# Reconstructing Visibility Graphs with Simple Robots

Davide Bilò[1], Yann Disser[1], Matúš Mihalák[1], Subhash Suri[2,⋆], Elias Vicari[1], and Peter Widmayer[1]

[1] Institute of Theoretical Computer Science, ETH Zürich
{dbilo,ydisser,mmihalak,vicariel,widmayer}@inf.ethz.ch
[2] Department of Computer Science, University of California, Santa Barbara
suri@cs.ucsb.edu

**Abstract.** We consider the problem of finding a minimalistic configuration of sensors that enable a simple robot inside an initially unknown polygon $\mathcal{P}$ on $n$ vertices to reconstruct the visibility graph of $\mathcal{P}$. The robot can sense features of its environment through its sensors, and it is allowed to move from vertex to vertex.

We aim at understanding which sensorial capabilities are sufficient for the reconstruction of the visibility graph of $\mathcal{P}$. We are able to show that the *combinatorial visibilities* at every vertex do not contain enough information even when combined with the knowledge of the exact interior angle at each vertex. Using sensors that can put distant vertices into a spatial relation on the other hand can in some cases enable our robot to reconstruct the visibility graph of $\mathcal{P}$. We show that this is true for a sensor that can distinguish whether the angle between two vertices the robot sees is convex or reflex, as long as the robot is capable of identifying the vertex it last visited. We also show that measuring angles exactly is enough, if the robot has a compass.

## 1 Introduction

We aim at finding minimalistic motor and sensory capabilities that enable simple robots to explore an unknown environment. The exploration of environments is an important robotic task [8]. Recently, it has also been studied in the context of simple robots giving rise to different modelling approaches [4,10,11]. We model robots as points in an initially unknown polygonal environment $\mathcal{P}$ whose number of vertices $n$ is assumed to be known. We allow a robot to collect sensory input while it is located at a vertex, and to move from its current vertex to any vertex that it sees. In the spirit of keeping robots simple, our robots in particular cannot sense while they move. Our basic sensing capability allows each robot to see all vertices that are visible from its current position, in counter-clockwise (ccw) order, where a vertex is said to be visible by a robot sitting on another vertex if the line segment connecting both vertices lies entirely in $\mathcal{P}$. Vertices have no

⋆ This author wishes to acknowledge the support provided by the National Science Foundation under grants CNS-0626954 and CCF-0514738.

S. Kutten and J. Žerovnik (Eds.): SIROCCO 2009, LNCS 5869, pp. 87–99, 2010.

characteristics that identify them globally; they can be distinguished only in a local sense by the relative position in ccw order when looking from some other vertex. For a variety of configurations of additional sensory capabilities, we analyze whether a robot is capable to infer the visibility graph of $\mathcal{P}$. Recall that the visibility graph consists of a vertex for each polygon vertex, with an edge between two vertices if the polygon vertices are mutually visible. The characterization of visibility graphs and their reconstruction from polygon geometry have been studied extensively [5]. We are interested in the problem of deciding whether a given set of sensory and motor capabilities is powerful enough to allow a robot to reconstruct the visibility graph of its polygonal environment without any prior knowledge of the polygon's geometry.

An earlier study [10] assumes that robots have no notion of and no way of measuring coordinates, distances or angles. Instead, these robots are limited to distinguish whether any two visible vertices are neighbors on the polygon boundary. This concept is usually referred to as *combinatorial visibility* and will be defined more formally below. In a convex polygon, for instance, a robot at any of the vertices sees all other vertices and sees that any two consecutive vertices in cyclic order are neighbors (this is obviously not true in any other polygon). There was hope that the knowledge of all combinatorial visibilities might be enough for a robot to derive the visibility graph of $\mathcal{P}$. In this paper, we show that this knowledge alone is in fact not sufficient. In fact our result implies that a robot that senses combinatorial visibilities cannot reconstruct the visibility graph, if it only moves along the boundary. The question whether the robot is capable of reconstructing the visibility graph, if allowed to move to any vertex it sees remains open.

For certain robotic tasks, such as the rendezvous of two robots in an unknown polygon, a visibility graph might not be needed, for instance if the simple polygon looks non-periodic to the robot(s). For periodic-looking polygons, there was hope that the vertices seen from a given vertex and those seen from a "periodic partner" of the given vertex (details follow in Section 3) would themselves be periodic partners; this property would have allowed a variety of tasks to be solved. We show that this, unfortunately, is not the case.

The question arises what kind of minimal information a robot needs to make the derivation of the visibility graph possible. We show that adding the knowledge of all the inner polygon angles to the knowledge of combinatorial visibilities is still not enough. Instead, we equip the robot with sensors that are able to put the vertices a robot sees into a certain spatial relation only. One example of such a sensor distinguishes whether the angle between any pair of vertices the robot sees is convex or reflex. We show that if we add the ability of the robot to know were it came from when moving from vertex to vertex, this simple sensor is already sufficient. We also show that sensing exact angles is sufficient as long as we add a compass that provides a global reference direction.

