Computer Science Department of The University of Auckland
CITR at Tamaki Campus (http://www.tcs.auckland.ac.nz**)**

CS-TR-153 / CITR-TR-10 September 1997

Multiresolution Surface Reconstruction: Edge Collapsing + Simplification Envelopes

Shao-zheng Zhou and Reinhard Klette*

Abstract

Multiresolution surfaces are especially useful for fast rendering, real-time display, and progressive transmission. This paper reviews major surface simplification techniques and multiresolution surface reconstruction approaches. Based on comparison among various approximation algorithms we propose an appropriate measure for surface approximation accuracy and essential concepts for multiresolution surface reconstruction. Having analyzed the surface simplification process, we propose our solution for multiresolution surface reconstruction - combination of the edge collapsing operation and simplification envelopes, which can generate continuous multiresolution surfaces with globally-guaranteed approximation errors.

* The University of Auckland, Tamaki Campus, Computing and Information Technology Research, Computer Vision Unit, Auckland, New Zealand

Multiresolution Surface Reconstruction: Edge Collapsing + Simplification Envelopes

Shao-zheng Zhou and Reinhard Klette

Computing and Information Technology Research
Computer Science Department, The University of Auckland
Tamaki Campus, Auckland, New Zealand
Email: {szho001, rklette}@cs.auckland.ac.nz

Abstract

Multiresolution surfaces are especially useful for fast rendering, real-time display, and progressive transmission. This paper reviews major surface simplification techniques and multiresolution surface reconstruction approaches. Based on comparison among various approximation algorithms we propose an appropriate measure for surface approximation accuracy and essential concepts for multiresolution surface reconstruction. Having analyzed the surface simplification process, we propose our solution for multiresolution surface reconstruction - combination of the edge collapsing operation and simplification envelopes, which can generate continuous multiresolution surfaces with globally-guaranteed approximation errors.

Keywords: Delaunay triangulation, edge collapse, surface approximation, surface simplification, triangulated irregular networks

1. Introduction

Many techniques in Computer Vision are directed on reconstructing 3D surfaces: binocular stereo, photometric stereo, structured lighting etc. Assuming that these techniques have successfully reconstructed a surface, then the next step is to represent this surface under special (e.g. incremental transmission) conditions. In Computer Graphics, surfaces are generally represented by very dense triangle meshes for the reasons that the triangle structure is simple and that the support for triangle rendering from computer hardware and commercial software is well-developed. With the advances of range scanners and satellite photography, 3D surface models have acquired more details and become more complex. Currently, surface models with upto millions of triangles are quite ordinary. The rendering and manipulation of so huge amounts of data may easily be beyond of the processing capability of the state-of-the-art computers. In many situations, however, the highly detailed models are not necessary. In Virtual Reality, for example, the background and farther objects are definitely unnecessary to have the same level of detail as the closer objects. They should be represented by simplified models with coarser resolution, which is actually perceptually correct. Recently, multiresolution surface reconstruction or modeling becomes very popular. Multi-resolution surfaces are especially useful for progressive transmission and compression in the *Internet*. For Computer Vision, multiresolution surface reconstruction techniques are employed to remove the redundancy and speed up the display and recognition. For the large scale map production in Cartography, surface simplification is vital to simplify the representation of rivers, roads, coastlines, buildings, and terrains. A typical application is the construction of the TIN (Triangulated Irregular Networks) from large amounts of geographical data acquired from satellites. For fast display and real-time rendering, multiresolution surfaces are necessary in Virtual Reality, Scientific Visualization and Computer Graphics. Recently, the view-dependent surface modeling or selective refinement is under development in which different regions of the same object should be represented in different resolutions or different levels of detail (LOD).

Surface simplification is a subproblem of multi-resolution surface reconstruction. Taking input in the form of a regular grid mesh and TIN, multiresolution surface reconstruction will generate a series of approximation surfaces with continuous incremental resolutions while surface simplification will generate only one simplified surface. Curve simplification is similar to surface simplification, but simpler. In short, multi-resolution surface reconstruction can be summarized as:

- Assume a given surface with n vertices and their neighbourhoods, find a series of approximation surfaces with m vertices $(m \in (i, i+1, \ldots, n); i > 0)$ or find a series of approximation surfaces with approximation error ε $(\varepsilon \in (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m); \varepsilon_1 \geq \varepsilon_2 \geq \ldots \geq \varepsilon_m \geq 0, m \leq n)$.

