Eurographics Symposium on Rendering 2022
A. Ghosh and L.-Y. Wei
(Guest Editors)

Volume 41 (2022), Number 4

Temporally sliced photon primitives for time-of-flight rendering

Yang Liu   Shaojie Jiao   Wojciech Jarosz

Dartmouth College

Abstract

We derive a class of new Monte Carlo estimators for volumetric time-of-flight rendering, generalizing prior work on transient photon points and beams. Conceptually, our method starts with any steady-state photon primitive – like a photon plane, parallelepiped, or parallelotope – and slices it with a temporal wavefront, producing a primitive of one dimension lower. We show how different unbiased temporally sliced primitives arise by analytically integrating any four dimensions within a novel extended spatio-temporal path space formulation. The differences between these estimators reduce to the determinant of a 4×4 *Jacobian matrix, with columns dictated by the chosen dimensions. We then show how to combine the relative strengths of different sliced primitives using multiple importance sampling. Finally, we implement several of the new estimators enabled by our theory and compare them to each other as well as previous techniques.*

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing;*** *Modeling and simulation;*

1. Introduction

Recently, time-of-flight imaging has shown transformative potential in many fields of study. Compared to traditional imaging, time-of-flight imaging allows recording the time it takes light to arrive at the sensor with high precision. Disambiguating light paths in this way enables many exciting imaging applications like seeing around corners [KHDR11], or reconstructing objects observed through (or embedded within) scattering materials [HXK*14] (as required for autonomous driving in foggy conditions or non-invasive imaging through human tissue).

Such advances in imaging technologies benefit from corresponding advances in rendering techniques. Techniques that accurately simulate such new, and yet unexplored, imaging modalities hold great potential to: 1) help researchers explore trade-offs between new imaging system before physically constructing them, and 2) provide simulators and realistic training data so that data-hungry deep learning algorithms can continue to make rapid advances on computer vision tasks as our definition of imaging evolves.

Related work. We are interested in efficiently simulating time-of-flight imaging in scenes with participating media and refer to a recent survey [JMMG17] for a more comprehensive overview of time-of-flight imaging from a computer graphics (rendering) and computer vision (imaging) perspective. We adopt the nomenclature proposed by Pediredla et al. [PVG19] and use "transient rendering" to refer to simulating the evolution of the light over a continuum of time (whether with a delta or a non-delta time gate), producing multiple images; and "time-gated rendering" to refer to simulating a single image for a specific (and typically short) time gate. We are interested in efficient "time-of-flight rendering", which encompasses both transient and time-gated settings.

Simulating delta light pulses and (near) delta time-gated cameras is particularly challenging since it restricts the set of admissible light paths to a lower-dimensional temporal manifold, akin to the specular manifold [JM12] of caustic light paths in steady-state rendering. Mirroring the success of photon mapping [JC98; Jen01] for such steady-state problems, photon density estimation [JMM*14; Mar13] techniques were some of the earliest general approaches for time-of-flight rendering. Photon beam methods [JNSJ11; JZJ08; NNDJ12a; NNDJ12b] tend to reduce both variance and bias in steady-state rendering and even have progressive variants [HOJ08; JNT*11; KZ11] which ensure error vanishes in the limit. These have since been adapted to the time-of-flight domain [JMM*14; MGJ*19]. We express these prior photon point and beam time-of-flight rendering methods in a new framework, leveraging their ability to support complex light paths with efficient path reuse across pixels, while generalizing them to the higher-dimensional photon primitive of Deng et al. [DJBJ19]. This allows us to achieve unbiased estimation (even in the delta light pulse and time gate regimes) and improved efficiency and robustness via multiple importance sampling (MIS) [VG95b].

Path tracing [Kaj86; LW93; LW96; VG95a] has the benefit of being unbiased, but traditional strategies struggle to sample valid light paths within the temporal manifold. Pediredla et al. [PVG19] developed specially crafted "ellipsoidal connections" which give bidirectional path tracing some control in sampling paths with a desired temporal duration. This algorithm currently only handles surface interactions, though one of our estimators (Sec. 4.5) can be viewed as a generalization of such ellipsoidal connections to the volumetric setting. Jarabo et al. [JMM*14] devised strategies to generate path samples distributed more uniformly in time to accelerate time-resolved rendering. This is orthogonal to, and could in theory be combined with, our approach. Importance sampling techniques have also been developed for the context of optical coherence tomography [LKS11; PP16] and differential time-gated rendering [WCRZ21; YKC*21]. We are interested in general methods for time-of-flight rendering.

Contributions. Our work builds off of recent steady-state rendering techniques which extended photon density estimation to form even higher dimensional and *unbiased* variants such as photon "planes", "surfaces", and "volumes" [BJ17; DJBJ19]. Our primary contribution is a transient path space formulation that extends the steady-state "photon primitives" framework to the transient domain. This results in a versatile and powerful framework that allows us to derive a whole new family of robust estimators using a straightforward procedure. We show that existing approaches can be unified and viewed as specific instantiations of our formulation. More importantly, our formulation enables the derivations of novel estimators, including ones that can simulate delta time-gated volumetric transport without introducing bias, and ones which can be much more efficient in specific circumstances. Our theoretical formulation places few restrictions on how the estimators can be implemented. We discuss implementation considerations and demonstrate two prototype implementations for a selection of the new estimators, comparing their performance with previous work. We also demonstrate the possibility of combining these estimators using multiple importance sampling [VG95b].

2. Preliminaries

Our methodology is closely related to the photon primitives framework [DJBJ19] which provides a general approach to derive unbiased density estimators for steady-state volumetric rendering. We briefly review this approach and introduce the notation we use throughout this paper, before extending the framework to time-of-flight rendering in Sec. 3. Like Deng et al. we assume homogeneous media and leave efficient heterogeneous media to future work.

The photon primitive framework starts with a simplified extended path space formulation [HGJ*17] in which a full light path $\bar{\mathbf{z}} := \bar{\mathbf{x}}\bar{\mathbf{y}}$ is broken up into two disconnected subpaths: the photon subpath $\bar{\mathbf{x}} := (\mathbf{x}_l \dots \mathbf{x}_0)$ with $\mathbf{x}_l$ situated on a light source and the camera subpath $\bar{\mathbf{y}} := (\mathbf{y}_0 \dots \mathbf{y}_k)$ with $\mathbf{y}_k$ on the sensor (see Fig. 1). The positions of the intermediate scattering vertices are expressed in terms of a sequence of distances and directions:

$$\mathbf{x}_i = \mathbf{x}_l + \sum_{m=l}^{i+1} t_m \omega_m, \quad \text{and} \quad \mathbf{y}_j = \mathbf{y}_k + \sum_{n=k}^{j+1} s_n \psi_n, \tag{1}$$

where $\overline{\omega} := (\omega_l \dots \omega_1)$ and $\bar{t} := (t_l \dots t_1)$, and $\overline{\psi} := (\psi_1 \dots \psi_k)$ and $\bar{s} := (s_1 \dots s_k)$ are the sequences of directions and distances along the photon subpath and camera subpath, respectively.

The measurement contribution function $f(\bar{\mathbf{z}})$ is the product

$$f(\bar{\mathbf{z}}) = \underbrace{\prod_{i=1}^{l} f_\omega(\omega_i) f_t(t_i)}_{f(\bar{\mathbf{x}})} f_\omega^{1,1} K_3(\mathbf{g}) \underbrace{\prod_{j=1}^{k} f_\omega(\psi_j) f_t(s_j)}_{f(\bar{\mathbf{y}})}, \tag{2}$$

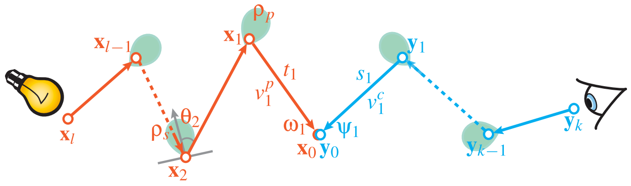


Figure 1: *The extended path space considers a photon subpath $\bar{\mathbf{x}} = (\mathbf{x}_l \dots \mathbf{x}_0)$ and a camera subpath $\bar{\mathbf{y}} = (\mathbf{y}_0 \dots \mathbf{y}_k)$ defined by a sequence of distances (t, s) and directions (ω, ψ) from opposite ends of the path. Once we extend this to the time-of-flight domain we also consider the sequence of light velocities along the two subpaths, $\overline{v^p} = (v_l^p \dots v_1^p)$ and $\overline{v^c} = (v_k^c \dots v_1^c)$.*

of distance functions

$$f_t(t_i) = T_r(t_i) V(\mathbf{x}_i, \mathbf{x}_{i-1}), \quad \text{with} \quad T_r(t) = \mathrm{e}^{-\sigma_t t}, \tag{3}$$

directional/scattering functions

$$f_\omega(\omega_i) = \begin{cases} L_e(\mathbf{x}_l, \omega_l) \cos\theta_i & \text{if } i = l \\ \rho_s(\omega_{i+1}, \omega_i) \cos\theta_i & \mathbf{x}_i \in \text{ surface and } i < l \\ \sigma_s \rho_p(\omega_{i+1}, \omega_i) & \mathbf{x}_i \in \text{ medium and } i < l, \end{cases} \tag{4}$$

as well as the phase function evaluated at the connection $f_\omega^{1,1}$ and a three-dimensional blurring kernel K_3 applied to the offset $\mathbf{g} := \mathbf{x}_0 - \mathbf{y}_0$ between the endpoints of the photon and camera subpaths. The distance function $f_t(t_i)$ is the product of the transmittance T_r and the visibility term $V(\mathbf{x}_i, \mathbf{x}_{i-1})$. The directional function $f_\omega(\omega_i)$ is the cosine-weighted emission function $L_e \cos\theta_i$, the cosine-weighted BSDF $\rho_s \cos\theta_i$, or the scattering coefficient-weighted phase function $\sigma_s \rho_p$, depending on where $\mathbf{x}_i$ resides.

The directional (4) and distance (3) functions f_ω and f_t in Eq. (2) are defined analogously when applied to the components of the camera subpath (ψ_j and s_j), though using the emitted importance $W_e(\mathbf{y}_k, \psi_k)$ at the sensor vertex (in place of the emitted radiance L_e) for the first case in Eq. (4) when $j = k$.

