



A Multiphase Approach to Efficient Surface Simplification

Michael Garland*

Eric Shaffer†

University of Illinois at Urbana–Champaign

ABSTRACT

We present a new multiphase method for efficiently simplifying polygonal surface models of arbitrary size. It operates by combining an initial out-of-core uniform clustering phase with a subsequent in-core iterative edge contraction phase. These two phases are both driven by quadric error metrics, and quadrics are used to pass information about the original surface between phases. The result is a method that produces approximations of a quality comparable to quadric-based iterative edge contraction, but at a fraction of the cost in terms of running time and memory consumption.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—surface and object representations

Keywords: multiphase simplification, quadric error metrics, massive meshes, out-of-core simplification

1 INTRODUCTION

Whether arising from isosurface extraction or high-precision laser scanners, automatically generated polygonal surfaces are often very densely tessellated. Indeed, they are frequently unnecessarily detailed for a variety of applications, most notably interactive display. Over the last decade, several effective methods have been developed to automatically simplify polygonal surfaces, producing approximations that use far fewer triangles. Such simplification techniques have become a standard component of many surface acquisition systems.

Over time, the accuracy of scanning technology has improved dramatically, and the size of available polygonal meshes has risen at a similar pace. Most existing simplification methods are iterative in nature, and can only realistically process models up to 1–2 million triangles in size. Beyond this point, they tend to either be impractically slow or require unacceptable amounts of memory. However, extremely high-precision scans of objects at sub-millimeter accuracy are now available. At sizes that reach above 1 billion triangles, they are far beyond the capacity of iterative algorithms. The desire to be able to simplify such massive models has led to the development of out-of-core simplification algorithms based on single-pass spatial clustering rather than iterative deletion of surface primitives.

These are the two fairly distinct classes of simplification algorithms now in common use. Iterative methods generally produce high quality results, but are incapable of processing truly large models because of excessive time and space requirements. In contrast, one-pass spatial clustering methods can process models of any size, but the quality of the resulting approximations tends to decline quite rapidly as the desired output size decreases.

We propose a new multiphase simplification system that provides the benefits of both of these algorithm classes. It is able to

process models of arbitrary input size, like one-pass clustering, but is able to produce approximations of a quality comparable to iterative methods. In fact, our system produces results of comparable quality to the QSLim method [5] but in a fraction of the time and using substantially less memory — we have routinely been able to produce nearly identical approximations in $1/5$ the time or less. It achieves this level of performance by combining two distinct simplification passes: an initial spatial clustering phase, followed by an iterative edge contraction phase. These two phases are coupled together by passing quadric error information between them. The resulting system, based on well-understood techniques, is quite simple, can be implemented with little effort, and provides an excellent balance of efficiency and quality.

2 BACKGROUND

Among the earliest of simplification methods is the *uniform vertex clustering* algorithm developed by Rossignac and Borrel [17]. The bounding box of the model is subdivided on a regular grid, and all vertices within a single cell are unified into a single representative vertex. This early algorithm was later generalized to use adaptive partitioning schemes such as octrees [13] and collections of spheres [11]. The greatest advantage of such clustering methods is their efficiency. They are typically extremely fast, although the quality of the resulting approximations degrades rapidly with decreasing output size. Clustering algorithms can also be designed to process input models by making a single linear scan through the data, and can have memory requirements that are independent of the input size. These properties make them a natural choice for out-of-core processing of extremely large meshes [8, 9]. Combining one-pass uniform clustering with a second adaptive BSP pass results in a slower but generally more accurate out-of-core method [18].

One of the most widely used classes of simplification algorithms is that based on *iterative edge contraction*. Such algorithms begin by ranking all edges of the model by some error metric, and proceed by repeatedly collapsing the minimum cost edge. It is also possible to connect topologically separate components by allowing the underlying algorithm to consider “virtual edges” that connect arbitrary pairs of vertices [14, 5]. Hoppe’s algorithm for constructing progressive meshes [6] generally produces high quality results, but also has relatively high time and space requirements. This method can be adapted for use on very large models by segmenting the original data into tiles, processing them independently, and merging partially simplified tiles [7, 16]. The quadric-based simplification algorithm developed by Garland and Heckbert [5] tends to provide a good compromise between output quality and computation cost, producing generally good approximations in short amounts of time. Lindstrom and Turk’s “memoryless” algorithm [10] is closely related to the quadric-based method, but consumes less memory at the cost of somewhat greater running time. El-Sana and Chiang [2] developed an iterative approach to processing very large models by combining data segmentation with out-of-core data structures.

These are the methods most directly related to our new multiphase algorithm. More complete details on existing simplification methods can be found in several published surveys [1, 3, 12].