*Related Work.* The capabilities of robots and strategies for different robotic tasks have been studied in a broad variety of settings [8]. Our setting of simple robots in a polygonal environment was first introduced in [10].

While we focus on mapping unknown environments, other robotic tasks have been studied using simple robots. One example is the gathering problem in the plane with multiple robots [3]. Examples in polygonal environments include localisation problems [7] and the construction of competitive watchman tours [6]. In contrast to our approach, the models used mostly allow robots to sense continuously while moving.

There have also been other results in the field of mapping unknown environments, again many focus on robots that perceive their surroundings continuously [9]. Mostly the aim is not to reconstruct combinatorial properties of the surroundings, but rather the exact geometrical layout. More strongly related to our setting is the mapping of graphs [1,2], where robots have discrete motion and sensing capabilities similar to our model. But while this problem is more general than the task of reconstructing the visibility graph, it was shown to be unsolvable for general environments without the ability to mark visited vertices.

## 2 Notation

In this work we consider simple polygons only. We denote the $n$ vertices of a (simple) polygon $\mathcal{P}$ by $V=\{v_0, v_1, \cdots, v_{n-1}\}$, ordered along the boundary in counterclockwise (ccw) order. The polygon has a set of $n$ edges $E = \{e_0, e_1, \ldots, e_{n-1}\}$, where $e_i = (v_i, v_{i+1})$, $i = 0, \ldots, n-1$. Note that from now on all primitive operations on vertex and edge indices are modulo $n$. In addition we assume general position, i.e. no three vertices are allowed to lie on a line.

**Definition 1.** *Two vertices $v_i, v_j \in V$ form a* visible pair *in $\mathcal{P}$, if the line segment $v_iv_j$ lies entirely within $\mathcal{P}$ (in particular, $v_i$ forms a visible pair with itself for any $i$). We say $v_i$ and $v_j$* see each other *and write $v_i \leftrightarrow_{\mathcal{P}} v_j$. We drop the index $\mathcal{P}$ and simply write '$\leftrightarrow$', if the corresponding polygon $\mathcal{P}$ is clear from the context. We say a robot at vertex $u$* sees *vertex $v_i$, if $u \leftrightarrow v_i$.*

**Definition 2.** *We define* view$(v_i)$ *of vertex $v_i$ in $\mathcal{P}$, the* view *of vertex $v_i$, to be the set of vertices that $v_i$ sees in $\mathcal{P}$. Formally,*

$$\text{view}(v_i) := \{\, v_j \in V \,|\, v_i \leftrightarrow v_j \,\}.$$

We write $\text{view}_j(v_i)$ to denote the $j$-th vertex, $j \geq 0$, that $v_i$ sees in ccw order, starting at $v_i$ itself, both $\text{view}_0(v_i)$ and $\text{view}_{|\text{view}(v_i)|}(v_i)$ denoting $v_i$. The *view* of a robot at vertex $v_r$ is the view of $v_r$ and we simply write *view* if the corresponding robot and its position are clear from the context. Similarly, we write $\text{view}_i$ to denote $\text{view}_i(v_r)$.

When presenting algorithms for a robot we make use of a specific operation: The operation '**move to** $i$' moves the robot to the vertex $\text{view}_i$. If the robot is equipped with the corresponding sensor, it may also support the operation '**look back**' in which the robot determines the index $b$ such that $\text{view}_b$ is the vertex it came from during the previous **move to** operation. For other sensors we do not define operations explicitly, but rather let the robot access the measured information directly.

In the next section we introduce "combinatorial visibility" which allows robots to sense for every pair of vertices of the polygon $\mathcal{P}$ in its view, whether those vertices are neighbors on the boundary of $\mathcal{P}$. We show that the knowledge of all combinatorial visibilities is not sufficient for the reconstruction of the visibility graph of $\mathcal{P}$, even when combined with the knowledge of the exact interior polygon angle at every vertex.

In Section 4 we focus on sensors that can put distant vertices into spatial relation. One such sensor is able to measure the exact angle between the lines connecting the position of the robot with any two vertices in sight. Even when the angle measurement is not precise and the sensor can only distinguish between convex and reflex angles, we show that the visibility graph can be inferred, if we allow the robot to look back. In addition, we show that measuring exact angles is enough, if we give the robot a compass that provides a global reference direction.

