A system to place observers on a polyhedral terrain in polynomial time ⋆

Maurício Marengoni [a], Bruce A. Draper [b], Allen Hanson [a] and Ramesh Sitaraman [a]

[a] *Department of Computer Science, University of Massachusetts, Amherst, MA 01003 - marengon,hanson,ramesh@cs.umass.edu*

[b] *Department of Computer Science, Colorado State University, Ft. Collins, CO 80523 - draper@cs.colostate.edu*

Abstract

The *Art Gallery Problem* deals with 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 time 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. The solution uses techniques from (i) computational geometry to first build a terrain hierarchy keeping the overall terrain's shape and to compute the visibility map for each observer, (ii) graph coloring to make a first placement of observers on the terrain, and (iii) set coverage to reduce the number of observers based on their visibility map. 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 the following 3-D visibility problem is considered: *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

⋆ 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, in part by NSF under grant CDA-8922572, grant CCR-94-10077 and Career Award CCR-9703017, and in part by the DVRIP - US Army Research Office under grant number DAAH 04-95-1-0068.

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 terrain will be visible to at least one of the observers.

This visibility problem and its variations (Section 3) are real world problems 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 purposes in banks, supermarkets or department stores is another. In the military context, commanders need to place scouts to cover a certain region, or to determine where to hide their resources.

This work was developed focusing on the military context. One of the goals of the Daedalus project [21] is to provide battlefield commanders with powerful new tools for planning and monitoring operations; the visibility problem solved here facilitates both.

The 2-D "coverage" problem was posed by Victor Klee in 1973 and is better known as *The Art Gallery Problem* [15]. 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 by using a triangulation of the polygon and showing that as a triangulated polygon is 3-colorable, selecting the least used color will generate the bound [9]. The placement of observers can be done in linear time [12]. 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 [15] and [22].

There are some similarities between the 2-D and the 3-D version of this visibility problem, but little is known about covering a polyhedral terrain in 3-D. In this paper an algorithmic solution to the 3-D version of the Art Gallery Problem is presented and a time complexity analysis is provided. The method presented here goes from an elevation map to an optimized placement of observers in polynomial time. The solution found here is not optimal, but the total number of observers used is within a known bound of the optimal solution.

1.1 General assumptions

There are two common approaches for solving this problem, both have an $O(n^3)$ time complexity (n is the number of vertices in the triangulated terrain). The first approach considers all points in the DEM and computes the intervisibility of every pair of points. Some applications of this technique are presented in Franklin and Ray [11], Ravela [19] and Wang [23]. Because all points are considered, the run time in this case makes it infeasible for most real applications. This paper adopts an approach that 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 (or Surface)* in computational geometry. Because this approach typically considers far fewer points than in the DEM, a visibility map can be computed much faster in practice. De Floriani [10] presents a small survey of algorithms for computing visibility using TINs. In this paper the following questions are answered: (1) how many observers are needed to cover a *polyhedral terrain*, such that every point on the polyhedral terrain will be visible by at least one observer, and (2) where should the observers be placed.

In a polyhedral terrain observers can be placed on an edge or on a vertex. Here only *vertex observers* are considered. By the 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 (that is, the height of the observer above the terrain is zero), which is the most conservative approach to the visibility problem.

Section 2 describes the algorithm for solving the 3-D visibility problem. Section 3 discusses other related problems that can be solved using the approach presented in here. Section 4 presents the conclusions obtained from this work.

2 Algorithm description

The algorithm is divided into three steps. In the first step a DEM is transformed into a multiresolution terrain hierarchy. The final terrain model used in the algorithm is either the coarsest model in the hierarchy or a combination of different resolutions from the hierarchy. Combining data from different resolutions results in a map with a variable "level of detail", which can improve the quality of the terrain model in designated areas. In the second step only local visibility is considered and the five-coloring algorithm is used to place a set of observers in the terrain. The third step removes the constraint of local visibility and computes global visibility maps for a subset of observers selected from step 2. It uses global visibility maps and the greedy approach to the set covering problem to reduce the number of observers in the terrain.

Notice that the five-coloring step (step 2) presented here does not give any gain in terms of time complexity, but because it reduces the number of points to be considered in step 3 the gain will be seen in terms of run time for the overall system.

