Visibility Preserving Terrain Simplification — An Experimental Study *

Boaz Ben-Moshe†
Matthew J. Katz†
Joseph S. B. Mitchell‡
Yuval Nir†

ABSTRACT

The terrain surface simplification problem has been studied extensively, as it has important applications in geographic information systems and computer graphics. The goal is to obtain a new surface that is combinatorially as simple as possible, while maintaining a prescribed degree of similarity with the original input surface. Generally, the approximation error is measured with respect to distance (e.g., Hausdorff) from the original or with respect to visual similarity. In this paper, we propose a new method of simplifying terrain surfaces, designed specifically to maximize a new measure of quality based on preserving inter-point visibility relationships. Our work is motivated by various problems of terrain analysis that rely on inter-point visibility relationships, such as optimal antenna placement.

We have implemented our new method and give experimental evidence of its effectiveness in simplifying terrains according to our quality measure. We experimentally compare its performance with that of other leading simplification methods.

Categories and Subject Descriptors

I.3 [**Computing Methodologies**]: Computer Graphics; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling

*Research by the first three authors is partially supported by grant no. 2000160 from the U.S.-Israel Binational Science Foundation. Research by the Israeli authors is also partially supported by the MAGNET program of the Israel Ministry of Industry and Trade (LSRT consortium). Research by Mitchell is also partially supported by HRL Laboratories, NASA Ames Research (NAG2-1325), National Science Foundation, Northrop-Grumman, Sandia National Labs, and Sun Microsystems.

†{benmoshe,matya,yoval}@cs.bgu.ac.il, Computer Science, Ben-Gurion University, Beer-Sheva 84105, Israel.

‡jsbm@ams.sunysb.edu, Applied Math & Statistics, SUNY, Stony Brook, NY 11794.

SoCG'02, June 5-7, 2002, Barcelona, Spain.

General Terms

Algorithms,Experimentation

Keywords

visibility, surface simplification, terrain modeling, geographic information systems, ridge networks

1. INTRODUCTION

There are numerous papers dealing with terrain and surface simplification. A terrain can be modeled as a triangulation (e.g., of a rectangular region, R), with a height (z-coordinate) assigned to each triangle vertex. Terrain models are commonly used to represent the surface of the earth.

Because terrain models can be huge, particularly if they have very high resolution, it is often necessary to simplify them prior to using them for analysis or visualization. Methods for terrain simplification have been devised that transform a detailed terrain into a less detailed terrain, having fewer triangles, in such a way that the simplified terrain is "similar" to the original terrain in some sense. 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)), while others rely on qualitative notions of similarity (e.g., based on human perception of similarity).

In this paper we propose a new way to measure quality of simplification that is especially suitable for applications that compute and use visibility information. Informally, a simplification (of the desired size) is considered "good" by this measure if for any set $\mathcal{X}$ of pairs of points from the underlying rectangular region R most of the visibility information is preserved with respect to $\mathcal{X}$, in that for most pairs $\{p, q\} \in \mathcal{X}$, if the points on the original surface corresponding to p and q are visible to each other (resp., not visible to each other), then the corresponding points in the simplified terrain should also be visible to each other (resp., not visible to each other). Here, $\mathcal{X}$ is assumed to be a "reasonable" subset of some large pre-specified set $\mathcal{Z}$ of pairs of points in R.

This criterion is quite different from the commonly used criteria, since, for example, we do not care if a very high and detailed mountain is replaced by a much lower and less detailed mountain, as long as this change is not expected to significantly affect visibility for a typical set $\mathcal{X}$ of pairs of points. Similarly, using this criterion, one can often replace a detailed valley covering a large area by a much less detailed valley, since in both terrains most pairs of points that lie in the valley see each other, while visibility between other pairs of points is usually not affected too much by this change. Notice that according to other criteria, e.g., the maximum vertical distance between the terrains, the simplified terrain in the above two examples can have a very large deviation from the original.

The motivation for this new quality of simplification measure comes, e.g., from the task of finding good locations on the surface of a terrain for the placement of antennas that require line-of-sight visibility to their reception points. Given a set of reception points on a terrain T, we wish to place a small number of antennas, such that each reception point can receive at least one of the antennas, where a reception point can receive an antenna if either the horizontal distance between it and the antenna does not exceed some small value r_1, or there is line-of-sight between the reception point and the antenna, and the horizontal distance between them does not exceed some larger value r_2. (Actually, the case in which there is no line-of-sight is more complicated and requires the analysis of the terrain profile along the vertical plane passing through the antenna and the reception point; in our treatment, we have simplified this case.)

In practice, the locations for the placement of antennas are selected from a very large collection of potential locations by generating and checking a long sequence of relatively small candidate subsets of potential locations; each candidate subset is tested to see if it gives the desired coverage (each reception point can receive from an antenna placed at one of the points in the subset). This process is very slow since it requires a huge number of visibility tests, as each subset is considered. We propose to speed-up this process by first computing a simplified terrain T' that approximately preserves visibility for a relevant class of pairs of points that are neither too close nor too far from each other. When checking a subset, we first check it using T', and only if the result seems promising do we need to continue and check it using the original terrain, T.

