チェーンガイドによる多角形の捜索アルゴリズム

譚 学厚
東海大学 開発工学部

概要

本論では、$k+1$ 人の捜索員が一つのチェーンになって単純多角形における移動対象を捜索する問題について調べる。まず、「リンク－k 図」と呼ばれるデータ構造を提案する。リンク－k 図では、リンク距離が k を超えない点対はすべて記録され、また最短リンクパスにおける推移関係も記録されている。次に、単純多角形における動的対象を捜索するのに必要な最小の捜索員の数 r^* を $O(n^2)$ 時間で計算するアルゴリズムを提案する。捜索できる場合には、捜索のスケジュールを $O(r^*n^2)$ 時間で報告する。我々の結果は今までのアルゴリズムを線形時間も改善した。

Sweeping simple polygons with the mimimum number of chain guards

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

Abstract We study the problem of detecting a moving target using a group of $k+1$ (k is a positive integer) mobile guards inside a simple polygon. Our guards always form a simple polygonal chain within the polygon such that consecutive guards along the chain are mutually visible. In this paper, we introduce the notion of the *link-k diagram* of a polygon, which records all the pairs of points on the polygon boundary such that the link distance between any of these pairs is at most k and a transition relation among minimum-link ($\leq k$) paths as well. An $O(n^2)$ time algorithm is then presented to compute the minimum number r^* of guards required to detect the target, no matter how fast the target moves. Moreover, a sweep schedule can be reported in $O(r^*n^2)$ time. Our results improve upon the previously known time bounds by a linear factor.

1 Introduction

Recently, much attention has been devoted to the problem of detecting an unpredictable, moving target in an n-sided polygon P by a group of mobile guards. Both the target and the guards are modeled by points that can continuously move in P. The goal of the guards is to eventually "see" the target, or to verify that no target is present in the polygon, no matter how fast the target moves. Many types of polygon shapes and the vision sensors of the guards have been studied by now. See [1, 2, 4, 5, 6, 3, 8, 9, 10, 11].

In this paper, we focus on the model of the guards studied in [1], in which $k+1$ (k is a positive integer) guards form a polygonal chain inside the polygon P such that consecutive guards along the chain are mutually visible. The goal here is to sweep P with a continuously moving chain of guards so that at any instant, the chain of guards partitions P into a "cleared" region (not containing the target) and an "uncleared" region (it may contain the target). Thus, the first guard and the last guard in the chain should always be kept on the boundary of P. In the end, we would like to make sure that the whole polygon P be cleared. This target-finding model may have applications in adversarial settings, as it has obvious advantages for safety and communication between guards. Efrat et al. [1] presented an $O(n^3)$ time and $O(n^2)$ space algorithm for computing the minimum number r^* of guards required to sweep P, and an $O(r^* n^3)$ time algorithm for generating a sweep schedule. Also, they gave an $O(n^2)$ time and $O(n^2)$ space (resp. $O(n \log n)$ time and $O(n)$ space) algorithm for computing an integer $r \leq r^* + 2$ (resp. $r \leq r^* + 16$) such that one can sweep P using r guards. Their algorithms are based on the so-called *link diagram*, which encodes the link distance between all pairs of points on the boundary of the polygon. Note that the link diagram can actually be applied to the two-point link-distance query problem [1], which is much stronger that the original problem of sweeping a simple polygon with a chain of guards.

The main contribution of this paper is an $O(n^2)$ time and space algorithm for computing the minimum number r^* of guards required to detect the target, and an $O(r^* n^2)$ time algorithm for generating a sweep schedule. To this end, we introduce the notion of the *link-k diagram* of a polygon, which records all the pairs of points on the polygon boundary such that the link distance between any of these pairs is at most k and a transition relation among minimum-link ($\leq k$) paths as well. Our link-k diagram is much simpler than the link diagram, as it does not contain all link distances between any pairs of points on the boundary of the polygon. The link-k diagram can be construted in $O(n^2)$ time and space. Furthermore, the link-k diagram can simply be modified so that whether there exists a sweep schedule for a chain of $k+1$ guards can be determined in $O(n^2)$ time, and a sweep schedule (if it exists) can be output in $O(kn^2)$ time. Combining with the approximation algorithms of Efrat et al. [1], the main results of this paper can then be obtained.

