A Real-Time Analytic Lighting Model for Anisotropic Scattering

Roger Hong[1] Tan-Chi Ho[1] Jung-Hong Chuang[1] R-Ming Shiu[2] Richard Kuo[2]

[1]Department of Computer Science, National Chiao Tung University
[2]Silicon Integrated Systems Corp.

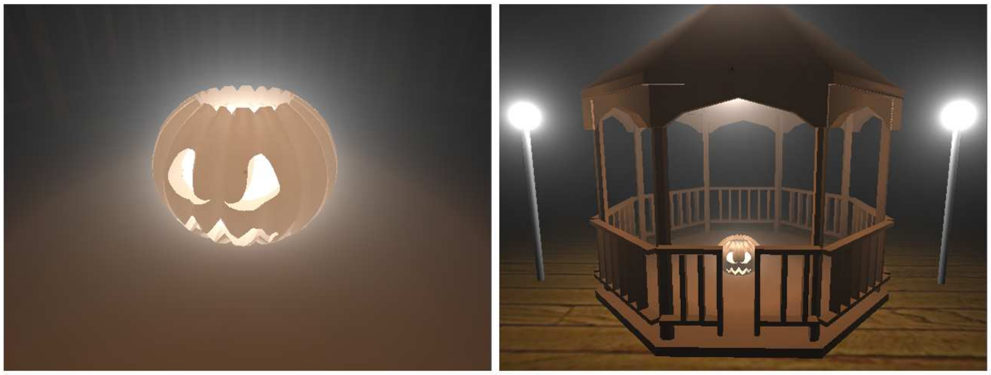


Figure 1: *The anisotropic scattering conditions of an environment visually affects both surface shading and glows around light sources.*

Abstract

Rendering the effects of light scattering for interactive applications is often made difficult by the demands of dynamic viewing, lighting, scene geometry, and environment parameters. Recent advances in rendering of participating media using analytic techniques have enhanced performance significantly to meet this challenge, albeit under assumptions and simplifications made to scenes. In this paper, we present an analytic lighting model for participating media with dynamic anisotropic scattering that may be used for scenes that include fog, mist, or haze. Our model applies an explicit analytic integration to an angle-formulated single scattering integral for a point light source in a homogenous participating medium. Users may create a dynamic scattering environment, such that the scattering conditions are easily manipulated using simple parameters of density and forward scattering level. Using our model, we demonstrate the effect of different levels of forward scattering on glows around light sources and surface radiance. Our method can be implemented on programmable graphics hardware in real-time.

1 Introduction

Rendering effects of light scattering in atmospheric environments has long been a field of interest within the computer graphics community. Light scattering, caused by particles such as water or dust in the air, creates visual effects including visibility loss, glows around light sources, blurred surface radiance, shafts of light, and volumetric shadows. All these visual cues add to the realism of scenes with participating media. The computation of light transport in volumetric environments, however, presents a challenge for developing practical, interactive models for computer graphics.

This paper develops a lighting model for atmospheric environments that allows users to create and adjust the scattering conditions interactively using simple parameters. The density and scattering distribution of particles affects the visual appearance of scenes with scattering media in both obvious and subtle ways. Accurately modeling anisotropic scattering in atmospheric environments requires use of the *phase function*, which describes the directional distribution of scattered light. For most atmospheric conditions, a phase function $p(\theta)$ symmetric about the incident direction is sufficient to describe the scattering distribution [Cerezo et al. 2005], such that the single parameter θ is the angle between the incident direction and exitant direction. The Henyey-Greenstein phase function [Ishimaru 1997] is widely used in computer graphics for its relative simplicity and effectiveness for common atmospheric environments. Its normalized form is

$$p(\theta) = \frac{1-g^2}{4\pi(1+g^2-2g\cos\theta)^{3/2}}, \qquad (1)$$

where the asymmetry parameter $g \in (-1,1)$ controls the amount of forward or back scattering. Values $g > 0$ correspond to forward scattering, $g = 0$ corresponds to isotropic scattering, and $g < 0$ correspond to backward scattering. Positive values for g can describe most weather conditions [Narasimhan and Nayar 2003].

In this paper, we present an analytic lighting model with dynamic anisotropic scattering that may be implemented on programmable graphics hardware in real-time. It focuses on the visual effects present in local scenes with fog, mist, and haze, such that the viewer and light sources are both immersed within the medium. Specifically, our model considers effects of scattered radiance emanating directly from light sources, commonly termed airlight. The model captures effects for variable levels of forward scattering on glows around light sources, diffusing of specular highlights, and shadowed regions. The scattering behavior of the model is controlled by just two parameters, the forward scattering parameter g and the extinction coefficient κ_t of the scattering medium. Both these parameters can be modified in real-time. A precomputation step that is entirely independent of the physical parameters of the scene enables the dynamic behavior. Our model can be easily integrated into an existing real-time rendering framework. To achieve interactive rates, our physics-based lighting model is restricted to single scattering, the assumption that light scatters only once in the medium. In environments with low albedo, single scattering is a reasonable approximation [Cerezo et al. 2005]. The demands typical in interactive applications such as dynamic viewing, lighting, scene geometry, and environment parameters require our model to assume isotropic point light sources, homogeneous media, and single scattering.

Our work can largely be viewed as an application of the methodology used in [Sun et al. 2005] towards an alternative angle-formulated version of the single scattering light equations, whereby we obtain new results that enables efficient and accurate anisotropic scattering. The dynamic scattering capability is presented as an extension to an initial analytic solution. The intention of this work is to consider more complex interaction and visual effects for the interactive rendering of participating media.

2 Related Work

A significant amount of work using Monte Carlo ray tracing methods has been done towards rendering volumetric scattering effects in complex scenes with multiple scattering and non-homogeneous media. These methods, however, are far from interactive. On the other hand, many interactive methods display scattering effects by solving the radiance transfer equation [Cerezo et al. 2005] using numerical techniques. Recently, a number of works have explored the use of analytic techniques for rendering participating media. A thorough survey of works for rendering the effects of light scattering can be found in [Cerezo et al. 2005]. We focus our discussion on recent advances in rendering participating media towards interactive rates.

