Computational Geometry 44 (2011) 418–426

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

A polygon is determined by its angles ☆

Yann Disser *, Matúš Mihalák, Peter Widmayer

Institute of Theoretical Computer Science, ETH Zürich, Switzerland

ARTICLE INFO

Article history:
Received 22 October 2010
Accepted 26 April 2011
Available online 29 April 2011
Communicated by J.-R. Sack

Keywords:
Visibility graph
Reconstruction
Simple polygon
Angles

ABSTRACT

We study the problem of reconstructing a simple polygon from angles measured at the vertices of the polygon. We assume that at each vertex v a sensing device returns a list of angles $\alpha_1, \alpha_2, \ldots$, where α_i is the angle between the i-th and the $(i+1)$-th vertices visible to v in counterclockwise (ccw) order starting with the ccw neighbor of v along the boundary. We prove that the angle measurements at all vertices of a simple polygon together with the order of the vertices along the boundary uniquely determine the polygon (up to similarity). In addition, we give an algorithm for reconstructing the polygon from this information in polynomial time.

1. Introduction

The reconstruction of geometric objects from measurement data has attracted considerable attention over the last decade [11,15,16]. In particular, many variants of the problem of reconstructing a polygon have been studied. We study the reconstruction problem induced by sensors that measure angles in a simple polygon. Our interest for this problem comes from settings in which mobile agents with limited sensing capabilities have to perform tasks such as meeting and exploration [3,8,17]. Our primary focus lies in understanding whether the available data is sufficient to *uniquely* reconstruct the polygon; considerations like the computational efficiency of reconstruction algorithms are secondary in this context.

We consider the reconstruction of a simple polygon from the data measured by an angle sensor at every vertex. More precisely, we assume that at each vertex v of a simple polygon, the sequence of vertices visible from v is perceived in counterclockwise (ccw) order as seen around v, starting on the polygon boundary. As usual, we call two vertices (mutually) visible if the straight line segment connecting them lies in the polygon (which we consider to be a closed region). Note that the segment connecting two mutually visible vertices may contain further collinear vertices without them blocking the line of sight. In addition to seeing visible vertices, the angle sensor at a vertex v measures the angles between adjacent rays from v to the vertices v sees, and it returns the ccw sequence of these angles (cf. Fig. 1). Note that from such an angle measurement it is easy to compute the angles between any pair of rays from v to vertices v sees, not only between adjacent pairs. Also, as rays are perceived in ccw order starting with the ray towards the ccw neighbor along the boundary, an angle measurement distinguishes the interior and the exterior of the polygon. The polygon reconstruction problem takes as input a sequence of angle measurements ordered along the boundary, one measurement at each vertex of a simple polygon, and asks for a simple polygon that fits the measured angles; we call this problem the *polygon reconstruction problem from angles* (cf. Fig. 2). Note that the order of the vertices along the boundary is assumed to be known through the order in which

☆ A preliminary version of this work was presented at the 12th Scandinavian Symposium and Workshops on Algorithm Theory, Bergen, Norway (Disser et al., 2010) [5].

* Corresponding author.
E-mail addresses: ydisser@inf.ethz.ch (Y. Disser), mmihalak@inf.ethz.ch (M. Mihalák), widmayer@inf.ethz.ch (P. Widmayer).

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2011.04.003

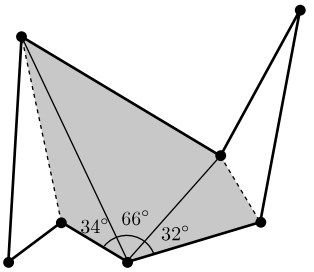

Fig. 1. Illustration of an angle measurement: the sensor returns the vector $(32°, 66°, 34°)$.

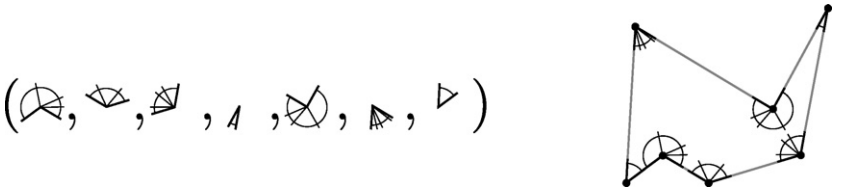

Fig. 2. Given a sequence of angle measurements in ccw order as they appear along the boundary (left), the goal is to find a polygon that fits these angles (right).

the measurements are given. Knowing the order of the vertices, for example, implies that the angle measurements can be oriented with respect to a common reference direction (see "Greedy Approach" in Section 3).

The key difficulty of the reconstruction problem lies in the fact that vertices in our setting have no recognizable labels, i.e. an angle measurement at a vertex returns angles between distant vertices but does not identify these vertices globally.

1.1. Our contribution

The main result of this paper is that no two different polygons can yield the same sequence of angle measurements, i.e. the ordered sequence of angle measurements uniquely determines a polygon (up to similarity). We propose a polynomial-time algorithm that solves the reconstruction problem. Our approach focuses on the reconstruction of the visibility graph of a polygon, i.e. the graph with a node for every vertex of the polygon and an edge between two nodes if the corresponding vertices see each other. It is sufficient to reconstruct the visibility graph, as from the visibility graph and the angle data the shape of the polygon can be inferred in linear time by first computing a triangulation, and then an embedding in the plane by fixing one edge – the geometry of all other edges can subsequently be determined in linear time. We distinguish between the visibility graph as a purely combinatorial object and embeddings of it in the plane which incorporate geometry. In these terms, we show that the only visibility graph compatible with the information contained in the angle data measured in a simple polygon $\mathcal{P}$ is the visibility graph of $\mathcal{P}$. Our algorithm finds this unique visibility graph (in polynomial time) and thus reconstructs the original polygon up to similarity.

While we assume that the measured angles come from a simple polygon, our algorithm is also capable of detecting false inputs, i.e. measurements that do not fit any simple polygon.

1.2. Related work

For our purposes, the combinatorial nature of a polygon is encoded in its visibility graph. While we show that the visibility graph can be computed from the angle data, the full characterization of visibility graphs has been an open problem for many years and has attracted considerable attention [9,10].

