International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-17

https://doi.org/10.70917/ijcisim-2026-0376
Article

Research on three-dimensional animation character hair rendering and virtual scene fusion technology based on ray tracing algorithm

Yuxuan Liu [1,*]

[1] NanJing XiaoZhuang University, 210000, Nanjing, China
* Correspondence author: lyx226500@163.com

Abstract: The increasingly powerful performance of computational graphics chips, the rise of augmented reality and virtual reality, real-time graphics rendering has a pivotal role in today's social life, and realistic hair rendering has been one of the difficult problems in computer graphics. In this paper, based on the changes of viewpoints, objects, and light sources in the 3D scene changes, we propose ray tracing model and calculate its pixel unbiasedness to ensure the correctness of the calculation results. A simplified Fresnel formula is used to calculate the contour light of hairs, and at the same time, an improvement of the basic Fresnel formula is proposed to enhance the calculation of the table, contour light intensity and width. Based on the principle of hierarchical shadow mapping, realize the simulation of hair sub-shadow effect. Construct the virtual scene, obtain the body skeleton and other data of the 3D animated character, and improve the fusion between the virtual scene and the 3D animated character. After analyzing the rendering performance and the fusion effect of the virtual scene, after turning on the simulation module, the GPU time increases by 16.98%, while the CPU time increases by 25 μs, and the system's overhead of the Compute Shader reaches 18.65%. The algorithm seeks the mean image in 40-60 images, and the 2×2 chunk fitting method can get better experimental results, at this time, the PSNR is 38.7651dB and 38.8879dB, respectively.

Keywords: light tracking model; Fresnel formula; hair contour light; hierarchical shadow mapping; virtual scene fusion

1. Introduction

Since the 1990s, computerized 3D animation works have been springing up. The quality of visual effects of animation works has been improving under the superb technical support, and 3D animation has become more and more internationalized. Therefore, in order to make the rendering of 3D animation with faster speed and detail processing to obtain the best visual effect, and to promote the animation works to be more competitive in the international market in terms of price and quality, animation enterprises around the world have spared no effort to update the rendering technology, and to choose effective ways to improve the rendering effect of 3D animation [1-2].

Rendering is the creation of a geometric model of the desired scene in the computer, with certain materials, textures and colors, plus a light source, according to certain lighting conditions, calculating and generating the scene graphics with a realistic effect [3]. From the viewer's point of view, high quality images are one of the main driving forces that attract them to watch animated movies. From the perspective of the enterprise, rendering is one of the most time-consuming aspects of 3D animation film production, the higher the quality of the image, the longer the time required, the higher the

technical requirements [4-6]. Generally speaking, in the process of animation film production, to ensure that the image rendering output quality remains unchanged under the condition of shortening the production time to improve the production speed and efficiency has always been the field of design and performance is constantly pursued. Under the condition that the computer hardware configuration and software version remain unchanged, if the image rendering speed is increased to shorten the rendering time, the image rendering quality will be reduced as a price, and vice versa [7-10]. This has been a real problem in the field of animation production. The speed and quality of 3D animation rendering is affected by a variety of factors, such as the amount of computer hardware resources, the development of rendering software or the way the production process is configured [11-12]. However, under the current 3D animation rendering technology, the creation and production situation is not very satisfactory, which is manifested in the simple setup of rendering by the workers to reduce costs, resulting in poor picture quality and leading to defective character details. Character details rendering link is not in place, the picture cuts the sense of serious, not well play the role of the rendering system [13-16]. Among them, animation details, such as fur color, is an important part of the rendering in the film and television or game industry, which is related to the character as a whole and the harmony of the screen.

In addition, in the creation of animation, not only the rendering effect puts forward high requirements, scene construction is no exception. Animation scenes usually provide services for the performance of animation characters, the design of animation scenes should meet the requirements, show the historical background of the story, cultural landscape, geographic environment and the characteristics of the times, and should clearly express the time and place of the story, combined with the overall style of the nuclear film design, to provide suitable occasions for the performance of animation characters, so that the audience or the player to produce a strong empathy to enhance the experience! The sense of experience is enhanced [17-20]. It is realized by constructing a virtual scene with three-dimensional sense and real sense through computer technology [21]. The 3D animation scene not only includes the modeling and rendering of the scene, but also involves the comprehensive application of light and shadow effects, special effects, color and sound [22-25]. Together, these components present a three-dimensional, realistic virtual world, bringing an immersive audiovisual experience to the audience. Character rendering and virtual scenes are both important aspects of animation creation and complement each other, and a technique that simultaneously satisfies character detail rendering and virtual scenes is of great practical significance.

In this paper, the unbiasedness of the ray tracing algorithm is calculated by combining the changing characteristics of viewpoints, objects and light sources in the 3D animation scene to ensure the correctness of the calculation results. Using the simple Fresnel formula, the contour light effect of the hair is calculated. In order to control the contour light intensity, width and other variables to adapt to the contour light effect of different hairs, corresponding improvements are proposed to the Fresnel formula. For the different properties of light in different directions, this paper applies the Kajiya-Kay light model to deal with the anisotropic light of the hair, and uses the hierarchical shadow drawing method to represent the self-shadowing effect of the hair. The virtual scene is constructed to obtain the body surface and skeleton data of the 3D animated character, focusing on the part of data that connects the character's appearance with the virtual scene, and realizing the integration of the virtual scene and the 3D animated character. Using simulation experiments, the rendering performance of the hair of the 3D animated character and the fusion effect of the virtual scene are analyzed respectively.

2. Ray tracing based animated character hair rendering

2.1. Light tracing design

2.1.1. Changes in point of view

Assuming that only the attributes of the point of view (camera) in the scene (the attributes of the point of view include position, viewing angle, aperture size, focal length, shutter time, etc.) change, the images seen by the point of view in different states are different, even if the change in the attributes of the point of view is very small, the images seen are still slightly different. For a scene where only the attributes of the viewpoint are changed S, all the states of the viewpoint are the set $A\{a_1 a_2 \ldots a_n \ldots\}$ and the set of images seen in all the states of the viewpoint are the elements of $F\{F_1 F_2 \ldots F_n \ldots\}$, A and F - correspondences, i.e., there are correspondences:

$$(S, a_n) \leftrightarrow F_n \tag{1}$$

2.1.2. Object changes

For a scene S with only object changes, the set of all object states is $B\{b_1 b_2 ... b_n ...\}$ and the set of images corresponding to all object states is $F\{F_1 F_2 ... F_n ...\}$. Then B corresponds to the elements in F one-to-one. In scenes with object changes, a ray that intersects a dynamic object cannot be directly reused because this ray changes in other frames. A regular expression can be used to express the intersection of a ray with an object. Suppose the light source is L, the viewpoint is E, the ray intersecting with a dynamic object is written as D and the ray intersecting with a static object is written as S, then the ray intersecting with a dynamic object is: $E(D \mid S)^* D(D \mid S)^* L$.

2.1.3. Light source changes

Light sources include point light sources, surface light sources, parallel light sources, etc. These light sources can be regarded as a special kind of object, i.e., an object that emits light, so the situation that needs to be taken into account for the change of the light source can be equated to the situation of the change of the object. Any effective light calculation must eventually be connected to the light source, if the light source is changing, then the last part of the light path must intersect with the changing light source, so such a light can not be directly reused. If all the light sources in the scene are changing, then all the rays in the scene have to be recalculated in some way.

