A system to place observers in 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* 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 to first build a terrain hierarchy keeping the overall terrain's shape and to compute the visibility map for each observer, Graph Coloring to make a first placement of observers on the terrain, and 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 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

⋆ 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 and grant CCR-94-10077, 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 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 alternativelythey need to determine where to hide their resources.

This work was developed focusing on the military context. The Daedalus project [21] has as one of its goals to provide battlefield commanders with a powerful new tool 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, using triangulation of the polygon and the fact that 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 [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 the problem, but little is known about covering a polyhedral 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 moves 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 common approaches for solving this problem and both have an $O(n^3)$ time complexity. In the first approach, all points in the DEM are considered and the intervisibility of every pair of points is computed, 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 of the first approach makes it practically impossible in rel time applications. This paper adopts a second approach which 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. Because this model consider fewer points than the DEM the run time of a visibility map can be computed much faster, a small survey of algorithms for computing visibility in a TIN were presented by De Floriani [10]. 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 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. Here only *vertex observers* are considered. 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 describes the algorithm for solving the 3-D visibility problem. Section 3 discusses other related problems that can be solved using the approach presented. 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 in a terrain hierarchy and the final model of the terrain is either the coarsest model in the hierarchy or a combination of different levels of the hierarchy. This combination is done in order to enhace the level of detail in certain 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 relaxes the local visibility condition and optimizes the number of observers using the greedy approach to the set covering problem.

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 a previous work [13] this algorithm was proposed but not fully integrated as a system and some of the steps were performed by hand. Here the whole system was implemented and all the steps can be shown more accuratelly. It is now possible to show some steps that were not performed in the previous work. Figure 1 at left presents the terrain that will be used as an example in order to illustrate each step of the algorithm presented here.

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*. 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 presented in Figure 1 at left.

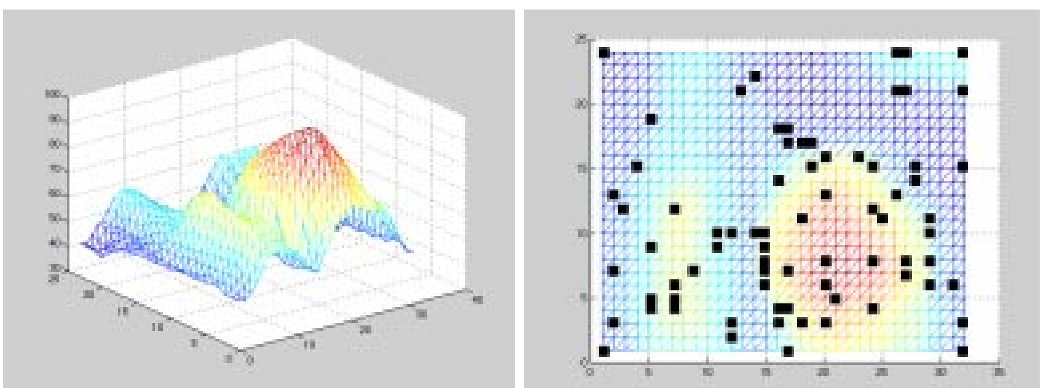

Fig. 1. At left: a Delaunay triangulation in 3-D of a terrain given as a DEM. The terrain here is a grid that has 32 by 24 points. At right: a Delaunay triangulation in 2-D for the original terrain with all fixed points marked. There are 74 fixed points in this case.

To precisely represent a terrain, millions of triangles are needed; the simple example presented in Figure 1 has 1426 triangles and 768 vertices. The terain in this case has the finest granularity possible but for many applications this level of detail is not necessary or not required. By removing points in a certain fashion a hierarchy of models for the same terrain can be built and different levels in the hierarchy can be combined to enhace detail in certain regions of the model. The overall computation for this model will be performed faster than in the original DEM.

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 intent to preserve the terrain structure. In our example 74 points have been selected from the level lines of the terrain, see Figure 1 at right. Because the points are sampled from the level lines and never removed the terrain's shape is preserved through the levels in the hierarchy.

The hierarchy starts with a Delaunay triangulation of the terrain and simplifies the model removing points as follows: if a point is not in V_{fixed} and is not marked remove the point, mark all neighbors of the point just removed and fix the Delaunay triangulation locally. 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.

Once the hierarchy is defined it is possible to select a region in the terrain and combine two different levels to get a more accurate model for the terrain. The hierarchy obtained in the example presents seven models for the same terrain. Figure 2 left presents the coarsest model (level 7) in the hierachy with all fixed points marked. The rectangle shown in the middle of the terrain shows a region selected by the user where the level of detail will be enhaced with the model in level 5. The final terrain model is presented in Figure 2 right, all points marked were added from the terrain model in level 5.

