An Approximate Algorithm for Solving the Watchman Route Problem

Fajie Li and Reinhard Klette

[1] Institute for Mathematics and Computing Science, University of Groningen
P.O. Box 800, 9700 AV Groningen, The Netherlands
[2] Computer Science Department, The University of Auckland
Private Bag 92019, Auckland 1142, New Zealand

Abstract. The watchman route problem (WRP) was first introduced in 1988 and is defined as follows: How to calculate a shortest route completely contained inside a simple polygon such that any point inside this polygon is visible from at least one point on the route? So far the best known result for the WRP is an $\mathcal{O}(n^3 logn)$ runtime algorithm (with inherent numerical problems of its implementation). This paper gives an $\kappa(\varepsilon) \times \mathcal{O}(kn)$ approximate algorithm for WRP by using a rubberband algorithm, where n is the number of vertices of the simple polygon, k a number of essential cuts and ε the chosen accuracy constant.

Keywords: *computational geometry, simple polygon, Euclidean shortest path, visual inspection, Watchman Route Problem, rubberband algorithm.*

1 Introduction

There are a number of computational geometry problems which involve finding a shortest path [24], for example, the safari problem, zookeeper problem, or watchman route problem (WRP). All are of obvious importance for robotics, especially the WRP for visual inspection. This paper presents algorithms for solving the touring polygons problem (TPP), parts cutting problem, safari problem, zookeeper problem, and watchman route problem. These problems are closely related to one-another. A solution to the first problem implies solutions to the other four problems. The safari, zookeeper, and watchman route problems belong to the class of art gallery problems [40].

1.1 Touring Polygons Problem

We recall some notation from [13], which introduced the touring polygons problem. Let π be a plane, which is identified with $\mathbb{R}^2$. Consider polygons $P_i \subset \pi$, where $i = 1, 2, \ldots, k$, and two points p, $q \in \pi$. Let $p_0 = p$ and $p_{k+1} = q$. Let $p_i \in \mathbb{R}^2$, where $i = 1, 2, \ldots, k$. Let $\rho(p, p_1, p_2, , \ldots, p_k, q)$ denote the path $pp_1p_2 \ldots p_kq \subset \mathbb{R}^2$. Let $\rho(p, q) = \rho(p, p_1, p_2, , \ldots, p_k, q)$ if this does not cause any confusion.

If $p_i \in P_i$ such that p_i is the first (i.e., along the path) point in $\partial P_i \cap \rho(p, p_i)$, then we say that path $\rho(p, q)$ *visits* P_i at p_i, where $i = 1, 2, \ldots, k$.

Let $A^\bullet$ be the topologic closure of set $A \subset \mathbb{R}^2$. Let $F_i \subset \mathbb{R}^2$ be a simple polygon such that $(P_i^\bullet \cup P_{i+1}^\bullet) \subset F_i^\bullet$; then we say that F_i is a *fence* [with respect to P_i and P_{i+1} (mod $k+1$)], where $i = 0, 1, 2, \ldots, k+1$. Now assume that we have a fence F_i for any pair of polygons P_i and P_{i+1}, for $i = 0, 1, \ldots, k+1$.

The *constrained* TPP is defined as follows: *How to find a shortest path* $\rho(p, p_1, p_2, , \ldots, p_k, q)$ *such that it visits each of the polygons* P_i *in the given order, also satisfying* $p_i p_{i+1} (mod\ k+1) \subset F_i^\bullet$*, for* $i = 1, 2, \ldots, k$*?*

Assume that for any i, $j \in \{1, 2, \ldots, k\}$, $\partial P_i \cap \partial P_j = \emptyset$, and each P_i is convex; this special case is dealt with in [13]. The given algorithm runs in $\mathcal{O}(kn\ log(n/k))$ time, where n is the total number of all vertices of all polygons $P_i \subset \pi$, for $i = 1, 2, \ldots, k$. Let Π be a simple polygon with n vertices.

The *watchman route problem* (WRP) is discussed in [4], and it is defined as follows: *How to calculate a shortest route* $\rho \subset \Pi^\bullet$ *such that any point* $p \in \Pi^\bullet$ *is visible from at least one point on the path?*

This is actually equivalent to the requirement, that all points $p \in \Pi^\bullet$ are visible just from the vertices of the path ρ, that means, for any $p \in \Pi^\bullet$ there is a vertex q of ρ such that $pq \subset \Pi^\bullet$; see Figure 1. If the start point of the route is given, then this refined problem is known as the *fixed* WRP.

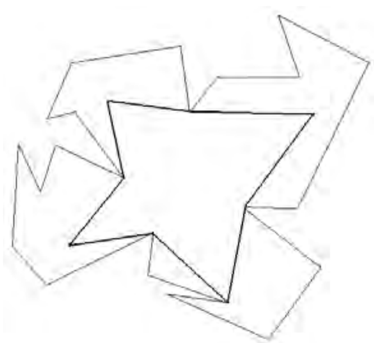

Fig. 1. A watchman route.

A simplified WRP was first solved in 1988 in [10] by presenting an $\mathcal{O}(n\ loglog\ n)$ algorithm to find a shortest route in a simple isothetic polygon. In 1991, [11] claimed to have presented an $\mathcal{O}(n^4)$ algorithm, solving the fixed WRP. In 1993, [32] obtained an $\mathcal{O}(n^3)$ solution for the fixed WRP. In the same year, this was further improved to a quadratic time algorithm [33]. However, four years later, in 1997, [17] pointed out that the algorithms in both [11] and [32] were flawed, but presented a solution for fixing those errors. Interestingly, two years later, in 1999, [34] found that the solution given by [17] was also flawed! By modifying the (flawed) algorithm presented in [32]. [34] gave an $\mathcal{O}(n^4)$ runtime algorithm for the fixed WRP.

In 1995, [7] proposed an $\mathcal{O}(n^{12})$ runtime algorithm for the WRP. In the same year, [27] gave an $\mathcal{O}(n^6)$ algorithm for the WRP. This was improved in 2001 by an $\mathcal{O}(n^5)$ algorithm in [35]; this paper also proved the following

Theorem 1. *There is a unique watchman route in a simple polygon, except for those cases where there is an infinite number of different shortest routes, all of equal length.*

So far the best known result for the WRP is due to [13] which gave in 2003 an $\mathcal{O}(n^3 \; log \; n)$ runtime algorithm.

