Hindawi
Journal of Sensors
Volume 2022, Article ID 2580575, 10 pages
https://doi.org/10.1155/2022/2580575

Research Article

Real-Time Simulation Method of Flame Animation Based on the Deep Stripping Algorithm and Texture Mapping

Jingjing Fei

Zhejiang Fashion Institute of Technology, Ningbo 315211, China

Correspondence should be addressed to Jingjing Fei; 2010707021@zjff.edu.cn

Received 17 March 2022; Revised 13 April 2022; Accepted 5 May 2022; Published 1 June 2022

Academic Editor: Yuan Li

With the gradual integration of computers into people's work and life, it is no longer an unattainable goal to use the power of science and technology to simulate various states of matter. In the future, virtual simulation modeling technology will become more and more high-end with the improvement of the needs of the times. However, nowadays, there are many problems in generating real-time image animation. For traditional simulation methods, huge computation and realistic rendering have become technical bottlenecks. This topic takes the generation of flame animation as an example to illustrate. On the basis of the depth stripping algorithm and texture mapping method, not only particle system modeling is introduced, but also a new N-S equation is improved for research. The research results show the following. (1) The model constructed by this method has superior performance and has been greatly improved; after iteration, the error value is as low as 0.03 and the accuracy value is as high as 98%. (2) After 500 simulations, there are 399 kinds of static and 698 kinds of dynamic flames, respectively; it takes 24.01 s and 98.21 s to generate 500 incomplete animations. (3) Compared with the particle system and deep learning model for flame animation recognition and detection, the highest false alarm rate of this model is 4.6%, and the lowest is 0.9%, and the experimental effect is stable; the highest recognition of flame animation can reach 99.2%, and the lowest is 98.5%. (4) Generally speaking, experts and ordinary people have a high evaluation on the effect of this model; except for the first group, the scores are all over 80%, and the highest score can reach 85.936%. Finally, the experimental results are good, which proves the effectiveness and feasibility of this method, which has certain research contribution value. However, for the follow-up work, the algorithm and model have room for further improvement.

1. Introduction

3D animation makes use of 3D space technology and related software modeling and realizes realistic dynamic and static pictures. It can even generate applications of various scenes in real time, which is the blueprint for the future of mankind. However, it is very difficult to simulate irregular natural products such as wind, water, smoke, and fire in life scenes. They cannot be defined and constructed by general mathematical models and physical attributes. Therefore, how to generate the real texture of flowing objects has always been a very challenging subject. For some skilled professionals, the process of realizing a flame is just a basic animation exercise. But the simpler the animation, the more difficult it is to control and realize the details. A flame contains a lot of knowledge in the field of computer graphics, and every dynamic and static performance is followed by unpredictable and ever-changing states. The purpose of this paper is to build the model faster and better in real time, simplify the calculation process, and reduce the redundant calculation. In addition, the paper also strengthens the color matching of flame, increases the detail texture, and ensures the realistic effect. After many investigations and consulting, there are many real simulation data about flame at home and abroad, and the research degree and cognitive degree are different.

This paper refers to the previous research literature and methods and combines new ideas of multiple modes to create innovations, which have certain theoretical and practical significance. The following literature can give some theoretical support and data support to this paper. The specific contents are as follows. The improved double depth stripping algorithm combined with Beer's law can be used to calculate the attenuated light intensity [1]. Use the network of depth feature fusion and reconstruction to distinguish the microfiber existence characteristics of the convolution neural network [2]. Aiming at the high detection rate, low false detection rate, and real-time requirements of the video flame detection algorithm, experiments are carried out based on target tracking and multifeature fusion [3]. Based on the trust model and dynamic and static features, a probability model of multifeature fusion is established [4]. A particle system is used to control the properties of firework particles at the moment of explosion, and texture mapping is used to draw firework particles [5]. The Gyarmathy model and Perlin noise are introduced for geometric modeling, and complex flames are drawn in virtual scenes, and the real-time and fidelity aspects are studied [6]. Deep stripping and GPU are combined to draw a real-time approximate soft shadow [7]. Based on Level-Set surface deformation, the modified MacCormack and smoke density evolution curve are introduced, and a flame blue core surface evolution model is proposed [8]. The algorithm is completely implemented in GPU, fast modeling and rendering, and the final result is obtained by alpha mixing [9]. Use C4D and AE software particle special effects to make animation in three-dimensional space [10]. For local texture mapping, the region is established by picking points, and the interactive topological structure of data is studied [11]. The computer is used to simulate fluid motion, and mathematical and physical models are constructed based on N-S equations to simulate real-time flame [12]. For the Z-buffer algorithm and ray tracing algorithm, a visibility detection scheme without setting deviation is formulated [13]. TLM modeling in GPU texture mapping operation is based on UML-SystemC [14]. Texture mapping of the 3D model used the ASM algorithm, RBF interpolation function, feature point constraint, and deformation [15].

2. Theoretical Basis

2.1. Flame Animation. A flame [16] is a physical and chemical phenomenon in nature. At the same time, it is an extremely complex and unpredictable state, with hot gasification parts. In essence, a flame is an exothermic reaction, which relies on the high-speed movement of surrounding air molecules in the reaction range to emit light. In the field of vision, it is a fluid phenomenon with random motion changes, so it is difficult to describe it concretely by geometry. At the same time, due to its ever-changing state in time and space, it is difficult to truly simulate the dynamic feeling of flame combustion. Making an excellent flame animation through simulation experiments is a piece of deep knowledge. Therefore, the best way to accurately describe the interaction of various molecules in a mass of gas is to model. The classical modeling materials are sorted out in order to deepen understanding, as shown in Table 1.

