Noname manuscript No.
(will be inserted by the editor)

An Efficient Algorithm for Determining the Connected Orthogonal Convex hull

Nguyen Kieu Linh · Phan Thanh An · Tran Van Hoai

Received: date / Accepted: date

Abstract The Quickhull algorithm for finding the convex hull of a finite set of points was independently conducted by Eddy in 1977 and Bykat in 1978. Inspired by the idea of this algorithm, we present a new efficient algorithm for finding extreme points of the connected orthogonal convex hull of a finite set of points that still keeps advantages of the quickhull algorithm. Consequently, our algorithm runs faster than the others (the algorithms introduced by Montuno and Fournier in 1982 and by An, Huyen and Le in 2020). We also show that the expected complexity of the algorithm is $O(n \log n)$, where n is the number of points.

Keywords Convexity · Extreme points · Quickhull algorithm · Orthogonal convex hulls · $x - y$ convex hulls.

Mathematics Subject Classification (2000) MSC 52A30 · MSC 52B55 · MSC 68Q25 · MSC 65D18.

1 Introduction

Orthogonal convexity (rectilinearity, or (x, y) convexity, or $x - y$ convexity) is one of the most extensively subjects studied in computational geometry and convex analysis. It is widely used in research fields, including illumination [1], polyhedron reconstruction [9], geometric search [16], and VLSI circuit layout

N. K. Linh[1]
[1]Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
E-mail: linhnk@ptit.edu.vn

P. T. An and T. V. Hoai[2,3]
[2]Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
[3]Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam
E-mail: thanhan@hcmut.edu.vn, hoai@hcmut.edu.vn

design [17], digital images processing [15]. The concept of orthogonal convex hull of a set was first mentioned in 1959 by Unger [18]. By 1983, Montuno and Fournier introduced efficient algorithms for computing the (x, y)-convex hulls of a finite set of planar points, an (x, y)-polygon and of a set of (x, y)-polygons under various conditions [11]. The condition under which the (x, y)-convex hull exists is given and an algorithm for testing if the given set of (x, y)-polygons satisfies the condition is also presented. Since then, there exist several algorithms for finding orthogonal convex hull proposed [10,12], and [14]. Recently, in 2020, An, Huyen and Le give a condition that ensures the unique of the orthogonal convex hull of a finite planar point set and determine the hull through its extreme points [3]. Their efficient algorithm is modified from the Graham's convex hull algorithm. They also show that the lower bound of computational complexity of such algorithms is $O(n \log n)$.

The Quickhull algorithm [4,6,7,13] determines the convex hull of a finite planar set of points. The worst case complexity of the algorithm is $O(n^2)$ and its average time is $O(n \log n)$. Quickhull is known as a powerful algorithm, which runs in practice much faster than in the worst case. The recursive nature of the Quickhull algorithm allows a fast implementation. This algorithm can also be easily designed as a parallel algorithm for finding convex hull of the point set. Recognizing the effectiveness of the Quickhull algorithm, in this paper, we apply the idea of this algorithm and its improved algorithm [8] to propose an algorithm, namely $\mathcal{O}$-QUICKHULL, for determining extreme points of the orthogonal convex hull of a finite set of points and compare it with the algorithm [3] and Montuno and Fournier's algorithm [11]. We also show that the expected complexity of $\mathcal{O}$-QUICKHULL is $O(n \log n)$, where n is the number of points.

The paper consists of several sections. Section 2 presents some concepts of connected orthogonal convexity that will be used in this paper. Section 3 introduces the definition of directed orthogonal lines and some other concepts. Section 4 is devoted to an algorithm based on the Quickhull algorithm to determine extreme points of the connected orthogonal convex hull of a finite planar point set and its expected complexity. Section 5 closes the paper with some numerical experiments.

2 Connected orthogonal convex hulls and their properties

Throughout this paper, we focus on the problem of determining the *connected orthogonal convex hull* of a finite planar point set.

Let be given $p, q, t \in \mathbb{R}^2$, denote $[p, q] := \{(1 - \lambda)p + \lambda q : 0 \leq \lambda \leq 1\}$, pq the straight line through the points p and q and $\mathrm{dist}(t, pq)$ the Euclidean distance from t to the line pq. We denote by x_p and y_p respectively the x-coordinate and y-coordinate of p. As usual, $\mathrm{dist}(p, q) := \sqrt{(x_p - x_q)^2 + (y_p - y_p)^2}$.

Definition 1 ([18]) A set $S \subset \mathbb{R}^2$ is said to be *orthogonal convex* if its intersection with any horizontal or vertical line is convex.

S is said to be *connected orthogonal convex* if it is orthogonal convex and connected.

Definition 2 ([14]) A *connected orthogonal convex hull* of S is a smallest connected orthogonal convex set containing S.

We define a line to be *rectilinear* if the line is parallel to either x-axis or y-axis. A half line or a line segment are *rectilinear* if the lines on which they lie are rectilinear.

Let $p \neq q$ be two given points in the plane. We define $\mathrm{l}(p, q)(x_p \neq x_q, y_p \neq y_q)$ through p, q to be union of two rectilinear half lines having the same starting point. If $x_p = x_q$ or $y_p = y_q$ then $\mathrm{l}(p, q)$ is the line through p and q. The set $\mathrm{l}(p, q)$ is called the *orthogonal line* through p and q. The common point of two the rectilinear half lines of $\mathrm{l}(p, q)$ is called the *vertex* of $\mathrm{l}(p, q)$. We also denote by $l^v(p, q)$ the orthogonal line $\mathrm{l}(p, q)$ having the vertex v.

An orthogonal line $\mathrm{l}(p, q), (x_p \neq x_q, y_p \neq y_q)$ separates the plane into two regions. The quadrant region together with the orthogonal line $\mathrm{l}(p, q)$ will be called a *quadrant* determined by the orthogonal line.

Definition 3 ([3]) Given a set $S \subset \mathbb{R}^2$. An $\mathrm{l}(p, q)$ is an orthogonal supporting line (O-support, for brevity) of a set S (p and q might not belong to S) if the intersection of $\mathrm{l}(p, q)$ with S is non-empty and either all points of $S \setminus (S \cap \mathrm{l(p, q)})$ are not on the quadrant of $\mathrm{l}(p, q)(x_p \neq x_q, y_p \neq y_q)$, or all points of $S \setminus (S \cap \mathrm{l(p, q)})$ are on one open half plane which is determined by the line $\mathrm{l}(p, q)(x_p = x_q, \text{or}\, y_p = y_q)$.

Two $\mathcal{O}$-supports of a set S is said to be *opposite* if their half lines meet in two distinct points.

We denote by $\mathcal{F}(S)$ the set of all connected orthogonal convex hulls of S. For $E \in \mathcal{F}(S)$, if there exist two opposite O-supports H and L of S intersecting in two distinct points, say p and q, with $x_p \neq x_q, y_p \neq y_q$, then there exists a monotone path connecting p and q in E. We define all points on such path (not including p and q) to be *semi-isolated* points of E. If there exists an element of $\mathcal{F}(S)$ that has no semi-isolated point then $\bigcap_{E \in \mathcal{F}(S)} E$ is a connected orthogonal convex hull of S. Therefore, $\mathcal{F}(S)$ has only one element, denoted it by $\mathrm{COCH}(S)$ [3]. From now on, we suppose that $\mathcal{F}(S)$ has only one element, i.e., its element $\mathrm{COCH}(S)$ has no semi-isolated point.

Definition 4 ([3]) Let P be a finite planar point set. We define a point $u \in \mathrm{COCH}(P)$ to be *extreme* of $\mathrm{COCH}(P)$ if there exists an orthogonal line l (u is the vertex of l) whose intersection with $\mathrm{COCH}(P)$ is only u and there is no point of $\mathrm{COCH}(P) \setminus \{u\}$ which lies in the quadrant determined by l. We denote all extreme points of $\mathrm{COCH}(P)$ briefly by $\text{o-ext}(\mathrm{COCH}(P))$.

Remark 1 If $|P| > 2$, there exist two distinct extreme points of $\mathrm{COCH}(P)$. Indeed, we consider the following cases:

– If P has more than two distinct points and the points of P belong to a straight line, then the two ending points are the two distinct extreme points of COCH(P).
– If P has more than two distinct points and the points of P are not collinear, then by Proposition 3 ii), two distinct extreme points of COCH(P) are chosen as the highest leftmost and the lowest rightmost, or the leftmost highest and the rightmost lowest, or the rightmost highest and the leftmost lowest, or the highest rightmost and the lowest leftmost (see the definition of these points in Definition 9).

