Abstract

This study introduces an approximation for rendering one bounce glossy interreflection in real time. The solution is based on the most representative point (MRP) and extends to a sampling disk near the MRP. Our algorithm represents geometry as rectangle proxies and specular reflections using a spherical Gaussian. The reflected radiance from the disk was efficiently approximated by selecting a representative attenuation axis in the sampling disk. We provide an efficient approximation of the glossy interreflection and can efficiently perform the approximation at runtime. Our method uses forward rendering (without using GBuffer), which is more suitable for platforms that favor forward rendering, such as mobile applications and virtual reality.

1. Introduction

Real-time glossy interreflections are important visual cues for synthetic images. These interreflections are important for interactive virtual environments, such as virtual reality (VR) applications or video games. The real-time rendering of glossy interreflections is a challenging task. The difficulty resides in the evaluation of a two-dimensional specular radiance integral for which no practical solution exists, except for expensive Monte Carlo-based sampling techniques. Most compelling solutions are found using the most representative point (MRP) approaches, thereby reducing the shading integration problem to a cheap point-lighting calculation. However, these methods fail to preserve the specular highlight shape of underlying bidirectional reflectance distribution functions (BRDFs). Moreover, finding an approximation for real-time performance remains a problem. Despite advances in graphics hardware and corresponding algorithms, it is currently impossible to render accurate glossy interreflections.

We classify real-time global illumination (GI) algorithms into precomputation and post-processing methods. The former uses precomputed data to accelerate real-time computation. Screenspace methods solve GI problems by sparsely sampling the space, which is simple to implement but requires additional filtering to remove noise, and most of them can only handle diffuse interreflection. In some platforms favoring forward rendering, such as VR applications and mobile platforms, the screenspace method is not applicable. In addition, in some platforms that lack computation performance, such as computer games running on mobile devices, there are currently no methods to approximate such effects.

This study presents a new forward rendering method for rendering glossy interreflections at real-time frame rates. Our method is based on a rectangle-based structure that is used to approximate the origin geometry. We thereafter used a spherical Gaussian (SG) to approximate the reflected lighting in a rectangle for a given shading point. Our algorithm can promptly approximate the integral of reflection lighting on a rectangle in real time.

In conclusion, the contributions of this study are as follows:

- We provide a functional approximation of the glossy interreflection, and we can efficiently perform the approximation at run time.
- We develop a novel spherical Gaussian representation of glossy reflection.
- We investigate the real-time GI algorithm using forward rendering.

2. Background

2.1. Related work

We first review some related works and introduce the necessary background of the SGs representation.

Precompute methods. Precomputed radiance transfer (PRT) [22] precomputes linear operators that map the light transport equation into indirect radiance or irradiance sampled over the scene surfaces. The precomputed functions are stored using basis functions,
such as spherical harmonics (SHs) [18, 28], wavelet [15], and SG [23]. This method assumes that the scene geometry is fixed and only the light and camera change. The main idea is to project the light transport function into the frequency domain using a spherical basis and thereafter perform real-time multiplication in the frequency domain rather than time-consuming convolution in the space domain. However, this method cannot handle dynamic objects and can only represent low-frequency features. Photon mapping [6, 5] is widely used in video games: First, photons are emitted from a light source and recorded on the geometric surface of the scene and subsequently organized in a spatial data structure. Thereafter, a second pass starting from the surfel of light-field texture while gathering nearby photons to estimate indirect illumination. Finally, these light-field textures were used for real-time rendering.

Postprocessing methods. Some methods do not require precomputation to calculate GI. A reflective shadow map [3, 4] considers each pixel in the colored shadow map an indirect light source (virtual point light [12, 25, 24]). Light propagation volumes (LPVs) [10] store lighting information from light in a 3D grid. Every grid represents the distribution of the indirect light in the scene. Light is propagated from the 3D grid to nearby grids. The drawback of LPV is that it can only use one distant light source. Screenspace directional occlusion (SSDO) [20] accounts for the direction of the incoming light and includes one bounce of indirect illumination. Because of the sparse sampling of SSDO, sophisticated nonlinear filters are required to remove noise (e.g., [13]). SSDO has been widely adopted in the video game industry because it relies on GBuffer; therefore, it cannot be used well under certain conditions such as VR and some mobile platforms and can only handle diffuse indirect illuminations. Our method is based on forward rendering, without the use of GBuffer. In addition, almost all of these methods reuse screen-space illumination generated by the camera to compute the next bounces of light transport. This approximation works well if the scene is diffuse, or the next bounces of light transport are insensitive to directions. However, for glossy reflections, such an approximation is no longer correct. Our method considers the BRDF direction of the glossy reflections.

