

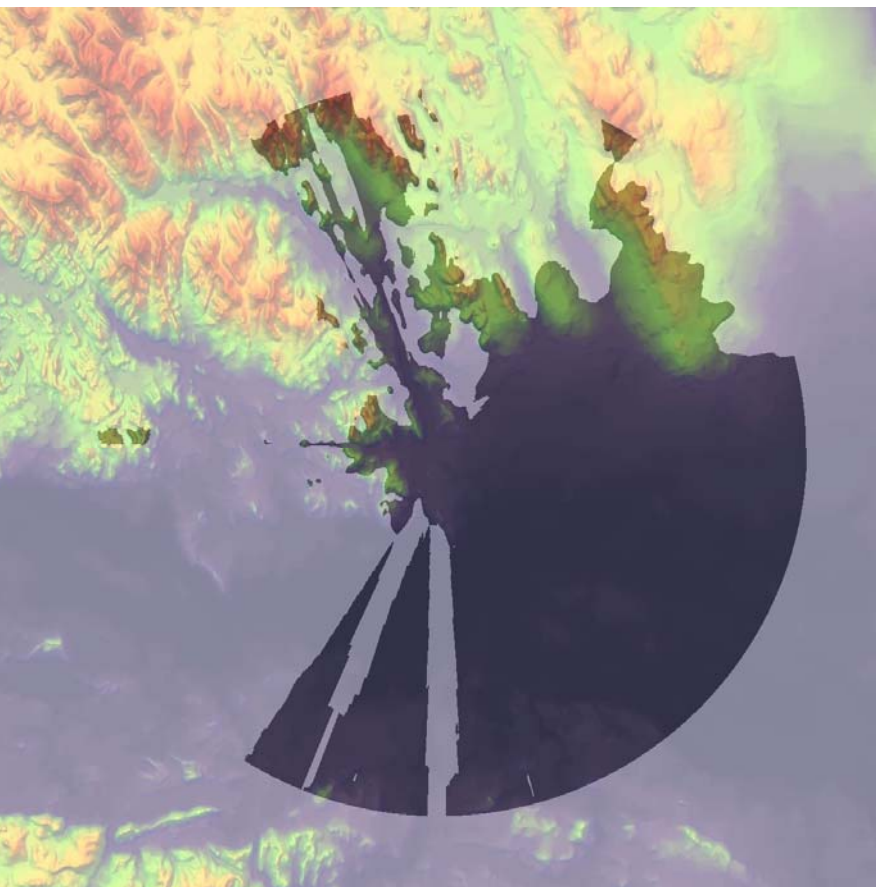


An efficient algorithm to compute the viewshed on DEM terrains stored in the external memory

Mirella A. Magalhães
Marcus V. A. Andrade
Salles V. G. Magalhães
Jugurta Lisboa Filho



What is viewshed?



- Points that are visible from an observer of the terrain
- Limited by interest ray
- Calculated with points height



Applications

- Viewshed Applications
 - Help on sitting multiple observers problem
 - Cell phone or internet towers
 - Fire watch-tower
 - Military applications
 - Sensors network
 - Surveillance positioning
- Technological advances in data collection (LiDAR)
 - Huge terrain models

