Hybrid Collision Detection Algorithm based on Image Space

XueLi Shen*[1,2], Qiong Wu[2], Yuwei Cheng[2]
[1]School of Information and Electrical Engineering, CUMT, XuZhou JiangSu China
[2]School of Electronics and Information Engineering, Liaoning Technical University, Huludao LiaoNing, China
*Corresponding author, e-mail: shenxueli@sina.com

Abstract

Collision detection is an important application in the field of virtual reality, and efficiently completing collision detection has become the research focus. For the poorly real-time defect of collision detection, this paper has presented an algorithm based on the hybrid collision detection, detecting the potential collision object sets quickly with the mixed bounding volume hierarchy tree, and then using the streaming pattern collision detection algorithm to make an accurate detection. With the above methods, it can achieve the purpose of balancing load of the CPU and GPU and speeding up the detection rate. The experimental results show that compared with the classic Rapid algorithm, this algorithm can effectively improve the efficiency of collision detection.

Keywords: *component, virtual reality, bounding volume hierarchy, hybrid collision detection, streaming pattern collision detection*

1. Introduction

With the development of computer technology, virtual reality [1-2] has been widely applied to computer animation, physical simulation, the robot technology, virtual prototype design and so on, and collision detection is a key research field of virtual reality technology. Collision detection is based on a fact in real world, namely [3]: it is impossible that two nonpenetrating objects to share the same space area at the same time. The object model of collision detection algorithm can be roughly divided into two sections [4]: those based on the geometrical space and those based on image space .

At present, the collision detection algorithm the most widely used is the algorithm of Bounding Volume Hierarchy [5-6]. The algorithm of Bounding Volume Hierarchy can greatly improve the rate and guarantee accuracy when the collision detection is processing. But it can increase the CPU's load and waste the resources of graphics hardware, because the test process of the algorithm named Bounding Volume Hierarchy is on the CPU. With the continuous development of graphics hardware, many methods of collision detection based on GPU have appeared and have obtained some research results. The algorithm based on GPU can make full use of the powerful computation ability of GPU and reduce the computation load of CPU , finally achieves the purpose of improving the efficiency of collision detection. The collision detection algorithm based on GPU [7] usually projects 3D virtual scene to image plane through GPU, reducing the dimension to the 2D image space, and then detects whether the objects are intersecting or not through querying and analyzing all kinds cache informations.

However, there is shortcomings if the collision detection only relies on GPU, because this algorithm directly aims at the attributes of the object (not processed, the model of geometry properties directly to test), and causes much unnecessary and redundant detection, finally affecting the efficiency of collision detection. Considering that the collision detection algorithm of Bounding Volume Hierarchy can solve this problem, it is meaningful to study combining the algorithm of Bounding Volume Hierarchy and the computation techniques about GPU [7].

In this paper, an algorithm of Hybrid Collision Detection based on Bounding Volume Hierarchy is proposed to conduct the collision detection of flow model at the end of collision

Received June 29, 2013; Revised August 17, 2013; Accepted August 28, 2013

detection, which can finally improve the speed of collision detection between the objects and enhance the authenticity and real-time of the virtual scene.

2. Algorithm Description

The hybrid collision detection algorithm based on image space mainly researches the hybrid bounding volume hierarchy and streaming pattern collision detection based on graphics hardware. At first, it constructs Top Bounding Box of the object, and quickly takes the way of scanning-clip [5] to remove object which could intersect impossibility, then finds the potential impact areas. According to the potential impact area it builds the child of Bounding Box Hierarchical Tree, and then gets the potential impact areas through traversing the tree. Basic flow of algorithm is shown in Figure 1:

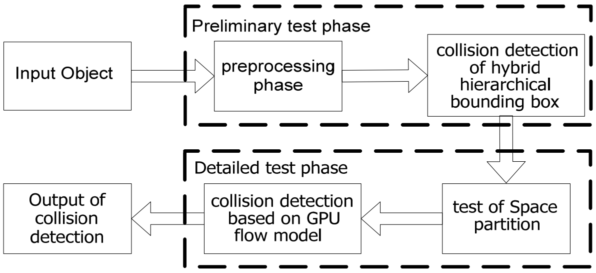

Preliminary test phase
Input Object
preprocessing phase
collision detection of hybrid hierarchical bounding box
Detailed test phase
Output of collision detection
collision detection based on GPU flow model
test of Space partition

Figure 1. Flow Chart of Collision Detection

At first, the objects in the virtual scene are rapidly sifted through the algorithm of Hybrid Collision Detection which is based on bounding box that can eliminate the objects that may impossibly intersect. Then the objects which may intersect each other are tested with flow pattern collision detection based on GPU. The collision detection algorithm presented by this paper can balance the load of the CPU and GPU and speed up the detection rate.

