Watchman Route in a Simple Polygon with a Rubberband Algorithm

Fajie Li* Reinhard Klette†

Abstract

So far, the best result in running time for solving the fixed watchman route problem (i.e., shortest path for viewing any point in a simple polygon with given start point) is $\mathcal{O}(n^3 \log n)$, published in 2003 by M. Dror, A. Efrat, A. Lubiw, and J. Mitchell. – This paper provides an algorithm with $\kappa(\varepsilon) \cdot \mathcal{O}(kn)$ runtime, where n is the number of vertices of the given simple polygon Π, and k the number of essential cuts; $\kappa(\varepsilon)$ defines the numerical accuracy in dependency of a selected constant $\varepsilon > 0$. Moreover, our algorithm is significantly simpler, easier to understand and implement than previous ones for solving the fixed watchman route problem.

1 Introduction

Let Π be a planar, simple, topologically closed polygon with n vertices, and $\partial\Pi$ be its frontier. A point $p \in \Pi$ is *visible* from point $q \in \Pi$ iff $pq \subset \Pi$. The *(floating) watchman route problem* (WRP) of computational geometry, as discussed in [2], is defined as follows: *Calculate a shortest route $\rho \subset \Pi$ such that any point $p \in \Pi$ is visible from at least one point on ρ.* If a start point of the route is given on $\partial\Pi$ then this refined problem is known as the *fixed* WRP. In the rest of this paper, let $s \in \partial\Pi$ be the starting point of the fixed WRP.

A simplified WRP of finding a shortest route in a simple isothetic polygon was solved in 1988 in [7] by presenting an $\mathcal{O}(n \log \log n)$ algorithm. In 1991, [8] claimed to have presented an $\mathcal{O}(n^4)$ algorithm, solving the fixed WRP. In 1993, [21] obtained an $\mathcal{O}(n^3)$ solution for the fixed WRP. In the same year, this was further improved to a quadratic time algorithm [22]. However, four years later, in 1997, [10] pointed out that the algorithms in both [8] and [21] were flawed, but presented a solution for fixing those errors. Interestingly, two years later, in 1999, [23] found that the solution given by [10] was also flawed! By modifying the (flawed) algorithm presented in [21], [23] gave an $\mathcal{O}(n^4)$ runtime algorithm for the fixed WRP. In 1995 and 1999, [17] and [6] gave an $\mathcal{O}(n^6)$ algorithm for the WRP respectively. This was improved in 2001 by an $\mathcal{O}(n^5)$ algorithm in [24]. So far the best known result for the fixed WRP is due to [9] by presenting in 2003 an $\mathcal{O}(n^3 \log n)$ runtime algorithm.

Given the time complexity of those algorithms for solving the WRP, finding efficient approximation algorithms became an interesting subject. Recall the following definition; see, for example, [11]: An algorithm is an *δ-approximation algorithm* for a minimization problem P iff, for each input of P, the algorithm delivers a solution that is at most δ times the optimum solution. In case of the WRP, the optimum solution is defined by the length of the shortest path.

In 1995, [14] published an $\mathcal{O}(\log n)$-approximation algorithm for solving the WRP. In 1997, [5] gave a 99·98-approximation algorithm with time complexity $\mathcal{O}(n \log n)$ for the WRP. In 2001, [25] 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 [26] in 2004. Most recently, [27] 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.

Let ESP denote the class of any Euclidean shortest path problem. Corresponding to the definition of δ-approximation algorithms, we introduce the following definition: A Euclidean path is a *δ-approximation (Euclidean) path* for an ESP problem iff its length is at most δ times the optimum solution.

