A Characterization of Polygonal Regions Searchable from the Boundary

Xuehou Tan

Tokai University, 317 Nishino, Numazu 410-0395, Japan
tan@wing.ncc.u-tokai.ac.jp

Abstract. We consider the problem of searching for a moving target with unbounded speed in a dark polygonal region by a searcher. The searcher continuously moves on the polygon boundary and can see only along the rays of the flashlights emanating from his position at a time. We present necessary and sufficient conditions for a polygon of n vertices to be *searchable from the boundary*. Our two main results are the following:

1. We present an $O(n \log n)$ time and $O(n)$ space algorithm for testing the searchability of simple polygons. Moreover, a search schedule can be reported in time linear in its size I, if it exists. For the searcher having full 360° vision, $I < 2n$, and for the searcher having only one flashlight, $I < 3n^2$. Our result improves upon the previous $O(n^2)$ time and space solution, given by LaValle et al [5]. Also, the linear bound for the searcher having full 360° vision solves an open problem posed by Suzuki et al [7].
2. We show the equivalence of the abilities of the searcher having only one flashlight and the one having full 360° vision. Although the same result has been obtained by Suzuki et al [7], their proof is long and complicated, due to lack of the characterization of boundary search.

1 Introduction

Recently, much attention has been devoted to the problem of searching for an unpredictable, moving target with unbounded speed in an n-sided polygon P by a mobile searcher [5, 6, 7]. Both the searcher and the target are modeled by points that can continuously move in P. A searcher is called the *k-searcher* if he holds k flashlights, and can see only along the rays of the flashlights emanating from his position at a time, or the ∞-*searcher* if he has a light bulb that gives full 360° vision. The searcher can rotate a flashlight, with bounded speed to change the direction of the flashlight. The objective is to decide whether there exists a *search schedule* for the searcher to detect the target (i.e., the target is finally illuminated by the ray of some flashlight, no matter how he moves), and if so, generate a search schedule. A polygon is said to be *k-searchable* or ∞-*searchable* if there exists a search schedule for the searcher to detect the target.

Motivated by robotics applications, LaValle *et al.* considered a simple model, in which the searcher continuously moves on the boundary of P and holds only

J. Akiyama et al. (Eds.): IJCCGGT 2003, LNCS 3330, pp. 200–215, 2005.
© Springer-Verlag Berlin Heidelberg 2005

one flashlight [5]. By constructing a two-dimensional diagram of size $\Omega(n^2)$, they gave an $O(n^2)$ time and space algorithm for generating a search schedule, if it exists [5]. On the other hand, Suzuki *et al.* showed that any polygon searchable by the ∞-searcher from the boundary is also searchable by the 1-searcher from the boundary [7]. Due to lack of the characterization of boundary search, their proof is long and complicated. Whether or not a good (e.g., linear) bound on the size of search schedules for the ∞-searcher can be established is left as an open problem in [7].

In this paper, we present necessary and sufficient conditions for a polygon to be *searchable from the boundary*, and provide efficient algorithms for determining the searchability of simple polygons and generating a search schedule if it exists. The first necessary condition states that a polygon P is not searchable from the boundary if there are three points p_1, p_2, p_3 on the boundary of P such that the Euclidean shortest path between any pair of p_i, p_j ($i, j \in \{1, 2, 3\}$) within P contains no point visible from the third point p_k ($k \neq i$ or j). The second and third conditions together state that a polygon P is not searchable from the boundary if *every* boundary point of P is surrounded by at least one of three special configurations; these configurations provide a place for the target to defend himself from the first (or initial) attack made by the searcher. If none of these conditions is true, then P is searchable from the boundary.

The paper is structed as follows. Section 2 reviews the two-guard problem [3, 4], which is used as a subroutine in our search algorithm. In Section 3, we give three necessary and sufficient conditions for the polygons to be searchable from the boundary. Based on this characterization, the equivalence of the abilities of the 1-searcher and the ∞-searcher is established. In Section 4, we describe an $O(n \log n)$ and $O(n)$ space algorithm for testing the searchability of simple polygons. A search schedule can be reported in time linear in its size I, if it exists. For the ∞-searcher, $I < 2n$, and for the 1-searcher, $I < 3n^2$.

2 Review of the Two-Guard Problem

Let P denote a simple polygon (without holes or self-intersections). Two points $x, y \in P$ are said to be mutually *visible* if the segment $\overline{xy}$ is entirely contained in P. For two regions $P_1, P_2 \subseteq P$, we say that P_1 is *weakly visible* from P_2 if every point in P_1 is visible from some point in P_2.

A *corridor* P is a simple polygon with two marked boundary points u and v. The two-guard problem [3, 4] asks if there exists a walk such that two guards l and r move along two polygonal chains L and R oriented from u to v, one clockwise and one counterclockwise, in such a way that l and r are always mutually visible. For two points $p, p' \in L$, we say that p *precedes* p' (and p' *succeeds* p) if we encounter p before p' when traversing L from u to v. If p precedes p', we write $p < p'$. We define these concepts for R in a similar manner.

For a vertex x of a polygonal chain, let $Succ(x)$ denote the vertex of the chain immediately succeeding x, and $Pred(x)$ the vertex immediately preceding x. A vertex of P is *reflex* if its internal angle is strictly larger than π. The backward ray shot from a reflex vertex r, denoted by *Backw(r)*, is the first boundary point of P hit by a "bullet" shot at r in the direction from $Succ(r)$ to r, and the forward ray shot *Forw(r)* is the first point hit by the bullet shot at r in the direction from $Pred(r)$ to r (Fig. 1). A pair of reflex vertices $p \in L$, $q \in R$ is said to give a *deadlock* if $q < Backw(p) \in R$ and $p < Backw(q) \in L$ hold or if $q > Forw(p) \in R$ and $p > Forw(q) \in L$ hold. See Fig. 1.

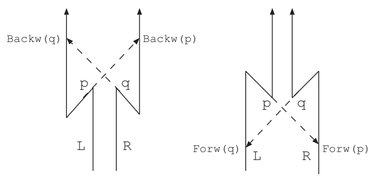

Fig. 1. Deadlocks

Lemma 1. [4] A corridor P is walkable if and only if the chains L and R are mutually weakly visible, and no deadlocks occur.

