DOI: 10.1111/cgf.13034

Pacific Graphics 2016
E. Grinspun, B. Bickel, and Y. Dobashi
(Guest Editors)

Volume 35 (2016), Number 7

Multiple Scattering Approximation in Heterogeneous Media by Narrow Beam Distributions

Mikio Shinya[†], Yoshinori Dobashi[‡], Michio Shiraishi[§], Motonobu Kawashima[¶] and Tomoyuki Nishita[||]

Abstract

Fast realistic rendering of objects in scattering media is still a challenging topic in computer graphics. In presence of participating media, a light beam is repeatedly scattered by media particles, changing direction and getting spread out. Explicitly evaluating this beam distribution would enable efficient simulation of multiple scattering events without involving costly stochastic methods. Narrow beam theory provides explicit equations that approximate light propagation in a narrow incident beam. Based on this theory, we propose a closed-form distribution function for scattered beams. We successfully apply it to the image synthesis of scenes in which scattering occurs, and show that our proposed estimation method is more accurate than those based on the Wentzel-Kramers-Brillouin (WKB) theory.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—

1. Introduction

Realistic simulation of light scattering can produce visually appealing images. Although rendering objects in scattering media can be performed by computationally expensive Monte-Carlo style methods, such as path tracing [KVH84, YIC*10] and photon mapping [JC98], fast realistic rendering of such objects is still a challenging topic in computer graphics.

Explicitly evaluating scattered beam distributions would efficiently simulate multiple scattering events, while avoiding recourse to stochastic procedures. This approach has been very successful in subsurface scattering simulation. Methods calculate BSSRDFs/BSSTDFs (Bi-Directional Scattering Surface Reflectance/Transmittance Distribution Functions) and apply linear filtering using distribution functions as filter kernels for the efficient simulation of sub-surface scattering. Dipole [JMLH01], multipole [DJ05], and improved multipole [dI11, FHK14] methods have proven excellent in rendering layered scattering material such as human skin. These methods are basically low-pass filters that blur object textures. However, they experience difficulties when dealing with heterogeneous media.

Scattering simulations in participating media are more challenging because 3D distributions are required and heterogeneous structures are more visually important. It is known that Wentzel-Kramers-Brillouin (WKB) theory [LRT82] can estimate beam distributions. Premoze et al. [PAS03] applied the WKB theory to global illumination in participating media. Their proposed path-integral method integrates scattered illumination using beam distributions as a filter kernel. Elek et al. [ERS13] applied the WKB theory to simulate object blur due to scattering, and presented an efficient filtering scheme for fast rendering.

Given a deviated light path in media, which follows a curve in 3D space, the WKB theory estimates its probability. To obtain filter kernels, the probabilities should be integrated over all possible light path curves starting at, and reaching specified points. This involves very complex computations that require the introduction of major simplifications on path curve shapes for practical computation. This can cause large errors that limit their applications.

In this paper, we present explicit formulae for beam distributions based on narrow beam theory [Ish78]. A naive application of the theory to the filtering process involves 9D integrals for each pixel calculation. We reduce the computation by evaluating moments and fitting a heuristic function to the moments. From analysis of a single scattering event, we found a distribution function based on the error function (an integral of the Gaussian function) to be suitable for this application. Reasonable agreement was experimentally observed between the results from stochastic photon counting and our proposed estimation. We applied the distribution function to object blur due to scattering, as in Elek et al.'s approach [ERS13], and confirmed significant improvements.

The contributions of this paper are summarized as:

- we introduce narrow beam theory, a powerful tool for estimating scattering events, to the field of computer graphics;

[†] Toho University/UEI Research
[‡] Hokkaido University/UEI Research
[§] Toho University
[¶] MomomoWorks Inc.
[||] UEI Research/Hiroshima Shudo University

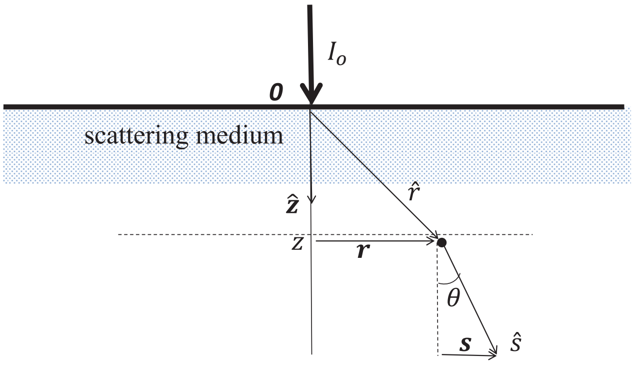


Figure 1: *Coordinate system and experimental arrangement.*

- we propose a heuristic distribution function based on narrow beam theory;
- we improve a screen-space scattering simulation by using our distribution function.

The proposed distribution function is also suitable for other image synthesis applications, such as shafts of light, Premoze et al.'s global illumination [PAS03], and BSSTDF calculations in heterogeneous media.

2. Narrow Beam Theory

This section gives a brief introduction to narrow beam theory [Ish78]. The major symbols used in the equations are listed in Table 1, and the associated coordinate system is shown in Figure 1.

As described in Section 1 of the supplemental document [PAS03, APRN04, PARN04], the WKB theory estimates the probability for a given light path, but the path-space formulation is not a simple and convenient tool to obtain distribution functions in 3D space. The narrow beam theory, on the other hand, analytically approximates solutions in real 3D space and leads to distribution functions in more explicit forms.

Table 1: *Symbols used in narrow beam theory.*

$\hat{r}, \boldsymbol{r}, z$	position ($\hat{r} = \boldsymbol{r} + z\hat{z}$)
$\hat{s}, \boldsymbol{s}$	direction ($\hat{s} = \boldsymbol{s} + \hat{z}$)
σ_t	extinction coefficient
σ_s	scattering coefficient
σ_a	absorption coefficient ($\sigma_a = \sigma_t - \sigma_s$)
ρ_n	particle density
ζ	integral of ρ_n along path
τ	optical depth ($\tau = \sigma_t\zeta$)
$\boldsymbol{\kappa}$	Fourier variable corresponding to $\boldsymbol{s}$
$\boldsymbol{q}$	Fourier variable corresponding to $\boldsymbol{r}$
$p(\boldsymbol{s})$	phase function
$P(\boldsymbol{\kappa})$	Fourier transform of p
$I_0(\boldsymbol{r}, \boldsymbol{s})$	incident light distribution
$F_0(\boldsymbol{\kappa}, \boldsymbol{q})$	Fourier transform of I_0

The light transport equation for intensity $I(\hat{r}, \hat{s})$ can be expressed by

$$\begin{aligned}(\nabla \cdot \hat{s})I(\hat{r}, \hat{s}) &= -(\rho_n\sigma_t)I(\hat{r}, \hat{s}) + \\ &\quad (\rho_n\sigma_s)\int_{4\pi} p(\hat{s}, \hat{s}')I(\hat{r}, \hat{s}')d\hat{s}', \quad (1)\\ \int_{4\pi} p(\hat{s}, \hat{s}')d\hat{s}' &= 1. \quad (2)\end{aligned}$$

where $\hat{s}$ and $\hat{r}$ represent the direction and position in 3D, ρ_n is the density of the media, σ_t and σ_s are the extinction and scattering coefficients, and p is the phase function normalized to 1. The integral domain 4π indicates the unit sphere, and the domain is infinite unless explicitly specified.

