Meshless Monte Carlo radiation transfer method for curved geometries using signed distance functions

Lewis McMillan,[a,*] **Graham D. Bruce**,[a] **and Kishan Dholakia**[a,b,c]
[a]University of St Andrews, SUPA School of Physics and Astronomy, St Andrews, Scotland
[b]Yonsei University, College of Science, Department of Physics, Seoul, South Korea
[c]The University of Adelaide, School of Biological Sciences, Adelaide, South Australia, Australia

Abstract

Significance: Monte Carlo radiation transfer (MCRT) is the gold standard for modeling light transport in turbid media. Typical MCRT models use voxels or meshes to approximate experimental geometry. A voxel-based geometry does not allow for the precise modeling of smooth curved surfaces, such as may be found in biological systems or food and drink packaging. Mesh-based geometry allows arbitrary complex shapes with smooth curved surfaces to be modeled. However, mesh-based models also suffer from issues such as the computational cost of generating meshes and inaccuracies in how meshes handle reflections and refractions.

Aim: We present our algorithm, which we term signedMCRT (sMCRT), a geometry-based method that uses signed distance functions (SDF) to represent the geometry of the model. SDFs are capable of modeling smooth curved surfaces precisely while also modeling complex geometries.

Approach: We show that using SDFs to represent the problem's geometry is more precise than voxel and mesh-based methods.

Results: sMCRT is validated against theoretical expressions, and voxel and mesh-based MCRT codes. We show that sMCRT can precisely model arbitrary complex geometries such as microvascular vessel network using SDFs. In comparison with the current state-of-the-art in MCRT methods specifically for curved surfaces, sMCRT is more precise for cases where the geometry can be defined using combinations of shapes.

Conclusions: We believe that SDF-based MCRT models are a complementary method to voxel and mesh models in terms of being able to model complex geometries and accurately treat curved surfaces, with a focus on precise simulation of reflections and refractions. sMCRT is publicly available at https://github.com/lewisfish/signedMCRT.

© *The Authors. Published by SPIE under a Creative Commons Attribution 4.0 International License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI.* [DOI: 10.1117/1.JBO.27.8.083003]

Keywords: Monte Carlo; light transport; signed distance functions; geometry; meshless.

Paper 210394SSRRR received Dec. 20, 2021; accepted for publication Jul. 20, 2022; published online Aug. 4, 2022.

1 Introduction

The modeling of light transport is important to our understanding of how light interacts with turbid media. It allows us to make predictions of the viability of treatment modalities,[1,2] simulate the behavior of complex shaped light in highly scattering media,[3] retrieve images of objects in highly scattering media,[4] and optimize light sensors in the food and drink industry[5] among other applications.

The radiation transfer equation (RTE) describes the transfer of energy in a medium. However, analytical solutions for the RTE only exist for simple geometries. Therefore, numerical methods such as the diffusion method[6] or the Monte Carlo radiation transfer method (MCRT) must be

*Address all correspondence to Lewis McMillan, lm959@st-andrews.ac.uk

used to compute a solution. The current "gold standard" of modeling light transport in turbid media is the MCRT method. MCRT can model light transport in arbitrary 3D geometries and model several microphysics phenomena such as Raman scattering,[7,8] fluorescence,[9,10] and polarization[11–13] and has been applied to problems ranging from light propagation in dynamic fluid systems[14,15] to simulating thermal gradients in illuminated tissue.[16,17]

To simulate the transport of light through a medium, the geometry of the problem must be modeled. Most Monte Carlo codes rely on voxels[18–20] or meshes[21,22] to approximate the geometry of the problem. Voxel models are only suitable for the simplest problems that do not require accurate treatment of curved surfaces, due to their cubic nature.[23] Curved surfaces arise in many problems where MCRT may be used, e.g., the propagation of light through an optical system, the anatomy of animals or humans such as the brain or vascular networks, among many other possible examples. In contrast, mesh-based models can more accurately treat curved surfaces but can be computationally expensive to produce[24,25] and MCRT codes require extensive software engineering to incorporate meshes in a computationally efficient manner. Several authors have created fast tetrahedral mesh-based codes,[21,26,27] which allow better approximate treatment of curved edges, provided that the tetrahedral meshes are refined to such a level that the underlying geometry can be precisely modeled. High levels of mesh refinement require more computational time and large amounts of RAM to create the mesh, and require additional storage space due to the amount of nodes and elements that make up the mesh. Additionally, even with high levels of mesh refinement, curved surfaces may still not be precisely modeled[28] unless vertex normals are used with interpolation.[29]

A number of previous methods have been introduced to tackle the problem of smooth surfaces in voxel-based models. Tran and Jacques preprocessed the voxels to determine where the material interfaces are and computed surface normals for each voxel, which can then be smoothed via interpolation to create curved surfaces.[30] While this method, on the whole, improves the modeling of curved surfaces, it can have an increased memory footprint and is more computationally intensive. Alternatively, implicit surfaces can be defined using mathematical formulas.[31–36] Periyasamy and Pramanik's work, Zhang et al. work, and molecular optical simulation environment (MOSE) by Li et al. all use a small subset of shapes (spheres, cylinder, and ellipsoids) to create geometries. Periyasamy and Pramanik's and Zhang et al.'s works do not appear to allow the combination of shapes to create more complex shapes via constructive solid geometry (CSG). However, MOSE allows some combination via a union operation. Majaron et al. introduce arbitrary mathematical functions to represent undulating skin layers and some limited shapes (spheres), but again lack any complex geometry via the combination of shapes via CSG. Finally, Glaser et al.'s work is a plugin for GAMOS, a medical-focused framework for GEANT4 (geometry and tracking), which is a platform for the simulation of the passage of particles through matter. GAMOS and GEANT4 both define a large range of shapes allowing the composition of complex models via CSG operations. While these methods of defining mathematical surfaces allow the accurate modeling of smooth surfaces, they have the drawback that each surface needs an accompanying intersection and surface normal routine. These can be computationally costly to evaluate and increase the workload on the programmer.

In this work, we present a novel Monte Carlo radiative transfer model where we eschew the common voxel or mesh-based approaches for an approach based upon signed distance functions (SDFs), which we call signedMCRT (sMCRT). SDFs have been commonly used to define implicit surfaces in computational fluid dynamics,[37,38] computer graphics,[39] video games,[40] and computer vision.[41] Recently, there has been considerable interest in using neural networks to define SDFs from point clouds and meshes. This interest has been led by computer graphics and deep learning researchers, looking for memory-efficient representations of meshes and point clouds at high spatial resolutions.[41–43]

We show in this work that SDFs allow the easier representation of shapes with only the need to define one function for each shape, the distance to surface function. This function allows the computation of intersection and the surface normal to be easily computed with just one function. Several features of SDFs make them an attractive complementary method to voxel and mesh models. SDFs allow the efficient transport of photon packets through the modeled geometry using sphere tracing, which is faster, in many cases, compared with traditional ray tracing methods used in MCRT.[39] SDFs, while being similar to the approach of mathematical surfaces, do not need individual intersection routines as they are naturally included in the SDF definition. Moreover, we can use numerical differentiation to provide the surface normals as SDFs are differentiable almost everywhere and for the special case when an SDF $= 0$, the gradient is the surface normal. SDFs can be incorporated into existing voxel- and mesh-based codes by minor modifications to the optical depth integration and geometry initialization routines. Alternatively, SDFs can be used to define the geometry of voxel- or mesh-based codes as SDFs can be easily rasterized into voxels, and into meshes via the marching cubes algorithm.[44]

We show in this paper that sMCRT is more precise than voxel or mesh models for curved surfaces. We also show that sMCRT is more precise than highly refined meshes in scenarios where reflections and refractions play an important part.

