IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 May 2024, accepted 6 June 2024, date of publication 17 June 2024, date of current version 27 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3415752

RESEARCH ARTICLE

3D Mesh Animation Optimization Algorithm Based on Integer Programming and Nonlinear Constraints

JING JIANG AND XIAOJUN WANG
College of Arts, Anhui Xinhua University, Hefei 230088, China

Corresponding author: Xiaojun Wang (wangxiaojun1299@163.com)

This work was supported in part by the Anhui Traditional Professional Transformation and Upgrading Project "Animation Professional Transformation and Upgrading Project" under Project 2023zygzts104; in part by the Anhui University Philosophy and Social Science Research Project, "Research on the Construction and Application of Anhui Literature and Tourism IP in the Field of Metacosmos" under Project 2023AH051783; in part by the Anhui Province Humanities and Social Sciences Key Research Project, "Research on the Design of Traditional Village Guide System in Huizhou Under the Perspective of Rural Revitalization" under Project SK2021A0788; and in part by the Anhui Xinhua University School-Level Key Teaching Team Project "Visual Communication Design Professional Teaching Team" under Project 2020jxtdx04.

ABSTRACT With the rapid development of digital media and entertainment industry, the importance of 3D mesh animation has become increasingly prominent. However, the traditional production methods are faced with problems such as high computational complexity and unnatural effects when dealing with complex motion. To improve the production efficiency and quality of 3D mesh animation, this study innovatively integrates integer programming, nonlinear constraint optimization, and machine learning algorithms to construct a new 3D mesh animation optimization algorithm. Comparative analysis of the improved machine learning algorithm shows that the mean absolute error of the algorithm is 0.00048 and the fit degree is 98.8%, which is better than the comparison algorithm. Then, the performance of the proposed 3D mesh animation optimization algorithm is analyzed. The results show that the rendering speed and average frame rate of the proposed algorithm are 29.5 FPS and 28.7 FPS, respectively, which is superior to the comparison algorithm. The algorithm can effectively improve the efficiency and quality of animation production, and inject new vitality into the development of digital media and entertainment industry. This study not only provides new ideas and methods for optimizing 3D mesh animation, but also provides useful references for research and applications in related fields.

INDEX TERMS 3D mesh animation, integer programming, nonlinear constraints, machine learning, optimization algorithms.

I. INTRODUCTION

In today's digital media and entertainment industry, 3D mesh animation has become an indispensable part [1]. From character animation in movies and games to applications in virtual reality and augmented reality, 3D mesh animation plays a crucial role [2], [3]. However, with the advancement of technology and the increasing expectations of audiences for visual effects, traditional 3D mesh animation production methods are no longer able to meet certain needs in certain aspects [4]. Therefore, finding a 3D mesh animation optimization algorithm with better overall performance has become an important research direction at present. Traditional 3D mesh animation often faces problems such as high computational complexity and unnatural effects when dealing with complex movements [5]. As a mathematical optimization method, integer programming (IP) can find the optimal integer solution while satisfying a series of constraint conditions [6]. Nonlinear constraints can more accurately describe the motion laws of objects in the real world [7]. Combining IP with nonlinear constraints is expected to provide new ideas and methods for optimizing 3D mesh animations. In actual 3D mesh animation production, animators often need to manually adjust the vertex positions, normal directions, and other

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

attributes of the mesh on the grounds of factors such as the object's motion trajectory and shape changes [8]. The above process is not only cumbersome, but also difficult to ensure the smoothness and realism of the animation. IP and nonlinear constraints can automatically or semi-automatically complete these adjustment tasks, greatly improving the efficiency and quality of animation production. In addition, with the continuous development of technologies such as deep learning and computer vision, optimization algorithms for 3D mesh animation can also be combined with these technologies to achieve more intelligent and automated animation production [9], [10]. Therefore, this study innovatively integrates IP and nonlinear constraint optimization into 3D mesh animation optimization algorithms, and combines them with machine learning algorithms to obtain a new 3D mesh animation optimization algorithm. It is expected that this algorithm can improve the production efficiency and quality of 3D mesh animation, and inject new vitality into the development of digital media and entertainment industry. The contributions of this research are as follows.

- Firstly, IP is combined with nonlinear constraints for the first time to optimize 3D mesh animation. This unique combination of algorithms not only reduces the amount of computation, but also makes the animation effect more natural and realistic.
- Secondly, the research greatly simplifies the work flow of animators by automatically adjusting the attributes of vertex position and normal direction of the mesh, thus significantly improving the production efficiency of 3D mesh animation.
- Thirdly, the combination of IP and nonlinear constraints can achieve finer motion control and more natural animation effects, thus improving the overall quality of 3D mesh animation.
- Fourthly, the research also lays a foundation for the combination of 3D mesh animation optimization algorithms with advanced technologies such as deep learning and computer vision, and promotes the development of animation production technology to a more intelligent and automated direction.

This paper is mainly divided into four parts. The first part is a review of relevant research on 3D mesh animation, mathematical methods, and machine learning technology. The second part explains the design of a 3D mesh animation optimization algorithm on the grounds of mathematical optimization methods and machine learning algorithms. The third part is for verifying the performance of the improved machine learning algorithm and the proposed 3D mesh animation optimization algorithm. The last part is a summary of the entire content and an outlook for the future.

II. RELATED WORKS

With the rapid development of computer graphics, 3D mesh animation has become a research hotspot. To improve the realism and smoothness of animation, Zheng et al. presented a 3D mesh animation optimization algorithm on the grounds of geometric deformation. This algorithm achieved more realistic animation effects by accurately calculating the displacement and deformation of mesh vertices. In practical applications, this algorithm was successfully utilized in multiple fields such as gaming, film and television, and achieved significant results [11]. Liu et al. proposed an optimization algorithm based on data compression to address performance bottlenecks in 3D mesh animation. This algorithm significantly improved the rendering and response speed of animations by reducing the storage and transmission of mesh data. The experiment showcased that the algorithm reduced the consumption of computing resources while maintaining animation quality, providing a new approach for real-time rendering of 3D mesh animations [12]. Marek et al. proposed an optimization algorithm based on detail enhancement to address the issue of detail loss in 3D mesh animations. This algorithm incorporated detailed information into the animation by extracting and analyzing the microscopic features of the mesh surface, thereby enhancing the realism and visual impact of the animation. This experiment demonstrated that the algorithm had significant advantages in processing complex scenes and fine textures, providing strong support for high-quality rendering of 3D mesh animations [13]. To further optimize the performance of 3D mesh animation, Luo et al. proposed an algorithm on the grounds of dynamic simplification. This algorithm dynamically adjusted the complexity and level of detail of the mesh based on the motion state of objects in the animation and the visual focus of the observer. The practice showcased that this algorithm possessed broad application prospects in interactive applications [14].

In recent years, with the increasing complexity of combinatorial optimization problems, IP methods have received widespread attention. To improve solving efficiency, Wu proposed an IP optimization algorithm on the grounds of a novel heuristic search. This algorithm combined ideas such as genetic algorithm and simulated annealing to find the optimal solution in the integer solution space through intelligent search strategies. The results showed that the algorithm could obtain high-quality solutions in a short period of time when solving large-scale IP problems, providing strong support for practical applications in related fields [15]. To improve the generality and scalability of the algorithm, Xia et al. proposed a mixed IP algorithm based on branch and bound and cutting plane methods. This algorithm achieved efficient solution to complex IP problems by introducing effective inequalities and reinforcement learning techniques. The experiment showcased that the algorithm performed well in solving various types of IP problems, providing new ideas for research and application in relevant fields [16]. Researchers are constantly exploring new solutions for nonlinear constrained optimization problems. Aiming at enhancing the convergence speed and global search ability of the algorithm, Zhao et al. proposed a hybrid algorithm on the grounds of particle swarm optimization and differential evolution. This algorithm effectively solved complex nonlinear constraint problems by introducing a non-linear constraint processing mechanism. The practical applications showcased that this algorithm has broad application prospects in solving engineering optimization problems [17]. Recently, as the boost of big data and artificial intelligence technology, machine learning technology has also been applied in 3D mesh animation. Mondal et al. presented a machine learning-based optimization algorithm for 3D mesh animation to improve its realism and fluency. This algorithm utilized machine learning algorithms to intelligently analyze and process mesh models, achieving more realistic animation effects through automatic learning and optimization of mesh structures. The experiment showcased that the algorithm enhances rendering speed and resource utilization while maintaining animation quality. This offers new ideas and methods for the production and application of 3D mesh animation [18].

According to the above research, the application of 3D mesh animation algorithms has shown diversity and innovation, and IP and nonlinear constraint strategies have shown wide applicability in optimization problems. In this context, this study adopted a 3D mesh animation algorithm that integrates mathematical optimization algorithms and machine learning algorithms. By integrating mathematical rigor with the intelligence of machine learning, this algorithm aims to further enhance the realism, smoothness, and rendering efficiency of animation, injecting new vitality into the field of 3D mesh animation.