We set the intensity I to be the sum of the direct component I_{ri} and indirect component I_d

$$I = I_{ri} + I_d.$$

Let us assume that

- the intensity I is concentrated near the $\hat{z}$ direction (i.e., $\hat{s} \cdot \hat{z} = \cos\theta \simeq 1$);
- the variation in the phase function is smooth, allowing I_d to be approximated by a second-order Taylor expansion in direction $\hat{s}$ in the spherical integral in Equation (1).

We also assume the following situations, to obtain explicit solutions:

- there is only one type of scattering particles in the media and σ_t, σ_s, and $p(\boldsymbol{s}, \boldsymbol{s}')$ are spacially invariant;
- the scattering media is structured into layers, and the particle density may change along the z axis, described as $\rho(z)$.

We set

$$\hat{r} = \boldsymbol{r} + z\hat{z}, \qquad \hat{s} \simeq \boldsymbol{s} + \hat{z},$$

where $\hat{z}$ is the unit vector in the z direction. Using this approximation, the spherical integral with $\hat{s}$ in Equation (1) can be approximated by a planar integral with a 2D vector $\boldsymbol{s}$. We also assume an axially symmetric phase function around the forward direction:

$$p(\boldsymbol{s}, \boldsymbol{s}') = p(|\boldsymbol{s} - \boldsymbol{s}'|). \quad (3)$$

Using these simplifications, the integro-differential equation in Equation (1) can be replaced by a second-order partial differential equation. Using Fourier transform, we can obtain the solution as

$$\begin{aligned}I_{ri}(z, \boldsymbol{r}, \boldsymbol{s}) &= 1/(2\pi)^4 \int_{-\infty}^{\infty} d\kappa \int_{-\infty}^{\infty} \exp(-i\boldsymbol{\kappa} \cdot \boldsymbol{r}) \cdot \exp(-i\boldsymbol{s} \cdot \boldsymbol{q}) \cdot \\ &\quad F_0(\boldsymbol{\kappa}, \boldsymbol{q} + \boldsymbol{\kappa}z) \cdot \exp(-\sigma_t\zeta(z))dq, \quad (4)\\ I_d(z, \boldsymbol{r}, \boldsymbol{s}) &= 1/(2\pi)^4 \exp(-\rho_n\sigma_a z) \int F_d(z, \boldsymbol{\kappa}, \boldsymbol{q} + \boldsymbol{\kappa}z) \cdot \\ &\quad \exp(-i\boldsymbol{s} \cdot \boldsymbol{q})dq \int \exp(-i\boldsymbol{\kappa} \cdot \boldsymbol{r})d\kappa, \quad (5)\\ F_d(z, \boldsymbol{\kappa}, \boldsymbol{q}) &= \sigma_t F_0(\boldsymbol{\kappa}, \boldsymbol{q}) \int_0^z \rho_n(z')P(\boldsymbol{q} - z'\boldsymbol{\kappa}) \\ &\quad \exp(-\sigma_s\zeta(z')) \exp(-q^2\hat{A}(z, z'))dz', \quad (6)\\ \zeta(z) &= \int_0^z \rho_n(z')dz', \quad (7)\end{aligned}$$

$$\begin{aligned}\tau(z) &= \sigma_t\zeta(z), &(8)\\ \hat{A}(z,z') &= \int_{z'}^{z} A(z'')dz'', &(9)\\ A(z) &= \rho_n(z)(\sigma_s/4) < \theta^2 >, &(10)\\ < \theta^2 > &= \int_{4\pi}\theta^2 p(\boldsymbol{s})ds / \int_{4\pi} p(\boldsymbol{s})ds, &(11)\end{aligned}$$

where $< . >$ represents an average, i indicates the imaginary unit, F_0 is the Fourier transform of the incident light distribution $I_0(\boldsymbol{r},\boldsymbol{s})$

$$F_0(\boldsymbol{\kappa},\boldsymbol{q}) = \int \exp(i\boldsymbol{q}\cdot\boldsymbol{s})ds \int I_0(\boldsymbol{r},\boldsymbol{s})\exp(i\boldsymbol{\kappa}\cdot\boldsymbol{r})dr,$$

and P is the Fourier transform of the phase function p. $\tau(z)$ indicates optical depth at z, and $\boldsymbol{\kappa}$ and $\boldsymbol{q}$ are the Fourier variables corresponding to $\boldsymbol{r}$ and $\boldsymbol{s}$, respectively. Integrals without a specified domain are planar integrals over the whole plane. For more details, readers are invited to refer to the book of Ishimaru [Ish78].

Note that Equation (5) involves a 5D integral. If we want to apply filtering to the image of an object $R(\boldsymbol{r},\boldsymbol{s})$ with the distribution I_d given by

$$\int\int I_d(\boldsymbol{r}-\boldsymbol{r}',\boldsymbol{s}')R(\boldsymbol{r}',\boldsymbol{s}')dr'ds',$$

the 5D integral should be repeated for all $\boldsymbol{r}'$ and $\boldsymbol{s}'$, requiring a 9D integral just to evaluate a single pixel. Consequently, we need further simplifications.

3. Distribution Function

As seen in the previous section, drastic reductions of the computation complexity is necessary to apply narrow beam theory to image synthesis.

The theory assumes that I_d is distributed close to the $\hat{z}$ direction. If the variation in the reflection from the object $R(\boldsymbol{r}',\boldsymbol{s}')$ is small, we can approximate the filtering integral by

$$\begin{aligned}&\int\int I_d(\boldsymbol{r}-\boldsymbol{r}',\boldsymbol{s}')R(\boldsymbol{r}',\boldsymbol{s}')dr'ds'\\ \simeq\ &\int\left(\int I_d(\boldsymbol{r}-\boldsymbol{r}',\boldsymbol{s}')ds'\right)R(\boldsymbol{r}',0)dr'.\end{aligned}$$

In this case, if we can efficiently calculate the filter kernel

$$\Psi(\boldsymbol{r}) = \int I_d(\boldsymbol{r}-\boldsymbol{r}',\boldsymbol{s}')ds', \tag{12}$$

we can reduce the filtering computation.

For further reduction, we took the following approaches:

1. *Analytic Fourier transform:* We selected the incident light I_0 at $z = 0$ and the phase function p so that we could analytically compute their Fourier transforms F_0 and P.
2. *Moment calculation:* We calculated the zeroth- and second-order moments of I_d, which, fortunately, can be calculated very efficiently.
3. *Fitting with a heuristic function:* We carefully selected a heuristic function that gives a good approximation to the distribution. We set the parameters of the function so as to have the same calculated moment values.

We will now detail these three reductions in the following sections.

