



# Rectilinear convex hull of points in 3D and applications

Pablo Pérez-Lantero<sup>1</sup> · Carlos Seara<sup>2</sup> · Jorge Urrutia<sup>3</sup>

Received: 28 September 2022 / Accepted: 11 April 2024 / Published online: 4 May 2024  
© The Author(s) 2024

## Abstract

Let  $P$  be a set of  $n$  points in  $\mathbb{R}^3$  in general position, and let  $RCH(P)$  be the rectilinear convex hull of  $P$ . In this paper we obtain an optimal  $O(n \log n)$  time and  $O(n)$  space algorithm to compute  $RCH(P)$ . We also obtain an efficient  $O(n \log^2 n)$  time and  $O(n \log n)$  space algorithm to compute and maintain the set of vertices of the rectilinear convex hull of  $P$  as we rotate  $\mathbb{R}^3$  around the  $Z$ -axis. We study some combinatorial properties of the rectilinear convex hulls of point sets in  $\mathbb{R}^3$ . Finally, as an application of the obtained results, we show an approximation algorithm to an optimization fitting problem in  $\mathbb{R}^3$ .

**Keywords** Rectilinear convex hull · Rotation around the  $Z$ -axis · Fitting in 3D

**Mathematics Subject Classification** 52C45 · 52C35 · 68U05 · 52A15

## 1 Introduction

We first give the definition and notation of the rectilinear convex hull in two dimensions. Let  $P$  be a set of  $n$  points in the plane. An *open quadrant* in the plane is the intersection of two open halfplanes whose supporting lines are parallel to the  $X$  and  $Y$ -axes. An open quadrant

---

Pablo Pérez-Lantero: Partially supported by project DICYT 042332PL Vicerrectoría de Investigación Desarrollo e Innovación USACH (Chile). Carlos Seara: Partially supported by project PID2019-104129GB-I00 funded by MICIU/AEI/10.13039/501100011033. Jorge Urrutia: Supported in part by SEP-CONACYT of México, Proyecto 80268, and by PAPIIT IN102117 Programa de Apoyo a la Investigación e Innovación Tecnológica, UNAM.

---

✉ Carlos Seara  
carlos.seara@upc.edu

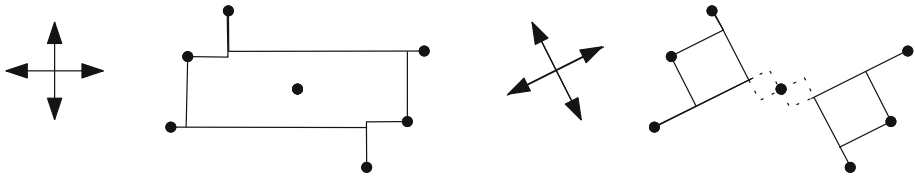
Pablo Pérez-Lantero  
pablo.perez.l@usach.cl

Jorge Urrutia  
urrutia@matem.unam.mx

<sup>1</sup> Facultad de Ciencia, Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile (USACH), Santiago, Chile

<sup>2</sup> Departament de Matemàtiques, Universitat Politècnica de Catalunya, Barcelona, Spain

<sup>3</sup> Instituto de Matemáticas, Universidad Nacional Autónoma de México, Mexico, Mexico



**Fig. 1** Left:  $RCH(P)$ . Right:  $RCH_{\pi/6}(P)$  of the same point set

is called  $P$ -free if it contains no points of  $P$ . The rectilinear convex hull of  $P$  is the set

$$RCH(P) = \mathbb{R}^2 \setminus \bigcup_{W \in \mathcal{W}(P)} W,$$

where  $\mathcal{W}(P)$  denotes the set of all  $P$ -free open quadrants; see Fig. 1, left.

The rectilinear convex hull of point sets has been studied mostly in the plane; e.g., see Ottmann et al. [23], Bae et al. [7], and Alegría-Galicia et al. [1, 2].

An open  $\theta$ -quadrant is the intersection of two open halfplanes whose supporting lines are orthogonal, one of which when rotated clockwise by  $\theta$  degrees becomes horizontal.

We define the  $\theta$ -rectilinear convex hull  $RCH_\theta(P)$  of a point set  $P$  as the set

$$RCH_\theta(P) = \mathbb{R}^2 \setminus \bigcup_{W \in \mathcal{W}_\theta(P)} W,$$

where  $\mathcal{W}_\theta(P)$  denotes the set of all  $P$ -free open  $\theta$ -quadrants; see Fig. 1, right, for  $\theta = \pi/6$ .

Note that  $RCH_\theta(P)$  changes as  $\theta$  changes. In fact, as  $\theta$  changes from 0 to  $\frac{\pi}{2}$  there may be  $O(n)$  combinatorially different rectilinear convex hulls; see [1, 3, 7]. See also the algorithm described in Sect. 3 in Díaz-Bañez et al. [15]. Figure 1 shows an example of a  $\theta$ -rectilinear convex hull that is disconnected when  $\theta = \frac{\pi}{6}$ .

An open octant in  $\mathbb{R}^3$  is the intersection of the three open halfspaces, whose supporting planes are perpendicular to the  $X$ -axis, to the  $Y$ -axis, and to the  $Z$ -axis, respectively. An octant is called  $P$ -free if it contains no elements of  $P$ . The rectilinear convex hull of a set of points in  $\mathbb{R}^3$  is defined as

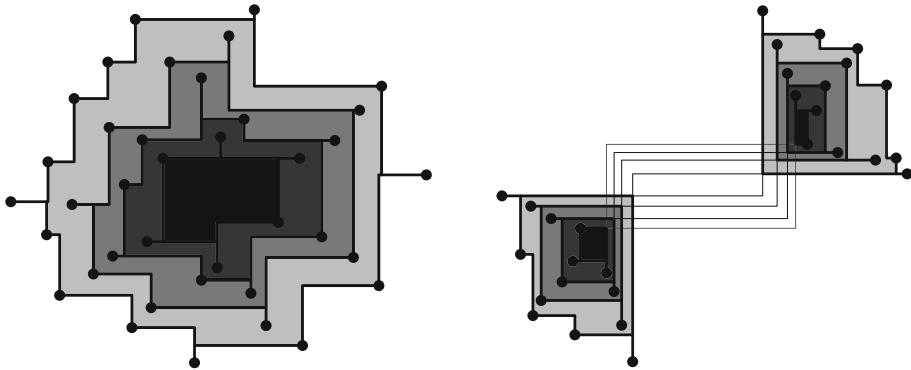
$$RCH^3(P) = \mathbb{R}^3 \setminus \bigcup_{W \in \mathcal{W}(P)} W,$$

where  $\mathcal{W}(P)$  denotes the set of all  $P$ -free open octants of  $\mathbb{R}^3$ .

As far as we know, the rectilinear convex hull of a point set  $P$  in three dimensions,  $RCH^3(P)$ , is constructed here for first time. On the other hand, Kung et al. [22] (see also [26]) compute the set of maxima points of  $P$  in two (resp. three) dimension, i.e., the set of points of  $P$  which are vertices for any of the four  $P$ -free open quadrants (resp. eight  $P$ -free open octants), in optimal  $O(n \log n)$  time and  $O(n)$  space. Thus, knowing the set of these maxima points for each one of the eight octants, we can have the points of  $P$  that are in the boundary of  $RCH^3(P)$ .

The set of maxima points of a point set  $P$  in  $d$  dimensions,  $d \geq 3$ , can be computed in  $O(n \log^{d-2} n)$  time. This would allow to compute the set of points of  $P$  in the boundary of the rectilinear convex hull of  $P$  in  $d$  dimensions.

The definition of  $RCH^3(P)$  can also be extended by using three pairwise non-perpendicular axes, i.e., using a set of three independent orientations in  $\mathbb{R}^3$  to define the coordinate axes (as it is done, for example, in the Sect. 3 in Alegría-Galicia et al. [2] for



**Fig. 2** Connected and non-connected rectilinear convex layers in  $\mathbb{R}^2$

points in the plane). Then, as a consequence, we would have to define the corresponding new octants which would be defined by the new three orientations.

It is clear that, analogous to the two dimensions case, the rectilinear convex hull of the set  $P$ ,  $RCH^3(P)$ , is contained in the convex hull of  $P$ ,  $CH(P)$ . Moreover, by definition, all the points in the boundary of  $CH(P)$  belong to the boundary of  $RCH^3(P)$ , but not vice versa. In fact, they could have the same boundary points but not, for example, the same area or volume.

Natural questions on computations on the convex hull of point sets in three dimensions can also be considered for the rectilinear convex hull in three dimensions. As for example, we can consider the computation of the *rectilinear convex layers* of a point set.

The rectilinear convex layers of a point set  $P$  in  $\mathbb{R}^3$  are defined recursively as follows: calculate  $RCH^3(P)$ , remove the elements from  $P$  in  $RCH^3(P)$ , and proceed recursively until  $P$  becomes empty. See Fig. 2 for examples of rectilinear convex layers of two different point sets in  $\mathbb{R}^2$ , the left one is connected, and the right one is non-connected. An optimal  $O(n \log n)$ -time and  $O(n \log n / \log \log n)$ -space algorithm for computing the three-dimensional layers of maxima for point sets in three dimensions was presented by Buchsbaum and Goodrich [12].

To avoid cumbersome terminology, from now on we will refer to  $RCH^3(P)$  simply as  $RCH(P)$ . For simplicity, we assume that the points of  $P$  are in general position both in 2D and in 3D. In 2D, we assume that no three points are on the same line, and no two points have the same  $x$ - or  $y$ -coordinate. In 3D, we assume that no four points are on the same plane, and no two points have the same  $x$ -,  $y$ -, or  $z$ -coordinate.

## 1.1 Previous work

The rectilinear convex hull of a point set in the plane was first studied by Ottmann et al. [23], where they obtain an optimal  $O(n \log n)$ -time algorithm to calculate them. The reader may also consult the monograph *Restricted-Orientation Geometry* [16], where they study topics related to the problems we study here. Other variants were also treated in Fink and Wood [16] and Franěk and Matoušek [17].

The rectilinear convex layers of a point set  $P$  in Euclidean spaces are defined recursively. The rectilinear convex layers of point sets in the plane were first studied by Peláez et al. [24],

where an optimal  $O(n \log n)$ -time and  $O(n)$ -space algorithm to calculate them was obtained. See Fig. 2. For the convex layers of a point set see Chazelle [13].

