

Research and Application of Caideng Model Rendering Technology for Virtual Reality

Xuefeng Wang¹, Yadong Wu², Yan Luo¹, Dan Luo¹

¹School of Automation and Information Engineering, Sichuan University of Science & Engineering, Yibin, China

²College of Computer Science and Engineering, Sichuan University of Science & Engineering, Yibin, China

Email: 1060076653@qq.com

How to cite this paper: Wang, X.F., Wu, Y.D., Luo, Y. and Luo, D. (2024) Research and Application of Caideng Model Rendering Technology for Virtual Reality. *Journal of Computer and Communications*, 12, 95-110.
<https://doi.org/10.4236/jcc.2024.124009>

Received: March 18, 2024

Accepted: April 19, 2024

Published: April 22, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

With the development of virtual reality (VR) technology, more and more industries are beginning to integrate with VR technology. In response to the problem of not being able to directly render the lighting effect of Caideng in digital Caideng scenes, this article analyzes the lighting model. It combines it with the lighting effect of Caideng scenes to design an optimized lighting model algorithm that fuses the bidirectional transmission distribution function (BTDF) model. This algorithm can efficiently render the lighting effect of Caideng models in a virtual environment. And using image optimization processing methods, the immersive experience effect on the VR is enhanced. Finally, a Caideng roaming interactive system was designed based on this method. The results show that the frame rate of the system is stable during operation, maintained above 60 fps, and has a good immersive experience.

Keywords

Virtual Reality, Caideng Model, Lighting Model, Point Light Rendering

1. Introduction

Caideng is China's national intangible cultural heritage. With the development of information technology, virtual reality (VR) technology has provided new ideas for the digital transformation and upgrading of the traditional Caideng industry. Caideng industry and digital technology begin to combine to improve the expression form of Caideng [1]. Wang Jiaqi *et al.* using Unity 3D engine and SteamVR2.0, combined with the interaction between handle ray and UGUI and the immersive cruise function and realized the immersive interactive experience system of making Caideng and the custom experience of Qinhuai Lantern Festival [2]. Wang Zhigang *et al.* used a light and shadow art installation in the

Lantern Festival, which provides a useful reference for the development of folk art in the new era [3]. Li *et al.* implemented a plugin-free virtual roaming based on WebGL, and comprehensively utilized technologies such as 3D MAX and Three.js to construct a lightweight immersive roaming scene with good interactivity and scalability [4]. Gao Yidong *et al.* developed a virtual library roaming system using the UDK (Unreal Development Kit) engine and 3ds Max and proposed a feasible solution for developing a highly realistic and interactive 3D virtual library roaming system [5]. Yang Xiaobo and others used 3D laser scanning technology and VR Platform design to achieve a 3D campus virtual roaming system [6]. Due to the particularity of lighting scenes, these methods cannot be well applied to the roaming of lighting scenes. How to achieve roaming of Caideng scenes in virtual reality environments is currently a research focus in the Caideng industry.

At present, there are few studies on the night scene rendering of the Caideng model. To move the Caideng scene to the Internet to realize digital cultural travel, it is necessary to focus on the rendering effect of the Caideng model. To this, this paper for lights model, a 3D model automatically generated, combined with scene light source characteristics, through the lighting model algorithm, implements the simulation effect in Unity, and then for display equipment to realize the real luminous effect, finally design and set up the virtual Caideng roaming scene, with strong immersion and interactive experience.

2. System Design

The production process of the Caideng roaming system includes the Caideng model modeling, the Caideng rendering effect, the Caideng scene design, and the interactive function design. The overall process is shown in **Figure 1**. In the Caideng model modeling stage, UAVs are used to take aerial photographs of the modeled objects or scenes to obtain a series of images or point cloud data. 3D reconstruction software is used to create high-precision 3D models from multiple images or laser scan data. Support for 3D reconstruction from images with different angles, resolutions, and exposure conditions, and can automatically align, match, and fuse these data to generate a complete 3D scene model. In the Caideng rendering effect stage, ShaderLab language is used to simulate the lighting effect of rendering Caideng by combining light rendering algorithms. In the Caideng scene design stage, use Unity software to set up the scene environment, including skybox, light rendering, terrain, and collision bodies. In the

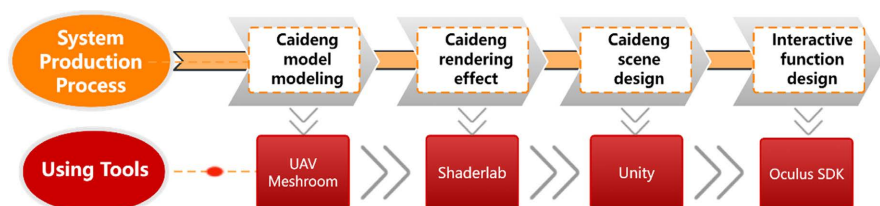


Figure 1. The production process of the Caideng roaming system.