Given the time complexity of those algorithms for solving the WRP, finding efficient (and numerically stable) approximation algorithm became an interesting subject. In 1995, [21] gave an $\mathcal{O}(log \; n)$-approximation algorithm for solving the WRP. In 1997, [8] gave a 99.98-approximation algorithm with time complexity $\mathcal{O}(n \; log \; n)$ for the WRP. In 2001, [36] presented a linear-time algorithm for an approximative solution of the fixed WRP such that the length of the calculated watchman route is at most twice of that of the shortest watchman route. The coefficient of accuracy was improved to $\sqrt{2}$ in [38] in 2004. Most recently, [39] presented a linear-time algorithm for the WRP for calculating an approximative watchman route of length at most twice of that of the shortest watchman route.

There are several generalizations and variations of watchman route problems; see, for example, [5, 6, 9, 12, 14–16, 18, 23, 26–30]. [1–3] show that some of these problems are NP-hard and solve them by approximation algorithms.

The rest of this paper is organized as follows: Section 2 recalls and introduces useful notions. Section **??** lists some known results which are applied later in this paper. Section 3 describes the new algorithms. Section 5 analyzes the time complexity of these algorithms. Section 4 proves the correctness of the algorithms. Section 6 concludes the paper.

2 Definitions and Known Results

We recall some definitions from [39]. Let Π be a simple polygon. The vertex v of Π is called *reflex* if v's internal angle is greater than 180°. Let u be a vertex of

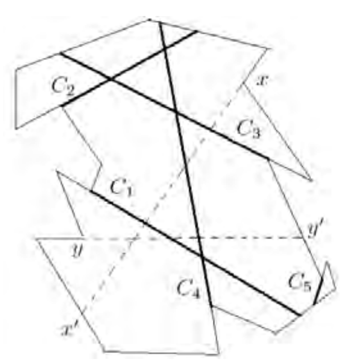

Fig. 2. Examples for cuts and essential cuts.

Π such that it is adjacent to a reflex vertex v. Let the straight line uv intersect an edge of Π at v'. Then the segment $C = vv'$ partitions Π into two parts. C is called a *cut* of Π, and v is called a *defining vertex* of C. That part of Π is called an *essential* part of C if it does not contain u; it is denoted by $\Pi(C)$. A cut C *dominates* a cut C' if $\Pi(C)$ contains $\Pi(C')$. A cut is called *essential* if it is not dominated by another cut. In Figure 2 (which is Figure 1 in [39]), the cuts xx' and yy' are dominated by C_2 and C_5, respectively; the cuts C_1, C_2, C_3, C_5 and C_4 are essential. Let $\mathcal{C}$ be the set of all essential cuts. The WRP is reduced to find the shortest route ρ such that ρ visits in order each cut in $\mathcal{C}$ (see Lemma 1, and also see [2] or [6]).

Let $S_C = \{C_1, C_2, \ldots, C_k\}$ be the sorted set $\mathcal{C}$ such that v_i is a defining vertex of C_i, and vertices v_i are located in anti-clockwise order around $\partial\Pi$, for $i = 1, 2, \ldots, k$.

Definition 1. *Let Π and S_C be the input of a watchman route problem. This problem is simplified iff, for each $i \in \{1, 2, \ldots, k-1\}$, for each $p_i \in C_i$ and $p_{i+1} \in C_{i+1}$ we have that $p_i p_{i+1} \cap \rho(v_{i+1}, \Pi, v_i) = \emptyset$.*

We lists some useful results used in the rest of this paper. The first one is about "order", and the remaining three are about time complexity.

Lemma 1. ([10], Lemma 3.3) *A solution to the watchman route problem (shortest tour) must visit the P_i's in the same order as it meets $\partial\Pi$.*

Lemma 2. ([24], pages 639–641) *There exists an $\mathcal{O}(n)$ time algorithm for calculating the shortest path between two points in a simple polygon.*

Lemma 3. ([31]) *The two straight lines, incident with a given point and being tangents to a given convex polygon, can be computed in $\mathcal{O}(log\ n)$, where n is the number of vertices of the polygon.*

Theorem 2. ([39], Theorem 1) *Given a simple polygon Π, the set $\mathcal{C}$ of all essential cuts for the watchman routes in Π can be computed in $\mathcal{O}(n)$ time.*

3 The Algorithms

3.1 A Local Solution for the Constrained TPP

The following procedure handles a degenerate case (see also Section 7.5 of [20]) of the rubberband algorithm, to be discussed later in this paper as "Algorithm 1". Such a case may occur and should be dealt with when we apply Algorithm 1 to the unconstrained TPP when the polygons are not necessarily pairwise disjoint. See Figure 3.

Procedure 1

Input: A point p and two polygons P_1 and P_2 such that $p \in \partial P_1 \cap \partial P_2$.
Output: A point $q \in \partial P_1$ such that $d_e(q, p) \leq \varepsilon$ and $q \notin \partial P_2$.

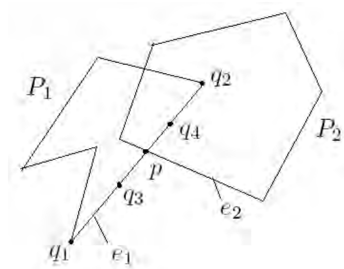

Fig. 3. Illustration for Procedure 1.

1. Let $\varepsilon = 10^{-10}$ (the accuracy).
2. Find a point $e_j \in E(P_j)$, where $j = 1, 2$, such that $p \in e_1 \cap e_2$.
3. Let $e_1 = q_1q_2$. Let q_3 and q_4 be two points in two segments q_1p and q_2p, respectively (see Figure 3) such that $d_e(q_j, p) \leq \varepsilon$ and $q_j \notin \partial P_2$, where $j = 3, 4$.
4. Let $q = \min\{q_3, q_4\}$ (with respect to lexicographic order).
5. Output q.

The following Procedure 2 is used in Procedure 3 which will be called in Algorithm 1 below.

Procedure 2

Input: Two polygons P_1 and P_2 and their fence F; two points $p_i \in \partial P_i$, where $i = 1, 2$, such that $p_1p_2 \cap \partial F = \{q_1, \ldots, q_2\}$.
Output: The set of all vertices of the shortest path, which starts at p_1, then visits ∂F, and finally ends at p_2 (but not including p_1 and p_2).

1. Compute $\rho(q_1, F, q_2)$, which is the subpath from q_1 to q_2 inside of F.
2. Apply the Melkman algorithm (see [22]) to compute the convex path from q_1 to q_2 inside of F, denoted by $\rho(q_1, q_2)$.
3. Compute a tangent p_it_i of p_i and $\rho(q_1, q_2)$ such that $p_it_i \cap F = t_i$, where $i = 1, 2$ (see [31]).
4. Compute $\rho(t_1, F, t_2)$, which is the subpath from t_1 to t_2 inside of F.
5. Output $V(\rho(t_1, F, t_2))$.

