

# 3D Collision Detection Algorithm based on Composite Balanced Binary Bounding Box Tree

Jinping Liu<sup>1, a\*</sup> and Jianxun Chen<sup>1, 2, b</sup>

<sup>1</sup>College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China

<sup>2</sup>City College, Wuhan University of Science and Technology, Wuhan 430065, China

<sup>a</sup>853982366@qq.com, <sup>b</sup>7500080@qq.com

**Keywords:** Collision detection; Composite bounding box; Sphere bounding box; OBB bounding box; Shape classification

**Abstract.** The existing bounding box has the contradiction between compactness and complexity. On the one hand, the poor compactness of the bounding box often leads to the poor effect of removing the disjoint objects, which leads to the low efficiency of collision detection. On the another hand, the good compactness of the bounding box often leads to the high complexity of the collision detection algorithm, which leads to the high time complexity of collision detection. To solve this problem, this paper proposes a parallel collision detection algorithm based on the composite balanced binary bounding box tree. In the process of constructing the multilayered bounding box of the object in the algorithm, the selection of the bounding box is classified according to the near-sphere value of the different levels of the element set. The shape of an object is similar to that of a sphere, it will use the sphere bounding box, and the shape of an object is not similar to that of a sphere, it will use the OBB bounding box. This method solves the contradiction of the bounding box better. At the same time, in order to speed up the efficiency of collision detection, this paper uses the parallel algorithm. The experimental results show that the proposed algorithm is better than other classical algorithms, and is suitable for large scale application environment.

## Introduction

Collision detection technology is widely used in computer animation, robotics and computer aided design. It is mainly used to detect whether the objects collide and react accordingly. This technology is essential for the authenticity of the virtual scene.

This technology has been concerned by scholars at home and abroad for many years. At present, there are two kinds of classical algorithms: hierarchical bounding box [1] and spatial partitioning [2]. Hierarchical bounding box algorithm is larger and the geometric characteristics of a simple bounding box to replace the complex geometric objects, and by constructing a hierarchical bounding box tree approximation of real objects, collision detection by traversing the two bounding boxes to roughly determine the intersection condition of objects. The typical hierarchical tree consists of the sphere bounding hierarchy tree [3], the axial bounding box (AABB) hierarchical tree [4], the Oriented Bounding Box (OBB) hierarchical tree [5], the discrete direction bounding box (K-Dop) hierarchical tree [6], etc... The method of spatial segmentation is to divide the virtual space into different regions, and only detect the objects in the same region. The classical algorithm includes the uniform grid, octree, BSP tree.

## The Basic Idea of the Algorithm

**Classification of the Shape of an Object.** Geometric objects in virtual space is a triangle, the triangle composed of point sets can fully display the characteristics of objects, shape distribution analysis can determine the object point set[7].

Define 1. Near-sphere value: reflects the extent to which an object is close to the sphere. The formula is as follows:

$$P = \frac{\frac{1}{n} \sum_{i=1}^n \sqrt{(x_i - o_x)^2 + (y_i - o_y)^2 + (z_i - o_z)^2}}{r} \quad (0 < P \leq 1) \quad (1)$$

Among them,  $n$  represents the number of points,  $(x_i, y_i, z_i)$  represents the coordinates of the point  $i$ ,  $(o_x, o_y, o_z)$  represents the central coordinates of the sphere, and  $r$  represents the radius of the sphere.

The size of the near-sphere value reflects the extent to which the object is close to the sphere. The more close to the numerical sphere, smaller values deviate from sphere. In theory, when an object is a sphere, the  $P$  reaches a maximum value—one.

The process of classifying it according to the shape of the element set is as follows:

(1) According to the set of points of the element set, the minimum bounding sphere is calculated. The center coordinates of the bounding spheres is the average value of each point of  $X$ ,  $Y$  and  $Z$  coordinates, surrounded by the radius of the sphere center with three maximum distance coordinates determined by point.

(2) The near sphere value of each element set is calculated according to formula (1).

(3) The bounding box is selected according to the near sphere value. In order to make all the bounding box of the objects close well, you need to set a threshold value near the sphere here, if the last sphere object value exceeds the value of the object to the sphere extent, then the sphere bounding box (Sphere); if the object value is less than the value that the last sphere it deviated from the bigger sphere, then use OBB bounding box on the surrounding.

