Shortest Watchman Tours in Simple Polygons under Rotated Monotone Visibility

Bengt J. Nilsson∗, David Orden †, Leonidas Palios‡, Carlos Seara§, and Paweł Żyliński¶

1Malmö University, Sweden. 2Universidad de Alcalá, Spain. 3University of Ioannina, Greece. 4Universitat Politècnica de Catalunya, Spain. 5University of Gdańsk, Poland.

Abstract

We present an \(O(nrG)\) time algorithm for computing and maintaining a minimum length shortest watchman tour that sees a simple polygon under monotone visibility in direction \(\theta\), while \(\theta\) varies in \([0, 180^\circ]\), obtaining the directions for the tour to be the shortest one over all tours, where \(n\) is the number of vertices, \(r\) is the number of reflex vertices, and \(G \leq r\) is the maximum number of gates of the polygon used at any time in the algorithm.

Keywords: Watchman tour, simple polygons, monotone, rotating

1 Introduction

Arguably, problems concerning visibility and motion planning in polygonal environments are among the most well-studied in computational geometry. A problem that encompasses both the visibility and motion planning aspects is that of computing

∗Email: bengt.nilsson.TS@mau.se
†Email: david.orden@uah.es
‡Email: palios@cs.uoi.gr
§Email: carlos.seara@upc.edu
¶Email: pawel.zyliinski@ug.edu.pl

This work has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734922.
a shortest watchman tour in an environment, i.e., the shortest closed tour that sees the complete free space of the environment. This problem, under the standard visibility model where two points see each other if the line segment connecting them does not intersect the exterior of the environment free space, has been shown NP-hard \cite{6,9} and even $\Omega(\log n)$-inapproximable \cite{18} for polygons with holes having a total of $n$ segments. With respect to simple polygons, Chin and Ntafos \cite{6} showed a linear time algorithm to compute a shortest watchman tour in a simple rectilinear polygon. Then, after a few false starts \cite{7,12,28,29}, Tan et al. \cite{30} proved an $O(n^4)$ time dynamic programming algorithm for computing a shortest watchman tour through a given boundary point in an arbitrary simple polygon, the so-called fixed watchman tour. This was later improved by Dror et al. \cite{8} to $O(n^3 \log n)$ time. Carlsson et al. \cite{4} showed how to generalize algorithms for a shortest fixed watchman tour to compute a shortest watchman tour in a simple polygon without any pre-specified point to pass through, a floating watchman tour, using quadratic factor overhead. Tan \cite{27} improved this to a linear factor overhead, thus establishing an $O(n^4 \log n)$ time for the floating case, if combined with the algorithm by Dror et al. \cite{8}.

Imagine now that a navigating robot is capable only to see along paths restricted to some (predefined) orientations, with respect to the coordinate system of its free space map. Since changing this coordinate system may affect an optimal trajectory of the robot, a natural variation on the aforementioned watchman tour problem is to allow, as an interrelated optimization process, rotation of the coordinate system (equiv. the input polygon), in order to obtain a better solution. Whereas under standard visibility, while rotating the coordinate system, the length of its shortest watchman tour is an invariant, under a non-standard one, it may not be the case, and moreover, the lengths of such tours may vary arbitrarily. Therefore in this paper, inspired also by some already existing work on the issue of optimizing with rotation \cite{11,21}, we introduce the problem of computing the orientation $\theta$ that minimizes the length of a watchmen tour taken over all rotations of the coordinate system, under monotone visibility \cite{3}. In particular, we present an $O(nrG)$ time algorithm for computing and maintaining a shortest (floating) watchman tour that sees a simple polygon under monotone visibility in direction $\theta$, while $\theta$ varies in $[0, 180^\circ)$, obtaining the directions for the tour to be the shortest one over all tours, where $n$ is the number of vertices, $r$ is the number of reflex vertices, and $G$ is the maximum number of gates of the polygon used at any time in the algorithm; see Section 2 for a formal definition. In particular, we have $G \leq r$ in all cases.

Visibility plays a central role in diverse advanced application areas, for example, in surveillance, computer graphics, sensor placement, and motion planning as well as in wireless communication. Therefore, our particular monotone visibility model has practical applications in material processing and manufacturing. We also note in passing that the problem of computing the shortest watchman tour for a given poly-
gon, under rotated monotone visibility, is related to a variety of problems concerning
the concept of “oriented” kernels in polygons, which has already attracted attention
in the literature. In particular, for a given set $\mathcal{O}$ of predefined directions, Schuierer
et al. [24] provided an algorithm to compute the $\mathcal{O}$-Kernel of a simple polygon.
Next, Schuierer and Wood [25] introduced the concept of the external $\mathcal{O}$-Kernel of
a polygon, in order to compute the $\mathcal{O}$-Kernel of a simple polygon with holes. In
addition, when restricted to $\mathcal{O} = \{0^\circ, 90^\circ\}$, Gewali [11] described a linear algorithm
for orthogonal polygons without holes, and a quadratic one for orthogonal polygons
with holes, whereas Palios [23] gave an output-sensitive algorithm for that problem
in orthogonal polygons with holes. More recently, Orden et al. [22] presented al-
gorithms for computing the orientations $\theta$ in $[-90^\circ, 90^\circ)$ such that the $\{\theta\}$-Kernel
of a simple (or orthogonal) polygon is not empty, has maximum/minimum area or
maximum/minimum perimeter.

Our work is organized as follows. Section 2 contains preliminary results and an
overview of the linear-time algorithm for computing a shortest watchman tour in
rectilinear polygons, being the basis for our approach. In Section 3, we present our
algorithm and prove its correctness. Finally, in Section 4, we analyze the running
time of the algorithm and conclude the presentation in Section 5.