Geometric approximations. Some methods use geometric approximations to perform faster integration. Point-based models [1] store surface elements of shape in a point cloud file and are organized into an octree hierarchy; thereafter, the illumination from the surface point in each octree node is approximated using SHs. Voxel cone tracing (VXGI) [2, 19] approximates one bounce of light in a dynamic scene, a voxel representation is created once for static objects and every frame for dynamic objects. Each voxel stores the amount of light that the geometry emits in all directions, and the irradiance is calculated using cone tracing. The results of VXGI are satisfactory, but cone tracing is time-consuming. In this study, we use a rectangle to accelerate integration.

Most representative point. In addition to geometric approximations, Picott [17] first introduced MRP approaches to alleviate the costly sampling computation by identifying a representative point in the light area that contributes the most to the illumination. Karis [11] used a modification of the specular distribution to ensure energy conservation, making the result better match the intensity highlight of specular microfacet models. We simply extend these methods by selecting a specular peak point as a representative point for specular interreflection, which builds on the property that the reflected light path half vector coincides with the surface normal at the specular peak point.

SG lighting. Compared to SHs, SGs [27] are more compact for all frequency effects. Since several fundamental rendering operations such as integral, product, and product integral have closed-form solutions in SG, SG can be used to represent the different parts of the rendering equation to approximate the illumination integral. The Gaussian approximation for the reflectance of several microfacet BRDF models can be derived in a closed form. Since a normalized SG is equivalent to the von Mises–Fisher distribution, several SGs can be merged into a single SG in an analytic way. Wang et al. [27] developed an SG approximation for spatially varying BRDFs (SVBRDFs), allowing for real-time all-frequency rendering. Iwasaki et al. [8] introduced an integral SG and reduced the need for precomputed data. Anisotropic SGs (ASGs) [31] have been used to represent anisotropic lighting and BRDFs. Pettineo [16] used SGs to encode light-field textures in the videogame The Order 1886. Their experiment demonstrated that with 12 SGs with fixed direction and sharpness (i.e., 36 floats), they can better represent the original light field than a 3-band SH representation (24 floats). Xu et al. [29] used the SG to approximate local interreflections. They integrated the projection of a triangle over a sphere as an SG, and the performance was linked to the geometry tessellation. However, their method cannot achieve real-time performance because they usually need to integrate several triangles, which severely reduces the performance. Our work approximates the geometry as rectangles first and subsequently uses this approximation to reduce the calculation cost; we only need to calculate a few rectangles
rather than isotropic: the lobe amplitude.

The term is defined as follows:
\[ G_i(v; p, \nu, a) = a \cdot e^\nu(v \cdot p - 1), \]
(1)
where \( v \in S^2 \) denotes the function input, \( p \in S^2 \) is the lobe axis, \( \nu \in \mathbb{R}_+ \) is the lobe sharpness, and \( a \in \mathbb{R}_+ \) is the lobe amplitude.

An ASG behaves similar to SGs but is anisotropic rather than isotropic:
\[ G_a(v; [z, x, y], [\lambda, \mu], c) = c \cdot S(v; z) \cdot e^{-\lambda(v \cdot x)^2 - \mu(v \cdot y)^2}. \]
(2)
where \( z, x, y \) denote the lobe, tangent, and bi-tangent axes, respectively, and \( \lambda \) and \( \mu \) are the bandwidths for the \( x- \) and \( y- \)axes, respectively, satisfying \( \lambda, \mu > 0 \).
The term \( c \) denotes the lobe amplitude; the smooth term is defined as follows: \( S(v; z) = \max(v \cdot z, 0) \).

