# Searching a Polygonal Region from the Boundary*

Ichiro Suzuki†
Yuichi Tazoe‡
Masafumi Yamashita§
Tiko Kameda¶

Technical Report TR-20000925
EECS Department
University of Wisconsin–Milwaukee
September 25, 2000

**Abstract**

Polygon search is the problem of finding unpredictable mobile intruders in a polygonal region using one or more mobile searchers with various levels of vision, where both the searcher and intruders are allowed to move freely within the region. In this paper we consider a variant of this problem, termed boundary search, in which a single searcher has to find the intruders from the boundary of the region. Our main result is that the searcher having one flashlight whose vision is limited to a single ray is just as capable as the searcher having a light bulb that gives 360° vision, that is, any polygon that can be searched by the latter from the boundary can also be searched by the former from the boundary. To our knowledge, the equivalence of these searchers, one having a very limited vision and another having the maximum vision, has never been established for any interesting subclass of the polygon search problem. The proof of the equivalence uses another new result, termed Monotonic Extension Theorem, together with a simple topological argument on a 2-dimensional map called p-map that represents the searcher's state during search. A similar topological argument is used also to partially settle a long-standing conjecture on

*This work was supported in part by a Scientific Research Grant-in-Aid from the Ministry of Education, Science and Culture of Japan.

†Department of Electrical Engineering and Computer Science, University of Wisconsin – Milwaukee, P.O. Box 784, Milwaukee, WI 53201, U.S.A. suzuki@blatz.cs.uwm.edu

‡Department of Computer Science and Communication Engineering, Kyushu University, Hakozaki, Higashi-ku, Fukuoka, 812-8581 Japan. tazoe@csce.kyushu-u.ac.jp

§Department of Computer Science and Communication Engineering, Kyushu University, Hakozaki, Higashi-ku, Fukuoka, 812-8581 Japan. mak@csce.kyushu-u.ac.jp

¶School of Computing Science, Simon Fraser University, Burnaby, B.C. V5A 1S6, Canada. tiko@cs.sfu.ca

the equivalence of the abilities of two searchers, one having two flashlights and another having full 360° vision, for the general, non-boundary polygon search problem.

# 1 Introduction

Polygon search is the problem of finding unpredictable mobile intruders in a polygonal region using one or more mobile searchers of various levels of vision. Both the searcher and intruder are a point that is allowed to move continuously and freely within the region, where the latter can move arbitrarily faster than the former. Depending on the vision, the searcher can be the $k$-searcher whose vision is limited to $k$ rays emanating from her position (the case of $k$ flashlights) for $k \geq 1$, or the $\infty$-searcher having full 360° vision (the case of a light bulb). The problem was first discussed in [13] as a dynamic version of the well-known art-gallery problem [11].

The following two factors seem to have contributed to the recent outburst of papers [3] [6] [7] [9] [10] [14] [15] [16] on polygon search and its variants in both computational geometry conferences and robotics conferences:

1. Despite its seeming simplicity, the problem has proven quite challenging. Indeed, the algorithm given in [3] for computing a schedule of the $\infty$-searcher to search a given polygon through state space enumeration has exponential worst case running time, and the exact complexity of the problem of generating a schedule to clear a given polygon by one $\infty$-searcher is still unknown. The capabilities of various searchers are not well-understood either — while it is known that the 1-searcher (having one ray) is strictly less capable than the 2-searcher (having two rays), a conjecture given in [13], that any polygon searchable by the $\infty$-searcher is also searchable by the 2-searcher, has not been settled.

2. Recent advances in mobile robots and distributed robotics have made it a realistic goal to use robots in surveillance, security enforcement, target detection and tracking. The polygon search problem neatly captures the key issue in these applications, namely, motion generation for robots to cope with an unpredictable target.

Due to the difficulty of the original problem, a number of special cases have been discussed in the literature. Regarding the conjecture mentioned above, the equivalence of the capabilities of the $\infty$-searcher and the 2-searcher has been established for the case the region to be searched is a "corridor" (a polygon with two exits, of which one is to remain clear during the search, and the other contaminated till just before the search is over) [1], and for the case the region is a "room" (a polygon with one exit that must remain clear at all times) [10]. (Intuitively, an area is considered "clear" during search if it cannot contain an undetected intruder; otherwise it is "contaminated." See Section 2 for details.) In both cases a polynomial time algorithm, $O(n^2 \log n)$ for a corridor and $O(n^2)$ for a room, for generating a schedule of the searcher to clear an $n$-sided polygon is presented. Another $O(n^2)$ time algorithm for searching a room using the 1-searcher is found in [9]. We already mentioned the exponential time algorithm for generating a search path for the $\infty$-searcher [3], and an algorithm based on a similar idea for a curved environment appears in [6]. Recently an $O(n^2)$ time algorithm was reported for the 1-searcher in an $n$-sided polygon [7]. Since sometimes a given polygon cannot be searched by a single searcher, attempts have been made to compute the the minimum required number of searchers using various parameters of the polygon, including the "bushiness" and the number of reflex vertices [14] [16]. Other related problems include the two-guard problem [4] [5] [15], and the searchlight problem [12].

In this paper we introduce a variant of the problem, termed *boundary search*, in which a single searcher has to find the intruders in a polygonal region from the boundary of the region, without entering the interior. This is akin to searching a cage in a zoo from outside, or searching an office space surrounded by glass walls from outside. Our main result, in striking contrast to the general (non-boundary) case, shows that the 1-searcher whose vision is limited to a single ray is just as capable as the $\infty$-searcher having 360° vision, that is, any polygon that can be searched by the latter from the boundary can also be searched by the former from the boundary. To our knowledge, the equivalence of these searchers, one having a very limited vision and another having the maximum vision, has never been established for any interesting subclass of the polygon search problem. The proof of the equivalence uses another new result, termed Monotonic Extension Theorem, and its corollary that shows the $\infty$-searcher need only walk in one fixed direction to search a polygon from its boundary.

To obtain the results, we introduce a 2-dimensional map called the p-map, that describes how the invisible regions of the polygon boundary changes over time during the search by the $\infty$-searcher. We then analyze this map and use a simple topological argument to derive a schedule of the 1-searcher for clearing the polygon.

A similar topological argument is used also to identify a special case in the general (non-boundary) polygon search problem in which a polygon searchable by the $\infty$-searcher turns out to be searchable by the 2-searcher. This result partially settles the long-standing conjecture [13] mentioned earlier.

We formalize the polygon search problem in Section 2, and present the Monotonic Extension Theorem in Section 3. The results on boundary search by the $\infty$-searcher are given in Section 4. Section 5 discusses the main result regarding the capabilities of the $\infty$-searcher and the 1-searcher in boundary search. Concluding remarks are found in Section 6.

# 2 Preliminaries

Let $P$ be a polygonal region whose boundary, denoted $\partial P$, is a simple polygon. Two points $p$ and $q \in P$ are said to be mutually *visible* if $\overline{pq} \subseteq P$. (This means one can see through a reflex vertex of $P$.) We denote by $V(p)$ the set of points in $P$ that are visible from $p \in P$.

Both the searcher and intruder are a point that can move continuously in $P$. The searcher can be either the *$k$-searcher* ($k \geq 1$) whose vision is limited to $k$ rays emanating from its position, or the *$\infty$-searcher* having 360° vision.

Formally, a *schedule* of the $k$-searcher is a $(k+1)$-tuple $S = (\alpha, \beta_1, \ldots, \beta_k)$ of continuous functions, where $\alpha : [0, \infty) \to P$ is the position of the searcher at time $t$, and for each $1 \leq i \leq k$, $\beta_i : [0, \infty) \to \mathcal{R}$ is the direction of the $i$-th ray, which is the maximal line segment in $P$ emanating from $S(t)$ ($\mathcal{R}$ is the set of reals). A point $p$ is said to be *illuminated* at time $t$ if and only if $p$ lies on one of the rays. We call $S(t) = (\alpha(t), \beta_1(t), \ldots, \beta_k(t))$ the *configuration* of the searcher at time $t$.

A schedule of the $\infty$-searcher is simply $S = (\alpha)$, where $\alpha : [0, \infty) \to P$ is a continuous function such that $\alpha(t)$ is the position of the searcher at time $t$. For this searcher a point $p$ is said to be illuminated at time $t$ if and only if $p \in V(S(t))$. The configuration $S(t) = (\alpha(t))$ is simply the position of the searcher at time $t$.

During the execution of a schedule a point $p \in P$ is said to be *contaminated* at time $t \geq 0$ if there exists a continuous function $I : [0, \infty) \to P$ such that $I(t) = p$, and that $I(t')$ is not illuminated at any time $t' \in [0, t]$. Intuitively, $I$ is the motion of an intruder and $p$ is contaminated at $t$ if and only if an intruder who has never been "detected" by the searcher can be located at $p$ at time $t$. A point that is not contaminated is said to be *clear*. A region $R \subseteq P$ is contaminated if it contains a contaminated point; otherwise, it is clear. A schedule $S$ is said to *clear* $P$ if $P$ becomes clear at some time $t \geq 0$ during the execution.

