International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 3 (2011) pp. 143-149

Multiple Observer Siting in Huge Terrains Stored in External Memory

Salles V. G. Magalhães[1], **Marcus V. A. Andrade**[2] **and W. Randolph Franklin**[3]

[1]Departamento de Informática, Universidade Federal de Viçosa,
Viçosa, Brazil
smagalhaes@dpi.ufv.br

[2]Departamento de Informática, Universidade Federal de Viçosa,
Viçosa, Brazil
marcus@dpi.ufv.br

[3]Electrical, Computer, and Systems Engineering Dept., Rensselaer Polytechnic Institute,
Troy, USA
wrf@ecse.rpi.edu

Abstract: **This paper presents an heuristic method to give an approximated solution to the observer siting problem on high resolution terrains that are too large to be processed in the internal memory. Informally, the problem is to determine an optimal positioning of as few as possible observers for being able to observe as many target points as possible. Tests have shown that the proposed heuristic can solve this problem using, on average, fifteen percent fewer observers than another heuristic described in the literature. This will permit more efficient positioning of facilities such as mobile phone towers, fire observation towers, and vigilance systems.**

Keywords: Heuristics, Geographic Information System, External Memory Algorithms, Terrain Visibility , Observer Siting

I. Introduction

Advances in remote sensing have produced large quantities of high resolution geographic data that require new Geographic Information Systems [1] (GISs) techniques to process.

The Earth's surface elevation (terrain) data is usually approximately represented by a digital elevation matrix (DEM) that stores the elevations of regularly sampled terrain points. Elevations of intermediate points are approximated using some interpolation process [2]. This simple representation requires more space for very large datasets, even after compression, than is available in most computers' internal memory. Therefore we require efficient applications to process the data in external memory.

The design and analysis of algorithms to manipulate data in external memory needs to focus on optimizing the data transfer since this is much slower than internal memory access. The algorithm design goal is to minimize the number of external memory accesses since this is the dominant cost.

An important group of GIS applications on terrain concerns visibility, i.e., determining the set of points on the terrain are visible from some particular observer. The observer can be located at some height above the terrain. Applications include telecommunications, environmental planning, autonomous vehicle navigation, and military monitoring [3],[2],[4],[5]. One important problem is the positioning of a given number of facilities in order to optimally "cover the terrain". These facilities may be radio, TV, internet or mobile phone towers, and monitoring towers [6],[7],[8].

In this paper we present a solution to the multiple observer siting problem, i.e., a method to site facilities in terrain represented by huge elevation matrices that are stored in the external memory. This work extends the approach presented in Franklin [9],[10] (developed to process data in internal memory) to external memory processing. The idea is to divide the terrain in smaller pieces and process each piece in the internal memory.

The paper is organized as follows: section II presents some definitions used in the algorithms description; in section III, the method proposed by Franklin et al [9],[10] is described and in section IV we describe the proposed method; in section V we describe an improvement that was implemented in the data structure used by the method and, in section VI, the method complexity is analyzed. Finally, in section VII we present the results of some tests and in section VIII our conclusions.

II. Terrain Visibility

Definition 1: A *terrain* represents a region of the earth's surface. In the context of this paper, it is a scalar field over a square (in the relevant coordinate system) domain. The terrain's value at any point is the elevation of the corresponding point of the earth's surface above some reference ellipsoid called the *geoid* that represents sea-level. For this paper, terrain is represented by a matrix of elevation posts on a square grid, whose vertical and horizontal spacing is uniform either in distance, e.g., 30m, or in angle, e.g. 1 arc-second.

Other representations such as triangular splines, or Triangulated Irregular Networks (TINs), are also common. Their implementation is more complicated, especially for operations such as line-of-sight. When only a coarse approximation to the terrain is needed, a TIN is compact. However, in this case, a DEM may be lossily compressed to an extremely small size. Indeed, by separating the abstract data structure, the grid, from the concrete implementation, the compression algorithm, the DEM is conceptually a better designed representation, while easier to implement, and apparently equally compact.

Definition 2: An *observer* is a point in space from which we wish to see or communicate with other points in space, called *targets*. The usual notation for observer and target is O and T. The *base points* of O and T are the points on the terrain directly below O and T, respectively. They are denoted as O_b and T_b. O and T are each at height h above O_b and T_b.

