Computational Problems in Strong Visibility

Eugene Fink

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
eugene@cs.cmu.edu
http://www.cs.cmu.edu/∼eugene

Derick Wood

Department of Computer Science
Hong Kong Univ. of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
dwood@cs.ust.hk
http://www.cs.ust.hk/∼dwood

ABSTRACT

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. In particular, we describe algorithms for the following tasks:

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

Keywords: computational geometry, generalized visibility and convexity, restricted orientations.

1 INTRODUCTION

The study of nontraditional notions of visibility is a fruitful branch of computational geometry, which has found applications in VLSI design, computer graphics, motion planning, and other areas. Researchers have investigated multiple variations and generalizations of standard visibility and convexity.[1,2,6,11–13,15,18]

Rawlins has introduced the notions of strong visibility and convexity as a part of his research on restricted-orientation geometry.[15] These notions are defined with respect to a fixed set of line orientations. They are stronger than standard visibility and convexity, hence the name. Rawlins and Wood studied the properties of strongly convex sets and demonstrated that strong convexity generalizes not only standard convexity but also the notions of iso-oriented rectangles and convex C-oriented polygons, described by Güting.[6] We extended strong convexity to higher dimensions and investigated the main properties of this extension.[3]

Martynchik, Metelski, Rawlins, Schuierer, and Wood explored computational problems in restricted-orientation geometry and developed a set of algorithms for solving them[9,16–18]; however, they have not presented algorithms for strong visibility and convexity.

The work reported here is the first step in exploring computational properties of strong visibility. We discuss the basic problems, develop methods for solving them, and give the worst-case complexity. Most algorithms are based on the reduction to the analogous problems in standard visibility.

We first define strong visibility and convexity (Section 2), and show how to test the strong visibility of two points in a polygon and the strong convexity of a polygon (Section 3). We then describe the computation of the strong convex hull of a point set (Section 4) and strong kernel of a polygon (Section 5). Finally, we consider the problem of identifying the points visible from a given point (Section 6).

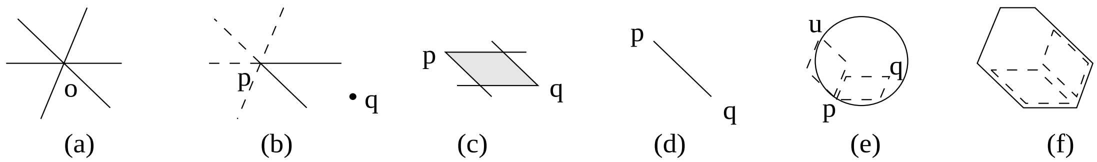

Figure 1. Strong visibility and strong convexity.

2 STRONG VISIBILITY AND STRONG CONVEXITY

We define strong visibility and convexity by analogy with the notions of standard 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**. We then define strong convexity in terms of this new visibility.

We first introduce an **orientation set** $\mathcal{O}$, which is a set of lines through some fixed point o. A straight line is **$\mathcal{O}$-oriented** if it is parallel to one of the lines of $\mathcal{O}$. We assume that the orientation set $\mathcal{O}$ is finite and that it contains at least two distinct lines. We have not used this assumption in the investigation of mathematical properties of strong convexity,[3] but it is essential for the study of computational properties. We give an example of an orientation set in Figure 1(a).

Next, we define the **$\mathcal{O}$-block** of two points, say p and q. We draw all $\mathcal{O}$-oriented rays with endpoint p and pick the two of them closest to q (see Figure 1b). The two selected rays, with the common endpoint p, form the boundary of an angle with vertex p; this angle contains q.

Similarly, we draw the $\mathcal{O}$-oriented rays from q closest to p and get an angle with vertex q, bounded by these rays (see Figure 1c). The **$\mathcal{O}$-block** of p and q is the parallelogram formed by the intersection of these two angles (the shaded parallelogram in Figure 1c). As a special case, if the line through p and q is $\mathcal{O}$-oriented, then the $\mathcal{O}$-block of p and q is the straight segment joining them (Figure 1d).

