Slope Preserving Terrain Simplification — An Experimental Study

Ahmad Biniaz*

Abstract

This paper introduces a new measure of quality for terrain surface simplification aiming at preserving the slope of the surface, as well as a new simplification algorithm which satisfies this measure. Experimental results and comparisons with other simplification algorithms show the effectiveness of proposed algorithm according to some quality measures.

1 Introduction

In this paper we study one of the most important problems on terrains; simplifying the terrain surface. One common representation of terrain data in GIS (geographical information systems) is the TIN (triangulated irregular network). A TIN is obtained by triangulating a collection of irregularly spaced sample points (e.g., of a rectangular region R), and then giving each triangulation vertex the elevation (z-coordinate) of the corresponding sample point. In many applications the input terrain model is huge, and since the computational cost of using a model is directly related to its complexity, it is useful to have simpler versions of complex models. So, it is often necessary to decrease the input size prior to using it for analysis or visualization. Surface simplification has applications in GIS, cartography, computer vision, computer graphics and FEM (finite element methods) [8]. The excellent survey article by Heckbert and Garland [8] gives an overview of different kinds of terrain simplification algorithms. Garland and Heckbert [6] categorizes the surface simplification algorithms into three classes:

- *Vertex Decimation*: This method iteratively removes a vertex and its adjacent faces, and retriangulates the resulting hole [1], [2].
- *Vertex Clustering*: The original model is divided into a grid. Then, the vertices within each cell are clustered together into a single vertex, and the model faces are updated accordingly [9].
- *Edge Contraction*: This method iteratively selects an edge and then contracting it [6].

*Department of Computer Science, Azad University of Lamerd, biniaz@cse.shirazu.ac.ir

There are many possible ways to measure the degree of similarity between the original and simplified terrains; some are exact (e.g., specifying an exact numerical error tolerance ϵ such that the simplified terrain must lie within vertical distance ϵ of the original, at every point $(x, y) \in R$), while other methods rely on qualitative notions of similarity (e.g., based on human perception of similarity) [2]. Most papers dealing with terrain simplification consider the preservation of some quality measures such as *inter-point visibility*, *inter-point distance*, *area*, *ridge and drainage networks*. Ben-Moshe et al. [1], [2] suggested two quality measures based on preserving the inter-point visibility, and inter-point distance. Bose et al. [5] study the area-preserving simplification problem for x-monotone polygonal paths in the plane. Gudmundsson et al. [7] studied the distance-preserving of polygonal paths. In this paper we study a new measure of quality based on *slope preservation*. Informally, a simplification is considered "good" by slope measure if for most points p of output terrain, the slope at p on the simplified terrain is not significantly different from the corresponding slope on the original terrain. The importance of slopes for terrains was also observed in [11]. We introduce a new vertex decimation simplification algorithm aiming at preserving the slope of the terrain. It improves other quality measures such as inter-point visibility preservation and ideal measure.

The remaining of this paper organized as follows. In Section 2 we propose our simplification algorithm in more details. Section 3 gives the experimental results and comparison with other simplification algorithms. Finally, we conclude this paper in Section 4.

2 The Algorithm

Let the Delaunay triangulation T be the original terrain model of a 2D point set P of n points, and T' be a simplified terrain with m vertices, such that $n \geq m$. Let P2R $= \frac{m}{n} * 100$ (percent to remain) be a parameter that tells us what percent of the vertices of T should remain in the output. We also define the *simplification rate* (*reduction rate*) to be $\beta = \lfloor \frac{n}{m} \rfloor$; it means that for each set of β vertices from the input model, at most one vertex can remain in the output.

Our method for simplification of T is based on the following observations. Since the slope inconsistency is attributable to the presence of ridges and valleys, our heuristic is designed to place priority on preserving the most salient ridges and deep valleys. Little et al. [12] showed that including ridges and valleys initially reduces the error in the resulting approximated terrain. The algorithm consists of the following three stages:

1. First, we compute the ridge and drainage networks, which are collections of chains of edges of T.

2. Next, we approximate the ridge and drainage networks accurately. The resulting approximated networks subdivide the terrain into planar patches.

3. Finally, we simplify the patches separately by a new slope preserver heuristic.

2.1 Computing the networks

The drainage network is comprised of a collection of chains of valley edges of T [10]. The ridge network is closely related to the drainage network of reverse T. A *valley (edge) component* is a maximal set of valley edges such that flow from all of these valley edges reaches the lowest vertex (pit or local minima) incident to these valley edges. So a valley component is a tree rooted at a pit (see Figure 1-left). The drainage network in a TIN is a collection of these trees. So, it has the topology of a forest (union of trees). Biniaz et al. [3] computed the drainage network of T in $O(n)$. They find all the local minima (roots) of the terrain first, and then, explore the valley edges connected to local minima in *bfs* manner. The ridge network of T can be computed as the drainage network of invert(T).

