Hindawi
Mobile Information Systems
Volume 2023, Article ID 9756205, 1 page
https://doi.org/10.1155/2023/9756205

Retraction

Retracted: Real-Time Collision Detection Optimization Algorithm Based on Snake Model in the Field of Big Data

Mobile Information Systems

Received 5 December 2023; Accepted 5 December 2023; Published 6 December 2023

Copyright © 2023 Mobile Information Systems. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] S. Cheng and H. Qu, "Real-Time Collision Detection Optimization Algorithm Based on Snake Model in the Field of Big Data," *Mobile Information Systems*, vol. 2023, Article ID 4960900, 12 pages, 2023.

Hindawi
Mobile Information Systems
Volume 2023, Article ID 4960900, 12 pages
https://doi.org/10.1155/2023/4960900

Research Article

Real-Time Collision Detection Optimization Algorithm Based on Snake Model in the Field of Big Data

Sining Cheng[1] **and Huiyan Qu**[2]

[1]*TJ-YZ School of Network Science, Haikou University of Economics, Haikou 571127, Hainan, China*
[2]*School of Information Technology, Jilin Agricultural University, Changchun 130118, Jilin, China*

Correspondence should be addressed to Huiyan Qu; huiyanq@jlau.edu.cn

Received 12 April 2022; Revised 15 June 2022; Accepted 27 July 2022; Published 20 April 2023

Academic Editor: Yang Gao

Copyright © 2023 Sining Cheng and Huiyan Qu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Big data processing includes multiple processing flows of data. But data quality is the most important part of the entire process. Every data processing link will have an impact on the quality of big data. Collision detection is an important research content in many fields such as computer graphics and computer virtual reality. In layman's terms, it means that the computer detects the signal voltage on the channel while sending data. When the signal voltage swing value detected by a station exceeds a certain threshold value, it will be considered that at least two stations on the bus are sending data at the same time, indicating that a collision has occurred. As the number of questions increases, the level becomes higher, the target information becomes more diverse, and the algorithm becomes more complex. The traditional evolutionary algorithm is far from being able to deal with this situation effectively, and the optimization target algorithm emerges as the times require. Evolutionary computing is a mature global optimization method with high robustness and wide applicability. It has the characteristics of self-organization, self-adaptation, and self-learning. It is not limited by the nature of the problem and can effectively deal with complex problems that are difficult to solve by traditional optimization algorithms. This paper aims to study the optimization algorithm of real-time collision detection based on Snake model in the field of big data. It is expected that, with the support of big data technology, the efficiency of real-time collision detection of Snake model will be improved and time will be saved. This paper proposes a multiline swarm particle swarm algorithm and combines it with the Snake model to improve the detection efficiency of the collision detection algorithm. It verifies the detection performance of traditional algorithms and tests their effectiveness in detection. The experimental results of this paper show that the frame rate of the Snake model algorithm is 15, the frame rate of the K-DOPs algorithm is 6.7, and the error of the algorithm is 1.04. It shows that the frame rate of Snake model algorithm is better.

RETRACTED

1. Introduction

With the deepening and development of economic globalization, the competition among countries is becoming more and more intense. In the final analysis, the competition between countries is still the competition of technology. Therefore, mastering high-end technology will greatly promote the country's international competitiveness. With the take-off of modern science and technology and the development of computer technology, the information resources in our life are increasing exponentially. All these bring us into the era of big data. Big data refers to the amount of data involved that is too large to be captured, managed, processed, and organized into information that can help companies make more active business decisions within a reasonable period of time through mainstream software tools. During the age of big digital data, the growth of all industries is inseparable from big data. The development of big data applications has an important impact on enterprises, governments, and individuals. At present, 3D models have become an unavoidable problem in virtual reality. Collision detection is also an important part of animation simulation. However, the amount of calculation in the 3D model is greatly increased, and how to simplify the calculation is a problem that needs to be explored at present.

This paper explores from the perspective of collision detection algorithm, improves its key technology, and finally achieves the goal of optimizing speed. Its fast and accurate collision detection plays a decisive role in improving the authenticity of virtual simulation scenes and enhancing the immersion of virtual scenes.

Aiming at the problem that the concave part of the traditional Snake model does not converge and is easily disturbed by noise, the centripetal energy and control energy are added to the model. It combines the multiline swarm particle swarm algorithm with the Snake model to improve the calculation process.

2. Related Work

As new and diverse IoT applications begin to emerge, handling streaming big data becomes critical. Liu et al. proposes a broker architecture for cloud environments to serve big data computing in a cost effective manner. The cloud service broker leases cloud services from multiple cloud providers with multiple service interfaces. A cloud environment refers to an Internet big data environment that can provide users or systems with computing power, storage capacity, or virtual machine services from dynamically virtualized resources. He also proposes a pricing strategy to maximize the revenue of multiple cloud intermediaries. Extensive simulations show that this pricing strategy results in higher revenue than other pricing methods [1]. Cloud mining removes the requirement of specialized storage area, and it only calls for massive calculation infrastructure to analyze and process data. Noraziah introduces big data handling in two dimensions: systematic and definite usage. It also presents the critical concerns of big data from the perspective of managing cloud data and big data handling mechanisms [2]. Lin proposed the serpentine model approach. In this method, map characteristics are transformed into agents. The quantitative properties of the attributes are controlled by the secondary associative agents. A corresponding task is given to each agent, and they all cooperate with one another to achieve ubiquity. Our experimental findings indicate that this approach can achieve good results [3]. Sha proposed the animated Snake phantom, which ensures the overall stability. It deals with the issue that the snake's shaft was often inhomogeneously distributed all over the silhouette line, thus achieving a nice localization of the parameters. The kinetic stencil solution intensely honors the morphological agility over the shape of the model, while preserving a respectable general roof structure for the snake. Experiments show that the algorithm can effectively prevent the Snake from self-intersecting or automatically decouple the self-intersecting contour lines [4]. Zhuo proposed a hybrid Gradient Vectored Flow Pattern on the basis of the hybrid gravity flow pattern. Since the characteristics differ among the LV apical and extracardiac membranes, these data streams are derived from the original cardiac TMR images separately. The average overlap of endocardial segmentation was 89.67%. The average degree of overlapping for epicardial segments was 95.88%. The experimental findings indicate that this approach effectively enhances the performance of fragmentation than some existing approaches [5]. To address the problem that particle filters cannot obtain accurate hand information, Yi proposes an algorithm combining the filter vector flow Snake model. It guides the rapid narrowing of the profiles to the profound recessed region of the palm profile and obtains the sophisticated palm profile precisely. The resulting method achieves online modification of the parameters of the chip element filtration and avoids the chip element float phenomenon. Test leads to the indication that the suggested arrangement can lower the root-mean-square estimation mistake of manual contour following by 53% [6]. Qu proposes an optimization operator-based collision detection algorithm. In finding the multidimensional community, a task-balancing principle is applied to allocate subtasks to be executed in advance to each core of multiple-core CPUs for concurrent implementation. It is demonstrated that the method provides superior capability in tackling the problem of man-machine communication in crash inspection compared to typical ICOLLIDE method [7]. In the process of human-robot collaboration and service robots, maintaining safety and reliability is very important. Lou introduced a new strain gauge knuckle moment load cell prototype. It has a low profile construction, fast reaction, and great flexibility. He proposed a perforated speaker construction to transfer the location of the largest force and to enhance the force allocation. The curved spoke structure is topologically optimized, thus obtaining the best structural parameters [8]. Although these theories discuss big data and real-time collision detection, the combination of the two is less and it is not practical.