2. Related work

There are substantial explorations in the area of surface simplification and multiresolution surface reconstruction in the literature. Survey papers [7,17] extensively describe, classify and survey various approaches. The simplification and reconstruction techniques can be mainly categorized as: polygon merging [8], wavelets [13], volumetric approach [6], vertex clustering [18], energy function minimization [9,10], hierarchical representation, re-tiling [21], and decimation of vertices [20], edges or polygons. We herein just review some typical algorithms.

2.1 Terrain surface reconstruction

Terrain surface or a height field is a function of two coordinate variables $Z = H(x,y)$. Besides TIN and a regular grid mesh, terrain surface reconstruction can take a set of organized or unorganized sample points $(x,y,H(x, y))$ as input. Incremental insertion algorithm [5,14] in Figure 1 is the most widely used algorithm. This algorithm's advantage is of high speed. It has a cost of $O((n+m)\log m)$. Terrain surface reconstruction is actually 2½D surface modeling so the calculation of approximation error, simply the vertical distance between the input sample point and the approximated position, is much simpler than that of generic surface modeling.

Incremental insertion algorithm

```
project all points onto x-y plane;
construct two big triangles to
enclose all other points;
until (reach approximation index) {
  find the point introducing
  the maximum error measure;
  insert that point into
  the enclosing triangle;
  Delaunay retriangulation  on
  the new triangle mesh;
}
return the final TIN;
```

Figure 1: Incremental insertion algorithm

2.2 Vertex clustering

Vertex clustering [18] is the most efficient approach and the simplest algorithm for surface simplification. The steps of this algorithm are described in Figure 2. The vertex clustering approach 1) is simple and fast, 2) does not preserve the topology, 3) is a very crude approximation, 4) and handles both manifolds and nonmanifolds.

Vertex clustering algorithm

```
(1) construct a bounding box for
all vertices on the triangle mesh;
(2) regularly subdivide the
bounding box into smaller cells
(cell size depends on
the approximation error);
(3) associate each vertex with a
single cell which enclose it and
form all the vertices inside a
specific cell a cluster;
(4) collapse all vertices in
all clusters into a
representative vertices;
(5) remove triangles having more
than one vertex in a cluster and
return the new mesh;
```

Figure 2: Vertex clustering algorithm

2.3 Vertex decimation

W. J. Schroeder actually has already modified his original vertex decimation algorithm [20] which is just a surface simplification approach

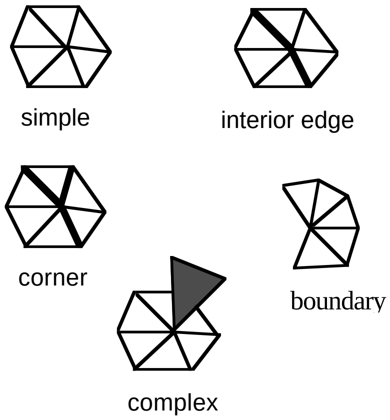

Figure 3: Vertex classification

and not designed for multiresolution surface reconstruction. The modified algorithm [19] has preserved the vertex classification, changed the vertex removal into edge collapse, and modified the simple local error calculation. The modified algorithm has become a progressive decimation algorithm. The steps for his original vertex decimation algorithm are: 1) Classify mesh vertices (see Figure 3), only vertices of class simple, boundary, and interior edge are valid candidates for removal. This classification will preserve the sharp features. 2) Calculate the local approximation error which is the distance to the local average plane or to the boundary line and remove the vertex introducing local approximation error less than the user-specified threshold. The simple local error calculation makes this algorithm fast. 3) Hole filling or retriangulate the created hole by recursive loop splitting algorithm.

The vertex decimation algorithm 1) is fast, 2) preserves sharp edges, 3) and has no global error guarantee, actually it is charaterized by local error accumulation.

2.4 Wavelets

The wavelets method [13] provides compact multiresolution representation and mesh editing functions. It is composed of three main steps: 1) partitioning with Delaunay triangulation; 2) parametrization; 3) resampling. The surface to be simplified has to be remeshed for wavelets decomposition so that it has a regular subdivision connectivity. This remeshing will introduce error in the highest level of details and then the simplification can not be lossless.