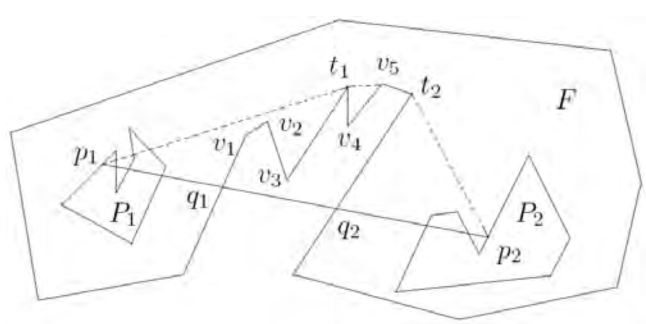

Fig. 4. Illustration for Procedure 2.

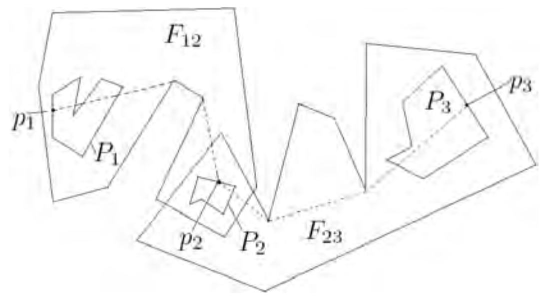

Fig. 5. Illustration for Procedure 3.

See Figure 4 for an example: $V(\rho(q_1, F, q_2)) = \{q_1, v_1, v_2, v_3, t_1, v_4, v_5, t_2, q_2\}$ and $V(\rho(t_1, F, t_2)) = \{t_1, v_4, v_5, t_2\}$. – The following Procedure 3 will be called in Step 4 in Algorithm 1 below.

Procedure 3

Input: Three polygons P_1, P_2 and P_3 in order, the fence of P_1 and P_2, denoted by F_{12}, the fence of P_2 and P_3, denoted by F_{23}, and three points $p_i \in \partial P_i$, where $i = 1, 2, 3$.
Output: The set of all vertices of the shortest path which starts at p_1, then visits P_2, and finally ends at p_3 (see Figure 5).

1.1. Let $\varepsilon = 10^{-10}$ (the accuracy).
1.2. If $(p_2 = p_1 \wedge p_2 \neq p_3) \vee (p_2 \neq p_1 \wedge p_2 = p_3) \vee (p_2 = p_1 \wedge p_2 = p_3)$, then apply Procedure 1 to compute a point to update p_2 such that $p_2 \neq p_1$ and $p_2 \neq p_3$.
1.3. Compute a point $p_2 \in \partial P_2$ such that

$$d_e(p_1, p_2') + d_e(p_2', p_3) = \min\{d_e(p_1, p') + d_e(p', p_3) : p' \in \partial P_2\}$$

2. Update V by letting $p_2 = p_2'$.
3. If $p_1p_2 \cap F_{12} = \emptyset$, then let $V_{12} = \emptyset$.
4. Otherwise suppose that $p_1p_2 \cap F_{12} = \{q_1, \ldots, q_2\}$.
5. Use P_1, P_2, F_{12}, q_1, and q_2 as input for Procedure 2; the output equals V_{12}.
6. Analogously, compute a set V_{23} from p_2, p_3 and F_{23}, as follows:
6.1. If $p_2p_3 \cap F_{23} = \emptyset$, then let $V_{23} = \emptyset$.
6.2. Otherwise suppose that $p_2p_3 \cap F_{23} = \{q_2', \ldots, q_3'\}$.
6.3. Use P_2, P_3, F_{23}, q_2', and q_3' as input for Procedure 2; the output equals V_{23}.
7. Let $V = \{v_1\} \cup V_{12} \cup \{v_2\} \cup V_{23} \cup \{v_3\}$.
8. Find q_1 and $q_3 \in V$ such that $\{q_1, p_2, q_3\}$ is a subsequence of V.
9.1. If $(p_2 = q_1 \wedge p_2 \neq q_3) \vee (p_2 \neq q_1 \wedge p_2 = q_3) \vee (p_2 = q_1 \wedge p_2 = q_3)$, then apply Procedure 1 to compute an update of point p_2 such that $p_2 \neq q_1$ and $p_2 \neq q_3$.

9.2. Find a point $p'_2 \in \partial P_2$ such that

$$d_e(q_1, p'_2) + d_e(p'_2, q_3) = \min\{d_e(q_1, p') + d_e(p', q_3) : p' \in \partial P_2\}$$

10. Update set V by letting $p_2 = p'_2$.
11. Output V.

Note that in Steps 1.2 and 9.1, the updated point p_2 depends on the chosen value of ε. – The following algorithm is for the constrained TPP.

Algorithm 1

1. For each $i \in \{0, 1, \ldots, k-1\}$, let p_i be a point on ∂P_i.
2. Let $V = \{p_0, p_1, \ldots, p_{k-1}\}$.
3. Calculate the perimeter L_0 of the polygon, which has the set V of vertices.
4. For each $i \in \{0, 1, \ldots, k-1\}$, use P_{i-1}, P_i, P_{i+1} (mod k) and F_{i-1}, F_i (mod k) (note: these are the fences of P_{i-1} and P_i, and P_i and P_{i+1} (mod k), respectively) as input for Procedure 3, to update p_i and output a set V_i.
5. Let $V_1 = V$ and update V by replacing $\{p_{i-1}, \ldots, p_i, \ldots, p_{i+1}\}$ by V_i.
6. Let $V = \{q_0, q_1, \ldots, q_m\}$.
7. Calculate the perimeter L_1 of the polygon, which has the set V of vertices.
8. If $L_1 - L_0 > \varepsilon$, then let $L_0 = L_1$, $V = V_1$, and go to Step 4. Otherwise, output the updated set V and (its) calculated length L_1.

If a given constrained TPP is not a simplified constrained TPP, then we can slightly modify Procedure 3: just replace "Procedure 2" in Step 5 of Procedure 3 by the algorithm in [24] on pages 639–641. In this way, Algorithm 1 will still work also for solving a non-simplified constrained TPP locally.

3.2 Solution for the Watchman Route Problem

The algorithm for solving the simplified constrained TPP (or the "general" constrained TPP) implies the following algorithm for solving the simplified WRP (or the "general" WRP).

Algorithm 2

We modify Algorithm 1 at two places:

1. Replace polygon "P_j" by segment "s_j" in Steps 1 and 4, for $j = i-1$, i, or $i+1$.
2. Replace "F_{i-1}, F_i (mod k)" (the fences of P_{i-1} and P_i, and P_i and P_{i+1}, all mod k, respectively) by the common polygon Π.

4 Proof of Correctness

A *single iteration* of a w-rubberband algorithm is a complete run through the main loop of the algorithm. – Let Π be a simple polygon.