Fig. 1 shows the motion of the $\infty$-searcher and the 1-searcher during a search. In both examples the searchers stay on the polygon boundary. The polygon shown in Fig. 2 can also be searched from the polygon boundary by either of the $\infty$-searcher and the 1-searcher.

In this paper we use only *one* searcher to search $P$. Thus at any time during the search every maximal subregion of $P$ not illuminated by the searcher contains a single nonempty maximal open section of $\partial P$, and consequently clearing $P$ is equivalent to clearing $\partial P$. So without loss of generality, in the following we assume that the intruder moves only along $\partial P$, i.e., any intruder is a function $I : [0, \infty) \to \partial P$.

At any time during the search by any searcher, we call each maximal section of $\partial P$ not illuminated by the searcher a *pocket* (Fig. 3). Note that for the $\infty$-searcher, the pockets are exactly those sections of $\partial P$ that are not visible from her position. However, since pockets are defined in terms of *illumination* rather than visibility, for *other* searchers pockets usually include additional points that are visible but not illuminated.

# 3 Monotonic Extension Theorem

A schedule $S = (\alpha, \beta_1, \ldots, \beta_k)$ of the $k$-searcher can be viewed as a continuous path (parameterized by time) within a $(k+2)$-dimensional *configuration space* $\mathcal{C}$, where we use two dimensions to specify the position $\alpha(t)$ and the remaining $k$ to specify the directions of the rays. Likewise, a schedule $S = (\alpha)$ of the $\infty$-searcher defines a

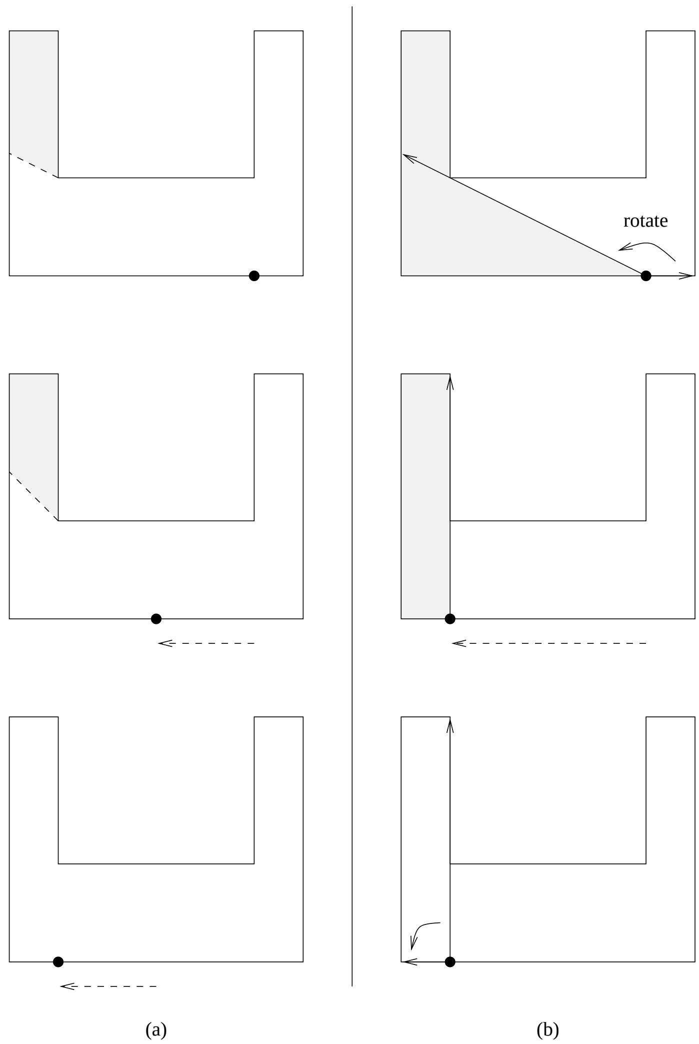



Figure 1: Search by (a) the $\infty$-searcher, and (2) the 1-searcher. The shaded areas are contaminated.

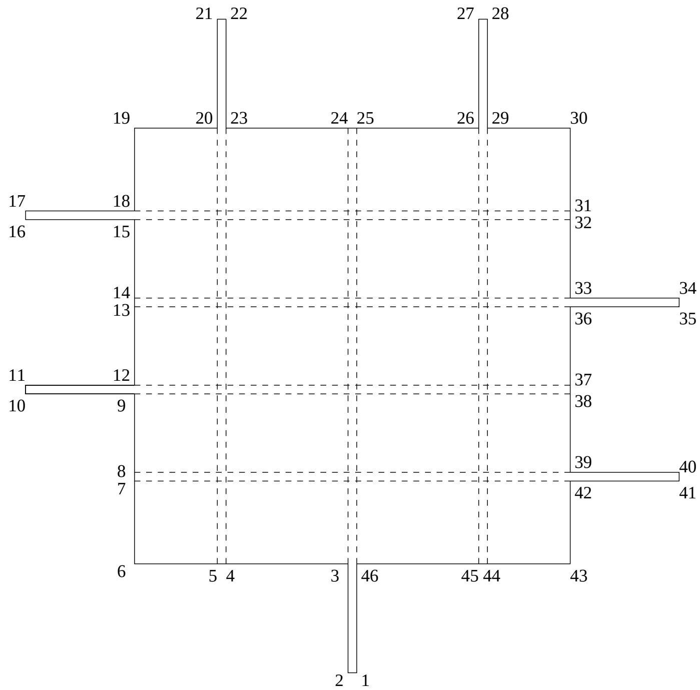



Figure 2: A polygon searchable by the ∞-searcher who traverses the boundary once, clockwise from point 1. It is also searchable by the 1-searcher who moves along the polygon boundary.

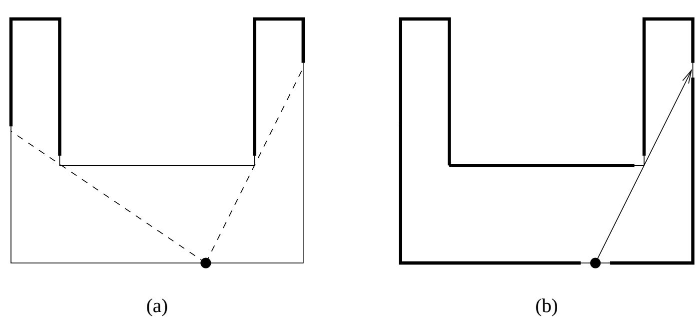



Figure 3: Bold lines show the pockets for (a) the ∞-searcher and (b) the 1-searcher.

continuous path in a two-dimensional configuration space, which is simply the plane containing $P$.

Conversely, given a continuous path in the configuration space $\mathcal{C}$ we can construct a schedule $S$ by arbitrarily parameterizing it by time, so that $S(t)$ traces the path as $t$ goes from 0 to infinity. Formally, let $C : [0,1] \to \mathcal{C}$ be a closed continuous path (i.e., loop) in $\mathcal{C}$, where $C(0) = C(1)$. Define an infinite continuous path $C^\infty$ in $\mathcal{C}$ that traces $C$ infinitely many times, by $C^\infty(u) = C(u - \lfloor u \rfloor)$ for any $u \in \mathcal{R}$. Now, for any configuration $c$ in $C$, we call a schedule $S$ an *extension* of $(C, c)$ if $S(0) = c$ and $S(t) = C^\infty(\gamma(t))$ for some continuous function $\gamma : [0, \infty) \to \mathcal{R}$. Furthermore if $\gamma$ is either monotonically increasing and unbounded, or monotonically decreasing and unbounded, then $S$ is said to be a *monotonic extension* of $(C, c)$.

See Fig. 4 for a schematic illustration for the case of the 1-searcher. Intuitively, if $S$ is an extension of $(C, c)$, then executing $S$ corresponds to tracing $C$ freely starting at $c$ in $\mathcal{C}$. In contrast, if $S$ is a *monotonic* extension, then the searcher's configuration has to change along $C$ indefinitely *in one direction only*. However, the position of the searcher whose schedule is a monotonic extension of $(C, c)$ does not necessarily change monotonically along some trajectory within $P$. It is $C$ that we trace monotonically, a path in the configuration space that specifies *both* the location of the searcher and the directions of the rays. Besides, if $C$ itself has self-intersections or self-overlaps, then certainly the searcher's motion may not "look" monotonic.