2 Methods

2.1 *Monte Carlo Radiation Transfer Algorithm*

The Monte Carlo radiation method uses interaction probabilities and probability distribution functions that describe the physics of light transport, to model light transport through turbid and nonturbid media. Each photon is propagated a distance τ/μ_t, where τ is the optical distance $[-]$ and μ_t (cm^{-1}) is the extinction coefficient, before it interacts with the medium. The value of τ is sampled from the probability distribution function for the mean free path of a photon using the Monte Carlo method,[18] as shown in Eq. (1), where ξ is a random number drawn in the range [0, 1]

$$\tau = -\log(\xi). \tag{1}$$

The MCRT code presented in this work is broadly based upon previous MCRT codes used in various astronomical, medical, and biophotonics applications.[3,45–47] We use the same routines for releasing photons, input/output, scattering, random number generation, and helper routines. What differs in this work is the optical depth integration routine, and the geometry modeling method, which is accomplished by the use of SDFs.

2.2 *Signed Distance Functions*

SDFs determine the distance from a point p to the boundary of a specified shape. The function returns a positive value if p is outside the boundary, and a negative value if inside the boundary. Formally, this can be described using level set representation. In level set representation, contours are modeled at the zero-level set ($\phi = 0$) of a function defined in a higher dimension. Let $\Phi\colon\Omega \to \mathcal{R}^3$ be a Lipchitz function that refers to a level set representation for a given shape $\mathcal{S}$[48] then

$$\Phi_s(x, y, z) = \begin{cases} 0, & (x, y, z) \in \mathcal{S} \\ +D((x, y, z), \mathcal{S}) > 0, & (x, y, z) \in \mathcal{R}_{\mathcal{S}} \\ D((x, y, z), \mathcal{S}) < 0, & -(x, y, z) \in [\Omega - \mathcal{R}_{\mathcal{S}}] \end{cases} \tag{2}$$

An example of an SDF is shown in Eq. (4) for a sphere, where r is the radius of the sphere, and p is the position of a photon

$$D_{\text{sphere}}(x, y, z) = |p| - r, \tag{3}$$

$$p = [x, y, z]. \tag{4}$$

SDFs can easily be translated, rotated, twisted, and scaled among many other operations. CSG operations such as union, intersection and difference can also be used on the SDFs. Figure 1 shows a subset of shapes and possible operations on SDFs.[51,52]

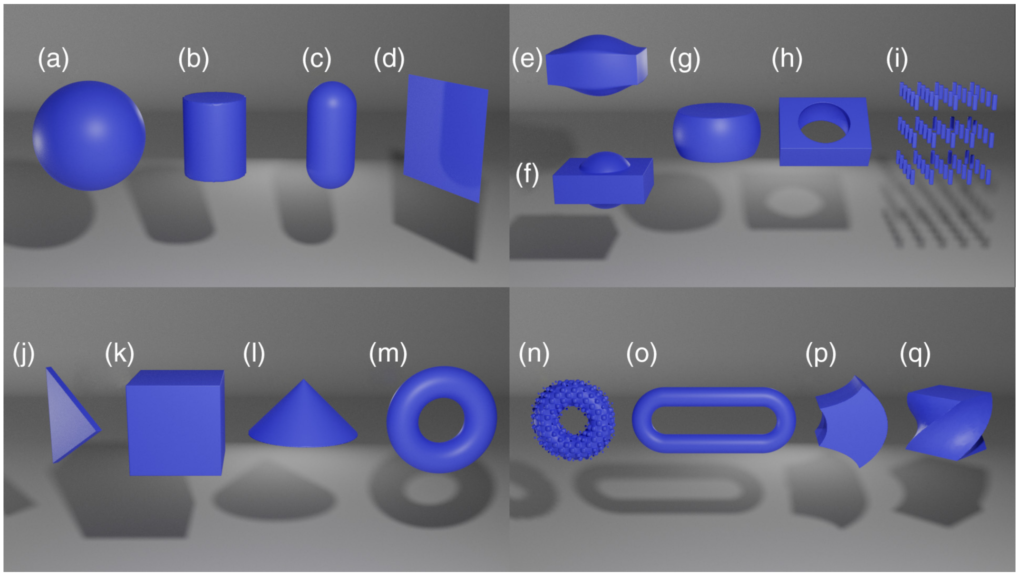

Fig. 1 Several examples of surfaces that can be created by SDFs, rendered in Blender. For illustrative purposes, SDFs are voxelized in sMCRT then transformed to a mesh using Skimage's[49] marching cubes algorithm and then rendered using Blender.[50] The left two panels show a subset of basic shapes calculated using SDFs (a)–(d) and (j)–(m). The right two panels show a subset of possible operations on SDFs: smooth (e) and nonsmooth union (f), intersection (g), subtraction (h), repetition (i), displacement (n), elongation (o), bend (p), and twist (q).

2.3 *sMCRT Algorithm*

To incorporate SDFs into a pre-existing voxel-based MCRT code requires only relativity small adjustments: modifications to the geometry initialization routine and to the optical depth integration routine. An overview of the complete MCRT algorithm is shown in the left panel of Fig. 2.

To create the geometry in voxel or mesh-based models, each voxel or tetrahedral element of the mesh is independently assigned a set of optical properties (scattering and absorption coefficients, refractive index, and anisotropy g value). In sMCRT the geometry is initialized by selecting the functional form, size, and location of SDF(s) required to model the problem, applying any CSG operations required to generate more complex shapes, and finally setting the optical properties for each SDF. Each SDF has its own set of optical properties, which include scattering and absorption coefficient, refractive index, and the anisotropy g value. We then create a bounding box around all the SDFs, which gives us a simulation volume of interest.

In voxel-based MCRT codes, each photon packet is randomly ascribed a specific optical path length that it travels before an interaction, such as scattering or absorption, according to Eq. (1) and is scaled by μ_t (μ_t can be different for each voxel). The photon packet is then propagated through the voxel grid using ray tracing until it reaches that interaction point or leaves the voxel grid.

In our SDF based MCRT algorithm, the first step in the SDF optical depth integration routine is the same as in the voxel case, i.e., randomly assign an optical depth. As before, this is calculated using Eq. (1) and μ_t can be different for each SDF. The next step is to acquire the distance from the current position of the photon packet to the nearest boundary. This is computed by using the SDFs to calculate the distance to each boundary in the modeled geometry and taking the minimum value (d_{sdf}). This process is called sphere tracing[51] and is illustrated in Fig. 3.

If the remaining optical depth for the photon packet is less than d_{sdf}, the photon packet undergoes some interaction, and the optical depth integration routine restarts. If the optical depth is not reached, then we move the full distance d_{sdf}, and then recalculate the distances to all boundaries. If the SDF of the bounding box returns a positive value we are outside the volume of interest, so we terminate the packet and start a new packet.