3. Rendering

3.1. Algorithm Overview

Our method aims to approximate the glossy interreflections with all-frequency BRDFs and describe the efficient approximation of one-bounce interreflections from a single rectangle (reflector) to a single shading point (receiver). We assume that the lighting is distant and that a single rectangle has a uniform BRDF. We observe that the reflected radiance in a plane (to a shading point) is attenuated around a point (Figure 1a). We thereafter select a representative point (referred to as the specular peak), and a sampling disk (centered at the specular peak) is subsequently created to integrate the reflected radiance. We integrated the reflected radiance as the disk was projected onto a sphere (centered on a shading point) (Figure 1b). Further, the reflected radiance is approximated with the ASG by preserving the area and function energy (Figure 1c). Finally, the normal distribution function (NDF) at the receiver is approximated as SG and subsequently convolved with ASG light to compute the final lighting.

3.2. Rendering Model

We start by deriving the one-bounce interreflection model from a single rectangular reflector under distant lighting, as shown in figure 2. Given a rectangle \( R \) with normal \( n_r \) and a shading point \( x \) with normal \( n_x \), we aim to compute the outgoing radiance from \( x \) to the view direction \( o \) after the reflection of light from the rectangle \( R \) toward \( x \). To simplify the derivation, we assume that there is no occlusion between the light, reflected rectangle, and shading point. The one-bounce radiance from \( x \) toward \( o \) can be expressed as a standard rendering equation [9]:
\[ L_x(o) = \int_{\Omega_R} L(r) f_r(-r, o), (-r \cdot n_x) dr. \]
(3)
where \( r \) denotes the direction from a point on the reflected rectangle \( R \) to \( x \), and the integral is over the rectangle \( \Omega_R \) subtended by \( R \); \( f_r \) and \( n_x \) are the BRDF and normal directions at the receiver point \( x \), respectively, and \( L(r) \) is the reflected radiance from any point (on rectangle \( R \)) to \( x \), defined as follows:
\[ L(r) = \int_{\Omega} L(i) f_r(i, r) (i \cdot n_r) di. \]
(4)
where \( n_r \) denotes the normal of the rectangle \( R \), and \( L(i) \) is the incident light.

Here, we use a simplified Disney BRDF model [14, 11] for \( f_r \):
\[ f_r(\omega_i, \omega_o) = M(\omega_i, \omega_o) D(h), \]
(5)
where \( \omega_i \) denotes the incident light direction, \( \omega_o \) is the viewing direction, \( h = (\omega_i + \omega_o)/\|\omega_i + \omega_o\| \) is the half vector, \( M \) is the Fresnel and shadowing effects, and \( D \) is the NDF.

We represent the NDF at shading point \( x \) as a uniform SG [27], which is also a von Mises–Fisher distribution:
\[ D(h) = G_a(h; n_x, \frac{2}{m^2}, \frac{1}{\pi m^2}) \]
(6)
where \( n_x \) is the surface normal, and \( m \) is the material roughness at the shading point.

We represent the \( L(r) \) as an ASG [31]:
\[ L(r) = G_a(-r; [z, m, n], [\lambda, \mu], t) \]
(7)
We denote as \( G_a(\cdot) \) for notation simplicity. The computation of the ASG parameters is described in Sec. 3.5.

To convolve with light ASG, we need to express \( D \) in terms of lighting vector rather than half vector. The Jacobian of this change is \( 4\|h \cdot o\| \) and \( D \) is,
\[ D(h) = G_i(-r; r_x, \frac{2}{4\|h \cdot o\| m^2}, \frac{1}{\pi m^2}) \]
(8)
where \( r_x \) denotes the new SG lobe axis, which can be calculated by \( r_x = 2\|o \cdot h\| h - o \). We denote as \( G_i(\cdot) \) for notation simplicity.

Now, we can rewrite Eq. 3 as follows:
\[ L_x(o) \approx \int_{S^2} G_a(\cdot) G_i(\cdot) M(-r, o), (-r \cdot n_x) dr. \]
(9)
A specular peak point $y$ is selected on the reflector plane; $D$ is then selected to gather the one-bounce reflected radiance from rectangle $R$ to shading point $x$, and we also observed that the reflected radiance can be represented with a representative attenuation axis $\mathbf{u}$. The integrated reflected radiance from the disk is treated as the disk (green) and projected onto a sphere; the projected area (red) is an ellipse-like shape, and the $n, m$ axes of the ellipse correspond to the $\mathbf{u}$ axes of the disk. The projected area (red dotted lines) was approximated using an ASG. The NDF (red lobe) is approximated with an SG (green lobe) then warped from the half-vector domain to the lighting domain (blue lobe), and subsequently convolved with ASG light (orange lobe).