3. Real-Time Collision Detection Optimization Algorithm Method Based on Snake Model in the Field of Big Data

3.1. Overview of Collision Detection. Collision refers to the contact between objects. The task of collision detection is to determine whether there is a coincidence between two target objects in a certain area. If the two are found to have overlapping parts, it is called a collision. If there is no coincidence between the two, then the target object does not collide [9, 10]. Collision detection is to detect whether there is a collision between two or more objects in the virtual courseware, that is, contact or penetration. This part of the content has been supplemented in the article. Not only can collision detection technology be used in two-dimensional space, but also, with the development of computer technology, it is widely used in many virtual scenes [11]. Figure 1 shows the algorithm classification of collision detection.

According to research, most of the current algorithm detection uses rigid object collision [12]. Because the morphological structure of the deformed object is complex and prone to change, it needs to perform collision update and collision response after each collision. Therefore, the collision efficiency of other objects is bound to be affected relatively to rigid objects. In order to meet market changes, people want to simulate the display world. Most important

RETRACTED

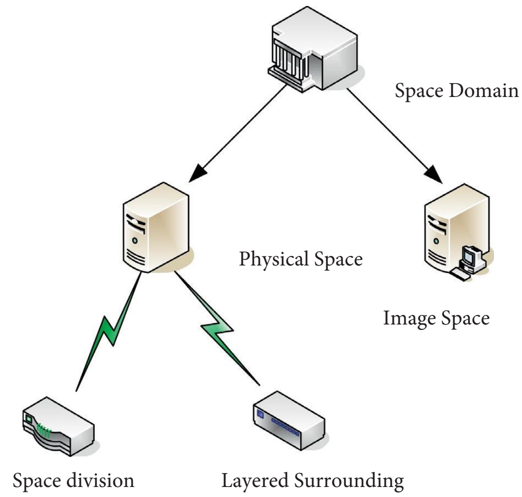

Figure 1: Classification of algorithms for collision detection.

in this technology is real-time collision detection. Although the traditional detection method can also be simulated, when the complexity of the objects designed in the scene is relatively high, the traditional detection cannot meet the real-time requirements of the virtual system. With the promotion of Internet technology, personal computers have become very common, and virtual simulation technology has also begun to be used in all walks of life [13, 14]. For example, in the development of RPS online games, developers need to use simulation to test the smoothness of the game. These scenarios require fast modeling of a large number of deformable complex objects and very efficient collision handling techniques. Although all countries in the world have explored collision detection technology and achieved results in this field, collision detection technology is still at the primary level of exploration, and there are still many unknowns [15, 16].

3.2. Overview of Algorithm Optimization. Computers have been developing and progressing continuously since their appearance in 1946 and have been fully used in various fields of social production. As the productive culture keeps improving, the scale of computation involved is increasing, especially in the fields of management and engineering [17]. With the constant evolution of PC skills, and in order to reduce the complexity of the calculation, the optimization calculation came into being. In essence, optimization refers to the use of certain rules to meet the needs of users [18, 19]. Although the development is very rapid, the production needs of human beings are also expanding. The traditional computing performance cannot meet the current development needs, so more convenient algorithms emerge as the times require. With the increasing demand for computing power, high-performance computing is also constantly developing [20, 21]. High-performance computing clusters are mainly used to deal with complex computing problems and are widely used in meteorological, oceanographic, and environmental fields (numerical forecasting, etc.), life science fields (gene sequencing, alignment, homology analysis, etc.), and computer-aided engineering fields. Figure 2 shows the basic flow of optimization calculation.

Bayesian theory belongs to the statistical theory of systems; this entails the ability to

$$\begin{aligned} Q_a \cap Q_z &= \alpha (a \neq z), \\ Q_a \cup Q_z \cup \cdots \cup Q_n &= W, \\ R(Q_a) > 0 \, (a &= 1, 2, \ldots, n). \end{aligned} \tag{1}$$

$Q_a \cup Q_z \cup \cdots \cup Q_n = W, \quad R(Q_a) > 0 \, (a = 1, 2, \cdots, n)$where $Q_1, Q_2, \ldots, Q_n$ are on behalf of the paramount expression; the following formula can be obtained:

$$T(Q_a|U) = \frac{T(U|Q_a)T(Q_a)}{\sum_{z=1}^{n} T(U|Q_a)T(Q_a)}. \tag{2}$$

In practical applications, spatial splits are considered to be standalone of each other. What if the decision of the market is $Q_1, Q_2, \ldots, Q_n$, a posteriori probability which is available from the inspection probability and critical probability can be represented as shown in

$$T\left(Q_a|S_1 \Lambda S_2 \ldots \Lambda, S_m\right) = \frac{\prod_{i=1}^{m} T(S_i|Q_a)T(Q_a)}{\sum_{z=1}^{n} \prod_{i=1}^{m} T(S|Q_Z)T(Q_Z)}. \tag{3}$$

GC-MS is an analytical instrument. During the experiment, a set of data in chronological order can be obtained. During the experiment, the target object is composed of t parts, then each component is $1, 2, 3, \ldots, n$. The specific function expression is as follows:

$$\begin{aligned} W_1 &= W_{11}, W_{12}, \ldots W_{1M}, \\ W_2 &= W_{21}, W_{22}, \ldots W_{2M}, \\ W_N &= W_{N1}, W_{N2}, \ldots, W_{NM}. \end{aligned} \tag{4}$$

Among them, W stands for different components.

Introducing the above formula into a matrix, it can be simplified to

$$W = \begin{pmatrix} W_{11}, & W_{12}, & ,\cdots & W_{1M} \\ W_{21}, & W_{22}, & \cdots, & W_{2M} \\ \cdots, & \cdots, & \cdots, & \cdots \\ W_{N1}, & W_{N2}, & \cdots, & W_{NM} \end{pmatrix}. \tag{5}$$

Evolutionary algorithms are random search methods. Compared with other enumeration techniques and heuristic search techniques, the global optimal solution probability of this problem is high [22, 23]. The evaluation information of the objective function is used to make it actionable and general. It is concise in form, can be operated with massively parallel computers, and can be easily combined with other methods [24, 25]. At present, evolutionary computing mainly includes genetic algorithm, evolutionary strategy, and evolutionary planning [26, 27]. Genetic algorithm refers to the population of solution sets, and the population is composed of several individuals encoded by genes. Essentially, genetic algorithms are optimization algorithms. It is a random search algorithm that uses natural selection and biological evolution to search for the optimal solution in the search space [28].