2 Preliminaries

Let $\theta$ be a direction, specified using its angle to the $x$-axis, and let $\mathbf{P}$ be a simple
polygon having $n$ edges. We define a path $\Pi$ inside $\mathbf{P}$ to be monotone w.r.t. direction
$\theta$ or $\theta$-monotone if and only if the intersection between any line parallel to direction
$\theta$ and $\Pi$ is a connected set. In standard visibility, two points $p$ and $q$ in $\mathbf{P}$ see each
other if and only if the line segment between $p$ and $q$ does not intersect the exterior
of $\mathbf{P}$. In $\theta$-monotone visibility, two points $p$ and $q$ $\theta$-see each other if and only if
there is a $\theta$-monotone path between $p$ and $q$ not intersecting the exterior of $\mathbf{P}$.

Let $v$ be a reflex vertex of $\mathbf{P}$ incident to the boundary edge $e$. If we extend $e$
maximally inside $\mathbf{P}$, we obtain a line segment $\bar{e}$ collinear with $e$, and we can associate
the direction to $\bar{e}$ being the same as that of $e$ as we traverse the boundary of $\mathbf{P}$ in
clockwise order; see Figure 1(a). Therefore, any reflex vertex is adjacent to
two extensions in $\mathbf{P}$. Any directed segment in $\mathbf{P}$ intersecting the interior of $\mathbf{P}$ and
connecting two boundary points of $\mathbf{P}$ is called a cut of $\mathbf{P}$. Thus, an extension $\bar{e}$ is
a cut in $\mathbf{P}$. A cut $c$ partitions $\mathbf{P}$ into two components, $L(c)$, the component locally
to the left of $c$ according to its direction, and $R(c)$, the component locally to the
right of $c$ according to its direction. Consider now a closed tour $T$ inside $\mathbf{P}$. Chin
and Ntafos [10, 11] argue for standard visibility that in order for $T$ to see the whole
polygon $\mathbf{P}$, it is sufficient for $T$ to see the vertices of $\mathbf{P}$, and therefore, it is sufficient
for $T$ to have a point in $L(\bar{e})$ for every extension $\bar{e}$ in $\mathbf{P}$, i.e., to intersect the left of
any extension in $\mathbf{P}$; again see Figure 1(a).
Next, for monotone visibility in direction $\theta$, we introduce the following definitions. Consider a reflex vertex $v$ of $P$ and let $l$ be the directed line with angle $\theta$ to the $x$-axis passing through $v$. If the two boundary edges incident to $v$ lie on the same side of $l$, let $s$ be the maximal segment collinear to $l$ inside $P$ that passes through $v$. The segment $s$ partitions $P$ into three components incident to $v$. Two components $P_L$ and $P_R$ have $v$ as a convex vertex and the third component has $s$ as a boundary edge; $P_R$ is the first subpolygon traversed by the counterclockwise traversal of $P$ starting at $v$, whereas $P_L$ is the other component having $v$ as convex vertex. The problem thus reduces to obtaining the shortest tour that intersect the left of a given set of cuts.

Now, we can argue similarly for monotone visibility in direction $\theta$ as in the standard visibility case. Assume the tour $T$ has points in $P_R$. Unless $T$ intersects $s$, $T$ cannot see any points in $P_L$. Similarly, if $T$ has points in $P_L$, then $T$ cannot see any points in $P_R$, unless it intersects $s$. To mimic the standard visibility situation, we introduce two cuts $c_l$ and $c_b$ incident to $v$, where $c_l$ is the portion of $s$ bounding $P_R$, and $c_b$ is the portion of $s$ bounding $P_L$. Specifically, the cut $c_l$ is directed away from $v$ and we call it a forward $\theta$-cut, and symmetrically, the cut $c_b$ is directed towards $v$ and we call it a backward $\theta$-cut; see Figure 1(b).

We also color the $\theta$-cuts and their associated reflex vertices. A $\theta$-cut is red, if the boundary edges incident to the associated vertex $v$ both lie locally to the right of the directed line $l$ defined above. The vertex $v$ is thus called a red vertex. Analogously, a $\theta$-cut is blue, if the boundary edges incident to the associated vertex $v$ both lie locally to the left of the directed line $l$, and $v$ is called a blue vertex. Other reflex vertices are not colored as they do not break monotonicity w.r.t. direction $\theta$ and are therefore not used; see Figure 1(b).

Similarly to the standard visibility case, we define the region $L(c)$ for a $\theta$-cut $c$ to be the part of the polygon $P$ locally to the left of $c$ according to its direction, and $R(c)$ to be the part of the polygon $P$ locally to the right of $c$. We claim the following lemma that corresponds to the standard visibility case [7]; its correctness follows from the definition of the $\theta$-cut.
Lemma 2.1. A tour $T$ in $P$ is a watchman tour under $\theta$-monotone visibility if and only if it intersects the region $L(c)$, for every $\theta$-cut $c$ in $P$.

This lemma allows us to use the algorithm of Chin and Ntafos [6] for computing the shortest watchman tour under $\theta$-monotone visibility, since it computes the shortest tour that intersect the left of a set of cuts. The algorithm works roughly as follows. First, it identifies the proper set of cuts inside the polygon. Second, it reduces the shortest tour problem to a shortest path problem in a triangulated two-manifold, computes the shortest path, and transforms the path to a tour in the original polygon. For the classical watchman tour in a simple polygon, the first step can be done with a ray shooting data structure [11, 14] in total $O(n \log n)$ time. In our case, since all $\theta$-cuts are parallel line segments, we can do the first step in $O(n)$ time using the algorithm by Chazelle [5] that partitions a simple polygon into $O(n)$ visibility trapezoids by introducing parallel line segments at reflex vertices in the polygon.

