International Journal of Computational Geometry & Applications

PLANAR STRONG VISIBILITY

EUGENE FINK
Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
eugene@csee.usf.edu
www.csee.usf.edu/∼eugene

DERICK WOOD
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
dwood@cs.ust.hk
www.cs.ust.hk/∼dwood

Strong visibility is a generalization of standard visibility, defined with respect to a fixed set of line orientations. We investigate computational properties of this generalized visibility, as well as the related notion of strong convexity, and describe algorithms for the following tasks:

1. Testing the strong visibility of two points in a polygon.
2. Finding the strong convex hull of a point set or polygon.
3. Constructing the strong kernel of a polygon.
4. Identifying the points that are strongly visible from a given point.

Keywords: Generalized visibility; convexity; restricted orientations.

1. Introduction

The study of nontraditional notions of visibility and convexity is a fruitful branch of geometry, which has found applications in VLSI design, graphics, and motion planning. Researchers have investigated multiple variations of these notions, including orthogonal,[15,16,17,18] finitely oriented,[8,24,28] and restricted-orientation[20,24,25] visibility and convexity, as well as NESW[12,26,28] and link[1,25,27] convexity.

Rawlins introduced the notions of strong visibility and convexity in his doctoral dissertation,[20] as part of his research on restricted-orientation geometry. Rawlins and Wood[23,24] studied strongly convex sets, and demonstrated that they generalize not only standard convexity, but also the notions of ortho-rectangles and C-oriented polygons.[7,9] We extended restricted-orientation geometry to higher dimensions and reported the results in a series of articles.[2,3,4]

Martynchik, Metelski, Rawlins, Schuierer, and Wood explored computational problems in restricted-orientation geometry and developed a suite of related

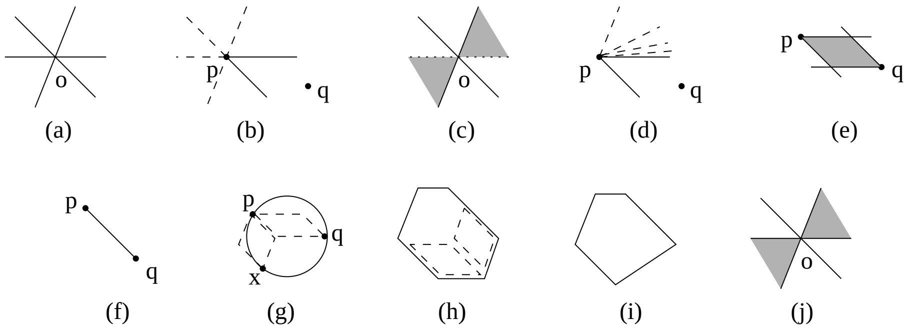

Fig. 1. Strong visibility and convexity.

algorithms.[13,21,22,25] Although Rawlins pointed out similar problems in strong visibility,[20] he has not pursued this research direction. We now develop methods for solving these problems; most algorithms are based on reduction to similar problems in standard visibility.

We begin with the definition of strong visibility and convexity (Section 2), and then consider five computational tasks. The first two tasks are testing strong visibility of two points in a polygon and strong convexity of a polygon (Section 3). The next two problems are finding the strong hull of a point set (Section 4) and strong kernel of a polygon (Section 5). Finally, we describe a technique for identifying all points visible from a given spot (Section 6).

2. Strong Visibility and Convexity

Traditionally, two points in a set are considered *visible* to each other if the straight segment joining them is completely in the set. If every two points of a set are visible to each other, then the set in *convex.* We define strong visibility by replacing straight segments with a different type of objects, called *$\mathcal{O}$-blocks,* and then define strong convexity in terms of this new visibility.

We first introduce the notion of an *orientation set* $\mathcal{O}$*,* which is a set of lines through some fixed point o (Figure 1a); we assume that it includes at least two distinct lines. A straight line parallel to one of the lines of $\mathcal{O}$ is called an $\mathcal{O}$-oriented line; for example, the dashed lines in Figures 1(g) and 1(h) are $\mathcal{O}$-oriented.

To construct the $\mathcal{O}$-block of two points, say p and q, we draw all $\mathcal{O}$-oriented rays with endpoint p and choose the two of them closest to q (Figure 1b). The two selected rays, with the common endpoint p, are the boundary of an angle with vertex p, which contains q.

If $\mathcal{O}$ is an infinite set, it may not be closed, and we may not be able to choose the ray closest to q. For instance, the orientation set in Figure 1(c) is not closed; all lines in the shaded area are elements of $\mathcal{O}$, whereas the dotted horizontal line is not in $\mathcal{O}$. If $\mathcal{O}$ is not closed, we choose two rays with common endpoint p such that, for each of the two selected rays, (1) there is a sequence of $\mathcal{O}$-oriented rays convergent to this ray and (2) there are no $\mathcal{O}$-oriented rays with endpoint p between this ray and the point q (Figure 1d). The two selected rays are again the boundary of an angle with vertex p.

