The Surveillance of the Walls of an Art Gallery

Aldo Laurentini
Dipartimento di Automatica ed Informatica, Politecnico di Torino
laurentini@polito.it

Abstract
The "Art Gallery" problem consists in stationing in a polygon P a minimum set of guards such that each point of P is seen by at least one guard. In this paper we explore the related edge covering problem, *where the guards are required to observe the edges of P, metaphorically the paintings on the walls of the art gallery, and not necessarily every interior point. Minimum edge and interior covers for a given polygon are compared, and bounds and complexity for the edge covering problem analyzed. Then we introduce a restricted problem where full visibility of each edge from at least one guard is required, compare this problem with unrestricted edge covering and analyze its bounds and complexity. For this problem we present an algorithm that computes a set of regions where a minimum set of guards must be located. The algorithm can also deal with the external visibility of a set of polygons (the "Fortress" Problem).*

I. INTRODUCTION

With the word polygon we refer to a closed set which includes interior and boundary points. If the line segment connecting two points of a polygon P lies entirely in P, we say that they are *visible* from each other. A polygon P is *covered* by a set of viewpoints, or *guards*, lying in P, if each point in P is visible from at least one guard.

The research in this area was triggered in 1975 by Chvatal's "Art Gallery" Theorem[3]. He proved that at most $g(n)=\lfloor n/3 \rfloor$ guards are required for covering a simple polygon P with n edges. The upper tight bound $g(n,h)= \lfloor (n+h)/3 \rfloor$ for polygons with h polygonal holes and n edges has been proven about fifteen years later[5][6]. The reader is referred to the monograph of O'Rourke[1] and the more recent paper of Shermer[2] for comprehensive surveys of the results obtained in this area.

In spite of these results, the *minimum cover problem*, that is the practical problem of stationing a minimum set of $G(P)$ guards in a given polygon P, is still open. It can be reformulated as the problem of finding the minimum number of *star–polygons* whose union is P. A star–polygon is a polygon such that there exists a set of internal points, the *kernel*, whose members can observe the entire polygon. We will use the terms ICH and IC referring to the minimum cover problems for the interior of polygons with and without holes.

O'Rourke and Supowit have shown that the *decision* problem corresponding to ICH is NP–hard [9]. The same result for problem IC has been obtained by Lee and Lin[10].

Many important application problems are not polynomial, and many algorithms have been constructed which work sufficiently well with the usual inputs, even though some unlikely worst case might occur. Rather surprisingly, this is not the case for IC and CH: up to now no exact algorithm for finding and locating in a given polygon a minimum set of guards has been found.

Another approach to non–polynomial problems is to construct approximate algorithms with guaranteed performances. Also this approach seems unable to cope with the elusive nature of IC and ICH. We will show that the worst–case performance of the approximate guard placing algorithms is as bad as possible, and the simplification introduced have no practical rationale.

The problem addressed in this paper is the related *Edge Covering Problem,* where the minimum set of $GE(P)$ guards must cover the *edges* of the polygon P. If we consider the original meaning of the Art Gallery Problem, the surveillance of paintings on the walls is in effect the main task of the guards, even if some internal region could be left uncovered.

First we discuss the relations between minimum interior and edge covers for a given polygon, and analyze worst cases and complexity for the edge covering problem. Then we introduce the restriction that each edge be *entirely visible* from at least one guard, analyze this sub–problem and compare it with unrestricted edge covering. As a result of this restriction, we discretize the problem and construct an algorithm which provides, for general polygons with or without holes, a set of regions where a minimum set of guards can be located. The algorithm is exponential in the worst case, but its behavior appears sufficiently good for many practical cases. In addition, sub–optimal solution can be obtained in polynomial time. With small changes, the algorithm also applies to the "Fortress" Problem, where the external polygon is missing.

The content of this paper is as follows. Section II deals with the worst case behavior of approximate guard placing algorithms for interior cover. Complexity, worst cases and relation with interior cover of edge cover are analyzed in Section III. Entire visibility edge covering is analyzed in Section IV. In Section V we present the algorithm for entire visibility edge covering and the complexity analysis.

In this conference paper we present some results without detailed proofs. The interested reader can find proofs and full details on the algorithm in [29].

II. APPROXIMATE GUARD PLACING

In this section we survey the approximate guard placing algorithms known for the IC and ICH. Letting $G^A(P)$ be the number of guards located by an approximate algorithm A, we will show that the ratio $G^A(P)/G(P)$ is unbounded both for polygons without holes and with an arbitrary number of holes. For each algorithm we will produce families of polygons where $G(P)$ is a constant (1 or 2) and $G^A(P)$ can be made arbitrarily large. In addition, we will verify that the geometrical simplification proposed hardly have a practical rationale.

Stationing $g(n)$ and $g(n,h)$ guards. Avis and Tuissaint [8] and Bjorling–Sachs and Suvaine[5] gave polynomial algorithms for placing $\lfloor (n)/3 \rfloor$ and $\lfloor (n+h)/3 \rfloor$ guards in polygons without and with holes. Since for any value of

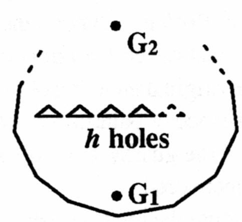


Fig.1–Worst case for stationing $g(n,h)$ guards.

n or of the pair (n, h) we can construct polygons which can be covered by one guard in the former case(a convex polygon), by two guards in the latter(Fig.1), the algorithms could place $O(n)$ and $O(n+h)$ more guards than necessary.

Vertex guards. IC with guards restricted to vertices is NP–complete[10]. A vertex cover which has at most $O(\lg n)$ time the minimum number $G^V(P)$ of vertex guards can be found in polynomial time[11]. Anyway, for polygons without holes $G^V(P)/G(P)$ can assume any value, as shown by the family of polygons in Fig.2 (a), where $G(P)=1$ and $G^V(P)$ is $O(n)$. The ratio is unbounded also for polygons with any number of holes, as shown by the example in Fig. 2(b). Two guards G_1 and G_2 are sufficient for covering a modified family of polygons where h tiny triangular holes are conveniently located in the regions highlighted .

