Hindawi
Mobile Information Systems
Volume 2022, Article ID 2247839, 11 pages
https://doi.org/10.1155/2022/2247839

Research Article

Considering Multisensor Data Fusion and Data-Driven Special Effects Animation Simulation

Song Peng

School of Design, Hefei University, Hefei 230022, Anhui, China

Correspondence should be addressed to Song Peng; weerrr@hfuu.edu.cn

Received 21 March 2022; Accepted 14 June 2022; Published 16 July 2022

Academic Editor: Wen Zhang

Copyright © 2022 Song Peng. 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.

The data-driven simulation algorithm uses the method of surface matching to establish the corresponding relationship between the surfaces of the two dynamic models at the same time. It also calculates the appropriate deformation field to align the animated character surface and then uses the interpolation operation to further expand the new animated character motion. With the promotion of industrial application requirements such as games, film and television, and education and entertainment, computer animation special effects simulation technology has a wide range of application prospects. Based on a multisensor and data-driven approach, this paper conducts a targeted study on the realistic simulation of typical special effects animations. Aiming at the high time complexity and time-consuming problems of simulation algorithms, data-driven simulation methods are studied and scenes or objects that simulate the real environment can be created. They are given different properties and parameters, and the desired results are obtained through computer calculation and rendering. Finally, it can be output into a smooth animation. Based on the animation simulation data and multisensor data fusion method proposed in this paper, the on-site calculation process can be accelerated and efficient animation simulation driven by data can be realized. In 3ds Max 2013, there is also a small function that can sort the bone list in ascending or descending order in alphabetical and numerical order. This helps manage bones attached to the skin modifier more efficiently. This saves a lot of time when picking bones. About 44.3% of the people during the study were proficient in the use of software. After 1200 frames, the number of foam particles increases sharply, and the two-way coupling algorithm takes time. As the number of foam particles decreases, the time required for bidirectional coupling decreases. The article gives a brief description of the data-driven fusion model. The fluid simulation algorithm generates two different sequences by utilizing two different sets of setup parameters. It is then transformed into a spatiotemporal surface as input data for the fusion model. This research contributes to the development of special effects animation simulations.

1. Introduction

As a product of the combination of computer graphics and art, special effects animation technologies have always been at the forefront of research. Among them, character animation technology has received extensive attention from many experts and scholars. How to model the character behavior in animation to make it have better visual effects and reflect the reality of life more realistically has always been a major problem faced by the researchers of character animation technology. Based on the data-driven fusion model, this paper proposes a matching model based on spatiotemporal deformation. This further solves the problem of matching data with large differences.

Special effects animation simulation is an important part of virtual reality. It has gradually penetrated into our daily life to serve people. The industries of film, TV, games, advertising, construction, medicine, and so forth are closely related to our daily life. All need 3D animation technology to perform and display. It can also be said that 3D animation has become an important form of product display in these industries. There are three levels around data fusion: pixel-level fusion, feature-level fusion, and decision-level fusion. In this paper, multisensor image fusion, color texture image analysis, and radar network detection system are deeply studied. Also a series of new ideas and methods were proposed, and they achieved good results.

The development of animation technology is inseparable from the continuous progress of human science and technology. The progress of human technology not only reflects the wisdom of human beings but also reflects the never-ending pursuit and exploration of the unknown. The reason why multisensor data fusion can effectively improve the performance of the system lies in the fusion of precise and imprecise data, especially when the data has uncertainty and unknown changes. It adopts object-oriented design ideas and uses Action Script 3.0 for animation programming, which can achieve good interactivity. In the analysis of animation data, animation playback, and scene rendering, more efficient and safe algorithms are adopted, which can meet the real-time requirements of users for the system. It designs and develops a basic class of application framework, which effectively realizes the system's expansibility and maintainability.

2. Related Work

The computational cost of data-driven animation is determined by the number of meshes or particles. As this number increases, the amount of computation will gradually increase, and the demand for computing resources will also increase accordingly. In order to break through the limitations of traditional animation effects simulation methods, this paper will describe in detail a new idea of animation effects simulation, that is, data-driven animation effects simulation methods. It also discusses the key technologies of model-based animation effects simulation methods. In real life, a joint is like a hinge, which is used to connect two objects. In ODE, its concept is similar. It acts as a link between two rigid bodies to ensure that the two rigid bodies maintain a certain constraint relationship in position and direction. Serban et al. believe that the past decade has seen major breakthroughs in data-driven models in several areas of speech and language understanding. In the field of dialogue systems, the trend is less clear, and most practical systems are still built with a lot of engineering and expert knowledge. However, some recent results suggest that data-driven approaches are feasible and promising. To facilitate research in this area, they conducted an extensive survey of publicly available datasets suitable for data-driven learning of dialogue systems. They discussed the important features of these datasets and how they can be used to learn different dialogue strategies. They also described other potential uses of these datasets, such as transfer learning methods between datasets and the use of external knowledge. They also discussed the selection of appropriate evaluation metrics for learning objectives [1]. Chang-Jiang and Liu introduced a leader. Its connection to the follower changes over time. They also proposed a new data-driven consensus protocol based on model-free adaptive control, where the reference input of each follower is designed to be the average of neighboring agent outputs over time. Cases where the leader has a prescribed reference input and others are taken into account. The protocol they proposed allows for time-varying delays, switching topologies, and does not use proxy structures or dynamic information implicitly or explicitly. They derived sufficient conditions to guarantee closed-loop stability and obtained conditions for consensus convergence, in which only one joint spanning tree is required. They conducted numerical simulations and practical experiments to demonstrate the effectiveness of the proposed protocol [2]. Wei et al. introduced an effective recurrent neural network to reconstruct the dynamics of the nonlinear system [3]. Bakirov et al. argued that recent data-driven soft sensors typically use multiple adaptive mechanisms to cope with nonstationary environments [4]. Feng et al. selected and compared many machine learning algorithms in two layers [5]. Their research on special effects animation lacks realistic features and needs to add multisensor and data fusion content to optimize special effects animation, so they collected the research content of others.