3. Implementation of Algorithm

3.1. Selection of the Bounding Box

The main purpose of building the bounding volume hierarchy is to quickly eliminate the object of collision which may not happen. The bounding volume on the top is the beginning of the traversal detection and the detecting time spent on it is more than that at the other steps, so it should be relatively simple to build, update and test the bounding volume. When constructing the child of bounding volume in potential collision set, it is needed to choose the accurate and close bounding volume to test.

Cost functions [8] are often used to evaluate the performance of the collision detection algorithm based on bounding volume. The definition of a cost function is shown as formula (1):

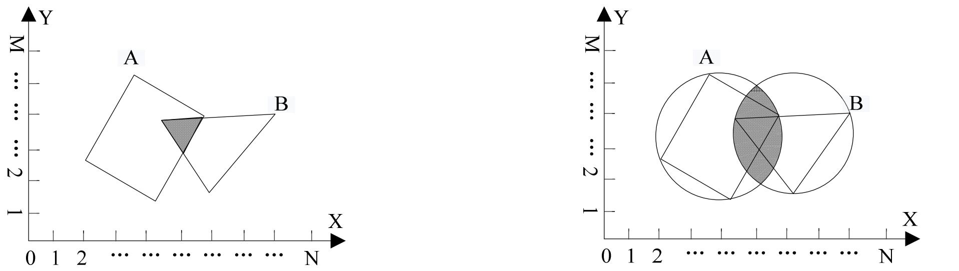
$$T = N_V \times C_V + N_P \times C_P + N_U \times C_U \tag{1}$$

T represents the total cost of collision detection, N_V is the logarithm of bounding volume which participates in the overlap test; C_V is the cost of a couple bounding volumes which make the overlap test; N_P is the logarithm of the geometry element test which make intersecting test; C_P is the cost of a couple geometry elements which make intersecting test; N_U is the number of the node in the bounding volumes which need to be modified after the motion of object; C_U is the cost of amending a node. According to the needs of algorithm and formula (1), the bounding volumes on the top use the simple Sphere to build and test; and close oriented bounding box (OBB) is made use of to construct the child of hierarchical bounding volume.

3.2. Determination of Potential Collision Set

Most of the tests of intersection of object in the virtual scene are almost converted to 2D space to deal with, which can improve the efficiency of the processing and solve the overlapping testing question of Bounding Sphere through transforming 3D virtual scene to 2D. At first, the objects should be projected to the coordinates of X, Y, Z, and then the borders which belong to projection of each Bounding Sphere are sorted.

The algorithm adopts the method which converts to 2D space to deal with the problem of overlapping test of Sphere. The concrete method is that, getting the projection of object form three coordinates of X, Y, Z, and sorting the border of every Sphere's projected zone. According to the projected zone of two objects it is checked whether there are overlapping regions or not, if there is overlapping area.The two objects may collide and potential collision set through the overlapping zone may be found as shown in Figure 2:



(1) (Object A) ∩ (Object B)

(2) Intersection of Object Bounding Volume

Figure 2. Determination of the Overlap Area in Bounding Volume

3.3. Build and Traverse the Structure of Child Hierarchical Bounding Volume

There are three ways to build the bounding volumeincluding bottom-up, top-down and incremental tree-insertion. The top-down aims at the top the bounding volume, and divides up the level with the recursive properties of set until all leaves a node of the bounding volume cannot be divided. At present, the construction of top-down in the actual application is widely used and the method is also relatively mature, so we adopt this method to build OBB tree structure. In this paper, it constructs OBB bounding box through the algorithm which proposed by Gottschalk [9-10]. The specific steps are as follows:

(1) To set t the vertex vector of triangle as p^i、q^i、r^i, the number of triangle facets as n which belongs to the bounding box, the mean of distribution on the vertex as U .

$$\mu = \frac{1}{3n}\sum_{i=0}^{n}(p^i + q^i + r^i) \tag{2}$$

(2) To Attain the covariance matrix named C through the distribution mean of above vertex:

$$C_{jk} = \frac{1}{3n}\sum_{i=1}^{n}(\overline{p_j^i p_k^i} + \overline{q_j^i q_k^i} + \overline{r_j^i r_k^i})(1 \le j,k \le 3) \tag{3}$$

In addition, $\overline{p_j^i} = p_j^i - \mu$、$\overline{q_j^i} = q_j^i - \mu$、$\overline{r_j^i} = r_j^i - \mu$, C_{jk} is the element of 3×3 covariance matrix.