A question closely related to the reconstruction of the visibility graph of a polygon appears in the area of robotic exploration, namely the question of what sensory and motion capabilities enable simple robots inside a polygon to reconstruct the visibility graph [2,17]. The idea to reconstruct it from angle data was first discussed in this context [2], but was also mentioned earlier [12]. In [2], the problem was solved for simple robots that can measure angles and additionally are equipped with a compass. In the case of robots that can only distinguish between angles smaller and larger than π, it was shown in the same study that adding the capability of retracing their movements empowers the robots to reconstruct the visibility graph. In both cases a polynomial-time algorithm was given. Recently, it was shown that the ability to retrace their movements alone already enables simple robots to reconstruct the visibility graph in polynomial time if an upper bound on the number of vertices is given [3,4]. Our result implies that if the number of vertices n is given, measuring angles alone is also already sufficient, even if the robot is restricted to moving along the boundary only. On the other hand, it is known that the boundary angles of the polygon (i.e. the angles formed by the boundary at every vertex) do not contain sufficient information to uniquely reconstruct the visibility graph, even when combined with certain combinatorial information that encodes whether two visible vertices form a boundary edge of the polygon [2].

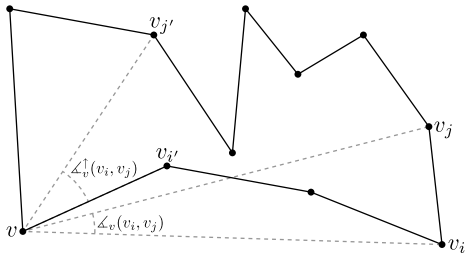

Fig. 3. Illustration of the approximation $\measuredangle_v^{\uparrow}(v_i, v_j) = \measuredangle_v(v_{i'}, v_{j'})$ of the angle $\measuredangle_v(v_i, v_j)$.

The general problem of reconstructing polygons from measurement data can be approached from different perspectives. One possible approach is to concentrate on the question of how to efficiently construct some polygon $\mathcal{P}^\star$ that is consistent with the data measured in the original polygon $\mathcal{P}$. For example, it was studied how a polygon $\mathcal{P}^\star$ compatible with the information obtained from "stabbing" $\mathcal{P}$ or compatible with the set of intersection points of $\mathcal{P}$ with given lines can be constructed [11,15]. While the first of the two studies assumes the number of vertices and their order to be known, the latter does not. We approach the reconstruction problem differently and ask whether it is possible to uniquely reconstruct the original polygon $\mathcal{P}$ from certain data measured in $\mathcal{P}$. The primary challenge here is to decide whether the information contained in the data suffices to reconstruct the original polygon, rather than the analysis of the computational complexity of finding a consistent polygon. A previous study shows, for instance, that the boundary angles of $\mathcal{P}$ together with the cross-ratios of its triangulation and the order of the vertices uniquely determine $\mathcal{P}$ [16]. Another study considers orthogonal polygons without assuming the order of the vertices to be known [13]. The study shows that the set of coordinates of the vertices uniquely determines the polygon.

For different sets of data and without knowledge about the order of the vertices, the polygon reconstruction problem has been shown to be NP-hard [6,7,14]. Recently, data from rather novel sensing devices like range-finding scanners has been considered, and most of the reconstruction problems that such devices naturally induce have been shown to be NP-hard as well, while a few others are polynomial time solvable without even knowing the number of vertices beforehand [1]. We show that the polygon reconstruction problem from angles can be solved in polynomial time if the order of the vertices is known.

The polygon reconstruction problem we consider belongs to the general field of computational geometry. While other areas like computer vision and pattern recognition are also concerned with the general challenge of reconstructing shapes from measurement data, the concrete problems that are formulated in these fields are not related to our setting.

1.3. Outline

We introduce the visibility graph reconstruction problem in detail in Section 2. In Section 3 we show that the problem always has a unique solution. We do so by proposing a polynomial-time algorithm. Section 4 gives a brief discussion and shows that the ccw ordering of the angle data cannot be relaxed.

2. The visibility graph reconstruction problem

Let $\mathcal{P}$ be a simple polygon with visibility graph $G_{\text{vis}} = (V, E_{\text{vis}})$, where V denotes the set of vertices of $\mathcal{P}$ and $n = |V|$. We fix a vertex $v_0 \in V$ and denote the other vertices of $\mathcal{P}$ by $v_1, v_2, \ldots, v_{n-1}$ in ccw order along the boundary starting at v_0's ccw neighbor. The degree of a vertex $v_i \in V$ in G_{vis} is denoted by $d(v_i)$ and the sequence $\text{vis}(v_i) = (v_i, u_1, u_2, \ldots, u_{d(v_i)})$ is defined to enumerate the vertices visible to v_i ordered in ccw order along the boundary starting with v_i itself. We write $\text{vis}_0(v_i)$ to denote v_i itself and $\text{vis}_k(v_i)$, $1 \leqslant k \leqslant d(v_i)$, to denote u_k. Note that knowing $\text{vis}(v_i)$ for every $v_i \in V$ is equivalent to knowing E_{vis}. For two distinct vertices $v_i, v_j \in V$, $\text{chain}(v_i, v_j)$ denotes the sequence $(v_i, v_{i+1}, \ldots, v_j)$ of the vertices between v_i and v_j along the boundary in ccw order. Similarly, $\text{chain}_v(v_i, v_j)$ denotes the subsequence of $\text{chain}(v_i, v_j)$ that contains only the vertices that are visible to v. Note that here and in the following all indices are understood to be modulo n.

We define the *visibility segments* of v to be the segments $\overline{vu_1}, \overline{vu_2}, \ldots, \overline{vu_{d(v)}}$ in this order, where $u_i = \text{vis}_i(v)$ for all $1 \leqslant i \leqslant d(v)$. Similarly, we define the *visibility angles* of v to be the ordered sequence of angles between successive visibility segments such that the i-th visibility angle is the angle between $\overline{vu_i}$ and $\overline{vu_{i+1}}$, for all $1 \leqslant i < d(v)$.