Numerical Methods. Based on the two-pass method of [Dobashi et al. 2000a], a complete system for real-time cloud rendering [Harris and Lastra] used a particle system to model clouds and precomputed an isotropic multiple forward scattering term in impostors during an initial illumination pass. The precomputation required, however, is dependent upon the scene illumination and scattering properties. Illuminated shafts of light were rendered interactively using *virtual planes* [Dobashi et al. 2000b], storing incident illumination on rectangular meshes lined up perpendicular to the view ray at regular intervals. By investigating the effects of multiple-scattering, [Premoze et al. 2004] developed a *point spread function* (PSF) that measures the spatial spreading of incident radiance. Based upon this multiple scattering term, [Hegeman et al. 2005] developed a lighting model for an inhomogeneous medium with multiple scattering effects at interactive rates, but illumination and medium parameters must be fixed. Other restrictions include the absence of volumetric shadows for objects within the medium.

Analytic Methods. Analytic methods significantly improve the performance for rendering of scenes with participating media. Their major drawback lies in the assumptions and simplifications made for the sake of performance. The work of [Hoffman and Preetham 2003] can render skies using programmable graphics hardware. A simple, analytic form for scattering from a directional light source is made possible when a constant density atmosphere is assumed. [Narasimhan and Nayar 2003] modeled multiple scattering from an isotropic point light source using an analytic expression. The model can be used for real-time rendering of glows in different atmospheric conditions; however, it does not include effects of surface shading, since it assumes distant light sources. [Venceslas et al. 2006] developed an analytic expression for single scattered radiance for a point light source using a polynomial approximation that can be implemented on graphics hardware. Unfortunately, inaccurate glows appear when the viewer is near the source. An algorithm for incorporating volumetric shadows is introduced that uses the shadow volume algorithm and sorts shadow polygons according to their estimated depths and renders them. Limitations occur when the estimated depths of large polygons have errors. Under the assumption of homogenous media, [Sun et al. 2005] developed a real-time analytic single scattering model for point light sources that can render visual effects on surface shading and handle complex environment map lighting using programmable graphics hardware. Significant effects such as glows around light sources, spreading of specular highlight on objects, and brightening of shadows were realized in real-time fog scenes. The effects of anisotropic scattering, though considered in the model, were not demonstrated. The inclusion of anisotropic scattering in the model decreases performance and accuracy.

3 Analytic Lighting Model

In this section, we develop an analytic model for single scattered radiance reaching the viewer for a point light source in a homogenous medium. The generality of our model is highlighted by the fact that any phase function of a single parameter θ may be used to consider any type of anisotropic scattering, with negligible affect to run-time performance and accuracy. Also, an extension to the model enabling dynamic anisotropic scattering is presented that is easily configurable by a single forward scattering parameter g. A description of the qualitative visual effects for different levels of forward scattering concludes the section.

3.1 Overview

We consider the paths of light transport depicted in Figure 2. Light rays may (a) directly transmit through the medium, leading to attenuation; (b) scatter once from source to viewer, leading to glows around sources [Narasimhan and Nayar 2003]; or (c) scatter once from source to surface point, leading to diffusing of reflected surface radiance and brightening of shadows [Sun et al. 2005]. The aforementioned paths and their visual effects can be captured by the sum of reflected surface radiance L_p with attenuation and airlight from source to viewer L_a [Sun et al. 2005]:

$$L = e^{-\kappa_t d_{vp}} L_p + L_a, \tag{2}$$

where κ_t is the extinction coefficient of the participating medium and d_{vp} is the distance between the viewpoint and surface point. In subsequent sections, we develop analytic models to compute L_p and L_a.

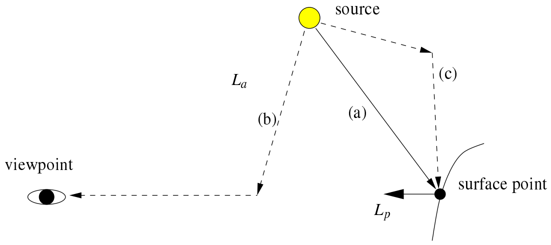


Figure 2: *Diagram illustrating light paths of (a) direct transmission, (b) airlight from source to viewer, and (c) airlight from source to surface point. Path (b) contributes to airlight from source to viewer L_a. Paths (a) and (c) contribute to reflected surface radiance L_p.*

3.2 Angle-Formulated Airlight Model

The contribution of airlight depicted in Figure 2b is given by the classic integral along the viewing direction for single-scattered airlight from an isotropic point light source [Nishita et al. 1987], which is expressed in terms of the distance along the ray. To reduce the number of physical parameters in the integral, we introduce the

Table 1: *Notation and formulae for angle-formulated airlight integral.*

θ	angle between source emissive direction and perpendicular bisector to the viewing direction
γ	angle between view ray and source
d_{sv}	distance between source and viewpoint
d_{vp}	distance between viewpoint and surface point
d_{sp}	distance between source and surface point
t	$t = d_{sv}\cos\gamma$
h	$h = d_{sv}\sin\gamma$
θ_0	$\theta_0 = \gamma - \frac{\pi}{2}$
θ_d	$\theta_d = \arctan\lvert(d_{vp} - t)/h\rvert$
I_0	point light source intensity

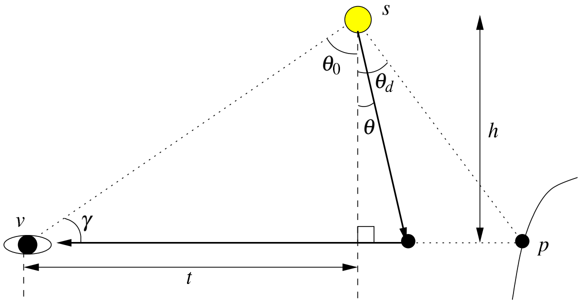


Figure 3: *Diagram for angular reformulation of the airlight integral.*

angle reformulation of the single scattering integral for airlight from a point light source [Lecocq et al. 2000]:

$$L_a(\gamma, d_{sv}, d_{vp}, \kappa_t) = \frac{\kappa_t e^{-\kappa_t t}}{h}\int_{\theta_0}^{\theta_d} e^{-\kappa_t h(\frac{\sin\theta+1}{\cos\theta})} I_0 p(\theta + \frac{\pi}{2}) d\theta. \quad (3)$$

The integral is expressed in terms of the angle θ between the light source emissive direction and the perpendicular bisector to the viewing direction with respect to s, as depicted in Figure 3. θ_d and θ_0 are the bounds of integration. See [Venceslas et al. 2006] for a derivation of the angle reformulation. Refer to Table 1 for the rest of the notation. We refer to Equation 3 in the remainder of the paper as the *angle-formulated airlight integral*.

