# How to Keep an Eye on a Few Small Things

Bengt J. Nilsson*　　Paweł Żyliński†


## Abstract

We present a $(k+h)$-FPT algorithm for computing a shortest tour that sees $k$ specified points in a polygon with $h$ holes. We also present a $k$-FPT approximation algorithm for this problem having approximation factor $\sqrt{2}$. In addition, we prove that the general problem cannot be polynomially approximated better than by a factor of $\Omega(\log n)$, unless P=NP, where $n$ is the total number of edges of the polygon.


## 1 Introduction

The problem of computing a shortest tour that sees a specified set of objects in an environment of obstacles has a long history. The first results were published in 1986 [2] considering shortest tours that see monotone and simple rectilinear polygons [3]. For simple polygons, a sequence of articles establishes polynomial time solutions [4, 14, 13, 9, 1, 15, 12, 5].

In a polygon with holes, finding a shortest tour that sees the complete environment is NP-hard [3]. Mata and Mitchell [10] construct an approximation algorithm with logarithmic approximation factor and Dumitrescu and Tóth [7] provide upper bounds on the length of such tours in this setting.

Dumitrescu *et al.* [6] consider the shortest guarding tour among a set of non-parallel lines. Here the lines are seen as thin corridors and the objective is for a shortest tour to visit each line to see it. They show that the problem is polynomially tractable for lines in 2D but NP-hard for lines in 3D.

We consider the problem of guarding or covering a specified set of points positioned in a geometric domain with a closed curve. We call this problem the *shortest guarding tour problem*. We show that computing the shortest guarding tour in a polygon with holes cannot be approximated better than by a factor of $\Omega(\log n)$ in polynomial time unless P=NP. On the other hand, we show that there is a $(k+h)$-fixed parameter tractable algorithm for the problem, where $k$ is the number of points to be guarded and $h$ is the number of holes. We also show a $k$-fixed parameter tractable approximation algorithm for the problem having approximation factor $\sqrt{2}$.


*Department of Computer Science, Malmö University, SE-205 06 Malmö, Sweden. email: `bengt.nilsson.TS@mah.se`

†Institute of Informatics, University of Gdańsk, 80-952 Gdańsk, Poland. email: `zylinski@inf.ug.edu.pl`


## 2 Computing the Shortest Guarding Tour

Let $\mathbf{P}$ be a polygon with $h$ holes, having a total of $n$ edges and let $\mathcal{S} = \{p_1, \ldots, p_k\}$ be a set of $k$ points to be guarded in $\mathbf{P}$. We assume that all vertices of $\mathbf{P}$ and the points in $\mathcal{S}$ are in general position. Consider the visibility polygon of a point $\mathbf{V}(p)$, $p \in \mathcal{S}$. The boundary edges of $\mathbf{V}(p)$ consist of line segments, either collinear to edges of $\mathbf{P}$, or properly interior to $\mathbf{P}$ but connecting two boundary points. We call these latter segments *windows* of $\mathbf{V}(p)$. A window is *complete*, if it partitions $\mathbf{P}$ into two disconnected pieces, i.e., the two endpoints of the window belong to the same hole (or the outer boundary of $\mathbf{P}$). A complete window is *useless*, if the two components of the partition do not both contain points of $\mathcal{S}$. All other windows (also incomplete ones) are *useful*.