Let $v, v_i, v_j \in V$. We write $\measuredangle_v(v_i, v_j)$ to denote the angle between the lines $\overline{vv_i}$ and $\overline{vv_j}$ (in that order) even if v, v_i, v_j do not mutually see each other. Similarly, for $1 \leqslant l < r \leqslant d(v)$ we write $\varphi_v(l, r)$ to denote $\measuredangle_v(\text{vis}_l(v), \text{vis}_r(v))$. We will need the notion of the approximation $\measuredangle_v^{\uparrow}(v_i, v_j)$ of the angle $\measuredangle_v(v_i, v_j)$, which is defined as follows (cf. Fig. 3): Let $v_{i'}$ be the last vertex in $\text{chain}_v(v_{i+1}, v_i)$ and $v_{j'}$ be the first vertex in $\text{chain}_v(v_j, v_{j-1})$. We then define $\measuredangle_v^{\uparrow}(v_i, v_j) = \measuredangle_v(v_{i'}, v_{j'})$. Observe that if $\{v, v_i\}, \{v, v_j\} \in E_{\text{vis}}$, we have $\measuredangle_v^{\uparrow}(v_i, v_j) = \measuredangle_v(v_i, v_j)$. Also note that knowing the visibility angles of a vertex v is equivalent to knowing $\varphi_v(l_v, r_v)$ for all $1 \leqslant l_v < r_v \leqslant d(v)$.

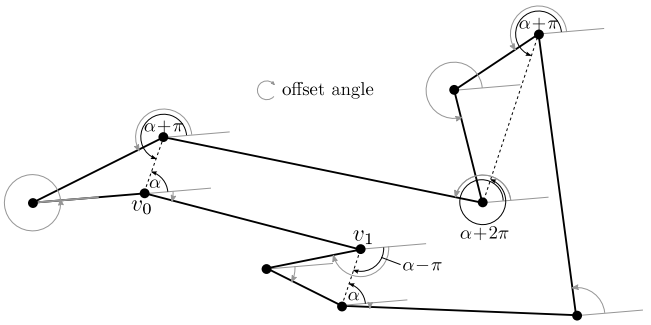

Fig. 4. Illustration of the idea behind the greedy pairing algorithm for a single angle α and starting vertex v_0. If we map angles to the range $[0, 2\pi)$, we allow v_0 and v_1 to be paired which is obviously impossible.

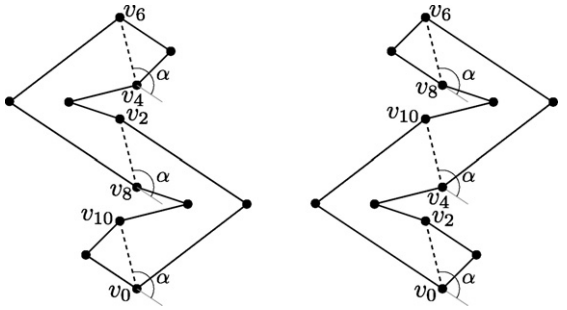

Fig. 5. An example in which only one visibility graph can correctly be reconstructed by any greedy pairing algorithm.

In terms of the above definitions, the goal of the visibility graph reconstruction problem is to find $\mathrm{vis}(v_i)$ for all $v_i \in V$, and thus E_{vis}, when we are given n, V, $d(v)$ for all $v \in V$, and $\varphi_v(l_v, u_v)$ for all $v \in V$ and all $1 \leqslant l_v < u_v \leqslant d(v)$, as well as the (ccw) order in which the vertices appear along the boundary.

3. Uniqueness

The key question when trying to reconstruct the visibility graph of a polygon is how to identify a vertex u visible to some known vertex v. Knowing all angles at every vertex may seem to be more information than necessary and the reconstruction problem may thus seem easily solvable by some greedy algorithm. Before we actually present an algorithm that solves the reconstruction problem, we show that some natural greedy algorithms do not work in general.

3.1. Greedy approach

It is a natural idea to first orient all angles w.r.t. a single, global orientation (e.g., the line $\overline{v_{n-1}v_0}$) by summing angles around the polygon boundary. Then, if a vertex v sees some other vertex u under a certain global angle α, u must see v under the inverse angle $\alpha + \pi$, as the line $\overline{uv}$ has a fixed orientation. A simple greedy approach to identify the vertex u in the view from v would be to walk from v along the boundary and find the first vertex that sees some other vertex under the global angle $\alpha + \pi$. The example in Fig. 4 however shows that this approach does not work in general.

A similar but somewhat stronger approach is to allow global angles to go beyond $[0, 2\pi)$ while summing around the polygon boundary, cf. Fig. 4. This prevents pairing vertex v_0 with vertex v_1 in that example. Nevertheless, there are still examples where this strategy fails, and in fact it is not possible at all to greedily match angles: Inspect Fig. 5 for an example of two polygons for which no matter how a greedy algorithm chooses to pair vertices, it has to fail for one of the two.

3.2. Triangle witness algorithm

We now give an algorithm for the reconstruction of the visibility graph from the visibility angles of all vertices. Note that from now on we map all angles to the range $[0, 2\pi)$. Our algorithm considers all vertices at once and incrementally identifies edges connecting vertices that lie further and further apart along the boundary. In step k of the algorithm we know which vertices in $\{v_{i+1}, v_{i+2}, \ldots, v_{k-1}\}$ are visible to a vertex v_i and we need to decide whether or not v_i sees v_k. Intuitively, the decision boils down to the question whether the next unidentified vertex of $\mathrm{vis}(v_i)$ is v_k. Our algorithm only needs to make decisions of this type. The key ingredient here is the use of a *triangle witness* vertex that indicates whether two other vertices see each other. Because any polygon can be triangulated, we know that for every two vertices $\{v_i, v_j\} \in E_{\mathrm{vis}}$ with $v_j \neq v_{i+1}$, there is a "witness" vertex $v_l \in \mathrm{chain}(v_{i+1}, v_{j-1})$ that they both see such that v_i, v_l, and v_j form a triangle with an angle sum of π. We now extend this notion to the case where $\{v_i, v_j\} \notin E_{\mathrm{vis}}$.

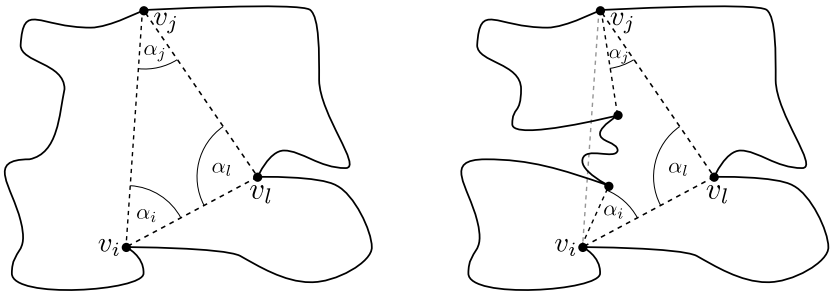

Fig. 6. Illustration of the generalized angle-sum condition of Definition 1. On the left $\{v_i, v_j\} \in E_{\text{vis}}$ and the angles α_i, α_j and α_l of the condition sum up to π – hence v_l is a triangle witness of (v_i, v_j). On the right, $\{v_i, v_j\} \notin E_{\text{vis}}$ and the sum of the angles is strictly less than π – hence v_l is no triangle witness of (v_i, v_j).

