Unifying radiative transfer models in computer graphics and remote sensing, Part I: A survey
Katherine Salesin a,*, Kirk D. Knobelspiesse b, Jacek Chowdhary c, Peng-Wang Zhai d, Wojciech Jarosz a

a Department of Computer Science, Dartmouth College, 15 Thayer Dr, Hanover, NH, 03755, USA
b Ocean Ecology Laboratory, GSFC/NASA, Code 616, Greenbelt, MD, 20771, USA
c Department of Applied Physics and Applied Mathematics, Columbia University, 2880 Broadway, New York, NY, 10025, USA
d Department of Physics, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD, 21250, USA

A R T I C L E  I N F O
Keywords:
Radiative transfer
Computer graphics
Remote sensing
Atmospheric and ocean optics
Polarization
Monte Carlo

A B S T R A C T
The constellation of Earth-observing satellites continuously collects measurements of scattered radiance, which must be transformed into geophysical parameters in order to answer fundamental scientific questions about the Earth. Retrieval of these parameters requires highly flexible, accurate, and fast forward and inverse radiative transfer models. Existing forward models used by the remote sensing community are typically accurate and fast, but sacrifice flexibility by assuming the atmosphere or ocean is composed of plane-parallel layers. Monte Carlo forward models can handle more complex scenarios such as 3D spatial heterogeneity, but are relatively slower. We propose looking to the computer graphics community for inspiration to improve the statistical efficiency of Monte Carlo forward models and explore new approaches to inverse models for remote sensing. In Part 1 of this work, we examine the evolution of radiative transfer models in computer graphics and highlight recent advancements that have the potential to push forward models in remote sensing beyond their current periphery of realism.

1. Introduction
Light collected by sensors observing the Earth’s atmosphere and oceans can provide enormous insight into the chemical, physical, and biological processes occurring within. In order to disentangle the contributions of such processes to sensor measurements, accurate simulations of how light interacts with real-world phenomena must be developed. To illustrate the immensity of this task, consider that fully accurate simulations of atmospheric–ocean systems must account for scattering and absorption by gases, water droplets, and particulates (aerosols) in the atmosphere; scattering and absorption by gases, liquids, particulates, and organisms in the ocean; scattering at the ocean surface, which may be sprinkled with white caps or sea ice; as well as fluorescence and phosphorescence caused by biological processes. Simulations must also match the capabilities of the sensor to capture angular, spectral, and polarized properties of incoming light. Lastly, these simulations must be computationally efficient — although there is some leeway to sacrifice speed for complexity or vice versa.

In order to be useful to the scientific community, sensor measurements of light must be transformed via simulation into atmospheric or oceanic parameters such as ocean surface roughness; chlorophyll-a concentration in the ocean; phase, height, optical thickness, and droplet radius of clouds; and shape, refractive index, optical depth, and vertical distribution of aerosols [1]. The retrieval of all parameters at once for a snapshot of place and time would be ideal, as it would capture their optically interrelated nature. However, this goal is not currently achievable due to both insufficient information per “snapshot” and the limitations of existing retrieval algorithms. More information can be gathered by developing more accurate or more advanced hardware, e.g. sampling more wavelengths in the visible, ultraviolet, and infrared range, sampling multiple viewing angles of the Earth, or detecting the polarization state and angle of incoming light. Many of these enhancements will be added to the instruments on NASA’s upcoming Plankton, Aerosol, Cloud, ocean Ecosystem (PACE) mission [2]: the Ocean Color Instrument (OCI) [3], a hyperspectral radiometer, and the Spectro-Polarimeter for Planetary Exploration (SPEXone) [4,5] and Hyper Angular Rainbow Polarimeter (HARP2) [5,6], multi-angle polarimeters of different angular and spectral resolutions. In order to harness the unprecedented volume and richness of incoming data to

* Corresponding author.
E-mail addresses: katherine.a.salesin.gr@dartmouth.edu (K. Salesin), kirk.d.knobelspiesse@nasa.gov (K.D. Knobelspiesse), jacek.chowdhary@nasa.gov (J. Chowdhary), pwzhai@umbc.edu (P. Zhai), wojciech.k.jarosz@dartmouth.edu (W. Jarosz).

https://doi.org/10.1016/j.jqsrt.2023.108847
Received 24 February 2023; Received in revised form 17 November 2023; Accepted 19 November 2023
Available online 29 November 2023
0022-4073/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
its full potential, more advanced remote sensing algorithms must be developed.

Toward this goal, we look to a fresh source of inspiration: the field of computer graphics. Since the dawn of the computer era, the field of computer graphics has studied how to recreate what we see virtually, building upon the prior insights of artists and scientists over hundreds of years. In pursuit of "photorealism", or computer-generated images indistinguishable from a photograph, computer graphics has evolved toward extremely realistic, physically based models. While applications in the past have typically been commercial (e.g. visual effects for movies and video games, artistic creation tools, virtual reality), the underlying Monte-Carlo-based radiative transfer models have finally reached the threshold of complexity and accuracy required for scientific applications. However, as of now, there is relatively little cross-pollination between radiative transfer literature in computer graphics and other scientific fields such as optics, astronomy, neutron transport, atmospheric science, oceanography, and remote sensing.

