Real-time Rendering and Editing of Scattering Effects for Translucent Objects

Rui Wang  Wei Hua  Yuchi Huo  Hujun Bao

Abstract

The photorealistic rendering of the transparent effect of translucent objects is a hot research topic in recent years. A real-time photorealistic rendering and material dynamic editing method for the diffuse scattering effect of translucent objects is proposed based on the bidirectional surface scattering reflectance function’s (BSSRDF) Dipole approximation. The diffuse scattering material function in the Dipole approximation is decomposed into the product form of the shape-related function and the translucent material-related function through principal component analysis; using this decomposition representation, under the real-time photorealistic rendering framework of pre-radiative transmission and the scattering transmission to realize real-time editing of translucent object materials under various light sources. In addition, a method for quadratic wavelet compression of precomputed radiative transfer data in the spatial domain is also proposed. Using the correlation of surface points in the spatial distribution position, on the premise of ensuring the rendering quality, the data is greatly compressed and the rendering is efficiently improved. The experimental results show that the method in this paper can generate a highly realistic translucent effect and ensure the real-time rendering speed.

Photorealistic rendering of translucent objects is very important for computer graphics, because many objects in our daily life have translucent properties, such as marble, skin, cheese, bread, snow, etc. Translucency is a special optical effect formed by multiple scattering and partial absorption after light hits the surface of an object. Due to the different degrees of light scattering and absorption, different substances present different translucent visual effects. With the development of computer application requirements, in many applications, such as games, film special effects, industrial design and other fields, there is an increasing demand for quickly generating highly realistic images of translucent objects. However, due to the complexity of the light scattering process, it is very challenging to use a computer to realize the real-time realistic rendering of the translucent effect. In recent years, a series of fruitful researches have been carried out in the domain, ranging from the direct calculation of physics-based subsurface scattering models, to the use of various approximate acceleration algorithms [19, 18, 7, 9], and the introduction of pre-computation into the rendering of translucent objects. Real-time rendering of translucent effects [10, 30]. However, there is still a lack of effective methods for real-time editing of translucent materials of objects, especially in multiple light source environments and variable viewing angles, to achieve real-time realistic rendering of translucent objects with changing materials.

In view of this situation, this paper proposes a method to change the translucent material in real time under a variety of light sources (ambient light, point light source, directional light, area light source, etc.) based on the bidirectional surface scattering reflectance function (bidirectional surface scattering reflectance function). Dipole approximation of surface scattering reflectance function (BSSRDF). Under the framework of precomputed radiative transfer, by decomposing the diffuse scattering material function and using the method of quadratic wavelet compression on the radiative transfer data in the spatial domain, the real-time rendering of highly realistic translucent effects and the translucent material are realized. real-time editing.

1. Related Work

Translucent Objects Rendering. Translucent effects a form of reflection of light of real world objects, which results from the scattering of light beneath the surface of an object. It can be described by BSSRDF in computer graphics to describe the light transmission that creates a translucent effect. However, since BSSRDF only describes the relationship between the shooting person and the light emitted from the translucent object, and does not correspond directly to a physical quantity that describes the effect of translucency (such as to light scatter, absorptivity, etc.). Therefore, for a long period of time, the rendering of translucent objects requires complex calculations to simulate the transmission of light in materials with specific scatter and absorptivity [18, 7, 9], and the rendering efficiency is low. Until Jensen et al. [19] expressed BSSRDF as an analytical form defined by physical parameters such as scattering degree and absorptivity through Dipole approximation, the rendering speed for semi-transparent objects was
greatly accelerated [17]. Even so, the rendering of translucent effects is still far from meeting the needs of real-time applications. On the basis of the Dipole method, Hao et al. [10] and Wang et al. [30] speed up the rendering of translucent effect achieves interactive and real-time rendering respectively. However, these algorithms need to know the translucent material parameters of the object during preprocessing, and cannot change the translucent material in real time when rendering. Xu et al. [34] further proposed a method to realize real-time editing of semi-transparent materials, but it needs to limit the drawing viewpoint or light source direction. In addition, the method in this paper also has greater advantages in the error control of the calculation and the storage capacity of the pre-calculated data.

