

# Generalized Watchman Route Problem with Discrete View Cost

Pengpeng Wang\*

Ramesh Krishnamurti †,

Kamal Gupta‡

## Abstract

In this paper, we introduce a generalized version of the Watchman Route Problem (WRP) where the objective is to plan a continuous closed route in a polygon (possibly with holes) and a set of discrete viewpoints on the planned route such that every point on the polygon boundary is visible from at least one viewpoint. The total cost to minimize is a weighted sum of the view cost, proportional to the number of viewpoints, and the travel cost, the total length of the route. We call this problem the Watchman Route Problem with Discrete View Cost or the Generalized Watchman Route Problem (GWRP). In this paper, we consider a restricted version of GWRP that arises naturally in inspection tasks in robotic applications, where each polygon edge is entirely visible from at least one planned viewpoint. We call it *Whole Edge Covering GWRP*. This whole edge covering restriction is not trivial in that WEC-GWRP has the same NP-hardness and inapproximability as GWRP.

The algorithm we propose first constructs a graph that connects  $O(n^{12})$  number of sample viewpoints in the polygon, where  $n$  is the number of polygon vertices; and then solves the corresponding *View Planning Problem with Combined View and Traveling Cost*, using an LP-relaxation based algorithm we introduced in [19]. We show that our algorithm has an approximation ratio in the order of either the view frequency, defined as the maximum number of sample viewpoints that cover a polygon edge, or a polynomial of  $\log n$ , whichever is smaller.

## 1 Introduction

The watchman route problem (WRP) refers to planning a closed curve, called a *watchman route*, in a polygon (possibly with holes), with the shortest distance such that every point on the polygon boundary is visible from at least one point on the route. Here we consider the anchored version where the *start position* is given [5]. Although seemingly very related to two well-known NP-hard problems, namely the *Art Gallery Problem with*

*Point Guards* [13] (*Point AGP*) and the *Euclidean Traveling Salesman Problem* [15] (*Euclidean TSP*), WRP is solvable in polynomial time for simple polygons. It is still NP-hard for polygons with holes [4].

WRP makes impractical assumptions that the watchman senses continuously along the route (taking infinite number of viewpoints) and that the sensing actions do not incur any cost. For instance, in an environment inspection task by a robot-sensor system, each sensing action incurs a large overhead, corresponding to image acquisition, processing, and integration [17]. In addition, often for better sensing qualities, the robot has to stop its movements during image acquisitions. We introduce the problem of generalized watchman route with discrete view cost, or GWRP in short, to relax the continuous sensing assumption of WRP. It refers to planning both a route and a number of discrete viewpoints on it, such that every point on the polygon boundary is visible from at least one planned viewpoint; while the cost is minimized. The cost is a weighted sum of both view cost, proportional to the number of viewpoints planned, and the traveling cost, the total length of the route.

GWRP is not a simple extension to the WRP. First, for cases where traveling cost is negligible, GWRP is reduced to Point AGP. So unlike WRP, which is in **P** for simple polygons, the GWRP is **NP**-hard. Second, as noticed in [10], the optimal WRP solution may incur an unbounded cost for the corresponding GWRP solution, i.e., infinite number of viewpoints are needed on the route to cover the whole polygon boundary. In [3, 9], the authors consider the problem of choosing a set of discrete viewpoints on a given route, while maintaining the same visible polygon boundary. However, it is not always successful and some routes may need infinite number of viewpoints. Their algorithm stops once the approximate viewpoints are too close to each other.

In this paper, we consider a nontrivial restricted version of the GWRP, called the *Whole Edge Covering GWRP* (WEC-GWRP), in which any polygon edge is required to be entirely visible from at least one planned viewpoint. The restriction arises naturally in robot inspection tasks, where the “map” given is often a discretized boundary representation and during inspection tasks each small discretized boundary piece is considered as inspected via one planned viewpoint if and only if all the points on it are visible. Thus, by regarding each piece as a polygon edge, we have a whole edge covering instance. The same restriction is also used in

\*School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada. Email: pwangf@cs.sfu.ca

†School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. Email: ramesh@cs.sfu.ca

‡School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada. Email: kamal@cs.sfu.ca

the terrain guarding problem [6]. WEC-GWRP has the same NP-hardness and inapproximability as GWRP.

Although a natural and nontrivial generalization to both the AGP and the WRP, to the best of our knowledge, there are few related works for the GWRP or WEC-GWRP. In [7, 12], the authors considered a local version of the robot exploration problem, “to look around a corner”, i.e., to detect an object hidden behind a corner while minimizing the sum of the robot traveling distance and the sensor scan time. The problem is considerably simpler since the goal is local, i.e., the objective is not to cover all the object surfaces.

In [20], we considered the problem of view planning with combined view and travel cost (Traveling VPP), which, given a number of discrete viewpoints connected via a graph, asks for a subset of the viewpoints connected by a route such that the boundary edges of a given object are all covered, while minimizing a weighted sum of the number of planned viewpoints and the length of the route. We gave an LP-based rounding algorithm *Round and Connect* that chooses the viewpoints greedily according to their LP optimal solution values, and then solves the Steiner tree problem [18] to connect the chosen viewpoints. We showed that the approximation ratio of *Round and Connect* is in the order of *view frequency*, the maximum number of viewpoints that cover a single boundary edge. We also gave a reduction from Traveling VPP to the Group Steiner Tree problem (GST) instance [8] in polynomial time. By calling the poly-log approximation algorithm for GST [8] after the reduction, we can approximate the optimal solution of Traveling VPP within a poly-log ratio. The result is summarized in the following lemma:

**Lemma 1** *The optimal solution to Traveling VPP can be approximated within the ratio of either the order of view frequency or a poly-log function.*

To use the approximation algorithm for the Traveling VPP, we propose a novel sampling algorithm that computes a bounded number ( $O(n^{12})$ , where  $n$  is the number of polygon vertices) of discrete viewpoints in the polygon, to reduce the GWRP to Traveling VPP. We emphasize here that the number of computed viewpoints does not depend on any geometric parameter of the polygon as opposed to [16] and [1]. We show that if we restrict the problem to choose planned viewpoints only from these sample viewpoints, the cost of the optimal solution to the problem is at most a constant times the cost of the true optimal WEC-GWRP solution. We then construct a Traveling VPP instance using the sample viewpoints and call the approximation algorithm in [20] for a solution. This implies that the cost of the resulting solution is at most the cost of the optimal solution to WEC-GWRP times the smaller of the order of the view frequency and a polynomial of  $\log n$ .