Similarly, we draw the $\mathcal{O}$-oriented rays from q closest to p and obtain an angle with vertex q (Figure 1e). The *$\mathcal{O}$-block* of p and q is the intersection of the two angles, shown by the shaded parallelogram in Figure 1(e). As a special case, if the line through p and q is $\mathcal{O}$-oriented, the $\mathcal{O}$-block of p and q is the straight segment joining p and q (Figure 1f).

Two points of a set are *strongly visible* to each other if their $\mathcal{O}$-block is contained in the set. For example, the points p and q in Figure 1(g) are strongly visible to each other for the orientation set in Figure 1(a), whereas the points p and x are not; we show the corresponding $\mathcal{O}$-blocks by dashed lines. Observe that, if two points are strongly visible to each other, then they are also standardly visible. Also note that every point is strongly visible to itself.

A set is *strongly convex* if every two of its points are strongly visible to each other. For example, the circle in Figure 1(g) is not strongly convex for the orientation set in Figure 1(a) since some of its points are not strongly visible to each other. On the other hand, the polygon in Figure 1(h) is strongly convex; we show two $\mathcal{O}$-blocks contained in it. As another example, the polygon in Figure 1(i) is strongly convex for the orientation set in Figure 1(c), but not for the set in Figure 1(a). The following properties of strong convexity readily follow from the definition.

Lemma 1.

1. *Every translate of a strongly convex set is strongly convex.*
2. *The intersection of strongly convex sets is strongly convex.*
3. *Every strongly convex set is standard convex.*

We next observe that, if $\mathcal{O}$ is not closed, and $\mathcal{O}_{\text{cl}}$ is the closure of $\mathcal{O}$, then the $\mathcal{O}$-block of two points is identical to their $\mathcal{O}_{\text{cl}}$-block. In other words, the strong visibility with respect to $\mathcal{O}$ is the same as the strong visibility with respect to $\mathcal{O}_{\text{cl}}$. As an example, the visibility for the set in Figure 1(c) is equivalent to that for the set in Figure 1(j). This observation means that *we may restrict attention to the study of strong visibility and convexity for closed orientation sets.*

We assume that $\mathcal{O}$ consists of a finite number of disjoint closed ranges; for instance, the set in Figure 1(j) comprises two ranges, one of which includes only one line. We also assume that the ranges are in *sorted order,* which allows binary search for a range that contains a given line. We have not used these assumptions in the study of mathematical properties of strong convexity,[4] but they are essential for the investigation of computational properties.

3. Visibility and Convexity Test

The first problem is determining whether two points of a polygon are strongly visible to each other. The polygon may not be simply connected; that is, it may

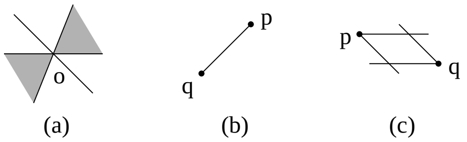

Fig. 2. Constructing the $\mathcal{O}$-block of two given points.

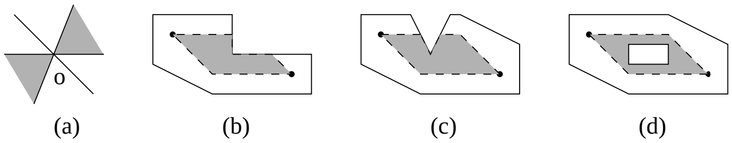

Fig. 3. Three cases when points are not strongly visible to each other.

have “holes.” To test the visibility of two points, we construct their $\mathcal{O}$-block and verify its containment in the polygon.

If the line through the two points belongs to a range of $\mathcal{O}$-orientations, their $\mathcal{O}$-block is the straight segment (Figure 2b). Otherwise, we find the two closest $\mathcal{O}$-orientations and use them to construct the $\mathcal{O}$-block (Figure 2c). If the orientation set $\mathcal{O}$ contains m ranges, the construction of the $\mathcal{O}$-block of two given points takes $O(\lg m)$ time because we need to find the two range boundaries closest to the line through these points. We use the following result to test whether the $\mathcal{O}$-block is inside a given polygon.

Lemma 2. *The $\mathcal{O}$-block of two points is wholly in a polygon if and only if the following three conditions hold:*

1. *The vertices of the $\mathcal{O}$-block are in the polygon.*
2. *Every two adjacent vertices of the $\mathcal{O}$-block are standardly visible to each other.*
3. *The polygon does not have holes inside the $\mathcal{O}$-block.*

Proof. We illustrate the violation of each condition in Figure 3. Clearly, if some condition does not hold, then the $\mathcal{O}$-block is not contained in the polygon. On the other hand, if the conditions hold, then the polygon's boundary does not intersect the $\mathcal{O}$-block's interior, which implies that the $\mathcal{O}$-block is wholly in the polygon. □

If the polygon has n vertices, and the $\mathcal{O}$-block of two given points is a parallelogram, then the test for each condition of Lemma 2 takes $O(n)$ time. If the $\mathcal{O}$-block of the given points is a straight segment, then their strong visibility is equivalent to standard visibility, which can also be verified in $O(n)$ time. Thus, the following result holds in both cases.