We use the notion of $\mathcal{O}$-blocks to define strong visibility and strong convexity. Two points of a set are **strongly visible** to each other if the $\mathcal{O}$-block of these two points is contained in the set. A set is **strongly convex** if every two of its points are strongly visible to each other.

For example, the points p and q in Figure 1(e) are strongly visible to each other for the orientation set in Figure 1(a), whereas the points p and u are not (we show the corresponding $\mathcal{O}$-blocks by dashed lines). The circle in Figure 1(e) is not strongly convex, since some of its points are not strongly visible to each other. On the other hand, the polygon in Figure 1(f) is strongly convex (we show two $\mathcal{O}$-blocks contained in it).

Observe that, if two points are strongly visible to each other, they are also "standardly" visible. Therefore, every strongly convex set is convex. On the other hand, a convex set may not be strongly convex (see the circle is Figure 1e).

Basic properties of strong visibility and convexity are similar to that of standard visibility and convexity.[15] In particular, the translation of a set preserves strong visibility between its points and, hence, a translation of a strongly convex set is strongly convex. For every set, each of its point is strongly visible from itself. Finally, the intersection of strongly convex sets is always strongly convex.

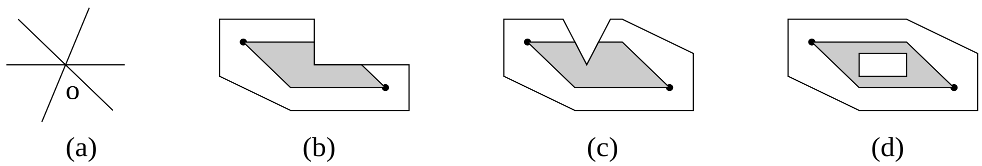

Figure 2. Three cases when points are not strongly visible to each other.

3 VISIBILITY AND CONVEXITY TEST

We begin with two simple problems, determining if two points of a polygon are strongly visible to each other and verifying the strong convexity of a polygon. We assume in all our algorithms that the *orientation set $\mathcal{O}$ is sorted.*

We test the visibility of two points by constructing their $\mathcal{O}$-block and verifying its containment in the polygon. The polygon may *not* be simply connected; that is, it may have "holes." The test is based on the following observation.

LEMMA 3.1. *The $\mathcal{O}$-block of two points is completely 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.*

SKETCH OF A PROOF. We illustrate the violation of each of the three conditions in Figure 2. Clearly, if some condition does not hold, then the $\mathcal{O}$-block is not contained in the polygon. If the conditions hold, then the polygon's boundary does *not* intersect the $\mathcal{O}$-block's interior and, thus, the $\mathcal{O}$-block is completely in the polygon. □

We denote the number of lines in the orientation set $\mathcal{O}$ by m and the number of the polygon's vertices by n. The construction of the $\mathcal{O}$-block of two points takes $O(\log m)$ time, because we need to find the two $\mathcal{O}$-orientations closest to the orientation of the line through these two points (see Figure 1b). The test for each of the three conditions of Lemma 3.1 takes $O(n)$ time.

We may encounter a special case when the line through the two points is $\mathcal{O}$-oriented and their $\mathcal{O}$-block is the straight segment. We detect this case in $O(\log m)$ time, when constructing the $\mathcal{O}$-block. Then, the points are strongly visible if and only if they are standardly visible, and the verification of standard visibility takes $O(n)$ time. We thus obtain the following result.

THEOREM 3.2. *If the orientation set $\mathcal{O}$ is a sorted set of m lines, then the worst-case time complexity of testing the strong visibility of two points in an n-vertex polygon is $O(n + \log m)$.*

We now consider the problem of verifying the strong convexity of a polygon. The verification algorithm is based on the following result.

THEOREM 3.3. *A polygon is strongly convex if and only if it is convex and all its edges are $\mathcal{O}$-oriented.*

SKETCH OF A PROOF. Clearly, if a polygon is not convex, it is not strongly convex. If some edge is not $\mathcal{O}$-oriented, then the $\mathcal{O}$-block of its endpoints is not completely in the polygon (see Figure 3a) and we again conclude 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 lines. Every two points of a halfplane with an $\mathcal{O}$-oriented boundary are strongly

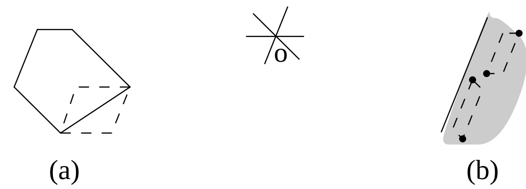

Figure 3. Proof of Theorem 3.3.