**Lemma 1** *The number of useful windows of* $\mathbf{V}(p)$ *is at most* $h(h+1)+k-1$.

**Proof.** Enumerate each hole from 1 to $h$ and let the outer boundary have index 0. Let the two endpoints of a window be indexed by the corresponding indices of their adjacent hole (or outer boundary of $\mathbf{P}$). We have two cases to consider. First, for each pair of different indices, it can be shown by induction on $h$, there can be only two windows having endpoints with these indices. This gives us at most $h(h+1)$ useful windows. Second, if the two window endpoints have the same index, this means that the window is complete and partitions $\mathbf{P}$ into different pieces. Since there are $k$ points in $\mathcal{S}$, at most $k-1$ complete windows can have points from $\mathcal{S}$ on both sides. Hence, the number of useful windows of $\mathbf{V}(p)$ is as stated. □

A shortest guarding tour, denoted $T^*$, that sees all the points in $\mathcal{S}$ is a shortest tour that intersects each of the visibility polygons $\mathbf{V}(p)$, $p \in \mathcal{S}$. Each subpath of $T^*$ between two consecutive visibility polygons $\mathbf{V}(p)$ and $\mathbf{V}(p')$ is a shortest path between points on useful windows of $\mathbf{V}(p)$ and $\mathbf{V}(p')$. One of the two component pieces of the interior of $\mathbf{P}$ partitioned by a useless window $w$, does not contain any points of $\mathcal{S}$. Hence, since subpaths of shortest paths are also shortest paths, $T^*$ will never properly intersect $w$ and we can therefore disregard any useless window.

The arrangement of the useful windows from all the visibility polygons $\mathbf{V}(p)$, $p \in \mathcal{S}$, consists of maximal line segments having window endpoints and window intersection points as endpoints. We call these maximal line segments *gates*. From Lemma 1, it follows that there are at most $k(h(h+1)+k-1)^2$ gates bounding a visibility polygon. For a point $p \in \mathcal{S}$, we denote by $\mathcal{G}(p)$ the set of gates being subsegments of useful windows of $\mathbf{V}(p)$. To a gate $g$ we also associate the set $\mathcal{B}(g)$ consisting of those points $p \in \mathcal{S}$ for which $g \subseteq \mathbf{V}(p)$. Every gate $g$ also has two *sides*, $s$ facing the interior, and $\bar{s}$ facing the exterior of the associated visibility polygon.

The tour $T^*$ visits the visibility polygons of $p \in \mathcal{S}$ in some order and does so by entering a visibility polygon through a gate $g$ from side $\bar{s}$, leaving $g$ from one of its two sides, and then moving to a gate $g'$ of the next visibility polygon using a shortest path, entering $g'$ through side $\bar{s}'$. Hence, in order to compute $T^*$, it suffices to establish the correct set of gates, their exit sides, their ordering as they are visited by $T^*$ and the correct intersection points between $T^*$ and the gates. Since there are few gates, we can do this by trying all possible configurations.

Let $\Gamma$ denote any set of at most one gate from each set $\mathcal{G}(p)$, $p \in \mathcal{S}$ such that $\bigcup_{g \in \Gamma} \mathcal{B}(g) = \mathcal{S}$. For every possible set $\Gamma$, every positive integer, $l \leq (|\Gamma| - 1)!$ and every non-negative integer $r \leq 2^{|\Gamma|}$, we compute a tour $T_{\Gamma,l,r}$. $\Gamma$ specifies the set of gates that $T_{\Gamma,l,r}$ will pass, $l$ specifies the ordering in which the visibility polygons are visited and $r$ specifies at which gates the tour makes reflection contact (or crossing contact). Given two gates $g_i$ and $g_{i'}$ such that $g_{i'} \notin \mathcal{G}(p)$, for any $p \in \mathcal{B}(g_i)$, we compute the shortest paths from the two endpoints of $\bar{s}_i$ to the two endpoints of $\bar{s}_{i'}$, and the shortest paths from the two endpoints of $s_i$ to the two endpoints of $\bar{s}_{i'}$, if they exist. This can be accomplished by considering the windows of $g_i$ and $g_{i'}$ to be thin obstacle walls connecting the holes at the window endpoints.

