

# Computing the Topology of Voronoï Diagrams of Parallel Half-Lines

Ibrahim Adamou · Bernard Mourrain

Received: date / Accepted: date

**Abstract** In this paper we consider the Voronoï Diagram of a finite family of parallel half-lines, with the same orientation, constrained to a compact domain  $\mathcal{D}_0 \subset \mathbb{R}^3$ , with respect to the Euclidean distance. We present an efficient approximation algorithm for computing such VD, using a subdivision process, which produces a mesh representing the topology of the VD in  $\mathcal{D}_0$ . The computed topology may not be correct for degenerate configurations or configurations close to degenerate. In this case, the output is a valid partition, which is close to the exact partition in Voronoï cells if the input data were given with no error. We also present the result of an implementation in Julia language with visualization using Axl software (axl.inria.fr) of the algorithm. Some examples and analysis are shown.

**Keywords** Voronoï Diagram · Parallel Half-Lines · Subdivision algorithm · kd-tree structure · Algebraic curves and surfaces · Topology

**Mathematics Subject Classification (2010)** 65D17 · 65D18

## 1 Introduction

The Voronoï diagram (VD) is a fundamental data structure in computational geometry with many and very diverse applications in theoretical and practical areas, such as proximity queries, high-clearance placements,

---

This work was partially supported by a grant from Simons Foundation and INRIA.

Ibrahim Adamou  
Département de Mathématiques,  
Faculté des Sciences et Techniques,  
Université Dan Dicko Dankoulodo de Maradi  
BP : 465 Maradi, Niger  
Tel.: +227 20410132  
Fax: +227 20410133  
E-mail: adamouhakime14@gmail.com

Bernard Mourrain  
INRIA, Université Côte d'Azur  
2004 route des Lucioles, BP 93  
06902 Sophia Antipolis Cedex, France  
E-mail: Bernard.Mourrain@inria.fr

motion planning problems, classification problems, and many more (see [5, 13, 21, 30, 35]). For a given discrete set of geometric objects (called *sites*)  $\mathcal{S} = \{s_1, \dots, s_n\}$  in a metric space  $(\mathcal{E}, d)$ , the VD subdivides  $\mathcal{E}$  in regions (called *cells*), where each region associated to one site  $s_i$  consists of the points closer to the site  $s_i$  than to any other site in  $\mathcal{S}$ , for the distance  $d$ . Generally, the VD can be defined as Minimization Diagram (MD) of a finite set of continuous functions  $\{f_1, \dots, f_n\}$ , each function  $f_i$  is defined as the distance function to a site  $s_i$ . The corresponding VD gives an identical subdivision of  $\mathcal{E}$  into cells (see [11]), i.e:

$$VD(s_1, \dots, s_n) \equiv MD(f_1, \dots, f_n).$$

Different types of sites and distance functions have been proposed for different kinds of applications depending on the context. Note that in the space  $\mathbb{R}^2$  the VDs have been studied extensively, the structure and computation are nowadays thoroughly well understood. Many robust and efficient algorithms have been proposed by several authors such as [16, 17, 20, 25, 37]. However, in three and higher dimension spaces, the VD have been much less studied, and many basic problems are still widely open. Some recent works for linear, quadrics, and semi-algebraic sites have been developed (see [5, 9, 15, 19, 23, 26, 29–31, 33, 35]).

In this paper, we consider the VD of a set of  $n$  parallel half-lines  $\{h_1, \dots, h_n\}$  with the same orientation  $\mathbf{u} = (\mathbf{a}, \mathbf{b}, \mathbf{c}) \neq (0, 0, 0)$ , with  $\|\mathbf{u}\| = 1$ .

$$h_i = A_i + t\mathbf{u}, t \geq 0, A_i = (x_i, y_i, z_i) \in \mathbb{R}^3, i = 1 \dots n, \quad (1)$$

where

$$\langle \overrightarrow{OA_i}, \mathbf{n} \rangle \neq \langle \overrightarrow{OA_j}, \mathbf{n} \rangle, \forall i \neq j \text{ and } \mathbf{n} \in \text{Span}(\mathbf{u})^\perp$$

constrained to a compact domain of the form

$$\mathcal{D}_0 = [a_{0,1}, b_{0,1}] \times [a_{0,2}, b_{0,2}] \times [a_{0,3}, b_{0,3}] \subset \mathbb{R}^3, \quad (2)$$

with respect to the Euclidean distance.

The Euclidean distance from a point  $Q = (x, y, z) \in \mathbb{R}^3$  to a site  $h_i$  is defined by:

$$g_i(x, y, z) = \begin{cases} d(Q, A_i) & \text{if } \langle \overrightarrow{A_i Q}, \mathbf{u} \rangle < 0 \\ \sqrt{d(Q, A_i)^2 - \langle \overrightarrow{A_i Q}, \mathbf{u} \rangle^2} & \text{else,} \end{cases} \quad (3)$$

where  $d(\cdot, \cdot)$  denotes the Euclidean distance.

The VD of this new kind of sites, could be used to provide an efficient solution to some theoretical and practical problems in the drilling industry (mining, offshore drilling, hydraulics, etc.), in particular for exploring drilling-deepwater closest ground, well collision and identifying unplanned ground avoidance (see [7, 22, 24, 32, 40, 41, 43]).

Two general classes of algorithms are available for computing VD: exact and approximate. Some of the exact algorithms require the manipulation of an exact representation of VD boundaries, dealing with high-degree curves and surfaces, and their intersection, which lead very often to algorithms, which are complex and difficult to implement (see [38]).

In Hemmer et al. [26] a complete exact algorithm is presented for computing the Voronoi Diagram of general configurations of lines in Three-Dimensional Space. The algorithm uses lower envelope algorithm and requires running time and space complexity  $\mathcal{O}(n^{3+\varepsilon})$ , and the data structures allows to respond for point-location queries in  $\mathcal{O}(\log^2 n)$  time.

Approximation algorithms are often used to overcome the difficulties of exact representation, but the topology of the resulting approximation is not accurate in many cases (see [9]). In this family of approaches,

subdivision methods have been employed successfully to compute Approximate Voronoï Diagrams see e.g. [14]. Boada et al. [9, 10] use a subdivision approach to compute approximate VDs for distinct site shapes and different distance functions but restricted to diagrams with connected Voronoï regions. Etzion and Rappoport [18] used a space subdivision algorithm to compute the Voronoï diagram for the sites that are polyhedra, and the algorithm is simple to implement. The benefit of fast graphics hardware has also been used in recent years for computing the approximate VDs for different types of sites see e.g. [12, 25, 27]. Two specific algorithms are presented in [39, 42] for computing the VDs of convex sites using adaptive subdivision approaches. It turns out that the algorithms are expensive.

The use of approximation algorithms is particularly well suited from a practical point of view, where it usually involves approximate numerical inputs, inaccurate measurements, etc.

In Aurenhammer et al. [4, 6] an important contribution on the same subject is made, with an extension to the cases of parallel line segments that could be used in the problem studied in [15]. They presented an exact method for the computation of this type of VD by using space-sweep by a perpendicular plane to the half-lines direction and making use of the relationship with the properly established planar power diagrams. The presented algorithm is easily implemented without requiring any manipulation of the involved three dimensional elements and provides an exact VD with logarithmic cost per diagram face.

In comparison, our approach uses approximate input data and computes an  $\varepsilon$ -approximation of the Voronoï-cells, with a running time in  $\mathcal{O}(\frac{1}{\varepsilon^3})$ . We observe an experimental quasi-linear dependency in the number of sites in our tests.

The method that we propose is based on a subdivision algorithm following a kd-tree structure to compute an approximate VD for a family of  $n$  parallel half-lines constrained to a compact domain  $\mathcal{D}_0 \in \mathbb{R}^3$  with respect to the Euclidean distance. Our algorithm considers the VD as minimization diagram and uses the lower envelope of squared distance functions to the sites, since the VD can be constructed from its projection. It produces a mesh representing an approximate VD which is topologically equivalent to the exact one in  $\mathcal{D}_0$  when no degenerate configurations occur. We estimate the complexity estimation of our algorithm from the experimentation. The execution time and the number of boxes produced evolve linearly in the number of sites. The first version of this work is presented in [1].

Our approach consists in subdividing the initial domain  $\mathcal{D}_0$  into subdomains until the topology for the boundary of the VD cells in each subdomain is fully determined from its intersection with the boundary of the subdomain, or if the size of the subdomain is smaller than a constant threshold initially fixed. Then the boundaries of the VD cell contained in each subdomain are meshed. The implementation of the method in Julia Language ([28]) is complete, and the experimentation shows accurate results with good timing.

