

# Maintaining Extremal Points and Its Applications to Deciding Optimal Orientations

Sang Won Bae<sup>1</sup>, Chunseok Lee<sup>1</sup>, Hee-Kap Ahn<sup>2</sup>, Sunghee Choi<sup>1</sup>,  
and Kyung-Yong Chwa<sup>1</sup>

<sup>1</sup> Division of Computer Science, KAIST, Daejeon, Korea  
{swbae, stonecold, sunghee, kychwa}@tclab.kaist.ac.kr

<sup>2</sup> Department of Computer Science and Engineering,  
POSTECH, Pohang, Korea  
heekap@postech.ac.kr

**Abstract.** We consider two non-convex enclosing shapes with the minimum area; the *L-shape* and the *quadrant hull*. This paper proposes efficient algorithms computing each of two shapes enclosing a set of points with the minimum area over all orientations. The algorithms run in time quadratic in the number of given points by efficiently maintaining the set of extremal points.

## 1 Introduction

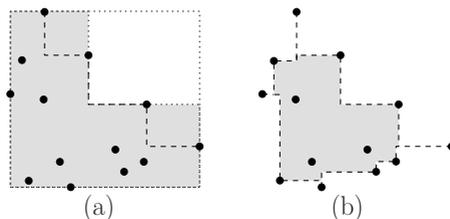
Given a set of geometric objects in the plane, there has been a fair amount of work on the smallest enclosing shapes (such as the convex hull, the smallest enclosing disk or the minimum enclosing rectangle) of the objects [2, 10, 11, 12].

In many cases, the enclosing shape is invariant to orientation, that is, the shape does not change over all orientations. Therefore, the enclosing shape for any fixed orientation is the optimal enclosing shape over all orientations (for example, the smallest enclosing circle and the convex hull.) If, however, this is not the case, that is, if the enclosing shape for some fixed orientation changes over different orientations, computing an optimal enclosing shape over all orientations becomes more difficult (for example, the minimum enclosing rectangle of points in the plane.)

Given a set  $P$  of points in the plane, we consider two enclosing shapes which are non-convex: the L-shape and the quadrant hull. For a fixed orientation, the *L-shape* can be defined as  $R \setminus R'$ , where  $R$  is an axis-aligned rectangle and  $R'$  is a sub-rectangle of  $R$  containing the upper right corner of  $R$ . Thus, the smallest enclosing L-shape  $\mathcal{L}(P)$  of  $P$  can be found by taking  $R$  as the smallest enclosing rectangle of  $P$  and  $R'$  as the largest empty sub-rectangle of  $R$  containing the upper right corner of  $R$ .

The quadrant hull  $\mathcal{QH}(P)$  of  $P$  is defined as follows: For a fixed orientation, a *quadrant* is the intersection of two half-planes whose supporting lines are axis-aligned and make the right angle. We call a quadrant *free* with respect to  $P$  if its interior contains no point in  $P$ . Then, the *quadrant hull*  $\mathcal{QH}(P)$  of  $P$  is

$$\mathcal{QH}(P) := \mathbb{R}^2 - \bigcup_{Q \text{ quadrant free to } P} Q.$$



**Fig. 1.** The minimum enclosing L-shape and the quadrant hull of the same set of points in a fixed orientation

Another equivalent definition of the quadrant hull was suggested by Matoušek [4]. The quadrant hull  $\mathcal{QH}(P)$  is also known as *orthogonal convex hull*, which for a fixed orientation can be constructed in time  $O(n \log n)$  [7, 8, 9], or faster using integer searching data structures for points with integer coordinates [3].

In this paper, we present efficient algorithms computing a smallest L-shape and quadrant hull of  $n$  given points in the plane over all orientations. In doing so, we reveal relations between both enclosing shapes and extremal points. We call a point  $p$  in  $P$  *extremal* if there is no such point  $q \in P$  that  $q_x > p_x$  and  $q_y > p_y$ , where  $p_x$  and  $p_y$  are the  $x$ - and  $y$ -coordinates of  $p$  in a specific coordinate system.<sup>1</sup> Extremal points form a linear order, for instance,  $x$ -coordinate increasing order, and we can build a *staircase* from those ordered points in a natural way (See the dashed segments in Figure 1.) Observe that such a staircase can be described by a sequence of free quadrants supporting two consecutive extremal points.

A minimum enclosing L-shape for a fixed orientation can be obtained by first computing the staircase in the upward and the rightward directions and then picking the best one among the pairs of consecutive points which describe the staircase as in Figure 1(a). On the other hand, the quadrant hull  $\mathcal{QH}(P)$  can be described by four staircases as in Figure 1(b). In either case, staircases can be computed in  $O(n \log n)$  time, and they allow us in linear time to compute both minimum-area enclosing shapes for a fixed orientation. Therefore, if we could maintain the staircases efficiently while we rotate the coordinate system, it would be helpful to use them in finding an optimal orientation for both enclosing shapes.

Following this motivation, we consider the problem of maintaining the staircase over all orientations in Section 2. We first present a simple, quadratic-time algorithm and then a subquadratic solution with a time/space tradeoff. Sections 3 and 4 are devoted to explaining the algorithms for optimal orientations for both enclosing shapes, which implicitly maintain the staircase and run in quadratic time. Finally, we conclude this paper in Section 5.