3.1. Incident Ray and Gaussian Phase Function

Let us specify the properties of participating media and the lighting conditions, and proceed with the calculations. Since we aim to evaluate the distribution around a viewing ray, it is natural to assume that the incident light be a ray, which we represent by Dirac delta functions

$$I_0(\boldsymbol{r},s) = \delta(\boldsymbol{r})\delta(\boldsymbol{s}).$$

The phase function is assumed to be Gaussian[†] and is set to

$$\begin{aligned}p(\boldsymbol{s}) &= 1/(4\pi B)\exp(-|\boldsymbol{s}|^2/4B), &(13)\\ \int p(\boldsymbol{s})ds &= 1.\end{aligned}$$

Then, both Fourier transforms have analytical forms, such that

$$\begin{aligned}F_0(\boldsymbol{\kappa},q) &= 1, &(14)\\ P(\boldsymbol{q}-z'\boldsymbol{\kappa}) &= \exp(-B(\boldsymbol{q}-z'\boldsymbol{\kappa})^2). &(15)\end{aligned}$$

Putting these expressions into Equation (5) yields a line integral

$$\begin{aligned}I_d(z,r,s) =\ &K_G(z)\exp(-|\boldsymbol{s}|^2)/(4\Sigma_G^2(z))\\ &\int_0^z(\pi/\Sigma_G^2)\exp(-\rho_n\sigma_s z')dz'(\pi/C^2(z'))\\ &\exp\left\{-(\boldsymbol{r}-z\boldsymbol{s}+Bz'\boldsymbol{s}/\Sigma_G^2)^2/(4C^2(z'))\right\}\\ =\ &K_G(z)\exp\left\{-|\boldsymbol{s}|^2/(4\Sigma_G^2)\right\}\cdot\pi^2\\ &\int_0^z dz'\exp(-\rho_n\sigma_s z')/(\Sigma_G^2(z')\cdot C^2(z'))\\ &\exp\left\{-(\boldsymbol{r}+(-z+Bz'/\Sigma_G^2(z')\boldsymbol{s})^2/(4C^2(z'))\right\},\end{aligned} \tag{16}$$

where

$$\begin{aligned}\hat{A}(z,z') &= \int_{z'}^{z} A(z'')dz''\\ &= (\sigma_t/4) < \theta^2 > (\zeta(z)-\zeta(z')), &(17)\\ \Sigma_G^2(z') &= B+\hat{A}(z,z'), &(18)\\ C^2(z') &= Bz'^2[1-B/\Sigma_G^2], &(19)\\ K_G(z) &= (1/2\pi)^4\sigma_s\exp(-\rho_n\sigma_a z). &(20)\end{aligned}$$

The direct component I_{ri} can be calculated from

$$I_{ri}(z,\boldsymbol{r},\boldsymbol{s}) = \delta(\boldsymbol{r}-z\boldsymbol{s})\delta(\boldsymbol{s})\exp(-\tau(z)). \tag{21}$$

Note that $\tau = (\rho_n\sigma_t)z$ when ρ_n is constant. More details can be found in Section 2 of the associated supplemental document.

3.2. Moments

To calculate the distribution, the line integral of Equation (16) has to be evaluated for all $\boldsymbol{r}$ and $\boldsymbol{s}$. In order to estimate it more quickly,

[†] A sum of Gaussians would also be possible for a more accurate representation.

let us calculate the zeroth- to second-order moments of I_d

$$M_0(z) = \int\int I_d(z,\boldsymbol{r},\boldsymbol{s})dsdr, \qquad (22)$$
$$M_1(z) = 0, \qquad (23)$$
$$M_2(z) = \int\int \rho^2 I_d(z,\boldsymbol{r},\boldsymbol{s})dsdr. \qquad (24)$$

Here, the first-order moment (center of mass) is null because of its symmetry.

When the density ρ_n is constant, we have closed-form formulae

$$\begin{aligned} M_0(z) &= \exp(-\rho_n\sigma_a z)(1-\exp(-\rho_n\sigma_s z)), & (25)\\ M_2(z) &= 4\exp(-\rho_n\sigma_a z) \\ &\quad \Big[\exp(-\rho_n\sigma_s z)\left\{Az^2/a - 2B/a^2\right\} + \\ &\quad \left\{2B/a^2 - 2Bz/a + (-A/a+B)z^2 + Az^3\right\}\Big], & (26)\\ a &= \rho_n\sigma_s. & (27)\end{aligned}$$

When density is not uniform but is dependent on z, we have a line integral

$$\begin{aligned} M_0(z) &= \exp(-\sigma_a\zeta(z))(1-\exp(-\sigma_s\zeta(z))), & (28)\\ M_2(z) &= 4\int_0^z (\rho_n(z')\sigma_s)\exp(-\sigma_s\zeta(z')) \\ &\quad \left[C(z')^2 + (z - Bz'/\Sigma_G^2(z'))^2\Sigma_G^2(z')\right]dz'. & (29)\end{aligned}$$

As seen in Equations (26) and (29), the moments can be efficiently estimated in constant time in homogeneous media, and in linear time in heterogeneous media. More details can be found in Section 3 of the supplemental document.

3.3. Heuristic Distribution Function

The distribution function $\Psi(|\boldsymbol{r}|)$ is set according to the moments M_0 and M_2. This allows for efficient estimation of the distribution without integrating Equation (16) for each value of $\boldsymbol{r}$ and $\boldsymbol{s}$. Because the distribution functions include single-scattering components, the shape of the single-scattering distributions can help us to design our heuristic function.

Let us calculate the single scattering component I_1 of I_d from

$$\begin{aligned} I_1(z,\boldsymbol{r},\boldsymbol{s}) &= (\rho_n\sigma_t)\int_0^z \exp(-\sigma_t\rho_n y)dy \\ &\quad \int_{4\pi} I_0(z-y,\boldsymbol{r}+y\boldsymbol{s},\boldsymbol{s}')p(\boldsymbol{s},\boldsymbol{s}')ds' \\ &= (\rho_n\sigma_t)\exp(-\sigma_t\rho_n z)\int_0^z \delta(\boldsymbol{r}+y\boldsymbol{s})p(\boldsymbol{s})dy. & (30)\end{aligned}$$

The integral over $\boldsymbol{s}$

$$\Psi_1(z,\boldsymbol{r}) = \int I_1(z,\boldsymbol{r},\boldsymbol{s})ds, \qquad (31)$$

can be calculated from

$$\begin{aligned} \Psi_1(z,\boldsymbol{r}) &= (\rho_n\sigma_s)\exp(-\sigma_t\rho_n z)\cdot(1/B)\cdot \\ &\quad (1/|\boldsymbol{r}|)(1/\sqrt{\pi})G(|\boldsymbol{r}|/(z\sqrt{2B})), & (32)\end{aligned}$$

using the finite integral of a Gaussian function (error function) G

$$G(x) = 1/\sqrt{2\pi}\int_x^\infty \exp(u^2/2)du. \qquad (33)$$

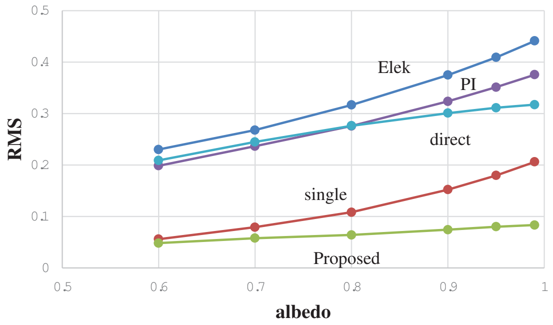


Figure 2: *RMS errors and albedo (σ_s/σ_t).*

We regard this as a fundamental feature of the distribution function, and adopt a heuristic distribution function described by

$$\Psi(z,\boldsymbol{r}) = \alpha(z)(1/|\boldsymbol{r}|)G(|\boldsymbol{r}|/\beta(z)). \qquad (34)$$

Parameters α and β can be determined from the moments M_i, calculated using Equations (26) or (29)

$$\alpha(z) = 2M_0/\beta^2(z), \qquad (35)$$
$$\beta^2(z) = 3(M_2(z)/M_0(z)). \qquad (36)$$

Using these formulae, we can determine the filtering kernel in constant time for homogeneous density distributions, and in linear time for heterogeneous distributions. The details are provided in Sections 4 and 5 of the supplemental document.

