TELKOMNIKA, Vol.14, No.3, September 2016, pp. 1077~1082
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v14i3.3590

Hybrid Hierarchical Collision Detection Based on Data Reuse

Jiancai Hu, Kejing He*, Xiaobin Lin, Funan Lin
School of Computer Science and Engineering, South China University of Technology,
Guangzhou 510641, Guangdong, China
*Corresponding author, e-mail: kjhe@scut.edu.cn

Abstract

To improve the efficiency of collision detection between rigid bodies in complex scenes, this paper proposes a method based on hybrid bounding volume hierarchies for collision detection. In order to improve the simulation performance, the method is based on weighted oriented bounding box and makes dense sampling on the convex hulls of the geometric models. The hierarchical bounding volume tree is composed of many layers. The uppermost layer adopts a cubic bounding box, while lower layers employ weighted oriented bounding box. In the meantime, the data of weighted oriented bounding box is reused for triangle intersection check. We test the method using two scenes. The first scene contains two Buddha models with totally 361,690 triangle facets. The second scene is composed of 200 models with totally 115, 200 triangle facets. The experiments verify the effectiveness of the proposed method.

Keywords: *collision detection, hierarchical structure, data reuse*

1. Introduction

Collision detection is widely used in computer games, virtual surgery, physical simulation, robotics and so on [1, 2]. It plays an extremely important role in these fields. The purposes of collision detection are to detect whether the collision between objects occurs or not, and to know when and where the collision occurs. With the advance of virtual reality technology in recent years, the difficulties of the scene simulation also increases. In the same time, data structures and algorithms about collision detection have become more and more complex in dealing with such large data sets, especially for real-time calculation. Research on the collision detection method has long history, and many researchers have researched deeply and put forward a series of efficient algorithms based on bounding box. Bounding box tightly contains the object but with simple geometric characteristics, to approximately describe the object. Before conducting the collision detection among real objects, the intersection of bounding boxes is checked firstly. If the intersection happens, a further collision detection is needed.

There are some common types of bounding boxes such as Axis-Aligned Bounding Box (AABB) [3], Oriented Bounding Box (OBB) [4], K-DOPs [5], and Spheres [6]. Different types of bounding boxes have different focus, for example the simplicity and tightness. The simplicity and tightness of bounding box are often contradictory. Recently, most researches mainly focus on improving the efficiency and accuracy of collision detection algorithms. Figueiredo [7] uses the overlapping multi-axis bounding box, which can filter disjoint objects fast and improve the efficiency of collision detection. Maciel [8] employs the sphere for fast collision detection among objects. Lai [9] uses sphere and cylinder to represent objects for quick navigation in the scene, which is fast but not accurate enough. Chang [10] combines the sphere with OBB to improve the speed of collision detection, but because of the limitations of the sphere, the accuracy of detection is not enough. Single type of bounding box has some deficiencies and accuracy problems in real-time collision detection [11], therefore more and more researchers propose many algorithms combining the advantages of different bounding boxes. Among them, Bounding Volume Hierarchies (BVH) [12-14] is one of the most widely used, which can work in complex environments. Arbabi [15] uses cylindrical and radial space segmentation method to perform collision detection for joint connection. This method has a better precision with limited usage in collision detection of boundary movement. In recent years, many researchers use hardware acceleration to speed up the collision detection with the help of better computer

Received April 2, 2016; Revised June 12, 2016; Accepted June 29, 2016

performance [16-18]. Xie [19] combines hierarchical bounding sphere with GPU acceleration to simulate collision detection among rigid bodies for rhinoplasty. Shen [20] adopts the mixed bounding volume hierarchy tree to detect the potential collision object sets quickly, and then uses the streaming pattern algorithm for accurate collision detection. But these approaches have not taken the advantages of different types of bounding boxes.

In this paper, we propose an improved hybrid hierarchical collision detection method based on data reuse. We combine the simplicity of improved AABB and the tightness of OBB to build hybrid BVH structure for objects. Our improved hybrid hierarchical structure has two layers. The uppermost layer uses optimized cubic bounding box, which is easy to rotate and transform. By this way, the objects are not intersected in the scene can be excluded quickly. The lower layers use the OBB, which has better tightness and can be used for further collision detection for those intersected objects found in the upper layer. The method can improve the efficiency and accuracy of collision detection by taking the advantages of the hybrid hierarchical bounding structures. In the meantime, we introduce a new algorithm for triangle intersection check by reusing the OBB data [21], which can effectively reduce the calculation and further improve the efficiency.