To summarize, assuming that the viewpoint state in scene S is a, the object state is b, the light source state is c, and the other factors are d, the image obtained in this state is F, and the set of all scene states $\{(a_1 b_1 c_1 d_1),(a_2 b_2 c_2 d_2),...(a_n b_n c_n d_n)...\}$, and the set of images corresponding to all of these states is $\{F_1, F_2, ...F_n ...\}$, and the elements of these two sets correspond one to one:

$$(S, a_n, b_n, c_n, b_n) \leftrightarrow F_n \tag{2}$$

2.1.4. Unbiased nature of calculations

By unbiasedness of computation, we mean that the expectation of each pixel is equal to the true value of that pixel, and unbiasedness theoretically guarantees the correctness of the computed results. Since the animation system needs to multiplex the samples between frames, to prove the unbiasedness of the computation of each frame of the animation, it is only necessary to prove that each original sample and its multiplexing process are unbiased. Since the method of light computation does not affect the results of the proof, the simplest ray tracing algorithm is used here as an example of the model [26]. Figure 1 shows the inverse ray tracing model, for a contributing primitive sample, its contribution in the current frame is L, and the contribution of multiplexing to another frame is L'. The light contribution of the light emitted from the light source to the intersection of the primitive sample and the object at the intersection point P after a number of reflections is L_1. Let the value of the bi-directional scattering distribution function of the primitive sample at the point P be s, and the probability of sampling be p. Since the process of primitive to multiplexing sampling, the scene changes, so the sampling probability of the multiplexed sampling in the new scene becomes p' and there is a new bidirectional scattering distribution function value of s' and a new light contribution from the new light source to the sampled intersection point P L_i'. The results of the original sampling computed with Eq. (3) are unbiased, and the values of the multiplexed sampling L' are also unbiased when computed according to Eq. (4), and next it is only necessary to show that the multiplexing strategy employed during viewpoint changes, object changes, and light source changes conforms to Eq. (4) in order to guarantee the unbiased nature of the multiplexing process. The proof needs to be discussed in three ways:

$$L = \frac{L_1 * s}{p} \tag{3}$$

$$L = \frac{L_1' * s'}{p'} \tag{4}$$

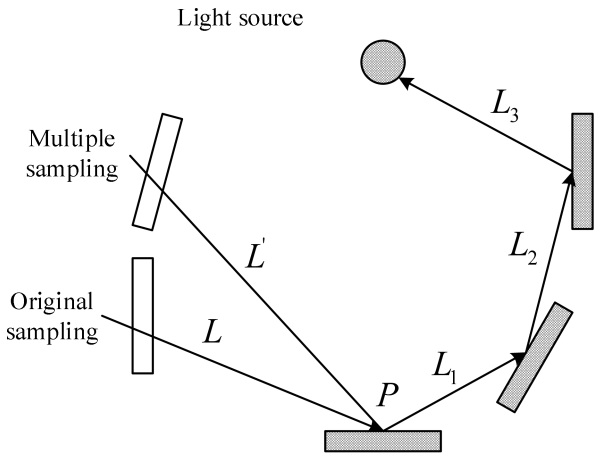


Figure 1. Inverse ray tracing model

The path of the raw sample intersects only the static object and there is no point-of-view movement from the raw sample to the multiplexed sample. Since the path of the original sampling intersects only the static object and the sampling probability at point P is only related to the two-way scattering distribution function of the object, the sampling probability p' of the multiplexed sampling at point P is the same as the probability p of the original sampling due to the stationarity of the object, and the values of the two-way scattering distribution function, s', are the same as s, and similarly the contribution of the source of light, L_1', is the same as L_1, so that L' and L are equal in this case, i.e., the results of the original sampling can be The results of the original sampling are directly used as the results of the multiplexed sampling, and this direct use is unbiased.

The path of the raw samples intersects only static objects, and there is a point-of-view shift from the raw samples to the multiplexed samples. Although the path of the original sample intersects only the static object, the change in the viewpoint results in a different outgoing direction at point P for the original sample and the multiplexed sample, so that they have different sampling probabilities and different values of the bidirectional scattering distribution function at point P. However, since the incident direction does not change, the contribution of the light source to point P remains unchanged. Therefore it is necessary to use the sampling probability p' of the multiplexed samples in the new scene and the new value of the bi-directional scattering distribution function s' to calculate the contribution of the multiplexed samples as in Eq. (5), and this situation illustrates the process of calculating the reprojection with viewpoint shift, which should take into account the change of the sampling probability and the change of the bi-directional scattering distribution function:

$$L' = \frac{L_1 * s'}{p'} \tag{5}$$

The path of the raw samples intersects with the dynamic object. Since the properties of the object at the intersection change, the sampling probability, the value of the bidirectional scattering distribution function, and the contribution of the light source to the intersection change for both the original and the multiplexed sample, so the value of the multiplexed sample should be calculated using Equation (4), which means that the value of the multiplexed sample needs to be recalculated. This situation shows that if the path of the original sample intersects with a dynamic object, this sample cannot be used as a multiplexed sample, and there should be a new process of path tracing at the multiplexed sample to perform a new calculation.

The three cases discussed above are able to include the process of reusing all the original samples during viewpoint changes, object changes and light source changes, so that every frame generated by this animation rendering system is unbiased.

2.2. Realization of 3D Animation Character Hair Rendering

2.2.1. Contour Lighting Effect for Hair

Contour light belongs to a part of the ambient lighting, in the real world, when the light passes through the edge of the illuminated object, it will produce the effect of contour light. The fundamental

reason for the formation of contour light is that hairs have sub-surface scattering properties. When light passes through a transparent or semi-transparent surface, the light passes from the object through the internal scattering, and then out through other vertices on the surface of the object.

Hair consists of very fine fibers, light can easily penetrate its surface, especially light-colored hair, and the darker the color, the stronger the ability to absorb light, dark hair light transmission is relatively low. In real-time rendering, the contour light effect is usually calculated by a simplified Fresnel formula, which is defined as follows [27]:

$$Fresnel = 1 - worldViewDir \cdot worldNormal \tag{6}$$

Where $worldNormal$ represents the normal of the vertex in the world coordinate system, and $worldViewDir$ represents the direction of the line of sight. After the dot product operation, the normal and the direction of the line of sight are perpendicular to each other in the edge region, and the result is 0 after dot product, while the middle region is closer to parallel, that is, white. After inversion, the Fresnel value is obtained.

In order to be able to control the intensity and width of the contour light to adapt to the contour light effect of different hairs, this paper improves the basic Fresnel formula accordingly, and the improved mathematical expression is as follows:

$$Fresnel = FreScale(1 - worldViewDir \cdot worldNormal)^{FreStrength} \tag{7}$$

In the formula, $FreScale$ represents the intensity of the contour light, the larger the value, the more obvious the color of the edge, and the $FreStrength$ value represents the width of the contour light, the larger the value, the smaller the width of the contour light.