Precomputed Radiance Transfer. The precomputed radiance transfer method is a real-time computing framework used to solve the realistic rendering of objects under complex light sources in recent years. It was first proposed by Sloan et al. [26]. The basic idea of this method is to decompose the function in the rendering equation into a specified set of basis functions, and use the linear combination of these basis functions to approximate the original function. When plotting in real time, take advantage of the good properties of integration between specified basis functions (for example, the orthogonality between spherical harmonic basis functions [26] and wavelet basis functions [23] or between certain spherical radial basis functions [28] that are easy for integrals) to speed up the integral operation of drawing equations. The computational modification transfer method is a hot research topic in recent years: from the beginning, it only supports ambient light, and extends to support local and regional light sources [37], BRDF, to gradually support the editing and modification of dynamic BRDF [3, 27]. The method in this paper takes advantage of the precomputed radiance transport framework and extends it to real-time editing of translucent object materials.

Spatial-temporal compression for Precomputed Radiance Transfer. The predecessors mainly used clustered principal component analysis [25], clustered tensor approximation [28] and other methods to compress the precomputed radiance transmission data. The disadvantages of using principal component analysis methods are that the calculation speed is slow and the stability of numerical calculation is more sensitive. We propose a method to further compress the data by transforming the 3D object surface parameters into a specific parameter domain and performing a wavelet transform on it. This method has faster calculation speed and better compression ratio.

2. Translucent material function approximation based on PCA

2.1. Approximating BSSRDF via Dipole

According to the Dipole rendering for translucent objects like [19], the rendering equation for the scattering effect of a translucent object is

$$L_i(x_o, \omega_o) = \int_A \int_\Omega S(x_i, \omega_i; x_o, \omega_o) \cdot L_i(x_i, \omega_i)(n \cdot \omega_i) d\omega_i dA(x_i), \quad (1)$$

where $L_i(x_o, \omega_o)$ is the outgoing ray on $X_o$ along $\omega_o$, $S(x_i, \omega_i, X_o, \omega_o)$ is BSSRDF, and $L_i(x_i, \omega_i)$ is the incident light energy of the point $x_i$ along direction $\omega_i$ near $x_o$. $(n \cdot \omega_i)$ reflects the incident angle of the incident light, $\Omega$ is the unit sphere defined by the above light energy and reflective function, and $A$ is the area of the object surface. $S(x_i, \omega_i; x_o, \omega_o)$ describes the relationship between the luminous flux $F_i(x, \omega_i)$ and $L_i(x_o, \omega_o)$ as shown in Figure 1 (b). $S(x_i, \omega_i, X_o, \omega_o)$ is usually expressed as the synthesis of single scattering effects and diffuse scattering, respectively, namely

$$S(x_i, \omega_i; x_o, \omega_o) = S^{(1)}(x_i, \omega_i; x_o, \omega_o) + S_d(x_i, \omega_i; x_o, \omega_o). \quad (2)$$

Generally, the translucent effect of solid matter in nature is mainly caused by diffuse scattering, while liquid translucent matter is mainly manifested as single scattering [6]. For objects with diffuse scattering and uniform material, Jensen et al. [19] proposed a Dipole approximation based on diffuse scattering theory, as shown in Figure 1 (C). The Dipole approximation assumes that the object has infinite depth and infinite surface. After the ray hits the plane point $x_i$, the diffuse scattering of the ray on the plane can be approximated as the effect of direct illumination of the plane by the two virtual light sources (Dipole points) directly above and below the surface $x_i$. The position of the two virtual light sources relative to the surface of the object is determined by the material function of the semi-transparent object, that is,

$$S(x_i, \omega_i; x_o, \omega_o; \sigma) = \frac{1}{\pi} F_i(\eta, \omega_i) R(\|x_i - x_o\|, \sigma) F_i(\eta, \omega_o), \quad (3)$$

where $F_i$ is the Fresnel transmission coefficient at point $x_i$ and $x_o$; $R(\|x_o - x_i\|, \sigma)$ is the diffuse scattering function, which can be approximated by Dipole as the analytical form related to the material coefficient of the translucent object

