Rectilinear Convex Hull with minimum area

Carlos Alegría-Galicia[3], Tzolkin Garduño[3], Areli Rosas-Navarrete[3],
Carlos Seara[2], and Jorge Urrutia[1]

[1] Universidad Nacional Autónoma de México, Instituto de Matemáticas[*]
[2] Universitat Politècnica de Catalunya[**]
[3] Universidad Nacional Autónoma de México, Posgrado en Ciencia e
Ingeniería de la Computación

{alegria_c,garduno_t,areli}@uxmcc2.iimas.unam.mx,
carlos.seara@upc.edu,urrutia@matem.unam.mx

Abstract. Let P be a set of n points in $\mathbb{R}^2$. We solve the problem of computing an orientation of the plane for which the rectilinear convex hull of P has minimum area in optimal $\Theta(n \log n)$ time and $O(n)$ space.

Keywords: rectilinear convex hull, orthoconvexity, optimization

1 Introduction

The interest in the rectilinear convex hull of planar point sets arises from the study of *ortho-convexity* [13], also known in the literature as *rectilinear*, *x-y*, or *orthogonal* convexity. This non-traditional notion of convexity has been widely studied since its formalization in the early eighties, and has found applications in theoretical research fields such as polyhedra reconstruction [15] and fixed point theory [16], as well as in the study of practical problems such as digital image shape analysis [4] and VLSI circuit layout design [14].

A set is said to be *ortho-convex* if its intersection with every horizontal or vertical line is connected. As it is easy to see, an ortho-convex set might be non-convex or disconnected. The latter property hinders the definition of an ortho-convex closure that resembles the properties of the traditional convex hull and as a consequence, several definitions have been presented by different authors [11]. Throughout this paper, we will use the *maximal ortho-convex*, or *mr-convex* hull stated by Ottmann et al. [11] (see also Matoušek et al. [10]), which is defined as follows.

An *orthogonal wedge* is the intersection of two open half-panes whose supporting lines are orthogonal to each other. Let X_θ and Y_θ denote respectively, the X and Y coordinate axes counter-clockwise rotated by θ degrees. We call *θ-orientation* to the coordinate system formed by X_θ and Y_θ. An *orthogonal θ-wedge* is an orthogonal wedge whose supporting lines are parallel to the axes of

the first quadrant of the θ-orientation of the plane. Hereafter, let $P \subset \mathbb{R}^2$ denote a set of n points in general position (i.e., no three collinear points in P). An orthogonal wedge is called *P-free* if it does not contain points of P. Let $W_\theta(P)$ be the set of all *P*-free orthogonal θ-wedges, and $\mathcal{W}_\theta(P)$ be the set

$$W_\theta(P) \cup W_{\theta+\frac{\pi}{2}}(P) \cup W_{\theta+\pi}(P) \cup W_{\theta+\frac{3}{2}\pi}(P). \tag{1}$$

The *Rectilinear Convex Hull* of P is the set

$$\mathcal{RH}_\theta(P) = \mathbb{R}^2 - \bigcup_{w \in \mathcal{W}_\theta(P)} w. \tag{2}$$

Figure 1(a) shows the rectilinear convex hull of a point set. Note in Figure 1(b) that the set $\mathcal{W}_\theta(P)$ changes along with θ and thus, $\mathcal{RH}_\theta(P)$ is an orientation-dependent shape.

(a)   (b)

Fig. 1: $\mathcal{RH}_\theta(P)$ changes along with θ.

From this property, several optimization problems naturally arise for $\mathcal{RH}_\theta(P)$ when considering the complete set of θ-orientations of the plane. In this paper we consider the problem of computing a value of θ for which $\mathcal{RH}_\theta(P)$ has minimum area. We show that the set $\{\theta : \theta \in [0, 2\pi)\}$ can be divided into an ordered set

$$\{I_1, I_2, \ldots, I_{O(n)}\}$$

of intervals such that, we can compute the angle $\theta \in I_1$ which minimizes the area of $\mathcal{RH}_\theta(P)$ in $O(n)$ time. We perform the computation of the angle of optimal area for the subsequent intervals in constant time. To obtain the set of intervals we use optimal $\Theta(n \log n)$ time and $O(n)$ space. Our work is based on techniques from Bae et al. [2], Avis et al. [1], and Díaz-Báñez et al. [5]. Our result improves the $O(n^2)$ time complexity solution presented recently by Bae et al. [2].