Definition 1. Let $v_i, v_j \in V$ be two different vertices and $v_j \neq v_{i+1}$. Let further $v_l \in \text{chain}(v_{i+1}, v_{j-1})$ with $\{v_i, v_l\}, \{v_j, v_l\} \in E_{\text{vis}}$. We say v_l is a *triangle witness* of (v_i, v_j) if it fulfills the *generalized angle-sum condition* (see Fig. 6)

$$\measuredangle^{\uparrow}_{v_i}(v_l, v_j) + \measuredangle^{\uparrow}_{v_j}(v_i, v_l) + \measuredangle_{v_l}(v_j, v_i) = \pi.$$

We will show later that two vertices $v_i, v_j \in V$, $|\text{chain}(v_i, v_j)| > 2$, see each other if and only if there is a triangle witness of (v_i, v_j). Note that if $v_j = v_{i+1}$, the *ordered pair* (v_i, v_j) does not have a triangle witness even though $\{v_i, v_j\} \in E_{\text{vis}}$. Let us briefly motivate the generalized angle-sum condition. As before, we know that if two vertices $v_i, v_j \in V$, $v_j \neq v_{i+1}$, see each other, there must be a vertex $v_l \in \text{chain}(v_{i+1}, v_{j-1})$ which sees both of them. For any such choice of v_l, the condition $\measuredangle_{v_i}(v_l, v_j) + \measuredangle_{v_j}(v_i, v_l) + \measuredangle_{v_l}(v_j, v_i) = \pi$ is trivially fulfilled. In the case that v_i does not see v_j, the only difference from v_i's perspective is that for any choice of v_l, the angle between $\overline{v_i v_l}$ and $\overline{v_i v_j}$ does not appear in v_i's visibility angles φ_{v_i} (although its value might still appear in φ_{v_i}). In order to capture this difference we replace v_j in $\measuredangle_{v_i}(v_l, v_j)$ by an expression that evaluates to v_j if and only if v_i sees v_j. We choose the expression "the first vertex in $\text{chain}_{v_i}(v_j, v_{j-1})$", which is v_j exactly if v_i sees v_j. If, similarly, we also replace v_i in $\measuredangle_{v_j}(v_i, v_l)$ by "the last vertex in $\text{chain}_{v_j}(v_{i+1}, v_i)$", we obtain the generalized angle-sum condition of Definition 1.

We can now describe the triangle witness algorithm. It iterates over an increasing number of steps k along the boundary, focusing at step k on all edges of the form $\{v_i, v_{i+k}\}$. Throughout, it maintains two maps F, B that store for every vertex all the edges identified so far that go at most k steps forward or backward along the boundary, respectively. Both $F[v_i][v_j] = s$ and $B[v_i][v_j] = s$ denote that $\{v_i, v_j\}$ is the s-th edge incident to v_i in ccw order. The difference between $F[v_i]$ and $B[v_i]$ is that $B[v_i]$ is filled in clockwise order by the algorithm, i.e. its first entry will be $B[v_i][v_{i-1}] = d(v_i)$. Whenever convenient, we use $F[v_i]$ and $B[v_i]$ like a set, e.g. we write $v_l \in F[v_i]$ to denote that there is an entry v_l in $F[v_i]$ and write $|F[v_i]|$ to denote the number of entries in $F[v_i]$. It is clear that once we computed the maps for k up to $\lceil \frac{n}{2} \rceil$, we essentially have computed E_{vis}.

The initialization of the maps for $k = 1$ is simple as every vertex sees its neighbors on the boundary. In later iterations, for every vertex v_i there is always exactly one candidate vertex for v_{i+k}, namely the $(|F[v_i]| + 1)$-th vertex visible to v_i. We decide whether v_i and v_{i+k} see each other by going over all vertices between v_i and v_{i+k} in ccw order along the boundary and checking whether there is a triangle witness $v_l \in \text{chain}(v_{i+1}, v_{i+k-1})$ of (v_i, v_{i+k}). If and only if this is the case, we update E_{vis}, F, and B with the edge $\{v_i, v_{i+k}\}$. For a listing of the triangle witness algorithm see Algorithm 1.

In the following we prove the correctness of the triangle witness algorithm. For this we mainly have to show that having a triangle witness is necessary and sufficient for a pair of vertices to see each other. To show this, we will need the notion of *blockers* and *shortest paths* in polygons.

Definition 2. Let $v_i, v_j \in V$. We say $v_b \in \text{chain}(v_{i+1}, v_{j-1})$ is a *blocker* of (v_i, v_j) if for all $u \in \text{chain}(v_i, v_{b-1}), v \in \text{chain}(v_{b+1}, v_j)$ we have $\{u, v\} \notin E_{\text{vis}}$.

Note that if v_b is a blocker of (v_i, v_j), v_b also is a blocker of (u, v) for all $u \in \text{chain}(v_i, v_{b-1})$, $v \in \text{chain}(v_{b+1}, v_j)$.

A *path* between two vertices $a, b \in V$ of a polygon $\mathcal{P}$ is defined as a curve that lies entirely in $\mathcal{P}$ and has a and b as its endpoints. A *shortest path* between a and b is a path of minimum (Euclidean) length among all the paths between a and b.

Lemma 3. *Let $v_i, v_j \in V$. The shortest path in $\mathcal{P}$ between v_i and v_j is unique and is a chain of straight line segments that connect at vertices of $\mathcal{P}$ (cf. Lemmas* 3.2.3 *and* 3.2.5 *in* [9]*).*

We can therefore write $(a, u_0, u_1, \ldots, b)$ to denote a shortest path, where no three subsequent vertices are collinear. The points u_i are the locations at which the path bends and we refer to them as the path's *interior vertices*. The following statements motivate the term 'blocker'.