In Part 1 of this work, we create a Rosetta Stone for two scientific communities that have been working in parallel for decades. We review the radiative transfer models used in remote sensing and current capabilities of their existing frameworks in Section 2, and likewise for computer graphics in Section 3. While both communities use Monte Carlo models for radiative transfer, they are more heavily favored by the computer graphics community, who has spent decades of research improving their statistical efficiency via sophisticated sampling techniques — we highlight areas that are pertinent to remote sensing in Section 4.

In Part 2 of this work, we extend a forward and inverse modeling framework recently developed in the computer graphics community, Mitsuba 3 [7], to perform simulations of interest to remote sensing and test our framework on recent benchmark tests of simple atmosphere-ocean systems.

Through both the literature survey in Part 1 and the benchmark tests in Part 2, we demonstrate that Monte Carlo models in computer graphics can not only match the accuracy of those in remote sensing, but have yet-unexplored potential to model even greater complexity of real-world environments and improve efficiency.

2. Review of forward and inverse radiative transfer models

To prime our survey of relevant computer graphics literature, we first review the priorities of each community with respect to radiative transfer models and their current capabilities. A radiative transfer model is a mathematical and algorithmic framework for conducting realistic simulations of light. Both forward and inverse radiative transfer models are necessary in order to transform sensor measurements into other parameters of interest (see Fig. 1). A forward radiative transfer model maps state space to measurement space, mimicking the same transformation as a real-world sensor. State space encompasses all geophysical parameters such as scattering and absorption properties as well as scene composition parameters such as sensor angle, sun angle, geometry, etc., while measurement space encompasses all possible measurements of radiance that the sensor could produce, e.g. the space of all photos a camera can capture.

Naturally, an inverse radiative transfer model maps measurement space to state space, the desired transformation for remote sensing retrievals. Inverse models typically rely on embedded forward models to produce intermediate results for optimization.

Retrieval of atmosphere and ocean parameters requires forward and inverse radiative transfer models that are flexible, accurate, and fast. Flexible models are able to represent realistic light-based phenomena of arbitrary complexity, while accurate models are able to estimate radiance precisely under the assumptions of the underlying model. Speed is also important in order to deliver data of interest promptly to the scientific and applications community for further analysis. Typically, there exists a tradeoff between these desired qualities: favoring one requires sacrificing another to some extent.

2.1. Review of forward radiative transfer models

The remote sensing community typically favors accurate and fast forward radiative transfer models that use deterministic solutions to a sub-problem of the classical radiative transfer equation (RTE), such as adding-doubling [8], successive orders of scattering [9,10], or the discrete-ordinate method [11]. We refer the reader to Chowdhary et al. [12] and references therein for a recent and comprehensive survey of such models available for remote sensing.

The accuracy of these models is controlled by the number of quadrature points used to numerically estimate the solutions of lower-dimensional integrals. The lower dimensionality of the new integral compared to the original RTE is critical because numerical integration suffers from the curse of dimensionality, in which the convergence rate of the estimate becomes exponentially slower as the dimension of the integral increases. While these methods may be able to handle complex effects such as polarization or fluorescence, they sacrifice flexibility by assuming the atmosphere or ocean is composed of plane-parallel layers, a stack of layers bounded by parallel planes whose properties are homogeneous within each layer. However, the assumption of plane-parallel layers breaks down both in the small scale (e.g. clouds are fluffy and organically shaped) and the large scale (e.g. the Earth is spherical).

An alternate solution to the RTE that is more general and does not suffer from the curse of dimensionality is a stochastic solution called Monte Carlo integration, which is heavily favored by the computer graphics community today. Monte Carlo models are able to handle more geometric complexity and spatial heterogeneity. One remaining assumption is geometric optics, in which light is modeled by rays traveling along straight paths that are diverted by scatter events. Geometric optics is compatible with some wave-based effects of light (e.g. polarization) but not others (e.g. diffraction) since it does not account for electromagnetic phase, although some phase-aware effects can be accounted for locally (e.g. Lorenz-Mie scattering [13,14]). Recent research in computer graphics has begun deconstructing this assumption to faithfully represent the wave nature of light in existing Monte Carlo frameworks [15–17].
Monte Carlo models are generally viewed as too computationally expensive for operational processing of satellite datasets, although some have been developed in remote sensing primarily for offline generation of data for look-up tables (see Chowdhary et al. [12] for a summary). Notably, some of the earliest atmosphere-ocean radiative transfer models developed in the late 1960s/early 1970s by Kattawar and Plass [18–21] were Monte-Carlo-based. However, Monte Carlo methods have progressed significantly in both accuracy and speed in the past two decades in computer graphics [22], partly due to the meteoric rise in computing power and partly due to research in physically based rendering (“to render” is to run a forward radiative transfer model). Some developments from computer graphics have just very recently trickled into Monte Carlo models in the remote sensing community, namely parallelization on the GPU and light/path tracing with next-event estimation [23–31] (notably, [27,30] use an earlier version of Mitsuba for forward modeling of clouds). However, they have not delved far into the huge library of variance reduction methods developed for computer graphics, like those we will highlight in Section 4. To prime further discussion, let us briefly review the basics of Monte Carlo. Monte Carlo integration approximates the integral of a function, \( F := \int_D f(x) \, dx \), by averaging the function \( f \) evaluated at a number of randomly sampled locations \( x_i \) over the domain \( D \):