Similarly as for standard visibility, we define a dominance relation between cuts. Given two $\theta$-cuts $c$ and $c'$, we say that $c$ dominates $c'$ if $L(c) \subset L(c')$. We call $c$ a dominating $\theta$-cut or gate, if $c$ is not dominated by any other $\theta$-cut in $P$. We refer to the issuing reflex vertex of a gate as the gate vertex, whereas the edge touched by the other endpoint of the gate is called the gate edge. Carlsson et al. [4] show how to compute the dominating cuts in $P$ with the standard visibility in $O(n)$ time, given the complete set of cuts ordered along the boundary. Their method transfers directly to the case of $\theta$-monotone visibility, since given the trapezoidation of the polygon, the ordering can be obtained by a traversal of the boundary, ordering the forward $\theta$-cuts and the backward $\theta$-cuts separately, and then merging these two sets of $\theta$-cuts. The process thus takes $O(n)$ time in total.

In the algorithm, we establish gates as explained above and remove the portions of the polygon that lie locally to the left of them, resulting in the polygon $P'(\theta)$; see Figure 2(a). The optimal tour will only reflect on the gates in $P$ to see everything on the other side of them, so it is completely contained in $P'(\theta)$. We then triangulate $P'(\theta)$ and establish a constant-size subset $V$ of vertices such that the optimal tour must pass through at least one of them; see Section 4. In the next step, we compute for each vertex $v$ in $V$, a triangulated two-manifold $H_v(\theta)$ (see Section 2.1) such that the shortest path from $v$ to its image $v'$ in $H_v(\theta)$ corresponds to the shortest watchman tour in $P$ that passes through $v$. We then establish the shortest path $S(v, v')$ in $H_v(\theta)$, for each $v$ in $V$, pick the shortest of these paths, and finally transform it back to the polygon $P'(\theta)$. The whole computation can be done in $O(n)$ time [6].
2.1 The Two-Manifold

In our approach, we extensively exploit the concept of the two-manifold used in the algorithm by Chin and Ntafos [6]. We therefore provide some more details.

Without loss of generality assume that $\theta = 0^\circ$. Consider the dominant extensions (or gates in our case) in $P$. Chin and Ntafos [6, 7] prove that an optimal watchman tour will never intersect the interior of any region $L(c)$ for any dominant extension $c$. Thus, we define $P' = P'(0^\circ) \overset{\text{def}}{=} (P \setminus \bigcup_{c \in G} L(c))^*$, where $G$ is the set of dominant extensions and $S^*$ denotes the closure of a set $S$. (We take the closure to include the boundary points of $P'$.) The dominant extensions (gates) of $P$ are now part of the boundary of $P'$, and so we refer to them as the essential edges of $P'$. The polygon $P'$ is triangulated and, given a vertex $v$, $P'$ is then unrolled from $v$ to $v'$ using the essential edges as mirrors giving $H_v \left(= H_v(0^\circ)\right)$, where $v'$ is the image of $v$, as follows. The counterclockwise traversal of the boundary of $P'$ starting at $v$ encounters the incident triangles from the triangulation in order along the edges and vertices of the traversal. When reaching an essential edge, we reflect all subsequent triangles using the essential edge as mirror. As the traversal continues, we repeat this step until the traversal reaches the vertex $v$ again. In the two-manifold, the second instance of the vertex $v$ is called the image of $v$ and denoted $v'$. Between the vertex $v$ and the first essential edge, subsequent consecutive pairs of essential edges, and between the final essential edge and $v'$, we perform a standard breadth-first-search to keep those triangles of the triangulation that form a path between the mirroring segments in $H_v$ (see Figure 2), as this will aid the shortest path finding algorithm in later steps [11, 17]. The size of $H_v$ is linear, because each triangle from the triangulation of $P'$ is used at most six times in the construction of $H_v$, once for each side of the triangle and once for each vertex of the triangle, since each boundary edge and vertex is passed only once as we perform the traversal of $P'$.

From Heron [13], we know that the shortest path between two points that also touches a line (with both points on the same side of the line) makes perfect reflection on the line; see Figure 2(b, c). Thus, the shortest path in $H_v$ between $v$ and its image $v'$ corresponds exactly to the shortest watchman tour in $P$ that passes through $v$ as the path is folded back along the gates in $P$. Computing the shortest path in a triangulated simple polygon can be done in linear time [11, 17], and since $H_v$ consists of a linear number of connected triangles, computing the path takes $O(n)$ time. Folding back the path to obtain the tour also takes $O(n)$ time by traversing the path and computing the points of intersection between the path and the triangle sides corresponding to gates in $P$. 
3 The Algorithm

Given that we can compute the shortest watchman tour under monotone visibility in a specific direction $\theta$ in linear time, our objective is to find the direction $\theta$ for which the length of a shortest watchman tour under monotone visibility in this direction is minimal. Let $T(\theta)$ denote a shortest watchman tour under monotone visibility in direction $\theta$. The idea of the algorithm is to compute the tour $T(0^\circ)$, and then rotate the direction $\theta$ from $0^\circ$ to $180^\circ$, updating $T(\theta)$ as the rotation proceeds.

Consider the tour $T(\theta)$ for a fixed direction $\theta$ and let $G(\theta)$ be the set of gates visited by $T(\theta)$. Treat $T(\theta)$ as a (weakly simple) polygon and divide the vertex set of $T(\theta)$ into two types. The stable vertices of $T(\theta)$ coincide with reflex vertices of $P$, even if they sometimes correspond to convex vertices in $T(\theta)$; see Figure 2(c). The moving vertices of $T(\theta)$ are the reflections on the gates in $G(\theta)$. We partition $T(\theta)$ into subpaths going from one stable vertex to the next one in counterclockwise order along $T(\theta)$. Any such path is either a line segment between stable vertices of $T(\theta)$ or it is a path that starts at a stable vertex, passes a consecutive sequence of moving vertices, and finishes then again at a stable vertex. We call such a subpath a maximal moving subpath of $T(\theta)$. Any maximal moving subpath of $T(\theta)$ has the following property.

**Lemma 3.1.** A maximal moving subpath $C(\theta)$ of $T(\theta)$ has at most three moving vertices and they touch gates in order having alternating colors.