The possible generalization of having separate heights for the observer and target, while adding another degree of freedom to the experiments, does not seem to add anything new to the science of the problem. That is, earlier tests lead us to believe that our conclusions are general. So, to simplify the algorithm description, we will consider the observers and targets at the same height.

Definition 3: The *radius of interest*, R, of O is the radius of the circle centered on O_b which contains all the points that the observer can see, in the absence of obstructions. E.g., if O is a radio transmitter, R is a function of the transmitter power and receiver sensitivity. For convenience, R is measured between O_b and T_b rather than between O and T, which is equivalent when h is much smaller than the radius of the earth.

Definition 4: T is *visible* from O iff $|T_b - O_b| \leq R$ and there is no terrain point between O and T blocking the line segment, called the *Line of Sight (LOS)*, between them; see Figure 1. In this Figure, T_1 is visible from O but T_2 is not.

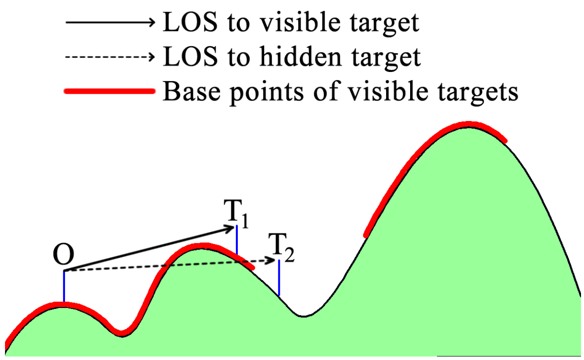


Figure. 1: Visibility queries using a line of sight

Determining whether or not some terrain blocks the LOS is non-trivial, and a subject of current research. The problem is that the terrain is defined only at the points, or posts, in the DEM, while the LOS in general passes between adjacent posts; that is, the LOS may not pass on the posts - see Figure 2. Indeed, the numbers in the DEM may not even be point elevations but rather averages over some areas.

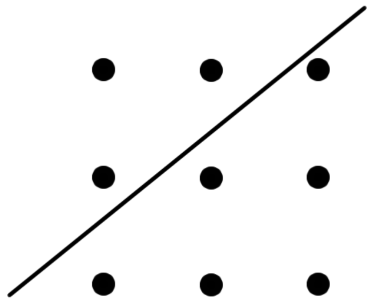


Figure. 2: LOS passing between (not through) elevation posts

Therefore the terrain elevation must be interpolated along the LOS. The true elevation at an interpolation point might well be greater than the greatest adjacent post, or less than the smallest adjacent post. Small, apparently unimportant, changes in the interpolation algorithm might cause major changes in the targets' visibility. Consider Figure 3, showing the United States Geological Survey Lake Champlain West DEM, where the lowest and highest elevations are indicated by red and blue hues, respectivelly, with the observer (the white point) positioned on Mt Marcy, the highest point, and the curvature of the earth being ignored. In this example, interpolating by using the maximum adjacent post elevation instead of the minimum adjacent post elevation changes the visibility of one half of all the targets.

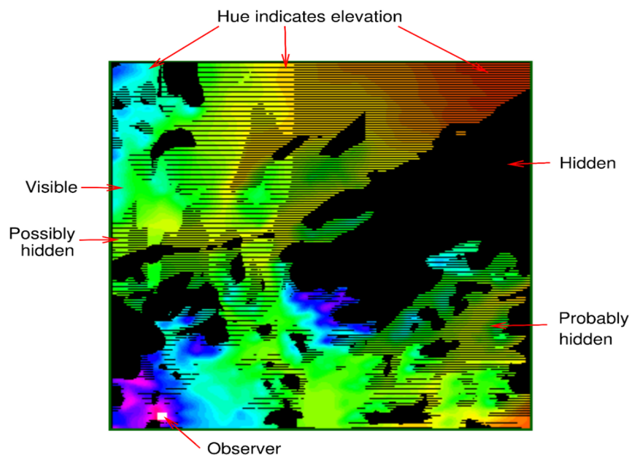


Figure. 3: Visibility of one-half of the terrain affected by LOS interpolation rule

Definition 5: The *viewshed*, V, of O is the set of base points whose corresponding targets are visible from O. V is stored as a bit matrix.

Definition 6: The *visibility index*, ω, of O is the number of targets with base points within the circle C of radius R centered at O_b that are visible from O.