Variants of the  $RCH(P)$  that were considered in the plane are: computing and maintaining  $RCH_\theta(P)$  when we rotate the coordinate axes around the origin, or determining the angle of rotation  $\theta$  such that the area of  $RCH_\theta(P)$  is minimized or maximized, or also optimizing some other parameters such as the perimeter or the number of points in the boundary of  $RCH_\theta(P)$ , see Alegría-Galicia et al. [1, 2] and Bae et al. [7]. The separability of bichromatic point sets in the plane by using rectilinear convex hulls have been recently studied by Alegría-Galicia et al. [4]. See also [9, 10, 19] for other related works.

A point set is a *rectilinear convex* set if all of its elements lie on the boundary of its rectilinear convex hull. Erdős-Szekeres type problems for finding rectilinear convex subsets of point sets were studied by González-Aguilar et al. [18]. They obtained algorithms to find the largest rectilinear convex subset  $S$  of a point set  $P$ , as well as finding their largest rectilinear convex hole, that is, subsets  $S$  of  $P$  such that  $RCH(S)$  contains no element of  $P$  in the interior; also finding a subset  $S$  of  $P$  such that  $RCH(S)$  has maximum area and its interior contains no element of  $P$ ; or also when each point of  $P$  is assigned a weight, positive or negative, and then finding a subset  $S$  of  $P$  that maximizes the total weight of the points in  $RCH(S)$ . They also revisit the problems of computing a maximum-area orthoconvex polygon and computing a maximum-area staircase polygon, amidst a point set in a rectangular domain.

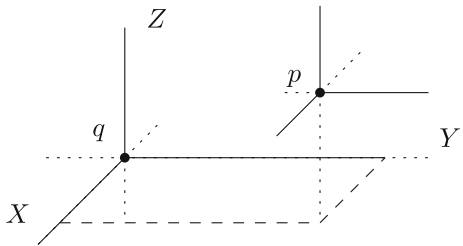
Optimization problems on the rectilinear convex hulls of points sets about several parameters like minimum area, minimum perimeter, or minimum number of points in the boundary, minimum number of connected components, etc, were studied by Alegría-Galicia et al. [2]. Also some variants and extensions of the concept of the rectilinear convex hull of points sets were studied by Alegría-Galicia et al. [1, 3].

The study of the rectilinear convex hull of point sets in the plane is closely related to that of finding the set of maximal points for sets of points under vector dominance; in fact, there is a similarity of our work with that of finding the set of maximal points under vector dominance. This problem was introduced by Kung et al. [22], see also [26]. They obtained an optimal  $O(n \log n)$ -time and  $O(n)$ -space algorithm to solve this problem in the plane and in the three-dimensional space. They also found algorithms to solve the maxima problem in higher dimensions whose time complexity is  $O(n \log^{d-2} n)$  for dimensions  $d \geq 3$ ; however, it is not known whether their algorithm is optimal. Buchsbaum and Goodrich [12] obtained an algorithm to solve the three-dimensional layers of maxima problem in  $O(n \log n)$  time and  $O(n \log n / \log \log n)$  space in  $\mathbb{R}^3$ . Their algorithm is time optimal. See Choi et al. [14], He et al. [20], and Ambrus et al. [5] for some recent developments related to layer numbers of configurations of points.

## 1.2 Results

In Sect. 2, we present an  $O(n \log n)$ -time and  $O(n)$ -space algorithm to compute the rectilinear convex hull of a point set  $P$  in  $\mathbb{R}^3$ . In Sect. 3, we give an  $O(n \log^2 n)$ -time and  $O(n \log n)$ -space algorithm to maintain the set of vertices of the rectilinear convex hull of  $P$  as we rotate  $\mathbb{R}^3$  around the  $Z$ -axis. As a consequence, we solve several optimization problems on some parameters relative to  $RCH_\theta(P)$  as  $\theta \in [0, 2\pi]$ : computing the intervals of  $\theta$  with the minimum number of points of  $P$  in the boundary, or with the minimum number of connected components, etc. In Sect. 4, we obtain some results on the possible number of combinatorially different rectilinear convex hulls of  $P$  in  $\mathbb{R}^3$ , which are related to our algorithmic results, and that are interesting in their own right. In particular, we show that the rectilinear convex hull of

**Fig. 3** Partial domination according to the sign pattern  $(+, +, +)$  of the first octant



a point set  $P$  can change a quadratic number of times while its vertex set remains unchanged. In Sect. 5, we present an application of our results to give an approximation algorithm to a fitting problem for points in 3D, more concretely, the *oriented 2-fitting problem*, see Díaz Bañez et al. [15]; see also Bhattacharya et al. [8] for the oriented  $k$ -fitting problem. Finally, in the last Sect. 6, we present some interesting open problems and future lines of research on this topic.<sup>1</sup>

### 1.3 Notation and preliminaries

For a point  $p \in \mathbb{R}^3$ , we refer to  $x_p, y_p$ , and  $z_p$  as the  $x$ -,  $y$ -, and  $z$ -coordinates of  $p$ , respectively. A point  $p$  satisfies a sign pattern; e.g.,  $(+, -, +)$  if  $x_p \geq 0, y_p \leq 0$ , and  $z_p \geq 0$ .

There are eight possible sign patterns that a point can satisfy, namely:  $(+, +, +), (-, +, +), (-, -, +), (+, -, +), (+, -, -), (-, -, -), (-, +, -)$ , and  $(+, +, -)$ . The first sign pattern corresponds to all of the points in the first octant of  $\mathbb{R}^3$ . Similarly, we will say that the points satisfying the second pattern,  $(-, +, +)$ , correspond to points in the second octant, ..., and those satisfying  $(+, +, -)$  correspond to the eighth octant of  $\mathbb{R}^3$ .

The *open* octants are defined in a similar way, except that we require strict inequalities; e.g., the first open octant corresponds to points for which  $x_p > 0, y_p > 0$ , and  $z_p > 0$ .

Given a point  $p \in \mathbb{R}^3$ , the octants with respect to  $p$  are the octants induced by a translation of the origin to  $p$ . In addition, each of the eight sign patterns defines a partial order on the elements of  $\mathbb{R}^3$  as follows: consider two points  $p, q \in \mathbb{R}^3$ .

We say that  $p$  dominates  $q$  according to the sign pattern  $(+, +, +)$  if

$$(x_q \leq x_p) \wedge (y_q \leq y_p) \wedge (z_q \leq z_p).$$

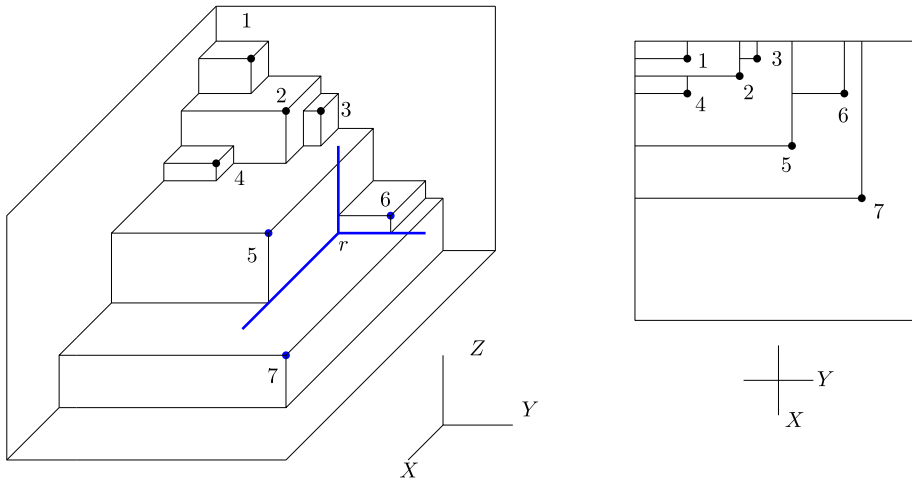
In other words, we can also say that  $p$  dominates  $q$  if  $p$  is in the first octant with respect to  $q$ , see Fig. 3.

We refer to this domination as  $q \preceq_1 p$ , see again Fig. 3. In a similar way, we define the domination relations with respect to the other seven sign patterns, which we denote as  $p \preceq_i q$ , for  $i = 2, 3, \dots, 8$ . For example, the dominance relation with respect to the second sign pattern is:

$$q \preceq_2 p \iff (x_q \geq x_p) \wedge (y_q \leq y_p) \wedge (z_q \leq z_p).$$

These dominance relations define partial orders in  $P$ , and they can be extended to any dimension  $d > 3$  in a straightforward way.

<sup>1</sup> A preliminary version of this paper was presented in the 14th Latin American Theoretical Informatics Symposium, LATIN 2020 [25].



**Fig. 4** Left: Maxima points in the first octant with the topmost antenna (the point of  $P$  with the highest  $z$ -coordinate) in  $p_1$ , and an extremal first open octant in blue. Right: The projection on the  $XY$  plane of the first octant maxima point set. (Color figure online)

**Definition 1** A point  $p \in P$  is a *maximal element* of  $P$  with respect to a partial order if there is no  $q \in P$  that dominates  $p$ .

The partial order  $\leq_1$  defined by  $(+, +, +)$  is usually known as *vector dominance* [22].

## 2 Computing $RCH(P)$

In this section, we show how to compute the rectilinear convex hull of an  $n$ -point set in  $\mathbb{R}^3$  with an optimal algorithm with  $O(n \log n)$  time and  $O(n)$  space.

First, we recall that the problem of finding the maximal elements of a point set in  $\mathbb{R}^2$  or  $\mathbb{R}^3$  with respect to vector dominance is called the maxima problem. The unoriented maxima problem in the plane was studied by Avis et al. [6]. The next result is well known.

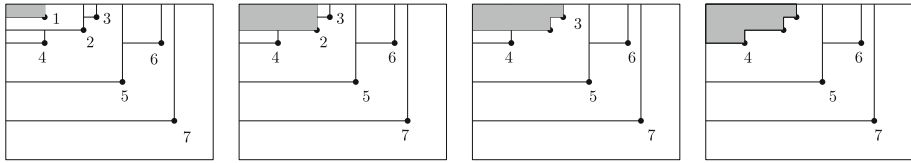
**Theorem 1** [22] *The maxima problem in  $\mathbb{R}^d$ ,  $d = 2, 3$ , can be solved in optimal  $O(n \log n)$  time and  $O(n)$  space.*

In fact, when solving the maxima problem, we obtain the set of faces and vertices of the *orthogonal polyhedron*,<sup>2</sup> i.e., the faces of the orthogonal polyhedron meet at right angles and the edges are parallel to the axes; call it  $\mathcal{P}^1$  as shown in Fig. 4 left. In the right part of Fig. 4 we show the top view of  $\mathcal{P}^1$ .

