Three-dimensional Neumann-series approach to model light transport in nonuniform media

Abhinav K. Jha,[1,*] Matthew A. Kupinski,[1,2] Harrison H. Barrett,[1,2] Eric Clarkson,[1,2] and John H. Hartman[3]

[1]*College of Optical Sciences, University of Arizona, Tucson, Arizona 85721, USA*
[2]*Department of Radiology, University of Arizona, Tucson, Arizona 85724, USA*
[3]*Department of Computer Science, University of Arizona, Tucson, Arizona 85721, USA*
**Corresponding author: akjha@email.arizona.edu*

Received March 29, 2012; revised June 29, 2012; accepted July 2, 2012;
posted July 2, 2012 (Doc. ID 165111); published August 17, 2012

We present the implementation, validation, and performance of a three-dimensional (3D) Neumann-series approach to model photon propagation in nonuniform media using the radiative transport equation (RTE). The RTE is implemented for nonuniform scattering media in a spherical harmonic basis for a diffuse-optical-imaging setup. The method is parallelizable and implemented on a computing system consisting of NVIDIA Tesla C2050 graphics processing units (GPUs). The GPU implementation provides a speedup of up to two orders of magnitude over non-GPU implementation, which leads to good computational efficiency for the Neumann-series method. The results using the method are compared with the results obtained using the Monte Carlo simulations for various small-geometry phantoms, and good agreement is observed. We observe that the Neumann-series approach gives accurate results in many cases where the diffusion approximation is not accurate. © 2012 Optical Society of America

OCIS codes: 170.3660, 000.3860, 110.2990, 110.6955.

1. INTRODUCTION

Diffuse optical tomography (DOT) is an emerging noninvasive biomedical imaging technique in which images of optical properties of the object are derived based on the measurements of near-infrared (NIR) light on the surface of the object [1–4]. The modality has received significant attention in the past decade due to its capability to provide functional images of the tissue under investigation using nonionizing radiation. DOT has been applied in breast-cancer detection and characterization [5,6], in functional brain imaging [7,8], in imaging of small joints for early diagnosis of rheumatoid arthritis [9], and in small-animal imaging for studying physiological processes and pathologies [10]. However, the task of image reconstruction in DOT is a nonlinear ill-posed inverse problem [4,11]. Small errors in measurements or inaccuracies in modeling the DOT system can cause significant errors in the reconstruction task. Therefore, it is essential that the modeling of the DOT system be very accurate. An important component of modeling the DOT system is to simulate the propagation of light through biological tissue. Models that accurately describe light propagation within biological tissues are therefore required.

A major difficulty in simulating light transport in biological tissue at NIR wavelengths is the phenomenon of scattering that occurs at these wavelengths in the tissue. We can use the radiative transport equation (RTE) to account for the scattering effects, but the RTE is a computationally intensive integro-differential equation. To reduce the computational complexity, a simplified approximation of the RTE, termed the diffusion approximation, is widely used [2,12,13]. The diffusion approximation has been implemented with finite element methods [14–17] and boundary element methods [18,19]. The approximation assumes that light propagates diffusely in tissues, an assumption that breaks down near tissue surfaces, in anisotropic tissues, and in high-absorption or low-scatter regions [1,3,20,21]. As a result, the diffusion approximation cannot model light propagation accurately in highly absorbing regions such as haematoma, and voidlike spaces such as ventricles and the subarachnoid space [21–25]. Moreover, when imaging small-tissue geometries, e.g., whole-body imaging of small animals, the diffusion model is not very accurate [20].

Higher-order approximations to the RTE such as the discrete-ordinates method (S_N) [26] and spherical harmonic equations (P_N) [27,28] have been developed to overcome these issues. The S_N method has been implemented with finite difference [21,24,29] and boundary element [30] methods, and the P_N method has been used along with finite element methods [31,32]. Although the approximations lead to exact solutions as $N \to \infty$, they require solutions to many coupled differential equations, so these methods are still computationally very expensive. To illustrate their computational requirement, a full three-dimensional (3D) reconstruction of the mouse model to recover the fluorescent-probe distribution can take up to several hours or days of computation time using these methods [20]. To improve the computational efficiency, a simplified spherical harmonics (SP_N) approximation has been validated [20,25]. The method is asymptotic, but it can model light propagation with small error in some circumstances, and is computationally faster.

There has thus been significant research in using the RTE to model light propagation for optical imaging. However, most of these studies have focused on using differential methods to solve the RTE. In nuclear imaging, integral approximations to the RTE are often used to model photon propagation [33–35]. The advantage with the integral methods is that, unlike the

 © 2012 Optical Society of America

differential methods, they do not require solutions to a large number of coupled equations. However, in nuclear imaging, photon propagation mostly occurs in the forward direction, and scattering effects are minimal. While the same is not true with optical imaging, the scattering phenomenon is strongly forward biased even in optical imaging [20,36–38]. Therefore, it is of interest for us to study the validity and performance of the integral form of the RTE, using a Neumann-series formulation, in diffuse optical imaging.

The Neumann-series formulation of the RTE has been explored for various tasks, such as modeling scattering effects in nuclear imaging [33,39,40], neutron-transport modeling [41], retrieving atmospheric properties from remotely sensed microwave observations [42], and modeling the scatter of sunlight in the atmosphere [43]. However, in optical imaging, the scattering phase function, imaging-system geometry, light emission source, and some other factors are quite different from the above-mentioned cases. Our study focuses on developing the mathematical methods and the software to implement the Neumann-series method specifically for optical imaging. As part of this study, we have developed a software to solve the Neumann-series RTE, specifically for uniform media [44]. The experiments with the uniform-medium software provide us with numerous insights into the feasibility, convergence characteristics, and computational requirements of the Neumann-series approach. We have presented these insights, along with a detailed treatment of the theory of the Neumann-series approach to model light propagation in uniform medium, in another paper [44]. In optical imaging, the problem of simulating light propagation in nonuniform tissue is also very important. The software we developed cannot be used for nonuniform media. Thus, in parallel, we attempted to solve the more general problem of light propagation in heterogeneous media. In a preliminary study, we have shown that the RTE in integral form can model photon propagation in simple heterogeneous media [45]. In the presented work, we extend the approach to completely nonuniform 3D media. We express the RTE as a Neumann series and solve it in the spherical harmonic basis. Implementation of this method on computing systems with only sequential processing units takes considerable execution time. However, the method is parallelizable, and to reduce the computational time, we implement it on parallel processing architectures, more specifically the NVIDIA graphics processing units (GPUs).

In this paper, our main objectives are the following: to present the theory and implementation details of the Neumann-series approach specific to a nonuniform medium, to describe the implementation of the algorithm for NVIDIA GPUs, to demonstrate that the method computes accurate results with small-phantom geometry for nonuniform medium, and to show the speedup obtained with our GPU implementation. Since we have presented detailed theory of the Neumann-series approach for optical imaging in uniform media [44], the mathematical treatment that is similar for uniform and nonuniform media will be just summarized in this paper.

2. THEORY

A. Radiative Transport Equation

The RTE takes into account all the physical processes of absorption, scattering, emission, and propagation of radiation through a medium. The fundamental radiometric quantity that we describe using the RTE is the distribution function $w(\boldsymbol{r},\hat{s},\mathcal{E},t)$. In terms of photons, $w(\boldsymbol{r},\hat{s},\mathcal{E},t)\Delta V\Delta\Omega\Delta\mathcal{E}$ can be interpreted as the number of photons contained in volume ΔV centered on the 3D position vector $\boldsymbol{r}=(x,y,z)$, traveling in a solid angle $\Delta\Omega$ about direction $\hat{s}=(\theta,\phi)$, and having energies between $\mathcal{E}$ and $\mathcal{E}+\Delta\mathcal{E}$ at time t. The emission source is described using the function $\Xi(\boldsymbol{r},\hat{s},\mathcal{E},t)$. In the DOT implementation, we assume a monoenergetic time-independent emission source, so that the emission function can be written as $\Xi(\boldsymbol{r},\hat{s})$. Moreover, in optical imaging, elastic scattering is the dominant scattering mechanism, and thus the scattered photon does not lose any energy. Since there are no other energy-loss mechanisms for the photon, the dependence of the distribution function on energy is dropped completely.

Let $\mathcal{K}$ and $\mathcal{X}$ denote the scattering and attenuation operators in integral form, which represent the effect of scattering, and the effect of attenuation and propagation of photons, respectively. Let the absorption and scattering coefficients at location $\boldsymbol{r}$ be denoted by $\mu_a(\boldsymbol{r})$ and $\mu_s(\boldsymbol{r})$, respectively, and let $\mu_{tot}(\boldsymbol{r})=\mu_a(\boldsymbol{r})+\mu_s(\boldsymbol{r})$. The effect of the scattering operator $\mathcal{K}$ on the distribution function is given by [46]

$$[\mathcal{K}w](\boldsymbol{r},\hat{s},t)=\int_{4\pi}d\Omega' K(\hat{s},\hat{s}'|\boldsymbol{r})w(\boldsymbol{r},\hat{s}',t), \tag{1}$$

where $\hat{s}$ and $\hat{s}'$ denote the direction of the outgoing and incoming photons, respectively, and $K(\hat{s},\hat{s}'|\boldsymbol{r})$ is the scattering kernel. Since the scattering phase function in biological tissue is typically given by the Henyey–Greenstein function [25,47], the scattering kernel is [44]

$$K(\hat{s},\hat{s}'|\boldsymbol{r})=\frac{\mu_s(\boldsymbol{r})c_m}{4\pi}\left\{\frac{1-g^2}{[1+g^2-2g\cos(\hat{s}\cdot\hat{s}')]^{3/2}}\right\}, \tag{2}$$

where c_m denotes the speed of light in the medium. The anisotropy factor g characterizes the angular distribution of scattering in the tissue.

The attenuation operator $\mathcal{X}$ is the standard attenuated x-ray transform, and its effect on the distribution function is given by [46]

$$[\mathcal{X}w](\boldsymbol{r},\hat{s},t)=\frac{1}{c_m}\int_{\lambda=0}^{\infty}d\lambda w(\boldsymbol{r}-\hat{s}\lambda,\hat{s},t)\times\exp\left[-\int_0^{\lambda}d\lambda'\mu_{tot}(\boldsymbol{r}-\hat{s}\lambda')\right]. \tag{3}$$

This transform denotes that the radiance at location $\boldsymbol{r}$ in direction $\hat{s}$ can be found by integrating the source distribution along a line parallel to $\hat{s}$ and passing through the point $\boldsymbol{r}$, where the more distant points along the line contribute less due to the exponential attenuation factor.

In terms of the defined attenuation and scattering operators, for the monoenergetic time-independent source $\Xi(\boldsymbol{r},\hat{s})$, the RTE can be written as a Neumann series [44,46]

$$w(\boldsymbol{r},\hat{s})=\mathcal{X}\Xi+\mathcal{X}\mathcal{K}\mathcal{X}\Xi+\mathcal{X}\mathcal{K}\mathcal{X}\mathcal{K}\mathcal{X}\Xi+\ldots, \tag{4}$$

where we note that the distribution function is also no more a function of time. An intuitive way to interpret this Neumann-series solution is that successive terms in the series represent

successive scattering events; in fact photons that have scattered n times contribute to the term $\mathcal{X}(\mathcal{K}\mathcal{X})^n\Xi$ in this series [46]. Figure 1 schematically illustrates the effect of the initial terms of the Neumann series.

This section has summarized the general theory behind the Neumann-series RTE, which has been presented in more detail by Jha *et al.* [44]. We will now discuss the implementation details, where the framework and mathematical treatment we suggest will be general enough to model light propagation through nonuniform media. Therefore, the sections that follow are the specific contribution of this paper.

