Automatic Generation

of Multiresolution Boundary Representations

C. Andújar
D. Ayala
P. Brunet
R. Joan-Arinyo
J. Solé

Report LSI-96-2-R

Automatic Generation of Multiresolution Boundary Representations

C. Andújar, D. Ayala, P. Brunet, R. Joan-Arinyo, and J. Solé
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Av. Diagonal 647, 8a, E–08028 Barcelona

Abstract

The paper focuses on automatic simplification algorithms for the generation of a multiresolution family of solid models from an initial boundary representation of a polyhedral solid. An algorithm for general polyhedra based on an intermediate octree representation is proposed. Simplified elements of the multiresolution family approximate the initial solid within increasing tolerances. A discussion among different octree-based simplification methods and the standard marching cubes algorithm is finally presented.

Keywords: Geometric Modelling, Multiresolution, Level of Detail Representations, Simplification Algorithms, Virtual Reality

1 Introduction

Many computer graphics applications, such as virtual reality, require modelling and visualization of very large and geometrically complex systems. Interactive techniques for handling, operating and rendering these models require new and powerful algorithms. Multiple level of detail representations or multiresolution models are specially well suited for these purposes. The need for a family of decreasing complexity representations for the same object was already stated in [6]. Although multiresolution models are sometimes obtained interactively [10], extensive research is being done in algorithms for automatic generation of multiresolution representations.

Most of the current automatic simplification methods are based on triangular meshes. Some of them have been proposed for surfaces (see for instance [8, 17, 20]), whereas other schemes are suitable for the simplification of polyhedral solids, [7, 11, 16, 13], or general objects in voxel representations, [9]. Some of them have however several drawbacks concerning the output model, the lack of approximation bounds and the lack of topology simplification,

	Output Model	Topology Simplification	Error Bounds	Feature Preservation
Schroeder'92	$C^o, \triangle$	N	N	N
Turk'92	$C^o, \triangle$	N	N	N
Hoppe'93	$C^o, \triangle$	N	N	Y
Gross'95	$C^o, \triangle$	N	N	N
Eck'95	$C^o, \triangle$	N	Y	N
Rossignac'92	-	Y	Y	Y
Kalvin'93	$C^o, \triangle$	N	N	N
He'95	MC	Y	Y	N
Proposed	two-manifold	Y	Y	Y

Table 1: Comparative performance of different algorithms.

as shown in Table 1. All These methods are based on the direct computation of the whole set of simplified representations. Concerning the model of the output multiresolution objects, and referring to Table 1, almost all cited schemes produce simplified objects that are modelled by a mesh of C^0 triangles. Exceptions are the approaches reported in [9] and in [16]. The approache in [16] is very efficient in time but may produce non-regular and non-solid objects with isolated faces, edges and points. The approach in [9] models output objects through a marching cubes algorithm. Concerning the kind of simplificacion, all approaches except the two above are unable to simplify the topology of the object and reduce its genus. Concerning the existance of precise bounds on the approximation degree of the output objects, only the methods in [7, 9, 16] guarantee bounded approximations. Finally, only [16] and [11] preserve the features present in the initial object.

In this paper, an algorithm for to simplify general polyhedra automatically is proposed. The algorithm is based on the generation of an intermediate MDCO representation which is used to generate a multiresolution family of approximating polyhedra. The algorithm uses the multiresolution structure implicit in the hierarchical models. Two different approaches are presented. The first approach is based on an intermediate compaction of the MDCO into a face octree representation. The second approach is based on the generation of a feasible TG map. Both approaches are discussed. The paper is organized as follows. Next section includes the statement of the simplification problem and a top level description of the proposed approach. Sections 3 and 4 present respectively the proposed approaches. These algorithms are discussed through practical examples in section 5. The work presented here is a continuation of [2] where the general scheme for octree-based simplification was proposed, and of [1] where a specific algorithm for the case of orthogonal polyhedra was presented.

2 Statement of the Problem and Proposed Approach

In the next sections, we address the automatic simplification problem for general two-manifold polyhedra. The problem can be stated as follows: Given a general, two-manifold polyhedron P with $n_f(P)$ faces, a multiresolution family of two-manifold polyhedra P_1, P_2, ... P_l approximating the initial object P must be generated. The simplification algorithm must fulfil the following requirements,

- The approximation degree of the individual polyhedra P_k must be monotonically decreasing from the closest approximation P_l to P, to the coarser one P_1. More precisely, a set of tolerances $\epsilon_1, \epsilon_2, \ldots \epsilon_l$ with $\epsilon_k > \epsilon_{k+1}$ must exist such that the distance between the surface of P and the surface of P_k is bounded by ϵ_k; that is,

$$dist(surf(P), surf(P_k)) \leq \epsilon_k$$

- The number of faces of the polyhedra P_k must be monotonically decreasing from the best approximation P_l to P_1,

$$n_f(P_{k-1}) \leq n_f(P_k)$$

- Both the geometry and the topology - genus - of the initial polyhedron P must be simplified. Usually, P_1 will be a genus-0 bounded approximation to P.

- Relevant features of P such as sharp edges must be kept as much as possible during the simplification sequence from P_l to P_1.

- Flat regions of the initial polyhedron P must be approximated by large, planar faces in P_k whenever possible.