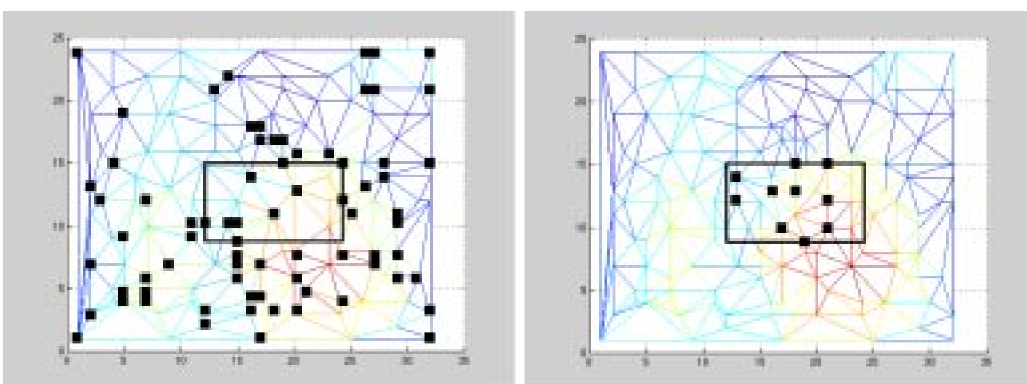

Fig. 2. At left: a Delaunay triangulation in 2-D for the coarsest model; all the fixed points are marked. This terrain model has 137 points and the rectangle marks the area where the level of detail will be increased. At right: the simplified terrain model in 2 D. 10 new points were added in the selected area. This model has 274 triangles and 147 vertices.

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

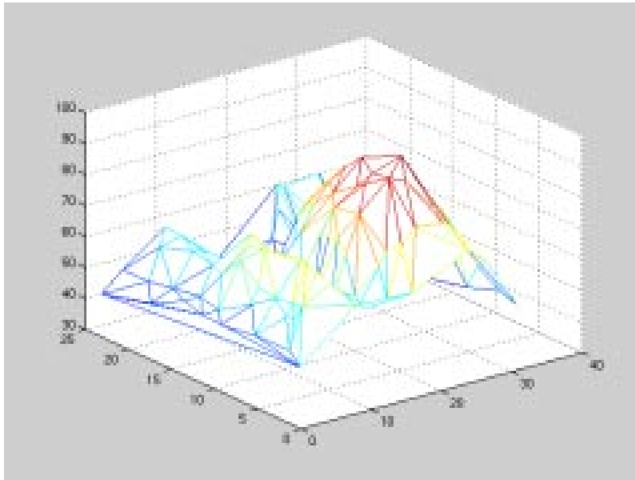

Fig. 3. The simplified terrain model in 3 D. The general shape of the original terrain presented in Figure 1 was preserved.

2.2 The first placement

The first placement of observers is done based only on local visibility. If an observer is placed on a 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 on 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]. This would require a 4-coloring of the graph and selecting the 2 least used colors 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 presented 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, 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 [14], so the algorithm always applies (special care is necessary when removing a vertex with degree 5 or 6, see [3]). Figure 4 left presents the results of the algorithm applied in the planar graph given in Figure 2 at right. There are 42 blue vertices, 40 yellow, 35 green, 22 red and 8 orange.

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 4 right gives the first placement of observers for the polyhedral terrain given in Figure 2 at right.

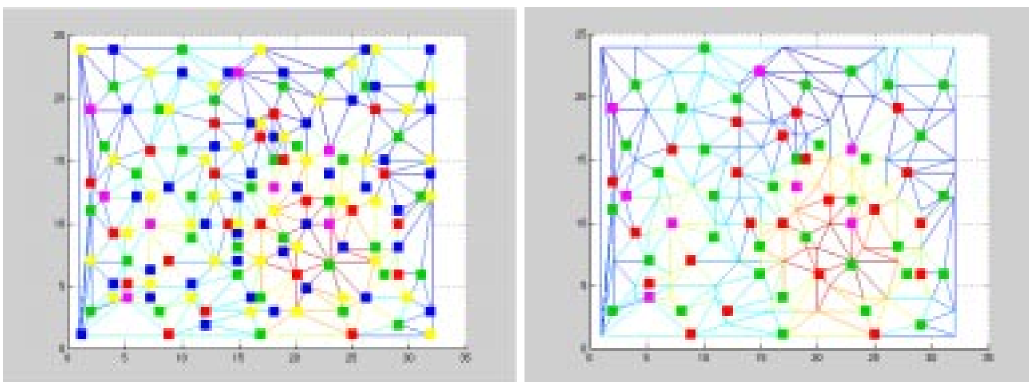

Fig. 4. At left: the 5-coloring of the planar graph given in Figure 2 right. At right: the first placement of observers. There are 65 observers placed and they cover the whole terrain given in Figure 2 right.

