Advanced Computer Graphics Project Report

Terrain Approximation

Wenli Li (liw9@rpi.edu)

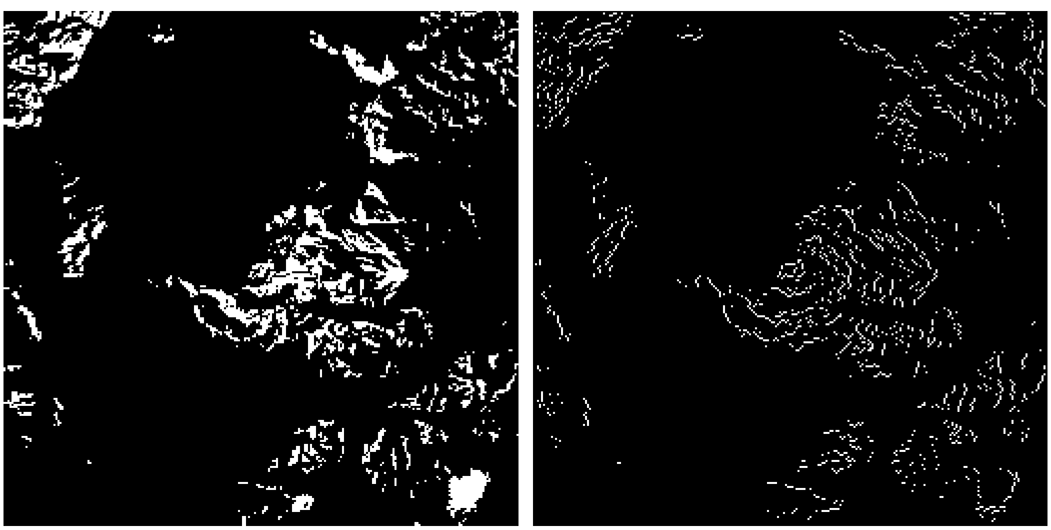
1. Introduction

DEM datasets are getting larger and larger with increasing precision, so that approximating DEM can be useful in some situations. Conventional terrain simplification algorithms try to minimize the elevation error between the simplified and the original terrains [3]. This project tries to minimize the visibility error in terrain approximation. The error metric is defined as the average viewshed error of a number of random viewpoints over the terrain. In order to minimize the error, the viewshed computed on the approximated terrain should be as close as possible to the viewshed computed on the original terrain. One possible way of keeping the viewshed is to keep the points defining the viewshed, which can be identified by computing viewsheds at all points of the terrain. This information can be used in triangulation to compute approximations of the terrain. The method is compared with the maximum elevation error based triangulation for the average viewshed error.

This project is inspired by the work of Ben-Moshe et al. [1], who studied visibility preserving simplification of TIN by computing the ridge network. They defined a visibility-based simplification measure, called the visibility similarity, which is the percentage of a predefined set of pairs of points that have the same visibility on both the simplified and the original terrains. The measure can be viewed as a generalization of the error metric of this project. In their algorithm, the ridge network is computed by analyzing the flow property of each edge. They acknowledged the close relationship between the ridge network and the drainage network, but they had not used the drainage network. Then the ridge network is approximated by collapsing edge chains, pruning short leaves, and adding back the farthest vertices. Finally, the terrain is approximated by a conventional method, using the ridge network as the constraint. Stuetzle et al. [2] studied hydrology preservation of DEM simplification. They computed the drainage network, and the ridge network as the drainage network of the inverted terrain, which they called the ridge-river network in together. Then they simplified the ridge-river network and reconstructed the terrain using ODETLAP. The inverse terrain is also used in this project.