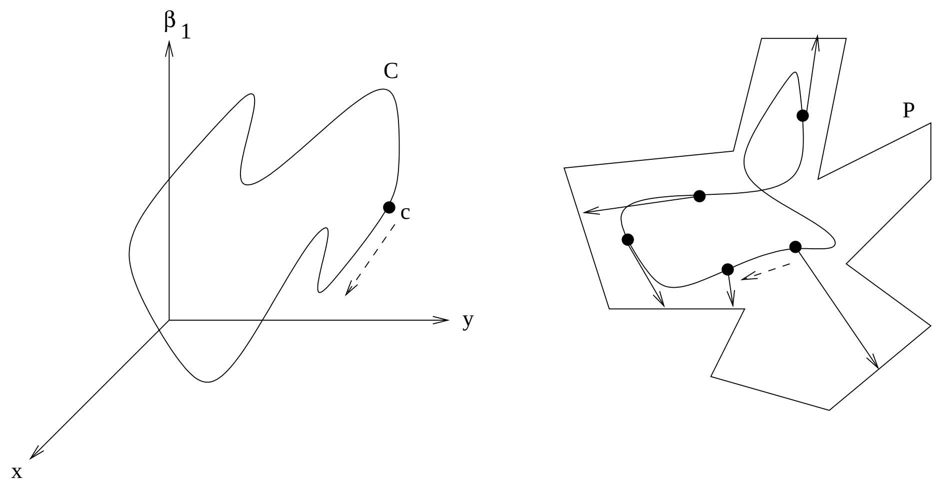



Figure 4: A loop $C$ in the configuration space and the motion of the 1-searcher executing a monotonic extension of $(C, c)$.

The following theorem states that if the searcher's configuration is restricted to change along a loop $C$ in $\mathcal{C}$, and if indeed it is possible to search the given polygon under that restriction, then it suffices for the searcher to trace the loop in one direction only.

**Theorem 1** [**Monotonic Extension Theorem**] *Let $C$ be a closed path in $\mathcal{C}$, and $S_0$ a point on $C$. Let $S$ be a monotonic extension of $(C, S_0)$ (so $S(0) = S_0$). If $S$ does* *not clear $P$, then $P$ cannot be cleared by any extension $S'$ of $(C, S_0)$.*

**Proof** Suppose $S$ does not clear $P$, and let $I : [0, \infty) \to \partial P$ be an intruder that remains undetected. Let $R_1, \ldots, R_N$ be the pockets when the searcher is in configuration $S_0$. When $S$ is executed the searcher returns to $S_0$ infinitely many times, and each time she does so, the intruder has to be in one of the pockets. So the intruder has to return to some pocket $R_h$, $1 \leq h \leq N$, infinitely many times. Let $t_1$ and $t_2$, $0 \leq t_1 < t_2$, be time instants such that $I(t_1) \in R_h$, $I(t_2) \in R_h$, and the searcher goes from $S(t_1) = S_0$ to $S(t_2) = S_0$ traversing $C$ $K \geq 1$ times. We can now choose a point $q_0 \in R_h$ and modify $I$ slightly, so that $I(t_1) = I(t_2) = q_0$. Henceforth $I$ refers to this revised undetectable intruder.

Let $[C_1 \cdots C_K]$ denote the concatenation of $K$ indexed copies of $C$ that the searcher traverses between $t_1$ and $t_2$, where the indices allow us to refer to a configuration $s \in C$ as $s_i$ if it is visited during the $i$-th traversal $C_i$ of $C$. Let us define the following "hiding" function $H : [C_1 \cdots C_K] \to \partial P$ that gives the location of intruder $I$ for every configuration of the searcher during the traversal of $[C_1 \cdots C_K]$, i.e., for every $s_i \in C_i$ (where $s \in C$), $H(s_i) = I(t)$ if $s_i = S(t)$, $t \in [t_1, t_2]$. (Function $H$ is well-defined since the searcher traverses $[C_1 \cdots C_K]$ continuously in one direction.) Note that $H$ is continuous (since $I$ is). Furthermore, since $I(t)$ is never illuminated while the searcher traverses $[C_1 \cdots C_K]$, if $H(x) = y$ then $y$ is not illuminated by the searcher in configuration $x \in [C_1 \cdots C_K]$. Note also that at both ends of $[C_1 \cdots C_K]$ the searcher is in configuration $S_0$ and the intruder is at $H(S_0) = q_0$.

Using this $H$, given any extension $S'$ of $(C, S_0)$, we can construct an undetectable intruder $I' : [0, \infty) \to \partial P$ as follows. First consider an infinite sequence of copies of $C$ and partition the copies into indexed sections of size $K$ each as shown below, where each section $[C_1 \cdots C_K]$ contains $K$ indexed copies of $C$:

$$\cdots [C_1 \cdots C_K]_{-2}\ [C_1 \cdots C_K]_{-1}\ [C_1 \cdots C_K]_0\ [C_1 \cdots C_K]_1\ [C_1 \cdots C_K]_2 \cdots$$

We assume that the initial configuration $S'(0)$ of the searcher is within the first copy $C_1$ in section $[C_1 \cdots C_K]_0$, and place the intruder at $I'(0) = H(S'(0))$ so that the intruder is not illuminated at time 0. When the searcher moves along $C$ and stays within $[C_1 \cdots C_K]_0$, the intruder moves by $I'(t) = H(S'(t))$. This ensures that $I'$ can never be illuminated. Since at both ends of $[C_1 \cdots C_K]_0$ the searcher is in configuration $S_0$ and the intruder is at position $q_0$, when the searcher moves to an adjacent section $[C_1 \cdots C_K]_{-1}$ or $[C_1 \cdots C_K]_1$, the intruder can still move continuously and undetectably by $I'(t) = H(S'(t))$, as long as the searcher stays within that section. By repeating this strategy each time the searcher moves from one section to another, we can keep $I'$ undetected indefinitely regardless of the motion of the searcher along $C$. $\square$

An argument similar to, but much simpler than, that in the proof of Theorem 1 can be used to show that, if the searcher's configuration is restricted to change along a continuous finite *path* $D : [0, 1] \to \mathcal{C}$ (in the sense that the path is not considered to "wrap around" even if $D(0)$ happens to coincide with $D(1)$), and if indeed it is possible to search the given polygon under that restriction, then it suffices for the searcher to trace $D$ in one direction only, either from $D(0)$ to $D(1)$, or from $D(1)$ to $D(0)$. We leave a formalization and a proof of this statement to the reader.

A schedule of any searcher for $P$ is called a *boundary schedule* if the searcher always stays within the polygon boundary $\partial P$. A boundary schedule is said to be *one-way* if the searcher continues to traverse the entire $\partial P$ indefinitely in only one direction, either clockwise or counterclockwise. Each of the schedules shown in Fig. 1 is an initial finite segment of a one-way boundary schedule, and both the $\infty$-searcher and the 1-searcher can clear the polygon shown in Fig. 2 by a one-way boundary schedule. In the rest of this paper we discuss mainly a subclass of the polygon search problem, termed *boundary search*, in which the objective is to search a region using a boundary schedule.

# 4 Boundary Search by the $\infty$-Searcher

For some searchers one-way boundary search is not always possible, even if the polygon itself is searchable from the boundary (Fig. 5). For the case of the $\infty$-searcher, however, as the next theorem shows moving in one way is always sufficient to search $P$ from the boundary, if $P$ is indeed searchable from the boundary.

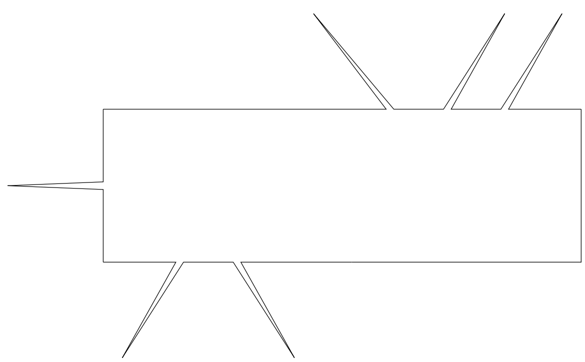


Figure 5: A polygon that is searchable by a boundary schedule of the 1-searcher, but not by any one-way boundary schedule of that searcher. The $\infty$-searcher can clear it by a one-way boundary schedule.

**Theorem 2** *Let $S$ be a one-way boundary schedule of the $\infty$-searcher for $P$. If $S$ does not clear $P$, then $P$ cannot be cleared by any boundary schedule of the $\infty$-searcher.*

**Proof** For the case of the $\infty$-searcher one complete (clockwise or counterclockwise) traversal of $\partial P$ is a closed path $C$ in the corresponding configuration space $\mathcal{C}$, and hence $S$ is a monotonic extension of $(C, S(0))$. Since $S$ does not clear $P$, by Theorem 1 $P$ cannot be cleared no matter how the searcher moves within $\partial P$. $\square$

As a corollary, we obtain:

**Theorem 3** *$P$ can be cleared by a clockwise one-way boundary schedule of the $\infty$-searcher if and only if it can be cleared by a counterclockwise one-way boundary schedule of the $\infty$-searcher.*

## 5 Equivalence of the $\infty$-Searcher and the 1-Searcher

In this section we show that if $P$ can be cleared by a boundary schedule of the $\infty$-searcher, then it can be cleared also by a boundary schedule of the 1-searcher. To our knowledge, the equivalence of the $\infty$-searcher and the 1-searcher has never been established for any interesting subclass of the polygon search problem.