The two non-crossing paths from the endpoints of $\bar{s}_i$ to the endpoints of $\bar{s}_{i'}$ bound a polygonal region $\mathbf{t}^0_{i,i'}$ and the two non-crossing paths from the endpoints of $s_i$ to the endpoints of $\bar{s}_{i'}$ bound another polygonal region $\mathbf{t}^1_{i,i'}$; see Figure 1(a). We call these regions *tubes*. The portion of $T_{\Gamma,l,r}$ between $g_i$ and $g_{i'}$ must lie in $\mathbf{t}^0_{i,i'}$, if $T_{\Gamma,l,r}$ makes a reflection at $g_i$, and in $\mathbf{t}^1_{i,i'}$, if the tour crosses $g_i$ properly. In this way, we construct a sequence of tubes $\mathbf{t}^{j_1}_{i_1,i_2}, \mathbf{t}^{j_2}_{i_2,i_3}, \ldots, \mathbf{t}^{j_{|\Gamma|}}_{i_{|\Gamma|},i_1}$, with each $j_z = 0$ or $1$ depending on whether the tour reflects or crosses at the corresponding gate, that we glue together in sequence at the gates to obtain an *hourglass* $\mathbf{H}_{\Gamma,l,r,g_{i_1}}$ connecting $g_{i_1}$ in $\mathbf{t}^{j_1}_{i_1,i_2}$ with its mirror image $g_{i_1}$ in $\mathbf{t}^{j_{|\Gamma|}}_{i_{|\Gamma|},i_1}$; see Figure 1(b). Note that, to account for the reflection contact at a gate $g_i$, we glue the reflection of the tube $\mathbf{t}^0_{i,i'}$ along gate $g_i$ to the hourglass. In this way, an hourglass is a two-manifold possibly containing obstacles in which the shortest path from a point $g_{i_1}$ in $\mathbf{t}^{j_1}_{i_1,i_2}$ to its mirror image point on $g_{i_1}$ in

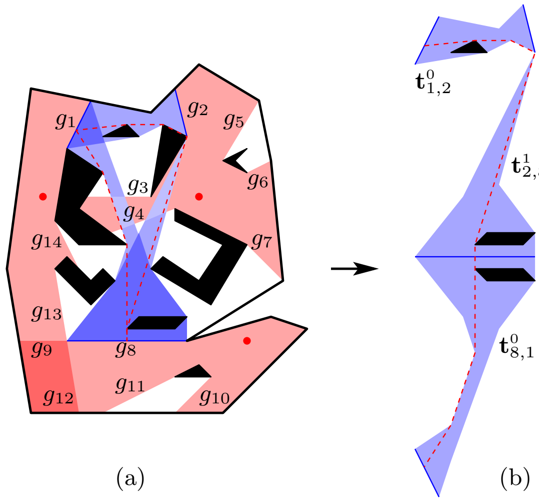


Figure 1: Illustrating the hourglass construction.

$\mathbf{t}^{j_{|\Gamma|}}_{i_{|\Gamma|},i_1}$ corresponds to the guarding tour $T_{\Gamma,l,r}$.

In the next section, we show how to compute the shortest path from a point $q$ on $g_{i_1}$ in $\mathbf{t}^{j_1}_{i_1,i_2}$ to its corresponding mirror image on $g_{i_1}$ in $\mathbf{t}^{j_{|\Gamma|}}_{i_{|\Gamma|},i_1}$ in $O(k^4n^4)$ time, given the hourglass $\mathbf{H}_{\Gamma,l,r,g_{i_1}}$. This path can then be folded at the appropriate reflection gates by establishing the intersection points between the path and the gates in $\Gamma$ to obtain the guarding tour.

The number of possible sets $\Gamma$ is bounded by $(k(h(h+1)+k)^2)^k$, the number of orderings of the visibility polygons is $(|\Gamma|-1)! \leq (k-1)!$ and the number of choices for reflection or crossing is $2^{|\Gamma|} \leq 2^k$.

