Eurographics Symposium on Geometry Processing 2016
Maks Ovsjanikov and Daniele Panozzo
(Guest Editors)

Volume 35 (2016), Number 5

Planar Minimization Diagrams via Subdivision with Applications to Anisotropic Voronoi Diagrams

H. Bennett[1], E. Papadopoulou[2†] and C. Yap[1‡]

[1]Courant Institute, NYU, New York, USA
[2]Faculty of Informatics, USI, Lugano, Switzerland

Abstract

Let $X = \{f_1, \ldots, f_n\}$ be a set of scalar functions of the form $f_i : \mathbb{R}^2 \to \mathbb{R}$ which satisfy some natural properties. We describe a subdivision algorithm for computing a clustered ε-isotopic approximation of the minimization diagram of X. By exploiting soft predicates and clustering of Voronoi vertices, our algorithm is the first that can handle arbitrary degeneracies in X, and allow scalar functions which are piecewise smooth, and not necessarily semi-algebraic.

We apply these ideas to the computation of anisotropic Voronoi diagram of polygonal sets; this is a natural generalization of anisotropic Voronoi diagrams of point sites, which extends multiplicatively weighted Voronoi diagrams. We implement a prototype of our anisotropic algorithm and provide experimental results.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms; F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—Geometrical problems and computations; G.1.0 [Numerical Analysis]: General—Interval arithmetic; G.1.2 [Numerical Analysis]: Approximation—Approximation of surfaces and contours; G.4 [Mathematical Software]: —Algorithm design and analysis.

1. Introduction

Voronoi diagrams have been extensively studied in Computational Geometry [AK00, AKL13, BWY06] and find applications in many areas [OBSC00]. As Voronoi diagrams can be defined in many different ways, let us informally indicate the kind that concerns us. Given a set $\mathcal{S} = \{S_1, \ldots, S_n\}$ of **sites**, the Voronoi diagram of $\mathcal{S}$ is a partition of an ambient space into Voronoi regions. Each region belongs to a site S_i, comprising those points that are "nearest" to S_i. In our setting the ambient space is $\mathbb{R}^2$ and each site S_i is a closed subset of $\mathbb{R}^2$ associated with a norm $\|\cdot\|_{S_i}$.

The Voronoi diagrams in Figures 1–2 are defined by a set $\mathcal{S}$ of three polygons (triangle, square, pentagon) with various norms specified by multiplicative weights. If their weights are $(1,1,1)$ then this corresponds to their usual Euclidean Voronoi diagram shown in Figure 1(a). By increasing the weight of the triangle to 2 in Figure 1(b) or to 4 in Figure 2(a), we see its Voronoi region growing and those corresponding to other sites shrinking. The diagram itself is computed by a subdivision process (using quadtrees) to any desired geometric precision $\varepsilon > 0$. By choosing smaller ε, we can obtain an arbitrarily more accurate representation as in Figure 2(b). However, our algorithm guarantees the correct topology, regardless of ε. Interestingly, weighted Voronoi diagrams of polygons generate all, and only, conic curves (see [Yap87] for another such class).

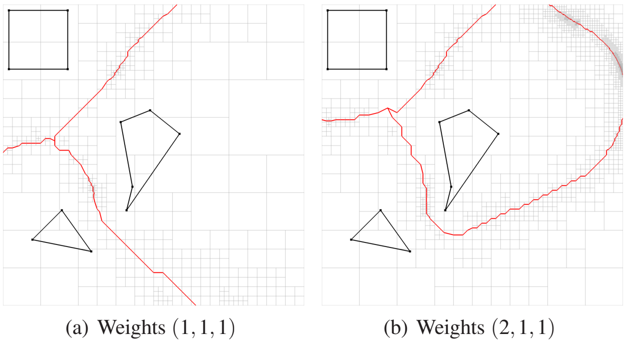
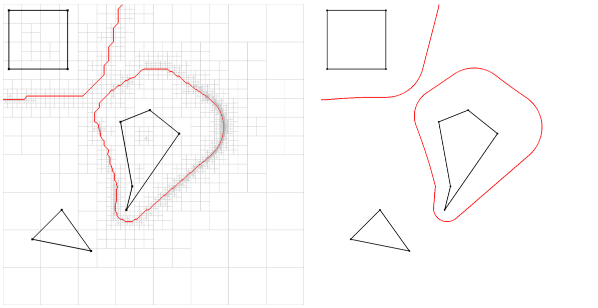

(a) Weights $(1,1,1)$ (b) Weights $(2,1,1)$

Figure 1: *Voronoi diagram of three polygons: unweighted and weighted*

More precisely, our goal is to compute *an ε-isotopic approximation of* Vor$(\mathcal{S})$ *restricted to a given box region* $B_0 \subseteq \mathbb{R}^2$. The input to our algorithm is $(\mathcal{S}, \varepsilon, B_0)$, and the output is an embedded planar graph $G = (V,E)$. Here, Vor$(\mathcal{S})$ is a collection of pairwise

[†] Evanthia is supported by SNSF #20GG21-134355, #200021E-154387.
[‡] Chee and Huck are supported by NSF Grant #CCF-1423228.



(a) Weights $(4,1,1)$ (b) Computed with smaller ε

Figure 2: *Voronoi diagram of three polygons with Weights* $(4,1,1)$

disjoint **cells** (subsets of $\mathbb{R}^2$) of dimensions $0,1,2$ called **Voronoi vertices**, **Voronoi curves** and **Voronoi regions** (respectively). It is necessary to slightly extend the standard algebraic topology notion of a cell complex [Mun84]: our cells are connected subsets of $\mathbb{R}^2$ which are homeomorphic to a finite union of open Euclidean balls. Our Voronoi vertex is a singleton as usual, but our Voronoi curve can be a finite or infinite curve or a closed loop; our Voronoi region need not be simply-connected (i.e., it may have one or more holes). Such non-standard cells are illustrated in Figure 2. In the planar setting, what is usually called the "Voronoi diagram" of $\mathcal{S}$ is just the subset of $\mathrm{Vor}(\mathcal{S})$ in which the Voronoi regions are omitted. Write $\mathrm{Vor}_1(\mathcal{S})$ for this subset; in Figures 1–2, this set is indicated in red. The ability to freely specify different norms with each site is a useful extension of weighted Voronoi diagrams.

This paper provides an algorithm to compute approximations of these Voronoi diagrams that are topological correct (up to isotopy) and geometrically accurate (up to ε in Hausdorff distance). Notably, our algorithm is able to handle arbitrary degeneracy through the concept of ε-clusters of Voronoi vertices. In fact, as we shall immediately proceed to greatly generalize the setting of our algorithm: sites are replaced by scalar functions, and Voronoi diagram by minimization diagrams.

1.1. Minimization Diagrams.

The above Voronoi diagrams can be generalized to the notion of "minimization diagrams" defined for any set X of scalar functions. Each $f \in X$ is a Lipshitz continuous function of the form $f : \mathrm{dom}(f) \to \mathbb{R}$, with $\mathrm{dom}(f) \subseteq \mathbb{R}^2$. For example, we associate a scalar function, called a **separation function** to each site $S_i \in \mathcal{S}$ as follows:

$$\mathrm{Sep}_{S_i} : \mathrm{dom}(S_i) \to \mathbb{R}$$

where $\mathrm{Sep}_{S_i}(p) := \inf\{\|p-q\|_{S_i} : q \in S_i\}$ and $\mathrm{dom}(S_i) := \mathbb{R}^2 \setminus S_i$ (the complement of S_i). Minimization diagrams were introduced by Edelsbrunner and Seidel [ES86], and their computation has been addressed by various authors (see Emiris, Mantzaflaris and Mourrain [EMM13] and references therein).

For any subset $Y \subseteq X$, the **domain** of Y is $\mathrm{dom}(Y) := \bigcap\{\mathrm{dom}(f) : f \in Y\}$, and its **Voronoi variety** is

$$Vvar(Y) := \Big\{ p \in \mathrm{dom}(Y) : (\forall f,g \in Y)\Big[f(p) = g(p) \Big] \Big\}. \quad (1)$$

When $Y = \{f,g\}$, $f \neq g$, we write $Vvar(f,g)$ for $Vvar(Y)$, calling it the **bisector** of f and g. Relative to X, the **Voronoi semi-variety** of Y is

$$Vvar(Y;X) := \Big\{ p \in Vvar(X) : (\forall f \in Y)(\forall g \in X \setminus Y)\Big[f(p) < g(p) \Big] \Big\}. \quad (2)$$

Each connected component of a non-empty semi-variety $Vvar(Y;X)$ is called a **Voronoi cell** of X (defined by Y). The **Voronoi complex** of X, denoted $\mathrm{Vor}(X)$, is defined as the partition of $\mathrm{dom}(X)$ into Voronoi cells of X. We may interchangeably call $\mathrm{Vor}(X)$ the **minimization complex** of X. Our original Voronoi complex of $\mathcal{S}$ can now be defined as $\mathrm{Vor}(X)$ where X comprises the separation function Sep_{S_i} for each $S_i \in \mathcal{S}$.

There are two issues with arbitrary minimization diagrams: (a) We have no assurance that for a general X, the concept of $\mathrm{Vor}(X)$ has nice geometric properties that we have come to expect from standard examples. (b) It is unclear how to compute or approximate $\mathrm{Vor}(X)$. One solution to these two issues is provided by Klein's [Kle89] concept of **abstract Voronoi diagrams** (AVD). He introduces axioms that control the interaction of any pair of bisectors (they intersect in a finite number of connected components) and ensure that the Voronoi regions of $\mathrm{Vor}(Y)$ is (path-)connected for all $Y \subseteq X$. Unfortunately, these axioms exclude many interesting examples such as the Voronoi diagrams in Figure 2. A more serious issue is the (implicit) computational model for AVD: it assumes a Real RAM computational model which has the capability to determine the (exact) intersection points of two bisectors; if there are multiple intersection points on a bisector, we can sort them along the bisector; if multiple bisectors have a common intersection v, we can circularly sort the bisectors around v, etc. Such capabilities are known to be computable if the bisectors are algebraic curves, but their exact implementation is expensive and rarely done. For non-algebraic functions, it is not even clear that these capabilities are Turing-computable (see [CCK*06]).

1.2. Computational Model: Subdivision and Soft Predicates.

In this paper, we initiate an alternative approach to computing minimization diagrams. The starting point is a numerical computational model based on interval methods (see e.g., [LY11a]). We basically need the ability to compute interval analogues of the scalar functions in X to any desired precision. Not only algebraic functions, but most common analytic functions fall within our scope.

We next formulate our algorithms using the well-known Subdivision Paradigm. For subdivision in the plane, the most common data structure is the quadtree [Sam90]. Subdivision algorithms are typically controlled by predicates on boxes $B \subseteq \mathbb{R}^2$. For instance, we may recursively split a B until a box predicate $P(B)$ holds. The (exact) predicate P is hard to implement, so we replace it by a **soft predicate** $\widetilde{P}$ [WCY15]: if $\widetilde{P}(B)$ holds, then $P(B)$ holds; but failure of $\widetilde{P}$ does not imply the negation of P. To compensate for this "one-sided" nature of $\widetilde{P}$, we assume that $\widetilde{P}$ is convergent – intuitively, this means that when B is small enough and $P(B)$ holds, then $\widetilde{P}(B)$ also holds. The first example of algorithms for Voronoi diagram using soft predicates is given by [YSL12].

Following the abstract Voronoi diagram approach of Klein, we also introduce axioms to ensure the nice behavior of the minimization diagrams. However it is seen that ours has wider applicability. To compute $\mathrm{Vor}(X)$, we require X to be **simple** in the following sense: for any triple of distinct functions $f, g, h \in X$:

(S1) The bisector $Vvar(f,g)$ is a 1-dimensional set.
(S2) The variety $Vvar(f,g,h)$ is 0-dimensional (i.e., is a finite set).

Axiom (S1) admits bisectors with more than one connected component, provided each component is either an infinite curve or a closed loop. In contrast, Klein restricts $Vvar(f,g)$ to be connected, and excludes closed loops. Axiom (S2) amounts to saying the bisectors $Vvar(f,g)$ and $Vvar(g,h)$ intersect finitely often; in contrast Klein requires that every pair of bisectors $Vvar(f,g)$ and $Vvar(f',g')$ intersect in finitely many components.

The following proposition shows that our axioms (S1,S2) are satisfied by natural examples:

LEMMA 1 Let $\mathcal{S}$ be a set of distinct points and lines. For each point or line $S \in \mathcal{S}$, let $\mathrm{Sep}_S(p) = \inf\{\|p-q\| : q \in S\}$ where $\|\cdot\|$ is the Euclidean norm. Then the set $X = \{\mathrm{Sep}_S : S \in \mathcal{S}\}$ is simple.

Note that the lines in $\mathcal{S}$ may intersect and a point may lie on a line. However, any two points or two lines in X must be distinct. Unlike Klein, our (S2) is not concerned about the intersection of arbitrary pairs of bisectors. This is important: consider $\mathcal{S} = \{A(-2,0), B(-1,0), C(1,0), D(2,0)\}$ comprising of four points. Then $Vvar(A,D) \cap Vvar(B,C)$ is a line, not a finite set of points. Nevertheless, $\mathcal{S}$ is a valid input for our algorithm.

1.3. Clustered ε-Approximations.

In this paper, we will generalize the notion of "ε-approximate Voronoi diagram" found in [YSL12]. Let us make this precise. First, recall that an **ε-approximation** for input (X, ε, B_0) is an embedded planar multigraph $G = (V,E)$ that is **ε-isotopic to** $\mathrm{Vor}_1(X) \cap B_0$. Here, $G = (V,E)$ is an embedded planar multigraph in the sense that $V \subseteq \mathbb{R}^2$ and edges in E are pairwise disjoint planar curves which are either closed loops or whose endpoints are in V. It is a multigraph (not just a graph) because there may be more than one curve having the same pair of vertices as its endpoints. Clearly, $\mathrm{Vor}_1(X)$ can similarly be viewed as an embedded planar multigraph. But we write "$\mathrm{Vor}_1(X) \cap B_0$" to denote the restriction of $\mathrm{Vor}_1(X)$ to B_0. More precisely, each cell in $\mathrm{Vor}_1(X) \cap B_0$ is a non-empty connected component of $c \cap int(B_0)$ or of $c \cap \partial(B_0)$ where $c \in \mathrm{Vor}_1(X)$ is a cell and $int(B_0)$ is the interior of B_0. We define G to be **ε-isomorphic** to another embedded planar multigraph $G' = (V', E')$ if that there are two bijections $\nu : V \to V'$, $\mu : E \to E'$ such that $\|u - \nu(u)\| \le \varepsilon$ and $d_H(e, \mu(e)) \le \varepsilon$ for all $u \in V, e \in E$ and d_H is the Hausdorff metric. But G is **ε-isotopic** to $G' = (V', E')$ is a stronger requirement, and it means that there is a continuous map $I : [0,1] \times \mathbb{R}^2 \to \mathbb{R}^2$ (called an **isotopy**) such that (by writing $I_t(x,y)$ for $I(t,x,y)$), we have I_0 is the identity map, $I_1(G) = G'$ (in the obvious sense), and for all t, $I_t(G)$ is a ε-isomorphic to G.