The rest of the paper is organized as follows: We give the basic notion and algebraic description of the VD components in the section 2, in the section 3 we introduce the regularity criteria for the algebraic curve and surface and the computation of the topology in section 4. The description of the algorithm is presented in the section 5 and the implementation of the method is explained in the section 6 and finally the last section is devoted to conclusions.

## 2 Voronoï diagram of a family of parallel half-lines

Let  $\{h_1, \dots, h_n\}$  be a family of  $n$  parallel half-lines in the same orientation  $\mathbf{u}$  and  $\mathcal{D}_0$  be a compact domain of  $\mathbb{R}^3$  defined respectively in (1) and (2). We are interested in computing the VD of  $\{h_1, \dots, h_n\}$  constrained to  $\mathcal{D}_0$ , with respect to the Euclidean distance, by minimization diagram.

In the following we consider the orientation vector  $\mathbf{u} = (0, 0, 1)$  (i.e: in the  $z$ -direction), then the sites and associated distance function can be respectively defined as follows:

$$h_i = (x_i, y_i, z_i + t), t \geq 0, i = 1 \dots n, \quad (4)$$

where

$$(x_i, y_i) \neq (x_j, y_j), \forall i \neq j,$$

$$g_i(x, y, z) = \begin{cases} \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} & \text{if } z < z_i, \\ \sqrt{(x-x_i)^2 + (y-y_i)^2} & \text{if } z \geq z_i, \end{cases} \quad (5)$$

Associating each site  $h_i$ , defined by (4), the distance function  $g_i$ , defined by (5), we have (see [11, 13]):

- The VD of a family of parallel half-lines  $\{h_1, \dots, h_n\}$  constrained to a domain  $\mathcal{D}_0$  for the Euclidean distance is defined by the minimization diagram of the family of distance function  $\{g_1, \dots, g_n\}$  in  $\mathcal{D}_0$ , i.e:

$$VD(h_1, \dots, h_n) = MD(g_1, \dots, g_n), \quad (6)$$

where the Voronoï cell associated to a site  $h_i$  is defined by:

$$VD(i) = \{Q \in \mathcal{D}_0 : g_i(Q) \leq g_j(Q), j = 1 \dots n\}.$$

- Furthermore, since the distance functions  $g_i$  are Euclidean, then minimizing  $g_i$  and  $g_i^2$  over  $i$  are equivalent, therefore

$$VD(h_1, \dots, h_n) \equiv MD(g_1^2, \dots, g_n^2).$$

In the following, instead of the distance defined by (5), we consider the squared distances, i.e.:

$$g_i(x, y, z) = \begin{cases} (x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 & \text{if } z < z_i, \\ (x-x_i)^2 + (y-y_i)^2 & \text{if } z \geq z_i, \end{cases} \quad (7)$$

such that  $g_i \in \mathbb{R}[x, y, z]$  (piecewise).

## 2.1 Bisectors, Trisectors and Quadrisectors of sites

Let introduce an algebraic description of the VD cell components.

### 2.1.1 Bisector of two sites

Let  $h_i$  and  $h_j$  be two sites with their respective associated distance functions  $g_i$  and  $g_j$ . Let  $A_i = (x_i, y_i, z_i)$ ,  $A_j = (x_j, y_j, z_j)$ , and  $\ell_i, \ell_j$  be their respective endpoints and supporting lines.

The bisector surface of two sites  $h_i$  and  $h_j$ , is the equidistant surface from the two sites. It determines, locally, a VD face and it is defined by :

$$b_{i,j} = \{(x, y, z) \in \mathbb{R}^3 : g_i(x, y, z) - g_j(x, y, z) = 0\}.$$

Expanding the expression

$$g_i(x, y, z) - g_j(x, y, z) = (g_i - g_j)(x, y, z)$$

for  $z_i \leq z_j$ , we obtain:

$$(\mathbf{g}_i - \mathbf{g}_j)(x, y, z) = \begin{cases} p_{i,j}(z) + \ell_{i,j}(x, y) & \text{if } z < z_i, \\ q_{i,j}(z) + \ell_{i,j}(x, y) & \text{if } z_i \leq z < z_j, \\ \ell_{i,j}(x, y) & \text{if } z \geq z_j. \end{cases} \quad (8)$$

where

$$p_{i,j}(z) = 2z(z_j - z_i) + z_i^2 - z_j^2, \quad (9)$$

$$q_{i,j}(z) = (z - z_j)^2, \quad (10)$$

$$\ell_{i,j}(x, y) = 2x(x_j - x_i) + x_i^2 - x_j^2 + 2y(y_j - y_i) + y_i^2 - y_j^2. \quad (11)$$

For  $z_i \neq z_j$ , the bisector surface consists of three connected pieces of surfaces:

- A half-plane (for  $z < z_i$ ) : supported by bisector-plane of the two endpoints  $A_i$  and  $A_j$  ;
- A half-parabolic-cylinder piece (for  $z_i \leq z < z_j$ ) : supported by bisector-surface of endpoint  $A_j$  and supporting line  $\ell_i$  (since  $z_i < z_j$ );
- A half-plane ( for  $z \geq z_j$ ) : supported by bisector-plane of the two supporting line  $\ell_i$  and  $\ell_j$  .

For  $z_i = z_j$ , the bisector surface consists of one plane (bisector plane of  $A_i$  and  $A_j$ ).

### 2.1.2 Trisector of three sites

Let  $h_i, h_j$  and  $h_k$  be three sites with their respective distance functions  $\mathbf{g}_i, \mathbf{g}_j$  and  $\mathbf{g}_k$ ;

Let  $A_i = (x_i, y_i, z_i)$ ,  $A_j = (x_j, y_j, z_j)$ ,  $A_k = (x_k, y_k, z_k)$ , and  $\ell_i, \ell_j, \ell_k$  be their respective endpoints and supporting lines.

The trisector of the three sites  $h_i, h_j$  and  $h_k$  is the equidistant curve from the three sites. It determines locally, VD edge and it is given by the common intersection curve of the three ( $3 = \binom{3}{2}$ ) bisectors of the 3 sites (2 by 2):

$$\mathcal{T}_{i,j,k} = b_{i,j} \cap b_{i,k} \cap b_{j,k}, \quad (12)$$

or

$$\mathcal{T}_{i,j,k} = \{(x, y, z) \in \mathbb{R}^3, (\mathbf{g}_i - \mathbf{g}_j)(x, y, z) = (\mathbf{g}_i - \mathbf{g}_k)(x, y, z) = (\mathbf{g}_j - \mathbf{g}_k)(x, y, z) = 0\}.$$

Any equidistant portion of curve from more than three sites is called a degenerate edge portion. This can occur, for example if there are more than three neighbour parallel half-lines which intersect a circle in the  $xy$ -plane.

For  $z_i \neq z_j \neq z_k$  (for example  $z_i < z_j < z_k$ ), the trisector  $\mathcal{T}_{i,j,k}$  consists of four connected components of rational curves, and it is parametrized as follows :

$$\mathcal{T}_{i,j,k}(z) = \begin{cases} \left[ \frac{a_{2,1} + a_{3,1}z}{c_0}, -\frac{a_{0,1} + a_{1,1}z}{c_0}, z \right]^T & \text{if } z < z_i, \\ \left[ \frac{a_{3,2} + a_{4,2}z + a_{5,2}z^2}{c_0}, -\frac{a_{0,2} + a_{1,2}z + a_{2,2}z^2}{c_0}, z \right]^T & \text{if } z_i \leq z < z_j, \\ \left[ -\frac{a_{3,3} + a_{4,3}z + a_{5,3}z^2}{c_0}, \frac{a_{0,3} + a_{1,3}z + a_{2,3}z^2}{c_0}, z \right]^T & \text{if } z_j \leq z < z_k, \\ \left[ -\frac{a_{1,4}}{c_0}, \frac{a_{0,4}}{c_0}, z \right]^T & \text{if } z \geq z_k, \end{cases} \quad (13)$$

where:  $c_0, a_{l,l'}$  are constant expressions in terms of  $x_i, y_i, z_i, x_j, y_j, z_j, x_k, y_k, z_k$ .