**Theorem 2** *A shortest guarding tour for $k$ points in a polygon with $h$ holes is computed by the algorithm in $k!2^k k^{k+3}(h(h+1)+k)^{2k} \cdot O(n^4)$ time.*

## 2.1 The Sliding Process

Given an hourglass $\mathbf{H}_g$ connecting a gate $g$ in the first tube of $\mathbf{H}_g$ with the image of $g$ in the last tube of $\mathbf{H}_g$, we call it $g'$. Each tube of $\mathbf{H}_g$ has complexity $O(n)$ and, since $\mathbf{H}_g$ consists of at most $2k$ tubes glued together, $\mathbf{H}_g$ has complexity $O(kn)$.

To compute the parameterized shortest path $\Pi(q)$ from every point $q$ on $g$ to its image $q'$ on $g'$, we begin by computing the shortest paths in $\mathbf{H}_g$ between all vertices visible from $g$ to all vertices visible from $g'$. This takes $O(k^4n^4)$ time. Let $q$ be one endpoint of $g$ and let $q'$ be its mirror on $g'$. Connect $q$ and $q'$ to each visible vertex in $\mathbf{H}_g$; see Figure 2(a). This gives us $O(k^2n^2)$ paths connecting $q$ with $q'$. As we slide $q$ and $q'$ along $g$ and $g'$, we maintain all the paths connecting the points with vertices visible to them. Any such path has length $||q,v|| + ||SP(v,v')|| + ||v',q'||$, where $SP(v,v')$ is the shortest path between vertices $v$ and $v'$. For each point $q$ during the sliding process, we also maintain the shortest of all the paths $\Pi(q)$.

As the sliding proceeds, we have to update the path $\Pi(q)$ when structural changes occur. This happens 1) when $\Pi(q)$ leaves a vertex where a turn of the path

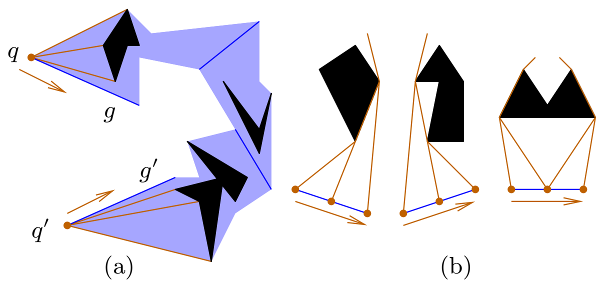


Figure 2: Illustrating the sliding process.

occurs, 2) when $\Pi(q)$ hits a new vertex, and 3) when a $\Pi(q)$ makes a complete subpath change; see Figure 2(b). The two first update cases occur $O(kn)$ times and the third case occurs $O(k^2n^2)$ times, since each path $SP(v, v')$ can be a subpath of $\Pi(q)$ at most once, for every pair $v$ and $v'$ of vertices visible to $q$ and $q'$. In this way, we obtain the parameterized path function $\Pi(q)$ for all points $q$ on $g$ in $\mathbf{H}_g$. Between any two update points, $||\Pi(q)||$ can have an optimum at most once and we can obtain this by differentiating the distance function $||\Pi(q)||$ on $q$. Thus, we can in $O(k^2n^2)$ time obtain the point $q^*$ for which $||\Pi(q^*)|| \leq ||\Pi(q)||$, for all points $q \in g$.

## 3 Approximating the Shortest Guarding Tour

We can trade computation time for accuracy in the algorithm above by using dynamic programming to reduce the number of configurations. For each point $p \in \mathcal{S}$ and each pair of gates $g_i$ and $g_{i'}$, with $g_i \in \mathcal{G}(p)$ and $g_{i'} \notin \mathcal{G}(p)$, we compute the shortest path $\pi_{i,i'}$ from $g_i$ to $g_{i'}$. For both gates $g_i$ or $g_{i'}$, the path either connects to one of the endpoints of the gate, or it is orthogonal to it. Let $d_{i,i'}$ be the length of $\pi_{i,i'}$, let $e_{i'',i,i'}$ be the segment between the intersection points of $\pi_{i'',i}$ and $\pi_{i,i'}$ on $g_i$ and let $\delta_{i'',i,i'}$ denote the length of $e_{i'',i,i'}$. The computation of these paths takes $(k(h(h+1)+k))^4 \cdot O(n^2) \subseteq (k^4h^8 + k^8) \cdot O(n^2)$.