2 Preliminary

Let P denote a simple polygon of n vertices, and let ∂P denote the boundary of P. Two points p and q are *visible* from each other if the segment connecting them does not intersect the exterior of P.

Let $G = \{g_1, g_2, \ldots, g_{k+1}\}$ be a set of point guards in P. Let $\gamma_i(t)$, $1 \le i \le k+1$, denote the position of the guard g_i in P at time t; we require that $\gamma_i(t) : [0, \infty) \to P$ be a continuous function. A *configuration* of G at time t is the set of points $\{\gamma_i(t) | 1 \le i \le k+1\}$. We say a configuration is *legal* if $\gamma_1(t)$, $\gamma_{k+1}(t)$ lie on ∂P and every line segment $\overline{\gamma_i(t)\gamma_{i+1}(t)}$ $(1 \le i \le k)$ does not intersect the exterior of P. A *motion action* is a specification of $\gamma_i(t)$, which is an algebraic path (usually a line segment in P) along which the guard g_i moves at time t.

Assume that the chain corresponding to the configuration of the guards is oriented from g_1 to g_{k+1}, and the guards g_1, g_2, ..., g_{k+1} are in counterclockwise order if $k \ge 2$. Also, assume that the initial positions of all guards are located at a vertex or on an edge of P. Let $P(t)$ denote the fraction of the area of P to the right of the configuration of guards at time t. Clearly, $P(0) = 0$. We say a *sweep schedule* exists for P if $P(t) = 1$ for some $t > 0$. The *complexity* of a sweep schedule is the total number of motion actions it consists of.

3 The link-k diagram

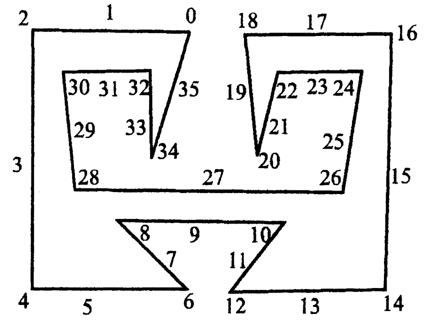
In this section, we introduce the notion of the *link-k diagram* of a polygon, which records the link distance of at most k between the pairs of points on the polygon boundary and a transition relation among minimum-link ($\le k$) paths. Also, we show that the link-k diagram can be constructed in $O(n^2)$ time and space.

Let $M = \{m_i | i \in Z_{2n}\}$ denote the set of vertices and edges of P numbered in counterclockwise order, where m_{2i} denotes the ith vertex and m_{2i+1} denotes the edge between two vertices m_{2i} and m_{2i+2}. Throughout of this paper, we assume that all edges of the polygon P are *open*, that is, the edge m_{2i+1} does not contain the vertices m_{2i} and m_{2i+2}. The indices are computed modulo $2n$.

Given two points p, $q \in P$, a *minimum-link path* between p and q is a piecewise-linear path between p and q that does not intersect the exterior of P and has the minimum number of line segments; the *link distance* $d_L(p,q)$ between p and q is the number of line segments of such a path. Similarly, we define the *link distance* $d_L(m_i, m_j)$ between m_i and m_j (possibly $i = j$) as the minimum value of $d_L(p,q)$, where $p \in m_i$ and $q \in m_j$.

The *window partition* $\mathcal{W}_p$ of a point $p \in P$ is a partition of P into maximal regions of constant link distance from p. An edge of $\mathcal{W}_p$ is either a portion of an edge of P or a segment that separates two regions of $\mathcal{W}_p$; we call such a segment a *window* of $\mathcal{W}_p$. Suri [7] introduced the notion of window partition and showed that it can be constructed in $O(n)$ time and space. The definition of window partitions extends naturally to the case when the source is a line segment, instead of a point.

Our link-k diagram G is constructed on the grid $2n \times 2n$. Given a positive integer k, all vertices and edges having the link distance of at most k to an element m_i of M can be computed in $O(n)$ time, using the *window partition* of m_i [7]. The polygon shown in Figure 1(a) consists of 36 elements, and the sets of the elements which are link-2 visible from the first five elements are shown in Figure 1(b). We define the node set $V(G)$ as $\{(m_i, m_j) | d_L(m_i, m_j) \le k\}$. The edge set $E(G)$ is obtained by connecting all the pairs of nodes such that two nodes are vertically or horizontally adjacent in the grid. Note that the nodes on the boundary of the grid are considered to be adjacent to the corresponding nodes on the other side of the grid, i.e., we "glue" together the top side and the bottom side of the grid, and together the left side and the right side of the grid.