Theorem 3. *If the orientation set $\mathcal{O}$ is a sorted collection of m disjoint ranges, then the time of testing strong visibility of two points in an n-vertex polygon is $O(n + \lg m)$.*

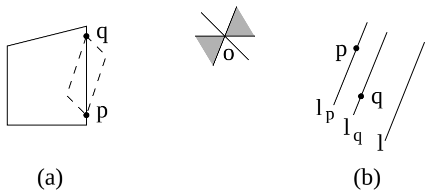

Fig. 4. Proof of Lemma 4.

Next, we describe an algorithm for verifying strong convexity of a given polygon based on the following result.

Lemma 4. *A polygon is strongly convex if and only if it is convex and its edges are $\mathcal{O}$-oriented.*

Proof. Clearly, if a polygon is not convex, then it is not strongly convex. If some edge is not $\mathcal{O}$-oriented, then, for any two distinct points p and q of this edge, the $\mathcal{O}$-block of p and q is not in the polygon (Figure 4a), which implies that the polygon is not strongly convex.

If a polygon is convex and all its edges are $\mathcal{O}$-oriented, then it is the intersection of several halfplanes whose boundaries are $\mathcal{O}$-oriented. To prove that it is strongly convex, we demonstrate that each of these halfplanes is strongly convex. Specifically, we show that, for every halfplane whose boundary l is $\mathcal{O}$-oriented, and every two points p and q of the halfplane, the $\mathcal{O}$-block of p and q is wholly in the halfplane.

Let l_p be the line through p parallel to l, and l_q be the line through q parallel to l (Figure 4b). Since l_p and l_q are $\mathcal{O}$-oriented, the $\mathcal{O}$-block of p and q is contained in the "strip" between l_p and l_q; hence, this $\mathcal{O}$-block is in the halfplane.

Thus, the polygon is the intersection of several strongly convex halfplanes, which implies that it is strongly convex. □

Testing standard convexity of a polygon with n vertices takes $O(n)$ time. If the polygon proves convex, then the orientations of its edges are in sorted order, and we need to check that each element in the sorted list of edge orientations belongs to one of the $\mathcal{O}$-orientation ranges, where the ranges are also in sorted order.

If $m \leq n$, we can test whether an n-element sorted list is a subset of a union of m ranges in $O(n)$ time; if $m > n$, then an efficient test takes $O(n \cdot \lg \frac{n+m}{n})$ time. We may express the running time as $O(n + n \cdot \lg \frac{n+m}{n})$, which is equivalent to $O(n)$ for $m \leq n$ and to $O(n \cdot \lg \frac{n+m}{n})$ for $m > n$.

Theorem 5. *If $\mathcal{O}$ is a sorted collection of m disjoint ranges, then the time of testing strong convexity of an n-vertex polygon is $O(n + n \cdot \lg \frac{n+m}{n})$.*

4. Strong Convex Hull

The *convex hull* of a geometric object is the intersection of all convex sets containing the object; in other words, it is the minimal convex set that completely covers the

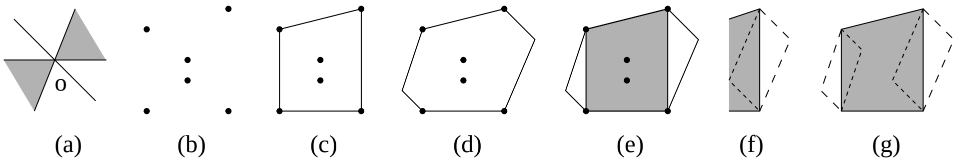

Fig. 5. Construction of the strong hull.

object. Similarly, the *strong convex hull* is the minimal strongly convex set that contains the object. For example, consider the orientations in Figure 5(a) and the set of six points in Figure 5(b). We show its standard convex hull in Figure 5(c) and its strong hull in Figure 5(d). The algorithm for computing the strong hull is based on the following observation.

Lemma 6. *The strong hull of a geometric object is identical to the strong hull of the standard hull of the object.*

Proof. Let P be a geometric object, Q be its standard hull, and *S-hull*(P) and *S-hull*(Q) be their strong hulls. For example, if P is the six-point set in Figure 5(e), then Q is the shaded polygon, and *S-hull*(P) is the outer hexagon. By the definition of convex hulls, P is a subset of Q, which implies that *S-hull*$(P) \subseteq$ *S-hull*(Q). On the other hand, since *S-hull*(P) is convex, and Q is the *minimal* convex set containing P, we have $Q \subseteq$ *S-hull*(P); hence, *S-hull*$(Q) \subseteq$ *S-hull*(P). These two opposite inclusions imply that *S-hull*$(Q) =$ *S-hull*(P). □

The first step of constructing the strong hull of an n-point set is finding its standard hull, which takes $O(n \cdot \lg n)$ time.[5] The resulting hull is a convex polygon with at most n vertices, and the next step is finding the strong hull of this polygon.

For every two adjacent vertices of the polygon, we determine their $\mathcal{O}$-block and identify the half of the $\mathcal{O}$-block located outside of the polygon, called the *outer halfblock.* This construction is illustrated in Figure 5(f), where the shaded region is the polygon's interior, the dashed lines show the outer halfblock of two adjacent vertices, and the dotted lines mark the other half of their $\mathcal{O}$-block. The concatenated boundaries of the outer halfblocks make the contour that bounds the strong hull of the polygon (Figure 5g).

