Journal of System Simulation

Volume 30 | Issue 11 Article 23

1-4-2019

Improved Hybrid Bounding Box Collision Detection Algorithm

Wang Chao
1.Missile Engineering Academy, Rocket Force Engineering University , Xi'an 710038, China; ;

Zhili Zhang
1.Missile Engineering Academy, Rocket Force Engineering University , Xi'an 710038, China; ;

Long Yong
2.China Aerospace Science and Engineering System Simulation Technology Company Ltd, Beijing 100095, China;

Shaodi Wang
1.Missile Engineering Academy, Rocket Force Engineering University , Xi'an 710038, China; ;

Improved Hybrid Bounding Box Collision Detection Algorithm

Abstract

Abstract: For the construction of the oriented bounding box (OBB) takes a long time and the efficiency of the scene collision detection algorithm is not high, *a fast-adaptive hybrid bounding box* collision detection algorithm is proposed, which *optimizes the bounding box of hierarchy and the way of storage*. A large number of objects that cannot be intersected are excluded by means of the eight-tree space division and the rough detection of the sphere, and the exact intersecting test of the OBB can be made. The experimental result shows that compared with the OBB algorithm, the complexity of the bounding box construction is reduced, and the more the number of objects in the scene, the higher the accuracy of collision detection.

Keywords

collision detection, space division, hybrid bounding box, spheres, oriented bounding box (OBB)

Recommended Citation

Wang Chao, Zhang Zhili, Long Yong, Wang Shaodi. Improved Hybrid Bounding Box Collision Detection Algorithm[J]. Journal of System Simulation, 2018, 30(11): 4236-4243.

第 30 卷第 11 期 系统仿真学报© Vol. 30 No. 11
2018 年 11 月 Journal of System Simulation Nov., 2018

Improved Hybrid Bounding Box Collision Detection Algorithm

Wang Chao[1], *Zhang Zhili*[1], *Long Yong*[2], *Wang Shaodi*[1]
(1.Missile Engineering Academy, Rocket Force Engineering University , Xi'an 710038, China;
2.China Aerospace Science and Engineering System Simulation Technology Company Ltd, Beijing 100095, China)

Abstract: For the construction of the oriented bounding box (OBB) takes a long time and the efficiency of the scene collision detection algorithm is not high, *a fast-adaptive hybrid bounding box* collision detection algorithm is proposed, which *optimizes the bounding box of hierarchy and the way of storage*. A large number of objects that cannot be intersected are excluded by means of the eight-tree space division and the rough detection of the sphere, and the exact intersecting test of the OBB can be made. The experimental result shows that compared with the OBB algorithm, the complexity of the bounding box construction is reduced, and the more the number of objects in the scene, the higher the accuracy of collision detection.

Keywords: collision detection; space division; hybrid bounding box; spheres; oriented bounding box (OBB)

改进的混合包围盒碰撞检测算法研究

王超[1]，张志利[1]，龙勇[2]，王韶迪[1]
（1.火箭军工程大学导弹工程学院，陕西 西安 710038；2.航天科工系统仿真科技有限公司，北京 100095）

摘要：针对方向包围盒(OBB)构造耗时长，场景碰撞检测算法效率不高的问题，提出了一种*快速自适应的混合包围盒*碰撞检测算法，优化了包围盒的层次结构和存储方式，通过八叉树空间分割和包围球的粗检测排除大量不可能相交的物体，再对可能相交的对象作 OBB 包围盒的精确相交测试。实验结果表明，改进的算法较经典的 OBB 算法相比，降低了包围盒构造的复杂度，且场景中物体数越多，碰撞检测的精确度越高。

关键词：碰撞检测；空间分割；混合包围盒；包围球；方向包围盒
中图分类号：TP391.9 文献标识码：A 文章编号：1004-731X (2018) 11-4236-08
DOI: 10.16182/j.issn1004731x.joss.201811023

Introduction

Virtual reality is the use of computer simulation technology to create a virtual three-dimensional world of human-computer interaction technology. In a virtual three-dimensional world, real-time collision detection and response between objects in a scene is an important guarantee of system realism. In order to enhance the user's immersive experience, it is necessary to accurately detect these collisions and make corresponding actions in real time. The task of collision detection is to determine whether multiple objects occupy the same area at the same time. The

