

# A Comprehensive Theory of Volumetric Radiance Estimation using Photon Points and Beams

WOJCIECH JAROSZ

Disney Research Zürich and UC San Diego

DEREK NOWROUZEZHRAI

Disney Research Zürich and University of Toronto

IMAN SADEGHI and HENRIK WANN JENSEN

UC San Diego

We present two contributions to the area of volumetric rendering. We develop a novel, comprehensive theory of volumetric radiance estimation that leads to several new insights and includes all previously published estimates as special cases. This theory allows for estimating in-scattered radiance at a point, or accumulated radiance along a camera ray, with the standard photon particle representation used in previous work. Furthermore, we generalize these operations to include a more compact, and more expressive intermediate representation of lighting in participating media, which we call “photon beams.” The combination of these representations and their respective query operations results in a collection of nine distinct volumetric radiance estimates.

Our second contribution is a more efficient rendering method for participating media based on photon beams. Even when shooting and storing less photons and using less computation time, our method significantly reduces both bias (blur) and variance in volumetric radiance estimation. This enables us to render sharp lighting details (e.g. volume caustics) using just tens of thousands of photon beams, instead of the millions to billions of photon points required with previous methods.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—*Color, shading, shadowing, and texture; Raytracing*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—*Raytracing*; I.6.8 [Simulation and Modeling]: Types of Simulation—*Monte Carlo*; G.3.8 [Mathematics of Computing]: Probability and Statistics—*Probabilistic algorithms (including Monte Carlo)*

General Terms: Theory, Algorithms, Performance

Additional Key Words and Phrases: global illumination, ray marching, rendering, density estimation, photon map, particle tracing, participating media

## 1. INTRODUCTION

Participating media is responsible for some of the most visually compelling effects we see in the world. The appearance of fire, water, smoke, clouds, rainbows, crepuscular “god” rays, and all

---

I. Sadeghi was funded in part by NSF grant CPA 0701992.

W. Jarosz and D. Nowrouzezhrai, Disney Research Zürich; email:{wjarosz,derek}@disneyresearch.com;

I. Sadeghi, H. W. Jensen, Department of Computer Science, University of California San Diego; email: {isadeghi,henrik}@cs.ucsd.edu.

This is the author’s personal copy of the article. The definitive version can be found at the ACM Digital Library.

© 2011 ACM 0730-0301/2011/11-ART5 \$10.00

DOI 10.1145/1899404.1899409

http://doi.acm.org/10.1145/1899404.1899409

organic materials is due to the way these media “participate” in light interactions by emitting, absorbing, or scattering photons. These phenomena are common in the real world but, unfortunately, are incredibly costly to simulate accurately. Because of this, computer graphics has had a long-standing interest in developing more efficient, accurate, and general participating media rendering techniques. We refer the reader to the recent survey by Cerezo et al. [2005] for a comprehensive overview.

The most general techniques often use a form of stochastic sampling and Monte Carlo integration. This includes unbiased techniques such as (bidirectional) path tracing [Lafortune and Willem 1993; Veach and Guibas 1994; Lafortune and Willem 1996] or Metropolis light transport [Pauly et al. 2000]; however, the most successful approaches typically rely on biased Monte Carlo combined with photon tracing [Keller 1997; Jensen and Christensen 1998; Walter et al. 2006; Jarosz et al. 2008]. Like bidirectional path tracing, photon tracing methods generate both camera and light paths but, instead of coupling these two types of paths directly, they trace and store a collection of paths from the lights first, and then determine a way to couple these light paths with the camera paths generated during rendering. Volumetric photon mapping [Jensen and Christensen 1998; Jarosz et al. 2008] performs density estimation on the vertices of these paths (the “photons”) to estimate volumetric radiance. This process is energy preserving, but blurs the results, introducing bias. However, this bias reduces noise and allows for efficient simulation of a wider range of light transport paths, such as caustics.

## 1.1 Motivation

One of the primary motivations for our work is that current photon tracing methods for participating media are limited by the data representation used to store light paths. Current methods use a photon particle representation that only retains information about the scattering event locations, discarding all other potentially important information accumulated during photon tracing. We observe that by retaining more information about the light paths during photon tracing, we can obtain vastly improved rendering results, as illustrated in Figure 1. In this example, retaining only light path *vertices* (photon points) results in a sparse sampling of the light field which, for density estimation techniques, either requires a large search radius with high bias or results in no photons being found within a fixed radius (highlighted in blue). In contrast, if we store full light *paths* (photon beams) and an approach for computing density estimation using these paths existed, the density of data would be implicitly higher. These benefits motivate the main contributions of our work.

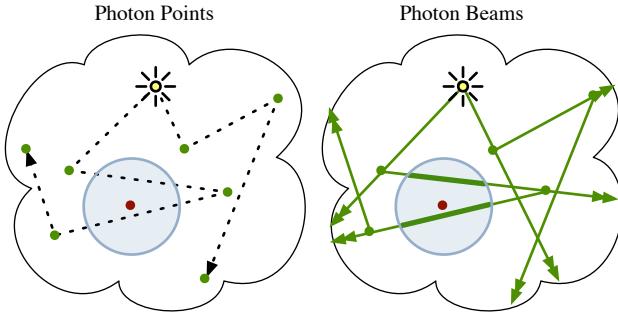


Fig. 1: Volumetric photon mapping (left) stores scattering events at points (green) and performs density estimation. With photon beams (right), the full trajectory (lines) of each photon is stored and density estimation is performed on line segments. Photon beams increase the quality of radiance estimation since the space is filled more densely (e.g. the blue search region does not overlap any photon points, but does overlap two photon beams).

## 1.2 Contributions

In order to develop a complete algorithm utilizing photon beams, we introduce a novel density estimation framework based around line segments. This allows us to extend standard volumetric photon mapping with a new data representation. We in fact go a step further and develop a generalized theory of radiance estimation in participating media which allows the use of points or beams for either the *data* or *query* representation, or *both*. Our main contribution is a theory that subsumes existing radiance estimates [Jensen and Christensen 1998; Jarosz et al. 2008; Schjøth 2009] and also expands the collection to nine estimates by including the new photon beam data representation and query operation.

To validate our new theory we develop a number of prototype volumetric rendering techniques utilizing these novel radiance estimates. We demonstrate their effectiveness on a number of scenes, and also discuss computational considerations, such as efficient data structures, to make photon beams practical.

Lastly, our unified theory provides new insights into how seemingly distinct volume rendering methods are in fact extremely similar. In addition to volumetric photon mapping, we show how virtual point-light methods [Keller 1997; Walter et al. 2006], light beam tracing methods [Nishita et al. 1987; Watt 1990; Nishita and Nakamae 1994], and deep shadow maps [Lokovic and Veach 2000] can all be seen as special cases or slight modifications of our theory.

## 2. RELATED WORK

Our work is related to a number of previous techniques for coupling light paths to camera paths in participating media.

### 2.1 Photon Mapping

We use volumetric photon mapping as the foundation for deriving a novel density estimation framework using points and beams. The original algorithm [Jensen and Christensen 1998] introduced a volumetric radiance estimate to approximate the in-scattered radiance at any *discrete point* in the medium. Radiance towards the eye is accumulated using ray marching, aggregating photon map queries along camera rays. Jarosz et al. [2008] target the inefficiencies of ray marching, formulating a new volumetric “beam radiance estimate” to consider all photons around the length of a camera ray (the “beam”) in a single query. BouDET et al. [2005] and Schjøth [2009] derived a photon splatting procedure which is mathematically equiv-

alent for camera rays. These estimates change the *query* definition from a point to a beam. In our work, we develop the tools necessary to change the *data* representation from a photon point to a photon beam. Moreover, we show how to couple any combination of point- or beam-query with a point- or beam-photon representation (e.g. a beam-query using photon-beams).

### 2.2 Ray Mapping

Previous researchers have proposed the use of beams or rays to solve surface illumination problems. Lastra et al. [2002] stored photon trajectories with each photon and used them to locate photons that *would have* intersected the tangent plane of the query point in order to reduce boundary bias in the surface radiance estimate. Havran et al. [2005] developed a specialized data structure called the “ray map” and formulated a number of other metrics to couple a query location on a surface with photon paths. Herzog et al. [2007] reformulated ray maps by splatting photon energy to all surface measurement points along a photon’s trajectory. Zinke and Weber [2006], on the other hand, discretely sampled the ray map back into photon points to improve query performance. None of these previous approaches, however, considered participating media.

Our concept of photon beams is very similar in spirit to that of the ray map. However, we apply this concept to simulating lighting in participating media, which is more challenging since radiance does not remain constant along lines through a medium. Moreover, the benefit of using beams as the data representation is much greater for participating media than for surfaces since, in a volume, beams not only reduce boundary bias but, as we will show, significantly reduce variance by implicitly increasing photon density.

