Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Computing minimum-area rectilinear convex hull and L-shape ☆

Sang Won Bae [a], Chunseok Lee [b], Hee-Kap Ahn [a,*], Sunghee Choi [b], Kyung-Yong Chwa [b]

[a] *Department of Computer Science and Engineering, POSTECH, Pohang, Republic of Korea*
[b] *Division of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea*

ARTICLE INFO

Article history:
Received 21 February 2008
Accepted 24 February 2009
Available online 24 March 2009
Communicated by T. Asano

Keywords:
Rectilinear convex hull
L-shape
Enclosing shapes
Extremal points
Staircases

ABSTRACT

We study the problems of computing two non-convex enclosing shapes with the minimum area; the *L-shape* and the *rectilinear convex hull*. Given a set of n points in the plane, we find an L-shape enclosing the points or a rectilinear convex hull of the point set with minimum area over all orientations. We show that the minimum enclosing shapes for fixed orientations change combinatorially at most $O(n)$ times while rotating the coordinate system. Based on this, we propose efficient algorithms that compute both shapes with the minimum area over all orientations. The algorithms provide an efficient way of maintaining the set of extremal points, or the staircase, while rotating the coordinate system, and compute both minimum enclosing shapes in $O(n^2)$ time and $O(n)$ space. We also show that the time complexity of maintaining the staircase can be improved if we use more space.

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 and square or the minimum area enclosing rectangle of the objects [2,10,12,13].

In many cases, the enclosing shape is invariant to orientation, that is, the shape is not dependent to the orientation of the coordinate system. Therefore, the enclosing shape for any fixed orientation is already 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, it becomes more difficult to compute the optimal orientation and an optimal enclosing shape over all orientations: 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 rectilinear convex hull. For a fixed orientation, the *L-shape* can be defined as $R \setminus R'$, where R and R' are axis-aligned rectangles sharing their upper right corner. Thus, the minimum area enclosing L-shape $\mathcal{L}(P)$ of P can be found by taking R as the minimum area enclosing rectangle of P and R' as the largest empty rectangle of R sharing its upper right corner with that of R.

The rectilinear convex hull $\mathcal{RH}(P)$ of P can be 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

☆ This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2007-331-D00372) and by the Brain Korea 21 Project.

[*] Corresponding author.
E-mail addresses: swbae@postech.ac.kr (S.W. Bae), stonecold@tclab.kaist.ac.kr (C. Lee), heekap@postech.ac.kr (H.-K. Ahn), sunghee@tclab.kaist.ac.kr (S. Choi), kychwa@tclab.kaist.ac.kr (K.-Y. Chwa).

0925-7721/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comgeo.2009.02.006

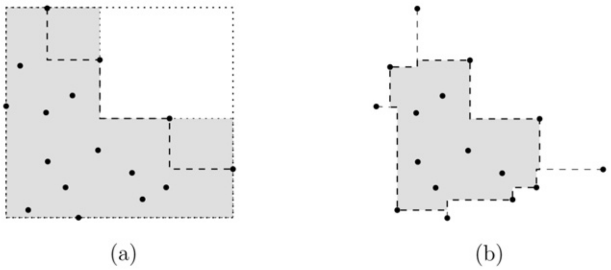

Fig. 1. The minimum enclosing L-shape and the rectilinear convex hull of the same set of points in a fixed orientation.

to P if its interior contains no point in P. Then, the *rectilinear convex hull* $\mathcal{RH}(P)$ of P is

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

The rectilinear convex hull $\mathcal{RH}(P)$ is also known as *orthogonal convex hull*, because any horizontal or vertical line intersects the hull in at most one connected component. For a fixed orientation the rectilinear convex hull can be constructed in time $O(n \log n)$ [7–9], or faster using integer searching data structures for points with integer coordinates [3]. Matoušek and Plecháč [6] studied more general concept of convexity, namely *D-convex sets* and *functional D-convex hulls*, which are defined by a set D of vectors in $\mathbb{R}^d$. According to the definition of D-convex hulls, the rectilinear convex hull $\mathcal{RH}(P)$ of P is the D-convex hull of P where $D = \{(0,1),(0,-1),(1,0),(-1,0)\}$.

