Mapping simple polygons: The power of telling convex from reflex

JÉRÉMIE CHALOPIN and SHANTANU DAS, LIF, CNRS & Aix-Marseille University
YANN DISSER, MATÚŠ MIHALÁK and PETER WIDMAYER, ETH Zurich

We consider the exploration of a simple polygon $\mathcal{P}$ by a robot that moves from vertex to vertex along edges of the visibility graph of $\mathcal{P}$. The visibility graph has a vertex for every vertex of $\mathcal{P}$ and an edge between two vertices if they see each other, i.e., if the line segment connecting them lies inside $\mathcal{P}$ entirely. While located at a vertex, the robot is capable of ordering the vertices it sees in counter-clockwise order as they appear on the boundary, and for every two such vertices, it can distinguish whether the angle between them is convex ($\leq \pi$) or reflex ($> \pi$). Other than that, distant vertices are indistinguishable to the robot. We assume that an upper bound on the number of vertices is known.

We obtain the general result that a robot exploring any locally oriented, arc-labelled graph G can always determine the *base graph* of G. Roughly speaking this is the smallest graph that cannot be distinguished by a robot from G by its observations alone, no matter how it moves. Combining this result with various other techniques allows to show that a robot exploring a polygon $\mathcal{P}$ with the above capabilities is always capable of reconstructing the visibility graph of $\mathcal{P}$. We also show that multiple identical, indistinguishable and deterministic robots of this kind can always solve the weak rendezvous problem in which they need to position themselves such that they mutually see each other, i.e., such that they form a clique in the visibility graph.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Polygons, robots, mapping, visibility graphs, rendezvous, angles

ACM Reference Format:
Jérémie Chalopin, Shantanu Das, Yann Disser, Matúš Mihalák, and Peter Widmayer, 2013. Mapping simple polygons: The power of telling convex from reflex. *ACM* V, N, Article A (January YYYY), 15 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Autonomous mobile robots are used for various tasks like cleaning, guarding, data retrieval, etc. in unknown environments. Many tasks require coordination of the robots [Agmon and Peleg 2007; Lin et al. 2007a; 2007b; Suzuki and Yamashita 1999] and exploration of the environment [Das et al. 2007; Katsev et al. 2011; Panaite and Pelc 1999; Suri et al. 2008]. We mainly focus on the latter problem, more precisely on the problem of mapping unknown

A preliminary version of this paper appeared in the proceedings of the 28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011).
Author's addresses: J. Chalopin and S. Das, LIF, CNRS & Aix-Marseille University; Y. Disser, (Current address) Department of Mathematics, TU Berlin; M. Mihalák and P. Widmayer, Institute of Theoretical Computer Science, ETH Zurich.

DOI:http://dx.doi.org/10.1145/0000000.0000000

Table I. Summary of known results for visibility graph reconstruction as well as open problems.

	initial		results		
sensors	info	movement	solvable	time	source
angles	n	boundary	yes	poly	[Disser et al. 2011]
angles	–	boundary	yes	poly	[Disser et al. 2013]
cvv, boundary angles	n	boundary	no		[Bilò et al. 2012]
angle types		boundary	*open*		
distances		boundary	*open*		
pebble	–	free	yes	poly	[Suri et al. 2008]
cvv, look-back	–	free	no		[Brunner et al. 2008]
look-back	$\bar{n}$	free	yes	poly	[Chalopin et al. 2011]
angle types, look-back	–	free	yes	poly	[Bilò et al. 2012]
angle types, directions	–	free	yes	poly	[Bilò et al. 2012]
directions	$\bar{n}$	free	yes*	exp	[Disser, Ghosh et al. 2013]
angle types	$\bar{n}$	**free**	**yes**	**exp**	**this paper**
angle types	–	free	*open*		
distances		free	*open*		
no sensors	n	free	*open*		

*This result holds even for polygons with holes.

polygons. However, we also show how a map of the environment can facilitate coordination and allow robots to solve the weak rendezvous problem. The difficulty of the mapping problem depends on the characteristics of the environment itself and on the sophistication of the robots, i.e., on their sensory and locomotive capabilities. A natural question is how much sophistication a robot needs to be able to solve the problem. The ultimate goal is to characterize the difficulty of the mapping problem by finding minimal robot configurations that allow a robot to create a map.

We consider robots operating in environments in the shape of simple polygons. For many tasks, instead of inferring a detailed map of the geometry of the environment, it is enough to obtain the visibility graph. The visibility graph has a node for each vertex of the polygon and an edge connecting two nodes if the corresponding vertices *see each other*, i.e., if the straight-line segment between them is contained in the polygon. The goal in this context becomes to find minimal robot models that allow a robot inside a polygonal environment to reconstruct the visibility graph of the environment. The information the robot can gather must be sufficient to *uniquely* infer the visibility graph.

A variety of minimalistic robot models have been studied, focusing on different types of environments and objectives [Ando et al. 1999; Brunner et al. 2008; Cohen and Peleg 2008; Ganguli et al. 2006; Katsev et al. 2011]. The model considered here originates from [Suri et al. 2008]. Roughly speaking, our robot is allowed to move along edges of the visibility graph. While at a vertex, the robot sees the vertices visible from its current location in counter-clockwise (ccw) order starting with its ccw neighbor along the boundary. Apart from this ordering the vertices are indistinguishable to the robot. In each move the robot may select one of them and move to it. The robot has no way of *looking back*, i.e., it has no immediate way of knowing which vertex it came from among the vertices it sees now. In this paper, the robot is assumed to be aware of an upper bound $\bar{n}$ on the number of vertices n.

Table 1 summarizes known results that are based on the robot model we use in this paper, as well as open problems. Besides employing different sensors, the results differ in the robot's initial knowledge about n, its movement capabilities, and, in case of positive results, the running times of the reconstruction algorithms. The first part of the table concerns robots that are restricted to moving along the boundary only. It was shown that even with this severe movement restriction robots can still reconstruct the visibility graph, as long as they can measure the exact angle between any pair of visible vertices [Disser et al. 2011; 2013]. On the other hand, only measuring the angle between the two neighboring vertices along the boundary is not sufficient, even if the robot can distinguish whether any two visible vertices are neighbors along the boundary ("cvv" in the table) [Bilò et al. 2012].

For robots that move across the polygon (as opposed to along the boundary), it is sufficient to be able to mark a single vertex (e.g., with a pebble) in order to reconstruct the visibility graph [Suri et al. 2008]. Without this powerful ability, not being capable to look back makes it difficult for the robot to relate the information it collected so far to subsequent observations. But even with a look-back sensor that allows to identify the vertex the robot came from in its last move, some knowledge of n is required to solve the reconstruction problem [Brunner et al. 2008; Chalopin et al. 2011]. A direction sensor that measures the angle between the boundary and a global reference direction makes it possible to reconstruct the visibility graph, even in the presence of holes [Disser, Ghosh et al. 2013]. In this paper, we equip the robot with an angle-type sensor that distinguishes convex ($\leq \pi$) from reflex ($> \pi$) angles. It was shown before that, even without knowledge of n, this sensor is powerful enough to allow visibility graph reconstruction, as long as it is combined with a look-back sensor or a direction sensor [Bilò et al. 2012]. In this paper, we show that these extra sensors are not required, provided that the robot knows at least a bound on the number of vertices. It remains an open problem whether the angle-type sensor is sufficient even when the robot is restricted to moving along the boundary. Other interesting open problems are whether knowledge of n on its own is already enough to reconstruct the visibility graph, and how a distance sensor may be used for reconstruction.

In contrast to most previous results our algorithm requires exponential time. This is due to the difficulty of collecting all of the data available via the agent's sensors. In settings where movement is restricted to be along the boundary, collecting the data essentially requires to visit each vertex once, which allows the corresponding reconstruction algorithms to be very efficient. Similarly, the agent can systematically collect all data if it has a way to retrace movements (e.g. via look-back sensor). In our setting, it is much more difficult to relate observations made at different vertices, which essentially forces us to try all possible ways of stitching together the collected data. The crucial point is that we can at all determine the correct relation between the observations, even if this requires exponential time.