## 3 Combinatorial Sensors

The *combinatorial visibility* of a vertex $v_i$ is given by a binary vector who's $j$-th element encodes whether the $j$-th visible vertex and the $(j+1)$-th visible vertex form an edge of $\mathcal{P}$ or not; we call this a *combinatorial visibility vector* $\mathrm{cvv}(v_i)$. The following definitions capture this more formally. Consult Fig. 1 along with the definitions.

**Definition 3.** *The* combinatorial visibility vector $\mathrm{cvv}(v_i) \in \{0,1\}^{|\mathrm{view}(v_i)|}$ *of vertex $v_i \in V$ is a binary vector with the $j$-th element, $j \geq 0$, given by*

$$\mathrm{cvv}_j(v_i) = \begin{cases} 1, \text{ if } (\mathrm{view}_j(v_i), \mathrm{view}_{j+1}(v_i)) \in E, \\ 0, \text{ else.} \end{cases}$$

Note that $\mathrm{view}_1(v_i) = v_{i+1}$ and $\mathrm{view}_{|\mathrm{view}(v_i)|-1}(v_i) = v_{i-1}$ as every vertex sees its neighboring vertices on the polygon boundary. Therefore $\mathrm{cvv}_0(v_i) = \mathrm{cvv}_{|\mathrm{view}(v_i)|-1}(v_i) = 1$ for all $v_i \in V$.

**Definition 4.** *The* combinatorial visibility sequence cvs *of $\mathcal{P}$ lists all combinatorial visibility vectors of the individual vertices of $\mathcal{P}$ in ccw order:*

$$\mathrm{cvs} := (\mathrm{cvv}(v_0), \ldots, \mathrm{cvv}(v_{n-1})).$$

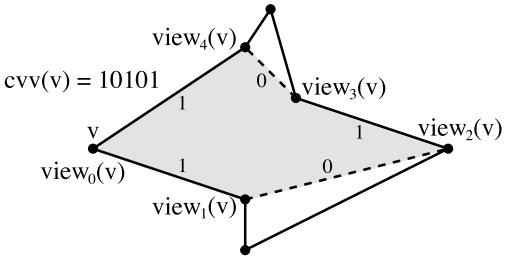



**Fig. 1.** Illustration of a combinatorial visibility vector

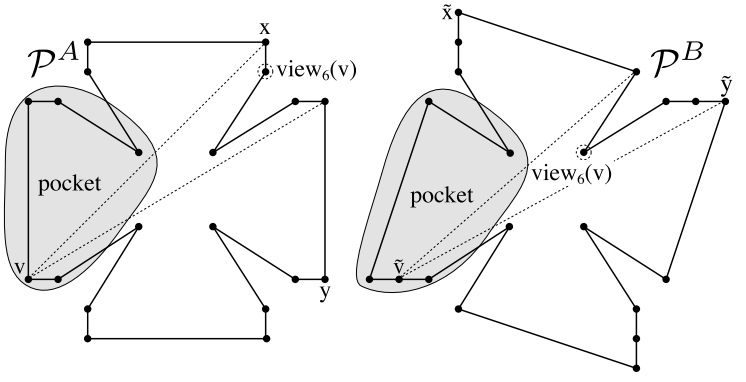


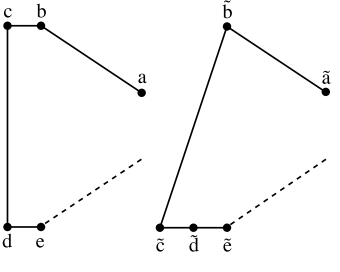

**Fig. 2.** Two polygons $\mathcal{P}^A$ and $\mathcal{P}^B$ with identical cvs and different visibility

The following result implies that a robot constrained to moving along the boundary of the polygon and to sensing combinatorial visibilities can in general not reconstruct the visibility graph of a polygon.

**Theorem 1.** *The* cvs *of a polygon $\mathcal{P}$ does not uniquely define its visibility graph.*

*Proof.* In Fig. 2 we present two polygons $\mathcal{P}^A$ and $\mathcal{P}^B$ which share the same cvs, yet have different visibility graphs. The proof is by inspection of the polygons together with the list of cvv's and view sequences of the relevant vertices in Fig. 3. Note that our construction is not in general position, however the polygons can easily be modified accordingly without changing visibilities or cvv's.