Outline of the paper. In Section 2 we show how to compute and maintain $\mathcal{RH}_\theta(P)$ over all the θ-orientations of the plane. In Section 3 we present the algorithm for computing a θ-orientation for which $\mathcal{RH}_\theta(P)$ has minimum area. Finally, in Section 4 we discuss our concluding remarks.

2   The Unoriented Rectilinear Convex Hull

In this section we present an algorithm to compute and maintain $\mathcal{RH}_\theta(P)$ over all θ-orientations of the plane in $\Theta(n \log n)$ time.

2.1   Preliminaries

A *vertex* of $\mathcal{RH}_\theta(P)$ is a point in P that lies on the boundary of $\mathcal{RH}_\theta(P)$. For any fixed θ, the computation of $\mathcal{RH}_\theta(P)$ has a close relation to the *Set Maxima Problem* [8,12], since the set of vertices of $\mathcal{RH}_\theta(P)$ is equal to the set $\mathcal{V}_\theta(P)$ of maximal points of P with respect to vector dominance in the four quadrants defined by X_θ and Y_θ [2,11].

Since we can specify $\mathcal{RH}_\theta(P)$ using $\mathcal{V}_\theta(P)$, it is possible to keep track of the changes in the set of vertices of $\mathcal{RH}_\theta(P)$ over all θ-orientations in the plane, by maintaining $\mathcal{V}_\theta(P)$ as the plane rotates in a counter-clockwise direction. The maintenance of this set can be done using the results given in [1,?] in the manner we describe bellow.

We call *apex* of an orthogonal wedge to the intersection point of its supporting lines. Every point $p \in \mathcal{V}_\theta(P)$ is the apex of a P-free orthogonal θ-wedge contained in $\mathcal{W}_\theta(P)$. Figure 2 shows us a point p that is vertex of $\mathcal{RH}_\theta(P)$ in the interval $I_p = [\theta, \theta')$, and the orthogonal θ and θ'-wedges with apex in p. The endpoints of I_p mark the *in-* and an *out*-events of p, i.e., the values of θ when p belongs and stops being in $\mathcal{V}_\theta(P)$. Moreover, a point can be the apex of an orthogonal θ-wedge during at most three intervals of θ-orientations.

Fig. 2: A set P and $\mathcal{RH}_{\theta''}(P)$.

Theorem 1 (Díaz-Bañez et al. [5]). *The set of intervals of orientations in which the elements of P are maximal with respect to vector dominance, and the ordered set of in- and out-events, can be computed in optimal $\Theta(n \log n)$ time and $O(n)$ space.*

Let us denote by $V_\theta(P)$ to the set of maximal points of P in the first quadrant defined by X_θ and Y_θ. Since $\mathcal{RH}_\theta(P)$ is an ortho-convex set, it is monotone with

respect to X_θ. Thus, the points of P in $V_\theta(P)$ can be relabelled as $v_1, \cdots, v_m$ in increasing order according to X_θ. Let $w_i(\theta)$ denote the P-free orthogonal θ-wedge supported by two consecutive points $v_i, v_{i+1} \in V_\theta(P)$. We say that two wedges $w_i(\theta)$ and $w_j(\theta + \pi)$ are *opposite* to each other. Note in Figure 3b that, if the intersection of $w_i(\theta)$ and $w_j(\theta + \pi)$ is non empty, then $\mathcal{RH}_\theta(P)$ is disconnected. In such case we say that $w_i(\theta)$ and $w_j(\theta + \pi)$ *overlap*, and denote its intersection as $O_\theta(i, j)$.

(a) Full interval start   (b)   (c) Full interval end

Fig. 3: An overlap disconnects $\mathcal{RH}_\theta(P)$. The ends of the overlap full interval.