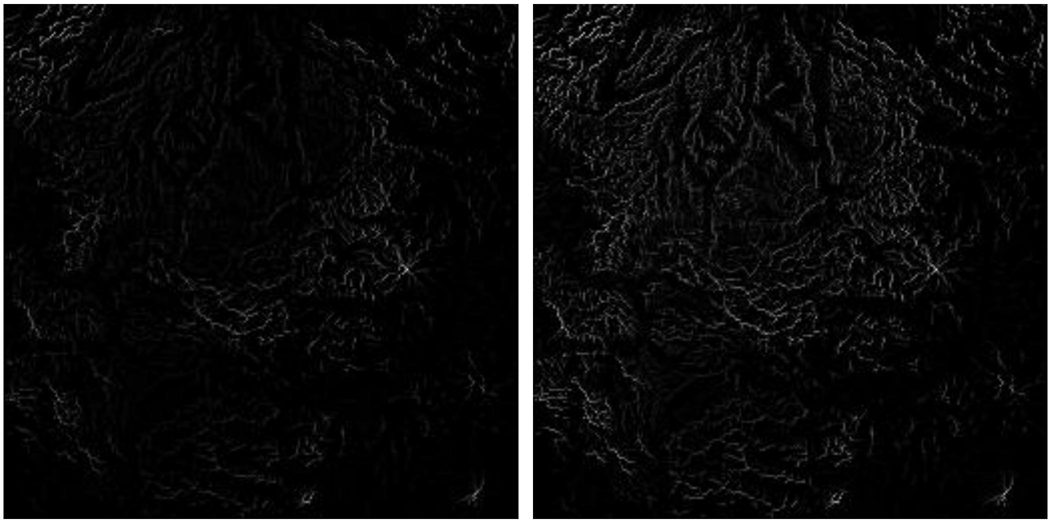
2. Blocking Index Map

There are two types of points along the boundaries of a viewshed. Along a ray emanating from the viewpoint in 2D, the viewshed either changes from visible to invisible through a boundary point (first type), or changes from invisible to visible through a boundary point (second type). A boundary point of the first type blocks the view of the terrain beyond the point along the ray until a boundary point of the second type. Call a boundary point of the first type a **blocking point** of the viewshed. A boundary point of the second type is the projection of the previous blocking point along the sightline onto the terrain. Therefore, although all points define the viewshed, blocking points could be the more important points because they and their projections are boundary points of the viewshed. If the elevations of the terrain are not changed much in an approximation, keeping the blocking points may also keep the boundaries of the viewshed. Figure 1(a) shows a viewshed where the viewpoint is located at the center of the test terrain. Figure 1(b) shows the blocking points of the viewshed. A point is even more important if it is a blocking point in a lot of viewsheds. To capture the notion, the **blocking index** of a point is defined as the number of times that the point is a blocking point of some viewshed. Assuming viewpoints can only be located over terrain points, to compute the blocking index of each point of the terrain, a viewshed is computed at each point of the terrain, and the blocking index of each blocking point of the viewshed is increased by 1. Call the blocking indices of all points of the terrain the **blocking index map** of the terrain. However, there is one problem of using the perspective viewshed to compute the blocking index map: the viewshed at a point is dependent on the height of the viewpoint. Different viewpoint heights produce different blocking index maps, for example, Figure 2(a) is the blocking index map of the test terrain with the viewpoint height = 0 and Figure 2(b) is the blocking index map with the viewpoint height = 1000. To compensate for this problem, this project defines and uses the **orthographic viewshed**, which is independent of the viewpoint height, to compute the blocking index map. Figure 3 shows how to compute the orthographic viewshed on a terrain profile. The 'viewpoint' is a vertical straight line and all the sightlines are parallel to the ground plane. A point of the terrain is visible if the sightline from the 'viewpoint' through the point is not blocked by other points of the terrain.



(a) (b)

Figure 1. (a) A viewshed and (b) its blocking points. The viewpoint is located at the center.



(a) (b)

Figure 2. (a) The blocking index map with the viewpoint height = 0. (b) The blocking index map with the viewpoint height = 1000.