Since the last two terms are low-frequency terms, they can be approximated by a constant, and we can find a specific location $y$ (explained in Sec. 3.3) at rectangle $R$ and thereafter pull them out of the integral:

$$L_x(o) \approx M(-r_y, o) \cdot \mathbf{r}_y \cdot \mathbf{n}_x \cdot \int_{S^2} G_a(\cdot)G_i(\cdot) dr,$$

where $r_y$ denotes a direction point from the representative point $y$ on the reflected rectangle $R$ to the shading point $x$.

The product integrals of $G_a(\cdot)$ and $G_i(\cdot)$ were derived by Xu [30]:

$$H(p) = \int_{S^2} G_a(v; z, x, y, [\lambda, \mu], 1) \cdot G_i(v; p, v, 1) dv$$

$$\approx G_a(p, [z, x, y], [\nu \lambda, \nu \mu], \nu \lambda + \nu \mu, \nu \mu, \pi \sqrt{\lambda + \nu})(\mu + \nu)$$

Now, the problem of calculating $L_x(o)$ is reduced to two questions: finding the representative point $y$ on rectangle $R$ to approximate the low-frequency terms and determining the parameters of $G_a(\cdot)$ to closely approximate $L(r)$.

### 3.3. Finding Specular Peak

As shown in figure 2), for a given rectangle $R$ centered at $c_r$, with a normal $n_r$, the shading point $x$ is at position $p_x$, and the distance in the light direction is $i$. Since we find a point on $R$ where the normal vector is equal to the half vector, we first compute the reflected direction $r$ at point $y$ by $r = 2(i \cdot n_r)n_r - i$. Thereafter, the position of the peak point $y$, $p_y$, can be calculated as follows:

$$p_y = p_x - r \cdot (p_x - c_r) \cdot n_r \cdot i \cdot n_r > 0$$

We thereafter used this point to approximate the low-frequency terms in Eq. 10.

### 3.4. Integrating Reflected Radiance

We approximate the ASG light $G_a(\cdot)$ by preserving the area and function energy of the reflected radiance
from the rectangle $R$. In this section, we introduce the approximation of the total reflected radiance from $R$, and the result is used to create $G_d(\cdot)$ in Sec.3.5.

The total reflected radiance from rectangle $R$ to shading point $x$ is obtained as follows:

$$C = \int_{\Omega_R} L(r)dr.$$  \hspace{1cm} (13)

To approximate the reflected radiance from rectangle $R$ to shading point $x$, we first select a sampling disk $D(y, r_d, n_d)$, with $y, r_d,$ and $n_d$ as the center point, radius, and normal of the disk, respectively, where $y$ is the position of the specular peak point, $n_d$ is equal to the normal of rectangle $n_r$, and $r_d$ is the radius of the disk; $r_d$ is a user-controlled variable. We approximate $C$ as follows:

$$C \approx A \cdot \int_{\Omega_D} L(r)dr.$$ \hspace{1cm} (14)

The integral changes to $\Omega_D$ subtended by $D$; $A$ is the intersection area divided by the disk area, and it is explained in Sec.3.6.

We first evaluate the reflected radiance from specular peak $y$ to shading point $x$ because $y$ is the sampled point on rectangle $R$, and its value can be estimated using the Monte Carlo method as follows:

$$L_y(r) \approx \frac{1}{N} \sum_{k=1}^{N} f_r(i_k, r_k) (i_k \cdot n_y) \frac{p(i_k, r_k)}{p(i, r)}.$$ \hspace{1cm} (15)

where $N$ denotes the light number in the scene, $i_k, r_k,$ and $n_y$ correspond to $i, r,$ and $n_y$ Eq 4, related to the $k$-th light. The term $p(i_k, r_k)$ is the probability density function(PDF) with respect to the solid angle for the $k$-th light. For notation simplicity, we subsequently omit the summation $\sum_{k=1}^{N}$ over index $k$.

