Placing observers to cover a polyhedral terrain in polynomial time*

Maurício Marengoni† Bruce A. Draper§ Allen Hanson† Ramesh Sitaraman†

Department of Computer Science†
University of Massachusetts
Amherst, MA 01003
marengon,hanson,ramesh@cs.umass.edu

Department of Computer Science§
Colorado State University
Ft. Collins, CO 80523
draper@cs.colostate.edu

Abstract

The *Art Gallery Problem* is the problem of determining the number of observers necessary to cover an art gallery room such that every point is seen by at least one observer. This problem is well known and has a linear solution for the 2 dimensional case, but little is known in the 3-D case. In this paper we present a polynomial time solution for the 3-D version of the Art Gallery Problem. Because the problem is NP-hard, the solution presented is an approximation, and we present the bounds to our solution. Our solution uses techniques from Computational Geometry, Graph Coloring and Set Coverage. A complexity analysis is presented for each step and an analysis of the overall quality of the solution is given.

1 Introduction

In this paper we consider the following 3-D visibility problem: *Given a 3-D terrain map, how many observers do we need to cover the whole terrain and where should we place them?* A *topographic terrain* is a graph of a continuous function that assigns to every point on the plane an elevation. In practice, the topographic terrain is discretized into a digital terrain model called a *Digital Elevation Map (DEM)*. The expression *cover the whole terrain* means that every point on the considered terrain will be visible by at least one of the observers.

This visibility problem and some of its variations (section 3) is a real world problem with several practical applications. The placement of antennas for cellular telephone companies, where the number of antennas has to be minimized, is one of them. A similar problem is to compute the coverage of a new set of antennas placed in some desired positions. The placement of cameras for security purpose on banks, supermarkets or department stores is another. In military scenarios, commanders need to place scouts to cover a certain region, or alternatively where to hide their resources.

This work was developed focusing on the military context. The Daedalus project [17] has as one of its goals to provide battlefield commanders with a powerful new tool for operations of planning and monitoring. The visibility problem solved here facilitates both planning and monitoring scenarios.

The 2-D "coverage" problem was posed by Victor Klee in 1973 and is better known as *The Art Gallery Problem* [13]. For a polygon with n vertices, $\lfloor \frac{n}{3} \rfloor$ observers are sufficient and sometimes necessary to cover the interior of the polygon. The first proof was given by Chvátal [4]; later Fisk gave a simpler proof, using triangulation of the polygon and, since a triangulated polygon is 3-colored, selecting the least used color will generate the bound [9]. The placement of observers can be done in linear time [11]. Alternative formulations of the 2-D coverage problem include orthogonal polygons, moving observers, polygons with holes and internal and external visibility of the polygons. For more details about the 2-D problem, its applications and solutions, see [13] and [18].

There are some similarities between the 2-D and the 3-D version of the problem, but little is known about observing an object in 3-D. In this paper an algorithmic solution to the 3-D version of the Art Gallery Problem is presented, providing an overall time complexity analysis. Many of the component algorithms have been implemented in Mathematica; a fully integrated system is currently being developed in C. The method presented here goes from an elevation map to an optimized placement of observers in polynomial time and within known bounds of the optimal solution.

1.1 General assumptions

There are two approaches for solving this problem. In the first, all points in the DEM are considered and the intervisibility of every pair of points is computed, as presented by Franklin and Ray [10]. The second approach models the terrain as a collection of disjoint triangles. This representation is called a *Triangulated Irregular Network (TIN)* in geographic information systems or a *Polyhedral Terrain* in computational geometry. This paper adopts the second approach and answers the questions of (1) how many observers are needed to cover a *polyhedral terrain*, such that every point on the polyhe-

*This work was supported in part by the National Council for Scientific Research- CNPq, Brazil under grant number 260185/92.2, in part by DARPA (via Army Res. Labs) under contract DAAL02-91-k-0047, NSF under grant CDA-8922572, DVRIP - Army Research Office grant number DAAH 04-95-1-0068, and NSF grant CCR-94-10077

dral terrain will be visible by at least one observer, and (2) where the observers should be placed. This approach was selected because it permits the simplification of the terrain model, and reduces the computation required to compute visibility.