### 2.3 Beam Tracing

Our use of photon beams is also related to the concept of beam tracing [Heckbert and Hanrahan 1984], where in place of infinitesimal rays, polygonal geometry is extruded to form thick “beams” which are reflected and refracted in the scene. The concept of beam tracing was later applied in reverse to light paths for e.g. visualizing caustics at the bottom of a swimming pool [Watt 1990]; however, neither of these techniques considered participating media. Nishita et al. [1987] used illumination volumes formed by spotlights to simulate single scattering and subsequently extended this algorithm to visualize underwater shafts of light [Nishita and Nakamae 1994]. Unfortunately, the resulting light beams have complicated boundaries and intensity profiles, which are approximated by sampling and interpolation. These volumetric light beam tracing methods can run interactively on the GPU in simple scenes [Iwasaki et al. 2001; Ernst et al. 2005]. Krüger et al. [2006] suggest a related GPU technique where image-space photons create underwater lines of light which are then blurred in image-space; however, they do not strive for physical correctness and cannot handle absorption.

The use of light beams to represent shafts of light within participating media is very similar to our concept of photon beams. In fact, one of the new estimates we introduce can be seen as a generalization of light beam tracing. Photon beams have a number of additional practical benefits. Firstly, light beams are tied to scene geometry and hence not well suited for scenes with high geometric complexity. Photon beams, on the other hand, are independent of the geometry, and can handle scenes with highly-tessellated or even procedural geometry. Secondly, it is difficult to handle higher-order scattering effects with light beams (e.g. multiple specular refractions) since subsequent bounces require non-trivial geometric light beam clipping. Photon beams are based on standard ray tracing and naturally handle multiple specular interfaces (for volumetric caus-

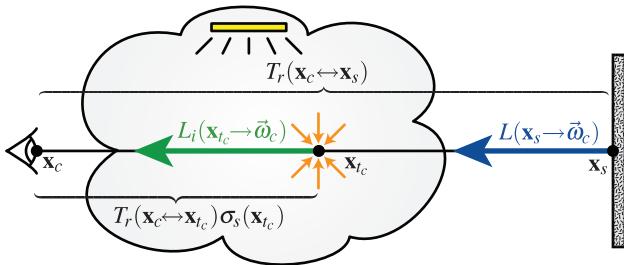


Fig. 2: Radiance reaching the eye  $L(\mathbf{x}_c \leftarrow \vec{\omega}_c)$  is the sum of surface radiance  $L(\mathbf{x}_s \leftarrow \vec{\omega}_c)$  and accumulated in-scattered radiance  $L_i(\mathbf{x}_{t_c} \leftarrow \vec{\omega}_c)$  along a ray.

tics) and even multiple scattering effects, which are not considered at all by light beam methods. Lastly, light beam tracing, unlike the approaches developed in this article, cannot easily handle area light sources.

## 2.4 Exact/Direct Techniques

Several techniques solve for the exact contribution of specific light paths, without an intermediate representation. Mitchell and Hanrahan [1992] solve for the exact reflection points off of curved surfaces and the corresponding lighting contribution, and discuss, but do not demonstrate, performing the same computation for refraction. This technique could be used to directly simulate difficult light paths within participating media, such as reflective caustics, but only with a single bounce and geometry limited to implicit surfaces.

Recently, Walter et al. [2009] developed a similar technique for computing refraction paths within triangle meshes. By using triangles, the technique is similar to light beam tracing, and can likewise produce light shafts within complex refractive volumetric boundaries. However, the solution is *exact* so it does not suffer from blurring or approximation artifacts present in light beam tracing or photon mapping methods. In contrast, volumetric photon mapping could require extremely high photon counts to reconstruct similarly sharp lighting features. Our photon beam estimates implicitly increase the data density compared to photon particles, providing a similar benefit. The accuracy of Walter’s technique, however, comes at a price. Firstly, it requires an expensive numerical solver for robustness, and performance is once again tied to the geometric complexity of the boundary mesh, making it impractical for complex scenes. Also, only a restricted class of light paths with a single refractive boundary are considered, whereas our methods can be used to simulate any number of bounces. Finally, Walter’s technique solves for light contribution only at individual points in the volume, so numerical integration using ray marching is still needed to compute the contribution along a camera ray. This integration can be expensive and, if not enough samples are taken, clamping is used to avoid noise which introduces unbounded bias due to loss of energy. We develop a theory to directly couple photon beams with entire camera rays, thereby eliminating the need to perform ray marching.

## 3. BACKGROUND

We will derive a novel theory of density estimation in participating media combining points and beams. In this section we describe the technical details of light transport in participating media and establish a consistent notation used throughout our exposition. Since we base our derivations on density estimation used in volumetric photon mapping, we also review the details of this algorithm.

### 3.1 Light Transport in Participating Media

In a vacuum, photons travel unobstructed until they interact with a surface. In participating media, photons interact with the surrounding medium. At any point, a photon traveling through a medium may be scattered or absorbed, altering its path, and reducing the contribution in the original direction. This process is described by the *radiative transfer equation* (RTE) [Chandrasekar 1960].

In its integral form, the RTE recursively defines radiance reaching the eye  $\mathbf{x}_c$  from direction  $\vec{\omega}_c$  as a sum of reflected radiance from the nearest visible surface and accumulated in-scattered radiance from the medium between the surface and the camera (see Figure 2):

$$\begin{aligned} L(\mathbf{x}_c \leftarrow \vec{\omega}_c) = & T_r(\mathbf{x}_c \leftarrow \mathbf{x}_s) L(\mathbf{x}_s \rightarrow \vec{\omega}_c) \\ & + \int_0^s T_r(\mathbf{x}_c \leftarrow \mathbf{x}_{t_c}) \sigma_s(\mathbf{x}_{t_c}) L_i(\mathbf{x}_{t_c} \rightarrow \vec{\omega}_c) dt_c, \end{aligned} \quad (1)$$

where  $T_r$  is the beam transmittance,  $s$  is the distance through the medium to the nearest surface at  $\mathbf{x}_s = \mathbf{x}_c - s\vec{\omega}_c$ , and  $\mathbf{x}_{t_c} = \mathbf{x}_c - t_c\vec{\omega}_c$  with  $t_c \in (0, s)$ . We summarize our notation in Table I.

In-scattered radiance,  $L_i$ , recursively depends on radiance arriving at  $\mathbf{x}_{t_c}$  from all directions  $\vec{\omega}_{t_c}$  over the sphere of directions  $\Omega_{4\pi}$ :

$$L_i(\mathbf{x}_{t_c} \rightarrow \vec{\omega}_c) = \int_{\Omega_{4\pi}} f(\theta_{t_c}) L(\mathbf{x}_{t_c} \leftarrow \vec{\omega}_{t_c}) d\vec{\omega}_{t_c}, \quad (2)$$

where  $f$  is the normalized phase function, and  $\theta_{t_c}$  is the angle between the incident and outgoing directions at  $\mathbf{x}_{t_c}$ :  $\cos \theta_{t_c} = \vec{\omega}_c \cdot \vec{\omega}_{t_c}$ . The surface radiance,  $L(\mathbf{x}_s \rightarrow \vec{\omega}_c)$ , governed by the *rendering equation* [Kajiya 1986], serves as the boundary condition for the RTE.

In heterogeneous media, the scattering properties may vary throughout the medium. In this case, we denote the scattering and absorption coefficients of the medium as  $\sigma_s(\mathbf{x})$  and  $\sigma_a(\mathbf{x})$ , and the extinction coefficient is  $\sigma_t(\mathbf{x}) = \sigma_s(\mathbf{x}) + \sigma_a(\mathbf{x})$ .

The beam transmittance,  $T_r$ , gives the fraction of radiance that can be transported between two points along a ray, and is defined as:

$$T_r(\mathbf{x} \leftarrow \mathbf{x}') = e^{-\int_0^s \sigma_t(\mathbf{x}+t\vec{\omega}) dt}. \quad (3)$$

In homogeneous media,  $\sigma_t$ ,  $\sigma_s$ , and  $\sigma_a$  do not depend on position, and a number of mathematical simplifications can be made. Specifically, the integral in Equation 3 can be replaced by a simple product:

$$T_r(\mathbf{x} \leftarrow \mathbf{x}') = e^{-\sigma_t \|\mathbf{x}' - \mathbf{x}\|}, \quad (4)$$

Table I.: Definitions of quantities used throughout this article.

