ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Design and Optimization of Unity-based Interactive Virtual Simulation System for Digital Media Arts

Yanan Zhang[1,*]

[1] Department of Digital Media, Tangshan University, Tangshan, Hebei, 063000, China

SUMMARY: *With the continuous development of computer simulation technology and digital media art, the traditional digital media art interactive virtual simulation system can not meet the needs of users. This paper compares several mainstream system development software, selects Unity3D software to carry out the preliminary design work of the system, and finally designs a Unity-based interactive virtual simulation system for digital media art. However, in the process of system performance testing, it is found that the redundancy of the 3D model leads to excessive system response time, memory and CPU occupation, which directly and seriously affects the user experience. In response to this problem, the collapsing algorithm realizes the simplification of the 3D model by constructing the appropriate weighted error function, aiming at enhancing the performance of the system. After the three-dimensional model simplification, it is obvious that the CPU occupancy value is released, and its value is reduced from 0.5~0.7 to 0~0.35, which proves that the three-dimensional model simplification based on the edge folding algorithm is effective in optimizing the performance of the interactive virtual simulation system of the digital media art, so as to make it better serve the field of the digital media art. The research in this paper can provide theoretical guidance for the design and optimization of the interactive virtual simulation system of digital media art, so as to promote the high-quality development of the field of digital media art.*

KEYWORDS: *Unity3D software; edge folding algorithm; weighted error function; digital media art; interactive virtual simulation system*

1 Introduction

In the context of the rapid development of computers and the Internet, digital media art, as an emerging field of the deep integration of science and technology and art and design, has become the core driving force of contemporary art creativity [1]. The creation of digital media art cannot be separated from various advanced digital technologies. With the continuous innovation of science and technology and the widespread popularization of the Internet, digital media art has developed rapidly, breaking through the boundaries of traditional art. It not only includes traditional digital images, video, audio and other technologies, but also expands to three-dimensional modeling, animation, programming, virtual reality (VR) and augmented reality (AR) and other cutting-edge technology fields, which greatly enriches the means of creating digital content [2-5]. The application scope of digital media art is very wide, covering a wide range of industries such as advertising design (e.g., creative videos, dynamic posters, H5 interactive advertisements), game development (including scenes, props, and game character production), film and television production, as well as education, etc., which not only provides

*max13831515656@163.com
https://doi.org/10.65102/is2026646 Published: 30 April, 2026

creative inspirations and technical support for various fields, but also promotes the innovation and rapid development of the industry [6-9].

With the continuous progress of science and technology, digital media art continues to integrate cutting-edge technology and realize innovative breakthroughs. In particular, virtual simulation technology can create a virtual simulation system with a sense of reality, which brings a highly real-time interactive immersive experience for digital media art creation [10]. With the help of VR, AR and mixed reality technologies, art creators can break through the traditional limitations of the creative space, build realistic virtual reality environments, and freely carry out artistic design and creation in three-dimensional virtual space [11, 12]. Unity, as a real-time 3D interactive content creation and operation platform with cross-platform, real-time rendering and other characteristics, brings diversified experiences and effects to digital media art creation [13]. Unity-based 3D virtual simulation technology combined with VR equipment, through the roaming function to create a realistic scene of interactive and immersive experience, to improve the user in the face of the reality of the operation of the scene, the correct understanding of the equipment, so as to enhance the correctness of the user's actual operation of the power to avoid repetitive errors, in the field of education and training and design has been widely used [14-16].

In this paper, with reference to the technical characteristics and advantages of several existing system development software, Unity3D software was finally selected to complete the design of the interactive virtual simulation system for digital media art, and on the basis of the system operating environment, system performance testing and analysis was carried out, and it was found that the redundancy of the three-dimensional model led to the performance of the system was not ideal. In view of the above problems, we propose to use the edge-folding algorithm to construct a suitable weighted error function to ensure that the simplified model can be streamlined to the greatest extent without distortion, and carry out a comparison test of the response time before and after the optimization, and a comparison test of the memory and CPU occupation before and after the optimization to verify the effect of the 3D model simplification based on the edge-folding algorithm on the optimization of the performance of the interactive virtual simulation system for digital media art. The effect.

2 Art Interactive Virtual Simulation System Design

2.1 Unity3D Technology Research and Analysis

2.1.1 Introduction to Unity3D

Unity3D is a game development tool developed by Unity Technologies that allows players to easily create interactive content such as real-time 3D animations and architectural visualizations on multiple platforms, and is a professional game engine that integrates a wide range of features in one. Unity3D offers a wide range of features inside, such as sound, graphics, rendering, and physics, and has built-in support for 3D Max, Maya, and many other file formats. Unity3D provides sound, graphics, rendering, and physics, and has a powerful built-in editor that supports 3D Max, Maya, and other file formats. Developers don't need to understand the underlying complex technologies, but only need to utilize some programming languages to develop and design high-quality game products in it. Games developed with Unity3D can be released on various platforms such as Windows, Mac, iPhone, and Android. In addition, Unity3D realizes publishing games for use in web browsers with the help of Unity web player plugin.

2.1.2 Unity3D Technical Features

Unity3D technical features shown in Figure 1, through the long-term use of Unity3D software, summarized eleven Unity3D technical features, which are comprehensive editing, graphics engine, resource import, shaders, one-click deployment, networking, Lightmapping light and shadow effects, physical effects, terrain editor, powerful performance analyzer, scripting.

