アルゴリズム 26−5
(1992. 3. 23)

掃き出し問題—多角形探索問題の一変形[1]

デビッド クラス†, 鈴木 一郎†, 山下 雅史‡

†ウイスコンシン大学ミルオーキー校
‡広島大学工学部2類

多角形探索問題の一変種である掃き出し問題を考察する. 多角形探索問題は, 与えられた多角形内を移動する侵入者を移動可能なロボットを用いて発見する問題であるが, (直観的には) 掃き出し問題では, ある辺 u を侵入者に触れさせてはならないという条件が新たに課せられる. 本報告では, 掃き出し問題が解決できるために多角形が満たすべき必要十分条件について述べる.

The "Open Edge" Variant of the Polygon Search Problem

David Crass†, Ichiro Suzuki†, and Masafumi Yamashita‡

† Department of Electrical Engineering and Computer Science
University of Wisconsin – Milwaukee
P.O. Box 784, Milwaukee, WI 53201, U.S.A.
davidc@csd4.csd.uwm.edu, suzuki@cvax.cs.uwm.edu
‡ Department of Electrical Engineering, Hiroshima University
Kagamiyama, Higashi-Hiroshima 724, Japan
mak@csl.hiroshima-u.ac.jp

This paper consideres the "open edge" variant of the polygon search problem, which is the problem of searching for mobile intruders in a simple polygon by a single mobile searcher. The open edge variant treats the case in which the given polygon must be searched without letting the intruders reach a given edge u, under an additional assumption that any number of intruders can leave and enter the polygon through another edge v at any time. We give a simple necessary and sufficient condition for a given polygon to be searchable.

[1]本研究は文部省科研費 03650305 から一部援助を受けた.

1 Introduction

Problems related to visibility inside a simple polygon have been the subject of many recent papers. Of particular interest to us among these problems is the watchman route problem [1, 2], which is an interesting variation of the well-known art gallery problem of stationing guards in a simple polygon so that every point in the interior of the polygon will be visible from at least one guard[3, 4]. The goal of the watchman route problem is to construct a shortest tour within a given simple polygon so that every point in the interior of the polygon will be visible from at least one point on the tour. Note that this goal can be interpreted as (constructing a path for) finding a *stationary* intruder located in the polygon by a single mobile searcher.

Detection of a mobile intruder in a simple polygon was first considered in the searchlight scheduling problem [5] in which the rays of stationary "searchlights" (i.e., stationary 1-searchers) are used to find the intruder. Detection of a mobile intruder in a simple polygon by a mobile searcher was considered in the polygon search problem[7]. In this paper, we adpot the following formalism in [7]: Both the searcher and the intruder are represented as points which can move continuously within a given polygon, and the intruder are assumed to be able to move arbitrarily faster than the searcher. To investigate the capabilities of the searchers having different degrees of visibility, we introduce the k-searcher for each integer $k \geq 1$ and the ∞-searcher. The k-searcher is the searcher having k "flashlights" whose visibility is limited to k rays emanating from his position, where the directions of the rays can be changed continuously with bounded angular rotation speed. The ∞-searcher is the searcher having the visibility of 360 degrees who can see in all directions simultaneously at any time.

As was observed in [7], one of the reasons for the difficulty of deciding whether a given polygon P is k-searchable or ∞-searchable is that some vertices and edges of P may have to be recontaminated[2] repeatedly during the search. This observation suggests us to consider a restricted version of the polygon search problem in which some vertices or edges of P must remain clear during the search. Specifically, the variant we consider in this paper is the following.

> Given P and two edges u and v of P, clear P under the following condition $\mathcal{B}$: (1) u must remain clear throughout the search, and (2) at any time, any point on v which is not illuminated is considered to be contaminated.

A possible interpretation of this requirement is that (1) v represents an open exit between P and its exterior through which any number of intruders can leave and enter P at any time, and (2) the searcher must force all the intruders out of P through v without allowing any of them to reach u. For this *open edge variant* of the polygon search problem we present a necessary condition for P to be ∞-searchable, and then show that the same condition is

[2]Intuitively, a point x is contaminated if the intruder can be at x.

also sufficient for P to be 2-searchable. Therefore as far as this variant is concerned, the 2-searcher and the ∞-searcher have the same capability.

In the rest of this paper, unless otherwise stated P is an n-sided simple polygon, $u = \overline{u_L u_R}$ is the edge of P which should remain clear during the search, and and $v = \overline{v_L v_R}$ is the edge of P whose points are assumed to be contaminated whenever it is not illuminated. We assume that u_L, v_L, v_R and u_R appear in this order clockwise in the bounrary ∂P of P.