Symbol	Description
$\mathbf{x}, \vec{\omega}$	Position, direction
$\Omega_{4\pi}$	Sphere of directions
$t$	Distance along a ray or beam
$\blacksquare_c$	Quantity associated with a camera ray, (e.g. $\mathbf{x}_c$ , $\vec{\omega}_c$ , $t_c$ )
$\blacksquare_p$	Quantity associated with photon particle $p$ , (e.g. $\mathbf{x}_p$ , $\vec{\omega}_p$ , $\Phi_p$ )
$\blacksquare_b$	Quantity associated with photon beam $b$ , (e.g., $\mathbf{x}_b$ , $\vec{\omega}_b$ , $t_b$ )
$\sigma_s, \sigma_a, \sigma_t$	Scattering, absorption, and extinction coefficients
$\theta_b$	Angle between photon beam and camera ray
$f(\theta)$	Normalized phase function
$R, R_b$	Abstract query/blurring region, aligned with photon beam $b$
$T_r$	Beam transmittance: $e^{-\sigma_t t}$
$\Phi$	Flux (power) or a photon particle or beam
$L(\mathbf{x} \leftarrow \vec{\omega})$	Incident radiance arriving at $\mathbf{x}$ from direction $\vec{\omega}$
$L_i(\mathbf{x} \rightarrow \vec{\omega})$	Excitant in-scattered radiance leaving $\mathbf{x}$ in direction $\vec{\omega}$
$L_b(\mathbf{x} \leftarrow \vec{\omega}, s)$	“Beam radiance”: incident integrated in-scattered radiance arriving at $\mathbf{x}$ from media from direction $\vec{\omega}$ .
$\xi$	A canonical random number between 0 and 1

which allows us to simplify the RTE in Equation 1 to:

$$L(\mathbf{x}_c \leftarrow \vec{\omega}_c) = e^{-\sigma_t s} L(\mathbf{x}_s \rightarrow \vec{\omega}_c) + \sigma_s \int_0^s e^{-\sigma_t t_c} L_i(\mathbf{x}_{t_c} \rightarrow \vec{\omega}_c) dt_c. \quad (5)$$

### 3.2 Volumetric Photon Mapping

Jensen and Christensen [1998] solve the RTE using a combination of photon tracing, ray marching, and density estimation.

In a preprocess, packets of energy, or “photons,” are shot from light sources, scattered at surfaces and within the medium, and their interactions are stored in a global data structure. This photon tracing stage is typically implemented using a Markov random-walk process, though a number of other sampling strategies are possible.

During rendering, ray marching is used to numerically integrate radiance seen directly by the observer. For a homogeneous medium, this involves approximating Equation 5 as:

$$L(\mathbf{x}_c \leftarrow \vec{\omega}_c) \approx e^{-\sigma_t s} L(\mathbf{x}_s \rightarrow \vec{\omega}_c) + \sigma_s \sum_{t_c=0}^{S-1} e^{-\sigma_t t_c} L_i(\mathbf{x}_{t_c} \rightarrow \vec{\omega}_c) \Delta t_c, \quad (6)$$

where  $\Delta t_c$  is the length of segments along the ray and  $\mathbf{x}_0, \dots, \mathbf{x}_{s-1}$  are the segment sample points ( $\mathbf{x}_0$  and  $\mathbf{x}_{s-1}$  are the first and last points within the medium, and  $\mathbf{x}_s$  is a surface point past the medium).

The most expensive part to compute in Equation 6 is the in-scattered radiance  $L_i$ , because it involves accounting for all light arriving at each point  $\mathbf{x}_t$  along the ray from any other point in the scene. Instead of computing these values independently for each location, photon mapping gains efficiency by reusing the computation performed during the photon tracing stage. The in-scattered radiance is approximated using density estimation by gathering photons within a small spherical neighborhood around each sample location.

### 3.3 Notation

This article deals with a wide variety of quantities, expressing all of which with absolute precision and generality would make equations unmanageably verbose. For conciseness, we will use the homogeneous RTE for most of the remainder of this article. When not immediately obvious, we discuss algorithmic changes needed to handle heterogeneous media. However, to make the meaning of terms more obvious in context, we will typically denote quantities relating to *camera rays* with a subscript  $c$  (e.g.,  $\mathbf{x}_c, \vec{\omega}_c, t_c$ ), relating to photon particles with a subscript  $p$  (e.g.,  $\mathbf{x}_p, \vec{\omega}_p, \Phi_p$ ), and relating to photon *beams* with a subscript of  $b$  (e.g.,  $\mathbf{x}_b, \vec{\omega}_b, t_b$ ). We use superscripts sparingly to denote other dependencies. We also make a notational distinction between incident/incoming and excitant/outgoing radiance using arrow notation (see Table I). Our illustrations use green for quantities relating to the data (photon points and photon beams), red for quantities relating to the query (query point or camera ray), and blue to visualize query regions or blurring kernel.

### 3.4 Overview

In Sections 4–7, we explore volumetric radiance estimation with photon mapping in more detail. In particular, we derive a comprehensive theory of volumetric radiance estimation, which leads to significant new insights and encompasses all previously published volumetric radiance estimates as special cases.

We first examine radiance estimation using photon points in Section 4 and then show how to generalize this concept to volumetric photon beams. Section 5 overviews the photon beams concept and sets the mathematical foundation for radiance estimation using this representation. In Section 6 we derive two ways to estimate in-scattered radiance in a volume using photon beams, and in Section 7

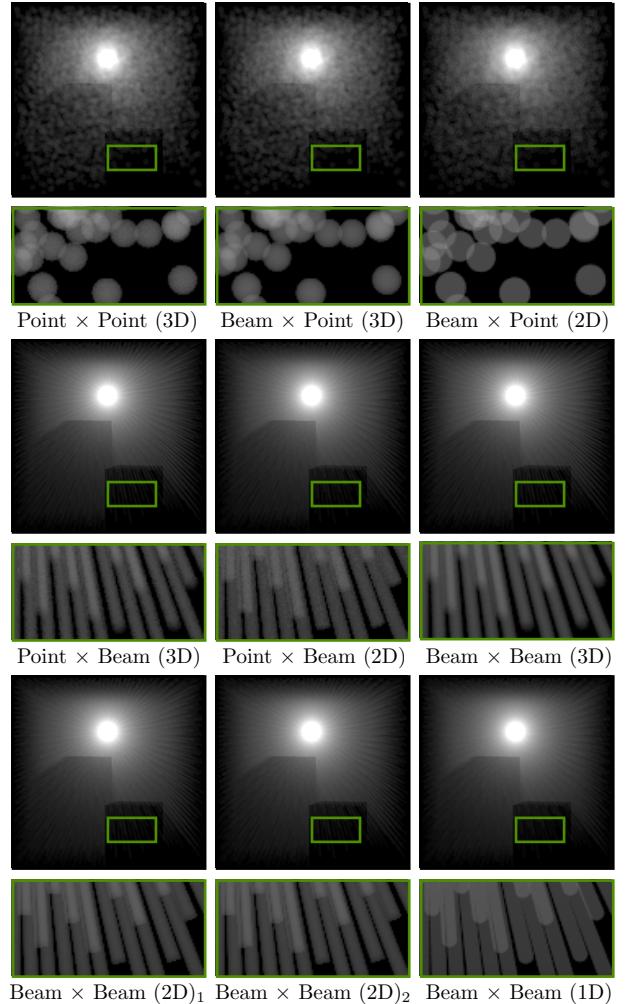


Fig. 3: Single-scattering in a Cornell box rendered with each of the nine estimates (see Table II). We only visualize media scattering and intentionally use a small number of photons and a constant kernel to highlight each method’s artifacts and blurring behaviors. 100k photon points (top row) and 5k photon beams (bottom two rows) are used to represent the illumination.

we derive four ways to estimate the accumulated in-scattered radiance along a camera ray directly using photon beams. In total, these generalizations result in nine distinct radiance estimates. We categorize these estimates based on the type of query (point or beam), the photon data representation (point or beam), and the dimensionality of the blur coupling these two terms (3D, 2D, or 1D). We demonstrate each of these estimators in Figure 3 and aggregate all the estimator formulae derived throughout the manuscript in Table II.

In Section 8 we provide practical details needed to implement several of the novel radiance estimates, and in Section 9 show and discuss rendered results using these methods. In Section 10 we discuss how our method fits into the larger scope of volumetric rendering, and how our theory sheds light on connections between other, seemingly disparate volume rendering approaches. Limitations and areas of future work are discussed in Section 11.

## 4. RADIANCE ESTIMATION USING PHOTON POINTS

We detail all the possible ways to estimate radiance with photon points. Standard volumetric radiance estimation is reviewed in Section 4.1, we then present novel derivations for two different radiance estimates computed along the length of the eye ray.