2.2 Approximating the networks

Moshe et al. [1] presented a heuristic for approximating a valley component which uses two operations *collapse* and *refine*. Here, we introduce a more accurate heuristic to simplify each component:

1. break down the valley component (tree) into convex valley chains, and

2. replace each convex chain by a single segment.

We define a *convex valley chain* as a path of valley edges in such a way that the signed area of each of its three consequent vertices are identical; all positive or all negative (see Figure 1-right). We limit each convex valley chain C by $1 \leq \text{length}(C) < \beta$.

For a given convex valley chain C, we define its *turning angle* as follows: let C be defined by the edges $\vec{e}_1, \vec{e}_2, ..., \vec{e}_m$ $(1 < m < \beta)$. The turning angle of C is defined as $\alpha(C) = \sum_{k=1}^{m-1} \measuredangle(\vec{e}_k, \vec{e}_{k+1})$, where $\measuredangle(\vec{e}_k, \vec{e}_{k+1})$ is the turn from $\vec{e}_k$ to $\vec{e}_{k+1}$. If $m = 1$ then $\alpha(C) = 0$. In the case $\alpha(C) \leq \pi$, the turning angle of C can be computed in $O(1)$ as the angle between the first segment and the last one.

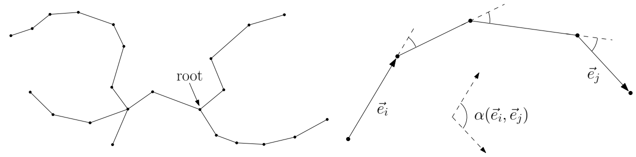

Figure 1: A valley component and a convex chain.

We decompose a valley component into convex chains in $O(s)$, by performing a preorder traversal, where s $(s \leq n)$ denotes the size of valley component. During the traversal, we stop the current chain C and start new chain(s) if one of the following conditions hold:

1. the turning direction change: this condition guarantees the convexity of C.

2. reach a vertex with degree greater than two: this guarantees that C be a path.

3. length$(C) \geq \beta$: it prevent the construction of long chains, and guarantees that the simplification be uniformly distributed along the network.

4. $\alpha(C) \geq \pi/2$: it prevent the construction of cyclic components (described below).

These conditions entail that the structure of approximated network is not significantly different from the original network. Figure 2-left shows the decomposition of the network with $\beta = 4$. The chain x terminated and a new chain y is started because the counterclockwise turn changed to clockwise turn (cond. 1). At the points with degree greater than 2, new chains are started (cond. 2). The chain z terminated, because it length reaches the maximum value 4 (cond. 3). The chain w terminated, because addition of another edge leads to $\alpha(w)$ becomes greater than $\pi/2$ (cond. 4). To simplify each multi-edge chain C, we use the collapse operation which replace C with a single edge (Figure 2-right).

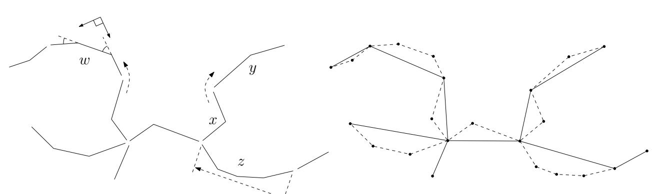

Figure 2: Network simplification.

A network may contains cycle form components. Cycles in networks of valleys are not just same as the cycles in graphs. We call a valley chain C, *cyclic*, if $\alpha(C) \geq 2\pi$. Yu et al. [10] give examples of cyclic components. Figure 3-left shows a cyclic component. In high simplification rates, the VPTS algorithm [1] approximates it as a single edge (Figure 3-middle), but our method approximates it as Figure 3-right, because of cond. 4.

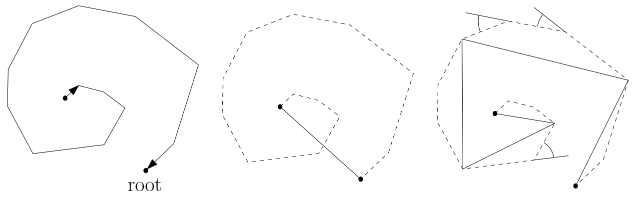

Figure 3: Cyclic component simplification.

If we desired a more simplified network, it takes in two ways: (i) Fine tuning of the parameters α and β; increase in α and β leads to more simplified network. (ii) Perform the simplification process iteratively (i.e., apply the above method on the simplified network again).