2.2.2. Anisotropic Lighting of Hair

Anisotropy, as opposed to isotropy, refers to the property of having different behaviors in different directions, i.e., its behavior is related to the direction. For example, in physics, if different behaviors occur when measured in different directions along a material, the material is usually said to be “anisotropic”. The anisotropy of a material is mainly due to the presence of small organized grooves on the surface of the material, and the anisotropy of hair is also due to the special structure of the surface. The unique surface structure of hair leads to the unique lighting effect of hair, and the Kajiya-Kay lighting model can better deal with the anisotropic lighting of hair, which is also the most widely used hair lighting model [28].

(1) Kajiya-Kay light model

The most significant difference between the Kajiya-Kay light model and other light models is that it uses the tangent to the hair rather than the normal to this point for its calculations. In the Kajiya-Kay light model, the hair is abstracted as an opaque cylinder without refraction and internal reflection, and the light illumination of the hair is mainly composed of the superposition of two parts: diffuse reflection $\psi_{difluce}$, and highlight reflection $\psi_{speular}$, and the final light intensity I can be calculated by the following formula:

$$I = I_l \psi_{diffuse} + I_l \psi_{specular} \tag{8}$$

1) Diffuse reflection term

The computation of the diffuse reflection term in the Kajiya-Kay light model is shown schematically in Fig. 2.

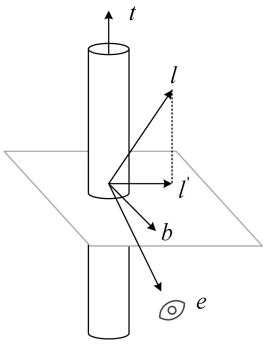


Figure 2. Diffuse reflection term of Kajiy-Kay illumination model

The unit vector l represents the direction of incidence of the light ray, the unit vector t represents the tangent direction vector of the cylinder, and the auxiliary vector b can be obtained by doing the cross product of l and t. The unit vector l' is the projection of the incident light ray l in the cross section of the cylinder, which is defined by the equation shown in (9):

$$l' = \frac{l-(t\cdot l)t}{\| l-(t\cdot l)t \|} \tag{9}$$

After determining vector b and vector l', the normal direction vector n passing through a point on the cylindrical cross-section can be calculated by equation (10).

$$n = b\cos(\theta) + l'\sin(\theta) \tag{10}$$

where θ takes values from 0 to π, which represents the position of a point on the cylindrical cross-section. The diffuse light intensity can be calculated from the Lambert light model, which is defined as follows [29]:

$$\begin{aligned} \psi_{diffuse} &= k_d r\int_0^{\pi}(n\cdot l)d\theta \\ &= k_d rl\int_0^{\pi}(b\cos(\theta)+l'\sin(\theta))d\theta \\ &= k_d r(l\cdot l')\int_0^{\pi}\sin(\theta)d\theta \\ &= k_d\left[l\cdot\frac{l-(t\cdot l)t}{\| l-(t\cdot l)t \|}\right] = k_d\sin(t,l) \end{aligned} \tag{11}$$

In Eq. (11), k_d represents the diffuse reflection coefficient, which takes a value between 0 and 1, and $\sin(t,l)$ represents the sine of the angle between the incident vector l and the tangent line t.

2) High light reflection term

The calculation of the high light reflection term is shown in Figure 3:

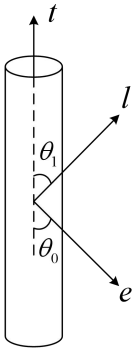


Figure 3. Highlight reflection term of Kajiy-Kay illumination model

θ_1 is the angle between the incident direction l and the tangent direction t, and e is the viewing angle vector. The high light reflection term can be derived from the Phong model, whose defining equation is shown in (12), θ_0 is the angle between the incident light l and the tangent direction t, where k_s is the high light reflection coefficient and p is the high light intensity coefficient:

$$\begin{aligned} \psi_{specular} &= k_s(\cos\theta_1\cos\theta_0+\sin\theta_1\sin\theta_0) \\ &= k_s((t\cdot l)(t\cdot e)+\sin(t,l)\sin(t,e))^p \end{aligned} \tag{12}$$

(2) Improvement of hair highlighting effects

The underlying Kajiya-Kay lighting model is the phenomenological, or empirical, model that is used to represent the most obvious scattering characteristics of hair fibers. Since the hair is abstracted as an opaque cylinder in the Kajiya-Kay model, it is not capable of producing internal reflections, which leads to some of the model's inability to represent some of the hair effects that can be observed with the naked eye, such as the process of light propagation within the hair fiber.

The Marschner model is a physically based light model, which not only needs to calculate the surface scattering of hair but also needs to further calculate the internal scattering, resulting in low computational efficiency. In order to be able to simulate this physically based hair lighting model and use it in real-time rendering, Thorsten decomposes the highlights of the same light source into two different highlight effects, which have different colors, different highlight indices, and are at a certain distance from each other. If the same tangent is used, the two layers of highlights will overlap each other. In order to stagger the two layers of highlights, different offsets are given based on the original tangent direction in the calculation. After improvement, the highlight term is represented by the following equation:

$$\begin{aligned}\psi_{specular} &= k_{s1}((t_1 \cdot l)(t_1 \cdot e) + \sin(t_1, l)\sin(t_1, e))^{p_1} \\ &+ k_{s2}((t_2 \cdot l)(t_2 \cdot e) + \sin(t_2, l)\sin(t_2, e))^{p_2}\end{aligned} \tag{13}$$

where parameters t_1 and t_2 represent tangents with different offsets and p_1 and p_2 represent different highlight indices.

2.3. Simulation of self-shadowing effect of hairs

2.3.1. Principle of Hierarchical Shadow Mapping

Based on Eqs. (11) and (10), the following equation was used to represent the self-shadowing effect of hairs:

$$FurShading = I_a k_a \times \alpha + (I_l \psi_{diffuse} + I_l \psi_{specular}) \times \eta \tag{14}$$

In Eq. (15), I_a represents the ambient light intensity, k_a represents the ambient light reflection coefficient, α is the layer attenuation coefficient of the ambient light, and η represents the ratio of the remaining light of the highlights and diffusely reflected light reaching the current layer. The layer attenuation coefficient α is determined by the following equation:

$$\alpha = \frac{(1-\lambda)L_{cur}}{L_{\max}} + \lambda, 0 \le \lambda \le 1 \tag{15}$$

where $L_{\max}$ represents the total number of layers of the mesh slice created to generate hairs and L_{cur} represents the number of layers in which the current layer is located. λ represents the self-shadowing ratio of the ambient light, the attenuation coefficient α is taken to be the minimum value on the side of the hairs near the skin, the opacity of the intersection point $L_0, L_1, L_2, \ldots, L_{n-1}$ between the light OL and each layer is $A_0, A_1, A_2, \ldots, A_{n-1}$ each, and the expression for the light ratio of the light reaching the point L is given by:

$$\eta = (1 - A_0) \times (1 - A_1) \times ...(1 - A_{n-1}) \tag{16}$$

In order to obtain the η values for each layer, a hierarchical shadow plotting method is proposed in this paper, which is described in detail in the next subsection.