Received: 2018-05-27 Revised:2018-06-25;
Biographies: Wang Chao(1994-), male, Anhui Suzhou, Master, research direction: for launching missile theory and technology; Zhang Zhili (1966-), male, Henan Puyang, Professor, Ph.D. research direction: weapon system simulation technology.

simplest collision detection algorithm is to test all the objects to be tested. However, when the complexity of the model increases, the number of tests and the complexity increase sharply. Therefore, how to reduce the number of intersection tests is to improve the collision detection. The focus of algorithmic efficiency[1].

Geometric methods, spatial decomposition methods, and hierarchical bounding box methods are common collision detection methods. It is difficult to meet the accuracy and real-time nature of collision detection by simply using a single method[2]. This paper combines the advantages of spatial segmentation and hybrid bounding box, rough detection and accurate detection, and uses adaptive fast construction of OBB algorithm. On the premise of ensuring the accuracy, the efficiency of collision detection is improved.

1 Hierarchical bounding box

1.1 Hierarchical bounding box tree construction

In the bounding box tree, the top-level bounding box is the beginning of each traversal, and the number of tests is the most. According to the literature[3], when the two levels of bounding box trees perform collision detection, the collision detection formula between them is:

$$T = N_b \times C_b + N_p \times C_p + N_u \times C_u + C_d \quad (1)$$

In the formula: N_b: The number of bounding boxes participating in the intersection test; C_b: The cost paid for each pair of bounding box intersection tests; N_p: Number of pairs of geometric elements participating in the intersection test; C_p: The cost paid for each pair of geometric element intersection tests; N_u: The number of bounding boxes updated each time; C_u: The cost of updating a bounding box; C_d: The cost of the software after deformation.

When the object rotates, the ball does not need to be updated. Without considering the deformation of the software, the collision detection cost of the system is:

$$T_{\text{Total}} = N_b \times C_{b\ \text{Surround the ball}} + N_p \times C_{\text{p Surround the ball}} \quad (2)$$

Therefore, the initial judgment of the bounding sphere can effectively reduce the consumption of collision detection. This paper adopts a top-down approach to construct a bounding box tree that surrounds the sphere-OBB[4], as shown in Fig. 1.

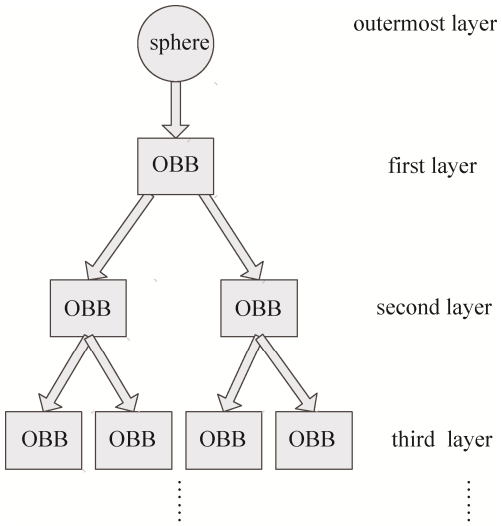


Fig. 1 Hierarchical bounding box tree

For the partition of the OBB, the bounding box of the global model is first calculated, and then each node is recursively divided. The division of the OBB is relatively mature. We will not go into details here. This article selects the division method of the split plane.

When the OBB is divided, sometimes some geometry will cross the splitting plane. In this paper, the penalty function method is used to determine the splitting plane. The penalty function is defined as follows:

$$f(\text{p}) = \left| n^l - n^r \right| + \lambda n^c \quad (3)$$

In the formula, the split surface, and the number

第 30 卷第 11 期 系统仿真学报 Vol. 30 No. 11
2018 年 11 月 Journal of System Simulation Nov., 2018

of geometric bodies in the left and right bounding boxes and across the split surface, respectively, is an influence factor. The ideal split plane is the one with the smallest penalty function in all axial directions, is:

$$\min\left\{\begin{array}{l}\min\left[f(p)P_x \perp x\_axis, P_x \in [x_{\min}, x_{\max}]\right]\\ \min\left[f(p)P_y \perp y\_axis, P_y \in [y_{\min}, y_{\max}]\right]\\ \min\left[f(p)P_z \perp z\_axis, P_z \in [z_{\min}, z_{\max}]\right]\end{array}\right\} \quad (4)$$

Among them, x、y、z are three axial directions for OBB.