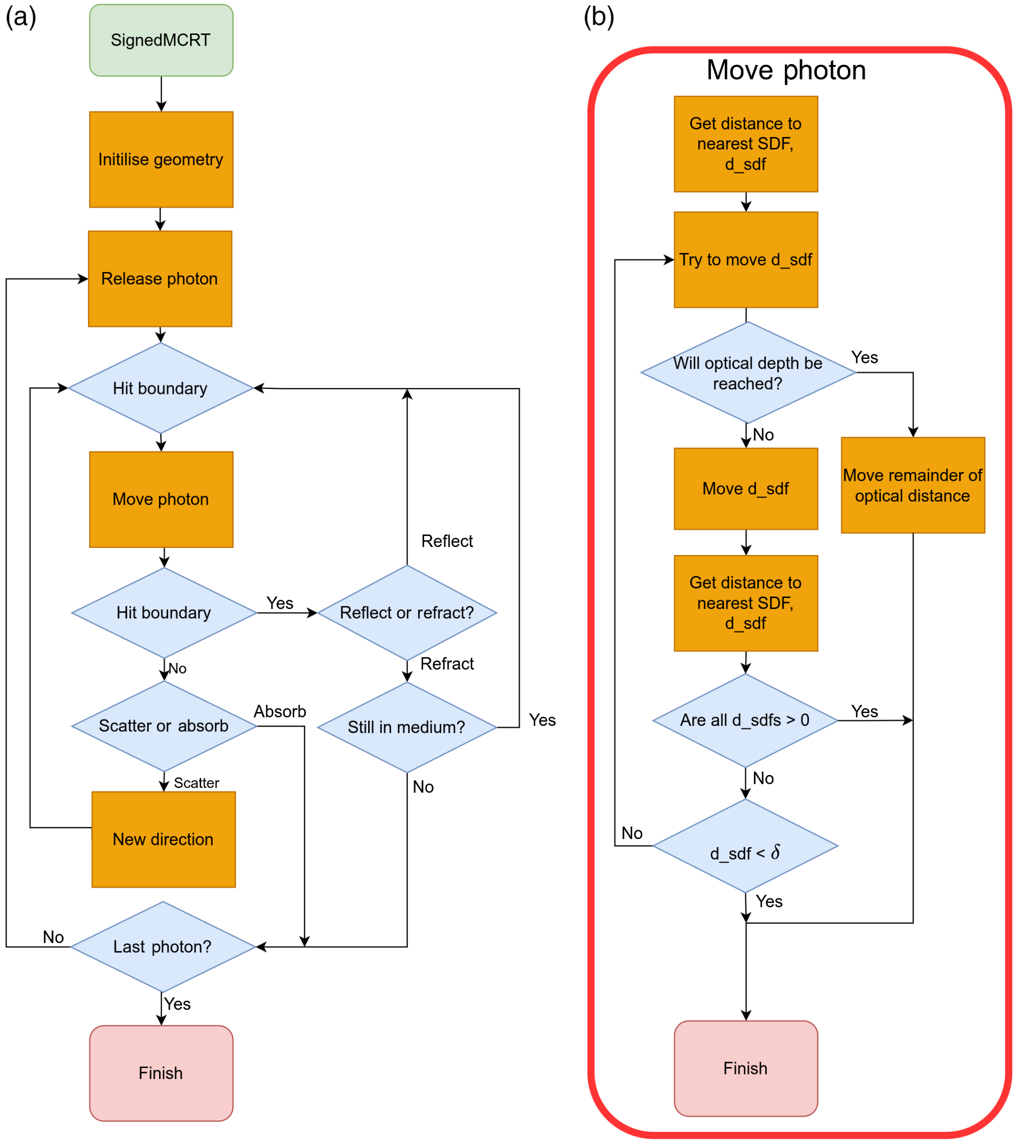

Fig. 2 (a) Flow diagram of an MCRT code. (b) The additional steps needed to incorporate SDFs into the optical depth integration routine, which governs the movement of photon packets through the simulated media.

If the SDF for the bounding box returns a negative value, we then check if the smallest distance, d_{sdf}, is less than some threshold, δ. In this case, the photon packet is on a boundary so we need to check if there is a change in refractive index. If there is a change in refractive index we calculate the Fresnel coefficients and the surface normals, then reflect or refract the photon

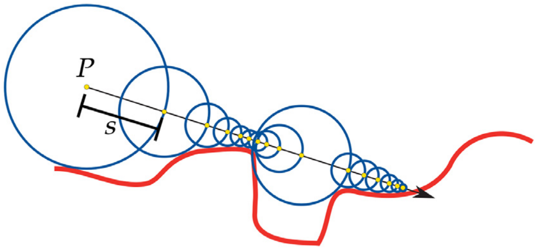

Fig. 3 Example of sphere tracing. Starting a position P, the photon is propagated by a step size, s (represented here by the blue circle), equal to the distance to the nearest surface (red line) until the step size is under some threshold δ.

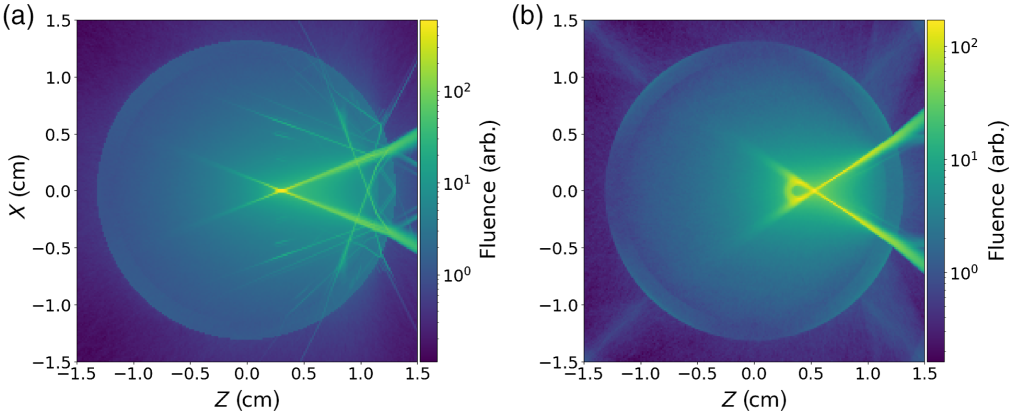

Fig. 4 Comparison of fluence for (a) the voxel model and (b) sMCRT model in a glass bottle. Both panels show the cross section of the bottle. This shows clearly that the voxel model cannot precisely model reflections/refraction in an experiment with curved surfaces, as it shows discrete reflections and refractions, whereas sMCRT shows the expected continuum of reflections and refractions. For this example the optical properties are set for the contents $\mu_s = 2.5$ cm^{-1} and $\mu_a = 0.01$ cm^{-1}. The glass has no scattering and has the same absorption coefficient. The refractive index of the glass is 1.5 and 1.3 for the contents. Both the glass and contents of the bottle have a g value of 0.7. The bottles radius is 1.75 cm, and the glasses thickness is 0.2 cm.

packet. If d_{sdf} is larger than δ, and all distances to the SDFs are not positive then we start this whole process again until one of the exit conditions has been met. Surface normals are calculated using a numerical method based upon central differences, see the Supplemental Material for full details.[53] The above algorithm is shown in Fig. 2(b).

Accurate modeling of curved surfaces is essential for some problems. To illustrate this, Fig. 4 shows a comparison of using voxel-based geometry and SDF-based geometry in our recent work on simulations of Raman spectroscopy of alcoholic beverages through glass bottles.[54]

3 Results and Discussion

3.1 *Validation*

To ensure that our novel SDF-based geometry method works accurately, we validate our algorithm against a theoretical expression and another MCRT code. All simulations are fully parallelized with OpenMP and were run on a workstation with an AMD Ryzen 9 3950X 16-Core Processor with 64 GB RAM using the full 32 threads available.

We first compare sMCRT's accuracy by computing the average number of scattering events occurring to a photon in an isotropic sphere.[55]

For a photon's random walk from the center to the edge of a uniformly scattering sphere of radius r, the average number of scatterings that take place can be written as (see Supplemental Material)

$$N \approx \frac{\tau^2}{2} + \tau. \tag{5}$$

To compare Eq. (5) with sMCRT, we model a sphere of radius 0.5 cm, vary the optical depth between 0.1 and 100 cm^{-1}, and release 10 million photons isotropically from its center. For $\tau_r =$ 100 cm^{-1} typical run-time for sMCRT was ≈42 s compared with ≈64 s for a voxel-based model. The agreement of the code and analytical expression can be seen in Fig. 5.