In this paper, we mainly pursue the "realism" of flame, which belongs to the visual research purpose. Therefore, we can abandon some normative requirements in physics. The complicated motion of heated gas is abstracted and simplified as a dynamic motion field. By changing the original monotonous motion law in the sports field, the generation of flame will be more flexible and real. Among them, we define the sports field as an n-dimensional vector in the real number field. It has 1 to n function attributes with time t. The vortex field is defined as Y_{vortex}, a five-dimensional vector over a real number field. The specific formula is as follows:

$$\begin{gathered} Y_{\text{motionfield}} = \{\text{property1, property2, property3}, \cdots, \text{property}n\} \quad (n \in N), \\ Y_{\text{vortex}} = \{\text{Angle, Angle Delta, Pos, Velocity, Acc}\}. \end{gathered} \tag{1}$$

The rotation angle characteristic value is as follows:

$$\text{Angle}_{j+1} = \text{Angle}_j + \text{AngleDelta} \times \Delta t. \tag{2}$$

Position and speed are as follows:

$$\begin{gathered} \text{Pos}_{j+1} = \text{Pos}_j + \text{Velocity}_j \times \Delta t, \\ \text{Velocity}_{j+1} = \text{Velocity}_j + \text{Acc} \times \Delta t. \end{gathered} \tag{3}$$

2.2. Deep Stripping Algorithm. Depth peeling's [21] full name is the deep stripping algorithm. This is a technology that can solve the problem of unordered rendering of opaque objects and realize sorting depth values. Using the shadow mapping proposed by Lance Williams, the depth test is simulated. Through multiple rendering methods, the color of the target will be stripped out layer by layer according to the flow. Then, according to the target requirements, the rendering results are obtained by mixing colors in a certain order.

The specific algorithm flow is as follows. (1) Firstly, render transparent objects, using the depth buffer test function to get color values. Then, copy the depth value into the depth texture. (2) Secondly, turn on the comparison function and filter the colors that can pass the test. This step mainly selects colors with large depth values. Then, the color value and the corresponding depth value of the next layer are obtained. (3) Repeat the second step continuously, separating layer after layer of color until all colors are stripped out. (4) Arrange the hierarchical colors in the order from back to front. Then, gradually render the depth and complexity of transparent objects to get the results.

Generally, API functions in the ARB extension function in OpenGL are used to realize the effect of this algorithm. OpenGL [22] is a software interface on graphics hardware and a state machine, which can write three-dimensional graphics applications. Its operation sequence is similar, and a series of processing stages are called rendering pipelines. In addition, based on the low-level mechanism of the window system, some function libraries can simplify programming.

2.3. Texture Mapping. Texture mapping [23], the Chinese whole process, is a kind of OpenGL hardware, which can save a variety of textures. It has the functions of reducing calculation amount and improving calculation speed. A texture image is attached to the surface of a three-dimensional object to improve the surface color richness. This mapping process can enhance the realism of objects. Sometimes, users can help adjust the surface shape, reflection, shadow, and

Table 1: Research status of establishing the model.

Research proponent	Method or model	General situation of research
Perlin	Empirical model [17]	It can preliminarily simulate natural textures such as marble, clouds, and flames.
Inakage	Digital synthesis algorithm	It can realistically simulate the fuzzy boundary of flame and the smoke effect around the flame.
Imakage	Texture mapping method [18]	A simple model of two-dimensional flame [19] can be constructed.
Philippe Beaudoin	Different equations [20]	Create a three-dimensional flame frame. Simulate the shape of combustion flame, including ambient temperature, oxygen content, and airflow speed.

other information through this method. It can be used with the deep stripping algorithm, illumination calculation, image mixing, and other technologies so as to design a more perfect and beautiful effect. The related function steps are as follows.

2.3.1. Two-Dimensional Mapping. According to the target, different intermediate mapping media are used to generate different textures. Then, paste the texture on a simple three-dimensional object. If S represents a mapping, there are

$$S:(u,v)\longrightarrow\left(x',y',z'\right). \tag{4}$$

2.3.2. Three-Dimensional Mapping. After the first step, the texture on the three-dimensional object that has been mapped once is mapped to the final target object surface. If O stands for mapping, there are

$$O:\left(x',y',z'\right)\longrightarrow(x,y,z). \tag{5}$$

In addition, there are some pieces of knowledge or operations related to texture mapping, which are supplemented here.