1.2 The structure of the bounding box

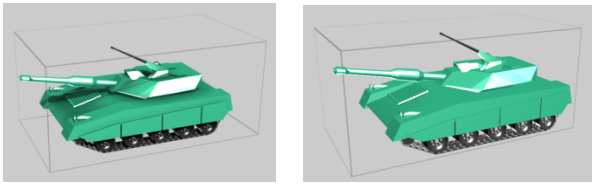
Choosing an appropriate bounding box, the tightness of the integrated envelope and the complexity of the algorithm are the key to efficient collision detection[5].

1) Sphere

Sphere: It refers to the smallest sphere that contains geometric objects. It only needs two parameters description of the sphere center and radius. The intersection test only needs to determine the distance between the two enveloping balls and the radius of the two enclosing balls.

2) Fast adaptive OBB based on approximate convex hull[6]

Since the influence of the object convex hull (including the minimum convex set of all vertices) is not considered, the conventional OBB takes the mean of the sum of the vertices of all the triangles as its center position, resulting in non-uniform size of each triangular patch and constructing a bounding box. There is a deviation in direction, as shown in Fig. 2(a). Therefore, this paper uses a fast adaptive OBB algorithm based on approximate convex hull, as shown in Fig. 2(b). The specific method is: In order to reduce the influence of the triangular facet in the model, a method based on the mean and covariance surrounds the box on the surface of the convex surface far away from the center of the geometric model.



(a) Conventional OBB (b) Improved OBB

Fig. 2 Traditional bounding box and improved OBB

Methods as below:

Step 1: Calculate the model center point[7]. The three vertices of the i-th triangle on the convex hull of the geometric model are denoted by a_i, b_i, and c_i. o_i denotes the center point, S_i denotes the surface area, and W denotes the total area of the convex hull. The center point of the triangle patch is represented as[8]:

$$o_i = (a_i + b_i + c_i)/3 \quad (5)$$

The surface area of a triangular patch is expressed as:

$$S_i = \left|(a_i - b_i) \times (a_i - c_i)\right|/2 \quad (6)$$

The total area of the convex hull is expressed as:

$$W = \sum_{i=1}^{n} S_i \quad (7)$$

The center point of the bounding box is expressed as:

$$O = \left(\sum_{i=1}^{n}(S_i \cdot o_i)\right)/2 \quad (8)$$

Step 2: Calculate the model projection value on the coordinate X axis. The model is projected on the X axis. The minimum and maximum values of the projection area are $P_{\min}$、$P_{\max}$。

Step 3: The model vertices are divided according to the x projection position. The projection area of the model on the x axis is divided into k-divisions. l is the length of a plot. The formula is:

$$l = \frac{P\max - P\min}{k}, k \in Z^+ \quad (9)$$

According to the projection position of each vertex on the *X* axis, the vertex set s of the model is divided into *k* subsets, and the encoding of each subset is calculated. The i-th subset is denoted by S_i:

$$Si = \{(x,y,z)\left|P_{x\min} + i \times l \le x - P_{x\min} < P_{x\min} + (i+1) \times l\right\} i = \overline{1,k} \quad (10)$$

Step 4: Extract the set of vertices away from the center of the model and construct the vertex set S for the bounding box. Find the vertices corresponding to the minimum and maximum values of the *Y* axis and *Z* axis coordinates of the *k* subsets and construct a set of maximal value points S.

$$S' = \{(x,y,z)\,|\ \text{At least one of x,y,z is the maximum or minimum value in the i subset, } i=\overline{1,k}\} \quad (11)$$

Step 5: The OBB is constructed on the extracted vertex set S. Calculate the expected value of the vertex set S and the covariance matrix[9] according to equations (12) and (13).

$$u = \frac{1}{6m}\sum_{i=1}^{m}(p^i + q^i + r^i) \quad (12)$$

$$C_{jk} = \frac{1}{3m}\sum_{i=1}^{m}\overline{p_j^i}\,\overline{p_k^i} + \overline{q_j^i}\,\overline{q_k^i} + \overline{r_j^i}\,\overline{r_k^i} \quad (13)$$

Among them, m is the number of triangular faces, $\overline{p^i} = P^i - u$, $\overline{q^i} = q^i - u$, $\overline{r^i} = r^i - u$ is a vector in R^3, C_{jk} is a 3×3 Covariance matrix.