(3) To compute and get the characteristic vector of matrix, then determine the three axes of local coordinates which belong to OBB bounding box. Because of C is symmetric matrix and its characteristic vector is orthogonal each other, after making the three features vector to unitization and then setting them as the three axial named (d_o, d_1, d_2) of the local coordinate belongs to OBB bounding box.

(4) To project all vertices of the triangles to three axes named (d_o, d_1, d_2). The minimum and maximum distance difference of the three axes is the size of bounding box named OBB.

$$\begin{aligned} u^0 &= \max(\Pr oject(d^0, v^i)); u^1 = \max(\Pr oject(d^1, v^i)); u^2 = \max(\Pr oject(d^2, v^i)); \\ w^0 &= \min(\Pr oject(d^0, v^i)); w^1 = \min(\Pr oject(d^1, v^i)); w^2 = \min(\Pr oject(d^2, v^i)) \end{aligned} \quad (4)$$

(5) To compute the center of bounding box named OBB cen .

$$cen = \frac{1}{2}(u^0 + w^0)d^0 + \frac{1}{2}(u^1 + w^1)d^1 + \frac{1}{2}(u^2 + w^2)d^2 \quad (5)$$

Finally, the size of OBB can be determined through projecting all fixed point of objects to the three coordinate axes and calculating maximum and minimum of the three coordinate axes. Taking Figure 2 as an example, it can be found that the overlapping area of objects named A and B may possibly collide each other through the above testing. So the algorithm can construct the OBB level tree of child which belongs to the object through aiming at the reverse projection of potential object set which belongs to the object named A and B.

In order to make the detailed Collision Detection more meaningful, the algorithm avoids the time spent in constructing and traversing the tree too long and guarantees that the depth of divisiory hierarchical tree is as low as possible. The size of OBB hierarchical tree's degree impacts the traversing detection rate, so it should be ensured that traversing path is short and the height of trees is low. Because the binary tree has the characteristics of fast traversing, simple computing and the low time of searching the node of binary tree, the algorithm chooses the binary tree. The detection of the intersecting situation of two bounding volume is accomplished through traversing two binary treeswhich is a dual traversing process of tree structure: judging whether the roots of two box trees is intersect or not, then judging the situations of their child nodes if they are intersecting. When there are some intersecting leaf nodes, it needs to make detailed collision detection. Specific algorithm of collision detection is as follows: Supposing W_a and W_b are the detecting nodes of the object bounding box, Q_a and Q_b are the corresponding node sets of W_a and W_b as shown in Figure 3:

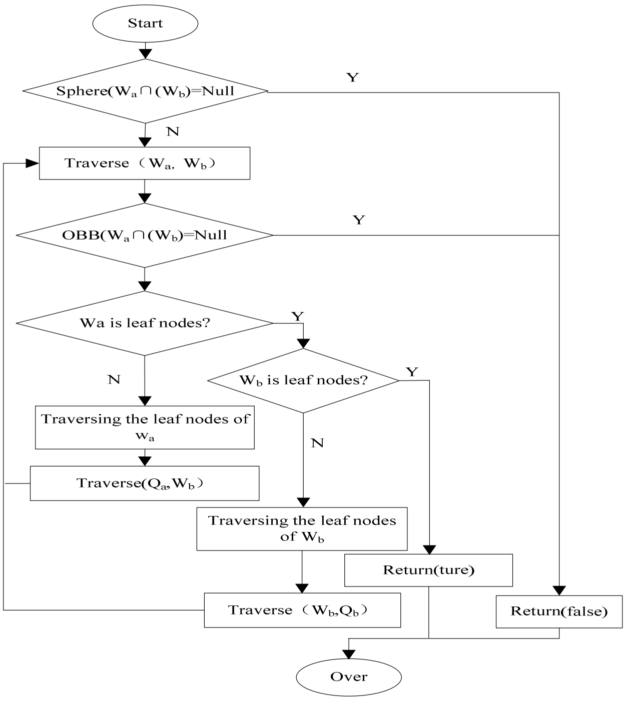


Figure 3. Flow of Collision Detection

3.4. Detailed Collision Detection

According to the detection of above two stages, find the more accurate intersection area can be found. For the accurate results of collision detection, this stage needs to test the graphic elements in the intersection area. The methods of intersecting detection between the basic graphic elements mainly are of the following kinds [7] :

(1) Intersection test between the edge and triangle;
(2) Intersection test between the edge and polygon;
(3) Intersection test between the plane and plane;
(4) Intersection test between triangle and triangle.