Suppose that the $\infty$-searcher can clear $\partial P$ (and hence $P$) by a clockwise one-way boundary schedule starting at $p_0 \in \partial P$. If there is a position in $\partial P$ for which there are no pockets, then $P$ is star-shaped [8] and is trivially searchable by the 1-searcher. (We can clear $P$ by placing the 1-searcher at that position and sweeping $\partial P$ once by the ray.) So in the following we assume that there exists at least one pocket at any time during the search.

During the search, pockets emerge, disappear, merge and split as the searcher traverses $\partial P$. To describe these events together with the locations of the pockets, we introduce a parallelogram map called *p-map*, shown in Fig. 6, whose top, bottom, left and right edges are referred to as $T$, $B$, $L$ and $R$, respectively. (The p-map is somewhat similar to the visibility obstruction diagram of [7].)

The bottom edge $B$ (as well as the top edge $T$) represents the searcher's position during the clockwise traversal of $\partial P$, beginning at $p_0$ and ending at some position *after* $\partial P$ has been cleared. We represent the pockets and the durations of their existence as open polygonal patches, so that, if we sweep the map with a vertical sweep line from $L$ (the beginning of the search) to $R$ (the end of the search), then at any moment the intersections of the sweep line and the patches are precisely the pockets at that moment during the search, appearing, as we view the sweep line from $B$ to $T$, in the order they are located in $\partial P$ counterclockwise from the searcher's position (Fig. 6). In other words, for any point $p \in P$, the intersection of the p-map and the vertical line passing through $p$ on $B$ (and necessarily $p$ on $T$) represents the status (whether visible or not from $p$) of the points in $\partial P$ counterclockwise from $p$ as we trace the line from $B$ to $T$. Both $T$ and $B$ are tilted to such a degree that any horizontal line through the resulting parallelogram corresponds to the status (whether visible or not from the searcher) of some *fixed* point on the polygon boundary when the searcher traverses $\partial P$ clockwise.

Let us now examine the p-map in detail, since we use it later to generate a boundary schedule of the 1-searcher. A boundary schedule $S$ (see Section 2) of the 1-searcher has two components, $S = (\alpha, \beta_1)$, where $\alpha$ is always a point on the boundary $\partial P$. Instead of dealing with such a schedule, it is more convenient to deal with the *state* of the searcher, $(p, q)$, where $p \in \partial P$ is the position of the searcher and $q \in \partial P$ is the

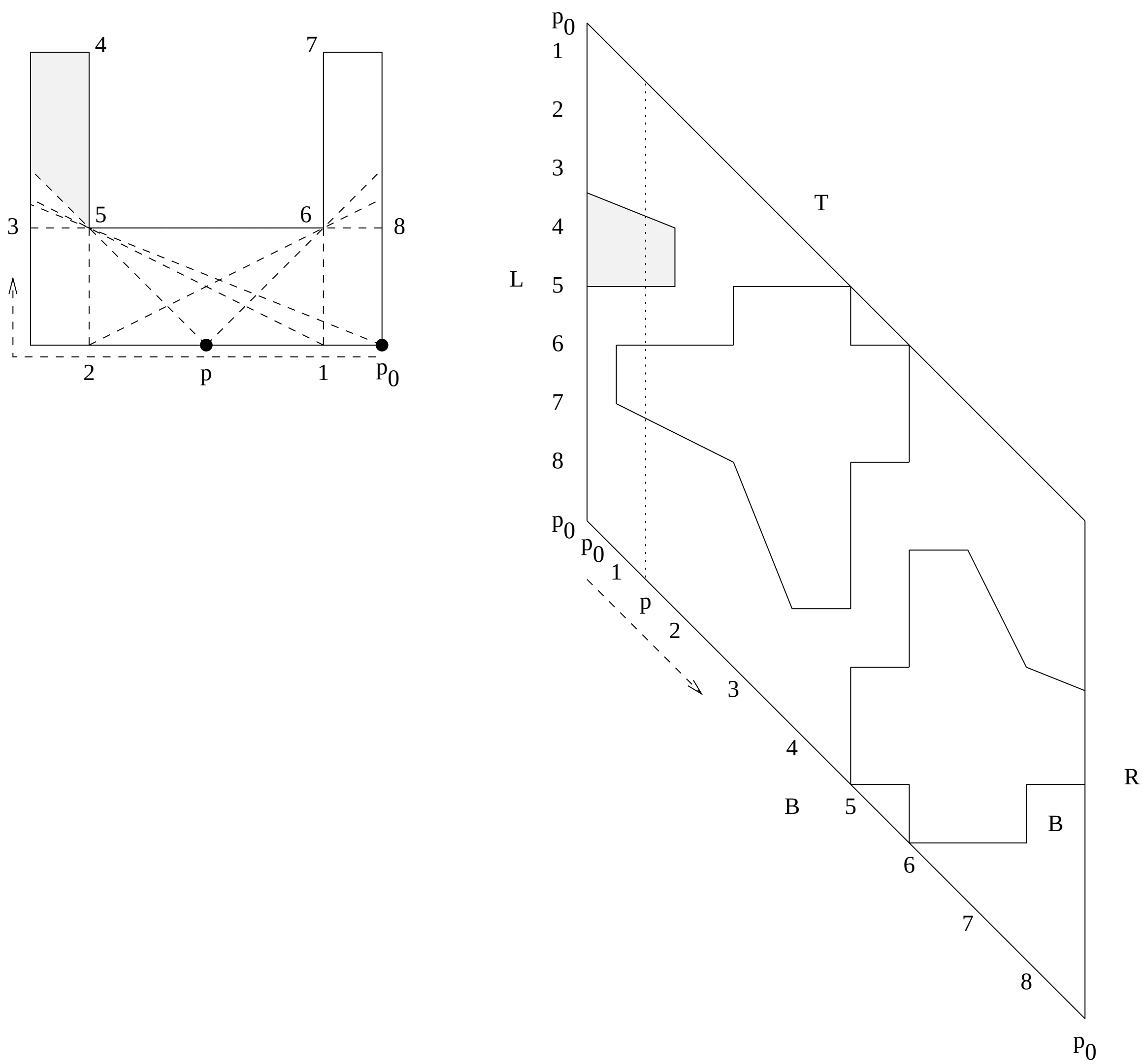



Figure 6: A polygon and its p-map corresponding to a complete clockwise traversal from $p_0$ to $p_0$. The shaded area of the polygon is contaminated when the searcher is at $p$. The vertical sweep line through $p \in B$ intersects two patches that represent the two pockets, one contaminated and one clear, not visible when the $\infty$-searcher is at $p$.

point which is currently being illuminated by the ray. As shown in Fig. 7, the p-map can then be viewed as a state-space representation of the searcher's state. Note that every point on $B$ (or $T$) represents a state of the form $(p, p)$ where the searcher is aiming the ray at her position.

For instance, in Fig. 7, the vertical upward trajectory (1) from $(p_0, p_0)$ to $(p_0, c)$ represents a counterclockwise rotation of the ray from $p_0$ to point $c \in \partial P$ while the searcher remains at location $p_0$. Continuing further horizontally to $(a, c)$ along (2) corresponds to the searcher's clockwise traversal along $\partial P$ from $p_0$ to $a$ while always aiming the ray at $c$. Of course, there are many other possible ways of reaching the state $(a, c)$ from $(p_0, p_0)$, by moving the searcher and ray simultaneously or alternately.

To interpret the meaning of patches in the p-map, let us look at the example shown in Fig. 7 again. Starting from state $(a, c)$ reached above, suppose that the searcher scans the boundary counterclockwise to point 3. Then the state trajectory is represented by the vertical line (3) going up from $(a, c)$ to $(a, 3)$. Beyond the state $(a, 5)$ reached on the lower edge of the patch, the states between it and $(a, b)$ are not physically realizable. This is exactly the meaning of a patch, namely, the states inside a patch are unrealizable, while those outside the patches are realizable. Therefore, if the searcher continues to scan beyond $(a, 5)$, then her state "jumps" to $(a, b)$. If such a jump occurs over a contaminated pocket, as is the case in Fig. 7, then contamination spreads. If the searcher moves clockwise further to point 4 while aiming the ray at point 3, then the corresponding horizontal trajectory (4) wraps around at $(3, 3)$ from $T$ to $B$. What remains is to relate a schedule to a trajectory.