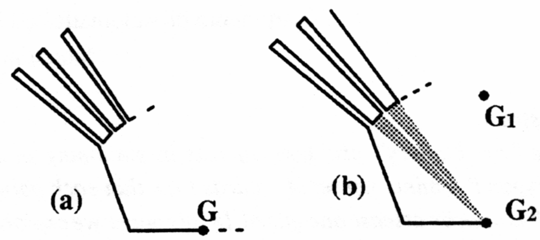


Fig.2–Worst cases for vertex guards.

Partitions. A partition is a cover with non–overlapping pieces. An algorithm for finding in polynomial time a *convex* partition with the minimum number $G^{CP}(P)$ of pieces has been found by Chazelle and Dobkin[12]. The problem is NP–hard for polygons with holes[13]. In any way, it is easy to guess that the ratio $G^{CP}(P)/G(P)$ is not bounded. This is shown for instance by the examples in Fig.2.

A partition into the minimum number $G^{SP}(P)$of star polygons seems more promising. A polynomial algorithm for this purpose, restricted to star polygons with vertices coincident with the vertices of P, was given by Keil[14]. However, also the worst–case behavior of any star partition is as bad as possible, as shown by the family of polygons found by Ntafos[15], (Fig.3(a)), where $G(P)=2$ and $G^{SP}(P)$ is $O(n)$. Also in this case we can modify these polygons by inserting any number of tiny triangular holes in the area highlighted without affecting the optimal cover (Fig.3(b)).

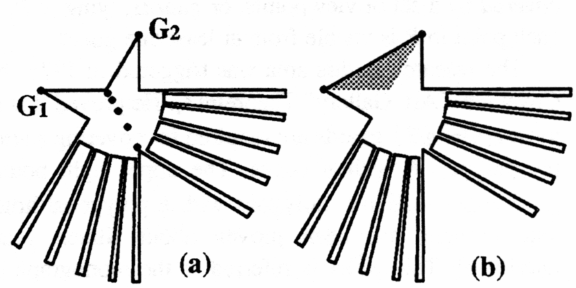


Fig.3– Worst cases for partition into star polygons.

Cover by restricted star polygons. Restricted star polygons have edges lying on the lines supporting the edges of P. Aggarwal et al.[16] gave a polynomial algorithm to find a cover by restricted star polygons that has at most $O(\lg n)$ times the $G^{RSP}(P)$ pieces of the minimal cover of this sort. A minimal cover could be found in finite exponential time. Restricted star polygons have been acknowledged to be unable in some cases to provide the minimal cover (see [1], Fig.9.3). Here we show that also this approximate cover can be as bad as possible.

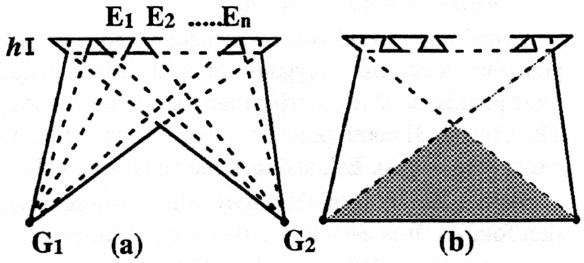


Fig.4–Worst cases for cover with restricted star polygons.

The minimum set of guards for the polygon without holes shown in Fig.4(a) consists of G_1 and G_2 at the bottom vertices. Covering each top edge requires both G_1 and G_2. Let us consider the top edges $E_1, E_2, ... E_n$. The kernel of any reduced star polygon covering E_i has an empty intersection with the kernel of any reduced star polygons covering E_j for $i \neq j$. Thus, a cover with reduced star polygon requires a different polygon for each edge. Reducing the height h, the example can be extended to any number of concavities, and $G_{RSP}(P)/G(P)$ can assume any value. The modified polygon in Fig.4(b), with tiny triangular holes in the shaded area, shows that $G_{RSP}(P)/G(P)$ is $O(n)$ also for polygons with holes.

Concluding, no approximate algorithm known to the author is guaranteed to provide in polynomial (or even exponential) time a cover close to the optimal solution. Moreover, the various simplifications introduced can hardly be justified on the ground of practical reasons.

III. COMPARING EDGE AND INTERIOR COVER

Even though covering the edges appears an interesting alternative to covering the interior, till now it received very little attention. Let *EC* and *ECH* indicate the edge covering problem for polygons without and with holes and the term *edge guards* indicate the guards required for this purpose.

Minimum edge and interior cover for the same polygon.

Let $GE(P)$ be the the minimum number of edge guards necessary for covering (the edges of) a polygon P. Obviously $GE(P) \leq G(P)$, since observing the interior implies observing the edges. It is not difficult to find examples where the minimum covers are different. For

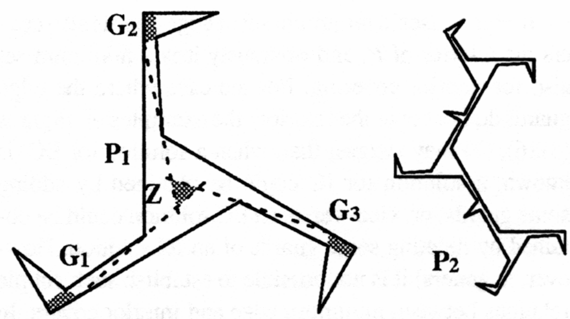


Fig.5–Three guards cover the edges of P_1, but four are needed for the interior. For P_2, the guards are 7 and 10 .

polygon P_1(Fig.5), three guards G_1, G_2 and G_3 , located anywhere in the areas highlighted, are the minimum solution for edge covering, but another guard is necessary for covering the central area Z. The difference $G(P)–GE(P)$ can assume any value (see polygon P_2 in Fig. 5, where the difference is three).