Analytic solution to L_a

To solve Equation 3, it is factorized into an analytic expression dependent on the physical parameters of the scene and a tabulated function independent of the physical parameters. Let $C_0(\gamma, d_{sv}, \kappa_t) = \kappa_t I_0 e^{-\kappa_t t}/h$ and $C_1(\gamma, d_{sv}, \kappa_t) = \kappa_t h$. Then

$$L_a = C_0 \int_{\theta_0}^{\theta_d} e^{-C_1(\frac{\sin\theta+1}{\cos\theta})} I_0 p(\theta + \frac{\pi}{2}) d\theta.$$

The angle-formulated airlight integral now is factorized into an analytic expression C_0 and an integral that may be expressed as a function of two variables:

$$H(u,v) = \int_{-\frac{\pi}{2}}^{v} e^{-u(\frac{\sin\theta+1}{\cos\theta})} p(\theta + \frac{\pi}{2}) d\theta. \quad (4)$$

As a function independent of the physical parameters of the scene, $H(u,v)$ can be numerically integrated during the precomputation stage into a table and stored into texture memory. The plot of $H(u,v)$ in the isotropic scattering case $p(\theta) = \frac{1}{4\pi}$ is shown in Figure 4, demonstrating that the function is smooth. It is worth mentioning that any phase function $p(\theta)$ may be used in our model. The tabulated function $H(u,v)$ in texture memory can be accessed during run-time using linear interpolation for evaluation of the integral. Thus, an analytic solution to Equation 3 is

$$L_a = C_0[H(C_1, \theta_d) - H(C_1, \theta_0)]. \quad (5)$$

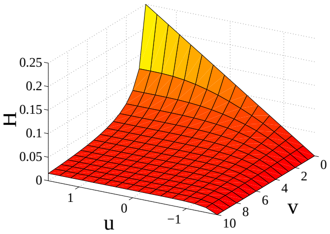


Figure 4: *3D plot of function $H(u,v)$ for $u \in [0,10]$ and $v \in [-\pi/2, \pi/2]$ for $p(\theta) = 1/(4\pi)$. The graph demonstrates that the function is smooth, thus making it appropriate for tabulation during precomputation and look-up during run-time.*

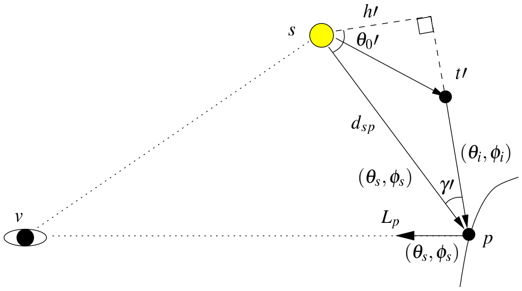


Figure 5: *Diagram for light paths of direct transmission and airlight to surface point that contribute to surface radiance L_p.*

Airlight from a point source may be computed with just evaluation of analytic functions and two texture lookups in the vertex or fragment shader using programmable graphics hardware. The choice of $p(\theta)$ does not affect affecting run-time performance since $H(u,v)$ is constructed in a precomputation stage. The special case of no objects along the view ray ($d_{sv} = \infty, \theta_d = \frac{\pi}{2}$) is important for the next subsection. In such a case, the airlight is

$$L_a(\gamma, d_{sp}, \infty, \kappa_t) = C_0[H(C_1, \frac{\pi}{2}) - H(C_1, \theta_0)]. \quad (6)$$

3.3 Surface Radiance Model using Angle Formulation

Reflected radiance L_p is the sum of contribution from direct transmission $L_{p,d}$ (Figure 2a) and contribution from airlight $L_{p,a}$ (Figure 2c), which is given by the *surface radiance single scattering integral* [Sun et al. 2005]. This is written

$$L_p = L_{p,d} + L_{p,a} \quad (7)$$

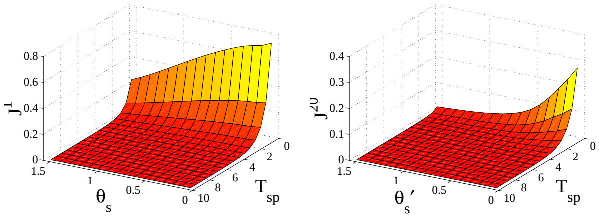
$$L_{p,d} = \frac{I_0 e^{-\kappa_t d_{sp}}}{{d_{sp}}^2} brdf(\theta_s, \phi_s, \theta_v, \phi_v)\cos\theta_s \quad (8)$$

$$L_{p,a} = \int_{\Omega_{2\pi}} L_a(\gamma', d_{sp}, \infty, \kappa_t) brdf(\theta_i, \phi_i, \theta_v, \phi_v)\cos\theta_i d\omega_i. \quad (9)$$

The surface radiance single scattering integral sums airlight contribution to surface radiance over the surface point's hemisphere of incident directions. The angle γ' is the angle between the source direction (θ_s, ϕ_s) and incident direction (θ_i, ϕ_i), such that $h' = d_{sp}\sin\gamma'$ and $t' = d_{sp}\cos\gamma'$. Refer to Figure 5 for a depiction of the these variables. To simplify the integral for evaluation, the airlight term L_a assumes no object obstructs each incident direction.