B. Discretization: Spherical Harmonics and Voxel Basis

Implementation of the RTE on a computing system requires discretization of the distribution function along the angular and spatial coordinates. To discretize the distribution function along the angular coordinates, we realize that the scattering kernel is only a function of the dot product between the entrance and exit direction of the photons, i.e., $\hat{s}\cdot\hat{s}'$ [Eq. (2)]. In the language of group theory, the symmetry group of the $\mathcal{K}$ operator is $SO(3)$, and the spherical harmonics are basis functions for irreducible representations of $SO(3)$ [46]. Since $\mathcal{K}$ is invariant to this group, the scatter kernel takes a simple diagonal form in the spherical harmonic basis [46]. To use this simplification, we solve the RTE in a spherical harmonic basis. The distribution function is expressed in the spherical harmonic basis as

$$w(\boldsymbol{r},\hat{s}) = \sum_{l=0}^{\infty}\sum_{m=-1}^{l} W_{lm}(\boldsymbol{r})Y_{lm}(\hat{s}), \tag{5}$$

where the spherical harmonic basis functions are denoted by $Y_{lm}(\hat{s})$, and the distribution function in spherical harmonics is denoted by $W_{lm}(\boldsymbol{r})$. For ease of notation, we denote all the spherical harmonic coefficients $W_{lm}(\boldsymbol{r})$ by $\boldsymbol{W}$.

To discretize the spatial variable $\boldsymbol{r}$, we use a voxel basis. We consider the domain over which we model photon propagation to be divided into voxels of length Δx, Δy, and Δz along the x, y, and z axis, respectively. The number of voxels along the x, y, and z axis are denoted by I, J, and K, respectively. We define the voxel basis function $\psi_{ijk}(\boldsymbol{r})$ to be 0 outside the (i,j,k)th voxel and 1 inside that voxel, where (i,j,k) denote the 3D index of the voxel along the x, y, and z axis, respectively. The center of the (i,j,k)th voxel is denoted by $\boldsymbol{r}_{ijk} = (x_i, y_j, z_k)$. For brevity of notation, we will often denote the 3D index (i,j,k) by the 3D vector $\boldsymbol{v}$. The voxel basis is an orthogonal basis, so the coefficients in this basis are obtained

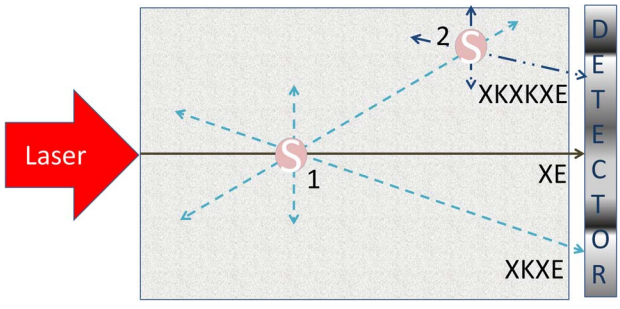

Fig. 1. (Color online) DOT setup: a 3D cuboidal phantom with a laser source along the optical axis. The figure also roughly shows the effect of different terms of the Neumann series.

by a simple integration of the distribution function inside the volume. For the (i,j,k)th voxel, let us denote the (l,m)th spherical harmonic coefficient of the distribution function by $W_{lm}(i,j,k)$. Then, we can obtain $W_{lm}(i,j,k)$ from $W_{lm}(\boldsymbol{r})$ as

$$W_{lm}(i,j,k) = \frac{1}{\Delta V}\int_{S_{ijk}} \mathrm{d}^3\boldsymbol{r} W_{lm}(\boldsymbol{r}), \tag{6}$$

where S_{ijk} denotes the support of the voxel function $\psi_{ijk}(\boldsymbol{r})$, and $\Delta V = \Delta x\Delta y\Delta z$ is the volume of the voxel. The distribution function is then represented in the spherical harmonic and voxel basis as

$$w(\boldsymbol{r},\hat{s}) = \sum_{i,j,k=1}^{I,J,K}\sum_{l=0}^{L}\sum_{m=-l}^{l} W_{lm}(i,j,k)\psi_{ijk}(\boldsymbol{r})Y_{lm}(\hat{s}), \tag{7}$$

where we truncate the spherical harmonic expansion at $l = L$. For ease of notation, we denote the distribution function represented in the spherical harmonic and voxel basis by $\boldsymbol{W}_{\mathbf{d}}$. The absorption and scattering coefficients are also discretized in the voxel basis. For the (i,j,k)th voxel, the absorption and scattering coefficients are denoted by $\mu_{\mathrm{a}}(i,j,k)$ and $\mu_{\mathrm{s}}(i,j,k)$, respectively.

Let us denote the scattering operator in the spherical harmonics and voxel basis by the discrete-discrete operator $\boldsymbol{D}$. The effect of this operator on the distribution function in a spherical harmonic and voxel basis is given by

$$[\boldsymbol{D}\boldsymbol{W}_{\mathbf{d}}]_{lm}(i,j,k) = \sum_{l'm'} W_{l'm'}(i,j,k)D_{lm,l'm'}(i,j,k), \tag{8}$$

where $D_{lm,l'm'}(i,j,k)$ denotes the kernel of the scattering operator in the spherical harmonic and voxel basis. As we mentioned earlier, since the scattering operator $\mathcal{K}$ depends only on the cosine of the angle between the entrance and exit direction of the photons, using Eqs. (1), (2), and (5), we can derive that the operator $\boldsymbol{D}$ is just a diagonal matrix with elements given by [44]

$$D_{lm,l'm'}(i,j,k) = c_{\mathrm{m}}\mu_{\mathrm{s}}(\boldsymbol{r})g^l\delta_{ll'}\delta_{mm'}, \tag{9}$$

where $\delta_{ll'}$ is the Kronecker delta. As we observe, this matrix exists only for $l = l'$ and $m = m'$, since the scattering operator is diagonalized in the spherical harmonic basis.

Let us denote the attenuation operators in the spherical harmonic basis, and the spherical harmonic and voxel basis, by $\mathcal{A}$ and $\boldsymbol{A}$, respectively. The effect of the attenuation operator $\mathcal{A}$ on the distribution function in spherical harmonic basis is given by

$$[\mathcal{A}\boldsymbol{W}]_{lm}(\boldsymbol{r}) = \sum_{l'=0}^{\infty}\sum_{m'=-l}^{l}\int_{\infty} \mathrm{d}^3\boldsymbol{r}' W_{l'm'}(\boldsymbol{r}')A_{lm,l'm'}(\boldsymbol{r},\boldsymbol{r}'), \tag{10}$$

where the elements of the attenuation kernel are given by

$$A_{lm,l'm'}(\boldsymbol{r},\boldsymbol{r}') = \frac{1}{c_{\mathrm{m}}|\boldsymbol{r}-\boldsymbol{r}'|^2}Y^*_{lm}\left(\frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|}\right)Y_{l'm'}\left(\frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|}\right) \times \exp\left[-\int_{\lambda'=0}^{|\boldsymbol{r}-\boldsymbol{r}'|}\mathrm{d}\lambda'\mu_{\mathrm{tot}}\left(\boldsymbol{r}-\lambda'\frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|}\right)\right]. \tag{11}$$

The effect of the attenuation operator on the distribution function in the spherical harmonic and voxel basis is given by

$$[\boldsymbol{A}\boldsymbol{W}_{\mathbf{d}}]_{lm}(i,j,k) = \sum_{i',j',k'=1}^{I,J,K} \sum_{l'=0}^{L} \sum_{m'=-l'}^{l'} A_{lm,l'm'}(i,j,k,i',j',k') \times W_{l'm'}(i',j',k'). \tag{12}$$

Thus the attenuation operator $\boldsymbol{A}$ accounts for the effect of the distribution function at the (i',j',k'), or the ν'th voxel, on the distribution function at the (i,j,k), or the νth voxel. This effect depends on the location of both the voxels and the path between them. Let us denote the vector that joins the center of these two voxels by $R_{\nu\nu'}\hat{\boldsymbol{u}}_{\nu\nu'}$. While $R_{\nu\nu'}$ denotes the magnitude of the vector, and, therefore, is the distance between the two voxel midpoints, $\hat{\boldsymbol{u}}_{\nu\nu'}$ denotes the unit vector joining the two voxel midpoints. Let $\hat{\boldsymbol{u}}_{\nu\nu'} = u_x\hat{\boldsymbol{x}} + u_y\hat{\boldsymbol{y}} + u_z\hat{\boldsymbol{z}}$, where $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ denote the unit vectors along the x, y, and z axis, respectively. Then, using Eqs. (6), (10), (11), and (12), we get the elements of the attenuation operator in matrix form as

$$A_{lm,l'm'}(i,j,k,i',j',k') = \frac{\Delta V}{c_{\mathrm{m}} R_{\nu\nu'}^2} Y_{lm}^*(\hat{\boldsymbol{u}}_{\nu\nu'}) Y_{l'm'}(\hat{\boldsymbol{u}}_{\nu\nu'}) \times \exp\left[-\sum_{q=0}^{R_{\nu\nu'}/\Delta\lambda'} \Delta\lambda' \mu_{\mathrm{tot}}\left(i - \frac{q u_x \Delta\lambda'}{\Delta x}, j - \frac{q u_y \Delta\lambda'}{\Delta y}, k - \frac{q u_z \Delta\lambda'}{\Delta z} \right) \right]. \tag{13}$$

Using this expression, we can evaluate the effect due to the attenuation operation between photons present in two different voxels. The photons in the same voxel also have an effect on each other due to the attenuation operation. This effect cannot be computed using the above expression, due to the $1/R_{\nu\nu'}^2$ term, which is zero when $\nu = \nu'$. Therefore, we consider a different approach in this case.

Let us consider that we wish to evaluate the effect that the photons in the (i,j,k)th voxel have on each other. We then require a method to determine the attenuation-kernel element $A_{lm,l'm'}(i,j,k,i,j,k)$, which we henceforth denote by $A_{lm,l'm'}(i,j,k)$ for ease of notation. We could consider determining this quantity by performing a numerical integration of the expression for $A_{lm,l'm'}(\boldsymbol{r},\boldsymbol{r}')$, given by Eq. (11), over the space of the (i,j,k)th voxel in Cartesian coordinates. However, this is difficult due to the $1/|\boldsymbol{r}-\boldsymbol{r}'|^2$ term in this expression, which becomes very high for locations $\boldsymbol{r}$ and $\boldsymbol{r}'$ within the same voxel. To solve this issue, we evaluate the attenuation operation [Eq. (10)] for this case in spherical coordinates instead. Let us denote the vector $\boldsymbol{r}-\boldsymbol{r}'$ by $R\hat{\boldsymbol{u}}$, so that we can replace $d^3\boldsymbol{r}'$ by $R^2 dR d\Omega_u$ in Eq. (10), where $d\Omega_u$ corresponds to the infinitesimally small solid angle around the unit vector $\hat{\boldsymbol{u}}$. Further, substituting the expression for $A_{lm,l'm'}(\boldsymbol{r},\boldsymbol{r}')$ from Eq. (11) into Eq. (10), we get

$$[\mathcal{A}W]_{lm}(\boldsymbol{r}) = \frac{1}{c_{\mathrm{m}}} \sum_{l'm'} \int_{4\pi} d\Omega_u Y_{lm}^*(\hat{\boldsymbol{u}}) Y_{l'm'}(\hat{\boldsymbol{u}}) \int_0^{\infty} dR W_{l'm'}(\boldsymbol{r}') \times \exp\left[-\int_{\lambda'=0}^{R} d\lambda' \mu_{\mathrm{tot}}(\boldsymbol{r} - \hat{\boldsymbol{u}}\lambda') \right], \tag{14}$$