The sampling algorithm works in two steps: first it reduces the viewpoint space from the polygon (2D) to a bounded number of line segments (1D), and then from these line segments (1D) to a bounded number of points. In the first step, we decompose the polygon into *visibility cells*, computed via a partition such that the same polygon edges are entirely visible from all points in each cell. We then restrict the planned viewpoints to be on the visibility cell edges. The reason is as follows. For any feasible WEC-GWRP solution, any other planned viewpoint  $X$  cannot belong to the same visibility cell as  $S$ , and the route connecting  $X$  and  $S$  must cross some edge of the visibility cell that  $X$  belongs to. After replacing  $X$  with the crossing point, we have a feasible WEC-GWRP solution with the same cost and all planned viewpoints are on the visibility cell edges.

Note that if traveling cost is ignored, it suffices to sample one viewpoint arbitrarily on each visibility cell edge. However, due to the view and travel tradeoff, we do not know where on each cell edge the optimal WEC-GWRP solution may choose as the viewpoint. This motivates us to utilize the metric structure in the problem to guide our sampling from 1D to points. We define a local region of each visibility cell edge, called *domain*, and compute a bounded number of viewpoints inside the domains such that the optimal WEC-GWRP solution can be approximated (within a constant ratio) locally using these sample points. For sampling inside each domain, intuitively, we would like to impose an “ordering” on the cell edges, which lets us exploit the weak “metric” between them. This is achieved via dividing domains into *strips* using the visibility cell vertices such that the cell edge ordering remains the same within a strip. We also show the optimal WEC-GWRP solution as a whole can be approximated within a constant ratio once all the local approximations are chained together.

The rest of the paper is organized as follows. First, we give notations and formulate the WEC-GWRP. Second, we detail the two steps of the sampling algorithm. Third, we give the approximation ratio analysis of the proposed algorithm. Last, we discuss potential applications of the proposed sampling algorithm.

## 2 Problem definition

We now formally state the WEC-GWRP. Let  $\mathcal{P}$  denote the given polygon (with or without holes). Let  $\partial\mathcal{P}$  denote its boundary, including the boundary of the holes. Let  $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$  and  $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$  denote the set of polygon vertices and the set of polygon edges, respectively. Let  $\mathcal{A}_r$  denote the set of reflex vertices of  $\mathcal{P}$  (internal angle  $> 180$  degrees). Point  $X_1$  is *visible* from point  $X_2$ , if the closed line segment  $\overline{X_1X_2}$  is contained in  $\mathcal{P}$  (including  $\partial\mathcal{P}$ ). Edge  $e$  is visible from point  $X$ , if every point of  $e$  is visible from  $X$ . Let  $S \in \mathcal{P}$

denote the start position of the watchman. Let  $\mathcal{V}'$  denote a subset of viewpoints, i.e.,  $\mathcal{V}' = \{X : X \in \mathcal{P}\}$  and  $route(\mathcal{V}')$  denote a route connecting the viewpoints in  $\mathcal{V}'$  and  $S$ . Let  $w_v$  and  $w_p$  denote the weights for the view and traveling costs, respectively. Let  $|\mathcal{B}|$  denote the cardinality of a discrete set  $\mathcal{B}$ , and let  $\|\phi\|$  denote the length of route  $\phi$ . The WEC-GWRP is defined as:

$$\begin{aligned} \min \quad & w_v |\mathcal{V}'| + w_p \|route(\mathcal{V}')\| \\ \text{Subject to} \quad & \forall e \in \mathcal{E}, \exists X \in \mathcal{V}' : e \text{ is visible from } X \end{aligned} \quad (1)$$

### 3 Sampling Algorithm

#### 3.1 Visibility cell decomposition

Our decomposition is a “finer” version than that given in [21], i.e., each cell defined here is completely contained in a single cell defined in [21]. This implies that the properties of the cells defined in [21] are preserved here. Similar terminologies (not by exactly the same names) and results can also be found in [2, 11].

The *visibility polygon* of a point  $X \in \mathcal{P}$  is the set of points in  $\mathcal{P}$  that is visible from  $X$ . Its edges are either those contained in  $\partial\mathcal{P}$  or the *constructed edges* incident on reflex vertices. We call these constructed edges the *windows* of point  $X$ . We further extend each window in the direction from  $X$  to the incident reflex vertex until it hits the polygon boundary for the last time, and call it the *extended window*. An extended window is a single line segment that may contain parts outside the polygon  $\mathcal{P}$ . For example, in Fig. 1, the visibility polygon of vertex  $A_1$  consists of a window  $A_5X_1$ , and the corresponding extended window is  $A_5X_3$ . We call the extended windows of the polygon vertices the *critical extended windows*, the number of which is  $O(n^2)$ .

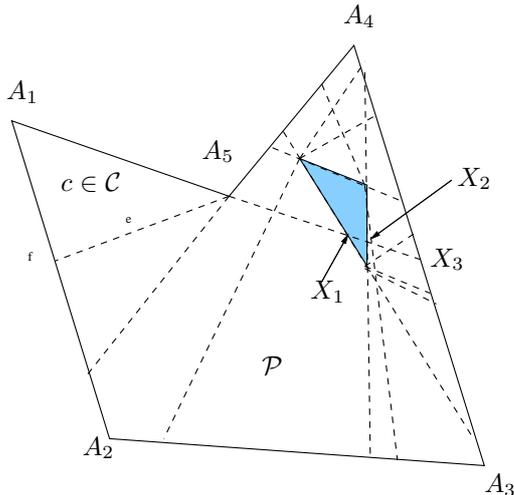


Figure 1: Visibility cell decomposition of polygonal  $\mathcal{P}$ . The shaded region is a hole.

Critical *extended windows* partition the polygon into *visibility cells*. This process is called *visibility cell decomposition*. See Fig. 1. (The decomposition given in [21] uses *critical windows*.) Let  $\mathcal{C}$  denote the set of all visibility cells and  $\mathcal{L}$  denote the set of all visibility cell edges. By the *Zone Theorem* [14], the number of visibility cells,  $|\mathcal{C}|$ , and the number of visibility cell edges,  $|\mathcal{L}|$ , are bounded by  $O(n^4)$ .

Our visibility cell decomposition preserves the following property.

