

Research Article

Fluid Equation-Based and Data-Driven Simulation of Special Effects Animation

Yujuan Deng 

School of Media Arts and Communication, Nanjing University of The Arts, Nanjing, Jiangsu 210013, China

Correspondence should be addressed to Yujuan Deng; d2003201@nua.edu.cn

Received 15 October 2021; Accepted 10 November 2021; Published 22 November 2021

Academic Editor: Miaochoao Chen

Copyright © 2021 Yujuan Deng. 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 paper analyzes the simulation of special effects animation through fluid equations and data-driven methods. This paper also considers the needs of computer fluid animation simulation in terms of computational accuracy and simulation efficiency, takes high real-time, high interactivity, and high physical accuracy of simulation algorithm as the research focus and target, and proposes a solution algorithm and acceleration scheme based on deep neural network framework for the key problems of simulation of natural phenomena including smoke and liquid. With the deep development of artificial intelligence technology, deep neural network models are widely used in research fields such as computer image classification, speech recognition, and fluid detail synthesis with their powerful data learning capability. Its stable and efficient computational model provides a new problem-solving approach for computerized fluid animation simulation. In terms of time series reconstruction, this paper adopts a tracking-based reconstruction method, including target tracking, 2D trajectory fitting and repair, and 3D trajectory reconstruction. For continuous image sequences, a linear dynamic model algorithm based on pyramidal optical flow is used to track the feature centers of the objects, and the spatial coordinates and motion parameters of the feature points are obtained by reconstructing the motion trajectories. The experimental results show that in terms of spatial reconstruction, the matching method proposed in this paper is more accurate compared with the traditional stereo matching algorithm; in terms of time series reconstruction, the error of target tracking reduced. Finally, the 3D motion trajectory of the point feature object and the motion pattern at a certain moment are shown, and the method in this paper obtains more ideal results, which proves the effectiveness of the method.

1. Introduction

Ever since the birth of computer graphics, people have wanted to depict the real world through computers. Movie special effects, video games, industrial production, virtual reality, etc. are all inseparable from the need for virtual scene simulation. For example, in the movie manufacturing industry, due to the cost and other reasons, many shots cannot be achieved in the real world through real scenes, such as floods, tsunamis, mudslides, and volcanic eruptions, when the animator is required to simulate and realize them using computer special effects technology. In addition, computer animation technology can also produce some scenes that do not exist in the real world, so that people's endless imagination can be realized and expressed [1]. With the development of technology, people have higher and higher requirements

for the realism of the simulated scenes they see. In addition to the use of professional 3D modeling and light rendering techniques, the realism of computer animation also relies on the accurate portrayal of the physical movement of the simulated objects. Computational physics also requires the use of computers for physics simulation, solving physics equations by numerical methods and pursuing high-precision numerical simulation of physics [2]. Then, through the traditional fluid simulation method, the corresponding training scene is designed and simulated, the training data is collected, and the overall training set is established. Considering the complexity of physics numerical simulation and the limited computing resources and time, to meet the strict requirements of real-time simulation and interactivity in virtual reality and other application scenarios as much as possible, physics-based computer animation technology has

emerged, whose goal is to improve the simulation speed and obtain visually realistic animation effects by designing and deriving simpler physics models at the expense of physical accuracy [3]. The goal is to improve the simulation speed and obtain visually realistic animation effects by designing and deriving simpler physical models at the expense of physical accuracy.

The emergence of digital technology brings a broader development prospect for the design and creation of animation, allowing animation design and creation to have a richer form of expression, and the emergence of animation production software, including two- and three-dimensional and CG animation production software, has largely reduced the technical threshold of animation production [4]. The development of digital technology has brought more options for animation creation. Digital technology runs through the whole animation creation process, including creator's conception of the picture depiction and production, as well as the formal creation of animation, finally forming a complete animation work. With its many advantages, digital technology is bound to become the most powerful tool for animation art creation. On the other hand, the animation industry is gradually emerging in the background of the digital media era, which can be said to be the sunrise industry in the global scope at present [5]. While promoting its space development, it also brings a bright future for the relevant practitioners and application enterprises. This shows that digital technology is constantly promoting the progress of the animation industry, and the animation industry is also better serving our production life; the development of digital technology will certainly make the animation industry to a new peak. 3D motion capture technology is used in various fields and has brought great convenience to people's life. The application of this technology not only stays in the shallow visual but also hopes to obtain more valuable information from the image video, such as the motion data of the target object, the shape, and pose, and give the corresponding conclusion, to solve the practical problems in life. For motion capture technology, the most researched motion capture is related to the human body. In addition to the motion capture with the model, the feature capture of point feature objects without a model can also be performed, thus making the application of motion capture technology more extensive.

In the current field of computer graphics, there are many research hotspots and difficulties in fluid animation simulation methods for Eulerian grids. This subsection provides further analysis on the research of fluid simulation acceleration algorithm for the Eulerian grid method. From the previous section, the effective acceleration of large-scale, high-quality, and high-precision fluid animation simulations has been one of the research difficulties in this field. The most direct way to accelerate the Eulerian grid method of fluid animation simulation is to use more efficient numerical methods to improve the computational efficiency of fluid simulation. In the traditional fluid simulation framework, the main computational bottleneck is the projection step Poisson equation solution process. Although the common preprocessing common choke gradient (PCG) method for

projection step solution is fast and easy to implement, it is limited by its convergence efficiency when dealing with large-scale, high-resolution grid discrete scenarios. Many excellent research works have been applied such as Milt-Grid as a preprocessing method for the common choke gradient method, which enables users to get more obvious convergence acceleration when simulating large-scale scenes. Also, some researchers have further improved its computational speed by using graphics processing units (GPUs) with parallel algorithms.