so that the $1/|\boldsymbol{r}-\boldsymbol{r}'|^2$ term cancels out. We next use Eq. (6) in the above expression to evaluate the effect of the photons over the (i,j,k)th voxel. We find that since we are integrating only over the volume of the (i,j,k)th voxel, $\mu_{\mathrm{tot}}(\boldsymbol{r}-\hat{\boldsymbol{u}}\lambda')$ and $W_{l'm'}(\boldsymbol{r}')$ are constant over this volume in the voxel basis and equal to $\mu_{\mathrm{tot}}(i,j,k)$ and $W_{l'm'}(i,j,k)$, respectively. Therefore, these terms come out of the integral over λ' and R, respectively. The exponential integral is then simply equal to $\exp[-\mu_{\mathrm{tot}}(i,j,k)R]$. Also, for a particular direction $\hat{\boldsymbol{u}} = (\theta_u,\phi_u)$, the integral over R varies only from 0 to $\beta(\theta_u,\phi_u)$, where $\beta(\theta_u,\phi_u)$ is the distance from the center of the voxel to the face of the voxel for a particular direction. Thus, using Eq. (6) in Eq. (14) yields

$$[\boldsymbol{A}_{\boldsymbol{D}}\boldsymbol{W}_{\mathbf{d}}]_{lm}(i,j,k) = \frac{1}{c_{\mathrm{m}}} \sum_{l'm'} W_{l'm'}(i,j,k) \int_{\theta=0}^{\pi} d\theta \sin\theta \times \int_{\phi=0}^{2\pi} d\phi Y_{lm}^*(\hat{u}) Y_{l'm'}(\hat{u}) \times \left\{ \frac{1 - \exp[-\mu_{\mathrm{tot}}(i,j,k)\beta(\theta_u,\phi_u)]}{\mu_{\mathrm{tot}}(i,j,k)} \right\}, \tag{15}$$

where $\boldsymbol{A}_{\boldsymbol{D}}$ is a diagonal matrix in the voxel basis that describes the effect that photons in a voxel have on each other. We evaluate the integral over (θ_u,ϕ_u) numerically by sampling the angular coordinates at discrete values, and determining $\beta(\theta_u,\phi_u)$ for each value. To determine $\beta(\theta_u,\phi_u)$, we inscribe the voxel with a sphere. We find the coordinates at which the ray from the center of the voxel at an angle (θ_u,ϕ_u) intersects the sphere. From these coordinates, we determine the coordinate of intersection of the ray with the planes corresponding to the voxel faces that lie in the path of the ray. The distance of the closest intersection gives $\beta(\theta_u,\phi_u)$. Therefore, the effect that photons in the same voxel have on each other is evaluated. Equation (15) is written in short-hand notation as

$$[\boldsymbol{A}_{\boldsymbol{D}}\boldsymbol{W}_{\mathbf{d}}]_{lm}(i,j,k) = \sum_{l'm'} W_{l'm'}(i,j,k) A_{lm,l'm'}(i,j,k). \tag{16}$$

Finally, the RTE in the spherical harmonic and voxel basis is given by [44]

$$\boldsymbol{W}_{\mathbf{d}} = \boldsymbol{A}\boldsymbol{\xi}_{\mathbf{d}} + \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}\boldsymbol{\xi}_{\mathbf{d}} + \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}\boldsymbol{D}\boldsymbol{A}\boldsymbol{\xi}_{\mathbf{d}} + \ldots, \tag{17}$$

where $\boldsymbol{\xi}_{\mathbf{d}}$ denotes the source function in the spherical harmonic and voxel basis.

C. Procedure

In our DOT set up, shown in Fig. 1, the source outputs a unidirectional beam along the optical axis, which we refer to as the z axis. We denote the two-dimensional (2D) profile of this beam by the function $h(x,y)$. Let the radiant exitance of the source across the circular region be α. The source can then be represented as

$$\Xi(\boldsymbol{r},\hat{\boldsymbol{s}}) = \alpha\delta(\hat{\boldsymbol{s}} - \hat{\boldsymbol{z}})h(x,y)\delta(z), \tag{18}$$

where $\delta(z)$ indicates that the photons are being emitted from only the $z = 0$ plane, and $\delta(\hat{\boldsymbol{s}} - \hat{\boldsymbol{z}})$ indicates that the source is a unidirectional beam along the z axis. To solve the RTE in spherical harmonic basis, we will have to represent this source term in the spherical harmonic basis. It is practically

impossible to do so for a beamlike source function, since that requires an infinite number of spherical harmonic coefficients. However, in many DOT setups, the emission source is a pencil beam or a combination of pencil-beam-like sources. We apply some mathematical procedures to solve this issue. We first realize that the $\mathcal{X}\Xi$ term is just an attenuation transform applied on a unidirectional-beam source. Thus, it can be easily computed in the normal $(\boldsymbol{r},\hat{s})$ basis. Substituting Eq. (18) into Eq. (3), we can derive that in our experimental setup, the distribution function due to the $\mathcal{X}\Xi$ term is given by

$$\begin{aligned}[\mathcal{X}\Xi](\boldsymbol{r},\hat{s}) &= \frac{\alpha}{c_{\mathrm{m}}}\int_0^{\infty} \mathrm{d}\lambda\,\delta(\hat{s}-\hat{z})h(x,y)\delta(z-\hat{z}\lambda)\\ &\quad\times\exp\left[-\int_0^{\lambda}\mathrm{d}\lambda'\mu_{\mathrm{tot}}(\boldsymbol{r}-\hat{s}\lambda')\right]\\ &= \frac{\alpha}{c_{\mathrm{m}}}\delta(\hat{s}-\hat{z})h(x,y)\exp\left[-\int_0^{z}\mathrm{d}\lambda'\mu_{\mathrm{tot}}(\boldsymbol{r}-\hat{z}\lambda')\right], \quad (19)\end{aligned}$$

where we have used the sifting property of the delta function. Also, in the Neumann series in spherical harmonic basis [Eq. (17)], except for the first term, $\boldsymbol{\xi_{\mathrm{d}}}$ is always preceded by $\boldsymbol{DA}$. In fact, an alternative way to rewrite the RTE is

$$\boldsymbol{W_{\mathrm{d}}} = \boldsymbol{A\xi_{\mathrm{d}}} + \boldsymbol{A}\left[\sum_{n=0}^{\infty}(\boldsymbol{DA})^n\right](\boldsymbol{DA\xi_{\mathrm{d}}}). \quad (20)$$

The representation of $\boldsymbol{DA\xi_{\mathrm{d}}}$ is feasible in the spherical harmonic basis since the scattering operator causes the pencil beam to spread out. Therefore, we compute the spherical harmonic coefficients for $\boldsymbol{DA\xi_{\mathrm{d}}}$. Using Eqs. (1), (2), and (19), the term $[\mathcal{K}\mathcal{X}\Xi](\boldsymbol{r},\hat{s})$ is derived to be

$$\begin{aligned}[\mathcal{K}\mathcal{X}\Xi](\boldsymbol{r},\hat{s}) &= \frac{\alpha\mu_{\mathrm{s}}(\boldsymbol{r})}{4\pi}h(x,y)\left\{\frac{1-g^2}{[1+g^2-2g\,\cos(\hat{s}\cdot\hat{z})]^{3/2}}\right\}\\ &\quad\times\exp\left[-\int_0^{z}\mathrm{d}\lambda'\mu_{\mathrm{tot}}(\boldsymbol{r}-\hat{z}\lambda')\right]. \quad (21)\end{aligned}$$

To represent this distribution function in the spherical harmonic basis, we follow a similar treatment as in Jha *et al.* [44], which leads the spherical harmonic coefficients due to the $\mathcal{K}\mathcal{X}\Xi$ term to be

$$[W]_{lm}(\boldsymbol{r}) = \alpha\mu_{\mathrm{s}}(\boldsymbol{r})h(x,y)g^l\sqrt{\frac{2l+1}{4\pi}}\exp\left[-\int_0^{z}\mathrm{d}\lambda'\mu_{\mathrm{tot}}(\boldsymbol{r}-\hat{z}\lambda')\right]. \quad (22)$$

Using Eq. (6) to represent the above distribution function in the voxel basis, we obtain the (l,m)th spherical harmonic coefficient of the distribution function due to the $\boldsymbol{DA\xi_{\mathrm{d}}}$ term in the (i,j,k)th voxel to be

$$\begin{aligned}W_{lm}(i,j,k) &= h(x_i,y_j)\frac{g^l}{\Delta V}\sqrt{\frac{2l+1}{4\pi}}\sum_{k'=1}^{K}\frac{\mu_{\mathrm{s}}(i,j,k')}{\mu_{\mathrm{tot}}(i,j,k')}\\ &\quad\times\exp[-\mu_{\mathrm{tot}}(i,j,k')z_{k'}]\{1-\exp[-\mu_{\mathrm{tot}}(i,j,k')\Delta z)]\}, \quad (23)\end{aligned}$$

where (x_i,y_j) denote the x and y coordinates of the center of the (i,j,k)th voxel. After evaluating the $\boldsymbol{DA\xi_{\mathrm{d}}}$ term in the spherical harmonic basis, we solve the RTE using an iterative approach based on Eq. (20). In each iteration, we apply the attenuation operator, and then the scattering operator, on an effective source term. We then add the resulting distribution function to the Neumann series, and also use it as the source for the next iteration. To illustrate our procedure, we first apply the attenuation and scattering operators on $\boldsymbol{DA\xi_{\mathrm{d}}}$. We thus obtain $\boldsymbol{DADA\xi_{\mathrm{d}}}$, which is added to the Neumann series, and also used as the source term in the next iteration. We run the iterations until convergence is achieved, evaluating convergence using a criterion that we have developed [44]. After that, we apply the attenuation operator on all the computed Neumann-series terms. To compute the flux at the transmitted face of the medium due to all the terms except the $\mathcal{X}\Xi$ term, we then integrate the computed distribution function in spherical harmonic and voxel basis over space and angles. In this study, we consider a pixilated contact detector with the detector plane parallel to the transmitted face of the medium. Assuming that the detector has uniform sensitivity over a given pixel, the flux measured by the (i,j)th pixel of the detector is then computed in units of Watts as [44]

$$\Phi_{ij} = c_{\mathrm{m}}A_{\mathrm{p}}\int_{2\pi}\mathrm{d}\Omega\,\hat{z}\cdot\hat{s}\;w(x_i,y_j,H,\hat{s}), \quad (24)$$

where (x_i,y_j) denote the x and y coordinates of the center of the (i,j)th detector pixel, H is the size of the medium along the z axis so that $z=H$ corresponds to the detector plane, and A_{p} is the area of a pixel. This flux can be directly obtained from the distribution function represented in the spherical harmonic and voxel basis [44]. Finally, the flux due to the $\mathcal{X}\Xi$ term at the transmitted face in the (i,j)th voxel, which we denote by $\Phi_{ij,\mathcal{X}\Xi}$, is derived by substituting the expression for $[\mathcal{X}\Xi](\boldsymbol{r},\hat{s})$ from Eq. (19) in Eq. (24). This yields

$$\Phi_{ij,\mathcal{X}\Xi} = \alpha A_{\mathrm{p}}h(x_i,y_j)\exp\left[-\sum_{k'=1}^{K}\Delta z\mu_{\mathrm{tot}}(i,j,k')\right]. \quad (25)$$

This flux is added to the flux computed from the rest of the terms to obtain the total flux at the transmitted face. We implement this algorithm, first without using any GPU hardware, which we refer to as the central processing unit (CPU) implementation, and then on a system consisting of multiple GPU cards, which we refer to as the GPU implementation.

