Mapping a polygon with holes using a compass

Yann Disser Subir Kumar Ghosh Matúš Mihalák Peter Widmayer

July 2, 2012

Abstract

We consider a simple robot inside a polygon $\mathcal{P}$ with holes. The robot can move between vertices of $\mathcal{P}$ along lines of sight. When sitting at a vertex, the robot observes the vertices visible from its current location, and it can use a compass to measure the angle of the boundary of $\mathcal{P}$ towards north. The robot initially only knows an upper bound $\bar{n}$ on the total number of vertices of $\mathcal{P}$. We study the mapping problem in which the robot needs to infer the visibility graph G_{vis} of $\mathcal{P}$ and needs to localize itself within G_{vis}. We show that the robot can always solve this mapping problem. To do this, we show that the minimum base graph of G_{vis} is identical to G_{vis} itself. This proves that the robot can solve the mapping problem, since knowing an upper bound on the number of vertices was previously shown to suffice for computing G_{vis}.

1 Introduction

The *mapping problem* and the *localization problem* are fundamental for many tasks in robotics and autonomous exploration. In the mapping problem, a robot is required to obtain a (rough) map of an initially unknown environment, while the localization problem requires the robot to identify its current position on the map. Both problems often arise together and need to be solved simultaneously. In this paper, we use the term *mapping problem* loosely to refer to the combination of both tasks.

The difficulty of mapping depends on the type of the environment as well as on the capabilities of the robot. Many variations in scenario, robot models, and questions have been studied in this context. We are interested in the following question: *What are minimal capabilities that a robot needs in order to solve the mapping problem?*

We study the mapping problem in polygonal environments. In particular, and in contrast to past work, we allow polygonal obstacles (or, equivalently, holes) in the environment. Our robot model is based on a minimalistic framework: Our *basic robot* can move from vertex to vertex along lines of sight, and, being at a vertex, the robot can observe other visible vertices in counterclockwise order. Other than that, the robot has no direct means of distinguishing vertices according to global identifiers or names, i.e., it can not even tell whether it has visited its current location before. Figure 1 illustrates the capabilities of the basic robot in a polygon with holes (for a formal definition, see Section 1.2). Using this model as a baseline we can compare different ways of equipping it with additional sensors (e.g., sensors that measure angles, distances, etc.). Our goal is to find the smallest set of extra capabilities that empowers the robot to solve the mapping problem. In the basic model, the robot obviously cannot hope to infer the geometry of the polygon it is exploring. Instead, we concentrate on reconstructing the *visibility graph* as a topological map of the polygon. The visibility graph of a polygon $\mathcal{P}$ is the graph $G_{\text{vis}} = (V, E)$, where V are the vertices of $\mathcal{P}$ and E contains the edge $\{u, v\}$ if and only if u and v see each other in $\mathcal{P}$ (i.e., the line segment connecting them does not leave $\mathcal{P}$). Figure 2 gives an example of a polygon with its visibility graph.

Suri et al. showed that a robot with a *pebble* can solve the mapping problem in polygons with holes, even without information about the size of the polygon [21]. Such a *pebble* is a way for the robot to mark a vertex: The robot can drop the pebble at its current location, it can distinguish

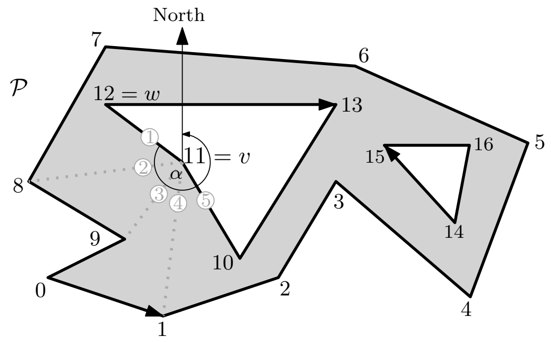

Figure 1: A robot located at vertex v of polygon $\mathcal{P}$. The grey line-segments connect v to the vertices $\{12, 8, 9, 1, 10\}$ visible to v. The line-segments are labeled in the order as they appear in a counterclockwise scan of $\mathcal{P}$, starting on the boundary. The depicted angle α is the angle between the boundary at v and north.

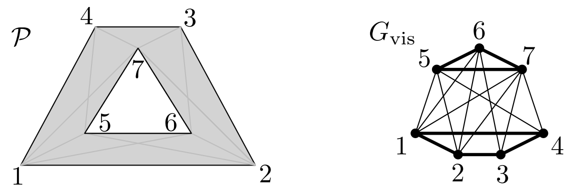

Figure 2: Left: A polygon $\mathcal{P}$ with vertices $V = \{1, 2, 3, 4, 5, 6, 7\}$ and a hole formed by the vertices 5,6,7. The grey line-segments depict the lines of sight. Right: the visibility graph G_{vis} of $\mathcal{P}$. The fat edges denote cycles corresponding to the boundaries $1, 2, 3, 4$ and $5, 6, 7$.

the vertex that holds the pebble as long as it is visible, and it can pick the pebble back up later. A pebble is a powerful tool for the robot. It has been shown for example that a much weaker pebble that cannot be sensed from a distance allows a robot exploring any directed graph to reconstruct the graph [2]. It is an important question whether a robot with weaker abilities can solve the mapping problem in polygons with holes. To the best of our knowledge, no such result is known.