$$R_d(r, \sigma) = \frac{a'}{4\pi} \left[ r (\sigma + \frac{1}{d_r}) e^{-\sigma v \frac{d_r}{d_v}} + z_v (\sigma + \frac{1}{d_v}) e^{-\sigma v \frac{d_v}{d_r}} \right], \quad (4)$$
where $r = \| x_o - x_i \|$ is the distance between $x_i$ and $x_o$; $\sigma' = (1 - g)\sigma_a$ and $\sigma' = \sigma_a + \sigma'_s$ is the scattering attenuation and scattering decay coefficients; $\sigma_a$ is the absorption coefficient; $\sigma = (\sigma'_s, \sigma_a)$ is the material coefficient of the scattering function $R_d(r, \sigma)$; $d' = \frac{z'}{r}$ is the decay ratio; $g$ is the average of scattering cosine term; $\sigma_{tr} = \sqrt{3\sigma'\sigma'_s}$ is the valid decay coefficient; $z_r = \frac{1}{\sigma'_s}$ and $z_v = z_r(1 + 3\frac{\eta}{4})$ are the distance between two Dipole to the surface, where $A = \frac{1 + F_{dr}}{r^2}, F_{dr}$ is the approximated Fresnel term, $F_{dr} = \frac{-1.440 + 0.718}{\eta} + 0.668 + 0.0636\eta; d_r = \sqrt{r^2 + z_r^2}$ and $d_v = \sqrt{r^2 + z_v^2}$ are the distances from $x$ to two Dipole points. The importance of the Dipole approximation is that it gives an analytical representation of the surface scattering effect. Based on this approximation, this paper mainly discusses the real-time photorealistic rendering and real-time material editing for the diffuse scattering effect of translucent materials under the Dipole approximation.

### 2.2. Diffuse Scattering Material Function Decomposition and Calculation

For static scenes, $r$ is known in advance, while $\sigma$ needs dynamic changes to realize real-time editing of materials. If one simply sample the value space of $\sigma$, it will bring a huge amount of data, which cannot be practically operated. Therefore, we denote $R_d(r, \sigma)$ as the factorial division of $r$ with the $\sigma$, $R_d(r, \sigma) \approx \sum_{k=1}^{K} S_k(\sigma) b_k(r)$.

Note that $r(x_i)$ is a one-variable function with a value range of $[0, +\infty)$. According to formula 4, $R_d(r, \sigma)$ function has a negative exponential form. When $r$ is large, its value is approximately 0. In addition, although the value range of $\sigma$ is $(0, +\infty)$, when $\sigma$ becomes large, the assumption of Dipole approximation will no longer be suitable, and BSSRDF can be considered to degenerate into BRDF. This means that the value of the $\sigma$ variable should be limited within a certain area. We can obtain the decomposed form of the diffuse scatter function by performing a principal component analysis on the matrix $M$, that is,

\[ M = USV. \]  

According to the nature of pivot analysis, the column vector of $V$ matrix is the discrete representation of basis function $b_k(r)$. Although the $U$ matrix also corresponds to a set of orthogonal functions of $s_k(\sigma)$, in real-time computing, for any given $\sigma$ value, $R_d(r, \sigma)$ can be projected onto the basis function $b_k(r)$ to find the corresponding coefficient $s_k(\sigma)$.

Figure 2 shows the value distribution of the first four basis functions $b_1(r), b_2(r), b_3(r)$ and $b_4(r)$ decomposed from the matrix $M$. The right side shows the values of the first 20 eigenvalues arranged from high to low. It can be found that the eigenvalues after No. 10 are very close to 0. In practical applications, the number of basis functions we choose is generally 12. Figure 3 shows the visualization results of the translucent object after approximating the diffuse scattering function $R_d(r, \sigma)$ with different numbers of basis functions. It can be seen that when 12 basis functions are used to approximate $R(r, \sigma)$, its Visual errors are hard to detect. It can be seen from Figure 4 (a) and 4 (b) that the diffuse scattering material function $R(r, \sigma)$ can be well approximated by using 12 basis functions decomposed by PCA.

Compared with Xu et al [34] of approximating $R(r, \sigma)$ by selecting piece-wise polynomial function, the method in this paper has the following advantages: 1) In the case of the same number of basis functions, the approximation error of the method in this paper is smaller. For example, when there are 15 basis functions, the $L^2$ error of the method in [34] is 0.0005, while our error is less than 0.0001. 2) The basis function selected according to the size of the eigenvalue can easily realize the approximation error. However, the method in [34] did not give the relationship between the number of basis functions and the approximation error using a piece-wise polynomial approximation. 3) The basis functions obtained by the principal component analysis are orthogonal, and the real-time calculation of the coefficient $s_k(\sigma)$ is faster. However, in the method in the literature [34], due to the non-orthogonality of the slice function, a linear system of equations needs to be solved.
Figure 2. Basis function value and eigenvalue distribution. Left: the basis functions of the first four largest eigenvalues. Right: eigenvalue distribution.

Figure 3. Visualization of approximation using different number basis functions.