The four components of the trisector are:

- A half-line (for  $z < z_i$ ) supported by the trisector line (the intersection of three bisector-planes) of the corresponding endpoints  $A_i, A_j$  and  $A_k$ .
- A piece of half-parabola (for  $z_i \leq z < z_j$ ) supported by the intersection of the bisector-plane of the endpoints  $A_j$  and  $A_k$ , a parabolic-cylinder (bisector of  $A_j$  and  $\ell_i$ ) and a parabolic-cylinder (bisector of  $A_k$  and  $\ell_i$ ).
- A piece of half-parabola (for  $z_j \leq z < z_k$ ) supported by the intersection of the bisector-plane of  $\ell_i$  and  $\ell_j$ , a parabolic-cylinder (bisector of  $A_k$  and  $\ell_i$ ) and a parabolic-cylinder (bisector of  $A_k$  and  $\ell_j$ ).
- A half-line parallel to the  $z$ -direction (for  $z \geq z_k$ ) supported by the trisector line (the intersection of the three bisector-planes) of the corresponding lines  $\ell_i, \ell_j$  and  $\ell_k$ .

The trisector  $\mathcal{T}_{i,j,k}$  consists of three connected components of rational curves if  $z_i \neq z_j = z_k$  or  $z_i = z_j < z_k$ , and it consists of one line if  $z_i = z_j = z_k$ .

### 2.1.3 Quadrisector of four sites

The quadrisector is the equidistant point from four sites  $h_i, h_j, h_k$ , and  $h_l$ . It determines the VD vertex and it is defined by the common intersecting point of four ( $4 = \binom{4}{3}$ ) trisectors spanned by the four sites (3 by 3):

$$Q_{i,j,k,l} = \mathcal{T}_{i,j,k} \cap \mathcal{T}_{i,j,l} \cap \mathcal{T}_{i,l,k} \cap \mathcal{T}_{j,l,k} \quad (14)$$

Any equidistant point from more than four sites is called a degenerate vertex.

## 3 Regularity criteria for the topology of algebraic curves and surfaces

Let's introduce some definitions and the notion of regularity criteria for the topology of algebraic curves and surfaces from [3] and [34] which can be applied to the trisector curve and the bisector surface case to compute their topology in a compact domain.

**Definition 1** Let  $\{g_1, \dots, g_n\}$  be a family of distance functions associated to a family of sites  $\{h_1, \dots, h_n\}$  and  $\mathcal{D}$  a compact domain of the form (2).

- A distance  $g_{i_0}$  is said to be active in  $\mathcal{D}$  if it contributes, locally, to the lower envelope of the family of distance functions  $\{g_1, \dots, g_n\}$ ,

$$i.e. : \exists (x_0, y_0, z_0) \in \mathcal{D}, g_{i_0}(x_0, y_0, z_0) \leq \min\{g_i(x_0, y_0, z_0), i = 1 \dots, n\}$$

and the site  $h_{i_0}$  is said active in the domain  $\mathcal{D}$ .

- For a domain  $\mathcal{D} \subset \mathbb{R}^3$ , we call  $d$ -list a list of distance functions that are held for  $\mathcal{D}$  denoted by  $B(\mathcal{D}) = \{g_{1|\mathcal{D}}, \dots, g_{p|\mathcal{D}}\}$  and  $|B(\mathcal{D})| = p$ .

*Remark 1* In practice, interval arithmetic is used on the distance functions to compute a set containing the corresponding active sites.

**Definition 2** An algebraic curve  $\mathcal{C}$  (or surface  $\mathcal{S}$ ) is said  $t$ -regular (for topological regularity) in a compact domain  $\mathcal{D}$  if its topology is uniquely determined from its intersection with the boundary of  $\mathcal{D}$ .

**Definition 3** An algebraic surface  $S$  is  $t$ - $z$ -regular (resp.  $t$ - $y$ -regular,  $t$ - $x$ -regular) in the domain  $\mathcal{D}$  if:

- $S$  has no tangent line parallel to the  $z$ -direction (reps.  $x$ ,  $y$ -direction),
- $S \cap F$  is  $t$ -regular, for  $F$  a  $z$ -facet (resp.  $x$ ,  $y$ -facet) of  $\mathcal{D}$ .

**Definition 4** A  $d$ -list  $B(\mathcal{D})$  is said to be **b-regular** if the corresponding bisectors and trisectors are  $t$ -regular in  $\mathcal{D}$ .

**Definition 5** A generic intersection with a domain  $\mathcal{D}$  is made as follows:

- For a bisector  $b_{i,j}$  of two sites:
  1. if its intersection with the faces of  $\mathcal{D}$  is of dimension 1, i.e:  $\dim(b_{i,j} \cap \mathcal{F}(\mathcal{D})) = 1$ , ( $\mathcal{F}(\mathcal{D})$  denotes the faces of  $\mathcal{D}$ );
  2. if its intersection with the edges of  $\mathcal{D}$  is of dimension 0 i.e:  $\dim(b_{i,j} \cap \mathcal{E}(\mathcal{D})) = 0$ , ( $\mathcal{E}(\mathcal{D})$  denotes the edges of  $\mathcal{D}$ ) with at most one intersection point on each edge;
  3. and if the part of the bisector delimited by the intersection points with the edges of  $\mathcal{D}$  is strictly contained in the domain  $\mathcal{D}$ .
- For the trisector  $\mathcal{T}_{i,j,k}$  of three sites:
  1. if its intersection with the boundary of  $\mathcal{D}$  is of dimension 0 with exactly two intersection points the boundary, i.e:  $\mathcal{T}_{i,j,k} \cap \partial \mathcal{D} = \{P_1, P_2\}$ ;
  2. and if the curve segment delimited by the two points is strictly contained in  $\mathcal{D}$ , i.e:  $\mathcal{T}_{i,j,k} \cap \mathcal{D} = [P_1, P_2] \subset \mathcal{D}$ .
- For the quadrisector of four sites: if it is strictly contained in domain  $\mathcal{D}$ .

The generic intersection allows to determine, during the subdivision process, the subdomains in which the topology of the bisectors and trisectors is easily determined from their intersection points with the boundary of domain  $\mathcal{D}$ .

However the degenerate intersections could be avoided during the subdivision either by subdividing the domain not quite in the middle of the direction, or by shifting position of the domain slightly.

The bisector surface and trisector curve are monotonous w.r.t. ( $oz$ ) orientation and we have following

**Proposition 1** Let  $\mathcal{D} \in \mathbb{R}^3$  be a compact domain.

1. The bisector surface  $b_{i,j}$  with generic intersection with the domain  $\mathcal{D}$  is  $t$ -regular in  $\mathcal{D}$ .
2. The trisector curve  $\mathcal{T}_{i,j,k}$  with generic intersection with the domain  $\mathcal{D}$  is  $t$ -regular in  $\mathcal{D}$ .

*Proof* 1. Prove that the bisector surface is  $t$ -regular in the domain  $\mathcal{D}$ :

Differentiating the algebraic equation (8) of the bisector we obtain:

$$\nabla(g_i - g_j)(x, y, z) = \begin{cases} (2(x_j - x_i), 2(y_j - y_i), 2(z_j - z_i)) & \text{if } z < z_i \\ (2(x_j - x_i), 2(y_j - y_i), 2(z - z_j)) & z_i \leq z < z_j \\ (2(x_j - x_i), 2(y_j - y_i), 0) & \text{if } z \geq z_j \end{cases} \quad (15)$$

(a) Since  $(x_i, y_i) \neq (x_j, y_j)$ , from (15) we can verify easily one of the conditions:

$$\partial_x(g_i - g_j) \neq 0, \quad \partial_y(g_i - g_j) \neq 0 \quad \text{or} \quad \partial_z(g_i - g_j) \neq 0;$$

- (b) The bisector surface  $b_{i,j}$  consists of three connected pieces of surfaces: (two half-planes and a half-parabolic-cylinder piece). Since  $b_{i,j}$  has a generic intersection with  $\mathcal{D}$  its intersection with the face  $\mathcal{F}(\mathcal{D})_x, \mathcal{F}(\mathcal{D})_y$  or  $\mathcal{F}(\mathcal{D})_z$  is a line segment, parabolic segment or a junction of two, which are  $t$ -regular on the faces; its intersection with each edge  $\mathcal{D}$  is at most one point; and the part of  $b_{i,j}$  delimited by the intersection points with the edges is contained in  $\mathcal{D}$ .

From (1a) and (1b) we can conclude that the bisector surface  $b_{i,j}$  is  $t$ -regular in  $\mathcal{D}$  and its topology is determined by a polyhedra delimited in  $\mathcal{D}$  by the intersection points with the edges.