Lemma 1 ([3]) *We have* o-ext(COCH(P)) $\subset P$.

Let $u = (x_u, y_u), v = (x_v, y_v) \in S \subset \mathbb{R}^2$, L_1 norm is determined by $\|u - v\|_1 = |x_u - x_v| + |y_u - y_v|$. We use the definition L_1 norm in Proposition 1 and Lemma 2.

Proposition 1 ([3]) *Let $S \subset \mathbb{R}^2$. Then, S is connected orthogonal convex iff for all $a, b \in S$, there exists a shortest path $SP(a, b) \subset S$ joining a and b with L_1 norm, and the length of $SP(a, b)$ is $\|a - b\|_1$. In addition, $SP(a, b)$ is an increasingly monotone path (i.e., for $u, v \in SP(a, b), (x_u - x_v)(y_u - y_v) \geq 0$).*

The minimum rectilinear rectangle of a planar point set is a minimum rectangle having edges parallel to x or y axises that contains the set.

Lemma 2 *Every connected orthogonal convex hull of a finite planar point set is included in the minimum rectilinear rectangle of the point set.*

Proof. Let E be a connected orthogonal convex hull of finite planar point set P, R be the minimum rectilinear rectangle bounded of P. Assume, on contrary, $E \not\subseteq R$. Let $F = R \cap E \subsetneq E$. For all $a, b \in F$, $a, b \in E$. By orthogonal convexity of E, Proposition 1 yields that there exists a L_1 shortest path γ joining a and b such that $\gamma \subseteq E$. Since $a, b \in R$, γ is a monotone path in $R \cap E = F$ and therefore, F is connected. We are in position to prove that F is orthogonal convex. Let h be an arbitrary horizontal line intersecting F (the case of h being a vertical line is similar). Then $S = h \cap E$ convex. Since $h \cap F = h \cap (R \cap E) = (h \cap E) \cap R = S \cap R$, we conclude that $h \cap F$ is convex. Therefore, F is connected orthogonal convex and $F \subsetneq E$. This contradicts the fact that E is a smallest connected orthogonal convex hull of P. Hence, $E \subseteq R$. □

Note that Proposition 1 and Lemma 2 imply the following Proposition 2 and Proposition 2 is used in Subsection 5.2.

Proposition 2 *Let $P := \{p_1, \dots, p_m\} \subset \mathbb{R}^2$. Then, every connected orthogonal convex hull of P is compact.*

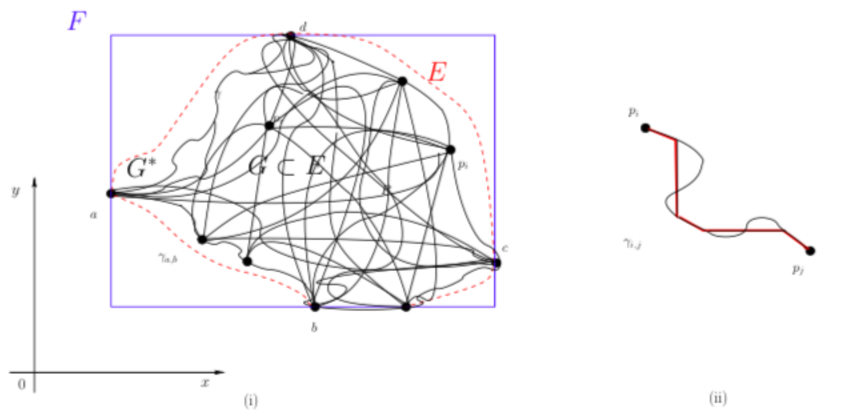

Fig. 1 (i) $G := \bigcup_{(i,j)\in[1,m]\times[1,m]} \gamma_{i,j}$ and the region G^* formed by the path γ. (ii) Adapting $\gamma_{i,j}$ to get an orthogonal convex set.

Proof. Let E be a connected orthogonal convex hull of P and F be the minimum rectangle having edges parallel to coordinates axises, where F is formed by $a, b, c, d \in P$. (see Fig. 1 (i)).

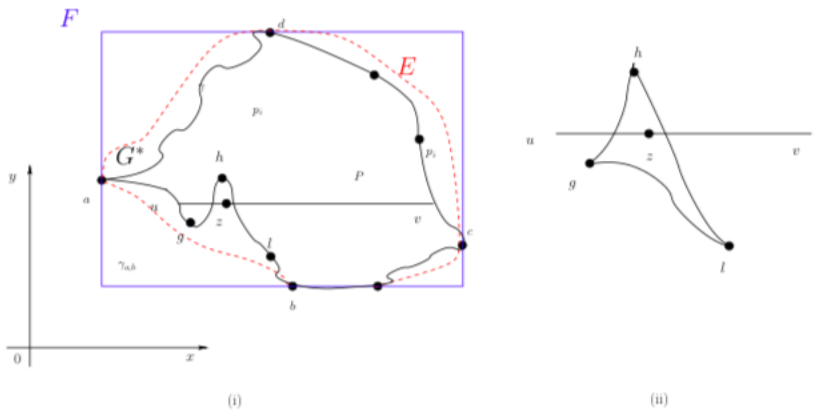

Fig. 2 $z \in [u, v]$ in the region G^* formed by the monotone paths $\gamma_{h,g}, \gamma_{h,l}$ and $\gamma_{g,l}$.

We now claim that there is a compact orthogonal convex subset set of E containing P. By Proposition 1, for each pair $(i, j) \in [1, m] \times [1, m]$, exists a staircase path belonging to E, say $\gamma_{i,j}$, joining p_i and p_j(see Fig. 1 (ii)).

Lemma 2 implies that $E \subset F$, then

$$G := \bigcup_{(i,j)\in[1,m]\times[1,m]} \gamma_{i,j} \subset F.$$

By the way, the closedness of $\gamma_{i,j}$ yields that G is closed. Thus G is compact. Let γ be the boundary of G. We prove that γ is a path. Let $\beta_1(x) := \min\{\gamma_{ij}(x) : (i,j) \in [1,m]\times[1,n]\}$, where $x \in [a_x, b_x]$. Since minimum of finite continuous functions is also continuous, $\beta_1(x)$ is continuous. Therefore, the part of γ from a to b is a path. By similar argument,

$$\beta_2(x) := \min\{\gamma_{ij}(x) : (i,j) \in [1,m] \times [1,n]\}, \text{where}\, x \in [b_x, c_x];$$
$$\beta_3(x) := \max\{\gamma_{ij}(x) : (i,j) \in [1,m] \times [1,n]\}, \text{where}\, x \in [b_x, c_x];$$
$$\beta_4(x) := \max\{\gamma_i(x) : (i,j) \in [1,m] \times [1,n]\}, \text{where}\, x \in [a_x, b_x]$$

are also continuous. Therefore, the parts of γ from b to c (β_2), from c to d (β_3) and from d to a (β_4) are paths. Then γ is a path. By the way chosen each part of γ, we get γ is not self-cross. Thus γ bounds a region G^*. We have $G^* \subset E$ and G^* is connected and contains P.

We are in position to prove that G^* is orthogonal convex. Take a rectilinear line k intersecting G^*. Let $u, v \in k \cap G^*$ being two "farthest" points which still lie in G^*. Assume without loss of generality that $[u,v]$ is parallel to x-axis. We claim that $[u,v] \subset G^*$. Assume the contrary that $z \in [u,v] \setminus G^*$ (see Fig. 2 (i)). Consider the case u belongs to the part $\gamma_{a,b}$ of γ between a and b and $a_x < z_x < b_x$ (the other cases are similar). As $\gamma_{a,b}$ is formed by some monotone paths joining two points of P, there are three points $g, h, l \in P$ such that h is above $[u,v]$, g, l are under $[u,v]$, $\gamma_{h,g}$ and $\gamma_{h,l}$ are monotone (see Fig. 2 (ii)). Since g, l are under $[u,v]$, the monotone path $\gamma_{g,l}$ is under $[u,v]$. Therefore z belongs to the region formed by $\gamma_{g,l}$, $\gamma_{h,g}$ and $\gamma_{h,l}$. This implies that $z \in G^*$, a contradiction. Thus, G^* is orthogonal convex.

