Review of Bounding Box Algorithm Based on 3D Point Cloud

He Siwei

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710032, China
E-mail: 759529454@qq.com

Liu Baolong

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710032, China
E-mail: liu.bao.long@hotmail.com

Abstract—**Collision detection is mainly to judge whether there is intersection between virtual models, which means there is collision. Bounding box is one of the important methods of collision detection. It uses regular geometry with simple structure to replace the complex model to be detected. A simple point cloud model usually contains hundreds or thousands of polygons when meshed. When collision tests are performed directly on the geometry of two model objects, the calculation process is relatively complex. In order to reduce the computational cost, the bounding box test of the model was performed before the geometry intersection test. Only when the bounding box has a collision, can further accurate intersection detection be carried out. This paper mainly introduces the content and significance of the bounding box, and compares four kinds of common bounding boxes and their advantages and disadvantages as well as the application scenarios of the bounding box. Firstly, the content and significance of the bounding box are expounded. Secondly, it is analyzed according to different bounding boxes. Finally, the different bounding box algorithms are summarized, and the advantages and disadvantages of different bounding boxes and application scenarios are pointed out.**

Keywords-Virtual Reality; Point Cloud; Collision Detection; Bounding Box

I. INTRODUCTION

Virtual reality technology is closely related to many disciplines, which is widely used in simulation teaching, medical surgery, urban planning, industrial simulation and other fields [1]. In various virtual reality applications, objects in the virtual system often collide with each other due to the interaction between users and objects in the virtual environment. In order to maintain the authenticity of the virtual environment, the system needs to timely judge the occurrence of collisions and make corresponding operations, otherwise it will violate the authenticity of the real world and the user's experience [2]. Collision detection in virtual reality technology is a hot research direction is also one of the most important problems. Collision detection is to determine whether two or more objects in the same space have contact or intersection [3]. If a collision occurs, the collision response will be triggered and the virtual reality system will be guided to the next operation. Collision detection algorithms can be roughly divided into two categories: time domain and space domain [4]. Bounding box is a kind of detection algorithm based on spatial domain. It has the advantages of high detection efficiency and simple detection process [5].

For the complex model to be tested, the thought of bounding box is to surround the model to be tested with simple bounding box whose volume is slightly larger than the model to be tested and whose geometric features are regular [6]. When collision detection is done on two objects, the first detection is to check whether the bounding box intersects. If not, the two objects do not intersect. On the contrary, it is necessary to further accurately detect the two objects. This method can quickly exclude disjoint objects and reduce unnecessary calculation.

II. BOUNDING BOX

Many scholars have studied and explored the bounding box algorithm. The results show that the choice of bounding box algorithm is one of the key factors affecting the result of collision detection. The more compact the virtual model under test is wrapped by a bounding box, the simpler the set features are, indicating that its wrapping effect is better. But at the same time in bounding box algorithm, its simplicity and compactness are usually contradictory [7]. So the researchers came up with a different algorithm. Palmer and Grimsdale et al. [8] proposed a collision detection algorithm based on spherical bounding boxes. Smith et al. [9] proposed an intersection detection algorithm based on AABB axial bounding box. Both the bounding sphere and the AABB bounding box are simple in construction, but poor in tightness. Therefore, Gottschalk et al. [10] proposed the construction of OBB bounding boxes based on the theory of Smith's AABB bounding boxes. Klosowski et al. [11] also proposed a collision detection algorithm using *k*-DOP bounding boxes by studying the characteristics and spatial tightness of AABB bounding boxes. Therefore, the common bounding box algorithms include Bounding Sphere (Sphere), Axis-Aligned Bounding Box (AABB), Oriented Bounding Box (OBB) and discrete oriented polytope (*k*-DOP).

A. Bounding Sphere

Bounding sphere is defined as the smallest volume sphere that can encase the model to be tested [12]. It is a kind of bounding body with good simplicity and flexible structure. To determine the bounding sphere, it first need to determine the center of the bounding sphere, which is composed of the mean value of the vertex coordinates of all elements in the model to be measured. Then determine the radius of the sphere, which is the distance between the center of the ball and the points specified by the three maximum coordinates. The three-dimensional structure of the bounding sphere is shown in Figure 1.
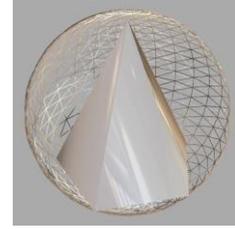
Figure 1.   Three dimensional of the Bounding Sphere.