Observe that if we intersect a horizontal plane  $\mathcal{Q}_c$  with equation  $z = c$  with  $\mathcal{P}^1$ , we obtain an orthogonal polygon  $\mathcal{C}_c^1$  that will change as we move  $\mathcal{Q}_c$  up or down; i.e., as we increase or decrease  $c$ .

In Fig. 5, we show how  $\mathcal{C}_c^1$  changes as we scan it from top to bottom starting at the top vertex of  $\mathcal{P}^1$ . We show the intersection of  $\mathcal{P}^1$  with  $\mathcal{Q}_c$ , as  $\mathcal{Q}_c$  sweeps through the first four top vertices of  $\mathcal{P}^1$ .

<sup>2</sup> An orthogonal polyhedron is a polyhedron such that all the faces meet at right angles, and all the edges are parallel to the coordinate axes. Apart from the hexahedron, orthogonal polyhedra are non-convex. They are the three dimension analogs of orthogonal polygons, also known as rectilinear polygons.



**Fig. 5** How  $C_c^1$  changes as  $Q_c$  moves down from point 1 to point 4

It is easy to see that when  $Q_c$  moves from one vertex of  $\mathcal{P}^1$  to the next,  $C_c^1$  changes in the following way: a new vertex appears, which is the vertex of an *elbow* from which two rays emanate, one horizontal and one vertical, that extend until they hit  $C_c^1$  or go to infinity; see Fig. 5.

Analogously, we can define  $\mathcal{P}^i, C_c^i, i = 2, \dots, 8$ . Since  $RCH(P) = \bigcap_{i=1, \dots, 8} \mathcal{P}^i$ , we have the following theorem.

**Theorem 2** *For each constant  $c$ , the intersection of  $RCH(P)$  with  $Q_c$  is the intersection of the orthogonal polygons  $C_c^i, i = 1, \dots, 8$ .*

To compute  $RCH(P)$  we will sweep a plane  $Q_c$  from top to bottom, stopping at each point of  $P$  on  $RCH(P)$ . Each time we stop, we need to update  $RCH(P) = \bigcap_{i=1, \dots, 8} \mathcal{P}^i$ . We claim that we can do this in  $O(\log n)$  time. To prove this, observe that when we move  $Q_c$  from a point  $p$  of  $P$  to the next point, say  $q$ , the only curves  $C_c^i, i = 1, \dots, 8$  that change are those containing  $q$ , and these can be recomputed in  $O(\log n)$  time.

This computation can be done as follows. Suppose that we have to update  $q$  in  $C_c^i$ : we need to maintain the staircase (with linear complexity) defining the *exterior* boundary of  $C_c^i$  (see Fig. 5) and update this staircase by inserting a new vertex (step)  $q$ , i.e., computing in  $O(\log n)$  time the part of the boundary/staircase dominated by  $q$  according to the domination for  $C_c^i$  in case that  $q$  is a maxima point for  $C_c^i$  (see Kung et al. [22]). Notice that  $q$  can belong to several (at most 8)  $C_c^i$ 's, so all of them can be updated in total  $O(\log n)$  time.

As a consequence of the discussion above, we have the following theorem.

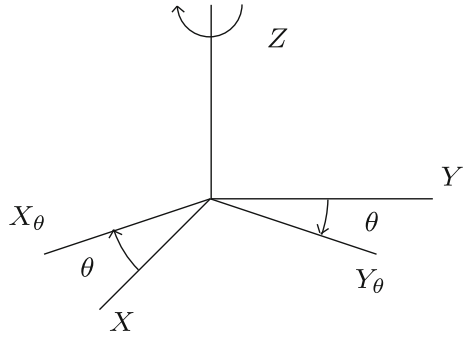
**Theorem 3** *Given a set  $P$  of  $n$  points in 3D, the rectilinear convex hull of  $P$ ,  $RCH(P)$ , can be computed in optimal  $O(n \log n)$  time and  $O(n)$  space.*

Notice that to compute  $RCH(P)$  it is enough to store the list of the maximal points in decreasing order of their  $z$ -coordinates, and then sweeping from top to bottom to construct the boundary of  $RCH(P)$ . Notice also that the numbers of edges and faces of  $RCH(P)$ , respectively, are also linear in the number of vertices of the boundary of  $RCH(P)$ , so from the list of maximal points we can construct  $RCH(P)$  in  $O(n \log n)$  time, doing an update of  $O(\log n)$  time for each point of the list (see the comments after Theorem 2).

On the other hand, the vertices and edges of  $CH(P)$  remains if we change the orientation of the coordinate system, but here the sets of vertices, edges, and faces change with the orientation of the coordinate axes thus, in order to store  $RCH(P)$  we will maintain the necessary information to reconstruct  $RCH(P)$  in linear time.

**Remark 1** The optimality of the time complexity in Theorem 3 can be trivially obtained either by a reduction to the maxima problem in Theorem 1 or by a reduction to the computation of the convex hull of  $P$  in 3D,  $CH(P)$ , since from the  $RCH(P)$ , i.e., having all the maximal points, we can compute  $CH(P)$  in  $O(n)$  time.

**Fig. 6** Rotating clockwise the axes: from  $X, Y, Z$  to  $X_\theta, Y_\theta, Z$



### 3 Maintaining $RCH_\theta(P)$

In the plane, the problem of maintaining  $RCH_\theta(P)$  as  $\theta$  changes from 0 to  $2\pi$  has been studied [1, 2, 7]. In this section, we will study the same problem in  $\mathbb{R}^3$  but restricted to rotations of  $\mathbb{R}^3$  around the  $Z$ -axis.

Thus, in the rest of this section we will use octants (instead of quadrants) which are defined as intersections of three mutually orthogonal halfspaces whose supporting planes are orthogonal to three mutually orthogonal lines through the origin, one of which is the  $Z$ -axis. Therefore, two of these three lines lie on the  $XY$ -plane, and correspond to rotations of the  $X$ - and  $Y$ -axis by an angle  $\theta$  in the clockwise direction, respectively. See Fig. 6. We call such octants as  $\theta$ -octants, and the corresponding rectilinear convex hulls as generated  $RCH_\theta(P)$ .

Thus, an open  $\theta$ -octant is the intersection of three open halfspaces whose supporting planes are orthogonal to three mutually orthogonal lines through the origin:  $X_\theta, Y_\theta$ , and  $Z$ . An open  $\theta$ -octant is called  $P$ -free if it contains no elements of  $P$ . We define the  $\theta$ -rectilinear convex hull  $RCH_\theta(P)$  of a point set  $P$  as the set

$$RCH_\theta(P) = \mathbb{R}^3 \setminus \bigcup_{W \in \mathcal{W}_\theta(P)} W,$$

where  $\mathcal{W}_\theta(P)$  denotes the set of all  $P$ -free open  $\theta$ -octants.

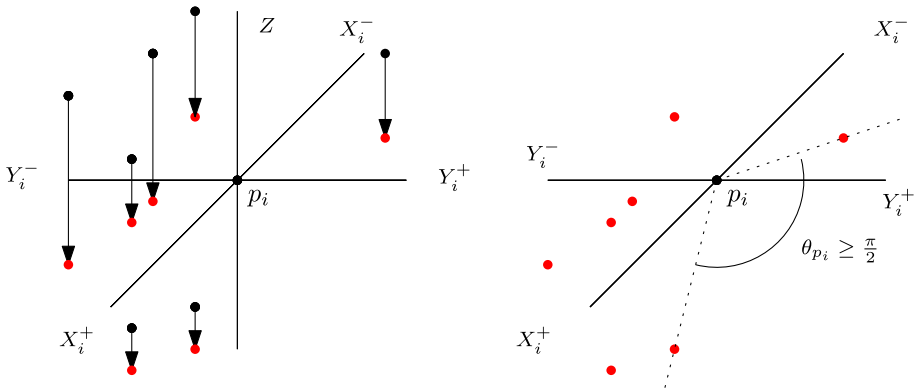
In the rest of this section, we assume that elements of  $P$  are labeled  $\{p_1, \dots, p_n\}$  from top to bottom according to their decreasing  $z$ -coordinates.

For every  $p \in \mathbb{R}^3$ , there are eight  $\theta$ -octants having  $p$  as their apex; we will call them  $p^\theta$ -octants. Exactly four  $p^\theta$ -octants contain points in  $\mathbb{R}^3$  above the horizontal plane  $\lambda_p$  through  $p$ , and the other four have points below  $\lambda_p$ . We call the first four up  $p^\theta$ -octants, and the other down  $p^\theta$ -octants. Note that if an up  $p^\theta$ -octant is no- $P$ -free, it contains elements of  $P$  above  $\lambda_p$ , and that no- $P$ -free down  $p^\theta$ -octants contain points in  $P$  below  $\lambda_p$ .

**Observation 1** *Observe that a point  $p \in P$  is a vertex of  $RCH_\theta(P)$  if there is a  $P$ -free  $p^\theta$ -octant. In this case, we will say that  $p$  is a  $\theta$ -active point, otherwise  $p$  is  $\theta$ -inactive. Furthermore, if there is a  $P$ -free up  $p^\theta$ -octant, we call  $p$  an up  $\theta$ -active vertex. We define down  $\theta$ -active vertices in a similar way.*

We first analyze the set of angles for which points in  $P$  are up  $\theta$ -active. Let  $p_i$  be a point of  $P$ , and consider the orthogonal projection on  $\lambda_{p_i}$  of the points  $p_1, \dots, p_{i-1}$  (the points in  $P$  above  $\lambda_{p_i}$ ), and let  $P'_i$  be the point set thus obtained; see Fig. 7 left. If for some  $\theta$ ,  $p_i$  is up  $\theta$ -active, then there is a wedge of angular size  $\theta_{p_i} \geq \frac{\pi}{2}$  on  $\lambda_{p_i}$  whose apex is  $p_i$ , that is,  $P'_i$ -free; see Fig. 7 right.