**Proof.** Let $v$ be the stable vertex at the first endpoint of $C(\theta)$. Let $S(v, v')$ be the shortest path between $v$ and its image $v'$ in the two-manifold $H_v(\theta)$. The moving vertices of $C(\theta)$ correspond to consecutive crossings of gates by $S(v, v')$ in $H_v(\theta)$ without touching a stable vertex. Since all gates are parallel, when following $S(v, v')$, no two consecutive gates in $H_v(\theta)$ can have the same color without $S(v, v')$ (and thus $T(\theta)$) touching a stable vertex, otherwise one of them dominates the other,
contradicting that they are both gates. Thus, the sequence of consecutive gates in $H_v(\theta)$ is color alternating. Next, it is clear that the sequence of gates cannot consist of more than three gates, since four or more would mean that $T(\theta)$ is self intersecting and thus could be shortened [3] [7].

Given $T(\theta)$ and the set of gates $G(\theta)$, assume we increase the rotation to $\theta + \varepsilon$ to obtain the tour $T(\theta + \varepsilon)$ and the set of gates $G(\theta + \varepsilon)$; we refer to such a rotation as an $\varepsilon$-rotation. We say that $T(\theta)$ and $T(\theta + \varepsilon)$ are close if each of the following properties hold:

1. The stable vertices of $T(\theta)$ and $T(\theta + \varepsilon)$ are the same.
2. The gate vertices for the gates in $G(\theta)$ and $G(\theta + \varepsilon)$ in $P$ are the same.
3. For any pair of gates $g \in G(\theta)$ and $g_e \in G(\theta + \varepsilon)$ with the same gate vertex, they also have the same gate edge.
4. For any pair of gates $g \in G(\theta)$ and $g_e \in G(\theta + \varepsilon)$ with the same gate vertex $v_g$, if $T(\theta)$ touches $v_g$, then $T(\theta + \varepsilon)$ also touches $v_g$, and if $T(\theta)$ touches the other endpoint of $g$, then $T(\theta + \varepsilon)$ also touches the other endpoint of $g_e$.

We claim the following lemma.

**Lemma 3.2.** If $T(\theta)$ and $T(\theta + \varepsilon)$ are close, then $\|T(\theta + \varepsilon)\| - \|T(\theta)\| + \sum_{k=1}^{\lvert G(\theta) \rvert} f_k(\varepsilon),$

where

$$f_k(\varepsilon) = \sqrt{\frac{a_{k,0}^2 + a_{k,1} \tan \varepsilon + a_{k,2} \tan^2 \varepsilon}{1 + a_{k,3} \tan \varepsilon + a_{k,4} \tan^2 \varepsilon} - a_{k,0}} + \sqrt{\frac{a_{k,5}^2 + a_{k,6} \tan \varepsilon + a_{k,7} \tan^2 \varepsilon}{1 + a_{k,8} \tan \varepsilon + a_{k,9} \tan^2 \varepsilon} - a_{k,5}}$$

$$+ \sqrt{\frac{a_{k,10}^2 + a_{k,11} \tan \varepsilon + a_{k,12} \tan^2 \varepsilon}{1 + a_{k,15} \tan \varepsilon + a_{k,16} \tan^2 \varepsilon} - a_{k,15}}$$

$$+ \sqrt{\frac{a_{k,19}^2 + \sum_{i=1}^{14} a_{k,19+i} \tan^i \varepsilon}{1 + \sum_{i=1}^{14} a_{k,33+i} \tan^i \varepsilon} - a_{k,19}},$$

for some constants $a_{k,0}, \ldots, a_{k,47}, \ 1 \leq k \leq \lvert G(\theta) \rvert$, only depending on the stable vertices, the gate vertices, the gate edges, and the angle $\theta$.

**Proof.** Let $C(\theta)$ be a maximal moving subpath of $T(\theta)$. We consider three cases, depending on the number (at most three by Lemma 3.1) of gates touched by $C(\theta)$.

Case 1: $C(\theta)$ touches only one gate $g$. Assume without loss of generality that $g$ is red and has gate vertex $v_g$, the other case is completely symmetric; see Figure 3(a). Let $u$ and $v$ be the endpoints of $C(\theta)$, where $v$ is reached before $u$ along a counterclockwise traversal of $T(\theta)$ starting at a point not on $C(\theta)$.
If $C(\theta)$ does not touch any endpoint of $g = g(\theta)$, then by reflecting $u$ along $g$, we obtain the point $u'$, where the coordinates only depend on $u$ and $g$. Thus, $\|C(\theta)\| = \|v, u'\| = \|v, p\| + \|p, u\|$, where $p$ is the intersection between $[v, u']$ and $g$. Similarly, $\|C(\theta + \varepsilon)\| = \|v, u'_{\varepsilon}\|$, where $u'_{\varepsilon}$ is the reflection of $u$ along $g_{\varepsilon}$; see Figure 3(a). By routine trigonometry, we have

$$f(\varepsilon) = \|C(\theta + \varepsilon)\| - \|C(\theta)\| = \sqrt{a_0^2 + a_1 \tan \varepsilon + a_2 \tan^2 \varepsilon} - a_0,$$  \hspace{1cm} (1)

for some constants $a_0, \ldots, a_4$ only depending on the points $u, v, v_g$, and the angle $\theta$.

Assume next that $C(\theta)$ touches an endpoint of $g$. If that endpoint is $v_g$, then $\|C(\theta)\| = \|C(\theta + \varepsilon)\|$, and so $f(\varepsilon) = 0$. If $C(\theta)$ touches the other endpoint of $g$, then

$$f(\varepsilon) = \sqrt{a_5^2 + a_6 \tan \varepsilon + a_7 \tan^2 \varepsilon} - a_5,$$ \hspace{1cm} (2)

for some constants $a_0, \ldots, a_9$ only depending on the points $u, v, v_g$, the gate edge, and the angle $\theta$; see Figure 3(b).

