SITING OBSERVERS ON TERRAIN

Wm Randolph Franklin

Rensselaer Polytechnic Institute, Troy, New York 12180–3590

`wrf@ecse.rpi.edu, http://www.ecse.rpi.edu/Homepages/wrf`

March 18, 2002

ABSTRACT

This paper presents an experimental study of a new algorithm that synthesizes separate programs, for fast viewshed, and for fast approximate visibility index determination, into a working testbed for siting multiple observers jointly to cover terrain from a full level-1 DEM, and to do it so quickly that multiple experiments are easily possible. Both the observer and target may be at a given fixed distance above the terrain. The process operates as follows. (1) An approximate visibility index is calculated for each point in the cell under consideration. (2) A set of tentative observers is selected from the highly visible points. (3) The actual observers are selected from them, so as to cover as much of the cell of possibe, using a greedy algorithm. Various experiments with varying parameters were performed on the Lake Champlain West cell, with observations such as the following. (1) Forcing tentative observers to be well spaced was more important than using the most visible tentative observers. (2) Most of the new observers added (because they covered the most unseen points) were already visible to an existing observer. (3) Randomly deleting many tentative observers before final selection didn't reduce the final area covered.

KEYWORDS: terrain visibility, viewshed, line of sight, multiple observers

Contents

1   Introduction

Consider a terrain elevation database, and an observer, $\mathcal{O}$. Define the *viewshed* as the terrain visible from $\mathcal{O}$ within some distance R of $\mathcal{O}$. The observer might be situated at a certain height above ground level, and might also be looking for targets at a certain height above the local ground. Also, define the *visibility index* of $\mathcal{O}$ as the fraction of the points within distance R of $\mathcal{O}$ that are visible from $\mathcal{O}$. This paper combines an earlier fast viewshed algorithm with an earlier approximate visibility index algorithm, to site multiple observer so as to jointly cover as much terrain as possible.

This paper extends the earlier visibility work in Franklin (2000) and Franklin and Ray (1994), which also survey the terrain visibility literature. Notable pioneer work on visibility includes De Floriani et al. (1993); De Floriani and Magillo (1994); Fisher (1993); Lee (1992); Shannon and Ignizio (1971). Shapira (1990) studied visibility, and provided the Lake Champlain W data used in this paper. Ray (1994) presented new algorithms and implementations of the visibility index, and devised the efficient viewshed algorithm that we use.

This multiple observers case is particularly interesting and complex, and has many applications. A cell phone provider wishes to install multiple towers so that at least one tower is visible (in a radio sense) from every place a customer's cellphone might be. Here, the identities of the observers of highest visibility index are of more interest than their exact visibility indices, or than the visibility indices of all observers. One novel future application of siting radio transmitters will occur when the moon is settled. The moon has no ionosphere to reflect signals, and no stable satellite orbits. The choices for long-range communication would seem to include either a lot of fiber optic cable or many relay towers. That solution is the multiple observer visibility problem.

As another example, a military planner needs to put observers so that there is nowhere to hide that is not visible from at least one. This leads to a corollary application, where the other side's planner may want to analyze the first side's observers to find places to hide. In this case, the problem is to optimize the targets' locations, instead of the observers'.

Again, a planner for a scenic area may consider each place where a tourist might be to be an observer, and then want to locate ugly infrastructure, such as work yards, at relatively hidden sites. We may wish site a forest clearcut to be invisible to observers driving on a highway sited to give a good view. Finally, an architect may be trying to site a new house while following the planning board's instruction that, "You can have a view, but you can't be the view."

Speed of execution on large datasets is of more importance than may be apparent. Many prototype implementations, demonstrated on small datasets, do not scale up well. That may happen either because of the size and complexity of the data structures used, or because of the asymptotic time behavior. For instance, even an execution time proportional to $N \log(N)$, where N is the size of the input, is problematic for $N = 10^6$. In that case, the $\log(N)$ increases the time by a factor of 20. Some preliminary published algoriths may even be exponential if performing a naive search. Therefore, we strive for the best time possible.

In addition, large datasets may contain cases, which did not occur in the small test sets, that require tedious special programming by the designer. In a perfect software development process, all such cases would have been theoretically analyzed *a priori*, and treated. However, in the real world, testing on the largest available datasets increases some confidence in the program's correctness.