In the robot model we use, robots move along edges of the visibility graph and can locally access some information about the edges. We can model this in the context of general robotic exploration of edge-labeled graphs, where the edge-labeling is usually restricted to be locally bijective at every vertex (i.e., no two edges incident to the same vertex have the same label). In this more general context, robots are aware of the degree of the vertex they are located at as well as of the labels of the edges incident to it. In every step, a robot selects an edge and moves to its other end. It is known that two distinct edge-labeled graphs (of the same size) can appear mutually indistinguishable to a robot, i.e., the reconstruction problem is not always solvable [Angluin 1980; Boldi and Vigna 2002]. The rendezvous problem is generally not solvable either [Chalopin et al. 2006; Yamashita and Kameda 1996]. In this paper we show that while a robot cannot solve the reconstruction problem in general graphs, it can always infer the *minimum base graph* of a (directed) graph G – the smallest graph among all graphs indistinguishable from G by a robot (a thorough discussion of base graphs and their properties can be found in [Boldi and Vigna 2002]). This result is of independent interest. We will see later that polygon exploration can be transformed to the exploration of a particular class of directed, arc-labeled graphs, where both the reconstruction problem as well as the weak-rendezvous problem become solvable.

As it is impossible to reconstruct general graphs, it is natural to ask how much information a robot can obtain about a graph. This information is encoded in the unique minimum base graph. In general, the mapping from a graph to its minimum base graph is not one-to-one in the sense that there are graphs which share the same minimum base graph. Our question whether a robot with certain capabilities can reconstruct the visibility graph of a polygon can be translated to whether the mapping is one-to-one for the class of visibility graphs with an appropriate labeling. As the main technical contribution of this paper we show that if the labeling locally encodes the convexity information about every angle at a vertex, this mapping becomes one-to-one. In other words, visibility graphs can always be reconstructed from their minimum base graph if the type of every angle (convex or reflex) is known. Combined with the result that a robot can reconstruct the minimum base graph even in general graphs when an upper bound on the number of vertices is given, this solves the reconstruction problem for visibility graphs in this setting.

2. THE VISIBILITY GRAPH RECONSTRUCTION PROBLEM

We consider the exploration of a (simple) polygon $\mathcal{P}$ by a robot that moves from vertex to vertex along straight lines in $\mathcal{P}$. Two vertices u, v that can be connected with a straight line inside $\mathcal{P}$ (possibly touching its boundary[1]) are said to *see each other*. We define the visibility graph $G_{\text{vis}} = (V, E)$ of $\mathcal{P}$ to be a directed graph, where V is the set of vertices of $\mathcal{P}$ and there is an arc from u to v (and vice versa) if u and v see each other. Whenever convenient, we identify G_{vis} with its canonical straight-line embedding in the polygon. For example, we speak of angles between arcs of G_{vis}, when we mean the angles between the corresponding line segments of its straight-line embedding.

Depending on the additional capabilities we equip a robot with, the robot might or might not be able to perform certain tasks. We focus on the *visibility graph reconstruction problem* in which the robot has to uniquely infer G_{vis}. Here and throughout this paper we consider isomorphic graphs to be the "same" graph, as we cannot hope to distinguish graphs further. We also consider the *weak-rendezvous problem* in which multiple identical and deterministic robots need to position themselves on vertices of the polygon that mutually see each other.

Before defining a specific robot model, we introduce some formalism for G_{vis} (cf. Figure 1). We fix a vertex v_0 and denote the vertices of $\mathcal{P}$ in ccw order along the boundary by $v_0, v_1, \ldots, v_{n-1}$. Note that $v_0, v_1, \ldots, v_{n-1}, v_0$ is a Hamiltonian cycle in G_{vis}. By $\text{chain}(v_l, v_r)$ we denote the sequence $(v_l, v_{l+1}, \ldots, v_r)$ and by $\text{chain}_v(v_l, v_r)$ we denote the subsequence of $\text{chain}(v_l, v_r)$ containing only the vertices visible to v. Here and throughout this paper all indices are understood modulo n. For $v_i \in V$ let $(u_1, \ldots, u_{d_i}) := \text{chain}_{v_i}(v_{i+1}, v_{i-1})$ be the vertices visible to v_i other than v_i. We say that d_i is the degree of v_i and define $\text{vis}_{v_i}(x) := \text{vis}_{v_i}(-(d_i + 1 - x)) := u_x$, to be the x-th vertex visible to v_i in ccw order or equivalently the $(d_i + 1 - x)$-th vertex visible to v_i in clockwise (cw) order for $1 \leq x \leq d_i$. Conversely, we set $O_{v_i}(u_x) := x$ or interchangeably $O_{v_i}(u_x) = -(d_i + 1 - x)$ for $1 \leq x \leq d_i$. For $1 \leq x < y \leq d_i$ we write $A_{v_i}(x, y) = A_{v_i}(y, x)$ to denote the ccw angle between the arcs (v_i, u_x) and (v_i, u_y) in that order. Furthermore, we define the *angle type* $T_{v_i}(\cdot, \cdot)$ as follows: $T_{v_i}(x, y) = T_{v_i}(y, x) = 1$ if $A_{v_i}(x, y) > \pi$ and $T_{v_i}(x, y) = T_{v_i}(y, x) = 0$ otherwise. For convenience we set $T_{v_i}(x, x) = 0$. A vertex v_i is called *reflex* if $T_{v_i}(1, d_i) = 1$ and *convex* otherwise.

The exploration of G_{vis} can be reduced to the general problem of exploring a strongly connected, directed and arc-labeled (multi-) graph[2] G (from now on we use the word "graph" to refer to such graphs). We write $\lambda(e)$ to denote the label of an arc e. A robot exploring a graph is assumed to be aware of the labels of all the outgoing arcs at its location. In every move, the robot may choose one of those arcs and follow it to its target. A directed walk is a sequence of arcs $(e_1, e_2, \ldots)$ such that the target of e_1 is the source of e_2 and so on. A directed path is a directed walk which does not visit any vertex more than once. Every walk $p = (e_1, e_2, \ldots)$ in the graph uniquely induces a label-sequence $\lambda(p) = (\lambda(e_1), \lambda(e_2), \ldots)$. Conversely, any label-sequence Λ induces a set of walks $\Lambda(G)$ such that $\lambda(p) = \Lambda$ for all $p \in \Lambda(G)$. By $\Lambda(v)$ we denote the set of walks in $\Lambda(G)$ that start at v. If no two outgoing arcs

[1] Note that we do not make the usual assumption that vertices of the polygon are in general position.

[2] For the most part of the paper it might help to think of symmetric graphs, i.e., graphs where for every directed edge (v, u) there is also a reverse edge (u, v).

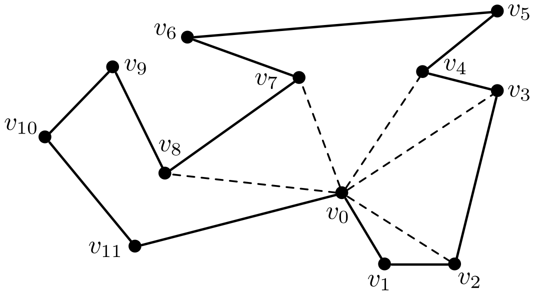

Fig. 1. In the example above we have $\text{chain}(v_3, v_{10}) = (v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10})$ while $\text{chain}_{v_0}(v_3, v_{10}) = (v_3, v_4, v_7, v_8)$. Similarly, $\text{chain}(v_{10}, v_3) = (v_{10}, v_{11}, v_0, v_1, v_2, v_3)$ and $\text{chain}_{v_0}(v_{10}, v_3) = (v_{11}, v_0, v_1, v_2, v_3)$. We have $\text{vis}_{v_0}(5) = \text{vis}_{v_0}(-3) = v_7$ and conversely $O_{v_0}(v_7) = 5$ or -3 because v_7 is the fifth vertex that v_0 sees in ccw order and the third in cw order. Finally, we have $T_{v_0}(2, 5) = 0$ and $T_{v_0}(6, 2) = 1$, since v_2 and v_7 form a convex angle at v_0 while v_2 and v_8 form a reflex angle.

of the same vertex share a label, we say the graph has a *local orientation* or is *locally oriented*. Then, for every label-sequence Λ and vertex v we have $\Lambda(v) = \emptyset$ or $|\Lambda(v)| = 1$; in the latter case we write $\Lambda(v)$ to denote this unique walk. With "$\circ$" we denote the concatenation of label-sequences.