2. Current Status of Research

The data-based training approach is also one of the classical approaches of precomputed data methods. For example, Zhang et al. used the SPH method to computationally generate fluid particle data, establish the overall state graph, and allocate the computational access of the graph with the best matching pattern to achieve game acceleration on the cell phone side [6]. Jiang et al., on the other hand, treated the fluid simulation process of the SPH method as a regression problem and trained the regression forest by historical data in the preprocessing step [7]. After completion, the states of the neighboring particles are input, and then, the approximate velocity value of the current particle in the next frame is quickly obtained according to the regression forest, and a better acceleration effect is obtained. However, these methods are limited to the Lagrangian particle method, which sacrifices the simulation accuracy, and the extrapolation ability of the algorithm is weak, which is not universal for different scenarios [8]. The above simulation methods combined with fluid data precomputation can effectively use the historical computational data and provide a new acceleration tool for real-time fluid computation, which is a hot spot in the current fluid animation simulation technology research, but there are still many key points that need an urgent breakthrough [9]. For example, in the model reduction method, the accuracy loss is largely due to the dimensionality reduction, and it still needs a lot of computational resources and time to solve the large linear equations formed by the discretization of the Poisson equation in the projection step; the keyframe interpolation method is limited to the interpolation between similar keyframes; the data training method is still limited to the Lagrangian particle method, and the simulation and computational accuracy and physical accuracy are not guaranteed [10]. In general, the existing data-based fluid simulation methods have two main drawbacks: one is that they largely sacrifice fluid simulation accuracy, and the other is that the algorithms limit the universality of fluid scenarios [11]. The second is that the existing methods do not provide an in-depth analysis of the main computational bottlenecks in fluid simulation and lack effective and fast solution methods [12].

More specifically, how to design special morphological difference recognition models using existing deep learning techniques? How to extend the simulation results based on low precision morphological correction to high precision simulation scenarios? These are important questions that need to be addressed in future research. In terms of spatial

3D reconstruction, firstly, according to the different feature attributes of the object, a targeted target detection algorithm is selected to find the center coordinates of the feature points [13]. The traditional target detection algorithm cannot find the effective feature centers, and the reconstruction effect is poor [14]. For objects with markers, the Hough gradient method based on the optimal ranking is used; for unmarked objects, the elliptic least-squares fitting method based on contour extraction is used to find the feature centroids. In terms of matching, the affine transformation algorithm based on the polar line constraint is proposed in this paper [15]. Firstly, aberration correction and stereo correction are applied to the original image, to eliminate aberrations and keep the corresponding image blocks in the left and right images on the same level. Then, the mapped image blocks of the feature points in the left image are generated by the affine transformation function, and the iterative search matches the feature points in the right image, constraining the point finding process within the polar lines. Finally, according to the projection matrix calibrated by the camera, the spatial position of each corresponding point is calculated in turn to reconstruct the 3D model [16].

The dimensional change from two to four dimensions is carried out throughout the creation process. From creator's perspective, we analyze the use of spatial dimension expansion and compression in the creation and provide a basis for exploring the inevitable connection between the technical basis and art form of 3D animation; from audience's perspective, we analyze the cognitive and experiential dimensions of 3D animation and explore the inner influence of visual principles and aesthetic psychology on the creation of 3D animation art; from the economic and cultural dimensions, we analyze the connection between commercial and cultural phenomena in the development of 3D animation. From the economic and cultural dimensions, we analyze the connection between commercial and cultural phenomena in the development of 3D animation. In the research of 3D animation, starting from the concept of dimension, different angles and levels of discussion are essential to clarify the relationship between 3D animation technology, artistic creation techniques, and basic theory construction, which is of great significance to the practice of animation creation and theoretical research.

3. Fluid Equations and Data-Driven Simulation Analysis of Special Effects Animation

3.1. Fluid Equations and Data-Driven Construction. In classical physics, Navier-Stokes's equation often used to describe the motion of viscous incompressible fluids and their momentum conservation relations, which was established by Laude-Louis Navier et al. Its specific vector form is formulated as:

$$\frac{\partial u}{\partial t} = -u \cdot \nabla u + \frac{1}{\rho} \nabla p - \varepsilon \nabla \cdot \nabla u - f, \quad (1)$$

$$\nabla \cdot u = 0. \quad (2)$$

Two of them, namely, the convection step and the projection step, are the substeps of most interest to all researchers. Earlier Eulerian grid-based fluid computational frameworks used finite difference methods to solve the convection step, but such methods could not obtain stable numerical simulation results, and due to the nonlinearity of the convection equation, a small-time step had to be used for the discrete difference computational format, which also made the computational speed and efficiency much less [17]. Automatically synthesize 3D models through software. Both image modeling technology and 3D scanning technology have greatly improved the modeling efficiency. However, image modeling technology also has a major flaw, that is, it is impossible to fine-tune the details of the model, and it is difficult to get what we think. For this reason, Jos Stam first proposed an unconditionally stable method for solving the convective step. This work applies the Semi-LaGrange method to solve the convective term of the fluid momentum equation, treating the fluid as a particle in the specific convection calculation and calculating the velocity and pressure using the velocity backtracking approach. This method of calculating the convective terms in the Eulerian grid framework combined with the Lagrangian particle perspective is also gradually becoming the standard method for solving the convective terms of fluid animation by the Eulerian grid method.

$$q^{n-1}(x_{t-1}) = q^n(x_t + \Delta t u^n(x_t)). \quad (3)$$

The numerical discretization of continuous variables is essential for the application of computer numerical methods to fluid animation simulations. It is very important to know how to discretize a problem and in what format, because different discretization methods imply different discretization formats, and there are different ways of storing information and calculating problems, and at the same time, there are bound to be the same continuity problems with different accuracy of numerical solution results when different discretization methods are applied. This paper introduces the two most used discretization methods uniform grid method and interleaved grid method for computer solutions such as constant and partial differential equations. First, the uniform grid method is the most common and simplest discretization format for continuous variables and fields. For a one-dimensional variable, this means that it divided into linear line segments. For example, for a continuous variable x , where $x \in (0, 1)$, we can simply divide it into ten segments with grid point values of x_0, x_1, \dots, x_{10} . The uniform grid division of the variables and fields in the two-dimensional case is shown in the left subfigure of Figure 1, where we can discretize the entire two-dimensional space into isometric square grids according to the Cartesian coordinate system (x, y -axis). Similarly, the three-dimensional case of uniform grid division can be seen in the left subfigure of Figure 1, which also means that the entire three-dimensional space is divided into an equiangular square grid. Then, based on this uniform grid discretization method, we can transform spatial variables such as velocity