Next, a large enough quantitative increase in execution speed leads to a qualitative increase in what we can do. Only if visibility can be computed efficiently, can it be used in a subroutine that is called many times, perhaps as as part of a search, to optimize the number of observers. This becomes more important when a more realistic function is being optimized, such as the total cost. E.g., for radio towers, there may be a tradeoff between a few tall and expensive towers, and many short and cheap ones. Alternatively, certain tower locations may be more expensive because of the

need to build a road. We may even wish to add redundancy so that every possible target is visible from at least two observers. In all these cases, where a massive search of the solution space is required, success depends on each query being as fast as possible.

Finally, altho the size of available data is growing quickly, it is not necessarily true that available computing power is keeping pace. There is a military need to offload computations to small portable devices, such as a Personal Digital Assistant (PDA). A PDA's computation power is limited by its battery, since, approximately, for a given silicon technology, each elemental computation consumes a fixed amount of energy. Batteries are not getting better very quickly; increasing the processor's cycle speed just runs down the battery faster.

There is also a compounding effect between efficient time and efficient space. Smaller data structures fit into cache better, and so page less, which reduces time. The point of all this is that efficient software is at least as important now as ever.

The terrain data structure used here is a 1201 by 1201 matrix of elevations, such as from a USGS level-1 Digital Elevation Model cell. The relative advantages and disadvantages of this data structure versus a triangulation are well known, and still debated; the competition improves both alternatives. This current paper utilizes the simplicity of the elevation matrix, which leads to greater speed and small size, which allows larger data sets to be processed.

For distances much smaller than the earth's radius, the terrain elevation array can be corrected for the earth's curvature, as follows. For each target at a distance D from the observer, subtract $\frac{D^2}{2E}$ from its elevation, where E is the earth's radius. (The relative error of this approximation is $\left(\frac{D}{2E}\right)^2$.) It is sufficient to process any cell once, with an observer in the center. The correction need not changed for different observers in the cell, unless a neighboring cell is being adjoined. Therefore, since it can be easily corrected for in a preprocessing step, our visibility determination programs ignores the earth's curvature.

The radius of interest, R, out to which we calculate visibility, has no relation to the distance to the horizon, but is determined by the technology used by the observer. E.g., if the observer is a radio communications transmitter, doubling R causes the required transmitter power to quadruple. If the observer is a searchlight, then its required power is proportional to R^4.

In order to simplify the problem under study enough to make some progress, this work also ignores factors such as vegetation that need to be handled in the real world. The assumption is that it's possible, and a better strategy, to incorporate them only later.

Franklin (2002) is the archival version of this paper, to which citations should be made.

2 Detailed Process

Our system of selecting a set of observers to cover a terrain cell consists of four core programs coordinated with a shell script. Assorted auxiliary programs, to format output for viewing, etc, were also written.

1. VIX. This calculates approximate visibility indices of every point in a cell. It takes a radius of interest, the observer and target height, and T, a sample size. VIX reads an elevation cell. For each point in the cell in turn, VIX considers that point as an observer, picks T random targets uniformly and independently distributed within the radius of the point, and computes what fraction are visible. That is this point's estimated visibility index.

2. FINDMAX. This selects a manageable subset of the most visible tentative observers from VIX's output. This is somewhat subtle since there may be a region containing all points of very high visibility. A lake surrounded by

mountains would be such a case. Since multiple close observers are redundant, we force the tentative observers to be spread out as follows.

    (a) Divide the cell into smaller blocks of points. If necessary, first perturb the given block size so that all the blocks are the same size, ± 1.

    (b) In each block, find the K points of highest approximate visibility index, for some reasonable K. If there were more than K points with equally high visibility index, then select K at random, to prevent a bias towards selecting points all on one side of the block. (This was implemented by using a multiplicative hash function of the point's coordinates as a secondary sort key.) If a block has fewer than K points, then return all its points.

3. VIEWSHED This finds the viewshed of a given observer at a given height out to a given radius, R. The procedure, which is an improvement over Franklin and Ray (1994), goes as follows.

    (a) Define a square of side $2R$ centered on the observer.

    (b) Consider each point around the perimeter of the square to be a target in turn.

    (c) Run a sight line out from the observer to each target calculating which points adjacent to the line, along its length, are visible, while remembering that both the observer and target are probably above ground level.

    (d) If the target is outside the cell, because R is large or the observer is close to the edge, then stop processing the sight line at the edge of the cell.