Interactive function design stage, the device used in this system is the Meta Quest PRO, so the interaction design is developed through the Oculus SDK provided by Meta.

3. Implementation

3.1. Caideng Model Modeling

Use a UVA to take aerial photography of modeled objects or scenes, resulting in a series of images or point cloud data. Meshroom is a 3D reconstruction software that can create high-precision 3D models from multiple image data [7]. The node architecture of Meshroom allows users to customize different pipelines to suit their specific needs. The modeling process is shown in **Figure 2**.

Feature extraction forms the basis of finding the relative pose of the camera in space by detecting the feature points in each image, which can be described by scaling the invariant properties, rotation in 2D and 3D, and several other attributes. In principle, to find the corresponding points between the images, the features of each pair of images must be compared and matched. In this step, the feature matching between the image pairs will subsequently allow for the determination of the 3D structure based on the corresponding 2D position. In the following, Structure from Motion (SfM) step, the calculated matches are fused into the trajectory: each trajectory is a candidate for a point in space, visible from multiple cameras. These traces are used to solve the 3D structure of the camera calibration and scene, thereby generating a sparse 3D representation. This step attempts to estimate the depth value for each input pixel. The area requires at least 2 SfM verified cameras to see. To create dense 3D surfaces, the meshing step merges all depth maps into a single dense point cloud, in which the surfaces are then extracted. For each input image, select the most interesting depth values and back-project them as 3D points. The final step of the 3D reconstruction pipeline projects the texture onto the grid.

3.2. Caideng Rendering Effect

The rendering process steps of the Caideng model are shown in **Figure 3**. First, the vertex data and texture maps of the model are obtained from the already-built Caideng model for subsequent lighting calculation. Second, the Lambert model and Blinn-Phong model are applied to calculate the diffuse and specular terms

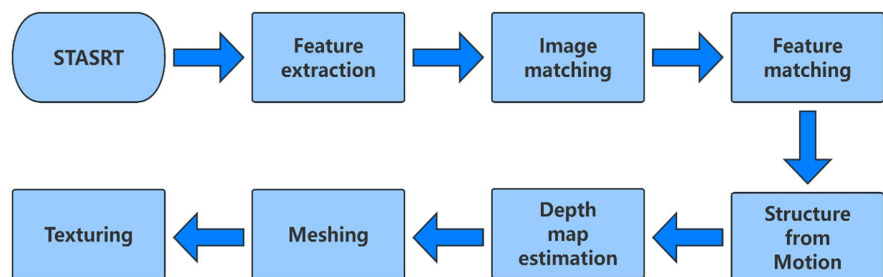


Figure 2. Caideng model modeling process.

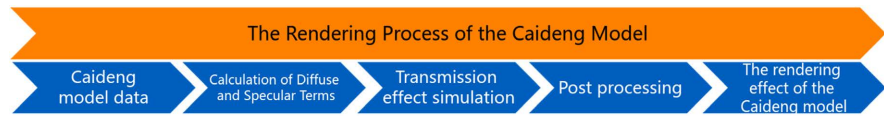


Figure 3. The rendering process of the Caideng model.

respectively, and the two calculation results are added up. Then, the BTDF transmission formula is referenced and adjusted according to the characteristics of the Caideng emission to achieve the effect of the Caideng transmission and emission. Finally, the screen post-processing of the Gaussian model is applied to achieve the final effect of the Caideng. In this rendering process, the lighting effect of Caideng under different point light sources can be simulated by adjusting the point light source parameters (such as luminous range, light source intensity, and light source color).

3.2.1. Implementation

In a virtual environment, a 3D model is a collection of points composed of multiple vertices, and the model requires a lighting model. The lighting model describes the quantified relationship between the incident and outgoing light energy on the object's surface. Traditional lighting models can be divided into two kinds: local lighting models and global lighting models [8]. Within the range of local lighting models, typical lighting models include the Lambert diffuse reflection model, Phong mirror reflection model, Blinn-Phong model, etc. Generally, the local lighting model only calculates the illumination emitted by the direct light source in the scene, and only calculates the ambient light, diffuse reflection, and mirror light, so the calculation amount is small, which is very suitable for real-time operation.

When rendering images in the Caideng virtual roaming system, the following physical phenomena need to be considered: 1) the type of light source; 2) the light emitted enters the human eye after interacting with the object; 3) the final screen effect presented by the camera. The calculation of local lighting models usually includes ambient light, diffuse light on the surface of the object, mirror-reflected light on the surface of the object, and self-illumination on the object's surface. The color of the vertices on the surface is composed of these lighting effects, as shown in Equation (1):

$$I_c = I_a + I_d + I_s + I_{em}. \quad (1)$$

where I_c is the color of the vertices on the surface of the object, I_a is ambient light, I_d is the diffuse reflection light on the surface of the object, I_s is the specular reflection light on the surface of the object, and I_{em} is the self-illumination of the surface of the object.