(A) For simplicity, we assume that Voronoi curves intersect ∂B_0 transversally and there are no Voronoi vertices in ∂B_0.

Assumption (A) is not essential; see [BCGY12] and [BSS*16] for ways to remove Assumption (A) without giving up our use of soft (numerical) methods The idea is to compute an embedded planar multigraph $G^* = (V^*, E^*)$ such that there exists a simply-connected set B^* satisfying $(1-\varepsilon)B_0 \subseteq B^* \subseteq (1+\varepsilon)B_0$, and G^* is is ε-isotopic to $\mathrm{Vor}(X) \cap B^*$. But in this paper, we focus on generalizing the notion of ε-approximation in another direction: to take full advantage of the ε parameter, we want the ability to replace a set of ε-close Voronoi vertices by a single "super vertex" in V.

For any disc $\Delta(m,r) \subseteq \mathbb{R}^2$ centered at m of radius r, let $\Delta_X(m,r)$ denote the set of Voronoi vertices in $\mathrm{Vor}(X)$ contained in $\Delta(m,r)$. If $\Delta_X(m,r)$ is non-empty, we call it a **Voronoi cluster**. If $r < \varepsilon$, it is called an ε-cluster. Following [YSS13], we call $\Delta_X(m,r)$ a **natural cluster** if $\Delta_X(m,r) = \Delta_X(m,3r)$. Natural clusters have the property that any two such clusters are either disjoint or have an inclusion relation. Thus the set of natural clusters forms a "cluster tree" whose nodes are natural clusters and parent-child relation in the tree is based on set inclusion. The root of this tree is the cluster containing all Voronoi vertices; if $\mathrm{Vor}(X)$ has n Voronoi vertices, this tree has at most $2n-1$ nodes. The leaves of this tree are the singleton clusters.

Note that the concept of natural clusters is directly applicable to embedded planar multigraphs. To exploit clusters, we define what it means for $G' = (V', E')$ to be a **simplification** of $G = (V,E)$: it means that there is a pair

$$\nu : V \to V', \qquad \mu : E \to V' \cup E'$$

(called **simplification maps**) such that ν is onto, and $|\mu^{-1}(e')| = 1$ for all $e' \in E'$ and if $\mu(e) \in V'$ (i.e., curve e collapses to a vertex in V') then $\nu(u) = \mu(e)$ for any endpoint u of the curve e. Moreover, we call $G' = (V', E')$ a **ε-simplification** of $G = (V,E)$ if, in addition, (1) $d_H(u', \nu^{-1}(u')) \le \varepsilon$, (2) $\nu^{-1}(u')$ is a natural ε-cluster, and (3) $d_H(e', \mu^{-1}(e')) \le \varepsilon$ for all $u' \in V', e' \in E'$.

Finally G' is **clustered ε-isotopic** to $G = (V,E)$ if there is a continuous map $I : [0,1] \times \mathbb{R}^2 \to \mathbb{R}^2$ such that I_0 is the identity map, for all $t \in [0,1]$, $I_t(G)$ is a ε-simplification of G, and in particular, $I_1(G) = G'$.

Our algorithm computes an embedded planar graph $G_\varepsilon(X)$ that is clustered ε-isotopic to $\mathrm{Vor}_1(X) \cap B_0$. The introduction of $G_\varepsilon(X)$ is a key ingredient towards computing minimization diagrams of non-algebraic functions. Without this generalization, it would be impossible to provide soft (i.e., numerical) methods to approximate $\mathrm{Vor}_1(X)$ when X has degenerate Voronoi vertices. A Voronoi vertex v is degenerate when it is defined by a set of $k > 3$ sites. Soft methods cannot distinguish between a single degenerate Voronoi vertex defined by $k > 3$ sites and a cluster of $k-2$ non-degenerate Voronoi vertices defined by k sites. Our algorithm would have a halting problem if it does not have the freedom to output ε-clusters. This inherent limitation of soft methods is usually called the "Zero Problem" of exact computation [CCK*06].

Remark that if we set $\varepsilon = \infty$, it means that we have no concern for geometric accuracy (but topological correctness remains in force). For simplicity, we tend to assume $\varepsilon = \infty$ as default.

1.4. Anisotropic Voronoi Diagrams

The implementation work reported here provides a validation of our general Voronoi diagram algorithm. We implement an algorithm to compute Voronoi diagrams defined by polygonal sites equipped with anisotropic norms. Anisotropic norms are those whose unit balls are ellipses, and is discussed in detail in Section 5. The well-known multiplicatively weighted Voronoi diagrams of a point set is a special case. Labelle and Shewchuk [LS03] introduced anisotropic norms for point sites, but their main focus was the associated Delaunay triangulation. Our extension to polygonal sites will greatly increase applicability in areas such as robot motion planning.

It is important to realize that the separation functions Sep_S ($S \in \mathcal{S}$) need not be smooth. This is an important aspect of our approach. Recall that a scalar function $f : \mathrm{dom}(f) \to \mathbb{R}$ is smooth if all its partial derivatives are well-defined in its domain. We say f is **semi-smooth** if the domain of f can be subdivided into a minimal number of connected subdomains $D_1, \ldots, D_m \subseteq \mathbb{R}^2$ with piecewise smooth boundaries such that f_i (the restriction of f to the interior of D_i) is smooth. Call each f_i a **feature function** of f. If the D_i's are defined by polynomial inequalities, and f_i satisfies some polynomial equations, we call f **semi-algebraic**. Thus f is smooth (resp., algebraic) if $m = 1$. This is illustrated in Figure 3 where S is (i) a line segment and (ii) a triangle. Assuming $\|\cdot\|_S$ is the Euclidean norm, then Sep_S is a semi-algebraic function with (i) $m = 4$ and (ii) $m = 6$ feature functions. Thus these separation functions are not algebraic.

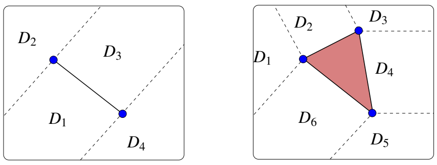


Figure 3: *The separation function* Sep_S *defined by (i) a line segment* ($m = 4$) *and (ii) a triangle* ($m = 6$)*. Assuming the Euclidean norm,* Sep_S *is algebraic in each of the subdomains* $D_1, \ldots, D_m$*.*

In implementation, it is useful to slightly shift our orientation and to replace each Sep_S by its smooth parts. Each polygonal site S can be replaced by the collection $\Phi(S)$ of its (boundary) **features**, i.e., $\Phi(S)$ is a partition of $\partial(S)$ into a finite set of corners (points) and edges (open line segments). E.g., if S is the triangle in Figure 3(ii), then $\Phi(S)$ has six such features. Each $t \in \Phi(S)$ has a "feature function" Sep_t with domain $\mathrm{dom}(t)$ such that $\mathrm{dom}(S) = \bigcup\{\mathrm{dom}(t) : t \in \Phi(S)\}$ and Sep_t is smooth in its domain. Note that $\mathrm{dom}(S)$ is the union, not intersection, of the $\mathrm{dom}(t)$'s. In [YSL12], we call $\mathrm{dom}(t)$ the **zone** of feature t. Finally, we can define **subcells** of $X^+ := \{\Phi(S) : S \in \mathcal{S}\}$ in such a way that each cell of $\mathrm{Vor}(X)$ is a union of subcells of X^+.

1.5. What is New

The main result of this paper is a general algorithm to compute a topologically correct (up to isotopy) and geometrically accurate (up to ε in Hausdorff distance) representation of the minimization diagram of a simple set X of semi-smooth function. Its computational primitives are explicitly formulated within a realistic model of computation, so the algorithm can be directly implementable. Two noteworthy features of this algorithm are: (1) it can handle arbitrary degeneracy in X by allowing ε-clusters of Voronoi vertices to be coalesced into super vertices, and (2) the scalar functions can be semi-smooth where the smooth parts admit interval approximations such as are commonly available with analytic functions.

The work closest to the present paper is [YSL12] where there is also a survey of subdivision algorithms for Voronoi diagrams. The basic approach of this paper first appeared there. They provided a subdivision algorithm for the standard Euclidean Voronoi diagram of a polygonal set, assuming non-degeneracy. Another closely related work is Emiris, Mantzaflaris and Mourrain [EMM13] who also use subdivision to compute minimization diagrams. They assume that the scalar functions are represented (explicitly or implicitly) by polynomials – this is essential for their use of Bernstein polynomials in exclusion test (what we call the C_0 predicate). They do not address non-algebraic scalar functions, nor semi-algebraic functions (e.g., sites which are not points). Degeneracy is not treated, and Voronoi vertices appears to be computed with exact operations.

A key challenge in our current setting is to design soft methods for detecting and constructing representations of Voronoi clusters within some box region. Some of these issues were addressed by [YSL12]; but we now face a more general problem arising from degeneracy. Our solution involves several non-trivial techniques, involving 5 Phases (see Section 4).

Degeneracy is always a severe challenge for soft methods. We overcame this barrier by introducing the notion of clustered ε-approximations $G_\varepsilon(\mathcal{S})$ which are indifferent in its representation of Voronoi clusters or degenerate vertices. This allows our algorithm to treats non-algebraic scalar functions; this appears to be the first such method.

Finally, our implementation of anisotropic Voronoi diagram, besides validating our approach, has independent interest for applications. Our `C++` implementation and datasets will be publicly distributed through our Core Library `http://cs.nyu.edu/exact/core_pages/`.

One limitation of our current implementation is that we use machine arithmetic. In theory, we could just replace this by a bigFloat number package to guarantee that we always compute the correct output. In practice, machine precision seems sufficient for the size of examples used in typical experimental work. We plan to perform error analysis to quantify the limitations of fixed precision computation in a future paper.

Another limitation is that our current algorithm only treat the case of limited degeneracy (Section 4). This is mainly due to space consideration. Limited degeneracy is already of interest, but we plan to describe the general solution in the full paper.

1.6. Overview of Paper

Following this introduction, Section 2 gives some background computational techniques that form the basis of our algorithm. Section

3 introduces the notion of root boxes: this is a box where one or more Voronoi vertices are located and "well-isolated". Section 4 provides the basic algorithm but it is correct provided we have a bound k_0 on the degree of degeneracy. Section 5 discusses the application of this algorithm to computing the Voronoi diagram of a polygonal set where each component is associated with its own anisotropic metric. We conclude in Section 6.

Appendix A provides an overview of three techniques: the Plantinga-Vegter (PV) construction of non-singular curves, the Moore-Kioustelidis (MK) test to detect singularities, and conformal subdivision to ensures correct connection between the PV curves and singularities. Appendix B contains proofs from the main algorithm. Appendix C contains proofs on anisotropic Voronoi diagrams.

2. Background

2.1. Box subdivisions and data structures.

We recall standard concepts related to quadtrees. Fix B_0 in the following. We maintain an implicit quadtree rooted in box B_0 whose nodes are subboxes of B_0 obtained by repeated splits, and whose leaves constitute a subdivision of B_0. All our boxes will be full-dimensional, squares and closed sets in $\mathbb{R}^2$. The split of a leaf B produces four congruent subboxes which become the children of B in the quadtree. Two boxes are k**-adjacent** if they intersect in a k-dimensional set ($k = -1, 0, 1, 2$), where $k = -1$ means they are disjoint. We are mainly interested in 1-adjacency, simply called "adjacent". Let $m_B = m(B)$ be the middle or center of B. The width $w_B = w(B)$ (resp., radius $r_B = r(B)$) of a box B is the length of one of its sides (resp., distance from $m(B)$ to a corner of B).

We write "$B \cap^* B'$" for the intersection of the interiors of B and B', i.e., $int(B) \cap int(B')$. We say that B and B' are **essentially disjoint** if $B \cap^* B' = \emptyset$. Alternatively, B and B' are k-adjacent for some $k < 2$. A set $\mathbb{S}$ of boxes is called a **subdivision** of the region $\bigcup \mathbb{S}$ if any pair of distinct boxes in $\mathbb{S}$ are essentially disjoint. E.g., the set of leaves of a quadtree forms a subdivision of the box at the root of the quadtree. Note that $\bigcup \mathbb{S}$ need not be connected or simply connected. A subdivision is **smooth** if any two adjacent boxes have width that differ by at most a factor of 2. Typically, a subdivision can be represented by the set of leaves of a quadtree. So a quadtree represents a smooth sudivision if the depths of any two adjacent leaves differ by at most one. In our root box construction, we will need subdivisions that are not obtained as the leaves of a single quadtree. Regardless, smoothness is essential for the correctness of the PV construction [PV04] (see Appendix A).

Given a subdivision $\mathbb{S}$, the operation $\mathtt{smooth}(\mathbb{S})$ returns a refinement of $\mathbb{S}$ into a smooth subdivision – this refinement is the unique minimal one. If B is a box in a smooth subdivision $\mathbb{S}$, the **smooth split** of B, denoted $\mathtt{sSplit}(B;\mathbb{S})$ will replace B by its four children in $\mathbb{S}$, and then apply $\mathtt{smooth}(\mathbb{S})$. Again, we return all the new boxes produced by splits. We write $\mathtt{sSplit}(B)$ for $\mathtt{sSplit}(B;\mathbb{S})$ when $\mathbb{S}$ is understood. More generally, if $C \subseteq \mathbb{S}$ is a set of boxes, we define $\mathtt{sSplit}(C;\mathbb{S})$ as the smooth split of each $B \in C$ in $\mathbb{S}$. Recently, it was shown that in any sequence of smooth split operations starting with $\mathbb{S} = \{B_0\}$, each operation has amortized constant cost [BY14]. Our data structure for $\mathbb{S}$ provides the ability to retrieve the set of boxes adjacent to any $B \in \mathbb{S}$. (see [BY14]).

By an **aligned box** we mean any box that can appear in some quadtree rooted in B_0. If B is an aligned box, $B \neq B_0$, then par(B) denotes the parent of B. If $\mathbb{S}$ is a set of boxes, aligned or not, let $\bigcup \mathbb{S}$ denote the union of the boxes in $\mathbb{S}$. For any box B and $k > 0$, let kB denote the box sharing the same center as B and scaled by factor k. Note that if B is aligned, then kB is non-aligned for $k \neq 1$.

We need a variety of priority queues such as Q_{root} and Q_k (for $k = 0, 1, 2, 3$). Each queue stores a set of boxes. For any queue Q, we have the two standard operations: $B \leftarrow Q.pop()$ returns the box B with highest priority (breaking ties arbitrarily); $Q.push(S)$ where S is a set of boxes and this operations puts all the boxes into Q. In addition, if we have a reference to a box B in Q, we can also directly remove it from Q via the operation $Q.remove(B)$. Usually, the priority has no effect on correctness (but might greatly affect performance).

2.2. Numerical Interval Methods.

We briefly review our numerical computational model based on interval methods [RR84]. Recall our goal is to compute the minimization diagram Vor(X) of a set X of scalar functions. The function $f : \mathrm{dom}(f) \to \mathbb{R}$ must be Lipshitz continuous and semi-smooth. We write $f_x(q) \uparrow$ if q is on the boundary of any D_i; otherwise $f_x(q) \downarrow$. Lipschitz continuity of f means that there is a constant $K_f > 0$ such that for all $p, q \in \mathbb{R}^2$, $\|f(p) - f(q)\| \leq K_f \|p - \|$. Unless otherwise noted, $\| \cdot \|$ refers to the Euclidean norm $\| \cdot \|_2$.