4. Experiments

4.1. Point Spread Functions

With the configuration shown in Figure 1, we randomly launched many photons at the origin along the z-axis, simulated their scattering, recorded their traces, and counted the number of photons passing through the cells (we refer to these simulations as photon counting). We compared the results calculated from our heuristic function and the moment equations (Equations (34), (35), and (36)), with the results measured by photon counting. We fixed $\sigma_t = 1.0$ and used a Gaussian phase function with various spread angles $d\theta^2 = 4B$. Figure 2 shows the measured RMS errors with respect to albedo (σ_s/σ_t) values. The phase function was fixed to $d\theta = 30$ degrees. We also calculated the direct light component using Equation (21), the single-scattering component from Equation (32), the distribution based on the path-integral (PI) method, and Elek et al.'s version [ERS13]. The distribution functions used in the PI method and Elek's method are provided in Section 1 of the supplement document (Equations (14) to (16)). As seen in Figure 2, for lower albedo values, the errors caused by all methods are relatively small because single-scattering components are dominant. Although errors grew up with higher albedo values in general, the proposed method keeps a reasonable accuracy.

Figure 3 shows as color-coded the distributions measured by photon counting and those calculated by our proposed method,

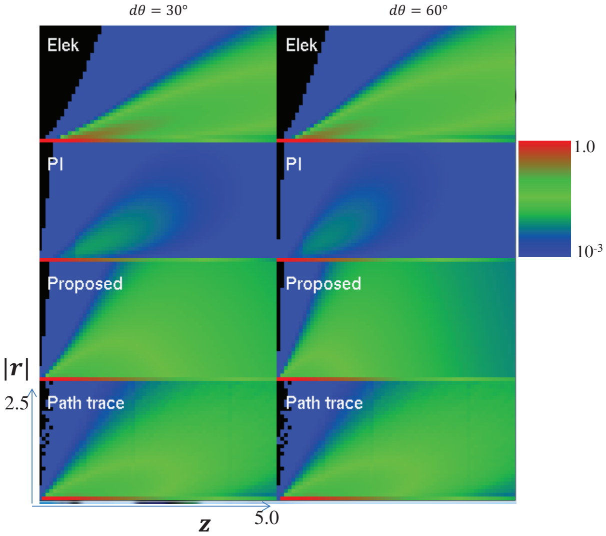


Figure 3: *Distribution functions measured by photon counting (path tracing), calculated by our (proposed) method, and by the methods of (PI) Premoze et al. [PAS03] and (Elek) Elek et al. [ERS13], all displayed with the same pseudo-colors. The spread angle of the phase function, dθ, is 30 degrees (left) and 60 degrees (right). Note that the axes are permuted, compared to Figure 1.*

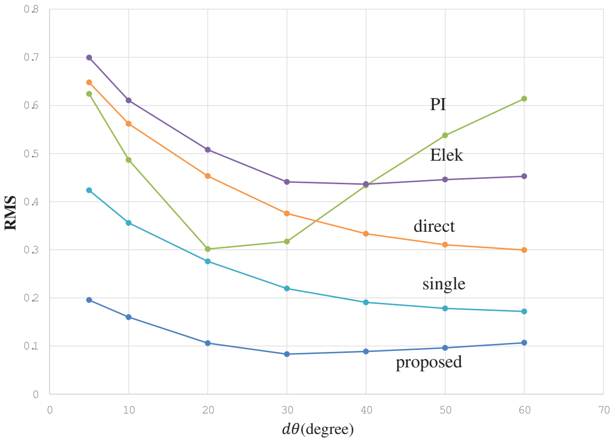


Figure 4: *RMS errors and spread angle of the phase function dθ.*

with two different phase functions. The circular integral $2\pi \cdot r \cdot \Psi$ is shown instead of Ψ, because Ψ itself is singular at the origin ($r = 0$). The distributions become broader with respect to the depth z and the variance of the phase function $d\theta$. Observe how the distributions calculated by our proposed method agree much better with the simulated reference solutions than the other two methods.

Figure 4 shows root-mean-square (RMS) errors. The errors from our proposed method are mostly around 10%. Observe how the accuracy has largely been improved by taking into account the multiple-scattering terms. The figure indicates significant improvements by the proposed distribution functions.

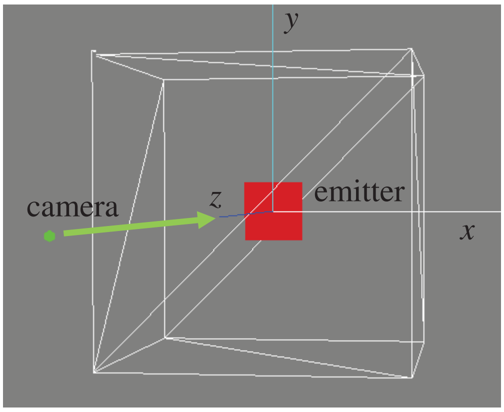


Figure 5: *Simple scene used for the comparison.*

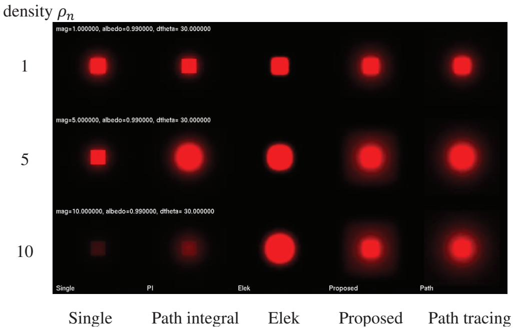

density ρ_n
1
5
10
mag=1.000000, albedo=0.990000, dtheta= 30.000000
mag=5.000000, albedo=0.990000, dtheta= 30.000000
mag=10.000000, albedo=0.990000, dtheta= 30.000000
Single
PI
Elek
Proposed
Path
Single
Path integral
Elek
Proposed
Path tracing

Figure 6: *Resulting images of the square light emitter. Applied computational methods are (from left to right) single scattering, the path integral, Elek et al.'s, the proposed method, and path tracing. The density is set to (from top to bottom) 1.0, 5.0, and 10.0.*