In an earlier paper [13] this algorithm was proposed as a theoretical solution to the problem and a hand generated example was used to illustrate the approach. The algorithm is now fully implemented and will be demonstrated on the terrain shown in Figure 1.

2.1 The terrain model

The input data is a DEM where each entry *(X, Y)* in the image represents the height at the coordinate *X, Y*. The dense elevation map can be approximated 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 [16], thereby reducing the roughness of the approximating surface [20]. A Delaunay triangulation of a terrain used as an example in this paper is shown in Figure 1.

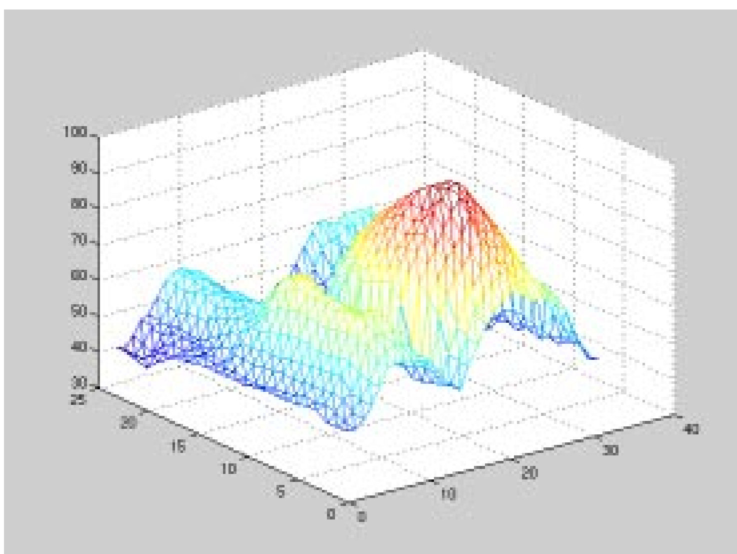

Fig. 1. A Delaunay triangulation of a 3-D terrain represented as a DEM. The terrain here is a grid that has 32 by 24 points.

To precisely represent a terrain, a large number of triangles is needed; the example presented in Figure 1 has 1426 triangles and 768 vertices. The terrain in this case has the finest granularity possible but for many applications this level of detail is not required. By selectively removing points (see below) a hierarchy of models for the terrain can be built and different levels in the hierarchy can be combined to enhance detail in certain regions of the model.

The algorithm proposed by de Berg [8] to build a terrain hierarchy from a DEM was used in the system described here. The system starts with a DEM and a set of points V_{fixed} that are never removed from the terrain and are intended to preserve the terrain structure. In this implementation the set V_{fixed} is selected as follows: the user chooses a percentage P of points to keep from the isometric lines obtained for the original terrain; the points in the isometric lines are sampled randomly to generate V_{fixed}. For the example presented in Figure 1, 74 points have been selected from the original terrain as fixed points (see Figure 2). Because these points are sampled from the isometric lines and never removed, the terrain's shape is preserved through the levels in the hierarchy.

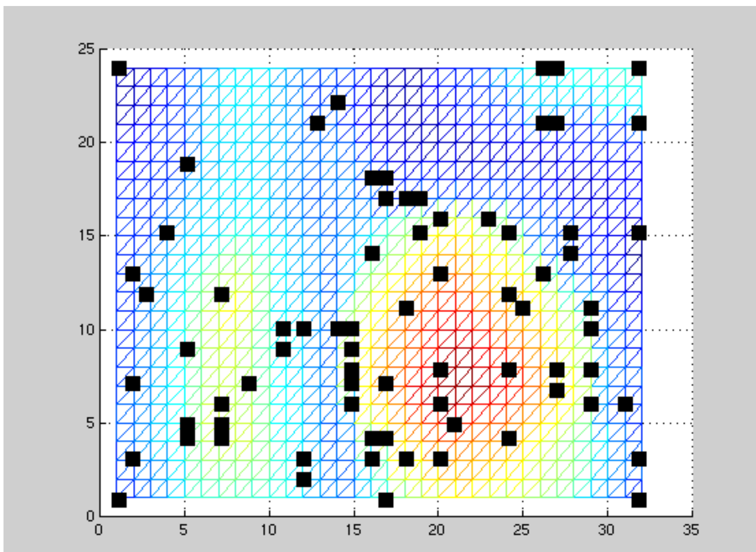

Fig. 2. The 2-D projection of the Delaunay triangulation for the original terrain with all fixed points (see text) marked by a black square. There are 74 fixed points in this case.