In Section 2 we formally define our visibility-based measure of quality of simplification. We obtain two concrete measures by specifying two possibilities for the large set $\mathcal{Z}$ of pairs of points (from which $\mathcal{X}$ will be selected). The second measure is especially suitable for our antenna location application.

Then, in Section 3, we develop a simplification method that experimentally produces good terrains by the second measure, making it useful for the antenna location application. This method is based on the assumption that the ridges in the original terrain T are the most salient feature in our context, and, thus, they should be approximated especially well, preserving in our simplified terrain a large portion of the original vertices of T that lie along the ridges. In this method we first construct a drainage network, from which the ridge network is constructed. Next we approximate the ridges using a relatively high level of detail. The ridges partition the underlying rectangle R into regions (terrain patches), and, in the final stage, we approximate each of these patches separately using one of the standard terrain simplification methods. The implementation of the method (called VPTS) has been developed using the CGAL [2] library. A few examples of terrains produced by VPTS are shown together with the corresponding original terrains.

In Section 5 we report on some of our experiments with VPTS, as well as with three other software packages implementing simplification methods that were developed for the standard maximum vertical difference measure. These other packages are Terra ([10]), GcTin ([15, 14]), and QSlim[1] ([9]). Our choice of packages is based on what has been readily available for download; we plan to extend our experiments to include other packages as well.

In our experiments we consider several different input scenes, each a detailed terrain representing a geographic region. For each of the scenes T, we apply our new method, together with the three other methods for comparison, with various degrees of simplification specified by the number of allowed vertices in the simplified output terrain. We compare the output terrains according to our visibility-based quality measures, each with several sets of pairs of points $\mathcal{X}$. Our findings suggest that our new simplification method is better suited for the antenna location application than the other methods. That is, using the second visibility-based measure (see above), our method gives better results than the other methods for small terrain sizes. This of course is not surprising since the other methods were designed for a different measure.

Related work

There has been extensive work on many aspects of terrain approximation; see Heckbert and Garland [10] for a survey. Visibility on terrains has also been studied extensively. Cohen-Or and Shaked [4] give a simple linear-time method to compute visibility from a point on a digital elevation map (DEM). De Floriani and Magillo [5] give methods to compute, in $O(n \log n)$ time, the horizon map from a point on a terrain of complexity $O(n)$; the map can be updated efficiently using varying levels of detail. The use of ridges in horizon mapping has been proposed by Max [12] to speed casting of shadows in bump mapped surfaces. Stewart [16] shows how horizons can be computed efficiently from every point of a digital elevation map; his motivations include the placing of radio transmission towers. Little and Shi [11] (extending earlier work of Fowler and Little [8]), use linear features, along ridges and channels, to guide their triangulation algorithm with the goal to minimize root-mean-square (RMS) error in approximating a DEM with a TIN. We too use linear features (the ridge network) to guide our approximation, as we discuss in Section 3.

2. A VISIBILITY-BASED QUALITY OF SIMPLIFICATION MEASURE

Let T (resp., T') be a terrain model consisting of n (resp., m) triangles, with $n > m$. We assume that T and T' are defined over a common underlying rectangular region, R, in the (x, y)-plane. For a point $p \in R$, let p_T (resp., $p_{T'}$) denote the point in $\mathbb{R}^3$ that is obtained by lifting p onto the surface of T (resp., T'). We will say that $p \in R$ and $q \in R$ are *visible* with respect to T if the segment $p_T q_T$ lies entirely on or above the terrain surface T. Let $\mathcal{X}$ be a finite set of pairs of points in R. (One can think of $\mathcal{X}$ as the set of edges in a graph defined on a discrete set of points of R.) Let $\mathcal{V}$ be the set of pairs $\{p, q\} \in \mathcal{X}$ for which visibility is the same with respect to both T and T'; i.e., either p and q are visible

[1]Note that QSlim is a more general package for surface simplification, not just terrain simplification.

with respect to both T and T' or they are both not visible with respect to T and T'.

We define the *visibility similarity*, $\sigma_{\mathcal{X}}(T, T')$, between terrains T and T', with respect to $\mathcal{X}$ to be the ratio

$$\sigma_{\mathcal{X}} = \frac{|\mathcal{V}|}{|\mathcal{X}|}.$$

We often consider T to be the original terrain and T' to be an approximation thereof, in which case we refer to $\sigma_{\mathcal{X}}(T, T')$ as the (visibility-based) quality of T' with respect to $\mathcal{X}$.

Our goal is to develop a simplification method that produces good quality simplifications for small values of m, e.g., on the order of $0.05n$, with respect to any "reasonable" set $\mathcal{X}$ of pairs of points in R. Here, we consider $\mathcal{X}$ to be a reasonable set if it is a subset of some pre-specified large (not necessarily finite) set $\mathcal{Z}$ of pairs of points in R, which is either generated randomly or is a typical subset of point pairs arising in the underlying application. Some natural choices of $\mathcal{Z}$ would include (i) the set of all pairs of points in R, or (ii) the set of all pairs of grid points, for some regular grid on R, or (iii) the set of all pairs of (projections onto the (x, y)-plane of) vertices of T. In Section 5 we report on experiments performed for choice (i), with the sets $\mathcal{X}$ defined to be all pairs of points in a random subset of R. We refer to the quality measure obtained for such sets $\mathcal{X}$ as the *ideal* measure.