RETRACTED

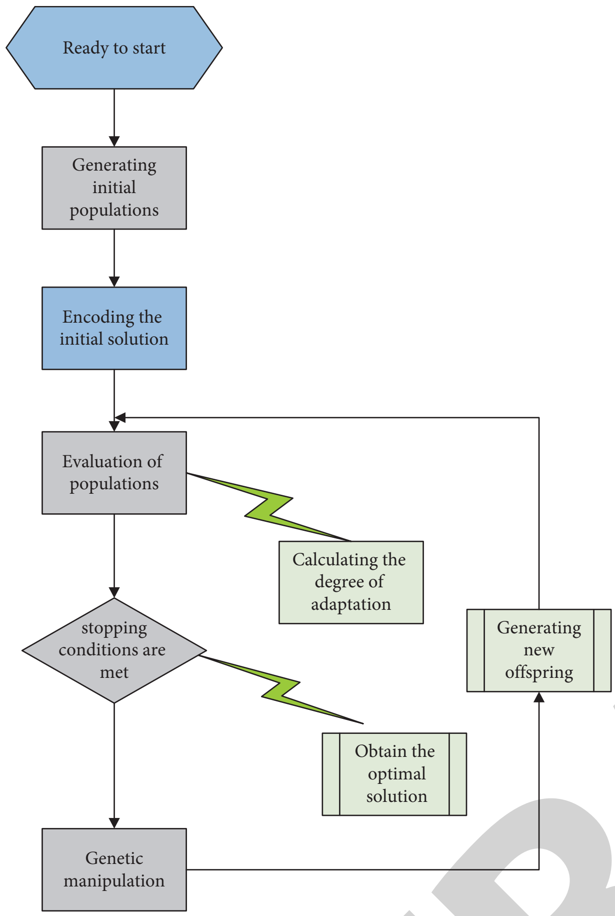

Figure 2: Basic framework of genetic algorithm.

$$\min\{g(a)|a \in Q\}. \quad (6)$$

Formula (6) reveals the function expression of the capabilities optimization model, where a represents the decision variable, $g(a)$ represents the objective capabilities, and Q represents the spatial subset.

$$\min g(a).\mathrm{st} * p(a) \geq 0. \quad (7)$$

Formula (7) represents the decision variable inequality.

$$Q = \{a \in W^s | p(a) \geq 0\}. \quad (8)$$

Formula (8) represents the feasible region of the decision solution, which is the set of all solutions in layman's terms. Figure 3 shows the basic flow structure of the algorithm.

To acquire the efficient method to solve the area, the model needs to be optimized.

$$\forall a \in Q \cap \{a \in W^s | \sqrt{\sum (a_1 - a^*)^2} < \alpha\}. \quad (9)$$

When formula (9) satisfies $g(a^*) \leq g(a)$, the model is said to have an optimal solution.

$$QW_j(c,r) = \frac{1}{\sqrt{c}} \int_{-\infty}^{+\infty} j(r)\chi^*\left(\frac{w-t}{c}\right)\mathrm{d}w. \quad (10)$$

Among them, $QW_j(c,r)$ is the wavelet coefficient, $\chi(w)$ is the wavelet basis function, and $j(r)$ is the analysis signal.

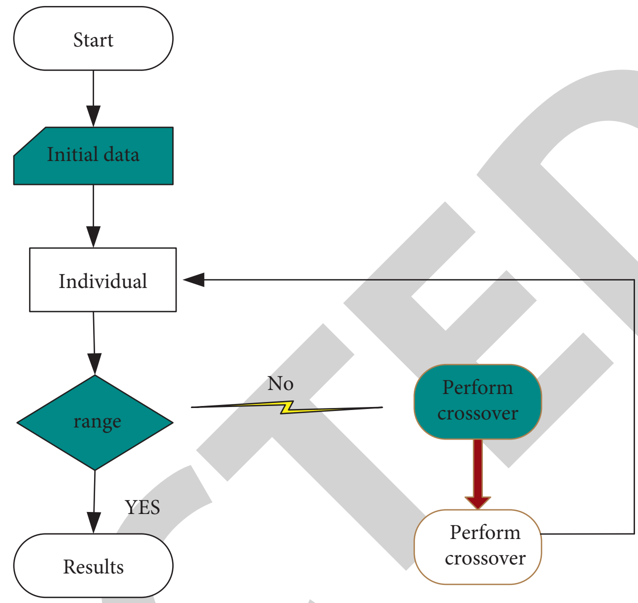

Figure 3: Model algorithm flow demonstration.

$$L_V = W\mathrm{diag}\left(s_{(v)}\right)D^U + Q_v, \quad v = 1, 2, \ldots, V. \quad (11)$$

Among them, L stands for cubic matrix, W stands for pure chromatogram, and D stands for pure mass spectrum.

In the genetic algorithm, the workout feature is usually employed to determine the condition of an individual, and the fitness function is used to meet the generality. Therefore, the calculation steps can be reduced in the actual use process.

$$\mathrm{Fitness}(g(a)) = g(a). \quad (12)$$

Formula (12) represents the functional expression for the maximal optimization problem.

$$\mathrm{Fitness}(g(a)) = -g(a). \quad (13)$$

Formula (13) represents the functional expression for the minimal optimization problem.

For multiobjective optimization problems, researchers have proposed different evolutionary multiobjective research algorithms. In the process of problem solving, there are two goals to be achieved. First, the frontier of the approximate Pareto optimal solution set should be as close as possible to the frontier of the true Pareto optimal solution set. The closeness of the two fronts reflects the convergence of the algorithm [29, 30]. Second, the target vectors corresponding to the Pareto optimal solution set should be distributed as widely and uniformly as possible. This goal reflects the diversity of knowledge sets. Many practical problems that can be solved by evolutionary algorithms are multiobjective problems of optimal performance. Multiobjective optimization can be described by the following formula.

If the Pareto boundary solution set can solve the problem, it means that the algorithm is feasible. Figure 4 is a structural diagram of data information transmission.

RETRACTED

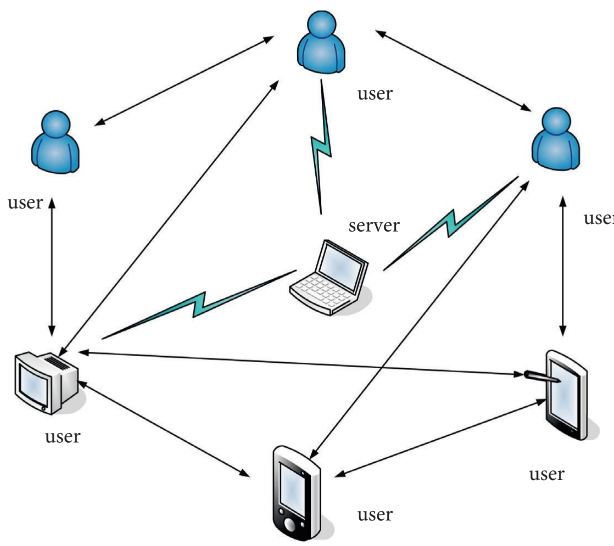

Figure 4: Data information transmission structure.

$$\begin{cases} \min\theta = G(i) = (g_1(i), g_2(i), \ldots, g_n(i))y, \\ sy.p_t(i) \le 0, \quad t = 1, 2 \ldots z, \\ t_e(i) = 0, \quad e = 1, 2 \ldots, q. \end{cases} \tag{14}$$

In the formula, $i = (i_1, i_2, \ldots, i_n)$ represents the decision vector of n dimension. $g(i)$ is the objective function. $p_t(i) \le 0, t = 1, 2 \ldots z$ defines inequality constraints for z. $t_e(i) = 0, e = 1, 2 \ldots, q$ defines the equality constraint function for q.