The proposed approach uses an intermediate MDCO octree structure which is pruned to different levels in order to obtain the multiresolution family elements P_k. The multiresolution structure which is implicit in the intermediate octree model is used in the generation of the different level of detail solids. See Figure 1. An MDCO,[2], is a classical octree containing white, black and grey nodes, together with terminal grey nodes. In what follows we will refer to the terminal grey nodes by *TG nodes*. White nodes correspond to cubic regions completely outside the solid and black nodes correspond to cubic regions completely contained in the solid. Grey nodes contain part of the object boundary and therefore must be subdivided. Finally, TG nodes are grey nodes at the deepest allowed level of the tree and are not subdivided. An inmediate consequence of these definitions is that

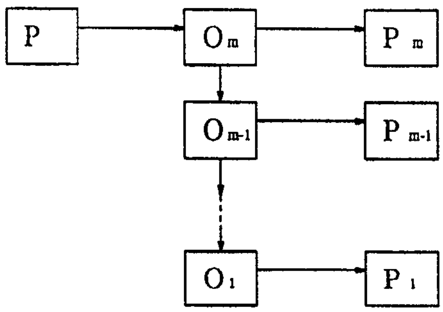


Figure 1: Scheme of the proposed approach.

the boundary of the object is completely contained in the region of the space corresponding to the set of TG nodes, [3]. TG nodes are represented by a particular code plus the color (white for outside, black for inside) of each of the eight vertices of the corresponding cubic region. It must be observed that, according to the above definitions, every TG node contains part of the boundary of the represented object. The colours of the eight vertices of TG nodes lead to 256 different combinations that can be grouped, using rotational symmetries and complements, into 14 equivalence classes, [19]. These classes are illustrated in Figure 2. Now, the proposed approach in this paper consists of three steps (see Figure 1),

1. The MDCO representation O_m of the initial polyhedron P is obtained. The MDCO generation is based on a simultaneous space subdivision and clipping of the boundary of P, [4]. The notation O_m stands for a m-level MDCO.

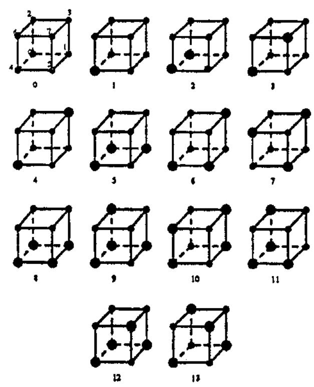


Figure 2: The 14 equivalence clases of TG nodes. Class 0 includes the numbering of vertices.

2. A multiresolution family of octree representations $O_1, O_2, \ldots O_m$ of P is obtained. Every O_k is the result of a simple pruning operation on the deepest level of O_{k+1}. Grey nodes at the k-level of O_{k+1} become the new TG nodes of O_k. See Figure 1.

3. Finally a feasible polyhedron P_k is reconstructed from every intermediate MDCO, O_k. A two-manifold polyhedron P_k is said to be feasible with respect to a MDCO O_k if the k-level MDCO representation of P_k is O_k. In other words, the boundary of P_k must be completely contained in the space region corresponding to the TG nodes of O_k.

An inmediate consequence of this approach is that a monotonically decreasing degree of approximation is automatically obtained:

$$dist(surf(P), surf(P_k)) \leq \epsilon_k = \epsilon \; 2^{m-k}$$

Where ϵ is the length of the main diagonal of the O_l TG nodes.

Steps 1 and 2 in the algorithm are straightforward and need no further comments. A first possibility for step 3 is the marching cubes algorithm, [9], although in this case the resulting boundary representations are too verbose in much cases as it is shown in section 5. Next two sections present two different approaches for step 3: section 3, presents a method based on an intermediate face octree compaction of the MDCOs O_k and section 4 presents a method based on the iterative detection of planar, connected TG regions.

3 Simplification Based on Face Octrees

Assume that a MDCO, O_k, is given. A method to generate a feasible polyhedron with respect O_k is based in the construction of a polygonal boundary using a face octree as an auxiliar data structure. The algorithm has two major steps. First a set of non planar quadrilaterals that covers the set of TG nodes in O_k is generated. Then a valid boundary representation is derived from the set of quadrilaterals with the help of a face octree extracted from the set of TG nodes in O_k.

3.1 Generating a Covering of the TG nodes Set

The set of quadrilaterals covering the TG nodes is computed in the form of a geometrically deformed model (GDM), [14]. It is an elastic network shaped by a set of points as follows. We start by associating a point with each TG node; we shall call these points the TG node *representative points.* Next each representative point is placed in the node centre and a mesh of quadrilaterals is generated by defining an edge between each representative point and the points that represent a face neighbour of the considered TG

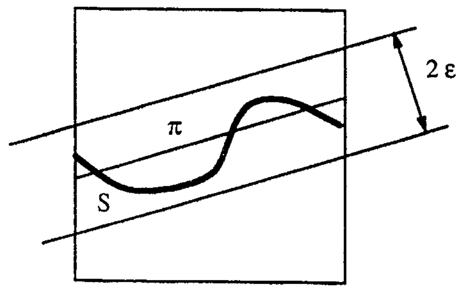


Figure 3: Face node and associated band.

node. A cost function is associated with every representative point in the network and each representative point is constrained to stay inside the limits of the TG node it represents. By minimization of the cost function while the constraints are satisfied, the network is deformed to fit the set of TG nodes, [12]. The result is a set of deformed, non planar quadrilaterals that cover the TG nodes.

3.2 Generating the Boundary Representation

A boundary representation can be derived from the quadrilaterals already computed making use of information about local planarity of the approximated surface represented by means of a *face octree.*

A *face octree* is an octree with white, black, face and grey nodes, and a tolerance ϵ that controls the degree of approximation of the representation, [3], [5]. White nodes represent regions outside the solid, black nodes represent regions inside the solid. Face nodes contain a connected part of the object boundary and each of them has an associated plane, π, that approximates the boundary S within the node with a given tolerance ϵ (see Figure 3). Grey nodes are those nodes that cannot be labeled as white, black or face nodes. Face octrees are halfway between classical and extended octrees; they are more concise than classical octrees, and they are well suited to approximate representations of objects with complex surface boundaries, [3, 5, 15]. Here, a face octree is computed by applying the procedure reported in [12] to the GDM obtained in section 3.1. The face octree is generated in such a way that its tolerance ϵ is equal to the length of the main diagonal of TG nodes. The face octree represents the local planarity of the approximated surface.

