

# Topology-Reducing Surface Simplification Using a Discrete Solid Representation

CARLOS ANDÚJAR, PERE BRUNET, and DOLORS AYALA  
Universitat Politècnica de Catalunya

---

This paper presents a new approach for generating coarse-level approximations of topologically complex models. Dramatic topology reduction is achieved by converting a 3D model to and from a volumetric representation. Our approach produces valid, error-bounded models and supports the creation of approximations that do not interpenetrate the original model, either being completely contained in the input solid or bounding it. Several simple to implement versions of our approach are presented and discussed. We show that these methods perform significantly better than other surface-based approaches when simplifying topologically-rich models such as scene parts and complex mechanical assemblies.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—Boundary representations; Geometric algorithms, languages, and systems

General Terms: Algorithms

Additional Key Words and Phrases: Geometry simplification, multiresolution models, surface reconstruction

---

## 1. INTRODUCTION

The main goal of geometry simplification algorithms is to transform a given geometric model into a new representation featuring less geometric complexity while preserving important object features and minimizing some error function. These simplified representations either replace the original model or are arranged into several level-of-detail (LOD) representations with varying accuracy and complexity.

Over the last few years, many different approaches for surface simplification have been proposed. One of the main caveats of current techniques concerns the simplification of groups of objects and mechanical assemblies. As noted by Varshney [1997], most of the current simplification techniques seem to be optimized for small numbers of high-complexity, densely-tessellated objects such as terrains and digitized surfaces, rather than for hundreds of simple objects forming a topologically-rich assembly. Most surface simplification methods are unable to modify the topology of the model [Varshney 1997]. Unfortunately, the amount of geometric reduction and the quality of the simplified models are severely affected by topology preservation, therefore these methods usually do not perform well on topologically-complex models. A few methods attempt topological simplification (reviewed in Section 2) but they are based on error metrics defined on the *boundary surface* of the objects and they cannot solve the topology simplification problem in general, complex surfaces [Andújar 1999].

On the other hand, model integrity issues, such as guaranteeing the two-manifold condition on the simplified model, have not received too much attention because these aspects do not play a critical role

---

Authors' address: Universitat Politècnica de Catalunya, Dept. LSI, Diagonal 647, E-08028 Barcelona, Spain, e-mail: {andujar, pere, ayala}@lsi.upc.es.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 2002 ACM 0730-0301/02/0400-0001 \$5.00

in visualization-oriented models. Most incremental methods (e.g. [Schroeder et al. 1992; Cohen et al. 1996]) proceed through a sequence of local reduction transformations starting from the initial mesh, which forces them to be quite sensitive to the way the input surface is tessellated, and thus to preserve input degeneracies such as small cracks, double walls, self-intersections and dangling parts. These features keep output models from being suitable for other applications besides visualization, such as approximate occlusion culling [Andújar et al. 2000], global illumination [Rushmeier et al. 1993] and query acceleration [Veltkamp 1992].

In this work we present a new method for geometry simplification which addresses the above problems. The key ingredient of our approach is an intermediate volumetric representation and the corresponding algorithms for converting the input model to and from this representation. The remainder of this paper is organized as follows: Section 2 surveys some previous work on geometry simplification focusing on topology-reducing methods. Section 3 states the problem being addressed and provides a conceptual distinction between surface-metric simplification and solid-metric simplification. A new method for stable topology simplification, called TDPS, is presented in Section 4. In Section 5 some results are discussed and compared with existing methods. Finally, Section 6 provides some concluding remarks and future work.

## 2. RELATED WORK

Surface simplification literature has focused mainly on simplification algorithms tuned for highly-tessellated, topologically-simple surfaces, which are frequently modeled as triangular meshes. Methods based on wavelets [Eck et al. 1995; Gross et al. 1996] require the input surface to have special subdivision properties and exhibit inherent limitations for modifying the genus and connecting disconnected shells. Most others methods follow a top-down strategy, performing face reduction directly on the mesh by the iterative application of reduction operators such as *vertex removal* [Schroeder et al. 1992; Soucy and Laurendeau 1996], *edge collapse* [Hoppe 1996; 1997; Ronfard and Rossignac 1996; Guezic 1996], *face removal* [Hamann 1994] or *surfaces merging* [Kalvin and Taylor 1996]. All these algorithms have been designed to preserve the original topology of the mesh. The interested reader can consult the work by Cignoni et al. [1998] for a comparative survey on these algorithms. In the rest of this section we will concentrate our attention only on those simplification methods that attempt topological simplification.

*Vertex clustering* methods group nearby vertices into *clusters*, then replace all vertices inside a cluster by a single vertex. Rossignac and Borrel [1993] present a simple yet efficient algorithm that subdivides the model by a uniform grid and uses a vertex clustering approach within each grid cell. Low and Tan [1997] and Luebke and Erikson [1997] improved this method by extending the shape and location of clusters. Low and Tan [1997] replace the uniform grid by arbitrary-shaped floating cells that are constructed by sorting the vertices on their importance, and then iteratively letting the most important vertex be the center of a new cluster, thus making Rossignac and Borrel's method less susceptible to producing artifacts resulting from a particular alignment of the cell boundaries. Luebke and Erikson [1997] presented a dynamic algorithm for view-dependent simplification which uses an octree structure for storing a hierarchy of potential vertex clusters. All these vertex-clustering methods allow topological changes in regions with nearby vertices lying within a cluster, but they tend to produce simplified models exhibiting dangling elements and do not guarantee the two-manifold condition.

*Pair contraction* methods proceed by the iterative elimination of geometric entities through a local transformation (sometimes called vertex collapse or vertex unification) consisting in joining a pair of vertices not necessarily connected by an edge. These methods differ basically on the decimation criteria adopted. Garland and Heckbert [1997] use quadric error metrics for their simplification algorithm based

on vertex collapses. Candidates for collapse are any two vertices that are within a user specified distance threshold or are connected by an edge. Garland and Heckbert's work has inspired other methods based on pair-contraction [Erikson and Minocha 1998; 1999]. Popovic and Hoppe [1997] extend Progressive Meshes by Hoppe [1996] and select candidate vertices using a cost function that includes a measure of distance to the original surface as well as a term that penalizes contractions that would merge vertices having different appearance properties. El-Sana and Varshney [1999] describe a pair-contraction method suitable for selective refinement. A Delaunay tetrahedralization of a subset of the vertices is computed, and candidate vertices for pair contraction are those sharing a Delaunay edge. Schroeder [1997] presents a further development of a previous decimation algorithm [Schroeder et al. 1992] that works with invertible operations allowing topology reduction and progressive mesh representation. All the above pair-contraction methods allow topological changes in regions with nearby vertices, but they do not identify and remove triangles that become internal after genus reductions. Furthermore, pair-contraction methods create non-manifold vertices.

El-Sana and Varshney [1998] proposed a controlled topology simplification scheme based on the notion of alpha-hulls which extends a previous genus-reducing approach [El-Sana and Varshney 1997]. Alpha prisms are constructed in regions with holes and cavities, and their union is performed by computing their pairwise intersections. Several heuristics are used for identifying regions to be tessellated. This method nicely reduces the genus, removes protuberances and fills cracks, but it does not support aggregation and it does not handle certain degeneracies such as coincident polygons and self-intersecting meshes.

*Volume-based* methods achieve topology simplification by converting a 3D model to and from a volumetric representation. Ayala et al. [1995] presented a simplification algorithm using an octree as the intermediate model. A two-manifold surface is computed from the octree by a discretized, orthogonal version of Marching Cubes [Lorensen and Cline 1987]. Unfortunately, some parts of the octree called non-regular regions were reconstructed separately thus this method increased the number of components of the original model. Andujar et al. [1996] extended this method to arbitrarily-oriented models by a further decimation of the reconstructed surface, but the method was not simple to implement and it still had the non-regular regions problem. He et al. [1995] proposed a similar topology reduction strategy which is based on discretization into volume rasters followed by isosurface extraction. First, the input solid is sampled and low-pass filtered resulting in a 3D grid from which the final surface is generated by applying an isosurface extraction algorithm. The extraction method adopted in He et al. [1995] is the original Marching Cubes, and an improved version is used in a further development [He et al. 1996]. The approach of He et al. greatly reduces the topology of the model, but it tends to produce verbose representations.