Lambert lighting model is an ideal diffuse reflection model, proposed by Lambert [9]. The model assumes that when a light source hits the surface of an object, the intensity of the reflected light will be uniformly reflected around, and its light intensity is proportional to the cosine of the direction of the incident

light and the angle of the surface normal. As shown in **Figure 4**, the intensity of the reflected light is proportional to the cosine of the Angle between the surface normal and the direction of the light source. Therefore, the diffuse reflection part is calculated as follows:

$$I_d = k_d * I * \cos \theta . \quad (2)$$

In Equation (2), k_d represents the diffuse reflection properties of the object surface, and I indicates the light intensity of the incident light. In **Figure 4**, N is the normal, L is the unit vector of the incident light, and θ is the angle of N and L , then $\cos \theta = N \cdot L$.

Phong lighting model only includes the diffuse reflection of ambient light and the diffuse reflection and mirror reflection [10], so the light intensity reflected by the mirror is only related to the direction of the incident light and the observation Angle of the user. In **Figure 5**, the Phong model formula is as follows:

$$I_s = k_s * I * (\cos \alpha)^n . \quad (3)$$

In Equation (3), I is the incident light, k_s is the high light coefficient of the object surface, α is the angle of the reflected light and the user perspective, and n is the high light index. The smoother the surface, the greater the n value, the smaller the high light range, and the stronger the high light intensity. As shown in **Figure 5**. V is the unit vector from the vertex to the line of sight, and R is the direction of the reflected light, then $\cos \alpha = V \cdot R$.

The bidirectional scattering distribution function (BSDF) includes both scattering and transmission phenomena [11]. As shown in **Figure 6**. BSDF can decompose two parts: Bidirectional Reflection Distribution Function (BRDF) and Bidirectional Transmission Distribution Function (BTDF), namely $BSDF = BRDF + BTDF$, where BRDF describes the reflection of light, while BTDF describes the transmission of light.

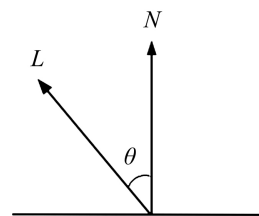


Figure 4. Lambert lighting model.

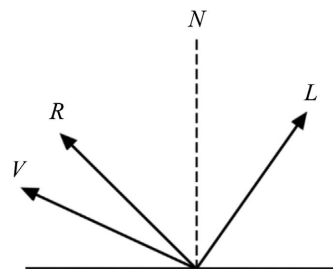


Figure 5. The Phong lighting model.

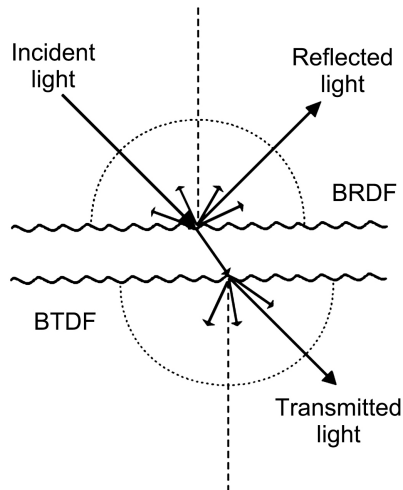


Figure 6. Schematic diagram of the BSDF.

According to the principle of the BTDF model and the luminous characteristics of the lights, the system integrates the algorithm formula with strong effects such as diffuse reflection, high light reflection, and shadow projection, and adjusts the corresponding parameter values, to better simulate the lighting effect of specific lights.

The custom lighting model uses a point light source for rendering, so for the factors in the formula, the Caideng rendering only needs to calculate the diffuse and high light reflection parts and superposition to achieve the final effect.

Diffuse light conforms to Lambert's Law: the intensity of the reflected light is proportional to the cosine of the angle between the surface normal and the source direction. Therefore, the diffuse reflection part is calculated as follows:

$$c_{diffuse} = (c_{light} \cdot m_{diffuse}) \max(0, N \cdot L). \quad (4)$$

In Equation (4), $m_{diffuse}$ is the diffuse color of the material, c_{light} is the light source color. To prevent the normal and source direction from being negative, the function taking the maximum value is used to intercept it to 0.

The calculation of high light reflection requires surface normal, perspective direction, light source direction, reflection direction, etc. As shown in **Figure 5**, in the four vector directions, the reflection direction can be calculated by other normal information, as shown in Equation (5):

$$R = 2(N \cdot L)N - L. \quad (5)$$

where R is high light reflection.

The Phong model can then be used to calculate the part of the highlight reflection:

$$c_{specular} = (c_{light} \cdot m_{specular}) \max(0, V \cdot R)^{m_{gloss}}. \quad (6)$$

In Equation (6), m_{gloss} is the gloss of the material, also known as shininess. Used to control how wide the highlights are in the highlights area. $m_{specular}$ is the highlight color of the material to control the intensity and color of the ma-

terial. c_{light} is the color and intensity of the light source. Similarly, the result of $V \cdot R$ is also prevented from being negative here.