Analytic solution to L_p

We solve L_p for both the Lambertian BRDF and Phong BRDF separately. For the Lambertian case, $brdf_{Lam} = k_d$. For the Phong case, all angles of Equation 7 are reparameterized



(a) (b)

Figure 6: *3D plots of functions (a) $J^1(T_{sp}, \theta_s)$ and (b) $J^{20}(T_{sp}, \theta_s\prime)$ for $T_{sp} \in [0,10]$ and $\theta_s \in [0, \pi/2]$. The graphs demonstrate that the functions are smooth, thus making them appropriate for tabulation and look-up during run-time.*

against the reflection of the view ray about the surface normal [Ramamoorthi and Hanrahan 2002] to reduce the complexity of the integral. We denote the reparameterizated source direction $\theta_s\prime$. After reparameterization, $brdf_{Phong} = k_s \cos^n \theta_i\prime$. The resultant solution for $L_{p,d}$ is straightforward:

$$L_{p,d} = \frac{I_0 e^{-\kappa_t d_{sp}}}{{d_{sp}}^2}(k_d \cos\theta_s + k_s \cos^n \theta_s\prime). \tag{10}$$

To solve $L_{p,a}$, we make the substitution $T_{sp} = \kappa_t d_{sp}$ and use our analytic model for airlight L_a, Equation 6. Other substitutions and simplifications are made, allowing us to re-express the integral portion as a function of two parameters. The details of this derivation for the solution to $L_{p,a}$ are listed in Appendix A. The resultant solution for $L_{p,a}$ is

$$L_{p,a} = \frac{I_0 \kappa_t}{d_{sp}}[k_d J^1(T_{sp}, \theta_s) + k_s J^n(T_{sp}, \theta_s\prime)], \tag{11}$$

where

$$J^n(T_{sp}, \theta_s\prime) =$$
$$\int_{\Omega_{2\pi}} \frac{e^{-T_{sp}\cos\gamma\prime}}{\sin\gamma\prime}[H(T_{sp}\sin\gamma\prime, \frac{\pi}{2}) - H(T_{sp}\sin\gamma\prime, \gamma\prime - \frac{\pi}{2})]\cos^n \theta_i d\omega_i. \tag{12}$$

Both $J^1(T_{sp}, \theta_s)$ and $J^n(T_{sp}, \theta_s\prime)$ can be numerically integrated, tabulated, and stored into texture memory during a precomputation stage, where n denotes the Phong exponent. Consequently, each of these functions can be evaluated with a texture lookup using linear interpolation during run-time. Plots of J^1 and J^n ($n = 20$) are shown in Figure 6, demonstrating that they are smooth.

The choice of phase function $p(\theta)$ greatly affects the visual appearance of participating media, since it indicates different scattering conditions. The lighting model discussed up to this point requires the phase function $p(\theta)$ to be predetermined, as is common in many scattering models. Without instant visual feedback, it is often cumbersome to manipulate $p(\theta)$ to achieve desirable visual results. In the next section, we develop an extension to our lighting model, allowing the user to change the level of forward scattering in real-time.

3.4 Dynamic Anisotropic Scattering

In this section, we provide an extension to the lighting model for dynamic forward scattering. The form of the Henyey-Greenstein phase function in Equation 1 hinders interactive modification of scattering parameter g. We use a well known fact in the literature on light transport that phase functions can be written as a series of Legendre polynomials [Dunn 1997]:

$$p(\cos\theta) = \frac{1}{4\pi}\sum_{k=0}^{N}(2k+1)b_k P_k(\cos\theta), \tag{13}$$

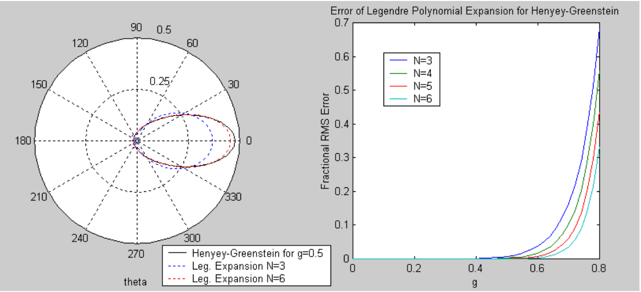


Figure 7: *(left) Plot of the normalized Henyey-Greenstein phase function in solid line and its Legendre polynomial expansion (Equation 13) for expansion terms $N = 3, 6$ in dotted line. Larger N increases the accuracy of the expansion approximation. (right) Plot of the fractional rms error of airlight using the Legendre polynomial expansion for $N = 3, 4, 5, 6$ with varying forward scattering parameter g. Note the loss in accuracy of the expansion approximation for increasing g and smaller N.*

where $P_k(\cos\theta)$ is the Legendre polynomial of order k. The first four Legendre polynomials are listed in Appendix B. In the case of the Henyey-Greenstein phase function, the expansion coefficients are $b_k = g^k$.

Using this alternative expression for the Henyey-Greenstein phase function, we can develop analytic solutions to airlight L_a (Equation 3) and surface radiance L_p (Equation 7) with dynamic scattering by using the Legendre expansion, Equation 13, for $p(\theta)$ and moving forward scattering parameter g outside of any precomputed integrals. This derivation can be found in Appendix C. An analytic solution to Equation 3 for airlight with dynamic anisotropic scattering then is

$$L_a = C_0 \sum_{k=0}^{N}(2k+1)g^k[H_k(C_1, \theta_d) - H_k(C_1, \theta_0)], \tag{14}$$

where

$$H_k(u, v) = \int_{-\frac{\pi}{2}}^{v} e^{-u(\frac{\sin\theta+1}{\cos\theta})} P_k[\cos(\theta + \frac{\pi}{2})]d\theta. \tag{15}$$

The functions H_k for $k = \{0, 1, ..., N\}$ are precomputed and tabulated in texture memory. An analytic solution to $L_{p,a}$ (Equation 7) for single scattering contribution to reflected surface radiance with dynamic anisotropic scattering is

$$L_{p,a} = \frac{I_0 \kappa_t}{d_{sp}} \sum_{k=0}^{N}(2k+1)g^k[k_d J_k^1(T_{sp}, \theta_s) + k_s J_k^n(T_{sp}, \theta_s\prime)], \tag{16}$$

where

$$J_k^n(T_{sp}, \theta_s\prime) =$$
$$\int_{\Omega_{2\pi}} \frac{e^{-T_{sp}\cos\gamma\prime}}{\sin\gamma\prime}[H_k(T_{sp}\sin\gamma\prime, \frac{\pi}{2}) - H_k(T_{sp}\sin\gamma\prime, \gamma\prime - \frac{\pi}{2})]\cos^n \theta_i d\omega_i. \tag{17}$$

The functions J_k^n for $k = \{0, 1, ..., N\}$ are precomputd and tabulated in texture memory. Under this new model with dynamic anisotropic scattering, asymmetry parameter g may be modified interactively to vary the amount of forward scattering.