Without a pebble, the presence of holes makes mapping substantially more difficult. For example, consider the robot model introduced in [8]. There, the robot is equipped with the ability to *look-back*, i.e., the robot can identify the vertex from which it arrived in its last move, among its visible vertices. Using such a model, it was then shown that the robot can compute the visibility graph of any *simple* polygon (i.e., without holes), provided that a bound on the number of vertices is known. Figure 3 illustrates that the robot cannot infer the visibility graph in general if the polygon may have holes. In each of the three polygons in the example, the robot senses exactly the same, no matter how it moves. Therefore, there is no way it can distinguish the polygons. Moreover, the example highlights other limitations: the robot cannot infer the number of vertices, and the robot cannot tell whether it is located on a hole. The example relies on the fact that the robot does not know the number of vertices exactly.

Figure 3 also illustrates an important structural property of simple polygons, which polygons with holes do not admit: A simple polygon always has an *ear*, i.e., a vertex whose neighbors on the boundary see each other. In the first polygon in Figure 3 every vertex is an ear, the other two polygons have no ears at all. This property is crucial for existing mapping techniques, because it allows an inductive approach based on "cutting off" ears repeatedly [8, 7]. For polygons with holes, we cannot hope to make use of ears in similar fashion. Solving the mapping problem may thus require a more capable robot.

In this paper we consider the following extension to the basic robot model: the robot knows

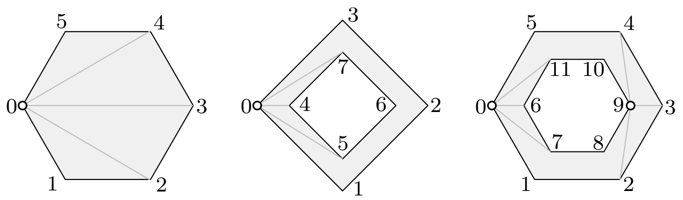

Figure 3: Three polygons that a robot with look-back cannot distinguish, even if it has an upper bound on the number of vertices. Observe that at every vertex of each of the three polygons, the robot observes exactly the same – including the information about which vertex it arrived from in its last move. For convenience, the lines of sight are depicted in grey for the vertices marked by a circle.

an upper-bound $\bar{n}$ on the number of vertices, and the robot has a *boundary compass*. A boundary compass allows to measure the angle at the robot's location formed by the line of sight to the counterclockwise neighbor along the boundary and towards *north*, where *north* is any global reference direction in the plane. Figure 1 illustrates the concept of a boundary compass. We show that such a robot can reconstruct the visibility graph of any polygon with or without holes.

1.1 Related work

Various approaches have been made to modeling minimalistic robots for various environments and objectives [1, 14, 18, 21]. Some effort has been devoted to classifying the power of robot models and to comparing different models in that respect [5, 13, 20]. The basic robot model that serves as a foundation in this paper was introduced in [21], and has been studied in [3, 5, 7, 6, 15, 19].

The focus of research regarding the mapping problem in polygonal environments has so far been on *simple* polygons (i.e., without holes). We provide a brief overview over the different extensions of the basic robot model that have been studied in the past in the context of simple polygons. For a detailed discussion, we refer to [10].

It has been shown that the basic model does not always allow to infer the visibility graph of a polygon [5]. Such a robot can in general not even infer the number of vertices n. On the positive site, it has been shown that a robot can compute the visibility graph with the following extensions to the basic model: (i) the robot has a pebble [21]; (ii) the robot knows an upper bound $\bar{n}$ on the number of vertices, and it has *look-back* [8]; (iii) the robot knows an upper bound $\bar{n}$ on the number of vertices, and it can tell convex from reflex angles, i.e., it can tell for any two visible vertices whether the angle between these two vertices is greater or smaller than π [7]; (iv) the robot has look-back and it can tell convex from reflex angles [3].

An even more minimalistic version of the basic robot model has been considered, the version that restricts the robot to moving along the boundary only. In this model it was shown that knowing the number of vertices n is not sufficient to reconstruct the visibility graph, even when the robot can measure the angle formed by the boundary at each vertex [3]. If the robot can measure the angles between any two lines of sight, however, reconstruction is possible even without prior knowledge of n [9, 12, 11].

An inherent difficulty of visibility graph reconstruction is that these graphs have not yet successfully been characterized [16, 17].

1.2 Problem Definition

Polygon. Throughout this paper we consider the exploration of a polygon $\mathcal{P}$ with polygonal holes and a total of n vertices. We write $\mathcal{H}_1, \mathcal{H}_2, , \mathcal{H}_h$ to denote the holes of $\mathcal{P}$ and $\bar{\mathcal{P}}$ for the enclosing polygon of $\mathcal{P}$ without holes. The *boundaries* of $\mathcal{P}$ consist of the boundary of $\bar{\mathcal{P}}$ together with the boundaries of $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_h$. We will sometimes refer to the boundary of $\bar{\mathcal{P}}$ as the *outer boundary.*

Two vertices u and v are *mutually visible* in $\mathcal{P}$ (or *see each other*) if the line segment connecting the two vertices does not leave $\mathcal{P}$. We call the line-segment between u and v the *line of sight* (between the two vertices).

