RC 17697 (#77951) 2/19/92
Computer Science 4 pages

Research Report

Multi-Resolution 3D Approximations for Rendering Complex Scenes

Jarek R. Rossignac and Paul Borrel

IBM Research Division
T. J. Watson Research Center
Yorktown Heights, NY 10598

Published as
"Multi-resolution 3D approximations for rendering complex scenes ", Jarek Rossignac and Paul Borrel.
In Geometric Modeling in Computer Graphics , pp. 455-465, Springer Verlag, Eds. B. Falcidieno and T.L. Kunii, Genova, Italy, June 28-July 2, 1993.

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents and will be distributed outside of IBM up to one year after the date indicated at the top of this page. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties).

IBM Research Division
Almaden • T.J. Watson • Tokyo • Zurich

Multi-resolution 3D approximations for rendering complex scenes

Jarek Rossignac and Paul Borrel

Interactive Geometric Modeling, IBM T.J. Watson Research Center, P.O. Box 704,
Yorktown Heights, New York 10598. Phone: 914-784-7630, Fax: -6273,
Email: jarek@watson.ibm.com and borrel@watson.ibm.com

Abstract

We present a simple and effective technique for approximating arbitrary polyhedra. It is based on triangulation and vertex-clustering, and produces a series of 3D approximations that resemble the original object from all viewpoints, but contain an increasingly smaller number of faces and vertices. Using the appropriate level of simplification when displaying small, distant, or background objects improves graphic performance without a significant loss of perceptual information, and thus enables realtime inspection of complex scenes.

CR Categories and Subject Descriptions: I.3.3 [Computer Graphics]: Picture/Image generation – *display algorithms;* I.3.5 [Computational Geometry and Object Modeling]: Solid Representation;

Additional Keywords and Phrases: Engineering Visualization, Simplification, Approximations, Data Compression.

1. Introduction

The interactive 3D navigation through scenes defined by millions of polygons is vital for industrial CAD applications, such as the design reviews for large mechanical assemblies. Yet, it cannot even be supported on emerging multi-processor high-end graphic servers (see [2] for recent progress). Since the galloping hardware developments will only stimulate the demand for support of even larger data-sets, a distinction must be made between (1) the accurate geometric models necessary for representing mechanical parts and (2) specialized representations of these parts tailored for efficient graphics.

Previously developed graphics performance improvements that deal with scene complexity by quickly eliminating objects that do not project on the screen, or by displaying crude approximations of objects whose projection is very small, are insufficient and exhibit important short-comings. Pre-computed hierarchical spatial directories [8] help to quickly prune portions of the model lying outside of the viewing space, but have no effect when the entire scene fits in that space. The graphics performance increases achieved by rendering isolated dots, mini-max boxes, or other geometrically simple bounds instead of distant or small objects are often offset by a significant loss of visual information and by distracting abrupt shape changes that occur when objects slowly approach the viewer. Geometric approximations with minimal rendering cost and a close resemblance with the original solids from all directions are the key to an acceptable solution for realtime graphics of complex scenes. Furthermore, a sequence of approximations offering different trade-offs between visual accuracy and graphic performance will prove effective for revealing details as objects approach the viewpoint, only if the transitions between one approximation and the next are barely noticeable.

Since a major factor of the shading cost for a polyhedron is the number of vertices that must be processed by the graphics pipeline when rendering the object's faces [7], approximations should strive to reduce the number of vertices and faces while preserving the overall aspect of the model.

We present here a new simplification technique that operates on boundary representations of an arbitrary polyhedron and generates a series of simplified models with a decreasing number of faces and vertices. The resulting models do not necessarily form valid boundaries of 3D regions—for example, an elongated solid may be approximated by a curve segment. However, the error introduced by the simplification is bounded (in the Hausdorf distance sense) by a user-controlled accuracy factor and the resulting shapes exhibit a remarkable visual fidelity considering the data-reduction ratios.

2. Alternative solutions

Most polyhedra are used to approximate more general shapes and are constructed as tesselations of curved faces. Approximations with fewer vertices and faces can thus in principle be produced by using coarser tesselation parameters. Indeed, emerging high-end graphic architectures support adaptive tesselation for trimmed NURBS surfaces [6]. However, in practice, polyhedral representations often result from Boolean operations (poorly supported for curved geometries) or from procedural models which are not available to the graphics system and are too expensive to regenerate in realtime. Consequently, re-tesselation of curved surfaces is not an acceptable alternative for simplifying the vast majority of existing polyhedral models.