Definition 2. *Let $P = (p_0, p_1, p_2, \ldots, p_m, p_{m+1})$ be a critical point tuple of Π. Using P as an initial point set, and n iterations of the w-rubberband algorithm, we get another critical point tuple of Π, say $P' = (p'_0, p'_1, p'_2, \ldots, p'_m, p'_{m+1})$. The polygon with vertex set $\{p'_0, p'_1, p'_2, \ldots, p'_m, p'_{m+1}\}$ is called an* nth polygon *of Π, denoted by $AESP_n(\Pi)$, or (for short) by $AESP_n$, where n = 1, 2, ...*

Let p, $q \in \Pi^{\bullet}$. Let $d_{ESP}(\Pi, p, q)$ be the length of the shortest path between p and q inside of Π. Let $AESP_n(\Pi)$ be an nth polygon fully contained inside of Π, where $n \geq 1$. Let

$$AESP = \lim_{n \to \infty} AESP_n(\Pi)$$

Let $p_i(t_{i_0})$ be the i-th vertex of $AESP$, for $i = 0, 1, \ldots,$ or $m + 1$. Let

$$d_{ESP_i} = d_{ESP}(\Pi, p_{i-1}, p_i) + d_{ESP}(\Pi, p_i, p_{i+1})$$

where $i = 1, 2, \ldots,$ or m. Let

$$d_{ESP}(t_0, t_1, \ldots, t_m, t_{m+1}) = \sum_{i=1}^{m} d_{ESP_i}$$

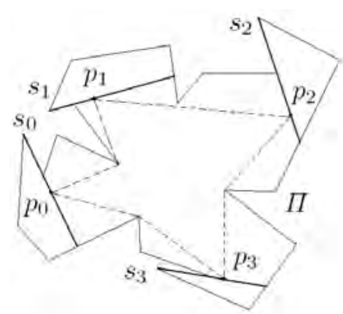

Fig. 6. Illustration for Definition 3.

Definition 3. *Let s_0, s_1, s_2, ... s_m and s_{m+1} be a sequence of segments fully contained in Π and located around the frontier of a simple polygon Π, with points $p_i \in \partial s_i$* (see Figure 6),[1] *for i = 0, 1, 2, ..., m or $m + 1$. We call the $m + 2$ tuple (p_0, p_1, p_2, ..., p_m, p_{m+1}) a* critical point tuple *of Π. We call it an* AESP critical point tuple *of Π if it is the set of all vertices of an $AESP$ of Π.*

Definition 4. *Let*

$$\frac{\partial d_{ESP}(t_0, t_1, \ldots, t_m, t_{m+1})}{\partial t_i}|_{t_i = t_{i_0}} = 0$$

for i = 0, 1, ..., or m+1. Then we say that $(t_{0_0}, t_{1_0}, \ldots, t_{m_0}, t_{m+1_0})$ is a critical point *of $d_{ESP}(t_0, t_1, \ldots, t_m, t_{m+1})$.*

[1] Possibly, the straight segment $p_i p_{i+1}$ is not fully contained in Π. In this case, replace $p_i p_{i+1}$ by the ESP(Π, p_i, p_{i+1}) between p_i and p_{i+1}, where $i = 1, 2, \ldots,$ or m.

Definition 5. *Let $P = (p_0, p_1, p_2, \ldots, p_m, p_{m+1})$ be a critical point tuple of Π. Using P as an initial point set and n iterations of the w-rubberband algorithm, we calculate an* n-w-rubberband transform *of P, denoted by $P\overrightarrow{(w-r-b)_n}Q$, or $P \to Q$ for short, where Q is the resulting critical point tuple of Π, and n is a positive integer.*

4.1 A Correctness Proof for Algorithm 2

We provide mathematic fundamentals for our proof that Algorithm 2 is correct for any input (see Theorem 5 below). At first we recall some basic definitions and theorems from topology, using the book [25] as a reference for basic notions such as topology, topologic space, compact, cover, open cover and so forth. The following is the well-known Heine-Borel Theorem for E_n (see [25], Corollary 2.2 on page 128):

Theorem 3. *A subset S of $\mathbb{R}^n$ is compact iff S is closed and bounded.*

Now let X be a set, and let d be a metric on $X \times X$ defining a metric space (X, d), for example, such as the Euclidean space $(\mathbb{R}^n, d)$. A map f of a metric space (X, d) into a metric space (Y, e) is *uniformly continuous* iff, for each $\varepsilon > 0$, there is a $\delta > 0$ such that $e(f(x), f(y)) < \varepsilon$, for any x, $y \in X$ with $d(x, y) < \delta$. Another well-known theorem (see [25], Theorem 3.2, page 84) is the following:

Theorem 4. *Let f be a map of a metric space (X, d) into a metric space (Y, e). If f is continuos and X is compact, then f is uniformly continuous.*

Now let s_0, s_1, s_2, …, s_m and s_{m+1} be a sequence of segments fully contained in a simple polygon Π, and located around the frontier of Π (see Figure 6). We express a point $p_i(t_i) = (x_i + k_{x_i} t_i, y_i + k_{y_i} t_i, z_i + k_{z_i} t_i)$ on s_i this way in general form, with $t_i \in \mathbb{R}$, for $i = 0$, 1, …, or $m+1$. In the following, $p_i(t_i)$ will be denoted by p_i for short, where $i = 0$, 1, …, or $m+1$.

Lemma 4. *$(t_{0_0}, t_{1_0}, \ldots, t_{m_0}, t_{{m+1}_0})$ is a critical point of $d_{ESP}(t_0, t_1, \ldots, t_m, t_{m+1})$.*

Proof. $d_{ESP}(t_0, t_1, \ldots, t_m, t_{m+1})$ is differentiable at each point

$$(t_0, t_1, \ldots, t_m, t_{m+1}) \in [0, 1]^{m+2}$$

Because $AESP_n(\Pi)$ is an nth polygon of Π, where $n = 1$, 2, …, and

$$AESP = \lim_{n \to \infty} AESP_n(\Pi)$$

it follows that

$$d_{ESP}(t_{0_0}, t_{1_0}, \ldots, t_{m_0}, t_{{m+1}_0})$$

is a local minimum of

$$d_{ESP}(t_0, t_1, \ldots, t_m, t_{m+1})$$

By Theorem 9 of [20], $\frac{\partial d_{ESP}}{\partial t_i} = 0$, for $i = 0$, 1, 2, …, $m+1$. □

Let s_0, s_1 and s_2 be three segments. Let $p_i(p_{i_1}, p_{i_2}, p_{i_3}) \in s_i$, for $i = 0, 1, 2$.