The first level in the hierarchy is the Delaunay triangulation of the DEM as shown in Figure 2. All the other levels in the hierarchy are obtained by removing points as follows: if a point v is not in V_{fixed} and is not marked remove point v and all edges incident to v, mark v's neighbors and fix the Delaunay triangulation locally, that is, make a Delaunay triangulation considering only the points that are neighbors of v. When all points in the terrain are either in V_{fixed} or marked a new level in the hierarchy is obtained. Unmark all marked points and repeat the process until the coarsest model of the terrain is obtained. In this implementation the coarsest model is obtained when the number of points in the terrain is at most twice the number of points in the V_{fixed} set.

In some applications, parts of the terrain may need to be represented at higher resolution than others. In this case, it is possible to combine different resolutions from the terrain hierarchy and get a more accurate model. The hierarchy obtained for the example contains seven resolutions for the same terrain. Figure 3 shows the coarsest model (level 7) in the hierarchy with all fixed points marked. The rectangle shown in the middle of the terrain marks a region selected by the user where the level of detail will be enhanced.

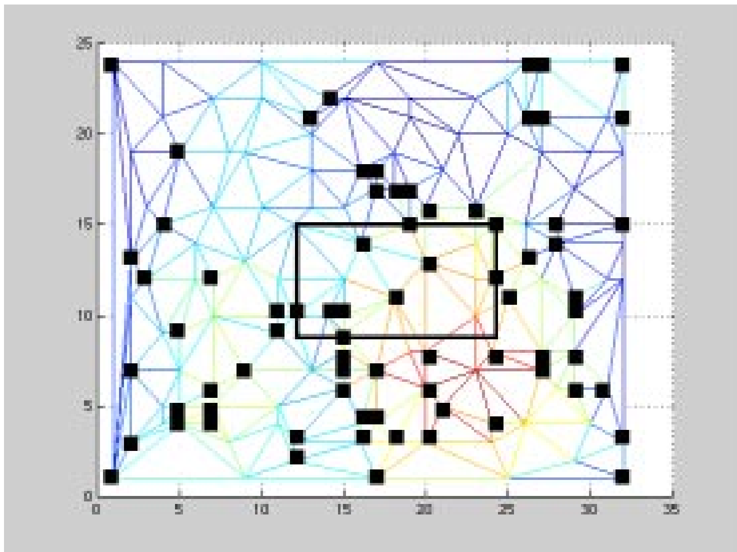

Fig. 3. A Delaunay triangulation in 2-D for the coarsest model; all the fixed points are marked with a black square. This terrain model has 137 points and the rectangle marks the area where the level of detail will be increased by using points from a finer resolution terrain in the hierarchy.

The user also selects the resolution level to be used when mixing different resolutions. Once the region is marked and the resolution level is selected the system gets all points inside the region marked in the finer resolution level and adds them to the coarsest resolution model. Notice that the Delaunay triangulation has to be fixed again, in this case all points from the coarsest resolution and the points just added to the final model are used to compute a new Delaunay triangulation. The final terrain model is shown in Figure 4, in which all points marked were added from the terrain model in level 5.

Figure 5 shows the polyhedral terrain in 3-D. Notice that the overall shape of the terrain was preserved. The planar graph of this terrain is now used to place the first set of observers considering only local visibility information.

2.2 The first placement

The first placement of observers is done based only on local visibility. Observers are placed on vertices of triangles. It is assumed that an observer placed on a vertex can see all triangles that share the vertex, and the observer can see the whole triangle adjacent to that vertex. This is called local visibility;