The idea behind the construction of the polygons is to use multiple copies of a "pocket" of vertices (cf. Fig. 2 for an illustration). Each pocket forms a convex curve, but the vertices connecting the pockets form reflex angles, resulting in a




| vertex | cvv | view sequence |
|---|---|---|
| $a$ | 1111101111011111 | $abcdeadeabcabcde$ |
| $\tilde{a}$ | | $\tilde{a}\tilde{b}\tilde{c}\tilde{d}\tilde{e}\tilde{a}\tilde{c}\tilde{d}\tilde{e}\tilde{a}\tilde{b}\tilde{a}\tilde{b}\tilde{c}\tilde{d}\tilde{e}$ |
| $b$ | 11110111101 | $bcdeadeabca$ |
| $\tilde{b}$ | | $\tilde{b}\tilde{c}\tilde{d}\tilde{e}\tilde{a}\tilde{c}\tilde{d}\tilde{e}\tilde{a}\tilde{b}\tilde{a}$ |
| $c$ | 11101111011 | $cdeadeabcab$ |
| $\tilde{c}$ | | $\tilde{c}\tilde{d}\tilde{e}\tilde{a}\tilde{c}\tilde{d}\tilde{e}\tilde{a}\tilde{b}\tilde{a}\tilde{b}$ |
| $d$ | 11011110111 | $deadeabcabc$ |
| $\tilde{d}$ | | $\tilde{d}\tilde{e}\tilde{a}\tilde{c}\tilde{d}\tilde{e}\tilde{a}\tilde{b}\tilde{a}\tilde{b}\tilde{c}$ |
| $e$ | 10111101111 | $eadeabcabcd$ |
| $\tilde{e}$ | | $\tilde{e}\tilde{a}\tilde{c}\tilde{d}\tilde{e}\tilde{a}\tilde{b}\tilde{a}\tilde{b}\tilde{c}\tilde{d}$ |

**Fig. 3.** The cvv and view sequence of every vertex within a pocket of $\mathcal{P}^A$ and $\mathcal{P}^B$, where $a, b, c, d, e$ each refer to all four vertices at the corresponding position within their pocket in $\mathcal{P}_A$ and $\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}, \tilde{e}$ refer to their counterparts in $\mathcal{P}_B$

non-convex polygon $\mathcal{P}$. The vertices inside a pocket thus do not see all vertices of $\mathcal{P}$, they see (apart from their own pocket) only parts of exactly two pockets. We use the fact that the vertices have no way to distinguish what pockets they are "looking into" and we modify the polygon $\mathcal{P}^A$ by shifting the vertex $c$ (cf. Fig. 2) so that in $\mathcal{P}^B$ the shifted vertex $\tilde{c}$ looks into different pockets, while not changing the cvv of any vertex. □

The polygons $\mathcal{P}^A$ and $\mathcal{P}^B$ have twenty vertices each, however we were also able to construct similar polygons for $n = 12$. We were able to show that no examples exist for $n \leq 10$. We do not expect there to be examples with $n < 12$, we have however not been able to prove this. We did not present the polygons with twelve vertices as their construction is more involved.

The following theorem considers a related question about polygons with periodical cvs. We start by defining periodicity formally:

**Definition 5.** *We say that a* cvs $C = (C_0, C_1, \ldots, C_{n-1})$ *with* $C_i = \mathrm{cvv}(v_i)$ *is* periodical *with period* $p \geq 2$*, if* $C_i = C_{i+k\cdot\frac{n}{p}}$ *for all* $0 \leq i < n$ *and all* $1 \leq k < p$*. For each* $0 \leq i < n$ *we say* $\{v_{i+k\cdot\frac{n}{p}} | 0 \leq k < p\}$ *are* periodical partners.

The question is whether two vertices visible from periodical partners at the same local position in a polygon with periodical cvs have to be periodical partners themselves. A positive answer to this question would have an impact on various interesting problems in the field of simple robots; for example, on a weak version of the *rendezvous* problem in symmetrical polygons in which two identical, deterministic robots try to gain sight of each other. We show that it is not the case; we even show a stronger result.

**Theorem 2.** *There is a polygon* $\mathcal{P}$ *with a periodical* cvs *of period* $p \geq 2$ *for which we have*

$$\exists v_i \in V\, \exists j \in \{1, \ldots, |\mathrm{view}(v_i)| - 1\} : \mathrm{cvv}(\mathrm{view}_j(v_i)) \neq \mathrm{cvv}\left(\mathrm{view}_j\left(v_{i+\frac{n}{p}}\right)\right).$$

*Proof.* We construct a polygon $\mathcal{P}$ with period $p = 2$ with the aforementioned property from the two polygons $\mathcal{P}^A$ and $\mathcal{P}^B$ in Fig. 2. The construction can easily be generalized to $p > 2$.