III. 3D MESH ANIMATION OPTIMIZATION ALGORITHM COMBINING MATHEMATICAL OPTIMIZATION METHODS AND MACHINE LEARNING ALGORITHMS

As the core technology of modern digital media, the optimization algorithm of 3D mesh animation is crucial for improving the quality and efficiency of animation [19]. This part delves into the 3D mesh animation optimization method that integrates IP, nonlinear constraints, and machine learning algorithms. By constructing precise mathematical models and introducing intelligent algorithms, more efficient and realistic 3D animation effects can be achieved, injecting new vitality into the development of the industry.

A. 3D MESH ANIMATION OPTIMIZATION ALGORITHM COMBINING INTEGER PROGRAMMING AND NONLINEAR CONSTRAINTS

The 3D mesh animation algorithm is one of the key technologies in modern computer graphics, which can present realistic and dynamic 3D objects on the screen [20]. The core of 3D mesh animation algorithm lies in the deformation and motion control of the mesh model. A simple 3D mesh model is usually composed of thousands to millions of vertices, which are connected by edges and faces to form a complete object surface [21]. The goal of 3D mesh animation algorithms is to update the positions of these vertices at each frame to simulate the motion, deformation, and interaction of objects. The traditional 3D mesh animation algorithm relies on key-frame animation and interpolation techniques, and the algorithm process is shown in Figure 1.

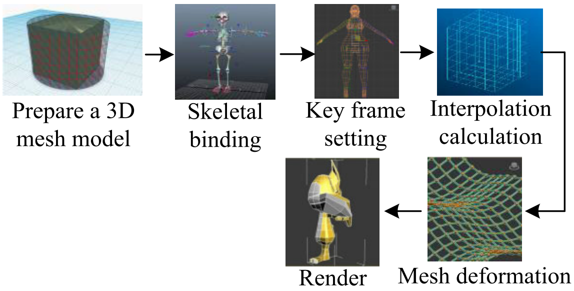


FIGURE 1. Traditional 3D mesh animation algorithm flow.

3D animation production involves multiple steps. Firstly, it prepares a 3D mesh model consisting of vertices, edges, and faces. To animation character movements, it binds the skeletal system to the model. The animator first defines the position and pose of the object in key-frames, and then the algorithm inserts intermediate frames between these key-frames to make the object's motion appear smoother and more natural. In 3D mesh animation, the position of each vertex changes over time. It assumes that the position of vertex i at time t is $p_i(t)$, where linear interpolation is used to smooth the transition model state. Linear interpolation is a method of estimating a new value between two known values. The linear interpolation expression is shown in equation (1).

$$p_i(t) = (1-\alpha)p_i(t_0) + \alpha p_i(t_1) \tag{1}$$

In equation (1), α is the interpolation factor that varies between 0 and 1, and t_0 and t_1 are the time points of the key-frame. The rotation of an object can be represented by a rotation matrix. For example, the rotation matrix of the rotation angle θ around the x-axis is shown in equation (2).

$$R_z(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{2}$$

The new position of the vertex can be represented by equation (3).

$$p_i' = R_z(\theta)p_i \tag{3}$$

After interpolation, the algorithm will cause the mesh to deform, updating vertex positions to reflect motion and deformation, ensuring continuous smoothness. Finally, it renders the deformed model to the screen, including lighting, texture, and depth testing, to present the final animation effect. 3D mesh animation, as a core technology in the digital media and entertainment industry, the complexity of its production process and high requirements for visual effects make algorithm optimization crucial. As a mathematical optimization method, IP plays an important role in the optimization of 3D mesh animations [22]. The principle of IP is shown in Figure 2 [23].

As shown in Figure 2, the principles of IP mainly involve four aspects: constraints and integer requirements, feasible regions and optimal solutions, solution methods, and practical

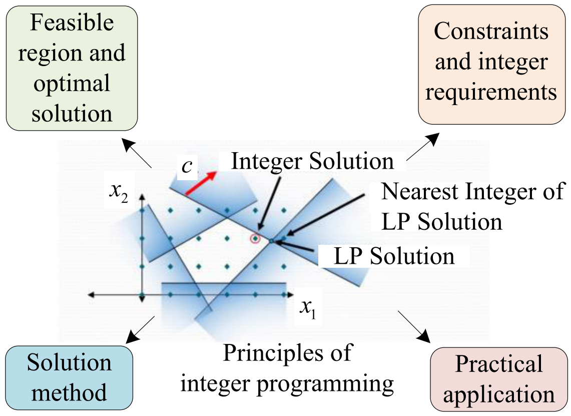

Feasible region and optimal solution
Constraints and integer requirements
c
Integer Solution
x_2
Nearest Integer of LP Solution
LP Solution
x_1
Solution method
Principles of integer programming
Practical application

FIGURE 2. Specific schematic diagram of integer programming.

applications. IP requires some or all variables to be integers, which increases the difficulty of the problem. The feasible domain is composed of integer points that satisfy the conditions, while the optimal solution reaches the extremum of the objective function. The solution method is more complex than linear programming, commonly used methods such as branch and bound method, cutting plane method, etc. In practical applications, such as production planning, logistics distribution, and other issues, IP is very effective. In animation production, many parameters such as the number of vertices and faces need to be rounded. Through IP optimization, more accurate and efficient animation effects can be achieved [24]. At each frame of a 3D animation, the position of vertices needs to be adjusted according to the motion and deformation of the object. IP can be used to optimize this process and find the vertex positions that achieve the best animation effect. Its expression is shown in equation (4).

$$\begin{cases} \min\limits_{P} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{m} w_j \left\| \left(p_i - v_j\right) - R_j \left(p_i - c_j\right) \right\|^2 \\ st. \quad P \in Z^{3\times n} \end{cases} \tag{4}$$

In equation (4), P serves as the position matrix of the vertex, p_i serves as the position of the i-th vertex, v_j and c_j are the linear and angular velocities of the j-th rigid body, R_j is the rotation matrix, and w_{ij} is the weight coefficient, which is utilized for measuring the degree of correlation between the vertex and the rigid body. Equation (4) aims to minimize the difference between vertex position and rigid body motion while ensuring the integer value of vertex position. In 3D mesh animation, the quantity of faces is also a key parameter that needs optimizing. Too many faces can increase computational complexity, while too few faces can affect the visual effect of the animation. IP can help find the optimal number of faces, as expressed in equation (5).

$$\begin{cases} \min\limits_{nf} nf, nf \in N \\ s.t. \quad Volume(Mesh) \geq V_{\min} \end{cases} \tag{5}$$

In equation (5), nf is the number of faces, $Volume(Mesh)$ is the volume of the mesh, and $V_{\min}$ is the minimum required volume. Equation (5) aims to minimize the number of faces while ensuring that the volume of the mesh is not less than the set minimum value. In animation production, the division of time frames is also a parameter that needs to be optimized. Reasonable time frame partitioning can make animations smoother while reducing computational complexity. IP can be used to optimize the division of time frames, and its expression is shown in equation (6).

$$\begin{cases} \min\limits_{\Delta t} \sum\limits_{k=1}^{T} \| p(k\Delta t) - p(k-1)\Delta t \|^2 \\ s.t. \Delta t \in N \end{cases} \tag{6}$$

In equation (6), Δt is the interval between time frames, T serves as the total time, and $p(k\Delta t)$ serves as the position of the object at time t. Equation (6) aims to minimize the difference in object positions between adjacent time frames while ensuring that the time frame interval is an integer value. By optimizing key parameters through IP, more precise, efficient, and realistic animation effects can be achieved. In the production of 3D mesh animation, although IP provides powerful tools for optimization, many elements and effects of animation exhibit nonlinear characteristics in actual scenes [25]. For simulating the deformation of soft objects, such as fabrics or water flow, a nonlinear deformation model can be used, which combines internal tension and external forces to more realistically simulate the dynamic behavior of the object [26]. The expression of the nonlinear deformation model is shown in equation (7).

$$M\ddot{x} + C\dot{x} + Kx = F_{ezt} + F_{int}(x, \dot{x}) \tag{7}$$

In equation (7), M serves as the mass matrix, C serves as the damping matrix, K serves as the stiffness matrix, x serves as the displacement vector of the object, $\dot{x}$ and $\ddot{x}$ are the velocity and acceleration vectors, respectively. F_{ezt} represents external force such as gravity and wind, while $F_{int}(x, \dot{x})$ represents internal tension, which is a nonlinear function of displacement and velocity. When simulating lighting effects, complex lighting models can be used to achieve more realistic lighting effects. The Phong lighting model combines ambient light, diffuse reflection, and specular reflection to simulate the surface lighting of objects. Its nonlinear expression is shown in equation (8).

$$I = I_a k_a + I_l k_d (n \cdot l) + I_l k_s (r \cdot v)^{\alpha} \tag{8}$$

In equation (8), I serves as the final illumination intensity, I_a serves as the ambient light intensity, I_l is the ambient light reflection coefficient, I_l is the point light source intensity, and k_d serves as the diffuse reflection coefficient. n serves as the surface normal vector, l serves as the unit vector from the point light source to the surface, k_s is the specular reflection coefficient, r serves as the specular reflection direction, v serves as the observation direction, and α serves as the specular reflection highlight index. In 3D mesh animation, texture mapping is the process of mapping 2D texture images to the surface of 3D objects. To achieve non-linear texture mapping effects, the expression shown in equation (9) can be used.

$$T = M_{tex} \cdot (U \cdot x + t) \tag{9}$$

In equation (9), T is the texture coordinate, M_{tex} is the texture transformation matrix, which can be any non-linear transformation, U is the linear transformation matrix between the texture coordinate and the object coordinate, x is the object coordinate, and t is the texture offset. By adjusting the texture transformation matrix, non-linear distortion, scaling, and rotation effects of textures can be achieved. Therefore, introducing nonlinear constraints is crucial for further improving the realism and precision of animation. The specific process of the 3D mesh animation optimization algorithm proposed in this study, which integrates IP and nonlinear constraints, is shown in Figure 3.