Points with a large ω are good candidate places to site observers in order to maximize the area of the terrain that is seen by at least one observer. ω, which is simply the number of 1-bits in V, is commonly estimated by counting how many of a random sample of targets inside C are visible.

Definition 7: The *joint viewshed*, $\mathcal{V}$, of a set of observers $\mathcal{O} = \{O_i\}$ is the union of the individual viewsheds V_i, i.e., the bitwise-*or* of their bit matrices.

Definition 8: The *joint visibility index*, Ω, of $\mathcal{O}$ is the number of targets in the terrain that are visible from at least one observer in $\mathcal{O}$. Ω may be normalized to be a percent of the terrain area.

Definition 9: *Multi-observer siting* means optimizing the locations of a set of observers, called *siting*, so that Ω is as large as possible. This has important practical facilities-location applications, such as siting mobile phone towers, fire monitoring towers, and radar systems.

In this paper we will consider the following equivalent Multi-observer siting problem: to find the minimum number of observers to cover a terrain, i.e., such that all the terrain points are visible in the joint viewshed. This is a NP-hard [4] problem, and can be reduced to classical set coverage optimization [11]. The set cover problem is: Given a set $S = \{s_i\}$ of sets, choose $C = \{c_i\} \subset S$ such that $\cup c_i = \cup s_i$ and $|C|$ is minimized [12]. Informally, find the smallest number of sets s_i whose union is equal to the union of all the s_i.

We will present an heuristic approximate solution to the following variation of the problem defined above: to find the minimum number of observers to achieve a given terrain coverage. Our solution can also be adapted to maximize the joint viewshed with a fixed number of observers.

III. The Site method

Since the observer siting problem is NP-hard, Franklin [9],[10] presented Site, an approximate heuristic solution, to find a set of observers to cover the terrain. Site uses a greedy approach to select the set $\mathcal{O} = \{O_i\}$ of observers from a much larger set $\mathcal{P} = \{P_i\}$ of potential observers, together with their viewsheds. Initially $\mathcal{O} = \{\}$. At each step, the P_i that would most increase the joint viewshed of $\mathcal{O}$ is inserted into $\mathcal{O}$. The details are as follows.

1. Estimate the visibility index of each point in the terrain M. More precisely, determine the points that have a certain minimum visibility index with a certain confidence level. This may be achieved by sampling random targets.
2. Compute $\mathcal{P} = \{P_i\}$ as the set of points with the largest visibility indexes. E.g., with a typical M with $|M| = 1\,442\,401$ (for a 1201×1201 matrix of posts, the standard for a level-1 USGS DEM), $|\mathcal{P}|$ might be 1000.

   However, do not select two points that are too close together, since their viewsheds will probably overlap considerably, and hence, one of them will be redundant.
3. Compute V_i, the viewshed of each P_i. V_i is that region of the terrain visible from an observer sited at P_i. V_i is conveniently stored as a bitmap.
4. Initialize $\mathcal{O} = \{O_i\} = \{\}$. This will accumulate the set of actual observers. $\mathcal{O} \subset \mathcal{P}$.
5. Initialize $\mathcal{V}$, the joint viewshed of $\mathcal{O}$, that is the union of the viewsheds of all the O_i, to all 0 bits.
6. Repeat the following until a termination condition is satisfied. Typical conditions include that $|\mathcal{O}|$ reaches a certain maximum, or $area(\mathcal{V})$ a certain minimum.
   (a) Iterate through $\mathcal{P}$ to find the P_i that would cause $\mathcal{V}$ to increase the most. That involves repeatedly finding the area of the union of two bitmaps ($\mathcal{V}$ and V_i), which is very fast.
   (b) Insert that P_i into $\mathcal{O}$ and update $\mathcal{V}$.
   (c) However, if it is desired that $\mathcal{V}$ be a connected set, to enforce *intervisibility*, then do not pick a P_i that would cause the new $\mathcal{V}$ to be disconnected.

A. *Using Site on huge terrain*

If the terrain is too large to be stored in internal memory, the obvious extension of Site is to simply subdivide the elevation matrix M into subregions M_i each small enough to fit into internal memory, and then use Site on each M_i. The problem is that the viewshed $\mathcal{V}$ of a point may cross into several M_i. Even for a small radius of interest, $\mathcal{V}$ may easily overlap four M_i. That is, visibilities on any particular M_i are affected by points not in that M_i. There are several possible solutions.