When placing observers on a polyhedral terrain, the observer can be placed on an edge or on a vertex. When placing an observer on an edge it is possible to consider 3 forms of visibility: Let P be a polygon and e an edge of P.

- P is *completely visible* from e if every point of P is visible to every point of e.
- P is *strongly visible* from e if there is at least one point on e that can see all of P.
- P is *weakly visible* from e if every point of P is visible to some point of e.

In the last case P is covered if the observer moves over the edge e. All these definitions and more details can be found in [13]. Here only *vertex observers* are considered, that is, an observer can be placed only on vertices of the polyhedra. By definition of a polyhedral terrain, an observer placed on vertex v can see at least all the triangles that are adjacent to v. Furthermore it is assumed that the observer can not move and that it can see in all directions from vertex v. The observer's height is not considered, which is the most conservative approach to the visibility problem.

Section 2 gives the idea of the algorithm divided into several steps. Section 3 discusses other related problems that can be solved using the approach presented. Section 4 gives the conclusions obtained with this work.

2 Algorithm description

The proposed algorithm is divided into self contained steps for a better understanding of the overall system. The first step computes the terrain model from a DEM, the second step places observers in the polyhedral terrain considering only local visibility and the third step optimizes the number of observers.

2.1 The terrain model

The input data is a DEM where each entry *(X, Y)* in the image represents the elevation at the coordinate X, Y. We can build a terrain model by computing a triangulation of the points X, Y in the plane and giving each vertex a height corresponding to the elevation of point X, Y. The Delaunay triangulation, which is the dual of the Voronoi diagram [1], has the nice property that it maximizes the minimum angle of the triangles [14], thereby reducing the roughness of the approximating surface [16]. A Delaunay triangulation of a terrain used as an example in this paper is presented in Figure 1.

To precisely represent a terrain, millions of triangles are needed; the simple example presented in Figure 1 has 1426 triangles and 768 vertices. The advantage of working with a triangulated terrain is that it is possible to adjust the level of detail desired for a specific application. One can use the triangulation as presented in Figure 1, or if it is possible to

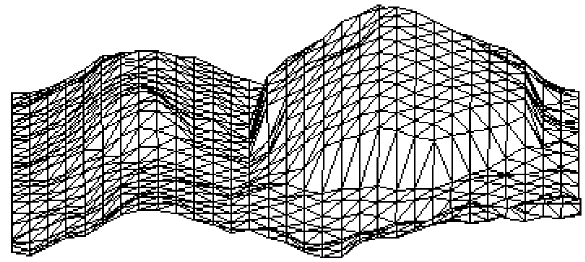

Figure 1: A Delaunay triangulation in 3-D of a terrain given as a DEM. The terrain here is a grid that has 32 by 24 points.

use a reduced model, the overall computation will be performed faster.

Mark de Berg [8] proposed a system that starts with a Delaunay triangulation of a terrain and simplifies the model to a desired level of detail preserving the property of the Delaunay triangulation. The interesting part of his approach is that it allows combinations of different levels of detail in the same model. This permits increased levels of detail in areas of specific interest, and less detail in other places. Because de Berg's algorithm works by removing vertices from the original terrain, in the final representation some important features might not be present. To avoid this problem it is possible to define a set of points V_{fixed} that are never removed from the terrain, keeping its original shape in the model. In this paper the set V_{fixed} is defined by sampling selected points on the terrain's level map; this preserves the overall shape of the final model. An example of the output of this algorithm is presented in Figure 2 and in Figure 3 as a planar graph.