Traditional information representation has uncertainty, which can be expressed as

$$\begin{aligned} W(A) &= f(a_1)j(a_1) + f(a_2)j(a_2) + \cdots + f(a_l)j(a_l) \\ &= -\sum_{k}^{l} f(a_k)\log_2 F(a_k). \end{aligned} \tag{15}$$

When $f(a_1) = f(a_2)$, $W(A) = 1$.

$$j(f, k) = -\frac{f}{f+k}\log_2\frac{f}{f+k} - \frac{k}{f+k}\log_2\frac{k}{f+k}. \tag{16}$$

Formula (16) is intended to replace the quantity of energy available for the successful categorization of the product.

3.3. Overview of Big Data. Since big data has different functions in different fields, there is no uniform definition. With the continuous development of science and technology, the connotation of data has been expanded. It can not only be an exponential value, but also be used to represent a regular symbol, so the data is considered to be processed by certain encoding. Big data is a new way of thinking. The information of production processes is transformed into the form of data, which contributes to the socioeconomic and cultural development [31, 32]. Figure 5 shows a big data structure diagram.

Big data is a comprehensive technology. It involves processes such as data storage, data collection, data analysis,

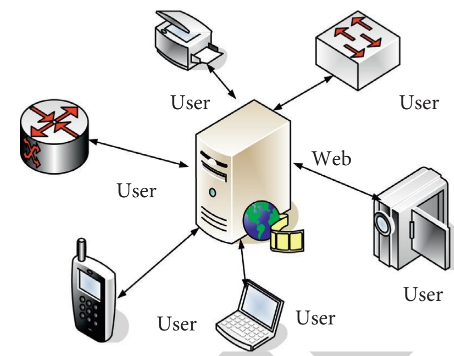

Figure 5: Big data structure diagram.

and data transformation. Companies that provide products or services to a large number of consumers can use big data for precise marketing. Small and microenterprises that are small and beautiful can use big data for service transformation. Some scholars believe that big data can be roughly divided into three stages of work [33]. Big data technology can be divided into four types, which cooperate with each other to realize information sharing. Figure 6 shows the information attribution structure in the big data processing flow.

4. Real-Time Collision Detection Optimization Algorithm Experiment of Snake Model

4.1. Experimental Data. As speed of scientific and artistic progress increases, computer skills can be utilized in a wide range of societies. Object collision detection between different dimensions is already very common. The task of collision detection is to determine whether the intersection of the geometric models of two objects is empty at a certain moment. In the actual calculation process, it needs to be analyzed according to the data. This is where an optimization algorithm is needed. To this end, the performance of the algorithm is briefly analyzed, and the specific situation is as follows.

From the experimental data in Table 1, it can be seen that comparing the values of ZDT1, ZDT2, and ZDT3, the algorithm NSGA-II is weaker than the algorithm UCMOEA in terms of the worst data, the mean value, or the best data. According to the data obtained by ZDT4, it can be seen that the algorithm NSGA-II is stronger and the algorithm UCMOEA is relatively weak. For the experimental data of ZDT6, it can be seen that the algorithm NSGA-II is better than the algorithm UCMOEA in the mean and worst cases. But the algorithm UCMOEA is stronger than the algorithm NSGA-II in the optimal data.

According to the data in Table 2, the storage capacity of the server used in this experiment is 16 TB and the memory is 64 GB. The network card standard is 2 Gbps, the CPU is 4 cores, and the hard disk is 2 TB.

4.2. Collision Algorithm Detection. Collision phenomena are very common, and there are many kinds of optimization algorithms. In order to verify the most suitable optimization algorithm, the performance of different algorithms is

RETRACTED

RETRACTED

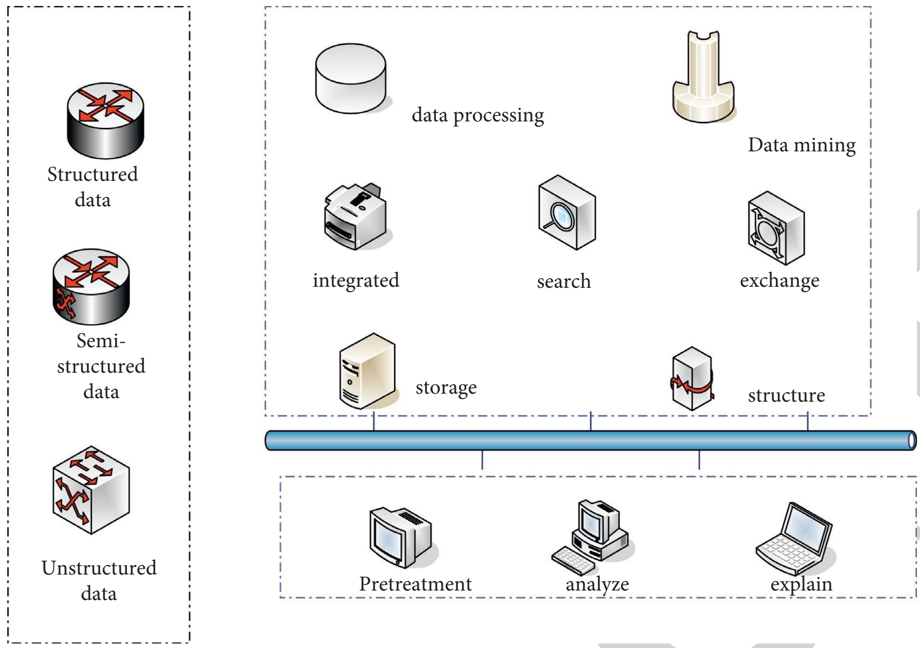

Figure 6: Information attribution structure in big data processing flow.

Table 1: Algorithm NSGA-II and Algorithm UCMOEA on the S-measure of the test function.

S-measure		ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
UCMOEA	BEST	0.0027	0.0021	0.0099	0.0163	0.0011
	MEAN	0.0214	0.0031	0.0196	0.0196	0.0039
	WORST	0.0299	0.0135	0.0213	0.0235	0.0047
NSGA-II	BEST	0.0036	0.0040	0.0100	0.0048	0.0016
	MEAN	0.0319	0.0075	0.0236	0.0089	0.0035
	WORST	0.0736	0.0120	0.0639	0.0132	0.0048

analyzed. Before the analysis of the experimental data, the computational goals of different algorithms are controlled to ensure the scientificity of the results. The specific analysis data is as follows.

According to the data in Table 3, this experiment simulates the performance of three different optimization algorithms. It can be seen from the experiment that the frame rate of the Snake model algorithm is 15, the time for every 800 steps is 613/s, and the error of the algorithm is 0.51. The frame rate of the experimental algorithm is 17, the time per 800 steps is 533/s, and the error of the algorithm is 0.44. The frame rate of the K-DOPs algorithm is 6.7, the time per 800 steps is 1632/s, and the error of the algorithm is 1.04. According to the data, the frame rate of the experimental algorithm is the best, and the frame rate of the K-DOPs algorithm is the worst. In terms of running time, the Snake model algorithm is better than the K-DOPs algorithm. From the error results, the error of the K-DOPs algorithm is larger, and the error of the other two algorithms is smaller.

4.3. Optimization Algorithm Data. In order to verify the effectiveness of different collision optimization algorithms and improve the accuracy of collision effects in different regions, the performance of different algorithms is explored. The details are as follows.

Table 2: Experimental environment configuration table.