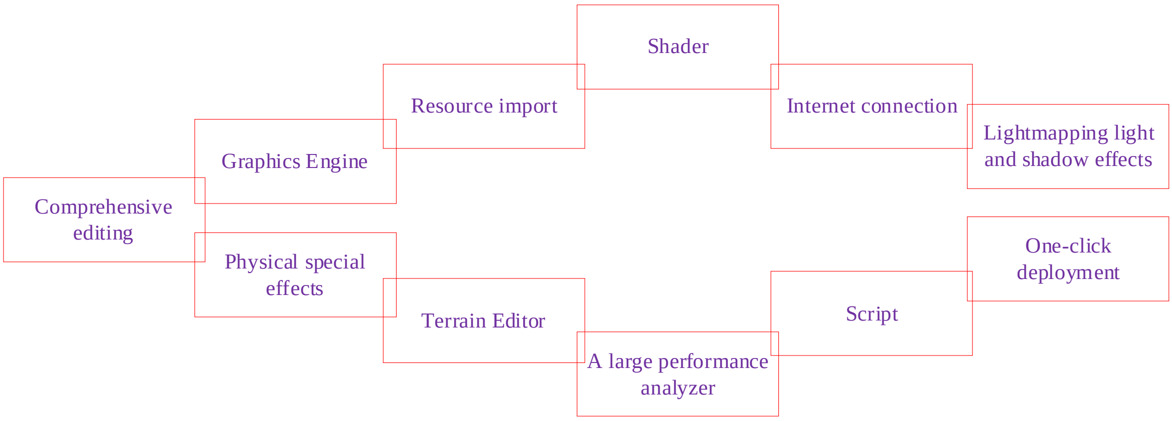

Figure 1: Features of Unity3D technology

2.1.3 Advantages of Unity3D technology

Unity3D, as the key technology for the development of interactive virtual simulation systems for digital media arts, has many unique features and advantages. Compared with other development engines, Unity3D is characterized by rich and diverse plug-ins and strong platform adaptability, i.e., one development environment can be used to develop digital media arts that can simultaneously adapt to different formats. From the perspective of digital media art development, Unity3D technology has the following two main advantages.

From the aspect of mobile platform, Unity3D has rich modules, shortens the development cycle, and is very popular with small development teams. The main reason why most teams choose Unity3D when choosing to build a digital media art simulation system is that Unity3D has a lot of ready-made modules that can be directly invoked, which greatly reduces the team's development costs. Therefore, Unity3D technology is favored by small team developers.

Unity3D has high portability and can realize very smooth support for multiple platforms. From the perspective of development speed and long-term development, Unity3D still has its own advantages.

(1) In general, Unity3D has its own online store, the development team can choose the effects suitable for their own development needs from the website, and purchase them without spending time on development.

(2) Unity3D has a very good editor development function, the editor interface is friendly, in the development process, developers can easily plan a specific set of editor. These are very critical for small development teams.

(3) From the perspective of development, in this society of rapid development of digital media art, when the efficiency of mobile devices has increased into the era of spelling expression, if you can utilize Unity3D virtual simulation technology, to produce near-real 3D effects is crucial.

2.2 System design

2.2.1 General structure

The framework of the interactive virtual simulation system for digital media art is shown in Figure 2, which mainly includes three parts: host computer, PLC and virtual environment. The simulation software mainly includes the virtual scene, interactive interface and communication data processing, the equipment in the virtual scene and the experimental environment can be modeled in 3D, while the construction of the interactive interface and the processing of communication data can be realized by writing scripts in the Unity3d engine development platform.

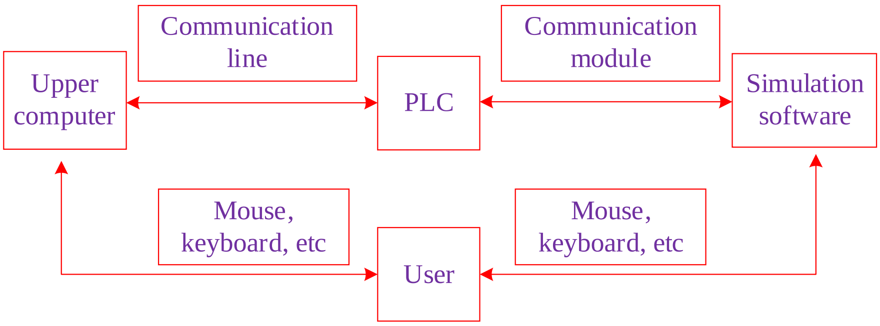

Figure 2: System framework

2.2.2 Development tools

In recent years, the rapid development of virtual reality technology also makes a lot of choices of related development tools, such as Unity3D, Unreal Engine, Vega Prime, WTK, VirTools, etc., and Unity3D development engine is chosen in the study. Its comparison with other tools mainly has the following points:

(1) Unity3D provides developers with a free version, which fully meets the development needs of the simulation software side, thereby reducing development costs

(2) The software supports 2D and 3D development, and is compatible with all kinds of common media files, not only can it build some models, but also import external 3D models.

(3) The programming languages used in this software are C# and JavaScript, of which C# is relatively easy to learn and use.

(4) Unity3D software provides Asset Store function, you can choose some extension resources in the store to save development time.

2.2.3 Three-dimensional modeling

Before the establishment of the three-dimensional model, the first thing that is needed is an in-depth analysis of digital media art and scene material collection, the study through the 3DMAX software to simulate the digital media art scene modeling work, after the completion of the three-dimensional model production. After the three-dimensional modeling is completed, the model can be mapped, mapping production to fully utilize the UVW mapping function of 3DSMAX software to ensure that the mapping matches the real situation as much as possible.

2.2.4 Software development

Using Unity3D to develop the simulation software end, some of the following steps are mainly carried out:

(1) Build the interactive interface to provide the basic functions of software operation and the interactive interface after the software enters the test.

(2) Import the model into Unity3D and further optimize the model to improve the simulation.

(3) Write software scripts to realize the logic of controlling the model animation and the data processing of communication with PLC.

(4) Test the simulation software side, mainly for real-time control test of PLC to the simulation software side.

2.2.5 Functional realization

After importing the model into Unity3D, we mainly adjust the model material and animation, and then realize the corresponding animation when clicking the button and the change of the IO point buttons and lights through the script. Example will be the overall process animation, through the sample script to realize the click animation to play the overall process.