To use interval methods, f, f_x, f_y must have interval analogues. More precisely, let $\square\mathbb{R}^n$ denote the set of closed boxes (i.e., Cartesian products of intervals) in $\mathbb{R}^n$. The interval analogue of g ($g = f, f_x, f_y$) has the form $\square g : \square\mathbb{R}^n \to \square\mathbb{R}$. We call $\square f$ a **soft version** of f if it is conservative and convergent. **Conservative** means $f(B) \subseteq \square f(B)$ for $B \in \square\mathbb{R}^n$ where $f(B) := \{f(q) : q \in B, f(q) \downarrow\}$. **Convergent** means that if $B_1 \supseteq B_2 \supseteq \cdots$ is an infinite monotone decreasing sequence that converges to a point p, then $\square f(B_i) = f(p)$ for i large enough. There are many well-known ways to construct such box functions using interval arithmetic [RR84].

Define the **clearance function** of X to be $\mathrm{Clr}_X : \mathrm{dom}(X) \to \mathbb{R}$ where $\mathrm{Clr}_X(p) := \min\{f(p) : f \in X\}$. Let $\phi : \mathrm{dom}(X) \to 2^X$ be the set of closest feature functions, i.e., $\phi(p) := \{f \in X : f(p) = \mathrm{Clr}(p)\}$. We call $\phi(p)$ the **label set** of p.

Let Vor$_1(X)$ denote the subset of Vor(X) comprising the Voronoi vertices and curves, also known as the 1-skeleton of Vor(X). This is an embedded planar graph, and intuitively, our goal is to compute a isotopic ε-approximation of Vor$_1(X)$ restricted to some box $B_0 \subseteq \mathbb{R}^2$ (see [YSL12] exact definition).

We extend the notion of label set from points to boxes: $\phi(B) := \bigcup_{p \in B} \phi(p)$, called the **exact label set** of B. Instead of exact label sets, we compute a conservative approximation denoted $\widetilde{\phi}(B)$; call this the **active feature set** for B. But to introduce this, so we need some concepts. It is easy to verify that if K_f, K_g are Lipshitz constants for $f, g \in X$, the $K_f + K_g$ is a Lipshitz constant for the function $f - g$. If $Y \subseteq X$, we define

$K_2(Y) := \max\left\{K_f + K_g : f,g \in Y, f \neq g\right\}$. In the subdivision tree rooted at B_0, we will compute $\widetilde{\phi}(B)$ recursively using the following rule. Let $\widetilde{\phi}(B_0) := X$; recursively, for any box B with parent $\mathrm{par}(B)$, define

$$\widetilde{\phi}(B) := \left\{f \in \mathrm{par}(B) : f(m_B) \leq \mathrm{Clr}(m_B) + K_2(\widetilde{\phi}(\mathrm{par}(B)) r_B\right\}.$$

This generalizes a formulation from Milenkovic [Mil93, YSL12].

Lemma 2 $\widetilde{\phi}(B)$ is a soft version of $\phi(B)$:
(a) Conservative: $\phi(B) \subseteq \widetilde{\phi}(B)$
(b) Convergence: if $B_i \to p$ (for $i \to \infty$) converges monotonically to p then $\widetilde{\phi}(B_i) = \phi(p)$ for i large enough.

The computation of $\widetilde{\phi}(B)$ is very efficient since it is hereditary (by definition): $\widetilde{\phi}(B) \subseteq \widetilde{\phi}(\mathrm{par}(B))$. Furthermore, $\mathrm{Clr}(m_B)$ is efficiently computed via the formula $\mathrm{Clr}(m_B) = \min\left\{f(m_B) : f \in \widetilde{\phi}(\mathrm{par}(B))\right\}$.

2.3. Three Preliminary Techniques.

Our algorithm will build on three previous techniques. The first technique is the PV Construction from Plantinga-Vegter [PV04] to construct an isotopic approximation to a non-singular curve. The second technique from Moore-Kioustelidis [MK80] (see [LSVY14, Appendix]) detect **root boxes** which contain the transversal intersection of two curves. The third technique from [LSVY14] combines the first two techniques to create a network of curves that is isotopic to the Voronoi curves and vertices. An alternative approach to computing intersection is based on root bounds [BCGY12], but that appears to be less practical. See Section A in the appendix for a review of these techniques.

3. Root Boxes and Voronoi Clusters

We address the critical issue of our algorithm: how to detect natural clusters and provide a topologically correct construction around such clusters.

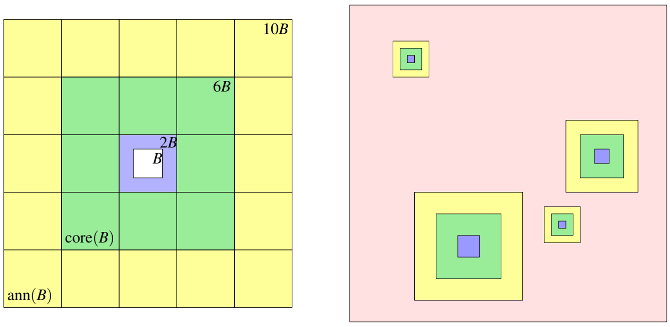
We focus on the concept of a "root box": informally, this is an aligned box B such that, among other things, has the property that it has at least 3 active features (i.e., $|\widetilde{\phi}(B)| \geq 3$) and for every triple $f,g,h \in \widetilde{\phi}(B)$, we have $Vvar(f,g,h) \cap 2B = Vvar(f,g,h) \cap 10B$ and $|Vvar(f,g,h) \cap 2B| = 1$. So $Vvar(f,g,h)$ has a unique Voronoi vertex in $2B$, and no other in $10B$. This root box would include the case where there is a single degenerate Voronoi vertex $v \in 2B$ defined by $Vvar(\widetilde{\phi}(B))$. With soft methods, we can now detect this as a root box, but we cannot distinguish this from two or more distinct vertices of total multiplicity k. Therefore we treat all the Voronoi vertices in $2B$ as a single (degenerate) Voronoi vertex. Formally, a **Voronoi cluster** is any set of Voronoi vertices that is contained in $2B$ for some root box B. Thus, our algorithm will output an approximation of such a "Voronoi cluster diagram".

An aligned box B is called a **root box** if

(R1) $\widetilde{\phi}(B)$ has at least 3 features.
(R2) $\widetilde{\phi}(B) = \widetilde{\phi}(10B)$.
(R3) $C_1(10B)$ holds: this means $C_1^{f,g}(10B)$ holds for any two $f,g \in \widetilde{\phi}(B)$.
(R4) $JC(12B)$ holds: this means $JC^{f,g,h}(12B)$ holds for any triple $f,g,h \in \widetilde{\phi}(B)$.
(R5) $MK(2B)$ holds: this means $MK^{f,g,h}(2B)$ holds for any triple $f,g,h \in \widetilde{\phi}(B)$.

We could add another requirement (R0) which asserts that the width of B must be smaller than some given $\varepsilon > 0$. This would ensure that our clusters are ε-small. We omit this for simplicity.

The significance of these conditions is as follows: (R1) ensures that there are enough active features in B to define at least a Voronoi vertex, while (R2) ensures that these features remain active throughout $10B$ and no new active features are involved. (R3) ensures that the bisectors defined by a pair of active features do not turn by more than 90 degrees (so are either x- or y-monotone). (R4) says that any triple of active features define at most one Voronoi vertex in $12B$. Finally (R5) ensures that every triple of active features (in isolation) actually define a Voronoi vertex in $2B$. Note that it does not mean that this Voronoi vertex will remain valid in the presence of other active features.



(a) Root box B

(b) Proper set of rootboxes

Figure 4: *Domain of root boxes*

Refer to Figure 4(a): for any root box B, the **domain** of B refers to the box $10B$. There is a subdivision of this domain into 25 subboxes, each congruent to $2B$ and semi-aligned. Of these 25 subboxes, let $\mathrm{core}(B)$ be the set of 9 subboxes that forms a subdivision of $6B$. The remaining 16 boxes (colored yellow in Figure 4) represent a subdivision of the annulus region around $6B$ and is denoted $\mathrm{ann}(B)$.

The listing of the root box conditions (R1-R5) is intended to determine the order of testing these conditions. If any of the conditions fail, we will abort the root box test. First of all, they are listed in order of increasing complexity in this sense: let $|\widetilde{\phi}(B)| = k$. To check (R1) and (R2), we need to compute $\widetilde{\phi}(B)$ and $\widetilde{\phi}(10B)$. This work is $O(|\widetilde{\phi}(B')|)$ and $O(|\widetilde{\phi}(10B')|)$ where B' is the parent of B. But in a certain amortized sense, it is $O(k)$. The complexity of checking (R3), i.e., computing $C_1(10B)$ is $O(k^2)$. Similarly, checking (R4) and (R5) requires computing $JC(12B)$ and $MK(2B)$, which is $O(k^3)$.

Next we observe that the properties (R2)-(R4) are **hereditary** in the sense that if they hold at box B, then they can be assumed to hold in any subbox of B. To exploit hereditary, we can associate a "sticky index" $i \geq 2$ with each box, indicating that property (Rj)

has been satisfied for all $j < i$. When we split a box, this index is inherited by its children.

3.1. Proper set of root boxes.

A set $\mathcal{R}$ of root boxes is said to be **proper** if

(P1) They are well-separated in the sense that for any two distinct boxes $B, B' \in \mathcal{R}$, the interiors of $12B$ and $10B'$ are disjoint. By symmetry, $10B$ and $12B'$ are essentially disjoint, so that there is a buffer between $10B$ and $10B'$ of width $\max\{w(2B), w(2B')\}$.

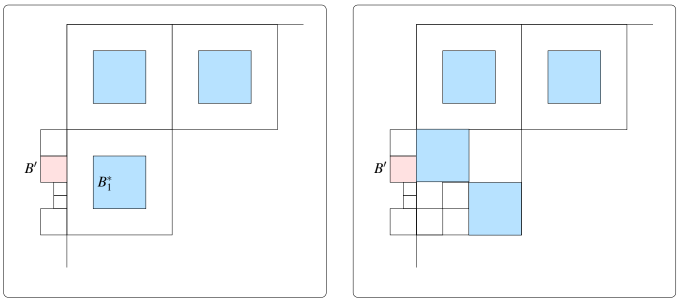
(P2) They are not close to the boundary of B_0 in the sense that $12B \subseteq B_0$.

Figure 4(b) shows a set $\mathcal{R}$ of four boxes that is proper. Each $B \in \mathcal{R}$ is represented by three concentric squares (blue $2B$, green $6B$, yellow $10B$).

Fix $\mathcal{R}$ to be a proper set of root boxes. The **complement** of $\mathcal{R}$ refers to the region

$$B_0 \setminus \bigcup_{B \in \mathcal{R}} 10B. \tag{3}$$

By property (P2), this is a region with $|\mathcal{R}|$ many holes. Suppose we know that there are no Voronoi vertices in this complement. Let $\mathbb{S}_1$ be any initial subdivision of B_0 (e.g., the subdivision obtained in the process of computing $\mathcal{R}$). It is easy to construct a unique subdivision $\mathbb{S}_2$ of the complement (3) such that the following holds: each box in $\mathbb{S}_1$ that is disjoint from $\bigcup_{B \in \mathcal{R}} 10B$ is also preserved in $\mathbb{S}_2$.



(i)

(ii)

Figure 5: *(i) $w(B') \leq w(B)/2$, (ii) $2B_1^*$ must split to be conformal*

Lemma 3 Let $B \in \mathcal{R}$ and $B' \in \mathbb{S}_2$ such that B' is adjacent to $10B$. Then B' is congruent to kB for some $k = 2^{-i}$ for some $i \geq 1$. Thus, $w(B') \leq w(B)/2$; see Figure 5(i).

We then refine the subdivision $\mathbb{S}_2$ into $\mathbb{S}_3$ as follows. First, we describe a general strategy: starting with $\mathbb{S}_2$, we keep subdividing (via smooth splits) each box $B \in \mathbb{S}_3$ until the following holds: for all $f, g \in \widetilde{\phi}(B)$, $C_0^{f,g}(B) \vee C_1^{f,g}(B)$ holds. This means that upon termination, we can do the PV construction on $\mathbb{S}_3$ for each bisector $Vvar(f,g)$ where $f, g \in X$. Moreover, we know that no two bisectors intersect since $\mathbb{S}_3$ is a subdivision of the complement of the root boxes. But there is a catch: in general, we only know about "bundles" of bisectors in each box B: we do not know their ordering relationship along the boundary of B. Of course, sometimes a bundle entering B may split into two or three bundles on exit, and this information can be maintained. But not all ordering information can be resolved this way until we treat the root boxes (below). In our main algorithm below, for simplicity, but at the expense of efficiency, we will enforce another condition: *each box in $\mathbb{S}_3$ has at most 2 active features* (i.e., Q_3 is empty). Thus we avoid the issue of bundles.

3.2. The Smooth Construction and Conformal Subdivision

By the "Smooth Construction" we mean the construction of the Voronoi diagram outside the domain of root boxes. Fundamentally, we apply the PV construction (Appendix A) to $\mathbb{S}_3$. The output is an embedded graph, which we may denote by $PV(\mathbb{S}_3)$.

In the PV construction, we need to know the boundary of the subdivision domain since we have special treatment for boundary boxes. Although the domain of Smooth Construction is basically $B_0 \setminus \bigcup_{B \in \mathcal{R}} 10B$, we must clarify that we consider a box B_1 to be a boundary box if it intersects ∂B_0; boxes that intersect $\partial(10B)$ ($B \in \mathcal{R}$) are *not* automatically considered as boundary boxes.

The reason is that a box B_1 that intersects $\partial(10B)$ ($B \in \mathcal{R}$) need not be considered a boundary box is that we will propagate smoothness requirement from B_1 into the interior of $10B$. This propagation turns out to be very well-behaved and local. Consider the subdivision $\text{ann}(B)$ comprising 16 boxes congruent to $2B$. For each $B_1 \in \text{ann}(B)$, consider the set $S(B_1) \subseteq \mathbb{S}_3$ of boxes that are adjacent to B_1. From Lemma 3, we know that $S(B_1)$ must have at least four boxes, all smaller than $0.5B_1$. We now compute the unique minimal smooth subdivision that is a refinement of $B_1 \cup S(B_1)$, call it $SS(B_1)$. This is illustrated in Figure 5(ii). Clearly, B_1 must be split in $SS(B_1)$. Among the four children of B_1, those children that are not adjacent to boxes in $S(B_1)$ need not be further split, i.e., will belong to $SS(B_1)$. There are two such children in general, but in case B_1 is one of the four corner boxes of $\text{core}(B)$, there is only one such child of B_1.