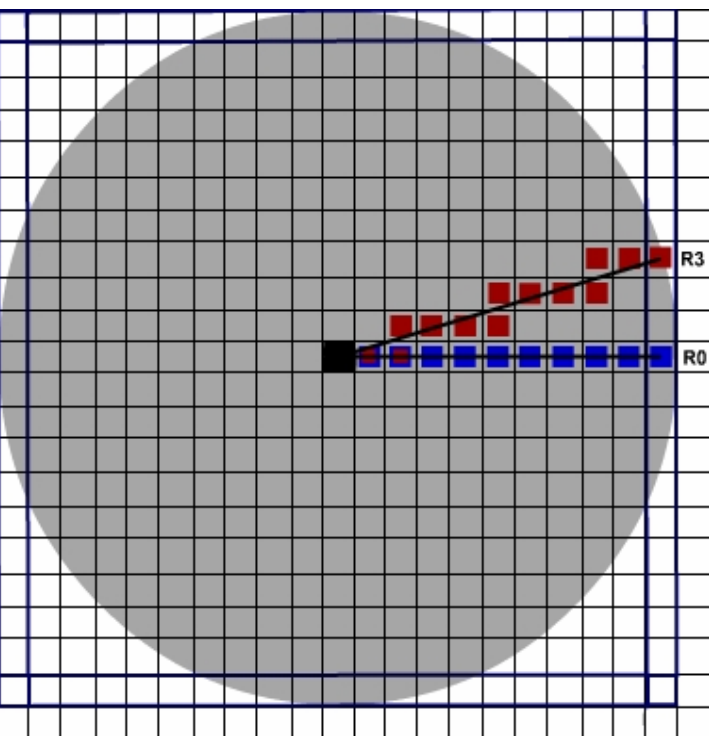


Objective

- To develop a method to calculate the viewshed on DEM terrains that exceed internal memory size
- DEM
 - Simplicity
 - Large use



