How Simple Robots Benefit from Looking Back⋆

Jérémie Chalopin[1],⋆⋆, Shantanu Das[1], Yann Disser[2],
Matúš Mihalák[2], and Peter Widmayer[2]

[1] LIF, CNRS & Aix-Marseille Université,
`jeremie.chalopin@lif.univ-mrs.fr`, `shantanu.das@acm.org`
[2] Institute of Theoretical Computer Science, ETH Zürich
`{ydisser,mmihalak,widmayer}@inf.ethz.ch`

Abstract. We study the sensor and movement capabilities that simple
robots need in order to create a map of an unknown polygon of size n,
and to meet. We consider robots that can move from vertex to vertex,
can backtrack movements, and see distant vertices in counter-clockwise
order but have no means of visibly identifying them. We show that such
robots can always solve the weak rendezvous problem and reconstruct
the visibility graph, given an upper bound on n. Our results are tight:
The strong rendezvous problem, in which robots need to gather at a
common location, cannot be solved in general, and without a bound on
n, not even n can be determined. In terms of mobile agents exploring
a graph, our result implies that they can reconstruct any graph that is
the visibility graph of a simple polygon. This is in contrast to the known
result that the reconstruction of arbitrary graphs is impossible in general,
even if n is known.

1 Introduction

We are interested in the study of simple but autonomous robots that are capable
of performing tasks like searching, cleaning, guarding or gathering data in un-
known territories. One of the most fundamental problems involved in these tasks
is to obtain a map of the environment. The complexity of this problem depends
on the environment in which a robot operates, and on the robot's sensor capa-
bilities. For example, it is a substantial difference whether there are identifiable
landmarks in the environment, or whether a robot has access to powerful sen-
sors like GPS. In some settings, large numbers of simple robots are deployed in
an environment and expected to cooperate and solve given tasks. A fundamen-
tal problem that arises in many settings with multiple robots is the *rendezvous*
problem in which robots either need to gather at the same location (strong ren-
dezvous) or at least need to position themselves such that they are visible to
each other (weak rendezvous).

The task of mapping an unknown environment has previously been studied
both in graphs and in geometric environments, where robots move in the plane.

In this paper we focus on the latter case, more precisely on the case of robots operating in polygonal environments. For many tasks a geometrically accurate map of the environment is not needed, but a qualitative description of its shape is sufficient. The *visibility graph* of a polygon is such a qualitative map: Its nodes are the vertices of the polygon and two nodes are joined by an edge if they are visible to each other, i.e., if the line segment connecting them in the plane lies entirely within the polygon. Visibility graphs and their characterization have been studied extensively [7]. We aim to find minimalistic configurations of sensors and movement capabilities that enable simple robots to reconstruct the visibility graph of a polygon, and, in the case of multiple robots, to solve the rendezvous problem.

There have been several studies concerned with the capabilities of minimalistic robots. For instance it has been shown that a single pebble alone already enables a simple robot to reconstruct the visibility graph [9]. The rendezvous problem in the plane has been solved for robots that measure distances. Studies in this setting consider for example robots with limited visibility and memory [1] or robots subject to measurement errors [5]. Other models for simplistic robots have been studied (e.g. [3,6,11]). The variant considered in this paper originates from [9]. Roughly speaking, we allow a robot to order the vertices it sees in counter-clockwise (ccw) order and to drive to any vertex it sees (for more details cf. Section 2). Bilò et al. showed that such a robot cannot reconstruct the visibility graph of a polygon by moving along the boundary only, even if the number of vertices n is known, unless provided with additional capabilities [2]. We extend the model by allowing the robot to *look-back* after moving, i.e. to know which vertex it came from among those now visible to it. In [2], another sensor was used in addition to the look-back capability and an algorithm for reconstructing the visibility graph was given for a robot equipped with both sensors, when the number n of vertices is known. In this paper we show that the capability to look-back alone is already sufficient, even if only an upper bound on n is known. It was previously shown that if robots have no previous knowledge about the polygon (i.e. not even a bound on n), they cannot reconstruct its size and thus they cannot reconstruct the visibility graph [3]. Furthermore we show that the look-back capability is sufficient for solving the weak rendezvous problem in all polygons, even if only an upper bound on n is known. This is in contrast to the fact that the strong rendezvous problem cannot be solved in general, even when the entire geometry of the polygon is known.