Because E is the smallest connected orthogonal convex set containing P, we conclude that $G^* = E$. Thus E is compact. □

The compactness of the connected orthogonal convex hull of a finite planar point set P is also shown in the Lemma 3 below.

A *rectilinear polygon* is a simple polygon whose edges are rectilinear (i.e., they are parallel to either x or y axis). The polygon has therefore only 90 and 270 degree internal angles. An $(x,y)-$*polygon* is one of the following: a) a point; b) connected rectilinear line segments; c) a rectilinear polygon; and d) a connected union of type b) and or type c) (x,y)-polygons (see [11]) .

Lemma 3 ([3]) *The connected orthogonal convex hull of a finite planar point set P is an orthogonal convex (x,y)-polygon whose boundary is union of finite set of O-supports, and each O-support goes through two extreme points of* COCH(P).

3 Directed orthogonal lines and some related concepts

In this content we present definition a *directed orthogonal line* and some other properties necessary to serve the following section.

Given an ordered triple of points (a, b, c) in $\mathbb{R}^2$, let

$$\text{orient}(a, b, c) = \begin{vmatrix} 1 & x_a & y_a \\ 1 & x_b & y_b \\ 1 & x_c & y_c \end{vmatrix}. \tag{1}$$

Definition 5 ([6], p.10) We say that
(i) The ordered triple (a, b, c) has *positive orientation* (*negative orientation, zero orientation*, resp.) if $\text{orient}(a, b, c) > 0$ ($\text{orient}(a, b, c) < 0$, $\text{orient}(a, b, c) = 0$, resp.).
(ii) The point c is called *on the left of* (*on the right of*, *on*, resp.) the directed line ab if $\text{orient}(a, b, c) > 0$ ($\text{orient}(a, b, c) < 0$, $\text{orient}(a, b, c) = 0$, resp.).

Definition 6 Let $\mathrm{l}^v(a, b)$ be the orthogonal line through two points a and b with its vertex v. If b is on the right of av then we call $l^v(a, b)$ the *directed orthogonal line* from a to b and denoted it by $\mathcal{L}^v(a, b)$ (Figure 3).

Definition 7 Let $\mathcal{L}^v(a, b)$ be a directed orthogonal line from a to b with its vertex v. A point p is called *is on the right of* $\mathcal{L}^v(a, b)$ if p is on the right of both av and vb. A point p is called *is on the left of* $\mathcal{L}^v(a, b)$ if p is either on the left of av or on the left of vb.

We denote by P_{ab} the set containing all points of P being on the right of $\mathcal{L}^v(a, b)$.

Definition 8 Let $\mathcal{L}^v(a, b)$ be a directed orthogonal line from a to b with its vertex v and $p \in P_{ab}$. We call the length of $[p, v]$ the *orthogonal distance* from p to $\mathcal{L}^v(a, b)$, denoted by $\text{Odist}(p, \mathcal{L}^v(a, b))$. The point p is called the *farthest point* of P to $\mathcal{L}^v(a, b)$ if p satisfies

$$\text{Odist}(p, \mathcal{L}^v(a, b)) = \max_{q \in P_{ab}} \{\text{Odist}(q, \mathcal{L}^v(a, b))\}.$$

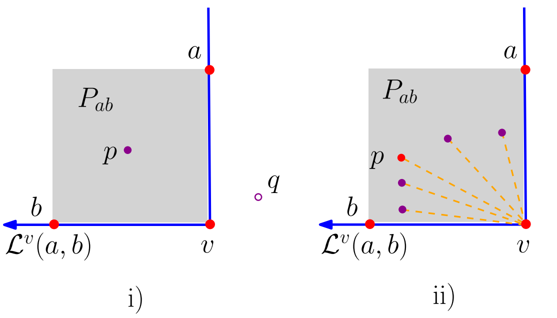

Fig. 3 i) p is on the right of $\mathcal{L}^v(a, b)$, q is on the left of $\mathcal{L}^v(a, b)$; ii) p is the farthest point to $\mathcal{L}^v(a, b)$.

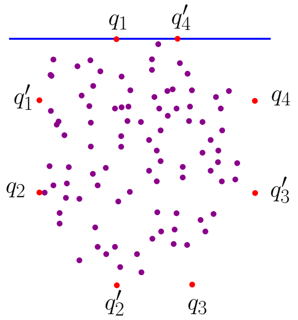

Fig. 4 Eight extreme points q_1, q'_1, q_2, q'_2, q_3, q'_3, q_4, q'_4.

Definition 9 ([8]) Let P be a finite planar point set. The point with the maximal y-coordinate (minimal y-coordinate, respectively) among the points of P having the minimal x-coordinate (maximal x-coordinate, respectively) is called the highest leftmost point (the lowest leftmost point, respectively). Similarly, we define seven other special points of P: leftmost highest, leftmost lowest, rightmost lowest, lowest rightmost, highest rightmost, rightmost highest points.

In Fig. 4, q_1 is the leftmost highest, q_1' is the highest leftmost, q_2 is the lowest leftmost, q_2' is the leftmost lowest, q_3 is the rightmost lowest, q_3' is the lowest rightmost, q_4 is the highest rightmost, and q_4' is the rightmost highest.

It is clear that the eight points in Definition 9 are the extreme points of COCH(P).

Given a point $p(x_p, y_p)$. The four orthants $o_1(p), o_2(p), o_3(p)$ and $o_4(p)$ are determined by the closed regions

$$\begin{aligned}
o_1(p) &:= [x_p, +\infty) \times [y_p, +\infty),\\
o_2(p) &:= (-\infty, x_p] \times [y_p, +\infty),\\
o_3(p) &:= (-\infty, x_p] \times (-\infty, y_p],\\
o_4(p) &:= [x_p, +\infty) \times (-\infty, y_p]
\end{aligned}$$

as the orthants of the point p (see Fig. 5).

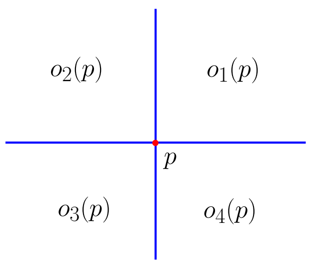

Fig. 5 $o_1(p), o_2(p), o_3(p)$ and $o_4(p)$ are the orthants of the point p.

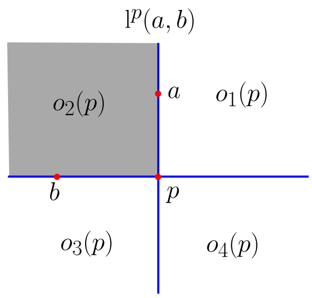

Fig. 6 $o_2(p)$ (shaded area) coincides with the quadrant defined by orthogonal line $\mathrm{l}^p(a,b)$.

Definition 10 (see [5]) A point $p \in P$ is called *maximal point* if at least one its orthant does not contain any points of $P \setminus \{p\}$.

Remark 2 It is easy to see that four orthants of a point coincide with the quadrants defined by four orthogonal lines of the same point.

Lemma 4 *If* $p \in$ COCH(P) *then each orthant of* p *contains at least an extreme point of* COCH*(P).*

Proof. Taking a point $p \in \text{COCH}(P)$, we consider three cases: p is an extreme point of $\text{COCH}(P)$, p is on the boundary but is not an extreme point of $\text{COCH}(P)$ (see Fig. 7) and p is in the interior of $\text{COCH}(P)$ (see Fig. 8).

Case 1: p is an extreme point of $\text{COCH}(P)$. Then clearly all four orthants of p contain the extreme point p of $\text{COCH}(P)$.

