POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Nearly Optimal Algorithm for covering the interior of an Art Gallery

(Article begins on next page)

16 April 2024

A Nearly Optimal Algorithm
for covering the interior of an Art Gallery

Andrea Bottino (**corresponding author**), Dipartimento di Automatica e Informatica,

Corso Duca degli Abruzzi 24, 10129 Torino, Italy, mail: andrea.bottino@polito.it tel.

+390115647175, fax +390115647099, Aldo Laurentini, Dipartimento di Automatica e

Informatica, Corso Duca degli Abruzzi 24, 10129 Torino, Italy, mail:

aldo.laurentini@polito.it

Abstract— The problem of locating visual sensors can be often modelled as 2D Art

Gallery problems. In particular, tasks such as surveillance require observing the interior

of a polygonal environment (*interior covering, IC*), while for inspection or image based

rendering observing the boundary (*edge covering*, *EC*) is sufficient. Both problems are

NP-hard, and no technique is known for transforming one problem into the other.

Recently, an incremental algorithm for EC has been proposed, and its near-optimality

has been demonstrated experimentally. In this paper we show that, with some

modification, the algorithm is nearly optimal also for IC. The algorithm has been

implemented and tested over several hundreds of random polygons with and without

holes. The cardinality of the solutions provided is very near to, or coincident with, a

polygon-specific lower bound, and then suboptimal or optimal. In addition, our

algorithm has been compared, for all the test polygons, with recent heuristic sensor

location algorithms. In all cases, the cardinality of the set of guards provided by our

algorithm was less than or equal to that of the set computed by the other algorithms. An

enhanced version of the algorithm, also taking into account range and incidence

constraints, has also been implemented.

1. INTRODUCTION

The IC and EC Art Gallery problems

Problems of visual sensor placement arise in several practical areas of computer vision, computer graphics and robotics, such as surveillance, object or environment reconstruction, inspection and image based rendering. The subject is covered by several surveys, encompassing 2D and 3D problems ([21], [22], [18], [23]). The case of known objects or environment is usually referred to as *model-based* view planning ([22], [5], [24]).

Locating sensors to inspect, or surveying known real environments, requires satisfying a number of constrains for distance, incidence, lighting and visibility where, clearly, visibility is the main one. In several environments, such as buildings, sensor location is essentially a bi-dimensional problem where the area to observe can be modelled using polygons or sets of polygons. Assuming multidirectional or rotating optical sensors, the visibility constraint in 2D is modelled by the classic Art Gallery Problem, or by one of its variants. The original Art Gallery Problem asked to locate in a given polygon the minimum set of point sensors, or *guards*, able to completely observe, or cover, any point of the polygon (*IC, interior cover*). The famous Art Gallery Theorem states that at most $\lfloor n/3 \rfloor$ guards are required for covering polygons with n edges. Many variations of the problem have been considered, including particular kinds of polygons, restricted positions for the guards and additional constraints. Much work has been done for finding upper tight bounds (Art Gallery Theorems) in these cases. The decision problem related to the original problem (are k guards sufficient for covering a given polygon?), as well as those related to several similar problems, is NP-hard [6]. No exact finite algorithm for locating a minimum set of sensors is known. For further details, the reader is referred to the monograph by O'Rourke [19] and to the surveys by Shermer [20] and Urrutia [25] .

Not all practical sensor positioning problems require covering the whole interior of a polygonal environment. Tasks such as inspection and image based rendering, where the geometry, but not the texture, is known, only require to observe the boundary edges (*EC, edge covering*). The EC problem and its relations with IC have been analyzed in [17]. As IC, also EC is NP-Hard, and the same upper tight bound holds for the minimum number of guards, but in general an optimum, or minimal, set of IC guards is not an optimum set of EC guards and vice versa. No simple rule is known for transforming an optimal solution of one problem into an optimal solution of the other. Extreme examples can be produced, showing that IC may require 2 times more sensors than EC for simple polygons, and $\mathbf{O}(n)$ times more for polygons with holes [17], as in Fig. 1, where the dots represent the sensors of the EC solution and each gray region requires an additional IC sensor.
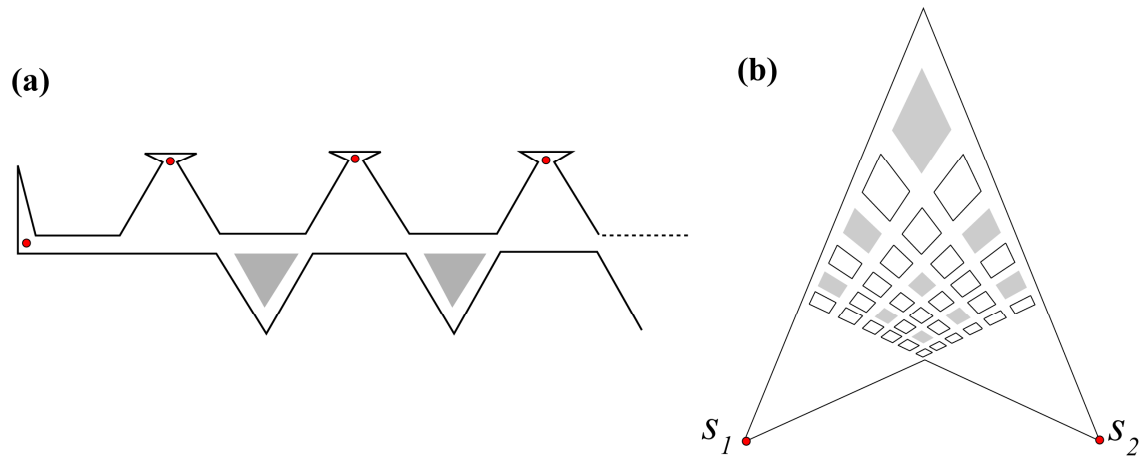

Fig. 1. (a) A family of simple polygons where the ratio between IC guards and IC guards can be arbitrarily near to two. (b) In this polygon with holes two guards s_1 and s_2 are sufficient for EC, but IC requires one additional guard for each area highlighted (a).

Approximate sensor location algorithms: state of the art

Although tight upper bounds (Art Gallery Theorems) are useful and theoretically interesting, practical problems require algorithms able to find and locate a set of guards whose cardinality is not far from that of an optimal or minimal cover. Unfortunately,

finding worst-case polynomial algorithm able to find solution with guaranteed fixed approximation is unlikely, according to the results of Eidenbentz et. al., [10], which show that several sensor location problems are APX-hard (APX hard means that the existence of such algorithms would imply P=NP). Then, approximate algorithms with *good average behaviour with respect to the optimal solution* are required.

