An Approximation Algorithm for the Two-Watchman Route in a Simple Polygon

Bengt J. Nilsson* Eli Packer†

Abstract

The *two-watchman route problem* is that of computing a pair of closed tours in an environment so that the two tours together see the whole environment and some length measure on the two tours is minimized. Two standard measures are: the minmax measure, where we want the tours where the longest of them has minimal length, and the minsum measure, where we want the tours for which the sum of their lengths is smallest. It is known that computing the minmax two-watchman route is NP-hard for simple rectilinear polygons and thus also for simple polygons. We exhibit a polynomial time 7.1416-factor approximation algorithm for computing the minmax two-watchman route in simple polygons.

1 Introduction

Some of the most intriguing problems in computational geometry concern visibility and motion planning in polygonal environments. A classical problem is that of computing a *shortest watchman route* in an environment, i.e., the shortest closed tour that sees the complete free-space of the environment. This problem has been shown NP-hard [5] and even $\Omega(\log n)$-inapproximable unless P=NP [7] for polygons with holes having a total of n segments.

Watchman route algorithms either compute a *fixed* watchman route which requires the tour to pass a given boundary point or they compute a *floating* watchman route, with no requirement to pass any specific point. Tan *et al.* [11] prove an $O(n^4)$ time algorithm based on dynamic programming for computing a shortest fixed watchman route through a given boundary point in a simple polygon. This is later improved to $O(n^3 \log n)$ time by Dror *et al.* [4]. Carlsson *et al.* [2] show how to generalize algorithms for the shortest fixed watchman route to compute a shortest floating watchman route in a simple polygon with a quadratic factor overhead. Tan [10] improves this to a linear factor overhead. Hence, the currently best algorithm for a shortest floating watchman route in a simple polygon uses $O(n^4 \log n)$ time.

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

†Proactive Location Intelligence Team, IBM, Haifa, Israel. ELIP@il.ibm.com

The problem of computing multiple watchman routes that together see the environment has received much less attention. Mitchell and Wynters [8] show that already computing the pair of tours that together see a simple rectilinear polygon is NP-hard, if we want to minimize the length of the longest of the two tours, the *minmax* measure. It is still an open problem whether it is possible to compute a pair of tours for which the sum of the lengths of the two tours is minimal, the *minsum* measure, in polynomial time. Packer [9] give some experimental results for multiple watchman routes in simple polygons. In the case when the watchmen are point sized, Belleville [1] shows an efficiently computable characterization of all simple polygons that are two-guardable with point guards.

We give a polynomial time 7.1416-factor approximation algorithm to compute a minmax pair of tours that together see a simple polygon.

2 Preliminaries

Let $\mathbf{P}$ be a simple polygon having n vertices and let $\partial\mathbf{P}$ denote the boundary of $\mathbf{P}$. We say that two points in $\mathbf{P}$ *see* each other, if the line segment connecting the points does not intersect the exterior of $\mathbf{P}$. For any connected object X inside $\mathbf{P}$, we denote by $\mathbf{VP}(X)$ the *weak visibility polygon* of X in $\mathbf{P}$, i.e., the set of points in $\mathbf{P}$ that see some point of X. $\mathbf{VP}(X)$ when X is a point, a segment, or a polygonal curve in $\mathbf{P}$ can be efficiently computed [6].

We define a *cut* to be a directed line segment in $\mathbf{P}$ with both end points on $\partial\mathbf{P}$ and having at least one interior point not on $\partial\mathbf{P}$. Hence, a polygon edge is not a cut. A cut separates $\mathbf{P}$ into two sub-polygons. If a cut is represented by the segment $[p, q]$ we say that the cut is directed from p to q and we call p the *start point* of the cut. For a cut c in $\mathbf{P}$, we let the *left polygon*, $\mathbf{L}(c)$, be the set of points in $\mathbf{P}$ locally to the left of c.