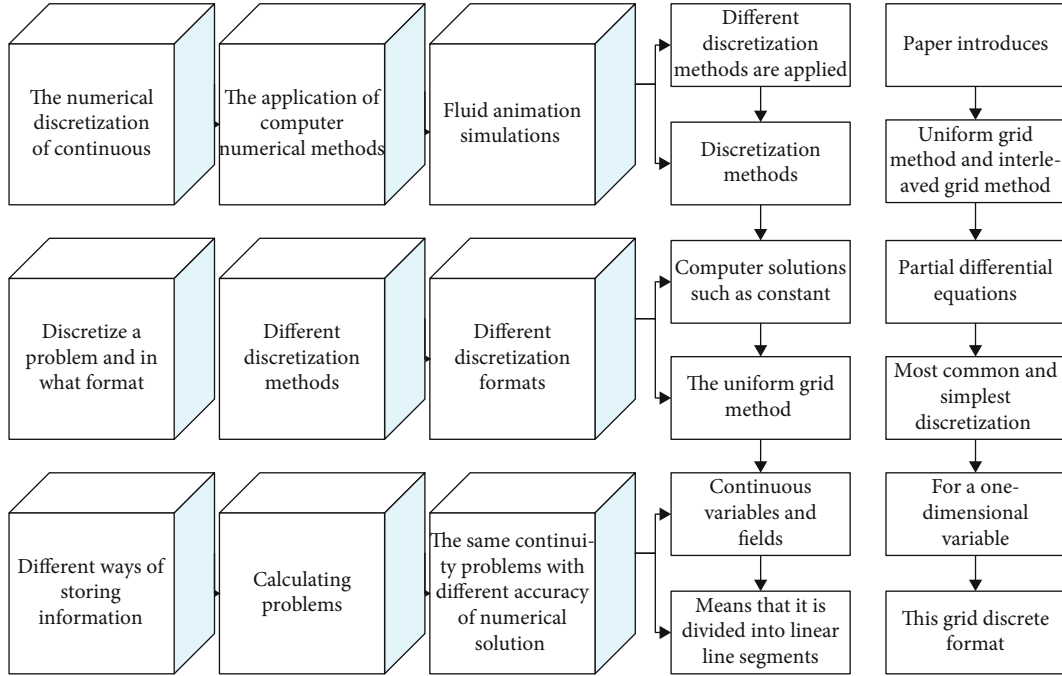


FIGURE 1: Fluid equation framework.

field u into discrete forms in discrete coordinates and store the corresponding values in the center of the grid.

On the other hand, for algorithms sensitive to storage space, this grid format can effectively reduce storage space by replacing vector storage with scalar storage [18]. Most importantly, this grid discrete format can effectively improve the accuracy of discrete calculation and avoid the “nonnegligible zero space problem”: the difference value of a function or variable is zero, but the actual value is very numerical. The most typical case is the derivative calculation of the sawtooth function; we use $F(i) = -1$ as an example of the sawtooth function, using the uniform grid derivative calculation formula (4), then $\nabla F = 0$, but in fact, this function is not a constant function with zero derivatives, so the calculation results and the actual function contradict each other phenomenon. The staggered grid method is a perfect solution to this problem.

$$\frac{\partial F(i)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{F(i+1) - F(i-1)}{\Delta x}. \quad (4)$$

However, this method has the disadvantage of being slower than the uniform grid method, because the velocities in the grid need to be computed as components, i.e., for each direction of the variable. The velocities need to be interpolated (usually linear or trilinear).

In the classical framework of the Eulerian grid method, the most time-consuming computational step is the projection step solution process. From the above description, the key to the solution of the projection step is the solution of Poisson’s equation, i.e., the solution of the discrete form of Poisson’s equation. Here, we first further analyze Poisson’s equation formed by the fluid projection step. Equation (5) is the discrete form of Poisson’s equation for a three-

dimensional fluid (under the MAC grid), and in general, the equations for the entire flow field can be reduced to matrix-vector form as follows.

$$Ap = d^2. \quad (5)$$

The matrix A is a symmetric positive definite sparse matrix, and in the three-dimensional case, each row of A has seven nonzero elements, so it can be called a 7-Point Laplacian Matrix, and similarly, in the two-dimensional case, each row of A contains five nonzero elements, so it is called a 5-Point Laplacian Matrix. The following equation is the matrix-vector expansion of the projection step Poisson equation in the two-dimensional case.

$$\begin{bmatrix} A_1 & I & I_1 & \cdots \\ -I & A_2 & I & \cdots \\ \cdots & I & A_3 & I \\ \cdots & \cdots & I & A_N \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}. \quad (6)$$

Therefore, we can see that solving the fluid projection step Poisson equation is solving the above linear equation system. Among the traditional numerical methods for solving linear systems, there are two main types of solutions, one is the direct solution method, such as the Gaussian elimination method and the LU decomposition method. Judging the quality of a scene environment mainly depends on whether the scene can provide a strong sense of substitution and bring people into the animation. The scene in this cartoon is very large, because it is an underground mine, crisscrossed and densely crossed, so the production of the scene model is the most important and the most time-

consuming. These methods are suitable for solving linear equations of small size and dimensionality, but cannot be applied to linear systems of large dimensionality. For example, in the three-dimensional case, the dimension of the linear system of equations is formed by solving the fluid system with a grid accuracy of 1024, i.e., the order of the matrix A will reach 1024×1024 , and such a huge linear system cannot be effectively handled by the traditional direct method.

$$A = \begin{bmatrix} A_1 & I & \cdots & \cdots \\ -I & A_2 & I & \cdots \\ \cdots & I & A_3 & I \\ \cdots & \cdots & I & A_N \end{bmatrix}. \quad (7)$$

To realize the numerical simulation of the fluid, the N-S equation needs to be decomposed, which is usually divided into three steps, namely, solving the advection term (advection), solving the external force term (body force), and solving the projection term (pressure projection).

$$\frac{Dq}{Dt} = 1, \quad (8)$$

$$\frac{\partial q}{\partial t} = f, \quad (9)$$

$$\frac{\partial v}{\partial t} - \frac{1}{\rho} \nabla \cdot \rho = 1. \quad (10)$$

The different order in which these three steps are performed can have an impact on the results of the numerical simulation. Among them, solving the advection term needs to ensure that the calculation is performed in a velocity field with zero scatter, so during the simulation, solving the advection term usually acts on the scatter-free velocity field after solving the projection term.

The advantage of using the Eulerian method for fluid simulation is that the fluid moves relative to the mesh, so there is no mesh distortion problem, and the incompressibility of the fluid can be guaranteed in a reasonable computing time. The disadvantage is that the advection term in the control equation needs to be handled; otherwise, it is easy to cause numerical dissipation. In addition, it is very challenging to construct regular meshes with irregular or complex geometries. When using the Eulerian method to calculate boundary problems, the fluid interface may cross the cell, so additional algorithms are needed to track and reconstruct the moving interface, as shown in Figure 2.