Any change either in the 1-searcher's position $p$ or the point illuminated by the ray $q$ is called a *move*. (The $\infty$-searcher has only the first type of move.) In a boundary schedule, we use the term *forward move* to mean a clockwise move for $p$ along $\partial P$ or a counterclockwise move for $q$ along $\partial P$. Similarly, we use the term *backward move* to mean a counterclockwise move for $p$ or a clockwise move for $q$. In Fig. 7, moves (1), (2), and (3) are all forward moves. In the p-map representing the 1-searcher, a forward move is represented by a trajectory that moves in the upper right quadrant relative to the point representing the current state. Similarly, a backward move is represented by a trajectory that moves in the lower left quadrant relative to the point representing the current state. A schedule is in general a sequence of forward and backward moves.

For the 1-searcher, the section of $\partial P$ that lies counterclockwise from $p$ to $q$ is called the *covered section*. If the covered section is clear, we refer to it as the *cleared section*. A move that keeps the covered section clear is called *safe*, although it may in some cases cause an event, referred to as "irreversible recontamination" in [7], in which the points in a clear pocket become recontaminated when that pocket merges with a contaminated pocket. In Fig. 7, moves (1) and (2) are both safe, while move (3), in particular, the move from state $(a, 5)$ to $(a, b)$ is unsafe. Clearly, an unsafe move is undesirable, since it makes the entire covered section recontaminated, wasting all the effort made up to that point. Observe that the unsafe move (3) in the p-map shown in Fig. 7 cuts through the shaded patch vertically upwards (towards the side $T$). It

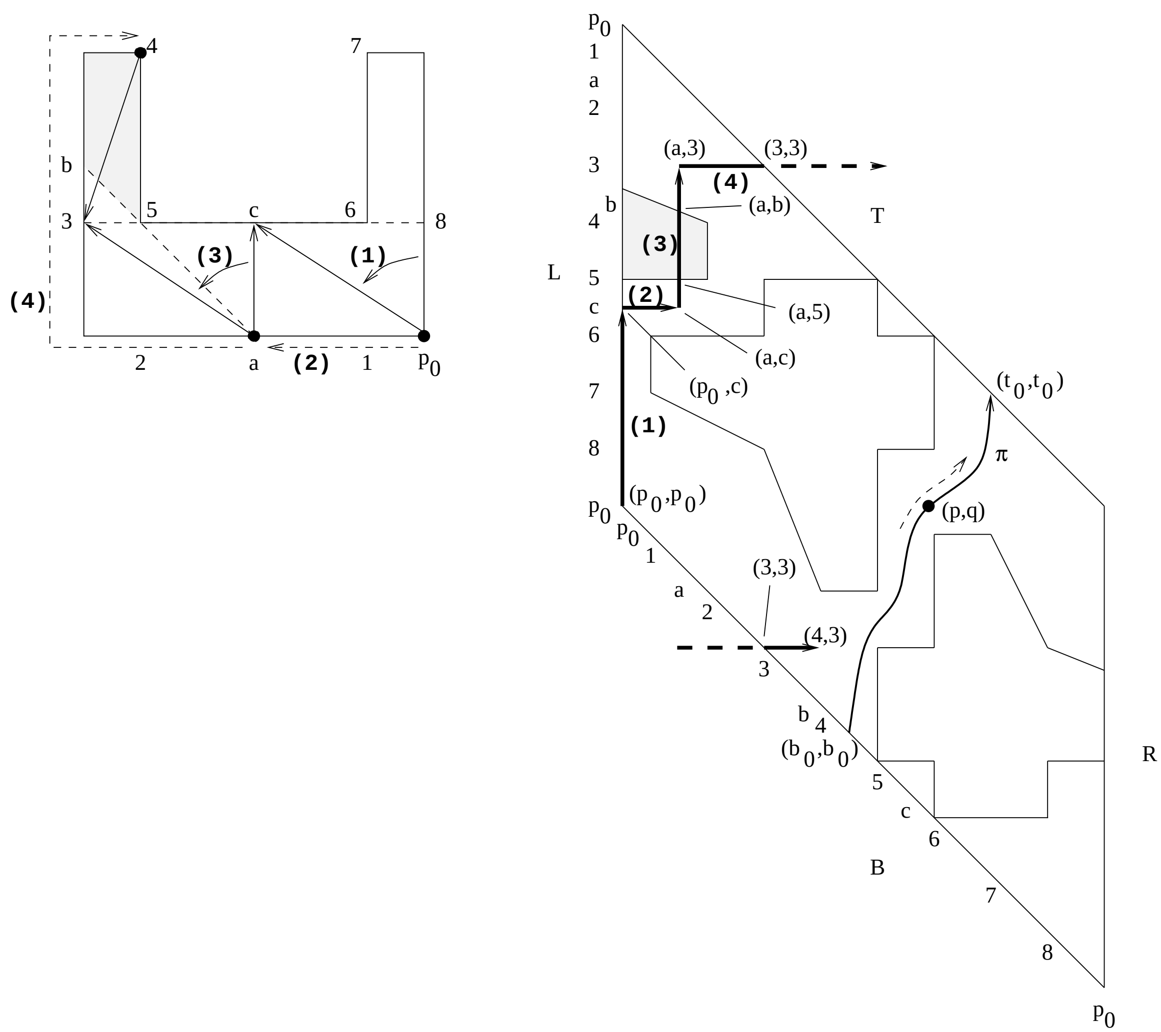



Figure 7: A trajectory in the p-map and the corresponding motion of the 1-searcher.

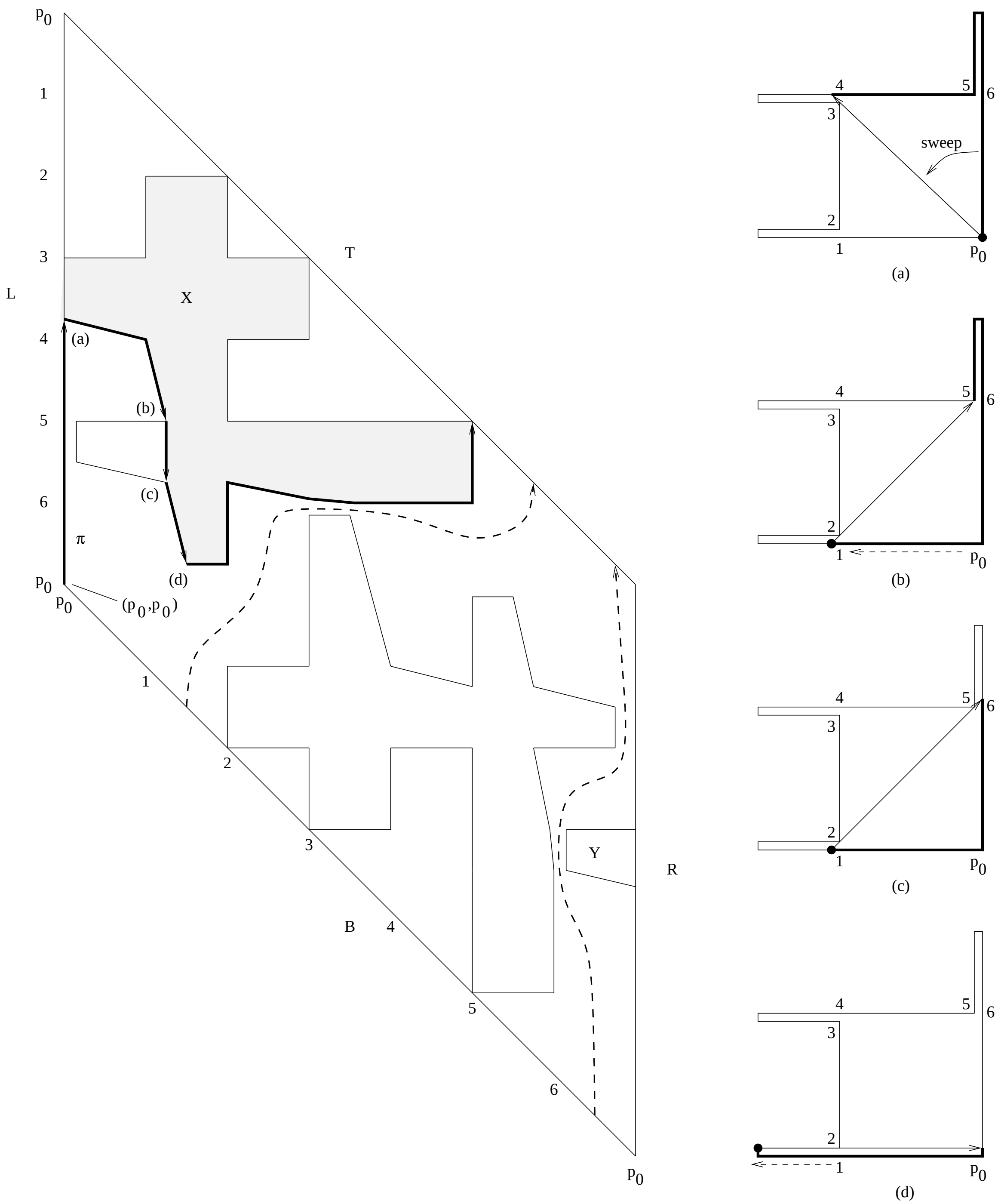



Figure 8: The 1-searcher clears polygon $P$ by tracing $\pi$ in the p-map. The bold lines in (a)-(d) indicate clear segments of $\partial P$ at the corresponding instances along $\pi$. Between (b) and (c) (along a downward segment of $\pi$) the ray jumps over a pocket, keeping the counterclockwise section of $\partial P$ between the 1-searcher and the illuminated point clear. This $\pi$ has been obtained using the construction in the proof of Lemma 3, but the figure shows $\pi$ only to the point where it hits $T$, since search is over at that moment. The dashed lines show two other paths that yield a schedule without a jump.

