Journal of Information Processing Vol.21 No.2 349–357 (Apr. 2013)

[DOI: 10.2197/ipsjjip.21.349]

Regular Paper

A Simplified Plane-parallel Scattering Model for Rendering Densely Distributed Objects such as Foliage

Mikio Shinya[1,a)] Yoshinori Dobashi[2] Kei Iwasaki[3] Michio Shiraishi[1] Tomoyuki Nishita[4]

Received: July 25, 2012, Accepted: January 11, 2013

Abstract: Fast computation of multiple reflections and scattering among complex objects is very important in photorealistic rendering. This paper applies the plane-parallel scattering theory to the rendering of densely distributed objects such as trees. We propose a simplified plane-parallel scattering model that has very simple analytic solutions, allowing efficient evaluation of multiple scattering. A geometric compensation method is also introduced to cope with the infinite plane condition, required by the plane-parallel model. The scattering model was successfully applied to tree rendering. Comparison with a Monte Carlo method was made and reasonable agreement was confirmed. A rendering system based on the model was implemented and multiple inter-reflections were effectively obtained. The view-independent feature of the model allows fast display of scenes. The pre-computation is also modest, permitting interactive control of lighting conditions.

Keywords: image synthesis, multiple scattering, plane-parallel model, foliage rendering

1. Introduction

Even with the latest graphics hardware, fast rendering of complex scenes is still a challenging task when inter-reflection/transmission among objects is taken into account. Global illumination algorithms such as radiosity, path tracing, and photon-mapping usually involve a huge number of visibility evaluations, which limits the performance. However, if a scene consists of densely distributed small objects, it is possible to regard them as continuous scattering media and to avoid visibility estimation by applying volumetric scattering techniques. In this case, visibility is averaged over the volume into scattering parameters, such as the total cross section (σ_t), and individual visibility among objects can be neglected. Such a volumetric approximation is known to be efficient for foliage [1], hair [2], knitwear [3] and so on.

Energy transport in scattering media can be described by an integro-partial differential equation, called the volume rendering equation. Although it is hard to solve exactly, there are useful approximation theories such as the diffusion theory and the plane-parallel theory [4].

The plane-parallel theory deals with uniformly illuminated media with infinite parallel planar boundaries. For such media, the volume rendering equation can be described as a linear ordinary integro-differential equation. By discretizing directions, the equation becomes a set of first order ordinary differential equations, which can be analytically solved via eigenvalue decomposition, in homogeneous cases. Max et al., applied the plane-parallel theory to the rendering of vegetation in which the equation is solved by the Runge-Kutta method in the Fourier domain [1].

The plane-parallel solution is complete for infinite slabs and well approximates dense vegetation. However, because the numerical process involved is still slow for runtime evaluation, the solution has to be calculated as a pre-process and stored in tables. Since the resulting radiance depends on many variables, such as the incident light direction, the scattering light direction, depth, and so on, it is necessary to use high dimensional tables, which can be huge and impinge on the flexibility. A more serious limitation is that the infinite hypothesis is unrealistic for those trees not completely surrounded by other trees.

Recently, a simplified plane-parallel (SPP) scattering model was reported and applied to hair rendering [5]. In this paper, we present more detailed discussions on the SPP model and decribe its application to tree rendering. The simplification was made by setting the cosine factor in the governing equation as a constant based on the observation that a large part of the scattering power is localized near to the incident light direction. This simplified plane-parallel model has very simple analytic solutions which reasonably approximate the original exact solutions especially in the case of isotropic scattering. To cope with the infinite hypothesis, we propose a geometric compensation factor, based on an analysis of second order scattering. The model was applied to tree rendering. The calculated radiance agrees reasonably with that obtained using a Monte Carlo method. The computational cost was very similar to the conventional single scattering models. Interactive speed was achieved in dynamic lighting environments.

In the following sections, Section 2 sets out the related work and Section 3 describes the simplified plane parallel model. Section 4 describes the geometric compensation and Section 5

[1] Toho University, Funabashi, Chiba 274–8510, Japan
[2] Hokkaido University, Sapporo, Hokkaido 060–8628, Japan
[3] Wakayama University, Wakayama 640–8510, Japan
[4] The University of Tokyo, Kashiwa, Chiba 277–8561, Japan
[a)] shinya@is.sci.toho-u.ac.jp

presents the experiments and the rendering system while the results are shown in Section 5.

2. Related Work

The literature regarding global illumination and scattering simulation is too extensive to list here, so we focus on work that utilizes scattering techniques for rendering densely distributed objects. Self-shadowing of complex objects, such as hair and trees, is often treated as a single scattering event by using, for example, the deep shadow map [6], the opacity map [7], and the deep opacity map [8].

Multiple reflection/transmission is recognized as crucial for realistic rendering of trees and hair. Qin et al., developed an approximation to capture multiple reflection among leaves, assuming local two-flux inter-reflection between neighboring parallel surfaces [9]. Dual scattering [2] applies a similar idea to hair rendering and successful results have been obtained. However, both methods strongly rely on model-based heuristics. Boulanger et al., presented a fast rendering method, taking second order scattering into account [10]. Moon et al., applied photon mapping to hair rendering [11] and encoded it by spherical harmonics [12]. The results are impressive, but the pre-computation was heavy and it is thus difficult to apply it to dynamic lighting and scenes. Max et al., applied plane-parallel scattering theory to tree rendering. The concept was very stimulating, and the potential of the plane-parallel theory was clearly demonstrated. However, problems related to the infinite plane assumption have limited its practical application.

The plane-parallel theory is ideal for calculating the BRDF (Bidirectional Reflectance Distribution Function) of scattering materials and has been applied to shading models for skin [14] and leaves [13].

3. Simplification of the Plane-parallel Scattering Theory

This section briefly reviews the plane-parallel scattering theory, and then presents a simplified plane-parallel model. The simplified plane-parallel model uses a pre-defined intensity distribution, which provides a view-independent feature to the rendering system and allows fast display.