Shermer, [20], reports several approximate polynomial algorithms for the IC problem. It can be easily verified that their worst-case performance in relation with the optimal solution can be very unsatisfactory ($\Theta(n)$ guards when $\mathbf{O}(1)$ are sufficient). Bjorling-Sachs and Souvaine [4] presented a polynomial algorithm that places always $\lfloor (n + h)/3 \rfloor$ guards, the upper bound, in a polygon with h holes. This technique could result in a large waste too, since it is easy to produce families of polygons were two guards are sufficient for any n and h. No data are supplied on the average behaviour of these algorithms.

Other approximate IC positioning techniques have been proposed more recently ([8], [9], [11], [12], [13], [27]). Several algorithms restrict the possible positions of the sensors, for instance on a grid, or on the vertices. Only a few of the IC algorithms have been implemented, and, in any case, their average behaviour with respect to the optimal unrestricted solution is unknown. Only one recent paper provides experimental results relative to the quality of the solution using a polygon specific lower bound [1].

As far as the EC problem is concerned, several attempts have been recently made for constructing practical algorithms. Kazakakis and Argyros [16] have implemented a heuristic that divides the polygon into a number of convex polygons, each of which can be inspected by a guard with visibility range restriction. Another approach consists, as for the interior cover, in selecting a cover among a large number of sensors located on a grid or at random. The randomized approach has been used by Danner and Kavraki [7], and Gonzales-Banos and Latombe ([14], [15]). Also in these cases, the quality of the

solutions obtained with respect to the optimum has not been investigated.

Our contribution

Recently, we have presented an incremental EC algorithm [2] for polygons with and without holes that converges towards the optimal solution in an unbounded number of steps. The algorithm locally refines a starting approximate solution provided by an integer edge covering (IEC) algorithm, where each edge is observed entirely by at least one sensor. A lower bound for the cardinality of the sensor set, specific of the polygonal environment considered, allows evaluating the quality of the solution obtained at each step. The algorithm can also take into account other geometrical constraints such as range and incidence. Tests performed over hundreds of random polygons have shown that the cardinality of the covers is always very close to, and in several cases coincident with, the lower bound, and therefore nearly optimal or optimal.

In this paper, we show that, with some modifications, the algorithm also applies to the IC problem. Extensive tests over more than 600 random polygons demonstrate that the modified algorithm is nearly optimal. In addition, we compare, for all the test polygons, the results of our algorithm with those obtained by some recently proposed sensor location algorithms [1]. It results that the cardinality of the solutions provided by our algorithm is always lower than or equal to that provided by these heuristics.

The new IC algorithm essentially exploits the fact that, in spite of the extreme cases of Fig.1, on the average an EC sensor set is extremely likely to be also an IC set, or, if not, is very close to an IC set.

The content of the paper is as follows. In Section 2, to make the paper self-contained we summarize the EC algorithm. Section 3 describes the new IC algorithm. In Section 4, we report the results of the tests performed over 600 random polygons of various categories, with or without holes and with various numbers of edges. In Section 5, we

compare our algorithm with two recent heuristic algorithms. Finally, the extension of the IC algorithm for accounting other constrains is discussed in Section 6.

2. SUMMARY OF THE EDGE COVERING ALGORITHM

The incremental Edge Covering (EC) has been presented in [2] and [3]. It starts from an initial solution optimal for the Integer Edge Covering (IEC) problem, where each edge is required to be entirely observed by at least one sensor. This solution consists of a set of convex polygons where the sensors must be located. A polygon specific lower bound, LB(P), is used to evaluate the quality of the solution. If the initial solution is not coincident, or sufficiently close to the lower bound and needs to be refined, the IEC algorithm is applied again after splitting in two some edges. A key component of the approach is the Indivisible Edges Algorithm (INDIVA), which allows finding *indivisible* edges. We call *indivisible* the edges that are entirely observed by one guard in some or all optimal solutions, and that therefore must not be split. The algorithm converges toward an optimal solution in an undefined number of steps.

The whole algorithm works as follows:

- Step 1. Compute a lower bound LB(P) for the cardinality of the minimum set of guards, specific for the polygon P, using the Lower Bound Algorithm (LBA).

- Step 2. Compute an integer edge cover of cardinality IECC using the Integer Edge Cover Algorithm (IECA).

- Step 3. Compare the lower bound and IECC. If they are equal, or the relative maximum error (IECC-LB(P))/LB(P) is less than a predefined threshold, STOP. Otherwise:

- Step 4. Apply algorithm INDIVA for finding indivisible edges. If all edges are indivisible, STOP, since IEC solution is optimal. Otherwise

- Step 5. Split in two the divisible edges and compute a new lower bound with LBA. Compare the new lower bound and the current IECC. If they are equal, STOP. Otherwise, go to Step 2.

For details about IECA and INDIVA, the reader is referred to [2]. Here we observe that the LB used in the present paper is that introduced in [3], which is higher than or equal to that described in [2] and thus better. This lower bound is defined as the cardinality of the maximal subset of not intersecting weak visibility polygons $W(e_i)$ of edges e_i, and visibility polygons $VP(v_i)$ of concave vertices v_i. We also underline the fact that the algorithm works for polygons with or without holes, which are dealt with exactly in the same way.

For understanding the changes introduced for constructing the IC algorithm, it is necessary to recall briefly some features of IECA. It produces a partition Π of the polygon into convex polygonal areas Z_i such that:

- The same set E_i of edges is entirely visible from each point of Z_i, $\forall i$

- The regions Z_i are *maximal* regions, that is $E_i \neq E_j$ where Z_j is any region contiguous to Z_i

Then, the algorithm selects the *dominant* and *essential* regions. A region Z_i is *dominant* if there is no other region Z_j such that $E_i \subset E_j$. An *essential* zone is a dominant zone that covers an edge not covered by any other dominant zone. In the final step, an instance of the set covering problem, the algorithm selects an optimal (or minimal) solution, consisting of a set $S = \{Z_{e1}, Z_{e2}, ..., Z_{ek}, Z_{d1}, Z_{d2}, ..., Z_{dh}\}$, containing all the essential zones Z_{ei} and some dominant regions Z_{dj}. Locating one sensor anywhere in each region we obtain a minimum cardinality set of sensors able to cover all edges.

A simple example will help to understand how the IEC algorithm works (see Fig. 2).