Let $\mathbb{S}_4$ be the union of the set set

$$SS' := \bigcup_{B \in \mathcal{R}} \left(\bigcup_{B' \in \text{ann}(B)} SS(B') \right)$$

and also the set

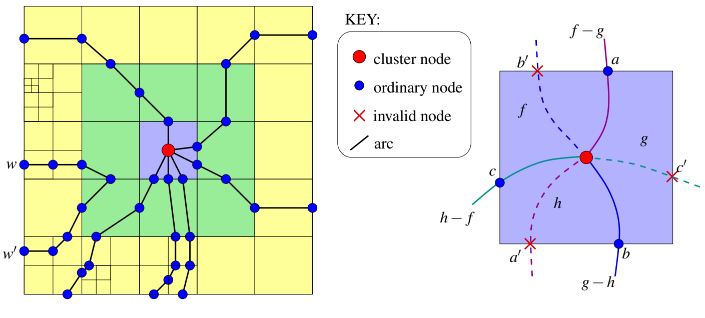
$$SS'' := \bigcup_{B \in \mathcal{R}} \text{core}(B').$$

Lemma 4
1. The set $\mathbb{S}_3 \cup \mathbb{S}_4$ is a smooth subdivision of $B_0 \setminus \bigcup Q_1$.
2. Moreover for each $B \in \mathcal{R}$, $\text{core}(B) \subseteq \mathbb{S}_4$. I.e., no box in $\text{core}(B)$ is split.

We say two subdivisions are **conformal** if their union is a smooth subdivision. Thus, lemma 4 says that the subdivisions $\mathbb{S}_3$ and $\mathbb{S}_4 = SS' \cup SS''$ are conformal. As reminder, conformality is needed because the correctness of PV construction depends on having smooth subdivision.

3.3. How to connect a cluster node

Given the smooth subdivision of $\mathbb{S}_4$ of lemma 4, we would like to construct an embedded graph $G(\mathbb{S}_4)$ representing our Voronoi diagram. This is amounts to constructing the graph $G(10B)$ for $B \in \mathcal{R}$. We split $G(10B)$ into two sub-tasks: to construct a graph $G(2B)$ and also $G(10B \setminus 2B)$.

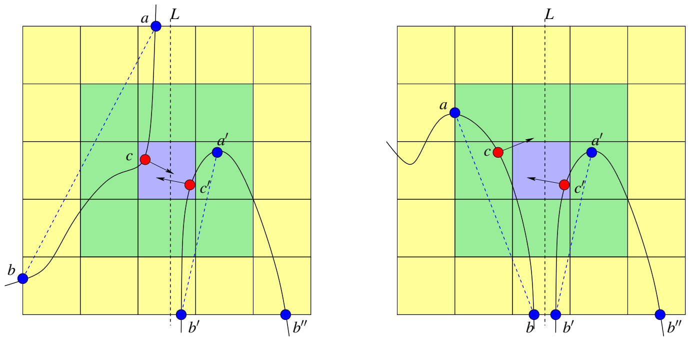


(a) Graph of root box domain

(b) Invalid nodes of $2B$

Figure 6: *(a) Smooth subdivision of* $10B$*, (b)* $Vvar(f,g,h)$ *has vertex in* $2B$

Before we address the sub-tasks, we will prove a lemma on the behavior of bisectors in the vicinity of a root box B. By definition of a root box, for each triple $f,g,h \in \widetilde{\phi}(B)$ of distinct features, we have $|Vvar(f,g,h) \cap 2B| = |Vvar(f,g,h) \cap 10B| = 1$. Let $V(B) \subseteq 2B$ denote the multiset of all such points. If $\widetilde{\phi}(B)$ has k features, then $|V(B)| = \binom{k}{3}$ (counted with multiplicity). All the Voronoi vertices in $10B$ is (already) contained in the set $V(B)$. For any bisector $Vvar(f,g)$ where $f,g \in \widetilde{\phi}(B)$, the set $Vvar(f,g) \cap 10B$ might have several connected components. Those components that intersect $2B$ are called (f,g)**-principal components** (or simply, **principal**).



(i)

(ii)

Figure 7: *Assuming two principal components (two possibilities)*

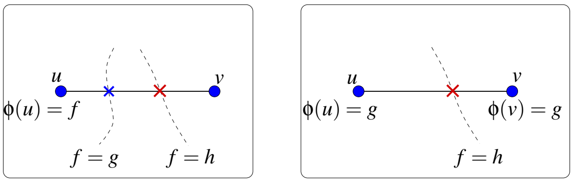
Lemma 5 Let $f,g \in \widetilde{\phi}(B)$.
(a) There is a unique (f,g)-principal component in $10B$.
(b) The Voronoi curve $Vvar(f,g;X)$ when restricted to the (f,g)-principal component is a connected (possibly empty) set.

We now address the first sub-task: Combinatorially, the graph $G(2B)$ is trivial: we introduce a single node at the center of $2B$ representing the Voronoi cluster. Call it a **cluster node**. See Figure 6(a). All the other kinds of nodes will be called **ordinary** to distinguish them from cluster nodes. If $k = |\widetilde{\phi}(B)|$, then we must introduce k nodes on the boundary $\partial(2B)$ and connect them to the cluster node by arcs. Note that if $k = 3$, the cluster is trivial. Figure 6(a) illustrates the case where $k = 8$. The main issue is where to place the nodes on $\partial(2B)$. We use a basic lemma in [LSVY14] which tells us how to order the nodes of bisectors along an edge of $2B$.

We need an additional tool: refer to Figure 6(b). Suppose the $Vvar(f,g,h)$ has a vertex in $2B$. The graph $PV(2B)$ will have three arcs, $[a,a']$, $[b,b']$ and $[c,c']$ corresponding to $f=g$, $g=h$ and $h=f$. For each of these arcs, we can invalidate one of their endpoints. In Figure 6(b), we show that a',b',c' are all invalid. This is based on the interactions of f,g,h alone; in the presence of other feature functions, a,b or c may also be invalid.

Consider a bisector $f=h$, where $f,h \in \widetilde{\phi}(2B)$, such that the graph $PV(2B)$ has a node n for $f=h$ on a side e of $2B$. We say that bisector $f=h$ is *invalid on* e, if there is a feature $g \in \widetilde{\phi}(2B)$ such that the portion of $f=h$ from the Voronoi vertex $Vvar(f,g,h)$ in $2B$ to e is contained in the Voronoi region $Vvar(g;\{f,g,h\})$. If bisector $f=h$ is invalid on e, because of feature g, we also say that node n is *invalidated by* g. A node that is not invalidated by any feature in $\widetilde{\phi}(2B)$ is called *valid*.

To construct $G(2B)$, we first identify the valid nodes on $\partial(2B)$, and then connect them to the cluster node in the center of $2B$. The ordering of the valid nodes along an edge e is implied by the labels of the endpoints of e that have been obtained from $\widetilde{\phi}(2B)$. Note that the first valid node on e must share the same feature as the label of the neighboring corner of e. Further, any two consecutive valid nodes must share a common feature in $\widetilde{\phi}(2B)$. Thus, the order of valid nodes is fixed. The following lemma gives a necessary and sufficient condition for a node on $\partial(2B)$ to be invalid.



(a)

(b)

Figure 8: *Invalidation of a* $f=h$ *node on an edge* uv

Lemma 6 Consider the graph $PV(2B)$ and a node of bisector $f=h$ on an edge $e=uv$ of the box $2B$, where $f,h \in \widetilde{\phi}(2B)$. See Figure 8. This node is invalidated by a feature $g \in \widetilde{\phi}(B)$ iff one of the following conditions holds:

(a) For an endpoint u of e, $\mathrm{Sep}_f(u) < \min(\mathrm{Sep}_g(u), \mathrm{Sep}_h(u))$ and bisectors $f=g$ and $f=h$ intersect e in this order as we move from u to v on e.

(b) For both endpoints of e, $\mathrm{Sep}_g(u) < \min(\mathrm{Sep}_f(u), \mathrm{Sep}_h(u))$ and $\mathrm{Sep}_g(v) < \min(\mathrm{Sep}_f(v), \mathrm{Sep}_h(v))$.

Note that the ordering of two bisectors on e can be determined as in [LSVY14].

3.4. Graph construction in the domain of root boxes

For a root box B, we let $(\mathbb{S}_4)_{10B}$ denote the set of boxes in the smooth subdivision $\mathbb{S}_4$, but restricted to the root box $10B$. Thus $(\mathbb{S}_4)_{10B}$ is a subdivision of the domain of B. Our goal is to do a graph construction using $(\mathbb{S}_4)_{10B}$. Figure 6(a) illustrates such a graph.

We know that $\mathrm{core}(B) \subseteq (\mathbb{S}_4)_{10B}$. In particular, $2B \in (\mathbb{S}_4)_{10B}$, and we had already constructed the graph $G(2B)$ above. Let v^* denote the cluster vertex in $G(2B)$.

Now consider the other boxes B' in $\mathrm{core}(B)$, $B' \neq 2B$. Suppose that $G(2B)$ has m nodes $u_1, u_2, \ldots, u_m$ ordered along the edge $B' \cap 2B$. For each u_i, there is bisector $Vvar(f_i, g_i)$ such that the arc $[v^*, u_i]$ is a normalized approximation of the Voronoi curve $Vvar(f_i, g_i; X)$ in $2B$. We can do the PV construction for $Vvar(f_i, g_i; X)$ in the box B': this will introduce a new node u_i' and arc $[u_i, u_i']$ in B'. Moreover, the ordering $u_1, \ldots, u_m$ on the edge $B' \cap 2B$ will induce a unique ordering of the nodes $u_1', u_2', \ldots, u_m'$ on the boundary of B'. This completes our description of the graph $G(B')$ to be constructed.

We can continue in this way to trace the Voronoi curve defined by $Vvar(f_i, g_i)$ until it reaches the boundary of $10B$. Note that this construction is valid according to the PV theory because $C_1^{f_i, g_i}(10B)$ holds (and hence C_1 holds for each box in $(\mathbb{S}_4)_{10B}$).

We are not quite done yet. Note that there might be nodes of $PV(\mathbb{S}_3)$ on the boundary of $10B$ which are not yet connected by an arc into $10B$. In Figure 6(a), we show such a node w. Therefore, we consider all such nodes and similarly "trace" them into the $10B$. This is just the straight PV construction for each node, associated with some bisector $Vvar(f, g)$. But note that such tracing will never reach the boundary of $2B$ because of Lemma 5; in short, if it enters the root box, it must eventually return to the boundary of $10B$ again (w' in Figure 6(a)). These represent "incursions" into $10B$. In tracing the the curve for $Vvar(f, g)$ through a box $B'' \in (\mathbb{S}_4)_{10B}$, we can always order the nodes on the boundary of B'' correctly based on the principle of non-crossing of edges.

To summarize: we have traced the curves from the boundary of $2B$ to the boundary of $10B$, and we have traced curves from the boundary of 10 back to the boundary of $10B$. This results in a graph $G(B'')$ for each $B'' \in (\mathbb{S}_4)_{10B}$, and their union is the desired graph $G(10B)$.

4. The Main Algorithm

Voronoi diagrams can be considerably simplified if X is non-degenerate. In fact, many papers in the literature on Voronoi diagrams (e.g., [YSL12, EMM13]) assume (implicitly or explicitly) non-degenerate inputs. We say X is **non-degenerate** if for every subset $Y \subseteq X$ with more than 3 functions, $Vvar(Y)$ is empty. Otherwise X is **degenerate**. A family of inputs is said to have **limited degeneracy** if there is a constant $k_0 \geq 3$ such that for all X in this family, if $Y \subseteq X$ and $|Y| > k_0$ then $Vvar(Y)$ is empty. Many practical data sets have limited degeneracy: for instance, planar architectural drawings are inevitably degenerate (four co-circular points or co-circular edges are universal) but the order of degeneracy is generally limited to $k_0 = 4$.

We now outline the algorithm to approximate $\mathrm{Vor}(X)$ under the assumption of limited degeneracy for some fixed k_0. Note that our algorithm will works for arbitrarily degenerate inputs, simply by setting $k_0 = |X|$. However, we do not see this as an efficient solution. Our algorithm has 5 phases:

```
Minimization Diagram Clustering Algorithm
    Input: $(X, \varepsilon, B_0)$ where $X$ is a simple set of feature functions
        with limited degeneracy $k_0 \geq 3$.
    Output: Embedded planar graph $G$ that is an $\varepsilon$-isotopic
        approximation of $\mathrm{Vor}(X) \cap B^*$
        (see Correctness Theorem, Appendix A).

    (I) SUBDIVISION PHASE
        Subdivide $B_0$ until all boxes have at most $k_0$ features.
    (II) ROOTBOX PHASE
        Among the boxes with $\geq 3$ features,
            compute a proper set $Q_{root}$ of root boxes.
    (III) SMOOTH PV CONSTRUCTION PHASE
        Apply the PV process to the complement
            of the domain of root boxes in $Q_{root}$
        Resulting subdivision is $\mathbb{S}_3$ (Section 3.1).
        Construct $PV(\mathbb{S}_3)$
    (IV) CONFORMAL SUBDIVISION PHASE
        For each $B \in Q_{root}$,
            construct a smooth subdivision of $10B$
                which is conformal with $\mathbb{S}_3$.
    (V) CLUSTER CONSTRUCTION PHASE
        For each $B \in Q_{root}$,
            construct the embedded graph $G(10B)$ (Section 3.4)
```

We next provide details for each phase.

4.1. (I) Subdivision Phase.

We will maintain four queues Q_k $(k = 0, 1, 2, 3)$ of boxes, with this invariant: *the union of these queues forms a subdivision of* B_0. We assume that the feature set $\widetilde{\phi}(B)$ is computed as each box B is created. Boxes in queue Q_k $(k = 0, 1, 2, 3)$ are characterized by the size $|\widetilde{\phi}(B)|$ of the active feature set of B:

$$|\widetilde{\phi}(B)| \begin{cases} > k_0 & \text{if } k = 0, \\ = 1 & \text{if } k = 1, \\ = 2 & \text{if } k = 2, \\ \in \{3, 4, \ldots, k_0\} & \text{if } k = 3. \end{cases} \tag{4}$$

Phase (I) will keep splitting the boxes in Q_0 until it is empty. Termination is assured under the assumption of limited degeneracy.

We would like the boxes in these queues form a smooth subdivision of B_0. However, the boxes in Q_1 have no part in the graph construction, and so the region $\bigcup Q_1$ may be omitted from this smoothing. Thus it is sufficient to maintain the following invariant:

The boxes in $Q_0 \cup Q_2 \cup Q_3$ form a smooth subdivision of the region $B_0 \setminus \bigcup Q_1$. (5)

It is easy to modify the smooth split routines to avoid splitting boxes in Q_1. The first phase can now be presented:

```
(I) SUBDIVISION PHASE
    $Q_0 \leftarrow \{B_0\}$ ; $Q_1 \leftarrow Q_2 \leftarrow Q_3 \leftarrow \emptyset$
    While $Q_0 \neq \emptyset$
        $B \leftarrow Q_0.pop()$
        $k \leftarrow |\widetilde{\phi}(B)|$
        If $k = 1$ or 2
            $Q_k.push(B)$
        If $k \leq k_0$
            $Q_3.push(B)$
        Else
            $Q_0.push(\texttt{sSplit}(B))$
```

4.2. (II) Root Box Phase.

At the beginning of this phase, Q_0 is empty and $Q_2 \cup Q_3$ is a smooth subdivision of $B_0 \setminus \bigcup Q_1$. Our goal in this phase is to construct a proper set of root boxes, to be stored in Q_{root}. Note that the boxes in Q_3 are candidates for root boxes. We keep removing boxes from Q_3 and test if they are root boxes. Moreover, Q_3 will be a priority queue in which larger boxes are removed first.