### 3. PROBLEM STATEMENT

The aim of this work is to contribute to the simplification of complex models and assemblies by providing a topology-reducing algorithm whose output is suitable not only for visualization but also for interference detection, global illumination and query acceleration.

A generic *approximation problem* is defined as follows. Given a solid  $P$  and an error parameter  $\varepsilon > 0$ , compute a polyhedral approximation  $P'$  such that (a)  $P'$  is a valid boundary representation of a solid, (b) the maximum distance between  $P$  and  $P'$  is at most  $\varepsilon$ , (c)  $P'$  exhibits less geometric complexity than  $P$ , and (d)  $P'$  exhibits topological complexity less than or equal to that of  $P$  (measured as the number of connected components of the solid).

The problem we address differs from classic surface simplification because we measure distances between solids, not just between their boundary surfaces. The remainder of this section discusses

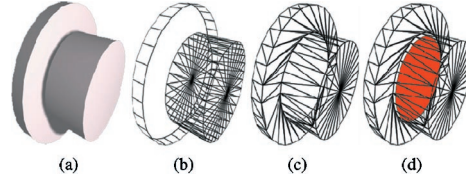


Fig. 1. Aggregation of two cylinders: (a) two nearby cylinders; (b) wireframe representation of these cylinders showing that they are actually disconnected; (c–d) result of aggregating the two cylinders. Note that the middle point of the red disk, which is on the surface of the original model, is far from the surface of the merged model.

the key differences between classic surface simplification and the solid simplification problem being addressed.

With a few exceptions [He et al. 1995; 1996], almost all surface simplification methods use metrics and geometric criteria based on the points of a surface, that is the domain of the distance function is a subset of the *boundary* points of  $S$  and  $S'$ ; such metrics will be referred to as *surface-metrics*.

For example, the surface-metric version of the Hausdorff distance is defined as follows,

$$d_{on}(S, S') = \max(d_h(\partial S, \partial S'), d_h(\partial S', \partial S)), \quad (1)$$

where  $\partial S$  denotes the boundary of  $S$  and  $d_h(A, B)$  is the directed Hausdorff distance from  $A$  to  $B$ , i.e.  $\max_{a \in A}(\min_{b \in B}(\text{dist}(a, b)))$ .

As shown in Figure 1, surface-metrics are not suitable for the simplification of complex assemblies because they strongly limit the amount of topology reduction. Consider, for example, the two cylinders shown in Figure 1-a. A surface-metric such as the surface-Hausdorff distance keeps simplification algorithms from merging the two cylinders into a single solid (Figure 1-c) because the surface-Hausdorff distance would report an approximation error equal to the length of the radius of the red circle shown in Figure 1-d.

Our approach uses a different domain for evaluating the approximation error. We use metrics based on the points *inside* the solid. Such metrics will be referred to as interior metrics or *solid-metrics*. The solid-metric version of the Hausdorff distance will be referred to as *solid-Hausdorff* and is defined as follows (where  $iS$  denotes the closure of the interior of  $S$ ):




$$d_{in}(S, S') = \max(d_h(iS, iS'), d_h(iS', iS)). \quad (2)$$

Note that, unlike  $d_{on}$ ,  $d_{in}$  reports a relatively small approximation error between the separated cylinders and the connected solid of Figure 1, thus allowing the aggregation of the two pointsets.

Since the domain pointset defines the space of possible solutions for a given error parameter and exerts such a strong influence upon the result of the simplification process, it has been suggested [Andújar 1999] that each family of error metrics actually corresponds to a different problem. More specifically, simplification methods based on *surface-metrics* are in fact concerned with a subproblem of geometry simplification that can be referred to as *surface-metric simplification*. Since the disparity between the original surface and the simplified one is measured by surface-metrics, the goal of surface simplification is to calculate a less complex surface that approximates the original one within a tolerance bound around the surface.

Our approach to the geometry simplification problem derives from *solid-metrics* and will be referred to as *solid-metric simplification*. Unlike surface-metric simplification, solid-metric simplification deals with 3D solids and takes into account the region enclosed by a two-manifold surface. Solid-metric simplification applications make us somewhat less concerned with preserving surface appearance attributes

Table I. Surface-Metric Simplification vs. Solid-Metric Simplification

	Surface simplification	Solid Simplification
Underlying entity	surface (2D)	solid (3D)
Error metric	points on the surface	points inside the solid
Topology simplification	very limited	high degree of freedom
Model validity	relaxed	enforced
Previous work	intense research	papers are still rare
Applications	visualization-oriented	visualization, collision detection, occluder synthesis...
		

such as curvature and color, whereas validity-preservation becomes crucial. The differences between surface-metric and solid-metric simplification are summarized in Table I.

#### 4. TRIHEDRAL DPS

This section describes a topology-reducing simplification algorithm called Trihedral DPS (TDPS) which belongs to the class of algorithms modeled by the discretized polyhedra simplification (DPS) framework described in Andújar [1999]. DPS models a family of simplification methods that achieve aggregation and genus reduction by converting a 3D model to and from an octree-based volumetric representation. This fact makes our approach completely insensitive to the way the input surface is tessellated and hence to the typical degeneracies of 3D models.

##### 4.1 Algorithm Overview

The TDPS method proceeds through three major steps described graphically in Figure 2:

—*Discretization*

During discretization (Section 4.2), the  $L$ -level octree representation of the input model is computed for a given positive integer  $L$ . Note that, as in any hierarchical space decomposition model, the removal of the deepest level of the octree has the effect of doubling the size of the terminal nodes and hence decreasing its accuracy. Consequently, an  $L$ -level octree can be viewed as a multiresolution series of octree representations ranging from  $O_L$  to  $O_1$ , where the subindex denotes the level of the tree at which octree nodes are considered to be leaves.

—*Reconstruction*

The reconstruction step (Section 4.3) consists in the extraction of a polyhedral surface from an octree representation. This process is applied iteratively to  $O_L, \dots, O_1$  yielding a series of polyhedral models  $R_L, \dots, R_1$  with decreasing accuracy and complexity.

—*Face reduction*

The face reduction process (Section 4.4) further simplifies the surface extracted from the octree.

##### 4.2 Discretization

The discretization step consists in the construction of an intermediate octree representation  $O_L$  corresponding to the input solid  $P$ . We use a variant of the classical octree representation [Samet 1990]

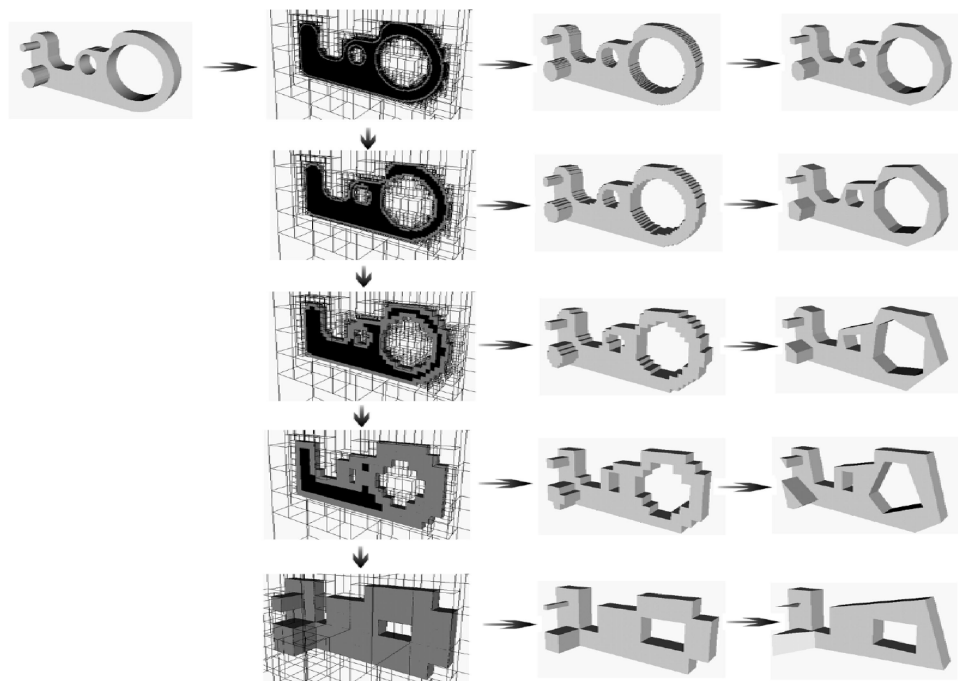


Fig. 2. Algorithm overview. From left to right: original polyhedron, multiresolution of octrees (some front GREY nodes have been culled to keep BLACK nodes visible), reconstructed models and final multiresolution.

