A study on tetrahedron-based inhomogeneous Monte Carlo optical simulation

Haiou Shen and Ge Wang*

School of Biomedical Engineering and Sciences, Virginia Tech, Blacksburg, VA 24061, USA
**wangg@vt.edu*

Abstract: Monte Carlo (MC) simulation is widely recognized as a gold standard in biophotonics for its high accuracy. Here we analyze several issues associated with tetrahedron-based optical Monte Carlo simulation in the context of TIM-OS, MMCM, MCML, and CUDAMCML in terms of accuracy and efficiency. Our results show that TIM-OS has significant better performance in the complex geometry cases and has comparable performance with CUDAMCML in the multi-layered tissue model.

OCIS codes: (170.3660) Light propagation in tissues; (170.6920) Time-resolved imaging.

References and links

1. L. Wang, S. L. Jacques, and L. Zheng, "MCML--Monte Carlo modeling of light transport in multi-layered tissues," Comput. Methods Programs Biomed. **47**(2), 131–146 (1995).
2. A. Appel, "Some techniques for shading machine renderings of solids," AFIPS Joint Computer Conferences. Atlantic City, New Jersey (1968).
3. J. D. Foley, *Computer Graphics: Principles and Practice*, 2nd ed (Addison-Wesley, Reading, Mass., 1995).
4. E. Margallo-Balbás, and P. J. French, "Shape based Monte Carlo code for light transport in complex heterogeneous tissues," Opt. Express **15**(21), 14086–14098 (2007).
5. B. W. Rice, M. D. Cable, and M. B. Nelson, "In vivo imaging of light-emitting probes," J. Biomed. Opt. **6**(4), 432–440 (2001).
6. W. F. Cheong, S. A. Prahl, and A. J. Welch, "A review of the optical-properties of biological tissues," IEEE J. Quantum Electron. **26**(12), 2166–2185 (1990).
7. H. Shen, and G. Wang, "A tetrahedron-based inhomogeneous Monte Carlo optical simulator," Phys. Med. Biol. **55**(4), 947–962 (2010).
8. H. Li, J. Tian, F. Zhu, W. Cong, L. V. Wang, E. A. Hoffman, and G. Wang, "A mouse optical simulation environment (MOSE) to investigate bioluminescent phenomena in the living mouse with the Monte Carlo method," Acad. Radiol. **11**(9), 1029–1038 (2004).
9. W. Vogt, H. Shen, G. Wang, and C. G. Rylander, "Parametric study of tissue optical clearing by localized mechanical compression using combined finite element and Monte Carlo simulation," J. Innovative Opt. Health Sci. (JIOHS) **3**(3), 203–211 (2010).
10. Y. Lu, B. Zhu, H. Shen, J. C. Rasmussen, G. Wang, and E. M. Sevick-Muraca, "A parallel adaptive finite element simplified spherical harmonics approximation solver for frequency domain fluorescence molecular imaging," Phys. Med. Biol. **55**(16), 4625–4645 (2010).
11. C. G. Rylander, D. P. Davé, T. Akkin, T. E. Milner, K. R. Diller, and A. J. Welch, "Quantitative phase-contrast imaging of cells with phase-sensitive optical coherence microscopy," Opt. Lett. **29**(13), 1509–1511 (2004).
12. G. Yao, and L. V. Wang, "Monte Carlo simulation of an optical coherence tomography signal in homogeneous turbid media," Phys. Med. Biol. **44**(9), 2307–2320 (1999).
13. M. N. Rylander, Y. Feng, J. Bass, and K. R. Diller, "Heat shock protein expression and injury optimization for laser therapy design," Lasers Surg. Med. **39**(9), 731–746 (2007).
14. T. J. Pfefer, J. K. Barton, D. J. Smithies, T. E. Milner, J. S. Nelson, M. J. van Gemert, and A. J. Welch, "Modeling laser treatment of port wine stains with a computer-reconstructed biopsy," Lasers Surg. Med. **24**(2), 151–166 (1999).
15. G. Wang, W. Cong, K. Durairaj, X. Qian, H. Shen, P. Sinn, E. Hoffman, G. McLennan, and M. Henry, "In vivo mouse studies with bioluminescence tomography," Opt. Express **14**(17), 7801–7809 (2006).
16. V. Ntziachristos, J. Ripoll, L. V. Wang, and R. Weissleder, "Looking and listening to light: the evolution of whole-body photonic imaging," Nat. Biotechnol. **23**(3), 313–320 (2005).
17. K. H. Song, G. Stoica, and L. V. Wang, "In vivo three-dimensional photoacoustic tomography of a whole mouse head," Opt. Lett. **31**(16), 2453–2455 (2006).
18. A. Rosenthal, D. Razansky, and V. Ntziachristos, "Fast semi-analytical model-based acoustic inversion for quantitative optoacoustic tomography," IEEE Trans. Med. Imaging **29**(6), 1275–1285 (2010).

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

19. Q. Fang, "Mesh-based Monte Carlo method using fast ray-tracing in Plücker coordinates.," Biomed. Opt. Express **1**(1), 165–175 (2010).
20. N. Platis, and T. Theoharis, "Fast ray-tetrahedron intersection using Plücker coordinates," J. Graphics GPU Game Tools **8**(4), 37–48 (2003).
21. E. Alerstam, T. Svensson, and S. Andersson-Engels, "CUDAMCML, User manual and implementation notes," Available from *http://www.atomic.physics.lu.se/fileadmin/atomfysik/Biophotonics/Software/CUDAMCML.pdf*. (2009).
22. 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**(6), 060504 (2008).
23. L. A. Piegl, and W. Tiller, *The NURBS Book*, 2nd ed, Monographs in Visual Communications. (Springer, Berlin, 1997). (
24. J. Schöberl, "NETGEN: An advancing front 2D/3D-mesh generator based on abstract rules," Comput. Visualization Sci. **1**(1), 41–52 (1997).
25. B. Dogdas, D. Stout, A. F. Chatziioannou, and R. M. Leahy, "Digimouse: a 3D whole body mouse atlas from CT and cryosection data," Phys. Med. Biol. **52**(3), 577–587 (2007).