In this paper, we present efficient algorithms computing a minimum area L-shape and a minimum area rectilinear convex hull of n 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.[1] We can define a linear order of extremal points, for instance, the order of non-decreasing x-coordinates, and we can build a *staircase* from these ordered points in a natural way. Fig. 1(a) shows a staircase of extremal points sorted in the order of non-decreasing x-coordinates. Observe that such a staircase can also 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 computing the staircase in the upward and the rightward directions, after then by picking the best one among the pairs of consecutive points which describe the staircase as in Fig. 1(a). On the other hand, the rectilinear convex hull $\mathcal{RH}(P)$ can be described by four staircases as in Fig. 1(b). In either case, staircases can be computed in $O(n \log n)$ time, and then we can compute both minimum-area enclosing shapes for a fixed orientation in linear time. 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 time solution with a time/space trade-off. 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.

As our algorithms deal with real values and involve trigonometric functions, we assume our model of computation to be the *real RAM* [11] throughout the paper.

2. Maintaining the staircase

Here we rotate our coordinate system. We shall denote by *orientation* θ the coordinate system with axes rotated by θ around the origin in counter-clockwise direction. Let P be a set of n points. We denote by $X_\theta(P)$ the set of extremal points of P in orientation θ. Let $\prec_\theta$ be the order of increasing x-coordinate on P in orientation θ. If two points have the same x-coordinate, the one with higher y-coordinate is ahead of the other. For $X_\theta(P) = \{p_1, \ldots, p_k\} \subset P$, 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 $1 \leqslant i < k$ and we denote such a sequence of steps by *the staircase* $S_\theta(P)$ of P in orientation θ. 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 θ increases from 0 to 2π. 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 ε. To achieve our goal, we focus on when $X_\theta(P)$ changes. We call such 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*.

[1] In this paper, we deal only with orthogonal coordinate systems where two axes make the right angle.

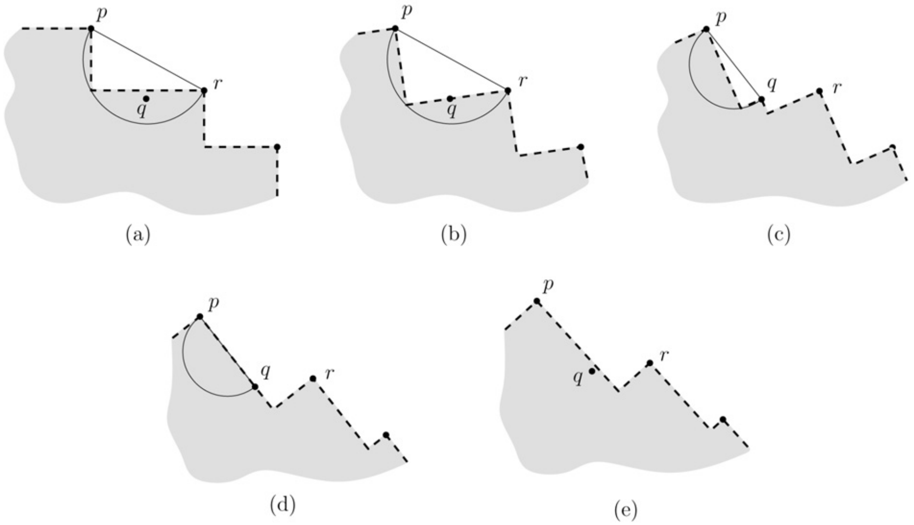

Fig. 2. Changes on $S_\theta(P)$ as θ increases; (a) q lies in $D(p,r)$. (b) $s^h_\theta(p,r)$ hits q, an in-event occurs, and q rises up. (c) $S_\theta(P)$ contains q. (d) $s^h_\theta(p,q)$ degenerates to point q, an out-event occurs, and q is about to disappear. (e) After the out-event. All the other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

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 θ, a point $q \in P \setminus X_{\theta-\varepsilon}(P)$ appears to $X_\theta(P)$ and also to $S_\theta(P)$. We say that q appears between p and r in $S_\theta(P)$. Then, at orientation θ, 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 $s_\theta(p,r)$. Fig. 2 shows some changes on $S_\theta(P)$ as θ increases, including an in-event and an out-event.

Observation 1. *When an event occurs at orientation θ, 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 $\prec_\theta$. We denote by $D(p,r)$ the half-disk with diameter pr that contains the triangle defined by $s_\theta(p,r)$ and pr as in Fig. 2(a). The corner of $s_\theta(p,r)$ goes along the circular arc of $D(p,r)$ counter-clockwise as θ increases, since $s_\theta(p,r)$ always makes the right angle at the corner. We denote two segments of $s_\theta(p,r)$ by $s^h_\theta(p,r)$ and $s^v_\theta(p,r)$, horizontal and vertical ones. Observe that as θ increases $s^h_\theta(p,r)$ 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. This corresponds to a new extremal point. Similarly, when $s^h_\theta(p,q)$ degenerates to q we have an out-event and q will disappear from the staircase. See Fig. 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 at another point q that is first encountered by $s^h_\theta(p,r)$ among points in $D(p,r)$ as θ increases. The out-event of q occurs when $s^h_\theta(p,q)$ 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 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 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 θ is at least 2π:

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

Proof. We claim that if a point $q \in X_\theta(P)$ is about to disappear at orientation θ, 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 θ. 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 θ there is a point $p \in P$ directly above q since q will disappear after θ (see Fig. 2(d)).