3. IMPLEMENTATION

A. CPU Implementation

The software to solve the Neumann-series form of the RTE in nonuniform media is developed using the C programming language on a computing system with a 2.27 GHz Intel Xeon quad core E5520 processor as the CPU running a 64 bit Linux operating system. The software reads the input-phantom specifications and follows the procedure described in Subsection 2.C to evaluate the Neumann series. However, performing the attenuation operation in this procedure is a computationally challenging and memory-intensive task. As is evident from Eq. (12), to evaluate the distribution function in a voxel after the attenuation operation, we need the contribution from all the other voxels in the medium. We also require the corresponding terms of the $\boldsymbol{A}$ matrix given by Eq. (13). Since we need the terms of the matrix $\boldsymbol{A}$ in every

iteration of the Neumann series, if the size of $\boldsymbol{A}$ is not very large, we can precompute and store these elements. For a homogeneous medium, we can exploit the shift-invariant nature of the attenuation operator, which reduces the size of the $\boldsymbol{A}$ matrix and therefore allows its storage [44,45]. However, in a heterogeneous medium, this operator is no longer shift-invariant, and storing the elements of the $\boldsymbol{A}$ matrix requires more memory than is typically available. To get around this issue, in a previous work [45], we devised a pattern-based scheme for simple heterogeneous phantoms. However, for more complicated heterogeneous phantoms, the pattern-based method is very difficult to implement. Because of the large memory requirements, it is instead more pragmatic to compute the elements of $\boldsymbol{A}$ in each iteration of the Neumann series. This implies that in a medium consisting of N voxels, since for each voxel, we have to compute the effect due to all the other voxels, even when $L = 0$, we need to perform N operations for each voxel. Thus, to compute the distribution function for all the N voxels, we must perform N^2 operations. In each of these operations, we have to compute an element of the matrix $\boldsymbol{A}$ using Eq. (13). When the number of the spherical harmonic coefficient is higher, such as when $L = L_0$, then we need to perform $N^2(L_0 + 1)^4$ operations. Furthermore, we have to perform these many operations in every iteration of the Neumann series.

To add to the computational requirement, computing every element of the matrix $\boldsymbol{A}$ requires determining the exponential attenuation factor present in Eq. (13). We refer to the negative of the exponent of this exponential as the radiological path between the voxels. To compute the radiological path, we use an improved version of the Siddon's algorithm [48]. We implement this algorithm for a 3D medium. We further modify this improved algorithm for rays completely enclosed inside the medium. This modification is required since the original Siddon's algorithm [48,49] is designed for computed-tomography applications. Thus, in the original algorithm, all the rays start from and end outside the medium. In contrast, in our algorithm, the rays start and end at the midpoints of the different voxels of the medium, and are thus completely enclosed within the medium. To briefly summarize the scheme, we first define a grid consisting of multiple planes along the x, y, and z axis, based on the voxel basis representation. To determine the voxels that a ray starting from a certain voxel passes through until it reaches the destination voxel, we trace a line between the midpoint of these voxels. Considering a parameterized form for this line, we determine the locations at which the line intersects the different planes in the grid, using which, on the fly, we determine the voxels that the ray passes through and the distance that it covers in those voxels. This is a computationally intensive set of operations.

To reduce the computational requirements of the attenuation operation, we implement a numerical approximation in our code. From the expression of the attenuation operation [Eq. (14)], it is evident that due to the exponential term in the attenuation kernel, voxels that are far away from a given voxel have very little effect on the distribution function of that voxel. Thus, the effect of the voxels that are beyond a certain threshold distance can be approximately neglected. This threshold is determined by computing the average value of the attenuation coefficient in the medium, and using this value to determine the distance $|\boldsymbol{r} - \boldsymbol{r}'| = R_{\text{thresh}}$ at which the exponential attenuation term becomes almost negligible in Eq. (11). Also, when $g = 0$, we use a simplification described in Jha *et al.* [44] that helps simplify the computational requirements considerably. To further increase the computational efficiency, we parallelize the execution of the attenuation kernel on the Intel E5520 CPU, a processor that consists of four cores and eight threads, using the open multiprocessing (OpenMP) application programming interface (API). In spite of these procedures, the execution of the Neumann-series RTE still takes considerable time, to the order of hours. We profile this code using the GNU profiler, and find that the attenuation operation takes more than 95% of the execution time for almost all phantom configurations.

Because of the excessive time taken by the attenuation operation, and the fact that the implementation of this operation can be parallelized, we studied the use of GPUs for this task.

B. GPU Implementation

In each term of the Neumann series, the scattering and attenuation operations are executed for every voxel. Therefore, for every voxel, the same code is executed, albeit on different data. Assigning the code that is executed for different voxels to different parallel processing elements can therefore increase the speed of the code significantly. This parallelization scheme, also known as data-level parallelism, can be implemented on the GPUs.

GPUs are rapidly emerging as an excellent platform to provide parallel computing solutions, due to their efficient design features, such as deeply optimized processing pipelines, hierarchical thread structures, and high memory bandwidth. With the advent of compute unified device architecture (CUDA) as an extension of the C language for general purpose GPU (GPGPU) computing, GPUs are being extensively used for scientific computing. Even in the field of simulating photon propagation in tissue, significant speedups have been achieved using GPUs. Alerstam *et al.* [50] have shown that using GPUs to perform Monte Carlo (MC) simulation of photon migration in a homogeneous medium can increase the speed by a factor of 1000. Similarly, for heterogeneous media, Fang and Boas [51] have shown that the speed of MC simulations can be increased by about 300 times by using GPUs. Huang *et al.* [52] have developed a radiative transfer model on GPUs for satellite-observed radiance. Gong *et al.* [53] and Szirmay-Kalos *et al.* [54] have also implemented the RTE on GPUs using a discrete-ordinates approach.

While GPUs are suitable for parallelizing our code, a constraint with the GPU is the limited global and on-chip memory available on these devices. The Neumann-series algorithm implementation requires significant memory. In particular, applying the attenuation operator is both computationally and memorywise intensive. As is evident from Eqs. (12) and (13), for a single computation using the attenuation operator, the code should have access to the distribution function values of all the voxels in the phantom, the attenuation coefficients of all the voxels, and finally the spherical harmonic values for all the angles and (l, m) values. This can translate to megabytes of memory requirement even in simple cases. Therefore, our code design should be optimized memorywise to exploit the benefits of parallelization. Another issue is that for an efficient GPU implementation, the code that runs on different parallel processing units should ideally require little storage

memory for its local variables. This is because the number of registers, which are shared by the different parallel processing units on the GPU, is limited, and if each thread's memory requirements are high, this would lead to a small number of threads running in parallel. Moreover, if each kernel's local memory requirements are high, then the code will be inefficient. However, the attenuation operation code requires many local variables. In the next few subsections, we describe how we handle these issues.

1. Device and Programming Tools

We implement the Neumann-series algorithm on a 2.26 GHz Intel quad core system running 64 bit Linux and consisting of four NVIDIA Tesla C2050 GPUs. These GPUs are based on the Fermi architecture, which apart from its other advantages is very efficient for double-precision computations. The Tesla C2050 is a device of compute capability 2.0. Each C2050 card consists of 14 multiprocessors, 448 CUDA cores, dedicated global memory of 3 GB, and single- and double-precision floating point performances of 1.03 Tflops and 515 Gflops, respectively [55].

To implement the RTE on our GPU system, we use POSIX thread (pthread) libraries [56] and NVIDIA CUDA [57]. While pthread libraries are used to split the programming task onto the multiple GPUs, CUDA is used to implement our algorithm on the individual GPUs. Pthread libraries are a computationally efficient, standards-based thread API for C/C++ that allows the programmer to spawn a new concurrent process flow. CUDA is a general purpose parallel computing architecture, with a parallel programming model and instruction set architecture to harness the computing power of NVIDIA GPUs for general purpose programming [57,58]. A CUDA application consists of two parts: a serial program that runs on the CPU and a parallel program, referred to as the kernel, that runs on the GPU [59]. The kernel executes in parallel as a set of threads. The threads are grouped into blocks, where the threads within a block can concurrently execute and cooperate among themselves through barrier synchronization and shared memory access to a memory space that is private to that block. A grid is one level higher in the hierarchy and is a set of blocks that can be executed independently and in parallel. When invoking a thread using CUDA, we can specify the number of threads per block and the number of blocks. The memory requirements of the kernel often restrict the number of threads that are executed together.

CUDA-enabled devices also adhere to a common memory management scheme. Each GPU device possesses a global random access memory (RAM), called the global memory, which can be accessed by all the multiprocessors on the GPU. In addition, each multiprocessor has its own on-chip memory that is accessible only to the cores on that multiprocessor. The on-chip memory of the GPU consists of registers, shared memory, and the L1 cache [60]. The shared memory is shared among the threads in a block. There are also two additional read-only memory spaces accessible by all threads: the constant and texture memory spaces [55], which are cached onto the on-chip memory of the GPU. The constant memory is efficient to use if different threads in a multiprocessor access the same memory location. As for texture memory, for applications that do not benefit from using texture memory, its usage can be counterproductive on devices with compute capability 2.x, since global memory loads are cached in the L1 cache, and the L1 cache has higher bandwidth than texture cache [61].

2. Parallelization Scheme and Software Design

The flowchart of the software implementation is shown in Fig. 2. The basic procedure is similar to the procedure outlined in Subsection 2.C, but we incorporate many computational and memory optimizations. The distribution function due to $\boldsymbol{DA\xi_d}$ terms is computed on the CPU using Eq. (23). Similarly, the spherical harmonic coefficient values $Y_{l,m}(\hat{s})$ for different angles are computed on the CPU. The distribution function due to the $\boldsymbol{DA\xi_d}$ term and the spherical harmonic coefficient values, due to their large memory requirement, are transferred to the global memory of the GPU using CUDA memory allocation techniques. Since the constant memory on the GPU is very efficient for storing data that all the threads access simultaneously, we store such data in our software, such as the geometric configuration and the properties of the phantom that are the same for all the voxels, in the constant memory. Storing the scattering and absorption coefficients for each voxel requires considerable memory. To reduce this memory requirement, we use the fact that although the number of voxels in the phantom is high, the number of tissue types, i.e., tissues with different absorption or scattering coefficients, is quite lower in most cases. For more efficient memory use, we instead store a *tissue map*, which is a data structure that stores the tissue type of the different voxels. In this scheme, only 1 byte storage space is required for each voxel. The absorption and scattering coefficients of the different tissue types are stored in the constant memory, while the tissue map is stored in the global memory. Also the terms of matrix $\boldsymbol{A_D}$, i.e., $A_{lm,l'm'}(i,j,k)$ are the same for voxels with the same tissue type, as is evident from Eqs. (15) and (16). Therefore instead of evaluating and storing these terms for each voxel, we instead just compute it for the different tissue types. We store these terms in the constant memory of the GPU since

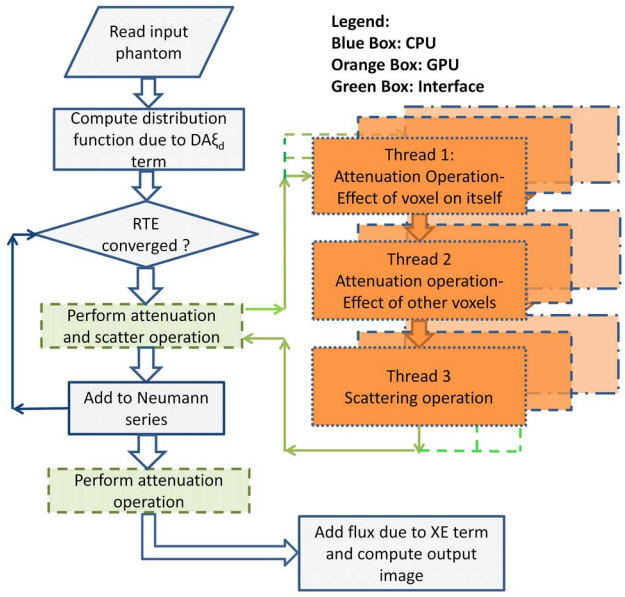

Fig. 2. (Color online) Flowchart of our algorithm implementation on the GPU. Blue boxes (filled with dotted pattern) denote the code running on the CPU, orange boxes (filled with solid pattern) denote code running on the GPU, and green boxes (filled with vertical-line pattern) are the interface routines between CPU and GPU.

often, many threads in a block will access the same terms. This is because voxels with similar tissue type are generally close to each other, and in our parallelization scheme, voxels in the same block are close to each other. The Neumann-series iteration procedure is then executed, which begins with applying the attenuation operator to the distribution function.