3. Simplification envelopes

The simplification envelopes method [3] forces the simplified mesh to lie within a user specifiable distance from the original mesh. The key advantages of this approach are: 1) a guarantee of global approximation error, 2) it preserves topology, 3) it gracefully preserves sharp features. The idea of the simplification envelopes could also be implemented for terrain reconstruction. For simplification envelopes *J. Cohen et. al* [3] construct two algorithms: the local algorithm (see Figure 4) and the global algorithm. The global algorithm has the advantage of removing multiple vertices in a single step which can avoid the "local minimum" problem. Although simplification envelopes as error bounds are perceptually correct, their local and global algorithms have the following drawbacks: 1) their hole filling will not produce optimized and unique retriangulation, 2) both algorithms are not incremental and can not reconstruct multiresolution surface continuously and automatically, 3) the global algorithm is too slow to reconstruct large models.

<u>local algorithm</u>

```
setup global error bound B and
construct simplification envelopes;
place all vertices in queue Q;
until (Q is empty) {
  pick one vertex i in Q;
  delete vertex i from the mesh;
  remove vertex i's adjacent edges
  and create a hole on the mesh;
  if (hole can be filled under B) {
    renew the mesh;
  }
  else {
    unchange the old mesh;
  }
  remove i from Q;
}
return the new mesh;
```

Figure 4: The local algorithm

4. Edge collapsing operation

Edge collapse operation takes the two endpoints of the target edge, contracts them to one of the two vertices, links all the incident edges to that

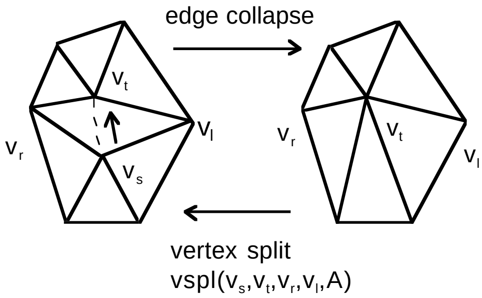

Figure 5: Edge collapse/vertex split

vertex and removes two or one triangles which have degenerated into lines or points. *H. Hoppe* [9,10,11], *W. J. Schroeder* [19] and *R. Ronfard et. al* [16] have used the edge collapse operation for multiresolution surface reconstruction. The fundamental algorithm for "progressive meshes" and "progressive decimation" is described in

```
        Fundamental algorithm
Reconstruction preprocessing:
until (reach approximation index) {
  find the edge whose collapse
  introduce the least error measure;
  collapse that edge into one vertex;
  save the collapse record including
  one collapsed  edge and two
  collapsed  triangles (vs,vt,vr,vl,A);
 }
return (base mesh & collapse records);
Display or reconstruction:
 based on the base mesh M0
 apply a sequence of vertex split
 operation using the saved records;
 generate a continuous family
 of meshes (M0,M1,...,Mn-1,Mn);
```

Figure 6: Fundamental algorithm

Because the edge collapse/vertex split operation in Figure 5 is invertible, *H. Hoppe* employs it to represent the original mesh M^n with the base mesh M^0 plus the sequence of vertex split records. That is to say, the vertex split records preserve the reverse order of the collapsing history. Therefore, *H. Hoppe and W. J. Schroeder*, starting from the base mesh M^0 and applying the sequence of vertex split operations, can reconstruct a continuos family of approximation meshes:

$$M^0 \rightarrow M^1 \rightarrow \ldots \rightarrow M^{n-1} \rightarrow M^n$$

Consequently, due to edge collapse operation, "progressive meshes" and "progressive decimation" has the following advantages: 1) "Progressive meshes" or representations of the original mesh M^n with the continuous family of meshes $(M^0,M^1,\ldots,M^{n-1},M^n)$ is space-efficient or has a smaller storage than the standard triangle mesh representation schemes; 2) Support smooth level-of-detail; The transformation from M^i mesh to M^{i+1} or the reverse can be completed just applying the *ith* vertex split/edge collapse operation; 3) Support progressive transmission; Reconstruct the original mesh from base mesh and the sequence of progressively transmitted vertex split records; 4) Support selective refinement or view-dependent refinement [11]. However, how to select the target edge for edge collapse? *H. Hoppe* defined a peculiar energy function for the error metric:

$$E(M) = E_{dist}(M)+E_{spring}(M)+E_{scalar}(M)+E_{disc}(M)$$

His energy function seems very practical and has considered comprehensive elements, but reasonless! *R. Ronfard's* LGE (local geometric error) and LTE (local tessellation error) [16] and *W. J. Schroeder's* modified local error metric both are not so perceptually correct.