By introducing the set of all integration dimensions, $\bar{\xi} := \{\mathbf{x}_l, \bar{t}, \overline{\omega}; \mathbf{y}_k, \overline{\psi}, \bar{s}\}$, the path integral over all such paths can be written

$$I = \int_{\Xi} f(\bar{\xi}) K_3(\mathbf{g}(\bar{\xi})) f_\omega^{1,1} \, \mathrm{d}\bar{\xi}, \quad \text{with} \quad f(\bar{\xi}) = \prod_{\xi_i \in \bar{\xi}} f(\xi_i), \text{ and} \tag{5}$$

$$f(\xi_i) = \begin{cases} f_\omega(\xi_i) & \xi_i \in \{\mathbf{x}_l, \mathbf{y}_k, \overline{\omega}, \overline{\psi}\} \\ f_t(\xi_i) & \xi_i \in \{\bar{t}, \bar{s}\}. \end{cases} \tag{6}$$

2.1. Deriving a Photon Primitive Estimator

This integral is in general biased if the blurring kernel K_3 has a finite extent, but using a Dirac delta kernel

$$K_3(\mathbf{g}) = \delta^3(\mathbf{g}) = \delta(x(\mathbf{g}))\delta(y(\mathbf{g}))\delta(z(\mathbf{g})), \tag{7}$$

where $x(\mathbf{g})$, $y(\mathbf{g})$ and $z(\mathbf{g})$ represent the Cartesian coordinates of offset vector $\mathbf{g}$, removes bias by considering only paths where $\mathbf{x}_0$ and $\mathbf{y}_0$ coincide exactly.

Unfortunately, generating a valid path by sampling the two subpaths independently is impossible since the probability that $\mathbf{x}_0$ and $\mathbf{y}_0$ coincide exactly is zero. Deng et al. [DJBJ19] solved this by separating the set of integration dimensions $\overline{\xi}$ into two subsets: an analytic dimension set $\overline{\xi_a}$ that will be integrated analytically and the numerical dimension set $\overline{\xi_n}$ that will be sampled. This allows us to rewrite the integral as

$$I = \int_{\Xi_n} f(\overline{\xi_n}) \underbrace{\int_{\Xi_a(\overline{\xi_n})} f(\overline{\xi_a})\,\delta^3(\mathbf{g}(\overline{\xi_a}))\,f_\omega^{1,1}\,\mathrm{d}\overline{\xi_a}}_{I_a(\overline{\xi_n})}\,\mathrm{d}\overline{\xi_n}, \tag{8}$$

where we denote the analytic integral as I_a.

By choosing three analytic integration dimensions $\overline{\xi_a} = \{\xi_{a_1}, \xi_{a_2}, \xi_{a_3}\}$, the analytic integral $I_a(\overline{\xi_a})$ collapses to an evaluation of the path contribution,

$$I_a(\overline{\xi_n}) = \int_{\Xi_a(\overline{\xi_n})} f(\overline{\xi_a})\,\delta^3(\mathbf{g}(\overline{\xi_a}))\,f_\omega^{1,1}\,\mathrm{d}\overline{\xi_a} = \sum_r \frac{f(\overline{\xi_a^{*r}})\,f_\omega^{1,1}}{\left|\mathbf{J}^{\mathbf{g}}_{\overline{\xi_a}}(\overline{\xi_a^{*r}})\right|}, \tag{9}$$

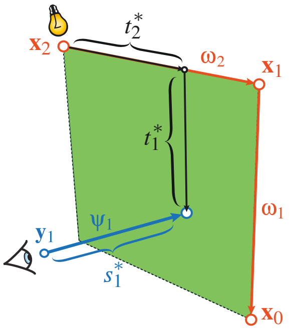
at the (potentially multiple) roots, $\overline{\xi_a^{*r}}$, of the equation $\mathbf{g}(\overline{\xi_a}) = \mathbf{0}$. The denominator contains the absolute value of the change-of-variable Jacobian determinant from $\mathbf{g}$ to $\overline{\xi_a}$ because we need to first express the delta kernel in terms of the integration variables before integrating it out.

The remaining dimensions $\overline{\xi_n}$ can now be estimated numerically using Monte Carlo. Each split of three analytic dimensions and remaining numeric ones results in an unbiased Monte Carlo estimator of the form

$$\langle I \rangle_{a,n} \approx \frac{f(\overline{\xi_n})\,I_a(\overline{\xi_n})}{p(\overline{\xi_n})} = \frac{f(\overline{\xi_n})}{p(\overline{\xi_n})} \sum_r \frac{f(\overline{\xi_a^{*r}})\,f_\omega^{1,1}}{\left|\mathbf{J}^{\mathbf{g}}_{\overline{\xi_a}}(\overline{\xi_a^{*r}})\right|}, \tag{10}$$

where $p(\overline{\xi_n})$ is the joint PDF of the numerically sampled dimensions.

Deng et al. [DJBJ19] limited three analytic dimensions to the last distance along the camera subpath, plus two dimensions along the photon subpath. The two analytic photon subpath dimensions can be interpreted as sweeping out a “photon surface”, which is intersected with a camera ray to form a complete path (see figure on the left). The formulation, is more general however, and in the next section we show how to adapt it to account for time-of-flight rendering.



3. A Framework for Rendering Time-gated Images

Here we describe how to adapt the photon primitives framework [DJBJ19] to time-of-flight rendering. Our key idea is that we can add a fourth temporal component to the vertices in light transport subpaths to fit time-of-flight rendering into the extended path space and the photon primitives framework. We denote 4D spatio-temporal quantities with a prime. A complete extended spatio-temporal path $\overline{\mathbf{z}}' := \overline{\mathbf{x}}'\overline{\mathbf{y}}'$ consists of two disconnected spatio-temporal subpath vertex sequences $\overline{\mathbf{x}}' = (\mathbf{x}'_l \ldots \mathbf{x}'_0)$ and $\overline{\mathbf{y}}' = (\mathbf{y}'_0 \ldots \mathbf{y}'_k)$. These are defined analogously to Eq. (1),

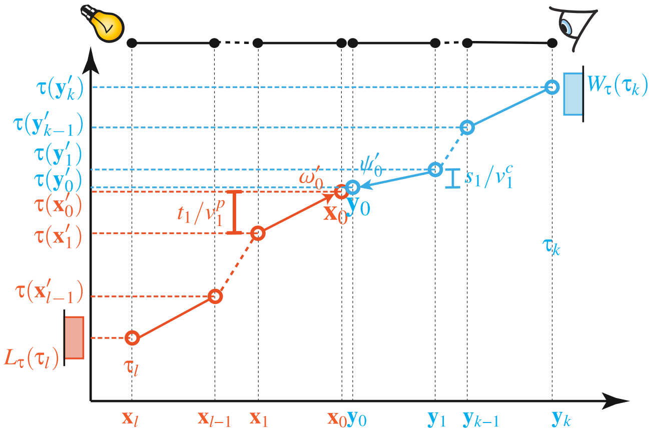


Figure 2: *Spatio-temporal diagram of light propagation in the extended path space. The photon subpath starts at time delay τ_l, and travels in the positive time direction. The camera subpath, on the other hand, starts at time delay τ_k and travels in the negative time direction. When the endpoints $\mathbf{x}'_0$ and $\mathbf{y}'_0$ overlap, it means we have found a complete path with total time duration τ_k. In the figure, both the time gate function W_τ and the emission pulse function L_τ are box functions.*

$$\mathbf{x}'_i = \mathbf{x}'_l + \sum_{m=l}^{i+1} t_m \omega'_m, \quad \mathbf{y}'_j = \mathbf{y}'_k + \sum_{n=k}^{j+1} s_n \psi'_n, \tag{11}$$

but where the starting vertex of the photon subpath $\mathbf{x}'_l = (\mathbf{x}_l, \tau_l)$ now includes the emission time τ_l, and the camera vertex $\mathbf{y}'_k = (\mathbf{y}_k, \tau_k)$ includes the detection time at the sensor τ_k. The subsequent vertices in Eq. (11) are now expressed in terms of 4D spatio-temporal directions $\overline{\omega}' := (\omega'_l \ldots \omega'_1)$ and $\overline{\psi}' := (\psi'_1 \ldots \psi'_k)$ defined as:

$$\omega'_i = (\omega_i, \frac{1}{v_i^p}), \quad \psi'_j = (\psi_j, -\frac{1}{v_j^c}), \tag{12}$$

where v_i^p and v_j^c denote the speed of light for the i-th and j-th segments along the photon and camera subpaths, respectively. Fig. 1 annotates a spatial path with some of these temporal quantities, and Fig. 2 illustrates this as a spatio-temporal diagram. Defined this way, the temporal component of the product $t_i\omega'_i$ represents the *propagation delay* between vertex $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$. For notational simplicity, we do not include *scattering delays* at the interior vertices, but these could easily be added if needed. Note that ω'_i and ψ'_j have opposite signs in the temporal dimension because spatially they point in opposing directions with respect to the flow of light.

The spatio-temporal offset vector $\mathbf{g}'$ is simply the difference (using 4D vector subtraction) between the endpoints of the two spatio-temporal subpaths: $\mathbf{g}' = \mathbf{x}'_0 - \mathbf{y}'_0$. Using the definitions in

Eqs. (11) and (12), the temporal component $\tau(\mathbf{g}')$ of the offset

$$\tau(\mathbf{g}') = \underbrace{\overbrace{\tau_l + \sum_{m=l}^{1} \frac{t_m}{v^p_m}}^{\tau(\mathbf{x}'_0)} - \overbrace{(- \sum_{n=1}^{k} \frac{s_n}{v^c_n}}}_{\tau(\overline{\mathbf{z}'})}\overbrace{{}+\tau_k)}^{\tau(\mathbf{y}'_0)}, \tag{13}$$

can be interpreted as comparing the total path duration $\tau(\overline{\mathbf{z}'})$ with the detection time at the sensor τ_k.