**Lemma 2** *All points in the same visibility cell have the same polygon edges entirely visible from them.*

#### 3.2 Sampling visibility cell edge domain

For a visibility cell edge  $l$ , as shown in Fig. 2, we draw a diamond shape consisting of two isosceles triangles with  $l$  as the common base. The sides of each triangle form an angle of  $\alpha < 90$  degrees with the base. We will subsequently show how to determine  $\alpha$  in Section 4. We define the domain of the cell edge, denoted by  $Dom(l)$ , as the set of all points of polygon  $\mathcal{P}$  inside the diamond (including the diamond boundary edges). In Fig. 2,  $Dom(l)$  is the set of points in the diamond shape excluding the shaded area.

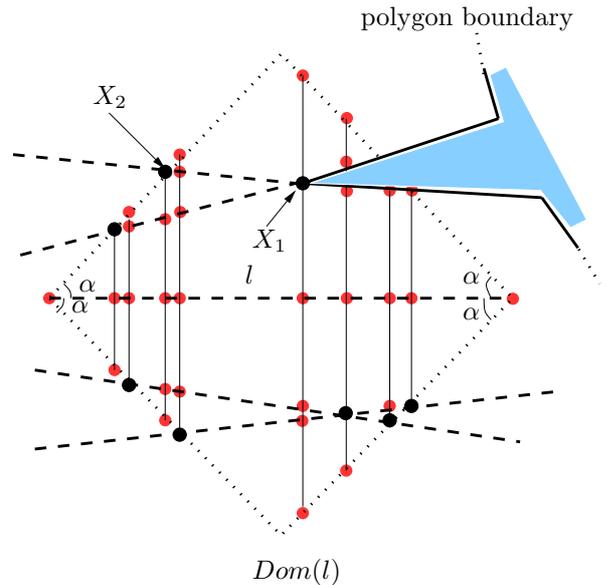


Figure 2: Illustration of the sampling algorithm. See text for details.

Lemma 3 states a simple, but crucial observation:

**Lemma 3** *For a visibility segment  $l$ , the slope (w.r.t.  $l$ ) of any other visibility segment  $l'$  that intersects  $Dom(l)$  is less than  $\alpha$ , i.e.,*

$$\forall l' : l' \cap Dom(l) \neq \emptyset, |\angle l'l| \leq \alpha$$

**Proof.** Otherwise, the extended critical window collinear with  $l'$  intersects  $l$  and splits it into two edges. This contradicts that  $l$  is a single visibility segment.  $\square$

See Fig. 2. Inside each visibility cell edge domain  $Dom(l)$ , we draw orthogonal (w.r.t.  $l$ ) lines from all the vertices of visibility cells. The segments of these vertical lines contained in  $Dom(l)$ , the other visibility cell edges, the polygon boundaries, and the boundaries of  $Dom(l)$  intersect each other. We call these intersection points *sample points* and denote the set of sample points for all domains by  $\Gamma$ . The number of sample points in each domain is the number of vertices in the arrangements of the line segments described above, and is bounded by  $(|\mathcal{L}| + |\mathcal{L}| + n + 4)^2 = O(n^8)$ , according to Zone Theorem [14]. (The terms in the brackets are the bounds on the number of vertical line segments in each domain (bounded by the number of visibility cell vertices), the number of other visibility cell edges, the number of polygon edges, the number of domain boundaries, respectively.) Thus,  $\Gamma$  is bounded:  $|\Gamma| \leq |\mathcal{L}| \cdot O(n^8) = O(n^{12})$ .

We construct the complete graph  $\mathcal{G}$  on  $\Gamma$  where the edge cost between two sample points is the shortest path distance between them in  $\mathcal{P}$ . This is done by constructing first the *visibility graph* of  $\Gamma$ ; and then the shortest path graph on the visibility graph. Now we have an *induced Traveling VPP* instance, with the set of viewpoints and traveling graph being  $\Gamma$  and  $\mathcal{G}$  respectively.

## 4 Sampling Algorithm Analysis

In this section, we show that the cost of the optimal solution to the induced Traveling VPP is at most a constant times that of the optimal solution to WEC-GWRP.

The idea is as follows. Assuming we have the optimal solution to the WEC-GWRP, we will find a solution to the induced Traveling VPP by first traversing the optimal route and partitioning it into pieces, then replacing each piece with a route passing through sample viewpoints while keeping the end-points of the piece fixed, and then moving the end-points to sample viewpoints. The partition process guarantees that the visibility segments that each piece passes through are ordered. The slope lemma, Lemma 3, then helps bound the length of the replacing piece w.r.t. that of the original piece on the optimal route.

For the optimal WEC-GWRP solution, let  $\mathcal{V}^*$  denote the set of planned viewpoints, and let  $\phi^*$  denote the shortest route connecting  $\mathcal{V}^*$ . As discussed in the Introduction, we can assume all planned viewpoints are on the visibility segments. We denote the subset of visibility segments where the planned viewpoints are located by  $\mathcal{L}^*$ , i.e.,  $\mathcal{L}^* = \{l \in \mathcal{L} : \exists X \in \mathcal{V}^*, X \in l\}$ .

First, we give an observation of the route planned in the optimal solution,  $\phi^*$ , Lemma 4.

**Lemma 4** *The planned optimal route  $\phi^*$  consists of a sequence of consecutive line segments. The end-points of these segments are either planned viewpoints,  $\mathcal{V}^*$ , or reflex vertices,  $\mathcal{A}_r$ .*

**Proof.** Note that  $\phi^*$  is the shortest route connecting  $\mathcal{V}^*$ . The part of  $\phi^*$  between any two consecutive planned viewpoints must be the shortest path in  $\mathcal{P}$  connecting them, hence consists of a sequence of consecutive line segments, whose end-points are either the reflex vertices or these two points. Thus,  $\phi^*$  as a whole has the property stated in the lemma.  $\square$

### 4.1 Partition of $\phi^*$

In the following, we traverse  $\phi^*$  and partition it in two steps. First, we partition it according to some visibility segment domains. Then, inside each domain, we further partition it using the vertical line segments incident on the cell vertices.

#### 4.1.1 Partition according to domains

To help define the partition, we first introduce a labeling of some (not all) edges in  $\mathcal{L}^*$ . Note that all the edges in  $\mathcal{L}^*$  are naturally ordered by the way  $\phi^*$  traverses and intersects them. We start with the first visibility segment and label it  $l_1$ . Skip the following visibility segments whose corresponding planned viewpoints ( $\mathcal{V}^*$ ) belong to the currently labeled domain,  $Dom(l_1)$ . We then label the next visibility segment that  $\phi^*$  passes through after it exits  $Dom(l_1)$  by  $l_2$ , and continue in this fashion. We partition  $\phi^*$  according to the labeled visibility segments. Let  $\phi_k^*$ ,  $k = 1, \dots, K$  denote these parts. Thus, the start- and end-points of  $\phi_k^*$  are  $S_k$  and  $S_{k+1}$ , respectively. By definition,  $S_k \in l_k \subset Dom(l_k)$  and  $S_{k+1} \notin Dom(l_k)$  for  $k \geq 1$ . Please see Fig. 3 for an example.