Assume a counterclockwise walk of $\partial\mathbf{P}$. Such a walk imposes a direction on each of the edges of $\mathbf{P}$ in the direction of the walk. Consider a reflex vertex of $\mathbf{P}$. The two edges incident to the vertex can each be extended inside $\mathbf{P}$ until the extensions reach a boundary point. These extended segments form cuts given the same direction as the edge they are collinear to. We call these cuts *extensions*.

A *guard set* is any set of points that together see all of $\mathbf{P}$. Any guard set must have points intersecting $\mathbf{L}(e)$ for every extension e of $\mathbf{P}$, since otherwise the edge collinear to e will not be seen by the guard set. Chin and Ntafos [3] prove that this is indeed also a sufficient requirement when the guard set is connected, as it is for a shortest watchman route.

Let c be a cut. If a guard set $\mathcal{G}$ intersects $\mathbf{L}(c)$, we say that c is *covered* by $\mathcal{G}$. Furthermore, if $\mathcal{G}$ intersects the interior of $\mathbf{L}(c)$, then $\mathcal{G}$ *properly covers* c. If $\mathcal{G}$ properly covers c and intersects c, we say that $\mathcal{G}$ *crosses* c. Finally, if $\mathcal{G}$ covers c, but does not properly cover c, then $\mathcal{G}$ *reflects* on c.

For two cuts, c and c', we say that c *dominates* c', if $\mathbf{L}(c) \subseteq \mathbf{L}(c')$ An extension that is not dominated by any other extension is called *essential*. By the transitivity of the domination relation, if a guard set has points to the left of each essential extension, it also has points to the left of every extension [3].

All exact watchman route algorithms for simple polygons [2, 3, 4, 10, 11] compute closed tours that cover every essential extension. They can also be used with any set of cuts $\mathcal{C}$ to compute the shortest tour that covers each cut in $\mathcal{C}$, in polynomial time. We call such a tour the *shortest visiting tour* of the cuts in $\mathcal{C}$ inside $\mathbf{P}$ and denote it $SVT_{\mathcal{C}}$. For the case that $\mathcal{C}$ consists of the essential extensions of $\mathbf{P}$, the tour is a *shortest watchman route*, W_{S}.

We also make use of the fact that shortest paths in $\mathbf{P}$ between combinations of segments and points can be computed efficiently [6]. We denote the shortest path between two objects X and Y in $\mathbf{P}$ by $SP(X, Y)$.

Let X_1 and X_2 be two closed polygonal cycles contained in a simple polygon $\mathbf{P}$, such that any point in $\mathbf{P}$ sees some point on X_1 or X_2. We call such a pair $\mathcal{X} = (X_1, X_2)$, a *two-watchman route*. The length of a cycle X in $\mathbf{P}$ is denoted $||X||$ and we let $||\mathcal{X}||_{\mathsf{sum}} \stackrel{\text{def}}{=} ||X_1|| + ||X_2||$ be the *sum length* of $\mathcal{X}$ and $||\mathcal{X}||_{\mathsf{max}} \stackrel{\text{def}}{=} \max\{||X_1||, ||X_2||\}$ be the *max length* of $\mathcal{X}$.

Let $\mathcal{S} = (S_1, S_2)$ and $\mathcal{T} = (T_1, T_2)$ be two two-watchman routes such that $||\mathcal{S}||_{\mathsf{sum}} \leq ||\mathcal{X}||_{\mathsf{sum}}$ and $||\mathcal{T}||_{\mathsf{max}} \leq ||\mathcal{X}||_{\mathsf{max}}$ for any two-watchman route $\mathcal{X}$ in $\mathbf{P}$. We say that $\mathcal{S}$ is a *minsum* two-watchman route and $\mathcal{T}$ is a *minmax* two-watchman route. The following inequalities are immediate from the definitions,

$$||\mathcal{T}||_{\mathsf{max}} \leq ||\mathcal{S}||_{\mathsf{sum}} \leq 2||\mathcal{T}||_{\mathsf{max}}.$$

3 Approximating a Minimum Two-Watchman Route

Our algorithm is illustrated in pseudo-code in Figure 1 and we show that it approximates a minmax two-watchman route.