Lemma 5. *There exists a unique point $p_1 \in s_1$ such that*

$$d_e(p_1, p_0) + d_e(p_1, p_2) = \min\{d_e(p', p_0) + d_e(p', p_2) : p' \in \partial s_2\}$$

Proof. We transform the segment from the original 2D coordinate system into a new 2D coordinate system such that s_1 is parallel to one axis, say, the x-axis. Then follow the proof of Lemma 16 of [20]. □

Lemma 5 and Algorithm 2 define a function f_w, which maps $[0,1]^{m+2}$ into $[0,1]^{m+2}$. A point p_i is represented as follows:

$$(a_{i_1} + (b_{i_1} - a_{i_1})t_i, a_{i_2} + (b_{i_2} - a_{i_2})t_i, a_{i_3} + (b_{i_3} - a_{i_3})t_i)$$

Then, following the proof of Lemma 5, we obtain

Lemma 6. *Function $t_2 = t_2(t_1, t_3)$ is continuous at each $(t_1, t_3) \in [0,1]^2$.*

Lemma 7. *If $P\overrightarrow{(w-r-b)_1}Q$, then, for every sufficiently small real $\varepsilon > 0$, there is a sufficiently small real $\delta > 0$ such that $P' \in U_\delta(P)$, and $P'\overrightarrow{(r-b)_1}Q'$ implies $Q' \in U_\varepsilon(Q)$.*

Proof. This lemma follows from Lemma 5; also note that Π has $m+2$ segments, that means we apply Lemma 6 repeatedly, $m+2$ times. □

By Lemma 7, we have the following

Lemma 8. *If $P\overrightarrow{(w-r-b)_n}Q$, then, for every sufficiently small real $\varepsilon > 0$, there is a sufficiently small real $\delta_\varepsilon > 0$ and a sufficiently large integer N_ε such that $P' \in U_{\delta_\varepsilon}(P)$, and $P'\overrightarrow{(w-r-b)_{n'}}Q'$ implies that $Q' \in U_\varepsilon(Q)$, where n' is an integer and $n' > N_\varepsilon$.*

Lemma 9. *Function f_w is uniformly continuous in $[0,1]^{m+2}$.*

Proof. By Lemma 8, f_w is continuous in $[0,1]^{m+2}$. Since $[0,1]^{m+2}$ is a compact and bounded set, by Theorem 4, f_w is uniformly continuous in $[0,1]^{m+2}$. □

By Lemma 9, we are now able to construct an open cover for $[0,1]^{m+2}$ as follows:

1. For each $t_i \in$ [0,1]:
1.1. Let

$$P = (p_0(t_0), p_1(t_1), p_2(t_2), \ldots, p_m(t_m), p_{m+1}(t_{m+1}))$$

be a critical point tuple such that $p(t_i) \in s_i$, for $i = 0, 1, 2, \ldots, m+1$.

1.2. By Lemma 9, there exists a $\delta > 0$ such that, for each $Q \in U_\delta(P) \cap [0,1]^{m+2}$, it is true that $f_w(P) = Q$.
1.3. Let $U'_\delta(P) = U_\delta(P) \cap [0,1]^{m+2}$.
1.4. Let $S_\delta = \{U'_\delta(P) : t_i \in [0,1]\}$.

By Theorem 3, there exists a subset of S_δ, denoted by S'_δ, such that, for each $t'_i \in$ [0,1], and a critical point tuple

$$P' = (p'_0(t'_0), p'_1(t'_1), p'_2(t'_2), \ldots, p'_m(t'_m), p'_{m+1}(t'_{m+1}))$$

there exists $U'_\delta(P) \in S'_\delta$ such that $P' \in U'_\delta(P)$. This proves the following:

Lemma 10. *The number of critical points of*

$$d_{ESP}(t_0, t_1, \ldots, t_m, t_{m+1})$$

in $[0,1]^{m+2}$ *is finite.*

Furthermore, in analogy to the proof of Lemma 24 of [20], we also have the following:

Lemma 11. *Π has a unique AESP critical point tuple.*

Analogously to the proof of Theorem 11 of [20], we obtain

Theorem 5. *The AESP of Π is the shortest watchman route of Π.*

Corollary 1. *For each $\varepsilon > 0$, the w-rubberband algorithm computes an approximative solution ρ' to a WRP such that $|l(\rho') - l(\rho)| < \varepsilon$, where ρ is the true solution to the same WRP.*

Corollary 2. *The WRP has a unique solution.*

Obviously, Corollary 2 is a stronger result than Theorem 1.

4.2 Another Correctness Proof for Algorithm 2

Let $P_0, P_1, \ldots, P_{k-1}$ be a sequence of simple polygons (not necessary convex).

Theorem 6. *For the constrained TPP, the number of local minima is finite.*

Proof. Let F_i be a fence of P_i and P_{i+1} (mod k), for i = 0, 1, ..., k - 1. For each segment s, let $\partial s = \{p : p \in s\}$ (i.e., $\partial s = s$). For each polygon P, let $\partial P = \{p : p \in e \wedge e \in E(P)\}$, where $E(P)$ is the set of all edges of P. Let (as default)

$$\prod_{i=0}^{k-1} \partial P_i = \partial P_0 \times \partial P_1 \times \cdots \partial P_{k-1}$$

For each $p_i \in \partial P_i$, there exists a unique constrained ESP

$$\rho(p_i q_{i_1} q_{i_2} \cdots q_{i_{m_i}} p_{i+1}) \subset F_i^\bullet$$

such that $q_{i_j} \in V(F_i)$, where j = 0, 1, ..., m_i, $m_i \geq 0$ and i = 0, 1, ..., k - 1. For a set S, let $\wp(S)$ be the power set of S. – We define a map f from

$$\prod_{i=0}^{k-1} \partial P_i \qquad \text{to} \qquad \prod_{i=0}^{k-1} \wp(V(F_i))$$

such that

$$f : (p_0, p_1, \ldots, p_{k-1}) \mapsto \prod_{i=0}^{k-1} \{q_{i_1}, q_{i_2}, \ldots, q_{i_{m_i}}\}$$