Compared to the Phong lighting model, Blinn proposed a simple modification method to get a similar effect [12]. The basic idea is to avoid calculating the emission direction R . To this end, the Blinn lighting model introduces a new vector H , obtained by averaging V and L , then normalization. As shown in Equation (7):

$$H = \frac{V + L}{|V + L|}. \quad (7)$$

Calculations are made using the angle between N and H as shown in **Figure 7**. The calculation of the Blinn lighting model is as shown in Equation (8):

$$c_{specular} = (c_{light} \cdot m_{specular}) \max(0, N \cdot H)^{m_{gloss}}. \quad (8)$$

The schematic diagram of the BSDF is shown in **Figure 6**. Generally, the direction of sight and the light source of the human eye are on the same side, and the refraction direction and the line of sight are different. At this time, people can see the diffuse reflection effect on the surface of the object. In reality, the light source is inside the Caideng, and the human eye observes from the outside, as shown in **Figure 8**.

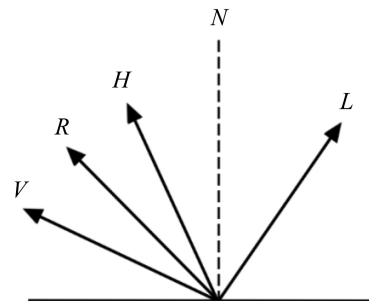


Figure 7. The Blinn lighting model.

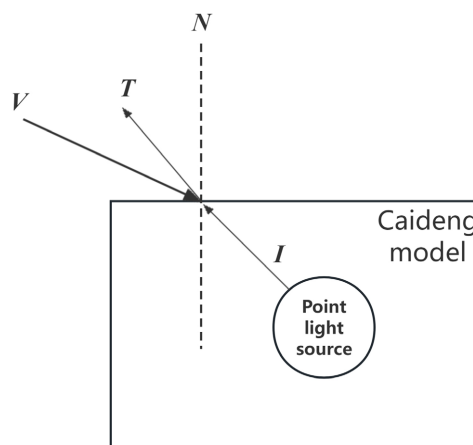


Figure 8. The transmission model of the Caideng point light source.

Some parts in the picture are not lit because when calculating the light in these places, the result of $N \cdot L$ and $V \cdot L$ in the formula is negative, and the result is 0 after calculation. To realize the effect of a point light source illuminating the whole Caideng from the inside, it is necessary to make the calculated results be rendered even if it is negative, so the lighting calculation formula is modified. The following is the lighting calculation formula for integrating the BTDF model.

For the diffuse reflection, the value of $N \cdot L$ is calculated first, as shown in Equation [9], and then modified into Equation [10] is obtained as follows:

$$D = N \cdot L. \quad (9)$$

$$c_{diffuse} = (c_{light} \cdot m_{diffuse}) \max(-D, D). \quad (10)$$

For high light reflection, the value of $V \cdot L$ is calculated first, as shown in Equation [11], and then modified in Equation [12], to obtain results:

$$S = V \cdot L. \quad (11)$$

$$c_{specular} = (c_{light} \cdot m_{specular}) \max(-S, S)^{m_{gloss}}. \quad (12)$$

Then the diffuse reflection and high light reflection are added up to obtain the light transmittance effect of the Caideng model. Finally, after adding and adjusting the material's diffuse color, the material highlights color and material gloss.

After rendering lighting models, it is needed to achieve the screen post-processing effect. Bloom special effects are a common screen effect that spreads the brighter area of the picture into the surrounding area, creating a hazy effect [13]. The principle of Bloom is first to extract the brighter area in the image according to a threshold, store them in a rendered texture, then use Gaussian blur to blur the rendered texture, simulate the effect of light diffusion, and then mix it with the original image to get the final effect. To realize the effect of hazy lights at night, Gaussian blur is adopted, the Gaussian blur utilizes the convolution calculation, using the convolution kernel named the Gaussian kernel [14]. The Gaussian kernel is a square-size filter kernel, where each element is calculated based on the following Gaussian equation:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}. \quad (13)$$

In Equation (13), σ is the standard variance, x and y respectively corresponds to the integer distance from the current position to the center of the convolution kernel.

3.2.2. Comparison

To verify the effectiveness of the above methods, three sets of comparative experiments were designed. **Figure 9** shows the comparison between the customized lighting effect in this article and the original engine effect. The customized lighting effect can achieve different luminous effects by adjusting the point light source parameters.

For the special luminous characteristics of Caideng, there is no shadow on the surface of the lights. **Figure 10** shows the comparison of rendering effects with the fused BTDF model. **Figure 10(a)** shows the effect of the unmixed BTDF model. Compared with the fused BTDF model as shown in **Figure 10(b)**, the shadow appears on the surface of the Caideng model. **Figure 10(c)** and **Figure 10(d)** show the contrast effect of removing texture under the same conditions. It can be found that rendering using the BTDF model does not produce shadow effects.