Despite its name, the *ideal* measure is not always ideal. For example, in many applications we are not interested in pairs of points that are very far from each other. Thus, sets $\mathcal{X}$ chosen to correspond to the edges of a complete graph on a randomly generated set of points may not be suitable for these applications, since it is likely that $\mathcal{X}$ will contain distant pairs. Also, distant pairs of points are usually not visible with respect to T, or with respect to simplifications of T, so they tend not to provide much information about the quality of a simplification.

An alternative choice of $\mathcal{Z}$ leads to a quality measure that is better suited for the antenna location application. Let $B \subseteq R$ be the set of (projections on the (x, y)-plane of) the reception points, and let $A \subseteq R$ be the set of (projections on the (x, y)-plane of) potential antenna locations. Recall that an antenna located at some point on the surface of T is received at some other point on the surface of T if and only if either the horizontal distance between the points does not exceed some value r_1, or there is a line-of-sight between the points and the horizontal distance between them does not exceed some $r_2 > r_1$. Thus, we choose

$$\mathcal{Z} = \{(a, b) | a \in A, b \in B, r_1 \leq d(a, b) \leq r_2\}.$$

That is, for each potential antenna location a, we are only interested in the reception points that lie in the annulus of radii r_1 and r_2 centered at a. Most of our experiments were performed for this choice of $\mathcal{Z}$, with the sets $\mathcal{X}$ defined accordingly (see Section 5). We refer to the quality measure obtained for such sets $\mathcal{X}$ as the *transmitter-receiver* measure.

3. A VISIBILITY-PRESERVING TERRAIN SIMPLIFICATION ALGORITHM

Our method for visibility-preserving simplification of a terrain model T is based on the observation that, typically, the blocking of the view from a point is attributable to the presence of ridges. Thus, our heuristic is designed to place priority on preserving the most salient ridges.

Our method consists of the following three stages.

(1) First, we compute the ridge network, which is comprised of a collection of chains of edges of T (see Figure 7). We do this by first computing the drainage information that is needed to define the ridge network.

(2) Next, we approximate the ridge network by keeping only a small subset of its vertices. This is done by a three-phase process described below. The resulting approximated ridge network induces a subdivision of the terrain into patches.

(3) Finally, we simplify each patch separately by applying one of the standard terrain simplification methods to it. This is done by adding the edges forming the approximated ridge network as constraints on the triangulation that is being computed.

3.1 Computing the ridge network

Since the ridge network is closely related to the drainage network, we first describe how we compute the drainage information, beginning with a brief overview of some terminology (see, e.g., [17]) and some of our assumptions in modeling drainage of water on terrain surfaces.

The faces, edges, and vertices of the terrain are *drain objects*, each of which can receive flow from a drain object and/or send flow to another drain object.

The water at any point on a terrain surface flows in the direction of steepest descent. This direction is, in general, unique. It is the same for all points in the interior of a face (i.e., a triangle), or in the interior of an edge. An edge receives flow from an adjacent face if the direction of steepest decent of the face crosses the edge from the interior (of the face) to its exterior. An edge sends flow to an adjacent face if the direction of steepest descent of the face crosses the edge from the exterior of the face to its interior. Water at a vertex flows either to an adjacent edge or to an adjacent face, depending on the direction of steepest decent.

An edge is *difluent* (or a *local ridge*) if it sends flow to both faces adjacent to it; it is *confluent* (or a *local channel*) if it receives flow from both faces adjacent to it. An edge is *transfluent* otherwise. See Figure 1. Faces flow to transfluent edges or to channel edges, transfluent edges flow to faces, channel edges flow to vertices, and vertices flow either to edges or to faces. Difluent edges cannot receive flow. Transfluent edges send flow to the adjacent face that receives flow. Confluent edges send flow to the lower of the two endpoint vertices.

We must be careful when computing the direction of flow for a vertex v, since it is not enough to consider the slopes of the appropriate adjacent edges and the steepest slopes of the appropriate adjacent faces, since it is not always possible to move from v in the direction of steepest slope of an adjacent face.

Faces (i.e., triangles) send flow to either one or two of their boundary edges. When a face sends flow to two of its edges, we refer to the face as a flow-splitting face; however, we select only one of the edges to receive the flow. (This simplification is done in order to avoid having to do a potentially extensive decomposition of the triangulation.) In particular, we select the receiving edge to be the one bounding the larger (in area) of the two triangles that result from splitting the face by a segment parallel to the steepest descent direction, through the common vertex shared by the two receiving edges. See Figure 2.

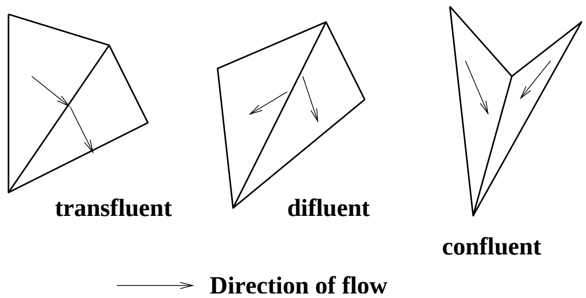

Figure 1: Classification of terrain edges according to fluency.

We also assume (as in [17]) that watercourses can merge, and watercourses end at one of possibly several local minima of the terrain. By our assumptions, the drainage network has the topology of a forest (union of trees).