The lower bound LB for the polygon is two, since at most two not intersecting weak visibility polygons of edges or visibility polygons of convex vertices can be found. The areas highlighted in the figure are both the weak visibility polygons of the edges e_1 and e_2 and the visibility polygons of the concave vertices v_1 and v_2. The first application of IECA gives three polygonal regions where to locate the sensors. Since this exceeds LB, we perform a further step. In Fig. 2 (a), and in the following, point sensors located inside these regions are shown. Applying INDIVA, all edges, except e_3, are found to be indivisible. Splitting e_3, a further iteration of IEC gives two sensors (Fig. 2 (b)), and this solution is optimal, being equal to the lower bound.
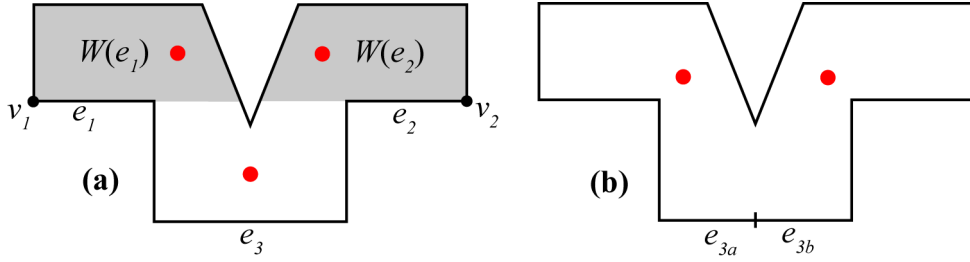

Fig. 2. A maximum cardinality set of non-intersecting weak visibility polygons and the initial IEC solution of size three (a). The final optimal IEC solution (b)

3. THE INTERNAL COVERING ALGORITHM (ICA)

First, let us observe that, as we will show in the following, for random polygons of any kind an edge cover *is very likely to be also an interior cover*. If this happens, since the EC lower bound also holds for IC, we have a cover that is also nearly optimal for IC. Hence, the IC algorithm (ICA) consists of the following steps:

Step 1. Apply to the polygon the EC algorithm and find an EC cover **S**

Step 2. Check if **S** is also an IC cover;

Step 3. If not, modify **S** for covering fully the interior.

In the following, we describe in detail Step 2 and Step 3.

Step 2. Checking the EC cover for full IC

The EC algorithm provides a set of convex polygonal regions where to locate each sensor for full edge covering. Checking full interior coverage requires to locate the sensors somewhere in these regions. Since we are not able to explore the infinite possible sensor arrangements, we choose to locate them at the centres of gravity of the regions. This also avoids locations near to the edges of the regions, which could put the sensors near lines containing edges [2] and thus produce poor visibility conditions. Even if there could exist better locations for interior covering, the experimental section will show that the quality of the algorithm is not seriously affected by this simplification.

The algorithm for checking if the EC cover is also an IC cover is the following:

- compute the m centres of gravity of the EC regions

- compute the visibility polygons of each centre of gravity

- merge all the visibility polygons and compare the result with the original polygon

Computing the centres of gravity takes O(nm) time. Computing the visibility polygon of a point takes O(n) time, and each visibility polygon has O(n) edges. Computing the union of two of them takes O($n^2 + n \log n$) time [32] and intersecting all of them takes O($mn^2 + mn \log n$) time, which is the overall complexity.

A preliminary test, carried on a set of 170 random polygons with 30 edges and summarized in Table 1, has shown that all but one of the EC sets were also IC sets. We will see in the following that similar results are obtained with all categories of random polygons.

Edges	Polygons	EC=IC
30	170	99,4%

Table 1: summary results of a preliminary experiment, showing that, for random generated polygons, the EC sensor set is very likely to be also an IC sensor set

Step 3: Modifying the EC sensor set to obtain an IC set

Although this is in practice a relatively marginal case, in order to avoid deteriorating the performance of the algorithm, we will first attempt to produce an IC set without adding new sensors. This could be possible since, in general, several optimal EC solutions exist even for relatively simple polygons, enclosing different sets of dominant regions (essential regions are common to all solutions). These various EC solutions cover more or less the interior, and one of them could be an IC solution.

An example is shown in Fig. 3. The lower bound LB(P) is 4 (Fig. 3(a)). A first run of IECA could provide the optimal solution shown in Fig. 3(b), where the sensors s_1, s_2, s_3 and s_4 are located at the centres of gravity of zones not shown in the figure. In particular, sensors s_3 and s_4 lie in essential regions, s_1 and s_2 in dominant regions. This is not an IC sensor set since the gray region highlighted in Fig. 3(b) remains uncovered. Actually, other EC sets of size 4 are also IC sets. For instance, consider the yellow dots d_1 and d_2, centroids of two dominant regions of Π. It is easy to see that both points observe the uncovered region in its entirety and that sensor s_1 can be substituted by any of these points, giving an optimal EC solution that is also an IC solution.
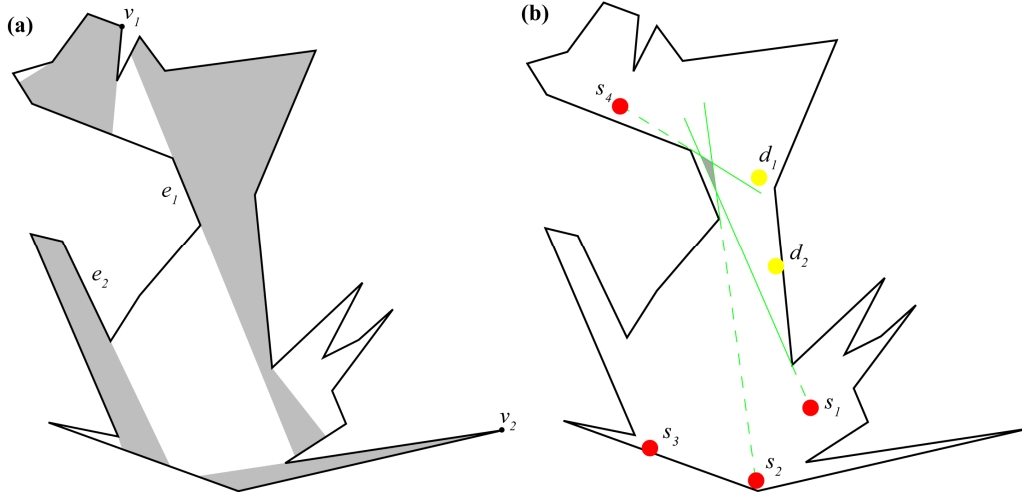
Fig. 3. The set of non-intersecting visibility polygons giving LB=4 (a). One EC set (sensors s_1, s_2, s_3 and s_4) that is not an IC set, while other optimal EC sets (for instance d_1, s_2, s_3, s_4) are also IC (b)

Thus, Step 3 of ICA consists of two Sub-steps:

- Sub-step 3.1- Look for another optimal EC solution that is also an IC solution

- Sub-step 3.2- If Sub-step 3.1 does not succeed, add some sensors for obtaining the IC cover

In the following, we describe these Sub-steps in detail.

