2022 21st International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

Research on 3D Simulation Technology of Multi-fragment Damage Assessment Based on Distributed Computing

Maocai Yuan, Kun Li,Yujiao Fan,Hao Zheng,Yuchong Xu
Wuxi Orient Software Technology Co., Ltd.
Wuxi,China
Eamil: yuanmc@cssrc.com.cn

***Abstract*—Aiming at the lack of multi-fragment collision detection for complex targets in damage assessment simulation, a technical solution is proposed to use open source scene graphics OSG and Bullet physics engine to combine distributed computing to solve collision detection of different fragments. A self-developed bounding box technology is proposed to construct the bounding box of multi-fragment damage elements, and the axial bounding box technology is selected to construct the bounding box of complex targets. Based on the traditional collision algorithm, a collision detection algorithm relying on distributed computing is proposed, which realizes the collision detection between tens of thousands of fragments and the target, and visualizes the damage effect after the fragments collide with the target. The simulation results show that the optimized solution can successfully achieve the target collision detection and damage assessment under the action of multiple fragments, which effectively improves the calculation efficiency and has a good rendering effect.**

Keywords- Open Source Scene Graph (OSG); Bullet Physics Engine; Distributed Computing; Collision Detection; Damage Assessment;

I. INTRODUCTION

At present, for the damage evaluation of complex targets with multi-fragment damage elements, three methods are mainly adopted: traditional theoretical calculation, numerical simulation and experiment [1]. Traditional theoretical calculation methods are based on empirical models and mainly use theoretical formulas to solve. Compared with the traditional theoretical calculation method, the numerical simulation can visualize the damage process, and the damage form of the warhead is relatively clear. The test method is more accurate, but requires a lot of time and cost, and is not economical. With the rapid development of computer technology, numerical simulation solving damage assessment has gradually become the mainstream. However, the calculation method of finite element simulation can only obtain the flying speed of a single fragment at different times, and cannot obtain the data of the flying angle and decay speed of each fragment completely. At the same time, the general finite element post-processing software will lead to a long time period when calculating large-scale damage scenarios, which is very unfavorable for the study of ammunition power and target damage [2].

In response to this problem, this paper proposes a technical solution for the combination of the 3D rendering engine OSG and the physical collision engine Bullet, and distributed computing to solve the collision detection of different fragments [3]. This solution can realize the visualization of multiple fragments and the detection of complex targets with multiple fragments. Damage assessment. Based on the calculation results of the theoretical formula, combined with the shooting trace method [4], according to the fragment scattering angle and flight speed attenuation, OSG technology is used to render the warhead power field, and the visualization of the fragment scattering field is realized. The penetrating damage of the fragment to the target is calculated by the THOR penetration equation [5~7], and the Bullet physics engine is used to detect the collision between each fragment and the target. Distributed computing is used to optimize the solution efficiency, determine the damage degree of the target, and use OSG to render different colors to represent different damage degrees.

II. FUNCTIONAL REQUIREMENTS ANALYSIS

For the warhead, the main damage elements are fragments and shock waves. Due to the short action time of the shock wave, fragments become the main means of damage to kill the target. The traditional 3D scene visualization can be realized by using the OSG 3D engine, including different system scenes and explosion scenes, as well as the visualization of projectiles and targets. For the fragments formed by the warhead, OSG can be realized by simulating the flight of particles. However, for the collision detection between fragments and complex targets, it is necessary to introduce the Bullet collision detection engine [8].

The process of multi-fragment acting on complex targets needs to go through the following basic steps: the establishment of 3D scenes, the realization of core algorithms, collision detection, and damage analysis. The establishment of the 3D scene is realized through the OSG engine, including the import of the 3D model and the setting of the OSG scene coordinates; the realization of the core algorithm is to give the angle and direction of the fragment flight in the scene and the attenuation of the fragment flight; collision detection needs The collision detection between the tens of thousands of fragments formed by the warhead and the target is realized; the damage analysis is realized based on the damage evaluation theory algorithm and the collision detection results, mainly using the multi-fragment damage criterion.