5. Discussions

Should topology of the model be preserved during the simplification process? Most algorithms in the literature except vertex clustering [18] preserve topology. However, there is a trend to allow the topology change during simplification process [4,15]. For faster rendering and handling nonmanifolds, the edge collapse/vertex split transformations are replaced by the more general vertex unification/generalized vertex split ones in [15]. The vertex unification or contraction unifies or contracts an pair of vertices which are not necessarily connected by one edge or having neighbourhood relationship into one vertex. When non-edge pairs of vertices are unified or contracted, unconnected sections of the model become connected. The topology of the model will thus be changed. Although it is especially difficult to handle nonmanifolds while preserving topology, we still maintain that the topology should be adaptively preserved because topology is more important than overall appearance for human perception.

Should sharp surface features be preserved? Vertex decimation algorithm [19,20] deliberately preserve the sharp features by not removing those vertices which belong to the “corner” and “complex” vertex classes (see Figure 3). There are also other algorithms which deliberately keep sharp features by giving more weights to vertices on sharp edges and corners. We suppose that deliberately preserving sharp features is not a good idea and is not perceptually correct. However, the simplification envelopes algorithm [4] can gracefully preserve sharp features in fine-resolution simplified surfaces and removes them in the coarser resolution ones. So we believe that the sharp features should also be preserved adaptively rather than deliberately.

How to compare the approximation accuracy among the simplified surfaces generated by various simplification algorithms? There are only a few explorations [2,12] in the literature. *P. Cignoni et. al* [2] develop a tool - *“Metro”* to compare the simplified surfaces. However, we believe that constructing a least simplification envelope for the simplified surface is a better choice than the “point-surface distance” employed in *“Metro”*.

6. Our solution

The multiresolution surface reconstruction process can be divided into two stages. One is the error metric calculation or simplification target selection; the other is the decimating or collapsing stage. The simplification envelopes algorithm can preserve topology and sharp features adaptively because the constructed envelopes are a reasonable error metric. However, the hole creating and hole filling of the simplification envelopes in the decimating stage are not efficient at all. “Progressive meshes [10]” and “progressive decimation [19]” algorithms can reconstruct a continuous family of multiresolution surfaces efficiently and automatically because the edge collapse/vertex spilt transformations in the decimating stage are efficient and invertible. For their simplified surfaces, however, there are no guarantees for the global approximation errors. We believe that constructing simplification envelopes is reasonable for simplification target selection stage while the invertible edge collapse/vertex spilt transformations are suitable for decimating stage. We then combine the simplification envelopes and edge collapse operations to reconstruct multiresolution surfaces. The fundamental algorithm is described in Figure 7.

```
Fundamental Algorithm

Reconstruction preprocessing:

until (reach approximation index) {
  for (each edge i on the mesh) {
    try to collapse edge i and
    calculate its error metric Ri;
    put edge i with value
    Ri in List L;
  }
  sort List L and find the
  edge k whose collapse introduce
  the least error metric;
  collapse the edge k into
  a vertex and renew the mesh;
  save the collapse record
  including one collapsed edge
  and two collapsed triangles;
}
return (base mesh & collapse records);

Calculating error metric Ri:

for (each newly-created triangle t) {
  construct its simplification
  envelopes with minimum height Ht;
  put triangle t with height
  Ht in List L;
}
sort List L and find maximum height H;
return (Hmax);
```

Figure 7: Fundamental algorithm

How to calculate the error metric? For example, assume that we collapse edge FE into vertex E (see Figure 8). For every newly-created triangle

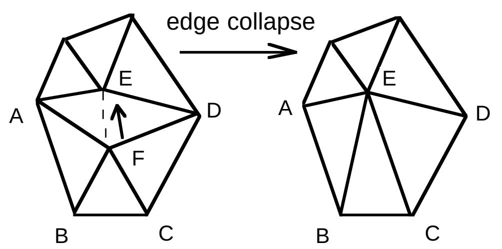

Figure 8: Newly-created triangles

(ΔABE, ΔBCE and ΔCDE), calculate the height H of its least simplification envelopes locally as indicated in Figure 9. The error metric R_{FE} for edge FE is: $R_{FE} = \max (H_{ABE}, H_{BCE}, H_{CDE})$

Our algorithm generates a continuous family of multiresolution surfaces with globally-guaranteed approximation errors

7. Conclusion

We have reviewed surface simplification and multiresolution surface reconstruction approaches. We maintain that the topology and sharp surface features of the model should be preserved adaptively. To compare the accuracy of reconstruction approaches, we also proposed a technique - constructing the least simplification envelopes for the simplified surfaces. Finally, we have described a multiresolution surface reconstruction algorithm which is a combination of the simplification envelopes and edge collapse/vertex split transformations.