### 4.1 Point Query × Point Data, 3D Blur

The original volumetric radiance estimate computes the in-scattered radiance at a query **point** by searching for photon **points** stored within the medium. In this estimate, the query is a point,  $\mathbf{x}$ , and the data are points,  $\mathbf{x}_p$ . The resulting estimate blurs the values stored at the points  $\mathbf{x}_p$  in three dimensions [Jensen and Christensen 1998]:

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) = \int_{\Omega_{4\pi}} f(\theta') L(\mathbf{x} \leftarrow \vec{\omega}') d\vec{\omega}' \approx \frac{1}{\sigma_s \mu_R(r^3)} \sum_{p \in R} f(\theta_p) \Phi_p, \quad (7)$$

where the summation loops over all photons  $p$  found within a three-dimensional query region  $R$ . This sum is divided by the measure of the query region. For the typical spherical query the measure is simply the volume of a sphere  $\mu_R(r^3) = \frac{4}{3}\pi r^3$ , but other three-dimensional queries are possible, as well as weightings by non-constant kernels. We illustrate this radiance estimate in Figure 4(a).

Our notation may seem to suggest a restriction to spatially constant (i.e. non-adaptive) query regions, by not explicitly specifying a dependence on  $\mathbf{x}$  or  $\mathbf{x}_p$ . This is only for simplicity and brevity of notation and is not a limitation of the methods described. We discuss the application of spatially adaptive blurring kernels in Section 8.1.

### 4.2 Beam Query × Point Data, 3D Blur

It is also possible to directly compute the accumulated in-scattered radiance along a camera ray, which we call “beam radiance”:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) = \sigma_s \int_0^s e^{-\sigma_t t_c} L_i(\mathbf{x}_c \leftarrow \vec{\omega}_c) dt_c. \quad (8)$$

The standard volumetric radiance estimate can be thought of as the convolution of the radiance at a point  $\mathbf{x}_c$  with a spherical 3D volumetric kernel. Substituting Equation 7 into Equation 8 yields:

$$\begin{aligned} L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) &\approx \sigma_s \int_0^s e^{-\sigma_t t_c} \frac{1}{\sigma_s \mu_R(r^3)} \sum_{p \in R(t_c)} f(\theta_p) \Phi_p dt_c, \\ &\approx \frac{1}{\mu_R(r^3)} \int_0^s e^{-\sigma_t t_c} \sum_{p \in R(t_c)} f(\theta_p) \Phi_p dt_c, \end{aligned} \quad (9)$$

where  $R(t_c)$  indicates that the 3D query region moves with  $t_c$  along the beam integral domain.

Unfortunately, Equation 9 is not immediately useful as it involves convolving the query region with the line integral. We can instead look at the density estimation problem from the dual perspective where each photon is the center of a volumetric 3D kernel. This interpretation allows us to swap the order of the integral and summation, yielding the following radiance estimate:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{1}{\mu_R(r^3)} \sum_{p \in R} f(\theta_p) \Phi_p \int_{t_{p,c}^-}^{t_{p,c}^+} e^{-\sigma_t t_c} dt_c, \quad (10)$$

where the summation loops over all photons whose volumetric (typically spherical) kernels overlap the camera ray,  $t_c$  is the distance from  $\mathbf{x}_c$  along the camera ray, and the integration bounds  $t_{p,c}^-$  and  $t_{p,c}^+$  are determined by the intersection of the camera ray with the kernel for photon  $p$ . We illustrate these quantities in Figure 4(b).

Table II.: The nine radiance estimates described in this article, categorized by the query type, the data representation, and the dimensionality of the blur. A rendering resulting from each of these estimates is shown in Figure 3.

Query × Data (Blur)	Radiance Estimate	Equation Number
Point × Point (3D)	$\frac{1}{\sigma_s \mu_R(r^3)} \sum_{p \in R} f(\theta_p) \Phi_p$	(7)
Beam × Point (3D)	$\frac{1}{\mu_R(r^3)} \sum_{p \in R} f(\theta_p) \Phi_p \int_{t_{p,c}^-}^{t_{p,c}^+} e^{-\sigma_t t_c} dt_c$	(10)
Beam × Point (2D)	$\frac{1}{\mu_R(r^2)} \sum_{p \in R} f(\theta_p) \Phi_p e^{-\sigma_t t_{p,c}}$	(12)
Point × Beam (3D)	$\frac{1}{\mu_R(r^3)} \sum_{b \in R} f(\theta_b) \Phi_b \int_{t_b^-}^{t_b^+} e^{-\sigma_t t_b} dt_b$	(21)
Point × Beam (2D)	$\frac{1}{\mu_R(r^2)} \sum_{b \in R_b} f(\theta_b) \Phi_b e^{-\sigma_t t_{b,c}}$	(25)
Beam × Beam (3D)	$\frac{\sigma_s}{\mu_R(r^3)} \sum_{b \in R} f(\theta_b) \Phi_b \int_{t_b^-}^{t_b^+} \int_{t_c^-}^{t_c^+} e^{-\sigma_t t_b} e^{-\sigma_t t_c} dt_b dt_c$	(27)
Beam × Beam (2D) <sub>1</sub>	$\frac{\sigma_s}{\mu_R(r^2)} \sum_{b \in R_b} f(\theta_b) \Phi_b \int_{t_c^-}^{t_c^+} e^{-\sigma_t t_c} e^{-\sigma_t t_b} dt_c$	(29)
Beam × Beam (2D) <sub>2</sub>	$\frac{\sigma_s}{\mu_R(r^2)} \sum_{b \in R} f(\theta_b) \Phi_b \int_{t_b^-}^{t_b^+} e^{-\sigma_t t_b} e^{-\sigma_t t_c} dt_b$	(33)
Beam × Beam (1D)	$\frac{\sigma_s}{\mu_R(r)} \sum_{b \in R_b} \frac{f(\theta_b) \Phi_b e^{-\sigma_t t_b^c} e^{-\sigma_t t_c^b}}{\sin \theta_b}$	(38)

In a homogeneous medium, the integral of the transmittance term can be computed analytically as:

$$\int_{t_{p,c}^-}^{t_{p,c}^+} e^{-\sigma_t t_c} dt_c = \frac{e^{-\sigma_t t_{p,c}^-} - e^{-\sigma_t t_{p,c}^+}}{\sigma_t}. \quad (11)$$

**4.2.1 Discussion.** An important distinction between the radiance estimates in Equations 10 and 7 is that they estimate different radiometric quantities:  $L_i(\mathbf{x} \rightarrow \vec{\omega})$  is an excitant quantity while  $L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s)$  is an incident quantity. To compute the incident radiance reaching the eye using the standard approach, numerical integration would need to be used by inserting Equation 7 into the ray marching process of Equation 6 (even for homogeneous media). In contrast, Equation 10 computes the camera ray integral directly and, in homogeneous media, does not require ray marching. In effect, multiple ray marching evaluations of Equation 7 can be replaced with a single evaluation of Equation 10. Furthermore, as the number of ray marching samples increases to infinity, these two approaches will provide identical results. Note that even though these estimates compute different quantities, they can use the same photon map.

### 4.3 Beam Query × Point Data, 2D Blur

Recently, Jarosz et al. [2008] introduced the so-called “beam radiance estimate,” where the contributions from all photons along a camera ray are queried at once, similarly to Equation 10. Expressed in our notation, this beam radiance estimate can be written as<sup>1</sup>:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{1}{\mu_R(r^2)} \sum_{p \in R} f(\theta_p) \Phi_p e^{-\sigma_t t_{p,c}}, \quad (12)$$

where  $t_{p,c} = (\mathbf{x}_c - \mathbf{x}_p) \cdot \vec{\omega}_c$  is the projected distance along the ray from the origin  $\mathbf{x}_c$  to photon  $p$ . The sum is over all photons within a region  $R$  and, to enforce integration bounds along the beam, only

<sup>1</sup>Jarosz et al. [2008] use a different definition of photon contribution and so their equations contain an additional  $\sigma_s$  factor. Here we use notation to remain consistent with other sources [Jensen and Christensen 1998].

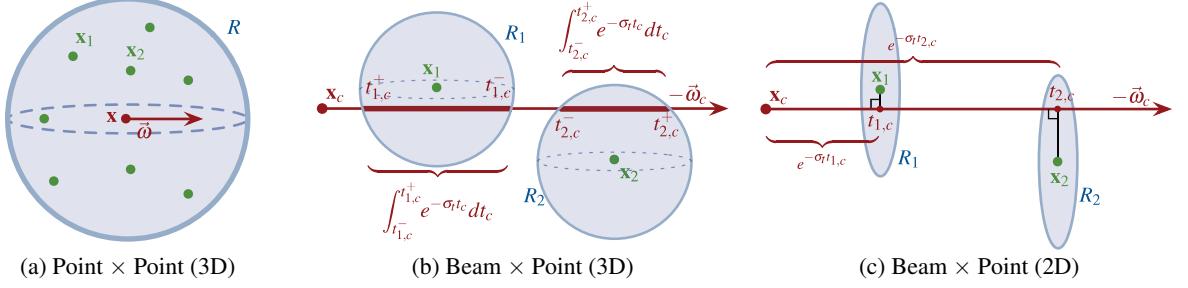


Fig. 4: Illustrations of the three possible radiance estimates using photon points. Radiance estimate (a) queries at a **point** and blurs in 3D, corresponding to the standard estimate introduced by Jensen and Christensen [1998]. Estimator (c) queries along a camera **beam**, only blurs in 2D, and corresponds to the beam radiance estimate introduced by Jarosz et al. [2008]. An estimator (b) which blurs in 3D along a camera **beam** is also possible.

considers photons with  $0 \leq t_{p,c} \leq s$ . This blurs each photon into a “photon disc” perpendicular to the query ray, as in Figure 4(c).