3.1 Plane-parallel Theory

Light energy propagates through scattering media via repeated scattering and absorption events. The irradiance $I(x, s)$ at x in the direction s satisfies the volume rendering equation:

$$(\nabla \cdot s)I(x, s) = -\sigma_t I(x, s) + \sigma_s \int_\Omega p(s, s')I(x, s')ds', \qquad (1)$$

where σ_t, σ_s represents the total cross section, and the scattering cross section respectively, while Ω denotes the unit sphere. The function p is the phase function, often defined by Henyey-Greenstein's function

$$p(s, s') = (1/4\pi)(1 - g^2)/(1 - 2g(s \cdot s') + g^2)^{3/2}, \qquad (2)$$

where g denotes the average cosine. The volume rendering equation also has its equivalent integral form, represented by

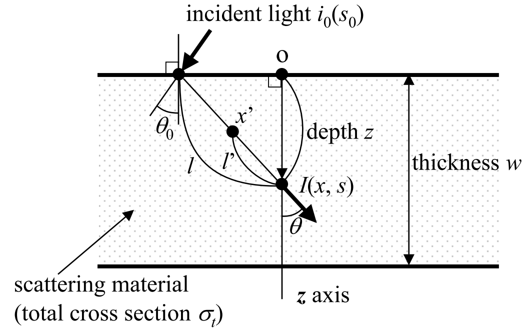


Fig. 1 Infinite slab of scattering material with thickness w.

$$I(x, s) = \int_0^l [\exp(-\sigma_t l')\sigma_s \int p(s, s')I(x', s')ds']dl' + I_0(x, s), \qquad (3)$$

$$x' = x - l's, \qquad (4)$$

where I_0 denotes the attenuated direct light, and l represents the length of the integration path (**Fig. 1**).

The volume rendering equation is simple when the material is homogeneous and the boundary is an infinite plane, as shown in Fig. 1. In this case, $I(x, s)$ depends only on the depth z and the direction s, so the gradient ∇ in Eq. (1) can be replaced by the ordinary differential d/dz, as follows:

$$\cos\theta dI(z, s)/dz = -\sigma_t I(z, s) + \sigma_s \int p(s, s')I(z, s')ds', \qquad (5)$$

where θ represents the angle between s and the z-axis.

The plane-parallel problem can be solved through discretization. Using an orthogonal function system on the unit sphere, $\phi_j(s)$, the irradiance $I(z, s)$ can be approximated by:

$$I(z, s) \simeq \sum_j I_j(z)\phi_j(s). \qquad (6)$$

For simplicity, let us adopt the delta-function $\delta(s - s_j)$ for ϕ_j as in Refs. [1], [14], leading to:

$$\cos\theta_i dI_j(z)/dz = -\sigma_t I_j + \sigma_s \sum_i p(s_i, s_j)I_i(z)ds', \qquad (7)$$

Equation (7) is a linear ordinary differential equation with respect to I_j and can be analytically solved through eigenvalue decomposition.

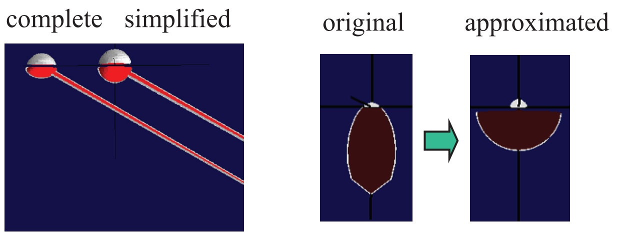
3.2 Simplified Plane-parallel Model

Calculation of the plane-parallel solution involves eigenvalue decomposition, which is still computationally expensive for runtime evaluation. In this section, we further simplify the model to obtain a faster, non-iterative solution.

Figure 2 (a) shows some examples of scattering fields, calculated from the plane-parallel model, where directional light illuminates the media. As seen in the figure, power is concentrated near the incident light direction. Thus, let us apply the following rough approximation to the cosine factor

$$\cos(\theta) \sim \pm\cos(\theta_0). \qquad (8)$$

where θ_0 denotes the incident light angle. We also approximate the phase function $p(s, s')$ by the sum of the uniform functions on



(a) Example of complete and simplified solutions

(b) Uniform approximation of phase function

Fig. 2 Directional distribution of the scattering field computed at depth z. In each image, the left hand side shows complete solutions obtained using the plane-parallel model, and the right hand side shows approximate solutions calculated using the simplified model. $\sigma_s = 0.95$ and $\sigma_t = 1.0$.

the upper and lower hemispheres, as shown in Fig. 2 (b):

$$p(s, s') \sim \begin{cases} TU_+(s') + RU_-(s') & (s \in \Omega_+), \\ RU_+(s') + TU_-(s') & (s \in \Omega_-), \end{cases} \tag{9}$$

where

$$T = \int U_+(s')p(s_0, s')ds', \ R = \int U_-(s')p(s_0, s')ds'$$

and

$$U_+(s') = \begin{cases} 1/2\pi & (s' \in \Omega_+), \\ 0 & (s' \in \Omega_-), \end{cases},$$
$$U_-(s') = \begin{cases} 0 & (s' \in \Omega_+), \\ 1/2\pi & (s' \in \Omega_-), \end{cases} \tag{10}$$

in which s_0 denotes the direction of the incident light.

For the positive direction $s \in \Omega_+$, Eq. (7) can be approximated by

$$\cos\theta_0 dI(z, s)/dz = -\sigma_t I + \sigma_s \left(T \int_{\Omega_+} I(z, s')ds' + R \int_{\Omega_-} Ids' \right), \tag{11}$$

and for the negative direction $s \in \Omega_-$ by,

$$-\cos\theta_0 dI(z, s)/dz = -\sigma_t I + \sigma_s \left(R \int_{\Omega_+} I(z, s')ds' + T \int_{\Omega_-} Ids' \right). \tag{12}$$

As shown in the next section, these equations have very simple analytical solutions. Let us call this model the simplified plane-parallel model and abbreviate it here to SPP model.

3.3 Solutions

As in Fig. 2 (a), the shapes of the solutions apparently consist of upper and lower hemispheres and a delta-function. This suggests that the solution might be constructed from the uniform functions $U_\pm$ and the incident light distribution (a delta function). In fact, it can be confirmed that there are two types of solutions related to the hemispheres and the incident light as follows.