**Fig. 7** Checking whether the point  $p_i$  is maximal with respect to the first octant for some angular interval: project the set of points  $\{p_1, p_2, \dots, p_{i-1}\}$  on the plane parallel to the  $XY$  plane passing through  $p_i$ , and obtain the projected points (red points in figure). (Color figure online)

Clearly, no more than three such disjoint wedges can exist. In a similar way, we can prove that  $p_i$  is down  $\theta$ -active in at most three angular intervals. Thus, the following result, equivalent to a result in Avis et al. [6] for points on the plane, follows.

**Theorem 4** *The set of angular intervals of the angles  $\theta$  for which a point  $p_i \in P$  is  $\theta$ -active consists of at most six disjoint intervals in the set of directions  $[0, 2\pi]$ .*

Finding the angular intervals at which  $p_i$  is up-active is now reduced to finding, if they exist,  $P'_i$ -free wedges in  $\lambda_{p_i}$  whose apex is  $p_i$  of angular size at least  $\frac{\pi}{2}$ . We solve this as follows: note first that if one such a wedge exists, it has to contain at least one of the four rays emanating from  $p_i$  parallel to the  $X$ - or  $Y$ -axis. Let those rays be  $X_i^+$ ,  $X_i^-$ ,  $Y_i^+$ , and  $Y_i^-$ ; see Fig. 7 right. For  $X_i^+$ , we will solve the following problem:

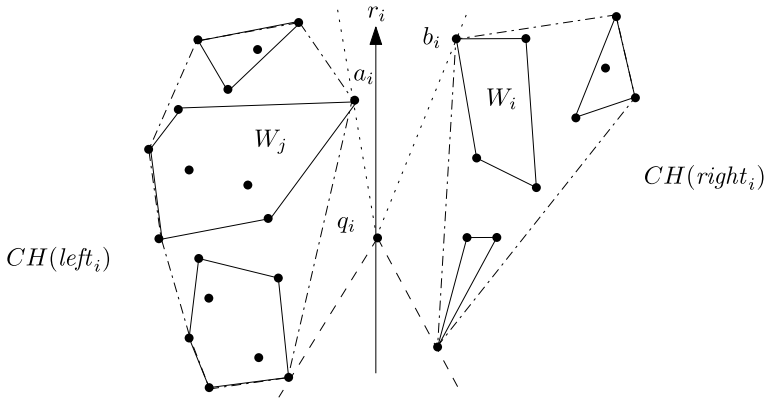
Rotate  $X_i^+$  clockwise until it hits a point in  $P'_i$ . Next, rotate  $X_i^+$  counter-clockwise until it hits another point in  $P'$ . Measure the angle  $\alpha$  formed by the two rays thus obtained. If  $\alpha \geq \frac{\pi}{2}$ , then we have found an interval of the set of intervals at which  $p_i$  is active, else discard  $X_i^+$ . Proceed in the same way with  $X_i^-$ ,  $Y_i^+$ , and  $Y_i^-$ . We will have to repeat this process for all of the points  $p_i \in P$  from top to bottom. We now show how to process all the points in  $P$  in  $O(n \log^2 n)$  time and  $O(n \log n)$  space.

The main difficulty in finding the wedges in the above discussion is that as we process the points of  $P$  from top to bottom, the number of points in  $\lambda_{p_i}$  increases one by one, and thus, we need a dynamic data structure to solve the following problem.

**Problem 1** *Let  $Q = \{q_1, \dots, q_n\}$  be a set of points in the plane, and let  $Q_{i-1}$  be the set  $Q_{i-1} = \{q_1, \dots, q_{i-1}\}$ . For each  $i$  we want to solve the following problem: let  $r_i$  be the vertical ray that starts at  $q_i$  and points up. Find the first point  $a_i$  (respectively,  $b_i$ ) of  $Q_{i-1}$  that  $r_i$  meets when we rotate  $r_i$  in the counter-clockwise (respectively, clockwise) direction around  $q_i$ . See Fig. 8.*

One last point before proceeding with our results; instead of projecting the points of  $P$  on the planes  $\lambda_{p_i}$ , we will project them one by one on the  $XY$ -plane. Everything else remains unchanged.

**Theorem 5** *Problem 1 can be solved in  $O(n \log^2 n)$  time and  $O(n \log n)$  space.*



**Fig. 8** The points  $a_i$  and  $b_i$  can be computed by the supporting lines of the convex hull of  $left_i$  and  $right_i$  passing through  $q_i$

**Proof** Assume without loss of generality that no two points of  $Q$  lie on a horizontal line, and let  $left_i$  and  $right_i$  be the set of points in  $P$  lying to the left (respectively, right) of the vertical line through  $q_i$ . See Fig. 8.

If we know the convex hulls of  $left_i$  and  $right_i$ , then the points we are seeking,  $a_i$  and  $b_i$ , can be computed by calculating the supporting lines of the convex hull of  $left_i$  and  $right_i$  passing through  $q_i$ . It is well known that we can compute these lines in  $O(\log n)$  time, if one keeps the vertices of each hull sorted counter-clockwise around its boundary, see Preparata and Shamos [26]. Figure 8 illustrates the supporting lines.

Furthermore, if  $left_i$  and  $right_i$  have been decomposed into  $k$  disjoint sets  $W_1, \dots, W_k$  where each  $W_i$  is contained in  $left_i$  or in  $right_i$ , and we have the convex hull  $conv(W_i)$  of all of these point sets, then we can find the supporting lines through  $q_i$  for all of them in overall  $O(\log |W_1| + \dots + \log |W_k|) = O(\log^2 n)$  time. See Fig. 8. This will now allow us to obtain  $a_i$  and  $b_i$  in  $O(\log n)$  time. This is the main idea that will enable us to design a data structure to solve Problem 1 in  $O(n \log^2 n)$  time.

The next observation on binary trees will be useful.

**Observation 2** Let  $T = (\mathcal{V}, \mathcal{E})$  be a balanced, binary, and rooted tree, where the edges have the orientation from the root. For each node  $v \in \mathcal{V}$ , let  $S_v \subset \mathcal{V}$  be the set of leaves of  $T$  that are descendants of  $v$ . Let  $u \in \mathcal{V}$  be a leaf of  $T$ , and consider the path  $p_u$  that joins  $u$  to the root of  $T$ , let  $\mathcal{P}_u$  be the set of nodes in the path  $p_u$ , and let  $\Delta_u$  be the set of nodes in  $\mathcal{V} \setminus \mathcal{P}_u$  that are direct descendants of a node in the path  $p_u$ . Then, the sets  $S_v$ ,  $v \in \Delta_u$  induce a partition of the set of leaves of  $T \setminus \{u\}$ . Moreover, if a node  $w$  in  $\Delta_u$  is the right (left) descendant of a node in the path  $p_u$ , then the elements of  $S_w$  lie to the right (respectively, left) of  $u$ . See Fig. 9.

Let  $D(Q)$  be a balanced binary tree whose leaves are the elements of  $Q$  sorted in order from left to right according to their  $x$ -coordinate. Note that this order does not necessarily coincide with the labeling  $q_1, \dots, q_n$  of the elements of  $Q$ .

Initially, every leaf of  $D(Q)$  is considered *inactive*. For each vertex of  $D(Q)$  we maintain the convex hull of  $W(q)$ , where  $W(q)$  is the set of descendant leaves (points of  $Q$ ) that are currently active. See Fig. 10. Using the data structure of Brodal and Jacob [11], we can maintain such a convex hull in  $O(\log |W(q)|)$  time per insertion of a new active point, and allowing tangent (or supporting line) queries also in  $O(\log |W(q)|)$  time.



Observe that the set of intervals in the set of directions  $[0, 2\pi]$  at which the points of  $P$  are active define intervals in the unit circle  $S^1$ , where the points on  $S^1$  correspond to angles in  $[0, 2\pi]$ . Thus, if an angle  $\theta$  belongs to an interval at which a point of  $P$  is active, this point is a vertex of  $RCH_\theta(P)$ . As  $\theta$  goes from 0 to  $2\pi$ , the vertices of  $RCH_\theta(P)$  are those for which one of its active angular intervals contains  $\theta$ .

As a consequence of the above discussion, we have the following result.

**Theorem 7** *Given a set  $P$  of  $n$  points in 3D, maintaining the elements of  $P$  that belong to the boundary of  $RCH_\theta(P)$  as  $\theta \in [0, 2\pi]$  can be done in  $O(n \log^2 n)$  time and  $O(n \log n)$  space.*

Note that if we store the set of angular intervals in the set of directions  $[0, 2\pi]$  at which the points of  $P$  are active, then for any angle  $\theta$  we can retrieve the points in  $P$  that are  $\theta$ -active in linear time.

In the case where we also want to compute  $RCH_\theta(P)$  for a particular value of  $\theta$ , all we have to do is to retrieve those  $\theta$ -active points of  $P$  in linear time, and then use the algorithm presented in Theorem 3.

**Corollary 1** *The algorithm in Theorem 7 is used to solve the following optimization problems: (i) Computing the intervals of  $\theta$  with the minimum/maximum number of vertices of  $P$  in  $RCH_\theta(P)$ , (ii) Computing the smaller/larger intervals of  $\theta$  without changes in the set of vertices of  $P$  in  $RCH_\theta(P)$ , (iii) Computing an upper bound of the optimal fitting of 2-halfplanes to a set of points in three dimensions (see Sect. 5).*

We note that computing the minimum/maximum number of connected components of  $RCH_\theta(P)$  as  $\theta \in [0, 2\pi]$  by using the algorithm in Theorem 7 is not straightforward and it requires extra details. Thus, we left this question for future work.

**Remark 2** The rectilinear convex hull of a point set, while rotating  $\mathbb{R}^3$  around any line through the origin, can be obtained easily in the same  $O(n \log^2 n)$  time and  $O(n \log n)$  space, as any such a rotation can be achieved as a composition of a rotation around the  $Z$ -axis followed by a rotation around the  $X$ -axis.

## 4 Combinatorial analysis of the rectilinear convex hulls

In the previous section, we studied the problem of maintaining the set of points of  $P$  that are vertices of  $RCH_\theta(P)$ . The problem of maintaining the boundary of  $RCH_\theta(P)$  does not follow from our previous results.