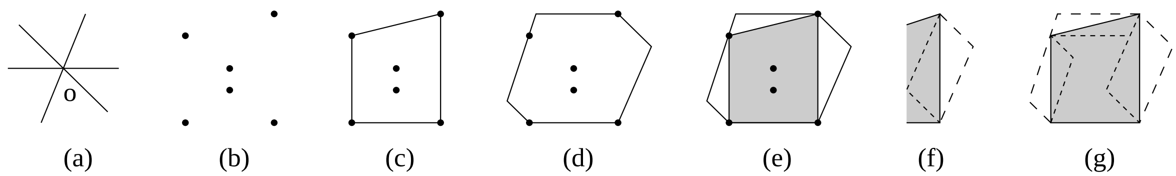

Figure 4. Construction of the strong convex hull.

visible to each other, as illustrated in Figure 3(b). Thus, the polygon is the intersection of strongly convex halfplanes, which implies that it is strongly convex. □

Testing the 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. We need to verify that the sorted list of the edge orientations is a subset of the sorted list of $\mathcal{O}$-orientations.

If $m \leq n$, we can test whether an n-element sorted list is a subset of an m-element sorted list in $O(n)$ time. If $m > n$, we can perform the test in $O(n \cdot (1 + \log \frac{m}{n}))$ time; note that this complexity is strictly better than $O(m)$. We may express the worst-case complexity of the test as $O(n \cdot (1 + \log \frac{n+m}{n}))$, which is equivalent to $O(n)$ for $m \leq n$ and to $O(n \cdot (1 + \log \frac{m}{n}))$ for $m > n$, thus obtaining the following result.

THEOREM 3.4. *If $\mathcal{O}$ is a sorted set of m lines, then the worst-case time complexity of testing the strong convexity of an n-vertex polygon is $O(n \cdot (1 + \log \frac{n+m}{n}))$.*

4 STRONG CONVEX HULL

The **convex hull** of a geometric object is the minimal convex set that contains the object. Similarly, we define the **strong convex hull** as the minimal *strongly* convex set that contains the object. For example, consider the set of six points in Figure 4(b). In Figure 4(c), we show the standard hull of this set. In Figure 4(d), we give the strong hull, for the orientation set of Figure 4(a).

We describe the computation of the strong convex hull of a finite point set. The algorithm is based on the following observation.

LEMMA 4.1. *The strong hull of a geometric object is identical to the strong hull of the standard hull of the object.*

SKETCH OF A PROOF. Let P be a geometric object, H be its standard hull, and *S-hull*(P) and *S-hull*(H) be their strong hulls. We illustrate it in Figure 4(e), where P is the set of six points and H is the shaded polygon. We note that $P \subseteq H$ and, thus, *S-hull*$(P) \subseteq$ *S-hull*(H). On the other hand, since *S-hull*(P) is convex and H is the *minimal* convex set containing P, we have $H \subseteq$ *S-hull*(P) and, hence, *S-hull*$(H) \subseteq$ *S-hull*(P). We conclude that *S-hull*$(H) =$ *S-hull*(P). □

We begin the construction of the strong hull of an n-point set by finding the standard hull of the set. The computation of the standard hull takes $O(n \cdot \log n)$ time.[4] The resulting hull is a convex polygon with at most n vertices. We now describe the computation of 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. We call it the **outer halfblock** of the adjacent vertices. We illustrate this construction in Figure 4(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 the $\mathcal{O}$-block. The concatenation of the boundaries of the outer halfblocks forms the contour that bounds the strong hull of the polygon (see Figure 4g).

The construction of every $\mathcal{O}$-block requires finding the two $\mathcal{O}$-orientations closest to the orientation of the corresponding edge, which takes $O(\log m)$ time. The overall time of finding the standard hull of the point set and then constructing the $\mathcal{O}$-block for each of its edges is $O(n \cdot (\log n + \log m))$ in the worst case.

