A Collision Detection Algorithm Based On Improved Quantum Particle Swarm Optimization

YUANHUA WANG, QIANG ZHANG*, DONGSHENG ZHOU
Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education,
Dalian University,
Dalian, 116622
China
*Corresponding author: zhangq30@yahoo.com

Abstract: - In the field of virtual reality, Collision Detection Technology was widely developed for improving the performance of 3D Graphics. Following rapid growth of virtual objects with complex shapes, conventional methods perform harder to effectively detect the collision. Facing the problem, we presented a collision detection algorithm based on improved quantum particle swarm optimization. Firstly, we converted the collision detection problem into nonlinear constrained optimization problem. Secondly, we employed the Euclidean distance to evaluate whether there was collision between two objects. Then, we improved the quantum particle swarm optimization (QPSO) algorithm by (1) using quantum H_ε gate and quantum rotation gate in changing quantum probability amplitude, (2) changing the mutation operator with Quantum Hadamard Gate and (3) modifying constant Inertia Weight to random inertia weight. In the end, the results of numerical simulation and analysis were provided to verify the validity of our algorithm.

Keywords: -Collision Detection; Quantum Rotation Gate; Quantum Hadamard Gate; Quantum H_ε Gate

1 Introduction

Collision detection problem was a long-term subject in 3D graphics of computer. It had been extensively applied in various fields, such as CAD/CAM technology, robot motion optimization, virtual reality, computer animation, computational geometry. The research of collision detection problem was based on a simple common sense: two objects cannot penetrate each other and share the same space[1].

Currently collision detection methods were roughly divided into hierarchy bounding box method[2], space subdivision method[3] and distance tracking method. Hierarchical bounding box method used bounding box to descript the complex geometrical crudely. The geometry was simple geometric object, slightly larger than the size of objects. After the bounding box test, people could remove the geometrical object that did not intersect rapidly, what reduced testing objects precisely. Among them, typical bounding box were axial aligned bounding box (AABB) [4], spherical bounding box[5] and space bounding box[6]. Axial bounding box contained oriented bounding box (OBB)[7], discrete oriented bounding box (K-DOP)[8] and so on. RG Zhang et al[9] proposed a z-buffer algorithm which combined the space projection method. Although such method accelerated the detection speed in a way, it expended the accuracy of collision detection at the same time. The bounding box method, because of its simple and convenient, was widely used in coarse detection. Due to the object bounding box was generally larger than the object actually occupied volume, the method had some limitations. Space subdivision method divided space into different regions and tested whether objects intersect with each other in the same space area or not. The above methods treated collision detection problem as computational geometry problem, but it was also treated as a mathematical optimization problem. Such as, Distance tracking algorithm. It included hierarchical level algorithm based on hierarchical data structure and distance calculation algorithm. H-Walk et al[10] proposed H-Walk algorithm. It was a hierarchical algorithm based on Dobkin-Kirkpatrick hierarchical data structure. The algorithm was simple to computer, convenience manipulate. But its efficiency and robustness was not very good. It was from this that we derived Lin-Canny algorithm[11]. Lin-Canny algorithm was the earliest distance calculation algorithm. Subsequently, many improved algorithms[12, 13] were proposed based on the Lin-Candy algorithm for enhancing its robustness and efficiency. Although many methods worked well, they still could not meet people's expectances completely. Ming et al[14, 15] used a linear programming to depict the collision detection problem, but king et al[16, 17] converted it into nonlinear programming. SI Vyatkin et al[18] pointed out a perturbation function method in literature [18]. The above methods did not depend on the relative position of the object being detected, but depended on the bounding to the being detected object which has some limits to the algorithm.

To better solve the collision detection, this paper presents an Improved Quantum Particle Swarm Optimization algorithm (IQPSO), which transforms the collision detection problem into constrained nonlinear optimization problem, then uses IQPSO algorithm to solve the nonlinear problem. The method in this article is simple, easy to implement. In addition, there are fewer parameters need to be adapted.

The rest of the paper is organized as follows: section two introduces the distance model. Section three describes the improved quantum particle swarm algorithm. Section four shows the experiment and results anises. In the end, section five draws the conclusions and future work.