Non-Newtonian fluids are widely present in the real world, with a wide variety of types and different physical properties. Among them, viscous non-Newtonian fluids can be classified into shear-thinning fluids, shear-thickening fluids, and Bingham plastic fluids according to the intrinsic structure relationship, as shown in Figure 2. The kinematic viscosity of shear-thinning fluids is inversely proportional to the shear rate, the kinematic viscosity of shear-thickening fluids is proportional to the shear rate, and Bingham plastic fluids exhibit flow only after the shear

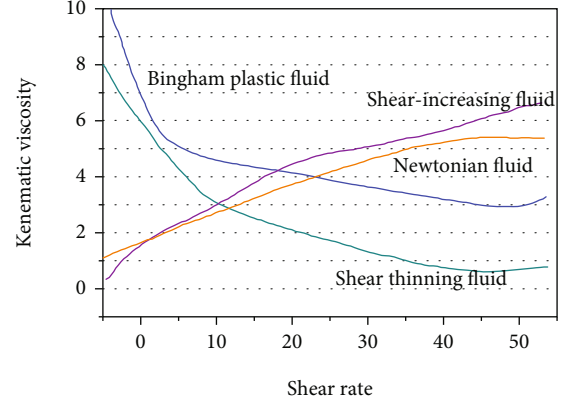


FIGURE 2: Schematic diagram of the intrinsic structure relationship for non-Newtonian fluids.

rate is greater than the minimum critical value, and the kinematic viscosity is inversely proportional to the shear rate.

Since the fluid animations produced by the discrete computation of meshes with different resolutions vary greatly, even under the same simulation parameters, different simulation accuracies (resolutions) can produce vastly different simulation results; therefore, many traditional fluid animation simulation algorithms optimize the simulation results with high accuracy, among which the typical one is the fluid guiding method. In recent years, many excellent algorithms for fluid animation have emerged from the academic community around the idea of using precomputed fluid data to rapidly generate new fluid results, which can significantly accelerate the simulation process by precomputed data. The fluid simulation framework is based on the traditional Eulerian grid method fluid simulation framework, which also decomposes the fluid simulation into four terms (convection step, external force step, diffusion step, and projection step) for a step-by-step solution.

$$\frac{\partial u}{\partial t} = -u \cdot \nabla \|u\| + \frac{1}{\rho} \nabla \cdot \rho - f. \quad (11)$$

It is also necessary to satisfy the incompressibility of the fluid.

$$\nabla \cdot u = 1. \quad (12)$$

There is a relatively stable mathematical relationship between the known data of the projection step Poisson equation and the requested data, so we can try to build a deep learning model instead of this equation, using the known data as the input data of the whole deep learning model (Input Data) and the requested data as the output data of the deep learning model (Output Data) [19]. In the traditional fluid simulation framework, the main computational bottleneck is the process of solving the Poisson equation of the projection step. Although the commonly used preprocessing conjugate gradient method for projection step calculation has the characteristics of fast convergence and easy implementation, it will be limited by its convergence

efficiency when dealing with large-scale, high-resolution grid discrete scenes. We first construct the corresponding feature vectors to extract the training samples based on the data features of the projection step. Then, we design the corresponding training scenarios and simulate them by traditional fluid simulation methods, collect the training data, and build the overall training set.

$$\left(\frac{4p_n(x(i+1, j+1)) + p_n(x(i-1, j-1))}{h^2}\right) = \frac{p}{\Delta t} \left(\frac{u_n^f(x(i, j))}{h}\right). \quad (13)$$

After constructing the input and output feature vectors of the deep learning model, the training samples can be constructed from this structure to generate the training set. In this chapter, the traditional Eulerian grid method is used for training data generation: firstly, a fluid scene is designed for numerical simulation, and the preprocessing conjugate gradient method (PCG) is used as the solution method for the Poisson equation in the projection step, and the incomplete Cholesky decomposition (IC) is used as the preprocessing operator.

3.2. Special Effects Animation Simulation Design. 3D scanning technology is to scan actors, animals, or objects in three dimensions through a 3D scanning system and transmit each coordinate point and color information of the scanned object to the computer to get a 3D color digital model of the scanned object. Compared with geometric modeling methods, 3D scanning technology is simpler and more convenient, but 3D scanning technology requires quite high-cost hardware equipment and limits the artistic creativity of animators. The technique is to take multiangle shots of the model object to be created by multiple digital cameras, to obtain digital photos of the object from all angles and automatically synthesize the 3D model through software. Both image modeling technology and 3D scanning technology make the modeling efficiency greatly improved, but image modeling technology also has a major drawback, that is, it is impossible to fine-tune the model details, and it is difficult to get the accurate model we want.

Before drawing the subplot, we must sort out the whole story according to the script narration, then design and arrange the screen, and finally, draw the footage in Photoshop through the hand drawing board and mark out the movement of the characters, different combinations of footage express different knowledge points, and then make the subplot into a complete presentation in the subplot table [20]. This animation belongs to science education film, divided into two sections, the first section is for the mine fire self-rescue, escape training education, and the second section is for the occurrence of the roof disaster disposal methods. Each section introduces the precursors of the accident, the process of the disaster, and the techniques of self-rescue and escape after the disaster, with a total of about 130 subshots, drawn with Photoshop and Sai, etc. The script dissected in detail and corresponds to the narration as much as possible.

Scene modeling, scene design drawings as a reference, based on which the scene modeling. The scene mainly reflects the environment atmosphere of the cartoon, the environment, and the plot and characters together constitute three important factors of an animated film, judging the goodness of a scene environment mainly depends on whether this scene can provide a strong sense of immersion and bring people into the animation. The scene of this animation film is very large, because it is an underground mine, crisscrossing and crossdense, so the production of the scene model is the most important and time-consuming. There are completely different information storage methods and calculation problems. At the same time, the same continuity problem will inevitably occur. The application of different discretization methods will result in numerical solution results with different precisions. At the same time, we need to consider the optimization problem and try to reduce the number of faces of the scene. Some small parts or decorative things in the model can be mapped to reduce the number of faces of the model, as shown in Figure 3.