Perspective correct interpolation is as follows:

$$\begin{bmatrix} u \\ v \\ 1 \\ x_r \\ y_r \\ z_r \\ w_r \end{bmatrix} \xrightarrow{\text{homogenize}} \begin{bmatrix} \dfrac{u}{w_r} \\ \dfrac{v}{w_r} \\ \dfrac{1}{w_r} \\ \dfrac{x_r}{w_r}=x_s \\ \dfrac{y_r}{w_r}=y_s \\ \dfrac{z_r}{w_r}=z_s \\ 1 \end{bmatrix}. \tag{6}$$

The pixel footprint [24] is as follows:

$$\psi=\phi\circ\pi^{-1}. \tag{7}$$

The higher-order Taylor expansion is as follows:

$$\psi(x)=\psi(x_0)+J(x-x_0). \tag{8}$$

The three-dimensional stripe texture is as follows:

$$\begin{gathered}\text{RGB stripe}(\text{point}\,p,\text{real}\,w),\\ t=\frac{(1+\sin(\pi px/w))}{2},\\ \text{return}\,(1-t)c_0+tc_1.\end{gathered} \tag{9}$$

2.4. Mathematical Physics Method. In this paper, the N-S (Navier-Stokes) equation is introduced [25]. This method is approved at home and abroad and can completely describe fluid phenomena. Its kinetic energy conservation equation is as follows:

$$\frac{\partial u}{\partial t}=(u\nabla)u-\frac{1}{\rho}+\nu\nabla^2u+F. \tag{10}$$

The mass conservation equation is as follows:

$$\nabla u=0, \tag{11}$$

where u represents the velocity field; ρ represents the density of the fluid; F refers to the external force; t denotes time; ν can explain the kinematic viscosity coefficient; $(u\nabla)u$ means the advection term.

The whole process steps are as follows:

$$u_0(x)\xrightarrow{\text{advact}}u_1(x)\xrightarrow{\text{diffuse}}u_2(x)\xrightarrow{\text{force}}u_3(x)\xrightarrow{\text{project}}u_4(x), \tag{12}$$

$$u_1(x)=u_0(x-\delta t u_0(x)), \tag{13}$$

$$u_2(x)=u_1(x)-\delta t\nu\nabla^2u_1(x), \tag{14}$$

$$u_3(x)=u_2(x)+\delta tF(x), \tag{15}$$

$$\nabla^2p(x)=\nabla u_3(x), \tag{16}$$

$$u_4(x)=u_3(x)-\nabla p(x). \tag{17}$$

Formulas (13)–(17) are the flow steps after formula (12) is decomposed. Formula (13) is the calculation of the advection term, which is operated by the semi-Lagrange formula. Formula (14) is related to the diffusion term and can be obtained by the implicit method. Formula (15) is the answer to add the external force term. Formulas (16) and (17) represent the velocity with a divergence of 0, which is mainly obtained by the projection of the H-H theorem.

3. Research on Simulating Real-Time Flame Animation

3.1. Introducing Particle System. The simulation method of the particle system is simple and easy to realize. However, it cannot accurately describe the process of flame combustion. But it takes a large number of tiny particles as its basic elements, which can effectively represent those irregular fuzzy substances. In this experiment, each particle can have many properties, which are consistent with the various properties of the flame itself. And the particle system contains two forms: dynamic and static, and each particle experiences a different life cycle. It coincides with some concepts abstracted by us—the motion of the dynamic sports field. As for some weaknesses of the particle system, we integrate the method of the dynamic motion field to make up for the design of the flame animation model. The initialization improvement of the particle system is described as follows.

Determine the initial position of new particles with the following:

$$\begin{aligned}\mathrm{Pos}X &= \frac{1}{\sqrt{2\pi R}}\exp\left\{-\frac{(x_i - x_0)^2}{2R^2}\right\},\\ \mathrm{Pos}Z &= \frac{1}{\sqrt{2\pi R}}\exp\left\{-\frac{(z_i - z_0)^2}{2R^2}\right\}.\end{aligned} \tag{18}$$

The location of new flame particles is as follows:

$$\begin{aligned}\mathrm{Pos}X &= \sum_{i=0}^{n} R\mathrm{and}() \times \mathrm{adjustnum},\\ \mathrm{Pos}Z &= \sum_{i=0}^{n} R\mathrm{and}() \times \mathrm{adjustnum}.\end{aligned} \tag{19}$$

3.2. Improvement of the Model Method

3.2.1. Double-Layer Depth Stripping Algorithm. In this experiment, although the traditional depth stripping algorithm can ensure the correct color results, the rendering of N times leads to the inefficiency of this method, which reduces the final animation effect. Therefore, we need to improve the original algorithm. On the basis of the original, considering the hardware requirements and financial constraints, this paper gives priority to the proposed double-layer deep stripping algorithm. The new algorithm can increase the running speed by nearly two times. This method uses new technology, is easy to implement, and can save the maximum or minimum depth value. At the same time, the new algorithm can store multiple color values. Thus, the target of the first layer and the last two layers can be stripped out by rendering again. It ensures the correctness of the original algorithm, improves the performance of the algorithm to the maximum extent under the consideration of the existing resource constraints, and can complete the complete real-time rendering task in complex scenes.

3.2.2. Modified N-S Equation. After the introduction of the particle system, the mathematical and physical methods in this paper need to be modified. That is to say, the advection term of the N-S equation is modified to meet the needs of experiments. This is because the introduction of the particle system affects the key to flame formation in the system. If it is not changed and corrected in time, it will directly destroy the formation of flame animation. When simulating flame, the external shape of the flame is constructed by the generation and disappearance of particles. At this time, the advection item needs to add parameters:

$$u_1(x) = \alpha u_0(x - \delta u_0(x)). \tag{20}$$

It should be noted that the value range of α is $[0, 1]$. In this way, the shape of the whole part is controlled by the particle system. For the detailed natural state of the fluid, it is debugged by the improved N-S equation.

3.3. Analog Implementation. We use a flow chart to illustrate the formation and implementation of the whole flame animation, as shown in Figure 1.