This paper is organized as follows. Following the introduction in Sec. 1, Sec. 2 presents the improved hybrid hierarchical collision detection method based on data reuse. In Sec. 3, the method is applied to the collision detection of two models and multiple models respectively. We also compare the performance of our method with other approaches. Finally, some conclusions and discussions are made in Sec. 4.

2. Research Method

The collision detection among AABB is simple and fast, which can be accomplished within six tests. When the object moves, AABB has to be rebuilt. A given object's OBB is the smallest cuboid that contains the object. Compared to AABB, OBB has better tightness, but the intersection check is more expensive.

The most used method to determine whether two convex hulls are intersected or not is to use the separating axis test [22]. The number of separating axis depends on the type of bounding box. For example, AABB has 3 (x , y , z) typical separating axes. OBB has 15 typical separating axes, so the intersection check of two OBBs could take up to 15 comparison operations, 60 addition operations, 81 multiplication operations, and 24 absolute value operations [23].

Traditional hybrid BVH just simply uses two different bounding boxes to enclose every object for better compactness and fast excluding disjoint objects [24]. When the objects are close but not collide, they can be separated simply because of bounding box's tightness, thus the performance of hybrid BVH is better than that of single bounding box. However, when two objects are intersected, the intersection of bounding box structures is going to be checked multiply times, which brings redundant calculations. Thus it is unable to fulfill the requirements of real-time collision detection. This paper provides an efficient hybrid hierarchical collision detection method based on data reuse. We integrate the cubic bounding box and weighted OBB to form a hybrid hierarchical structure in the pretreatment phase. Then we introduce the triangle intersection check algorithm by reusing the OBB data, which can effectively reduce the calculation time and improve the performance of collision detection.

2.1. From AABB to Cubic Bounding Box

To reduce the memory consumption and to speed up the calculation, we use cubic bounding box, which is a special kind of AABB. The cubic bounding box has the same half-width extent (or radius r) in three axes, and the center-radius representation is adopted. We suppose that the model is composed of N triangles, and $\vec{o}_i$, $\vec{p}_i$, $\vec{q}_i$ are the vertexes of triangle i , and the center point of cubic bounding box is:

$$\vec{C} = \frac{1}{3N}\sum_{i=1}^{N}(\vec{o}_i + \vec{p}_i + \vec{q}_i) \quad . \tag{1}$$

The half-width extent or radius is:

$$r = \max_i \left(\max \left(dist(\vec{o}_i, \vec{C}), dist(\vec{p}_i, \vec{C}), dist(\vec{q}_i, \vec{C}) \right) \right) , \tag{2}$$

Where $dist(\cdot,\cdot)$ is the function to calculate the distance between two points. Since the cubic bounding box has the same radius in three axes, therefore, the object can move or rotate without changing the radius. This method not only reduces the storage usage, but also accelerates the bounding box updating. The intersection check of cubic bounding box is simpler than sphere's. Suppose that (x, y, z) is the center point of cubic bounding box, then the minimum coordinate ($x-r$, $y-r$, $z-r$) and the maximum coordinate ($x+r$, $y+r$, $z+r$) can be gotten quickly. It can achieve intersection check within six tests.

In order to improve the efficiency, the spatial and temporal correlation is introduced. We use three lists to save the projective coordinates of objects in three (x, y, z) axes respectively. When the object's state changes, the method updates the three lists and quickly finds the bounding box pairs that intersect in the projection, and puts those pairs into a global dictionary. It will get object pairs out from the global dictionary when further intersection check is required. A traversal algorithm is used for detecting the collisions between sibling subtrees. We build a new list for each object to record the neighbors those collide with the object. We deduplicate the list to exclude redundant collisions. When two objects collide, the object with larger ID number will be stored in the list of the object with smaller ID number. If the list is empty, there was no collision with the object. According to the above analysis, our method takes advantages of the spatial and temporal correlation of object's movement, and makes it not necessary to traverse from the tree root, avoiding useless calculation, so speeds up the collision detection process.