Type I

Based on the observation, we assume the Type I solution to have a hemispheric form as follows:

$$I(z, s) = (\alpha U_+(s) + \beta U_-(s)) \exp(-\lambda z), \tag{13}$$

where, α, β, and λ are constants. Substituting this into Eqs. (11) and (12) yields a set of quadratic equations with respect to α, β, and λ:

$$-\lambda\alpha \cos\theta_0 = -\sigma_t\alpha + \sigma_s(\alpha T + \beta R)$$
$$\lambda\beta \cos\theta_0 = -\sigma_t\beta + \sigma_s(\alpha R + \beta T). \tag{14}$$

Since the solution has scaling freedom, we set $\beta = 1$. From Eq. (14), we have:

$$\alpha(\sigma_t - \sigma_s T - \lambda \cos\theta_0) = \sigma_s R \tag{15}$$
$$\alpha\sigma_s R = (\sigma_t - \sigma_s T + \lambda \cos\theta_0). \tag{16}$$

By dividing Eq. (15) by Eq. (16), we have a quadratic equation for λ:

$$(\sigma_t - \sigma_s T - \lambda \cos\theta_0)/\sigma_s R = \sigma_s R/(\sigma_t - \sigma_s T + \lambda \cos\theta_0),$$
$$(\sigma_t - \sigma_s T)^2 - (\lambda \cos\theta_0)^2 = (\sigma_s R)^2$$

By solving this equation,

$$\lambda = \pm(1/\cos\theta_0)[(\sigma_t - \delta_\sigma)(\sigma_t - \sigma_s)]^{1/2},$$
$$\alpha = \sigma_s R/(\sigma_t - \sigma_s T - \lambda \cos\theta_0),$$
$$\beta = 1,$$
$$\delta_\sigma = \sigma_s(T - R), \tag{17}$$

where we used $R + T = 1$.

With these parameters, Eq. (13) satisfies Eqs. (11) and (12).

Type II

This type of solution is related to the incident light. Using an arbitrary function F_+ that satisfies:

$$F_+(s) = 0 \quad \text{for } s \in \Omega_-,$$
$$\int_\Omega F_+(s)ds = 0,$$

the Type II solution is defined by:

$$I(z, s) = F_+(s) \exp(-\sigma_t z). \tag{18}$$

It can be easily confirmed that Eq. (18) satisfies Eqs. (11) and (12). This solution involves negative values, but these are cancelled in Type I solutions by applying boundary conditions.

3.4 Boundary Conditions

Let us describe the incident light distribution by $i_0(s)$, which is a delta function $\delta(s - s_0)$ in the case of directional light. The boundary conditions can be satisfied by a linear combination of type I and II solutions. We assume the solution $I(z, s)$ to be:

$$I(z, s) = c_1(\alpha U_+(s) + \beta U_-(s)) \exp(-\lambda z) + (c_2 U_+(s) + i_0(s)) \exp(-\sigma_t z) + c_3(\alpha U_+(s) + \beta U_-(s)) \exp(\lambda z), \tag{19}$$

where i_0 represents the incident light distribution. Note that the first and the third terms are Type I solutions. The second term is a Type II solution by setting

$$c_2 = -\int_{\Omega_+} i_0(s')ds'/T. \tag{20}$$

The boundary conditions are:

$$
\begin{aligned}
I(0, s) &= i_0(s) \quad \text{for } s \in \Omega_+,\\
I(w, s) &= 0 \quad \text{for } s \in \Omega_-,
\end{aligned} \tag{21}
$$

where i_0 and w denote the incident light distribution and the thickness of the media, respectively (Fig. 1). These lead to a linear equation with c_1 and c_3, which yields

$$
\begin{aligned}
c_1 &= -\alpha E_3 c_2/(\alpha^2 E_3 - \beta^2 E_1),\\
c_3 &= -\beta E_1 c_2/(\alpha^2 E_3 - \beta^2 E_1),
\end{aligned} \tag{22}
$$

where $E_1 = \exp(-\lambda w)$ and $E_3 = \exp(\lambda w)$.

The evaluation of the scattering field can be summarized as follows:

(1) Calculate λ, α and β according to Eq. (17)
(2) Calculate c_1, c_2 and c_3 according to Eqs. (20) and (22)
(3) Calculate $I(z, s)$ by Eq. (19).

Note that the calculation is simple enough to be evaluated in runtime per vertex or pixel.

3.5 Refinement

The approximation in Eqs. (8) and (9), neglects directional variations of scattering events, and the calculated scattering fields are almost directionless. However, when we can apply ray-marching or the usual surface shading operations to the approximated solutions, they can be improved by taking directional variations into account.

The ray-marching operation is described by the line integral (Eq. (4)). Since the simplified solution, Eq. (19), is a sum of exponential functions, the integration is straight forward. The calculation is shown in Appendix A.1.

3.6 Experiment 1

We examined the manner in which this approximation works. **Table 1** shows the root-mean-square difference, averaged over several incident directions and depth values. As shown in the table, the simplified plane-parallel model generally provides a good approximation, and the ray-marching integral makes further improvements. For lower albedo values, the approximation errors were very low because the direct light components are dominant in these cases, but even in cases when multiple scattering is dominant with higher albedo values, the approximation is reasonable. **Table 2** shows the RMS errors for several thickness values. It can be seen that the errors slightly increase as the thickness decreases, but the influence is not significant.

Table 1 Measured RMS errors due to the simplification. For given albedo values and phase functions, the error in the simplified solution (S.) and that in the refined solution (R.) are described. The errors are averaged over a number of directions and depth values, normalized by the total power. The total error throughout all the albedo values is shown in the last row. The phase function is a Henyey-Greenstein's function (Eq. (2)). The thickness of the media was set to $10/\sigma_t$.

albedo	$g = 0.0$		$g = 0.5$		$g = -0.5$	
	S.	R.	S.	R.	S.	R.
0.5	2.7%	0.3	1.3	0.3	1.2	0.2
0.7	1.9	0.5	2.1	0.7	1.8	0.4
0.9	2.1	1.2	3.4	1.7	2.7	1.3
0.95	2.9	1.7	4.1	2.7	3.4	2.0
0.99	4.5	3.9	5.5	5.8	5.0	4.7
total	2.8	1.7	3.3	2.5	2.9	2.1

Figure 3 shows the solutions to Eq. (11) and the ones refined by the ray-marching integral. As shown in the figure, the SPP solutions are in reasonable agreement with the exact solutions and improvements made by using ray-marching can also be observed.

