AN IMPROVED REFINEMENT AND DECIMATION METHOD FOR ADAPTIVE TERRAIN SURFACE APPROXIMATION

Hélio Pedrini

Universidade Federal do Paraná
Departamento de Informática
Curitiba-PR, Brazil, 81531-990
helio@inf.ufpr.br

ABSTRACT

An improved method for adaptively constructing a terrain surface representation from a set of data points is presented. Refinement and decimation steps are repeatedly applied to triangular meshes, incrementally determining a better distribution of the data points, while a specified error tolerance is preserved. Even though not asymptotically optimal or monotonically convergent, it produces approximations that are, experimentally, significantly better than those generated by straightforward greedy insertion algorithms.
A new local error metric is used to select points to be inserted into the triangulation, based on the maximum vertical error weighted by the standard deviation calculated in a neighborhood of the candidate point. Conversely, a measure of angle between surface normals is used to determine whether a vertex should be removed from the triangulation. The method has been implemented and tested on both synthetic test cases and real terrain data sets.

Keywords: surface simplification, triangulated irregular network, mesh optimization.

1 INTRODUCTION

The approximation of a bivariate function from a set of data points occurs in a number of applications, such as computer-aided design, computer vision, computer graphics, finite element methods, and terrain modeling. In many of these applications, the surface representation can be generally viewed as a $2\frac{1}{2}$- dimensional modeling problem, where a bivariate function $z = f(x, y)$ expresses the elevation z of the surface at a point (x, y) of the Euclidean plane. Therefore, any line parallel to z axis penetrates the surface at most once.

In terrain modeling, our major interest, a common method for approximating topographic surfaces is to use regular grid digital elevation models (DEMs), in which a set of sampled points representing measures of altitude or elevation are stored at regular intervals. A disadvantage of the DEM is its inherent spatial invariability, since the structure is not adaptive to the irregularity of the terrain. This may produce a large amount of data redundancy, especially where the topographic information is minimal.

Alternatively, triangulated irregular networks (TINs) represent the terrain surface as a mesh of adjacent triangles, whose vertices are the elevation points. The points need not lie in any particular pattern and the density may vary over space. There are many advantages associated with TINs. First, terrain data are commonly irregularly distributed in space, therefore, the structure of the triangulation can be adjusted to reflect the density of the data. Consequently, cells become larger where data are sparse, and smaller where data are dense. Second, terrain features can be incorporated into the model. For instance, vertices in a TIN can describe nodal terrain features such as peaks, pits or passes, while edges can represent linear terrain features such as break, ridge or channel lines. Third, TINs can be organized into a hierarchical model so that they can represent a terrain in various levels of detail. Finally, triangles are simple geometric objects which can be easily manipulated and rendered.

The triangulation of a set of data points in the plane can be defined in terms of a planar graph in which pairs of vertices are connected by edges intersected only at their endpoints, forming triangular faces. The topology of the triangulation can be generally chosen either using only the xy projections of the data points or using the elevations of the points as well. The latter approach is called data-dependent triangulation [Dyn90, Quak90]. The most common triangulation method that uses only the xy projections is the Delaunay triangulation. The Delaunay triangulation has the property that the circumcircle of any triangle in the triangulation contains no other data points in its interior, known as *circle property*. The Delaunay triangulation generates the triangulation that maximizes the minimum angle of all triangles. This property is known as *max-min angle property*. In a Delaunay triangulation, most of its triangles are nearly equiangular, which helps to minimize the occurrence of thin and long triangles since they can lead to undesirable behavior, affecting numerical stability and producing visual artifacts. Another interesting property is that Delaunay triangles define nearest natural neighbors in the sense that the data points at the vertices are closer to their circumcenter than is any other data point. These circumcenters are the positions of vertices in the geometrically dual Voronoi diagram, also known as Dirichlet, Thiessen or Wigner-Seitz tessellation. Other criteria can be used to construct triangulations, for instance, the minimum weight triangulation (MWT) is a triangulation that minimizes the sum of the lengths of all the edges.

Most triangulation methods produce poor approximations in regions of discontinuity or poor vertex selection in the presence of noise. This occurs because there is no obvious strategy for determining the optimal vertex locations in advance, and vertices inserted (or deleted) early in the refinement (or decimation) process may later become unnecessary by better vertices. Since the insertion of additional vertices can result in a nonsmoooth surface approximation or in an inadequate data distribution, an approach that might be useful is to alternate refinement and decimation steps, inserting several vertices with an incremental triangulation algorithm, and then removing a few vertices that appear the least important.