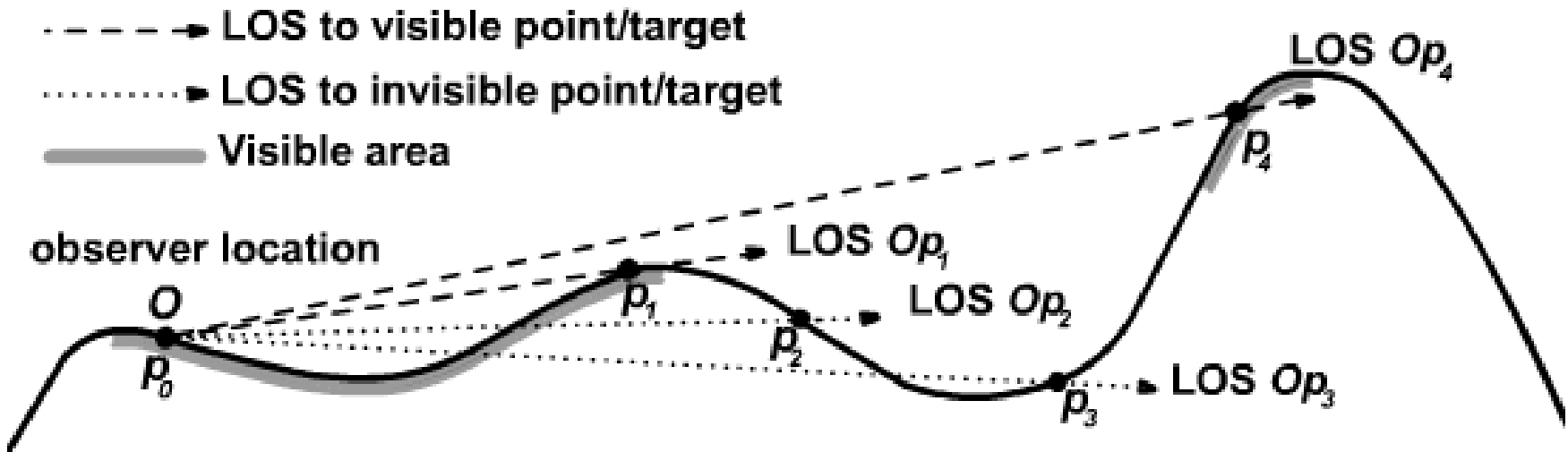
Original Viewshed Calculation



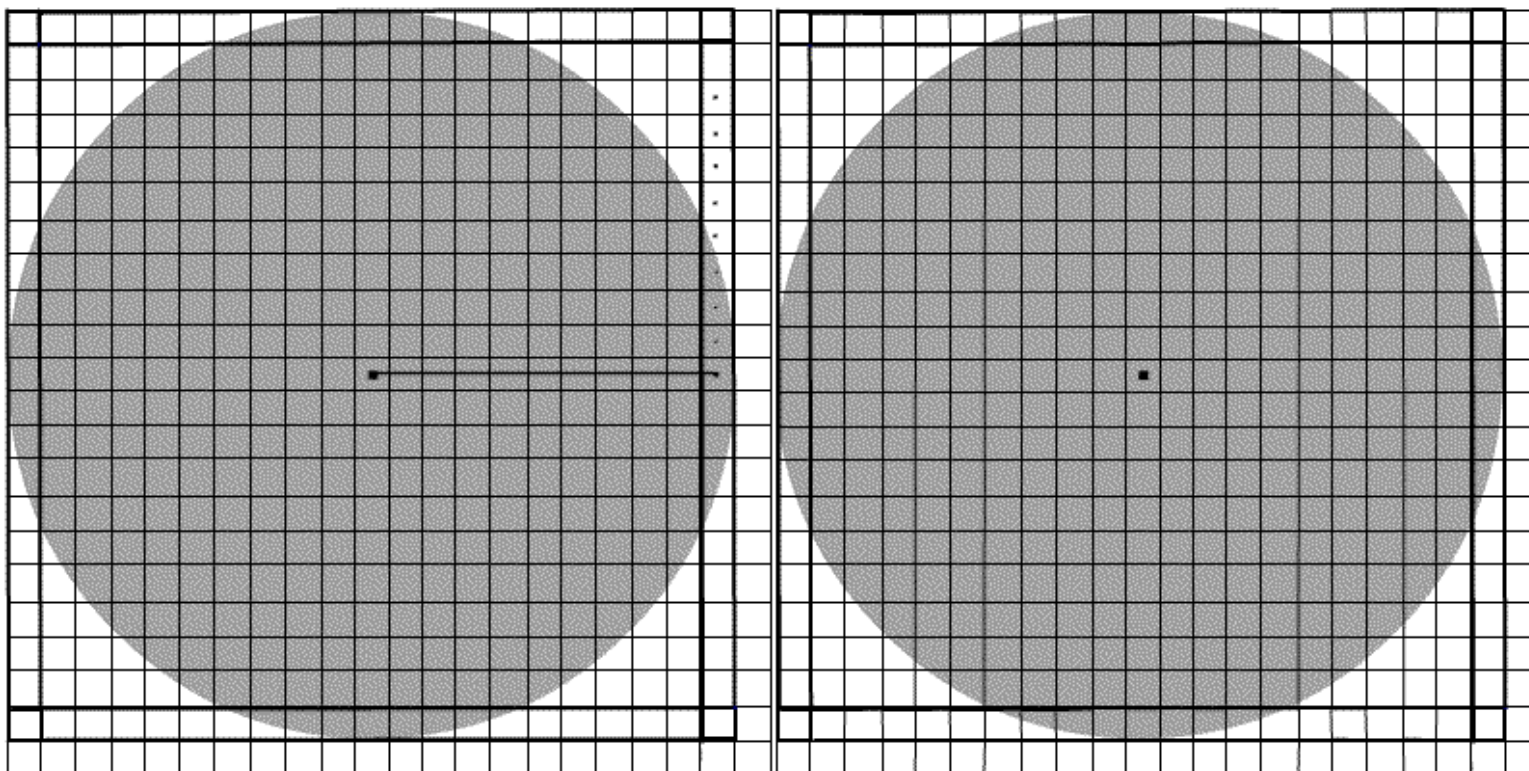
- First, all points on terrain are considered not visible
- Terrain sweep is based on “sweep rays” by the xy plane
- “Sweep ray” is a vertical cut on terrain