Implementation of Equations 14 and 16 requires choosing a finite number of expansion terms N to approximate the Henyey-Greenstein phase function. Larger values for N naturally produce more accurate results, as shown by the plot in Figure 7(left). The approximation suffers increasingly for values g closer to 1, as demonstrated by Figure 7(right). In practice, we use $N = 3$ for efficiency purposes, and visually, it produces sufficient results.

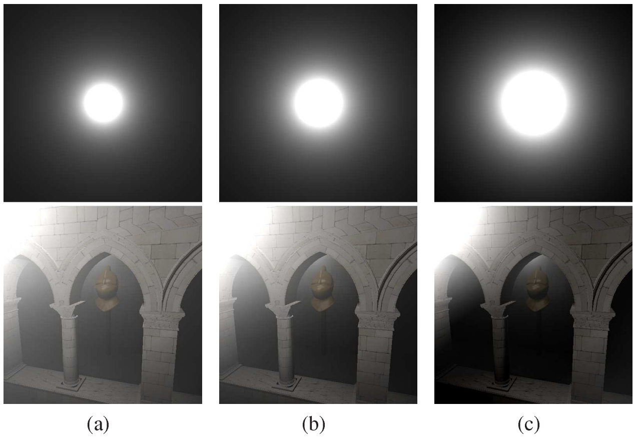


Figure 8: *Captures images of glows around point light sources in (top row)direct view and (bottom row) out of direct view for forward scattering parameter (a) $g = 0$, (b) $g = 0.5$, and (c) $g = 0.9$. When increasing forward scattering parameter g, glows in direct view increase in size, whereas glows out of direct view decrease in size.*

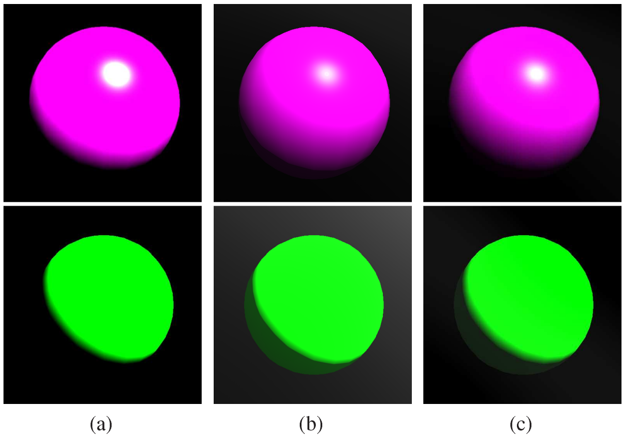


Figure 9: *Images of rendered sphere using only specular radiance in top row and only diffuse radiance in bottom row where scattering conditions are set to (a) clear day, (b) isotropic scattering $g = 0$, and (c) strong forward scattering $g = 0.9$. Note that the diffusing of highlights exhibited in isotropic scattering (top row b) lessens under strong forward scattering conditions (top row c). Also note that the brightening of shadowed regions exhibited in isotropic scattering conditions (bottom row b) lessens under strong forward scattering conditions (bottom row c).*

A key step for efficient implementation is that functions H_k for four values of k can be numerically precomputed and tabulated into the four channels of a single floating point texture. The same is true for the set of functions J_k^1 and J_k^n . Consequently, only one texture lookup is necessary to compute four function evaluations of H_k, J_k^1, or J_k^n, since the values indexed are identical across all values of k. For $N = 3$, Equations 14 and 16 each only require evaluation of analytic functions and two texture lookups. These costs require the same number of texture lookups as the original lighting model(Equations 5 and 11); thus, adding dynamic anisotropic scattering to the model does not significantly affect performance. A final detail of note is that each term $(2k+1)g^k$ in the series is computed on the CPU and passed as a variable into the vertex or fragment shader to lessen the load on the GPU.

3.5 Visual Effects of Anisotropic Scattering

Our experiments have shown that the size of glows around light sources increases for smaller γ and decreases for larger extinction coefficient κ_t, distance from source to viewer d_{sv}, and forward scattering parameter g. For common rendering scenarios, this effectively means glows around light sources in direct view(small γ) increase with larger g. When light sources are outside of direct view, however, visibility increases and glows decrease considerably with larger g, due to the decrease in backscattering. Using Figure 8, we illustrate the visual effect of g on glows around light sources in and out of direct view by rendering with varying g $(g = 0, 0.5, 0.9)$ while keeping other parameters constant.

As discussed in subsection 3.1, scattering leads to diffusing of specular highlights and brightening of shadows. We illustrate the effect of forward scattering parameter g on surface shading by rendering a sphere with only specular radiance in the top row and only diffuse radiance in the bottom row of Figure 9. When compared to the isotropic scattering case $g = 0$, increasing the amount of forward scattering lessens the diffusing of specular highlights(Figure 9c, top row) and lessens the brightening of shadows (Figure 9c, bottom row). In our rendering experiments, we have found that the effect of anisotropic scattering on diffuse radiance, namely the dark regions, is rather common. In contrast, the effect of anisotropic scattering on specular radiance can only be seen in rare cases.

4 Hardware Implementation and Results

Our lighting model can be implemented by evaluating Equation 2 in either the vertex or fragment shader using programmable graphics hardware. For static anisotropic scattering, resolution of Equation 2 requires implementation of Equation 5 for L_a as well as Equations 10 and 11 for L_p. For dynamic anisotropic scattering, resolution of Equation 2 requires implementation of Equation 14 for L_a as well as Equations 10 and 16 for L_p. Non-optimized pseudocode of a fragment shader version with dynamic anisotropic scattering is shown in Figure 10 for $N = 3$ Legendre expansion terms. The entire computation only requires evaluation of some analytic expressions and 4 texture lookups. Extinction coefficient κ_t and asymmetry parameter g can be interactively modified by the user to generate different scattering conditions.