## 2 Maintaining the Staircase

Here, we rotate our coordinate system. We shall denote by *orientation*  $\theta$  the coordinate system with axes rotated by  $\theta$  around the origin in counter-clockwise direction. Let  $P$

<sup>1</sup> In this paper, we deal only with orthogonal coordinate systems where two axes make the right angle.

be a set of  $n$  points. We denote by  $X_\theta(P)$  the set of extremal points of  $P$  in orientation  $\theta$ . Let  $\prec_\theta$  be the order of increasing  $x$ -coordinate on  $P$  in orientation  $\theta$ . For  $X_\theta(P) = \{p_1, \dots, p_k\} \subset P$ , where the indices are given in order  $\prec_\theta$ , we draw two axis-parallel rays until both meet; one downwards from  $p_i$  and the other leftwards from  $p_{i+1}$ . Then we get a step  $s_\theta(p_i, p_{i+1})$  between  $p_i$  and  $p_{i+1}$  with  $1 \leq i < k$  and we denote such a sequence of steps by the staircase  $S_\theta(P)$  of  $P$  in orientation  $\theta$ . Observe that each quadrant obtained by extending a step is free to  $P$ .

Our goal in this section is to maintain  $X_\theta(P)$  and  $S_\theta(P)$  while  $\theta$  increases from 0 to  $2\pi$ . Once the points in  $X_\theta(P)$  are given in order  $\prec_\theta$ , we can easily build  $S_\theta(P)$  from  $X_\theta(P)$ . Therefore, we would like to efficiently update  $X_\theta(P)$  to obtain  $X_{\theta+\varepsilon}(P)$  for sufficiently small positive  $\varepsilon$ . To achieve our goal, we focus on when a change between  $X_\theta(P)$  and  $X_{\theta+\varepsilon}(P)$  occurs. We call an orientation  $\varphi \in [0, 2\pi)$  an *event orientation* if  $X_{\varphi-\varepsilon}(P) \neq X_{\varphi+\varepsilon}(P)$  for any  $\varepsilon > 0$ . At each event orientation, an “event” occurs; either a new point in  $P$  appears at the staircase or an existing one disappears. We call the former type of events *in-events* and the latter *out-events*.

We use standard data structures. The *event queue*  $\mathcal{Q}$  is a priority queue which stores events indexed by their occurring time (or, orientation). We also store the points in  $X_\theta(P)$  in a balanced binary search tree  $\mathcal{T}$ , in order  $\prec_\theta$ , so that we can add and delete a point in  $O(\log n)$  time.

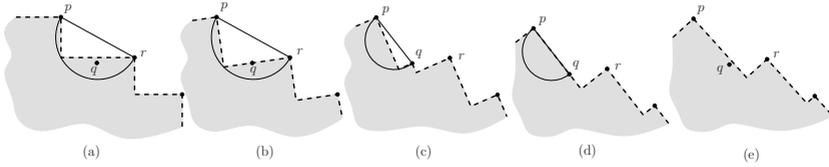
If an in-event occurs at orientation  $\theta$ , a point  $q \in P \setminus X_{\theta-\varepsilon}(P)$  appears to  $X_\theta(P)$  and also to  $S_\theta(P)$ . Say that  $q$  appears between  $p$  and  $r$  in  $S_\theta(P)$ . Then, at orientation  $\theta$ , the step  $s_\theta(q, r)$  between  $q$  and  $r$  degenerates to a line segment, that is,  $q$  lies on  $s_\theta(p, r)$ . Similarly, when an out-event occurs and  $q$  between  $p$  and  $r$  with  $p, q, r \in S_\theta(P)$  is about to disappear,  $s_\theta(p, q)$  degenerates to a line segment and  $q$  lies on  $s_\theta(p, r)$ . Figure 2 shows some changes on  $S_\theta(P)$  as  $\theta$  increases, including an in-event and an out-event.

**Observation 1.** *When an event occurs at orientation  $\theta$ , there is a degenerate step in  $S_\theta(P)$  and one of its two corresponding points is the event point.*

Now, we consider the disk with diameter  $pr$  with  $p$  and  $r$  consecutive in  $X_\theta(P)$  with respect to  $\prec_\theta$ , and its half-disk  $D(p, r)$  containing the triangle  $T_\theta$  defined by  $s_\theta(p, r)$  and  $pr$ . See Figure 2(a). The corner of  $s_\theta(p, r)$  goes along the circular arc of  $D(p, r)$  counter-clockwise as  $\theta$  increases, since  $T_\theta$  is always a right triangle. We denote two segments of  $s_\theta(p, r)$  by  $s_\theta^h(p, r)$  and  $s_\theta^v(p, r)$ , horizontal and vertical ones. Observe that as  $\theta$  increases  $s_\theta^h(p, r)$  sweeps region in  $D(p, r)$ , and if it encounters a point  $q \in P$ , we have an in-event and  $q$  comes up to the staircase. Among in-events are those corresponding to new extreme points in  $y$ -direction; just imagine that we have one more point at infinity in  $-x$  direction whose  $y$ -coordinate is the same as the point with largest  $y$ -coordinate in the current orientation. Also, when  $s_\theta^h(p, q)$  degenerates to  $q$  we have an out-event and  $q$  will disappear from the staircase. See Figure 2.

**Observation 2.** *Let  $p$  and  $r$  be two consecutive points in  $X_\theta(P)$  with respect to  $\prec_\theta$ . The next upcoming in-event between  $p$  and  $r$  occurs by another point  $q$  that is first encountered by  $s_\theta^h(p, r)$  among points in  $D(p, r)$  as  $\theta$  increases. The out-event of  $q$  occurs when  $s_\theta^h(p, q)$  degenerates to  $q$ .*

By those observations, we know that every event can be captured locally. It is easy in  $O(n)$  time to predict the very next in-event between two consecutive extremal points



**Fig. 2.** Changes on  $S_\theta(P)$  as  $\theta$  increases; (a)  $q$  lies in  $D(p, r)$ . (b)  $s_\theta^h(p, r)$  hits  $q$ , an in-event occurs, and  $q$  rises up. (c)  $S_\theta(P)$  contains  $q$ . (d)  $s_\theta^h(p, q)$  degenerates to point  $q$ , an out-event occurs, and  $q$  is about to disappear. (e) After the out-event. All the other points in  $P$  lie in the shaded side of each figure.

with respect to  $\prec_\theta$  and the out-event of each extremal point at the current orientation. Indeed, one can reduce this time complexity to predict next events to sublinear by using some complicated data structures, which will be discussed at the end of this section.