Case 2: p is on the boundary but is not an extreme point of $\text{COCH}(P)$. According to Lemma 3, p must belong to an $\mathcal{O}$-support $\mathrm{l}(a,b)$ that goes through two extreme points a and b of $\text{COCH}(P)$. If $\mathcal{O}$-support $\mathrm{l}(a,b)$ is a straight line (i.e, $x_a = x_b$ or $y_a = y_b$), then clearly two orthants of p contain the extreme point a and the remaining two orthants contain the extreme point b (see Figure 7 i)). As two some orthants of p contain the extreme point a, without loss of generality assume that p belongs to the rectilinear half line containing b (see 7 ii)). Then two orthants of p contain the extreme point b. One of the remaining two orthants contains the extreme point a. We will show that the final orthants of p (e.g., $o_4(p)$ as shown in Fig. 7 ii)) contain at least one other extreme point of $\text{COCH}(P)$. Indeed, suppose that $o_4(p)$ does not contain any extreme point of $\text{COCH}(P)$. Then, by Lemma 3, $o_4(p)$ does not contain any point of P. Let c be a smallest y-coordinate point among the points of P in $o_2(p)$, and d be a greatest x-coordinate point among the points of P in $o_3(p)$ (because P is finite, such c and d exist). Then an orthogonal line $\mathrm{l}(c,d)$ is an O-support and the intersection of $\mathrm{l(c,d)}$ and $\mathrm{l(a,b)}$ consists of two distinct points. Therefore, connected orthogonal convex hulls of P have semi-isolated points. This is contrary to assumption (A).

Case 3: p is in the interior of $\text{COCH}(P)$. Then the orthants $o_1(p)$, $o_2(p)$, $o_3(p)$, and $o_4(p)$ intersect $\mathrm{COCH(P)} \setminus \{\mathrm{p}\}$. Applying the similar argument for $o_4(p)$ in Case 2 to these orthants, we conclude that each orthant contains at least one extreme point of $\text{COCH}(P)$. □

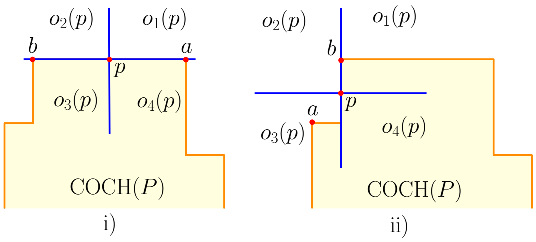

Fig. 7 p lies on the boundary of COCH(P).

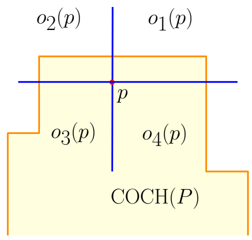

Fig. 8 p is inside the boundary point of COCH(P).

We recall the following result introduced in [5].

Lemma 5 (see [5]) *Let P be the set of n points chosen according to any probability distribution Δ. Then the expected number of maximal is $O(\log n)$.*

We denote the set of maximal points of P by $\mathcal{M}(P)$.

Lemma 6 *Let P be a finite planar point set satisfying* (A)*. Then*

$$\mathcal{M}(P) = \text{o-ext}(\text{COCH(P)}).$$

Proof. Let $p \in \mathcal{M}(P)$, according to Definition 10, at least one of its orthants does not contain any points of $P \setminus \{p\}$. Without loss of generality we assume that $o_2(p)$ does not contain any points of $P \setminus \{p\}$. We will prove $p \in$ o-ext(COCH(P)), that is, according to Definition 4 and Remark 2, we will prove that the intersection of $o_2(p)$ and COCH(P) is only u and there is no point of COCH(P) $\setminus \{p\}$ which lies in $o_2(p)$. We will prove this by contradiction. Indeed, suppose that there exists $u \in o_2(p) \cap$COCH(P) and $u \neq p$. There are two following cases:

- u is an extreme point of COCH(P). According to Lemma 1, $u \in P$, namely $o_2(p)$ contains a points $u \in P \setminus \{p\}$.
- u is not an extreme point of COCH(P). According to Lemma 4, each orthant of u contains at least one extreme point of COCH(P). Therefore, $o_2(u)$ also contains at least one extreme point, say t, of COCH(P) and according to Lemma 1, $t \in P$. Furthermore, since $u \in o_2(p)$, we have $o_2(u) \subset o_2(p)$. It follows that $t \in o_2(p)$.

Both cases above contradict the hypothesis that $o_2(p)$ does not contain any points of $P \setminus \{p\}$. Hence, $p \in$ o-ext(COCH(P)). Therefore

$$\mathcal{M}(P) \subset \text{o-ext}(\text{COCH(P)}). \tag{2}$$

Conversely, let $p \in$ o-ext(COCH(P)), we are in position to prove that $p \in \mathcal{M}(P)$. Indeed, Lemma 1 implies $p \in P$. According to Definition 4 and remark 2, there exists an orthant $o(p)$ of p that does not contain any point of COCH(P). Therefore $o(p)$ also does not contain any points of P. It follows that the point $p \in \mathcal{M}(P)$. Therefore

$$\text{o-ext}(\text{COCH(P)}) \subset \mathcal{M}(\text{P}). \tag{3}$$

It follows from (2) and (3) that $\mathcal{M}(P) =$ o-ext(COCH(P)). □

Let P be a finite planar point set. The following proposition is needed to prove the correctness of the algorithm in the next section.

Proposition 3 *Let p, q $(x_p \neq x_q, y_p \neq y_q)$ be any two distinct extreme points of* COCH(P) *and r be a farthest point of P_{pq} from the directed orthogonal line $\mathcal{L}^v(p, q)$. Then*

i) *r is an extreme point of* COCH(P)*.*
ii) *For each $w \in o(r)_v \cap P_{pq}$,* $\text{dist}(\text{w}, \text{vq}) \leq \text{dist}(\text{p}, \text{v})$ *and* $\text{dist}(\text{w}, \text{vp}) \leq \text{dist}(\text{q}, \text{v})$*, we have $o(r)_v \cap o(w)_v \cap P_{pq} \cap$ o-ext(COCH(P)) $= \emptyset$, where $o(w)_v$ is an orthant of w and contains v. Consequently, w is not an extreme point of* COCH(P)*.*

Proof. Consider the case $x_p > x_q$ and $y_p > y_q$ (the other cases are similar). i) We claim that

$$o_2(r) \text{ does not contain any point of } P_{pq} \setminus \{r\}. \tag{4}$$

Assume the contrary that there is a point $t \in o_2(r) \cap (P_{pq} \setminus \{r\})$. Because $\angle(tr, vr) \geq \pi$, we get $\mathrm{dist(v,t)} > \mathrm{dist(v,r)}$ and therefore r is not a farthest point of P_{pq}, a contradiction. Thus (4) holds true and therefore r is an extreme point of COCH(P).

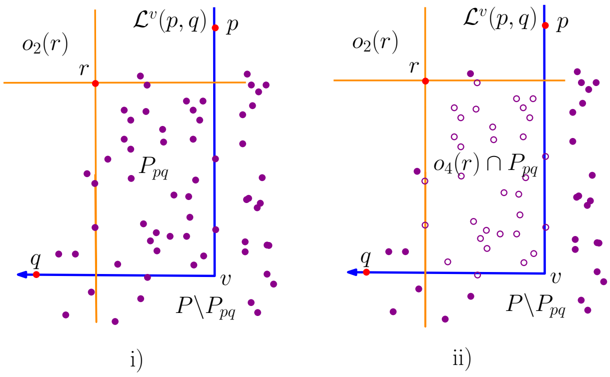

Fig. 9 i) The farthest point r to $\mathcal{L}^v(p,q)$ is an extreme point of COCH(P); ii) All the points in the set $o_4(r) \cap P_{pq} \setminus \{r\}$ are not the extreme points of COCH(P).

ii) Take $w1 \in o(r)_v \cap o(w)_v \cap P_{pq}$. Then, $r \in o_2(w1)$, $p \in o_1(w1)$ (as $\mathrm{dist(w1,vq)} \leq \mathrm{dist(w,vq)} \leq \mathrm{dist(p,v)}$), $q \in o_3(w1)$ (as $\mathrm{dist(w1,vp)} \leq \mathrm{dist(w,vp)} \leq \mathrm{dist(q,v)}$), and $v \in o_4(w)$. By the definition of extreme points of COCH(P), $w1$ is not an extreme point of COCH(P). Choose $w1 = w$, we conclude that w is not an extreme point of COCH(P).

□

Remark 3 Among the points of the set P_{pq}, there can be more than one farthest point from the directed orthogonal line $\mathcal{L}^v(p,q)$. However, all of them are the extremes of COCH(P) (according to Proposition 3).

4 Algorithm based on Quickhull for finding the connected orthogonal convex hull

4.1 $\mathcal{O}$-Quickhull algorithm

Consider the following four cases: the leftmost highest concides with the highest leftmost, and the lowest leftmost concides with the leftmost lowest, and the rightmost lowest concides with the lowest rightmost, and the highest rightmost concides with the rightmost highest, COCH(P) is a rectangle formed by these points. That is reason why we can assume from now on that at least one of the cases above does not hold.