4.2. Rectangular Emitter Images

To evaluate accuracy, we rendered images of a very basic scene with path tracing and with our proposed method, and compared

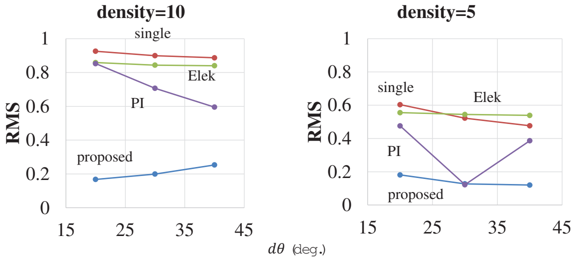


Figure 7: *RMS errors for particle density values (ρ_n) of 10.0 and 5.0.*

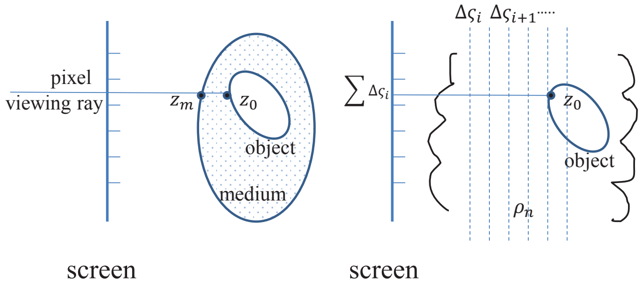


Figure 8: *Calculation of the distribution functions in (left) homogeneous media and in (right) heterogeneous media.*

the synthesized images. Figure 5 illustrates the configuration of the scene, where a red square light emitter is located in a transparent box filled with homogeneous scattering media. For path tracing, we traced about 10K paths per pixel, which took a couple of hours per image. For reference purposes, we also rendered images with single scattering, path-integration filter kernels, and Elek et al.'s version (Equation 12 in the supplemental document).

Figure 6 shows the corresponding results. When the density is low, the direct ray components are dominant, and differences between the generated images are rather subtle. However, with increases in density, differences become more significant: the images produced by single scattering and path integration are too dark, while our proposed method captures the blurred light caused by multiple scattering. Figure 7 shows RMS errors measured in two conditions. The filtering based on path integration does a better job than the single-scattering approximation, but still causes large errors at higher densities. Elek et al.'s distribution function has larger errors than those of the path integral in general, possibly because of its additional simplification. With our proposed method, on the other hand, errors lie around 10 to 20%, and we observe reasonable agreements with the path-tracing results.

5. Application to Screen-space Scattering Simulations

To demonstrate advantages of the proposed distribution function, it is applied to screen-space scattering simulations, as introduced by Elek et al. [ERS13].

5.1. Method

5.1.1. Distribution

The distribution Ψ can be calculated from the moments M_0 and M_2. In the case of homogeneous media, the moments are calculated with Equation (24), using only depth values of media and object surfaces. To do so, we save depth values in a preprocessing stage and refer to these values when the moments are evaluated, as illustrated in Figure 8 (left).

In heterogeneous media, the moments are calculated with Equation (29), where the integral is calculated by referring to the accumulated density ζ. Therefore, we perform the integral by setting sample planes parallel to the screen, and simultaneously sum ζ and M_2 in an alpha-blending fashion, as illustrated in Figure 8 (right).

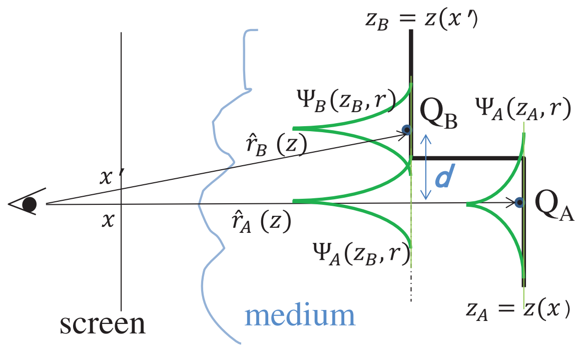
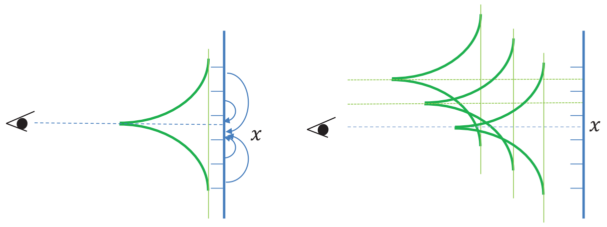


Figure 9: *Kernel selection.*



(a) gathering filter (b) distributing filter

Figure 10: *Filtering implementation with the gathering and distributing schemes.*

5.1.2. Filtering

Elek et al. [ERS13] propose an efficient multi-scale filtering scheme for real-time execution. In this paper, we employ a simpler and more generic filtering scheme to avoid potential artifacts.

Consider the configuration shown in Figure 9, in which pixel value I_{filter} at pixel x is evaluated from

$$I_{filter}(x) = \sum_{x'} w(x,x')R(x'),$$

where x and x' denote pixel positions, z_A and z_B are the object depth values, and w and R represent the filter kernel and the object image, respectively.

Along the central ray $\hat{r}_A(z)$ passing through pixel x, the distribution $\Psi_A(z,|\boldsymbol{r}|)$ varies with respect to depth z. Without loss of generality, let us assume $z_B < z_A$. In this case, some of the scattered viewing rays reach object point B, which can be evaluated from the distribution $\Psi_A(z_B,\boldsymbol{r})$. Therefore, we can set weight w as

$$\begin{aligned} w(x,x') &= \Psi_A(z_B,d_r), && (37)\\ d_r &= |\hat{r}_A(z_B) - \hat{r}_B|. && (38) \end{aligned}$$

Furthermore, if we assume local similarity for nearby pixels

$$\Psi_A(z,|\boldsymbol{r}|) \simeq \Psi_B(z,|\boldsymbol{r}|),$$

we can set w as

$$w(x,x') \simeq \Psi_B(z_B,d) \qquad (39)$$

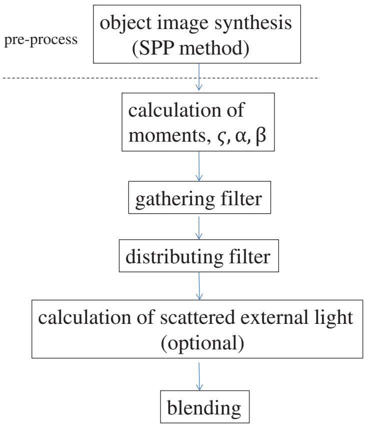

pre-process
object image synthesis (SPP method)
calculation of moments, ς, α, β
gathering filter
distributing filter
calculation of scattered external light (optional)
blending

Figure 11: *Pipeline of our rendering system.*

for $z_B < z_A$. This switching is necessary to avoid “illuminance leaking” as discussed by Elek et al. [ERS13].

In general, filtering can be implemented in two ways: gathering contributions from other pixels, or distributing their contribution to other pixels, as shown in Figure 10. We implemented filtering in both ways depending on the z-value comparison. For those pixels with larger depth, we apply the gathering filter; for those with smaller depth, we use the filtering result from the distributing filter.