Sub-step 3.1

To speed up the process, we implemented an heuristic algorithm that starts from the current EC solution $\mathbf{S} = \{Z_{e1}, Z_{e2}, \ldots, Z_{ek}, Z_{d1}, Z_{d2}, \ldots, Z_{dh}\}$, where Z_{ei} is an essential zone and Z_{di} is a dominant zone, and attempts to substitute some of the dominant regions in $\mathbf{S}$ with other dominant regions not in $\mathbf{S}$ which provide the same EC but better IC. The dominant zone candidates to the substitution are those that see some centroids of the uncovered areas, and are considered in decreasing order of observed centroids. In more detail, the algorithm is as follows.

1) Create the list $\mathbf{L}= \{Z_{c1}, Z_{c2}, \ldots, Z_{cp}\}$, of the dominant regions of the partition Π candidate for substitution in $\mathbf{S}$. These zones are those not enclosed in $\mathbf{S}$, and such that their centroids d_i are visible from the centroid of at least one of the uncovered regions of P (it is clear that the other dominant regions are not useful candidates). Then sort $\mathbf{L}$ in descending order of number of centroids of uncovered regions visible from each d_i

2) For each dominant region Z_{dj} in the current $\mathbf{S}$:

   o Inspect orderly the list $\mathbf{L}$ and construct the list $\mathbf{L}_j$ of dominant zones (if any) that cover (at least) the same edges of P covered exclusively by Z_{dj} (this is easily done since the algorithm that produces the partition Π also supplies the list of edges covered)

   o Substitute orderly each of the dominant region of $\mathbf{L}_j$ to the dominant region Z_{dj} in $\mathbf{S}$, and check if an IC cover has been obtained. STOP in this case, otherwise check if the interior area covered by the new set of sensors is greater than the previous one. If this is the case, confirm the substitution in $\mathbf{S}$, recompute the number of centroids of uncovered zones seen by the remaining candidate dominant regions, removing from $\mathbf{L}$ candidates with zero labels, and go back to 2. If all the dominant regions in $\mathbf{S}$ have been checked for possible substitution without obtaining an IC cover, go to Sub-step 3.2

An example, relative to the polygon of Fig. 3, can be seen in Fig. 4. The initial set $\mathbf{S}$ of regions gives the sensors $\{s_1, s_2, s_3, s_4\}$, where $\{s_1, s_2\}$ are centroids of dominant regions and $\{s_3, s_4\}$ of essential regions. U is the uncovered region, $d_1 \ldots d_6$ are the centroids of the dominant regions of Π not in $\mathbf{S}$. Since only $\{d_1, d_2, d_4, d_5\}$ are visible from the centroid of U, the list $\mathbf{L}$ is composed by the four corresponding regions. Let us

consider possible substitutions of the zone with centroid s_1; e is the edge seen exclusively by this zone. Inspecting **L** we find that the list L_1 is composed by the two zone with centroid $\{d_1, d_2\}$, which cover e in addition to other edges. Substituting the zone with centroid s_1 with the zone with centroid d_1, we find that that the new set provides full IC, and the algorithm stops supplying an optimal IC cover.
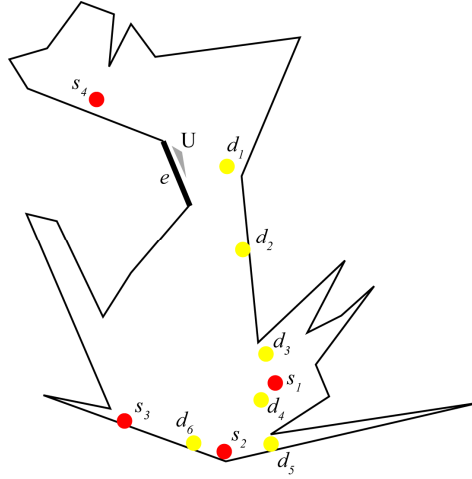

Fig. 4: the initial EC solution and the set of dominant regions of Π

Sub-step 3.2

If Sub-step 3.1 fails to produce an IC set, new sensors must be added to **S**. This is done with the following simple algorithm.

1. compute the centroid c_i of the regions left uncovered and assign to it a weight w_i given by the number of other centroids of other uncovered regions visible from c_i

2. repeat the following operations until **S** is an IC set:

   - add to **S** the centroid with the maximum weight

   - re-compute the weights of the remaining candidates

The algorithm described attempts to balance the quality of the solutions obtained when the initial EC set is not an IC set, with computation times.

Concerning the complexity of Step 3 of ICA, in sub-step 3.1 we have m regions that can be (potentially) substituted with p regions in list **L**, checking each time the new solution for internal coverage, at the cost of $O(pm^2n^2)$. As for sub-step 3.2, given q

uncovered regions, its complexity is $O(qmn^2)$.

4. EXPERIMENTAL RESULTS

The proposed approach has been thoroughly tested in order to assess the quality of the results. The algorithm has been applied to more than 400 random and custom polygons in addition to the 170 polygons of the preliminary test of Table 1. The randomly generated polygons used to test the algorithm belong to the following four categories:

- Generic random polygons, with edges oriented in random directions

- Orthogonal random simple polygons

- Random polygons with holes

- Orthogonal random polygons with holes

The generic polygons have been generated with a Random Polygon Generator (RPG), already used for testing the EC algorithm [2], based on the Delaunay triangulation of a set of point randomly distributed in a square region. The interested reader is referred to [30] and [31] for similar RPG approaches. The orthogonal polygons are in *general position*, that is they have no collinear edges, and have been generated with the RPG described in [26], whose code has been kindly supplied by the authors. Examples of generic random and orthogonal polygons with 50 edges are shown in Fig. 5 and Fig. 6, respectively.

For each category, four groups of twenty random polygons, each with 30, 40, 50 and 60 edges have been generated. The only exception is the group of generic polygons with 30 edges, which includes also the initial 170 samples of Table 1.