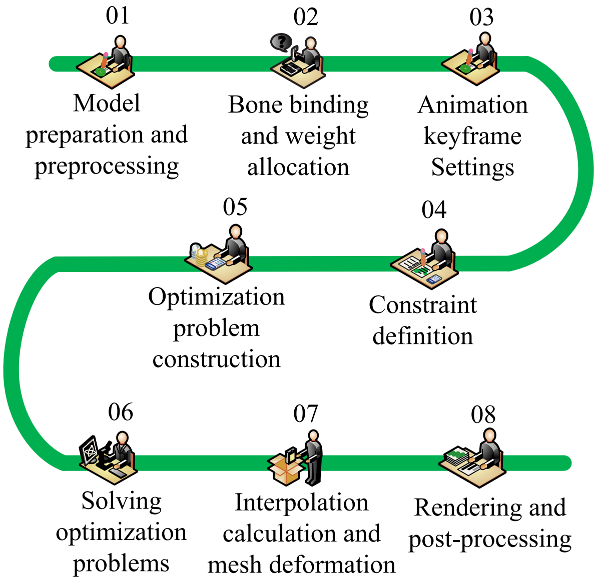

01
Model preparation and preprocessing
02
Bone binding and weight allocation
03
Animation keyframe Settings
05
Optimization problem construction
04
Constraint definition
06
Solving optimization problems
07
Interpolation calculation and mesh deformation
08
Rendering and post-processing

FIGURE 3. Flow of 3D mesh animation optimization algorithm combining integer programming and nonlinear constraints.

Figure 3 shows that the proposed 3D mesh optimization algorithm is mainly divided into eight steps. Firstly, it is the model preparation and pre-processing stage. In this stage, a 3D mesh model composed of vertices, edges, and faces is imported or generated, and pre-processing operations such as cleaning invalid data and optimizing the mesh are carried out. If the animation involves character motion, a skeletal system needs to be created and bound to the model before assigning weights. Next, it sets animation key-frames and defines and sets the model state of key-frames on the timeline. To meet the needs of animation, IP and nonlinear constraint conditions are defined, and the animation deformation problem is transformed into a mathematical optimization problem, with corresponding constraints introduced. Afterwards, it uses optimization algorithms to solve the problem. After solving the problem, it uses interpolation techniques to calculate the intermediate frame state and deforms the mesh. Finally, it is rendered and post-processed to obtain animation frames after completion of rendering and post-processing. To improve the effect or handle more complex scenes, the above steps can be iterated in a loop.

B. DESIGN OF 3D MESH ANIMATION OPTIMIZATION ALGORITHM COMBINED WITH IMPROVED GREY WOLF OPTIMIZATION (GWO) ALGORITHM

In the production process of 3D mesh animation, prediction algorithms play a crucial role. Prediction algorithms can estimate the position, pose, and deformation of objects in future frames on the grounds of previous frames, providing smooth animation effects [27]. The BP algorithm, as a common prediction algorithm, has strong nonlinear mapping and self-learning capabilities, can adaptively extract data rules and remember them, and is effectively applied to complex prediction problems [28]. However, there are some shortcomings in the BP algorithm, such as slow convergence speed and inconsistent network structure. Therefore, this study aims to improve the BP algorithm and apply it to the design of 3D mesh animation optimization algorithms. To improve the BP algorithm, the GWO algorithm was selected for optimization in this study. The GWO algorithm originated from the collective behavior of gray wolves. There are four levels of wolves in grey wolf (GW) society: α, β, χ, and δ. The GWO algorithm optimizes search by simulating social behavior in wolf packs [29]. Firstly, it randomly initializes a group of gray wolves and calculates the fitness value of each wolf. The expression for initializing the GW pack is shown in equation (10).

$$X(0) = [x_1(0), \ x_2(0), \ldots, x_n(0)], i = 1, 2, \ldots, D \tag{10}$$

It updates the wolf's position on the grounds of the distance between each wolf and wolves α, β, and χ. Finally, after multiple iterations, the wolf pack will gradually converge to the optimal solution. The GWO-BP algorithm uses the global search performance of the GW algorithm for finding the initial parameter range in the BP algorithm, and then assigns the optimal weights and thresholds within this range to the BP algorithm. This approach can effectively solve the problem of easily getting stuck in local minima due to inappropriate initial weights and thresholds. The relevant process of GWO-BP algorithm is showcased in Figure 4.

The solution steps for GWO-BP are as follows. First, it defines the topology of the BP network and uses the network's weights and thresholds as the initialization parameters for GWO. Then, it initializes the GWO algorithm population, treating the individual GW as a component composed of weights and thresholds between the layers of the BP neural network. Next, it determines the fitness value function, takes the initial parameters of the GW individual as the initial value of the BP algorithm, and trains the neural network to obtain the output value and the desired average error as the fitness value. The fitness value of the first generation GW is shown

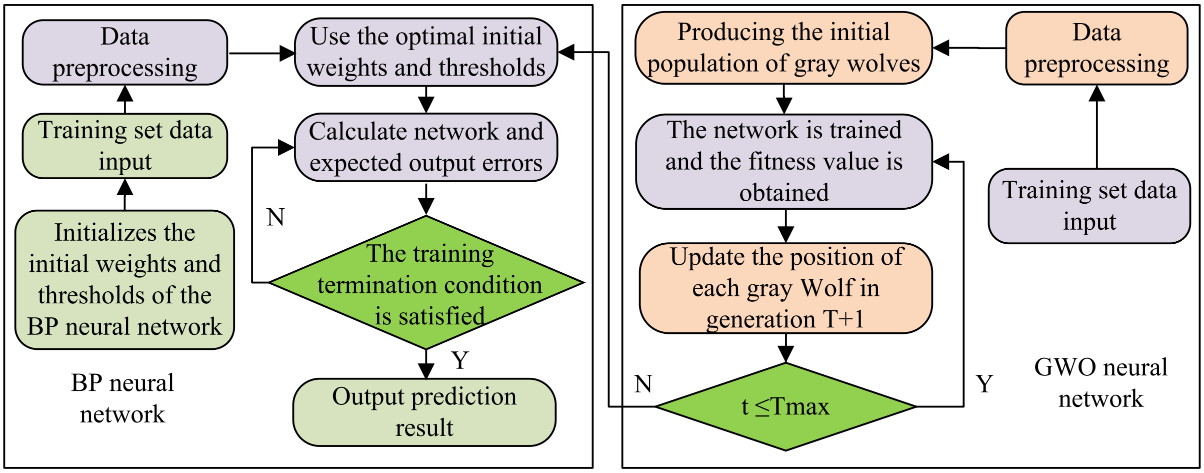

Data preprocessing
Use the optimal initial weights and thresholds
Training set data input
Calculate network and expected output errors
Initializes the initial weights and thresholds of the BP neural network
N
The training termination condition is satisfied
Y
BP neural network
Output prediction result
Producing the initial population of gray wolves
Data preprocessing
The network is trained and the fitness value is obtained
Training set data input
Update the position of each gray Wolf in generation T+1
N
t ≤Tmax
Y
GWO neural network

FIGURE 4. Flow of GWO-BP prediction model.

in equation (11).

$$F = \frac{1}{A}\sum_{s=1}^{A}\sqrt{\sum_{k=1}^{A}\left(y_k^s - o_k^s\right)^2} \tag{11}$$

Then it selects the highest 3 wolves in the wolf pack as α, β, and δ, and updates the parameters of $\vec{r}_1$, $\vec{r}_2$, and $\vec{a}$ in GWO. Finally, it checks whether the number of iterations set by the algorithm has been reached. If it is not completed, it will update the parameters in GWO again. If it is completed, it will assign the optimal initial parameters to the BP neural network.

The basic idea of the principal component analysis (PCA) algorithm is for projecting the original data onto a new feature space, which is composed of a set of orthogonal principal components [30]. Each principal component represents the maximization of variance of the original data in a specific direction, which can retain the most information. In the 3D mesh animation optimization scene, the traditional PCA method is superior to other deep learning algorithms due to its simplicity, strong interpretability and fast computational efficiency. Therefore, this study constructed a GW optimized BP prediction model on the grounds of PCA. By integrating PCA algorithm and GWO algorithm, the feature dimension can be reduced and the training effect of the neural network can be improved simultaneously. The PCA algorithm can reduce redundant information in data and improve its expressive power. The GWO algorithm can optimize the weights and biases of neural networks, improve the accuracy of the model, and the PCA-GWO-BP fusion algorithm has certain advantages in solving high-dimensional data training problems, as shown in Figure 5.