This paper uses a variety of methods to simulate animation effects. Firstly, the concept of the dynamic motion field is put forward, and some shortcomings are compensated by combining the particle system. For GPU objects, OpenGL tools are used to accelerate the implementation of Navier-Stokes equations. It is worth noting that the N-S equation here has been improved on the advection term based on the characteristics of the particle system. At this time, the motion of particles can be completely controlled. Use the double-layer depth stripping algorithm to render real color, and then add the texture mapping method to enhance the realism of animation and improve the calculation speed.

4. Experimental Analysis

4.1. Experimental Environment. In this section, we show some of the main experimental equipment, as shown in Table 2.

4.2. Flame Effect Display. Based on the particle polygon patch, the running speed of the system can be accelerated without too many particles. We designed five kinds of flame particle textures. It is also necessary to map textures to enhance the realism of details in order to achieve the most ideal flame dynamic effect, as shown in Figure 2.

There are many modes in the formal simulation of flame effects, and we intercept some simulation states to show them. We can see that there are clearly visible flames and fireworks, as well as simulations of the surrounding tiny particles or smoke patterns, and the details are clearer than those in the traditional particle system animation. It can also simulate the random form of fire being blown when it is blown by the wind, as shown in Figure 3.

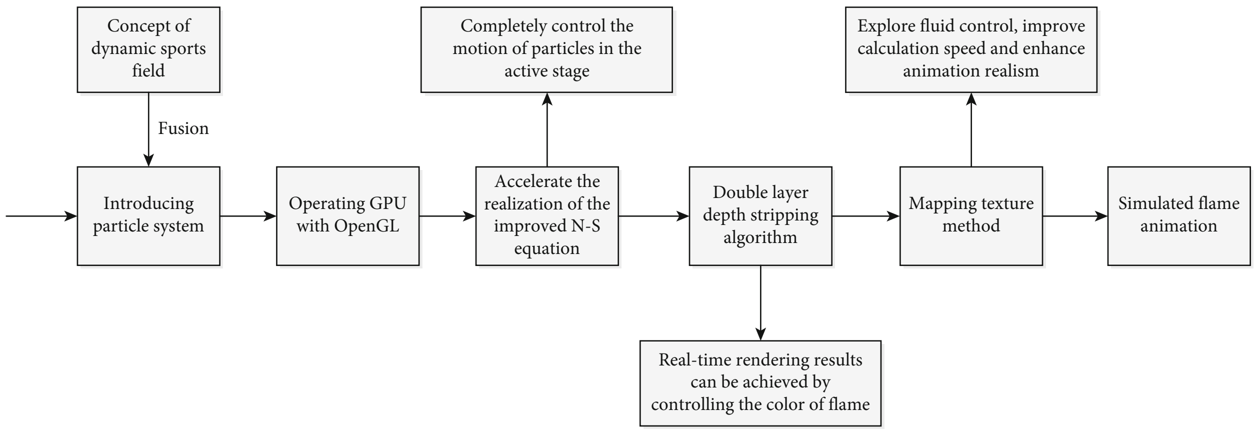
Concept of dynamic sports field
Fusion
Introducing particle system
Operating GPU with OpenGL
Accelerate the realization of the improved N-S equation
Completely control the motion of particles in the active stage
Double layer depth stripping algorithm
Real-time rendering results can be achieved by controlling the color of flame
Mapping texture method
Explore fluid control, improve calculation speed and enhance animation realism
Simulated flame animation

Figure 1: The flow of simulating flame.

Table 2: Hardware and software environment settings of the experiment.

Environment name	Configuration content of the environment
Processor	Intel i5-7200U
Memory	6 GB
Experimental operating system	Windows 10
CPU	3.0 GHz
GPU	NVIDIA GeForce 7900
Development tools	Microsoft Visual Studio.NET 2005 OpenGL2.0

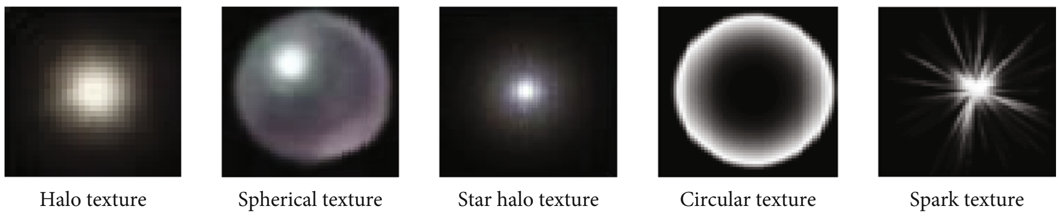

Figure 2: Texture effect display.

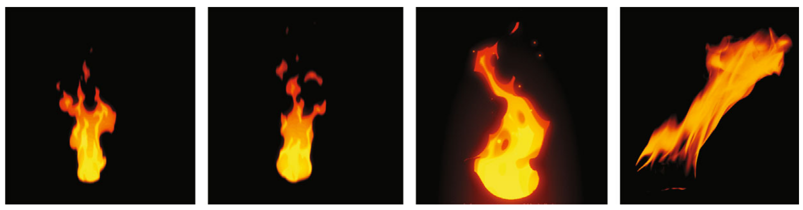

Figure 3: Simulated flame animation effect.