The paper is organized as follows: Section 2 defines some notations for later usage. Section 3 proposes and discusses the main algorithm of this paper. Section 4 concludes.[1]

2 Preliminaries

We recall some definitions from [9] and [27]. A vertex v of Π is called *reflex* if v's internal angle is greater than 180°. Let u be a vertex of Π which is adjacent to a reflex vertex v. Assume that the straight line uv intersects an edge of Π at v'. Then the segment $C = vv'$ partitions Π into two parts. C is called a *cut* of Π if C makes a convex vertex at v in the part containing

*College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian, China, li.fajie@yahoo.com

†Computer Science Department, The University of Auckland, Private Bag 92019, Auckland 1142, New Zealand, r.klette@auckland.ac.nz

[1] An expanded version is MI-tech report no. 51 at www.mi.auckland.ac.nz/.

the starting point s, and v is called a *defining vertex* of C. That part of Π which contains s is called *essential* part of C and is denoted by $\Pi(C)$. The other part of Π is called the *pocket* induced by cut C, and C is the *associated* cut of the pocket. A cut C *dominates* a cut C' iff $\Pi(C)$ contains $\Pi(C')$. A cut is called *essential* if it is not dominated by another cut. (Also known as 'non-redundant chord' in the literature.) A pocket is called *essential* if it does not contain any other pocket. A pocket is essential iff its associated cut is essential.

If two points u and v are on two different edges of Π, such that the segment uv partitions Π into two parts, then we say that uv is a *general cut* of Π. We may arbitrarily select one of both endpoints of the segment uv to be its *start point*. In the rest of this paper, for an essential cut C of Π, we identify the defining vertex of C with its start point. If C_0, C_1, $\ldots$, C_{k-1} are all the essential cuts of Π such that their start points are ordered clockwise around on $\partial\Pi$, then we say that C_0, C_1, $\ldots$, C_{k-1} and Π *satisfy the condition of the fixed watchman route problem*.

Let p, $q \in \Pi$; if $pq \subset \Pi$ then q *can see* p (with respect to Π), and p is a *visible point* of q. Let $q \in \Pi$ and assume a segment $s \subset \Pi$. If, for each $p \in s$, q can see p, then we say that q *can see* s.

Let $q \in \Pi$, segment $s \subset \Pi$, $p \in s$, and p is not an endpoint of s. If q can see p, but for any sufficiently small $\varepsilon > 0$, q cannot see p', where $p' \in s$ and Euclidean distance $d_e(p, p') = \varepsilon$, then we say that p is a *visible extreme point* of q (with respect to s and Π).

Let segment $s \subset \Pi$ and $q \in \Pi \setminus s$. If there exists a subsegment $s' \subseteq s$ such that q can see s', and each endpoint of s' is a visible extreme point of q or an endpoint of s, then we say that s' is a *maximal visible segment* of q (with respect to Π). Let s_0, s_1, $\ldots$, and s_{k-1} be k segments ($k \geq 2$) in three-dimensional Euclidean space (in short: $3D$), $p \in s_0$, and $q \in s_{k-1}$.

Let $L_S(p, q)$ be the length of the shortest path, starting at p, then visiting segments s_1, $\ldots$, and s_{k-2} in order, and finally ending at q, where $S = \langle s_0, s_1, \ldots, s_{k-1}\rangle$. Let p, $q \in \Pi$. We denote by $L_\Pi(p, q)$ the length of the shortest path from p to q inside of Π.

Let ρ be a polygonal path and $V(\rho)$ the set of all vertices of ρ; $|V(\rho)|$ is the number of vertices of ρ. Denote by $C(S)$ the convex hull of a set S. Let S_0, S_1, $\ldots$, and S_{k-1} be k non-empty sets; let $\prod_{i=0}^{k-1} S_i$ be the cross product of those sets.

This ends our introduction of technical terms. We also recall in one place here two results which will be cited later in this paper:

- **Lemma 1** ([9], page 475) *A solution to the fixed watchman route problem (i.e., a shortest tour) visits the essential cuts in the same order as the defining vertices meet $\partial\Pi$.*
- **Theorem 2** ([27], Theorem 1) *Given a simple polygon Π; the set $\mathcal{C}$ of all essential cuts for the watchman route in Π can be computed in $\mathcal{O}(n)$ time.*

3 Algorithms

In this section, we describe and discuss now the promised algorithm for solving the fixed watchman route problem.

3.1 Two Procedures and Main Algorithm

The main algorithm uses two procedures; the second applies a 2D ESP algorithm (see [16], pages 639–641). We present the used procedures first, and the main algorithm later.