Inspired by the idea of the Quickhull algorithm [4,6,7,13], we now present a new efficient algorithm, namely $\mathcal{O}$-QUICKHULL, for finding the connected orthogonal convex hull COCH(P) of a finite planar point set P under the assumption (A). The first step of the $\mathcal{O}$-QUICKHULL algorithm is to find two distinct extreme points, say a and b, of COCH(P) (this is always guaranteed according to Remark 1). Let $\mathcal{L}^s(a,b)$ be the directed orthogonal line with its vertex s from a to b. Note that, P_{ab} the set containing all points on the right of $\mathcal{L}^s(a,b)$. Then, from P_{ab} find the farthest point, say c, from the directed orthogonal line $\mathcal{L}^s(a,b)$. Add the point c to o-ext(COCH(P)). Let $\mathcal{L}^{s_1}(a,c)$ ($\mathcal{L}^{s_2}(c,b)$, resp.) be the directed orthogonal line with its vertex s_1 (s_1, resp.) from a to c (from c to b). To find the next extreme points of COCH(P), we replace the directed orthogonal line $\mathcal{L}^s(a,b)$ by $\mathcal{L}^{s_1}(a,c)$ and $\mathcal{L}^{s_2}(c,b)$, and recursively continue the algorithm.

$\mathcal{O}$-QUICKHULL algorithm illustrates the function $\mathcal{O}$-Quickhull(a,b,P_{ab}), where a,b is two distinct extreme points of COCH(P) and P_{ab} is the set of all points on the right of the directed orthogonal line $\mathcal{L}^s(a,b)$ with its vertex s from a to b. The output of $\mathcal{O}$-QUICKHULL algorithm is the subset of P_{ab} containing all the extreme points of COCH(P). We use "$\cup$" to represent list concatenation. The final orthogonal convex hull is found when we choose pairs of extreme points to apply $\mathcal{O}$-QUICKHULL algorithm (see Subsection 5.2).

Proposition 3 ii) allows us not to consider points $w \in o(r)_v \cap P_{pq}$, $\mathrm{dist(w,vq)} \leq \mathrm{dist(p,v)}$ and $\mathrm{dist(w,vp)} \leq \mathrm{dist(q,v)}$, here p,q,r play the roles of a,b,c, respectively. To do so at every step, we choose initial distinct points a and b, respectively such that $\mathrm{dist(c,sb)} \leq \mathrm{dist(a,s)}$ and $\mathrm{dist(c,sa)} \leq \mathrm{dist(b,s)}$.

Algorithm 1 $\mathcal{O}$-QUICKHULL algorithm

function $\mathcal{O}$-Quickhull(a,b,P_{ab})

1. **If** $P_{ab} = \emptyset$ then **return** ()
2. **else**
   (a) $c \leftarrow$ the farthest point from $\mathcal{L}^s(a,b)$.
   (b) $P_{ac} \leftarrow$ the set of points on the right of the directed orthogonal line $\mathcal{L}^{s_1}(a,c)$ from a to c with its vertex s_1.
   (c) $P_{cb} \leftarrow$ the set of points on the right of the directed orthogonal line $\mathcal{L}^{s_2}(c,b)$ from c to b with its vertex s_2.
   (d) **return** $\mathcal{O}$-Quickhull(a,c,P_{ac}) $\cup \{c\} \cup$ $\mathcal{O}$-Quickhull(c,b,P_{cb}).

4.2 The correctness and the complexity of $\mathcal{O}$-QUICKHULL algorithm

Next we will prove the correctness of $\mathcal{O}$-QUICKHULL algorithm in the following Theorem 1.

Theorem 1 *The output of $\mathcal{O}$-*Quickhull *algorithm is the subset of P_{ab} containing all the extreme points of* COCH*(P).*

Proof. Because the function $\mathcal{O}$-Quickhull(a, b, P_{ab}) chooses the farthest point c from $\mathcal{L}^s(a, b)$, c is an extreme point of COCH(P) (Proposition 3 i)). Thus, $\mathcal{O}$-Quickhull detects points in $P_{ac} \cup P_{cb}$ and the output consists of extreme points of COCH(P). Because initial distinct points a and b are such that $\mathrm{dist(c, sb)} \leq \mathrm{dist(a, s)}$ and $\mathrm{dist(c, sa)} \leq \mathrm{dist(b, s)}$, we get that $o(c)_s \cap P_{ab}$ does not contain any extreme points of COCH(P) (Proposition 3 ii)). It follows that the output is the set of all extreme points of COCH(P) in P_{ab}.

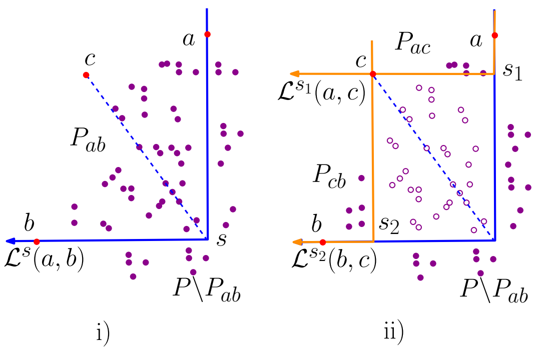

Fig. 10 i) The set P_{ab} with the farthest points c from $\mathcal{L}^s(a, b)$; ii) The set P_{ac}.

□

We will discuss the following simple analysis of the time complexity of $\mathcal{O}$-Quickhull algorithm.

Remark 4 In $\mathcal{O}$-Quickhull algorithm, after finding the extreme point c, in order to find the next extreme points, we only need to consider the points in the set P_{ac} and P_{cb}, that is, the points in the set $P_{ab} \setminus (P_{ac} \cup P_{cb})$ are not considered anymore because these points cannot be the extreme points of the COCH(P) (see Fig. 10 ii) and Proposition 3 ii)). Thus, the number of points to to detect the next extreme point will decrease significantly.

Theorem 2 *Suppose that the set P_{ab} consists n points. The worst case complexity of the $\mathcal{O}$-*Quickhull *algorithm is $O(n^2)$ and its expected complexity is $O(n \log n)$.*

Proof. Suppose that the output of $\mathcal{O}$-Quickhull algorithm has m extreme points of COCH(P). The algorithm calls $\mathcal{O}$-Quickhull functions $(m+1)$ times. In which, each of the first m functions finds exactly one extreme point of COCH(P) and need $O(n)$ time complexity. The last time work with an empty set. So the time complexity of $\mathcal{O}$-Quickhull algorithm is $O(mn)$. In the worst case, when $m = n$ (here n is the number of points of P_{ab}), we have the worst time complexity of $O(n^2)$. According to Lemma 5 and Lemma 6, the expected number of extreme points of COCH(P) in P_{ab} is $O(\log n)$, i.e., $m = O(\log n)$. Therefore, the expected complexity of $\mathcal{O}$-QUICKHULL algorithm is $O(n \log n)$. $\square$

5 Implementation

In this section, we are going to compare the running times of our algorithm to $\mathcal{O}$-Graham introduced by An, Huyen and Le in [3] and an other algorithm proposed by Montuno and Fournier in [11].

5.1 The Test Sets

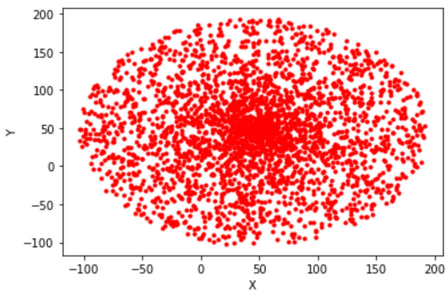

Fig. 11 Disc data with 3000 points.

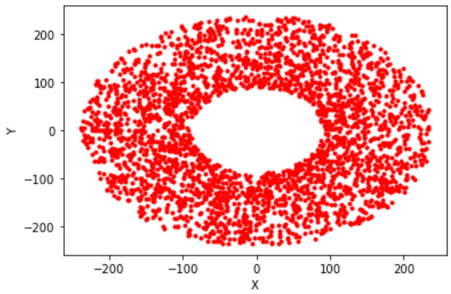

Fig. 12 Hollow disc data with 3000 points.