The *counterclockwise* neighbor of vertex v of $\mathcal{P}$ is v's neighbor u on the boundary of $\mathcal{P}$ such that by moving along the line segment from u to v, the interior of $\mathcal{P}$ lies to the left of the line segment. Figure 1 illustrates this by the arrows on the boundary suggesting the order of vertices which we encounter if we move (iteratively) to the counterclockwise neighbor. Observe the difference of the order if placed on a hole or on the outer boundary of $\mathcal{P}$: an iterative process of moving to the counterclockwise neighbor results in a (a) a counterclockwise walk if the robot moves on the outer boundary $\bar{\mathcal{P}}$, or (b) a clockwise walk if the robot moves on a hole.

Robot. A robot is modeled as a "moving point". Initially, the robot is placed at a vertex of $\mathcal{P}$. Being at a vertex v, the robot observes the following information about $\mathcal{P}$: (i) the number of vertices visible to v, and (ii) the angle $\alpha_v^{\uparrow}$ of the ray from v to its counterclockwise neighbor on the boundary towards a globally fixed direction that we will refer to as *north*. See Figure 1 for illustration.

The robot can order the visible vertices: starting on the boundary, the robot can sort all lines of sight at v as they appear in a counterclockwise scan of the polygon. This naturally induces an ordering of the visible vertices (Figure 1 illustrates this ordering). The robot can select a position in this ordering and move to the corresponding vertex (without knowing the global identity of it).

Visibility graph. The visibility graph of a polygon $\mathcal{P}$ is an undirected graph $G_{\text{vis}} = (V, E)$, where V is the set of vertices of $\mathcal{P}$, and E contains the edge $\{u, v\}$ if and only if u and v see each other in $\mathcal{P}$ (i.e., edges of G_{vis} are lines of sight of $\mathcal{P}$). To reflect the local sensing of a robot at a vertex $u \in V$, we will consider a directed and edge-labeled version of the visibility graph. We replace every undirected edge $\{u, v\}$ with two directed edges (u, v) and (v, u). We label every edge (u, v) by a label $l(u, v)$ which encodes the information observed by a robot at vertex u. Formally, we set $l(u, v) = (i, \alpha_u^{\uparrow})$, where i denotes that (u, v) is the i-th line of sight at u in counterclockwise order, and $\alpha_u^{\uparrow}$ is the angle between the ray towards u's counterclockwise neighbor and north. Observe that the edge (u, v) will generally have a different label than the edge (v, u). With this transformation, we can regard the robot operating inside the polygon $\mathcal{P}$ as an agent moving along the edges of the directed and edge-labeled visibility graph G_{vis}, where the agent sees the labels of the outgoing edges of the vertex it is located at.

Minimum base graph. An edge-labeled directed multi-graph G' is a *base graph* of an edge-labeled directed graph G, if every vertex v of G can be mapped to a vertex v' of G' such that every path in G starting at v with an induced sequence λ of edge labels has a corresponding path starting at v' in G' with the same induced sequence λ of edge labels, and vice versa, i.e., every path in G' has a counterpart in G. A *minimum* base graph G^* of G is a base graph of G of minimum size. Every graph G has a unique minimum base graph G^* (up to isomorphism) [4].

A useful interpretation of the minimum base graph is to see every vertex of the minimum base graph G^* as representing a class of vertices of G. Every two vertices of G that map to the same vertex of G^* belong to the same class. Each class groups vertices together according to the same observation along paths in G specified by a sequence of edge-labels (recall that at any vertex u there are no two adjacent outgoing edges with the same label, and thus any sequence of edge-labels uniquely specifies a path in G). This is useful when arguing about the robot: Starting from any vertex in a class, the robot observes the same for every sequence of movement decisions. In other words, the vertices of the same class are indistinguishable by the robot by means of moving and sensing. Moreover, a minimum base graph G^* can be used as a kind of map as well. Being located in vertex $v \in G_{\text{vis}}$ and knowing the corresponding class v^* in G^*, we can use G^* to navigate the robot to any other class of G^*.

For an example, consider the visibility graphs in Figure 3. For the basic robot, i.e., without sensing the angles $\alpha_v^{\uparrow}$, the edge-label of every edge in the directed visibility graph only encodes the position of the corresponding line of sight in the local ordering. In that case, it is easy to observe that the multi-graph consisting of one node with five self-loops labeled (1), (2), (3), (4), and (5), is indistinguishable from the three visibility graphs by the robot. Because there is obviously no smaller such graph, it is the *minimum* base graph of each of the three visibility graphs. Obviously, computing the minimum base graph does not help the basic robot to solve the mapping problem: as far as it can tell, it could be in any of the three polygons – the minimum base graph does not help to distinguish them.

In this paper we will show that if the robot also has a boundary compass (i.e., it can measure $\alpha_v^{\uparrow}$), then the minimum base graph of every correspondingly edge-labeled visibility graph G_{vis} is the visibility graph itself. Therefore, computing the minimum base graph is enough to compute G_{vis}.