\[
(F) := \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}
\]

As long as the probability density (PDF) of the samples \( p(x) > 0 \) for any location within the domain \( D \) where \( f(x) \neq 0 \), then the estimator \( \langle F \rangle \) is unbiased, i.e. its expected value \( \mathbb{E}(F) = F \), but the estimator will still exhibit error due to noise, or random fluctuations. We can tame these fluctuations by averaging more samples, which should bring the estimate closer to the true solution. We can quantify error due to noise by the estimator’s variance, which expresses how far each sample is from the expected value, on average:

\[
\mathbb{V} \left[ \langle F \rangle \right] = \mathbb{E} \left[ \left( \langle F \rangle - \mathbb{E}(\langle F \rangle) \right)^2 \right].
\]

Plugging Eq. (1) into Eq. (2) gives:

\[
\mathbb{V} \left[ \langle F \rangle \right] = \frac{1}{N^2} \sum_{i=1}^{N} \mathbb{V} \left[ \frac{f(x_i)}{p(x_i)} \right] = \frac{1}{N} \mathbb{V} \left[ \frac{f(x)}{p(x)} \right],
\]

where the first step assumes the samples \( x_i \) are independent, and the second step assumes they are all drawn from the same PDF. This leads to the key observation that the variance (squared error) of a Monte Carlo estimator depends on the number of averaged samples \( N \) and the variance of the quotient \( f(x) / p(x) \). We can reduce variance by increasing \( N \) or by choosing a \( p(x) \) that more closely matches the shape of \( f(x) \) across its domain. Ideally, when \( p(x) \) is perfectly proportional to \( f(x) \), \( p(x) \sim f(x) \), the variance \( \mathbb{V} \left[ \frac{f(x)}{p(x)} \right] \) is zero, and so is that of the estimator.

Monte Carlo radiative transport codes approximate the integrals within the RTE using estimators like Eq. (1). In this setting, finding a neatly proportional \( p(x) \) is quite hard since the RTE is the composition of many nested and recursive functions (see Part 2, Eqs. (3) and (5)). For example, the amount of light arriving at a point in space is the sum of many paths scattered throughout the scene and influenced by the materials and geometry of everything they have touched! Crafting a probability distribution well-suited to its integrand is called importance sampling, and is a key research concept in computer graphics. Many examples are given in Section 4.

2.2. Review of inverse radiative transfer models

For a comprehensive historical perspective on inverse radiative transfer models for remote sensing, we refer the reader to Chowdhary et al. [12], Frouin et al. [32] and Remer et al. [3,5] (for aerosol retrievals in particular). Available solutions fall roughly into four classes:

- **Heuristic models** use simple formulas to retrieve a single parameter based on its known properties, e.g. relating chlorophyll-a concentration to radiation measurements at its absorption peak of 443 nm [33].
- **Look-up tables** store the output of a forward model for every permutation of a densely sampled set of parameters.
- **Optimization-based models** optimize guesses of parameters of interest using a forward model to generate intermediate results. These typically use the gradients (sometimes referred to as Jacobians) of the forward model in order to update parameters at each iteration.

Optimization-based models are typically the most accurate, but also the most expensive. They use a loss function to quantify the difference between the forward model and some reference value, typically the radiance measured by a sensor. Gradients should always be used in the optimization when available: the convergence rate of optimizations with gradients is better than without (e.g. finite differences) on noisy objective functions [34]. However, gradients are not always available if the forward model is not differentiable, or computing them by automatic methods (see differentiable rendering in Section 4) is not possible due to computing resource constraints.