For the ratio $G(P)/GE(P)$ we have found the following results.

– *For polygons with holes,* $G(P)/GE(P)$ *can assume any value;*

– *For polygons without holes,* $(G(P)/ GE(P)) < 1.5$. *The difference* $G(P) - 1.5\, GE(P)$ *can be arbitrarily small.*

Proof. The first statement is proved by the example in Fig. 6. The polygon in the figure belongs to a family where $GE(P)$ is two and $G(P)$ is $O(n)$. The relatively complex proof of the second statement can be found in [29].

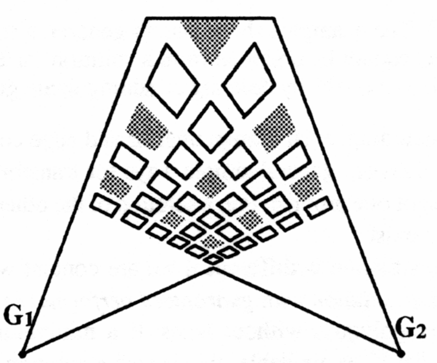


Fig.6– G_1 and G_2 cover all edges, but covering the interior requires one more guard for each area highlighted.

An example where $(G(P)/ GE(P))$ can be made arbitrarily near to 1.5 is shown in Fig.7

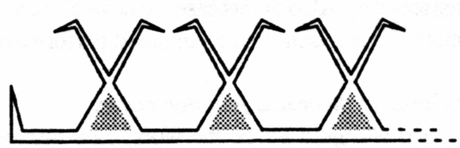


Fig.7– $G(P) - 1.5\, GE(P)$ *can be arbitrarily small.*

In some cases, a minimum set of edge guards also covers the interior of *P*, and obviously it is a minimum set also for interior covering. For the case where the edge guards do not cover the interior, the examples of Figures (5),(6), (7) may suggest that, when a solution for EC is known, a solution for IC could be obtained by adding some guards, or, vice–versa, an EC solution could be obtained by deleting some guards of an IC solution. However, in general it is not possible to establish such simple relations between minimum edge and interior covers. In fact, let us consider the polygon in Fig 8(a). In this case G_1, G_2 and G_3 are a solution for EC but not for IC since region Z is not covered. However, a minimum set of guard for IC is not obtained by adding a fourth guard for covering Z: only three guards as G_1', G_2' and G_3' are sufficient. The polygon in Fig.8(b) requires four guards for IC and three for EC. The four guards of Fig. 8(b) are a solution for IC, but no subset of three guards covers the edges.

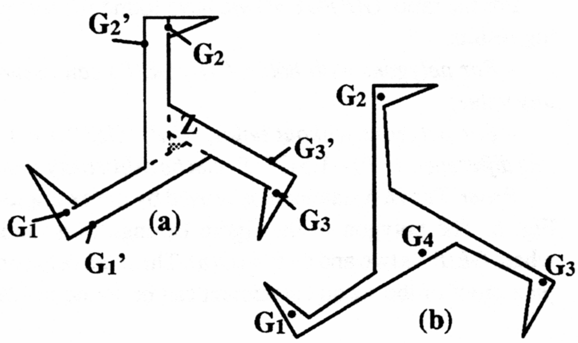


Fig.8–The examples show that, in general, a solution for IC cannot be obtained from a solution for EC(a) or vice–versa(b) by deleting or adding some guards.

Concluding, even though interior and edge cover are close relatives, general simple ways for transforming a solution of one problem into a solution of the other do not seem to exist .

The situation is different if we are content with *approximate solution with guaranteed performance*. Let us consider polygons without holes. If a minimum set of guards for IC is available, this is also a solution for EC which contains at most 1.5 times more guards than strictly necessary. Vice–versa, starting from a minimum set of EC guards, we can add in polynomial time at most $\lfloor (GE(P)-1)/2 \rfloor$ guards for covering the interior uncovered areas(see[29]). Also in this case we have obtained an approximate IC solution with guaranteed performance.

"Art Gallery" theorems for edge cover.

Let $ge(n)$ and $ge(n,h)$ be the worst case minimum numbers of point guards necessary to cover the edges of polygons with *n* edges without holes and with *h* holes respectively. We have:

– $ge(n) = g(n) = \lfloor n/3 \rfloor$

– $ge(n,h) = g(n,h) = \lfloor (n+h)/3 \rfloor$

Proof. $ge(n) \leq g(n)$, otherwise there would exist polygons(the worst case polygons for EC) which require more guards for edge covering than for interior covering. The Chvatal[3] comb polygon, a worst case for IC, is a worst case also for EC, and thus the equal sign holds.

It must also be $ge(n,h) \leq g(n,h)$, otherwise a contradiction follows. It is easy to see that the polygons used as worst cases for ICH([1], pp.128–129) requires $\lfloor (n+h)/3 \rfloor$ guards also for ECH , and thus the bound is tight.

Computational complexity of edge covering

Many covering and decomposition problems have been shown to be NP–hard(see O'Rourke[1], Shermer[2] and Culberson and Reckow[17]). We have found that:

– EC and ECH are both NP–hard

This can be shown by verifying that the constructions given by Lee and Linn[10] and by O'Rourke and Supowit [9] for reducing 3–SAT to IC and ICH also apply to EC and ECH. More details can be found in [29]

IV. THE ENTIRE VISIBILITY EDGE COVERING PROBLEM

Now we introduce a restriction for the edge covering problem: we require that *each edge must be seen in its entirety by at least one guard.* Unlike the restrictions seen for interior cover, entire visibility of the edges could make practical sense. Let EEC and EECH be the problems of finding a minimum cover of this kind for polygons without and with holes. In this section we compare these problems with unrestricted edge cover, and discuss their worst cases and computational complexity.