We can now define the spatio-temporal extension of the measurement contribution function (2) as

$$f(\overline{\mathbf{z}'}) = \underbrace{\prod_{i=1}^{l} f_\omega(\omega'_i) f_t(t_i)}_{f(\overline{\mathbf{x}'})} f^{1,1}_\omega \underbrace{K_3(\mathbf{g}')K_\tau(\mathbf{g}')}_{K_4(\mathbf{g}')} \underbrace{\prod_{j=1}^{k} f_\omega(\psi'_j) f_t(s_j)}_{f(\overline{\mathbf{y}'})}. \tag{14}$$

We use a 4D spatio-temporal kernel K_4, which is the product of the original kernel K_3 operating on the spatial dimensions of $\mathbf{g}'$, and a new kernel K_τ operating on the temporal dimension of $\mathbf{g}'$. Since we do not model relativistic effects, the distance/propagation function f_t remains unchanged (3), but the directional/scattering function f_ω now incorporates a time-dependent emission pulse function $L_\tau(\tau_l)$ indicating the weight of light emitted at time τ_l for the photon subpath starting vertex $\mathbf{x}_l$,

$$f_\omega(\omega'_i) = \begin{cases} L_e(\mathbf{x}_l, \omega_l) L_\tau(\tau_l) \cos\theta_i & \text{if } i = l \\ f_\omega(\omega_i) & \text{otherwise,} \end{cases} \tag{15}$$

and a time gate function $W_\tau(\tau_k)$ which weights the sensor response to light arriving at time τ_k for the camera subpath starting vertex $\mathbf{y}_k$,

$$f_\omega(\psi'_j) = \begin{cases} W_e(\mathbf{y}_k, \psi_k) W_\tau(\tau_k) \cos\theta_k & \text{if } j = k \\ f_\omega(\psi_j) & \text{otherwise.} \end{cases} \tag{16}$$

Any complete path from the light source to the camera with a travel time outside the range will have 0 contribution. For instance, the time gate could either be a finite box function

$$W_\tau(\tau_k) = \begin{cases} \frac{1}{\Delta\tau} & \text{if } \tau_k \in [\tau_{\min}, \tau_{\min} + \Delta\tau] \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

or a Dirac delta, and likewise for the emission pulse function L_τ.

Crucially, our formulation distinguishes between the time gate and emission pulse functions (W_τ, L_τ) and the temporal blurring kernel (K_τ). Although each is a 1D temporal function, they serve distinct roles. A non-delta blurring kernel would introduce bias, but a non-delta time gate or emission pulse would not since it is part of the contribution. We keep W_τ and L_τ distinct, though it is possible to remove one degree of freedom and use their convolution instead.

Finally, we can write our transient extension of the steady-state path integral (5) by integrating over the set of all spatial-temporal integration dimensions $\overline{\xi'} = \{\mathbf{x}_l, \tau_l, \bar{t}, \overline{\omega}'; \overline{\psi}', \bar{s}, \mathbf{y}_k, \tau_k\}$:

$$I = \int_{\Xi'} f(\overline{\xi'}) K_4(\mathbf{g}'(\overline{\xi'})) f^{1,1}_\omega \, \mathrm{d}\overline{\xi'}. \tag{18}$$

Camera unwarping. Sometimes it is desirable to render a scene using "camera unwarping" [VWJ*13] by ignoring the time delay along the camera ray $(\mathbf{y}'_{k-1}\mathbf{y}'_k)$. We can easily achieve this in our framework by setting the speed of light along the camera segment v^c_k to infinity. Conceptually, this prevents the time along camera rays from *warping* the shape of the light wavefronts, which can provide a clearer visualization of how light propagates *locally* in the scene. The opposite setting, "camera warping", accounts for camera ray propagation and provides a closer simulation of real-world sensors.

4. Temporally sliced photon primitives

Our extended transient path space formulation (18) can be readily used to express prior transient estimators. For instance, the transient photon mapping algorithm [JMM*14] estimates Eq. (18) as

$$\langle I \rangle \approx \frac{f(\overline{\xi'}) K_3(\mathbf{g}'(\overline{\xi'})) K_\tau(\mathbf{g}'(\overline{\xi'})) f^{1,1}_\omega}{p(\overline{\xi'})}, \tag{19}$$

where $p(\overline{\xi'})$ is the joint PDF of sampling all dimensions.

Equation (18), however, is more general than that, and we will show how to derive a plethora of new estimators, which we call *temporally sliced photon primitives*, by choosing to integrate some of the dimensions analytically.

4.1. A recipe for deriving temporally sliced photon primitives

The way we have generalized the path contribution function in Eq. (14) allows us to easily derive novel transient estimators using a procedure analogous to the steady-state one described in Sec. 2.1.

As in Eq. (7) we shrink the blurring kernel, but this time to a product of three spatial and one temporal Dirac delta kernels:

$$K_4(\mathbf{g}') = \delta^4(\mathbf{g}') = \delta(x(\mathbf{g}'))\delta(y(\mathbf{g}'))\delta(z(\mathbf{g}'))\delta(\tau(\mathbf{g}')). \tag{20}$$

If we then choose a subset of at least four dimensions $\overline{\xi'_a} \subset \overline{\xi'}$ to integrate analytically,

$$I = \int_{\Xi'_n} f(\overline{\xi'_n}) \underbrace{\int_{\Xi'_a(\overline{\xi'_n})} f(\overline{\xi'_a}) \, \delta^4(\mathbf{g}'(\overline{\xi'_a})) \, f^{1,1}_\omega \, \mathrm{d}\overline{\xi'_a}}_{I_a(\overline{\xi'_n})} \, \mathrm{d}\overline{\xi'_n}. \tag{21}$$

we can write a transient estimator analogously to steady-state (10) as

$$\langle I \rangle_{a,n} \approx \frac{f(\overline{\xi'_n}) I_a(\overline{\xi'_n})}{p(\overline{\xi'_n})}, \tag{22}$$

where $p(\overline{\xi'_n})$ is the joint PDF of all numerically integrated dimensions, and $I_a(\overline{\xi'_n})$ is the analytically pre-integrated contribution function.

If we choose any four dimensions to pre-integrate:

$$\overline{\xi'_a} = \{\xi'_{a_1}, \xi'_{a_2}, \xi'_{a_3}, \xi'_{a_4}\}, \tag{23}$$

then the delta functions collapse the integral I_a to an evaluation

$$I_a(\overline{\xi'_n}) = \int_{\Xi'_a(\overline{\xi'_n})} f(\overline{\xi'_a}) \delta^4(\mathbf{g}'(\overline{\xi'_a})) f^{1,1}_\omega \, \mathrm{d}\overline{\xi'_a} = \sum_r \frac{f(\overline{\xi^{*r}_a}) f^{1,1}_\omega}{\left|\mathbf{J}^{\mathbf{g}'}_{\overline{\xi'_a}}(\overline{\xi^{*r}_a})\right|}, \tag{24}$$

at the (possibly multiple) roots $\overline{\xi^{*r}_a}$ of the delta constraint $\mathbf{g}'(\overline{\xi'_a}) = \mathbf{0}$. As before, the Jacobian in the denominator arises from re-expressing the delta kernels—initially defined in spatio-temporal space—in terms of the analytic integration variables. For any choice of four dimensions, this is simply a 4×4 matrix with the partial derivatives of vector $\mathbf{g}'$ with respect each of the chosen analytic dimensions:

$$\mathbf{J}^{\mathbf{g}'}_{\overline{\xi_a}}(\overline{\xi_a^{*r}}) = \begin{bmatrix} | & | & | & | \\ \frac{\partial \mathbf{g}'(\overline{\xi_a^{*r}})}{\partial \xi'_{a_1}} & \frac{\partial \mathbf{g}'(\overline{\xi_a^{*r}})}{\partial \xi'_{a_2}} & \frac{\partial \mathbf{g}'(\overline{\xi_a^{*r}})}{\partial \xi'_{a_3}} & \frac{\partial \mathbf{g}'(\overline{\xi_a^{*r}})}{\partial \xi'_{a_4}} \\ | & | & | & | \end{bmatrix} . \tag{25}$$

Deriving and implementing a new estimator in our framework requires two steps:

1. Solve for each root, $\overline{\xi_a^{*r}}$, of the delta constraint $\mathbf{g}'(\overline{\xi'_a}) = \mathbf{0}$;
2. Derive and evaluate the absolute value of the Jacobian determinant, $|\mathbf{J}^{\mathbf{g}'}_{\overline{\xi_a}}(\overline{\xi_a^{*r}})|$, at each root.

Each estimator derived in our framework has a geometric interpretation which is closely related to its steady-state counterpart. If we assume one of the analytic dimensions is the last distance along the camera subpath, then we can view one of our new estimators as a steady-state photon primitive *sliced* by the temporal kernel to produce a temporal wavefront. When the camera response and light pulse time τ_k, τ_l are numerically sampled, each such sliced photon primitive is one dimension lower than the steady-state counterpart, e.g. a sliced parallelopiped is a planar transient primitive (assuming camera unwarping).

In the following subsections, we derive a few estimators enabled by our theory to demonstrate the generality of this recipe.

4.2. Sliced Photon Planes

In steady-state rendering, the photon plane is one typical example. It chooses the last two distances (t_1, t_2) along the photon subpath and the last distance along the camera subpath s_1 as analytic dimensions. Here, we derive an estimator that also pre-integrates these three dimensions analytically and additionally analytically integrates the time gate τ_k, meaning its analytic dimensions are

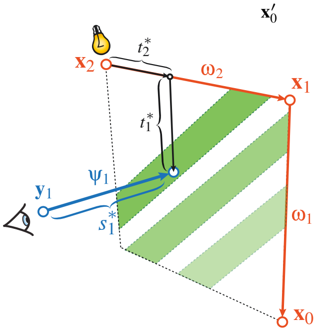
$$\overline{\xi_a} = \{t_2, t_1, s_1, \tau_k\} , \tag{26}$$

and I_a becomes

$$I_a = \int_{\Xi'} f_t(t_2) f_t(t_1) f_t(s_1) W_\tau(\tau_k) \delta^4(\mathbf{g}'(\overline{\xi'_a})) f_\omega^{1,1} \, \mathrm{d}t_2 \, \mathrm{d}t_1 \, \mathrm{d}s_1 \, \mathrm{d}\tau_k . \tag{27}$$

Following our recipe, we need to find the roots of the equation,

$$\mathbf{g}'(\overline{\xi'_a}) = \underbrace{\mathbf{x}'_2 + \omega'_2 t_2 + \omega'_1 t_1}_{\mathbf{x}'_0} - \underbrace{(\mathbf{y}'_1 + \psi'_1 s_1)}_{\mathbf{y}'_0} = \mathbf{0}. \tag{28}$$



Conceptually, this is just the intersection of a camera ray with a plane and then further restricted to a diagonal band (shown in green in the inset figure to the left). This band is essentially a visualization of the time gate function $W_\tau(\tau_k)$ which slides diagonally along the photon plane as time advances (faded green bands). Both the plane and the diagonal strip become curved when considering camera warping.