2. Quantum Particle Swarm optimization Algorithm (QPSO)

Quantum particle swarm optimization (QPSO) algorithm is a kind of particle swarm algorithm based on the principles of quantum computing [19]. According to the characteristics of quantum entanglement and probability amplitude, a quantum bit can be represented not just $|0\rangle$ or $|1\rangle$, but also a superposition of the two[20], in which proportions of zero-ness and one-ness are combined in a single state. That is $|\phi\rangle = k|0\rangle + l|1\rangle$ (where $|k|^2 + |l|^2 = 1$). Probability amplitude k and l are seemed as position of each particle in search spaces. It could avoid the random of measure and decoding process which changes from binary to decimal.

In QPSO, solution of the optimization problem is presented by the qubit probability amplitude. The movement of particle is achieved by quantum revolving door. The variation of particle is performed by quantum gate. Then, we will give details of the specific operations in QPSO.

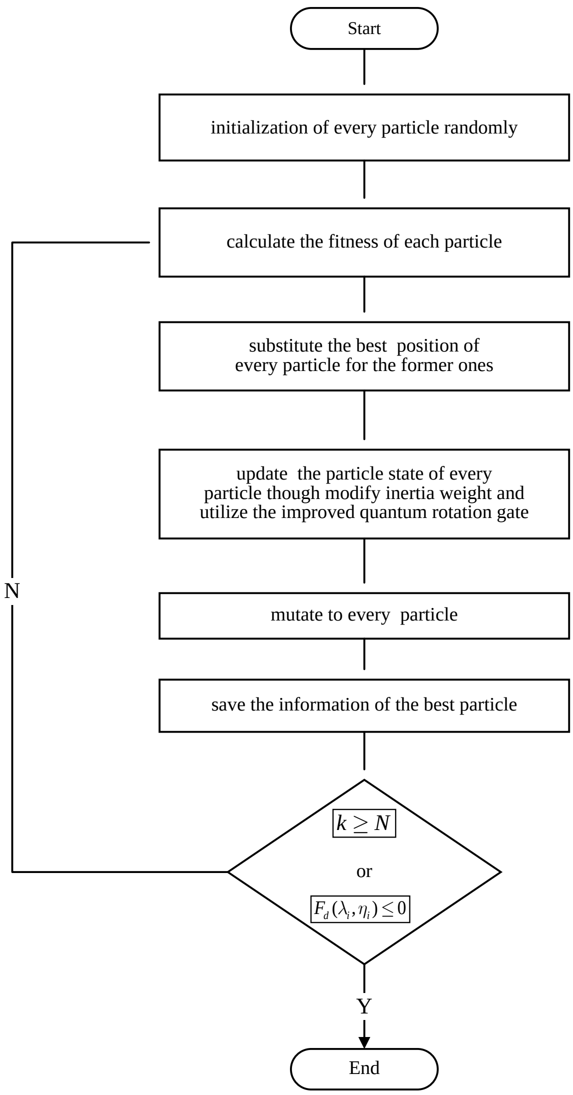

Fig.1 The flowchart of QPSO

Step1: initialize the particle swarm randomly. QPSO adopt the method of using qubit probability amplitude as particle current location coding. Considering the coding random during initialization, we use the following encoding scheme.

$$q_i = \begin{bmatrix} \begin{vmatrix} \cos(\varphi_{i1}) \\ \sin(\varphi_{i1}) \end{vmatrix} \begin{matrix} \cos(\varphi_{i2}) \\ \sin(\varphi_{i2}) \end{matrix} \begin{vmatrix} \cdots \\ \cdots \end{vmatrix} \begin{matrix} \cos(\varphi_{in}) \\ \sin(\varphi_{in}) \end{matrix} \end{bmatrix} \tag{1}$$

where $\varphi_{i1} = 2\pi \times Rnd$; Rnd is a random in $(0,1)$; $i = 1, 2, \cdots, m; j = 1, 2, \cdots, n$; m is the population size , n is dimension of space. Hence, the population of each particle occupied two positions traversing the space, these two positions respectively corresponding to the probability amplitude of the quantum states.

$$p_{ic} = \left(\cos(\varphi_{i1}), \cos(\varphi_{i2}), \cdots \cos(\varphi_{in})\right) \tag{2}$$

$$p_{is} = \left(\sin\left(\varphi_{i1}\right), \sin\left(\varphi_{i2}\right), \cdots \sin\left(\varphi_{in}\right)\right) \quad (3)$$