3. Precomputing and Compression of Scatter Transmission

We substitute the decomposed form of \( R(r, \sigma) \) into the translucent object rendering equation and get

\[
L_o(x_o, \omega_o) = \frac{1}{\pi} F_i(\eta, \omega_o) \int_A \int_{2\pi} F_i(\eta, \omega_i) R_o(r, \sigma) \\
\times L_i(x_i, \omega_i)(n \cdot \omega_i)d\omega_i dA(x_i) \approx \frac{1}{\pi} F_i(\eta, \omega_o) \sum_k s_k(\sigma) \int_A E(x_i) b_k(r(x_i)) dA(x_i) = \frac{1}{\pi} F_i(\eta, \omega_o) \sum_k s_k(\sigma) \cdot B_k(x_i, r(x_i)). \tag{6}
\]

\[
B_k(x_i, r(x_i)) = \int_A E(x_i) b_k(r(x_i)) dA(x_i).
\]

\[
E(x_i) = \int_{2\pi} L_i(x_i, \omega_i)(n \cdot \omega_i)d\omega_i.
\]

It can be seen that the calculation of the outgoing ray can be done in 2 steps: 1) Calculate outgoing ray \( E(x_i) \) at point \( x_i \) on the lower object surface. 2) Calculate \( L_o \) according to the outgoing ray \( E(x_i) \) of \( x_i \). We use the precomputed radiative transfer framework to accelerate the rendering, in which we rewrite Equation 6 into matrix form following the method of [26] as

\[
L_o = \sum_k s_k(\sigma) \cdot (T_k \cdot E) = \sum_k s_k(\sigma) \cdot v_k, \tag{7}
\]

where \( T_k \) is a scattering transportation matrix describes \( b_k(||x_i - x_o||) \). This matrix describes the scattering and transmission of light by the whole object, the row vector corresponds to the dimension of light transmission, and the column vector corresponds to the spatial dimension of the surface point of the object; \( E \) is the outgoing ray vector representing \( E(X_i); v_k = T_k \cdot E \).

3.1. Scattering Transmission Compression

Following wavelet-based precomputed radiance transfer methods [23], we project the outgoing ray vector \( E \) and the scattering transfer matrix \( T \) to a basis

\[
v_k = (T_k \cdot w^{-1}) \cdot (w \cdot E) = T_k^w \cdot w E, \tag{8}
\]

where \( w \) is the orthogonal wavelet projection matrix \( w^{-1} \), \( w = I; T_k^w \) and \( w E \) are the scattering transmission coefficient and outgoing ray coefficient after wavelet projection, respectively. Since the scattered transmission and outgoing rays are both defined on the surface of the object, we use the tree-based hierarchical method proposed by Hasan et al. [11] to parameterize the surface points distributed in the 3D space to a regular 2D wavelet domain: \( M \times 2^n \times 2^n \) sampling points are evenly distributed on the top of the sample; secondly, according to their distribution in space, the unstructured points on these distributions are divided into \( m \) parts, and for each division, the hierarchical method proposed by Hasan is used to construct a balanced quadtree to ensure the uniformity of division; finally, each quadtree is flattened, that is, a regular 2D wavelet domain can be obtained.
3.2. Spatial Compression

In addition to the fact that the scatter transmission received at each point on the surface of the object can itself be compressed, the scatter transmission between adjacent points in space is also correlated. Therefore, we can further compress in the airspace. Equation 8 can be further expressed as

\[ v_k = w_{1}^{-1} \cdot (w_{1} \cdot T_{k} \cdot w_{0}^{-1}) \cdot (w_{0} \cdot E) = w_{1}^{-1} \cdot (w_{1} T_{k}^{w_{0}^{-1}} \cdot w_{0} E) \]  

(9)

Since the wavelet transform is linear, we get

\[ L_{0} = w_{1}^{-1} \cdot \sum_{k} s_{k}(\sigma) \cdot (w_{1} T_{k}^{w_{0}^{-1}} \cdot w_{0} E) \]  

(10)

where \( w_{1} T_{k}^{w_{0}^{-1}} \) indicates that a scattering transmission matrix performs an inverse wavelet transform \( w_{0}^{-1} \) and a wavelet transform \( w_{1} \) at two levels of light transmission and spatial domain, respectively. The wavelet inverse transformation represented by \( w_{1}^{-1} \) in formula 10 only needs to be completed once in the last step of rendering, and what is realized is the reconstruction of the surface color of the object.