The construction of every $\mathcal{O}$-block requires finding the two $\mathcal{O}$-orientations closest to the corresponding edge, which takes $O(\lg m)$ time. The overall time of finding the standard hull of the point set and constructing the $\mathcal{O}$-blocks is $O(n \cdot (\lg n + \lg m))$.

Observe that the orientations of the standard hull's edges are in sorted order, which allows reducing the time of finding the two closest $\mathcal{O}$-orientations for each of the edges; specifically, we can construct the $\mathcal{O}$-blocks in $O(n + n \cdot \lg \frac{n+m}{n})$ time. Unfortunately, it does not improve the overall time because adding $O(n \cdot \lg n)$ for computing the standard hull makes the total of $O(n \cdot (\lg n + \lg m))$.

The next observation implies that the same algorithm can compute the strong

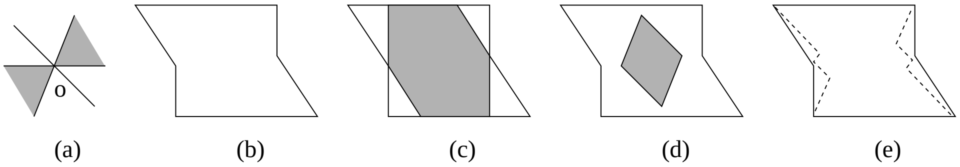

Fig. 6. Standard kernel, strong kernel, and inner halfblocks of a polygon.

hull of a polygon; thus, the time of finding a polygon's hull is also $O(n \cdot (\lg n + \lg m))$.

Lemma 7. *The strong hull of a polygon is identical to the strong hull of the point set formed by the polygon's vertices.*

Proof. Clearly, the standard hull of a polygon is identical to the standard hull of its vertices; for example, see *Convex Polytopes* by Grünbaum *et al.*[6] Furthermore, the strong hull of a point set is identical to the strong hull of the standard hull of the set (Lemma 6), which immediately implies the desired result. □

If a polygon is simple, we can find its standard convex hull in $O(n)$ time,[14] thus reducing the time of constructing the strong hull to $O(n + n \cdot \lg \frac{n+m}{n})$. We state the results on the strong-hull computation as a theorem.

Theorem 8. *Suppose that $\mathcal{O}$ is a sorted collection of m disjoint ranges.*

1. *The time of finding the strong hull of an n-point set or n-vertex polygon is $O(n \cdot (\lg n + \lg m))$.*
2. *The time of finding the strong hull of a simple polygon with n vertices is $O(n + n \cdot \lg \frac{n+m}{n})$.*

5. Strong Kernel

The *kernel* of a geometric object is the set of points that are visible from all points of the object. For example, the polygon in Figure 6(b) has a nonempty kernel, shown as a shaded region in Figure 6(c). Note that only simply connected objects may have nonempty kernels, and that an object is convex if and only if its kernel is identical to the object itself. The computation of the standard kernel of a polygon with n vertices takes $O(n)$ time.[11]

The *strong kernel* is analogous to the standard kernel; it is the set of all points that are strongly visible from all points of the object. In Figure 6(d), we show the strong kernel of the same polygon for the orientations in Figure 6(a). The problem of finding the strong kernel of a polygon is reducible to the standard-kernel computation. We derive the properties of the strong kernel used in the reduction and then describe the algorithm.

First, observe that a point of a polygon belongs to the standard kernel if and only if it is standardly visible from every *vertex;* for example, see the textbook by Preparata and Shamos.[19] The analogous result holds for strong visibility.

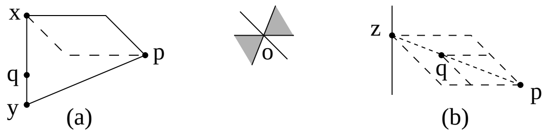

Fig. 7. Proof of Lemma 9.

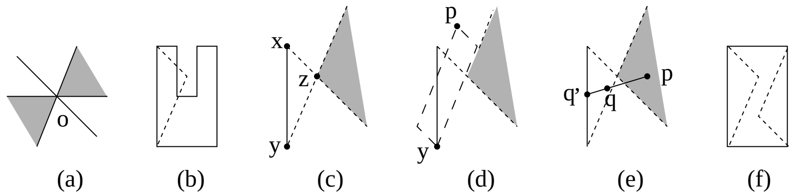

Fig. 8. Proof of Lemma 10.

Lemma 9. *A point of a polygon is in the strong kernel if and only if it is strongly visible from all vertices of the polygon.*

Proof. If a point is in the strong kernel of a polygon, then by definition it is strongly visible from all points of the polygon, which implies that it is strongly visible from all vertices.

To prove the converse, we consider a point p that is strongly visible from all vertices, and show that p is strongly visible from every point q of the polygon. First, suppose that q is on some edge, whose endpoints are denoted x and y (Figure 7a). Note that the $\mathcal{O}$-block of p and x is in the polygon, and the $\mathcal{O}$-block of p and y is also in the polygon; therefore, the contour formed by these $\mathcal{O}$-blocks and the edge, which is marked by solid lines in Figure 7(a), is completely in the polygon. Clearly, the $\mathcal{O}$-block of p and q is contained inside this contour, which implies that p is strongly visible from q.