Case 2: $C(\theta)$ touches two gates $g$ and $g'$. First, it follows from the proof of Lemma 3.1 that $g$ and $g'$ cannot have the same color. Next, consider the segment $s$ of $C(\theta)$ connecting $g$ and $g'$. $C(\theta)$ cannot have its endpoints on either side of $s$ (in $P'(\theta)$), as otherwise $T(\theta)$ must self intersect, a contradiction [7]. Thus, both $u$ and $v$ lie on the same side of $s$, whereby $s$ must touch the boundary of $P$, as otherwise $T(\theta)$ could be made shorter by shifting $s$ along the gates $g$ and $g'$ towards $u$ and $v$, another contradiction. Since our assumption is that $C(\theta)$ does not touch a stable vertex in its interior (a reflex vertex in $P$), $s$ must touch at least one endpoint of one of the gates $g$ or $g'$. If this endpoint is a gate vertex, we can view this gate vertex as a fixed point of $T(\theta)$ (or stable vertex) and either $f(\varepsilon) = 0$, if also the other
endpoint of \( s \) touches a gate vertex, or one of Equalities (1) and (2) hold by the previous argument.

If \( s \) touches the boundary of \( P \) at a gate edge, we have two cases. Either, the other endpoint of \( s \) touches the interior of the other gate or it also touches a gate edge. In the first case, we have by routine trigonometry that

\[
f(\varepsilon) = \sqrt{\frac{a_0^2 + a_1 \tan \varepsilon + a_2 \tan^2 \varepsilon}{1 + a_3 \tan \varepsilon + a_4 \tan^2 \varepsilon}} - a_0 + \sqrt{\frac{a_5^2 + \sum_{i=1}^{12} a_{5+i} \tan^i \varepsilon}{1 + \sum_{i=1}^{12} a_{17+i} \tan^i \varepsilon}} - a_5,
\]

for some constants \( a_0, \ldots, a_{29} \) only depending on the points \( u, v, v_g, v_g' \), the touched gate edge, and the angle \( \theta \); see Figure 3(c). In the second case, we have two endpoints of three segments that both touch gate edges, and we obtain

\[
f(\varepsilon) = \sqrt{\frac{a_0^2 + a_1 \tan \varepsilon + a_2 \tan^2 \varepsilon}{1 + a_3 \tan \varepsilon + a_4 \tan^2 \varepsilon}} - a_0 + \sqrt{\frac{a_5^2 + a_6 \tan \varepsilon + a_7 \tan^2 \varepsilon}{1 + a_{8} \tan \varepsilon + a_9 \tan^2 \varepsilon}} - a_5
\]

\[+ \sqrt{\frac{a_{10}^2 + a_{11} \tan \varepsilon + a_{12} \tan^2 \varepsilon + a_{13} \tan^3 \varepsilon + a_{14} \tan^4 \varepsilon}{1 + a_{15} \tan \varepsilon + a_{16} \tan^2 \varepsilon + a_{17} \tan^3 \varepsilon + a_{18} \tan^4 \varepsilon}} - a_{10},
\]

for some constants \( a_0, \ldots, a_{18} \) only depending on the points \( u, v, v_g, v_g' \), the two touched gate edges, and the angle \( \theta \).

Case 3: \( C(\theta) \) touches three gates \( g, g', \) and \( g'' \). Again, observe that they must have alternating color (see the proof of Lemma 3.1). Next, by the same argument as in the previous case, at least two segments of \( C(\theta) \) connecting the three gates must touch the boundary of \( P \). If such a touching point is a gate vertex, this reduces to one of Equalities (1)–(4) since we can view this vertex of \( T(\theta) \) as stable. If \( C(\theta) \) touches the middle gate in the interior, then again by routine trigonometry we obtain

\[
f(\varepsilon) = \sqrt{\frac{a_0^2 + a_1 \tan \varepsilon + a_2 \tan^2 \varepsilon}{1 + a_3 \tan \varepsilon + a_4 \tan^2 \varepsilon}} - a_0 + \sqrt{\frac{a_5^2 + a_6 \tan \varepsilon + a_7 \tan^2 \varepsilon}{1 + a_{8} \tan \varepsilon + a_9 \tan^2 \varepsilon}} - a_5
\]

\[+ \sqrt{\frac{a_{10}^2 + \sum_{i=1}^{14} a_{10+i} \tan^i \varepsilon}{1 + \sum_{i=1}^{14} a_{24+i} \tan^i \varepsilon}} - a_{10},
\]

for some constants \( a_0, \ldots, a_{38} \) only depending on the points \( u, v, v_g, v_g', v_g'' \), the two touched gate edges, and the angle \( \theta \). Finally, if \( C(\theta) \) touches the middle gate at its gate edge, then we obtain
\[ f(\varepsilon) = \sqrt{a_0^2 + a_1 \tan \varepsilon + a_2 \tan^2 \varepsilon} - a_6 + \sqrt{a_5^2 + a_6 \tan \varepsilon + a_7 \tan^2 \varepsilon} - a_5 + \sqrt{a_10^2 + a_{11} \tan \varepsilon + a_{12} \tan^2 \varepsilon + a_{13} \tan^3 \varepsilon + a_{14} \tan^4 \varepsilon} - a_{10} + \sqrt{a_{19}^2 + a_{20} \tan \varepsilon + a_{21} \tan^2 \varepsilon + a_{22} \tan^3 \varepsilon + a_{23} \tan^4 \varepsilon} - a_{19}, \]  

for some constants \( a_0, \ldots, a_{27} \) only depending on the points \( u, v, v_g, v_{g'}, v_{g''} \), the three touched gate edges, and the angle \( \theta \). Clearly, combining Equalities (1)–(6) gives us the lemma.