Since the amount of precomputed data is huge (several gigabytes), we’ll do \( o \) first. The transformed wavelet coefficients are subjected to nonlinear compression, and then the compressed vector is subjected to \( w_{1} \) transform. Using 2-step progressive compression can greatly reduce the precomputation time and preserve the high-frequency effects of the signal as accurately as possible.

An important significance of airspace compression is that after using spatial compression, the included scattering effect does not change greatly with the change of the number of object grid points, and has roughly the same entropy value. Spatial compression can be regarded as the entropy value approximated by compression, ignoring the subtle changes in the surface scattering signal caused by the increase of surface mesh points.

In conclusion, our pre-calculation process includes two steps:

Step 1. Traverse every vertex of the object and every basis that approximates \( R_{d}(r, \sigma) \) function term \( b_{k}(r) \), computes the scattering transfer matrix, and will perform a wavelet transformation, keeping the \( n \) item with the largest coefficient stored.

Step 2. Perform airspace compression. For each basis function for all vertices in \( b_{k}(r) \) wavelet transforms and compresses the wavelet coefficients retained on the number of terms. In compression, a threshold energy is predetermined, and certain wavelet coefficients are reserved so that the reserved energy exceeds the threshold.

4. Real-time Rendering of Translucent Effects

Since the method in this paper does not depend on the form of direct lighting calculation \( E \), it can integrate all the lighting forms that meet the real-time calculation into the above-mentioned framework, such as point light source, directional light, local light source, ambient light and so on. We implement the above models separately and further accelerate rendering computations for some lighting models.

We use the standard Shadow Mapping algorithm to deal with the situation of point light source and directional light, organize the position and normal direction of the surface sampling point as a texture, and calculate the radiation transfer value affected by the shadow in the shader of the graphics card; Then read the data from the graphics card GPU back to the memory, and project it onto the wavelet basis to get \( w_{0} E \).

Using standard wavelet-based precomputed radiance for ambient light [23], the transmission method calculates outgoing rays at each sampling point as

\[ E(x_{i}) = \int_{2\pi} L_{i}(\omega_{i})(V(x_{i}, \omega_{i})F_{i}(\omega_{i})(n \cdot \omega_{i}))d\omega_{i}. \]  

(11)

Express this equation as a matrix, and take the ambient light \( L_{i}(w_{i}) \) after wavelet transform, we can get

\[ E = V \cdot L_{i} = (V \cdot w_{2}^{-1}) \cdot (w_{2} \cdot L_{i}) = V^{w_{2}^{-1}} \cdot w_{2} L_{i}, \]  

(12)
where \( V \) is the visibility vector at each sample point matrix, which can be obtained by precomputing. Due to the scattering transmission compression, the sample points of the surface have been organized in 2D domain, and after substituting equation it into equation 8, we can further do wavelet transform as

\[
w_0 E = w_o \cdot (V \cdot w_2^{-1}) \cdot (w_2 \cdot L_i) = w_o \cdot V \cdot w_2^{-1} \cdot w_2 L_i, \quad (13)
\]

then we get

\[
L_o = w_1^{-1} \cdot \sum_k s_k(\sigma) \cdot (w_1 T_k^{-w_2^{-1}} \cdot w_o E) = \sum_k s_k(\sigma) \cdot (w_1 T_k^{-w_2^{-1}} \cdot w_o V \cdot w_2^{-1} \cdot w_2 L_i) = \quad (14)
\]

\[
w_1^{-1} \cdot \sum_k s_k(\sigma) \cdot (w_1 T_k^{-w_2^{-1}} \cdot w_o V \cdot w_2^{-1} \cdot w_2 L_i),
\]

where \( w_1 T_k^{-w_2^{-1}} = w_1 T_k^{-w_2^{-1}} \cdot w_o V \cdot w_2^{-1} \).

Therefore, we can integrate the pre-computed visibility information into the spatial compression. After precomputing the radiation transfer matrix \( w_1 T_k^{-w_2^{-1}} \) including the visibility information and the scattering transmission information, spatially compressing can reduce the storage cost of pre-computed visibility and further improve the speed of real-time rendering.

In order to be able to deal with the situation of the local light source, we introduce the idea of shadow field proposed by Zhou et al. [37]. By calculating the radiation divergence field of the light source, the situation of the local light source is integrated into the processing of the ambient light, so as to realize the lower half of the local light source. Real-time rendering of transparent objects.