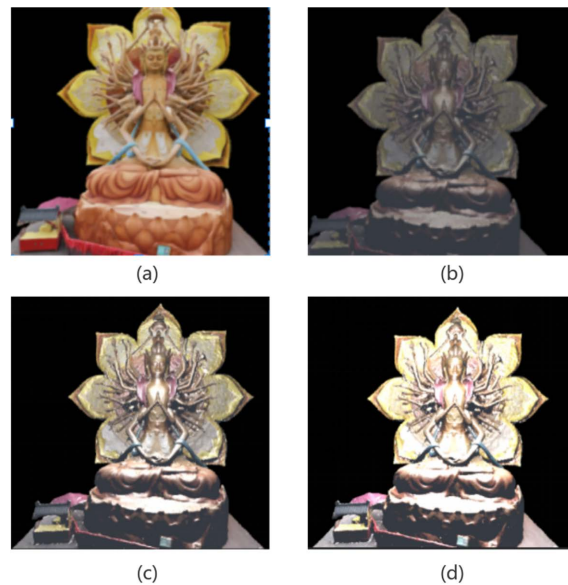


Figure 9. Custom shader comparison. (a) Unity original effect; (b) Light source intensity (0.5); (c) Light source intensity (1.0); (d) Light source intensity (1.5).

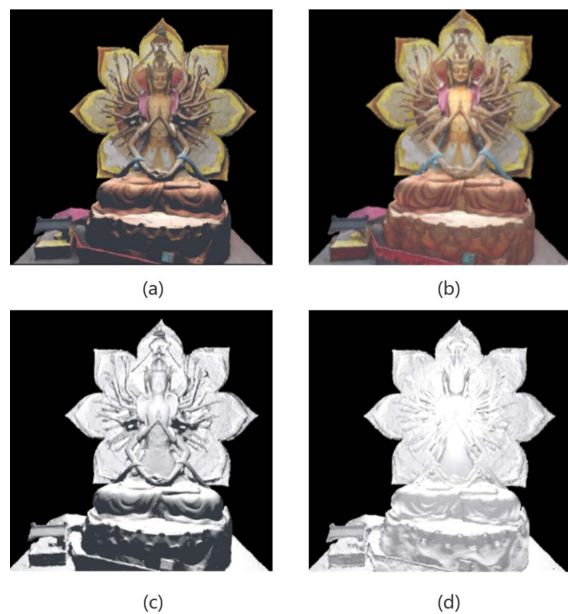


Figure 10. Comparison of fusion BTDF. (a) No BTDF; (b) Fusion BTDF; (c) No BTDF without texture; (d) Fusion of BTDF without texture.

Figure 11 shows a comparison of the effects of post-processing applications. Figure 11(a) shows the effects without post-processing, while Figure 11(b), Figure 11(c), and Figure 11(d) shows the effects of different blur spread values when post-processing is used. The rendering algorithm proposed in this paper can adjust the parameters according to the actual situation to simulate the real effect more realistically.

3.3. Caideng Scene Design

The Caideng scene design section includes environment Settings, Caideng scene models, and others, as shown in Figure 12.

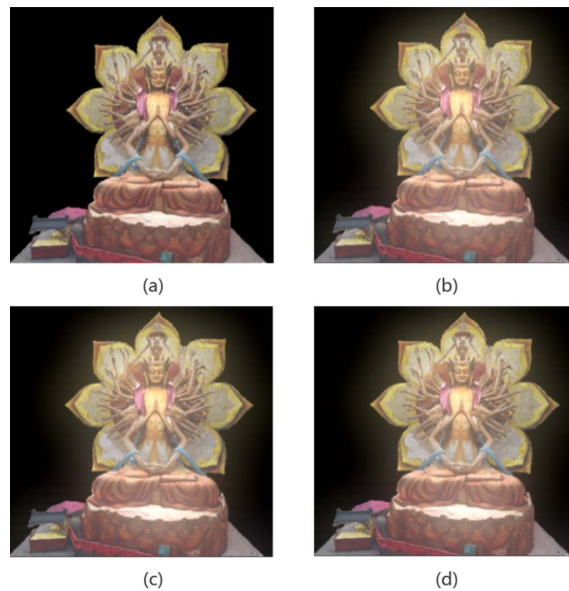


Figure 11. Comparison of post-processing effects. (a) No post-processing effect; (b) Blur spread (0.2); (c) Blur spread (1.5); (d) Blur spread (3.0).

