Voxel Based Object Simplification

Taosong He, Lichan Hong, Arie Kaufman, Amitabh Varshney, and Sidney Wang

Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11794-4400

Abstract

We present a simple, robust, and practical method for object simplification for applications where gradual elimination of high frequency details is desired. This is accomplished by sampling and low-pass filtering the object into multi-resolution volume buffers and applying the marching cubes algorithm to generate a multi-resolution triangle-mesh hierarchy. Our method simplifies the genus of objects and can also help existing object simplification algorithms achieve better results. At each level of detail a multi-layered mesh can be used for an optional and efficient antialiased rendering.

1. Introduction

Automatic generation of multi-resolution object hierarchies has become a crucial process for reconciling scene realism with interactivity through level-of-detail-based rendering [3, 4, 7]. The basic idea in a level-of-detail-based rendering scheme is to use the perceptual importance of a given object in the scene to select its appropriate level-of-detail representation. Thus, higher detail representations are used when the object is perceptually more important and lower detail representations are used when the object is perceptually less significant. This method allows one to achieve higher frame update rates while maintaining good visual realism.

Research on automatically generating multi-resolution object hierarchies has proliferated in the last few years [5, 6, 8, 12, 14-16]. A constraint common to most existing work has been the genus preservation criterion. Our work, on the other hand, provides a method to simplify the genus of an object in a *controlled* fashion. Preservation of topology (or the genus) is crucial for certain applications such as molecular surface modeling, where the presence (or absence) of interior tunnels and cavities in a molecule conveys important structural and chemical information to the biochemist. Clearly, if the target application demands topology preservation, then the simplification algorithm should adhere to it. However, if the goal is fast and realistic rendering, such as for virtual reality or some other time critical applications, topology preservation criterion could stand in the way of efficient simplification.

Let us consider a virtual flythrough in a CAD model. A tiny hole on the surface of a mechanical part in this model will gradually disappear as the observer is moving away from the part. However, genus preserving simplification of this object will retain such features, thereby reducing frame rates (due to limits on the amount of geometry-simplification one can achieve while preserving topology) and increasing image-space aliasing (due to undersampling, especially in perspective viewing). This idea has also been demonstrated in [4], where a chair has been shown at three levels of detail with no preservation of the topology across them.

Object simplification process can be viewed as consisting of the following two stages:
(a) geometry simplification, in which the number of vertices, edges, and faces is reduced; and
(b) genus simplification, in which the number of holes, tunnels, and cavities is reduced.
Depending upon the target application, these two stages can be performed either independently or jointly. For example, given a certain criterion of genus simplification, geometry simplification can be applied on the genus-simplified objects to further reduce their complexity. Most of the existing work in the area of object simplification deals exclusively with stage (a) above, and the extension to stage (b) is usually difficult and complicated. The goal of our research is to address stage (b) in a simple and robust way, and thereby also help the current geometry simplification algorithms to achieve better results for certain applications.

In this paper we present a method for the generation of multi-resolution hierarchies with gradual elimination of high-frequency features including, but not limited to, tiny holes, tunnels, and cavities. An interesting feature of our approach is that it can simplify not only individual objects, but also collections of objects. In our approach the input object is sampled and low-pass filtered into a three-dimensional volume buffer of uniform voxels, a process referred to as volume sampling by Wang and Kaufman [17]. Then a triangle mesh fitting technique, such as the marching cubes, is used on the volume buffer of filtered sample points to produce a low-pass-filtered mesh. By

(See color plates, page CP-35)

simply adjusting the size of each voxel, thereby adjusting the resolution of the volume buffer, the desired level of detail can be achieved, and consequently, a multi-resolution hierarchy of triangle meshes can be generated.

A significant amount of object-space aliasing can be eliminated by removing the high-frequency components using the object simplification technique outlined above. To further reduce the image-space aliasing during rendering, we have developed a multi-layered triangle mesh rendering algorithm. Our idea is to smooth out the transition between the boundary of an object and empty space surrounding it by using multiple layers of triangle meshes with increasing translucency from the innermost layer to the outermost one. Unlike the earlier antialiasing techniques presented in [1, 2], the prefiltering of the projected objects in image-space is replaced by a view-independent filtering in object-space, which is performed only once in a pre-rendering stage. Our object-space antialiasing approach involves rendering of translucent polygons, which could be more expensive than traditional image-space antialiasing on some platforms. However, it has the advantage of higher accuracy.