By Lemma 3.2 as long as \( T(\theta) \) maintains the closeness properties in a small neighborhood, \( \theta + \varepsilon \) of \( \theta \), with \( \varepsilon > 0 \), the length function \( \| T(\theta) \| \) is smooth (continuous and differentiable), and we can obtain the angles of minima for \( \| T(\theta) \| \) using standard analytic methods. Since the function consists of \( O(|G(\theta)|) \) terms, requiring us to test \( O(|G(\theta)|) \) potential solutions, we can do this in \( O(|G(\theta)|) \) time.

However, when the closeness properties do not hold, at least one of the following changes occur: the current set of stable vertices of \( T(\theta) \) changes, the current set of gate vertices changes, some gate in the current set \( G(\theta) \) changes its gate edge, or the tour \( T(\theta) \) reaches or leaves an endpoint of a gate. We call the angles where such changes occur events and present them in further detail next.

### 3.1 Events

In general, we have two types of events: those defined by the vertices of the polygon (or pairs of them), and those defined by stable and moving vertices of the current tour. We further subdivide them into the following six types.

**Validity event:** a new gate arises or an old gate disappears. This happens when the gate becomes collinear to a polygon edge adjacent to the gate vertex.

**Domination event:** a gate “changes gate vertex”, i.e., a cut \( c \) issued from a vertex \( v \), previously dominated by a gate \( g \) with gate vertex \( v' \), becomes collinear to \( g \), \( v \) and \( v' \) have the same color, and as the rotation proceeds, \( v \) becomes the new gate vertex.

**Jumping event:** the endpoint of a gate \( g \) on the gate edge, reaches a reflex vertex of \( P \) issuing a cut of different color to that of \( g \).

**Passing event:** the endpoint of a gate on the gate edge, reaches an uncolored reflex vertex or a convex vertex of \( P \).
**Bending event:** a maximal moving subpath of $T(\theta)$ reaches or leaves a reflex vertex of $P'(\theta)$.

**Cuddle event:** a moving vertex of the tour reaches or leaves a gate endpoint.

We have the following lemma.

**Lemma 3.3.** The set of events is complete.

*Proof.* Consider the four properties necessary for two tours $T(\theta)$ and $T(\theta + \varepsilon)$ to be close. We take the contrapositive for each property and show that the only cases when these can occur is if one of the listed events occurs for $T(\theta)$.

1. **The current set of stable vertices of $T(\theta)$ changes.** Since the only part of $T(\theta)$ that changes under $\varepsilon$-rotation are the maximal moving subpaths, a stable vertex can never be directly exchanged for another vertex. Therefore, the only other possibilities are that a stable vertex is either added to or removed from $T(\theta)$, but these are exactly the **bending events**.

2. **The current set of gate vertices changes.** There are three possibilities that this can happen. First, a gate vertex is exchanged for another gate vertex. In order for this to happen, there must be some angle when a gate is collinear to a $\theta$-cut of some other reflex vertex with the same color. These are exactly the **domination events**. The two other possibilities are that a gate is either added to or removed from $G(\theta)$, but these are exactly the **validity events**.

3. **Some gate in the current set $G(\theta)$ changes gate edge.** For this to happen, the endpoint of the gate opposite the gate vertex must lie at a vertex of the polygon. If this vertex is reflex and has the same color as the current gate, we have a **domination event**. If the vertex is reflex but has the opposite color of the gate, by definition, we have a **jumping event**. If the vertex is either reflex but uncolored or convex, then we have exactly a **passing event**. Since vertices can be of no other types, these three event types cover this case.

4. **The tour $T(\theta)$ reaches or leaves an endpoint of a gate.** This is exactly the definition of the **cuddle events**.

Thus, the six event types completely cover all the cases. □

The reason for defining six types of events is that our algorithm will handle each of them slightly differently, as is explained in the next subsection.
3.2 Handling events

The algorithm maintains, for a given angle \( \theta \), the following information: \( T(\theta) \), \( \mathcal{G}(\theta) \), \( H_v(\theta) \) for a reflex vertex \( v \) coinciding with a stable vertex of \( T(\theta) \), the change function \( \| T(\theta + \varepsilon) \| = \| T(\theta) \| + \sum_{1 \leq k \leq |\mathcal{G}(\theta)|} f_k(\varepsilon) \) (a function of \( \varepsilon \)), for each gate in \( \mathcal{G}(\theta) \), the visibility polygon of the gate vertex (for standard straight line visibility, not monotone visibility), and a priority queue \( Q \) maintaining \( O(|\mathcal{G}(\theta)|) \) angles of future potential events, a constant number for each gate. We next present how each event is handled during the running of the algorithm as the direction \( \theta \) rotates from \( 0^\circ \) to \( 180^\circ \).

**ALGORITHM**

Let \( T_{\text{opt}} \leftarrow T(0^\circ) \), \( \theta \leftarrow 0^\circ \), compute the event angles (c.f. Step 1 in the validity event routine below), insert them in \( Q \), and repeat the following steps while \( \theta < 180^\circ \).

Step 1. Get the next event angle \( \theta \) from the priority queue \( Q \).

Step 2. Depending on the event type, perform one of the following routines as described below.

**Validity event routine.** For each such event:

1. Compute the gates \( \mathcal{G}(\theta) \) and the optimal tour \( T(\theta) \) for direction \( \theta \) in \( O(n) \) time as explained in Section 2.1.

2. Empty the priority queue \( Q \).

3. For each segment \( s \) of the shortest path in \( H_v(\theta) \) that crosses a gate, we establish the shortest path from the endpoints of the gate to the endpoints of \( s \), and associate the reflex vertices on those paths that are closest to the endpoints of \( s \) (at most four). We can quickly test (in \( O(|\mathcal{G}(\theta)|) \) total time) whether any such vertex crosses \( s \) during the subsequent rotation and establish at what rotation angle this happens, i.e., the potential next bending and cuddle events. Insert each of the two events per gate in \( Q \).