Configuration items		Parameters
Servers	64 GB	64 GB
	2 Gbps	2 Gbps
	4 cores	4 cores
	2 TB	2 TB
	64 bit	64 bit
Storage	16 TB	16 TB

According to the data in Table 4, this experiment simulates the performance of four different optimization algorithms. It can be seen from the experiment that the frame rate of the ICDS algorithm is 24, the time for every 800 steps is 487/s, and the error of the algorithm is 0.31. The frame rate of the ACSMCD algorithm is 22, the time for every 800 steps is 499/s, and the error of the algorithm is 0.38. The frame rate of the CDSM algorithm is 19.3, the time per 800 steps is 531/s, and the error of the algorithm is 0.45. The frame rate of the CFCD algorithm is 7.3, the time per 800 steps is 1236/s, and the error of the algorithm is 1.01. Based on the figures, it is clear from that the frame rate of the ICDS algorithm is the best and the frame rate of the CFCD algorithm is the worst. In terms of running time, the CFCD algorithm is superior to the CDSM algorithm. From the error results, the error of the CFCD algorithm is larger, and the error of the other three types of algorithms is smaller.

5. Real-Time Collision Detection Optimization Algorithm for Snake Model

5.1. Collision Efficiency of Different Algorithms. When testing the swarm intelligence optimization algorithm, the parameter of the fitness value is an important reason that

RETRACTED

Table 3: Comparative analysis of experimental data of different algorithms.

Category	Frame rate/s	Time/s for running 800 steps	Error analysis
Snake model algorithm	15	613	0.51
Experimental algorithms	17	533	0.44
K-DOPs algorithm	6.7	1632	1.04

Table 4: Statistical analysis of individual algorithm data.

Category	Frame rate/s	Time/s for running 800 steps	Error analysis
ICDS	24	487	0.31
ACSMCD	22	499	0,38
CDSM	19.3	531	0.45
CFCD	7.3	1236	1.01

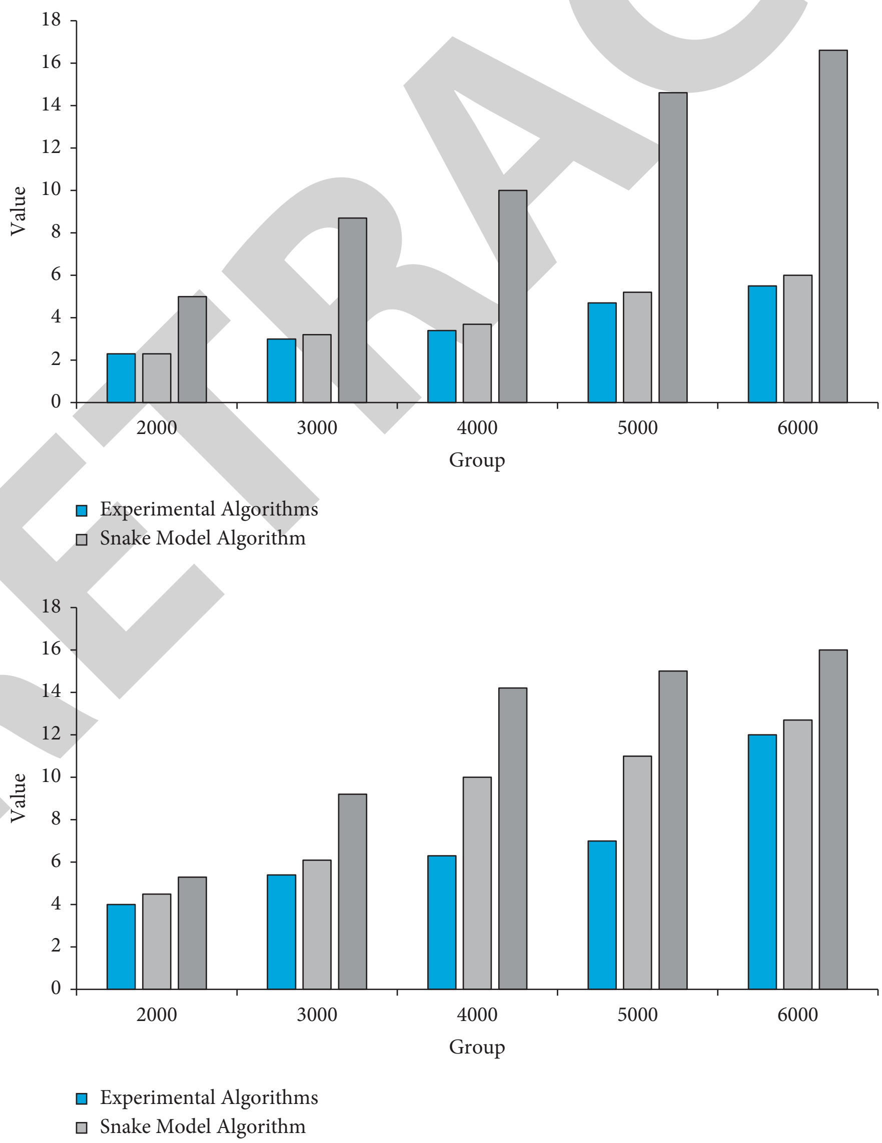

Figure 7: Analysis of collision efficiency of different algorithms.