The problem of reconstructing the visibility graph of a polygon can be seen as a special case of the problem of reconstructing a general graph. In a graph a robot is allowed to move along edges and sense the degree of the vertex it is currently located at. A usual assumption is that the edges incident to a vertex are distinctly labeled by a *port-numbering* function. The resulting model is equivalent to the *message-passing network model* of distributed computation [4]. A characterization of the graphs for which the rendezvous problem is solvable has been given [4,10]. From the results of [10] it is easy to infer that there are graphs which cannot be distinguished by a robot, i.e. it is not always possible

to reconstruct a graph. A robot with look-back capability moving in a polygon can imitate an agent that is moving in the corresponding visibility graph, which allows us to borrow concepts that led to the above results. We are able to show that while the graph reconstruction problem and the rendezvous problem are not solvable in general, the reconstruction problem as well as the weak rendezvous problem are solvable for visibility graphs, if a bound on n is known and if the edges on the polygon boundary can be distinguished.

2 Polygons and LB-Robots

Polygons. In this work we consider simple polygons only. Let in the following $\mathcal{P}$ be a simple polygon with n vertices. We denote the set of vertices of $\mathcal{P}$ by $V(\mathcal{P})$, where we drop the argument $\mathcal{P}$ whenever the polygon is evident from the context. The boundary of $\mathcal{P}$ together with a (arbitrary) choice of a starting vertex v_0 induces an order among the vertices and we write $v_0, \ldots, v_{n-1}$ to denote the vertices along the boundary in counter-clockwise (ccw) order. We define $\mathrm{chain}(v_i, v_j) := (v_i, v_{i+1}, \ldots, v_j)$. Note that all indices of vertices are modulo n. In the following we assume polygons to be in general position, i.e. no three vertices lie on a common line.

We say two vertices $u, w \in V$ *see* each other or u *sees* w, if and only if the line segment $\overline{uw}$ lies entirely in $\mathcal{P}$ – in particular v_i sees v_{i+1} for all i. If v_{i-1} sees v_{i+1}, we say v_i forms an *ear*.

The visibility graph of a polygon has a node for every vertex of the polygon and an edge for every pair of vertices that see each other. We define the degree of a vertex to be the degree of the corresponding node in the visibility graph.

With $\mathrm{vis}(v_i) = (u_1, \ldots, u_d)$ we denote the sequence of vertices that a vertex $v_i \in V$ of degree d sees, enumerated in ccw order along the boundary starting at $u_1 = v_{i+1}$ and ending at $u_d = v_{i-1}$. We write $\mathrm{vis}_j(v_i)$, $1 \le j \le d$, to denote u_j, $\mathrm{vis}_{-j}(v_i)$ to denote u_{d+1-j} and $\mathrm{vis}_0(v_i)$ to denote v_i itself.

Let $C = (u_0, \ldots, u_{m-1})$ be a cycle of length m in the visibility graph of $\mathcal{P}$. We say C is an *ordered cycle*, if and only if $u_0, \ldots, u_{m-1}$ appear on the boundary of $\mathcal{P}$ in that order (ccw). As u_i sees $u_{i+1 \bmod m}$ for $0 \le i \le m-1$, C induces a subpolygon of $\mathcal{P}$. Note that C being an ordered cycle implies that the boundary of the induced subpolygon does not self-intersect.

Lemma 1. *Let $\mathcal{P}$ be a simple polygon of size $n \ge 4$. For all $0 \le i < n$ we have that the degree of v_i or v_{i+1} is greater than two.*

Proof. As any polygon has a triangulation, two vertices that see each other need to share a third vertex they both see. In our instance two vertices v_i and v_{i+1} thus need to share a third vertex w that they both see. For the sake of contradiction assume there is a vertex v_i such that v_i and v_{i+1} have degree two. It follows that w must be a neighbor of both vertices on the boundary which is a contradiction to the assumption $n \ge 4$. □

Look-Back Robot. We now define a basic robot which we use as a foundation for more sophisticated models later on. A basic robot r in a polygon $\mathcal{P}$ is modeled to be a moving point initially situated at some vertex of $\mathcal{P}$. We allow it to move to the neighboring vertices along the boundary of $\mathcal{P}$, and we write v_r to denote its current location. We say r sees a vertex u if $u \in \mathrm{vis}(v_r)$. While r remains at v_r, we allow it to sense the degree of v_r and to put the vertices it sees into ccw order starting with v_{r+1}. Note that the robot has no immediate way of globally identifying the vertices it sees, i.e. knowing their global index with respect to v_r (i.e. their index counting from v_r in ccw order along the boundary). As our intention lies in obtaining a weak model, we do not allow robots to gather sensory information while moving. We do not impose a limitation on the robots' memory.