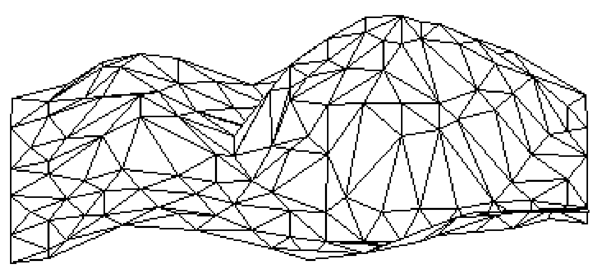

Figure 2: A Delaunay triangulation in 3-D for the simplified model.

The points marked with a square in Figure 3 represents the set of fixed points, V_{fixed}, given by the level curves of the DEM. This planar graph is now used to place the first set of observers considering only local information.

2.2 The first placement

The first placement of observers is done based only on local visibility. If an observer is placed on a

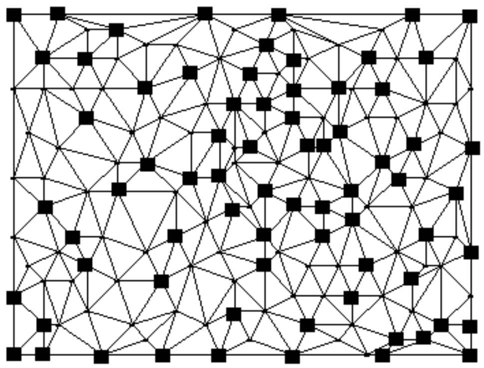

Figure 3: A Delaunay triangulation presented in Figure 2 as a planar graph. There are 278 triangles, 150 vertices and 72 fixed points.

plane and there are no obstacles in this plane then the observer can see the whole plane. Because each triangle is a plane and an observer is placed on a vertex (which is part of the plane), the observer can see the whole triangle adjacent to that vertex. Because a vertex is shared, in general, by more than one triangle, an observer placed in a vertex can see all triangles that share such a vertex. This is called local visibility; later this condition will be relaxed to reduce the number of observers.

Bose et al. showed that $\lfloor \frac{n}{2} \rfloor$ vertex observers are sometimes necessary and always sufficient to cover a polyhedral terrain [2]. They presented an algorithm to place $\lfloor \frac{3*n}{5} \rfloor$ vertex observers in linear time using the 5-coloring algorithm and selecting the 3 least used colors. This idea is applied in this work using a modified version of the 5-coloring algorithm given by Chiba [3]. In his algorithm, Chiba selects colors randomly among the set of possible colors to paint a vertex. In the work presented here a priority list of colors is used to maximize the number of times 2, out of 5, colors are used, thus minimizing the number of times the 3 least used colors are applied and reducing the number of observers in the first placement.

The algorithm works recursively, removing vertices from the original graph until it has only 5 vertices, and then paints the 5 vertices. After that it starts to insert vertices in the painted graph in reverse order, that is, the last vertex removed is the first vertex inserted. After insertion the neighborhood is checked and the vertex is properly painted. It uses 3 lists of vertices as a guide for removing vertices, one for vertices with degree 4 or less, one for vertices of degree 5, and one for vertices of degree 6.

Euler's formula guarantees the presence of at least one vertex with degree 6 or less in a planar graph [12], so the algorithm always applies (special care is necessary when removing a vertex with degree 5 or 6, see [3] for details). Figure 4 presents the results of the algorithm applied in the planar graph given in Figure 3. There are 40 green vertices, 37 purple, 36 red, 26 brown and 11 yellow.

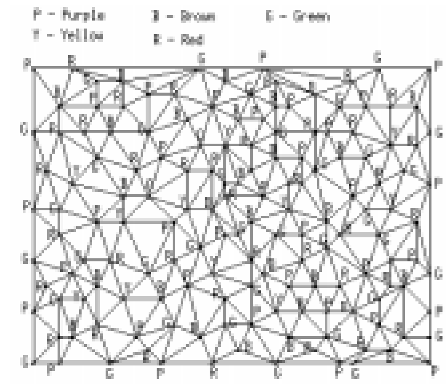

Figure 4: The 5-coloring of the planar graph given in Figure 3.