The algorithm begins by running Belleville's algorithm [1] to establish if the polygon is guardable by two point guards. If this is the case, it returns the two point guards computed by the algorithm. Otherwise, it computes, the set of essential extensions $\mathcal{E}$, a

Algorithm *Two-Watchman-Route*
Input: A simple polygon $\mathbf{P}$
Output: A two-watchman route $\mathcal{W}_{\mathsf{T}}$ that sees $\mathbf{P}$
1 Run Belleville's algorithm [1] to establish if the polygon is guardable by two point guards. If this is the case, return the two point guards computed by the algorithm
2 Compute the set of essential extensions $\mathcal{E}$ in $\mathbf{P}$
3 Compute a shortest watchman route $W_{\mathsf{S}} = SVT_{\mathcal{E}}$ in $\mathbf{P}$
4 Let $\mathcal{W}^*_{\mathsf{T}} := (W_{\mathsf{S}}, W_{\mathsf{S}})$
5 **for** every pair of extensions $e_1, e_2 \in \mathcal{E}$, $e_1 \neq e_2$ **do**
5.1 Compute the V-structure $\mathcal{V}_{e_1,e_2}$ and establish its bases q_1 and q_2
5.2 Let $\mathcal{F}_1 := \emptyset$ and $\mathcal{F}_2 := \emptyset$
5.3 **for** every boundary edge $b = [v, v']$ **do**
Compute the minimum tentacle pair $\mathcal{Z}^{\min}_{q_1,q_2}(b) = \mathcal{Z}^{r}_{q_1,q_2}(b)$ giving r on b
if b is a double edge ($r \neq v, v'$) **then**
Let c_1 and c_2 be the cuts through r and the end points of $\mathcal{Z}^{\min}_{q_1}(b)$ and $\mathcal{Z}^{\min}_{q_2}(b)$
Add c_1 to $\mathcal{F}_1$ and c_2 to $\mathcal{F}_2$
else /* b is a single edge ($r = v$ or $r = v'$) */
if $\mathcal{Z}^{\min}_{q_1}(b)$ sees b **then**
Let c and c' be the cuts through v, v' and the end point of $\mathcal{Z}^{\min}_{q_1}(b)$
Add c and c' to $\mathcal{F}_1$
else /* $\mathcal{Z}^{\min}_{q_2}(b)$ sees b */
Let c and c' be the cuts through v, v' and the end point of $\mathcal{Z}^{\min}_{q_2}(b)$
Add c and c' to $\mathcal{F}_2$
5.4 Compute the two tours $\mathcal{W}_{\mathsf{T}} = (SVT_{\mathcal{F}_1}, SVT_{\mathcal{F}_2})$
5.5 **if** $||\mathcal{W}_{\mathsf{T}}||_{\mathsf{max}} < ||\mathcal{W}^*_{\mathsf{T}}||_{\mathsf{max}}$ **then** $\mathcal{W}^*_{\mathsf{T}} := \mathcal{W}_{\mathsf{T}}$
6 **return** $\mathcal{W}^*_{\mathsf{T}}$
End *Two-Watchman-Route*

Figure 1: The Two-Watchman-Route algorithm.

shortest watchman route W_{S} and initializes the solution to be two copies of W_{S}. The rest of this section is devoted to showing how to implement Step 5 of the algorithm.

We claim the following lemma without proof.

Lemma 1 *If two tours in* $\mathbf{P}$ *see all of* $\partial\mathbf{P}$, *then they see all of* $\mathbf{P}$.

The lemma implies that it is sufficient to construct two tours that see the whole boundary of $\mathbf{P}$ to guarantee that all of $\mathbf{P}$ is guarded.