The idea of the construction is to "glue" $\mathcal{P}^A$ and $\mathcal{P}^B$ together at vertices $v$ and $\tilde{v}$ of $\mathcal{P}^A$ and $\mathcal{P}^B$, respectively, where $v$ and $\tilde{v}$ are as depicted in Fig. 2. We want to glue the polygons such that every two corresponding vertices $w$ and $\tilde{w}$ of the two polygons form periodical partners in $\mathcal{P}$. Thus, we need to glue the polygons such that the cvv's of corresponding vertices $w$ and $\tilde{w}$ are the same. We can then use the result of Theorem 1 which guarantees the existence of vertices $w$ and $\tilde{w}$ with the same cvv but different views. Formally, if $w$ from $\mathcal{P}^A$ is a vertex $v_i$ in $\mathcal{P}$ and $\tilde{w}$ from $\mathcal{P}^B$ is a vertex $v_{i+n/2}$ in $\mathcal{P}$ (where $n$ is the number of vertices of $\mathcal{P}$), there is a position $j$ in their views such that $\mathrm{view}_j(v_i) = v_k$ and $\mathrm{view}_j\left(v_{i+\frac{n}{2}}\right) = v_l \neq v_{k+\frac{n}{2}}$. Because of the structure of the two polygons, we will have $\mathrm{cvv}(v_k) \neq \mathrm{cvv}(v_l)$ which proves the theorem.

The problem when gluing at $v/\tilde{v}$ is that these vertices have to be split in the process, which makes them distinguishable from all other vertices. By inserting

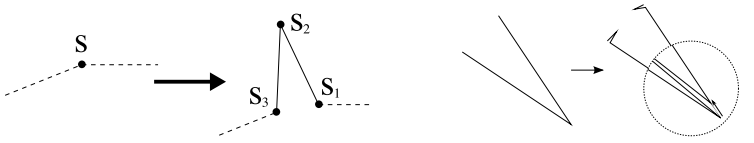



**Fig. 4.** Left: The concept of inserting spikes at vertices. Right: Illustration of how the spikes are inserted at reflex vertices. We chose our modification such that the right neighbor of the spike tip retains the visibility of the original vertex.

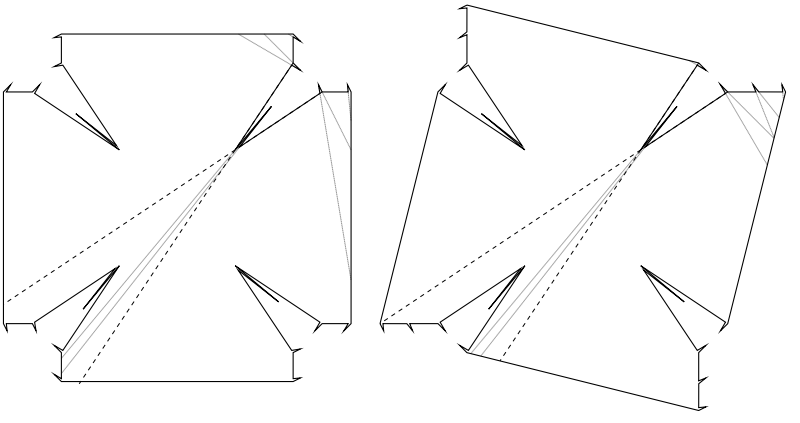
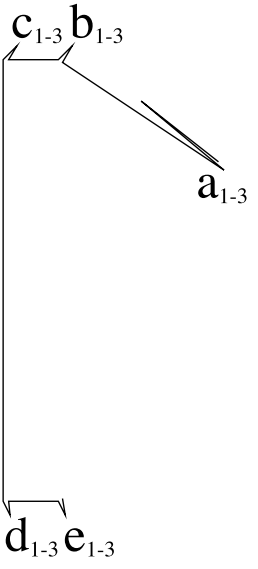


**Fig. 5.** The two polygons from Fig. 2 equipped with spikes and still with identical cvs. The areas visible from the different spike-tips are indicated.

spikes (cf. Fig. 4) at all vertices, we can again make vertices indistinguishable while still maintaining equal cvs'. Spikes can easily be inserted at convex vertices such that no distant vertex is visible from the spike tip and the spike tip's neighbors retain the vision of the original vertex (except for seeing vertices as gaps and seeing the spike tip). It is however not generally clear how to do that for reflex vertices, Fig. 4 shows how this can be done with the four reflex vertices in our case. Fig. 5 shows the spiked versions of $\mathcal{P}^A$ and $\mathcal{P}^B$ before gluing. Fig. 6 lists how the cvv's change with the introduction of spikes.