The algorithm is a long-term target tracking algorithm that describes and identifies targets by shape analysis. First, the target in the first frame of the video is manually selected; second, the target bounding box is determined by region segmentation; then, the tracker and detector are updated by the learning module; finally, the shape transformation is defined, the set of multiscale level functions is constructed, and the deformation of the target is constrained by the set of functions. On the one hand, this method requires pre-learning target samples and manually marking the target to be measured in the first image frame; on the other hand, it cannot effectively deal with the deformation and occlusion of the target and cannot maintain the accuracy and stability of the data for the experimental environment where the difference between the target and the background is not obvious.

$$b = v_i^* - 2(d-2)\Delta t \sum_{ib} \frac{\mu_b m_{ib}}{p_i^2}. \quad (14)$$

A particle system is an indispensable part of digital animation; using a particle system can make us get twice the result with half the effort; many films with waves, fireworks, debris effects, etc. are done by using a particle system. In a particle system, each particle has its own birth time, survival time, and death time, a very free life cycle, its motion path, and the ability to apply a texture to each particle or replace it with its model. This is very useful for large scenes.

$$\frac{p_{\text{rigid}}^0 + p_{\text{rigid}}}{\Delta t} = p_{\text{rigid}} \nabla v_{\text{rigid}}^{\text{next}}. \quad (15)$$

The experimental object in this paper is a point feature object, which needs to extract the contour information of each point on the surface. In the extraction process, all edge pixel points of the image randomly sampled to determine the target, but errors often occur due to the presence of noise. Therefore, before the target detection, the nonlocal

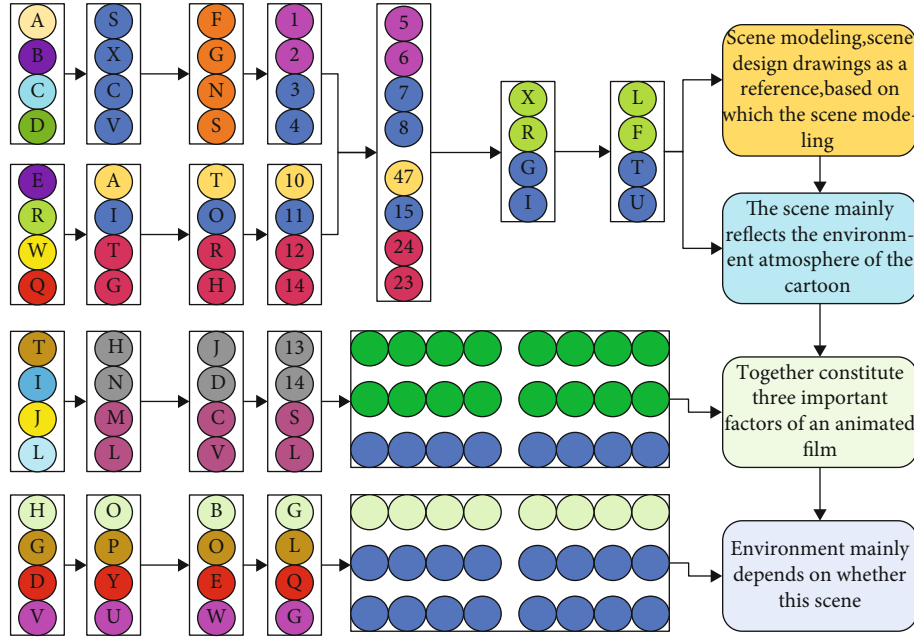


FIGURE 3: Fast projection method for signal processing.

Euclidean median filtering algorithm is used to eliminate the Gaussian noise in the image, and then, the morphological filtering function is called for image edge enhancement, and the following two methods are used for target detection and feature extraction.

$$\begin{cases} y = k_{KL}x - d_{KL}, \\ y = k_{KM}x - d_{LM}. \end{cases} \quad (16)$$

The role of the computer in creation is not simply the intervention of physical tools, but as the continuation and reinterpretation of creator's thinking. The creative intention is realized through the software program running on the hardware platform, making creator's creative process more indirect and operational. Physics-based fluid animation simulation is a very widely used research area, and the study of acceleration algorithms for simulation has become a popular direction in computer graphics. It has significant prospects for development in film and television special effects production, game scene production, and various applications of virtual reality and mixed reality, which are relatively popular recently. Meanwhile, due to the rapid development of big data and machine learning technology, especially the increasing modeling capability of deep neural networks in deep learning, it also provides a new way for efficient and high-quality simulation of computer fluid animation simulation, as shown in Figure 4.

With advances in technology, people have gained unprecedented freedom to create 3D virtual worlds [21]. The reduced cost of input technology means that everything in the real world can be reproduced and duplicated in the virtual world. People's instinct for a better life inspires creators to use 3D technology as a tool and means of artistic creation. 3D animation uses technology to create images that fit

audience's everyday experience: the use of three-dimensional modeling technology to provide close to the actual life of the scene props and three-dimensional space structure; the use of three-dimensional coloring technology to create a realistic image with the texture of real people; the use of three-dimensional movement technology to simulate the human body structure relationship to generate expression and movement; and the use of three-dimensional rendering technology to create a rich light and shadow effects.

At the level of tools, the technological revolutions in history have promoted the advancement of tools that are the strengthening and extension of human somatic functions, such as the advancement of mechanical technology is the strengthening of human limb functions, and the advancement of telecommunication technology is the strengthening of human audio-visual functions. Computers, on the other hand, are more complex and versatile than previous technological tools and can be seen as a complement to and extension of the functions of the human brain. The role of computers in creation is not simply a physical tool intervention, but as a continuation and reinterpretation of creator's thinking.