#### 4.1.2 Partition inside a domain

See Fig. 4. The vertical line segments in a domain partition it into vertical *strips*. The strips are bounded by three types of boundaries: the polygon boundary; the *strip boundary* that separates two neighboring strips; and the *domain boundary*, the boundary of the diamond shape defining a domain.

Note that all the visibility segments are ordered inside a strip, since otherwise a vertical line at the intersection point where the order changes would have divided the strip into smaller strips. We order the visibility segments in a strip according to their intersection point

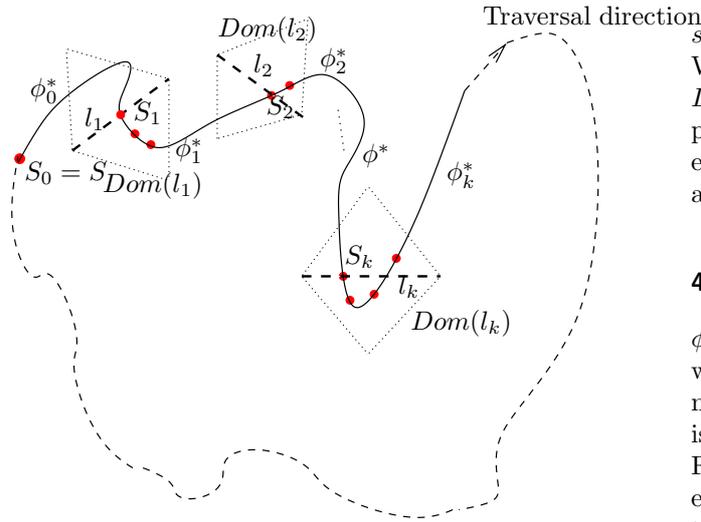


Figure 3: Partition of the optimal WEC-GWRP route  $\phi^*$  according to the visibility segments and corresponding domains it crosses. Note that although shown disjoint, the labeled domains may intersect each other and planned viewpoints on  $\phi_k^*(st)$  may be contained in domains corresponding to labeled visibility segments  $l_1, l_2, \dots, l_{k-2}$ .

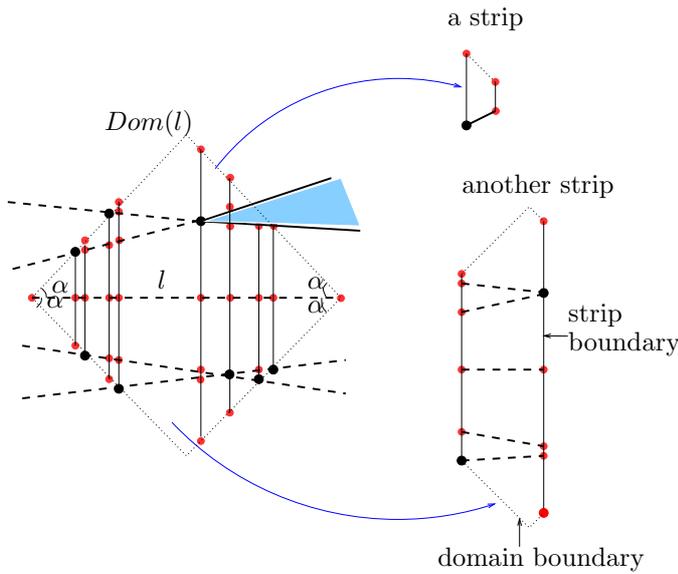


Figure 4: The vertical line segments from visibility cell vertices partition a domain into strips, bounded by domain boundary and/or polygon and strip boundary.

with the left strip or domain boundary. The position relation between visibility segments, *above/below*, is thus well defined inside a strip.

We further partition each part of the optimal route,  $\phi_k^*$ , inside the domain  $Dom(l_k)$  by its strips. For a strip

$st$ , we denote the partitioned piece by  $\phi_k^*(st) = \phi_k^* \cap st$ . We denote the start-point and end-point of  $\phi_k^*(st)$  by  $L_{st}$  and  $R_{st}$  respectively, also called the entry and exit points respectively in the following. We denote the highest and lowest visibility segments that  $\phi_k^*(st)$  crosses (if applicable) by  $l_{high}$  and  $l_{low}$  respectively.

#### 4.1.3 Categories and properties of $\phi_k^*(st)$

$\phi_k^*(st)$  can be categorized into five cases, according to whether its entry point is on the labeled visibility segment or on the strip boundary and whether its exit point is on the strip boundary or on the domain boundary. See Fig. 5 for illustrations. For cases (Ia) and (Ib),  $\phi_k^*(st)$  enters on  $l_k$  (at point  $S_k$ , i.e.,  $L_{st} = S_k$ ), and exits either through the strip boundary, Case (Ia), or through the domain boundary, Case (Ib). For cases (IIa), (IIb) and (IIc),  $\phi_k^*(st)$  enters  $st$  through a strip boundary and exits through either the other strip boundary, Case (IIa), or the same strip boundary, Case (IIb), or the domain boundary, Case (IIc).

