Reconstruction of a polygon from angles without prior knowledge of the size

Author(s):
Disser, Yann; Mihalák, Matúš; Widmayer, Peter

Publication Date:
2010-12

Permanent Link:
https://doi.org/10.3929/ethz-a-006902867

Rights / License:
In Copyright - Non-Commercial Use Permitted
Reconstruction of a polygon from angles without prior knowledge of the size

Technical Report Nr. 700, Department of Computer Science, ETH Zurich

Yann Disser, Matúš Mihalák, Peter Widmayer

December 1, 2010

Abstract

We consider a simple agent exploring a polygon by moving from vertex to vertex along the boundary. At every vertex, the agent observes all angles between pairs of edges of the visibility graph, i.e. the angle between all pairs of visible vertices. The goal of the agent is to reconstruct the shape of the polygon, or equivalently its visibility graph. A polynomial time algorithm exists if the agent knows the total number of vertices a priori. We adapt this algorithm to work even if the number of vertices is not known beforehand.

1 Introduction

We consider the problem of mapping an initially unknown simple polygon $P$ with a minimalist agent. The agent starts at some vertex $v_0$ of $P$ and moves along the boundary in counter-clockwise order step-by-step. The vertex reached by the agent in step $i$ is denoted $v_i$ in the following, where a single vertex can have multiple names if the agent travels further than once around the boundary. Roughly speaking, while situated at a vertex, the agent may use its sensory to collect local data about $P$, and perform computations involving all data collected so far. The goal of the agent is to infer the visibility graph $G_{vis}$ of $P$: the graph having a node for every vertex of $P$ and an edge connecting two nodes if the corresponding vertices see each other. Two vertices are said to see each other if the line segment connecting them lies within the polygon entirely. As every node of $G_{vis}$ corresponds to a vertex of $P$ and every edge of $G_{vis}$ corresponds to a line segment in $P$, we use these terms interchangeably.

Whether or not the agent is capable of reconstructing $G_{vis}$ in the above setting depends on the data it can perceive at each vertex. In this work, we assume the agent to be equipped with an angle sensor. This sensor returns a list of angles $\alpha_1, \alpha_2, \ldots, \alpha_{d-1}$, where $d$ is the number of vertices currently visible to the agent (i.e. the degree of the agent’s location in $G_{vis}$). The angle $\alpha_i$, $0 < i < d$, is the counter-clockwise angle formed by the $i$-th and the $(i+1)$-th edge incident to the current vertex in counter-clockwise order starting on the boundary.

We show that an agent as described above can infer $G_{vis}$ after a number of steps polynomial in the total number of vertices of $P$. In particular, the agent can infer the size $n$ of $P$ from its observations alone. In fact, if $n$ is already known a priori, the agent can initially travel once along the boundary collecting all data. If this angle information is sufficient in order to reconstruct $G_{vis}$, the agent can do this in the end in a single computation phase. It was shown in [1] that this problem can always be solved uniquely by a polynomial-time algorithm, i.e. the agent can determine $G_{vis}$ of $P$ in polynomial time and guarantee that $G_{vis}$ is the only valid solution.

Here, we show that even if $n$ is not known a priori, the agent can reconstruct $G_{vis}$ in polynomial time, where a move is treated like a computational step requiring constant time.
We define the graph \( G^i \) to be the subgraph of \( G_{\text{vis}} \) induced by \( \{v_i, v_{i+1}, \ldots, v_j\} \). The degree of \( v_k \) in \( G^i \) is denoted by \( d^i_k(v_k) \), its degree in graph \( G \) is denoted by \( d_G(v_k) \), and we denote \( d_k := d_G(v_k) = d_G^{n-1}(v_k) \). Observe that \( G_{\text{vis}} = G_i^{n-1} \). By \( \vec{\alpha}_k = (\alpha_{k,1}, \alpha_{k,2}, \ldots, \alpha_{k,d_k-1}) \) we denote the sequence of angles at \( v_k \), such that \( \alpha_{k,x} \) is the angle between the \( x \)-th and \((x+1)\)-th edge incident to \( v_k \) (in ccw order). Furthermore, we write \( \angle^i_{v_i}(v_j) := \sum_{x=1}^{d^i_k(v_j)} \alpha_{i,x} \) and \( \angle^j_{v_j}(v_i) := \sum_{x=1}^{d^j_k(v_i)} \alpha_{j,x} \) (cf. Figure 1). Note that the latter two quantities can be computed from \( (G^i_0, \vec{\alpha}_i) \) and \( (G^j_0, \vec{\alpha}_j) \), respectively.