Step2: Transform the resolution space and calculate the fitness of particles. We compared the current position of the particle with the current optimal position, if the current position of the particle was better than the current best position, then substitute the current position for the current optimal position; if the current global optimum position was better than the searched global optimal location, then replace the global optimal location with current global optimum position. The fitness of two objects was defined below:

Through using convex hull to represent convex polyhedron, the solution of the shortest distance between convex polyhedral could be transformed into a nonlinear optimization problem with constraints. The fitness is the distance value between two convex hulls. The value was computed by distance model which described as follow:

Definition 1：a given linear function

$$f(X_1, X_2, ...X_n) = \lambda_1 X_1 + \lambda_2 X_2 + ... + \lambda_n X_n \quad (4)$$

where $X_1, X_2, ...X_n \in R^n$, λ_i are real number, R^n is the n dimensional space. When $\lambda_1 + \lambda_2 + ... + \lambda_n = 1$ and $\lambda_1, \lambda_2, ..., \lambda_n \geq 0$, $f(X_1, X_2, ...X_n)$ is the convex combination of $X_1, X_2, ...X_n$.

Definition 2 : if $A \in R^n$, all of convex combination which composed of a finite number of arbitrary points of A is convex polytop, written as:

$$H(A) = \left\{ \lambda_i \geq 0, i = 1, 2, ..., n, \sum_{i=1}^{n} \lambda_i = 1, n \in N^+ \right\} \quad (5)$$

Convex body had the following properties[24]: the distance between two points on different convex bodies is not only a local minimum, but also the global minimum. But concave body objects did not meet the feature. So this article gave two polyhedral named A and B, assumed that A and B in the same reference coordinate system. The formulation of calculating the shortest distance between A and B was:

$$\text{distance}(A, B) = \min\left\{\|a - b\| : a \in A, b \in B\right\} \quad (6)$$

where a is a point on the convex polyhedron A and satisfies $a = \sum_{i=1}^{n} \lambda_i x_i$, b is a point on the convex polyhedron B and satisfies $b = \sum_{i=1}^{n} \eta_i y_i$. λ_i satisfy $\sum_{i=1}^{n} \lambda_i = 1$, $\lambda_i \geq 0$ ($i = 1, 2, ..., n$). η_i satisfy with $\sum_{i=1}^{n} \eta_i = 1$, $\eta_i \geq 0$ ($i = 1, 2, ..., n$.). So the fitness is:

$$F_d(\lambda_i, \eta_i) = \sqrt{(\lambda_i x_{i1} - \eta_i y_{j1})^2 + (\lambda_i x_{i2} - \eta_i y_{j2})^2 + (\lambda_i x_{i3} - \eta_i y_{j3})^2} \quad (7)$$

where any point p on convex polyhedron A and any point q on convex polyhedron B whose coordinates are $p(x_{i1}, x_{i2}, x_{i3})$, $q(y_{j1}, y_{j2}, y_{j3})$, $i = 1, 2, ..., m$ and $j = 1, 2, ..., n$. In addition, λ_i and η_i are the Lagrange multiplier of PSO algorithm. If $F_d(\lambda_i, \eta_i) \leq 0$, then convex polyhedral A and B collide, and vice versa.

Step3: Updating the particle state. We used the inertia weight to change amplitude angle of quantum bit and utilized the quantum rotation gate to update quantum bit quantum probability amplitude. Quantum revolving door is

$$G = \begin{bmatrix} \cos(\Delta\varphi) & -\sin(\Delta\varphi) \\ \sin(\Delta\varphi) & \cos(\Delta\varphi) \end{bmatrix} \quad (8)$$

Where $\Delta\varphi$ is the quantum incremental phase, Quantum probability amplitude is varied by changing the quantum phase which realize by quantum revolving door.

Step4: According to the mutation probability of each particle, we achieved mutation operation. The mutation probability amplitude is converting two probability amplitude, which is achieved through a quantum non-gate. Its realization is as following

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \quad (9)$$

where α, β are probability amplitude.