We also validated sMCRT against Jacques et al. MCRT code.[56] We validate against Jacques code as it incorporates all the relevant physics we need in an MCRT code; scattering, absorption, and refractive index mismatches.

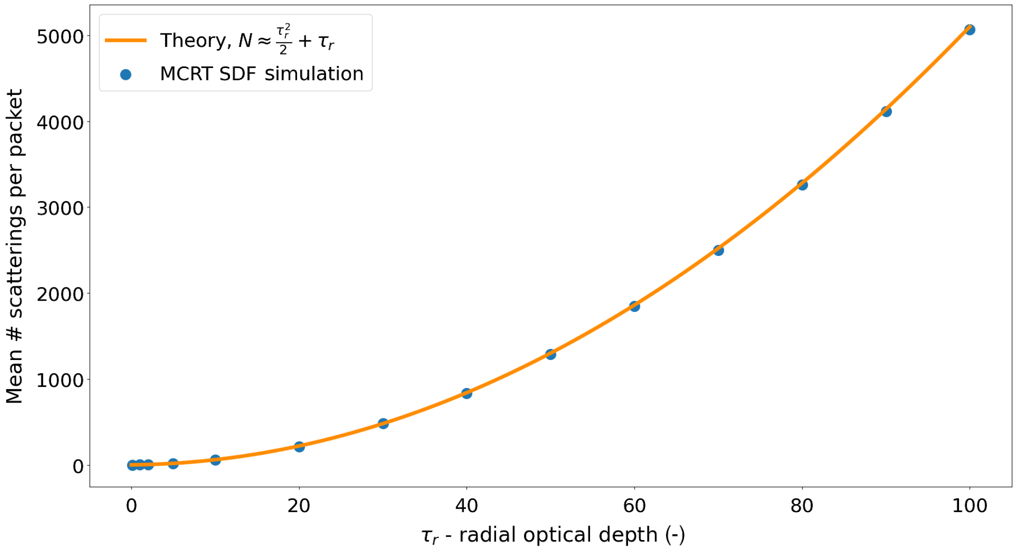

Fig. 5 Agreement of analytical expression [Eq. (5)] and sMCRT for several radial optical depths in the range [0.01, 100]. Photons are released from the center of an isotropic scattering sphere. The optical density (scattering coefficient) is varied and the average number of scatterings per photon packet is recorded.

For this validation, the medium is set up as a semi-infinite slab and light is uniformly incident on the surface of the slab (negative z-direction) and propagates until it is absorbed or escapes via the top surface (positive z-direction). We then fit it against Eq. (6) to compare between codes

$$\Psi(z) = \Psi_0(C_1 e^{-(zk_1/\delta)} - C_2 e^{-(zk_2/\delta)}), \tag{6}$$

where $\Psi(z)$ is the penetration of the incident light or equivalently the fluence rate (W cm^{-2}), Ψ_0 is a normalization constant (W cm^{-2}), C_n and k_n are the fitted coefficients [−], and δ is the optical penetration depth, defined as $\delta = 1/\sqrt{3\mu_a(\mu_a + \mu_s(1-g))}$ (cm). The optical properties for the slab are shown in Table 1, where we use the Henyey–Greenstein phase function[57] with a g of 0.9, and we model two wavelengths in separate simulations. The refractive index for the medium was set to 1.38 to mimic the rat skin used in Jacques code,[56] and for the surrounding medium a refractive index of 1.0 was set, to mimic air. sMCRT took ≈33 s, compared with ≈20 s for the voxel model to run the Jacques test case.

As evidenced in Fig. 6, sMCRT closely matches the results in Jacques et al. An exact match is not possible, due to the difference in the code underlying workings such as cylindrical fluence bin shape used by Jacques et al. versus our rectangular bin shape.

3.2 *Comparing Voxel and Mesh Models to sMCRT*

In this section we compare the accuracy of our new SDF-based MCRT code (sMCRT) to two alternative MCRT methods: a modified MCRT, which uses interpolated surface normals to

Table 1 Table of optical properties and determined coefficients from Jacques et al.[56]

	Absorption	Scattering	Penetration				
Wavelength (nm)	μ_a (cm^{-1})	$\mu_s(1-g)$ (cm^{-1})	C_1	k_1	C_2	k_2	δ (cm^{-1})
420	1.8	82	5.76	1.00	1.31	10.2	0.047
630	0.23	21	6.27	1.00	1.18	14.4	0.261

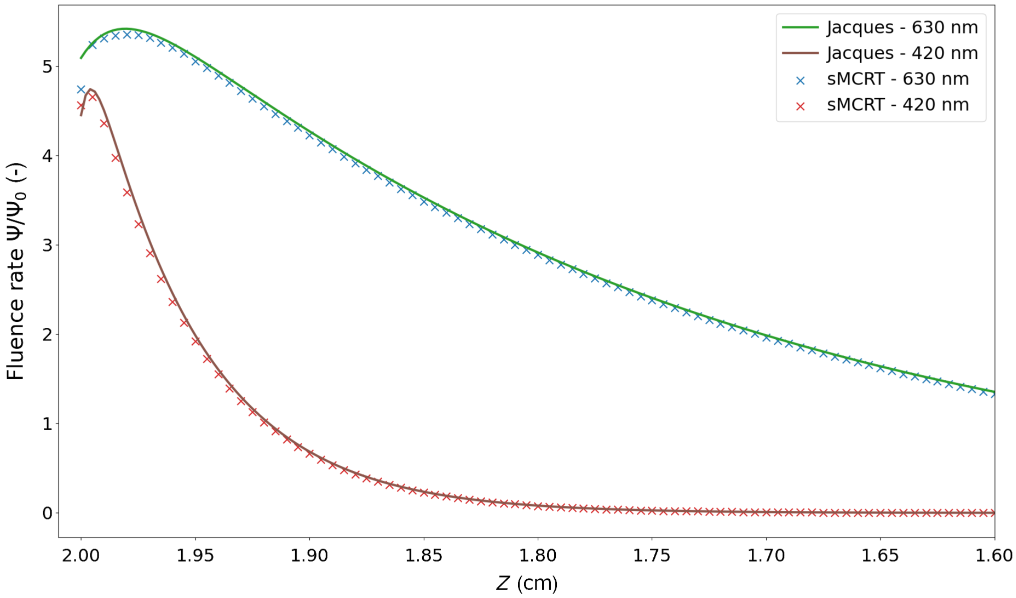

Fig. 6 Validation of sMCRT against Jacques et al. MCRT model. The simulation medium is a semi-infinite slab (infinite in the x and y dimensions), and has the optical properties as in Table 1. The medium is uniformly illuminated via the top surface, i.e., is incident from the left of this figure.

approximate smooth surface;[30] and one of the most widely used mesh-based MCRT methods (MMC)[21] with the most recent BlenderPhotonics[58] plug-in for initialization.

To illustrate that sMCRT is more precise than voxel and mesh-based models, for cases where the geometry can be defined analytically or via the construction of multiple shapes, we devise a test case. The test is a simple one of modeling a smooth surface: for this we use a similar test case to the one used to demonstrate the accuracy by Tran and Jacques surface normal approach seen in reference.[30] For our test, we model a glass sphere with radius 0.75 cm with its centre at $[0.0, -1.0]$ cm and set the refractive index to be 1.33. The glass sphere is set in a medium of air ($n = 1.0$) with cubic size 2 cm centered at the origin [0.0, 0.0, 0.0] cm. We illuminate the geometry with a 2D beam of light of width 0.3 cm propagating along $-z$. Figure 7 shows a slice of the fluence through the sphere for the sMCRT, Tran and Jacques approach, and MMC. Additionally, it shows some rays traced through the sphere as a theoretical comparison for each of the codes.