Note that the orientations of the standard hull's edges are in sorted order, which allows us to reduce the time of finding the two closest $\mathcal{O}$-orientations for each of the edges. We can find the closest orientations and construct the $\mathcal{O}$-blocks in $O(n \cdot (1 + \log \frac{n+m}{n}))$ time. Unfortunately, it does *not* reduce the overall time complexity, because adding the $O(n \cdot \log n)$ time of computing the standard hull makes the total time $O(n \cdot (\log n + \log m))$.

The next observation implies that we may use the same algorithm for computing the strong hull of a polygon and, thus, the worst-case time complexity of finding a polygon's hull is also $O(n \cdot (\log n + \log m))$.

LEMMA 4.2. *The strong hull of a polygon is identical to the strong hull of the point set formed by the polygon's vertices.*

SKETCH OF A PROOF. We observe that the standard hull of a polygon is identical to the standard hull of its vertices.[5] This observation and Lemma 4.1 imply that the analogous property holds for strong hulls. □

If a polygon is simple, we can find its standard convex hull in $O(n)$ time,[10] which allows us to reduce the time complexity of constructing the strong hull to $O(n \cdot (1 + \log \frac{n+m}{n}))$. We now state the results on the computation of the strong hull as a theorem.

THEOREM 4.3. *Suppose that $\mathcal{O}$ is a sorted set of m lines.*

1. *The worst-case time complexity of finding the strong convex hull of a set of n points or that of an n-vertex polygon is $O(n \cdot (\log n + \log m))$.*
2. *The worst-case time complexity of finding the strong convex hull of a simple polygon with n vertices is $O(n \cdot (1 + \log \frac{n+m}{n}))$.*

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, consider the polygon in Figure 5(b). The shaded region in Figure 5(c) is the kernel of this polygon. Note that an object is convex if and only if its kernel is identical to the object itself. Also note that only simply connected objects may have nonempty kernels.

We define the **strong kernel** as the set of points that are *strongly* visible from all points of the object. In Figure 4(d), we give the strong kernel of the same polygon, for the orientation set of Figure 5(a).

The computation of the standard kernel of a polygon with n vertices takes $O(n)$ time.[8] We solve the problem of finding the strong kernel of a polygon by reducing it to the standard-kernel computation. We derive the properties of the strong kernel used in the reduction and then describe the algorithm.

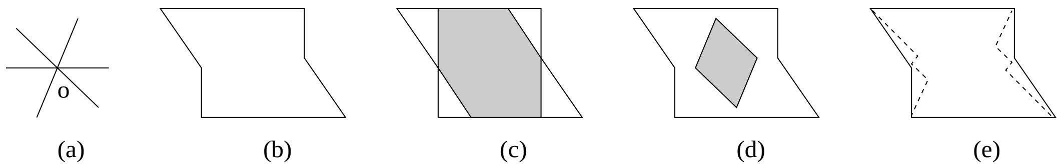

Figure 5. Standard kernel, strong kernel, and inner haflblocks of a polygon.

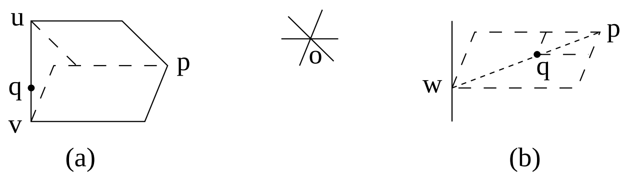

Figure 6. Proof of Lemma 5.1.

We begin by observing that a point of a polygon belongs to the kernel if and only if it is visible from every *vertex* of the polygon (see, for example, the textbook by Preparata and Shamos[14]). We generalize this observation to strong visibility.