One obvious "improvement", when the target is outside the cell, would be to move the target in to the edge of the cell before running the sight line. However, this would cause the computed viewshed to depend slightly on R, which looks bad.

The above procedure, due to Ray (1994), is an approximation, but so is representing the data as an elevation grid, and this method probably extracts most of the information inherent in the data. There are combinatorial concepts, such as Davenport-Schintzel sequences, (Agarwal and Sharir, 1999), i.a., which present asymptotic worst-case theoretical methods.

4. SITE This takes a list of viewsheds and find a quasi-minimal set that covers the terrain cell as thoroughly as possible. The method is a simple greedy algorithm. At each step, the new tentative observer whose viewshed will increase the cumulative viewshed by the largest area is included, as follows.

    (a) Let $\mathcal{C}$ be the cumulative viewshed, or set of points visible by at least one selected observer. Initially, $\mathcal{C}$ is empty.

    (b) Calculate the viewshed, $\mathcal{V}_i$, of each tentative observer $\mathcal{O}_i$.

    (c) Repeat the following until it's not possible to increase $area(\mathcal{C})$, either because all the tentative observers have been included, or (more likely) because none of the unused tentative observers would increase $area(\mathcal{C})$.

        i. For each $\mathcal{O}_i$, calculate $area(\mathcal{C} \cup \mathcal{V}_i)$.

        ii. Select the tentative observer that increases the cumulative area the most, and update $\mathcal{C}$. Not all the tentative observers need be tested every time, since a tentative observer cannot add more area this time than it would have added last time, had it been selected. Indeed, suppose that the best new observer found so far in this step would add new area A. However we haven't checked all the tentative new observers yet in this loop, so we continue. For each further tentative observer in this execution of the loop, if it would have added less than A last time, then do not even try it this time.

3   Lake Champlain W Cell Experiments

We tested the above process on the level-1 USGS DEM Lake Champlain West, Figure 1, containing 1201 by 1201 points. (The intensities of all the figures in this paper have been nonlinearly transformed to enhance visibility.) This cell has a variety of terrain, ranging from 1591 meter high Mt Marcy in the Adirondack Mountains of New York State, to a large flat lowland and Lake Champlain. We used a viewshed radius of 100 points. The observers and targets are both 100 m above the terrain surface.

In VIX, the approximate visibility indices of the points were determined with 10 random targets per observer. The resulting visibility map is shown in Figure 2. The brown areas have high visibility; the blue low.
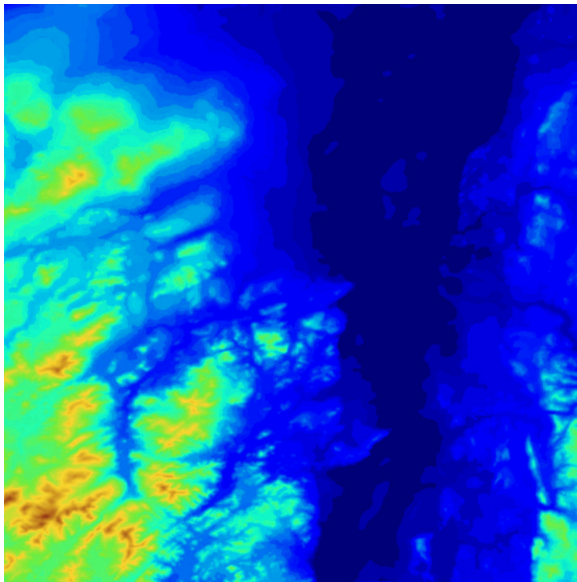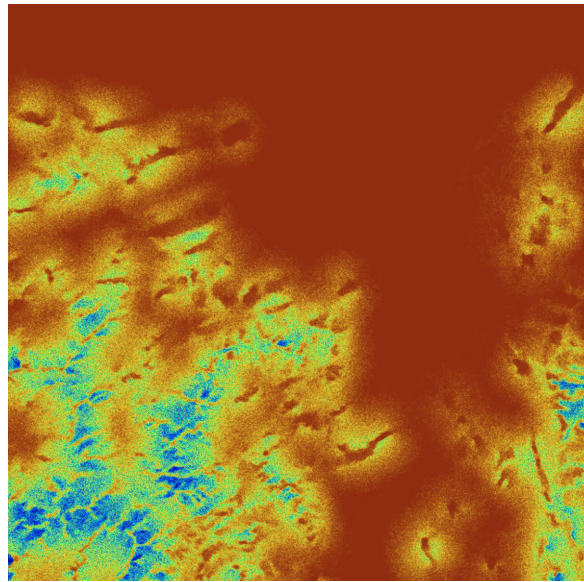

Figure 1: Lake Champlain W Cell

Figure 2: Visibility Indices for Radius=100, Observer Ht=100

In FINDMAX, the cell was divided into blocks of size 101 by 101 points, and 7 tentative observers were selected in each block, for a total of 1008 tentative observers. Figure 3 shows some sample viewsheds returned by FINDMAX.