The term $p(i, r)$ can be calculated using the NDF. In this study, we used the GGX microfacet BRDF [26] as NDF, which is currently considered the most realistic parametric BRDF [21]. The definition of GGX is obtained as follows:

$$D(\theta, m) = \frac{m^2}{\pi(1 - (1 - m^2) \cdot \cos(\theta)^2)^2}.$$ \hspace{1cm} (16)

where $m$ denotes the material roughness, and $\theta$ is the angle between the normal vector and the half vector. The parameter $\theta$ is 0 at the specular peak $y$, that is, the normal vector is equal to the half vector.

Next, we substitute $p(i, r) = D(\theta, m_r) \cos \theta$ with $\theta = 0$ in Eq 15, which yields,

$$L_y(r) \approx \pi m_r^2 f_r(i, r) (i \cdot n_y).$$ \hspace{1cm} (17)

where $m_r$ denotes the material roughness of rectangle $R$.

After estimating the reflected radiance from $y$, we integrate the total reflected radiance from the sampling disk $D$. First, we create a sphere centered at $x$ with radius $\|p_y - p_x\|$ then integrate over a spherical cap to compute the reflected radiance from disk $D$ because the disk is projected onto this sphere, as shown in Figure 1b. For notation simplicity, but without loss of generality, we assume that the distance between $x$ and $y$ is 1 in the remainder of this study, that is, we are working on a unit sphere. We observe that the reflected radiance is proportional to the angle between the reflected rectangle normal $n_r$ and shading normal $n_x$. Subsequently, we select a representative attenuation axis $\bar{u}$ and assume that the radiance attenuation from specular peak $y$ toward all directions is the same as that in direction $\bar{u}$, and it can be calculated as follows (Figure 1a):

$$\begin{align*}
\vec{v} &= y \hat{x} \times \vec{n}_r \\
\vec{u} &= \vec{n}_r \times \vec{v}
\end{align*}$$ \hspace{1cm} (18)

To demonstrate the derivation of the integral but without loss of generality, we use $\hat{x}$ as the zenith axis, where the axis $\vec{v}$ points toward the screen, as shown in Figure 3.

Now, we approximate the integral term in Eq. 14 as follows:

$$E = \int_{\Omega_D} L(r)dr$$
$$\approx \int_0^{2\pi} \int_0^\alpha L_c(\theta, \phi) \sin \theta \, d\theta \, d\phi.$$ \hspace{1cm} (19)

![Figure 3: Solving $\psi(\theta)$.](image-url)
where \( L_c(\theta, \phi) \) denotes the reflected radiance in spherical coordinates \((\theta, \phi)\), and \( \alpha \) is the aperture angle of integrating the spherical cap (as shown in Figure 3); the calculation of \( \alpha \) will be explained later. Since we already assume that the attenuation of the reflected radiance is represented by axis \( \mathbf{u} \), we can directly integrate the \( \phi \) term as follows:

\[
E \approx 2\pi \int_0^\alpha L_c(\theta) \sin \theta \, d\theta \tag{20}
\]

\( L_c(\theta) \) can be estimated in the same manner as Eq. 17:

\[
L_c(\theta) \approx \pi m_r^2 f_r(i, r) (i \cdot \mathbf{n}_f) = \pi m_r^2 M(i, r) \cdot D(\psi(\theta), m_r) (i \cdot \mathbf{n}_f). \tag{21}
\]

where \( \psi(\theta) \) denotes an angle function that represents the angle between the normal and half vectors at a differential point (e.g., \( y' \) on the reflector disk in Figure 3), and its value depends on the \( \theta \) in the integrator. Notably, \( L_c(0) = L_y(r) \).