A natural extension of the basic model lies in allowing a robot r to not only move along the boundary, but to any vertex it sees. In general r does not know the identity (= global index) of any vertex but its neighbors. This means that any local information it gathered while driving along the boundary may be lost once it decides to drive to a distant vertex. In order to get around this problem we further enhance our model by adding the look-back capability: Consider a robot at some vertex v_i that drives to a distant visible vertex v_j. Without look-back, the robot afterwards has no way of knowing the identity of v_i in $\mathrm{vis}(v_j)$. Look-back provides the robot with the index k such that $v_i = \mathrm{vis}_k(v_j)$. We call such a robot a *look-back robot* or *LB-robot*.

3   Results

As soon as there is one vertex $v^\star$ of the environment that a robot can distinguish from all other vertices, it can easily determine the visibility graph: It can define $v_0 = v^\star$ and identify the global index of any vertex it sees easily by driving there and then along the boundary until it encounters $v^\star$, counting the number of steps along the boundary (cf. [9]).This means that if the robot can perceive any non-symmetrical structure, it can define a unique vertex $v^\star$ and thus reconstruct the visibility graph. In other words, for most polygons reconstructing the visibility graph (or meeting other identical robots) is an easy task. Only highly symmetrical polygons present a challenge. In order to solve the problem in general, we make use of concepts first introduced in the context of distributed networks.

In distributed computing, a network is modelled as a graph whose edges are labelled by so-called port numbers (local orientation), such that edges incident to a vertex are assigned distinct labels [10]. As our robot is capable of bringing the vertices it sees into ccw order, and as this order is the same whenever a vertex is visited, the robot has access to a port numbering. Look-back allows the robot to backtrack any number of moves or, in the sense of distributed computing, to send messages back and forth between vertices. In distributed networks it has proven beneficial [10] to define a *view* of a vertex as the collection of information a node can gather by sending and receiving messages. We will introduce the same concept in our setting of a robot moving along the visibility graph of a polygon and show that it is a powerful tool for analysing the capabilities of the robot.

In the following we adapt the definition of the view of a vertex, to suit our robot perspective. Consult Figure 1 (left) along with the definitions.

Definition 1. *Let v be a vertex of a simple polygon $\mathcal{P}$ and d be its degree. The level-1-label $l(v) \equiv l^{(1)}(v)$ of v is given by $l(v) := (r_1, \ldots, r_i, \ldots, r_d)$ where r_i is defined such that $\mathrm{vis}_{r_i}(\mathrm{vis}_i(v)) = v$, i.e. r_i is the index of v in $\mathrm{vis}_i(v)$'s list of neighbors. We write $l_j(v)$ to denote r_j and $l_{-j}(v)$ to denote r_{d+1-j} for $0 < j \leq d$. Similarly $l_i(v) = -k$ means $\mathrm{vis}_{-k}(\mathrm{vis}_i(v)) = v$.*

Definition 2. *Let v be a vertex of a simple polygon $\mathcal{P}$ and d be its degree. Let $\mathrm{vis}(v) = (u_1, \ldots, u_d)$. For $k \in \mathbb{N}, k > 1$, the level-k-label $l^{(k)}(v)$ of v is given by $l^{(k)}(v) := \left(l^{(k-1)}(v), \; \left(l^{(k-1)}(u_1), \ldots, l^{(k-1)}(u_d) \right) \right)$.*

Note that the definition of the level-k-label of a vertex v is equivalent to its view up to depth k as originally defined in [10]. Intuitively it contains the information a backtracking agent can gather by exploring the visibility graph up to a distance of k from v.

In the following we introduce the notion of a level-k-class of a vertex for the equivalence class containing all vertices with the same level-k-labels.

Definition 3. *Let v be a vertex of a simple polygon $\mathcal{P}$. The level-k-class $C_v^{(k)}(\mathcal{P})$ of v is the set of vertices of $\mathcal{P}$ that have the same level-k-label as v, including v itself. Formally $C_v^{(k)}(\mathcal{P}) := \left\{ u \in V \,\middle|\, l^{(k)}(u) = l^{(k)}(v) \right\}$. The class $C_v(\mathcal{P})$ of v is defined to be the set of vertices that have the same level-j-label for all j: $C_v(\mathcal{P}) := \left\{ u \in V \,\middle|\, \forall j \geq 1 : l^{(j)}(u) = l^{(j)}(v) \right\}$. We drop the argument $\mathcal{P}$ whenever the polygon is clear from the context.*

The following lemma summarizes the key properties of classes. In particular it states that a finite exploration depth is sufficient for fully characterizing the class of a vertex.