As mentioned in Subsection 3.A, the attenuation operation, given by Eq. (13), takes the most computation time. Therefore our main objective is to parallelize the attenuation-kernel execution. To achieve this objective, we divide the voxelized phantom into four layers along the z dimension, and each layer is allotted to one of the four GPUs using pthreads. On the GPU, the allotted layer is further split along the z dimension into sublayers. Each sublayer is assigned to one grid, and the grid is further divided into blocks. The number of threads in a block is user-configurable. The basic idea can be to assign the attenuation operation code for one voxel to one thread. However, the attenuation operation code performs a number of computations including computing the radiological path. Thus, this code requires many local variables, which can lead to inefficient implementation, as described earlier. To solve this issue, the attenuation operation code for each voxel is split into two separate kernels that execute one after the other. Splitting the attenuation operation code into two kernels results in each thread having a smaller number of register requirements. The first kernel evaluates the effect that the voxel has on itself using Eq. (16). The second kernel evaluates the effect that the other voxels have on the distribution function in a given voxel by computing the elements of the attenuation kernel $A_{lm}(i,j,k,i',j',k')$ using Eq. (13), followed by evaluating the attenuation operation using Eq. (12). In this kernel, the radiological path between two voxels is also computed using the Siddons algorithm. The parallelization scheme is illustrated in Fig. 3.

The attenuation kernel is evaluated on the GPUs with the parallelization scheme mentioned above. The scattering operation, given by Eq. (23), is then performed on the GPU by allotting one thread to each voxel. The computed distribution function as a result of the attenuation and the scattering operation is then transferred to the host memory from the global memories of the four GPUs. This distribution function is added to the Neumann series on the host memory and is also used as the source for the next iteration. The Neumann-series iterations are performed until convergence is achieved, following which, the attenuation operation is applied for one last time. Using the computed distribution function in spherical harmonics, the transmitted flux at the exit face is computed. The flux due to the $\mathcal{X}\Xi$ term, given by Eq. (25), is added to the computed transmitted flux at the exit face. Finally, the computed output image is displayed.

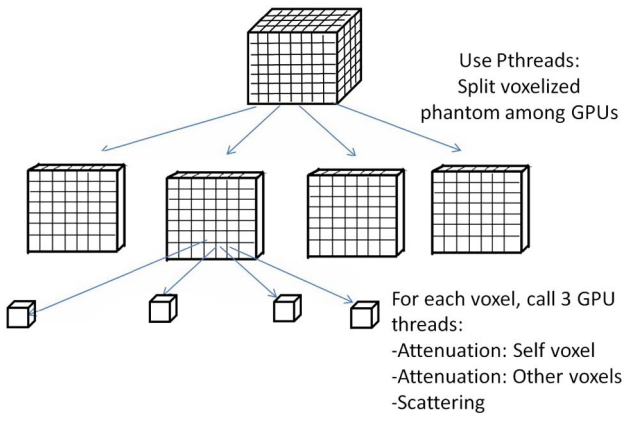

Fig. 3. (Color online) Summary of the parallelization scheme to implement the RTE.

We experiment with various mechanisms such as coalesced global memory access, on-chip shared memory use, thread block synchronization, constant memory use, and minimum-memory assignment for the different local variables to achieve efficient GPU implementation of the algorithm. The various local variables that are required in the kernel code are stored in as little memory as possible, which helps minimize GPU register usage. We try to perform less global memory access, but once the data from global memory is accessed, it is stored in the L1 cache for a small duration. We make use of this caching mechanism also while implementing the kernel. Also, threads that are part of the same block correspond to voxels that are near each other in a 3D sense. Therefore, the data required by the threads along one of these dimensions lie in memory locations that are close to each other, so we can take advantage of coalesced global memory access to a certain extent. We were very keen to use the on-chip shared memory and experimented various configurations for its usage. However, the use of shared memory results in computationally inefficient code, due to the limited amount of shared memory and the high memory requirements of the Neumann-series algorithm. Although threads in a block share data such as the source distribution function, in most of the computations, the memory required by such data exceeds the available shared memory. Therefore, we instead configure the GPU before each kernel call so that most of the on-chip memory is allotted to the L1 cache instead of the shared memory. Also, since we did not find any specific advantage in using the texture memory, we prefer the global memory to texture memory for our application. Because of the multiple kinds of optimization mechanisms [60], each with its own tradeoffs, we perform a number of experiments with different mechanisms to achieve a computationally efficient code. To help with the optimization of the code, we use various methods such as inspecting the assembly level file, using the CUDA Occupancy Calculator, and compiling with specific flags to study the usage of different memories [60]. We also develop a complex-arithmetic library of functions for the GPU that performs some specific complex number computations that our code requires.

4. EXPERIMENTS AND RESULTS

In this section, we compare results obtained using our Neumann-series approach with the results obtained using an MC approach to model photon propagation [62]. It has been shown that the diffusion approximation is valid in optically thick media and in cases where scattering dominates absorption [15]. In our experiments, we study different cases where the standard diffusion approximation breaks down, such as media illuminated by collimated light sources [63,64], optically thin media [65], media with low-scattering voidlike regions [32], and media where the absorption coefficient is similar to the scattering coefficient [20,24].

The simulation setup is shown in Fig. 1, where the scattering medium is a 3D slab, characterized by a reduced-scattering coefficient $\mu_s' = \mu_s(1-g)$. The collimated source emits NIR light with a transverse circular profile. The beam is incident on the center of the entrance face of the scattering medium.

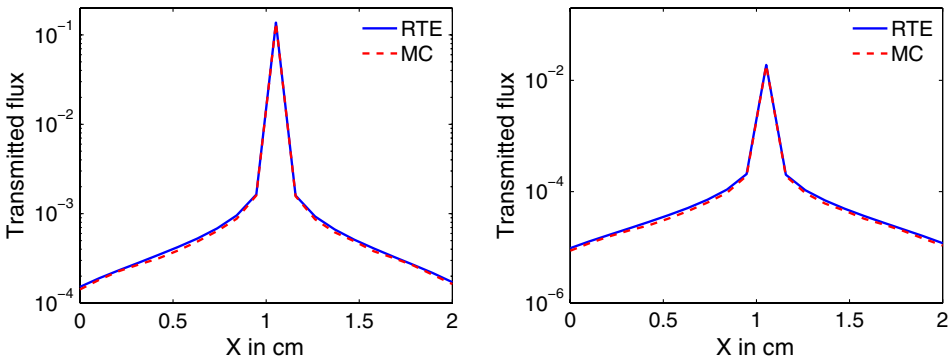

Fig. 4. (Color online) Neumann-series and MC transmittance outputs in a homogeneous low-scattering medium ($\mu_s' = 1$ cm^{-1}) with (a) low-absorption coefficient ($\mu_a = 0.01$ cm^{-1}) and (b) high-absorption coefficient ($\mu_a = 1$ cm^{-1}).

The boundary-transmittance measurements are made at the exit face of the scattering medium. All the experiments are carried out for nonreentry boundary conditions, so that the refractive index of the medium is equal to 1. In the Neumann-series method, for all the experiments, we choose the number of spherical harmonic coefficients to be such that $L = 3$. In the GPU simulation, the number of threads per block is kept as $10 \times 10 \times 5$ along the x, y, and z axis, respectively. The MC simulations are performed using the tMCimg software [62]. The number of simulated photons in each MC simulation is 1×10^7. The simulation results shown in all the cases denote the linear profile of the transmitted image. The linear profile is taken along the line passing through the center of the transmitted image.

A. Homogeneous Medium

We first validate our code on homogeneous scattering media. In our first experiment, the homogeneous medium has optical properties $\mu_a = 0.01$ cm^{-1}, $\mu_s' = 1$ cm^{-1}, and $g = 0$. The medium has a size of $2 \times 2 \times 2$ cm^3 and is discretized into $20 \times 20 \times 20$ voxels. We know that the standard diffusion approximation does not yield good results when the medium is optically thin [65], although it can be used with several modifications [66]. The Neumann-series method accurately models this situation, as shown in Fig. 4(a).

Another issue with the diffusion approximation is that it breaks down when $\mu_a \sim \mu_s'$ [24]. To verify the performance of the Neumann-series method in this regime, we perform another experiment in which the absorption and scattering coefficients in an isotropic medium are both equal to 1 cm^{-1}. As the results in Fig. 4(b) show, using the Neumann-series approach, we can model light propagation accurately in this scenario.

We next consider a midscattering medium with $\mu_s' = 4$ cm^{-1}, of size $1 \times 1 \times 1$ cm^3 with $\mu_a = 0.01$ cm^{-1} and $g = 0$. Since this medium has a higher scattering coefficient and thus a smaller mean free path for the photon, for accurate spatial discretization of the distribution function, we need to have smaller-sized voxels. From the analysis performed in our uniform-media paper [44], we know that the voxel size should be 0.1 times the mean free path of the photon to obtain an accurate output using the Neumann-series approach. The medium is thus divided into $40 \times 40 \times 40$ voxels. The results comparing the MC and the Neumann-series outputs are shown in Fig. 5(a), and we see a very good match between the two. In another experiment, we increase the absorption coefficient of this medium by 100 times, so that $\mu_a = 1$ cm^{-1}, which leads to μ_a being close to μ_s'. The Neumann-series approach is accurate in this scenario, as confirmed by the result shown in Fig. 5(b).

B. Heterogeneous Medium

To validate the performance of the Neumann-series approach for a nonuniform midscattering medium, we consider a $1 \times 1 \times 1$ cm^3 sized phantom with different kinds of heterogeneity models as shown in Fig. 6. The first experiment is with a phantom that has three different tissue types scattered randomly, as

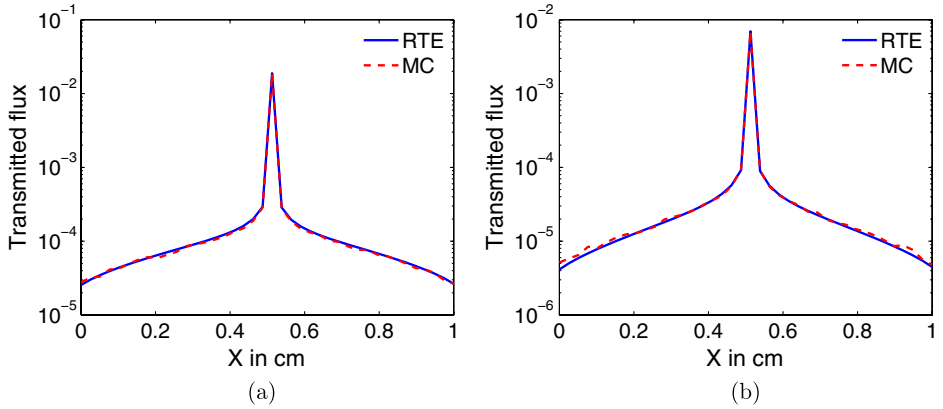

Fig. 5. (Color online) Neumann-series and MC transmittance outputs in a midscattering medium ($\mu_s' = 4$ cm^{-1}) with (a) low-absorption coefficient ($\mu_a = 0.01$ cm^{-1}) and (b) high-absorption coefficient ($\mu_a = 1$ cm^{-1}).