1. Introduction

Optical Monte Carlo (MC) simulation for complex heterogeneity has been a long-standing challenge since highly zigzag paths of photons in biological tissue make the determination of the ray-boundary intersection rather expensive. In simple geometry, such as the multi-layered tissue model used for MCML [1], a program can check the "Z" value of a photon to determine the ray-plane intersection. In complex geometry, there is no easy way to do the ray-boundary intersection test. With the advancement of computer graphics, the ray-tracing approach [2,3], which uses triangle mesh to represent a boundary, was introduced to the Monte Carlo simulation field [4]. However, there is a major difference between image rendering in computer graphics and photon propagation in biological tissue. A photon usually has a very long step size for image rendering, because it only changes its direction when it hits a surface [2,3]. On the other hand, biological tissue is highly scattering, and the step size of a photon is usually very small (~0.1 mm) [5,6]. Hence, a classic ray-tracing algorithm will take a huge amount of computational power in traveling around a complex data structure used.

In February 2010, we published a tetrahedron-based inhomogeneous optical Monte Carlo simulator (TIM-OS) [7] to handle complex heterogeneous geometry and reduce computational time. In the algorithm, the photon propagation is guided by a tetrahedral mesh. Since each tetrahedron has only four triangular surfaces, it becomes straightforward to compute the photon-surface interaction for a photon in a tetrahedron. Hence, it is not surprising that TIM-OS runs at least one order of magnitude faster than other ray-tracing-based implementations, such as MOSE [8] and Tri3DMC [4], as shown in [7]. With such an improvement, we applied TIM-OS in several applications such as optical clearing [9] and frequency-domain MC simulation [10]. We will see more applications for TIM-OS, such as, optical coherence tomography [11,12], laser therapy design [13,14], bioluminescence/fluorescence tomography [15,16], and photoacoustic tomography [17,18].

In July 2010, Fang proposed a "Mesh-based Monte Carlo Method" (MMCM) and developed a tetrahedron-based optical MC simulator [19]. MMCM uses Plücker coordinates for photon-tetrahedron intersection. Plücker coordinates has been widely used for ray-tracing to perform ray-triangle/tetrahedron intersection testing [20]. A major advantage with Plücker coordinates is the ability to handle a general mesh. As a result, MMCM can be extended to support various meshes such as a hexahedral mesh. Furthermore, MMCM uses linear Lagrange basis functions to represent fluence distributions, i.e., fluence is stored in the nodes of the mesh, and interpolation is used to compute the fluence at any position. In addition to use linear Lagrange basis functions for fluence quantification, it can be used to support continuous varying media, which is desirable in some applications. However, there is no detail information on how to support this feature in the MMCM paper, and the current MMCM simulation package (downloaded at 09/29/2010 from MMCM website at: http://mcx.sourceforgo.net/mmc) does not support this feature either.

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

In this article, we first compare the Plücker coordinate scheme in the MMCM paper with the TIM-OS implementation. Then, we analyze the problems in a direct linear Lagrange basis function scheme for continuous optical parameter. To overcome those problems, we propose a hybrid scheme to combine the linear Lagrange scheme and the piece-wise constant scheme, compare the piece-wise constant implementation (original TIM-OS) with the linear Lagrange scheme, and demonstrate the capability of TIM-OS for time-resolved MC simulation. Finally, we compare TIM-OS with three simulators in the two scenarios: MCML [1] and CUDAMCML [21,22] for the multi-layered tissue model and MMCM [19] for the general heterogeneous model.

2. Methods

2.1. Photon-mesh intersection

2.1.1. General mesh support

Biological tissue is turbid with strong scattering for visible/NIR light. In such media, a photon usually has a very small mean free path. Thus, a photon changes its direction frequently. In optical Monte Carlo simulation, whenever a photon changes its direction, the photon-mesh intersection test needs to be performed. Hence, the efficiency of the photon-mesh intersection test has a great impact on the overall performance. Because an element (no matter it is a tetrahedron, a hexahedron or any other shape) in a mesh is on average much larger than the mean free path of a photon, a mesh with a simpler shape shall be more efficient for the photon-mesh intersection test. Thus, the tetrahedron, which is simple in shape and powerful in representation, should have the speed advantage over all other types of mesh elements.

For the above reason, if an application needs to use a mesh different from a tetrahedron mesh, we can always use a preprocessor to decompose such a general mesh to a finer tetrahedral mesh, and then a postprocessor can convert the simulation result back to the original mesh with little additional computational cost. For example, a hexahedron can be partitioned into 5 or 6 tetrahedra, a pentahedron into 3 tetrahedra, and a quadratic tetrahedron into 8 tetrahedra.

2.1.2. Computational analysis

If a tetrahedral mesh is selected to represent an object, an efficient algorithm for photon-tetrahedron intersection is needed to achieve high performance. Plücker coordinates has been widely used to perform ray-triangle/ray-tetrahedron intersection in computer graphics [20]. MMCM used the Plücker coordinates for the same purpose in the optical Monte Carlo simulation [19]. The Plücker scheme provides a rigorous way to determine whether a photon will penetrate a triangle. When ray-tracing is being conducted in an image rendering scenario, the ability to determine whether the ray hits a triangle internally is needed. Since the ray may not hit a triangle, even if the triangle is the closest one to the ray path, as shown in Fig. 1(a) in which the closeness of a triangle to a photon path is defined in terms of the distance from the photon position to the intersection of the photon path and the plane containing the triangle. Nevertheless, if we already know that a photon is inside a tetrahedron, the extra computation to determine whether or not the intersection point is inside the triangle becomes unnecessary. As shown in Fig. 1(b), if a photon is inside a tetrahedron, the photon will hit inside the closest triangle along its path first. Hence, instead of determining if a photon will hit a triangle internally, we only need to find the closest triangle along the photon path, as implemented in TIM-OS [7]. As shown in the simplified pseudo-code in Appendix A, the average operations needed using the Plücker scheme for one photon-tetrahedron test is about 78 basic operations, including “+”, “-”, “×”, “/”, branching, and Boolean operations. On the other hand, the TIM-OS scheme only needs 44 basis operations. Hence, the TIM-OS scheme is faster than the Plücker scheme for optical Monte Carlo simulation.

