


**Multiresolution Approximation
of Polyhedral Solids**

D. Ayala
P. Brunet
R. Joan-Arinyo
I. Navazo

Report LSI-96-35-R

 **UPC**
Facultat d'Informàtica
de Barcelona - Biblioteca

17 JUN. 1996

Multiresolution Approximation of Polyhedral Solids

D. Ayala, P. Brunet, R. Joan-Arinyo, I. Navazo ¹

Multiresolution models with different levels of detail are a basic tool for handling very large and complex systems in computer aided design applications such as ship design. In this work, we discuss several automatic simplification algorithms that generate a multiresolution family of solid models that approximate with different levels of detail the boundary representation of a given polyhedral solid. Methods based on the simplification of triangular meshes are discussed together with octree based algorithms. Comparison criteria include the possibility of geometry and topology simplification, the availability of approximation bounds and feature preservation.

Introduction

Classical geometrical models used in present CAD systems present important limitations when used for modelling very large and geometrically complex systems. Integrated applications - in the automotive and aeronautic industries for instance - are asking for almost interactive rates in the modelling and visualization of very complex assemblies. Multiple level of detail (LOD) representations or multiresolution models are becoming a powerful tool for handling and modelling these systems, [HG94], as they can automatically support different simultaneous approximations in different modelled parts of the system, depending on the relative importance of the parts. A typical example is virtual reality navigation. In very complex systems, distant objects and parts have small projections and allow coarse approximations. An efficient virtual reality navigator must automatically switch among different LOD representations of the individ-

¹Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. 08028 Barcelona

ual objects, depending on their distance and visual relevance, [FS93]. A complementary technique to multiple LOD representations is the precomputation of visibility graphs of the polygon databases in order to reduce the complexity of the visibility computations, [TH93], [NV96].

Working with multiple LOD representations requires handling multiple representation databases and generating the family of the approximate representations from the geometric model of every modelled object. Of course, automatic data simplification is fundamental in these multiresolution representation approaches. Automatic data simplification deals with algorithms for obtaining a complete multiresolution (LOD) family of different approximating representations of an initial object. The need for a family of decreasing complexity representations for the same object was already stated by Clark in [Cla76]. Although multiresolution models are sometimes obtained interactively [HG94], extensive research is being performed in algorithms for the automatic generation of multiresolution models.

This Section presents several automatic simplification algorithms in the case of polyhedral models. Next section reviews several recent algorithms based on remeshing triangular meshes together with specific algorithms for - polyhedral - solids. The relevant aspects of the algorithms and their relative performances are presented and discussed. The rest of the chapter is devoted to octree-based simplification algorithms. A comparative analysis with respect to the schemes based on triangular meshes is included.

Previous Work

In this section, several proposed simplification algorithms are presented and discussed. We start by briefly describing several schemes that have been derived for piecewise triangular smooth surfaces, although they can be used for triangular-faced polyhedra.

Hamann's method [Ham94] simplifies a triangular mesh by eliminating triangles and re-triangulating the resulting region. First, the method computes principal curvature estimates for all vertices in the given triangulation [Ham93]. After that, a weight is assigned to each triangle based on the curvature at the vertices and used in such a way that triangles with low weight are first removed. Finally, each removed triangle is replaced by a point and the region affected by the removal of triangles is triangu-

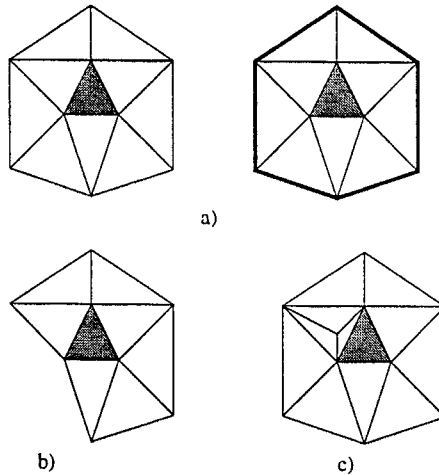


Figure 1: a) Triangle with its triangle platelet and its connected acyclic corona. The boundary polygon is also showed. b) A disconnected corona. c) A cyclic corona.

lated again. Only triangles surrounded by a connected and acyclic corona and satisfying the halfspace test can be removed. A *triangle platelet* P_i associated with a triangle T_i is the set of all triangles sharing at least one of T_i vertices. The *corona* C_i of P_i is the set of all triangles of P_i except T_i . A corona is *connected* if all its triangles are connected and is *cyclic* if there exists, at least, one cycle of three neighboring triangles. The *boundary polygon* is the external 3D contour defined by P_i (see Figure 1). The *halfspace test* guarantees that the centroid of the triangle T_i lies within the projection of the boundary polygon on the supporting plane of the triangle T_i . The weight assigned to a triangle is $\omega = \sigma\rho$. The angle weight σ is computed as $\sigma = 2(\sum \cos(\alpha_j) - 1)$, where $j \in \{1, 2, 3\}$ are the triangle interior angles, and the curvature weight ρ is given by the sum of the absolute curvatures at the triangle vertices.