Figure 3: Sample High Visibility Viewsheds

Several observations, only some obvious, were made from this experiment.

1. Initially the cumulative area grows quickly, but then that slows since most of the so-far unused viewsheds cover

very few so-far unseen points. Figure 4 shows this. One set of points shows the cumulative visible area growing with the number of included observers. The other shows the decreasing amount of new visible area added by each new observer. The latter set is scaled up by a factor of 30 so that both datasets could be meaningfully shown on one plot.

2. The order that the tentative observers were included was not strongly related to their visibility indices. There was an exception at the start, when few of the viewsheds overlapped, because the viewsheds' radius, 100, was so much smaller than the cell size, 1201.

3. Starting with the 50th observer to be added, almost every new observer was already visible to an observer that had been added earlier. This suggests that it would be a bad idea to select new observers only from terrain that was not yet covered.

4. After 180 tentative observers were chosen, the process stopped. None of the remaining 828 tentative observers' viewsheds covered a single new point that was not already visible.

   This tells us that covering the whole cell would require some observers of lower visibility index.

5. When the process stopped with the 180 observers, 99.91% points were visible from some observer. Indeed after only 90 observers, Figure 4 shows that already 98% of the points are visible.
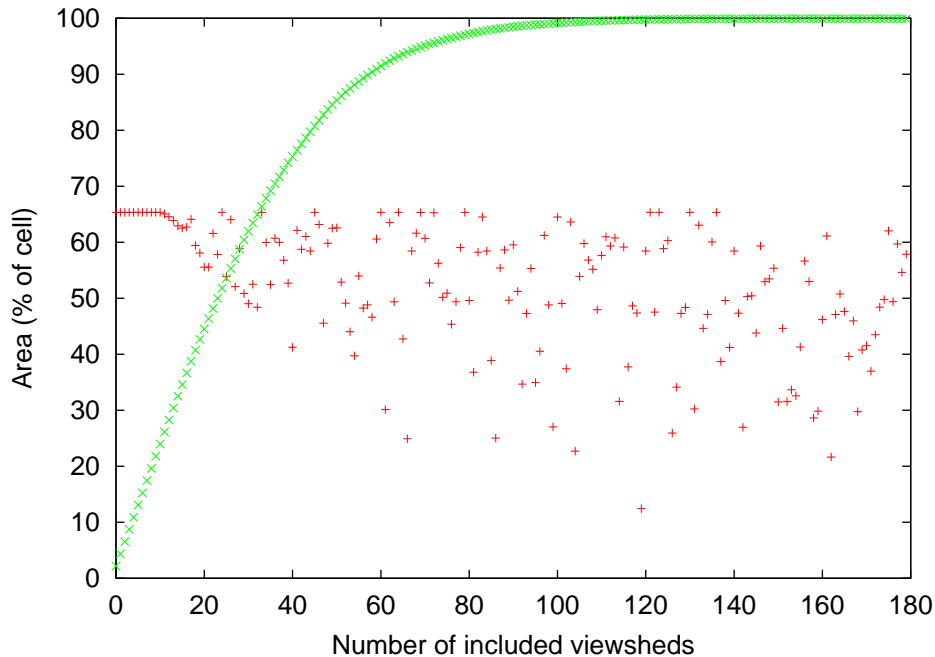

Figure 4: Area of Last Inserted Viewshed $\times 30$, and Cumulative Visible Area, as Observers are Inserted, for Lake Champlain W Cell

We recognize that running these experiments on other data sets may produce varying conclusions, and are testing that.

3.1 Execution Times

The worth of the above system lies, not in its theoretical depth, but in its speed. Great care was taken, first, in algorithm and data structure selection, and then in balancing the system and in micro-optimization. Observations

from preliminary experiments, not described here, were used to refine the system. Remember that quantitative speed translate into a qualitative change in the science, since faster SW allows more and larger experiments to be run, hypotheses to be tested, and observations to be made.