**Constructs Balanced Compound Binary Bounding Box Tree.** Whether the object bounding box structure is reasonable or not has a significant impact on the efficiency of the algorithm, and whether the object bounding box tree is balanced or not directly affects the effect of the parallel algorithm. In order to balance the bounding box tree [8], this paper uses the following methods, which is divided into three steps:

(1) The bounding box selection method described in last section is used to establish the bounding box of the whole object, which is the root node of the bounding box tree of the object.

(2) By using the plane perpendicular to the local coordinate axis, the bounding box is divided into two child bounding boxes (which can be called the left and right child bounding boxes) to form the two child nodes of the root node. The bounding box contains a set of polygons. For the left and right child surrounded by a polygonal box in roughly the same, to ensure that the bounding box tree balance, to minimize the segmentation and add two times across the cut surface of the polygon with two bounding box, we adopt a penalty function and to determine compliance with the requirements of the split face segmentation the function based on from three possible directions in. The specific penalty function  $f(p)$  is defined as follows:

$$f(p) = |n_l - n_r| + \lambda n_c \quad (2)$$

Among them,  $p$  is the dividing surface, respectively, the number of the left bounding box, the right bounding box and the spanning polygon. Is an influence factor. The ideal dividing surface is the one that has the least penalty function value on all axes:

$$\min \left\{ \begin{array}{l} \min\{f(p) \mid p_x \perp x\_axis, p_x \in [x_{\min}, x_{\max}]\} \\ \min\{f(p) \mid p_y \perp y\_axis, p_y \in [y_{\min}, y_{\max}]\} \\ \min\{f(p) \mid p_z \perp z\_axis, p_z \in [z_{\min}, z_{\max}]\} \end{array} \right\} \quad (3)$$

(3) The partition process of the bounding box is recursively performed on the two sub nodes obtained in (2) to obtain the final composite two fork bounding box tree.

**Termination Strategy of Balanced Compound Binary Bounding Box Tree.** Appropriate

termination conditions are important for recursive processes. To this end, we designed the following 3 exits for the recursive process to create the bounding box tree:

The recursion depth exceeds the maximum tree depth given in advance. Test examples of this, the maximum tree depth determined by the number of polygon objects more, the greater the value of maximum tree depth.

The number of polygons contained in the child nodes is less than the minimum of the number of polygons contained in the given leaf nodes. For the test example, the minimum number of polygons in the leaf nodes is 10.

The left and right child nodes have at least one number of polygons is close to its parent node contains a number of polygons. Because after this step and then divided down apparently lost meaning.

### Collision Detection Algorithm

**Intersection Test between Bounding Boxes.** In the real time detection of the compound binary bounding box algorithm, it is necessary to carry out the intersection test between the three kinds of bounding boxes, namely Sphere-Sphere, OBB-OBB and OBB-Sphere.

**Sphere-Sphere Test.** Whether the intersection judgment basis for the two center distance is greater than two and the radius of the sphere, in order to reduce the computational complexity, the square form:

$$|o_1 o_2|^2 > (r_1 + r_2)^2 \quad (4)$$

Among them,  $o_1$  and  $o_2$  are as the center of two bounding sphere,  $r_1$  and  $r_2$  is as two sphere radius,  $|o_1 o_2|$  is the center distance. If the formula (4) was established, the two sphere are disjoint, and otherwise the two sphere is judged to be the intersection.

**OBB-OBB Test.** 1) Prejudgment of the contact situation

In the detection of the two OBB bounding box before the intersection, the first of their contact to predict the situation, so as to avoid unnecessary intersection test process.

There are two OBB bounding boxes ( $B_i, B_j$ ) to be detected,  $o_i(x_i, y_i, z_i)$  and  $o_j(x_j, y_j, z_j)$  represent the center of the two bounding boxes, respectively.  $o_i.max$  and  $o_i.min$  are the maximum distance and the minimum distance from the boundary of the bounding box  $B_i$  to its center  $o_i$ , respectively,  $o_j.max$  and  $o_j.min$  are the maximum distance and the minimum distance from the boundary of the bounding box  $B_j$  to its center  $o_j$ ,  $D$  represents the distance from A to B. Prediction of the intersection of two OBB by the following inequality.

$$D = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (5)$$

$$D < o_i.min + o_j.min \quad (6)$$

$$D > o_i.max + o_j.max \quad (7)$$

If the inequality(6) is established, two OBB will return to the intersection, the collision detection results is collision; if the inequality (7) was established, the two OBB will not return to the intersection, the collision detection results is not collision; otherwise, OBB will continue to implement the basic intersection test.