As shown in Figure 5, the specific steps for solving the GWO-BP algorithm that integrates PCA algorithm are as follows. First, it determines the position and speed of the initialized population's gray wolves, and calculates the fitness of each GW. It ranks the strengths and weaknesses of gray wolves, and updates their position and speed. On the grounds of the updated GW position, it uses PCA algorithm to reduce the dimensionality of the position. On the grounds of the dimensionality reduction data, the BP algorithm is used for network training and the fitness of the network is calculated. Finally, the position and speed of the GW pack are updated through fitness values. It repeatedly initializes the position and speed of the GW population until the stopping condition is reached. In the PCA-GWO-BP algorithm, the GW position update method is shown in equation (12).

$$x(i, t+1) = x(i, t) \ + A * D \tag{12}$$

In equation (12), A is the control parameter, D is the position difference of the GW, and the expression for updating the GW speed is shown in equation (13).

$$v(i, t+1) = r * v(i, t) + C * P - x(i, t) \tag{13}$$

In equation (13), C represents the control parameter, r represents a random number, and P serves as the position of the current optimal solution. The selection of control parameters A and C has a significant influence on the convergence speed and quality. The expression for calculating the fitness of GW is shown in equation (14).

$$fitness(i) = \frac{1}{(1 \ + error(i))} \tag{14}$$

In equation (14), $error(i)$ is the training error of the BP algorithm. The PCA-GWO-BP algorithm combines the PCA algorithm with the GWO-BP algorithm, utilizing the dimensionality reduction ability of the PCA algorithm and the optimization ability of the GWO-BP algorithm, for quickly finding the optimal solution to the optimization problem. Meanwhile, by flexibly adjusting parameters, better performance can be achieved on different issues. The specific process of the new 3D mesh optimization algorithm obtained by integrating PCA-GWO-BP algorithm into IP and nonlinear constraint optimization of the 3D mesh animation algorithm is shown in Figure 6.

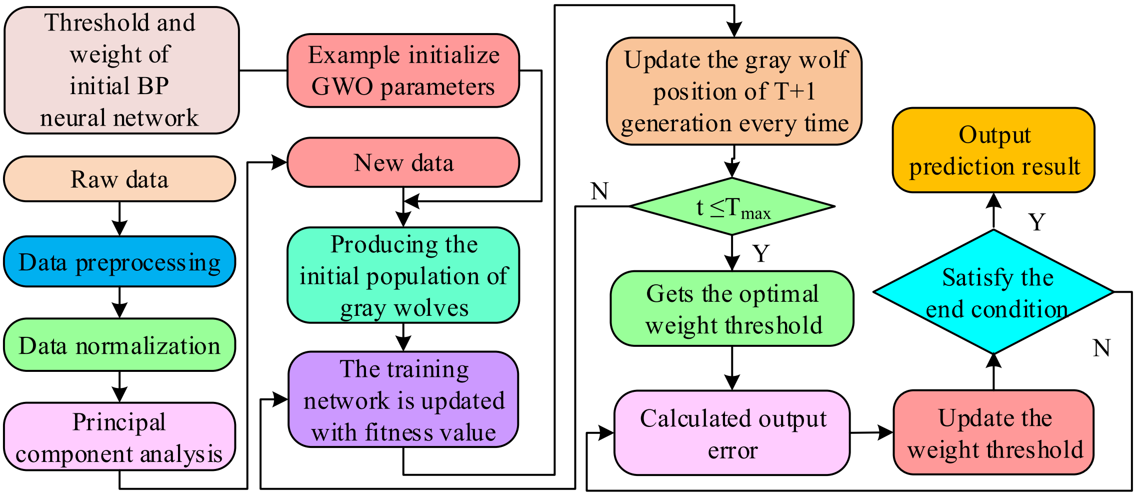

Threshold and weight of initial BP neural network
Example initialize GWO parameters
Update the gray wolf position of T+1 generation every time
Output prediction result
Raw data
New data
N
$t \leq T_{max}$
Y
Data preprocessing
Producing the initial population of gray wolves
Gets the optimal weight threshold
Y
Satisfy the end condition
Data normalization
N
The training network is updated with fitness value
Principal component analysis
Calculated output error
Update the weight threshold

FIGURE 5. PCA-GWO-BP model flow.

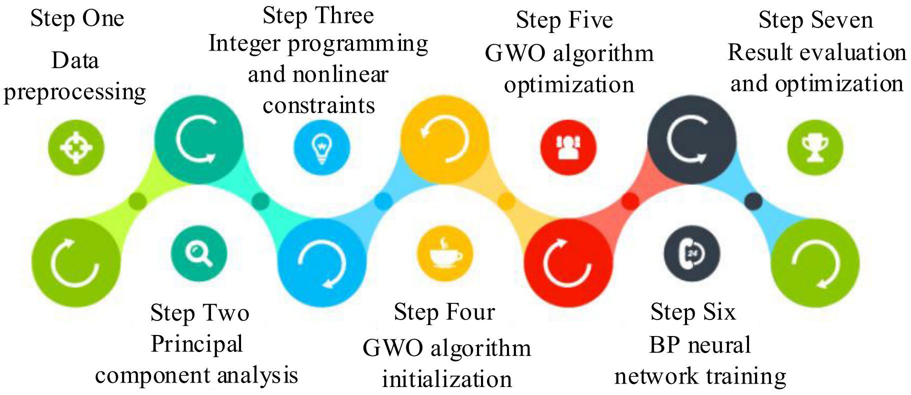

Step One
Data preprocessing
Step Three
Integer programming and nonlinear constraints
Step Five
GWO algorithm optimization
Step Seven
Result evaluation and optimization
Step Two
Principal component analysis
Step Four
GWO algorithm initialization
Step Six
BP neural network training

FIGURE 6. 3D mesh optimization algorithm flow integrated with PCA-GGO-BP algorithm.

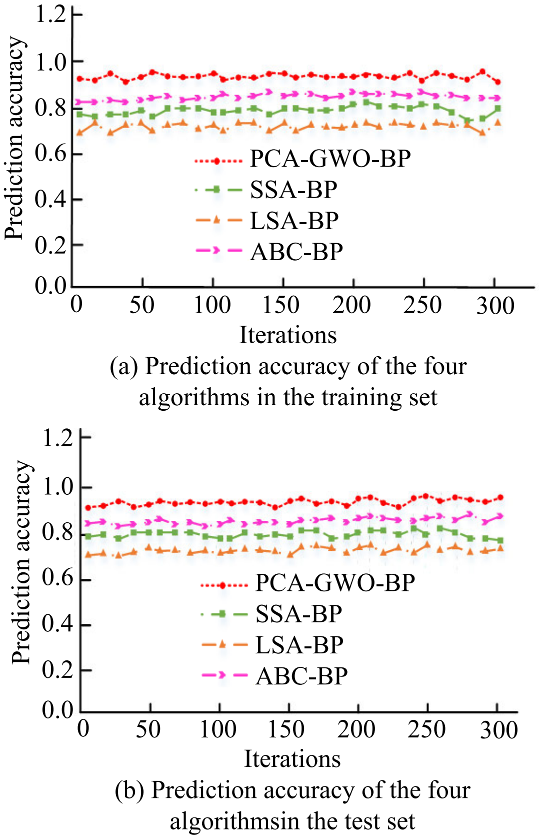
TABLE 1. Basic environment of improved prediction algorithm comparison experiment.

Environment type	Environmental composition	Types and specifications
Hardware environment	processor	Intel Core i7-8700K
	Graphics card	Nvidia GeForce GTX 1080 Ti
	Operating system	Windows 10 Pro(64-bit)
	Running memory	16GB DDR4 RAM
	Storage memory	512GB SSD
Software environment	Programming language	Python
	Machine learning library	Tensorflow
	Numerical calculation	SciPy
	PCA implementation	Scikit-learn (Python library)
	Implementation of GGO-BP algorithm	Custom Python

As shown in Figure 6, the improved 3D mesh optimization algorithm consists of seven steps. Firstly, data pre-processing is performed for ensuring the quality of the 3D mesh animation data. Next, the PCA algorithm is applied for dimensionality reduction, extracting key features, and reducing computational burden. Then, according to the animation requirements, IP and nonlinear constraints are set to delineate the actual solution space for the optimization process. On this basis, the study initializes the GWO algorithm, sets parameters, and prepares to search for the optimal solution. It utilizes the GWO algorithm to globally optimize under constraint conditions and find the optimal 3D mesh animation parameters. Subsequently, these parameters are utilized for training the BP neural network and simulating the complex relationships of animations. Finally, it evaluates the trained neural network model, including prediction accuracy, performance, and visual effects. It further optimizes the algorithm on the grounds of the evaluation results. This algorithm combines the dimensionality reduction ability of PCA, the global optimization ability of GWO, the learning ability of BP neural network, as well as the practicality of IP and nonlinear constraints, and it is expected to achieve significant results in 3D mesh animation optimization.