Next, we need the Jacobian matrix. When τ_k is chosen to be one of the analytic dimensions, the column in the Jacobian matrix that corresponds to $\frac{\partial \mathbf{g}'}{\partial \tau_k}$ is simply $[0,0,0,1]$ since τ_k only affects the temporal domain. The Jacobian matrix is

$$\mathbf{J}^{\mathbf{g}'}_{t_2,t_1,s_1,\tau_k} = \begin{bmatrix} | & | & | & | \\ \omega_2 & \omega_1 & -\psi_1 & 0 \\ | & | & | & | \\ \frac{1}{v_2^p} & \frac{1}{v_1^p} & -\frac{1}{v_1^c} & 1 \end{bmatrix}, \tag{29}$$

which gives the same determinant as the steady-state photon plane:

$$\left| \mathbf{J}^{\mathbf{g}'}_{t_2,t_1,s_1,\tau_k} \right| = |\omega_2 \times \omega_1 \cdot \psi_1|. \tag{30}$$

This estimator is unbiased, but the analytically preintegrated time gate plays a similar role as the temporal blur in biased transient photon beams, allowing the primitives to be intersected with the camera ray. If the time gate is a delta function, then we could instead use the emission pulse τ_l as the fourth analytic dimension. This creates an identical primitive since it simply negates the last column of the Jacobian matrix, and the absolute value of the determinant remains the same. If both the time gate and emission pulse are delta functions, then extra blur will need to be introduced or else the primitive will have 0 probability of intersecting the camera ray.

4.2.1. Re-interpreting Transient Photon Beams

There's a noticeable visual similarity between the sliced photon plane and transient photon beam in previous work. In fact, the photon beam can be viewed as replacing one analytic dimension and keeping the blurring kernel on it a box function instead of a delta function, giving us the analytic dimension set

$$\overline{\xi_a} = \{B, t_1, s_1, \tau_k\} . \tag{31}$$

which includes the blur B instead of t_2. The partial derivative of $\mathbf{g}$ over the blur is $\frac{\omega_1 \times \psi_i}{|\omega_1 \times \psi_i|}$ as the blur is perpendicular to both the camera ray and the last photon segment.

$$\mathbf{J}^{\mathbf{g}'}_{B,t_1,s_1,\tau_k} = \begin{bmatrix} | & | & | & | \\ \frac{\omega_1 \times \psi_i}{|\omega_1 \times \psi_i|} & \omega_1 & -\psi_1 & 0 \\ | & | & | & | \\ \frac{1}{v_1^p} & \frac{1}{v_1^p} & -\frac{1}{v_1^c} & 1 \end{bmatrix}, \tag{32}$$

$$\left| \mathbf{J}^{\mathbf{g}'}_{B,t_1,s_1,\tau_k} \right| = \left| \omega_1 \times (-\psi_1) \cdot \frac{\omega_1 \times \psi_i}{|\omega_1 \times \psi_i|} \right| = |\omega_1 \times \psi_1| . \tag{33}$$

This gives us the same Jacobian determinant as 1D-blurred transient photon beams as in previous work [MGJ*19].

In previous work [MGJ*19], a transient photon beam is interpreted as having a 2D spatio-temporal blur with one spatial blur and one temporal blur. Our formulation, however, shows that only a 1D spatial blur is needed if we use the time gate τ_k or emission time τ_l as one of the analytic dimensions. In essence, this allows the time gate W_τ or emission pulse function L_τ to take the place of the temporal blur kernel K_τ, without introducing bias. In fact, since τ_k and τ_l reside in the same space, we could trivially integrate both analytically, which would simplify to using the convolution of W_τ and L_τ instead of the temporal blur kernel. The ability to remove the temporal blur implies that the estimator in our framework would have less bias, and forming a progressive variant would only require reducing the single 1D spatial blur kernel to ensure consistency.

In the next section, we show an estimator applicable to the case where both the time gate and the emission pulse are delta functions.

4.3. Sliced Photon Parallelepipeds

The steady-state photon parallelepipeds uses three edges $\mathbf{x}_3\mathbf{x}_2$, $\mathbf{x}_2\mathbf{x}_1$, $\mathbf{x}_1\mathbf{x}_0$ to form a parallelepiped, and a numerically sampled point $\mathbf{y}_1 + s_1\psi_1$ to query it. Since we need one more analytic integration dimension for unbiased delta time-gated rendering, we can choose to also preintegrate s_1, giving us $\overline{\xi'_a} = \{t_3, t_2, t_1, s_1\}$. This particular choice of $\overline{\xi'_a}$ yields a primitive we call a delta-sliced photon parallelepiped. I_a in Eq. (21) then becomes:

$$I_a = \int_{\Xi'} f_t(t_3) f_t(t_2) f_t(t_1) f_t(s_1) \delta^4(\mathbf{g}'(\overline{\xi'_a})) f_\omega^{1,1} \,\mathrm{d}t_3\,\mathrm{d}t_2\,\mathrm{d}t_1\,\mathrm{d}s_1. \quad (34)$$

Solving $\mathbf{g}'(\overline{\xi'_a}) = \mathbf{0}$ is trivial since it is a linear system. For this specific set of $\overline{\xi'_a}$, we have

$$\mathbf{g}'(\overline{\xi'_a}) = \underbrace{\mathbf{x}'_3 + \omega'_3 t_3 + \omega'_2 t_2 + \omega'_1 t_1}_{\mathbf{x}'_0} - \underbrace{(\mathbf{y}'_1 + \psi'_1 s_1)}_{\mathbf{y}'_0} = \mathbf{0} \quad (35)$$

$$\Leftrightarrow \mathbf{J}^{\mathbf{g}'}_{t_3,t_2,t_1,s_1} \overline{\xi'_a} = \mathbf{y}'_1 - \mathbf{x}'_3 \quad (36)$$

where

$$\mathbf{J}^{\mathbf{g}'}_{t_3,t_2,t_1,s_1} = \begin{bmatrix} | & | & | & | \\ \omega_3 & \omega_2 & \omega_1 & -\psi_1 \\ | & | & | & | \\ \frac{1}{v^p_3} & \frac{1}{v^p_2} & \frac{1}{v^p_1} & \frac{1}{v^c_1} \end{bmatrix}. \quad (37)$$

For simplicity, we will only consider the case where all the preintegrated photon subpath segments have the same speed of light v: $v = v^p_3 = v^p_2 = v^p_1$. When rendered using camera unwarping ($1/v^c_1 = 0$), the Jacobian of the transient photon parallelepiped is

$$\left|\mathbf{J}^{\mathbf{g}'}_{t_3,t_2,t_1,s_1}\right| = \frac{1}{v}\left|\psi_1 \cdot (\omega_1 \times \omega_2 + \omega_2 \times \omega_3 + \omega_3 \times \omega_1)\right|. \quad (38)$$

We can also interpret the Jacobian in Eq. (38) as the dot product between the camera direction ψ_i and a scaled *temporal surface normal* $\mathbf{n} = (\omega_1 \times \omega_2 + \omega_2 \times \omega_3 + \omega_3 \times \omega_1)$, multiplied by an inverse speed of light term $\frac{1}{v}$.

To figure out the geometry of the temporal surface in this setting, we can start by expressing the spatial intersection point $\mathbf{x}^*_0$ using the constraint on the temporal dimension. The temporal constraint in the 4th row of the linear system in Eq. (36) can be written as

$$\frac{1}{\Lambda}(t_3 + t_2 + t_1) = 1, \quad \text{with} \quad \Lambda := v(\tau(\mathbf{y}'_1) - \tau(\mathbf{x}'_3)). \quad (39)$$

Inserting Eq. (39) into the first three spatial dimensions of $\mathbf{x}'_0$ defined in Eq. (35) and regrouping the terms yields:

$$\mathbf{x}^*_0 = (\mathbf{x}_3 + \omega_3\Lambda)(1 - \frac{t_2}{\Lambda} - \frac{t_1}{\Lambda}) + (\mathbf{x}_3 + \omega_2\Lambda)\frac{t_2}{\Lambda} + (\mathbf{x}_3 + \omega_1\Lambda)\frac{t_1}{\Lambda}, \quad (40)$$

which gives the Barycentric coordinate formulation of a point on the triangle formed by vertices $\mathbf{p}_3 = \mathbf{x}_3 + \omega_3\Lambda$, $\mathbf{p}_2 = \mathbf{x}_3 + \omega_2\Lambda$, $\mathbf{p}_1 = \mathbf{x}_3 + \omega_1\Lambda$. Please refer to Fig. 3 for the visualization of this triangle. The normal of this triangle gives the direction of the scaled temporal surface normal $\mathbf{n}$, while the area of this triangle is a scaled version of the length of $\mathbf{n}$.

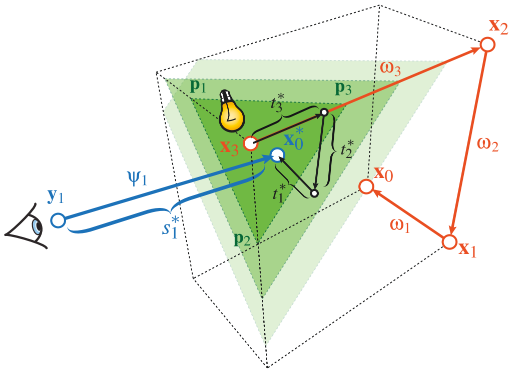


Figure 3: *Slicing a photon parallelepiped with a delta time gate in the camera-unwarped case results in the green triangle traveling diagonally through the parallelepiped as time progresses.*

The procedures described above can also be thought of as slicing a 3D steady-state photon primitive with a time constraint to yield a 2D transient primitive.