Wang et al. established an iterative neural dynamic programming method. They used data-driven control formulations to design near-optimal regulators for discrete-time nonlinear systems [6]. Cheung et al. believed that websites often use animation to capture the attention resources of online consumers. While previous research has focused on the effect of animation on animated banner ads, limited research has examined the effect of animation on other items on the same web page [7]. Liao et al. reported on the results of the 2014 Data Fusion Competition organized by the Image Analysis and Data Fusion Technical Committee (IADF TC) of the IEEE Geoscience and Remote Sensing Society (IEEE GRSS). They presented and discussed the results obtained by the winners of the two tracks [8]. Panetta et al. introduced a new concept, Selective Color Transfer (SCT). It allows color experimentation and visualization of selected objects in an image without affecting any other image content or color [9]. Wu et al. presented a new algorithm and implementation for real-time identification and tracking of speckled filaments in fusion reactor data. Similar spatiotemporal features are also important in many other applications, such as tumor cells. They proposed a method to extract these features. They divided the whole task into three steps: local identification of feature units, grouping of feature units into extended features, and tracking of the movement of features through spatial overlap [10]. Pixel-level fusion is fusion performed directly on the original data layer. It performs data synthesis and analysis before the raw forecasts of various sensors are preprocessed. This is the lowest level of fusion. For example, in an imaging sensor, the process of confirming the target attribute by performing image processing and pattern recognition on a blurred image containing several pixels belongs to pixel-level fusion.

3. Research Methods of Special Effects Animation Simulation

3.1. Animation Film and Television Production. The research focus of this paper is how to design and use data-driven approach to realize animation effects simulation.

(1) Animation frame. Animation frame is a very widely used concept in the field of animation. It refers to the

animation picture of animation at a certain point in time, each frame has a time attribute, and multiple frames are connected together to form a complete connected animation. However, animation frames are stored in various forms. Some animation frames store the entire animation scene at one point in time. It may be stored in the form of pixels, or it may be stored in other forms such as function parameters. In this system, animation frames only store joint information, because it is enough to complete the reproduction of the character's actions. Specifically, starting from the character's root joint, the system will sequentially store the joint rotation information of all joints at this frame moment. The root joint needs to store its position and rotation in world coordinates. The rest of the joints will store their rotation information relative to the parent joint. The production of 3D movies needs to be carried out according to the customer's requirements, communicate with the production team, estimate the production cost, and investigate the market demand.

(2) Animation sequence. Animation sequence is a collection of animation frames sorted by time, which stores all animation information of the character. No matter which way to generate animation, it is ultimately to get its animation sequence. Animation sequences have many properties, such as the total number of animation frames, the first frame, the last frame, and the current frame. This information is used when the animation sequence is played by the system. The length of the time steps between animation frames determines whether the animation is smooth or not.

(3) The storage of animation. An animation sequence is an abstract concept. It is only valid while the program is running; with the end of the program running, all the animation data that has been generated is lost. Therefore, the generated animation data should be stored in the external memory in the form of files. Animation sequences are structured data. Therefore, it needs to choose a suitable format for file storage, which can be defined by itself or compatible with existing mature animation files. Now there are many animation files, such as X files, BVH files, and FBX files. Due to some advantages of BVH, this system adopts BVH file as the external storage form of animation. BVH (Biovision Hierarchy) is the Biovision hierarchical model file. It is a file format developed by Biovision Corporation to describe data obtained from motion capture. This file can describe realistic human animation. Because many BVH files come from animation data based on real actor performances, BVH files can be generated in many ways; in addition to motion capture, 3ds Max, POSER, and other software programs can also be produced. Additionally, BVH files are stored in text form. Its structure is simple and clear, which is convenient for storage and analysis in the development process. The BVH file is divided into 2 main parts: the skeleton information of the character and the animation data block. Among them, the skeleton information recursively defines the relevant information of all bones and joints of the character according to the hierarchical structure, such as the relative displacement of the joints and each rotation channel. The other part is the animation data block, which is stored frame by frame. It first stores the total number of frames and frame rate of the animation and finally all the animation information from the first frame to the last frame. In this system, the animation sequence of characters will be stored in BVH format. Moreover, the animation effects that have been created can be reproduced by reanalyzing. The general production process of 3D stereoscopic movies is shown in Figure 1.

The basic principle of inverse kinematics [7] is

$$\Delta x = J\Delta\theta. \tag{1}$$

In inverse kinematics calculations, the following formula is more commonly used [8]:

$$\Delta\theta = J^{+}\Delta X + \alpha\left(I - J^{+}\right)\Delta Z, \tag{2}$$

where J^{+} is the representation of the pseudoinverse of a matrix commonly found in mathematical linear algebra.

3.2. Special Effects Animation System Design. In this special effects animation system, each character is assigned a behavior. The behavior can be understood from the moment the simulation key is pressed to the moment when the simulation key is pressed again to stop the simulation. A series of state changes are experienced by a character, with each state corresponding to a type of action. It works by adding multiple roles to the system and editing the behavior of each role. Finally, it simulates the behavior of each character and finally generates a series of actions of multiple characters. The composition of the character behavior is shown in Figure 2. It is mainly divided into three parts: finite state machine (FSM), variable part (Variables), and sequence (Sequence). These three parts are organically integrated, and finally the simulation of character behavior is realized. The animation special effect system performs speech synthesis and output on the constructed animation through the voice sensor, and the visual sensor is responsible for the output of the dynamic effect. Users can add multiple states to the role and add some transitions between states (Transition) and some events bound to some transitions. For example, if it wants the character to fall after running for 3 seconds, you can set a fall event in the third second of the sequence. When the system simulates to the third second, the falling event will be triggered, so that the character will enter the falling state and make the falling action.

