Collision Detection Methods for High-speed Virtual Simulation Objects

Mingcong Ma, Gang Xu, Xiao Zhan, Xiaopei Yang
Training Department, Xuzhou Air Force Academy, Jiangsu Xuzhou, 221000, China
xgeml@163.com

Abstract—**In virtual reality simulation, simulated objects under high-speed motion are easy to occur collision detection omission and inaccurate initial point of collision detection at adjacent discrete time point, considering this problem, collision detection methods of functions motion with time are given, which are based on integration of relative motion and space geometrical operation. Relative motion theory is used to reckon movable coordinates transformation matrix functional equation, perform space curve and geometry intersecting test under new motion path, pre-calculate collision time and initial point of collision, finally move and transform this collision point back to system coordinate space and achieve collision position under motion state. This method can effectively and accurately detect the problem of collision under high-speed motion.**

Keywords-collision detection; collision response; virtual reality; target-sensitivity projectile

I. INTRODUCTION

Collision, collision detection and collision response are very important technology in virtual reality, and the important factor impacts fidelity and feeling of reality of virtual reality system [1]. With the deepening application of simulation technology in military and expansion of scope, the requirements for precision and effectiveness of military simulation system have become more, collision detection and response problems in military simulation system must be deeply researched with pertinence[2-4].

Because the time in virtual simulation environment is discrete, objects under high-speed motion are easy to occur collision detection omission and inaccurate initial point of collision detection at adjacent discrete time point, thus directly resulting in unreal and inaccurate collision response results. Focusing on this specific problem, this paper develops discussion and gives detection methods for collision under high-speed motion state.

II. ANALYSIS FOR COLLISION PROBLEM UNDER HIGH-SPEED MOTION STATE

Collision problem in military simulation system include two aspects: first, how to perform real-time collision detection according to battlefield environment; second, how to perform collision post processing according to specific conditions and simulation requirements, that is collision response[1]. Collision detection will directly influence collision response results.

Virtual simulation system displays dynamic scene content with picture frame by frame, each picture corresponds to a time point t_i, display efficiency of picture is measured by "frame/second (fps)" unit, indicated with FS (average display efficiency) symbol, then time step size can take $\Delta t=t_{i+1}-t_i\approx 1/\mathrm{FS}$; set up two motion objects, which are respectively O_A and O_B, the velocity of O_A and O_B are respectively $v_A(t)$, $v_B(t)$ based on time function, system space coordinates are respectively $P_A(t)$ and $P_B(t)$ based on time t function, diameter of bounding spheres is respectively D_A, D_B. If O_A and O_B are under high-speed and straight-line motion in Δt time, the circumstances in Fig. 1 will occur.

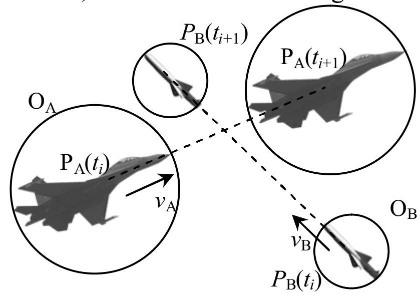

Figure 1. Missiles and planes under high-speed motion.

According to Fig. 1, the bounding spheres of O_A and O_B at t_i and t_{i+} is not intersected, then it is unable to know whether collision occurs during adopt Δt process by using static state detection method, even if penetration test is adopted at t_i, it is unable to accurately find out initial point of collision. Obviously, traditional static collision test method adopted for such high-speed motion will cause big error, even detection failure. In addition, if two objects are regarded as space geometric solid with certain volume, then such collision detection problem under high-speed motion state should not be simply summarized as space curve intersecting problem.

III. COLLISION DETECTION METHODS

Considering virtual object collision detection under high-speed motion is easy to generate big error and detection omission phenomenon, the feasible methods for solving this problem include:

Subdivide time or path from starting point to finish, then carry out static collision detection at each subdivision point; if too many subdivision joints are utilized, large quantity of calculation time will be consumed; if subdivision number is too small, missing judgment or fault judgment phenomenon will occur as well, number of subdivisions is hard to control, so number of subdivisions will impact the correctness and algorithm efficiency of detection results.