The rest of the paper is organized as follows. We outline our object simplification process in Section 2 and the multi-layered marching cubes algorithm for antialiasing in Section 3. We have implemented our algorithm and tested it on several kinds of objects and summarize our results in Section 4. Our conclusions and some ideas on future work appear in Section 5.

2. Object Simplification

In this section, we adopt a signal-processing approach to object simplification by sampling the input object and using low-pass filtering to gradually remove the high frequencies (i.e., detailed features) of the object. The class of input objects that our algorithm can accept and process includes polygonal meshes, volume datasets, objects derived from range-scanners, and algebraic mathematical functions such as fractals. Since our algorithm adopts a signal-processing approach to object simplification, it cannot output infinitely high frequencies, such as those introduced by sharp edges. Thus, although it generates reasonable results for all classes of objects outlined above, it works best for objects that represent volumes with no sharp discontinuities.

Our simplification algorithm starts by first overlaying the object with a three-dimensional grid and applying a low-pass filter at each grid point. A three-dimensional volume buffer data-structure is used to store these filtered grid-point values. That is, for a grid point (i, j, k) in the volume buffer, the resulting filtered density $f(i, j, k)$ is calculated as:

$$f(i, j, k) = \int\int\int h(i-\alpha, j-\beta, k-\gamma)\, S(\alpha, \beta, \gamma)\, d\alpha\, d\beta\, d\gamma \quad (1)$$

where h is a radially symmetric low-pass filter and $S(\alpha, \beta, \gamma)$ is a binary function defined as:

$$S(\alpha, \beta, \gamma) = \begin{cases} 1 & \text{if point } (\alpha, \beta, \gamma) \in \text{object} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Once the filtering and sampling process is completed, a polygon mesh fitting technique is employed to produce the detail-eliminated polygon mesh from the set of filtered sample points represented in the volume buffer.

Theoretically, high frequencies that exceed the Nyquist frequency of the volume raster can be filtered out by applying an ideal low-pass filter (*sinc*) with infinite support. In practice, this ideal low-pass filter is always approximated by filters with finite support. For volume modeling of objects, lower sampling resolution of the volume raster corresponds to lower Nyquist frequency, and therefore requires a low-pass filter with wider support for a good approximation. This direct correspondence between the size of the filter support and the resolution of the volume raster leads to a hierarchical representation of the model. The base of the proposed hierarchy contains the most detailed and the highest resolution of the object. As one moves up the hierarchy, low-pass filters with wider support are applied, and the top contains the blurriest low-resolution version of the objects.

At first glance, this approach might seem somewhat similar to the clustering scheme [12] or the three-dimensional "mip-map" approach [9, 13]. However, our approach is based on volume-based filtering for gradual elimination of high frequencies, which is different from the locality-based clustering of geometry as presented in [12]. In the three-dimensional "mip-map" approach, every level of the hierarchy is formed by averaging several voxels from the previous level. In our approach, each level of the volume buffer hierarchy is created by convolving the original object with a low-pass filter of an appropriate support, whose size can be theoretically any positive real number. Thus, errors caused by a non-ideal filter do not propagate and accumulate from level to level. Furthermore, depending on the requirements of simplification speed and accuracy, a variety of low-pass filters can be applied. For example, for efficiency and ease of implementation, a hyper-cone filter can be used. A hyper-cone filter has a spherical filter support of radius R and is weighted such that its contribution is maximum at the center of the filter support, and linearly attenuates to zero at a distance R from the center. Mathematically, the hyper-cone filter is

defined as:

$$h(r) = \begin{cases} \dfrac{(R-r)w}{R} & 0 \le r \le R \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where w is the normalization factor which constrains the total contribution of the filter to be one. To achieve better results, one can employ a higher-order filter, such as a Gaussian filter.