According to the covariance matrix, its eigenvectors are calculated, and the results are taken as the three directions of the direction bounding box. Calculate the projection length of the vertex set S' in each direction and complete the calculation of the direction bounding box.

For the bounding box of the tank, the volume calculated by this method is 6 835.488 cm^3, and the volume obtained by the traditional method is 7 319.264 cm^3. Compared with the traditional method, the time cost of the improved algorithm to construct the OBB is shown in Tab. 1.

Tab. 1 Bounding box time overhead

Parameter/s	value
OBB Build Time	2.728
Improved Algorithm Bounding Box Build Time	1.463
OBB time/improvement algorithm time	1.865

By adopting the area weighting method of approximate convex edges, the deviation of the direction of the bounding box brought about by the construction method of the conventional structure OBB is compensated, and the construction time is shortened by about 1.9 times.

1.3 Bounding box tree storage optimization

A bounding box tree is constructed using a binary tree structure. Each node stores corresponding bounding box information. The leaf nodes contain geometric information of the triangles. Then, the bounding box tree of *N* leaf nodes has 2*N*–1 nodes that need to be stored. To skip the intersection test of the leaf node bounding box, the triangle information of the leaf node is stored in the parent node, and the intersection test between the basic geometric elements is directly performed. At this time, only *N*–1 nodes need to be stored, thereby saving storage space[10], As shown in Fig. 3. The number of traversal nodes required for collision detection is also significantly reduced.

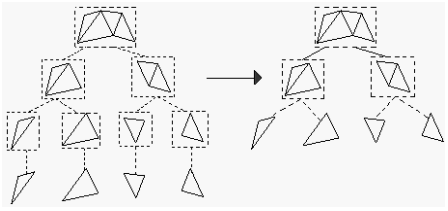


Fig. 3 Optimization of the bounding box storage

2 Collision detection algorithm

2.1 Algorithm idea

The bounding box collision detection algorithm in this paper includes two stages: preliminary detection

第 30 卷第 11 期 系统仿真学报 Vol. 30 No. 11
2018 年 11 月 Journal of System Simulation Nov., 2018

and precise detection. First, the space is divided by the uniform partition method, the objects in the same space are sorted before detection, the bounding sphere intersection detection is performed only on the objects in the same space, the objects that cannot be intersected are excluded, and then the OBB pair may be used. The intersecting objects are accurately detected. The algorithm flow is shown in Fig. 4[11].

2.2 Initial inspection phase

2.2.1 Preliminary judgment of space division

Spatial segmentation[12] is to divide the space to be detected into several contiguous regions and only detect objects in the same region to reduce the number of tests. The commonly used spatial decomposition methods are uniform grids, trees, and spatial sorting algorithms. In this paper, the improved octree method is used to partition the uniform mesh space, as shown in Fig. 5.

The space is first divided into cubes of size *a* and numbered. Use the number $O_i\left(O_1,O_2,\cdots,O_n\right)$ to mark the *n* objects in the space. Set up two linked lists *Alist*1 and *Alist*2 to record the spatial units in which each object is located. The linked list *Alist*1 is set in the subspace data structure to record the number *F* of objects in the same subspace. The linked list *Alist* 2 is set in the object structure data to record the object identifiers adjacent to the object in each unit. Only the adjacent objects whose object identifiers are larger than the self-identifiers are recorded here, to prevent one object from being included in multiple subspaces at the same time. Repeated tests or missed calculations. For object $O_i\left(1<i<n\right)$, the set of objects that occupy the same small space as it is $O_k\left(1<k<n\right)$, then the identifier stored in the linked list is $O_k\left(i<k<n\right)$, so as to avoid repetitive testing.

