Acta Technica 62 (2017), No. 7A, 1113–1124

Collision detection method for sports ball game in virtual reality designed for user experience

Bin Zhang[1], Dazhi Feng[1]

Abstract. In order to improve effectiveness of collision detection method for sports ball game, a collision detection method of sports ball game in virtual reality designed for user experience and based on particle swarm algorithm is proposed in the Thesis. Firstly, sports ball collision model based on bounding box is constructed in virtual reality designed for user experience and intersection test model of FDH is given, which is very beneficial to improve performance of collision detection system of the whole ball objects; then, an existing adaptive variable space algorithm is improved, and thought of the algorithm is introduced into improvement of particle swarm algorithm. The purpose of finding proper searching space automatically, improving convergence speed and accuracy, and preventing premature convergence of particle swarm algorithm effectively is realized; finally, effectiveness of proposed method in collision detection of sports ball game is verified through simulation experiment.

Key words. User experiment, Virtual reality, Sports ball, Ball object, Collision detection.

1. Introduction

Along with constant development of computer hardware technology, virtual environment has become an important research field of computer science, and can be widely applied to education, national defense, medicine, art, entertainment and multiple aspects. In order to guarantee reality of virtual environment, participators shall not only be able to see virtual objects and their expression in virtual environment truthfully and visually, but also can have various interactions with them just like on the scene, which requires that solid object in virtual environment is impenetrable firstly. Participators can feel existence of collision truthfully and can make corresponding reaction in real time when they touch objects and implement operation of pull, push, grab and so on. In addition, a virtual environment generally includes

[1]Department of physical education, Jinzhou Medical University, Jinzhou, China

several static environment objects and moving active objects. Geometric model of each virtual object is constituted by thousands of basic geometrical elements (such as tetrahedron or triangle). Geometry complexity of virtual environment makes calculation complexity of collision detection of ball objects improved greatly, while interaction between participators and virtual environment requires real-time completion, therefore collision detection of ball objects always becomes a bottleneck of virtual environment.

Thus it can be seen that accurate collision detection of ball objects plays an vital role in improving reality of virtual environment and strengthening immersion of virtual environment, while own complexity and real-time performance of virtual environment have higher requirements for collision detection of ball objects. Taking virtual campus roaming as application background, campus of Central China Normal University as virtual space, virtual campus construction of Central China Normal University as purpose, a collision detection algorithm for ball objects of layered bounding box tree with convex hull in fixed direction based on particle swarm is proposed in the Thesis by combining collision detection algorithm for ball objects of hierarchical bounding box of space decomposition and particle swarm algorithm. Layered tree method is adopted for hierarchical bounding box algorithm of space decomposition, top-down method is adopted for organization of bounding box and layered bounding box tree is established according to certain rules. In the process of traversal for layered bounding box tree, geometric element pairs that are impossible to be intersected are eliminated as soon as possible by rapid intersection test among bounding boxes, thus reducing quantity of intersection test of bounding boxes effectively and improving efficiency of algorithm. Particle swarm algorithm reflects a simple and plain intelligence thought, and this thought generates a optimization algorithm mathematically. Combination of two algorithms reduces response time of collision detection of ball objects, improves operation efficiency of algorithm, and strengthens reality of virtual environment perception.

Realization of collision detection technology for ball objects in roaming system of virtual reality is studied and discussed in the Thesis, and function realization is implemented by experimental simulation, which provides important approach and methods to realize roaming system of virtual natural scenic spots and research and realization on virtual laboratory and so on.