called *maximal division classical octree* (MDCO) [Brunet and Navazo 1990; Ayala et al. 1985]. A MDCO uses a recursive subdivision of a cubic universe into eight octants that are arranged into an 8-ary tree. Nodes corresponding to cubic regions completely inside the object are labeled as BLACK (B) and nodes completely outside the object are labeled as WHITE (W). WHITE and BLACK nodes are not further subdivided. Nodes containing a part of the boundary are labeled as GREY (G) and are recursively subdivided up to a certain level  $L$ , resulting in *TERMINAL GREY* (TG) nodes. The boundary of the cubes associated to octree nodes are defined so that TG nodes corresponding to connected regions are 6-connected [Andújar 1999].

Classical octree construction from a BRep is a well known problem based on a simultaneous space subdivision and clipping of the boundary of the polyhedron [Brunet and Navazo 1990]. We start with a single GREY node  $n$  representing the bounding cube of the 3D model and then we recursively subdivide  $n$  until the maximum depth  $L$  has been reached. At each step, the geometry intersected by  $n$  is distributed into the eight sons following the algorithm presented by Brunet and Navazo [1990]. Each son is labeled according to its interior geometry. If the son node does not contain any geometry, then it must be either completely inside or completely outside  $P$  thus it is labeled as NON-GREY and it is not further subdivided. Otherwise, it is labeled as GREY and is recursively subdivided.

After the subdivision process, NON-GREY nodes are segmented into BLACK and WHITE nodes. Instead of performing a point-inside-solid classification as in Brunet and Navazo [1990], we use the robust seed algorithm presented in Andújar [1998] which labels as WHITE those NON-GREY nodes that can be reachable following a 26-connected path from a boundary NON-GREY seed. After all WHITE nodes have been identified, the remainder NON-GREY nodes are considered to be BLACK. This segmentation allows TDPS to remove interior cavities and to accept degenerate models as input.

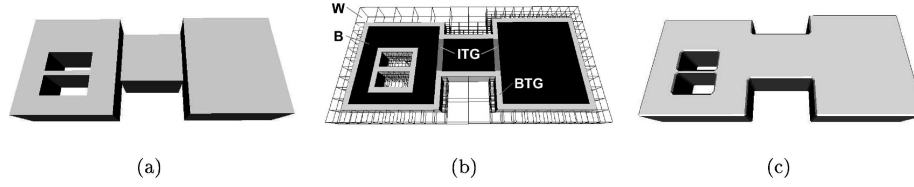


Fig. 3. Feasible reconstruction in TDPS: (a) input model (three components); (b) interior section of the octree showing node classification; (c) feasible reconstruction (one component).

### 4.3 Reconstruction

The reconstruction step consists in the extraction of a polyhedral surface  $R$  from each octree representation  $O_k$ . Before describing the reconstruction algorithm, we first discuss how to guarantee a precise error bound in terms of the solid-Hausdorff distance.

The error bound of the reconstructed solid  $R$  with respect to the original solid  $P$  is guaranteed by constraining  $R$  to lie inside a particular subset of octree nodes. Given a MDCO  $O$ , a *restrictor* is a classification rule for which, nodes must be traversed by the boundary of  $R$ , which nodes must be completely inside  $R$ , and which nodes must be completely outside  $R$ , in order to fulfill a given error distance. A restrictor can be seen as an outer boundary and an inner boundary (similar to simplification envelopes [Cohen et al. 1996]), defined in terms of the union of octree nodes.

Several restrictors have been discussed in Andújar [1999]. We have adopted the *in-restrictor*, which distinguishes between border and interior TG nodes. A TG node is said to be *border* (BTG node) if at least one of its 26-neighbors is WHITE. Otherwise, the TG node is said to be *interior* (ITG node). The *in-restrictor* establishes that the output solid  $R$  has to be reconstructed in such a way that (a) BLACK and ITG nodes are completely inside  $R$ , (b) WHITE nodes are completely outside  $R$ , and (c) all BTG nodes are intersected by the boundary of  $R$  (see Figure 3). Since we aim at minimizing the number of connected pointsets, we also add a further requirement: for any two BTG nodes, if they are 6-connected, then they must be intersected by a single connected faceset of  $R$  (i.e. we require maximally connected facesets in  $R$ ).

A solid  $R$  fulfilling the above conditions is said to be *feasible*, and this is a sufficient condition for the solid-Hausdorff distance between  $R$  and the original polyhedron to be bounded by the length of the diagonal of a TERMINAL GREY node [Andújar 1999]. Consider for example, an  $L$ -level octree  $O_L$  representing a cubic universe of size  $U$ ,  $U$  being the length of its main diagonal. The length of the diagonal of its TG nodes is  $\varepsilon = U/2^{L-1}$ . Therefore, the solid-Hausdorff distance between the original polyhedron and the reconstructions provided by this octree is guaranteed to be bounded by  $\varepsilon$ . For instance,  $L = 8$  means a 0.78% error threshold.

Topology simplification is also controlled through the size of TG nodes. Two originally disconnected pointsets  $A$  and  $B$  will be aggregated in those regions where nodes in BTG( $A$ ) are 6-adjacent to nodes in BTG( $B$ ) (see Figure 8). A through hole (often called a tunnel) is closed (and therefore genus is reduced) in those regions where its TG nodes form a single 6-connected component (i.e., when terminal nodes are large enough to keep white nodes from appearing inside the hole, see Figure 9).

Therefore, both geometry and topology simplification are controlled through the size of TG nodes, which in turn depends on the user-defined maximum octree depth  $L$ . Our results demonstrate that controlling the topology reduction through the size of octree nodes is quite intuitive. Although topological changes are a little sensitive to octree alignment, in practical cases these changes are easy to predict.

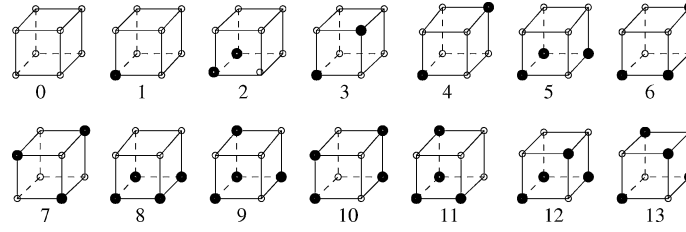


Fig. 4. The 14 equivalence classes.

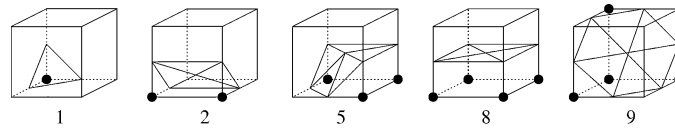


Fig. 5. Triangular reconstruction of non-ambiguous classes.

Given a MDCO  $O$ , our reconstruction algorithm computes a feasible triangle mesh  $R$  by first converting  $O$  into a *3D digital picture* and then extracting an isosurface from it (see Figure 8-f). A *3D digital picture* is a set of points (called *lattice points*) arranged into a regular grid. Digital pictures are often converted into 3D surfaces by using *beveled-form* algorithms such as marching cubes [Lorensen and Cline 1987]. These algorithms consider the cube defined by eight adjacent lattice points. There are 256 possible configurations of such a cube regarding the color at its eight vertices. These configurations can be grouped by means of symmetries into the 14 equivalence classes shown in Figure 4. The surface inside each cube is assumed to intersect those edges whose end-points have different color, and it is reconstructed through a look-up table which describes the reconstruction of each equivalence class. Some classes are non-ambiguous (Figure 4, classes 1, 2, 5, 8, 9 and 11) whereas other classes accept two topologically-different reconstructions (Figure 4, classes 3, 4, 6, 7, 10, 12 and 13) and caused topological inconsistencies in the original marching cubes.