2.2. Weighted Oriented Bounding Box

Traditional method for computing the center point of OBB is to get the mean position of all vertexes. But in practice, the sizes of the triangles that constitute the object are nonuniform, so using the traditional method will make the calculated center point tend to crowded triangle facets. This paper proposes weighted oriented bounding box. We make a dense sampling for the points of convex hull surfaces in order to reduce the impact of crowded triangle facets. In the i-th triangular facet of convex hull, the center point is:

$$\vec{c}_i = \frac{\vec{o}_i + \vec{p}_i + \vec{q}_i}{3} , \tag{3}$$

And the surface area is:

$$S_i = \frac{\left|(\vec{o}_i - \vec{p}_i) \times (\vec{o}_i - \vec{q}_i)\right|}{2} . \tag{4}$$

The total area of the convex hull is:

$$W = \sum_{i=1}^{n} S_i . \tag{5}$$

The center of the bounding box is:

$$\vec{C} = \frac{1}{n} \sum_{i=1}^{n} (S_i \cdot \vec{c}_i) \cdot \tag{6}$$

Let j and k be the component of (x, y, z), and according to the above analysis, the covariance $Cov_{j,k}$ is:

$$Cov_{j,k} = \sum_{i=1}^{n} \frac{S_i}{12W} \left(9c_{i,j}c_{i,k} + o_{i,j}o_{i,k} + p_{i,j}p_{i,k} + q_{i,j}q_{i,k} \right) - C_j C_k . \tag{7}$$

2.3. Hybrid Hierarchical Bounding Volume Tree

In this paper, we create hybrid bounding box tree for objects in the virtual environment. The tree is composed of several layers. The uppermost layer uses the cubic bounding box (Sec. 2.1), and the lower layers use weighted OBB (Sec. 2.2) with better tightness. This algorithm not only can exclude nonintersecting objects in complex scenes quickly, but also can do accurate collision detection. It has better real-time detection efficiency and tightness. Even in the worst case, there is only one intersection check for every bounding box in each layer, which fully reflects the advantages of the hybrid bounding box structure.

Compared with RAPID algorithm [4], which is a widely-used bounding volume hierarchy tree implementation based on OBB, the improved hybrid BVH has several advantages:

1. Reduce the amount of memory usage. Because cubic bounding box requires 4 floats, and OBB only needs 15 floats, the amount of memory that used to store the bounding boxes are reduced.
2. Simple intersection check. In this algorithm, the root node uses cubic bounding box instead of OBB. The intersection checks among cubic bounding boxes need fewer calculation operations and are simpler.
3. Reduce collision detection time. In real-time detection process, the cubic bounding box does not need updates, and can quickly exclude most nonintersecting objects, thus save a lot of unnecessary intersect checking time.

2.4. Triangle Intersection Check by Reusing the OBB Data

If two bounding boxes are intersecting, it doesn't mean that the two objects really collide, so a further intersection check is required. We use the interval intersection algorithm [25] to check whether triangle facets intersect or not. The detail of this algorithm is as follow:

1. Calculate plane equations of two triangles respectively. If all vertices of one triangle reside in the same side of another triangle, the two triangles do not intersect.
2. If these two planes intersect, figure out the intersecting line L of two planes. Then establish a coordinate system which has an axis paralleling to L.
3. Calculate the projection intervals of the two triangles in the coordinate system.
4. If these two intervals overlap, the two triangles intersect.

In order to improve the efficiency of the above algorithm, the OBB-based triangle intersection check method [21] is adopted. In traditional triangle intersection check, coordinate transformation is necessary, which means that one bounding box or triangle must be represented in another one's coordinate system.

In our method, each leaf node in the hybrid BVH is a rectangle, which contains only one triangle. At leaf nodes, we represent each triangle using the coordinate system of its bounding box. By reusing the coordinate system information, it is easy to calculate the point-to-plane distance. In this way, the redundant calculations in triangle intersection check are reduced. Thereby the efficiency of collision detection is increased.

3. Experiments and Results

To test the performance of the improved method, it is compared with RAPID. The algorithm I is based on the improvement Sec. 2.1 to 2.3. And improved algorithm II takes all the improvements into account.

This algorithm is implemented using OpenGL graphics library and is run on a machine with Intel Core2 Duo T5750 processor, 2 GB DDR2 667MHz memory and NVIDIA GeForce 8400M GS graphics card. We test the method using two scenes. The first scene (Figure 1) contains two Buddhas, each of which has 180,845 triangular facets and moves in a specific trajectory. The second scene (Figure 1) includes 200 models with totally 115,200 triangle facets.