2. Collision model of sports ball based on bounding box

2.1. Hierarchical structure of bounding box

As for collision detection of sports ball game in virtual reality designed for user experience, adopting hierarchical bounding box method is that describe complex geometric objects by bounding box with slightly large volume and simple geometric characteristics, and then make it closer and closer to geometric models of objects by constructing tree hierarchy until it almost completely obtains geometric characteristics of objects, thus implementing further intersection test for overlapping parts with bounding box is only required. In collision test of sports ball game in virtual reality designed for user experience, node of hierarchical structure corresponding to a ball object is geometric approximation body (bounding box) surrounding a part of geometric objects of the ball object in the space. Root node of hierarchical structure surrounds the whole ball object, geometric objects surrounded by each parent node is the sum of geometric objects surrounded by its all root nodes, and note approaches geometric object surrounded by it generally from top to bottom.

When hierarchical bounding box method is adopted for collision detection of ball objects, if bounding box is not intersected, ball objects will not be intersected absolutely. Only when bounding box is intersected, can further intersection test be implemented for surrounded basic geometrical elements, namely: judge whether root nodes of two bounding box trees are intersected firstly, if they are intersected, judge intersection conditions of their child nodes further, if there are intersected leaf node pairs finally, judge whether basic geometrical elements surrounded by each pair of leaf node are intersected. In collision detection algorithm of ball objects, hierarchical bounding box is the simplest and most common hierarchical data structure. This structure representation is easy to be constructed and transplanted, and it is not determined by topological property of any models at the same time. Hierarchical bounding box is a dendritical structure based on bounding box hierarchy, including rectangular bounding box, sphere bounding box and so on. Each node in the tree is corresponding to certain model space, root node is set constituted by geometric model space of object or geometrical elements of object, child node is set constituted by some geometrical elements or space occupied by some geometries. Structure of bounding box is constructed according to size, volume, diameter of space object or some other characteristics, making it minimum or the most compact. The key of constructing hierarchical bounding box is grouping of ball objects in the space, and shape of hierarchical structure relies on spacial distribution of ball objects in operating environment. Generally speaking, bounding box is decreased along with increase of tree hierarchy increase. The process of constructing hierarchical tree is based on recursive approach of ball objects. This representing structure is simple and it is easy to implement intersection test, but overlapping phenomena will appear in the area.

In collision detection of sports ball game in virtual reality designed for user experience, as for given set S of n basic geometrical elements, hierarchical structure of bounding box $\mathrm{BVT}(S)$ in S is defined as a tree, is called bounding box tree for short. It has the following properties:

(1) Each node in the tree is corresponding to one subset $S_v (S_v \in S)$ of S;

(2) Bounding box $b(S_v)$ of set Sv is also related to each node v;

(3) Root node is corresponding to universal set S and bounding box $b(S)$ of S;

(4) There are more than two child nodes in each internal node (non leaf node) of the tree, the biggest child node number in internal node is called degree and denoted as δ;

(5) Subsets of basic geometrical elements corresponded by all child nodes of node V constitute one division of subset S_v of basic geometrical elements corresponded by V.

We call a bounding box tree is complete when and only when each leaf node of bounding box tree is corresponding to one single element subset of S, namely when only one basic geometrical element is included. It can be known from the above description that there are $2n-1$ nodes of bounding box tree at most, including n leaf node; height of one complete bounding box tree is $\log_\delta n$ at least, it is called that the tree is balanced at that time. Construction process of hierarchical bounding box is shown in Fig. 1.