Now suppose that q is in the polygon's interior. We draw the line from p to q, extend it to the intersection with the boundary, and consider a point z of the intersection (Figure 7b). Since p is strongly visible from z, and the $\mathcal{O}$-block of p and q is inside the $\mathcal{O}$-block of p and z, we conclude that p is strongly visible from q. □

The first step of constructing the strong kernel is finding the inner halfblocks of a polygon, which are analogous to the outer halfblocks (Figure 5f); the *inner halfblock* of two adjacent vertices is the half of their $\mathcal{O}$-block opposite to the outer halfblock. We show the inner halfblocks of the polygon in Figure 5(g) by dotted lines. Note that an inner halfblock may not be completely in the polygon, as shown in Figure 8(b).

Lemma 10. *If a polygon has a pair of adjacent vertices whose inner halfblock is not in the polygon, then the polygon's strong kernel is empty.*

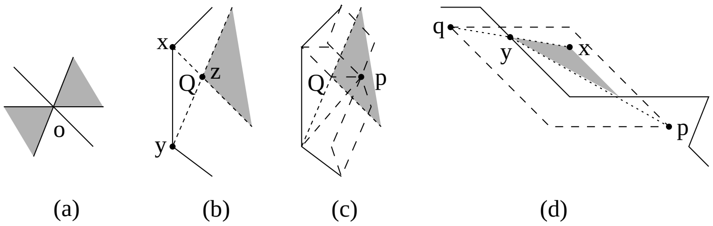

Fig. 9. Proofs of Lemma 11 and Lemma 12.

Proof. Let x and y be adjacent vertices, whose inner halfblock is not completely in the polygon, and z be the third vertex of their halfblock. For convenience, we assume without loss of generality that the edge between x and y is vertical, x is above y, and z is on the right side of the edge (Figure 8c). Suppose that the kernel of the polygon is *nonempty,* and p is one of its points. Note that p cannot be above the line through y and z because then the $\mathcal{O}$-block of p and y would not be completely in the polygon (Figure 8d). Similarly, p cannot be below the line through x and z, which implies that p is in the shaded angle.

We now pick a point q that is in the halfblock and not in the polygon, and define a point q' as the intersection of the line through p and q with the edge between x and y (Figure 8e). Then, p is not strongly visible from q', which leads to a contradiction. □

The converse of Lemma 10 does not hold; that is, a polygon's kernel may be empty even if every inner halfblock is completely in the polygon. For example, the kernel of the polygon in Figure 8(f) is empty.

Lemma 11. *Suppose that P_1 is a polygon, $Q \subseteq P_1$ is the inner halfblock of some pair of adjacent vertices, and we construct a polygon P_2 by cutting Q from P_1; that is, P_2 is the closure of $P_1 - Q$. Then, the strong kernel of P_2 is identical to the strong kernel of P_1.*

Proof. We first show that every point p of P_2's strong kernel is in the strong kernel of P_1. If p is in the strong kernel of P_2, it is strongly visible from all vertices of P_2. Since every vertex of P_1 is a vertex of P_2, and P_1 is a superset of P_2, we conclude that p is strongly visible from all vertices of P_1. Therefore, p is in the strong kernel of P_1 by Lemma 9.

We next show that, conversely, every point p of P_1's strong kernel is in the strong kernel of P_2. Consider adjacent vertices x and y, which give rise to the inner halfblock Q, and let z be the third vertex of Q. We again assume that the edge between x and y is vertical, x is above y, and z is to the right of the edge (Figure 9b).

We have shown in the proof of Lemma 10 that p is in the shaded angle. Therefore, for every vertex q of P_2, the $\mathcal{O}$-block of p and q is either above the line through x and z or below the line through y and z (Figure 9c); thus, the $\mathcal{O}$-block of p and q

does not intersect the interior of Q. We conclude that p is strongly visible from all vertices of P_2, which implies that p is in the strong kernel of P_2 by Lemma 9. □

Lemma 12. *If all edges of a polygon are $\mathcal{O}$-oriented, then its strong kernel is identical to its standard kernel.*

Proof. If a point p is in the strong kernel of the polygon, then it is in the standard kernel. To show the converse, suppose that p is not in the strong kernel, and q is a point that is not strongly visible from p (Figure 9d). We pick a point x that is in the $\mathcal{O}$-block of p and q, and not in the polygon. Let y be the first intersection of the straight path from x to q with the polygon's boundary; we show the boundary by solid lines in Figure 9(d). Since the polygon's edge through y is $\mathcal{O}$-oriented, it does not intersect the shaded angle, except in its vertex y. Thus, y is not standardly visible from p, which implies that p is not in the standard kernel. □

To find the strong kernel of a polygon, we construct the inner halfblock for every edge, as shown in Figure 6(e). The concatenation of the halfblock boundaries forms a new polygon, whose standard kernel is identical to the strong kernel of the original polygon. If some halfblocks are not contained in the original polygon, then the new polygon is not simple, and its kernel is empty. If all halfblocks are in the original polygon, then the new polygon is simple, and the computation of its kernel takes $O(n)$ time. The construction of each inner halfblock takes $O(\lg m)$ time, which leads to the following result.