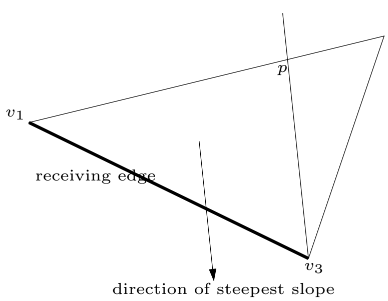

Figure 2: Selecting the (one) receiving edge.

The ridge network is defined to be a graph. Its node set consists of (i) the local ridges (difluent edges) and (ii) the receiving edge corresponding to each flow-splitting face (which can be thought of as containing an "implicit" local ridge). There is an arc of the ridge network joining two nodes if those nodes are adjacent (i.e., the corresponding edges share a vertex v), and water cannot flow from one side of the two-edge chain obtained by linking the two nodes to the other. For example, if both nodes correspond to difluent edges that share a vertex v, then, we add an arc between the corresponding nodes if and only if there is no flow of water through v from one side of the two-edge chain to the other. Refer to Figure 3.

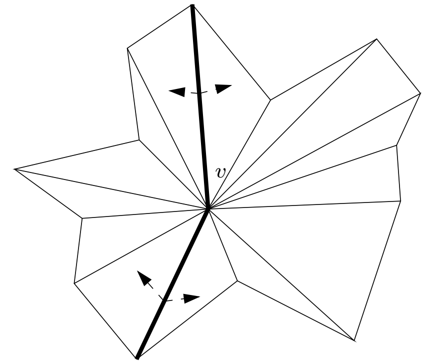

Figure 3: In order to determine if the two highlighted ridges should be joined by an arc in the ridge network we examine the neighbors of vertex v to see if there is flow from one side to the other of the two-ridge chain.

3.2 Approximating the ridge network

The ridge network corresponds to a collection of chains of edges of the input terrain T. (For example, the ridge network presented in Figure 4 (left) consists of six chains, assuming there is an arc of the ridge network whenever two edges share a vertex.) Unless we are interested in a simplification whose size (in terms of the number of triangle vertices) is comparable to the size of the input terrain, we cannot utilize the ridge network in the terrain simplification without approximating it first. Let k be the *budget* number of vertices that we permit in an approximate ridge network. (Experiments suggest that $k \cong m/4$ is usually a good choice, where m is the desired number of vertices in the simplified terrain.) Our goal is to replace the original ridge network with an approximate network of size k.

The ridge network may contain cycles (although usually it does not). If so, we break each cycle by removing one of the two edges adjacent to the lowest vertex along the cycle.

We first analyze the ridge network in order to assign a (heuristic) level of importance to the different chains. For example, a long chain that separates the underlying region into two large (in area) valleys is more important to retain in a visibility-preserving simplification than a chain consisting of only a few short edges. In the current implementation, the level of importance of a chain is determined by the number of edges in the chain and their total length.

We shall use the following two basic operations applied to a chain c: *collapse*(c) and *refine*(c). The operation *collapse*(c) replaces a multi-edge chain c with the single-edge chain c' that is defined by the two endpoints of c. The operation *refine*(c'), where c' is a chain approximating some original chain c, adds back to c' one of the vertices of c that is not present in c'; the vertex that is added is the "most needed" one, according to a heuristic test.

Assume we wish to approximate the ridge network using only k of its vertices. The approximation process consists of three phases.

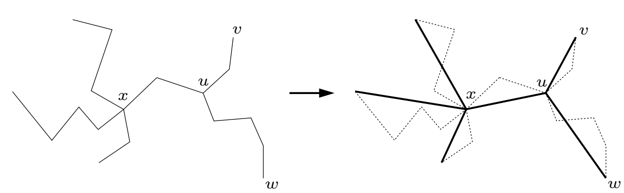

Figure 4: Phase I: Collapsing all chains. (We assume that there is an arc of the ridge network whenever two edges share a vertex.)

Phase I. In the first phase we apply the collapse operation to each of the chains of the ridge network. With each collapsed chain we keep the corresponding original chain, which also determines the importance of the collapsed chain. See Figure 4.

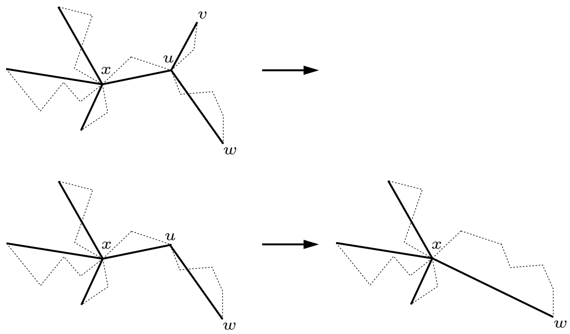

Figure 5: Phase II: First the leaf edge uv is dropped, next the two-edge chain xuw is collapsed.