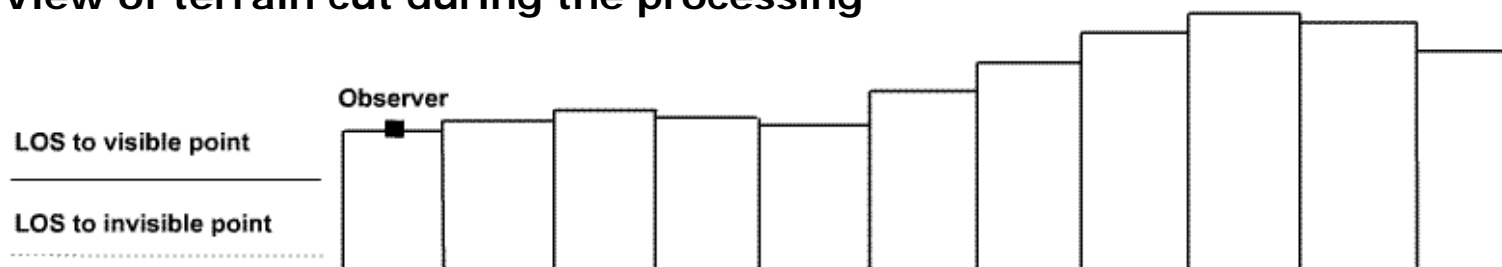
Original Viewshed Calculation



- In each cut we determine the visible points
- If the line of sight (LOS) OP (O – observer and P – current point) doesn't intercept any other point, P is visible



View of terrain cut during the processing





Internal memory algorithm problem

- Random access – inefficient on terrains bigger than the internal memory
- Sequential access is faster than random access
- Random access on disk is about 10^6 times slower than the same access in internal memory



I/O Efficient Algorithms

- The bottleneck are the I/O operations (not the CPU processing)
- The complexity is determined by the number of I/O operations



I/O efficient algorithms complexity

- Aggarwal and Vitter model
- Complexity orders
 - $O(\textit{scan}(N))$ – read N contiguous items on disk
 - $O(\textit{sort}(N))$ – sort N contiguous items on disk
- $O(\textit{scan}(N)) < O(\textit{sort}(N)) \ll O(N)$
- I/O efficient algorithms try to achieve $O(\textit{scan}(N))$ or $O(\textit{sort}(N))$ complexity



Proposed algorithm

- Uses external memory lists
- Sort algorithm (mergesort)
- *STXXL* library – Standard Template Library for Extra Large Terrains