0: 0,1,2,3,4,5,28,29,30,31,32,33,34,35

1: 0,1,2,3,4,5,6,7,8,9,10,15,26,27,28,29,30,31,32,33,34,35

2: 0,1,2,3,4,5,6,7,8,9,10,15,26,27,28,28,30,31,32,33,34,35

3: 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,24,25.26, 27,28,29,30,31,32,33,34,35

4: 0,1,2,3,4,5,6,7,8,9,10,15,26,27.28,29,30,31,32,35

(a) (b)

図 1 A simple polygon P (a), and the sets of the elements link-2 visible from the first five elements of P (b).

It is easy to see that the node set $V(G)$ records the link distance of at most k between the pairs of points on the polygon boundary, and an edge of $E(G)$ shows the transition relation between the sets of minimum-link ($\leq k$) paths represented by two nodes adjacent in the grid.

Lemma 1 The link-k diagram of a simple polygon can be constructed in $O(n^2)$ time and space.

Proof. Since all vertices and edges having the link distance of at most k to an element m_i of the set M can be computed in linear time [7], the total time required for computing the node set $V(G)$ is $O(n^2)$. After $V(G)$ is obtained, each edge of $E(G)$ can be constructed in constant time. Hence, the lemma follows. □

4 Algorithms

We first describe how to modify the diagram G into a new diagram G' so that whether there exists a sweep schedule for a group of $k+1$ guards can be determined. Assume without loss of generality that the chain of $k+1$ guards starts at some m_i. So we add a starting node s to the diagram and connect s to all nodes (m_i, m_i), for $1 \leq i \leq 2n$.

Suppose that a sweep schedule exists for P. Since we have assumed that the cleared region is to the right of the chain of guards (as viewed from g_1), a sweep schedule is complete if it ever reaches some node (m_j, m_{j-1}). Thus, we also add an ending node t to G and connect t to all nodes (m_j, m_{j-1}), for $1 \leq j \leq 2n$. In order to avoid an empty (or trivial) sweep schedule, we delete all edges connecting (m_i, m_i) and (m_i, m_{i-1}) from G. Let G' denote the diagram obtained after these modifications are made.

Lemma 2 The polygon P can be swept by a chain of $k+1$ guards if and only if the graph G' contains an st-path.

Proof. Assume first that P can be swept by a chain of $k+1$ guards. Fix a sweep schedule S.The configuration of $k+1$ guards at any time instant can be mapped to a node in G' as follows: If the first guard is located at some point of m_i and the last guard at some point of m_j, this configuration corresponds to the node (m_i, m_j). Note that several consecutive configurations may correspond to the same node in G'. (Actually, it is an onto mapping.) Thus, the sweep schedule S can be mapped to a sequence of nodes in G'. Add s and t to two ends of the sequence, and denote the resulting sequence by S'.

What we need to do is then to show that S' is an st-path in G'. It suffices to show that any two consecutive (and different) nodes, say, u and v, of S' are connected by an edge in G'. (Since the cleared region at any instant of the sweep schedule S is to the right of the configuration of guards, we have assumed that S does not contain any trivial motion action from (m_i, m_i) to (m_i, m_{i-1}).) If $u = s$ or $v = t$, G' clearly contains an edge from u to v. Otherwise, the difference between the indices of u and v is exactly one, which occurs due to a motion action of g_1 or g_{k+1} between an edge and one of its incident vertices. So there is an edge in G' that connects u and v. Hence, we have that if P can be swept by a chain of $k+1$ guards, then G' contains an st-path.

Assume now that G' contains an st-path. Fix an edge of the st-path, say, from (m_i, m_j) to (m_i, m_{j+1}). Take two points $p \in m_i$, $q \in m_j$ such that the link distance between p and q is at most k, and two points $p' \in m_i$, $q' \in m_{j+1}$ such that the link distance between p' and q' is at most k. (Possibly, $p = p'$ or/and $q = q'$.) Let $\pi_{p,q}$ (resp. $\pi_{p',q'}$) denote a minimum-link path between p and q (resp. p' and q'). Two paths $\pi_{p,q}$ and $\pi_{p',q'}$ may cross (Figure 2(a)) or not (Figures 2(b)-2(c)).