The following lemma suggests an algorithm for reconstructing \( G_{\text{vis}} \) in polynomial time.

**Lemma 1.** The graph \( G^i_0, 0 \leq i < j \), can be computed from \( G^j_0, G^j_{i+1} \), and \( \vec{\alpha}_i, \vec{\alpha}_{i+1}, \ldots, \vec{\alpha}_j \) in time polynomial in \( j - i \).

Assume that Lemma 1 holds. The agent can then employ the following algorithm to compute \( G_{\text{vis}} \).

We use **sense** and **move** to denote the sensing and moving operations of the agent, respectively. A **sense**-operation at vertex \( v_k \) returns \( \vec{\alpha}_k \). Both operations are assumed to require constant time.

\[
\textbf{function} \quad \text{compute}G_{\text{vis}} \\
G^0_0 \leftarrow (v_0, \emptyset) ; \\
\vec{\alpha}_0 \leftarrow \text{SENSE} ; \\
\text{for } j \leftarrow 1, 2, \ldots \\
\quad \text{if } d^{j-1}_0(v_0) = |\vec{\alpha}_0| + 1 \\
\quad \quad \text{return } G^0_0^{j-1} ; \\
\quad \text{MOVE} ; \\
\quad \vec{\alpha}_j \leftarrow \text{SENSE} ; \\
G^j_j = (v_j, \emptyset) ; \\
\text{for } l \leftarrow j-1, j-2, \ldots, 1, 0 \\
\quad \text{compute } G^l_l \text{ from } G^j_{l+1}, G^{j-1}_l, \vec{\alpha}_l, \vec{\alpha}_{l+1}, \ldots, \vec{\alpha}_j ; \quad \text{(Lemma 1)}
\]

In the outer loop, the algorithm maintains the invariant that, for all \( 0 \leq x \leq y < j \), \( G^y_x \) has been computed. The algorithm terminates in the \( n \)-th iteration of the outer loop, as \( d^0_0(v_0) = |\vec{\alpha}_0| + 1 \). Correctness follows from the invariant. The agent performs exactly \( n - 1 \) moves and \( n \) sensing operations. The total running time is polynomial in \( n \), due to Lemma 1.

It remains to show Lemma 1. Observe that \( G^i_j \) simply is the union of \( G^{j-1}_i \) and \( G^j_{i+1} \), except for maybe the edge \( \{v_i, v_j\} \). In order to compute \( G^i_j \), it is hence sufficient to find out whether \( \{v_i, v_j\} \)
is an edge of $G_{\text{vis}}$, i.e. whether $v_i$ sees $v_j$ or not. Consider the following condition that was proved necessary and sufficient in [1, Lemma 4]:

$v_i$ sees $v_j$ if and only if there is a vertex $v_l \in V_{i+1}^{j-1}$ such that $\{v_i, v_l\} \in E_i^{j-1}$, $\{v_l, v_j\} \in E_{i+1}^j$, and

$$\Delta^\uparrow_{v_l}(v_i, v_j) + \Delta^\uparrow_{v_i}(v_i, v_l) + \Delta_{v_l}(v_j, v_i) = \pi,$$

where $\Delta^\uparrow_{v_l}(v_i, v_j) = \Delta^\uparrow_{v_i}(v_i, v_j) - \Delta^\uparrow_{v_l}(v_i)$, $\Delta^\downarrow_{v_l}(v_i, v_l) = \Delta^\downarrow_{v_l}(v_l, v_i) - \Delta^\downarrow_{v_l}(v_j)$, and $\Delta_l(v_i, v_j) = \Delta^\downarrow_{v_l}(v_l) - \Delta^\uparrow_{v_l}(v_l)$. This condition can easily be tested in polynomial time for every $v_l \in V_i^{j-1}$, provided that $G_i^{j-1}, G_{i+1}^j$, and $\vec{\alpha}_i, \vec{\alpha}_{i+1}, \ldots, \vec{\alpha}_j$ are known. If there is a $v_l$ which satisfies the condition, $\{v_i, v_j\}$ has to be added to $G_i^j$, and otherwise not. Thus, the agent has a polynomial time algorithm for building $G_i^j$ from $G_i^{j-1}, G_{i+1}^j$, and $\vec{\alpha}_i, \vec{\alpha}_{i+1}, \ldots, \vec{\alpha}_j$, which concludes the proof of Lemma 1.

References