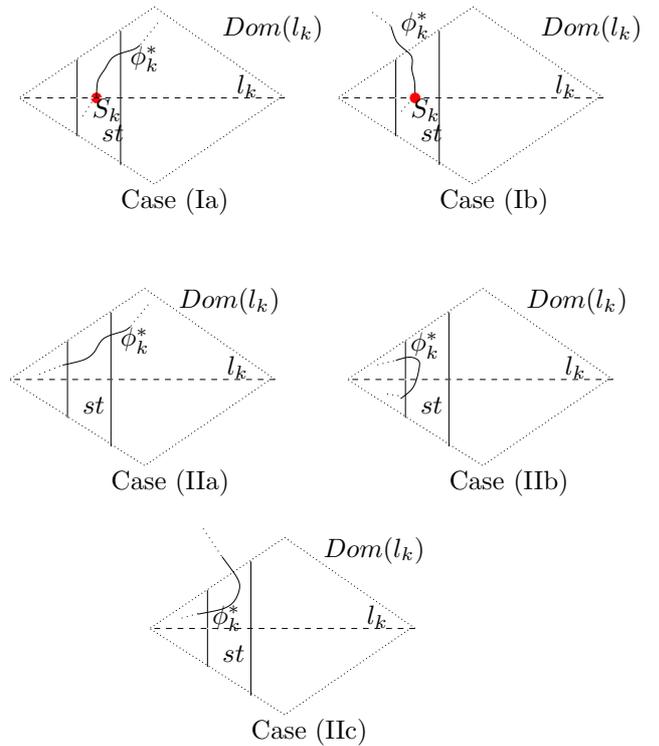


Figure 5: Five cases of  $\phi_k^*(st)$ . Case (Ia):  $\phi_k^*(st)$  starts at  $S_k$  and exits from the strip boundary. Case (Ib):  $\phi_k^*(st)$  starts at  $S_k$  and exits through the domain boundary. Case (IIa):  $\phi_k^*(st)$  enters and exits through different strip boundaries. Case (IIb):  $\phi_k^*(st)$  enters and exits through the same strip boundary. Case (IIc):  $\phi_k^*(st)$  enters through the strip boundary and exits through the domain boundary.

We now show that  $\phi_k^*(st)$  consists of at most three straight line segments:

**Lemma 5**  $\phi_k^*(st)$  consists of at most three straight line segments. If  $\phi_k^*(st)$  consists of three segments, the exit point must lie between the highest and lowest visibility segments  $\phi_k^*(st)$  touches,  $l_{high}$  and  $l_{low}$ .

**Proof.** The proof is by contradiction. If the above mentioned condition is not satisfied, we can replace  $\phi_k^*(st)$  by a strictly shorter route in  $st$  with the same entry and exit points, which crosses the same visibility segments as  $\phi_k^*(st)$ . This implies that  $\phi^*$  cannot be optimal, because we can replace  $\phi_k^*(st)$  by this shorter route and choose viewpoints on the visibility segments that are collinear with the viewpoints taken on  $\phi_k^*(st)$  and hence reduce the solution cost.

Wlog, suppose  $\phi_k^*(st)$  goes upwards first after it enters  $st$  at point  $L_{st}$ . When  $\phi_k^*(st)$  changes directions, it has to go downwards. Otherwise, we can simply replace the two upward sloping segments by a single line segment and the route is shorter. Since this single line segment passes all the visibility segments the two segments pass, it has exactly the same set of entirely visible polygon edges. Similarly,  $\phi_k^*(st)$  cannot change direction when going downwards. We can also easily show that if afterwards  $\phi_k^*(st)$  goes upwards again, it has to exit  $st$ . Otherwise, we can replace it with a shorter route.

If  $\phi_k^*(st)$  consists of three segments, and it exits the strip by intersecting the highest (resp. lowest) visibility segment, we can replace the first (resp. last) two segments by a single segment and reduce solution cost. So the exit point must lie between  $l_{high}$  and  $l_{low}$ .  $\square$

## 4.2 Approximating $\phi_k^*(st)$ using a route connecting sample viewpoints

We can show, case by case, how to approximate  $\phi_k^*(st)$  using a route, denoted by  $\phi'_k(st)$ , which connects sample viewpoints in  $\Gamma$  and the start- and end-points of  $\phi_k^*(st)$ . To illustrate the idea, in the following, we detail the proof for Case (Ia) and summarize the other cases.

### 4.2.1 Approximation for Case (Ia)

This is the case where a smaller  $\alpha$  gives us a better approximation ratio.  $\phi_k^*(st)$  consists of two segments, Fig. 6(a), or three segments, Fig. 6(b). The route  $\phi'_k(st)$  consists of the straight line segment between  $S_k$  and  $R_{st}$  and one or two detours on the strip boundary from  $R_{st}$ .

We now show that the alternative route can approximate the corresponding part of the optimal WEC-GWRP solution within a constant ratio (for a given  $\alpha$ ).

**Lemma 6** For case (Ia),  $\phi'_k(st)$  can approximate  $\phi_k^*(st)$  within the ratio of  $1 + \frac{4}{\cos \alpha}$ .

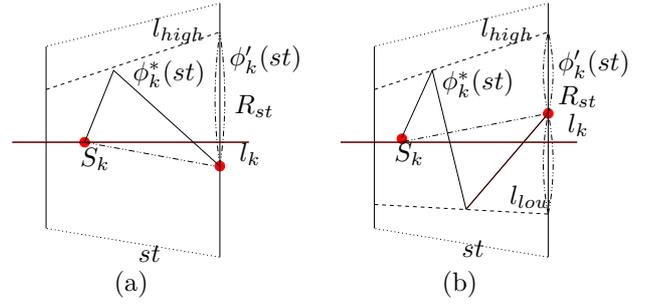


Figure 6: Approximate  $\phi_k^*(st)$  for case (Ia). (a)  $\phi_k^*(st)$  consists of two segments to reach  $l_{high}$ ;  $\phi'_k(st)$  consists of the line segment  $\overline{S_k R_{st}}$  and a detour going upwards to  $l_{high}$  and back to  $R_{st}$ . (b)  $\phi_k^*(st)$  consists of three segments to reach both  $l_{high}$  and  $l_{low}$ ;  $\phi'_k(st)$  consists of  $\overline{S_k R_{st}}$  and two detours, one going upwards to  $l_{high}$  and back, and the other going downwards to  $l_{low}$  and back.

**Proof.** In the following, we show for the subcases when  $\phi_k^*(st)$  consists of two segments and when it consists of three segments, respectively. The one-segment case is true trivially.

See Fig. 7.  $\phi_k^*(st)$  consists of two segments,  $a$  and  $b$ ;  $\phi'_k(st)$  consists of the segment  $c$  and a detour consisting of segments  $d_1$  and  $d_2$ .  $\beta_1$  corresponds to the slope of  $l_{high}$  and is measured counterclockwise from  $d$ , thus bounded by  $\alpha$ , i.e.,  $-\alpha \leq \beta_1 \leq \alpha < \pi/2$ .  $\beta_2$  is measured clockwise from  $d$ . So its range is between  $-\alpha$  and  $\pi/2$ , i.e.,  $-\pi/2 \leq -\alpha \leq \beta_2 \leq \alpha < \pi/2$ .

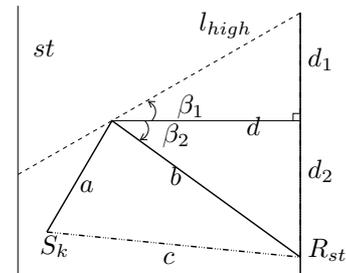


Figure 7: Approximate  $\phi_k^*(st)$  for case (Ia) when  $\phi_k^*(st)$  consists of two segments.