Goal. We want to know whether the robot can infer the visibility graph of any polygon $\mathcal{P}$ and determine its location in it. More precisely, given a number $\bar{n} \geq n$, we want to know whether there exists a deterministic algorithm that (i) navigates the robot inside any polygon $\mathcal{P}$ with at most $\bar{n}$ vertices, and (ii) computes from the collected observations the visibility graph G_{vis} of $\mathcal{P}$, as well as the robot's location in G_{vis}.

2 Algorithm

In this section we show that a robot with boundary compass can compute the visibility graph G_{vis} of any polygon $\mathcal{P}$ if it knows an upper bound $\bar{n}$ on the number of vertices of $\mathcal{P}$. We do so by showing that the minimum base graph G^* is equal to the visibility graph G_{vis}. Using the algorithm of [7, 10] for determining the minimum base graph, the algorithm then trivially follows: Since $G^* = G_{\text{vis}}$, we can simply apply the algorithm of [7, 10] and return its result. The algorithm operates on general edge-labeled graphs and also determines the location of the robot in G^* (and thus, in our case, in G_{vis} as well). We note that the running time of the algorithm can be exponential in $\bar{n}$ in the worst case.

We can see this approach as a generic black-box method for solving the mapping problem by some variant of the basic robot (with extended sensing capabilities), assuming that an upper bound $\bar{n}$ on the number of vertices is known. The method is as follows, with its core difficulty lying in step 2.

1. Encode the sensed information in the edge-labels of the directed version of G_{vis};
2. Show that $G^* = G_{\text{vis}}$;
3. Use the fact that the robot can compute the minimum base graph G^* (applying the algorithm of [7, 10]).

2.1 Labeling the visibility graph

We consider the directed and edge-labeled version of G_{vis} as described in Section 1.2. This labelling reflects the local sensing of the robot. Recall that every outgoing edge (u, v) of a vertex u is labeled with $(i, \alpha_u^{\uparrow})$, where i denotes the rank of v in the counterclockwise order of the vertices visible to u, and $\alpha_u^{\uparrow}$ is the angle formed by the ray to the counterclockwise neighbor of u along the boundary and north.

Because of the ordering of the lines of sight, no two labels of outgoing edges at a vertex are the same. Therefore, any walk in the visibility graph can uniquely be described by a starting vertex and a sequence of edge labels.

2.2 Showing that $G^* = G_{\text{vis}}$

To show that the minimum base graph G^* is equal to the visibility graph G_{vis}, we show that every two vertices u and v of G_{vis} are distinguishable by a walk in G_{vis} (thus showing that the two

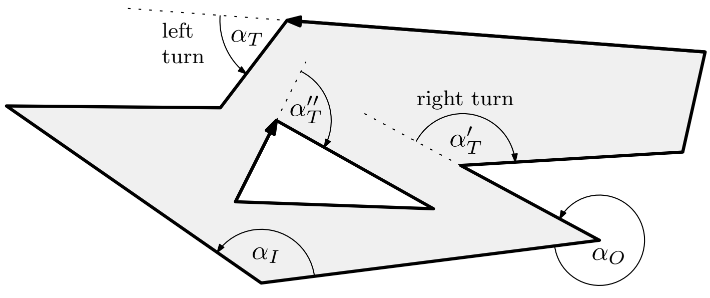

Figure 4: A polygon with one hole. Marked inner angle α_I, outer angle α_O and three turn angles $\alpha_T > 0$, $\alpha'_T < 0$, and $\alpha''_T < 0$. Observe that α_T and α'_T are the turn angles in counterclockwise direction, whereas α''_T is the turn angle in clockwise direction.

vertices cannot be in the same class of G^*). We proceed in several steps. In the following, we let $\mathcal{H} \in \{\bar{\mathcal{P}}, \mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_h\}$ be a hole $\mathcal{H}_i$ or the enclosing polygon $\bar{\mathcal{P}}$.

Distinguishing vertices of $\mathcal{H}$

We will show that no two vertices of $\mathcal{H}$ belong to the same class of G^*, i.e., we show that every two vertices of $\mathcal{H}$ are distinguishable by the robot.

The robot can consciously walk along the boundary of $\mathcal{H}$: It can just repeatedly move to its counterclockwise neighbor on the boundary (i.e., to its first visible vertex). This will result in a counterclockwise (if $\mathcal{H} = \bar{\mathcal{P}}$) or clockwise (if $\mathcal{H} \in \{\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_h\}$) walk along the boundary of $\mathcal{H}$, in which the robot possibly visits each vertex of $\mathcal{H}$ more than once. Any such walk induces a sequence of observations (provided by the sensing capabilities of the robot). Let $n_{\mathcal{H}}$ denote the number of vertices of $\mathcal{H}$. After at most $\bar{n}$ steps, the robot is guaranteed to have visited every vertex of $\mathcal{H}$ at least once. Therefore, in a walk along the boundary of $\mathcal{H}$, observations repeat with a *period* of at most $n_{\mathcal{H}}$. Formally, $p \in \mathbb{N}$ is a period of a sequence $(a_i)_{i\in\mathbb{N}}$ if $a_i = a_{i+kp}$ for all $k \in \mathbb{N}$, and we say that the first p elements $(a_1, \dots, a_p)$ *repeat* in the sequence. We show in the following that we can uniquely identify the exact value of $n_{\mathcal{H}}$ by considering the sequence of observations induced by $\bar{n}$ moves along the boundary.