We can now introduce the robot model used in this paper in detail. As described above, a robot is allowed to move along arcs of the visibility graph. In addition, while situated at a vertex v of degree d, the robot can order all outgoing arcs in ccw order starting with the arc to its ccw neighbor along the boundary, and it is aware of $T_v(x, y)$ for all $1 \leq x, y \leq d$. We assume the robot to know an upper bound $\bar{n} \geq n$ on the total number of vertices n. From now on, when we talk about a robot in a polygon, we refer to this robot model.

The exploration of $\mathcal{P}$ by a robot is in fact equivalent to the exploration of an arc-labeled version of G_{vis} that encodes the information available to the robot in its labeling. In this setting, upon entering a node u, the robot learns all the labels of arcs leaving u. The labeling needs to encode the local orientation and the angle type information into the labeling of the outgoing arcs at the corresponding vertex in G_{vis}. We introduce a labeling in which each label is a sequence of integers. Let v be a vertex of the visibility graph with degree d and (v, u) be an outgoing arc of v. We label (v, u) with the label $(x_0, x_1, \ldots, x_d)$, where $x_0 := O_v(u)$ and $x_i := T_v(x_0, i)$, $1 \leq i \leq d$. Note that by the definition of O_v our labeling is a local orientation. Further note that the arcs (v, u) and (u, v) may be labeled differently. It is immediate to check that a robot exploring the labeled graph G_{vis} encounters the exact same information as a robot inside the polygon if both start at the same vertex. It is thus sufficient to show that the labeled version of G_{vis} can be reconstructed in the framework of exploring general graphs in order to show that a robot can indeed solve the visibility graph reconstruction problem. From now on we write G_{vis} to denote the arc-labeled visibility graph.

3. OVERVIEW OF THE ALGORITHM

The visibility graph reconstruction algorithm that we design in this paper combines several old and new graph-theoretical and geometrical properties of visibility graphs as well as techniques developed in earlier studies. Rather than formally introducing all relevant concepts right away, this section aims to give an intuitive overview of the algorithm. We informally describe the underlying techniques and defer their formal discussion to later sections. Note that we are primarily interested in showing that a robot is at all capable of uniquely reconstructing the visibility graph of any simple polygon. The algorithm we provide as a proof does not need to be particularly efficient as long as it is guaranteed to terminate in finite time. An algorithm that solves the weak-rendezvous problem is obtained as a byproduct.

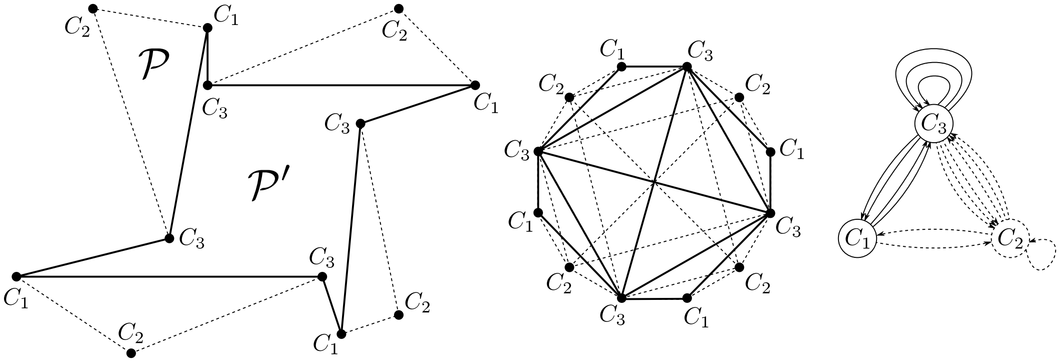

Fig. 2. Left: cutting away a class of vertices (ears) from $\mathcal{P}$ to obtain $\mathcal{P}'$. Middle: visibility graph G_{vis} of $\mathcal{P}$. Right: minimum base graph $G^{\star}_{\text{vis}}$ of G_{vis}. Dashed edges are in $\mathcal{P}$ but not in $\mathcal{P}'$.

In Section 2 we argued that the exploration of $\mathcal{P}$ by a robot is equivalent to the exploration of G_{vis} in the context of general graph exploration. In general and without any prior knowledge of the graph, there can be infinitely many graphs that are compatible with the observations of the robot no matter how far it moves, i.e., all these graphs are indistinguishable to the robot (consider for example the family of all cycles). However, it is known [Boldi and Vigna 2002] that for every graph G, there is always a unique graph $G^{\star}$ that is indistinguishable from G and has minimum size (for a bidirected cycle, $G^{\star}$ is a graph with one node and two (directed) self-loops). We say $G^{\star}$ is the *minimum base graph* of G. Using the fact that G_{vis} is locally oriented and that an upper bound $\bar{n}$ on n is known a priori, we are able to show the following result.

THEOREM 3.1. *A robot in $\mathcal{P}$ can determine $G^{\star}_{\text{vis}}$.*

The main ingredient for this theorem is the observation that given two candidate graphs for $G^{\star}_{\text{vis}}$, the robot can eliminate one of them in finite time by following an appropriate sequence of arc labels. It is then sufficient to iterate over pairs of graphs with size at most $\bar{n}$, discarding one of the two in every step. Once the robot has determined $G^{\star}_{\text{vis}}$, it has extracted all the information it can possibly gather by moving around. Subsequent steps of the algorithm can thus operate on $G^{\star}_{\text{vis}}$ directly without further need of moving at all in G_{vis}.

We associate each vertex of G_{vis} with a vertex of $G^{\star}_{\text{vis}}$ such that each vertex of $G^{\star}_{\text{vis}}$ represents a *class* of vertices of G_{vis}. For two vertices u, v of G_{vis} in the same class we have $\Lambda(u) \neq \emptyset \Leftrightarrow \Lambda(v) \neq \emptyset$ for all label-sequences Λ. Furthermore, the classes with which the vertices of G_{vis} are associated repeat periodically along the boundary of the polygon and in particular all classes have the same size. We define a unique order between the classes and use a procedure similar to the one in [Chalopin et al. 2011] to show that at least one of them forms a clique in G_{vis}. The idea is to repeatedly "cut off" *ears* of the polygon, i.e., vertices whose neighbors on the boundary see each other. Cutting off such an ear yields a subpolygon of $\mathcal{P}$ and we can repeat the process on the subpolygon. However, the robot cannot operate on G_{vis} directly as it only has access to $G^{\star}_{\text{vis}}$. The following lemma allows the robot to cut off an entire class of vertices at a time, an operation that can be performed in $G^{\star}_{\text{vis}}$ simply by deleting the corresponding vertex (and adjusting the arc labels of its neighboring vertices).

LEMMA 3.2. *Let v be an ear of $\mathcal{P}$. Every vertex in the same class as v is an ear of $\mathcal{P}$.*

As every polygon has at least one ear, the robot can thus "cut off" an entire class of $\mathcal{P}$ in order to obtain a new and smaller polygon $\mathcal{P}'$ (cf. Figure 2). By removing the corresponding vertex of $G^{\star}_{\text{vis}}$ and updating the arc labels, it obtains a graph $G'^{\star}_{\text{vis}}$ that is indistinguishable from the visibility graph of $\mathcal{P}'$. If this process is repeated, by always selecting for removal the smallest class with respect to a unique order relation, eventually a situation is reached in which only one (uniquely defined) class $C^{\star}$ remains. As the corresponding subpolygon must again have at least one ear, by the above lemma the entire class $C^{\star}$ consists of ears and the corresponding subpolygon thus is convex. A convex subpolygon is a clique in the original visibility graph and we may conclude the following crucial theorem.

THEOREM 3.3. *There exists a uniquely defined class $C^{\star}$ in G_{vis} whose vertices form a clique.*

The above yields an algorithm for multiple robots to weakly meet: As $C^{\star}$ is unique, every robot can determine $C^{\star}$ and then simply position itself on a vertex of $C^{\star}$. We get

THEOREM 3.4. *Any number of robots in $\mathcal{P}$ can solve the weak-rendezvous problem.*

Starting from the clique $C^{\star}$, we show that by sequentially "gluing" ears back to the polygon, a robot can extend the initial clique and reconstruct the entire visibility graph step by step. Every step relies on a recursive counting method that was introduced in [Bilò et al. 2012]. In order to know how to glue ears back on, the robot explicitly needs to construct $C^{\star}$ by repeatedly cutting off ears and remembering in which order the classes are cut off in the process.