4.3. Performance Test. In order to verify whether the final performance of the model is superior to the traditional algorithm model after the application of this method, we test the error and accuracy of the model. Note that the model needs to be pretested to check the error. If the loss function is too large and does not reach the expected target value, the model generation is unstable, and further modification of the model is needed. As for the accuracy test, the accuracy of the algorithm is mainly judged when the model error is stable and the formal test is carried out. Both the pretest stage and the formal test stage are iterated 60,000 times, and the test results every 5000 times are recorded and evaluated after quantification, as shown in Figures 4 and 5.

In the pretest stage, we set the error target value to be less than 0.05. If the test results do not meet this requirement, then this experiment is invalid, and the model and algorithm need to be modified again. From the figure, we can find that after 5000 iterations, both methods can converge rapidly to varying degrees. At this time, the traditional method can only converge from 0.85 to 0.134, and the fluctuation value

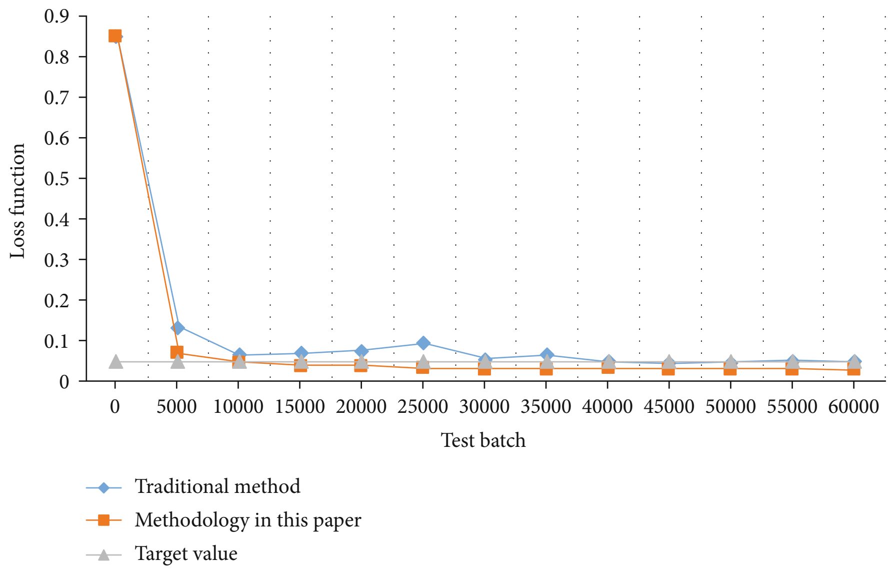

Figure 4: Pretest error analysis.

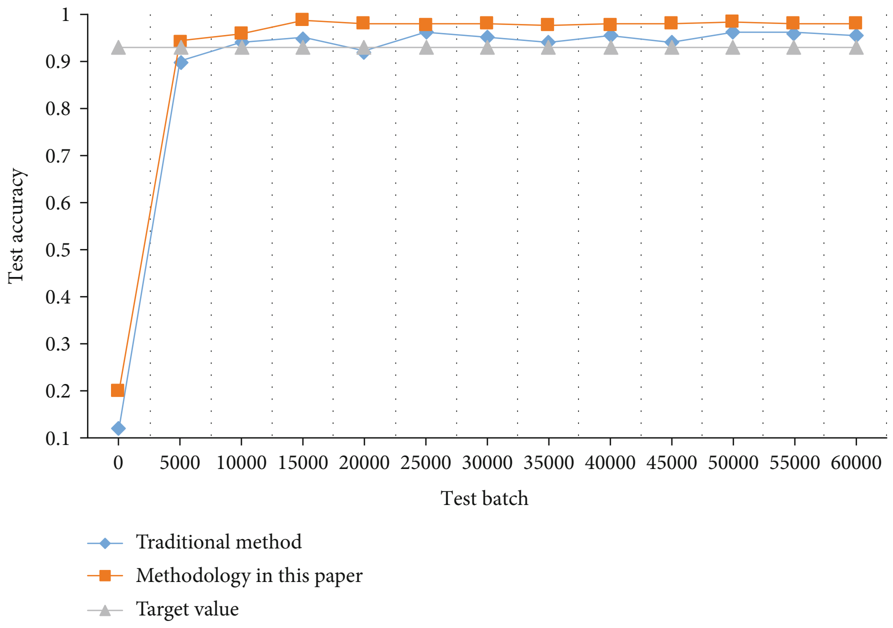

Figure 5: Test accuracy analysis.

of subsequent iterations is large, and it is not stable to the target value of about 0.05 until 40,000 iterations. The method in this paper can quickly converge to a smaller size and directly reduce it to 0.07. Then, after 30,000 iterations, the error value can be stabilized at about 0.03. Not only the target value is reached, but also the final error value is smaller, and the effect is obvious.

In the formal test phase, we set the final accuracy target value as 0.93. That is to say, the accuracy of the model should reach more than 93% before it is qualified. For the traditional method, after 10,000 iterations, the model accuracy is above 0.93, but in 20,000 tests, the test accuracy is only 0.92. The values of subsequent iterations are also unstable, and the overall curve is in a state of twists and turns, and the average accuracy can reach 95%. For this method, we can find that after 5000 iterations, the accuracy value rapidly increased to 0.943, and the subsequent test curve gradually increased and then quickly tended to be flat. Finally, after 15,000 iterations, the ultra-high-precision value of 0.98 can be basically maintained, exceeding the target value.