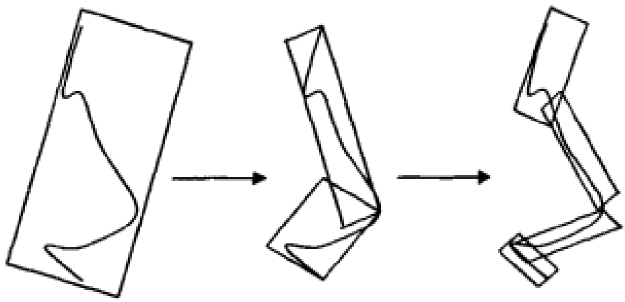

Fig. 1. Construction process of hierarchical bounding box

2.2. Collision detection of ball objects based on bounding box

Collision detection of sports ball game in virtual reality designed for user experience is a mechanism of automatic report when surface of a geometry is about to or has been collided, is a problem with more discussion in computer graphics field. Collision detection of ball objects is widely applied to CAD/CAM, robots, automation, computer animation and virtual environment and other fields. Through collision detection of ball objects, application of design and engineering analysis, virtual browse and so on requiring contact analysis and spatial reasoning based on simulation can be realized. In these application at the same time, realization efficiency of collision detection algorithm of ball objects is also usually regarded as efficiency bottleneck of system. In collision detection operating stage of ball objects, algorithm will traverse hierarchical binary tree of ball objects. Each node in hierarchical tree is bump surrounded by OBB. In collision detection of sports ball game in virtual reality designed for user experience, when hierarchical trees of two ball objects are traversed at the same time, whether OBB bounding box corresponded by bump node is intersected shall be detected firstly; when two OBB bounding boxes are intersected, triangular intersection shall be implemented for two OBB bounding boxes and correct results of collision detection of ball objects are analyzed and obtained. Overall framework of algorithm is shown in Fig. 2.

2.3. Intersection test of FDH

The purpose of using bounding box tree in fixed direction for collision detection of ball objects is to eliminate all basic geometrical element pairs that are impossible to be intersected as soon as possible, and only implement accurate intersection test for

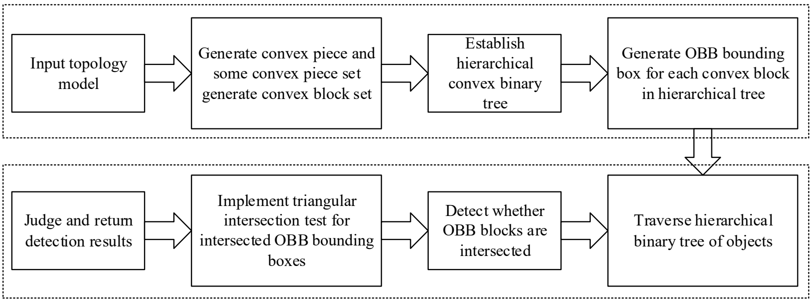

Fig. 2. Collision detection algorithm diagram of ball objects based on convex hull tree

possibly intersected basic geometrical elements by overlapping test among bounding boxes in fixed direction corresponded by all nodes in bounding box tree of two objects. Therefore, speed of overlapping test among bounding box influences speed of collision detection of ball objects directly. It can be known from definition of FDH that all FDH are defined by $k/2$ vector pairs with opposite direction in set D in the same fixed direction. So a FDH can be defined by its projection range in $k/2$ axis completely, when k=6 and vector value in D is direction of coordinate axis, FDH is degraded into AABB. Therefore we hope this simple and effective interval overlapping test method can be promoted to intersection test among FDH.

Inference 1: as for straight line , if areas where convex polyhedrons P and Q project vertically to L are not intersected, straight line L is called separate axis of convex polyhedrons P and Q.

Because all FDH are defined by a fixed direction set D, and it indicates normal vectors of their all sides come from . Sides of FDH are ensured by their two sides, therefore, side direction of FDH can be obtained through defining pair cross product for normal vector of its k sides. Hence, edge directions of FDH also comes from a fixed direction set. From Inference 1, it can be concluded that: if two projective intervals do not overlap in one direction of FDH edge direction set and they will not intersect absolutely, however, if projective interval overlap in all directions, they will absolutely intersect.

The purpose of using bounding box tree is to exclude all impossible intersections earlier with efforts. If two FHD are not intersected, basic geometrical element set surrounded by them are intersected; even two FHD are intersected, and we can not conclude that basic geometrical elements surrounded by them are intersected and it needs to be explored further. Therefore, we can still use interval test method for FDH because the situation of non-intersection for FDH judged by it is accurate, however, if all $k/2$ ranges of two FDH are overlapped, we can consider they are intersected conservatively. If they are not intersected, the occasion for eliminating non-intersection will only be a little later with it and it has no influence on accuracy for result of collision detection of ball objects. Rapid FHD intersection test is very beneficial to improve performance of entire collision detection system for ball objects.

