Collision Detection Notes

Applications [1]:

Real-time collision detection is of critical importance in computer graphics, visualization, simulations of physical systems, robotics, solid modeling, manufacturing, and molecular modeling, as well as a number of other fields.

Naive Approach Problems [1]:

A simple-mined approach to CD involves comparing all pairs of primitive geometric elements. This method quickly becomes infeasible as the model complexity rises to realistic sizes. Consider, for example, an 3D environment consisting of 100,000 triangles together with a single flying object consisting of 20,000 triangles. Using this naïve approach to CD, for each location of the flying object, we would check each of the triangles in the flying object against all of the triangles in the environment. In total, therefore, we would need to perform two billion of these triangle-triangle checks for each end every position of the flying object to determine if it comes into contact with the (static) environment. Using our current implementation of the triangle-triangle check , testing each location of the flying object would take almost two hours to perform; a far cry from the 1000 intersection checks per second required by applications utilizing haptic force-feedback. Thus, many approaches have recently been proposed to address the issue of efficiency; we discuss these below.

Previous Work [1]:

Many of the approaches have used hierarchies of bounding volumes or spatial decompositions to address the problem. The idea behind these approaches is to approximate the objects (with bounding volumes) or to decompose the space they occupy (using decompositions), to reduce the number of pairs of objects or primitives that need to be checked for contact.

Octrees, k-d trees, BSP-trees, brep-indices, tetrahedral meshes, and (regular) girds are all examples of spatial decomposition techniques. By dividing the space occupied by the objects, one needs to check for contact between only those pairs of objects (or parts of objects) that are in the same or nearby cells of the decomposition. Using such decompositions in a hierarchical manner (as in octrees, BSP-trees, etc.) can further speed up the collision detection process.

Hierarchies of bounding volumes have also been a very popular technique for collision detection algorithms. In building hierarchies on objects, one can obtain increasingly more accurate approximations of the objects, until the exact geometry of the object is reached.

Vanecek has tried another approach to this reduction problem by using a modified backface culling technique [2].

There has been a collection of innovative work which utilizes Voronoi diagrams to keep track the closest features between pairs of object. One popular system, I-COLLIDE, uses spatial and temporal coherence in addiction to a “sweep-and-prune” technique to reduce the pairs of objects that need to be considered for collision. Others: Q-COLLIDE, V-Clip, V-COLLIDE, etc.

Theoretical Results [1]:

In particular, the distance (and thus intersection) between two convex polytopes can be determined in time, where n is the total number of vertices of the polytopes , by using the Dobkin-Kirkpatrick hierarchy, which takes O(n) time and space to construct.

## English Sentences:

### Introduction:

Our experiments indicate that the overhead added to the hardware rendering process by the collision detection and the determination of the contact regions is relatively small on the average. Further tests show a consistent performance improvement with respect to some popular object-space collision detection algorithms.

The algorithm exhibits the same degree of robustness as finding all the closest features and contact points as I-COLLIDE, while outperforming in some cases RAPID and Q-COLLIDE.

We make no assumptions about the motions of objects in the scene and there is no differentiation between the states of a collision event. For example, there is no need to differentiate between objects just before a collision, on contact, or penetrating each other.

Hence, there is no need to construct….

Furthermore, this approach benefits form a variable image-space resolution. This allows to dynamically trade off accuracy for computational time.

BIBLOGRAPHY

1. James Thomas Klosowski, Efficient Collision Detection for Interactive 3D Graphics and Virtual Environments, 博士学位论文, State University of New York, Stony Brook, 1998
2. George Vanecek, Back-Face Culling Applied to Collision Detection of Polyhedra, The Journal of Visualization and Computer Animation, Volume5, Issue 1, pages 55-63, January/March 1994