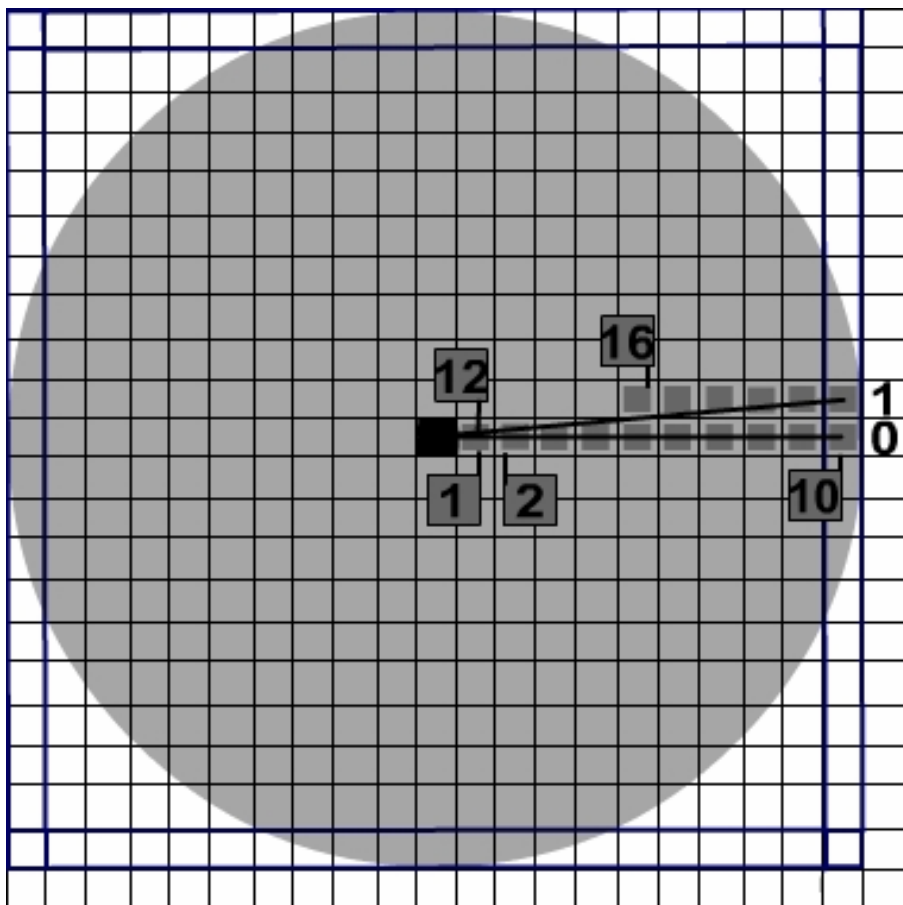


Proposed Algorithm

- Read the terrain sequentially
 - Determine the process order of each cell
 - Insert each cell in an external memory list L with the respective order of processing (index)
 - Sort L by the index
 - Calculate the visibility of each point using an algorithm similar to original



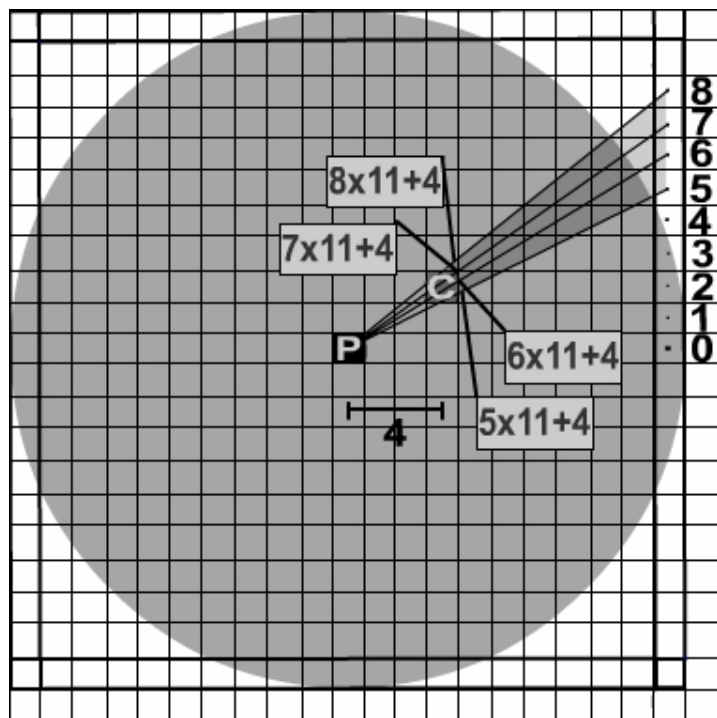
Index determination



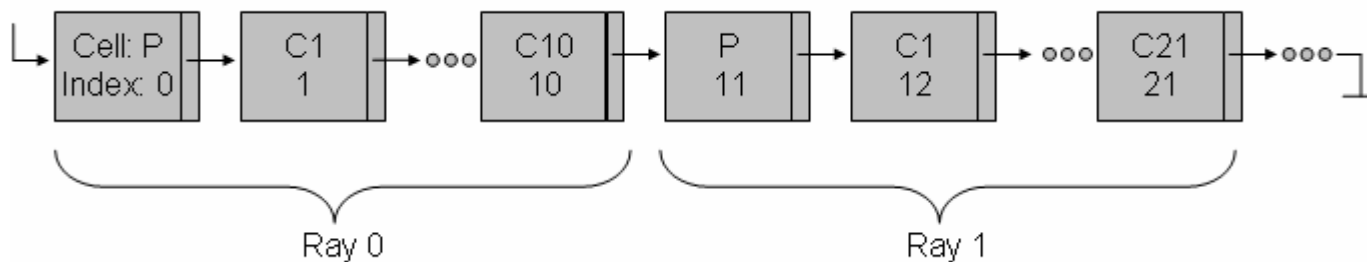
- Ray 0's points:
 - Order = 0,1,2,...,10
- Ray 1's points:
 - Order = 11,12,...,21
- $Index = N \times RAY + D$