The surface normal approach (middle top panel of Fig. 7) shows despite accounting for curved surfaces with interpolated surface normals, it still suffers from inaccuracies. These inaccuracies, as evidenced from the missing reflections, arise from their model still being based upon voxels. They interpolate the surface normals at the refractive index mismatches; however, this new virtual surface is not in the correct place for precise photon-surface interactions due to discretization errors.

The middle row shows results from MMC for several mesh refinement levels. All three levels used the same maximum tet volume (0.01): only the refinement level of the input sphere was varied. MMC also displays an imprecise result for all three levels. This is also due to discretization errors much like the surface normal approach by Tran and Jacques. In MMC the mesh is made up of tetrahedrons, where each tetrahedron is made up of triangular faces. Thus, when light is incident on the surface of a tetrahedron it interacts with a planar surface. This discretization can be alleviated to a certain extent by increasing the number of tetrahedrons in the mesh. However, this leads to an increased computational load and memory usage when creating and using the meshes.[25] One further method of accounting for the discretization errors would be to calculate vertex normals when creating the mesh, and then interpolating the vertex normals on the triangular faces, resulting in a smoother appearing surface.[59] To the best of our knowledge, none of the mesh-based MCRT codes surveyed (fullMonte, TIM-OS, and MMC) include this as

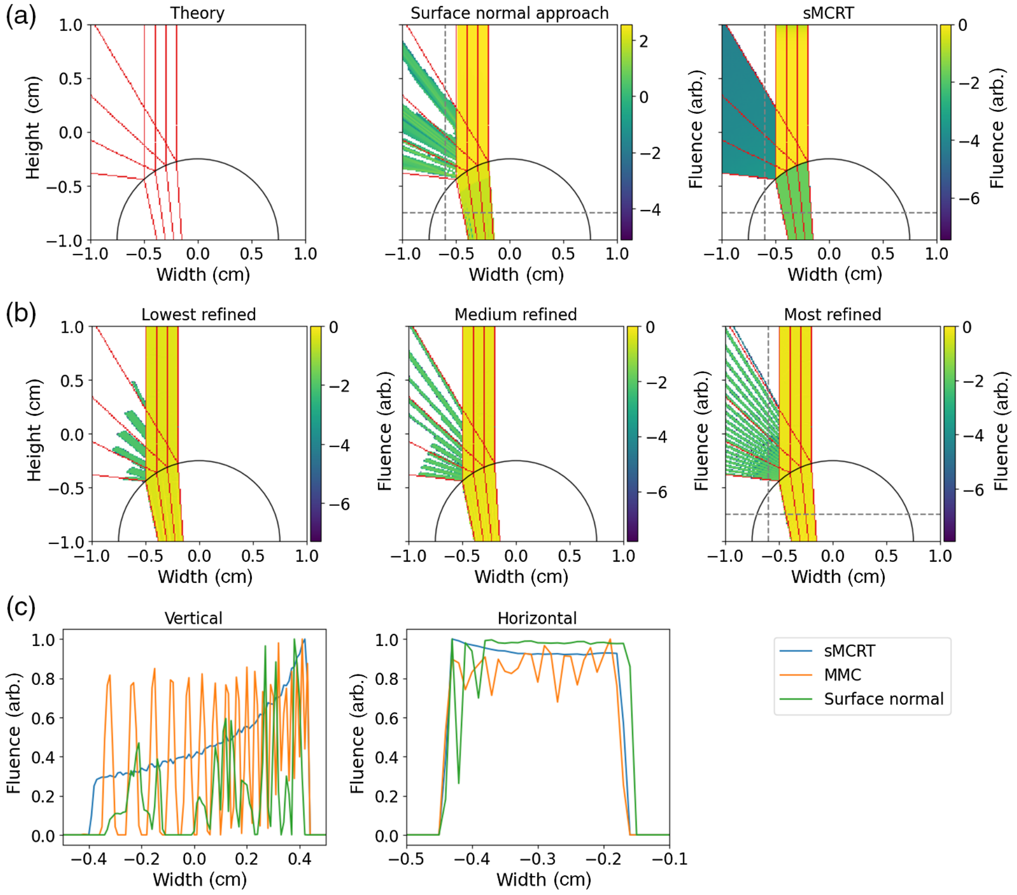

Fig. 7 Comparison of simulation accuracy between A.P Tran and S. Jacques surface normal approach,[30] MMC,[21] sMCRT models, and theory. (a) Left image shows light rays from theory incident on the glass sphere. Top middle shows the results from surface normal approach. Top right shows sMCRT. (b) The output from MMC for different levels of mesh refinement. The lowest refinement contains 34,752 elements and 6137 nodes, the highest refinement has 490,256 elements and 78,976 nodes. (c) A comparison of all three models (with the highest refiment model for MMC) along lines that intersect the reflected light and refracted light (dashed gray lines in the top panels). For this test, sMCRT is clearly more precise than either mesh and modified-voxel-based models. Note, for sMCRT and the surface normal approach we use a one voxel wide slice though the fluence. For MMC we use a three voxel slice though the fluence to account for some of the discretization error.

an option. However, this issue may be overcome with postprocessing or may not be relevant for certain quantities of interest.

sMCRT, for this test case, shows the most accurate result when compared with the theory output. This is due to sMCRT precisely modeling the spherical surface with no discretization errors. The bottom row shows a direct comparison of the accuracies of each model for profiles of the reflected and refracted light (gray dashed lines in some panels). All three models exhibit markedly different reflected light profiles. sMCRT shows the expected slowly rising curve due to more light being reflected by the outer-side of the sphere. Both MMC and the surface normal approach display noisy profiles due to the aforementioned discretization errors. All three models show general agreement in their predictions of the refracted light profile, though MMC and the surface normal approach both exhibit increased noise profiles due to discretization errors. Additionally, the surface normal approach shows an offset profile due to the virtual surface location in relation to the true geometric surface. In the future, it would be interesting to further compare our approach to other modified-voxel-based approaches such as SVMC.[60]

4 Complex Geometry

Thus far we have shown that sMCRT can model simple geometries, so to demonstrate that sMCRT can model complex shapes, we model a blood vessel network embedded in tissue. We also show that sMCRT can model arbitrary SDF generated by neural networks, such as DeepSDF or SIREN, (see Fig. S1 in the Supplemental Material),[61] and model other arbitrary shapes such as the logo of a university after converting a scalable vector graphics image to an SDF (see Fig. S2 in the Supplemental Material).

The vessels are a 3D synthetic microvascular network from data published in Ref. 62 and preprocessed by Yuan et al.[63] Yuan et al.'s data set comprises of the endpoints of cylinders and their radii, thus we can easily convert this data set into an SDF model. We model the slab of tissue using a box SDF (second shape in bottom left panel of Fig. 1), and the vessels as a collection of capsule SDFs (third shape in top left panel of Fig. 1), which are then joined together using the CSG operator union (bottom left operation in top right panel of Fig. 1). The simulation volume is $326 \times 305 \times 611\ \mu\text{m}^3$ and we use a voxel grid of 400^3 to record the fluence. The optical properties of the slab of tissue and the vessels are taken from Ref. 19 and are shown in Table 2. The slab is uniformly illuminated on its top surface by 10 million photons, which are allowed to propagate until they are absorbed or leave the simulated medium. Figure 8 shows the fluence on the vessel network and slices of fluence through the tissue slab.

Table 2 Table of optical properties for the tissue and vessel network.

