in Voronoi diagram and medial axis constructions for planar domains with curved boundaries, in: Differential and Topological Techniques in Geometric Modeling and Processing '98, Bookplus Press, 1998, pp. 1–26.
[34] T.W. Sederberg, S.R. Parry, Comparison of three curve intersection algorithms, Comput. Aided Design 18 (1986) 58–64.
[35] V. Srinivasan, L.R. Nackman, Voronoi diagram for multiply-connected polygonal domains I: Algorithm, IBM J. Res. Develop. 31 (1987) 361–372.
[36] V. Srinivasan, L.R. Nackman, J.M. Tang, S.N. Meshkat, Automatic mesh generation using the symmetric axis transform of polygonal domains, IEEE Proc. 80 (1992) 1485–1501.
[37] T.K.H. Tam, C.G. Armstrong, 2D finite element mesh generation by medial axis subdivision, Adv. Eng. Software 13 (1991) 313–324.
[38] C.-K. Yap, An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments, Discrete Comp. Geom. 2 (1987) 365–393.