The computation of the digital picture starts with a traversal of the MDCO looking for *BTG* nodes. Each *BTG* node  $n$  is subdivided into eight octants denoted as  $n_{1...8}$ . The color at each corner of the octants is computed by simply labeling as WHITE those corners which have contact with a WHITE node and labeling as BLACK those corners which do not. Fortunately, the digital picture computed by the above color definition only includes the non-ambiguous classes numbered as 0, 1,  $\bar{1}$  (the bar indicates the inside-outside complement), 2,  $\bar{2}$ , 5,  $\bar{5}$ , 8 and 9 in Figure 4, thus no explicit disambiguation method is required. The actual reconstruction of each of these classes is shown in Figure 5. Note that the intersection point on stabbed edges is always at the midpoint of the edge [Montani et al. 1994].

The extracted surface  $R$  is guaranteed to be feasible, therefore the solid-Hausdorff distance between  $R$  and the original polyhedron is bounded by the length of the diagonal of TG nodes. In fact,  $R$  is enclosed inside the region confined between the White Surface and the Black Surface, which are defined below. Given a MDCO  $O$ , the *White Surface*  $WS(O)$  is the cuberille surface [Fuchs et al. 1989] that separates WHITE nodes from the rest of nodes. Symmetrically, the *Black Surface*  $BS(O)$  is the cuberille surface that separates BLACK and *ITG* nodes from the rest. These discrete surfaces can be seen as outer and inner bounds respectively, within which a two-manifold surface completely confined inside *BTG* nodes can be found (see Figure 3).



MDCOs also provide a convenient framework for non-penetrating simplification, that is approximations completely bounding or completely enclosed inside the input solid. The octree transformation required for providing bounding approximations is as follows. Before the reconstruction step, TG are converted to BLACK, and WHITE nodes 26-adjacent to new BLACK are converted to TG (suppose that WHITE nodes have been subdivided). This operation guarantees that the feasible reconstruction computed from this new enlarged octree is a bounding volume of the input solid. Likewise, if TG become WHITE and 26-adjacent BLACK nodes become TG, then reconstructed solids will be enclosed inside the input solid.

#### 4.4 Face Reduction

The aim of the face reduction step is to further reduce the geometric complexity of the extracted solid  $R$  while preserving its validity. Our face reduction differs from classic surface simplification in many ways. TDPS face reduction starts with a valid two-manifold solid whose geometry, incidence and topology has been transformed so that (a) the new surface inside each MDCO node is quite simple, and (b) it exhibits less topological complexity since those parts of the boundary which were not visible from the outside have been removed, small through holes have been closed, and nearby disconnected parts have been merged into a single component. A key aspect is that our face reduction is not concerned about changing the topology of  $R$ , an operation that would involve primitives such as pair-contraction, which create non-manifold boundaries.

TDPS uses an incremental simplification approach based on the edge-collapse operator (see Figure 7). We have adapted the geometric criteria proposed in Garland and Hekbert [1997] and Lindstrom and Turk [1999] along with the use of restrictors. After each local update, the new surface is required to be feasible, that is it must stab all BTG nodes and it must enclose a 3D volume containing all BLACK and ITG nodes. Since face reduction starts with a reconstructed surface that feasible, we need to guarantee that each local update of the surface preserves this condition. A local update on surface  $S$  replaces a group of faces  $F \in S$  with another group of faces  $F'$  covering the same area. Such an update is accepted if (a) all the octree nodes intersected by  $F'$  are labeled as BTG, and (b) for each node  $n$  intersected by  $F$ , if  $n$  is not intersected by  $F'$  then  $n$  is intersected by  $S - F$ , so the new surface  $S' = (S - F) \cup F'$  is feasible [Andújar 1999].

Before the face reduction step, we relax the vertices of the reconstructed surface  $R$  by the iterative application of an attracting force defined via the gradient vector [Ohtake and Belyaev 2001]. The relaxation process stops before surface  $R$  intersects a non-BTG node. We use a positive parameter  $\tau$  for scaling the attracting forces. In our implementation  $\tau$  is set so that only 4–5 iterations are performed.

#### 4.5 Alternative Orthogonal Reconstruction

As shown in Section 5, the simplification method described above, produces reasonably good results on a wide range of 3D models. However, a reconstruction providing support to nontriangular faces would lead to more concise approximations of those octrees having large linearly-separable regions. This section introduces an alternative reconstruction which is able to produce models involving non-triangular faces (see the third column of Figure 2).

Given a MDCO, the  $\alpha$ -inner offset of its White Surface is defined as  $\bar{W} \ominus C_\alpha$ , where  $\bar{W}$  denotes the set of non-WHITE nodes (i.e. B and TG nodes), ' $\ominus$ ' is the Minkowski subtraction operator and  $C_\alpha$  is a cube of length  $2L\alpha$  centered at the origin,  $L$  being the side of a TG node. For  $0 \leq \alpha \leq 1$ , inner offsets define a continuous series of orthogonal polyhedra between the White Surface (0-offset) and the Black Surface (1-offset). It can be shown that  $0 < \alpha < 1/2$  is a sufficient condition for an inner-offset to be a feasible two-manifold [Andújar 1999].

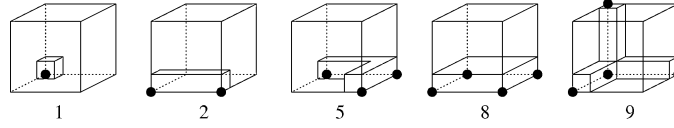


Fig. 6. Orthogonal reconstruction of non-ambiguous classes.

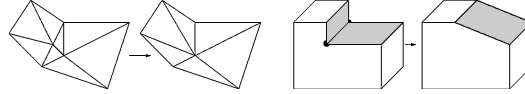


Fig. 7. Edge-collapse and its dual primitive, face-merge.

Our orthogonal reconstruction algorithm generates the  $\frac{1}{4}$ -inner offset by just using the lookup table shown in Figure 6. The extracted surface contains a large number of adjacent coplanar faces which can be merged into large nontriangular faces. This compactation step can be greatly optimized by first computing the *geometry* of the orthogonal reconstruction  $R$  as a list of vertex coordinates, and then connecting these vertices into edges, loops and faces by a domino-like algorithm which performs an ordered traversal of coplanar vertices. We have implemented the algorithm described in Brunet and Navazo [1990] because of its simplicity, although other efficient methods exist [Montani and Scopigno 1991].

The extracted surface  $R$  is guaranteed to be feasible and involves orthogonal faces with an arbitrary number of rings and vertices, although its incidence graph is far simpler than arbitrary Polyhedral models, because all its vertices have degree three.<sup>1</sup> This model is topologically equivalent to that produced by the triangular reconstruction described in Section 4.3.

Our face reduction approach for orthogonal solids is based on an extension of the classic *coplanar face-merge* [Kalvin et al. 1991; Montani et al. 1994] called *generalized face-merge*. The generalized face-merge is the dual equivalent of the edge-collapse (see Figure 7). It accepts as input, two adjacent faces (not necessarily lying on the same plane) and then merges them into a single face. Since the input faces are not generally coplanar, the orientation of the new face determines the position of its vertices, which must be recomputed by intersecting the planes of its corresponding faces (in the above orthogonal solid there are only three planes sharing a vertex).