2 A necessary condition

For a point $x \in P$, define $V^2(x) = \bigcup_{y \in V(x)} V(y)$, where $V(z)$ is the set of points in P visible from a point $z \in P$. Note that $V(x) \subseteq V^2(x)$. If $y \in V^2(x)$, then we say that y is *2-visible* from x. For regions Q and $R \subseteq P$, we say that Q is *weakly 2-visible* from R if every point in Q is 2-visible from some point in R. For points x, y and $z \in P$, y and z are said to be *2-separable* from x if every path within P between y and z contains at least one point in $V^2(x)$. Note that since P is simple, y and z are 2-separable from x iff $\pi(y, z)$ contains at least one point $V^2(x)$, where $\pi(y, z)$ is the Euclidean shortest path within P between y and z.

For points x and $y \in \partial P$, we let $\partial P_L(x, y)$ denote the portion of ∂P from x to y taken clockwise (i.e., the "left" boundary of P from x to y). Similarly, we let $\partial P_R(x, y)$ denote the portion of ∂P from x to y taken counterclockwise (i.e., the "right" boundary of P from x to y). $\partial P_L(u_L, v_L)$ and $\partial P_R(u_R, v_R)$ are simply written as ∂P_L and ∂P_R, respectively. For convenience, we use "$<$" to denote the order in which the points in $\partial P_L(u_L, v_L)$ (or $\partial P_R(u_R, v_R)$) appear in a traversal from u_L to v_L (or from u_R to v_R).

For vertices $x \in \partial P_L$ and $y \in \partial P_R$, we say that x and y are in *conflict* with respect to u if (1) u_L and x are not 2-separable from y and (2) u_R and y are not 2-separable from x. Similarly, x and y are said to be in *conflict* with respect to v if (1) v_L and x are not 2-separable from y and (2) v_R and y are not 2-separable from x. We say that u (or v) is *conflict-free* if there do not exist such vertices which are in conflict with respect to it. Finally, we say that u and v *satisfy the weak 2-visibility condition* if ∂P_L is weakly 2-visible from $\pi(u_R, v_R)$ and ∂P_R is weakly 2-visible from $\pi(u_L, v_L)$.

Theorem 1 *If P is ∞-searchable under condition $\mathcal{B}$, then (1) u is conflict-free, (2) v is conflict-free, and (3) u and v satisfy the weak 2-visibility condition.*

Proof (1) Suppose that $x \in \partial P_L$ and $y \in \partial P_R$ are in conflict with respect to u. Then since u_R is illuminated at time zero and $x \notin V^2(u_R)$, x is contaminated at time zero. Similarly, y is contaminated at time zero. If x is illuminated y at time t, then y and u_R are not separable from the position of the ∞-searcher at t, and hence u_R becomes contaminated. Similarly, if y is illuminated before x, then u_L becomes contaminated. Thus P cannot be cleared without contaminating u. **(2)** Suppose that $x \in \partial P_L$ and $y \in \partial P_R$ are in conflict with respect to v. When x is illuminated, y is contaminated since (1) v_R is not illuminated (and hence is contaminated by assumption) and (2) v_R and y are not separable from the location of the ∞-searcher. Similarly, x is contaminated when y is illuminated. Therefore x and y cannot become clear simultaneously. **(3)** Suppose that $x \in \partial P_L$ is not 2-visible from $\pi(u_R, v_R)$. Then u_R becomes contaminated when x is illuminated, since (1) v_R is not illuminated (and hence is contaminated by assumption) and (2) v_R and u_R are not separable from the location of the ∞-searcher. Similarly, u_L becomes contaminated when $y \in \partial P_R$ not 2-visible from $\pi(u_L, v_L)$ is illuminated. □

3 Sufficiency

The following theorem, together with Theorem 1, shows that the condition given in Theorem 1 is in fact necessary and sufficient for P to be searchable under condition $\mathcal{B}$ by the ∞-searcher and the k-searcher for any $k \geq 2$.