Also, a walk from one segment $\overline{p_0q_0}$ to another segment $\overline{p_1q_1}$, where $p_0 < p_1$ and $q_0 < q_1$, is possible if and only if two subchains from p_0 to p_1 and from q_0 to q_1 are mutually weakly visible and no deadlocks occur between them. For a walkable corridor, we need to give a *walk schedule*. The walk schedule consists of the following elementary actions: (i) both guards move forward along single edges, and (ii) one guard moves forward, but the other moves backward, along segments of single edges.

Lemma 2. [3,4] It takes $O(n)$ time to test the two-guard walkability of a corridor, and $O(nlogn + I)$ time to generate a walk schedule, where I ($\leq n^2$) is the minimal number of walk instructions.

3 Searching a Polygon from the Boundary

Let P be a simple polygon. Given a boundary point d, we can order all boundary points counterclockwise, starting and ending at d. For a complete ordering, we consider d as two points d_l and d_r such that $d_l \leq p \leq d_r$, for all points p on the boundary of P. Similar definitions can then be given as those in Section 2. For two boundary points p, p', we say that p *precedes* p' (and p' *succeeds* p) if we encounter p before p' when traversing from d_l to d_r. We write $p < p'$ if p precedes p'. For a vertex x, we denote by $Succ(x)$ the vertex immediately succeeding x, and $Pred(x)$ the vertex immediately preceding x. For a reflex vertex r, the backward and forward ray shots $Backw(r)$ and $Forw(r)$ are the first boundary points of P hit by the bullets shot at r in the directions from $Succ(r)$ to r and from $Pred(r)$ to r, respectively. In the case that d is a reflex vertex, the shots $Backw(d)$ and $Forw(d)$ can similarly be defined using $Succ(d_l)$

and $Pred(d_r)$. A pair of reflex vertices x, y is said to give a *deadlock* for the point p if $p_l < x < Forw(y) < Backw(x) < y < p_r$ holds.

In order to simplify the presentation, we denote, by $[u, v]$, the boundary interval from u to v counterclockwise. For an interval X, the point $y \in X$ is said to be the *maximum* (resp. *minimum*), if $y \geq x \in X$ (resp. $y \leq x \in X$).

3.1 Necessity

We present three necessary conditions for a polygon P to be searchable by the ∞-searcher from the boundary. A point $x \in P$ is said to be *detected* or *illuminated* at a time t, if x is contained in the region that is visible from the position of the ∞-searcher at t. Any region that might contain the target at a time is said to be *contaminated*; otherwise, it is said to be *clear*. If a region becomes contaminated for the second or more time, it is referred to as *recontaminated*.

What important in clearing P is to avoid a 'cycle' of recontaminations. Obviously, a cycle of recontanimations occurs if there are three boundary points such that when the ∞-searcher moves between any two of them, the third point is contaminated or recontaminated (Fig. 2).

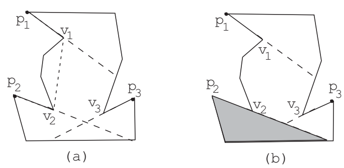

Fig. 2. A polygon satisfying the condition **C1**

Theorem 1. A simple polygon is not ∞-searchable from the boundary if (**C1**) there are three points p_1, p_2 and p_3 on the boundary such that the shortest path between any pair of p_i, p_j $(i, j \in \{1, 2, 3\})$ within the polygon contains no point visible from the third point p_k $(k \neq i$ or $j)$.

Proof. Assume that P is a simple polygon. Let p_1, p_2 and p_3, given in counterclockwise order, denote three boundary points of P which satisfy the condition **C1**. See Fig. 2. Without loss of generality, assume that the ∞-searcher starts at p_1. To clear the next point, say, p_2, it suffices for the ∞-searcher to move within the interval $[p_1, p_2]$ (Fig. 2a). Since the shortest path between p_1 and p_2 contains no point visible from p_3, the third point p_3 remains contaminated when the ∞-searcher moves within $[p_1, p_2]$. To clear the point p_3, the ∞-searcher has to move outside of the interval $[p_1, p_2]$ at least once. However, when the ∞-searcher moves to the point p_2 (resp. p_1), the target may sneak from p_3 into p_1 (resp. p_2). See Fig. 2b for an example, where the ∞-searcher is located at the point p_2 and the cleared region is shaded. Thus, whenever the ∞-searcher moves within $[p_i, p_j]$ $(i, j \in \{1, 2, 3\})$, the third point p_k $(k \neq i$ or $j)$ is contaminated or recontaminated. Hence, P is not searchable from the boundary. $\square$

For any three points p_1, p_2 and p_3 satisfying **C1**, we can find the reflex vertices v_1, v_2 and v_3 such that three adjacent vertices of them, each per vertex v_i $(1 \leq i \leq 3)$, satisfy the condition **C1**. See Fig. 2a for an example, where p_i is just the vertex adjacent to v_i. The condition **C1** is then said to become true due to the existence of v_1, v_2 and v_3, or shortly, due to v_1, v_2 and v_3.

There are some other cases in which a cycle of recontaminations occurs (Fig. 3). We need more definitions. A pair of vertices v_1, v_2 is said to give a *BF-pair* for a boundary point p if $p_l < v_1 < Backw(v_1) < v_2 < p_r$ and $v_1 < Forw(v_2)$ hold. For the polygon shown in Fig. 3a, each boundary point has a BF-pair. A triple of vertices v_1, v_2 and v_3 is said to give an *F-triple* for the point p if $p_l \leq v_1 < Forw(v_2) < v_2 < Forw(v_3) < v_3 < p_r$ and $v_2 < Forw(v_1) < v_3$ hold. See Fig. 3b for an example. Also, a triple of vertice v_1, v_2 and v_3 is said to give a *B-triple* for the point p if $p_l < v_1 < Backw(v_1) < v_2 < Backw(v_2) < v_3 \leq p_r$ and $v_1 < Backw(v_3) < v_2$ hold.

Theorem 2. A simple polygon is not ∞-searchable from the boundary if (**C2**) either the BF-pair or the F-triple occurs for each boundary point, or if (**C3**) either the BF-pair or the B-triple occurs for each boundary point.

Proof. We give below a proof for the condtion **C2**. (The condition **C3** can be proved analogously.) Some examples satisfying **C2** are shown in Fig. 3. Let P be a simple polygon such that **C2** applies, but **C1** doesn't. To show that P is not searchable from the boundary, we distinguish the following three cases.