Analytic evaluation of Equation 1 is sometimes possible for objects which are represented by algebraic mathematical functions. However, for general mathematical functions, polygon meshes, or volume datasets, an analytical solution either does not exist or is too expensive to be calculated. For such cases, a discrete approximation can be used. This is accomplished by convolving a precomputed-discrete filter with the object.

Once the multi-resolution volume representations have been established, the marching cubes algorithm [10] is used for reconstructing isodensity surfaces. In this algorithm, an isodensity surface is approximated by determining its intersections with edges of every voxel in the volume buffer. Up to five triangles are used to approximate the surface within a voxel. One advantage of the marching cubes algorithm is that it can be efficiently implemented using a precomputed lookup table for the various arrangements of surface-voxel intersections.

3. Multi-Layered Marching Cubes Rendering

Although the binary surface classification used by the traditional marching cubes algorithm generally generates good results from the point of view of modeling, it does introduce infinitely high frequencies. Since these frequencies cannot be fully represented in the discrete image, they can cause image-space aliasing. As an alternative to the commonly used hardware-supported antialiasing for rendering, we have developed a multi-layered marching cubes antialiased rendering algorithm. This approach takes advantage of the low-pass filtering applied during the volume-sampling stage. The non-binary surface classifier that we have used permits surfaces to be associated with a continuous range of densities, thereby allowing a smooth transition from the object-boundary to the empty space.

In this section, we present a marching cubes based approach to discretely approximating the surface boundary by generating several layers of triangle meshes with increasing translucency from the innermost layer to the outermost one (Figure 1b). Then, by appropriately compositing these layers of triangle meshes using

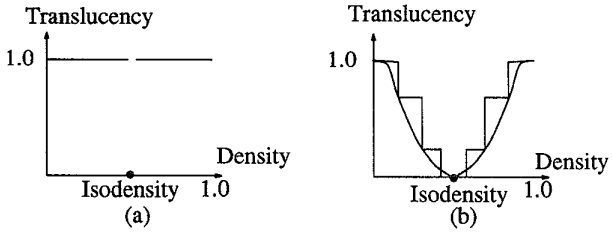

Figure 1: *(a) Binary surface classification. (b) Continuous surface classification and discrete approximation.*

hardware-assisted blending, a high frame rate of antialiased rendering can be achieved. The accuracy of the discrete approximation of the continuous surface classification is determined by the number of mesh layers used and their corresponding isodensities. Better approximation can be achieved with more layers, at the cost of increased storage space and rendering time. The minimum number of layers needed for the approximation to be within a user-specified error bound depends upon both the low-pass filter employed to generate the volume buffer and the geometry of the original polygon mesh (e.g., Figure 2). However, it can be computed approximately by the following method.

First, it is assumed that the translucency of a point with a certain density d is

$$1 - \frac{d}{m} \qquad (4)$$

where m is the maximum isodensity value associated with the innermost layer M of the multi-layered surfaces. Then, by assuming that the density of a point is decided solely by its distance to M, we can approximate the density at every point. Mathematically, centering the low-pass filter h with support R at a point with distance r from M, and assuming the filter intersects a planar surface (Figure 2a), the density of this point is:

$$d(r) = \int_r^R \int_{-\sqrt{R^2-\alpha^2}}^{\sqrt{R^2-\alpha^2}} \int_{-\sqrt{R^2-\alpha^2-\beta^2}}^{\sqrt{R^2-\alpha^2-\beta^2}} h(\alpha, \beta, \gamma) d\alpha d\beta d\gamma \qquad (5)$$

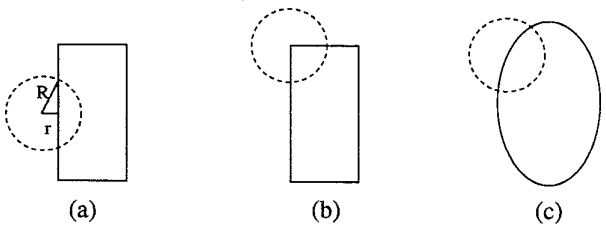

Figure 2: *Different intersections between a surface (solid) and the filter (dashed).*

Therefore, given an error bound ε, the minimum number of layers needed and their corresponding isodensities are decided by a piecewise constant function p with a minimum number of segments which satisfies:

$$\int_0^R |p(x) - d(x)|\, dx \leq \varepsilon \qquad (6)$$

The optimal piecewise constant function p might not be analytically derivable for certain filters. However, by using the heuristic that more layers should be placed where the derivative of the function d is high, sub-optimal p can be recursively generated. The number of layers of triangle meshes is then equal to the number of segments in the function p, and the isodensities of the meshes are the corresponding constants of that function. In addition, the corresponding translucency can be computed using Equation 4.