IV. COMPARATIVE ANALYSIS OF IMPROVED PREDICTION ALGORITHMS AND 3D MESH ANIMATION OPTIMIZATION ALGORITHMS

This study compared and analyzed the performance of the improved prediction algorithm in this part to verify its superiority. Moreover, the performance of the proposed 3D



(a) Prediction accuracy of the four algorithms in the training set

(b) Prediction accuracy of the four algorithmsin the test set

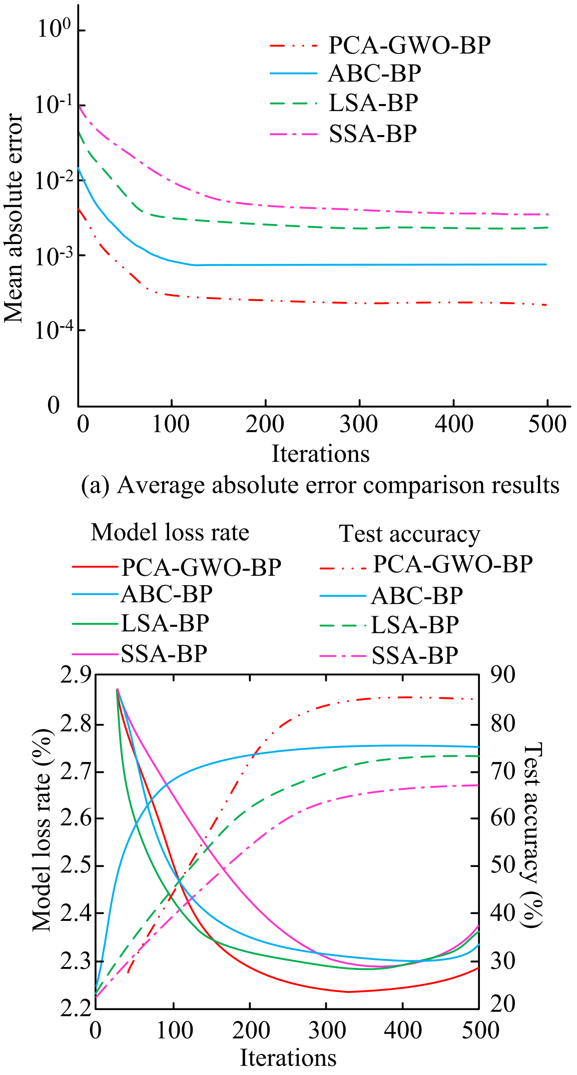
FIGURE 7. Prediction accuracy of the three algorithms in two data sets.

mesh animation algorithm was verified through comparative experiments. This reflects the practicality of proposing optimization algorithms for 3D mesh animation in research.

A. PERFORMANCE COMPARISON AND ANALYSIS OF PCA-GWO-BP ALGORITHM

Aiming at verifying the performance of the proposed PCA-GWO-BP algorithm, this study was conducted in MATLAB, and the specific experimental environment is showcased in Table 1.

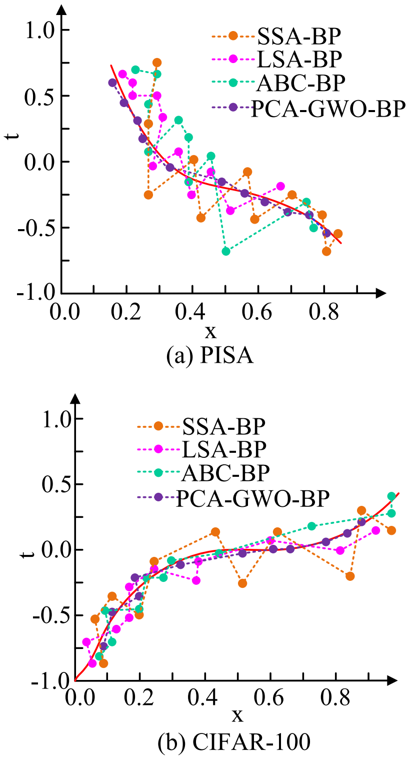
In this setting, the PCA-GWO-BP algorithm was implemented using Python programming language and machine learning operations were performed using TensorFlow. The hardware environment included high-performance processors, sufficient memory, and powerful graphics cards that support efficient computing. In addition, the study selected the CIFAR-100 dataset as the test set, which contains 100 images of different categories, with 600 images in each category. The goal was to classify various objects such as fish, flowers, fruits and vegetables on the grounds of the content of these images. Aiming at better analyzing the performance of PCA-GWO-BP algorithm, this study tested its predictive accuracy, precision, recall, ROC curve, mean absolute error, and other indicators on the CIFAR-100 dataset, along with SSA-BP algorithm [31], LSA-BP algorithm [32], and ABC-BP algorithm [33]. Among them, Sparrow Search Algorithm (SSA) is a new swarm intelligent optimization algorithm. Inspired by the search behavior of sparrows, the algorithm realized the purpose of optimizing the objective function by simulating the behavior of sparrows in the process of searching for food and avoiding natural enemies [34]. Lightning Search Algorithm (LSA) is an optimization algorithm inspired by the lightning phenomenon in natural weather, which simulates the fast propagation characteristics of lightning and therefore has a fast convergence speed [35]. Artificial Bee Colony Algorithm (ABC) is an optimization algorithm that simulates bees' foraging behavior. The core idea of this algorithm was to find the optimal solution by simulating bees' foraging behavior. In this algorithm, bees selected the target location according to pheromone concentration when searching for food and explored within a certain range of the surrounding area [36]. Figure 7 shows the comparison of prediction accuracy of PCA-GWO-BP, SSA-BP, LSA-BP and ABC-BP algorithms in the training set and the test set.



(a) Average absolute error comparison results

(b) Results of comparison of loss rate and accuracy

FIGURE 8. Comparison results of average absolute error, loss rate and accuracy of the four algorithms.



(a) PISA

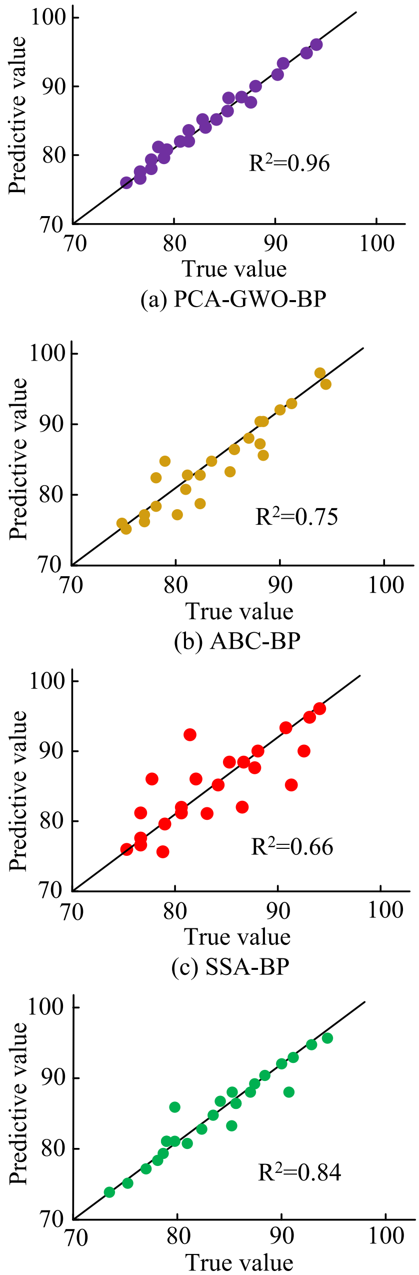
(b) CIFAR-100

FIGURE 9. Comparison of fit of different algorithms.

Figures 7 (a) and (b) show the prediction accuracy curves of the four algorithms in the training and testing sets, respectively. Figure 7 (a) shows that the PCA-GWO-BP algorithm had the highest prediction accuracy among the four algorithms, with an average prediction accuracy of 93.9%, while the LSA-BP algorithm possessed the lowest prediction accuracy, with an average accuracy of 73.6%. By observing Figure 7 (b), the prediction accuracy curve of the PCA-GWO-BP algorithm was also significantly higher than the comparison algorithm. Its average prediction accuracy was 94.3%, significantly better than ABC-BP algorithm's 83.8%, SSA-BP algorithm's 80.2%, and LSA-BP algorithm's 73.8%. The above results indicated that the prediction accuracy of the PCA-GWO-BP algorithm was significantly more excellent than the comparison algorithm. In addition, the average absolute error, model loss rate, and test accuracy results of the four algorithms in the CIFAR-100 dataset are shown in Figure 8.

Figure 8 (a) shows that the minimum average absolute error of PCA-GWO-BP algorithm was 0.00048, which was significantly lower than 0.00098 of ABC-BP algorithm, 0.0066 of LSA-BP algorithm, and 0.0079 of SSA-BP algorithm. Figure 8 (b) shows that the minimum model loss rate and maximum test accuracy of the PCA-GWO-BP algorithm were 2.25% and 84.4%, respectively, which are better than the three comparative algorithms. The above results indicated that from the dimensions of average absolute error, loss rate, and testing accuracy, the PCA-GWO-BP algorithm had better actual performance. On the grounds of the comparison of multiple dimensions mentioned above, it is evident that the PCA-GWO-BP algorithm performed better than the comparison algorithms. Additionally, four algorithms were tested on two datasets, and the fitting results of the four algorithms on both datasets were statistically plotted in Figure 9. The two different datasets in Figure 9 are the CIFAR-100 dataset and the PISA dataset.