Input: n, $d(\cdot)$, $\varphi_\cdot(\cdot,\cdot)$
Output: E_{vis}

1. $F \leftarrow$ [array of n empty maps], $B \leftarrow$ [array of n empty maps], $E_{\text{vis}} \leftarrow \emptyset$
2. **for** $i \leftarrow 0, \dots, n-1$
3. $\quad E_{\text{vis}} \leftarrow E_{\text{vis}} \cup \{v_i, v_{i+1}\}$
4. $\quad F[v_i][v_{i+1}] \leftarrow 1$
5. $\quad B[v_{i+1}][v_i] \leftarrow d(v_i)$
6. **for** $k \leftarrow 2, \dots, \lceil \frac{n}{2} \rceil$
7. $\quad$ **for** $i \leftarrow 0, \dots, n-1$
8. $\quad\quad j \leftarrow i+k$
9. $\quad\quad$ **for** $l \leftarrow i+1, \dots, j-1$
10. $\quad\quad\quad$ **if** $v_l \in F[v_i] \wedge v_l \in B[v_j]$
11. $\quad\quad\quad\quad \alpha_i \leftarrow \varphi_{v_i}(F[v_i][v_l], |F[v_i]|+1)$ $\quad (= \measuredangle^{\uparrow}_{v_i}(v_l, v_j)$, cf. Theorem 8)
12. $\quad\quad\quad\quad \alpha_j \leftarrow \varphi_{v_j}(d(v_j) - |B[v_j]|, B[v_j][v_l])$ $\quad (= \measuredangle^{\uparrow}_{v_j}(v_i, v_l)$, cf. Theorem 8)
13. $\quad\quad\quad\quad \alpha_l \leftarrow \varphi_{v_l}(F[v_l][v_j], B[v_l][v_i])$ $\quad (= \measuredangle_{v_l}(v_j, v_i)$, cf. Theorem 8)
14. $\quad\quad\quad\quad$ **if** $\alpha_i + \alpha_j + \alpha_l = \pi$
15. $\quad\quad\quad\quad\quad E_{\text{vis}} \leftarrow E_{\text{vis}} \cup \{v_i, v_j\}$
16. $\quad\quad\quad\quad\quad F[v_i][v_j] = |F[v_i]| + 1$
17. $\quad\quad\quad\quad\quad B[v_j][v_i] = d(j) - |B[v_j]|$
18. $\quad\quad\quad\quad\quad$ abort innermost loop

Algorithm 1. Triangle witness algorithm.

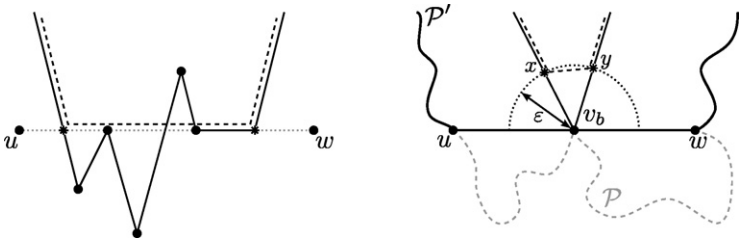

Fig. 7. Illustration of the objects in the proof of Lemma 4. Left: a path that crosses or touches $\overline{uw}$ four times and can thus be shortened (dashed line). Right: v_b lies on a segment of the boundary of $\mathcal{P}'$, therefore we can shorten any path that lies in $\mathcal{P}'$ and bends at v_b.

Lemma 4. *Let $v_i, v_j \in V$ with $\{v_i, v_j\} \notin E_{\text{vis}}$. Every interior vertex of the shortest path from v_i to v_j is a blocker of either (v_i, v_j) or (v_j, v_i).*

Proof. Let p_{ij} be the shortest path from v_i to v_j. We start by observing that, for each $\{u, w\} \in E_{\text{vis}}$, p_{ij} cannot cross (or touch) the segment $\overline{uw}$ more than once (cf. Fig. 7 (left)). Otherwise, we could find a shorter path by replacing the part of p_{ij} from the first to last point of its intersection with $\overline{uw}$ by a straight line segment. For the sake of contradiction assume that $v_b \in V$ is an interior vertex of the shortest path p_{ij} from v_i to v_j which is not a blocker of either (v_i, v_j) or (v_j, v_i). W.l.o.g. assume $v_b \in \text{chain}(v_{i+1}, v_{j-1})$. As v_b is not a blocker of (v_i, v_j), there are two vertices $u \in \text{chain}(v_{i+1}, v_{b-1})$, $w \in \text{chain}(v_{b+1}, v_{j-1})$ with $\{u, w\} \in E_{\text{vis}}$. Let $\mathcal{P}'$ be the subpolygon induced by the vertices in $\text{chain}(w, u)$ (cf. Fig. 7 (right)). Because p_{ij} cannot cross (or touch) the segment $\overline{uw}$ more than once, it must lie in $\mathcal{P}'$ completely. Hence, v_b must lie in $\mathcal{P}'$ and be collinear with u and w. Because $\mathcal{P}'$ is a simple polygon, there is an $\varepsilon > 0$ such that no part of the boundary of $\mathcal{P}'$, except for $\overline{uw}$, lies closer to v_b than ε. Let x, y be the intersection points of p_{ij} with a ball of radius ε around v_b. If we replace the part of p_{ij} from x to y via v_b by a straight line segment, we obtain a shorter path, which is a contradiction to p_{ij} being the shortest path. Note that x and y cannot both lie on $\overline{uw}$, as v_b is an interior vertex of p_{ij}. □

Corollary 5. *Let $v_i, v_j \in V$. If $\{v_i, v_j\} \notin E_{\text{vis}}$, there is either a blocker of (v_i, v_j) or of (v_j, v_i).*

We now relate the definition of a blocker to the geometry of the polygon.