As with the previous estimate, this beam radiance estimate computes the 1D beam integral directly; however, it only blurs the contributions of each photon in 2D, perpendicular to the ray. Therefore we divide by the 2D measure of the region to compute the density. For a cylinder with a circular cross-section,  $\mu_R(r^2) = \pi r^2$ . This is an important distinction, which may initially seem unintuitive, and which is unfortunately hidden in the definition of the blurring kernel in the original expressions presented by Jarosz et al. [2008].

Jarosz et al. [2008] derived Equation 12 using a reformulation of photon mapping in terms of the measurement equation [Veach 1997]. For completeness, we will show that it is possible to derive the exact same estimate without this reformulation by starting from Equation 10. This derivation also allows us to examine the relationship between the 3D and 2D beam radiance estimates more precisely.

**4.3.1 Derivation.** Equation 10 blurs photons with an abstract 3D kernel. The shape of this kernel does not influence the algorithm’s correctness, only the behavior of bias and variance. In fact, this fact has been exploited previously by aligning kernels with structures in the lighting to minimize bias [Schjøth et al. 2006; Schjøth et al. 2007; Schjøth et al. 2008]. Consequently, we could choose a 3D region other than a sphere to perform our blur (e.g. a cylinder). This only requires changing  $\mu_R$  to express e.g. the volume of a cylinder,  $\mu_R(r^2 h) = \pi r^2 h$ .

We start by expressing Equation 10 using a cylindrical kernel:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{1}{\pi r^2 h} \sum_{p \in R} f(\theta_p) \Phi_p \int_{t_{p,c}^-}^{t_{p,c}^+} e^{-\sigma_i t_c} dt_c. \quad (13)$$

If we also align cylinders with the camera ray, the extent of the integration bounds always equals the cylinder height:  $t_{p,c}^+ - t_{p,c}^- = h$ . With this alignment, we express  $t_{p,c}^- = t_{p,c} - \frac{h}{2}$ , and  $t_{p,c}^+ = t_{p,c} + \frac{h}{2}$ .

Integrating over a distance  $h$  and dividing by  $h$  (from the measure of the cylinder), we effectively compute the average transmittance through the cylinder. In the limit, reducing the cylinder’s height to integrate smaller regions, this average becomes a simple evaluation:

$$\begin{aligned} L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) &\approx \lim_{h \rightarrow 0} \left( \frac{1}{\pi r^2 h} \sum_{p \in R} f(\theta_p) \Phi_p \int_{t_{p,c}^-}^{t_{p,c}^+} e^{-\sigma_i t_c} dt_c \right), \\ &= \frac{1}{\pi r^2} \sum_{p \in R} f(\theta_p) \Phi_p e^{-\sigma_i t_{p,c}}. \end{aligned} \quad (14)$$

Note that if we use a circular cross section with  $\mu_R(r^2) = \pi r^2$ , this is identical to the beam radiance estimate in Equation 12.

**4.3.2 Discussion.** We will discuss the relationship between the 2D and 3D beam radiance estimates. In Equation 10, transmittance

is integrated across the 3D kernel’s depth, whereas in Equation 12, transmittance is simply evaluated since the 2D kernel has no depth. However, in the 3D version we divide by an extra dimension.

Equation 14’s derivation solidifies the connection: combining the extra division and integral along the ray effectively averages the transmittance through the kernel. In the limit, we obtain a 2D blur, and this average becomes an evaluation of transmittance at the intersection of the kernel and camera ray. Strictly speaking, Equation 10 and Equation 12 produce very similar, but **not identical** results. The difference lies in the averaging of the transmittance term across depth in the 3D kernel. Since transmittance is non-linear, averaging is not equivalent to evaluation at the 3D kernel’s center.

It is also instructive to discuss the connection between these beam radiance estimates and the photon splatting approach developed by Bouget et al. [2005] and Schjøth [2009]. The splatting approaches consider a 3D region around each photon, and integrate each photon’s contribution onto the image plane as a splat. This integration, however, only considers the value of the kernel through the 3D region, and not the transmittance which is evaluated at the center of the kernel. The resulting radiance estimate divides by the volume of the region, but multiplies by the 1D integral of the kernel along the ray. These two operations combined produce an effective 2D kernel surrounding each photon. Hence, the splatting estimates are mathematically equivalent to the 2D beam radiance estimate in Equation 12.

In homogeneous media both these estimates can be evaluated analytically, so the benefit of one over the other is not immediately obvious. In heterogeneous media, however, using Equation 12 is more practical, since we only need to *evaluate* the transmittance (using ray marching) instead of *averaging* the transmittance through each photon. Equation 12’s beam radiance estimate is computationally much more efficient than the point-wise volumetric radiance estimate in Equation 7 with ray marching (see Jarosz et al. [2008]). In heterogeneous media the beam radiance estimate requires ray marching, but only to compute transmittance to each photon. This allows us to find all photons around the beam in one query, and take a weighted sum of their powers with Equation 12. Equation 7 is wasteful, requiring many queries per camera ray, which may result in the same photons being found multiple times.

## 5. PHOTON BEAMS OVERVIEW

We now generalize the concept of photon mapping to use photon beams, resulting in several novel radiance estimates.

The most commonly described photon shooting method is based on random-walk sampling where each emitted photon traces a Markov-chain of scattering events through the scene; however, other

approaches are possible as well<sup>2</sup>. Here we describe a different form of photon shooting, which resembles ray marching from the lights, and call this process “photon marching.” This approach can be much more effective in certain scenarios (e.g. light through a stained-glass window). Though this technique is not new (it was, for instance, used to generate volumetric water beams by Jensen and Christensen [1998]), it is not widely known and, to our knowledge, not described in the literature. More importantly, our definition of photon beams is based on the concept of photon marching, so we describe this process here in more detail. However, we only use this process to *derive* a mathematical definition of photon beams, and in practice photon beams are still created using a random-walk process.

## 5.1 Photon Marching

Photon marching is identical to ray marching with rays originating at the lights. At each step, instead of sampling the in-scattered light, a photon is deposited. The distinction between photon tracing using a Markov random-walk or a photon marching process is analogous to the difference between synthesizing an image using random-walk volumetric path tracing and ray marching from the camera.

Ray marching numerically solves 1D integrals using a Riemann sum. A step-size  $\Delta t$  is chosen and the ray is sampled at discrete locations. We assume a uniform step size for simplicity of the derivations, which implies the medium is assumed to have finite extent.

To employ ray marching in photon tracing, the propagation distance is no longer chosen once for a single photon (as in a random-walk), but instead the power of each photon passing through the medium is distributed among photons deposited at uniform intervals along the length of the ray through the medium. We call this discrete collection of photons along a single photon marching ray a “discrete photon beam.” Mathematically, this is expressed as:

$$\Phi_{pb} = \sigma_s e^{-\sigma_t \|\mathbf{x}_b - \mathbf{x}_t\|} \Phi_b \Delta t, \quad (15)$$

where  $\Phi_b$  is the power of the photon beam upon entering the medium (the emitted photon power if the light is within the medium), the exponential computes transmittance from the ray origin to the current photon  $\mathbf{x}_t$ , and  $\Phi_{pb}$  is the power of each photon. This scheme stores a photon at each  $\mathbf{x}_t = \mathbf{x}_b + t\vec{\omega}_b$ , with  $t = (\xi + [0, \dots, N-1])\Delta t$ . The total number of photons deposited along the discrete beam,  $N$ , depends on the step-size and extent of the beam through the medium. Each beam uses a different random offset  $\xi$  to avoid aliasing, but this is inconsequential for our derivations.

## 5.2 Multiple Scattering

Any photon entering the medium induces a discrete photon beam through the *entire* length of the medium. To simulate multiple scattering, photon marching can be combined with random-walk sampling.