turns out all unsafe moves have this property, namely they manifest themselves as an upward vertical path through a patch in the p-map, while all safe backward moves across a pocket manifest themselves as a downward vertical path through a patch. Clearly the former should be avoided, while the latter can be tolerated in a schedule. In Fig. 8, the backward move from state (b) to state (c) is safe, but it causes the irreversible recontamination of the pocket between points 5 and 6.

We now prove this property formally. Let $\Sigma$ be the set of points in the p-map (i.e., the entire parallelogram), and $\Gamma$ the set of points in the patches.

**Lemma 1** *If there exists a directed path (trajectory) $\pi$ in $\Sigma - \Gamma$ from $B$ to $T$, then $P$ can be searched by a boundary schedule of the 1-searcher.*

**Proof** Suppose that $\pi$ runs from a point $(b_0, b_0) \in B$ to a point $(t_0, t_0) \in T$ (Fig. 7). Let $(p, q)$ be a (variable) point in the p-map. As $(p, q)$ moves from $(b_0, b_0)$ to $(t_0, t_0)$ within $\Sigma - \Gamma$, there will be no "jumping," which implies that the searcher never loses sight of the illuminated point $q$ that moves along the boundary $\partial P$ continuously (but not necessarily monotonically) without skipping any portion of it. Therefore the searcher can keep the covered section clear at all times, where that section, initially just point $b_0$, is the entire $\partial P$ from $t_0$ to $t_0$ when state $(t_0, t_0)$ on $T$ is reached. $\square$

The motion of the 1-searcher shown in Fig. 8 along path $\pi$ from state $(p_0, p_0)$ to state (b) has been obtained using the strategy given in the proof of Lemma 1. The dashed lines shows two other paths from $B$ to $T$ that do not intersect $\Gamma$ and hence, along which the 1-searcher can complete the search using the same strategy.

**Lemma 2** *If there exists a directed path $\pi$ in $\Sigma$ from $B$ to $T$ that intersects $\Gamma$ only along vertical line segments directed downwards, then $P$ can be searched by the boundary schedule of the 1-searcher.*

**Proof** First, note that every intersection of $\pi$ and $\Gamma$ is a vertical line segment (see trajectory (3) of Fig. 7), since the point $q$ in $\partial P$ illuminated by the ray "jumps" instantaneously over a pocket to a different point. If this vertical segment is directed downwards (towards the side $B$) when we traverse $\pi$, then the jump associated with it is always a clockwise jump that brings the illuminated point $q$ closer to the searcher's position $p$ clockwise along $\partial P$. So the covered section remains clear after the jump, if it was clear before the jump. The claim of this lemma follows from this observation and Lemma 1. $\square$

We now examine some topological properties of the p-map and show that, if $P$ is searchable by a one-way boundary schedule of the $\infty$-searcher, then the resulting p-map contains a directed path having the properties required in Lemma 2.

Since we assume that there exists at least one pocket at any time during the search, $\Gamma$ intersects both $L$ and $R$. However, since $B$ as well as $T$ represent the position of the searcher, $\Gamma$ intersects neither $T$ nor $B$. ($\Gamma$ is open except where they intersect $L$ or $R$, and its boundary may "touch" $T$ or $B$.)

Let us define a $+path$ to be a directed path within $\Gamma$ that never proceeds to left. Similarly, define a $-path$ to be one that never proceeds to right.

Since every pocket is either clear or contaminated during the search, accordingly we say that a point $(p, q) \in \Gamma$ is clear (or contaminated) if $q$ is clear (or contaminated) when the $\infty$-searcher is at $p$ during the search. Then, since

1. a contaminated pocket remains contaminated,
2. if a contaminated pocket splits into two pockets, then the new pockets are both contaminated, and
3. if a contaminated pocket merges with a clear pocket, then the resulting pocket is contaminated,

contamination "propagates" to the right (or vertically) in $\Gamma$, i.e., if a point in $\Gamma$ is contaminated, then all points in $\Gamma$ reachable from that point by a +path are also contaminated. Based on this, we identify two disjoint subsets of $\Gamma$, called $X$ and $Y$ (Figs. 6 and 9).

1. $X$ consists of all points of $\Gamma$ that are reachable by a +path from $\Gamma \cap L$.
2. $Y$ consists of all points in $\Gamma$ that are reachable by a −path from $\Gamma \cap R$.

Since all pockets are contaminated at the beginning of the search, $X$ represents precisely the pockets that are contaminated during the search. On the other hand the pockets represented in $Y$ are clear, since all pockets are clear when the search is over. So $X$ and $Y$ are disjoint. Therefore $X$ intersects $L$ and no other sides of the p-map, and $Y$ intersects $R$ and no other sides of the p-map.

A directed simple path $\pi$ in $\Sigma$ from $B$ to $T$ is called a *cut* if $X$ and $Y$ are entirely to the left and right of $\pi$, respectively (Fig. 9). Note that unless $X$ and $Y$ belong to different connected components of $\Gamma$, a cut has to cross some areas of $\Gamma - (X \cup Y)$.

**Lemma 3** *If $P$ is searchable by the $\infty$-searcher, then there exists a cut in the p-map.*

**Proof** This claim follows from an argument similar to that used in the proof of a simple topological lemma, Lemma 1 of [2]. That is, we start on $B$ at the lower left corner of the p-map, move up along $L$ until we hit $X$, and then move along the boundary of $X$. See Fig. 10. Since $X$ does not interset any of $Y$, $T$, $B$ and $R$, eventually we return to $L$. We then go up along $L$ and repeat as above each time we encounter $X$. Then we eventually reach $T$, and the path $\pi$ we just traced from $B$ to $T$ is a cut. □

**Lemma 4** *Let $\pi$ be a cut constructed in the proof of Lemma 3. Every intersection of $\pi$ and $\Gamma$ is a vertical line segment directed downwards.*

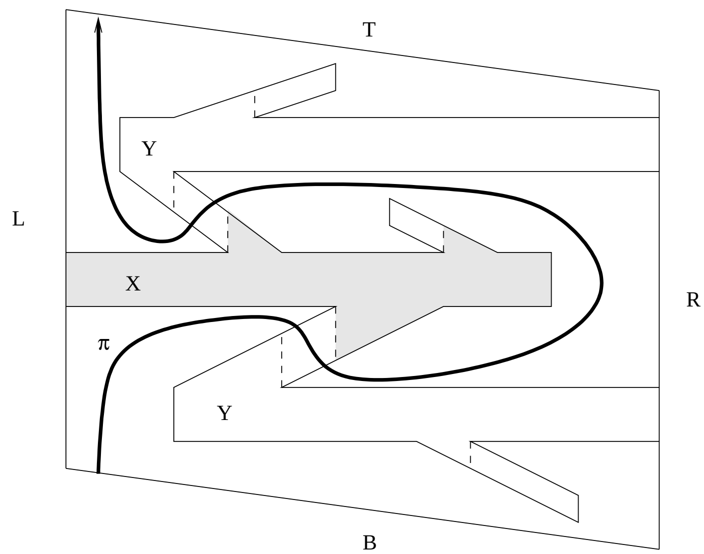



Figure 9: Schematic illustration of $X$ and $Y$, and cut $\pi$. (This p-map does not necessarily correspond to an actual polygon.)

**Proof** The cut $\pi$ constructed in the proof of Lemma 3 travels along the boundary of $X$ (except when it is on $L$). So any portion $s$ of $\pi$ inside $\Gamma$ is part of the boundary between $X$ and $\Gamma - X$, and such $s$ is a vertical segment with $X$ to the right of $s$, as shown in Fig. 10, since otherwise contamination would propagate further along some +path. Furthermore, since $X$ lies to the left of $\pi$, $\pi$ must cross $s$ downwards toward $B$. $\square$

Fig. 8 shows a polygon searchable by a boundary schedule of the $\infty$-searcher, together with cut $\pi$ in the proof of Lemma 3 (shown only to the point where it hits $T$). The section of $\pi$ between states (b) and (c) is a downward vertical segment discussed in Lemma 4. Every cut in the p-map shown in Fig. 11 has several downward vertical segments. This means that the polygon shown in Fig. 2 cannot be cleared without irreversible recontamination.