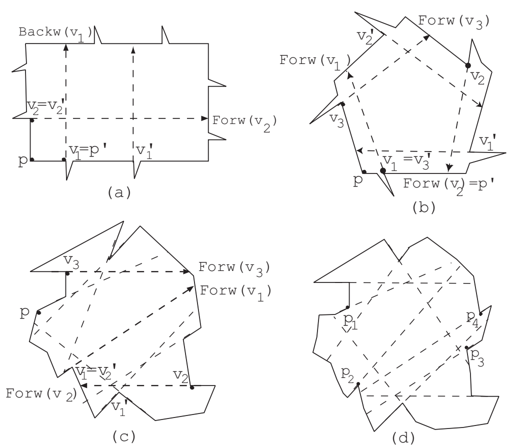

Fig. 3. Several examples satisfying the condition **C2**

Case 1. All boundary points have their BF-pairs (Fig. 3a). For a boundary point p, there are two vertices v_1 and v_2 such that $p_l < v_1 < Backw(v_1) < v_2 < p_r$ and $v_1 < Forw(v_2)$ hold. Suppose that the ∞-searcher starts at p. The ∞-searcher has to move over v_1 or v_2 at least once; otherwise, P cannot be cleared. When the ∞-searcher moves into the interior of the edge $\overline{v_1 Succ(v_1)}$ (resp. $\overline{v_2 Pred(v_2)}$) at a time, say, t, the target can sneak from $Pred(v_2)$ (resp. $Succ(v_1)$) to the point p, making p be recontaminated. Consider now the point v_1 (resp. v_2) as a new starting point p'. There are also two vertices v_1' and v_2' such that $p_l' < v_1' < Backw(v_1') < v_2' < p_r'$ and $v_1' < Forw(v_2')$ hold. Note that $Pred(v_2')$ and $Succ(v_1')$ are contaminated at the time t. Since all boundary points have their BF-pairs, the starting points considered eventually give a cycle of recontaminations. Hence, P is not searchable from the boundary.

Case 2. All boundary points have their F-triples (Fig. 3b). For a boundary point p, there are two vertices v_2 and v_3 such that $p_l < Forw(v_2) < v_2 < Forw(v_3) < v_3 < p_r$ holds. Assume that v_2 is the maximum vertex satisfying the above inequality, with respect to p. Let p' denote the point $Forw(v_2)$. There are also two vertices v_2' and v_3' such that $p_l' < Forw(v_2') < v_2' < Forw(v_3') < v_3' < p_r'$ holds. Since v_2 is the maximum vertex satisfying $p_l < Forw(v_2) < v_2 < Forw(v_3) < v_3 < p_r$, we have $v_3 \neq v_3'$. Then, $p_l' < Forw(v_3) < Forw(v_3') < v_3 < p_r'$ holds; otherwise, **C1** becomes true due to v_2, v_3 and v_3'. Let us restrict p to be a point of the (half-open) interval $[v_3, v_3')$ and let $v_1 = v_3'$. Then, for any boundary point x, there are three vertices v_1, v_2 and v_3 such that $x_l < p_1 < Forw(p_2) < p_2 < Forw(p_3) < Forw(p_1) < p_3 < x_r$ holds. This inequality gives the F-triple for the point x, and is the key to the following proof.

Suppose that the ∞-searcher starts at a point p. Both $Pred(v_2)$ and $Pred(v_3)$ are contaminated initially. Assume first that the ∞-searcher moves on the boundary of P counterclockwise. We repeatedly consider $Forw(v_2)$ for the current point p as a new starting point p'. When the ∞-searcher moves into the interior of the edge $\overline{v_2 Pred(v_2)}$, the target can sneak, say, from $Pred(v_3)$ to $Pred(v_3')$, making p be recontaminated. Since all boundary points have their F-triples, the starting points considered eventually give a cycle of recontaminations. Assume now that the ∞-searcher moves clockwise. When he moves to the interior of the edge $\overline{v_3 Pred(v_3)}$, the target can sneak from $Pred(v_2)$ to any point of the interval $[Pred(v_2), v_3)$, making p be recontaminated. Take v_3 as a new starting point. Again, the starting points considered eventually give a cycle of recontaminations. Finally, suppose that the ∞-searcher can change his moving direction. Also, we repeatedly consider $Forw(v_2)$ or v_3 for the current point p as a new starting point, depending on which one is first encountered. As discussed above, a cycle of recontaminations eventually occurs among these starting points. Hence, P is not searchable from the boundary.

Case 3. Some boundary points have the BF-pairs and the others have the F-triples. Let p denote a point, for which the F-triple occurs but the BF-pair does not. Then, there are three vertices v_1, v_2 and v_3 such that $p_l \leq v_1 < Forw(v_2) < v_2 < Forw(v_3) < v_3 < p_r$ and $v_2 < Forw(v_1) < v_3$ hold. Assume that v_1 and v_3 are the maximum and minimum vertices satisfying the above inequalities, with respect to p. Any point $x \in [v_1, v_2] \cup [Forw(v_3), v_3]$ has the BF-triple; otherwise, either all boundary points of P have their F-triples or the condition **C2** cannot be satisfied. Since v_2 and v_3 may contribute to the other F-triple, the boundary of P is divided into two or four intervals such that the BF-pairs and the F-triples appear alternately. In Fig. 3d, each point of the interval $[p_1, p_2]$ or $[p_3, p_4]$ has the F-triple, and each point of the interval $[p_2, p_3]$ or $[p_4, p_1]$ has the BF-pair. In Fig. 3c, two alternate intervals can be found.

From the discussion made in Case 1, it is ineffective for the ∞-searcher to start at a point for which the BF-pair occurs. Consider a search schedule that starts at a point p, for which only the F-triple occurs. Let us see what happens (or which points are contaminated) when the ∞-searcher moves from one interval having the F-triple to the other having the BF-pair. Suppose that the ∞-searcher moves from p to $Forw(v_2)$ counterclockwise in the time interval $[0, t]$, $0 < t$. Let p' denote the point $Forw(v_2)$. Then, there are two vertices v_1', v_2' such that $p_l' < v_1' < Backw(v_1') < v_2' < p_r'$ and $v_1' < Forw(v_2')$ hold (Fig. 3c). Since v_1 is the maximum vertex giving the F-triple for the point p, we have $v_1 = v_2'$. Note that $v_1' < v_2$ holds; otherwise, **C1** becomes true due to v_1', v_2' and v_2. Since $v_1 < Forw(v_2) < v_2 < Forw(v_1)$ holds, the vertex $Succ(v_1')$ is contaminated or recontaminated at the time t.

