

Collimated Whole Volume Light Scattering in Homogeneous Finite Media

Zdravko Velinov, and Kenny Mitchell

Abstract—Crepuscular rays form when light encounters an optically thick or opaque medium which masks out portions of the visible scene. Real-time applications commonly estimate this phenomena by connecting paths between light sources and the camera after a single scattering event. We provide a set of algorithms for solving integration and sampling of single-scattered collimated light in a box-shaped medium and show how they extend to multiple scattering and convex media. First, a method for exactly integrating the unoccluded single scattering in rectilinear box-shaped medium is proposed and paired with a ratio estimator and moment-based approximation. Compared to previous methods, it requires only a single sample in unoccluded areas to compute the whole integral solution and provides greater convergence in the rest of the scene. Second, we derive an importance sampling scheme accounting for the entire geometry of the medium. This sampling strategy is then incorporated in an optimized Monte Carlo integration. The resulting integration scheme yields visible noise reduction and it is directly applicable to indoor scene rendering in room-scale interactive experiences. Furthermore, it extends to multiple light sources and achieves superior converge compared to independent sampling with existing algorithms. We validate our techniques against previous methods based on ray marching and distance sampling to prove their superior noise reduction capability.

Index Terms—raytracing, color, shading, shadowing, texture.

1 INTRODUCTION

INDOOR scenes in video games, AR, VR and other interactive experiences are constrained by the room geometry. Destruction, fire, dust and smoke are gameplay elements that are often incorporated in those experiences. The microscopic particles created in these events can fill up the entire room and lead to light scattering. This phenomenon is observed as light shafts stretching from windows, doorways or other elements of the geometry.

Most previous solutions targeting real-time rendering focused on scattering in outdoor scenarios and did not account for the room geometry. The overall effect of these solutions is that the outdoor environment is greatly obscured by dense fog which might be undesired look for a certain experience. Furthermore, the rendering algorithm should incur low computational cost to deliver noise- and artifact-free images within the allotted rendering time.

Integration of the light scattered towards a camera can be expensive since an unbiased solution can potentially involve solving an integral with infinite number of dimensions [1], expressed as a Neumann series. Most real-time applications limit the evaluation to a single scattering event. Integration is performed by taking a set of samples along the camera ray defined by pinhole perspective projection. Visibility and transmittance at each scattering event along the camera ray is determined by performing multiple intersection tests against the scene geometry and evaluating the distance traveled through the medium to reach a light source.

Many techniques were proposed in the real-time rendering community to efficiently estimate the single-scattering integral of radiative transfer [2]. They range from numerical integration by a

Monte Carlo integrator or ray marching for estimation based on projection on Fourier basis functions or moments [3], [4] which can be further filtered. These basic frameworks are derived by separating geometry from radiance evaluation.

By considering the many facets of integration of single scattering in indoor scenes, we propose a set of techniques for sampling and integration of radiance. We explain most elements of our techniques with respect to rectilinear box-shaped medium. However, these ideas are general enough to extend to other geometry and they already seamlessly integrate within existing real-time game engines. Our central idea is to use the remarkably simple solution of unoccluded radiance in a medium constrained by a polygonal shape to derive

- 1) an algorithm for analytic integration of the unoccluded single scattering in box-shaped medium,
- 2) importance sampling algorithm matching exactly the distribution of the geometry-dependent transmittance along a camera ray,
- 3) an optimized Monte Carlo integration method exploiting variable caching for computing the radiance scattered inside the medium from a collimated light source towards the camera.

We are now going to clarify the nature of those three main contributions. Each of them can be considered separately since they are applicable within very different rendering frameworks.

1.1 Analytic Integration of Unoccluded Single Scattering and Ratio Estimators

The types of lighting conditions discussed in this work involve directional lights, used to approximate illumination by a very distant light source such as a star illuminating a planet, and collimated light sources masked by a gobo. We exploit the assumption of

- Zdravko Velinov was with Disney Research Los Angeles, Glendale, CA. ORCID: 0000-0002-1258-1223
- Kenny Mitchell was with Disney Research Los Angeles, Edinburgh Napier University and Roblox Corporation, San Mateo, CA. ORCID: 0000-0003-2420-7447

Manuscript received December 22, 2020; revised December, 2021.

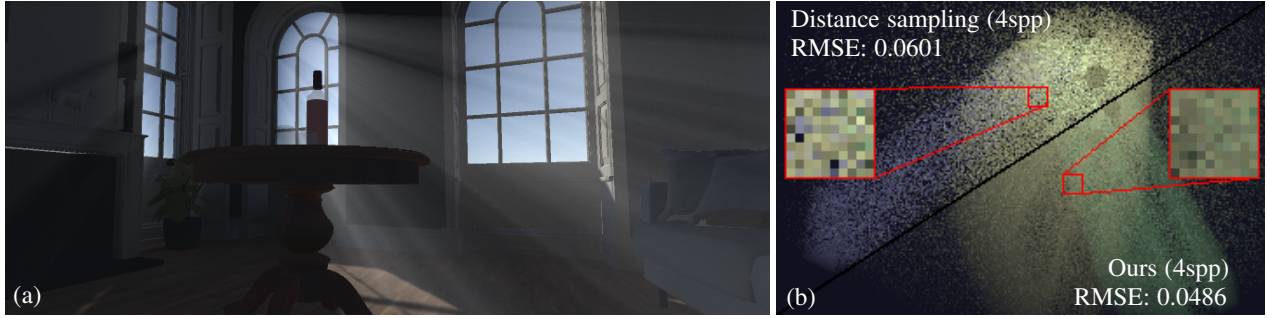


Fig. 1. We present a set of algorithms for efficiently computing single scattering by a box-constrained medium which are applicable to rendering real-time volumetric lighting effects in indoor scenarios. Our techniques correctly separates the indoor environment from the rest of scene while rendering at real-time frame rates with improved performance and variance reduction (a). Furthermore, they generalize to multiple scattering and can improve convergence in scenes with multiple collimated light sources (b).

collimated light to construct a skewed coordinate system with axes corresponding to a pair of light and camera rays and origin at the camera position. We intersect the box-shaped medium against the plane of this coordinate system and form linear segments. We will call them *trapezoid segments* for brevity. Radiance is analytically integrable, assuming homogeneous medium within those segments. We apply this integration in estimators that split the computation into two terms: unoccluded single-scattering term and an ratio term accounting for visibility. Our integration method solves the first term, while we use existing techniques to estimate the ratio between the occluded and unoccluded radiance in the second term. We investigate integration with single-scattering ratio estimators and approximate estimators based on moments [4]. This framework forms our first integration approach. The application of a ratio estimator enables standalone use of this technique since it can be easily combined with existing sampling schemes, such as equidistant and distance sampling algorithms. Previous approaches used it for shadowing BRDFs [5] or considered it as an approximation when in fact it can asymptotically converge to the correct solution of the integral. We furthermore show that it generalizes to multiple scattering and progressive rendering.

1.2 Whole Volume Importance Sampling

Our second contribution is an importance sampling scheme applicable in Monte Carlo integrators. Sampling is carried according to the geometry of the entire medium, thus incorporating both camera and light rays in the computation. This differs from classic techniques that limit sampling only according to transmittance along the current light or camera ray, depending on the integrator. The resulting sampling strategy can deliver images with lower variance compared to existing methods. This technique is applicable on its own in path tracing frameworks where existing approaches enable combining multiple sampling strategies [6].

1.3 Optimized Monte Carlo Integrator

The third contribution regards the implementation of the integrator using our sampling scheme. Our method is split into two steps: geometry traversal and segment selection. The geometry of the medium is successfully broken down into trapezoid-shaped segments by traversing the edges of the box from corner to corner to connect the incident and outgoing exit points from the medium. The repeated traversal for each sample can be avoided by caching variables describing each segment. Based on this data a segment can be selected and samples can be drawn proportional to transmittance

accounting for both the light and camera ray in paired fashion. This is the main optimized algorithm which we developed with real-time rendering application in mind. The final computation is significantly simplified and delivers performance improvements over ray marching.

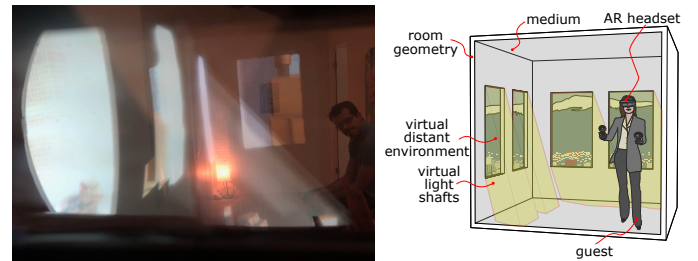


Fig. 2. **AR application.** Our technique deployed to an optical see-through AR headset experience as an additive lighting effect where virtual light shines through virtual windows onto the real world. The finite extent of the medium allows the distant buildings in the environment to be still visible and not obstructed by a thick fog.

Mixed reality room-scale experiences are one potential target for our technique. The additive nature of light shafts allowed us to successfully deploy the technique in a non blocking optical see-through AR setup (Fig. 2). We further see our techniques as general enough to incorporate them in video games and even as a sampling strategy in path tracing frameworks.

2 OVERVIEW

The main problem being solved is how to efficiently integrate radiance within a well-defined medium's bounds specified as a rectilinear box-shaped medium (Fig. 3). We are going to use this medium representation to clearly separate indoor and outdoor geometry allowing the distant landscape to be still visible from a room in our environment (Fig. 2), but retain the compelling effect of light scattering in a room filled with microscopic dust and smoke particles.