Lemma 2. *Let v be a vertex of a simple polygon $\mathcal{P}$ of size n. For all $k \geq 1$ we have the following properties:*

1: $C_v^{(k+1)} \subseteq C_v^{(k)}$; 2: $C_v \subseteq C_v^{(k)}$; 3: $C_v^{(n-1)} \subseteq C_v^{(k)}$; 4: $C_v^{(n-1)} = C_v$.
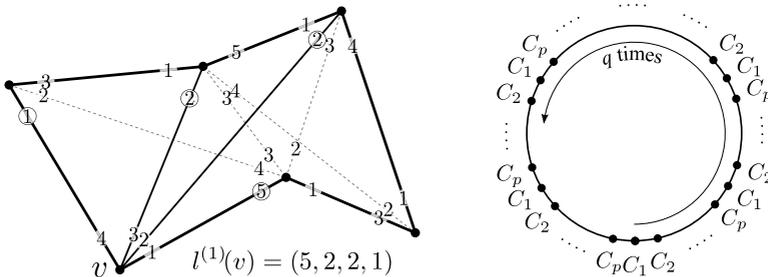
Fig. 1. Left: illustration of the definition of the level-1-label of a vertex v. Right: illustration of how classes repeat along the boundary of a highly symmetrical polygon.

Proof. Properties 1 and 2 follow immediately from the definitions of level-k-labels and classes. Property 3 was proven by Norris [8] for general graphs (not only visibility graphs) using the observation that if there is a vertex v such that for some $k > 1$ we have $C_v^{(k)} \neq C_v^{(k+1)}$, then there is also a vertex u for which $C_u^{(k-1)} \neq C_u^{(k)}$. Property 4 follows from Property 3 and the definition of C_v.   □

In [10] it was first shown that all classes have the same size q even in general graphs. The following lemma adds that in the case of polygons, the sequence of classes to which the vertices along the boundary belong is periodical with period $\frac{n}{q}$ (cf. Figure 1 (right)).

Lemma 3. *Let v_i be a vertex of a simple polygon $\mathcal{P}$ of size n. For all vertices $u \in V$ we have $q := |C_{v_i}| = |C_u|$ and $p := \frac{n}{q}$ is an integer equal to the number of different classes of $\mathcal{P}$. For all integers k we have $C_{v_i} = C_{v_{i+kp}}$.*

Proof. For every $v_j \in V$ we have $|C_{v_j}| = |C_{v_{j+1}}|$ as every vertex of C_{v_j} has a vertex of $C_{v_{j+1}}$ as its ccw neighbor on the boundary and conversely every vertex of $C_{v_{j+1}}$ has a vertex of C_{v_j} as its cw neighbor on the boundary. Since this is true for every vertex v_j, all classes must have the same size $q = |C_{v_i}|$ and thus $p = \frac{n}{q}$ is an integer and there are p classes. As the sequence $(C_{v_l}, C_{v_{l+1}}, \ldots, C_{v_{l-1}})$ is the same for all $v_l \in C_{v_i}$, the vertices of C_{v_i} need to be equally spaced in it. Using that an equal spacing is given by $n/|C_{v_i}| = p$, we get $C_{v_i} = C_{v_{i+kp}}$ for all integers k.   □

We now show that level-1-labels are enough to find the ears of a polygon.

Lemma 4. *Let v_i be a vertex of a simple polygon $\mathcal{P}$ of size $n \geq 3$. We have that v_i is an ear, if and only if $l_2(v_{i-1}) = -2$ or equivalently $l_{-2}(v_{i+1}) = 2$.*

Proof. If v_i is an ear, we obviously have $\text{vis}_2(v_{i-1}) = v_{i+1}$ and $\text{vis}_{-2}(v_{i+1}) = v_{i-1}$ which implies $l_2(v_{i-1}) = -2$ and $l_{-2}(v_{i+1}) = 2$.

Conversely $\text{vis}_2(v_{i-1}) = v_{i+1}$ obviously implies that v_i is an ear. As $l_2(v_{i-1}) = -2$ and $l_{-2}(v_{i+1}) = 2$ are equivalent, it remains to show that $l_2(v_{i-1}) = -2$ implies $\text{vis}_2(v_{i-1}) = v_{i+1}$.

For the sake of contradiction assume $w := \text{vis}_2(v_{i-1}) \neq v_{i+1}$. Consider the subpolygon $\mathcal{P}'$ induced by the ordered cycle chain(v_{i-1}, w). In $\mathcal{P}'$, v_{i-1} and w have degree 2. As $w \neq v_{i+1}$, we have $|\text{chain}(v_{i-1}, w)| \geq 4$ which contradicts Lemma 1. Therefore $\text{vis}_2(v_{i-1}) = v_{i+1}$ and v_i is an ear.   □

Lemma 5. *Let v_i be a vertex of a simple polygon $\mathcal{P}$. If v_i is an ear of $\mathcal{P}$, then every vertex in $C_{v_i}^{(2)}$ is an ear.*

