Treasures in an Art Gallery

Extended Abstract

Linda L. Deneen
Shashikant Joshi

Department of Computer Science
University of Minnesota, Duluth
Duluth, Minnesota 55812

Keywords: Art gallery problems, polygons, visibility, mobile guards.

Abstract

We define the *treasures in an art gallery problem* as this: given a simple polygon P with n vertices and t treasure sites in P, find the minimum number of mobile guards and their regions of location so that each treasure is completely visible from at least one guard region and each point in the polygon is weakly visible from at least one guard region. We show that this problem is NP-hard and that $t+1$ guards are sometimes necessary and always sufficient. We describe an $O(tn + r^2t^3)$ heuristic algorithm for guard placement, where r is the number of reflex angles in P.

1 Introduction

The original art gallery problem was posed by Victor Klee in 1973, and the first art gallery theorem was proved by Chvátal in 1975 [2]. Since that time, many variations on the problem have arisen. In 1987, Joseph O'Rourke wrote a monograph describing the work done on the problem up to that time [7].

The original art gallery problem is this: given a simple polygon P, the art gallery, how many stationary guards must be placed in the interior of the polygon so that every point in the interior of the polygon is visible from at least one of the guards? Chvátal proved that $\lfloor n/3 \rfloor$ guards are always sufficient and sometimes necessary to solve the problem for a polygon with n vertices [2].

In this paper we examine a new variation of the art gallery problem. Suppose we have a simple polygon P with n vertices, and suppose that in this polygon we have t sites at which invaluable treasures are to be located. What is the minimum number of mobile guards needed to guard the entire gallery and also to keep each treasure constantly in sight? While treasures must be kept constantly in sight, the rest of the gallery need only be checked periodically. Here we allow our guards unlimited mobility: a guard can walk anywhere in the interior of the polygon. The fact that the treasures must be kept constantly in sight, however, limits how far certain guards can move. Moreover, for polygons with complex shapes and for multiple treasures, it is easy to see that multiple guards are required.

In this paper, we define basic terms and describe fundamental results in Section 2. The problem of treasures in art galleries is described formally in Section 3. We discuss two heuristic algorithms for solving the problem in Section 4. A run-time analysis of the first algorithm is given in Section 5. Finally, conclusions are given in Section 6.

2 Basics

A *polygon* P is a sequence of vertices $(v_1, v_2, \ldots, v_n)$ and edges $\overline{v_1v_2}, \overline{v_2v_3}, \ldots, \overline{v_nv_1}$. For a simple polygon, no two non-consecutive edges should meet each other. The polygon consists of the interior and its boundary. A point $p \in P$ is said to be *visible* from a point $q \in P$ if the line segment pq lies completely inside P. Given a fixed site t in P, the *visibility polygon of* t, denoted $V(t)$, is the set of all points in P that are visible from t. $V(t)$ can be computed in $O(n)$ time, where n is the number of vertices in P [3, 4, 6].

In [1] Avis and Toussaint introduced the definitions of *complete*, *strong* and *weak visibility* of a polygon from a line segment. In this paper we are interested in the visibility of a single point from a guard as the guard moves along a line segment or throughout a region. Thus we extend Avis and Toussaint's definitions in the following way. A point $p \in P$ is said to be *completely visible* from a region R in P if it is visible from all points $r \in R$; it is *weakly visible* from R if it is visible from some point $r \in R$. We will be most interested in regions that are subpolygons of P or line segments in P. When R degenerates to a single point, both definitions reduce to the original definition of visibility.

In our problem we are interested in letting a guard range over as large a region as possible while still keeping a set of treasure sites constantly in sight. Therefore, we are interested in identifying regions from which a treasure site is completely visible. The following lemmas will assist us with this task.

Lemma 2.1 *Let P be a simple polygon, and let t be a fixed site in P. If $\overline{ab}$ is a line segment in P, and if t is visible from both a and b, then t is completely visible from $\overline{ab}$.*

Proof: Omitted.

Lemma 2.2 *Let P be a simple polygon, and let R be a subpolygon of P. If a site t in P is visible from all of the vertices of R, then t is completely visible from R.*

Proof: Omitted.

Figure 1 illustrates these lemmas. Both t_1 and t_2 are visible from v_4. Both t_3 and t_2 are visible from v_5. Lemma 2.1 shows that only t_2 is completely visible from edge $\overline{v_4v_5}$. Since t_2 is also visible from u_3 and u_4, then t_2 is completely visible from the subpolygon (v_4, v_5, u_3, u_4).

3 The Problem of Treasures in an Art Gallery

We formally define the *treasures in an art gallery problem* as this: given a simple polygon P with n vertices and t sites in P, find the minimum number of mobile guards and their regions of location so that each treasure is completely visible from at least one guard region and each point in the polygon is weakly visible from at least one guard region.