4. Geometric Compensation

A major limitation of the plane-parallel theory is that it assumes infinite planar boundaries, which is unrealistic in many cases. This section describes a compensation method that takes the boundary geometry into account.

4.1 Second Order Analysis

Let us consider the situation illustrated in **Fig. 4** (a). We observed that both the complete and simplified plane-parallel models tend to over-estimate irradiance at points with a large depth

Table 2 Measured RMS errors with several thickness values (w). The total cross section $\sigma_t = 1.0$. Other conditions are the same as Table 1.

albedo	w = inf	w = 16	w = 10	w = 5
0.5	0.3%	0.3	0.3	0.4
0.7	0.7	0.7	0.7	0.8
0.9	1.7	1.7	1.6	1.6
0.95	2.3	2.2	2.1	2.6
0.99	4.0	4.6	4.6	4.4
total	2.3	2.5	2.4	2.5

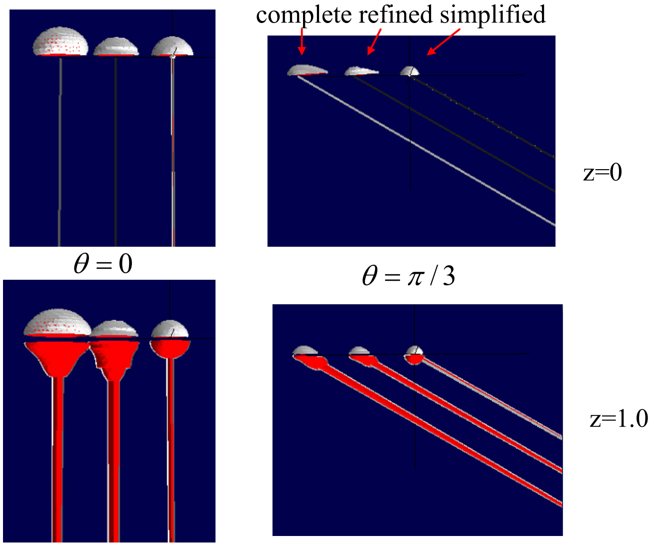
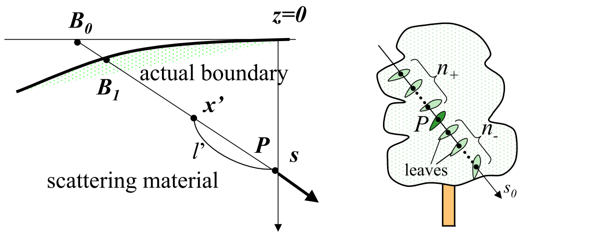


Fig. 3 Directional distribution of scattering field computed at depth z. In each image, the left hand side shows complete solutions obtained by the plane-parallel model, the right hand side shows approximated solutions calculated by the simplified model, and the middle shows refined solutions. $\sigma_s = 0.99$, $g = 0.5$ and $\sigma_t = 1.0$.



(a) G-factor calculation

(b) Counting intersections for point P

Fig. 4 G-factor and depth computation.

value, and often the single scattering model provides a much better approximation in such areas. On the other hand, the plane-parallel models provide much better results near directly illuminated points by capturing multiple reflection light. Therefore, it is an attractive idea to interpolate between two results by estimating the degree to which the plane-parallel condition is satisfied.

In the plane-parallel models, the boundary is assumed to be the plane $z = 0$. At P in Fig. 4 (a), for example, the irradiance in s direction is calculated from the sum of the contributions along P-B_0 by the plane-parallel model. However, the real boundary point is B_1, and the contribution from B_0-B_1 gives rise to errors in the plane-parallel models. These errors can be estimated in the following way. When the boundary point is B_0, the irradiance at point x can be estimated using the integral form (Eq. (3))

$$\begin{aligned} I(P, s; B_0) &= \int_P^{B_0} \exp(-\sigma_t l')\sigma_s \int p(s, s')I'(x', s')ds'dl' \\ &\quad + I_0(P, s), \end{aligned} \tag{23}$$

where $I'(x, s)$ represents the approximate irradiance at x. Therefore, the difference in irradiance can be estimated from:

$$\delta I(P, B_1) = I(P, s; B_0) - I(P, s; B_1) \tag{24}$$

$$= \int_{B_1}^{B_0} \exp(-\sigma_t l')\sigma_s \int p(s, s')I'(x', s')ds'dl'. \tag{25}$$

When $B_1 = B_0$, then $\delta I = 0$ and the plane-parallel models provide good approximations. On the other hand, when $B_1 = P$, δI is maximized and the single scattering model provides a better approximation. Therefore, the value

$$G_s = \delta I(P, B_1)/\delta I(P, P) \tag{26}$$

can measure the validity of the plane-parallel models. When we adopt the direct light I_0 as $I'(x's')$ in the integral, the integrand becomes a simple exponential function, which can be analytically evaluated (see Appendix A.2).

Summing the difference G_{s_i} for N sample directions, s_i, we define the geometric compensation factor (G-factor) G by

$$G(P) = 1.0 - \sum_i^N G_{s_i}/N. \tag{27}$$

When the plane-parallel condition holds, G takes the value 1, and it takes low values when the boundary departs from the plane. Using the G-factor, we interpolate the irradiance value calculated from the SPP model, I_{pp}, and the direct light I_0, using

$$I_G = GI_{pp} + (1 - G)I_0. \tag{28}$$

4.2 Experiment 2

We carried out experiments to estimate the accuracy of the proposed method by comparing the results with those from a Monte Carlo method. The test object was a tree model composed of Lambertian meshes. In the Monte Carlo method, a number of light rays were shot and traced, and the number of intersections were recorded in each mesh.

Depth information is fundamental in the SPP model. Instead of estimating the mean free path, $1/\sigma_t$, and storing the geometric depth, we counted the number of intersections n_+ and n_- along the incident light direction (Fig. 4 (b)), and assigned the depth d and thickness w to

$$d = n_+/\sigma_t, \quad w = (n_+ + n_-)/\sigma_t, \tag{29}$$

where σ_t can be any fixed value because all the calculations with depth d are factorized by σ_t. Each mesh stores a depth and thickness value by averaging n_+ and n_- of the light rays that intersect the mesh.

