International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 7675-7685

https://doi.org/10.70517/ijhsa464661

Optimization of dual processes for 3D model construction and rendering in film and television special effects

Xiao Zhang[1,*]
[1] College of Movie and Media, Sichuan Normal University, Chengdu, Sichuan, 610066, China
Corresponding authors: (e-mail: izxcn116@163.com).

Abstract With revolutionary breakthroughs in 3D modeling technology, innovative film special effects design and creation have been given a powerful technical boost. To explore the application and optimization of 3D modeling technology in film special effects, this paper proposes a research plan for the application and optimization of 3D modeling technology in film special effects. Starting from the definition of 3D modeling technology, this study utilizes OpenGL and 3ds Max modeling software to achieve solid model construction and texture rendering and coloring. Based on this, video effects editing technology algorithms are employed to optimize film and television special effects. Under the theoretical guidance of this study's research plan, the application and optimization efficacy of 3D modeling technology in film and television special effects are deeply analyzed. Through experimental analysis, it was found that as the number of particles increased from 1,000 to 4,000, the frame rate of film and television special effects did not show a significant decline trend, with frame rate values ranging from 30 fps to 210 fps. This indicates that under the influence of the algorithms proposed in this paper, the frame rate of film and television special effects has been optimized and improved, making the special effects in film and television works highly aligned with user aesthetic standards.

Index Terms OpenGL, 3ds Max, 3D model construction technology, film and television special effects

I. Introduction

As audiences increasingly demand higher-quality films, visual effects have become a key criterion for evaluating the quality of a movie. Excellent visual effects can enhance a film's visual impact, improve the viewing experience for audiences, and attract more viewers and market share [1], [2]. From 3D modeling to visual effects compositing, and from animation design to post-production rendering, the application of computer technology has made special effects design more precise, efficient, and diverse [3], [4]. Various advanced software tools and rendering engines continue to emerge, providing special effects designers with a wealth of technical means and creative platforms [5], [6]. Researching the application of related digital technologies in the field of film and television special effects design can, to a large extent, inject new vitality and momentum into the development of the film and television industry, while also promoting the continuous progress and development of computer graphics and digital media technology [7]-[9].

Three-dimensional model construction technology, as the core of transformation in the film and television special effects industry, has become an indispensable part of modern film and television works [10], [11]. It not only greatly expands the boundaries of creators' imagination but also brings audiences an unprecedented visual feast [12]. From grand cosmic scenes to intricate facial expressions, the application of three-dimensional model construction technology has elevated content creation in films, television, and other visual media to new heights [13], [14]. Additionally, 3D modeling technology offers an economical and efficient solution for film and television works, particularly in scenarios requiring large-scale scene reconstruction or difficult-to-achieve special effects [15]. This flexibility not only reduces costs but also enhances production efficiency, enabling filmmakers to allocate more resources and effort toward storytelling and performance [16].

Numerous scholars have explored computer-based special effects production and practice in film and television, aiming to provide references and insights for research and practice in the field of special effects production. Huang, Y. studied the evolution and impact of the integration of 3D modeling technology with special effects in film and television, from early computer-generated imagery narratives to current immersive film experiences, with 3D modeling technology gradually expanding its scope and exerting a significant influence on the film industry [17]. Zhang, J. addressed the multi-frame image requirements of the film and television special effects industry by proposing a computer-generated model based on 3D convolutional neural networks. By capturing the critical spatial and temporal nuances in film and television images, this model generates high-fidelity and visually stunning scene special effects modeling [18]. Ju, Y. and Wei, G. introduced a particle swarm optimization algorithm into the Unity3D engine to model character movement trajectories in film and television special effects, effectively improving the rendering quality of scene modeling in film and television works, resulting in images with high real-time frame rates and realism [19]. Du, N. and Yu, C. designed a post-processing system tailored for film and television special effects image rendering, incorporating common special effects post-processing techniques used during the creation of three-dimensional scene content, significantly improving the production quality and efficiency of film and television special effects [20]. Di, C., et al. developed a lightweight convolutional neural network (LW-CNN) for facial modeling in film and television special effects, which can accurately extract three-dimensional facial feature points, playing a significant role in the film and television special effects industry [21]. Song, Z. demonstrated that 3D modeling technology has had a profound impact on the film and television industry. This technology can quickly create a large number of virtual motion shots, vast and rich environments, and distinctive characters. These impressive visual effects have become an indispensable component of the film and television industry [22].

This paper first uses the NURBS surface modeling method in 3ds Max software to complete the construction of solid models in film and television special effects. To make the realism of the modeled objects better align with user aesthetic standards, Photoshop or existing materials are used to perform texture rendering and coloring on the solid models. Then, the 3ds Max program is converted into an OpenGL program to obtain an OpenGL-loaded model. Finally, based on the existing framework, we introduce film and television special effects editing technology algorithms to design a research approach for film and television special effects based on 3D model construction technology. In an experimental environment, we explore the application and optimization of 3D model construction technology in film and television special effects.