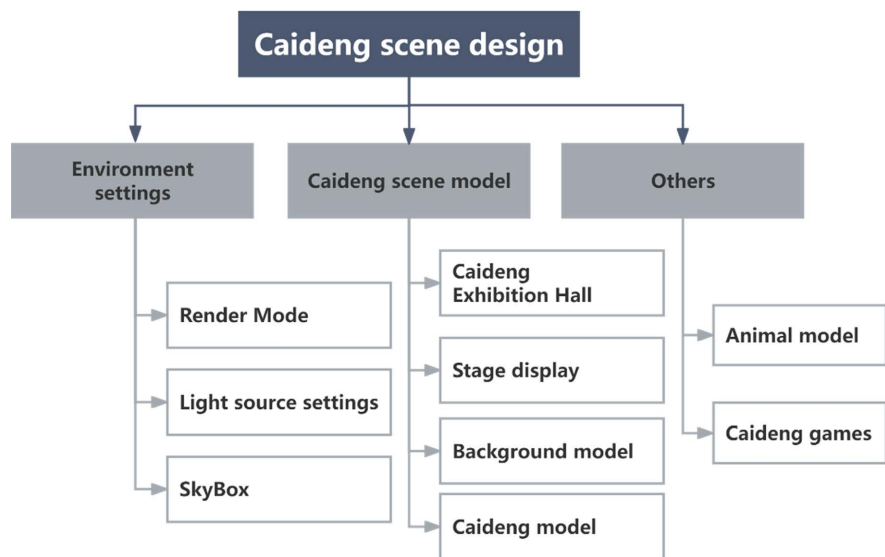


Figure 12. Caideng scene design.

Scene environment settings include skybox settings, light source settings, and rendering mode settings. Modify the Skybox and adjust the lighting effects so that scenes can be switched between day and night. The Caideng scene is composed of multiple Caideng group models. These models are modeled by the drone and then processed by the precision of the model, and the model is optimized into the scene.

Light rendering is the process of achieving the actual effect that a model presents during system runtime, while materials define how the surface should be rendered by including references to the texture used, slice information, color shading, and so on. The material defines how the system renders the surface of the model by selecting a shader for the material and then setting the common properties of the base shader. The 3D scene is built based on computer graphics, and the renderer describes the real world through “surfaces” and “shaders”. The renderer views geometric models as surfaces of various shapes, and material properties are essentially the renderer assigning various shaders to each surface.

To better experience the lighting rendering effect under this system, the rendering process is implemented through Unity Shader, and this Shader is used for all lighting models. The setting steps are:

Step 1: Create a shader and write the rendering algorithm that integrates the BTDF model into the shader file according to the rendering process.

Step 2: Create materials for each Caideng model and add corresponding textures.

Step 3: Match the material with each Caideng model, and set the light source parameters and shader parameters separately.

Step 4: Create a Bloom script component and Bloom Shader, and mount the Bloom component to the Camera to achieve the final blur effect in the rendering process.

The Caideng scene is shown in **Figure 13**. In addition to moving the model of the original Caideng scene into the virtual scene, something else needs to be added. Due to the large scene space, add small animal models to make the scene



Figure 13. The Caideng scene.

too empty; add birds in the air and fly in a fixed path; set up the stage area; and add a large screen to play video. To increase the fun and interactivity, the game area is also set up in the scene to play.

3.4. Interaction Design

The interaction design of Caideng includes scene change, roaming mode, and Caideng game, as shown in **Figure 14**.

Press the right handle “A” key to wake up the UI control interface, as shown in **Figure 15(a)**. In the UI main interface, the optional options include “System Settings”, “Scene switching”, “Select Scene Function”, “Reset scene”, “Caideng Game” and “Exit scene”.

“System settings” can modify sound, lighting, height, and other properties.

The “Scene change” function allows users to switch between each other in the Caideng scene and the Caideng exhibition hall.

“Scene function selection” allows users to select roaming modes, including free-roaming, fixed path roaming, and free roaming with flight functions.

“Reset the scene” brings the user back to the initial state of the scene.

“Caideng games” increases the fun of the roaming scene.

“Exit Scene” will end the system.

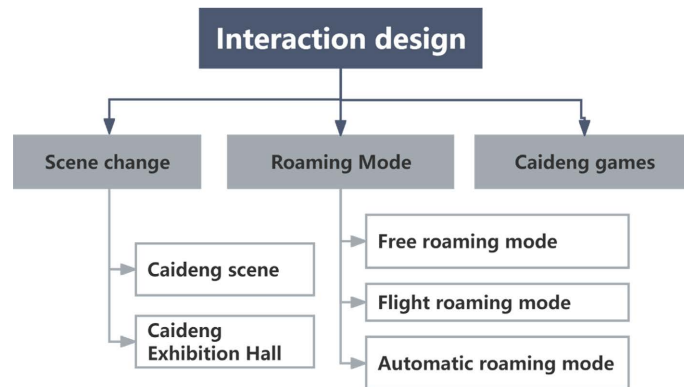


Figure 14. The interaction design of Caideng.