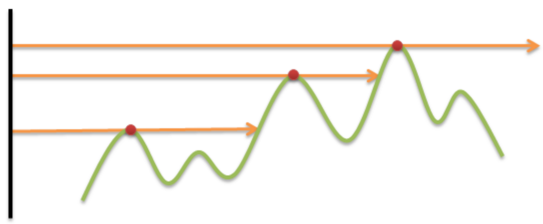


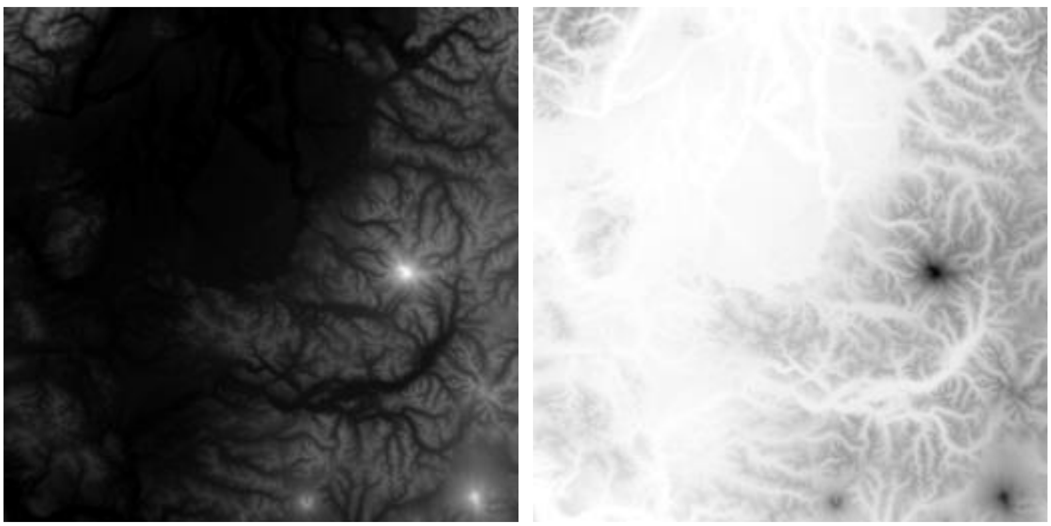
Figure 3. Compute the orthographic viewshed on a terrain profile. The black line is the 'viewpoint'; the orange rays are sightlines, which are parallel to the ground plane; the red points on the terrain profile are blocking points.

3. Triangulation

The terrain is approximated using triangulation. The basic method for comparison is to triangulate the terrain based on the maximum elevation error [3]. Given an initial triangulation of the terrain, for example, two triangles for a rectangle, the next point to be inserted has the maximum difference in elevation from the triangulation surface to the terrain. The process continues until the desired number of points has been inserted, and the underlying triangulation is a Delaunay triangulation. The approximation method of this project is to triangulate another map of the same size as the terrain, specified in the next chapter, using the conventional maximum elevation error based triangulation, and then assign the elevations of the original terrain to that of the vertices of the triangulation, to obtain a triangulation of the original terrain. To get an approximated terrain, the triangulation is projected onto a grid of the same size.