4.4. Sliced Photon Parallelotope

The slicing operation described above can be applied to even higher dimensions. Here we describe how to derive a 3D temporal primitive by slicing a 4D steady-state primitive generalized from photon parallelepipeds. We start by replacing the camera subpath segment distance s_1 in the analytic dimension set $\overline{\xi'_a}$ with an additional propagation distance t_4: $\overline{\xi'_a} = \{t_4, t_3, t_2, t_1\}$. The resulting steady-state primitive is a parallelotope whose edges are the last 4 segments in the photon subpath. With this particular choice of $\overline{\xi'_a}$, we can rewrite I_a in Eq. (21) as:

$$I_a = \int_{\Xi'} f_t(t_4) f_t(t_3) f_t(t_2) f_t(t_1) \delta^4(\mathbf{g}'(\overline{\xi'_a})) f_\omega^{1,1} \,\mathrm{d}t_4\,\mathrm{d}t_3\,\mathrm{d}t_2\,\mathrm{d}t_1. \quad (41)$$

Solving $\mathbf{g}'(\overline{\xi'_a}) = \mathbf{0}$ under this new definition of $\overline{\xi'_a}$ gives:

$$\mathbf{J}^{\mathbf{g}'}_{t_4,t_3,t_2,t_1} \overline{\xi'_a} = \mathbf{y}'_0 - \mathbf{x}'_4, \quad (42)$$

where Jacobian matrix

$$\mathbf{J}^{\mathbf{g}'}_{t_4,t_3,t_2,t_1} = \begin{bmatrix} | & | & | & | \\ \omega_4 & \omega_3 & \omega_2 & \omega_1 \\ | & | & | & | \\ \frac{1}{v^p_4} & \frac{1}{v^p_3} & \frac{1}{v^p_2} & \frac{1}{v^p_1} \end{bmatrix} \quad (43)$$

converts variables from $\overline{\xi'_a}$ to $\mathbf{g}'$.

If we assume the speed of light is constant inside the parallelotope $v = v^p_4 = v^p_3 = v^p_2 = v^p_1$, and insert Eq. (43) into Eq. (41), we obtain the preintegrated intensity of the photon parallelotope:

$$I_a = \frac{f_\omega^{1,1} f_t(t^\star_4) f_t(t^\star_3) f_t(t^\star_2) f_t(t^\star_1)}{\left|\mathbf{J}^{\mathbf{g}'}_{t_4,t_3,t_2,t_1}\right|}, \quad (44)$$

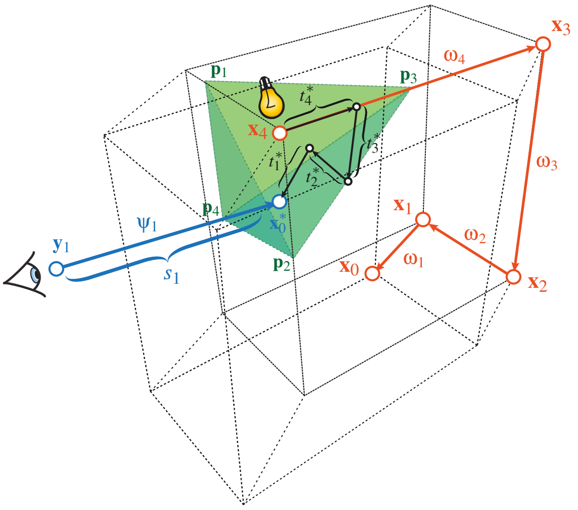


Figure 4: *The geometry of a camera-unwarped delta-sliced photon parallelotope is a tetrahedron.*

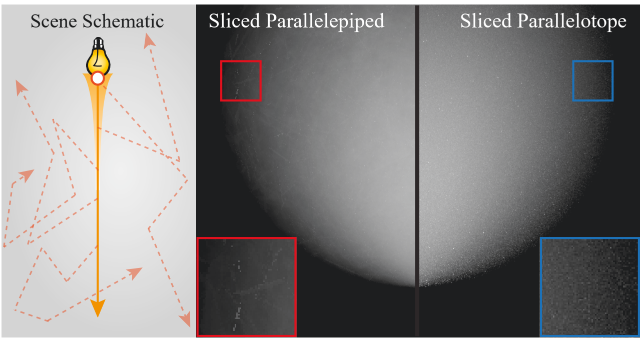


Figure 5: *This searchlight scene (left) contains an infinite homogeneous medium illuminated by a collimated beam source aiming downwards. We compare 4-bounce-only transport simulated using sliced parallelepipeds (left half) and sliced parallelotopes (right half), using a delta time gate. Both estimators use 2M photons and are unbiased, but variance in sliced parallelepipeds produces structured artifacts due to analytic evaluation along camera rays, while sliced parallelotopes produce high frequency noise due to independent numerical distance sampling along camera rays.*

where $\left|\mathbf{J}^{\mathbf{g}'}_{t_4,t_3,t_2,t_1}\right|$ is the determinant of matrix $\mathbf{J}$ defined in Eq. (43), and $\{t_4^*, t_3^*, t_2^*, t_1^*\}$ are the solutions for Eq. (42). The Jacobian $\left|\mathbf{J}^{\mathbf{g}}_{t_4,t_3,t_2,t_1}\right|$ here can also be viewed as a scaled volume for a tetrahedron formed by vertices $\mathbf{p}_4 = \mathbf{x}_4 + \omega_4$, $\mathbf{p}_3 = \mathbf{x}_4 + \omega_3$, $\mathbf{p}_2 = \mathbf{x}_4 + \omega_2$, $\mathbf{p}_1 = \mathbf{x}_4 + \omega_1$. Fig. 4 illustrates the geometric construction.

Since the sliced parallelotope estimator samples the distance along the camera ray numerically, it has a different kind of noise pattern compared to the sliced parallelepiped estimator (see Fig. 5).

4.5. Sliced Photon Balls/VPLs

The main limitation of the sliced photon parallelepiped and sliced photon parallelotope (44) estimators is that they need 3 bounces and 4 bounces respectively, so they cannot handle single scattering. We can create an unbiased single scattering estimator by setting $\overline{\xi'_a} = \{\cos\theta_1, \phi_1, t_1, s_1\}$. The Jacobian matrix for the change-of-variable then becomes:

$$\mathbf{J}^{\mathbf{g}'}_{\cos\theta_1,\phi_1,t_1,s_1} = \begin{bmatrix} | & | & | & | \\ \frac{\partial\mathbf{g}}{\partial\cos\theta_\mathbf{1}} & \frac{\partial\mathbf{g}}{\partial\phi_1} & \omega_1 & -\psi_1 \\ | & | & | & | \\ 0 & 0 & \frac{1}{v_1^p} & \frac{1}{v_1^c} \end{bmatrix}, \tag{45}$$

and the estimator becomes:

$$I_a = \frac{f_\omega^{1,1} f_\omega(\omega_1^*) f_t(t_1^\star) f_t(s_1^\star)}{\left|\mathbf{J}^{\mathbf{g}}_{\cos\theta_1,\phi_1,t_1,s_1}\right|}, \tag{46}$$

where ω_1^*, t_1^*, s_1^* are the solutions for $\mathbf{g}'(\overline{\xi'}) = \mathbf{0}$.

For the camera-unwarped case, the Jacobian for the sliced photon ball can be written as $\left|\mathbf{J}^{\mathbf{g}'}_{\cos\theta_1,\phi_1,t_1,s_1}\right| = \frac{1}{v} {t_1^*}^2 \omega_1^* \cdot \psi_1$. A steady-state photon ball is equivalent to placing a point light (a VPL) at a path vertex. Slicing it with the temporal wavefront then creates a sphere, which we need to intersect with the camera ray due to the analytic integration of s_1 (see Fig. 6 left). The Jacobian can be interpreted as the dot product between the camera ray direction ψ_1 and the surface normal ω_1^* of this spherical wavefront, scaled to account for standard inverse-squared falloff ${t_1^*}^2$ from the point light and the speed of light $\frac{1}{v}$.

If we consider the camera-warped case, the sliced photon ball is a generalization of Pediredla et al.'s ellipsoidal connections [PVG19] to participating media. Geometrically, the temporal wavefront becomes an ellipsoid (see Fig. 6 right) with foci at $\mathbf{x}_1$ and $\mathbf{y}_1$, and the length of its major axis is $v(\tau(\mathbf{x}'_1) - \tau(\mathbf{y}'_1))$. The resulting Jacobian can be expressed as $\left|\mathbf{J}^{\mathbf{g}'}_{\cos\theta_1,\phi_1,t_1,s_1}\right| = \frac{1}{v} {t_1^*}^2 (\omega_1^* + \psi_1) \cdot \psi_1$, which is a dot product between the camera ray direction and a scaled normal of the ellipsoid at the path connection.

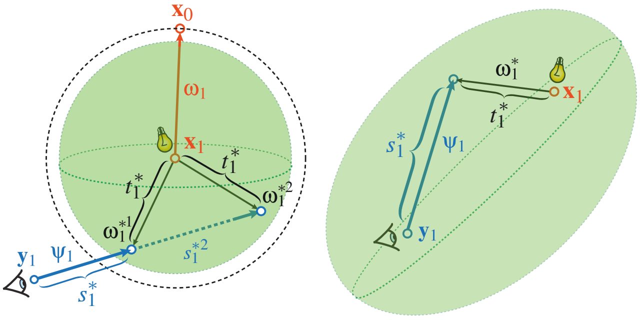


Figure 6: *A delta-sliced photon ball produces a sphere (left) if camera-unwarped and an ellipsoid (right) when camera-warped.*

4.6. Combining estimators using MIS

We can also combine different estimators using MIS in a similar manner as steady-state photon primitives [DJBJ19].