4. Analysis of Results

4.1. Fluid Equations and Data-Driven Performance Results.

At the level of tools, the technological revolutions in history have promoted the advancement of tools that are the strengthening and extension of human somatic functions, such as the advancement of mechanical technology is the strengthening of human limb functions, and the advancement of telecommunication technology is the strengthening of human audio-visual functions. Computers, on the other hand, are more complex and versatile than previous technological tools and can be seen as a complement to and

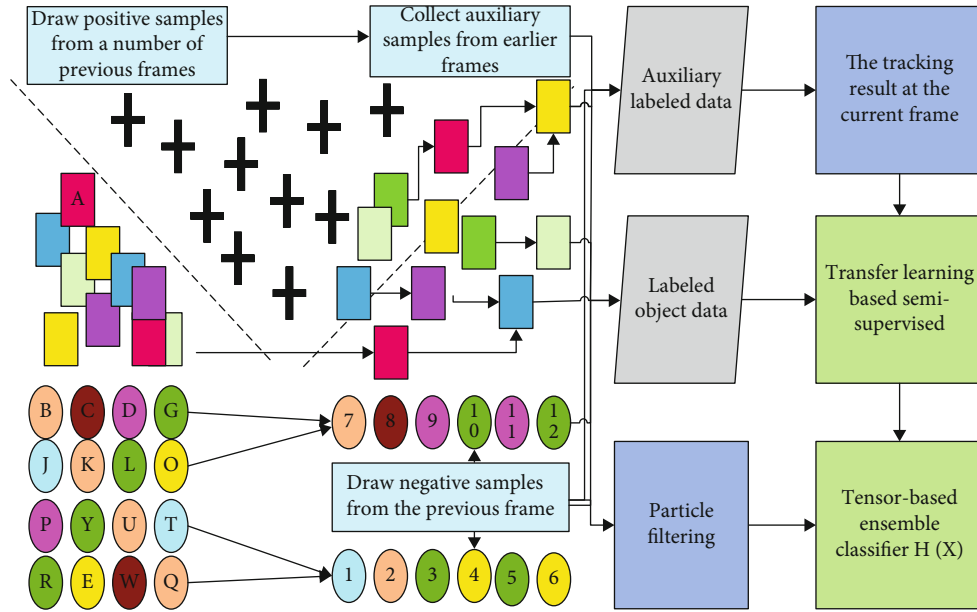


FIGURE 4: Model building.

extension of the functions of the human brain. Respectively, non-Newtonian fluid dam burst and column collision, Newtonian fluid dam failure, and Newtonian fluid water block entering water. These scenes cover a wide range of fluid phenomena from strong splashes to smooth surface waves, from fluid-solid coupling to liquid-liquid interaction. The role of the computer in creation is not simply a physical tool intervention, but as a continuation and reinterpretation of creator's thinking. The creative intent is realized through software programs running on the hardware platform, making the creative process more indirect and manipulative for the creator. As one of the important branches of modern digital art, 3D animation is also like this. Art creation is carried out in the context of technology, which is expressed in the technical existence of art forms and the technical process of art activities. In 3D animation, the function of spectacle rises to a dominant position, which is reflected in the simplification of narrative and its service to the visual expression of spectacle, weakening the depth of thought centered on the ultimate concern of human beings.

The unique flow behavior of non-Newtonian fluids is usually due to shear caused by contact with solids, so the boundary treatment between fluid and solids plays a crucial role in showing the physical properties of non-Newtonian fluid simulation. The established non-Newtonian fluid simulation algorithms use free slip boundary conditions to handle the flow-solid boundary, and the physical properties of non-Newtonian fluids with rising viscosity due to frictional forces on solids cannot be represented. The first experiment demonstrates the effect of a non-Newtonian fluid smashing a slanted plate in the shape of a rabbit composed of 25746 fluid particles, and the experimental scene is set up as shown in Figure 5.

Fluid phenomena are very rich; low viscosity fluids will flow vividly and produce splashes and splatters; high viscosity fluids will produce surface folds in the process of flow. To

cover as wide a range of fluid phenomena as possible as well as to verify the generality of FluidsNet, three different 3D datasets are applied to FluidsNet in this paper, namely, a non-Newtonian fluid breaching dam hitting a column, a Newtonian fluid breaching a dam, and a Newtonian fluid water block entering the water. These scenarios cover a wide range of fluid phenomena from strong splashes to smooth surface waves and from fluid-solid coupling to fluid-liquid interaction. An example of the initial state of the animation for each training dataset is shown in Figure 6, where each animated simulation is initialized with random sizes for the included fluids, and the water block above the pool in dataset 3 has a random size and location.

In this section, we further discuss the effect of the principal component analysis (PCA) method in the whole data sample optimization process. Taking smoke simulation as an example, we first obtained over 11 billion sets of feature input matrices and vectors in the data sample generation process and reduced the sample size to 8.62 billion by dezeroing and deweightening operations. After the dimensionality reduction by principal component analysis, we finally obtained 4.96 billion sets of training samples. When we use 8.62 billion sets of training data as the training set to train the designed deep convolutional neural network model, it takes more than 52 hours to converge the loss function value to 2.5×10^{-4} , but when using principal component analysis to reduce the dimensionality, i.e., 4.96 billion sets of optimized training samples for deep learning, the training time used is the same convergence accuracy achieved in only 30.5 hours. In terms of prediction accuracy, the values of max-error and mean-error are almost the same for different datasets (relative error less than 2%).

4.2. Special Effects Animation Simulation Results. At the same time, we further tested the correction effect of the low-resolution fluid scene simulation results based on

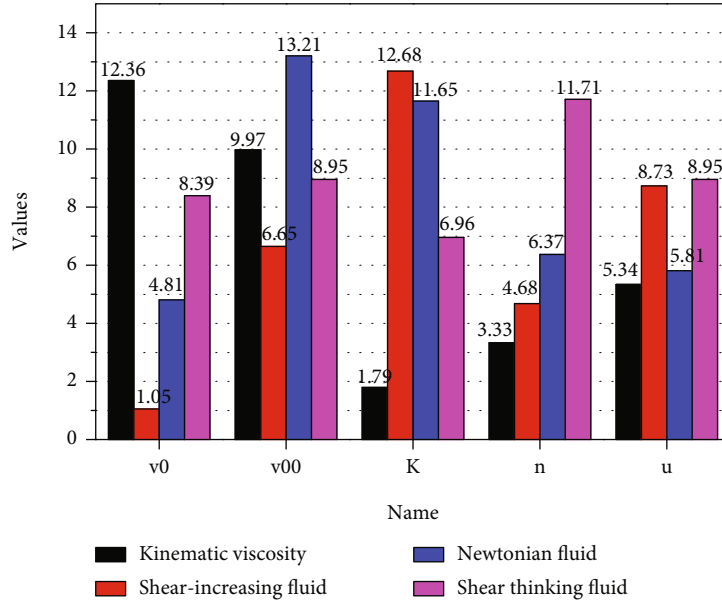


FIGURE 5: Experimental parameters for non-Newtonian fluids.