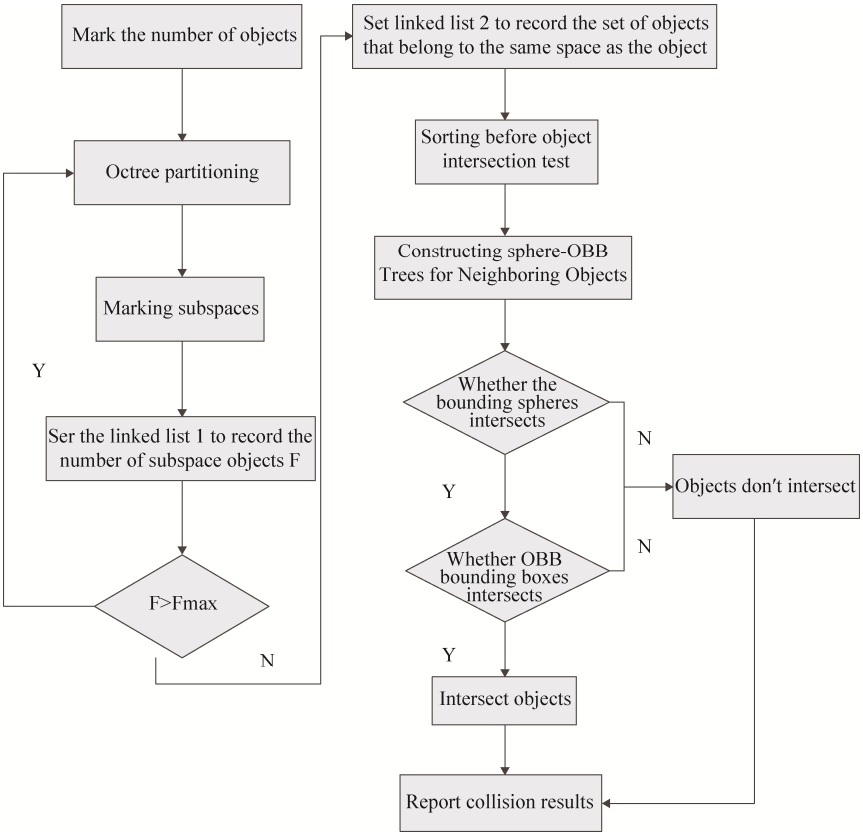


Fig. 4 Algorithm flowchart

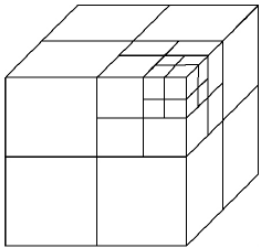


Fig. 5 Space splitting using octree

Set the maximum number of single space objects to F_{max}. If the number F of a spatially adjacent object exceeds the threshold F_{max}, the space continues to be split until the number of recorded objects in this space is less than or equal to the threshold F_{max}. Set Δ as the difference between the number of objects E in the two-record linked list, if Δ=0, then stop the space split.

2.2.2 Space object detection before sorting

Collision detection is performed on the same space area as described above, and a double traversal process is conventionally performed. The intersection test is performed on each two objects to cause each object to be detected twice by the same object. For example, after No. 1 was tested by No. 2, No. 2 was repeated test by No. 1. A list of object numbers in the same area is detected only for objects in the same table, avoiding duplication, as shown in Tab. 2.

Tab. 2 Intersection test table

Object number	1	2	3	...	n
1	×	(2,1)	(3,1)	...	(n,1)
2	×	×	(3,2)		(n,2)
3	×	×	×	...	(n,3)
...	×	×	×	×	(n,...)
n	×	×	×	×	×

2.2.3 Intersecting ball detection

Judging whether the two bounding spheres intersect is based on the distance between the center of the two spheres and the radius of the two spheres[13]:

$$|o_1o_2|^2 > (r_1 + r_2)^2 \tag{14}$$

In the formula, o_1 and o_2 are two spheres that surround the sphere, r_1 and r_2 are two spheres that surround the sphere, and $|o_1o_2|$ is the distance between the two spheres. When formula (14) is established, it is determined that the two bounding spheres are apart, otherwise, it is judged as intersecting, and the programming is performed using the Visual C++ language as follows:

```
int Test SS（Sphere a，Sphere b）
{
Vector d=a.c－b.c；
float dist=Dot（d，d）； // The square of the distance between two surrounding spheres
float radius Sum=a.r+b.r；
return dist<=radius Sum*radius Sum；
}
```

2.3 Accurate detection phase

The precise detection phase is to use the OBB that surrounds the object to perform the intersection test, using the separation axis theorem method, as shown in Fig. 6[14].