II. Research on film and television special effects based on three-dimensional model construction technology

II. A. Three-dimensional model construction technology

II. A. 1) OpenGL

OpenGL is a professional programming interface for graphics applications, featuring cross-language and cross-platform interface specifications (API) [23], [24]. At its core, OpenGL is a standard that strictly defines the execution behavior and scope of functions, as well as their output value ranges. The implementation of individual functions within the standard is determined by the developers of the OpenGL library. The latest version of the OpenGL standard library is OpenGL 4.6. As a convenient and powerful low-level graphics library, OpenGL serves as a standard for nearly all 3D software and game interfaces. Professional 3D software such as Maya and 3ds Max is designed based on the OpenGL graphics interface, particularly in the 3D rendering pipeline. When combined with high-end graphics cards that support OpenGL, professional 3D software enables image rendering to meet the requirements of specific tasks. OpenGL is essentially a large state machine, with users controlling its operation through various combinations of inputs. Therefore, OpenGL divides the rendering pipeline into two relatively independent processes: one is executed on the client side, providing control functions, stored in memory used by the central processor, integrated into the application, and executed in the main system memory; the other is the server side, providing actual rendering functions, primarily executed by the GPU. The two run asynchronously, with the client side providing rendering commands and graphic data, and the server side executing rendering tasks. The graphics rendering pipeline can be divided into several stages, each of which is the result of the previous stage. These stages are highly specialized and have parallel processing characteristics. Each graphics card that calls the OpenGL graphics library has tens of thousands of small core processors, which run their own microprograms for each stage on the GPU and quickly process the graphics rendering process. This microprogram is the shader. Shaders are written in the shader language (GLSL). Figure 1 illustrates the principle of shaders. Since there are no default vertex and fragment shaders on the GPU, at least one vertex shader and one fragment shader must be defined (Figure 1). Typically, the OpenGL graphics library provides shader standards for graphics input and output for Maya and 3ds Max, which are professional programming software for the OpenGL graphics interface. They are standard programming interface software, so a good OpenGL graphics card can work well with Maya to provide a solid foundation for video effects production.

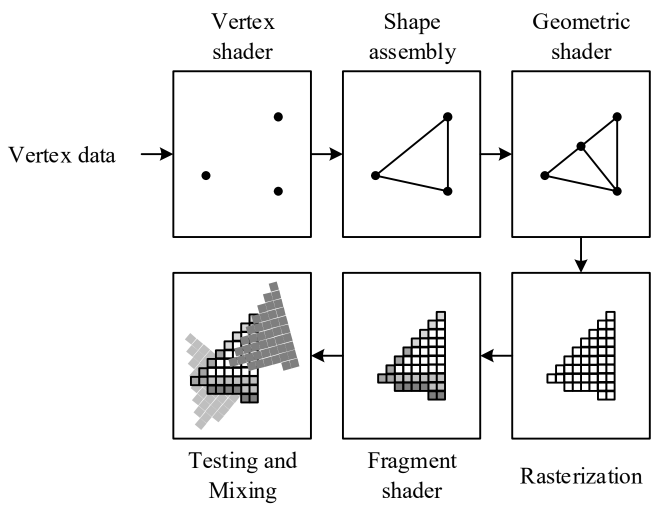

Figure 1: Schematic diagram of the shader principle

II. A. 2) 3dsMax

3D Studio Max, commonly abbreviated as 3dsMax or MAX, is an early 3D software developed by Autodesk for PC-based 3D animation rendering and production [25], [26]. Its predecessor was the 3D Studio series of software based on the DOS operating system, with the latest version being 2024. Before the advent of Windows NT, industrial-grade CG production was dominated by SGI graphics workstations. The combination of 3D Studio Max and Windows NT significantly lowered the barrier to entry for CG production, initially being adopted for animation creation in computer games, and later expanding into special effects production for films such as X-Men II and The Last Samurai. In terms of application scope, 3ds Max and Maya, both 3D software products under the Autodesk umbrella, share similar functionalities but target different domains. Maya excels in character animation and film special effects, while 3ds Max is more adept at architectural animation, game production, and television program packaging. However, in China, most professionals are more familiar with 3ds Max and widely apply it in fields such as advertising, film, industrial design, architectural design, multimedia production, games, educational assistance, and engineering visualization. 3DMAX is currently the most widely used software in the fields of animation production and special effects creation. The software originated from simple drawing tools and, through multiple upgrades and iterations, has evolved to offer powerful three-dimensional spatial motion visualization capabilities. Additionally, it can reproduce real-world scenes and add special effects, enabling more cinematic concepts and ideas to be realized, breaking through previous technical limitations, enhancing the viewing experience of film works, and elevating the overall production standards of the film industry.