THEOREM 3.5. *A robot in $\mathcal{P}$ can solve the visibility graph reconstruction problem.*

4. FINDING THE MINIMUM BASE GRAPH

This section focuses on the problem of exploring a general, finite, strongly connected, locally oriented directed (multi-) graph $G = (V, E)$ with a robot. Again, we assume an upper bound $\bar{n}$ on the number of vertices n to be known and we do not impose a limitation on the memory of the robot. We prove a generalization of Theorem 3.1 to general, locally oriented, directed graphs (cf. Theorem 4.2).

Before we define the notion of the *minimum base graph* $G^{\star}$ of G we need to introduce a few graph-theoretical concepts. First, given an arc e from vertex u to vertex v, we denote by $s(e)$ the *source* of arc e, i.e., the vertex u, and by $t(e)$ the *target* of arc e, i.e., the vertex v. Note that in the following we allow graphs to have parallel arcs between a pair of vertices. A *morphism* $\mu : G \to G'$ from G to a graph G' is a mapping from G to G' that maps vertices to vertices and arcs to arcs and maintains adjacencies and arc labels. More formally, if e is an arc in G from u to v then $s(\mu(e)) = \mu(u)$, $t(\mu(e)) = \mu(v)$, and $\lambda(e) = \lambda(\mu(e))$. An *opfibration* $\varphi : G \to \bar{G}$ with $\bar{G} = \left(\bar{V}, \bar{E}\right)$ is a morphism such that for every arc $\bar{e} \in \bar{E}$ with $\bar{u} = s(\bar{e})$ and for every vertex $u \in \varphi^{-1}(\bar{u})$ in the preimage of $\bar{u}$ there is a *unique* arc e with source $s(e) = u$ such that $\varphi(e) = \bar{e}$. We say that $\bar{G}$ is a *base graph* of G and G is a *total graph* of $\bar{G}$. Trivially, G is both its own base graph and total graph. If G has no base graph smaller than itself, we say G is *opfibration prime.* An *out-tree* is a graph that has a *root* vertex r such that there is exactly one directed walk from r to every other node.

We give the following properties without proof. For a detailed discussion and proofs, refer to [Boldi and Vigna 2002].

PROPERTY 1. *Let $\varphi : G \to \bar{G}$ be an opfibration. For every label-sequence Λ and every vertex $v \in V$ we have that $\Lambda(v) \neq \emptyset$ iff $\Lambda(\varphi(v)) \neq \emptyset$.*

PROPERTY 2. *There is exactly one opfibration prime base graph of G. We call it the* minimum base graph *of G and denote it by $G^{\star}$.*

PROPERTY 3. *For every $v \in V$, there is a unique (but not necessarily finite) total graph H_v of G that is an out-tree with root in $\varphi^{-1}(v)$, where φ is the opfibration mapping H_v to G. We call H_v the* universal total graph *of G at v.*

PROPERTY 4. *A graph is opfibration prime iff all its universal total graphs are distinct.*

PROPERTY 5. *Two different opfibration prime graphs have different sets of universal total graphs.*

We can now show that if we have a local orientation, there is a label-sequence of finite length that can be used to distinguish two rooted, opfibration prime graphs.

LEMMA 4.1. *Let $G_v = (V, E), G'_{v'} = (V', E')$ be two distinct, rooted, locally oriented opfibration prime graphs. There is a finite label-sequence Λ_{diff} for which $\Lambda_{\text{diff}}(v) \neq \emptyset$ and $\Lambda_{\text{diff}}(v') = \emptyset$ or vice versa.*

PROOF. First, consider the case that G_v and $G'_{v'}$ are the same graph, rooted at different vertices, i.e., $G = G'$. By Proposition 4, the universal total graphs H_v and $H_{v'}$ are distinct. Let r, r' be the roots of $H_v, H_{v'}$, respectively. Because H_v and $H_{v'}$ are distinct and locally oriented, there is a finite label-sequence Λ_{diff} with $\Lambda_{\text{diff}}(r) \neq \emptyset$ and $\Lambda_{\text{diff}}(r') = \emptyset$ or vice versa. By Property 1, this implies $\Lambda_{\text{diff}}(v) \neq \emptyset$ and $\Lambda_{\text{diff}}(v') = \emptyset$ or vice versa.

Now assume that $G \neq G'$. By Proposition 5, and without loss of generality, we may assume that there is a vertex $u \in V$, such that the universal total graph H_u of G at u is not a universal total graph of G'. Let Λ_1 be the label-sequence associated with the path from v to u in G. This path exists since G is strongly connected. If $\Lambda_1(v') = \emptyset$, we have found the desired label-sequence. Otherwise, let u' be the vertex at which the path $\Lambda_1(v')$ ends. By our choice of u, we have $H_u \neq H_{u'}$. By the same argument as above, there is a label-sequence Λ_2 with $\Lambda_2(u) \neq \emptyset$ and $\Lambda_2(u') = \emptyset$ or vice versa. But then, the concatenated label-sequence $\Lambda_1 \circ \Lambda_2$ has the desired property. □

The following theorem holds for directed graphs, i.e., for robots that cannot backtrack their moves. Similar results are known for undirected graphs where robots can identify the edge along which they reached their current location [Chalopin et al. 2011; Dereniowski and Pelc 2012; Yamashita and Kameda 1996].

THEOREM 4.2. *A robot exploring any finite, directed, strongly connected, locally oriented multi-graph G can determine $G^\star$ if it knows an upper bound $\bar{n}$ on the number of vertices of G.*

PROOF. In the following, we describe a strategy for finding $G^\star$ under the assumption that the maximum degree Δ of G is known, as well as the set $\mathcal{L}$ of all arc labels occurring in G. Otherwise, we start with initial guesses of Δ and $\mathcal{L}$ according to the observations of the agent at its starting location. Whenever the agent makes an observation that is inconsistent with its belief of Δ and $\mathcal{L}$, the agent updates its belief and restarts the whole procedure. As both Δ and $\mathcal{L}$ are finite, the number of times the agent needs to restart is finite.

Now, let v_{start} denote the vertex of G at which the robot is initially located, and let $v^\star_{\text{start}}$ denote the corresponding vertex of $G^\star$. By Property 2, $G^\star$ is unique. We will give an algorithm that maintains a finite set C of rooted graphs. This set is always guaranteed to contain $G^\star$ rooted at $v^\star_{\text{start}}$, provided that the belief of Δ and $\mathcal{L}$ is correct. We begin by setting C to contain all opfibration prime graphs of size at most $\bar{n}$ and maximum degree Δ, with arc-labelings from $\mathcal{L}$ and all possible roots.

While $|C| > 1$, we let $G^1_{v_1}, G^2_{v_2}$ be two rooted graphs in C and describe how to conclude that either $G^1_{v_1}$ or $G^2_{v_2}$ can safely be eliminated from C. Once $|C| = 1$, the only graph left will have to be $G^\star$. Observe that while the robot has not visited all nodes of G, more than one consistent base graph remains in C (for different outgoing arcs at the unvisited node, different labels and multiplicities). Once $|C| = 1$, the agent must thus have visited all nodes of G and hence have a correct belief of Δ and $\mathcal{L}$. In the following, let p_{hist} denote the walk in G the robot has travelled along so far during the execution of the algorithm, and let $\Lambda_{\text{hist}} = \lambda(p_{\text{hist}})$ be the associated label-sequence. Note that the robot is aware of Λ_{hist} but not of p_{hist}, since it does not know the graph G, nor v_{start}. For a rooted graph G'_v, we use $v_{\text{hyp}}(v)$ to denote the last vertex on the walk $\Lambda_{\text{hist}}(v)$ in G', i.e., the vertex the robot would currently be located at if it had started the algorithm at vertex v in graph G'.