Recursive difference techniques [4] replace a solid S by the difference, H – D, between its convex hull H and a delta solid, D = H – S, and apply recursively this process to (a subdivision of) the connected components of D. The resulting CSG tree may be truncated, replacing all delta solids at a given recursion depth d by their convex hull. The truncation removes cavity or protrusion details, depending on the parity of d. Unfortunately, this process does not simplify convex objects; tends to increase the number of faces; and, for small values of d, produces approximations whose overall shape considerably differs from the original solids.

Line simplification algorithms (also called "line generalization") used in Cartography [5] apply recursive subdivision to approximate a plane polygonal curve by a small number of vertices lying on that curve. The maximum deviation between the curve and an "anchor-floater" line joining the end-points is evaluated. If the deviation exceeds a given threshold, the curve is split at the maximal deviation point for further iterations, otherwise, the curve is replaced by the line segment. We have designed a 3D extension of this approach requiring the construction and triangulation of the topology of the ob-

ject's boundary. After considering its algorithmic complexity and storage requirements, we have opted for the simpler and more efficient solution described in the remainder of this paper. Fig. 1 illustrates the simplification process on a small assembly of mechanical parts.

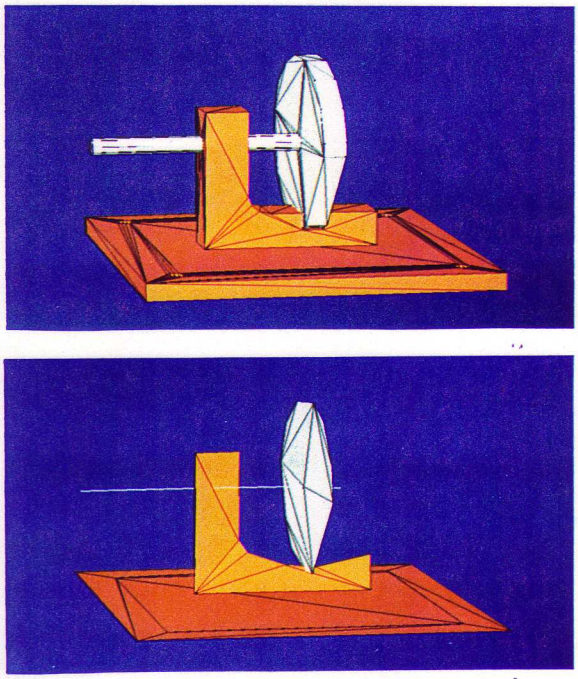

Fig. 1: The solids are triangulated (top) and simplified (bottom).

3. Single simplification process

The original model of each object is represented by a vertex table V containing vertex coordinates and a face table F containing references to V, sorted and organized according to the edge-loops bounding the face. The simplification involves the processing steps presented below and summarized in Fig. 2. They manipulate the data-structure of Fig. 3.

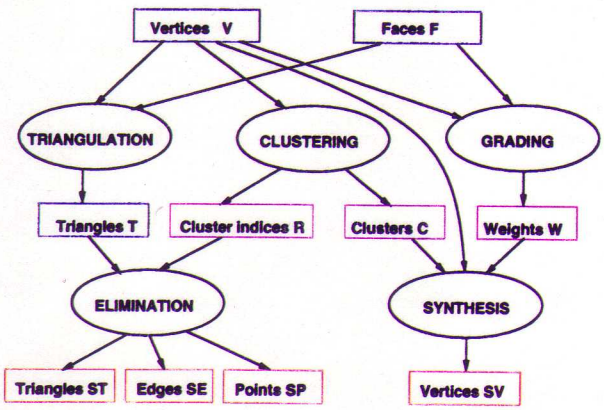

Fig. 2: Overview of the simplification process.

3.1 Grading

A weight is computed for each vertex of V and stored in the W table. The weight defines the relative perceptual importance of the vertex. The subjectivity of our approach is confined to the choice of criteria for weight evaluation. We favor (1) vertices that have a higher probability of lying on the object's silhouettes from an arbitrary viewing direction and (2) vertices that bound large faces that should not be affected by the removal of small details. The first factor may be efficiently estimated using the inverse of the maximum angle between all pairs of incident edges on the candidate vertex. The second factor may be estimated using the length of the longest among all of the edges incident upon the vertex.