LEMMA 5.1. *A point of a polygon is in the polygon's strong kernel if and only if it is strongly visible from all vertices of the polygon.*

SKETCH OF A PROOF. 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 of the polygon. We denote the endpoints of this edge by u and v (see Figure 6a). $\mathcal{O}$-block(p, u) and $\mathcal{O}$-block(p, v) are both in the polygon. Therefore, the pentagon formed by these $\mathcal{O}$-blocks and the edge, which is marked by solid lines in Figure 6(a), is completely in the polygon. $\mathcal{O}$-block(p, q) is contained in this pentagon, 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 w of the intersection (see Figure 6b). Since p is strongly visible from w and $\mathcal{O}$-block$(p, q) \subseteq \mathcal{O}$-block$(p, w)$, we conclude that p is strongly visible from q. □

We now define inner halfblocks of a polygon. The **inner halfblock** of two adjacent vertices is the half of their $\mathcal{O}$-block opposite to the outer halfblock (see Figure 4f). We show the inner halfblocks of the polygon in Figure 4(g) by dotted lines. An inner halfblock may *not* be completely in the polygon, as illustrated in Figure 7(b).

LEMMA 5.2. *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.*

SKETCH OF A PROOF. Let u and v be adjacent vertices, whose inner halfblock is not completely in the polygon, and w be the third vertex of their halfblock (see Figure 7c). 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 v and w, because then $\mathcal{O}$-block(p, v) would not be completely in the polygon (see Figure 7d). Similarly, p cannot be below the line through u and w. We conclude that p is in the shaded angle.

We now pick a point x that is in the halfblock and *not* in the polygon, and denote the intersection of the line through p and x with the edge by q (Figure 7e). Then, p is not strongly visible from q, which leads to a contradiction. □

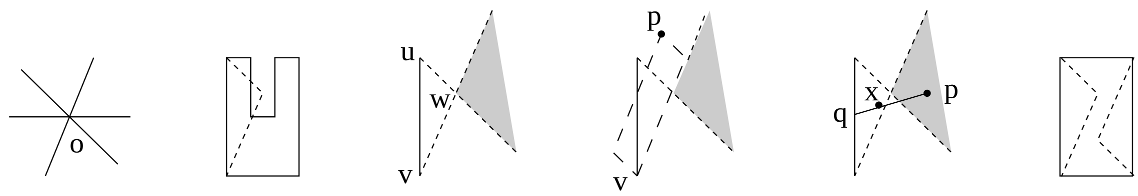

Figure 7. Proof of Lemma 5.2.

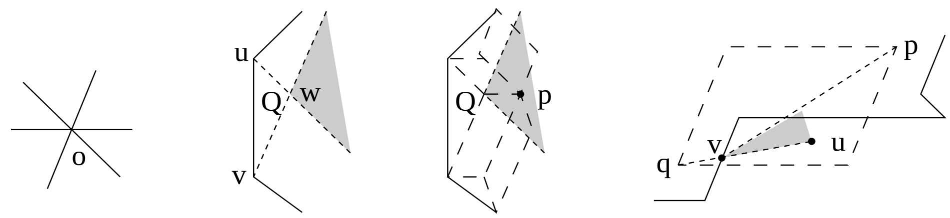

Figure 8. Proofs of Theorem 5.3 and Theorem 5.4.

The converse of Lemma 5.2 does not hold: a polygon's kernel may be empty even if every inner halfblock is completely in the polygon. We show such a polygon in Figure 7(f).

THEOREM 5.3. *Suppose that P_1 is a polygon and $Q \subseteq P_1$ is the inner halfblock of some pair of adjacent vertices, and that we construct a polygon P_2 by cutting Q from P_1; that is, $P_2 = Closure(P_1 - Q)$. Then, the strong kernel of P_2 is identical to the strong kernel of P_1.*

SKETCH OF A PROOF. Clearly, the strong kernel of P_2 is a subset of P_1's strong kernel. We show that, conversely, every point p of the P_1's strong kernel is in the strong kernel of P_2. We denote the adjacent vertices, whose inner halfblock is Q, by u and v, and the third vertex of Q by w (see Figure 8b).