State-of-the-art inverse models for remote sensing target the retrieval of several atmosphere–ocean parameters simultaneously. For example, the MAPOL retrieval algorithm [35] uses a non-linear least squares optimization to retrieve aerosol (size, complex refractive index) and ocean (wind speed, chlorophyll-a, absorption and backscattering coefficients of plankton and detritus) parameters. Successor FastMAPOL [36] replaced the optimization backbone with a neural network. Retrieval algorithm MAPP [37] uses the optimal estimation method to retrieve aerosol (top height, optical depth, size, complex refractive index) and ocean (wind speed, chlorophyll concentration) parameters. Retrieval algorithm GRASP [38] uses a multi-term Least Square Method statistical model to retrieve aerosol (optical depth, single-scattering albedo, size, shape, composition, etc.), ocean (BRDF/BPDF), and land (BRDF/BPDF) parameters (BRDF: bidirectional reflectance distribution function/BPDF: bidirectional polarization distribution function). These algorithms have been applied successfully to real-world multi-angle polarimeter measurements from the recent 2016 ORACLES [39] and 2017 ACEPOL campaigns [36,40–43].

However, inverse models can only be as accurate as their underlying forward models, which are still dominated by those assuming plane-parallel layers. Recent surveys have pointed out the need for inverse models that can account for the three-dimensional geometry of clouds, which are especially important for aerosol retrievals near cloud edges [44]. Monte-Carlo-based inverse models, such as those that have been developed for computer graphics, could provide more realistic estimates for such cases. Some in the remote sensing domain have developed methods to estimate gradients for cloud tomography by ad-hoc approximation [27–29,31], and retrieval of spatially varying cloud properties via differential radiative transfer simulation [26]. Czerinski and Schechner [26] use the same underlying principles as the radiative backpropagation method introduced to computer graphics (see Section 4.6) to estimate derivatives for Monte Carlo simulations. Our framework in Part 2 also uses differential radiative transfer simulation to estimate derivatives; while Czerinski and Schechner [26] focus on retrieval of spatially varying cloud properties, our goal is to build an extensible framework general to many types of scenes and properties.

As in remote sensing, inverse methods for acquiring material properties have a long and bountiful history in computer graphics; we choose to limit our discussion to the subset of methods that use differentiable rendering, described in Section 4.6.
movies, and television. Like remote sensing, early production environ-
guments (e.g. the dancing light patterns at the bottom of a pool).

3. The evolution of forward radiative transfer models in computer
graphics

In the 1980s and 1990s, one of the main applications of computer
graphics was to generate frames of synthetic images for video games,
movies, and television. Like remote sensing, early production envi-
ronments required image synthesis algorithms that were accurate and fast,
but in this case favored geometric complexity over lighting complexity
when allocating limited computational resources.

The dominant rendering algorithms at the time were various flavors
of rasterization: a multi-step pipeline that loads each object in a scene
individually, projects it onto the screen, and then "shades" (determines
the final color of) each pixel. This system was advantageous for early
computer systems that had little memory for storing geometry: each
object could be processed one at a time and there was assumed to be
no global interdependency of color. Dedicated graphics hardware was
developed around this assumption and can execute the rasterization
pipeline with extreme efficiency. However, this meant that effects
such as shadows cast by light sources and interreflection between
surfaces were missing and the final image was (by modern standards)
quite crude in appearance. Over time, some of these global effects
were wedged into the rasterization pipeline by either pre-processing
algorithms (e.g. creating shadow maps for shadows [47]) or human
effort (artists would place light sources strategically to create a false
impression of global illumination). However, adding each effect piece
by piece ultimately proved to be an inelegant solution to realistic
lighting.

While rasterization dominated production, important research in
the academic community in the 1980s began building the foundations
of physically based rendering. This research progressed along two parallel
tracks: "radiosity" and Monte Carlo. Finite-element-based "radiosity"
methods, inspired by heat transfer literature, discretize the scene ge-
ometry into patches and solve for the exchange of light between them
(notable early works include Goral et al. [48] and Rushmeier [49],
while Cohen and Wallace [50] provides an early overview). These
methods produced images of then-unprecedented realism, since the im-
ages were generated using a physical simulation of light. However, this
method scales poorly with increasingly complex geometry, emitters,
and materials, and hence failed to find widespread adoption within
commercial computer graphics settings.

Meanwhile, others began to integrate Monte Carlo sampling into
rendering algorithms to simulate glossy reflections, soft shadows, depth
of field, and motion blur [51,52]. Kajiya’s seminal work [53] (con-
currently introduced by Iimmel et al. [54]) merged Whitted’s recursive
ray tracing algorithm [55] with radiative transfer theory to introduce
"the rendering equation" (a formulation of the RTE for surfaces in a
vacuum) to computer graphics:

$$L_o(x, \omega) = L_s(x, \omega) + \int_{\Omega} L_i(x, \omega_i) f_s(x, \omega, \omega_i) \, d\omega_i.$$  (4)

Here \(L_s(x, \omega)\) is the outgoing radiance leaving a surface from position \(x\) in direction \(\omega\). This is equal to the sum of emitted radiance \(L_e\)
and reflected radiance \(L_r\). The reflected radiance is itself an integral of
incoming radiance \(L_i(x, \omega_i)\) from all directions \(\omega_i\) over the sphere of
directions \(\Omega\). This light is weighted by the surface’s cosine-weighted
BSDF \(f_s\). In a vacuum, the incoming and outgoing radiances are related
as \(L_i(x, \omega) = L_r(r(x, \omega), \omega)\), where \(r(x, \omega)\) is the ray tracing operator that
returns the closest surface point \(x_r := x - \omega \omega\) along the ray starting at \(x\)
in direction \(\omega\). Eq. (4) is therefore a recursive integral equation.