It is called Streaming Pattern Collision Detection [10] to transform the process of intersecting detection between the elements to the Processing mode of GPU and to make full use of the programmable of GPU. The algorithm can ensure the intersection situation between the objects by testing the intersection between all sides of the surface which belong to the objects and all triangular. This process mainly includes three stages: the generation of edge texture, intersection testing of edge and triangle, test results, and the specific test process (see references [11]).

3.5. Optimization method

The stage of detailed collision detection mainly makes use of the flow calculation model [11] based on graphics hardware to take collision detection, and the key of flow calculation is mapping the calculation of collision detection mapping to calculation of flow model. The collision detection based on the flow pattern is a kind of calculation which turns the test process from computing to graphics hardware, graphics hardware when drawing detects the intersectant situation of sides and triangles, and the efficiency of the algorithm is decided by the number of side and triangle. To improve the efficiency of the algorithm, the numbers of the edges and triangle should be reduced in input stream. This algorithm adopts the space partition strategies, and it makes use of the space partition to divide and test the intersectant regional of the double objects before testing with the flow pattern detection, reducing a lot of basic geometrical elements, get more accurate intersectant situation, reduce the sides and triangles in the input stream and improve the detection efficiency.

The main idea of Space-partitioning [5] is that the scene of the object is divided into small grids by some rules with the benchmark of X, Y, Z axis, and the characteristics of the same or adjacent grid are detected to judge whether the objects happen to collide or not. Because of the following step needs the accurate results to reduce the number of input data flow, the paper adopts the method of uniform space partition whose collision detection accuracy is higher. Uniform space partition divides the detecting area into the same size of the grids, then maps the object which will be detected to grid and traverses the grid for testing. The key problem is to ensure the size of space partition when making the space partition, which will impact the accuracy and efficiency of the testing if division is not suitable. The result of experiments has proved that, the best size of grids is the same as bounding volume of a single element of object, and the calculation efficiency is the highest. So the algorithm adopts the specifications.

Supposing that the overlapping area of bounding box leaves node of object A and B may happen to crash, so it only need to make division just in the overlapping area of bounding box. Dividing it into uniform grid of $n \times m$, as shown in Figure 4. The cell is traversed and a side is obtained where ΔADS and ΔFGH exists in the same gird, then the model of flow is tested.

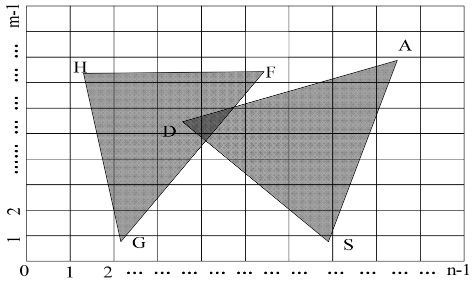


Figure 4. Uniform Gird Division

When the space division is finished, the test results are finally got through making accurate collision detection of flow pattern with the elements which can not be eliminated. In this paper, the method is adopted that tests all sides of an object with all triangles of another object to determine whether all the intersection of the collision or not. This process mainly includes three stages [7]: texture generation, collision inquires and obtaining the result.

4. Results and Analysis

The experimental environment is as below: CPU Intel Core 2 Quad Q8200, RAM 6GB, Video Card RADEON HD 5750, and completing the experiment about colliding between two objects with the language of C++. For verifying the effectiveness of the algorithm, we contrast with the detection time of classic Rapid algorithm and our algorithm in the case of different modes of triangular diagram.

In order to verify the effectiveness of the proposed algorithm, it takes the experiments under the same scene and model with different collision detection algorithm to test the model which mutual movement, compute the average testing time of collision experiment and compare the collision detection effects of different algorithms. The classic algorithm named Rapid which based on OBB bounding box can effectively solve the collision detection problem of one active object in the large-scale and complex model and it is as the standard algorithm to estimate other collision detection algorithms, but it still exist some problems with CPU load and low utilization of the resource about graphics hardware. The algorithm which based on Hybrid Collision Detection Algorithm based on Image Space improve the problems of Rapid. So, it emphatically compares rates of Hybrid Collision Detection Algorithm based on Image Space and Rapid in this paper, and the main comparative content is that: first, it takes the collision detection experiment with the algorithm of Rapid and Hybrid Collision Detection Algorithm based on Image Space; then compare the experiment results of the two algorithm, the specific process is that: set a threshold which on behalf of test times, and set another static variable to record the times of collision detection, it obtains the cost of time through the test times and times which recorded, the shorter the time, the better the performance.

$$T = \frac{t}{C} \tag{6}$$

The t is the cost of time, T is the average detect time, C is the test times. It can effectively compare the two collision detection algorithms through the three parameters t , T and C .