In conclusion, there are no more than k comparison calculations needed for intersection test of two FDH. Although it is more complicated than intersection test (6 times)of AABB however, compared with OBB, complicated degree of its algorithm is reduced greatly. (Intersection test of two OBB needs 15 times comparison calculation)

3. Improving particle swarm algorithm of self-adaptive strain space

Because searching space of basic PSO algorithm is fixed, it is unavoidable that there is blindness for swarm searching. If there is excessive researching space, convergence speed of algorithm will slow down and it is easy to be trapped in locally optimal solution; if the researching space is too small, global optimal solution may not be searched. Therefore, when basic PSO algorithm is used for optimization of high-dimensional function with multiple-peak values particle swarm algorithm, it is easy to be premature with unsatisfied convergence precision and speed. A self-adaptive strain space strategy proposed in literature [8] is used. Researching direction of solution is adjusted automatically with certain rules according to current solution distribution situation so that algorithm can operate in proper researching space through adjustment. At the same time, in order to accelerate convergence speed of self-adaptive strain space algorithm, change rules of its upper and lower bounds are modified and supplied properly.

3.1. Standard PSO algorithm

Initially PSO algorithm is used for imitating elegant and unpredictable movement of bird swarm graphically. Through observation of animal social behaviors, people find that social sharing of information in swarm is beneficial to acquire advantage in evolution, and it is regarded as base for developing PSO algorithm. Through adding speed matching of near neighbour, unnecessary varieties have been eliminated, and Initial version of PSO has been formed when multiple-dimensional researching is considered according to speed of distance [1]. Later on, shi et al. [5] have introduced inertia weight ω for better controlling development and exploration and current standard version is formed.

PSO is a algorithm based on swarm evolution, and swarm compiling of m particles flies with a certain speed in $n-$dimensional researching space. At the time of researching for each particle, considering past best point researched by it and past best point researched by other particles in group, position will be changed.

Speed and position of particle is changed according to the following formula [7]

$$\begin{cases} v_{id}^{k+1} = \omega \cdot v_{id}^{k} + c_1 \cdot \xi \cdot \left(p_{id}^{k} - x_{id}^{k}\right) + c_2 \cdot \eta \cdot \left(p_{gd}^{k} - x_{id}^{k}\right) \\ x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \end{cases} \tag{1}$$

Where, ω is inertia weight, c_1 and c_2 are acceleration coefficients, and ξ, $\eta \in$ are pseudo-random numbers distributing averagely in range [0, 1]. Speed of particle is limited within a maximum speed $v_{\max}$ range.

3.2. Improving thinking of self-adaptive strain space

Improving thinking of self-adaptive strain space consists of two stages: expansion stage and shrinking stage. First of all, let researching space as $[l^t, u^t] = \{[l_k^t, u_k^t], k = 1, 2, \cdots, n\}$, t as update algebra of researching space.

Let initial researching space is set as $[l^0, u^0] = \{[l_k^0, u_k^0], k = 1, 2, \cdots, n\}$ and initial zero point as $x_0 = (l_k^0 + u_k^0)/2$.

Expansion stage: in consideration of upper bound of researching space, if variable x_k^* of optimal individual x^* in current population satisfies, therefore, it illustrates that upper bound of current researching space and upper bound of last researching space for the variable are smaller. In order to include optimal solution in researching space, it should be expanded and the similar treatment method for lower bound is as follows:

(1) If $x_k^* \in (u_k^t/2, u_k^t)$, $t = t + 1$, $u_k^t = 2x_k^*$;

(2) If $x_k^* \in (l_k^t, l_k^t/2)$, $t = t + 1$, $l_k^t = 2l_k^t$;

If researching spaces of all parameters have not been expanded, shrinking stage will be shifted.