4.4. Flame Modal Test. The flame animation generated by us will generate different modes due to various factors. After 500 tests, count the clear flame patterns that can be captured by ultraclear lens under static and dynamic conditions. In addition, the generation time of incomplete animation (unified as the state when the flame integrity is one-fifth) and complete animation should be counted. We can find that

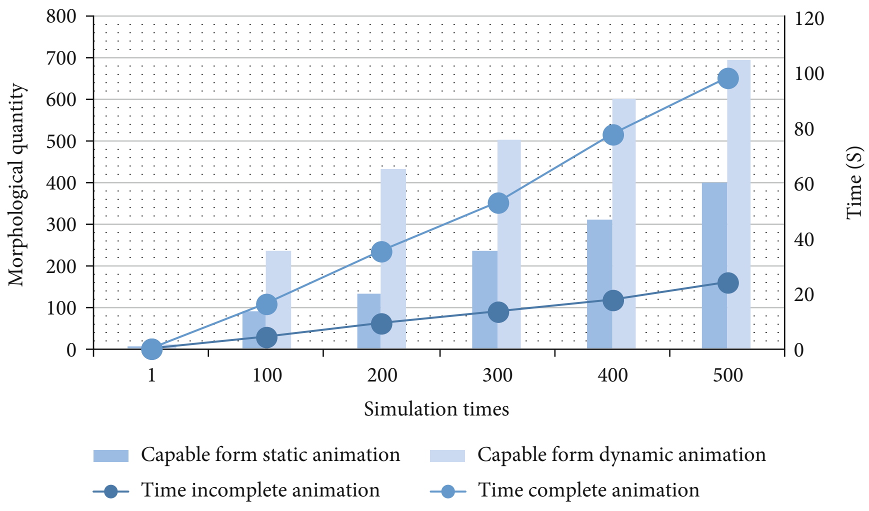

Figure 6: Capture of flame shape.

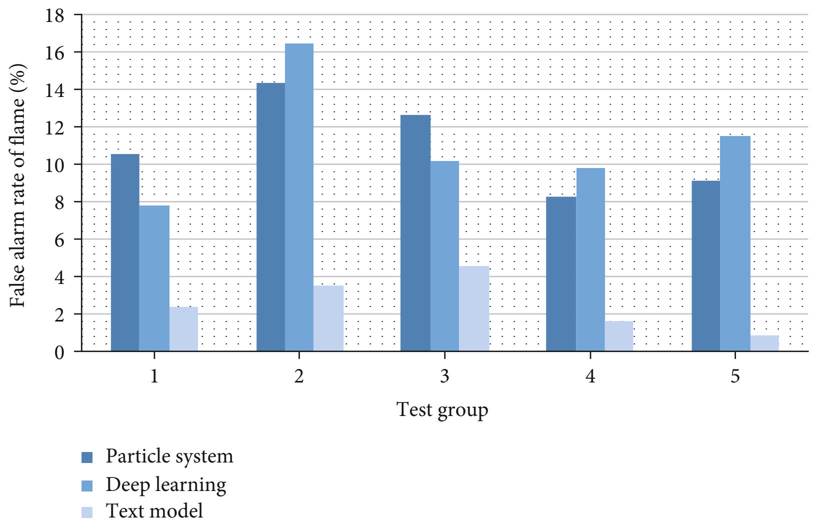

Figure 7: Nonflame identification detection.

with the increase in test times, the flame can capture more forms. The static state can capture up to 399 forms; the dynamic state can capture up to 698 different forms. In the initial state, the first test shows that the generation time of incomplete animation is only 0.05 s, and that of complete animation is 0.18 s, so the animation generation speed is very fast. After 500 simulations, it takes 24.01 s for incomplete animation and 98.21 s for complete animation. This result takes less time than originally expected, indicating that there is a faster generation speed during the test than the initial test, as shown in Figure 6.

4.5. Detection Recognition Rate Effect. After the production of flame animation, we need to use professional tools and software to identify and detect flame animation. A total of 5 test groups were set up, and each group carried out 20 recognition tests on different flame animations, with a total of 100 recognition tests. In order to better highlight the recognition effect of this method, we add the model recognition effect comparison between the unimproved particle system method and the deep learning method, as shown in Figures 7 and 8.

Before the formal test, we pretest the nonflame animation to test the false-positive rate of the three methods. Also, set up 5 sets of tests. From Figure 7, it can be found that the particle system method has the highest false alarm rate in test group 1 and test group 3. Among test groups 2, 4, and 5, the false alarm rate of the deep learning method is the highest, and the highest false alarm degree can reach 16.5%. However, the false-positive rate of this model is the lowest among the five test groups, the highest false-positive rate is only 4.6%, and the lowest is 0.9%. The effect is stable, and the interference degree is the least.