Phase II. In the second phase we remove vertices until we are left with only k' vertices, where $k' < k$ is a parameter. In this phase we repeatedly drop a leaf edge of minimum importance from the current network. A leaf edge is an edge with at least one endpoint of degree one. Thus, dropping a leaf edge decreases the number of vertices by either one or two. After deleting a leaf edge uv, the current network may contain a chain of length two; see Figure 5. In this case we collapse the two-edge chain, storing with the collapsed chain the chain that is obtained by concatenating the chains that were stored with each of the two edges. The importance value associated with the collapsed chain is the sum of the levels of importance of the two edges from which it was obtained. We continue deleting leaf edges (and collapsing two-edge chains, if applicable) until we are left with only k' vertices.

Phase III. At this point, all chains of the current network are one-edge chains. In this third phase we add missing vertices to chains of the current network until the desired size k is reached. The vertices are added by repeatedly applying the refine operation. In more detail, for each chain c of the current network we compute a score reflecting the quality of c as an approximation of the full chain that is represented by c. This score is simply the distance between c and the vertex of the corresponding full chain that is farthest from c. (Thus, the smaller the score is, the better.) Now, while the number of vertices is less than k, we repeatedly pick a chain with highest score and refine it, adding to it the missing vertex that is farthest from it and then updating its score.

3.3 Approximating the terrain

The approximated ridge network induces a subdivision of the terrain T into patches. In the final stage we simplify each patch separately by applying one of the standard terrain simplification methods to it. In the current implementation of VPTS this is done by first adding the edges forming the approximated ridge network as constraints on the triangulation that is being computed, and then applying to T a variant of Terra [10] that we developed to be able to handle constraint edges. The number of vertices added in this stage should not exceed $m - k$, where m is the desired simplification size, and k is the number of vertices that were used to approximate the ridge network. As mentioned, we found that, in practice, $k \cong m/4$ is a good choice.

4. GALLERY OF IMAGES

Figure 7 illustrates the ridge network superimposed on a terrain dataset and on a simplification of it obtained by VPTS. Figure 8 shows a comparison of VPTS with Terra, selecting ~ 1000 vertices in the approximations. Notice the difference in the distribution of the vertices due to the emphasis put by VPTS on ridges.

5. EXPERIMENTAL RESULTS

In this section we report on some of our experiments with VPTS, as well as comparisons with three other software packages (Terra [10], GcTin [15, 14], and QSlim [9]) that implement simplification methods developed for the standard maximum vertical difference measure.

5.1 Methods for comparison

Terra. This algorithm, implemented by Garland [10], is based on a simple greedy insertion algorithm with some optimizations to make it run faster. The input is assumed to be a height field given by a regular grid of elevation data. It begins with a trivial triangulation of the domain and then iteratively adds vertices according to which input point has the greatest vertical error with respect to the approximating surface. Retriangulation is done using the Delaunay triangulation.

GcTin. GcTin, developed by Silva et al. [15, 14], uses an advancing-front technique for simplification of digitized terrain models. The algorithm takes greedy cuts ("bites") out of a simple closed polygon that bounds a connected component of the yet-to-be triangulated region. The method begins with a large polygon, bounding the whole extent of the terrain to be triangulated, and works its way inward, performing at each step one of three basic operations: ear cutting, greedy biting, and edge splitting. One of the main advantages of GcTin is that it requires very little memory beyond that for the input height array.

QSlim. QSlim (developed by Garland and Heckbert [9]) is a more general algorithm designed for simplifying all types of surfaces, not just terrains. QSlim uses simple edge contraction to perform simplification, while using a quadric error measure for efficiency and for visual fidelity.

5.2 Terrain datasets

Ten input terrains representing ten different and varied geographic regions were used. Each input terrain covers a rectangular area of approximately 15×10 km^2, and consists of approximately 20,000 triangle vertices.

5.3 Experiments using the transmitter-receiver measure

Most of our tests use the quality measure devised for the antenna location application, so we begin with them.

For each input terrain T, four tests $t_1, \ldots, t_4$ were generated, according to the specifications in Table 1. (We cannot

test	no. of antennas	range in km (r_1, r_2)	no. of receivers per antenna
t_1	30	2–2.5	30
t_2	50	3–4	30
t_3	50	4–5	50
t_4	30	2–6	100

Table 1: The tests.

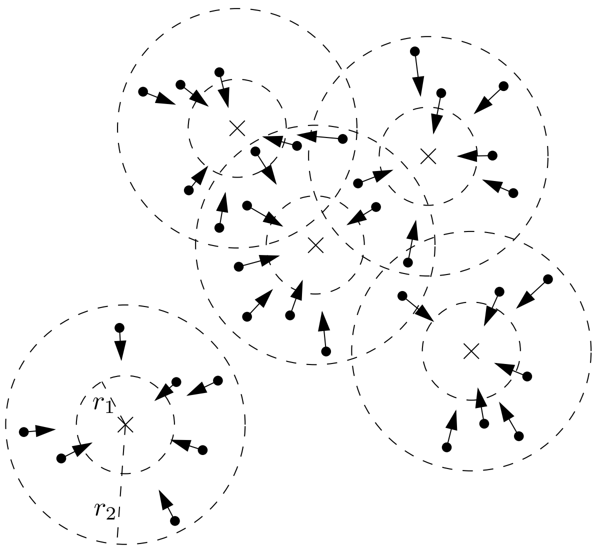

Figure 6: A test with 5 antennas and 7 reception points per antenna.

use the same set of tests for all input terrains, since the underlying rectangles vary slightly in their dimensions.) For example, test t_4 for T is generated by selecting a random sample A of 30 points in the underlying rectangle R. These points are the (x- and y-coordinates of the) locations of the antennas. For each antenna location $a \in A$, a random sample B_a of 100 points is chosen from the annulus with radii 2 and 6 km, centered at a and confined to R; the points B_a are the locations of the reception points associated with a. An example is shown in Figure 6.