The steps of the rendering algorithm are as follows:

Step 1. Calculate the direct illumination \( E \) and project it onto the wavelet base to get \( w_o E \).

Step 2. For each \( k \) calculate the scattering transfer matrix \( w_1 T_k^{-w_2^{-1}} \), and its product with \( w_o E \), \( w_1 V_k = w_1 T_k^{-w_2^{-1}} \cdot w_o E \).

Step 3. Real-time projection of diffuse scattering material functions onto decomposed basis functions, calculate the material parameter \( s_k(\sigma) \).

Step 4. Calculate the sum of all \( w_1 V_k \) with \( s_k(\sigma) \) as the weight.

Step 5. Do the inverse transform \( w_1^{-1} \) of \( w_1 \) to get the object surface outgoing ray \( L_o \) of the grid point.

It can be seen that if the user needs to adjust the scattering material, he only needs to update Step 3 and Step 4, and the recalculation of the material of the translucent object can be completed quickly to meet the requirements of real-time calculation.

5. Results

5.1. Compression of Scatter Transmission

In the compression of the first step, we use the Haar wavelet basis; since the Haar wavelet will produce obvious drawing errors when compressing the spatial domain, in the compression of the second step, a smoother and higher-order wavelet is selected. We find that the Daub echos 9/7 wavelet can meet the requirements well, and even at a very high compression rate, it will not produce too much rendering error.

Figure 5 shows the precomputed data and compression results for several different models. The uncompressed original data is very huge, the amount of data is 35 GB 1.1 TB. After the first step of compression, retaining the wavelet coefficient of about 1% can already preserve the effect of light scattering with high quality. The second step is to perform spatial compression: due to the small amount of data in the Angel model, we did not perform spatial compression on it; through experiments on Beethoven and Candle models, we found that the compression ratio of about 1:4 is preserved, and the The energy of 95% – 98% has little effect on the rendering quality; and the compressed data can be put into the memory, which meets the needs of real-time rendering. Figure 6 shows the visual effects under different spatial compression ratios. The numbers in the title of the sub-plots represent: the ratio of energy saving, the compression ratio, and the frame rate of drawing (frame/s). It can be seen that with the increase of the compression ratio and the decrease of the retained energy, the control error will become more and more obvious; and as more and more high-frequency components are discarded, the rendering results become blurred. Such as the shadows on the neck and cheeks of the Beethoven model.

5.2. Real-time Photorealistic Rendering and Editing of Translucent Materials

We have developed a system that can easily realize real-time photorealistic rendering and editing of different mesh models, translucent materials, and materials under multiple light source environments, as shown in Figure 7. Figure 8 shows the rendering result of the material change of the Angel model under ambient light. Figure 9 shows the result of material changes for a candle under a local light source (center flame) (drawing of semi-transparent objects is limited to candles, and the center flame is only a representation). Figure 10 shows the translucent effect of the candle model when the light source changes. It can be seen from these results that the method in this paper can obtain high-quality semi-transparent rendering effects with a high degree of realism. Figure 11 shows the drawing frame rate of the method in this paper. It can be seen that the method in this paper can achieve real-time speed whether it is the ren-
dering of semi-transparent objects or the editing of transparent material coefficients.

6. Conclusion

This paper discusses the technology of real-time photorealistic rendering and material editing based on Dipole approximation for diffuse scattering effect of semi-transparent objects. Firstly, a decomposition approximation method of diffuse scattering material function based on principal element analysis is proposed, and based on this method, the pre-computing and compression methods of scattering transmission are further studied, and the high-fidelity rendering and real-time material editing of semi-transparent objects are realized. The innovation of this method lies in: 1) This paper proposes a method to obtain the optimal approximation of diffuse scattering function through principal element analysis. Compared with other forms of basis functions, the basis functions decomposed by principal component analysis have better properties. Under the same approximation error requirements, fewer basis functions are required and the drawing efficiency is higher. 2) A wavelet-based spatial domain compression method is proposed. Different from the traditional clustering principal component analysis method, the wavelet-based compression method can better preserve the high-frequency information of the signal, and the calculation is simple while maintaining high-quality rendering. At the same time, it greatly reduces the amount of data and improves the drawing speed. It is believed that the method in this paper has application value in the fields of industrial design and the generation of high-fidelity images.