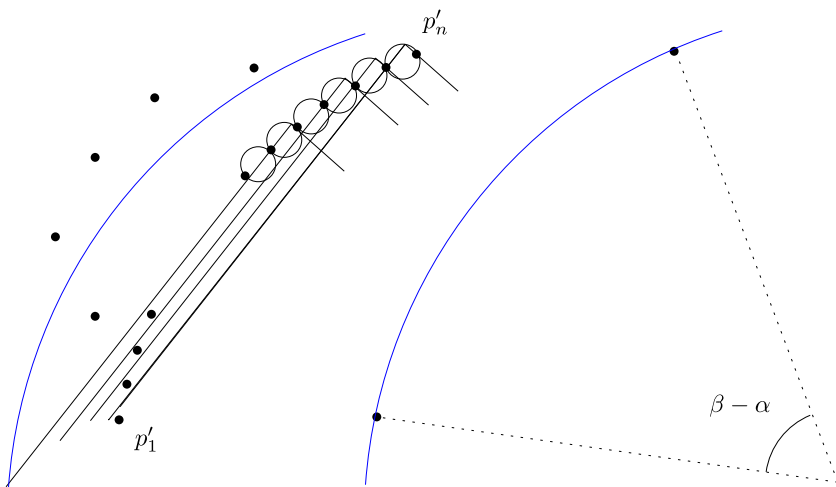
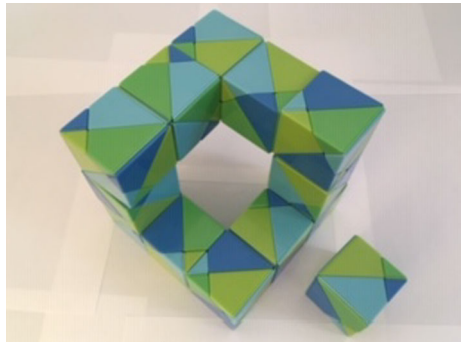
As we shall see, there are examples of configurations of point sets  $P$  in  $\mathbb{R}^3$  such that the number of combinatorially different rectilinear convex hulls of  $P$  can be at least  $\Omega(n^2)$  while the set of vertices of  $RCH_\theta(P)$  remains unchanged.

To begin, we notice that  $RCH_\theta(P)$  is not necessarily connected; this is easy to see, as for point sets in the plane this property does not hold; e.g., see Fig. 1 right.

What is a bit more interesting is that even when  $RCH_\theta(P)$  is connected and has non-empty interior, it is not necessarily simply connected.

In Fig. 11, we show a rectilinear convex hull of a point set whose rectilinear convex hull is homeomorphic to a torus. The elements of  $P$  are the vertices of the cubes glued together to obtain the figure. The reader will notice immediately that the points in  $P$  are not in general position. A slight perturbation of the elements of  $P$ , that would bring them to a point set in

**Fig. 11** The rectilinear convex hull of a point set that is not simply connected



**Fig. 12** A configuration of points that illustrates a point set for which its set of vertices remains unchanged while its rectilinear convex hull changes a quadratic number of times

general position, will also yield a rectilinear convex hull whose interior is homeomorphic to a torus.

Evidently, we can construct similar examples to that shown in Fig. 11 in which we obtain oriented polyhedron or three-dimensional shapes of arbitrarily large genus.

We now construct a point set such that for an angular interval  $[\alpha, \beta]$  while  $\theta \in [\alpha, \beta]$ ,  $RC H_\theta(P)$  will maintain the same vertices while it changes a quadratic number of times.

Consider a circular cylinder  $\mathcal{C}$  that is perpendicular to the  $XY$ -plane, and consider a geodesic curve  $\mathcal{H}$  on  $\mathcal{C}$  that joins two points  $p$  and  $q$  on  $\mathcal{C}$ . Choose a set of  $n$  points  $P' = \{p'_1, \dots, p'_n\}$  on  $\mathcal{H}$  such that their projection on the  $XY$ -plane is a set of equidistant points on a small interval of the circle in which  $\mathcal{C}$  and the  $XY$ -plane intersect, and such that if  $i < j$  the  $z$ -coordinate of  $p'_i$  is smaller than the  $z$ -coordinate of  $p'_j$ ; see Fig. 12.

Observe that there is a  $P'$ -free up octant  $Q_{1,n}$  whose bottom, left, and back faces contain  $p'_1, p'_{n-1}$  and  $p'_n$ , respectively. Observe now that there is a translation of  $Q_1$  up and to the left so as to produce a second octant  $Q_{1,n-1}$  whose bottom, left and back faces contain  $p'_1, p'_{n-1}$ , and  $p'_{n-2}$ , respectively. We can iterate this process  $\frac{n-1}{2}$  times to obtain a set of  $\frac{n-1}{2}$

$P'$ -free octants that have on their bottom, left and back faces  $p'_1$ ,  $p'_i$ , and  $p'_{i+1}$ , respectively,  $i = \lfloor \frac{n-1}{2} \rfloor, \dots, n-1$ ; see Fig. 12.

Rotating  $Q_2$  slightly in the clockwise direction and moving it up we can obtain a  $P$ -free octant  $Q_{2,n}$  containing on its bottom, left, and back faces  $p'_2$ ,  $p'_n$ , and  $p'_{n-1}$ , respectively. We can now repeat the same process as we did with  $\{p'_1, \dots, p'_n\}$ , and with  $\{p'_2, \dots, p'_n\}$ , starting with  $p'_2$  and  $Q_{2,n}$  to obtain a new set of  $P'$ -free extremal octants. Repeat this process with  $\{p'_3, \dots, p'_n\}$ , ...,  $\{p'_{n-3}, \dots, p'_n\}$ , to obtain a quadratic number of  $P'$ -free extremal  $P'$ -octants.

Let  $\alpha$  and  $\beta$  be the angles of rotation of the  $XY$ -plane such that the  $X$ -axis becomes parallel to the line at which the back faces of  $Q_{1,n}$  and  $Q_{n-3,n}$  intersect the  $XY$ -plane. Observe that all of  $p'_1, \dots, p'_n$  are active points and on the boundary of  $RCH_\theta$  for all  $\theta \in [\alpha, \beta]$ .

In the meantime, all of the  $P$ -free octants we obtained above become active during an angular interval contained in  $[\alpha, \beta]$ . To complete our construction, we add a few points to the set  $\{p'_1, \dots, p'_n\}$ . These points are located in the exterior of the circular cylinder  $\mathcal{C}$ , placed appropriately to ensure that the rectilinear convex hull of the point set  $P$  thus obtained has non-empty interior, and all of  $\{p'_1, \dots, p'_n\}$  are on its boundary. See Fig. 12.

Thus, the rectilinear convex hull of  $P$  changes a quadratic number of times while its vertex set remains unchanged.

By the discussion above we have the following result.

**Theorem 8** *There are configurations of points in  $\mathbb{R}^3$  such that for an angular interval  $[\alpha, \beta]$  while  $\theta \in [\alpha, \beta]$ ,  $RCH_\theta(P)$  will maintain the same vertices while it changes a quadratic number of times.*

## 5 An application to optimize a 3D fitting problem

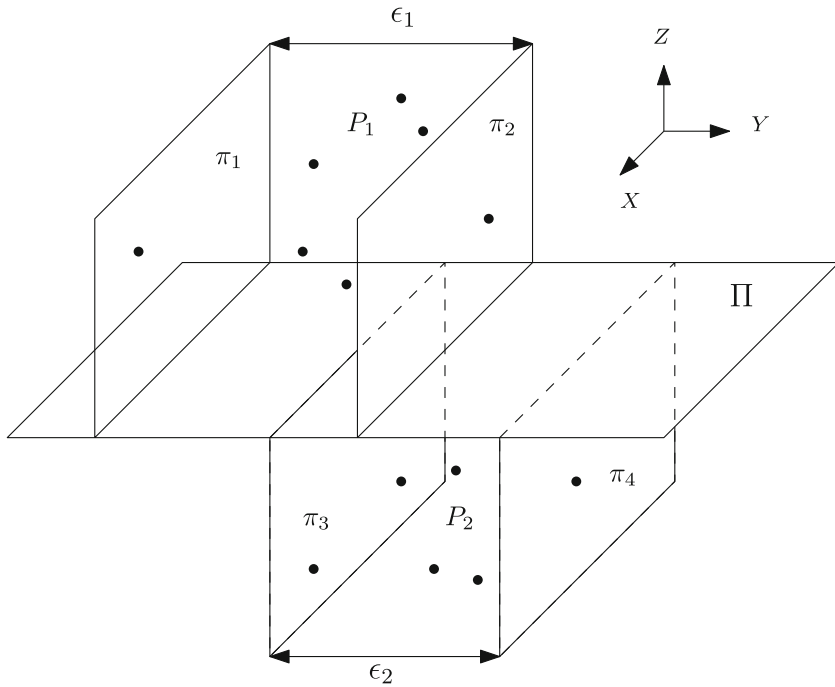
The 1-fitting problem for a set  $P$  of  $n$  points in  $\mathbb{R}^3$  consists in finding a plane  $\Pi$  such that the maximum Euclidean distance from the points of  $P$  to the plane  $\Pi$  is minimized. This distance is called the *error tolerance* of the plane  $\Pi$ . Clearly,  $\Pi$  is given by the mid-plane between the two supporting parallel planes which define the width of  $P$ .

Concretely, the width of  $P$  can be computed in  $O(n^2)$  time and  $O(n)$  space by considering all the antipodal possibilities once the convex hull of  $P$ ,  $CH(P)$ , is computed, i.e., by analyzing and computing all the Euclidean distances between two vertices, a vertex and a face, a vertex and an edge, and two edges of  $CH(P)$ , and maintaining the minimum one (see Houle and Toussaint [21]).

Precisely, the number of different edge pairs is the reason for the  $O(n^2)$  time of the algorithm. Nevertheless, if we impose some additional information, we can find the plane  $\Pi$  easily. We distinguish the two following cases:

(i) If the orientation of the solution plane is fixed, say, if its normal vector is  $\vec{u} = (0, 0, 1)$ , then the problem can be solved in  $O(n)$  time by computing the points with maximum and minimum  $z$ -coordinates.

(ii) If the orientation of the solution plane has one degree of freedom, e.g., that the normal vector  $\vec{u}$  to  $\Pi$  is orthogonal to  $\vec{v} = (0, 1, 0)$ , then we can solve the problem by projecting the points on a plane with normal  $\vec{v}$ , and then computing the width of the convex hull of the projected points with a rotating caliper algorithm. The total running time is  $O(n \log n)$  which is optimal. The lower bound follows from the complexity of computing the width of a set of points on the plane.