Removing a triangle implies substituting its associated platelet by the corresponding boundary polygon and a new point which is computed considering the local surface geometry. Finally the re-triangulating is done by joining the new point with each vertex of the triangle platelet followed by an iterative edge swapping algorithm in order to optimize the angle weight. In this algorithm the degree of reduction can be specified either by a percentage or by an error tolerance.

The method proposed by Schroeder *et al.*, [SZL92], simplifies a triangular mesh by removing vertices and re-triangulizing the resulting region.

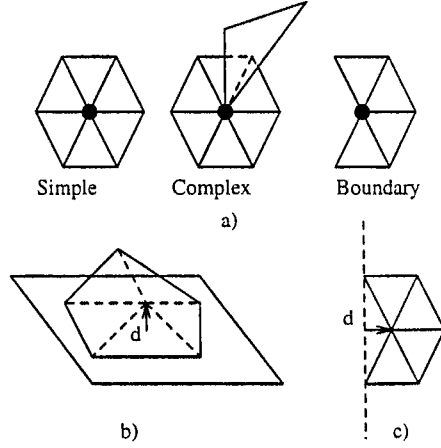


Figure 2: a) Vertex classification. b) Distance to plane criterion. c) Distance to edge criterion.

Vertices are classified into simple, boundary and complex (non-manifold). Figure 2a) shows this classification. Complex vertices cannot be removed and simple and boundary vertices are removed if they satisfy a specified criterion. Simple vertices use the distance-to-plane criterion. The distance between this vertex and an average plane, constructed using triangle normals, centers and areas of all triangles sharing this vertex, is computed and compared with a specified tolerance. Boundary vertices use the distance-to-edge criterion. In this case the distance between the vertex and the line defined by its two neighbor boundary vertices is computed and tested (see Figures 2b and c). After deleting a vertex and all its associated triangles, the resulting region is re-triangulated. The removal of a simple or boundary vertex reduces the mesh by two or one triangles respectively. This algorithm preserves the original topology of the mesh and can treat non-manifold forms. The degree of reduction can be specified either by a percentage or by an error tolerance.

Hoppe *et al.*, [HDD⁺93], present a method to solve an optimization problem in which a set of points, X , and an initial triangular mesh, M_0 , near them are the given input and a new reduced mesh M , that fits the set X well, is found. A mesh M is represented by its topology, K , and its geometry, V . Then the optimization problem can be solved by minimizing the following energy function:

$$E(K, V) = E_{dist}(K, V) + E_{rep}(K) + E_{spring}(K, V)$$

The first term is the sum of squared distances from the points of X to

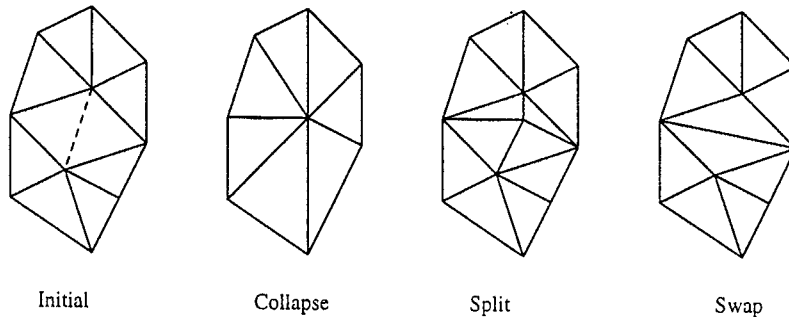


Figure 3: Legal moves: collapse, split and swap.

the mesh, the second term is proportional to the number of vertices of M and the third term is a spring energy. This minimization problem is partitioned into two subproblems: an inner minimization over V for a fixed K and an outer minimization over K . The inner minimization is a continuous optimization solved by a conjugate gradient method whereas the outer minimization is a discrete optimization based on the application of legal moves which are the three elemental transformations of edge collapse, split or swap between two neighboring triangles (see Figure 3). This method produces good results although it does not guarantee that a global minimum is found. It works as well for smooth surfaces as for surfaces with sharp edges so it preserves the initial model features.