First, since this is only an approximate method, we might simply ignore the effect of viewsheds that cross into another M_i and work with truncated viewsheds. Since the effective viewsheds of observers near the edge of the M_i would be too small, those observers' visibility indexes would be underestimated and they might be excluded when it would be optimal to include them. The effect of this on the quality of the resulting set of observers would need to be determined empirically, but might be acceptable if the M_i are much larger than the viewsheds.

To illustrate this situation, see Figure 4 where the terrain was divided into four regions: M_1, M_2, M_3 and M_4 and a potential observer O close to M_2's lower left corner is being considered. It is possible that O's viewshed V is large, but, since V is split between $M_{1 \ldots 4}$, the amount of V inside M_2 is small. Therefore O, erroneously, might not be added to the set of potential observers for M_2.

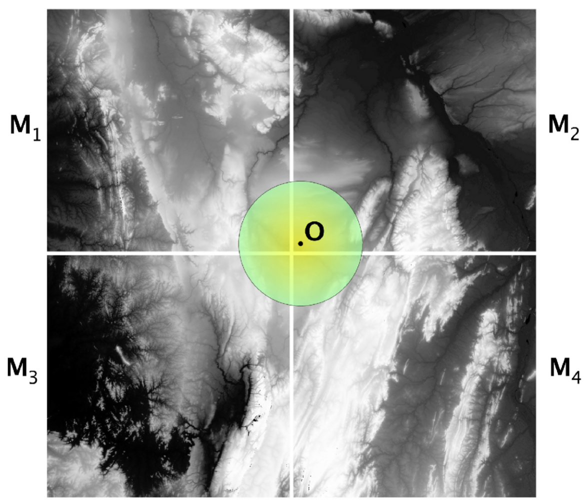


Figure. 4: Terrain subdivision into four subregions

Also, even if O is sited in M_2, observers might not be sited optimally in M_i because the part of V in M_1 will be ignored (because the M_i are being processed independently).

Another limitation of this approach is that it reaches a uniform coverage in each subregion when processing a given terrain (for example, if the desired coverage is 70% then the method will site observers in each subterrain M_i until it reaches 70% of coverage in M_i). This strategy could lead to bad solutions since it might be better to achieve a small coverage in a subregion where it is difficult to site observers (in other words, in a subregion that needs many observers to reach a given coverage) and balance this subregion by reaching a bigger coverage in a subregion where it is easier to site observers (for example, a plain region which needs few observers to reach a big coverage).

IV. The EMSite method

EMSite (External Memory Site), our new method, extends the idea described above in Section III-A in order to consider the influence of observers sited near to the borders of the subregions. It also correctly computes the joint viewshed of any given set of observers while working within the available main memory and minimizing I/O.

The major idea is to add a band of width R around each region when subdividing the terrain into small subregions, During the processing of each subregion, observers will be sited only at points in the core region, not in the additional band. However, those observer's viewsheds can extend into the band. See Figure 5, where terrain subregion M_2 has an additional band A. During M_2's processing, observers may be sited only in M_2 but their computed viewsheds may extend into A, and the viewshed portions in A will be properly used.

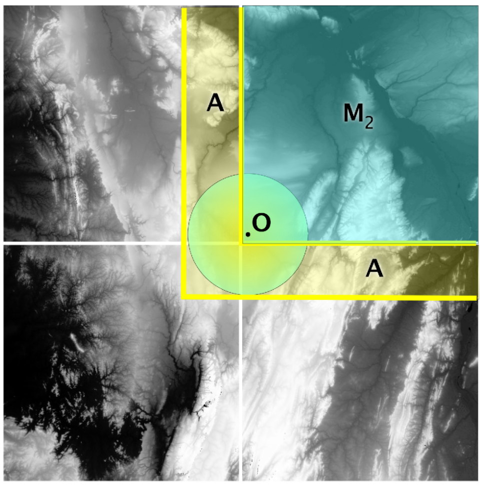


Figure. 5: Additional band for M_2

Another key to EMSite is the sophistication of the observer siting heuristics, which proceeds in two or more stages. The joint viewshed for the whole terrain is represented by a matrix $\mathcal{V}$ stored in external memory. EMSite proceeds as follows.