The main concern when performing this procedure is how to avoid corner cases and reuse computations. Scattering within a participating medium is governed by the equation of radiative transfer [2]. We will consider paths representing single-scattering events, which connect each point along the camera ray to the light source. Furthermore, we determine the bounds of the integration domain w.r.t. the medium geometry. In the case of a directional light source and a pinhole camera, the entire integration domain

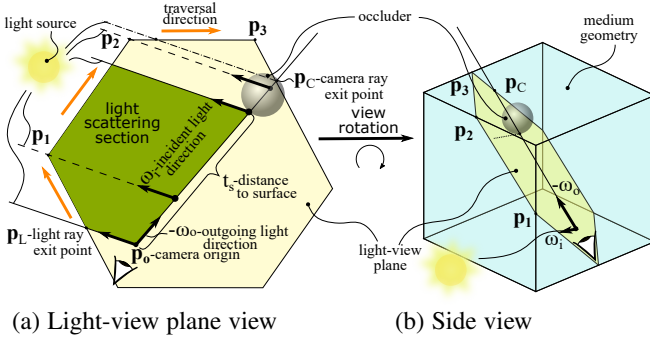


Fig. 3. **Box medium section.** (a) The complete medium is intersected by the light-view plane, resulting in a (b) section with complex geometry. The light exit point lies on the first face along the box slice. Following the boundary according to the shown direction (orange arrows), the camera view vector lies on the third face. The combination of the exit points and box corners forms a set of trapezoid segments. The illustrates the maximum possible corners lying between the light and camera exit point that can result in segments that contribute to the final integral. Refer to Tab. 1 for symbols explanation.

lies on a single plane for each individual pixel on the imaging plane of the camera sensor.

The plane is defined by the direction of the camera ray and light source. We will call each of these planes a *light-view plane*. We outline the section of the box formed by intersecting the (yellow) light-view plane against the (blue) medium geometry in Fig. 3. The shape of the section can be a point or a polygon with three to six edges. Moreover, scattering happens only in the (green) section between the camera and closest occluder or medium exit point. Combining those factors makes it non-trivial to construct an algorithm that handles all corner cases and performs all operations analytically. We chose to solve this problem by iteratively breaking down the section into trapezoid-shaped domains that allow analytic integration. The iterative loop is executed by selecting the starting point \mathbf{p}_L which is the exit point of the ray starting at the camera origin and pointing towards the light source. The algorithm then traverses the outline of the section until it reaches the exit point of the camera ray \mathbf{p}_C .

The vectors parallel to each segment of the boundary can be determined in advance in the coordinate system of the box where each of the three parallel pairs of faces is orthogonal to one of the axes. This is accomplished by finding the vectors that are both orthogonal to the specific axis and lie on the light-view plane¹. We use each \mathbf{q} in the process of traversing the outline to determine the next direction that lies on the adjacent face. Their direction is selected to progress the computation towards the exit point of the camera ray. Each of these vectors is parallel to the orange arrows in Fig. 3 and they outline the direction of traversal.

The specified geometry allows analytic integration for unoccluded geometry by following the integral form of the radiative transfer equation (RTE) [2]. However, shadowing needs to be further applied to recreate the appearance of crepuscular rays cast from doorways, windows and other geometry masking out light. We are going to discuss three techniques to achieve it. The first approach takes the analytic integration of the single-scattered radiance in a section of the box determined by the camera ray and directional light source and pairs it with a ratio estimator

1. We name these *face vectors* as they lie on a specific face without being explicitly assigned to a certain position

or approximate estimator based on projection on moments. Both approaches use the same basic framework of analytically computing the unoccluded radiance and then numerically integrating the ratio between unoccluded and occluded radiance. In the case of moment-based estimation it uses a heuristic formula which leads to an approximate solution, while ratio estimators with a sampling-based numerical integrator are capable of computing an asymptotically converging correct solution. Even in multiple scattering cases.

The second approach exploits the fact that the integral over all segments can be analytically inverted. Thus an efficient sampling strategy can be derived that accounts for scattering within the entire section of the box. The derived sampling function and probability density function (PDF) can be used in general path tracing frameworks.

Finally, an efficient integrator is developed that exploits the sampling method to analytically integrate radiance using caching to avoid executing at every iteration the geometry traversal algorithm outlined in Fig. 3. Its optimized equation allows cancellation of the exponential part of the integrand since the probability density function in use matches that component perfectly. We used this integrator for real-time rendering applications on the GPU.

The following section will proceed with explanation of how our methods fit within ratio estimators and Monte Carlo integrators (Sec. 3) used to solve the equation of radiative transfer [2]. Following those general concepts, the discussion proceeds with geometry traversal and integration of individual segments of the box section (Sec. 4). Sampling is derived afterwards by inverting the integral analytically (Sec. 5). Finally, a Monte Carlo integrator is defined using the sampling algorithm (Sec. 6).

3 THEORETIC FOUNDATION AND RELATED WORK

Light scattering in uncorrelated participating medium is traditionally modeled using the radiative transfer equation (RTE) [1], [2]. Graphics rendering systems estimate the solution of its integral form [1], [7], [8]. *Volume rendering systems* based on path tracing [8] solve light transport by alternating between generating path segments representing a scattering event and then connecting them to a light source, and selecting a new direction of light propagation with probability proportional to a phase function. *Photon tracing systems* [9] solve radiative transfer by accumulating the contribution of a set of points spread throughout the medium, and a system working at interactive rates was shown by Jimenez *et al.* [10].

One common simplification is to treat the medium as locally or completely homogeneous. Then path segments are generated by drawing distance samples from distribution with exponential probability density function proportional to transmittance along the current path segment, also known as *distance sampling* [1], [8]. However, this simplification also gives rise to more sophisticated algorithms tracing well-defined primitives, such as points, beams [11], planes and volumes [12], even surfaces [13]. Unconverged images using those custom integrators show structured artifacts in the shape of the primitive which is less desirable than noise introduced by path tracing approaches.

In real-time graphics most methods are constrained to single scattering. We first assume homogeneous medium and ignore light emitted by the medium, leading to the equation

$$L(\mathbf{p}_o, \omega_o) = L_s(\omega_o) e^{-\sigma_t t_{SC}} + \int_0^{t_{SC}} dt e^{-\sigma_t t} \int_{\mathbb{S}} d\omega_i \sigma_s f(\omega_i, \omega_o) L(\mathbf{p}_o - t\omega_o, \omega_i). \quad (1)$$

The total radiance $L(\mathbf{p}_o, \omega_o)$ arriving at the camera is a sum of the in-scattered radiance $L(\mathbf{p}_o - t\omega_o, \omega_i)$ from all directions \mathbb{S} at each point along the current ray and radiance arriving after scattering off a surface $L_s(\omega_i, \omega_o)$ – both exponentially attenuated by the medium according to Beer’s law based on the extinction σ_t and distance. The ray is selected to end either at the surface t_S or exit point from the medium t_C depending on what is closest ($t_{SC} = \min(t_S, t_C)$). Radiance at each scattering event is weighted by the scattering coefficient σ_s and phase function $f(\omega_i, \omega_o)$, depending on the incident ω_i and outgoing ω_o light direction. Single scattering means that only direct light illumination is computed at each scattering event ($L(\mathbf{p}_o - t\omega_o, \omega_i) = L_i e^{-\sigma_t d_m(t, \omega_i)}$). The distance to the medium boundary $d_m(t)$ leads to spatially-varying transmittance dependent on its geometry. Another common assumption that we will follow is uniform illumination ($L_i = \text{const}$). Multiple light sources can be linearly combined. See App. A for a detailed derivation.

The traditional method for solving this equation is using Monte Carlo integration employing importance sampling. The integrand is evaluated at a fixed number of samples and weighted by the probability density of each sample (PDF(t')),

$$L(\omega_o, \omega_i) = L_s(\omega_i, \omega_o) e^{-\sigma_t t_{SC}} + \sigma_s f(\omega_i, \omega_o) L_i \frac{1}{N} \sum_{k=0}^{N_{\text{sample}}} \frac{e^{-\sigma_t(t'+d_m(t'))}}{\text{PDF}(t')} V(t', \omega_i), \quad (2)$$

The visibility function $V(t, \omega_i)$ accounts for occlusion by opaque geometry. Ray marching [14] is considered as the most basic method of integrating single-scattering along a ray by performing equidistant sampling using the truncated RTE. The offline rendering community has extensively studied sample generation in participating media. The most straightforward sampling approach is *distance sampling* based on inversion of the transmittance term which is performed in closed form for homogeneous media [1]. It works with bidirectional path tracing [15] and Metropolis light transport [16]. Kulla and Fajardo [17] proposed *equi-angular sampling* which distributes samples closer to the light source in participating media. More advanced techniques were developed in the case of connecting multiple paths throughout participating media [18]. We propose another approach to integrate the radiative transfer by using importance sampling. Our sampling strategy, which we call *whole volume distance sampling*, accounts for the complete shape of the medium when distributing samples along the camera ray. In comparison, traditional *distance sampling* generates samples with probability proportional only to transmittance along the current (camera) ray.

Other approaches consider different integration methods of this equation with improved performance. Analytic solutions and approximations exist for a camera and point light source immersed in unoccluded participating media [19], [20]. Very early works rely on shadow volumes for integration [21]. In the real-time community different implementations exist of this basic approach [22], such as rendering shadow planes [23] and interleaved sampling with the programmable shading pipeline [24]. Moro *et al.* [25] further optimized shadow plane placement and used 3D textures to store cloud densities. Wyman and Ramsey [26] proposed a hybrid between ray marching and shadow volumes employing bilateral upsampling in the shadowed regions. Munoz [27] extensively studied integration strategies to derive a set of algorithms with improved convergence properties. Sampling in a set of points and gradient-based interpolation was studied

for rendering heterogeneous medium [28]. Radial basis functions (RBF) were used for low-frequency approximation of density fields and illumination by low-frequency environment light sources [29]. Other approaches use voxelized shadow volumes for scattering [30], [31]. More specialized methods were proposed to re-parameterize the integral in epipolar coordinates [32], [33] or linearly rectifying the integration domain [3]. Epipolar coordinates were further used with a min-max tree to speed-up traversal [34]. We outline an algorithm in our work based on Monte Carlo integration that exploits the analytic solution of unoccluded radiance under the above-stated assumptions in a finite medium to cancel out the transmittance component and estimate only visibility.