2.3 System testing

This subsection will test the performance of the system through a simple digital media art interaction experiment, selected digital media art users as test users to test the system to use, tested the complete functionality of the system process, and issue, stability and timeliness, and make a detailed analysis of its performance.

2.3.1 System operating environment

For the system usage test, the client and server installation and operation environments are shown in Table 1. Based on the table, it can be seen that on the system client installation environment, the processor is Intel(R) Core(TM) i7-7700HQ CPU @2.60 GHz and the operating system is Microsoft(R) Windows 8. On the system server installation environment, the processor is Intel(R) Xeon(R) CPU E5-2682 v4 @2.70 GHz, and the operating system Linux version 2.6.18-164.el5 CentOS release 6.4 (Final).

Table 1: System operating environment

System operating environment	System client installation environment	System server installation environment
Processor	Intel(R) Core(TM) i7-7700HQ CPU @2.60GHz	Intel(R) Xeon(R) CPU E5-2682 v4 @2.70 GHz
Operating system	Microsoft(R) Windows 8	Linux version 2.6.18-164.el5 CentOS release6.4(Final)

2.3.2 System performance testing

After the system was written, it was put to use in an enterprise's digital media arts program with four product managers, four architects, and 100 testers. The platform was first tested for centralized use, and then kept open for one week for testers to experiment.

(1) System centralized performance test

Each tester in the digital media art project is equipped with a computer installed with the client software of the system, and then all the testers carry out experiments at the same time, and each tester is responsible for recording his or her own experimental results and response time, and the experimental simulation results are consistent with the expected results in the case of no error in the input files, which means that the system functions are correct and can work normally.

In order to test the response time of the system for multi-user concurrent access, the above

10 test users are required to apply for simulation in groups after the design is completed, which are divided into 10-member group, 20-member group, 30-member group, 40-member group, 50-member group, 60-member group, 70-member group, 80-member group, 90-member group, 100-member group, and each group of testers clicks on the simulation button at the same time to submit the application, and records the time used by each group for the return of their respective simulation results, so that the response time of the system can be analyzed. Time, so as to system response time statistics, user login response time statistics shown in Table 1, interactive virtual simulation response time statistics shown in Table 2. The above results show that: concurrent access, the greater the number of users, the longer the average response time, user login and interactive virtual simulation of the maximum response time of 102.66s, 102.44s, but the response delay time are within the acceptable range, the feeling that the system response speed appears to be a significant decline, greatly affecting the user's experience of the use of interactive virtual simulation system of digital media art.

Table 2: Statistics on user login response time

Number of concurrent users	Average response time/s	Maximum response time/s
10	1.24	1.21
20	7.32	13.27
30	13.49	27.36
40	20.92	41.13
50	34.51	67.32
60	39.26	71.89
70	45.79	79.47
80	55.25	86.21
90	59.64	94.53
100	68.93	102.66

Table 3: Interactive virtual simulation response time statistics

Number of concurrent users	Average response time	Maximum response time.
10	1.37	1.39
20	7.52	13.35
30	13.88	27.55
40	20.93	41.45
50	34.72	59.22
60	38.88	71.81
70	45.58	79.3
80	54.77	85.83
90	59.48	94.15
100	68.46	102.44

(2) Memory and CPU Occupancy

Based on the above test environment, this paper uses Edge, Chrome and Firefox three commonly used browsers to test the software's CPU and memory usage, through the browser's debugging tool to compare and analyze the web page's memory usage, memory usage as shown in Figure 3, CPU usage as shown in Figure 4. Demonstration of the digital media art interactive virtual simulation system memory and CPU consumption test results can be seen, the digital media art interactive virtual simulation system is compatible with commonly used Edge, Chrome and Firefox browsers, the browser's memory consumption during operation is controlled within 893MB. The CPU usage of the browser process during the running process is maintained between 0.5 and 0.7. Based on the above analysis, the interactive virtual simulation system of digital media art can run stably in the test environment, but the performance is not satisfactory, which is greatly due to the redundancy of 3D models affecting the overall performance of the system, and will be optimized in the following section to deal with this situation.

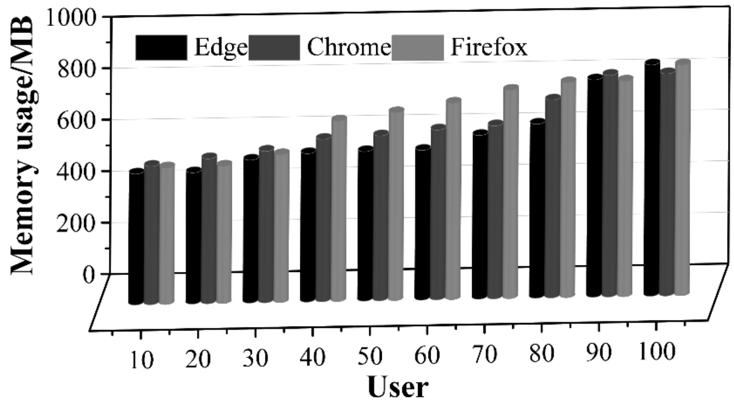

Figure 3: Memory usage

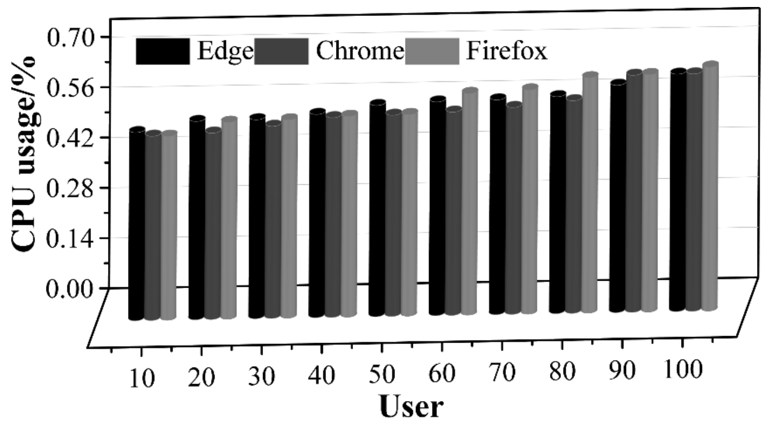

Figure 4: CPU usage

3 Edge-folding algorithm for 3D model optimization in the system

Through the content of the second chapter, it can be seen that the system performance is less than ideal, the great reason is that the redundancy of 3D model affects the overall performance of the system, this chapter for the performance of the art interactive virtual simulation system is not ideal, proposed the use of edge folding algorithm to optimize the processing of the 3D model in the system, thus reducing the redundancy of the 3D model in the system, in order to achieve the optimization of the art interactive virtual simulation system.