Theorem 13. *If $\mathcal{O}$ is a sorted collection of m disjoint ranges, then the time of constructing the strong kernel of an n-vertex polygon is $O(n \cdot \lg m)$.*

6. Visibility from a Point

Suppose that a finite set of "obstacles" obstructs visibility, and two points are considered strongly visible to each other if their $\mathcal{O}$-block does not intersect any obstacles. We develop an algorithm for finding all points visible from a given spot.

First, suppose that all obstacles are *points,* and consider the problem of finding *all obstacle points* that are strongly visible from a given point p. We illustrate this problem in Figure 10(b), where little squares mark the obstacle points visible from p, and triangles show the invisible obstacles.

Consider the angles formed by the $\mathcal{O}$-oriented lines through p (Figure 10c), and observe that an obstacle point affects visibility from p only within the angle that contains this point. In particular, if an obstacle is inside a range of $\mathcal{O}$-oriented lines, shown by the shaded ranges in Figure 10(c), then it obstructs visibility only along the line through p and the obstacle.

If a point is in an angle between two ranges, then it obstructs part of the angle, and does not affect visibility in other angles; thus, we can solve the visibility problem separately for each angle. We use the sides of an angle as coordinate axes, and

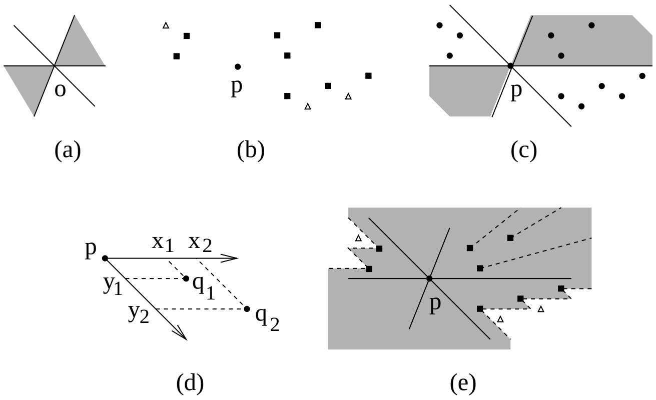

Fig. 10. Construction of a strong-visibility polygon for point obstacles.

Let $q_1, q_2, ..., q_k$ be the obstacles in the angle,
sorted in the increasing order of x-coordinates,
and $y_1, y_2, ..., y_k$ be their y-coordinates.

Visible $:= \{q_1\}$ (set of visible obstacle points)
$y_{\min} := y_1$ (minimal y among the processed obstacles)
for $i := 2$ **to** k **do**
  if $y_i < y_{\min}$
    then *Visible* $:=$ *Visible* $\cup \{q_i\}$
      $y_{\min} := y_i$
return *Visible*

Fig. 11. Finding the obstacles that are strongly visible from p, for one of the angles.

assign x and y coordinates to each obstacle in the angle, as shown in Figure 10(d). Then, an obstacle q_2 is not visible from p if and only if there is an obstacle q_1 such that $x_1 \le x_2$ and $y_1 \le y_2$. This observation readily leads to a procedure for identifying the visible obstacles, given in Figure 11.

The procedure first sorts the obstacle points, contained in the angle, in the increasing order of their x-coordinates. If several points have the same x-coordinate, it sorts these points in the increasing order of their y-coordinates. Then, the procedure processes the obstacle points in the sorted order. If an obstacle's y-coordinate is strictly smaller than the y-coordinates of all previously processed obstacles, then it is visible from p. A convenient visualization of this procedure is a sweep of a parallel-to-y line through the obstacles, in the direction of increasing x, as illustrated in Figure 12.

The top-level algorithm groups the obstacles by angles and then calls the sweep procedure for each angle. If the total number of obstacles is n, then finding all angles that contain at least one obstacle takes $O(n \cdot \lg m)$ time, grouping the obstacles by

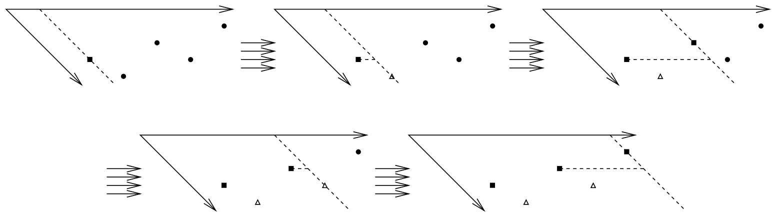

Fig. 12. Line-sweep view of the visibility computation.

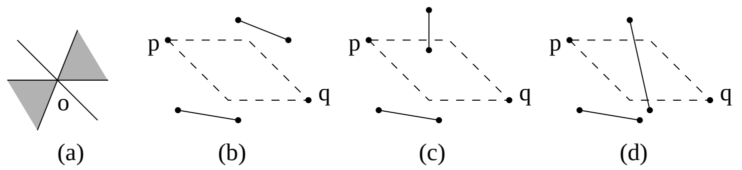

Fig. 13. Proof of Lemma 14.

angle and sorting them within each angle takes $O(n \cdot \lg n)$ time, and applying the sweep procedure to all angles takes $O(n)$ time. Thus, the overall running time of identifying the obstacles visible from p is $O(n \cdot (\lg n + \lg m))$.