Once the face octree is computed, for each face node in the face octree, those deformed quadrilaterals that are inside the node are projected on the plane associated with the node by projecting orthogonally the representative points. The projected quadrilaterals that are inside each face node are collapsed in one planar polygon defined by the subset of the projected rep-

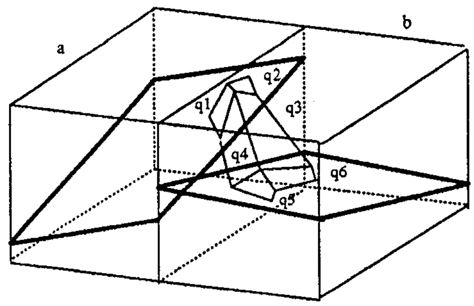


Figure 4: Projected quadrilaterals. Quadrilaterals q_1 and q_2 are inside node a. Quadrilaterals q_5 and q_6 are inside node b. Quadrilaterals q_3 and q_4 have vertices in both nodes.

resentative points such that at least one of the quadrilateral edges incident on it is partially inside and partially outside the face node. The set of non planar, deformed quadrilaterals that have different vertices inside different face octree nodes are split into two triangles. This completes the boundary representation. Figure 4 shows two face nodes in the octree that are neighbours. Quadrilaterals q_1 to q_6 are projected quadrilaterals; q_1 and q_2 are inside node a, q_5 and q_6 are inside node b; all of them are planar quadrilaterals embedded in the plane associated with the respective face octree node and will be included in the planar polygon. Quadrilaterals q_3 and q_4 have vertices in different face octree nodes, in general they are not planar and are split into two triangles. For an indepth discussion see [18].

The resulting boundary representation computed in this way approximates the initial set of TG nodes with a precision given by the tolerance of the auxiliar face octree, that is, the edge length of the initial TG node set, [12, 18].

4 Direct MDCO to Boundary Representation Conversion

In the particular case of orthogonal polyhedra, a simple, non-iterative algorithm for the MDCO to boundary representation conversion in step 3 of the algorithm, has been proposed by Ayala *et al.* in [1]. In this work we have designed and implemented an heuristic algorithm for the general case of two-manifold polyhedra. The procedure that converts MDCO to boundary representation in the step 3 of the algorithm is based on the concept of TG maps.

TG nodes in a MDCO can be classified into non-regular nodes that corresponds to class zero in Figure 2, and regular nodes which include classes 1 to 13. Regular nodes can be classified in turn into planar nodes (classes

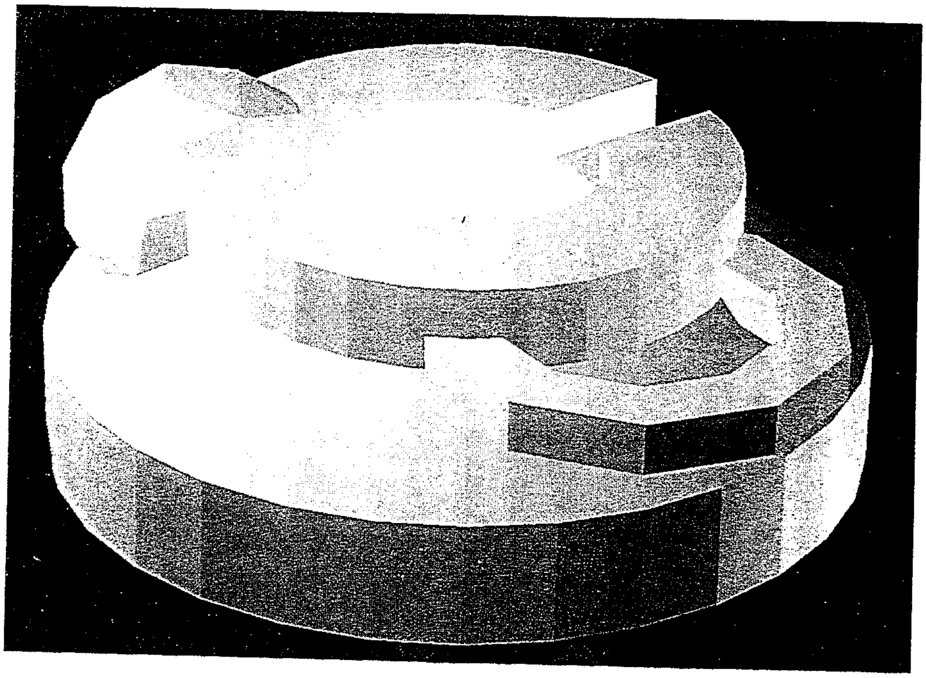


Figure 5: Object with 326 faces used to test the general algorithm.

1, 2, 5, 8 and 9) and non-planar nodes. In planar TG nodes, an splitting plane can be found such that separates black vertices from white vertices. Obviously, connected regions of planar nodes can contain single planar faces of the reconstructed polyhedron P_k.

A TG map is defined as a grouping of regular TG nodes of a MDCO into regions r_1, r_2, ... r_n. This grouping induces a classification of the TG nodes into *in*, *on* and *join* types which define the TG map. A TG node is classified as *in* if the node is inside any region r_i, that is, the node is $in(r_i)$. It is classified as *on* if it is on the boundary between two regions: it is $on(r_i)$ and $on(r_j)$. Finally, a TG node is classified as *join* if there exist at least three values i, j, k such that the node is $on(r_i)$, $on(r_j)$ and $on(r_k)$. There are many valid TG maps for a given MDCO: any grouping of TG nodes into connected regions leads to a different TG map.

Figures 5, 6 and 7 give an example of a TG map. Figure 5 shows the object on which the experiments have been carried out. Figure 6 shows the MDCO representations of octrees O_5, O_6 and O_7 of the test object. A TG map defined on the octree O_6 is presented in Figure 7. The upper part of the picture shows *in*, *on* and *join* nodes in white, green and red respectively. For the sake of clarity, only the *wireframe* of *on* and *join* nodes that separate different regions in the TG map are represented in the picture bottom.