3.1 Edge Folding Algorithm

Edge folding algorithm is based on the folding of edges, belongs to the geometric elements algorithm, in this algorithm, in different regions of the model surface to generate the envelope, here the envelope refers to the generation of a bounded enclosing surfaces, the mesh is simplified under the overall control of each envelope, the basic operation is the triangle folding: the triangle T as a folding object, its three vertices V1, V2, V3 aggregated to generate a new vertex V, Figure 5 The number of mesh faces after folding is obviously reduced, and then the folding operation is continued sequentially in other envelope regions, and the cycle is executed until the simplification requirements are met.

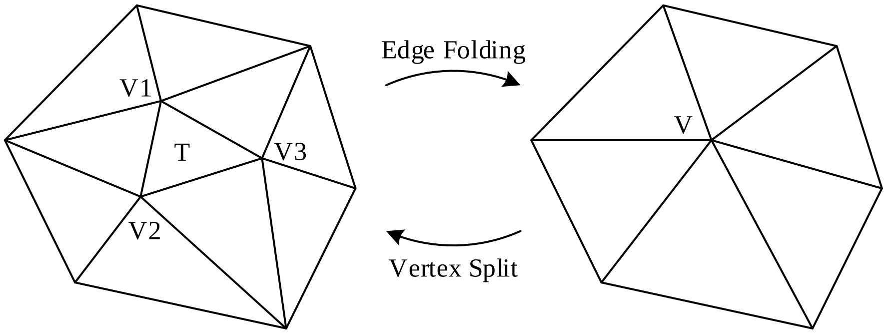

Figure 5: Schematic diagram of side folding

To solve the corresponding problem, the quadratic error metric algorithm chosen here defines a variable Δ describing the cost of edge folding and contraction when selecting a suitable edge for folding and contraction, and for each vertex v in the mesh, a symmetric error matrix Q of 4×4 is built corresponding to it, then the vertex $v_{i0}=\begin{bmatrix}x_{i0} & y_{i0} & z_{i0} & 1\end{bmatrix}^T$ has an error function of its binomial form, expressed as the sum of squares of distances of the vertices from the set of associated planes. Suppose that for a contracted edge (v_1,v_2), whose contracted vertices become v, the error matrix for the new vertices is $Q=Q1+Q2$.

The expression of the error function is shown in equation (1):

$$\Delta(T_i)=v_{i0}^T Q_i v_{i0} \tag{1}$$

The expression for Q_i in the above equation is shown in (2):

$$Q_i=\begin{bmatrix} q_{i11} & q_{i12} & q_{i13} & q_{i14} \\ q_{i12} & q_{i22} & q_{i23} & q_{i34} \\ q_{i13} & q_{i23} & q_{i33} & q_{i34} \\ q_{i4} & q_{i24} & q_{i34} & q_{i44} \end{bmatrix} \tag{2}$$

Substituting equation (2) into the expansion of equation (1) yields equation (3):

$$\begin{aligned}\Delta(T_i)=q_{i1}x_{i0}^2+2q_{i12}x_{i0}y_{i0}+2q_{i13}x_{i0}z_{i0}+2q_{i14}x_{i0}\\ +q_{i22}y_{i0}^2+2q_{i23}y_{i0}z_{i0}+2q_{i24}y_{i0}+q_{i33}z_{i0}^2+2q_{i34}z_{i0}\end{aligned} \tag{3}$$

From Eq. (1), we can intuitively see that the calculation of the contraction cost of edge folding requires the position information of the new vertices after the contraction, and there are two ways to determine the position information of the new vertices: a simpler way is to choose a position that minimizes the contraction cost of Δv_{i0} from either v_1, v_2 or $(v_1+v_2)/2$, which is to choose from the two vertices as well as the position of the midpoint of the folded edges; and the other way is to directly carry out numerical computations The other method is to do a direct numerical calculation, i.e., to make the first-order derivative of Eq. (3) zero, and thus to obtain the position of the vertex that minimizes the value of the error function. The partial derivatives of x_{i0}, y_{i0}, z_{i0} in Eq. (3) are obtained separately and made zero. As shown in (4):

$$\frac{\partial\Delta(T_i)}{\partial x_{i0}} = \frac{\partial\Delta(T_i)}{\partial y_{i0}} = \frac{\partial\Delta(T_i)}{\partial z_{i0}} = 0 \tag{4}$$

Then the above calculation is equivalent to solving equation (5):

$$\begin{bmatrix} q_{i11} & q_{i22} & q_{i13} & q_{i14} \\ q_{i12} & q_{i22} & q_{i23} & q_{i24} \\ q_{i13} & q_{i23} & q_{i33} & q_{i34} \\ 0 & 0 & 0 & 1 \end{bmatrix} v_{i0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5}$$

where q_{iij} is the corresponding element in matrix Q. If the coefficient matrix $q_i = \begin{bmatrix} q_{i11} & q_{i12} & q_{i13} & q_{i14} \\ q_{i12} & q_{i22} & q_{i23} & q_{i24} \\ q_{i13} & q_{i23} & q_{i33} & q_{i34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is an invertible matrix, then the position of the new vertex v_{i0} is shown in Eqn. (6), but if the matrix q_i is not invertible, the new vertex position will be chosen according to the first and simpler way position. That is:

$$v_{i0} = \begin{bmatrix} q_{i11} & q_{i12} & q_{i13} & q_{i14} \\ q_{i12} & q_{i22} & q_{i23} & q_{i24} \\ q_{i13} & q_{i23} & q_{i33} & q_{i34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{6}$$

