A Note on Some Applications of Interval Arithmetic in Hierarchical Solid Modeling

Eva Dyllong[1]

University of Duisburg-Essen, Department of Computer Science and Applied Cognitive Science
D-47057 Duisburg, Lotharstrasse 65, Germany
`dyllong@inf.uni-due.de`

Abstract. Techniques of reliable computing, like interval arithmetic, can be used to guarantee reliable solutions even in the presence of numerical round-off errors. The need to trace bounds for the error function separately can be eliminated using these techniques.
In this extended abstract, we focus on presenting how the techniques and algorithms of reliable computing can be applied to the construction and further processing of hierarchical solid representations using the octree model as an example.

Keywords. Reliable solid modeling; hierarchical data structure

1 Introduction

Accurate and reliable computations are important requests of many applications. Techniques of reliable computing, like interval arithmetic, can solve problems such that the results are guaranteed to be correct, even when the computation has been done using floating point operations with finite precision. In the field of solid modeling, we can be sure using these techniques that all points of a modeled object are included or that the computed path among obstacles is collision-free. Nevertheless, up till now, interval methods have been applied only to a limited class of problems in solid modeling.

An interval arithmetic approach to performing collision detection between constructive solid geometry (CSG) objects using binary subdivisions of space along each axis was presented by Duff [1]. Snyder et al. [2] used an interval Newton method to solve the system of equations that specifies the tangency constraints at the touching points to detect collisions between curved surfaces. The utilization of affine arithmetic and other improved techniques of reliable computing in the field of modeling can be found in [3,4]. For moving multibody models a continuous collision detection algorithm that relies on Taylor models to compute dynamic bounding volumes (AABB hierarchies) was proposed in [5].

In this paper, we present a novel generalization of the octree model created from a CSG object that uses interval arithmetic. The octree model was chosen because it is particularly suitable for accurate and reliable computation. On the

Dagstuhl Seminar Proceedings 08021
Numerical Validation in Current Hardware Architectures
http://drops.dagstuhl.de/opus/volltexte/2008/1440

one hand, an octree node belonging to an arbitrary hierarchy level geometrically defines an axis-aligned box. All boxes have as vertices machine numbers that are multiples of powers of two. In the case of an axis-aligned octree, accurate algorithms for proximity queries are feasible [4]. On the other hand, interval-based evaluation of the structure allows us to extend the tests for classifying points in space as inside the object, on the boundary or outside the object to whole sections of the space at once.

2 Reliable Hierarchical Solid Modeling

An octree is a common hierarchical data structure with which to represent 3D geometrical objects in solid modeling systems or to reconstruct a real scene. The solid representation is based on recursive cell decompositions of the space [6]. This object representation yields relatively high visual quality and features stable numerical computations. A further advantage is the adaptive control of the approximation level that can be used in such applications as robotic simulations to speed up geometrical computation.

The use of interval arithmetic allows us to provide validated octree representations of implicit objects [1]. In our case, the implicit functions describe CSG primitives (spheres, cubes, cylinders) and are used as test functions for classifying points as inside the object, on the boundary or outside the object [7]. We specify the volume expansion of a axis-aligned cuboid as a three-dimensional interval vector. If we extend an implicit function on real numbers to an implicit function on intervals (using an interval extension of the characteristic function) and solve it for an interval X, we get an interval Y. If the interval X intersects the surface of the implicit object, the interval Y includes zero. On the other hand, if Y does not include zero, X does not intersect the surface of the object, and we can reliably decide whether X is contained within the implicit object or not. Thus, we are able to construct an octree which approximates the implicit object and provides a guaranteed superset of the given object if we consider *gray* leaf nodes as solid.

Unfortunately, the classical octree data structure may require a large amount of memory if it uses a set of very small cubic nodes to approximate a solid. In recent years several generalizations (polytrees, integrated polytrees, extended octrees) have been developed to reduce the depth of required subdivisions by including additional information about the relevant parts of an object. Brunet et al. [8] suggested an efficient way to handle a boundary representation hierarchically. In the context of ray tracing, Wyvill et al. [9] proposed the use of octree representations to deal with CSG objects. Our interval-based octree model maintains not only additional information about relevant parts of the CSG object within a node but also applies two types of gray nodes: terminal and non-terminal gray nodes. *Terminal gray* nodes are nodes that contain only parts of the surface that result from exactly one primitive of the CSG object and thus are not further divided. All other *gray* nodes are *non-terminal gray* and have to be subdivided if a higher octree level is required. The subdivision process can be restarted for the non-terminal gray nodes in any part of postprocessing if a more detailed hierarchical approximation model of the CSG object is required. Following Duff [1], we simplify the CSG object for each octree node to a CSG object that is equivalent to the original object within that octree node. If the octree has sufficient subdivision depth, the CSG tree can be reduced to only one primitive for most octree nodes.