Proof. By Lemma 4 we know that $l_2(v_{i-1}) = -2$ and $l_{-2}(v_{i+1}) = 2$. Let $v_j \in C_{v_i}^{(2)}$. Because $l^{(2)}(v_i) = l^{(2)}(v_j)$, we have $l(v_{i-1}) = l(v_{j-1})$ and $l(v_{i+1}) = l(v_{j+1})$ from which it follows that $l_2(v_{j-1}) = -2$ and $l_{-2}(v_{j+1}) = 2$. This however implies that v_j is an ear (again using Lemma 4).   □

The following lemma allows a robot to 'cut off' ears of the polygon, i.e. the operation that removes a vertex v_i yielding the subpolygon induced by the ordered cycle $v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{n-1}$ in $\mathcal{P}$'s visibility graph. Cutting off a single ear is problematic for a robot as it has no obvious way of deciding which edges of the visibility graph it has to ignore afterwards in order to restrict itself to the remaining polygon. An edge might lead to a vertex of the same class as the one the robot cut off, in which case it has no way of distinguishing whether the vertex is still there or not. Cutting off all vertices of one class however is possible as the robot can then simply ignore edges leading to vertices of the corresponding class altogether.

Lemma 6. *Let v be a vertex of a simple polygon $\mathcal{P}$ of size n with $|C_v| < n$, i.e. $\mathcal{P}$ has more than one class. If v is an ear of $\mathcal{P}$, the subpolygon $\mathcal{P}'$ of $\mathcal{P}$ obtained by cutting off the vertices $C_v(\mathcal{P})$ is well-defined and for all vertices u of $\mathcal{P}'$ we have $C_u(\mathcal{P}) \subseteq C_u(\mathcal{P}')$.*

Proof. As v is an ear, Lemma 5 gives us that all vertices in $C_v^{(2)}$ are ears and thus all vertices in $C_v \subseteq C_v^{(2)}$ are (Lemma 2). The subpolygon $\mathcal{P}'$ is thus well-defined as the inducing set of vertices lies on an ordered cycle.

Let u, w be vertices of $\mathcal{P}'$ such that $C_u(\mathcal{P}) = C_w(\mathcal{P})$ and thus for all $k \geq 1$ we have $l^{(k)}(u) = l^{(k)}(w)$ in $\mathcal{P}$. The level-k-label of a vertex v' in $\mathcal{P}'$ can be obtained from its level-k-label in $\mathcal{P}$ by recursively removing all occurrences of labels that belong to a vertex in $C_v(\mathcal{P})$ (i.e. we remove the level-$(k-1)$-labels belonging to vertices in $C_v(\mathcal{P})$ from $l^{(k)}(v')$ and apply the procedure recursively for $k-1$ to the other labels in $l^{(k)}(v')$). As $C_{v'}(\mathcal{P}) = C_{v'}^{(n-1)}(\mathcal{P})$, i.e. a finite depth determines the class of a vertex, the same occurrences are removed for vertices in the same class. Thus in $\mathcal{P}'$, $l^{(k)}(u) = l^{(k)}(w)$ holds for all $k \geq 1$ and hence $C_u(\mathcal{P}') = C_w(\mathcal{P}')$. □

The following result is the main insight for reconstructing the visibility graph of a polygon and for solving the weak rendezvous problem.

Theorem 1. *For any simple polygon $\mathcal{P}$ there is a vertex v for which $C_v^{(n-1)} = C_v$ forms a clique in the visibility graph of $\mathcal{P}$.*

Proof. Using the fact that any simple polygon has an ear, we can select an ear u of our polygon and cut off all vertices in C_u. Lemma 6 gives us that we can do this and we never divide classes in the process. We can repeatedly apply this procedure until we obtain a polygon $\mathcal{P}'$ where every vertex is in the same class. As $\mathcal{P}'$ needs to have at least one ear, it follows from Lemma 5 that all vertices of $\mathcal{P}'$ are ears. This in turn implies that $\mathcal{P}'$ is convex and thus its visibility graph is a complete graph. As we never divide classes in our procedure, we know that there is a vertex v of $\mathcal{P}$ such that every vertex of $C_v(\mathcal{P})$ is still present in $\mathcal{P}'$. Hence, $C_v(\mathcal{P})$ forms a clique in $\mathcal{P}'$ and thus forms a clique in $\mathcal{P}$. □

Lemma 7. *Let $\mathcal{P}$ be a simple polygon of n vertices. Given n and some $k \geq 1$, an LB-robot located at v can determine $l^{(k)}(v)$.*

Proof. The LB-robot can compute $l^{(1)}(v)$ by moving to every neighbor of v in turn. It can recursively compute $l^{(k)}(v)$ by moving to every neighbor of v in turn and computing its level-$(k-1)$-label. In both cases, as the robot can look-back, it is capable of returning to v after visiting a neighbor.   □

Using Lemma 7 and Theorem 1, we show that any number of deterministic LB-robots can position themselves such that they are mutually visible.