Once we have spiked versions of $\mathcal{P}^A$ and $\mathcal{P}^B$, we can glue them together in a straightforward way by simply splitting the spike tip of $v$ and $\tilde{v}$ and attaching the open ends. It can easily be seen that the gluing does not break the periodicity of the cvs of $\mathcal{P}$. Fig. 7 shows the resulting polygon $\mathcal{P}$. The extension to $p > 2$ is easily made, as we can attach more than two copies of the two spiked polygons around a common center. □

Theorem 1 shows that the knowledge of the cvs is not sufficient to reconstruct the visibility graph of a polygon. A natural question is how to extend this information "minimally" in order to make the reconstruction possible. In the following we




| vertex | cvv |
|---|---|
| $a_1$ | 1100101010100101010101 |
| $a_2$ | 101 |
| $a_3$ | 10101010100010101010010101010111 |
| $b_1$ | 1110101010001010101001 |
| $b_2$ | 101 |
| $b_3$ | 101010100010101010100111 |
| $c_1$ | 111010100010101010100101 |
| $c_2$ | 101 |
| $c_3$ | 101010001010101010010111 |
| $d_1$ | 111010001010101010010101 |
| $d_2$ | 101 |
| $d_3$ | 101000101010101001010111 |
| $e_1$ | 111000101010101001010101 |
| $e_2$ | 101 |
| $e_3$ | 100010101010100101010111 |

**Fig. 6.** The combinatorial visibilities of each vertex in a pocket of $\mathcal{P}^A$ after adding spikes (the same cvv's arise for $\mathcal{P}^B$). We write $v_{1-3}$ to denote the group of vertices $v_1, v_2, v_3$.

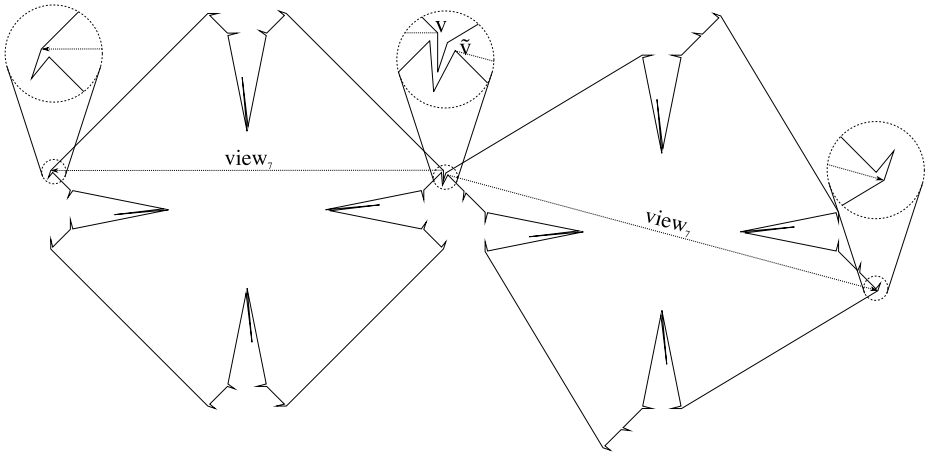



**Fig. 7.** The polygon with $n = 120$ that proves Theorem 2

show that adding the knowledge of all interior angles of the polygon is still not enough. We prove the following theorem.

**Theorem 3.** *The* cvs *and all interior angles of a polygon $\mathcal{P}$ do not uniquely determine the visibility graph of $\mathcal{P}$.*

*Proof.* Figure 8 shows a modified version of the polygons $\mathcal{P}^A$ and $\mathcal{P}^B$ of Fig. 2. As one can easily check, the polygons still have the same cvs and different visibility graphs. In addition, they also have the same set of inner angles at the vertices. The existence of such polygons proves the theorem. □

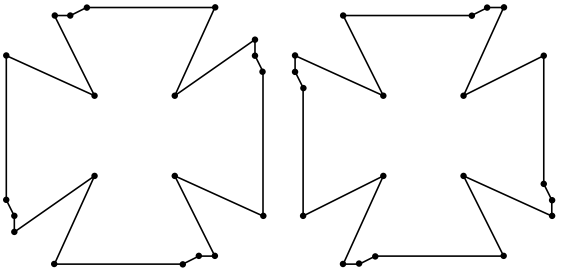


**Fig. 8.** Two polygons with identical cvs and identical interior angles but different visibility graphs. The visibilities are similar to those of $\mathcal{P}^A$ and $\mathcal{P}^B$.

Note that Theorems 1 and 3 do not imply that a robot equipped with sensors for measuring cvv's and/or inner angles cannot reconstruct the visibility graph, as such a robot would be able to distinguish $\mathcal{P}^A$ and $\mathcal{P}^B$ by moving to a vertex of the most distant corner and inspecting its cvv. However it seems difficult for such a robot to reconstruct the visibility graph - to prove that this indeed is not possible remains open.

## 4 Geometrical Sensors

In the previous section we saw that the simple combinatorial information we used is not enough to infer global properties of a polygon $\mathcal{P}$, namely its visibility graph. We now focus on geometrical sensors for measuring angles between distant vertices and show two sets of capabilities that enable a robot to reconstruct the visibility graph. We start by defining the two notions of angle sensors we consider.