In the following, we show that a morphing from the configuration of guards represented by $\pi_{p,q}$ to that by $\pi(p', q')$ needs $O(k)$ motion actions. Assume that m_i is an edge of the polygon P. (The simple case that m_i is a vertex can be dealt with analogously.) So p and p' are mutualy visible, as they belong to the same edge m_i. Since one of m_j and m_{j+1} is an edge and the other is the vertex incident to that edge, two points q and q' are mutually visible, too. Let $k1 = d_L(p, q)$ and $k2 = d_L(p', q')$. Clearly, $|k1 - k2| \leq 2$ holds. Without loss of generality, assume that if $k1 < k$ (resp. $k2 < k$), then the guards $g_1, g_2, \cdots, g_{k-k1+1}$ (resp. $g_1, g_2, \cdots, g_{k-k2+1}$) are located at the starting point p (resp. p'), and other guards are located at all other vertices of the path $\pi_{p,q}$ (resp. $\pi_{p',q'}$), one per vertex.

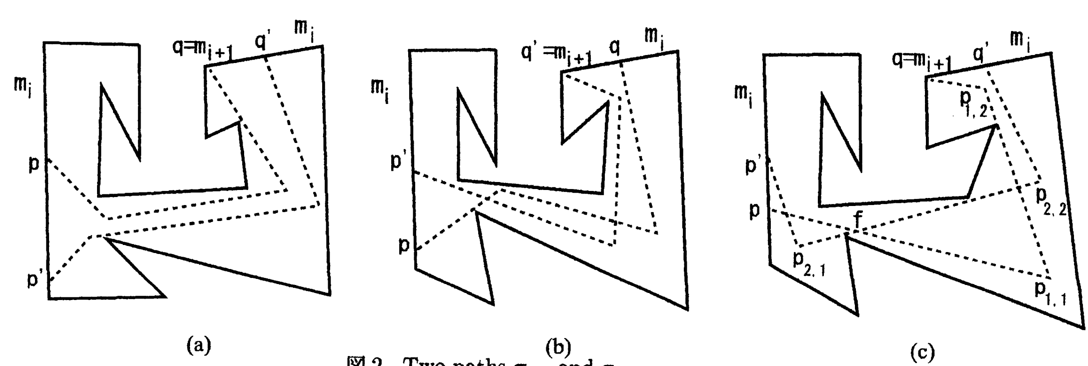


図2 Two paths $\pi_{p,q}$ and $\pi_{p',q'}$.

Suppose first that $k1 = k2$. Let $\pi_{p,q} = (p_{1,0}, p_{1,1}, \ldots, p_{1,k1})$ and $\pi_{p',q'} = (p_{2,0}, p_{2,1}, \ldots, p_{2,k1})$, with $p_{1,0} = p$, $p_{2,0} = p'$, $p_{1,k1} = q$ and $p_{2,k1} = q'$. The morphing strategy between $\pi_{p,q}$ and $\pi_{p',q'}$ is then performed by moving all the guards located at the vertices of $\pi_{p,q}$ to the corresponding vertices of $\pi_{p',q'}$. This can be shown by an induction on the number of the intersections, denoted by I, between $\pi_{p,q}$ and $\pi_{p',q'}$. Note that as the path $\pi_{p,q}$ or $\pi_{p',q'}$ is a minimum-link path, no point $p_{h,l}$ is visible from $p_{h,l-2}$ or $p_{h,l+2}$, for $h = 1$ or 2. If $I = 0$, then any two points $p_{1,l}$ and $p_{2,l}$, $0 \leq l \leq k1$, are mutually visible within the polygon P, and the morphing between $\pi_{p,q}$ and $\pi_{p',q'}$ can be done by moving all the guards along their corresponding segments $\overline{p_{1,l}p_{2,l}}$, for $0 \leq l \leq k1$, at time t simultaneously. See Figure 2(a) for an example. In the case $I > 0$, if the intersections occur only between the lth segments $\overline{p_{1,l-1}p_{1,l}}$ and $\overline{p_{2,l-1}p_{2,l}}$ of two paths, any two points $p_{1,l}$ and $p_{2,l}$, $1 \leq l \leq k1$, are also mutually visible, and thus the morphing between $\pi_{p,q}$ and $\pi_{p',q'}$ can similarly be done (Figure 2(b)). Suppose now that an intersection occurs between a lth segment of a path and a $(l+1)$-th segment of the other. (It is impossible for a lth segment of a path to intersect with the $(l+2)$-th segment of the other; otherwise, either $\pi_{p,q}$ or $\pi_{p',q'}$ is not a minimum-link path, a contradiction.) Let f denote the first of such intersections, say, the intersection between $\overline{p_{1,l-1}p_{1,l}}$ and $\overline{p_{2,l}p_{2,l+1}}$. See Figure 2(c) for an example. By induction hypothesis, we can first morph the part of $\pi_{p,q}$ from $p_{1,l+1}$ to q into the part of $\pi_{p',q'}$ from f to q'. (Note that the guard at $p_{1,l+1}$ is temporarily moved to f.) Next, we morph the part of $\pi_{p,q}$ from p to f into the part of $\pi_{p',q'}$ from p' to $p_{2,l}$. So we obtain a morphing between $\pi_{p,q}$ and $\pi_{p',q'}$. Since we only need to check the intersections along two paths $\pi_{p,q}$ and $\pi_{p',q'}$, our morphing takes $O(k)$ time.