Initially, we compute  $X_0(P)$  and  $S_0(P)$ , and store it into  $\mathcal{T}$  in  $O(n \log n)$  time. Then, we predict in-events and out-events corresponding to  $S_0(P)$  and insert them into  $\mathcal{Q}$ . Now, we are ready to run the main loop. As usual, we extract the upcoming event from the event queue  $\mathcal{Q}$ , and handle it according to its type. We end this loop if the current orientation  $\theta$  is at least  $2\pi$ :

- In-event.** We put the new point  $q$  between  $p$  and  $r$  into  $\mathcal{T}$ . Compute the in-events between  $p$  and  $q$ , and between  $q$  and  $r$ ; also the out-event of  $q$ . Insert all the computed events into  $\mathcal{Q}$ .
- Out-event.** Remove the disappearing point  $q$  between  $p$  and  $r$  from  $\mathcal{T}$ , and in-events in which  $q$  is involved from  $\mathcal{Q}$ . Compute the in-event between  $p$  and  $r$ , and insert it into  $\mathcal{Q}$ .

The total running time is proportional to the number of events we handle during the algorithm. The following lemma answers the essential question.

**Lemma 3.** Any point in  $P$  can appear in  $X_\theta(P)$  at most four times as  $\theta$  increases from 0 to  $2\pi$ .

*Proof.* We claim that if a point  $q \in X_\theta(P)$  is about to disappear at orientation  $\theta$ , then  $q$  cannot appear again to  $X_{\theta+\varphi}(P)$  for  $0 < \varphi < \pi/2$ . This simply implies the lemma.

We now prove our claim. Assume that  $q \in X_\theta(P)$  is about to disappear from  $X_\theta(P)$ , that is, the out-event of  $q$  occurs at  $\theta$ . Recall by definition that a point  $q' \in X_\theta(P)$  if and only if the intersection of two quadrants by extending two steps incident on  $q'$  is free to  $P$ . (Obviously, the intersection is a quadrant again.) Thus, in orientation  $\theta$ , observe that there is a point  $p \in P$  directly above  $q$  since  $q$  will disappear after  $\theta$  (See Figure 2(d).)

Note that maintaining the staircase is invariant under rigid motions. Also, increasing  $\theta$  is equivalent to rotating the points  $P$  clockwise. Now, we transform  $P$  by an affine mapping so that  $q$  is mapped to the origin. Instead of increasing  $\theta$ , we rotate  $p$  clockwise around  $q$ . It is easy to see that  $p$  lies in the first quadrant until we rotate the points by more than  $\pi/2$ . Thus, we conclude our claim and then the lemma.  $\square$

Finally, we conclude one of our main results.

**Theorem 4.** *The staircase has at most  $O(n)$  combinatorial changes while rotating the orientation. These changes can be maintained in  $O(n^2)$  time and  $O(n)$  space.*

**2.1 Predicting an In-Event in Sublinear Time**

In this subsection, we present a faster way to maintain the staircase. Indeed, the bottleneck of our algorithm is the part of predicting an in-event. Note that predicting an out-event can be performed in constant time. Here, we present how to predict an in-event in sublinear time so that the total running time of the algorithm reduces to subquadratic, and moreover a time/space tradeoff can be obtained.

Observe that the next in-event between  $p$  and  $q$  occurs by a point contained in the lune shape  $L_\theta(p, q)$  defined by  $D(p, q)$  and the half-plane below  $s_\theta^h(p, q)$ . The first key idea is to restrict candidates to those inside  $L_\theta(p, q)$  by the range searching with such a lune shape, which is the intersection of a disk and a half-plane. Since a disk range on the plane can be processed by a half-space range in 3 dimensional space by a well-known lifting-up transformation, the lune shape range searching in  $\mathbb{R}^2$  can be viewed as the range searching with intersections of two half-spaces in  $\mathbb{R}^3$ . Thus we adopt the range searching structure by Matoušek:

**Theorem 5 (Matoušek [6]).** *Let  $P$  be an  $n$ -point set in  $\mathbb{R}^d$  and let  $m$  be a parameter,  $n \leq m \leq n^d$ . The range searching problem with the ranges being intersections of  $p$  half-spaces,  $1 \leq p \leq d + 1$ , can be solved with space  $O(m)$ , query time  $O\left(\frac{n}{m^{1/d}} \log^{p-(d-p+1)/d} \frac{m}{n}\right)$ , and preprocessing time  $O(n^{1+\delta} + m(\log n)^\delta)$ , for  $\delta > 0$ .*

We fix  $d = 3$  and  $p = 2$  so that we have a structure  $\mathcal{R}_m(P)$  for the lune range searching among  $P$ . This structure can report all the points in the queried lune shape  $L_\theta(p, q)$ , but what we need is only one point that will cause the in-event between  $p$  and  $q$ . In order to get such a point without reporting other points in the range, we make use of special secondary data structures. Note that the range searching structure  $\mathcal{R}_m(P)$  is based on a partition tree [1, 5, 6]. Thus, with each internal node  $v$ , we associate a secondary structure  $\mathcal{C}(P_v)$  for its canonical subset  $P_v \subseteq P$ . The structure  $\mathcal{C}(P_v)$  is supposed to be able to answer a point  $r \in P_v$  which we first meet with the line supporting  $s_\theta^h(p, q)$  rotated counter-clockwise at  $q$ . Such a point  $r$  should be in convex position and indeed be the contact point of a tangent line to  $\text{conv}(P_v)$  through  $q$ . Hence, once we have computed the convex hull  $\text{conv}(P_v)$  of  $P_v$ , it can be done in time  $O(\log |P_v|) = O(\log n)$ .