Turk reported in [Tur92] a method that obtains a triangular mesh with a given number of vertices from an initial polyhedron. The algorithm consists of three steps. First, new points are distributed on the polyhedron's surface. These points are placed randomly and, after applying a neighbor repelling process, they become uniformly distributed. These new points will be the vertices of the final triangular mesh. In the second step a *mutual tessellation* surface is constructed that contains vertices of the initial polyhedron and the mentioned new points. This surface is obtained by triangulating each face of the initial polyhedron incorporating the new points lying on it. The third step removes old vertices. Each vertex to be deleted must fulfill the following condition. Let T be the boundary polygon of all the triangles sharing the vertex and let π be the average plane of these triangles. Then, only old vertices for which the projection of T on π does not self-intersect may be deleted. In practice, nearly all old vertices can be removed and a failing of this condition indicates that the vertex needs to be retained in order to preserve the topology of the initial object. The method is well suited for models with smooth shapes but is

poorly suited for models with well-defined corners and sharp edges.

Other schemes have been proposed specially for the simplification of closed polyhedra. Let us briefly describe some of them.

In [KT94], Kalvin *et al.* present a method that produces a simplified triangular mesh from an initial polyhedron. The method is based on *super-faces* generation and simplification. A super-face is a connected set of faces. The algorithm partitions the set of faces of the initial polyhedron into a set of super-faces and, after simplifying their contour, triangulates each super-face interior. Creating a super-face is done by a greedy process that begins with a randomly selected face and grows with faces that satisfy the specified merging criteria. These criteria check the location and orientation of the candidate face with respect to the super-face faces and also prevent super-faces from becoming too long or thin. Simplifying super-face contours (which are 3D polygons) is done by polygon simplification methods and taking into account the neighbouring super-faces. Finally the resulting simplified contours are triangulated. It is possible for two triangles from different super-faces to intersect. The method is based on a bounded approximation criterion and preserves the topology of the initial object. A serious drawback of this method is that it may produce a simplified polyhedron that self-intersects.

The algorithm from Rossignac and Borrel, [RB93], ensures an increasingly smaller number of faces and vertices in the sequence of 3D approximations to the original polyhedron. The scheme works in three steps: triangulation of faces, grading of vertices and clustering of vertices. In the first step, each face is decomposed into triangles supported by its original vertices. Then, the second step computes a weight for each vertex of the initial model, related to its relative perceptual - visual - importance. Priority is given to vertices bounding large faces and to vertices that are most likely to be on the object silhouette. The clustering of vertices embeds the initial object in a regular 3D rectangular mesh and computes a representative vertex for each group of vertices - cluster - in a particular cell of the mesh. The size of the mesh controls the approximation degree of the simplified object. Once vertices have been clustered and substituted by their representative, triangular faces are simplified. They can degenerate into edges or single points. A specific data structure supporting triangles, isolated edges and isolated points receives the geometry of the simplified objects. The authors claim that significant graphics performance improvements can be obtained by a postprocess that removes duplicates from the data structure. The algorithm is time efficient and does not require the complete BRep of the initial object, but a simple array of vertices and triangles. The approximation tolerance based on

the Hausdorff distance between the original and simplified model is increased during the simplification process while reducing the number of triangles. On the other hand, the method can simplify the topology of the object besides its geometry, and even merge an arbitrary number of small neighboring objects into a single point. The overall size of the object is approximately invariant during the simplification. The main drawback of the method is however that the output simplified objects are no longer polyhedra: the algorithm may produce non-regularized models with dangling faces and isolated edges and points.

The work from Gross, Gatti and Staad [GGS95] presents a remeshing algorithm for triangular meshes which controls the local level-of-detail of the surface approximation using local spectral estimates. It is presented as a 2D remeshing method, but it can be extended to 3D isosurface meshes. In the algorithm, the initial regular surface data grid is first transformed into a quadtree structure, and the remeshing algorithm works by iteratively removing vertices of the mesh and performing a local retriangulation of the neighborhood of the removed vertex. The main contribution of the paper is in the way it is decided whether a particular vertex can be removed. The algorithm starts by performing a wavelet transform of the data and, at every iteration, the detail signal is reconstructed by means of an inverse wavelet transform. The amplitude of this detail signal is used as a criterion for deciding on the removal of points. Local retriangulations are based on direct access to look-up tables. Approximation bounds can be derived, based on the properties of the wavelet transforms. The method is however limited to regular grid topologies in its present form, and should be extended to general triangulations in order to be usable for 3D simplification applications.

The method by Eck *et al.*, [EDD⁺95], extends the previous work of Lounsbery on surface wavelets to the problem of approximating an arbitrary mesh by a simpler mesh with subdivision connectivity, and gets an algorithm for the automatic multiresolution approximation of arbitrary meshes. The key idea in the remeshing procedure is the construction of a continuous parametrization of an arbitrary mesh over a simple domain mesh. This is obtained through the generation of a Voronoi diagram on the arbitrary surface of the initial object. One of the main advantages of this scheme is the existence of a tolerance that bounds the approximation in the resampling process. The method is however unable to simplify the topology of the object: only a multiresolution geometry simplification of the triangular mesh is obtained.