Let $\mathcal{O}_\theta(P)$ be the set of all overlaps $O_\theta(i, j)$ of $\mathcal{RH}_\theta(P)$ in the θ-orientation of the plane. It is easy to see that $\mathcal{O}_\theta(P)$ can be computed from $V_\theta(P)$ in $O(n)$ time. Thus, $\mathcal{RH}_\theta(P)$ can be computed in $\Theta(n \log n)$ time and $O(n)$ space [8,12] for a fixed value of θ.

To maintain the set $\mathcal{O}_\theta(P)$ we will compute an ordered list of angles that resembles the one computed for the in- and out- events. Since overlap events do not necessarily match up with in- or out-events [2], they have to be computed independently.

2.2   Overlaps characterization

As the θ-orientation of the plane *rotates* counter-clockwise from 0 to 2π, the apex of every P-free orthogonal θ-wedge supported by at least two points in $\mathcal{V}_\theta(P)$ traces a circular arc that can be oriented as shown in Figure 4.

Definition 1. *The arc-chain $\mathcal{A}(P)$ of P is the closed curve formed by the union of the set of arcs traced by the apexes of the wedges supported by at least two elements of P which belong to $\mathcal{V}_\theta(P)$ for some $\theta \in [0, 2\pi)$.*

The arc-chain can be computed using the in- and out-events list in $O(n)$ time, since every arc is a segment of a circumference with diameter in consecutive points of the sets $V_\theta, V_{\theta + \frac{\pi}{2}}, V_{\theta + \pi}$ and $V_{\theta + \frac{3\pi}{4}}$, and for every arc, its ends correspond to the in- and out-events of one of the points determining the diameter of the circumference to which the arc belongs (see Figures 4a and 4c). Thus, each arc can be associated with an interval of orientations in which, the wedge that traces the arc, is part of the rectilinear convex hull (see Figure 5).

(a) In-event   (b)   (c) Out-event

Fig. 4: The apexes path as the θ-orientation changes.

Fig. 5: The interval of orientations of an arc.

Let $\{e_1, \cdots, e_h\}$, be the set of edges of the convex hull of P in counter-clockwise order. A *subchain* $A_i(P) \in \mathcal{A}(P)$ is a subsequence of consecutive elements of $\mathcal{A}(P)$, whose endpoints are the endpoints of a convex hull edge e_i. It is easy to see that $A_i(P)$ is monotone in the direction determined by e_i. Thus, the orthogonal projection of $A_i(P)$ on e_i defines a total order $(\prec_i)$ on the set of endpoints of its arcs. Moreover, using the convex hull edges and $(\prec_i)$, it is possible to label the arcs in $\mathcal{A}(P)$ starting by the leftmost point of P in counter-clockwise order, in such a way that $\mathcal{A}(P) = \langle a_1, \cdots, a_l \rangle$ (Figure 7a). Let us consider that the ends of the arcs in any subchain $A_i(P)$ are labeled as $p_1, \ldots, p_t$ such that if $r < s$, then $p_r \prec_i p_s$. A *link* $\ell_{r,s}$ is a subsequence $p_r, \ldots, p_s$ of $A_i(P)$ such that for $r < t < s$, $p_r, p_s \in P$, $p_r \prec_i p_t \prec_i p_s$ and $p_t \notin P$.

Now we characterize an overlap using the arc-chain. Consider four points $v_i, v_{i+1} \in V_\theta(P)$ and $v_j, v_{j+1} \in V_{\theta+\pi}(P)$ that support two overlapping wedges (Figure 3). Let us observe that if two opposite wedges in $\mathcal{W}_\theta(P)$ overlap, since these wedges are P-free, the arcs traced by their apices belong to *links* that intersect (see Figure 7a). Now, the overlap between the wedges $w_i(\theta)$ and $w_j(\theta + \pi)$ is possible only in an interval $[\theta', \theta'']$ where the θ'-orientation is parallel to the segment $\overline{v_{j+1}v_{i+1}}$ and in θ''-orientation the Y-axis is parallel to the segment

$\overline{v_iv_j}$, see Figures 3a and 3c, respectively. We call *full overlap interval* to this interval of orientations and we say that this interval is a proper interval when $\theta' \le \theta''$.