As each triangle has 3 vertices, selecting the 3 least used colors from the 5 colored graph guarantees that at least one of the vertices of all triangles will be painted with one of the 3 selected colors. Placing an observer at every vertex of one of these colors ensures that all triangles will have at least one observer, and therefore the whole terrain will be covered. Figure 5 gives the first placement of observers for the polyhedral terrain given in Figure 2.

The ideal condition for the first placement would be to select $\lfloor \frac{n}{2} \rfloor$ vertices such that we could cover the whole terrain in the worst case, as proved by Bose et al. This would require a 4-coloring of the graph and selecting the 2 least used colors resulting in a first placement with $\lfloor \frac{n}{2} \rfloor$ observers. Unfortunately there is no algorithmic solution for the 4-coloring problem to date.

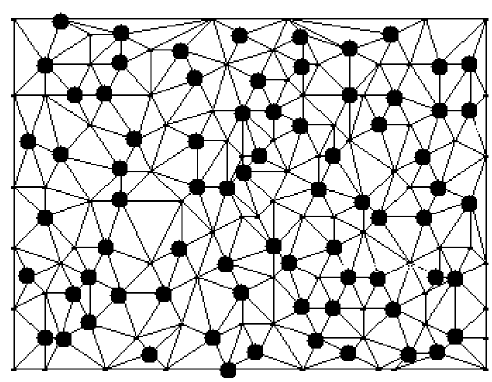

Figure 5: The first placement of observers. There are 74 observers placed and they cover the whole terrain given in Figure 3.

The outputs of the above algorithm are two cross-indexed lists: the first one is a list of observers and for each observer a list of triangles that the observer can see, as well as the total number of triangles visible to the observer. The second list gives for each triangle a list of observers that can see it. Now, considering global visibility in the polyhedral terrain, it is possible to reduce the number of observers based on redundancies.

2.3 *Reducing the number of observers*

To reduce the number of observers a visibility map is computed. The visibility map relaxes the condition of local visibility and gives for each observer a list of all triangles that are visible to that observer in the terrain. The two lists given as input are augmented during this process such that at the end they reflect local and global visibility information.

After the visibility map is computed for all observers, the triangle's list is sorted, such that the triangle with the smallest number of observers is first, and the one with the largest number of observers is last. Ties are resolved randomly.

Observers are placed in the terrain using the following loop: the triangle that is viewed by the fewest observers is selected, and among those observers, the one who can see the highest number of triangles is placed in the terrain. All triangles that the observer can see are marked. The loop is repeated until all triangles are marked and the next unmarked triangle in the sorted list is selected for the next loop. In the second and all subsequent loops the number of unmarked triangles covered by an observer placed has to be maximized (to do this the list of observers has to be updated after each placement). Note that the number of triangles that an observer can see at the beginning is also the number of unmarked triangles he can see, so in every loop the number of unmarked triangles that each observer can see has to be updated. This is done as follows: for each triangle marked, go through its list of observers and decrease the number of triangles that the observer can see by 1. This gives for each observer the number of unmarked triangles that he can see, making the greedy selection more efficient for the next loop. The pseudocode of this step is presented in Figure 6:

Input: List of observers where each observer has a list of triangles it can see and the list of triangles where each triangle has the list of observers that can see it.

Output: An optimized number of observers that can see the whole terrain.

1 For each observer g_i in the list do
2   For each observer $g_j; i \neq j$ in the list do
3   If g_i can see g_j then
4   For each triangle t_k in $g_j's$ list do
5   if g_i can see t_k then
6   add t_k to $g_i's$ list
7   add g_i to $t_k's$ list
8 Sort the list of triangles
9 While not all triangles marked do
10   Select one unmarked triangle t_k
11   Select the observer g_i in $t_k's$ list that can see the largest number of unmarked triangles.
12   Mark all triangles that g_i can see and update the list of observers.
13   Add g_i in the output list.