Theorem 2 *If (1) u is conflict-free, (2) v is conflict-free, and (3) u and v satisfy the weak 2-visibility condition, then P is 2-searchable under condition $\mathcal{B}$.*

Proof To save space we only give an outline of the proof. Let us denote the two flashlights by F_L (the "left" flashlight) and F_R (the "right" flashlight). Suppose that the 2-searcher is located at a point $s \in P$ aiming F_L and F_R at points $x \in \partial P_L$ and $y \in \partial P_R$, respectively, where x, s and y are collinear. Intuitively, we view the segment $\overline{xy}$ as a variable-length bar $\mathcal{L}$ determined by the rays of the flashlights, and clear P by sweeping it by $\mathcal{L}$ from u to v, in such a way that at any time, the subregion of P "below" $\mathcal{L}$, denoted BELOW(x, y) when $\mathcal{L}$ is at $\overline{xy}$, remains clear. Note that BELOW(x, y) contains edge u which must remain clear. As is shown in Figure 1, during this sweep $\mathcal{L}$ must occasionally be "bent" to clear the regions not visible from the "opposite" boundary.

Let C denote the set of nonredundant chords of P induced by a vertex adjacent to a reflex vertex. We denote by C_L (or C_R) the set of chords $c \in C$ such that both endpoints of c are in ∂P_L (or ∂P_R). It is easy to see that if bar $\mathcal{L}$ intersects every chord $c \in C_L \cup C_R$ during the sweep, then every point in P will be visible from the 2-searcher at least once and hence can be illuminated by one of the flashlights. In fact, we sweep P by repeatedly finding a suitable "next" chord $c \in C_L \cup C_R$ and advancing $\mathcal{L}$ so that it intersects c.

For each chord $c \in C_L \cup C_R$, let $\eta(c)$, $B(c)$ and $T(c)$ be the vertex inducing c and the "bottom" and "top" endpoints of c, respectively, where $B(c) < \eta(c) < T(c)$. Suppose that currently bar $\mathcal{L}$ is at $\overline{xy}$ for some $x \in \partial P_L$ and $y \in \partial P_R$. First, we find the chord $c_L \in C_L$ whose B point is encountered first in a traversal of $\partial P_L(x, v_L)$ from x to v_L. (If such c_L does not exist, then we treat vertex v_L as a chord such that $B(v_L) = T(v_L) = v_L$. A similar comment applies to c_R mentioned next.) We find $c_R \in C_R$ in a similar manner. Clearly, either c_L or c_R must be the next chord to be intersected by $\mathcal{L}$. To decide which of the two chords should be intersected next, we do the following. For c_L, let $\alpha(c_L)$ be the first point encountered in a traversal of $\partial P_R(y, v_R)$ from y to v_R from which at least one point in c_L is visible, if such a point exists; otherwise let $\alpha(c_L)$ be the first point encountered in a traversal of $\partial P_R(u_R, y)$ from y to u_R from which at least one point in c_L is visible. Then let p be the point in c_L closest to $B(c_L)$ visible from $\alpha(c_L)$, and let $\beta(c_L) \in \partial P_L$ be the first point at which the ray emanating from $\alpha(c_L)$ in the direction from $\alpha(c_L)$ to p penetrates ∂P. See Figure 2 for illustration. Note that $\mathcal{L}$ will coincide with $\overline{\beta(c_L)\alpha(c_L)}$ if we intersect it with c_L with "minimum" advancement. We also define $\alpha(c_R) \in \partial P_L$ and $\beta(c_R) \in \partial P_R$ for c_R in a completely symmetrical manner.

It is easy to show that, since v is conflict-free, the conditions $\alpha(c_L) < y$ and $\alpha(c_R) < x$ cannot hold simultaneously. If one of the two conditions, say $\alpha(c_L) < y$, holds, then we can advance $\mathcal{L}$ from $\overline{xy}$ to $\overline{\beta(c_L)\alpha(c_L)}$ as follows. Let q be the intersection of $\overline{xy}$ and $\overline{\beta(c_L)\alpha(c_L)}$. Here, $\partial P_L(x, \beta(c_L))$ is 2-visible from q, since otherwise, as is easily shown either (1) there exists a chord in C_L between x and $B(c_L)$, or (2) there exists a chord in C_L between $B(c_L)$ and $\beta(c_L)$ and hence c_L is redundant. Thus we move the 2-searcher to q aiming F_L and F_R at x and y, respectively, and then sweep $\partial P_L(x, \beta(c_L))$ from q using F_L, in such a way that (1) F_L and F_R are aimed in opposite directions whenever the 2-searcher is located at q, and (2) F_R is aimed through q whenever the 2-searcher is not located at q. Since F_R is rotated only clockwise, BELOW$(\beta(c_L), \alpha(c_L))$ becomes clear when the sweep is completed. Note that the right endpoint of $\mathcal{L}$ has been moved backward from y to $\alpha(c_L)$. Thus we refer to this case as a *back-up* case.