Ratio estimators are another approach to estimate radiance based on unoccluded radiance estimate and a ratio of the visibility divided by the sum of all weights. Previously, Heitz *et al.* [5] applied Monte Carlo ratio term on unoccluded area light approximations. Predating this work, techniques based on filtering shadow maps computed volumetric single scattering [3], [4] using an approximate ratio term and exact unoccluded term. In our work we derive and study variations of ratio estimators that asymptotically converge to the correct solution in the form of a product between unoccluded analytic expression or Monte Carlo estimate, and the ratio of two Monte Carlo estimates (occluded and unoccluded). The more widely discussed and related to previous works is its single-scattering variant,

$$\hat{V}(\omega_i, \omega_o) = \frac{\sum_{k=0}^{N_{\text{sample}}} e^{-\sigma_t(t'+d_m(t'))} V(t', \omega_i) / \text{PDF}(t')}{\sum_{k=0}^{N_{\text{sample}}} e^{-\sigma_t(t'+d_m(t'))} / \text{PDF}(t')} \\ L(\omega_i, \omega_o) = L_s(\omega_i, \omega_o) e^{-\sigma_t t_{SC}} + \sigma_s f(\omega_i, \omega_o) \mathcal{T}_{\text{box}}(\omega_i, \omega_o, t_S) \hat{V}(\omega_i, \omega_o) L_i, \quad (3)$$

where the distance ($t' = S(\xi(k/(N_{\text{sample}} - 1)))$) is generated by an importance sampling function $S(x)$ which transforms samples from uniform distribution $\xi(x)$ to the target distribution. Estimation is separated in two terms. The first term is the unoccluded integral $\mathcal{T}_{\text{box}}(\omega_i, \omega_o, t_S)$. It is essentially an integral over transmittance at each point along the camera ray where light source radiance L_i is separated out. Our analytic computation of the single-scattering radiance in a box section corresponds to that term (Sec. 4). The second term is a ratio between the current occluded and unoccluded estimate. Evaluation is performed numerically at a finite number of samples N_{sample} . In the case of completely unoccluded section, the estimator converges in a single sample, since the ratio term, which is dependent on visibility $\hat{V}(\omega_i, \omega_o, t')$, equals 1. Ratio estimators can be further generalized to multiple scattering by evaluating both the unoccluded and occluded light transport component. A generic formula for the analytic unoccluded term in that particular case does not exist for complex medium, therefore it can be analytically integrated after each scattering event and combined with next scattering events using stochastic methods. Other surfaces can be handled either by folding them into the ratio term or splitting the evaluation using the linearity of light transport. More details are explained in the supplemental material, since this step does not significantly alter the ratio estimator framework. It has to be noted that progressive rendering requires accumulation of three surfaces containing both unoccluded terms (estimated and analytic) and the occluded term. Otherwise, the denominator in the estimator will not be properly canceled and it will lead to bias. Further information about potential bias and convergence behavior is explained in the supplemental material.

The previously proposed estimators using Fourier coefficients [3] and moments [4] compute the ratio term \hat{V} based on heuristics. These approaches were inspired by earlier works on filtering shadow maps [35]. In those previous works computation is targeted towards outdoor scene rendering. Estimators of radiance scattered in the atmosphere were extensively studied in the graphics literature [36], [37], [38], [39]. Our techniques, on the other hand, are focused on indoor scenes constrained by the room geometry. We use the integral separation according to a ratio estimator using both numerical and approximate moment-based estimation [4] and combine it with our unoccluded single-scattering solution in rectilinear box medium.

TABLE 1
List of Definitions

Symbol	Definition
$\mathbf{B}_{3 \times 4}$	box coordinates linear transform matrix (cf. Sec. 4.1)
$\tilde{\mathbf{x}}$	point or vector in box coordinates (see below)
$\omega_i, \tilde{\omega}_i$	incident light vector in world and box coordinates (cf. Fig. 3)
$\omega_o, \tilde{\omega}_o$	outgoing light vector in world and box coordinates (cf. Fig. 3)
$\mathbf{p}_o, \tilde{\mathbf{p}}_o$	camera origin in world and box coordinates (cf. Fig. 3)
d_L	distance to exit point from origin along light ray (cf. Eq. (4))
$\mathbf{p}_L, \tilde{\mathbf{p}}_L$	light ray medium exit point (cf. Eq. (4))
$\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$	positions of corners in world coordinates (cf. Fig. 3)
t_C	distance to exit point along camera ray (cf. Eq. (5))
$\mathbf{p}_C, \tilde{\mathbf{p}}_C$	camera ray medium exit point (cf. Fig. 3)
$t_{C,1}, t_{C,2}$	distance to corner projected on camera ray (cf. Fig. 4, Fig. 5)
t_S, t_{SC}	distance to surface and furthest point in medium (cf. Fig. 4, Fig. 5)
$\tilde{\mathbf{w}}$	vector along box face (cf. Sec. 4.4)
L	radiance scattered at a point in space (cf. Eq. (1))
L_s	radiance scattered by a surface (cf. Eq. (1))
L_i	radiance emitted by a light source (cf. Sec. 3)
σ_s	single-scattering coefficient (cf. Eq. (1))
σ_t	extinction coefficient (cf. Eq. (1))
f	phase function (cf. Eq. (1))
V	visibility of light source (cf. Eq. (2))
\tilde{V}	term estimating total visibility (cf. Sec. 3)
\hat{V}	ratio term (cf. Eq. (3))
N_{sample}	number of samples (cf. Eqs. (2) and (3))
\mathbf{K}	cached variables (cf. Sec. 4.2)
$\xi(x)$	generator of uniformly distributed values (cf. Eqs. (2) and (3))
\mathcal{I}	integral over transmittance (cf. Sec. 4.2)
\mathbf{q}	face vector (cf. Sec. 2)
$d_i \dots$	distance between corner and camera ray (cf. Sec. 4.4)
$d_{E,S}, d_{E,E}$	distance to corner at start and end of trapezoid (cf. Sec. 4.4)
t, t'	distance along camera ray (variable and random – cf. Eqs. (2) and (3))
d_m	variable distance to medium exit point (cf. Eq. (2))
P	probability (cf. Sec. 5.2)
r	random value (cf. Eqs. (2) and (3))

4 INTEGRATION OF UNOCCLUDED RADIANCE

We are now ready to discuss our first contribution. The first method performs radiance integration in a section of the box enclosed between the camera origin and the view and light ray. Integration is only performed by evaluating transmittance at each scattering event along the camera ray. The radiance integration method is going to be central in our ratio estimator-based solutions and some of its intermediate results are essential to explain the importance sampling scheme. We are first going to discuss the coordinate system as it is essential to understand how the algorithm is being evaluated. Then we are going to discuss the main loop of the

algorithm to give a perspective why the different steps are required and then we are going to explain the underlying procedures in detail. We provide pseudo-code for all algorithms in App. E.

4.1 Coordinate system

All operations are performed in box coordinates where the x-, y- and z-axis of each point are within range $[-1, 1]$. If the camera origin lies outside the box, it is moved to the first intersection with the box, or the contribution is assigned to 0 when the camera ray does not intersect the box. The orientation, scale and translation of the box medium is expressed as a matrix $\mathbf{B}_{3 \times 4}$. The origin of the light ray is transformed in box coordinates ($\tilde{\mathbf{p}}_o = \mathbf{B}_{3 \times 4} [\mathbf{p}_o \ 1]^T$), and the incident light vector is similarly expressed in non-normalized box coordinates ($\tilde{\omega}_i = \mathbf{B}_{3 \times 4} [\omega_i \ 0]^T$). The intersection of the light ray with the box determines the starting point of the traversal algorithm,

$$\begin{aligned} d_L &= \text{IntersectBox}(\tilde{\mathbf{p}}_o, \tilde{\omega}_i) \\ \tilde{\mathbf{p}}_L &= \tilde{\mathbf{p}}_o + d_L \tilde{\omega}_i. \end{aligned} \quad (4)$$

The intersection algorithm based on the slab test [40], [41] is explained in App. B. Since that information doesn't change, it can be assigned as a shader constant in the final implementation. The exit point along the camera ray is needed as a destination point of the traversal, as seen in Fig. 3. It can be computed by rasterization of the box or in a similar fashion in box coordinates ($\tilde{\omega}_o = \mathbf{B}_{3 \times 4} [\omega_o \ 0]^T$) by performing an intersection,

$$\begin{aligned} t_C &= \text{IntersectBox}(\tilde{\mathbf{p}}_o, -\tilde{\omega}_o) \\ \tilde{\mathbf{p}}_C &= \tilde{\mathbf{p}}_o - t_C \tilde{\omega}_o. \end{aligned} \quad (5)$$

4.2 Integration algorithm

The exit points along the light ray and camera ray give us the starting and end point and determine the boundaries of the box section that will be traversed by the algorithm. The main objective is to break down the box section defined by the intersection of the light-view plane against the box into multiple segments (Fig. 3). Since the light source is assumed to emit uniform light, the integral can be split into a product of integral over transmittance and emitted radiance from the light source.

Three is the number of iterations performed by the algorithm which corresponds to the maximum number of edges involved in a box section. The main explanation for this fact is that a box is composed of three parallel planes and any set of rays constrained to a line segment inside the box can intersect at maximum three faces. We are going to outline the main steps in each iteration and discuss them in detail in their own subsections. Each iteration starts by finding the next corner. Then the distance to the corner and along camera ray is computed. They are used to integrate radiance analytically for a trapezoid segment,

$$\mathcal{I}_k = \mathcal{I}_{\text{trapezoid}}(t_{C,S}, t_{C,E}, d_{E,S}, d_{E,E}).$$

The trapezoid is geometrically defined by the distance of the starting point ($t_{C,S}$) and end point ($t_{C,E}$) along the camera ray combined with distance from the starting point ($d_{E,S}$) and end point ($d_{E,E}$) to the edge of the box. Each iteration advances the distances along the camera ray and recomputes the distance to the edge. Transmittance of all segments is summed to derive the final integral ($\mathcal{I}_{\text{total}} = \sum_{k=1}^3 \mathcal{I}_k$). The sampling methods and optimized Monte Carlo algorithm use this procedure. They further pre-cache a set of variables (\mathbf{K}) and the individual integral values for each trapezoid segment (\mathcal{I}).