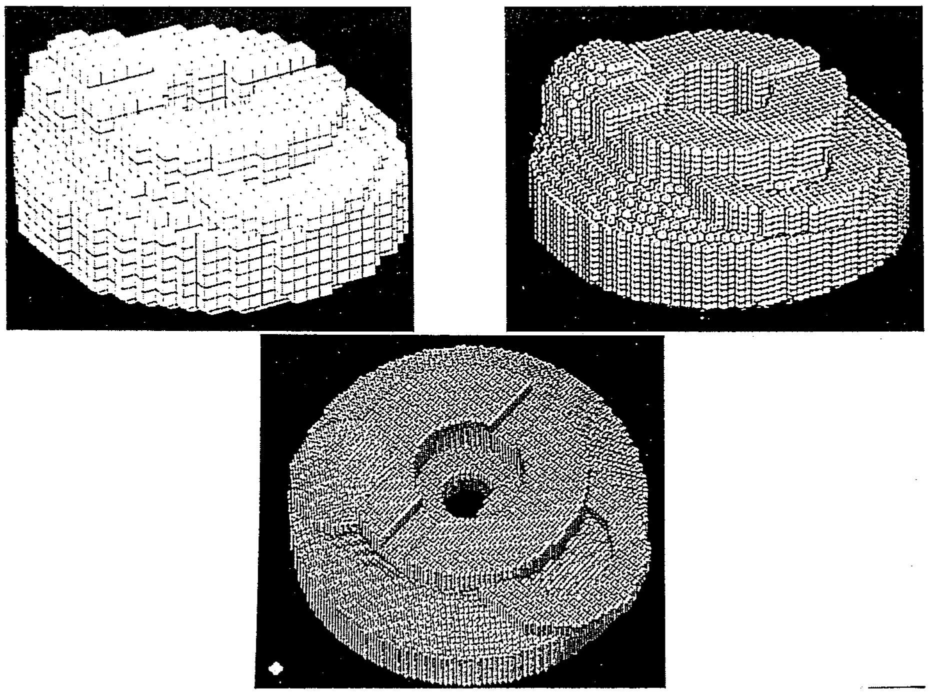


Figure 6: O_5, O_6 and O_7 MDCO representations of the object in Figure 5.

A feasible TG map is a TG map that defines a polyhedron contained in the corresponding MDCO. More precisely, a TG map defined on a MDCO, O_k, is said to be feasible if there exists a two manifold polyhedron P_k such that 1) The k-level MDCO representation of P_k is O_k; 2) every vertex of the polyhedron P_k is contained in a *join* node and every *join* node contains a vertex of P_k; and 3) every edge of the polyhedron is contained in *join* and *on* nodes and every *on* node contains part of an edge of P_k. From these conditions, it is obvious that every *in* node will contain a part of a face of the polyhedron.

Given these definitions, the present algorithm works following the scheme shown in Figure 8 and, starting from O_k, it generates an intermediate TG map from which the reconstructed polyhedron P_k is obtained.

This is inmediate at the deepest level m of the MDCO, as we already know that the resulting polyhedron P_m is the initial one P. For a general level k, The octree O_{k-1} is computed by a one-level pruning operation on O_k. Then, a first TG map of the $k-1$ level MDCO can be easily obtained through a classification of the O_{k-1} TG nodes with respect to the boundary of P_k (arrows b and c in Figure 8): nodes with one or more vertices of P_k are considered *join*, nodes with no vertex and containing part of one or