We will use the following facts about the sum of the *inner* and *outer* angles, and about the *rotation number* of a simple polygon. Inner angles of $\mathcal{P}$ are the angles on the inside of $\mathcal{P}$ formed by two adjacent segments of a boundary. An outer angle is the counterpart of an inner angle – the angle on the outside of $\mathcal{P}$ formed by two adjacent boundary segments. See Figure 4 for illustration. The *rotation number* of a simple polygon measures (in angles), informally, how much the boundary turns. Formally, consider three consecutive neighbors u, v, w on the boundary of a simple polygon $\mathcal{P}$ in a chosen direction (counterclockwise or clockwise direction). The *turn angle* of the polygon at vertex v (in the chosen direction) is the angle at v formed by the rays $\overrightarrow{uv}$ and $\overrightarrow{vw}$ in this order (!). The rotation number in the chosen direction of $\mathcal{P}$ is the sum of its turn angles in the chosen direction. Turn angles are signed: a "left" turn gives a positive angle $\alpha_T \in (0, \pi)$, and a "right" turn gives a negative turn angle $\alpha_T \in (-\pi, 0)$. Figure 4 illustrates these angles in an example.

Fact 1. *The sum of all inner angles of a simple polygon is $(n-2)\pi$. The sum of all outer angles of a simple polygon is $(n+2)\pi$. The rotation number in the counterclockwise direction is 2π, and the rotation number in the clockwise direction is -2π.*

We use the *robot direction* to denote the direction which the robot induced on $\mathcal{H}$ if it walks along the boundary by iteratively moving to its counterclockwise neighbor. That is, if $\mathcal{H}$ is the outer boundary $\bar{\mathcal{P}}$, then *robot direction* is the counterclockwise direction, otherwise (if $\mathcal{H}$ is a hole $\mathcal{H}_i$) then *robot direction* is the clockwise direction.

We use the rotation number to infer $n_{\mathcal{H}}$ – the size of $\mathcal{H}$. While moving along the boundary of $\mathcal{H}$ the turn angles can be computed by using the angles provided by the boundary compass.

Proposition 2. *Let u be a vertex of $\mathcal{H}$ and let v be the counterclockwise neighbor of u on $\mathcal{H}$. Then, the turn angle in the robot direction is equal to $\alpha_u^{\uparrow} - \alpha_v^{\uparrow}$ (mapped to the interval $(-\pi, \pi)$).*

For every vertex $u \in \mathcal{H}$, the walk along the boundary induces a sequence $\vec{\alpha}(u) = (\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_{\bar{n}}, \ldots)$ of turn angles. Recall that $\alpha_i \in (-\pi, \pi)$. By Fact 1 we know that $\sum_{i=1}^{n_{\mathcal{H}}} \alpha_i = \pm 2\pi$. For simplicity of the exposition, we will assume that the sum equals 2π. The case when the sum equals -2π can be handled analogously. Thus, we have $\sum_{i=1}^{k \cdot n_{\mathcal{H}}} \alpha_i = k \cdot 2\pi$. Obviously, the sequence $(\alpha_1, \ldots, \alpha_{n_{\mathcal{H}}})$ appears periodically in $\vec{\alpha}(u)$ with period $n_{\mathcal{H}}$. We claim that this sequence is the smallest sequence that periodically repeats in $\vec{\alpha}(u)$ and sums to 2π.

Lemma 3. *Sequence $(\alpha_1, \ldots, \alpha_{n_{\mathcal{H}}})$ is the only sequence that periodically repeats in $\vec{\alpha}(u)$ and sums to 2π.*

Proof. Consider the sequence $(\alpha_1, \ldots, \alpha_k)$, $k \in \mathbb{N}$. We will show that if the sequence $(\alpha_1, \ldots, \alpha_k)$ sums to 2π and periodically repeats in $\vec{\alpha}(u)$, then $k = n_{\mathcal{H}}$. Assume therefore that the sequence repeats and sums to 2π, i.e., $\sum_{i=1}^{k} \alpha_i = 2\pi$. Consider the sum $X := \sum_{i=1}^{k \cdot n_{\mathcal{H}}} \alpha_i$. By the assumption, we can write the sum as $X = n_{\mathcal{H}} \sum_{i=1}^{k} \alpha_i = n_{\mathcal{H}} \cdot 2\pi$. At the same time, because $(\alpha_1, \ldots, \alpha_{n_{\mathcal{H}}})$ periodically repeats in $\vec{\alpha}(u)$, we can also write $X = k \sum_{i=1}^{n_{\mathcal{H}}} \alpha_i = k \cdot 2\pi$. Therefore, $k = n_{\mathcal{H}}$. $\square$

Lemma 3 immediately gives the robot a way to compute the number of vertices of $\mathcal{H}$. It suffices to identify the smallest period of $\vec{\alpha}(u)$ that sums to 2π. This is an easy task since the robot has an upper bound $\bar{n}$ on the total number of vertices – the robot walks $\bar{n}!$ number of steps along the boundary and identifies the smallest period in the resulting sequence $(\alpha_1, \ldots, \alpha_{\bar{n}!})$ that sums to 2π.