Kajiya [53] also introduced an algorithm to estimate the rendering
equation by Monte Carlo integration that became known as path tracing.
A basic form of this algorithm arises by simply using a 1-sample Monte
Carlo estimator to approximate the integral in Eq. (4):

$$\langle L_o(x, \omega) \rangle = L_s(x, \omega) + \frac{\langle L_i(r(x, \omega), \omega) \rangle f_s(x, \omega, \omega_i) \, d\omega_i}{p(\omega_i)}.$$  (5)

Rather than tracing the paths of light or " photons" from light sources,
pathing tracing is an adjoin method that instead shoots rays from the
sensor into the scene to intersect the nearest objects. Then, a new ray di-
rection \(\omega_i\) is chosen based on random sampling; recursively estimating
\(L_o\) via Eq. (5) builds a random path by alternating between intersection
and random directional sampling, until the path is absorbed, hits an
emitter, or escapes the scene. The contributions of many paths from the
sensor are averaged to produce a final Monte Carlo estimate of
radiance. This method scales well with increasingly complex geometry,
emitters, and materials; however, it requires enough memory to store
the entire scene at once (so that paths can be intersected with the
scene’s geometry) and time to take sufficient samples to get rid of

---

1 We use the convention that unit vectors always point along the flow of
light, so into a scattering event \(x\) when measuring incident radiance \(L_i\),
and out when measuring outgoing radiance \(L_o\) or \(L_r\).
objectionable noise (the variance in Eq. (3)). More early work drew connections between the rendering equation and particle transport theory in physics [56,57] and began to adapt well-established sampling strategies to the computer graphics context [58,59]. Others began to investigate ways to speed up the computation of “global illumination”, or indirect lighting: for example, Ward’s “irradiance caching” [60] performs the expensive indirect lighting calculations at sparse locations in a scene and stores them for later lookup and interpolation by nearby paths.

In the 2000s, as computer memory and processing power became more abundant and less expensive, Monte Carlo approaches such as path tracing became more favorable due to their elegance and realism (see [61–64] for more historical discussion). Subsequent research in the new millennium has focused on improving the statistical efficiency of Monte Carlo rendering algorithms (there are several examples of this in Section 4). Today, production renderers for animation and visual effects for movies (e.g. Pixar’s RenderMan [65], Disney’s Hyperion [66], Solid Angle’s Arnold [67]) are typically Monte-Carlo-based, often utilizing large parallel computing clusters called “render farms” to mitigate the increased computational cost. Video games and other “real-time” applications, meanwhile, have lagged behind the path tracing revolution since the faster rasterization pipeline is necessary in order to produce many frames per second. However, much like the movies in the 90s and 00s, real-time applications are starting to incorporate Monte Carlo methods to achieve specific isolated effects [68]. As graphics hardware continues to evolve rapidly in performance and new architectures are designed around ray tracing, a similar revolution may be on the horizon [69,70].

While production renderers have embraced physically based approaches to achieve photorealism, they are not good candidates for scientific research since they are typically proprietary/closed-source and are tailored to the artist-driven production environment. However, a number of open-source, extensible, physically based renderers have been developed for academic research, including PBRT [22] and Mitsuba 3 [7].

4. Highlights of relevant computer graphics research

From the above foundations, recent computer graphics research has developed Monte Carlo frameworks for polarization [71–76], fluorescence [77,78], Lorenz-Mie and T-matrix scattering [79,80], speckle statistics [81,82], and other wave interference effects [15,16,83,84]. While this non-exhaustive list demonstrates that forward radiative transfer models in computer graphics are capable of the base level of realism required for remote sensing, we highlight here several threads of computer graphics research that are especially relevant and ripe for migration to remote sensing (see Table 1 for terminology).

4.1. Multiple importance sampling and the generalization of rendering algorithms

Unidirectional tracing. While the introduction of path tracing was an important milestone in computer graphics, in its most basic form it is not well-suited to certain scenarios, for example finding bright caustics or small/narrow light sources (Fig. 2(a)). One could imagine mitigating these issues by reversing the path direction and instead starting paths at a light source, a variant known as light tracing (Fig. 2(b)); however, this simply reverses the problem as paths often miss the sensor.