**Basic Intersection Test.** Using the separation axis theory [8] to realize the intersection test of OBB. As shown in the figure, for a shaft L, if the length of the sum of the radius of the two

bounding boxes is smaller than the projection distance of the center line ( $|T \cdot L| > r_1 + r_2$ ), they are disjoint [9]. The fifteen potential separation axes must be tested to determine the intersection of OBB. These axes consist of three axes, the three axes and the 9 axes perpendicular to each axis. If there is no overlap on any axis of the above axis, the two OBB are separated, and if all axes are detected, the overlap of each axis, then the two OBB intersect in time.

**Sphere-OBB Test.** Similar to the intersection between the OBB-OBB test to determine whether the intersection of Sphere-OBB is also based on inequality  $|T \cdot L| > r_1 + r_2$ . Suppose there are sphere bounding O and OBB bounding B,  $r$  is the radius of the O,  $b_i$  is half of length, width and height ( $i=1,2,3$ ),  $B_i$  is a unit vector parallel to the axis of B ( $i=1,2,3$ ),  $T$  is the distance from the center of the two bounding box,  $L$  is a unit vector parallel to the OBB axis of the separation[9].  $r_1$  is the projection of  $r$  on  $L$ ,  $r_2$  is the sum of the projection of  $b_i$  on  $L$ . Then get the formula as follows:

$$r_1 = r \quad (8)$$

$$r_2 = \sum_{i=1}^3 |b_i B_i \cdot L| \quad (9)$$

$$|T \cdot L| > r + \sum_{i=1}^3 |b_i B_i \cdot L| \quad (10)$$

If the formula (10) is established, the two bounding boxes do not intersect, otherwise they will continue to calculate whether they overlap on the other two separate axes.

## Experiment and Analysis

In this paper, the algorithm achieve in the win7 system, with CPU 3.6GHz, quad core processor, 8G memory on the PC application development environment and Program code is implemented by the eclipse development tool with Java3D graph library[10] and Java math library colt-1.2.0[11]. The experimental environment set a simple dynamic scene, the scene of the random motion of 30 objects, the number of triangles of more than 8000, as shown below.

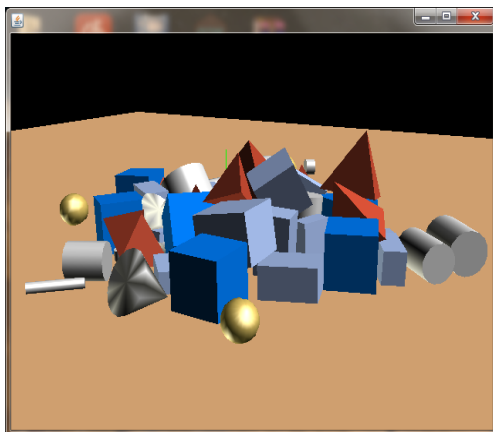


Figure 1

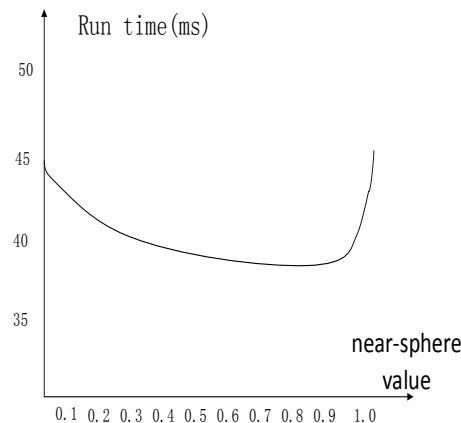


Figure 2

Figure 1. The program running effect diagram

Figure 2. The influence of the near-sphere value on the algorithm

Experiment 1: the influence of the near-sphere value on the algorithm.