*garland@cs.uiuc.edu

†shaffer1@cs.uiuc.edu

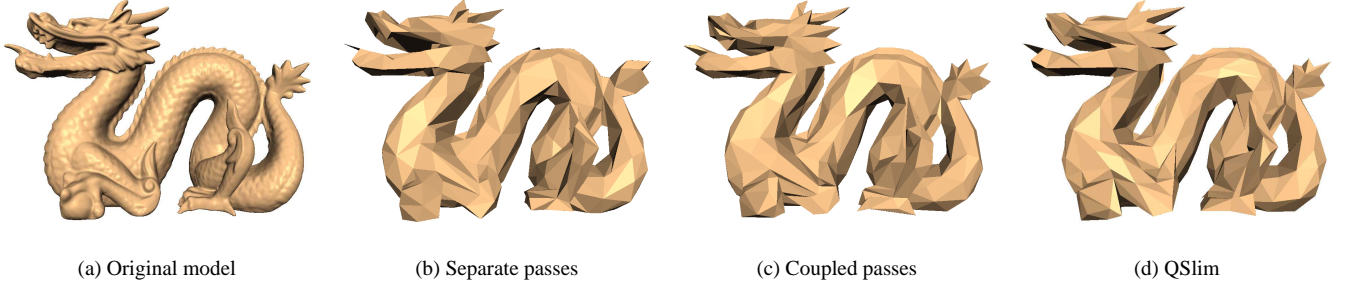


Figure 1: An 870,000 triangle model is reduced to 1000 faces by (b) two separate simplification passes and (c) two passes coupled by quadrics. Note the greater fidelity resulting from our coupled multiphase system (e.g., around the tail), comparable to the (d) QSLim results.

2.1 Quadric Error Metrics

Our approach to the simplification problem is based on the quadric error metric [5], which we briefly summarize. A given triangle with unit normal vector \mathbf{n} defines a unique plane $\mathbf{n}^T \mathbf{v} + d = 0$. We can define a *weighted quadric* Q for which $Q(\mathbf{v})$ is the squared distance of the point \mathbf{v} to this plane, scaled by the weight ω :

$$Q = (\omega \mathbf{A}, \omega \mathbf{b}, \omega c) = (\omega \mathbf{n} \mathbf{n}^T, \omega d \mathbf{n}, \omega d^2) \quad (1)$$

$$Q(\mathbf{v}) = \omega(\mathbf{v}^T \mathbf{A} \mathbf{v} + 2 \mathbf{b}^T \mathbf{v} + c) \quad (2)$$

This is the *fundamental quadric* for the given triangle. The fundamental quadric of a vertex is the sum of the fundamental quadrics of its adjacent faces. The weighted sum of squared distances of a point \mathbf{v} to the planes defined by a set of triangles $\{f_j\}$ can be evaluated by summing the quadrics for each plane:

$$\sum_j Q_j(\mathbf{v}) = (\sum_j Q_j)(\mathbf{v}) \quad (3)$$

We choose the weights ω_j to be $1/3$ the areas of the corresponding triangles in order to achieve invariance of the error metric over all geometrically identical tessellations of the surface [4].

The point \mathbf{v}^* for which $Q(\mathbf{v})$ is minimal is the solution to the linear system

$$\mathbf{A} \mathbf{v}^* = -\mathbf{b} \quad (4)$$

Since the matrix \mathbf{A} is positive semidefinite, Cholesky decomposition is the preferred approach to solving this linear system [15].

3 MULTIPHASE SIMPLIFICATION

Experience has shown that different approaches to simplification are often suitable for varying simplification tasks. For example, uniform vertex clustering is suitable when time and space efficiency are the paramount concern and the desired level of simplification is not too aggressive. In contrast, iterative contraction algorithms tend to be slower and consume more memory, but produce much higher quality results, particularly as the desired output size shrinks. Our goal is to produce a single unified scheme for efficiently processing models of any size, while maintaining the ability to produce high quality approximations.

3.1 Overview

The idea behind our multiphase simplification system is quite simple. We combine a uniform clustering phase and an iterative contraction phase in a simplification pipeline. The structure of the system can be quickly summarized:

1. Perform uniform clustering on input model (of size n), producing an intermediate approximation (of size r) with 1 quadric per vertex.
2. Beginning with this intermediate approximation *and* its associated quadrics, perform iterative edge contraction.

Notice that this is not equivalent to merely passing the input model through a clustering step, writing an intermediate approximation, and then passing that model through an iterative contraction step. The crucial difference is that our system couples successive phases together by passing quadrics accumulated in one phase onto the next.

While a simple idea, this use of quadrics to pass information between phases has a very significant practical impact. Figure 1 shows the results of simplifying a dragon model with (b) two separate passes and (c) two coupled passes. Qualitatively, important details of the surface (e.g., the tail and mouth) are preserved much better by our coupled multiphase approach. The mean squared error is also about 50% lower than that of the model produced by two separate passes. Indeed, it is nearly identical in quality to (d) the approximation produced by QSLim alone. But whereas QSLim requires $O(n \log n)$ time and $O(n)$ space, our system requires only $O(n + r \log r)$ time and $O(r)$ space.