In order to generate the correct composition of semi-transparent meshes, the triangles should be projected in either a back-to-front or a front-to-back order, each of which generally involves an expensive sorting process. A nice property of a marching cubes generated mesh is that it is associated with a volume buffer. As a result, sorting can be accomplished by traversing only the surface-intersected voxels in a slice-by-slice fashion. Projection of multiple objects, however, is more complex. One simple solution is to perform the sorting on bounding boxes of the volume buffers associated with the objects. A more accurate and efficient sorting algorithm is to take advantage of the volume buffers associated with the meshes of these objects, since intersections among the regularly partitioned volume buffers are easy to compute. Therefore, all the surface-intersected voxels can be rapidly traversed in the correct order.

4. Results

We have tested our object simplification algorithm on a variety of objects such as a fractal sphereflake dataset, a CSG-generated mechanical part, and a sampled dataset of a human head. The results of our algorithm on the genus-simplification (as well as on the triangle count simplification to a certain extent), have been very encouraging and are summarized below.

Figure 3 illustrates the triangle-mesh hierarchy of a sphereflake fractal with 820 spheres. The original triangle mesh, shown in Figure 3a, is reconstructed from a high resolution volume buffer to preserve the details. By convolving the original fractal functions with Gaussian filters with different radius supports, we decrease the resolution of volume buffers accordingly, and the resulting number of triangles in the simplified mesh is reduced. The simplification results are presented in Table 1, with the index specifying the corresponding image in Figure 3. The discrete approximations of the Gaussian filters applied are at resolution $11 \times 11 \times 11$. The surfaces have been reconstructed from multi-resolution volume buffers using an isodensity of 0.5 on a normalized scale of 0 to 1.

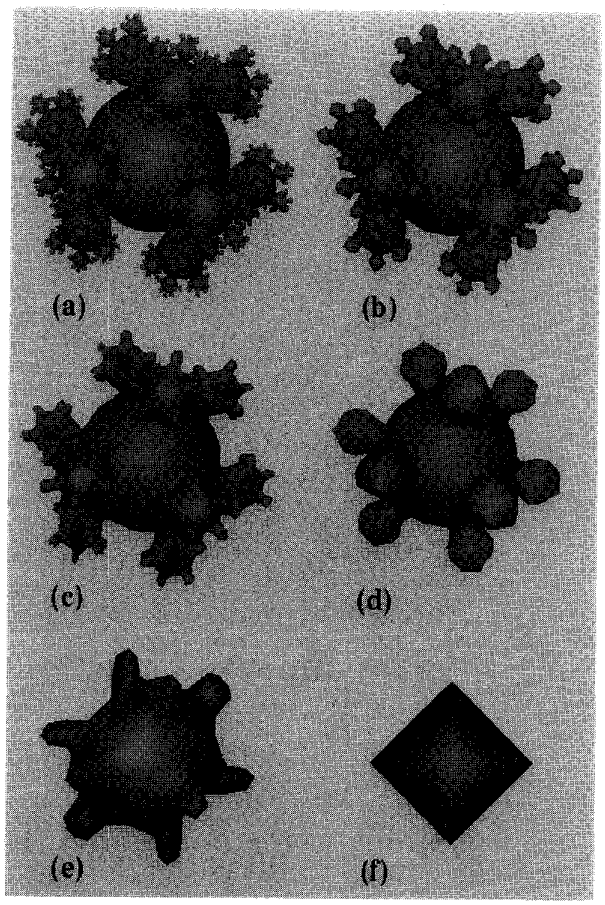

Figure 3: *Different levels of detail of a fractal sphereflake using a triangle mesh hierarchy.*