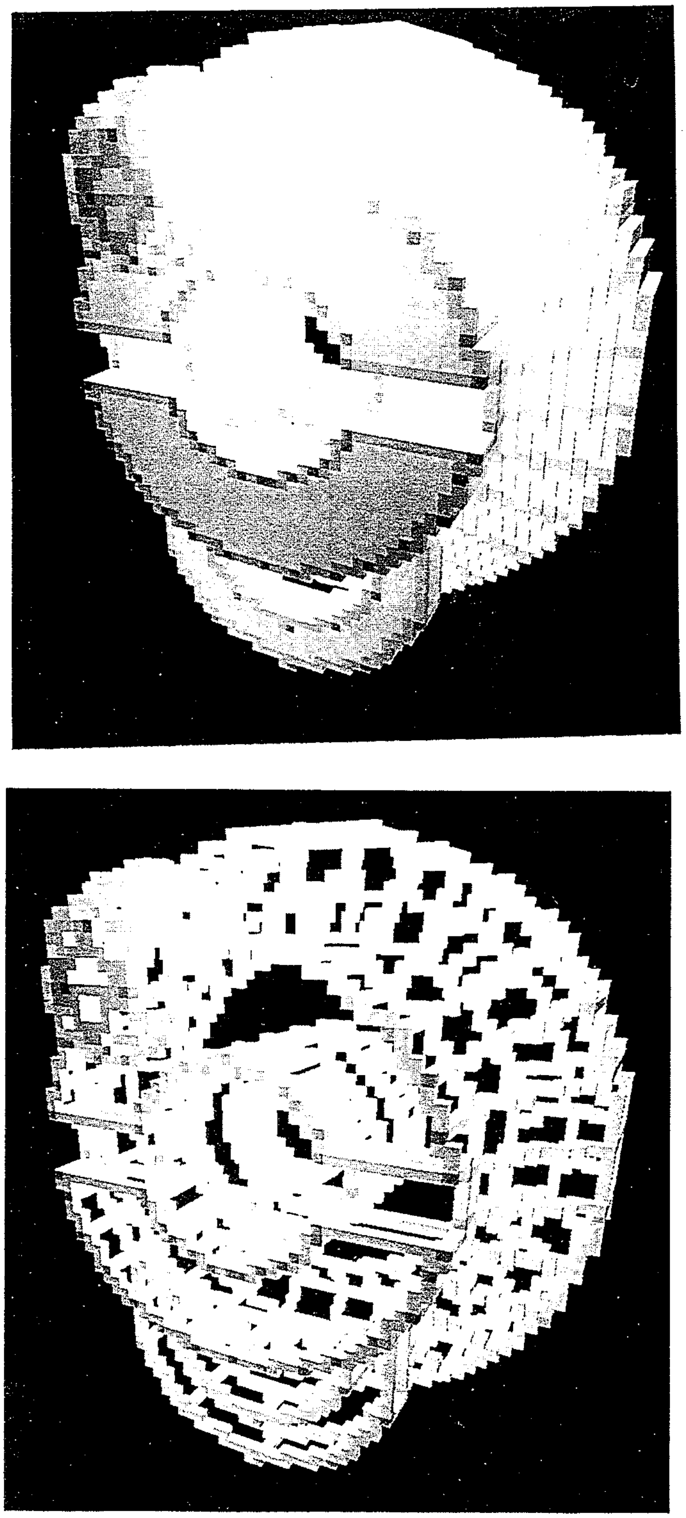


Figure 7: A feasible TG map defined on the octree O_6. The top part shows *in*, *on* and *join* nodes. The bottom shows only *on* and *join* nodes.

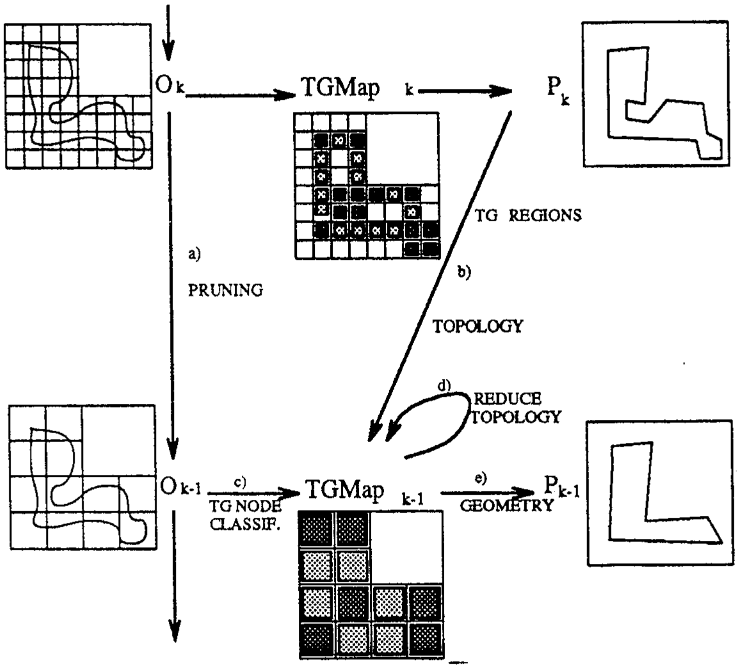


Figure 8: Scheme of the MDCO to boundary representation conversion algorithm for general, two-manyfold polyhedra.

more edges of P_k are considered *on*, and the rest of nodes - with faces of P_k - are considered *in*. Next, its topology is iteratively reduced (arrow d in Figure8) while the TG map is still feasible. The topology reduction of the TG map works on the TG map data structure by iteratively grouping pairs of neighbour planar regions in the TG map. It is worth to note that the topology reduction process is a discrete problem where the number of neighbour regions to be tested for possible groupings is finite and decreasing. When there is no possible region grouping that generates a feasible TG map, the polyhedron P_{k-1} is generated automatically as a by-product of the feasible function (arrow e in Figure 8). The structure of the proposed algorithm automatically guarantees a decreasing number of faces and a decreasing approximation on P_k.

The present version of the algorithm uses a weak, correct but not complete version of the feasible function that will be improved as part of the future work. Given the TG map defined on a MDCO O, for each region a plane that approximates the region is computed in such a way that it intersects all *join* type nodes of the region. If, for some region, it is not possible to figure out such a plane, then the TG map is not feasible. Once an approximating plane has been determined for each region, the approximating planes in every *join* type node are intersected in order to figure out

# levels	1	2	3	4	5	6	7	8	9	10	11	12	13
# divisions	1	2	4	8	16	32	64	128	256	512	1024	2048	4096
# faces	6	6	6	6	6	300	433	809	1779	2311	1984	2439	2439

Table 2: Hierarchy of approximations for an orthogonal building.

the vertices of a polyhedron Q. The TG map is said to be feasible if the MDCO representation of Q coincides with octree O. Non planar faces can be generated when more than three planes intersect within a *join* node. In this case, these faces are triangulated in order to compute the geometry of Q.

It must be observed that regions of the MDCO with non-regular (class 0) TG nodes represent parts of the space where the object has part of its boundary in every TG node but at the same time all vertices of all TG nodes are either inside or outside the solid. It is easy to see, [1], that any representation for P_k that ensures these two properties is admissible. Reconstruction of the polyhedron in non-regular zones may result in disconnected components (shells) of P_k in order to ensure the approximation bounds, [1]. This is acceptable for most applications, since the approximation guarantees the desired precision.

5 Results and Discussion

The results of the particular algorithm for orthogonal polyhedra, [1] are presented in Figure 9 and in Table 2. Figure 9 shows a building modeled from documents of the nineteenth-century architect Ildefons Cerdà, who designed the modern Barcelona urban planning, and the approximate simplifications P_10, P_8 and P_7. Table 2 contains the corresponding number of faces $n_f(P_k)$ as a function of the level k. The approximation error in P_k is bounded by $d\ 2^{13-k}$, where d is the main diagonal of the cubes of the 4096 x 4096 x 4096 spatial decomposition of the initial volume. For a detailed discussion see [1]. Observe that, for very coarse approximations, P_k is simply a rectangular prism with six faces.