Any arc a_j corresponds, for some interval $I_j = [\theta_{j1}, \theta_{j2}]$, to a particular P-free orthogonal θ-wedge, $w_i(\theta)$ for every $\theta \in I_j$ (Figure 5). It is important to notice that I_j is different from the intervals of the points that support $w_i(\theta)$, indeed I_j is contained in the intersection of the intervals of these points.

Since an overlap is possible only between opposite wedges, in order to determine if two given arcs a_j, a_k admit opposite wedges, we define the *facing interval* for them as the interval $I_f = [\theta_{j1} + \pi, \theta_{j2+\pi}] \cap [\theta_{k1}, \theta_{k2}]$.

Lemma 1. *(An overlap interms of arcs) Given two arcs a_j and a_k, with its corresponding intervals $I_j = [\theta_{j1}, \theta_{j2}]$ and $I_k = [\theta_{k1}, \theta_{k2}]$. Let $w_q(\theta)$ denote the orthogonal θ-wedge whose apex traces a_j for $\theta \in I_j$ and let $w_r(\theta)$ denote the respective wedge of a_k for any $\theta \in I_k$. This wedges overlap if and only if the full overlap interval is proper, the facing interval is not empty and the intersection between the full overlap interval and the facing interval is a proper interval.*

Then, if there is an overlap between the wedges of two given arcs, the interval of the overlap is the intersection between the full interval, the facing interval, and the intervals of both arcs.

Let $I_{i,j}$ be the the interval of θ for which $w_i(\theta)$ and $w_j(\theta+\pi)$ overlap. We call the ends of $I_{i,j}$ the *start-* and *stop-* events of $O_\theta(i,j)$. The thick arc segments show the overlap interval in Figure 5).

Fig. 6: The interval of orientations of an overlap $I_{i,j} = (\theta', \theta'')$.

Once we have characterized the overlaps, let us notice that if an overlap takes place, then the links that contain the arcs involved in the overlap had intersected. In consequence, it is possible to compute the overlaps and its corresponding events by computing the intersection points between links, but before using the

intersections, it is critical to show that there are $O(n)$ intersection points between links in order to keep the algorithm subquadratic. The following paragraphs show some properties that help us to show the linear number of intersections between links in the arc chain.

Using the fact that every point in any subchain $A_i(P)$ is an apex of an orthogonal P-free θ-wedge, the following lemma follows easily:

Lemma 2. *Let $a, b, c \in A_i(P)$ such that $a \prec_i b \prec_i c$. Then, $\frac{\pi}{2} \leq \angle abc < \pi$.*

Fig. 7: The *arc-chain* $\mathcal{A}(P)$ of P.

As a consequence of Lemma 2, any link $\ell_{r,s}$ is contained in the disk that has the segment $\overline{p_r p_s}$ as diameter (the link-disk $D_{r,s}$), thus the number of links that intersect the link-disk is greater than or equal to the number of links that intersect $\ell_{r,s}$. We denote by $d(r, s)$ to the diameter of $\ell_{r,s}$.

Let $\mathcal{D}(P) = \{D_0, D_1, \ldots, D_h\}$ be the set of disks such that D_i has the convex hull edge e_i as a diameter and radius r_i. Using a result given in [1] we show the following lemma.

Lemma 3. *Given $p, q \in P$, the disk that has $\overline{pq}$ as diameter and radius r is intersected by at most 28 disks $D_i \in \mathcal{D}(P)$ such that $r_i \geq r$.*

Into each of the disks in $\mathcal{D}(P)$ that intersect a link-disk $D_{r,s}$, there is a constant number of links with diameter greater than $d(r, s)$ that intersect $\ell_{r,s}$.