Given two rooted, opfibration prime graphs $G^1_{v_1}, G^2_{v_2}$, we argue how to conclude that one of them cannot be $G^\star$ rooted at $v^\star_{\text{start}}$. First, we can check whether $\Lambda_{\text{hist}}(v_1) = \emptyset$ or $\Lambda_{\text{hist}}(v_2) = \emptyset$; if one of the two is the case, we can discard the corresponding rooted graph since it is not compatible with the observations the robot has made so far. Otherwise, we consider G^1 rooted at $v_{\text{hyp}}(v_1)$ and G_2 rooted at $v_{\text{hyp}}(v_2)$. By Lemma 4.1, there is a label-sequence Λ_{diff} of finite length, for which $\Lambda_{\text{diff}}(v_{\text{hyp}}(v_1)) \neq \emptyset$ and $\Lambda_{\text{diff}}(v_{\text{hyp}}(v_2)) = \emptyset$ or vice versa. The robot tries to move along a path corresponding to Λ_{diff}. If that turns out not to be possible because at some point no outgoing edge has the required label, it can discard $G^1_{v_1}$ from C if $\Lambda_{\text{diff}}(v_{\text{hyp}}(v_1)) \neq \emptyset$, and $G^2_{v_2}$ otherwise. If it successfully reached the end of Λ_{diff}, the robot can eliminate $G^1_{v_1}$ from C if $\Lambda_{\text{diff}}(v_1) = \emptyset$, and $G^2_{v_2}$ otherwise. □

We obtain Theorem 3.1 immediately by applying Theorem 4.2 to G_{vis}. Note that the results of this section are not restricted to visibility graphs.

5. IDENTIFYING THE CLIQUE $C^\star$

In this section we study structural properties of $G^\star_{\text{vis}} = (V^\star, E^\star)$ which we later use to show Theorem 3.3.

Let $\varphi : G_{\text{vis}} \to G^\star_{\text{vis}}$ be the opfibration from G_{vis} to $G^\star_{\text{vis}}$. As $G^\star_{\text{vis}}$ is the minimum base of G_{vis}, φ is unique. Every vertex $v^\star$ of $G^\star_{\text{vis}}$ corresponds to a set of vertices of G_{vis}. We write $C_{v^\star} := \varphi^{-1}(v^\star) \subseteq V$ and say $C_{v^\star}$ is the *class* of $v^\star$. For all $v \in \varphi^{-1}(v^\star)$, we set $C_v := C_{v^\star}$. From the definition of opfibrations and the minimality of the base graph it follows that every two vertices u, v of the same class C_u have the same degree d and that due to local orientation we have $C_{\text{vis}_u(i)} = C_{\text{vis}_v(i)}$ for all $1 \leq i \leq d$. We may thus write $C_u(i) := C_{\text{vis}_u(i)}$. Finally, we define $\mathcal{B} := \big(C_{v_0}, C_{v_1}, \ldots, C_{v_{n-1}}\big)$ to be the sequence in which the classes appear along the boundary.

As $G^\star_{\text{vis}}$ is opfibration prime, by Property 4 every vertex has its unique universal total graph. We use this and define a natural order $\mathcal{O}$ on the vertices of $G^\star_{\text{vis}}$ and thus on the classes of G_{vis}.

LEMMA 5.1. *The sequence $\mathcal{B}$ is periodical with period $|V^\star|$ and thus all classes have the same size.*

PROOF. The boundary can be traced by following the first edge at every vertex, i.e., the unique edge who's label starts with '1'. It follows that the image of the boundary under φ consists of $n/|V^\star|$ copies of a Hamiltonian cycle of $G^\star_{\text{vis}}$. Hence, $\mathcal{B}$ is periodical with period $|V^\star|$ and all classes have the size $n/|V^\star|$. □

We show that if a vertex from some class is an ear, then every vertex of the class is an ear. Recall that an ear of G_{vis} is a vertex $v_i \in V$ for which v_{i-1} and v_{i+1} see each other.

LEMMA 5.2. *Let $|V^\star| > 2$ and $v_x, v_y \in V$ such that $C_{v_x}(2) = C_{v_y}$ and $C_{v_y}(-2) = C_{v_x}$. Then, $C_{v_{x+2}} = C_{v_y}$ and every vertex in $C_{v_{x+1}}$ is an ear.*

PROOF. We first prove that for all $v_i \in V$ and $u = \text{vis}_{v_i}(2)$, we have that if $v_i = \text{vis}_u(-2)$, then $u = \text{vis}_{v_i}(2) = v_{i+2}$ and thus v_{i+1} is an ear. For the sake of contradiction assume for some $v_i \in V$ and $u = \text{vis}_{v_i}(2)$ we have $\text{vis}_u(-2) = v_i$ but $\text{vis}_{v_i}(2) \neq v_{i+2}$. Consider the subpolygon induced by $\text{chain}(v_i, \text{vis}_{v_i}(2))$. This subpolygon has size at least

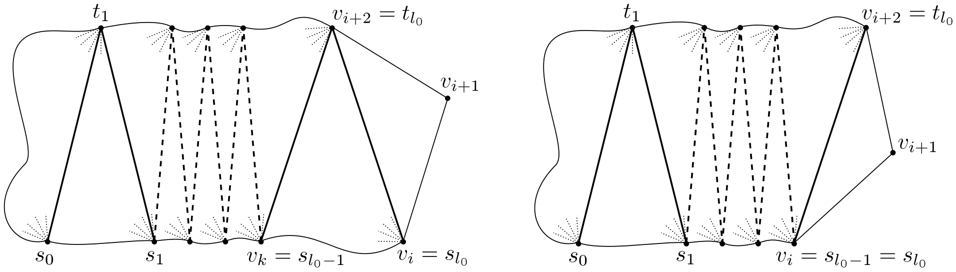

Fig. 3. Visualization of the "zig-zag" sequence Z. As Z does not self-intersect, there is a point l_0 from which on Z's entries do not change anymore. There are two cases how this point is reached: either s_{l_0-1} is distinct from s_{l_0} (left) or they are the same (right).

four as $\text{vis}_{v_i}(2) \notin \{v_{i+1}, v_{i+2}\}$. In the visibility graph of the subpolygon, v_i and $\text{vis}_{v_i}(2)$ are neighbors on the boundary and both have degree two, which is a contradiction to the fact that every polygon must admit a triangulation. Therefore $\text{vis}_{v_i}(2) = v_{i+2}$ and v_{i+1} is an ear as its neighbors on the boundary see each other.

Because of the above observation, it is sufficient to show that for every $v \in C_{v_x}$ we have $\text{vis}_u(-2) = v$, where $u := \text{vis}_v(2)$. For the sake of contradiction assume in the following that there is a vertex $s_0 \in C_{v_x}$ with $t_1 := \text{vis}_{s_0}(2)$ and $\text{vis}_{t_1}(-2) \neq s_0$.

We define an infinite sequence $Z = (s_0, t_1, s_1, t_2, \ldots)$ where $t_l := \text{vis}_{s_{l-1}}(2)$ and $s_l := \text{vis}_{t_l}(-2)$ for all $l > 0$. Obviously $s_l \in C_{v_x}, t_{l+1} \in C_{v_y}$ for all $l \geq 0$. Intuitively, Z is the zig-zag line obtained by alternately traveling along the first and the last non-boundary arc in ccw order, starting at s_0 (cf. Figure 3). It is immediate to see that for any fixed index $l' > 0$ we have $s_l, t_l \in \text{chain}(s_{l'}, t_{l'})$ for all $l \geq l'$. Hence the part of the boundary in which these vertices lie becomes smaller and smaller and from some index $l_0 > 0$ on we have $s_l = s_{l_0}$ and $t_l = t_{l_0}$ for all $l \geq l_0$ (we set l_0 to be the smallest such index). Let $0 \leq i, j < n$ be such that $v_i = s_{l_0}, v_j = t_{l_0}$. We then have $\text{vis}_{v_i}(2) = v_j$ and $\text{vis}_{v_j}(-2) = v_i$. Thus, by the above observation, v_{i+1} is an ear and $v_j = v_{i+2}$. As $v_i \in C_{v_x}$ and $v_j \in C_{v_y}$, this implies $C_{v_{x+2}} = C_{v_y}$. It remains to show that every vertex in $C_{v_{x+1}}$ is an ear.