8. Acknowledgment

We wish to thank Dr. Karsten Schlüns for his helpful discussions on our project.

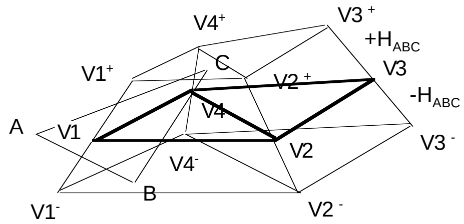

Figure 9: The least simplification envelopes construction for ΔABC

References

[1] P. Cignoni, E. Puppo, and R. Scopigno, “Representation and Visualization of Terrain Surfaces at Variable Resolution”, The Visual Computer, 13:(in press), 1997.

[2] P. Cignoni and C. Rocchini, “Metro: Measuring Error on Simplified Surfaces” , Technical Report B4-01-01-96, I.E.I.-C.N.R., Pisa, Italy, January 1996.

[3] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks and W. Wright, “Simplification Envelopes”, In SIGGRAPH '96, August, 1996.

[4] M. Garland and P. S. Heckbert, “Surface Simplification Using Quadric Error Metrics”, In SIGGRAPH '97 Proc., August 1997.

[5] M. Garland and P. S. Heckbert, “Fast Triangular Approximation of Terrains and Height Fields”, submitted for publication, In SIGGRAPH '97 Course Notes.

[6] T. He, L. Hong, A. Varshney, and S. Wang, “Controlled Topology Simplification”, IEEE Trans. on Visualization & Computer Graphics, 2(2), pp. 171-184, June 1996.

[7] P. S. Heckbert and M. Garland, “Survey of Polygonal Surface Simplification Algorithms”, Carnegie Mellon University technical report, 1997.

[8] P. Hinker and C. Hansen, “Geometric Optimization”, IEEE Visualization ‘93 Proceedings, pp. 189-195, San Jose, CA, October 1993.

[9] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Mesh optimization”, SIGGRAPH '93 Proc., pp. 19-26, Aug. 1993.

[10] H. Hoppe, “Progressive Meshes”, SIGGRAPH '96 Proc., pp. 99-108, August 1996.

[11] H. Hoppe, “View-Dependent Refinement of Progressive Meshes”, SIGGRAPH '97 Proc., Aug. 1997.

[12] R. Klette, “Sound analysis of 3D objects based on digitized data”, Technical Report CS-TR-140/CITR-TR-8, The University of Auckland, Auckland, February 1997.

[13] M. Lounsbery, T. D. DeRose, and J. Warren, “Multiresolution analysis for surfaces of arbitrary topological type”, ACM Transaction on Graphics, 16(1):34-73, 1997.

[14] T. Modth⌀, “Spatial Modeling by Delaunay Network of Two and Three Dimensions”, Dr. Ing. Thesis, Norwegian Institute of Technology, University of Trondheim, February 1993.

[15] J. Popovic and H. Hoppe, “Progressive Simplicial Complexes”, In SIGGRAPH ‘97 Proc., Aug. 1997.

[16] R. Ronfard and J. Rossignac, “Full-range approximation of triangulated polyhedra”, Computer Graphics Forum (Eurographics '96 Proc.), 15(3):67-76, 1996.

[17] J. Rossignac, “Geometric simplification and compression”, In SIGGRAPH '97 Course Notes.

[18] J. Rossignac and P. Borrel, “Multi-resolution 3D approximations for rendering complex scenes”, In Geometric Modeling in Computer Graphics, Springer Verlag, pp. 455-465, Italy, June 1993.

[19] W. J. Schroeder, “A Topology Modifying Progressive Decimation Algorithm”, submitted for publication, In SIGGRAPH ‘97 Course Notes.

[20] W. Schroeder, J. Zarge, and W. Lorensen, “Decimation of triangle meshes”, Computer Graphics, 26(2): 65-70, July 1992.

[21] G. Turk, Re-tiling polygonal surfaces, In Edwin E. Catmull, editor, ACM Computer Graphics (SIGGRAPH '92 Proceedings), volume 26, pp. 55-64, July 1992.