4.3 Selecting the direction of traversal to next corner

The direction of traversal of the boundary outline is defined as the intersection of the light-view plane with the box which leads to progress towards the exit point along the camera ray. We will call *face vectors* the direction of each line formed by this computation. They are associated with a pair of parallel faces. We compute them in advance before traversing the outline to determine the position of each light segment. They are computed to be perpendicular to each axis (x -, y - and z -axis) and the light-view plane. The direction of these vectors is selected to follow the direction of progress towards the camera ray exit point or the shortest path to the same point. The direction of progress is determined in skewed coordinates determined by the incident and outgoing light direction vector. The exit point is projected on the outgoing light direction and its sign is taken as a direction of progress. Detailed derivation is provided in the supplemental material and the appendix. The default case is to multiply by the progress along the camera ray ($v = g_{\text{progress}}(\mathbf{a}, -\tilde{\omega}_o)$). When the light plane is parallel to the camera ray direction, traversal is not performed, instead the integral over transmittance is directly computed. Therefore, this case is not considered by the algorithm. Another possibility is that the vector connecting the two points is parallel to the light plane ($|v| < \epsilon$). Then the direction along the camera ray is selected instead ($v = -(\mathbf{a} \cdot \tilde{\omega}_o)$). The final face vector is determined by multiplying the value representing the direction of progress ($\mathbf{q} = v\mathbf{a}$).

The resulting *face vectors* are used when selecting the next direction of traversal. At each corner there is a choice between one of three directions corresponding to each face vector associated with each axis. A face vector is selected only if the current position lies on an edge that connects a face corresponding to it and progress can be made over the remaining coordinates by traversing that face vector. The rest of the algorithm finds the smallest distance that must be traversed based on all coordinates given the selected face vector.

4.4 Computing the next corner

The next algorithm forms trapezoid segments by simultaneously computing in the skewed light-view coordinate system the length of the two line segments connecting its projection on the camera ray with the corner and camera origin. First, overstepping past the exit point along the camera ray ($\tilde{\mathbf{p}}_C$) is prevented by finding the progress along the edge and clamping it to fall within the line segment ($t_{\text{cut}} = \min(1, g_{\text{progress}}(\tilde{\mathbf{p}}_C - \tilde{\mathbf{p}}, \tilde{\mathbf{w}}))$). Then the end point along the edge is computed ($\tilde{\mathbf{p}}_{\text{edge}} = \tilde{\mathbf{p}} + t_{\text{cut}}\tilde{\mathbf{w}}$). Next, the projected distance along the camera ray is computed ($t_{C,E} = g_{\text{progress}}(\tilde{\mathbf{p}}_{\text{edge}} - \tilde{\mathbf{p}}_o, -\omega_o)$) in a similar manner to form the distance vector between the line and the edge. The result of the computation is the distance along the camera ray ($t_{C,E}$). Combining it with the origin ($\tilde{\mathbf{p}}_o$) and outgoing light direction (ω_o) yields the final point ($\tilde{\mathbf{p}}_{\text{cam}} \leftarrow \tilde{\mathbf{p}}_o - t_{C,E}\omega_o$). The distance from the camera ray and the edge is computed as the difference between the point on the camera ray and edge which is projected on the scaled incident light ray vector ($d_{E,E} \leftarrow (\tilde{\mathbf{p}}_{\text{edge}} - \tilde{\mathbf{p}}_{\text{cam}}) \cdot \tilde{\omega}_{i,s^2}$). The incident light vector $\tilde{\omega}_{i,s^2}$ is corrected following the procedure in App. C.

4.5 Radiance computation in a trapezoid segment

We now outline how radiance is integrated for each segment. Each segment is a trapezoid section defined by two parallel bases along the incident light vector ($\tilde{\omega}_i$) and two legs containing the line segments: along the camera ray and medium boundary. Essentially,

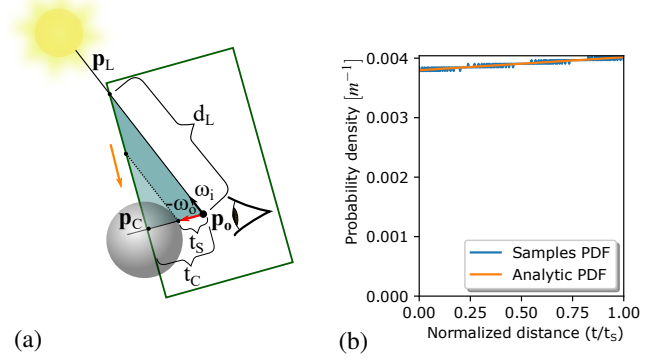


Fig. 4. **Sample distribution when the box section includes a single trapezoid segment.** (a) Light and view exit points lie on the same face which leads to (b) samples being placed towards the back section where the attenuation is lower. Distance is along the camera ray ($-\omega_o$) and follows the direction outlined in red.

a linear segment in a skewed coordinate system. The domain has a slope,

$$c = 1 + \frac{d_{E,E} - d_{E,S}}{t_{C,E} - t_{C,S}}.$$

The final expression is a product of transmittance to the start of the segment and integral over transmittance in the linear segment, which has remarkably simple solution and continues to be compatible with distance sampling,

$$\begin{aligned} \mathcal{I}_{\text{trapezoid}} &= e^{-(t_{C,S} + d_{E,S})\sigma_t} \int_0^{t_{C,B} - t_{C,S}} e^{-\sigma_t ct} dt \\ &= e^{-(t_{C,S} + d_{\text{edge},S})\sigma_t} (1 - e^{-\sigma_t(t_{C,B} - t_{C,S})c}) = E_l(1 - E_u). \end{aligned} \quad (6)$$

The distance along the camera ray is clamped by the distance to the surface to account for occlusion along the camera ray ($t_{C,B} = \text{clamp}(t_S, t_{C,S}, t_{C,E})$). In the limit when the slope approaches zero ($c \rightarrow 0$), the computation turns into a linear equation ($\mathcal{I} = (t_{C,B} - t_{C,S})E_l$). The same expression is used to avoid numerical issues when the size of the segment approaches zero ($t_{C,E} - t_{C,S} < \epsilon$). Note, that it is possible to reduce the number of exponent evaluations by extracting the product of the lower and upper bound part ($E_l \cdot E_u$) outside of the evaluation and assigning it as lower bound part at each consecutive evaluation (E_l). There are two cases when the incident $\tilde{\omega}_i$ and outgoing light $\tilde{\omega}_o$ vectors are parallel and require to be solved separately. The same procedure applies without traversing the outline. When they are facing opposite directions ($\tilde{\omega}_i \cdot \tilde{\omega}_o < 0$), the end distance goes to zero,

$$\mathcal{I}_{\text{box}} = \mathcal{I}_{\text{trapezoid}}(0, t_C, d_L, 0). \quad (7)$$

and when they are facing the same direction it is the sum of the distance to both exit points along the incident d_L and outgoing light vector t_C ,

$$\mathcal{I}_{\text{box}} = \mathcal{I}_{\text{trapezoid}}(0, t_C, d_L, d_L + t_C). \quad (8)$$

The integration algorithm as explained requires an estimator capable to account for shadowing. We chose to implement both a ratio estimator and an approximate moment-based estimator in our work. Another approach is to use Monte Carlo integration. We are going to explore it in the following and compare it against moment-based estimation and ratio estimator in the Results section.

Integration can be trivially extended to a collimated light source with a gobo (convex binary map or stencil) by offsetting the starting point of the integration to lie on the gobo edge and limiting it to the far end of the gobo. The result is exponentially attenuated

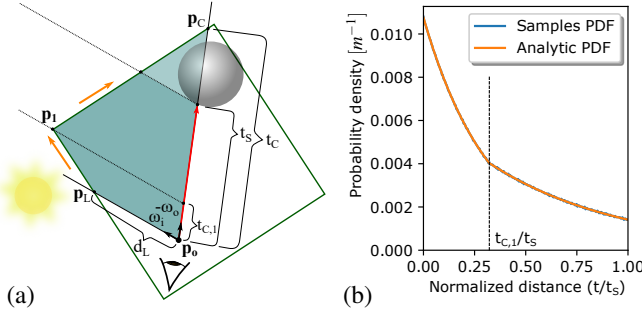


Fig. 5. **Sample distribution when a box section includes two trapezoid segments.** (a) Light exit point lies on the first face along the box slice and the camera view vector lies on the second face which leads to (b) a change of absorption and redistribution of samples. Distance is along the camera ray ($-\omega_0$) and follows the direction outlined in red.

by the distance from the camera origin to the start of the gobo. Multiple lights can be also combined thanks to the linearity of light transport. An example of both effects is illustrated in Fig. 1. We use our unoccluded radiance computation to compute weights for the multiple lights sampling following Monte Carlo integration similar to Shirley and Wang [42]. We demonstrate this novel generalization to rendering light scattered in finite medium in Fig. 1 (b). More details regarding gobo masking and multiple light integration are available in the supplemental material.

5 WHOLE VOLUME DISTANCE SAMPLING

The two important elements required to incorporate a sampling strategy in path tracing or single-scattering integrators are a sampling function and probability density function (PDF). The sampling function distributes samples according to the probability defined by the PDF. In the following we discuss both functions separately.