We can now show that every two vertices of $\mathcal{H}$ are distinguishable.

Lemma 4. *No two vertices of $\mathcal{H}$ belong to the same class of the minimum base graph G^* of G_{vis}.*

Proof. We have shown that, for any vertex $u \in \mathcal{H}$, the sequence $(\alpha_1, \ldots, \alpha_{n_h})$ is the only sequence that periodically repeats in $\vec{\alpha}(u)$ and sums to 2π.

Consider any two vertices $u, v \in \mathcal{H}$. We claim that the walk along the boundary of size $n_{\mathcal{H}}$ distinguishes the two vertices. Obviously, if $\vec{\alpha}(u) \neq \vec{\alpha}(v)$, then also the corresponding sequences of observed angles $\angle_w^{\uparrow}$ at vertices w along the walk cannot be the same (as the latter implies the former). In the case when $\vec{\alpha}(u) = \vec{\alpha}(v)$ we have that the subsequence of $\vec{\alpha}(u)$ between u and v repeats in $\vec{\alpha}(u)$. Let p be the distance between u and v on the walk. Thus, p is the period of the subsequence in $\vec{\alpha}(u)$ and we have that p divides $n_{\mathcal{H}}$. Thus, the subsequence between u and v repeats $n_{\mathcal{H}}/p$ times within the first $n_{\mathcal{H}}$ elements of $\vec{\alpha}(u)$. It follows that $\sum_{i=1}^{p} \alpha_i =: \beta \neq 0$ (as otherwise $\sum_{i=1}^{n_{\mathcal{H}}} \alpha_i = 0$, a contradiction). Also, $\beta < 2\pi$ by Lemma 3. But then $\alpha_u^{\uparrow} \neq \alpha_v^{\uparrow}$ because $\alpha_v^{\uparrow} = \alpha_u^{\uparrow} - \beta$. Hence, obviously, the two vertices are distinguishable. $\square$

With this lemma, we have not excluded the case that there may be two vertices $u \in \mathcal{H}$, $v \in \mathcal{P} \setminus \mathcal{H}$ belonging to the same class of the minimum base graph.

Distinguishing vertices of $\bar{\mathcal{P}}$ from the rest

We show that the robot can distinguish any vertex of $\bar{\mathcal{P}}$ from the vertices in $\mathcal{P} \setminus \bar{\mathcal{P}}$. Obviously, any vertex $u \in \bar{\mathcal{P}}$ can be distinguished from a vertex $v \in \mathcal{H}_i$, if $n_{\bar{\mathcal{P}}} \neq n_{\mathcal{H}_i}$: The walk along the boundary from the respective vertices u and v will induce sequences $\vec{\alpha}(u)$ and $\vec{\alpha}(v)$ of different periods. Let us therefore concentrate on the case where $n_{\bar{\mathcal{P}}} = n_{\mathcal{H}_i}$. We use Fact 1 again.

For Fact 1 we have defined an inner angle of a *simple* polygon. We can naturally define an *inner angle of polygon $\mathcal{P}$ with holes* to be the angle lying inside $\mathcal{P}$ and formed by two adjacent boundary segments. Note that for a vertex $v \in \mathcal{H}_i$, the corresponding inner angle in $\mathcal{P}$ is actually the outer angle of the simple polygon $\mathcal{H}_i$ (see Figure 5 for illustration).

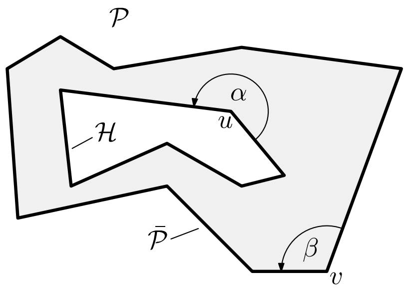

Figure 5: Inner angles of polygon $\mathcal{P}$: α is the inner angle of the polygon at vertex of $u \in \mathcal{H}$ and β is the inner angle of the polygon at vertex $v \in \bar{\mathcal{P}}$. Observe that when $\mathcal{H}$ is considered as a simple polygon, then α is its outer angle.

Therefore, by Fact 1, the sum of the inner angles of $\mathcal{P}$ at vertices of $\bar{\mathcal{P}}$ is $(n_{\bar{\mathcal{P}}} - 2)\pi$, whereas the sum of the inner angles of $\mathcal{P}$ at vertices of $\mathcal{H}_i$ is $(n_{\mathcal{H}_i} + 2)\pi$.

Let $\mathcal{H}$ be a hole or $\bar{\mathcal{P}}$ at which the robot is positioned. The observed angles by the robot in a walk around the boundary of $\mathcal{H}$ induce the inner angles of $\mathcal{P}$ at vertices of $\mathcal{H}$.

Proposition 5. *The inner angle of $\mathcal{P}$ at a vertex $v \in \mathcal{H}$ is equal to $\pi - \alpha_T$, where α_T is the signed turn angle at vertex v.*

Proposition 5 gives an immediate way to distinguish $\bar{\mathcal{P}}$ from any hole $\mathcal{H}_i$.

