Article   Talk

Read   Edit   View history

Search

Sweep and prune

From Wikipedia, the free encyclopedia

In physical simulations, **sweep and prune** is a broad phase algorithm used during collision detection to limit the number of pairs of solids that need to be checked for collision, i.e. intersection. This is achieved by sorting the starts (lower bound) and ends (upper bound) of the bounding volume of each solid along a number of arbitrary axes. As the solids move, their starts and ends may overlap. When the bounding volumes of two solids overlap in all axes they are flagged to be tested by more precise and time consuming algorithms.

Sweep and prune exploits temporal coherence as it is likely that solids do not move significantly between two simulation steps. Because of that, at each step, the sorted lists of bounding volume starts and ends can be updated with relatively few computational operations. Sorting algorithms which are fast at sorting almost-sorted lists, such as insertion sort, are particularly good for this purpose.

According with the type of bounding volume used, it is necessary to update the bounding volume dimensions every time a solid is reoriented. To circumvent this, temporal coherence can be used to compute the changes in bounding volume geometry with fewer operations. Another approach is to use bounding spheres or other orientation independent bounding volumes.

Sweep and prune is also known as *sort and sweep*[1] being called that way at David Baraff Ph. D thesis in 1992.[2] Later works like the 1995 paper about I-COLLIDE by Cohen *et al.* [3] refer to the algorithm as *sweep and prune*.

See also   [edit]

- Collision detection
- Bounding volume
- Physics engine
- Game physics

References   [edit]

1. ^ Ericson, Christer (2005), *Real-time collision detection*, Morgan Kaufmann series in interactive 3D technology, Amsterdam: Elsevier, pp. 329–338, ISBN 978-1-55860-732-3
2. ^ Baraff, D. (1992), *Dynamic Simulation of Non-Penetrating Rigid Bodies, (Ph. D thesis)*, Computer Science Department, Cornell University, pp. 52–56
3. ^ Cohen, Jonathan D.; Lin, Ming C.; Manocha; Ponamgi, Madhav K. (April 9–12, 1995), *I–COLLIDE: an Interactive and Exact Collision Detection System for Large Scale Environments)* (PDF), Proceedings of the 1995 Symposium on Interactive 3D Graphics (Monterey, CA), pp. 189–196

External links   [edit]

- Bullet user manual
- Collision Detection in Interactive 3d Environments

This computing article is a stub. You can help Wikipedia by expanding it.

Categories: Computational physics | Computer physics engines | Computing stubs