II. B.Model Design

Strictly speaking, OpenGL is a 3D graphics library. OpenGL, which stands for Open Graphics Library, is a professional graphics programming interface API that can be used across different platforms and programming environments. It is a powerful library that supports bottom-up rendering and modeling capabilities. Although it contains hundreds of commands and graphics functions, the models it creates are built by specifying the coordinates of points, lines, and polygons, and it does not provide complex modeling commands. If the model being created is complex, using OpenGL for modeling is extremely labor-intensive and not very intuitive. Typically, models are constructed using other modeling software and then converted into the required model data, significantly reducing the difficulty and workload of building complex models. Therefore, this paper uses third-party software 3ds Max for solid modeling and rendering, then converts it into an OpenGL program to obtain an OpenGL-loadable model, and finally incorporates film and television special effects editing techniques and algorithms to complete the three-dimensional model design task for film and television special effects.

II. B. 1) Solid modeling based on 3ds Max

To create 3D models for film and television special effects, the first step is to model the required objects. OpenGL is not well-suited for constructing complex 3D objects. Therefore, this paper utilizes the third-party software 3ds Max for object modeling. 3ds Max offers various modeling methods, such as polygon-based modeling, patch-based modeling, and NURBS surface-based modeling. Considering the high standards required for object models in film and television special effects, this paper adopts the NURBS surface-based modeling method. The NURBS modeling method uses mathematical formulas to control curves and surfaces. Compared to other modeling techniques, NURBS uses relatively fewer control points to represent similar curves. However, since the representation of surfaces is controlled by surface algorithms, it requires higher hardware specifications. The most significant feature of NURBS is its ability to control curves with extreme precision without altering the surface, thereby maximizing the realism of film and television special effects. First, create a cube-shaped surface to roughly obtain the desired model outline, then make corresponding adjustments to achieve the desired model. To enhance the realism of the modeled object, textures are often added to the initial model for rendering purposes. Additionally, textures can be sourced or designed using Photoshop or existing assets to match the model's style.

II. B. 2) OBJ format

OBJ files are 3D model files commonly used for transferring models between different software applications. OBJ files can be opened directly in Notepad, allowing users to read and modify the file contents. The file contains essential model data such as vertex coordinates, normal coordinates, texture coordinates, and polygon information. OBJ files can include lines, polygon faces, and curves. After texture maps are organized in modeling software, texture map information is also saved into the file during the model data export process. OBJ format files can store faces with four or more vertices, while most other model data files can only contain faces with three vertices.

The content and data in an OBJ file are typically preceded by a “#” comment. The definition format for the various data types in an OBJ file is as follows:

(1) Vertex list: v 1.0000 1.0000-1.0000. The position of a vertex is directly specified by its three-dimensional coordinate values. “v” denotes a vertex, and the values following “v” represent the specific coordinate values of that vertex. Each vertex's coordinates must be on a separate line, with each line starting with “v.”

(2) Texture coordinates, vt 1.0000 0.0000 1.0000. The format for texture coordinates is the same as for vertex coordinates, with each line starting with “vt” followed by the specific coordinate values.

(3) Normal vectors, vn 1.0000 1.0000 1.0000. Normal vector coordinates begin with “vn,” followed by the specific coordinate values.

(4) Faces, f 1/1/1. Faces are defined using indices for vertices, textures, and normal vectors. There are various ways to define faces depending on specific requirements, but each line begins with “f”.

II. B. 3) Loading models with OpenGL

After the initial entity model is completed, an OBJ file is exported. Meanwhile, data related to lighting, materials, and other aspects of the model is stored in an MTL file. OpenGL programming loads the OBJ data. Based on the types of data included in the OBJ format, encountering “#” indicates a comment, which is neither read nor stored. Encountering “v” indicates vertex coordinates, which are stored in the vertex coordinate array. When encountering “vt,” which represents texture coordinates, the data is stored in the texture coordinate array. When encountering “vn,” which represents normal coordinates, the data is stored in the normal coordinate array. When encountering “g,” which represents groups, the data is stored in the group structure. If the beginning of a line is “f,” the index type must be determined. The index can be a vertex, vertex/texture coordinates, vertex/texture coordinates/normal vector, or vertex//normal vector. Depending on the index type, the data is stored in the corresponding array. The same method is used to read MTL files. A single OBJ data file can use multiple material files, ultimately resulting in an OpenGL-loadable model.