Lemma 6. *No two vertices $u \in \bar{\mathcal{P}}$, $v \in \mathcal{H}_i$ belong to the same class of the minimum base graph G^*.*

Proof. We have already argued that the vertices are distinguishable if the number of vertices in $\bar{\mathcal{P}}$ and $\mathcal{H}_i$ is different. Without loss of generality, assume that $n_{\bar{\mathcal{P}}} = n_{\mathcal{H}_i} = n'$. By Fact 1 we know that the inner angles of $\bar{\mathcal{P}}$ sum up to $(n' - 2)\pi$ and the inner angles of any hole sum up to $(n' + 2)\pi$. Proposition 5 provides a correspondence between the sequence of n' observed angles from the boundary compass and the inner angles of $\mathcal{P}$. As the sums of the inner angles are different for vertices of $\bar{\mathcal{P}}$ and vertices of a hole, the sequence of observed angles have to be different, too. Therefore, a counterclockwise walk along the boundary allows to distinguish between u and v. $\square$

It remains to distinguish vertices of different holes $\mathcal{H}_i$ and $\mathcal{H}_j$ for $i \neq j$.

Distinguishing the vertices of two holes

Recall that the vertices of $\bar{\mathcal{P}}$ can be uniquely distinguished in G^*, i.e., they form a singleton class in G^*. Therefore, as the robot can navigate in G^* to get to any class of G^*, it can get to any vertex of $\bar{\mathcal{P}}$.

We use the boundary of $\bar{\mathcal{P}}$ as a reference point to identify all other vertices of $\mathcal{P}$ uniquely. Consider an arbitrary vertex $v_1 \in \bar{\mathcal{P}}$. We use it as a kind of "origin" of G_{vis} to distinguish any two vertices $u \in \mathcal{H}_i$, $v \in \mathcal{H}_j$, $i \neq j$. Observe that, because G_{vis} is strongly connected, there exists a closed walk from v_1 that visits all vertices of G_{vis} – a Hamiltonian walk. We will now see the walk as a sequence L of both classes (vertices) of G^* and of edge-labels: every walk in G_{vis} translates to a walk in G^*; we add the visited vertices of G^* into L in the order induced by the walk. We will abuse the notation a bit, and use L to sometimes refer to the walk and sometimes to the edge-labels.

A sufficient condition to distinguish any two vertices u and v is that the walk L does not have a period smaller than $|L|$ (where $|L|$ denotes the length of the sequence). Having such a walk at hand, we can easily distinguish the vertices u and v. We consider L as an infinite sequence formed by an infinite concatenation of L. Let W_u be the closed walk in L of length $|L|$ starting from the first occurrence of u in L, and let W_v be the closed walk in L of length $|L|$ starting from the first occurrence of v in L. Obviously, because L has a period $|L|$, $W_u \neq W_v$, and thus these paths are distinguishing paths for u and v.

It remains to be shown that a Hamiltonian walk L of period $|L|$ exists in G_{vis}.

Lemma 7. *The visibility graph G_{vis} of any polygon $\mathcal{P}$ with holes contains a Hamiltonian walk L of period $|L|$.*

Proof. We construct one such Hamiltonian walk as follows. Let v_1 be a vertex of $\bar{\mathcal{P}}$. We initially set L to be the walk from v_1 along the boundary of $\bar{\mathcal{P}}$. If $v_1, \ldots, v_{n_{\bar{\mathcal{P}}}}$ denote the vertices of the boundary of $\bar{\mathcal{P}}$, then initially $L = (v_1, \ldots, v_{n_{\bar{\mathcal{P}}}})$. Obviously, L is not Hamiltonian. We extend L as follows. We mark all vertices of $\bar{\mathcal{P}}$ as *visited*; all other vertices are marked as *unvisited*. For every vertex v_i of $\bar{\mathcal{P}}$ we compute, in the order as the vertices appear on the boundary (starting from v_1), a depth-first search tree in the graph induced by the unvisited vertices of G_{vis}. The depth-first search from v_i induces a closed walk $L(v_i)$ on the computed depth-first search tree. We add this walk into the walk L in the place of v. We mark all vertices from the depth-first search tree as visited and proceed with the next vertex v on the boundary of $\bar{\mathcal{P}}$.

We have computed a closed walk L in G_{vis} of the form $L(v_1), L(v_2), \ldots, L(v_{n_{\bar{\mathcal{P}}}})$. Obviously, the walk visits every vertex of G_{vis}.

Moreover, the walk has period $|L|$: Recall that we can identify v_1 and v_2 (as they are from the boundary of $\bar{\mathcal{P}}$); Observe that the occurrences of v_i come consecutively in L without being "interrupted" by another vertex v_j, $i \neq j$; Therefore, we can uniquely identify the last occurrence of v_1 in L as it comes before the first occurrence of v_2; Thus, any two vertices u and v are distinguishable by the different distances from u and v to the last occurrence of v_1, respectively. $\square$

Lemma 7 thus implies the following.