Step5: If the fitness value is less than or equal to zero, then output the value of the global optimum and end the program. Otherwise, go back to step 2 loop calculations until it reaches the maximum number of iterations (N) or to meet the convergence condition.

3. Improved Quantum Particle Swarm Optimization Algorithm (IQPSO)

By changing the quantum phase, IQPSO algorithm can change the position and velocity of a particle at the same time by quantum rotation gate [21]. Then IQPSO algorithm will execute the mutation operation by quantum non gate. Among them, the update of quantum spin angular is equal to the displacement of particles in ordinary PSO algorithm commonly; the update of quantum bit probability amplitude is equal to the updated position of ordinary PSO particle.

In IQPSO algorithm, different search spaces make algorithm to attend different solution spaces, but different inertia weights lead particles to move in different search space. In the case of constant population, quantum coding could extend the ergodicity of search, but not change its search space, which is not conducive to improve the efficiency of the optimization algorithm. Quantum rotation gate just changes the phase quantum bit, without changing the length of the quantum bits. It is not helpful to escape from local optimal value for the algorithm. Quantum phase is limited to a smaller range by quantum non gate, so that the collision detection algorithm could not achieve the optimal solution. Quantum H_ε gate [22] limits the quantum probability amplitude in a certain range, so algorithm could find the optimal value easily. Quantum hadamard gate [23] get the $|\phi\rangle$ to rotate around the y-axis by 90^o, then get the $|\phi\rangle$ to rotate around the x-axis by 180^o. In that way, we increased the diversity of population with the locked particle. And for speeding up the progress of the collision detection algorithm, we introduced the following three improved aspects in the algorithm: the choice of inertia weight, the update strategy of quantum probability amplitude, selection of aberrance operation. Fig.1 is the whole flowchart of our algorithm. In it, K is the number of iterations; N is the maximum number of iterations. $F_d(\lambda_i, \eta_i)$ is the fitness. The details of the algorithm description are as follows:

3.1 The choice of inertia weight
In QPSO algorithm, as inertia weight become smaller, the density and the search space of particles, which hold different speed, decrease responding that lead to particles are more likely to fall into the local optimum. So that inertia weight should be changed following search space. Its expression is as follows:

$$\omega = \begin{cases} \omega_{\min} - \dfrac{(\omega_{\max} - \omega_{\min}) * (f_i - f_{\min})}{(f_{avg} - f_{\min})}, & f_i \le f_{avg} \\ \omega_{\max} & , f_i \le f_{avg} \end{cases} \tag{10}$$

where $\omega_{\max}$ is the maximum weight, $\omega_{\min}$ is the lightest weight, f is the fitness function, $f_{\min}$ is the minimum fitness value, f_{avg} is the average fitness value.

3.2 The update strategy of quantum probability amplitude
The operation of quantum rotation gate mainly converts the probability amplitude of each quantum bit to 0 or 1, then searched for the optimal solution. It varied according to the rotation angle, that is to say, the amplitude of rotation angle affects the convergence speed. If the amplitude was too large, it would result to premature convergence; On the contrary, it would builddown rate of convergence. The quantum H_ε gate could converge the probability amplitude to $\sqrt{1-\varepsilon}$ or $\sqrt{\varepsilon}$, instead of 0 or 1. Therefore, this paper used quantum rotation gate and quantum H_ε gate to perform quantum bit probability updates. We took $[p'' \quad q'']^T$ as the changed quantum probability amplitude. Its update rules are as follows:

(1) If $|q'| \ge 1 - \varepsilon$ and $|q'| \ge 1 - \varepsilon$, then:

$$[p'' \quad q'']^T = \left[\sqrt{\varepsilon} \quad \sqrt{1-\varepsilon}\right]^T. \tag{11}$$

(2) If $|p'|^2 \ge 1 - \varepsilon$ and $|q'|^2 \le \varepsilon$, then:

$$[p'' \quad q'']^T = \left[\sqrt{1-\varepsilon} \quad \sqrt{\varepsilon}\right]^T. \tag{12}$$