**Lemma 5** *If cut $\pi$ in the proof of Lemma 3 exists, then $P$ can be cleared by a boundary schedule of the* 1*-searcher.*

**Proof** The claim follows immediately from Lemmas 2, 3 and 4. $\square$

**Theorem 4** *$P$ can be cleared by a boundary schedule of the $\infty$-searcher if and only if it can be cleared by a boundary schedule of the* 1*-searcher.*

**Proof** Obviously any polygon that the 1-searcher can clear from the boundary can be cleared by the $\infty$-searcher from the boundary. Conversely, suppose $P$ can be cleared by a boundary schedule of the $\infty$-searcher. Then by Theorem 2 there exists a one-way boundary schedule of the $\infty$-searcher that clears $P$. Lemmas 3 and 5 then imply that $P$ can be cleared by a boundary schedule of the 1-searcher. $\square$

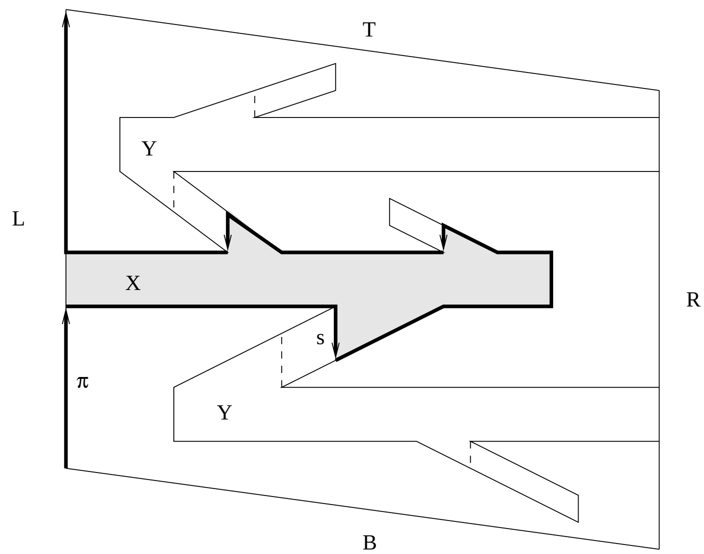



Figure 10: Cut $\pi$ in the proof of Lemma 3 and its intersection $s$ with $\Gamma$.

Note that each intersection of cut $\pi$ of Lemma 3 and $\Gamma$ represents an "irreversible recontamination" in which the points in a clear pocket become recontaminated when that pocket merges with a contaminated pocket during the search by the $\infty$-searcher. Of course, some polygons cannot be searched without irreversible recontamination. We conclude this section with the following theorem that partially settles a long-standing conjecture [13] mentioned in Section 1.

**Theorem 5** *In the general (non-boundary) polygon search problem, any polygon $P$ searchable by the $\infty$-searcher without irreversible recontamination can also be searched by the 2-searcher.*

**Proof** Construct a 2-dimensional map on the surface $\Psi$ of a cylinder that shows, as patches, the invisible pockets of $\partial P$ during the search by the $\infty$-searcher, as shown in Fig. 12. The horizontal axis of the cylinder represents time, and the left and right edges $L$ and $R$ of $\Psi$ show the status of $\partial P$ at the beginning (time 0) and end of the search (time $T > 0$), respectively, in much the same way as in the p-map. Again, if we sweep $\Psi$ by a plane perpendicular to the horizontal axis of the cylinder from $L$ (time 0) to $R$ (time $T$), then the intersection of the plane and $\Psi$ represents the locations of the pockets at any given time during the search.[1]

Now, let $\Gamma$ be the set of points in the patches, and define subsets $X$ and $Y$ of $\Gamma$ as we did earlier for the p-map, so that $X$ extends to right from $\Gamma \cap L$, and $Y$ extends to left from $\Gamma \cap R$. Since $X$ and $Y$ are disjoint, and since by assumption there are no irreversible recontamination during the search, there must exist a cut $\pi$, i.e., a simple

[1]If the $\infty$-searcher traverses $\partial P$ clockwise as in boundary search considered earlier, then the resulting cylindrical map, if cut open along the searcher's trajectory (which now lies on the surface of the cylinder), becomes essentially the same as the p-map.

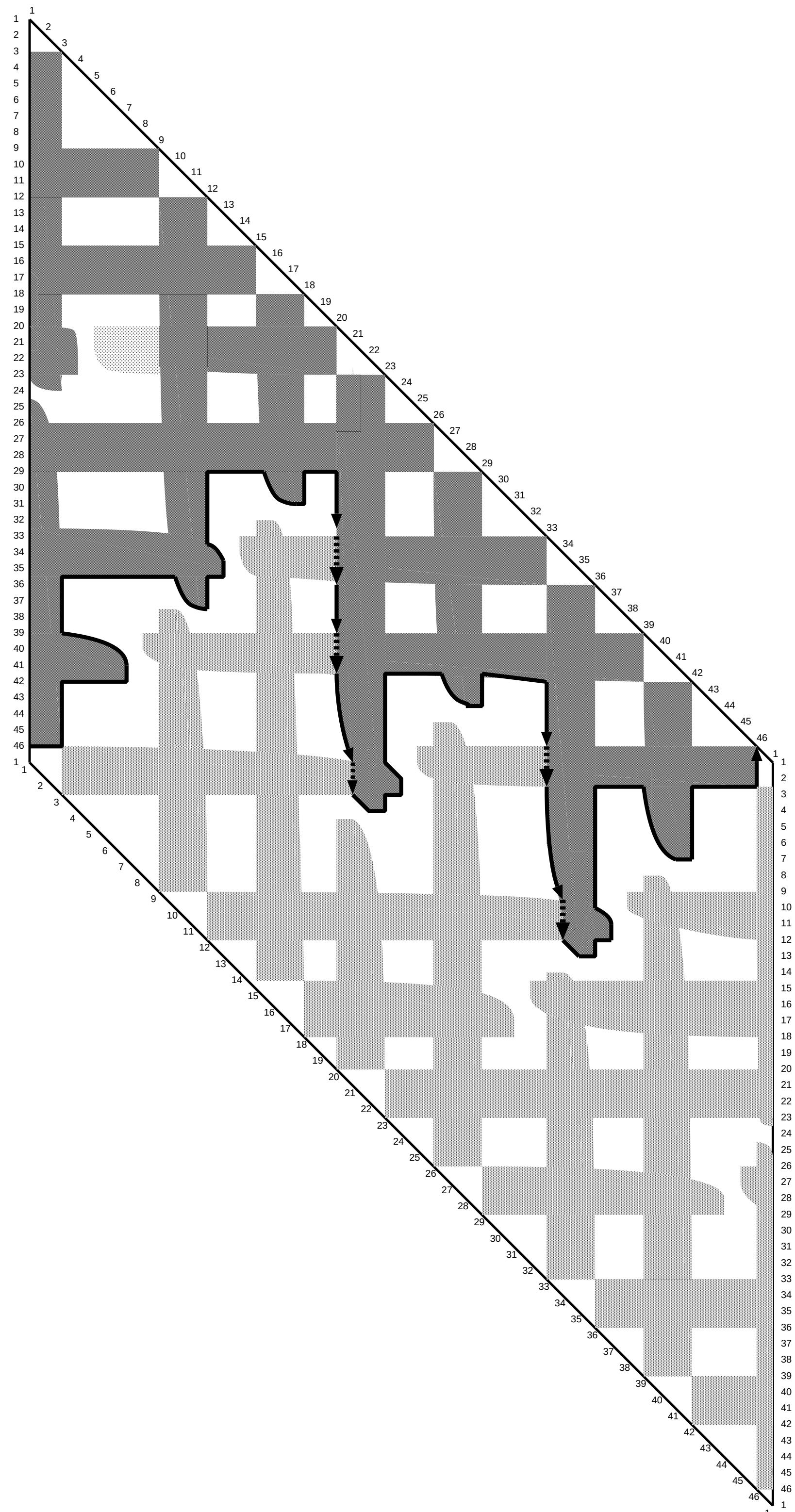



Figure 11: A schematic rendering of the p-map of the polygon shown in Fig. 2 when the $\infty$-searcher traverses the boundary from point 1. The dark region extending from left is $X$. The 1-searcher can clear this polygon by tracing the cut, shown in bold lines, that has five jumps.