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

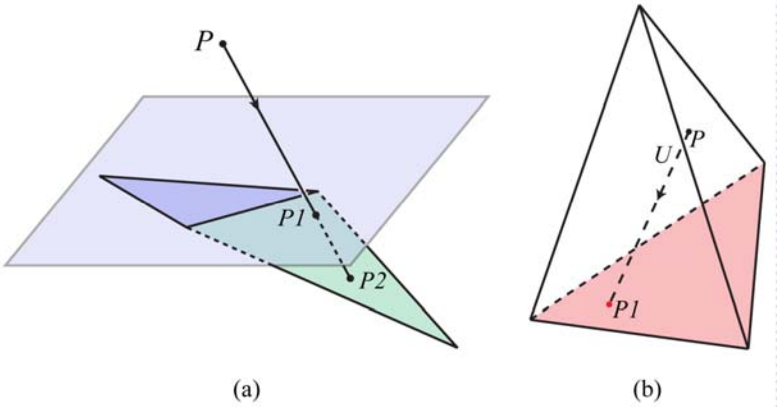


Fig. 1. Illustration of photon-mesh intersection under imaging rendering scenario and optical Monte Carlo simulation. (a) A photon may not hit the closest triangle in a general triangle mesh in imaging rendering, and (b) if a photon is inside a tetrahedron, then the photon will hit the closest triangle internally.

2.2. Linear Lagrange scheme

To support continuous varying media using linear Lagrange basis functions, we need to assign optical parameters to the nodes of a mesh. Then, the optical parameters at an internal position in a tetrahedron can be interpolated from the optical parameters of the four nodes of the tetrahedron. Unfortunately, there are several major issues for this scheme. First, since linear Lagrange basis functions can only represent a linear continuous function, the fundamental limitation of this approach is that discontinuity in an optical parameter distribution cannot be represented. Second, a photon shall follow a curvilinear path if the nodes of the tetrahedron have different refractive indexes, and it is not trivial to follow a curved path. Third, it is more complicated to determine the step size of a photon if it is in a tetrahedron with different optical coefficients on its nodes.

Although there are several difficulties in representing a linear continuous optical parameter distribution, we agree that such a representation capability would be interesting for finite element analysis. For example, optical clearing [9], which creates a continuing optical parameter distribution in the tissue with mechanical compression, could be a potential application of this model.

To overcome the fundamental limitation of linear Lagrange basis functions, a possible solution is that we can use a non-uniform rational B-spline (NURBs) [23] technique to represent an optical distribution. Nevertheless, this will make the computation of the photon step size extremely complex. Another way is a hybrid method that combines the piece-wise constant scheme with the linear Lagrange scheme. In this hybrid scheme, optical parameters are stored in tetrahedral mesh: the four nodes share the same optical parameters for a tetrahedron labeled with a piece-wise constant scheme and the four nodes have different optical parameters for a tetrahedron labeled with a linear Lagrange scheme. Definitely, there are other possible methods to overcome this problem. In the following section, we report a hybrid scheme in our TIM-OS framework.

Since the scheme used in MCML [1] can be applied to piece-wise constant scheme directly [7], in the following sections, we will only describe the MC simulation in the linear Lagrange scheme. We assume that the refractive index is always piece-wise constant to avoid the curvilinear path of a photon in a tetrahedron. The general case can be handled using variational calculus.

At a position $\vec{P} = (x, y, z)^T$ in a tetrahedron with four nodes: $\vec{P}_1$, $\vec{P}_2$, $\vec{P}_3$, and $\vec{P}_4$, $\vec{P}$ can be expressed as

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

$$\vec{P}=\sum_{i=1}^{4} l_i\vec{P}_i \quad \text{with} \quad \sum_{i=1}^{4} l_i=1 \quad \text{and} \quad 0\le l_i\le 1. \tag{1}$$

If $\vec{P}$ is not within the tetrahedron, then some l_i will be negative or greater than 1. Let $l_4=1-(l_1+l_2+l_3)$, then $[\vec{P}_1-\vec{P}_4,\vec{P}_2-\vec{P}_4,\vec{P}_3-\vec{P}_4][l_1,l_2,l_3]^T=[\vec{P}-\vec{P}_4]$. Hence,

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}=[\vec{P}_1-\vec{P}_4,\vec{P}_2-\vec{P}_4,\vec{P}_3-\vec{P}_4]^{-1}[\vec{P}-\vec{P}_4]. \tag{2}$$

If the optical parameters on the four nodes of the tetrahedron are μ_1, μ_2, μ_3, and μ_4, respectively, then the optical parameter on node $\vec{P}$ is $\mu=\sum_{i=1}^{4} l_i\mu_i$. Now, assume that a photon will move along a direction $\vec{U}$ with a step size s, then the new position will be $\vec{P}'=\vec{P}+\vec{U}s=\sum_{i=1}^{4} l'_i\,\vec{P}_i$. Hence,

$$\begin{bmatrix} l'_1 \\ l'_2 \\ l'_3 \end{bmatrix}=[\vec{P}_1-\vec{P}_4,\vec{P}_2-\vec{P}_4,\vec{P}_3-\vec{P}_4]^{-1}[\vec{P}+\vec{U}s-\vec{P}_4]=\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}+[\vec{P}_1-\vec{P}_4,\vec{P}_2-\vec{P}_4,\vec{P}_3-\vec{P}_4]^{-1}[\vec{U}]s.$$

Let

$$\begin{bmatrix} \Delta l_1 \\ \Delta l_2 \\ \Delta l_3 \end{bmatrix}=[\vec{P}_1-\vec{P}_4,\vec{P}_2-\vec{P}_4,\vec{P}_3-\vec{P}_4]^{-1}[\vec{U}]$$

and

$$\begin{bmatrix} l'_1 \\ l'_2 \\ l'_3 \\ l'_4 \end{bmatrix}=\begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \end{bmatrix}+\begin{bmatrix} \Delta l_1 \\ \Delta l_2 \\ \Delta l_3 \\ -(\Delta l_1+\Delta l_2+\Delta l_3) \end{bmatrix}s. \tag{3}$$

Then, we have

$$\mu'=\begin{bmatrix} l'_1 \\ l'_2 \\ l'_3 \\ l'_4 \end{bmatrix}\cdot\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix}=\mu+\big(\Delta l_1(\mu_1-\mu_4)+\Delta l_2(\mu_2-\mu_4)+\Delta l_3(\mu_3-\mu_4))s. \tag{4}$$

The above formula shows that the optical parameter changes linearly in a tetrahedron along any direction.