Processing a query  $L_\theta(p, q)$ , we search  $\mathcal{R}_m(P)$  and get at most  $O(n/m^{1/3} \log^{4/3} m/n)$  number of canonical subsets. Then, at each corresponding internal node of  $\mathcal{R}_m(P)$  we find the same number of candidates for the in-event point. This query process takes  $O(n/m^{1/3} \log^{7/3} m/n)$  time. The preprocessing and the storage need an extra factor of  $O(\log m) = O(\log n)$ , since  $\mathcal{R}_m(P)$  has depth at most  $O(\log m)$ . Thus, we can conclude the following time/space tradeoff result.

**Theorem 6.** *While rotating the orientation, the staircase of  $n$  points can be maintained in  $O\left(n^{1+\delta} \log n + m(\log n)^{1+\delta} + \frac{n^2}{m^{1/3}} \log^{7/3} \frac{m}{n}\right)$  time and  $O(m \log n)$  storage, where  $m$  is a parameter with  $n \leq m \leq n^2$ .*

Consequently, if we choose  $m = n^{3/2}$ , then the above theorem yields an algorithm which runs in  $O(n^{3/2} \log^{7/3} n)$  time and  $O(n^{3/2} \log n)$  space.

### 3 Minimum Enclosing L-Shapes

In this section, we present an efficient algorithm for computing an optimal orientation for minimum-area enclosing L-shapes of a set  $P$  of  $n$  points. First, we observe the following.

**Observation 7.** *In any orientation, the smallest enclosing L-shape of  $P$  touches at least one point in  $P$  on each of the six sides of its boundary.*

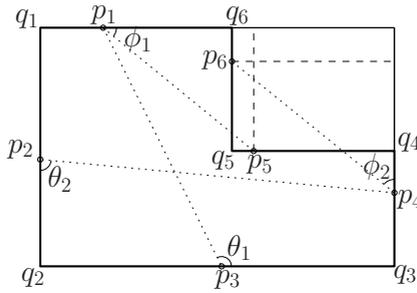
Thus, we can restrict our candidate L-shapes for each side to have at least one point, and can describe them as these (possibly not disjoint) six points. Note that these points can be shared by two adjacent sides, or possibly by more than two sides when some of them have length zero. Let  $R_\theta$  be the smallest enclosing rectangle of  $P$  in orientation  $\theta$  and  $E_\theta$  be the largest empty rectangle which shares the upper right corner with  $R_\theta$ . Then the smallest enclosing L-shape  $\mathcal{L}_\theta(P)$  is represented by  $\mathcal{L}_\theta(P) = R_\theta - E_\theta$ . Let  $p_1, \dots, p_6$  be six points touching each side of  $\mathcal{L}_\theta(P)$ ; the index is ordered counter-clockwise from the point on the top side of  $R_\theta$  as shown in Figure 3. Now, we assume  $p_i$ 's with  $1 \leq i \leq 6$  to represent the smallest L-shapes  $\mathcal{L}_\theta(P)$  during orientation  $0 \leq \alpha < \theta < \beta < 2\pi$ ; that is, for any  $\alpha < \theta < \beta$ , the boundary of  $\mathcal{L}_\theta(P)$  touches the same sequence of six points in  $P$ . The following lemma shows how to get a local optimal orientation over  $[\alpha, \beta]$ .

**Lemma 8.** *One can minimize the area of  $\mathcal{L}_\theta(P)$  over  $0 \leq \alpha \leq \theta \leq \beta < 2\pi$ , where we have the same sequence of six points representing  $\mathcal{L}_\theta(P)$  in orientation  $\theta \in (\alpha, \beta)$ .*

*Proof.* We use all the symbols as defined above. Furthermore, we need more terms. Define  $d_1 := \text{length}(p_1p_3)$ ,  $d_2 := \text{length}(p_2p_4)$ ,  $c_1 := \text{length}(p_1p_5)$ , and  $c_2 := \text{length}(p_4p_6)$ . Let  $q_1, \dots, q_6$  be the six corners of  $\mathcal{L}_\alpha(P)$  that are ordered by traversing its boundary counter-clockwise from the upper left corner  $q_1$ . In orientation  $\alpha$ , we consider following angles as in Figure 3: Let  $\theta_1 := \angle p_1p_3q_3$ ,  $\theta_2 := \angle p_4p_2q_2$ ,  $\phi_1 := \angle p_5p_1q_6$ , and  $\phi_2 := \angle p_6p_4q_4$ . We then have  $\text{area}(R_{\alpha+\delta}) = d_1d_2 \sin(\theta_1 - \delta) \sin(\theta_2 - \delta)$  and  $\text{area}(E_{\alpha+\delta}) = c_1c_2 \sin(\phi_1 + \delta) \sin(\phi_2 - \delta)$ , where  $0 \leq \delta \leq \beta - \alpha$ . Thus, we can express the area of  $\mathcal{L}_\theta(P)$  as a function  $A_L$  of  $\delta$ :  $A_L(\delta) := \text{area}(\mathcal{L}_{\alpha+\delta}(P)) = \text{area}(R_{\alpha+\delta}) - \text{area}(E_{\alpha+\delta})$ . In order to obtain the minimum of  $A_L$  on  $(0, \beta - \alpha)$ , if any, we solve the equation  $A'_L(\delta) = 0$ , where  $A'_L$  is the first derivative of  $A_L$ . Through some tedious work on equations, we get  $A'_L(\delta) = d_1d_2 \sin(2\delta - \theta_1 - \theta_2) + c_1c_2 \sin(2\delta + \phi_1 - \phi_2)$ . Solving  $A'_L(\delta) = 0$ , we have  $\delta = \frac{\theta_1 + \theta_2}{2} - \frac{1}{2} \arctan\left(\frac{c_1c_2 \sin(\theta_1 + \theta_2 + \phi_1 - \phi_2)}{d_1d_2 + c_1c_2 \cos(\theta_1 + \theta_2 + \phi_1 - \phi_2)}\right)$ .