3.3. Scenarios for Constructing Complex Simulations. The actual application scenario is quite complex. It generally includes multiple geometric models, and the effective

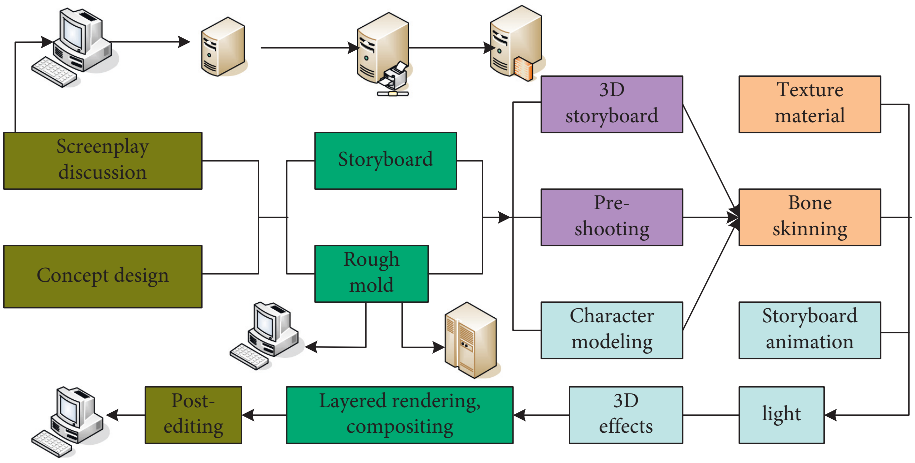

FIGURE 1: The rough production process of 3D stereoscopic movies.

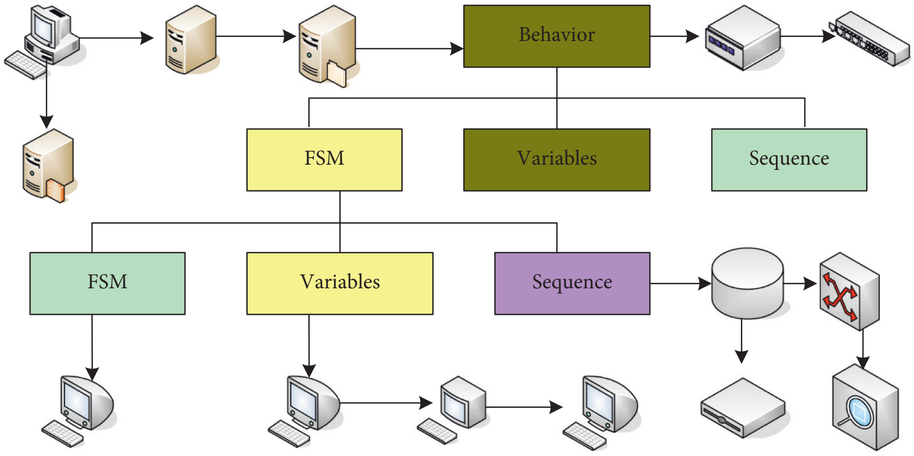

FIGURE 2: Composition of character behaviors.

arrangement of the simulated scene is the content that many users pay attention to. Complex scenes with multiple models can be easily generated based on signed distance fields. First, it is necessary to define a macro scene into which other geometric models are to be placed. For convenience, a hexahedron is used here to define the scope of the macroscopic scene. It is assumed that the scope of the considered scene is limited to the following hexahedron [9]:

$$W = (s_x, e_x) \times (s_y, e_y) \times (s_z, e_z). \quad (3)$$

W will be discretized into a differential grid G as the background grid for solving the formula system.

After loading the triangular patch, the system performs appropriate translation, scaling, and rotation operations to calculate the bounding box of the triangular patch. The so-called bounding box is a hexahedron whose edges are parallel to the corresponding coordinate axes. This hexahedron contains exactly all the points in the triangular patch. It assumes that the bounding box is [10]

$$m = (x_{\min}, x_{\max}) \times (y_{\min}, y_{\max}) \times (z_{\min}, z_{\max}). \quad (4)$$

Its general requirements are $m \subset W$.

If the signed distance fields of two objects are φ_1 and φ_2, respectively, the following formula can be used to calculate the result obtained by combining the two objects.

Signed distance field [11] is

$$\varphi_{12} = \min\{\varphi_1, \varphi_2\}. \quad (5)$$

The calculation method for merging $n (n \geq 2)$ signed distance fields is [12]

$$\varphi = \min\{\varphi_1, \varphi_2, \ldots, \varphi_n\}. \quad (6)$$

The signed distance field for each model can be quickly computed using the Open CL version of the two-layer particle method. It then uses the above formula to combine all signed distance fields and can quickly construct complex scenes for animation effects.

Smoke Simulation. An actual smoke simulation may require an arbitrary shape of the density field. The density field characterizes the concentration of smoke and is a function of spatial location. The value of the density field is generally set to be between 0 and 1. A value of 0 represents no smoke at the point, and a value of 1 represents the maximum density considered. A signed distance value of $\varphi(x) < 0$ means point x is inside the object; otherwise, it is outside. The two-layer particle algorithm proposed in this study can rapidly construct signed distance fields from triangular patches. Here, we discuss how to use the resulting signed distance field to conveniently construct a density field that can be used for smoke simulation. Here the signed distance field and density field are denoted by symbols φ and d, respectively. The simplest method of density field construction is to use the following Heaviside function [13]:

$$d(x) = \begin{cases} 0, & \varphi(x) < 0, \\ 1, & \mathit{other}. \end{cases} \tag{7}$$

The actual smoke density function should be a smoother function. The simple Heaviside function will produce sharp boundaries, causing distortion. Therefore, it is generally advisable to take the smoothed Heaviside function. The following is a version of the smoothed Heaviside function [14]:

$$d(x) = \begin{cases} 0, & \varphi(x) \geq \varepsilon, \\ \dfrac{1}{2} + \sin\varphi, & -\varepsilon \leq \varphi(x) \leq \varepsilon, \\ 1, & \varphi(x) \leq \varepsilon. \end{cases} \tag{8}$$

It does the following for arbitrary grid points (a, b, c) [15]:

$$\begin{aligned} C_Z = {} & d(a,b,c) + d(a-1,b,c) + d(a+1,b,c) \\ & + d(a-1,b+1,c). \end{aligned} \tag{9}$$

Generally, a smoother density field can be obtained by performing two averaging operations.