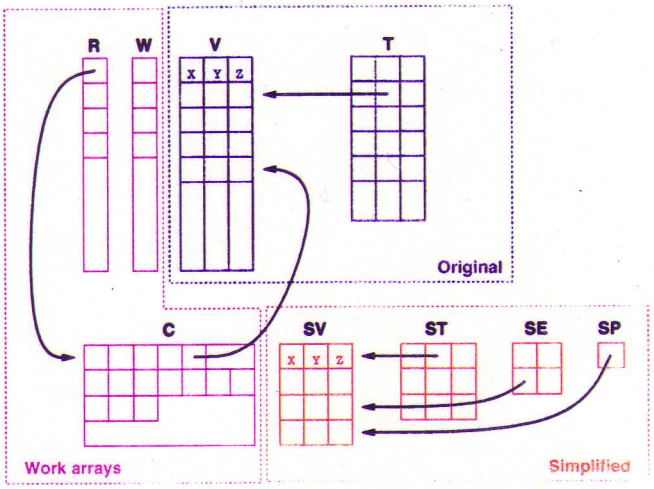

Fig. 3: Data-structures and their interdependencies.

3.2 Triangulation

Each face is decomposed into triangles supported by its original vertices. (We use a simple greedy cutting process to obtain nearly optimal results, although more efficient and optimal techniques are available [1]. The resulting T table contains 3 vertex-references per entry.

3.3 Clustering

Based on geometric proximity, the vertices of V are grouped into clusters. Clusters are numbered by the order in which they are created. The R table indicates, for each vertex of V, the corresponding cluster number. Conversely, the C table contains, for each cluster, a list of references to the vertices falling into that cluster. We have opted for a simple clustering process based on the truncation of vertex coordinates. A box, or other bound, containing the object is uniformly subdivided into cells. Vertices falling within one cell form a cluster and will be replaced by a unique vertex. The clustering procedure takes as parameters the box in which the clustering should occur and the maximum number of cells along each dimension. The solid's bounding box or a common box for the entire scene may be used.

3.4 Synthesis

For each cluster, a representative vertex is computed using the C, W, and V tables and is stored in the SV table of simplified vertices. For data smoothing, the representative vertex may be defined as the center of mass of all the vertices of the cluster weighted with the values stored in W. For removing details without perturbing retained faces, the vertex with maximal weight may be selected. The synthesis maps each vertex, V(i), into the representative vertex, SV(R(i)), of the corresponding

cluster. The mapping is typically many-to-one and thus reduces the total number of vertices.

3.5 Elimination

Table R maps the vertices of the original triangles into new representative vertices. When all three representative vertices are equal, the triangle degenerates into a point. When exactly two representative vertices are equal the triangle degenerates into an edge. Such edges and points, when they do not bound any other triangle of the simplified object, are stored in SE and SP tables, and rendered as part of the solid's approximation. Significant graphics performance improvements are obtained by removing, from the simplified model, all triangle-duplicates, all edge-duplicates, all point-duplicates, all edges bounding a triangle, all points bounding an edge, and all points bounding a triangle. The elimination process performs these tasks and produces three tables: the simplified triangles, ST; the dangling edges, SE; and the isolated points, SP. They respectively contain three, two, and one reference to entries in the SV table.

The search for duplicates uses a temporary data-structure, which associates, with each cluster, a list of incident edges and, with each edge, a list of triangles. Triangles are stored only once in this data-structure using their lowest-index vertex for locating the cluster, using the intermediate-index vertex for locating the edge incident upon that cluster, and using the last vertex for locating the triangle. Dangling edges are implicitly defined as edges with empty triangle-lists. Isolated points are defined as clusters with empty edge-lists.

3.6 Adjustment of normals

A final step computes new normals for all the triangles in ST using its simplified vertex coordinates. The normals are only used for rendering. Normals of back-facing triangles are automatically inverted by the graphics processor.

4. Series of increasing simplification

Approximate models of the same objects with increasing degree of simplification may be obtained by executing the above process several times, with decreasing clustering resolutions.

A more efficient approach performs a first simplification with the highest resolution, then recursively merges adjacent clusters into new ones to produce the other simplified models. An octtree used to store the representative vertices for all the clusters of the first simplification provides a convenient data-structure for merging adjacent clusters.

A series of models with different degrees of simplification is shown Fig. 4.

5. Rendering modes

Simplified models may be used in different ways, depending on the application and the type of user interaction.