Note that maintaining the staircase is invariant under rigid motions. Also, increasing θ 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 θ, 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. □

Finally, we conclude one of our main results.

Theorem 1. *The staircase has at most $O(n)$ combinatorial changes while rotating the orientation by 2π radians. These changes can be maintained in $O(n^2)$ time and $O(n)$ space.*

Proof. By Lemma 1, we have at most $O(n)$ events during the algorithm. Since each event is associated with exactly one combinatorial change in the staircase while rotating the orientation, we have at most $O(n)$ such changes. Also, it suffices to handle and predict each event in linear time. Hence, our algorithm reveals all such changes in $O(n^2)$ time and $O(n)$ space. □

2.1. Predicting the next 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 the next in-event. Note that predicting an out-event can be preformed in constant time. Here, we present how to predict the next in-event in sublinear time so that the total running time of the algorithm reduces to subquadratic, and moreover a time/space trade-off 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 $s^h_\theta(p,q)$. 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 2. *(See Matoušek [5].) Let P be an n-point set in $\mathbb{R}^d$ and let m be a parameter, $n \leqslant m \leqslant n^d$. The range searching problem with the ranges being intersections of p half-spaces, $1 \leqslant p \leqslant d+1$, 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 on P. This structure enables us to report all the points in the queried lune shape $L_\theta(p,q)$. What we need here, however, 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 a secondary data structure. Note that the range searching structure $\mathcal{R}_m(P)$ is based on a partition tree [1,4,5]. Thus, with each internal node v, we store its canonical subset $P_v \subseteq P$ in a separate structure $\mathcal{C}(P_v)$ and associate that structure with v. The structure $\mathcal{C}(P_v)$ is supposed to find the point $r \in P_v$ which we first encounter during the rotation of the line supporting $s^h_\theta(p,q)$, counter-clockwise around q. Such a point r must be in convex position and indeed be the contact point of a tangent line to the convex hull of P_v through q. Hence, once we have computed the convex hull of P_v, the point r can be found in time $O(\log|P_v|) = O(\log n)$.

To process 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. We need an extra factor of $O(\log m) = O(\log n)$ for preprocessing and storage, since $\mathcal{R}_m(P)$ has depth at most $O(\log m)$. Thus, we can conclude the following time/space trade-off result.

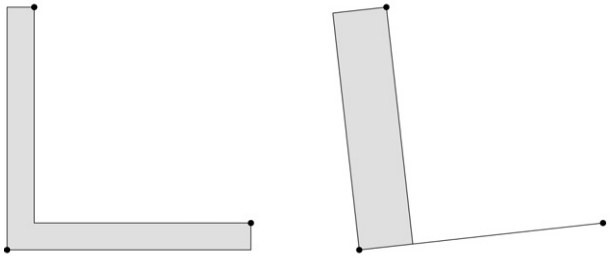

Fig. 3. Three points $P = \{(1, \tan\epsilon), (\tan\epsilon, 1), (0, 0)\}$ with $0 < \epsilon < \pi/12$ and enclosing L-shapes $\mathcal{L}_0(P)$ and $\mathcal{L}_\epsilon(P)$ in orientation 0 and ϵ.

Theorem 3. *While rotating the orientation by 2π radians, the staircase of n points can be maintained in time*

$$O\left(n^{1+\delta}\log n + m(\log n)^{1+\delta} + \frac{n^2}{m^{1/3}}\log^{7/3}\frac{m}{n}\right)$$

in space $O(m\log n)$, where m is a parameter with $n \leqslant m \leqslant n^2$.

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. 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 minimum area enclosing L-shape over all orientations by testing a small number of candidate orientations. However, L-shapes do not have such a property.

Lemma 2. *There exists a set P of points such that no orientations parallel or perpendicular to a line through a pair of points in P are optimal for the area of L-shapes.*

Proof. See Fig. 3. Let $\epsilon < \pi/12$ be a positive constant, and P be a set of three points $P = \{(1, \tan\epsilon), (\tan\epsilon, 1), (0, 0)\}$. It is not difficult to check that 0 is the only optimal orientation, which implies the lemma. □

Instead, we observe the following.