II. B. 4) Introduction of video special effects editing technology algorithms

The key technical points in the design of 3D models for film and television special effects primarily involve special effects processing algorithms, including linear transformations, nonlinear transformations, scaling, blurring, mirroring, fade-in/fade-out, page turning, and venetian blinds. The key functions for implementing special effects are provided to control the transformation rules for each frame of the video, thereby achieving different special effects. As shown below:

(1) Linear transformation

For example, in a single frame of an image, the brightness values range from $[n_1, n_2]$. To transform this range to $[m_1, m_2]$, a linear transformation function can be used. Let the brightness value before transformation be x and the brightness value after transformation be y; then the following expression can be obtained:

$$y = \frac{m_2 - m_1}{n_2 - n_1}(x - n_1) + m_1 \tag{1}$$

Simplified version:

$$y = a \cdot x + b \tag{2}$$

In the formula: $a = \frac{m_2 - m_1}{n_2 - n_1}, b = m_1 - \frac{n_1(m_2 - m_1)}{n_2 - n_1}$.

When performing a linear transformation, substitute the brightness values of each pixel into the above formula to obtain new brightness values. Replace the original brightness values with the new values to obtain the transformed image.

By altering the shape of the transformation line, i.e., adjusting the values of the four parameters n_1, n_2, m_1, m_2, different transformation effects can be produced. If the value of a is greater than 1, the brightness range is expanded, and the final image appears brighter; conversely, if the value of a is less than 1, the brightness range is reduced, and the final image appears darker. When using this method, you can adjust the values of each parameter as needed to achieve a darker or brighter display effect for the image.

(2) Non-linear transformation

Transformation using an exponential function involves expanding the brightness interval in high-brightness areas, which is a stretching operation, causing this part of the image to appear brighter after transformation. In low-brightness areas, the brightness interval is reduced, which is a compression operation, causing this part of the image to appear darker after transformation. The formula is expressed as:

$$y = b \cdot e^{ax} + c \tag{3}$$

In the formula, the three parameters a, b, and c can be adjusted separately to change the shape of the exponential function curve, thereby achieving different transformation ratios and different image transformation effects.

When using a logarithmic function for transformation, the opposite of exponential function transformation occurs, meaning that the low brightness value area is stretched while the high brightness value area is compressed. The formula is expressed as:

$$y = b \cdot \ln(ax + 1) + c \tag{4}$$

The three parameters a, b, and c in the formula can all be adjusted. Users can adjust the parameter values according to their needs.

(3) Scaling processing

Scaling processing refers to enlarging or reducing the size of a video image by a certain ratio. Assuming that the scaling factor is s, the transformation formula is:

$$[x \quad y \quad 1] = [u \quad v \quad 1]\begin{bmatrix} s_u & 0 & 0 \\ 0 & s_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The above formula expands as follows:

$$\begin{cases} x = u \cdot s_u \\ y = v \cdot s_v \end{cases} \tag{6}$$

In the formula, (x, y) represents the coordinate position of the pixel in the transformed image, (u, v) represents the coordinate position before transformation, S_u is the scaling ratio in the u direction, and S_v is the scaling ratio in the v direction. When the scaling ratios have different values, they produce different scaling transformation effects:

1) When S_u, S_v are greater than 1, the image is enlarged.

2) When S_u, S_v are greater than 0 and less than 1, the image is reduced in size.

3) When S_u, S_v are less than 0, the image is not only scaled but also inverted.

4) When $S_u = 1$ and $S_v = -1$, the image undergoes vertical mirroring.

5) When $S_u = -1$ and $S_v = 1$, the image undergoes horizontal mirroring.

(4) Blur Effects

Blur processing is primarily used to remove noise from video images. “Noise” refers to smaller pixel areas compared to the filter mask. In nonlinear video editing, blur effects include Gaussian blur and smooth blur, with the corresponding filters as follows:

$$\frac{1}{16}\cdot\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} \text{ and } \frac{1}{9}\cdot\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \tag{7}$$

The smoothing filter produces the average value of standard pixels, which can be simplified as follows:

$$y=\frac{1}{9}\sum_{i=1}^{9}x_i \tag{8}$$

In the formula: x_i represents the pixel values surrounding pixel point x and within the filter mask area.

When calculating Gaussian blur, each pixel is multiplied by a weighting system. Pixels with larger weight values have a relatively higher influence, and the weight values of pixels at the center are typically greater than those of surrounding pixels. The formula for the weighted average is as follows:

$$y=\frac{1}{9}\sum_{i=1}^{9}a_i\cdot x_i \tag{9}$$

In the formula: a_i is the mask weight coefficient, and x_i is the pixel value of the pixel point x and within the filter mask area.