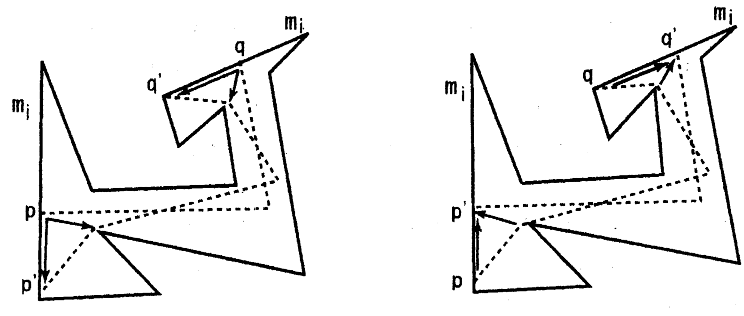


図 3 Illustration for the case $1 \leq |k1 - k2| \leq 2$.

Turn to the case $1 \leq |k1 - k2| \leq 2$. The difference between $k1$ and $k2$ occurs because an edge of P can be partitioned into at most three intervals of constant link distance from a point inside P (Figure 3). Since the numbers of the guards located at turn points of two paths are different in this case, two gaurds at the point p (resp. q') may move to the first (resp. last) two vertices of $\pi_{p',q'}$ (Figure 3(a)), or two guards on the path $\pi_{p,q}$ may move to the same point p' or q' (Figure 3(b)). Moreover, a guard may have to transfer along $\pi_{p.q}$ befor, or along $\pi_{p',q'}$ after the morphing between $\pi_{p,q}$ and $\pi_{p',q'}$.

Without loss of generality, assume that that the edge m_j (resp. m_i) is partitioned into three intervals of constant link distance to the point p (resp. q). Suppose first that q belongs to the middle interval of m_j and q' belongs to the other interval. In this case, we take the guard g_{k-k1+1} from p and move it to the first turn point of $\pi_{p,q}$. A guard is then transferred along $\pi_{p,q}$ until it reaches the point q. See Figure 3(a) for an example. Two guards located at q are then moved to q' and the last turn point of $\pi_{p',q'}$ by the following morphing between $\pi_{p,q}$ and $\pi_{p',q'}$. In the case that q' belongs to the middle interval of m_j and q' belongs to the other interval, the guard at the last turn point of $\pi_{p,q}$ is first moved to q' by the morphing of $\pi_{p,q}$ into $\pi_{p',q'}$, and then a guard is tranferred along $\pi_{p',q'}$, starting at q', until it reaches p'. See also Figure 3(b). Suppose now that p belongs to the middle interval of m_i and p' belongs to the other interval. In this case, the morphing between $\pi_{p,q}$ and $\pi_{p',q'}$ described above can be performed, except that the last of the guards currently located at p is moved to the first turn point of $\pi_{p',q'}$ (and all others are still moved to p'). See Figure 3(a). On the other hand, if p' belongs to the middle interval of m_i and p' belongs to the other interval, then the guard at the first turn point of $\pi_{p,q}$ is moved to p' (Figure 3(b)). Hence, the morphing of $\pi_{p,q}$ into $\pi_{p.q'}$ also takes $O(k)$ time in this case.