**Fig. 13** The splitting-plane plane  $\Pi$  and the two pairs of parallel supporting-halfplanes

In order to do a better fitting of a point set  $P$  in  $\mathbb{R}^3$  with smaller error tolerance, one can do the following: instead of doing the fitting of the points to one plane and computing the corresponding error tolerance, we do the fitting of the points with two different planes by computing the two corresponding error tolerances and minimizing the maximum of the two values. This problem was studied by Díaz-Bañez et al. [15] and it is formalized next.

The *oriented 2-fitting problem* [15] is defined as follows: Given a point set  $P$  in  $\mathbb{R}^3$ , find a plane  $\Pi$ , called the *splitting plane* of  $P$  (assume that  $\Pi$  is parallel to the  $XY$ -plane), and four parallel halfplanes  $\pi_1, \pi_2, \pi_3, \pi_4$ , called the *supporting halfplanes* of  $P$ , such that:

1.  $\Pi$  splits  $P$  into two non-empty subsets  $P_1$  and  $P_2$ , i.e.,  $\{P_1, P_2\}$  is the bipartition of  $P$  produced by the splitting-plane  $\Pi$ .
2.  $\pi_1, \pi_2, \pi_3, \pi_4$  are orthogonal to  $\Pi$ ,  $\pi_1$  and  $\pi_2$  lie above  $\Pi$ , and  $\pi_3$  and  $\pi_4$  lie below  $\Pi$ .
3. The maximum of  $\epsilon_1$  and  $\epsilon_2$  is minimized, where  $\epsilon_1$  is the error tolerance of  $P_1$  with respect to  $\pi_1$  and  $\pi_2$ , and  $\epsilon_2$  is the error tolerance of  $P_2$  with respect to  $\pi_3$  and  $\pi_4$ .
4. The solution for the 2-fitting problem is given by the mid-halfplane of the supporting halfplanes  $\pi_1$  and  $\pi_2$ , and the mid-halfplane of the supporting halfplanes  $\pi_3$  and  $\pi_4$ .

Clearly, the error tolerances  $\epsilon_1$  and  $\epsilon_2$  are defined by the Euclidean distances between the two parallel supporting halfplanes on either sides of  $\Pi$ , and the problem consists in getting the bipartition  $\{P_1, P_2\}$  of  $P$  such that the value  $\max\{\epsilon_1, \epsilon_2\}$  is minimum, i.e., it is a min-max problem. See Fig. 13 for an illustration.

If the orientations of the splitting-plane and the supporting halfplanes are fixed, then the 2-fitting problem can be solved in linear time. If only the orientation of the supporting-halfplanes is fixed, then the problem can be solved in  $O(n \log n)$  time and  $O(n)$  space by doing a binary search on the possible bipartitions of  $P$  with the orthogonal splitting plane.

Finally, if only the orientation of the splitting plane is fixed, then the problem can be solved in  $O(n^2)$  time and  $O(n)$  space, as proved by Díaz-Báñez et al. [15].

We recall a sketch of the algorithm of Díaz-Báñez et al. [15] for the last case: Sweep the point set with the splitting-plane and stop at each point of  $P$  for computing and updating the new error tolerances  $\epsilon_1$  and  $\epsilon_2$ , and the value  $\max\{\epsilon_1, \epsilon_2\}$ ; where the error tolerances are computed by projecting the respective points of the current bipartition on different planes updating their respective convex hulls and computing the respective simultaneous widths  $\epsilon_1$  and  $\epsilon_2$ .

We will design an algorithm that returns an upper bound to the optimal solution for this case: when the orientation of the splitting plane is fixed.

In our algorithm, instead of doing a sequence of bipartitions of  $P$  as done by Díaz-Báñez et al. [15], we will use our algorithm of Sect. 3 for maintaining the elements of  $P$  that belong to the boundary of  $RC H_\theta(P)$  as we rotate the space around the  $Z$ -axis, and compute optimal solutions in each of a linear number of events at which  $RC H_\theta(P)$  changes for  $\theta \in [0, \pi]$ .

We assume that the splitting-plane  $\Pi$  is parallel to the  $XY$  plane, all the points of  $P$  are above the  $XY$  plane and sorted with respect to the  $z$ -coordinate, where  $p_1$  is the point with the largest  $z$ -coordinate, and  $p_n$  is the point with the smallest  $z$ -coordinate. We also assume that the  $Z$ -axis passes through  $p_1$ , and thus, its coordinates do not change as we rotate  $\mathbb{R}^3$  around the  $Z$ -axis, and  $p_1$  and  $p_n$  are always in the boundary of  $RC H_\theta(P)$ .

Suppose that the supporting halfplanes have normal unit vectors in the unit circle  $S^1$  on the  $XY$ -plane. Thus, the 2-fitting problem reduces to computing four parallel supporting halfplanes  $\pi_1, \pi_2, \pi_3, \pi_4$  of a bipartition  $\{P_1, P_2\}$  of  $P$ , and therefore, to computing four points in the boundary of  $RC H_\theta(P)$ , for some  $\theta \in [0, \pi]$ . See Fig. 14.

We will discretize the problem by considering the angular sub-intervals of  $[0, \pi]$  such that in each sub-interval the points in the boundary of  $RC H_\theta(P)$  do not change. By Theorem 6, their number is linear in  $n$ . Then, we will show how to optimize the error tolerance at the endpoints of each of these sub-intervals.

Recall that a point  $p \in P$  is said to be  $\theta$ -active if at least one of the  $p^\theta$ -octants is  $P$ -free (see Observation 1). The definition of a  $\theta$ -active point considering an octant can be easily adapted to considering a *dihedral* (two perpendicular and axis-parallel planes) as follows:

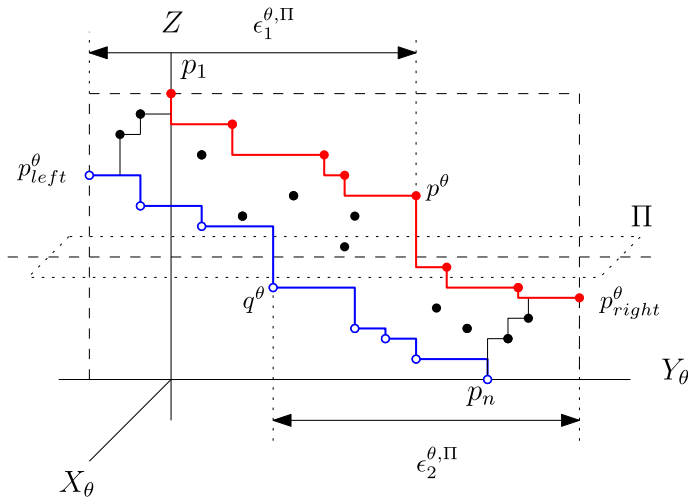
**Definition 2** Let  $p \in P$  and  $\{s, t\} \in \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$ . We say that  $p$  is a  $\theta_{\{s,t\}}$ -active point if  $p$  is  $\theta$ -active for both the  $s$ -th and  $t$ -th octants.

For example, a point  $p \in P$  is  $\theta_{\{1,2\}}$ -active if  $p$  is  $\theta$ -active for both the first and second octants. In fact, the union of the first and second  $p^\theta$ -octants is a dihedral, which is  $P$ -free and its edge goes through  $p$ .

**Lemma 1** *The boundary of the projection of  $RC H_\theta(P)$  on the  $ZY_\theta$  plane is formed by the points of  $P$  which are  $\theta_{\{1,2\}}$ -active,  $\theta_{\{3,4\}}$ -active,  $\theta_{\{5,6\}}$ -active, and  $\theta_{\{7,8\}}$ -active in the direction defined by the  $ZY_\theta$  plane in the unit circle  $S^1$ . Thus, the four staircases of the boundary of the projection of  $RC H_\theta(P)$  on the  $ZY_\theta$  plane are as follows: the first staircase is formed by the  $\theta_{\{1,2\}}$ -active points, the second staircase is formed by the  $\theta_{\{3,4\}}$ -active points, the third staircase is formed by the  $\theta_{\{5,6\}}$ -active points, and the fourth staircase is formed by the  $\theta_{\{7,8\}}$ -active points.*

**Proof** The proof follows by observing that any point  $p$  in the interior of the projection of  $RC H_\theta(P)$  on the  $ZY_\theta$  plane is dominated by at least one point for each of the four quadrants, and it is so because  $p$  is not  $\theta_{\{s,t\}}$ -active for any  $\{s, t\} \in \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$ , see Fig. 14. Furthermore, the rightmost point in the projection on the  $ZY_\theta$  is both  $\theta_{\{1,2\}}$ -active and  $\theta_{\{7,8\}}$ -active, and the leftmost point is both  $\theta_{\{3,4\}}$ -active and  $\theta_{\{5,6\}}$ -active.  $\square$





**Fig. 14** Projection of  $RCH_\theta(P)$  on the  $ZY_\theta$  plane. The bipartition plane  $\Pi$  is determined by the  $\theta_{\{1,2\}}$ -active point  $p_\theta$  and the  $\theta_{\{5,6\}}$ -active point  $q_\theta$ , which determine the error tolerance functions  $\epsilon_1^{\theta, \Pi}$  and  $\epsilon_2^{\theta, \Pi}$

Let  $p_{left}^\theta$  and  $p_{right}^\theta$  denote, respectively, the leftmost and rightmost points of the projection of  $P$  on the plane  $ZY_\theta$ . For any angle  $\theta$ , let  $L_\theta$  be the list consisting of the  $\theta_{\{1,2\}}$ -active points and the  $\theta_{\{5,6\}}$ -active points of  $P$ , sorted in decreasing order of their  $z$ -coordinate. For simplicity, we assume without loss of generality that no two elements of  $P$  have the same  $z$ -coordinate. Let  $m = O(n)$  denote the number of elements of  $L_\theta$  and let  $z_1, z_2, \dots, z_m$  denote the sorted elements of  $L_\theta$ . The  $\theta_{\{1,2\}}$ -active points of  $L_\theta$ , those of the first staircase of  $RCH_\theta(P)$  are colored red, and the  $\theta_{\{5,6\}}$ -active points, those of the third staircase of  $RCH_\theta(P)$ , are colored blue (see the red and blue staircases of Fig. 14, containing as vertices the red and blue elements of  $L_\theta$ , respectively). We can represent  $L_\theta$  as a standard binary search tree, with extra  $O(1)$ -size data at each node, such that inserting/deleting an element can be done in  $O(\log n)$  time, and also the following queries can all be answered in  $O(\log n)$  time:

- (1) Given an element in the list, retrieve its position.
- (2) Given a position  $j \in \{1, 2, \dots, m\}$ , retrieve the rightmost element in the sublist  $z_1, z_2, \dots, z_j$ .
- (3) Given a position  $j \in \{1, 2, \dots, m\}$ , retrieve the leftmost element in the sublist  $z_j, z_{j+1}, \dots, z_m$ .