5.1 Importance sampling function

Sampling according to transmittance dependent on medium geometry is achieved in two steps. The first step is to build a histogram. The integration of transmittance of each trapezoid segment is according to the procedure explained above. Trapezoid is selected using comparison against the cumulative density function (CDF) of all segments. Considering the geometry along the entire camera line segment is what separates our technique from existing distance sampling techniques. CDF is computed by summing the integral of transmittance up to that segment ($\sum_k \mathcal{T}_k$) and dividing it by the unoccluded integral of transmittance ($\mathcal{T}_{\text{total}} \dots$) = $\mathcal{T}_{\text{box}}(\tilde{\omega}_i, \tilde{\omega}_o, t_S)$. Then the appropriate sampling function for each trapezoid segment is computed,

$$S_{\text{hist}}(r) = \begin{cases} S_s(T_1(r), \tilde{\mathbf{p}}_L, \tilde{\mathbf{w}}_1, 0, d_L) & r < \frac{\mathcal{T}_1}{\mathcal{T}_{\text{total}}} \\ S_s(T_2(r), \tilde{\mathbf{p}}_1, \tilde{\mathbf{w}}_2, t_{C,1}, d_1) & \frac{\mathcal{T}_1}{\mathcal{T}_{\text{total}}} < r < \frac{\mathcal{T}_1 + \mathcal{T}_2}{\mathcal{T}_{\text{total}}} \\ S_s(T_3(r), \tilde{\mathbf{p}}_2, \tilde{\mathbf{w}}_3, t_{C,2}, d_2) & \text{otherwise,} \end{cases} \quad (9)$$

where the random value $r \in [0, 1]$ is transformed after selecting the segment. Its value is multiplied by the unoccluded integral $\mathcal{T}_{\text{total}}$, then the integral over previous segments ($\sum_{k=1}^i \mathcal{T}_{k-1}$) is subtracted and the whole result is normalized by the integral value of the current segment \mathcal{T}_i ,

$$T_i(r) = \frac{\mathcal{T}_{\text{total}} r - \sum_{k=1}^i \mathcal{T}_{k-1}}{\mathcal{T}_i}; \quad \mathcal{T}_0 = 0. \quad (10)$$

The outcome is an intermediate random value in range $[0, 1]$. The next step is to determine the distance along the camera ray within the segment. Position of corners ($\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \tilde{\mathbf{p}}_3$) and distance vectors between corners ($\tilde{\mathbf{w}}_1, \tilde{\mathbf{w}}_2, \tilde{\mathbf{w}}_3$) are required to determine the distance between camera ray and box boundary (Eq. (9)). The computation of these components (Sec. 4.4) is performed similarly to the analytic radiance integration algorithm (Sec. 4.2). The final sampling method accounts for distance to the edges ($d_{E,S}$, $d_{E,E}$) and distance along the camera ray ($t_{C,S}$, $t_{C,E}$) in a paired fashion. It differs from *distance sampling* that accounts only for transmittance along the current path segment [8]. The proposed importance sampling scheme is equivalent to *distance sampling* only when evaluating a single segment where the distance to the boundary is constant ($d_{E,S} = d_{E,E}$). More details are available in the supplemental material. Our importance sampling considers the connection from each path segment to box boundary which is bound by trapezoids. The bases are the path segments of each pair of vertices along the camera ray to the exit point along the light ray. The legs are the box boundary and camera ray. Computation requires similarly the slope (c), bounding of the distance along the camera ray ($t_{C,B}$) and upper bound component of the trapezoid segment radiance integral (E_u) computed like previously explained. The inversion method is used to derive the sampling component. Random values r are selected by transforming from uniform distribution to distance by a scaled logarithm function ($-\log(1 - (1 - E_u)r) / (\sigma_r c)$). Note, that the values differ for each trapezoid segment and they are dependent on geometry. In the case when the slope approaches zero ($c \rightarrow 0$), the sampling simplifies to multiplication by the distance between starting and end point. The result of the algorithm is distance offset by the starting distance $t_{C,S}$, representing distance from the origin along the camera ray. Visibility can be determined by sampling towards the light source from the point at that distance. Successive accumulation is used to solve the Monte Carlo integral (Eq. (2)).

5.2 Probability density function

The PDF matches perfectly the exponential component in the integral equation and cancels it (Eq. (2)). The probability of selecting each trapezoid segment is equal to the ratio of radiance scattered within the segment and the maximum (unoccluded) radiance ($P(x \in [t_{C,i-1}, t_{C,i}]) = \mathcal{T}_i / \mathcal{T}_{\text{total}}$). Taken on its own this function is defined over continuous domain ($x \in [t_{C,i-1}, t_{C,i}]$) and fits the definition of probability density function (PDF). The conditional probability of selecting a given distance, provided that a segment is selected, is equal to the transmittance at the particular point divided by the integral over transmittance within the segment ($P(x|x \in [t_{C,i-1}, t_{C,i}]) = \mathcal{T}'_i / \mathcal{T}_i = e^{-\sigma_r(x+d_m(x))} / \mathcal{T}_i$). We can assemble the final probability density function by considering where the point x is with respect to the three projected corners of the box on the camera ray ($t_{C,1}, t_{C,2}, t_{C,3}$),

$$\text{PDF}(x) = \begin{cases} \frac{\mathcal{T}_1}{\mathcal{T}_{\text{total}}} \cdot \frac{\mathcal{T}'_1(x)}{\mathcal{T}_1} & x < t_{C,1} \\ \frac{\mathcal{T}_2}{\mathcal{T}_{\text{total}}} \cdot \frac{\mathcal{T}'_2(x)}{\mathcal{T}_2} & t_{C,1} < x < t_{C,2} \\ \frac{\mathcal{T}_3}{\mathcal{T}_{\text{total}}} \cdot \frac{\mathcal{T}'_3(x)}{\mathcal{T}_3} & \text{otherwise} \end{cases} \quad (11)$$

We provide a straightforward validation of the equation in App. D. Further validation is performed by comparing the sample distribution against the probability density function. Examples are shown in Fig. 4 and Fig. 5. The probability density is transmittance over the unoccluded integral in both cases. The probability density of

the sampling function is computed numerically by using stratified random samples from uniform distribution. These samples are collected in 256 bins and normalized against the sample count to produce the final probability density function. It is visible that the numerical and analytically computed PDF match in both cases. The function is also affected by geometry and it is not necessarily monotonically decreasing as seen in the first example (Fig. 4). The presence of a corner leads to a function that is not smooth with a clear difference in its rate of change around the distance of the projected corner (Fig. 5). A *distance sampling* function will fail in the cases when the exit point along the camera ray is much closer than the distance from the origin towards the light source (Fig. 4). It will also deviate in all cases where the distance to the box boundary is not constant ($d_{E,S} \neq d_{E,E}$).

6 OPTIMIZED MONTE CARLO INTEGRATION

Finally, we provide an optimized integrator incorporating our sampling strategy with variable caching to reduce the number of redundant computations. Monte Carlo integration according to our implementation is performed by connecting a set of light rays along the camera ray. Each vertex is selected by performing importance sampling derived by choosing a trapezoid segment and inverting the analytic integration algorithm. Repeated traversal of the outline of the box is avoided in the final algorithm by caching values. Unoccluded integration follows the algorithm from Sec. 4 and caches values in a matrix \mathbf{K} . The cumulative distribution histogram is built by first computing the probability of selecting a segment ($G_k = \mathcal{T}_k / \mathcal{T}_{\text{total}}$). Then probabilities are summed up to the given segment ($H_k \leftarrow \sum_{i=1}^k G_i$). When performing sampling, the random values must be rescaled to fit the range of the selected trapezoid segment. The multiplicative weight is computed as the inverse probability ($D_k = 1/G_k$). While the summand is the previous CDF histogram value scaled by the multiplicative term ($M_k \leftarrow -H_{k-1}D_k$). Combined with the cached values (\mathbf{K}), it is enough to compute sample location based on random values selected from uniform distribution ($r \leftarrow \xi(k/N_{\text{sample}})$). Monte Carlo integration is performed over a number of samples (N_{sample}). Each sample from random distribution (r) is compared against the step functions of the histogram ($s \leftarrow \text{step}(H_1, r) + \text{step}(H_2, r)$). The sample is then transformed in segment coordinates ($r_s \leftarrow rD_s + M_s$). Distance is sampled according to the importance sampling (Sec. 5.1). However, the upper bound integral part ($\mathbf{K}_{s,1} = 1 - E_u$), slope ($\mathbf{K}_{s,2} = \sigma_r c$), size of segment along camera ray ($\mathbf{K}_{s,3} = t_{C,B} - t_{C,S}$), starting position along the camera ray ($\mathbf{K}_{s,4} = t_{C,S}$) are all cached in advance. The exponential function is cancelled by the probability density function. Therefore, the integration loop accumulates only visibility ($V_{\text{total}} = V_{\text{total}} + V(\mathbf{p}_o - t \omega_o)$). The final integral over visibility V_{total} is re-weighted by the number of samples N_{sample} , scattering coefficient σ_s and phase function $f(\omega_i, \omega_o)$ to compute the solution of the radiative transfer integral ($L = \frac{1}{N_{\text{sample}}} \sigma_s f(\omega_i, \omega_o) V_{\text{total}} \mathcal{T}_{\text{total}} L_i$). Only in the single-scattering case a ratio estimator with the same importance sampling scheme yields the same integration algorithm.

7 IMPLEMENTATION AND RESULTS

Now that we have a complete picture of how our techniques are built from theoretic perspective, we can discuss the practical implementation details and how they impact execution performance and quality. Furthermore, we compare our estimators and samplers against existing techniques and provide details about how they

integrate and perform in more practical approximate frameworks targeted at high-performance graphics.

Multiple integration algorithms were outlined within this work. Our aim was to use them in a real-time application built on Unity. Thus we have implemented everything as Unity scripts using the shading language in this engine. Our implementation does not rely on special features from this engine, so it can be ported to any rendering framework. Additionally, we provide a supplemental HTML page using JavaScript and WebGL for rendering that will assist reproduction of our work. We did not re-implement our technique in any offline rendering framework, but as we stated, we see it as a viable sampling strategy for that kind of application.