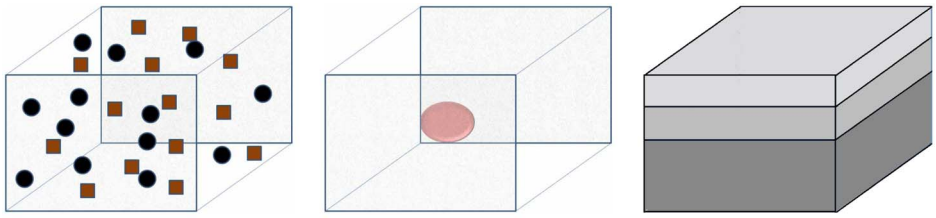

Fig. 6. (Color online) Different heterogeneous tissue geometries: (a) three different tissue types scattered randomly, (b) spherical inclusion in the center of the medium, and (c) medium consisting of three layers.

shown in Fig. 6(a). The three tissue types have their reduced-scattering and absorption coefficients, (μ_s', μ_a), equal to $(4\,\mathrm{cm}^{-1}, 0.01\,\mathrm{cm}^{-1})$, $(4\,\mathrm{cm}^{-1}, 1\,\mathrm{cm}^{-1})$, and $(2\,\mathrm{cm}^{-1}, 0.01\,\mathrm{cm}^{-1})$, respectively. The anisotropy factor g is 0 for all the tissue types. As shown in Fig. 7(a), for this particular case, the Neumann-series results are in agreement with the MC results.

In the next experiment, we place a high-absorption spherical inclusion ($\mu_s' = 4\ \mathrm{cm}^{-1}$, $\mu_a = 1\ \mathrm{cm}^{-1}$) of diameter 0.67 cm at the center of an otherwise homogeneous isotropic medium with $\mu_s' = 4\ \mathrm{cm}^{-1}$ and $\mu_a = 0.01\ \mathrm{cm}^{-1}$, as shown in Fig. 6(b). As we observe from the output shown in Fig. 7(b), the Neumann-series output matches well with the MC output. To study the performance of the Neumann-series approach when a very-low-scattering inclusion is present in the medium, we consider a low-scattering spherical inclusion with diameter 0.67 cm and optical properties of $\mu_s' = 0.1\ \mathrm{cm}^{-1}$, $\mu_a = 0.001\ \mathrm{cm}^{-1}$, and $g = 0$, placed in an isotropic medium with $\mu_s' = 4\ \mathrm{cm}^{-1}$ and $\mu_a = 0.01\ \mathrm{cm}^{-1}$. The geometry of the scattering medium is shown in Fig. 6(b). The reduced-scattering and absorption coefficients of the low-scattering region mimic the cerebro-spinal-fluid [24]. We find that the Neumann-series results match with the MC results, as shown in Fig. 7(c).

Finally, we consider a layered phantom model as shown in Fig. 6(c), where each of the layers has different optical properties. The three layers have thicknesses of 2.5, 2.5, and 5 mm, respectively. The reduced-scattering and absorption coefficients, (μ_s', μ_a), of the media in the three layers from top to bottom are $(2\ \mathrm{cm}^{-1}, 0.001\ \mathrm{cm}^{-1})$, $(3\ \mathrm{cm}^{-1}, 0.001\ \mathrm{cm}^{-1})$, and $(4\ \mathrm{cm}^{-1}, 0.01\ \mathrm{cm}^{-1})$, respectively. The Neumann-series and MC results match well for this layered phantom, as shown in Fig. 7(d).

A similar set of experiments is repeated with a low-scattering heterogeneous medium of size $2 \times 2 \times 2\ \mathrm{cm}^3$. The

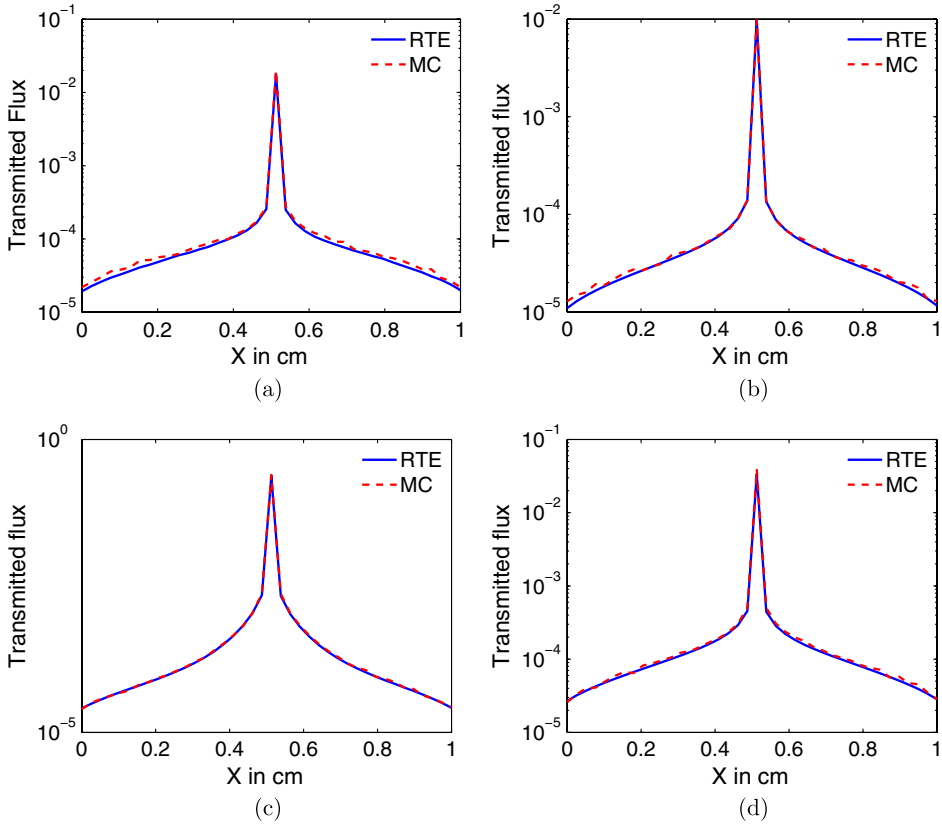

Fig. 7. (Color online) Neumann-series and MC transmittance outputs in a midscattering heterogeneous medium with (a) three different tissue types scattered randomly, (b) a high-absorption spherical inclusion at the center of the medium, (c) a very-low-scattering inclusion at the center of the medium, and (d) layered medium.

first experiment has three different isotropic tissue types scattered randomly [Fig. 6(a)], with reduced-scattering and absorption coefficients (μ_s', μ_a) equal to $(1\ \text{cm}^{-1}, 0.01\ \text{cm}^{-1})$, $(1\ \text{cm}^{-1}, 1\ \text{cm}^{-1})$, and $(0.5\ \text{cm}^{-1}, 0.01\ \text{cm}^{-1})$, respectively. The second experiment is with a phantom consisting of a high-absorption spherical inclusion ($\mu_s' = 1\ \text{cm}^{-1}$, $\mu_a = 1\ \text{cm}^{-1}$, $g = 0$) of diameter 0.67 cm placed at the center of a homogeneous isotropic medium with $\mu_s' = 1\ \text{cm}^{-1}$ and $\mu_a = 0.01\ \text{cm}^{-1}$ (Fig. 6(b)]. In the third experiment, we have a isotropic medium with $\mu_s' = 1\ \text{cm}^{-1}$, $\mu_a = 0.01\ \text{cm}^{-1}$, that has a 0.67 cm diameter low-scattering spherical inclusion at the center, as shown in Fig. 6(b). The optical properties of the low-scattering inclusion are $\mu_s' = 0.1\ \text{cm}^{-1}$, $\mu_a = 0.001\ \text{cm}^{-1}$, and $g = 0$. The fourth experiment is with an isotropic medium consisting of three layers [Fig. 6(c)] of thickness 5, 5, and 10 mm, respectively, and the reduced-scattering and absorption coefficients (μ_s', μ_a) equal to $(1\ \text{cm}^{-1}, 0.01\ \text{cm}^{-1})$, $(0.5\ \text{cm}^{-1}, 0.1\ \text{cm}^{-1})$, and $(0.5\text{cm}^{-1}, 0.01\text{cm}^{-1})$, respectively. The results for these four types of low-scattering heterogeneous media are shown in Figs. 8(a)–8(d). We see that in all the cases, the Neumann-series output is in agreement with the MC output.

C. Computational Requirements

In this subsection, we first show the speedup obtained with the GPU-based implementation of the code. As mentioned earlier, the CPU-version of the code is executed on an Intel quad core E5520 2.65 GHz processor, which has four cores and eight threads, so the execution on this system is not really single-threaded. To evaluate the timing improvements, we perform multiple experiments, in which we vary the number of voxels in the medium, the scattering and absorption coefficients, and the number of spherical harmonics. The results are shown in Table 1; in the second and third columns, we show the time required by the CPU and GPU implementations for a single execution of the attenuation kernel. The final column shows the corresponding speedup obtained with the GPU implementation. In the table, nVox denotes the number of voxels in the phantom. We find that even in simple cases, there is up to two orders of speed improvement when using the GPU as compared to a state-of-the-art but non-GPU hardware.

The GPU implementation leads to good computational efficiency for the code. For example, all the simulations executed for the low-scattering isotropic medium in Subsections 4.A and 4.B require less than a minute. However, the computation time increases significantly for medium and high-scattering media, since such media require a higher number of voxels for accurate discretization, and the execution of a larger number of Neumann-series terms. Also, the computation time increases significantly as the number of spherical harmonic coefficients required to represent the distribution function increases. Theoretically, the timing requirements increase almost linearly with the value of $(L+1)^2$. For anisotropic-scattering media ($g \neq 0$), the number of spherical harmonics required to represent the distribution function accurately is high. Therefore, for such media, the computational time required will be considerably higher.

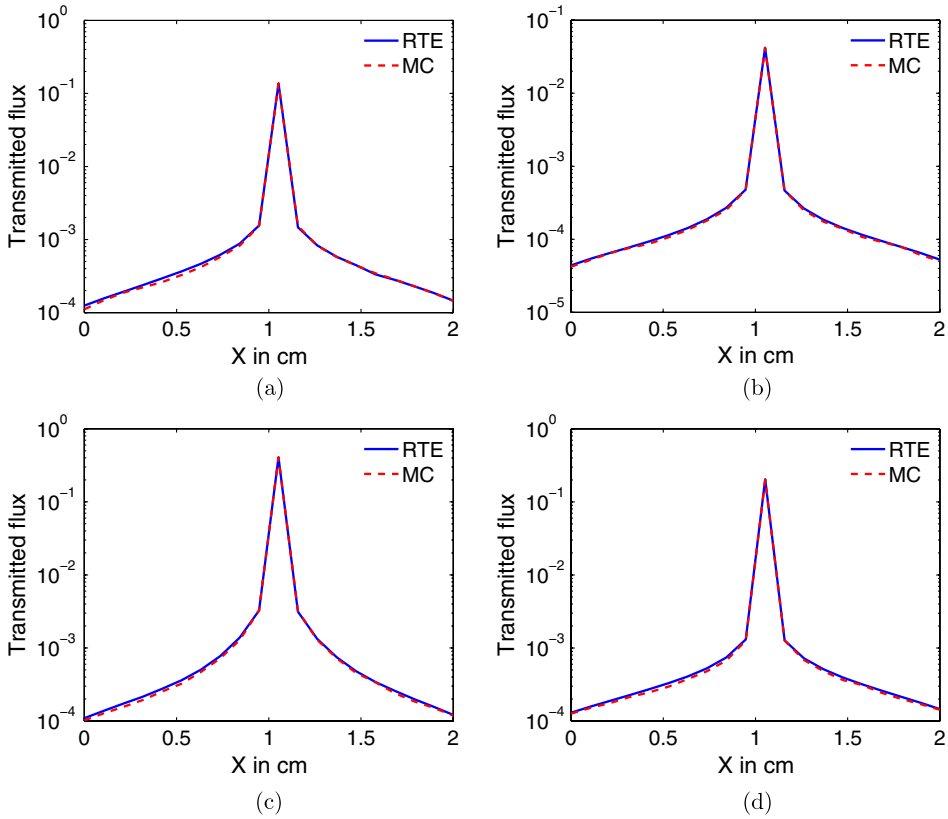

Fig. 8. (Color online) The Neumann-series and MC transmittance outputs in a low-scattering heterogeneous medium with (a) three different tissue types scattered randomly, (b) a high-absorption spherical inclusion at the center of the medium, (c) a very-low-scattering inclusion at the center of the medium, and (d) layered medium.