(5) Fade in/fade out

Generally speaking, when fading in, the image is slowly and uniformly displayed on the screen from the background color of the screen; and during fade-out, the image gradually disappears from the screen at a constant speed. This is the fade-in/fade-out form for a single video, primarily using the screen's background color as a reference point. During multi-segment video editing, the images between multiple video segments are typically used as reference points. Thus, during the fade-in/fade-out process, the effect manifests as one frame of the image gradually transitioning into another frame. The mathematical expression for the fade-in/fade-out effect is as follows:

$$f(x,y)=\begin{cases}f_1(x,y) & 0<K\le N_1\\ r\cdot f_2(x,y)+(1-r)\cdot f_1(x,y) & N_1<K\le N_1+N\\ f_2(x,y) & (N_1+N)<K\le(N_1+N+N_2)\end{cases} \tag{10}$$

$$r=\frac{K-N_1}{N} \tag{11}$$

In the formula: N_1 represents the number of frames in the video sequence $f_1(x,y)$ that is to be faded out, N_2 represents the number of frames in the video sequence $f_2(x,y)$ that is to be faded in, and N represents the total number of video frames involved in the transformation. K is the current frame, and r is the transparency channel value of the Kth frame. During the transformation process, the r value is used to control the display weight ratio of each frame, thereby achieving the fade-in/fade-out effect.

(6) Page-turning effect

The implementation principle of the page-turning effect is shown in Figure 2. Taking the page-turning effect in the top-left corner as an example, the pixels of the right-angled triangle aef are filled with the pixel values of the incoming video sequence, and their size expands along the direction of the dashed line until the entire screen is filled; the pixels of the triangle efg are filled with the background color; the remaining parts are filled with the pixel values of the outgoing video sequence. Page-turning effects in other directions can be achieved by adjusting the parameters.

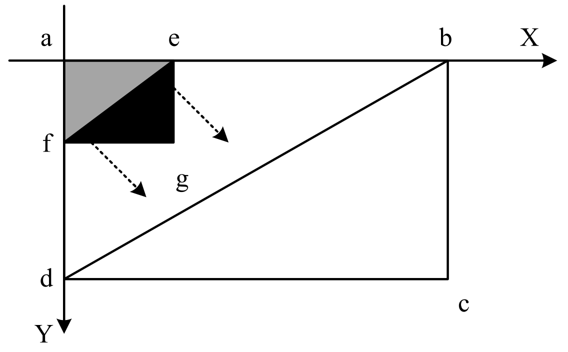

Figure 2: Schematic diagram of page-turning effect

The function formula for the page-turning effect is as follows:

$$f(x,y)=\begin{cases} f_2(x,y) & 0 \le (x+y) < r \\ f_1(2\times r-y,2\times r-x) & (x+y)\ge r, x\le r, y\le r \\ f_1(x,y) & other \end{cases} \tag{12}$$

(7) Venetian blind special effect

The Venetian blind special effect refers to cutting a frame of an image into several small fragments, each of which moves along a certain trajectory and then returns to its original position after the movement is completed, restoring the image to its original state.

The motion trajectory function is typically chosen as a periodic function, as the motion trajectory of a venetian blind is a periodic motion. Here, a sine function is selected as the motion trajectory function. Its function expression is as follows:

$$[x,y,1]=[u,v,1]\begin{bmatrix} 1-\sin(r\cdot t) & 0 & 0 \\ 0 & 1-\sin(r\cdot t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Simplified version:

$$\begin{cases} x = u\cdot(1-\sin(r\cdot t)) \\ y = v\cdot(1-\sin(r\cdot t)) \end{cases} \tag{14}$$

In the formula: parameter r represents the random factor of the k th fragment, and parameter t represents the time used, whose value range is in $[0,\pi]$.

III. Model validation analysis

III. A. Solid modeling and texture rendering analysis based on 3dsMax

III. A. 1) Solid modeling analysis based on 3dsMax

To validate the NURBS surface modeling method in 3dsMax software, polygon modeling and patch modeling methods were set as control methods. Modeling time and contour error were used as the primary evaluation criteria, with numerical data sourced from the 3dsMax software. A comparative analysis of the evaluation metric data for different modeling methods was conducted, with 100 experimental runs to ensure the reliability of the research results. The solid modeling analysis results based on 3dsMax are shown in Figures 3–4. In the figure, X1, X2, and X3 represent the NURBS surface modeling method, polygon modeling method, and patch modeling method, respectively. The points represent the evaluation metric values, with a total of 300 points. The values in the figure indicate the statistical mean of the evaluation metric values. Based on the data sizes in Figures 3–4, the average modeling times for the surface modeling method, polygon modeling method, and patch modeling method are 10.02 seconds, 10.99 seconds, and 11.87 seconds, respectively. with corresponding contour errors of 0.01 cm, 0.0129 cm, and 0.0141 cm, respectively. This indicates that the NURBS surface modeling method has a priority in solid modeling, while also reflecting its optimization effects in terms of modeling time and contour error.