Let $\mu_t=\mu_a+\mu_s$ be the interaction coefficient at position $\vec{P}$, and μ_{t_i} are the interaction coefficients for nodes $\vec{P}_i$, $i=1,\cdots,4$. Then, along the direction $\vec{U}$, the interaction coefficients will be $\mu'_t=\mu_t+as$, where $a=\big(\Delta l_1(\mu_{t_1}-\mu_{t_4}\big)+\Delta l_2(\mu_{t_2}-\mu_{t_4})+\Delta l_3(\mu_{t_3}-\mu_{t_4}))$, then the step size s of a photon should satisfy the following equation:

$$\int_0^s \mu_t+axdx=\frac{1}{2}as^2+\mu_t s=-\ln(\xi) \tag{5}$$

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

where ξ is a uniform random number between 0 to 1. If $a = 0$, then $s = -\ln(\xi)/\mu_t$. If $a > 0$, $s = \left(-\mu_t + \sqrt{\mu_t^2 - 2aln(\xi)}\right)/a$. If $a < 0$, then it is possible that there is no real solution for the equation. In this case, if $\mu_t^2 - 2aln(\xi) \geq 0$, then $s = \left(-\mu_t + \sqrt{\mu_t^2 - 2aln(\xi)}\right)/a$. If $\mu_t^2 - 2aln(\xi) < 0$, it means that the step size of the photon is too large. To solve this problem, we can first have a step size of $s = -\mu_t / a$ and save the rest part: $\ln(\xi') = \ln(\xi) - \mu_t^2 / 2a$ for the next step. If a photon needs to go through several regions, we can use the same scheme in MCML [1].

The anisotropy scattering factor can be treated simply. Each time when we move a photon to a scattering location, we can compute the local anisotropy scattering factor using the following equation:

$$g' = \sum_{i=1}^{4} l_i g_i \tag{6}$$

where g_i is the anisotropy scattering factor for node $\vec{P}_i$, and l_i defined by Eq. (2).

In addition to the aforementioned issues on how to use linear Lagrange basis functions for the optical parameter presentation, there is also a problem in the volumetric absorption/fluence quantification. In MC, the volumetric fluence for a tetrahedron is computed as: $\Phi = E_a / (V\mu_a)$, where Φ is the fluence, E_a the absorbed energy, V the volume, and μ_a the absorption coefficient. In the piece-wise constant scheme, this formula works well. We can easily convert from absorbed energy to fluence or from fluence to absorbed energy. In the linear Lagrange scheme, since in a tetrahedron the local absorption coefficient may change over absorption locations, the fluence computation needs a different formula:

$$\Phi = (\sum \frac{E_{a_j}}{\mu_{a_j}})/V, \tag{7}$$

where μ_{a_j} is the local absorption coefficient at the absorption location for energy E_{a_j}. However, after the simulation we cannot precisely convert the fluence back to the absorbed energy. The only way to have the information is to store both the absorbed energy and the fluence separately. This will increase the memory usage and decrease the simulation speed, especially in the time-domain simulation.

3. Results

TIM-OS was implemented in C/C + + and compiled with an Intel C compiler on Ubuntu 9.10 64bit Linux. We implemented the linear Lagrange basis function to represent optical properties in the same framework of TIM-OS. A workstation with a non-uniform memory architecture (NUMA), two Xeon L5520 CPUs (8 CPU cores with hyper-threading enabled), 32GB memory and 4 Nvidia Tesla C1060 graphics card was used in all the simulation tests. We used 32 threads for TIM-OS. In all the following tests, the unit is in millimeter for length and mm^{-1} for absorption and scattering coefficients. Optical parameters vary significantly with wavelength and tissue type. At 700nm, scattering coefficients for mouse organs are between 6.48mm^{-1} and 23.4mm^{-1} and the absorption coefficients between 0.0027mm^{-1} to 0.23mm^{-1} [10]. While all of our examples are artificial examples, we selected the optical parameters within or close to the aforementioned ranges.

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

3.1 Comparison of linear Lagrange and piece-wise constant schemes

In this experiment, a cubic phantom was used. It is of size $6\times6\times6$ mm centered at the origin $(0,0,0)$ mm (Fig. 2). Three different sources were used in the experiment: a point source located at $(0.1,0.2,1.5)$ mm, a pencil beam source located at $(-0.1,3,-0.1)$ mm in a direction $(0,-1,0)$ mm, or a pencil beam source located at $(-0.1,0.1,3)$ mm in a direction $(0,0,-1)$ mm. Four tetrahedral meshes were generated with 6, 750, 6,000, and 48,000 tetrahedra. We first compared the linear Lagrange and piece-wise constant schemes under a uniform optical condition ($\mu_a = 0.05\,\text{mm}^{-1}$, $\mu_s = 10\,\text{mm}^{-1}$, $g = 0.9$, and $n = 1.3$). The results showed that the two schemes matched almost exactly. Then, we considered an optical distribution where the top surface of the cube had lower absorption and higher scattering ($\mu_a = 0.01\ \text{mm}^{-1}$, $\mu_s = 10\,\text{mm}^{-1}$, $g = 0.9$, and $n = 1.3$) and the lower surface of the cube had higher absorption and lower scattering ($\mu_a = 0.05\ \ \text{mm}^{-1}$, $\mu_s = 5\,\text{mm}^{-1}$, $g = 0.9$, and $n = 1.3$). In the piece-wise constant scheme, the optical parameters for a tetrahedron were determined by the center position of the tetrahedron. In linear Lagrange scheme, the optical parameters were assigned to the nodes of the mesh. We then simulated light propagation using the two schemes respectively for 12 combinations of four meshes and three source positions. In each case, 10^9 photons were simulated.

Let $\Phi = \{\phi_1,...,\phi_n\}$ represent the fluence values, and Φ^{sl} the surface fluence values from the linear Lagrange scheme, Φ^{vl} the volumetric fluence from the linear Lagrange scheme, Φ^{sp} the surface fluence values from the piece-wise constant scheme, Φ^{vp} the volumetric fluence values from the piece-wise constant scheme. In this example, we compared five quantities for the two schemes: (1) the average relative difference in the surface fluence ($E1 = \left(\sum_{i=1}^{n} \frac{|\phi_i^{sp} - \phi_i^{sl}|}{\phi_i^{sl}}\right) / n$), (2) the relative difference in the maximal surface fluence ($E2 = \frac{\max(\Phi^{sp}) - \max(\Phi^{sl})}{\max(\Phi^{sl})}$), (3) the average relative difference in the volumetric fluence ($E3 = \left(\sum_{i=1}^{n} \frac{|\phi_i^{vp} - \phi_i^{vl}|}{\phi_i^{vl}}\right) / n$), (4) the relative difference in the maximal volumetric fluence ($E4 = \frac{\max(\Phi^{vp}) - \max(\Phi^{vl})}{\max(\Phi^{vl})}$), and (5) the relative difference in the total escaped energy ($E5 = \frac{\sum \phi_i^{sp} s_i - \sum \phi_i^{sl} s_i}{\sum \phi_i^{sl} s_i}$), where s_i is the size of the triangle for surface fluence ϕ_i^{sp} or ϕ_i^{sl} . Table 1 shows the results obtained with the two schemes. While the number of tetrahedra becomes larger, the size of each tetrahedron becomes smaller; the elements far away from the source will have low absorbed energy and high noise. To reduce the noise, we only considered the 1,000 largest fluence values for comparison. As shown in Table 1, when the mesh contains only 6 tetrahedra, the two schemes have major difference. When the number of tetrahedra reaches 750 (the average volume of a tetrahedron was 0.288 mm^3), the two schemes performed almost exactly the same. In addition to that, the positions of the maximal fluence values from the two schemes matched exactly on all the four meshes. Also, as shown in Table 1, the piece-wise constant scheme was about 30% faster than the linear Lagrange scheme.