Let us proceed to show that P is not searchable from the boundary. Assume that all points of the interval $[v_1, v_3]$ have their BF-pairs. At first, when the ∞-searcher moves within the interval $[Pred(v_3), v_1]$, the target can hide himself at $Pred(v_2)$. When the ∞-searcher reaches a point $x \in [v_1, Pred(v_3)]$, the target can always hide himself at the successor of the first vertex of the two giving the BF-pair for x. A cycle of recontaminations occurs when the ∞-searcher moves back to the interval $[Pred(v_3), v_1]$. See Fig. 3c for an example. Assume now that there is a sub-interval of $[v_2, Forw(v_3)]$, whose points have their F-triples (see Fig. 3d). In this case, when the ∞-searcher moves within that sub-interval, the target can hide himself at $Pred(v_3)$. Also, a cycle of recontaminations eventually occurs. It completes the proof. □

3.2 Sufficiency

In this section, we show that the absence of all configurations specified by **C1**, **C2** and **C3** ensures that a polygon is 1-searchable from the boundary.

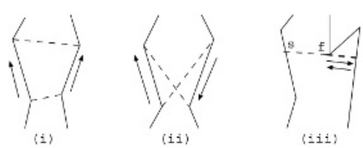

Fig. 4. Instructions for the 1-searcher in boundary search

Consider the elementary actions performed by the 1-searcher [5]. The 1-searcher s and the endpoint f of his flashlight can move along segments of single edges such that (i) no (proper) intersections occur among all segments $\overline{sf}$ during the movement or (ii) any two of the segments $\overline{sf}$ intersect each other, and (iii) f jumps from one point to another point on the boundary of P such that the ray between s and f is extended or shortened. See Fig. 4. The first two

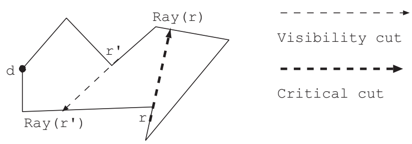

Fig. 5. Visibility cuts and critical cuts

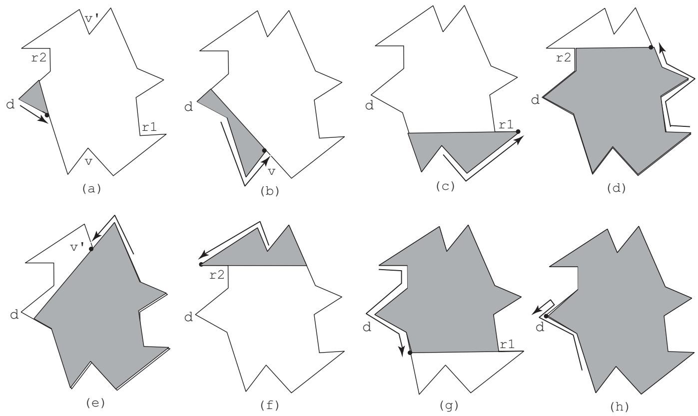

Fig. 6. Snapshots of a search schedule

instructions are allowed for two guards, but the last one is not. So any polygon that is walkable by two guards is 1-searchable from the boundary. We will refer to a *flashlight rotation* as a set of continuous instructions (ii) and (iii), including at least one instruction (ii), and a *walk* as a set of continuous instructions (i) and (ii), including at least one instruction (i).

Let us define the visibility events occurred in any search schedule starting at a boundary point d. Let r denote a reflex vertex. The polygon P can be divided into two pieces by a "cut" that extends an edge incident to r until it hits the boundary of P. A cut is a *visibility cut* if it produces a convex angle at r in the piece of P containing d. Let $Ray(r)$ denote the other endpoint of the visibility cut produced by r, and let $P(\overline{rRay(r)})$ denote the piece of P containing d. A visibility cut $\overline{rRay(r)}$ is *critical* if $P(\overline{rRay(r)})$ is not contained in any other $P(\overline{r'Ray(r')})$, where $\overline{r'Ray(r')}$ is also a visibility cut. See Fig. 5 for an example. We call the reflex vertices, whose visibility/critical cut are defined, the *visibility/critical vertices*.

Our general strategy is to clear the corners incident to critical vertices counterclockwise. But, all these corners as well as the starting point d are allowed to be recontaminated. This is the major difficulty that arises in boundary search. In order to clear the corner incident to a critical vertex, we design a simple "greedy" algorithm, i.e., a walk or a flashlight rotation is performed to the utmost limit. Fig. 6 gives an example, in which two critical vertices r_1 and r_2 are defined.

Snapshots of a search schedule are shown (the arrow shows the movement of the 1-searcher and the cleared region at each step is shaded).

Theorem 3. A simple polygon is 1-searchable from the boundary if none of **C1**, **C2** and **C3** applies.

Proof. Let P be a simple polygon, for which none of **C1**, **C2** and **C3** applies. Then, there is a boundary point d such that none of the BF-pair, the F-triple and the B-triple occurs for d. Order all boundary point of P counterclockwise, starting at d. Both inequalities $d_l < Forw(v_1) < v_1 < Forw(v_2) < v_2 < d_r$ and $d_l < v_1 < Backw(v_1) < v_2 < Backw(v_2) < d_r$ cannot hold simultaneously; otherwise, either the condition **C1** becomes true or the BF-pair for d occurs. In the following, assume that only $d_l < Forw(v_1) < v_1 < Forw(v_2) < v_2 < d_r$ may hold. (The situation in which $d_l < v_1 < Backw(v_1) < v_2 < Backw(v_2) < d_r$ holds can be dealt with analogously.)

Let m denote the number of critical vertices, and let $r_1, \ldots, r_m$ be the sequence of the critical vertices in the increasing order. Denote by $P(r_i)$ and $P - P(r_i)$ the regions which are to the left and right of the segment $\overline{r_i Ray(r_i)}$, respectively. (If d is contained in $P(r_i)$, then $P(r_i) = P(\overline{r_i Ray(r_i)})$; otherwise, $P(r_i) = P - P(\overline{r_i Ray(r_i)})$.) Our search algorithm is so designed that the clear portion of P is always to the left of the ray emanating from the flashlight, as viewed from d. To be exact, we clear the regions $P(r_i)$ in the order $i = 1, \ldots, m$ and finally the whole polygon P (Case 1), except for the situation where $d_l < Forw(v_1) < v_1 < Forw(v_2) < v_2 < d_r$ holds (Case 2).