In his five coloring algorithm Chiba selects colors randomly among the set of possible colors 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 $\lfloor \frac{3*n}{5} \rfloor$ bound presented in Bose's implementation). Instead of selecting colors randomly a priority list of colors is implemented 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 selected only 65 points as presented in Figure 4 which is even below the tight upper bound of $\lfloor \frac{n}{2} \rfloor$.

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 for each observer is computed as follows: for each observer in the list compute a radial sort of the triangles (with appropriate data structure this can be done in linear time [10], compute the horizon line of the observer considering only local visibility, then go through the sorted triangles and test them using the horizon line to see if they are visible or not, mark the triangle if it is visible and update the horizon line as required. The upper bound of this algorithm is $O(n^2)$ for one observer thus $O(n^3)$ total time.

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 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 for each observer the number of unpainted triangles that he 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 can be removed from the terrain. The pseudocode of this step is presented in Figure 5:

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 presented in Figure 6; the original 65 observers have been reduced to 17.

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      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's list.
```

Fig. 5. The algorithm for reducing the number of observers.

2.4 *Complexity Analysis*

The problem of computing the minimum number of observers to cover the whole terrain can be proven [5] to be NP-hard using a proof based on reduction from Satisfiability (SAT [17]) to this problem: let F be a conjuntive 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 variable in F, and m columns, one per clause. In each row there are $2m$ pits arranged in

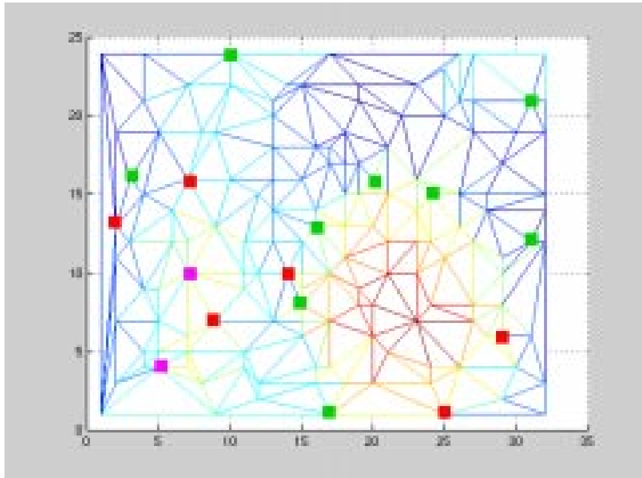

Fig. 6. Final placement of observers given by step 3 - 17 observers are kept on the terrain.

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 7 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].

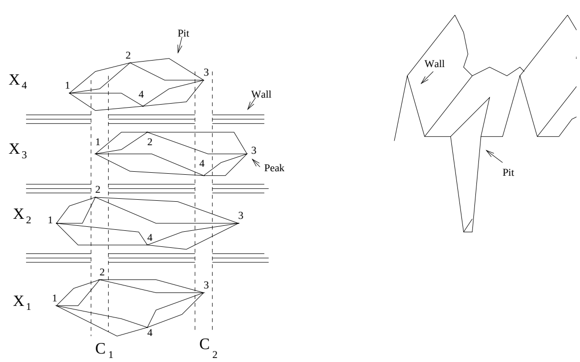

Fig. 7. The idea on how to reduce SAT to our visibility problem

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

– Step 1:
(i) Delaunay triangulation: $O(nlog(n))$ see Preparata and Shamos [18].
(ii) The hierarchical representation: $O(n)$ see de Berg [8].

- Step 2: 5-coloring: $O(n)$ see Chiba [3]. Note that the changes to Chiba's algorithm described here do not affect its time complexity.
- Step 3:

(i) Visibility Map - lines 1 to 7: $O(n^3)$ see de Berg [7]
(ii) Sort - $O(nlog(n))$
(iii) Selection of observers: the algorithm presented in Figure 7 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 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 help in 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 so the overall run time is faster.

The solution quality of step 3 which is the greedy approach to the set coverage problem has a rate of $O(log(n))$ from 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.

In this example the visibility map for the original terrain obtained by the final placement has 100% of coverage using only 17 observers.

3 Other questions that we can answer

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?*

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 observers 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 fourth 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 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 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 levels of the terrain hierarchy and using a priority queue it can reduce the number of observers in the first placement which will reduce the overall run time. 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.

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 Visio* (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 on Computer Graphics and Visualization Univ. of West Bohemia, Plzen, Czech Republic* (1996).