3.2 Phase I: Uniform Clustering

The first phase of the system is quite similar to Lindstrom’s OoCS method [8]. Briefly, we begin by subdividing the axis-aligned bounding box of the input on a regular grid¹. Within each voxel of this grid, we maintain a cumulative quadric and mean vertex position. Each triangle of the input is scanned, and its fundamental quadric is derived as in Equation 1. This fundamental quadric is added into the cumulative quadrics for the (up to) three voxels containing the corners of the triangle. The output triangles are exactly those whose corners fall in three different voxels. Following the approach proposed by Lindstrom and Silva [9], we also accumulate edge normals to help preserve open boundaries. Once all input triangles have been scanned, we can compute a representative vertex location \mathbf{v}^* for each voxel using Equation 4, or the mean vertex position if \mathbf{A} is singular. The choice of position here is not terribly important, as the next phase will derive its own vertex positions. Only along boundary curves, where Phase II will reference the intermediate geometry to construct boundary constraints, does our choice of intermediate vertex location effect the final approximation.

¹Only a small fraction of the many voxels are typically occupied. Sparse grid data structures are essential for good performance.

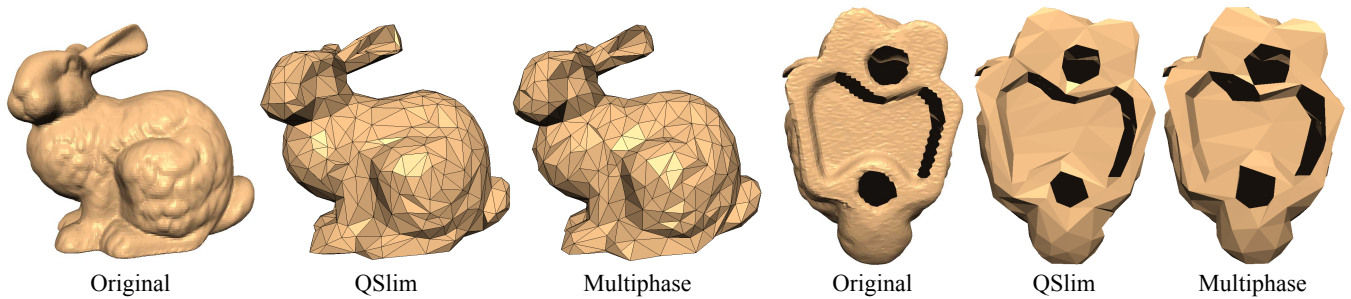


Figure 2: Multiphase results nearly identical to those of QSlim (except on boundaries), but generated in $1/5$ the time.

This initial phase produces an intermediate approximation and a set of quadrics. Each grid voxel containing one or more input vertices produces exactly one intermediate vertex. Associated with each intermediate vertex is exactly one quadric which is the sum of the fundamental quadrics of all the input vertices contained in the corresponding voxel. This is the only data passed to the following phase. All other data, such as mean vertex position and cumulative edge normals are discarded.

3.3 Phase II: Iterative Contraction

The standard quadric-based contraction algorithm QSlim [5] is divided into two fundamental stages: (1) *initialization* assigns quadrics to input vertices and (2) *iteration* continues contracting the current edge of least cost until the desired output size is reached. Our system uses the standard iteration stage unaltered, but it is preceded by a new initialization stage that accepts intermediate results from Phase I.

Initialization of the iterative phase would normally involve constructing fundamental quadrics for each vertex from the planes of its adjacent faces. But instead of constructing quadrics from the geometry of the intermediate approximation, we merely use the quadrics accumulated during the previous phase of the system. This means that we continue to use quadrics derived from the *input geometry*. Provided that the grid used in Phase I was suitably fine, these quadrics will reliably characterize the local shape of the original surface. Having assigned these basic quadrics to each vertex, we accumulate additional constraint quadrics to preserve boundaries [5]. The rest of the simplification algorithm proceeds as normal, producing the final approximation.

3.4 Selecting Grid Resolution

In order for this system to perform well, it is of course important that the resolution of the voxel grid used in Phase I be chosen appropriately. As a general rule of thumb, we have observed that the grid resolution should be chosen so that Phase I produces at least 4 times more intermediate vertices than desired output vertices. While the user could simply specify the dimensions of the grid directly, we recommend two general strategies for automatically selecting an appropriate resolution.

The first alternative is for the user to specify a desired memory usage limit for the Phase I grid. Since we use a sparse grid data structure, we cannot guarantee absolute memory limits *a priori* as the memory required will depend on the number of occupied voxels. While we cannot determine this exactly, we can in fact estimate it with reasonable reliability by sampling the input vertex density on a fixed size grid. This estimation comes at the price of an additional scan over the data, but it provides the user with a useful

level of control over the storage requirements of the simplification system. A user particularly concerned with the quality of the output could thus select an extremely fine resolution by allowing the system to consume a large fraction of all available memory.

A second approach to selecting the grid resolution is to directly specify the size of an individual voxel. This is currently our preferred approach, because it provides a guaranteed bound on the Hausdorff error introduced during Phase I. When clustering on a grid with voxel diameter d , no output vertex is more than d away from an input vertex.

3.5 Discussion

Our primary motivation for coupling quadric-based iterative contraction with a uniform clustering preprocess was to produce high quality approximations of arbitrarily large input meshes. However, there are some additional advantages to this initial clustering step.

First and foremost, is the increased speed and decreased memory consumption of the overall system. Given that quality of the multiphase output is comparable to QSlim, this multiphase system can be used as a much more efficient replacement for QSlim even on meshes of moderate sizes. Unlike most iterative contraction methods, the multiphase system can easily process models where coincident vertices are duplicated (so called “triangle soups”), as the initial clustering will unify all duplicated vertices. It can also remove small topological features, which we have found works at least as well as, if not better than, iterative contraction using additional virtual edges.