Entire edge cover versus edge cover for the same polygon.

Let $GEE(P)$ be the minimum number of guards required for the entire edge cover of a polygon *P*. Obviously $GEE(P) \geq GE(P)$. In Fig. 9 we show a polygon with holes and another without holes where these minimum numbers are different. Connecting together polygons as those shown in the figure we can easily obtain cases where $GEE(P) - GE(P)$ is $O(n)$.

What about the ratio $GEE(P)/GE(P)$? Were it bounded, a solution of the restricted problem could be an approximate solution with guaranteed performance for edge cover. Unfortunately this is not the case, and the ratio is unbounded both for polygons with and without

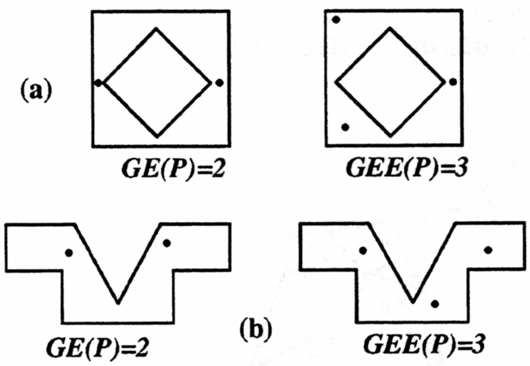


Fig.9–Two examples where entire edge cover requires one additional guard.

holes as shown by the examples already presented in Fig.4. Two unrestricted edge guards are sufficient in both cases, but each upper edge requires a different guard to be entirely observed.

Observe that $GEE(P)$ could be equal (for instance for the Chvatal comb polygon), greater(Fig.9) or smaller (Fig.5) than $G(P)$.

Finally, in general, a solution for EEC cannot be obtained from a solution for EC, or vice–versa, by deleting or adding some guards, as Fig.9(b) may suggest. This is shown by the example in Fig.9(a). No minimum set of $GEE(P)=3$ guards can be obtained adding one guard to the $GE(P)=2$ edge guards shown, and no two guards subset of the $GEE(P)=3$ guards shown covers the edges.

"Art Gallery" theorems for entire edge cover.

Let $gee(n)$ and $gee(n,h)$ be the worst–case minimum numbers of guards required by problems EEC and EECH. For polygons without holes we have:

$-gee(n) = ge(n) = g(n) = \lfloor n/3 \rfloor$

For the case of one hole, we have found :

$- gee(n,1) = \lfloor (n+2)/3 \rfloor$

The proofs can be found in [29]. An example where this bound is thight is shown in Fig.10. In this polygon no

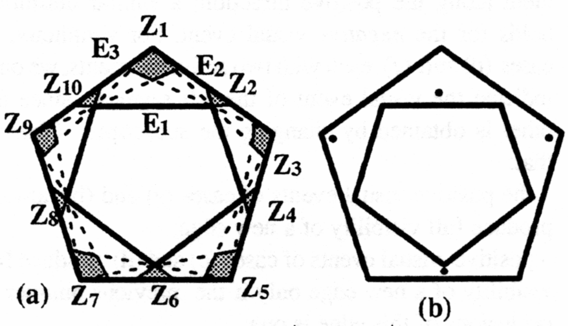


Fig.10. An example where $\lfloor (n+2)/3 \rfloor = 4$ guards are required .

guard can entirely observe four edges or more; from the regions highlighted in Fig.10(a) it is possible to entirely observe three edges. Thus 4 guards at least are required for observing the ten edges, for instance those in Fig.10(b). The polygon belongs to a family composed by a regular polygonal hole inside a larger regular polygon with the same number of edges. If the polygons are sufficiently close and $n=2m=3p+1$, with m and p integers, $\lfloor (n+2)/3 \rfloor$ guards are required.

For two or more holes, we conjecture:

$- gee(n,h) = \lfloor (n+h)/3 \rfloor$

Computational complexity of EEC and EECH

The reduction of 3–SAT to EC and ECH also hold for EEC and EECH, since both proofs construct polygons where each edge is entirely observed by at least one guard.

V. AN ALGORITHM FOR ENTIRE EDGE COVER

The entire visibility restriction allows to discretize the edge guarding problem. In this section we describe an algorithm for both problems EEC and EECH. Given a polygon P, the algorithm computes a set of polygonal regions. A minimum set of $GEE(P)$ guards can be obtained locating independently one guard anywhere in each region. A minor addition makes the algorithm also suitable for the "Fortress" or "external guarding" problem, where the edges of the holes are observed from an unbounded region.

The algorithm consists of the following steps:

Step 1–Compute a partition Π of P into regions Z_i such that:

– the same set $\mathbf{E_i}=(E_p, E_q, ...E_t)$ of edges is completely visible from all points of Z_i $\forall i$,

– Z_i are maximum regions, i.e. $\mathbf{E_i \neq E_j}$ for contiguous regions.

Step 2–Select the *dominant regions* (*d.r.*). A region Z_i is defined to be dominant if there is no other region Z_j of the partition such that $\mathbf{E_i \subset E_j}$.

Step 3–Select an optimal (or minimum) solution. A minimum solution consists of a set of dominant regions $S_j= (Z_{j1}, Z_{j2}, ...Z_{jk}....)$ which covers $\mathbf{E=\cup E_i}$ with the minimum number of members.

Observe that there could be minimal solution also containing non–dominant regions. For instance, in Fig. 10(a) the dominant regions are those highlighted. The upper guard in Fig. 10(b)does not lie in a dominant region.

We choose to consider only sets of dominant regions for the following reasons.

First, a non–dominant region can be substituted by a dominant region covering the same edges plus some more. Multiple coverage of edges appears preferable, for instance in the case of sensor failure. Second, we are looking for one optimal solution, not for all optimal solution: to consider only the dominant regions reduces the computations in Step.3, exponential in the worst case.