To check if $B \in Q_3$ is a root box, we need to check conditions (R1)-(R5) in Section 3. In fact, we will need a bit more, as encoded in the following "IsRootBox(B)" predicate:

```
IsRootBox($B$):
    return   [ $(12B \subseteq B_0)$
           $\wedge$ $(10B \cap^* 12B' = \emptyset$ for all $B' \in Q_{root})$
           $\wedge$ $(\widetilde{\phi}(B) = \widetilde{\phi}(10B))$
           $\wedge$ $(C_0 \vee C_1)(10B))$
           $\wedge$ $(JC(12B))$
           $\wedge$ $(MK(2B)))$ ]
```

This is a conjunction of 6 conditions, ordered in order of increasing complexity; we can terminate the evaluation as soon as a condition fails. The first two conjuncts concern the $12B$ and $10B$ respectively. They relate to the requirement that the set of root boxes must be *proper*. This predicate requires construction of the set $\widetilde{\phi}(10B)$ in addition to $\widetilde{\phi}(B)$. As with $\widetilde{\phi}(B)$, we want to exploit fact that $\widetilde{\phi}(10B)$ is a subset of $\widetilde{\phi}(10B')$ where B' is the parent of B. Therefore, we modify Phase (I) so that $\widetilde{\phi}(10B)$ is computed alongside $\widetilde{\phi}(B)$ when box B is created. If this predicate fails, we will split B and put its children into Q_1, Q_2, Q_3, accordingly. Here then is the second phase:

```
(II) ROOTBOX PHASE
    $Q_{root} \leftarrow \emptyset$   // Initialization
    While $Q_3 \neq \emptyset$
        $B \leftarrow Q_3.pop()$
        If IsRootBox($B$)
            $Q_{root}.push(B)$
            CULL($B$)
        Else
            for all $B'$ in $split(B)$
                Push $B'$ into $Q_1$ or $Q_2$ or $Q_3$ (see (4))
```

Note that just after adding a root box B to Q_{root}, we call a routine called CULL(B). The goal of this operation is to ensure the following invariant:

$$Q_2 \cup Q_3 \textit{ is a smooth subdivision of the region } B_0 \setminus (\bigcup Q_1) \cup \left(\bigcup_{B \in Q_{root}} 10B\right). \tag{6}$$

The motivation for this invariant is to ensure that Q_{root} is a proper set of root boxes. Indeed, at the end of this phase, Q_3 is empty, and the boxes in Q_2 is a smooth subdivision of $(\bigcup Q_1) \cup \left(\bigcup_{B \in Q_{root}} 10B\right)$. The following implementation of CULL(B) uses a temporary queue Q_{tmp} that satisfies a simple invariant:

INVARIANT: $B' \in Q_{tmp}$ implies B' intersects the interior of $10B$

We can now understand the operations of CULL(B) as removing boxes from $Q_2 \cup Q_3$ which satisfies the invariant, and putting them into Q_{tmp}. The boxes in Q_{tmp} fall under two cases. (Case A) If a box is contained in $10B$, then we use it to pull more boxes into Q_{tmp}. (Case B) otherwise, we smooth-split the box, and each of its children are either placed in Q_{tmp} or into $Q_1 \cup Q_2 \cup Q_3$ using the usual criteria in (4).

```
CULL($B$): // $B$ is a newly discovered root box
        $Q_{tmp} \leftarrow \{B\}$
        While $Q_{tmp} \neq \emptyset$
            $B' \leftarrow Q_{tmp}.pop()$
            If $B' \subseteq 10B$, // Case A
                for all $B'' \in Q_1 \cup Q_2 \cup Q_3$ adjacent to $B'$
                    If $B'' \cap^* 10B \neq \emptyset$,
                        remove $B''$ from $Q_1 \cup Q_2 \cup Q_3$
                        $Q_{tmp}.push(B'')$
            Else // Case B
                Perform $\texttt{Split}(B')$
                for all $B'' \in \texttt{Split}(B')$
                    If $B'' \cap^* 10B \neq \emptyset$,
                        $Q_{tmp}.push(B'')$
                    Else
                        Place $B''$ into $Q_1, Q_2$ or $Q_3$.
```

The following lemma assures us that the process will halt, and the final set is a natural cluster:

Lemma 7
(a) Phase (III) halts.
Upon halting, we have:
(b) The set of boxes in Q_2 forms a smooth subdivision $\mathbb{S}_2$ of the region $B_0 \setminus (\bigcup Q_1) \cup \left(\bigcup_{B \in Q_{root}} 10B\right)$.
(c) Every Voronoi vertex in B_0 in contained in the set $\bigcup_{B \in Q_{root}} 2B$.
(d) The set Q_{root} of root boxes is proper.

4.3. (III) Smooth or PV Construction Phase.

After Phase (II), the queue Q_3 is empty and the boxes in Q_2 forms a smooth subdivision (denoted $\mathbb{S}_2$) of $B_0 \setminus (\bigcup Q_1) \cup \left(\bigcup_{B \in Q_{root}} 10B\right)$. Our goal is to carry out the PV construction. This goal is clearly achievable because there are no Voronoi vertices in $\bigcup \mathbb{S}_2$:

```
(III) SMOOTH CONSTRUCTION PHASE
    While $Q_2$ is non-empty
        $B \leftarrow Q_2.pop()$
        If $C_0(B)$
            $Q_1.push(B)$
        Elif $C_1(B)$
            Construct $PV(B)$
        Else
            for each $B' \in$ sSplit$(B)$,
                place $B'$ in $Q_1$ or $Q_2$ accordingly.
```

Note that the boxes in Q_1 are basically discarded since they play no role in our construction (but note that in Phase (II), Q_1 is actually useful in the CULL(B) subroutine). If we interpete Q_1 as the set of discarded boxes, we can also put into Q_1 any box B where the exclusion predicate $C_0(B)$ holds, i.e., $C_0^{f,g}(B)$ holds for all $f,g \in \widetilde{\phi}(B)$ (see Appendix A on PV construction). We could have done this in earlier phases, but it would be expensive (quadratic in the size of $\widetilde{\phi}(B)$). But now that $\widetilde{\phi}(B)$ has at most 2 elements, we could apply this test.

After this phase, we have constructed the output graph for the region $B_0 \setminus (\bigcup Q_1) \cup \left(\bigcup_{B\in Q_{root}} 10B\right)$.

4.4. (IV) Conformal Subdivision Phase

At this point, we have a smooth subdivision $\mathbb{S}_3$ of the region $B_0 \setminus (\bigcup Q_1) \cup \left(\bigcup_{B\in Q_{root}} 10B\right)$. For each root box $B \in Q_{root}$, we also have the subdivision of $10B$ into 25 boxes. According to Lemma 4 (Section 3.1), we must subdivide each of the 16 boxes in ann(B). The lemma tells us that this can be achieved by a very local method (i.e., if $B_1 \neq B_2 \in$ ann(B), the subdivision of B_1 can be done independently of B_2).

```
(IV) CONFORMAL SUBDIVISION PHASE
    For each $B \in Q_{root}$
        for each $B_1 \in$ ann$(B)$
            let $S(B_1)$ the set of boxes in $\mathbb{S}_3$ adjacent to $B_1$
            Subdivide $B_1$ and its descendants until
                they are conformal with $S(B_1)$
```

4.5. (V) Cluster Construction Phase

The final phase will construct the graphs $G(10B)$ for each root box $B \in Q_{root}$. Again, the construction is local to each B. The details from Section 3.4 are summarized here:

```
(V) CLUSTER CONSTRUCTION PHASE
    For each $B \in Q_{root}$
        Construct a cluster node in the center of $B$.
        Determine the set of valid nodes on $\partial(2B)$.
        Sort the valid nodes on each side of $2B$.
        Add an arc from the cluster node to each valid node.
        For each valid node $u$ in $\partial(2B)$,
            trace arcs from $u$ until we reach $\partial(10B)$.
        For each node on $\partial(10B)$ from Phase (III),
            trace arcs from $u$ until we return to $\partial(10B)$.
```

We summarize with the following theorem:

Theorem 8 (Correctness) The Algorithm terminates and the constructed graph is a clustered ε-isotopic approximation of $\mathrm{Vor}(X) \cap B_0$.

5. Anisotropic Voronoi Diagrams

In this section we introduce the class of Voronoi diagrams of point and polygonal sites equipped with anisotropic norms in the plane. This generalizes [LS03] which introduced anisotropic norms for point sites only. We show how to compute these Voronoi diagrams as an application of the algorithm that we developed in the previous section. We assume that the ambient space is a square box $B_0 \subseteq \mathbb{R}^2$.

Let $M \in \mathbb{R}^{2\times 2}$ be a symmetric positive definite matrix, and let $Q_M(v) = v^T M v$ where $v \in \mathbb{R}^2$. The anisotropic norm associated with M is $\|v\|_M := \sqrt{Q_M(v)}$.

We can define any norm in terms of its unit ball which must be a centrally symmetric convex body. In this interpretation, anisotropic norms are those whose unit balls are ellipses. Using this view it is easy to see that anisotropic Voronoi diagrams generalize multiplicatively weighted Voronoi diagrams. The latter represents the special case when $M = \begin{bmatrix} c^2 & 0 \\ 0 & c^2 \end{bmatrix}$ for some $c > 0$, which corresponds to a norm with a $(1/c)$-scaled Euclidean disk as its unit ball.

5.1. Distance Computations

We wish to compute the separation between a point r and a point site p or a line segment site (p,q) under an anisotropic norm $\|\cdot\|_M$ where $p,q,r \in \mathbb{R}^2$. In order to simplify our computations, we instead work with the squares of the separation functions. These induce the same bisectors. We also compute the gradients of the separation functions, which we need in order to compute our box predicates. Let $v = q - p$ in the case of the line segment and $w = r - p$ in both cases.

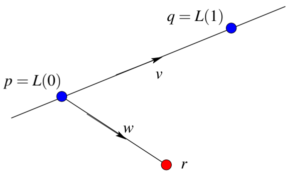


Figure 9: *Notations for distance to line computation*

For a point p,

$$\mathrm{Sep}_p(r)^2 = Q_M(w) \tag{7}$$

$$\nabla \mathrm{Sep}_p(r)^2 = 2Mw. \tag{8}$$

Next consider a line feature L with parametrization $L : \mathbb{R} \to \mathbb{R}^2$. Then the separation of a point r from L is given by $\min\{\|L(t) - r\|_M : t \in \mathbb{R}\}$. Let $t^*(r) = \operatorname{argmin}\{\|L(t) - r\|_M : t \in \mathbb{R}\}$.

In fact, we are interested in the separation function not for a line L, but for a line segment (p,q). Given such a segment, let L be the parametrization of the line passing through p,q where $L(0)=p$ and $L(1)=q$. Then $\mathrm{Sep}_{(p,q)}(r)$ is defined piecewise, with the active component depending on whether r is closest to the interior of L or one of the endpoints of L.

We give a formula for $t^*(r)$, and use it to derive a formula for $\mathrm{Sep}_L(r)$ and $\mathrm{Sep}_{(p,q)}(r)$. See Appendix C for the proof.

Lemma 9 Let $p,q,r \in \mathbb{R}^2$ be points, let $L(t)$ be the parametrization of the line running through p,q with $L(0)=p, L(1)=q$, let $v=q-p$ and let $w=r-p$. Then

$$\mathrm{Sep}_L(r) = \sqrt{Q_M(w) - (v^T M w)^2 / Q_M(v)}.$$

This minimum distance is achieved at the point $L(t^*(r))$ where

$$t^*(r) = \frac{v^T M w}{Q_M(v)}.$$

Lemma 10 Consider the square separation function $\mathrm{Sep}_{(p,q)}(r)^2$ and its gradient:
(a) Its piecewise algebraic formula is given by:

$$\mathrm{Sep}_{(p,q)}(r)^2 = \begin{cases} \mathrm{Sep}_p(r)^2 & \text{if } t^*(r) \le 0, \\ Q_M(w) - (v^T M w)^2/Q_M(v) & \text{if } t^*(r) \in (0,1), \\ \mathrm{Sep}_q(r)^2 & \text{if } t^*(r) \ge 1. \end{cases}$$

$$\nabla \mathrm{Sep}_{(p,q)}(r)^2 = \begin{cases} \nabla \mathrm{Sep}_p(r)^2 & \text{if } t^*(r) \le 0, \\ 2M(w - \frac{v^T M w}{Q_M(v)} v) & \text{if } t^*(r) \in (0,1), \\ \nabla \mathrm{Sep}_q(r)^2 & \text{if } t^*(r) \ge 1. \end{cases}$$

(b) The square separation function is C^1, i.e., it is continuous and its the gradient $\nabla \mathrm{Sep}_{(p,q)}(r)^2$ is well defined for all $r \in \mathbb{R}^2$.

5.2. Lipschitz Constant Computations

In order to track active features (i.e., compute $\widetilde{\phi}(B)$ for box B) we need an upper bound on the Lipschitz constants of anisotropic norms. In fact, we compute the Lipschitz constants exactly; see Appendix C for a derivation of the following expression.

Lemma 11 Let $M = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ be a symmetric positive definite matrix. Then for a site S equipped with $\|\cdot\|_M$ we have

$$K(S) = \frac{1}{\sqrt{2}} \sqrt{a + c + \sqrt{(a-c)^2 + 4b^2}}.$$

5.3. Implementation

We have implemented a prototype of our algorithm for anisotropic Voronoi diagrams. It follows our algorithm for tracking active features, and using box predicates to achieve a topologically correct Voronoi diagram. The visualization component uses OpenGL, and supports basic interaction such as zooming.

However, we emphasize that the code is preliminary. In particular it does not yet implement all of the details described in the construction phase of the algorithm, and computes up to machine (double) precision. Additionally, it only guarantees topological correctness for input in general position (meaning that exactly 3 Voronoi bisectors meet at every Voronoi vertex), rather than supporting limited degeneracy as described in our algorithm.

The program takes two parameters, ε_a and ε_g, which control precision by either limiting or forcing quadtree boxes to split. The first, ε_a, specifies an "absolute" minimum radius for quadtree boxes. If a box's radius is smaller than ε_a and the box is still unresolved then the program ceases splitting and marks the box as unresolved. A box may be marked as unresolved either because it contains a degenerate Voronoi vertex (more than 3 Voronoi bisectors intersecting at a point), or because ε_a was set higher than necessary for convergence. If a box B is marked as unresolved then we do not guarantee anything about the topology of the Voronoi diagram inside B.

The second parameter, ε_g, specifies a bound on the desired "geometric" accuracy of the computation. If a box intersects an active Voronoi bisector then it will always be split down to radius $\varepsilon_g/2$, ensuring that the Hausdorff distance between the actual Voronoi diagram and the computed approximation is less than ε_g.

5.3.1. Examples

We next give examples of Voronoi diagrams computed by our program. Input sites are shown in black, the subdivision grid in gray, and the computed (approximate) Voronoi diagram in red. Unresolved boxes are shown in blue. We show four diagrams produced by our program.

6. Conclusion

We have provided a general algorithm to compute a clustered ε-isotopic approximation of the minimization diagram $\mathrm{Vor}(X)$ of a set of scalar functions. Our requirements on X are natural properties commonly found in Voronoi diagram applications. Our current algorithm can treat full degeneracy, but it is not expected to be efficient. In practice, this is not an issue since unbounded degeneracy seems to arise only by deliberate design. The current implementation assumes non-degeneracy (so it does not attempt to construct root boxes). Further, it does not treat polygonal sites; though we plan to handle this extension.