An example caused by the short-sightedness of most incremental triangulation algorithms is given by Garland and Heckbert [Garla95], where the object to be approximated contains a step discontinuity or *cliff* (Figure 1). The left half of the grid has constant height 0 and the right half has constant height 1. From a 100×100 grid, 99 vertices were selected to achieve zero error. Since only 8 vertices suffice, the triangulation algorithm generated several unnecessary vertices.

Agarwal and Suri [Agarw94] prove that the problem of approximating surfaces while minimizing the number of vertices for a given accuracy is *NP-hard*. Practical solutions found in the literature are often based on heuristics that attempt to produce an approximate model by either iteratively adding new vertices to a coarse triangulation or iteratively removing points from an initial triangulation built over the entire data set.

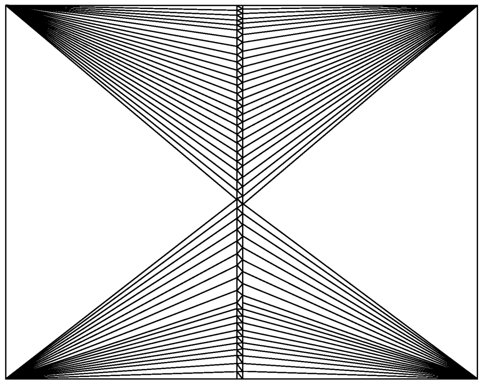


Figure 1: Redundant triangulation generated by incremental triangulation algorithm. Figure adapted from Garland and Heckbert (1995).

This paper presents a method for adaptively approximating surfaces through repeated refinement and decimation passes, providing higher quality triangulations with great flexibility. The method, analogous in some aspects to other approximation techniques such as stepwise linear regression techniques, incrementally improves the triangulation by determining a better distribution of the data points, while maintaining a specified error tolerance. Although the method is not necessarily optimal or monotonically convergent, it produces approximations that are significantly better than those generated by straightforward greedy insertion algorithms. A Delaunay triangulation is used to create the sequence of triangulations whose vertices lie at a subset of the data points. Section 2 reviews the main methods for simplifying polygonal surfaces. Section 3 describes our new model, which generates a sequence of triangulations based on a set of refinement and decimation operations. In Section 4, some experimental results are presented and discussed. Section 5 concludes with some final remarks and directions for future research.

2 RELATED WORK

Although many surface representations have been proposed in the literature, polygonal surfaces are the most common choice for representing three-dimensional data sets in computer graphics, scientific visualization, digital terrain, modeling, planetary exploration, rapid prototyping, and computer-aided design. Polygonal surface data are widely available and supported by the vast majority of modeling and rendering packages. Hardware support for polygon rendering is also becoming more popular.

Most polygonal surface representation methods found in the literature can be classified as *refinement* and *decimation* methods. Refinement methods start with a minimal initial approximation of the surface and repeatedly add new points to the triangulation until the model satisfies a specified approximation criterion. Decimation methods start with a triangulation containing the entire set of data points and iteratively simplify it, until the desired approximation criterion is achieved.

The concept of *multiresolution modeling* is generally associated with the possibility of representing a geometric object at different levels of detail [Linds96] and accuracy. For a given application, a coarse representation can be used to describe less relevant areas, while high resolution can be focused on specific parts of interest.

Several surface simplification approaches [Cohen96, Hoppe93, Hoppe96, Schro92, Turk92] have been proposed in recent years, a survey of the relevant work is this field is given by Heckbert and Garland [Heckb97]. Our focus here is on those methods that are more related to simplification of height fields.

3 NEW METHOD

We propose a new method for adaptively approximating terrain surfaces through repeated refinement and decimation passes, incrementally determining a better distribution of the data points, while a specified error tolerance is preserved. Initially, a minimal approximation consisting of two triangles is constructed. This mesh is then incrementally refined until either a specified error is achieved or a given number of points is reached. Once the desired level of accuracy has been satisfied, the approximation is simplified by eliminating a small number of points based on a vertex removal criterion. Finally, the approximation is again refined to the given error tolerance and partially resimplified. This alternate refinement and decimation process is repeated until no further improvement in the accuracy of the approximation can be achieved.