In the following of this section we present some details of the algorithm and the complexity analysis.

Step 1: Computing partition Π

We will divide *P* into maximal regions Z_i from which the same set of edges E_i is entirely visible, and label each region Z_i with the set E_i, using a visiting algorithm.

Let us discuss which lines are relevant to Π. Obviously, the lines supporting the edges are necessary. When a guard crosses line L_a as shown in Fig.11(a), the supported edge E_i turns entirely visible. Taking into account

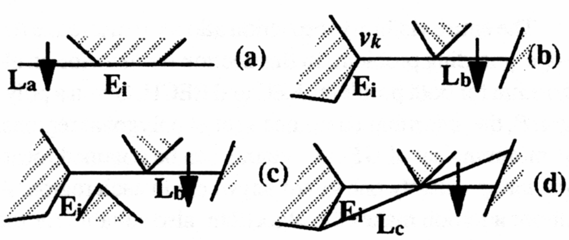


Fig.11–Examples of lines which change the visibility condition of an edge

occlusions requires other categories of lines. For instance, if a guard crosses line L_b in Fig. 11(b)), vertex v_k turns visible. If there are no other occlusions, as in Fig.11(b), the entire edge E_i to which the vertex v_k belongs turns visible. Otherwise, as in the case of Fig. 10(c), the edge turns visible only partially. Thus line L_b is potentially relevant to Π.

For dealing with such cases, we will compute Π as a refinement of a more detailed partition Π', which also contains potentially relevant lines, as L_b, and auxiliary lines. Auxiliary lines are not relevant to Π, but change the state of occlusion of an edge, as for instance L_c in Fig.13(d).

Before defining partition Π', let us first define the *aspect* A(G) of a point G belonging to *P* be:

$A(G) = ((E_h, n_h), (E_k, n_k), \ldots\ldots\ldots\ldots (E_q, n_q))$

where $E_h, E_k, \ldots E_q$ are the edges fully or partially visible from G, and $n_h, n_k \ldots n_q$ are the numbers of occlusions of these edges. For instance, the aspect relative to point G in Fig.12 is:

$A(G) = ((E_i, 0), (E_j, 3), (E_k, 0), (E_h, 0) \ldots\ldots)$

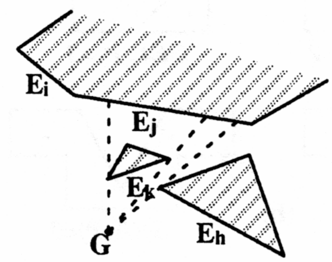


Fig.12– In the aspect relative to G, edge E_j has three occlusions.

Π' is defined as the partition which divides *P* into regions Z_i' such that:

– all points of Z_i' have the same aspect A_i

– Z_i' are maximum regions, i.e. $A_i \neq A_j$ for contiguous regions.

Clearly, to belong to the same region of Π' is a necessary, but not sufficient condition for two points to belong to the same region of Π.

Computing Π' is related to the computation of the *aspect graph* of a set of polygons. For further details on this matter, the reader is referred to [19], [20], [21] and [22]

Let us present now the catalogue of lines which form partition Π', and the associated *visual events*, or changes in the aspects. A line is defined to be *active* if it contains one or more *active segments*. An active segment is the boundary between points whose aspects are different.

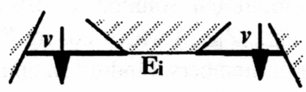
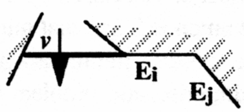
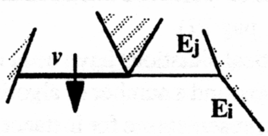
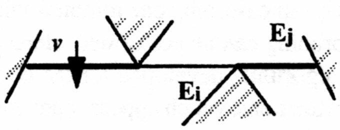
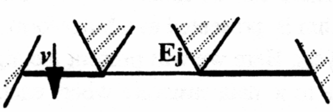
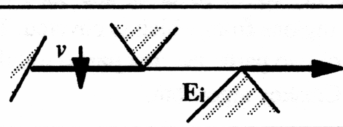
All possible active lines are those which join two vertices in the cases of Table 1. The active segments are highlighted with a thicker line. The arrows mark the *positive crossing directions*. The *positive visual event* is the change of aspect of a point which crosses the active segment along the positive direction; a similar definition holds for the *negative* visual event. For simplicity, in cases (d) and (e), each with two active segments, we only indicate the visual event of the left segment, since the other is obtained by changing the subscripts. Observe that:

– the positive visual events of cases (a) and (b) always produce full visibility of a new edge;

– positive visual events of cases (c) and (d) produce full visibility of a new edge only if the previous number of occlusions of this edge is one;

– visual events of case e) only affect the occlusion numbers.

TABLE 1

	Active lines	Positive visual event	Negative visual event
(a)		Add (E_i, 0)	Delete (E_i, 0)
(b)		$n_j \leftarrow n_j - 1$ Add (E_i, 0)	$n_j \leftarrow n_j + 1$ Delete (E_i, 0)
(c)		$n_i \leftarrow n_i - 1$ if E_j is already in the aspect: $n_j \leftarrow n_j + 1$; otherwise, add (E_j, 1)	$n_i \leftarrow n_i + 1$ if $n_j = 1$: delete (E_j, 1); otherwise, $n_j \leftarrow n_j - 1$
(d)		$n_i \leftarrow n_i - 1$ if E_j is already in the aspect: $n_j \leftarrow n_j + 2$; otherwise, add (E_j, 2)	$n_i \leftarrow n_i + 1$ if $n_j = 2$: delete (E_j, 2); otherwise, $n_j \leftarrow n_j - 2$
(e)		if E_j is already in the aspect: $n_j \leftarrow n_j + 1$; otherwise, add (E_j, 1)	if $n_j = 1$: delete (E_j, 1); otherwise, $n_j \leftarrow n_j - 1$
(d')		$n_i \leftarrow n_i - 1$	$n_i \leftarrow n_i + 1$

Note. A←B means that B replaces A in the aspect.