Index Determination



- The index of each point is easily determined
- One cell can be intercepted by more than one ray
 - It is inserted more than one time with different indices



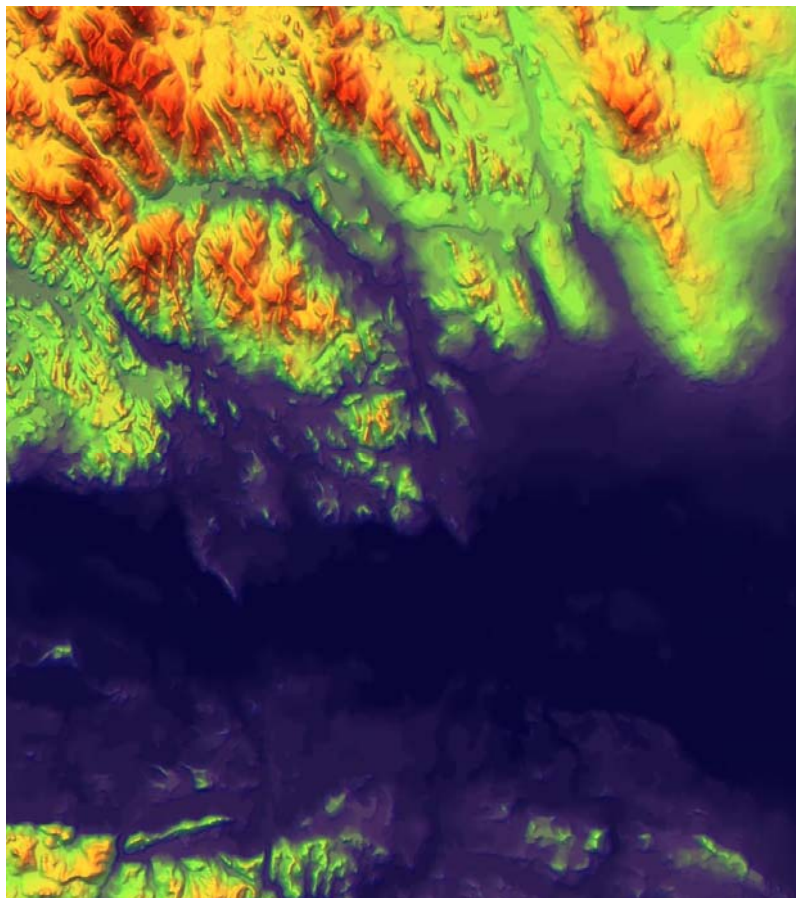


Visibility Computation

- Algorithm similar to the original using *LOS*
- All visible points are stored in another external memory list L' that is sorted by coordinates (x,y)
- The final result is a matrix with all points on the terrain. Visible points are indicated by 1 and not visible by 0.