Next-event estimation. One way to significantly improve on this is to sample direct connections to emitters (in the case of path tracing) or to the sensor (in the case of light tracing) at every scattering event — a technique known as next-event estimation (NEE) in computer graphics, or local estimate/directional estimate in remote sensing [85,86]. Mathematically, this can be accomplished by expanding Eq. (4) into itself once and separating the integral of directly visible emissive surfaces (direct illumination) from reflective surfaces (indirect illumination):

\[
L_i(x, \omega) = \int I_o(r(x, \omega), \omega_i) f_i(x, \omega, \omega_i) \, d\omega_i
\]

(6)

\[
+ \int I_o(r(x, \omega), \omega_i) f_i(x, \omega, \omega_i) \, d\omega_i.
\]

(7)

Since we know a-priori where all emissive surfaces are in the scene, Monte Carlo estimation of the first integral can leverage improved “emitter importance sampling” techniques [87] that send sampled directions only toward emitters. This can provide a significant variance reduction in the presence of concentrated emitters or small sensors, which would otherwise have a very low probability of being sampled.

Multiple importance sampling. While NEE reduces variance caused by missing the light sources/sensor, a direct connection cannot always be made (e.g. connections through refractive surfaces) and it may not be the optimal importance sampling strategy in some cases — for example,
sampling $E_2$ proportional to $L_2$, may still result in high variance if the remaining parts of the integrand (the BSDF $f$, and cosine term) vary significantly over the integration domain. Multiple importance sampling (MIS) [46] allows for multiple strategies to be combined by sampling according to several strategies and appropriately weighting the results. In its basic form, MIS generates samples from the mixture density of several sampling techniques. The key idea is that complex integrands like in the RTE are the product of many different functions (incident radiance, BSDF, cosine foreshortening term). MIS can mix several different strategies, where each could be tailored to different terms of the integrand. While MIS does not always reduce variance compared to the best strategy for a particular scenario, it allows the overall estimator to be more robust to a wide variety of scenarios than using any of the individual strategies alone.

MIS also allows for combining meta-sampling strategies such as rendering algorithms themselves. Bidirectional path tracing [53,88,89], for example, starts paths from both light sources and the sensor and connects them (Fig. 2(c)), then weights the full paths using MIS. Bidirectional path tracing is especially good at handling scenes with complex light transport paths where traditional next-event estimation may fail, such as bright caustics (especially relevant in scenes with an ocean surface).

**Advanced techniques.** In addition to the above path tracing variants, computer graphics has developed a cornucopia of Monte Carlo rendering algorithms unfamiliar to the outside scientific world (see [90] for a comprehensive overview). For example, photon mapping [91] breaks rendering into two passes: a preprocessing pass akin to light tracing, where the position, direction, and power of each light subpath vertex are stored as a “photon” in a “photon map” (typically a k-d tree). This is followed by a rendering pass akin to path tracing, but where recursion is short-circuited by performing density estimation of nearby photons stored in the first pass (Fig. 2(d)). This method can successfully sample complex light transport paths like bright caustics, at the cost of introducing bias to the radiance estimate (i.e. error that will not vanish as the sample count increases). Later variants like photon beams [92] and photon surfaces [93,94] store not just the path vertices but segments, resulting in lower-variance estimates, and, in some cases, no bias. Another example is Metropolis light transport [95], a Markov Chain Monte Carlo algorithm that begins with path tracing or some other variant, but then mutates existing paths to find those with high path throughput via the Metropolis–Rosenbluth–Hastings sampling routine [96,97].

Each rendering algorithm is tailored to reducing variance in certain scenarios; there is no universal rendering algorithm that performs better than all others under all circumstances (yet).

### 4.2. Lower-variance and multiple-scattering microfacet models

Microfacet models are commonly used to model the appearance of rough surfaces by describing the statistical behavior of the surface using a distribution of normals and a tunable “roughness” parameter. The Cox–Munk model [98], often used to represent the ocean surface in remote sensing, is a microfacet model in which the wind speed at the surface (or is “absorbed” by the masking-shadowing term). However, real-world rough surfaces exhibit significant multiple scattering, energy which is simply lost in a microfacet model. Heitz et al. [106] introduced a method to simulate multiple scattering in microfacet models by treating the surface as a special kind of “scattering medium” and performing a nested random walk at each scatter event. Later work introduced lower-variance [107,108] and analytic (for special cases) [109] estimators for multiple scattering in microfacet surfaces, as well as estimators for layered surfaces with different statistics [110]. Such extensions to microfacet models could provide more accurate estimates of water-leaving radiance, especially at grazing angles and in higher-wind conditions.

#### 4.3. Lower-variance estimators of spectrally and spatially varying scattering media

In homogeneous scattering media, estimating transmittance (the fraction of light that successfully passes through a stretch of medium) can easily be done by sampling “free-flight distances” (distances to the next scatter event within the medium) from an exponential probability distribution perfectly proportional to the falloff of light. However, the heterogeneous case is much more challenging because no closed-form probability distribution is available when the medium’s optical properties vary spatially.