Theorem 2. *Let $\mathcal{P}$ be a simple polygon of n vertices. Given n, any number of identical and deterministic LB-robots can weakly meet in $\mathcal{P}$, i.e. they can position themselves such that every robot sees all other robots.*

Proof. By Lemma 7 a robot can calculate $l^{(n)}(v)$ for every vertex v along the boundary and thus not only find the classes of all vertices but also the classes of all vertices they see. By Theorem 1 at least one class forms a clique. The robot can easily check which classes form a clique (by comparing the level-n-labels) and drive to a vertex of the class $C_{\min}$ with the lexicographically smallest level-n-label among all these classes. This strategy will choose the same class $C_{\min}$ for every robot independent of its starting location, thus if all robots execute it they will eventually all be located on vertices of $C_{\min}$ seeing each other as $C_{\min}$ forms a clique.   □

In the following we again use the fact that an LB-robot can find a class of vertices that forms a clique, and we show that using this clique as a frame, it can incrementally build up the visibility graph of the polygon.

Theorem 3. *Let $\mathcal{P}$ be a simple polygon of n vertices. Given n, an LB-robot can determine the visibility graph of $\mathcal{P}$.*

Proof. By Lemma 7 the robot can determine $l^{(n)}(v)$ for all vertices v along the boundary and thus not only find the classes of all vertices but also the classes of all vertices they see. As in the proof of Theorem 2, this means the robot can identify the class $C_{\min}$ with the lexicographically smallest level-n-label among the classes forming a clique.

Let $u \in C_{\min}$ be a vertex of this class. We argue that the edges of u in the visibility graph of $\mathcal{P}$ are easily identified. Assume that the i-th edge of u (in ccw order) leads to a vertex w of class C' and let x_i be the number of edges with index $j < i$ that lead to vertices of class $C_{\min}$. If $C' = C_{\min}$, w is easily identified as the $(x_i + 1)$-th vertex of class $C_{\min}$ counting along the boundary starting at u. If $C' \neq C_{\min}$, w is the first vertex of class C' after the x_i-th vertex of class $C_{\min}$ in ccw order counting from u (by Lemma 3, there is exactly one vertex of class C' between the x_i-th and the $(x_i + 1)$-th vertex of class $C_{\min}$). In the following we show that the robot can identify the edges to other vertices ($u \notin C_{\min}$).

Let v_i be a vertex of $\mathcal{P}$. With $d_k(v_i)$ and $d_{-k}(v_i)$ we denote the set of vertices in $\text{chain}_{v_i}(v_i, v_{i+k})$ and $\text{chain}_{v_i}(v_{i-k}, v_i)$, respectively. Finding the visibility graph is the same as finding $d_{\frac{n}{2}}(v)$ and $d_{-\frac{n}{2}}(v)$ (i.e. the global indices of the vertices in $d_{\frac{n}{2}}(v)$, $d_{-\frac{n}{2}}(v)$) of every vertex v along the boundary. In the following we assume that ($\star$) for two vertices u, w, $C_u = C_w$ implies $|d_k(u)| = |d_k(w)|$ and

$|d_{-k}(u)| = |d_{-k}(w)|$ for all $k \geq 1$. We will prove $(\star)$ later, but for now observe that it trivially holds for $k = 1$.

We show inductively how to obtain $d_{\pm\frac{n}{2}}$ for all vertices. The robot knows $d_{\pm 1}$ for every vertex. It remains to be shown how to obtain $d_{k+1}(v_i)$ for some vertex v_i assuming $d_k(v_i)$ is known – this can then be applied to all vertices in order to obtain $d_{\pm(k+1)}$. Let $x := |d_k(v_i)|$ and $\text{vis}(v_i) = (u_1, \dots, u_d)$, where d is v_i's degree. We have

$$d_{k+1}(v_i) = \begin{cases} d_k(v_i) \cup \{v_{i+k+1}\}, & \text{if } u_{x+1} = v_{i+k+1} \\ d_k(v_i), & \text{otherwise.} \end{cases}$$

