Interactive Display of Large Solid Models for Walkthroughs

S. Kumar  S. Krishnan  D. Manocha
Department of Computer Science
University of N. Carolina
NC 27599-3175
{kumar,krishnas,manocha}@cs.unc.edu
http://www.cs.unc.edu/~{kumar,krishnas,manocha}

1 Introduction

Over the last few decades, computer graphics and CAD/CAM technology has been providing a new paradigm in the design of complex engineering systems. The goal has been to develop a comprehensive virtual design environment that allows the users to design and visualize all stages of the product. Not only will it free the designer from the two dimensions of pencil and paper, but also eliminate construction of physical mock-ups and result in better designs. Currently, many research and commercial organizations like Lockheed Martin, Boeing, Newport News Shipbuilding, Electric Boat and automotive industries are developing systems for simulation based design (SBD). An integral component of such an SBD system is the development of a virtual environment, where the designers can experience a sense of reality.

Our efforts at the University of North Carolina are directed towards developing the necessary technology for interactive walkthroughs of large mechanical and architectural models. This involves model construction, display, interaction as well as development and interface with high-performance graphics systems and immersive technologies. Our ultimate goal is to create an environment where the user will explore the complete model of a large design, e.g., a submarine, and verify if all constraints are satisfied in the same way as he would in a real one – by walking around it.

The CAD models of complex geometries like submarines, shipmodels etc. are composed of hundred of millions of primitives. These primitives typically correspond solid models represented using nonlinear mathematical functions, like quadrics and higher order splines, and their boolean combinations. Such models are also known as CSG (constructive solid geometry) models. One such example has been shown in figure 2. These models are increasingly

being used in the design of complex geometries like submarines, ship-models, machine parts, airplanes etc. Many techniques are known in the literature for rendering the solid models. These include direct rendering, ray-tracing and polygonization. The systems for direct rendering either restrict the number of boolean operations or the degrees of the primitives or are not able to render complex models. Techniques based on ray-tracing are too slow for interactive display of complex geometries. The fastest techniques for based on polygonization and rendering the resulting polygons over the graphics pipeline. However, the polygonal models corresponding to complex geometries are composed of billions of polygons. Current high-end graphics systems like SGI’s Reality Engine 2 and UNC’s Pixel-planes 5 can render up to two million triangles per second. Some of the future systems like Pixel#0Do w, being developed at UNC Chapel Hill, are expected to provide one order of magnitude improvement in polygon rendering capabilities. Given the geometric complexity of large and complex 3D models for submarines and ships, we need to develop appropriate model representations, algorithms and systems for displaying such models at interactive frame rates (i.e. more than 15 frames a second).

In this article, we present efficient representations, algorithms and systems for interactive display of solid models defined using boolean combinations. One of our themes has been the integration of model generation, model representation and model display. It includes efficient and accurate algorithms for computing the boundary of the solid models in terms of trimmed spline models. The resulting models are polygonized on-line as a function of the viewing parameters and rendered over the graphics pipeline. We give a brief overview of the algorithms and systems and demonstrate their performance on a model of a submarine torpedo storage and handling room (as shown in figure 1).

2 Overview

Given a CSG model, we compute its boundary representation (B-rep) as an off-line process and generate a more explicit representation of the model. The input consists of a CSG model consisting of primitives with curved surfaces (represented as collection of spline patches), our system performs accurate boolean operations and computes the B-rep as a collection of trimmed Bézier surfaces. It represents the trimming curves as piecewise algebraic space curves along with bounding volumes for intermediate computations. It also computes accurate parametric spline approximations of these curves for efficient rendering. In addition, it maintains topological information for each solid as an adjacency graph. It makes use of algorithms for surface intersection, polygon triangulation, domain partitioning and ray-shooting.
to classify regions according to the boolean operation (component classification) to compute the B-reps (see the block diagram in figure 3).

Our display system performs visibility culling and dynamically tessellates the trimmed Bézier surfaces into triangles as a function of viewing parameters. (See figure 4.) In addition, it makes use of frame to frame coherence computing incremental triangulation. On parallel graphics systems, the display system makes use of multiple processors for tessellation, minimizes communication between processors and load balances the work between polygon generation and polygon rendering. To prevent cracks in images due to non-matching tessellation at trimmed boundaries, the system uses dual representations of the trimming curves, and the topological graph of the solids to tessellate the boundary between Bézier surfaces in the same number of steps.

The ability to compute varying resolutions of the B-reps in terms of trimmed curves and surfaces is fundamental to the performance of the overall system. As compared to earlier systems for rendering such models, it has the following advantages:

- **Fidelity:** A high degree of fidelity of display is essential for any meaningful visualization and design validation. Our B-reps for CSG models are more accurate than those generated by modelers using polygonal representation for the curved primitives and the final solids. Besides rendering, it is also useful for other applications like collision detection.