1. Initialize $\mathcal{V}$ by setting all points to be not visible (since no observers to see them have been sited yet).
2. For each subregion M_i:
   (a) Execute Site in M_i to site an initial number of observers $\mathcal{I}$, e.g., $|\mathcal{I}| = 2$. That means to find two observers that increase the joint viewshed as much as is possible.
   (b) However, during this process note that (if $i > 0$) some points in M_i are already visible by observers sited in M_j for $j < i$. That is, don't identify observers that are good for M_i when considered in isolation, but identify observers whose viewsheds cover parts of M_i that have not already been covered by observers sited in the earlier M_j.
   (c) Compute $\mathcal{I}$'s joint viewshed as a bit matrix $\mathcal{C}_i$ that may extend into M_i's border zone, that is into other subregions.
   (d) $\mathcal{V} \leftarrow \mathcal{V}$ bitwise-or $\mathcal{C}_i$
3. Now that an initial set of observers has been sited, process the regions again to fill in visibility gaps.
4. Choose δ, an amount by which any future observer that is sited will be required to increase the size of the joint viewshed.
5. Sort $\mathcal{M} = \{M_i\}$, from large to small, by the increment in the joint viewshed in each M_i given by the last observer sited. The goal is to start siting observers in regions that are easier to cover.
6. For each M_i in sorted order, site more observers with Site, stopping when a new observer would add less than δ new visible area to the joint viewshed.
7. Stop when either the joint viewshed is the desired size, or the maximum number of observers is used.
8. Otherwise, reduce δ and go back to step 5.

The presented heuristic first sites few observers in each subterrain (step 2) in order to estimate the facility of siting observers in them (it is assumed that the bigger the contribution of the last observer sited the easier is to site observers). This estimative is used by the method in order to sort the subterrains and process first the ones where it is easier to do the siting (this is a greedy approach).

As stated in the section III-A, one of the problems with the direct adaptation of Site to external memory is that it may return bad results when a uniform desired coverage is used as a stoping criterea in each subterrain. To solve this problem, the proposed method uses the contribution of the observer that is being added to the current solution as stopping criterea. The idea is to define a value δ and use it as an inferior threshold for the contribution of observers that are being sited. Thus, during the processing of a subregion (in step 6), if the contribution of the current observer selected by the greedy method is smaller than δ, the siting method is stopped and another region is selected to be processed. After processing all regions δ is decreased (in step 8) and the siting method continues in the terrains using a smaller threashold. The method stops when the desired terrain coverage is reached.

In the proposed method, δ was decreased by multiplying it by a constante equal to 0.9. Thus, in each step δ is decreased in 10%. The initial value of δ was set to $0.8\pi R^2$; So, in the first iteration, the method will site observers whose contribution is greather or equal to 80% of the area of a fully visible viewshed.

Figure 6 graphically displays those stages. The radius of interest was chosen small to make the problem harder, since then more observers are needed. Since the first observers to be sited are the best observers, their viewsheds are complete circles. As seen in (c) and (d), the later observers to be chosen have incomplete viewsheds; their visibility indexes are less than 1.

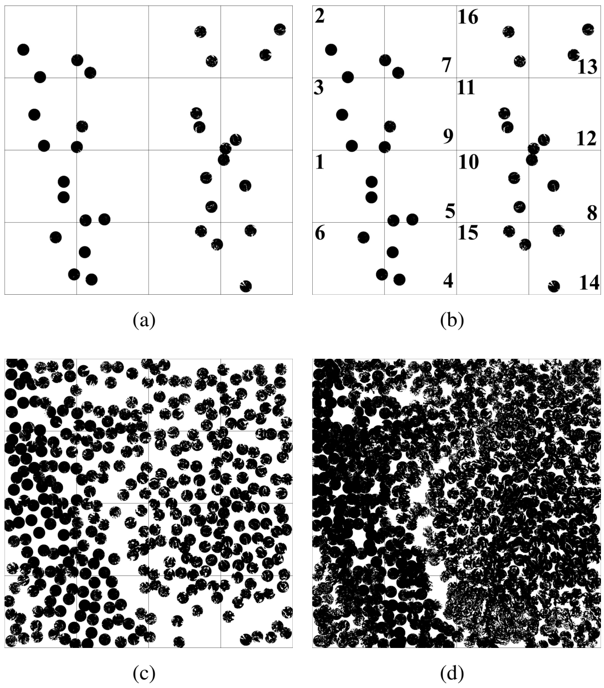


Figure. 6: EMSite stages: (a) Two observers are sited in each region. (b) The regions are sorted in decreasing order by the increment in their joint viewshed given by the last observer sited in each region. (c) Partway through the process, more observers are sited in each region until the contribution of the last observer to the joint viewshed is smaller than a threshold. (d) The final result showing the desired joint visibility index..

