Computer-Aided Design 40 (2008) 1113–1120

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Improved Binary Space Partition merging[☆]

Mikola Lysenko [a,*], Roshan D'Souza [b], Ching-Kuan Shene [a]

[a] Michigan Technological University, Department of Computer Science, Houghton, MI 49931, USA

[b] Michigan Technological University, Department of Mechanical Engineering, Houghton, MI 49931, USA

ARTICLE INFO

Article history:
Received 23 April 2008
Accepted 4 November 2008

Keywords:
Constructive solid geometry
Binary space partition
Linear programming
Tree merging

ABSTRACT

This paper presents a new method for evaluating boolean set operations between Binary Space Partition (BSP) trees. Our algorithm has many desirable features, including both numerical robustness and $O(n)$ output sensitive time complexity, while simultaneously admitting a straightforward implementation. To achieve these properties, we present two key algorithmic improvements. The first is a method for eliminating null regions within a BSP tree using linear programming. This replaces previous techniques based on polygon cutting and tree splitting. The second is an improved method for compressing BSP trees based on a similar approach within binary decision diagrams. The performance of the new method is analyzed both theoretically and experimentally. Given the importance of boolean set operations, our algorithms can be directly applied to many problems in graphics, CAD and computational geometry.

1. Introduction

Boolean operations are important in a wide variety of computer aided geometric design problems, including range searching, collision detection, motion planning and visibility. Yet, despite their ubiquity, simple algorithms for evaluating boolean operators are largely unknown. Current approaches are dogged by poor performance, numerical instability and labyrinthine complexity, which forces programmers to resort to expensive commercial packages. In this paper, we give a simple method for computing boolean set operations using Binary Space Partition (BSP) trees. A key improvement within this algorithm is the use of a linear programming feasibility test which removes the need to perform difficult tree partitioning and polygon cutting used in current BSP tree merging methods [1]. Compared to existing BSP tree algorithms, our new approach is substantially simpler, more efficient and robust. We also derive a method for reducing the amount of memory consumed by a labeled leaf BSP tree, using a collapsing scheme derived from binary decision diagrams [2].

2. Previous work

Rossignac gives a good overview of current solid modeling techniques and applications [3]. For this paper, we briefly summarize two general approaches to evaluating boolean operations: Boundary Representations (BREPs) and Constructive Solid Geometry (CSG). BREP methods directly operate on meshes and easily interface with standard file formats and display systems. Laidlaw et al. [4] gave the first BREP algorithm for boolean operations, which Hubbard [5] later improved for triangulated meshes. Recently, Smith and Dodgson gave a BREP intersection algorithm with provable conditions for topological robustness [6]. For NURBs surfaces, Krishnan and Manocha discovered an optimal output sensitive algorithm [7]. Though BREP methods are the most popular category of boolean algorithms, they require complex case-by-case analysis and are difficult to implement. Achieving reasonable robustness and performance requires the use of multiple supporting data structures. Our algorithm eliminates these dependencies by exploiting the implicit spatial indexing within a BSP tree. Like Krishnan and Manocha, we use linear programming to narrow intersection searches, however we go one step further, in that the same pruning operation is used throughout the entire BSP tree, thereby reducing the number of special cases.

CSG methods are dual to BREPs, in that they represent objects using their interior instead of their boundary [8,9]. Under a sufficiently broad interpretation, this includes voxels, implicit surfaces and spatial indexing trees. CSG defers evaluation of the boolean expression until the last possible moment, such as the time of ray intersection. A similar effect can be achieved with polygon rasterization and z-buffer clipping, though performance scales badly with expression size [10,11]. Recently, Rossignac et al. used modern graphics processing units to dramatically improve the performance of this approach through the use of B-Lists [12]. Common in many BREP and CSG schemes is the use of spatial indexing data structures for accelerating various geometric

☆ Expanded version of a poster presented at SIGGRAPH 2007 (San Diego, CA. August 2007) (Lysenko M. Realtime constructive solid geometry. In: SIGGRAPH 07: Posters. New York (NY, USA): ACM; 2007. p. 132. [39]).

* Corresponding author. Tel.: +1 9895132146.
E-mail addresses: mikolalysenko@gmail.com (M. Lysenko), rmdsouza@mtu.edu (R. D'Souza), shene@mtu.edu (C.-K. Shene).

0010-4485/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2008.11.002