As described above, the first step of our method is to generate a coarse piecewise linear approximation of the surface according to a predefined error tolerance. This initial triangulation is then refined by iteratively adding new points, updating it after each point is inserted. The Delaunay triangulation is used to incrementally construct the mesh from a large number of points, reducing the occurrence of thin and long triangles since they can lead to undesirable behavior, affecting numerical stability and producing visual artifacts.

The vertex selection criterion is crucial during the triangulation process since it determines the degree of fidelity between the original data and the approximation. The magnitude of the error can be estimated by using the L_n function norm, defined as

$$L_n(\phi) = ||\phi||_n = \sqrt[n]{\int_\Phi |\phi(x,y)|^n dxdy} \quad (1)$$

where ϕ is a function defined over domain Φ.

When used to characterize the error of an approximation, $\phi(x,y)$ represents the difference between the original and approximate surface. The norm can be computed over a limited region to estimate *local* error or over the entire domain to measure *global* error. Global error measures usually produce better approximations, however, the resulting algorithms are significantly slower than those using local metrics. The most commonly used norms are the L_1, L_2, and L_∞ norms. The L_1 norm of the approximation error, ϵ, corresponds to the volume between the surfaces. The L_2 norm provides a measure of the average root mean square (RMS) error between the original and the approximation. Another common measure is based on the maximum difference between the actual elevation data and the surface approximation (Figure 2), known as L_∞ norm.

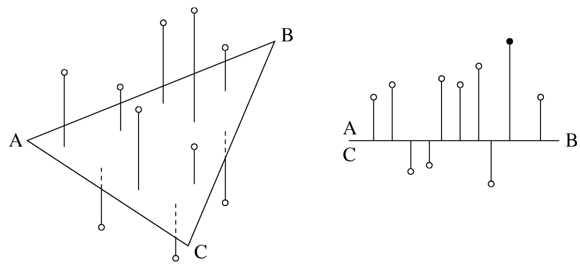


Figure 2: Maximum vertical error.

A variation to the conventional maximum vertical error measure is proposed as the vertex selection criterion, which is based on the maximum vertical error weighted by the standard deviation calculated in a neighborhood of the candidate point, given by

$$C = \frac{h(p) - z(p)}{\sigma(p)} \quad (2)$$

where $h(p)$ is the height value of point p, $z(p)$ is the height value of the interpolated surface at point p, and

$\sigma(p)$ is the standard deviation calculated in a 3×3 neighborhood of the candidate point p.

The idea is to associate greater importance to the points in regions where the local variability of the data is high, allowing the surface to conform to the local trends in the data. Our experiments have demonstrated that this vertex selection criterion is slightly superior to the vertical error measure. In other words, to minimize the maximum error, it is better not to refine the triangle with the maximum error, but rather to refine triangles where the curvature is high.

A priority queue stores the sequence of vertices used to refine the triangulation, ordered by increasing approximation error. For each refinement step, only those vertices affected by the insertion process need to have their approximation error recalculated.

Since this strategy for determining the vertex locations is based on a local heuristic, suboptimal insertions can eventually be performed by the refinement process. The approach proposed to deal with this problem is to identify and remove those points that may have become unnecessary by later insertions.

A decimation algorithm is then defined to produce a sequence of triangulations through a set of vertex removal operations. At each iteration, the vertex with the smallest error is removed and the area affected by its removal is retriangulated. This process is repeated until a specified error tolerance is achieved.

The criterion for removing a vertex v is computed by averaging the surface normals n_i of the triangles surrounding v weighted with their areas A_i (see Figure 3) and taking the maximum angle, $\alpha_{\max}$, between the averaged normal, n_{av} and the surrounding triangles, that is

$$\alpha_{\max} = \max\left(\arccos\frac{\vec{n}_{av}.\vec{n}_i}{|\vec{n}_{av}|.|\vec{n}_i|}\right) \qquad (3)$$

where $\vec{n}_{av} = \dfrac{\sum \vec{n}_i.A_i}{\sum A_i}$ and $0 \leq \vec{n}_i \leq \pi$.

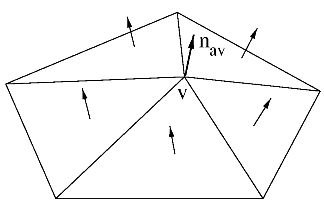


Figure 3: Criterion for vertex removal.