Lemma 4. *Let $D_i \in \mathcal{D}(P)$ be a disk of radius r_i that intersects some given link disk $D_{r,s}$ of radius $r < r_i$. There are at most five links in D_i that intersect $\ell_{r,s}$ with diameter greater than $d(r, s)$.*

Proof. Let R be the closed region bounded by the lines that project $D_{r,s}$ on the line that crosses the endpoints of e_i (see Figure 8), since subchains are monotone, then the links in $A_i(P)$ that intersect $D_{(}r, s)$ must have at least one endpoint in R.

Now let us consider the links of $A_i(P)$ with endpoints in R. For simplicity we suppose that all these links have greater diameter than $d(r, s)$. It is easy to see

Fig. 8: Construction to count the links in D_i that intersect another link $\ell_{r,s}$

that in order to have many links into region R, the endpoints of the links should be placed in the configurations showed in Figure 9. The configuration showed by Figure 9a, considers three points p_{i-1}, p_i, p_{i+1} such that p_i is a *peak* (i. e. the distances from the other points to e_i are smaller then the distance between p_i and e_i). In this case, with the help of a circle with diameter parallel to e_i and lenght $d(r, s)$ that passes through p_{i-1} it is possible to show that if p_i must be at distance grater than $d(r, s)$ from p_{i-1} and from p_{i+1} then p_i cannot be part of a link in $A_i(P)$, since the angle $\angle p_{i-1}p_ip_{i+1}$ would be less than $\frac{\pi}{2}$, contradicting Lemma 2.

Since there are no crests into region R, there is at most one *valley*, like the one showed in Figure 9b, therefore there are at most five links in $A_i(P)$ that intersect $\ell_{r,s}$.

The open area bounded by $A_i(P)$ and e_i is P-free since it is covered by P-free θ-wedges. Thus, two intersecting links have at least two intersection points, and by Lemma 2 this number is tight, as none of the intersecting links can cross a line segment joining its intersection points.

As a consequence of lemmas 2, 3 and 4, we have:

Lemma 5. *Any link $\ell_{j,k}$ is intersected at most 280 times by links of diameter greater than or equal to $d(j, k)$.*

Considering the previous results, and that we have a linear number of links, we have the following result that is a central tool for computing the start- and stop-events list in $O(n \log n)$ time.

Theorem 2. *There are $O(n)$ intersections between links in $\mathcal{A}(P)$.*

2.3   Computing the start- and stop- events

Since in the previous section it has been proved that there are a linear number of intersection points between arcs, in this section we describe how to compute the start- and stop- overlap event list.

Fig. 9: A link $\ell_{j,k}$ can be intersected by at most five links contained in some $D_i \in \mathcal{D}(P)$.

Let us color the arcs in $\mathcal{A}(P)$ in the following way: red if the arc belongs to a subchain that corresponds to an edge in the upper chain of the convex hull, and the arc will be blue otherwise. Then we compute the bichromatic intersections using the Bentley and Ottmann plane sweep algorithm [3] in $O(n \log n)$.

Every intersection point will be associated with the respective links. After sorting these intersection points lexicographically according to the labels of the links involved in the intersection, we will be able to track corresponding pairs of links (Figure 10), since this links correspond to a couple of intersection points that involve the same links. We say that each of this couples form an *alternate link*.

For each alternate link let $A_\alpha = \alpha_1, \alpha_2, \ldots, \alpha_v$ be the set of red arcs and $A_\beta = \beta_1, \beta_2, \ldots, \beta_w$ the set of blue arcs comprised in the alternate link.

Let us recall that an arc α_i is associated with an interval of orientations $(\theta_{\alpha i_1}, \theta_{\alpha i_2})$. We say that $\theta_{\alpha i_1}$ is the *left* endpoint and $\theta_{\alpha i_2}$ is the *right* endpoint of the interval. Using these intervals we can verify if the conditions of Lemma 1 are fulfilled. Bellow we describe how the verification is done in $O(v + w)$ time.

Let us consider two functions: A boolean function verifyOverlap that will accept a couple of arcs as parameters and will return TRUE if the conditions for Lemma 1 are fullfilled by both arcs. And function overlap that takes two arcs and returns the interval of orientations in which the overlap takes place between the arcs. Each overlap interval, the corresponding arcs and the points of P that support the θ-wedges that overlap, are stores in an ordered list that we call *OverlapList*.