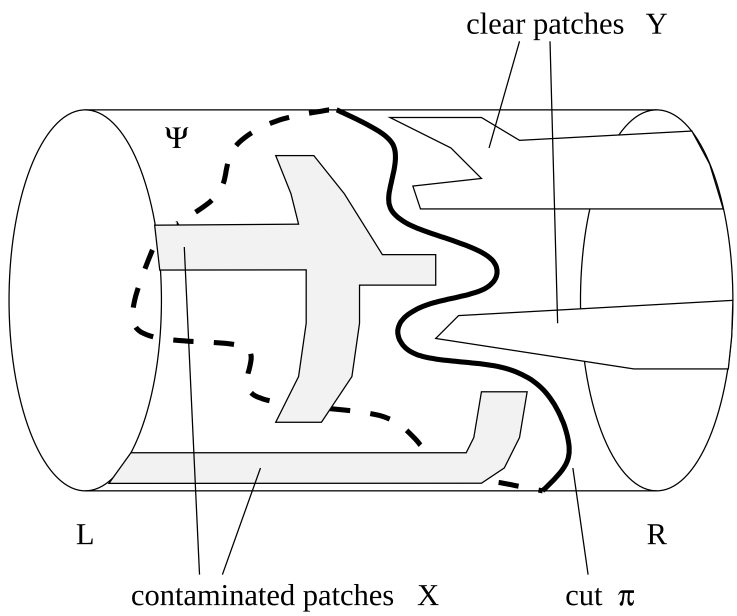
loop that goes around $\Psi$ once, that separates $X$ and $Y$ *and* that does not intersect $\Gamma$, as shown schematically in Fig. 12.

View $\pi$ as a "mountain range" where the "elevation" of a point in $\pi$ is simply the time between 0 and $T$ at which the sweep plane passes through that point. Then the points on $\pi$ having the same elevation are simultaneously visible from the $\infty$-searcher from her position at that elevation (time). Now, imagine we let two climbers walk along the mountain range $\pi$, in opposite directions starting at a point of lowest elevation, untill they meet from opposite directions at a point of highest elevation, while maintaining the same elevation at all times. By the "Mountain Climbers' Theorem" this is always possible [2]. (The climbers do not necessarily walk monotonically.) We interpret the motions of the two climbers as the points illuminated by the rays of the 2-searcher who moves along the trajectory of the $\infty$-searcher while maintaining the same elevation as the climbers at all times. That is, the 2-searcher starts by aiming both rays at the same point in $\partial P$ (the initial position of the climbers), and at the end the rays meet at some point in $\partial P$ (the final position of the climbers), after they together have swept the entire $\partial P$. At this moment the entire $\partial P$ has been cleared. Note that since $\pi$ does not intersect $\Gamma$, any subsection of $\pi$ is "reversible," making it possible for the climbers to accomplish the desired walk. □

# 6 Concluding Remarks

Using a 2-dimensional p-map and a topological argument, we have proved the equivalence of the capabilities of the $\infty$-searcher and the 1-searcher in boundary search in which the searcher has to stay within the polygon boundary. This result can also be interpreted in terms of the beam width — in boundary search a searcher with a 360° beam has no more ability than a searcher with a thin beam of infinitesimal width. We believe that the topological argument has made the proof of the equivalence simple and intuitive. The proof of the partial equivalence of the $\infty$-searcher and the 2-searcher for the general case (Theorem 5) would have been much more complex if we had attempted a "direct simulation" of the former by the latter inside $P$.

The $O(n^2)$ time algorithm for generating a schedule of the 1-searcher for an $n$-sided polygon $P$ given in [7] can naturally be used to determine whether $P$ is searchable from the boundary by the 1-searcher (since the 1-searcher must necessarily stay inside $\partial P$ to search $P$), or equivalently, by the $\infty$-searcher. Once $P$ is found to be searchable from the boundary, then one could simply use the $\infty$-searcher and move her in one way around $\partial P$ until $P$ becomes clear. The number of times $\partial P$ has to be traversed can easily be shown to be no larger than the minimum number of pockets over all points in $\partial P$, but an actual upper bound may be smaller. We are currently working on this bound using the p-map and exploiting its regularity (i.e., the same patches repeatedly appear if the searcher continues to traverse $\partial P$).



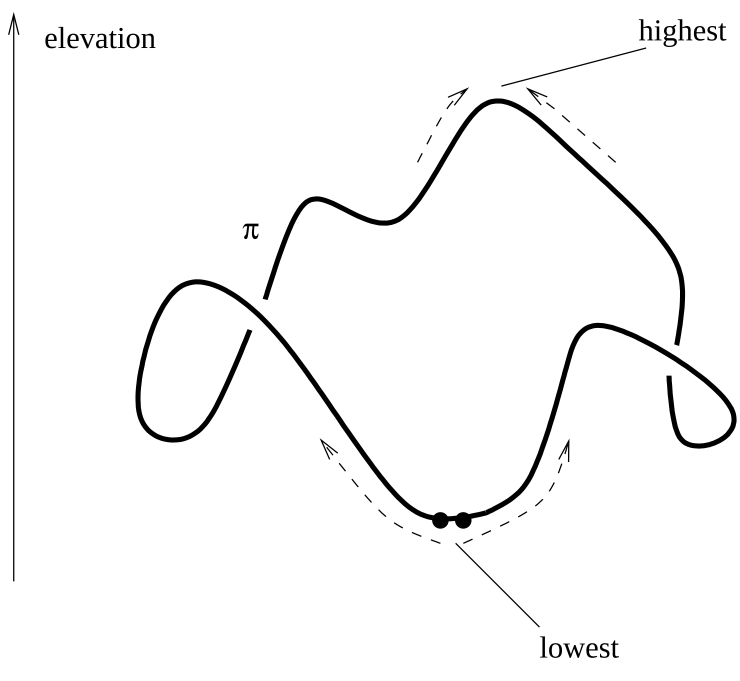




Figure 12: Cut $\pi$ on cylinder $\Psi$ viewed as a mountain range.

# References


[1] D. Crass, I. Suzuki and M. Yamashita, "Searching for a mobile intruder in a corridor – The open edge variant of the polygon search problem," *International J. Computational Geometry and Applications 5,* 4, 1995, 397–412.

[2] J.E. Goodman, J. Pach and C.K. Yap, Mountain climbing, ladder moving, and the ring-width of a polygon," *Amer. Math. Monthly,* 1989, 494—510.

[3] L.J. Guibas, J.-C. Latombe, S.M. LaValle, D. Lin and R. Motwani, "Visibility-based pursuit-evasion in a polygonal environment," in F. Dehne et al. Eds., *WADS '97 Algorithms and Data Structures (LNCS 1272)*, 1997, 17-30, Springer.

[4] P.J. Heffernan, "An optimal algorithm for the two–guard problem," *International Journal of Computational Geometry and Applications 6*, 1996, 15–44.

[5] C. Icking and R. Klein, "The two guards problem," *International Journal of Computational Geometry and Applications 2*, 1992, 257–285.

[6] S.M. LaValle and J.E. Hinrichsen, "Visibility-based pursuit-evasion: The case of curved environments," *Proc. IEEE International Conference on Robotics and Automation*, Detroit, Michigan, 1999, 1677-1682.

[7] S.M. LaValle, B.H. Simov and G. Slutzki, "An algorithm for searching a polygonal region with a flashlight," *Proc. ACM Symposium on Computational Geometry*, Hong Kong, 2000.

[8] D.T. Lee and F.P. Preparata, "An optimal algorithm for finding the kernel of a polygon," *J. ACM 26*, 1979, pp. 415–421.

[9] J.H. Lee, S.-M. Park and K.-Y. Chwa, "Searching a polygonal room with a door by a 1-searcher," *International Journal of Computational Geometry and Applications*, to appear.

[10] J.H. Lee, S.Y. Shin and K.-Y. Chwa, "Visibility-based pursuit-evasion in a polygonal room with a door," *Proc. ACM Symposium on Computational Geometry*, Miami Beach, Florida, 1999, 281–290.

[11] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, Oxford, 1987.

[12] K. Sugihara, I. Suzuki and M. Yamashita, "The searchlight scheduling problem," *SIAM J. Computing 19,* 6, 1990, 1024–1040.

[13] I. Suzuki and M. Yamashita, "Searching for a mobile intruder in a polygonal region," *SIAM J. Computing 21,* 5, 1992, 868–888.

[14] I. Suzuki, M. Yamashita, H. Umemoto and T. Kameda, "Bushiness and a tight worst-case upper bound on the search number of a simple polygon," *Information Processing Letters*, Vol. 60, 1998, pp. 49–52.

[15] L.H. Tseng, P. Heffernan and D.T. Lee, "Two-guard walkability of simple polygons," *International Journal of Computational Geometry and Applications 8*, 1998, 85–116.

[16] M. Yamashita, H. Umemoto, I. Suzuki and T. Kameda, "Searching for mobile intruders in a polygonal region by a group of mobile searchers," *Algorithmica*, to appear.