Figure 6: The algorithm for reducing the number of observers.

The greedy technique used in this step (line 11) is a well known solution to the *Set Coverage* problem and is described in [6]. The test executed in line 3 eliminates some visibility computation and reduces the overall run time. If observer g_i can see observer g_j then we have to test all triangles surrounding g_j to check whether g_i can see each of the triangles or not, but if g_i can not see g_j then it has at most partial view of the surrounding triangles and the visibility computation is not necessary. The final set of observers placed in the terrain is presented in Figure 7.

2.4 *Complexity Analysis*

The problem of computing the minimum number of observers to cover the whole terrain was proved to be NP-hard by reduction from Satisfiability (SAT) to this problem [5]. Let F be a CNF formula with clauses $C_1, \cdots, C_m$, and variables $x_1, \cdots, x_n$ then reduce the satisfiability for F to the problem of determining whether a certain polyhedral terrain with $O(nm)$ faces can be completely viewed by nm points on it. The reduction is done by constructing a polyhedral terrain with the following features: n rows and $n-1$ walls, one row for each variable in F, and m columns, one per each clause. In each row there are $2m$ pits arranged in a circular fashion. The upper rim of the pits are quadrilaterals and the rims of each pair of adjacent pits in the same row have a common vertex called a *peak*. Each pit is deep enough so an observer can only see the whole pit from the boundary or from its interior.

Assuming row r corresponds to variable x_r, the choice of selecting even peaks for the viewing points in r will correspond to setting $x_r = true$, otherwise $x_r = false$. Figure 8 shows the basic idea for the formula $F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$. Using geometric properties of the terrain it is possible to show that the whole terrain can be seen by nm observers if and only if the formula F is satisfiable. More details can be found in [5].

The solution presented here runs in polynomial time. A complexity analysis for each step in the algorithm is presented below with a pointer to more detailed references.

- Step 1:
  1. Delaunay triangulation: $O(nlog(n))$ see Preparata and Shamos [15].
  2. The hierarchical representation: $O(n)$ see de Berg [8].
- Step 2: 5-coloring: $O(n)$ see Chiba [3]. Note

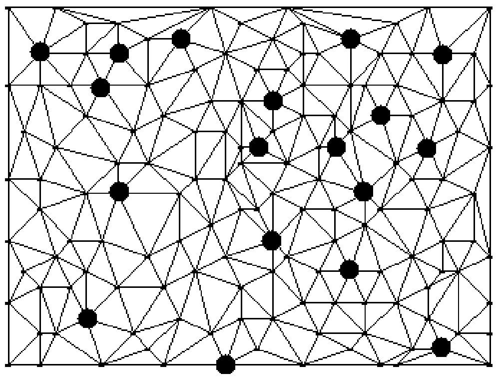

Figure 7: Final placement of observers given by step 3 - 18 observers are kept on the terrain. Note that this final step was approximated by hand (the complete algorithm is currently being implemented).

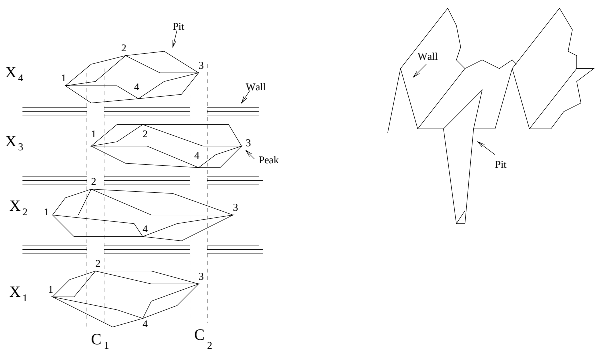

Figure 8: The idea on how to reduce SAT to our visibility problem

that the changes to Chiba's algorithm described here do not affect its time complexity.