2. Prove that the trisector curve is  $t$ -regular in the domain  $\mathcal{D}$ : The trisector curve is piecewise smooth, and its tangent vector is given by:

$$\mathcal{T}'_{i,j,k}(z) = \begin{cases} \left[ \frac{a_{3,1}}{c_0}, -\frac{a_{1,1}}{c_0}, 1 \right]^T & \text{if } z < z_i, \\ \left[ \frac{a_{4,2}+2a_{5,2}z}{c_0}, -\frac{a_{1,2}+2a_{2,2}z}{c_0}, 1 \right]^T & \text{if } z_i \leq z < z_j, \\ \left[ -\frac{a_{4,3}+2a_{5,3}z}{c_0}, \frac{a_{1,3}+2a_{2,3}z}{c_0}, 1 \right]^T & \text{if } z_j \leq z < z_k, \\ [0, 0, 1]^T & \text{if } z \geq z_k, \end{cases}$$

For  $z \in \mathbb{R}$ ,  $\mathcal{T}'_{i,j,k}(z) \neq (0, 0, 0)$ .

For  $z_0 \in \mathbb{R}$ , the plane  $(z = z_0)$  intersects the trisector curve at exactly a unique point then taking into account the Proposition 4.1 in [3] and since  $\mathcal{T}_{i,j,k}$  has generic intersection with  $\mathcal{D}$  we can conclude that the trisector curve is  $t$ -regular in  $\mathcal{D}$  and its topology determined by the segment delimited in  $\mathcal{D}$  by the two intersections points with faces, and the proof is complete.

Since in our context we are interested in computing the topology of VD faces (bisectors), VD edges (trisectors) and VD vertices (quadrisections) in a domain  $\mathcal{D}$ , any  $d$ -list  $B(\mathcal{D})$  with more than four distance functions is naturally not  $b$ -regular, hence the  $b$ -regularity is effectively checked for a  $d$ -list of at most four distance functions as follows:

**Regularity Criterion 1** Let  $\mathcal{D} \subset \mathcal{D}_0$  and  $B(\mathcal{D}) = \{g_{i_{\mathcal{D}}}, i = 1, \dots, |B(\mathcal{D})|\}$

- A  $d$ -list  $B(\mathcal{D})$  with one active distance  $g_i$  (i.e:  $\mathcal{D} \subset \mathcal{V}\mathcal{D}(i)$ ) is trivially  $b$ -regular.
- A  $d$ -list  $B(\mathcal{D})$  of two active distances  $g_i$  and  $g_j$  is  $b$ -regular if the bisector  $b_{i,j}$  is  $t$ -regular in  $\mathcal{D}$ .
- A  $d$ -list  $B(\mathcal{D})$  of three active distances  $g_i, g_j$  and  $g_k$  is  $b$ -regular if:
  1. each one of the three bisectors  $b_{i,j}, b_{i,k}$  and  $b_{j,k}$ , taken individually, is  $t$ -regular in  $\mathcal{D}$ ;
  2. the trisector  $\mathcal{T}_{i,j,k}$  is  $t$ -regular in  $\mathcal{D}$ .
- A  $d$ -list  $B(\mathcal{D})$  of four active distances  $g_i, g_j, g_k$  and  $g_l$ , is  $b$ -regular if:
  1. each one of the six bisectors  $b_{m,n}$ ,  $m < n \in \{i, j, k, l\}$ , taken individually, is  $t$ -regular in  $\mathcal{D}$ ;
  2. each one of the four trisectors  $\mathcal{T}_{m,n,p}$ ,  $m < n < p \in \{i, j, k, l\}$ , taken individually, is  $t$ -regular in  $\mathcal{D}$ ;
  3. the corresponding quadrisection  $Q_{i,j,k,l}$  has a generic intersection  $\mathcal{D}$ , i.e:  $Q_{i,j,k,l}$  is strictly inside  $\mathcal{D}$ .
- Any  $B(\mathcal{D})$  such that  $|B(\mathcal{D})| > 4$ , is not  $b$ -regular.



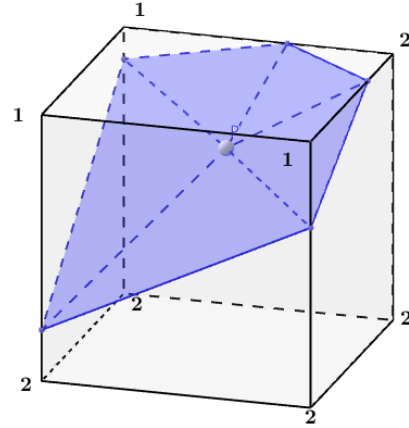
## 4 Computing the Topology of Bisector and Trisector in a domain $\mathcal{D}$

### 4.1 Meshing a domain $\mathcal{D}$ with two active sites

For a domain  $\mathcal{D}$  with two active sites  $h_i$  and  $h_j$ , such that  $B(\mathcal{D})$  is  $b$ -regular, the topology of the corresponding bisector will be computed as follows:

- Compute the intersection points of bisectors with the edges  $\mathcal{E}(\mathcal{D})$  of the domain  $\mathcal{D}$ :  $B_{i,j} = b_{i,j} \cap \mathcal{E}(\mathcal{D})$ , and its barycentric point  $B$ .
- Then connect the points  $B_{i,j}$  in a star-shaped set of triangles centered at the point  $B$ .

We obtain a mesh which is isotopic to the bisector  $b_{i,j}$  in the domain  $\mathcal{D}$ . An illustration is given in Fig. 1.



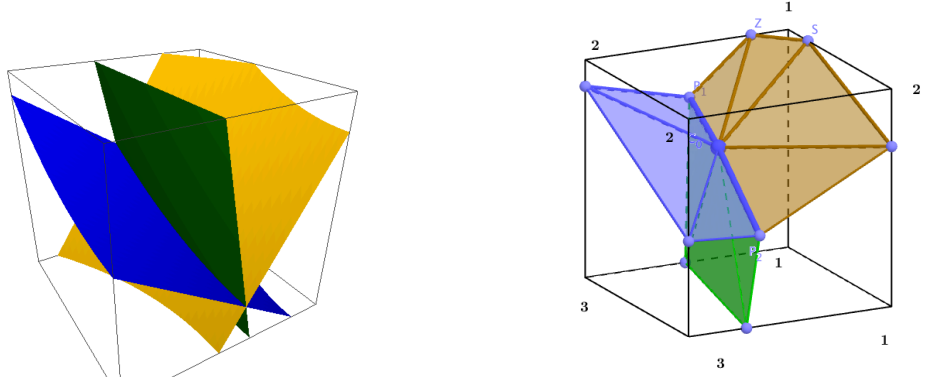
**Fig. 1** The bisector surface of two sites and its topology in a domain  $\mathcal{D}$ , respectively in the first and second pictures.

#### 4.2 Meshing a domain $\mathcal{D}$ with three active sites

For a domain  $\mathcal{D}$  with three active sites  $h_i, h_j$  and  $h_k$ , such that  $B(\mathcal{D})$  is  $b$ -regular, the topology of the corresponding involved parts of the three bisectors and trisector will be computed as follows:

- Compute the two intersection points of the trisector with the faces  $\mathcal{F}(\mathcal{D})$  of the domain  $\mathcal{D}$ :  
 $\{T_1, T_2\} = \mathcal{T}_{i,j,k} \cap \mathcal{F}(\mathcal{D})$ ,  $T_1 = \mathcal{T}_{i,j,k}(t_1)$ ,  $T_2 = \mathcal{T}_{i,j,k}(t_2)$ ,  
 and their middle point  $T = \mathcal{T}_{i,j,k}(\frac{t_1+t_2}{2})$ .
- Compute the intersection points of the three bisectors with the edges  $\mathcal{E}(\mathcal{D})$  of the domain  $\mathcal{D}$ :  
 for  $\alpha \leq \beta \in \{i, j, k\}$ :  $\begin{cases} B_{\alpha,\beta} = b_{\alpha,\beta} \cap \mathcal{E}(\mathcal{D}) \\ \text{such that:} \\ \forall \kappa \neq \alpha, \beta \in \{i, j, k\}, g_{\kappa}(B_{\alpha,\beta}) > g_{\alpha}(B_{\alpha,\beta}) = g_{\beta}(B_{\alpha,\beta}) \end{cases}$
- Then connect the points  $B_{\alpha,\beta}, T_1$  and  $T_2$  in a star-shaped set of triangles centered at the point  $T$ .

We obtain a mesh which is isotopic to the part of three bisectors involved the VD in the domain  $\mathcal{D}$ . An illustration is given in Fig. 2.



**Fig. 2** The three bisector surfaces and the topology of corresponding part of the VD in a domain  $\mathcal{D}$ , respectively in the first and second pictures.