For collision detection, the traditional collision detection technology is to achieve collision detection between two different objects, but for the multi-fragment damage simulation of warheads, the number of fragments formed is tens of thousands, and each fragment must be Collision detection between targets. Compared with the previous introduction of collision detection technology into the 3D engine, this paper adopts distributed computing for collision detection technology to optimize the detection efficiency for the damage assessment simulation of complex targets with multiple fragments.

2473-3636/22/$31.00 ©2022 IEEE
DOI 10.1109/DCABES57229.2022.00056

III. DISTRIBUTED COMPUTING

A. Definition of Distributed Computing

With the continuous increase of computing requirements, a single computer can no longer fully meet the computing tasks, and needs to be completed by a group of computers, which is commonly referred to as a cluster architecture. Now what we call distributed computing mainly refers to the distributed computing work under the cluster architecture. This kind of work requires starting from a client node, dividing a computing work into several small subtasks, assigning them to multiple server nodes for processing, and obtaining the results from these servers.

Distributed is a solution to large-scale computing problems using the computing power of other idle computers.With the development of computing technology, some applications require huge computing power to complete, such as centralized computing, it takes a long time to complete. Distributed computing decomposes the application into many small parts and distributes them to multiple computers for processing. In this way, the overall computing time can be saved, the computing efficiency can be greatly improved, and the computing load can be balanced on multiple computers through distributed computing.

B. Distributed Computing Combines Fragment Flight and Collision Detection

High-quality fragmentation damage simulation requires the simulation of the physical state of each fragment of the warhead. The simulation of large-scale damage scenarios requires the coupling of hundreds of thousands of fragments and targets at the same time, including the coupling of multiple fragments and the coupling of multiple damage elements. The computing efficiency of a single computer can no longer meet the demand, and the application of distributed computing will greatly improve the computing efficiency.

When the software triggers a large number of collision coupling tasks, the task will be passed to the task node manager. The task node manager first evaluates the computer resources that can be allocated, and traverses the status of these computer nodes, and allocates subtasks to each computer node as a whole by analyzing the amount of computing tasks. The task node manager monitors the status of each computing node at the same time, and feeds back the completed task results to the main computer node to complete distributed computing.

IV. COMBINATION OF OSG AND BULLET

A. OSG and Bullet

OSG (Open Scene Graph, Open Scene Graph) is a 3D rendering engine developed by Don Burns and Robert Osfield, and its main functions focus on structural scene management and 3D scene rendering [9]. Real or imagined 3D scenes can be simulated through OSG simulation, and its sub-project OsgEarth provides a complete development environment for 3D Earth.

Bullet, an open-source collision dynamics detection library, is known for its fast and stable computation [10]. The Bullet physics engine obtains the real motion of the object, the deflection during the motion, and the collision during the motion by giving the rigid or elastic object real physical properties and calculating based on the collision dynamics.

B. Combination basis of OSG and Bullet

The functions of the OSG physics engine include the management of the scene structure and the rendering of the 3D scene, but it is far from meeting the accuracy requirements of real collision detection in describing the collision detection between objects in the 3D scene. The Bullet physics engine analyzes the motion and force of the object by setting the physical characteristics of the simulated object, and obtains the constraint information and mechanical characteristics of the object at different times according to the Newtonian mechanics principle, and simulates the collision response in real situations. The introduction of the Bullet physics engine technology into the OSG scene structure can effectively make up for the lack of collision detection in OSG, make the motion in the 3D scene more in line with the actual mechanical laws, and effectively improve the realism of the virtual 3D scene[11].

V. COLLISION SIMULATION IMPLEMENTATION

A. Multi-fragment and target collision detection and response process

Figure 1 shows the process of realizing the collision detection between multiple fragments of the warhead and the target in OSG.