The phase function can be estimated from the distribution of surface normal vectors and the shading model. In this experiment, we simply set the same value for the Lambertian reflectance and the transmittance. In this case, the phase function becomes an isotropic phase function.

In general, scattering theories can only provide average behavior of light reflection/transmission. To examine how well the SPP model estimates the average scattering field with respect to depth and thickness, we classified the meshes based on the depth and thickness values, and evaluated the RMS error, based on averages within the depth-thickness categories, using

$$\mathrm{RMS}_{dw} = \sqrt{\sum_{d,w}(\langle L^{d,w}\rangle^2 - \langle L_0^{d,w}\rangle^2)/\sum_{d,w}\langle L_0^{d,w}\rangle^2}, \tag{30}$$

where $L_0^{d,w}$ and $L^{d,w}$ represent the irradiance of a mesh with depth d and thickness w, calculated from the MC method and the other models, respectively, and $\langle L\rangle$ indicates the average of L.

We first examined an isolated tree, and calculated the irradiance using the Monte Carlo method (MC), the simplified plane-parallel model (SPP), the SPP model with G-factor compensation (SPPG), and the single scattering model (SS). In the single scattering model, reflection/transmission of the direct light, I_0, was simply calculated. The tree model consisted of 18K triangles, with directional light illuminating it from above. In the Monte Carlo method, 184K rays were traced.

Figure 5 shows the generated images and **Table 3** presents the measured RMS values. As shown in the figure, the SPP model produces a too bright image, especially in the lower part of the canopy, making the RMS value the largest of all

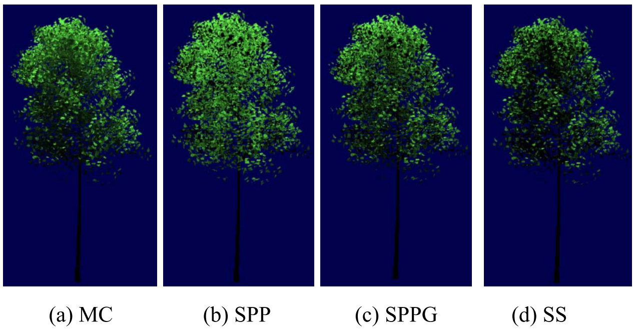

(a) MC (b) SPP (c) SPPG (d) SS

Fig. 5 Comparison of scattering fields for a single tree.

Table 3 Comparisons of measured RMS errors defined by Eq. (30). The albedo value was set to 0.95.

	single tree	multiple trees
SPPG	6.2%	9.2%
SPP	65	43
SS	16	28

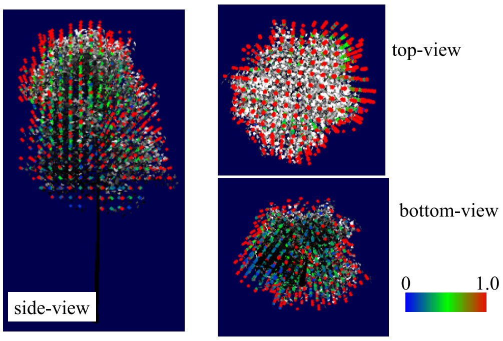
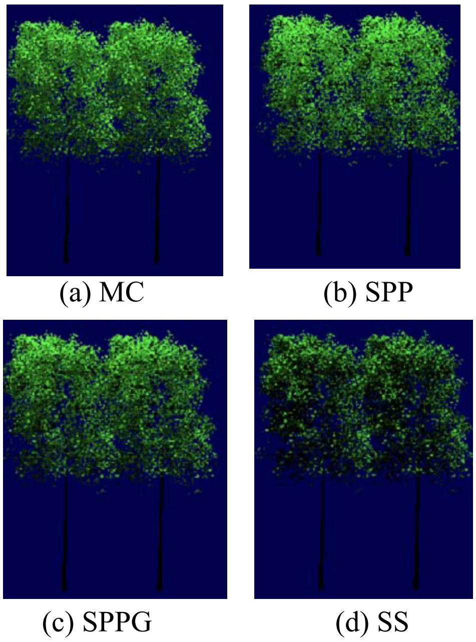


Fig. 6 Evaluated G-factors represented by psuedo colors.


(a) MC (b) SPP

(c) SPPG (d) SS

Fig. 7 Comparison of scattering fields for multiple trees.

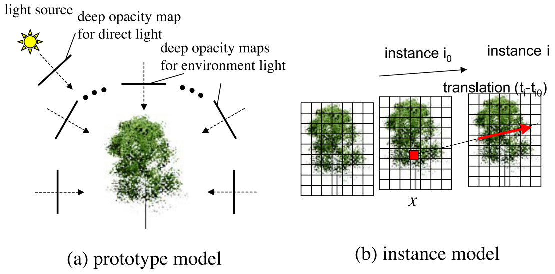
the models. The SPP model with the G-factor well compensates the over-estimation while taking account of multiple reflection/transmission of leaves. **Figure 6** shows the evaluated G-factor at each cell. The red color indicates a high G-factor value while blue represents a low value. The observed RMS was very small. The image from the single scattering model is somewhat dark in the upper part, and the RMS was twice that of the SPPG.

Next, we repeated the same tree model periodically in the scene. The generated images are shown in **Fig. 7**. Due to inter-reflection among the trees, the reference MC image appears brighter. However, the SS image has stayed dark, making its RMS larger. The SPP image is still brighter than the reference, but the G-factor again has adapted well to this situation and captured inter-reflections among the trees.

5. Rendering System

We constructed a system for tree rendering based on the SPP model and the geometric compensation. The view-independent feature of the SPP model allows fast display as shown in this section.

5.1 Data Representation



(a) prototype model (b) instance model

Fig. 8 Deep opacity maps and cells in the tree model.

Vegetation scenes are usually constructed by repeatedly placing the same tree models in different positions. To efficiently deal with such structures, we represent scene data by prototype tree models, instance models, and ground models. A prototype model represents data for an original tree, and an instance model is a copy of the prototype model, placed in the scene.

Prototype model

Prototype models contain all the detailed data, such as geometric data, texture, shading parameters, and depth information. To represent high resolution depth information, we adopted the deep opacity map [8], which efficiently encodes the number of intersections on a GPU. We prepared a deep opacity map for each light source to calculate the scattering field due to each source. In addition, we sampled a small number of directions (typically 16) and set a deep opacity map for each direction (**Fig. 8** (a)).