We have the following relations:

$$\begin{aligned} \|c\| &\leq \|a\| + \|b\| \\ d_1 &= \|d\| \cdot \tan \beta_1 = \|b\| \cos \beta_2 \tan \beta_1 \\ d_2 &= \|b\| \sin \beta_2 \end{aligned}$$

Since  $-\alpha \leq \beta_1 \leq \alpha < \pi/2$ ,  $\cos \beta_1 \neq 0$ , we have,

$$\begin{aligned} \|d_1 + d_2\| &= \|b\| \cdot |\cos \beta_2 \tan \beta_1 + \sin \beta_2| \\ &= \frac{\|b\|}{\cos \beta_1} |\sin(\beta_1 + \beta_2)| \\ &\leq \frac{\|b\|}{\cos \beta_1} \leq \frac{\|a\| + \|b\|}{\cos \beta_1} \leq \frac{\|a\| + \|b\|}{\cos \alpha}. \end{aligned}$$

So we have,

$$\begin{aligned} \|\phi'_k(st)\| &= \|c\| + 2(\|d_1\| + \|d_2\|) \\ &\leq (\|a\| + \|b\|)(1 + \frac{2}{\cos \alpha}) \\ &= \|\phi_k^*(st)\|(1 + \frac{2}{\cos \alpha}), \end{aligned}$$

$$\text{i.e., } \frac{\|\phi'_k(st)\|}{\|\phi_k^*(st)\|} \leq (1 + \frac{2}{\cos \alpha}).$$

The case for the equality occurs for condition as shown in Fig. 8.

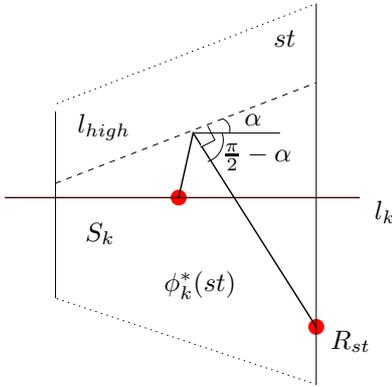


Figure 8: The worst case of the approximation of case (Ia).  $S_k$  is arbitrarily close to  $l_{high}$ ;  $l_{high}$  is parallel to the domain boundary, i.e. its slope is  $\alpha$ ; and the line segment connecting  $R_{st}$  has slope  $\frac{\pi}{2} - \alpha$ , i.e. it is perpendicular to  $l_{high}$ .

For the second sub-case where  $\phi_k^*(st)$  consists of three segments,  $\phi'_k(st)$  includes the segment  $\overline{S_k R_{st}}$  and detours at  $R_{st}$  on the strip boundary to reach (and then come back)  $l_{high}$  and  $l_{low}$  respectively. In the following, we analyze the approximation ratio. We use an intermediate tour, a path consisting of  $\overline{S_k R_{st}}$ , a detour at  $S_k$  to reach  $l_{high}$  and back, and a detour at  $R_{st}$  to reach  $l_{low}$  and back. We first show the approximation ratio of this tour to  $\phi_k^*(st)$ , and then the approximation ratio of  $\phi'_k(st)$  to this intermediate tour.

Let  $M$  denote the intersection point between the straight line segment  $\overline{S_k R_{st}}$  and  $\phi_k^*(st)$ . We divide  $\phi_k^*(st)$  into two parts, the part from  $S_k$  to  $M$  and the part from  $M$  to  $R_{st}$ . Note that each part consists of two line segments. We can approximate these two parts by the corresponding parts of the intermediate tour shown

in Fig. 9(b1) and (b2), respectively. It consists of a straight line segment between  $S_k$  and  $M$  and a vertical detour at  $S_k$  to reach  $l_{high}$  and back, Fig. 9(b1), followed by a straight line segment between  $M$  and  $R_{st}$  and a vertical detour at  $R_{st}$  to reach  $l_{low}$  and back, Fig. 9(b2). From the above analysis for the two-segment case, we know the approximations are within  $(1 + \frac{2}{\cos \alpha})$ -ratio for both, thus the intermediate tour can approximate  $\phi_k^*(st)$  within the same ratio,  $(1 + \frac{2}{\cos \alpha})$ .

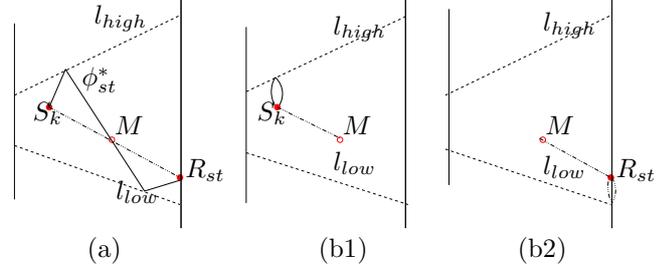


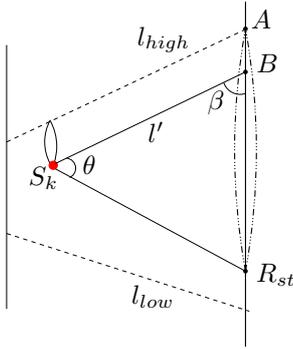
Figure 9: Approximate  $\phi_k^*(st)$  for case (Ia) when  $\phi_k^*(st)$  consists of three segments using an intermediate tour. The two parts of this intermediate tour are shown in (b1) and (b2) respectively.