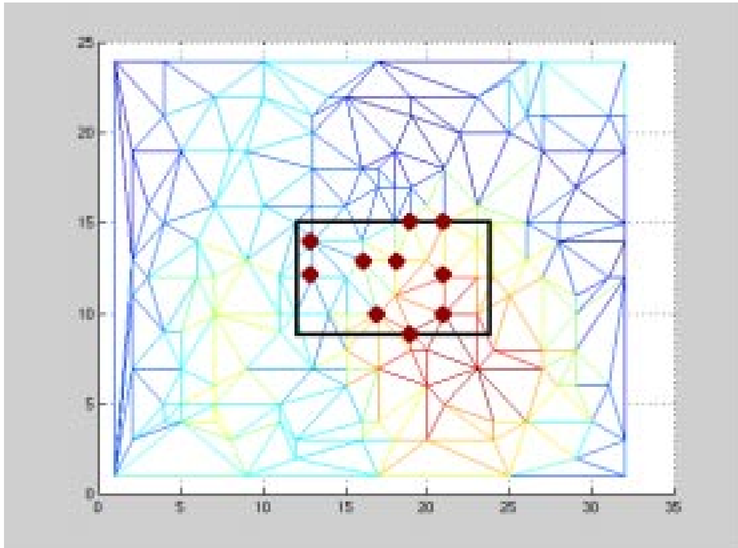

Fig. 4. The enhanced terrain model in 2 D. The 10 points marked with red dots were added in the selected area from resolution level 5. This model has 274 triangles and 147 vertices.

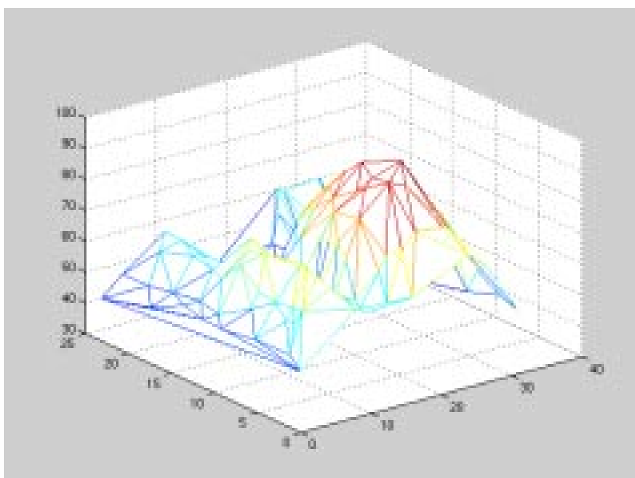

Fig. 5. The simplified terrain model in 3 D. The general shape of the original terrain presented in Figure 1 was preserved.

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]. Thus 4-coloring the correspondent planar graph of the polyhedral terrain, selecting the 2 least used colors and placing an observer on each vertex with one of these 2 colors would result in a first placement with $\lfloor \frac{n}{2} \rfloor$ observers. Unfortunately there is no algorithmic solution for the 4-coloring problem to date. They also present an algorithm to place $\lfloor \frac{3*n}{5} \rfloor$ vertex observers in linear time using the 5-coloring algorithm developed by Chiba [3] and selecting the 3 least used colors. This idea is applied in this work using a modified version of the 5-coloring algorithm.

The algorithm works recursively using 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. It inspects the lists in order of degree and removes the first vertex available (arbitrary choice) in one of the lists. It updates the vertex information in all lists and repeats the process until it has only 5 vertices, these are painted accordingly (this depends on the version of the algorithm used, Chiba paints each vertex with a different color and we used a priority queue - see detailed explanation below). 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 using the standard graph coloring constraint (no adjacent vertices can have the same color).

Euler's formula guarantees the presence of at least one vertex with degree 6 or less in a planar graph [14], so the algorithm always applies (special care is necessary when removing a vertex with degree 5 or 6, see [3]). Figure 6 presents the results of the algorithm applied to the planar graph given in Figure 4. There are 42 blue vertices, 40 yellow, 35 green, 22 red and 8 purple. Placing an observer at every vertex of one of the 3 least used colors ensures that all triangles will have at least one observer, and therefore the whole terrain will be covered. Figure 7 shows the first placement of observers for the polyhedral terrain given in Figure 4.