	Absorption	Scattering		
	μ_a (cm^{-1})	μ_s (cm^{-1})	g	n
Skin	0.459	357	0.9	1.38
Vessels	231	94.0	0.9	1.38

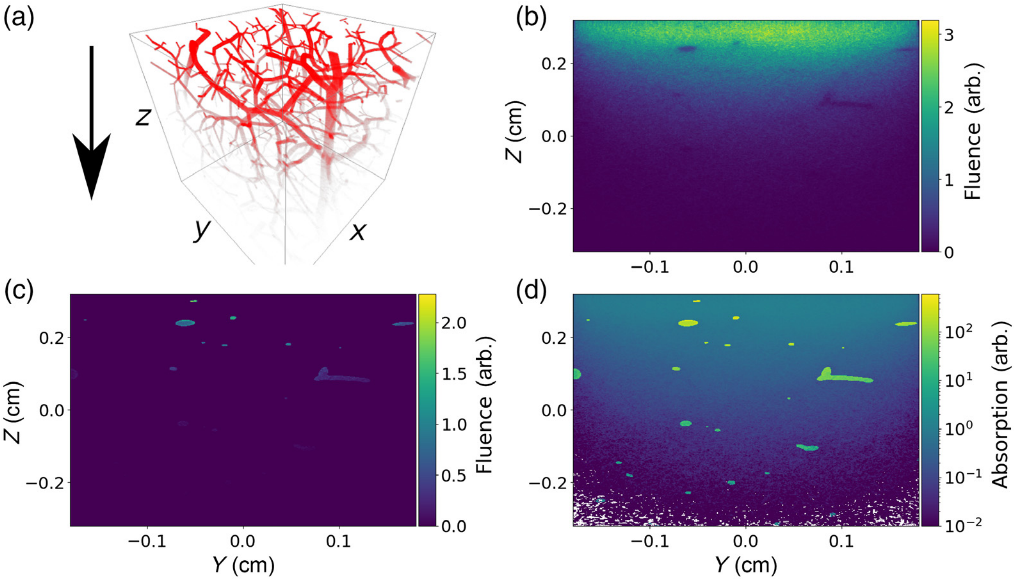

Fig. 8 Fluence and absorption images for the vessel network. Light is uniformly incident on the $X - Y$ plane of a slab of tissue with an embedded vessel network. (a) The 3D fluence for the vessel network with tissue's fluence removed for clarity, arrow indicates direction of incident light. (b) A slice through the fluence for the tissue and vessels in the $Y - Z$ plane. (c) A slice though the fluence for the vessels in the $Y - Z$ plane. (d) A slice of absorption for the tissue and vessels on the $Y - Z$ plane.

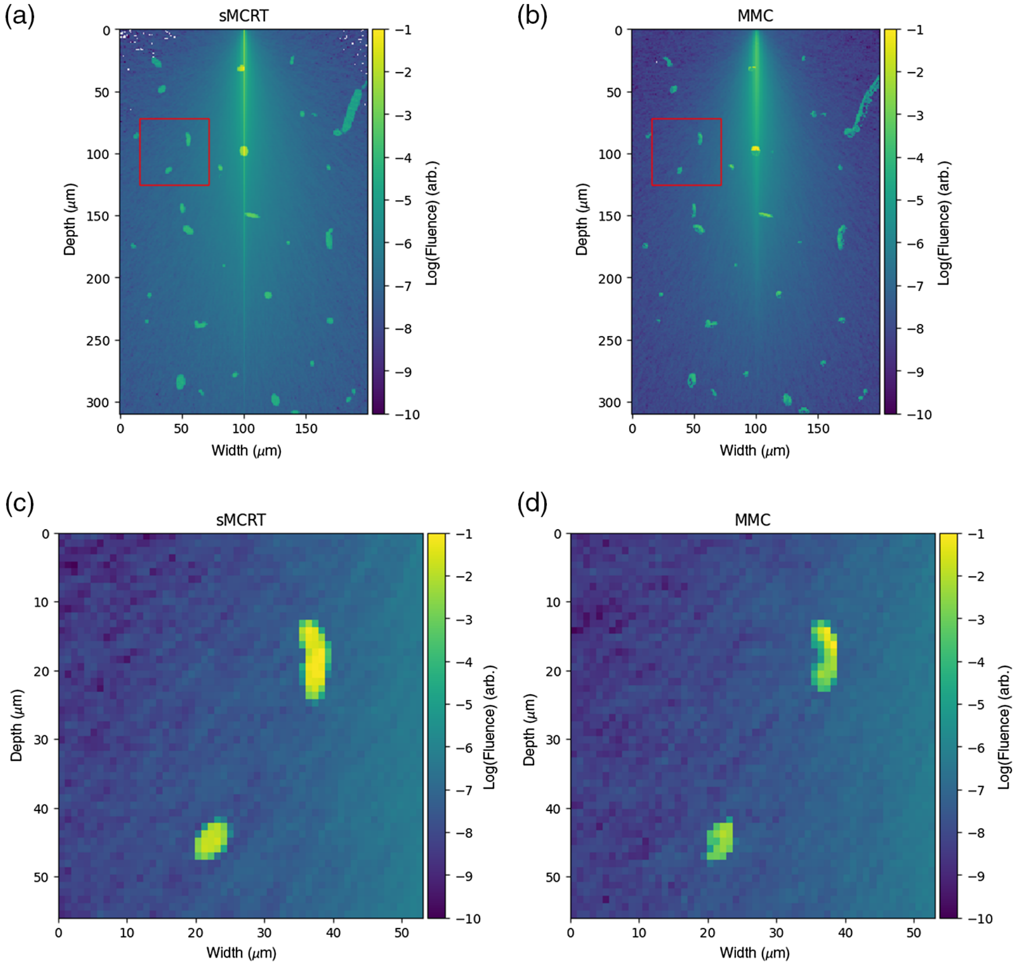

Fig. 9 Comparison of MMC and sMCRT for the complex blood vessel geometry. (a) and (c) sMCRT and (b) and (d) MMC. (a) and (b) Energy absorbed for a slice though the center of the geometry. (c) and (d) The red box indicated in the (a) and (b).

Figure 9 shows a comparison between sMCRT and MMC for the complex blood vessel network. Figure 9(a) and 9(b) show a slice of the fluence in the $x - z$ plane. Both sMCRT and MMC exhibit broadly the same results, with sMCRT's background fluence at a higher level with more noise [Figs. 9(a) and 9(b) of the fluence slice] than MMC's due to the different fluence computation method used (path length estimators[64] compared with Russian roulette weights). This is because the path length estimator "deposits" more energy along its path and is eventually absorbed, whereas the weight system used by MMC deposits less energy and is absorbed in a later point. Therefore, for the same number of photon packets, the results will not match exactly.

5 Conclusion and Outlook

We have shown a meshless, geometrical method for Monte Carlo radiation transport, using SDFs. SDF-based models achieve higher precision than voxel and mesh-based models, particularly for modeling smooth surfaces, such as computing fluence in droplets or accurate modeling of human anatomy for light transport calculations. We envision that SDF-based models provide a complementary method to that of voxel and mesh-based methods for modeling geometry in MCRT simulations.

However, there are a number of potential downsides to using SDFs. In certain configurations, the number of steps needed to be taken by a photon packet can be extremely large, see Fig. S3 in the Supplemental Material. This occurs when the photon is approximately parallel to a surface while the distance between the photon and the surface is small. Recent work has been undertaken to alleviate this problem. This includes segment tracing, which accelerates the sphere tracing method by improving the marching step computation and enhanced sphere tracing, which uses an over-relaxation-based method for accelerating sphere tracing.[65,66]

Large collections of SDFs can also cause massive slowdowns due having to evaluate every SDF each time the photon needs to be moved (i.e., is a global method), which equates to $O(n)$ time complexity where n is the number of SDFs in the geometry. This is analog to the same issue in Monte Carlo models, which use triangular meshes. As in the triangular mesh case, another global method, this can be diminished by using a space-partitioning data structure leading to at best time complexity of $O(\log\ n)$.[67] Tetrahedral meshes intersections are local and therefore are $O(1)$ in time complexity, as each intersection test only needs to evaluate four different face-ray intersections. However, these intersections tests are more frequent as they are a function of mesh refinement. In global methods, the evaluation, in general, is reduced compared with that of local methods. However, global method's evaluation complexity depends on the model complexity whereas local methods do not depend on model complexity.