- Step 3:
  1. Visibility Map - lines 1 to 7: $O(n^3)$ see de Berg [7]
  2. Sort - $O(nlog(n))$
  3. Selection of observers: the algorithm presented in Figure 11 is $O(n^3)$.

The overall complexity of the algorithm as presented here is $O(n^3)$ because of the visibility test (3.1) and the selection of observers (3.3). Cormen et al [6] suggests that it is possible to implement the selection of observers in linear time. Doing so may improve the run-time of the algorithm, but will not improve its overall time complexity bound. To improve the overall complexity it will be necessary to reduce the complexity bound during the construction of the visibility map.

2.5 The quality of the solution

The approach presented here starts with a DEM and places observers in the terrain in $O(nlog(n))$ time. As the observers are placed based only on local visibility information, the number of observers can be reduced using global visibility information.

The solution quality given by the greedy approach has a rate of $O(log(n))$ from the optimal solution. The idea of the proof is the following: a cost c is attributed to each observer selected and it is amortized among all unpainted triangles that the observer adds to the solution. The cost of the overall solution C is computed and compared with the cost of the optimal solution C^*, which gives a rate of $O(log(n))$, see [6] for details in the greedy technique applied to the set covering problem. This bound is acceptable for a small number of observers and reasonable for a large number of observers.

The visibility map in the original terrain obtained by the final placement is presented in Figure 9. Note that by using only 18 observers, 99% of the original DEM map is covered The 1% of omission is the result of details lost when simplifying the DEM.

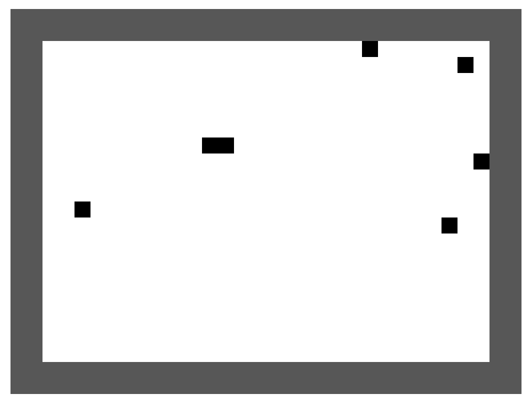

Figure 9: Visibility map for the placement of observers given by step 3. The black spots represent the invisible area in the original terrain. More than 99% of the original terrain is covered; the 1% omission is the result of simplifying the terrain map

3 Other questions that we can answer

The solution of the *Terrain Coverage* presented here can also be used to solve other similar problems. A simple change in the approach is to limit the number of observers, that is, *Given n observers and a terrain T, is it possible to cover T using only n observers? If so, where should we place them?*

This problem is also NP-complete. It is NP because given a placement of n observers it is possible to check in polynomial time if they cover the whole terrain or not (just compute the visibility map, which can be done in $O(n^3)$). It is complete because it is possible to make a reduction from SAT to this problem similar to the one presented earlier [5].

A useful and similar optimization problem is the following: *Given n observers and a terrain, where should we place them in order to maximize the overall coverage of the terrain?*.

Because the highest coverage is desired it is necessary to maximize coverage of the uncovered area for every selection of an observer. So it is necessary to sort the list of observer at each step of the loop and select the one who can see the largest number of unmarked triangles at each step. Note that because the sorting can be performed in $O(nlog(n))$ the overall complexity of the solution is still $O(n^3)$.

Another variation for the problem is to constrain the placement of observers. For example, if part of the terrain is occupied by the enemy or it is a lake, observers should not be placed there. If we have the boundaries of the area where we can not place a observer, we can just select the observer that sees the largest number of triangles and is not inside the forbidden region.