The algorithm from He *et al.*, [HHK⁺95], is based on first sampling and low-pass filtering the object into a voxel model. In a second step, the

	Output Model	Topology Simplification	Error Bounds	Feature Preservation
Hamann'94	N	N		N
Schroeder'92	N	N	N	
Hoppe'93		N	N	Y
Turk'92		N	N	N
Kalvin'93		N	Y	N
Rossignac'93	—	Y	Y	Y
Gross'95		N	Y	N
Eck'95		N	Y	N
He'95	MC	Y	Y	N

Table 1: Comparison of several simplification algorithms performances.

marching cubes algorithm is applied to a hierarchy of voxel volume representations in order to generate a multiresolution triangle-mesh family of approximate models of the initial object. The hierarchy of volume buffers is created by convolving the original object with a low-pass filter with a suitable spatial support. Different parametric filters can be considered, depending on the requirements of simplification speed and accuracy. In the second step, an approximate object isosurface is generated through a standard marching cubes algorithm, using up to five triangles to approximate the surface within a voxel. In addition to the multiresolution simplification algorithm, the authors also propose an antialiased rendering algorithm based on a multilayered marching cubes and on a translucency of the inner layers.

Table 1 presents a comparative analysis of the algorithms that have been discussed in this section. In all cases, the whole multiresolution set of simplified representations must be generated prior to their use. All algorithms presented in Table 1 require triangular meshes or triangular faced polyhedra as input. Concerning the output model, they usually generate conform meshes of triangular, planar faces. This is a limitation in cases with large flat faces, a triangulation being mandatory. The basic exception is Rossignac and Borrel's method, where a set of triangles, edges and points can be obtained. The scheme by He *et al.* generates triangulated objects as a result of using a marching cubes algorithm.

Concerning topology simplification, all algorithms based on remeshing techniques, except Rossignac's and He's, do not support topology simplification. This is an important requirement in practical applications, because otherwise, it will be impossible to get high simplification rates in

objects of complex topologies.

The existence of error bounds or tolerances for the approximations is also a basic user requirement. Approximating tolerances can be used for instance to detect, in a visualization application, the optimal approximate representation for a distant object. Algorithms that compute approximate models by sequential remeshing of the previous triangular mesh (the first four rows in Table 1) cannot guarantee an error bound with respect to the surface of the initial polyhedron. Finally, Table 1 shows that some of the proposed schemes preserve features - as sharp edges - in the initial object. This is also an interesting property that helps in having smooth transitions between successive objects of the approximating family.

Next section presents an octree-based algorithm for the automatic simplification of polyhedral objects. The algorithm is based on the generation of an intermediate octree representation that is subsequently used to generate the resulting multiresolution family. When compared to the schemes presented in this section, the new algorithm obtains two-manifold polyhedra as output models, it simplifies both the geometry and the topology -genus - of the initial polyhedron, an error bound for the approximation error can be derived, relevant features of the initial object such as sharp edges are preserved as much as possible during the simplification, and flat regions of the initial polyhedron are approximated by large, planar faces whenever possible.

Octrees Based Approach

The octree-based approach, [AAB⁺96], uses an intermediate octree structure which is pruned to different levels in order to obtain a multiresolution family P_1, P_2, \dots, P_m of two-manifold polyhedra approximating the initial polyhedral object P as illustrated in Figure 4. The algorithm consists on the following three steps, [AAB⁺96],

1. The maximal division classical octree (MDCO) representation O_m of the initial polyhedron P is obtained. See Figure 4. This conversion from P to O_m is based on a simultaneous space subdivision and clipping of the boundary faces of P . O_m stands for a m -level octree. The surface of P is completely contained in the Terminal Grey (TG) nodes of the octree O_m , TG nodes being the grey nodes without descendants, at the deepest level of the tree.

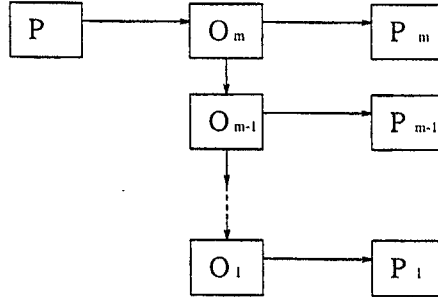


Figure 4: Multiresolution family of polyhedra.

2. A multiresolution family of octree representations O_1, O_2, \dots, 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 containing the whole boundary surface of P .
3. A feasible polyhedron P_k is reconstructed from every intermediate MDCO O_k . A polyhedron P_k is said to be feasible with respect to an octree O_k if the k -level MDCO representation of P_k is O_k . This is equivalent to the requirement of the boundary of P_k being completely contained in the space region corresponding to the TG nodes of O_k .