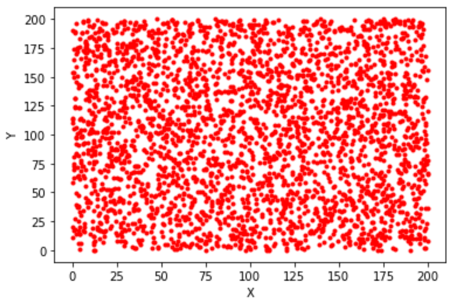

Fig. 13 Square data with 3000 points.

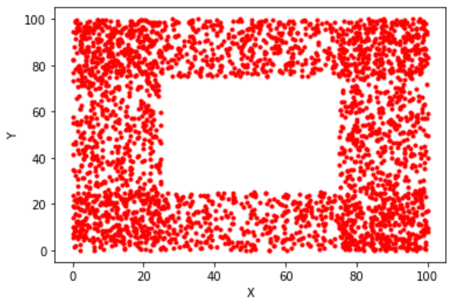

Fig. 14 Hollow square data with 3000 points.

To test the algorithms we create five data types, below are specific descriptions for these 6 data types

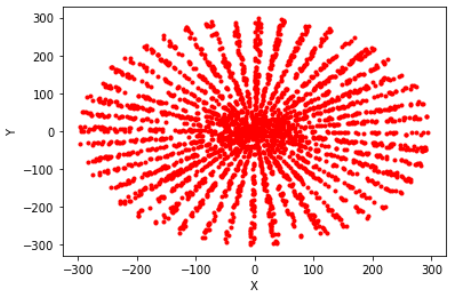

Fig. 15 Sun data with 3000 points.

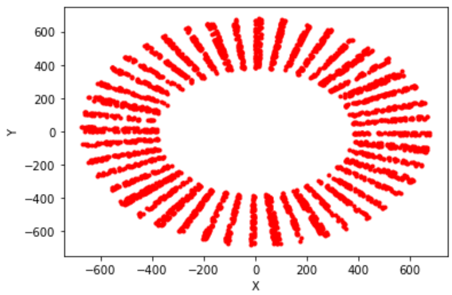

Fig. 16 Hollow Sun data with 3000 points.

- Disc data: We generate random real points in a disc. For instance, the input data in Fig. 11 consists of 3000 points.
- Hollow disc data: We create two concentric discs (or ellipses) of different radii. The input points are created randomly outside the smaller disc (or ellipse) but inside the bigger one. Different examples are created with different radii ratios (see Fig. 12).
- Square data: The input points are randomly generated inside a square (see Fig. 13).
- Hollow square data: The points are created randomly inside a square and outside another smaller concentric square. We create data corresponding to different sizes of the smaller square (see Fig. 14).
- Sun data: The points are randomly generated according to the central angle of a disc and interspersed equal angles with no points (see Fig. 15).
- Hollow sun data: The points are randomly generated according to the central angle of a disc and interspersed equal angles with no points. In addition, these points are also created outside the concentric disc with the one above (see Fig. 16).

5.2 Numerical Results

In this subsection we present the selection of pairs of extreme distinct points a and b such that $\mathrm{dist(c, sb)} \leq \mathrm{dist(a, s)}$ and $\mathrm{dist(c, sa)} \leq \mathrm{dist(b, s)}$ hold true to apply $\mathcal{O}$-QUICKHULL algorithm to find the final orthogonal convex hull of a finite set of a planar points.

It is known that the points lying inside or on the edges of the polygon which its edges are parallel to x-axis or y-axis connecting eight extreme points q_1, q_1', q_2, q_2', q_3, q_3', q_4 and q_4' (except the eight these points) are not extreme points of COCH(P) and can be deleted (see Figure 17).

Due to the compactness of COCH(P), we can determine its boundary according to Lemma 3 as the union of finite set of O-supports, where each O-support goes through two extreme points of COCH(P). Assume that $q_1 \neq q_1'$ and $p_0 = q_1, p_1, \ldots, p_{k-1}, p_k = q_1'$ are such extreme points of COCH(P) on the the right of the directed orthogonal line from q_1 to q_1'. They form a staircase

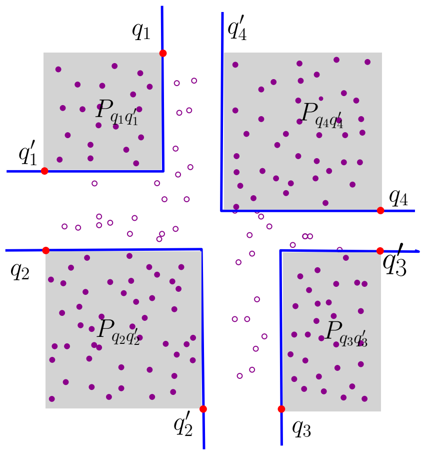

Fig. 17 Four sets $P_{q_iq'_i}$, $i = 1, 2, 3, 4$.

path, say, $\mathcal{P}_{q_1q'_1}$ (i.e., a union of parts of O-supports through p_i and p_{i+1}, $i = 0, \ldots, k-1$) joining q_1 and q'_1. Similarly, we can also determine $\mathcal{P}_{q_iq'_i}$, $i = 2, 3, 4$. Thus COCH(P) is an orthogonal convex (x, y)-polygon whose boundary is union of the rectilinear line segments $[q_i, q'_i]$, $i = 1, 2, 3, 4$ and staircase paths $\mathcal{P}_{q_iq'_i}$, joining q_i and q'_i,$i = 1, 2, 3, 4$, respectively.

To find staircase paths $\mathcal{P}_{q_iq'_i}$, $i = 1, 2, 3, 4$, we apply $\mathcal{O}$-Quickhull algorithm for set $P_{q_1q'_1}$ if $q_1 \neq q'_1$, set $P_{q_2q'_2}$ if $q_2 \neq q'_2$, set $P_{q_3q'_3}$ if $q_3 \neq q'_3$, set $P_{q_4q'_4}$ if $q_4 \neq q'_4$. Therefore, the final orthogonal convex hull COCH(P) is

$$\{q_1\} \cup \mathcal{O}\text{-Quickhull}(q_1, q'_1, P_{q_1q'_1}) \cup \{q'_1, q_2\} \cup \mathcal{O}\text{-Quickhull}(q_2, q'_2, P_{q_2q'_2})$$

$$\cup\{q'_2, q_3\} \cup \mathcal{O}\text{-Quickhull}(q_3, q'_3, P_{q_3q'_3}) \cup \{q'_3, q_4\} \cup \mathcal{O}\text{-Quickhull}(q_4, q'_4, P_{q_4q'_4}).$$

If no case occurs (i.e., $q_1 = q'_1$, $q_2 = q'_2$, $q_3 = q'_3$, $q_4 = q'_4$), the rectangle $q_1q_2q_3q_4$ is the orthogonal convex hull to look for.

The algorithms are implemented in python and run on PC 1.8 GHz Intel Core i5 with 8 GB RAM. The Fig. 18- 23 illustrate the results of finding orthogonal convex hull of the sets of points corresponding to the data sets.

Tables 1 - 6 list the running times (in seconds) of the three algorithms: $\mathcal{O}$-Quickhull algorithm, $\mathcal{O}$-Graham introduced by An, Huyen and Le in [3] ($\mathcal{O}$-Graham, in short) and the algorithm proposed by Montuno and Fournier in [11] (Montuno and Fournier 's algorithm, in short).

In general, the $\mathcal{O}$-Quickhull algorithm runs faster than the others. For convenience, in Table 7, we list the ratios of the actual running times of $\mathcal{O}$-graham algorithm and Montuno and Fournier's algorithm to the $\mathcal{O}$-Quickhull algorithm. In the last row of Table 7, we compute the ratios of all the tested data for the overall results. All ratios are calculated using the geometric mean. The average result on all the data, our algorithm is 1.431 times faster than $\mathcal{O}$-Graham algorithm and 1.401 times faster than Montuno and Fournier's algorithm. To explain this, after finding an extreme point of COCH(P), a large

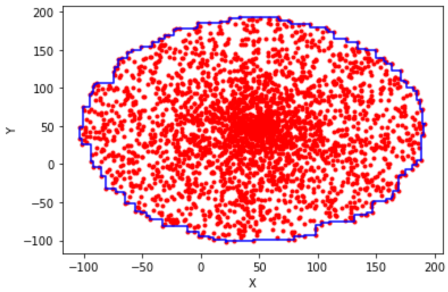

Fig. 18 Disc data with 3000 points.