Since  $0 < \delta < \beta - \alpha < 2\pi$ , the equation  $A'_L(\delta) = 0$  has at most a constant number of solutions in the domain, which are either local minima or maxima. A local optimal orientation can be searched among those orientations. □

Now, we are ready to conclude the main theorem of this section.



**Fig. 3.** The smallest L-shape  $\mathcal{L}_\alpha(P)$  of  $P$  in orientation  $\alpha$  with six defining points and other properties

**Theorem 9.** Given a set  $P$  of  $n$  points, one can decide an optimal orientation, which minimizes the area of L-shape  $\mathcal{L}_\theta(P)$  enclosing  $P$  over  $0 \leq \theta < 2\pi$ , in  $O(n^2)$  time and  $O(n)$  space.

*Proof.* In this proof, we present an algorithm for computing optimal orientations for L-shapes, which applies the algorithm maintaining the staircase presented in Section 2. Also, we make use of the method described in the proof of Lemma 8. By Lemma 8, if we can fix one point on each side of  $\mathcal{L}_\theta(P)$  during  $\alpha \leq \theta \leq \beta$  for some  $0 \leq \alpha \leq \beta$ , then we are able to decide the local optimum in the domain for those at most 6 points.

We need to handle another type of events that occur when the current orientation  $\theta$  is parallel or perpendicular to an edge of  $\text{conv}(P)$ . Such an event is easy to predict once we have computed  $\text{conv}(P)$ . The total number of all events still remains linear in  $n$ . Let  $\alpha_i$  be the event orientations ordered by their occurrences. For each interval  $[\alpha_i, \alpha_{i+1}]$ , we have the same four points determining  $R_\theta$  for every  $\theta \in (\alpha_i, \alpha_{i+1})$  and further  $X_\theta(P)$  does not change in the interval. Recall that  $\mathcal{L}_\theta(P) = R_\theta - E_\theta$  and  $E_\theta$  is determined by one step of  $X_\theta(P)$ . Thus, we minimize every L-shape determined by each of step in  $X_\theta(P)$  in the interval  $[\alpha_i, \alpha_{i+1}]$ , and then pick the minimum of these minima. This in  $|X_\theta(P)|$  time gives us the local optimal orientation in the orientation interval. Since we have  $O(n)$  number of such intervals, all the process can be done in  $O(n^2)$  time in order to get a global optimum from gathered local optima.  $\square$

Note that the bottleneck of our algorithm is part of computing a local optimum in each step, consuming at most  $O(n)$  time. If one could improve this, a subquadratic algorithm will be obtained.

One might wonder if there exists such a nice nature that, for instances, optimal orientations are parallel or perpendicular to an edge of the convex hull of  $P$ , or there are at least two points on a segment of the boundary of the L-shape in optimal orientation. If so, one can easily compute the smallest enclosing L-shape over all orientations by testing a small number of candidate orientations. However, L-shapes do not have such a property.

See Figure 4. Let  $P := \{(1, \tan \epsilon), (\tan \epsilon, 1), (0, 0)\}$ . Then, for  $0 < \epsilon < \pi/12$ , it is not difficult to check that 0 is the only optimal orientation, which does not come from any pair of points in  $P$ . Precisely, there exists such a set  $P$  of points that no orientations

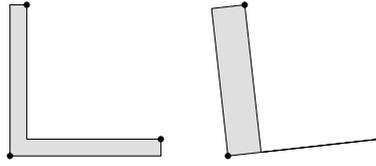


Fig. 4. Three points  $P$  and their enclosing L-shapes  $\mathcal{L}_0(P)$  and  $\mathcal{L}_\epsilon(P)$

parallel or perpendicular to a line through a pair of points in  $P$  are optimal for the area of L-shapes.

### 4 Minimum Quadrant Hulls

In this section, we consider how to find an optimal orientation for quadrant hulls. As aforementioned, we make use of maintaining the staircase described in Section 2 to get an efficient solution. First, recall that the quadrant hull  $\mathcal{QH}_\theta(P)$  in orientation  $\theta$  can be described by four staircases,  $S_\theta(P)$ ,  $S_{\theta+\frac{\pi}{2}}(P)$ ,  $S_{\theta+\pi}(P)$ , and  $S_{\theta+\frac{3\pi}{2}}(P)$ . We call two staircases  $S_\theta(P)$  and  $S_{\theta'}(P)$  *opposite staircases* if  $\theta$  and  $\theta'$  differ by  $\pi$ , and *adjacent staircases* if  $\theta$  and  $\theta'$  differ by  $\pi/2$ . We denote the set of the four staircases by  $\mathcal{S}_\theta(P)$ . Note that  $\mathcal{QH}_\theta(P) = \mathcal{QH}_{\pi/2+\theta}(P)$  and  $S_\theta(P) = \mathcal{S}_{\pi/2+\theta}(P)$ . We are thus interested only in orientations up to  $\pi/2$ , and each event orientation  $\varphi = \frac{\pi}{2}k + \varphi'$  with  $0 \leq \varphi' < \frac{\pi}{2}$  will be processed at  $\varphi'$ .