Table 1: *Simplification of a fractal sphereflake*

Index	Resolution	Radius	Triangles
a	200×200×200	1	322214
b	100×100×100	2	64926
c	50×50×50	4	13576
d	30×30×30	6.7	3756
e	15×15×15	13.3	636
f	5×5×5	40	8

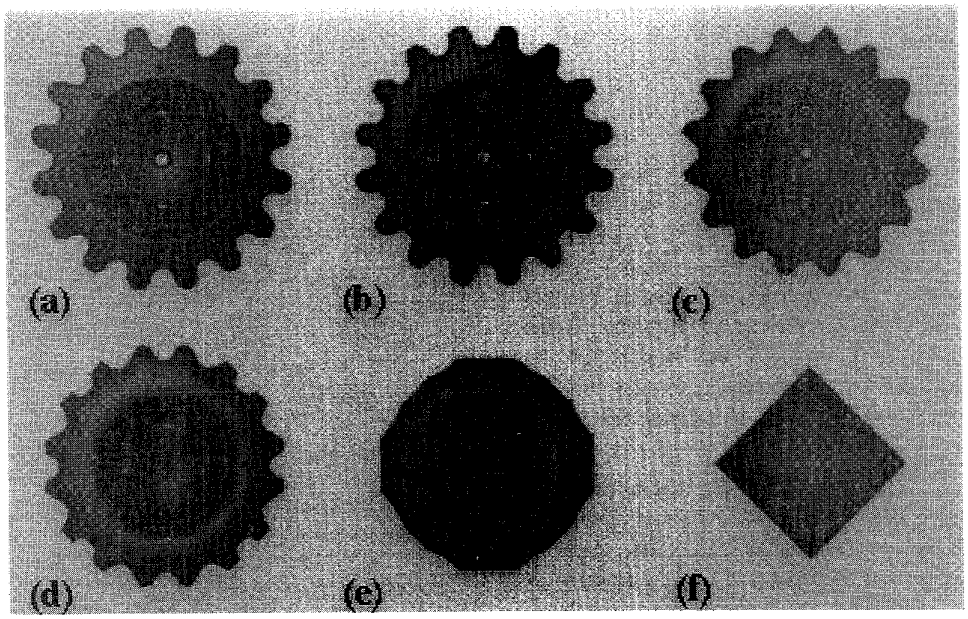

Figure 4: *Different levels of detail of a synthetically CSG-generated mechanical part.*

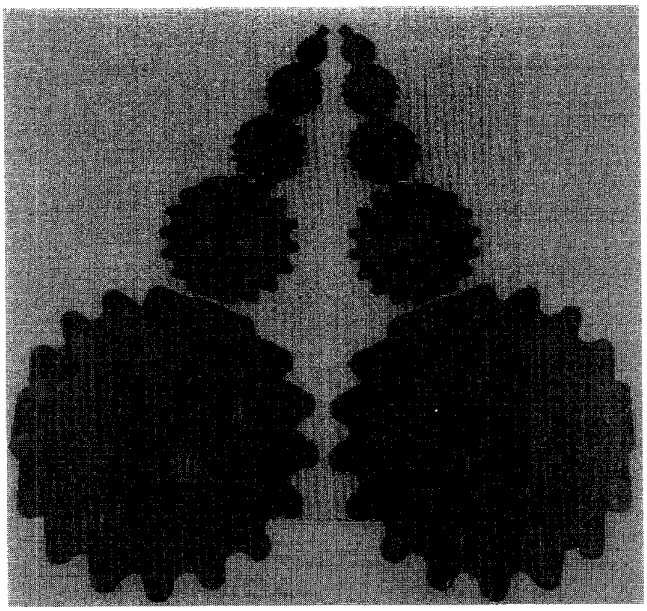

Figure 5: *Level-of-detail-based rendering of mechanical parts at six levels.*

Figure 4 demonstrates a mechanical part generated by CSG operations using volume-sampled voxelized primitives [17]. The level-of-detail meshes established by applying Gaussian filters of different radius supports are presented in Table 2, with the index specifying the corresponding image in Figure 4. The surfaces have been reconstructed from multi-resolution volume buffers using an isodensity of 0.5 on a normalized scale of 0 to 1. These images have been rendered using a solid steel texture. From these results, it can be seen that our algorithm provides an elegant way to gradually reduce the genus and small features. Figure 5 presents the effect of the simplification on an assembly of identical mechanical parts at different resolutions (as shown in Figure 4) which depend on the distance of the parts from the viewpoint.