A third variation of this problem is to give a set of points where it is desired to place observers. In this case it is possible to place the observers and show the area not covered in the terrain and also give the number of extra observers needed to cover the remainder of the terrain. The first thing we do is to add these points to the set V_{fixed} used in the first step to compute the hierarchical representation of our terrain. For the second step we have to give the set of desired points (DP) separated from the planar graph and while painting the vertices we test each vertex v. If v belongs to DP then we paint it using the inverse of our priority list of colors. Finally, during step 3 we pass DP as a parameter and when selecting observers we use the DP list as a priority list for picking observers.

4 Conclusion

We present an idea about how to solve the problem of placing a reduced number of observers in a terrain such that each part of the terrain can be seen by at least one observer. Although the problem is NP-hard, combining techniques such as the Delaunay triangulation, graph coloring, and the greedy solution for the set covering problem, allow us to solve the placement of observers covering the whole terrain in polynomial time. The overall quality of this placement is within $O(log(n))$ from the optimal solution.

The complexity analysis shows that the time required for the whole system is bounded by $O(n^3)$, and we are investigating the possibility of reducing this bound.

We are currently implementing the algorithm presented here for practical use in the military context and to verify its run time performance in real DEMs. In parallel we are looking for better implementations of the visibility map construction in order to reduce the time complexity bound.

5 Acknowledgements

We would like to thank Prof. Ileana Streinu of Smith College and Prof. Robert Moll from UMass, for many valuable discussions.

References

[1] Aurenhammer, F. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys 23*, 3 (1991), 345–405.

[2] Bose, P., Shermer, T., Toussaint, G., and Zhu, B. Guarding polyhedral terrains. Tech. Rep. CMPTTR 93-1, Simon Fraser University, 1993.

[3] Chiba, N., Nishizeki, T., and Saito, N. A linear 5-coloring algorithm of planar graphs. *Journal of Algorithms 2* (1981), 317–327.

[4] Chvátal, V. A combinatorial theorem in plane geometry. *Journal of Comb. Theory Ser B 18* (1975), 39–41.

[5] Cole, R., and Sharir, M. Visibility problems for polyhedral terrains. *Journal of Symbolic Computation 7* (1989), 11–30.

[6] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. *Introduction to algorithms.* MIT Press, 1990.

[7] de Berg, M. *Ray shooting, depth orders and hidden surface removal.* Springer-Verlag, 1993.

[8] de Berg, M., and Dobrindt, K. T. G. On levels of detail in terrains. Tech. Rep. UU-CS-1995-12, Utrecht University, April 1995.

[9] Fisk, S. A short proof of chvátal's watchman theorem. *Journal of Comb. Theory Ser B 24* (1978), 374.

[10] Franklin, W. R., and Ray, C. K. Higher isn't necessarily better: visibility algorithms and experiments. Tech. Rep. -, Rensselaer Polytechnic Institute, May 1994.

[11] Kooshesh, and Moret. Three-coloring the vertices of a triangulated simple polygon. *Pattern Recognition 25* (1992), 443–444.

[12] Nishizeki, T., and Chiba, N. *Planar graphs: theory and algorithms.* North-Holland, 1988.

[13] O'Rourke, J. *Art Gallery Theorems and Algorithms.* Oxford University Press, 1987.

[14] O'Rourke, J. *Computational Geometry in C.* Cambridge University Press, 1994.

[15] Preparata, F. P., and Shamos, M. I. *Computational Geometry: An Introduction.* Springer-Verlag, 1985.

[16] Rippa, S. Minimal roughness property of the dealunay triangulation. *Computer Aided Geometric Design 7(7)* (1990), 489–497.

[17] Riseman, E., Schultz, H., Lim, W. J., Draper, B., and Hanson, A. Daedalus battlefield visualization system. *Proceedings of ARPA Image Understanding Workshop* (1996), –.

[18] Shermer, T. Recent results in art galleries. Tech. Rep. -, Simon Fraser University, August 1991.