(3) Or else

$$\begin{aligned}[p'' \quad q'']^T &= \begin{bmatrix} \cos(\Delta\alpha_{ij}(t+1)) & -\sin(\Delta\alpha_{ij}(t+1)) \\ \sin(\Delta\alpha_{ij}(t+1)) & \cos(\Delta\alpha_{ij}(t+1)) \end{bmatrix} \begin{bmatrix} \cos(\alpha_{ij}(t)) \\ \sin(\alpha_{ij}(t)) \end{bmatrix} \\ &= \begin{bmatrix} \cos(\alpha_{ij}(t) + \Delta\alpha_{ij}(t+1)) \\ \sin(\alpha_{ij}(t) + \Delta\alpha_{ij}(t+1)) \end{bmatrix}. \end{aligned} \tag{13}$$

where $p' = \cos(\alpha_{ij}(t) + \Delta\alpha_{ij}(t+1))$, $q' = \sin(\alpha_{ij}(t) + \Delta\alpha_{ij}(t+1))$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ and $0 \le \varepsilon \le 1$. ε has a great impact in the algorithm. According to the study [25], we took

$\varepsilon=0.01$. Then the algorithm of this paper avoided the premature convergence phenomenon, achieves the global convergence of the algorithm.

3.3 The selection of mutation operator.

Although QPSO algorithm enlarged the search space of quantum particle [26], the diversity of the population was still easy to lose. In order to increase the diversity of population, this paper joined the QPSO algorithm into the mutation operator while improving quantum rotation gate. In our method, the hadamard gate was employed, variation procedure is as follows:

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}\begin{bmatrix}\cos(\alpha_{ij})\\ \sin(\alpha_{ij})\end{bmatrix}=\begin{bmatrix}\cos(\frac{\pi}{4}+\alpha_{ij})\\ \sin(\frac{\pi}{4}+\alpha_{ij})\end{bmatrix}. \tag{14}$$

where $i=1,2,...,m$ and $j=1,2,...,n$.

We took mutation probability to p_m , set random number r_i for each particle and $r_i \in (0,1)$. If $r_i < p_m$, we selected $\lceil n/2 \rceil$ of quantum bits on particles randomly, the optimal position of particle memory and rotation vector stayed the same. The mutation operator was actually on quantum bits of rotation angle, such as given an angle of quantum bit amplitude α and the final value was $\frac{\pi}{4}+\alpha$, which made the next step with regard to the last step.

4. Experiment and result analysis

The hardware condition of test is: Intel (R) Core (TM) CPU E8400, 2.0GHz, 2GB RAM, AMD radeon HD6450 graphics card, XP system. We tested in Matlab2010a. Tests included: different scenes of the performance in QPSO algorithm only with changed inertia weight (W), the update strategy of quantum probability amplitude (P), mutation operator (M). Different scenes of the performance in the same algorithm, the same scene performance comparison of different algorithms. Test results are shown as follows. The following figures show the collision effects in all scenes. One pair is demonstrated in the following figure in each scene.