The radius of the sphere can be expressed as the distance between the farthest vertex and the center of the sphere, and the center of the longest axis is the center surrounding the sphere. The bounding sphere is expressed as formula (1):

$$R = \{(x, y, z) \mid (x - O_x)^2 + (y - O_y)^2 + (z - O_z)^2 < r^2\} \qquad (1)$$

O_x, O_y, O_z represent the x, y, z coordinates of the spherical center O, and r is the radius.

The principle of intersecting detection between bounding spheres is as follows: judging from the relationship between the sum of the center distance and radius between two spheres, if A and B represent two bounding spheres, the intersection of A and B is only necessary and sufficient if $|d| < (r_1 + r_2)$ is true. This is shown in Figure 2.
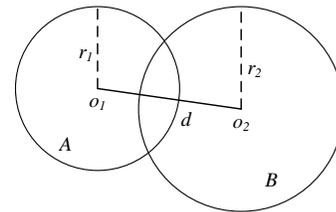
Figure 2.   Intersecting test between spheres.

For update operations, only rotation and translation are needed to satisfy any operation in space. The update operation of the bounding sphere is that no matter the object rotates or moves along any axis or any point, the bounding sphere always wraps the model under test. Therefore, for a translation operation, it just need to do the same transformation of the center of the sphere according to the translation vector, while for a rotation, it don't need to do any update operation, and the position of the center of the sphere doesn't change no matter how rotate it. When compared to other bounding boxes, the advantage of the bounding sphere is obvious. It requires the least

amount of computation compared to other bounding boxes.

By reason of the foregoing, although the calculation of the bounding sphere is small, its tightness is also poor. There are too many redundant calculations for most objects with uneven spatial distribution such as irregular geometric models or elongated objects. Because of its structural characteristics, it is suitable for the movement environment where the detection accuracy is not high. Therefore, when selecting the bounding box, the bounding box that is closer to the model to be tested should be selected, and the best scheme should be selected after comprehensive consideration of the advantages and disadvantages.

B. Axis-Aligned Bounding Box

An Axis-Aligned Bounding Box(AABB) is defined as the smallest hexahedron that wraps the model under test in the same direction as the coordinate axis, so only six scalars are needed to describe an AABB [13]. It was the first bounding box to be used. Determining an AABB requires calculating the maximum and minimum values of the x, y, and z coordinates of the vertices of each element in the set of basic geometric elements that make up the object. The three-dimensional structure of the AABB is shown in Figure 3.
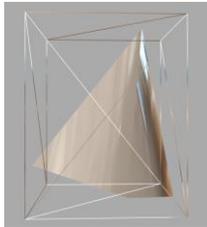
Figure 3.   Three dimensional of the Axis-Aligned Bounding Box.

In the construction of AABB, all bounding boxes have the same direction along the axial direction of the local coordinate system of the model. AABB is expressed as formula (2):

$$R = \{(x, y, z) | x_{\min} \le x \le x_{\max}, y_{\min} \le y \le y_{\max}, z_{\min} \le z \le z_{\max}\} \quad (2)$$

x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max} represents the minimum and maximum values of the projection of the model on the x, y, and z axes.

The principle of intersection detection between AABB is as follows: if the vertex of the model intersects all the projected intervals on the three coordinate axes, collision may occur between the detection models. If A and B represent two AABB, where l_A and l_B are the projections of A and B bounding boxes on the X-axis respectively, x_A and x_B are A respectively, and the projections of O_A and O_B on the X-axis of the middle points of B bounding boxes. If $|d| < (l_A + l_B)$, then it means that the projection of the bounding box A and B on the X-axis overlaps, otherwise, the projection of the bounding box on the X-axis does not overlap. In the three-dimensional space, the intersection test of AABB bounding box needs at most 6 comparison operations. This is shown in Figure 4.