The area around the removed point is retriangulated, which requires careful consideration, in particular when such area is not a convex polygon. The edges that form the triangulation of the polygon surrounding the vertex v must be checked to determine if they do not intersect one another. Figure 4, for instance, illustrates an invalid retriangulation.

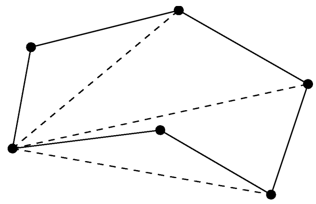


Figure 4: An invalid retriangulation.

Similarly to the refinement algorithm, for each vertex affected by the local retriangulation, the approximation error is recalculated and stored in the priority queue. The algorithm stops when the smallest retriangulation error of a vertex becomes larger than error tolerance ϵ.

Another important aspect of our method is that *a priori* information about topographic characteristics of the terrain can be incorporated into the triangulation. This information can describe nodal features (such as peaks, pits, or passes) and linear features (such as ridges, rivers, roads, channels, or cliffs). These nodal and linear features are inserted in the triangulation as constrained vertices and edges, respectively, in a such way that subsequent operations will preserve them.

Heller [Helle90] describes an algorithm where linear features (breaklines) are inserted in an existing triangulation. The reorganization of the mesh to adjust such breaklines, however, produces regions of long and thin triangles.

In our approach, the constraints are included as the first elements of the triangulation, then reducing the number of edge swaps necessary to update the mesh each time a new breakline needs to be inserted. Breaklines are positioned along the edges of triangles, which is another surface behavior that cannot be easily handled by grid-based methods. Additionally, new points can be added on the constrained edges to guarantee that the final triangulation satisfies the Delaunay criterion.

The sequence of local modifications generated during the refinement and decimation steps is applied to a triangulation until no further improvement in the accuracy of the approximation can be achieved. The repeated application of these steps can be viewed through the diagram shown in Figure 5.

Pseudocodes for the refinement and decimation steps are presented in Figures 6 and 7, respectively.

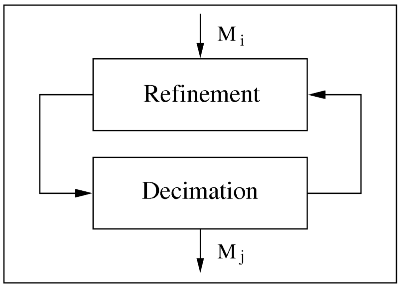


Figure 5: Refinement and decimation steps.

```
// refinement process
construct initial triangulation using
 domain boundary;
if ∃ constrained vertices and edges
  include them in the triangulation;
compute approximation error for each vertex;
while (maximum error > error tolerance ε) {
 find vertex v with maximum error;
 insert v;
 update triangulation;
 recompute error for vertices affected
  by the local update;
}
```

Figure 6: Pseudocode for refinement step.

The extraction of a representation of the terrain at a given tolerance level is obtained by using a coarse triangulation and iteratively inserting vertices into the triangulation until the desired precision is satisfied. If a given triangulation already guarantees a smaller error tolerance, then vertices are removed from the triangulation, starting with the vertex with the smallest error.

4 EXPERIMENTAL RESULTS

Our combined refinement and decimation method has been tested on a number of data sets in order to illustrate its performance. Due to space limitations, only one sample is presented here. The algorithms were implemented in C++ programming language on UNIX platform.

Figure 8 shows the digital elevation model of Crater Lake, where elevations range from 1533m and 2478m, standard deviation of 162.6m, and 30- by 30-meter data spacing. The sample consists of 336×459 elevation points.

Figures 9a-b show approximations obtained by applying our method and a greedy insertion algorithm, respectively, to the Crater Lake DEM. The triangulation produced by our method has 126 vertices and 230 triangles, whereas the other triangulation contains 140 vertices and 263 triangles. The corresponding root mean square error (RMSE) for both approximations is 29m. Figure 10 shows an approximation for the

```
// decimation process
while (maximum error < error tolerance ε) {
 find vertex v with minimum error;
 find vertices w_i = w_1,...,w_k adjacent to v;
 if v satisfies decimation criteria {
  remove v;
  delete all triangles connected to v;
  retriangulate polygon defined by
   vertices w_i;
  recompute error for each w_i;
  modify list of vertices;
 }
}
```

Figure 7: Pseudocode for decimation step.