We have to consider two cases. Either s_{l_0-1} is distinct from s_{l_0} or it is the same vertex (cf. Figure 3). We assume $s_{l_0-1} \neq s_{l_0}$ and omit the discussion of the second case which is essentially analogous (the same arguments hold for the other case if we switch the roles of s and t and reverse the boundary order). Let $0 \leq k < n$ such that $v_k = s_{l_0-1}$. As $\text{vis}_{v_k}(2) = v_{i+2}$, we have that v_k does not see any vertex in $\text{chain}(v_{k+2}, v_{i+1})$ (note that this chain is not empty as $v_k \neq v_i$) and thus as $v_{k+1} \in C_{v_{x+1}}$ is in the same class as (the ear) v_{i+1}, the interior angle of the polygon at v_{k+1} is strictly smaller than π. Since all vertices in $\text{chain}(v_{i+3}, v_k) \cup \text{chain}(v_{k+2}, v_{i+1})$ lie on the same side of the line through v_k and v_{i+2} and are separated by v_{i+2} (cf. Figure 4), no vertex in $\text{chain}(v_{i+3}, v_k)$ can see any vertex in $\text{chain}(v_{k+2}, v_{i+1})$. Let $X \subset C_{v_x}$ be the set of vertices of C_{v_x} in $\text{chain}(v_{i+3}, v_k)$ and let $Y \subset C_{v_y}$ be the set of vertices of C_{v_y} in $\text{chain}(v_{i+3}, v_k)$. As $|V^\star| > 2$, $C_{v_x}, C_{v_{x+1}}$, and $C_{v_{x+2}} = C_{v_y}$ are all different and thus X and Y are disjoint. Note that because $\mathcal{B}$ is periodical with period $|V^\star|$ (Lemma 5.1) we have $|X| = |Y| + 1$.

We define the (undirected) bipartite graph $B_{xy} = (C_{v_x} \cup C_{v_y}, E_{xy})$ with the edge-set $E_{xy} = \{\{u, v\} \in C_{v_x} \times C_{v_y} \,|\, (u, v) \in E\}$. In B_{xy} all vertices need to have the same degree d as $|C_{v_x}| = |C_{v_y}|$ and all vertices in either class have the same degree. We have $|X| = |Y| + 1$, we have that vertices in X can only have edges to vertices in $Y \cup \{v_{i+2}\}$ and that vertices in Y can only have edges to vertices in X. For all vertices to have the same degree, v_{i+2} cannot have any edges leading to $C_{v_x} \backslash X$. This is a contradiction to the fact that v_{i+2} sees v_i which is not in $\text{chain}(v_{i+3}, v_k)$ and thus not in X. □

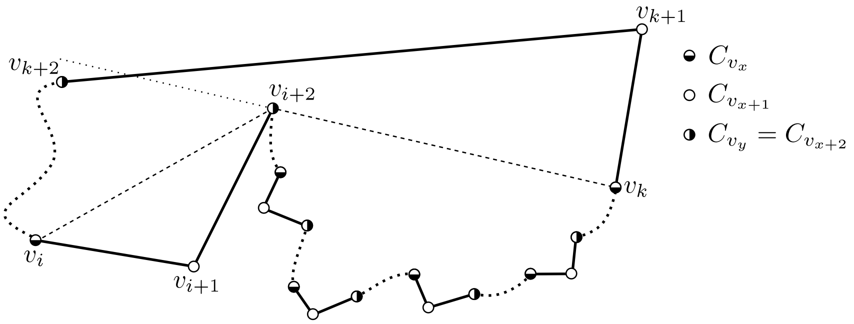

Fig. 4. No vertex in chain(v_{i+3}, v_k) can see any vertex in chain(v_{k+2}, v_{i+1}).

We can now consider arbitrary values of $|V^\star|$ and prove Lemma 3.2. We will need the following property of the shortest curve between two vertices of $\mathcal{P}$.

THEOREM 5.3 ([LEE AND PREPARATA 1984]). *Let $s, t \in V$. There is a unique shortest curve p from s to t that lies in $\mathcal{P}$. This curve is a chain of straight-line segments connected at reflex vertices of $\mathcal{P}$, and the two line segments at any vertex of p form a reflex angle. We say p is the* (euclidean) shortest path *in $\mathcal{P}$ between s and t.*

PROOF OF LEMMA 3.2. In the following we let $v_i \in V$ be an ear and show that all vertices in C_{v_i} are ears.

First consider the case $|V^\star| > 2$. As $(v_{i-1}, v_{i+1}) \in E$, we have $\text{vis}_{v_{i-1}}(2) = v_{i+1}$ and $\text{vis}_{v_{i+1}}(-2) = v_{i-1}$, and thus $C_{v_{i-1}}(2) = C_{v_{i+1}}$ and $C_{v_{i+1}}(-2) = C_{v_{i-1}}$. By Lemma 5.2 all vertices in C_{v_i} are ears. Now consider the case $|V^\star| = 1$. In that case, since v_i is convex, so are all vertices in C_{v_i}, as convexity is encoded in the arc-labeling. As $|V^\star| = 1$, this means that the polygon is convex and thus all vertices are ears.

It remains to consider the case $|V^\star| = 2$. Let $C_{v_j} \neq C_{v_i}$ be the second class in G_{vis}. Again, v_i is convex and thus all vertices in C_{v_i} are. For the sake of contradiction assume that there is a vertex $v_x \in C_{v_i}$ which is not an ear. Then v_{x-1} and v_{x+1} do not see each other, and by Lemma 5.1, $v_{x-1}, v_{x+1} \in C_{v_j}$. Let p be the shortest path in $\mathcal{P}$ between v_{x-1} and v_{x+1}. By Theorem 5.3, all vertices on p are reflex. This means that all vertices on p must be from C_{v_j} and thus all vertices of C_{v_j} must be reflex. Moreover, every vertex u in C_{v_j} has two neighbors u', u'' in C_{v_j} such that the angle between (u, u') and (u, u'') is reflex. If we cut off v_i from $\mathcal{P}$, we do not affect this property (every vertex u in C_{v_j} still has two neighbors from C_{v_j} forming a reflex angle) and we thus obtain a new polygon in which all vertices in C_{v_j} are still reflex (i.e., cutting off an ear cannot make a vertex of C_{v_j} convex). We can continue to obtain smaller and smaller subpolygons by selecting ears and cutting them off, maintaining the property that all vertices in C_{v_j} are reflex. Thus, in this process, we never cut off a vertex of C_{v_j}. This is a contradiction, as every polygon has at least one ear and thus the above process has to cut off all vertices eventually. □

PROOF OF THEOREM 3.3. Lemma 3.2 allows us to employ the following procedure repeatedly until only one class $C^\star$ remains: In step $i = 1, \ldots, |V^\star| - 1$, select the class $C^{(i)}$ which is smallest w.r.t. the order $\mathcal{O}$ among all classes of ears. We remove $C^{(i)}$ from the polygon by deleting the corresponding vertex from $G^\star_{\text{vis}}$ and updating the arc labels of its neighborhood accordingly. Removing class $C^{(i)}$ in that way produces a (not necessarily minimum) base graph $G^\star_i$ of the visibility graph $G^{(i)}_{\text{vis}}$ of the subpolygon $\mathcal{P}^{(i)}$ obtained by cutting off all ears in $C^{(i)}$. Since the minimum base graph of $G^{(i)}_{\text{vis}}$ is also a (minimum) base graph of $G^\star_i$, all vertices of G_{vis} corresponding to a single vertex of $G^\star_i$ (i.e., to a class of G_{vis}) fall into the same class of $G^{(i)}_{\text{vis}}$. As $\mathcal{P}^{(i)}$ has again at least one ear, Lemma 3.2 guarantees the existence of a class of $G_{\text{vis}}^{(i)}$ that contains only ears. Each of the remaining classes of G_{vis} contains vertices from a single class of $G_{\text{vis}}^{(i)}$, thus one of them contains only ears of $\mathcal{P}^{(i)}$. This allows us to repeat our procedure.