For a walk or a flashlight rotation, we denote by $R(x1, y1)$ $(x1 \leq y1)$ and $L(x2, y2)$ $(x2 \leq y2)$ the chains on which the 1-searcher s and the endpoint f of his flashlight move, respectively. The ray of the flashlight is often denoted by $\overline{sf}$. A reflex vertex in a chain is called a *blocking* vertex if it blocks one of its adjacent vertices from being visible from any point in the opposite chain.

Case 1. The inequality $d_l < Forw(v_1) < v_1 < Forw(v_2) < v_2 < d_r$ never holds. Let $r_0 = P(r_0) = d$. We will show how to clear the region $P(r_i)$, $i \geq 1$, assuming that $P(r_{i-1})$ has been cleared. The absence of **C1** and the **BF**-pair for d is sufficient for P to be searchable from the boundary in this case.

Case 1.1. $i = 1$. Two subcases are distinguished according to whether d is contained in $P(r_1)$ or not.

Case 1.1.1. The point d is contained in $P(r_1)$. Two chains $R(d_l, r_1)$ and $L(Ray(r_1), d_r)$ are shown by bold lines in Fig. 7. They are mutually weakly visible, and there are no deadlocks between them; otherwise, there are some other critical vertices before r_1 (Fig. 7a-b), the inequality $v_1 < Backw(v_1) < v_2 < Backw(v_2)$ (Fig. 7c) or the BF-pair for d (Fig. 7d-e) holds, or the condition **C1** is true (Fig. 7f), a contradiction. Hence, the region $P(r_1)$ can be cleared by a walk from the point d to the segment $\overline{r_1 Ray(r_1)}$.

Case 1.1.2. The point d is not contained in $P(r_1)$. Assume first that there are no other visibility vertices in the interval $[d_l, r_1]$. Consider the shortest path between d and r_1. Extend all segments of this path until they hit the boundary of P. Let d' denote the other endpoint of the first extended segment (Fig. 8a).

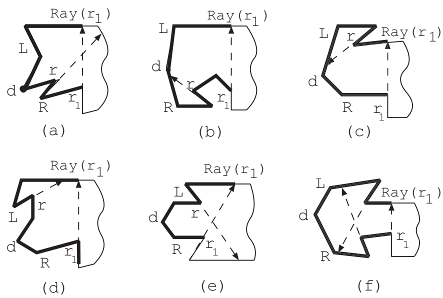

Fig. 7. Case 1.1.1

All points preceding d' are visible from d; otherwise, there are some visibility vertices in $[d_l, r_1]$, a contradiction. The region being to the left of $\overline{d'd}$ (oriented from d' to d) can then be cleared by moving the 1-searcher s from d to d', while keeping f at d (Fig. 9a). If $d' > Ray(r_1)$, then all points of the interval $[d', r_1]$ are visible from the intersection point of $\overline{d'd}$ and $\overline{r_1 Ray(r_1)}$; otherwise, there are some other critical (i.e., blocking) vertices preceding r_1 (and possibly r_1 is not critical), a contradiction. Thus, $P(r_1)$ can be cleared by rotating the flashlight from $\overline{d'd}$ to $\overline{r_1 Ray(r_1)}$. If $d' < Ray(r_1)$, the region $P(r_1)$ can analogously be cleared by rotating the flashlight from $\overline{d'd}$ to $\overline{r_1 Ray(r_1)}$, through the remaining extended segments of the shortest path between d and r_1.

Suppose now that there are some visibility vertices v' before r_1. Clearly, d is contained in these regions $P(\overline{v' Ray(v')})$. Let v denote the maximum of these visibility vertices. Observe that if we ignore (or delete) the critical vertices r whose regions $P(\overline{r Ray(r)})$ contain $P(\overline{v Ray(v)})$, the vertex v as well as some vertices preceding v become *critical*. Then as in Case 1.1.1 and Case 1.2 (see below), we can clear the region $P(\overline{v Ray(v)})$. If $v < Ray(r_1)$, the region $P(r_1)$ can be cleared by finding the shortest path between v and r_1, extending the segments of the path until they hit the boundary of P, and rotating the flashlight through every extended segment of the path (Fig. 8b). Consider the case that $v > Ray(r_1)$. If all points in the chain $R(v, r_1)$ are visible from the intersection point of $\overline{v Ray(v)}$ and $\overline{r_1 Ray(r_1)}$, the flashlight can simply be rotated from $\overline{v Ray(v)}$ to $\overline{r_1 Ray(r_1)}$. Otherwise, let $r^\star$ be the minimum vertex in $R(v, r_1)$ such that $Pred(r^\star)$ is not visible from the intersection point (Fig. 8c). The flashlight can then be rotated from $\overline{v Ray(v)}$ to $\overline{r^\star Forw(r^\star)}$. This procedure is repeatedly performed until the flashlight is rotated to $\overline{r_1 Ray(r_1)}$.

Case 1.2. $1 < i \leq m$. Two subcases are distinguished according to whether $\overline{r_{i-1} Ray(r_{i-1})}$ intersects with $\overline{r_i Ray(r_i)}$ or not.