References

[AK00] Aurenhammer F., Klein R.: Voronoi diagrams. In *Handbook of computational geometry*, Sack J. R., Urrutia J., (Eds.). Elsevier Publishing House, 2000, pp. 201–290. 1

[AKL13] Aurenhammer F., Klein R., Lee D.-T.: *Voronoi Diagrams and Delaunay Triangulations*. World Scientific, 2013. 1

[BCGY12] Burr M., Choi S., Galehouse B., Yap C.: Complete subdivision algorithms, II: Isotopic meshing of singular algebraic curves. *J. Symbolic Computation 47*, 2 (2012), 131–152. Special Issue for ISSAC 2008. 3, 6, 14, 15

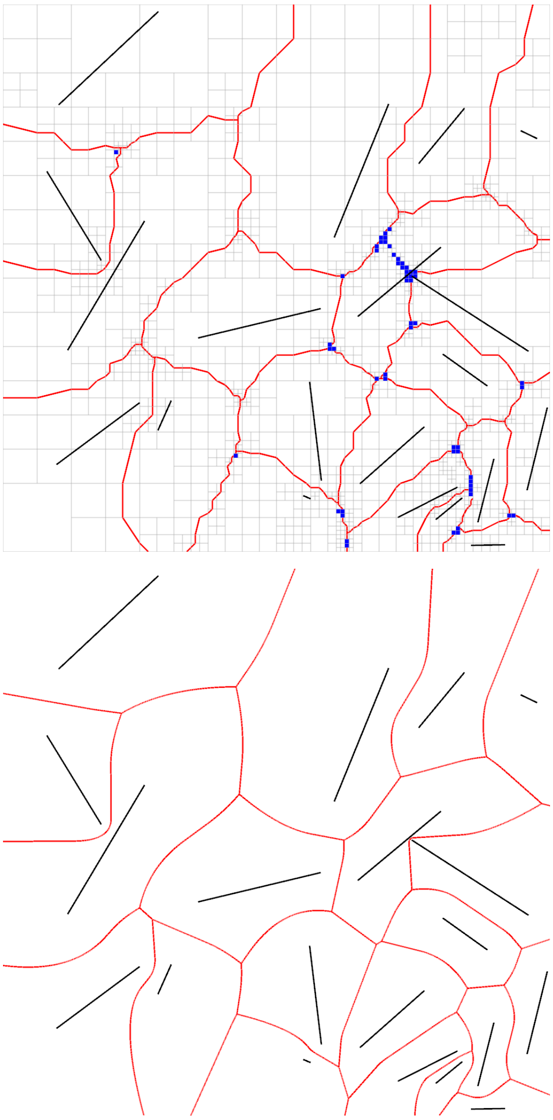


Figure 10: *These two images show a Voronoi diagram computed on the same collection of line segments. The first image was produced with ε_a set to be relatively large, and with no ε_g, while the second image was produced with small ε_g. The first image shows that relatively little splitting is necessary to trace bisectors and confirm many Voronoi vertices. The second image (in which the grid is turned off) shows the effect of computing to high geometric precision (small ε_g).*

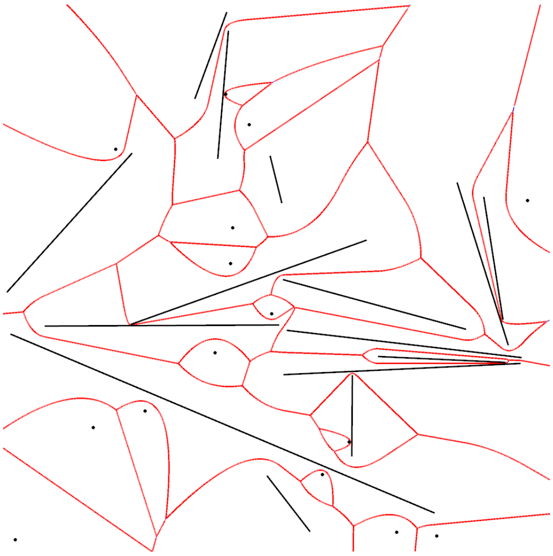


Figure 11: *A Voronoi diagram with mixed point and line segment input sites with small ε_g.*

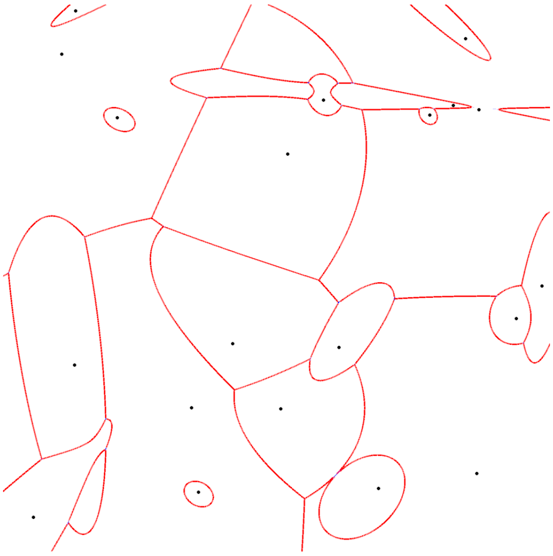


Figure 12: *A Voronoi diagram with point sites each equipped with a different anisotropic metric. Some of the metrics are very different, leading to disconnected Voronoi regions.*

[BSS*16] BECKER R., SAGRALOFF M., SHARMA V., XU J., YAP C.: Complexity analysis of root clustering for a complex polynomial. In *41st Int'l Symp. Symbolic and Alge. Comp.* (2016). To appear, ISSAC 2016. July 20-22, Wilfrid Laurier University, Waterloo, Canada. 3

[BWY06] BOISSONNAT J.-D., WORMSER C., YVINEC M.: Curved voronoi diagrams. In *Effective Computational Geometry for Curves and Surfaces*, Boissonnat J.-D., Teillaud M., (Eds.). Springer, 2006. Chapter 2. 1

[BY14] BENNETT H., YAP C.: Amortized analysis of smooth box subdivisions in all dimensions. In *14th Scandinavian Symp. and Workshops on Algorithm Theory (SWAT)* (2014), vol. 8503 of *Lect. Notes in C.S.*, Springer-Verlag, pp. 38–49. July 2-4 2014. Copenhagen, Denmark. To appear in CGTA. 5

[CCK*06] CHANG E.-C., CHOI S. W., KWON D., PARK H., YAP C.: Shortest paths for disc obstacles is computable. *Int'l. J. Comput. Geometry and Appl. 16*, 5-6 (2006), 567–590. Special Issue of IJCGA on Geometric Constraints. (Eds. X.S. Gao and D. Michelucci). 2, 3

[EMM13] EMIRIS I., MANTZAFLARIS A., MOURRAIN B.: Voronoi Diagrams of algebraic distance fields. *Computer Aided Design 45*, 2 (2013), 511–516. 2, 4, 9

[ES86] EDELSBRUNNER H., SEIDEL R.: Voronoi diagrams and arrangements. *Discrete and Comp. Geom. 1* (1986), 25–44. 2

[Kle89] KLEIN R.: *Concrete and abstract Voronoi diagrams*. Lecture Notes in Computer Science, No. 400. Springer-Verlag, Berlin, 1989. 2

[LS03] LABELLE F., SHEWCHUK J. R.: Anisotropic voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *Proc. 19th ACM Symp. on Comp. Geom.* (New York, NY, USA, 2003), ACM, pp. 191–200. 4, 11

[LSVY14] LIEN J.-M., SHARMA V., VEGTER G., YAP C.: Isotopic arrangement of simple curves: An exact numerical approach based on subdivision. In *ICMS 2014* (2014), Springer, pp. 277–282. LNCS No. 8592. Download from http://cs.nyu.edu/exact/papers/ for a version with Appendices and details on MK Test. 6, 8, 16, 17

[LY11a] LIN L., YAP C.: Adaptive isotopic approximation of nonsingular curves: the parameterizability and nonlocal isotopy approach. In *Discrete and Comp. Geom.* [LY11b], pp. 760–795. 2

[LY11b] LIN L., YAP C.: Adaptive isotopic approximation of nonsingular curves: the parameterizability and nonlocal isotopy approach. *Discrete and Comp. Geom. 45*, 4 (2011), 760–795. 14

[Mil93] MILENKOVIC V.: Robust construction of the Voronoi diagram of a polyhedron. In *Proc. 5th Canadian Conf. on Comp. Geom. (CCCG)* (1993), Lubiw A., Urrutia J., (Eds.), pp. 473–478. Univ. of Waterloo, Ontario, Canada. Aug 5-9, 1993. 6

[MK80] MOORE R. E., KIOUSTELIDIS J. B.: A simple test for accuracy of approximate solutions to nonlinear (or linear) systems. *SIAM J. Numer. Anal. 17*, 4 (1980), 521–529. 6, 15, 16

[Mun84] MUNKRES J. R.: *Elements of Algebraic Topology*. The Benjamin/Cummings Publishing Company, Inc, Menlo Park, CA, 1984. 2

[OBSC00] OKABE A., BOOTS B., SUGIHARA K., CHIU S. N.: *Spatial Tessellations — Concepts and Applications of Voronoi Diagrams*, 2nd ed. John Wiley and Sons, 2000. 1

[PV04] PLANTINGA S., VEGTER G.: Isotopic approximation of implicit curves and surfaces. In *Proc. Eurographics Symposium on Geometry Processing* (New York, 2004), ACM Press, pp. 245–254. 5, 6, 14, 15

[RR84] RATSCHEK H., ROKNE J.: *Computer Methods for the Range of Functions*. Horwood Publishing Limited, Chichester, West Sussex, UK, 1984. 5

[Sam90] SAMET H.: *The Design and Analysis of Spatial Data Structures*. Addison Wesley, 1990. 2

[WCY15] WANG C., CHIANG Y.-J., YAP C.: On Soft Predicates in Subdivision Motion Planning. *Comput. Geometry: Theory and Appl. 48*, 8 (Sept. 2015), 589–605. DOI:10.1016/j.comgeo.2015.04.002. 2

[Yap87] YAP C. K.: An $O(n \log n)$ algorithm for the Voronoi diagram for a set of simple curve segments. *Discrete and Comp. Geom. 2* (1987), 365–394. 1

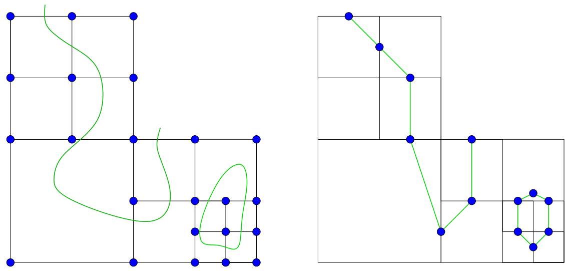
[YSL12] YAP C., SHARMA V., LIEN J.-M.: Towards Exact Numerical Voronoi diagrams. In *9th Proc. Int'l. Symp. of Voronoi Diagrams in Science and Engineering (ISVD).* (2012), IEEE, pp. 2–16. Invited Talk. June 27-29, 2012, Rutgers University, NJ. 3, 4, 5, 6, 9

[YSS13] YAP C., SAGRALOFF M., SHARMA V.: Analytic root clustering: A complete algorithm using soft zero tests. In *Computability in Europe (CiE2013)* (Heidelberg, 2013), Bonizzoni P., Brattka V., Lowe B., (Eds.), vol. 7921 of *Lect. Notes in C.S.*, Springer, pp. 434–444. Invited Talk. Special Session on "Computational Complexity in the Continuous World", July 1-5, Milan, Italy. 3

Appendix A: Three Fundamental Techniques

PV Construction.

We review the basic theory from [PV04,LY11b,BCGY12]. In computing $\mathrm{Vor}(X)$ restricted to a box B_0. Suppose we want to approximate the bisector $f = g$ where $f, g \in X$. Here, we will rely on two box predicates from [PV04]: *Exclusion Predicate* $C_0^{f,g}(B) : 0 \notin \square(f-g)(B)$ and *Normal Variation Predicate* $C_1^{f,g}(B) : 0 \notin (\square(f-g)_x(B))^2 + (\square(f-g)_y(B))^2$. Note that $(f-g)_x$ indicates partial derivative with respect to x. Given a smooth subdivision $\mathbb{S}$, we define two embedded planar graphs $G(\mathbb{S})$ and $PV(\mathbb{S})$. This is illustrated in Figure 13.



(a) $G(S) = (V,E)$ where $|V| = 23$

(b) $PV(S) = (N,A)$ with $|N| = 13$

Figure 13: *$G(\mathbb{S})$ and $PV(\mathbb{S})$ where $|\mathbb{S}| = 14$*

- The graph $G(\mathbb{S}) = (V,E)$ is basically a "non-uniform grid graph" where $V = V(\mathbb{S}) \subseteq \mathbb{R}^2$ is the set of **vertices** which are corners of boxes in $\mathbb{S}$, and $E = E(\mathbb{S})$ is the set of **edges** which are line segments $[u,v] \subseteq \mathbb{R}^2$ such that $u, v \in V(\mathbb{S})$ and the interior of $[u,v]$ does not contain any vertices of $V(\mathbb{S})$. In Figure 13(a), we have $|\mathbb{S}| = 14$, $|V| = 23$ and $|E| = 34$.
- The graph $PV(\mathbb{S}) = PV^{f,g}(\mathbb{S})$ is defined relative to a bisector $f = g$. It is well defined under these conditions:

(P1) The curve $f = g$ does not pass vertices of $V(\mathbb{S})$.
(P2) The curve is non-singular in $\bigcup \mathbb{S}$.
(P3) Each box $B \in \mathbb{S}$ satisfies the predicate $C_0 \vee C_1$.

Then $PV(\mathbb{S}) = (N(\mathbb{S}), A(\mathbb{S}))$ where $N(\mathbb{S}) \subseteq \mathbb{R}^2$ is the set of **nodes** and $A(\mathbb{S})$ is the set of **arcs**. This is illustrated in Figure 13(b), where $|N| = 23$ and $|A| = 34$.

In fact $PV(\mathbb{S})$ is best seen as the union of subgraphs $PV(B) = (N(B), A(B))$ for each box $B \in \mathbb{S}$. The subgraph $PV(B)$ is extremely simple having at most 4 nodes and at most 2 edges.

For completeness, we explicitly describe it here. For each edge $[u,v] \in E(\mathbb{S})$ on the boundary of B, we introduce a **node** $(u+v)/2$ if $f-g$ has different signs at u and at v. Let $N(B)$ be the set of these nodes. In general, $N(B)$ can have up to 8 nodes, but under the condition (P3), it can be shown that $N(B)$ has $0,2$ or 4 nodes. We then introduce the set $A(B)$ of **arcs** which connect the nodes of $N(B)$ in pairs as follows: if $N(B)$ has two nodes u,v, there is only one way to introduce an arc, namely $[u,v]$. In case $N(B)$ has four nodes u,v,u',v', then two of them (say u,v) must lie on one of the four sides of B. In this case we introduce the arcs $[u,u']$ and $[v,v']$ such that they do not intersect. This completes our description of $PV(B)$ and hence of $PV(\mathbb{S})$.