– case d') refers to the "Fortress" problem, i.e.,the case where there are only the polygonal holes and the region where the guards can be located is unbounded[1, pag.146]. In this case, an active segments could be unbounded. However, inspecting the previous cases in the Table we can verify that this only affects case (d). If the right active segment is unbounded, the positive and negative visual events reduce to the parts affecting E_i. This produces the new entry (d').

Computing Π requires to perform the following sub–steps:

1(a)–Computing the active segments of Π'

1(b)–Constructing Π'

1(c)–Refining Π' *into* Π *using a visiting algorithm.*

In the following we omit the details of the substeps, which can be found in [29]. In summary:

–$O(n^2)$ active segments are obtained in $O(n^2)$ time using Welzl's algorithm ([1] pag. 211).

–The partition Π' can be constructed using a plane sweep algorithm in $O(p \log p)$ time, where p is the number of vertices of the partition (regions and edges also are $O(p)$)(see[19]). For each edge of Π' lying on an active segment we store in the data structure the positive direction and the visual event.

–For computing Π' from Π some construction lines must be removed and the regions which these lines separate merged together. The time for computing the aspect of the starting region, visiting Π', computing Π and storing all aspects is $O(n^2+pn)$.

Adding up, the overall time bound of Step 1 is $O(n^2+p\log p+pn)$.

Step 2. Computing the dominant regions

Finding the dominant regions requires to compare the sets of fully visible edges $\mathbf{E}_i$ of each region Z_i. This process can be shortened observing that:

1) A necessary condition for a region Z_i of Π to be dominant is that $\mathbf{E}_j \subset \mathbf{E}_i$ for all the regions Z_j adjacent to Z_i, or, in other words, all the positive crossing directions of the edges of Z_i are toward the inside of the region (except for the edges of P).
2) If all the edges of Z_i (except for the edges of P)are due to cases a) and b) of Table 1, i.e., they lie on lines supporting edges of P, condition 1) is also sufficient.

The first statement is obvious. For the second, it is sufficient to observe that if the edges of Z_i lie on the lines supporting the edges $E_p, E_q,...E_k$, no other region is able to observe this set of edges. Thus, the dominant regions are found as follows:

– first visit all regions of Π and check condition 1) for selecting c candidate dominant regions in $O(p)$ time. Regions also satisfying condition condition 2) are immediately recognized to be dominant
– perform $O(c^2)$ comparisons in $O(nc^2)$ time for selecting d dominant zones. Adding up, Steps 1 and 2 of the algorithm require $O(n^2+p\log p+pn+nc^2)$ time. In Fig.13 we show both partition Π' and Π for a polygon with 9 edges and one hole and the 5 resulting dominant regions.

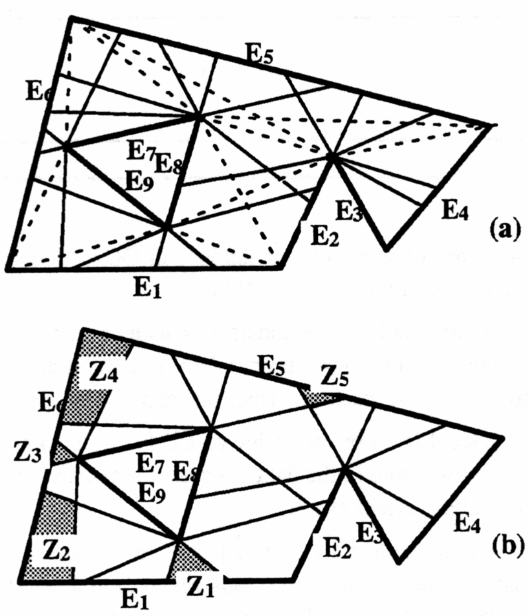


Fig.12–Partitions Π'(a) and Π (b) of a polygon. In Π' the active segments are solid, the rest of the active lines is dotted. Five dominant regions result. $\mathbf{Z_1}$, $\mathbf{Z_3}$, $\mathbf{Z_5}$ are immediately identified from the positive crossing directions of the edges.

Step 3. Finding an optimal solution

An optimal (or minimum) solution consists of a set of regions $\mathbf{S_j}$= $(Z_{j1}, Z_{j2},...Z_{jk}....)$ which covers $\mathbf{E}$ with the minimum number of members. Finding an optimal solution is an instance of the *set covering problem.* In general, given a set S and a number of subsets, an optimal cover is a set of subsets whose union is S and that minimizes the sum of the costs of the subsets. In our case all costs are equal. The corresponding decision problem (there is a cover with k subsets or less?) has been shown to be NP–complete(see [25], pag 37).

Numerous practical situations have been modeled as set covering problems, and a number of algorithms for set covering have been presented(see for instance[26], Chapter 13). When, as in our case, only one minimal solution is required, much pruning can be performed. Here we will briefly recall an algorithm, developed for the minimization of switching functions, which appears convenient for our case. Full details can be found in [27], Chapter 4, where the algorithm is presented in tabular form without complexity analysis. Here we will present the algorithm making reference to a data structure where each region has pointers to the edges covered, and each edge has pointers to the regions from which is covered. The algorithm consists of two parts; the first part is usually called the Quine–McCluskey algorithm.

Part 1(Quine–McCluskey algorithm) Perform (in any order) the following steps:

Step 1) Select the *essential* regions, i.e., those which cover at least one edge not covered by any other zone. Update the data structure by deleting these regions and the edges covered .
Step 2) Select and delete the *dominated* regions. A region Z_i is dominated by another region Z_j if the set of edges covered by Z_i is equal to or a subset of the set of edges covered by Z_j.
Step 3) Select and delete the *dominated* edges. An edge E_i is dominated by an edge E_j if the set of regions which cover E_j is equal to or a subset of the set of regions covering E_i.