An immediate consequence of this approach is that a monotonically decreasing degree of approximation is automatically obtained, [AAB⁺96]

$$\text{dist}(\text{surf}(P), \text{surf}(P_k)) \leq \varepsilon_k = 2^{m-k} \varepsilon$$

On the other hand, the number of faces of the polyhedra P_k is monotonically decreasing from the best approximation P_m to the coarser P_1 , $n_f(P_{k-1}) \leq n_f(P_k)$.

Steps 1 and 2 in the simplification algorithm are straightforward. Concerning step 3, and besides the possible use of the marching cubes algorithm, [HHK⁺95], that will be discussed in the next section, several octree based reconstruction algorithms can be considered. They are briefly introduced in the next paragraphs.

In the particular case of orthogonal polyhedra with all planar faces parallel to one of the coordinate planes, a simple non-iterative algorithm for the reconstruction of a feasible polyhedron P_k from O_k can be derived, [AAB95]. The algorithm works by reconstructing the vertices of P_k which are located in specific TG node classes of O_k .

In the case of general polyhedra P , the face octree-based approach [AAB⁺96], [JAS95] first generates a set of non planar quadrilaterals that covers the set of TG nodes in O_k . The set of quadrilaterals is computed in the form of a geometrically deformed model, GDM [MBL⁺91]. Then, a face octree [BNV93] is computed from the GDM 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. In the final step, a valid boundary representation of P_k is derived from the set of quadrilaterals with the support of the face octree.

The TG map-based approach, [AAB⁺96] is also valid for general polyhedra P . It is based on an initial estimation and further refinement of the P_k topology. This topology is captured in the TG map data structure, with a first estimation from O_k and P_{k+1} , and is represented through TG node sets - or regions - of 6-connected nodes that are associated to the P_k faces. The TG map is afterwards refined and simplified by collapsing neighbour regions in planar zones, while being feasible. A feasible TG map is a TG map that defines a polyhedron P_k contained in the corresponding MDCO O_k . Figure 5 presents a polyhedron used as a test object and Figure 6 shows two feasible TG maps defined on its octree representations O_4 and O_6 . The above octree-based algorithms guarantee a bounded approximation based on the maximum surface-to-surface distance, and a decreasing degree of approximation in the multiresolution set. An automatic simplification of both the geometry and the topology is obtained, the coarser approximation being a simple 0-genus prism in most cases. The algorithms preserve relevant features such as sharp edges, and approximate flat regions by single large, non triangular faces. Their performance is discussed in the next section, based on the simplification of the object presented in Figure 5. Possible disconnected components within the approximation tolerances can be produced in some cases, however (see Figure 6).

Results and Discussion

It has been already mentioned in the discussion of results in Table 1, the interest of having topology simplification and error bounds for the approximations. Some of the basic requirements in many applications are not guaranteed by most of the existing automatic simplification algorithms but can be ring octree based schemes or similar methods.



Figure 5: Polyhedron used as test object.

Some results obtained for polyhedra with their faces oriented in three orthogonal directions, [AAB95], are presented in Figure 7. This figure shows a building modeled from documents of the 19-century architect Ildefons Cerdà, who designed the modern Barcelona urban planning. The approximate simplifications P_{10} , P_8 and P_7 are presented. The number of faces of these approximating polyhedra is 2311, 809 and 433 respectively. The approximation error in P_k is bounded by 2^{13-k} times the main diagonal of the elementary voxel cubes of the 4096 spatial decomposition of the initial universe volume. For a detailed discussion, see [AAB95]. It can be observed that polyhedra P_1 to P_5 are simply rectangular prisms with six faces.

The results of several algorithms in the simplification of the test object shown in Figure 5 which has 1856 faces, are presented in Table 2. Results obtained by approximating the initial object with MDCOs of 4, 5 and 6 levels are shown. The table includes the total number of nodes of the in-