As described in Section **??**, the main idea of a *Rubberband Algorithm* (RBA) is as follows: In each iteration, we update (by finding a local minimum or optimal vertex) the second vertex p_i for every three-subsequent-vertices subsequence p_{i-1}, p_i, p_{i+1} in a *step set* $\{S_1, S_2, \ldots, S_k\}$. The first procedure below computes the maximal visible segment, which is actually an element of the step set of the used RBA. The second procedure is used for updating the vertices.

Procedure 1 Compute Maximal Visible Segment
Input: Polygon Π and a general cut C of Π; let v_1 and v_2 be two endpoints of C; two points p and q such that $p \in C$ and p is a visible point of $q \in \partial\Pi \setminus C$.
Output: Two points p'_1, $p'_2 \in C$ such that p is in the segment $p'_1p'_2$, and $p'_1p'_2$ is the maximal visible segment of q.

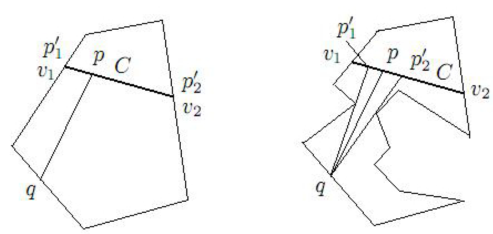

Figure 1: Illustration for Procedure 1.

We describe Procedure 1 informally. – Case 1: p is not an endpoint of C. For $i \in \{1, 2\}$, if q can see v_i, (see left, Figure 1), let p'_i be v_i; otherwise, let V_i be the set of vertices in $V(\partial\Pi)$ such that each vertex in V_i is in $\triangle qpv_i$. Apply the convex hull algorithm (see, e.g., [15] or Figure 13.7, [12]) to compute $C(V_i)$. Apply the tangent algorithm (see [20]) to find a point $p'_i \in C$ such that qp'_i is a tangent to $C(V_i)$ (see right of Figure 1). – Case 2: p is an endpoint of C. Without loss of generality, assume that $p = v_1$. Let p'_1 be p. Let V_2

be the set of vertices in $V(\partial\Pi)$ such that each vertex in V_2 is in $\triangle qpv_i$. Apply the convex hull algorithm to compute $C(V_2)$. Apply the tangent algorithm to find a point $p'_2 \in C$ such that qp'_2 is a tangent to $C(V_2)$.

Procedure 2 Handling of Three General Cuts
Input: Three general cuts C_1, C_2 and C_3 of Π; three points $p_i \in C_i$, for $i = 1, 2, 3$; and a degeneration accuracy constant $\varepsilon_2 > 0$.
Output: An updated shorter path $\rho(p_1, \ldots, p_2, \ldots, p_3)$ that might also contain vertices of the polygon Π.

1: For both $i \in \{1, 2\}$, let $\{p_i, p_{i+1}\}$ (where $p_i \in C_i$) be the input for the 2D ESP algorithm; the output is a set V_{ii+1} - the set of vertices of a shortest path from p_i to p_{i+1} inside of Π. Let V be $V_{12} \cup V_{23}$.
2: Find q_1 and $q_3 \in V$ such that $\langle q_1, p_2, q_3\rangle$ is a subsequence of V (i.e., q_1, p_2, q_3 appear consecutively in V).
3: Let $C = C_2$, $p = p_2$, $q = q_i$, apply Procedure 1 to find the maximal visible segment $s_i = p'_1 p'_2$ of q_i, i = 1, 3.
4: Find vertex $p'_2 \in s_2 = s_1 \cap s_3$ such that $d_e(q_1, p'_2) + d_e(p'_2, q_3) = \min\{d_e(q_1, p') + d_e(p', q_3) : p' \in s_2\}$.
5: If $C_2 \cap C_1$ (or C_3) $\neq \emptyset$ and p'_2 is the intersection point, then ε_2-transform p'_2 into another point (still denoted by p'_2) in C_2.
6: Update V by letting p_2 be p'_2.

Note that in Procedure 2, if C_1 or C_3 degenerates to a single point, then this procedure still works correctly.