SDFs are not as general as mesh-based geometries. This means that creating a mesh of a mouse for example is easier than creating an SDF of a mouse. The mesh can be generated from experimental data using an image-based mesh generator.[68] There are no such comparable tools for SDFs currently. The only possible way of using this type of data with SDFs is to use a neural network like SIREN[42] to convert the mesh into an SDF, which can have a high computational cost. Though currently, this is not a precise process (see Fig. S1 in the Supplemental Material). However, neural representation of meshes and point clouds is a highly active topic in Computer Science so this may change in the near future. However, it is not impossible to create complex models with SDFs. As shown, we created a model with a complex vessel network, converted a university logo to an SDF, and converted a mesh to an SDF. Furthermore, there are hundreds of examples of complex SDF model in the field of computer graphics, where programmers/artists have created complex animated scenes using SDFs alone.[69–71] Some recent work in the field of computer graphics has created an open data set of complex SDF models for research purposes.[72]

The final potential issue is the combination of multiple CSG operations can lead to nonbounded SDFs. Nonbounded SDFs can pose a problem in terms of accuracy and speed.[73] In terms of speed, nonbounded SDFs only give a conservative distance to the surface, resulting in more SDF evaluations, which can cause computational slowdown. The accuracy problem only affects the computation of surface normals and is therefore only applicable at refractive index interfaces. Despite these potential drawbacks, we envision MCRT codes using SDFs to model the geometry to probe problems such as the effect of skin color on pulse oximetry accuracy, fluence calculation of droplets with viral loads such as COVID-19, and accurate simulations of light propagation in fruit. In these problems, using SDF over voxels would allow precise modeling of curved surfaces allowing better accuracy in the simulations. We believe that SDF-based MCRT models will occupy the position between voxel and mesh-based MCRT models in terms of being able to model complex geometries and accurately treat curved surfaces but with the caveat that currently producing SDF models of experimental data remains challenging.

Disclosures

No conflicts of interest, financial or otherwise, are declared by the authors.

Acknowledgment

The work was supported by funding from the UK Engineering and Physical Sciences Research Council (EP/R004854/1) and the European Union H2020 projects "Dynamic" (EC-GA 863203) and "Proscope" (871212). KD acknowledges support of the Australian Research Council through a Laureate Fellowship.

Data, Materials, and Code Availability

All code is publicly available at: https://github.com/lewisfish/signedMCRT. Data is available at https://doi.org/10.5281/zenodo.5780513.

References

1. C. L. Campbell et al., "Monte Carlo modelling of daylight activated photodynamic therapy," *Phys. Med. Biol.* **60**(10), 4059 (2015).
2. I. R. M. Barnard et al., "Could psoralen plus ultraviolet A1 ("PUVA1") work? Depth penetration achieved by phototherapy lamps," *Br. J. Dermatol.* **182**(3), 813–814 (2020).
3. L. McMillan et al., "Imaging in thick samples, a phased Monte Carlo radiation transfer algorithm," *J. Biomed. Opt.* **26**(9), 096004 (2021).
4. A. Lyons et al., "Computational time-of-flight diffuse optical tomography," *Nat. Photonics* **13**(8), 575–579 (2019).
5. J. Qin and R. Lu, "Monte Carlo simulation for quantification of light transport features in apples," *Comput. Electron. Agric.* **68**(1), 44–51 (2009).
6. W. M. Star, "Diffusion theory of light transport," in Optical-*Thermal Response of Laser-Irradiated Tissue*, A. J. Welch and M. J. C. Van Gemert, eds., pp. 131–206, Springer (1995).
7. M. D. Keller et al., "Monte Carlo model of spatially offset Raman spectroscopy for breast tumor margin analysis," *Appl. Spectrosc.* **64**(6), 607–614 (2010).
8. S. Wang et al., "Monte Carlo simulation of in vivo Raman spectral measurements of human skin with a multi-layered tissue optical model," *J. Biophotonics* **7**(9), 703–712 (2014).
9. Q. Liu, C. Zhu, and N. Ramanujam, "Experimental validation of Monte Carlo modeling of fluorescence in tissues in the UV-visible spectrum," *J. Biomed. Opt.* **8**(2), 223–236 (2003).
10. A. J. Welch et al., "Propagation of fluorescent light," *Lasers Surg. Med.* **21**(2), 166–178 (1997).
11. S. L. Reidt et al., "Polarised light sheet tomography," *Opt. Express* **24**(10), 11239–11249 (2016).
12. J. C. Ramella-Roman, S. A. Prahl, and S. L. Jacques, "Three Monte Carlo programs of polarized light transport into scattering media: part I," *Opt. Express* **13**(12), 4420–4438 (2005).
13. J. C. Ramella-Roman, S. A. Prahl, and S. L. Jacques, "Three Monte Carlo programs of polarized light transport into scattering media: part II," *Opt. Express* **13**(25), 10392–10405 (2005).
14. B. Vandenbroucke and K. Wood, "The Monte Carlo photoionization and moving-mesh radiation hydrodynamics code CMacIonize," *Astron. Comput.* **23**, 40–59 (2018).
15. T. J. Harries et al., "The TORUS radiation transfer code," *Astron. Comput.* **27**, 63–95 (2019).
16. L. McMillan et al., "Development of a predictive Monte Carlo radiative transfer model for ablative fractional skin lasers," *Lasers Surg. Med.* **53**(5), 731–740 (2021).
17. J. C. G. Jeynes et al., "Monte Carlo simulations of heat deposition during photothermal skin cancer therapy using nanoparticles," *Biomolecules* **9**(8), 343 (2019).
18. 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).
19. D. Marti et al., "MCmatlab: an open-source, user-friendly, MATLAB-integrated three-dimensional Monte Carlo light transport solver with heat diffusion and tissue damage," *J. Biomed. Opt.* **23**(12), 121622 (2018).
20. D. A. Boas et al., "Three dimensional Monte Carlo code for photon migration through complex heterogeneous media including the adult human head," *Opt. Express* **10**, 159–170 (2002).
21. Q. Fang, "Mesh-based Monte Carlo method using fast ray-tracing in Plücker coordinates," *Biomed. Opt. Express* **1**(1), 165–175 (2010).
22. R. H. Wilson et al., "Mesh-based Monte Carlo code for fluorescence modeling in complex tissues with irregular boundaries," *Proc. SPIE* **8090**, 80900E (2011).
23. T. Binzoni et al., "Light transport in tissue by 3d Monte Carlo: influence of boundary voxelization," *Comput. Methods Programs Biomed.* **89**(1), 14–23 (2008).