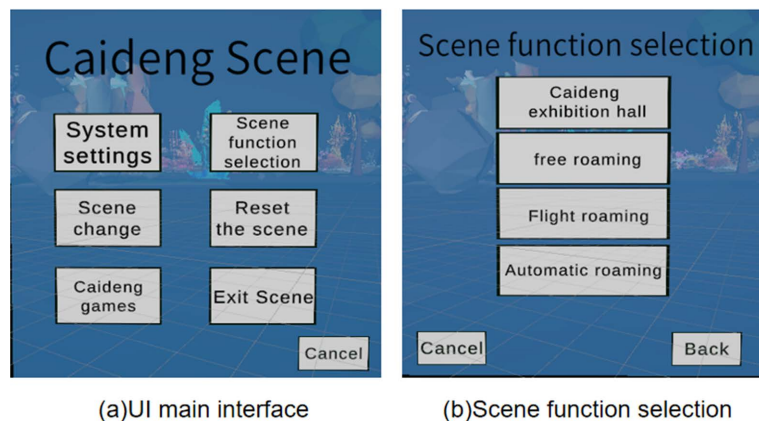


Figure 15. Caideng scene UI interaction.

The roaming function under this system is divided into three modes, including free roaming, free-roaming with flight function, and fixed path roaming, as shown in **Figure 15(b)**. In the free-roaming mode, the user controls the translation of the left hand handle. The rotation of the user's body in situ also follows the perspective in the virtual environment, as shown in **Figure 16(a)**. The free roaming mode with flight function increases the ability of the user to fly in the world of colored lights. By pressing the left handle, the user will rise, and the user pressing the right handle will drop. The perspective is shown in **Figure 16(b)**. In fixed path roaming mode, the user moves under the predefined path and can only rotate his head around the Caideng world.

4. Application Results

The system is running and tested in the Unity environment, with Windows 10 as the operating system, AMD Ryzen 7 5800H with Radeon Graphics as the CPU, NVIDIA GeForce RTX 3060 Laptop GPU as the GPU, 32.0GB (16.0GB X 2) of memory, and Meta Quest Pro as the VR device used. By clicking the Unity interface Window menu, select analysis profile, and then run the system. The frame rate of the running time of the system can be viewed through the profile interface. The system operation test results are shown in **Figure 17**. The frame rate of the system is stable and kept above 60 fps, Meehan et al. found that in an immersive environment, the refresh rate of displayed images is positively correlated with the intensity of user-reported immersion, and the minimum refresh rate that can generate immersion is 15 Hz [15]. Therefore, the system can maintain a stable refresh rate of over 60 Hz to meet the user's sense of proximity experience.

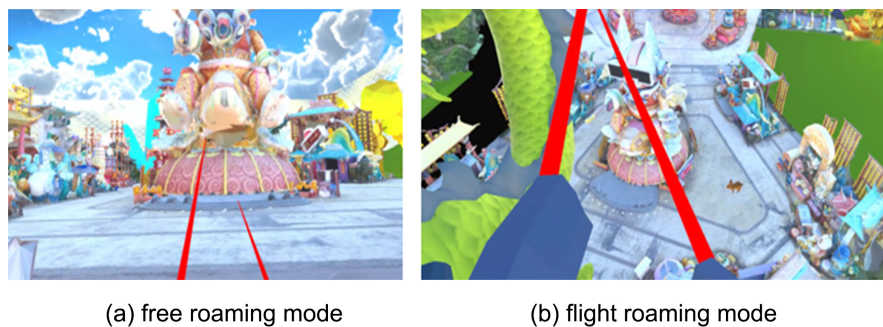


Figure 16. The perspectives in free roaming mode and flight roaming mode.



Figure 17. Test result.

The System Availability Scale (SUS) is a widely used method for evaluating the availability of software and hardware systems [16]. SUS is a fast and reliable evaluation tool that can provide scores for the overall availability of the system. This scale consists of ten items that measure system availability, such as usability, learnability, and user satisfaction. The performance requirements for evaluating the usability of the design scheme in this article are only to evaluate whether it reaches a level acceptable to the user. Therefore, it is decided to use the SUS scale method for the usability evaluation of the scheme, as shown in **Figure 18**.

This experiment invited a total of 10 participants (8 males and 2 females), aged between 22 and 26 years old. Before the experiment, each participant was provided with a detailed introduction to the system’s functions and interaction methods, and participants were encouraged to operate and experience the system in advance to become familiar with its functions and interaction methods. After completing the experience with 10 participants, the SUS scale data was collected and analyzed, and the results are shown in **Table 1**.

Among them, #1 to #10 represent experimenter numbers, Q1 to Q10 represent scale questions, odd-numbered questions represent positive statements, and even-numbered questions represent negative statements. Therefore, it is necessary to standardize the scores of each question to obtain the final score. After calculation, the average SUS score of this system is 81.5, indicating that it has good usability and can provide users with a better immersive experience.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
7. I would imagine that most people would learn to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5

Figure 18. SUS questionnaire.

Table 1. Table type styles.