When testing a simplification T' of T with t_4, we need to determine for each of the $30 \times 100 = 3000$ pairs (a, b), where $a \in A$ and $b \in B_a$, whether a and b see each other in T and whether they see each other in T'. (Points p and q see each other in T if the segment $p_T q_T$, obtained by lifting p and q to the surface of T, lies entirely above or on the surface of T.)

For each input terrain T and for each of the four simplification methods, we generate simplifications of T of several sizes (where the size of a simplification is the number of its triangle vertices). The range of these sizes was determined so that we obtain simplifications that are both small and good, as required by the antenna location application. A somewhat surprising finding is that if the size of a simplification of T, obtained by *any* of the four methods, is above 10% of the size of T, then $\sigma_{\mathcal{X}}(T, T') \geq 0.99$, for any set of pairs $\mathcal{X}$ corresponding to one of the four tests. Thus, we set the upper limit of the range of sizes to 1000 vertices, which is roughly 5% of the size of T. On the other hand, for all simplifications of T of size less than 0.5% of the size of T and for all tests, $\sigma_{\mathcal{X}}(T, T') \leq 0.95$. Thus we set the lower limit of the range of sizes to 125 vertices, which is roughly 0.625% of the size of T. The actual sizes that we consider therefore are the limit sizes, 1000 and 125, plus two intermediate sizes — 500 and 250.

method	t_1	t_2	t_3	t_4	average
GcTin	0.885	0.920	0.942	0.937	0.921
QSlim	0.912	0.943	0.960	0.952	0.942
Terra	0.917	0.949	0.963	0.951	0.945
VPTS	0.945	0.950	0.963	0.954	0.953

Table 2: Results for simplifications of size 125.

method	t_1	t_2	t_3	t_4	average
GcTin	0.933	0.956	0.967	0.965	0.955
QSlim	0.936	0.968	0.973	0.971	0.962
Terra	0.947	0.970	0.974	0.970	0.965
VPTS	0.961	0.971	0.975	0.974	0.970

Table 3: Results for simplifications of size 250.

method	t_1	t_2	t_3	t_4	average
GcTin	0.949	0.962	0.960	0.967	0.959
QSlim	0.959	0.973	0.970	0.974	0.969
Terra	0.968	0.976	0.983	0.980	0.977
VPTS	0.973	0.978	0.981	0.980	0.978

Table 4: Results for simplifications of size 500.

method	t_1	t_2	t_3	t_4	average
GcTin	0.964	0.968	0.953	0.970	0.964
QSlim	0.982	0.978	0.967	0.977	0.976
Terra	0.975	0.984	0.988	0.985	0.983
VPTS	0.980	0.988	0.987	0.985	0.985

Table 5: Results for simplifications of size 1000.

The results for simplifications of size 125 (resp., 250, 500, or 1000) are presented in Table 2 (resp., Table 3, Table 4, or Table 5). Consider, e.g., Table 3. The values in, e.g., the first row correspond to the 10 simplifications obtained by GcTin for the 10 input terrains. The ith value in this row is the average over the scores of these simplifications with respect to the corresponding t_i tests.

From Tables 2–5 we see that our method improves with respect to the other methods when the simplification size decreases. It is better than the others for simplifications of size 125, 250, or 500, while it is among the two best (together with Terra) for simplifications of size 1000. This is not surprising. As long as any simplification is detailed enough, it also approximates the ridges quite well. However, when the amount of detail goes down, the high priority that is given to ridges in our method becomes very significant.

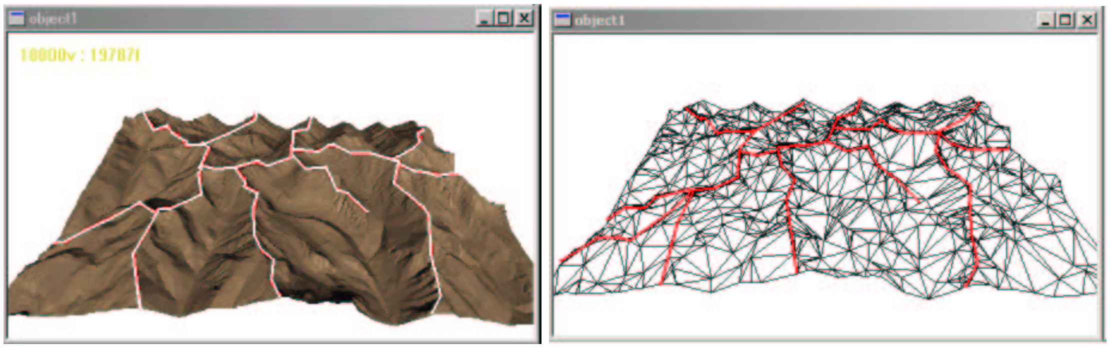

Figure 7: The ridge network (partially) marked on a terrain dataset (left) and on a simplification of it obtained by VPTS (right).