Unfortunately, we cannot exercise much control over these topological modifications. This points to the single greatest disadvantage of the initial clustering pass: we cannot guarantee topological preservation. The use of a fixed voxel grid also results in the output approximation being sensitive to translation/rotation of the input mesh.

4 RESULTS

Our experiments have shown that our multiphase simplification system performs very well. We have compared its performance with our own implementations of uniform out-of-core clustering (OoCS) [8], adaptive out-of-core clustering (BSP) [18], and quadric-based contraction (QSlim) [5]. Multiphase simplification is able to simplify models of arbitrary size, like OoCS, but is able to generate much higher quality approximations at moderate to small output sizes. Indeed, it consistently produces approximations of quality comparable to QSlim, but using considerably less running time and memory, both asymptotically and in practice.

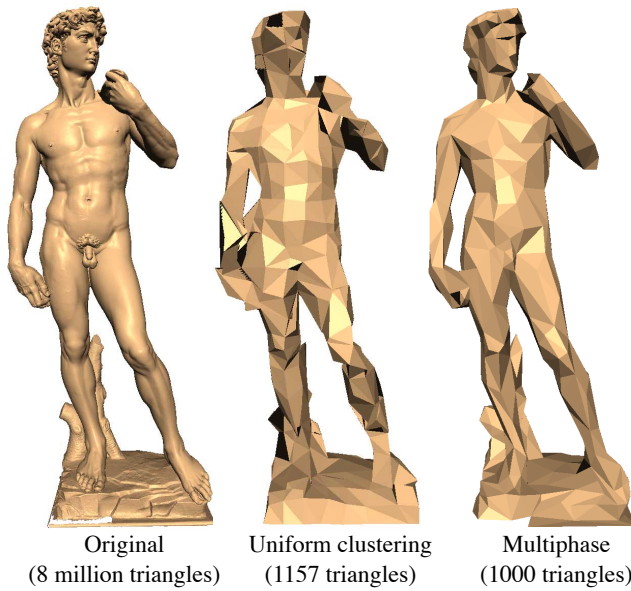


Figure 3: The multiphase system clearly outperforms uniform clustering when aggressively simplifying large meshes.

All performance tests were run on an 1 GHz Pentium III system with 4 GB of RAM and running a Linux kernel. Our implementations all share a common code base, including identical implementations of the quadric error metric and iterative edge contraction. Our QSlim software uses an ASCII input format, while the others use memory-mapped binary files. For this reason, all running times exclude file I/O time.

Figure 2 demonstrates the comparative performance of our multiphase system with the iterative QSlim algorithm. Both systems were used to produce a 1000 face approximation of the 69,451 face original. The multiphase system produced an intermediate approximation of 9127 faces using a $40 \times 40 \times 31$ grid. Visually, the quality of the approximations is very similar. Rather surprisingly, the mean squared error of the multiphase approximation is actually 40% lower. And the multiphase system produced this approximation 4.5 times faster than the QSlim system. The only significant difference in the quality of the results is in the preservation of the boundary curves along the base of the model. Because it is forced to use a weaker heuristic, the initial clustering phase does not preserve the open boundary as well as iterative contraction.

Figure 3 illustrates the simplification of a moderately large input model: Michelangelo's David scanned at 2mm resolution, containing 8 million triangles. The out-of-core uniform clustering system used a $11 \times 28 \times 6$ grid, and produced an approximation with 1157 faces. The multiphase system produced an intermediate approximation of 19,000 faces using a $40 \times 101 \times 23$ grid. As we would expect from uniform clustering at this aggressive level, the quality of the approximation has deteriorated severely. The topology of the surface has been adversely effected (e.g., the arm joining the hip) and various features such as the arm have been seriously distorted. The black regions (e.g., atop the head) are areas where part of the surface has been turned inside out. The rough outlines of the regular grid are also apparent in the triangulation, particularly on the stomach. In contrast, the results from the multiphase system have preserved the surface details of the original to a much greater extent. In particular, notice that the eye brow, hair line, nose, and other facial features are still quite identifiable. This very substan-

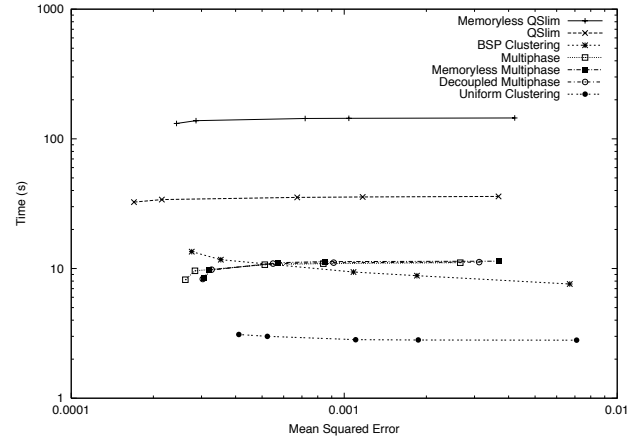


Figure 4: Tradeoff of running time vs. error for the dragon model.

