Theoretical Computer Science 553 (2014) 106–113

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Mapping a polygon with holes using a compass

Yann Disser [a], Subir Kumar Ghosh [b], Matúš Mihalák [a,*], Peter Widmayer [a]

[a] Institute of Theoretical Computer Science, ETH Zurich, Zurich, Switzerland
[b] School of Technology & Computer Science, Tata Institute of Fundamental Research, Mumbai, India

ARTICLE INFO

Available online 18 December 2013

Keywords:
Autonomous robot
Polygon
Hole
Mapping
Exploration
Visibility graph
Compass
Algorithm

ABSTRACT

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

1. Introduction

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

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

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

* Corresponding author.
E-mail addresses: yann.disser@inf.ethz.ch (Y. Disser), ghosh@tifr.res.in (S.K. Ghosh), matus.mihalak@inf.ethz.ch (M. Mihalák), widmayer@inf.ethz.ch (P. Widmayer).

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2013.12.010

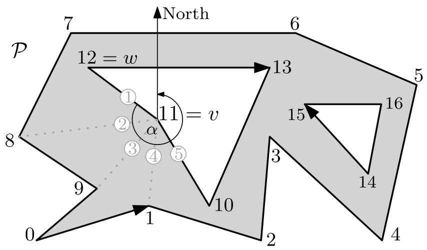

Fig. 1. A polygon $\mathcal{P}$ with two holes. A robot is located at vertex $v = 11$. The gray line-segments connect v to the vertices $\{12, 8, 9, 1, 10\}$ visible to v. The line-segments are labeled in the order as they appear in a counterclockwise scan of $\mathcal{P}$, starting on the boundary. Angle α is the angle between the boundary at v and north.

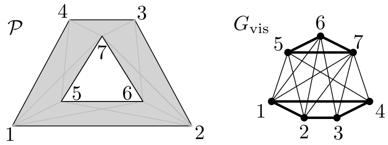

Fig. 2. Left: A polygon $\mathcal{P}$ with vertices $V = \{1, 2, 3, 4, 5, 6, 7\}$ and a hole formed by the vertices 5,6,7. The gray line-segments depict the lines of sight. Right: the visibility graph G_{vis} of $\mathcal{P}$. The fat edges denote cycles corresponding to the boundaries 1, 2, 3, 4 and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

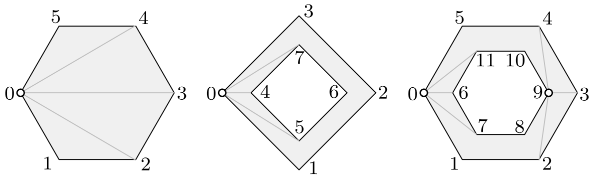

Fig. 3. Polygons that a robot with look-back cannot distinguish, even if it has an upper bound on the number of vertices. At every vertex of the three polygons, the robot observes exactly the same – including the information about which vertex it arrived from in its last move. For convenience, the lines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

$G_{\mathrm{vis}} = (V, E)$, where V are the vertices of $\mathcal{P}$ and E contains the edge $\{u, v\}$ if and only if u and v see each other in $\mathcal{P}$ (i.e., the line segment connecting them does not leave $\mathcal{P}$). Fig. 2 gives an example of a polygon with its visibility graph.

Suri et al. showed that a robot with a *pebble* can solve the mapping problem in polygons with holes, even without information about the size of the polygon [1]. Such a *pebble* is a way for the robot to mark a vertex: The robot can drop the pebble at its current location, it can distinguish the vertex that holds the pebble as long as it is visible, and it can pick the pebble back up later. A pebble is a powerful tool for the robot. It has been shown for example that a much weaker pebble that cannot be sensed from a distance allows a robot exploring any directed graph to reconstruct the graph [2]. It is an important question whether a robot with weaker abilities can solve the mapping problem in polygons with holes. To the best of our knowledge, no such result is known.

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

Fig. 3 also illustrates an important structural property of simple polygons, which polygons with holes do not admit: A simple polygon always has an *ear*, i.e., a vertex whose neighbors on the boundary see each other. In the first polygon in Fig. 3 every vertex is an ear, the other two polygons have no ears at all. This property is crucial for existing mapping techniques, because it allows an inductive approach based on "cutting off" ears repeatedly [3,4]. For polygons with holes, we cannot hope to make use of ears in similar fashion. Solving the mapping problem may thus require a more capable robot.

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