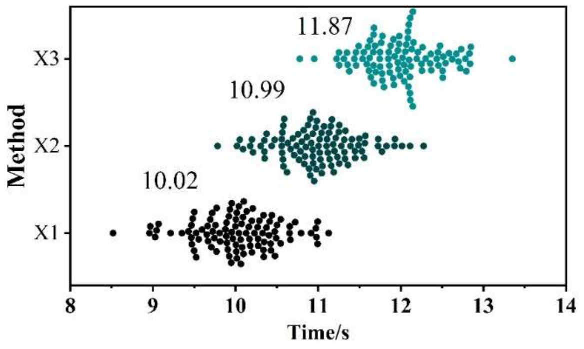

Figure 3: Comparative analysis of the time consumed in modeling

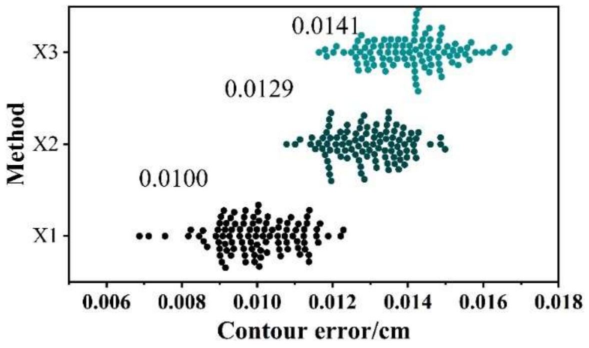

Figure 4: Contour error comparison results

III. A. 2) Texture rendering analysis based on 3dsMax

Texture rendering is related to modeling software. The testing tool used in this design is a mobile phone, whose GPU, when only performing rendering without other computations, can render approximately 30,000 to 50,000 triangles per frame at a frame rate of over 30 frames per second (frame rate refers to the number of frames rendered per second; typically, a frame rate above 30 fps results in smoother visuals). When loading a batch of objects into a film special effects scene and distributing them relatively evenly across the scene, the traditional approach is to send all objects into the rendering pipeline. By comparing the traditional texture rendering method (3ds Max) with the texture rendering method proposed in this paper (3ds Max + Photoshop), the rendering efficiency under both scenarios is shown in Figure 5, where A and B represent the traditional method and the method proposed in this paper, respectively. It can be seen that the number of triangular faces in entities within a film special effects scene significantly impacts rendering quality. However, current film special effects scenes contain an extremely large number of triangular faces, and the rendering efficiency of the traditional texture rendering method decreases as the number of triangular faces increases. In contrast, the texture rendering method proposed in this paper continuously renders entities within a fixed range, thereby reducing the rendering of unnecessary entities. As shown in the experimental results, the texture rendering method (40–60 fps) achieves a noticeable improvement in rendering efficiency compared to the traditional method (30–50 fps), demonstrating the optimization effectiveness of the proposed texture rendering approach.

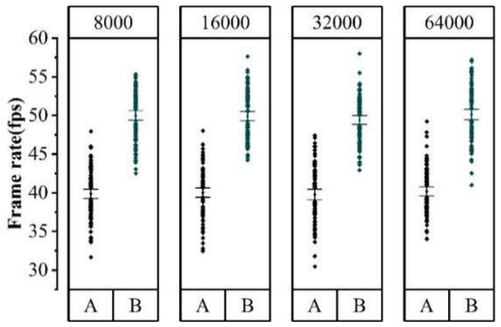

Figure 5: Comparison results of texture rendering efficiency

III. B. OpenGL model loading verification analysis

III. B. 1) Experimental Environment

The computer model used in the experimental platform is a MacPo Intel Xeon 5300 processor with 4GB DDR2 ECC memory and Mac OS X v10.4.4 Tiger operating system. The graphics acceleration card is an NVDIA GeForce 7800 GT 512MB GDDR2SDRAM, and the OpenGL engine uses NVDIA OpenGL hardware extensions and Apple OpenGL software implementation. The experimental program is based on Cocoa and written in Objective-C. It utilizes the OpenGL and Quartz Core framework programming interfaces provided by Mac OS X Tiger, with compilation and execution performed in the Xcode 2.8 development environment. The graphical system performance monitoring tool OpenGL Driver Monitor and the analysis tool OpenGL Profiler from the Xcode debugging toolkit are used to track program flow and perform real-time sampling of the graphics acceleration card's operational status.

III. B. 2) Data Analysis

To evaluate the performance of the OpenGL loading model in film and television special effects applications, this paper uses the OpenGL Driver Monitor tool to test the shading process of the OpenGL loading model through the hardware driver layer, monitoring the working parameters of the GPU and display buffer of the graphics acceleration card. Figure 6 illustrates that the filter chain based on delayed assignment and RO1 sampling only delivers GPU pixel shader calculations when refreshing the buffer, resulting in a peak in processing time. The Core Image engine processes the image source input kernel on the leading edge of the peak, while the pixel shader processes the pixels and outputs them to the buffer on the trailing edge of the peak. The OpenGL loading model accepts texture data output from 3dsMax, adds thread locks to protect critical sections, and has a cycle width equal to the GPU processing cycle width. However, the OpenGL loading model consumes less time.