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	AVG
Q1	4	3	5	4	3	4	4	4	4	3	3.8
Q2	2	2	3	2	2	1	2	3	1	2	2
Q3	4	4	5	4	4	4	5	4	4	5	4.3
Q4	1	2	2	1	1	2	1	1	1	2	1.4
Q5	4	3	4	4	4	4	3	4	4	4	3.8
Q6	2	2	1	2	3	2	2	1	2	1	1.8
Q7	4	4	5	4	4	5	5	4	4	4	4.3
Q8	1	1	2	1	3	1	2	2	1	1	1.5
Q9	4	4	4	5	5	5	4	4	5	5	4.5
Q10	1	2	2	1	2	2	1	1	1	1	1.4
SCORE	82.5	72.5	82.5	85	72.5	85	82.5	80	87.5	85	81.5

5. Conclusion

This paper according to the characteristics of the nighttime Caideng model, integrates several lighting model algorithms, successfully applied to the simulation of virtual Caideng lighting, and improves the Caideng model reality. And image post-processing effect makes the picture rendering more in line with the night observation effect, letting the user have a stronger immersion. Finally, the design and development of a virtual Caideng roaming system were completed. But there are still some functions to be perfected. Firstly, this paper uses the traditional lighting model, but to realize the BTDF algorithm, it needs to be described through a complex mathematical model. Secondly, the current model needs to be optimized to achieve a more real effect. Finally, add a real physical engine system to the Caideng model, so that users can interact with the Caideng model more naturally and increase immersion.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Wu, Y.D., Chen, J.M., Luo, Y., Wang, X.F., *et al.* (2023) An Overview of Caideng Metaverse Research. *Big Data Research*, **9**, 97-113.
- [2] Wang, J.Q., Cheng, Z.M., Liu, L., *et al.* (2021) Immersive Interactive Experience of Qinhuai Lantern Festival Based on Virtual Reality. *Journal of Beijing Institute of Graphic Communication*, **29**, 157-160.
- [3] Wang, Z.G., Chen, Y.S. and Huang, W.X. (2020) The Inheritance and Innovation of

Folk Lantern Art. *Entertainment Technology*, No. 6, 1-5+10.

- [4] Li, A. and Xu, H.H. (2020) Research on Key Technologies of Plugin Free Virtual Roaming Based on WebGL. *Computer Application Research*, **37**, 227-229.
- [5] Gao, Y.D., Li, M.M., Yong, Y.Y., *et al.* (2013) Design and Implementation of a Library Virtual Roaming System Based on UDK Engine. *Modern Education Technology*, **23**, 121-126
- [6] Yang, X.B., Chen, B.Z., Zhang, W.X., *et al.* (2017) Design of Campus Virtual Roaming System Based on 3D Laser Scanning Technology and VR Platform. *Laboratory Research and Exploration*, **36**, 110-114
- [7] Griwodz, C., Gasparini, S., Calvet, L., *et al.* (2021) AliceVision Meshroom: An Open-Source 3D Reconstruction Pipeline. *Proceedings of the 12th ACM Multimedia Systems Conference*, Istanbul, Türkiye, 28 September-1 October 2021, 241-247. <https://doi.org/10.1145/3458305.3478443>
- [8] Du, X.L. and Tan, J.W. (2014) A Review of Lighting Models in Realistic Graphics. *Electronic Technology & Software Engineering*, No. 1, 112-113.
- [9] Lambert, J.H. (1760) *Photometria*. Eberhard Klett Verlag, Augsburg.
- [10] Phong, B.T. (1998) Illumination for Computer Generated Pictures. In: Phong, B.T., Ed., *Seminal Graphics. Pioneering Efforts That Shaped the Field*, Association for Computing Machinery, New York, 95-101. <https://doi.org/10.1145/280811.280980>
- [11] Bartell, F.O., Dereniak, E.L. and Wolfe, W.L. (1981) The Theory and Measurement of Bidirectional Reflectance Distribution Function (BRDF) and Bidirectional Transmittance Distribution Function (BTDF). *Radiation Scattering in Optical Systems. SPIE*, Huntsville, United States, 3 March 1981, 154-160. <https://doi.org/10.1117/12.959611>
- [12] Blinn, J.F. (1977) Models of Light Reflection for Computer Synthesized Pictures. *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*, San Jose, California, 20-22 July 1977, 192-198. <https://doi.org/10.1145/563858.563893>
- [13] Magdics, M., Sauvaget, C., García, R.J., *et al.* (2013) Post-Processing NPR Effects for Video Games. *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, Hong Kong, 17-19 November 2013, 147-156. <https://doi.org/10.1145/2534329.2534348>
- [14] Flusser, J., Farokhi, S., Höschl, C., *et al.* (2015) Recognition of Images Degraded by Gaussian Blur. *IEEE Transactions on Image Processing*, **25**, 790-806. <https://doi.org/10.1109/TIP.2015.2512108>
- [15] Michael, M., Brent, I.M.W. and Frederick, P.B. (2002) Physiological Measures of Presence in Stressful Virtual Environments. *ACM Transactions on Graphics*, **21**, 645-652. <https://doi.org/10.1145/566654.566630>
- [16] Brooke, J. (2013) SUS: A Retrospective. *Journal of Usability Studies*, **8**, 29-40.