We recall that the iterative EC algorithm is halted when the solution is guaranteed to be optimal (cardinality equal to the lower bound, or no more divisible edges), when no improvements have been obtained in a predefined number of steps, or when a

predefined time has been exceeded. A time bound of 20 minutes and two consecutive iterations without improvement were chosen. Experiments were run on a Core2 processor at 2.66 GHz and 2GB of RAM.

The results of the tests summarized in Table 2-Table 5 contain the following data. E*dges* indicates the number of edges of each set of polygons. *C* is the average cardinality of the final IC solution. *LB* is the mean value of the lower bound at the last iteration. *UB* is the mean upper bound for each group. The tight bounds $\lfloor n/3 \rfloor$ and $\lfloor (n+h)/3 \rfloor$ ([17]) have been used respectively for random polygons and random polygons with holes, while the bound $\lfloor n/4 \rfloor$ ([33]) has been used for rectilinear polygons with and without holes. *EC=IC* is the percentage of polygons whose EC set is also an IC set, *IC=LB* the percentage of solutions which are guaranteed to be optimal, *G* the average absolute gap between the lower bound and the cardinality of the IC solution, *G/C* the average relative gap, *Maxdiff* the maximum difference over the set between cardinality of the solution and lower bound, and *Time* is the average time to compute the IC solution, in seconds.

Edges	C	LB	UB	EC=IC	IC=LB	G	G/C	MaxDiff	Time
30	4,2	4,0	10	99,5%	82,6%	0,2	5,0%	1	1,57
40	5,6	5,4	13	95,0%	75,0%	0,3	5,3%	1	2,97
50	6,7	6,4	16	100,0%	75,0%	0,3	5,4%	2	221,92
60	8,6	8,1	20	95,0%	60,0%	0,5	6,3%	2	271,50

Table 2: generic polygons

Edges	C	LB	UB	EC=IC	IC=LB	G	G/C	MaxDiff	Time
30	4,6	4,4	7	100,0%	85,0%	0,2	4,2%	1	1,08
40	6,1	5,9	10	100,0%	85,0%	0,2	2,8%	1	9,30
50	7,8	7,6	12	100,0%	80,0%	0,3	3,5%	2	6,41
60	9,3	9,0	15	95,0%	70,0%	0,3	3,4%	1	81,95

Table 3: rectilinear polygons

Edges	C	LB	UB	EC=IC	IC=LB	G	G/C	MaxDiff	Time
30	5,8	5,2	10,3	90,0%	55,0%	0,6	11,3%	2	88,53
40	7,0	5,7	13,7	95,0%	15,0%	1,4	25,9%	2	810,54
50	7,7	6,8	17	95,0%	40,0%	0,9	14,2%	3	637,39
60	8,5	7,4	20	95,0%	35,0%	1,1	15,1%	2	485,28

Table 4: polygons with holes (up to three holes)

Edges	C	LB	UB	EC=IC	IC=LB	G	G/C	MaxDiff	Time
30	4,8	4,5	7	90,0%	75,0%	0,3	6,3%	1	1,83
40	6,4	5,9	10	95,0%	50,0%	0,5	8,9%	1	70,87
50	8,5	7,7	12	85,0%	45,0%	0,8	11,1%	3	241,47
60	9,8	8,9	15	85,0%	30,0%	0,9	10,3%	2	209,57

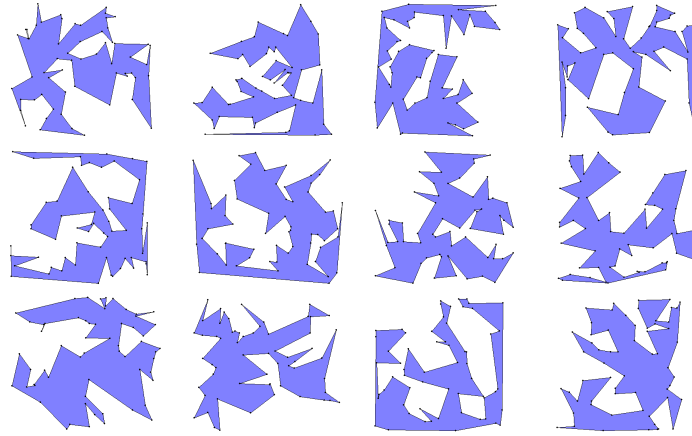Table 5: rectilinear polygons with holes (up to three holes)

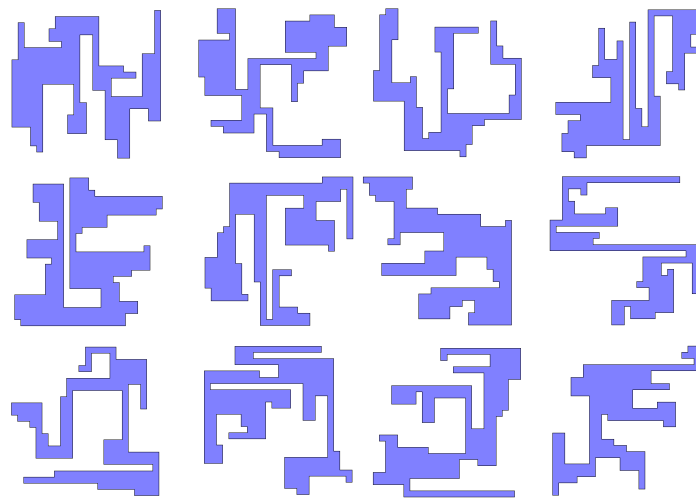Fig. 5: Examples of generic random polygons with 50 edges

Fig. 6: Examples of rectilinear random polygons with 50 edges

The experiments show that:

- Most of the times (about 96% of the cases), the EC sensor set is also an IC set

- The solutions provided by the algorithm are very close to the lower bound. For all the categories, several times the solution is guaranteed to be optimal (about 68% of the times) or, on the average, its cardinality exceeds LB by a small percentage. For instance, for random 40 edges rectilinear polygons, the solution is optimal (at least) in 85% of the tests, and the average percent difference is 2.8%. Therefore, our solutions can be considered nearly optimal.

As for the computational times, the results are comparable to that obtained in [2]. The time spent in checking if the EC sensor set is also an IC set is, in the worst case, lower than 1.5 sec.

5. COMPARISON WITH OTHER ALGORITHMS

As already mentioned, only few IC algorithms have been implemented, and only one recent paper provides experimental results about the quality of the solution using a polygon specific lower bound [1].