Lemma 6. *Let $v_i, v_j \in V$ with $i = j + 2$, $\{v_i, v_j\} \notin E_{\text{vis}}$. If $w := v_{j+1} = v_{i-1}$ is convex (inner angle $\leqslant \pi$), then $v_{j'} = \arg\min_{v_b \in \text{chain}_{v_i}(v_{i+1}, v_{j-1})} \measuredangle_{v_i}(v_b, w)$ and $v_{i'} = \arg\min_{v_b \in \text{chain}_{v_j}(v_{i+1}, v_{j-1})} \measuredangle_{v_j}(w, v_b)$ are blockers of (v_i, v_j) that lie in the interior of the triangle defined by v_i, v_{i+1}, and v_j.*

Proof. As w is convex, the shortest path p_{ij} from v_i to v_j only contains vertices of $\text{chain}(v_i, v_j)$. As p_{ij} only makes right turns (i.e. any three consecutive vertices on p_{ij} form a ccw triangle), all interior vertices of p_{ij} lie strictly left of the oriented line $\overline{v_i v_j}$ and hence in the interior of the triangle defined by v_i, v_{i+1}, and v_j. Furthermore $v_{j'}$ and $v_{i'}$ are the first and the last interior vertices of p_{ij} respectively. By Lemma 4 we thus know that both $v_{j'}$ and $v_{i'}$ are blockers of (v_i, v_j). □

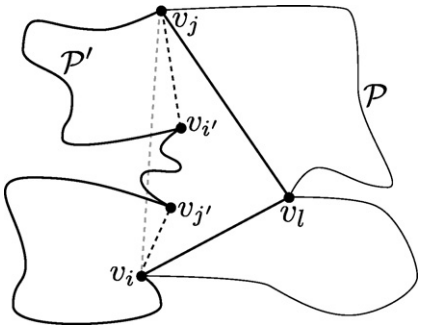

Fig. 8. Sketch of the definitions in the proof of Lemma 7.

We now get to the central lemma that essentially states that the existence of a triangle witness is necessary and sufficient for a pair of vertices to see each other.

Lemma 7. *Let $v_i, v_j \in V$ with $|\text{chain}(v_i, v_j)| > 2$. There is a triangle witness v_l of (v_i, v_j) if and only if $\{v_i, v_j\} \in E_{\text{vis}}$.*

Proof. If $\{v_i, v_j\} \in E_{\text{vis}}$, because there exists a triangulation of the polygon containing the edge $\{v_i, v_j\}$, there must be a vertex $v_l \in \text{chain}(v_{i+1}, v_{j-1})$ for which both edges $\{v_i, v_l\}$ and $\{v_l, v_j\}$ are in E_{vis}. For this vertex we have $\measuredangle^{\uparrow}_{v_i}(v_l, v_j) + \measuredangle^{\uparrow}_{v_j}(v_i, v_l) + \measuredangle_{v_l}(v_j, v_i) = \measuredangle_{v_i}(v_l, v_j) + \measuredangle_{v_j}(v_i, v_l) + \measuredangle_{v_l}(v_j, v_i) = \pi$ as all three relevant edges are in E_{vis} and the sum over the angles of any triangle is π.

For the converse implication assume there is a triangle witness v_l of (v_i, v_j). For the sake of contradiction, assume $\{v_i, v_j\} \notin E_{\text{vis}}$.

Consider the polygon $\mathcal{P}'$ induced by the vertices $v_i, v_l, v_j, \text{chain}(v_{j+1}, v_{i-1})$, cf. Fig. 8. As $\{v_i, v_l\}, \{v_l, v_j\} \in E_{\text{vis}}$, $\mathcal{P}'$ is simple and well defined. In $\mathcal{P}'$, v_l is a convex vertex, as it fulfills the generalized angle-sum condition of Definition 1 and thus $\measuredangle_{v_l}(v_j, v_i) \leqslant \pi$, because all angles are non-negative. We can therefore apply Lemma 6 (on v_j, v_i) w.r.t. $\mathcal{P}'$ and conclude that both $v_{j'}$ and $v_{i'}$ block (v_j, v_i), where $v_{j'} = \arg\min_{v_b \in \text{chain}_{v_j}(v_{j+1}, v_{i-1})} \measuredangle_{v_i}(v_l, v_b)$ and $v_{i'} = \arg\min_{v_b \in \text{chain}_{v_j}(v_{j+1}, v_{i-1})} \measuredangle_{v_j}(v_b, v_l)$. This is then also true in our original polygon $\mathcal{P}$ and thus $v_{i'} \in \text{chain}(v_j, v_{j'})$ as otherwise $v_{j'}$ would block $(v_j, v_{i'})$ and $v_{i'}$ would block $(v_{j'}, v_i)$ contradicting the definition of $v_{j'}$ and $v_{i'}$, respectively. Observe that $v_{i'}$ is the last vertex in $\text{chain}(v_{i+1}, v_i)$ visible to v_j and $v_{j'}$ is the first vertex in $\text{chain}(v_j, v_{j-1})$ visible to v_i.

By applying Lemma 6 to $\mathcal{P}'$, we know that both $v_{j'}$ and $v_{i'}$ lie inside the triangle defined by v_i, v_l, and v_j. This means $\measuredangle^{\uparrow}_{v_i}(v_l, v_j) = \measuredangle_{v_i}(v_l, v_{j'}) < \measuredangle_{v_i}(v_l, v_j)$ and $\measuredangle^{\uparrow}_{v_j}(v_i, v_l) = \measuredangle_{v_j}(v_{i'}, v_l) < \measuredangle_{v_j}(v_i, v_l)$ and thus $\measuredangle^{\uparrow}_{v_i}(v_l, v_j) + \measuredangle^{\uparrow}_{v_j}(v_i, v_l) + \measuredangle_{v_l}(v_j, v_i) < \measuredangle_{v_i}(v_l, v_j) + \measuredangle_{v_j}(v_i, v_l) + \measuredangle_{v_l}(v_j, v_i) = \pi$, which is a contradiction with our assumption that v_l is a triangle witness of (v_i, v_j). □

Theorem 8. *The triangle witness algorithm is correct and computes a unique solution.*

Proof. As the edges in E_{vis} are the same as the edges stored in F and the same as the edges stored in B throughout the algorithm, it is sufficient to show that after step k of the iteration both F and B contain exactly the edges between vertices that are at most k steps apart along the boundary. As no two vertices can be further apart than $\lceil \frac{n}{2} \rceil$ steps along the boundary, this immediately implies that E_{vis} eventually contains exactly the edges of the visibility graph. More precisely, we inductively show that after step k of the iteration, $F[v_i]$ contains the vertices of $\text{chain}_{v_i}(v_{i+1}, v_{i+k})$ and $B[v_i]$ contains the vertices of $\text{chain}_{v_i}(v_{i-k}, v_{i-1})$ for all $v_i \in V$. For the sake of simplicity we abuse notation and write $F[v_i] = \text{chain}_{v_i}(v_{i+1}, v_{i+k})$ and $B[v_i] = \text{chain}_{v_i}(v_{i-k}, v_{i-1})$.