Since \( M \) is a low-frequency term, we use its value at point \( y \) and pull the low-frequency terms out of the integral as follows:

\[
E = 2\pi \int_0^\alpha \pi m_r^2 M(i, r) D(\psi(\theta), m_r) (i \cdot \mathbf{n}_f) \sin \theta \, d\theta
\]

\[
\approx 2\pi \cdot \pi m_r^2 M(i, r) (i \cdot \mathbf{n}_f) \int_0^\alpha D(\psi(\theta), m_r) \sin \theta \, d\theta
\]

\[
= 2\pi L_y(r) \int_0^\alpha \frac{D(\psi(\theta), m_r)}{D(0, m_r)} \sin \theta \, d\theta,
\]

\[
= 2\pi L_y(r) \int_0^\alpha f(\theta) \sin \theta \, d\theta. \tag{22}
\]

We denote \( f(\theta) = \frac{D(\psi(\theta), m_r)}{D(0, m_r)} \) and substitute Eq. 16 into \( f(\theta) \):

\[
f(\theta) = \frac{m_r^4}{(1 - (1 - m_r^2) \cos^2(\psi(\theta)))^2} \tag{23}
\]

The solution of \( \psi(\theta) \) is shown in Figure 3. For a differential angle \( \theta \), we can find a point \( y' \) on the reflector disk \( D \) along the representative axis \( \mathbf{u} \). We denote \( \sigma \) as the angle between the light vector \( \mathbf{i} \) and the reflector normal vector \( \mathbf{n}_r \), \( \mathbf{h} \) is the half vector at point \( y' \), and \( d \) is the distance between \( y' \) and \( y \). Thereafter, \( d \) can be approximated as \( d \approx \frac{\tan \theta}{\cos \sigma} \). Hence,

\[
\psi(\theta) \approx -\frac{\sigma}{2} + \frac{1}{2} \arctan(\sec^2 \sigma \tan \theta + \tan \sigma) \tag{24}
\]

Unfortunately, the integral \( E \) does not have an analytic solution. To solve Eq. 22, we first fit \( f(\theta) \) to the following equation:

\[
f(\theta) \approx e^{-k\sin^2 \theta} \cos \theta, \quad k = \frac{0.288 \cos \sigma}{m_r^2} - 0.673 \tag{25}
\]

The fit error is shown in Figure 4, and the fitted shape (Eq. 25) is close to the original curve shape (Eq. 23) when \( \sigma \) and \( m_r \) are both small (Figure 4a), but will produce large errors when one of them is large (Figure 4b, Figure 4c, Figure 4d). However, in practice, when \( \sigma \) or \( m_r \) is large, the value of \( f(\theta) \) is small and contributes less to the entire reflected radiance. Hence, the approximation error in this case does not significantly affect the result.

Substituting Eq. 25 into Eq. 22, we obtain the following:

\[
E \approx 2\pi L_y(r) \int_0^\alpha e^{-k\sin^2 \theta} \cos \theta \sin \theta \, d\theta
\]

\[
= 2\pi L_y(r) \cdot \left( -\frac{1}{2k} e^{-k\sin^2 \theta} \right)^\alpha_0. \tag{26}
\]

\[
= \frac{\pi}{k} L_y(r) \cdot (1 - e^{-k\sin^2 \alpha}).
\]

Thereafter, we approximate \( \alpha \), the upper limit of the integral, as \( \alpha \approx \arctan(r_d \cdot \cos \sigma) \), where \( r_d \) denotes the radius of the sampling disk \( D \); then, \( \sin^2 \alpha \approx 1 - \frac{1}{1 + r_d^2 \cos^2 \sigma} \). The radius of the sampling disk is a user control variate, and the choice of \( r_d \) will be discussed in Sec. 4.
In conclusion, the total reflected radiance from the sampling disk \( D \) to the shading point \( x \) is approximated by the following equation:

\[
C \approx A \frac{\pi}{k} (1 - e^{-kt}) L_\theta(r)
\]

(27)

with \( k = \frac{0.288 \cos \sigma}{m_r^2} \) and \( t = 1 - \frac{1}{1 + r_d^2 \cos^2 \sigma} \).

Since \( \cos \sigma = n_r \cdot i \), this approximation has a performance advantage in avoiding costly inverse trigonometry operations on the GPU.

3.5. Approximating ASG light

We compute the parameters of the ASG light introduced in Eq.7. The shape of the projected area is similar to that of an ellipse, as shown in Figure 1b. We subsequently approximate the reflected radiance as an ASG light(Figure 1c), which is denoted as \( G_a(v; [z, m, n], [\lambda, \mu], t) \). The lobe axis \( z \) can be directly calculated using \( z = \frac{p_y - p_x}{\|p_y - p_x\|} \).