If within Δt time step size, high-speed virtual geometry motion path is approximate to a straight line, then it is able to adopt "four-dimensional intersecting test" method proposed in reference [2], take a three-dimensional object to generate one scanned volume included in its time course and constitute new 3D object, if object collision occurs, then their scanning volume must be intersected. At first, make preliminary prediction of collision time according to the intersection point of two scanning volumes under high-speed motion, then take samples at predicted time for static and precise collision. This method is very easy to be perceived, but one of the problems is very difficult to clearly indicate scanning volume; if motion path is non-linear motion within Δt time, and geometric solid of the object is complicated, it will be more difficult to construct scanning volume, so this method has certain limitations, and its application scope is small.

This paper regards one of high-speed virtual objects as "object of reference", simplify two high-speed motion collision problems and implement collision detection under new motion path. The specific calculating process is given below.

A. Straight line high-speed motion within Δt time

When Δt takes a very small value, it is able to approximately regard the motion within Δt time as straight line motion, this is the most common method for virtual simulation system.

Take O_A as object of reference, speed of O_A is $v_{A\text{-}A}(t)=0$, then speed of O_B is

$$v_{B\text{-}A}(t)=v_B(t)-v_A(t) \tag{1}$$

Integrate both sides of Equation (1) at the same time, get

$$\int_t \begin{bmatrix} v_{B-A}(t)_X \\ v_{B-A}(t)_Y \\ v_{B-A}(t)_Z \end{bmatrix} dt = \int_t \begin{bmatrix} v_B(t)_X \\ v_B(t)_Y \\ v_B(t)_Z \end{bmatrix} dt - \int_t \begin{bmatrix} v_A(t)_X \\ v_A(t)_Y \\ v_A(t)_Z \end{bmatrix} dt$$

that is

$$P_{B-A}(t) = P_B(t) - P_A(t) \tag{2}$$

Write Equation (2) as matrix form equation:

$$P_{B-A}(t) = \left[P_B(t)_X \ P_B(t)_Y \ P_B(t)_Z \ 1 \right] \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -P_A(t)_X & -P_A(t)_Y & -P_A(t)_Z & 1 \end{bmatrix} \tag{3}$$

Equation (3) means matrix variation of O_B compared with spatial position of O_A as time t moves, furthermore, relative motion of O_B at t moment is determined by $v_{B\text{-}A}(t)$; therefore, from Equation (3) can we get: "take O_A as object of reference", virtually meaning any moment t, move and transform O_A to origin of system coordinate, and makes the same movement and transformation for O_B, until now, O_B has formed new motion path.

When time step size takes $\Delta t=t_{i+1}-t_i$, 由 it is not easy to deduct from Equation (2)

$$P_{B-A}(\Delta t) = (P_B(t_{i+1}) - P_B(t_i)) - (P_A(t_{i+1}) - P_A(t_i)) \tag{4}$$

Then two high-speed motion collisions in Fig. 1 is transformed into high-speed motion and "static" virtual object collision problems, as shown in Fig. 2.

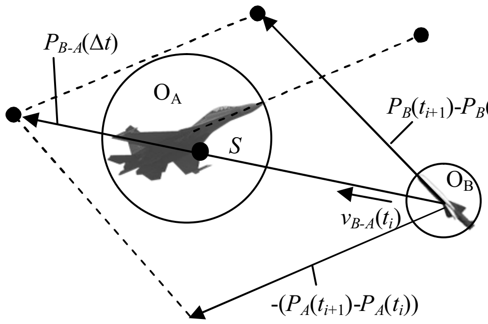

Figure 2. Preliminary judgment of collision point.

If straight line high-speed motion is in Δt time, then motion path of O_B compared with O_A coincides with $P_{B-A}(\Delta t)$, at this time, under new motion path, O_B adopts intersection of directed line segment and geometry O_A with starting point $P_{B-A}(t_i)$ and finish $P_{B-A}(t_{i+1})$ to judge collision problem, if neglecting O_B appearance, then regard the point nearest to O_B as preliminary collision point, as point S in Fig. 2, if unable to neglect O_B appearance, sampling is required near point S for precise collision detection, find collision and assign to point S.