Lemma 8. *No two vertices from different holes appear in the same class of G^*.*

Putting pieces together

Lemma 4, Lemma 6, and Lemma 8 imply the main result of the paper:

Theorem 9. *The minimum base graph G^* is equal to the edge-labeled visibility graph G_{vis}.*

Theorem 10. *The robot can compute the visibility graph G_{vis} of any polygon $\mathcal{P}$ with holes, and it can localize its position in G_{vis}.*

Proof. The robot can compute the minimum base graph G^* of G_{vis} and its position therein using the algorithm in [7]. Theorem 9 implies that the computed graph G^* is actually what we want – the visibility graph G_{vis}. $\square$

3 Conclusions

We have studied the mapping and localization problem by a simple robot inside a polygon $\mathcal{P}$ with a boundary compass. We have presented a black-box solution approach to show that such a robot can always compute the visibility graph G_{vis} of $\mathcal{P}$ whenever it knows an upper bound on the number of vertices of $\mathcal{P}$. The central part of the black-box approach is to prove that the minimum base graph G^* of G_{vis} is the visibility graph G_{vis}, i.e., $G^* = G_{\text{vis}}$. Our algorithm uses the generic algorithm of Chalopin et al. [7] for computing the minimum base graph of any edge-labeled directed graph G by a robot that only senses the edge-labels of the outgoing edges and knows an upper bound on the number of vertices of G. This algorithm has an exponential running time in the worst case. We leave it open whether the time complexity can be improved for a robot with boundary compass.

Due to the fundamental importance of the mapping problem, our solution has further implications for other tasks. For example, it follows that a collection of robots with boundary compass and knowledge of an upper bound on the number of vertices can solve the *strong rendezvous* problem, i.e., they can meet in a vertex of $\mathcal{P}$ (even in an asynchronous model with no communication between the robots).

References

[1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, 1999.

[2] M. Bender, A. Fernandez, D. Ron, A. Sahai, and S. Vadhan. The power of a pebble: Exploring and mapping directed graphs. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, pages 269–287, 1998.

[3] D. Bilò, Y. Disser, M. Mihalák, S. Suri, E. Vicari, and P. Widmayer. Reconstructing visibility graphs with simple robots. *Theoretical Computer Science*, 444:52–59, 2012.

[4] P. Boldi and S. Vigna. Fibrations of graphs. *Discrete Mathematics*, 243(1–3):21–66, 2002.

[5] J. Brunner, M. Mihalák, S. Suri, E. Vicari, and P. Widmayer. Simple robots in polygonal environments: A hierarchy. In *Proceedings of the Fourth International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, pages 111–124, 2008.

[6] J. Chalopin, S. Das, Y. Disser, M. Mihalák, and P. Widmayer. How simple robots benefit from looking back. In *Proceedings of the 7th International Conference on Algorithms and Complexity*, pages 229–239, 2010.

[7] J. Chalopin, S. Das, Y. Disser, M. Mihalák, and P. Widmayer. Telling convex from reflex allows to map a polygon. In *Proceedings of the 28th International Symposium on Theoretical Aspects of Computer Science*, pages 153–164, 2011.

[8] J. Chalopin, S. Das, Y. Disser, M. Mihalák, and P. Widmayer. Mapping simple polygons: How robots benefit from looking back. *Algorithmica (to appear)*, to appear.

[9] D. Chen and H. Wang. An improved algorithm for reconstructing a simple polygon from the visibility angles. *Computational Geometry: Theory and Applications*, 45:254–257, 2012.

[10] Y. Disser. *Mapping Polygons*. PhD thesis, ETH Zurich, 2011.

[11] Y. Disser, M. Mihalák, and P. Widmayer. Reconstruction of a polygon from angles without prior knowledge of the size. Technical Report 700, ETH Zürich, Institute of Theoretical Computer Science, November 2010.

[12] Y. Disser, M. Mihalák, and P. Widmayer. A polygon is determined by its angles. *Computational Geometry: Theory and Applications*, 44:418–426, 2011.

[13] B. R. Donald. On information invariants in robotics. *Artificial Intelligence*, 72(1-2):217–304, 1995.

[14] A. Ganguli, J. Cortés, and F. Bullo. Distributed deployment of asynchronous guards in art galleries. In *Proceedings of the 2006 American Control Conference*, pages 1416–1421, 2006.

[15] B. Gfeller, M. Mihalak, S. Suri, E. Vicari, and P. Widmayer. Counting targets with mobile sensors in an unknown environment. In *Proceedings of the 3rd International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, pages 32–45, 2007.

[16] S. K. Ghosh. *Visibility Algorithms in the Plane*. Cambridge University Press, 2007.

[17] S. K. Ghosh and P. P. Goswami. Unsolved problems in visibility graph theory. In *Proceedings of the India-Taiwan Conference on Discrete Mathematics*, pages 44–54, 2009.

[18] M. Katsev, A. Yershova, B. Tovar, R. Ghrist, and S. M. LaValle. Mapping and pursuit-evasion strategies for a simple wall-following robot. *IEEE Transactions on Robotics*, 27(1):113–128, 2011.

[19] A. Komuravelli and M. Mihalák. Exploring polygonal environments by simple robots with faulty combinatorial vision. In *Proceedings of the 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 458–471, 2009.

[20] J. M. O'Kane and S. M. LaValle. On comparing the power of robots. *International Journal of Robotics Research*, 27(1):5–23, 2008.

[21] S. Suri, E. Vicari, and P. Widmayer. Simple robots with minimal sensing: From local visibility to global geometry. *International Journal of Robotics Research*, 27(9):1055–1067, 2008.