The discussion for $k = 1$ is trivial as every vertex has an edge to both its neighbors. The algorithm initializes F and B to consist of these edges. It remains to show for all $0 \leqslant i < n$ that assuming $F[v_i] = \text{chain}_{v_i}(v_{i+1}, v_{i+k-1})$ and $B[v_i] = \text{chain}_{v_i}(v_{i-k+1}, v_{i-1})$ after step $k-1$, we have $F[v_i] = \text{chain}_{v_i}(v_{i+1}, v_{i+k})$ and $B[v_i] = \text{chain}_{v_i}(v_{i-k}, v_{i-1})$ after step k.

The algorithm adds an edge between two vertices v_i and v_{i+k} if and only if there is a vertex $v_l \in \text{chain}(v_{i+1}, v_{i+k-1})$ with $v_l \in F[v_i]$ and $v_l \in B[v_{i+k}]$ for which $\alpha_i + \alpha_j + \alpha_l = \pi$, where $\alpha_i, \alpha_j, \alpha_l$ are defined as in Algorithm 1. As v_i and v_l are less than k steps apart on the boundary, the induction assumption implies that $F[v_i] = \text{chain}_{v_i}(v_{i+1}, v_{i+k-1})$ and $B[v_{i+k}] = \text{chain}_{v_{i+k}}(v_{i+1}, v_{i+k-1})$. Thus, $v_l \in F[v_i]$ and $v_l \in B[v_{i+k}]$ is equivalent to $\{v_i, v_l\}, \{v_{i+k}, v_l\} \in E_{\text{vis}}$ and by Lemma 7 it suffices to show that $\alpha_i = \measuredangle^{\uparrow}_{v_i}(v_l, v_{i+k}), \alpha_j = \measuredangle^{\uparrow}_{v_{i+k}}(v_i, v_l)$ and $\alpha_l = \measuredangle_{v_l}(v_{i+k}, v_i)$ for all $v_l \in F[v_i] \cap B[v_{i+k}]$. Again, by induction, we have $F[v_i] = \text{chain}_{v_i}(v_{i+1}, v_{i+k-1})$, and thus $\text{vis}_{F[v_i][v_l]}(v_i) = v_l$ and $\text{vis}_{|F[v_i]|+1}(v_i) = \arg\min_{v_b \in \text{chain}_{v_i}(v_{i+k}, v_{i-1})} \measuredangle_{v_i}(v_{i+1}, v_b)$ and, consequently, we obtain that $\alpha_i = \varphi_{v_i}(F[v_i][v_l], |F[v_i]| + 1) = \measuredangle^{\uparrow}_{v_i}(v_l, v_{i+k})$. Similarly, as v_l and v_{i+k} are less than k steps apart on the boundary, we get $\alpha_j = \measuredangle^{\uparrow}_{v_{i+k}}(v_i, v_l)$. By the induction assumption we also have $\text{vis}_{F[v_l][v_{i+k}]}(v_l) = v_{i+k}$ and $\text{vis}_{B[v_l][v_i]}(v_l) = v_i$ and thus $\alpha_l = \varphi_{v_l}(F[v_l][v_j], B[v_l][v_i]) = \measuredangle_{v_l}(v_{i+k}, v_i)$.

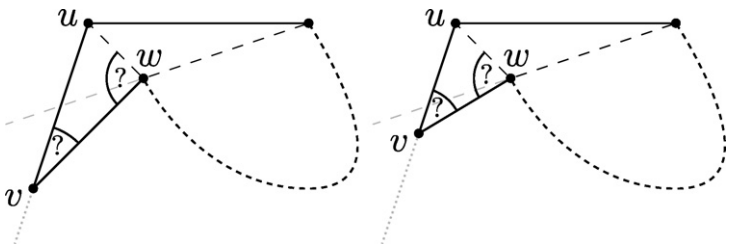

Fig. 9. Sketch of how to construct two polygons of arbitrary size and different shape which have the same (exact) angles except for two. The vertex v in the figure could be located anywhere on the dotted line as long as v does not see other vertices than u and w.

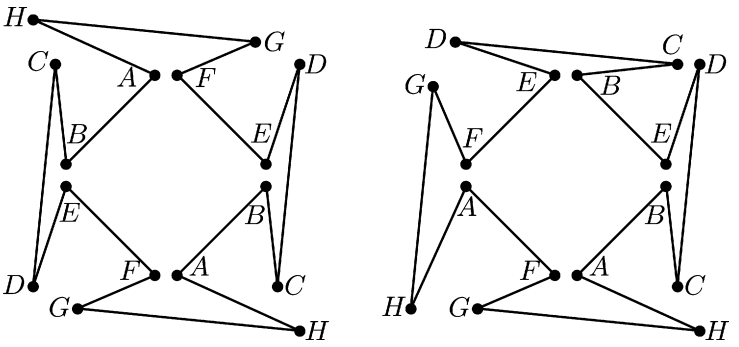

Fig. 10. Two polygons of different shape with the same angle measurements (A–H). The angle measurements appear in different orders along the boundary.

The uniqueness of the algorithm's solution follows immediately from the fact that the existence of a triangle witness is necessary and sufficient for two vertices to see each other. □

3.3. Geometry reconstruction

So far we have developed an algorithm for reconstructing the visibility graph of a polygon from its visibility angles given in ccw order. The geometry of the polygon (up to similarity) can easily be obtained from the visibility graph together with the visibility angles as follows. First we triangulate the polygon. This is possible since every polygon admits a triangulation, and it is easy to see that such a triangulation can be found in linear time from the visibility graph. The visibility angles fix the shape of every triangle in the triangulation, and the shapes of the individual triangles can be combined into the shape of the polygon. It remains to argue that independent of the choice of triangulation, the resulting polygon is the same. This is due to the fact that every possible triangulation *uniquely* leads to a polygon, and this polygon in turn admits every single one of the possible triangulations.

4. Discussion

While our main focus in the above has been to show that it is at all possible to uniquely reconstruct a polygon from its angles, it is worth mentioning that the triangle witness algorithm we provide runs in polynomial time. A straightforward implementation using self-balancing trees for F and B achieves a running time of $O(n^3 \log n)$. Also, as the triangle witness algorithm computes a unique solution, it provides an immediate way of identifying inconsistent input, i.e. angle data that does not belong to any polygon. If upon termination of the algorithm $|F[v_i] \cup B[v_i]| \neq d(v_i)$ for some vertex v_i, the input must be inconsistent. Otherwise, we can compute a triangulation of the visibility graph and infer the shape of the polygon from it. Because of the uniqueness of the algorithm's solution, the input was consistent if and only if this shape is valid (i.e. has no self-intersections).