Let $v_j := u_{x+1}$ be the first vertex (in ccw order) visible to v_i and not in $d_k(v_i)$. In order to derive $d_{k+1}(v_i)$, it is now enough for the robot to decide whether $v_j = v_{i+k+1}$ or $v_j \neq v_{i+k+1}$. As the robot knows $C_{v_{i+k+1}}$ and can compute C_{v_j}, this is trivial for $v_{i+k+1} \notin C_{v_j}$. We therefore restrict ourselves to the case $v_{i+k+1} \in C_{v_j}$. We then have $y := |d_{-k}(v_{i+k+1})| = |d_{-k}(v_j)|$, using $(\star)$. Assuming $v_j = v_{i+k+1}$ immediately leads to $l_{x+1}(v_i) = -(y+1)$. We want to show that $l_{x+1}(v_i) = -(y+1)$ if and only if $v_j = v_{i+k+1}$. For the sake of contradiction assume $v_j \neq v_{i+k+1}$ and $l_{x+1}(v_i) = -(y+1)$.

Let $a \in C_{\min}$ be the first vertex in $\text{chain}(v_i, v_j)$ lying in $C_{\min}$ and likewise let $b \in C_{\min}$ be the last vertex in $\text{chain}(v_i, v_j)$ lying in $C_{\min}$. Note that a and b are well defined as $v_{i+k+1} \in \text{chain}(v_i, v_j)$ and there is a vertex of $C_{\min}$ between v_{i+k+1} and v_j as $C_{v_{i+k+1}} = C_{v_j}$ (Lemma 3). Consider the case $a \neq b$ (cf. Figure 2 (left)). We define s to be the last vertex in $\text{chain}(v_{i+1}, a)$ visible to v_i and t to be the first vertex in $\text{chain}(b, v_{j-1})$ visible to v_j. Let $\mathcal{P}'$ be the subpolygon induced by $v_i, \text{chain}(s, a), \text{chain}(b, t), v_j$. This subpolygon is well defined since a sees b as $C_{\min}$ forms a clique. Note that v_i does not see any vertices in $\text{chain}(b, v_{j-1})$, likewise v_j does not see any vertices in $\text{chain}(v_{i+1}, a)$ (recall that $v_j = u_{x+1}$ and $l_{x+1}(v_i) = -(y+1)$). In $\mathcal{P}'$, v_i and v_j are neighbors on the boundary and both have degree 2 which is a contradiction with Lemma 1. We may thus assume $a = b$ (cf. Figure 2 (right)). As there has to be a vertex of class $C_{\min}$ in $\text{chain}(v_{i+k+1}, v_j)$, there can be none in $\text{chain}(v_i, v_{i+k+1})$ (and thus none in $\text{chain}(v_{j-k-1}, v_j)$ as $v_{j-k-1} \in C_{v_i}$). This however means that $k < p - 2$ with $p = n/|C_{v_0}|$ from Lemma 3, and thus $\text{chain}(v_i, v_{i+k})$ and $\text{chain}(v_{j-k}, v_j)$ do not overlap ($|\text{chain}(v_{i+k+1}, v_j)| \geq p + 1$ as $v_{i+k+1} \in C_{v_j}$). Let now s be the last vertex in $\text{chain}(v_{i+1}, v_{i+k})$ visible to v_i and t be the first vertex in $\text{chain}(v_{j-k}, v_{j-1})$ visible to v_j. We can then define the subpolygon $\mathcal{P}'$ induced by $v_i, \text{chain}(s, t), v_j$ in which again v_i and v_j are neighbors of both degree 2, which is a contradiction to Lemma 1.

We have seen that $l_{x+1}(v_i) = -(y+1)$ is necessary and sufficient for $v_j = v_{i+k+1}$. Both x and y as well as level-1-labels can be derived from $l^{(n)}(v_i)$. This proves that our robot can compute the visibility graph inductively. To complete the proof we still need to show $(\star)$. Obviously we have $(\star)$ for $k = 1$. By inspection of the inductive method above for $k > 1$, we see that whenever we conclude $v_j = v_{i+k+1}$ for some vertex v_i, the same conclusion will be made for any other vertex $v_l \in C_{v_i}$. In step k of the induction, we only use $(\star)$ for $d_{\pm k}$. Hence $(\star)$ is maintained throughout. $\square$
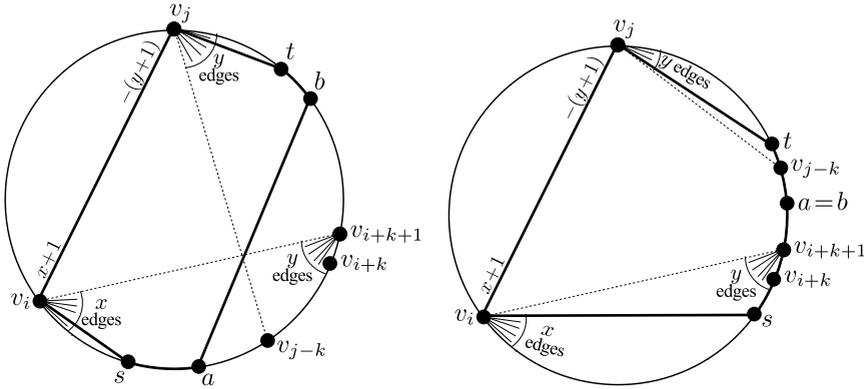
Fig. 2. Illustration of the visibility graph for the cases $a \neq b$ (left) and $a = b$ (right). The outline of the subpolygon causing a contradiction is highlighted.