Using an NVidia GeForce 6800 graphics card, we implemented our lighting model with dynamic anisotropic scattering in the fragment shader using GLSL for $N = 3$. In our implementation, functions H_k are precomputed offline and tabulated into a single 512 x 512 floating point texture. Each set of functions J_k^1 and J_k^n are precomputed offline and tabulated into a single 64 x 64 floating point texture.

The performance of our lighting model depends on a number of factors and is linear in the number of light sources. We demonstrate our lighting model using varying levels of anisotropic scattering and extinction coefficients on various scenes with several light sources. The images in Figure 11 were captured at upward of 15 fps at a resolution of 640 x 480. This pavilion and pumpkin scene consists of 16229 vertices, 32152 triangles, and 4 light sources. Rendered images of an atrium scene are shown in Figure 12. This scene consists of 43,479 vertices and 83530 triangles. Images of this scene with 2 light sources were captured at upward of 20fps. With 4 light sources, the frame rate became 10-12 fps.

5 Conclusion

We have presented a simple, efficient, and easy-to-integrate lighting model for participating media that is effective for interactive applications. Scattering parameters such as the phase function and the extinction efficient may be changed in real-time, enabling the user a high level of interactivity in complex scenes with participating

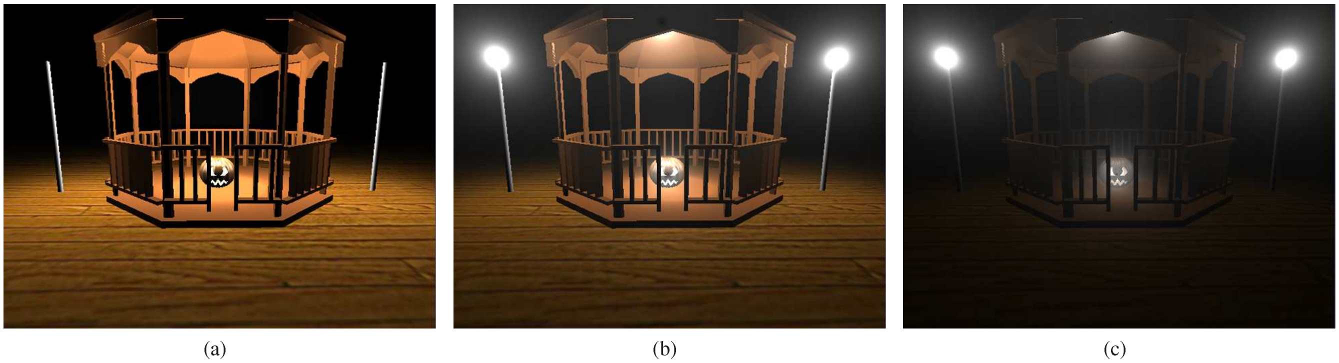

(a) (b) (c)

Figure 11: *Captured images using different densities: (a) clear day, (b) $\kappa_t = 0.05$, and (c) $\kappa_t = 0.20$.*

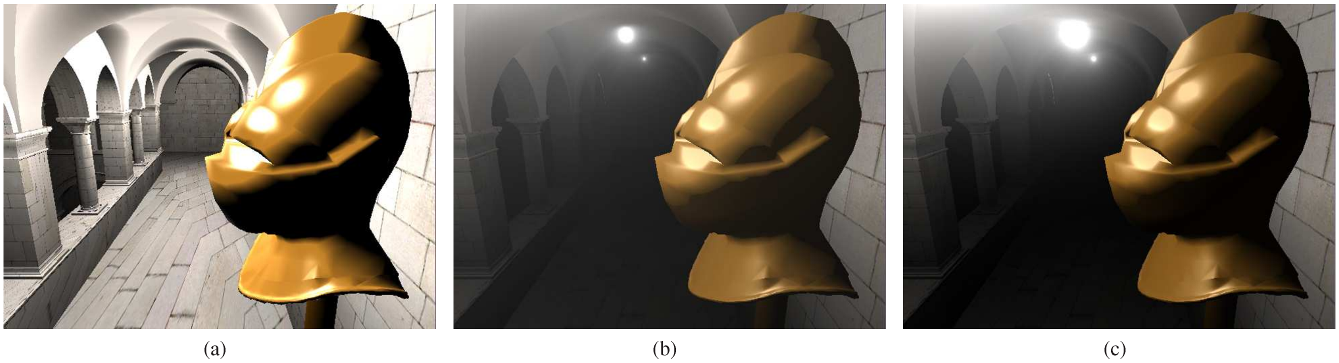

(a) (b) (c)