The experimental results are shown in Figure 2. The experiments show that when the number of intersected triangle grows, the collision detection time for all methods increases. In both the two-model scene and the multi-model scene, our improved hybrid hierarchical collision detection method performs better than RAPID. When the number of models is small, the difference between algorithm I and RAPID is slight. But when there are a lot of models, the advantage of improved Algorithm I gradually appears. This is because the intersection check of improved cubic bounding box is faster than OBB. And there is no need to update when the object's state changes. Reusing the OBB data for triangle intersection check and introducing spatio-temporal correlation can reduce redundant calculation and have significant contribution to performance improvement.

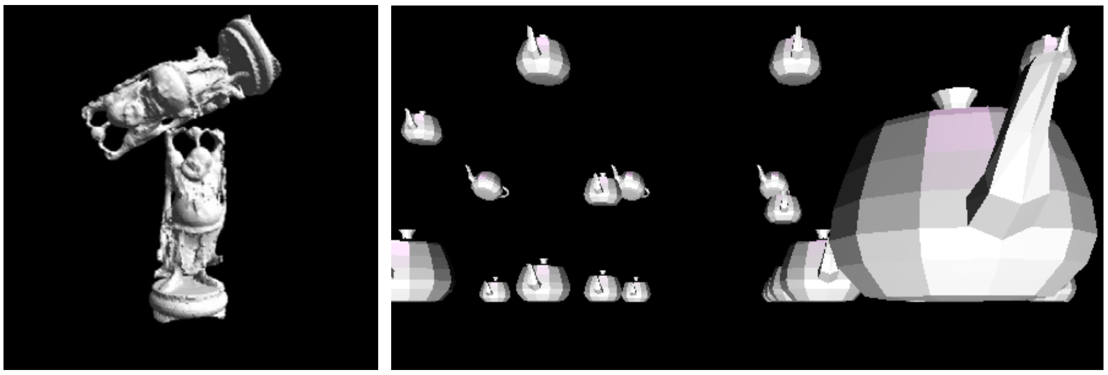

Figure 1. The scenes of Buddhas (left) and teapots (right).

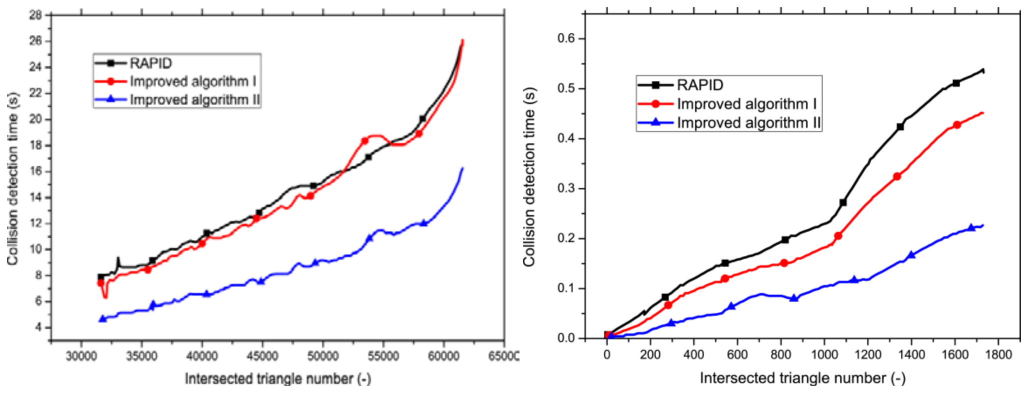

Figure 2. The experimental results of Buddhas (left) and teapots (right).

4. Conclusion

This paper has presented an improved hybrid hierarchical collision detection method that takes the advantages of improved cubic bounding box, weighted OBB, hybrid hierarchical bounding volume tree, and data reusing. The method improves the efficiency of collision detection. Compared with traditional collision detection algorithms, the experimental results have shown that this method has better performance in both two-model scene and multi-model scene.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) (No. 61272200, 10805019), the Program for Excellent Young Teachers in Higher Education of Guangdong, China (No. Yq2013012), the Fundamental Research Funds for the Central Universities (D2153440), the Special Support Program of Guangdong Province, and the Pearl River Science & Technology Star Project.