The following theorem implies that the results of Theorems 2 and 3 still hold if only an upper bound $N \geq n$ on the number of vertices is given.

Theorem 4. *Let $\mathcal{P}$ be a simple polygon of n vertices. Given a bound $N \geq n$, an LB-robot can determine the number of vertices n.*

Proof. By Lemma 7, the robot can determine the sequence of level-N-labels $l^{(N)}(v_i), l^{(N)}(v_{i+1}), \ldots, l^{(N)}(v_{i+N-1})$, where v_i is its initial location. By Lemma 2, we have $C_{v_j}^{(N-1)} = C_{v_j}$ and thus the robot knows the classes of the vertices $v_i, v_{i+1}, \ldots, v_{i+N-1}$ as well as the classes of the vertices they see. Let $q = |C_{v_i}|$ and $p = \frac{n}{q}$ as in Lemma 3. The robot can determine p as the number of different classes it encounters since $N \geq n$. Once the robot knows q, it can immediately infer $n = p \cdot q$. By Theorem 1 there is a class $C^\star$ that forms a clique in the visibility graph. Let $M_i := \text{vis}(v_i) \cap C_{v_i}$. As $N \geq n$, there is a vertex of $C^\star$ among $v_i, v_{i+1}, \ldots, v_{i+N-1}$ and hence $q = \max_{i \leq j \leq i+N-1} |M_i| + 1$. As the robot knows the classes of the vertices $v_i, v_{i+1}, \ldots, v_{i+N-1}$ as well as the classes of the vertices they see, it can infer $M_i, M_{i+1}, \ldots, M_{i+N-1}$ and thus q. □

References

1. Ando, H., Oasa, Y., Suzuki, I., Yamashita, M.: Distributed memoryless point convergence algorithm for mobile robots with limited visibility. IEEE Transactions on Robotics and Automation 15(5), 818–828 (1999)
2. Bilò, D., Disser, Y., Mihalák, M., Suri, S., Vicari, E., Widmayer, P.: Reconstructing visibility graphs with simple robots. In: Proceedings of the 16th International Colloquium on Structural Information and Communication Complexity, pp. 87–99 (2009)
3. Brunner, J., Mihalák, M., Suri, S., Vicari, E., Widmayer, P.: Simple robots in polygonal environments: A hierarchy. In: Fekete, S.P. (ed.) ALGOSENSORS 2008. LNCS, vol. 5389, pp. 111–124. Springer, Heidelberg (2008)

4. Chalopin, J., Godard, E., Métivier, Y., Ossamy, R.: Mobile agent algorithms versus message passing algorithms. In: Shvartsman, M.M.A.A. (ed.) OPODIS 2006. LNCS, vol. 4305, pp. 187–201. Springer, Heidelberg (2006)
5. Cohen, R., Peleg, D.: Convergence of autonomous mobile robots with inaccurate sensors and movements. SIAM Journal of Computing 38(1), 276–302 (2008)
6. Ganguli, A., Cortés, J., Bullo, F.: Distributed deployment of asynchronous guards in art galleries. In: Proceedings of the 2006 American Control Conference, pp. 1416–1421 (2006)
7. Ghosh, S.K.: Visibility Algorithms in the Plane. Cambridge University Press, Cambridge (2007)
8. Norris, N.: Universal covers of graphs: isomorphism to depth $n-1$ implies isomorphism to all depths. Discrete Applied Mathematics 56(1), 61–74 (1995)
9. Suri, S., Vicari, E., Widmayer, P.: Simple robots with minimal sensing: From local visibility to global geometry. International Journal of Robotics Research 27(9), 1055–1067 (2008)
10. Yamashita, M., Kameda, T.: Computing on anonymous networks: Part I – characterizing the solvable cases. IEEE Transactions on Parallel and Distributed Systems 7(1), 69–89 (1996)
11. Yershova, A., Tovar, B., Ghrist, R., LaValle, S.M.: Bitbots: Simple robots solving complex tasks. In: Proceedings of the 20th National Conference on Artificial intelligence, vol. 3, pp. 1336–1341 (2005)