In order to assess the capabilities of our algorithm, we compared our results to those obtained by the algorithms proposed in [1]. For a given polygon, it determines several different IC sensor sets, starting from different sets of candidates, chosen according to various heuristics. We implemented the variants of the algorithm based on the heuristics A1 and A11, which according to the results presented in [1] were on the average the most effective. The algorithm starts from two initial sets C_1 and C_{11} of candidates composed in both cases by the vertices of the polygon P, and by the centroids of the regions of two different partitions determined by:

- the set of *edge extensions*, for heuristic A1,

- the set of *visibility extensions*, that is the lines separating regions of P where an edge is entirely or partially visible from regions where the edge is not visible, for A11.

Both partitions can be easily obtained from the partition Π used by our incremental EC algorithm, since edge and visibility extensions are a subset of the active lines used for Π (see [2] for details). Examples of edge and visibility extensions for a polygon P, and the resulting partitions, are shown in Fig. 7.
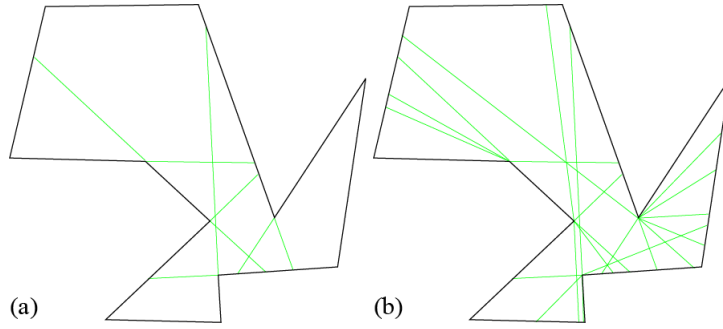

(a)   (b)

Fig. 7. Edge (a) and visibility (b) extensions

The IC solutions are built iteratively, adding to the current set G at each iteration the candidate with the largest score, until P is fully covered. For a candidate c, the score μ(c) at every iteration is given by the number of candidate points seen by c but not by any other point in G. A final post-processing step is applied. Since the solution could be redundant, each sensor is in turn removed from G. If the remaining set is still an IC sensor set, this sensor is discarded. The reader is referred to [1] for further details.

We run the two variants of the algorithm for all the polygons used in the tests presented in the previous section. The average results for each category are summarized in Table 6-Table 9, where *BIC*, *BA1* and *BA11* are the percentage of solutions for, respectively, our IC algorithms and the two heuristics A1 and A11, whose cardinality is equal to the minimum of the three results (and thus better then or equal to that provided by the other algorithms). *BA=LB* is the percentage of cases where the result of the best

heuristic between A1 and A11 is guaranteed to be optimal. *Diff* is the mean percentual difference between the best sensor cardinality obtained by A1 or A11 and the result of our algorithm.

Edges	IC=LB	BIC	BA1	BA11	BA=LB	Diff
30	82,6%	100,0%	48,4%	37,4%	45,8%	11,0%
40	75,0%	100,0%	20,0%	25,0%	10,0%	15,8%
50	75,0%	100,0%	25,0%	10,0%	15,0%	18,4%
60	60,0%	100,0%	15,0%	5,0%	0,0%	15,5%

Table 6. Generic polygons

Edges	IC=LB	BIC	BA1	BA11	BA=LB	Diff
30	85,0%	100,0%	55,0%	35,0%	45,0%	9,9%
40	85,0%	100,0%	40,0%	10,0%	35,0%	12,9%
50	80,0%	100,0%	20,0%	0,0%	15,0%	15,9%
60	70,0%	100,0%	15,0%	10,0%	10,0%	14,5%

Table 7. Rectilinear polygons

Edges	IC=LB	BIC	BA1	BA11	BA=LB	Diff
30	55,0%	100,0%	20,0%	0,0%	10,0%	17,6%
40	15,0%	100,0%	25,0%	15,0%	10,0%	11,6%
50	40,0%	100,0%	25,0%	5,0%	0,0%	12,8%
60	35,0%	100,0%	5,0%	0,0%	0,0%	19,8%

Table 8. Polygons with holes

Edges	IC=LB	BIC	BA1	BA11	BA=LB	Diff
30	75,0%	100,0%	30,0%	20,0%	25,0%	17,8%
40	50,0%	100,0%	25,0%	5,0%	0,0%	18,6%
50	45,0%	100,0%	20,0%	15,0%	5,0%	15,9%
60	30,0%	100,0%	10,0%	0,0%	0,0%	17,2%

Table 9. Rectilinear polygons with holes

The tables clearly show that our algorithm provides IC solutions whose cardinality is always lower than or equal to the cardinality of the solution provided by the algorithms A1 and A11.

Since for most random polygons the EC solution is also an IC solution, we also created a set of 21 custom polygons, constructed in such a way that the optimal EC and IC sensor set are different. Clearly, this situation, although unlikely, is strongly unfavourable for our algorithm. Some examples can be seen in Fig. 8. The results for

these cases are summarized in Table 10 where *Edges* is the mean number of edges of the polygons and, obviously, we did not report the *EC=IC* column. It is clear that none of the samples can reach LB, since it is the lower bound for the EC solution, which is, in these cases, lower than the IC solution. The table shows that, also in these unlikely cases, our algorithm is slightly better and never worse.
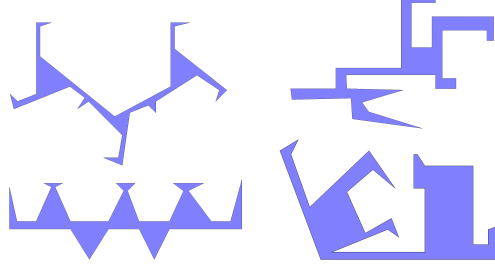