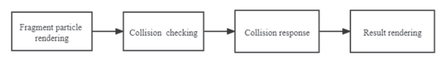


Fig.1 Collision detection process between multiple fragments and targets

B. OSG Bounding Box Collision Detection Technology

1) target bounding box

Before building the collision detection module, it is necessary to build the bounding box of fragment particles and complex targets in the OSG 3D scene. There are four kinds of bounding box technologies commonly used at present, namely bounding sphere technology, axial bounding box AABB technology, directional bounding box OBB and discrete directional polyhedral bounding box k-DOP technology. By comprehensively comparing the advantages and disadvantages of the four commonly used bounding boxes, this paper finally selects the AABB axial bounding box technology to construct the bounding box of complex targets[12~13]. On the one hand, most commonly used target structures for damage are axisymmetric and the shape is close to a cuboid, while the axial bounding box AABB is often used for collision detection of cuboid or strip-shaped object models [14]; on the other hand, the AABB axial bounding box can be used to detect collisions. While ensuring the collision detection accuracy, the calculation time and calculation efficiency are guaranteed.

In the OSG 3D scene, the axial bounding box is usually constructed along the axis (X, Y, Z) of the coordinate system where the object is located, and only six scalars can describe this bounding box. Taking the aircraft target as an example, construct the axial bounding box AABB of the target. Project the aircraft target to the X, Y, and Z coordinate axes, and obtain the maximum and minimum values of the target projection. The 6 scalar values obtained by division are 3 matrices: Mat[0]=[Xmin, Xmax], Mat[1]=[Ymin, Ymax], Mat[2]=[Zmin, Zmax], by combining The coordinate information of the 8 vertices of the axial bounding box can be obtained. The largest projected area of the aircraft on the XY plane is a rectangle composed of Xmin, Xmax, Ymin, and Ymax, and the largest projected area on the XZ plane is a rectangle composed of Xmin, Xmax, Zmax, and Zmin, and the largest projected area on the YZ plane is A rectangle composed of Ymax, Ymin, Zmin, and Zmax. The collision detection principle is as follows: if the projection intervals of the colliding object's bounding box and the axial bounding box AABB of the collided object overlap on three planes at the same time, it can be judged that there is a collision between the two objects, otherwise there is no collision.

2) *Fragment particle bounding box*

For the bounding box of the fragment particle system, the key is to realize the realization of the bounding box of tens of thousands of fragment particles. Collision coupling between a large number of fragments and hundreds of thousands of target model meshes will result in a very large computational load. In order to improve the efficiency of collision detection, this paper adopts the self-developed bounding box technology for fragment particles. Based on the collision detection model of triangular meshes, feature elements such as points, lines, and surfaces are added to the triangular mesh to describe triangles. . At the same time, the bounding box of fragment particles is constructed by combining the method of hierarchical bounding box.

3) *Collision detection optimization*

The basic common method of collision detection is to test all objects in pairs. When the number of objects is small, this method is simple and fast. However, for the collision detection of tens of thousands of fragments formed by the warhead and complex targets, the algorithm of collision detection must be improved.

The design principle of the optimized collision detection algorithm is: based on grid detection, first divide the entire area into grids, and then based on grid detection, traverse all grids, and compare grid feature data. The size of the grid division is based on the size of the largest colliding object, ensuring that the grid is larger than the bounding box of the colliding object. When traversing the grid, it is necessary to filter and optimize the scattering angle of the fragments and the characteristic data of the grid, and eliminate the grids that are obviously unable to collide. The detection of grids only needs to perform grid-by-grid detection along the scattering direction, and does not need to detect each grid, which improves the computing performance. At the same time, distributed computing is adopted for the collision detection task of each fragment, and each task is divided into multiple servers for processing to improve computing efficiency.The module establishes a class for collision detection algorithm, which is mainly used to detect whether there is a collision between each fragment bounding box and the complex target bounding box.