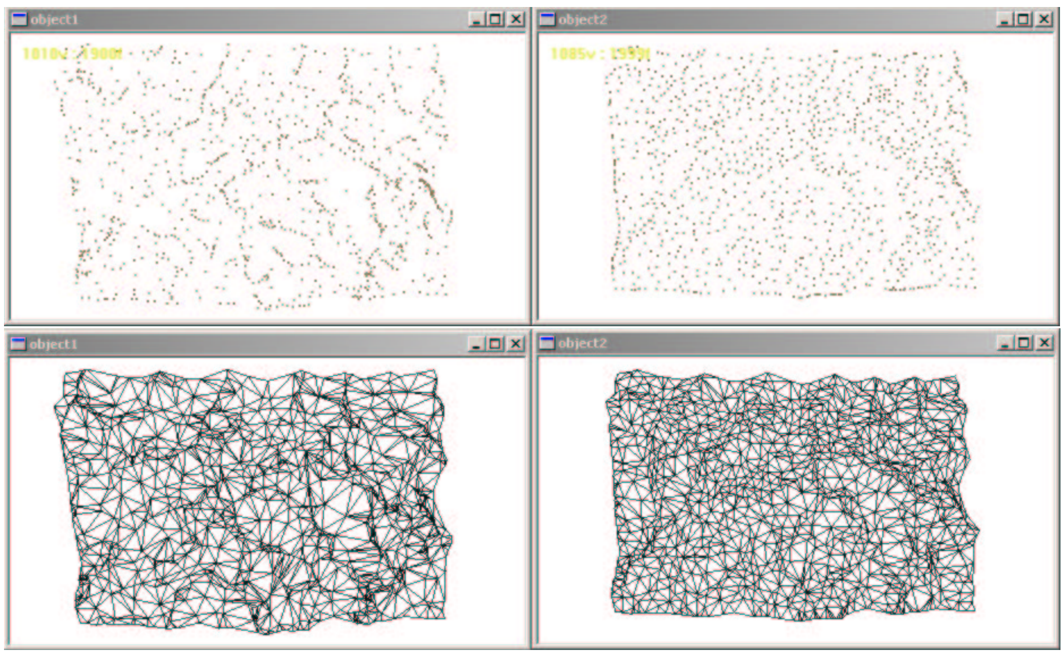

Figure 8: Selecting ~ 1000 vertices. Above: VPTS on the left vs. Terra on the right. Below: The corresponding triangulations.

method	quality 0.985	quality 0.978	quality 0.970	quality 0.953
Grid	2950	2100	1775	780
Terra	1155	545	315	165
QSlim	1305	1015	550	185
GcTin	1700	1435	1110	310
VPTS	1000	500	250	125

Table 6: Size needed for given quality.

method	size 250	size 1000
GcTin	0.961	0.980
QSlim	0.969	0.983
Terra	0.974	0.988
VPTS	0.975	0.988

Table 7: Average results for simplifications of size 250 and 1000 (ideal measure).

Table 6 provides more evidence for the above conclusion. In this table we see the average simplification size (based on our input terrains and tests) that is needed in order to achieve a given quality. For example, while the average over the sizes of the minimum size simplifications generated by our method that achieve quality 0.970 is only 250, this average for the other methods is larger; it is 315 for Terra, 550 for QSlim, and 1110 for GcTin.

It is also interesting to note that, as expected, the maximal vertical distance (i.e., along the z-axis) between a vertex of the input terrain and the corresponding point on the surface of the simplification, is much larger in simplifications obtained by our method, even when our simplification is better than the others according to the transmitter-receiver measure.