Observation 3. *In any orientation, the minimum area 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 by at most six touching 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_θ be the minimum bounding box of P at orientation θ and E_θ be the largest empty rectangle which shares the upper right corner with R_θ, both parallel to θ. Then the minimum area enclosing L-shape $\mathcal{L}_\theta(P)$ is represented by $\mathcal{L}_\theta(P) = R_\theta - E_\theta$. Let $p_1, \ldots, p_6$ 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_θ as shown in Fig. 4. Now, we assume that the polygon formed by the p_i with $1 \leqslant i \leqslant 6$ represent the minimum area L-shape $\mathcal{L}_\theta(P)$ during orientation $0 \leqslant \alpha < \theta < \beta < 2\pi$; 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 3. *One can minimize the area of $\mathcal{L}_\theta(P)$ over $0 \leqslant \alpha \leqslant \theta \leqslant \beta < 2\pi$, where the sequence of six touching points remains the same in orientation $\theta \in (\alpha, \beta)$.*

Proof. We use all the symbols as defined above. Furthermore, we need more terms. Define

$$d_1 := \mathrm{length}(p_1p_3), \quad d_2 := \mathrm{length}(p_2p_4), \quad c_1 := \mathrm{length}(p_1p_5), \quad \text{and} \quad c_2 := \mathrm{length}(p_4p_6).$$

Let $q_1, \ldots, q_6$ 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 α, we consider following angles as seen in Fig. 4:

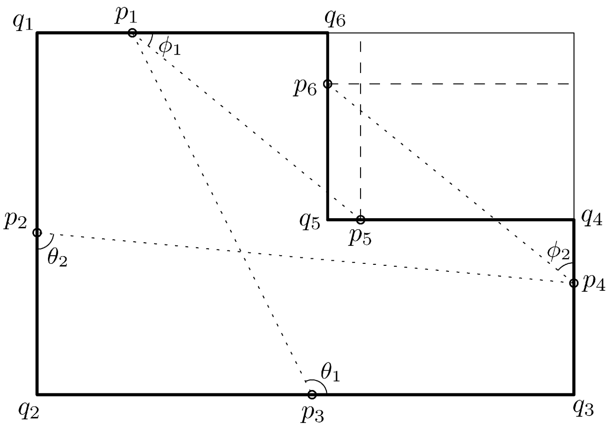

Fig. 4. The minimum area L-shape $\mathcal{L}_\alpha(P)$ of P in orientation α with six defining points $p_1, \ldots, p_6$ and six vertices $q_1, \ldots, q_6$.

$$\theta_1 := \angle p_1 p_3 q_3, \qquad \theta_2 := \angle p_4 p_2 q_2,$$
$$\phi_1 := \angle p_5 p_1 q_6, \qquad \phi_2 := \angle p_6 p_4 q_4.$$

We then have $\text{area}(R_\alpha) = d_1 d_2 \sin\theta_1 \sin\theta_2$, $\text{area}(E_\alpha) = c_1 c_2 \sin\phi_1 \sin\phi_2$, and

$$\text{area}(R_{\alpha+\delta}) = d_1 d_2 \sin(\theta_1 - \delta)\sin(\theta_2 - \delta),$$
$$\text{area}(E_{\alpha+\delta}) = c_1 c_2 \sin(\phi_1 + \delta)\sin(\phi_2 - \delta),$$

where $0 \leqslant \delta \leqslant \beta - \alpha$. Thus, we can express the area of $\mathcal{L}_\theta(P)$ as a sinusoidal function A_L of δ: $A_L(\delta) := \text{area}(\mathcal{L}_{\alpha+\delta}(P)) = \text{area}(R_{\alpha+\delta}) - \text{area}(E_{\alpha+\delta})$. 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

$$\begin{aligned} A'_L(\delta) &= d_1 d_2\big(-\cos(\theta_1 - \delta)\sin(\theta_2 - \delta) - \sin(\theta_1 - \delta)\cos(\theta_2 - \delta)\big) \\ &\quad - c_1 c_2\big(\cos(\phi_1 + \delta)\sin(\phi_2 - \delta) - \sin(\phi_1 + \delta)\cos(\phi_2 - \delta)\big) \\ &= -d_1 d_2 \sin(\theta_1 + \theta_2 - 2\delta) - c_1 c_2 \sin(-\phi_1 + \phi_2 - 2\delta) \\ &= d_1 d_2 \sin(2\delta - \theta_1 - \theta_2) + c_1 c_2 \sin(2\delta + \phi_1 - \phi_2). \end{aligned}$$