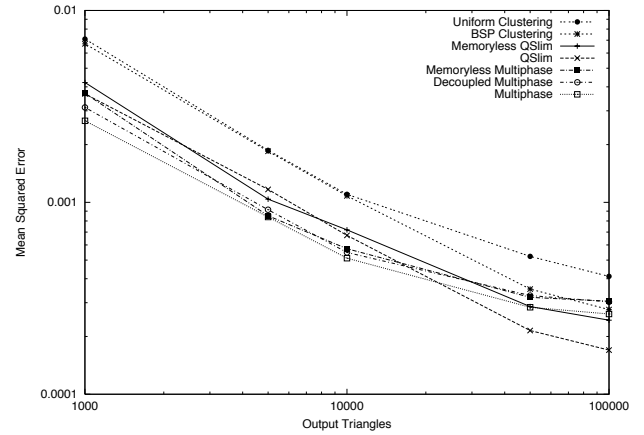


Figure 5: Approximation error as a function of output size for the dragon model.

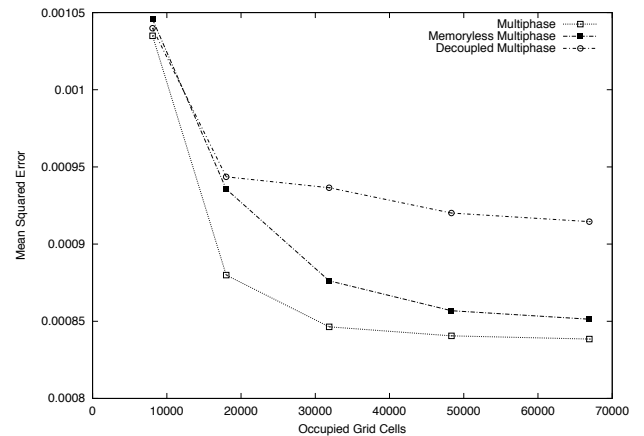


Figure 6: Effect of grid size on approximation error.

Dimensions	Number of Cells		Occupancy Rate
	Total	Occupied	
$60 \times 42 \times 27$	68,040	8073	11.9%
$90 \times 63 \times 40$	226,800	17,999	7.9%
$120 \times 85 \times 54$	550,800	31,869	5.9%
$150 \times 106 \times 67$	1,065,300	48,349	4.5%
$180 \times 127 \times 80$	1,828,800	66,920	3.7%

Table 1: Grid sizes used for results in Figure 6.

tial increase in quality was achieved with a fairly modest increase in running time. Multiphase simplification required a total of 43 seconds, as compared to 19 seconds for uniform clustering.

For further comparison, Figures 4 and 5 present the results of running all four simplification systems on the dragon pictured in Figure 1. Our QSlim implementation supports both the original Garland–Heckbert algorithm as well as a “memoryless” variant inspired by the Lindstrom–Turk algorithm. These implementations are equivalent except that the standard form accumulates quadrics at each step and the memoryless form recomputes quadrics at each step. We have also tested three variants of the multiphase algorithm: (1) the standard multiphase algorithm presented in Section 3 (2) separate de-coupled passes and (3) de-coupled passes where the second pass is a memoryless form of iterative contraction. All multiphase trials in these figures were run with a fixed grid size of $180 \times 127 \times 80$. Approximations for the 800,000 face original were produced at the 100,000, 50,000, 10,000, 5000, and 1000 face levels.

Figure 4 shows the running times of the simplification systems plotted as a function of the mean squared error of the approximations generated. As expected, both QSlim variants were substantially slower than the other methods. Uniform clustering was the fastest, with the multiphase method roughly twice as slow. However, the error of the approximations produced by the multiphase system are consistently lower. Surprisingly, it produces higher quality approximations even than the QSlim system at lower output sizes. In contrast, the purely clustering-based systems are producing approximations with more than an order of magnitude greater error by the time we reach the 1000 face level.

Figure 5 demonstrates in greater detail the output quality of the various methods. The pure clustering methods produce output with substantially higher error than all other methods. The quality of the approximations generated by all three multiphase variants are all comparable to the results produced by the QSlim variants, and even produce lower total error at lower output resolutions. Of the three multiphase variants, the coupled variant that we have proposed consistently produces results with the lowest total error. On average, our proposed method produced results with 20–30% lower error than the two de-coupled variants. We have observed that at very fine grid resolutions, the difference between coupled and de-coupled multiphase methods becomes fairly negligible. This is what we would expect, since as the grid resolution increases, more and more of the simplification work is actually being done in Phase II. Similarly, as the grid used becomes coarser, the performance gap between the coupled and de-coupled methods grows significantly, until the coupled version produces errors which are a factor of 2–3 below that of the de-coupled version.