Table 2: *Simplification of a CSG mechanical part*

Index	Resolution	Radius	Triangles
a	200×200×120	1	271504
b	100×100×60	2	64344
c	50×50×30	4	13292
d	40×40×24	5	8660
e	20×20×12	10	1508
f	5×5×3	40	88

Figure 6 presents the results of applying our algorithm on a volumetric dataset, a CT-scanned human head of $256 \times 256 \times 225$ resolution. The original and the simplified meshes reconstructed from marching cubes are presented in Table 3, with the index specifying the corresponding images in Figure 6. Hyper-cone filters with different radius supports are applied in this example.

Table 3: *Simplification of a CT-scanned human head*

Index	Resolution	Radius	Triangles
a	256×256×225	1	865698
b	128×128×113	2	193790
c	64×64×57	4	42688
d	32×32×29	8	9246
e	8×8×8	32	300
f	4×4×4	64	52

Unlike the volume buffer generated from a solid object, a medical dataset such as the CT-scanned head generally does not have a well-defined surface. However, for a given point it is still possible to test whether this point is inside or outside the surface by tri-linearly interpolating the point value from the neighboring eight vertices and comparing it to the isodensity, and therefore Equations 1 and 2 can still be applied. Another method of simplifying volumetric datasets without well-defined surfaces is to directly apply the reconstruction filters with different radius supports on the original volumes. The application of 3D reconstruction filter for volumetric datasets has been previously discussed for volume rendering [18].

The effect of our antialiasing algorithm is demonstrated by employing five layers of meshes on a bolt, shown at the bottom half of Figure 7, and contrasted with the aliased result of applying the traditional algorithm with binary surface classification shown at the top half of Figure 7. It should be emphasized that the multi-layered marching cubes rendering generally requires more memory, and the rendering speed might be slower than the other hardware-supported antialiasing algorithms. However, it provides a competitive object-space antialiasing method, and is quite useful when a high-quality antialiasing effect is required.

5. Discussion and Future Work

We have outlined a practical and robust method for genus simplification of objects. The strengths of our method are that it (a) works for a wide variety of objects; (b) supplements existing geometry-based object simplification algorithms by gradually eliminating higher-frequency features; (c) is relatively easy to implement; and (d) is based on the robust theoretical foundation of signal-processing theory.

In order to reduce temporal aliasing, smooth interpolation between two adjacent resolution meshes should be generated on-the-fly, which is generally a non-trivial task. However, it is straightforward and efficient to interpolate between two adjacent resolution volume buffers for generating an in-between resolution volume buffer. To generate the corresponding polygon mesh on-the-fly using the marching cubes algorithm, only those voxels which might contain surfaces are examined. An interpolated voxel might contain a surface only if at least one of the corresponding regions in the two volume buffers contains a surface, or exactly one of the corresponding regions is inside the surface. Such voxels can be efficiently generated since the regions in the two volume buffers satisfying the above conditions can be pre-computed.

A potential problem with our voxel-based simplification method is that the marching cubes algorithm could generate a large number of redundant triangles in the regions of low surface curvature. However, a nice property of our approach is that it can be used as the first stage of the object simplification process to eliminate the undesirable high frequencies. Any of the other existing geometry-simplification methods that preserve the topology can then be applied to further reduce the number