Fig. 10: Ejemplo de cadenas intercaladas

SEARCHOVERLAP(A_α, A_β)

```
1   i, m = 1
2   while (i ≤ Aα.length)  and  (m ≤ Aβ.length)
3       if (αi. right < βm. left)
            // disjoint intervals. Take next arc in Aα
4           i = i + 1
5       elseif (βm. right < αi. left)
            // disjoint intervals. Take next arc in Aβ
6           m = m + 1
7       elseif (αi. left < βm. left  and  αi. right > βm. right)
            // interval of βm contained in αi's interval
8           if verifyOverlap(αi, βm)
                // Add overlap to list
9               OverlapList. add(αi, βm, overlap(αi, βm))
                // end if. Take next arc in Aβ
10          m = m + 1
11      elseif (βm. left < αi. left  and  βm. right > αi. right)
            // αi's interval is contained in βm's.
12          if verifyOverlap(αi, βm)
                // Add overlap to list
13              OverlapList. add(αi, βm, overlap(αi, βm))
                // end if. Take next arc in Aα
14          i = i + 1
15      elseif (βm. left < αi. left)  and  (βm. right < αi. right)
            // alternated intervals
16          if verifyOverlap(αi, βm)
                // Add overlap to list
17              OverlapList. add(αi, βm, overlap(αi, βm))
                // end if. Take next arc in Aβ
18          m = m + 1
19      else
20          i = i + 1
```

Lemma 6. *Computational complexity of* SEARCHOVERLAP *is* $O(v + w)$*, where* v *and* w *are the number of arcs in the alternate link.*

Proof. The algorithm finishes when at least one of the chains has been completely traversed. On each iteration is its possible to leave behind one arc of one link. Functions verifyOverlap and overlap take constant time to execute each time.

3   Minimum area algorithm

The list of event points obtained in the previous sections generate a set of angular intervals (θ_1, θ_2), in which the set of vertices of $\mathcal{RH}_\theta(P)$ and the set of overlaps in $\mathcal{RH}_\theta(P)$ remains unchanged.

For any $\theta \in (\theta_1, \theta_2)$, the area of $\mathcal{RH}_\theta(P)$ is given by the formula in [2]. The formula has three main components: the first is a polygon constructed by joining counterclockwise consecutive vertices of $\mathcal{RH}_\theta(P)$ (Figure 11a), the second is the sum of the areas of triangles formed by consecutive vertices of $\mathcal{RH}_\theta(P)$ and the apex of the P-free θ-wedge that they support (Figure 11b), finally the third one is the sum of the overlaps area (Figure 11c).

(a) The polygon   (b)   (c) Out-event

Fig. 11: Updating the area of the polygon

It takes $O(n)$ time to compute the area of the polygon for the first time. This polygon changes with each in- and out-event. Let us say that a point p is getting in the $\mathcal{RH}_\theta(P)$, between points v_i, v_{i+1} (see Figure 12a). The area of the polygon that includes p can be computed in constant time from the area of the previous polygon by subtracting the area of the triangle $\triangle(v_i, p, v_{i+1})$ (Figure 12b). In the out event of p, the area can be computed adding the area of the same triangle (Figure 12c).

Using the in and out-event list, it is possible to compute the set of triangles that conform the second element of the formula. This list has a linear number of changes, since on each in-event two triangles are added to the list and a triangle is deleted form it and the same amount of triangles changes on each out-event.

The areas of the overlaps can be computed in a way similar to the triangles, but with the help of the ordered list of start- and stop-events.

Fig. 12: Updating the area of the polygon

For each interval without changes, we update the corresponding values of the formula in constant time by subtracting or adding new constant values.

There can be more than one θ-orientation in which $\mathcal{RH}_\theta(P)$ has minimum area, but our algorithm is able to report them all.