The choice of grid resolution during Phase I is one of the most significant factors determining the quality of the approximations generated by our multiphase system. To help quantify its impact, we generated several 5000 face approximations of the dragon model and measured the error of the resulting output. The specific grid sizes used are shown in Table 1, and the results of the experiment

are shown in Figure 6. As noted above, we see that phase coupling significantly reduces approximation error, except at very fine and very coarse grid resolutions. We also see that the memoryless variant consistently produces less accurate approximations at moderate resolutions. And as we would expect, there is some point beyond which error grows very rapidly as the grid resolution is decreased, ultimately reaching the much higher error rate characteristic of pure uniform clustering.

Figure 7 provides a more qualitative demonstration of the effect of Phase I grid resolution on the quality of the final result. The original input model has roughly 28 million triangular faces, and we produced several 5000 face approximations using our multiphase system. Each approximation is labeled by the Phase I grid size and the number of triangles in the resulting intermediate approximation. The rightmost approximation corresponds to uniform clustering, as the Phase I grid has already simplified the input beyond the target size. For each subsequent approximation, the Phase I dimensions were doubled along each axis. As expected, increasing the grid size does indeed produce better results, and larger intermediate models. We also see the diminishing returns of increasing grid size. Increasing the grid size beyond that shown here yields extremely small incremental improvements. Total simplification times ranged from 145 seconds for the coarsest grid to 160 seconds for the finest grid. This relatively narrow variation in running time, as compared with the variation in grid sizes, is because for data of this size — 500 MB in this particular case — the running time is largely dominated by the cost of simply scanning through the data stream.

We conclude by looking at the performance of our system on a truly large surface model. The model of Michelangelo’s St. Matthew shown in Figure 8 was scanned at 0.25mm resolution, contains roughly 372 million triangles, and is over 6.5 GB in size when uncompressed. The three approximations shown in Figure 8 were generated in approximately 47 minutes using a constant intermediate grid size of $300 \times 847 \times 273$. The first approximation, containing 1 million triangles, has lost very little of the detail of the original, and our system continues to produce high-fidelity approximations at an output size of 100,000 triangles. Even with only 10,000 output triangles, the approximation preserves the overall shape of the statue quite well. Note that the holes apparent here are actually preserving holes present in the input data, although they have grown in size.

The images shown in Figure 9 illustrate the performance of our system on the St. Matthew model in closer detail. The primary effect of reducing the model from 372 million to 1 million triangles has been to remove the very fine grained chisel marks that cover the surface, hence the smoother look of the leg in the reduced model. Even with 100,000 output triangles, the shape of the surface is largely intact. In particular, notice that the contours of the clothing and the indentation above the knee are represented reasonably accurately. Finally, with 10,000 output triangles, a substantial amount of the original detail has been removed. Nevertheless, the contours of the leg and the clothing surrounding it remain easily identifiable.

5 CONCLUSION

We have presented a new multiphase method for efficiently producing high quality approximations of polygonal models of arbitrary size. By combining one-pass spatial clustering with iterative edge contraction, it provides the benefits of both methods. It can process models far larger than iterative methods are capable of, and produces higher quality approximations than pure clustering methods, especially at small output sizes. The two phases of the system are coupled together by attaching quadric error data to the intermediate model passed between them. Just as quadrics have proven effective at driving iterative edge contraction and spatial clustering

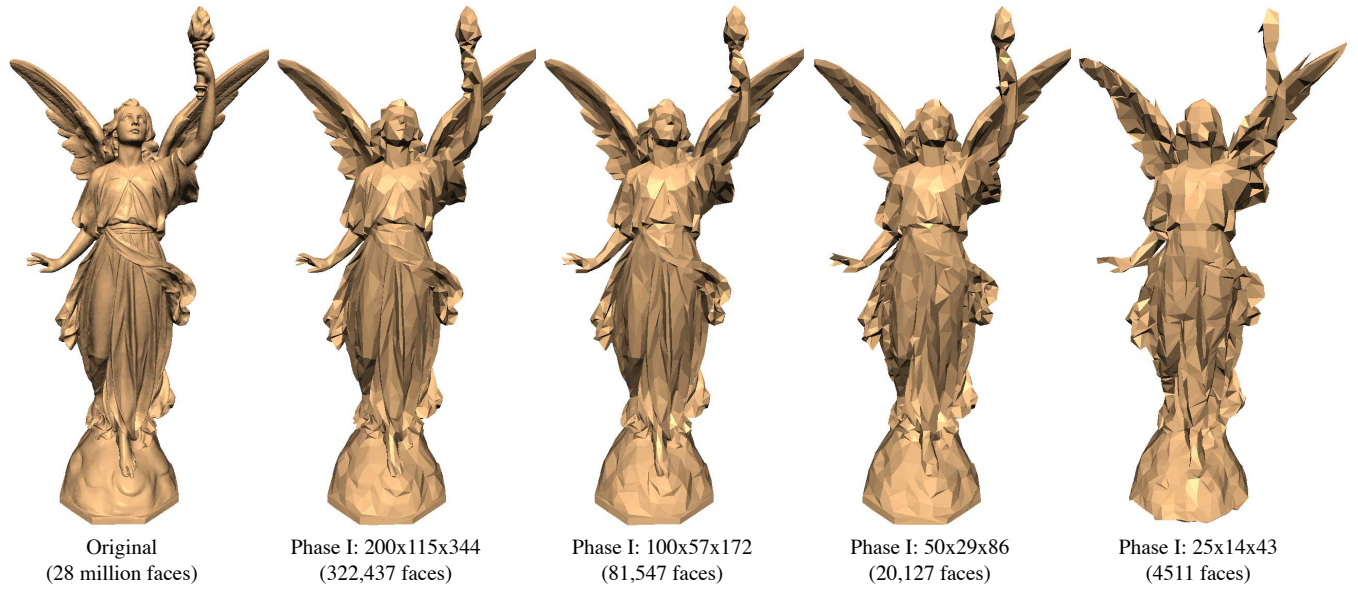


Figure 7: Simplification to a target output size of 5000 faces using different Phase I grid resolutions. Model sizes shown are for intermediate models; the rightmost model is already beyond the target size, so this is also the output size.