Based on intersection operation of directed line segment O_A and all triangle mesh surface in $P_{B-A}(\Delta t)$, get collision occurrence time t_x, specific algorithm steps for solving t_x are as follows:

Step 1: first, judge $|P_{B-A}(\Delta t)|>|P_{B-A}(t_i)|-0.5\times D_A$, if it is true, enter into step2, otherwise $P_{B-A}(\Delta t)$ is impossible to intersect with bounding sphere of O_A.

Step 2: traverse all triangle grid patches in O_A and do intersection operation with $P_{B-A}(\Delta t)$, calculate t_x sequence and point of intersection P_0 sequence:

From three vertexes of triangle in O_A can get the equation of plane where this triangle locates:

$$ax+by+cz+d=0 \tag{5}$$

Then substitute relative motion track parameter equation of O_B

$$\begin{cases} x = P_B(t_x)_X - P_A(t_x)_X \\ y = P_B(t_x)_Y - P_A(t_x)_Y \\ z = P_B(t_x)_Z - P_A(t_x)_Z \end{cases} \tag{6}$$

into Equation (5), get t_x and point of intersection P_0.

Then point of intersection P_0 should be within $P_{B\text{-}A}(\Delta t)$, and inside the triangle, otherwise t_x and P_0 at this time will not be listed into sequence, and traverse the next triangle.

Set ΔABC is one triangle in O_A, then the necessary and sufficient condition of one point P_0 on the plane of this triangle in triangle ΔABC is: cross product $(A\text{-}P_0)\times(B\text{-}P_0)$ is same as $(B\text{-}P_0)\times(C\text{-}P_0)$ symbol[4].

Step 3: if t_x sequence and point of intersection P_0 sequence are empty, indicating $P_{B\text{-}A}(\Delta t)$ does not intersect with O_A; otherwise, regard point P_0 nearest to $P_{B\text{-}A}(t_i)$ as point of intersection S, time t_x corresponding to this point of intersection is collision time, until now, t_x has been solved.

Because O_A is moving before being regarded as "object of reference", actual collision position of t_x moment should be considered: transform O_A back to system spatial position at t_x moment, so the spatial position where actual collision occurs is:

$$S' = \begin{bmatrix} S_X & S_Y & S_Z & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ P_A(t_x)_X & P_A(t_x)_Y & P_A(t_x)_Z & 1 \end{bmatrix} \quad (7)$$

B. Non-straight line high-speed motion is in Δt time

For the convenience of research, first neglect O_B object appearance, and regard it as particle, specific steps are as follows.

- First conduct spatial movement and transformation for O_A to origin of system coordinates, make corresponding transformation for O_B as well, that is equation (3).
- Set time starting point as t_{start} and time finish t_{end}, then the coordinate starting point and finish of O_B after transformation are respectively $P_{start}=P_{B\text{-}A}(t_{start})$, $P_{end}=P_{B\text{-}A}(t_{end})$, suppose motion track of O_B is as shown in Fig. 3.

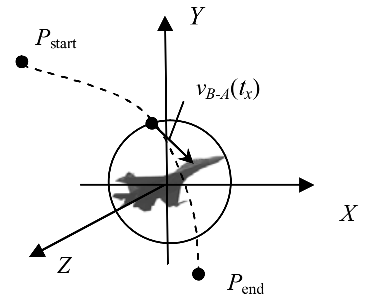

Figure 3. Preliminary Judgment of Collision.

- In order to improve calculation efficiency, first consider intersection operation of space curve $P_{B\text{-}A}(t)$ and bounding sphere of O_A:

Equation of bounding sphere surface of O_A is:

$$x^2 + y^2 + z^2 = \left(\frac{D_A}{2}\right)^2 \quad (7)$$

Substitute Equation (6) into Equation (8) to get quadratic equation or equation of higher degree. The equation may have many solutions, then optimum solution t_x should be within Δt, and t_x- t_{start} is minimum value, in this way, collision time can be got; if no solution, then consider no collision occurred.

- If solution is available, implement static collision detection and intersection test along the direction of $v_{B\text{-}A}(t_x)$, the nearest point of intersection is approximately used as collision initial point S, specific algorithm steps are shown in the previous section.
- Finally substitute *point S* into equation (7), transform back to original system space and get actual space collision position S'.