3.2 Weighted error function with improved weights

In order to be able to reflect the local surface changes continuously and also to be able to select the obtained adaptively, this section constructs the concept of triangle importance and uses this definition as the weights to measure the quadratic error. The order in which the triangles are folded is determined by the magnitude of the folding cost, which is determined by the following three importance factors:

3.2.1 Average area of local triangles

If a localized surface contains a larger average area of triangular mesh, it means that the surface represents the flat part of the object, and because of its lower level of detail, it will not result in a lack of detail on the surface of the object after folding, otherwise it is just the opposite. The formula for the average area of localized triangles of triangle T_i is as follows:

$$\overline{A_t} = \frac{\sum_{i=1}^{m} A_i}{m} \tag{7}$$

In Eq. (7), A_i represents the area of each triangle adjacent to triangle T_i, and m denotes the number of triangles associated with T_i.

3.2.2 Regularity

Regularity is used to judge how close a triangle is to a normal triangle. If a triangle is closer to a normal triangle, then it should be folded so as to retain the faces of the triangle. The regularity of a triangle can be calculated by the side length method, the area method and the angle method, and the side length method is chosen for this section. The formula for the side length method is shown in equation (8):

$$r_e = \frac{l_3 - l_1}{l_2} \tag{8}$$

where l_1, l_2, l_3 represent the lengths of each side of the triangle and $l_1 \le l_2 \le l_3$ respectively. The regularity degree of the triangle, on the other hand, is represented by equation (9), which indicates that the triangle is equilateral when r_t is 1, and the range of values of r_t is $0 \le r_t \le 1$. i.e:

$$r_t = 1 - r_e = \frac{l_1 + l_2 - l_3}{l_2} \tag{9}$$

3.2.3 Sharpness

3D models tend to have a large number of triangular faces in regions with distinctive detail features, and the curvature of triangles under this region varies greatly, so local region sharpness is introduced and used to prevent regions with distinctive features from being over-optimized. In this section, the sharpness is defined as the sum of the approximate curvatures of the three vertices of a triangle with the following formula:

$$K_{\mathrm{t}} = \overline{K_{vi}} + \overline{K_{vj}} + \overline{K_{vk}} \tag{10}$$

where v_i, v_j, v_k denote the three vertices of T_i, and the approximate curvature formula for the vertex v_i is shown as follows, where v_j is the first-order neighboring point of v_i, and m is the number of edges containing the vertex v_i. i.e:

$$\overline{K_{vi}} = \frac{\sum_{j=1}^{m} \overline{K\left(v_i, v_j\right)}}{m} \tag{11}$$

From the above formula, it can be seen that the greater the sharpness, indicating that the geometric characteristics of the part of the more three-dimensional, if the priority folding will cause the object characteristics change, should be folded after the order, on the contrary, it represents the surface is more gentle, should be given priority to folding optimization.

The folding cost of a triangle is determined by the average area $\overline{A_t}$, regularity r_t, and sharpness K_t, the smaller $\overline{A_t}$, the larger r_t, and the larger K_t, the larger the cost of the folding, which indicates that the triangle has a higher degree of importance, and can be represented by (12) to denote the importance of the triangle:

$$\overline{w_t} = r_t \frac{K_t}{\overline{A_t}} \tag{12}$$

Combining QEM and importance, the error matrix of the triangle is calculated as follows:

$$Q(t) = \overline{w_t} Q_i \tag{13}$$

Fold the triangle T_i to the new vertex $v = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ with a folding error of:

$$\Delta(t) = v^T Q(t) v \tag{14}$$

where $Q(t)$ can be expressed as:

$$Q(t) = \begin{bmatrix} q_{t11} & q_{t12} & q_{t13} & q_{t14} \\ q_{t12} & q_{t22} & q_{t23} & q_{t24} \\ q_{t13} & q_{t23} & q_{t33} & q_{t34} \\ q_{t14} & q_{t24} & q_{t34} & q_{t44} \end{bmatrix} \tag{15}$$

And $\Delta(t)$ can be expressed as:

$$\begin{aligned} \Delta(t) = {} & q_{t11}x^2 + 2q_{t12}xy + 2q_{t13}xz + 2q_{t14}x \\ & + q_{t22}y^2 + 2q_{t23}yz + 2q_{t24}y + q_{t33}z^2 + 2q_{t34}z + q_{t44} \end{aligned} \tag{16}$$

3.3 Determination of new vertex position

From Eq. (16), it can be seen that the calculation of the folding error still needs the position information of the new vertex. Although the new vertex can be determined by the midpoint of the three edges, the three vertices or the center of gravity of the triangle, in order to ensure the accuracy of the calculation, the new vertex position is derived by using the method of applying a partial derivative to the folding error. I.e.:

$$\frac{\partial \Delta(t)}{\partial x} = \frac{\partial \Delta(t)}{\partial y} = \frac{\partial \Delta(t)}{\partial z} = 0 \tag{17}$$

It is listed in the equation below:

$$\begin{cases} 2q_{t11}x + 2q_{t12}y + 2q_{t13}z + 2q_{t14} = 0 \\ 2q_{t12}x + 2q_{t22}y + 2q_{t23}z + 2q_{t24} = 0 \\ 2q_{t13}x + 2q_{t23}y + 2q_{t33}z + 2q_{t34} = 0 \end{cases} \tag{18}$$

Rewritten in the following form, $v = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ is the new vertex after folding. i.e:

$$\begin{bmatrix} q_{t11} & q_{t12} & q_{t13} & q_{t14} \\ q_{t12} & q_{t22} & q_{t23} & q_{t24} \\ q_{t13} & q_{t23} & q_{t33} & q_{t34} \\ 0 & 0 & 0 & 0 \end{bmatrix} v = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{19}$$