#### 4.3 Meshing a domain $\mathcal{D}$ with four active sites

For a domain  $\mathcal{D}$  with four active sites  $h_i, h_j, h_k$  and  $h_l$ , such that  $B(\mathcal{D})$  is  $b$ -regular, the topology of the corresponding involved parts of six bisectors and three trisectors will be computed as follows:

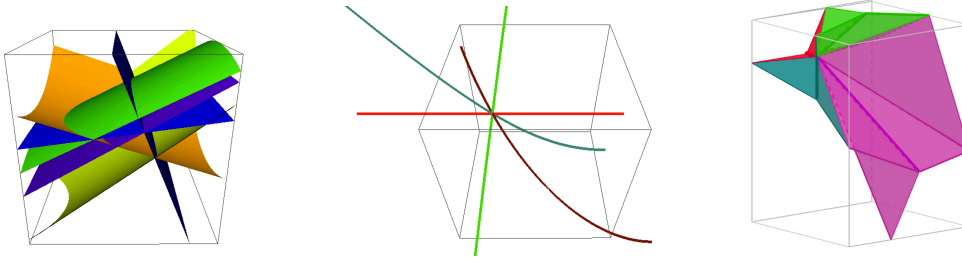
- Compute the quadrisector point:  $Q_{i,j,k,l} = \mathcal{T}_{i,j,k} \cap \mathcal{T}_{i,j,l} \cap \mathcal{T}_{j,k,l} \cap \mathcal{T}_{i,k,l}$  (**VD Vertex** in  $\mathcal{D}$ )
- Compute the intersection points of the trisectors with the faces  $\mathcal{F}(\mathcal{D})$  of the domain  $\mathcal{D}$ :  
 for  $\alpha \leq \beta \leq \gamma \in \{i, j, k, l\}$   $\begin{cases} P_{\alpha,\beta,\gamma} := \mathcal{T}_{\alpha,\beta,\gamma} \cap \mathcal{F}(\mathcal{D}) \\ \text{such that:} \\ \forall \kappa \neq \alpha, \beta, \gamma, g_{\kappa|_{\mathcal{D}}}(P_{\alpha,\beta,\gamma}) > g_{\alpha|_{\mathcal{D}}}(P_{\alpha,\beta,\gamma}) = g_{\beta|_{\mathcal{D}}}(P_{\alpha,\beta,\gamma}) = g_{\gamma|_{\mathcal{D}}}(P_{\alpha,\beta,\gamma}) \end{cases}$

- Compute the intersection points of the bisectors with the edges  $\mathcal{E}(\mathcal{D})$  of  $\mathcal{D}$  :

$$\text{for } \alpha \leq \beta \in \{i, j, k, l\} \left\{ \begin{array}{l} B_{\alpha, \beta} := b_{\alpha, \beta} \cap \mathcal{E}(\mathcal{D}) \\ \text{such that:} \\ \forall \kappa \neq \alpha, \beta, g_{\kappa|\mathcal{D}}(B_{\alpha, \beta}) > g_{\alpha|\mathcal{D}}(B_{\alpha, \beta}) = g_{\beta|\mathcal{D}}(B_{\alpha, \beta}) \end{array} \right.$$

- Then connect the points  $P_{\alpha, \beta, \gamma}$  and  $B_{\alpha, \beta}$  in a star-shaped set of triangles centered at the point  $Q_{i, j, k, l}$ .

We obtain a mesh which is isotopic to the part of six bisectors involved the VD in the domain  $\mathcal{D}$ . An illustration is given in Fig. 3.



**Fig. 3** The six bisector surfaces , the four trisector curves and the topology corresponding part of the VD in a domain  $\mathcal{D}$ , respectively in the first, second and first pictures.

## 5 Algorithm

The approach consists of two main parts: the subdivision phase of the initial domain  $\mathcal{D}_0$  and the reconstruction phase of VD cells constrained to  $\mathcal{D}_0$ :

- During the subdivision phase, using the regularity criteria (section 3) for the topology of algebraic curves and surfaces, the initial domain  $\mathcal{D}_0$  will be subdivided into subdomains following  $kd$ -trees structure ([8]), i.e.  $\mathcal{D}_0$  will be, firstly, subdivided in two subdomains by a bisector plane which is perpendicular to one of the directions of the coordinate system  $(O, \vec{i}, \vec{j}, \vec{k})$ . Then, each one of the two subdomains is subdivided recursively in a different direction. An **adjacency graph**  $\mathcal{G}$  of the subdomains spanning VD cells is computed. The graph  $\mathcal{G}$  is constructed using similar method used in the Dual Marching Cube Method (see [36]) and it ensures the connectivity of triangles in each subdomain during the reconstruction phase.
- The reconstruction phase is performed by traversing the adjacency graph  $\mathcal{G}$  while meshing all involved parts of the bisector surfaces and trisectors curves in each subdomain.

**Subdivision** Let  $\mathcal{D}_0 = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$  a domain in which the Voronoï diagram of the  $n$  sites  $h_1, \dots, h_n$ , will be computed and a threshold  $\varepsilon > 0$ .

The domain  $\mathcal{D}_0$  will be subdivided into subdomains, and all subdomains spanning VD cells (i.e: subdomain for which the  $d$ -list is **b-regular**) will be identified and the corresponding lists of the bisectors and trisectors, will be computed and sorted into adjacency graph. We use interval arithmetic to compute the **upper- $d$ -list**  $\tilde{B}(\mathcal{D})$  containing  $B(\mathcal{D})$ , i.e.  $B(\mathcal{D}) \subset \tilde{B}(\mathcal{D})$ .

1. The process begin with the initial  $d$ -list  $\tilde{B}(\mathcal{D}_0) := \{g_{1|\mathcal{D}_0}, \dots, g_{n|\mathcal{D}_0}\}$ : the domain  $\mathcal{D}_0$  carrying all associated distance functions of the  $n$  sites.
2. Then we initialize
  - a stack  $\mathcal{P}$  by the  $d$ -list  $\tilde{B}(\mathcal{D}_0)$  and
  - the adjacency graph  $\mathcal{G}$  with empty graph, between the subdomains.
3. The principal step starts with a loop by taking a  $d$ -list  $\tilde{B}(\mathcal{D})$  from the stack  $\mathcal{P}$ , using interval arithmetic, the distance functions are bounded over the subdomain  $\mathcal{D}$ , i.e.:

$$m_{\mathcal{D}}^i \leq g_{i|\mathcal{D}} \leq M_{\mathcal{D}}^i, i = 1, \dots, |\tilde{B}(\mathcal{D})|.$$

All distance functions which are not active in  $\mathcal{D}$  are filtered out, i.e.: all distance  $g_{i_0|\mathcal{D}}$  verifying

$$m_{\mathcal{D}}^{i_0} \geq \min\{M_{\mathcal{D}}^i, i = 1, \dots, |\tilde{B}(\mathcal{D})|\},$$

will be eliminated from the  $d$ -list  $\tilde{B}(\mathcal{D})$ :

$$\tilde{B}(\mathcal{D}) := \tilde{B}(\mathcal{D}) \setminus \{g_{i_0|\mathcal{D}}\}.$$

The regularity test is applied to the resulting  $d$ -list:

- If the  $d$ -list  $\tilde{B}(\mathcal{D})$  is  $b$ -regular or if the threshold size on the subdomain is reached (i.e.:  $|\mathcal{D}| = \max_{1 \leq i \leq 3} (|b_i - a_i|) < \varepsilon$ ), the  $d$ -list is transformed to a list of the corresponding bisectors and trisectors in  $\mathcal{D}$ :

$$B^{vd}(\mathcal{D}) := \{b_{i,j}, \mathcal{T}_{i,j,k}, \forall i < j < k \in \{1, \dots, |B(\mathcal{D})|\}\},$$

and it will be placed in the graph  $\mathcal{G}$ .

- Else, the domain  $\mathcal{D}$  will be subdivided in two subdomains  $\mathcal{D}_1$  and  $\mathcal{D}_2$  by a bisector plane which is perpendicular to the direction of longest edge of  $\mathcal{D}$ . The adjacency graph  $\mathcal{G}$  will be updated by replacing the  $d$ -list  $\tilde{B}(\mathcal{D})$  by the two new corresponding  $d$ -lists  $\tilde{B}(\mathcal{D}_1)$  and  $\tilde{B}(\mathcal{D}_2)$ , by adding an adjacency edge between them and connecting them with their neighbour  $d$ -lists. Finally they are returned back to the stack  $\mathcal{P}$ , and the process continues until the stack is empty.

The Algorithm is summarized as follows:

---

### Algorithm 1 Subdivision

---

**Input**  $\mathcal{D}_0$  a domain,  $g_1, \dots, g_n \in \mathbb{R}[x, y, z]$  the associated distances to  $h_1, \dots, h_n$  and  $\varepsilon > 0$  a threshold.

**Output**  $\mathcal{G}$  the adjacency graph of the domains spanning the MD cell of  $\{g_1, \dots, g_n\}$  in  $\mathcal{D}_0$ .