The resulting approximated networks subdivide the terrain into patches (flat regions). The patches take place between ridges and valleys.

2.3 Approximating the patches

We simplify the patches separately by a new vertex-decimation slope preserver heuristic. The main algorithm is done by first adding the edges of the approximated networks as constraints to the Delaunay triangulation that is being computed, and then apply the slope preserver heuristic on it.

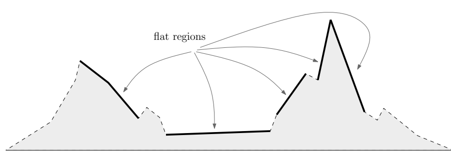

Figure 4: Some flat regions.

The patches are locally flat (planar) regions, witch are more slope consistent parts of the terrain (see Figure 4). The slope preserver heuristic iteratively removes the vertices of maximum smoothness from the patches and update the triangulation. These vertices do not influence the slope consistency, and removing them don't affect the slope considerably. Biniaz et al. [4] showed that the slope consistency in a terrain is directly related to its dihedral angles (the angle between two contiguous triangles which is less than or equal to π); as the dihedral angles become wider and approach π, the surface becomes more plain and consistent. Thus, a locally flat region consists of vertices of wide dihedral angles. We define the *smoothness* (*flatness*) of a vertex v as mean of all dihedral angles incident to it, which is a value between 0 and π:

$$S_T(v) = \frac{\sum_{u\in N[v]} \mathrm{da}(v,u)}{|N[v]|} \qquad (1)$$

where, $N[v]$ denotes the set of neighbors of v in T, and $\mathrm{da}(v,u)$ is the dihedral angle of the edge (v,u). The smoothness of a vertex v is computed right at the beginning, and is updated when ever v is affected by the deletion of another vertex. If v has plain dihedral angles, then it belongs to a flat part of T and must be omitted.

3 Experimental Results and Comparisons

This section gives some results of our experiments with SPTS, as well as comparisons with two other software packages: QSlim [6] and VPTS [1]. Our software package, SPTS (Slope Preserving Terrain Simplification), was developed in C++, and use CGAL-3.3.1 library. The input terrains were chosen from four different geographic regions: California Hot Springs, Quinn Peak, Sphinx Lakes, and Split Mountains. The terrains have roughly 20,000 vertices. Our tests and comparisons are based on three measure of quality: slope preservation, inter-point visibility preservation, and ideal measure.

3.1 Experiments of Slope Measure

For each terrain T, four simplifications P2R = 20%, 40%, 60%, and 80% were computed. We compute the quality of simplification $T^{'}$ of T as follows. For each vertex $v \in T^{'}$ we characterize the error at v as the ratio $\delta_v = \frac{|S_T(v)-S_{T'}(v)|}{\pi}$, and the error of simplification $T^{'}$ is computed as the average over all the vertex errors, $\Delta_{T'} = \frac{\sum_{v\in V(T')}\delta_v}{|V(T')|}$ which is a value between 0 and 1.

Table 1 shows the error $\Delta_{T'}$ of simplified terrains. It is clear that, the error decreases as P2R increases. In all cases, SPTS is significantly better than VPTS, and VPTS is slightly better than QSlim. Table 2 shows the average simplification size over four input terrains that is needed to achieve a given threshold error. For example to achieve the threshold error of 0.005, the minimum average simplification size that is needed by SPTS is 78%, while it is 92% for VPTS, and 96% for QSlim.

Table 1: Slope measure, simplification errors.

	California Hot Springs			Quinn Peak		
	QSlim	VPTS	SPTS	QSlim	VPTS	SPTS
20%	0.201	0.143	0.092	0.196	0.145	0.088
40%	0.145	0.101	0.051	0.142	0.099	0.049
60%	0.102	0.070	0.008	0.113	0.068	0.008
80%	0.074	0.048	0.003	0.080	0.038	0.003
	Sphinx Lakes			Split Mountains		
	QSlim	VPTS	SPTS	QSlim	VPTS	SPTS
20%	0.220	0.135	0.088	0.211	0.147	0.090
40%	0.161	0.098	0.047	0.153	0.098	0.050
60%	0.122	0.068	0.008	0.112	0.067	0.009
80%	0.091	0.039	0.004	0.088	0.039	0.005

Table 3: Visibility measure, average of σ_χ.