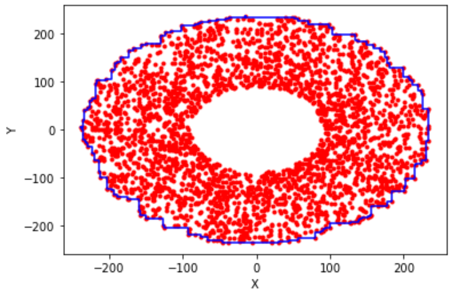

Fig. 19 Hollow disc data with 3000 points.

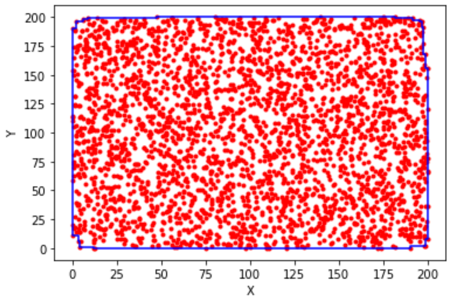

Fig. 20 Square data with 3000 points.

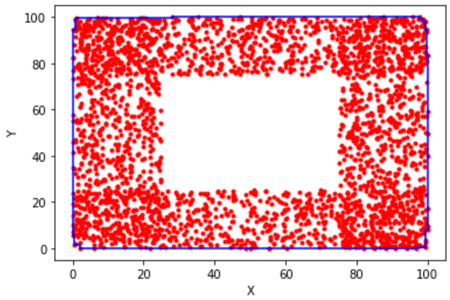

Fig. 21 Hollow square data with 3000 points.

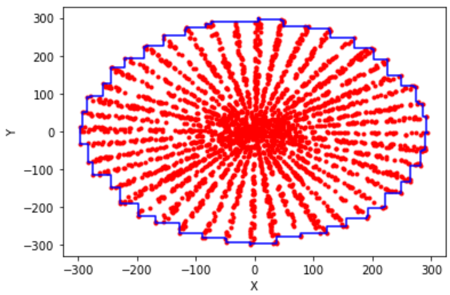

Fig. 22 Under triangle with 3000 points.

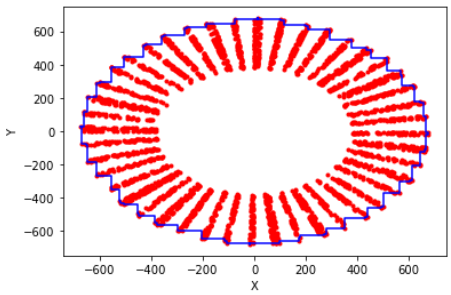

Fig. 23 Hollow Sun data with 3000 points.

number of points that are certainly not the extreme points of COCH(P) is ignored, i.e, the number of points to consider when finding a new extreme point will reduce significantly (see Remark 4). Furthermore, our algorithm does not need a preprocessing step, which rearranges the input points as $\mathcal{O}$-graham algorithm and Montuno and Fournier's algorithm.

Input	$\mathcal{O}$-Graham	Montuno and Fournier 's algorithm	$\mathcal{O}$-Quickhull
1000	0.18	0.332	0.138
3000	0.259	0.473	0.187
5000	0.362	0.625	0.245
7000	0.685	0.777	0.344
10000	0.999	1.115	0.594
13000	1.333	1.379	0.993
15000	1.724	1.868	1.131
17000	2.096	1.964	1.418
20000	2.746	2.363	2.012
25000	4.420	3.940	2.861
30000	5.868	5.140	4.331
35000	8.635	7.503	6.321
40000	11.532	9.143	7.558
45000	15.217	12.099	9.939
50000	18.504	15.844	12.590
70000	42.312	34.126	29.271
100000	98.457	71.852	66.852

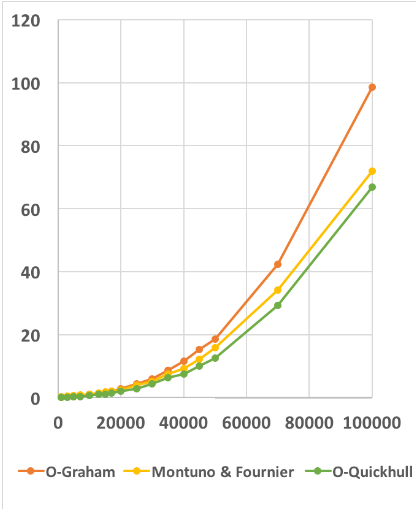

Table 1 The actual running times (in seconds) of the algorithms for disc data.

Input	$\mathcal{O}$-Graham	Montuno and Fournier 's algorithm	$\mathcal{O}$-Quickhull
1000	0.144	0.280	0.127
3000	0.201	0.446	0.189
5000	0.429	0.798	0.258
7000	0.497	0.873	0.351
10000	0.914	1.224	0.635
13000	1.298	1.444	0.929
15000	1.839	1.763	1.145
17000	2.132	2.652	1.597
20000	2.923	3.687	2.139
25000	4.471	4.059	3.280
30000	6.073	5.166	4.820
35000	9.167	6.918	6.411
40000	11.373	9.940	8.744
45000	14.945	12.750	10.934
50000	19.502	15.873	14.236
70000	42.137	38.068	30.196
100000	91.748	79.085	73.189

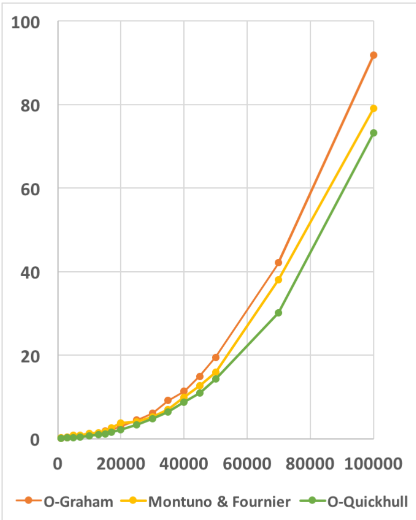

Table 2 The actual running times (in seconds) of the algorithms for hollow disc data.

6 Concluding Remarks

We have provided an efficient algorithm, based on the idea of quickhull, for finding the connected orthgonal convex hull of a finite planar point set and have compared it with the algorithm [3] and Montuno and Fournier's algorithm [11]. A similar algortihm for finding $\mathcal{O}_\beta$-convex hulls (introduced in [2])

Input	$\mathcal{O}$-Graham	Montuno and Fournier 's algorithm	$\mathcal{O}$-Quickhull
1000	0.278	0.230	0.147
3000	0.247	0.288	0.188
5000	0.406	0.375	0.253
7000	0.493	0.451	0.318
10000	0.896	0.646	0.557
13000	1.195	1.036	0.748
15000	1.448	1.379	1.180
17000	1.932	1.742	1.335
20000	2.283	2.079	1.734
25000	3.556	3.264	2.738
30000	5.452	4.841	4.063
35000	7.867	6.991	5.868
40000	10.376	9.561	7.698
45000	12.003	10.478	10.033
50000	17.462	16.018	13.466
70000	39.573	38.954	30.962
100000	81.678	78.239	70.768

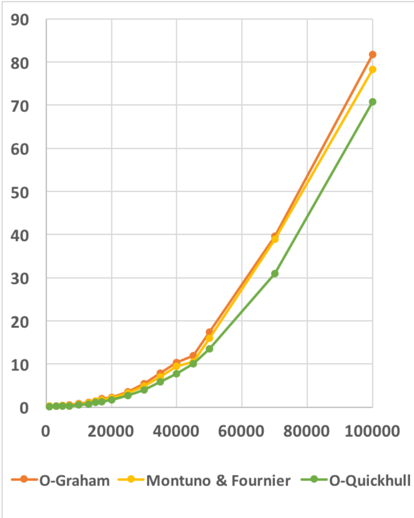

Table 3 The actual running times (in seconds) of the algorithms for square data.