Suppose $B_0 \subseteq \mathbb{R}^2$ is a box. Our goal is to compute an isotopic approximation of the bisector $f=g$ restricted to B_0. The basic algorithm amounts to computing a smooth subdivision $\mathbb{S}$ of B_0 satisfying the conditions (P1-P3) above.

```
PV Algorithm for $B_0$
    I. SUBDIVISION PHASE:
        $Q_0 \leftarrow \{B_0\}$; $Q_1 \leftarrow \emptyset$
        While $Q_0 \neq \emptyset$
            $B \leftarrow Q_0.pop()$
            If $C_0(B)$ holds, discard $B$
            Elif $C_1(B)$ holds, $Q_1.push(B)$
            Else $Q_0.push(\texttt{sSplit}(B))$
    II. CONSTRUCTION PHASE:
        $G \leftarrow \emptyset$
        While $Q_1 \neq \emptyset$
            $B \leftarrow Q_1.pop()$
            Construct $PV(B)$, and add this to $G$
    Return $G$
```

The output of the algorithm is $G = PV(\mathbb{S})$ where $\mathbb{S}$ is the smooth subdivision produced at the end of the Subdivision Phase. The main correctness question is to characterize $PV(\mathbb{S})$ in relation to the bisector $f=g$. In [PV04], it is shown that $PV(\mathbb{S})$ is isotopic to $f=g$ provided $f=g$ is contained in B_0. Clearly this proviso is too limiting and [BCGY12] generalizes this. We will now provide such a statement of correctness. To achieve this, we slightly modify the above PV algorithm: suppose B is a **boundary box** (i.e., $\partial B_0 \cap \partial B$ is 1-dimensional).

Let $B \subseteq B_0$ be a boundary box. Another box $c(B)$ is called a **complement** of B if there is a line L through one of the sides of B_0 such that $c(B)$ is the reflection of B across L, and $c(B) \cap B_0$ is 1-dimensional. Let $co(B)$ be the set of complements of B. Note that $co(B)$ has one, two or four boxes. By definition, $co(B)$ is empty if B is not a boundary box. Define the predicate $\text{safe}(B)$ to mean $(\forall B' \in co(B))\Big[C_0(B') \vee C_1(B')\Big]$. In the PV algorithm above, we replace the line

$$\boxed{\text{Elif } C_1(B) \text{ holds, } Q_1.push(B)}$$

by

$$\boxed{\text{Elif } (C_1(B) \wedge \text{safe}(B)),\ Q_1.push(B)}$$

Let $\mathbb{S}$ be the smooth subdivision of B_0 produced by the modified PV algorithm. Let Z denote the set of boxes $B \in \mathbb{S}$ that are boundary boxes satisfying C_1 but not C_0. Let $\mathbb{S}^- := \mathbb{S} \setminus Z$ and $\mathbb{S}^+ := \mathbb{S} \bigcup \{co(B) : B \in Z\}$.

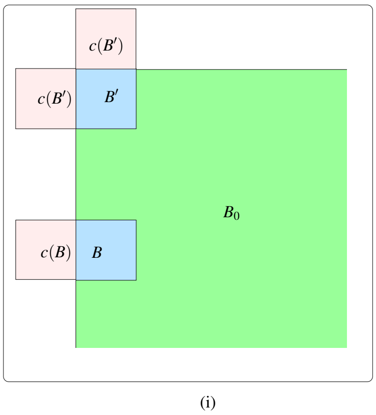


Figure 14: *The complements of boundary boxes B and B′*

Theorem A (Correctness of PV with boundary)
(Termination) If the curve $f=g$ is regular with respect to $\mathbb{S}$ and does not have a singularity in B_0, then the modified algorithm terminates.
(Correctness) On termination, the output graph $G = PV(\mathbb{S})$ is isotopic to the curve $f=g$ restricted to some region B^* where

$$\bigcup \mathbb{S}^- \subseteq B^* \subseteq \mathbb{S}^+$$

We can improve the uncertainty around the boundary of B_0 by making sure that boxes in Z must be ε-small (for any desired $\varepsilon > 0$).

Moore-Kioustelidis Test.

The second technique is necessary to compute Voronoi diagrams. The fundamental issue is to determine whether a box B contains a Voronoi vertex. Suppose we want to check if a Voronoi vertex defined by three features f,g,h lies inside B. We might think that since PV can approximate the bisector $f=g$, $g=h$ and $f=h$, it should be able to determine if these bisectors intersect inside a box. For consider the box $B = [0,1] \times [0,1]$ with three feature functions f,g,h illustrated in Figure 15. We order the feature functions at each corner of B. E.g., at the top left corner $(0,1)$, we have $f < g < h$. Based on these orderings, the graph $PV^{f,g}(B)$ has the arc $[(\frac{1}{2},0),(\frac{1}{2},1)]$ and similarly, the graphs $PV^{f,h}(B)$ and $PV^{g,h}(B)$ have the arc $[(0,\frac{1}{2}),(1,\frac{1}{2})]$. To visualize the construction, let us give a direction to each of the bisectors as indicated in Figure 15(a). Based on the labeling of corners, we may conclude that each of these bisectors enter B but they terminate at the Voronoi vertex defined by $\{f,g,h\}$ inside B. Thus, we introduce one node (not two nodes) for each of these bisectors. But Figure 15 shows that the three bisectors actually meet outside B. This example shows why the PV theory for smooth curves does not immediately extend to singularities.

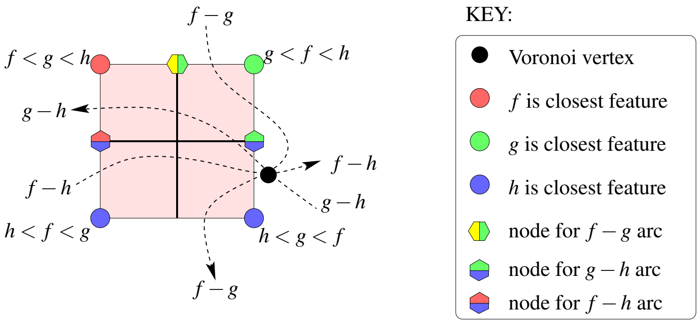


Figure 15: *PV construction mistaken about the presence of a Voronoi vertex: the dashed curves represent bisectors*

The technique we will use is based on the famous Miranda Test [MK80] on a box B: suppose the four sides of B are e_1, e'_1, e_2, e'_2 where (e_i, e'_i) form a pair of opposite edges (for $i = 1$ and 2). Miranda tells us that a sufficient condition for the $f = g$ and $g = h$ bisectors to intersect inside B is this: $f > g$ on e_1 and $f < g$ on e'_1, and $g > h$ on e_2 and $g < h$ on e'_2. By symmetry, there are several other sufficient conditions obtained by interchanging the roles of f, g and g, h, or by reversing the inequalities. If the test fails on B, we subdivide B and continue testing the children. There are two twists to this idea: first, the iterated application of Miranda tests may fail to terminate. Intuitively, the effective solution [MK80] to apply the test to the system $J^{-1}F$ where $F = \begin{bmatrix} f-h \\ g-h \end{bmatrix}$, and J is the Jacobian of F evaluated at the midpoint of B. To ensure convergence, this must be properly implemented as the mean value form of $J^{-1}F$ as discussed in [MK80]. The transformation J^{-1} is a preconditioning transformation. The second twist is to apply the test, not to B, but to $2B$. We denote this test by $MK^{f,g,h}(B)$. See the appendix of [LSVY14] for details.

The success of $MK^{f,g,h}(B)$ implies that $Vvar(f,g,h) \cap 2B$ is nonempty. To ensure that $Vvar(f,g,h) \cap 2B$ has exactly one root, we use the **Jacobian test** $JC^{f,g,h}(B)$: if J_F is the Jacobian of the system $F = \begin{bmatrix} f-h \\ g-h \end{bmatrix} : \mathbb{R}^2 \to \mathbb{R}^2$, the test succeeds if $0 \notin \square J_F(B)$. Success implies that F has at most one root in B.

Conformal Subdivisions.

The third technique provides the glue between smooth curves and the Voronoi vertices produced by the previous two techniques. Suppose we have detected a root box B containing a $\{f,g,h\}$ Voronoi vertex. We still need to know how the bisectors outside of B connects to this Voronoi vertex. Imagine what can happen after the PV construction in the complement of B: in case we see just three nodes on the boundary ∂B corresponding to the $f = g$, $g = h$ and $h = f$ bisectors, then we simply connect these nodes to a new node u (representing the Voronoi vertex inside B) at the center of B. But we might see more than three nodes on ∂B. For instance, if we see three nodes corresponding to the $f = g$ bisector, we know that two of these nodes may be spurious, and they represent an incursion of the $f = g$ bisector into B. But how can we determine which two are spurious? We will modify (simplify) the technique in [LSVY14] whereby we provide B with a buffer area $10B$ such that we can smoothly merge a subdivision of $10B$ with the PV construction on the complement of $10B$ to provide the proper connections into the Voronoi vertex inside B.

Appendix B: Minimization Diagrams

We provide all the proofs of lemmas from the main section of Minimization Diagrams.

Lemma 1 *Let $\mathcal{S}$ be a set of distinct points and lines. For each point or line $S \in \mathcal{S}$, let* $\mathrm{Sep}_S(p) = \inf\{\|p-q\| : q \in S\}$ *where $\|\cdot\|$ is the Euclidean norm. Then the set $X = \{\mathrm{Sep}_S : S \in \mathcal{S}\}$ is simple.*
Proof. (S1) The bisectors of pairs of points or pairs of lines are straight lines. The bisection of a point and a line is a parabola. In the degenerate case of a point p that lies on a line L, the "parabola" degenerates into a line perpendicular to L and passing through p.
(S2) The set $Vvar(f,g,h)$ is either intersection of the intersection of a straight line with a parabola, or two straight lines. In the former case, there are at most two points in $Vvar(f,g,h)$. In the latter case, we may verify that these two lines are distinct and therefore has at most one intersection point. **Q.E.D.**

Before we prove the next lemma, we prove a helper result:

Lemma B.1 *Let B be a box with midpoint m_B and radius r_B. If $p \in B$, and $f, g \in X$ such that $Clr(p) = f(p)$ and $Clr(m_B) = g(m_B)$. Then $f(m_B) \le Clr(m_B) + (K_f + K_g) r_B$.*
Proof.

$$\begin{aligned} f(m_B) &\le f(p) + K_f\|p - m_B\| \\ &\le g(p) + K_f\|p - m_B\| \\ &\le g(m_B) + (K_f + K_g)\|p - m_B\| \\ &\le \mathrm{Clr}(m_B) + (K_f + K_g) r_B \end{aligned}$$

Q.E.D.

Lemma 2 *$\widetilde{\phi}(B)$ is a soft version of $\phi(B)$:*
(a) Conservative: $\phi(B) \subseteq \widetilde{\phi}(B)$
(b) Convergence: if $B_i \to p$ (for $i \to \infty$) converges monotonically to p then $\widetilde{\phi}(B_i) = \phi(p)$ for i large enough.
Proof.
(a) Recall that $f \in \widetilde{\phi}(B)$ iff

$$f(m_B) \le \mathrm{Clr}(m_B) + K_2(\widetilde{\phi}(\mathrm{par}(B))) r_B. \tag{9}$$

Suppose $f \in \phi(B)$. Then Lemma B.1 above shows that (9) holds. I.e., $f \in \widetilde{\phi}(B)$, as we desired.
(b) As $B_i \to p$, we see that $f(m_{B_i}) \to f(p)$, $\mathrm{Clr}(m_{B_i}) \to \mathrm{Clr}(p)$, and $r_{B_i} \to 0$. Therefore (9) eventually holds only when $f \in \phi(p)$. **Q.E.D.**

Lemma 3 *Let $B \in \mathcal{R}$ and $B' \in \mathbb{S}_2$ such that B' is adjacent to $10B$.*

Then B' is congruent to kB for some $k = 2^{-i}$ for some $i \geq 1$. Thus, $w(B') \leq w(B)/2$; see Figure 5(i).
Proof. Recall that $10B$ is tiled by 25 boxes each congruent to $2B$. Let $B_1, \ldots, B_{25}$ be these 25 boxes. We can write B_i $(i = 1, \ldots, 25)$ in the form $2B_i^*$ where B_i^* is congruent to B. It is not hard to see that B_i^* is an aligned box (and therefore B_i is not aligned). Wlog, let $B' \in \mathbb{S}_2$ be adjacent to B_1.

Suppose B' has width greater that $w(B)/2$. Then there is an aligned box B'' congruent to B that is adjacent to B_1^* and overlapping B'. This implies that B' is contained in B'', and hence B' has width $\leq w(B'')/2$, contradiction. **Q.E.D.**

Lemma 4
1. The set $\mathbb{S}_3 \cup \mathbb{S}_4$ is a smooth subdivision of $B_0 \setminus \bigcup Q_1$.
2. Moreover for each $B \in \mathcal{R}$, $core(B) \subseteq \mathbb{S}_4$. I.e., no box in $core(B)$ is split.
Proof. 1. Let $B \in \mathcal{R}$. It is sufficient to prove that if B' and B'' are adjacent boxes in $\mathrm{ann}(B)$ then $SS(B') \cup SS(B'')$ is smooth. To see this, consider the 1-dimensional subdivision of $B' \cap B''$ induced by $SS(B')$: this is a sequence of intervals, say $(I'_1, \ldots, I'_m)$ for some m. Assuming I'_1 is the interval touching the boundary of $10B$, the interval widths satisfy $w(I'_1) = w(I'_2)$ and $w(I'_i) = w(I'_{i+1})/2$ for $i = 2, \ldots, m$. Similarly, consider the subdivision of $B' \cap B''$ induced by $SS(B'')$, say $(I''_1, \ldots, I''_p)$ for some p. We have already argued above that $I'_m = I''_p$. Consider the smallest $i \geq 1$ such that $I'_{m-i} \neq I''_{p-i}$. Wlog, let $m \leq p$. Then it is easy to see that $m - i = 1$ and $p - i = 2$. This proves our claim that $SS(B') \cup SS(B'')$ is smooth.
2. This is based on our observation above that any for $B' \in \mathrm{ann}(B)$, the subdivision $SS(B')$ will contain the two children of B' that are adjacent to $\mathrm{core}(B)$. This means that none of the boxes in $\mathrm{core}(B)$ need to be split. **Q.E.D.**

Lemma 5 *Let $f, g \in \widetilde{\phi}(B)$.*
(a) There is a unique (f, g)-principal component in $10B$.
(b) The Voronoi curve $Vvar(f, g; X)$ when restricted to the unique (f, g)-component is a connected (possibly empty) set.

Proof. (a) Clearly $Vvar(f, g) \cap 10B$ has at least one component since $Vvar(f, g, h) \subseteq 2B$ is non-empty. Suppose there is more than one such component. Figure 7 illustrates two possibilities. Since $C_1^{f,g}(10B)$ holds, we may assume wlog that $Vvar(f, g)$ is x-monotone in $10B$, i.e., every vertical line intersect $Vvar(f, g) \cap 10B$ in at most one point. If $C, C' \subseteq 10B$ are the two principal components, it means that they lie on opposite sides of some vertical line L, and this line intersects $2B$. Let $a, b \in C$ (resp. $a', b' \in C'$) be points with the largest and smallest y-coordinates on the respective components. Note that there might be two points with the same largest (or smallest) y-coordinates, but in any case, we can always choose a, b (resp., a', b') such that the curve $C[a, b]$ (resp., $C'[a', b']$) passes through $2B$ (for instance, in Figure 7, if we replace b' by b'', the curve $C'[a', b'']$ would not pass through $2B$).