The treasures in Figure 1 can be guarded with two guards. The guard region for one guard is the polygon $(v_1, v_2, u_1, v_4, u_4, v_1)$. Both t_1 and t_2 are completely visible from this region. The second guard region is the polygon $(u_3, v_5, u_2, v_7, v_8, u_3)$, from which both t_2 and t_3 are completely visible. Every other point in P is visible from at least one point in the union of these two polygons.

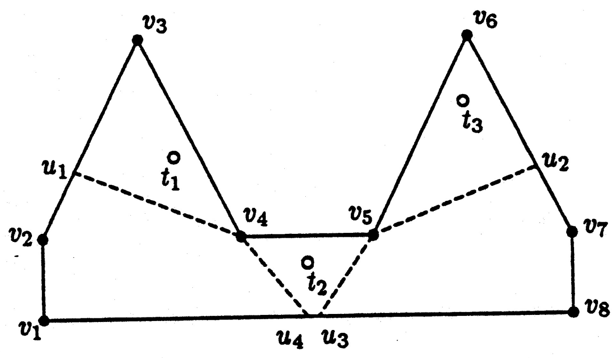

Figure 1: Regions and visibility

Theorem 3.1 *For the treasures in an art gallery problem with t treasures, $t+1$ guards are always sufficient and sometimes necessary.*

Proof: To see that $t+1$ guards are sufficient, place t of the guards at the t treasure sites, and let the remaining guard move around the boundary of the polygon to guard the remainder of the gallery. To see that $t+1$ guards are sometimes necessary, consider the comb polygon with $t+1$ teeth. (This is a generalization of Figure 1, where the points of the teeth are vertices like v_3 and v_4.) Place the t treasures at t of the points of the teeth. Then a $(t+1)$st guard is required to guard the $(t+1)$st tooth. ∎

Theorem 3.2 *The treasures in an art gallery problem is NP-hard.*

Proof: If we place a treasure site at each vertex of P, then the problem transforms into the original art gallery problem, which is known to be NP-hard [5]. ∎

4 A Greedy Heuristic Algorithm for the Treasures in an Art Gallery Problem

Because the treasures in an art gallery problem is NP-hard (Theorem 3.2), it is reasonable to explore heuristic algorithms for solving the problem. Such an algorithm is not guaranteed to find the minimum number of guards, but it should run fast and perform well for most reasonable problems.

We base a heuristic algorithm on the arrangement of line segments formed from the original polygon P together with the edges in the visibility polygons $V(t_i)$ for the treasure sites t_i. Figure 1 shows such an arrangement for the polygon and treasure sites given. The following theorem shows why this is the right structure to examine.

Theorem 4.1 *Let P be a polygon and let $\{t_1, t_2, \ldots, t_t\}$ be a set of sites in P. Let $V(t_1), V(t_2), \ldots, V(t_t)$ be the visibility polygons of $\{t_1, t_2, \ldots, t_t\}$ in P. Let A be the arrangement formed from P and the $V(P_i)$'s. Then the set of sites visible from any point in the interior of a region in A is constant throughout the region. Similarly, the set of sites visible from any point in the interior of an edge in A is constant throughout the edge.*

Proof: In order for a new site to become visible as we move from one point to another, we must cross an edge or vertex of the visibility polygon for the new site. ∎

Our primary data structure is a linked structure that stores the original polygon together with additional internal edges arising from the visibility polygons that form the arrangement. Each node of our data structure represents a vertex in the arrangement, and additional information about the vertex can be stored in the node as well. The fields in a vertex node are these:

Vertex Node:

v_num	Identifying label for the vertex; vertices on the boundary of the polygon are numbered in clockwise order
id	Index into an array of vertices
pt	Coordinates of the vertex
sites_list	Treasure sites visible from the vertex
e_count	Number of internal edges incident on the vertex
e_list	List of internal edges incident on the vertex
pnext, pprev	Pointers for a doubly-linked list of vertices of P
vnext, vprev	Pointers used to construct a visibility polygon
anext, aprev	Pointers used to construct the arrangement

In constructing this arrangement, we make use of the special properties of the polygon P and the visibility polygons that we are inserting. In particular, an edge of a visibility polygon that is internal to the polygon P must have one endpoint at a reflex vertex of P and the other on the boundary of P. For each such edge added to the arrangement, we compute the intersections with other such edges and store them as part of the e_list field in the vertex node. Each new vertex added on the boundary of P is inserted into the list of vertices maintained with the *anext* and *aprev* pointers.