Let us consider the case in which $y \le \alpha(c_L)$ and $x \le \alpha(c_R)$. In this case at least one of $\alpha(c_R) \le B(c_L)$ and $\alpha(c_L) \le B(c_R)$ must hold, since otherwise $\eta(c_L)$ and $\eta(c_R)$ are in conflict with respect to u. If $\alpha(c_L) \le B(c_R)$, then we advance $\mathcal{L}$ to $\overline{\beta(c_L)\alpha(c_L)}$ so that it intersects c_L. (If $\alpha(c_R) \le B(c_L)$ then we advance $\mathcal{L}$ to $\overline{\alpha(c_R)\beta(c_R)}$ so that it intersects c_R.) This is done by the following algorithm $\mathcal{A}$.

Algorithm $\mathcal{A}$;
begin
  Compute $\pi(y, \beta(c_L)) = \overline{s_1 s_2 \ldots s_m}$ where $s_1 = y$, $s_m = \beta(c_L)$, none of $s_1, \ldots, s_{m-1}$ is in $\partial P_L(\beta(c_L), v_L)$, and no three consecutive points in $s_1, \ldots, s_m$ are collinear;
  for $i := 1$ **to** $m-1$ **do**
    if $s_i \in \partial P_R$ and $s_{i+1} \in \partial P_R$ **then**
      begin
        $x_i :=$ SHOOT(s_i, s_{i+1});
        L_ADVANCE_BY_SWEEP(x_i);
        R_ADVANCE_FROM_LID(s_{i+1}); {$\mathcal{L}$ is at $\overline{x_i s_{i+1}}$.}
      end
    else if $s_i \in \partial P_R$ and $s_{i+1} \in \partial P_L$ **then** L_ADVANCE_BY_SWEEP(s_{i+1}); {$\mathcal{L}$ is at $\overline{s_{i+1} s_i}$.}
    else if $s_i \in \partial P_L$ and $s_{i+1} \in \partial P_R$ **then** R_ADVANCE_BY_SWEEP(s_{i+1}); {$\mathcal{L}$ is at $\overline{s_i s_{i+1}}$.}
    else {$s_i \in \partial P_L$ and $s_{i+1} \in \partial P_L$}
      begin
        $x_i :=$ SHOOT(s_{i+1}, s_i); {$x_{m-1} =$ SHOOT$(s_m, s_{m-1}) = \alpha(c_L)$}

R_ADVANCE_BY_SWEEP(x_i);
L_ADVANCE_FROM_LID(s_{i+1}); {$\mathcal{L}$ is at $\overline{s_{i+1}x_i}$.}
end;
end;

Algorithm $\mathcal{A}$ is written using the following four operations and SHOOT, where SHOOT(r, s) is the first point at which the ray emanating from point r in the direction from r to point s intersects ∂P. In the following explanations, assume that $\mathcal{L}$ is currently at $\overline{xy}$ and BELOW(x, y) is clear.

1. L_ADVANCE_FROM_LID(z): z is a point in $\partial P_L(x, v_L)$ such that (1) z, x and y are collinear and (2) $\partial P_L(x, z)$ is weakly visible from $\overline{xz}$. The 2-searcher moves to x aiming F_L and F_R at x and y, respectively, and then clears the region whose boundary is $\partial P_L(x, z) \cup \overline{xz}$ using F_L, while aiming F_R continuously at y. When this is done, $\mathcal{L}$ is at $\overline{zy}$ and BELOW(z, y) is clear.

2. R_ADVANCE_FROM_LID(z): This is symmetric to L_ADVANCE_FROM_LID(z), and can be used to advance $\mathcal{L}$ to $\overline{xz}$.

3. L_ADVANCE_BY_SWEEP(z): z is a point in $\partial P_L(x, v_L)$ such that (1) $z \in V(y)$ and (2) $\partial P_L(x, z)$ is 2-visible from y. The 2-searcher moves to y aiming F_L and F_R at x and y, respectively, and then sweeps $\partial P_L(x, z)$ using F_L, When this is done, $\mathcal{L}$ is at $\overline{zy}$ and BELOW(z, y) is clear.

4. R_ADVANCE_BY_SWEEP(z): This is symmetric to L_ADVANCE_BY_SWEEP(z), and can be used to advance $\mathcal{L}$ to $\overline{xz}$.