5.1.3. Implementation

We implemented a rendering system, which is outlined in Figure 11.

Preprocess: In a preprocessing stage, the shaded image $col[]$ and depth map $zbuf[]$ of the scene’s objects are calculated using conventional methods. The shaded image will be filtered by the distribution function estimated in the following main process. Although illumination from an external source may be evaluated by simple attenuation, we adopted the simplified plane-parallel (SPP) model [SSD*10] to capture multiple-scattering features of the light distributed in the scene.

Calculation of moments: Next, the moments and parameters of distribution Ψ, α and β, are calculated at each pixel according to Equations (25), (29), (35), and (36). We implemented this in three passes (Algorithm 1). First, we prepare a set of sample planes perpendicular to the viewing direction. Each sample plane is drawn and the density ρ_n is accumulated via alpha-blending up to the object depth, resulting in ζ_0. Second, we accumulate the second-order moments M_2, using ζ_0. Finally we obtain the parameters of distributions, α and β.

Gathering filter: Using the calculated filter kernels, both gathering filter and distributing filter are applied to every pixel. For those pixels with a larger depth value than that of the pixel in process, we apply the gathering filter (Algorithm 2). We draw points corresponding to each pixel through the vertex shader, and the fragment shader processes the filtering. We limit the size of the filter at pixel p by simply multiplying $\beta[p]$ by user-specified factor cut_{off}, which was set to 1.0 during our experiments. For all pixels p' within the filter, we calculate the filter weight, $weight$, and sum up the weighted colors if the depth $zbuf[p']$ is larger than $zbuf[p]$. The error function $G(x)$ can be evaluated by Hastings’ approximation [Has55], for example.

Algorithm 1 Calculation of moments

```
Require: $zbuf[], \rho_n[]$ /* [] indicates an array */
Ensure: $\alpha[], \beta[], \zeta_0[]$
  initialize: $M_0[], M_2[]\zeta_0[], \zeta \Leftarrow 0$
  /* Draw sample planes through vertex shader */
  for all sample planes $z_plane$ do
    /* Process in fragment shader */
    for all pixel $p$ do
      if $zbuf[p] > z_{plane}$ then
        $\zeta_0 += \rho_n * dz$
      end if
    end for
  end for
  for all sample planes $z_{plane}$ do
    for all pixel $p$ do
      if $zbuf[p] > z_{plane}$ then
        $z = zbuf[p], z' = z_{plane}$
        $\zeta += \rho_n * \sigma_t * dz$
        $\hat{A} = \sigma_s B(\zeta_0[p] - \zeta)$
        $\Sigma_G^2(z') = B + \hat{A}$ /* Eq. (18) */
        $C^2(z') = Bz'^2[1 - B/\Sigma_G^2]$ /* Eq. (19) */
        $e_0 = (\rho_n(z')\sigma_s)\exp(-\sigma_s\zeta)$
        $e_1 = [C(z')^2 + (z - Bz'/\Sigma_G^2)^2\Sigma_G^2)]$
        $M_2[p] += e_0 * e_1$ /* Eq. (29) */
      end if
    end for
  end for
  for all pixel $p$ do
    $M_0[p] = \sigma_t \exp(-\sigma_a\zeta(z))(1.0 - \exp(-\sigma_s\zeta_0))$
    $\beta[p] = 3 * (M_2[p]/M_0)$ /* Eq. (36) */
    $\alpha[p] = 2 * M_0/\beta[p]$ /* Eq. (35) */
  end for
```

Distributing filter: For those pixels with a larger depth value than that of the pixel in process (p), we apply the distributing filter (Algorithm 3). In the vertex shader, we calculate the filter size for each pixel and draw a rectangle centered at the location of the pixel in process covering the filter. In the fragment shader, we calculate the filter weight if the depth $zbuf[p']$ is larger than $zbuf[p]$. The weighted colors are summed up by alpha-blending.

Calculation of scattered external light: Since this screen-space filtering cannot capture effects from light sources out of the image, we optionally calculate their contributions using conventional methods. This may be conducted by a standard ray-marching method capturing single-scattering components. In our implementation, we adopted the SPP method as in the pre-processing phase.

Algorithm 2 Gathering filter

```
Require: $zbuf[], col[], \alpha[], \beta[], cut_{off}$
Ensure: $I_{gather}[]$
  initialize: $I_{gather} \Leftarrow 0$
  /* draw pont and process in fragment shader */
  for all pixel $p$ do
    $z_0 = zbuf[p]$
    $\boldsymbol{r} = WorldCoord(p)$ /* transform to world coordinate */
    $FilterSize = cut_{off} * \beta[p]$ /* filter size */
    for all pixel $p'$ within the filter centered at $p$ do
      if $zbuf[p'] > z_0$ then
        $\boldsymbol{r}' = WorldCoord(p), d_r = |\boldsymbol{r}' - \boldsymbol{r}|$
        $weight = (\alpha[p]/d_r)G(d_r/\beta[p])$ /* Eq. (34) */
        $I_{gather}[p] += weight * col[p']$
      end if
    end for
  end for
```

Algorithm 3 Distributing filter

```
Require: $zbuf[], col[], \alpha[], \beta[], cut_{off}$
Ensure: $I_{distr}[]$
  initialize: $I_{distr}[] \Leftarrow 0$
  /* draw rectangles centered at each pixels via vertex shader */
  for all pixel $p$ do
    $z_0 = zbuf[p], I_0 = col[p]$
    $\boldsymbol{r} = WorldCoord(p)$ /* transform to world coordinate*/
    $FilterSize = cut_{off} * \beta[p]$ /* filter size */
    /* draw a rectangle and process filtering in fragment shader */

    for all pixel $p'$ within the filter rectangle do
      if $zbuf[p'] > z_0$ then
        $\boldsymbol{r}' = WorldCoord(p), d_r = |\boldsymbol{r}' - \boldsymbol{r}|$
        $weight = (\alpha[p]/d_r)G(d_r/\beta[p])$ /* Eq. (34) */
        $I_{distr}[p'] += weight * I_0$
      end if
    end for
  end for
```