Note the only difference of the intermediate tour from  $\phi'_k(st)$  is that its detour at  $S_k$ , the solid loop shown in Fig. 9(b1), is not at the exit point. See Fig. 10. We now replace the detour at  $S_k$ , the solid loop, with a detour at  $R_{st}$  to reach  $l_{high}$  at point  $A$  and back, the dashed-dotted loop. From  $S_k$ , we draw a line segment  $l'$  parallel to  $l_{high}$  and it intersects the right strip boundary at point  $B$ . The part of the dashed-dotted loop between points  $A$  and  $B$  has the same length as the solid loop. In the triangle  $\triangle S_k B R_{st}$ , let  $\theta$  denote the angle between  $\overline{S_k B}$  and  $\overline{S_k R_{st}}$  and  $\beta$  denote the angle between  $\overline{S_k B}$  and  $\overline{B R_{st}}$ . We have  $0 \leq \theta < \alpha + \pi/2$  and  $\pi/2 - \alpha \leq \beta < \pi/2$ . By the law of sines,  $\|\overline{B R_{st}}\| = \frac{|\sin \theta|}{|\sin \beta|} \|\overline{S_k R_{st}}\| \leq \frac{\|\overline{S_k R_{st}}\|}{\sin \beta} \leq \frac{1}{\cos \alpha} \|\phi_k^*(st)\|$ . The last inequality is due to the slope lemma, Lemma 3, and the fact that  $\|\overline{S_k R_{st}}\|$  is at most  $\|\phi_k^*(st)\|$ . Thus,  $\|\phi'_k(st)\|$  is at most the length of the intermediate tour, which is at most  $(1 + \frac{2}{\cos \alpha}) \|\phi_k^*(st)\|$ , plus the length of the part of the dashed-dotted loop between  $B$  and  $R_{st}$ , which is at most  $\frac{2}{\cos \alpha} \|\phi_k^*(st)\|$ . Consequently,  $\|\phi'_k(st)\| \leq (1 + \frac{4}{\cos \alpha}) \|\phi_k^*(st)\|$ .

To summarize, we have that  $\|\phi'_k(st)\|$  is at most  $\|\phi_k^*(st)\|$  times  $1 + \frac{4}{\cos \alpha}$ .  $\square$

#### 4.2.2 Approximation for Case (Ib)

This is the case where bigger  $\alpha$  gives us better approximation ratio. Wlog, here we only consider the case where  $\phi_k^*(st)$  exits from the top left domain boundary. First, by Lemma 5, if  $\phi_k^*(st)$  has three segments, the


 Figure 10: Approximate first part of  $\phi_k^*(st)$ .

exit point must lie between the highest and the lowest visibility cell edges  $\phi_k^*(st)$  touches. Since we know the exit point is on the domain boundary, higher than  $l_{high}$ ,  $\phi_k^*(st)$  consists of at most two segments. Here we illustrate only the single segment sub-case.

See Fig. 11. We approximate  $\phi_k^*(st)$  using the straight line segment connecting  $S_k$  and the top-left strip boundary vertex,  $V_{st}$ , and a detour on the strip boundary, if necessary, from  $V_{st}$  to reach  $l_{low}$  and back to  $V_{st}$ . In Lemma 7, we show the approximation ratio using the left domain vertex on strip  $st$ . For lack of space, the proof of Lemma 7 is skipped. Intuitively, since the angle  $\theta$  is at least  $\alpha$ , the length of  $route(S_k V_{st} S_{k+1})$  is at most a constant times that of  $\overline{S_k S_{k+1}}$ .

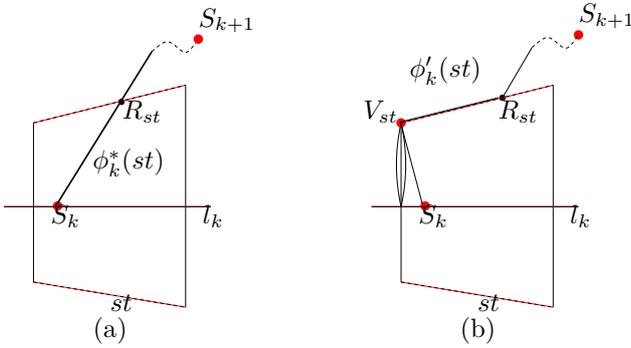


Figure 11: Approximate  $\phi_k^*(st)$  for case (Ib): one-segment sub-case. (a)  $\phi_k^*(st)$  consists of a line segment  $\overline{S_k S_{k+1}}$ . (b)  $\phi_k^*(st)$  consists of two segments  $\overline{S_k V_{st}}$  and  $\overline{V_{st} S_{k+1}}$  and a detour from  $V_{st}$  to reach  $l$  and back.

**Lemma 7** For case (Ib),  $\phi_k^*(st)$  can approximate  $\phi_k^*(st)$  within the constant ratio of  $\frac{1}{\sin \frac{\alpha}{2}} + \frac{2}{\cos \alpha}$ .

Similarly, for cases (IIa), (IIb), (IIc), we can find a  $\phi_k^*(st)$  whose length is at most that of  $\phi_k^*(st)$  times  $1 + \frac{4}{\cos \alpha}$  or  $\frac{1}{\sin \frac{\alpha}{2}} + \frac{2}{\cos \alpha}$ .

### 4.3 Chaining together the partitions

By its construction,  $\phi_k^*(st)$ 's chained together form a continuous route, a sequence of line segments with loops at their endpoints, Fig. 12. However, the start- and end-points of  $\phi_k^*(st)$  may not belong to the sample point set  $\Gamma$ . We construct a solution  $\hat{\phi}$  from  $\phi'$  as follows. For endpoints inside their loops, e.g. point  $V$  in Fig. 12, we move them to the nearest sample point *inside* the loops. By the triangular inequality, we loose at most an additional constant (2) approximation ratio. For  $S_k$  and endpoints without detours, e.g. point  $W$  and  $S_k$  in Fig. 12, we can simply ignore it, since it is taken care of by the endpoints of its previous and next strips. For example, in Fig. 12, let  $U$  and  $V$  denote the planned viewpoints that are immediately before and immediate after  $S_k$  on  $\phi'$  respectively. We can bypass  $S_k$  and connect directly  $U$  and  $V$ , (the dotted line segment).

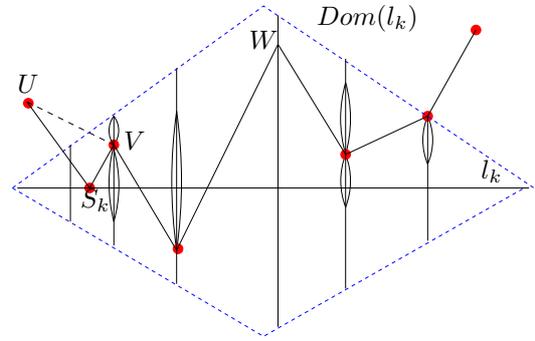


Figure 12: The approximation solution within a domain  $Dom(l_k)$ , after chaining together  $\phi_k^*(st)$  for the strips  $\phi^*$  crosses.

It is clear that  $\hat{\phi}$  is a solution to the induced Traveling VPP, i.e., all the planned viewpoints are from the sample point set. Using this solution, we can bound the cost of the optimal solution to the induced Traveling VPP w.r.t. the true optimal to the WEC-GWRP.