Solving $A'_L(\delta) = 0$, we have

$$\delta = \frac{\theta_1 + \theta_2}{2} - \frac{1}{2}\arctan\left(\frac{c_1 c_2 \sin(\theta_1 + \theta_2 + \phi_1 - \phi_2)}{d_1 d_2 + c_1 c_2 \cos(\theta_1 + \theta_2 + \phi_1 - \phi_2)}\right).$$

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 of A_L. A local optimal orientation can be searched among those orientations. □

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

Theorem 4. *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 $0 \leqslant \theta < 2\pi$, in $O(n^2)$ time and $O(n)$ space.*

Proof. In this proof, we present an algorithm for computing optimal orientations for L-shapes. As a subroutine, it makes use of the algorithm maintaining the staircase presented in Section 2. Also, we make use of the method described in the proof of Lemma 3. By Lemma 3, if we can fix one point on each side of $\mathcal{L}_\theta(P)$ during $\alpha \leqslant \theta \leqslant \beta$ for some $0 \leqslant \alpha \leqslant \beta \leqslant 2\pi$, then we are able to decide the local optimum in the domain (α, β).

In addition we need to handle another type of events that occur when the current orientation θ is parallel or perpendicular to an edge of $\text{conv}(P)$; at this moment, one of the four points defining the smallest bounding box R_θ is being changed. 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 α_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_θ 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_θ is determined by one step of $X_\theta(P)$. Thus, we minimize every L-shape determined by each 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. □

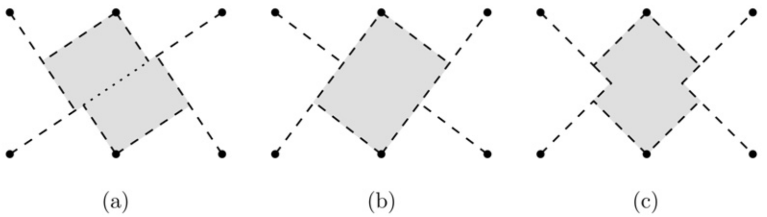

Fig. 5. A set P of 6 points and its rectilinear convex hulls in three orientations.

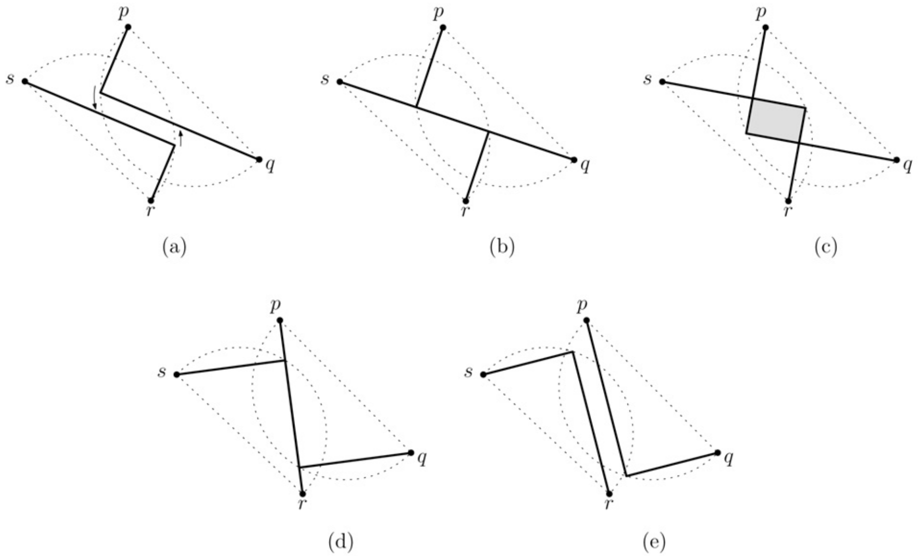

Fig. 6. (a) Two steps in opposite staircases. (b) A step-cross event occurs. (c) The shaded region is the overlap. (d) A step-release event occurs. (e) Two steps are separated.

4. Minimum-area rectilinear convex hulls

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

As in the L-shape problem, there seems no hope to restrict the number of candidate orientations for the minimum-area rectilinear convex hull to a small number as shown in the following lemma.