Exit if all edges have been covered; go to Part 2 if none of the three steps reduces the data structure.

Part 2) Apply the branch–and–bound technique as follows. Select arbitrarily one edge and one region covering the edge. Create two sub–cases: in the first case the region is selected for the cover, in the second, the region is deleted from the structure without deleting the edges covered. Re–enter the Quine–McClusky algorithm for both cases. The bounding can be obtained by stopping the cover of a sub–case as soon as it going to have more regions of the best cover already found.

In some cases an optimal cover can be found without branching, as for the polygon in Fig.13. In this case, Step 1 (Fig.14 (a))finds one essential region (Z_5). Step 2

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9
Z_1	*	*						*	*
Z_2	*					*			*
Z_3						*	*		*
Z_4					*	*	*		
Z_5		*	*	*	*		*	*	

(a)

	E_1	E_6	E_9
Z_1	*		*
Z_2	*	*	*
Z_3		*	*
Z_4		*	

(b)

Fig.14. A tabular representation of the selection algorithm for the polygon of Fig. 13.

(Fig.14(b)) deletes the dominated regions Z_1, Z_3, Z_4. Step 3 has no effect, and eventually Z_2 is selected by Step 1. If we apply Step 3 immediately after Step 1, the edge E_9 dominated by E_1 is deleted. A subsequent application of Step 2 deletes again Z_1, Z_3, Z_4.

Let us consider the case of Fig.10, where some branching is required. No essential region is present since each edge is covered by three dominant regions. Thus we chose an arbitrary edge (E_1) and an arbitrary region (Z_1) covering E_1, E_2 and E_3. In the first sub–

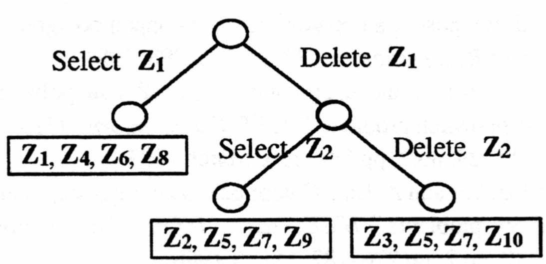


Fig.15–The branching required by the polygon in Fig.10. At each leaf of the tree there is one of the solutions provided by the Quine–McCluskey algorithm.

case(see Fig. 15), the Quine–McClusky algorithms obtains one cover without any further branching. It is easy to verify that: Step 2 deletes Z_2, Z_3, Z_9, Z_{10}; Step 3 deletes E_3, E_4, E_7; Step 1 selects Z_4 and Z_8; further applications of Steps 2 and 1 lead to the selection of Z_5 or Z_6 or Z_7. In the second case (delete Z_1 without deleting any edge) a further branching is necessary before re–entering the Quine–McCluskey algorithm. Also in these cases(we omit the details), the Quine–McClusky algorithm supplies one cover without any further branching. All the covers obtained are optimal.

Let us analyze the complexity of the algorithm. If no branching is required, $O(n)$ steps of Quine–McClusky algorithm are performed. Straightforward implementations of Steps 1, 2 and 3 are $O(n)$, $O(nd^2)$ and $O(dn^2)$. Adding up, the algorithm without branching is $O(n^2d^2+dn^3)$. In the worst case, branching is always necessary. Each branching is $O(n)$ and a binary tree with depth $O(d)$ and $O(2^d)$ nodes is constructed, thus the overall complexity is $O(n2^d)$.

A greedy near–optimal solution

Although the above algorithm is likely to perform satisfactorily in most cases, we are not able to make precise statements about its average behavior. Thus, a near–optimal solution obtained with a polynomial selection algorithm could be interesting. Such a solution can be obtained with a greedy heuristic, which selects each time the region which covers the largest number of uncovered edges. A straightforward implementation of this algorithm is $O(np^2)$. Although its performance cannot be guaranteed to be data–independent, it does not depend on the number of edges n. Let $GEE^G(P)$ be the number of regions obtained by the greedy algorithm and let r be the largest number of edges observed by a dominant region of P. It can be shown (see [28], pag.466) that:

$GEE^G(P)/GEE(P) \leq 1+\lg(r)$

Observe that r could be small even if n is large.

VI. SUMMARY AND DISCUSSION

Many efforts have been made and many results obtained in the "Art Gallery" area, but the main practical problem of stationing a minimum set of guards for covering the interior of given polygon, is still open. Up to now, neither exact finite algorithms, nor approximate algorithms with guaranteed performance have been found able to cope with this elusive problem. In this paper we have explored the related problem of covering the edges of a polygon with a minimum set of guards, neglecting possible interior uncovered areas.

The minimum edge and interior covers have been compared for a given polygon. In some case, a minimum set of edge guards also covers the interior. For polygons with holes however the interior cover could be much more requiring: we have found that the interior guards can be $O(n)$ times the edge guards. On the contrary, for polygons without holes the interior guards are at most 1.5 times the edge guards. For polygons such that the minimum numbers of edge and interior guards are different, no simple rule seems to exist for obtaining a minimum set of interior guards starting from a minimum set of edge guards or vice–versa. On the contrary, a guaranteed approximate solution for one problem can be easily obtained from a solution of the other problem for polygons without holes. Whether it be possible to obtain similar results for polygons with holes is an open question.

The worst case numbers of guards have been found to be equal for edge and interior cover, and the edge covering problem to be NP–hard for polygons with and without holes.

For the edge covering problem a restriction is possible which could make practical sense. The entire edge covering problem requires that each edge be entirely visible from at least one guard. Also this problem is NP–hard. For polygons without holes, the worst case number of guards is $\lfloor n/3 \rfloor$ as for the unrestricted problem. For polygons with one hole, we have found that at most $\lfloor (n+2)/3 \rfloor$ guards are always sufficient and sometime required. For more than one hole, we conjecture that $\lfloor (n+h)/3 \rfloor$ is the tight bound.