There are some extended articles about applying physical lighting computation in various applications:

1. Deep Learning-Based Monte Carlo Noise Reduction

By training a neural network denoiser through offline learning, it can filter noisy Monte Carlo rendering results into high-quality smooth output, greatly improving physics-based Availability of rendering techniques [16], common research includes predicting a filtering kernel based on g-buffer [2], using GAN to generate more realistic filtering results [33], and analyzing path space features Perform manifold contrastive learning to enhance the rendering effect of reflections [4], use weight sharing to quickly predict the rendering kernel to speed up reconstruction [8], filter and reconstruct high-dimensional incident radiation fields for unbiased reconstruction rendering guide [15], etc.

2. The multi-light rendering framework is an important rendering framework outside the path tracing algorithm. Its basic idea is to simplify the simulation of the complete light path illumination transmission after multiple refraction and reflection to calculate the direct illumination from many virtual light sources, and provide a unified Mathematical framework to speed up this operation [5], including how to efficiently process virtual point lights and geometric data in external memory [29], how to efficiently integrate virtual point lights using sparse matrices and compressed sensing [14], and how to handle virtual line light data in translucent media [13], use spherical Gaussian virtual point lights to approximate indirect reflections on glossy surfaces [12], and more.

3. Automatic optimization of rendering pipelines Apply high-quality rendering technology to real-time rendering applications by optimizing rendering pipelines. The research contents include automatic optimization based on quality and speed [31], automatic optimization for energy saving [32, 35], LOD optimization for terrain data [22], automatic optimization and fitting of pipeline rendering signals [21], anti-aliasing [36], etc.

4. Using physically-based process to guide the generation of data for single image reflection removal [20]; propagating local image features in a hypergraph for image retrieval [1]; managing 3D assets in a block chain-based distributed system [24].

References

[1] G. An, Y. Huo, and S.-E. Yoon. Hypergraph propagation and community selection for objects retrieval. Advances in Neural Information Processing Systems, 34, 2021.

[2] S. Bako, T. Vogels, B. McWilliams, M. Meyer, J. Novák, A. Harvill, P. Sen, T. Derose, and F. Roussele. Kernel-predicting convolutional networks for
Figure 6. Spatial compression comparison.

Figure 7. Real-time photorealistic rendering and editing system for translucent materials.

Figure 8. Angel model in ambient light.

Figure 9. The candle translucent material under local lighting.

Figure 10. Translucent rendering effect of candles under dynamic light source.

Figure 11. Rendering frame rate.

denoising monte carlo renderings. *ACM Trans. Graph.*, 36(4):97–1, 2017.

[3] A. Ben-Artzi, R. Overbeck, and R. Ramamoorthi. Real-time brdf editing in complex lighting. *ACM Transactions on Graphics (TOG)*, 25(3):945–954, 2006.

[4] I.-Y. Cho, Y. Huo, and S.-E. Yoon. Weakly-supervised contrastive learning in path manifold for monte carlo image reconstruction. *ACM Transactions on Graphics (TOG)*, 40(4):38–1, 2021.

[5] C. Dachsbacher, J. Křivánek, M. Hašan, A. Arbree, B. Walter, and J. Novák. Scalable realistic rendering with many-light methods. In *Computer Graphics Forum*, volume 33, pages 88–104. Wiley Online Library, 2014.

[6] C. Donner and H. W. Jensen. Light diffusion in multilayered translucent materials. *ACM Transactions on Graphics (ToG)*, 24(3):1032–1039, 2005.

[7] J. Dorsey, A. Edelman, H. W. Jensen, J. Legakis, and H. K. Pedersen. Modeling and rendering of weathered stone. In *ACM SIGGRAPH 2005 Courses*, pages 4–es. 2005.

[8] H. Fan, R. Wang, Y. Huo, and H. Bao. Real-time monte carlo denoising with weight sharing kernel prediction network. In *Computer Graphics Forum*, volume 40, pages 15–27. Wiley Online Library, 2021.

[9] P. Hanrahan and W. Krueger. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 165–174, 1993.

[10] X. Hao and A. Varshney. Real-time rendering of translucent meshes. *Acm Transactions on Graphics (Tog)*, 23(2):120–142, 2004.

[11] M. Hašan, F. Pellacini, and K. Bala. Direct-to-indirect transfer for cinematic relighting. *ACM transactions on graphics (TOG)*, 25(3):1089–1097, 2006.

[12] Y. Huo, S. Jin, T. Liu, W. Hua, R. Wang, and H. Bao. Spherical gaussian-based lightcuts for glossy interreflections. In *Computer Graphics Forum*, volume 39, pages 192–203. Wiley Online Library, 2020.