This is accomplished by scattering a new discrete beam off of the current beam. For discrete beam  $b_0$ , a random propagation distance  $t_s$  is first chosen for the scattering event location. In homogeneous media we compute this analytically based on the mean-free path as  $t_s = -\frac{\log(1-\xi)}{\sigma_t}$  with  $pdf(t_s) = \sigma_t e^{-\sigma_t t_s}$ . For heterogeneous media, the inversion method [Pharr and Humphreys 2004] or delta-tracking [Coleman 1968; Raab et al. 2008] can be used. After the location is chosen, a new discrete beam is initiated with origin  $\mathbf{x}_{b_1} = \mathbf{x}_{b_0} + t_s \vec{\omega}_{b_0}$ , direction  $\vec{\omega}_{b_1}$  (sampled from  $f(\theta)$ ), and a starting power of  $\Phi_{b_1} = \frac{\sigma_s}{\sigma_t} \Phi_{b_0}$ . The scattering albedo,  $\frac{\sigma_s}{\sigma_t}$ , can be used for

<sup>2</sup>Any strategy is valid as long as the joint probability density of the photon powers, positions, and directions satisfies certain conditions [Veach 1997].

Russian roulette [Arvo and Kirk 1990], in which case the starting power of an accepted beam simplifies to  $\Phi_{b_1} = \Phi_{b_0}$  (in grey media). Even though the scattered beam is initiated at a distance  $t_s$  along the parent  $b_0$ , photon marching of  $b_0$  continues to the media’s boundary (see Figure 1). This simplifies our remaining derivations and provides the added benefit of increased photon density throughout the medium.

## 6. IN-SCATTERED RADIANCE ESTIMATION USING PHOTON BEAMS

In this section we generalize the theory of radiance estimation to compute in-scattered radiance at a point using photon beams.

### 6.1 Point Query $\times$ Beam Data, 3D Blur

We derive our first in-scattered radiance estimate at a point in the medium due to photon beams by combining the photon marching concept with the volumetric radiance estimate in Equation 7. Each discrete photon beam contains photons  $\Phi_{pb}$  deposited with photon marching. Re-writing Equation 7 in terms of these photons yields:

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \frac{1}{\sigma_s \mu_R(r^3)} \sum_{b \in R} \sum_{p_b \in R} f(\theta_{p_b}) \Phi_{p_b}, \quad (16)$$

where the outer sum iterates over all discrete beams  $b$  overlapping region  $R$ , and the inner sum finds all photons within the search region belonging to beam  $b$ . Note that this is exactly equivalent to Equation 7, we have just split up the single summation into two.

As the step size of a discrete photon beam goes to zero, we call the limit a “continuous photon beam.” Conceptually, a photon resides at each point along a continuous beam. Mathematically, we substitute the photon power expression (Equation 15) into Equation 16 and move the phase function and power out of the inner sum, since discrete photons on a beam share incident directions:

$$\begin{aligned} L_i(\mathbf{x} \rightarrow \vec{\omega}) &\approx \frac{1}{\sigma_s \mu_R(r^3)} \sum_{b \in R} \sum_{p_b \in R} f(\theta_{p_b}) \sigma_s \Phi_b e^{-\sigma_t t_{p,b}} \Delta t, \\ &= \frac{1}{\mu_R(r^3)} \sum_{b \in R} f(\theta_b) \Phi_b \sum_{p_b \in R} e^{-\sigma_t t_{p,b}} \Delta t, \end{aligned} \quad (17)$$

where  $\vec{\omega}_b$  and  $\Phi_b$  are the direction and power of discrete beam  $b$ ,  $\theta_b$  is the angle between the eye ray and the beam ( $\cos \theta_b = \vec{\omega}_c \cdot \vec{\omega}_b$ ), and  $t_{p,b}$  is the distance from the start of the beam  $\mathbf{x}_b$  to photon  $p_b$ . To arrive at continuous beams, we take the limit as  $\Delta t$  goes to zero:

$$\begin{aligned} L_i(\mathbf{x} \rightarrow \vec{\omega}) &\approx \frac{1}{\mu_R(r^3)} \lim_{\Delta t \rightarrow 0} \sum_{b \in R} f(\theta_b) \Phi_b \sum_{p_b \in R} e^{-\sigma_t t_{p,b}} \Delta t, \\ &= \frac{1}{\mu_R(r^3)} \sum_{b \in R} f(\theta_b) \Phi_b \lim_{\Delta t \rightarrow 0} \sum_{p_b \in R} e^{-\sigma_t t_{p,b}} \Delta t. \end{aligned} \quad (18)$$

In the limit, a Riemann sum computes an integral,

$$\lim_{\Delta x \rightarrow 0} \sum_{x \in R} f(x) \Delta x = \int_{x \in R} f(x) dx, \quad (19)$$

hence, as we increase the number of photons along a beam to infinity, the inner sum in Equation 18 becomes a continuous 1D integral along each beam:

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \frac{1}{\mu_R(r^3)} \sum_{b \in R} f(\theta_b) \Phi_b \int_{t_b \in R} e^{-\sigma_t t_b} dt_b, \quad (20)$$

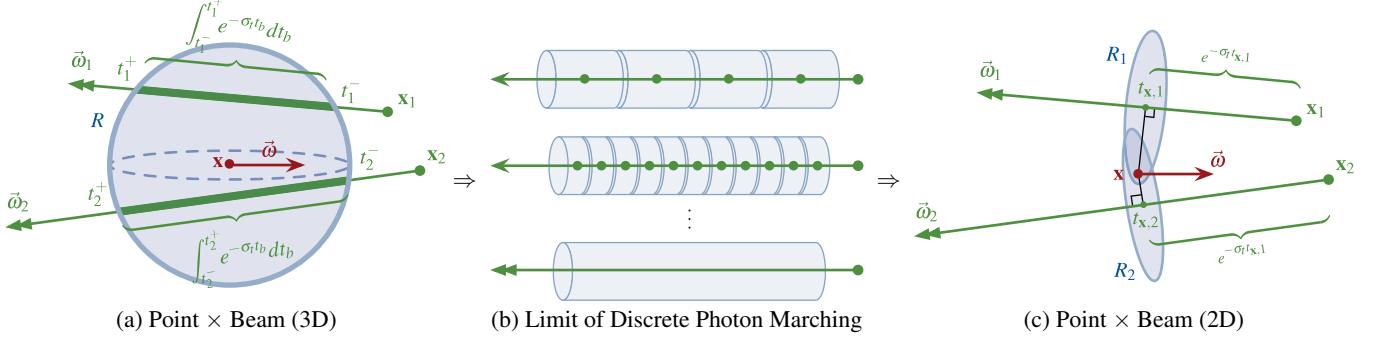


Fig. 5: In-scattered radiance estimation with photon beams. Radiance estimator (a) computes in-scattered radiance at a **point** using photon **beams** with a **3D blur**. Taking the limit of discrete photon marching (b) while shrinking the cylindrical kernel's height, we can obtain a similar estimator (c) which only blurs in **2D**.

where  $t_b$  is the parametric distance along beam  $b$ . The integral bounds  $t_b \in R$  represent the range along beam  $b$  that overlaps with region  $R$ . Defining overlap endpoints,  $t_b^-$  and  $t_b^+$ ,  $L_i$  is expressed:

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \frac{1}{\mu_R(r^3)} \sum_{b \in R} f(\theta_b) \Phi_b \int_{t_b^-}^{t_b^+} e^{-\sigma_t t_b} dt_b. \quad (21)$$

In homogeneous media, the integral of transmittance can be solved analytically (Equation 11). We illustrate this estimate in Figure 5(a).

**6.1.1 Discussion.** Equation 21 is a true volumetric estimate, averaging over a 3D region and dividing by the volume measure  $\mu_R(r^3)$ . This can be thought of as the convolution of the continuous photon beam function with a 3D blurring kernel. It is interesting to note that the point query of a 3D-blurred photon beam in Equation 21 and the beam query of 3D-blurred photon point in Equation 10 are nearly identical. They differ only in the direction transmittance is computed. The estimate in this section evaluates transmittance from the query point to the start of the beam and integrates it around the query point; in contrast, the estimate in Section 4.2 evaluates transmittance from the photon point towards the eye and integrates it around the photon point. In a sense, these estimates are radiometric “complements” of each other and, due to the bidirectionality of light transport, end up being nearly identical mathematically. This concept of complementary radiance estimates will appear several more times in our derivations. More precisely, it occurs anytime we swap query and data representations but maintain the blur dimensionality.

## 6.2 Point Query × Beam Data, 2D Blur

Photon mapping can be viewed as computing random samples of the energy distribution in the scene [Veach 1997; Jarosz et al. 2008]. The value at each photon point is exact, but a blur is required to utilize these samples for estimating radiance at arbitrary points in the scene. With the standard volumetric radiance estimate (Equation 7) we blur the photon energy in three dimensions to obtain a valid estimate at all points within the volume. Equation 21 also blurs in 3D. Continuous photon beams, however, express the exact value of the lighting at all points along the photon beam, not just at a single point. Hence, blurring along the length of the beam only serves to introduce bias. We can exploit this observation to obtain another radiance estimate for photon beams which only blurs in 2D, perpendicular to the beam.