In general, the initial values of velocity and pressure are set to zero.

$$u \mid \Gamma_s = u_{s,n}. \tag{10}$$

Here, n is the outer normal vector at the region boundary and u_n is the velocity normal component.

The level set function φ needs to be continuously updated. Its rule follows the following level set formula:

$$\frac{\alpha\varphi}{\alpha t} + (u + \nabla)\varphi = 0. \tag{11}$$

Here, u is the velocity field, which can be obtained by solving the Navier-Stokes formula.

3.4. Data Driven. Compared with traditional special effects animation simulation methods, data-driven algorithms have also had a huge impact in various fields. With the rapid development of big data technology and many machine learning methods, data-driven methods have been widely used in various fields. Especially in the related fields of special effect animation modeling, a wide range of usage scenarios have been generated, such as the application of 3D special effect animation model transformation and deformation, the application of 3D model reconstruction and surface reconstruction, and the application of geometric material design. In the field of animation simulation, many data-driven algorithms have also been widely used.

There are several problems that need to be paid attention to when applying the level set method:

$$\varphi_t + S(\varphi_0)(|\nabla\varphi| - 1) = 0. \tag{12}$$

This formula is also known as the reinitialization formula of the level set function. However, the symbol function lacks smoothness, and it is easy to affect the numerical calculation, so the smoothed symbol function is generally used, such as

$$S(\varphi_0) = \frac{\varphi_0}{\sqrt{\varphi_0^2 + (\Delta x)^2}}. \tag{13}$$

From the point of view of partial differential formulas, the extrapolation process is to solve the following formula:

$$\nabla\varphi \cdot \nabla q = 0. \tag{14}$$

Foams in flow fields are usually generated in regions of high velocity and complex motion. The greater the kinetic energy in the region is, the higher the degree of disorder is, and the more foam is produced. The numbers of foam particles Q_K^i and Q_i^v produced are related to the kinetic energy and velocity difference:

$$Q_K^i = \sum \frac{1}{2} m v^2, \tag{15}$$

$$Q_i^v = \lambda \left\| v_{ij} \right\| \cdot w_{ij}^Q, \tag{16}$$

where j represents a neighbor particle.

It maps the number of foam particles produced by each particle to the [0, 1] interval to obtain

$$R_i = \frac{Q_{\max} - Q_i}{Q_{\max} - Q_{\min}}, \tag{17}$$

where $Q_{\max}$ and $Q_{\min}$ are user-defined parameters representing the maximum and minimum values, respectively. Overall size of candidate particles n_a is:

$$n_a = R_i^k R_i^v \cdot n_{\max} \cdot \Delta t. \tag{18}$$

In the above formula, $n_{\max}$ is the maximum sampling rate, which is used to adjust the final generation number of foam particles.

It obtains the velocity and position of foam particles in the flow field according to coordinate transformation and basis transformation:

$$v(t + \Delta t) = r_a \cdot \ \sin\theta \cdot n + v, \tag{19}$$

$$s(t + \Delta t) = x_f(t) + r_a \cdot \ \cos\theta \cdot n + v + h_a. \tag{20}$$

We have that n_1 and n_2 are two random vectors that are linearly independent and perpendicular to vector Δx.

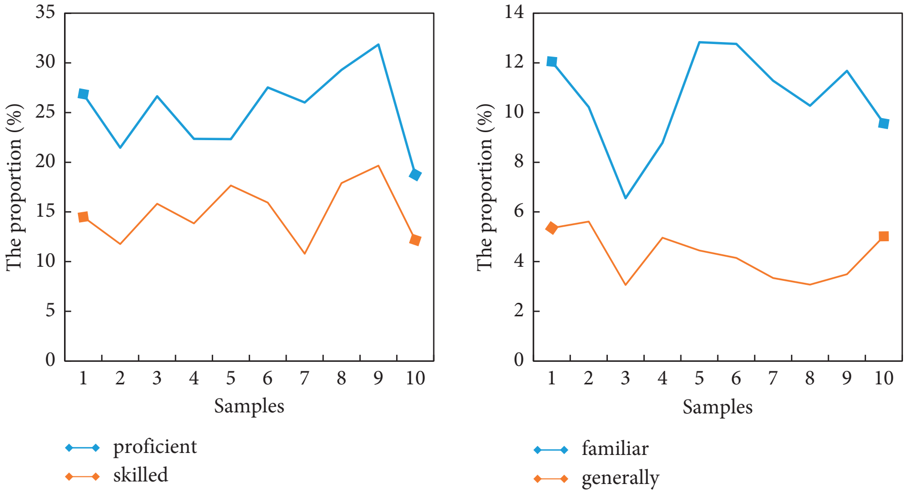

Figure 3: Animation familiarity.

4. Special Effects Animation Simulation Results

The number of people who are proficient in action production is 31, and there are 20 people who are proficient (proficiency statistics are shown in Figure 3(a)). Almost half (about 44.3%) of people are proficient in the use of software (familiarity, general statistics are shown in Figure 3(b)).

There are 18 people who are good at making models, 11 people who specialize in materials, and 9 people who are good at lighting production (model, material, and lighting statistics are shown in Figure 4(a)). There are 12 people who are good at scene rendering and 16 people who are good at special effects animation. Finally, there are 4 people who are good at the production of special effects (rendering, animation, and special effects statistics are shown in Figure 4(b)).

There are only 5 people who can design the plug-in and 21 people who can modify the plug-in. The number of people who have used the plug-in is 28, and the remaining 16 have never used the plug-in (the statistics can be designed and modified, as shown in Figure 5(a)). According to the analysis of this result, it can be found that the proportion of people who can develop plug-ins is 7.14% (the statistics of those who will use it and those who have never used it are shown in Figure 5(b)), which is a low proportion. Compared with the status quo of plug-in development abroad, the number of people engaged in plug-in development in China needs to be increased urgently. In addition, 44 people, that is, 62.9% of the investigators, did not use the plug-in enough and only had a relatively simple understanding.