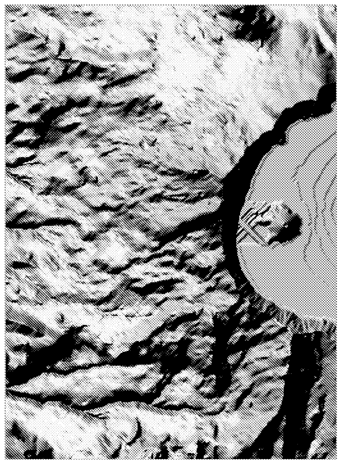


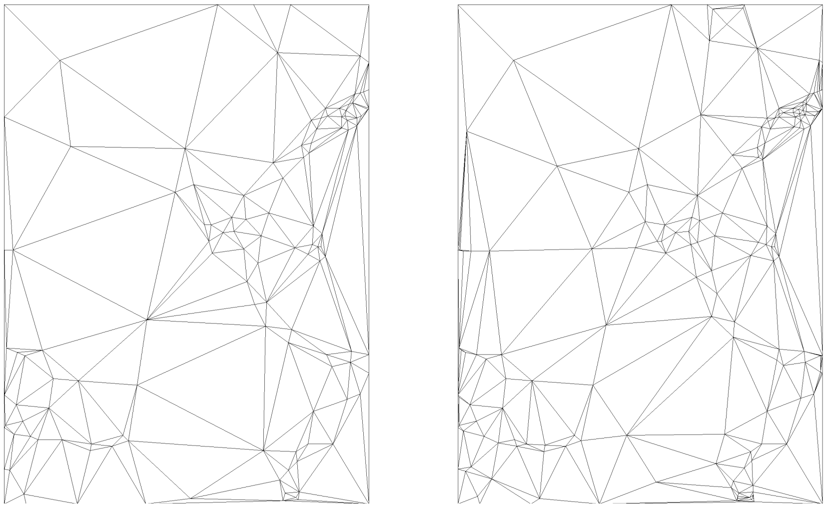
Figure 8: The USGS Crater Lake West DEM.

digital elevation model of Crater Lake using 5.0% of the original points.

Our method has also been tested on several other terrain data sets, including non-terrestrial terrains. The results have demonstrated a good balance between speed and ability to process large terrain data sets. In the refinement step, the algorithm is able to select 55,000 points in approximately 60 seconds (measured on an SGI O2 workstation (IRIX 6.5, R5000 with a 200MHz MIPS processor and 64 Mbytes of main memory).

5 CONCLUSIONS

We have described a method for the simplification of triangulations approximating a bivariate function. The simplification of the triangulation is performed by a hybrid refinement and decimation approach. A new local error metric is used to select points to be inserted into the triangulation, which is based on the maximum vertical error weighted by the standard deviation calculated in a neighborhood of the candidate point. Conversely, a measure of angle between surface normals is used to determine whether a vertex should be removed from the triangulation. Our combined refinement/decimation method produces locally optimal approximations, which are significantly bet-



(a) (b)

Figure 9: Approximations of Crater Lake DEM constructed by using (a) our combined refinement/decimation method (230 triangles) and (b) greedy insertion algorithm (263 triangles).

ter than those generated by greedy insertion algorithms.

Some ideas for future research include the use of more sophisticated techniques for evaluating the accuracy of the approximation, which can incorporate relevant features of the objects. In the context of terrain modeling, for instance, ridge lines, valley lines, measures of visibility [Frank94], or drainage networks [Yu96] might be used to guide the triangulation process. Indeed, preservation of these properties in the approximated surface is more important than just minimizing the maximum error.

Finally, superior quality representations can be obtained by constructing a curved approximation of the original triangulation through higher-order approximating surfaces, instead of piecewise-planar subdivisions. It's possible that merely fitting a C^1 spline to the existing triangulation might tend to reduce the error, even though it adds no information. Additional work is needed to establish practical merits of such techniques, particularly in cartographic applications.

ACKNOWLEDGEMENTS

This work was partially supported by a grant from Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, Brazil. The digital elevation model is courtesy of the United States Geological Survey.

REFERENCES

[Agarw94] Pankaj K. Agarwal and Subhash Suri. Surface approximation and geometric partitions. In *Proceedings 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 24–33, Arlington, Virginia, USA, January 1994.