We utilize the dual interpretation of density estimation, where a blurring kernel is placed at each data “point”: in this case, the standard volumetric radiance estimate places a spherical kernel at each photon point and the in-scattered radiance is accumulated from the energy of all photon-spheres that overlap the query point.

As mentioned earlier, the blurring kernel shape only influences the bias and variance behavior, not the correctness of the algorithm. We exploit this fact, and again employ a cylindrical blurring region (this time aligned with the photon beam) to derive the next radiance estimate. The cylindrical blur can be expressed using the discrete photon beams with only a slight modification to Equation 17:

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \frac{1}{\pi r^2 h} \sum_{b \in R_b} f(\theta_b) \Phi_b \sum_{p_b \in R_b} e^{-\sigma_t t_{p,b}} \Delta t, \quad (22)$$

where the first sum loops over beams that pass near the query location  $\mathbf{x}$ , and the second finds all photon-marching photons for that discrete beam that fall within the oriented cylindrical region  $R_b$ . The orientation of the cylindrical blur depends on the beam, denoted  $R_b$ .

To obtain continuous photon beams, we could repeat a similar procedure as in the previous section by taking the limit as  $\Delta t$  goes to 0 and arrive at an expression analogous to Equation 21. However, we can reduce bias by simultaneously reducing the blur along the length of the beam to zero, as illustrated in Figure 5(b). To do so, we set the cylinder height to the photon-marching step size,  $h = \Delta t$ :

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \lim_{\Delta t \rightarrow 0} \frac{1}{\pi r^2 \Delta t} \sum_{b \in R_b} f(\theta_b) \Phi_b \sum_{p_b \in R_b} e^{-\sigma_t t_{p,b}} \Delta t. \quad (23)$$

To simplify this expression, we note that none of the kernels for a single beam overlap because the photon spacing and size of the kernels are equal (see Figure 5(b)). This means that for any query point  $\mathbf{x}$ , at most one photon (the one closest to  $\mathbf{x}$ ) from each discrete beam contributes to the estimate. This allows us to eliminate the inner summation. After solving for the limit behavior, we have:

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \frac{1}{\pi r^2} \sum_{b \in R_b} f(\theta_b) \Phi_b e^{-\sigma_t t_{x,b}}, \quad (24)$$

where  $t_{x,b} = (\mathbf{x} - \mathbf{x}_b) \cdot \vec{\omega}_b$  is the scalar projection of  $\mathbf{x}$  onto beam  $b$ .

Even though we start with a volumetric blur, the limit process collapses the cylinder to a disc resulting in a 2D blur normalized by the cylindrical cross-section area,  $\pi r^2$ , as illustrated in Figure 5(c).

We used a circular cross-section for convenience of derivation, but an estimate using an arbitrary cross-section can be expressed as:

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \frac{1}{\mu_R(r^2)} \sum_{b \in R_b} f(\theta_b) \Phi_b e^{-\sigma_t t_{x,b}}. \quad (25)$$

**6.2.1 Discussion.** It is informative to compare this radiance estimate to the other ones presented so far. When compared to Equation 21, we observe that the main differences are that here we divide by the 2D area measure instead of the 3D volume measure, and we also only need to *evaluate* the transmittance term along the beam, instead of *integrating* it along an overlapping region.

Though different mathematically, these expressions estimate the same radiometric quantity (the two differences effectively cancel each other out). Roughly speaking, if we integrate the transmittance and divide by an extra  $r$  term, this is approximately equal to simply evaluating the transmittance. The extra division can be thought of as computing the “average” transmittance through the region. In the limit, this is equivalent to evaluating the transmittance and only dividing by the area measure. Note that if the function being integrated (transmittance) were linear, these two formulations would in fact be mathematically identical. The 2D estimate presented here has a number of advantages. Firstly, it is simpler to evaluate since it does not involve integrating along the length of beams. Secondly, since no blurring is performed along the beam, this radiance estimate introduces less bias for the same radius. It can be thought of as the limit case of an anisotropic density estimation.

Comparing Equation 25 to the beam radiance estimate (Equation 12), we see that even though these compute very different quantities (in-scattered radiance at a point vs. accumulated in-scattered radiance along an eye ray), they have a very similar structure and are in fact complementary. Both estimates divide by the cross-sectional area of either the camera beam or the photon beam, and both weight the photon contributions by a transmittance term (though computed in different directions). This elegant similarity is not coincidental, arising again due to the bidirectionality of light transport.

Equation 25 is also similar to the surface radiance estimates developed by Herzog et al. [2007] and can be seen as a generalization of their photon ray splatting estimate to participating media.

## 7. BEAM RADIANCE ESTIMATION USING PHOTON BEAMS

We now consider direct computation of the beam radiance integral using photon beams; hence, all remaining estimates apply beam queries to photon beams.

### 7.1 Beam Query × Beam Data, 3D Blur

To directly estimate the beam radiance integral with 3D blurred photon beams, we substitute Equation 21 for the in-scattered radiance in the beam radiance integral along the camera ray (Equation 8):

$$\begin{aligned} L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) &= \sigma_s \int_0^s e^{-\sigma_i t_c} L_i(\mathbf{x}_{t_c} \rightarrow \vec{\omega}_c) dt_c, \\ &\approx \frac{\sigma_s}{\mu_R(r^3)} \int_0^s e^{-\sigma_i t_c} \sum_{b \in R} f(\theta_b) \Phi_b \int_{t_b^-}^{t_b^+(t_c)} e^{-\sigma_i t_b} dt_b dt_c. \end{aligned} \quad (26)$$

Note that the bounds of the inner integral depend on the location along the camera ray  $t_c$ .

After moving the integral inside the summation, and defining per-beam integration bounds, we arrive at:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{\sigma_s}{\mu_R(r^3)} \sum_{b \in R} f(\theta_b) \Phi_b \int_{t_b^-}^{t_b^+} \int_{t_c^-}^{t_c^+(t_c)} e^{-\sigma_i(t_c + t_b)} dt_b dt_c, \quad (27)$$

where  $t_c^-$  and  $t_c^+$  are the overlap of the camera ray with the extruded 3D kernel along the photon beam, and at each point in this range we integrate along the length of the photon beam from  $t_b^-(t_c)$  to  $t_b^+(t_c)$ .

**7.1.1 Discussion.** Unfortunately, this radiance estimate is fairly complicated since it involves the integral of a 3D convolution. This computation may be possible to express analytically for homogeneous media if we were to write out the expressions for the integration bounds more explicitly. However, for heterogeneous media, this

estimate is fairly impractical and included simply for completeness. Practically, the remaining estimates we derive are much more useful.

### 7.2 Beam Query × Beam Data, 2D Blur

As discussed earlier, reducing the blur dimensionality (and replacing integration with evaluation) increases radiance estimate efficiency.

We can derive a radiance estimate which directly computes the beam radiance along a camera ray, blurring each photon beam with a 2D kernel. Two possible routes to derive this type of estimate exist. We could take the 2D continuous photon beam estimate for in-scattered radiance at a point (Equation 25) and *integrate* along the eye ray by substituting into Equation 8. Alternatively, we could start with the beam radiance estimate using photon points in Equation 12, substitute the photon points with the photon-marching process, and take this discrete marching process to the *limit* to obtain continuous photon beams. We derive both approaches and discuss their similarities and differences.

**7.2.1 Integral of Continuous Photon Beams.** Integrating Equation 25 along the eye ray by inserting into Equation 8 yields:

$$\begin{aligned} L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) &= \sigma_s \int_0^s e^{-\sigma_i t_c} L_i(\mathbf{x}_{t_c} \rightarrow \vec{\omega}_c) dt_c, \\ &\approx \frac{\sigma_s}{\mu_R(r^2)} \int_0^s e^{-\sigma_i t_c} \sum_{b \in R_b} f(\theta_b) \Phi_b e^{-\sigma_i t_b} dt_c. \end{aligned} \quad (28)$$

This continuous 2D photon beam estimate blurs the energy of each photon beam along a 2D kernel perpendicular to the direction of the beam. The extent of this kernel along the beam forms the region  $R_b$ . Using the dual-interpretation of density estimation, we can swap the order of integration and summation, and define integration bounds per beam to arrive at:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{\sigma_s}{\mu_R(r^2)} \sum_{b \in R_b} f(\theta_b) \Phi_b \int_{t_c^-}^{t_c^+} e^{-\sigma_i t_c} e^{-\sigma_i t_b} dt_b dt_c. \quad (29)$$

The integration is along the length of the camera ray, and the bounds  $t_c^-$  and  $t_c^+$  are the intersection distances of the camera ray with each photon beam region. The integrand is a product of two transmittance terms: one along the length of the camera ray and the other along the length of the photon beam. For beam regions that vary along the length of the beam (such as cones), the region measure  $\mu_R(r^2)$  is a function of  $t_b$  and needs to be moved inside the integral. We illustrate this estimate in Figure 6(a).