For the entire edge covering problem we have described an algorithm which computes a set of polygonal regions where the guards of a minimum set can be independently located. The algorithm is also suitable for the "Fortress" or "external guarding" problem. The last step of the algorithm is an instance of the set covering problem, exponential in the worst case. However, a selection algorithm, well known in the area of switching functions minimization and aimed at finding only one minimum solution, could show a satisfactory average behavior. In any way, a greedy selection supplies in polynomial time near optimal solutions within a factor only dependent on the logarithm of the largest number of edges observed by a guard, and independent on *n*.

In practice, placing visual sensors probably requires to face some additional constraint. A feature of the algorithm described is that it can easily be modified for taking into account geometrical restrictions. For instance, a maximum and a minimum distance from each point observed could be required. This constrains the guards required for observing each edge into a region whose boundary lines can be used for obtaining a modified partition Π^* of P. It is easy to see(we omit the straightforward details) how the sets E_i^* of entirely visible edges of each region Z_i^* of Π^* can be obtained. The rest of the algorithm is unchanged.

We have shown that a solution for the entire visibility problem is not an approximate guaranteed solution of the unrestricted edge covering problem. However, the observed difficulties of transforming a solution of the edge covering problems into a solution of the interior covering problem and vice–versa suggests some hope that for edge covering it be possible to solve some of the unsolved interior covering problems, as finding exact algorithms or approximate algorithms with guaranteed performance.

REFERENCES

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

[2] Thomas C. Shermer,"Recent results in art galleries,"*IEEE Proc.* Vol.80, no.9, pp.1384–1399, Sept.1992

[3] V Chvatal,"A combinatorial theorem in plane geometry,"*J. Combinatorial Theory,* ser. B, vol.18, pp.39–41, 1975

[4] S. Fisk,"Short proof of Chvatal's Watchman Theorem,"*J. Combinatorial Theory,* ser. B, vol.24, p. 374, 1978

[5] I. Bjorling–Sachs and D.L. Souvaine,"An efficient algorithm for guard placement in polygons with holes,"*Discrete&Computational Geometry,* Vol. 13, pp77–109, 1995

[6] F. Hoffmann, M. Kaufmann and K. Kriegel,"The art gallery theorem for polygons with holes,"*Proc. Symp. of Foundations of Comp. Science*, pp39–48, 1991

[7] F. Hoffmann,"The art gallery theorem for rectilinear polygons with holes,"Internal Rept. Dec.1994

[8] D. Avis and G. T. Toussaint,"An efficient algorithm for decomposing a polygon into star–shaped polygons," *Pattern Recognition,* vol 13, pp.395–398, 1981

[9] J. O'Rourke and K. Supowit,"Some NP–hard polygon decomposition problems," *IEEE Trans. Inform. Theory,* vol. IT–29, no.2, pp.181–190, March 1983

[10] D. Lee and A. Lin,"Computational complexity of art gallery problems," *IEEE Trans. Inform. Theory,* vol. IT–32, no.2, pp.276–282, March 1986

[11] S.Gosh,"Approximation algorithms for art gallery problems,"*Proc. Canadian Information Processing Society Congress*, 1987

[12] B. Chazelle and D.Dobkin,"Optimal convex decomposition," in *Computational Geometry,* G. Touissant, Ed. New York: North–Holland, 1985

[13] A. Lingas, "The power of non–rectilinear holes," in *Proc. 9th Colloq. Automata, Languages, and Programming, 1982, pp. 369–383*

[14] J.M.Kail, "Decomposing a polygon into simpler components," *SIAM J. Computing*, Vol.14, pp.799–814, 1985

[15] S. Ntafos,"On gallery watchmen in grids," *Info. Proc. Let.* Vol.23, pp. 99–102, 1986

[16] A. Aggarwal, S. Ghosh and R. Shyamasundar, "Computational complexity of restricted polygon decomposition," in *Computational Morphology*, G. Touissant, Ed. New York: North–Holland, pp.1–11, 1988

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

[18]F.Preparata and M.Shamos,"Computational geometry",Springer–Verlag,New York, 1985

[19] Z. Gigus, J. Canny and R.Seidel,"Efficiently computing and representing aspect graphs of polyhedral objects," *IEEE Trans.Pattern Anal. Machine Intell.*,Vol. 13, pp 542–551, June 1991

[20] H.Plantinga and C.R.Dyer,"Visibility, occlusion and the aspect graph," *Int. Journal of Comput. Vision,* Vol.5, No.2, pp.137–160, 1990

[21] M.Seibert and A.Waxman,"Adaptive 3–D object recognition from multiple views", *IEEE Trans. Patt. Anal. Machine Intell.*, vol.14, no.2, pp.107–124, February 1992

[22] J. A. Gualtieri et al., "The visual potential: one convex Polygon", *Computer Vision, Graphics and Image Processing*, Vol 46, pp. 96–130, 1989

[23] S.Baase, *Computer algorithms*, Addison–Wesley, New York, 1988

[24] H.Edelsbrunner,"Algorithms in combinatorial geometry", Springer–Verlag, Berlin Heidelberg, 1987

[25] R.G.Parker and R.L. Rardin," Discrete Optimization", Academic Press, San Diego, 1988

[26] H.M. Salkin and K. Mathur,"Foundations of Integer Programming", Elsevier, 1989

[27] S. Muroga,"Logic Design and Switching Theory", John Wiley & Sons, 1979

[28] G. Nemhauser and L.Wolsey,"Integer and Combinatorial Optimization,",John Wiley& Sons, 1988

[29] A.Laurentini,"Guarding the walls of an art gallery,"Int. Rept.1995