Incremental simplification of orthogonal solids proceeds by the iterative application of the extended face-merge transformation. First, we sort the edges of the reconstructed surface into a priority queue according to the degree of its two incident faces  $f_1$  and  $f_2$  [Andújar 1999]. We have adopted a low-degree criterion because it is very fast to compute and it usually improves the output quality. This criterion contrasts with the purely geometric criteria used for triangle meshes. However, these geometric criteria tend to lead orthogonal solids to reach local minima faster than the low-degree criterion [Andújar 1999]. In order to merge faces  $f_1$  and  $f_2$ , a plane  $\Pi$  stabbing all nodes intersected by  $f_1$  and  $f_2$  can be computed in optimal  $O(n)$  time using the linear programming technique presented in deBerg et al. [1997],  $n$  being the number of MDCO nodes traversed by  $f_1$  and  $f_2$  [Andújar 1999]. As in the original face reduction step, after each local update the new surface is required to be enclosed inside the set of BTG nodes.

## 5. RESULTS

TDPS has been implemented and tested on a number of polyhedral models with varying shape and complexity. Figure 8 shows the output of TDPS on a basketball model with 23,730 faces that has

<sup>1</sup>Some vertices have degree six, but they can be trivially split into two vertices of degree three [Andújar 1999].

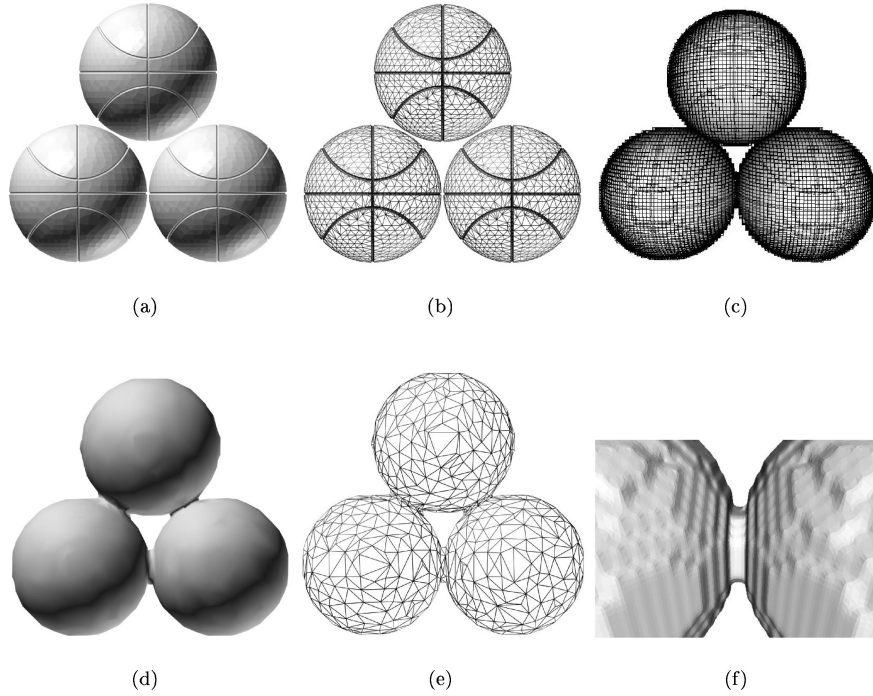


Fig. 8. Results of TDPS for the basketball model: (a–b) original model (23,730 t); (c) original model and its 8-level TG nodes; (d–e) simplified model (1,797 t); (f) detail of the surface reconstructed from the octree, before decimation.

been selected for demonstrating aggregation. Note that the three disconnected basketballs have been merged in those regions where their corresponding TG nodes are 6-adjacent. Figure 8 is also a good example of the fact that our method is able to both increase and decrease the genus. In fact, the input model has three 0-genus components, whereas the result has a single 1-genus component. This is just a consequence of allowing the aggregation of parts belonging to the same pointset.

Figure 9 shows the output of TDPS on a mechanical piece model with 14,832 faces that has been selected for demonstrating genus simplification. Note that some small holes have been closed, particularly in those regions where no WHITE nodes appear inside the hole.

Figure 10 shows the output of TDPS on several complex models. The first three models have been simplified using an 8-level octree which guarantees the error to be bounded by the 0.78% of the diagonal of the cubic universe, which in our implementation is about 25% larger than the bounding box of the model. The engine model has been simplified using a 9-level octree guaranteeing a 0.39% bound. Note that most of the piping of the factory model (20,794 faces) has been transformed into large cylindrical parts, showing the effects of aggregation. Topology reduction is even more dramatic in the piping model (1,976 objects, 34,532 faces), the truck model (47,817 faces) and the engine model (140,113 faces), which after simplification had only a single connected component. Although octrees used for discretization are axis-aligned, the output surface is reasonably good in non-orthogonal regions.

Table II shows the running times for the three main steps of TDPS on a Pentium III 800MHz processor. Note that discretization is the only process that is sensitive to the number of faces of the input model. Reconstruction and face reduction processes basically depend on the number of BTG nodes of the octree, which in turn depends on the maximum octree depth. Our experiments show that increasing

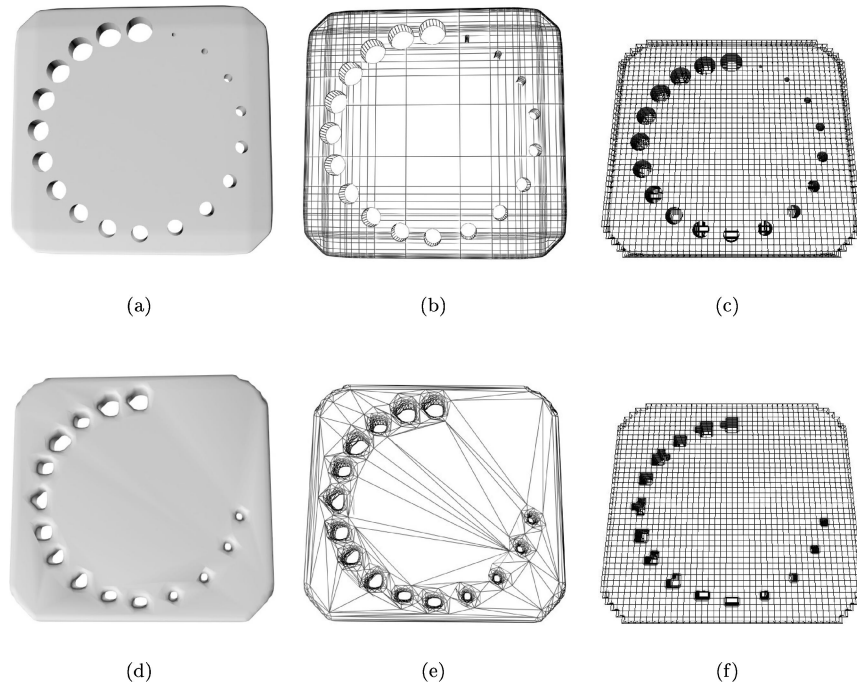


Fig. 9. Results of TDPS for the mechanical piece: (a–b) original model (14,832 t); (c) original model and its 7-level TG nodes; (d–e) simplified model (4,860 t); (f) surface reconstructed from the octree, before decimation.

the octree level by one unit causes the running times of the reconstruction and face reduction steps to be approximately multiplied by 4.

There are a few cases where the alternative reconstruction described in Section 4.5 is able to produce slightly better results in terms of both face reduction and sharp edge preservation. Particularly, the orthogonal reconstruction is suitable for those octrees having large linearly-separable regions, for instance, octrees corresponding to orthogonal and 2.5D solids.

Figure 11 shows the output of the orthogonal version of TDPS compared with the results of *Jade* which implements the incremental mesh simplification technique described in Ciampalini et al. [1997]. To draw the comparison, we had to triangulate both the mechanical piece (1,366 faces resulting in 2,728 triangles) and the models returned by TDPS. We have selected the Metro tool described in [Rochini et al. 1996] for computing the resulting approximation error. Figure 11 shows the results of each method with increasing tolerances. The model on the left is the original model with the color-coded approximation error. The error distribution is similar in most levels, although in very coarse levels the average error is greater on the TDPS output. Note that the Metro tool estimates the error from boundary points and TDPS is based on interior points, therefore Metro overestimates the solid-Hausdorff distance in topologically simplified regions of TDPS outputs.