Using q_t for $\begin{bmatrix} q_{t11} & q_{t12} & q_{t13} & q_{t14} \\ q_{t12} & q_{t22} & q_{t23} & q_{t24} \\ q_{t13} & q_{t23} & q_{t33} & q_{t34} \\ 0 & 0 & 0 & 0 \end{bmatrix}$, when q_t is invertible, the unique solution for the new vertex can be determined as follows:

$$v = \begin{bmatrix} q_{t11} & q_{t12} & q_{t13} & q_{t14} \\ q_{t12} & q_{t22} & q_{t23} & q_{t24} \\ q_{t13} & q_{t23} & q_{t33} & q_{t34} \\ 0 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{20}$$

When the determinant of q_t is zero, the above method will not be able to derive the coordinates of the new vertices, and if the midpoints and centers of the three edges are directly chosen as the collapsed vertices, the factor of the influence of the neighboring regions on the triangle is ignored. In this section, the coordinates of the new vertices are obtained by weighting the three vertices v_i, v_j, v_k of the triangle T_i with the following formula:

$$v = L_i v_i + L_j v_j + L_k v_k \tag{21}$$

where L_i, L_j, L_k are the weights of the three vertices and the sum of the three is 1. The expressions for the three weights are shown below:

$$\begin{cases} L_i = \dfrac{Av_i}{Av_i + Av_j + Av_k} \\ L_j = \dfrac{Av_j}{Av_i + Av_j + Av_k} \\ L_k = \dfrac{Av_k}{Av_i + Av_j + Av_k} \end{cases} \tag{22}$$

where Av_i, Av_j, Av_k are the sums of the areas of neighboring triangles at each vertex of triangle T, respectively.

3.4 Optimization of processes

The core idea of the edge folding algorithm is to construct a suitable weighted error function, calculate the error value of the edge folding, and sort the size, choose the edge with small cost to fold, and pay attention to the boundary problem, to ensure that the simplified model can achieve the maximum degree of streamlining on the basis of no distortion. The basic process is as follows:

(1) Read all the vertices and edges involved in the folding in the model.

(2) Calculate the QEM (v) of all vertices and QEM = QEM (v_1) + QEM (v_2) of all edges.

(3) Find the edge with the smallest value in the ordering of QEM.

(4) Collapse the edge and find the optimal new vertex v and remove v_1, v_2 and the associated faces, updating the vertex QEM to QEM (v_1) + QEM (v_2) and the associated edges.

(5) If the simplification requirement is met, end the algorithm, otherwise go to (3) to continue.

4 System optimization effect test

Through the previous test results of the Unity-based interactive virtual simulation system for digital media art, it can be seen that the main reason for the unsatisfactory performance of the system is the redundancy of the 3D model content. To address this situation, the 3D model simplification based on the edge folding algorithm is proposed, and in order to prove the effect of the 3D model simplification based on the edge folding algorithm on the optimization of the performance of the interactive virtual simulation system for digital media art, a comparison test of the response time before and after the optimization, and a comparison test of the memory and CPU usage before and after the optimization are carried out.

4.1 Response time comparison test before and after optimization

The system performance test environment is set unchanged, and the response time comparison test of system user login before and after optimization, and the response time comparison test of interactive virtual simulation before and after optimization are performed to verify the effect of 3D model simplification based on the edge-folding algorithm on the optimization of system response efficiency. The response time comparison test of the system before and after optimization is:

4.1.1 System user login response time comparison test

The above system performance test environment remains unchanged, and the system response time comparison test is conducted to verify the effect of 3D model simplification based on the edge-folding algorithm on the system optimization, and the results of the user login response time comparison are shown in Table 4, and the system user login response time data before optimization comes from Table 2, while the system user login response time data after optimization is required to be re-tested to obtain. Based on the data in the table, it can be seen that in terms of user login response time, the average response time and maximum response time of the digital media and art interactive virtual simulation system before optimization are [1.24s, 68.93s], [1.21s, 102.66s], and under the role of three-dimensional model simplicity based on the edge-folding algorithm, the average response time and maximum response time of the digital media and art interactive system are [0.24s, 68.93s], [1.21s, 102.66s], and the average response time and maximum response time of the digital media and art interactive system are [0.24s, 68.93s]. The average response time and maximum response time of the digital media art interactive system under the effect of 3D model simplification based on the edge folding algorithm are [0.83s, 19.24s], [0.95s, 29.31s], and the difference between the two is [0.41, 49.69], [0.26, 68.18], and the difference between the two is the magnitude of the response efficiency enhancement, which can be seen intuitively that the simplification of the

3D model based on the edge folding algorithm has the effect of the enhancement of the response efficiency of the system's user logging in.

Table 4: Comparison results of user login response times

Number of concurrent users	Before optimization		After optimization		Difference	
	Average response time/s	Maximum response time/s	Average response time/s	Maximum response time/s	Average response time/s	Maximum response time/s
10	1.24	1.21	0.83	0.95	0.41	0.26
20	7.32	13.27	2.34	5.13	4.98	8.14
30	13.49	27.36	7.09	6.55	6.4	20.81
40	20.92	41.13	8.35	8.44	12.57	32.69
50	34.51	67.32	8.95	17.33	25.56	49.99
60	39.26	71.89	14.19	18.85	25.07	53.04
70	45.79	79.47	16.09	18.9	29.7	60.57
80	55.25	86.21	16.4	24.76	38.85	61.45
90	59.64	94.53	17.81	26.35	41.83	68.18
100	68.93	102.66	19.24	29.31	49.69	73.35

4.1.2 Interactive virtual simulation response time comparison test

The interactive virtual simulation response time comparison results are shown in Table 5, the interactive virtual simulation response time data before the optimization comes from Table 3, and the interactive virtual simulation response time data after the optimization needs to be obtained by re-system performance testing. Through the data performance in the table, it can be seen that in the interactive virtual simulation response time, before the optimization effect, the average response time and the maximum response time value domain is [1.37s, 68.48s], [1.39s, 102.44s], after the edge-folding algorithm in the system of the three-dimensional model simplification effect, so that the redundancy of the system has been released, and its average response time and the maximum response time value domain are [0.62s, 18.76s], [0.77s, 29.23s], demonstrating the difference between the average response time and maximum response time before and after optimization, whose value domains are [0.75s, 49.7s], [0.62s, 73.21s]. In summary, after the edge-folding algorithm to simplify the role of the three-dimensional model in the system, can effectively improve the response efficiency of the interactive virtual simulation system of digital media art.