When evaluating each estimator, we need to first find the roots of the equation $\mathbf{g}'(\overline{\xi_a'})$. Each root corresponds to a complete light-path. For estimators that use delta kernels, these estimators are all expressed in a path space of the same dimensionality with the same path contribution function, allowing us to view them as path sampling strategies.

To combine multiple estimators we use score-based MIS [Jen18]. At each hitpoint, we re-evaluate the score using the other estimator(s), which requires computing the Jacobians of those estimators for the paths corresponding to the hitpoint. Since each estimator shares the same path contribution in our formulation, the differences between them are only in the their different Jacobians, which MIS reinterprets as being part of the pdf. This allows us to MIS any unbiased estimator expressed in our framework.

Fig. 7 demonstrates this in a delta time-gated scenario where MISing the sliced photon ball and the sliced photon parallelepiped provides substantial variance reduction compared to either estimator in isolation.

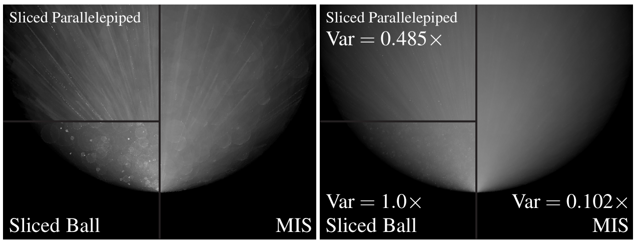


Figure 7: *In the same searchlight scene as in Fig. 5, here we visualize the third bounce and compare the performance of sliced parallelepipeds, sliced balls, and their MIS combination for a delta time gate. Left: rendered with a time budget of 10s. Right: rendered with a time budget of 3m. MIS is able to reduce variance by alleviating the singularities, which show up as bright lines for sliced parallelepipeds and bright circles for sliced balls.*

5. Implementation

Interactive CPU-GPU renderer. We first implemented a collection of our new sliced estimators in a hybrid CPU-GPU renderer based on the one released by Bitterli and Jarosz [BJ17]. This is implemented in Javascript + WebGL and allows interactive exploration in a web browser. We provide the full implementation of this interactive demo within the supplemental material.

The renderer simulates the searchlight problem (see schematic in Fig. 5), in a scene containing a collimated beam source in an infinite homogeneous medium.

On the CPU, we first trace photons through the medium starting at the beam source, generate un-sliced (steady-state) primitives along the photon paths, and send the un-sliced primitives to the GPU to rasterize. We construct the un-sliced primitives because this implementation aims to produce transient animations, and the un-sliced primitives can be reused for all transient frames.

In a fragment shader on the GPU, we evaluate whether the time delay of the path corresponding to each fragment falls within the time gate window, and calculates the path contribution. For the camera-warped sliced ball primitive, we draw a full-screen quad since the ellipsoid-shaped temporal wavefront will always encompass all camera rays. The results, visualized in Fig. 8, confirm that sliced photon primitives can potentially improve the quality for time-gated volumetric rendering.

Offline ray tracing renderer. Motivated by the promising results from the hybrid renderer, we then implemented some of our sliced estimators in Bitterli's open-source Tungsten renderer [Bit18]. We choose to use sliced photon balls to handle 1-2 scattering events, and combine sliced photon balls with sliced photon parallelepiped using MIS to handle 3+ scattering. We also uses sliced photon balls to deal with surface-to-medium transport.

We first trace photon subpaths from the light source, computing the time delay for each vertex along the way. Then we iterate over all bounces, generate corresponding photon primitives and store them in a BVH. Instead of storing un-sliced primitives as in the hybrid CPU-GPU renderer, here we first sample a specific time in the time gate and store the corresponding delta-sliced photon primitive. This simplifies our intersection code, and improves the performance of our renderer since the delta-sliced primitives usually have a much smaller bounding box than their steady-state counterparts. When the emission function L_e or the surface BSDF ρ_s is a Dirac delta function (e.g. collimated beam light sources and smooth dielectric surfaces), we fall back to using photon beams.

Long and short primitives. As in the work by Bitterli et al. [BJ17], whenever we include a distance as one of the analytic dimensions, we can choose to evaluate the corresponding transmittance in the style of a "long" or a "short" primitive [KGH*14]. A long primitive will have infinite extent and requires evaluating the exponential transmittance (3) along the path segment. A short primitive is bounded by using a binary track-length estimator of the transmittance along the path segment: returning 1 up to the sampled free-flight distance, and 0 beyond. For estimators that analytically integrate more than one distance term, this choice is possible at each one.

In the previous sections, the mathematical formulations used the long primitives but the illustrations correspond to the short ones since it is harder to convey geometry on unbounded primitives. In our implementations, we also use the short primitives since they are easier to put into an acceleration structure.

6. Results

We evaluate our offline renderer implementation in Fig. 9 by comparing the MIS combination of our sliced parallelepiped and sliced ball estimators against progressive transient photon beams [MGJ*19] in three scenes: SUBSURFACE SCATTERING, CORNELL BOX and VOLUMETRIC CAUSTIC, using the camera-unwarped setting. All the scenes use a Dirac delta as both the time gate function and the emission pulse function. We implement both methods in the same renderer, and render each frame at equal time using 8 cores of Intel Xeon E5-2640V3 on a Linux cluster. For the progressive transient photon beams, we use the spatio-temporal variant with parameters $\alpha = 2/3$, $\beta = 1/2$, and shoot 10K photons in each progressive

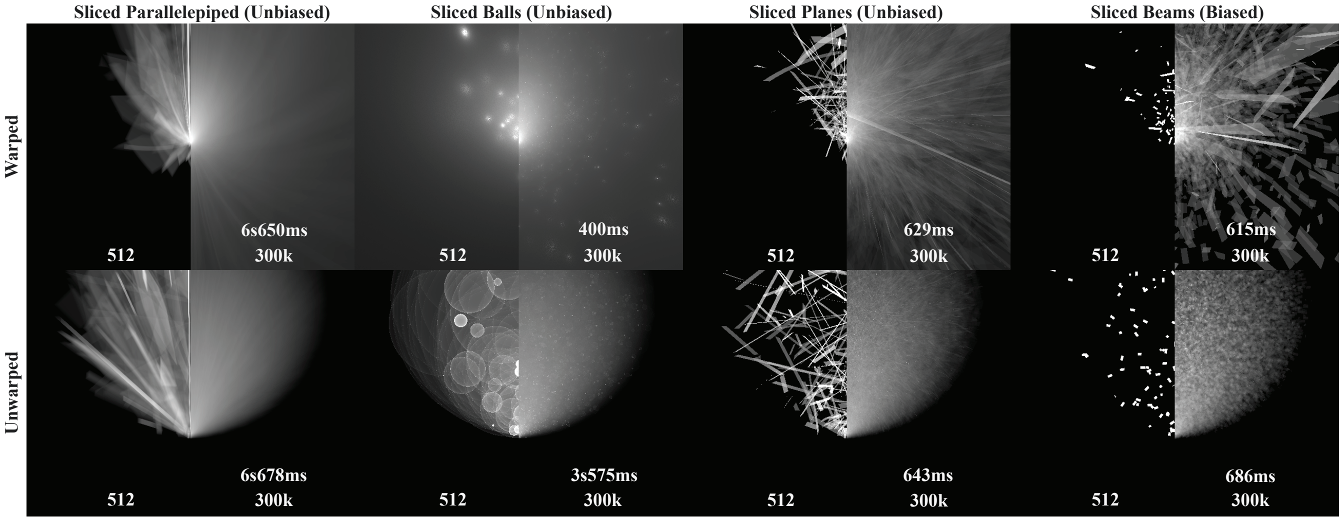


Figure 8: *Comparison of various sliced estimators in the searchlight scene from Fig. 5 using a box time gate. For each estimator we show a split with 512 (left) and 300k (right) photon paths and include the render time for the latter (the render time for the former is roughly equal for all estimators since it is dominated by overhead). For the same number of traced paths, our estimators (first 3 columns) outperform transient photon beams [MGJ*19] (4th column) while providing a variety of strengths and weaknesses.*

iteration. For paths that cannot be handled by either the sliced parallelepiped or the sliced ball estimator (such as caustics), we fall back to using progressive transient photon beams. Please refer to the supplement materials for time-gated animations of these scenes.

The SUBSURFACE SCATTERING scene in Fig. 9 (top) is inspired by the searchlight problem from Habel et al. [HCJ13]. We shine a beam of light at a semi-infinite highly-scattering medium at an incident angle of 30°. (Parameters: $\sigma_s = 1$, $\sigma_a = (0.01, 0.1, 1.0)$, $g = 0$.) The comparison row shows the evolution of light transport at three different times. Our MIS estimator dramatically reduces variance of the multiple scattering compared to progressive transient photon beams, by a factor of roughly 20–40.

The CORNELL BOX scene in Fig. 9 (middle) shows photons from a collimated beam source hitting a diffuse box underneath and gradually revealing the geometry in the scene. Our MIS estimator still shows strengths in handling multiple bounces even with surfaces and occluders. The variance reduction—while still on the order of 2–4×—is less dramatic than in the simpler SUBSURFACE SCATTERING scene. Our method robustly improves variance only for the categories of transport where we can combine multiple estimators with MIS. Visible singularities remain around surfaces since we only have one estimator that handles surface-to-media transport. This suggests that using MIS to combine sliced beams and sliced balls may be beneficial in such scenarios.

The VOLUMETRIC CAUSTIC scene in Fig. 9 (bottom) compares the two methods in a setting with complex volumetric caustics. Some of the photons leaving the point light source travel through a glass ball (IOR = 1.5), delaying the temporal wavefront of refracted light. Our MIS estimator still excels at simulating the multiple scattering, especially at later times in the sequence. We have to fall back to progressive transient photon beams for simulating specular paths like the refractive caustic wavefront, so these paths remain blurry. These paths are also noisier in our result since, in the same render time, we are not able to trace as many specular paths for transient beams due to the comparatively more expensive sliced parallelepipeds and sliced balls we use for multiple scattering. Using different photon counts for specular and non-specular paths may balance their sources of error better.