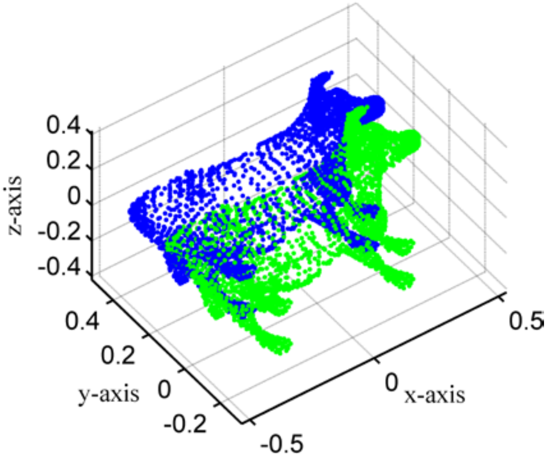

Fig.2 Collection in scene 1

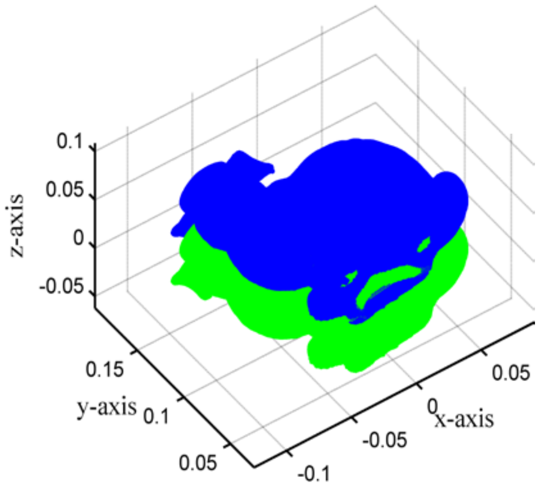

Fig.3 Collection in scene 2

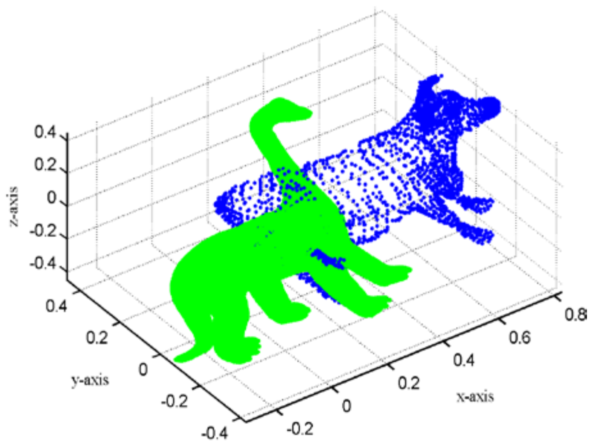

Fig.4 Collection in scene 3

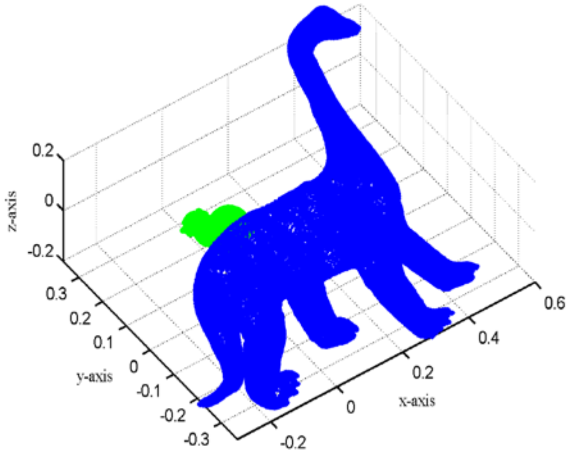

Fig.5 Collection in scene 4

The experimental data of scene 1-4 is obtained through scanning. Mat file, the data of object C is Stanford bunny which was download on the web (http://graphics. stanford.edu/data/3Dscanrep /). We listed all points of scene1-6 in Table 1.

Table 1 Points of objects in each scene

	1	2	3	4
Points of A	2904	35947	2904	23982
Points of B	2904	35947	23982	35947

For IQPSO, the search results were not only affected by the size of search space, but also influenced by the parameters of algorithms, such as the size of initiation, mutation probability. In order to achieve the better results, this article set parameters of the QPSO and IQPSO algorithms based on empirical analysis. In the others, we set the size of individual population as 50, quantum probability amplitude and aberrance operation should be update respectively. In QPSO, the mutation probability is 0.05. In IQPSO algorithm, inertia weight, quantum probability amplitude and aberrance operation should be update with Eqs.(10)-(14) respectively. All seniors ran 20 times for each other. The results were as follows:

Experiment 1: different scenes of the performance in QPSO algorithm only with changed inertia weight (W), the update strategy of quantum probability amplitude (P), mutation operator (M). In QPSO, Any such improvement shall affect the convergence of the algorithm. Therefore the method of improvement was very important. We compared different improvement of the algorithm to the QPSO. (Where IQPSO(w) is the changed with mutation probability of QPSO only, IQPSO(P) is that updating strategy of quantum probability amplitude of QPSO only, IQPSO(M) represents the changed of mutation operator in QPSO. F is the fitness value. T is the mean time needed to run the algorithm once, gen is the number of iterations.).The results were as follows:

Table 2 different inertia weight（W）in different scene

scene	T(ms)		F(mm)	
algorithm	QPSO	IQPSO(w)	QPSO	IQPSO(w)
1	0.062	0.053	0.107	0.094
2	0.078	0.007	0.351	0.125
3	0.093	0.023	0.631	0.304
4	0.109	0.078	0.781	0.519

Table 3 different strategy of quantum probability amplitude（P）in different scene

scene	T(ms)		F(mm)	
algorithm	QPSO	IQPSO(P)	QPSO	IQPSO(P)
1	0.062	0.046	0.107	0.003
2	0.078	0.071	0.351	0.011
3	0.093	0.101	0.631	0.053
4	0.109	0.087	0.781	0.147

The results of Table 3 show that IQPSO(P) is better than QPSO in computing time. the update strategy of quantum probability amplitude expand the search area, so the algorithm is able to search more optimal solution to some extent, seek more quality solutions expanding the search space. So it is better to maintain the diversity of population, avoid the effects of premature convergence.