Shrinking stage: after completion of expansion stage, the optimal solution is included in researching space, however, given researching space is rather rough, therefore, elaborate expansion operation and shrinking operator is subject to refine researching space in shrinking space.

In similar, in consider of upper bound of the space, because range for inspection is in forward direction of original zero point x_0, shrinking operation is added uniformly in the adjustment of upper bound with upper bound u_k^t of current researching space or upper bound u_k^{t-1} of last researching space as metric, and it is shown in Fig. 3.

Under situation (1), $x^* \quad < \quad u_k^{t-1}$ expresses upper bounds u_k^t and u_k^{t-1} current and last researching space are all overlarge, and they shall be shrunk with the following treatment method:

Case1: $x_k^* \in ((l_k^0 + u_k^0)/2, u_k^{t-1})$, $t = t + 1$,

$$u_k^t = u_k^{t-2}, u_k^{t-1} = 0.5[u_k^{t-2} + (l_k^0 + u_k^0)/2], l_k^t = l_k^{t-2}, l_k^{t-1} = 0.6[l_k^{t-2} + (l_k^0 + u_k^0)/2];$$

Case1: if $x_k^* \in ((l_k^0 + u_k^0)/2, u_k^{t-1})$, $u_k^t = u_k^{t-2}$, $u_k^{t-1} = 0.5[u_k^{t-2} + (l_k^0 + u_k^0)/2]$, $l_k^{t-1} = 0.6[l_k^{t-2} + (l_k^0 + u_k^0)/2]$;

Under situation (2), x^* is slightly bigger u_k^{t-1}, but much less than u_k^t, and it expresses upper bound u_k^t of current researching space is overlarge. In order to acquire better approximation of x^*, the following treatment method is conducted:

Case2: $x_k^* \in [u_k^{t-1}, 0.75u_k^{t-1} + 0.25u_k^t)$, $t = t + 1$,
$u_k^t = 0.5(u_k^{t-2} + u_k^{t-1})$, $u_k^{t-1} = 1.2u_k^{t-2}$, $l_k^t = l_k^{t-2}$, $l_k^{t-1} = 0.6[l_k^{t-2} + (l_k^0 + u_k^0)/2]$.

Case 2: if $x_k^* \in [u_k^{t-1}, 0.75u_k^{t-1} + 0.25u_k^t)$, $t = t + 1$, $u_k^t = 0.5(u_k^{t-2} + u_k^{t-1})$, $u_k^{t-1} = 1.2u_k^{t-2}$, $l_k^t = l_k^{t-2}$, $l_k^{t-1} = 0.6[l_k^{t-2} + (l_k^0 + u_k^0)/2]$.

Under situation (4), is much bigger than , and it expresses that upper bound of researching space is too small. In order to acquire sufficient researching space, the following treatment method is conducted:

Case3: $x_k^* \in (0.25u_k^{t-1} + 0.75u_k^t, \infty)$, $t = t + 1$, $u_k^{t-1} = 0.5(u_k^{t-2} + u_k^{t-1})$, $u_k^t = 1.2x_k^*$, $l_k^t = l_k^{t-2}$, $l_k^{t-1} = 0.6[l_k^{t-2} + (l_k^0 + u_k^0)/2]$.

Case 3: if $x_k^* \in (0.25u_k^{t-1} + 0.75u_k^t, \infty)$, $t = t + 1$, $u_k^{t-1} = 0.5(u_k^{t-2} + u_k^{t-1})$, $u_k^t = 1.2x_k^*$, $l_k^t = l_k^{t-2}$, $l_k^{t-1} = 0.6[l_k^{t-2} + (l_k^0 + u_k^0)/2]$.

It is suitable for researching space under situation (3) without any adjustment. After such adjustment, the actual optimal individual is near $0.5(u_k^{t-1} + u_k^t)$, and the bound is suitable. The lower bound can be treated with similar method.