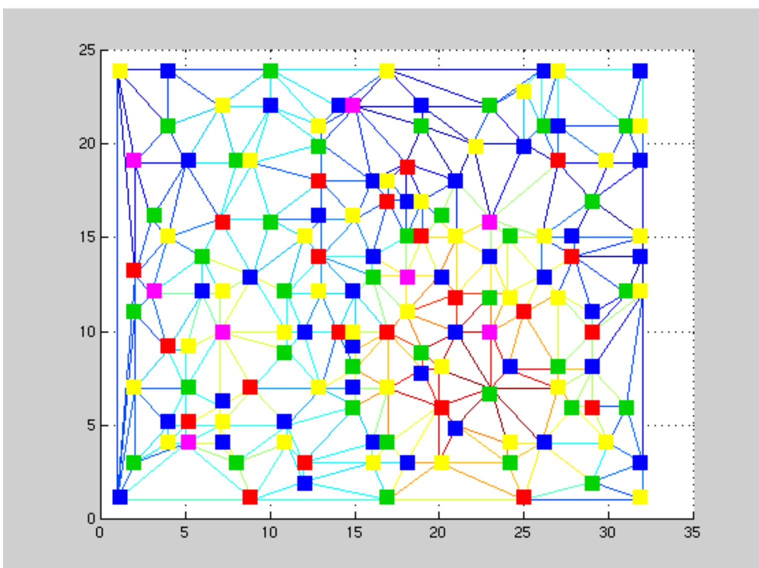

Fig. 6. The 5-coloring of the planar graph given in Figure 4.

In his five-coloring algorithm, Chiba selects colors randomly among the set of possible colors available to paint a vertex [3]. If this approach was used here the first placement step would have selected 83 points (above the $\lfloor \frac{n}{2} \rfloor$ upper bound but below the $\lfloor \frac{3*n}{5} \rfloor$ bound presented in Bose's implementation). Instead, 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 priority list approach resulted in the selection of 65 points as shown in Figure 7 which is even below the tight upper bound of $\lfloor \frac{n}{2} \rfloor$.

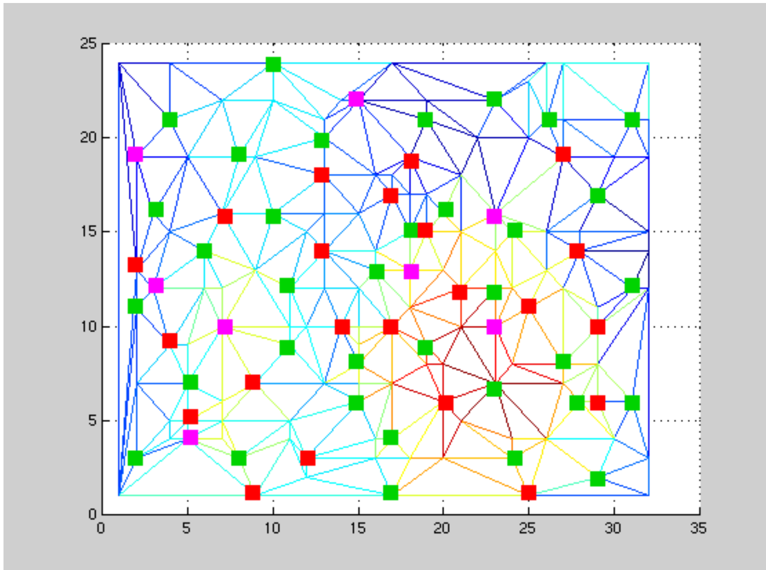

Fig. 7. The first placement of observers. There are 65 observers placed and they cover the whole terrain model given in Figure 4. Notice that this placement does not guarantee that the original terrain will be totally covered by this set of observers.

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 global visibility map for each observer is computed. It uses the observer list and the triangle list computed in the last step as input and proceeds as follows: for each observer in the observer list, sort all triangles in the terrain using a radial sort (with appropriate data structures this can be done in linear time [10]) and compute a horizon line for the observer considering only local visibility. The horizon line is defined to be the boundary of a star shaped polygon in 3-D for which all points inside have been tested and marked for visibility. To make future visibility tests more efficient, the elevation angle associated with the line from the observer through each of the points of the horizon line is saved. For each triangle in the sorted list, determine whether or not the triangle is visible to the observer using the horizon line and the elevation angle. Add the triangle to the observer list and the observer to the triangle list if the triangle is visible and update the horizon line as needed. The upper bound of this algorithm is $O(n^2)$ for one observer and thus $O(n^3)$ total time.

After the visibility map is computed for all observers, the triangle 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 arbitrarily.

Observers are now marked 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 unpainted triangles is marked. All triangles that the observer can see are painted. The loop is repeated and the next unpainted triangle in the sorted list is selected until all triangles are painted.