From the formal test in Figure 8, we can find that the recognition rate of flame in the animated video in this model is the highest among the five groups of data, with the highest recognition rate reaching 99.2% and the lowest recognition rate reaching 98.5%. The highest recognition rate of the

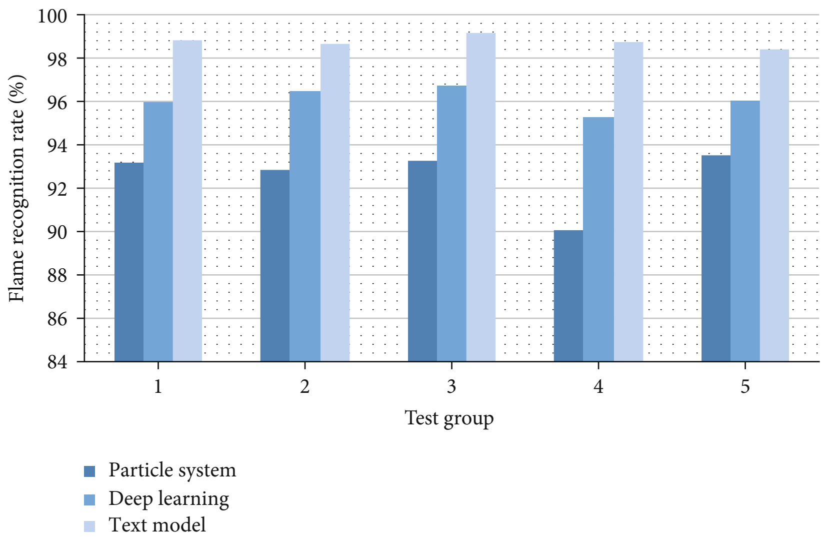

Figure 8: Flame animation recognition and detection.

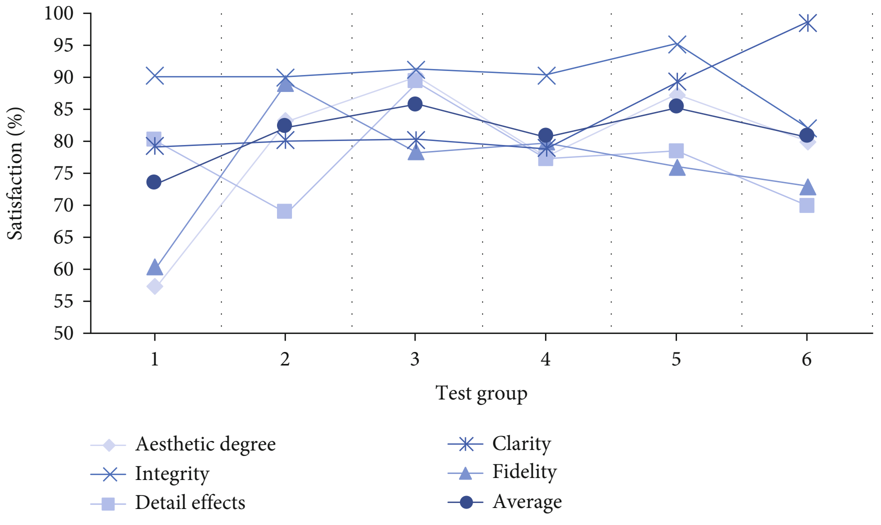

Figure 9: Satisfaction degree of effect.

particle system is 93.6%. The highest recognition rate of the deep learning model is 96.8%, which is slightly higher than that of the particle system. This proves that the flame animation produced in this experimental test is excellent and the effect is excellent.

4.6. Satisfaction Test. Making animation needs to have a certain degree of appreciation and beauty. Therefore, the flame animation completed in our experiment is evaluated from five aspects: beauty, integrity, fidelity, clarity, and detail. Five experts and 25 volunteers were invited to score six groups of flame animations based on this model. Statistical animation effect satisfaction is shown in Figure 9.

We can find that the animation integrity of the other five groups is higher than 90%, except for the integrity of the sixth group, which is 82.2%. The group with the highest aesthetic and detailed effects is the third group, which is 90.23% and 89.34%, respectively. At the same time, the third group is also the group with the highest average evaluation, reaching 85.936%. The second group has the best realistic effect, which can reach 89.23%. Group 6 is the group with the highest clarity, but the overall score is not high. The worst animation effect is the first group, with an overall score of only 73.526%.

5. Conclusion

This paper is located in a three-dimensional space angle to generate computer graphics, using CG technology to simulate virtual reality. This is an applied subject with wide application and rapid development. It intersects with many research fields and has a bright future. In this article, we take flame animation as a demonstration research example, organically combined with a variety of technical methods. In this paper, the animation model and algorithm are analyzed and tested, and the various forms of fireworks are finally captured. In addition, the dynamic and static flame animation simulation pictures are intercepted to verify the application value of the experiment. The specific research results show that based on the deep stripping algorithm and the function of texture mapping, the flame data is visualized, and the professional simulation satisfaction evaluation is carried out. Use professional tools and software to identify and detect flame animation so as to confirm the discrimination degree of flame. It overcomes the limitation of computer performance and calculates and optimizes the error and accuracy. Improve the mathematical and physical equations to reduce the resistance that affects the real-time performance because of the complexity of calculation. The final experimental effect is close to that in nature, and the details are close to reality. Whether in the fields of games, advertisements, or special effects, it is very meaningful to study the construction of irregular real-time animation.

Although this paper simulates flame animation from a new angle and obtains ideal realistic flame effects of various forms, however, more attention should be paid to the optimization of future work. For example, OpenGL rendering efficiency is not excellent enough, and there is room for improvement in the limitation of specific hardware requirements. The generation speed of animation needs to be finely controlled. Also, the delay performance of animation may lead to poor audience perception, which needs further optimization. Improve the interface and the research on the calling effect when the parameters meet all conditions and the lack of consideration of some scenes, such as the simulation when the flame is blocked by obstructions and the firepower is weak. Texture mapping may be distorted. The above description of the shortcomings of this paper will be the focus of future work.