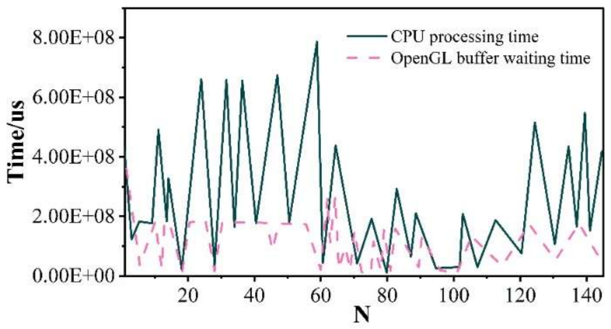

Figure 6: Sampling of working parameters of GPU and display buffer

Next, the OpenGL Driver Monitor tool was used to collect statistics and analyze the execution time of the OpenGL loading model during the film and television special effects design process. The results of the execution time analysis are shown in Table 1. The data shows that the execution time of the OpenGL loading model is less than 1 second, which fully meets the requirements of film and television special effects design.

Table 1: Execution time analysis results

Operation by OpenGL	Number of calls	Total time/us	Average time/us
Coloring operation	29242	138467	58.26
Coordinate transformation	24247	20293	13.15
Parameter Settings	27529	54114	7949.39
State control	43918	622738	346.44
Statistics	124936	835612	8367.24

III. C. Analysis of algorithm application examples

This section will examine the application and optimization of the algorithm described in this paper in film and television special effects from the perspectives of snowflakes and raindrops. The analysis process is as follows:

III. C. 1) Snow Scene Simulation

This section conducted multiple experiments simulating snowy landscape effects, with several representative results. Using a typical forest as the background, the experiments simulated snowfall scenarios. Table 2 lists the experimental data results for the snowy landscape simulations. When the number of particles is 1,000, the number of snowflakes is very sparse, and they are barely visible. When the number of particles is 2,000, the number of snowflakes is appropriate and meets visual requirements. When the number of particles is 4,000, the number of snowflake particles is sufficient, and the algorithm in this paper can simulate excellent snow scene effects. It also demonstrates the algorithm's optimization effects for film and television special effects, enabling snow scene effects in film and television works to align well with user aesthetic standards.

Table 2: Snow scene special effects simulation

Particle number and	The current number of triangles	Frame rate (FPS)	Visual effect
1000	715	46.541	The snowfall is scarce
2000	1121	40.128	It usually snows and the amount of snow is appropriate
4000	5109	31.357	The snowfall is heavy enough

III. C. 2) Rain Scenario Simulation

To evaluate the effectiveness of the algorithm proposed in this paper in simulating rain droplets under different particle counts, the results of the rain scene simulation analysis are shown in Table 3. When the particle count is 1,000, the number of rain droplets is very low, making them barely visible, and the simulation fails to realistically replicate the scene. When the number of particles is 2,000, the number of raindrops is sufficient to achieve the visual effects of a raining scene. When the number of particles is 4,000, the number of raindrops is abundant, and the algorithm can effectively simulate a raining scene without a significant drop in frame rate, maintaining a moderate speed. This fully demonstrates the application and optimization of the algorithm in film and television special effects, holding significant strategic importance for the development of the film and television industry.

Table 3: Rain scene simulation analysis results

Particle number and	The current number of triangles	Frame rate (FPS)	Visual effect
1000	1826	203.905	There are too few raindrops.
2000	5035	154.176	The rainfall is moderate, meeting the visual requirements
4000	5208	124.238	Heavy rain has a good visual effect

IV. Conclusion

This paper proposes the use of OpenGL and 3ds Max modeling software within the scope of three-dimensional model construction technology to complete the tasks of physical model construction and texture rendering and coloring. Additionally, video effects editing technology algorithms are introduced to ultimately design a research plan for the application and optimization of film and television effects based on three-dimensional model construction technology. First, an experimental environment is established to conduct instance verification and analysis of the research scheme proposed in this paper. When the number of particles increases from 1,000 to 4,000, the algorithm simulates a frame rate of 30fps to 210fps for film and television special effects, all of which are greater than 25fps. This verifies the application and optimization efficiency of the algorithm proposed in this paper in film and television special effects, enabling high-quality development of film and television special effects.

References

[1] Lorditch, E. (2019). Turning science to movie magic. Physics World, 32(11), 40.

[2] Murodillayev, B. (2024). The Impact of Visual Effects on the Cinema Experience: A Comprehensive Analysis. Art and Design Review, 12(4), 238-249.

[3] 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.