The specific process is as follows:

- By classifying and cutting the target mesh, multiple bounding boxes are formed.
- By calculating the position of the center of the warhead, the corresponding target grid bounding box is optimized and screened, and a collision scene is constructed.
- Add the warhead to the scene as a dynamic rigid body and the target as a static rigid body.
- Perform collision detection on the models in the scene.

C. *Collision structure rendering and calculation separation*

1) *Build the scene hierarchy tree*

The reasonable construction of the scene hierarchy tree is the premise to improve the computational efficiency of the collision detection algorithm.

Obtain the model file through osgDB::readNodeFile, and adjust the pose position through osg::MatrixTransform. The warhead model does not need to traverse the surfels to join the main node directly, while the target surfel needs to be traversed and added to the main node after the surfels are traversed. Traverse and save the target surfel information for subsequent collision detection and surfel assignment.

2) *Collision detection result rendering optimization*

Although the optimized collision detection algorithm improves the efficiency of collision detection, the collision between tens of thousands of fragments and millions of grids still cannot be completed in an instant. In order to improve the user experience, the real-time rendering based on the collision results of a large number of objects adopts an algorithm that separates real-time collision and rendering. In this paper, distributed computing is used to solve the collision detection results, and the rendering results are processed in a multi-threaded parallel manner. The result calculated by the collision detection is transmitted to the rendering thread in real time through the server. After the rendering thread receives the data, it performs grid positioning, damage determination and other operations, and finally renders the damage result to the target grid.

VI. SIMULATION EFFECT

A. *Fragment particles scatter*

The realization of fragment particle scattering is the basis for realizing the collision detection module between fragments and targets. The initial parameters and motion properties of fragments can be obtained through the calculation of theoretical formulas. The parameters involved in fragment particle scattering include fragment initial velocity, fragment axial velocity distribution, fragment scattering angle, fragment number, fragment mass distribution, fragment velocity attenuation and fragment density distribution. The sources of these parameters are calculated by theoretical calculation formulas, and the results can be displayed in the form of visual charts.

First, through Shuyuan Sui and Shushan Wang's "End Point Effect Science" [5], a mature theoretical empirical formula was obtained.

Then, the algorithm is implemented in the VS platform through the C++ language, and the program is packaged into .dll after verifying the correctness.

Finally, the .dll of the calculation module is called in the simulation of damage assessment of complex targets with multiple fragments to realize the visualization of the results of fragment particle scattering. Figure 2 shows the particle scattering attenuation curve of a fragment.

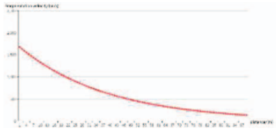


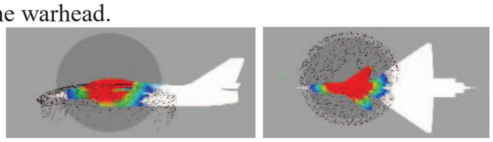
Fig.2 Fragment particle velocity attenuation

Taking the movement of the warhead as an example, the main parameters such as the dynamic scattering angle and the motion decay time of the fragments are obtained through theoretical formulas. OSG is used to render the scattering of fragments, and each fragment particle is assigned a value to obtain a 3D visualization scene that simulates the scattering of real particles. Figure 3 shows the 3D visualization scene of the fragment .

Fig.3 Visualization of fragment dynamic power field

B. Damage assessment simulation

Taking the aircraft target as an example, construct a bounding box of multiple fragments and the aircraft target. Figure 4 shows the three-dimensional damage scene of the aircraft target under the action of multiple fragments formed by the warhead.



(a) Side view (b) Top view

Fig.4 Damage process of aircraft target under the action of multiple fragments

Figure 5 shows the 3D damage scene of the aircraft target at the end of the damage calculation. The red particles in the picture represent the fragments formed by the warhead. Different colors represent different degrees of damage to aircraft components, with red for severe damage, cyan for moderate damage, blue for minor damage, and white for no damage.