A popular algorithm for estimating the transmittance of heterogeneous media called delta tracking, Woodcock tracking, or null scattering was developed for electron and neutron transport [45,111–116] and imported to graphics [117–123]. This method fills the scattering medium with “virtual” particles until it is a homogeneous density — then, sampling free-flight distances proceeds as if the medium is homogeneous, except the sampled scatter event is accepted or ignored with a probability proportional to the fraction of real particles at that location.

However, until recently, there was no way to calculate the sampling probability distribution of this method, which rendered it incompatible with MIS. Miller et al. [124] derived this missing probability distribution and demonstrated how combining spectral and spatial importance sampling via MIS significantly reduces overall variance. This, and other recent MIS developments [125], could be especially relevant for simulating mixtures of gases, which have distinct, narrow absorption spectra that are challenging to simulate efficiently.

#### 4.4. “Random” sample generation

The error convergence rate of a Monte Carlo estimator using uncorrelated random samples is $\Theta(\sqrt{N})$ [46] (from Eq. (3)), which crucially depends only on the number of samples $N$ and not the dimensionality of the integral. This is the property that makes it a feasible option for high-dimensional integrals such as the radiative transfer equation. However, this convergence rate is relatively slow: it takes four times the number of samples to reduce error by half. Computer graphics has imported a number of sample generation techniques that have been shown to improve the variance and convergence rate of Monte Carlo estimates of radiance. These sample sets are typically stratified (e.g. jittered [52], multijittered [126], Latin hypercube [127]/N-rooks [128], orthogonal arrays [129]) or carefully crafted (e.g. Halton [130] and Sobol [131]) to avoid clumps and gaps in the sample domain. The latter category of “crafted” sample sets are called quasi Monte Carlo — they replace the pseudorandom numbers used in the sampling process with deterministic, low-discrepancy point sets (see Keller [132] for a good discussion).

In practice, rendering frameworks in computer graphics typically abstract the concept of a “sampler” into a base class that is called to generate a sample whenever one is required and can be extended to implement any of these variants. Any sample generation algorithm can then easily be swapped in for a default independent random sampler, which is simply a wrapper for a pseudorandom number generator.
4.5. Non-exponential transport through scattering media

Traditional radiative transfer models assume that scattering particles within a medium are statistically independent, which results in exponential light decay along paths in the medium. However, real-world materials can exhibit both positive and negative correlations between scattering particles; notably, clouds exhibit positive correlations between droplets and therefore slower-than-exponential light decay [133,134]. Work in computer graphics [135–138], inspired by neutron transport [139–141] and the aforementioned work from atmospheric sciences, developed Monte Carlo frameworks that remove this assumption and allow for arbitrary (non-exponential) light falloff through a scattering medium. Non-exponential transport has the potential to achieve even more realistic simulation of clouds than existing forward radiative transfer models in remote sensing.

4.6. Differentiable rendering

One of the most exciting threads of computer graphics research to emerge within the past few years is differentiable rendering: computing the derivatives of arbitrarily complex Monte Carlo radiative transfer simulations, which can then be used to build inverse models via gradient-based optimization.

Many optimization methods require the gradients of a loss function with respect to parameters of interest in order to guide them toward a more accurate solution. However, deriving those gradients in the context of radiative transfer is clearly challenging — a single Monte Carlo radiance measurement is the sum of many paths scattered throughout the scene and influenced by the materials and geometry of everything they have touched. Initial work in computer graphics tackled subproblems such as estimating gradients of irradiance on surfaces [142–144] and in scattering media [145–147], direct lighting shadows [148,149], or material parameters [150–152]. Deriving the gradients of radiance with respect to arbitrary scene parameters (e.g. material parameters, vertex positions, sensor pose) remained elusive due to non-differentiable visibility discontinuities between objects in a scene. Consider a ray traveling from one object toward another: if its direction were perturbed slightly, that ray may hit or miss an occluder edge or silhouette edge — this is the discontinuity, a binary hit or miss. Li et al. [153] finally overcame this limitation by proposing an edge sampling method that finds the relevant edges of geometry in a scene and samples points along them in order to estimate the visibility gradient. This method was later generalized beyond surfaces [154]. The edge sampling routine incurs a hefty computational expense, however, and follow-up works have re-parameterized parts of the integral to avoid such discontinuities [155,156]. See Zhao et al. [157] for further discussion of recent advances in differentiable rendering.

The differentiable renderer Mitsuba 2 [71] (now Mitsuba 3 [7]), developed concurrently to the above work, takes a different approach by calculating the derivatives of light transport using reverse-mode automatic differentiation, or backpropagation as it is called in the context of neural networks, which is essentially recursive application of the chain rule [158]. Loubet et al. [159] introduced a change of variables formulation that can be used to avoid integrating over discontinuities.