In the second and all subsequent loops the number of unpainted triangles covered by an observer has to be updated. This is done as follows: for each triangle painted, go through its list of observers and decrease the number of triangles that the observer can see by 1. This gives the number of unpainted triangles that each observer can see, making the greedy selection more efficient for the next loop. Note that the number of triangles that an observer can see at the beginning is also the number of unpainted triangles he can see. At the end of this step all unmarked observers are removed from the terrain. The pseudocode of this step is shown in Figure 8.

The greedy technique used in this step (lines 10 to 13) is a well known solution to the *Set Coverage* problem and is described in [6]. The final set of observers placed in the terrain is shown in Figure 9; the original 65 observers have been reduced to 17. This reduced set of observers is enough to cover the terrain model presented in Figure 4, but it is not guaranteed that they will cover the original terrain model.

2.4 Placement of observers is NP-Hard

In order to justify the approximate algorithm presented here it is first shown (based on the work of Cole and Sharir [5]) that computing the minimum number of observers to cover the whole terrain is NP-hard. This is proved based on the reduction of Satisfiability (SAT [17]) to this problem: let F be a conjunctive normal form (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

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: A reduced number of observers that can see the whole terrain.

1 For each observer g_i in the list do

2 Make the radial sort of the triangles.

3 Computes the horizon line of g_i using local visibility.

4 For each triangle t_k in the sorted list do

5 if t_k is visible

6 add t_k to $g_i's$ list

7 add g_i to $t_k's$ list

8 update the horizon line of g_i

9 Sort the list of triangles

10 While not all triangles painted do

11 Select one unpainted triangle t_k

12 Mark the observer g_i in $t_k's$ list that can see the largest number of unpainted triangles.

13 Paint all triangles that g_i can see and update the list of observers.

14 Remove all unmarked observers form the observer list.

Fig. 8. The algorithm for reducing the number of observers.

variable in F, and m columns, one per 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 10 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].

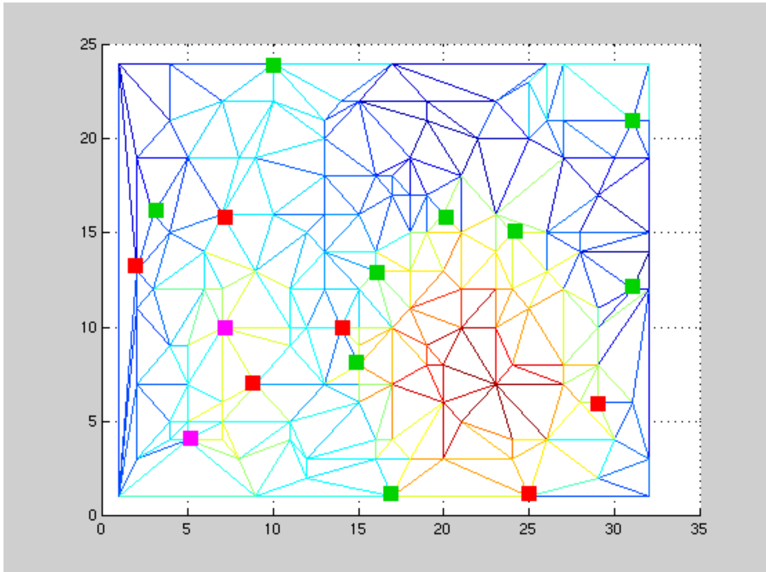

Fig. 9. Final placement of observers given by step 3 - 17 observers are kept on the terrain.

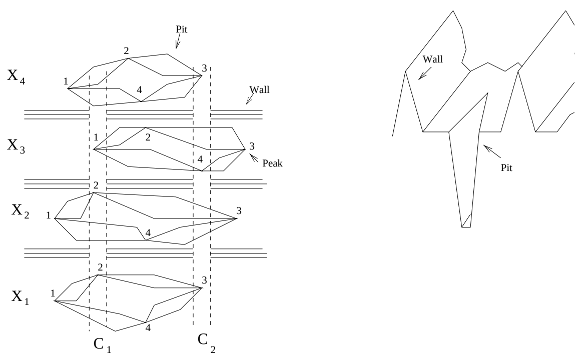

Fig. 10. The idea on how to reduce SAT to our visibility problem

2.5 Complexity Analysis

A complexity analysis for each step in the algorithm is presented below, with pointers to more detailed references.