For simplicity, we will assume that  $p_{left}^\theta$  is always above  $p_{right}^\theta$  in the  $z$ -coordinate order. Hence, any bipartition plane  $\Pi$  must have  $p_{left}^\theta$  above it and  $p_{right}^\theta$  below it. Furthermore,  $\Pi$  is determined by the closest  $\theta_{\{1,2\}}$ -active point  $p$  above  $\Pi$  and the closest  $\theta_{\{5,6\}}$ -active point  $q$  below  $\Pi$  (see Fig. 14). This is why the list  $L_\theta$  is for the  $\theta_{\{1,2\}}$ -active and  $\theta_{\{5,6\}}$ -active points. If the assumption is not considered, then our arguments must include a similar list with the  $\theta_{\{3,4\}}$ -active and  $\theta_{\{7,8\}}$ -active points for the situations in which  $p_{left}^\theta$  is below  $p_{right}^\theta$ .

The next facts are the keys for the algorithm:

1. Since each point of  $P$  can change its condition of being  $\theta_{\{s,t\}}$ -active a constant number of times, then the total number of times there is a change in some of the four staircases, hence in  $L_\theta$ , is  $O(n)$ . Thus, we have  $O(n)$  intervals of  $[0, \pi]$  with no change in the staircases.

We can then define the sequence  $\Theta$  of the  $N = O(n)$  angles  $0 = \theta_0 < \theta_1 < \theta_2 < \dots < \theta_N = \pi$ , such that for each interval  $[\theta_i, \theta_{i+1})$ ,  $i = 0, 1, \dots, N - 1$  the list  $L_\theta$  do not change.

- For an angle  $\theta \in [0, \pi]$  and a point  $p$  of  $P$ , let  $p^\theta$  be the projection of  $p$  on  $ZY_\theta$ , and let  $\alpha_p$  be the angle formed by the  $X$ -axis and the line through the origin  $O$  and the projection of  $p$  on the  $XY$ -plane. For any point  $q$ , let  $d(q, Z)$  denote the distance from  $q$  to the  $Z$ -axis. We have that

$$d(p^\theta, Z) = d(p, Z) \cdot \cos(\alpha_p - \theta), \tag{1}$$

which is a function depending only on  $\theta$  since  $d(p, Z)$  and  $\alpha_p$  are constants.

- For a fixed angle  $\theta \in [0, \pi]$ , a bipartition of  $P$  by a plane  $\Pi$  induces a partition of the list  $L_\theta = z_1, z_2, \dots, z_m$  into two sublists:  $z_1, z_2, \dots, z_k$  with the elements above  $\Pi$ , and  $z_{k+1}, z_{k+2}, \dots, z_m$  with the elements below  $\Pi$ . And vice versa, every such a partition of  $L_\theta$  into two lists induces a plane  $\Pi$  that bipartitions  $P$ . Let the  $\theta_{\{1,2\}}$ -active point  $p$  and the  $\theta_{\{5,6\}}$ -active point  $q$  be the *witnesses* of this bipartition. That is,  $p$  is the rightmost red element in  $z_1, z_2, \dots, z_k$ , and  $q$  is the leftmost blue element in  $z_{k+1}, z_{k+2}, \dots, z_m$  (see Fig. 14). The error tolerances for this bipartition, denoted  $\epsilon_1^{\theta, \Pi}$  and  $\epsilon_2^{\theta, \Pi}$ , are given by the distances

$$\epsilon_1^{\theta, \Pi} = d(p_{left}^\theta, Z) \pm d(p^\theta, Z) \text{ and } \epsilon_2^{\theta, \Pi} = d(p_{right}^\theta, Z) \pm d(q^\theta, Z), \tag{2}$$

where the  $+$  or  $-$  depends on whether  $p^\theta$  (or  $q^\theta$ ) is to the left or right of the  $Z$ -axis in the  $ZY_\theta$  plane<sup>3</sup>. Note that when moving  $\Pi$  upwards, the functions  $\epsilon_1^{\theta, \Pi}$  and  $\epsilon_2^{\theta, \Pi}$  are non-increasing and non-decreasing, respectively. Hence, to find an optimal  $\Pi$  for a given angle  $\theta$ , we can perform a binary search in the range  $\{k_1, k_1 + 1, \dots, k_2 - 1\} \subset \{1, 2, \dots, m - 1\}$  to find an optimal partition  $z_1, z_2, \dots, z_k$  and  $z_{k+1}, \dots, z_m$  of  $L_\theta$ , where  $k_1$  and  $k_2$  are the positions of  $p_{left}^\theta$  and  $p_{right}^\theta$  in  $L_\theta$ , respectively.

The binary search does the following steps for a given value  $k \in \{k_1, k_1 + 1, \dots, k_2 - 1\}$ : Consider a bipartition plane  $\Pi$  induced by the partition  $z_1, z_2, \dots, z_k$  and  $z_{k+1}, \dots, z_m$  of  $L_\theta$ , and find the witnesses points  $p^\theta$  and  $q^\theta$ , each in  $O(\log n)$  time by using the queries of the tree supporting  $L_\theta$ . Then, compute  $\epsilon_1^{\theta, \Pi}$  and  $\epsilon_2^{\theta, \Pi}$  in constant time. If  $\epsilon_1^{\theta, \Pi} = \epsilon_2^{\theta, \Pi}$ , then stop the search. Otherwise, if  $\epsilon_1^{\theta, \Pi} < \epsilon_2^{\theta, \Pi}$  (resp.  $\epsilon_1^{\theta, \Pi} > \epsilon_2^{\theta, \Pi}$ ), then we increase (resp. decrease) the value of  $k$  accordingly with the binary search and repeat. We return the value of  $k$  visited by the search that minimizes  $\max\{\epsilon_1^{\theta, \Pi}, \epsilon_2^{\theta, \Pi}\}$ . This search makes  $O(\log n)$  steps, each in  $O(\log n)$  time, thus it costs  $O(\log^2 n)$  time.

- Let  $\theta_i$  and  $\theta_{i+1}$  be two consecutive angles of the sequence  $\Theta$ . It may happen for some angle  $\theta \in (\theta_i, \theta_{i+1})$ , and some bipartitioning plane  $\Pi$ , that

$$\epsilon_1^{\theta, \Pi} = \epsilon_2^{\theta, \Pi} < \max \left\{ \epsilon_1^{\theta_i, \Pi}, \epsilon_2^{\theta_i, \Pi} \right\}, \max \left\{ \epsilon_1^{\theta_{i+1}, \Pi}, \epsilon_2^{\theta_{i+1}, \Pi} \right\}.$$

That is, the objective function improves inside the interval  $[\theta_i, \theta_{i+1})$  for the angle  $\theta$ . In fact, this can happen for a linear number of angles. For example, suppose that  $p_{left}^\theta$  and  $p_{right}^\theta$  are sufficiently far from the  $Z$ -axis, and the rest of the elements of  $L_\theta$  are sufficiently close to the  $Z$ -axis. Further suppose that function  $d(p_{left}^\theta, Z)$  is increasing, and function  $d(p_{right}^\theta, Z)$  is decreasing in  $(\theta_i, \theta_{i+1})$ , and that they coincide for some

<sup>3</sup> Notice that there are four possible situations: (i) If  $p_{left}^\theta$  is above  $p_{right}^\theta$ , we can have two possibilities, either:  $p_1$  is to the left of  $p_n$ , or to the right of  $p_n$ . (ii) Analogously, we have the same two possibilities if  $p_{left}^\theta$  is below  $p_{right}^\theta$ . As we pointed out above, for simplicity, we are considering only the case illustrated in Fig. 14.

$\theta \in (\theta_i, \theta_{i+1})$ . For any bipartition plane  $\Pi$ , we will have that the tolerance functions  $\epsilon_1^{\theta, \Pi} \approx d(p_{left}^\theta, Z)$  and  $\epsilon_2^{\theta, \Pi} \approx d(p_{right}^\theta, Z)$  are increasing and decreasing, respectively, and they will also coincide for some angle  $\theta \in (\theta_i, \theta_{i+1})$ .

Considering all the facts above, we next describe an approximation algorithm running in subquadratic time for the 2-fitting problem in 3D, in the case that the orientation of the splitting-plane is fixed. The approximation consists in computing the best bipartition plane for a discrete set of critical angles. That is, we find such a plane for the  $O(n)$  angles of the sequence  $\Theta$ . Our algorithm leaves apart the fact number 4 above, which would imply to consider a quadratic number of critical angles.