4. For each of the gates in \( \mathcal{G}(\theta) \), compute the visibility polygon (for standard straight line visibility) for the gate vertex and obtain the next passing event, jumping event, domination event, and validity event (if they exist) by traversing the boundary of the visibility polygon starting from the angle \( \theta \) from the gate vertex. This takes \( O(n) \) time per gate. Insert each of the four events per gate in \( Q \). This takes a total of \( O(n|\mathcal{G}(\theta)|) \) time for all the gates.

5. Look at the next potential event angle \( \theta' \) in \( Q \). Compute the change function \( \| T(\theta + \varepsilon) \| \) and the best local angle \( \theta + \varepsilon \), for \( \varepsilon > 0 \), such that \( \theta < \theta + \varepsilon \leq \theta' \). If \( \| T(\theta + \varepsilon) \| < T_{\text{opt}} \), then update \( T_{\text{opt}} \). This
takes \( O(|\mathcal{G}(\theta)|) \) time, since the change function has \( O(|\mathcal{G}(\theta)|) \) terms, each being the square root of rational polynomials of degree at most 14; see Lemma 3.2.

The time complexity is \( O(n|\mathcal{G}(\theta)|) \) for this case.

**Domination event routine.** For each such event:

1. Update the set of gates in \( \mathcal{G}(\theta) \) by exchanging one gate for a collinear gate with different gate vertex. This takes constant time.

2. Remove the events in \( Q \) associated to the old gate. This takes \( O(|\mathcal{G}(\theta)|) \) time by a traversal of \( Q \).

3. For the new gate vertex, compute the visibility polygon around the gate vertex, and obtain the next passing event, jumping event domination event, and validity event (if they exist) by traversing the boundary of the visibility polygon starting from the angle \( \theta \) from the gate vertex. Insert each of the four events in \( Q \). This takes \( O(n) \) time.

4. Perform Step 5 as for validity events.

The time complexity is \( O(n) \) for this case.

**Jumping event routine.** For each such event:

1. Since the set of gate vertices does not change, only update the gate edge and then recompute the tour obtaining the next jump event by continuing the traversal of the boundary of the visibility polygon starting from the angle \( \theta \) from the gate vertex. Add it to \( Q \). This takes \( O(n) \) time.

2. Perform Step 5 as for validity events.

**Passing event routine.** For each such event:

1. Update the change function \( \|T(\theta + \varepsilon)\| \) with the appropriate term consisting of the square root of rational polynomials of degree at most 14. This takes \( O(|\mathcal{G}(\theta)|) \) time.

2. Perform Step 5 as for validity events.

**Bending event routine.** Here, neither the set of gate vertices nor the set of gate edges change, so we proceed as in Steps 3 and 5 for validity events. The time complexity is \( O(|\mathcal{G}(\theta)|) \).

**Cuddle event routine.** Handled (and detected) as bending events, with the same time complexity.
4 Analysis

The correctness of our algorithm follows directly from Lemmas 3.2 and 3.3. To analyze the time complexity, define \( G \overset{\text{def}}{=} \max_{0^\circ \leq \theta < 180^\circ} |G(\theta)| \). We note that the number of validity events is \( 2r \) and take \( O(nG) \) time each, the number of domination and jumping events is \( O(rG) \) and take \( O(n) \) time each, and the number of passing, bending, and cuddling events is \( O(nr) \) (since they each associate a vertex, either of the tour or of the polygon, with a reflex vertex of the polygon) and take \( O(G) \) time each. Thus, the complexity of our algorithm is \( O(nrG) \).

The storage complexity of our algorithm is \( O(nG) \), since the algorithm maintains at each iteration of the main loop: the tour \( T(\theta) \) having linear size, the hourglass \( H_v(\theta) \) having linear size, the set \( G(\theta) \) having size \( O(G) \in O(n) \), the change function \( ||T(\theta + \varepsilon)|| \) having size \( O(G) \in O(n) \), the priority queue \( Q \) having size \( O(G) \in O(n) \), and finally, at most \( G \) visibility polygons, each of \( O(n) \) size (dominating the storage usage).

It remains to prove that for a fixed angle \( \theta \), we can quickly, in linear time, obtain a constant sized set \( V \) of polygon vertices so that \( S(v, v') \) from \( v \) to its image \( v' \) in \( H_v(\theta) \) corresponds to a shortest watchman tour in \( P \), for some \( v \in V \). If \( P'(\theta) \) has two essential edges (corresponding to gates in \( P \)) with the same color, we know from the proof of Lemma 3.1 that the highest reflex vertex along a path between the gates in a coordinate system where the gates are parallel to the \( x \)-axis must be touched by the tour, otherwise it is not the shortest. Since there are two paths between those gates, we obtain a set \( V \) of two vertices in this case. If \( P'(\theta) \) has one red and one blue essential edge, a shortest tour either touches one of the edge endpoints or, since the gates are parallel, it can be slide along the essential edge until it touches a reflex vertex in the polygon. We obtain such a reflex vertex by computing \( S(p, p') \) for each essential edge endpoint \( p \) and following \( S(p, p') \) to the last vertex before it intersects the first gate in \( H_p(\theta) \). We thus obtain a set \( V \) of at most eight vertices, four essential edge endpoints, and four vertices obtained from \( H_p(\theta) \). We have shown the following theorem.

**Theorem 4.1.** The presented algorithm computes the minimum length shortest (floating) watchman tour under \( \theta \)-monotone visibility over all \( 0^\circ \leq \theta < 180^\circ \) in a simple polygon in \( O(nrG) \) time and \( O(nG) \) storage, where \( n \) is the number of vertices, \( r \) is the number of reflex vertices, and \( G \leq r \) is the maximum number of gates of the polygon used at any time in the algorithm.

5 Conclusions

Observe that our approach can also be used to obtain optimal tours for other parameters which are dependent on the rotation angle \( \theta \), e.g., the longest of all shortest
watchman tours, the one with smallest or largest area, etc. All we need is to adapt Lemma 3.2 for the specific problem.