**Theorem 8** The cost of the optimal solution to the induced Traveling VPP is at most 11.657 times that of the optimal solution to the WEC-GWRP.

**Proof.** By summarizing cases (Ia), (Ib), (IIa), (IIb) and (IIc), the length of  $\phi_k^*(st)$  is at most that of  $\phi_k^*(st)$  times  $\max(1 + \frac{4}{\cos \alpha}, \frac{1}{\sin \frac{\alpha}{2}} + \frac{2}{\cos \alpha})$ . After chaining and moving the entry and exit points, the length of  $\hat{\phi}$  is at most twice that of  $\phi_k^*(st)$ . Since  $\hat{\phi}$  is a feasible solution to the induced Traveling VPP, its cost is a lower bound on the optimal solution to the induced Traveling VPP. Thus, the cost of the optimal solution to the induced Traveling VPP is at most  $2 \max(1 + \frac{4}{\cos \alpha}, \frac{1}{\sin \frac{\alpha}{2}} + \frac{2}{\cos \alpha})$  times the cost of the true optimal solution to the WEC-GWRP.

Now we choose the  $\alpha$  value to minimize this ratio, i.e., to solve  $\min_{\alpha} 2 \max(1 + \frac{4}{\cos \alpha}, \frac{1}{\sin \frac{\alpha}{2}} + \frac{2}{\cos \alpha})$ . And the solution is approximately 11.657, when  $\alpha \approx 34^\circ$  (via numerical minimization using Matlab).  $\square$

After constructing the induced Traveling VPP using the sample points, we call the approximation algorithm in [19] to get a solution. The approximation ratio result of this Traveling VPP solver [19] and Theorem 8 implies that the cost of the solution is at most the cost of the optimal WEC-GWRP solution times either the order of the view frequency or a polynomial of  $\log n$ , whichever is smaller.

## 5 Conclusion

In this paper, we introduce the generalized watchman route problem of planning a continuous route and a set of discrete viewpoints on the route such that every polygon edge is entirely visible from at least one planned viewpoint, while minimizing a (weighted) sum of view cost and travel cost. The problem is NP-hard and log-inapproximable. We propose an approximation algorithm that consists of two steps. The first step is a novel sampling method that, for any given WEC-GWRP instance, produces a polynomial number of discrete viewpoints in the polygon. We then restrict the planned viewpoints to be from these sample points and construct a corresponding Traveling VPP instance. The second step uses the approximation algorithm in [19] to solve the constructed Traveling VPP instance. We show that the optimal solution cost to the Traveling VPP instance is at most a constant times the optimal solution cost of the WEC-GWRP instance. As a result, the approximation ratio of our algorithm is in the order of the view frequency or a poly-log function of the input size, whichever is smaller.

We believe that the sampling algorithm proposed here is a general technique and can also be used for other shortest route problems where one would like to get an approximation algorithm by first reducing the infinite input space to a discrete sample point set, and then solving the resulting discrete problem. For example, the algorithm can be applied to a generalized version of the 2.5D terrain guarding problem [6] with additional travel cost in the objective function. We can then first apply the cell decomposition to reduce the input space to a set of line segments, the cell edges, (same decomposition was used in [6]), then use the sampling algorithm proposed in this paper to reduce it to a Traveling VPP instance, and then call the Traveling VPP solver. The resulting algorithm has an approximation ratio of the order of the view frequency or a poly-log function of the input size, whichever is smaller.

## References

- [1] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Determining approximate shortest paths on weighted polyhedral surfaces. *Journal of the ACM*, 52(1):25–53, January 2005.
- [2] P. Bose, A. Lubiw, and J. Munro. Efficient visibility queries in simple polygons. In *Proc. CCCG*, pages 23–28, 1992.
- [3] S. Carlsson and B. Nilsson. Computing vision points in polygons. *Algorithmica*, 24(1):50–75, 1999.
- [4] W. Chin and S. Ntafos. Optimum watchman routes. *Information Processing Letters*, 28:39–44, 1988.
- [5] W. Chin and S. Ntafos. Watchman routes in simple polygons. *Discrete and Computational Geometry*, 6(1):9–31, 1991.
- [6] S. Eidenbenz. Approximation algorithms for terrain guarding. *Information Processing Letters*, 82:99–105, 2002.
- [7] S. Fekete, R. Klein, and A. Nuchter. Online searching with an autonomous robot. In *Proc. Workshop on Algorithmic Foundation of Robotics*, pages 350–365, 2004.
- [8] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37:66–84, 2000.
- [9] S. Ghosh and J. Burdick. Understanding discrete visibility and related approximation algorithms. In *Proc. CCCG*, pages 106–111, 1997.
- [10] S. Ghosh and J. Burdick. Exploring an unknown polygonal environment with discrete visibility. Unpublished Manuscript, Tata Institute of Fundamental Research, Mumbai, India, 2004.
- [11] L. Guibas, R. Motwani, and P. Raghavan. The robot localization problem. *SIAM Journal on Computing*, 26(4):1120–1138, 1997.
- [12] V. Isler, S. Kannan, and K. Daniilidis. Local exploration: online algorithms and a probabilistic framework. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1913 – 1920, 2003.
- [13] D. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32:276–282, 1986.
- [14] J. O’Rourke. *Computational Geometry in C*. Cambridge University Press, second edition, 1998.

- [15] C. Papadimitriou. Euclidean TSP is NP-complete. *Theoretical Computer Science*, 4:237–244, 1977.
- [16] C. Papadimitriou. An algorithm for shortest-path motion three dimensions. *Information Processing Letters*, 20:259–263, 1985.
- [17] W. Scott, G. Roth, and J. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys*, 35(1):64–96, March 2003.
- [18] V. Vazirani. *Approximation algorithms*. Springer, 2001.
- [19] P. Wang, R. Krishnamurti, and K. Gupta. View planning problem with combined view and traveling costs: Problem formulation, hardness of approximation, and approximation algorithms. Technical Report TR2006-17, Simon Fraser University, Burnaby, B.C., Canada, May 2006. Available at <ftp://fas.sfu.ca/pub/cs/TR/2006/CMPT2006-17.ps>. A version to appear in *Proc. 2007 IEEE International Conference on Robotics and Automation*.
- [20] P. Wang, R. Krishnamurti, and K. Gupta. View planning problem with combined view and traveling cost. In *Proc. IEEE ICRA*, pages 711 – 716, 2007.
- [21] A. Zarei and M. Ghodsi. Efficient computation of query point visibility in polygons with holes. In *Proc. SOCG*, pages 314–320, 2005.