2-FITTING ALGORITHM IN 3D. FIXED ORIENTATION OF THE SPLITTING- PLANE

1. By Theorems 6 and 7, and Lemma 1, we compute in  $O(n \log^2 n)$  time and  $O(n \log n)$  space, for all points  $p \in P$  the angular intervals  $I(p)$  in which  $p$  is  $\theta_{\{s,t\}}$ -active for some  $\{s, t\} \in \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$ . We have  $O(1)$  intervals for each  $p$ , each one associated with the corresponding  $\{s, t\}$ . For each  $p$ , we intersect pairwise the intervals of  $I(p)$  to find the set  $I'(p)$  of  $O(1)$  intervals such that for each interval we have:  $p$  is only  $\theta_{\{1,2\}}$ -active;  $p$  is both  $\theta_{\{1,2\}}$ -active and  $\theta_{\{7,8\}}$ -active (i.e.,  $p$  is  $p_{right}^\theta$ );  $p$  is only  $\theta_{\{5,6\}}$ -active; or  $p$  is both  $\theta_{\{3,4\}}$ -active and  $\theta_{\{5,6\}}$ -active (i.e.,  $p$  is  $p_{left}^\theta$ ).
2. We sort in  $O(n \log n)$  time the endpoints of  $I'(p)$  for all  $p \in P$  to obtain the sequence  $\Theta$  of the  $O(n)$  angles  $0 = \theta_0 < \theta_1 < \theta_2 < \dots < \theta_N = \pi$ , such that the list  $L_\theta$  do not change for all  $\theta \in [\theta_i, \theta_{i+1})$ ,  $i = 0, 1, \dots, N - 1$ . Thinking on sweeping the sequence  $\Theta$  with the angle  $\theta$  from left to right, we associate with each  $\theta_i$  the point  $p_i$  of  $P$  and the interval of  $I'(p_i)$  with endpoint  $\theta_i$ . Then, for each  $\theta_i$  we know which point of  $P$  changes some  $\theta_{\{s,t\}}$ -active condition, and the precise conditions it changes.
3. We sweep  $\Theta$  from left to right: As a initial step, for  $\theta = 0$ , we compute the projection of  $RCH_0(P)$  on the plane  $ZY_0$ , the points  $p_{left}^0$  and  $p_{right}^0$  in the projection, and build the list  $L_0$  (as a tree) with the  $\theta_{\{1,2\}}$ -active and  $\theta_{\{5,6\}}$ -active points in  $O(n \log n)$  time. In the next steps, for  $i = 1, 2, \dots, N$ , we have  $\theta = \theta_i$  and we update  $p_{left}^{\theta_i}$  and  $p_{right}^{\theta_i}$  in constant time from  $p_{left}^{\theta_{i-1}}$ ,  $p_{right}^{\theta_{i-1}}$ , and the point  $p_i$  associated with  $\theta_i$ , and update  $L_\theta$  by inserting/deleting  $p_i$  in  $O(\log n)$  time. The color of  $p_i$  (red or blue) is known according to the  $\theta_{\{s,t\}}$ -active condition that  $p_i$  changes. In each step, the initial one and the subsequent ones, we perform the binary search in  $L_\theta$  in  $O(\log^2 n)$  time in order to find the bipartition plane  $\Pi$  that minimizes the value  $\epsilon_\theta = \max\{\epsilon_1^{\theta, \Pi}, \epsilon_2^{\theta, \Pi}\}$ . At the end, we return the angle  $\theta$  of  $\Theta$  (joint with its corresponding optimal plane  $\Pi$ ) such that  $\epsilon_\theta$  is the smallest over all angles of  $\Theta$ .

It is clear that the running time of the above algorithm is  $O(n \log^2 n)$ . We note that the quality of the solution can be improved in terms of  $\epsilon$ -approximations, in the case where the distances of the elements of  $P$  to the  $Z$ -axis are bounded by a constant. Indeed, for  $\epsilon > 0$ , if we split the interval  $[0, \pi]$  into sub-intervals of length  $\delta = \epsilon/D$ , where  $D$  is an upper bound of the absolute value of the first derivative of the functions  $\epsilon_1^{\theta, \Pi}$  and  $\epsilon_2^{\theta, \Pi}$  for all  $\theta$ , and apply the binary search also for  $\theta$  being the endpoints of these sub-intervals, then the solution  $APROX$  given by the algorithm is such that  $APROX - OPT \leq \delta D$ , where  $OPT$  denotes the optimal solution. This implies that  $OPT \leq APROX \leq OPT + \epsilon$ . The running time will be  $O(n \log^2 n + (\pi/\delta) \log^2 n) = O(n \log^2 n + (D\pi/\epsilon) \log^2 n)$ . According to the definitions of  $\epsilon_1^{\theta, \Pi}$  and  $\epsilon_2^{\theta, \Pi}$  (see Eqs. (1) and (2)), a value for  $D$  can be twice the maximum distance of a point of  $P$  to the  $Z$ -axis, and is a constant. Hence, the final running time is  $O(n \log^2 n + \epsilon^{-1} \log^2 n)$ .

## 6 Conclusions, open problems, and future lines of research

We have shown how to compute the rectilinear convex hull of a set  $P$  of  $n$  points in  $\mathbb{R}^3$  in  $O(n \log n)$  time and  $O(n)$  space. The rectilinear convex layers of  $P$  are defined in a recursive way. Thus, if  $P$  has at most  $k$  rectilinear convex layers, they can be computed in  $O(kn \log n)$  time and  $O(n)$  space. We conjecture that there exists a faster algorithm that computes the rectilinear convex layers of  $P$ , whose time complexity is independently of  $k$ .

We proved that the number of times that the set of vertices of  $RC H_\theta(P)$  changes while rotating  $\mathbb{R}^3$  around the  $Z$ -axis is linear. An open problem is to design an optimal algorithm to maintain all the vertices of  $RC H_\theta(P)$  in  $O(n \log n)$  time and  $O(n)$  space.

Several open problems related to the optimization of some parameters of  $RC H_\theta(P)$  are: to determine the angle of rotation  $\theta$  such that the volume of  $RC H_\theta(P)$  is minimum or maximum, and the same problem on the angle  $\theta$  but for the perimeter, area, and the number of connected components of  $RC H_\theta(P)$ .

A future line of research is related to applications in the simplification of the 3D-printing algorithms: there are several ways one can position a model on your print bed, also known as *part orientation*. The best orientation of parts for 3D printing are ones which reduce  $z$ -height, minimize the presence of overhangs and take into account where structural weakness may arise in the  $X$ ,  $Y$  and  $Z$ -axis. Thus, our results can have possible applications on techniques for 3D printers and automatic production of 3D assets, making a contribution to the improvement of the existing algorithms.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Alegría-Galicia, C., Orden, D., Seara, C., Urrutia, J.: On the  $\mathcal{O}$ -hull of planar point sets. *Comput. Geom.: Theory Appl.* **68**, 277–291 (2018)
2. Alegría-Galicia, C., Orden, D., Seara, C., Urrutia, J.: Efficient computation of minimum-area rectilinear convex hull under rotation and generalizations. *J. Global Optim.* **79**(3), 687–714 (2021). <https://doi.org/10.1007/s10898-020-00953-5>
3. Alegría-Galicia, C., Orden, D., Seara, C., Urrutia, J.: Optimizing an oriented convex hull with two directions. In: *European Workshop on Computational Geometry* (2015)
4. Alegría-Galicia, C., Orden, D., Seara, C., Urrutia, J.: Separating bichromatic point sets in the plane by restricted orientation convex hulls. *J. Global Optim.* (2022). <https://doi.org/10.1007/s10898-022-01238-9>
5. Ambrus, G., Nielsen, P., Wilson, C.: New estimates for convex layer numbers. *Discrete Math.* **344**(7), 112424 (2021)
6. Avis, D., Beresford-Smith, B., Devroye, L., Elgindy, H., Guévremont, E., Hurtado, F., Zhu, B.: Unoriented  $\Theta$ -maxima in the plane: complexity and algorithms. *SIAM J. Comput.* **28**(1), 278–296 (1999)
7. Bae, S.W., Lee, Ch., Ahn, H.-K., Choi, S., Chwa, K.-Y.: Computing minimum-area rectilinear convex hull and L-shape. *Comput. Geom.: Theory Appl.* **42**(9), 903–912 (2009)
8. Bhattacharya, B., Das, S., Kameda, T.: Linear-time fitting of a  $k$ -step function. *Discrete Appl. Math.* **280**, 43–52 (2020)

9. Biedl, T., Genç, B.: Reconstructing orthogonal polyhedra from putative vertex sets. *Comput. Geom.: Theory Appl.* **44**(8), 409–417 (2011)
10. Biswas, A., Bhowmick, P., Sarkar, M., Bhattacharya B.B.: Finding the orthogonal hull of a digital object: a combinatorial approach. In: *Proceedings of the 12th International Conference on Combinatorial Image Analysis, IWCIA'08*, pp. 124–135 (2008)
11. Brodal, G.S., Jacob, R.: Dynamic planar convex hull. In: *43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 617–626 (2002)
12. Buchsbaum, A.L., Goodrich, M.T.: Three-dimensional layers of maxima. *Algorithmica* **39**(4), 275–286 (2004)
13. Chazelle, B.: On the convex layers of a planar set. *IEEE Trans. Inf. Theory* **31**, 509–517 (1985)
14. Choi, I., Joo, W., Kim, M.: The layer number of  $\alpha$ -evenly distributed point sets. *Discrete Math.* **343**, 112029 (2020)
15. Díaz-Bañez, J.M., López, M.A., Mora, M., Seara, C., Ventura, I.: Fitting a two-joint orthogonal chain to a point set. *Comput. Geom.: Theory Appl.* **44**(3), 135–147 (2011)
16. Fink, E., Wood, D.: *Restricted-Orientation Convexity*. Monographs in Theoretical Computer Science (An EATCS Series). Springer (2004)
17. Franěk, V., Matoušek, J.: Computing  $D$ -convex hulls in the plane. *Comput. Geom.: Theory Appl.* **42**(1), 81–89 (2009)
18. González-Aguilar, H., Orden, D., Pérez-Lantero, P., Rappaport, D., Seara, C., Tejel, J., Urrutia, J.: Maximum rectilinear convex subsets. *SIAM J. Comput.* **50**(1), 145–170 (2021)
19. Güting, R.H.: *Conquering contours: efficient algorithms for computational geometry*. Ph.D. thesis, Fachbereich Informatik, Universität Dortmund (1983)
20. He, M., Nguyen, C.P., Zeh, N.: Maximal and convex layers of random point sets. *LATIN 2018, Lecture Notes in Computer Science*, vol. 10807
21. Houle, M.E., Toussaint, G.T.: Computing the width of a set. *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(5), 761–765 (1988)
22. Kung, H.-T., Luccio, F., Preparata, F.P.: On finding the maxima of a set of vectors. *J. ACM* **22**(4), 469–476 (1975)
23. Ottmann, T., Soisalon-Soininen, E., Wood, D.: On the definition and computation of rectilinear convex hulls. *Inf. Sci.* **33**(3), 157–171 (1984)
24. Peláez, C., Ramírez-Vigueras, A., Seara, C., Urrutia, J.: On the rectilinear convex layers of a planar set. In: *Mexican Conference on Discrete Mathematics and Computational Geometry, 60th Birthday of Jorge Urrutia* (2013)
25. Pérez-Lantero, P., Seara, C., Urrutia, J.: Rectilinear convex hull of points in 3D. In: *14th Latin American Theoretical Informatics Symposium, São Paulo, Brazil, January 5–8, (2021)*, LNCS 12118, pp. 296–307. <https://doi.org/10.1007/978-3-030-61792-9-24>
26. Preparata, F.P., Shamos, M.I.: *Computational Geometry: An introduction*. Springer (1985)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.