[13] Y. Huo, R. Wang, T. Hu, W. Hua, and H. Bao. Adaptive matrix column sampling and completion for ren-
dering participating media. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016. 7

[14] Y. Huo, R. Wang, S. Jin, X. Liu, and H. Bao. A matrix sampling-and-recovery approach for many-lights rendering. *ACM Transactions on Graphics (TOG)*, 34(6):1–12, 2015. 7

[15] Y. Huo, R. Wang, R. Zheng, H. Xu, H. Bao, and S.-E. Yoon. Adaptive incident radiance field sampling and reconstruction using deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 39(1):1–17, 2020. 7

[16] Y. Huo and S.-e. Yoon. A survey on deep learning-based monte carlo denoising. *Computational Visual Media*, 7(2):169–185, 2021. 7

[17] H. W. Jensen and J. Buhler. A rapid hierarchical rendering technique for translucent materials. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 576–581, 2002. 2

[18] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 311–320, 1998. 1

[19] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518, 2001. 1, 2

[20] S. Kim, Y. Huo, and S.-E. Yoon. Single image reflection removal with physically-based training images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5164–5173, 2020. 7

[21] S. Li, R. Wang, Y. Huo, W. Zheng, W. Hua, and H. Bao. Automatic band-limited approximation of shaders using mean-variance statistics in clamped domain. In *Computer Graphics Forum*, volume 39, pages 181–192. Wiley Online Library, 2020. 7

[22] S. Li, C. Zheng, R. Wang, Y. Huo, W. Zheng, H. Lin, and H. Bao. Multi-resolution terrain rendering using summed-area tables. *Computers & Graphics*, 95:130–140, 2021. 7

[23] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. In *ACM SIGGRAPH 2003 Papers*, pages 376–381. 2003. 2, 4, 5

[24] H. Park, Y. Huo, and S.-E. Yoon. Meshchain: Secure 3d model and intellectual property management powered by blockchain technology. In *Computer Graphics International Conference*, pages 519–534. Springer, 2021. 7

[25] P.-P. Sloan, J. Hall, J. Hart, and J. Snyder. Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics (TOG)*, 22(3):382–391, 2003. 2

[26] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, 2002. 2, 4

[27] X. Sun, K. Zhou, Y. Chen, S. Lin, J. Shi, and B. Guo. Interactive relighting with dynamic brdfs. In *ACM SIGGRAPH 2007 papers*, pages 27–es, 2007. 2

[28] Y.-T. Tsai and Z.-C. Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transactions on graphics (TOG)*, 25(3):967–976, 2006. 2

[29] R. Wang, Y. Huo, Y. Yuan, K. Zhou, W. Hua, and H. Bao. Gpu-based out-of-core many-lights rendering. *ACM Transactions on Graphics (TOG)*, 32(6):1–10, 2013. 7

[30] R. Wang, J. Tran, and D. Luebke. All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Transactions on Graphics (TOG)*, 24(3):1202–1207, 2005. 1, 2

[31] R. Wang, X. Yang, Y. Yuan, W. Chen, K. Bala, and H. Bao. Automatic shader simplification using surface signal approximation. *ACM Transactions on Graphics (TOG)*, 33(6):1–11, 2014. 7

[32] R. Wang, B. Yu, J. Marco, T. Hu, D. Gutierrez, and H. Bao. Real-time rendering on a power budget. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 7

[33] B. Xu, J. Zhang, R. Wang, K. Xu, Y.-L. Yang, C. Li, and R. Tang. Adversarial monte carlo denoising with conditioned auxiliary feature modulation. *ACM Trans. Graph.*, 38(6):224–1, 2019. 7

[34] K. Xu, Y. Gao, Y. Li, T. Ju, and S.-M. Hu. Real-time homogenous translucent material editing. In *Computer Graphics Forum*, volume 26, pages 545–552. Wiley Online Library, 2007. 2, 3

[35] Y. Zhang, R. Wang, Y. Huo, W. Hua, and H. Bao. Powernet: Learning-based real-time power-budget rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2021. 7

[36] Y. Zhong, Y. Huo, and R. Wang. Morphological anti-aliasing method for boundary slope prediction. *arXiv preprint arXiv:2203.03870*, 2022. 7

[37] K. Zhou, Y. Hu, S. Lin, B. Guo, and H.-Y. Shum. Precomputed shadow fields for dynamic scenes. In *ACM