There is a partitioning of the extensions in $\mathcal{E}$ into nonempty subsets $\mathcal{E}_1$ and $\mathcal{E}_2$, such that each tour T_i of a minmax two-watchman route covers the extensions in $\mathcal{E}_i$, $i \in \{1, 2\}$. We even have a stronger claim.

Lemma 2 *Each tour* T_i *in a minmax two-watchman route* $\mathcal{T} = (T_1, T_2)$ *intersects some extension in* $\mathcal{E}_i$.

Consider two tours X_1 and X_2 and a polygon boundary edge b. We claim the following lemma.

Lemma 3 *For any two tours* X_1 *and* X_2 *and a polygon boundary edge* b, *the sets* $\mathbf{VP}(X_i) \cap b$ *and* $\mathbf{VP}(X_1) \cap \mathbf{VP}(X_2) \cap b$ *are each connected.*

For a point q (or an extension e) in $\mathbf{P}$ and a (possibly point sized) subsegment s_b of boundary edge b, we call the shortest path from q (or e) to some point

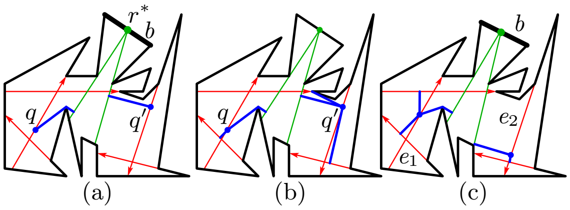

Figure 2: (a) A tentacle pair $\mathcal{Z}^{\min}_{q,q'}(b)$, (b) a jellyfish pair $\mathcal{J}_{q,q'}$, (c) a minimum jellyfish pair $\mathcal{J}^{\min}_{e_1,e_2}$.

in **P** that sees all points of s_b a *tentacle* from q (or e) to s_b, denoted $Z_q(s_b)$ $\big($or $Z_e(s_b)\big)$.

For a boundary segment $b = [v, v']$ and a point r on b, we let $b(r)$ be the subsegment $[v, r]$ and $\bar{b}(r)$ be the subsegment $[r, v']$. For two points q and q' and a point r on b, the *tentacle pair* that sees b is the shorter of the pairs $(Z_q(b(r)), Z_{q'}(\bar{b}(r)))$ and $(Z_q(\bar{b}(r)), Z_{q'}(b(r)))$. We denote this pair $\mathcal{Z}^r_{q,q'}(b)$ and define its length to be the length of the longer of the two tentacles in the pair.

For some point r^* on b, it holds that $||\mathcal{Z}^{r^*}_{q,q'}(b)|| \leq \min_{r\in b}\{||\mathcal{Z}^r_{q,q'}(b)||\}$. If r^* is one of the end points of b, one of the tentacles in the tentacle pair degenerates into a single point q or q'. We denote this minimum tentacle pair by $\mathcal{Z}^{\min}_{q,q'}(b)$. The two tentacles attached to q and q' are denoted $\mathcal{Z}^{\min}_q(b)$ and $\mathcal{Z}^{\min}_{q'}(b)$ respectively; see Figure 2(a); and we have that

$$||\mathcal{Z}^{\min}_{u_1,u_2}(b)|| \leq ||(T_1, T_2)||_{\max}/2, \qquad (1)$$

where u_1 and u_2 are intersection points of T_1 and T_2 with e_1 and e_2 respectively. The inequality holds since T_1 and T_2 together see b.

For two points q and q' in **P**, we call $\mathcal{J}_{q,q'} = \{\mathcal{Z}^{\min}_{q,q'}(b) \mid b \in \partial\mathbf{P}\}$ the *jellyfish pair* with origins q and q'; see Figure 2(b). We define the length of a jellyfish pair to be the length of its longest tentacle.