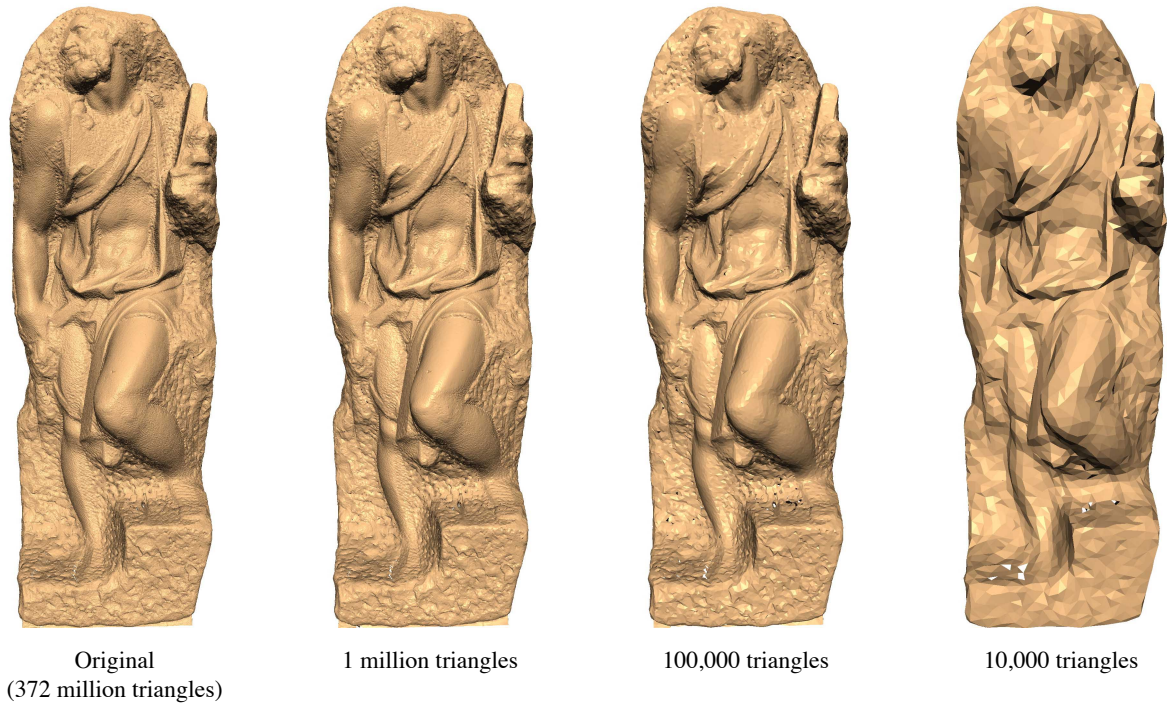


Figure 8: Multiple approximations of a very large (6.5 GB) surface model using a constant Phase I grid resolution of $300 \times 847 \times 273$.