Let $g_{i_1}$ be a starting gate for the guarding tour and $g_{i_2}$ some other gate. Let $\mathcal{T}$ be some subset of the points in $\mathcal{S} - (\mathcal{B}(g_{i_1}) \cup \mathcal{B}(g_{i_2}))$. Let $\mathcal{L}(\mathcal{T}, g_{i'}, g_i)$ denote the length of the shortest sequence of paths,

$$\pi_{i_1,i_2}, e_{i_1,i_2,i_3}, \pi_{i_2,i_3}, \ldots, \pi_{i',i}, e_{i',i,i_1}, \pi_{i,i_1}, e_{i,i_1,i_2},$$

forming a tour that starts at $g_{i_1}$, passes $g_{i_2}$, intersects all the visibility polygons of the points in $\mathcal{T}$, ends at $g_i$ via the gate $g_{i'}$ and goes back to $g_{i_1}$. $\mathcal{L}(\mathcal{T}, g_{i'}, g_i)$ is given recursively as

$$\begin{aligned}\mathcal{L}(\mathcal{T}, g_{i'}, g_i) = \min_{\substack{g_i \notin \mathcal{G}(p) \\ p \in \mathcal{T} - \mathcal{B}(g_i)}} & \{\mathcal{L}(\mathcal{T} - \mathcal{B}(g_i), g_{i''}, g_{i'}) \\ & - \delta_{i'',i',i_1} - d_{i',i_1} - \delta_{i',i_1,i_2} \\ & + \delta_{i'',i',i'} + d_{i',i} + \delta_{i',i,i_1} + d_{i,i_1} + \delta_{i,i_1,i_2}\}.\end{aligned}$$

Performing the dynamic programming requires $(k(h(h+1)+k))^4$ tables of size $(k(h(h+1)+k))^4 \cdot 2^k$, where each position is filled in according to the recursion above, so the complexity of this part is bounded by $(k^8h^{16} + k^{16}) \cdot 2^k$ steps. Adding the time for preprocessing and the fact that $h \leq n$, we can prove the following theorem.

**Theorem 3** *An approximate shortest guarding tour for $k$ points in a polygon with $h$ holes having approximation factor $\sqrt{2}$ is computed by the dynamic programming algorithm in $2^k \cdot O(k^{16} + k^8n^{16})$ time.*

It remains to show the approximation factor. Consider the sequence of gates that the shortest guarding tour $T^*$ intersects. If we, for each gate $g$, replace the segments of $T^*$ incident to $g$, with the shortest segment to $g$ possibly followed by a segment along along $g$, we obtain a new tour $T_r$. The, at most, two segments incident to $g$ are replaced with axis parallel segments in a coordinate system where $g$ is parallel to the $x$-axis. For any sequence of gates, we say that such a tour has the *rectilinearity property*. The detour of $T_r$ is bounded by the length of two sides of a rectangle connecting the segment endpoints not on $g$, which in turn is at most a factor $\sqrt{2}$. The algorithm computes a shortest tour having the rectilinearity property, thus having length bounded by that of $T_r$.

