November 2008

MADALGO seminar by Peter Hachenberger, Aarhus University

Using the right BVH in each situation

Bounded volume hierarchies (BVHs) are efficient and versatile search data structures for geometric data, which are similar to binary space partitions (BSPs). Like BSPs, they store the geometric data in the leafs of a search tree. While each non-leaf node of a BSP stores a half-plane that decides what data is stored in which sub-tree, the non-leaf nodes of a BVH stores a bounding volume of the data of each of its sub-trees.

Different BVHs differ mostly in two ways, in the shape of the bounding volumes, and in the way they are constructed. The first criterion usually determines the name of a BVH. The simplest BVH is the box tree, which stores axis-aligned boxes as bounding volumes. Its strength is its simplicity and the efficiency of intersection tests with a box. Its weakness is input data aligned to a diagonal, which may cause linear instead of the usual logarithmic query running time.

A *c*-DOP tree stores convex polygonal bounding volumes whose sides are parallel to a set of *c* pre-determined directions. *c*-DOP trees have better worst-case query running times than box trees, but lack the simplicity of box trees and are therefore often slower.

We introduce a new BVH, the *c*-rotating-box tree (*c*-rb tree), which is a compromise of box tree and *c*-DOP tree. Every bounding volume stored in a *c*-rb tree is a box, but they are not necessarily axis-aligned. The boxes are aligned to c pre-determined directions. The *c*-rb tree allows more efficient intersection operations than *c*-DOP trees and has the same worst-case behavior.

We prove that *c*-rb trees indeed have the same worst case complexity as *c*-DOP trees, and we compare several variants of the three data structures by experiments.

Joint work with Mark de Berg.