In general, for a map $g : A \mapsto B$ let

$$Im(g) = \{b : b \in B \wedge \exists\, a\, [a \in A \wedge b = g(a)]\}$$

For each subset $B_1 \subseteq B$, let

$$g^{-1}(B_1) = \{a : a \in A \wedge \exists\, b\, [b \in B \wedge b = g(a)]\}$$

Since $V(F_i)$ is a finite set, $Im(f)$ is a finite set as well. Let

$$Im(f) = \{S_1, S_2, \ldots, S_m\} \;\; (m \geq 1)$$

Then we have that

$$\cup_{i=1}^{m} f^{-1}(S_i) = \prod_{i=0}^{k-1} \partial P_i$$

In other words,

$$\prod_{i=0}^{k-1} \partial P_i$$

can be partitioned into m pairwise disjoint subsets

$$f^{-1}(S_1), f^{-1}(S_2), \ldots, f^{-1}(S_m)$$

For each constrained ESP

$$\rho(p_0 q_{0_1} q_{0_2} \cdots q_{0_{m_0}} p_1 \cdots p_i q_{i_1} q_{i_2} \cdots q_{i_{m_i}} p_{i+1} \cdots p_{k-1} q_{k-1_1} q_{k-1_2} \cdots q_{k-1_{m_{k-1}}} p_0)$$

there exists an index $i \in \{1, 2, \ldots, m\}$ such that

$$(p_0, p_1, \ldots, p_{k-1}) \in f^{-1}(S_i)$$

and

$$\prod_{i=0}^{k-1} \{q_{i_1}, q_{i_2}, \ldots, q_{i_{m_i}}\} \subseteq S_i$$

On the other hand, analogously to the proof of Lemma 24 of [20], for each $f^{-1}(S_i)$, there exists a unique constrained ESP. – This proves the theorem. □

By Theorems 6 and 1, we have a second and shorter proof for Corollary 2 which implies that Algorithm 2 computes an approximative and global solution to a WRP.

5 Time Complexity

5.1 Constrained TPP

Lemma 12. *Procedure 1 can be computed in* $\mathcal{O}(|E(P_1)| + |E(P_2)|)$ *time.*

Proof. Steps 1 and 5 only need constant time. Step 2 can be computed in time $\mathcal{O}(|E(P_1)| + |E(P_2)|)$, Step 3 in time $\mathcal{O}(|E(P_2)|)$, and Step 4 in time $\mathcal{O}(1)$. Therefore, Procedure 1 can be computed in $\mathcal{O}(|E(P_1)| + |E(P_2)|)$ time. □

Lemma 13. *Procedure 2 can be computed in* $\mathcal{O}(|V(\rho(q_1, F, q_2))|)$ *time.*

Proof. Step 1 can be computed in $\mathcal{O}(|V(\rho(q_1, F, q_2))|)$ time. According to [22], Step 2 can be computed in $\mathcal{O}(|V(\rho(q_1, F, q_2))|)$ time. By Lemma 3, Step 3 can be computed in $\mathcal{O}(log\ |V(\rho(q_1, F, q_2))|)$ time. Steps 4 and 5 can be computed in $\mathcal{O}(|V(\rho(t_1, F, t_2))|)$ time. Altogether, the time complexity of Procedure 2 is equal to $\mathcal{O}(|V(\rho(q_1, F, q_2))|)$. □

Lemma 14. *Procedure 3 can be computed in time*

$$\mathcal{O}(|E(P_1)| + 2|E(P_2)| + |E(P_3)| + |E(F_{12})| + |E(F_{23})|)$$

Proof. Step 1.1 requires only constant time. By Lemma 12, Steps 1.2 and 9.1 can be computed in time $\mathcal{O}(|E(P_1)| + 2|E(P_2)| + |E(P_3)|)$. Steps 1.3 and 9.2 can be computed in $\mathcal{O}(|E(P_2)|)$, and Steps 2 and 10 in $\mathcal{O}(1))$ time. Steps 3–4 can be computed in time $\mathcal{O}(|E(F_{12})|)$. By lemma 13, Step 5 can be computed in time $\mathcal{O}(|V(\rho(q_1, F_{12}, q_2))|) = \mathcal{O}(|V_{12}|)$. Since $|V_{12}| \leq |E(F_{12})|$, Steps 3–5 can be computed in $\mathcal{O}(|E(F_{12})|)$ time. Step 6 can be computed in $\mathcal{O}(|E(F_{23})|)$, and Steps 7, 8 and 11 in $\mathcal{O}(|V_{12}|) + \mathcal{O}(|V_{23}|) \leq \mathcal{O}(|E(F_{12})|) + \mathcal{O}(|E(F_{23})|)$ time.

Therefore, Procedure 3 can be computed in

$$\mathcal{O}(|E(P_1)| + 2|E(P_2)| + |E(P_3)| + |E(F_{12})| + |E(F_{23})|)$$

time. This proves the lemma. □

Lemma 15. *Algorithm 1 can be computed in time* $\kappa(\varepsilon) \cdot \mathcal{O}(n)$*, where* n *is the total number of all vertices of the polygons involved.*

Proof. Steps 1–3 can be computed in $\mathcal{O}(k)$ time. By Lemma 14, each iteration in Step 4 can be computed in time

$$\mathcal{O}(\sum_{i=0}^{k-1}(|E(P_{i-1})| + 2|E(P_i)| + |E(P_{i+1})| + |E(F_{i-1})| + |E(F_i)|))$$

Steps 5 and 8 can be computed in $\mathcal{O}(k)$, and Steps 6 and 7 in $\mathcal{O}(|V|)$ time. Note that

$$|V| \leq \sum_{i=0}^{k-1}(|V(P_i)| + |V(F_i)|)$$

$|V(P_i)| = |E(P_i)|$, and $|V(F_i)| = |E(F_i)|$, where $i = 0, 1, \ldots, k-1$. Thus, each iteration (Steps 4–8) in Algorithm 1 can be computed in time

$$\mathcal{O}(\sum_{i=0}^{k-1}(|E(P_{i-1})| + 2|E(P_i)| + |E(P_{i+1})| + |E(F_{i-1})| + |E(F_i)|))$$

Therefore, Algorithm 1 can be computed in

$$\kappa(\varepsilon) \cdot \mathcal{O}(\sum_{i=0}^{k-1}(|E(P_{i-1})| + 2|E(P_i)| + |E(P_{i+1})| + |E(F_{i-1})| + |E(F_i)|))$$

time, where each index is taken mod k. This time complexity is equivalent to $\kappa(\varepsilon) \cdot \mathcal{O}(n)$ where n is the total number of vertices of all polygons. □

Lemma 15 allows to conclude the following

Theorem 7. *The simplified constrained TPP can be solved locally and approximately in $\kappa(\varepsilon) \cdot \mathcal{O}(n)$ time, where n is the total number of all vertices of involved polygons.*

At the end of this subsection, we finally discuss the case when the constrained TPP is not simplified. In this case, we replaced "Procedure 2" in Step 5 of Procedure 3 by the algorithm in [24] on pages 639–641. Then, by Lemma 2 and the proof of Lemma 14, Steps 3–5 can *still* be computed in $\mathcal{O}(|E(F_{12})|)$ time, and Step 6 can *still* be computed in $\mathcal{O}(|E(F_{23})|)$ time. All the other steps are analyzed in exactly the same way as those in the proof of Lemma 14. Therefore, the modified Procedure 3 has the same time complexity as the original procedure. Thus, we have the following

Theorem 8. *The constrained TPP can be solved locally and approximately in $\kappa(\varepsilon) \cdot \mathcal{O}(n)$ time, where n is the total number of vertices of involved polygons.*

According to the following theorem, Theorem 8 is the best possible result in some sense:

Theorem 9. ([13], Theorem 6) *The touring polygons problem (TPP) is NP-hard, for any Minkowski metric L_p ($p \geq 1$) in the case of nonconvex polygons P_i, even in the unconstrained ($F_i = \mathbb{R}^2$) case with obstacles bounded by edges having angles 0, 45, or 90 degrees with respect to the x-axis.*

5.2 Watchman Route Problem

We consider the time complexity of Algorithm 2 for solving the simplified watchman route problem.