We adapt the same algorithm to finding the set of *all* points that are strongly visible from p. We show the boundary of this set in Figure 10(e); the boundary includes (1) "invisible" rays in the ranges of $\mathcal{O}$-oriented lines through p and (2) "stair-shaped" polygonal lines through the visible obstacles in the angles between ranges, one polygonal line per angle. The procedure first identifies the invisible rays and then constructs the stair-shaped lines. It constructs these lines during the sweeps, shown in Figure 12, which does not increase the overall sweep time.

The resulting set of visible points is an unbounded star-shaped polygon, shown by the shaded area in Figure 10(e), which is called the *strong-visibility polygon* of p. This polygon is open; that is, it does not include its boundary. The next result allows extending the construction to *straight-segment* obstacles.

Lemma 14. *Suppose that $Obst_1$ is a set of segment obstacles, and $Obst_2$ is the obstacle set formed by the segments' endpoints. Then, two points are strongly visible to each other with respect to $Obst_1$ if and only if they are strongly visible with respect to $Obst_2$ and standardly visible with respect to $Obst_1$.*

Proof. Clearly, if points p and q are strongly visible with respect to $Obst_1$, then they are strongly visible for $Obst_2$ and standardly visible for $Obst_1$ (Figure 13b). If p and q are not strongly visible with respect to $Obst_1$, then either (1) the endpoint of some obstacle segment is in their $\mathcal{O}$-block, which means that they are not strongly visible with respect to $Obst_2$ (Figure 13c), or (2) one of the obstacles cuts through the $\mathcal{O}$-block and obstructs standard visibility between p and q (Figure 13d). $\square$

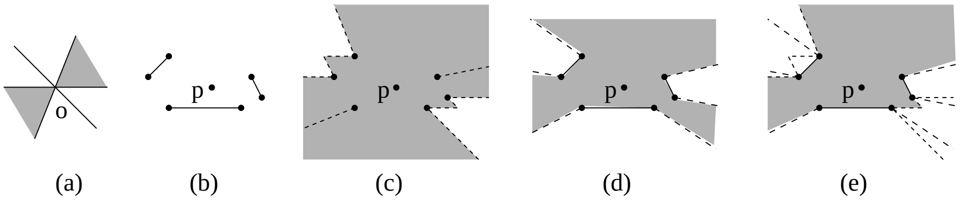

Fig. 14. Construction of a strong-visibility polygon for segment obstacles.

The algorithm for constructing the strong-visibility polygon of a point p, with respect to a set $Obst_1$ of n segment obstacles, consists of three steps. We illustrate these steps for the obstacle set in Figure 14(b). First, the algorithm identifies the set $Obst_2$ of the obstacle endpoints, and finds the strong-visibility polygon with respect to $Obst_2$ (Figure 14c). Second, it constructs the standard-visibility polygon with respect to $Obst_1$ (Figure 14d), which takes $O(n \cdot \lg n)$ time.[10]

Finally, the third step is to compute the intersection of the two polygons, as shown in Figure 14(e). Most linear-time algorithms for finding the intersection of convex polygons[19] are readily applicable to the intersection of two star-shaped polygons with a common kernel point. Since the two visibility polygons have the common kernel point p, we can compute their intersection in $O(n)$ time.

Theorem 15. *If $\mathcal{O}$ is a sorted collection of m disjoint ranges, and the obstacle set contains n edges, then the time of constructing the strong-visibility polygon of a point is $O(n \cdot (\lg n + \lg m))$.*

Finally, note that we can use the same algorithm for polygonal obstacles and for visibility inside a polygon since these problems are readily reducible to visibility with respect to segment obstacles.

7. Concluding Remarks

The reported work is the first step in exploring computational properties of strong visibility and convexity. We have developed algorithms for computing the strong convex hull of a point set, the strong kernel of a polygon, and the strong-visibility polygon of a point surrounded by segment obstacles. The dependency between the size m of the orientation set and the running time is logarithmic, which means that the computation is efficient for large m. In Table 1, we compare the running times of the developed procedures with the running times of algorithms for standard visibility and convexity.

We have not investigated the lower bounds for the running times, which leaves the possibility of improving the described algorithms. We also have not studied strong visibility defined by an asymmetric orientation set, that is, a set of rays from a point o rather than lines through o. Other related problems include dynamic maintenance of the strong hull, construction of maximal strongly convex subsets of a given polygon, and dynamic maintenance of the strong-visibility polygon.