Lemma 4. *There exists a set P of points such that no orientations parallel or perpendicular to a line through a pair of points in P are optimal for the area of rectilinear convex hulls.*

Proof. Let P be a set of six points; $P = \{(-3,-2), (0,-2), (3,-2), (-3,2), (0,2), (3,2)\}$. Among orientations from the pairs of points in P are 0, $\tan^{-1}\frac{2}{3}$ and $\tan^{-1}\frac{4}{3}$. $\mathcal{RH}_0(P)$ coincides with $\operatorname{conv}(P)$ and its area is 24. If we take $\theta = \tan^{-1}\frac{2}{3}$, $\mathcal{RH}_\theta(P)$ is depicted in Fig. 5(a). Its area is calculated as $\operatorname{area}(\mathcal{RH}_\theta(P)) = 8 + \frac{4}{13}$. If $\theta' = \tan^{-1}\frac{4}{3}$, $\mathcal{RH}_{\theta'}(P)$ looks like Fig. 5(b) and its area is $\operatorname{area}(\mathcal{RH}_{\theta'}) = 7 + \frac{17}{25}$. However, the area of $\mathcal{RH}_{\pi/4}(P)$ as in Fig. 5(c) is only 7. This orientation $\pi/4$ is neither parallel nor perpendicular to any line through a pair of points in P. □

We thus devise a solution to the rectilinear convex hull in a similar way as we did in L-shapes. Furthermore, we have another difficulty in calculating the area of $\mathcal{RH}_\theta(P)$; as seen in Fig. 6(c), two staircases among those in $\mathcal{S}_\theta(P)$ may cross each other.

Observation 4. *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.*

In order to deal with such overlaps between staircases, we refine the definition of combinatorially equivalent rectilinear convex hulls since the shape of $\mathcal{RH}_\theta(P)$ may change regardless of the equivalence of $\mathcal{S}_\theta(P)$. Thus we also take how $\mathcal{S}_\theta(P)$ crosses 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 staircase opposite 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. Now, we explain how to handle events.

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

Step-cross event We compute the step-release event of the involved pair of steps and insert it into $\mathcal{Q}$.

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

The following lemma counts the total number of events.

Lemma 5. *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 at most $O(n)$ step-cross events, which implies the same number of step-release events, and thus at last $O(n)$ events in total.

In this proof, we count the number of possible step-cross events only between $S_\theta(P)$ and $S_{\pi+\theta}(P)$, then the other case being symmetric. If we have such a step-cross event at θ between $S_\theta(P)$ and $S_{\pi+\theta}(P)$, then the two touching horizontal segments of the involved steps are aligned on a horizontal line segment as in Fig. 6(b). Note that the endpoints of the line segment are extremal points of the involved steps. We call such a segment *a step-cross segment*. Note that such a step-cross segment induced by a step-cross event between $S_\theta(P)$ and $S_{\pi+\theta}(P)$ is parallel to θ.

Consider a graph G with vertex set P where there is an edge between $p \in P$ and $q \in P$ if p and q define a step-cross segment between $S_\theta(P)$ and $S_{\pi+\theta}(P)$ for $\theta \in [0, \pi/2)$. We claim that no two step-cross segments cross each other, which implies that G has $O(n)$ edges and thus there are at most $O(n)$ step-cross segments, including those between $S_{\pi/2+\theta}(P)$ and $S_{3\pi/2+\theta}(P)$ by symmetry.

Assume to the contrary that G contains two distinct step-cross segments ab and pq in orientations θ_1 and θ_2, respectively, with $\theta_1 < \theta_2$ such that they cross each other. Note that $\theta_2 - \theta_1 < \pi/2$ since $0 \leqslant \theta_1 < \theta_2 < \pi/2$, and we let $s_{\theta_1}(c, b)$, $s_{\theta_1}(d, a)$, $s_{\theta_2}(r, p)$, and $s_{\theta_2}(s, q)$ be the corresponding four steps defining ab and pq, for some $c, d, r, s \in P$.

By definition, the quadrants supporting $s_{\theta_2}(r, p)$ and $s_{\theta_2}(s, q)$ are free from P and their horizontal segments overlap along pq. Since ab and pq cross, a lies below and b lies above the line supporting pq, and a lies strictly to the left of b with respect to orientation θ_2 (Fig. 7). To make the quadrants supporting the steps free to P, r must be to the left of a and s must be to the right of b with respect to orientation θ_2. This contradicts that the horizontal segments of $s_{\theta_2}(r, p)$ and $s_{\theta_2}(s, q)$ overlap and we have a step-cross event at θ_2 by the two steps. □

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.