Data Availability

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

Conflicts of Interest

The authors declared that they have no conflicts of interest regarding this work.

References

[1] C. Rui, “Real-time transmission rendering of translucent objects based on deep peeling,” *Modern Computer*, vol. 9, p. 5, 2021.

[2] L. Lulu, C. Shuyue, and W. Liping, “Microfiber recognition algorithm based on deep feature fusion and reconstruction,” *Modern Electronic Technology*, vol. 45, no. 1, p. 6, 2022.

[3] Z. Xuemin, C. Xiaodong, and L. Yumin, “Flame detection algorithm based on target tracking and multi-feature fusion,” *Television Technology*, vol. 37, no. 15, pp. 205–210, 2013.

[4] Z. Jinhua, Z. Jian, and D. Haifeng, “A flame recognition algorithm based on video multi-feature fusion,” *Journal of Xi’an Jiaotong University*, vol. 40, no. 7, p. 4, 2006.

[5] H. Ang and W. Jiwen, “Fireworks animation simulation with controllable trajectory based on particle system,” *Computer Technology and Development*, vol. 22, no. 7, p. 3, 2022.

[6] W. Biyun, K. Yanghong, and L. Hui, “3D dynamic flame simulation based on particle system,” *Journal of Wuhan Institute of Technology*, vol. 40, no. 1, p. 6, 2018.

[7] L. Sun Mingyan, L. X. Weiwei, and W. Enhua, “Approximate soft shadow algorithm combining deep stripping with GPU,” *Chinese Journal of Image and Graphics*, vol. 15, no. 9, pp. 1391–1397, 2010.

[8] H. Yi, W. Zhaoqi, and Z. Dengming, “Research on flame animation generation method based on Level-Set,” *Computer Research and Development*, vol. 47, no. 11, pp. 1849–1856, 2010.

[9] Y. Wang Peng and C. H. Mengting, “A new method for rapid simulation of flame animation,” *Computer Age*, vol. 9, no. 2, pp. 23-24, 2012.

[10] Z. Tingting and W. Wenbin, “Shengshi Birthday-C4D and AE are matched to realize particle special effects animation in three-dimensional space,” *Film and Television Production*, vol. 11, no. 1, pp. 51–61, 2021.

[11] W. Mengwei and M. Xiuli, “Interactive texture mapping based on intermediary,” *Electronic Measurement Technology*, vol. 43, no. 12, p. 6, 2020.

[12] W. Jun and W. Jiwen, “Real-time flame simulation based on N-S equation and texture mapping,” *Computer and Modernization*, vol. 6, no. 12, p. 4, 2013.

[13] H. Xiangxiang, Z. Quansheng, and J. Wanshou, “Rapid visibility detection without setting deviation in multi-view texture mapping,” *Journal of Surveying and Mapping*, vol. 49, no. 1, p. 16, 2020.

[14] W. Meirong, T. Ze, and W. Xiaocheng, “Research on texture mapping modeling method based on UML-SystemC,” *Aviation Computing Technology*, vol. 50, no. 6, p. 5, 2020.

[15] L. Heng, L. Zhong, and Y. Ce, “Texture mapping of 3D face model based on feature constraint and deformation,” *Industrial Control Computer*, vol. 1, no. 2, p. 5, 2022.

[16] W. Jidong and P. Mingyong, “Orientation optimization algorithm of 3D printing model based on deep stripping,” *Journal of Computer Aided Design and Graphics*, vol. 30, no. 9, p. 7, 2018.

[17] D. Dingsheng, “A real-time flame simulation algorithm based on GPU,” *Journal of Jilin Normal University: Natural Science Edition*, vol. 40, no. 2, p. 5, 2019.

[18] G. Bic, Z. Guohua, and N. Junyong, “Early fire detection algorithm based on irregular patterns of flames and hierarchical Bayesian networks,” *Journal of Fire Safety*, vol. 45, no. 4, pp. 262–270, 2010.

[19] Z. Wenhui, S. Kwok, and L. Zian, “Improved direct 3D flame simulation algorithm based on particle system,” *Journal of Guilin University of Electronic Technology*, vol. 34, no. 1, p. 5, 2022.

[20] Z. Juan, “Texture mapping flame simulation technology under the background of particle system,” *Electronic Technology & Software Engineering*, vol. 16, pp. 69-70, 2022.

[21] P. Chuanlai, “Research on laser marking technology of curved surface parts based on two-step texture mapping,” *Journal of Datong University: Natural Science Edition*, vol. 37, no. 4, p. 3, 2021.

[22] C. Jiaqing, Z. Bing, and S. Yinglei, “Flame recognition based on RGB statistical color model,” *Journal of Jiangsu University of*

Science and Technology: Natural Science Edition, vol. 31, no. 2, p. 7, 2017.

[23] W. Huaibing, X. Xin, and C. Yaojie, “Three-dimensional simulation of ship traveling waves based on particle system,” *Computer Technology and Development*, vol. 31, no. 8, p. 5, 2021.

[24] L. Ying, L. Qian, L. Daxiang, and Y. Wenzong, “Texture mapping algorithm of cultural relics based on PTM model,” *Computer Engineering and Application*, vol. 56, no. 12, p. 6, 2020.

[25] Y. Luqing, “Image metonymy and metaphor of emotional representation in animated films–taking the image of Little Nezha in Ne Zha as an example,” *Journal of Wuyi University*, vol. 40, no. 2, p. 6, 2021.