2.3.2. Hierarchical Shadow Mapping Steps

The specific steps for shadow generation are as follows:

(1) If the current layer is the outermost layer S_0, this layer is not affected by hair self-shadowing, so the $shadowColor_0$ value of the current layer is 0.

(2) If the current layer is S_1, first set the background color to white, and then draw the mesh surface S_0 with P as the view center, mapping the texture map of the first layer on S_0. Then use the pixel shader to draw S_1 the layer shadow with the color value set to $shadowColor_1 = 1 - A_0$ and A_0 indicating the opacity component of the first layer texture map mapped by the current pixel. The result is saved as $shadowMap_1$ texture.

(3) If the current layer is S_2, as in step (2), set the background color to white, draw the laminated surface S_1 with the light source as the view center, map the second layer texture on S_1 and

$shadowMap_1$. Then use the pixel shader to draw the pixel color. Here, unlike step (2), the shadow color value is set to $shadowColor_2 = (1 - A_4) \times \eta$, where A_1 represents the opacity component of the second layer texture mapped by the current pixel, and η is the $shadowMap_1$, texture value mapped by the current pixel. The $shadowColor_2$ value is the remaining light value after the light passes through S_0 and S_1, saving the drawing result as $shadowMap_2$.

(4) According to the self-shadow drawing formula $shadowColor_i = (1 - A_{i-1}) \times \eta$, draw the self-shadow of the remaining layers in turn, using the same method as in step (3).

3. Integration of 3D animated characters and virtual scenes

3.1. Virtual scene construction

Using the modeling software 3DsMAX can quickly build large 3D animated virtual scenes, but only through it to complete all the 3D model scene texture is unrealistic and will consume a lot of system resources. Usually the pictures of some real objects are pasted onto the solid model, which also makes the scene have a more realistic effect. In the modeling process, a large number of pictures are taken on site and processed by cropping and splicing to get the desired texture mapping.

The modeling of the tree model in the scene can be done by a single plate mapping, but this method cannot meet the requirements of the senses, and the sense of depth is too poor. The above problem can be solved by utilizing the notice board technology of osg::Billboard class of OSG scene interface, combined with the Alpha fusion algorithm of osg::AlphaFunc class.The Alpha fusion algorithm is defined as follows:

$$\begin{cases} R(C) = alpha * R(B) + (1 - alpha) * R(A) \\ G(C) = alpha * G(B) + (1 - alpha) * G(A) \\ B(C) = alpha * B(B) + (1 - alpha) * B(A) \end{cases} \tag{17}$$

Where alpha takes values from 0 to 1, with 0 being fully transparent and 1 being fully opaque. It is a polygonal surface simulation method that ensures that the polygon will rotate around the specified axis with the change of viewpoint, keeping the 2D texture tree image pointing at the observer at all times, and also utilizing the human visual error, the feeling is that the same tree can always be seen at different angles. As the viewpoint changes to rotate the 2D texture, and the rotational transformation of a solid is a matrix multiplication operation, it is computationally intensive and has a large amount of shading calculations. Using the crossover method the amount of computation is greatly reduced and the display speed is higher, only the shading effect is not good.

3.2. Integration of virtual scenes and animated characters

3.2.1. Body surface data acquisition

The processing of data from the outer surface of the human body, where it is directly connected to the outer virtual environment, is of particular importance. The use of Kinect software for deep data processing, in which the technology will be customized sensors placed on the surface of the human body, in the sensor to accept a certain length of infrared light in the environment, infrared information can be read on the outer surface of the human body depth information can be a short period of time on the human body's movement to capture and monitor the situation, and its main components are as follows:

(1) Infrared light-emitting equipment, the sensor can emit a specific wavelength of infrared light, the emission power is small, no interference with the environment.

(2) Infrared optical lens and filter, the device is mainly used to collect the infrared signals emitted by the light emitting unit, and the role of the filter is to filter the signals of other wavelengths to improve the accuracy of the signals.

(3) Image processor, using TOF core camera, this part can accept infrared reflection signal, the light intensity of the pixel point and the distance from the luminous unit to the receiver directly is estimated.Kinect II device can on the one hand, obtain the depth data of the human body (the detection distance is 0.1-10 m), on the other hand, it can also be used to identify the color data of the object, and the recognition of the frame rate reaches 50 fps.

3.2.2. Skeletal data acquisition

Human skeletal data is the key to the accurate simulation of human motion data, through the Kinect technology to obtain depth data on the basis of skeletal data analysis and application, in this paper the skeletal data acquisition steps are as follows:

(1) Extract the depth distance image in the system, and extract the human body contour in the image using fuzzy neural network algorithm.

(2) In the human contour image, firstly, the movements of different regions of the human body are categorized and extracted using background differencing, then the clutter filter and noise points of the human body are removed using Poisson's equation, and finally, the refined contour image of the human body removed is used for editing the skeletal data.

(3) Skeletal point data determination. In the refined skeletal data obtained in the previous step, 25 points in the human skeletal joints are intelligently detected and analyzed to form a mapping relationship map. Before the algorithm processing, the height, posture and behavioral data of the human body can be obtained in advance as the matching library of the skeletal point data, and when the human body enters the virtual system, it automatically matches the skeletal point data, which achieves the recognition effect.

3.2.3. Virtual Scene Fusion

Image depth fusion is the main fusion acquisition point in virtual scene and 3D animation character fusion technology. In performing depth data processing, the natural barrier between the human body and the virtual reality data leads to more anomalies. Therefore, noise reduction processing of virtual reality data is required. In the noise reduction process, the mean filter noise reduction method is used for processing, and the calculation process summarizes the gray value of the gray value of the center pixel of the filter window for smoothing, and the response formula of the window center is obtained as:

$$f(x,y)=\frac{1}{mn}\sum_{(r,s)\subset S_{mn}} g(r,s) \tag{18}$$

$$g(r,s)=\sqrt[2]{[(x_i-x_0)^2+(y_i-y)^2+(z_i-z_0)^2]} \tag{19}$$

where: $f(x)$ is the degree of response, $g(r,s)$ is the gray value of a point, and r and s are the coordinates of the 2D UI interface, respectively.

4. 3D animation character hair rendering and scene fusion effect analysis

4.1. 3D animation character hair rendering performance

4.1.1. Rendering module

By loading the hair information into the scene multiple times, the performance performance is tested when multiple objects are on the same screen. Since both the Kajiya-Kay lighting model and the Fermi Hair Demo in this paper have camera-based LOD algorithms, it is only possible to keep the camera as close to the objects as possible. In fact, in applications, it is generally less likely to appear that several hair objects are discharged side by side in front of the camera and based on the state of full-screen display.

Considering that all three applications were tested with different data size and complexity of the chosen use cases, it is not reasonable to put them together for horizontal evaluation. However, in terms of complexity, the final number of rendered faces is Fermi Hair>Kajiya-Kay lighting model>TressFX, and in terms of efficiency TressFX Demo still leads in frame rate only when rendering 1-2 objects, but not as good as the other two apps in all other cases, so in the case of rendering a smaller number of faces, TressFX Demo still can't achieve good results in multi-object same-screen rendering. Also, purely from the frame rate itself, TressFX Demo can't maintain a frame rate of 30fps, and the rendering of multiple hairy objects is already below 30fps when 4 objects need to be rendered.