Table 1. Speedup with GPU Implementation

Configuration	CPU (in sec)	GPU (in sec)	Speedup
$n\text{Vox} = 20^3$, $L = 0$	50	1.3	45X
$n\text{Vox} = 20^3$, $L = 1$	237	3.3	72X
$n\text{Vox} = 20^3$, $L = 2$	674	7.2	94X
$n\text{Vox} = 20^3$, $L = 3$	1578	14.2	111X
$n\text{Vox} = 40^3$, $L = 0$	5614	35.4	159X
$n\text{Vox} = 40^3$, $L = 1$	22764	134.2	170X

5. DISCUSSION AND CONCLUSIONS

In this paper, we have presented the mathematical methods and the software to implement the integral form of the RTE for modeling light propagation in nonuniform media for a diffuse-optical-imaging setup. Using the Neumann-series approach, the RTE is solved for a completely heterogeneous 3D media. The method is parallelizable and, with the implementation on the GPU, the computational efficiency of the method is also very good, and up to 2 orders of magnitude higher than a non-GPU implementation. As part of this software, we have also implemented the Siddon's algorithm on the GPU. We demonstrate the accuracy of the Neumann-series method in simulating light propagation in small-geometry heterogeneous phantoms with different kinds of models.

We observe that the Neumann-series RTE algorithm simulates photon propagation accurately in scenarios where the diffusion approximation is inaccurate. For example, it can handle pencil-beam-like sources as the emission source, unlike the diffusion approximation [63,64]. Also, the Neumann-series approach can simulate photon propagation accurately when the absorption coefficient is close to the scattering coefficient. The Neumann-series method works well in these scenarios since it handles the effect due to the absorption event accurately. This is possible due to the mathematical formalism of the Neumann series, and our implementation procedure, where we evaluate the effect of the $\mathcal{X}\Xi$ term separately and in the normal $(\boldsymbol{r}, \hat{s})$ basis instead of the spherical harmonic basis. We also see that the Neumann-series method can simulate photon propagation even when the scattering and absorption coefficients are extremely small, which is true in voidlike spaces such as ventricles and the subarachnoid space. Our method can also simulate light propagation through optically thin regions, which can be useful in many studies [65,67]. We also show that the Neumann-series approach can simulate light propagation in thin layered media, where the diffusion approximation is not applicable [68]. Therefore, there are many advantages to the Neumann-series implementation as compared to the diffusion approximation.

While the Neumann-series method is completely general and can be used to model light propagation in any media, it is computationally very intensive for optically thick media. We have observed that the Neumann-series approach is practical to use for media that have the product of their scattering coefficient and length, i.e., $\mu_s H$ less than 4. However, for media with greater optical thickness, increase in computational capacity will lead to the method being more practical. In contrast, the diffusion-approximation is very effective for optically thick media. Thus, the Neumann-series method can be used to complement the diffusion-approximation-based methods in many cases. The usage of the Neumann-series method for modeling light propagation through optically thin media such as skin, and in imaging techniques like window chamber imaging [69], can also be explored. Also, with the increase in computing technology, the Neumann-series approach can be implemented in lesser time in future. NVIDIA plans to release the Maxwell generation of GPU architecture in 2013, which is predicted to perform about 16 times faster than the current GPU architectures [70]. Similarly, there are significant advances occurring in the fields of multiprocessor, multicore, and cluster computing [71,72], all of which will lead to higher computational capacity. A single-equation framework to simulate light propagation can be advantageous when the computational power increases.

In the proposed Neumann-series implementation, we have two variable simulation parameters, namely the number of spherical harmonic coefficients, and the number of voxels used to discretize the medium. To determine an appropriate value for these parameters, we execute the Neumann-series algorithm with increasing values for these parameters. We have verified the convergence of the Neumann-series approach to the correct solution as the value of these parameters is increased, in our uniform-media paper [44]. Therefore, the value at which the output of the Neumann-series approach converges approximately gives us a good value for these simulation parameters. We have also added another feature in our software that can help determine suitable values for these parameters. This feature computes the number of photons that are output from all the faces of the medium. For a nonabsorbing medium, by the principle of photon conversation, the number of photons that enter and exit the medium should be equal. Let us consider a medium that has its scattering coefficient much greater than the absorption coefficient. For this medium, we execute the Neumann-series simulations with the absorption coefficient set to zero and find out those values of spherical harmonic coefficients and the number of voxels at which the number of output photons is approximately equal to the number of input photons. The values thus determined give us an approximate number for the two simulation parameters for this media. Since scattering is often the dominant photon-interaction mechanism at optical wavelengths, this feature can be used in many scenarios.

We would also like to discuss the performance of the Neumann-series method with MC-based methods for simulating photon propagation. We find that for low-scattering media, the times required by the GPU implementation of the Neumann-series method and CPU-based MC simulation are similar. However, for high-scattering media, the time required by the CPU-based MC method is smaller. Based on these comparisons, we believe that the GPU-based MC method would outperform the Neumann-series method in terms of computational time requirements. However, the MC method is a stochastic approach to model photon simulation, which can lead to uncertainty in the computed output. The Neumann-series method is an analytic deterministic approach, and thus the output computed using this method is exact and does not suffer from issues due to randomness. The Neumann-series method can have errors due to finite discretization, but these errors can be reduced by increasing the number of voxels or the number of spherical harmonics.

Another advantage of the Neumann-series method compared to the MC method is with regard to the reconstruction task in DOT. As we mentioned earlier, the forward modeling

of photon propagation in DOT is mainly required to perform the reconstruction task, i.e., to determine the optical properties of the tissue. The image reconstruction approaches are often based on some kind of gradient-descent mechanism, where the gradient of some distance measure between the experimentally determined output image and the simulated output with respect to the optical properties of the medium are evaluated [4]. The value of this gradient is then used to update the optical properties of the medium. Therefore, it is essential in these gradient-descent schemes that the gradient be computed accurately. In a MC-based scheme, this gradient can be computed only as a numerical approximation; i.e., we evaluate the output image at two nearby values of the optical properties of the tissue, compute the distance between these output images, and divide it by the difference between the optical properties of the tissue. The computed gradient can also be erroneous due to the uncertainty in the output obtained using MC-based schemes. In contrast, the Neumann-series approach, due to its analytic nature, provides us with an avenue to evaluate the gradient analytically. We are currently investigating in this direction and plan to discuss such an approach in a future publication. Thus, there are some advantages of the Neumann-series method as compared to the MC-based method, in spite of the high computational requirements of the former.

We would like to highlight that in the Neumann-series RTE formulation, the number of terms in the series is equal to the number of scattering events. Therefore, our method can be used to separate the measured output light in an experiment into components with a different number of scattering events, as has previously been attempted [73,74]. Also, using the Neumann-series method, we can separate the scattered light from the measured light distribution in a DOT imaging system. This information can be used to design reconstruction algorithms to obtain optical properties of the media. Another important application for the Neumann-series method is in the field of computer graphics, where the suggested software can be used either for rendering or to estimate the scattering properties of the media [67].

A limitation of the suggested method is the zero-reflection boundary conditions. While this condition can be partially met in some experimental setups with highly absorptive boundaries, in a tissue, it is unlikely that no reflection will occur at the surfaces. Based on our initial studies, incorporating the boundary conditions will require a fundamental change in the Neumann-series form of the RTE and we are currently investigating this. Another improvement in the suggested approach is to use block structured grids that are adaptively discretized, similar to the approach suggested by Montejo *et al.* [29]. This can also improve the computational efficiency of the method.

To summarize, we have developed a software to propagate light propagation through completely nonuniform 3D media using the Neumann-series form of the RTE. The method is completely general and gives very accurate results in cases where the diffusion approximation breaks down, but is computationally intensive for optically thick media. With further mathematical and algorithmic research in optimizing the Neumann-series algorithm and with the advances in parallel computing hardware, we believe that the Neumann-series approach can produce accurate results for many types of media in reasonable times. With the advances in technology, an interdisciplinary research in this field by scientists in the fields of physics, mathematics, computer science, and medicine can lead to important advances and has exciting prospects.

ACKNOWLEDGMENTS

This work was supported by Canon U.S.A., Inc., the National Institute of Biomedical Imaging and Bioengineering of the National Institute of Health under grants RC1-EB010974, R37-EB000803, and P41-EB002035, and the Society of Nuclear Medicine Student Fellowship Award. AKJ is partially funded by the Technology Research Initiative Fund (TRIF) Imaging Fellowship. The authors would also like to thank Dr. Kyle J. Myers and Mr. Takahiro Masumura for reviewing the draft of this paper, Mr. Peter Bailey for helpful discussions, and the anonymous reviewers for their comments. AKJ would also like to thank the National Science Foundation for financial support to attend the Pan-American Advanced Studies Institute on Frontiers in Imaging Science, Bogota, Colombia, which provided a platform to present and discuss this work.

REFERENCES

1. A. P. Gibson, J. C. Hebden, and S. R. Arridge, "Recent advances in diffuse optical imaging," Phys. Med. Biol. **50**, R1–R43 (2005).
2. D. A. Boas, D. H. Brooks, E. L. Miller, C. A. DiMarzio, M. Kilmer, R. J. Gaudette, and Q. Zhang, "Imaging the body with diffuse optical tomography," IEEE Signal Process. Mag. **18**, 57–75 (2001).
3. A. Gibson and H. Dehghani, "Diffuse optical imaging," Phil. Trans. R. Soc. A **367**, 3055–3072 (2009).
4. H. Dehghani, S. Srinivasan, B. W. Pogue, and A. Gibson, "Numerical modelling and image reconstruction in diffuse optical tomography," Phil. Trans. R. Soc. A **367**, 3073–3093 (2009).
5. H. Dehghani, B. W. Pogue, S. P. Poplack, and K. D. Paulsen, "Multiwavelength three-dimensional near-infrared tomography of the breast: initial simulation, phantom, and clinical results," Appl. Opt. **42**, 135–146 (2003).
6. S. Srinivasan, B. W. Pogue, S. Jiang, H. Dehghani, C. Kogel, S. Soho, J. J. Gibson, T. D. Tosteson, S. P. Poplack, and K. D. Paulsen, "In vivo hemoglobin and water concentrations, oxygen saturation, and scattering estimates from near-infrared breast tomography using spectral reconstruction," Acad. Radiol. **13**, 195–202 (2006).
7. T. Austin, A. P. Gibson, G. Branco, R. M. Yusof, S. R. Arridge, J. H. Meek, J. S. Wyatt, D. T. Delpy, and J. C. Hebden, "Three dimensional optical imaging of blood volume and oxygenation in the neonatal brain," NeuroImage **31**, 1426–1433 (2006).
8. B. W. Zeff, B. R. White, H. Dehghani, B. L. Schlaggar, and J. P. Culver, "Retinotopic mapping of adult human visual cortex with high-density diffuse optical tomography," Proc. Natl. Acad. Sci. USA **104**, 12169–12174 (2007).
9. A. H. Hielscher, A. D. Klose, A. K. Scheel, B. Moa-Anderson, M. Backhaus, U. Netz, and J. Beuthan, "Sagittal laser optical tomography for imaging of rheumatoid finger joints," Phys. Med. Biol. **49**, 1147–1163 (2004).
10. A. H. Hielscher, "Optical tomographic imaging of small animals," Curr. Opin. Biotechnol. **16**, 79–88 (2005).
11. T. Tarvainen, M. Vauhkonen, v. Kolehmainen, J. P. Kaipio, and S. R. Arridge, "Utilizing the radiative transfer equation in optical tomography," Piers Online **4**, 655–661 (2008).
12. A. Yodh and B. Chance, "Spectroscopy and imaging with diffusing light," Phys. Today **48**, 34–40 (1995).
13. A. P. Schweiger, M. Gibson, and S. R. Arridge, "Computational aspects of diffuse optical tomography," IEEE Comput. Sci. Eng., **5**, 33–41 (2003).
14. S. R. Arridge, M. Schweiger, M. Hiraoka, and D. T. Delpy, "A finite element approach for modeling photon transport in tissue," Med. Phys. **20**, 299–309 (1993).
15. H. Dehghani, B. Brooksby, K. Vishwanath, B. W. Pogue, and K. D. Paulsen, "The effects of internal refractive index variation