TDPS produced, after triangulation, fewer triangles than mesh decimation in accurate approximations, but more triangles in coarse ones (first three columns of Table III). The key for explaining the results on the last two levels is found in validity preservation. TDPS generated valid two-manifold solids without self-intersections at all levels. The preservation of closed, non-self-intersecting boundaries slightly limits the amount of geometry reduction especially when working with high error tolerances. Running times on a MIPS R4600 SGI workstation are shown in the last columns of Table III.

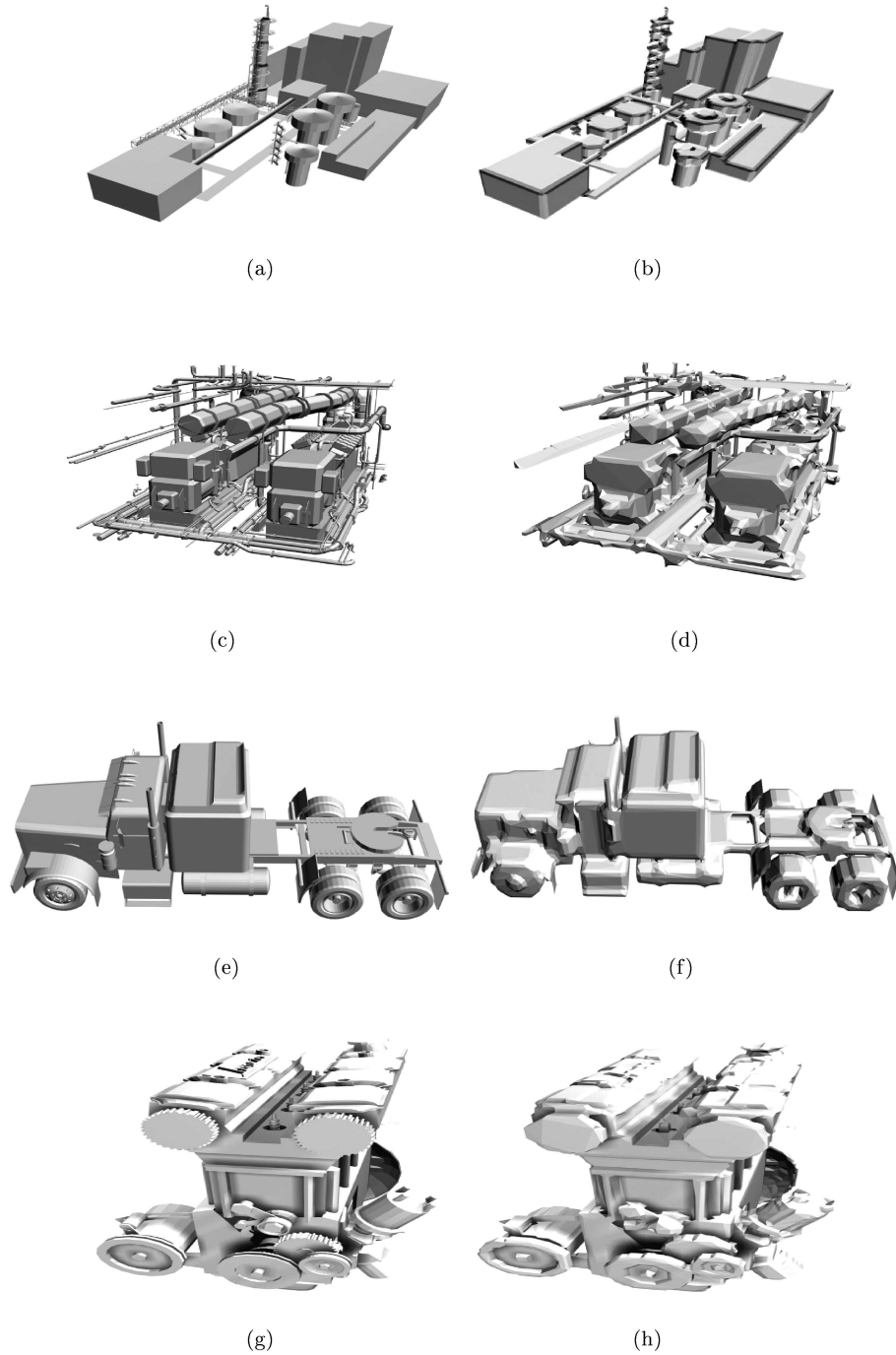


Fig. 10. Results of TDPS for complex test models: (a–b) factory model (20,794 t) and TDPS output (4,751 t); (c–d) ship model (34,532 t) and TDPS output (11,948 t); (e–f) truck model (47,817 t) and TDPS output (9,320 t); (g–h) engine model (140,113 t) and TDPS output (17,497 t).

Table II. Running Times of TDPS on the Test Models

Model	Octree depth	Discretization	Reconstruction	Face reduction	Total
Mechanical piece (14,832 f)	7	5.5 s	2.4 s	3.8 s	11.7 s
Basketballs (23,730 f)	8	12.1 s	14.3 s	19.5 s	45.9 s
Factory (20,794 f)	8	10.6 s	7.6 s	7.1 s	25.6 s
Ship piping (34,532 f)	8	22.9 s	9.3 s	11.7 s	43.9 s
Truck (47,817 f)	8	26.5 s	7.7 s	7.8 s	42.0 s
Engine (140,113 f)	9	118 s	66 s	64 s	248 s

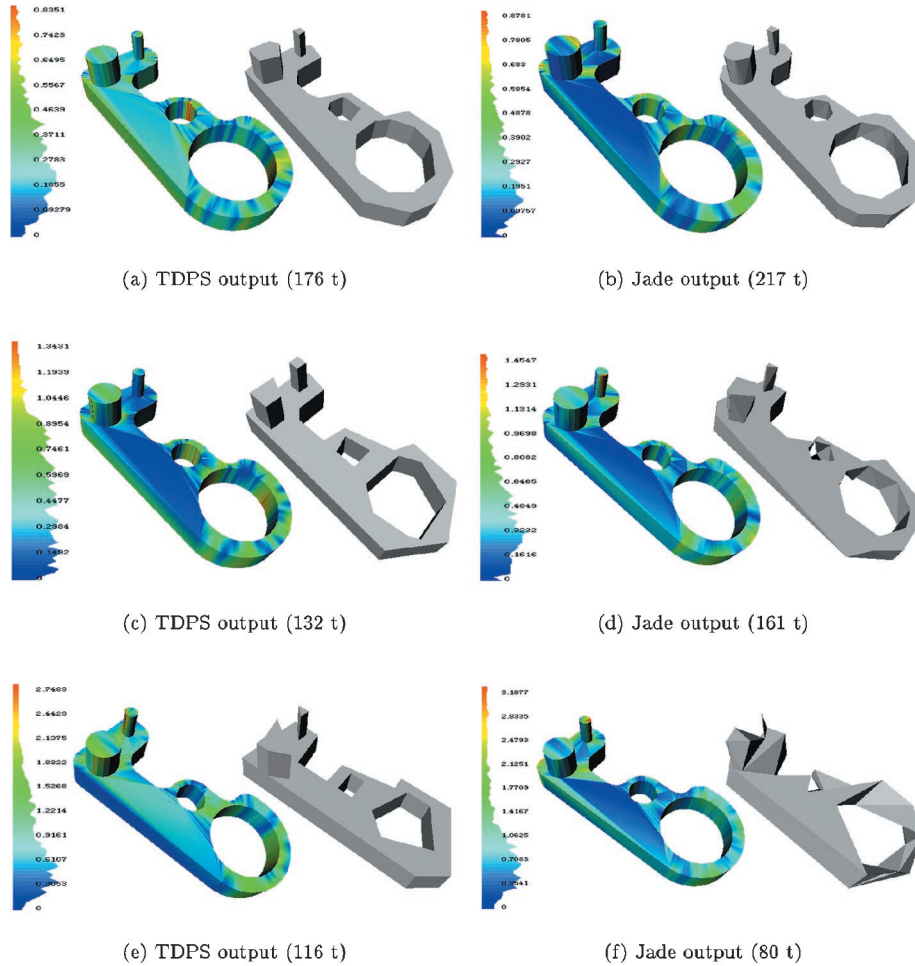


Fig. 11. Results of the orthogonal version of TDPS and Jade for the mechanical piece.