Table 4 different mutation operator (M) in different scene

scene	T(ms)		F(mm)	
algorithm	QPSO	IQPSO(w)	QPSO	IQPSO(w)
1	0.062	0.056	0.107	0.005
2	0.078	0.062	0.351	0.031
3	0.093	0.085	0.631	0.412
4	0.109	0.110	0.781	0.613

From the Table 2-4, for the simulation results, the improve algorithm have a little excellent in average time and fitness value. IQPSO algorithm sometimes gets a small part of the solution which is not completely superior to the solution of the other two algorithms, but most solutions are stable and superior to the other two algorithms. To observe the effect of the method in this paper, we used punition function (PF) [26], QPSO algorithm and IQPSO algorithm to compare.

Table 5 comparing the results of three algorithms

scene	T(ms)			F(mm)		
algorithm	PF	QPSO	IQPSO	PF	QPSO	IQPSO
1	13.128	0.062	0.006	21.495	0.107	0.001
2	31.505	0.078	0.053	27.945	0.351	0.006
3	39.094	0.093	0.097	32.001	0.631	0.031
4	57.836	0.109	0.103	34.333	0.781	0.143

Shown in Table 5, IQPSO algorithm is better than publish functioning from average result and mean time. For IQPSO algorithm and QPSO algorithm, the former is better than the latter in average iterations and mean time. But to the seniors of all algorithms, time-consuming performance of IQPSO algorithm is superior to the others. That is to say, IQPSO algorithm raised the convergence speed of the particle distinctly. This suggests that quantum coding could simplify calculations and accelerate the speed of calculation. Also, it clearly shows that sometimes IQPSO algorithm gets a small part of the solution which is not completely superior to the other two algorithms, but most solutions are stable solution and superior to the other two algorithms. The experimental results shows that the IQPSO algorithm either in the stability of the algorithm or in terms of convergence than the QPSO algorithm had varying degrees of improved. In all scenarios, IQPSO algorithm has less convergence time than the QPSO algorithm. IQPSO algorithm reduces the number of calculations in the absence of no increase new arguments under control. Improved algorithm not only maintains local search capabilities of QPSO, but also improves the global search capability, the search space for further expansion. In addition, the algorithm in this paper broadened the search space of the particle, found more suitable fitness. It is consistent with the results in Table 5.

4. Conclusion and future work

This paper presented a random collision detection algorithm based on distance calculation. We used probability amplitude of quantum encoding, according the properties of quantum. The quantum bits described the position and velocity of the particle phase. Changing quantum bits, both position and velocity of the particle were changed simultaneity. This was not only give full utilize the computing quantum, but also simplifies the calculation. In addition, the performance of the algorithm and detection quality was all increased.

We can improve the performance of the algorithm that provided above from two aspects. On the one hand, we will discuss affects of various parameters of the algorithm that act on the convergence speed of the algorithm. On the other hand, the performances of the algorithm in vinous models, beyond the model that is utilized on this article will be analyzed in the future.

References:

[1] YM Wei, YQ Wu, JY Shi, Research on fixed direction hull bounding volume in collision detection. Journal of Software, 12,7, 2001,pp. 1056-1063.

[2] James H Clark, Hierarchical geometric models for visible surface algorithms. Communications of the ACM, 19,10, 1976, ,pp. 547-554.

[3] Matthias Teschner, Bruno Heidelberger, Matthias Müller, et al, Optimized spatial hashing for collision detection of deformable objects. Proceedings of Vision, Modeling, Visualization VMV’03, 2003,pp.47-54.

[4] Gino van den Bergen, Efficient collision detection of complex deformable models using AABB trees. Journal of Graphics Tools, 2,4,1997, pp.1-13.