An interesting extension of our problem is to minimize the longest out of multiple tours that together see a polygon under rotated monotone visibility. For standard visibility the problem is known to be NP-hard even for two tours [19] and efficient constant factor approximation algorithms also exist for this case [20].

6 Acknowledgements

Bengt J. Nilsson was supported by grant 2018-04001 from the Swedish Research Council. David Orden was supported by project 734922-CONNECT from H2020-MSCA-RISE, by project MTM2017-83750-P from the Spanish Ministry of Science (AEI/FEDER, UE), and by project PID2019-104129GB-I00 from the Spanish Ministry of Science and Innovation. Carlos Seara was supported by project 734922-CONNECT from H2020-MSCA-RISE, and by projects MTM2015-63791-R MINECO/FEDER and Gen. Cat. DGR 2017SGR1640. Paweł Żyliński was supported by grant 2015/17/B/ST6/01887 from the Polish National Science Centre.

References

[1] C. ALEGRÍA-GALICIA, D. ORDEN, L. PALIOS, C. SEARA, J. URRUTIA. Capturing points with a rotating polygon (and a 3D extension). Theory Comput. Syst. 63(3):543–566, 2019.

[2] C. ALEGRÍA-GALICIA, D. ORDEN, C. SEARA, J. URRUTIA. Efficient computation of minimum-area rectilinear convex hull under rotation and generalizations. arXiv:1710.10888v2, 2019.

[3] T. ASANO, S.K. GHOSH, T. SHERMER. Chapter 19. Visibility in the plane. In J.R. SACK, J. URRUTIA, editors. Handbook on Computational Geometry. Elsevier Science Publishers, 1999.

[4] S. CARLSSON, H. JONSSON, B.J. NILSSON. Finding the shortest watchman route in a simple polygon. Discrete Comp. Geom. 22:377–402, 1999.

[5] B. CHAZELLE. Triangulating a simple polygon in linear time. Discrete Comp. Geom. 6(3):485–524, 1991.

[6] W. CHIN, S. NTAFOS. Optimum watchman routes. Inf. Proc. Lett. 28:39–44, 1988.

[7] W. CHIN, S. NTAFOS. Shortest watchman routes in simple polygons. Discrete Comp. Geom. 6(1):9–31, 1991.
[8] M. Dror, A. Efrat, A. Lubiw, J.S.B. Mitchell. Touring a sequence of polygons. 35th STOC’03, pages 473–482, 2003.

[9] A. Dumitrescu, C.D. Tóth. Watchman tours for polygons with holes. Comput. Geom. 45(7):326–333, 2012.

[10] L.P. Gewali. Recognizing s-star polygons. Patt. Rec. 28(7):1019–1032, 1995.

[11] L. Guibas, J. Hershberger, D. Leven, M. Sharir, R. Tarjan. Linear time algorithms for visibility and shortest path problems inside triangulated simple polygons. Algorithmica. 2:209–233, 1987.

[12] M. Hammar, B.J. Nilsson. Concerning the time bounds of existing shortest watchman route algorithms. 11th FCT’97, LNCS 1279, pages 210–221, 1997.

[13] Heron of Alexandria. Catoptrica (On Reflection). ~62.

[14] J. Hershberger, S. Suri. A pedestrian approach to ray shooting: shoot a ray, take a walk. Journal of Algorithms. 18(3):403–431, 1995.

[15] G. Huskić, S. Buck, A. Zell. GeRoNa: Generic robot navigation. J. Intell. Robot. Syst. 95(2):419–442, 2019.

[16] C. Icking, R. Klein. Searching for the kernel of a polygon: a competitive strategy. 11th SoCG’95, pages 258–266, 1995.

[17] D.T. Lee, F.P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. Networks. 14:393–410, 1984.

[18] J.S.B. Mitchell. Approximating watchman routes. 24th SODA’13, pages 844–855, 2013.

[19] J.S.B. Mitchell, E.L. Wynters. Watchman routes for multiple guards. 3rd CCCG’91, pages 126–129, 1991.

[20] B.J. Nilsson, E. Packer. An approximation algorithm for the two-watchman route in a simple polygon. 32nd EuroCG’16, pages 111–114, 2016.

[21] D. Orden, L. Palios, C. Seara, J. Urrutia, P. Żyliński. On the width of the monotone-visibility kernel of a simple polygon. 36th EuroCG’20, pages 13:1–13:8, 2020.

[22] D. Orden, L. Palios, C. Seara, P. Żyliński. Generalized kernels of polygons under rotation. 34th EuroCG’18, pages 74:1–74:6, 2018.

[23] L. Palios. An output-sensitive algorithm for computing the s-kernel. 27th CCCG’15, pages 199–204, 2015.
[24] S. Schuierer, G.J.E. Rawlins, D. Wood. A generalization of staircase visibility. 3rd CCCG’91, pages 96–99, 1991.

[25] S. Schuierer, D. Wood. Generalized kernels of polygons with holes. 5th CCCG’93, pages 222–227, 1993.

[26] S. Schuierer, D. Wood. Multiple-guards kernels of simple polygons. Theoretical Computer Science Center Research Report HKUST-TCSC-98-06, 1998.

[27] X.-H. Tan. Fast computation of shortest watchman routes in simple polygons. Inf. Process. Lett. 77(1):27–33, 2001.

[28] X.-H. Tan, T. Hirata. Constructing shortest watchman routes by divide and conquer. 4th ISAAC’93, LNCS 762, pages 68–77, 1993.

[29] X.-H. Tan, T. Hirata, Y. Inagaki. An incremental algorithm for constructing shortest watchman routes. Int. J. Comput. Geom. Appl. 3(4):351–365, 1993.

[30] X.-H. Tan, T. Hirata, Y. Inagaki. Corrigendum to “An incremental Algorithm for constructing shortest watchman routes”. Int. J. Comput. Geom. Appl. 9(3):319–324, 1999.