We have implemented a marching cubes algorithm to reconstruct P_k from the set of TG nodes of the corresponding MDCO, and we have tested the general algorithm with the object in Figure 9 which depicts a general polyhedron with 326 faces. Results obtained approximating the initial object by means of MDCOs with with four, five and six levels that shown in Table 3. The second column gives the number of non regular nodes, the third gives the numer of planar nodes, the fourth the number of non planar nodes and the fifth column the toal number of nodes. The two last columns contain respectively the number of faces of the reconstructed polyhedra P_k by applying the standard marching cubes algorithm to the TG nodes of the

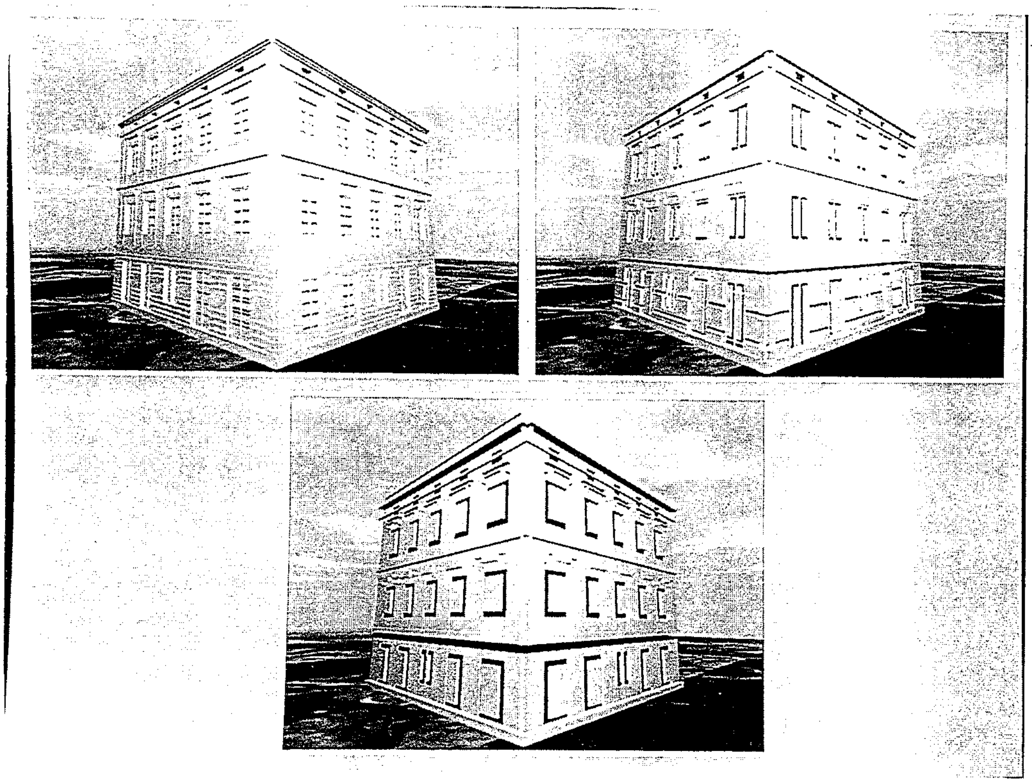


Figure 9: Simplifications of a real building with 2311, 809 and 433 faces.

MDCO, and the number of faces when the polyhedra is reconstructed using the face octree-based approach presented in section 3.

The improvement of the face octree-based simplification over the marching cubes results is quite obvious. The upper part of the Figure 10 presents the face octree compaction of O_5, O_6 and O_7 . Whether or not both face octree based and marching cubes approaches yield a real simplification depends to a large extent on the number of faces in the initial object, the larger the number the higher the simplification. In any case, the results

MDCO	Number of nodes of the MDCO				Faces of P_k	
	non regular	planar	non planar	total	Marching Cubes	Face octree
O_4	8	739	10	757	1504	685
O_5	621	3179	29	3229	6427	1351
O_6	33	12520	6	12559	25448	2596

Table 3: Number of nodes of the MDCO and number of faces resulting in different approximations to the object in Figure 9.