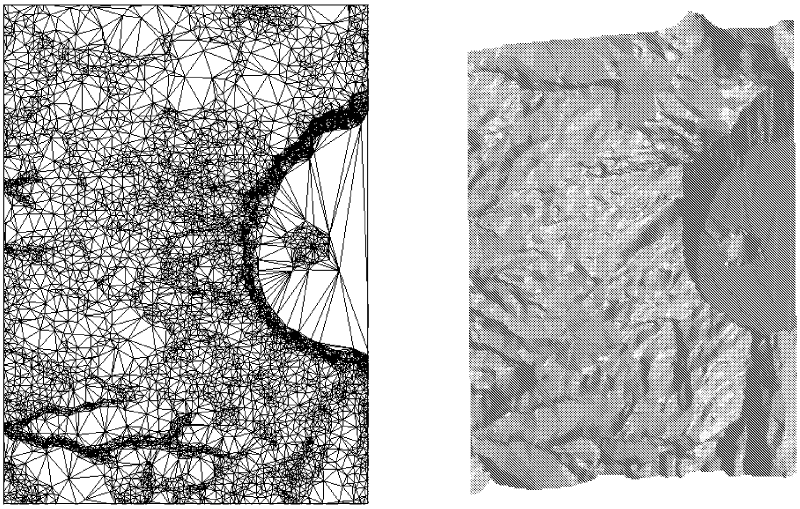
[Cohen96] Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Jr. Frederick P. Brooks, and William Wright. Simplification envelopes. In *SIGGRAPH'96 Conference Proceedings*, Annual Conference Series, pages 119–128, New Orleans, Louisiana, USA, August 1996.

[Dyn90] Nira Dyn, David Levin, and Shmuel Rippa. Data dependent triangulations for piecewise linear interpolation. *IMA Journal on Numerical Analysis*, 10(1):137–154, January 1990.

[Frank94] William Randolph Franklin and Clark Ray. Higher isn't necessarily better: Visibility algorithms and experiments. In *Advances in GIS Research Proceedings: 6th International Symposium on Spatial Data Handling*, volume 2, pages 751–770, Edinburgh, Scotland, September 1994.

[Garla95] Michael Garland and Paul S. Heckbert. Fast polygonal approximation of terrains and height fields. Technical Report CMU-CS-95-181, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, September 1995.

[Heckb97] Paul S. Heckbert and Michael Garland. Survey of polygonal surface simplification algorithms. Technical Report CMU-CS,



7,711 vertices; 15,231 triangles; RMSE = 1.91m

Figure 10: Approximation of the Crater Lake DEM constructed by using our combined refinement/decimation method. The algorithm uses 5.0% of the original points.

Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, May 1997.

[Helle90] Martin Heller. Triangulation algorithms for adaptive terrain modeling. In *Proceedings 4th International Symposium on Spatial Data Handling*, pages 163–174, Zurich, Switzerland, July 1990.

[Hoppe93] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *SIGGRAPH'93 Conference Proceedings*, pages 19–26, Anaheim, California, USA, August 1993.

[Hoppe96] Hugues Hoppe. Progressive meshes. In *Computer Graphics, SIGGRAPH'96 Proceedings*, pages 99–108, New Orleans, Louisiana, USA, August 1996.

[Linds96] Peter Lindstrom, David Koller, William Ribarsky, Larry F. Hodges, Nick Faust, and Gregory A. Turner. Real-time, continuous level of detail rendering of height fields. In *SIGGRAPH'96 Conference Proceedings*, Annual Conference Series, pages 109–118, New Orleans, Louisiana, USA, August 1996.

[Quak90] Ewald Quak and Larry L. Schumaker. Cubic spline fitting using data dependent triangulations. *Computer-Aided Geometric Design*, 7(1–4):293–301, June 1990.

[Schro92] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65–70, July 1992.

[Turk92] Greg Turk. Re-tiling polygonal surfaces. *Computer Graphics*, 26(2):55–64, July 1992.

[Yu96] Sidi Yu, Marc van Kreveld, and Jack Snoeyink. Drainage queries in TINs: From local to global and back again. In *Proceedings 7th International Symposium on Spatial Data Handling*, pages 13A.1–13A.14, Delft, Netherlands, August 1996.