The tangent axis \( m \) and bitangent axis \( n \) correspond to the \( \vec{v} \) and \( \vec{u} \) axes of the sampling disk \( D \), respectively. As shown in Figure 1b, \( m \) and \( n \) can be solved using the following equation:

\[
\begin{align*}
m &= \vec{v} \\
n &= y \hat{x} \times \vec{v}
\end{align*}
\]

(28)

We denote \( l_m, l_n \) as the length of \( m, n \); then,

\[
\begin{align*}
l_m &= r_d \\
l_n &= r_d (n_d \cdot y \hat{x})
\end{align*}
\]

(29)

Before solving the bandwidth, we first introduce the polar version of \( G_a \), which is equal to Eq. 2:

\[
G_p(\theta, \phi, \eta, [\lambda, \mu], c) = c \cdot \cos \theta \cdot e^{-\lambda \cos^2 \phi - \mu \cos^2 \eta}.
\]

(30)

Compared to Eq. 2, \( \theta \) denotes the polar angle between vector \( v \) and \( z \), \( \phi \) is the polar angle between vector \( v \) and \( x \), \( \eta \) is the polar angle between vectors \( v \) and \( y \), and the definitions of \( \lambda, \mu, \) and \( c \) remain the same.

Since ASGs are compactly \( \epsilon \)-supported, we can ensure that in the closed area, \( G_a \) attains values larger than the threshold \( \epsilon \), and subsequently obtains the bandwidth \( \lambda, \mu \) from shape size \( l_m, l_n \):

\[
\begin{align*}
G_p(\arctan(l_m), \pi \frac{2}{2} \arctan(l_m), \pi \frac{2}{2}, \lambda, \mu, 1) &= \epsilon \\
G_p(\arctan(l_n), \frac{2}{2}, \pi \frac{2}{2} \arctan(l_n), \lambda, \mu, 1) &= \epsilon
\end{align*}
\]

(31)

Particularly, we use \( \epsilon = 0.05 \) to calculate the bandwidth:

\[
\begin{align*}
\lambda &= (1 + \frac{1}{1 + m_n^2})(2.996 - 0.5 \ln(1 + m_n^2)) \\
\mu &= (1 + \frac{1}{1 + m_n^2})(2.996 - 0.5 \ln(1 + m_n^2))
\end{align*}
\]

(32)

We obtained the amplitude by preserving the function energy in Eq. 27, as follows:

\[
\int_{\Omega} G_a(v; [z, m, n], [\lambda, \mu], t)dv = C
\]

(33)

The approximation of the ASG integral is as follows:

\[
\int_{\Omega} G_a(v; [z, m, n], [\lambda, \mu], t)dv \approx \frac{\pi}{\sqrt{\lambda \mu}} t
\]

(34)

then:

\[
t \approx \frac{\sqrt{\lambda \mu}}{\pi} C
\]

(35)

3.6. Rectangle Proxy

Our algorithm relies on a rectangle proxy (RP) constructed upon each triangle geometry, which closely approximates the geometry surface. For a given scene, the RPs are generated by offline geometry tools for each geometry object. Thereafter, all the proxies are organized into an acceleration structure such as kdtree. We subsequently create one or more light-field textures for each geometry; each texel of light-field texture stores the indices of possible reflected RPs. The indices of static RPs are one-off cache in the related light-field texture, whereas the indices of dynamic RPs are updated to the related light-field texture in each frame. To handle scenes with greater geometric complexity, we can partition the scene, and represent the subspaces with separate and larger RPs that can reduce the RP number and evaluate more rapidly. Because of its compactness and ease of evaluation, the RP model enables real-time rendering with full global illumination effects, particularly for high-frequency glossy interreflections. Since we represent the scene with rectangles, we can calculate the one-bounce radiance from a rectangle to a shading point.

Using a rectangle, we can quickly approximate the intersection area of the rectangle and disk, as shown in figure 6. We approximate \( A \) in Eq.14 with the following function:

\[
A = \frac{S_I}{\pi r_d^2} \approx \frac{S_J}{\pi r_d^2}
\]

(36)

where \( r_d \) denotes the radius of the disk, \( \pi r_d^2 \) is the area of the disk, \( I \) is the real intersection area of the rectangle and disk, \( S_I \) is the area of region \( I \), \( J \) is the axis-aligned bounding rectangle of \( I \), and \( S_J \) is the area of \( J \).
4. Results and Discussions

This section presents the results of our algorithm. We present our rendering results on the Cornell Box and the effect when changed to apply distant light to the scene. The error caused by the convolution of the ASG and SG is also discussed.