Table 1. Comparison of worst-case complexity results for standard and strong visibility.

	standard visibility	strong visibility
visibility of two points	n	$n + \lg m$
convexity of a polygon	n	$n + \lg \frac{n+m}{n}$
hull of a point set	$n \cdot \lg n$	$n \cdot (\lg n + \lg m)$
hull of a simple polygon	n	$n + n \cdot \lg \frac{n+m}{n}$
kernel of a polygon	n	$n \cdot \lg m$
visibility from a point	$n \cdot \lg n$	$n \cdot (\lg n + \lg m)$

References

1. C. K. Bruckner and J. B. Bruckner. On L_n-sets, the Hausdorff metric, and connectedness. *Proceedings of the American Mathematical Society*, 13:765–767, 1962.
2. Eugene Fink and Derick Wood. Fundamentals of restricted-orientation convexity. *Information Sciences*, 92:175–196, 1997.
3. Eugene Fink and Derick Wood. Generalized halfspaces in restricted-orientation convexity. *Journal of Geometry*, 62:99–120, 1998.
4. Eugene Fink and Derick Wood. Strong restricted-orientation convexity. *Geometriae Dedicata*, 69:35–51, 1998.
5. Ronald L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1:132–133, 1972.
6. Branko Grünbaum, Victor Klee, M. A. Perles, and G. C. Shephard. *Convex Polytopes.* John Wiley and Sons, New York, NY, 1967.
7. Ralf Hartmut Güting. *Conquering Contours: Efficient Algorithms for Computational Geometry.* PhD thesis, Fachbereich Informatik, Universitat Dortmund, 1983.
8. Ralf Hartmut Güting. Stabbing C-oriented polygons. *Information Processing Letters*, 16:35–40, 1983.
9. Ralf Hartmut Güting. Dynamic C-oriented polygonal intersection searching. *Information and Control*, 63:143–163, 1984.
10. Der-Tsai Lee and I. M. Chen. Display of visible edges of a set of convex polygons. In Godfried T. Toussaint, editor, *Computational Geometry*, pages 249–265. North-Holland Publishing Co., Amsterdam, Netherlands, 1985.
11. Der-Tsai Lee and Franco P. Preparata. An optimal algorithm for finding the kernel of a polygon. *Journal of the* ACM, 26:415–421, 1979.
12. Witold Lipski and Christos H. Papadimitriou. A fast algorithm for testing for safety and detecting deadlock in locked transaction systems. *Journal of Algorithms*, 2:211–226, 1981.
13. V. Martynchik, Nikolai Metelski, and Derick Wood. O-convexity: Computing hulls, approximations, and orientation sets. In *Proceedings of the Eighth Canadian Conference on Computational Geometry*, pages 2–7, 1996.
14. Duncan McCallum and David Avis. A linear algorithm for finding the convex hull of a simple polygon. *Information Processing Letters*, 9:201–206, 1979.
15. D. Y. Montuno and Alain Fournier. Finding the x-y convex hull of a set of x-y polygons. Technical Report 148, Department of Computer Science, University of Toronto, 1982.
16. J. Ian Munro, Mark H. Overmars, and Derick Wood. Variations of visibility. In *Proceedings of the Third Annual Symposium on Computational Geometry*, pages 291–299,

1987.
17. Tina M. Nicholl, Der-Tsai Lee, Y. Z. Liao, and Chak-Kuen Wong. Constructing the X-Y convex hull of a set of X-Y polygons. BIT, 23:456–471, 1983.
18. Thomas Ottmann, Eljas Soisalon-Soininen, and Derick Wood. On the definition and computation of rectilinear convex hulls. *Information Sciences*, 33:157–171, 1984.
19. Franco P. Preparata and Michael I. Shamos. *Computational Geometry*. Springer-Verlag, New York, NY, 1985.
20. Gregory J. E. Rawlins. *Explorations in Restricted-Orientation Geometry*. PhD thesis, School of Computer Science, University of Waterloo, 1987. Technical Report CS-87-57.
21. Gregory J. E. Rawlins, Sven Schuierer, and Derick Wood. Convexity, visibility, and orthogonal polygons. In Prabir Bhattacharya, Robert A. Melter, and Azriel Rosenfeld, editors, *Vision Geometry, Contemporary Mathematics 119*, pages 225–237. American Mathematical Society, Providence, RI, 1991.
22. Gregory J. E. Rawlins and Derick Wood. Optimal computation of finitely oriented convex hulls. *Information and Computation*, 72:150–166, 1987.
23. Gregory J. E. Rawlins and Derick Wood. Ortho-convexity and its generalizations. In Godfried T. Toussaint, editor, *Computational Morphology*, pages 137–152. Elsevier Science Publishers B. V., North-Holland, Amsterdam, Netherlands, 1988.
24. Gregory J. E. Rawlins and Derick Wood. Restricted-orientation convex sets. *Information Sciences*, 54:263–281, 1991.
25. Sven Schuierer. *On Generalized Visibility*. PhD thesis, Institut für Informatik, Universität Freiburg, 1991.
26. Eljas Soisalon-Soininen and Derick Wood. Optimal algorithms to compute the closure of a set of iso-rectangles. *Journal of Algorithms*, 5:199–214, 1984.
27. Frederick Albert Valentine. Local convexity and L_n-sets. *Proceedings of the American Mathematical Society*, 16:1305–1310, 1965.
28. Peter Widmayer, Ying-Fung Wu, and Chak-Kuen Wong. On some distance problems in fixed orientations. SIAM *Journal on Computing*, 16:728–746, 1987.