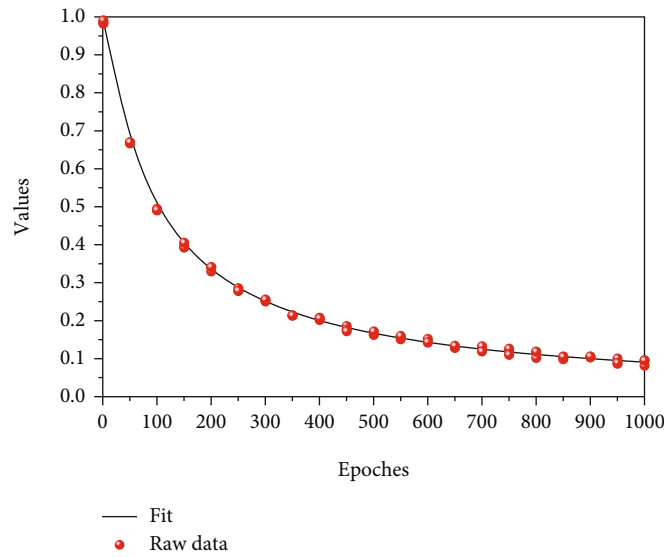


FIGURE 6: Loss functions of different datasets during iteration.

moving smoke with moving obstacles. To solve the problem that other data-driven algorithms require many large scene datasets for model learning, FluidsNet has considerable performance in accelerating large scene animation simulation based on ensuring the reasonableness of prediction. As shown in Figure 7, our algorithm also performs well in the continuously moving smoke scenes. And in the up-and-down fast-moving spherical obstacle scene, our algorithm can very obviously correct the overall morphology and local details of the low-resolution simulation results, including the obstacle (boundary) handling ability.

We have pointed out in the introduction of the morphology correction algorithm based on deep learning models proposed in this chapter that our method can be effectively embedded in detail enhancement postprocessing methods

such as wavelet turbulence. As shown in Figure 7, we first perform morphology correction on the low-resolution simulation results with a resolution of $96 \times 96 \times 96$ and then perform detail synthesis on the corrected low-resolution results using the wavelet turbulence (WT) method to obtain the results of the subplots in the left column of Figure 7 with a synthetic final resolution of $384 \times 384 \times 384$. The middle column of Figure 7 shows the results of the wavelet turbulence method based on the original low-resolution simulation. Our method can capture the overall physical characteristics of the high-resolution simulation more effectively than the high-resolution simulation results on the far right.

Next, the video-based target tracking method is specified, and a linear dynamic model tracking algorithm based on pyramidal optical flow is proposed according to the

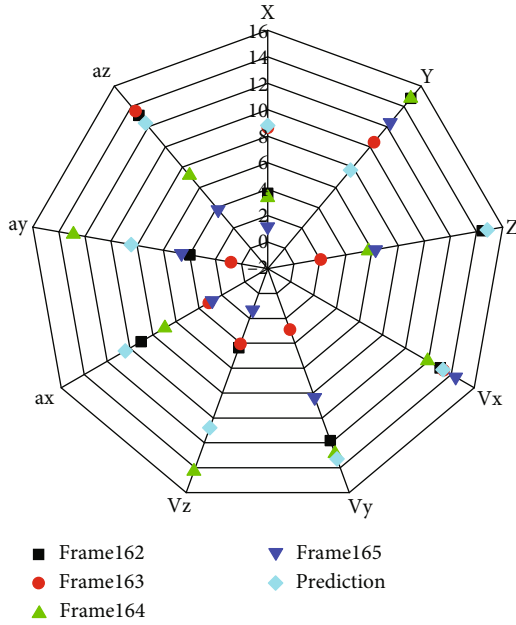


FIGURE 7: Motion parameters and trajectory prediction of feature points.

characteristics of the experimental object; compared with the classical LK optical flow, the tracking accuracy is improved and the tracking error is reduced. Then, the tracking effect is demonstrated by fitting the motion trajectory of the feature points by the least-squares method. Then, the motion trajectory is repaired by forward and reverse tracking methods based on the equation of motion, and the complete 3D motion trajectory is obtained by trilinear interpolation. Finally, the motion parameters of the feature points are calculated, and the moving trend of the target is analyzed and the motion trajectory of the feature points is predicted by the state transfer equation, which enables the 3D motion capture. Several factors affect the accuracy of FluidsNet prediction. The first is the setting of the loss function, i.e., which physical quantity should be selected as the learning direction of FluidsNet. According to Newton's second law, the acceleration field usually considered to be the best physical quantity to measure physical motion. In this section, experiments were conducted with the acceleration field as the predicted output, and the error changes during the training iterations were recorded on dataset 2, as shown in Figure 8. The experimental results show that the SPH fluid simulation method often requires a large compressive force on the fluid particles at the fluid-solid boundary to ensure the incompressibility of the particle system, which in turn leads to values of the acceleration field much larger than 10 g. Therefore, there is a serious numerical fluctuation problem using the acceleration field as a measure, and the neural network is unable to build a reasonable fluid dynamics model with it in the face of unreasonable data.

Learn fluid dynamics from SPH fluid animation simulation data and implement large-scene simulation acceleration. Learn irregular Lagrangian data structures by treating each fluid particle identically and independently, using sym-

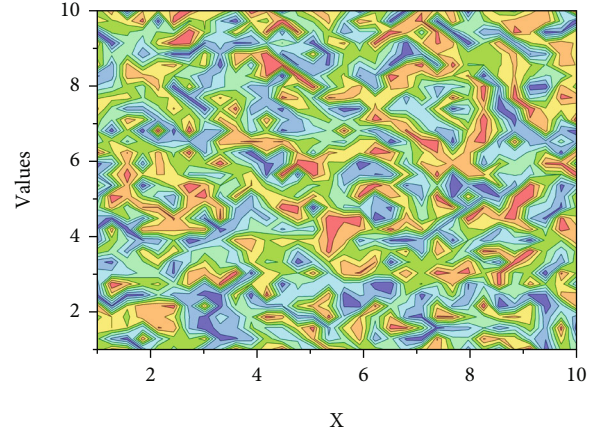


FIGURE 8: Comparison of computational efficiency.

metric functions for the spatial structure and interactions of the particles. Different levels of features are learned by using different network structures to achieve high-quality fluid animations with rich details. Experimental results show that FluidsNet has good accuracy and stability and can reasonably predict the velocity fields of various fluid simulation scenarios with random initial states. In addition, FluidsNet can predict the velocity field of the same type of large scenes with the trained model from small scene datasets, which solves the problem of slow speed and high memory consumption of traditional fluid animation simulation methods for large scene simulation and solves the problem that other data-driven algorithms need many large scene datasets for model learning. FluidsNet has considerable performance in accelerating large scene animation simulation.