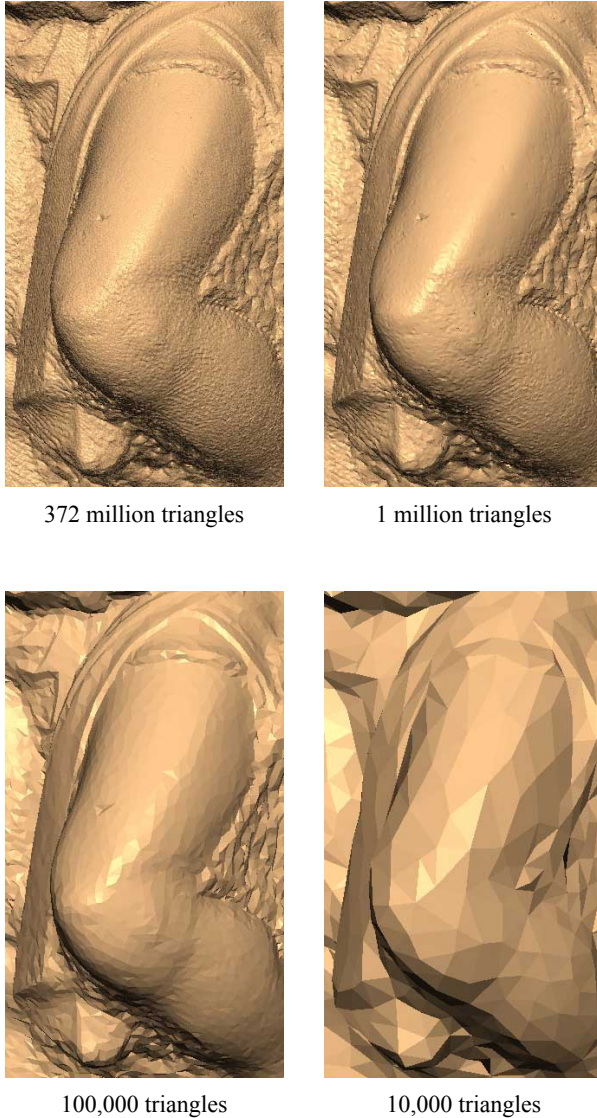


Figure 9: Close-up view of models shown in Figure 8. Note the preservation of surface details even after extremely aggressive reduction.

separately, they are also an effective means of communicating information between these different simplification frameworks. We believe that this new algorithm is truly general purpose, in that it can be applied to inputs of any size, from a few thousand to a few billion faces, and can successfully produce high quality approximations in fairly little time.

While the current system has performed quite well, there are a number of areas for future work that could potentially improve its performance appreciably. Currently, the choice of grid resolution for the clustering phase is completely fixed, either by memory limit or voxel size. It would be desirable to have a method that could automatically select an adaptive grid size based on the structure of the input surface itself. It is also apparent that our proposed multiphase system is only a single instance of a much more general system where multiple simplification phases are combined in a pipelined fashion. Many alternative designs for the individual stages could be explored, but we believe that it would be most productive to investigate alternative adaptive clustering strategies, such as octrees or BSP trees. Finally, we were quite surprised to find that the multiphase algorithm frequently produced results with error lower than the results of the fully iterative QSLim method. This rather counter-intuitive result deserves more careful scrutiny.

6 ACKNOWLEDGEMENTS

We would like to thank the Stanford Graphics Lab and the Digital Michelangelo Project for providing the surface models shown in this paper. This work was funded in part by the National Science Foundation under grant CCR-0098170.

REFERENCES

- [1] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computers & Graphics*, 22(1):37–54, 1998.
- [2] Jihad El-Sana and Yi-Jen Chiang. External memory view-dependent simplification. *Computer Graphics Forum*, 19(3):139–150, August 2000.
- [3] Michael Garland. Multiresolution modeling: Survey & future opportunities. In *State of the Art Report*, pages 111–131. Eurographics, September 1999.
- [4] Michael Garland. *Quadric-Based Polygonal Surface Simplification*. PhD thesis, Carnegie Mellon University, CS Dept., 1999. Tech. Rept. CMU-CS-99-105.
- [5] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of SIGGRAPH 97*, pages 209–216. ACM SIGGRAPH, August 1997.
- [6] Hugues Hoppe. Progressive meshes. In *Proceedings of SIGGRAPH 96*, pages 99–108. ACM SIGGRAPH, August 1996.
- [7] Hugues Hoppe. Smooth view-dependent level-of-detail control and its application to terrain rendering. In *IEEE Visualization 98 Conference Proceedings*, pages 35–42, 516, Oct 1998.
- [8] Peter Lindstrom. Out-of-core simplification of large polygonal models. *Proceedings of SIGGRAPH 2000*, pages 259–262, July 2000.
- [9] Peter Lindstrom and Cláudio T. Silva. A memory insensitive technique for large model simplification. In *Proceedings of IEEE Visualization 2001*, pages 121–126, October 2001.

- [10] Peter Lindstrom and Greg Turk. Fast and memory efficient polygonal simplification. In *Proceedings of IEEE Visualization 98*, pages 279–286, Oct 1998.
- [11] Kok-Lim Low and Tiow-Seng Tan. Model simplification using vertex-clustering. In *1997 Symposium on Interactive 3D Graphics*. ACM SIGGRAPH, 1997. <http://www.iscs.nus.sg/~tants/>.
- [12] David Luebke, Jonathan Cohen, Martin Reddy, Amitabh Varshney, and Benjamin Watson. *Advanced Issues in Level of Detail*. Number 45 in SIGGRAPH 2001 Course Notes. ACM SIGGRAPH, August 2001.
- [13] David Luebke and Carl Erikson. View-dependent simplification of arbitrary polygonal environments. In *Proceedings of SIGGRAPH 97*, pages 199–208. ACM SIGGRAPH, August 1997.
- [14] Jovan Popović and Hugues Hoppe. Progressive simplicial complexes. In *Proceedings of SIGGRAPH 97*, pages 217–224. ACM SIGGRAPH, 1997.
- [15] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Second edition, 1992.
- [16] Chris Prince. Progressive meshes for large models of arbitrary topology. Master’s thesis, University of Washington, 2000.
- [17] Jarek Rossignac and Paul Borrel. Multi-resolution 3D approximations for rendering complex scenes. In B. Falcidieno and T. Kunii, editors, *Modeling in Computer Graphics: Methods and Applications*, pages 455–465, 1993.
- [18] Eric Shaffer and Michael Garland. Efficient adaptive simplification of massive meshes. In *Proceedings of IEEE Visualization 2001*, pages 127–134, October 2001.