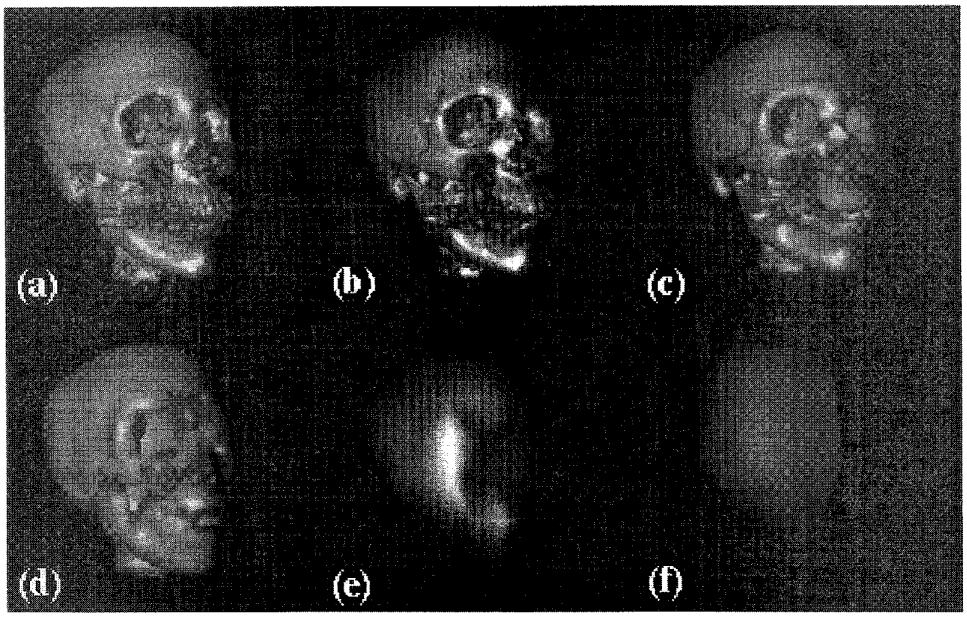

Figure 6: *Levels of detail of a CT-scanned human head.*

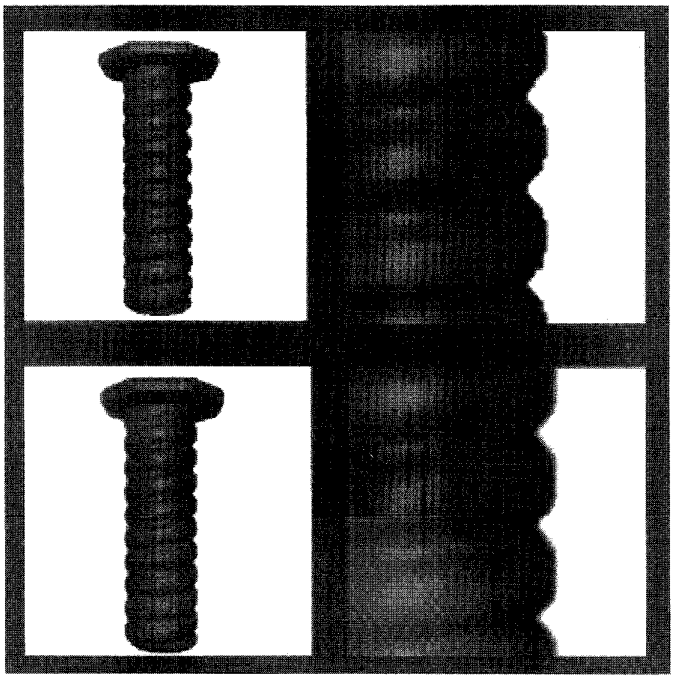

Figure 7: *A bolt rendered using multi-layered marching cubes (bottom) compared to a traditionally rendered bolt (top).*

of triangles. A fundamental solution to the above problem is to use the idea of adaptive subdivision of the volume space [11]. As part of our ongoing research in this area, we intend to explore the use of our voxel-based object simplification in conjunction with a local curvature-based adaptive volume buffer scheme which will enable us to simplify both the geometry as well as the genus of an object in a single pass. We are currently developing a method of low-pass filtering and sampling a polygon mesh into volume buffer with adaptive size voxels, where the high curvature areas are represented by small voxels and the smooth areas by large voxels. A corresponding adaptive marching cubes algorithm is being investigated.