The stochastic integration algorithms using our techniques are: Monte Carlo with *whole volume distance sampling* (MCWVDS), *ratio estimator with distance sampling* (REDS) and *ratio estimator with equidistant sampling* (REES). MCWVDS is implemented following the optimized algorithm explained in Sec. 6. Ratio estimators use the basic framework defined in Sec. 3 paired with the analytic unoccluded radiance computation outlined in Sec. 4. REDS incorporates traditional *distance sampling* to generate samples proportional to transmittance along the current path segment, while REES distributes samples at equal distance which is jittered by noise. Those stochastic integrators converge to the same solution of single-scattering. Additionally, we implemented an approximate estimator based on moments following the method described by Peters *et al.* [4]. We defer to the code bundled with this previous publication for more details. This method relies on heuristics, so regardless of the sample count, it will deviate from the correct solution when dealing with complex occluders. Finally, we implemented *ray marching* (RM) which distributes samples at equal distance and jitters them with random offsets and *distance sampling* (MCDS) distributing samples according to transmittance along the current path segment and ignoring the overall shape of the medium. Both methods directly intersect the box at each step of the evaluation following the algorithm discussed in App. B to determine the distance from the scattering event to the box boundary along the light ray.

TABLE 2
Performance Benchmark

GPU	Scene	Sample count	Method	Time, ms
NVIDIA Quadro RTX 8000	Breakfast room	1024 spp	RM	11.0 ± 0.5
			MCWVDS	10.4 ± 0.1
	Fireplace room		RM	10.7 ± 0.5
			MCWVDS	9.5 ± 0.5
NVIDIA GeForce GTX 980 Ti	Breakfast room	256 spp	RM	10.3 ± 0.2
			MCWVDS	9.2 ± 0.1
	Fireplace room		RM	10.2 ± 0.1
			MCWVDS	9.3 ± 0.1
AMD Radeon Pro 560X	Breakfast room	128 spp	RM	37 ± 4*
		1024 ² pixels	MCWVDS	40 ± 4*
	Fireplace room		ME	29 ± 4*
		RM	24 ± 4*	
		MCWVDS	28 ± 4*	
		1024 ² pixels	ME	16 ± 1*

Performance benchmark. All integration techniques are benchmarked in Tab. 2 at 1080p resolution. We captured times on three different GPUs. Our MCWVDS integration approach depends on a logarithmic function which is not directly supported by the GPU and in the corner cases it fallbacks to linear segment sampling through branching. Regardless, these computations are

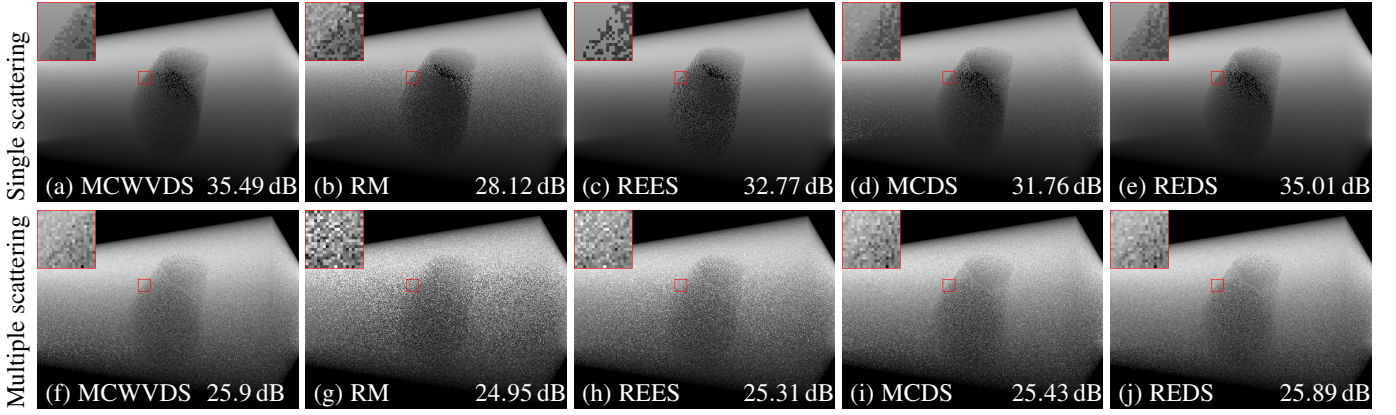


Fig. 6. **Integrator comparison.** Comparison of five estimators at low sample count (4spp). (a, f) Our Monte Carlo with *whole volume distance sampling* (MCWVDS) integrator, (b, g) *ray marching* (RM) integrator with stratified (jittered) random sampling, (c, h) ratio estimator with equidistant sampling (REES), (d, i) Monte Carlo with distance sampling (MCDS) integrator and (e, j) ratio estimator with distance sampling (REDS) integrator. (a, f) Our importance sampling algorithm results in less visible noise compared to all other techniques and the unoccluded integration algorithm helps reducing the noise when ratio estimators are used (c), (e).

less demanding than performing an intersection test to compute the distance between the scattering event and box boundary and then evaluating an exponential function used by ray marching. We noticed clear performance improvements on NVIDIA GPUs - possibly related to how the cached values are indexed. Precise comparison between the code emitted for different GPU ISAs is still difficult with the currently available tools. We used the GPU profiler in Unity to measure only passes of our technique. Our AMD setup (*) was on a MacBook Pro laptop where we captured the complete frame times which lead to high timing variance. GPU profiling in Unity was not available on that OS. It is also possible to implement a variant of ray marching which uses our technique to find the corners and replaces the intersection test with linear interpolation along cached box slice edges. Moment-based estimation [4] provides further performance improvements because it approximates the integral with lower number of dimensions by evaluating a heuristic formula dependent on depth and converting the result into moments which are linearly integrated.

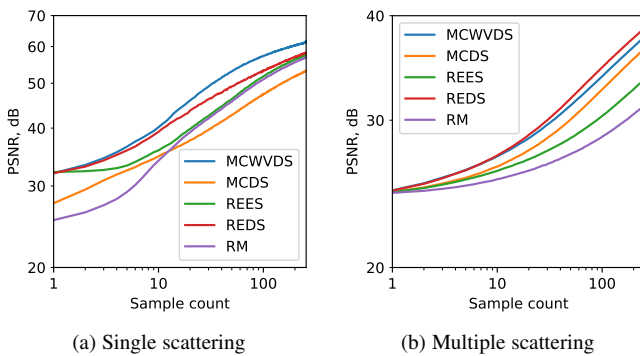


Fig. 7. **Integrator convergence rate.** Comparison of the integrators shown in Fig. 6. Our main technique using *whole volume importance sampling* achieves the fastest convergence rate. Ratio estimators combined with our unoccluded radiance computation follow since they converge quickly in unoccluded areas. Distance sampling in that case is closest to the distribution of transmittance, but overall misses important parts of the geometry which places it in second place and results in lower convergence rate when it is used with Monte Carlo integrator. Ray marching performs poorly at lower sample count, but it can outperform distance sampling at higher sample count, when considering single scattering.

Integrator comparison. We evaluated how our technique

integrates with different estimators on a simple scene with a single box medium and a sphere occluder (Fig. 6). We compared it against jittered ray marching (RM) which uses a slab test to find the medium boundaries at each step (Fig. 6 (b)). This example is considered as previous work since it relies on existing approaches. Overall, it fails to distribute samples proportional to transmittance. Our *whole volume distance sampling* technique MCWVDS (Fig. 6 (a)) follows closely the transmittance integrand. In the unoccluded sections, it converges in a single sample, while in the rest of the scene, it leads to noticeable variance reduction. Ratio estimator using our unoccluded analytic integral computation (Alg. 2) and ratio term using ray marching (REES) yields better convergence in the unoccluded parts of the scene, but leads to overall poor distribution of samples in the occluded parts. Distance sampling is another previous work (Fig. 6 (d)). It distributes samples proportional to the distance to the observer, however that does not perfectly match the transmittance integrand component, leading to higher variance. Combining this technique with a ratio estimator and our analytic unoccluded radiance integration (Fig. 6 (e) REDS), leads to improved variance reduction because in certain areas the distribution of transmittance follows more closely exponential decay. Ratio estimators are general enough that it is possible to combine them with more sophisticated integration techniques [43]. We demonstrate multiple scattering with our importance sampling and ratio estimators (Fig. 6 (f, g, h, i, j)) and show their potential as integrators in path tracing frameworks. These example figures were generated with the HTML page bundled in the supplemental material.

Integrator convergence rate. Fig. 6 shows results at a fixed number of samples per pixel. We further studied how the algorithms behave when increasing the sample count in Fig. 7 (a). Overall, our optimized integration algorithm (MCWVDS) achieves the fastest convergence rate. Ratio estimator based techniques similarly achieve lower variance in unoccluded areas which leads to higher convergence rate. Finally, ray marching and *distance sampling* that are considered as previous techniques perform the worst. Distance sampling performs good at very low sample count, but the fact that it relies on very simple assumptions regarding the geometry leads to lower convergence rate at higher sample count. In the case of multiple scattering distance sampling (Fig. 7 (b)), both ratio estimators and our importance sampling algorithm lead to lower

variance. The sample count reflects the number of camera rays and after each scattering event only a single shadow ray connect with the light source. The resulting integral corresponds to a single sample when considering only single-scattering integration spread over multiple path segments. Thus distance sampling retains its advantage over ray marching, even when multiple camera ray samples are taken into account. However, augmenting it with a ratio estimator or using our sampling technique leads to noticeable variance reduction.

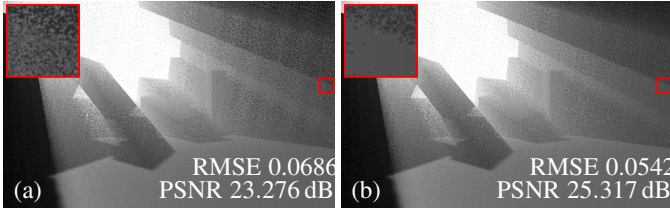


Fig. 8. **Destruction scene.** Comparison between (a) ray marching and (b) Monte Carlo importance sampled box according to our sampling approach at 4spp in a dynamic scene. RMSE and PSNR are computed over the entire image.