RETRACTED

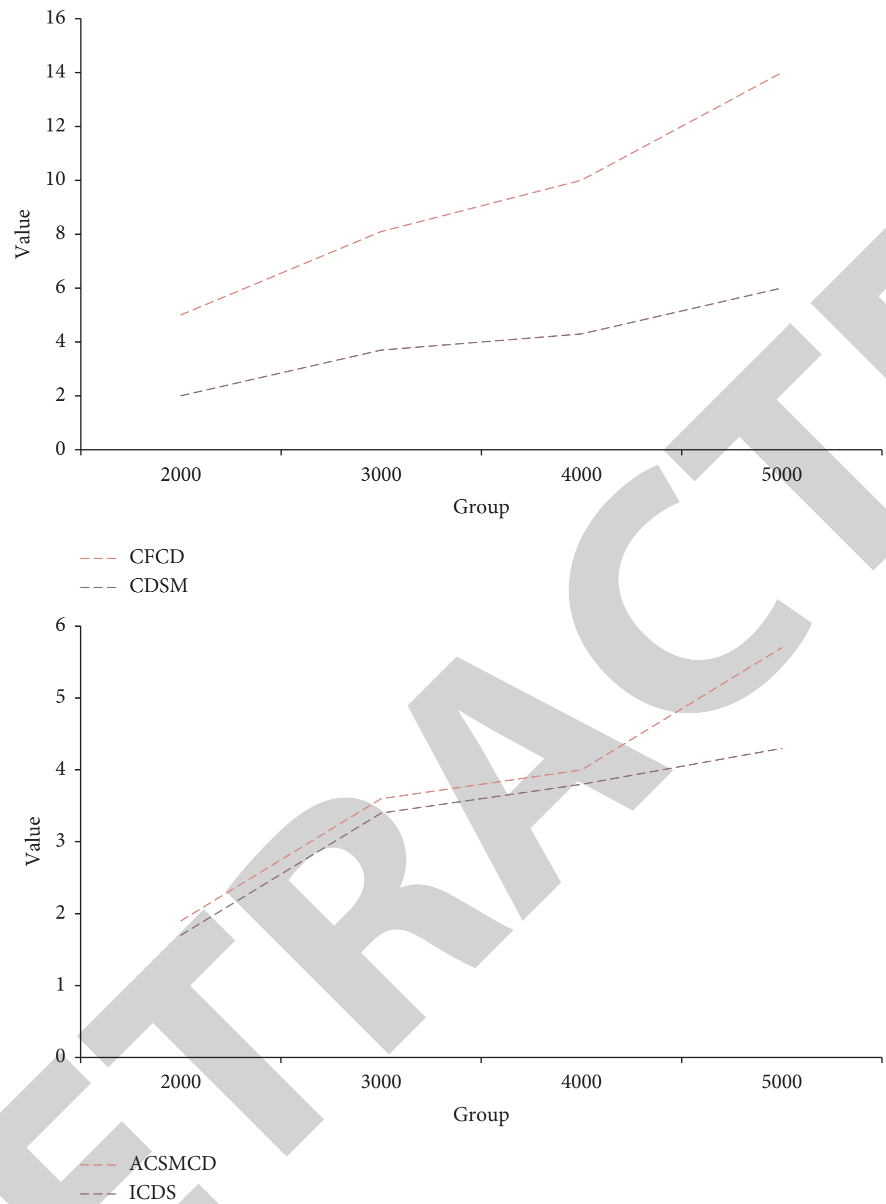

Figure 8: Small area collision efficiency.

affects the test result. In the collision detection process, the difference of the collision area of the target object will have different effects on the test results. Therefore, when comparing different test algorithms, the variables of the target object must be known to prevent errors.

The size of the object collision area affects the experimental results, so the collision situation is presented differently. According to the data in Figure 7, it first analyzes the situation when the collision area is small. According to the data, when the triangle slice is 2000k, the detection time of the experimental algorithm is 2.3 ms, the detection time of the Snake model algorithm is 2.3 ms, and the detection time of the K-DOPs algorithm is 5 ms. When the triangle slice is 3000k, the detection time of the experimental algorithm is 3 ms, the detection time of the Snake model algorithm is 3.2 ms, and the detection time of the K-DOPs algorithm is 8.7 ms. When the triangle slice is 4000k, the detection time of the experimental algorithm is 3.4 ms, the detection time of the Snake model algorithm is 3.7 ms, and the detection time of the K-DOPs algorithm is 10 ms. When the triangle slice is 5000k, the detection time of the experimental algorithm is 4.7 ms, the detection time of the Snake model algorithm is 5.2 ms, and the detection time of the K-DOPs algorithm is 14.6 ms. When the triangle slice is 6000k, the detection time of the experimental algorithm is 5.5 ms, the detection time of the Snake model algorithm is 6 ms, and the detection time of the K-DOPs algorithm is 16.6 ms. According to this data, when there are more and more triangles, the collision detection time will also increase. But, horizontally, the detection efficiency of K-DOPs algorithm is the lowest.

The detection efficiency when the collision area is large is analyzed. According to the data, when the triangle slice is 2000k, the detection time of the experimental algorithm is 4 ms, the detection time of the Snake model algorithm is

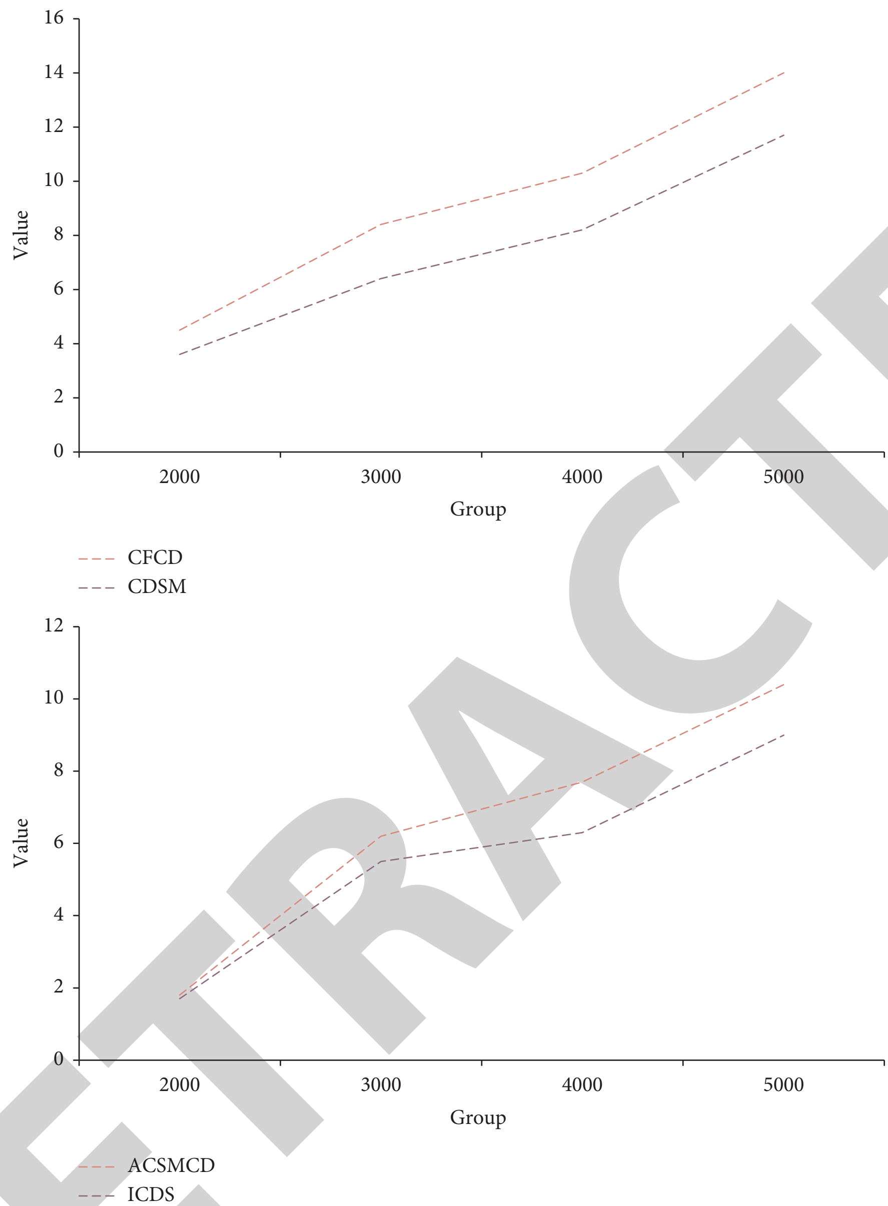

Figure 9: Collision efficiency in large deformation area.

RETRACTED

4.5 ms, and the detection time of the K-DOPs algorithm is 5.3 ms. When the triangle slice is 3000k, the detection time of the experimental algorithm is 5.4 ms, the detection time of the Snake model algorithm is 6.1 ms, and the detection time of the K-DOPs algorithm is 9.2 ms. When the triangle slice is 4000k, the detection time of the experimental algorithm is 6.3 ms, the detection time of the Snake model algorithm is 10 ms, and the detection time of the K-DOPs algorithm is 14.2 ms. When the triangle slice is 5000k, the detection time of the experimental algorithm is 7 ms, the detection time of the Snake model algorithm is 11 ms, and the detection time of the K-DOPs algorithm is 15 ms. When the triangle slice is 6000k, the detection time of the experimental algorithm is 12 ms, the detection time of the Snake model algorithm is 12.7 ms, and the detection time of the K-DOPs algorithm is 16 ms. According to this data, when the number of triangles remains the same, but the collision area increases, the detection time increases accordingly. But the detection efficiency of K-DOPs algorithm is still the lowest.

