IMPROVING THE EFFECTIVENESS OF INTEGRAL PROPERTY CALCULATION IN A CSG SOLID MODELING SYSTEM BY EXPLOITING PREDICTABILITY

Alan L. Clark
Ford Motor Company, Research Staff
Computer-Aided Engineering Department
Dearborn, MI

Integral property calculation is an important application for solid modeling systems. Algorithms for computing integral properties for various solid representation schemes are fairly well known. It is important to designers and users of solid modeling systems to understand the behavior of such algorithms. Specifically the trade-off between execution time and accuracy is critical to effective use of integral property calculation. This paper investigates the average behavior of two algorithms for Constructive Solid Geometry (CSG) representations. Experimental results from the PADL-2 solid modeling system show that coarse decompositions can be used to predict execution time and error estimates for finer decompositions. Exploiting this predictability allows effective use of the algorithms in a solid modeling system.

Integral properties of a solid, sometimes called "mass properties," are defined by a volumetric integral of the form:

$$I = \int_S f(x,y,z)\ dV$$

where the function f is a polynomial, dV is the volume differential, and S is a solid that may be geometrically complex. Thus, for $f(x,y,z) = 1$, the integral represents the volume of the solid. Other functions are used to obtain the centroid, moments of inertia, products of inertia, etc.

There are few algorithms for computing integral properties directly from the representation schemes used in current solid modeling systems. Analytical computation methods for complex solids are extremely difficult. With the exception of polyhedral representations, approximate representation conversion provides the most effective means for computation of integral properties. For solid modeling systems using the CSG representation scheme, the two most effective representation conversion algorithms are column decomposition and block decomposition.

Column decomposition produces a collection of simple elements whose integral properties are trivial to compute. The decomposition is performed using ray tracing techniques. A ray is cast perpendicular to each square from a 2-D grid. Line/solid classification is performed for each ray, producing a columnar decomposition whose elements have a square cross section. Similarly, block decomposition produces a collection of cubical elements whose integral properties are also trivial to compute. The decomposition is performed by recursively subdividing a bounding cube into octants and classifying each octant cube against the CSG solid. The subdivision continues for each octant cube which is neither inside nor outside of the CSG solid, producing a hierarchical decomposition known as an octree. Point/solid classification is performed at the final level of subdivision using a sample point from the octant cube. Using Monte Carlo theory, an error estimate (variance) for the result can be introduced into the algorithms by randomly selecting the target rays and sample points.

The two most important measures of the algorithms' behavior are execution time (a measure of efficiency) and variance estimate (a measure of accuracy). A theoretical analysis of the algorithms only yields results for their worst-case behavior. The worst-case formulas for execution time and variance estimate are based on the level of subdivision used (2-D or 3-D grid size) and the number of primitives in the solid's CSG tree (a rough measure of its complexity). For column decomposition the worst-case execution time is a quadratic function of the number of primitives in the solid's CSG tree and a quadratic function of the number of grid squares along an edge of the solid's bounding cube. For block decomposition the worst-case execution time is a linear function of the number of primitives and a cubic function of the number of minimal size grid cubes along an edge of the bounding cube.

An implementation of the two algorithms can be found in the PADL-2 solid modeling system. Solid models for several automotive mechanical parts were used to generate experimental data for the above two measures of behavior (execution time and variance estimate). An analysis of the data yields a set of formulas for the average behavior of the algorithms that are quite different from the worst-case analysis. For each formula, the variables are the level of subdivision (2-D or 3-D grid size) and a proportionality constant K. This constant K varies from solid to solid and is roughly proportional to the number of primitives in the solid's CSG tree. For column decomposition, the worst-case analysis predicted a quadratic function of the number of primitives, not a linear one. For block decomposition, the worst-case analysis predicted a cubic function of the number of minimal size grid cubes along an edge of the bounding cube, whereas a quadratic function was observed instead.

The most useful fact is that the proportionality constant K is independent of the level of subdivision. A user would like to address questions such as: "How long will it take to compute the integral

properties of a solid to 1% accuracy?" or alternatively "What level of accuracy can be obtained from one hour's execution time?". These questions can be answered by actually computing the integral properties for a subject solid using a low level of subdivision. The values for execution time and variance estimate can be used as "seed" values to compute the K constants for the solid. The formulas will then predict the algorithms' behavior at higher levels of subdivision.

A prediction command has been implemented in the PADL-2 solid modeling system. A comparison of the predicted execution times and variance estimates with actual values shows this prediction cabability accurate enough to be quite useful.