Algorithm 1 Main Algorithm
Input: k essential cuts C_0, C_1, …, C_{k-1}, and Π, which satisfy the condition of the fixed WRP, and points $p_i \in C_i$, where $i = 0, 1, 2, \ldots, k-1$; and an accuracy constant $\varepsilon > 0$ and a degeneration accuracy constant $\varepsilon_2 > 0$.
Output: An updated closed $\{1 + 4k[r(\varepsilon) + \varepsilon_2]/L\}$-approximation path $\rho(s, p_0, \ldots, p_1, \ldots, p_{k-1}, s)$, which may also contain vertices of Π, where L is the length of an optimal path, $r(\varepsilon)$ the upper error bound[2] for distances between p_i and the corresponding optimal vertex p'_i: $d_e(p_i, p'_i) \leq r(\varepsilon)$, for $i = 0, 1, \ldots, k-1$.

The following pseudo code is fairly easy to read, and we defer from providing another (more informal) high level description of Algorithm 1.

1: For $i \in \{0, 1, \ldots, k-1\}$, let p_i be the center of C_i.
2: Let V_0 and V be a sequence of points $\langle p_0, p_1, \ldots, p_{k-1}\rangle$; L_1 be $\sum_{i=-1}^{k} L_\Pi(p_i, p_{i+1})$; and L_0 be ∞ ($p_{-1} = p_k = s$).
3: **while** $L_0 - L_1 \geq \varepsilon$ **do**
4:     **for** each $i \in \{0, 1, \ldots, k-1\}$ **do**
5:         Let C_{i-1}, C_i, C_{i+1}, p_{i-1}, p_i, p_{i+1} and Π be the input for Procedure 2, which updates p_i in V_0. ($C_{-1} = C_k = p_{-1} = p_k = s$)
6:         Let U_i be the sequence of vertices of the path $\rho(p_{i-1}, \ldots, p_i, \ldots, p_{i+1})$ with respect to C_{i-1}, C_i and C_{i+1} (inside of Π); let U_i be $\langle q_1, q_2, \ldots, q_m\rangle$.
7:         Insert (after p_{i-1}) the points of sequence U_i (in the given order) into V_0; i.e., we have that $V_1 = \langle p_0, p_1, \ldots, p_{i-1}, q_1, q_2, \ldots, q_m, p_{i+1}, \ldots, p_{k-1}\rangle$. (Note: sequence V_1 is the updated sequence V_0, after inserting U_i)
8:     **end for**
9:     Let L_0 be L_1 and V_0 be V (Note: we use the updated original sequence V instead of V_1 for the next iteration).
10:     Calculate the perimeter L_1 of the polygon, given by the sequence V_1 of vertices.
11: **end while**
12: Output sequence V_1, and the desired length equal to L_1.

3.2 Correctness and Time Complexity

We state without proof:

Theorem 3 *If the chosen accuracy constant $\varepsilon > 0$ is sufficiently small, then Algorithm 1 outputs a unique $\{1 + 4k \cdot [r(\varepsilon) + \varepsilon_2]/L\}$-approximation (closed) path with respect to the step set $\langle S_0, S_1, \ldots, S_{k-1}, S_0\rangle$, for any initial path.*

Theorem 3 says that Algorithm 1 outputs an approximate solution to the fixed WRP; we have the following:

Theorem 4 *Algorithm 1 outputs an $\{1 + 4k \cdot [r(\varepsilon) + \varepsilon_2]/L\}$-approximation solution to the fixed WRP.*

Proof. $\sum_{i=-1}^{k} L_\Pi(p_i, p_{i+1}) : \prod_{i=-1}^{k} C_i \to \mathbb{R}$ is a convex function, where $L_\Pi(p_i, p_{i+1})$ is defined as in Step 2 of Algorithm 1. Basic results in the theory of convex functions and Theorem 3 prove then the theorem. □

Regarding the time complexity of our solution to the fixed WRP, we first state the fact that Procedure 1 and Procedure 2 can be computed in time $\mathcal{O}(|V(\partial\Pi)|)$. Furthermore, note that the main computation is in the two stacked loops. The while-loop takes $\kappa(\varepsilon)$ iterations. By the stated fact, the for-loop can be computed in time $\mathcal{O}(k \cdot |V(\partial\Pi)|)$. Thus, Algorithm 1 can be computed in time

$$\kappa(\varepsilon) \cdot \mathcal{O}(k \cdot |V(\partial\Pi)|)$$

By Lemma 1 and Theorem 2, we may conclude that this paper provided an $\{1+4k \cdot [r(\varepsilon)+\varepsilon_2]/L\}$-approximation solution to the fixed WRP, having time complexity $\kappa(\varepsilon) \cdot \mathcal{O}(k \cdot |V(\partial\Pi)|)$, where k is the number of essential cuts, and L is the length of an optimal watchman route.