We have shown in the proof of Lemma 5.2 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 u and w or below the line through v and w (Figure 8c), which means that $\mathcal{O}$-block(p, q) does *not* intersect the interior of Q. We conclude that p is strongly visible from all vertices of P_2 and hence, by Lemma 5.1, it is in the strong kernel of P_2. $\square$

THEOREM 5.4. *If all edges of a polygon are $\mathcal{O}$-oriented, then its strong kernel is identical to its standard kernel.*

SKETCH OF A 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 the polygon's point that is not strongly visible from p (see Figure 8d). We pick a point u that is in $\mathcal{O}$-block(p, q) and *not* in the polygon. Let v be the first intersection of the straight path from u to q with the polygon's boundary (we show the boundary by solid lines in Figure 8d). Since the polygon's edge through v is $\mathcal{O}$-oriented, it does not intersect the shaded angle, except in its vertex v. Therefore, v is not standardly visible from p and, thus, p is not in the standard kernel. $\square$

To find the strong kernel of a polygon, we first construct the inner halfblock for every edge of the polygon, as shown in Figure 5(e). The concatenation of the halfblock boundaries forms a 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

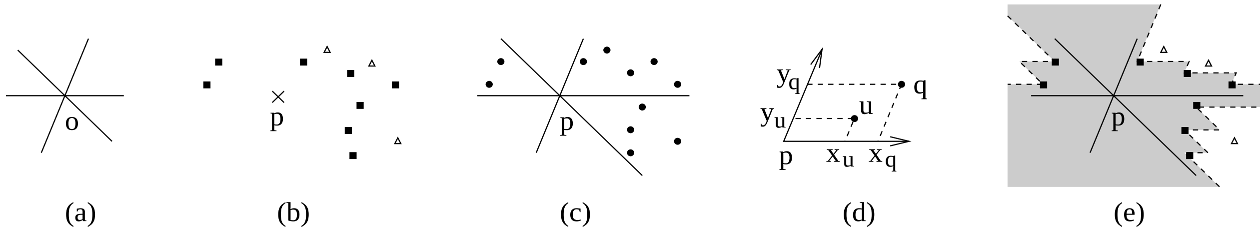

Figure 9. Construction of a strong visibility polygon for point obstacles.

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

Visible $:= \emptyset$; $y_{\min} := \infty$
for $i := 1$ **to** k **do**
  if $y_i \leq y_{\min}$ **then** Visible := Visible $\cup \{q_i\}$; $y_{\min} := y_i$

Figure 10. Finding the obstacles that are strongly visible from p, for one of the angles.

original polygon, the new polygon is simple. The construction of each inner halfblock takes $O(\log m)$ time, which gives us the following complexity result.

THEOREM 5.5. *If $\mathcal{O}$ is a sorted set of m lines, then the worst-case time complexity of constructing the strong kernel of an n-vertex polygon is $O(n \cdot \log m)$.*

6 STRONG VISIBILITY FROM A POINT

Suppose that a finite set of "obstacles" obstructs visibility in the plane. Two points are strongly visible to each other if the interior of their $\mathcal{O}$-block does not intersect any obstacles. We consider the problem of finding the points visible from a given point.

We first suppose that all obstacles are *points* and describe an algorithm for finding *all obstacle points* that are strongly visible from some point p. We illustrate this problem in Figure 9(b), where squares mark the obstacle points visible from p and triangles show the invisible obstacles.

We consider the angles formed by the $\mathcal{O}$-oriented lines through p (see Figure 9c). An obstacle point obstructs the visibility from p only within the angle that contains this point, which allows us to solve the visibility problem separately for each angle.

We use the sides of an angle as coordinate axes and assign x and y coordinates to each obstacle in the angle, as shown in Figure 9(d). Then, an obstacle q is not visible from p if and only if there is an obstacle u such that $x_u < x_q$ and $y_u < y_q$. We use this observation to find the obstacles visible from p.

We sort the obstacle points, contained in the angle, on their x-coordinates and then process them in order. If an obstacle's y-coordinate is no larger than the y-coordinate of any previously processed obstacle, then it is visible from p. We may visualize this algorithm as a sweep of a y-oriented line through the obstacles, in the direction of increasing x. We summarize the algorithm in Figure 10 and illustrate the sweep in Figure 11.