Results



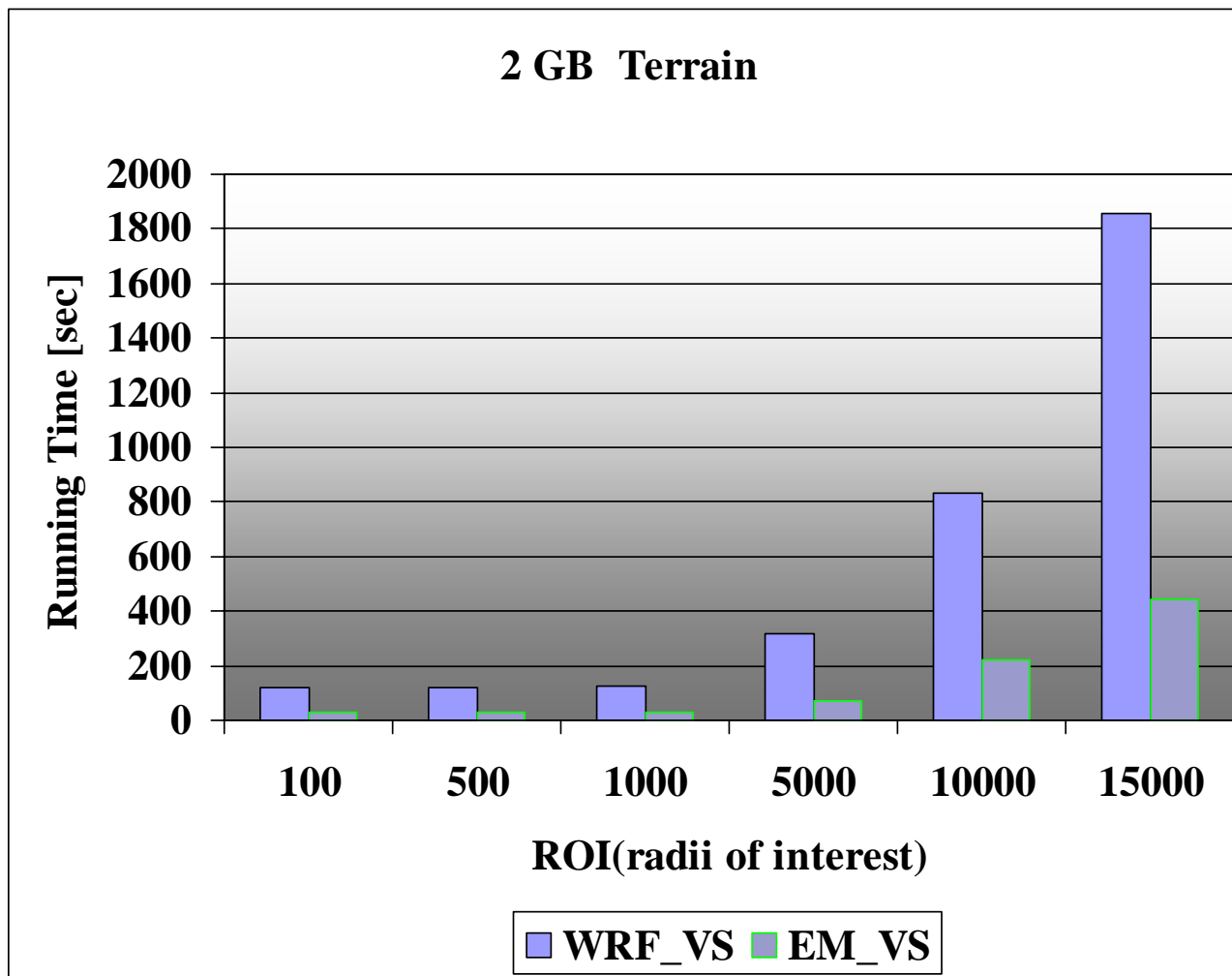


Results

- EM_VS is 3.5 faster than WRF_VS
- EM_VS is 3.5 faster than Haverkort et al method
 - Simpler and easier to implement



Results





References

- Aggarwal, A. and Vitter, J. S. (1988). The input/output complexity of sorting and related problems. *Communications of the ACM*, 9:1116–1127.
- Dementiev, R., Kettner, L., and Sanders, P. (2005). Stxxl : Standard template library for xxl data sets. Technical report, Fakultät für Informatik, Universität Karlsruhe. <http://stxxl.sourceforge.net/> (accessed on September 2007).
- Franklin, W. R. and Ray, C. (1994). Higher isn't necessarily better - visibility algorithms and experiments. In 6th Symposium on Spatial Data Handling, Edinburgh, Scotland.
- Haverkort, H., Toma, L., and Zhuang, Y. (2007). Computing visibility on terrains in external memory. In *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments / Workshop on Analytic Algorithms and Combinatorics (ALENEX/ANALCO)*.