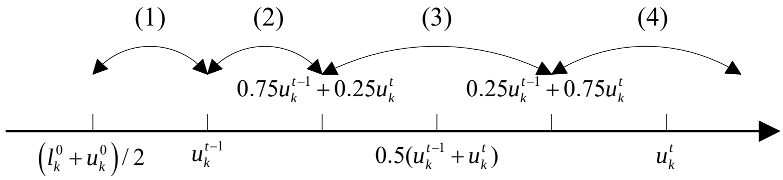

Fig. 3. Space division

When researching space of a variable is changed, m individuals are generated in changed researching space and its adaptive degree can be calculated. Individual with better adaptive degree is selected to substitute individual with worse adaptive degree and selected substitution rate is 5%-15% that is an overlarge rate for causing premature.

The improvement for algorithm of original variable space is that change of lower bound (upper bound) is added for it correspondingly and a simple and effective modification rule is given for accelerating of self-adaptive research when upper bound (lower bound) in research space is changed in shrinking stage. Algorithm for original variable space is referred to literature [8].

3.3. Improving particle swarm algorithm of self-adaptive space

Algorithm steps:

Step1: set swarm size of m, dimension number of n, inertia weight of ω, acceleration coefficient of c_1 and c_2, variable space cycle NC, and initial researching space $[l^0, u^0]$, $s = 1$.

Step2: swarm x with size of m is generated in $[l^0, u^0]$, and its adaptive degree is calculated.

Step3: If end condition is satisfied, it will stop and result will be output.

Step4: position and speed are changed according to (1).

Step5: if s is integral multiple of NC, $[l^{t-1}, u^{t-1}]$, $[l^t, u^t]$, and x is renewed according to above specified improvement of thinking in variable space.

Step6: Let $s = s + 1$, shift to Step3.

4. Experimental Analysis

4.1. Performance test

Compare ISAPSO, SAPSO, PSO and algorithm proposed in literature [9], and adapted test functions are: $f1$ indicates function of Dejong, $f2$ indicates function of Griewank, $f3$ indicates function of Rastrigin, $f4$ indicates function of Schaffer. Function of global optimum values of $f1 \sim f3$ functions are all 0, and global optimum values of $f4$ is -1.

$$f1 = \sum_{i=1}^{30} x_i^2 \,.$$

$$f2 = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1 \,.$$

$$f3 = \sum_{i=1}^{30} \left(x_i^2 - 10\cos(2\pi x_i) + 10 \right) .$$

$$f4 = \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} - 0.5 \,.$$

Initial researching space is selected as , parameter of particle swarm algorithm is set: swarm size: , dimension number: , inertia weight: , speed coefficient , variable space cycle , maximum evolution algebra , and average value is acquired with 20 times independent run of each algorithm for each function. Simulation result is shown in Table 1.

Table 1. ISAPSOSAPSOMPSO and PSO run 20 times for average

f	Algorithm	Optimal value	Iteration times	Variance	Time
f 1	ISAPSO	4.18×10^{-9}	2856	1.30×10^{-17}	249.7
	SAPSO	1.19	10000	1.09	919.4
	MPSO	5.07×10^{-5}	10000	4.34×10^{-9}	317.2
	PSO	6.06×10^{-2}	10000	4.43×10^{-4}	317.6
f 2	ISAPSO	5.54×10^{-9}	2162	1.41×10^{-17}	250.7
	SAPSO	9.25×10^{-2}	10000	1.09×10^{-2}	1172.0
	MPSO	1.13×10^{-5}	10000	1.73×10^{-4}	306.3
	PSO	1.61×10^{-2}	10000	1.97×10^{-5}	330.3
f 3	ISAPSO	9.17×10^{-10}	3085	3.26×10^{-18}	293.9
	SAPSO	2.06×10^{2}	10000	1.59×10^{3}	983.3
	MPSO	5.01×10^{2}	10000	2.57×10^{4}	303.8
	PSO	1.21×10^{2}	10000	6.82×10^{2}	342.4
f 4	ISAPSO	-1	1131	3.77×10^{-20}	1.37
	SAPSO	-0.99	10000	1.23×10^{-10}	12.9
	MPSO	-0.99	9052	1.13×10^{-4}	9.18
	PSO	-0.99	10000	2.26×10^{-5}	9.49

It can be seen from simulation result that if algorithm of original self-adaptive variable space is introduced to PSO algorithm directly for constructing algorithm without improving performance of SAPSO algorithm, there will be opposite function.