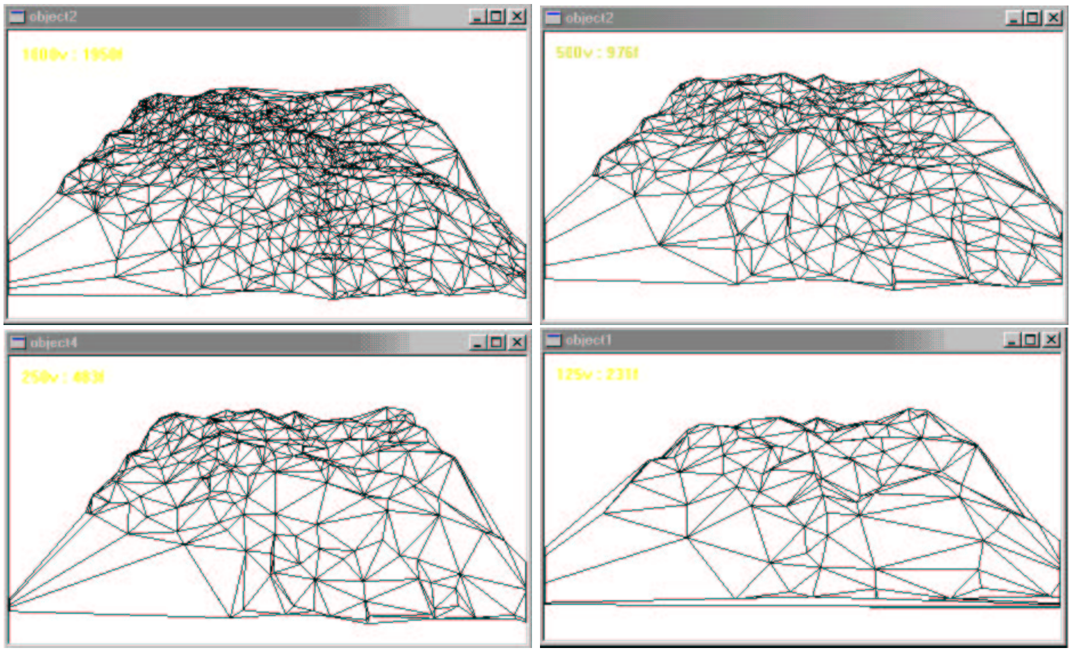

Figure 9: Simplifications obtained by VPTS of size 1000 (top left), 500 (top right), 250 (bottom left), and 125 (bottom right).

5.4 Experiments with the ideal measure

For each of the 10 input terrains T, two tests t_5 and t_6 were generated. Both tests were generated by picking a random sample of size 50 from the underlying rectangle R. In both cases the corresponding set of pairs of points $\mathcal{X}$ consists of all $\binom{50}{2}$ pairs. Table 7 shows the results. As expected, the values for the ideal measure are slightly higher and less indicative than the values for the transmitter-receiver measure (see discussion in Section 2).

6. CONCLUSION

We have proposed a new quality measure for terrain simplification that is based on inter-point visibility. This measure is potentially relevant in several applications; we were especially motivated by the antenna location application, as part of our industrially sponsored LSRT (Large Scale Rural Telephony) project. We have developed a simplification method that is well suited for this measure, especially with relatively small sample sizes, as our experimental analysis shows.

Several improvements are currently under development and may be reported in the final paper, including:

1. Additional tests with other terrain simplification methods.

2. Improved methods of approximating the ridge network, including fine tuning of the various parameters.

3. Extension of the quality measure to account for the fact that sometimes strict visibility is not necessary, and the reception at a point depends on the profile of the terrain between the point and the antenna. This motivates an alternative definition of quality measure in terms of the quality of the 2-dimensional profile approximation along each pair of sample points.

4. Theoretically based approximation algorithms for computing nearly optimal terrain approximations in the context of visibility preservation.

Acknowledgment. The authors wish to thank Amir Glatt who helped setting up the environment in which the tests were performed.

7. REFERENCES

[1] P. K. Agarwal and P. K. Desikan. An efficient algorithm for terrain simplification. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 139–147, 1997.

[2] *CGAL — Computational Geometry Algorithms Library*, Release 2.3, `www.cgal.org` .

[3] P. Cignoni, E. Puppo, and R. Scopigno. Representation and visualization of terrain surfaces at variable resolution. *The Visual Computer*, 13:199–217, 1997.

[4] D. Cohen-Or and A. Shaked. Visibility and dead-zones in digital terrain maps. *Comput. Graph. Forum*, 14(3):171–179, 1995. Proc. EUROGRAPHICS '95.

[5] L. De Floriani and P. Magillo. Horizon computation on a hierarchical triangulated terrain model. *Visual Comput.*, 11:134–149, 1995.

[6] L. De Floriani, P. Marzano, and E. Puppo. Multiresolution models for topographic surface description. *The Visual Computer*, 12(7):317–345, 1996.

[7] M. Duchaineau, M. Wolinsky, D. Sigeti, M. Miller, C. Aldrich, and M. Mineev-Weinstein. ROAMing terrain: Real-time optimally adapting meshes. In *IEEE Visualization '97 Proceedings*, pages 81–88. ACM/SIGGRAPH Press, Oct. 1997.

[8] R. J. Fowler and J. J. Little. Automatic extraction of irregular network digital terrain models. In *Comput. Graph. (SIGGRAPH'79)* 13(2), pages 199–207, 1979.

[9] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proc. SIGGRAPH '97*, pages 209–216, 1997.

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

[11] J. J. Little and P. Shi. Structural lines, TINs, and DEMs. *Algorithmica*, 30(2):243–263, 2001.

[12] N. L. Max. Horizon mapping: Shadows for bump-mapped surfaces. *The Visual Computer*, 4(2):109–117, 1988.

[13] L. Scarlatos and T. Pavlidis. Hierarchical triangulation using cartographic coherence. *CVGIP: Graph. Models Image Process.*, 54(2):147–161, Mar. 1992.

[14] C. T. Silva, J. S. B. Mitchell, and A. E. Kaufman. Automatic generation of triangular irregular networks using greedy cuts. In *Visualization 95*, pages 201–208, San Jose CA, 1995. IEEE Computer Society Press.

[15] C. T. Silva and J. S. B. Mitchell. Greedy cuts: An advancing front terrain triangulation algorithm. In *Proc. 6th ACM Workshop on Advances in GIS*, pages 137–144, 1998.

[16] A. J. Stewart. Fast horizon computation at all points of a terrain with visibility and shading applications. *IEEE Trans. Visualizat. Comput. Graph.*, 4(1):82–93, Jan. 1998.

[17] S. Yu, M. van Kreveld, and J. Snoeyink. Drainage queries in TINs: from local to global and back again. In *Proc. 7th Internat. Sympos. Spatial Data Handling*, pages 13A.1–13A.14, 1996.