Here are some sample times, on a 600MHz Pentium for the preceding experiment. These are real, elapsed, times. Those are larger than CPU times, because of disk I/O, and the time needed to uncompress the input before feeding it to the program, and then to compress the output. Therefore, elapsed times more accurately reflect a program's performance.

- VIX took 114 seconds to test 10 random targets against each of the 1442401 observers.

- FINDMAX took 7 seconds to pick 1008 tentative observers from the 1442401 points.

- VIEWSHED took 188 seconds to calculate 1008 viewsheds.

- SITE took 149 seconds to select 180 observers from the 1008 tentatives.

The total elapsed time was only 458 seconds to find 180 observers to cover 99.9% of this level-1 DEM cell. If necessary, that time could be reduced with further hand tuning.

Various tradeoffs are possible among the above components. For a fixed cell size, the time for VIX depends linearly on the number of random targets per observer. More targets means that the estimated visibility indices are more accurate. However, SITE often seems to do about as well w/o using exactly the most visible tentative observers. VIEWSHED's time depends linearly on the radius multiplied by the number of tentative observers. SITE's time depends roughly quadratically on the number of tentative observers. In addition, storing all those viewsheds requires memory, 190MB in the above experiment. Further, this memory should be real memory, not virtual, otherwise the paging would slow the program's real elapsed time by a factor of several.

3.2   Not Forcing Tentative Observers to be Spaced

Because subdividing the cell into blocks seemed artificial, we selected extra tentative observers per block, combined them, and returned the top 1000 tentative observers, which now had slightly higher visibility indices on average, but were not as well spaced out as before. The results from SITE became noticeably worse. It stopped after including 181 observers, because no new potential observer saw even one unseen point. At this point, only 92% of the cell was visible from some observer. Therefore, forcing the tentative observers to be spaced out is good.

3.3   Requiring New Observers to be Hidden from Existing Ones

Because selecting new tentative observers only from, as yet, uncovered terrain is such an obvious heuristic, Franklin (2000), we modified SITE to try that. In an earlier experiment, this had led to much worse performance. In the specific experiment here, the result was that the cumulative area grew slightly more slowly. The process stopped after 90 observers, covering 96% of the cell, had been selected.

3.4   Arbitrarily Culling Tentative Observers

Since there were so many highly visible tentative observers, we suspected that a certain redundancy. So, we modified the previous experiment as follows. Of the 1008 tentative observers written by FINDMAX, we deleted every second observer, and reran SITE. Surprisingly, there was essentially no deterioration in the results. SITE found 180 observers to cover 1439522 points, in only 26 elapsed seconds.

More experimentation is needed to determine a reasonable minimal number of tentative observers, as a function of the input parameters.

3.5 Lower Height

One might object that the relatively high observer and target height (100m) in the above experiments, which led to many highly visible points, made the process easier. Therefore, we lowered those heights to 30m and reran the process. In this case, SITE needed 324 observers to cover 99% of the cell. The elapsed time for the whole process was only 527 seconds, with the increase due to SITE needing to use more observers. The issue is not just that the average visibility of even the top observers is a little lower, but also that the viewsheds have more holes that require other viewsheds to fill.

We next stressed the system by repeating the process with the observer and target height at 10m. This time, SITE needed 515 observers to cover 98% of the cell, using 517 elapsed seconds. One reason for needing more observers was that, in some blocks, even the best observers had only 80% visibility index. Figure 5 shows samples of the some of the best viewsheds.

Figure 5: Some of the best Viewsheds for H=10

Figure 6 on the next page shows the cumulative viewshed after 10, 20, 30, ..., 90 observers have been included.

3.6 Larger Radius

We then stressed the system by increasing the viewshed radius from 100 points to 1000 points, while keeping the height at 10. By the time SITE had added 53 observers, 99% of the cell had been covered, altho SITE was able to place 217 different observers before no more would cover even a single new point. This process took 3800 elapsed seconds, mostly in VIEWSHED, since its time depends on the radius.

3.7 Hailey E Cell

Finally we tried the USGS Hailey East level 1 DEM cell, with observer and target height of 10, and radius 100. This cell is more mountainous, so the average visibility index would be expected to be less. For this test, SITE used 991 of the 1008 tentative observers before giving up with only 94% of the area covered. However, the first 240 observers covered a full 90% of the area. This suggests that trying to cover the last few points may be counterproductive, unless we add more tentative observers to cover the points that are still hidden.