## 4 Inapproximation of the Shortest Guarding Tour

To guard a discrete set of points in a polygon with holes using a shortest tour is NP-hard as can be shown with a reduction from TSP [3]. We show a gap preserving reduction from Set Cover to our guarding problem, essentially modifying the construction of Eidenbenz *et al.* [8] to prove that approximating our guarding problem within a logarithmic factor is NP-hard in general [11]. Let $(\mathcal{X}, \mathcal{F})$ be a set system with $\mathcal{X} = \{x_1, \ldots, x_k\}$ a set of $k$ items and $\mathcal{F} = \{F_1, \ldots, F_m\}$ a family of $m$ sets containing the items in $\mathcal{X}$, i.e., each $F_i \subseteq \mathcal{X}$. We transform this instance into a polygon $\mathbf{P}$ with holes and a set of points $\mathcal{S}$ to be guarded.

Given a bipartite graph representing the items in $\mathcal{X}$ and the sets in $\mathcal{F}$; see Figure 3(a). We build $\mathbf{P}$ as follows: construct $2k+1$ points evenly spaced along a parabola and connect the points to form a path, denoted $\Pi$. The path $\Pi$ forms the lower boundary of $\mathbf{P}$. Identify the points on $\Pi$ having even index with the items $x_1, \ldots, x_k \in \mathcal{X}$. Above $\Pi$ construct $m$ points corresponding to each set in $\mathcal{F}$ evenly spaced along a horizontal line segment $L$ and connect the left and right endpoints of $L$ with the left and right endpoints of $\Pi$ respectively and connect the left and right endpoints of $L$ with a point $q$ slightly above $L$ and to the left of the left endpoint of $L$. At $q$, $\mathbf{P}$ has an extra notch with an additional point $x_0$ at the bottom vertex; see Figure 3(b). We fill the region inside the polygon with holes in such a way that $q$ sees $x_0$ and the points corresponding to each $F_j \in \mathcal{F}$ and each $x_j$ sees $F_j$ if and only if $x_i \in F_j$.

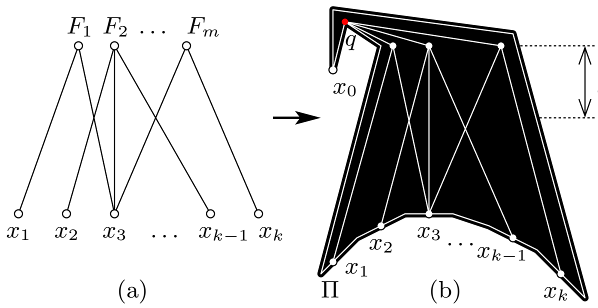


Figure 3: Illustrating the reduction from Set Cover.

To finalize the construction, let $d$ and $d'$ denote the distance from $q$ to the furthest and closest among the points corresponding to $F_j \in \mathcal{F}$ respectively. The visibility lines connecting $x_i$ and $F_j$, if $x_i \in F_j$, can be seen as thin corridors making up the interior of the polygon. These corridors can intersect and thus determine regions where more than one item $x_i$ can be seen. We call these regions *X-regions*. Let $l$ denote the difference in height between the highest $X$-region and the horizontal line segment $L$. By placing $q$ sufficiently far to the left of $L$ and then placing $L$ sufficiently high above $\Pi$, we can guarantee that $d(m-1) < d'm$ and $dm < l$ ($m = |\mathcal{F}|$).

The construction can be built in polynomial time and fits in a polynomially sized bounding box with integer vertex coordinates for $\mathbf{P}$. Our instance of the shortest guarding tour problem consists of the polygon $\mathbf{P}$ and the set $\mathcal{S}$, the $k+1$ vertices corresponding to $x_0, \ldots, x_k$.

Let $\mathcal{F}^*$ be an optimal solution to the set cover instance $(\mathcal{X}, \mathcal{F})$. We construct a solution to the shortest guarding tour problem in $\mathbf{P}$ seeing the points $x_0, \ldots, x_k$ as follows: from $q$ visit each of the points corresponding to the sets $F_j \in \mathcal{F}^*$ in order from left to right along $L$, each time going back to $q$. The length of the tour constructed is at least $2d'|\mathcal{F}^*|$ and at most $2d|\mathcal{F}^*|$ and it sees each of the points $x_i \in \mathcal{X}$ in addition to $x_0$. No other tour that sees these points can have shorter length since either 1) it corresponds to a non-optimal solution to the set cover instance, or 2) it must go below the regions where the visibility lines between points of $F_j$ and points of $x_i$ intersect each other, thus having length at least $2l > 2dm \geq 2d|\mathcal{F}^*|$.