References

[1] Jimenez P, Thomas F, Torras C. Collision detection: a survey. *Computers and Graphics*. 2001; 25(2): 269-285.

[2] Suaib NM, Bade A, Mohamad D. Hybrid Collision Culling by Bounding Volumes Manipulation in Massive Rigid Body Simulation. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(6): 3115-3122.

[3] Cai PP, Indhumathi C, Cai YY, Zheng JM, Gong Y, Lim TS, Wong P. Collision detection using axis aligned bounding boxes. *Simulations, Serious Games and Their Applications*. 2014: 1-14.

[4] Gottschallk S, Lin MC, Manocha D. *OBBTree*, *A hierarchical structure for rapid interference detection*. Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. 1996: 171-180.

[5] Klosowski JT, Held M, Mitchell JSB, Sowizral H, Zikan K. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Transactions on Visualization and Computer Graphics*. 1998; 4(1): 21-37.

[6] Palmer IJ, Grimsdale RL. Collision detection for animation using sphere-trees. *Computer Graphics Forum*. 1995; 14(2): 105-116.

[7] Figueiredo M, Feenando T. *An efficient parallel collision detection algorithm for virtual prototype environments*. Proceedings of the 10th International Conference on Parallel and Distributed Systems. 2004: 249-256.

[8] Maciel A, Boulie R, Thalmann D. Efficient collision detection within deforming spherical sliding contact. *IEEE Transactions on Visualization and Computer Graphics*. 2007; 13(3): 518-529.

[9] Lai KC, Kang SC. Collision detection strategies for virtual construction simulation. *Automation in Construction*. 2009; 18(6): 724-736.

[10] Chang JW, Wang WP, Kim MS. Efficient collision detection using a dual OBB-sphere bounding volume hierarchy. *Computer Aided Design*. 2010; 42(1): 50-57.

[11] Christer E. Real-time collision detection. Morgan Kaufmann Publishers Inc. 2005.

[12] Bade A, Ping CS, Tanalol SH. Collision detection for cloth simulation using bounding sphere hierarchy. *Jurnal Teknologi*. 2014; 75(2): 1-5.

[13] Schwesinger U, Siegwart R, Furgale P. *Fast collision detection through bounding volume hierarchies in workspace-time space for sampling-based motion planners*. Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA). 2015: 63-68.

[14] Wu HY, Shu ZM, Liu YG. Study based on hybrid bounding volume hierarchy for collision detection in the virtual manipulator. *Applied Mechanics and Materials*. 2013; 454: 74-77.

[15] Arbabi E, Boulic R, Thalmann D. Fast collision detection methods for joint surfaces. *Journal of Biomechanics*. 2009; 42(2): 91-99.

[16] Guo AB, Wang QZ, Li XL. *Research on collision detection algorithm of tankin virtual battlefield*. Proceedings of International Conference on Automation. 2015: 1944-1949.

[17] Du P, Zhao JY, Pan WB, Wang YG. GPU accelerated real-time collision handling in virtual disassembly. *Journal of Computer Science and Technology*. 2015; 30(3): 511-518.

[18] Tang M, Manocha D, Tong RF. MCCD: multi-core collision detection between deformable models using front-based decomposition. *Graphical Models*. 2010; 72(2): 7-23.

[19] Xie K, Yang J, Zhu YM. Fast collision detection based on nose augmentation virtual surgery. *Computer Methods and Programs in Biomedicine*. 2007; 88(1): 1-7.

[20] Shen XL, Wu Q, Cheng YW. Hybrid Collision Detection Algorithm based on Image Space. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(12): 7159-7165.

[21] Chang JW, Kim MS. Efficient triangle-triangle intersection test for OBB-based collision detection. *Computers and Graphics*. 2009; 33(3): 235-240.

[22] Spuy R. Advanced Game Design with Flash. Apress. 2010: 224-236.

[23] Man RR, Zhou DS, Zhang Q. An improved collision detection algorithm based on OBB. *Computer Modelling and New Technologies*. 2014; 18(1): 71-79.

[24] Hahn JK. Realistic animation of rigid bodies. *Computer Graphics*. 1988; 22(4): 299-308.

[25] Moller T. A fast triangle-triangle intersection test. *Journal of Graphics Tools*. 1997; 2(2): 25-30.