**Definition 6.** *Let $v_r \in V$ be the vertex of polygon $\mathcal{P}$ which the robot is located at. We write* angle$(i, j)$ *with $i < j$ for the angle between the lines $v_r$*view$_i$ *and $v_r$*view$_j$ *in ccw direction. A sensor capable of determining* angle$(i, j)$ *for all $i, j$ is called* angle sensor. *The type of the angle* angle$(i, j)$ *('reflex' or 'convex' depending on whether the angle is larger than $\pi$ or not) is denoted by* angle_type$(i, j)$. *A sensor capable of determining* angle_type$(i, j)$ *for all $i, j$ is called* angle-type sensor.

While it is clear that the angle sensor is stronger than the angle-type sensor, the angle-type sensor has the advantage that it is very robust with respect to measurement imprecision.

We have preliminary results that suggest that an angle sensor alone suffices for the reconstruction of the visibility graph. It is however not clear whether the angle-type sensor alone is sufficient, even when combined with the combinatorial sensor of Section 3. For a robot equipped with an angle-type sensor that is allowed to look back (cf. Section 2) however, we are able to prove the following result:

**Theorem 4.** *A robot with an angle-type sensor and with the ability to look back can uniquely reconstruct the visibility graph of any polygon $\mathcal{P}$.*

*Proof.* We prove this by presenting an algorithm for the robot to construct the visibility graph.

The robot moves from vertex to vertex along the boundary of $\mathcal{P}$ in ccw order. At each vertex $v_i$ it iteratively identifies all visible vertices. It starts by identifying the vertices $\text{view}_1$, $\text{view}_{|\text{view}|-1}$ which trivially have the global index $i+1, i-1$. Further vertices can be identified as follows:

Let $v_k$ be the first visible vertex in ccw order that has not yet been identified and $v_j$ be the previous vertex that is visible, so that $j$ is known to the robot and it needs to find $k$. The robot does this by counting all vertices "beyond" $v_j$ and those "beyond" $v_k$. In order to understand the notion of vertices lying beyond some vertex $v_b$, consider the intersection $x$ of the ray from $v_i$ to $v_b$ with the boundary of $\mathcal{P}$. The vertices between (either in ccw order or in clockwise order) $x$ and $v_b$ are said to lie beyond $v_b$. The total number of the vertices beyond $v_j$ and $v_k$ (in ccw order and clockwise order, respectively) then simply needs to be added to $j+1$ in order to obtain $k$ (cf. Fig. 9).

It remains to be seen how the robot situated at $v$ counts the number of vertices beyond another vertex with local index $b$. The first step is moving to $b$. By looking back, the robot can identify $v$ in its new view. Therefore it can decide which of the now visible vertices form a reflex angle with $v$ and are thus

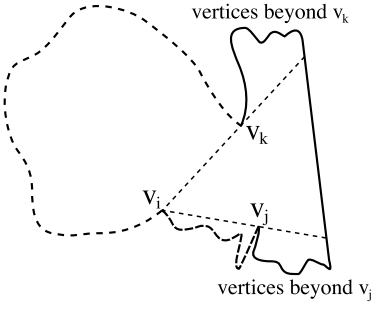



**Fig. 9.** Visualization of the procedure for inferring the global index of $v_k$ if the previous vertex $v_j$ has already been identified. It is enough to count the number of vertices beyond $v_j$ and $v_k$ as those are the ones between $v_j$ and $v_k$ in ccw order.

behind the current vertex $b$ when looking from $v$. All these vertices have to be counted as well as the number of vertices beyond them (which in turn are not visible to $b$). This is done recursively, so that at the end the robot sums up all vertices that are directly or indirectly behind $b$. The following listing shows the procedure for counting the vertices behind $b$ in pseudocode.

**function** beyond($b$,$o$)
**input**: local index $b$, order $o$ that specifies on which side of vision$_b$ to count
**output**: *count* of vertices beyond vision$_b$ w.r.t. the current position of the robot

1. $count \leftarrow 0$
2. **move to** $b$
3. $i \leftarrow$ **look back**
4. **for each** $j \in [1, \ldots, |\text{view}| - 1]$ with $(o = \mathbf{ccw} \wedge j < i) \vee (o = \mathbf{cw} \wedge j > i)$ **do**
5. **if** type$(j, i) =$ **reflex**
6. $count \leftarrow count + 1 +$ beyond$(j, \mathbf{ccw})$ + beyond$(j, \mathbf{cw})$
7. **move to** $i$

In order to prove the correctness of our algorithm, we need to show that no vertex is counted twice when counting the vertices beyond $v_j$ and beyond $v_k$. The two calls of beyond() for $v_j$ and $v_k$ consider distinct sets of vertices as the first considers those to the right of the line to $v_j$ and the second considers those on the left of the line to $v_k$. As $v_k$ by definition lies left of $v_j$, there is no overlap. We show that a single call to beyond() does not count vertices twice either: It is obvious that the recursive calls in line 6 consider distinct sets of vertices, as one considers only vertices on the left and the other only on the right of view$_j$. The only possible overlap could be between two calls of the form beyond$(x, \mathbf{ccw})$, beyond$(y, \mathbf{cw})$ with $x < y$. Again, because by definition view$_y$ lies to the right of view$_x$, there can be no overlap. The entire algorithm is at no point ambiguous, so that the solution found has to be unique. □

We can enable a robot with angle sensor to emulate the robot from Theorem 4 by giving it a *compass*. A compass provides the robot with a global reference direction. The angle sensor combined with a compass can measure the global direction to each vertex in sight.