The process of advancing $\mathcal{L}$ by algorithm $\mathcal{A}$ is illustrated in Figure 3. The fact that the executions of the four operations always succeed in algorithm $\mathcal{A}$ follows from the way we choose c_L and c_R, the condition $\alpha(c_L) \leq B(c_R)$, and the assumption that u and v satisfy the weak 2-visibility condition. We leave details to the reader since the argument is elementary.

Note that the endpoints of $\mathcal{L}$ are moved backward toward u_L and u_R only when a back-up case occurs, and hence a back-up case can cause some chords to be intersected by $\mathcal{L}$ more than once. However, it turns out that a chord which is intersected by $\mathcal{L}$ in a back-up case will never be selected again as the next chord to be intersected. To see this, suppose that $\mathcal{L}$ intersects a chord, say $c_L \in C_L$, in a back-up case, and c_L is selected again as the next chord to be intersected later. Then there must have been another back-up case which brings the left endpoint of $\mathcal{L}$ to a point below $B(c_L)$. If that back-up case is caused by a chord $c_R \in C_R$ such that $\alpha(c_L) < B(c_R)$, then $\eta(c_L)$ and $\eta(c_R)$ are in conflict with respect to v. Otherwise, by an elementary analysis we can show that there exist chords $c'_L \in C_L$ and $c_R \in C_R$ such that $\eta(c'_L)$ and $\eta(c_R)$ are in conflict with respect to v. In either case, the assumption that v is conflict-free is violated. Thus there can be only $O(n)$ back-up cases during the sweep, and hence each chord in $C_L \cup C_R$ is intersected by $\mathcal{L}$ only $O(n)$ times. Therefore eventually $\mathcal{L}$ is advanced to $v = \overline{v_L v_R}$ and hence P becomes clear. This completes the proof of Theorem 2. □

For arbitrary vertices x, y and z of P, y and z are not 2-separable from x iff y and z belong to the same maximal connected region of $P - V^2(x)$. This condition can be tested for any given vertices x, y and z in constant time, once we construct $V^2(x)$ for each vertex x and find, for each vertex $y \notin V^2(x)$, the maximal connected region of $P - V^2(x)$ containing y. Since $V^2(x)$ can be constructed in $O(n)$ time (n is the number of vertices of P) for each vertex x from a triangulation of P [6], whether there exist vertices which are in conflict with respect to u or v can be tested in $O(n^2)$ time. Since whether u and v satisfy the weak 2-visibility condition can also be tested in $O(n^2)$ time by constructing $V^2(x)$ for each vertex x, we can test whether P satisfies the condition of Theorem 2 in $O(n^2)$ time.

Theorem 3 *If P satisfies the condition of Theorem 2, then a search schedule of the 2-searcher consisting of $O(n^2)$ elementary actions for clearing P under condition $\mathcal{B}$ can be generated in $O(n^2 \log n)$ time.* □

As a final note, the length of the schedule generated in the proof of Theorem 2 is asymptotically worst-case optimal, since there is a polygon for which any search schedule of the ∞-searcher under condition $\mathcal{B}$ contains $\Omega(n^2)$ elementary actions.

参考文献

[1] W.-P. Chin and S. Ntafos, "Optimum watchman routes," *Proc. of the 2nd Annual Symposium on Computational Geometry*, Yorktown Heights, New York, June 1986, pp. 24–33.

[2] W.-P. Chin and S. Ntafos, "Shortest watchman routes in simple polygons," Technical Report, Computer Science Dept., Univ. of Texas at Dallas, 1987.

[3] V. Chvátal, "A combinatorial theorem in plane geometry," *J. Combinatorial Theory (B) 18*, 1975, pp. 39-41.

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

[5] K. Sugihara, I. Suzuki and M. Yamashita, "The searchlight scheduling problem," *SIAM J. Computing 19*, No. 6, 1990, pp. 1024–1040.

[6] S. Suri, "Minimum link paths in polygons and related problems," Ph.D. Dissertation, Department of Computer Science, Johns Hopkins University, 1987.

[7] I. Suzuki and M. Yamashita, "Searching for a mobile intruder in a polygonal region," *SIAM J. Computing* (to appear).