Table 5: Comparison results of response times for interactive virtual simulation

Number of concurrent users	Before optimization		After optimization		Difference	
	Average response time/s	Maximum response time/s	Average response time/s	Maximum response time/s	Average response time/s	Maximum response time/s
10	1.37	1.39	0.62	0.77	0.75	0.62
20	7.52	13.35	2.01	4.94	5.51	8.41
30	13.88	27.55	6.9	6.35	6.98	21.2
40	20.93	41.45	8.05	8.43	12.88	33.02
50	34.72	59.22	8.92	17.07	25.8	42.15
60	38.88	71.81	13.88	18.55	25	53.26
70	45.58	79.3	15.81	18.42	29.77	60.88
80	54.77	85.83	16.12	24.74	38.65	61.09
90	59.48	94.15	17.63	26.35	41.85	67.8
100	68.46	102.44	18.76	29.23	49.7	73.21

4.2 Memory and CPU usage comparison test before and after optimization

Similarly, the system performance test environment is set unchanged, and the system memory usage comparison test before and after optimization, and the system CPU usage comparison test before and after optimization are conducted to verify the effect of 3D model simplification based on the edge-folding algorithm on the optimization of system response efficiency. The memory and CPU occupation comparison tests before and after optimization are:

4.2.1 Memory usage comparison test before and after optimization

The conditions in the system memory occupation comparison test before and after optimization remain unchanged, where the system memory occupation data before optimization comes from Figure 3, and the system memory occupation data after optimization needs to be collected by re-system testing. The results of the system memory usage comparison before and after optimization are shown in Figure 6. Based on the data performance in Fig. 6, it can be seen that before the optimization, the memory occupation of Edge, Chrome and Firefox browsers is controlled within 500~893MB, and after the simplification of the 3D model in the system by the edge folding algorithm, the memory occupation is controlled within 200MB~500MB, which means that the simplification of the 3D model in the system by the edge folding algorithm can effectively reduce the unified memory occupation, aiming to improve the system performance of digital media. This means that the simplification of the 3D model in the system through the side-folding algorithm can effectively reduce the memory occupation, aiming to improve the performance of the interactive virtual simulation system for digital media art and make it have a better user experience.

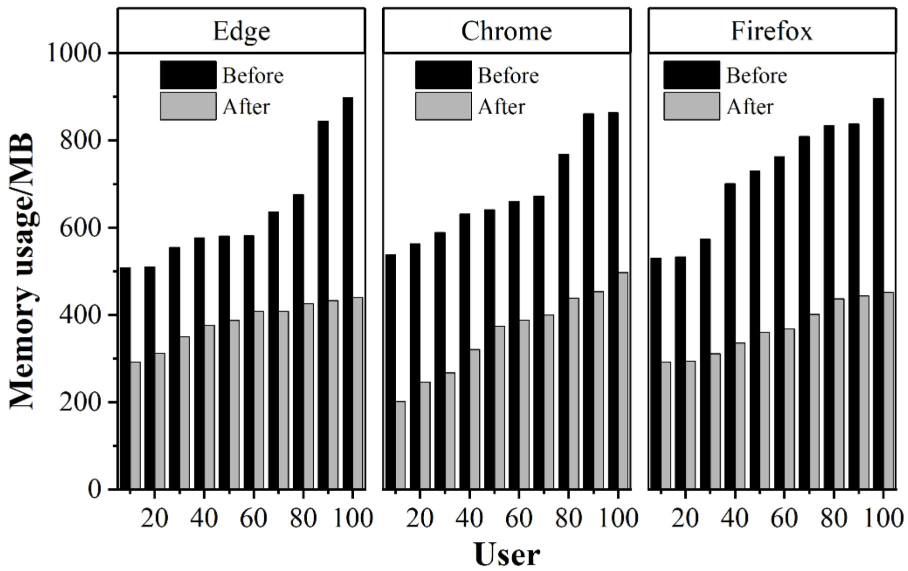

Figure 6: Comparison results of memory usage

4.2.2 Comparative CPU usage test before and after optimization

The conditions in the system CPU occupancy comparison test before and after optimization remain unchanged, where the system memory and CPU occupancy data before optimization are sourced from Figure 4, and the system memory and CPU occupancy data after optimization are re-tested through the system. The results of the comparison of system CPU occupancy before and after optimization are shown in Figure 7. Based on the data performance in Figure 7, it can be seen that before the 3D model simplification based on the edge folding algorithm, the distribution of the CPU occupancy values of running Edge, Chrome and Firefox is between 0.5 and 0.7, and after optimization, it is clearly noticed that the CPU occupancy values have been reduced, and the distribution of the values ranges from 0 to 0.35, which indicates that the 3D model simplification based on the edge folding algorithm has a positive effect on the system CPU occupancy. That is to say, the simplification of 3D model based on the edge folding algorithm has a contributing effect on the CPU occupancy of the system, which greatly improves the phenomenon of unsatisfactory performance of the interactive virtual simulation system of digital media art. In summary, the effect of 3D model simplification based on the edge folding algorithm on the performance optimization of the interactive virtual simulation system for digital media art is fully verified.