Both translation and rotation have an impact on the AABB update operation, which is not complicated due to the simplicity of AABB. In the case of translation operation, there is no need to rebuild AABB, but only need to change the vertex coordinates of AABB according to the direction vector and displacement of target model translation. In the case of rotation operation, the coordinates of the rotated vertices need to be calculated by rotation Angle and then re-projected to the x, y and z axes to get the maximum and minimum values, and then rebuild AABB. Similarly, AABB can be used in a wide range of scenarios, including rigid body and soft body.
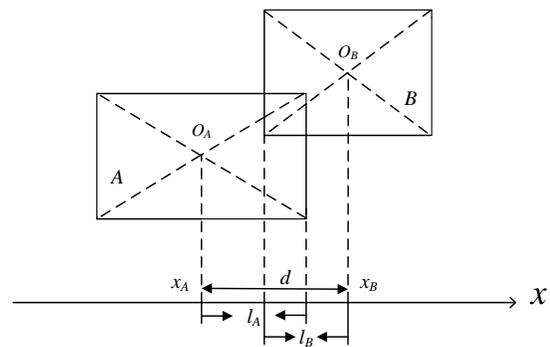
Figure 4.   Intersecting test between AABB.

C. Oriented Bounding Box

Oriented Bounding Box(OBB) is defined as the smallest cuboid that wraps the model to be tested and is in an arbitrary direction relative to the coordinate axis[14]. The most important feature of

OBB is its arbitrary orientation, which makes it possible to surround the model as closely as possible according to the shape characteristics of the surrounded model, but also makes its intersection test complicated. OBB approximates the model more closely than bounding sphere and AABB, which can significantly reduce the number of bounding volumes, thus avoiding the intersection detection of a large number of bounding volumes. The difficulty is to determine the direction to minimize the volume of OBB according to the three-dimensional structure of the model to be tested. The three-dimensional structure of the OBB is shown in Figure 5.
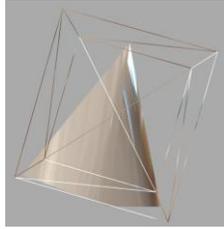
Figure 5.   Three dimensional of the Oriented Bounding Box.

Because of the flexible orientation of the OBB and its ability to wrap the model under test more tightly, it is difficult to create and update the bounding box for collision detection. The OBB region can be expressed as formula (3):

$$R = \{O + ar_1v_1 + br_2v_2 + cr_3v_3 | a,b,c \in (-1,1)\} \qquad (3)$$

O represents the center of OBB. r_1, r_2, r_3 represent the radii of x, y, z coordinate axes respectively. v_1, v_2, v_3 are used to calculate the direction of OBB respectively, which is a vector mutually orthogonal to the direction of coordinate axes.

The principle of OBB's intersecting detection is complicated, and it uses the Separating Axis Theorem (SAT) test. If you can find A straight line so that the bounding box A is exactly on one side of the line and the bounding box B is exactly on the other side, then the two bounding boxes do not overlap. This line is the separation line and must be perpendicular to the separation axis. The two-dimensional schematic diagram of the testing principle is shown in Figure 6. If A and B

represent two OBB bounding volumes, then A and B intersect only if $|T \cdot L| \le l_A + l_B$ is true. T represents the displacement vector between A and B, and L is the unit vector parallel to the separation axis.
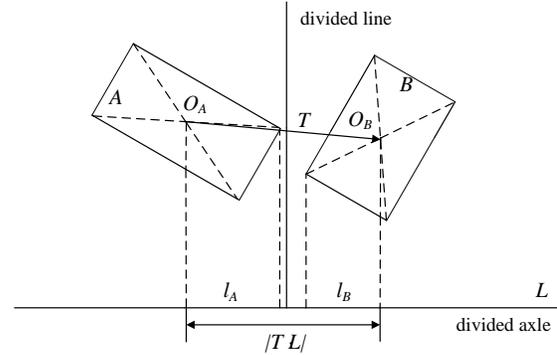
Figure 6.   Intersecting test between OBB.

OBB's update operations for rotation and translation are less computationally intensive. When the model is translated or rotated, the OBB needs to be updated. A new OBB can be obtained by carrying out the same displacement or rotation on all direction vectors of the model.