Figure 8 shows the cumulative viewshed, and the area of the latest added viewshed, as a function of the number of viewsheds added. As before, there is little relation between a tentative observer's visibility index and how soon it is added.
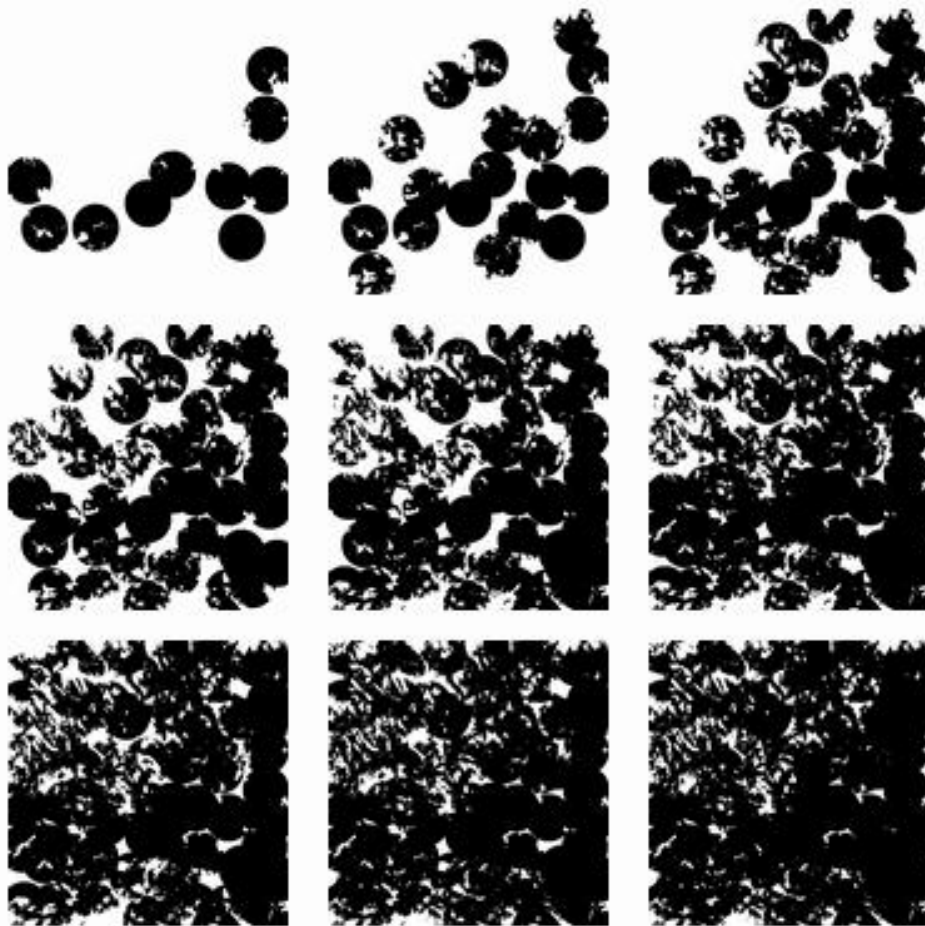
Figure 6: Cumulative Viewshed for R=100, H=10, After 10, 20, ... 90 Observers Added, for Lake Champlain W

4   The Future

The various tradeoffs mentioned above on page 7, and the above experiments illuminate a great opportunity. They tell us that shortcuts are possible in siting observers, which will produce just as good results in much less time.

Another area for investigation is the connectivity of either the viewshed, or its complement. Indeed, it may be sufficient for us to divide the cell into many separated small hidden regions, which could be identified using the fast connected component program described in Nagy et al. (2001).

There is also the perennial question of how much information content there is in the output, since the input dataset is imprecise, and is sampled only at certain points. A most useful, but quite difficult, problem is to determine what, if anything, we know with certainty about the viewsheds and observers for some cell. For example, given a set of observers, are there some regions in the cell that we know are definitely visible, or definitely hidden?