V. Improvements to viewshed's data structure

We also improved the original method Site by changing the viewshed representation, originally a bit matrix of the same size as the terrain. Now, each viewshed is represented as a square bit matrix of side $2R-1$ where R is the observer radius of interest, together with a header describing the number of bytes in each viewshed line, the viewshed bounding-box coordinates in the original terrain, R, and the observer's coordinates, both in the bounding-box and in the original terrain.

Since the viewsheds now use less storage, time and space resource requirements needed by the observer siting method are significantly smaller.

VI. Analysis of the method

Since the terrain M is divided in k subterrains M_1,M_2, M_3, ... , M_k (each one with dimensions $S \times S$) and each subterrain has C potential observers (each observer represented by its viewshed, with radius R), the external memory operations done by the method during the processing of each subterrain M_i can be found using the expression:

$$\frac{C(2R)^2}{8} + \frac{4S^2}{8} = \frac{CR^2 + S^2}{2} \quad (1)$$

In this expression, $C(2R)^2/8$ represents the loading of the C viewsheds representing candidate points in M_i and $4(S^2)/8$ represents the operations of cuting the cumulative viewshed of M relative to M_i, loading it in the siting method, storing the resulting cumulative viewshed processed by the siting method and, finally, joining it with the Ms cumulative viewshed.

It is important to say that these operations are done during the processing of each subterrain and , thus, the total number of I/O operations done by EMSite may be defined by the expression:

$$L\frac{CR^2 + S^2}{2} \quad (2)$$

Where L indicates the total number of times that subterrains were processed. It is important to say that this number depends on the target coverage, on viewsheds and terrains characteristics and on the way that δ is decreased.

VII. Results

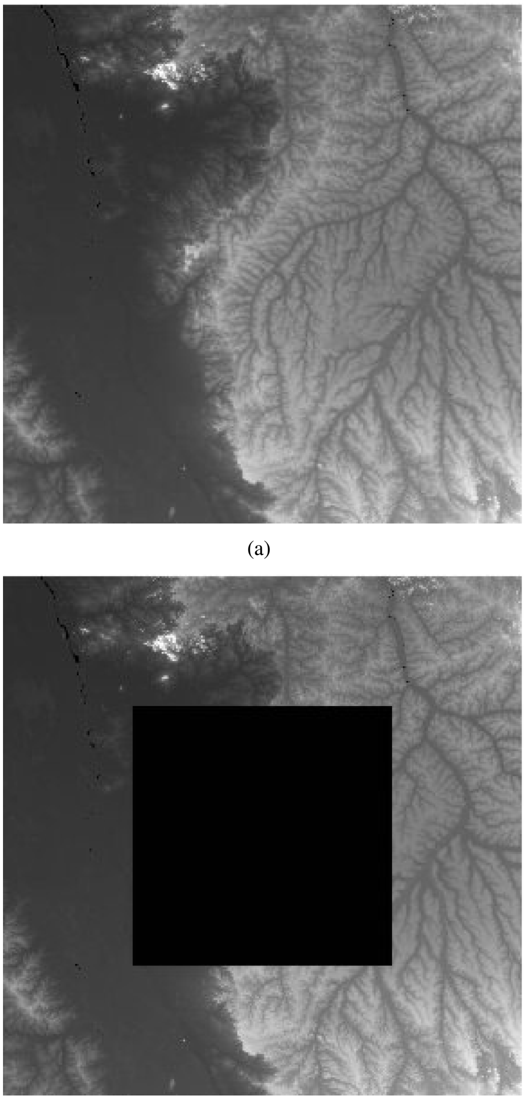
We implemented EMSite in C++. This is apparently the only observer siting method able to process data in external memory. We tested EMSite by comparing it to the simple external version of Site, described above in section III-A, using terrain representing the northeast of Brazil — see Figure 7. The test data is from the NASA SRTM [13] as follows.

- Terrain 1: A 4804×4804 block from northeast Brazil.
- Terrain 2: Another 4804×4804 block from northeast Brazil. The elevation of the points around the center of the region was modified to be equal to 0. This was done to simulate a terrain with a center plane area surrounded by hills. This seemed to be a more difficult dataset to achieve a good joint visibility index on, and we wished to stress-test EMSite.