From the discussion above, and the fact that the convex hull of P can be computed from the rectilinear convex hull of P in $O(n)$ time, we obtain the following:

Theorem 3. *Computing the set of orientations for which the rectilinear convex hull of P has minimum area, can be done in optimal $\Theta(n \log n)$ time and $O(n)$ space.*

4   Concluding remarks

We have proved that the orientation in which $\mathcal{RH}_\theta(P)$ has minimum area can be obtained in optimal $O(n \log n)$ time, since with the help of $O(n \log n)$ time preprocessing it is possible to maintain $\mathcal{RH}_\theta(P)$ as the θ-orientation changes from cero to 2π.

Our result improves the quadratic algorithm given by Bae et al. [2]. It would be interesting to apply this results to optimization problems in bichromatic sets, like finding the θ-orientation in which the rectilinear convex hull of the red points is empty of blue points or contains the minimum number of blue points, and the minimum or maximum area rectilinear red convex hull, for example.

It is almost immediate to extend this algorithm to dynamic pointsets and it would be very interesting to consider the problem of maintaining the rectilinear hull of a moving pointset.

References

1. Avis, D., Beresford-Smith, B., Devroye, L., Elgindy, H., Guévremont, E., Hurtado, F., Zhu, B.: Unoriented Θ-maxima in the plane: complexity and algorithms. SIAM J. Comput. Vol. 28 Iss. 1, 278–296 (1999)
2. Bae, S.W., Lee, C., Ahn, H-K., Choi, S., Chwa, K-Y.: Computing minimum-area rectilinear convex hull and L-shape. Computational Geometry: Theory and Applications Vol. 42 Iss. 3, 903–912 (2009)

3. Bentley, J., Ottmann, T.: Algorithms for reporting and counting geometric intersections. IEEE Trans. Computers C-28, 643–647 (1979)
4. 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, pp. 124–135. Springer-Verlag Berlin, Heidelberg (2008)
5. Díaz-Báñez, J.M., López, M.A., Mora, M., Seara, C., Ventura, I.: Fitting a two-joint orthogonal chain to a point set. Computational Geometry: Theory and Applications Vol. 44 Iss. 3, 135–147 (2011)
6. Fink, E., Wood, D.: Strong restricted-orientation convexity. Geometriae Dedicata Vol. 69, 35–51 (1998)
7. Franěk, V., Matousěk, J.: Computing D-convex hulls in the plane. Computational Geometry: Theory and Applications Vol. 42, 81–89 (2009)
8. Kung, H.T., Luccio, F., Preparata, F.P.: On finding the maxima of a set of vectors. Journal of the ACM Vol. 22, 469–476 (1975)
9. Martynchik, V., Metelski, N., Wood, D.: O-convexity: computing hulls, approximations, and orientation sets. In: Proceedings of the 8th Canadian Conference on Computational Geometry, pp. 2–7. Carleton University Press (1996)
10. Matoušek, J., Plecháč, P.: On functional separately convex hulls. Discrete and Computational Geometry Vol. 19, 105–130 (1998)
11. Ottman, T., Soisalon-Soisinen, E., Wood D.: On the definition and computation of rectilinear convex hulls. Information Sciences Vol. 33, 157–171 (1984)
12. Preparata, F., Shamos, M.I.: Computational Geometry: An Introduction. Springer-Verlag (1985)
13. Rawlins, G.J.E., Wood, D.: Ortho-convexity and its generalizations. In: Toussaint, G.T. (editor). Computational Morphology: A Computational Geometric Approach to the Analysis of Form, pp. 137–152. Elseiver Science Publishers B.V., North-Holland (1988)
14. Uchoa, E., De Aragão, M.P., Ribeiro, C.C.: Preprocessing Steiner problems from VLSI layout. Networks Vol. 40 Iss. 1, 38–50 (2002)
15. Biedl, T., Genç, B.: Reconstructing orthogonal polyhedra from putative vertex sets. Computational Geometry: Theory and Applications Vol. 44 Iss. 8, 409–417 (2011)
16. Khamsi, M.A., Kirk, W.A.: An introduction to metric spaces and fixed point theory. Wiley-Interscience (2001)