5.1 Preview on simplified models

If the full precision model does not fit in the graphic workstation's memory, a simplified version may be loaded and used to specify interactively viewing angles or walkthrough trajectories for a camera. Full precision

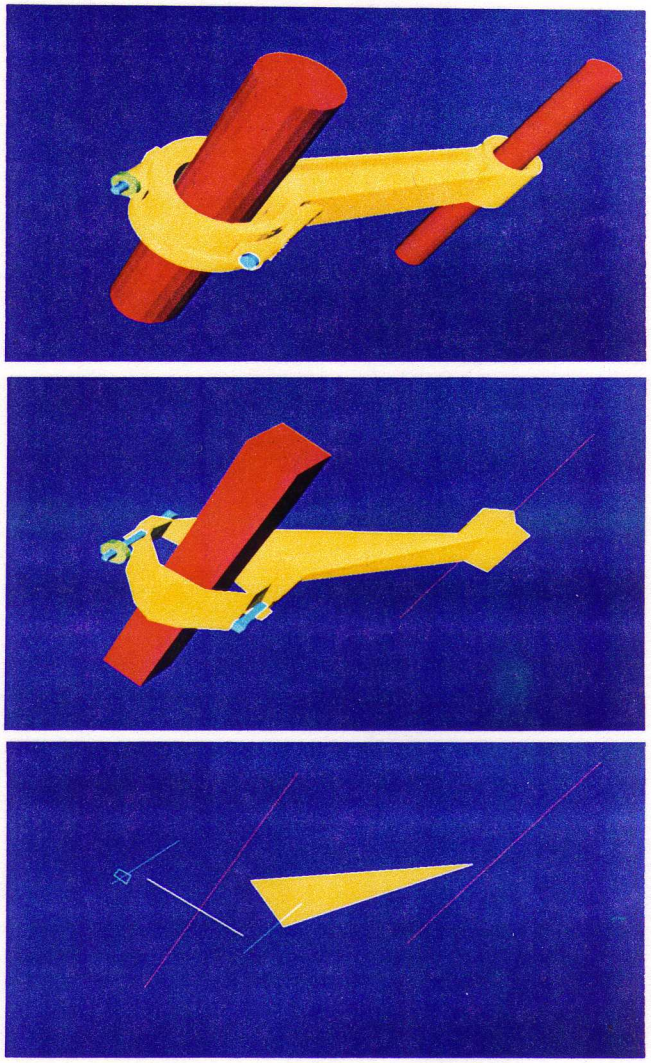

Fig. 4: The original solid (top) has 4804 triangular faces. The two approximating models below have respectively 218 faces plus 1 dangling edge and 1 face plus 9 dangling edges.

images or walkthrough animations may be computed in batch mode and viewed later.

5.2 Simplified models during motion

If the full precision model fits on the workstation and can be displayed at acceptable but non-interactive rates, a crude simplified model is constructed and stored. (It typically only increases the storage requirements by a few percents.) The user may then toggle between the full resolution original data and a crude approximation of all the objects, used mainly for realtime feedback during interactive navigation.

5.3 Simplified background

In the above scheme, selected details of the scene may be rendered with full precision, while the other objects, considered as background information are rendered using simplified models (see Fig. 5).

5.4 Dynamic selection

Simplification levels may be selected adaptively depending on the viewpoint. Models that are further away from the viewer are displayed with less details. The distance to the viewer may be estimated using precomputed

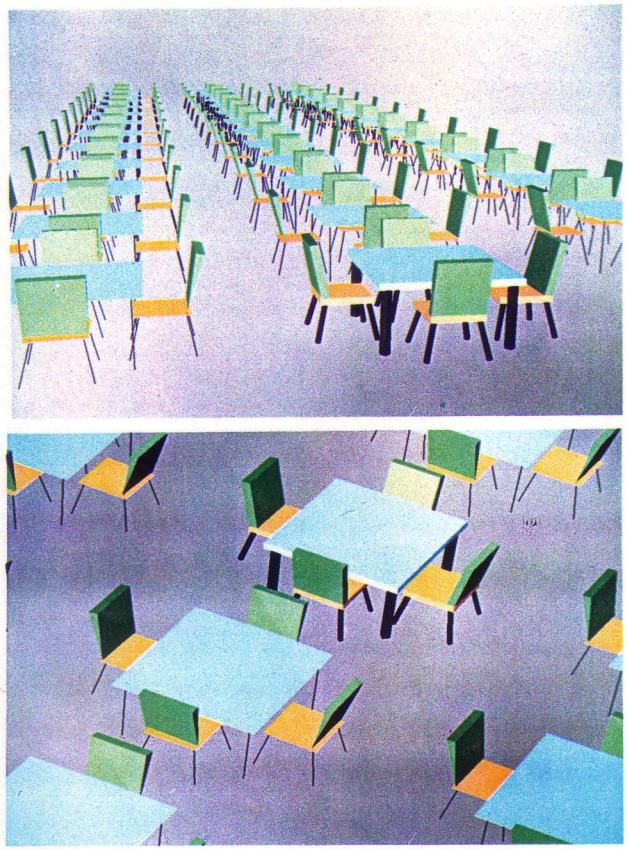

Fig. 5: Only one table and its set of chairs are rendered with full precision, the others sets are rendered using simplified models (top), which reduces the rendering time by 80%. A detailed view comparing the approximated and the original models is shown (below).