Figure 5 shows our rendering result with different lighting directions.

In addition, the approximation of Eq. 11 has a large error when the bandwidth of the ASG is smaller than 50, as shown in Figure 7.
We can approximate the ASG light with an SG to improve performance, which removes the anisotropic property but reduces the calculation cost. The ASG can be approximated as follows:

\[ G_a(v;[z,x,y],[\lambda,\mu],1) \approx G_i(v;z,2\lambda,1) \]  

(37)

We assume that \( \lambda \geq \mu \), Eq. 11 can be approximated by the product integral of two SGs, and the calculation of the product integral of two SGs was derived in the work by Iwasaki [7].

References

[1] Per H. Christensen. Point-based approximate color bleeding. Pixar Technical Memo 08–01, 2008. 2
[2] Cyril Crassin, Fabrice Neyret, Miguel Sainz, Simon Green, and Elmar Eisemann. Interactive indirect illumination using voxel cone tracing. In Symposium on Interactive 3D Graphics and Games on - I3D ’11. ACM Press, 2011. 2
[3] Carsten Dachsbacher and Marc Stamminger. Reflective shadow maps. In Proceedings of the 2005 symposium on Interactive 3D graphics and games - SIGGRAPH ’05. ACM Press, 2005. 2
[4] Carsten Dachsbacher and Marc Stamminger. Splatting indirect illumination. In Proceedings of the 2006 symposium on Interactive 3D graphics and games - SI3D ’06. ACM Press, 2006. 2
[5] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. In ACM SIGGRAPH Asia 2009 papers on - SIGGRAPH Asia ’09. ACM Press, 2009. 2
[6] Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. In ACM SIGGRAPH Asia 2008 papers on - SIGGRAPH Asia ’08. ACM Press, 2008. 2

Figure 7: Convolution error.

The error caused by the product integral of the ASG and SG when the bandwidth is small. We show the error in Eq. 11 with \( G_a(v;[0,0,1],(1,0,0),(0,1,0)], [\lambda,\lambda], 1) \), and \( G_i(v; (0.5,0,1), (2\lambda,1) \) when \( \lambda \leq 50 \).
[21] HILL S., MCAULEY S., BURLEY B., CHAN D., FASCIONE L., IWANICKI M., HOFFMAN N., JAKOB W., NEUBELT D., PESCE A., and PETTINO M. Physically based shading in theory and practice. 2015.

[22] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics*, 21(3):527–536, July 2002.

[23] Yu-Ting Tsai and Zen-Chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transactions on Graphics*, 25(3):967–976, July 2006.

[24] Bruce Walter, Adam Arbree, Kavita Bala, and Donald P. Greenberg. Multidimensional lightcuts. In *ACM SIGGRAPH 2006 Papers on - SIGGRAPH ’06*. ACM Press, 2006.

[25] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. Lightcuts. In *ACM SIGGRAPH 2005 Papers on - SIGGRAPH ’05*. ACM Press, 2005.

[26] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. pages 195–206, 2007.

[27] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Transactions on Graphics*, 28(5):1–10, Dec. 2009.

[28] Rui Wang, Rui Wang, Kun Zhou, Minghao Pan, and Hujun Bao. An efficient GPU-based approach for interactive global illumination. In *ACM SIGGRAPH 2009 papers on - SIGGRAPH ’09*. ACM Press, 2009.

[29] Kun Xu, Yan-Pei Cao, Li-Qian Ma, Zhao Dong, Rui Wang, and Shi-Min Hu. A practical algorithm for rendering interreflections with all-frequency BRDFs. *ACM Transactions on Graphics*, 33(1):1–16, Jan. 2014.

[30] Kun Xu, Yue Gao, Yong Li, Tao Ju, and Shi-Min Hu. Real-time homogenous translucent material editing. *Computer Graphics Forum*, 26(3):545–552, Sept. 2007.

[31] Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Transactions on Graphics*, 32(6):1–11, Nov. 2013.