We define the *bases* along segments s and s' to be a pair of points $(q_*, q'_*) = \arg\min_{q\in s, q'\in s'}\{||\mathcal{J}_{q,q'}||\}$, i.e., two points q_* on s and q'_* on s' where $||\mathcal{J}_{q_*,q'_*}||$ is minimal. We denote the jellyfish pair $\mathcal{J}_{q_*,q'_*}$ by $\mathcal{J}^{\min}_{s,s'}$. From this definition and (1), we have

$$||\mathcal{J}^{\min}_{e_1,e_2}|| \leq ||\mathcal{J}_{u_1,u_2}|| \leq ||(T_1, T_2)||_{\max}/2. \qquad (2)$$

We can select two longest tentacle pairs of $\mathcal{J}^{\min}_{e_1,e_2}$, at least one pair of which attains the length $||\mathcal{J}^{\min}_{e_1,e_2}||$. The two tentacle pairs have two bases q_1 on e_1 and q_2 on e_2, one pair is the shortest tentacle pair $\mathcal{Z}^{\min}_{q_1,q_2}(b)$, the other is the shortest tentacle pair $\mathcal{Z}^{\min}_{q_1,q_2}(b')$, for boundary edges b and b'. We call the two tentacle pairs that attain the maximum length a *V-structure* on e_1 and e_2, and denote it $\mathcal{V}_{e_1,e_2}$. The length of $\mathcal{V}_{e_1,e_2}$ is the length of its longest tentacle. From this definition and (2) we have

$$||\mathcal{V}_{e_1,e_2}|| = ||\mathcal{J}^{\min}_{e_1,e_2}|| \leq ||(T_1, T_2)||_{\max}/2. \qquad (3)$$

The algorithm needs to find the two bases q_1 on e_1 and q_2 on e_2. Therefore, the algorithm must determine the two boundary edges b and b', and the two points r and r' on b and b' for which the maximum length of the V-structure is attained. Since we do not know which pair of boundary edges produce the V-structure that attains the length of $\mathcal{J}^{\min}_{e_1,e_2}$, we try all possible pairs of boundary edges $b_i = [v_i, v'_i]$ and $b_j = [v_j, v'_j]$, $1 \leq i \leq j \leq n$ in Step 5.1 of the algorithm. We allow $i = j$ to take care of the case when the longest tentacle in $\mathcal{J}^{\min}_{e_1,e_2}$ is unique.

In Step 5.1, we begin by computing $Z_{e_1}(b_i)$ and $Z_{e_1}(b_j)$ as well as the two pairs $Z_{e_2}(v_i)$, $Z_{e_2}(v'_i)$ and $Z_{e_2}(v_j)$, $Z_{e_2}(v'_j)$. Assume that $Z_{e_2}(v_i)$ and $Z_{e_2}(v_j)$ are the shorter of the two tentacles in each pair.

We obtain the two points q and q' on the extensions e_1 and e_2 that minimize $\max\{||Z_q(b_i)||, ||Z_q(b_j)||\}$ and $\max\{||Z_{q'}(v_i)||, ||Z_{q'}(v_j)||\}$. We let two points r_i on b_i and r_j on b_j slide independently, r_i from v_i to v'_i and r_j from v_j to v'_j. We can express the position on e_1 of q and on e_2 of q' as functions of r_i and r_j, and hence also the expressions $\max\{||Z_q(b_i(r_i))||, ||Z_q(b_j(r_j))||\}$ and $\max\{||Z_{q'}(\bar{b}_i(r_i))||, ||Z_{q'}(\bar{b}_j(r_j))||\}$.

The difference between these two expressions is a multivariate function $\mathcal{D}_{ij}(r_i, r_j)$ on r_i and r_j that locally only depends on the contact points of the supporting segments for r_i and r_j and the corresponding paths $Z_q(b_i(r_i))$, $Z_q(b_j(r_j))$, $Z_{q'}(\bar{b}_i(r_i))$, and $Z_{q'}(\bar{b}_j(r_j))$, a total of at most eight polygon vertices. We compute the values of r_i and r_j that produce the minimum absolute value $|\mathcal{D}_{ij}(r_i, r_j)|$ in all intervals for r_i and r_j where the contact points do not change.[1] As r_i moves from v_i to v'_i, the supporting lines for r_i can change at most $O(n)$ times and the same holds for r_j so in at most $O(n^2)$ time the minimum can be obtained. We maintain the pair of bases q and q' for which the corresponding V-structure $\mathcal{V}_{e_1,e_2}$ has maximum length. We denote these points by q_1 and q_2.