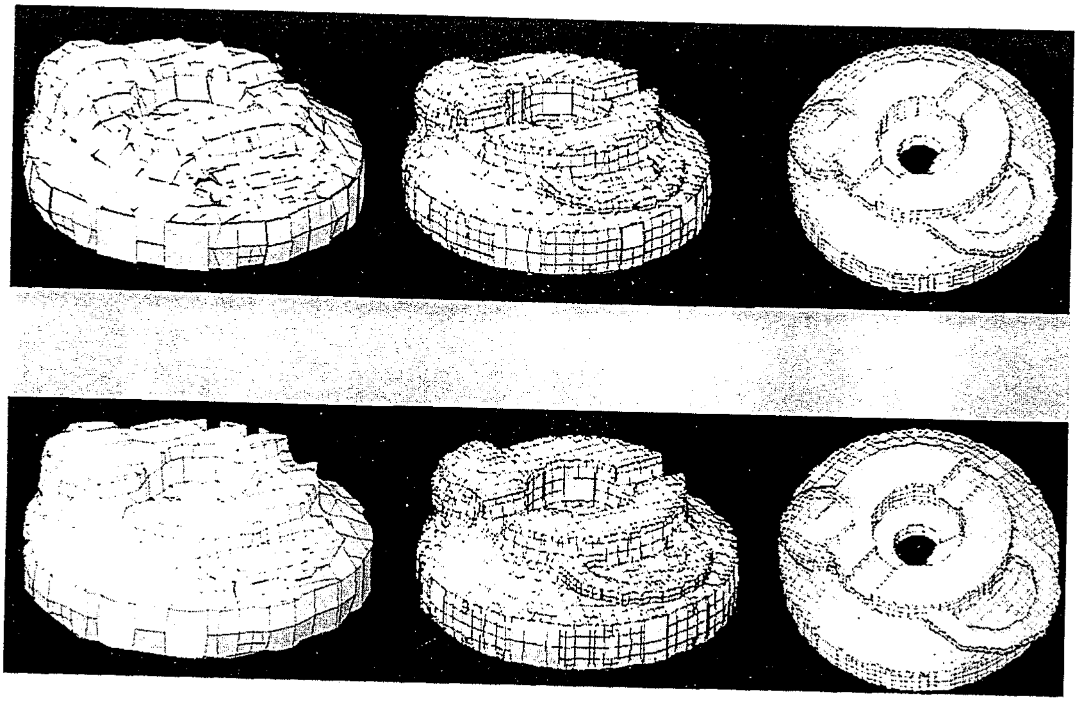


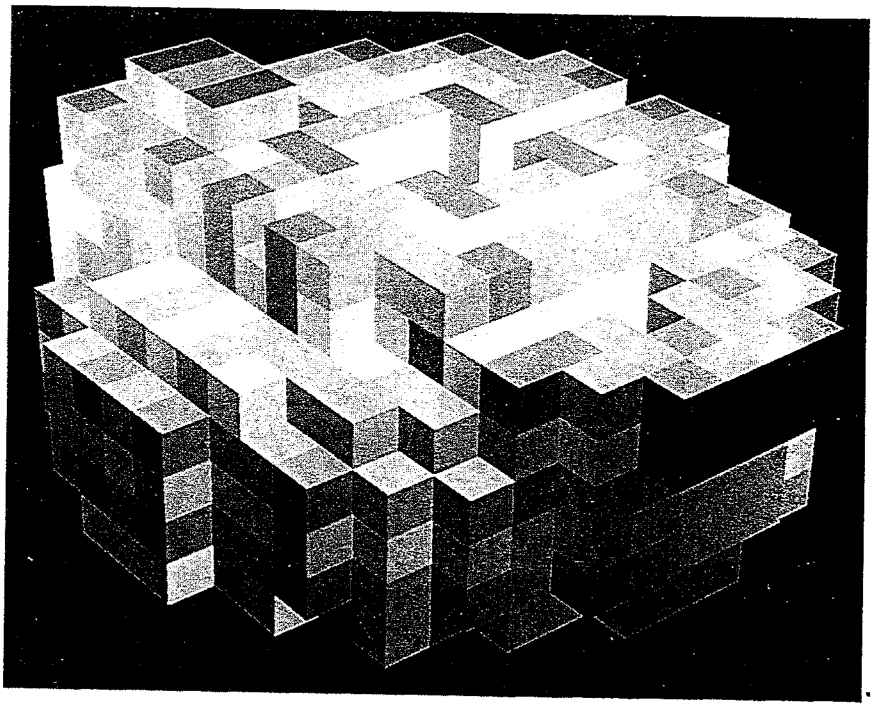
Figure 10: On top: face octree compaction of O_5, O_6 and O_7. On the bottom: Corresponding final boundary representation of the polyhedra P_5, P_6 and P_7.

obtained by the face octree approach are much better than those yielded by the marching cubes approach.

In the lower part of the Figure 10 the corresponding final boundary representations of polyhedra P_5, P_6 and P_7 are displayed. Finally, Figure 11 presents a feasible TG map defined on O_4, and the simplified polyhedron P_4 generated by the general algorithm. Polyhedron P_4 only has 89 faces, which clearly improves both the marching cubes and the face octree simplification results.

6 Conclusions and Future Work

In this paper, an algorithm for automatic simplification of general polyhedra has been presented. The algorithm is based on the generation of an intermediate MDCO representation which is used for the generation of a multiresolution family of approximating polyhedra. Two different approaches based on an intermediate compaction of the MDCO into a face octree representation and on the generation of a feasible TG map have been presented. The algorithm guarantees a bounded approximation based on the maximum



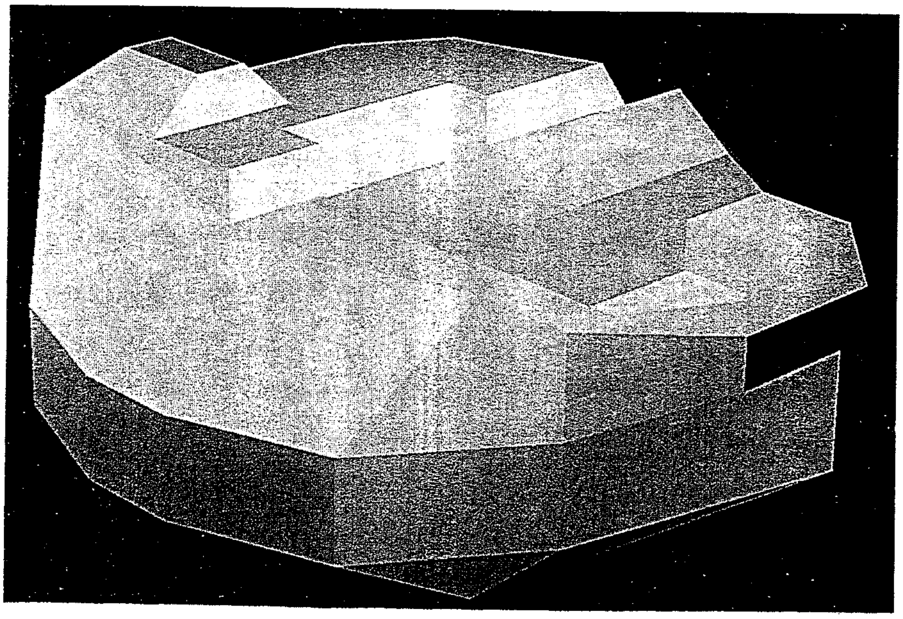


Figure 11: The feasible TG map defined on O_4 and the simplified polyhedron P_4 generated by the general algorithm.

surface-to-surface distance, and a decreasing degree of approximation in the multiresolution set. An automatic simplification of both geometry and topology is obtained, the coarser approximation being a simple 0-genus prism in most cases. The algorithm preserves relevant features like sharp edges and approximates flat regions by large, non-triangular faces. On the other hand, possible disconnected components within the approximation tolerances can be produced.

Future work includes devising more accurate algorithms for the general case, specially for the feasible function and for the detection of candidate neighbour regions for grouping TG nodes. The use of a finite set of discrete candidate planes for the generation of the geometry of feasible TG maps will also be investigated together with to which extent this discretization may affect the quality of the resulting approximations; in particular the effect on visualization artifacts generated by changes in the normal vector between subsequent approximations will be investigated. This approach will be compared with discrete marching cubes algorithms.