In all tests, the set of potential observers was built with 16000 points selected by *findmax/vix* methods proposed by Franklin et al. [9],[10].

Table 1 presents the results obtained by EMSite and Site using several different values of desired joint visibility index Ω, with observers' and target's heights of $1m$ and $15m$ above the ground. Also, it was used two different R (radii of interest): 100 and 250 points. Column *# Obs* shows the number of observers sited by each method. Column *EMSite Improv.* gives the improvement of EMSite compared to Site.

Both sets of terrain were subdivided into 16 subregions. Although each terrain size is about 44MB, it can't be processed in internal memory using the original Site because this method needs to load several viewsheds into the main memory during the observer siting. More precisely, the original Site method would need to load 16000 viewsheds in memory in order to process these terrains. Since each viewshed is stored as a bit matrix, Site would need a total of



(a)

(b)

Figure. 7: Images representing the terrains used in the tests: Terrain 1 (a), Terrain 2 (b).

$(16000 \times (4804 \times 4804))$ bits of memory (more than 42 gigabytes).

As one can see, EMSite can achieve the desired Ω using a much smaller number of observers than Site. The median number of fewer observers is 25%.

In Table 2, the two methods were reevaluated with the observers and targets now positioned at 15m above the ground. Again, EMSite produced better results, using a much smaller number of observers for the same Ω; the median improvement is 6%.

Finally, Table 3 shows the results of tests considering higher desired Ω. Notice that, in some cases, as for $\Omega = 88\%$, EMSite can be much better than Site using less than a half of observers to achieve the desired Ω. In other cases, as for $\Omega = 90\%$, EMSite can succeed while Site can not.

VIII. Conclusion

This paper presented EMSite, which is able to site observers on huge terrain datasets that can not be stored in internal memory, and so need to be processed externally. The test results show that EMSite works well, and, compared with a straight adaptation of Site to process huge terrains, EMSite required up to 54% fewer observers, with a median of 15% fewer. This may lead to significant financial savings, example, by decreasing the number of mobile phone towers needed to cover a city.

The EMSite source code is available in [14].

Our next step is to adapt EMSite to site observers with a very large radius of interest, so that the terrain subregion including the band does fit into internal memory.

Table 1: Comparison of the number of observers used by EMSite and Site to achieve the desired Ω, with observers' and targets' heights 1 meter above the ground.

Terrain	R	Desired Ω	EMSite # Obs	Site # Obs	EMSite Improv.
1	100	25%	363	402	11%
		50%	992	1074	8%
		75%	2 810	2 940	4%
		80%	3 963	4 057	2%
2	100	25%	221	344	36%
		50%	656	879	25%
		75%	1 724	2 305	25%
		80%	2 294	3 143	27%
1	250	25%	84	113	26%
		50%	291	338	14%
		75%	954	1 073	11%
		80%	1 574	1 964	20%
2	250	25%	44	85	48%
		50%	179	237	24%
		75%	573	781	26%
		80%	789	1 107	29%

Table 2: Comparison of the number of observers used by EMSite and Site to achieve the desired Ω, with observers' heights 15 meters above the ground.

Terrain	R	Desired Ω	Site # Obs	EMSite # Obs	EMSite Improv.
1	100	25%	198	189	5%
		50%	409	390	5%
		75%	697	658	6%
		85%	889	833	6%
2	100	25%	197	186	6%
		50%	399	393	2%
		75%	679	643	5%
		85%	861	809	6%
1	250	75%	158	135	15%
		90%	276	235	15%
2	250	75%	145	122	16%
		90%	244	213	13%

Table 3: Comparison of the number of observers used by EMSite and Site to achieve high Ω. The observers' and targets' heights are $1m$.

Terrain	R	Desired Ω	Site # Obs	EMSite # Obs	EMSite Improv.
1		80%	1 964	1 574	20%
	250	88%	11 866	5 424	54%
		90%	–	13 259	∞

Acknowledgment

This work was partially supported by FAPEMIG – the State of Minas Gerais Research Foundation, by CNPq – the Brazilian Research Council, by FUNARBE, the Arthur Bernardes Fundation, by Sydle, by NSF grant CMMI-0835762 and by DARPA/IPTO under the Geo* program.