Lemma 16. *Consider Procedure 3. If P_i is an essential cut, for i = 1, 2, 3, and $F_{12} = F_{23} = \Pi$, then this procedure can be computed in time $\mathcal{O}(|V(\rho(v_1, \Pi, v_3))|)$, where v_i is the defining vertex of the essential cut P_i, for i = 1, 3.*

Proof. Step 1.1 requires constant time only. By Lemma 12, Steps 1.2 and 9.1 can be computed in $\mathcal{O}(|E(P_1)| + 2|E(P_2)| + |E(P_3)|) = \mathcal{O}(1)$ time. Steps 1.3 and 9.2 can be computed in $\mathcal{O}(|E(P_2)|) = \mathcal{O}(1)$ time. Steps 2 and 10 can be computed in constant time. Steps 3–4 can be computed in time $\mathcal{O}(|E(\rho(v_1, \Pi, v_2))|)$, where v_i is the defining vertex of the essential cut P_i, for $i = 1, 2$.

By Lemma 13, Step 5 can be computed in $\mathcal{O}(|V(\rho(v_1, \Pi, v_2))|)$ time. Thus, Steps 3–5 can be computed in $\mathcal{O}(|V(\rho(v_1, \Pi, v_2))|)$ time. Analogously, Step 6 can be computed in time $\mathcal{O}(|V(\rho(v_2, \Pi, v_3))|)$, where v_i is the defining vertex of the essential cut P_i, for $i = 2, 3$.

Steps 7, 8 and 11 can be computed in $\mathcal{O}(|V(\rho(v_1, \Pi, v_2))|) + \mathcal{O}(|V(\rho(v_2, \Pi, v_3))|) = \mathcal{O}(|V(\rho(v_1, \Pi, v_3))|)$ time. Therefore, the time complexity of Procedure 3 is equal to $\mathcal{O}(|V(\rho(v_1, \Pi, v_3))|)$. □

Lemma 17. *Consider Algorithm 1. If P_i is an essential cut, and $Q_i = \Pi$, for $i = 0, 1, \ldots, k-1$, then this algorithm can be computed in time $\kappa(\varepsilon) \cdot \mathcal{O}(n+k)$, where n is the number of vertices of Π, and k the number of essential cuts.*

Proof. Steps 1–3 can be computed in $\mathcal{O}(k)$ time. By Lemma 16, each iteration in Step 4 can be computed in time

$$\mathcal{O}(\sum_{i=0}^{k-1}(|V(\rho(v_{i-1}, \Pi, v_{i+1}))|))$$

where v_i is the defining vertex of the essential cut P_i, for $i = 0, 1, \ldots, k-1$. This time complexity is actually equivalent to $\mathcal{O}(|V(\Pi)|)$. Steps 5 and 8 can be computed in $\mathcal{O}(k)$, and Steps 6 and 7 in $\mathcal{O}(|V|)$ time. Note that $|V| \leq k + |V(\Pi)|$, and $|V(P_i)| = 2$, where $i = 0, 1, \ldots, k-1$, and $\sum_{i=0}^{k-1}(|V(\rho(v_{i-1}, \Pi, v_{i+1}))|) = 2|E(\Pi)| = 2|V(\Pi)|$. Thus, each iteration (Steps 4–8) in Algorithm 1 can be computed in time $\mathcal{O}(n+k)$. Therefore, Algorithm 1 can be computed in time $\kappa(\varepsilon) \cdot \mathcal{O}(n+k)$, where n is the total number of vertices of Π, and k the number of essential cuts. □

Lemma 17 allows to conclude the following

Theorem 10. *The simplified WRP can be solved approximately in $\kappa(\varepsilon) \cdot \mathcal{O}(n+k)$ time, where n is the number of vertices of polygon Π, and k the number of essential cuts.*

If the WRP is not simplified, then we have the following

Lemma 18. *Consider Algorithm 1. If P_i is an essential cut, and $F_i = \Pi$, for $i = 0, 1, \ldots, k-1$, then this algorithm can be computed in $\kappa(\varepsilon) \cdot \mathcal{O}(kn)$ time, where n is the number of vertices of Π, and k the number of essential cuts.*

Proof. Steps 1–3 can be computed in $\mathcal{O}(k)$ time. By Lemma 16, each iteration in Step 4 can be computed in

$$\mathcal{O}(\sum_{i=0}^{k-1}(|E(P_{i-1})| + 2|E(P_i)| + |E(P_{i+1})| + |E(F_{i-1})| + |E(F_i)|))$$

time, what is equivalent to the asymptotic class $\mathcal{O}(k|E(\Pi)|)$, because of $|E(P_{i-1})| = |E(P_i)| = |E(P_{i+1})| = 1$ and $|E(F_{i-1})| = |E(F_i)| = |E(\Pi)| = |V(\Pi)|$. Steps 5 and 8 can be computed in $\mathcal{O}(k)$, and Steps 6 and 7 in $\mathcal{O}(|V|) \leq |V(\Pi)|$ time.

Thus, each iteration (Steps 4–8)) in Algorithm 1 can be computed in $\mathcal{O}(kn)$ time. Therefore, Algorithm 1 can be computed in $\kappa(\varepsilon) \cdot \mathcal{O}(kn)$ time, where n is the total number of vertices of Π, and k the number of essential cuts. □

6 Conclusions

Lemma 18 and Theorem 8 allow to conclude the following

Theorem 11. *The general WRP can be solved approximately in $\kappa(\varepsilon) \cdot \mathcal{O}(kn)$ time, where n is the number of vertices of polygon Π, and k the number of essential cuts.*

The subprocesses of Algorithms 1 and 2 (as typical for rubberband algorithms in general) are also numerically stable. The proposed algorithms can be recommended for optimization of visual inspection programs.