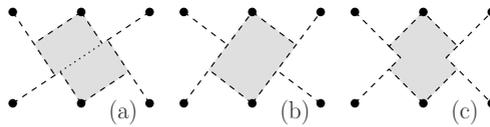
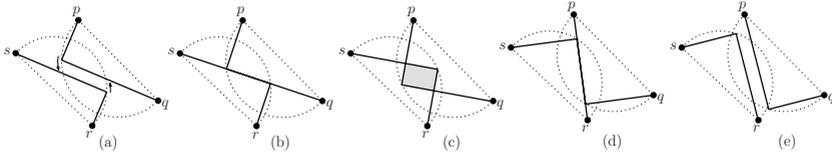


Fig. 5. A set  $P$  of 6 points and its quadrant hulls in three orientations

As in the L-shape problem, there seems no hope to restrict the number of candidate orientations for the minimum quadrant hull to a small number. Figure 5 shows a set  $P$  of points such that no orientations parallel or perpendicular to a line through a pair of points in  $P$  are optimal. Explicitly,  $P = \{(-3, -2), (0, -2), (3, -2), (-3, 2), (0, 2), (3, 2)\}$ , and Figure 5(a) and (b) show  $\mathcal{QH}_{\tan^{-1} \frac{2}{3}}(P)$  and  $\mathcal{QH}_{\tan^{-1} \frac{3}{4}}(P)$ ; these two orientations come from pairs of the points. However,  $\mathcal{QH}_{\pi/4}(P)$  as shown in Figure 5(c) is indeed optimal.

We thus devise a solution to the quadrant hull in a similar way as done in L-shapes. Furthermore, we have another difficulty in calculating the area of  $\mathcal{QH}_\theta(P)$ ; as seen in Figure 6, two staircases among those in  $\mathcal{S}_\theta(P)$  can cross each other. We observe the following about such overlaps.

**Observation 10.** *At most one pair of opposite staircases can cross each other. If a step crosses another, it does not cross any other and thus all such overlaps are axis-aligned rectangles.*



**Fig. 6.** (a) Two steps in opposite staircases. (b) A step-cross event occurs. (c) The shaded region is the overlap. (d) A step-release event occurs. (e) Two steps are separated.

Considering overlaps, we should refine the definition of combinatorially equivalent quadrant hulls since the shape of  $\mathcal{QH}_\theta(P)$  can be changed regardless of the equivalence of  $\mathcal{S}_\theta(P)$ . Thus we take how  $\mathcal{S}_\theta(P)$  crosses also into account, and define *step-cross* and *step-release* events as when two steps in opposite staircases start crossing and when both become free to each other, respectively.

Predicting and handling these events including in-events and out-events is not very difficult if we are allowed  $O(n)$  time: For a step  $s$  in  $\mathcal{S}_\theta(P)$ , we check all steps in the opposite staircase to  $s$  whether and when each will cross  $s$ , and pick the earliest one. We call the resulting event “the step-cross event of a step  $s$ ”. The step-release event of a pair of crossed steps is easy to compute.

**In-event.** An in-event removes a step but creates two new steps. We perform the same operations as in maintaining the staircase. Further, compute the corresponding step-cross events, if any, and insert them into  $\mathcal{Q}$  and delete the useless step-cross event of the removed step from  $\mathcal{Q}$ .

**Out-event.** Similarly, compute corresponding step-cross events and delete useless events from  $\mathcal{Q}$ .

**Step-cross event.** Compute the step-release event of the involved pair of steps and insert it to  $\mathcal{Q}$ .

**Step-release event.** Compute the step-cross event of each of the two involved steps and insert them into  $\mathcal{Q}$ .

**Lemma 11.** *We have at most  $O(n)$  events in total.*

*Proof.* In Section 2, we have shown that there are at most  $O(n)$  number of in-events and out-events. Here, we show that we have only  $O(n)$  step-cross events, which implies the same number of step-release events, and thus at last  $O(n)$  events in total.

If we have a step-cross event at  $\theta$ , we can see two corresponding steps just meeting and a segment joining two points among the four points in the two steps. In Figure 6(b), we can see segment  $sq$  as the union of horizontal segments of two steps at the moment of the step-cross event. We call the segment  $sq$  a *step-cross segment*. Now, we draw such a step-cross segment whenever we have a step-cross event. We claim that no two step-cross segments cross each other. This implies that we have drawn a plane graph on  $n$  points and thus we have at most  $O(n)$  such segments.

Let  $G = (P, E)$  be a graph whose vertices are  $P$  and edges are drawn as explained above. Assume to the contrary that there exist  $ab, pq \in E$  such that the two corresponding step-cross segments cross. There should be involved steps, namely,  $s_{\theta_1}(b, c)$  and  $s_{\theta_1}(d, a)$ , and  $s_{\theta_2}(q, r)$  and  $s_{\theta_2}(s, p)$ . Without loss of generality, assume that  $a$  is to the

left of  $b$  in orientation  $\theta_1$ . We can find two lines  $\ell_1$  and  $\ell_2$  perpendicular to  $ab$  and  $pq$ , respectively, that separate  $c$  and  $d$ , and  $r$  and  $s$ , respectively.

In orientation  $\theta_1$ ,  $\ell_1$  and the line supporting  $ab$  partition the plane into four quadrants. We denote the left upper region by  $LU_1$  and the right lower region by  $RL_1$ .  $c \in LU_1$  and  $d \in RL_1$ . Note that there is no point in the right upper and the left lower regions since these two regions are contained in the two quadrants defined by two steps  $s_{\theta_1}(b, c)$  and  $s_{\theta_1}(d, a)$ . Thus,  $p$  and  $q$  must be separated by  $\ell_1$  since  $ab$  and  $pq$  cross. Now, we can assume that  $p \in LU_1$  and  $q \in RL_1$ . Also in orientation  $\theta_2$ , the line supporting  $pq$  and  $\ell_2$  partition the plane into four regions. Similarly, we define two regions  $LU_2$  and  $RL_2$ , and then  $r \in LU_2$  and  $s \in RL_2$ .