Figure 12: *Captured images using different levels of forward scattering: (a) clear day, (b) isotropic scattering $g = 0$, and (c) strong forward scattering $g = 0.75$. Notice the glows around light sources and brightening of the shadowed regions on the knight's face in isotropic scattering conditions (b). When increasing the amount of forward scattering in (c), the glows increase in size, and the shadowed regions darken.*

```
uniform sampler2D $H$(x:$H_0$,y:$H_1$,z:$H_2$,w:$H_3$)
uniform sampler2D $J^1$(x:$J^1_0$,y:$J^1_1$,z:$J^1_2$,w:$J^1_3$)
uniform sampler2D $J^{20}$(x:$J^{20}_0$),y:$J^{20}_1$,z:$J^{20}_2$,w:$J^{20}_3$)
uniform float $\kappa_t$
uniform $w_0, w_1, w2, w3$ $[w_k = (2k+1)g^k]$

main()

Find $\gamma, d_{sv}, d_{sp}, \theta_s, \theta_s\prime$.
$h = d_{sv} * \sin\gamma;$
$t = d_{sv} * \cos\gamma;$
$T_{sp} = d_{sp} * \kappa_t$
$C_0 = \kappa_t * I_0 * exp(-\kappa_t * t)/h;$
$C_1 = \kappa_t * h;$
$\theta_0 = \gamma - 0.5 * \pi;$
$\theta_d = \arccos(h/d_{sp})$

*****airlight equation 14*****
$h_1$ = texture2D($H, vec2(C_1, \theta_d)$)
$h_2$ = texture2D($H, vec2(C_1, \theta_0)$)
airlight = $C_0 * [w_0 * (h_1.x - h_2.x) + w_1 * (h_1.y - h_2.y) +$
$w_2 * (h_1.z - h_2.z) + w_3 * (h_1.w - h_2.w)];$

*****diffuse surface radiance, equation 10 and 16*****
$j^1$=texture2D($J^1, vec2(T_{sp}, \theta_s)$);
$d_1 = I_0 * k_d * exp(-T_{sp}) * \cos\theta_s/(d_{sp} * d_{sp});$
$d_2 = I_0 * k_d * \kappa_t * [w_0 * j^1.x + w_1 * j^1.y + w_2 * j^1.z + w_3 * j^1.w]/d_{sp};$
diffuse= $d_1 + d_2$;

*****specular surface radiance, equation 10 and 16*****
$j^{20}$ =texture2D($J^{20}, vec2(T_{sp}, \theta_s\prime)$);
$s_1 = I_0 * k_s * exp(-T_{sp}) * \cos^n\theta_s\prime/(d_{sp} * d_{sp});$
$s_2 = I_0 * k_s * \kappa_t * [w_0 * j^{20}.x + w_1 * j^{20}.y + w_2 * j^{20}.z + w_3 * j^{20}.w]/d_{sp};$

*****equation 2*****
glFragColor = airlight + (diffuse + specular)$* exp(-\kappa_t * d_{vp})$;
```

Figure 10: *Fragment shader pseudocode for lighting model with dynamic anisotropic scattering using $N = 3$ Legendre expansion terms.*

media. This flexibility is a key step towards practical inclusion of complex volumetric scattering effects in the general real-time application.

A possible direction of future work is the development of an analytic angle-formulated scattering model which considers anisotropic light sources, since the intensity is singularly dependent upon only angle. The reduction of complexity using angle reformulations may also extend to outdoor scattering with non-uniform density scattering effects. Lastly, algorithms for volumetric shadows using analytic techniques has potential for real-time applications.

Acknowledgements

This project was fully supported by Silicon Integrated Systems Corp. Thanks to Chang Jyh Shyang for modeling help and Warren Lin for editing work. Thanks to http://hdri.cgtechniques.com/ for use of the sponza atrium model.

References

ASHIKHMIN, M., PREMOZE, S., RAMAMOORTHI, R., AND NAYAR, S., 2004. Blurring of light due to multiple scattering by the medium: a path integral approach. Technical report CUCS-017-04, Columbia University, April.

CEREZO, E., PEREZ-CAZORLA, F., PUEYO, X., SERON, F., AND SILLION, F. 2005. A survey on participating media rendering techniques. *The Visual Computer*.

DOBASHI, Y., KANEDA, K., YAMASHITA, H., OKITA, T., AND NISHITA, T. 2000. A simple, efficient method for realistic animation of clouds. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 19–28.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2000. Interactive rendering method for displaying shafts of light. In *PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, 31.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2002. Interactive rendering of atmospheric scattering effects using graphics hardware. In *HWWS '02: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, 99–107.

DUNN, A. 1997. *Light Scattering Properties of Cells*. PhD thesis, University of Texas at Austin.

HARRIS, M. J., AND LASTRA, A. Real-time cloud rendering. In *EG 2001 Proceedings*.

HEGEMAN, K., ASHIKHMIN, M., AND PREMOZE, S. 2005. A lighting model for general participating media. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, 117–124.

HOFFMAN, N., AND PREETHAM, A. J. 2003. Real-time light-atmosphere interactions for outdoor scenes. *Graphics programming methods*, 337–352.

ISHIMARU, A. 1997. *Wave Propagation and Scattering in Random Media*. IEEE Press.

IWASAKI, K., DOBASHI, Y., AND NISHITA, T. 2003. A volume rendering approach for sea surfaces taking into account second order scattering using scattering maps. In *VG '03: Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics*, 129–136.

JENSEN, H. W., AND CHRISTENSEN, P. H. 1998. Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 311–320.

JENSEN, H. W. 2001. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd.

KAJIYA, J. T., AND HERZEN, B. P. V. 1984. Ray tracing volume densities. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, 165–174.

LECOCQ, P., KEMENY, A., MICHELIN, S., AND ARQUES, D. 2000. Mathematical approximation for real-time lighting rendering through participating media. In *PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, 400.

NARASIMHAN, S. G., AND NAYAR, S. K. 2003. Shedding light on the weather. In *IEEE CVPR*.

NISHITA, T., OKAMURA, I., AND NAKAMAE, E. 1985. Shading models for point and linear sources. *ACM Trans. Graph. 4*, 2, 124–146.

NISHITA, T., MIYAWAKI, Y., AND NAKAMAE, E. 1987. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 303–310.

NISHITA, T., SIRAI, T., TADAMURA, K., AND NAKAMAE, E. 1993. Display of the earth taking into account atmospheric scattering. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 175–182.

NISHITA, T., DOBASHI, Y., AND NAKAMAE, E. 1996. Display of clouds taking into account multiple anisotropic scattering and sky light. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 379–386.

PHARR, M., AND HUMPHREYS, G. 2004. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc.

PREETHAM, A. J., SHIRLEY, P., AND SMITS, B. 1999. A practical analytic model for daylight. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 91–100.

PREMOZE, S., ASHIKHMIN, M., AND SHIRLEY, P. 2003. Path integration for light transport in volumes. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, 52–63.

PREMOZE, S., ASHIKHMIN, M., RAMAMOORTHI, R., AND NAYAR, S. K. 2004. Practical rendering of multiple scattering effects in participating media. In *Rendering Techniques*, 363–373.

RAMAMOORTHI, R., AND HANRAHAN, P. 2002. Frequency space environment map rendering. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 517–526.

RILEY, K., EBERT, D. S., KRAUS, M., TESSENDORF, J., AND HANSEN, C. 2004. Efficient rendering of atmospheric phenomena. In *Eurographics Symposium on Rendering*, 375–386.