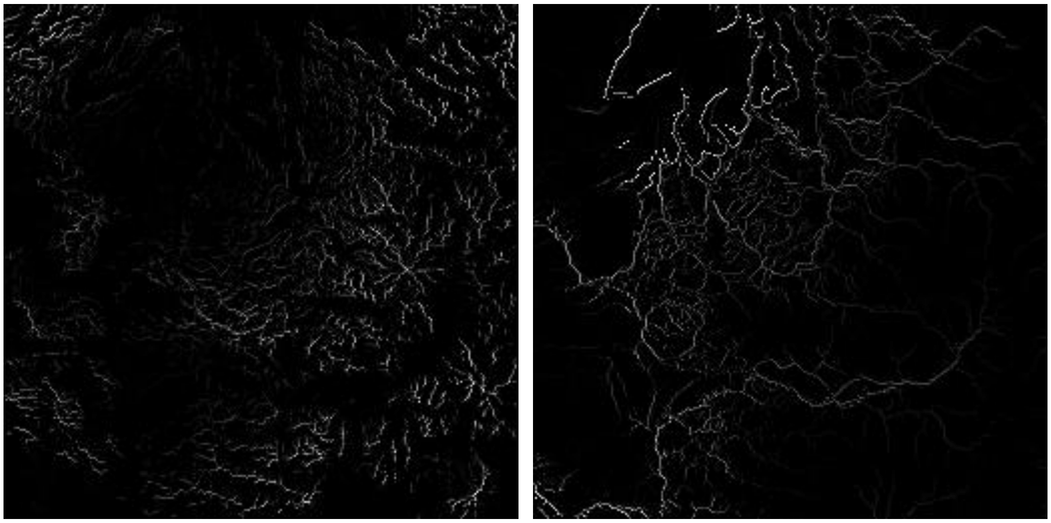
4. Results

The implementation of the algorithm is in C++ and the Delaunay triangulation is implemented using the 2D triangulations module of the CGAL-4.2 algorithms library [4]. The test terrain is a Puget Sound 257×257 DEM, which is a complex terrain with rivers, plains and mountains. Figure 4(a) shows the grayscale image of the terrain and Figure 4(b) shows the image of the inverse terrain, by negating the values at all points of the terrain, as suggested by Stuetzle et al. [2]. Then the blocking index map of the terrain and the blocking index map of the inverse terrain are computed, as shown in Figure 5(a) and Figure 5(b). The map in Figure 5(a) indicates ridges of the terrain, with brighter points representing higher or more important ridges, and the map in Figure 5(b) indicates the valleys with brighter points representing deeper or more important valleys. The subtraction of the two blocking index maps is computed, shown in Figure 6, to combine ridges and valleys in a single map.



(a) (b)

Figure 4. (a) Puget Sound 257×257 DEM. (b) The inverse terrain.



(a) (b)

Figure 5. (a) The blocking index map of the terrain. (b) The blocking index map of the inverse terrain.

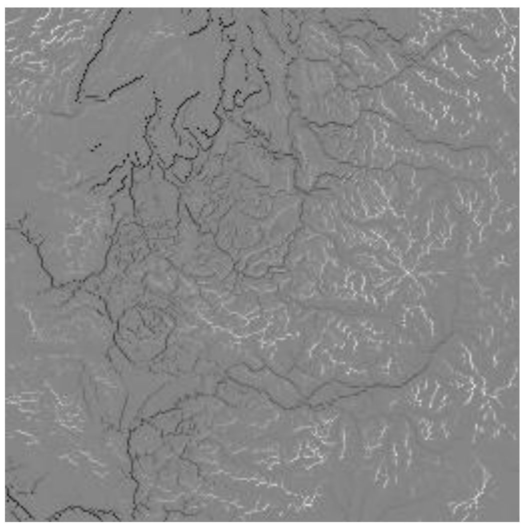


Figure 6. The subtraction of the blocking index map of the terrain and the blocking index map of the inverse terrain.

The following maps are computed for triangulation: DEM + k × BIM, the addition of the terrain with k times the blocking index map; DEM – k × $BIM_{inverse}$, the subtraction of the terrain with k times the blocking index map of the inverse terrain; DEM + k × $BIM_{subtraction}$, the addition of the terrain with k times the subtraction of the blocking index maps. These maps can be viewed as augmenting the original terrain with the blocking index maps, hence exaggerating its geomorphological features. Then the maps are triangulated and projected onto a 257×257 grid as approximations of the original terrain.

To compute the error of the triangulations of the maps in computing viewsheds, 100000 random viewpoints are generated over the terrain with height above the terrain in range [1, 431] and range of interest in range [1, 128]. For the projection of the triangulation of each map, the viewsheds of the viewpoints are computed and compared with the viewsheds on the original terrain. For each viewpoint, the total number of points within the range of interest is computed, as well as the number of points that the viewshed on the projection is different from the viewshed on the terrain. The average viewshed error is computed as the total number of different viewshed points for all viewpoints over the total number of points in range for all viewpoints.