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

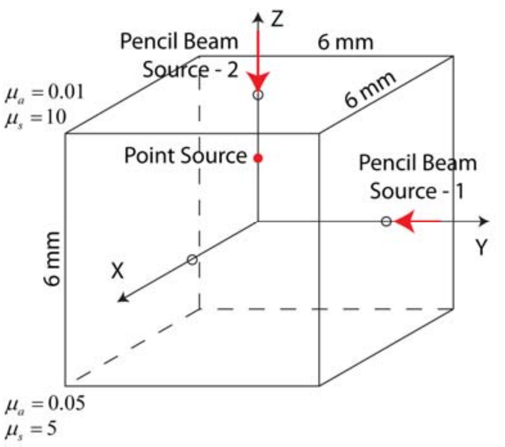


Fig. 2. A cubic phantom with linearly varying optical parameter distributions and three sources.

One needs to pay an attention to the scalability of TIM-OS. As show in Table 1, when the number of tetrahedra was increased by 8,000 times, the running time of the original TIM-OS was only increased by about one fold. Although there is some advantage in representing an optical parameter distribution more accurately with linear Lagrange basis functions, the original TIM-OS scheme can always be applied to a finer mesh for an almost identical result at a less computational cost than that with linear Lagrange basis functions.

Table 1. Differences between the original and hybrid TIM-OS schemes. TIM-OS represents piece-wise constant scheme and TIM-OS-L stands for linear Lagrange scheme.

			# Tetrahedra			
Meshes			6	750	6000	48000
Point source	Relative Difference	E1	3.39%	0.31%	0.16%*	0.18%*
		E2	10.1%	1.23%	0.81%*	0.95%*
		E3	4.07%	0.3%	0.12%*	0.094%*
		E4	7.16%	1.85%	0.53%*	0.51%*
		E5	0.97%	0.043%	0.0076%	0.0015%
	Run Time (S)	TIM-OS	732	791	952	1437
		TIM-OS-L	898	1092	1364	2166
Pencil beam source - 1	Relative Difference	E1	3.39%	0.27%	0.27%*	0.18%*
		E2	5.13%	0.76%	1.54%*	0.81%*
		E3	0.83%	0.21%	0.35%*	0.14%*
		E4	1.50%	0.82%	1.44%*	0.44%*
		E5	0.74%	0.0003%	0.09%	0.019%
	Run Time (S)	TIM-OS	835	983	1150	1649
		TIM-OS-L	1314	1334	1649	2490
Pencil beam source - 2	Relative Difference	E1	5.31%	0.28%	0.21%*	0.18%*
		E2	12.85%	1.45%	0.97%*	0.73%*
		E3	3.28%	0.21%	0.23%*	0.083%*
		E4	4.03%	1.05%	1.52%*	0.32%*
		E5	2.65%	0.032%	0.041%	0.0036%
	Run Time (S)	TIM-OS	793	905	1091	1625
		TIM-OS-L	1031	1255	1549	2458

*Indicates that the result comes from the first 1,000 largest fluence values from surface triangles or tetrahedra.

3.2. Time-resolved simulation

While we already demonstrated and validated the time-resolved TIM-OS for frequency-domain fluorescence imaging [10], here we show another interesting example, which illustrates the use of glass material in the simulation to support some realistic light source types. As shown in Fig. 3, we set up two numerical phantoms. The first phantom contained a

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

spherical lens in the air to demonstrate that a laser pulse can be focused by the lens. A tissue layer was added above the spherical lens in the second phantom to demonstrate that the light propagation in tissue. The mesh used in the simulation was generated using NetGen [24] with 1,202,941 tetrahedra. Table 2 shows the optical parameters used in this simulation.

It has always been tricky for an MC simulator to deal with regions with no absorption and scattering, such as ideal glass and air. Since in MC the quantification of volumetric fluence relies on an equation similar to Eq. (6) [1], which needs to normalize the absorbed energy by the absorption coefficient. However, in the ideal air/glass material the absorption is zero.

In TIM-OS, we can setup a material to be pure glass/air by setting the value of g to 1. A user can then set the absorption coefficient to some small value, and set the scattering coefficient to a value whose inverse is about 10 times of the average edge length of the mesh. In the simulation, when TIM-OS detects that a photon is in a glass/air region, it will still compute the absorbed weight (weight lost by absorption) in the glass/air region, but it will not reduce the absorbed weight from the photon's weight. Hence, the absorption/scattering coefficient in the glass/air region will neither affect the photon weight nor the photon path, because TIM-OS does not scatter the photon. A lower scattering coefficient in the glass/air region will allow TIM-OS to give a smooth volumetric fluence, while a larger scattering coefficient allows a faster simulation speed. By doing so, we need to remember that the true absorption and scattering coefficients in the pure glass/air region are zero, and the absorbed energy in the glass/air region is only for the sake of volumetric fluence computation.

In both the simulation cases, the time step was set to 0.1×10^{-12} second (0.1 picosecond), and 500 steps were simulated. Figure 3 shows the simulation results at 0.1ps, 4.9ps, 10ps, and 15ps respectively. It can be seen that TIM-OS correctly simulated the focusing effect of the spherical lens and the wave front of the light pulse on the "Z" axis matched the analytical expected frontier well, which can be computed by the light speed in glass, air, and tissue. Please note that the tetrahedron size is much larger than the distance a photon travels in 0.1 ps. The large tetrahedron size certainly blurs the simulated wave front. In this simulation, it was also found that a large portion of the light from the source was directly reflected by the spherical surface of the lens. A different lens shape may have better light delivery efficiency. Two time-resolved movies were generated and uploaded to youtube.com and can be accessed at TIM-OS website at http://www.imaging.sbes.vt.edu/software/.