[5] Philip M Hubbard, Approximating polyhedra with spheres for time-critical collision detection. ACM Transactions on Graphics, 15,3,1996, pp.179-210.

[6] Brian Johnson, Ben Shneiderman, Tree-maps: A space-filling approach to the visualization of hierarchical information structures. Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on, 1991,pp.284-291.

[7] MC Lin, D. Manocha, J. Cohen and S. Gottschalk, Collision detection: Algorithms and applications, 1996.

[8] Gabriel Zachmann, Rapid collision detection by dynamically aligned DOP-trees. Virtual Reality Annual International

Symposium, 1998. Proceedings., IEEE 1998, 1998,pp.90-97.

[9] Rong Guo Zhang, Zhi Fang Wang, Xiao Jun Liu, et al, Objects Collision Detection in Virtual Scene. Advanced Materials Research, 605,2013, pp,2391-2394.

[10] Leonidas J Guibas, David Hsu, Li Zhang, H-Walk: Hierarchical distance computation for moving convex bodies. Proceedings of the fifteenth annual symposium on Computational geometry, 1999,pp, 265-273.

[11] Ming C Lin, Dinesh Manocha, Fast interference detection between geometric models. The Visual Computer, 11,10,1995,pp,542-561.

[12] Brian Mirtich. V-Clip: Fast and robust polyhedral collision detection. ACM Transactions on Graphics, 17,3, 1998, 177-208.

[13] Vasyl Tereshchenko, Sergii Chevokin, Andriy Fisunenko, Algorithm for Finding the Domain Intersection of a Set of Polytopes. Procedia Computer Science, 18,2013,pp, 459-464.

[14] Ming Lin, Stefan Gottschalk. Collision detection between geometric models: A survey. Proc. of IMA Conference on Mathematics of Surfaces, 1998,pp. 602-608.

[15] L Zeng, T Ning, P Xi. Research and Application of General Collision Detection Simulation Platform. The 19th International Conference on Industrial Engineering and Engineering Management, 2013,pp.1315-1323.

[16] Hanjun Jin, Yanlin Wang, Xiaorong Wang, et al, An Algorithm on Collision Detection by Computing the Minimum Distance between Two Convex Polyhedra. Intelligent Control and Automation. WCICA 2006. The Sixth World Congress on,2006,pp. 9425-9428.

[17] Liu Hui, Jin Hanjun, An algorithm for computing the minimum distance between two convex polyhedra in three-dimensional space. Knowledge Acquisition and Modeling, 2008. KAM'08. International Symposium on, 2008:,pp.313-317

[18] Sergey I Vyatkin, Boris S Dolgovesov, Mikhail A Gorodilov. Perturbation Functions in Computer Graphics. Modern Instrumentation, 2,2013, pp.26-32.

[19] Shuyuan Yang, Min Wang and Licheng Jiao, A quantum particle swarm optimization. IEEE Congress on Evolutionary Computation, 2004, pp.320-324.

[20] Shiyong L, Quantum particle swarms algorithm for continuous space optimization. Chinese Journal of Quantum Electronics, 24,5,2007,pp. 569.

[21] Bing Han, Junna Jiang, Yanhui Gao, et al,A Quantum Genetic Algorithm to Solve the Problem of Multivariate.Information Computing and Applications, Springer, 2011.

[22] Kristiaan De Greve,All-Optical Hadamard Gate: Direct Implementation of a Quantum Information Primitive.Towards Solid-State Quantum Repeaters, Springer, 2013.

[23] In-Won Park, Ki-Baek Lee, Jong-Hwan Kim, Multi-objective evolutionary algorithm-based optimal posture control of humanoid robots. Evolutionary Computation, 2012 IEEE Congress on, 2012,pp. 1-7.

[24] Christer Ericson, Real-time collision detection. Morgan Kaufmann, 2004.

[25] Kuk-Hyun Han, Jong-Hwan Kim. Quantum-inspired evolutionary algorithms with a new termination criterion, H gate, and two-phase scheme. IEEE Transactions on Evolutionary Computation, 8,2, 2004, 156-169.

[26] Hongbin Wang. Liyi Zhang,A Kind of New Optimization Lagrange Neural Network Principle and Implementation. Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on, 2009,pp,689-692.