Given q_1 and q_2, we compute, in Step 5.3, the minimum tentacle pairs $\mathcal{Z}^{\min}_{q_1,q_2}(b) = \mathcal{Z}^r_{q_1,q_2}(b)$ for every boundary edge $b = [v, v']$, giving us the minimum jellyfish pair on e_1 and e_2, $\mathcal{J}^{\min}_{e_1,e_2}$. If the expression $\max\{||\mathcal{Z}^{\min}_{q_1}(b)||, ||\mathcal{Z}^{\min}_{q_2}(b)||\}$ is minimized for $r = v$ or $r = v'$, then b is a *single edge*, otherwise it is a *double edge*. If b is a double edge, the point r and the endpoint of $\mathcal{Z}^{\min}_{q_1}(b)$ different from q_1 defines a cut in **P** that passes through the two points. The direction of the cut is such that q_1 does not lie to the left of the cut and is added to the set $\mathcal{F}_1$. We also construct the cut through r and the endpoint of $\mathcal{Z}^{\min}_{q_2}(b)$ different from q_2. This cut is directed so that q_2 does not lie to the left of the cut and is added to the set $\mathcal{F}_2$. The green segments in Figure 2(c) are the two cuts for boundary edge b.

Similarly, if b is a single edge, it is seen by one of $\mathcal{Z}^{\min}_{q_1}(b)$ or $\mathcal{Z}^{\min}_{q_2}(b)$. If it is seen by $\mathcal{Z}^{\min}_{q_1}(b)$, the endpoints v and v' of b together with the endpoint of

[1] We assume a real RAM computational model that allows us to compute arbitrary algebraic functions and roots of algebraic functions.

$\mathcal{Z}_{q_1}^{\min}(b)$ different from q_1 define two cuts. The direction of the cuts are such that q_1 does not lie to the left of them and they are added to the set $\mathcal{F}_1$. If b is seen by $\mathcal{Z}_{q_2}^{\min}(b)$, we construct two cuts in the same way and add these to $\mathcal{F}_2$.

To finalize, we let $W_1 = SVT_{\mathcal{F}_1}$ and $W_2 = SVT_{\mathcal{F}_2}$, two shortest visiting tours of the cut sets $\mathcal{F}_1$ and $\mathcal{F}_2$, and return the pair (W_1, W_2) as our two-watchman route.

Lemma 4 *The tours (W_1, W_2) obtained by algorithm Two-Watchman-Route form a two-watchman route and $||(W_1, W_2)||_{\max} \leq (\pi + 4)||(T_1, T_2)||_{\max}$.*

Proof. (Sketch) It follows from Lemma 1 and the fact that the two tours together see every boundary edge that they form a two-watchman route.

The algorithm computes the minimum jellyfish pair $\mathcal{J}_{e_1,e_2}^{\min}$ in the loop of Step 5.3. By trying all pairs of extensions in Step 5, the algorithm must necessarily consider a pair intersected by the tours T_1 and T_2; see Lemma 2. Consider the tentacles attached to the base q_1 on e_1. If we follow the shortest path from each tentacle endpoint not on q_1 to the next, cyclically around q_1, we obtain a tour U_1 that visits every cut in the set $\mathcal{F}_1$. Every tentacle has length at most $R = ||(T_1, T_2)||_{\max}/2$ by (2), hence U_1 is inscribed in a circle of radius R. Since $||W_1|| \leq ||U_1||$, the convex chains of W_1 together have length $\leq 2\pi R$.