Destruction scene. We tested our MCWVDS integration algorithm on a dynamic scene involving destruction (Fig. 8) and compared it against ray marching (RM). The relatively simple geometry of the occluders and high extinction contribute to the overall higher variance reduction introduced by our algorithm compared to ray marching. We further evaluated our technique

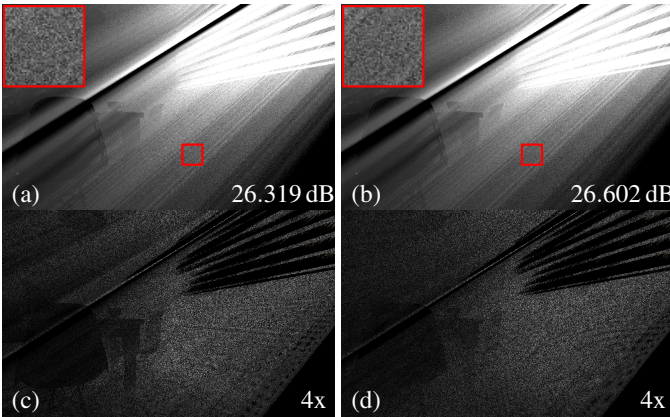


Fig. 9. **Breakfast room scene.** Comparison between (a) ray marching and (b) Monte Carlo integration with our importance sampling approach at 48spp with PSNR computed over the entire image. Difference images against ground truth of (c) ray marching and (d) Monte Carlo show the distribution of samples.

MCWVDS against standard research test scenes. Results for the *breakfast room* scene are shown in Fig. 9. Samples are preferentially distributed close to the camera and thus resulting in noise reduction. The more heavily occluded scene results in less variance reduction than the destruction scene (Fig. 8) because it leads to deviation of the integral from the probability density of the importance sampling function. As a rule of thumb our sampling approach has advantage in rooms with higher extinction, while ray marching allows taking more samples in unoccluded sections at the back of the room. Benchmark of the rendering performance is shown in Tab. 2.

We implemented also a variant of the moment-based single-scattering estimation [4] with our analytic unoccluded radiance integration technique (Alg. 2). We call it moment-based estimation

(ME). Since it relies on our technique, we don't consider it completely as previous work. It was tested in a very simple outdoor scenario where a set of basic shapes lie outside of the medium and thus cast detached shadows (Fig. 10). In that very simple scenario, the moment-based estimation matched really closely the results of the integration.

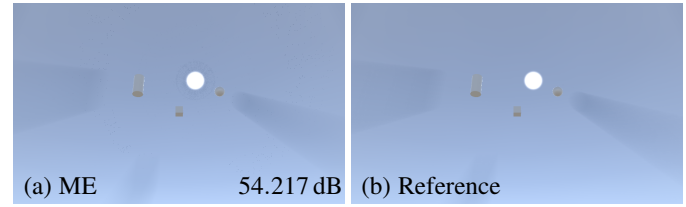


Fig. 10. **Simple outdoor scene.** Comparison between (a) moment-based estimation paired with our unoccluded radiance integration algorithm and (b) reference image using Monte Carlo integration in a simple outdoor scene. Moment-based estimation matches closely the reference.

Fireplace scene. We further tested the moment-based estimation technique against the *fireplace room* (Fig. 12) scene where the quality was still acceptable, but overestimation artifacts became more apparent. We noticed significant performance improvements when using this technique (cf. Tab. 2).

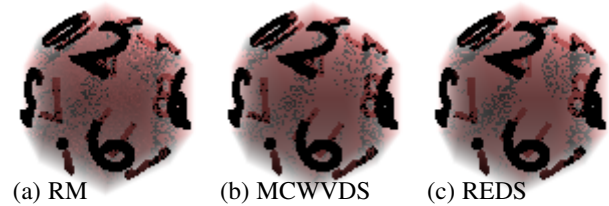


Fig. 11. **Dodehedral medium.** The general principles outlined for a box-shaped medium extend trivially to other convex medium shapes. All examples are rendered at 1spp.

Other medium shapes. We demonstrate in Fig. 11 that the principles outlined for a box-shaped medium also extend to other convex shapes. Concave shapes are also possible, but they will require more complex integration procedure. We expand the discussion in our supplemental material.

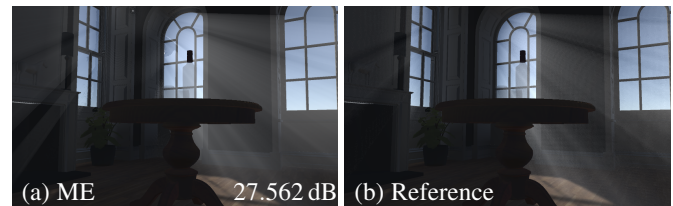


Fig. 12. **Fireplace scene.** Comparison between (a) moment-based estimation paired with our unoccluded radiance integration algorithm and (b) reference image using Monte Carlo integration in a high-quality indoor scene. The integration heuristic of the moment-based estimator starts to deviate from the integral and results in overestimation and artifacts.

Failure case of moments-based estimation. In the case of the *breakfast room* scene (Fig. 13), the moment-based estimation significantly deviates from the approximated integral and leads to severe artifacts. We do not use noise metrics, such as PSNR and RMSE, for those comparisons since the artifacts have very visible structure caused by the heuristic formulas used by this approach [4].

The original paper prescribes a set of heuristic terms to correct for overestimation issues. However, they fail to remove the issue and might further result in false shadows behind thinner objects which is equally distracting as an artifact. Possible future work is to enhance the heuristic to better match the integral.

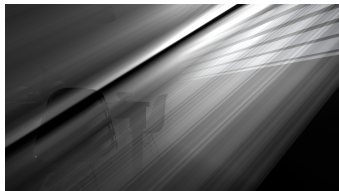


Fig. 13. **Failure case of moment-based estimation.** Underestimation is visible when using the prescribed corrective terms in the work by Peters Peters *et al.* [4]. The false shadows are caused by the more complex geometry that is not correctly estimated by the heuristic integration scheme.

Supplemental video. We provide a supplemental video with further comparisons of our techniques when the camera is in motion. Convergence rate of the optimized Monte Carlo algorithm (Alg. 8) is demonstrated by using a set of monotonically increasing number of samples. Furthermore, we provide a fully dynamic scene to outline the behavior in interactive scenes of our Monte Carlo integration algorithm and our analytic unoccluded radiance integration algorithm paired with the approach described by Peters *et al.* [4]. Moment-based estimation might exhibit artifacts when the heuristic deviates from the integral, but provides better performance. While our approach based on Monte Carlo integration leads to noise proportional to the number of samples. Denoising algorithms alleviate this problem. We defer to existing surveys for more details [44].

8 DISCUSSION

We presented a set of algorithms for integration and sampling that are directly applicable to Monte Carlo integrators and ratio estimators in indoor scenes. We demonstrated them predominately with a rectilinear box-shaped medium and showed how to extend them to generic shapes. Together, they can represent many scenes in video games, AR and VR experiences. Ratio estimators were used to shadow the unoccluded radiance computation by uniform collimated light which is part of our first main contribution. They provide greater flexibility regarding combining different sampling strategies, but lower convergence rate than our final technique. We are unaware of any prior work based on this kind of stochastic estimator for finite participating media which we believe is a valuable contribution on its own. The unoccluded radiance computation was further used in a moments-based approximate estimator that has better performance characteristics than all proposed techniques, but leads to visual artifacts caused by its heuristic nature. We leave as future work any perceptual studies that would evaluate whether those artifacts are objectionable enough to users of systems using our techniques.

We further derived a Monte Carlo sampling strategy which takes into account the entire shape of the medium and we showed the equivalence of estimators using it to a ratio estimator only when considering single-scattering integration. That is an important result showing where both theoretic frameworks lead to equivalent expressions. In the multiple scattering case they significantly deviate since the individual terms of the ratio estimator include

complex integrals. The superior convergence rate of our sampling strategy was quantified in a series of experiments (c.f. Fig. 6, Fig. 7). Our application targeted real-time performance and we optimized the integrator for this particular case, however the sampling strategy is compatible with path tracing frameworks, which makes it a contribution on its own. The final optimized Monte Carlo integrator has the potential to outperform previous approaches even at the same sample count (c.f. Tab. 2) and it is our final contribution.

One major application of our approach is room-scale AR where we surrounded the entire room by the volume to enable compositing of light shafts. The main advantage being that the windows to the outside virtual world are not affected by scattering and allow gameplay elements to be visible even at a great distance outside the room. This configuration was a useful building block to deliver the AR experience shown in Fig. 2.

Limitations. We derived our techniques under the assumption of homogeneous media, collimated light, uniform illumination and sampling decisions based on a single scattering event. We will provide in the next few paragraphs ideas about how to resolve those limitations in the future.

We have shown how to handle convex media, however, concave shapes will require more meticulous splitting to allow the use of our sampling technique (Alg. 7). Ratio estimators can be directly applied using the signed area to determine the sign for each linear segment when analytically computing the integral over transmittance. The approach can be extended to sampling heterogeneous medium by breaking it down into multiple shapes.

Single-scattering estimators based on Fourier coefficients or moments require the unoccluded term to be computed for all geometry contributing to the radiance scattered towards the camera. Taking the opposite route of starting with semi-infinite and subtracting the contribution of the environment outside the box leads to overall more complex equations which we demonstrate in the supplemental material. Those approximations can consequently lead to objectionable artifacts. Thus, we do not perceive this idea as viable future direction. Furthermore, moment-based estimators in general are more likely to fail with complex geometry since they rely on heuristics to compute the single-scattering integral as seen in Fig. 13. In the future, we are interested in developing heuristics and integration schemes to prevent these artifacts.

We demonstrated our sampling strategies in single- and multiple-scattering scenarios using the main principles of path tracing frameworks. Thus, we are able to leverage existing techniques to incorporate other types of luminaires by exploiting the linearity of light transport.