By the intermediate value theorem, there is point $c \in C[a, b]$ with gradient perpendicular to the vector $a - b$, i.e., $\nabla_{f-g}(c) \perp (a - b)$. Similarly, there is a point $c' \in C'[a', b']$ satisfying $\nabla_{f-g}(c') \perp (a' - b')$. Here $\nabla_{f-g}(c)$ denotes the gradient of $f - g$ evaluated at c. Moreover, the gradients $\nabla_{f-g}(c)$ and $\nabla_{f-g}(c')$ must each have horizontal components that points towards each other or directly away from each other. This means $\langle \nabla_{f-g}(c), \nabla_{f-g}(c') \rangle < 0$. This contradicts our assumption that the predicate $C_1^{f,g}(10B)$ holds.
(b) This part of the lemma looks at the Voronoi curve $Vvar(f, g; X)$. Again, $Vvar(f, g; X) \cap 10B$ may several connected components. Suppose we look that those components of $Vvar(f, g; X) \cap 10B$ which are subsets of the (f, g)-principal component $P \subseteq 10B$. Part(b) amounts to saying that $Vvar(f, g; X) \cap P$ is a connected set. To prove this, and we will order the features in $\widetilde{\phi}(B)$ as $f_1, f_2, f_3, \ldots, f_k$ for some $k \geq 3$. The order is arbitrary except that $f = f_1$ and $g = f_2$. Moreover, let $X_i = \{f_1, f_2, \ldots, f_i\}$ $(i = 2, \ldots, k)$. Part(b) is a consequence of this CLAIM: *$Vvar(f, g; X_i) \cap P$ is connected for all $i = 2, \ldots, k$.*

We show this by induction on i. If $i = 2$, the claim follows from part(a). For $i \geq 2$, the claim is trivial if $Vvar(f, g; X_{i+1}) \cap P$ is empty. But it is non-empty, then the curve $Vvar(g, f_{i+1})$ must intersect $Vvar(f, g; X_i) \cap P$ at a unique point, namely, $Vvar(f, g, f_{i+1}) \cap 10B$. That means that $Vvar(f, g; X_i) \cap P$ is split into two connected subcurves by $Vvar(f, g, f_{i+1})$, and is exactly one of the subcurves is $Vvar(f, g; X_{i+1}) \cap P$. This completes our induction. **Q.E.D.**

Lemma 6 *Consider the graph $PV(2B)$ and a node of bisector $f = h$ on an edge $e = uv$ of the box $2B$, where $f, h \in \widetilde{\phi}(2B)$. See Figure 8. This node is invalidated by a feature $g \in \widetilde{\phi}(B)$ iff one of the following conditions holds:*

(a) For an endpoint u of e, $\mathrm{Sep}_f(u) < \min(\mathrm{Sep}_g(u), \mathrm{Sep}_h(u))$ and bisectors $f = g$ and $f = h$ intersect e in this order as we move from u to v on e.
(b) For both endpoints of e, $\mathrm{Sep}_g(u) < \min(\mathrm{Sep}_f(u), \mathrm{Sep}_h(u))$ and $\mathrm{Sep}_g(v) < \min(\mathrm{Sep}_f(v), \mathrm{Sep}_h(v))$.

Proof. First we give some terminology and a property from [LSVY14]. Let v^* denote the Voronoi vertex of $Vvar(f, g, h)$ in $2B$. Recall that connected component within $2B$ of a bisector incident to v^* is called *principal*. Since the box $2B$ satisfies $MK(2B)$, the principal component of any bisector in $2B$ must have endpoints on two different sides of $2B$. This is shown in Lemmas 7, 8 of [LSVY14]. Consider the endpoint p of the principal component of $f = h$ on e. Because the bisectors involving the features f, g, h intersect exactly once within $2B$ (at v^*), point p is in $Vvar(g; \{f, g, h\})$ iff the entire segment $v^* p$ is in $Vvar(g; \{f, g, h\})$. Thus, $f = h$ is invalid on e iff $p \in Vvar(g; \{f, g, h\})$.

Suppose that $f = h$ is invalid on e; then $p \in Vvar(g; \{f, g, h\})$. Suppose first that $\mathrm{Sep}_f(u) < \min(\mathrm{Sep}_g(u), \mathrm{Sep}_h(u))$; thus, $u \in Vvar(f; \{f, g, h\})$. Then the Voronoi edge $Vvar(f, g; \{f, g, h\})$ intersects segment up. But this implies that bisectors $f = g$ and $f = h$ intersect e in the order given by Condition 1. If we exchange the roles of f and h (i.e., if $u \in Vvar(h; \{f, g, h\})$) the argument is symmetric resulting in Condition 1. Since the principal component of $f = h$ intersects e, not both endpoints of e can be in $Vvar(f; \{f, g, h\})$, nor in $Vvar(h; \{f, g, h\})$. Thus, the only remaining possibility is that both endpoints of e are in $Vvar(g; \{f, g, h\})$, resulting in Condition 2.

Conversely, suppose that Condition 1 holds. That is, suppose that $\mathrm{Sep}_f(u) < \min(\mathrm{Sep}_g(u), \mathrm{Sep}_h(u)))$ and bisectors $f = g$ and $f = h$ intersect e in this order as we move from u to v. Then, by the definition of bisector $f = g$, and since $\mathrm{Sep}_f(u) < \mathrm{Sep}_g(u)$, it follows that $\mathrm{Sep}_g(x) < \mathrm{Sep}_f(x)$ for any point x that is the intersection of $f = h$ and e. Thus, the endpoint p of the principal component of $f = h$ must be in $Vvar(g; \{f, g, h\})$. That is, $f = h$ is invalid on e.

Suppose now that Condition 2 holds, that is, both endpoints of e are in $Vvar(g; \{f, g, h\})$. Since any principal component of a bisector in $2B$ can have at most one endpoint on e, and since both endpoints of e belong in $Vvar(g; \{f, g, h\})$, it follows that no principal component of bisector $g = h$ or bisector $g = f$ can intersect e. But no other component of $g = h$ or $g = f$ (other than the principal one) can intersect $f = h$ in $2B$. Thus, $p \in Vvar(g; \{f, g, h\})$. Hence, bisector $f = h$ is invalid on e. **Q.E.D.**

Lemma 7
(a) Phase (III) halts.
Upon halting, we have:
(b) The set of boxes in Q_2 forms a smooth subdivision $\mathbb{S}_2$ of the region $B_0 \setminus (\bigcup Q_1) \cup \left(\bigcup_{B \in Q_{root}} 10B\right)$.
(c) Every Voronoi vertex in B_0 in contained in the set $\bigcup_{B \in Q_{root}} 2B$.
(d) The set Q_{root} of root boxes is proper.
Proof. (a) To show halting, it is enough to show that for every Voronoi vertex $v \in Vvar(Y; X)$ in the interior of B_0, if $v \in B$ and B is small enough, the IsRootBox(B) will hold.
(b) This is simply a consequence of our invariant (6) since upon termination Q_3 is empty.
(c) Clearly there are no boxes outside the root box domains since these boxes has at most two active features. Inside the domain of a root box B, we also know that any Voronoi vertex is confined to $2B$.
(d) Here we exploit the fact that Q_3 is a priority queue where boxes of largest width are popped first. In the IsRootBox(B) predicate, we ensured that $10B \cap^* 12B' = \emptyset$ for all $B' \in Q_{root}$. Since $w(B') \geq w(B)$, it follows that $12B \cap^* 10B'$ is empty. This, plus the test that $12B \subseteq B_0$ shows that Q_{root} will remain proper after we add B to Q_{root}. **Q.E.D.**

Appendix C: Anisotropic Voronoi Diagrams

This appendix contain proofs for Section 5.

Lemma 9 *Let $p, q, r \in \mathbb{R}^2$ be points, let $L(t)$ be the parametrization of the line running through p, q with $L(0) = p, L(1) = q$, let $v = q - p$ and let $w = r - p$. Then*

$$\mathrm{Sep}_L(r) = \sqrt{Q_M(w) - (v^T M w)^2 / Q_M(v)}.$$

This minimum distance is achieved at the point $L(t^*(r))$ where

$$t^*(r) = \frac{v^T M w}{Q_M(v)}.$$

Proof. We first compute $t^*(r)$. Note that $L(t) = L(0) + t \cdot (L(1) - L(0)) = L(0) + t \cdot v$. Thus $r - L(t) = (r - L(0) - t \cdot v) = w - tv$. The squared distance from r to $L(t)$ is given by $Q_M(r - L(t)) = Q_M(w - tv) = (w - tv)^T M (w - tv)$. The minimum is achieved by the t which satisfies $Q_M(w - tv)' = 0$. Here $(\cdot)'$ denotes differentiation with respect to t. Thus

$$\begin{aligned}
0 &= ((w - tv)^T M (w - tv))' \\
&= (w - tv)'^T M (w - tv) + (w - tv)^T M (w - tv)' \\
&= -v^T M (w - tv) - (w - tv)^T M v \\
&= -2v^T M (w - tv). \\
v^T M t v &= v^T M w \\
t &= (v^T M w)/(v^T M v),
\end{aligned}$$

which is our minimizing parameter $t^*(r)$.

To obtain the squared separation, we first expand the expression for $Q_M(w - tv)$:

$$\begin{aligned}
Q_M(w - tv) &= (w - tv)^T M (w - tv) \\
&= Q_M(w) + t^2 Q_M(v) - t(w^T M v + v^T M w) \\
&= Q_M(w) + t(t Q_M(v) - 2v^T M w).
\end{aligned}$$

Plugging in $t = (v^T M w)/Q_M(v)$:

$$\begin{aligned}
&= Q_M(w) + \frac{v^T M w}{Q_M(v)}(v^T M w - 2v^T M w) \\
&= Q_M(w) + \frac{v^T M w}{Q_M(v)}(-v^T M w) \\
&= \frac{Q_M(w) Q_M(v) - (v^T M w)^2}{Q_M(v)},
\end{aligned}$$

which yields our formula for the squared separation. **Q.E.D.**

Lemma 10 *Consider the square separation function $\mathrm{Sep}_{(p,q)}(r)^2$ and its gradient:*
(a) Its piecewise algebraic formula is given by:

$$\mathrm{Sep}_{(p,q)}(r)^2 = \begin{cases} \mathrm{Sep}_p(r)^2 & \text{if } t^*(r) \leq 0, \\ Q_M(w) - (v^T M w)^2 / Q_M(v) & \text{if } t^*(r) \in (0,1), \\ \mathrm{Sep}_q(r)^2 & \text{if } t^*(r) \geq 1. \end{cases}$$

$$\nabla \mathrm{Sep}_{(p,q)}(r)^2 = \begin{cases} \nabla \mathrm{Sep}_p(r)^2 & \text{if } t^*(r) \leq 0, \\ 2M(w - \frac{v^T M w}{Q_M(v)} v) & \text{if } t^*(r) \in (0,1), \\ \nabla \mathrm{Sep}_q(r)^2 & \text{if } t^*(r) \geq 1. \end{cases}$$

(b) The square separation function is C^1, i.e., it is continuous and its the gradient $\nabla \mathrm{Sep}_{(p,q)}(r)^2$ is well defined for all $r \in \mathbb{R}^2$.

Proof. The formula for the separation function follows immediately from the previous lemma. The formula for the gradient of $\mathrm{Sep}_{(p,q)}(r)^2$ follows by direct computation.

It remains to show that the square separation function is C^1.

From our geometric intuition, it should be clear that the function is continuous; but since we have a piece-wise formula, we expect to algebraically verify this. I.e., we must verify that when $t^*(r) = 0$ or 1, then the distance to the line is equal to the distance to the endpoints $L(0)$ or $L(1)$. This is obvious when $t^*(r) = 0$; we ask the reader to also check the case $t^*(r) = 1$.

Finally, we show that the piecewise defined gradient $\nabla \mathrm{Sep}_{(p,q)}(r)^2$ is also continuous, i.e., well-defined when $t^*(r) = 0$ or $t^*(r) = 1$. If $t^*(r) = 0$, we have two formulas for the gradient: $\nabla \mathrm{Sep}_p(r)^2$ and $2M(w - \frac{v^T Mw}{Q_M(v)} v)$. But $t^*(r) = 0$ implies $v^T Mw = 0$ and so the second formula becomes $2Mw$. As noted in (8) above, $\nabla \mathrm{Sep}_p(r)^2 = 2Mw$.

If $t^*(r) = 1$, we have two formulas for the gradient: $\nabla \mathrm{Sep}_q(r)^2$ and $2M(w - \frac{v^T Mw}{Q_M(v)} v) = 2M(w - v) = 2M(r - q)$. Again, (8) implies their equality. **Q.E.D.**

Lemma 11 *Let* $M = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ *be a symmetric positive definite matrix. Then for a site S equipped with* $\|\cdot\|_M$ *we have*

$$K(S) = \frac{1}{\sqrt{2}} \sqrt{a + c + \sqrt{(a-c)^2 + 4b^2}}.$$

Proof. If $q = (x,y)^T \in S^1$, then $\|q\|_M = \sqrt{ax^2 + 2bxy + cy^2} \leq \sqrt{a + 2|b| + c}$. Thus we have a simple upper bound $K_M \leq \sqrt{a + 2|b| + c}$.

To derive an exact bound, define the functions $f, h : \mathbb{R}^2 \to \mathbb{R}$ and point q_* by

$$\begin{aligned} f(q) &:= (\|q\|_M)^2 = ax^2 + 2bxy + cy^2, \\ h(q) &:= \|q\|^2 = x^2 + y^2, \\ q_* &:= \operatorname{argmax}\{f(q) : h(q) = 1\}. \end{aligned}$$

where $q = (x,y)^T$. Clearly, $f(q_*) = K_M^2$. The theory of Lagrange multipliers tells us that there is a $\lambda \neq 0$ such that

$$\nabla f(q_*) = \lambda \nabla h(q_*). \tag{10}$$

A computation shows that $\nabla f(q) = 2Mq$ and $\nabla h(q) = 2q$. So Equation (10) amounts to $2Mq^* = \lambda 2q^*$. This shows that λ is an eigenvalue of M, and q_* a corresponding unit eigenvector: $(M - \lambda I)q_* = \vec{0}$. Hence

$$f(q_*) = q_*^T M q_* = q_*^T(\lambda q_*) = \lambda.$$

We check that the eigenvalues of M are $\lambda = \frac{1}{2}\left(a + c \pm \sqrt{(a-c)^2 + 4b^2}\right)$, and since $f(q_*) = \lambda$ is a maximum, we conclude that

$$\lambda = \frac{1}{2}\left(a + c + \sqrt{(a-c)^2 + 4b^2}\right).$$

This proves that

$$K_M = \sqrt{\lambda} = \frac{1}{\sqrt{2}} \sqrt{a + c + \sqrt{(a-c)^2 + 4b^2}}.$$

This is a strict improvement of the simple upper bound, and when $M = aI$ then $K_M = \sqrt{a}$ as expected. **Q.E.D.**