Notation summary

O	observer
T	target
O_b	observer's base point
T_b	target's base point
h	height of observer and target above terrain
R	radius of interest
V	viewshed
ω	visibility index of an observer
$\mathcal{O}$	set of observers
$\mathcal{V}$	joint viewshed of a set of observers
Ω	joint visibility index of a set of observers
P	potential observer
$\mathcal{P}$	set of potential observers
M	terrain
$\mathcal{I}$	initial set of observers

References

[1] R. Laurini and D. Thompson, *Fundamentals of Spatial Information Systems*. Academic Press, 1992.

[2] Z. Li, Q. Zhu, and C. Gold, *Digital Terrain Modeling - principles and methodology*. CRC Press, 2005.

[3] W. R. Franklin and C. Ray, "Higher isn't necessarily better - visibility algorithms and experiments," in *6th Symposium on Spatial Data Handling*, (Edinburgh, Scotland), 1994. ISBN 2-540-43802-5.

[4] G. Nagy, "Terrain visibility," *Computers and Graphics*, vol. 18, pp. 763–773, 1994.

[5] M. V. A. Andrade, S. V. G. Magalhes, M. A. Magalhes, W. R. Franklin, and B. M. Cutler, "Efficient viewshed computation on terrain in external memory," *GeoInformatica*, vol. 1, pp. 1–17, 2009.

[6] B. Ben-Moshe, *Geometric Facility Location Optimization*. Phd thesis, Ben-Gurion University, Israel, Department of Computer Science, 2005.

[7] Y. Ben-Shimol, B. Ben-Moshe, Y. Ben-Yehezkel, A. Dvir, and M. Segal, "Automated antenna positioning algorithms for wireless fixed-access networks," *Journal of Heuristics*, vol. 13, no. 3, pp. 243–263, 2007.

[8] S. V. G. de Magalhes, M. V. A. Andrade, and C. Ferreira, "Heuristics to site observers in a terrain represented by a digital elevation matrix," in *In Prof. of XI Brazilian Symposium on Geoinformatics (GEOINFO10*, pp. 110–121, 2010.

[9] W. R. Franklin, "Siting observers on terrain," in *Advances in Spatial Data Handling: 10th International Symposium on Spatial Data Handling* (D. Richardson and P. van Oosterom, eds.), pp. 109–120, Springer, 2002.

[10] W. R. Franklin and C. Vogt, "Tradeoffs when multiple observer siting on large terrain cells," in *Progress in spatial data handling: 12th international symposium on spatial data handling* (A. Riedl, W. Kainz, and G. Elmes, eds.), pp. 845–861, Vienna: Springer, 2006. ISBN 978-3-540-35588-5.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2001.

[12] P. E. Black, "Set cover," in *Dictionary of Algorithms and Data Structures [online]* (P. E. Black, ed.), U.S. National Institute of Standards and Technology, 17 Dec 2004.

[13] Jet Propulsion Laboratory, "Shuttle radar topography mission," 2009. `http://www2.jpl.nasa.gov/srtm`, [accessed 23-Jan-2011].

[14] S. V. G. de Magalhes, M. V. A. Andrade, and W. R. Franklin, "EMSite code," 2009. `http://www.dpi.ufv.br/~marcus/projects/EMSite.htm`, [accessed 23-Jan-2011].

Author Biographies

Salles V. G. Magalhes is currently a Graduate Student of Computer Science in Federal University of Viçosa, Brazil. He received his B.Sc degree in Computer Science from Federal University of Viçosa. His research interests include terrain simplification, heuristics, parallel programming and algorithms to process terrains stored in external memory.

Marcus V. A. Andrade is a Associate Professor of Computer Science at Federal University of Viosa. He received the B.Sc. degree in Mathematics from Federal University of Viçosa and the M.Sc. and D.Sc. degrees from University of Campinas, Brazil. His research interests include computational geometry applied to GIS, terrain modeling and simplification and geometry processing.

W. Randolph Franklin is an Associate Professor in the Electrical, Computer, and Systems Engineering Dept. at Rensselaer Polytechnic Institute. His degrees include a B.Sc. from Toronto in Computer Science and A.M. and Ph.D. from Harvard in Applied Mathematics. His prime research interest is efficient geometric algorithms on large datasets. One application is to computational cartography, e.g., compact representations of terrain and operations thereon.