To find the regions, we employ a process of *paring off* regions one at a time; each region found in this way is stored on a list of regions, together with a list of the treasure sites completely visible from the region. We start with regions that have only one internal edge. During this process, edges which have been internal to the polygon become external on the pared-down polygon. When we get to regions with two internal edges, an internal vertex of the arrangement then becomes a new vertex on the pared-down polygon. This process is fairly complex, and we do not describe it further in this extended abstract.

Next we give a high-level algorithm for guard placement based on this arrangement. In this algorithm, we iteratively make a greedy choice of the best region (one that guards the most remaining treasures) until all treasures are guarded.

Algorithm:

1. Initialize data structures.
2. Read in and store both the vertices of the polygon and the treasure sites.
3. Compute $V(t)$ for each treasure site t and store it in the polygon data structure to form an arrangement.
4. Construct a list of all of the regions in the arrangement by paring off one region at a time from the polygon. Store each region, together with the list of treasure sites it guards, in a separate list. Regions that do not guard any treasure can be discarded.
5. Choose guard regions from the list of regions in a greedy manner until all treasure sites are covered:

```
while not all treasures are guarded
      Choose the region that guards the most treasures
      Remove the treasures guarded from the remaining regions
end while
```

6. Add one remaining guard to patrol the boundary of P.

Notice that if we were willing to work harder at step 6 to determine whether the entire polygon is already weakly visible from the chosen guard regions, we might reduce the number of guards by one in some cases. It is unclear whether we want to work this hard to save at most one guard, but we are considering this as an improvement to the algorithm.

A slight modification of this algorithm gives rise to a second algorithm. This algorithm places guards at vertices of the arrangement and then tries to expand the range of the guard by searching from the chosen vertex to find a connected set of vertices that guard the same set of treasure sites. If this set forms a cycle, the region enclosed by the cycle can be included in the guard's walk. The rationale for this approach is that there may be single vertices or line segments that can guard more treasure sites than any proper region of the arrangement.

We are also interested in applying the technique of branch-and-bound to do a complete search of the possible guard placements within the arrangement, thus obtaining an optimal algorithm to solve the treasures in an art gallery problem. Such an algorithm would, of course, have an exponential run time.

5 Analysis

In this section we provide a run-time analysis of the first algorithm in Section 4. An analysis of the second algorithm is omitted from this extended abstract. The analysis is based on three parameters: the number of vertices n in P, the number of treasure sites t, and the number of reflex angles r in P.

Step 1 of the algorithm can be done in constant time, and step 2 takes $O(n+t)$ time. In step 3, we must compute t visibility polygons, and each one can be computed in $O(n)$ time [3, 4, 6]. In addition to computing these visibility polygons, we must also insert them into the arrangement. Since there are at worst r internal edges for each visibility polygon, there are at most rt internal edges in the arrangement, and there are at worst r^2t^2 intersections to be computed. This gives rise to a worst-case run time of $O(tn + r^2t^2)$ for step 3. Moreover, there are at worst $O(r^2t^2)$ regions in the resulting arrangement, so step 4 requires $O(r^2t^2)$ time. Step 5 looks at $O(r^2t^2)$ regions t times for a run time of $O(r^2t^3)$. Step 6 takes constant time. This yields a worst-case run time of $O(tn + r^2t^3)$ for the entire algorithm.

6 Conclusion

We have introduced a new variation on the art gallery problem and described two heuristic algorithms for solving it. These algorithms are based on an arrangement formed from the polygon and the visibility polygons associated with the treasure sites. The fact that we can compute this special arrangement of line segments in a reasonable amount of time contributes to the literature on arrangements.

We are currently implementing the first heuristic algorithm described in Section 4 and hope to have some empirical results to show soon. It should be easy to modify this implementation to obtain an implementation of the second heuristic algorithm we described. We are also interested in developing an optimal branch-and-bound algorithm for choosing guard regions based on the arrangement.

References

[1] D. Avis and G. T. Toussaint. An optimal algorithm for determining the visibility of a polygon from an edge. *IEEE Transactions on Computers*, C-30:910–914, 1981.

[2] V. Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory*, Series B, 18:39–41, 1975.

[3] H. Elgindy and D. Avis. A linear algorithm for computing the visibility polygon from a point. *Journal of Algorithms*, 2:186–197, 1981.

[4] B. Joe and R. B. Simpson. Corrections to Lee's visibility polygon algorithm. *BIT*, 27:458–473, 1987.

[5] D. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, IT-32:276–282, 1986.

[6] D. T. Lee. Visibility of a simple polygon. *Computer Vision, Graphics, and Image Processing*, 22:207–221, 1983.

[7] J. O'Rourke. *Art Gallery Theorems and Algorithms*. The International Series of Monographs on Computer Science. Oxford University Press, New York, 1987.