Finally, we demonstrated throughout our work how to apply finite media estimators to our specific problem. However, we think that those general principles can be further developed and applied to novel estimators and approaches to solving radiative transfer in finite media and they will be a valuable starting point for future research.

ACKNOWLEDGMENT

The authors would like to thank Llogari Casas, Elliott Baumbach, Jason Yeung and Tritia Medrano.

REFERENCES

- [1] J. Novák, I. Georgiev, J. Hanika, and W. Jarosz, "Monte carlo methods for volumetric light transport simulation," *Comput. Graph. Forum*, vol. 37, no. 2, 2018.

- [2] S. Chandrasekhar, *Radiative Transfer*, ser. Dover Books on Intermediate and Advanced Mathematics. Dover Publications, 1960.
- [3] O. Klehm, H.-P. Seidel, and E. Eisemann, "Filter-based real-time single scattering using rectified shadow maps," *Journal of Comput. Graph. Techniques (JCGT)*, vol. 3, no. 3, August 2014.
- [4] C. Peters, C. Münstermann, N. Wetzstein, and R. Klein, "Improved moment shadow maps for translucent occluders, soft shadows and single scattering," *Journal of Comput. Graph. Techniques (JCGT)*, vol. 6, no. 1, March 2017.
- [5] E. Heitz, S. Hill, and M. McGuire, "Combining analytic direct illumination and stochastic shadows," in *Proc. of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '18. New York, NY, USA: ACM, 2018.
- [6] E. Veach and L. J. Guibas, "Optimally combining sampling techniques for monte carlo rendering," in *Proc. of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '95. New York, NY, USA: ACM, 1995.
- [7] E. Cerezo, F. Pérez, X. Pueyo, F. J. Seron, and F. X. Sillion, "A survey on participating media rendering techniques," *Vis. Comput.*, vol. 21, no. 5, Jun. 2005.
- [8] M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016.
- [9] H. W. Jensen and P. H. Christensen, "Efficient simulation of light transport in scenes with participating media using photon maps," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '98. New York, NY, USA: Association for Computing Machinery, 1998, p. 311–320. [Online]. Available: <https://doi.org/10.1145/280814.280925>
- [10] J.-R. Jiménez, K. Myszkowski, and X. Pueyo, "Interactive global illumination in dynamic participating media using selective photon tracing," in *Proc. of the 21st Spring Conference on Computer Graphics*, ser. SCCG '05. New York, NY, USA: ACM, 2005.
- [11] W. Jarosz, D. Nowrouzezahrai, R. Thomas, P.-P. Sloan, and M. Zwicker, "Progressive photon beams," in *Proc. of the 2011 SIGGRAPH Asia Conference*, ser. SA '11. New York, NY, USA: ACM, 2011.
- [12] B. Bitterli and W. Jarosz, "Beyond points and beams: Higher-dimensional photon samples for volumetric light transport," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017.
- [13] X. Deng, S. Jiao, B. Bitterli, and W. Jarosz, "Photon surfaces for robust, unbiased volumetric density estimation," *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019.
- [14] K. Perlin and E. M. Hoffert, "Hypertexture," in *Proc. of the 16th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '89. New York, NY, USA: ACM, 1989.
- [15] E. P. Lafortune and Y. D. Willems, "Rendering participating media with bidirectional path tracing," in *Proc. of the Eurographics Workshop on Rendering Techniques '96*. London, UK: Springer-Verlag, 1996.
- [16] M. Pauly, T. Kollig, and A. Keller, "Metropolis light transport for participating media," in *Proc. of the Eurographics Workshop on Rendering Techniques 2000*. London, UK: Springer-Verlag, 2000.
- [17] C. Kulla and M. Fajardo, "Importance sampling techniques for path tracing in participating media," *Comput. Graph. Forum*, vol. 31, no. 4, Jun. 2012.
- [18] I. Georgiev, J. Křivánek, T. Hachisuka, D. Nowrouzezahrai, and W. Jarosz, "Joint importance sampling of low-order volumetric scattering," *ACM Trans. Graph.*, vol. 32, no. 6, Nov. 2013.
- [19] B. Sun, R. Ramamoorthi, S. G. Narasimhan, and S. K. Nayar, "A practical analytic single scattering model for real time rendering," in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH '05. New York, NY, USA: ACM, 2005.
- [20] V. Pegoraro, M. Schott, and P. Slusallek, "A mathematical framework for efficient closed-form single scattering," in *Proc. of Graphics Interface 2011*, ser. GI '11. University of Waterloo, Canada: Canadian Human-Computer Communications Society, 2011.
- [21] N. L. Max, "Atmospheric illumination and shadows," *SIGGRAPH Comput. Graph.*, vol. 20, no. 4, Aug. 1986.
- [22] K. Mitchell, "Volumetric light scattering as a post-process," *GPU Gems*, vol. 3, pp. 275–285, 2007.
- [23] Y. Dobashi, T. Yamamoto, and T. Nishita, "Interactive rendering of atmospheric scattering effects using graphics hardware," in *Proc. of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, ser. HWW '02. Aire-la-Ville, Switzerland: EG, 2002.
- [24] B. Toth and T. Umenhoffer, "Real-time Volumetric Lighting in Participating Media," in *Eurographics 2009 - Short Papers*. EG, 2009.
- [25] Y. Moro, R. Miyazaki, Y. Dobashi, and T. Nishita, "A fast rendering method for shafts of light in outdoor scene," 01 2006.
- [26] C. Wyman and S. Ramsey, "Interactive volumetric shadows in participating media with single-scattering," in *2008 IEEE Symposium on Interactive Ray Tracing*, Aug 2008.
- [27] A. Muñoz, "Higher order ray marching," *Comput. Graph. Forum*, vol. 33, no. 8, Dec. 2014.
- [28] Z. Ren, K. Zhou, S. Lin, and B. Guo, "Gradient-based interpolation and sampling for real-time rendering of inhomogeneous, single-scattering media," *Comput. Graph. Forum*, vol. 27, no. 7, 2008.
- [29] K. Zhou, Z. Ren, S. Lin, H. Bao, B. Guo, and H.-Y. Shum, "Real-time smoke rendering using compensated ray marching," in *ACM SIGGRAPH 2008 Papers*, ser. SIGGRAPH '08. New York, NY, USA: ACM, 2008.
- [30] C. Wyman, "Voxelized shadow volumes," in *Proc. of the ACM SIGGRAPH Symposium on High Performance Graphics*, ser. HPG '11. New York, NY, USA: ACM, 2011.
- [31] C. Wyman and Z. Dai, "Imperfect voxelized shadow volumes," in *ACM SIGGRAPH 2013 Talks*, ser. SIGGRAPH '13. New York, NY, USA: ACM, 2013.
- [32] I. Baran, J. Chen, J. Ragan-Kelley, F. Durand, and J. Lehtinen, "A hierarchical volumetric shadow algorithm for single scattering," *ACM Trans. Graph.*, vol. 29, no. 6, Dec. 2010.
- [33] T. Engelhardt and C. Dachsbacher, "Epipolar sampling for shadows and crepuscular rays in participating media with single scattering," in *Proc. of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '10. New York, NY, USA: ACM, 2010.
- [34] J. Chen, I. Baran, F. Durand, and W. Jarosz, "Real-time volumetric shadows using 1d min-max mipmaps," in *Symposium on Interactive 3D Graphics and Games*, ser. I3D '11. New York, NY, USA: ACM, 2011.
- [35] T. Annen, T. Mertens, P. Bekaert, H.-P. Seidel, and J. Kautz, "Convolution shadow maps," in *Proc. of the 18th Eurographics Conference on Rendering Techniques*, ser. EGSR'07. Aire-la-Ville, Switzerland: EG, 2007.
- [36] A. J. Preetham, P. Shirley, and B. Smits, "A practical analytic model for daylight," in *Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999.
- [37] N. Hoffman and A. J. Preetham, "Real-time light-atmosphere interactions for outdoor scenes," *Graphics Programming Methods*, p. 337–352, 2003.
- [38] S. G. Narasimhan and S. K. Nayar, "Vision and the atmosphere," *Int. J. Comput. Vision*, vol. 48, no. 3, Jul. 2002.
- [39] S. Hillaire, "A scalable and production ready sky and atmosphere rendering technique," *Comput. Graph. Forum*, vol. 39, no. 4, 2020.
- [40] A. T. Áfra, I. Wald, C. Benthin, and S. Woop, "Embree ray tracing kernels: Overview and new features," in *ACM SIGGRAPH 2016 Talks*, ser. SIGGRAPH '16. New York, NY, USA: ACM, 2016.
- [41] S. Laine, T. Karras, and T. Aila, "Megakernels considered harmful: Wavefront path tracing on gpu," in *Proc. of the 5th High-Performance Graphics Conference*, ser. HPG '13. New York, NY, USA: ACM, 2013.
- [42] P. Shirley and C. Wang, "Direct lighting calculation by monte carlo integration," in *Photorealistic Rendering in Computer Graphics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994.
- [43] J. Vorba, J. Hanika, S. Herholz, T. Müller, J. Křivánek, and A. Keller, "Path guiding in production," in *ACM SIGGRAPH 2019 Courses*, ser. SIGGRAPH '19. New York, NY, USA: ACM, 2019.
- [44] M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rouselle, P. Sen, C. Soler, and S.-E. Yoon, "Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering," *Comput. Graph. Forum*, vol. 34, no. 2, May 2015.

Zdravko Velinov is a researcher with a background in Computer graphics, he worked at Disney Research as a Postdoctoral Associate, and is organiser of ACM i3D 2020 and 2021. He is interested in physically based rendering and real-time interactive applications.

Kenny Mitchell is Technical Director of Rendering at Roblox and Professor of video game technology at Edinburgh Napier University. He is also co-founder/CTO of 3FINERY Ltd a spinout bringing real world objects to life and non-executive Director of Cobra Simulation Ltd for immersive training and simulation.