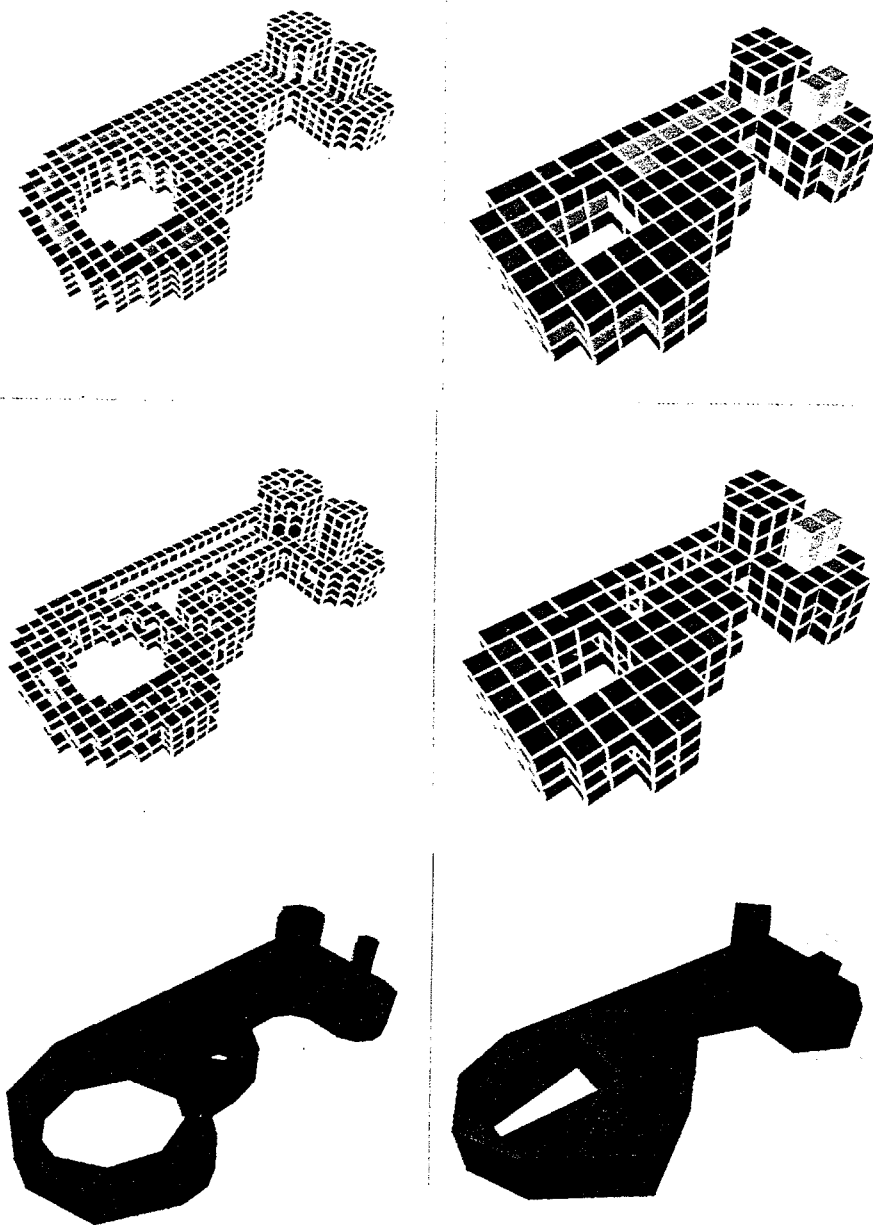


Figure 6: Left: A feasible TG map defined on the octree O_6 (the top picture shows *in*, *on* and *join* nodes whereas the center picture shows only the wireframe of *on* and *join* nodes) and the resulting simplified polyhedron (bottom picture, 65 faces). Right: The corresponding TG map and simplified polyhedron (33 faces) obtained from O_4 .

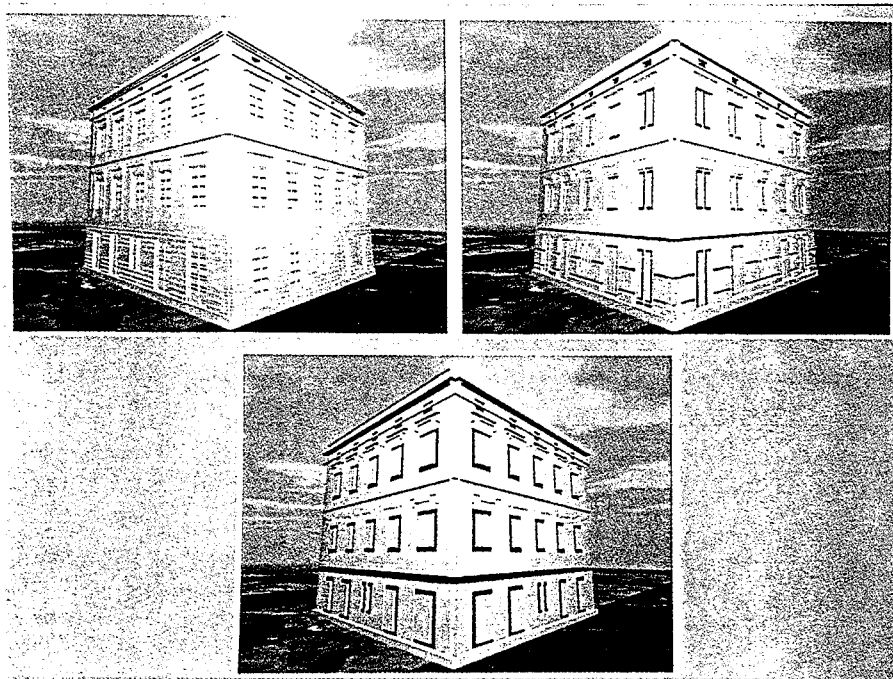


Figure 7: Simplifications of a real building.

intermediate octrees and the number of resulting faces in the reconstructed polyhedra P_k .