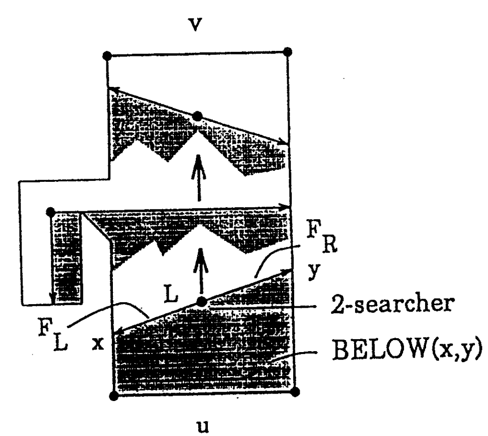

図 1: Sweeping P by bar $\mathcal{L}$ at $\overline{xy}$ keeping BELOW(x,y) clear.

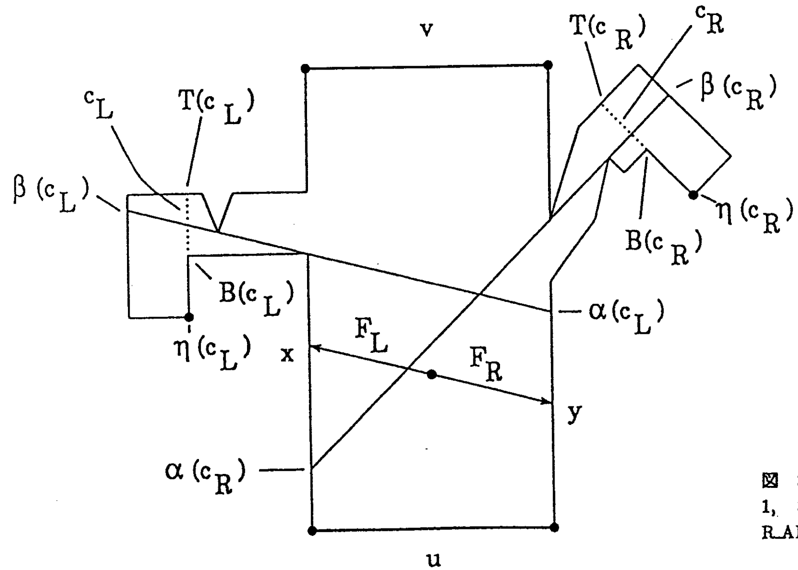

図 2: c_L, $\alpha(c_L)$, $\beta(c_L)$, c_R, $\alpha(c_R)$ and $\beta(c_R)$.

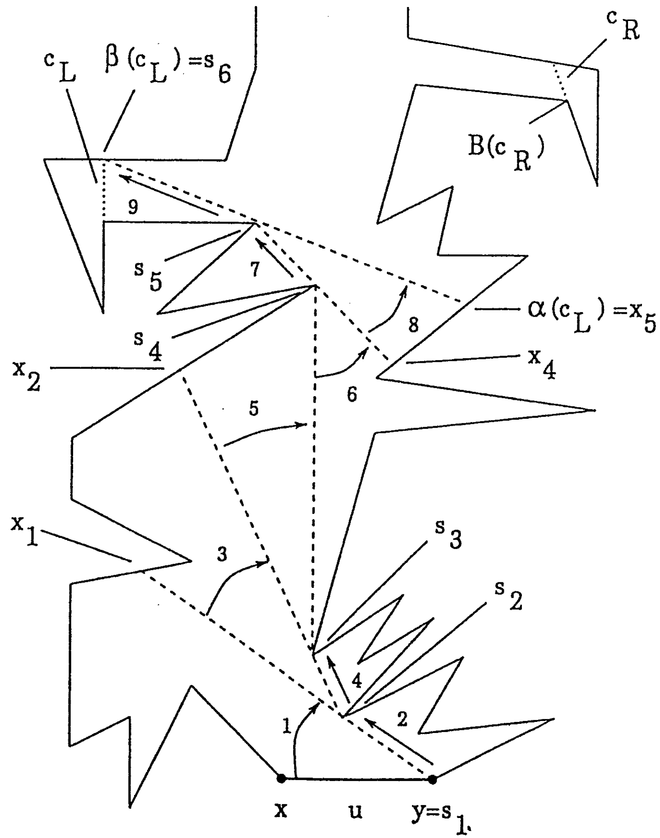

図 3: Advancement of $\mathcal{L}$ from $\overline{xy} = u$ to $\overline{\beta(c_L)\alpha(c_L)}$ by algorithm $\mathcal{A}$. 1, 3, 5: L_ADVANCE_BY_SWEEP; 2, 4: R_ADVANCE_FROM_LID; 6, 8: R_ADVANCE_BY_SWEEP; 7, 9: L_ADVANCE_FROM_LID.