In conclusion, any st-path can be translated to a sweep schedule for P. This completes the proof. □

Lemma 3 It takes $O(kn^2)$ time to generate a sweep schedule (if it exists) in the graph G'.

Proof. When a node (m_i, m_j) of G' is constructed [7], we can arbitrarily take two points $p \in m_i$ and $q \in m_j$, such that they are mutually link-k visible, and find a minimum-link path between p and q. The partition of the edge m_j (resp. m_i) into at most three intervals of constant link distance to the point p (resp. q) can also be precomputed.

To construct a sweep schedule, we only need to find a simple st-path in the diagram G'. This can be done in by performing a reachability query in G'. Since the size of G' is $O(n^2)$, a simple st-path (if it exists) can be found in $O(n^2)$ time. Any edge of the st-path corresponds to a morphing between the minimum-link paths for two pairs of chosen points. Then, it follows the proof of Lemma 2 that the morphing of a configuration of guards into the other takes $O(k)$ time. □

Theorem 1 Given a simple polygon P, we can compute in $O(n^2)$ time the minimum number r^* of guards needed to sweep P. Moreover, a sweep schedule can be generated in $O(r^* n^2)$ time.

Proof. Using the approximation algorithm of Efrat et al. [1], we first compute in $O(n^2)$ time a number r, so that P can be swept with r guards and $r \leq r^* + 2$. By making use of Lemma 2 at most three times, for $k+1 = r, r-1$ and $r-2$, we can then determine the real value of r^*. Finally, it follows from Lemma 3 that a sweep schedule can be output in $O(r^* n^2)$ time. □

5 Conclusions

We have proposed an $O(n^2)$ time algorithm for computing the minimum number r^* of guards required to detect a moving target in a simple polygon, and an $O(r^* n^2)$ time algorithm for generating a sweep schedule. Our results improve upon the previously known time bounds by a linear factor.

It is worth pointing out that our algorithms run in $O(n^2)$ time, even when $k = 1$; In this case, it is well known as the two-guard problem [2, 3, 11]. An interesting work is then to find a linear time algorithm for determining whether a polygon is swept by two guards. Moreover, it is challenging to find a solution to the polygon search problem [8], without requiring that the guards always from a simple polygonal chain.

Acknowledgements

This research is partially supported by the Grant-in-Aid of the Ministry of Education, Science, Sports and Culture of Japan.

参考文献

[1] A.Efrat, L.J.Guibas, S. Har-Peled, D.C.Lin, J.S.B. Mitchell and T.M.Murali, Sweeping simple polygons with a chain of guards, In *Proc., ACM-SIAM SODA* (2000) 927-936.

[2] P.J.Heffernan, An optimal algorithm for the two-guard problem, *Int. J. Comput. Geom. & Appl.* **6** (1996) 15-44.

[3] C. Icking and R. Klein, The two guards problem, *Int. J. Comput. Geom. & Appl.* **2** (1992) 257-285.

[4] S.M.LaValle, B.Simov and G.Slutzki, An algorithm for searching a polygonal region with a flashlight, *Int. J. Comput. Geom. & Appl.* **12** (2002) 87-113.

[5] J.H. Lee, S.Y.Shin and K.Y.Chwa, Visibility-based pursuit-evasion in a polygonal room with a door, *Proc. 15th Annu. ACM Symp. Comput. Geom.* (1999) 281-290.

[6] J.H.Lee, S.M.Park and K.Y.Chwa, Searching a polygonal room with one door by a 1-searcher, *Int. J. Comput. Geom. & Appl.* **10** (2000) 201-220.

[7] S.Suri, On some link distance problems in a simple polygon, *IEEE Trans. Robotics Automation* **6** (1990) 108-113.

[8] I.Suzuki and M.Yamashita, Searching for mobile intruders in a polygonal region, *SIAM J. Comp.* **21** (1992) 863-888.

[9] I.Suzuki, Y.Tazoe, M.Yamashita and T.Kameda, Searching a polygonal region from the boundary, *Int. J. Comput. Geom. & Appl.* **11** (2001) 529-553.

[10] X.Tan, A characterization of polygonal regions searchable from the boundary, *Lect. Note Comput. Sci.* **3330** (2004) 200-215.

[11] X.Tan, The two-guard problem revisited and its generalization, *Lect. Note Comput. Sci.* **3341** (2004) 847-858.