Instance model

Instance models contain a pointer to a prototype model, a translation vector, links to neighboring instances, and a low resolution ($\sim 16^3$) 3D grid that represents the G-factors and depth information (Fig. 8 (b)). Each grid cell stores the G-factor and the depth values in the sampling directions of the prototype deep opacity maps.

Ground model

Ground models contain geometric data, textures, shading parameters, and shadow maps in the sampling directions.

5.2 Depth and G-factor Calculation

In a prototype model, the depth information is stored in deep opacity maps. Since a scene usually consists of a small number of prototypes, the maps can be generated quickly.

The depth values in instance models are calculated by using the prototype deep opacity maps. Let us consider the situation shown in (Fig. 8 (b)), where the instance model i_0 is surrounded by neighboring instance models i. The depth value, $d(x, s_j)$, at the grid point x in the sampling direction s_j is the sum of the depth values of the instance models and can be calculated using

$$d(x, s_j) = \sum_{i}^{\text{neighbor}} D_{map}(x - (t_i - t_{i_0}), s_j), \tag{31}$$

where D_{map} denotes the depth value obtained from the deep opacity map of the prototype, and t_i represents the translation vector of the instance i. From the depth values $d(x, s_j)$, the G-factor can be calculated using Eqs. (27) and (A.3).

5.3 Shading

The system deals with directional light sources and environmental light. The prototype deep opacity maps are referred to for direct light shading in order to capture higher resolution variations, while depth values stored in the grid are used for environmental light shading.

Direct light

Let the light source direction, the view direction, and the surface normal be s_0, s_e and n. Referring to the prototype deep opacity map in the s_0 direction, we first calculate the depth $d(x, s_0)$ at a surface point x according to Eq. (31). By substituting d into Eq. (19), the irradiance $I(x, s)$ can be obtained. Although it is possible to use $I(x, s_e)$, we integrate BRDF/BTDF B_{tr} over the surface to obtain the refined solution,

$$I_{\text{refine}} = \int_{\Omega} B_{tr}(s_e, s'; n) I(x, s')(n \cdot s') ds' \qquad (32)$$

Since $I(x, s')$ is the sum of the unit hemisphere functions (Eq. (10)) and a delta function, the integral can be easily evaluated by referring to the integral table $S_{\pm}$:

$$S_{\pm}(s_e, n) = \int_{\Omega} B_{tr}(s_e, s'; n) U_{\pm}(s') ds'$$

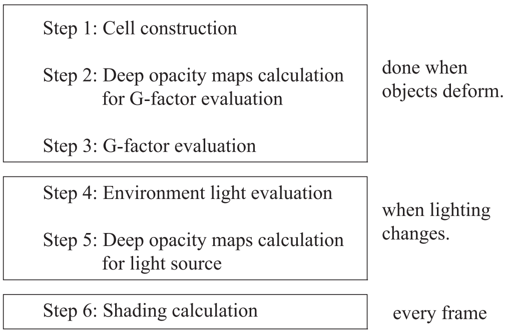
Finally, the G-factor $G(x)$ stored in the corresponding grid cell is multiplied by I_{refine},

$$I_{\text{final}} = G(x) I_{\text{refine}}.$$

5.4 Procedure

The outline of the procedure is listed in **Fig. 9**. First, we construct the cell structure, setting the density and the gradient (Step 1). Second, the deep opacity maps in the sample directions are constructed (Step 2), and the G-factor is evaluated at each cell position (Step 3). For static scenes, these three steps are performed only once.

Next, the environmental light intensity is calculated for each cell (Step 4), and the deep opacity map in the light source directions are constructed (Step 5). These steps are recalculated when lighting conditions change.



Step 1: Cell construction

Step 2: Deep opacity maps calculation for G-factor evaluation

Step 3: G-factor evaluation

done when objects deform.

Step 4: Environment light evaluation

Step 5: Deep opacity maps calculation for light source

when lighting changes.

Step 6: Shading calculation

every frame

Fig. 9 Outline of the procedure.

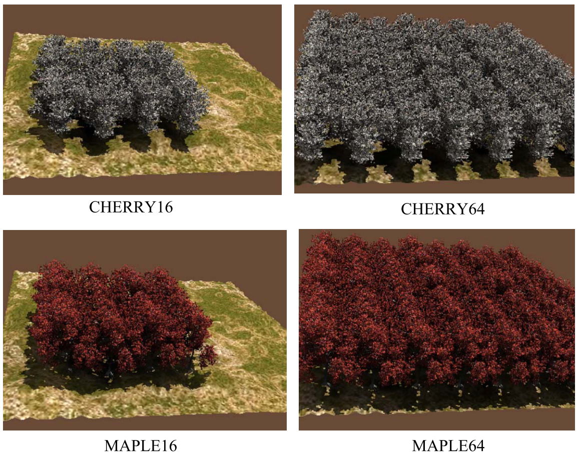
Finally, at each pixel, the scattering field is evaluated according to Eqs. (17), (22), and (19), by referring to the deep opacity map in the light source direction and the normal table of the cells (Step 6). The pixel color is then calculated for the eye direction using the shading information. This color evaluation generally involves an integral over direction s, but in isotropic scattering, the calculation becomes simple, as described in the Appendix A.2. Note that branches and other opaque objects can also be rendered in the same way by simply setting their transparency to zero. Also note that only Step 6 is necessary for the display as long as the objects and the lighting do not change.

5.5 Results

The performance was measured for the scenes shown in **Fig. 10**, and **Table 4** shows the execution time on a PC with an Intel Core2 Quad Q8200 at 2.33 GHz and NVIDIA GeForce GTX 295. The CHERRY16 and the CHERRY64 consist of 16 and 64 instance models of a cherry blossom prototype model, and the MAPLE16 and MAPLE64 consist of 16 and 64 instances of a Japanese maple prototype. As seen in the table, interactive speeds were achieved for dynamic lighting environments.

Figure 11 shows sample images rendered by the proposed model (SPPG) and the single scattering model. For reference purposes, images generated by the MC method and real photographs are also shown. As seen in the figure, the single scattering model has created an unnaturally dark image, while the proposed model has produced a much better image by taking account of multiple transmission/reflections.