5. Conclusion

In response to the problems of large scene simulation in traditional animation methods, which are computationally intensive, slow, and high memory consumption, a data-driven large scene simulation acceleration method is proposed to learn fluid dynamics from SPH fluid animation data and realize large scene simulation acceleration. The method has good accuracy and stability and can reasonably predict the velocity fields of various fluid simulation scenarios with random initial states. The method can predict the velocity field of the same type of large-scene fluid simulation with a trained model from a small scene dataset, which solves the problem of slow-speed and high-memory consumption of traditional fluid animation simulation methods for large-scene simulation and solves the problem that other data-driven algorithms require a large number of large-scene datasets for model learning, and has considerable performance in accelerating large-scene animation simulation based on ensuring the reasonableness of prediction. It has a considerable performance in accelerating the simulation of large scene animation based on reasonable predictions. A boundary treatment method is proposed for non-Newtonian fluid simulation under the prediction-correction method to address the problem of unreasonable boundary conditions in

existing non-Newtonian fluid prediction-correction methods, resulting in the loss of physical properties of the simulated fluid. The method can be applied in the framework of the existing prediction-correction-based non-Newtonian fluid simulation algorithm; by adjusting the parameters, the physical properties of the non-Newtonian fluid can be presented more comprehensively, and the non-Newtonian fluid-solid bidirectional coupling animation with a good sense of realism can be provided.

Data Availability

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

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work was performed as part of the author employment under Nanjing University of The Arts.

References

- [1] X. Yao, J. Zhou, C. Zhang, and M. Liu, "Proactive manufacturing-a big-data driven emerging manufacturing paradigm," *Computer Integrated Manufacturing Systems*, vol. 23, no. 1, pp. 172–185, 2017.
- [2] Y. Feng, X. Huang, R. Hong, and J. Chen, "A multi-dimensional data-driven method for large-size slewing bearings performance degradation assessment," *Journal of Central South University (Science and Technology)*, vol. 48, no. 3, pp. 684–693, 2017.
- [3] W. Rongxiao, C. Bin, Q. Sihang, Z. Zhengqiu, and Q. Xiaogang, "Data driven simulation of polluted gas dispersion using source estimation and particle filter," *Journal of System Simulation*, vol. 29, no. 9, pp. 2100–2109, 2017.
- [4] Z. Y. Jin, L. G. Han, Y. Hu, Q. X. Ge, and P. Zhang, "Low frequency information compensation based data-driven Marchenko imaging," *Chinese Journal of Geophysics*, vol. 60, no. 9, pp. 3601–3615, 2017.
- [5] R. Hao, Q. Ai, and F. Xiao, "Architecture based on multivariate big data platform for analyzing electricity consumption behavior," *Electric Power Automation Equipment*, vol. 37, no. 8, pp. 20–27, 2017.
- [6] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: a survey," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2213–2227, 2019.
- [7] X. Jiang, M. Coffee, A. Bari et al., "Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity," *Computers, Materials & Continua*, vol. 62, no. 3, pp. 537–551, 2020.
- [8] T. Kirchdoerfer and M. Ortiz, "Data-driven computing in dynamics," *International Journal for Numerical Methods in Engineering*, vol. 113, no. 11, pp. 1697–1710, 2018.
- [9] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2021.
- [10] K. Bansak, J. Ferwerda, J. Hainmueller et al., "Improving refugee integration through data-driven algorithmic assignment," *Science*, vol. 359, no. 6373, pp. 325–329, 2018.
- [11] A. Clauset, D. B. Larremore, and R. Sinatra, "Data-driven predictions in the science of science," *Science*, vol. 355, no. 6324, pp. 477–480, 2017.
- [12] N. Cheng, F. Lyu, J. Chen et al., "Big data driven vehicular networks," *IEEE Network*, vol. 32, no. 6, pp. 160–167, 2018.
- [13] L. A. Marsch, "Digital health data-driven approaches to understand human behavior," *Neuropsychopharmacology*, vol. 46, no. 1, pp. 191–196, 2021.
- [14] F. Ludbrook, K. F. Michalikova, and Z. Musova, "Business models for sustainable innovation in industry 4.0: smart manufacturing processes, digitalization of production systems, and data-driven decision making," *Economics*, vol. 7, no. 3, pp. 21–26, 2019.
- [15] Q. Jiang, X. Yan, and B. Huang, "Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes," *Industrial & Engineering Chemistry Research*, vol. 58, no. 29, pp. 12899–12912, 2019.
- [16] G. Rong, J. Yang, L. Cheng, J. Tan, J. Peng, and C. Zhou, "A Forchheimer equation-based flow model for fluid flow through rock fracture during shear," *Rock Mechanics and Rock Engineering*, vol. 51, no. 9, pp. 2777–2790, 2018.
- [17] J. Cai, T. Jin, J. Kou, S. Zou, J. Xiao, and Q. Meng, "Lucas-washburn equation-based modeling of capillary-driven flow in porous systems," *Langmuir*, vol. 37, no. 5, pp. 1623–1636, 2021.
- [18] W. A. Mingbo, G. U. Yaliang, F. A. Mingjun, and Z. Songyang, "Dynamics simulation and laws of drilling fluid loss in fractured formations," *Acta Petrolei Sinica*, vol. 38, no. 5, pp. 597–606, 2017.
- [19] L. Sun and J. X. Wang, "Physics-constrained Bayesian neural network for fluid flow reconstruction with sparse and noisy data," *Theoretical and Applied Mechanics Letters*, vol. 10, no. 3, pp. 161–169, 2020.
- [20] S. Nomura, Y. Yamamoto, and H. Sakaguchi, "Modified expression of Kozeny-Carman equation based on semilog-sigmoid function," *Soils and Foundations*, vol. 58, no. 6, pp. 1350–1357, 2018.
- [21] C. Netzer, L. Seidel, M. Pasternak et al., "Three-dimensional computational fluid dynamics engine knock prediction and evaluation based on detailed chemistry and detonation theory," *International Journal of Engine Research*, vol. 19, no. 1, pp. 33–44, 2018.