Subsequent work introduced a method called radiative backpropagation [160–162] that avoids a cumbersome computation graph entirely by performing a differential radiative transfer simulation analogous to an ordinary radiative transfer simulation. This method has significantly sped up computation time and memory overhead compared to the previous approach, and also opened up the potential to study differential sampling strategies for this adjoint transport problem akin to existing sampling strategies for normal light transport [163].

Different radiative transfer methods have the advantage that effects, even complex ones such as Lorenz-Mie scattering, can be implemented normally in the forward sense and their derivatives can be automatically calculated without extra effort, as opposed to hand-calculated. As paths are mutually independent, they are just as “embarrassingly parallel” as ordinary radiative transfer. However, as with all Monte Carlo estimators, differential radiative transfer will produce noisy gradients and the user must choose an appropriate sample count. Differential radiative transfer is also just as computationally expensive as ordinary radiative transfer: subsequent works have explored ways to reuse information from one iteration of optimization to the next [26, 164,165].

5. Conclusion

By tracing the parallels and contrasts between radiative transfer models in the remote sensing and computer graphics communities, we lay the groundwork for a bridge of collaboration. This moment in time is ripe for cross-pollination of ideas due to the maturity of physically based rendering techniques for computer graphics and the need for increasingly sophisticated retrievals for remote sensing. We have also drawn attention to a number of research threads that have the potential to improve the accuracy and efficiency of state-of-the-art radiative transfer models in remote sensing, both through more faithful modeling of real-world, light-based phenomena (e.g. multiple-scattering microfacet models, non-exponential light transport) and carefully crafted sampling schemes (e.g. multiple importance sampling, stratified sampling, lower-variance estimators for microfacet models and scattering media). We are especially interested in the potential application of differentiable rendering to inverse models in remote sensing, and that is the avenue we intend to explore next. In Part 2 of this work, we take first steps toward putting these concepts into practice by extending Mitsuba 3, a differentiable rendering framework from the computer graphics community, to perform forward simulations of interest to remote sensing and validate our framework on benchmark tests of simple atmosphere–ocean systems.

CRediT authorship contribution statement

**Katherine Salesin:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Kirk D. Knobelspiese:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing, Methodology. **Jacek Chowdhary:** Supervision, Writing – review & editing. **Peng-Wang Zhai:** Supervision, Writing – review & editing. **Wojciech Jarosz:** Conceptualization, Funding acquisition, Methodology, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Code will be made publicly available at the author’s Github. Data will be made available upon request.

Acknowledgments

The authors thank Wenzel Jakob and the rest of the Mitsuba 3 development team for consultations on Mitsuba 3. We also thank Meng Gao for insightful discussions on radiative transfer codes.
Fascione L, Hanika J, Heckenberg D, Kulla C, Droske M, Schwarzhaupt J. Path
Nimier-David M, Vicini D, Zeltner T, Jakob W. Mitsuba 2: A retargetable
Marrs A, Shirley P, Wald I, editors. Ray tracing gems II: Next generation real-
Haines E, Akenine-Möller T, editors. Ray tracing gems: High-quality and
Georgiev I, Ize T, Farnsworth M, Montoya-Vozmediano R, King A, Lommel BV,
Arvo JR, Kirk D. Particle transport and image synthesis. Comput Graph (Proc
Arvo JR. Transfer functions in global illumination. In: Global illumination, ACM
Whitted T. An improved illumination model for shaded display. Commun ACM
Williams L. Casting curved shadows on curved surfaces. Comput Graph (Proc
Veach E, Guibas LJ. Bidirectional estimators for light transport. In: Photo-
Lafortune EP, Willems YD. Bi-directional path tracing. In: Proceedings of the
Deng X, Jiao S, Bitterli B, Jarosz W. Photon surfaces for robust, unbiased
Steinberg S. Accurate rendering of liquid-crystals and inhomogeneously optically
Steinberg S. Analytic spectral integration of birefringence-induced iridesc-
Veach E, Guibas LJ. Bi-directional estimators for light transport. In: Photo-
Steinberg S. Analytic spectral integration of birefringence-induced iridesc-
Immel DS, Cohen MF, Greenberg DP. A raidity method for non-
Shirley P. Physically based lighting calculations for computer graphics (Ph.D. thesis), Urbana–Champaign: University of Illinois; 1990.
Keller A, Fassione L, Fajardo M, Georgiev I, Christensen P, Hanika J, Eis-
Fassione I, Hanika J, Heckenberg D, Kulla C, Drosek M, Schwarzaupt J. Path
Christensen P, Fong J, Shae J, Wooten W, Schubert B, Kendal A, Friedman S, Kip-
Haines E, Akenine-Möller T, editors. Ray tracing gems: High-quality and
television path reuse in real-time in: ACM SIGGRAPH courses. New
Marrs A, Shirley P, Wald I, editors. Ray tracing gems II: Next generation real-
Mojit M, Skřivan T, Wilkie A, Křížanek J. Bi-directional polarised light