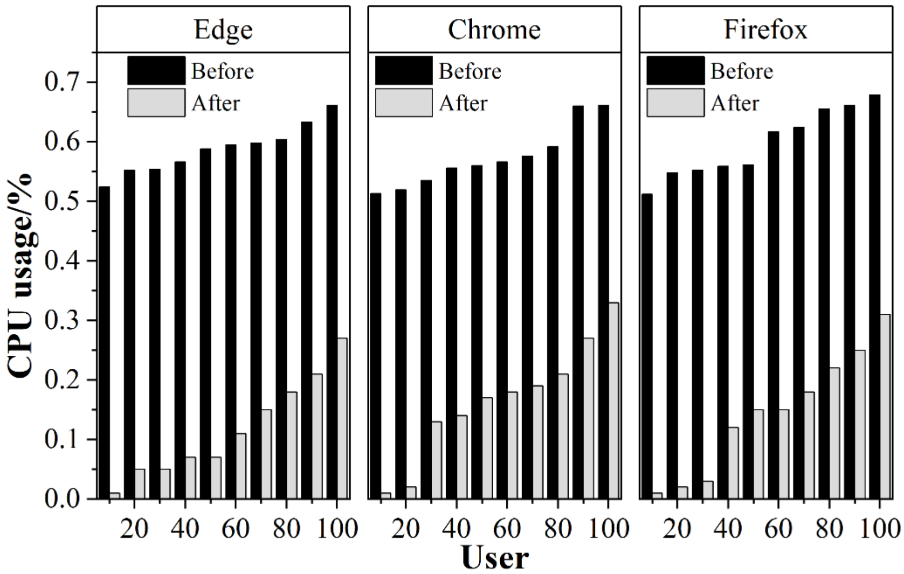

Figure 7: Comparison results of CPU usage

5 Conclusion

With the rapid development of computer technology, the development of computer simulation technology is also becoming more mature, and deeply into the field of digital media art. This paper combines the development needs of digital media art, with the help of Unity3D software to design an interactive virtual simulation system for digital media art, for the system performance is not ideal, proposed the use of side folding algorithm to simplify the three-dimensional model in the system, aiming to achieve the optimization of system performance.

(1) After the edge-folding algorithm's effect on the 3D model simplification in the system, the average response time and maximum response time value domains of the interactive virtual simulation of the system are reduced from [1.37s, 68.48s], [1.39s, 102.44s] to [0.62s, 18.76s], [0.77s, 29.23s], which demonstrates that the simplification of 3D model based on the edge-folding algorithm promotion effect on the system response efficiency.

(2) After the simplification of the 3D model in the system by the edge-folding algorithm, the memory occupation of the system is released to a large extent, and its value is reduced from 500~893MB to 200MB~500MB, which indicates that the simplification of the 3D model based on the edge-folding algorithm is able to reduce the redundancy phenomenon of the system, so as to optimize the performance of the system.

References

[1] Yanqiong, Y. (2024). Analysis of Utilizing Artificial Intelligence to Improve the Efficiency of Digital Media Art Creation. Journal of Artificial Intelligence Practice, 7(4), 101.

[2] Gong, Y. (2021). Application of virtual reality teaching method and artificial intelligence technology in digital media art creation. Ecological Informatics, 63, 101304.

[3] Zheng, R., & An, S. (2023). Digital art design and media practice integrating CAD and virtual reality technology. Computer-Aided Design and Applications, 20(S13), 86-97.

[4] Gui-Wei, B., & Guo-Bao, Z. (2024). Research on the Visual Impact of Digital Media Art Based on Augmented Reality Technology. Comput. Aided. Des. Appl, 21(S2), 186-201.

[5] Li, B., & Lu, W. (2024). Application of image processing technology in the digital media era in the design of integrated materials painting in installation art. Multimedia Tools and Applications, 83(18), 54211-54228.

[6] Gao, Y., & Chen, X. (2021, October). Application of digital media art design in modern advertising under the background of big data. In International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy (pp. 346-352). Cham: Springer International Publishing.

[7] Gintere, I. (2020, May). Art space: An experimental digital art game. In SOCIETY. INTEGRATION. EDUCATION. Proceedings of the International Scientific Conference (Vol. 5, pp. 649-660).

[8] Sun, L. (2022). Research on digital media art film and television special effects technology based on virtual and reality algorithm. Scientific Programming, 2022(1), 4424772.

[9] Yu, W., & Jiang, T. (2021). Research on the direction of innovation and entrepreneurship education reform within the digital media art design major in the digital economy. Frontiers in psychology, 12, 719754.

[10] Zhang, H., & Wu, X. (2024, April). Interactive Virtual Simulation Technology in Artistic Lighting Design System. In 2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT) (pp. 818-822). IEEE.

[11] Ye, W., & Li, Y. (2022). Design and research of digital media art display based on virtual reality and augmented reality. Mobile Information Systems, 2022(1), 6606885.

[12] Piankarnka, V., Lertbumroongchai, K., & Piriyasurawong, P. (2023). A Digital Painting Learning Model Using Mixed-Reality Technology to Develop Practical Skills in Character Design for Animation. Advances in Human-Computer Interaction, 2023(1), 5230762.

[13] Ouyang, S. G., Wang, G., Yao, J. Y., Zhu, G. H. W., Liu, Z. Y., & Feng, C. (2018). A Unity3D-based interactive three-dimensional virtual practice platform for chemical engineering. Computer Applications in Engineering Education, 26(1), 91-100.

[14] Li, S., & Li, J. (2022). Construction of Interactive Virtual Reality Simulation Digital Media System Based on Cross-Media Resources. Computational Intelligence and Neuroscience, 2022(1), 6419128.

[15] Yang, C. (2021). Online art design education system based on 3D virtual simulation technology. Journal of internet technology, 22(6), 1419-1428.

[16] Wei, J. (2025). Design of Virtual Simulation Experiment System Based on Unity3D Engine. Innovative Computing 2025, Volume 3: Proceedings of IC 2025: Innovative System, 1442, 253.