It can explain the advantages and disadvantages of 3ds Max, accounting for 62.9% of the total number of people, and the remaining 37.1% of the respondents are not very clear about this. A survey of the strengths and weaknesses of character animation is shown in Figure 6.

Compared to other plug-ins, Char Rigger's performance is the best plug-in. The character M plug-in developed and designed in this research is only supported by 12 in terms of practicality. Compared with the same type of skeleton plug-ins abroad, it is in a middle and downstream position, only slightly better than some domestic plug-ins. The performance comparison of the plug-in is shown in Table 1.

In order to more comprehensively reflect the application advantages of plug-ins in character skeleton production, three students were randomly selected from the participants in the survey. Then, they did a second test from their time on the bones of the character's hands, legs, and feet. The test results are shown in Table 2. The first column of data is the completion time required by this plug-in, and the second column is the completion time required by the old-fashioned method.

It can be seen from Table 3 that the acceleration effect is obvious on Tesla GPU, and the parallel algorithm has at least 10 times faster acceleration. On Intel CPU, the acceleration is faster when the number of particles is small, and the acceleration ratio becomes smaller as the number of particles increases. There are many reasons for this problem. It may be that the number of processors that can be parallelized by Intel CPU is relatively small, or it may be related to the hardware architecture. Table 3 shows the runtime comparison of particle flow simulation using OpenCL parallelization.

The change in the number of particles with time is shown in Figure 7. Among them, red represents the total number of particles, blue represents the number of droplet particles, green represents the number of foam particles, and yellow represents the number of bubble particles. In this experiment, the upper limit of the number of foam particles is set to 5.5×10^5 to prevent the number of foam particles from increasing indefinitely. It can be seen that, at the 2000th time step, all nine blocks touch the liquid-free surface. At this time, the degree of chaos in the scene is the largest, and the amount of foam generated reaches the peak (the foam and all particles are shown in Figure 7(a)). Due to the limitation of the maximum sampling rate, the total number of foam particles does not increase for a period of time thereafter (the statistics of droplets and bubbles are shown in Figure 7(b)). Droplets, scum, and bubbles will be transformed into each other during the simulation process, and finally all converted into scum under the action of buoyancy and gravity.

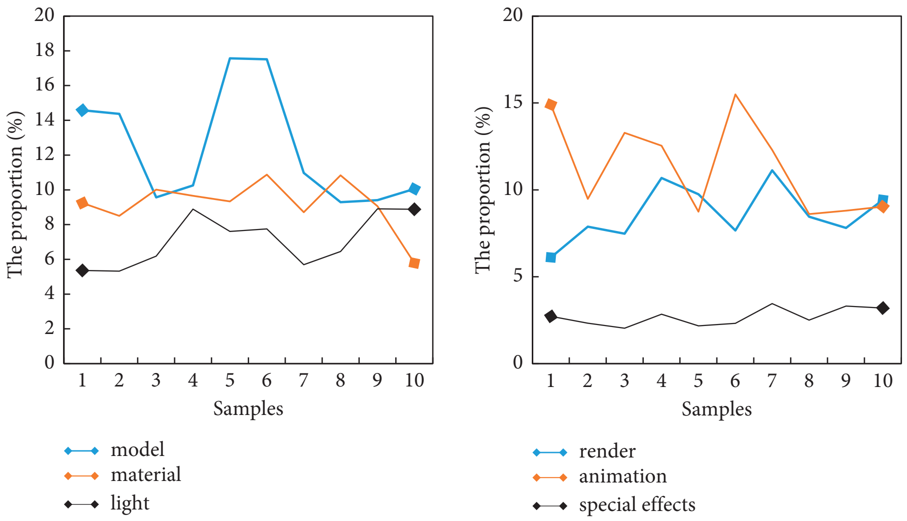

Figure 4: Production expertise. (a) Model, material, and lighting statistics. (b) Rendering, animation, and special effects statistics.

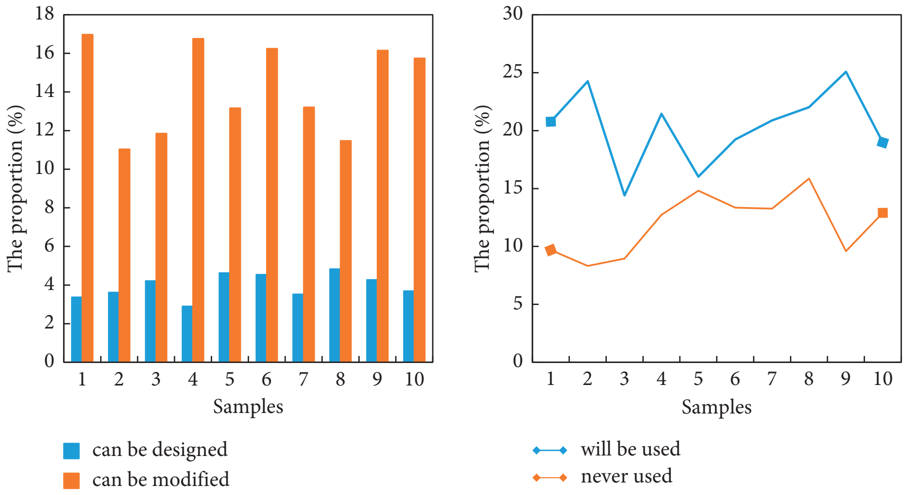

Figure 5: Making feats. (a) Designable, modifiable statistics. (b) Statistics of future use and no previous use.