The above process is to judge whether collision occurred under different motion conditions within Δt time step size , but if requiring to precisely find out collision initial point, sampling should be done near the places where collision may occur to do intersection operation of grid surfaces, which is bound to result in decrease of calculation efficiency, therefore, it is able to integrate optimized algorithm to improve its calculation efficiency on the basis of this method. When the precision is not required to be very high, collision point S' calculated above can meet the general demand of military virtual simulation system.

IV. Application

During research of damage effect of target-sensitivity projectile, virtual simulation technology is introduced, as shown in Fig. 4. At the stage of "projectile intersection", target-sensitivity projectile casts explosion forming projectile (EFP), strike the armored target with fast maneuvering and excellent protection. Different hitting positions will result in different damage levels and effects.

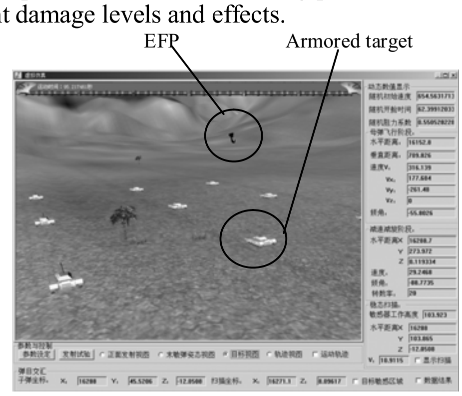

Figure 4. Target-sensitivity projectile is attacking armored target.

According to explosives combustion model and various parameters, it is able to calculate EFP is shooting to target at high speed more than 2000m/s, if scanning height parameter of target-sensitivity projectile is set 150m, scanning angle 30 degrees, then attacking distance should be within 200m; in addition, when display efficiency of virtual simulation system FS= 30 fps, then moving distance of each frame of EFP will be more than 67m, but armored target height generally will not exceed 3m; so armored target is completely possible to penetrate between two adjacent frames. Since collision has occurred, traditional static collision detection is easy to occur omission.

Since speed of EFP is very fast at initial stage, motion track is approximate to one straight line, and the speed is much faster than that of armored target, therefore, the moment when EFP attacks the target is regarded as static armored target, the problem has been simplified. P_1P_2 in Fig. 5 is a possible motion path, by adopting the methods provided in this paper, consider point P is collision initial point, hitting cockpit belongs to Level M damage, unable to be repaired on battlefield; and by using LOS algorithm in Vega software[5], consider point C as collision initial point, which is obviously not correct enough, providing a wrong judgment for observers.

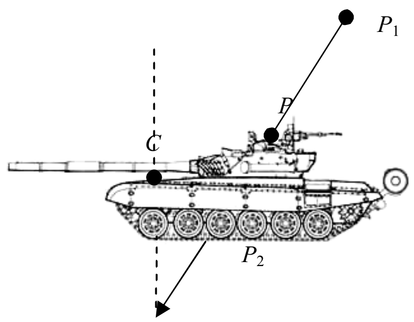

Figure 5. Error occurred when target-sensitivity projectile is attacking armored target.

V. CONCLUSION

Collision detection omission and error of virtual target under high-speed motion is one of the most common problems in military virtual simulation system. This paper integrates the knowledge of motion theory and computer graphics to specifically describe the judgment methods of collision detection and collision points under high-speed motion state. The methods are simple but effective and practical, which are suitable for both collision detection under high-speed motion and accurate judgment of collision problems among the objects not in high-speed motion.

REFERENCES

[1] Shanda Feng, Yixin Liu, "Research on representative problems of collision in virtual field environment," Journal of system simulation, China, Vol. 18 No. 6, 2006, pp. 1582–1584.

[2] Cameron S, "Collision detection by four-dimensional intersection testing," IEEE Transactions on Robotics and Automation, June 2009, pp. 291–302.

[3] Zhiqiang Wang, Jiazhen Hong, Hui Yang, "A survey of collision detection problem," Journal of software, China, Vol. 10 No. 5, 2007 pp.545–551.

[4] Jiaguang Shun, "Computer graphics (third edition)," Beijing: Tsinghua university press, China, Dec. 2008, pp. 393–482.

[5] Bo Yiang, "Virtual reality technology—MultiGen, Vega applied research," University of electronic science and technology, China, 2002.