Table 2. Optical parameters for air, glass, and tissue used in TIM-OS

Material	Absorption (mm^{-1})	Scattering (mm^{-1})	Anisotropy	Refractive index
Air	0.001	1	1	1.0003
Glass	0.001	1	1	1.4580
Tissue	0.050	10	0.9	1.3600

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

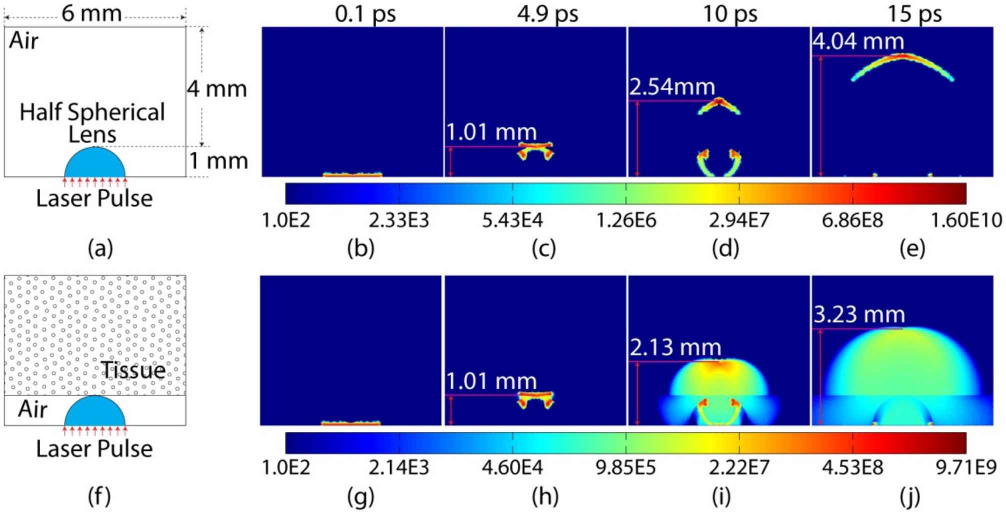


Fig. 3. Time resolved simulation for a laser pulse. (a) A spherical lens in air, (b), (c), (d) and (e) the simulation results for the phantom (a) at time instants 0.1 ps, 4.9 ps, 10 ps and 15 ps, respectively; (f) A spherical lens under tissue, (g), (h), (i) and (j) the simulation results for the phantom (f) at time instants 0.1 ps, 4.9 ps, 10 ps and 15 ps, respectively.

3.3. Comparison of MMCM, TIM-OS, MCML, and CUDAMCML

Based on the MMCM paper, Fang implemented a tetrahedron-based Monte Carlo simulator which is referred to as MMC [19]. While both MMC and TIM-OS use a tetrahedral mesh to guide photon propagation, the two schemes are quite different. It is necessary to validate the two simulators such as with MCML, which is a widely-used and well-accepted standard in the optical Monte Carlo simulation. The MMC software package was downloaded on September 29, 2010 from its website (http://mcx.sourceforgo.net/mmc) and compiled with OpenMP enabled (OMP_NUM_THREADS = 32 was set in the Linux system). MCML was enhanced by replacing the random number generator with the Mersenne Twister pseudorandom number generator in the Intel Math Kernel Library (MKL). The random number generator can create an array of random numbers to be used one by one as needed. In the following, we used the new MCML. Since MMC uses single precision floating point operations, we also included CUDAMCML, which is also in the single precision floating point format.

Initially, we did a comparison on some single-layer tissues with a refractive index mismatch on the boundary, because the refractive index mismatch is common in biophotonics. We observed that MMC had errors larger than the other three simulators. After some discussion with the author of MMC, we realized that the current MMC only solves problems with a constant refractive index (including the environment). Thus, in the following we will study problems with no refractive index mismatch.

The four simulators were tested on a simple single-layer tissue problem and a two-layer tissue problem. The thickness of the single-layer tissue is 1 mm. In MCML and CUDAMCML, a pencil light beam aimed at (0, 0, 0) mm with a direction (0, 0, 1). A tetrahedral mesh was generated using the MATLAB code in the MMC package. The mesh was of size 20 × 20 × 1 mm centered at the point (0, 0, 0.5) mm and contained 2,400 tetrahedra. In MMC and TIM-OS, we shifted the light source to (–3.333E-4, –6.667E-4, 0) mm to prevent the light from hitting a node directly. In the single-layer case, we simulated a low absorption case ($\mu_a = 0.05\,\text{mm}^{-1}$, $\mu_s = 10\ \text{mm}^{-1}$, $g = 0.9$) and a high absorption case ($\mu_a = 0.5\,\text{mm}^{-1}$, $\mu_s = 10\ \text{mm}^{-1}$, $g = 0.9$). Next, we compared the four simulators in the case of a two-layer tissue with two different materials: ($\mu_a = 0.05\,\text{mm}^{-1}$, $\mu_s = 10\ \text{mm}^{-1}$, $g = 0.9$)

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

and ($\mu_a = 0.10\ \text{mm}^{-1}$, $\mu_s = 5\ \text{mm}^{-1}$, $g = 0.9$). Each layer was 1 mm in thickness. A tetrahedral mesh was generated using the MATLAB code in the MMC package. The mesh was of size 40 × 40 × 2 mm centered at the point (0, 0, 1) mm and consisted of 12 tetrahedra. In MMC and TIM-OS, we shifted the light source to (3.333E-2, –3.333E-2, 0) mm to prevent the light from hitting a node directly.

In all the following simulation tests, we only focused on one index - the absorbed fraction, i.e., the total absorbed energy divided by the total source energy. In each case, 10 simulation runs were performed with different random seeds. In each case, 30 million of photons were simulated. The averages and standard deviations of the absorbed fraction were listed in Table 3. Then, we used the average absorbed fraction of MCML as the reference to compute the relative errors. As shown in Table 3, the results of the four simulators all matched well.