This problem of inadequate data is also told by soldiers undergoing training in the field. Someone working with only maps of the training site will lose to someone with actual experience on the ground there.
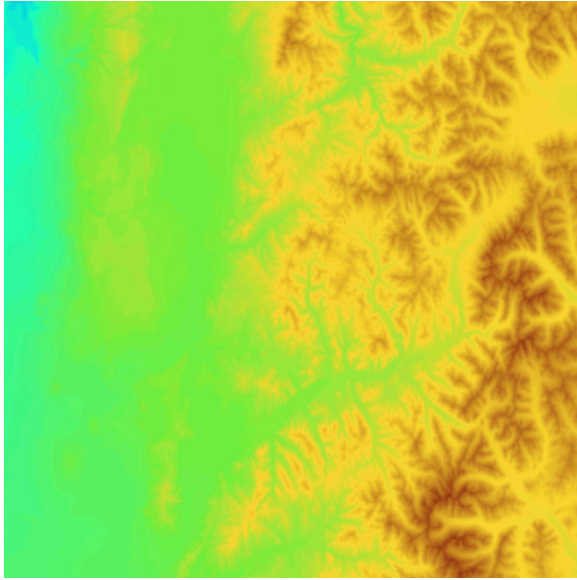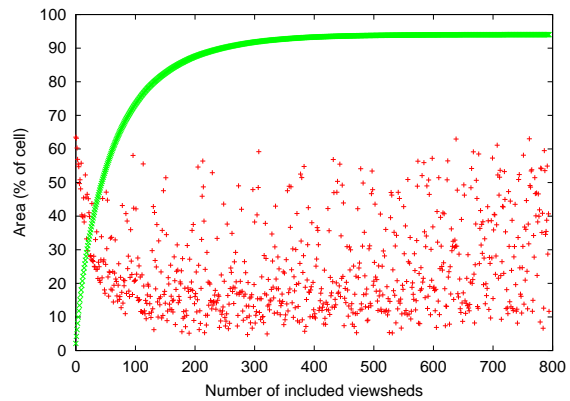
Figure 7: Hailey East Cell

Figure 8: Area of Last Inserted Viewshed $\times 30$, and Cumulative Visible Area, as Observers are Inserted, for Hailey-E Cell.

Finally, the proper theoretical approach to this problem would start with a formal model of random terrain. Then we could at least start to ask questions about the number of observers theoretically needed, as a function of the parameters. Until that happens, continued experiments will be needed.

5 Acknowledgements

6 References

Agarwal P and Sharir M (1999) Davenport-Schinzel sequences and their geometric applications. *in* J.-R Sack and J Urrutia, eds, Handbook of Computational Geometry. Elsevier Science Publishers B.V. North-Holland Amsterdam

De Floriani L, Jeanne P and Nagy G (1993) Visibility-related image features. *Pattern Recognition Letters* **13**, 463–470

De Floriani L and Magillo P (1994) Visibility algorithms on DTMs. *Int. J. Geographic Information Systems* **8**(1), 13–41

Fisher PF (1993) Algorithm and implementation uncertainty in viewshed analysis. *Int. J. Geographical Information Systems* **7**, 331–347

Franklin WR (2000) Applications of analytical cartography. *Cartography and Geographic Information Systems* **27**(3), 225–237

Franklin WR (2002) Siting observers on terrain. *in* Symposium on Spatial Data Handling. Ottawa

Franklin WR and Ray C (1994) Higher isn't necessarily better: Visibility algorithms and experiments. *in* T. C Waugh and R. G Healey, eds, Advances in GIS Research: Sixth International Symposium on Spatial Data Handling. Taylor & Francis Edinburgh pp. 751–770

Lee J (1992) Visibility dominance and topographic features on digital elevation models. *in* P Bresnahan, E Corwin and

D Cowen, eds, Proceedings 5th International Symposium on Spatial Data Handling. Vol. 2 International Geographical Union, Commission on GIS, Humanities and Social Sciences Computing Lab, U. South Carolina, Columbia, South Carolina, USA pp. 622–631

Nagy G, Zhang T, Franklin W, Landis E, Nagy E and Keane D (2001) Volume and surface area distributions of cracks in concrete. *in* 4th International Workshop on Visual Form. Capri, Italy

Ray CK (1994) Representing Visibility for Siting Problems. PhD thesis, Rensselaer Polytechnic Institute

Shannon RE and Ignizio JP (1971) Minimum altitude visibility diagram – MAVD. *Simulation* pp. 256–260

Shapira A (1990) Visibility and terrain labeling. Master's thesis, Rensselaer Polytechnic Institute