- **Rendering:** Our algorithms based on visibility culling and dynamic tessellation generate fewer polygons for the spline models. Furthermore, we can easily generate on-line any level-of-detail with correct topology using incremental computations. This is in contrast with the difficulty of computing multiresolution models for large polygonal datasets of arbitrary topology with visible artifacts introduced due to few and discrete levels of detail. Our method results in better images and faster display.

- **Memory:** The memory requirements for our B-reps are about an order of magnitude lower than polygonal models and their multiresolution representations. On the other hand our display system needs more processing power for on-line triangulation of the multiresolution representations.

### 3 Performance

We have implemented our algorithms and applied it to a number of solids comprising the model of a submarine storage and handling system, made available to us by Electric Boat, a division of General Dynamics. The model consists of about 2,000 CSG trees. Many of the primitives are composed of polyhedra and conicoids like spheres, cylinders. Additional primitives include prisms and surfaces of revolution of degrees six and more. A few of the
primitives are composed of Bézier surfaces of degree as high as 12. Most of the CSG trees have heights ranging between six and twelve and some of them are as high as 40. The B-reps of many of the solids consist of more than 40—45 trimmed Bézier surfaces and some of them have up to 200 surfaces. The 71 CSG trees of the torpedo model (figure 5) result in 3,346 surfaces. The pivot model shown in figure 6 consists of 168 CSG trees and results in 4,435 trimmed Bézier surfaces.

3.1 Model Generation and Representation

The running time of the system depends on the number of boolean operations, number of intersecting pairs of surfaces and the number of connected components generated. In most cases, it spends about half the time in computing intersections between pairs of surfaces and the other half in computing the components of new solids. The current implementation is not interactive and it takes 1 — 2 minutes for B-rep computation of an average CSG tree in the submarine torpedo storage and handling system model (defined using 12 — 14 CSG operations).

A major issue in the application of our system to different models is numerical accuracy of computations and its impact on the robustness of the entire system. The problem of building robust solid modeling systems based on floating-point computation is fairly open and no good solutions are known. In our case, the algorithm uses tolerances at different parts of the overall algorithm. Depending on the values of the tolerances, the robustness of the algorithm can vary considerably. As an input, the user specifies a set of four tolerance values and they are used as part of the surface intersection algorithm, for ray-shooting, merging intersection curves and detecting planar overlaps. The surface intersection algorithm normalizes the input surface parameterizations and ensures that the output of the intersection algorithm has certain digits of accuracy. The intersection algorithm is based on iterative numerical algorithms and we set the termination criterion accordingly. Similar criteria are used in computing the intersection of trimming curves represented as piecewise algebraic curves. At the end of every CSG operation, the system makes sure that the topology of the resulting solid is consistent (i.e., the solid boundary partitions \( R^3 \) into two or more regions).

3.2 Interactive Display

The display system has been implemented on an SGI Onyx and Pixel-Planes 5 graphics system. On Pixel-Planes 5 it uses multiple graphics processors (GP’s) for visibility computations, evaluating Bézier functions and triangulating polygons. The trimmed Bézier surfaces are evenly distributed over different GP’s and the system associates each surface with the parent solid for visibility computations. Each GP has about 2.5 Megabytes of memory for storing the surface representations and caching the triangle vertices and their normals. The system uses a dynamic memory allocation scheme for caching triangles.

The rendering algorithm produces topologically correct triangulations. As we zoom in or out on a model, it produces varying levels of detail incrementally and no visual artifacts can be noticed. The trimming algorithm also works for trimming curves represented as piecewise linear or spline curves. On an SGI Onyx, our implementation can render more
than a thousand surfaces at 15 frames a second. Compared to IRIS GL library's microcoded NURBS implementation, our implementation is faster by a factor of 10 to 12 on models consisting of about 2,000 surfaces on an SGI Onyx. The Pixel-Planes 5 implementation can render over 30,000 trimmed Bézier surfaces at more than 10 frames a second.

4 Conclusion

We have demonstrated a system for generating an accurate B-rep for CSG models composed of curved primitives and rendering the resulting primitives on current graphics systems. Its application to parts of a submarine storage and handling system model helped us improve the overall frame rate by three to four times, as compared to rendering the polygonal B-rep. The overall system is currently being used for walkthrough of large CAD models like submarines, Bradley fighting vehicle and ship-models. The on-line triangulation allows us to spend our rendering resources in the parts of the model significant for the current image. The memory requirements are also reduced.

5 Acknowledgements

We are grateful to Fred Brooks, Anselmo Lastra, Atul Narkhede and members of UNC Walkthrough and Modeling groups for productive discussions. The CSG model of the submarine storage and handling system was provided to us by Greg Angelini, Jim Boudreaux and Ken Fast at Electric Boat.

6 For Further Reading

1. Interactive display of large scale trimmed NURBS models. http://www.cs.unc.edu/~kumar/render.html, 1995, and