Since  $ab$  and  $pq$  cross and  $p \in LU_1$ , we know that  $a$  lies below  $pq$  and  $b$  lies above  $pq$ . Thus,  $a$  must lie in  $RL_2$  and  $b$  in  $LU_2$ . This implies that  $a$  lies to the left of  $\ell_1$  and to the right of  $\ell_2$ , and  $b$  lies to the right of  $\ell_1$  and to the left of  $\ell_2$ .  $\ell_1$  and  $\ell_2$  divide the plane into four cones. We let  $A$  be one of the cones where  $a$  is located and  $B$  be where  $b$  is located. The angle of  $A$  (or  $B$ ) at apex  $\ell_1 \cap \ell_2$  should be larger than  $\pi/2$  since  $\ell_1$  and  $ab$  are perpendicular. However, this is turned to be false: Since  $p \in LU_1$  and  $q \in RL_1$ ,  $p$  lies to the left of  $\ell_1$  and of  $\ell_2$ , and  $q$  lies to the right of  $\ell_1$  and of  $\ell_2$ . Since  $\ell_2$  and  $pq$  are also perpendicular, it is required for other cones than  $A$  and  $B$  to have angle at the apex larger than  $\pi/2$ , which leads to a contradiction to our assumptions.  $\square$

The main loop is performed while  $\theta < \pi/2$ ; since we handle four staircases at the same time, all the required events occur before  $\pi/2$ . The space of orientations is partitioned into a linear number of intervals  $[\alpha_i, \alpha_{i+1})$ , where  $i$  runs from 0 to the total number of events. The following subsection gives us a way to compute a local optimum in each such interval in linear time.

### 4.1 Finding Local Optima of $\mathcal{QH}_\theta(P)$

Here, we assume that an orientation interval  $(\alpha, \beta)$  does not contain any event orientation and the current orientation  $\theta$  runs in between  $(\alpha, \beta)$ . Thus, we can also say that  $S_\theta(P)$  contains  $m$  points of  $P$  and  $k$  overlaps. Let  $p_1$  be the first point in  $S_\theta(P)$  (or the point with highest  $y$  coordinate in orientation  $\theta$ ), and  $p_i$  be the  $i$ -th point in  $S_\theta(P)$  in clockwise order. We get a polygon  $\mathcal{P}$  by a sequence of  $m$  sides  $p_i p_{i+1}$  with  $1 \leq i \leq m-1$ , and  $p_m p_1$ . Observe that two sides cannot cross each other so that we can compute the area of  $\mathcal{P}$  in  $O(m)$  time. Recall that since we have no event in  $\theta \in (\alpha, \beta)$ , these extremal points and  $\mathcal{P}$  do not change in the interval. Also, we let  $s_i(\theta)$  be the step of  $p_i$  and  $p_{i+1}$  at  $\theta$ , and  $\Delta_i(\theta)$  be the right triangle defined by  $s_i(\theta)$  as in Section 2.

Let  $\square_j(\theta)$  be the rectangle defined by each overlap in  $S_\theta(P)$ , where  $1 \leq j \leq k$ . Then, the area of  $\mathcal{QH}_\theta(P)$  is  $\text{area}(\mathcal{QH}_\theta(P)) = \text{area}(\mathcal{P}) - \sum_i \text{area}(\Delta_i(\theta)) + \sum_j \text{area}(\square_j(\theta))$ . To minimize the area of  $\mathcal{QH}_\theta(P)$  over  $\theta \in (\alpha, \beta)$ , we analyze the functions  $\text{area}(\Delta_i(\theta))$  and  $\text{area}(\square_j(\theta))$ , and obtain a nice representation so that we can get an optimum in an analytic way.

**Lemma 12.** *Let  $(\alpha, \beta)$  with  $0 \leq \alpha < \beta < \pi/2$  be an orientation interval such that it contains no event orientation. Once we have computed  $S_\alpha(P)$  and  $\mathcal{QH}_\alpha(P)$ , it is possible in linear time to compute a local optimal orientation in which the area of  $\mathcal{QH}_\theta(P)$  is minimized over  $\theta \in (\alpha, \beta)$ .*

*Proof.* Let  $d_i$  be the length of the hypotenuse of  $\Delta_i(\alpha)$  and  $\phi_i$  be the internal angle of  $\Delta_i(\alpha)$  at  $p_i$ . Then,  $\text{area}(\Delta_i(\theta)) = d_i^2 \cos(\phi_i + (\theta - \alpha)) \sin(\phi_i + (\theta - \alpha))$ . Substituting  $\theta - \alpha$  by  $\varphi$ , we get  $\text{area}(\Delta_i(\theta)) = d_i^2 \cos(\phi_i + \varphi) \sin(\phi_i + \varphi) = \frac{1}{2}d_i^2 \sin 2(\phi_i + \varphi) = \frac{1}{2}d_i^2 \sin 2\phi_i \cos 2\varphi + \frac{1}{2}d_i^2 \cos 2\phi_i \sin 2\varphi$ , which is linear in  $\cos 2\varphi$  and  $\sin 2\varphi$ . Therefore, the sum over all  $i$  is of the same form:  $\sum_i \text{area}(\Delta_i(\theta)) = \frac{1}{2} (\sum_i d_i^2 \sin 2\phi_i) \cos 2\varphi + \frac{1}{2} (\sum_i d_i^2 \cos 2\phi_i) \sin 2\varphi$ .