Case 1.2.1 The segment $\overline{r_{i-1} Ray(r_{i-1})}$ intersects with $\overline{r_i Ray(r_i)}$. If the chain $R(r_{i-1}, r_i)$ is weakly visible from $L(Ray(r_{i-1}), Ray(r_i))$, the flashlight is rotated from $\overline{r_{i-1} Ray(r_{i-1})}$ to $\overline{r_i Ray(r_i)}$ as follows. If all points between r_{i-1} and r_i are visible from the intersection point of $\overline{r_{i-1} Ray(r_{i-1})}$ and $\overline{r_i Ray(r_i)}$, the flashlight can be rotated around the intersection point. Otherwise, let $r^\star$ be the minimum

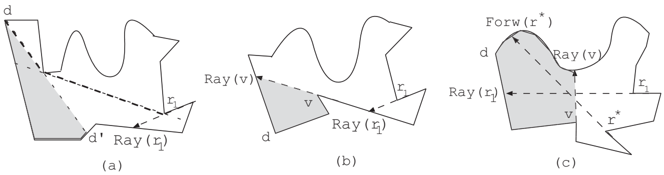

Fig. 8. Case 1.1.2

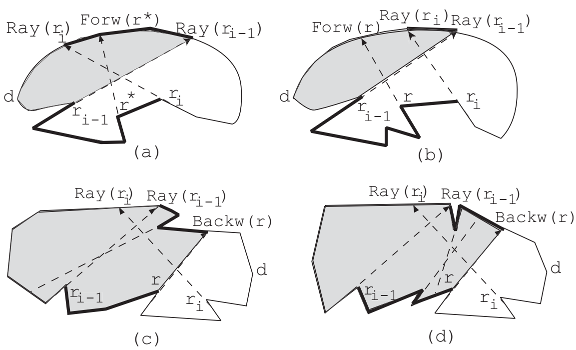

Fig. 9. Case 1.2.1

vertex in $R(r_{i-1}, r_i)$ such that $Pred(r^\star)$ (if d is contained in $P(r_{i-1})$) or $Succ(r^\star)$ (if d is not contained in $P(r_{i-1})$) is not visible from the intersection point. The flashlight can then be rotated from $\overline{r_{i-1}Ray(r_{i-1})}$ to $\overline{r^\star Ray(r^\star)}$. See Fig. 9a for an example. The chain $R(r^\star, r_i)$ is still visible from $L(Ray(r^\star), Ray(r_i))$; otherwise, $R(r_{i-1}, r_i)$ is not weakly visible from $L(Ray(r_{i-1}), Ray(r_i))$ or the blocking vertices in $R(r^\star, r_i)$ are critical, a contradiction. Hence, the flashlight can eventually be rotated to $\overline{r_iRay(r_i)}$.

Suppose now that $R(r_{i-1}, r_i)$ is not weakly visible from $L(Ray(r_{i-1}), Ray(r_i))$. Let r be the blocking vertex in $R(r_{i-1}, r_i)$, whose shot $Ray(r)$ is the furthest from $Ray(r_i)$ on the boundary of P among those of the blocking vertices. See Fig. 9b. The segment $\overline{rRay(r)}$ should exactly intersect with one of $\overline{r_{i-1}Ray(r_{i-1})}$ and $\overline{r_iRay(r_i)}$. The flashlight can first be moved to $\overline{rRay(r)}$, and then to $\overline{r_iRay(r_i)}$; one moverment is a flashlight rotation and the other is a walk. Becasue of our choice of the "furthest" shot $Ray(r)$, the flashlight rotation is always possible. The walk is also possible, since otherwise **C1** becomes true (see Fig. 9c for an example where two considered chains considered are not mutually weakly visible, and Fig. 9d for an example where a deadlock occurs).

Case 1.2.2. The segment $\overline{r_{i-1}Ray(r_{i-1})}$ does not intersect with $\overline{r_iRay(r_i)}$. Following from the definition of critical vertices and the fact that $d_l < Forw(v_1) < v_1 < Forw(v_2) < v_2 < d_r$ does not holds, the point d is contained in $P(r_i)$, but isn't in $P(r_{i-1})$. The chain $R(r_{i-1}, r_i)$ is weakly visible from $L(Ray(r_i), d_r) \cup L(d_l, Ray(r_{i-1}))$; otherwise, there are the critical vertices between r_{i-1} and r_i,

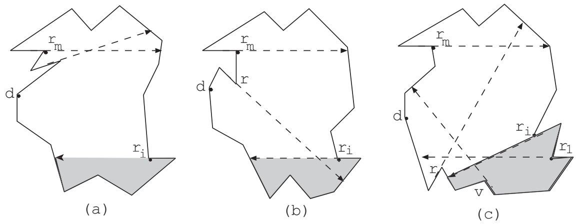

Fig. 10. Case 2.1

or the condition **C1** becomes true due to r_{i-1}, r_i and the blocking vertex in $R(r_{i-1}, r_i)$. The converse is also true; otherwise, **C1** becomes true due to r_{i-1}, r_i and the blocking vertex in $L(Ray(r_i), d_r) \cup L(d_l, Ray(r_{i-1}))$. There are no deadlocks between these two chains; otherwise, **C1** beomes true due to two vertices giving a deadlock and the vertex r_{i-1} or r_i. Hence, the flashlight can be moved from $\overline{r_{i-1}Ray(r_{i-1})}$ to $\overline{r_iRay(r_i)}$ using a walk.

Case 1.3 The region $P(r_m)$ is cleared. If we order all boundary points of P clockwise, then r_m becomes the first critical vertex, and $P - P(r_m)$ is the first region to be cleared. Thus, by a reversed operation of Case 1.1, we can clear the region $P - P(r_m)$ and obtain a complete search schedule. (Note that the 1-searcher traverses the boundary of P only once in Case 1.)

Case 2. The inequality $d_l < Forw(v_1) < v_1 < Forw(v_2) < v_2 < d_r$ holds. Following the definition of critical vertices, this inequality should be satisfied by some pair of critical vertices. The absence of the F-triple for d will be used in this case. (Symmetrically, the absence of the B-triple for d is used in the case that $d_l < v_1 < Backw(v_1) < v_2 < Backw(v_2) < d_r$ holds.)

Case 2.1. There are two consecutive critical vertices r_i and r_{i+1} such that $Forw(r_i) < r_i < Forw(r_{i+1}) < r_{i+1}$ holds. An example of Case 2.1 can be found in Fig. 6, where a complete search schedule is shown. In this case, $Forw(r_h) < Forw(r_i) < r_h < r_i$ holds for $1 \leq h < i$; otherwise, r_h and r_{i+1} give the BF-pair for d (if $Ray(r_h) = Backw(r_h)$), a contradiction. So the segment $\overline{r_hRay(r_h)}$ intersects with $\overline{r_{h+1}Ray(r_{h+1})}$, for $1 \leq h < i$. As in Case 1.1.2 (for $P(r_1)$) and Case 1.2.1, the region $P(r_i)$ can be cleared. The inequality $Forw(r_i) < r_i < Forw(r_{i+1}) < r_{i+1}$ does not affect the operations performed in Case 1.1.2 and Case 1.2.1.