Here they are plotted together to see their FPS performance for simultaneous rendering of multiple objects by comparing the slope of their respective frame rate drops. Figure 4 shows the same frequency multi-object rendering performance test. Consistent with the previous results, the TressFX Demo has the fastest FPS drop, and its FPS performance cannot meet the needs when there are more than one object in the scene, and the FPS has already dropped from 139 fps to 73 fps at two items.

The Fermi Hair Demo and the Kajiya-Kay lighting model in this paper are very close to each other in terms of performance and drop rate, the average rendering frame rate of the two is 48.39 and 62.04

respectively, and the Kajiya-Kay lighting model in this paper is a bit better. It is judged that Kajiya-Kay illumination model can have better performance in the case of multiple objects, and through the optimization of technology and algorithm in this paper, Kajiya-Kay illumination model can achieve relatively better efficiency.

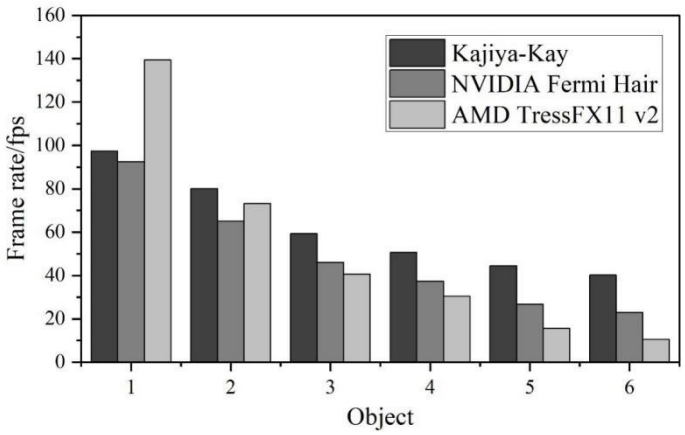


Figure 4. Multi object rendering test

4.1.2. Analog modules

Comparing the performance of turning on and off the simulation module is shown in Table 1. Compared with the previous comparison, when the simulation was turned on, the GPU time increased by 16.98%, while the CPU time increased by 25μ s. The main overhead was in the simulation operation of the hairs, and also the Draw Call was increased to 3 times. At the same time. The overhead of the system for Compute Shader reaches 18.65%.

Table 1. Kajiya-Kay simulation test

Application	Kajiya-Kay no simulation	Kajiya-Kay with simulation
GPU time avg/min/max[μs]	5936/5948/5987	6944/6969/6991
CPU time avg/min/max[μs]	<1/<1/<1	26/20/37
Number Draw Calls	1	3
Bottleneck top three average%	33.98% Rasterizer 27.69%L2-Cache 10.48% Blending and Z-buffer	35.94% Rasterizer 32.09%L2-Cache 11.18% Blending and Z-buffer
Utilization top three average%	30.52% Shader 26.12%Frame Buffer 17.59% Rasterizer	28.44% Shader 27.93%Frame Buffer 18.98% Rasterizer
Shader type	0% VS 3.42% HS 21.49% DS 18.33% GS 57.26% PS -	0% VS 2.38% HS 23.65% DS 13.21% GS 43.48% PS 18.65% CS

4.2. Virtual Scene Fusion

4.2.1. Objective experimental results

Tests on the distribution of light sources were grouped, and intra-group testing was carried out, within the same group indicates that the intensity of the light source is the same, but the direction of the distribution of light sources is not consistent. The 11 groups of light conditions were tested, but due to space limitations, some of the experimental data are given below, but from the experimental results, the average gain and the amplitude of the given data is not large. Table 2-Table 4 show the objective experimental results for simple, general and complex scenes respectively. The value after means in the

table indicates the number of images used to find the mean value, and the result in the table is the PSNR value in dB.From the table, it can be seen that, whether it is a simple, general, or complex scene, the image quality PSNR of means_40 is the best, and in the simple scene, overexposure, uniform luminance, unilateral light source, and symmetric light source under the size of the 1×1 bisection of the means_40 's image quality PSNR values are 37.8486, 44.7349, 43.5798 and 45.9648 respectively.

Table 2. Objective experimental results of simple scenes

Light source distribution	Block size	means_5	means_10	means_20	means_40	means_80	VSRS
Overexposure	1×1	30.2745	32.8415	35.7855	**37.8486**	36.2546	28.0856
	2×2	30.1069	32.6152	35.1688	36.7615	35.4925	
	4×4	29.2645	31.0684	32.5987	33.3188	32.7348	
	8×8	27.9255	29.0485	29.8485	30.2348	29.9558	
Uniform brightness	1×1	36.5485	39.7524	42.4265	**44.7349**	43.1718	37.5168
	2×2	36.5633	39.3485	41.8593	43.6484	42.4595	
	4×4	35.4225	38.0318	39.6154	40.4568	39.8945	
	8×8	35.1665	36.3485	37.1985	37.6185	37.3456	
Unilateral light source	1×1	35.5939	38.4265	41.4065	**43.5798**	41.9452	35.3854
	2×2	35.4928	38.1934	40.7245	42.3645	41.1168	
	4×4	34.8155	36.8152	38.4658	39.2648	38.6448	
	8×8	34.0345	35.1298	36.0495	36.4298	36.1388	
Symmetrical light source	1×1	37.9188	41.0258	43.9585	**45.9648**	44.4358	39.7485
	2×2	37.8515	40.7498	43.2498	44.9536	43.7645	
	4×4	37.3698	39.3948	40.9425	41.7856	41.2095	
	8×8	36.0485	37.4588	38.5265	38.9044	38.6348	

Table 3. Objective experimental results of general scenes

Light source distribution	Block size	means_5	means_10	means_20	means_40	means_80	VSRS
Overexposure	1×1	32.7456	34.3188	36.5085	**38.0918**	37.1648	31.3915
	2×2	32.5631	33.9948	35.7563	36.8048	36.1567	
	4×4	30.4185	31.1254	31.7652	31.9487	31.8164	
	8×8	28.0642	28.2948	28.4651	28.4635	28.4685	
Uniform brightness	1×1	38.5435	40.4098	44.2863	**44.2986**	43.5198	37.1625
	2×2	38.3152	39.8452	42.6485	42.6482	42.1098	
	4×4	36.7554	37.5154	38.6485	38.8468	38.4262	
	8×8	34.7498	35.0685	35.3485	35.3785	35.3645	
Unilateral light source	1×1	39.7641	41.7298	44.0864	**45.5298**	44.5987	37.8419
	2×2	39.4268	40.8456	42.5935	43.5198	42.9185	
	4×4	37.2185	37.8584	38.4521	38.6456	38.7248	
	8×8	35.0795	35.2948	35.4982	35.6048	35.6155	
Symmetrical light source	1×1	41.9463	44.1642	46.7198	**48.3985**	47.4985	40.6452
	2×2	41.6452	43.3048	45.0865	46.0985	45.5596	
	4×4	39.7955	40.6348	41.3898	41.6548	41.4685	
	8×8	37.5168	37.7955	37.9896	38.1297	38.0948	