We have established that the information contained in the angles of a polygon uniquely determines its shape. A natural follow-up question is whether we actually need all of this information or whether we can do with less. Recall that we assume to be given a list of angle measurements for every vertex, where both the list of measurements as well as each individual measurement are in ccw order. There are different ways how we might try to weaken the available data. First, we might not be given every angle value (exactly). It is evident, that we can still infer the shape of the polygon if we only miss a single (exact) angle value. We can compute the missing value, as the angles of a polygon need to sum to $(n-2)\pi$. Fig. 9 shows that as soon as we do not know the (exact) value of two or more angles, we cannot uniquely infer the shape any more.

Another way of weakening the data is to relax the ordering in which the angles are given. On the one hand, we might not be given the list of measurements in ccw order. Fig. 10 gives two polygons with the same *sets* of measurements and

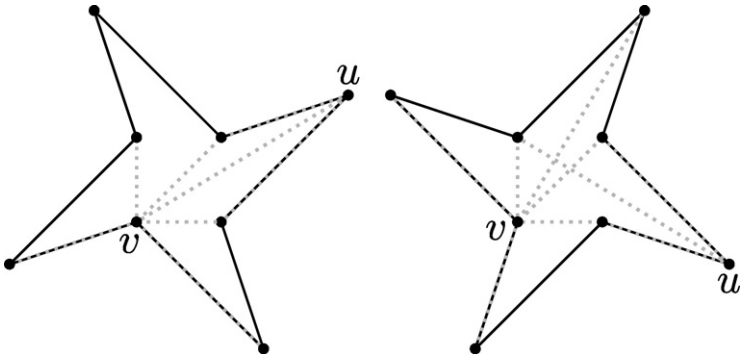

Fig. 11. Two polygons of different shape with the same sets of angles at corresponding vertices. Compared to the first polygon, all angles at a vertex of type u or v in the second polygon appear in reverse order.

different shape, proving that without an order on the list of measurements the shape is not uniquely determined. On the other hand, the individual measurements might not be ordered. In that case, we again cannot uniquely infer the shape of the polygon as Fig. 11 proves.

While it might be possible to reconstruct the shape of a polygon from weaker data, we have shown that reconstruction from angle measurements is impossible if the angles are unordered or incomplete. In Section 3 we have shown that an ordered and complete list of angle measurements on the other hand allows the unique reconstruction of the polygon shape. In that sense our results are tight.

5. Conclusion

In this paper we have considered the problem of reconstructing a simple polygon from an ordered list of angle measurements at every vertex along the boundary. We have shown that the visibility graph of the original polygon can uniquely be reconstructed from this data, and we proposed a polynomial-time algorithm for the reconstruction. From the visibility graph and the angle data, the geometry of the polygon can be inferred up to similarity. Our results thus imply that the ordered list of angle measurements is sufficient to uniquely reconstruct a polygon. In terms of reconstructing a polygon from angles, we have seen that essentially all angles are needed and that they must be given in cyclic order for unique reconstruction to be possible.

References

[1] T. Biedl, S. Durocher, J. Snoeyink, Reconstructing polygons from scanner data, Theoretical Computer Science (2010), doi:10.1016/j.tcs.2010.10.026, in press.
[2] D. Bilò, Y. Disser, M. Mihalák, S. Suri, E. Vicari, P. Widmayer, Reconstructing visibility graphs with simple robots, in: Proceedings of the 16th International Colloquium on Structural Information and Communication Complexity, 2009, pp. 87–99.
[3] J. Chalopin, S. Das, Y. Disser, M. Mihalák, P. Widmayer, How simple robots benefit from looking back, in: Proceedings of the 7th International Conference on Algorithms and Complexity, 2010, pp. 229–239.
[4] J. Chalopin, S. Das, Y. Disser, M. Mihalák, P. Widmayer, How simple robots benefit from looking back: Reconstructing visibility graphs of polygons, Tech. Rep. 692, ETH Zürich, Institute of Theoretical Computer Science, October 2010.
[5] Y. Disser, M. Mihalák, P. Widmayer, Reconstructing a simple polygon from its angles, in: Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory, 2010, pp. 13–24.
[6] C. Evrendilek, B. Genç, B. Hnich, Covering oriented points in the plane with orthogonal polygons is NP-complete, in: Electronic Notes in Discrete Mathematics, vol. 36, Elsevier, 2010, pp. 303–310.
[7] M. Formann, G. Woeginger, On the reconstruction of simple polygons, Bulletin of the EATCS 40 (1990) 225–230.
[8] A. Ganguli, J. Cortes, F. Bullo, On rendezvous for visually-guided agents in a nonconvex polygon, in: Proceedings of the 44th IEEE Conference of Decision and Controls, and the European Control Conference, 2005, pp. 5686–5691.
[9] S.K. Ghosh, Visibility Algorithms in the Plane, Cambridge University Press, 2007.
[10] S.K. Ghosh, P.P. Goswami, Unsolved problems in visibility graph theory, in: Proceedings of the India–Taiwan Conference on Discrete Mathematics, 2009, pp. 44–54.
[11] L. Jackson, S. Wismath, Orthogonal polygon reconstruction from stabbing information, Computational Geometry 23 (1) (2002) 69–83.
[12] T. Kameda, M. Yamashita, On the reconstruction of polygons with (simple) robots, personal communication, 2009.
[13] J. O'Rourke, Uniqueness of orthogonal connect-the-dots, in: G. Toussaint (Ed.), Computational Morphology, North-Holland, 1988, pp. 97–104.
[14] D. Rappaport, On the complexity of computing orthogonal polygons from a set of points, Tech. Rep. SOCS-86.9, McGill University, Montreal, Canada, 1986.
[15] A. Sidlesky, G. Barequet, C. Gotsman, Polygon reconstruction from line cross-sections, in: Proceedings of the 18th Annual Canadian Conference on Computational Geometry, 2006, pp. 81–84.
[16] J. Snoeyink, Cross-ratios and angles determine a polygon, Discrete and Computational Geometry 22 (4) (1999) 619–631.
[17] S. Suri, E. Vicari, P. Widmayer, Simple robots with minimal sensing: From local visibility to global geometry, International Journal of Robotics Research 27 (9) (2008) 1055–1067.