Finally, the temporal wavefronts are much sharper in our results compared to those from progressive transient photon beams, as shown in the four renders on the left in CORNELL BOX and VOLUMETRIC CAUSTIC. This is because our estimator can produce an unbiased estimate when using a delta time gate with a delta light pulse, while the progressive transient photon beams method needs more iteration steps to reduce its temporal bias.

7. Conclusions, Limitations & Future Work

We have presented a novel path integral framework and a general recipe for deriving novel unbiased estimators for time-of-flight rendering. Our formulation is also general enough to express many existing techniques, providing a new perspective of previous work. We instantiated this recipe to derive and implement several novel estimators, each taking the same basic form with a differing Jacobian term. Each estimator has different strengths, weaknesses, and applicable sceniaros, and we showed how to combine complementary estimators using MIS to reduce variance.

While we have demonstrated the potential for this theory, there are still limitations and also exciting directions for future work.

Our theory supports both camera warping and unwarping, and while we've shown working examples of both, our support for camera warping is less efficient. Camera warping is more challenging because the shape of the photon primitives can become view-dependent. This raises interesting questions of how to efficiently

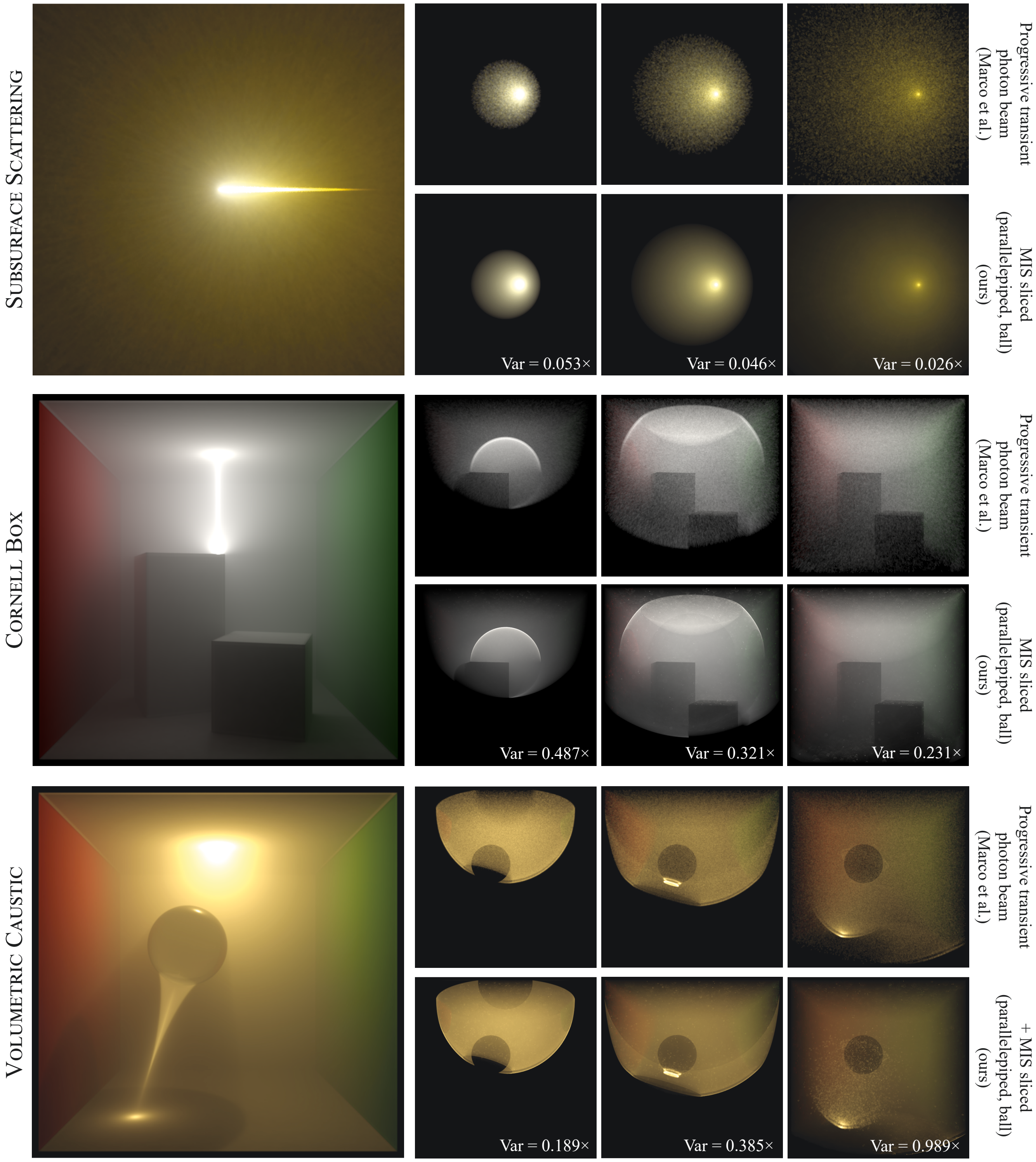

Subsurface Scattering
Progressive transient photon beam (Marco et al.)
MIS sliced (parallelepiped, ball) (ours)
Var = 0.053×
Var = 0.046×
Var = 0.026×
Cornell Box
Progressive transient photon beam (Marco et al.)
MIS sliced (parallelepiped, ball) (ours)
Var = 0.487×
Var = 0.321×
Var = 0.231×
Volumetric Caustic
Progressive transient photon beam (Marco et al.)
+ MIS sliced (parallelepiped, ball) (ours)
Var = 0.189×
Var = 0.385×
Var = 0.989×

Figure 9: *For each scene, we show the steady-state render on the left and two rows of time-of-flight renders on the right. Each top row is rendered using only progressive transient photon beams, while each bottom row additionally uses our MIS estimator. Each column in a row corresponds to a different time. The time-of-flight renders include only volumetric transport for clarity. For the* ***Subsurface Scattering*** *scene, each image took 1.5 minutes to render; for the other 2 scenes, each image took 10 minutes. For the* ***Cornell Box*** *scene, the steady-state render and the two renders on the left have their exposure set to -2EV to avoid overexposing.*

render sliced primitives, or build effective acceleration structures for ray tracing. Here, different use cases (e.g. rendering a single time-gated image vs. a full transient animation vs. an interactive application allowing dynamic camera or scene manipulation) would likely lead to different design decisions both in how to represent these sliced primitives, and which primitives are most effective. We have only explored a tiny slice of this design space in our interactive rasterization-based GPU implementation and CPU path tracing-based implementation, but more work is needed.

For now, we only support MIS between unbiased estimators, and thus our method performs relatively poorly for paths that cannot currently be MISed (Fig. 10). Since we have shown how to cast some prior biased transient estimators into our framework, it should be theoretically possible to MIS between biased and unbiased transient primitives to increase robustness, as has been previously done for steady-state rendering by VCM/UPS/UPBP [GKDS12; HPJ12; KGH*14]. This would be particularly helpful for surface-to-media transport where we currently only have one unbiased estimator (the sliced photon ball). Enabling MIS with biased strategies here could dramatically reduced singularities that currently occur near surfaces.

Our framework allows estimators to be combined using MIS, and we have found this to dramatically reduce variance/singularities in the Jacobians. While the MIS is theoretically straightforward, the shear number of possible estimators at our disposal does make it challenging to (correctly) implement them by hand. Since the unbiased estimators have the same expected value, each one can serve as a validation baseline for the others. This does help debug each individual strategy, but most strategies are only applicable to certain types of light paths (e.g. 2+ transport for a sliced photon plane), which complicates such validations and the logic needed to determine which MIS weights are even needed. A more automated way to implement such estimators could dramatically reduce common errors. Domain-specific programming languages have shown promise for similar problems [ALLD17].

Depending on characteristics of the imaging setup, other physical light transport effects might need to be considered. One natural extension is to support polarization and fluorescence effects [JA18] as the phase is closely related to the travel time of light. Heterogeneous media is also ubiquitous, but our implementations currently do not support it. Besides the medium coefficients, the index of refraction could varying spatially, which affects the speed of light within the medium. It would be interesting to see if our estimators could be adapted to such scenarios by building off the refractive RTE [ABW14; PCS*20].

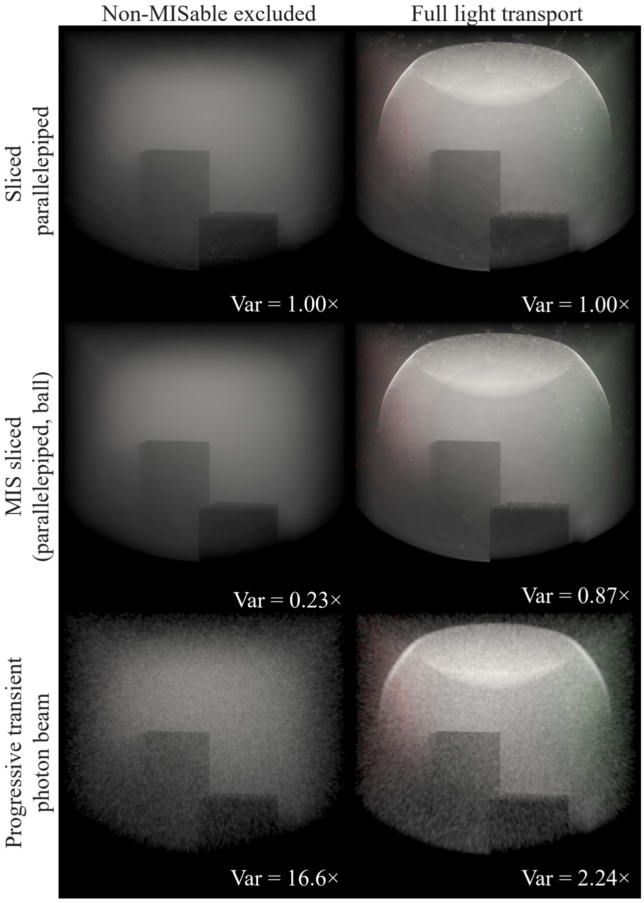


Figure 10: *In the left column we show only the light transport for which we can MIS two of our estimators, while the right column includes full light transport. When available, MIS greatly reduces variance, but these improvements are diluted by the singularities in non-MISable paths when computing full light transport. All images take 5 minutes using the* CORNELL BOX *scene described in Sec. 6.*

References

[ABW14] AMENT, M., BERGMANN, C., and WEISKOPF, D. "Refractive radiative transfer equation". *ACM Transactions on Graphics* 33.2 (Apr. 2014). ISSN: 0730-0301. DOI: 10/gbf323 11.