We denote the number of obstacles by n. Finding all angles that contain at least one obstacle takes $O(n \cdot \log m)$ time. Grouping the obstacles by angle and sorting them within each angle takes $O(n \cdot \log n)$

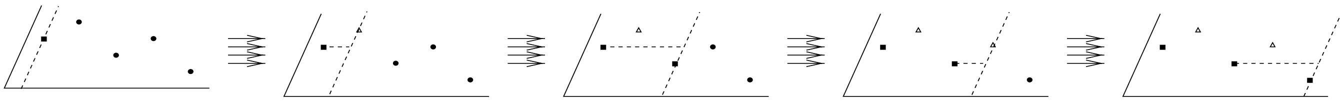

Figure 11. Line-sweep view of the visibility computation.

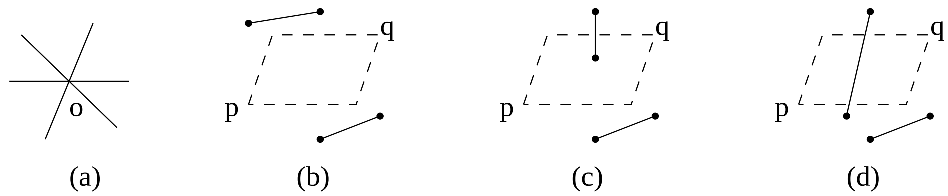

Figure 12. Proof of Lemma 6.1.

time. Finally, the sweeps in all angles together take $O(n)$ time. The overall worst-case time complexity of identifying the obstacles visible from p is $O(n \cdot (\log n + \log m))$.

We can use the same algorithm to find the set of *all* points that are strongly visible from p. This set is bounded by "stair-shaped" polygonal lines through the obstacles visible from p, one polygonal line in every angle, as shown in Figure 9(e). We construct these lines during the sweeps, which does not increase the overall time complexity. The resulting set of visible points is an unbounded star-shaped polygon, called the **strong visibility polygon** of p. We next extend the construction of the visibility polygon to allow *segment* obstacles, using the following result.

LEMMA 6.1. *Suppose that S_1 is a set of segment obstacles and S_2 is the obstacle set formed by the segment's endpoints. Then, two points are strongly visible to each other with respect to S_1 if and only if they are strongly visible with respect to S_2 and standardly visible with respect to S_1.*

SKETCH OF A PROOF. Clearly, if points p and q are strongly visible w.r.t. S_1, they are strongly visible w.r.t. S_2 and standardly visible w.r.t. S_1 (see Figure 12b). If p and q are *not* strongly visible w.r.t. S_1, then either (1) the endpoint of some obstacle segment is in their $\mathcal{O}$-block and they are not strongly visible w.r.t. S_2 (Figure 12c), or (2) one of the obstacles cuts through the $\mathcal{O}$-block and obstructs the standard visibility between p and q (Figure 12d). □

We use three steps to construct the strong visibility polygon of a point p, with respect to a set S_1 of n segment obstacles (see Figure 13b). First, we identify the set S_2 of their endpoints and find the strong visibility polygon with respect to S_2 (Figure 13c). Second, we construct the standard visibility polygon with respect to S_1, which takes $O(n \cdot \log n)$ time in the worst case[7] (see Figure 13d).

Finally, we find the intersection of the two polygons (Figure 13e). Most linear-time algorithms for finding the intersection of two convex polygons[14] are readily applicable to the intersection of two star-shaped

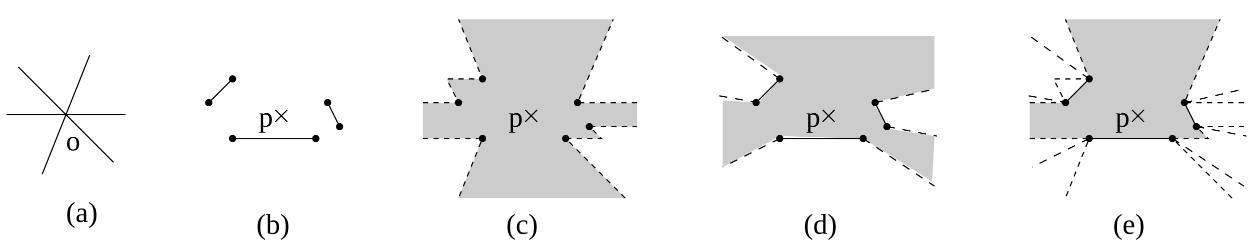

Figure 13. Construction of a strong visibility polygon for segment obstacles.

polygons with a common kernel point. Since our two polygons have the common kernel point p, we can compute their intersection in $O(n)$ time.