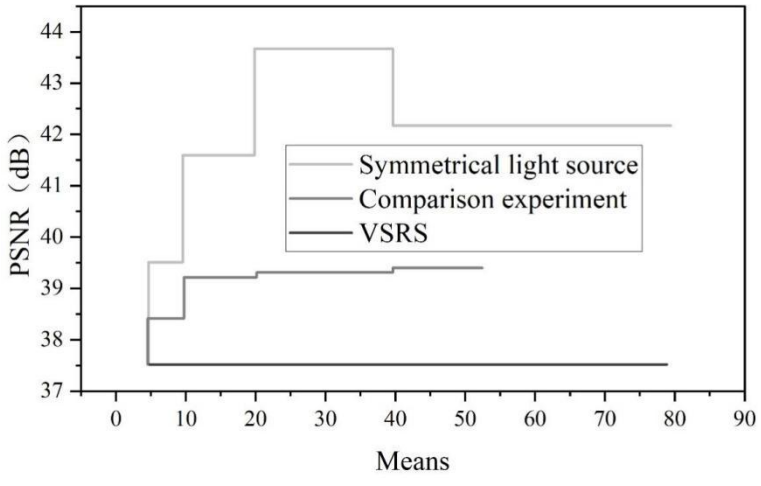
Table 4. Objective experimental results of complex scenes

Light source distribution	Block size	means_5	means_10	means_20	means_40	means_80	VSRS
Overexposure	1×1	27.7655	29.2485	31.1884	**33.0845**	31.7348	26.7215
	2×2	27.6748	29.0657	30.8162	32.3865	31.2648	
	4×4	27.1635	28.2048	29.3948	30.2488	29.7644	
	8×8	26.3485	26.8648	27.4968	27.9485	27.6985	
Uniform brightness	1×1	33.5487	35.1654	37.1986	**39.1685**	37.9185	33.1754
	2×2	33.4182	34.8865	36.6648	38.2486	37.1687	
	4×4	32.9852	34.0284	35.2195	36.1289	35.5485	
	8×8	32.4282	32.7853	33.8485	33.8585	33.5345	
Unilateral light source	1×1	35.5495	36.1952	38.2489	**40.1948**	38.6482	33.1645
	2×2	34.2988	35.7165	37.3948	38.8655	37.8168	
	4×4	33.5485	34.5615	35.5985	36.4952	35.9168	
	8×8	32.5482	33.0485	33.4285	33.9485	33.8385	
Symmetrical light source	1×1	37.5356	39.4152	41.6248	**43.5988**	42.1985	37.2684
	2×2	37.3185	38.9155	40.6448	42.1654	41.1948	
	4×4	36.5912	37.6125	38.6548	39.5618	39.0425	
	8×8	35.6485	36.1648	36.5498	37.0498	36.8485	

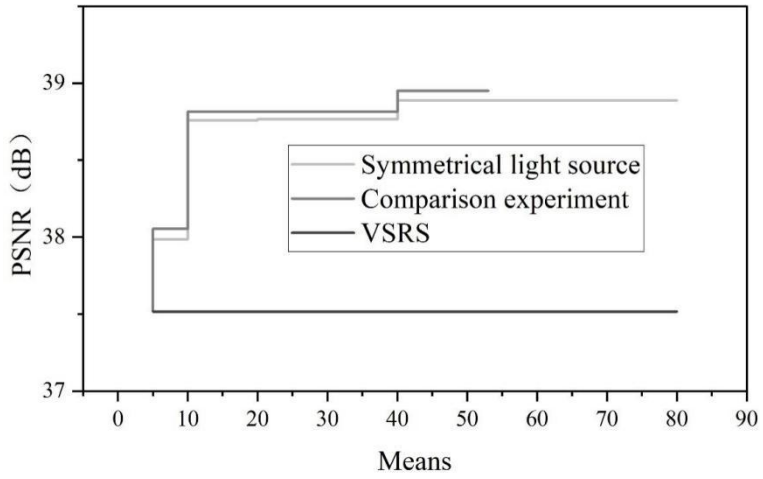
4.2.2. Subjective experimental results

From the experimental results, it can be seen that when the fitting is carried out for light restoration, the light restoration effect is best when the fitting is carried out point by point with the number of reference images remaining constant, and the larger the fitting chunks are divided into, the experimental results become worse, which is due to the fact that the larger the chunks are, the larger the block effect of the fitted image is. In the case of constant chunk size, the more the number of images used to obtain the mean image, the better the synthesized effect will be, but when it increases to a certain extent, it will decline. In the above case, the reason is that the correlation between the image to be fitted and the mean image is larger when there are fewer pictures, and when the fitting is performed, the fitting effect is poorer where there is a large difference between the two images. As the number of images increases, the correlation between the target image and the mean image decreases, and the fitting is more accurate. As the number of images increases, the number of light sources in each direction increases gradually, and the light sources interact with each other, so when the number of images is increased to a certain number, the phenomenon of overfitting will occur. In order to verify this conjecture, another set of comparative experiments, remove some of the darker pictures used to obtain the mean value before, remove the number of light sources distributed in the picture is less, for a single light source or two light sources, the position of the light source has a regular change, the resulting experimental results will be improved than the results of VSRS, but worse than the existing experimental results. Fig. 5 shows the comparison of the experimental results after changing the reference image, Fig. (a) is the chunk size 1×1, Fig. (b) is the chunk size 2×2, and Fig. (c) is the chunk size 4×4.

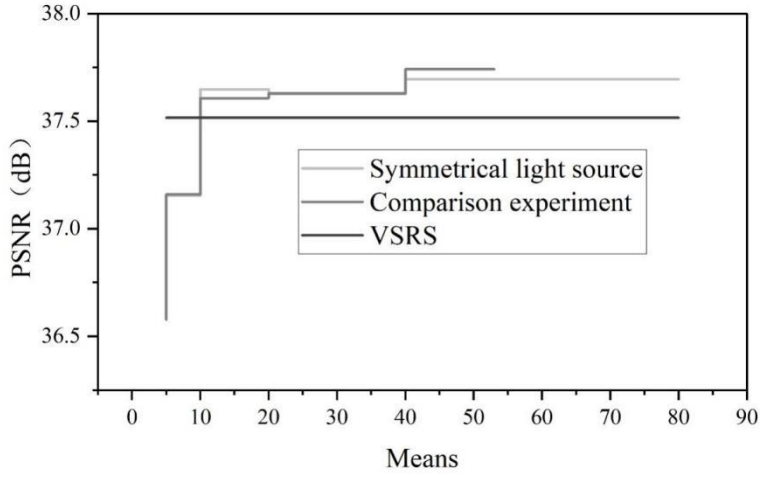
From this experiment, it can be concluded that the picture brightness for the fitting effect will not have a great impact, and not the brighter the picture, the better the fitting effect, which also coincides with the previous experimental results, when the light sources interact with each other the less, the better the fitting effect. From the experimental results and algorithm complexity considerations, elected to use 40-60 pictures to find the average value of the picture, the use of 2 × 2 chunks of the fitting method can be used with less experimental complexity to get better experimental results, at this time the PSNR were 38.7651dB and 38.8879dB. This experiment from the scene of the complexity and the distribution of the light source and the strength of the virtual viewpoint synthesis results of the influence of the scene, we obtained the results of different scenes under different conditions, the distribution of light sources and the intensity of the virtual viewpoint synthesis. The experimental results of this algorithm in different scenes are obtained. From the above results, it can be seen that in any scene, the algorithm has a substantial improvement over the VSRS synthesis effect, and has better robustness. In the experimental process, we also set the variables such as the selection of the block size and the number of images used to obtain the mean image during the fitting, so that we can get better experimental verification results.