24. R. Sawhney and K. Crane, "Monte Carlo geometry processing: a grid-free approach to PDE-based methods on volumetric domains," *ACM Trans. Graph.* **39**(4), 1–18 (2020).
25. Y. Hu et al., "Fast tetrahedral meshing in the wild," *ACM Trans. Graph.* **39**(4), 1–8 (2020).
26. H. Shen and G. Wang, "A tetrahedron-based inhomogeneous Monte Carlo optical simulator," *Phys. Med. Biol.* **55**(4), 947 (2010).
27. T. Young-Schultz et al., "FullMonteCUDA: a fast, flexible, and accurate GPU-accelerated Monte Carlo simulator for light propagation in turbid media," *Biomed. Opt. Express* **10**(9), 4711–4726 (2019).
28. H. Shen and G. Wang, "A study on tetrahedron-based inhomogeneous Monte Carlo optical simulation," *Biomed. Opt. Express* **2**(1), 44–57 (2011).
29. C. W. Van Overveld and B. Wyvill, "Phong normal interpolation revisited," *ACM Trans. Graph.* **16**(4), 397–419 (1997).
30. A. P. Tran and S. L. Jacques, "Modeling voxel-based Monte Carlo light transport with curved and oblique boundary surfaces," *J. Biomed. Opt.* **25**(2), 025001 (2020).
31. V. Periyasamy and M. Pramanik, "Monte Carlo simulation of light transport in turbid medium with embedded object—spherical, cylindrical, ellipsoidal, or cuboidal objects embedded within multilayered tissues," *J. Biomed. Opt.* **19**(4), 045003 (2014).
32. B. Majaron, M. Milanič, and J. Premru, "Monte Carlo simulation of radiation transport in human skin with rigorous treatment of curved tissue boundaries," *J. Biomed. Opt.* **20**(1), 015002 (2015).
33. H. Li et al., "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).
34. Y. Zhang et al., "Efficient and accurate simulation of light propagation in bio-tissues using the three-dimensional geometric Monte Carlo method," *Numer. Heat Transfer A: Appl.* **68**(8), 827–846 (2015).
35. A. K. Glaser et al., "A GAMOS plug-in for GEANT4 based Monte Carlo simulation of radiation-induced light transport in biological media," *Biomed. Opt. Express* **4**(5), 741–759 (2013).
36. S. Ren et al., "Molecular optical simulation environment (MOSE): a platform for the simulation of light propagation in turbid media," *PloS One* **8**(4), e61304 (2013).
37. M. Sakai et al., "Recent progress on mesh-free particle methods for simulations of multiphase flows: a review," *KONA Powder Particle J.* **37**, 132–144 (2020).
38. A. Roosing, O. Strickson, and N. Nikiforakis, "Fast distance fields for fluid dynamics mesh generation on graphics hardware," arXiv:1903.00353 (2019).
39. J. C. Hart, D. J. Sandin, and L. H. Kauffman, "Ray tracing deterministic 3-D fractals," in *Proc. 16th Annu. Conf. Comput. Graph. and Interactive Tech.*, pp. 289–296 (1989).
40. A. Evans, "Learning from failure: a survey of promising, unconventional and mostly abandoned renderers for 'Dreams PS4', a geometrically dense, painterly UGC game," ACM SIGGRAPH Course Notes (2015).
41. J. J. Park et al., "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conf. Comput. Vis. and Pattern Recognit.*, pp. 165–174 (2019).
42. V. Sitzmann et al., "Implicit neural representations with periodic activation functions," in *Adv. Neural Inf. Process. Syst.*, Vol. 33 (2020).
43. T. Takikawa et al., "Neural geometric level of detail: Real-time rendering with implicit 3d shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. and Pattern Recognit.*, pp. 11358–11367 (2021).
44. W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," *ACM SIGGRAPH Comput. Graph.* **21**(4), 163–169 (1987).
45. K. Wood and R. J. Reynolds, "A model for the scattered light contribution and polarization of the diffuse H galactic background," *Astrophys. J.* **525**(2), 799 (1999).
46. I. R. M. Barnard et al., "Quantifying direct DNA damage in the basal layer of skin exposed to UV radiation from sunbeds," *Photochem. Photobiol.* **94**(5), 1017–1025 (2018).
47. I. R. M. Finlayson et al., "Depth penetration of light into skin as a function of wavelength from 200 nm-1000 nm," *Photochem. Photobiol.* **12**(10), 99–100 (2021).

48. N. Paragios, M. Rousson, and V. Ramesh, "Matching distance functions: a shape-to-area variational approach for global-to-local registration," *Lect. Notes Comput. Sci.* **2351**, 775–789 (2002).
49. S. Van der Walt et al., "scikit-image: image processing in python," *PeerJ* **2**, e453 (2014).
50. Blender Online Community, *Blender - A 3D Modelling and Rendering Package*, Blender Foundation, Blender Institute, Amsterdam (2021).
51. J. C. Hart, "Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces," *Visual Comput.* **12**(10), 527–545 (1996).
52. I. Quilez, "Distance functions," https://iquilezles.org/www/articles/distfunctions/distfunctions.htm.
53. I. Quilez, "Normals for an SDF," https://iquilezles.org/www/articles/normalsSDF/normalsSDF.htm.
54. G. E. Shillito et al., "To focus-match or not to focus-match inverse spatially offset Raman spectroscopy: a question of light penetration," *Opt. Express* **30**, 8876–8888 (2022).
55. G. B. Rybicki and A. P. Lightman, *Radiative Processes in Astrophysics*, John Wiley & Sons (1991).
56. S. L. Jacques, R. Joseph, and G. Gofstein, "How photobleaching affects dosimetry and fluorescence monitoring of PDT in turbid media," *Proc. SPIE* **1881**, 168–180 (1993).
57. L. G. Henyey and J. L. Greenstein, "Diffuse radiation in the galaxy," *Astrophys. J.* **93**, 70–83 (1941).
58. Y. Zhang and Q. Fang, "BlenderPhotonics – a versatile environment for 3-d complex bio-tissue modeling and light transport simulations based on blender," bioRxiv (2022).
59. I. Gkioulekas, "Ray tracing and geometric representations, lecture 2," http://graphics.cs.cmu.edu/courses/15-468/lectures/lecture2.pdf.
60. S. Yan and Q. Fang, "Hybrid mesh and voxel based monte carlo algorithm for accurate and efficient photon transport modeling in complex bio-tissues," *Biomed. Opt. Express* **11**(11), 6262–6270 (2020).
61. G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proc. 21st Annu. Conf. Comput. Graph. and Interactive Tech.*, pp. 311–318 (1994).
62. G. Tetteh et al., "Deepvesselnet: vessel segmentation, centerline prediction, and bifurcation detection in 3-D angiographic volumes," *Front. Neurosci.* **14**, 1285 (2020).
63. Y. Yuan, S. Yan, and Q. Fang, "Light transport modeling in highly complex tissues using the implicit mesh-based Monte Carlo algorithm," *Biomed. Opt. Express* **12**(1), 147–161 (2021).
64. L. B. Lucy, "Computing radiative equilibria with Monte Carlo techniques," *Astron. Astrophys.* **344**, 282–288 (1999).
65. E. Galin et al., "Segment tracing using local Lipschitz bounds," *Comput. Graph. Forum* **39**, 545–554 (2020).
66. B. Keinert et al., "Enhanced sphere tracing," *Smart Tools Apps Graph.* **8**(4) (2014).
67. A. Watt, "3D computer graphics," (1999).
68. Q. Fang and D. R. Kaeli, "Accelerating mesh-based Monte Carlo method on modern CPU architectures," *Biomed. Opt. Express* **3**(12), 3223–3230 (2012).
69. I. Quilez, "Slisesix," https://www.shadertoy.com/view/NtlSDs.
70. F. Berger, "zztop '33 ford eliminator," https://www.shadertoy.com/view/tltGWj.
71. I. Quilez, "Selfie girl," https://www.shadertoy.com/view/WsSBzh.
72. T. Takikawa, A. Glassner, and M. McGuire, "A dataset and explorer for 3D signed distance functions," *J. Comput. Graph. Tech.* **11**(2), 1–29 (2022).
73. I. Quilez, "Interior SDFs," https://iquilezles.org/www/articles/interiordistance/interiordistance.htm.

Lewis McMillan is a post-doctoral research fellow in physics at the University of St Andrews. He holds a PhD in computational physics from the University of St Andrews. His interests lie in using code to solve various research problems in the fields of marine biology, biophotonics, optics, medicine, and physics.

Biographies of the other authors are not available.