THEOREM 6.2. *If $\mathcal{O}$ is a sorted set of m lines and the obstacle set comprises n edges, then the worst-case time complexity of constructing the strong visibility polygon of a point is $O(n \cdot (\log n + \log m))$.*

We may use the same algorithm for polygonal obstacles and for visibility inside a polygon, since these problems are reducible to visibility with respect to segment obstacles.

7 CONCLUSIONS

We have begun the investigation of computational properties of strong visibility and convexity. We 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. The dependency between the running time of these algorithms and the number m of $\mathcal{O}$-orientations is logarithmic. Hence, they are efficient for large m.

Our research leaves many open problems, such as dynamic maintenance of the strong convex hull, construction of maximal strongly convex subsets of a polygon, finding all pairs of points in a finite point set that are strongly visible to each other, and dynamic maintenance of the strong visibility polygon, as well as extending planar algorithms to three and higher dimensions. We also have not investigated the lower bounds for the worst-case time complexity of solving strong-visibility problems, which leaves the possibility of improving the complexity of our algorithms.

ACKNOWLEDGEMENTS

This work was supported under grants from the Natural Sciences and Engineering Research Council of Canada, the Information Technology Research Centre of Ontario, and the Research Grant Committee of Hong Kong.

REFERENCES

1. Eugene Fink and Derick Wood. Fundamentals of restricted-orientation convexity. *Information Sciences*, 1997. To appear.
2. Eugene Fink and Derick Wood. Generalized halfspaces in restricted-orientation convexity. *Journal of Geometry*, 1997. To appear.
3. Eugene Fink and Derick Wood. Strong restricted-orientation convexity. *Geometriae Dedicata*, 1997. To appear.
4. R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1:132–133, 1972.
5. Branko Grünbaum, Victor Klee, M. A. Perles, and G. C. Shephard. *Convex Polytopes*. John Wiley & Sons, New York, NY, 1967.
6. Ralf Hartmut Güting. *Conquering Contours: Efficient Algorithms for Computational Geometry*. PhD thesis, Universitat Dortmund, Germany, 1983.
7. D. T. 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, 1985.
8. D. T. Lee and Franco P. Preparata. An optimal algorithm for finding the kernel of a polygon. *Journal of the ACM*, 26:415–421, 1979.
9. 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.

10. D. McCallum and D. Avis. A linear algorithm for finding the convex hull of a simple polygon. *Information Processing Letters*, 9:201–206, 1979.
11. 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.
12. Tina M. Nicholl, D. T. Lee, Y. Z. Liao, and C. K. Wong. Constructing the X-Y convex hull of a set of X-Y polygons. *BIT*, 23:456–471, 1983.
13. Thomas Ottmann, Eljas Soisalon-Soininen, and Derick Wood. On the definition and computation of rectilinear convex hulls. *Information Sciences*, 33:157–171, 1984.
14. Franco P. Preparata and Michael I. Shamos. *Computational Geometry*. Springer-Verlag, New York, NY, 1985.
15. Gregory J. E. Rawlins. *Explorations in Restricted-Orientation Geometry*. PhD thesis, University of Waterloo, Ontario, 1987. Technical Report CS-87-57.
16. Gregory J. E. Rawlins, Sven Schuierer, and Derick Wood. Convexity, visibility, and orthogonal polygons. In P. Bhattacharya, R. A. Melter, and A. Rosenfeld, editors, *Vision Geometry, Contemporary Mathematics 119*, pages 225–237. American Mathematical Society, Providence, RI, 1991.
17. Gregory J. E. Rawlins and Derick Wood. Optimal computation of finitely oriented convex hulls. *Information and Computation*, 72:150–166, 1987.
18. Sven Schuierer. *On Generalized Visibility*. PhD thesis, Universität Freiburg, Germany, 1991.