Note again that, in contrast to Ciampalini et al. [1997], our implementation of TDPS has running times inversely proportional to the tolerance, because its cost depends basically on the octree depth.

Figure 12 shows the results of the orthogonal version of TDPS on a VCR model with 25,396 faces and 68 shells, that has been selected to show the reduction ratio for very coarse approximations. Output approximations (151, 26 and 19 faces) involve non-triangular faces.

Table III. Results of TDPS and Jade on the Mechanical Piece

Depth	#f TDPS	#t Jade	Error param	TDPS times	Jade times
8	122 (248 t)	294	0.6%	1 min 52 s	53 s
7	86 (176 t)	217	1.2%	24 s	55 s
6	64 (132 t)	161	2.1%	6 s	1 min
5	56 (116 t)	80	4.5%	2 s	1 min 3 s
4	34 (68 t)	59	6.1%	0.6 s	1 min 5 s

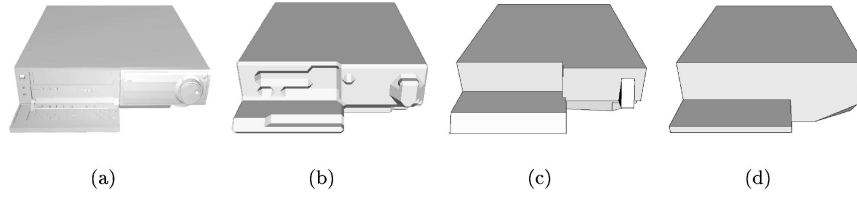


Fig. 12. TDPS results on the VCR model: (a) original model (25,396 f); (b–d) simplified models with 151, 26 and 19 faces.

Table IV. Comparison of TDPS with other Topology-Reducing Methods. VC Stands for Vertex-Clustering [Rossignac and Borrel 1993; Low and Tan 1997; Luebke and Erikson 1997] and pair-contraction methods [Garland and Heckbert 1997; Erikson and Minocha 1998, 1999; Popovic and Hoppe 1997; Shroeder 1997];  $\alpha$ -hulls refers to the El-Sana and Varshney [1998] method.

Feature	VC	$\alpha$ -hulls	He et al. [1995]	Ayala et al. [1995]	TDPS
Strict error bound	*	*		*	*
Aggregation	*		*	*	*
Genus simplification	*	*	*	*	*
Removal of internal structures					*
Sensitive to original tessellation	*	*			
Support to bounding approximations					*
Two-manifold output regardless of input			*	*	*

Table IV compares our approach with previously reported methods focusing on the main aspects of solid simplification. Vertex-clustering methods (represented by Rossignac and Borrel [1993] in Table IV) also rely on defining cells inside which the geometry is arbitrarily collapsed. However, geometry reconstruction in DPS methods is based on cell neighborhood instead of directly collapsing the input geometry, which is the key for guaranteeing valid two-manifold outputs. Pair-contraction methods allow a finer control for topology modifications, but these modifications are still restricted to regions with nearby vertices, and output models involve non-manifold vertices.

The simplification envelopes approach presented in Cohen et al. [1996] uses an offset surface that covers both sides of the input surface for error control. Offset surfaces are suitable for bounding the surface-Hausdorff distance but not for the solid-Hausdorff distance. In TDPS, such offset surfaces are replaced by the discrete *offset volumes* provided by the cells of the MDCO. These inner and outer bounds can be seen as a discrete version of the alpha-hulls proposed by El-Sana and Varshney [1998]. However, TDPS requires no heuristics to detect regions with holes and relevant cavities; it is completely insensitive to the tessellation of the input surface and it scales very well as the polygon count increases.

TDPS is specially suitable for LOD-based applications where surface appearance preservation is not as important as topology reduction, such as indirect illumination, collision detection, occlusion analysis,

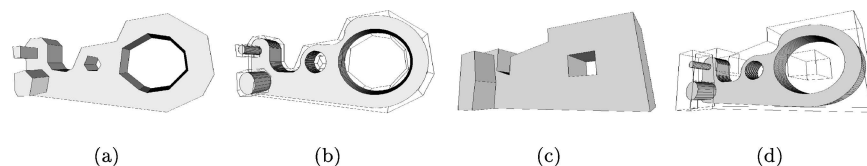


Fig. 13. Bounding approximations of the mechanical piece model: (a,c) bounding approximations with 46 and 21 faces; (b,d) original models (shaded) and their bounding approximations (wire).

multiresolution robust boolean operations, acoustic modeling, and query acceleration. In occlusion synthesis, TDPS provides a simple way of computing big occluders from a complex scene. Due to its ability to produce bounding and bounded approximations (Figure 13), it is also suitable for conservative query acceleration such as real-time collision detection. TDPS is not restricted to BRep inputs and it can simplify any discretizable model, although open surfaces not enclosing a 3D volume are reconstructed for both sides.

Regarding its limitations, TDPS is not intended for small error thresholds. The cost of the TDPS method largely depends on the maximum subdivision level of the MDCO, that is, on the error parameter. For very small error tolerances, a surface-metric simplification method might be better because not many topology changes would be expected and the running times of most surface simplification methods decrease with the tolerance.<sup>2</sup>

Due to the isothetic nature of the octree, TDPS is sensitive to octree alignment and orientation. Therefore, it is convenient to apply a linear transformation to the input model so that its main directions become axis-aligned [Wu 1992].

## 6. CONCLUSIONS AND FUTURE WORK

TDPS provides an efficient, general solution to the topology simplification problem including aggregation, genus reduction and removal of interior structures. TDPS has a number of advantages over previous simplification methods:

- It produces concise, error-bounded representations using the solid-Hausdorff distance. The octree provides a natural way of generating a multiresolution series of LOD representations.
- The generation of models that do not interpenetrate the original, either being completely bounded by or enclosed in the original solid is fully supported. This property is only provided by a few surface-based methods [Sander et al. 2000].
- It is not affected by some degeneracies of the input surface and produces two-manifold solids even with non-manifold inputs.
- It is suitable for simplifying both individual objects and assemblies. It can generate simplifications describing whole portions of the scene graph and thus it is suitable for hierarchical levels of detail and simplification culling [Erikson and Manocha 1998].

We plan to develop new decimation criteria adapted for the special properties of the polygonal meshes generated by TDPS. A hybrid method using the octree for topology reduction and the original model for guiding the face reduction step would improve the fitting to the original surface. We also plan to add surface appearance attributes, such as color, to the TERMINAL GREY nodes of the octree.

<sup>2</sup>However, running times of some incremental methods are quite independent of the tolerance since the computation of the queue prioritizing the entities to be collapsed dominates the overall cost.



## ACKNOWLEDGMENTS

The authors would like to thank Jarek Rossignac, Greg Turk and the anonymous reviewers for their helpful comments.