If we repeat the procedure $|V^{\star}|-1$ times, we are finally left with a single class $C^{(|V^{\star}|)} = C^{\star}$ and the sequence $(C^{(1)}, C^{(2)}, \ldots, C^{(|V^{\star}|-1)})$, which is fixed by our order relation $\mathcal{O}$. As $C^{\star}$ again corresponds to a subpolygon and thus must contain at least one ear, every vertex in $C^{\star}$ must be an ear. Therefore the corresponding subpolygon is convex and $C^{\star}$ forms a clique in G_{vis}. □

The existence of a clique gives us a way of computing n from $\bar{n}$ using $G_{\text{vis}}^{\star}$. By Lemma 5.1 we have $n = |V^{\star}| \cdot |C|$, where C is any class of G_{vis}. If we inspect the number of self-loops of every vertex of $G_{\text{vis}}^{\star}$, we are sure to encounter at least one vertex with $|C|-1$ self-loops, which corresponds to a clique in the visibility graph, and thus $|C|$ is equal to the maximum number of self-loops of any vertex plus one.

By Theorem 3.1, a robot can determine $G_{\text{vis}}^{\star}$ in finite time. Furthermore it can identify which of the classes consist of ears: If $|V^{\star}| \leq 2$, a class of convex vertices only contains ears, and for $|V^{\star}| > 2$ the robot can use Lemma 5.2. The robot can hence execute the above procedure explicitly and we obtain

THEOREM 5.4. *A robot in $\mathcal{P}$ can determine the lexicographically smallest sequence $\mathcal{C} = (C^{(1)}, C^{(2)}, \ldots, C^{(|V^{\star}|)})$, such that for every $1 \leq i \leq |V^{\star}|$, all vertices in $C^{(i)}$ are ears in the subpolygon obtained by removing the vertices in $\bigcup_{j=1}^{i-1} C^{(j)}$ from $\mathcal{P}$.*

6. RECONSTRUCTING G_{VIS}

In the following, we assume that $G_{\text{vis}}^{\star}$ and the sequence $\mathcal{C} = (C^{(1)}, C^{(2)}, \ldots, C^{(|V^{\star}|)})$ from Theorem 5.4 have already been determined. For all $1 \leq i \leq |V^{\star}|$ we denote by $G_{\text{vis}}^{(i)} = (V^{(i)}, E^{(i)})$ the subgraph of G_{vis} induced by $\bigcup_{j=i}^{|V^{\star}|} C^{(j)}$. By definition of $\mathcal{C}$, $G_{\text{vis}}^{(i)}$ is the visibility graph of a subpolygon of $\mathcal{P}$, and we denote this subpolygon by $\mathcal{P}^{(i)}$. As $C^{(|V^{\star}|)} = C^{\star}$, by Lemma 5.1 we have that $G_{\text{vis}}^{(|V^{\star}|)}$ is the complete graph on $n/|V^{\star}|$ vertices. We will show that $G_{\text{vis}}^{(i)}$ can be inferred from $G_{\text{vis}}^{(i+1)}$, suggesting a way to reconstruct $G_{\text{vis}} = G_{\text{vis}}^{(1)}$. First, we need one more lemma that was used before in [Bilò et al. 2012].

Definition 6.1. Let v_i, v_h be two vertices that do not see each other, and v_b be the first vertex other than v_i on the (euclidean) shortest path from v_i to v_h (and thus v_b is reflex). We say v_h *is hidden from* v_i *by* v_b.

LEMMA 6.2. *Let $G_{\text{vis}}^{(i+1)}$ be given, as well as a vertex $v_j \in V^{(i)}$, a vertex $v_y \in V^{(i+1)}$ visible to v_j in $\mathcal{P}$, and the index $b = O_{v_y}(v_j)$. It is possible to determine the number of vertices in $C^{(i)}$ hidden from v_j by v_y in $\mathcal{P}$.*

PROOF. By construction, two vertices of $\mathcal{P}^{(i)}$ see each other in $\mathcal{P}$ if and only if they see each other in $\mathcal{P}^{(i)}$. In particular, the shortest path in $\mathcal{P}$ between two vertices of $\mathcal{P}^{(i)}$ only bends at vertices of $\mathcal{P}^{(i+1)}$, since the vertices of $C^{(i)}$ are convex in $\mathcal{P}^{(i)}$. We can thus restrict ourselves to counting the number of vertices in $C^{(i)}$ hidden from v_j by v_y in $\mathcal{P}^{(i)}$, using the fact that no vertex in $C^{(i)}$ is hidden by another vertex of $C^{(i)}$.

We give an algorithm to count the number of vertices in $C^{(i)}$ hidden from v_j by v_y in $\mathcal{P}^{(i)}$ (cf. Figure 5). Let $H = H_{\text{direct}} \cup H_{\text{indirect}}$ be set of these vertices, where H_{direct} are the ones visible to v_y and H_{indirect} are the ones not visible to v_y. Since $b = O_{v_y}(v_j)$ is given, we know which arc of $G_{\text{vis}}^{\star}$ corresponds to (v_y, v_j). Hence, by inspecting $G_{\text{vis}}^{\star}$, we can obtain the arc-labels of all arcs at v_y in $G_{\text{vis}}^{(i)}$ that form a reflex angle with (v_y, v_j). We can infer

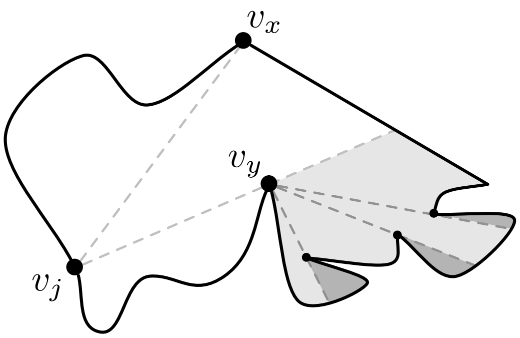

Fig. 5. We can count the vertices of $C^{(i)}$ hidden from v_j by v_y by counting the vertices of $C^{(i)}$ that form a reflex angle with v_j at v_y (light grey region) and repeating the method recursively on the other (non-$C^{(i)}$) vertices that form reflex angles with v_j at v_y (dark grey).

$|H_{\text{direct}}|$, since $G^{\star}_{\text{vis}}$ encodes to which class each arc-label at v_y leads. It remains to show how to determine $|H_{\text{indirect}}|$.

Let $U \subseteq V^{(i+1)}$ be the set of vertices of $G^{(i+1)}_{\text{vis}}$ that are visible to v_y and form a reflex angle with v_j at v_y. Since $G^{(i+1)}_{\text{vis}}$ is given and since $v_y \in V^{(i+1)}$, we can infer the identities of the vertices in U. Every vertex of H_{indirect} is hidden from v_y by exactly one vertex of U. Conversely, every vertex of $C^{(i)}$ that is hidden from v_y by a vertex of U is part of H_{indirect}. We are given $G^{(i+1)}_{\text{vis}}$, hence, for every $u \in U$, we know the index $b' = O_u(v_y)$. We can thus use our algorithm recursively for v_y and every vertex $u \in U \subseteq V^{(i+1)}$ to obtain H_{indirect}. □

Lemma 6.3. *Let $1 \leq i < |V^{\star}|$. It is possible to determine $G^{(i)}_{\text{vis}}$ from $G^{(i+1)}_{\text{vis}}$.*

Proof. The set of vertices $V^{(i)}$ of $G^{(i)}_{\text{vis}}$ is given by $V^{(i)} = C^{(i)} \cup V^{(i+1)}$. It remains to show how to construct $E^{(i)}$. Let A be the set of arcs in G_{vis} between vertices of $C^{(i)}$ and $V^{(i+1)}$, and B be the set of arcs between vertices of $C^{(i)}$. We will first show how to construct A using the information contained in $G^{(i+1)}_{\text{vis}}$ and $G^{\star}_{\text{vis}}$. After having determined A, we can apply the same approach in order to obtain B. This completes the proof as $E^{(i)} = E^{(i+1)} \cup A \cup B$.