(a)Block size 1×1



(b)Block size 2×2



(c)Block size 4×4

Figure 5. Compare the results of the reference image

4.3. Visual Emotional Presentation of Scene Integration

4.3.1. Emotional emotions in the fusion of virtual scenes and 3D characters

From the perspective of emotion-cognition-evaluation theory, the five elements of emotion are subjective experience, physiological arousal, expression, behavioral tendency, and evaluation, and "evaluation" is both the cause of emotion arousal and the basic component of emotion. The initial appraisal of external stimuli induces an emotional experience, while the process of reappraisal causes a change in emotional activity. Emotions are perceived in three dimensions: pleasantness, arousal, and dominance. Pleasantness is commonly referred to as whether an emotion is positive or negative in the moment, and ranges from unpleasant to pleasant. Arousal degree is the degree to which the emotion is aroused, if the emotion is not aroused, it is a calm state, ranging from excitement to calmness, and dominance degree refers to whether the learner can control and dominate the emotion after it occurs.

To summarize, based on the research and sorting out of emotions by many scholars: emotion refers to an individual's evaluation of the world, a reaction to interact with a specific stimulus, and in this study emotion is defined as the reaction that elicits the learner's interaction with the current environment in the environment of the virtual scene. The elements that induce positive emotions in this study refer only to visual 3D animation elements.

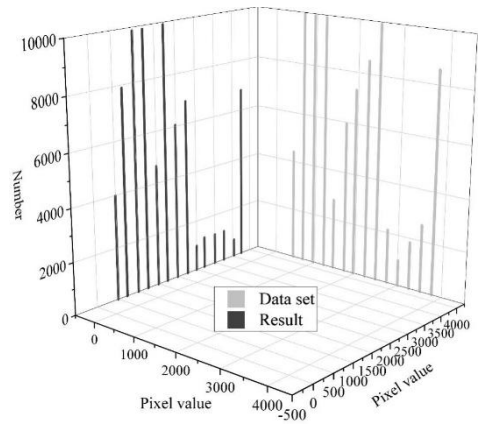
4.3.2. Virtual scenes

Through the above research related to emotional design of multimedia learning materials, it is found that the concept of emotional design was discovered and studied in human engineering a long time ago

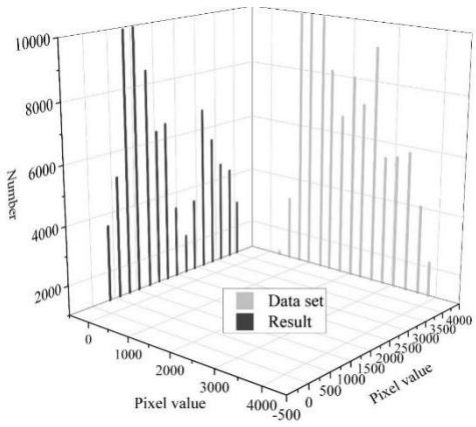
and has a long history, but the research of emotional design in the field of education is still in the primary stage. Therefore, in the study, the emotional design of virtual scenes is defined as the design of elements in the virtual scene environment that may affect the emotional changes of the learner, that is, the design of the virtual scene itself, with the purpose of enhancing the user's positive emotional level. Based on the above definition of virtual scene integration, the emotional design of visual elements of 3D animated characters of virtual scenes is defined in the study as the design of visual elements in the virtual scene environment that may affect users' emotional changes, that is, the design of the virtual scene itself, to achieve the purpose of inducing users' positive emotions. Emotional design is an effective way to promote the effect of 3D animation presentation and emotional enhancement, some researchers meta-analysis of the literature on emotional design, from the meta-analysis results found to play a positive role in the user's mood and presentation effect of this type of emotional design has a common feature.

4.3.3. Analysis of virtual scene fusion effects

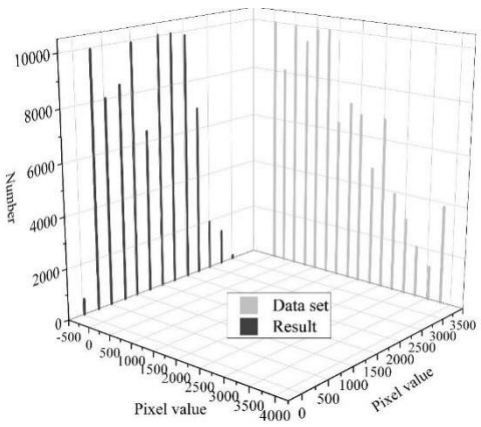
Related research proposes a scene fusion emotion presentation method based on color histogram, in order to evaluate the optimization results of this paper's algorithm with the scene fusion emotions, this paper adopts the histogram to compare the similarity between the fusion results and the target emotion dataset, and to judge the similarity between the fusion results' color theme and the color theme of the target emotion image dataset, to validate the correctness and effectiveness of this paper's algorithm. As Figure 6 shows the color histograms of the fusion emotion presentation results of the five scenes of this paper's algorithm with its emotion image dataset, Figure (a) is pleasure, Figure (b) is horror, Figure (c) is calmness, Figure (d) is hesitation, and Figure (e) is romance. From the figure, it can be seen that the scene fusion emotion results presented by the algorithm in this paper are similar to the results of the target dataset, and the optimized results presented range from 2293 to 10000 for the horror scene, and the target emotion ranges from 1100 to 10000, for example.