Figure 9 (a) shows the comparison of fitting degrees among four algorithms in the PISA dataset. As shown in Figure 9 (a), the PCA-GWO-BP algorithm had a fitting degree of 98.8%, with the best fitting degree, which was 23.4%, 17.5%, and 20.1% higher than the SSA-BP algorithm, LSA-BP



(a) PCA-GWO-BP

(b) ABC-BP

(c) SSA-BP

(d) LSA-BP

FIGURE 10. Prediction accuracy of different algorithms.

TABLE 2. Comparison results of visual quality effects of the four algorithms.

Test data \ algorithm	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
Complex urban landscape	Excellent (details preserved, no obvious distortion)	Good (some details missing)	General (some structure oversimplified)	Good (some details missing)
High-precision character model	Excellent (facial details, clothing texture clear)	Good (loss of facial details)	Medium (facial features simplified)	Good (some clothing texture blurred)
Dynamic flow simulation	Excellent (natural water texture, no fracture)	Good (slightly stiff water texture)	General (water flow simplified, unnatural)	Medium (partial flow detail missing)
Complex mechanical devices	Excellent (clear mechanical structure, smooth movement)	Good (some mechanical structure simplification)	General (some key structure missing)	Good (some movement details are not smooth)
Large-scale buildings	Excellent (building detail, no distortion)	Good (some building details are blurred)	Medium (some buildings oversimplified)	Good (some building details missing)

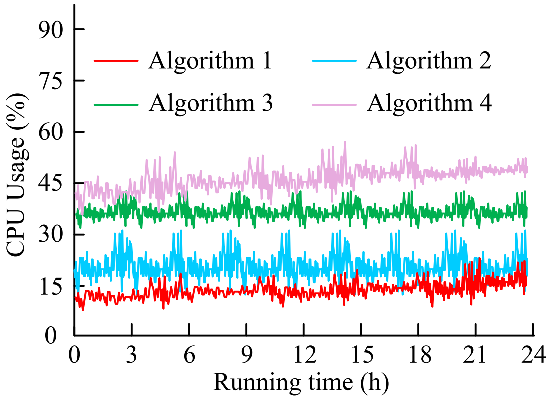


FIGURE 11. Comparison results of memory usage of the four algorithms.

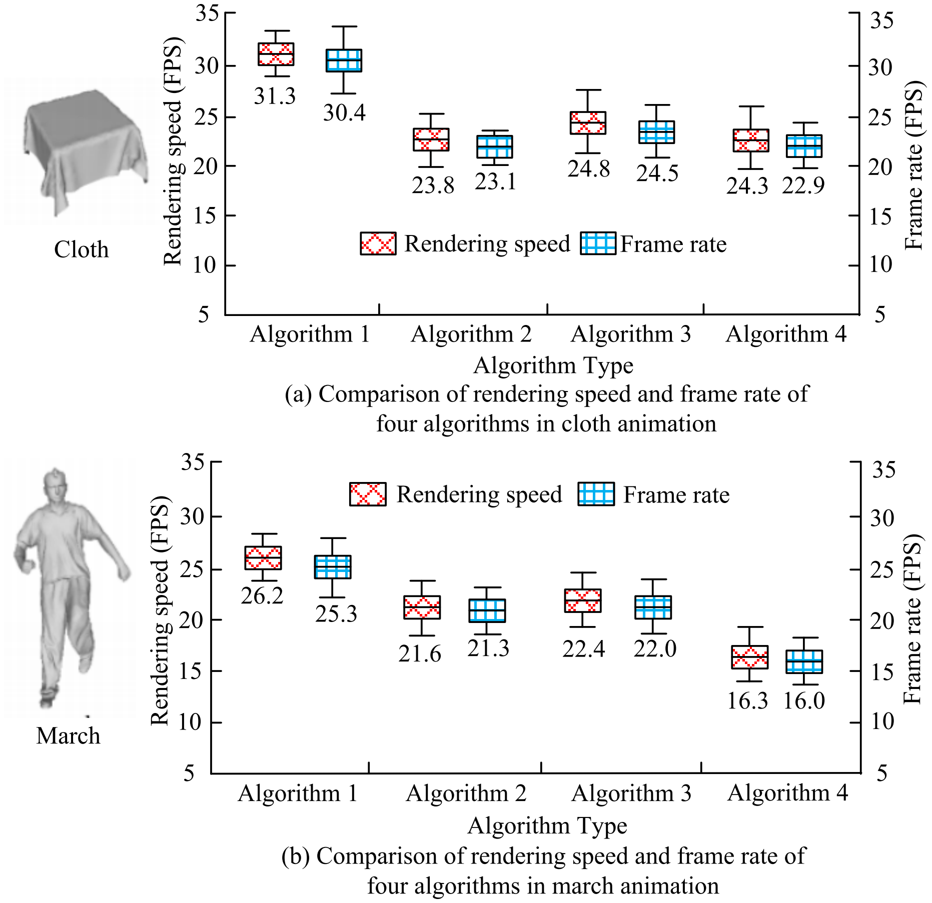
algorithm, and ABC-BP algorithm, respectively. Figure 9 (b) showcases a comparison of the fit of four algorithms in the CIFAR-100 dataset. Figure 9 (b) shows that the PCA-GWO-BP algorithm had the best fit, at 97.2%, which is significantly higher than the comparison algorithm, and the SSA-SVM algorithm showed over-fitting. The above research results showcased that from the perspective of algorithm fit, PCA-GWO-BP algorithm performed better than SSA-BP, LSA-BP, and ABC-BP algorithms. Finally, aiming at further validating the superiority of the proposed algorithm, four algorithms were applied to the PISA dataset for prediction accuracy testing. The test results are shown in Figure 10. R^2(determination coefficient) in Figure 10 is a commonly used regression model evaluation index, which is mainly used to measure the model's fitting degree and prediction accuracy of the observed data [37], [38].

Figure 10 (a) shows the test results of prediction accuracy of PCA-GGO-BP algorithm. Figure 10 (b) shows the test results of the prediction accuracy of ABC-BP algorithm. Figure 10 (c) shows the test results of prediction accuracy of SSA-BP algorithm. Figure 10 (d) shows the test results of the prediction accuracy of LSA-BP algorithm. The PCA-GWO-BP algorithm had the highest R^2 at 0.96, and among the four algorithms, the SSA-BP algorithm had the lowest prediction accuracy at 0.66. By comparing Figure 10 comprehensively, it can be concluded that the R^2 value of PCA-GWO-BP algorithm was 0.30, 0.21, and 0.12 higher than that of SSA-BP, ABC-BP, and LSA-BP, respectively. This result indicated that from the comparison of R^2 values representing prediction accuracy, the PCA-GWO-BP algorithm possessed better prediction performance than the comparison algorithm. On the grounds of the comparison of the above dimensions, it can be found that the overall performance of PCA-GWO-BP algorithm is superior to the comparison algorithm.

B. PERFORMANCE COMPARISON AND EMPIRICAL ANALYSIS OF OPTIMIZATION ALGORITHMS FOR 3D MESH ANIMATION

To conduct comparative experiments on 3D mesh animation optimization algorithms, a specific experimental environment was constructed in this study. This environment adopted a high-performance computing platform, equipped with advanced graphics processing units and sufficient memory resources to ensure the accuracy and efficiency of the experiment. Meanwhile, multiple representative 3D mesh models and animation scenes were selected as test data to comprehensively evaluate the performance of various optimization algorithms. In this experimental environment, the proposed 3D mesh animation optimization algorithm (Algorithm 1) would be compared with the dynamic subdivision optimization algorithm (Algorithm 2), deep learning-based mesh simplification method (Algorithm 3), and adaptive geometric level of detail algorithm (Algorithm 4). The performance of the four algorithms were compared using rendering speed, memory usage, frame rate, and visual quality as comparison indicators. The memory usage results of the four algorithms are showcased in Figure 11.

Figure 11 showcases that the CPU usage of the proposed algorithm fluctuated between 10% and 20% within 24 hours. The CPU usage of other algorithms significantly exceeded the proposed algorithm, and among them, Algorithm 4 had the highest CPU usage, with a fluctuation range of 37% to 48%. In addition, the CPU usage fluctuations of Algorithm 2 and Algorithm 3 ranged from 13% to 30% and 37% to 40%, respectively. The above results indicated that from the perspective of memory usage, the proposed 3D mesh animation optimization algorithm performed better. In addition, four algorithms were applied to different datasets for visual quality effect testing, and the comparison results of the



(a) Comparison of rendering speed and frame rate of four algorithms in cloth animation

(b) Comparison of rendering speed and frame rate of four algorithms in march animation

FIGURE 12. Comparison of rendering speed and frame rate of four algorithms in tablecloth and quickwalk animation.