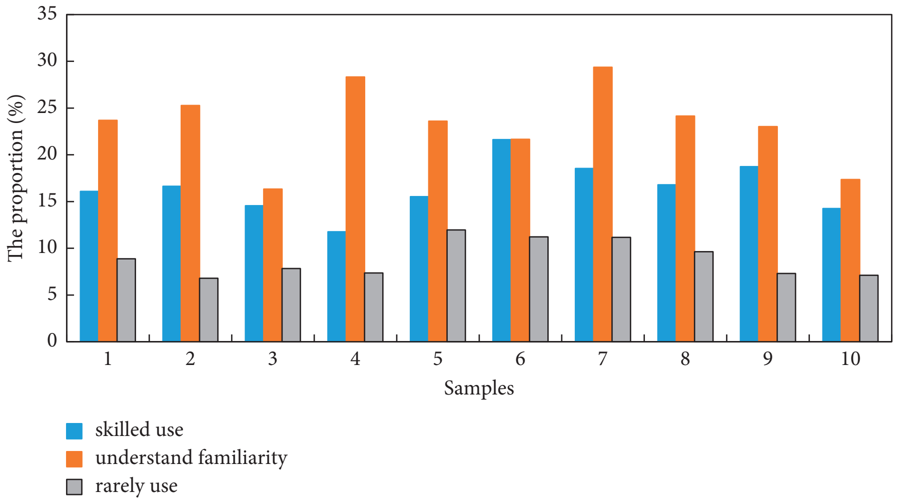

Figure 6: Survey of the pros and cons of character animation.

Table 1: Performance comparison of plug-ins.

Performance	Char rigger	Character M	Rig_creator	Spline IK	Robustness	Pea skeleton
Practicality	19	12	15	11	Bone	10
Stability	14	8	17	13	18	17
Convenience	11	17	13	19	14	14
Professional	9	15	15	14	11	13

Table 2: Results of skeletal testing.

Location	Classmate A (min)		Classmate B (min)		Classmate C (min)	
Custom hand bones	3	13	6	20	5	13
Custom leg bones	2	8	4	12	4	8
Custom foot bones	2	15	3	15	3	16

Table 3: Runtime comparison of granular flow simulations parallelized using OpenCL.

Number of spherical particles	No CL	CL (Tesla)	CL (Intel)	R (Tesla)	R (Intel)
10000	90.49	8.32	7.29	10.88	12.41
20000	176.14	13.70	16.8	12.86	10.48
40000	332.21	26.97	40.82	12.32	8.14
80000	609.12	46.67	105.82	13.05	5.76

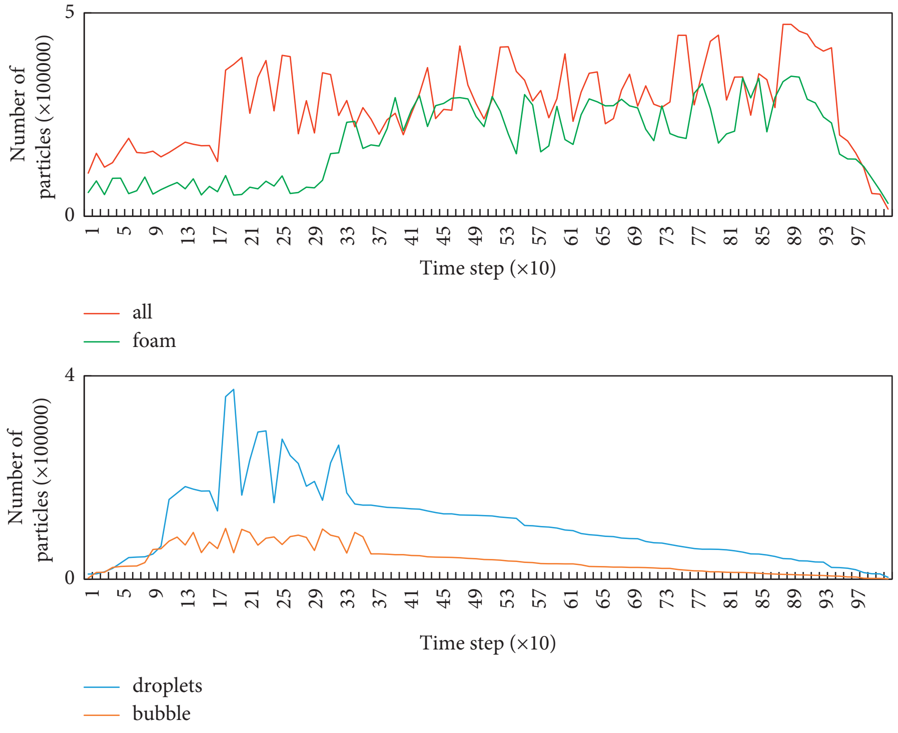

Figure 7: Change in number of particles over time. (a) Foam and total particles. (b) Droplets and air bubbles.

Time Efficiency Comparison. In order to verify the efficiency of the model, this study compared the time consumption of the simulation algorithm without foam particles, the method, and the two-way coupling algorithm. The time for the two-way coupling algorithm is obtained by calculating the cross-neighbor lookup time. The yellow represents the pure simulation method without considering the diffusion material, the blue represents the method of this study, and the green represents the two-way coupling method. When the foam particles have not been generated, the time consumption of the three algorithms is basically the same. After 1200 frames, the number of foam particles surged, and the time consumption of the two-way coupling algorithm increased sharply. As the number of foam particles decreases, the time required for bidirectional coupling decreases gradually. The computational efficiency of the two-way coupling method is proportional to the number of foam particles, while the efficiency of the method in this study is

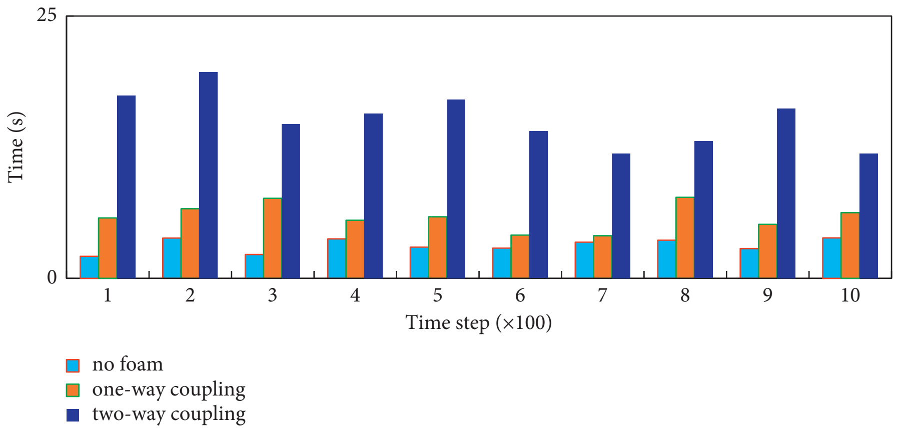

Figure 8: Time-consuming comparison of pure-fluid transverse fitting method, research method, and two-way coupling method.