Fig. 8: Examples of polygons whose optimal EC and IC sensor sets are different

Edges	BIC	BA1	BA11	Diff
44,8	100,0%	61,9%	38,1%	3,5%

Table 10. Polygons whose EC is not an IC

For a fair comparison, we also discuss a very special case were these heuristics could provide better results. This happens when the polygons have edges aligned. Since the heuristics A1 and A11 are allowed to place guards also at the vertices, there are cases where a single vertex-sensor covers a region that requires two non vertex-sensors, as those used by our algorithms. An example is shown in Fig. 9. In the left part of the figure, the solution with 5 sensors given by the A1 heuristic is shown. This solution is also equal to the lower bound and therefore optimal. The 7 sensors given by our algorithm are shown on the right. One of the vertex-sensors on the left is highlighted, together with its visibility polygon. Clearly, this visibility polygon requires two different non vertex-sensors. Overall, our algorithm requires two more sensors since this situation occurs two times in the example. However, we observe that: a) in practical cases no real sensors is punctiform and can be placed exactly in the vertices b) in any

case, sensors aligned with an edge could not reliably observe the edge itself.
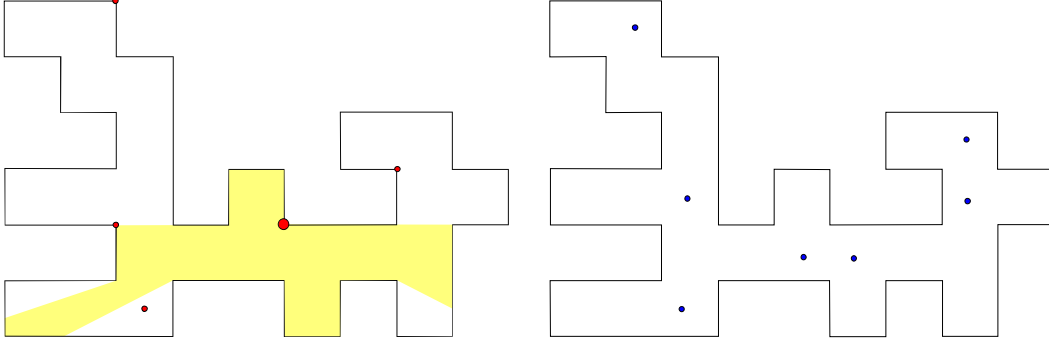

Fig. 9: A special case were heuristic A1 (left) is better than our algorithm (right)

6. TAKING INTO ACCOUNT RANGE AND INCIDENCE

In [2] and [3], we showed that the EC algorithm can take into account other geometrical constraints, namely minimal and maximal distances between the sensors and the observed boundary points, and minimal angle of incidence between an edge and the viewline. For each edge e_i these constraints define a restricted region $C(e_i)$ of P where the sensor must be located. These not polygonal regions can be easily computed, as well as the restricted visibility polygons $C(v_i)$ of convex vertices, required by the computation of LB(P). Examples of $C(e)$ and $C(v)$ regions for range and incidence constraints can be seen in Fig. 10. Observe that range and incidence cannot be arbitrarily fixed, otherwise the regions allowed could vanish. For instance, narrow corridor could not be covered if the minimal range is too large, and edges forming acute angles, for a given angle, could be not fully observable. Details of these problems can be found in [2] and [3].
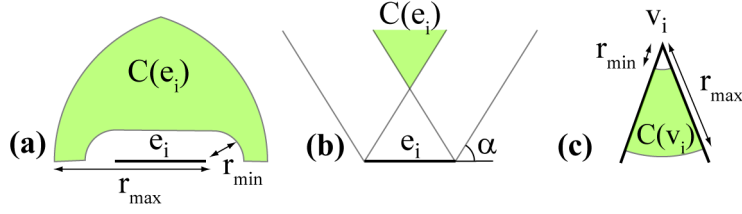
Fig. 10: C(e_i) for range (a) and for incidence (b) constraints; C(v_i) for range constraints (c)

In this section, we discuss range and incidence constraints for the IC algorithm. As for incidence, it is clear that the approach of the EC algorithm does not change for IC.

Dealing with range is more complex. For taking into account the maximal distance only, it is sufficient to restrict the visibility region of each sensor by intersection with a circle of ray r_{max} centred in the sensor. For sub-step 3.2 of ICA, if an uncovered region R is included in a circle of ray r_{max}, it can be covered with a single sensor. Otherwise, we select the location inside R that encloses the greater number of its vertices, and we add more sensors with the same rule until R is fully covered.

The current implementation of our IC algorithm takes into account incidence and maximum range constraints. An example, using maximal distance only, is shown in Fig. 11, where r_{max} has been defined as the 27.5% of the longest edge. After an initial edge splitting, since some edges are longer than r_{max}, we obtain a solution with 6 sensors, with 4 as lower bound. For each sensor, we also show its visibility region.
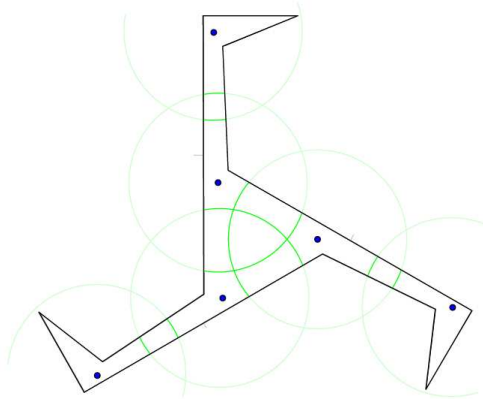

Fig. 11. The final IC solution, showing also the visibility regions of the sensors placed

Taking also into account minimal distance is much more complex, and cannot be easily implemented in our algorithm. The visibility polygon of any sensor s_i should be intersected with a doughnut region C(s_i), delimited by two circles of ray r_{min} and r_{max} centered in the sensor. Unfortunately, an EC solution is unlikely to be also an IC constrained solution, since each circle of ray r_{min} around the sensors must be covered by another sensor. In addition, no sensors can be placed inside possible uncovered regions *R*.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a novel IC sensor positioning algorithm, which is based on a recent EC incremental algorithm that provides optimal or sub-optimal solutions. The basic idea is that the EC sensor set is extremely likely to be also an IC sensor set or it can be easily extended into an IC one. The approach exploits a lower bound for the number of sensors, specific of the polygonal environment, which can be used to evaluate the closeness to optimality of the solution. The algorithm has been implemented and tested over about 600 random and custom polygons of various categories and with different number of edges. As expected, we have found that the initial EC solution is in most of the cases, about 96%, also an IC sensor set, and that the solutions provided by the IC algorithm are several times optimal, about 68% of the cases, or very close to the lower bound. Therefore, we can state that our IC algorithm is nearly optimal. Furthermore, the approach has been compared with other approximate sensor location algorithms reported in the scientific literature, showing better or equal performances.

We underline that, since the algorithm is incremental, even better results could be obtained at the expense of more computation time.

The algorithm can be easily extended to take into account several geometric constraints, and a version able to deal also with maximum range and incidence has been implemented.

Future works aim at extending the algorithm in 3D. A preliminary version of the 3D incremental boundary covering algorithm, which provides the coverage of the faces of a polyhedral environment, has been already presented in [28]. The present IC approach in 2D could be used as a basis for developing an IC algorithm in 3D.