It is difficult to clear the next region $P(r_{i+1})$, as $P(r_i)$ is completely separated from $P(r_{i+1})$. But, we can directly clear the region $P - P(r_m)$ at present time. Note that $Forw(r_{i+1}) < Forw(r_j) < r_{i+1} < r_j$ holds for $i + 1 < j \leq m$; otherwise, **C1** becomes true due to r_i, r_{i+1} and r_j. Thus, the segment $\overline{r_iRay(r_i)}$ does not intersect with $\overline{r_mRay(r_m)}$. Two chains $R(r_i, Ray(r_m))$ and $L(r_m, d_r) \cup L(d_l, Ray(r_i))$ are mutually weakly visible; otherwise, there are the critical vertices between r_i and $Ray(r_m)(< r_{i+1})$, preceding $Ray(r_i)$ $(< r_1)$ or succeeding r_m (Fig. 10a), or r_i, r_m and the blocking vertex r in $L(r_m, d_r) \cup L(d_l, Ray(r_i))$ make **C1** (Fig. 10b) be true or give the F-triple for d (Fig. 10c). There are no deadlocks between these two chains; otherwise, **C1** becomes true due to two vertices giving a deadlock and the vertex r_i or r_m. The region $P - P(r_m)$ can then be cleared using a walk from $\overline{r_i Ray(r_i)}$ to $\overline{Ray(r_m) r_m}$. Next, we clear the region $P(r_m)$ by finding the shortest path between r_m and d, extending the segments of the path, and rotating the extended segments intersecting $\overline{r_m Ray(r_m)}$. (Since $P - P(r_m)$ is already cleared, the flashlight has to be rotated only through the extended segments intersecting $\overline{r_m Ray(r_m)}$.) And, move back the flashlight from $\overline{r_m Ray(r_m)}$ to $\overline{Ray(r_i) r_i}$ using a walk so that the region $P - P(r_i)$ is cleared. Note that the 1-searcher s is now located at $Ray(r_i)$. It is important to see that no instructions (iii) are used in the work of clearing the region $P(r_h)$ (Case 1.1.2 and Case 1.2.1), $1 \le h \le i$; otherwise, there is a vertex r ($< Ray(r_h)$) such that r, r_h (e.g., r_i in Fig. 10c) and r_m give the F-triple for d, a contradiction. So the operation done for clearing $P(r_i)$ can *reversely* be performed, even in the sense that the roles of the 1-searcher s and the endpoint f of the flashlight are exchanged. It completes the search schedule for clearing P.

Case 2.2. No two of consecutive critical vertices r and r' satisfy the inequality $Forw(r) < r < Forw(r') < r'$. See Fig. 11 for some examples. Without loss of generality, assume that there are two critical vertices r_i and r_j satisfying the inequality $Forw(r_i) < r_i < Forw(r_j) < r_j$, $i + 1 < j$. As discussed in Case 2.1, $Forw(r_h) < Forw(r_i) < r_h < r_i$ holds for $1 \le h < i$, and $Forw(r_j) < Forw(r_l) < r_j < r_l$ holds for $j < l \le m$. Then, as in Cases 1.1.2 and 1.2.1, the region $P(r_m)$ can be cleared. Assume that r_k is the maximum among the critical vertices r satisfying $Forw(r) < r < Forw(r_m) < r_m$. So we have $Ray(r_{k+1}) = Backw(r_{k+1})$ and $r_{k+1} > Forw(r_m)$. By an argument similar to that made in Case 2.1, we can show that the flashlight can be moved from $\overline{r_m Ray(r_m)}$ to $\overline{Ray(r_k) r_k}$ using a walk. This clears the region $P - P(r_k)$. Finally, as shown in Case 2.1, the operation of clearing $P(r_k)$ can reversely be performed. It completes the search schedule for clearing P.

All cases above ensure that P is 1-searchable from the boundary. □

Theorem 4. Any polygon that is ∞-searchable from the boundary is also 1-searchable from the boundary.

Proof. It immediately follows from Theorems 1, 2 and 3. □

4 Algorithm and Complexity

In this section, we give the algorithms for testing the searchability of simple polygons, and reporting a search schedule if it exists.

Theorem 5. It takes $O(n \log n)$ time and $O(n)$ space to determine the searchability of simple polygons.

Proof. Let P be a simple polygon. All ray shots can be computed in $O(n \log n)$ time [1]. In the following, we first present a procedure for finding the vertices of P for which the BF-pairs occur, and then extend it to find the vertices for

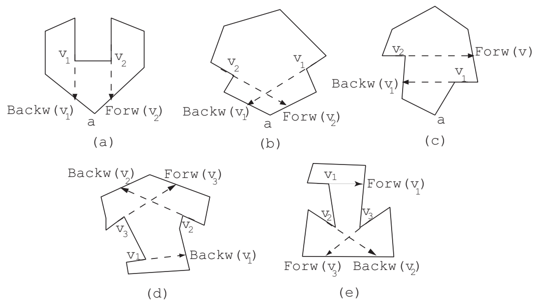

Fig. 11. Illustration of the proof of Theorem 5

which the F-triples (resp. B-triples) occur. Whether or not the condition **C2** (resp. **C3**) is true for P can then be determined from these computed results. Also, the condition **C1** can similarly be verified.