1.  $\tilde{B}(\mathcal{D}_0) := \{g_{1|\mathcal{D}_0}, \dots, g_{n|\mathcal{D}_0}\}$
2. **Initialize** a stack  $\mathcal{P} := \{\tilde{B}(\mathcal{D}_0)\}$ ; a graph  $\mathcal{G} := \emptyset$ ;
3. **While**  $\mathcal{P} \neq \emptyset$  **do**
  - (a) **pick**  $\tilde{B}(\mathcal{D})$  from  $\mathcal{P}$
  - (b) **filter**  $\tilde{B}(\mathcal{D})$ :

---

```

i. compute  $M_{\mathcal{D}} := \min\{\max(g_{i|\mathcal{D}}), i = 1 \dots, |\tilde{B}(\mathcal{D})|\};$ 
ii. for  $i = 1 \dots |\tilde{B}(\mathcal{D})|$  do
    if  $\min(g_{i|\mathcal{D}}) \geq M_{\mathcal{D}}$ 
        then  $\tilde{B}(\mathcal{D}) := \tilde{B}(\mathcal{D}) \setminus \{g_{i|\mathcal{D}}\};$ 
    end if;
iii. end for;
(c) if  $\tilde{B}(\mathcal{D})$  IsRegular or  $|\mathcal{D}| < \varepsilon$ 
    then compute  $B^{vd}(\mathcal{D}) := \{b_{i,j}, \mathcal{T}_{i,j,k}, b_{i,j} = g_{j|\mathcal{D}} - g_{i|\mathcal{D}}, \forall i \leq j \leq k \in \{1, \dots, |\mathcal{B}(\mathcal{D})|\}\}$ 
    and add it to  $\mathcal{G}$ ;
(d) else
    i. subdivide  $\mathcal{D}$  into  $\mathcal{D}_1$  and  $\mathcal{D}_2$ ;
    ii. update  $\mathcal{G}$  with  $\tilde{B}(\mathcal{D}_1)$  and  $\tilde{B}(\mathcal{D}_2)$ ;
    iii.  $\mathcal{D} \leftarrow \mathcal{D} \cup \{\tilde{B}(\mathcal{D}_1), \tilde{B}(\mathcal{D}_2)\}$ ;
(e) end if;
4. End While;
5. Return  $\mathcal{G}$ ;

```

---

At the end of the subdivision process, we obtain an adjacency graph  $\mathcal{G}$  of the identified subdomains  $\mathcal{D}$  with their corresponding lists of bisectors and trisectors, and the adjacencies are represented by the edges. These subdomains  $\mathcal{D}$  could contain none, one bisector or three bisectors and one trisector, six bisectors and four trisectors and one quadrisector, in which the topology will be easily determined from the intersection points with the boundary. However the  $d$ -lists encountered when the threshold size is reached for  $\mathcal{D}$ , might span degenerate VD edges and/or degenerate VD vertices.

**Reconstruction VD cells** In this last part, we describe the reconstruction of the VD cells by traversing the adjacency graph  $\mathcal{G}$  and meshing all lists of bisectors and trisectors in the corresponding identified subdomains from the subdivision.

- The construction of the graph  $\mathcal{G}$  is similar to the one used in the Dual Marching Cube method (see [36]). It identifies the faces ( of subdomains) that are adjacent. it ensures connectivity of triangles in each subdomain and avoiding blunders at the end of the mesh.
- The list of bisectors and trisectors is meshed in the corresponding subdomain using the method developed in the section 4.

The topology of bisectors and trisectors is thus correctly determined not only within, but also between (i.e: on faces and edges of) the adjacent subdomains traversed. All open VD cells are completed by the boundary pieces  $\partial\mathcal{D}_0$  such that the reconstruction will be constrained in the domain  $\mathcal{D}_0$ . At the end we obtain a polyhedral approximation of each VD cell, which is isotopic to VD cell in the domain  $\mathcal{D}_0$ .

*Remark 2* The approximate VD computed is topologically equivalent to exact VD when the  $d$ -list list  $\tilde{B}(\mathcal{D})$  is regular or when  $|\mathcal{D}| < \varepsilon$  and the corresponding  $d$ -list  $\tilde{B}(\mathcal{D})$  spans exactly unique VD edge or unique VD vertex. This happen for  $\varepsilon > 0$  suitably. In the other case the topology is unknown.

The same process for computing the topology of VD faces and VD edge in the regular case can be used in the case of degenerate VD edge and degenerate VD vertex.

As analyzed in [2], the number of boxes produced to cover the boundary of the Voronoï cells is in  $\mathcal{O}(\frac{1}{\varepsilon^2})$ .

## 6 Implementation

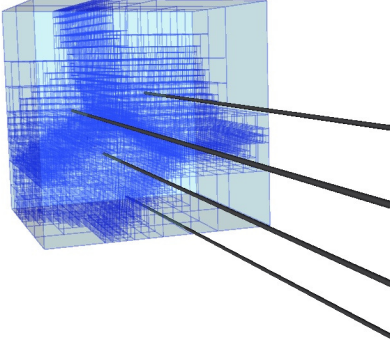
The preliminary step of the implementation consists in implementing the function called **closest-site** which for each domain, it computes the minimum distance from each of its corners to the different sites  $h_1, \dots, h_n$  and assigns to corners of a domain the index of the site which is closer to it than to any other, i.e: the active sites in the domain. This function is very efficient and allows us very quickly and easily to identify, during the subdivision, the b-regular d-lists  $B(\mathcal{D})$  from the number of active sites in a domain. It is thus possible to detect the edges and the faces of a domain which intersect the equidistant points respectively of two and three sites which will be used for the mesh of the VD cells. This is how the subdivision phase is performed by the function **subdivision**, and the reconstruction by the functions **mesher**. The code of the different functions established for this implementation is developed in Julia language and follows the same structure as the theoretical part and it is available at <https://github.com/ibradam/Voronoi.jl>.

### 6.1 Examples

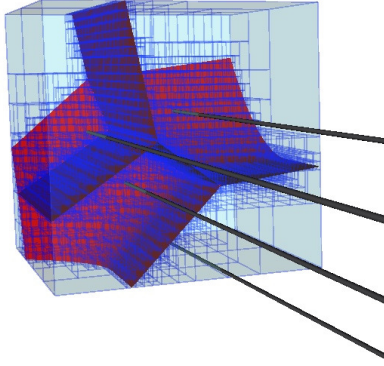
1. Let  $\mathcal{D}_0 = [-5, 10] \times [-5, 10] \times [-5, 10]$  and four sites:  $h_1 = (-2.0, 3.0, 5.0 + t)$ ,  $h_2 = (4.0, -4.0, -5.0 + t)$ ,  $h_3 = (3.0, 4.0, 5.0 + t)$ ,  $h_4 = (0.0, 0.0, 5.0 + t)$ ,  $t \geq 0$
2. Let  $\mathcal{D}_0 = [-10, 10] \times [-10, 10] \times [-10, 20]$  and ten sites:  $h_1 = (4.0, -4.0, -3.0 + t)$ ,  $h_2 = (-2.0, 3.0, -5.0 + t)$ ,  $h_3 = (2.0, 2.0, 5.0 + t)$ ,  $h_4 = (0.0, 0.0, 0.0 + t)$ ,  $h_5 = (1.0, 1.0, 6.0 + t)$ ,  $h_6 = (-1.0, 0.0, 8.0 + t)$ ,  $h_7 = (-3.0, -7.0, 1.0 + t)$ ,  $h_8 = (-5.0, 7.0, -1.0 + t)$ ,  $h_9 = (3.0, 7.0, -2.0 + t)$ ,  $h_{10} = (3.0, 4.0, 3.0 + t)$ ,  $t \geq 0$ .

Applying the algorithm, we have successively:

1. For example 1
  - (a) The subdivision phase is given in Figure 4. There is certain concentration of subdomains between the sites, spanning the Voronoï diagram cells.
  - (b) The meshing result is shown in Figure 5. All bisectors and trisectors are meshed in subdomains spanning the Voronoï diagram cells. We have a red-colored polyhedric structure which represents the topology of the Voronoï diagram of the four sites in  $\mathcal{D}_0$ .
  - (c) The topology of the VD of  $h_1, h_2, h_3, h_4$  in  $\mathcal{D}_0$  is given in Figure 6. The topology of the Voronoï diagram is obtained by eliminating the trace of the subdomains resulting from the subdivision and leaving only the mesh and sites, which gives the partition of  $\mathcal{D}_0$  into four (4) cells.
2. For example 2
  - (a) The subdivision phase is given in Figure 7. There is certain concentration of subdomains concentration between the sites, thus spanning the Voronoï diagram cells.
  - (b) The meshing phase is given in Figure 8. All bisectors and trisectors are meshed in subdomains spanning the Voronoï diagram cells. We, thus, have a red-colored polyhedric structure which represents the topology of the Voronoï diagram of the ten sites in  $\mathcal{D}_0$ .
  - (c) The the topology of the VD of  $h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, h_{10}$  in  $\mathcal{D}_0$  is given in Figure 9. The topology of the Voronoï diagram is obtained by eliminating the trace of the subdomains resulting from the subdivision and leaving only the mesh and sites, which gives the partition of  $\mathcal{D}_0$  into ten (10) cells.