[2] It is obvious to see that $\lim_{\varepsilon \to 0} r(\varepsilon) = 0$

4 Concluding Remarks

This paper applies basic ideas of RBAs, which were proposed in digital geometry [4, 12] for the specific 3D ESP of calculating shortest Euclidean "loops" in a sequence of cubes. (We refined those ideas such that we now also have a general "arc" version of an RBA.)

The basic idea of an RBA might be generalized to establish a whole class of rubberband algorithms (RBAs) for solving various Euclidean shortest path problems. The main algorithm of this paper (Algorithm 1) is just an example for such an RBA. As indicated in Note 1, in distinction to already published approximation algorithms, our algorithm offers a high accuracy. In some simple polygons, we find the exact solution to the fixed WRP, in the others we converge to the correct solution. A large number of experimental results also indicate that $\kappa(\varepsilon) = \mathcal{O}(k)$, where k is the number of essential cuts. It remains a challenge to prove a smallest upper bound for $\kappa(\varepsilon)$.

Altogether, our algorithm is not only faster than previously published solutions to the fixed WRP, but also significantly simpler, easier to understand and to implement.

References

[1] E. M. Arkin, J. S. B. Mitchell, and C. Piatko. Minimum-link watchman tours. Report, University at Stony Brook, 1994.

[2] 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.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, UK, 2004.

[4] T. Bülow and R. Klette. Digital curves in 3D space and a linear-time length estimation algorithm. *IEEE Trans. Pattern Analysis Machine Intelligence*, **24**:962–970, 2002.

[5] S. Carlsson, H. Jonsson, and B. J. Nilsson. Approximating the shortest watchman route in a simple polygon. Technical report, Lund University, Sweden, 1997.

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

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

[8] W.-P. Chin and S. Ntafos. Shortest watchman routes in simple polygons. *Discrete Computational Geometry*, **6**:9–31, 1991.

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

[10] 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.

[11] D. S. Hochbaum (editor). *Approximation Algorithms for NP-Hard Problems.* PWS Pub. Co.,Boston, 1997.

[12] R. Klette and A. Rosenfeld. *Digital Geometry.* Morgan Kaufmann, San Francisco, 2004.

[13] H. Luo and A. Eleftheriadis. Rubberband: an improved graph search algorithm for interactive object segmentation. In Proc. *Int. Conf. Image Processing*, volume 1, pages 101–104, 2002.

[14] C. Mata and J. S. B. Mitchell. Approximation algorithms for geometric tour and network design problems. In Proc. *Ann. ACM Symp. Computational Geometry*, pages 360–369, 1995.

[15] A. Melkman. On-line construction of the convex hull of a simple polygon. *Information Processing Letters*, **25**:11–12, 1987.

[16] 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.

[17] B. J. Nilsson. Guarding art galleries; Methods for mobile guards. Ph.D. Thesis, Lund University, Sweden, 1995.

[18] A. W. Roberts and V. D. Varberg. *Convex Functions.* Academic Press, New York, 1973.

[19] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, N.J., 1970.

[20] D. Sunday. Algorithm 14: Tangents to and between polygons. See `http://softsurfer.com/Archive/algorithm_0201/` (last visit: November 2008).

[21] 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.

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

[23] X. Tan, T. Hirata, and Y. Inagaki. Corrigendum to 'An incremental algorithm for constructing shortest watchman routes'. *Int. J. Comp. Geom. and Appl.*, **9**:319–323, 1999.

[24] X. Tan. Fast computation of shortest watchman routes in simple polygons. *Information Processing Letters*, **77**:27–33, 2001.

[25] 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.

[26] X. Tan. Approximation algorithms for the watchman route and zookeeper's problems. *Discrete Applied Mathematics*, **136**:363–376, 2004.

[27] X. Tan. A linear-time 2-approximation algorithm for the watchman route problem for simple polygons. *Theoretical Computer Science*, **384**:92–103, 2007.