TABLE 3. Comparison of rendering speed and frame rate of four algorithms in horse and aeroplane animation production.

Algorithm name	Animation scene	Average render speed (FPS)	Average frame rate (FPS)	Maximum frame rate (FPS)	Minimum frame rate (FPS)
Algorithm 1	The horse	29.5	28.7	32.5	21.8
Algorithm 1	Aeroplane	35.2	33.8	36.7	28.4
Algorithm 2	The horse	25.2	24.3	27.1	20.6
Algorithm 2	Aeroplane	28.3	27.1	29.9	23.2
Algorithm 3	The horse	26.8	26.1	31.2	21.5
Algorithm 3	Aeroplane	31.2	29.7	32.5	25.4
Algorithm 4	The horse	20.1	19.5	23.7	17.2
Algorithm 4	Aeroplane	22.2	20.6	25.1	17.8

visual quality effects of the four algorithms are showcased in Table 2.

According to the comparison results in Table 1, the proposed 3D mesh animation optimization algorithm (Algorithm 1) showed good visual quality effects on multiple test data, retaining details and presenting natural and clear animation effects. However, other algorithms suffered from some degree of detail loss or over simplification. This result indicated that the proposed 3D mesh animation optimization algorithm performed better than the comparison algorithm on different datasets, and had high practicality. In addition, to further compare the performance of the four algorithms, the study applied the four algorithms to different animation productions, compared the rendering speed and frame rate of the four algorithms in different animation productions, and analyzed the performance of the four algorithms. The comparison results of rendering speed and frame rate of four algorithms in tablecloth and fast-paced animation production are shown in Figure 12.

The rendering speed serves as the quantity of frames that the algorithm can render per second, and the higher the value, the faster the algorithm's rendering speed. The average frame rate was used to evaluate the smoothness of an animation, and the higher the average frame rate, the better the overall smoothness of the animation. As shown in Figure 12 (a), in the animation production of cloth, Algorithm 1 proposed in the study had the highest rendering speed of 31.3 FPS, significantly higher than Algorithm 2's 23.8 FPS, Algorithm 3's 24.8 FPS, and Algorithm 4's 24.3 FPS. In addition, Algorithm 1 had a frame rate of 30.4 FPS, which was also significantly better than the comparison algorithm. As shown in Figure 12 (b), in the production of fast-paced animation, Algorithm 1 had a rendering speed and frame rate of 26.2 FPS and 25.3 FPS, respectively, which were significantly better than the comparison algorithms. The relevant outcomes indicated that Algorithm 1 had better optimization performance in tablecloth and fast-paced animation production. The comparison results of rendering speed and frame rate of four algorithms in horse and airplane animation production are shown in Table 3.

According to Table 3, the average rendering speed of Algorithm 1 in horse animation production was 29.5 FPS, which was the highest among the four algorithms. Next were Algorithm 3 and Algorithm 2, with 26.8 FPS and 25.2 FPS respectively. Algorithm 4 had the lowest rendering speed of 20.1 FPS. In the animation production of airplanes, Algorithm 1 still had the highest average rendering speed, reaching 35.2 FPS. Next were Algorithm 3 and Algorithm 2, which were 31.2 FPS and 28.3 FPS respectively. The rendering speed of Algorithm 4 was still the lowest, at 22.2 FPS. This indicated that Algorithm 1 significantly outperformed the other three algorithms in rendering speed, whether it is the animation of horses or airplanes. In addition, in the animation production of horses, the average frame rate of Algorithm 1 was 28.7 FPS, which was also the highest among the four algorithms. The average frame rates of Algorithm 3 and Algorithm 2 were 26.1 FPS and 24.3 FPS, respectively, while Algorithm 4 had the lowest average frame rate of 19.5 FPS. In the animation production of airplanes, Algorithm 1 still had the highest average frame rate of 33.8 FPS. Next were Algorithm 3 and Algorithm 2, with 29.7 FPS and 27.1 FPS respectively. Algorithm 4 had the lowest average frame rate of 20.6 FPS. In summary, in the animation production of horses and airplanes, Algorithm 1 proposed in the study was markedly more excellent than the other three comparative algorithms in terms of rendering speed and frame rate. The optimization performance of Algorithm 1 performed well in two different types of animation production, horse and airplane, indicating that the algorithm has good universality and stability. In addition, since the introduction of the time factor can further optimize the fluency and coherence of the animation, it will help generate more natural and realistic animation effects. Therefore, exploring different architectures incorporating time factors is a worthwhile goal to pursue in future research.

V. CONCLUSION

This study addressed the issues of large computational complexity and unnatural effects in traditional 3D mesh animation production methods. It innovatively integrated IP, nonlinear constraint optimization, and machine learning algorithms to construct a better type of 3D mesh animation optimization algorithm. This algorithm achieved smoother and more realistic animation effects by automatically or semi-automatically adjusting attributes such as mesh vertex positions and normal directions. The experimental results showed that the new algorithm had the highest rendering speed at 31.3 FPS, significantly higher than Algorithm 2's 23.8 FPS, Algorithm 3's 24.8 FPS, and Algorithm 4's 24.3 FPS. In addition, Algorithm 1 had a frame rate of 30.4 FPS, which was also significantly better than the comparison algorithm. The above data fully demonstrated the effectiveness of the algorithm in improving the efficiency and quality of animation production. However, there are still some shortcomings in this study. When dealing with extremely complex animation scenes, the algorithm's runtime may increase and further optimization is needed to improve real-time performance. In addition, for certain types of animation effects, it may be necessary to make targeted adjustments to the algorithm on the grounds of domain knowledge. Future work will focus on these shortcomings to further improve the algorithm and expand its application scope.

REFERENCES

[1] D. Varılsüha, "3D inversion of magnetotelluric data by using a hybrid forward-modeling approach and mesh decoupling," *Geophysics*, vol. 85, no. 5, pp. E191–E205, Sep. 2020, doi: 10.1190/geo2019-0202.1.

[2] Y. Yuan, S. Yan, and Q. Fang, "Light transport modeling in highly complex tissues using the implicit mesh-based Monte Carlo algorithm," *Biomed. Opt. Exp.*, vol. 12, no. 1, p. 147, Jan. 2021, doi: 10.1364/boe.411898.

[3] J. Hwang, G. Park, I. H. Suh, and T. Kwon, "Primitive object grasping for finger motion synthesis," *Comput. Graph. Forum*, vol. 40, no. 1, pp. 266–278, Feb. 2021, doi: 10.1111/cgf.14187.

[4] X. Zou, H. He, Y. Wu, Y. Chen, and M. Xu, "Automatic 3D point cloud registration algorithm based on triangle similarity ratio consistency," *IET Image Process.*, vol. 14, no. 14, pp. 3314–3323, Dec. 2020, doi: 10.1049/iet-ipr.2019.1087.

[5] A. A. Borzunov, D. V. Lukyanenko, E. I. Rau, and A. G. Yagola, "Reconstruction algorithm of 3D surface in scanning electron microscopy with backscattered electron detector," *J. Inverse Ill-Posed Problems*, vol. 29, no. 5, pp. 753–758, Oct. 2021, doi: 10.1515/jiip-2020-0136.

[6] N. Wei and J. L. Walteros, "Integer programming methods for solving binary interdiction games," *Eur. J. Oper. Res.*, vol. 302, no. 2, pp. 456–469, Oct. 2022, doi: 10.1016/j.ejor.2022.01.009.

[7] M. S. Cecconello, M. T. Mizukoshi, and W. Lodwick, "Interval nonlinear initial-valued problem using constraint intervals: Theory and an application to the Sars-Cov-2 outbreak," *Inf. Sci.*, vol. 577, pp. 871–882, Oct. 2021, doi: 10.1016/j.ins.2021.08.045.

[8] Y.-C. Sun, W. Zhang, H. Ren, X. Bao, J.-K. Xu, N. Sun, Z. Yang, and X. Chen, "3D seismic-wave modeling with a topographic fluid–solid interface at the sea bottom by the curvilinear-grid finite-difference method," *Bull. Seismolog. Soc. Amer.*, vol. 111, no. 5, pp. 2753–2779, Oct. 2021, doi: 10.1785/0120200363.

[9] M. Hasanvand, "Machine learning methodology for identifying vehicles using image processing," *Artif. Intell. Appl.*, vol. 1, no. 3, pp. 170–178, Jul. 2023, doi: 10.47852/bonviewaia3202833.

[10] K. Bhosle and V. Musande, "Evaluation of deep learning CNN model for recognition of devanagari digit," *Artif. Intell. Appl.*, vol. 1, no. 2, pp. 114–118, Feb. 2023, doi: 10.47852/bonviewaia3202441.

[11] P. Zheng, Q. Liu, J. Lou, C. Lian, and D. Lin, "A free-form surface flattening algorithm that minimizes geometric deformation energy," *IET Image Process.*, vol. 16, no. 9, pp. 2544–2556, Jul. 2022, doi: 10.1049/ipr2.12508.

[12] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021, doi: 10.1109/JIOT.2020.3011726.