**Fig. 4** Subdivision process for the domain  $\mathcal{D}_0$  for the example 1.



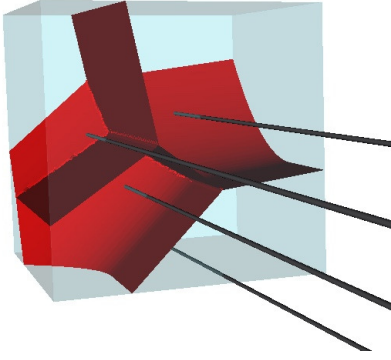
**Fig. 5** Meshing process of bisectors and trisectors in (in red color) subdomain spanning VD for the example 1.

Sites	Regular boxes	Singular boxes	Times (secs)
10	863	25	0.07
20	1362	86	0.12
40	1810	309	0.23
60	1720	629	0.29
120	2282	1345	0.63

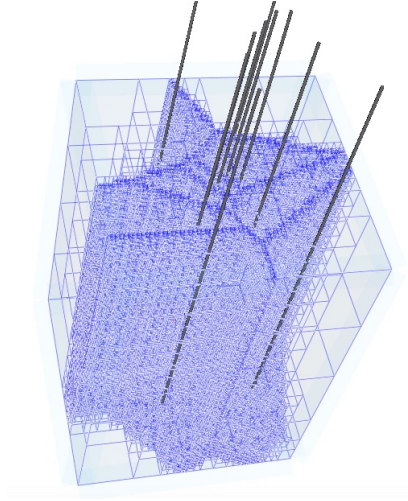
**Table 1**  $\mathcal{D} = [-2, 2] \times [-2, 2] \times [-3, 3]$ ,  $\sigma = 0.1$

## 6.2 Experimentations of algorithm

We give the experimentations of the algorithm for a number of random of inputs 10 up to 120 parallel half-lines with endpoints strictly in the domain  $\mathcal{D} = [-2, 2] \times [-2, 2] \times [-3, 3]$  under Fedora 27 System intel CORE i7 7th Gen. The execution time and the numbers of boxes identified during subdivision process is respectively given in Table 1 and Table 2 for  $\sigma = 0.1$  and  $\sigma = 0.01$ , where  $\sigma$  is the maximal relative size of the domains at the end of the subdivision process. It corresponds roughly to the relative size of the triangles in the output mesh. The threshold  $\varepsilon$  used to determine the singular domains is  $\varepsilon = \frac{\sigma}{2}$ .



**Fig. 6** The topology of the VD  $h_1, h_2, h_3, h_4$  in  $\mathcal{D}_0$  for the example 1



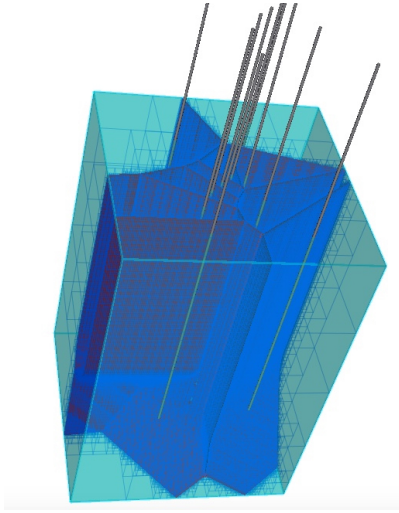
**Fig. 7** Subdivision process for the domain  $\mathcal{D}_0$  for the example 2.

Sites	Regular boxes	Singular boxes	Times (secs)
10	84226	7	103.42
20	126557	31	252.04
40	163193	206	427.45
60	179887	127	593.05
120	267794	556	1620.21

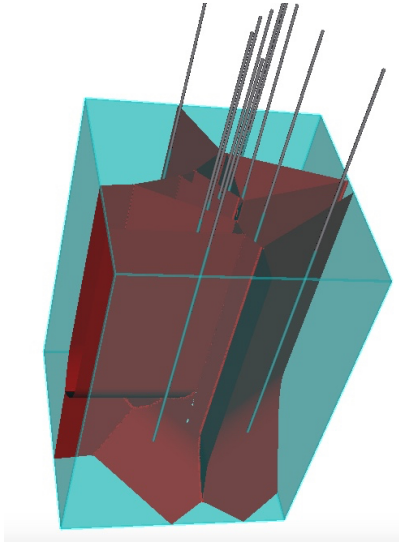
**Table 2**  $\mathcal{D} = [-2, 2] \times [-2, 2] \times [-3, 3]$ ,  $\sigma = 0.01$

The analysis of the timing for different sizes of the input (number of sites) allows to estimate the practical complexity of the algorithm. The execution time and the number of boxes produced evolve linearly in the number of sites. Moreover, for a given number of sites, the number of produced boxes and the execution time evolve according to  $\sigma$ , more precisely they varie respectively in  $\mathcal{O}\left(\frac{1}{\sigma^2}\right)$  and  $\mathcal{O}\left(\frac{1}{\sigma^3}\right)$ .





**Fig. 8** Meshing process of bisectors and trisectors in (in red color) subdomain spanning VD for in the example 2



**Fig. 9** The topology of the VD  $h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9, h_{10}$  in  $\mathcal{D}_0$  for the example 2

Note that scaling to much larger input can be envisaged by exploiting parallel computation, since the subdivision and meshing process can be parallelized easily.

## 7 Conclusion and Future Work

We presented a new algorithmic approach, based on subdivision process and meshing algebraic curves and surfaces for computing an approximative Voronoï diagram of parallel half-lines, constrained to a compact

domain  $\mathcal{D}_0 \subset \mathbb{R}^3$ , with respect to the Euclidean distance. The algorithm developed uses some tolerance for computing an approximation of the VD, which is topologically equivalent to the exact one in  $\mathcal{D}_0$  when the size of all subdomain  $\mathcal{D}$  is not smaller than the subdivision threshold  $\varepsilon$  (i.e.:  $|\mathcal{D}| \geq \varepsilon$ ) or when  $|\mathcal{D}| < \varepsilon$  and the corresponding list  $B(\mathcal{D})$  spans exactly unique VD edge or unique VD vertex, for  $\varepsilon > 0$  suitably choose. In other cases, an advanced method have to be developed for guarantying correct topology of the VD. The approach can be applied to a compact domain of any metric space, in which, the distance function is transformable in a polynomial expression. We also described its implementation in Julia Language and the experimentation allows an effective analysis of the approach.

In the future, we plan to extend the approach to the case of non-Euclidean distance, non parallel half-lines, or some specific rational curves (such as rational curves with rational offsets) which will have very interesting applications such as motion planning, collision avoiding in minning industry, in robotic, etc.

**Acknowledgements** Most of the implementation part of the work was done during research visits at the research center INRIA Sophia Antipolis with partial funding from the Simons Foundation and INRIA.