Alternative algorithms for the MDCO to boundary representations conversion will also be investigated and compared with those presented in sections 3 and 4 in this paper. In particular, seed algorithms similar to [7] may be specially interesting. Finally, and besides the analysis of improved algorithms for the reconstruction of non-regular zones, algorithms for the estimation of the initial polyhedron P_l in the case that P is unknown will be investigated. Possible approaches include face compaction in the discrete marching cubes reconstruction of the O_l TG map, and the use of face octree-based reconstruction.

Acknowledgements

The authors would like to thank Isabel Navazo and Imma Boada for their help in the implementation of the marching cubes algorithm and the corresponding simulations. They also would like to thank Alain Fournier for the valuable discussions on the use of discrete planes in the reconstruction algorithm.

This work has been supported in part by the Comision Interministerial de Ciencia y Tecnologia (CICYT), under the project TIC-95-0630.

References

[1] D. Ayala, C. Andújar, and P. Brunet. MDCO to BRep conversion algorithm. Technical Report LSI-95-16-R, Universitat Politècnica de Catalunya. Dept. Llenguatges i Sistemes Informàtics, 1995.

[2] D. Ayala and P. Brunet. Octree-based simplification of polyhedral solids. Technical Report LSI-95-1-R, Universitat Politècnica de Catalunya. Dept. Llenguatges i Sistemes Informàtics, 1995.

[3] P. Brunet. Face octrees: Involved algorithms and applications. Technical Report LSI–90–14, Universitat Politècnica de Catalunya, Department of LiSI, 1990.

[4] P. Brunet, R. Juan, I. Navazo, A. Puig, J. Solé, and D. Tost. *Scientific Visualization. Advances and challenges*, chapter Modeling and visualization through data compression, pages 157 – 170. Academic Press, 1988.

[5] P. Brunet, I. Navazo, and A. Vinacua. A modelling scheme for the approximate representation of closed surfaces. *Computing*, 8:75–90, 1993.

[6] J. H. Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19(10):547–554, 1976.

[7] M. Eck *et al.* Multiresolution analysis of arbitrary meshes. In R. Cook, editor, *Computer GraphicsSiggraph 95*, pages 173–182, Los Angeles, August 1995.

[8] M.H. Gross, R. Gatti, and O. Staad. Fast multiresolution surface meshing. In G.M. Nielson and D. Silver, editors, *Visualization'95*, pages 135–142. Atlanta, GA, October 29 – November 3 1995.

[9] T. He, L. Hong, A. Kaufman, A. Varshney, and S. Wang. Voxel based object simplification. In G.M. Nielson and D. Silver, editors, *Visualization'95*, pages 296–303, Atlanta, GA, October 29 – November 3 1995.

[10] P.S. Heckbert and M. Garland. Multiresolution modeling for fast rendering. pages 43 – 50. Graphics Interface'94, 1994.

[11] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *SIGGRAPH'93, Computer Graphics Proceedings, Annual Conference Series*, pages 19 – 26, 1993.

[12] R. Juan-Arinyo and J. Solé. Constructing face octrees from voxel-based volume representations. *Computer-Aided Design*, 27(10):783–791, 1995.

[13] A.D. Kalvin and R.H. Taylor. Superfaces: Polyhedral approximation with bounded error. pages 2 – 13, Newport Beach, CA, 1994. Medical Imaging 1994: Image Capture, Formatting, and Display, Proc. SPIE 2164.

[14] J.V. Miller, D.E. Breen, W.E. Lorensen, R.M. O'Bara, and M.J. Wozny. Geometrically deformed models: A method for extracting closed geometric models from volume data. *Computer Graphics*, 25, 4:217–226, 1991.

[15] N. Pla-Garcia. Recovering a smooth boundary representation from an edge quadtree and from a face octree. *Computer Graphics Forum*, 13(4):189–198, 1994.

[16] J. Rossignac and P. Borrel. Multiresolution 3D aproximations for rendering complex scenes. In B. Falcidieno and T.L. Kunii, editors, *Modeling in Computed Graphics*, pages 455 – 465. Springer-Verlag, 1993.

[17] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *Computer graphics*, 26(2):65 – 70, 1992.

[18] J. Solé. *Parallel Operations on Octree representations*. PhD thesis, Universitat Politècnica de Catalunya, February 1996.

[19] S.N. Srihari. Representation of three-dimensional digital images. *ACM Computing Surveys*, 13(4):399 – 424, 1981.

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

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Research Reports – 1996

LSI–95–55–R "A co-semidecision procedure for behavioral equivalence", Vicent-Ramon Palasí Lallana.

LSI–95–56–R "Reduction of behavioral equivalence to inductive theorems", Vicent-Ramon Palasí Lallana.

LSI–95–57–R "Where should concurrent rotations take place to rebalance a distributed arbitrary search tree?", Luc Bougé, Joaquim Gabarró and Xavier Messeguer.

LSI–95–58–R "Approximating convex quadratic programming is P-complete", Maria Serna and Fatos Xhafa.

LSI–95–59–R "Characterizing the learnability of Kolmogorov easy circuit expressions", José L. Balcázar and Harry Buhrman.

LSI–95–60–R "Compressibility and uniform complexity", Montserrat Hermo.

LSI–95–61–R "Implementació d'un mètode de planificació basat en l'actualització de vistes en bases de dades deductives", Dolors Costal (written in Catalan).

LSI–96–1–R "(Pure) Logic out of Probability", Ton Sales.

LSI–96–2–R "Automatic Generation of Multiresolution Boundary Representations", C. Andújar, D. Ayala, P. Brunet, R. Joan-Arinyo, and J. Solé.

Hardcopies of reports can be ordered from:

Nuria Sánchez
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Pau Gargallo, 5
08028 Barcelona, Spain
`secrelsi@lsi.upc.es`

See also the Department WWW pages, `http://www-lsi.upc.es/www/`