**Definition 7.** *Let $p = (0, \infty)$. Let $v_r$ be the position of the robot and* view$'$ *be the view of the robot if $p$ was a vertex of $\mathcal{P}$ visible to $v_r$. A* compass *enables a robot to determine the index $i$ for which $p =$ view$'_i$. When combined with an angle sensor, a compass also provides the angles between the lines $v_r$view$'_i$ and $v_r$view$'_j$ in ccw direction, for all indices $j$.*

The next theorem follows immediately from Theorem 4.

**Theorem 5.** *A robot with an angle sensor and a compass can uniquely reconstruct the visibility graph of any polygon $\mathcal{P}$.*

*Proof.* The angle sensor can obviously emulate an angle type sensor. It therefore suffices to show that the robot can imitate the capability of looking back and

thus apply the strategy described in the proof of Theorem 4. Assume the robot moves from a vertex $v$ to a vertex $u$ that it sees in the global direction $d$. From its new location $u$ the robot knows that $v$ lies in direction $-d$. Because of general position, the robot is guaranteed to see only $v$ in that direction and thus the robot is capable of uniquely identifying the vertex it came from, in other words the robot is capable of looking back. □

Note that the last two results do not rely on the knowledge of $n$ and that in fact the corresponding robots are capable of inferring $n$.

## 5 Conclusion

We have studied the problem of reconstructing the visibility graph of a polygon $\mathcal{P}$ using simple robots. In this context, we have discussed three different configurations of sensors for simple robots. We have proven that the two configurations based on geometrical sensor enable the robot to infer the visibility graph of a polygon while purely combinatorial knowledge, in terms of the cvs of the polygon, does not suffice. In addition we have shown a property of symmetric polygons which makes combinatorial visibility even weaker in that case.

It is clear that a robot with one of the two geometrical sensor configurations is stronger than a robot equipped with the combinatorial sensor, as combinatorial visibilities can be derived from the visibility graph. The task of finding the weakest configuration that allows reconstructing the visibility graph remains unsolved.

## References

1. Dudek, G., Freedman, P., Hadjres, S.: Mapping in unknown graph-like worlds. Journal of Robotic Systems 13(8), 539–559 (1998)
2. Dudek, G., Jenkins, M., Milios, E., Wilkes, D.: Robotic exploration as graph construction. IEEE Transactions on Robotics and Automation 7(6), 859–865 (1991)
3. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Hard Tasks for Weak Robots: The Role of Common Knowledge in Pattern Formation by Autonomous Mobile Robots. In: Aggarwal, A.K., Pandu Rangan, C. (eds.) ISAAC 1999. LNCS, vol. 1741, pp. 93–102. Springer, Heidelberg (1999)
4. Ganguli, A., Cortés, J., Bullo, F.: Distributed deployment of asynchronous guards in art galleries. In: Proceedings of the American Control Conference, pp. 1416–1421 (2006)
5. Ghosh, S.K.: Visibility Algorithms in the Plane, 1st edn. Cambridge University Press, Cambridge (2007)
6. Hoffmann, F., Icking, C., Klein, R., Kriegel, K.: The polygon exploration problem. SIAM Journal on Computing 31(2), 577–600 (2001)
7. O'Kane, J.M., LaValle, S.: Localization with limited sensing. IEEE Transactions on Robotics 23(4), 704–716 (2007)
8. LaValle, S.: Planning Algorithms. Cambridge University Press, Cambridge (2006)

9. Oommen, B., Iyengar, S., Rao, N., Kayshap, R.: Robot navigation in unknown terrains using learned visibility graphs. Part I: The Disjoint Convex Obstacle Case. IEEE Journal of Robotics and Automation RA-3(6), 672–681 (1987)
10. Suri, S., Vicari, E., Widmayer, P.: Simple robots with minimal sensing: From local visibility to global geometry. International Journal of Robotics Research 27(9), 1055–1067 (2008)
11. Yershova, A., Tovar, B., Ghrist, R., LaValle, S.: Bitbots: Simple robots solving complex tasks. In: Proceedings of the Twentieth National Conference on Artificial Intelligence, pp. 1336–1341 (2005)