The improvement of the face octree based simplification over the marching cubes results is quite obvious. It must be observed, however, that whether or not both face octree based and marching cubes approaches yield a real simplification depends strongly on the number of faces of the initial object P , the larger the number yielding a higher simplification. Any object producing the same MDCO representation O_m will be approximated by the same polyhedra P_k in Table 2. On the other hand, the results in Figure 6 and Table 2 show that the TG map based simplification algorithm is a very promising technique.

Future work will include the design of more accurate algorithm for the TG map based scheme, specially focusing the derivation of improved fea-

MDCO	Number of nodes of the MDCO	Faces of P_k Marching Cubes	Faces of P_k Face Octree	Faces of P_k TG map
O_4	266	484	218	33
O_5	962	1904	433	48
O_6	4216	8379	852	65

Table 2: Number of nodes of the MDCO and number of faces of the resulting P_k using different approximation algorithms to the object in Figure 5.

sible functions. The use of a finite set of discrete candidate planes for the generation of the geometry of P_k will also be investigated, in order to reduce visualization artifacts produced by changes in the normal vector of the faces between subsequent approximations P_i and P_{i-1} . Alternative algorithms for the MDCO to boundary reconstruction must also be considered; seed algorithms similar to that reported in [EDD⁺95] may be specially interesting.

Acknowledgements

The authors would like to thank C. Andújar, Imma Boada and J. Solé for their help in programming the algorithms.

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

References

- [AAB95] D. Ayala, C. Andújar, and P. Brunet. Automatic simplification of orthogonal polyhedra. In D.W. Fellner, editor, *Modelling Virtual Worlds Distributed Graphics*, pages 137–147. International Workshop MVD'95, Infix, 1995.
- [AAB⁺96] C. Andújar, D. Ayala, P. Brunet, R. Joan-Arinyo, and J. Solé. Automatic generation of multiresolution boundary representations. *Computer Graphics Forum*, 1996. To appear.
- [BNV93] P. Brunet, I. Navazo, and A. Vinacua. A modelling scheme for

the approximate representation of closed surfaces. *Computing*, 8:75–90, 1993.

- [Cla76] J. H. Clark. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19(10):547–554, 1976.
- [EDD⁺95] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In R. Cook, editor, *Computer Graphics ACM SIGGRAPH 95*, pages 173–182, Los Angeles, August 1995.
- [FS93] T.A. Funkhouser and C.H. Sequin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In *Computer Graphics ACM SIGGRAPH 93*, pages 247–254, 1993.
- [GGS95] 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.
- [Ham93] B. Hamann. Curvature approximation for triangulated surfaces. In G. Farin, H. Hagen, H. Noltemeier, and W. Knödel, editors, *Geometric Modelling*, pages 139–153. Springer, 1993.
- [Ham94] B. Hamann. A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design*, 11:197–214, 1994.
- [HDD⁺93] 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.
- [HG94] P.S. Heckbert and M. Garland. Multiresolution modeling for fast rendering. In *Graphics Interface'94*, pages 43 – 50, 1994.
- [HHK⁺95] 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.
- [JAS95] R. Juan-Arinyo and J. Solé. Constructing face octrees from voxel-based volume representations. *Computer-Aided Design*, 27(10):783–791, 1995.