Note that every arc in G_{vis} has a counterpart of opposite orientation. In order to construct A it is thus sufficient to consider $e \in V^{(i+1)} \times C^{(i)}$ and show how to decide whether $e \in A$ or $e \notin A$. Deciding which elements of $C^{(i)} \times V^{(i+1)}$ are in A is then immediate. Equivalently, we can consider $v_j \in V^{(i+1)}$ with degree d in $G^{(i)}_{\text{vis}}$ and $1 \leq k \leq d$ such that $\text{vis}_{v_j}(k) \in C^{(i)}$, and show how to "identify" $\text{vis}_{v_j}(k)$, i.e., how to find the index x such that $v_x = \text{vis}_{v_j}(k)$ in $G^{(i)}_{\text{vis}}$. If $k = 1$, we have $x = j + 1$ and if $k = d$, we have $x = j - 1$ because v_j sees its two neighbors on the boundary. Now assume $1 < k < d$. We will show that $v_y := \text{vis}_{v_j}(k - 1)$ cannot lie in $C^{(i)}$. For the sake of contradiction assume that $v_y \in C^{(i)}$. In $\mathcal{P}^{(i)}$ all vertices of $C^{(i)}$ are ears and thus convex. By Lemma 5.1 and $i < |V^{\star}|$, there is more than one class and thus there is a vertex $v_z \in \text{chain}(v_{y+1}, v_{x-1})$ which is not visible to v_j. The (euclidean) shortest path in $\mathcal{P}$ from v_j to v_z must visit v_x or v_y, which is a contradiction to both vertices being convex (Theorem 5.3). We can deduce that $v_y \notin C^{(i)}$ and thus $(v_j, v_y) \in E^{(i+1)}$ is part of $G^{(i+1)}_{\text{vis}}$ and has already been identified, i.e., the index y is known. Because of Lemma 5.1, it is sufficient to know how many vertices of $C^{(i)}$ are in $\text{chain}(v_{y+1}, v_{x-1})$ in order to find x. All these vertices are hidden from v_j by v_y, again because v_x is convex. Either $\text{chain}(v_{y+1}, v_{x-1})$ is empty, or all vertices hidden from v_j by v_y are in $\text{chain}(v_{y+1}, v_{x-1})$. We can distinguish these cases by inspecting $G^{\star}_{\text{vis}}$, as knowing $G^{(i+1)}_{\text{vis}}$ allows us to infer which edge in $G^{\star}_{\text{vis}}$ corresponds to (v_y, v_j). In the first case there trivially are no vertices of $C^{(i)}$ in $\text{chain}(v_{y+1}, v_{x-1})$. In the second case, since we know $b = O_{v_y}(v_j)$ as $v_j \in V^{(i+1)}$, we can use Lemma 6.2 to count the number of vertices of $C^{(i)}$ in $\text{chain}(v_{y+1}, v_{x-1})$ hidden from v_j by v_y to determine x (cf. Figure 5). Once we have determined all arcs in A, we can easily obtain their labels by inspecting $G^{\star}_{\text{vis}}$: For every vertex in $V^{(i+1)}$, the corresponding vertex of $G^{\star}_{\text{vis}}$ gives us the different arc labels that belong to arcs leading to vertices of $C^{(i)}$. Since we already identified all those vertices, we know which label belongs to which arc.

Using the fact that the arcs in A have already been identified, we can apply the exact same approach to construct B. More precisely, for each $v_j \in C^{(i)}$ with degree d, and $1 < k < d$ such that $v_x := \text{vis}_{v_j}(k)$ is in $C^{(i)}$, we can infer the index x by counting (Lemma 6.2) the number of vertices in $C^{(i)}$ hidden from v_j by $v_y := \text{vis}_{v_j}(k-1)$. We can do this because again $v_y \notin C^{(i)}$, and because the edge $(v_y, v_j) \in A$ has been identified before. □

Theorem 3.5 follows directly from Theorem 5.4, Lemma 6.3 and the fact that $G^{(|V^{\star}|)}$ is the complete graph on $n/|V^{\star}|$ vertices.

7. OUTLOOK

We gave a visibility graph reconstruction algorithm for an agent that is able to distinguish convex from reflex angles. The algorithm first determines the minimum base graph, then finds a class that forms a clique, and starting from this clique constructs the visibility graph by repeatedly adding classes of ears and determining the induced edges of the visibility graph.

Note that knowledge of an upper bound $\bar{n}$ on the number of vertices is required only in the first step to find the minimum base graph. In addition, the exponential running time of our algorithm is caused by this step. On the other hand, we do not rely on the geometric meaning of the angle data to find the minimum base graph at all. It would be interesting to see whether a specialized method for finding the base graph exists that makes use of the angle data to avoid an exponential running time or to eliminate the need to know $\bar{n}$.

Table 1 lists other configurations of sensors for which the reconstruction problem remains open. Most prominently, we do not even know whether knowledge of $\bar{n}$ alone suffices.

ACKNOWLEDGMENT

We would like to thank our reviewers that invested significant effort to help us improve the paper.

References

N. Agmon and D. Peleg. 2007. Fault-Tolerant Gathering Algorithms for Autonomous Mobile Robots. *SIAM J. Comput.* 36, 1 (2007), 56–82.

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

D. Angluin. 1980. Local and global properties in networks of processors. In *Proceedings of the twelfth annual ACM symposium on Theory of computing.* 82–93.

D. Bilò, Y. Disser, M. Mihalák, S. Suri, E. Vicari, and P. Widmayer. 2012. Reconstructing Visibility Graphs with Simple Robots. *Theoretical Computer Science* 444 (2012), 52–59.

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

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

J. Chalopin, S. Das, Y. Disser, M. Mihalák, and P. Widmayer. 2011. Mapping Simple Polygons: How Robots Benefit from Looking Back. *Algorithmica* (2011).

J. Chalopin, E. Godard, Y. Métivier, and R. Ossamy. 2006. Mobile agent algorithms versus message passing algorithms. In *Proceedings of the 10th International Conference on the Principles of Distributed Systems.* 187–201.

R. Cohen and D. Peleg. 2008. Convergence of Autonomous Mobile Robots with Inaccurate Sensors and Movements. *SIAM J. Comput.* 38, 1 (2008), 276–302.

S Das, P. Flocchini, S. Kutten, A. Nayak, and N. Santoro. 2007. Map construction of unknown graphs by multiple agents. *Theoretical Computer Science* 385, 1–3 (2007), 34–48.

D. Dereniowski and A. Pelc. 2012. Drawing maps with advice. *J. Parallel and Distrib. Comput.* 72, 2 (2012), 132–143.

Y. Disser, S. K. Ghosh, M. Mihalák, and P. Widmayer. 2013. Mapping a polygon with holes using a compass. *Theoretical Computer Science* (2013).

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

Y. Disser, M. Mihalák, and P. Widmayer. 2013. Mapping polygons with agents that measure angles. In *Proceedings of the 10th International Workshop on the Algorithmic Foundations of Robotics (WAFR).* 415–425.

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

M. Katsev, A. Yershova, B. Tovar, R. Ghrist, and S. M. LaValle. 2011. Mapping and Pursuit-Evasion Strategies For a Simple Wall-Following Robot. *IEEE Transactions on Robotics* 27, 1 (2011), 113–128.

D.-T. Lee and F. P. Preparata. 1984. Euclidean shortest paths in the presence of rectilinear barriers. *Networks* 14, 3 (1984), 393–410.

J. Lin, A. Morse, and B. Anderson. 2007a. The Multi-Agent Rendezvous Problem. Part 1: The Synchronous Case. *SIAM Journal on Control and Optimization* 46, 6 (2007), 2096–2119.

J. Lin, A. Morse, and B. Anderson. 2007b. The Multi-Agent Rendezvous Problem. Part 2: The Asynchronous Case. *SIAM Journal on Control and Optimization* 46, 6 (2007), 2120–2147.

P. Panaite and A. Pelc. 1999. Exploring unknown undirected graphs. *Journal of Algorithms* 33 (1999), 281–295.

S. Suri, E. Vicari, and P. Widmayer. 2008. Simple Robots with Minimal Sensing: From Local Visibility to Global Geometry. *International Journal of Robotics Research* 27, 9 (2008), 1055–1067.

I. Suzuki and M. Yamashita. 1999. Distributed Anonymous Mobile Robots: Formation of Geometric Patterns. *SIAM J. Comput.* 28, 4 (1999), 1347–1363.

M. Yamashita and T. Kameda. 1996. Computing on Anonymous Networks: Part I – Characterizing the Solvable Cases. *IEEE Transactions on Parallel and Distributed Systems* 7, 1 (1996), 69–89.