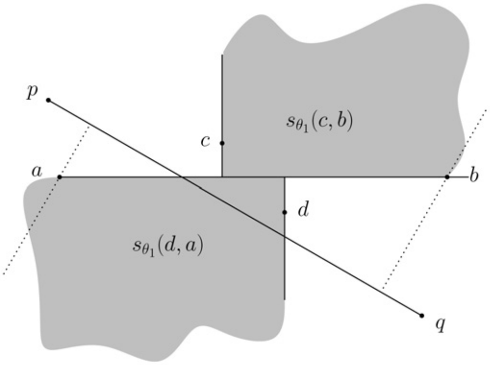

Fig. 7. Proof of Lemma 5.

4.1. Finding local optima of $\mathcal{RH}_\theta(P)$

Here, we assume that an orientation interval (α, β) does not contain any event orientation and the current orientation θ runs in between (α, β). Thus, we can also say that $\mathcal{S}_\theta(P)$ 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 θ), and p_i be the i-th point in $\mathcal{S}_\theta(P)$ in clockwise order. We get a polygon $\mathcal{P}$ by a sequence of m sides $p_i p_{i+1}$ with $1 \leqslant i \leqslant m-1$, 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 orientation in $\theta \in (\alpha, \beta)$, these extremal points and $\mathcal{P}$ do not change in the interval. Also, we denote by $s_i(\theta)$ the step $s_\theta(p_i, p_{i+1})$ of p_i and p_{i+1} at θ, and by $\triangle_i(\theta)$ the right triangle defined by $s_i(\theta)$ as in Section 2.

Let $\square_j(\theta)$ be the rectangle defined by each overlap in $\mathcal{S}_\theta(P)$, where $1 \leqslant j \leqslant k$. Then, the area of $\mathcal{RH}_\theta(P)$ is

$$\operatorname{area}\bigl(\mathcal{RH}_\theta(P)\bigr) = \operatorname{area}(\mathcal{P}) - \sum_i \operatorname{area}\bigl(\triangle_i(\theta)\bigr) + \sum_j \operatorname{area}\bigl(\square_j(\theta)\bigr).$$

To minimize the area of $\mathcal{RH}_\theta(P)$ over $\theta \in (\alpha, \beta)$, we analyze the functions $\operatorname{area}(\triangle_i(\theta))$ and $\operatorname{area}(\square_j(\theta))$, and obtain a nice representation so that we can get an optimum in an analytic way. Observe that no two overlapping regions can intersect each other.

Lemma 6. *Let (α, β) with $0 \leqslant \alpha < \beta < \pi/2$ be an orientation interval such that it contains no event orientation. Once we have computed $\mathcal{S}_\alpha(P)$ and $\mathcal{RH}_\alpha(P)$, it is possible in linear time to compute a local optimal orientation in which the area of $\mathcal{RH}_\theta(P)$ is minimized over $\theta \in (\alpha, \beta)$.*

Proof. Let d_i be the length of the hypotenuse of $\triangle_i(\alpha)$ and ϕ_i be the internal angle of $\triangle_i(\alpha)$ at p_i. Then,

$$\operatorname{area}\bigl(\triangle_i(\theta)\bigr) = d_i^2 \cos\bigl(\phi_i + (\theta - \alpha)\bigr) \sin\bigl(\phi_i + (\theta - \alpha)\bigr).$$

Substituting $\theta - \alpha$ by φ, we get $\operatorname{area}(\triangle_i(\theta)) = d_i^2 \cos(\phi_i + \varphi)\sin(\phi_i + \varphi) = \frac{1}{2} d_i^2 \sin 2(\phi_i + \varphi) = \frac{1}{2} d_i^2 \sin 2\phi_i \cos 2\varphi + \frac{1}{2} d_i^2 \cos 2\phi_i \sin 2\varphi$, which is linear in $\cos 2\varphi$ and $\sin 2\varphi$. Therefore, the sum over all i is of the same form:

$$\sum_i \operatorname{area}\bigl(\triangle_i(\theta)\bigr) = \frac{1}{2}\Bigl(\sum_i d_i^2 \sin 2\phi_i\Bigr)\cos 2\varphi + \frac{1}{2}\Bigl(\sum_i d_i^2 \cos 2\phi_i\Bigr)\sin 2\varphi.$$

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 $S_{\pi+\theta}(P)$. Denoting the coordinate of a point p in orientation α by (x_p, y_p),

$$\operatorname{area}\bigl(\square_j(\alpha)\bigr) = |x_{p_a} - x_{p_b}| \times |y_{p_{a+1}} - y_{p_{b+1}}|,$$