Another area that promises to be of interest, and one that we are currently exploring, is the use of multi-resolution object hierarchies in collision detection. The idea here is to recursively perform collision detection among the multi-resolution descriptions of objects, starting from the lowest resolution representations and moving up to the higher resolutions only when an intersection is suspected. To test whether two objects collide at a certain resolution, the volume buffers associated with them are directly used. All the voxels of a volume buffer whose values are above a certain threshold are transformed into the local coordinate system of the other volume buffer. By checking the neighboring eight voxels in the other volume buffer, the possible intersections are detected. This approach works because every time a low-pass filter is applied with a larger support, the area affected by it becomes a superset. Thus, computation time is saved by avoiding intersection detection in regions that cannot possibly collide. Furthermore, this hierarchical approach can be interrupted, allowing us to trade accuracy for speed.

Acknowledgments

This work has been partially supported by the National Science Foundation under grants CCR-9205047 and CCR-9502239 and by the Department of Energy under the PICS grant. The images in Figure 3-7 are generated using the VolVis volume visualization system developed at Stony Brook. VolVis can be obtained by sending e-mail to volvis@cs.sunysb.edu.

References

1. Amanatides, J., "Ray Tracing with Cones", *Computer Graphics (SIGGRAPH '84 Proceedings)*, **18**, 3 (July 1984), 129-135.
2. Carpenter, L., "The A-buffer, an Antialiased Hidden Surface Method", *Computer Graphics (SIGGRAPH '84 Proceedings)*, **18**, 3 (July 1984), 103-108.
3. Clark, J., "Hierarchical Geometric Models for Visible Surface Algorithms", *Communications of the ACM*, **19**, 10 (1976), 547-554.
4. Crow, F. C., "A More Flexible Image Generation Environment", *Computer Graphics (SIGGRAPH '82 Proceedings)*, **16**, 3 (1982), 9-18.
5. DeHaemer, Jr., M. and Zyda, M. J., "Simplification of objects rendered by Polygonal Approximations", *Computers and Graphics*, **15**, 2 (1991), 175-184.
6. Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M. and Stuetzle, W., "Multiresolution Analysis of Arbitrary Meshes", *SIGGRAPH'95 Conference Proceedings* , August 1995.
7. Funkhouser, T. A. and Sequin, C. H., "Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments", *Computer Graphics (SIGGRAPH '93 Proceedings)*, August 1993, 247-254.
8. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., "Mesh Optimization", *Computer Graphics (SIGGRAPH '93 Proceedings)*, **27**, (August 1993), 19-26 .
9. Levoy, M. and Whitaker, R., "Gaze-Directed Volume Rendering", *Computer Graphics (Proc. 1990 Symposium on Interactive 3D Graphics)*, **24**, 2 (March 1990), 217-223.
10. Lorensen, W. E. and Cline, H. E., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *Computer Graphics (SIGGRAPH '87 Proceedings)*, **21**, 4 (July 1987), 163-169.
11. Muller, H. and Stark, M., "Adaptive generation of surface in volume data", *The Visual Computer*, 1993, 182-199.
12. Rossignac, J. and Borrel, P., "Multi-Resolution 3D Approximations for Rendering Complex Scenes", in *Modeling in Computer Graphics*, B. Falcidieno and T. L. Kunni, (eds.), Springer-Verlag, 1993, 455-465.
13. Sakas, G. and Hartig, J., "Interactive Visualization of Large Scalar Voxel Fields", *Proceedings Visualization '92* , Boston, MA, October 1992 , 29-36 .
14. Schroeder, W., Zarge, J. and Lorensen, W., "Decimation of Triangle Meshes", *Computer Graphics (SIGGRAPH '92 Proceedings)*, **26**, 2 (July 1992), 65-70.
15. Turk, G., "Re-Tiling Polygonal Surfaces", *Computer Graphics (SIGGRAPH '92 Proceedings)*, **26**, 2 (July 1992), 55-64.
16. Varshney, A., "Hierarchical Geometric Approximations", *Ph.D. Thesis, Tech. Rep.-050-1994*, Department of Computer Science, University of North Carolina, Chapel Hill, NC 27599-3175, 1994.
17. Wang, S. W. and Kaufman, A. E., "Volume-Sampled 3D Modeling", *IEEE Computer Graphics & Applications*, **14**, 5 (September 1994), 26-32.
18. Westover, L., "Footprint Evaluation for Volume Rendering", *Computer Graphics*, **24**, 4 (August 1990), 367-376.