MPSO algorithm proposed in literature [9] expresses better performance, however, performances of $f3$ and $f4$ are not better than original PSO algorithm. Performance of new algorithm ISAPSO constituted of improved self-adaptive variable space algorithm in the Thesis and PSO algorithm is obvious better than SAPSO, MPSO, and original PSO algorithms. ISAPSO algorithm can jump out of original minimal point effectively and reach global optimal value and it has been verified by simulation result.

4.2. Collision test

In order to evaluate performance of collision detection algorithm of ball objects, we have introduced a cost function. It is proposed originally in research on ray trace in help of analyzing performance of hierarchy structure for bounding box. And Gottschalk, Lin, and Manochehave have quoted the same function for evaluating performance of detection algorithm for ball objects in collision detection for ball objects. After hierarchy structure of bounding box tree is established for two given geometric models, total cost in intersection test between them can be quantified as:

$$T_{cd} = N_b \times C_b + N_p \times C_p \,.$$

Where, T_{cd} serves as total cost for collision detection of ball objects among geometric models; N_b serves as number of intersection test for bounding box; C_b is cost of intersection test in a pair of bounding box; N_p serves as number of intersection test in basic geometric elements; C_p serves as cost of intersection test in a pair of elements.

In order to test performance and feature of algorithm, a group of experiments have been conducted by us. All experiments are conducted in CPU i5-3.0GHZ and internal storage of 4G. Scene 1: There are building model and vehicle model constituted of 845 vertices, for instance, collision test of ball objects for eigen pair of sampling group 4 is shown in Table 2. consuming time above 105, 505, 70%, and 90% and proportion of collision pair number for total collision pairs detection at the time of NN% over 90 are detected namely. Large researching space promises more sampling number with more collisions. Under situation without higher sampling precision, reducing sampling number can accelerate detection speed. Therefore, sampling is in accordance with requirement of scale and precision of ball object feature in actual practice for better detection effect.

Table 2. Collision detection comparison in four groups of sampling

Sampling number	10% of detection time/ms	50% of detection time/ms	70% of detection time/ms	90% of detection time/ms	Proportion for 90% of detecting collision pairs
1000*1000	2	3	7	11	5.6%
4000*4000	3	5	11	19	29.6%
7000*7000	5	8	17	28	78.0%
9000*9000	7	13	29	37	94.7%

Efficiency of algorithm is not only related to number of sampling but also influenced by scale of particle swarm. Within relative fixed time scope, there are more iteration times for particle swarm with small scale . However, under situation of larger solution space, speed for seeking optimum with small swarm will be slow, while there is huge cost of each iteration for swarm with larger scale and there will be fewer time for iteration in fixed time and particle is hard to converge. Therefore, it is better for most problems of seeking optimum to select middle-scale swarm as shown in Fig.4.

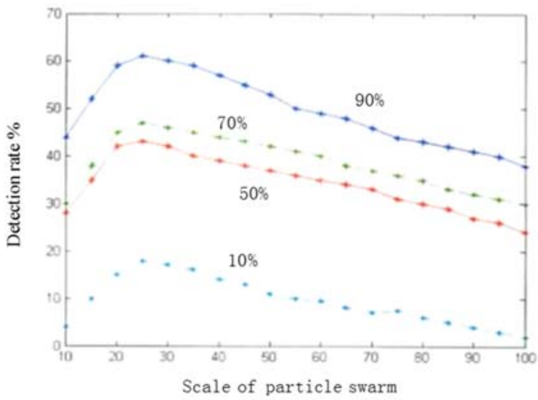

Fig. 4. Influence of change for particle swarm scale in 10-100 on collision detection rate of ball objects