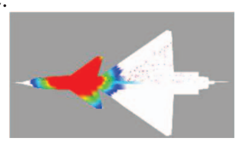


Fig.5 3D damage scene of aircraft target

Through multiple statistics, the average calculation time of the complete fragment collision aircraft detection process is about 10s. It can be concluded that the optimized collision detection algorithm based on distributed computing can quickly realize the collision detection between tens of thousands of fragments and the aircraft target, and at the same time show the damage effect of the aircraft target under the action of multiple fragments.

VII. CONCLUSION

In this paper, aiming at the 3D simulation of damage assessment of multi-fragmented targets, a calculation scheme combining OSG scene and Bullet collision and a collision detection optimization algorithm based on distributed computing are proposed. Taking an aircraft as an example, the optimized algorithm can quickly realize the collision detection between tens of thousands of fragments and the target, and at the same time realize the cloud image rendering of the damage assessment results. The damage calculation results obtained through this technical solution can effectively simulate the actual combat effect of the warhead, and provide a basis for optimizing the design of the warhead and evaluating the damage of complex targets.

REFERENCES

[1] Xiangyu Li, Fangyun Lu. Numerical Simulation of Three Types of Warhead Fragment Dispersion [J]. Journal of Propellants and Explosives,2007(01):44-48.

[2] Jiang Yu, Dangyong Shi, Changfu Xia,et al. Research on the Numerical Simulation Method of Natural Fragmentation Warhead Based on LS-DYNA [J]. Pyrotechnics, 2009, (1): 25-29.

[3] Keping Guan, Jingnan Jiang, Tianyu Wu,et al. Three-dimensional simulation of ship collision response based on OSG and Bullet [J]. Journal of Shanghai Maritime University, 2014, 35(04): 1-6.

[4] Jianwei Jiang, Yonggang Lu, Lixin Qian. Application of shooting trace technology in warhead fragmentation field simulation [J]. Journal of Projectiles, Rockets, Missiles and Guidance, 2001(01): 29-34.

[5] Shushan Wang. End effect theory [M]. 2nd Ed. Beijing: Science Press, 2019: 180-200.

[6] Binggui Cui. Target damage engineering calculation [M]. 1st Ed. Beijing: Beijing Institute of Technology Press, 1995.

[7] Ruce Wang. End effect of projectiles [M]. 1st Ed. Beijing Institute of Technology Press, 1993.

[8] Yaojie Chen, Xingliang Zhu, Feng Ye,et al. Three-dimensional simulation system for marine fire fighting based on OSG and Bullet [J]. Computer Simulation, 2017, 34(03): 268-272.

[9] Chang Wei, Yue Cheng, Jiansheng Wang. Design and implementation of collision detection algorithm under OSG [J]. Information Technology and Information Technology, 2021(03): 100-102.

[10] Xiao Wang, Wanlin Lu, Fenghui Wang. Three-dimensional simulation of ballistic missile fire attack based on OSG [J]. Firepower and Command Control, 2021, 46(02): 157-162.

[11] Qingxuan Jia, Yuning Liao. Research and implementation of the combination of OSG and Bullet [J]. Computer Simulation, 2014, 31(08): 296-300.

[12] Chao Liu, Xiajun Jiang, Huibin Shi. Improvement of collision detection algorithm based on OBB bounding box [J]. Computer Technology and Development, 2018, 28(06): 43-48.

[13] KlosowskiJ T, Teld M, etal. Efficient collision detection using bounding volume hierarchies of K-DOPs [J]. IEEE Transactions on Visualization and Computer Graphics, 1998, 4(1): 21-36.

[14] Yi Xiong, Hua Huang, Lei Wang,et al. A fast algorithm for solving the quasi-minimum bounding box of rectangular objects [J]. Application Research of Computers, 2020, 37(S1): 157-158.

[15] Jialin Quan, Runyuan Miao, Zhengfeng Liang. The speed attenuation law of different types of prefabricated fragments and the correction of the windward area [J]. Blasting Materials, 2019, 48(06): 28-32.