Table 1 shows the average viewshed errors of triangulating and projecting the maps with different k's. All triangulations use 1% of the original terrain points. DEM is the original terrain whose triangulation is compared with. The value of k ranges from 0.1 to 0.6. For a wide range of k the triangulations of the maps obtain a smaller viewshed error than that of the original terrain. The smallest error, about 0.174, appears in k = 0.1 for DEM – k × $BIM_{inverse}$ and k = 0.2 for DEM + k × $BIM_{subtraction}$, which is a significant improvement over the base value, 0.263. However, the error gets worse with increasing k and is worse than the base value for large k's, for example, k = 0.4 for DEM + k × BIM and k >= 0.5 for DEM + k × $BIM_{subtraction}$. The error also gets worse with decreasing k's from somewhere around 0.1 (results not shown). Therefore, k = 0.1 seems to be the sweet spot for the particular terrain and particular level of triangulation. Overall, DEM + k × $BIM_{subtraction}$ is the best because it is mostly better than the other maps and never worse than the original terrain in each k.

Table 1. Average viewshed errors.

Triangulation (1%)	**Average Viewshed Error**					
DEM	0.263433					
	k = 0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**
DEM + k × BIM	0.210981	0.229637	0.230849	0.27303	0.245065	0.24368
DEM – k × $BIM_{inverse}$	0.17366	0.188704	0.238911	0.2578	0.270015	0.296286
DEM + k × $BIM_{subtraction}$	0.187634	0.173688	0.181667	0.181894	0.181032	0.190783

However, how closely do the triangulated terrains resemble the original terrain? Table 2 shows the average elevation errors of the various triangulations, while the elevation range of the terrain is [0, 43165]. The maximum elevation error based triangulation indeed achieves the minimum average elevation error, and the error gets worse and worse with the increasing portion of the blocking index map. However, when the average viewshed error is small and k is also small, the average elevation error is not too much worse, for example, k = 0.1 for DEM – k × $BIM_{inverse}$ and k = 0.1 or 0.2 for DEM + k × $BIM_{subtraction}$. Therefore, one can decide how much the viewshed error improves and how much the elevation error deteriorates in choosing k.

Table 2. Average elevation errors.

Triangulation (1%)	**Average Elevation Error**					
DEM	1069.08					
	k = 0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**
DEM + k × BIM	1225.11	1304.76	1342.39	1253.94	1527.22	1404.16
DEM – k × $BIM_{inverse}$	1158.59	1493.79	1793.36	1971	2057.01	2327.71
DEM + k × $BIM_{subtraction}$	1090.55	1179.82	1357	1491.65	1481.29	1469.74

5. Conclusions

This project has studied terrain approximation while retaining visibility properties, which are measured by the accuracy of viewshed computations. A point is defined as a blocking point if the viewshed changes from visible to invisible pass the point. The blocking index map is calculated by computing orthographic viewsheds at all points of the terrain and the blocking index of each point is the number of times that it is a blocking point of some viewshed. The blocking index map captures the ridges while the blocking index map of the inverse terrain captures the valleys. The subtraction of the blocking index maps captures both. To compute the approximations, the terrain is augmented with the blocking index maps, and then triangulated and projected onto the grid. Results show that the approximations can truly reduce the average viewshed error, although the average elevation error is increased.

There is a lot of future work to do. The first thing is to test the method on different and larger terrains to further verify its effectiveness and to find good values of k for general terrains or for particular types of terrains. Another thing is to better utilize the blocking index map, as adding it to the terrain is only one way of using it. Besides, instead of computing viewsheds at random viewpoints, the approximations can be used in other visibility related applications.

References:

[1] Boaz Ben-Moshe, Joseph S. B. Mitchell, Matthew J. Katz, and Yuval Nir. 2002. Visibility preserving terrain simplification: an experimental study. In *Proceedings of the eighteenth annual symposium on Computational geometry* (SCG '02). ACM, New York, NY, USA, 303-311.

[2] Christopher Stuetzle, W. Randolph Franklin, and Barbara Cutler. 2009. Evaluating hydrology preservation of simplified terrain representations. *SIGSPATIAL Special* 1, 1 (March 2009), 51-56.

[3] P. S. Heckbert and M. Garland. Fast polygonal approximation of terrains and height fields. Report CMU-CS-95-181, Carnegie Mellon University, 1995.

[4] Computational Geometry Algorithms Library. http://www.cgal.org/ (accessed 2 May 2013).