[13] M. Dvorožňák, D. Sýkora, C. Curtis, B. Curless, O. Sorkine-Hornung, and D. Salesin, ''Monster mash: A single-view approach to casual 3D modeling and animation,'' *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–12, Nov. 2020, doi: 10.1145/3414685.3417805.

[14] G. Luo, Z. Deng, X. Zhao, X. Jin, W. Zeng, W. Xie, and H. Seo, ''Spatio-temporal segmentation based adaptive compression of dynamic mesh sequences,'' *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 1, pp. 1–24, Feb. 2020, doi: 10.1145/3377475.

[15] T. Wu, ''Predictive search for capacitated multi-item lot sizing problems,'' *INFORMS J. Comput.*, vol. 34, no. 1, pp. 385–406, Jan. 2022, doi: 10.1287/ijoc.2021.1073.

[16] W. Xia, J. C. Vera, and L. F. Zuluaga, ''Globally solving nonconvex quadratic programs via linear integer programming techniques,'' *INFORMS J. Comput.*, vol. 32, no. 1, pp. 40–56, Jan. 2020, doi: 10.1287/ijoc.2018.0883.

[17] D. Zhao, Y. Liu, and H. Li, ''Fast-time complete controllability of nonlinear fractional delay integrodifferential evolution equations with nonlocal conditions and a parameter,'' *Math. Methods Appl. Sci.*, vol. 45, no. 10, pp. 5649–5669, Jul. 2022, doi: 10.1002/mma.7993.

[18] S. Mondal, M. P. Singh, A. Kumar, S. Chattopadhyay, A. Nandy, Y. Sthanikam, U. Pandey, D. Koner, L. Marisiddappa, and S. Banerjee, ''Rapid molecular evaluation of human kidney tissue sections by in situ mass spectrometry and machine learning to classify the nephrotic syndrome,'' *J. Proteome Res.*, vol. 22, no. 3, pp. 967–976, Jan. 2023, doi: 10.1021/acs.jproteome.2c00768.

[19] H. Li, T. Kondoh, P. Jolivet, K. Furuta, T. Yamada, B. Zhu, K. Izui, and S. Nishiwaki, ''Three-dimensional topology optimization of a fluid–structure system using body-fitted mesh adaption based on the level-set method,'' *Appl. Math. Model.*, vol. 101, pp. 276–308, Jan. 2022, doi: 10.1016/j.apm.2021.08.021.

[20] Z. Yu, L. Xia, G. Xu, C. Wang, and D. Wang, ''Improvement of the three-dimensional fine-mesh flow field of proton exchange membrane fuel cell (PEMFC) using CFD modeling, artificial neural network and genetic algorithm,'' *Int. J. Hydrogen Energy*, vol. 47, no. 82, pp. 35038–35054, Sep. 2022, doi: 10.1016/j.ijhydene.2022.08.077.

[21] F. Li, X. Cai, L. Gao, and W. Shen, ''A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems,'' *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1390–1402, Mar. 2021, doi: 10.1109/TCYB.2020.2967553.

[22] F. Yan, W. Li, and J. Zhang, ''Simultaneous synthesis of heat-integrated water networks by a nonlinear program: Considering the wastewater regeneration reuse,'' *Chin. J. Chem. Eng.*, vol. 44, pp. 402–411, Apr. 2022, doi: 10.1016/j.cjche.2020.11.044.

[23] R. A. Dollar and A. Vahidi, ''Multilane automated driving with optimal control and mixed-integer programming,'' *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 6, pp. 2561–2574, Nov. 2021, doi: 10.1109/TCST.2020.3046570.

[24] A. Khassiba, F. Bastin, S. Cafieri, B. Gendron, and M. Mongeau, ''Two-stage stochastic mixed-integer programming with chance constraints for extended aircraft arrival management,'' *Transp. Sci.*, vol. 54, no. 4, pp. 897–919, Jul. 2020, doi: 10.1287/trsc.2020.0991.

[25] Y. Yang, Y. Li, X. Liu, and D. Huang, ''Adaptive neural network control for a hydraulic knee exoskeleton with valve deadband and output constraint based on nonlinear disturbance observer,'' *Neurocomputing*, vol. 473, pp. 14–23, Feb. 2022, doi: 10.1016/j.neucom.2021.12.010.

[26] Y. Hwang, C. M. Kang, and W. Kim, ''Robust nonlinear control using barrier Lyapunov function under lateral offset error constraint for lateral control of autonomous vehicles,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1565–1571, Feb. 2022, doi: 10.1109/TITS.2020.3023617.

[27] L. Kumarapu and P. Mukherjee, ''AnimePose: Multi-person 3D pose estimation and animation,'' *Pattern Recognit. Lett.*, vol. 147, pp. 16–24, Jul. 2021, doi: 10.1016/j.patrec.2021.03.028.

[28] Q. Wang, H. Xi, F. Deng, M. Cheng, and G. Buja, ''Design and analysis of genetic algorithm and BP neural network based PID control for boost converter applied in renewable power generations,'' *IET Renew. Power Gener.*, vol. 16, no. 7, pp. 1336–1344, Nov. 2021, doi: 10.1049/rpg2.12320.

[29] M. Badi, S. Mahapatra, and S. Raj, ''Hybrid BOA-GWO-PSO algorithm for mitigation of congestion by optimal reactive power management,'' *Optim. Control Appl. Methods*, vol. 44, no. 2, pp. 935–966, Mar. 2023, doi: 10.1002/oca.2824.

[30] D. Khachai, R. Sadykov, O. Battaia, and M. Khachay, ''Precedence constrained generalized traveling salesman problem: Polyhedral study, formulations, and branch-and-cut algorithm,'' *Eur. J. Oper. Res.*, vol. 309, no. 2, pp. 488–505, Sep. 2023, doi: 10.1016/j.ejor.2023.01.039.

[31] M. Li, Y. Gu, Y. Zhang, X. Gao, S. Ge, and G. Wei, ''Quantitative prediction of ternary mixed gases based on an SnO_2 sensor array and an SSA-BP neural network model,'' *Phys. Chem. Chem. Phys.*, vol. 25, no. 15, pp. 10935–10945, Apr. 2023.

[32] C. Soumya, P. Raj, B. Deepanraj, and N. Senthilkumar, ''Islanding power quality detection using lighting search optimization with deep learning model on distributed generation systems,'' *Renew. Energy Focus*, vol. 43, pp. 74–83, Dec. 2022, doi: 10.1016/j.ref.2022.08.007.

[33] A. Tong, X. Tang, H. Liu, H. Gao, X. Kou, and Q. Zhang, ''Differentiation of NaCl, NaOH, and β-Phenylethylamine using ultraviolet spectroscopy and improved adaptive artificial bee colony combined with BP-ANN algorithm,'' *ACS Omega*, vol. 8, no. 13, pp. 12418–12429, Apr. 2023, doi: 10.1021/acsomega.3c00271.

[34] F. S. Gharehchopogh, M. Namazi, L. Ebrahimi, and B. Abdollahzadeh, ''Advances in sparrow search algorithm: A comprehensive survey,'' *Arch. Comput. Methods Eng.*, vol. 30, no. 1, pp. 427–455, Jan. 2023, doi: 10.1007/s11831-022-09804-w.

[35] Y. Kawano, F. Valdez, and O. Castillo, ''Fuzzy combination of moth-flame optimization and lightning search algorithm with fuzzy dynamic parameter adjustment,'' *Computación Sistemas*, vol. 26, no. 2, pp. 743–757, Jun. 2022, doi: 10.13053/cys-26-2-4269.

[36] Y. A. Sarumaha, D. R. Firdaus, and I. Moridu, ''The application of artificial bee colony algorithm to optimizing vehicle routes problem,'' *J. Inf. Syst., Technol. Eng.*, vol. 1, no. 1, pp. 11–15, May 2023, doi: 10.61487/jiste.v1i1.9.

[37] Y. Yu and M. Yao, ''When convolutional neural networks meet laser-induced breakdown spectroscopy: End-to-end quantitative analysis modeling of ChemCam spectral data for major elements based on ensemble convolutional neural networks,'' *Remote Sens.*, vol. 15, no. 13, p. 3422, Jul. 2023, doi: 10.3390/rs15133422.

[38] Y. Yu, M. Yao, J. Huang, and X. Xiao, ''When process analysis technology meets transfer learning: A model transfer strategy between different spectrometers for quantitative analysis,'' *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–19, 2024, doi: 10.1109/TIM.2024.3353273.

JING JIANG was born in Anqing, Anhui, China, in November 1988. She received the bachelor's degree in art and design from Yunnan University, in 2010, and the master's degree in art from the Kunming University of Science and Technology, in 2014. Since 2014, she has been teaching with the Art School, Anhui Xinhua University. During this period, she has published eight academic papers and presided over ten textbook research projects. Her research interest includes animation new media.

XIAOJUN WANG was born in Hefei, Anhui, China, in November 1986. He received the bachelor's degree in painting from Huaibei Normal University, in 2008, and the master's degree in fine arts from Anhui Normal University, in 2011. Since 2012, he has been teaching with the Art School, Anhui Xinhua University. During this period, he published a total of 15 academic papers and presided over 12 textbook research projects. His research interests include visual communication design and fine arts.