## References

1. Adamou, I., Fioravanti, M., Gonzalez-Vega, L., Mourrain, B. : *Bisectors and Voronoi Diagram of a Family of Parallel Half-Lines*. In: Dokken, T., Muntingh, G. (eds) SAGA – Advances in ShApes, Geometry, and Algebra. Geometry and Computing, Springer, Cham., 2014, 241–279.
2. Alberti, L., Comte, G., Mourrain, B. : *Meshing implicit algebraic surfaces: the smooth case*. In L.L. Schumaker M. Maehlen, K. Morken, editor, Mathematical Methods for Curves and Surfaces: Tromsø'04, pages 11–26. Nashboro, 2005.
3. Alberti, L., Mourrain, B. : *Regularity criteria for the topology of algebraic curves and surfaces*. In Martin, R., Sabin, M., Winkler, J. (eds.) Proceedings of IMA international conference on Mathematics of Surfaces XII, Springer-Verlag Berlin, Heidelberg, 2007, 1–28.
4. Aurenhammer, F., Jüttler, B., Paulini, G. : *Voronoi Diagrams for Parallel Halflines and Line Segments in Space*, in Proc. 28th Int Symp. Algorithms and Comput., Okamoto, Y., Tokuyama, T., eds. vol. 92, pp. 7:1–7:10, 2017.
5. Aurenhammer, F., Klein, R. : *Voronoi diagrams*, in Handbook of computational geometry, J.-R. Sack, J. Urrutia, eds., North-Holland, Amsterdam, 2000, 201–290.
6. Aurenhammer, F., Paulini, G., Jüttler, B. : *Voronoi diagrams for parallel halflines in 3D*. In Proc. 32nd European Workshop on Computational Geometry, 2016, 127–130.
7. Barros, P.G., Pessoa, D.A., Leite, P.J.S., Teichrieb, V., Kelner, J. : *Three dimensional Oil Well Planning in Ultra-deep Water*. In Proceedings of Symposium on Virtual Reality (SVR), 2006, Belém, Pará. v. 1, 285–296.
8. Bentley, J.L. : *Multidimensional binary search trees used for associative searching*: Commun. ACM, 18, 509–517, 1975.
9. Boada, I., Coll, N., Madern, N., Antoni Sellares, J. : *Approximations of 2d and 3d generalized voronoi diagrams*, Int. J. Comput. Math., 85(7), 1003–1022, 2008.
10. Boada, I., Coll, N., Sellares, J. : *The voronoi quadtree: construction and visualization*. Eurographics 2002 Short Presentation 2002, pp. 349–355.
11. Boissonnat, J.-D., Wormser, C., Yvinec, M. : *Curved voronoi diagrams*. In Boissonnat, J.-D., Teillaud, M. (eds.) Effective Computational Geometry for Curves and Surfaces, Springer Berlin Heidelberg, 2006, 67–116.
12. Cao, T.-T., Tang, K., Mohamed, A., Tan, T.-S. : *Parallel banding algorithm to compute exact distance transform with the gpu*. In Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games (2010), ACM, pp. 83–90.
13. Edelsbrunner, H., Seidel, R. : *Voronoi diagrams and arrangements*. Discrete and Computational Geometry, 1:25–44, 1946.
14. Edwards, J., Daniel, E., Pascucci, V., Bajaj, C. : *Approximating the Generalized Voronoi Diagram of Closely Spaced Objects*, Computer graphics forum : journal of the European Association for Computer Graphics., vol. 34, no. 2, Dec. 2015, 299–309, doi:10.1111/cgf.12561.
15. Emiris, I., Malamatos, T., Tsigaridas, E. : *Approximate nearest neighbor queries among parallel segments*. Proc. 26th European Workshop on Computational Geometry, 2010.
16. Emiris, I., Mantzaflaris, A., Mourrain, B. : *Yet Another algorithm for generalized Voronoi Diagrams*, Proceedings of the 27th Annual ACM Symposium on Applied Computing, 2012, 109–110.
17. Emiris, I.Z., Tsigaridas, E.P., Tzoumas, G.M. : *Exact Delaunay graph of smooth convex pseudo-circles: general predicates, and implementation for ellipses*, SIAM/ACM Joint Conference on Geometric and Physical Modeling, October 2009, San Francisco, California.

18. Etzion, M., Rappoport, A. : *Computing voronoi skeletons of a 3-d polyhedron by space subdivision*. Computational Geometry: Theory and Applications, 21(3): 87-120, 2002.
19. Everett, H., Gillot, C., Lazard, D., Lazard, S., Pouget, M.: *The Voronoi diagram of three arbitrary lines  $\mathbb{R}^3$* . In: Abstracts of 25th Eur. Workshop Comput. Geom. 2009.
20. Fan, C., Luo, J., Liu, J., Xu, Y. : *Half-plane Voronoi Diagram*. In Proceedings of the 2011 Eighth International Symposium on Voronoi Diagrams in Science and Engineering, 127–133.
21. Fortune, S. : *Voronoi diagrams and Delaunay triangulations*, in Handbook of Discrete and Computational Geometry, Goodman, J. E., O'Rourke, J. (eds.), CRC Press LLC, Boca Raton, FL, 1997, 377–388.
22. Fredrich, J.T., Coblenz, D., Fossum, A.F., Thorne, B.J. : *Stress Perturbations Adjacent to Salt Bodies in the Deepwater Gulf of Mexico*, SPE Annual Technical Conference and Exhibition, 5-8 October 2003, Denver, Colorado.
23. Hanniel, I., Elber, G. : *Computing the Voronoi cells of planes, spheres and cylinders in  $\mathbb{R}^3$* , Computer Aided Geometric Design Vol 26, Issue 6, August 2009, 695–710.
24. Hasan, M., Gavrilova, M.L. : *A Geometric Approach to Drill Path Collision Avoidance*, Proceedings of the 2010 International Symposium on Voronoi Diagrams in Science and Engineering, 244–253.
25. Hoff, K.E. , Keyser, J., Lin, M., Manocha, D., Culver, T. : *Fast computation of generalized Voronoi diagrams using graphics hardware*. Computer Graphics, 33(Annual Conference Series):277–286, 1999. Proceedings of ACM SIGGRAPH 1999.
26. Hemmer, M., Setter, O., Halperin, D. : *Constructing the Exact Voronoi Diagram of Arbitrary Lines in Three-Dimensional Space*. In: de Berg M., Meyer U. (eds) Algorithms-ESA 2010. ESA 2010. Lecture Notes in Computer Science, vol 6346. Springer, Berlin, Heidelberg.
27. Hsieh, H.-H., Tai, W.-K. : *A simple gpu-based approach for 3d voronoi diagram construction and visualization*. Simulation modelling practice and theory 13, 8 (2005), 681–692.
28. The Julia Programming Language, <https://julialang.org>, February 2019.
29. Kim, D., Kim, D.-S. : *Region-expansion for the Voronoi diagram of 3D spheres* Computer-Aided Design 38, 2006, 417–430.
30. Klein, R.: *Lecture Notes in Computer Science: Concrete and Abstract Voronoi Diagrams*, Goos, G., Hartmanis, J. (eds.) Springer-Verlag, 1989.
31. Koltun, V., Sharir, M. : *Three dimensional euclidean Voronoi diagrams of lines with a fixed number of orientations*, SIAM J. Comput. Vol. 32, No. 3, 2003, 616–642.
32. Lafon, J.C., Blanc, G. : *Meshing and Visualization of Oilfields*, IEEE Conference on Information Visualization (IV '97).
33. Ledoux, H. : *Computing the 3D Voronoi Diagram Robustly: An Easy Explanation*, 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007).
34. Liang, C., Mourrain, B., Pavone, J.P. : *Subdivision methods for 2d and 3d implicit curves*. In Jüttler, B., Piene, R. (eds.), Geometric Modeling and Algebraic Geometry, Springer Berlin Heidelberg, 2008, 199–214.
35. Okabe, A., Boots, B., Sugihara, K. : *Spatial Tessellations, Concepts and Applications of Voronoi Diagrams*, John Wiley and Sons, New York, 2000.
36. Schaefer, S., Warren, J. : *Dual Marching Cubes: Primal Contouring of Dual Grids*. 12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings. 2004, 70-76. ISSN: 1550-4085, doi:10.1109/PCCGA.2004.1348336.
37. Seong, J. K., Cohen, E., Elber, G. : *Voronoi diagram computations for planar NURBS curves*, Proceedings of the 2008 ACM symposium on Solid and Physical Modelling, June 02-04, 2008, Stony Brook, New York.
38. Setter, O., Sharir, M., Halperin, D. : *Constructing two-dimensional Voronoi diagrams via divide-and-conquer of envelopes in space*. In Gavrilova, M.L., Tan, C.J.K. (eds.) Transactions on computational science IX, Springer-Verlag, 2010, 1–27.
39. Teichmann, M., Teller, S. : *Polygonal approximation of voronoi diagrams of a set of triangles in three dimensions*. Technical Report 766, Laboratory of Computer Science, MIT, 1997.
40. Thorogood, J.L., Hogg, T.W., Williamson, H.S. : *Application of risk analysis methods to subsurface well collisions*, SPE Drill. Eng., vol. 6, no. 4, Dec. 1991, 299–304, doi:10.2118/23941-PA.
41. Tsao, Y.H., Dawson, C.R., Ure, D.W. : *Well collision avoidance*, U.S. Patent 5 901 795, May 11, 1999.
42. Vleugels, J., Overmars, M : *Approximating voronoi diagrams of convex sites in any dimension*. International Journal of Computational Geometry & Applications Volume 8, Issue 02 1998, pp.201–221.
43. Wang, Z. Inglis, T.A. : *Planning directional wells through a high density cluster of existing wells*, SPE Drill. Eng., vol. 5, no. 4, Dec. 1990, 291–293, doi:10.2118/17594-PA.