3 Applications and Illustrative Examples

The interval-based data structure for solid modeling has been implemented in an object-oriented style in the C++ programming language. We used the C++-library for scientific computing, C-XSC [10] for interval representations and calculations and the OpenGL-library for visualizations [11]. In order to check out the performance of the structure in the field of robotic applications, suitable interval-based algorithms for reliable proximity queries have been developed (see Figure 1). Moreover, to speed up collision detection between two octree-encoded CSG objects, hierarchical polyhedral enclosures have been constructed on the basis of high accurate convex hull algorithms. In two-dimensional space, to achieve a narrower range of space overestimation of the hierarchical polyhedral enclosures, high accurate algorithms for convex decomposition of the structure have been developed. The adaptation of the decomposition algorithm to higher dimension is a current work in progress. Within the scope of single-query path planning, several approaches for efficiently searching high-dimensional spaces based on the presented data structure are being investigated. Furthermore, we intend to improve the quality of enclosure and structure handling considering other variants of interval arithmetic, like Taylor models or affine arithmetic.

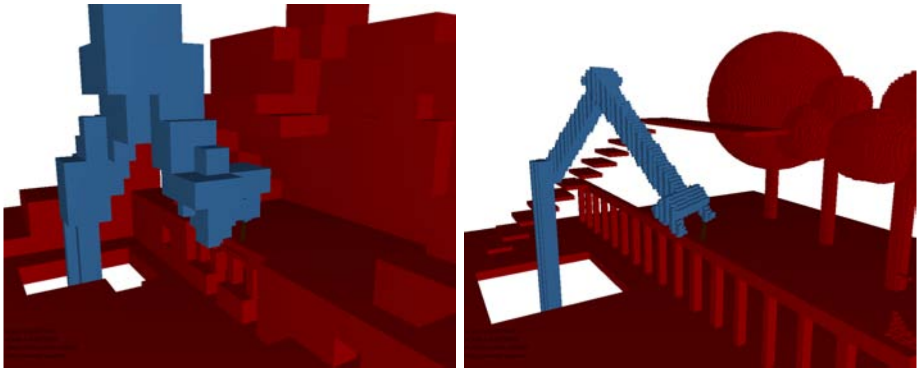

Fig. 1. An octree-encoded robot and its environment.

References

1. Duff, T.: Interval arithmetic and recursive subdivision for implicit functions and constructive solid geometry. In: SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM Press (1992) 131–138
2. Snyder, J.M.: Generative Modeling for Computer Graphics and CAD: Symbolic Shape Design Using Interval Analysis. Academic Press, San Diego (1992)
3. Ratschek, H., Rokne, J.: Geometric Computations with Interval and New Robust Methods. Horwood Publishing, Chichester (2003)
4. Bühler, K., Dyllong, E., Luther, W.: Reliable Distance and Intersection Computation Using Finite Precision Geometry. Numerical Software with Result Verification. LNCS 2991 **Volume 2991** (2004) 160–190
5. Zhang, X., Redon, S., Lee, M., J., K.Y.: Continuous Collision Detection for Articulated Models Using Taylor Models and Temporal Culling. ACM Transactions on Graphics **26** (2007)
6. Samet, H.: The quadtree and related hierarchical data structures. ACM Computing Surveys **16** (1984) 187–260
7. Samet, H., Tamminen, M.: Bintrees, CSG trees, and time. SIGGRAPH Comput. Graph. **19** (1985) 121–130
8. Brunet, P., Navazo, I.: Solid Representation and Operation Using Extended Octrees. ACM Transactions on Graphics **9** (1990) 170–197
9. Wyvill, G., Kunii, T., Shirai, Y.: Space division for ray tracing in CSG. IEEE Comp. Graphics Applic. **6** (1986) 28–34
10. Hofschuster, W., Krämer, W.: C-XSC 2.0: A C++ Library for Extended Scientific Computing. Numerical Software with Result Verification. LNCS 2991 **Volume 2991** (2004) 15–35
11. Dyllong, E., Grimm, C.: Verified Adaptive Octree Representations of Constructive Solid Geometry Objects. In: SimVis. (2007) 223–236