## REFERENCES

- ANDÚJAR, C. 1998. Space efficient connectivity test for n-dimensional images. *Comput. Graph.* 22, 4 (Aug.), 557–558.
- ANDÚJAR, C. 1999. Octree-based simplification of polyhedral solids. Ph.D. dissertation. Dept. LSI, Universitat Politècnica de Catalunya, Barcelona, Spain. [www.lsi.upc.es/~andujar](http://www.lsi.upc.es/~andujar).
- ANDÚJAR, C., AYALA, D., BRUNET, P., JOAN-ARINYO, R., AND SOLÉ, J. 1996. Automatic generation of multiresolution boundary representations. *Comput. Graph. Forum* 15, 3, 87–96.
- ANDÚJAR, C., SAONA-VÁZQUEZ, C., AND NAVAJO, I. 2000. LOD visibility culling and occluder synthesis. *Comput. Aided Des.* 32, 13, 773–784.
- AYALA, D., ANDÚJAR, C., AND BRUNET, P. 1995. Automatic simplification of orthogonal polyhedra. In *Proceedings of the International Workshop on Modelling, Virtual Worlds and Distributed Graphics (MVD'95)*, D. Fellner, Ed. Infix, 137–147.
- AYALA, D., BRUNET, P., JUAN, R., AND NAVAJO, I. 1985. Object representation by means of non-minimal division quadrees and octrees. *ACM Trans. Graph.* 4, 1, 41–59.
- BERG, M., VAN KREVELD, M., OVERMARS, M., AND SCHWARZKOPF, O. 1997. *Computational Geometry Algorithms and Applications*. Springer-Verlag, Berlin.
- BRUNET, P. AND NAVAJO, I. 1990. Solid representation and operation using extended octrees. *ACM Trans. Graph.* 9, 2, 170–197.
- CIAMPALINI, A., CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. 1997. Multiresolution decimation based on global error. *Vis. Comput.* 13, 5.
- CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. 1998. A comparison of mesh simplification algorithms. *Comput. Graph.* 22, 1 (Feb.), 37–54.
- COHEN, J., VARSHNEY, A., MANOCHA, D., TURK, G., WEBER, H., AGRAWAL, P., BROOKS, F., AND WRIGHT, W. 1996. Simplification Envelopes. In *Proceedings of ACM SIGGRAPH'96*, 119–128.
- ECK, M., DE ROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERY, M., AND STUETZLE, W. 1995. Multiresolution analysis of arbitrary meshes. In *Proceedings of ACM SIGGRAPH'95*, 173–182.
- EL-SANA, J. AND VARSHNEY, A. 1997. Controlled simplification of genus for polygonal models. In *Proceedings of IEEE Visualization'97*, R. Yagel and H. Hagen, Eds. 403–412.
- EL-SANA, J. AND VARSHNEY, A. 1998. Topology simplification for polygonal virtual environments. *IEEE Trans. Vis. Comput. Graph.* 4, 2 (Apr.).
- EL-SANA, J. AND VARSHNEY, A. 1999. Generalized view-dependent simplification. *Comput. Graph. Forum* 18, 83–94.
- ERIKSON, C. AND MANOCHA, D. 1998. Simplification culling of static and dynamic scene graphs. Tech. Rep. TR98-009, Department of Computer Science, University of North Carolina—Chapel Hill. Mar. 16.
- ERIKSON, C. AND MANOCHA, D. 1999. GAPS: Genoral and automatic polygonal simplification. In *proceedings of the Conference on the 1999 Symposium on interactive 3D Graphics*, S. N. Spencer, Ed, ACM Press, New York.
- FUCHS, H., LEVOY, M., AND PIZER, S. M. 1989. Interactive visualization of 3-D medical data. *Computer* 22, 8 (Aug.), 46–51.
- GARLAND, M. AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *proceedings of ACM SIGGRAPH'97*. Addison Wesley, 209–216.
- GROSS, M. H., STAADT, O. G., AND GATTI, R. 1996. Efficient triangular surface approximations using wavelets and quadtree data structure. *IEEE Trans. Vis. Comput. Graph.* 2, 2 (June).
- GUEZIEC, A. 1996. Surface simplification inside a tolerance volume. Tech. rep. Yorktown Heights, NY 10598. Mar. IBM Research Report RC 20440.
- HAMANN, B. 1994. A data reduction scheme for triangulated surfaces. *Comput. Aided Geom. Des.* 11, 197–214.
- HE, T., HONG, L., KAUFMAN, A., VARSHNEY, A., AND WANG, S. 1995. Voxel based object simplification. In *Proceedings of Visualization'95*, G. Nielson and D. Silver, Eds. Atlanta, GA, 296–303.
- HE, T., HONG, L., VARSHNEY, A., AND WANG, S. W. 1996. Controlled topology simplification. *IEEE Trans. Vis. Comput. Graph.* 2, 2, 171–184.
- HOPPE, H. 1996. Progressive meshes. *Computer Graphics* 30, Annual Conference Series, 99–108.
- HOPPE, H. 1997. View-dependent refinement of progressive meshes. In *SIGGRAPH 97 proceedings*. Addison Wesley, 189–198. ISBN 0-89791-896-7.
- KALVIN, A. D., CETTING, C. B., HADDAD, B., AND NOZ, M. E. 1991. Constructing topologically connected surfaces for the comprehensive analysis of 3D medical structures. In *Medical Imaging V: Image Processing*. Vol. 1445. SPIE.

- KALVIN, A. D. AND TAYLOR, R. H. 1996. Superfaces: Polygonal mesh simplification with bounded error. *IEEE Comput. Graph. Appl.* 16, 3, 64–77.
- LINDSTROM, P. AND TURK, G. 1999. Evaluation of Memoryless Simplification. *IEEE Trans. Vis. Comput. Graph.* 5, 2, 98–115.
- LORENSEN, W. AND CLINE, H. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *Comput. Graph.* 21, 4.
- LOW, K.-L. AND TAN, T.-S. 1997. Model simplification using vertex-clustering. In *Proceedings of the Symposium on Interactive 3D Graphics*. ACM Press, New York.
- LUEBKE, D. AND ERIKSON, C. 1997. View-dependent simplification of arbitrary polygonal environments. In *Proceedings of ACM SIGGRAPH'97*, T. Whitted, Ed. Addison Wesley, 199–208.
- MONTANI, C., SCATENI, R., AND SCOPIGNO, R. 1994. Discretized Marching Cubes. In *Proceedings of Visualization'94*. IEEE Computer Society Press, 281–287.
- MONTANI, C. AND SCOPIGNO, R. 1991. Quadtree/Octree-to-Boundary Conversion. In *Graphics Gems II*, J. R. Arvo, Ed. Academic Press Professional, Boston, MA.
- OHTAKE, Y. AND BELYAEV, A. 2001. Mesh optimization for polygonized isosurfaces. *Comput. Graph. Forum* 20, 3.
- POPOVIC, J. AND HOPPE, H. 1997. Progressive simplicial complexes. In *ACM SIGGRAPH 97 Conference Proceedings*, T. Whitted, Ed. Annual Conference Series. ACM SIGGRAPH, Addison Wesley, 217–224.
- ROCCHINI, C., CIGNONI, P., AND SCOPIGNO, R. 1996. Metro: measuring error on simplified surfaces. Tech. rep., Istituto I.E.I.-C.N.R., Pisa, Italy. B4-01-01-96.
- RONFARD, R. P. AND ROSSIGNAC, J. 1996. Full-range approximation of triangulated polyhedra. *Comput. Graph. Forum* 15, 3 (Sept.), C67, C76, C462.
- ROSSIGNAC, J. AND BORREL, P. 1993. Multiresolution 3D approximations for rendering complex scenes. In *Modeling in Computer Graphics*. Springer-Verlag.
- RUSHMEIER, H. E., PATTERSON, C., AND VEERASAMY, A. 1993. Geometric simplification for indirect illumination calculations. In *Proceedings of Graphics Interface'93*. Canadian Inf. Proc. Soc., Toronto, Ontario, 227–236.
- SAMET, H. 1990. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, Reading, MA.
- SANDER, P. V., GU, X., GORTLER, S. J., HOPPE, H., AND SNYDER, J. 2000. Silhouette clipping. In *Proceedings of ACM SIGGRAPH'2000*, 327–334.
- SCHROEDER, W. J. 1997. A topology modifying progressive decimation algorithm. In *Proceedings of IEEE Visualization'97*, R. Yagel and H. Hagen, Eds. IEEE, 205–212.
- SCHROEDER, W. J., ZARGE, J. A., AND LORENSEN, W. E. 1992. Decimation of triangle meshes. In *Proceedings of ACM SIGGRAPH'92*.
- SOUICY, M. AND LAURENDEAU, D. 1996. Multiresolution surface modeling based on hierarchical triangulation. *Computer Vision and Image Understanding: CVIU* 63, 1 (Jan.), 1–14.
- VARSHNEY, A. 1997. A hierarchy of techniques for simplifying polygonal models. In *Course notes of ACM SIGGRAPH'97*. ACM Press.
- VELTKAMP, R. C. 1992. Closed object boundaries from scattered points. Ph.D. dissertation, Erasmus University Rotterdam.
- WU, X. 1992. A linear-time simple bounding volume algorithm. In *Graphics Gems III*, D. Kirk, Ed. Academic Press, 301–306.

Received —; revised —; accepted —