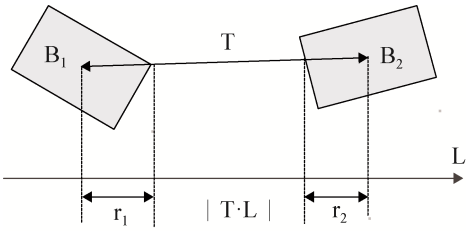


Fig. 6 Separation axis theorem

The two bounding boxes are on the L axis. The projected radii of the two bounding boxes are r_1 and r_2, and the projection distance between the center points is $|T \cdot L|$. If $|T \cdot L| > r_1 + r_2$, they are separated.

The determination of the intersection state of the OBB requires the testing of up to 15 potential

第 30 卷第 11 期 系统仿真学报 Vol. 30 No. 11
2018 年 11 月 Journal of System Simulation Nov., 2018

separation axes, including the three axes of B_1, the three axes of B_2, and the nine axes perpendicular to each axis. If the projections of two bounding boxes on any of the above axes do not overlap, the two OBBs are separated. Otherwise, the two bounding boxes intersect and are programmed using the Visual C++ language as follows:

```
int Test OO（OBB a，OBB b，int k）
{
..... // Get OBB a and OBB b basic geometric information
// Separation axis detection
for (int i = 0; i < 3; i++)
{
ra = box0.extents[i];
rb =box1.extents[0] * FR [i, 0] + box1.extents[1] * FR [i, 1] + box1.extents[2] * FR [i, 2];
t = Mathf.Abs(T[i]);
if (t > ra + rb) return 0;
 }
...../// The separation plane is parallel to one side of OBB a, OBB b
ra=box0.extents[1]*R[2,0]+box0.extents[2] * FR[1, 0];
rb=box1.extents[1]*FR[0,2]+box1.extents[2] * FR[0,1];
t = Mathf.Abs(T[2] * R[1, 0] - T[1] * R[2, 0]);
if (t > ra + rb) return 0；
..... // The separation plane is also perpendicular to one side of OBB a and one side of OBB b
return 1；
}
```

3 Simulation results and analysis

This algorithm is implemented on the VS2010 development platform using OpenGL graphics program development. In order to effectively verify the performance of this algorithm, different numbers of triangular facets were used for collision testing. The object is randomly moved at any initial speed and collision detection is performed using two algorithms respectively. The collision detection time is shown in Fig. 7.

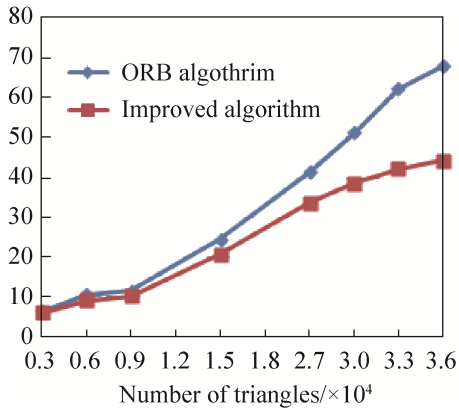


Fig. 7 Comparison of collision detection algorithms

It can be seen from Fig. 7 that using the improved algorithm shortens the collision detection time and improves the real-time performance of the collision detection, and when the number of triangular facets increases, the superiority of the algorithm can be better displayed.

This is because the OBB algorithm constructed in this paper extracts a small number of points close to the model surface, reduces the time overhead of the Hybrid level bounding box tree that needs to be constructed, and combines the space segmentation method with the bounding sphere intersection test in a complex multi-object scene. As soon as possible, the objects farther away were excluded, which greatly reduced the amount of calculations.