Table 4: Vertex count versus drawing efficiency.

Performance	Number of vertices (pieces)	500	5,000	50,000	100,000
Drawing method based on geometry	Simulation efficiency (frames/second)	115–180	52–89	27–42	16–30
Drawing method based on multigroup volume texture	Simulation efficiency (frames/second)	91–120	47–69	39–72	26–45
Scheme of this paper	Simulation efficiency (frames/second)	86–93	34–44	30–38	7–15

Table 5: Comparison of visual effects and simulation efficiency of three rendering methods.

Method	Number of geometric vertices (pieces)	Visual effect
Geometry-based method	100,000	16–30
Single-group texture method	100,000	33–35
Multigroup volume texture method	100,000	27–46

not affected by the number of foam particles. The experimental results show that the method in this study can deal with the coupling of the particles with the foam particles more efficiently, and the computational cost is negligible. The time-consuming comparison of the pure-fluid transverse simulation method, the research method, and the two-way coupling method is shown in Figure 8.

For scenes of different scales, the number of vertices and rendering efficiency are compared as shown in Table 4. When the number of vertices is much larger than 100,000, the frame rate of the algorithm based on the geometric model drops significantly, and real-time rendering is difficult to achieve. The method based on multigroup volume texture can still maintain real-time rendering efficiency regardless of scene complexity.

In addition, this paper also compares the visual effects and simulation efficiency of the three rendering methods, as shown in Table 5. The rendering method based on geometric model has obvious advantages in realistic performance. However, if the number of primitives to be drawn increases, the drawing speed will be affected and drop sharply. The method based on the horizontal monolithic texture is fast in rendering, but the texture data is lost seriously and the realism is poor. The method in this paper is in the middle in terms of visual effect performance, which is slightly aliased compared to the geometry-based method. But it is better than the single-group horizontal volume texture method. In terms of algorithm complexity, the method of multigroup volume texture is to sample and fuse three groups of volume texture data. The simulation speed is slightly slower than that of a single-group texture, but it can still achieve the characteristic of real time.

When using the particle model to control the simulation, the acceleration process of the flowing water is relatively fast. It can reach a relatively large speed in a relatively short period of time. In the animation simulation effect produced, the acceleration of flowing water is more natural and real. The flow simulation is shown in Figure 9.

When drawing a moving geometry, the normals of triangles and points on the mesh must be updated after the point is moved. Due to the use of local fields, only those points and triangles within the valid area need to be updated. This can speed up the rendering process and increase the frame rate of the water wave simulation. Table 6 shows the comparison results of the local and global calculation consumption of the geodesic distance field and the time required for grid update.

5. Discussion

Sensor sampling points are usually distributed on the head, torso, and limbs of the motion model. Secondary movements such as gestures and expressions are added to the captured primary movement using other animation methods. The key to motion capture is that the sensor sampling points on the bound actor and the joints of the simulated actor controlled in the computer must strictly correspond. The number of sensors depends on the type of movement of the character, the method of data transfer, and

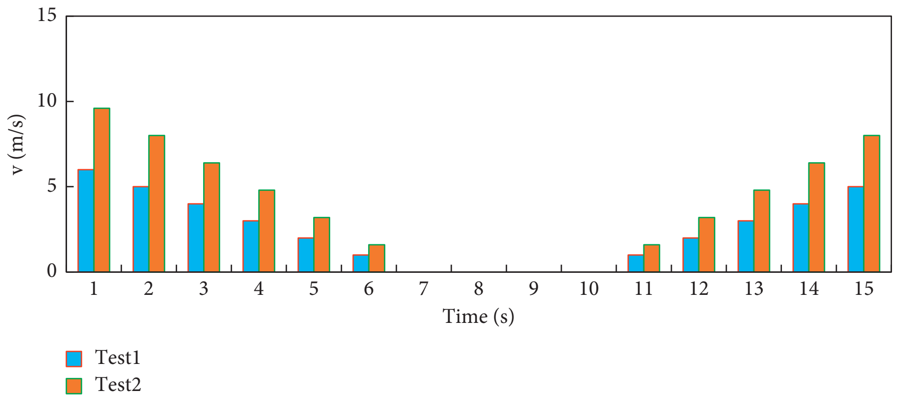

Figure 9: Flow simulation.

Table 6: Comparison results of local and global computation consumption and grid update time for geodesic distance fields.

Model information		Global simulation		Local simulation	
Model	Number of points	Distance calculation	Grid update	Distance calculation	Grid update
Sphere	16,002	0.141	0.054	0.125	0.115
Venus-Head	33,587	0.375	0.128	0.407	0.176
Horse	48,486	0.531	0.186	0.031	0.013

the type of movement constraints. This technology can greatly reduce the manual operation of character action design, but the technology is complicated and the production cost is expensive.

Point force is the easiest way to apply force. It applies a force of a given magnitude and a given direction to the specified position of the character's specified joint. It can directly act on the character's joints, so that the character's action can be deformed. In addition, forces of different sizes in different directions will be superimposed and finally applied to the character joints together, and after the current simulation step ends, all superposition of forces will be cleared. With the promotion of industrial application demands such as games, film and television, and digital entertainment, the application prospects of computer animation technology are becoming more and more extensive. Computer special effects simulation is a digital technology that is different from the real simulation of the physical world, and it pursues a special visual effect.