5.2. Experimental Data. It can be seen from the data in Figure 8 that the collision detection efficiency is improved to a certain extent in the collision area using this experimental method. In order to detect the collision efficiency of different algorithms, different algorithms are analyzed. According to the data in Figure 9, when the target object belongs to a collision with a small deformation area and the number of triangles is 2000k, the detection time of the CFCD algorithm is 5 ms, the detection time of the CDSM algorithm is 2 ms, the detection time of the ACSMCD algorithm is 1.9 ms, the detection time of the ICDS algorithm is 1.9 ms, and the detection time is 1.7 ms. When the number of triangles is 3000k, the detection time of CFCD algorithm is 8.1 ms, that

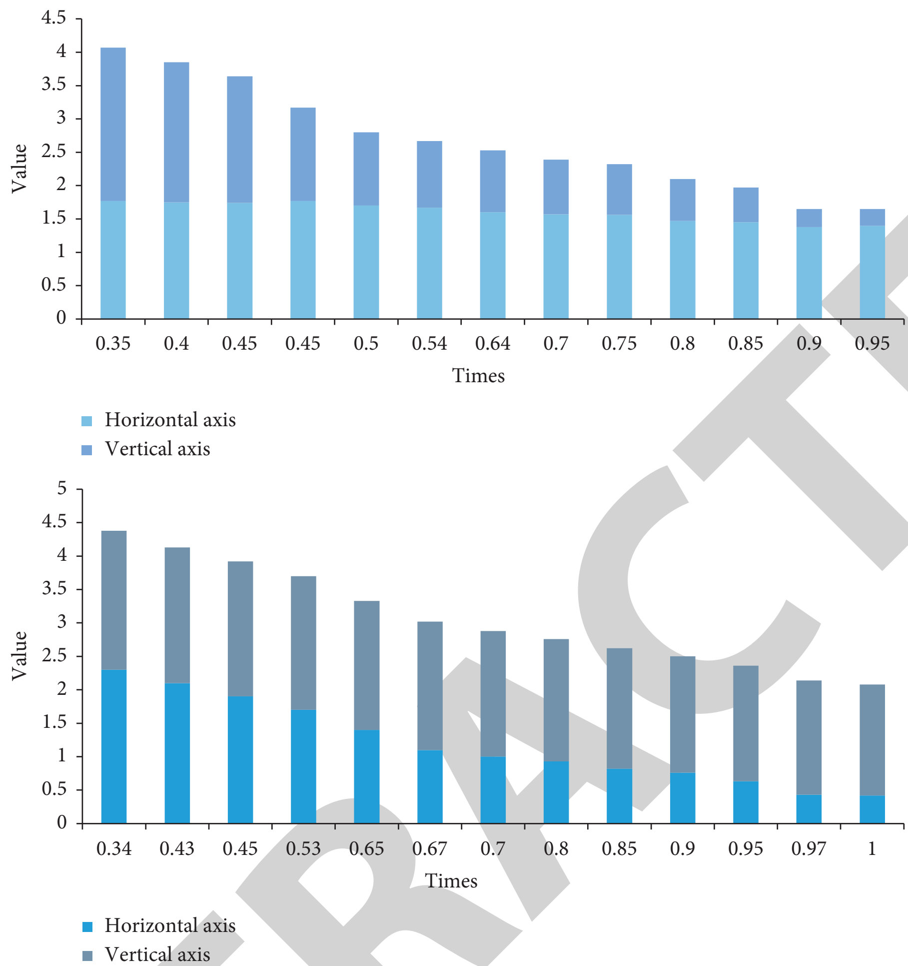

Figure 10: Comparison of the results of the two algorithms.

RETRACTED

of CDSM algorithm is 3.7 ms, that of ACSMCD algorithm is 3.6 ms, and that of ICDS algorithm is 3.4 ms. When the number of triangles is 4000k, the detection time of CFCD algorithm is 10 ms, the detection time of CDSM algorithm is 4.3 ms, the detection time of ACSMCD algorithm is 4 ms, and the detection time of ICDS algorithm is 3.8 ms. When the number of triangle slices is 5000k, the detection time of CFCD algorithm is 14 ms, the detection time of CDSM algorithm is 6 ms, the detection time of ACSMCD algorithm is 5.7 ms, and the detection time of ICDS algorithm is 4.3 ms. According to the data, the CFCD algorithm has the highest detection efficiency.

When the collision area of the target object is enlarged during the experiment, it is found that the four different algorithms have the same trend as the small area in terms of detection efficiency. The detection efficiency of the CFCD algorithm is still the highest.

5.3. Convergence. According to the data in Figure 10, the convergence of the algorithm USMOEA is better than that of the algorithm NSGA-II. It can be clearly seen that the nondominated solution set obtained by the algorithm USMOEA is closer to the Pareto front than that of NSGA-II. The convergence of the algorithm USMOEA has been verified. Although the algorithm USMOEA is not as good as NSGA-II in terms of broadness retention, the original intention of our algorithm is to obtain a nondominated solution set that is closer to the Pareto front and more uniformly distributed. It can be known that the expected solution goal has been basically achieved. The algorithm can obtain the nondominated solution set with more uniform distribution and be closer to the Pareto front.

6. Conclusions

With the development of science and technology, people's production needs are getting bigger and bigger, and social needs are getting wider and wider. The emergence of the Snake model just solves this problem; it aims to minimize the energy objective function and control the deformation of the parameter curve. In particular, the emergence of virtual reality makes traditional computing unable to meet the needs of production. This paper aims to study the optimization algorithm of real-time collision detection based on Snake model in the field of big data. It is expected that, with the support of big data technology, the efficiency of real-time collision detection of Snake model will be improved and time will be saved; at the same time, the contradiction between upper-level knowledge and underlying image features is better reconciled. In this paper, a new calculation method is designed under the improvement of the traditional calculation method. Although experiments have been carried out, there are still shortcomings: when the single-objective particle swarm optimization algorithm solves high-dimensional complex problems, it is still easy to fall into the local optimal solution. And the convergence accuracy still has a large room for improvement. [3–5].

Data Availability

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Conflicts of Interest

The authors state that this article has no conflicts of interest.

Acknowledgments

This work was financially supported by Hainan Provincial Natural Science Foundation of China, Large-Scale Complex Scenario Based on Virtual Reality Is Rapid Real-Time Collision Detection Technology Research, item number 619QN247; and Key Project of “The Thirteenth-Five” Science and Technology Research of Jilin Provincial Department of Education on Optical Motion Capture Technology “Rapid Image Generation and Detection Technology Based on Virtual Reality,” item number JJKH20190907KJ.

References

[1] H. Li, M. Dong, K. Ota, and M. Guo, “Pricing and repurchasing for big data processing in multi-clouds,” *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 2, pp. 266–277, 2016.

[2] A. Noraziah, M. A. I. Fakherldin, K. Adam, and M. A. Majid, “Big data processing in cloud computing environments,” *Advanced Science Letters*, vol. 23, no. 11, pp. 11092–11095, 2017.

[3] L. Wang, Q. Guo, Z. Wei, and Y. Liu, “Spatial conflict resolution in a multi-agent process by the use of a snake model,” *IEEE Access*, vol. 5, no. 99, pp. 24249–24261, 2017.