4 Conclusion

This paper presents an improved collision detection algorithm based on hybrid bounding box and space segmentation. The use of an approximate

convex hull area weighting method for the construction of the OBB increases the compactness of the bounding box and reduces the time overhead of the bounding box construction. Improved the storage structure of the bounding box tree, and numbered the objects prior to the intersection detection. This improves the detection speed while ensuring the accuracy of the operation, and fully combines the advantages of the bounding sphere and the OBB. Experimental results show that the algorithm reduces the computational complexity and improves the real-time performance of collision detection. With the increase of the number of triangular faces of the three-dimensional object, the efficiency of this collision detection algorithm is more obvious.

This algorithm also has some deficiencies. The next research will focus on the following two points:

(1) How to balance the relationship between the partition depth and the volume of the grid space in the space division as the model changes.

(2) Look for a generic bounding box type that can adaptively adjust the surrounding objects.

References:

[1] Pi Xuexian, Yang Xudong, Li Sikun. High-perform ance navigation and rendering of very-largescale landscape and seascape[C]//Computer A ided Design and Computer Graphics，2005 Ninth International Conference, 2005.

[2] SONG Tao, SHU Tao, MEI Zhao, et al. A Collision Detection Algorithm Based on Spatial Partitioning and Hybrid Bounding B[J]. Firepower and Command Control (S1002-0640), 2016, 41(11): 94-97.

[3] JIANG Guangyan. Research and Application of Collision Detection Algorithm Based on Bounding Box [D]. Chengdu: University of Electronic Science and Technology of China, 2012.

[4] M Teschner, S Kimmerle, B Heidelberger, et al. Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs [J]. IEEE Transactions on Visualization & Computer Graphics (S1077-2626), 2002, 4(1): 21-36.

[5] LAI K-C, KANG S-C. Collision detection strategies for virtual construction simulation[J]. Automation in Construction(S0926-5805), 2009, 18(6): 724-736.

[6] GAN Jianhong, PENG Qiang, DAI Pei-dong, et al. Improved Collision Detection Algorithm Based on Oriented Bounding Box [J]. Journal of System Simulation(S1004-731X), 2010, 23(10): 2169-2173.

[7] DIMITORY D, KNAUER C, KRIEGEL K, et al. Bounds on the quality of the PCA bounding boxes[C]. Computational Geometry: Theory and Applications. Amsterdam: Elsevier, 2009, 42(8): 772-789.

[8] WANG Wei, MA Jun, LIU Wei. Research and Application of Collision Detection Based on Oriented Bounding Box[J]. Computer Simulation (S1006-9348), 2009, 26(9): 180-183.

[9] YU Haijun, MA Chunyong, ZHANG Tao, et al. Fast collision detection algorithm based on image space [J]. Journal of Computer Applications (S1001-9081), 2013, 33(2): 530-533.

[10] CHEN Chen, ZHANG Weiming, CHU Ning. Optimized collision detection algorithm based on bounding volume hierarchy and triangular facets [J]. Manufacturing Technology and Machine Tools (S1005-2402), 2013.

[11] Wang Wei, Zhou Cheng, Yang Yun, et al. Virtual Maintenance-oriented collision detection algorithm[J]. Computer Applications and Software (S1000-386X), 2016, 33(4): 235-238.

[12] GUO Lingyun, ZHENG Yanbin, LIU Jingjing. Fast collision detection algorithm based on spatio--temporal correlation [J]. Computer Applications and Software (S1000-386X), 2013, 30(5): 174-176.

[13] JIANG Xiaolu, LIU Yuan. New collision detection algorithm based on hybrid hierarchical bounding box[J]. Computer Engineering and Applications (S1002-8331), 2012, 48(6): 143-145.

[14] SHI Xusheng, QIAO Lihong, ZHU Zuowei. Algorithm of Collision Detection Based on Improved Orieted Bounding Box [J]. Journal of Hunan University(Natural Sciences) (S1674-2974), 2014, 41(5): 26-31.

https://dc-china-simulation.researchcommons.org/journal/vol30/iss11/23
DOI: 10.16182/j.issn1004731x.joss.201811023