- [KT94] A.D. Kalvin and R.H. Taylor. Superfaces: Polyhedral approximation with bounded error. In Yongmin Kim, editor, *Medical Imaging 1994: Image Capture, Formatting, and Display*, pages 12 – 13, Newport Beach, CA, 1994. Proceedings SPIE 2164.
- [MBL⁺91] 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.
- [NV96] I. Navazo and S. Vila. Handling very complex environments using a discrete visibility graph. *This issue*, 1996.
- [RB93] J. Rossignac and P. Borrel. Multiresolution 3D approximations for rendering complex scenes. In B. Falcidieno and T.L. Kunii, editors, *Modeling in Computer Graphics*, pages 455 – 465. Springer-Verlag, 1993.
- [SZL92] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *Computer graphics*, 26(2):65 – 70, 1992.
- [TH93] S. Teller and P. Hanrahan. Global visibility algorithms for illumination computations. *Computer Graphics Annual Conference Series*, pages 239 – 246, 1993.
- [Tur92] 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-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.
- LSI-96-3-R “A Frame-Dependent Oracle for Linear Hierarchical Radiosity: A Step towards Frame-to-Frame Coherent Radiosity”, Ignacio Martin, Dani Tost, and Xavier Pueyo.
- LSI-96-4-R “Skip-Trees, an Alternative Data Structure to Skip-Lists in a Concurrent Approach”, Xavier Messeguer.
- LSI-96-5-R “Change of Belief in SKL Model Frames (Automatization Based on Analytic Tableaux)”, Matías Alvarado and Gustavo Núñez.
- LSI-96-6-R “Compressibility of Infinite Binary Sequences”, José L. Balcázar, Ricard Gavaldà, and Montserrat Hermo.
- LSI-96-7-R “A Proposal for Word Sense Disambiguation using Conceptual Distance”, Eneko Agirre and German Rigau.
- LSI-96-8-R “Word Sense Disambiguation Using Conceptual Density”, Eneko Agirre and German Rigau.
- LSI-96-9-R “Towards Learning a Constraint Grammar from Annotated Corpora Using Decision Trees”, Lluís Màrquez and Horacio Rodríguez.
- LSI-96-10-R “POS Tagging Using Relaxation Labelling”, Lluís Padró..
- LSI-96-11-R “Hybrid Techniques for Training HMM Part-of-Speech Taggers”, Ted Briscoe, Greg Grefenstette, Lluís Padró, and Iskander Serail.
- LSI-96-12-R “Using Bidirectional Chart Parsing for Corpus Analysis”, A. Ageno and H. Rodríguez.
- LSI-96-13-R “Limited Logical Belief Analysis”, Antonio Moreno.
- LSI-96-14-R “Logic as General Rationality: A Survey”, Ton Sales.
- LSI-96-15-R “A Syntactic Characterization of Bounded-Rank Decision Trees in Terms of Decision Lists”, Nicola Galesi.
- LSI-96-16-R “Algebraic Transformation of Unary Partial Algebras I: Double-Pushout Approach”, P. Burmeister, F. Rosselló, J. Torrens, and G. Valiente.

- LSI-96-17-R "Rewriting in Categories of Spans", Miquel Monserrat, Francesc Rosselló, Joan Torrens, and Gabriel Valiente.
- LSI-96-18-R "Strong Law for the Depth of Circuits", Tatsue Tsukiji and Fatos Xhafa.
- LSI-96-19-R "Learning Causal Networks from Data", Ramon Sangüesa i Solé.
- LSI-96-20-R "Boundary Generation from Voxel-based Volume Representations", R. Joan-Arinyo and J. Solé.
- LSI-96-21-R "Exact Learning of Subclasses of CDNF Formulas with Membership Queries", Carlos Domingo.
- LSI-96-22-R "Modeling the Thermal Behavior of Biosphere 2 in a Non-Controlled Environment Using Bond Graphs", Angela Nebot, François E. Cellier, and Francisco Mugica.
- LSI-96-23-R "Obtaining Synchronization-Free Code with Maximum Parallelism", Ricard Gavaldá, Eduard Ayguadé, and Jordi Torres.
- LSI-96-24-R "Memoisation of Categorical Proof Nets: Parallelism in Categorical Processing", Glyn Morrill.
- LSI-96-25-R "Decision Trees Have Approximate Fingerprints", Víctor Lavín and Vijay Raghavan.
- LSI-96-26-R "Visible Semantics: An Algebraic Semantics for Automatic Verification of Algorithms", Vicent-Ramon Palasí Lallana.
- LSI-96-27-R "Massively Parallel and Distributed Dictionaries on AVL and Brother Trees", Joaquim Gabarró and Xavier Messeguer.
- LSI-96-28-R "A Maple package for semidefinite programming", Fatos Xhafa and Gonzalo Navarro.
- LSI-96-29-R "Bounding the expected length of longest common subsequences and forests", Ricardo A. Baeza-Yates, Ricard Gavaldà, and Gonzalo Navarro.
- LSI-96-30-R "Parallel Computation: Models and Complexity Issues", Raymond Greenlaw and H. James Hoover.
- LSI-96-31-R "ParaDict, a Data Parallel Library for Dictionaries (Extended Abstract)", Joaquim Gabarró and Jordi Petit i Silvestre.
- LSI-96-32-R "Neural Networks as Pattern Recognition Systems", Lourdes Calderón.
- LSI-96-33-R "Semàntica externa: una variant interessant de la semàntica de comportament", (written in Catalan) Vicent-Ramon Palasí Lallana.
- LSI-96-34-R "Automatic verification of programs: algorithm ALICE", V.R. Palasí Lallana.
- LSI-96-35-R "Multiresolution Approximation of Polyhedral Solids", D. Ayala, P. Brunet, R. Joan-Arinyo, I. Navazo.

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/>