[4] S. Yu, Y. Lu, and D. Molloy, “A dynamic-shape-prior guided snake model with application in visually tracking dense cell populations,” *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1513–1527, 2019.

[5] Z. Yu, Q. Wang, W. Xiong, C. Zhang, and H. Hu, “Segmentation of cardiac tagged MR images using a snake model based on hybrid gradient vector flow,” *Multimedia Tools and Applications*, vol. 77, no. 17, Article ID 21879, 2018.

[6] Y. Q. Sun, A. G. Wu, N. Dong, and Y. Z. Shao, “A hand tracking algorithm with particle filter and improved GVF snake model,” *Optoelectronics Letters*, vol. 13, no. 4, pp. 314–317, 2017.

[7] H. Y. Qu, W. Zhao, and A. H. Qin, “A fast collision detection algorithm based on optimization operator,” *Journal of Jilin University (Engineering and Technology Edition)*, vol. 47, no. 5, pp. 1598–1603, 2017.

[8] Y. Lou, J. Wei, and S. Song, “Design and optimization of a joint torque sensor for robot collision detection,” *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6618–6627, 2019.

[9] X. Wang, B. Tang, X. Zhou, and X. Gu, “Double-robot obstacle avoidance path optimization for welding process,” *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5697–5708, 2019.

[10] Y. Ma and Y. Liang, “An obstacle avoidance algorithm for manipulators based on six-order polynomial trajectory planning,” *Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University*, vol. 38, no. 2, pp. 392–400, 2020.

[11] W. Youn, H. Ko, H. Choi, I. Choi, J. H. Baek, and H. Myung, “Collision-free autonomous navigation of A small UAV using low-cost sensors in GPS-denied environments,” *International Journal of Control, Automation and Systems*, vol. 19, no. 2, pp. 953–968, 2020.

[12] X. R. Wang, P. J. Ni, and B. Yi, “Research on collision detection in ultrasonic automatic testing of rotary assembly,” *Binggong Xuebao/Acta Armamentarii*, vol. 39, no. 4, pp. 780–786, 2018.

[13] J. Zhao, Y. Chen, H. Zhang, H. Xia, Z. Wang, and Q. Peng, “Physically based modeling and animation of landslides with MPM,” *The Visual Computer*, vol. 35, no. 9, pp. 1223–1235, 2019.

[14] I. Hosni, “Distributed scheduling with efficient collision detection for end-to-end delay optimization in 6TiSCH multi-hop wireless networks,” *Annals of Telecommunications*, vol. 74, no. 5-6, pp. 239–255, 2019.

[15] Q. Guo, L. Zhou, and L. Wang, “Improvement of snake displacement model for roads considering cartographic rules,” *Geomatics and Information Science of Wuhan University*, vol. 42, no. 11, pp. 1629–1634, 2017.

[16] W. Xie, J. Duan, L. Shen, Y. Li, M. Yang, and G. Lin, “Open snake model based on global guidance field for embryo vessel location,” *IET Computer Vision*, vol. 12, no. 2, pp. 129–137, 2018.

[17] W. U. Xin, L. I. Qiang, and Q. I. Bojin, “Application of Snake model and genetic algorithm in special weld seam extraction,” *Hanjie Xuebao/Transactions of the China Welding Institution*, vol. 39, no. 9, pp. 83–89, 2018.

[18] V. Persico, A. Pescapé, A. Picariello, and G. Sperli, “Benchmarking big data architectures for social networks data processing using public cloud platforms,” *Future Generation Computer Systems*, vol. 89, pp. 98–109, 2018.

[19] Z. Sun and P. P. Wang, “Big data, analytics, and intelligence: an editorial perspective,” *New Mathematics and Natural Computation*, vol. 13, no. 2, pp. 75–81, 2017.

[20] Z. Lv, D. Chen, R. Lou, and Q. Wang, “Intelligent edge computing based on machine learning for smart city,” *Future Generation Computer Systems*, vol. 115, pp. 90–99, 2021.

[21] L. Li and J. Zhang, “Research and analysis of an enterprise E-commerce marketing system under the big data environment,” *Journal of Organizational and End User Computing*, vol. 33, no. 6, pp. 1–19, 2021.

[22] Z. Wang, R. Liu, Q. Liu, J. S. Thompson, and M. Kadoch, “Energy-efficient data collection and device positioning in UAV-assisted IoT,” *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1122–1139, Feb. 2020.

[23] A. Ramesh Khaparde, F. Alassery, A. Kumar et al., “Differential evolution algorithm with hierarchical fair competition model,” *Intelligent Automation & Soft Computing*, vol. 33, no. 2, pp. 1045–1062, 2022.

RETRACTED

[24] X. Zhu, "Self-organized network management and computing of intelligent solutions to information security," *Journal of Organizational and End User Computing*, vol. 33, no. 6, pp. 1–16, 2021.

[25] V. Sathiyamoorthi, P. Keerthika, P. Suresh, Z. Justin, A. P. Rao, and K. Logeswaran, "Adaptive fault tolerant resource allocation scheme for cloud computing environments," *Journal of Organizational and End User Computing*, vol. 33, no. 5, pp. 135–152, 2021.

[26] M. Rajalakshmi, V. Saravanan, V. Arunprasad, C. A T Romero, O. I Khalaf, and C. Karthik, "Machine learning for modeling and control of industrial clarifier process," *Intelligent Automation & Soft Computing*, vol. 32, no. 1, pp. 339–359, 2022.

[27] T.-Y. Kim, S.-H. Kim, and H. Ko, "Design and implementation of BCI-based intelligent upper limb rehabilitation robot system," *ACM Transactions on Internet Technology*, vol. 21, no. 3, pp. 1–17, 2021.

[28] F. Meng, Q. Ji, H. Zheng, H. Wang, and D. Chu, "Modeling and solution algorithm for optimization integration of express terminal nodes with a joint distribution mode," *Journal of Organizational and End User Computing*, vol. 33, no. 4, pp. 142–166, 2021.

[29] H. Chen, Y. Lu, and L. Tu, "Fault identification of gearbox degradation with optimized wavelet neural network," *Shock and Vibration*, vol. 20, no. 2, pp. 247–262, 2013.

[30] Y. Zeng, G. Chen, K. Li, Y. Zhou, X. Zhou, and K. Li, "M-skyline: taking sunk cost and alternative recommendation in consideration for skyline query on uncertain data," *Knowledge-Based Systems*, vol. 163, pp. 204–213, 2019.

[31] X. Li, H. Liu, W. Wang, Y. Zheng, H. Lv, and Z. Lv, "Big data analysis of the internet of things in the digital twins of smart city based on deep learning," *Future Generation Computer Systems*, vol. 128, pp. 167–177, 2022.

[32] Q. Liu, S. Sun, B. Rong, and M. Kadoch, "Intelligent Reflective Surface Based 6G Communications for Sustainable Energy Infrastructure," *IEEE Wireless Communications Magazine*, vol. 28, no. 6, pp. 49–55, 2021.

[33] J. Y. Hong, H. Ko, L. Mesicek, and M. B. Song, "Cultural intelligence as education contents: exploring the pedagogical aspects of effective functioning in higher education," *Concurrency and Computation Practice and Experience*, vol. 33, no. 4, 2019.

RETRACTED