As far as the computational efficiency is concerned, MCML used one CPU core, CUDAMCML used one Nvidia Tesla C1060 graphics card, MMC and TIM-OS used 32 threads to fully utilize 8 CPU cores. For the multi-layered tissue models, among the four simulators, CUDAMCML and TIM-OS were the fastest with comparable speeds. Note that MCML used only one CPU core in this comparative study. If MCML is parallelized on all eight cores of the computer, it should have a comparable speed as CUDAMCML and TIM-OS.

Table 3. Comparison among MCML, CUDAMCML, MMC, and TIM-OS in a numerical study with a single-layer and two-layer tissue phantoms

		Single layer					
		Low Absorption Case		High Absorption Case		Two layer	
		Absorbed Fraction	Time (sec)	Absorbed Fraction	Time (sec)	Absorbed Fraction	Time (sec)
MCML	Average	0.0872044	91.9	0.5452665	96.5	0.2215219	155.8
	Std Deviation	0.0000105		0.0000410		0.0000218	
CUDA MCML	Average	0.0870503	15.0	0.5443552	15.3	0.2211585	15.5
	Relative Error	–0.17674%		–0.16712%		–0.16405%	
	Std Deviation	0.0000080		0.0000246		0.0000127	
MMC	Average	0.0872160	85.2	0.5453470	86.0	0.2214840	201.9
	Relative Error	0.01330%		0.01476%		–0.01710%	
	Std Deviation	0.0000117		0.0000343		0.0000267	
TIM-OS	Average	0.0872111	11.0	0.5452786	11.6	0.2215785	17.2
	Relative Error	0.00771%		0.00222%		0.02555%	
	Std Deviation	0.0000037		0.0000280		0.0000187	

We also observed one interesting phenomenon with MMC. We used MMC to simulate cases with different numbers of photons: 10^5, 10^6, 10^7, 10^8, 2×10^8, and 4×10^8. The absorbed fractions for these six cases were shown in Table 4. When the number of photon larger than 10^8, the absorbed fraction quickly decreased, which should be incorrect. We first thought that the problem came from the cumulative error in single precision floating point operations, but we did not observe a similar behavior using CUDAMCML, which also used single precision floating point operations.

Table 4. Absorbed fractions with the single layer tissue phantom and different numbers of photons

# Photon	10^5	10^6	10^7	10^8	2×10^8	4×10^8
MMC	0.087240	0.087230	0.087230	0.087910	**0.081950**	**0.064700**
MCML	0.087611	0.087204	0.087205	0.087203	0.087206	0.087206
CUDA MCML	0.086021	0.086962	0.087048	0.087052	0.087053	0.087060
TIM-OS	0.087340	0.087207	0.087206	0.087215	0.087210	0.087208

In addition to the above studies using the homogeneous problem, we also tested TIM-OS and MMC with a heterogeneous problem provided in the MMC package (mmc/examples/meshtest/sph2.inp). The geometry and optical properties of the problem were detailed as the second example in the MMC paper. In this example, the mesh contained 209,847 tetrahedra. We first compared the two simulators in the static case. While MMC spent 474 seconds to simulate 3×10^7 photons, TIM-OS took 82 seconds. In the time-domain simulation, a time window of 0 to 5ns was used with 0.1 ns time step. While MMC spent 440 seconds, TIM-OS took 100 seconds.

We then test the two solvers with a digital mouse model downloaded from the DIGIMOUSE website [25]. This digital mouse model contains all the major organs represented in 306,773 tetrahedra. We studied a high absorption case with optical parameters as in the original TIM-OS publication [7] and a low absorption case by reducing the absorption coefficient of muscle and spleen to 0.05 mm^{-1}. A pencil beam was placed at (9.8, 41.13, 11.47) mm with the direction (0.873, 0.4364, –0.218). Ten million of photons were simulated in each case. As shown in Table 5, MMC ran about 5 times slower than TIM-OS.

Table 5. Absorbed fractions computed with the mouse atlas in the high and low absorption cases

	High Absorption Case		Low Absorption Case	
	Absorbed Fraction	Time	Absorbed Fraction	Time
MMC	0.738700	198	0.306610	314
TIM-OS	0.738887	36	0.307645	64

Discussions and conclusion

In this article, we have studied several implementation issues for tetrahedron-based inhomogeneous Monte Carlo optical simulation. Our analysis shows that the scheme used in TIM-OS is faster than the Plücker coordinate scheme used in MMC. Our experimental comparison shows that TIM-OS is more efficient than MMC. We have also compared TIM-OS with MCML and CUDAMCML for the multi-layered tissue model and found that TIM-OS has the same level of accuracy as MCML and CUDAMCML and a comparable simulation speed as CUDAMCML. While the current MMC program could not solve problems with a refractive index mismatch, a more complete comparison will be done when a new version of MMC is publicly available.

Given the speed advantage of TIM-OS, we can use an indirect way to support a general mesh. For example, we can partition a hexahedral mesh into a finer tetrahedral mesh and then convert the simulation result on a tetrahedral mesh to that on a general mesh. We believe that this approach should be faster than a directly supported general mesh using the Plücker coordinate scheme.

We have presented a hybrid scheme to support linear Lagrange basis function to represent optical properties and developed an algorithm to determine the photon step size for the linear Lagrange scheme. We then implemented the linear Lagrange scheme in our TIM-OS framework and compared the linear Lagrange scheme with the piece-wise constant scheme (the original TIM-OS). The results show that the piece-wise constant scheme is sufficiently accurate even with a coarse mesh. Since the utility of the linear Lagrange scheme is rather limited, we believe that the piece-wise constant scheme is generally effective and efficient.

Yet another interesting feature demonstrated in section 3.2 is that TIM-OS can support various real light source types by including environmental material and a lens in a mesh. In the second example, we set up a mesh to include tissue, air, and a spherical lens. The simulator has correctly produced the focusing effect of the lens and the refraction/reflection phenomenon on the tissue surface. Even the mesh is very large and complex (>1 million tetrahedra), TIM-OS does not suffer from any significant speed penalty.