Similarly, any shortest guarding tour for $x_0$ and the points corresponding to the items in $\mathcal{X}$ must visit the points corresponding to the sets $F_j \in \mathcal{F}^*$, hence from the tour we can obtain these sets and return the optimal solution to the set system $(\mathcal{X}, \mathcal{F})$.

Since the reduction is gap preserving, the approximation ratio for our tour problem is also $\Omega(\log m) = \Omega(\log n)$, where $n$ is the total number of edges. To see this, note that we can assume that $k \in \Theta(m^c)$, for some constant $c$. The number of holes is bounded by $(k+1)(m+1)$, each hole has at most $mk+6$ edges, and the outer boundary has $2k+7$ edges. Hence, $\Omega(m^c) \ni 2k+7 \leq n \leq (mk+6)(k+1)(m+1)+2k+7 \in O(m^{2+2c})$, proving our bound.

**Theorem 4** *A shortest guarding tour for a discrete set of points in a polygon with holes cannot be approximated in polynomial time with an approximation factor of $\Omega(\log n)$ unless P=NP, where $n$ is the total number of edges of the polygon.*

## References


[1] S. Carlsson, H. Jonsson, and B.J. Nilsson. Finding the shortest watchman route in a simple polygon. *Discrete and Computational Geometry*, 22:377–402, 1999.

[2] W. Chin and S. Ntafos. Optimum watchman routes. In *Proc. 2nd ACM SoCG*, pages 24–33, 1986.

[3] W. Chin and S. Ntafos. Optimum watchman routes. *Information Processing Letters*, 28:39–44, 1988.

[4] W. Chin and S. Ntafos. Shortest watchman routes in simple polygons. *Discrete and Computational Geometry*, 6(1):9–31, 1991.

[5] M. Dror, A. Efrat, A. Lubiw, and J. Mitchell. Touring a sequence of polygons. In *Proc. 35th ACM STOC*, pages 473–482, 2003.

[6] A. Dumitrescu, J. Mitchell, and P. Żyliński. Watchman routes for lines and segments. *Computational Geometry*, 47(4):527–538, 2014.

[7] A. Dumitrescu and C. Tóth. Watchman tours for polygons with holes. *Computational Geometry: Theory and Applications*, 45(7):326–333, 2012.

[8] S. Eidenbenz, C. Stamm, and P. Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001.

[9] M. Hammar and B.J. Nilsson. Concerning the time bounds of existing shortest watchman route algorithms. In *Proc. 11th FCT*, LNCS 1279, pages 210–221, 1997.

[10] C. Mata and J. Mitchell. Approximation algorithms for geometric tour and network design problems. In *Proc. 11th ACM SoCG*, pages 360–369, 1995.

[11] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proc. 29th ACM STOC*, pages 475–484, 1997.

[12] X.-H. Tan. Fast computation of shortest watchman routes in simple polygons. *Information Processing Letters*, 77(1):27–33, 2001.

[13] X.-H. Tan and T. Hirata. Constructing shortest watchman routes by divide and conquer. In *Proc. 4th ISAAC*, pages 68–77. LNCS 762, 1993.

[14] X.-H. Tan, T. Hirata, and Y. Inagaki. An incremental algorithm for constructing shortest watchman routes. *International Journal of Computational Geometry and Applications*, 3:351–365, 1993.

[15] X.-H. Tan, T. Hirata, and Y. Inagaki. Corrigendum to "an incremental algorithm for constructing shortest watchman routes". *International Journal of Computational Geometry and Applications*, 9(3):319–324, 1999.