[4] Zhang, J. (2021). Application analysis of special effects technology in film and television post-production. In Frontier Computing: Proceedings of FC 2020 (pp. 1007-1013). Springer Singapore.

[5] Yang, Z. (2025). Method for Rendering Large-Scale Movie Late-Stage Special Effects Under Mixed Use of Level of Detail Fusion. International Journal of High Speed Electronics and Systems, 34(03), 2540007.

[6] Christensen, P., Fong, J., Shade, J., Wooten, W., Schubert, B., Kensler, A., ... & Liani, M. (2018). Renderman: An advanced path-tracing architecture for movie rendering. ACM Transactions on Graphics (TOG), 37(3), 1-21.

[7] Cheng, Y., & Wang, Y. (2022). Movie Special Effects Processing Based on Computer Imaging Technology. Scientific Programming, 2022(1), 1384589.

[8] Wan, X. (2024). The potential of virtual production based on the special effects of films. In SHS Web of Conferences (Vol. 193, p. 01013). EDP Sciences.

[9] Zhang, W., & Wang, Y. (2020). Analyze the application of multimedia Technology in the Special Effects of film and television animation-from the animated short film of" day after day". International Journal of Social Science and Education Research, 2(11), 10-15.

[10] Harkema, G. J., & Rosendaal, A. (2020). From cinematograph to 3D model: how can virtual reality support film education hands-on?. Early popular visual culture, 18(1), 70-81.
[11] Kuang, R. (2021). Design and implementation of 3D film and television scene production algorithm based on the internet of things. Wireless Communications and Mobile Computing, 2021(1), 1219849.
[12] Yi, Z. (2021, May). Application of Computer Vision in 3D Film. In Journal of Physics: Conference Series (Vol. 1915, No. 2, p. 022026). IOP Publishing.
[13] Hashim, M. E. A., Albakry, N. S., Mustafa, W. A., Grahita, B., Ghani, M. M., Hanafi, H. F., ... & Ugap, C. A. (2024). Understanding the impact of animation technology in virtual reality: A systematic literature review. International Journal of Advanced Research in Computational Thinking and Data Science, 1(1), 53-65.
[14] Li, K. (2024). Application of Communication Technology and Neural Network Technology in Film and Television Creativity and Post-Production. International Journal of Communication Networks and Information Security, 16(1), 228-240.
[15] Swords, J., & Willment, N. (2024). ‘It used to be fix-it in post production! now it’s fix-it in pre-production’: how virtual production is changing production networks in film and television. Creative Industries Journal, 1-17.
[16] Cui, L., Zhang, Z., Wang, J., & Meng, Z. (2022). Film effect optimization by deep learning and virtual reality technology in new media environment. Computational intelligence and neuroscience, 2022(1), 8918073.
[17] Huang, Y. (2024). 3D special effects modelling based on computer graphics technology. Applied and Computational Engineering, 50, 106-112.
[18] Zhang, J. (2024). Visual Communication Method of Multi frame Film and Television Special Effects Images Based on Deep Learning. Scalable Computing: Practice and Experience, 25(6), 5460-5468.
[19] Ju, Y., & Wei, G. (2024). Film and Television Special Effects AI System Integrating Computer Artificial Intelligence and Big Data Technology. Scalable Computing: Practice and Experience, 25(4), 2532-2539.
[20] Du, N., & Yu, C. (2020, October). Research on special effects of film and television movies based on computer virtual production VR technology. In Proceedings of the 2020 International Conference on Computers, Information Processing and Advanced Education (pp. 115-120).
[21] Di, C., Peng, J., Di, Y., & Wu, S. (2021). 3D face modeling algorithm for film and television animation based on lightweight convolutional neural network. Complexity, 2021(1), 6752120.
[22] Song, Z. (2021). Research on Post Production of Film and Television Based on Computer 3D Technology. In 2020 International Conference on Data Processing Techniques and Applications for Cyber-Physical Systems: DPTA 2020 (pp. 1557-1561). Springer Singapore.
[23] S. A. Zolotarev & A. T. Taruat. (2024). Parallel Image Reconstruction Using the Maximum Likelihood Method with a Graphics Processor and the OpenGL Library. Russian Journal of Nondestructive Testing,60(6),648-657.
[24] Lei Xia & Ran Yan. (2025). A Fast Slicing Method for Colored Models Based on Colored Triangular Prism and OpenGL. Micromachines,16(2),199-199.
[25] Xing Yue. (2025). 3D digital reconstruction and application of traditional ceramic design works based on 3dMAX virtual simulation technology. Journal of Computational Methods in Sciences and Engineering,25(1),918-930.
[26] Neo Ping Xi & Loh Yoke Ling. (2022). Malaysian Film and Visual Effects Industry: Challenges and Future. Quarterly Review of Film and Video,39(2),308-322.