Let a denote an arbitrary vertex of P. Order all vertices and ray shots counterclockwise, starting at a. Let v_1 denote the minimum vertex such that $v_1 < Backw(v_1)$ holds, and v_2 the maximum vertex such that $Forw(v_2) < v_2$ holds. If only one of v_1 and v_2 is found, no BF-pairs occur for the polygon P and we are done. If $v_1 > v_2$ holds, the BF-pairs occur only for the vertices between v_1 and v_2 (Fig. 11a), and we are done. If $Forw(v_2) < v_1 < v_2 < Backw(v_1)$ holds, neither v_1 nor v_2 can contribute to a BF-pair for a (Fig. 11b). In this case, we search for the vertex v_1' next to v_1 such that $v_1' < Backw(v_1')$ holds and the vertex v_2' next to v_2 such that $Forw(v_2) < v_2$ holds, and then call the same procedure to test if v_1' and v_2' give a BF-pair for a. If $v_1 < Forw(v_2) < v_2 < Backw(v_1)$ holds, the region $P(\overline{v_1 Backw(v_1)})$ (containing the point a) is contained in $P(\overline{v_2 Forw(v_2)})$. See Fig. 11c for an example. (The situation in which $Forw(v_2) < v_1 < Backw(v_1) < v_2$ holds can be dealt with analogously.) Then, v_1 cannot contribute to any BF-pair for a. We search for the vertex v_1' next to v_1 such that $v_1' < Backw(v_1')$ holds, and call the testing procedure with the new pair of v_1' and v_2. If $v_1 < Backw(v_1) < v_2$ and $v_1 < Forw(v_2) < v_2$ hold, a BF-pair occurs for all vertices of $[v_2, a_r] \cup [a_l, v_1]$. Next, take $Succ(v_1)$ as the new starting point a', and order all critical vertices and their shots. Since it is equivalent to take a_l' and a_r' as the minimum and maximum points respectively, this ordering can be obtained in constant time. Then search for the minimum vertex v_1' such that $v_1' < Backw(v_1')$ holds, and call the same procedure to test if v_1' and v_2 give the BF-pair for a'. This procedure is terminated when the BF-pair for a is verified again. Clearly, the time taken to find the vertices of P having the BF-pairs is $O(n)$.

In order to compute the vertices of P having the F-triple, we find, for each vertex a, two vertices v_0, v_1 such that v_1 is the minimum vertex satisfying $v_0 < Forw(v_1) < v_1 < Forw(v_0)$, and two vertices v_1', v_2 such that v_1' is the maximum vertex satisfying $Forw(v_1') < v_1' < Forw(v_2) < v_2$. If $v_1 \leq v_1'$, then v_0, v_1 and v_2 give an F-triple for a; otherwise, no F-triples occur for a. By an argument similar to that for computing the vertices of P having the B-triples, we can find, in $O(n)$ time, the vertices of P for which the F-triples (resp. B) occur. We leave the detail to readers.

Turn to the condition **C1**. Using Das et al.'s algorithm [2], we can find in linear time if there are three *critical vertices* (all boundary points are considered as the starting point once) such that no intersections occur among three segments connecting a critical vertex with its ray shot. If *yes*, the condition **C1** is true. The remaining situations in which **C1** applies are shown in Fig. 11d-e. By an argument similar to that for finding an F-triple or a B-triple, we can verify in $O(n)$ time if the situation shown in Fig. 11d or Fig. 11e occurs or not.

Finally, the space requirement of our algorithm is $O(n)$. □

Theorem 6. A search schedule can be reported in time linear in its size I, if it exists. For the ∞-searcher, $I < 2n$, and for the 1-searcher, $I < 3n^2$.

Proof. Let P be a simple polygon, for which none of **C1**, **C2** and **C3** applies. Then, there is a boundary point d such that at most one of $Forw(v_1) < v_1 < Forw(v_2) < v_2$ and $v_1 < Backw(v_1) < v_2 < Backw(v_2)$ holds, and no BF-pairs occur for d. To obtain a search schedule for the 1-searcher, we run the constructive algorithm presented in the proof of Theorem 3. Clearing a region $P(r_i)$ consists of a number of flashlight rotations and walks. If we consider the polygonal chain, which is traversed by the 1-searcher for the second or third time, as a new different chain, the chains $R(x, y)$ considered for all walks and for all flashlight rotations as well are disjoint. Since the total size of these chains is equal to n in Case 1, and smaller than $3n$ in Case 2, the number I of search instructions output is smaller than $3n^2$ (see also [4]).

Consider now the size of search schedules for the ∞-searcher. We directly apply the search algorithm given in the proof of Theorem 3 to the ∞-searcher. In Case 1, a complete search schedule is obtained before or when the ∞-searcher returns to d. In Case 2.1, we claim that a complete search schedule is obtained when or before the ∞-searcher moves to the point $Ray(r_i)$ for the second time. Consider the walk from $\overline{r_m Ray(r_m)}$ to $\overline{Ray(r_i) r_i}$, which is performed after $P(r_m)$ is cleared. The movement of the ∞-searcher along the chain $R(r_m, d_r) \cup R(d_l, Ray(r_i))$ clears the chain $L(r_i, Ray(r_m))$ in Case 2.1. Now, we show that the remaining chain from $Ray(r_i)$ to r_i, denoted by $L'(Ray(r_i), r_i)$, is also cleared by this movement of the ∞-searcher. Assume that there are no visibility vertices in the interval $[r_m, d_r]$; otherwise, the flashlight can be moved to the maximum of these visibility vertices and its (forward) ray shot, using a walk. Then, two chains $L'(Ray(r_i), r_i)$ and $R(r_m, d_r) \cup R(d_l, Ray(r_i))$ are mutually weakly visible; otherwise, r_m and the blocking vertex in $L'(Ray(r_i), r_i)$ give the BF-pair for the point d, or r_i, r_m and the blocking vertex in $R(d_l, Ray(r_i))$ give the F-triple for d, a contradiction. Thus, any point of $L'(Ray(r_i), r_i)$ has to be illuminated once during the movement of the ∞-searcher from r_m to $Ray(r_i)$, and any clear point can never be recontaminated. Our claim is proved. Also, in Case 2.2, a complete search schedule is obtained when or before the ∞-searcher moves to the point $Ray(r_k)$ for the second time. Hence, we have $I < 2n$. □

References

1. B.Chazelle and L.J.Guibas, Visibility and intersection problems in plane geometry, *Disc. Comput. Geom.* **4** (1989) 551-581.
2. G.Das, P.J.Heffernan and G.Narasimhan, LR-visibility in polygons, *Comput. Geom. Theory Appl.* **7** (1997) 37-57.
3. P.J.Heffernan, An optimal algorithm for the two-guard problem, *IJCGA* **6** (1996) 15-44.
4. C. Icking and R. Klein, The two guards problem, *IJCGA* **2** (1992) 257-285.
5. S.M.LaValle, B.Simov and G.Slutzki, An algorithm for searching a polygonal region with a flashlight, *IJCGA* **12** (2002) 87-113.
6. I.Suzuki and M.Yamashita, Searching for mobile intruders in a polygonal region, *SIAM J. Comp.* **21** (1992) 863-888.
7. I.Suzuki, Y.Tazoe, M.Yamashita and T.Kameda, Searching a polygonal region from the boundary, *IJCGA* **11** (2001) 529-553.