REFERENCES

[1] Y. Amit, J. Mitchell and E.Parker," Locating Guards for Visibility Coverage of Polygons, "Workshop on Algorithm Engineering and Experiments, ALENEX, 07

[2] A. Bottino, A. Laurentini, "A Nearly Optimal Sensor Placement Algorithm for Boundary Coverage". Pattern Recognition 41 (11), 3343–3355. 2008

[3] A. Bottino, A. Laurentini, L. Rosano, "A Tight Lower Bound for Art Gallery Sensor Location Algorithms". Proceedings of 12th IEEE Conference on Emerging Technologies and Factory Automation. Patras - Greece. September 25-28, 2007

[4] I. Bjorling-Sachs and D.Souvaine, "An efficient algorithm for guard placement in polygons with holes," Discrete and Computational Geometry,pp 77-109, 1995

[5] S. Chen and Y.Li,"Automatic sensor placement for model-based robot vision," *IEEE Trans. On Systems, Man, and Cybernetics- Part B:Cybernetics*, pp.1-16, 2003

[6] J.C. Culberson and R.A. Reckhow," Covering polygons is hard," J Algorithms , Vol.17, pp. 2–44,1994

[7] T. Danner and L.E. Kavraki," Randomized planning for short inspections paths," IEEE Int. Conf. On Robotics and Automation,April 2002, pp. 971-976

[8] A. Efrat and S. Har-Peled, "Locating guards in art galleries," 2nd IFIP Interat. Conf. Theo. Comp. Sci., pp. 181-192, 2002

[9] A. Efrat and S. Har-Peled,"Guarding galleries and terrains," Info. Proc. Letters, Vol.100, pp. 238-245, 2006

[10] S.Eidenbenz, C. Stamm and P. Widmayer," Inapproximability results for guarding polygons and terrains," Algorithmica, Vol.31, pp.79-113, 2001

[11] A. Elnagar and L.Lulu, "An Art Gallery based approach to autonomous robot motion planning in global environments," Proc. IEEE Int. Conf. on Intelligent Robotics and Systems, pp.1367-1372, 2005

[12] A. Elnagar and L.Lulu," A comparative study between visibility-based roadmap path planning algorithms," Proc. IEEE Int. Conf. on Intelligent Robotics and Systems, pp.3700-3705, 2005

[13] U. M. Erdem and S. Sclaroff,"Automatic camera layout to satisfy task-specific and floor plan-specific coverage requirements," Comp. Vision and Image Understanding, Vol.103, pp. 156-169, 2006

[14] H. Gonzales-Banos and J-C. Latombe," A randomised art gallery algorithm for sensor placement," ACM SCG'01, June3-5,2001

[15] H. Gonzales-Banos and J-C. Latombe.”Planning robot motion for range-image acquisition and automatic 3D model construction,” Proc. AAAI Fall Symposyum Series, 1998

[16] G. D. Kazakakis, and A.A. Argyros, “Fast positioning of limited visibility guards for inspection of 2D workspaces,” Proc. 2002 IEEE/RSJ Intl. Conf. On Intelligent Robots and Systems, pp.2843-2848, October 2002

[17] A.Laurentini,”Guarding the walls of an art gallery,” The Visual Computer, vol.15, pp.265-278, 1999

[18] T.S. Newman and A.K. Jain, “A survey of automated visual inspection,” Comput. Vis. Image Understand. , vol.61, no.2, pp.231-262, 1995

[19] J. O'Rourke , Art gallery theorems and algorithms, Oxford University Press, New York,1987

[20] T. Shermer, “Recent results in art galleries,” IEEE Proc. Vol. 80, pp.1384-1399, 1992

[21] W.R.Scott and G.Roth,”View planning for Automated Three-Dimensional Object Reconstruction and Inspection,” ACM Comput. Surveys,Vol.35, No.1, March 2003, pp.64-96

[22] K.A. Tarabanis, P.K. Allen, and R. Y. Tsai, “ A survey of sensor planning in computer vision”, IEEE Trans. Robot. and Automat., vol. 11, no. 1 , pp. 86 –104, 1995

[23] G. Tarbox and S. Gottschlich,”Panning for completesensor coverage in inspection,” *Comp.Vis. and Image Und.,*Vol.61, pp.84-111, 1995

[24] E. Trucco, M.Umasuthan, A. Wallace, and V.Roberto,”Model-based planning of optimal sensor placement for inspection,” *IEEE Trans. Robot. Automat.* Vol.13, pp.182-194, 2007

[25] J. Urrutia, "Art Gallery and Illumination Problems" Handbook on Computational Geometry, Elsevier Science Publishers, J.R. Sack and J. Urrutia eds. pp. 973-1026, 2000.

[26] A. P. Tomás, A. L. Bajuelos, “Generating Random Orthogonal Polygons”. Current Topics in Artificial Intelligence: 10th Conf. of the Spanish Association for Artificial Intelligence, CAEPIA 2003, and 5th Conference on Technology Transfer, TTIA 2003. Revised Selected Papers, Lecture Notes in Computer Science 3040, pp. 364-373, 2004

[27] M. Couto, C.de Souza, and P. Rezende,” An exact and efficient algorithm for the Orthogonal Art Gallery Problem, “IEEE proc. XX Bras. Symp. on Comp. Graphics and Image Proc., pp.87-94, 2007

[28] A. Bottino, “Towards an Iterative Algorithm for the Optimal Boundary Coverage of a 3D Environment”, Proceedings of CIARP 2009, 2009

[29] P. Bose, A. Lubiw, J. I. Munro, “Efficient visibility queries in simple polygons”, Computational Geometry, Volume 23, Issue 3, November 2002, Pages 313-335

[30] T. Auer, M. Held,”Heuristics for the Generation of Random Polygons”. Proc. 8th Canad. Conf. Computat. Geometry, Ottawa, Canada, Aug 12-15, pp. 38-44, 1996

[31] D. Dailey, D. Whitfield, “Constructing random polygons”. In Proceedings of the 9th ACM SIGITE Conference 2008, pp. 119-124.

[32] F. L. Dévai, "On the Complexity of Some Geometric Intersection Problems", Proc. Int. Conf. on Computing and Information, 1994, pp. 333-352

[33] F. Homann, “The Art Gallery Problem for rectilinear polygons with holes”, Technical Report B 94-22, Freie Universiẗat Ber-lin 1994