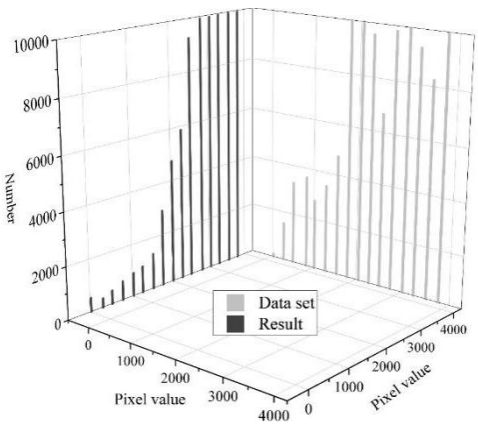
(a)Pleasure



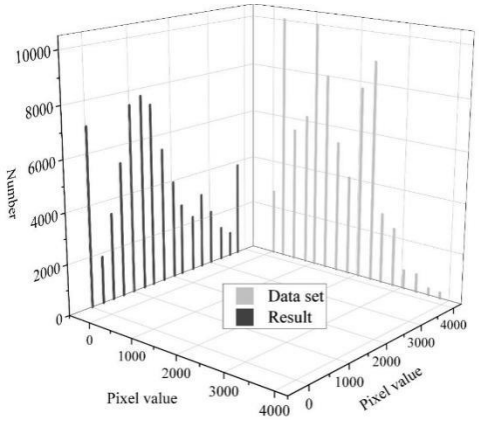
(b)Horror



(c)Peace



(d)Hesitation



(e)Romance

Figure 6. Optimization results and target emotion histogram

5. Conclusion

In this paper, ray tracing algorithm is utilized to design the change of viewpoint, object, and light source of 3D animated characters and calculate the pixel unbiasedness. Aiming at the scattering characteristics of the hair surface of the 3D animated character, the Fresnel formula is obtained by calculation, which is used to represent the hair contour light rendering effect of the 3D character. The hair self-shadowing effect is simulated, and the virtual scene is constructed to realize the fusion of the virtual scene and the animated character. Through simulation experiments, the performance of the hair rendering and simulation module is analyzed separately, and the Kajiya-Kay light model used in this paper has a rendering frame rate average of 62.04, which is better than other models, so the Kajiya-Kay light model in this paper can achieve a relatively better efficiency. Objective and subjective experiments are performed on the fused images of virtual scenes. In the simple scene, the PSNR values of image quality of means_40 under 1×1 chunk size for overexposure, uniform brightness, unilateral light source, and symmetric light source are 37.8486, 44.7349, 43.5798, and 45.9648, respectively. 40-60 images were elected to be used to find out the mean image, and the 2×2 chunk fitting method can be used to obtain better results with a smaller experimental complexity. Better experimental results are obtained, at which time the PSNR is 38.7651 dB and 38.8879 dB, respectively.

References

1. Papaefthymiou, M., Kateros, S., Georgiou, S., Lydatakis, N., Zikas, P., Bachlitzanakis, V., & Papagiannakis, G. (2017). Gamified AR/VR character rendering and animation-enabling technologies. Mixed Reality and Gamification for Cultural Heritage, 333-357.
2. Jeon, J. (2023). Current State of Animation Industry and Technology Trends-Focusing on Artificial Intelligence and Real-Time Rendering. The Journal of the Convergence on Culture Technology, 9(5), 821-830.
3. Hiranyachattada, T., & Kusirirat, K. (2020). Using Mobile Augmented Reality to Enhancing Students' Conceptual Understanding of Physically-Based Rendering in 3D Animation. European Journal of Science and Mathematics Education, 8(1), 1-5.
4. Huang, Z., Gong, G., & Han, L. (2014). Physically-based modeling, simulation and rendering of fire for computer animation. Multimedia tools and applications, 71, 1283-1309.
5. Luciano, G. (2019). Essential computer graphics techniques for modeling, animating, and rendering biomolecules and cells: a guide for the scientist and artist. Crc Press.
6. Zheng, J. (2024, July). Computer 3D Animation Rendering Technology Based on Programmable Processor. In 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS) (pp. 1352-1356). IEEE.

7. Laparra, V., Berardino, A., Ballé, J., & Simoncelli, E. P. (2017). Perceptually optimized image rendering. JOSA A, 34(9), 1511-1525.
8. Chang, Y., & Guo-Ping, W. A. N. G. (2019). A review on image-based rendering. Virtual Reality & Intelligent Hardware, 1(1), 39-54.
9. Zeng, B., Liu, B., Li, H., Liu, X., Liu, J., Chen, D., ... & Zhang, B. (2022). FNeVR: Neural volume rendering for face animation. Advances in Neural Information Processing Systems, 35, 22451-22462.
10. Yulia, A. F., Bintoro, P., Andini, D. Y. A., & Triloka, J. (2024). Comparison of EEVEE and Cycles Rendering Performance in Blender 3.5 in the Context of Interactive Visuals for 3D Animation. Jurnal Penelitian Pendidikan IPA, 10(7), 4086-4091.
11. Goswami, P. (2021). A survey of modeling, rendering and animation of clouds in computer graphics. The Visual Computer, 37(7), 1931-1948.
12. Sahputra, E., & Sucahyo, M. H. (2022). Analysis of Eevee engine rendering engineering in making 3D animation videos Mukomuko hospital. Jurnal Komputer, Informasi dan Teknologi, 2(2), 229-238.
13. Bühler, P. (2021). 3D mit Blender: Modeling-Animation-Rendering. Springer Vieweg.
14. Sari, A., Usman, A., & Handoko, D. (2023). Comparison Analysis of 3D Animation Rendering With Cycles Methods and Workbench in Blender. INTERNATIONAL JOURNAL OF DATA SCIENCE AND VISUALIZATION (IJDSV), 1(1).
15. Zhang, Y., Su, R., Yu, J., & Li, R. (2024). 3D facial modeling, animation, and rendering for digital humans: A survey. Neurocomputing, 598, 128168.
16. Eid, M., De Cecco, C. N., Nance Jr, J. W., Caruso, D., Albrecht, M. H., Spandorfer, A. J., ... & Schoepf, U. J. (2017). Cinematic rendering in CT: a novel, lifelike 3D visualization technique. American Journal of Roentgenology, 209(2), 370-379.
17. An, S., & Tian, M. (2023). Based on the Aesthetics of Film and Television: Animation Scene Design. Probe-Media and Communication Studies, 5(4).
18. Liu, J. (2021, October). Research on the design and production of 3D animation. In 2nd International Conference on Computer Vision, Image, and Deep Learning (Vol. 11911, pp. 54-59). SPIE.
19. Shati, A. A. (2024). Artistic treatment of animation scenes in feature films. Al-Academy, (111), 71-90.
20. Van Rooij, M. (2019). Carefully constructed yet curiously real: How major American animation studios generate empathy through a shared style of character design. Animation, 14(3), 191-206.
21. Jeong-Shick, Y. (2023). Unity: A Powerful Tool for 3D Computer Animation Production. Journal of the Korea Computer Graphics Society, 29(3), 45-57.
22. Yu, G., & Ma, C. (2022). Analysis of Scene Design in 3D Animation from the Perspective of Digital Media Art Design Psychology. Psychiatria Danubina, 34(suppl 1), 187-188.
23. Wang, J., & Gao, J. (2023). Research on 3D Animation Technology of Human and Scene Based on Virtual Reality. Art and Performance Letters, 4(1), 12-17.
24. Yang, L. (2021, February). Functionality and artistry in 3d animation scene design. In Journal of Physics: Conference Series (Vol. 1744, No. 4, p. 042212). IOP Publishing.
25. Wu, Z. (2023). On the Expression and Application of Color in the Scene Design of Animated Films. Academic Journal of Humanities & Social Sciences, 6(4), 33-38.
26. Shengchun Wang,Hao Wang,Yunlai Zhou,Junbo Liu,Peng Dai,Xinyu Du & Magd Abdel Wahab. (2021). Automatic laser profile recognition and fast tracking for structured light measurement using deep learning and template matching. Measurement108362-.
27. Ting Xiao,Xing xing Yao & Yuan Zheng. (2025). See More from Reflection with Fresnel's Equations. The Physics Teacher(1),56-59.

28. Jie Wu,Rui Xu,Runhui Huang & Xuezhi Hong. (2024). Data-Efficient Training of Gaussian Process Regression Models for Indoor Visible Light Positioning. Sensors(24),8027-8027.
29. Geyi Su,Fei Deng,Cunjin Sun & Mingxu Su. (2025). A study on hybrid light extinction model and inversion based on Lambert-Beer Law. Optics Communications131342-131342.