Now, we deal with each overlap  $\square_j(\theta)$  in which two steps  $s_a(\theta)$  and  $s_b(\theta)$  are involved. Assume without loss of generality that  $s_a(\theta)$  belongs to  $S_\theta(P)$  and  $s_b(\theta)$  to  $S_{\pi+\theta}(P)$ . If the coordinate of  $p_i$  in orientation  $\alpha$  is  $(x_{p_i}, y_{p_i})$ ,  $\text{area}(\square_j(\alpha)) = |x_{p_a} - x_{p_b}| \times |y_{p_{a+1}} - y_{p_{b+1}}|$ , where  $|\cdot|$  returns the absolute value. The point  $p_i$  in orientation  $\theta = \alpha + \varphi$  can be regarded as the transformed points by rotation by  $-\varphi$  in orientation  $\alpha$ . Also, the area of  $\square_j(\theta)$  is invariant under translations. Thus,  $\text{area}(\square_j(\theta)) = \text{area}(\square_j(\alpha + \varphi)) = |x_{p'_a} - x_{p'_b}| \times |y_{p'_{a+1}} - y_{p'_{b+1}}| = |(x_{p_a} - x_{p_b}) \cos \varphi + (y_{p_a} - y_{p_b}) \sin \varphi| \times |(y_{p_{a+1}} - y_{p_{b+1}}) \cos \varphi - (x_{p_{a+1}} - x_{p_{b+1}}) \sin \varphi| = C_1 \cos^2 \varphi + C_2 \sin^2 \varphi + C_3 \sin \varphi \cos \varphi$ , where  $p'_i$  is the rotated point of  $p_i$  by  $-\varphi$  with  $\varphi < \pi/2$ , and  $C_1, C_2$ , and  $C_3$  are constants. This equation can be reformed to be linear in  $\sin 2\varphi$  and  $\cos 2\varphi$  again  $\text{area}(\square_j(\theta)) = C'_1 + C'_2 \cos 2\varphi + C'_3 \sin 2\varphi$ , where  $C'_1 = \frac{C_1 + C_2}{2}$ ,  $C'_2 = \frac{C_1 - C_2}{2}$ , and  $C'_3 = \frac{C_3}{2}$ .

Consequently, the area of  $\mathcal{QH}_\theta(P)$  with  $\alpha < \theta < \beta$  can be represented by a function  $f$  of  $\varphi$  which is linear in  $\sin 2\varphi$  and  $\cos 2\varphi$ , where  $0 < \varphi < \beta - \alpha$ . By solving  $f'(\varphi) = 0$ , where  $f'$  is the derivative of  $f$ , we get a local minimum (or maximum) of the area of  $\mathcal{QH}_\theta(P)$  with  $\theta \in (\alpha, \beta)$ . All this process is sufficiently done in linear time. □

Now, we conclude the following theorem.

**Theorem 13.** *Given a set  $P$  of  $n$  points, one can decide an optimal orientation, which minimizes the area of the quadrant hull  $\mathcal{QH}_\theta(P)$  over  $0 \leq \theta < 2\pi$ , in  $O(n^2)$  time and  $O(n)$  space.*

## 5 Concluding Remarks

We presented algorithms for computing the smallest enclosing L-shape and quadrant hull of given points for arbitrary orientation. Our solutions for finding smallest enclosing shapes internally maintain extremal points, constituting the staircase, over all orientations. Indeed, maintaining extremal points has its own interest. In computational geometry, dynamic data structures have been intensively researched over last several decades; they mainly focus on how to efficiently handle online updates of their underlying data. Our problem of maintaining the staircase deals with predictable updates over bounded domain, and was proven to be useful in solving geometric optimization problems which are variant to orientations. We carefully expect that many such optimization problems could be efficiently solved by maintaining a certain data structure over all orientations.

## References

- [1] Agarwal, P.K., Erickson, J.: Geometric range searching and its relatives. In: Chazelle, B., Goodman, J.E., Pollack, R. (eds.) *Advances in Discrete and Computational Geometry. Contemporary Mathematics*, vol. 233, pp. 1–56. American Mathematical Society Press, Providence (1999)
- [2] de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: *Computational Geometry: Algorithms and Applications*, 2nd edn. Springer, Heidelberg (2000)
- [3] Karlsson, R.G., Overmars, M.H.: Scanline algorithms on a grid. *BIT Numerical Mathematics* 28(2), 227–241 (1988)
- [4] Matoušek, J., Plecháč, P.: On functional separately convex hulls. *Discrete Comput. Geom.* 19, 105–130 (1998)
- [5] Matoušek, J.: Efficient partition trees. *Discrete Comput. Geom.* 8, 315–334 (1992)
- [6] Matoušek, J.: Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.* 10, 157–182 (1993)
- [7] Montuno, D.Y., Fournier, A.: Finding the  $x - y$  convex hull of a set of  $x - y$  polygons. Technical Report 148, University of Toronto (1982)
- [8] 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)
- [9] Ottman, T., Soisalon-Soisinen, E., Wood, D.: On the definition and computation of rectilinear convex hulls. *Information Sciences* 33, 157–171 (1984)
- [10] Preparata, F.P., Hong, S.J.: Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM* 20(2), 87–93 (1977)
- [11] Schwarzkopf, O., Fuchs, U., Rote, G., Welzl, E.: Approximation of convex figures by pairs of rectangles. In: *Proc. 7th Ann. Symp. on Theoretical Aspects of Computer Science*, pp. 240–249 (1990)
- [12] Welzl, E.: Smallest enclosing disks (balls and ellipsoids). In: Maurer, H.A. (ed.) *New Results and New Trends in Computer Science. LNCS*, vol. 555, pp. 359–370. Springer, Heidelberg (1991)