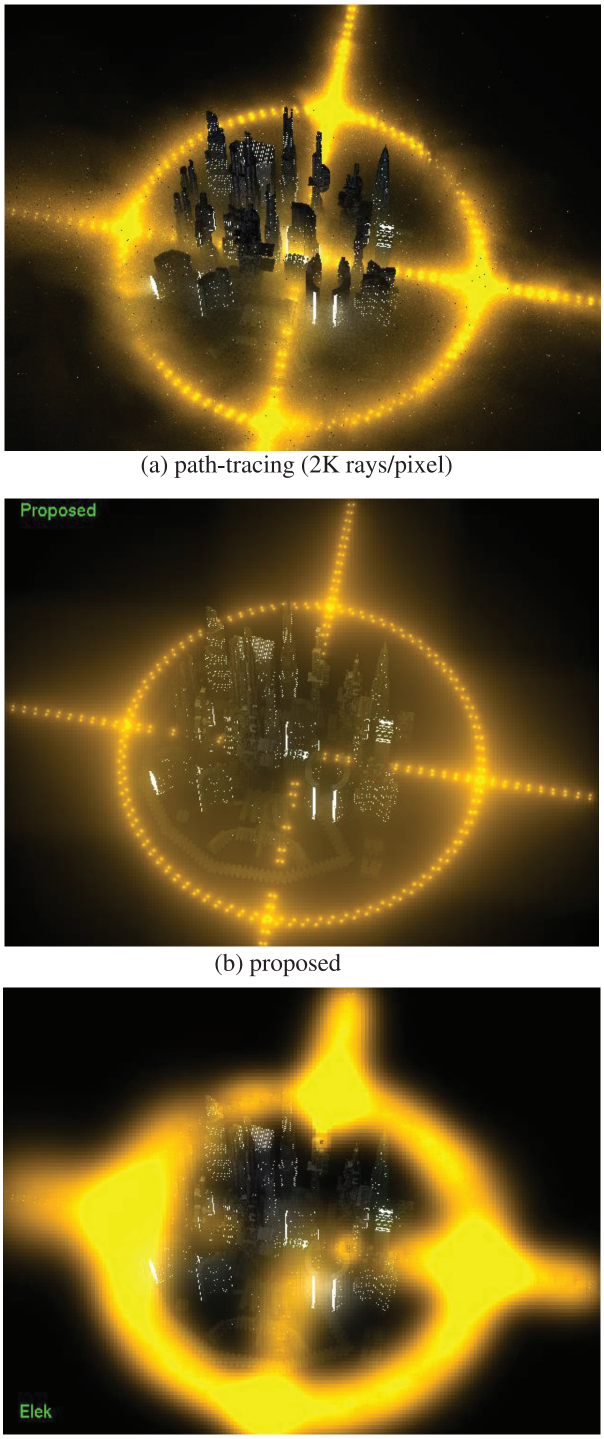
Blending: Finally, we bend the calculated components into the final image (Algorithm 4).

Algorithm 4 Blending

```
Require: $\zeta_0[], col[], I_{distr}[], I_{gather}[], I_{ext}[]$
Ensure: $I_{final}[]$
  /* draw points corresponding to pixels*/
  for all pixel $p$ do
    $T = \exp -\zeta_0[p]$
    $I_{final}[p] = T * col[p] + I_{distr}[p] + I_{gather}[p] I_{ext}[p]$
  end for
```

5.2. Synthesized Images

We synthesized images of scenes with heterogeneous scattering media.

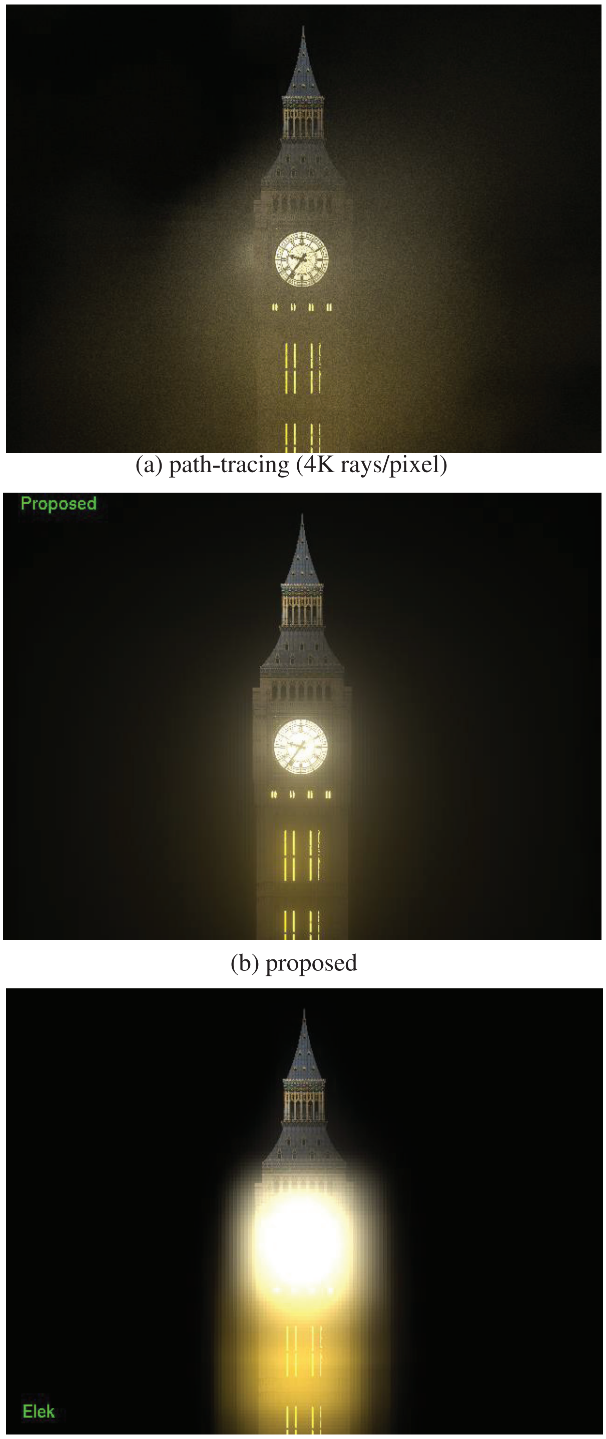


(a) path-tracing (2K rays/pixel)

(b) proposed

(c) Elek's

Figure 12: *Comparison for the CityLight images.*

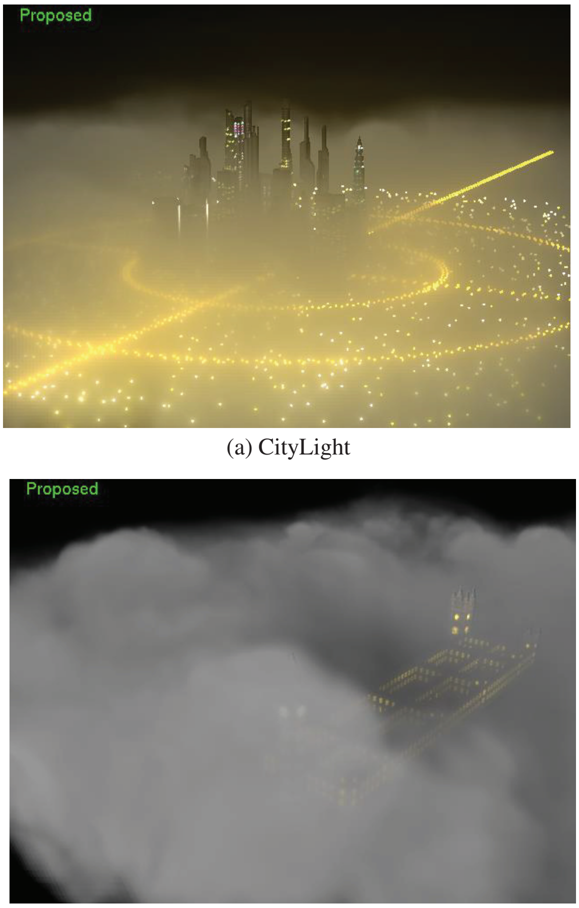


(a) path-tracing (4K rays/pixel)

(b) proposed

(c) Elek's

Figure 13: *Comparison for the Westminster images.*



(a) CityLight

(b) Westminster

Figure 14: *Example from the video sequence.*

Figure 12 shows a night scene, rendered by (a) path tracing, (b) the proposed distribution function, and (c) Elek et al.'s [ERS13] approximation. In the scene, buildings with colored lit windows are partially immersed in thick heterogeneous fog. Light in the lower parts of buildings are more attenuated and blurred by the fog, which was successfully simulated by our proposed method, which reasonably agrees with the result from path tracing. The RMS difference from the path-traced image is 28.6%. Elek et al.'s kernel is more concentrated in its center, and over-estimates the blurring near light sources; its RMS difference is 52.7%.