5. Conclusion

Particle swarm algorithm has been introduced to the Thesis based on collision detection of ball objects for convex hull in fixed direction, and traversal trace strategy based on space-time correlation has been proposed. Traversal route of activity object in current time point has been ensured through tacking for activity object in last time point in traversal process of environment object tree for reducing number of intersection test for bounding box effectively and improving efficiency of algorithm, at the same time, accuracy and validity of collision detection for ball objects are guaranteed through maintenance of tracking table for decreasing complicated degree of algorithm time. With introducing algorithm of particle swarm optimization, the problem of collision detection of three-dimensional ball object in feature field has been shifted to two-dimensional discrete space for solving. Therefore, not only run speed and detection quality of algorithm will be controlled but also three-dimensional topological information doesn't have to be included into input information so as to reduce storage space and improve detection efficiency. The algorithm is available to real-time simulation system with high requirement of detection speed but lower requirement for precision as so to improve collision detection efficiency in these systems obviously.

References

[1] D. S. Abdelhamid, Y. Y. Zhang, D. R. Lewis, P. V. Moghe, W. J. Welsh, and K. E. Uhrich: *Tartaric Acid-based Amphiphilic Macromolecules with Ether Linkages Exhibit Enhanced Repression of Oxidized Low Density Lipoprotein Uptake*, Biomaterials, *53* (2015), 32–39.

[2] W. S. Pan, S. Z. Chen, Z. Y. Feng: *Automatic Clustering of Social Tag using Community Detection.* Applied Mathematics & Information Sciences, *7* (2013), No. 2, 675–681.

[3] Y. Y. Zhang, E. Mintzer, and K. E. Uhrich: *Synthesis and Characterization of PEGylated Bolaamphiphiles with Enhanced Retention in Liposomes*, Journal of Colloid and Interface Science, *482* (2016), 19–26.

[4] N. Arunkumar, K. M. Mohamed Sirajudeen: *Approximate Entropy based ayurvedic pulse diagnosis for diabetics - A case study*, TISC 2011 - Proceedings of the 3rd International Conference on Trendz in Information Sciences and Computing, (2011), art. No. 6169099, 133–135.

[5] N. Arunkumar, K. Ramkumar, S. Hema, A. Nithya, P. Prakash, V. Kirthika: *Fuzzy Lyapunov exponent based onset detection of the epileptic seizures*, 2013 IEEE Conference on Information and Communication Technologies, ICT 2013, (2013), art. No. 6558185, 701–706.

[6] J. J. Faig, A. Moretti, L. B. Joseph, Y. Y. Zhang, M. J. Nova, K. Smith, and K. E. Uhrich: *Biodegradable Kojic Acid-Based Polymers: Controlled Delivery of Bioactives for Melanogenesis Inhibition*, Biomacromolecules, *18* (2017), No. 2, 363–373.

[7] N. Arunkumar, V. Venkataraman, T. Lavanya: *A moving window approximate entropy based neural network for detecting the onset of epileptic seizures* International Journal of Applied Engineering Research, *8* (2013), No. 15, 1841–1847.

[8] J. W. Chan, Y. Y. Zhang, and K. E. Uhrich: *Amphiphilic Macromolecule Self-Assembled Monolayers Suppress Smooth Muscle Cell Proliferation*, Bioconjugate Chemistry, *26* (2015), No. 7, 1359–1369.

[9] Y. J. Zhao, L. Wang, H. J. Wang, and C. J. Liu: *Minimum Rate Sampling and Spectrum Blind Reconstruction in Random Equivalent Sampling.* Circuits Systems and Signal Processing, *34* (2015), No. 8, 2667–2680.

Received May 7, 2017