By gradually changing the size of the sphere near the threshold value of statistical time average collision detection algorithm in this paper runs the 1000 step operation (ms)

It can be seen from the Fig. 1, The running time of the algorithm decreases gradually when the near-sphere value increases from 0 to 0.85, and up to 0.85, the running time reached minimum running time, so the paper use 0.85 as the threshold value of near-sphere value. At the same time, it can be seen that when the threshold value increases from 0.85 to 1, there is a rapid increase in the running time of the algorithm, and the threshold value reached 1 when the running time reaches the maximum. It can be found that the efficiency of OBB algorithm is higher than that of Sphere algorithm for the object of this paper.

Experiment 2: using the bounding sphere, OBB and the composite bounding box used in this paper to detect the collision between objects. In this case, the near-sphere value is set to be 0.85 in the experiment. The program record the average collision detection time (ms) and frame rate (frame/s) when it run 1000 steps. Time complexity analysis is order to compare the differences among the basic collision detection, serial collision detection and parallel collision detection.

Table 1 Test results of collision detection algorithm

Collision detection algorithm	Time complexity	Average time	FPS
Collision detection algorithm based on bounding sphere	$O(n^2)$	47.356	31.14
Collision detection algorithm based on OBB	$O(n^2)$	45.152	33.31
Collision detection algorithm based on hierarchical bounding box tree	$O(n \log n)$	42.173	35.26
Algorithm in this paper	$O(n \log n)$	38.106	38.53

From the table 1. We can see that by using the composite bounding boxes the average collision detection time is smaller than the uniform bounding boxes, and the collision detection time box, frame rate also have a little increase. After the parallel detection method is added into the algorithm, the efficiency of the algorithm is further improved.

## Conclusion

In this paper, it propose a parallel collision detection algorithm based on the complex balanced binary bounding box tree. According to the algorithm in Zhu Yuan Feng [12], this paper achieved a reasonable hybrid hierarchical bounding box tree algorithm, according to each object or collection of elements near-sphere value will be divided into two kind of bounding box, thereby increasing the tightness of the bounding box, and the collision elimination effect is very good. In addition, the algorithm uses a parallel detection method, which makes the collision detection process significantly improved. Finally, through the contrast experiment proved that, compared to other unified bounding box collision detection algorithm or serial algorithm based on this algorithm, this paper requires less time and collision detection efficiency has been greatly improved in a certain extent.

## References

- [1] Ding X J. Research on Collision Detection Algorithm Based on Combined Bounding Box [J]. Advanced Materials Research, 2014, 912-914(912-914): 1353-1356.
- [2] Chang J W, Wang W, Kim M S. Efficient collision detection using a dual OBB-sphere bounding volume hierarchy[J]. Computer-Aided Design, 2010, 42(1):50-57.

- [3] Palmer I J, Grim dale R L. Collision Detection for Animation using Sphere-Trees [J]. Computer Graphics Forum, 2003, 14(2):105-116.
- [4] Gino van den Bergen. Efficient Collision Detection of Complex Deformable Models using AABB Trees [J]. Journal of Graphics Tools, 1998, 2(4):1-13.
- [5] Eberly D. Dynamic Collision Detection using Oriented Bounding Boxes [J]. Geometric Tools, 2001.
- [6] Klosowski J T, Held M, Mitchell J S B, et al. Efficient collision detection using bounding volume hierarchies of k-DOPs[J]. IEEE Transactions on Visualization & Computer Graphics, 1998, 4(1):21-36.
- [7] Sun Jinguang, Wu Suhong, Zhou Lin plot. Bounding box collision detection optimization algorithm based on shape classification [J]. Computer applications and software, 2016 (2): 242-245(In Chinese).
- [8] Zhang Yingzhong, fan Chao, Luo Xiaofang. An algorithm for the separation of motion trajectory of convex polyhedron by collision detection [J]. Journal of computer aided design and computer graphics, 2013, 25(In Chinese).
- [9] Bai Lifang, Chang Jiang Zhi, Wang Yu, et al. A test method for oriented bounding box intersection based on effective constraints [J]. Journal of computer aided design and graphics, 2016, 28 (10): 1757-1766(1): 7-14(In Chinese).
- [10] Information on <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138252.html>.
- [11] Information on <http://dst.lbl.gov/ACSSoftware/colt/>.
- [12] Zhu Yuan Feng, Meng Jun, Xie Guanghua, et al. Research on real time collision detection based on hierarchical bounding box [J]. Journal of system simulation, 2008, 20 (2): 372-377(In Chinese).