Figure 13 shows another familiar example, where a famous clock tower stands in heavy heterogeneous fog. The image rendered by our proposed function reasonably agrees with the one by path tracing, demonstrating significant improvements.

We also created fly-through sequences using these scenes. Representative frames from the sequences are shown in Figure 14. Since this screen-space method cannot capture effects from out-of-screen light sources, we set a weak parallel light, calculated its scattered intensity by the SPP method [SSD*10], and blended it to the filtered images. Varying fog density was created from a fluid simulation, and stored as voxel data.

As an indication of typical rendering times, complete processing times were 110 msec and 100 msec, for Figure 14-(a) and (b), respectively, on a Windows PC (Intel Core i7-3770K @3.50GHz, GeForce GTX 690), while path tracing took several days. The image size is 640 × 480. The most time-consuming part of the algorithm is the linear filtering process, especially for large kernel sizes. To accelerate this process, we utilized multi-resolution images, similar to a MIP map, and applied a filter to a coarser image when the kernel size is large (larger than 9 pixels in our implementation).

5.3. Limitations

As mentioned in Section 2, the theory assumes narrow and smooth distribution of scattering light as well as layered, single-type scattering media. Screen-space algorithms, in general, assume that everything necessary is visible on screen and do not count contributions from invisible/out-of-screen light sources. These assumptions limit applicable scenes and the accuracy of the method, for example, in the following situations:

- *Mixed material:* Scattering media composed with more than two different materials, such as milk poured into coffee, may have spatially varying mixture ratios, which makes the phase function spatially variant and hard to be correctly handled by the theory.
- *Occlusion/shadowing:* Occlusion causes sharp changes in light distribution, which the theory cannot correctly handle. The proposed method tries to reduce artifacts by switching gathering and distributing filters based on depth comparisons, but errors still remain.
- *Large variation of density:* When the density function is not layered and variation perpendicular to viewing direction is large, the estimated distribution may involve large errors.
- *Very small particles:* Particles much smaller than the wavelength tend to have strong backscattering properties as in Rayleigh scattering. The theory ignores backward-scattering components and the method may produce large errors.
- *External light sources:* Lighting effects due to external light sources out of screen are not taken into account by the filtering process. The proposed method separately estimates the external light components and simple blending is applied, which may cause errors.

Note that these general limitations are related to the filtering approach, which affect also Elek's method [ERS13].

6. Conclusion

In this paper, we proposed useful beam distribution functions in heterogeneous scattering media. Based on the narrow beam theory, we formulated the moments of the distribution and proposed a heuristic distribution function to fit to the calculated moments. Experiments show that the proposed functions are more accurate than WKB-based functions previously used. The proposed function was successfully applied to screen-space simulations of object blurring due to multiple scattering, and demonstrated significant image improvements over previous functions.

The proposed distribution functions can be also applied to multiple scattering of directional light sources, such as car headlights, shafts of light through clouds, or city light beams in the night sky. They are also suitable for other image synthesis applications, such as Premoze et al.'s global illumination [PAS03], and BSSTDF calculations in heterogeneous media.

Acknowledgements

The authors would like to thank the reviewers for their careful reviews and helpful comments. They also thank Pierre Poulin for his technical dicussions and linguistic advices, and Tatsuki Kawaguchi for his scene models used in the experiments. This research was partially supported by JSPS KAKENHI, Grant-in-Aid for Scientific Research on Innovative Areas, Grant Number JP15H05924.

References

[APRN04] Ashikhmin M., Premoze S., Ramamoorthi R., Nayar S.: *Blurring of light due to multiple scattering by the medium: a path Integral approach*. Tech. Rep. CUCS-017-04, Columbia University, 2004. 2

[dI11] d'Eon E., Irving G.: A quantized-diffusion model for rendering translucent materials. *ACM Trans. on Graphics 30*, 4 (2011), 56. 1

[DJ05] Donner C., Jensen H. W.: Light diffusion in multi-layered translucent materials. *ACM Trans. on Graphics 24*, 3 (2005), 1032–1039. 1

[ERS13] Elek O., Ritschel T., Seidel H.: Real-time screen-space scattering in homogeneous environments. *IEEE Computer Graphics and Applications 33*, 3 (2013), 53–65. 1, 4, 5, 6, 7, 9, 10

[FHK14] Frisvad J. R., Hachisuka T., Kjeldsen T. K.: Directional dipole model for subsurface scattering. *ACM Trans. on Graphics 34*, 1 (2014), 5:1–12. 1

[Has55] Hastings C.: *Approximations for Digital Computers*. Princeton University Press, 1955. 7

[Ish78] Ishimaru A.: *Wave propergation and scattering in random media*. Academic Press, 1978. 1, 2, 3

[JC98] Jensen H. W., Christensen P.: Efficient simulation of light transport in scenes with participating media using photn mapping. In *Proc. SIGGRAPH* (1998), ACM, pp. 311–320. 1

[JMLH01] Jensen H. W., Marschner S. R., Levoy M., Hanrahan P.: A practical model for subsurface light transport. In *Proc. SIGGRAPH* (2001), ACM, pp. 511–518. 1

[KVH84] Kajiya J., Von Herzen B. P.: Ray tracing volume densities. In *Computer Graphics (Proc. SIGGRAPH)* (1984), vol. 18, ACM, pp. 165–174. 1

[LRT82] Langouche F., Roekaerts D., Tirapegui E.: *Functional Integration and Semiclassical Expansions*. D. Reidel Publishing Company, 1982. 1

[PARN04] Premoze S., Ashikhmin M., Ramamoorthi R., Nayar S.: Practical rendering of multiple scattering effects in participating media. In *Proc. Eurographics Symp. on Rendering* (2004), Eurographics, pp. 130–140. 2

[PAS03] Premoze S., Ashikhmin M., Shirley P.: Path integration for light transport in volumetric materials. In *Proc. of Eurographics Symp. on Rendering* (2003), Eurographics, pp. 130–140. 1, 2, 5, 10

[SSD*10] Shinya M., Shiraishi M., Dobashi Y., Iwasaki K., Nishita T.: A simplified plane-parallel scattering model and its application to hair rendering. In *Proc. Pacific Graphics* (2010), ACM, pp. 85–92. 7, 9

[YIC*10] Yue Y., Iwasaki K., Chen B.-Y., Dobashi Y., Nishita T.: Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Trans. on Graphics 29*, 6 (2010), 177:1–8. 1