Table 5 shows the computation time required to generate the pictures in Fig. 11. As shown in the table, interactive display was



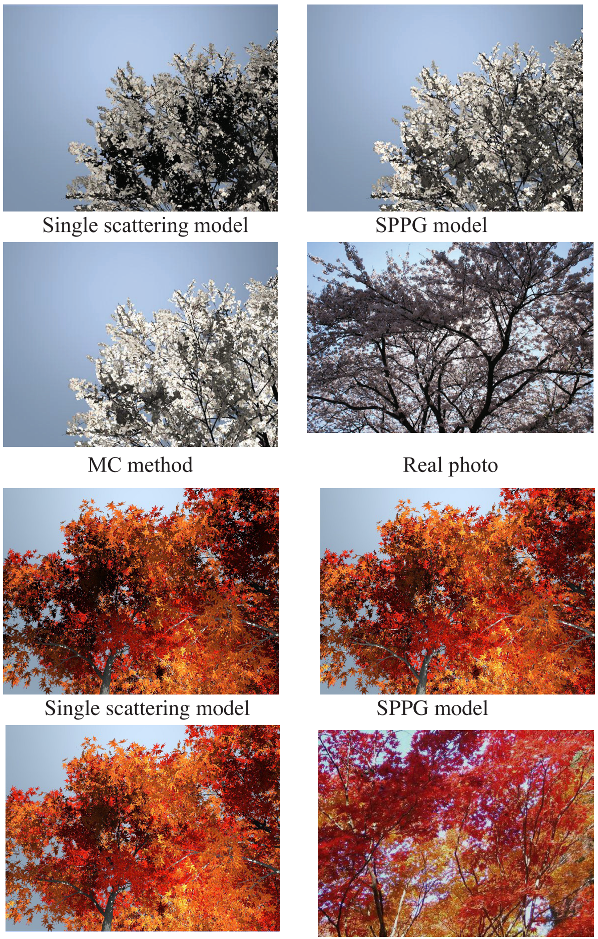
CHERRY16 CHERRY64

MAPLE16 MAPLE64

Fig. 10 Scenes used in the performance evaluation.

Table 4 Timing data. The number of the sampling directions of the deep opacity maps was 22.

	CHERRY16	CHERRY64	MAPLE16	MAPLE64
# of polygons	1.0 M	4.0 M	3.5 M	14 M
# of cells	13 K	52 K	10 K	40 K
Step 2 and 3 (Preprocess)	117 ms	533 ms	95.3 ms	426 ms
Step 4 (Skylight change)	25 ms	100 ms	20 ms	80 ms
Step 5 (Sunlight change)	3.9 ms	3.9 ms	6.7 ms	6.7 ms
Step 6 (display)	27 ms	101 ms	29 ms	131 ms



Single scattering model

SPPG model

MC method

Real photo

Single scattering model

SPPG model

MC method

Real photo

Fig. 11 Rendered images.

Table 5 Timing data for Fig. 11. "Preprocess" includes Step 1 to 5 for SPPG, and Step 4 and 5 for SS. For MC, it includes the ray tracing process that calculates indirect light irradiance.

	CHERRY		MAPLE	
	preprocess	display	preprocess	display
SS	4.8 ms	40.0 ms	6.0 ms	120.0 ms
SPPG	58.4 ms	40.2 ms	50.9 ms	120.3 ms
MC	71×10^3 ms	40.2 ms	433×10^3 ms	120.3 ms

achieved, and the display performance was almost the same for the three methods.

The MC method was implemented on a CPU using a voxel-based ray tracer. We randomly generated light rays (ten times the number of meshes), traced them, and recorded the intersections on a per-mesh basis. To improve the visual resolution, we did not save the first intersections, but calculated the direct light component in the display stage using the deep opacity map, as SS and SPPG do. This is the reason why the display speed of MC was almost the same as SS. Since the difference between the SS and SPPG display processes is just a few arithmetic calculations such as those in Eqs. (17), (22), and (19), the required computation time was almost the same.

6. Conclusion

A simplified plane-parallel scattering model was proposed for rendering densely distributed objects. A simplification was made by setting the cosine factor constant, based on the observation that a large part of the scattering power tends to be localized near to the incident light direction. The simplified plane-parallel model has very simple analytic solutions which reasonably approximate the original exact solutions especially in the case of isotropic scattering. A geometric compensation was also introduced to cope with the infinite plane assumption required by the original plane-parallel model. The model was successfully applied to tree rendering. Comparisons with a Monte Carlo method were made and reasonable agreement was confirmed.

A rendering system based on the model was implemented, and multiple inter-reflection was efficiently achieved. The pre-computation was also modest, allowing interactive control of lighting conditions. Future work includes the application to larger scale scenes, which would be possible by dealing with simpler tree models, such as billboards.

Acknowledgments Part of this work was supported by Research-In-Aid No.21500108 and No.23135528.

References

[1] Max, N., Mobley, C., Keating, B. and Wu, E.-H.: Plane-parallel radiance transport for global illumination in vegetation, *Eurographics Workshop on Rendering '97*, pp.239–250 (1997).

[2] Zinke, A., Yuksel, C., Weber, A. and Keyser, J.: Dual scattering approximation for fast multiple scattering in hair, *ACM Trans. Gr.*, Vol.27, No.3, p.32 (2008).

[3] Jakob, W., Arbree, A., Moon, J.T., Bula, K. and Marschner, S.: A radiative transfer framework for rendering materials with anisotropic structures, *ACM Trans. Gr.*, Vol.29, No.4, p.53 (2010).

[4] Ishimaru, A.: *Wave propagation and scattering in random media*, volume 1, Academic Press, New York (1978).

[5] Shinya, M., Shiraishi, M., Dobashi, Y., Iwasaki, K. and Nishita, T.: A Simplified Plane-Parallel Scattering Model and Its Application to Hair Rendering, *Pacific Graphics 2010*, Poster, pp.85–92 (in IEEE Digital Library) (2010).

[6] Lokovic, T. and Veach, E.: Deep shadow map, *Proc. SIGGRAPH 2000*, pp.385–392 (2000).

[7] Kim, T.-Y. and Neumann, U.: Opacity shadow map, *Proc. 12th Eurographics Workshop on Rendering*, pp.177–182 (2001).