OBB is used in a narrow range of scenarios, only between rigid bodies. For soft body, OBB is not applicable. When the object is deformed, the update and reconstruction of OBB hierarchy tree is cumbersome and the calculation is very large, so the real-time detection conditions cannot be guaranteed. There is no good solution to the problem of updating the OBB hierarchy tree, and the cost of rebuilding the bounding box hierarchy tree is too high, so OBB is not suitable for collision detection between soft bodies.

D. k-DOP

The discrete oriented polytope (*k*-DOP) is defined as a convex polyhedron that wraps the model to be tested and the normal vectors of all faces come from a fixed set of orientation (k vectors) [15]. *K*-DOP can surround the model to be tested more closely than other bounding volumes. The more directional axes it has, the closer it is to the model to be tested. Therefore, its construction difficulty and phase test are more complex. In fact, the AABB bounding box mentioned above is a special case of 6-DOP. When k value is infinite, the *k*-DOP bounding box

is actually the convex hull of the model, so the tightness of the *k*-DOP bounding box is the best among these kinds of bounding boxes.

The common *k*-DOP generally includes 8-DOP, 14-DOP, 18-DOP and 24-DOP. Since each *k*-DOP of the same category uses the same and fixed directional axis, the test process of intersection between them is similar to that of AABB. The test process can be simply described as finding whether there are separate projection intervals between two *k*-DOP on k/2 directional axes in turn. If there is a separate projection interval, the two *k*-DOP are separated from each other, and vice versa, the two *k*-DOP bounding boxes intersect.

The biggest drawback of *k*-DOP is that even if objects in space rarely collide, it is necessary to perform an update rotation operation on the bounding box to redetermine the maximum and minimum values on the axis. *K*-DOP update operation is that when the model rotates, *k*-DOP update computation is relatively large. In this case, we cannot simply update *k*-DOP by the same rotation, but recalculate *k*-DOP after rotation. However, the cost of recalculating a *k*-DOP vertex is high, so the idea of linear programming is adopted for optimization. For soft body, *k*-DOP

needs to update the deformed leaf node first, and then update the parent node according to the bottom-up method, so *k*-DOP is suitable for both rigid body and soft body environment.

III.   SIMULATION AND ANALYSIS

This paper simulates the above three bounding box algorithms. The experiment uses VS2107 compilation environment and VTK visualization library. The 3D point cloud data is the rabbit point cloud data of Stanford in the format of pcd file. Entering point cloud data into it can get the bounding sphere, the bounding box of AABB and OBB that wraps the whole point cloud data. The simulation results are shown in Figure 7.

(a) Sphere   (b) AABB   (c) OBB
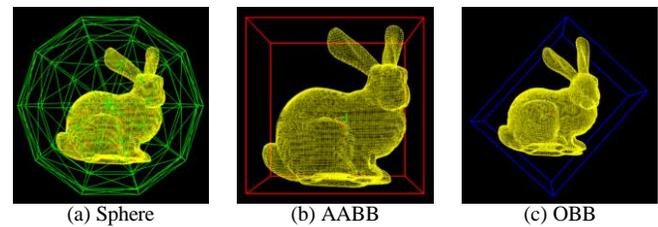
Figure 7.   Simulation results of Sphere, AABB and OBB.

Comparison of various bounding box algorithms is shown in Table Ⅰ.

TABLE I.   COMPARISON OF VARIOUS BOUNDING BOX ALGORITHMS

Bounding box type	Simplicity	Tightness	Collision Detection Difficulty	Update the Difficulty	Application Scenarios
Sphere	simpler	worst	easiest	easiest	All
AABB	simplest	worse	easier	more difficult	Both rigid body and soft body are applicable
OBB	most complex	better	most difficult	easier	All
***k*-DOP**	more complex	best	more difficult	most difficult	All

Through the analysis of the table, it is found that the bounding sphere is relatively simple compared with other bounding boxes in terms of construction difficulty, collision detection difficulty and update calculation difficulty, and it has a wide range of applications, including rigid body and soft body. The only drawback is that it has the worst tightness surrounding the model. In the scene where the detection accuracy is not high, the bounding sphere algorithm can be given

priority. AABB is relatively simple in terms of construction difficulty and collision detection difficulty, and has the same disadvantage as the bounding sphere, which has a relatively weak enveloping tightness. It is also relatively difficult to update the calculation. In particular, AABB is not suitable for models with narrow and long shapes and direction deviating from the coordinate axis. OBB has strong bounding compactness and less updating calculation, but its disadvantage is

that compared with other bounding boxes, it is more difficult to construct and collision detection, and its application scope is smaller. The tightness of *k*-DOP is ideal, but its disadvantage is that it is more difficult to construct, detect, and update calculations than other bounding boxes. In the case of no consideration of real-time, *k*-DOP bounding box is the most suitable for the scene with very high precision requirements.