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 α. Also, the area of $\square_j(\theta)$ is invariant under translations. Thus, $\operatorname{area}(\square_j(\theta)) = \operatorname{area}(\square_j(\alpha + \varphi))$ $= |x_{p'_a} - x_{p'_b}| \times |y_{p'_{a+1}} - y_{p'_{b+1}}| = |(x_{p_a} - x_{p_b})\cos\varphi + (y_{p_a} - y_{p_b})\sin\varphi| \times |(y_{p_{a+1}} - y_{p_{b+1}})\cos\varphi - (x_{p_{a+1}} - x_{p_{b+1}})\sin\varphi| =$ $C_1 \cos^2\varphi + C_2 \sin^2\varphi + C_3 \sin\varphi\cos\varphi$, 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 depending on the coordinates and α. This equation can be rewritten to be linear in $\sin 2\varphi$ and $\cos 2\varphi$ again as follows:

$$\operatorname{area}\bigl(\square_j(\theta)\bigr) = C'_1 + C'_2 \cos 2\varphi + C'_3 \sin 2\varphi,$$

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 $\mathcal{RH}_\theta(P)$ with $\alpha < \theta < \beta$ can be represented by a function f of φ 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 $\mathcal{RH}_\theta(P)$ with $\theta \in (\alpha, \beta)$. All this process can be done in linear time. □

Now, we conclude the following theorem.

Theorem 5. *Given a set P of n points, one can decide an optimal orientation, which minimizes the area of the rectilinear convex hull $\mathcal{RH}_\theta(P)$ over $0 \leqslant \theta < 2\pi$, in $O(n^2)$ time and $O(n)$ space.*

5. Concluding remarks

We presented algorithms for computing the minimum area enclosing L-shape and rectilinear convex hull of given points for arbitrary orientation. The algorithms run in quadratic time, while the problems can be solved in $O(n \log n)$ time for a fixed orientation. We must ask about their lower bounds: Computing the staircase for a fixed orientation can be reduced to sorting numbers, and so is computing the rectilinear convex hull. But the lower bound for the L-shape is not known except for trivial bounds. Also, the upper bounds could be improved to subquadratic. However, for the minimum area rectilinear convex hull, the objective function consists of a linear number of terms in worst cases; it seems not so easy to improve the upper bound.

Our solutions for finding minimum area 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 studied 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 some geometric optimization problems whose solutions select an optimal orientation. Obviously, there are many data structures and optimization problems whose solutions are variant to orientations, and we expect that many such optimization problems could be efficiently solved by maintaining a certain data structure over all orientations.

References

[1] P.K. Agarwal, J. Erickson, Geometric range searching and its relatives, in: B. Chazelle, J.E. Goodman, R. Pollack (Eds.), Advances in Discrete and Computational Geometry, in: Contemporary Mathematics, vol. 233, American Mathematical Society Press, Providence, 1999, pp. 1–56.

[2] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, Computational Geometry: Algorithms and Applications, Springer, 2000.

[3] R.G. Karlsson, M. Overmars, Scanline algorithms on a grid, BIT Numer. Math. 28 (2) (1988) 227–241.

[4] J. Matoušek, Efficient partition trees, Discrete Comput. Geom. 8 (1992) 315–334.

[5] J. Matoušek, Range searching with efficient hierarchical cuttings, Discrete Comput. Geom. 10 (1993) 157–182.

[6] J. Matoušek, P. Plecháč, On functional separately convex hulls, Discrete Comput. Geom. 19 (1998) 105–130.

[7] D.Y. Montuno, A. Fournier, Finding the $x-y$ convex hull of a set of $x-y$ polygons, Technical Report 148, University of Toronto.

[8] T.M. Nicholl, D.T. Lee, Y.Z. Liao, C.K. Wong, On the $X-Y$ convex hull of a set of $X-Y$ polygons, BIT Numer. Math. 23 (4) (1983) 456–471.

[9] T. Ottman, E. Soisalon-Soisinen, D. Wood, On the definition and computation of rectilinear convex hulls, Information Sciences 33 (1984) 157–171.

[10] F.P. Preparata, S.J. Hong, Convex hulls of finite sets of points in two and three dimensions, Comm. ACM 20 (2) (1977) 87–93.

[11] F.P. Preparata, M.I. Shamos, Computational Geometry: An Introduction, Springer, 1985.

[12] O. Schwarzkopf, U. Fuchs, G. Rote, E. Welzl, Approximation of convex figures by pairs of rectangles, in: Proc. 7th Ann. Symp. on Theoretical Aspects of Computer Science, 1990, pp. 240–249.

[13] E. Welzl, Smallest enclosing disks (balls and ellipsoids), in: New Results and New Trends in Computer Science, in: LNCS, vol. 555, Springer, 1991, pp. 359–370.