Table 1. Detection Time of our Algorithm and Classic Rapid Algorithm

Number of triangular diagram	Rapid/ms	this algorithm/ms
968	4.1620	3.9716
3162	7.6131	6.5753
5201	11.0231	9.5957
7591	14.4713	11.7126
9613	18.3837	13.2147

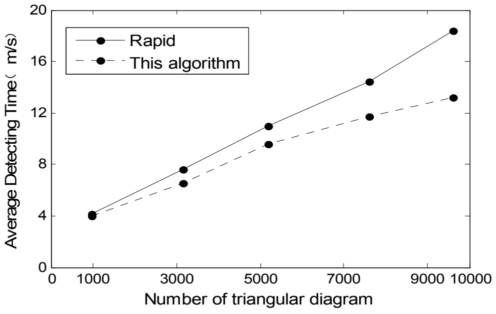


Figure 5. Contrastive Chart of Average Detecting Time

As can be seen from the Figure 5, the detecting time of the algorithm named Rapid is quickly increase with the growth of Modelling Primitives, but the algorithm's which presented by this paper is increasing more slowly than Rapid's. The reason is that, Rapid makes the load of CPU overweight and wastes the resource of graphics hardware. The algorithm which presented by this paper improve those shortcomings belonged to Rapid, the average detecting rate of algorithm which presented by this paper is higher than Rapid's with the same detecting objects. As can be seen from the above argument, it shows that the Hybrid Collision Detection Algorithm based on Image Space can make full use of the computing power which belonged to the CPU and GPU, and it effectively overcomes the shortcomings of low efficiency and redundant computing. The algorithm can get the collision detection result of models with keeping the accuracy of collision detection. The experimental results can also be shown that, the Hybrid Collision Detection Algorithm based on Image Space can make the space division to get the optimization in the stage of collision detection and reduce the redundant computing of GPU, it can get the aim of improving the effective availability.

5. Conclusion

This paper presents hybrid collision detection algorithm based on bounding volume hierarchy of geometric space and Graphics Hardware. At the beginning, the algorithm can eliminate the non-intersecting object with the Bounding Volume Hierarchy; after optimizing when the objects of intersection which may intersect each other make the space division, it can make an accurately intersecting test with the Streaming Pattern Collision Detection based on Graphics Hardware. The algorithm combines with the advantages of classical algorithm, so it can balance the loads between the CPU and GPU. Compared with the collision detection algorithm based on the geometric space, the algorithm has certain limitation which lacks the precision of collision detection. So it looks forward to finding the better method.

References

[1] Christer Ericson. Real-Time Collision Detection. Beijing: Tsinghua University Press. 2010.

[2] Hong Bingrong, Cai Zesu. Virtual Reality and Application. Beijing: *Defense industry press.* 2005.

[3] Baciu G, Wong S. Image-based techniques in a hybrid collision detector. *Visualization and Computer Graphics.* 2002; 1(3): 17-23.

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

[5] Liu Zhilia. Research on Collision Detection Algorithm based on Bounding Box and Space Division. Wu Han: *Central China Normal University.* 2009.

[6] Palmer I, Grimsdale R, Collision Detection for Animation using Sphere-trees. *Computer Graphics Forum*. 1995; 14(2): 105-116.

[7] Zhuang Hua, Image Space Collision Detection. Qing Dao: *China University of Petroleum.* 2007.

[8] Dai Gaole. Research of Collision Detection Algorithm based on Bounding Box. Luo Yang: *Henan University of Science and Technology.* 2009.

[9] Wang Peng, Liu XuMin, Guan Yong. Improved collision detection algorithm based on OBB. *Computer Engineering and Design.* 2009; 30(13).

[10] Gottschalk S. Collision Queries using Oriented Bounding Box. *University of N. Carolina.* 2000: 41-47.

[11] Fan Zhaowei, Wang Hua Gen, Gao Shu Ming. Streaming Real Time Collision Detection Using Programmable Graphics Hardware. *Journal of Software.* 2004; 15(10): 1505-1514.