(1) Step 1: Building the terrain model
   (a) Delaunay triangulation is $O(nlog(n))$ (Preparata and Shamos [18]).
   (b) The hierarchical representation takes $O(n)$ (de Berg [8]).
(2) Step 2: First Placement
   (a) 5-coloring is $O(n)$ (Chiba [3]). Note that the changes to Chiba's algorithm described here do not affect its time complexity.
(3) Step 3: Observer reduction
   (a) Visibility Map - lines 1 to 8 - is $O(n^3)$ (see de Berg [7]).
   (b) Sort the triangle list is $O(nlog(n))$.
   (c) Selection of observers - lines 10 to 14 - is $O(n^3)$.

The overall complexity of the algorithm as presented here is $O(n^3)$ because of the visibility test (Step 3 item a) and the selection of observers (Step 3 item c). 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.

2.6 The quality of the solution

The approach presented here starts with a DEM, builds a terrain hierarchy, and makes the first placement of 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. Notice that these two steps do not determine the complexity of the overall algorithm, that is the third step could be applied directly to the triangulation of the original DEM and the system would still run in $O(n^3)$. The first two steps are used to reduce the number of points to be considered as observers and the number of triangles in the visibility analysis, thereby reducing run time.

The solution produced by the greedy approach to the set coverage problem (step 3) is known to be within a factor of $O(log(n))$ of the optimal solution [6]. 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 of 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.

As it was mentioned before the final set of observers guarantees that the whole simplified model of the terrain will be covered, but nothing can be said about the original terrain. The final set of observers from the example used here was placed in the original terrain (the DEM presented in Figure 1) and the overall visibility was computed. In this case, it turned out that 100% of coverage was obtained using only 17 observers.

3 Variations of the original problem

The solution to the *Terrain Coverage* problem presented here can also be used to solve 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?* Note that this problem is also NP-complete. It is in 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 NP-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 the non-overlapping coverage for each observer. Therefore it is possible to sort the list of observers at each step of the loop and select the one who can see the largest number of unpainted 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 an observer, we can select observers that see the largest number of triangles and are not inside the forbidden region.

A fourth variation of this problem is to give a set of observer positions. In this case it is possible to show the area not covered by these observers and to give the number of extra observers needed to cover the remainder of the terrain. This is accomplished by adding the observer points to the set V_{fixed} used in the first step to compute the hierarchical representation of our terrain. For the second step we 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 use the DP list as a priority list for picking observers.

4 Conclusion

A system to solve the problem of placing a reduced number of observers in a 3-D terrain such that each part of the terrain can be seen by at least one observer was presented in this paper. The system has some nice properties: it can combine detail from different resolutions of the terrain hierarchy and using the five-coloring algorithm with a priority queue it places a first set of observers on the terrain which will reduce the run time for the computation of global visibility maps. This set of observers is reduced using global visibility maps and the greedy approach to the set covering problem. Although the problem is NP-hard, combining techniques above 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.

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] Floriani, L. D., and Magillo, P. Visibility algorithms on triangulated digital terrain models. *International Journal of Geographic Information Systems 8(1)* (1994), 13–41.

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

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

[13] Marengoni, M., Draper, B., Hanson, A., and Sitaraman, R. Placing observers to cover a polyhedral terrain in polynomial time. *Proceedings of the Third Workshop on Applications of Computer Vision* (1996), 77–82.

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

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

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

[17] Papadimitriou, C. *Computational Complexity.* Addison-Wesley, 1994.

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

[19] Ravela, S., Weiss, R., Draper, B., Pinette, B., Hanson, A., and Riseman, E. Stealth navigation: Planning and behaviors. *Proceedings of ARPA Image Understanding Workshop* (1994), 1093–1100.

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

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

[22] Shermer, T. Recent results in art galleries. Tech. rep., Simon Fraser University, August 1991.

[23] Wang, X., Lim, J., Collins, R., and Hanson, A. Automated texture extraction from multiple images to support site model refinement and visualization. *Proceedings of the Fourth International Conference in Central Europe on Computer Graphics and Visualization* (1996), 399–408.