If T_1 intersects T_2, then $||W_s|| \leq ||T_1|| + ||T_2|| \leq 2||(T_1, T_2)||_{\max}$ proving the lemma since $||(W_1, W_2)||_{\max} \leq ||W_s||$.

If T_1 does not intersect T_2 and W_1 has at most four reflex chains, then $||W_1|| \leq 2\pi R + 8R \leq (\pi + 4) \cdot ||(T_1, T_2)||_{\max}$.

If T_1 does not intersect T_2, W_1 has at least five reflex chains and W_1 does not intersect T_2, then use the segments of W_1 to cut $\mathbf{P}$, thus partitioning $\mathbf{P}$ into separate components. Let $\mathbf{Q}$ be the component containing T_2. The convex chain W_c of W_1 bounding $\mathbf{Q}$ has length $\leq 2\pi R$. The two reflex chains of W_1 adjacent to W_c have length $\leq 4R$ and the remainder of W_1 follows the same path as T_1, giving us $||W_1|| \leq ||T_1|| + 4R + 2\pi R \leq (\pi + 3)||(T_1, T_2)||_{\max}$.

Finally, if W_1 intersects T_2, then use T_2 to cut $\mathbf{P}$, partitioning it into components. Let $\mathbf{Q}'$ be the component containing T_1. The intersection $W_c' = W_1 \cap \mathbf{Q}'$ follows the same path as T_1, the two reflex chains of W_1 adjacent to W_c' have length $\leq 4R$ and the remaining reflex chains of $W_c'' = W_1 \cap (\mathbf{P} \setminus \mathbf{Q}')$ follow T_2. The convex chains of W_c'' have total length $\leq 2\pi R$ so we have $||W_1|| \leq ||T_1|| + 4R + 2\pi R + ||T_2|| \leq (\pi + 4)||(T_1, T_2)||_{\max}$.

We bound W_2 similarly, proving the lemma. ☐

The analysis of the algorithm is straightforward. The for-loop in Step 5 considers $O(n^2)$ pairs of extensions. Computing $\mathcal{V}_{e_1,e_2}$ takes $O(n^4)$ time by going through all pairs of boundary edges. The work in the remaining steps of the outermost for-loop is dominated by the cost of computing the shortest visiting tours in Step 5.4 taking $O(n^4 \log n)$ time. Hence, the total time complexity for the algorithm is $O(n^6 \log n)$.

Theorem 5 *The Two-Watchman-Route algorithm computes a 7.1416-approximation of the minmax two-watchman route in $O(n^6 \log n)$ time.*

4 Conclusions

Our algorithm relies heavily on the fact that for two tours it is sufficient to guarantee that the boundary is seen to ensure that the complete polygon is seen. This does not hold for three or more tours. It is therefore very possible that the problem is inapproximable for three watchmen.

Establishing the complexity for the minsum two-watchman route is still open although our algorithm provides a polynomial 14.2832-approximation.

The authors would like to thank Paweł Żyliński for fruitful discussions.

References

[1] P. Belleville. Two-guarding simple polygons. Master's thesis, School of Computer Science, McGill University, September 1991.

[2] 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.

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

[4] M. Dror, A. Efrat, A. Lubiw, and J.S.B. Mitchell. Touring a sequence of polygons. In *Proc. 35th ACM Symposium on Theory of Computing, STOC'03*, pages 473–482. ACM, 2003.

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

[6] J.E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*, 2nd ed. CRC Press, 2004.

[7] J.S.B. Mitchell. Approximating watchman routes. In *Proc. 24th ACM-SIAM Symposium on Discrete Algorithms, SODA'13*, pages 844–855, 2013.

[8] J.S.B. Mitchell and E.L. Wynters. Watchman routes for multiple guards. In *Proc. 3rd Canadian Conference on Computational Geometry*, pages 126–129, 1991.

[9] E. Packer. Computing multiple watchman routes. In *Proc. 7th International Workshop, WEA 2008*, pages 114–128. Springer Verlag, Lecture Notes in Computer Science 5038, 2008.

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

[11] 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.