SUN, B., RAMAMOORTHI, R., NARASIMHAN, S. G., AND NAYAR, S. K. 2005. A practical analytic single scattering model for real-time rendering. *ACM Transactions on Graphics (SIGGRAPH)* (August).

VENCESLAS, B., SYLVAIN, M., AND ARQUES, D. 2006. Real-time rendering of atmospheric lighting and volumetric shadows. In *The 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2006*.

Appendix A: Solution to Surface Scattering Integral

Lambertian BRDF Case

$$L_{p,a} = \int_{\Omega_{2\pi}} L_a(\gamma', d_{sp}, \infty, \kappa_t) brdf(\theta_i, \phi_i, \theta_v, \phi_v) \cos\theta_i d\omega_i$$

Substitute $L_a(\gamma', d_{sp}, \infty, \kappa_t) =$ Equation 6 and $brdf(\theta_i, \phi_i, \theta_v, \phi_v) = k_d$.

$$= \int_{\Omega_{2\pi}} C_0(\gamma', d_{sp}, \kappa_t)[H(C_1(\gamma', d_{sp}, \kappa_t), \frac{\pi}{2}) - H(C_1(\gamma', d_{sp}, \kappa_t), \theta_0')] k_d \cos\theta_i d\omega_i$$

Substitute for $C_0(\gamma', d_{sp}, \kappa t), C_1(\gamma', d_{sp}, \kappa_t), \theta_0',$ and $T_{sp} = \kappa_t d_{sp}$.

Take constants out of integral.

$$= \frac{I_0 k_d \kappa_t}{d_{sp}} \int_{\Omega_{2\pi}} \frac{e^{-T_{sp}\cos\gamma'}}{\sin\gamma'} [H(T_{sp}\sin\gamma', \frac{\pi}{2}) - H(T_{sp}\sin\gamma', \gamma' - \frac{\pi}{2})] \cos\theta_i d\omega_i$$

Recall $\gamma'(\theta_s, \omega_i)$. So

$$= \frac{I_0 k_d \kappa_t}{d_{sp}} J^1(T_{sp}, \theta_s)$$

Phong BRDF Case

$$L_{p,a} = \int_{\Omega_{2\pi}} L_a(\gamma', d_{sp}, \infty, \kappa_t) brdf(\theta_i, \phi_i, \theta_v, \phi_v) \cos\theta_i d\omega_i$$

Let R be the reflection of the view ray about the surface normal.

Reparameterize the integral about R.

Denote reparameterized source direction θ_s'[s.t. $\gamma'(\theta_s', \omega_i)$ and $\theta_0'(\theta_s', \omega_i)$].

Substitute $L_a(\gamma', d_{sp}, \infty, \kappa_t) =$ Equation 6 and $brdf(\theta_i, \phi_i, \theta_v, \phi_v) = k_s \cos^n \theta_i'$.

$$= \int_{\Omega_{2\pi}} C_0(\gamma', d_{sp}, \kappa_t)[H(C_1(\gamma', d_{sp}, \kappa_t), \frac{\pi}{2}) - H(C_1(\gamma', d_{sp}, \kappa_t), \theta_0')] k_s \cos^n \theta_i d\omega_i$$

Substitute for $C_0(\gamma', d_{sp}, \kappa t), C_1(\gamma', d_{sp}, \kappa_t), \theta_0',$ and $T_{sp} = \kappa_t d_{sp}$.

Take constants out of integral.

$$= \frac{I_0 k_d \kappa_t}{d_{sp}} \int_{\Omega_{2\pi}} \frac{e^{-T_{sp}\cos\gamma'}}{\sin\gamma'} [H(T_{sp}\sin\gamma', \frac{\pi}{2}) - H(T_{sp}\sin\gamma', \gamma' - \frac{\pi}{2})] \cos^n \theta_i d\omega_i$$

$$= \frac{I_0 k_d \kappa_t}{d_{sp}} J^n(T_{sp}, \theta_s')$$

Appendix B: First 4 Legendre Polynomials

$$P_0(x) = 1$$
$$P_1(x) = x$$
$$P_2(x) = \frac{1}{2}(3x^2 - 1)$$
$$P_3(x) = \frac{1}{2}(5x^3 - 3x)$$

Appendix C: Solutions for Dynamic Anisotropic Scattering

Solution to Airlight L_a

$$L_a(\gamma, d_{sv}, d_{vp}, \kappa_t) = \frac{\kappa_t e^{-\kappa_t t}}{h} \int_{\theta_0}^{\theta_d} e^{-\kappa_t h \frac{\sin\theta+1}{\cos\theta}} I_0 p(\theta + \frac{\pi}{2}) d\theta.$$

Substitute Equation 13 for $p(\theta + \frac{\pi}{2})$.

Take constants out of integral.

$$= C_0 \sum_{k=0}^{N} (2k+1) g^k \int_{\theta_0}^{\theta_d} e^{\kappa_t h \frac{\sin\theta+1}{\cos\theta}} P_k[cos(\theta + \frac{\pi}{2})] d\theta.$$

$$= C_0 \sum_{k=0}^{N} (2k+1) g^k [H_k(C_1, \theta_d) - H_k(C_1, \theta_0)],$$

Solution to $L_{p,a}$

$$L_{p,a} = \int_{\Omega_{2\pi}} L_a(\gamma\prime, d_{sp}, \infty, \kappa_t) brdf(\theta_i, \phi_i, \theta_v, \phi_v) \cos\theta_i d\omega_i$$

Substitute $L_a(\gamma\prime, d_{sp}, \infty, \kappa_t) =$ Equation 14.

$$= \int_{\Omega_{2\pi}} C_0 \sum_{k=0}^{N} (2k+1) g^k [H_k(C_1, \theta_d) - H_k(C_1, \theta_0)] brdf(\theta_i, \phi_i, \theta_v, \phi_v) \cos\theta_i d\omega_i$$

Follow steps for Lambertian and Phong BRDF cases in Appendix A.

$$= \frac{I_0 \kappa_t}{d_{sp}} \sum_{k=0}^{N} (2k+1) g^k [k_d J_k^1(T_{sp}, \theta_s) + k_s J_k^n(T_{sp}, \theta_s\prime)].$$