Motion capture technology captures the movement trajectories of the actor's main joints through sensors. It realizes the automatic recording of animation character motion information. It automatically generates the basic trajectory of the movement and then records the movement. In movie scenes or game scenes, it often occurs that characters are subjected to various field forces, such as gravity, wind, and strong attraction fields. All these phenomena can be simulated by means of point force. First, the user provides the magnitude of the field force, which can be given by means of acceleration, and then calculates the product of the mass and acceleration of each rigid body of the character through Newton's third law. It obtains the final force of each joint of the character and acts on each rigid body in turn through the force function of ODE and finally presents the effect of the character under the action of the field force as a whole. Animation-based special effects simulation involves many aspects: distance field, implicit surface, 3D object morphing, and so forth.

Collision geometry can be created for a character's limbs to participate in collisions in the simulation. It is also possible to create collision geometry for static geometry in the environment, which allows the character to interact dynamically with the environment. For each character, collision data based on the following types of geometry can be created for each of its joints: Cube, Plane, Sphere, Capsule, and Model.

ODE has implemented the collision simulation of rigid bodies. It can realistically and efficiently reproduce the collision effect of objects in contact with each other. This effect is exactly what the character needs to demonstrate physical realism. It can vividly and realistically simulate the collision effect between the character in action and itself, other dynamic objects, and the surrounding static environment. This thus overcomes its disadvantages such as keyframe animation and motion capture animation that suffer from bleed-through. This is also an important feature of generating power based on dynamics. At present, the movement of particles has not considered the problem of collision detection. In the future particle path calculation, it is necessary to add collision prediction, path adjustment, or collision response calculation to make the dynamics of particle illusion more realistic and natural in detail. With the development of computer technology and its wide application in the field of education and entertainment, in order to realize the reproduction of real natural scenes, the research on natural scene simulation in graphics has also become a hot spot. The use of multisensor data fusion simulation algorithms in graphics can simulate water, clouds, fire, smoke, snow, and other natural scenes and their animations very realistically.

6. Conclusion

The continuous innovation of science and technology makes the production of animated films have to use three-dimensional animation effects to enrich the picture. Shooting and synthesizing from the earliest two-dimensional hand-painted frame animation save a lot in terms of cost and time. It can get better results as expected. Three-dimensional (3D) animation special effects technology is in an emerging period. Since the reform and opening up, China's science and technology have also been changing with each passing day. More and more new ideas and new technologies are known and used by Chinese people. With the rapid development of computer animation technology, the simulation and interactive presentation of realistic special effects animation have gradually become an important and critical content in animation creation. Based on multisensor and data-driven methods, this paper conducts targeted research on the simulation of realistic special effects animation. The research in this paper will improve the quality and efficiency of special effect animation production and increase the authenticity and immersion of real-time somatosensory interactive experience. It has important theoretical and practical significance. In future work, multiple special effects can be combined to simulate and draw animations that also rely on special effects simulation in the scene at the same time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This research study was sponsored by 2020 Anhui Provincial First-Class Professional Construction Project (Project no. 165(Animation)). The author acknowledges the project for supporting this article.

References

[1] I. V. Serban, R. Lowe, and P. Henderson, "A survey of available corpora for building data-driven dialogue systems," *Computer Science*, vol. 33, no. 16, pp. 6078–6093, 2017.

[2] L. I. Chang-Jiang and G. P. Liu, "Data-driven leader-follower output synchronization for networked non-linear multi-agent systems with switching topology and time-varying delays," *Journal of Systems Science and Complexity*, vol. 31, no. 001, pp. 87–102, 2018.

[3] Q. Wei, R. Song, and P. Yan, "Data-driven zero-sum neuro-optimal control for a class of continuous-time unknown nonlinear systems with disturbance using ADP," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 444–458, 2017.

[4] R. Bakirov, B. Gabrys, and D. Fay, "Multiple adaptive mechanisms for data-driven soft sensors," *Computers & Chemical Engineering*, vol. 96, no. jan.4, pp. 42–54, 2017.

[5] C. Feng, M. Cui, B. M. Hodge, and J. Zhang, "A data-driven multi-model methodology with deep feature selection for short-term wind forecasting," *Applied Energy*, vol. 190, pp. 1245–1257, 2017.

[6] D. Wang, C. X. Mu, and D. R. Liu, "Data-driven nonlinear near-optimal regulation based on iterative neural dynamic programming," *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 43, no. 3, pp. 366–375, 2017.

[7] M. Y. M. Cheung, W. Hong, J. Y. L. Hong, and J. Thong, "Effects of animation on attentional resources of online consumers," *Journal of the Association for Information Systems*, vol. 18, no. 8, pp. 605–632, 2017.

[8] W. Liao, X. Huang, and F. V. Coillie, "Processing of multi-resolution thermal hyperspectral and digital color data: outcome of the 2014 IEEE GRSS data fusion contest," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2984–2996, 2017.

[9] K. Panetta, B. Bao, and S. Agaian, "Fast hue-division-based selective color transfer," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 2853–2866, 2020.

[10] L. Wu, K. J. Wu, and A. Sim, "Towards real-time detection and tracking of spatio-temporal features: blob-filaments in fusion plasma," *IEEE Transactions on Big Data*, vol. 2, no. 3, pp. 262–275, 2017.

[11] K. Amasyali and N. M. El-Gohary, "A review of data-driven building energy consumption prediction studies," *Renewable and Sustainable Energy Reviews*, vol. 81, no. 1, pp. 1192–1205, 2018.

[12] B. Cai, Y. Zhao, H. Liu, and M. Xie, "A data-driven fault diagnosis methodology in three-phase inverters for PMSM drive systems," *IEEE Transactions on Power Electronics*, vol. 32, no. 7, pp. 5590–5600, 2017.

[13] X. Xie, Eb Moh, M. Ujjaman, and L. G. R Eb Holz, "Data-driven filtered reduced order modeling of fluid flows," *SIAM Journal on Scientific Computing*, vol. 40, no. 3, pp. B834–B857, 2017.

[14] M. Cerrada, R. V. Sánchez, C. Li et al., "A review on data-driven fault severity assessment in rolling bearings," *Mechanical Systems and Signal Processing*, vol. 99, pp. 169–196, 2018.

[15] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and Buildings*, vol. 140, pp. 81–97, 2017.