[ALLD17] ANDERSON, L., LI, T.-M., LEHTINEN, J., and DURAND, F. "Aether: an embedded domain specific sampling language for Monte Carlo rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 36.4 (July 2017). ISSN: 0730-0301. DOI: 10/ggfg5n 11.

[Bit18] BITTERLI, B. *Tungsten Renderer*. 2018. URL: https://github.com/tunabrain/tungsten/ 8.

[BJ17] BITTERLI, B. and JAROSZ, W. "Beyond points and beams: higher-dimensional photon samples for volumetric light transport". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 36.4 (July 2017). ISSN: 0730-0301. DOI: 10/gfznbr 2, 8.

[DJBJ19] DENG, X., JIAO, S., BITTERLI, B., and JAROSZ, W. "Photon surfaces for robust, unbiased volumetric density estimation". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38.4 (July 2019). DOI: 10/gf6rx9 1–3, 7.

[GKDS12] GEORGIEV, I., KŘIVÁNEK, J., DAVIDOVIČ, T., and SLUSALLEK, P. "Light transport simulation with vertex connection and merging". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 31.6 (Nov. 2012). ISSN: 0730-0301. DOI: 10/gbb6q7 11.

[HCJ13] HABEL, R., CHRISTENSEN, P. H., and JAROSZ, W. "Photon beam diffusion: a hybrid Monte Carlo method for subsurface scattering". *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 32.4 (June 2013). ISSN: 1467-8659. DOI: 10/f445m4 9.

[HGJ*17] HACHISUKA, T., GEORGIEV, I., JAROSZ, W., KŘIVÁNEK, J., and NOWROUZEZAHRAI, D. "Extended path integral formulation for volumetric transport". *Proceedings of EGSR (Experimental Ideas & Implementations)*. Eurographics Association, June 2017. DOI: 10/gfznb3 2.

[HOJ08] Hachisuka, T., Ogaki, S., and Jensen, H. W. "Progressive photon mapping". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 27.5 (Dec. 1, 2008). issn: 07300301. doi: 10/cn8h39 1.

[HPJ12] Hachisuka, T., Pantaleoni, J., and Jensen, H. W. "A path space extension for robust light transport simulation". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 31.6 (Nov. 1, 2012). issn: 0730-0301. doi: 10/gbb6n3 11.

[HXK*14] Heide, F., Xiao, L., Kolb, A., Hullin, M. B., and Heidrich, W. "Imaging in scattering media using correlation image sensors and sparse convolutional coding". *Optics Express* 22.21 (2014). doi: 10/gfz5mc 1.

[JA18] Jarabo, A. and Arellano, V. "Bidirectional rendering of vector light transport". *Computer Graphics Forum* 37.6 (Sept. 1, 2018). issn: 0167-7055. doi: 10/gdvk6m 11.

[JC98] Jensen, H. W. and Christensen, P. H. "Efficient simulation of light transport in scenes with participating media using photon maps". *Annual Conference Series (Proceedings of SIGGRAPH)*. ACM Press, July 1998. doi: 10/b64p36 1.

[Jen01] Jensen, H. W. *Realistic Image Synthesis Using Photon Mapping*. Natick, MA, USA: AK Peters, Ltd., 2001. isbn: 1-56881-147-0 1.

[Jen18] Jendersie, J. "Path Throughput Importance Weights". June 4, 2018. arXiv: 1806.01005 [cs] 8.

[JM12] Jakob, W. and Marschner, S. "Manifold exploration: a Markov chain Monte Carlo technique for rendering scenes with difficult specular transport". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 31.4 (July 2012). issn: 0730-0301. doi: 10/gfzq4p 1.

[JMM*14] Jarabo, A., Marco, J., Munoz, A., Buisan, R., Jarosz, W., and Gutierrez, D. "A framework for transient rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 33.6 (Nov. 2014). issn: 15577368. doi: 10/gfznb8 1, 2, 4.

[JMMG17] Jarabo, A., Masia, B., Marco, J., and Gutierrez, D. "Recent advances in transient imaging: a computer graphics and vision perspective". *Visual Informatics* 1.1 (2017). doi: 10/gfz5mh 1.

[JNSJ11] Jarosz, W., Nowrouzezahrai, D., Sadeghi, I., and Jensen, H. W. "A comprehensive theory of volumetric radiance estimation using photon points and beams". *ACM Transactions on Graphics* 30.1 (Jan. 1, 2011). issn: 0730-0301. doi: 10/fcdh2f 1.

[JNT*11] Jarosz, W., Nowrouzezahrai, D., Thomas, R., Sloan, P.-P., and Zwicker, M. "Progressive photon beams". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 30.6 (Dec. 2011). issn: 07300301. doi: 10/fn5xzj 1.

[JZJ08] Jarosz, W., Zwicker, M., and Jensen, H. W. "The beam radiance estimate for volumetric photon mapping". *Computer Graphics Forum (Proceedings of Eurographics)* 27.2 (Apr. 2008). issn: 1467-8659. doi: 10/bjsfsx 1.

[Kaj86] Kajiya, J. T. "The rendering equation". *Computer Graphics (Proceedings of SIGGRAPH)* 20.4 (Aug. 1986). issn: 0097-8930. doi: 10/cvf53j 1.

[KGH*14] Křivánek, J., Georgiev, I., Hachisuka, T., Vévoda, P., Šik, M., Nowrouzezahrai, D., and Jarosz, W. "Unifying points, beams, and paths in volumetric light transport simulation". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 33.4 (July 2014). issn: 0730-0301. doi: 10/f6cz72 8, 11.

[KHDR11] Kirmani, A., Hutchison, T., Davis, J., and Raskar, R. "Looking around the corner using ultrafast transient imaging". *International Journal of Computer Vision (IJCV)* 95.1 (2011). doi: 10/dsk2dt 1.

[KZ11] Knaus, C. and Zwicker, M. "Progressive photon mapping: A probabilistic approach". *ACM Transactions on Graphics* 30.3 (May 1, 2011). issn: 07300301. doi: 10/bcw2ph 1.

[LKS11] Lima, I. T., Kalra, A., and Sherif, S. S. "Improved importance sampling for Monte Carlo simulation of time-domain optical coherence tomography". *Biomedical optics express* 2.5 (2011). doi: 10/bv83k6 2.

[LW93] Lafortune, E. P. and Willems, Y. D. "Bi-directional path tracing". *Proceedings of the International Conference on Computational Graphics and Visualization Techniques (Compugraphics)* (Alvor, Portugal). Vol. 93. Alvor, Portugal, Dec. 1993 1.

[LW96] Lafortune, E. P. and Willems, Y. D. "Rendering participating media with bidirectional path tracing". *Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*. Vienna: Springer-Verlag, June 1996. isbn: 978-3-211-82883-0. doi: 10/fzth2c 1.

[Mar13] Marco, J. "Transient Light Transport in Participating Media". PhD thesis. Universidad de Zaragoza, 2013 1.

[MGJ*19] Marco, J., Guillén, I., Jarosz, W., Gutierrez, D., and Jarabo, A. "Progressive transient photon beams". *Computer Graphics Forum* 38.1 (Mar. 2019). issn: 1467-8659. doi: 10/gfvr9w 1, 5, 8, 9.

[NNDJ12a] Novák, J., Nowrouzezahrai, D., Dachsbacher, C., and Jarosz, W. "Progressive virtual beam lights". *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 31.4 (June 2012). issn: 01677055. doi: 10/gfzndw 1.

[NNDJ12b] Novák, J., Nowrouzezahrai, D., Dachsbacher, C., and Jarosz, W. "Virtual ray lights for rendering scenes with participating media". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 31.4 (July 2012). issn: 0730-0301. doi: 10/gbbwk2 1.

[PCS*20] Pediredla, A., Chalmiani, Y. K., Scopelliti, M. G., Chamanzar, M., Narasimhan, S., and Gkioulekas, I. "Path tracing estimators for refractive radiative transfer". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 39.6 (Nov. 26, 2020). issn: 0730-0301. doi: 10/gj4msh 11.

[PP16] Periyasamy, V. and Pramanik, M. "Importance sampling-based Monte Carlo simulation of time-domain optical coherence tomography with embedded objects". *Applied Optics* 55.11 (2016). doi: 10/gfz5m3 2.

[PVG19] Pediredla, A., Veeraraghavan, A., and Gkioulekas, I. "Ellipsoidal path connections for time-gated rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 38.4 (July 2019). issn: 0730-0301. doi: 10/gf5jbm 1, 7.

[VG95a] Veach, E. and Guibas, L. J. "Bidirectional estimators for light transport". *Photorealistic Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*. Springer-Verlag, 1995. isbn: 978-3-642-87825-1. doi: 10/gfznbh 1.

[VG95b] Veach, E. and Guibas, L. J. "Optimally combining sampling techniques for Monte Carlo rendering". *Annual Conference Series (Proceedings of SIGGRAPH)*. Vol. 29. ACM Press, Aug. 1995. isbn: 978-0-89791-701-8. doi: 10/d7b6n4 1, 2.

[VWJ*13] Velten, A., Wu, D., Jarabo, A., Masia, B., Barsi, C., Joshi, C., Lawson, E., Bawendi, M., Gutierrez, D., and Raskar, R. "Femto-photography: capturing and visualizing the propagation of light". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 32.4 (July 2013). issn: 0730-0301. doi: 10/gfxpkt 4.

[WCRZ21] Wu, L., Cai, G., Ramamoorthi, R., and Zhao, S. "Differentiable time-gated rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 40.6 (Dec. 2021). doi: 10/h2c9 2.

[YKC*21] Yi, S., Kim, D., Choi, K., Jarabo, A., Gutierrez, D., and Kim, M. H. "Differentiable transient rendering". *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 40.6 (Dec. 2021). doi: 10/h2db 2.