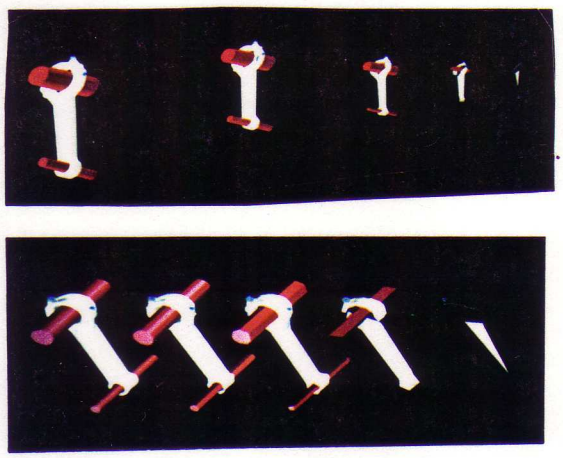

Fig. 6: Five instances of the same assembly (top) are shown using different levels of approximation that depend on their distance to the viewer. For comparison, the five approximations are shown with the same scale (below).

spherical or other simple bounds for each object. Fig. 6 shows the same parts as in Fig. 4, but with different degrees of approximation selected automatically according to the distance from the viewer.

5.5 Continuous evolution

When the more simplified models are derived from less simplified ones, one can produce parametric models, which simulate the metamorphosis between two consecutive simplification levels. We use the technique described in [3], where each vertex of the polyhedron is replaced by a linear parameterized trajectory. As the object moves closer to the viewer, the common parameter for computing all the vertices is smoothly adjusted. The result simulates the migration of all the vertices towards the representative vertex of their clusters. As the model, moving towards the viewer, traverses a threshold between two consecutive simplification levels, the system automatically switches between consecutive interpolating models at their common limit shape.

6. Conclusion

State of the art hardware graphic performance is insufficient for the realtime visualization of complex 3D objects. We have presented a new technique, which automatically computes one or several simplified graphic representations of each object that may be used selectively in lieu of the original model to accelerate the display process while preserving the overall perceptual information content of the scene. The described method clusters vertices of the model and produces an approximate model where original faces are approximated with fewer faces defined in terms of selected vertices. Several simplified representations with different simplification factors may be stored in addition to the original model. Actual viewing conditions are used to establish automatically for each object which representation should be used for graphics.

References

[1] Edelsbrunner, H., Tan, S.T., and Waupotitsch, A., "A polynomial time algorithm for the mini-max angle triangulation," *6th ACM Symp. on Computational Geometry*, pp. 44-52, 1990.

[2] Garlick, B.J., Baum, D.R., and Winget, J.M., "Interactive viewing of large geometric databases using multiprocessor graphics workstations," *SIGGRAPH Course Notes*, vol. 28, pp. 239-245, 1990.

[3] Kaul, A. and Rossignac, J., "Solid-Interpolating Deformations: Constructions and Animation of PIPs," *Proceedings of EUROGRAPHICS 91*, pp. 493-505, Vienna, September 91.

[4] Kim, Y.S., *Convex decomposition and solid geometric modeling*, PhD thesis, Dept. of Mechanical Engineering, Stanford University, 1990.

[5] McMaster, R.B., "Automated Line Generation," *Cartographica*, vol. 24, no. 2, pp. 74-111, 1987.

[6] Rockwood, A., Heaton, K., and Davis, T., "Real-time Rendering of Trimmed Surfaces," *SIGGRAPH Proc. 1989, Computer Graphics*, vol. 23, no. 3, pp. 107-116, 89.

[7] Silicon Graphics, Inc., Graphic Library - Reference Manual, Iris 4D VGX, 1990.

[8] Teller, S.J. and Séquin, C.H., "Visibility Preprocessing for interactive walkthroughs," *Computer Graphics (Proc. SIGGRAPH)*, vol. 25, no. 4, pp. 61-69, July 1991.