References

1. M. H. Alsuwaiyel and D. T. Lee. Minimal link visibility paths inside a simple polygon. *Comput. Geom.*, **3**: 1–25, 1993.
2. M. H. Alsuwaiyel and D. T. Lee. Finding an approximate minimum-link visibility path inside a simple polygon. *Information Processing Letters*, **55**:75–79, 1995.
3. E. M. Arkin, J. S. B. Mitchell, and C. Piatko. Minimum-link watchman tours. Report, University at Stony Brook, 1994.
4. T. Asano, S. K. Ghosh, and T. C. Shermer. Visibility in the plane. In *Handbook of Computational Geometry* (J.-R. Sack and J. Urrutia, editors), pages 829–876, Elsevier, 2000.
5. S. Carlsson, H. Jonsson, and B. J. Nilsson. Optimum guard covers and m-watchmen routes for restricted polygons. Proc. *Workshop Algorithms Data Struct.*, LNCS 519, pages 367–378, Springer, 1991.
6. S. Carlsson, H. Jonsson, and B. J. Nilsson. Optimum guard covers and m-watchmen routes for restricted polygons. *Int. J. Comput. Geom. Appl.*, **3**:85–105, 1993.
7. S. Carlsson and H. Jonsson. Computing a shortest watchman path in a simple polygon in polynomial-time. Proc. *Workshop Algorithms Data Struct.*, LNCS 955, pages 122–134, Springer, 1995.
8. S. Carlsson, H. Jonsson, and B. J. Nilsson. Approximating the shortest watchman route in a simple polygon. Technical report, Lund University, Sweden, 1997.
9. S. Carlsson, H. Jonsson, and B. J. Nilsson. Finding the shortest watchman route in a simple polygon. *Discrete Computational Geom.*, **22**:377–402, 1999.
10. W. Chin and S. Ntafos. Optimum watchman routes. *Information Processing Letters*, **28**:39–44, 1988.
11. W.-P. Chin and S. Ntafos. Shortest watchman routes in simple polygons. *Discrete Computational Geometry*, **6**:9–31, 1991.

12. J. Czyzowicz, P. Egyed, H. Everett, W. Lenhart, K. Lyons, D. Rappaport, T. Shermer, D. Souvaine, G. Toussaint, J. Urrutia, and S. Whitesides. The aquarium keeper's problem. In Proc. *ACM-SIAM Sympos. Data Structures Algorithms*, pages 459–464, 1991.
13. M. Dror, A. Efrat, A. Lubiw, and J. Mitchell. Touring a sequence of polygons. In Proc. *STOC*, pages 473–482, 2003.
14. L. P. Gewali, A. Meng, J. S. B. Mitchell, and S. Ntafos. Path planning in 0/1/infinity weighted regions with applications. *ORSA J. Comput.*, **2**:253–272, 1990.
15. L. P. Gewali and R. Lombardo. *Watchman Routes for a Pair of Convex Polygons.* Lecture Notes in Pure Appl. Math., volume 144. 1993.
16. L. P. Gewali and S. Ntafos. Watchman routes in the presence of a pair of convex polygons. In Proc. *Canad. Conf. Comput. Geom.*, pages 127–132, 1995.
17. M. Hammar and B. J. Nilsson. Concerning the time bounds of existing shortest watchman routes. In Proc. *FCT'97*, LNCS 1279, pages 210–221, 1997.
18. P. Kumar and C. Veni Madhavan. Shortest watchman tours in weak visibility polygons. In Proc. *Canad. Conf. Comput. Geom.*, pages 91–96, 1995.
19. F. Li and R. Klette. Exact and approximate algorithms for the calculation of shortest paths. IMA Minneapolis, Report 2141 on *www.ima.umn.edu/preprints/oct2006*.
20. C. Mata and J. S. B. Mitchell. Approximation algorithms for geometric tour and network design problems. In Proc. *Annu. ACM Sympos. Comput. Geom.*, pages 360–369, 1995.
21. A. Melkman. On-line construction of the convex hull of a simple polygon. *Information Processing Letters*, **25**:11–12, 1987.
22. J. S. B. Mitchell and E. L. Wynters. Watchman routes for multiple guards. In Proc. *Canad. Conf. Comput. Geom.*, pages 126–129, 1991.
23. J. S. B. Mitchell. Geometric shortest paths and network optimization. In *Handbook of Computational Geometry* (J.-R. Sack and J. Urrutia, editors). pages 633–701, Elsevier, 2000.
24. T. O. Moore. *Elementary General Topology.* Prentice-Hall, Englewood Cliffs, N.J., 1964.
25. B. J. Nilsson and D. Wood. Optimum watchmen routes in spiral polygons. In Proc. *Canad. Conf. Comput. Geom.*, pages 269–272, 1990.
26. B. J. Nilsson. Guarding art galleries; Methods for mobile guards. Ph.D. Thesis, Lund University, Sweden, 1995.
27. S. Ntafos. The robber route problem. *Inform. Process. Lett.*, **34**:59–63, 1990.
28. S. Ntafos. Watchman routes under limited visibility. *Comput. Geom.*, **1**:149–170, 1992.
29. S. Ntafos and L. Gewali. External watchman routes. *Visual Comput.*, **10**:474–483, 1994.
30. D. Sunday. Algorithm 14: Tangents to and between polygons. See $http://softsurfer.com/Archive/algorithm_0 201/$ (last visit: June 2006).
31. X. Tan, T. Hirata, and Y. Inagaki. An incremental algorithm for constructing shortest watchman route algorithms. *Int. J. Comp. Geom. and Appl.*, **3**:351–365, 1993.
32. X. Tan and T. Hirata. Constructing shortest watchman routes by divide-and-conquer. In Proc. *ISAAC*, LNCS 762, pages 68–77, 1993.
33. X. Tan, T. Hirata, and Y. Inagaki. Corrigendum to 'An incremental algorithm for constructing shortest watchman routes'. *Int. J. Comp. Geom. App.*, **9**:319–323, 1999.
34. X. Tan. Fast computation of shortest watchman routes in simple polygons. *Information Processing Letters*, **77**:27–33, 2001.

35. X. Tan. Approximation algorithms for the watchman route and zookeeper's problems. In Proc. *Computing and Combinatorics*, LNCS 2108, pages 201–206, Springer, Berlin, 2005.
36. X. Tan. Approximation algorithms for the watchman route and zookeeper's problems. *Discrete Applied Mathematics*, **136**:363–376, 2004.
37. X. Tan. Linear-time 2-approximation algorithm for the watchman route problem. In Proc. *Theory Applications Models Computation*, LNCS 3959, pages 181–191, Springer, Berlin, 2006.
38. J. Urrutia. Geometric shortest paths and network optimization. In *Handbook of Computational Geometry* (J.-R. Sack and J. Urrutia, editors). pages 973–1027, Elsevier, 2000.