Input	$\mathcal{O}$-Graham	Montuno and Fournier 's algorithm	$\mathcal{O}$-Quickhull
1000	0.196	0.247	0.133
3000	0.211	0.304	0.226
5000	0.331	0.371	0.237
7000	0.405	0.451	0.295
10000	0.791	0.687	0.528
13000	1.117	1.034	0.803
15000	1.411	1.212	1.027
17000	1.626	1.525	1.176
20000	2.234	2.112	1.769
25000	3.399	3.318	2.641
30000	4.987	4.991	3.978
35000	6.363	6.544	5.630
40000	8.958	9.591	7.812
45000	10.776	11.496	9.924
50000	15.834	16.945	12.901
70000	32.323	33.717	31.130
100000	75.321	76.787	70.491

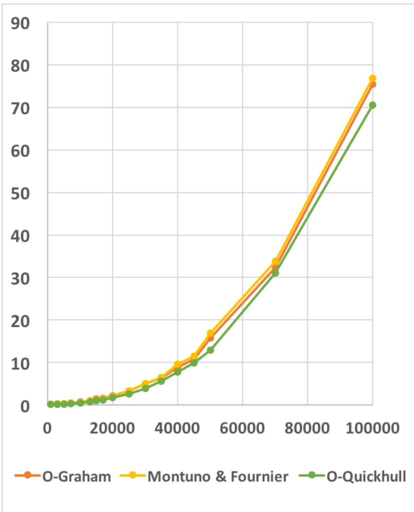

Table 4 The actual running times (in seconds) of the algorithms for hollow square data.

can be given. In addition, we can use the idea of space subdivision [19] to give efficient algorithms for finding such hulls. They will be the subject of another paper.

Input	$\mathcal{O}$-Graham	Montuno and Fournier 's algorithm	$\mathcal{O}$-Quickhull
1000	0.174	0.280	0.144
3000	0.250	0.442	0.181
5000	0.348	0.554	0.239
7000	0.588	0.647	0.359
10000	0.841	0.888	0.512
13000	1.433	1.250	0.849
15000	1.867	1.555	1.093
17000	2.334	2.054	1.300
20000	2.966	2.219	1.861
25000	4.801	3.477	2.961
30000	6.661	4.958	4.360
35000	10.259	6.934	5.857
40000	13.051	10.096	7.812
45000	18.031	14.050	11.042
50000	20.918	16.801	13.147
70000	47.831	38.197	31.588
100000	108.946	87.346	70.990

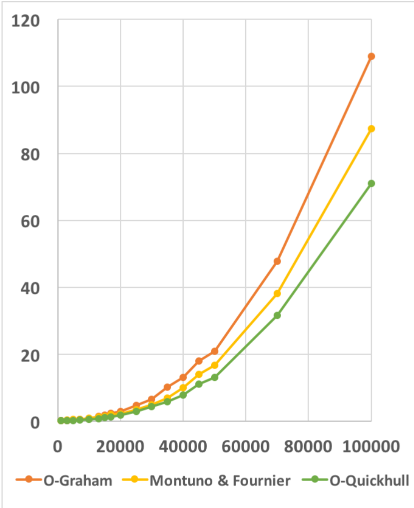

Table 5 The actual running times (in seconds) of the algorithms for sun data.

Input	$\mathcal{O}$-Graham	Montuno and Fournier 's algorithm	$\mathcal{O}$-Quickhull
1000	0.156	0.347	0.137
3000	0.251	0.339	0.177
5000	0.311	0.451	0.240
7000	0.442	0.681	0.327
10000	0.891	0.867	0.653
13000	1.158	1.134	0.869
15000	1.811	1.412	1.112
17000	2.026	1.725	1.407
20000	2.834	2.212	2.016
25000	4.099	5.118	3.567
30000	5.887	5.791	4.334
35000	8.163	6.440	5.241
40000	11.558	7.591	7.032
45000	14.476	11.996	10.496
50000	17.234	14.745	13.709
70000	40.323	31.717	29.320
100000	89.321	76.787	73.227

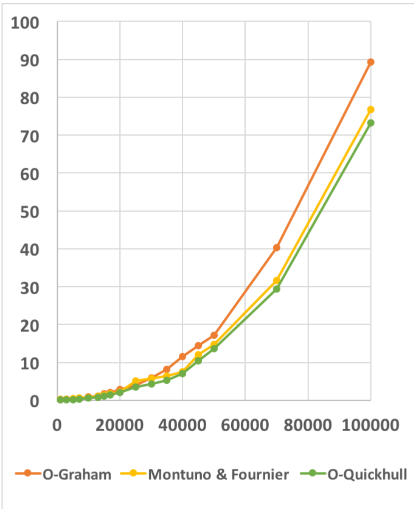

Table 6 The actual running times (in seconds) of the algorithms for hollow sun data.

Acknowledgment

The second and the third authors would like to thank the Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for the support of time and facilities for this study.

Table 7 The ratios of the actual running times of $\mathcal{O}$-graham algorithm and Montuno and Fournier's algorithm to the $\mathcal{O}$-QUICKHULL algorithm.

Data types	The ratio of $\mathcal{O}$-Graham to $\mathcal{O}$-QUICKHULL	The ratio of Montuno and Fournier's algorithm to $\mathcal{O}$-QUICKHULL
Disc data	1.479	1.516
Hollow disc data	1.355	1.539
Square data	1.394	1.264
Hollow square data	1.244	1.288
Sun data	1.579	1.467
Hollow sun data	1.364	1.365
All data	1.431	1.401

References

1. Abello, J., Estivill-Castro, V., Shermer, T., Urrutia, J., Illumination of orthogonal polygons with orthogonal floodlights, *International Journal of Computational Geometry & Applications*, **8**(1), 25–38, 1998.
2. Alegri'a-Galiciaa, C., Orden, D., Searac, C., Urrutia, J.: On the O-hull of a planar point set. Computational Geometry, **68**, pp. 277–291 (2018)
3. An P.T., Huyen P.T.T, and Le N.T., A modified Grahams scan algorithm for finding the smallest connected orthogonal convex hull of a finite planar point set, *Applied Mathematics and Computation*, **397**, 125889, 2021.
4. Bykat A., Convex hull of a finite set of points in two dimensions, Information Processing Letters, **7**, 296–298, 1978.
5. Damerow V., and Sohler C., Extreme points under random noise, *European Symposium on Algorithms*, **3221**, 264–274, 2004.
6. David M. M., *Computational geometry*, Department of Computer Science, University of Maryland, 2020.
7. Eddy W., A new convex hull algorithm for planar sets, *ACM Transactions on Mathematical Software*, **4**, 398 – 403, 1977.
8. Hoang N.D. and Linh N.K., Quicker than quickhull, *Vietnam Journal of Mathematics*, **43**, 57–70, 2015.
9. Biedl T. and Genc B., Reconstructing orthogonal polyhedra from putative vertex sets, *Computational Geometry*, **44**(8), 409–417, 2011.
10. Karlsson R. G., Overmars M. H., Scan line algorithms on a grid, *BIT Numerical Mathematics*, **28**(2), 227–241, 1988.
11. Montuno D. Y., Fournier A., Finding the x-y convex hull of a set of x-y polygons. Technique Report University of Toronto, **148**, 1982.
12. Nicholl T. M., Lee D. T., Liao Y. Z., Wong C. K., On the X-Y convex hull of a set of X-Y polygons, *BIT Numerical Mathematics*, **23**(4), 456–471, 1983.
13. O'Rourke, J., *Computational geometry in C*, Cambridge University Press, 1998.
14. Ottmann T., Soisalon-Soininen E., Wood D., On the definition and computation of rectilinear convex hulls, *Information Sciences*, **33**(3), 157–171, 1984.
15. Seo J., Chae S., Shim J., Kim D., Cheong C., Han T., Fast contour-tracing algorithm based on a pixel-following method for image sensors, *Sensors*, **16**(3), 353, 2016.
16. Son W., Hwang S-W., Ahn H-K., MSSQ: Manhattan spatial skyline queries, *Information Systems*, **40**, 67–83, 2014.
17. Uchoa E., Aragao M. P. de, Ribeiro C. C., Preprocessing Steiner problems from VLSI layout, *Networks*, **40**(1), 38–50, 2002.
18. Unger S. H., Pattern detection and recognition, *Proceedings of the IRE, IEEE*, 1737–1752, 1959.
19. Skala V., Diameter and convex hull of points using space subdivision in $\mathbb{E}^2$ and $\mathbb{E}^3$, In: Gervasi O. et al. (eds) Computational Science and Its Applications ICCSA 2020. ICCSA 2020. *Lecture Notes in Computer Science*, **12249**. Springer, Cham.