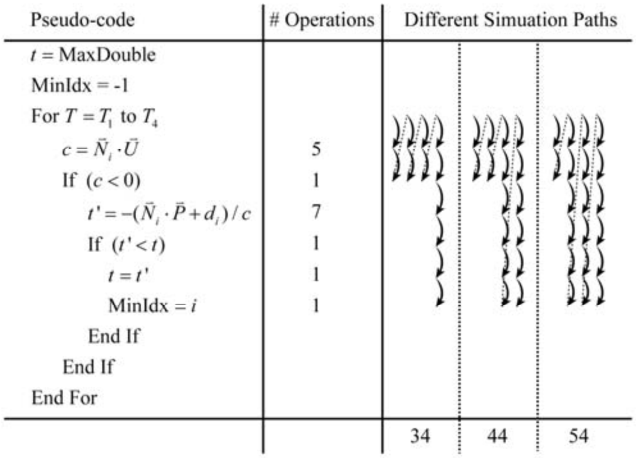
In conclusion, TIM-OS is superior to MMC in terms of efficiency for the general heterogeneous models, as discussed above. Among optical MC simulators, TIM-OS seems the most efficient one that handles general inhomogeneous geometry. Also, TIM-OS allows users to combine traditional optical simulation and Monte Carlo simulation. Therefore, we believe

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010

that TIM-OS will find more applications in the future. A TIM-OS resource is at http://www.imaging.sbes.vt.edu/software/.

Appendix A: Photon-Tetrahedron Intersection Test in the TIM-OS scheme

In this analysis, we assume all the operations, including “+”, “–”, “×”, “/”, “IF”, “≥”, “≤”, “==”, and “AND”, have the same computational cost. We also ignore the overhead of all “for loops”. Assume T_1 to T_4 are the four triangles of the tetrahedron, $\vec{N}_i$ is the unit normal vector of T_i and the unit normal vector points inside the tetrahedron, d_i is a constant such that $\vec{N}_i \cdot \vec{P}' + d_i = 0$ for any point $\vec{P}'$ on the plane contains the triangle T_i. Now, assume $\vec{P} = \{x, y, z\}$ is the photon position, $\vec{U} = \{U_x, U_y, U_z\}$ the unit direction of the photon movement, t the step size needed for the photon to reach the intersection point along the photon path. The photon-tetrahedron intersection test in TIM-OS can be represented by the pseudo-code in Fig. 4. In the best case, in the four triangles only one triangle faces the photon’s path. In the worst case, three triangles face the photon’s path. Hence, we need about 44 operations on average to perform one photon-tetrahedron intersection test in TIM-OS.



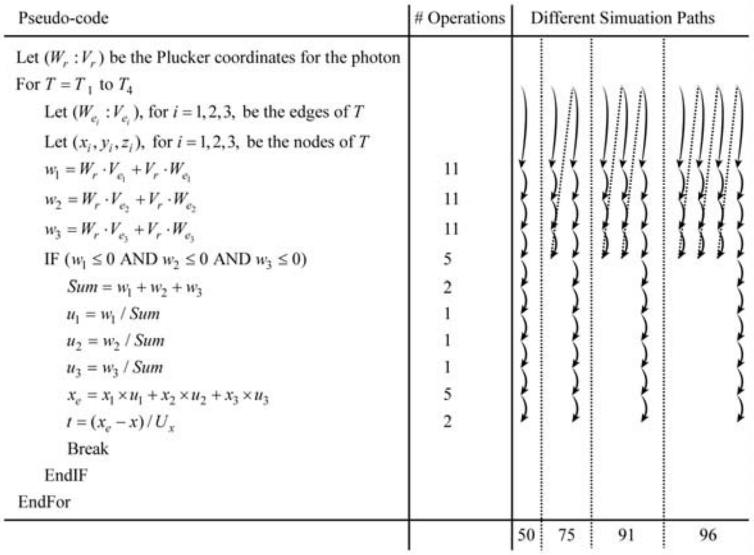
Pseudo-code	# Operations	Different Simuation Paths		
t = MaxDouble				
MinIdx = -1				
For $T = T_1$ to T_4				
$c = \vec{N}_i \cdot \vec{U}$	5			
If $(c < 0)$	1			
$t' = -(\vec{N}_i \cdot \vec{P} + d_i)/c$	7			
If $(t' < t)$	1			
$t = t'$	1			
MinIdx = i	1			
End If				
End If				
End For				
		34	44	54

Fig. 4. Pseudo-code for the photon-tetrahedron intersection test in the TIM-OS scheme.

Appendix B: Photon-Tetrahedron Intersection Test in the Plücker scheme

The photon-tetrahedron intersection test in Plücker Coordinates can be represented by the simplified pseudo-code in Fig. 5. In this pseudo-code, we ignored the computational cost of the photon’s Plücker coordinates, i.e. (W_r, V_r). The pseudo-code first computes w_i ($i = 1, 2, 3$) for the selected triangle. If $\forall i, w_i \le 0$, then, it determines the step size (t) needed to reach the intersection point. In the best case, the first triangle in the loop will be hit by the photon. In this case, we need at least 50 operations. If the program needs to test with two triangles, then we need at least 75 operations (one of the w_i value can be reused from the previous loop). If the program needs to test with three and four triangles, then we need at least 91 and 96 operations respectively. Then, on average we need at least 78 operations for one photon-tetrahedron intersection test in Plücker coordinates.

#137688 - $15.00 USD Received 5 Nov 2010; revised 29 Nov 2010; accepted 29 Nov 2010; published 3 Dec 2010



```
Pseudo-code                                                             # Operations   Different Simuation Paths
Let $(W_r : V_r)$ be the Plucker coordinates for the photon
For $T = T_1$ to $T_4$
    Let $(W_{e_i} : V_{e_i})$, for $i = 1,2,3,$ be the edges of $T$
    Let $(x_i, y_i, z_i)$, for $i = 1,2,3,$ be the nodes of $T$
    $w_1 = W_r \cdot V_{e_1} + V_r \cdot W_{e_1}$                       11
    $w_2 = W_r \cdot V_{e_2} + V_r \cdot W_{e_2}$                       11
    $w_3 = W_r \cdot V_{e_3} + V_r \cdot W_{e_3}$                       11
    IF ($w_1 \le 0$ AND $w_2 \le 0$ AND $w_3 \le 0$)                    5
        $Sum = w_1 + w_2 + w_3$                                         2
        $u_1 = w_1 / Sum$                                               1
        $u_2 = w_2 / Sum$                                               1
        $u_3 = w_3 / Sum$                                               1
        $x_e = x_1 \times u_1 + x_2 \times u_2 + x_3 \times u_3$        5
        $t = (x_e - x) / U_x$                                           2
        Break
    EndIF
EndFor
                                                                                       50   75   91   96
```

Fig. 5. Pseudo-code for the photon-tetrahedron intersection test in the Plücker scheme.

Acknowledgment

The work is partially supported by NIH R01HL098912.