P2R sample size	3%			5%			10%		
	QSlim	VPTS	SPTS	QSlim	VPTS	SPTS	QSlim	VPTS	SPTS
30	0.887	0.897	0.929	0.937	0.948	0.964	0.954	0.956	0.984
50	0.895	0.912	0.935	0.943	0.941	0.966	0.960	0.967	0.985
100	0.890	0.907	0.937	0.953	0.956	0.970	0.969	0.976	0.989

Table 2: Average simplification size.

threshold	0.001	0.005	0.010	0.050	0.100	0.500
QSlim	99%	96%	93%	90%	67%	13%
VPTS	97%	92%	88%	76%	39%	10%
SPTS	89%	78%	57%	39%	17%	4%

3.2 Experiments of Visibility Measure

A simplification is considered "good" by visibility measure if for any set χ of pairs of points from the underlying rectangle R, most of the visibility information is preserved, i.e. for most pairs $\{p, q\} \in \chi$, if the points on T corresponding to p and q are visible (resp. not visible) to each other, then the corresponding points in T' should also be visible (resp. not visible) to each other [1]. Let ν be the set of pairs $\{p, q\} \in \chi$ for which visibility is the same. The *visibility similarity*, $\sigma_\chi(T, T')$, is defined as $\sigma_\chi = \frac{|\nu|}{|\chi|}$.

We compare SPTS with QSlim and VPTS for visibility measure. We pick random samples of size 30, 50 and 100 points from the underlying rectangle R. For each pairs $\{p, q\} \in \chi$ in each sample, we determined whether p and q see each other in T and T', or not. Table 3 shows the average values of σ_χ obtained from four terrains, for three simplifications. SPTS improves the visibility similarity, because of the fact that the blocking of the view from a point is attributable to the presence of ridges, and SPTS approximates the ridges by more accurate conditions than VPTS.

3.3 Experiments of Ideal Measure

We define the *ideal measure* as the volume enclosed between the surface of input terrain and the surface of simplified terrain. We estimate it as the average vertical distance (i.e., along the z-axis) between a random sample of points on the input terrain and the corresponding points in the surface of simplification (see Table 4). SPTS improves the ideal measure, because of deleting the vertices from flat regions makes smaller volume between the surfaces.

Table 4: Ideal measure.

P2R	20%	40%	60%	80%
QSlim	0.100	0.050	0.014	0.007
VPTS	0.102	0.063	0.019	0.010
SPTS	0.070	0.026	0.007	0.002

4 Conclusion

We proposed a new quality measure for terrain simplification based on preserving slope of the surface, as well as a new simplification algorithm. The algorithm approximates the ridge and drainage networks accurately, and then simplify the patches which formed between them by a new heuristic. It improves other quality measures such as inter-point visibility and ideal measure.

References

[1] B. Ben-Moshe, M. J. Katz, S. B. Mitchell and Y. Nir. Visibility preserving terrain simplification an experimental study. *Comput. Geom. Theory Appl.*, 28:175–190, 2004.

[2] B. Ben-Moshe, M. J. Katz and I. Zaslavsky. Distance preserving terrain simplification an experimental study. In *Proc. CCCG'07*, pages 129–132, 2007.

[3] A. Biniaz and G. Dastghaybifard. Drainage reality in terrains with higher order Delaunay triangulations. In *Adv. in 3D Geoinf. Sys.*, Springer, pages 199–213, 2008.

[4] A. Biniaz and G. Dastghaybifard. Slope fidelity in terrains with higher order Delaunay triangulations. In *Proc. WSCG'08*, pages 17–23, 2008.

[5] P. Bose, S. Cabello, O. Cheong, J. Gudmundsson, M. van Kreveld and B. Speckmann. Area-preserving approximations of polygonal paths. *J. of Discrete Alg.*, 4:554–566, 2006.

[6] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proc. SIGGRAPH97*, pages 209–216, 1997.

[7] J. Gudmundsson, G. Narasimhan and M. Smid. Distance-preserving approximations of polygonal paths. *Comput. Geom. Theory Appl.*, 36:183-196, 2007.

[8] P. S. Heckbert and M. Garland. Survey of polygonal surface simplification algorithms. Carnegie Mellon University, Manuscript, 1997.

[9] J. Rossignac and P. Borrel. Multi-resolution 3d approximations for rendering complex scenes. In *Modeling in Comp. Graphics Methods Appl.*, pages 455-465, 1993.

[10] S. Yu, M. van Kreveld and J. Snoeyink. Drainage queries in tins: from local to global and back again. In *Advances in GIS research II*, pages 829–842, 1997.

[11] C. M. Gold and M. Dakowicz. Terrain modelling based on contours and slopes. In *Advances in Spatial Data Handling*, Springer-Verlag Berlin, pages 95-108, 2002.

[12] J. J. Little and P. Shi. Structural lines, TINs and DEMs. *Algorithmica*, 2001.