[8] Yuksel, C. and Keyser, J.: Deep opacity map, *Computer Graphics Forum*, Vol.27. No.2, pp.675–680 (2008).

[9] Qin, X., Nakamae, E., Tadamura, K. and Nagai, Y.: Fast photo-realistic rendering of trees in daylight, *Computer Graphics Forum*, Vol.22, No.3, pp.243–252 (2003).

[10] Boulanger, K., Bouatouch, K. and Pattanaik, S.: Rendering trees with indirect lighting in real time, *Computer Graphics Forum*, Vol.27, No.4, pp.1189–1198 (2008).

[11] Moon, J. and Marschner, S.R.: Simulating multiple scattering in hair using a photon mapping approach, *ACM Trans. Gr.*, Vol.25, No.3, pp.746–753 (2006).

[12] Moon, J., Walter, B. and Marschner, S.: Efficient multiple scattering in hair using spherical harmonics, *ACM Trans. Gr.*, Vol.27, No.3, p.31 (2008).

[13] Wang, W., Dorsey, J., Yang, X., Guo, B. and Shum, H.-Y.: Real-time rendering of plant leaves, *ACM Trans. Gr.*, Vol.24, No.3, pp.712–719 (2005).

[14] Stam, J.: An illumination model for a skin layer bounded by rough surfaces, *Proc. 12th Eurographics Workshop on Rendering*, pp.39–52 (2001).

Appendix

A.1 Refinement by Ray-marching

Let us consider the situation shown in Fig. 1. For simplicity, let us define a function E as

$$E(z;\lambda,\sigma) = \begin{cases} z & (\text{if } \lambda = \sigma), \\ 1/(-\sigma+\lambda)\exp((-\sigma+\lambda)z) & (\text{otherwise}), \end{cases} \tag{A.1}$$

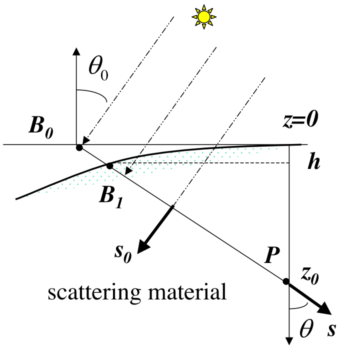


Fig. A·1 G-factor evaluation.

and set values $E_i^{\pm}$ as

$$\begin{aligned}E_i^+ &= (1/\cos\theta)(E(z_0;\gamma_i,\sigma/\cos\theta) - E(0;\gamma_i,\sigma/\cos\theta))\\ E_i^- &= (1/\cos\theta)(E(w;\gamma_i,\sigma/\cos\theta) - E(z_0;\gamma_i,\sigma/\cos\theta))\end{aligned}$$

for $\gamma_1 = -\lambda$, $\gamma_2 = \sigma$, and $\gamma_3 = \lambda$. We also define the function $S_\pm$ by

$$S_\pm(s) = \int_\Omega p(s,s')U_\pm(s')ds'.$$

Substituting Eq. (19) into Eq. (4), the refined solution can be expressed by

$$\begin{aligned}I^\pm_{\text{refine}}(z,s) &= S_+(s)(c_1\alpha_1E_1^\pm + c_2E_2^\pm + c_3\alpha_2E_3^{pm})\\ &\quad +S_-(s)(c_1\beta_1E_1^\pm + c_3\beta_2E_3^\pm)\\ &\quad +p(s,s_0)E_2^\pm, \qquad\qquad (A.2)\end{aligned}$$

where $I^\pm_{\text{refine}}(z,s)$ denotes the refined solution for $s \in \Omega_\pm$. By preparing an integration table for $S_\pm$, Eq. (A.2) can be efficiently evaluated.

A.2 G-factor Evaluation

The direct light component can be described by

$$I'(z,s) = \delta(s-s_0)\exp(-\sigma_t z),$$

where s_0 denotes the incident light direction (**Fig. A·1**). By substituting this into Eq. (25), we have:

$$\delta I(P,B_1) = \exp(-\sigma_t z_0/\cos\theta)(1.0 - E(h;\sigma_t/\cos\theta_0,\sigma_t/\cos\theta)),$$

and

$$\begin{aligned}G_{s_i} &= (1.0 - E(h;\sigma_t/\cos\theta_0,\sigma_t/\cos\theta))\\ &\quad (1.0 - E(z_0;\sigma_t/\cos\theta_0,\sigma_t/\cos\theta)), \qquad\qquad (A.3)\end{aligned}$$

where the function E is defined by Eq. (A.1).

Mikio Shinya is currently a professor in the Department of Information Science, Toho University. He received a B.Sc. in 1979, a M.S. in 1981, and a Ph.D. in 1990 from Waseda University. He joined NTT Laboratories in 1981, and moved to Toho University in 2001. He was a visiting scientist at the University of Toronto in 1988–1989. His research interests include computer graphics and visual science.

Yoshinori Dobashi received his B.E., M.E., and Ph.D. degrees in engineering in 1992, 1994, and 1997, respectively, from Hiroshima University. He worked at Hiroshima City University from 1997 to 2000 as a research associate. He is currently an associated professor at Hokkaido University in the graduate school of engineering. His research interests are computer graphics including lighting models.

Kei Iwasaki received his B.S., M.S., and Ph.D. degrees from the University of Tokyo in 1999, 2001, 2004, respectively. He is presently a research associate at the Faculty of Systems Engineering, Wakayama University. His research interests are mainly in computer graphics.

Michio Shiraishi was born in 1974. He received his Ph.D. degree from the University of Tokyo in 2003. He has been a lecturer at Toho University since 2005. His current research interests are computer graphics and web application systems.

Tomoyuki Nishita received his B.E., M.E., and Ph.D. degrees in electrical engineering from Hiroshima University, Japan, in 1971, 1973, and 1985, respectively. He worked for Mazda Motor Corp. from 1973 to 1979. He has been a lecturer at Fukuyama University in 1979, and then became an associate professor in 1984, and later a professor in 1990. He moved to the Department of Information Science of the University of Tokyo as a professor in 1998 and since 1999 has been a professor in the Department of Complexity Science and Engineering at the University of Tokyo. His research interests are mainly in computer graphics.