in near-infrared optical tomography: a finite element modelling approach," Phys. Med. Biol. **48**, 2713–2727 (2003).
16. M. Schweiger and S. R. Arridge, "The finite-element method for the propagation of light in scattering media: frequency domain case," Med. Phys. **24**, 895–902 (1997).
17. F. Gao, H. Niu, H. Zhao, and H. Zhang, "The forward and inverse models in time-resolved optical tomography imaging and their finite-element method solutions," Image Vis. Comput. **16**, 703–712 (1998).
18. A. D. Zacharopoulos, S. R. Arridge, O. Dorn, V. Kolehmainen, and J. Sikora, "Three-dimensional reconstruction of shape and piecewise constant region values for optical tomography using spherical harmonic parametrization and a boundary element method," Inverse Probl. **22**, 1509–1532 (2006).
19. S. Srinivasan, B. W. Pogue, C. Carpenter, P. K. Yalavarthy, and K. Paulsen, "A boundary element approach for image-guided near-infrared absorption and scatter estimation," Med. Phys. **34**, 4545–4557 (2007).
20. A. Klose and E. Larsen, "Light transport in biological tissue based on the simplified spherical harmonics equations," J. Comput. Phys. **220**, 441–470 (2006).
21. A. H. Hielscher, R. E. Alcouffe, and R. L. Barbour, "Comparison of finite-difference transport and diffusion calculations for photon migration in homogeneous and heterogeneous tissues," Phys. Med. Biol. **43**, 1285–1302 (1998).
22. E. D. Aydin, C. R. de Oliveira, and A. J. Goddard, "A comparison between transport and diffusion calculations using a finite element-spherical harmonics radiation transport method," Med. Phys. **29**, 2013–2023 (2002).
23. E. D. Aydin, "Three-dimensional photon migration through void-like regions and channels," Appl. Opt. **46**, 8272–8277 (2007).
24. A. H. Hielscher and R. E. Alcouffe, "Discrete-ordinate transport simulations of light propagation in highly forward scattering heterogenous media," in *Advances in Optical Imaging and Photon Migration* (Optical Society of America, 1998), paper ATuC2.
25. M. Chu, K. Vishwanath, A. D. Klose, and H. Dehghani, "Light transport in biological tissue using three-dimensional frequency-domain simplified spherical harmonics equations," Phys. Med. Biol. **54**, 2493–2509 (2009).
26. M. L. Adams and E. W. Larsen, "Fast iterative methods for discrete ordinates particle transport calculations," Prog. Nucl. Energy **40**, 3–159 (2002).
27. J. K. Fletcher, "A solution of the neutron transport equation using spherical harmonics," J. Phys. A: Math. Gen. **16**, 2827–2835 (1983).
28. K. Kobayashi, H. Oigawa, and H. Yamagata, "The spherical harmonics method for the multigroup transport equation in *x-y* geometry," Ann. Nucl. Energy **13**, 663–678 (1986).
29. L. D. Montejo, A. D. Klose, and A. H. Hielscher, "Implementation of the equation of radiative transfer on block-structured grids for modeling light propagation in tissue," Biomed. Opt. Express **1**, 861–878 (2010).
30. K. Ren, G. S. Abdoulaev, G. Bal, and A. H. Hielscher, "Algorithm for solving the equation of radiative transfer in the frequency domain," Opt. Lett. **29**, 578–580 (2004).
31. S. Wright, M. Schweiger, and S. Arridge, "Reconstruction in optical tomography using the PN approximations," Meas. Sci. Technol. **18**, 79–86 (2007).
32. E. Aydin, C. de Oliveira, and A. Goddard, "A finite element-spherical harmonics radiation transport model for photon migration in turbid media," J. Quant. Spectrosc. Radiat. Transfer **84**, 247–260 (2004).
33. R. Wells, A. Celler, and R. Harrop, "Analytical calculation of photon distributions in spect projections," IEEE Trans. Nucl. Sci. **45**, 3202–3214 (1998).
34. B. F. Hutton, I. Buvat, and F. J. Beekman, "Review and current status of SPECT scatter correction," Phys. Med. Biol. **56**, R85–R112 (2011).
35. M. A. King, S. J. Glick, P. H. Pretorius, R. G. Wells, H. C. Gifford, and M. V. Narayanan, *Emission Tomography: The Fundamentals of PET and SPECT* (Academic, 2004).
36. W.-f. Cheong, S. A. Prahl, and A. J. Welch, "A review of the optical properties of biological tissues," IEEE J. Quantum Electron. **26**, 2166–2185 (1990).
37. V. G. Peters, D. R. Wyman, M. S. Patterson, and G. L. Frank, "Optical properties of normal and diseased human breast tissues in the visible and near infrared," Phys. Med. Biol. **35**, 1317–1334 (1990).
38. P. Gonzalez-Rodriguez and A. D. Kim, "Comparison of light scattering models for diffuse optical tomography," Opt. Express **17**, 8756–8774 (2009).
39. B. Gallas and H. H. Barrett, "Modeling all orders of scatter in nuclear medicine," in *Proceedings of IEEE Nuclear Science Symposium* (IEEE, 1998), pp. 1964–1968.
40. H. H. Barrett, B. Gallas, E. Clarkson, and A. Clough, *Computational Radiology and Imaging: Therapy and Diagnostics* (Springer, 1999).
41. Z. Wang, M. Yang, and G. Qin, "Neumann series solution to a neutron transport equation of slab geometry," J. Syst. Sci. Complex. **6**, 13–17 (1993).
42. M. Kim, G. Skofronick-Jackson, and J. Weinman, "Intercomparison of millimeter-wave radiative transfer models," IEEE Trans. Geosci. Remote Sens. **42**, 1882–1890 (2004).
43. T. Deutschmann, S. Beirle, U. F. M. Grzegorski, C. Kern, L. Kritten, U. Platt, Cristina Prados-Román, Pukīte Jānis, T. Wagner, B. Werner, and K. Pfeilsticker, "The Monte Carlo atmospheric radiative transfer model McArtim: introduction and validation of Jacobians and 3D features," J. Quant. Spectrosc. Radiat. Transfer **112**, 1119–1137 (2011).
44. A. K. Jha, M. A. Kupinski, T. Masumura, E. Clarkson, A. A. Maslov, and H. H. Barrett, "Simulating photon-transport in uniform media using the radiative transfer equation: A study using the Neumann-series approach," J. Opt. Soc. Am. A **29**, 1741–1757 (2012).
45. A. K. Jha, M. A. Kupinski, D. Kang, and E. Clarkson, "Solutions to the radiative transport equation for non-uniform media," in *Biomedical Optics* (Optical Society of America, 2010), p. BSuD55.
46. H. H. Barrett and K. J. Myers, *Foundations of Image Science*, 1st ed. (Wiley, 2004).
47. L. G. Henyey and J. L. Greenstein, "Diffuse radiation in the galaxy," Astrophys. J. **93**, 70–83 (1941).
48. F. Jacobs, E. Sundermann, B. D. Sutter, M. Christiaens, and I. Lemahieu, "A fast algorithm to calculate the exact radiological path through a pixel or voxel space," J. Comput. Inf. Technol. **6**, 89–94 (1998).
49. R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array," Med. Phys. **12**, 252–255 (1985).
50. E. Alerstam, T. Svensson, and S. Andersson-Engels, "Parallel computing with graphics processing units for high-speed Monte Carlo simulation of photon migration," J. Biomed. Opt. **13**, 060504 (2008).
51. Q. Fang and D. A. Boas, "Monte Carlo simulation of photon migration in 3D turbid media accelerated by graphics processing units," Opt. Express **17**, 20178–20190 (2009).
52. B. Huang, J. Mielikainen, H. Oh, and H.-L. A. Huang, "Development of a GPU-based high-performance radiative transfer model for the Infrared Atmospheric Sounding Interferometer (IASI)," J. Comput. Phys. **230**, 2207–2221 (2011).
53. C. Gong, J. Liu, L. Chi, H. Huang, J. Fang, and Z. Gong, "GPU accelerated simulations of 3D deterministic particle transport using discrete ordinates method," J. Comput. Phys. **230**, 6010–6022 (2011).
54. L. Szirmay-Kalos, G. Liktor, T. Umenhoffer, B. Toth, S. Kumar, and G. Lupton, "Parallel iteration to the radiative transport in inhomogeneous media with bootstrapping," IEEE Trans. Vis. Comput. Graphics **17**, 146–158 (2010).
55. "TESLA C2050/C2070 GPU Computing Processor: NVIDIA Tesla Datasheet" (2010).
56. "The Linux programmers manual" (2008).
57. J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with CUDA," ACM Queue **6**, 40–53 (2008).
58. J. W. Moore, "Adaptive X-ray computed tomography," Ph.D. thesis (College of Optical Sciences, University of Arizona, 2011).
59. "NVIDIA CUDA C programming guide" (2012), Version 4.2.
60. "CUDA C best practices guide" (2012), Version 4.1.
61. "Tuning CUDA Applications for Fermi" (2010), Version 1.3.
62. D. Boas, J. Culver, J. Stott, and A. Dunn, "Three dimensional Monte Carlo code for photon migration through complex

heterogeneous media including the adult human head," Opt. Express **10**, 159–170 (2002).
63. T. Tarvainen, M. Vauhkonen, V. Kolehmainen, and J. P. Kaipio, "Hybrid radiative-transfer-diffusion model for optical tomography," Appl. Opt. **44**, 876–886 (2005).
64. T. Spott and L. O. Svaasand, "Collimated light sources in the diffusion approximation," Appl. Opt. **39**, 6453–6465 (2000).
65. Z. Q. Zhang, I. P. Jones, H. P. Schriemer, J. H. Page, D. A. Weitz, and P. Sheng, "Wave transport in random media:the ballistic to diffusive transition," Phys. Rev. E **60**, 4843–4850 (1999).
66. A. Garofalakis, G. Zacharakis, G. Filippidis, E. Sanidas, D. D. Tsiftsis, V. Ntziachristos, T. G. Papazoglou, and J. Ripoll, "Characterization of the reduced scattering coefficient for optically thin samples: theory and experiments," J. Opt. A **6**, 725–735 (2004).
67. S. Narasimhan, M. Gupta, C. Donner, R. Ramamoorthi, S. Nayar, and H. Jensen, "Acquiring scattering properties of participating media by dilution," ACM Trans. Graph. **25**, 1003–1012 (2006).
68. G. Alexandrakis, T. J. Farrell, and M. S. Patterson, "Accuracy of the diffusion approximation in determining the optical properties of a two-layer turbid medium," Appl. Opt. **37**, 7401–7409 (1998).
69. A. A. Tanbakuchi, A. R. Rouse, and A. F. Gmitro, "Monte Carlo characterization of parallelized fluorescence confocal systems imaging in turbid media," J. Biomed. Opt. **14**, 044024 (2009).
70. J. Kurzak, S. Tomov, and J. Dongarra, "Autotuning GEMMs for Fermi," in "SC11 (2011).
71. K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz, N. Morgan, D. Patterson, K. Sen, J. Wawrzynek, D. Wessel, and K. Yelick, "A view of the parallel computing landscape," Commun. ACM **52**, 56–67 (2009).
72. D. S. Mishra BP, "Parallel computing environments: a review," IETE Technical Review **28**, 240–247 (2011).
73. Y. Mukaigawa, Y. Yagi, and R. Raskar, "Analysis of light transport in scattering media," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2010), pp. 153–160.
74. S. Nayar, G. Krishnan, M. Grossberg, and R. Raskar, "Fast separation of direct and global components of a scene using high frequency illumination," ACM Trans. Graph. **25**, 935–944 (2006).