IV. CONCLUSION

Bounding box algorithm is a very important algorithm idea in ray tracing and collision detection. This paper introduces some classical bounding box algorithms and analyzes their advantages and disadvantages as well as their application scenarios and scope. It can be seen from this article that there is no one kind of bounding box is the best choice, but according to different scenes to select the most appropriate bounding box. The bounding sphere is simple but with poor accuracy; AABB is simple but complicated to calculate; OBB has good accuracy and simple update but complicated to construct; *k*-DOP has the best compactness but the most complex update, so it is not suitable for scenes with high real-time performance. This paper provides a theoretical basis for constructing hierarchical tree by selecting appropriate bounding boxes to balance construction difficulty, compaction difficulty, detection difficulty and updating calculation difficulty.

ACKNOWLEDGMENT

REFERENCES

[1] Hongjie Chen. The Common Development of Virtual Reality Technology and Computer Technology Application [J]. International Journal of Computational and Engineering, 2020, 5(1).

[2] SUZUKI Masaya, NIWA Takeru, MASUDA Hiroshi. Path Planning in Engineering Plants Using Fast Collision Detection for Large-Scale Point-Clouds. 2016, 2016.26

[3] Gino Van Den Bergen. Collision Detection in Interactive 3D Environments [M]. Morgan Kaufmann Publishers Inc, 2004

[4] Yan S C, Zhou J Z, Zhu H J, et al. Summary of Collision Detection Algorithms in Virtual Maintenance Training System[J]. Machine Building & Automation, 2012.

[5] He Qunqiang, Chunhua Qian. A Collision Detection Algorithm in Virtual Assembly Technology [J]. Applied Mechanics and Materials, 2015, 3744.

[6] Palmer I, Grimsdale R. Collision detection for animation using sphere-trees [J].Computer Graphics Forum 1995; 14(2):105-16.

[7] Rongguo Zhang, Zhifang Wang, Xiaojun Liu, Kun Liu. Objects Collision Detection in Virtual Scene [J]. Advanced Materials Research, 2013, 2109.

[8] Palmer, I.J. and Grimsdale, R.L. Collision Detection for Animation using Sphere Trees. Computer Graphics Forum[C]. Computer Graphics Forum. Blackwell Science Ltd, 14: 105-116.

[9] Smith A, Kitamura Y, Takemura H, et al. A Simple and Efficient Method for Accurate Collision Detection Among Deformable Polyhedral Objects in Arbitrary Motion [C]. Proceedings Virtual Reality Annual International Symposium '95. IEEE, 2002.

[10] Gottschalk, Stefan & Lin, Ming & Manocha, Dinesh. OBBTree: A Hierarchical Structure for Rapid Interference Detection [J]. 171-180.

[11] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral and K. Zikan, "Efficient collision detection using bounding volume hierarchies of k-DOPs," in IEEE Transactions on Visualization and Computer Graphics, vol. 4, no. 1, pp. 21-36.

[12] Xue Jing Ding. Research on Collision Detection Algorithm Based on Combined Bounding Box [J]. Advanced Materials Research, 2014, 3137.

[13] Huaiyu Wang, Shugui Liu. A Collision Detection Algorithm Using AABB and Octree Space Division [J]. Advanced Materials Research, 2014, 3326.

[14] Xuejing Ding. Research on Collision Detection Algorithm Based on OBB [J]. Applied Mechanics and Materials, 2013, 2755.

[15] Mingquan Wang, Wei Zhao, Huiyan Qu. An Improved Collision Detection Algorithm Based on GPU [J]. Applied Mechanics and Materials, 2014, 3634.