In homogeneous media, the integral of the two transmittance terms can be computed analytically. For conical shaped beams, the integral becomes the well-known airlight integral [Sun et al. 2005; Pegoraro and Parker 2009]. For beams with a constant cross-section area, this can be simplified even further. To do this, we must express the projected distance along the photon beam  $t_b$  in terms of the integration variable  $t_c$ . The relationship between these two terms, illustrated by the triangle formed in Figure 6(a), is:  $t_b = t_b^- - |\cos \theta_b|(t_c - t_c^-)$ , where  $t_b^-$  is the scalar projection of the integration start point  $t_c^-$ , expressed as a distance along the photon beam, and  $\cos \theta_b = \vec{\omega}_c \cdot \vec{\omega}_b$ . This allows us to solve the integral analytically as:

$$\begin{aligned} \int_{t_c^-}^{t_c^+} e^{-\sigma_i t_c} e^{-\sigma_i t_b} dt_c &= \int_{t_c^-}^{t_c^+} e^{-\sigma_i t_c} e^{-\sigma_i(t_b^- - (t_c - t_c^-)|\cos \theta_b|)} dt_c, \\ &= \frac{e^{-\sigma_i(t_c^- - t_c^+)(|\cos \theta_b| - 1)} - 1}{e^{\sigma_i(t_c^- + t_b^-)} \sigma_i (|\cos \theta_b| - 1)}. \end{aligned} \quad (30)$$

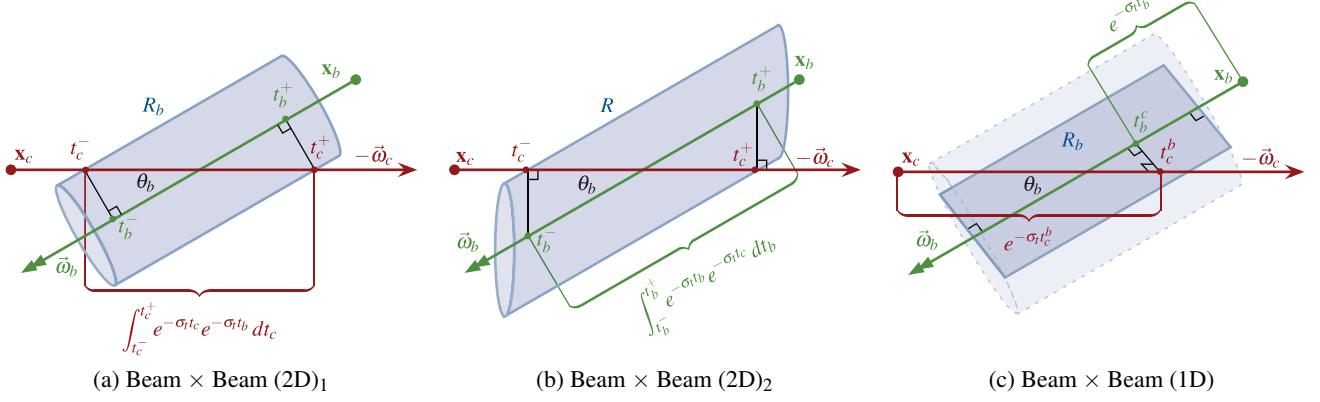


Fig. 6: Illustrations of **beam** radiance estimation using photon **beams**. We describe two possible alternatives for blurring beams in **2D**. Estimator (a) blurs perpendicular to the photon beam, expanding each photon beam into a cylinder. Estimator (b) blurs perpendicular to the camera ray, resulting in a skewed cylinder about the photon beam. The final estimator (c) simplifies these to a simple **1D** blur by compressing the cylinder into a camera-aligned rectangle.

**7.2.2 Limit of Discrete Photon Beams.** Alternatively, discrete photon-marching beams may be inserted into Equation 12:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{\sigma_s}{\mu_R(r^2)} \sum_{b \in R} \sum_{p_b \in R} f(\theta_{p_b}) \Phi_{p_b} e^{-\sigma_t t_{p,c}}, \quad (31)$$

where  $t_{p,c} = (\mathbf{x}_c - \mathbf{x}_{p_b}) \cdot \vec{\omega}_c$  is the scalar projection of the photon position onto the camera ray. Expanding the discrete photon beam power  $\Phi_{p_b}$  into Equation 31 and re-arranging terms, we get:

$$\begin{aligned} L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) &\approx \frac{\sigma_s}{\mu_R(r^2)} \sum_{b \in R} \sum_{p_b \in R} f(\theta_b) \Phi_b e^{-\sigma_t t_{p,b}} e^{-\sigma_t t_{p,c}} \Delta t, \\ &= \frac{\sigma_s}{\mu_R(r^2)} \sum_{b \in R} f(\theta_b) \Phi_b \sum_{p_b \in R} e^{-\sigma_t t_{p,b}} e^{-\sigma_t t_{p,c}} \Delta t. \end{aligned} \quad (32)$$

Note that, as in the previous derivation, this results in two transmittance terms: one which computes the transmittance from the photons towards the origin of the photon beam and one which computes the transmittance from the photons towards the origin of the camera ray.

To obtain continuous beams, we take the limit of discrete beams:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{\sigma_s}{\mu_R(r^2)} \sum_{b \in R} f(\theta_b) \Phi_b \lim_{\Delta t \rightarrow 0} \sum_{p_b \in R} e^{-\sigma_t t_{p,b}} e^{-\sigma_t t_{p,c}} \Delta t,$$

which replaces the inner summation with an integral:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{\sigma_s}{\mu_R(r^2)} \sum_{b \in R} f(\theta_b) \Phi_b \int_{t_b^-}^{t_b^+} e^{-\sigma_t t_b} e^{-\sigma_t t_c} dt_b. \quad (33)$$

Here the integration is performed along the length of the beam  $b$ , not the camera ray. The integration bounds  $t_b^-$  and  $t_b^+$  are determined by the overlap between the camera ray and the extrusion of the 2D blur kernel along the continuous photon beam. The second transmittance term uses  $t_c$ , the projected distance along the camera ray of the point at  $t_b$ . The relation between these distances is  $t_c = t_c^- - |\cos \theta_b| (t_b - t_b^-)$ , where  $t_c^-$  is the scalar projection of the integration start point,  $t_b^-$ , expressed as a distance along the eye ray, and  $\cos \theta_b = \vec{\omega}_c \cdot \vec{\omega}_b$  (see Figure 6(b)). We can use this relationship to compute the integral analytically by following a similar procedure as in Equation 30.

**7.2.3 Discussion.** The estimates in Equations 29 and 33 appear nearly identical and are indeed complements of each other (recall that complementary estimates have the same blur dimensionality, with swapped query and data representations). However, unlike previous complementary estimates, these actually compute the **same**

radiometric quantity, making them even more similar. Though nearly identical, we highlight a number of conceptual differences.

Both estimates weight photon powers with an integral of two transmittance terms: one towards the origin of the photon beam and another towards the origin of the camera ray. The main difference between these integrals is that, in Equation 29, the integration is performed along the camera ray, whereas in Equation 33 the integration is performed along the photon beam. This difference comes about from the fact that in Equation 29 the energy of the photons is always blurred perpendicular to the photon beam (forming a cylinder around the beam, as in Figure 6(a)), whereas in Equation 33 the energy of the photons is always blurred perpendicular to the camera ray (forming a sheared cylinder about the beam, as in Figure 6(b)). This results in slightly different bias between these estimates.

This can be better understood by examining the behavior of both estimates in the same situation. When the camera ray and the photon beam are perpendicular,  $\cos \theta_b = 0$  and in Equation 33 the transmittance towards the **camera** is computed exactly (not integrated) since  $t_c^+ - t_c^- = 0$ . On the other hand, the situation is reversed for the radiance estimate in Equation 29: when  $\cos \theta_b = 0$ , the transmittance towards the origin of the **photon beam** can be pulled out of the integral in Equation 30. The two estimates effectively make different bias tradeoffs when computing the two transmittance terms: equal bias is introduced in both estimates if the same radius is used, but depending on the relative orientation of the camera ray and photon beam, this bias is distributed differently between the two transmittance terms. Note, however, despite this slight difference, both estimates converge to the same answer in the limit. We did not find a noticeable difference between the results of these two estimates in our comparisons.

Of all the radiance estimates presented in this paper, the two presented in this section are the most related to the beam tracing volume caustics approach developed by Nishita and Nakamae [1994].

### 7.3 Beam Query × Beam Data, 1D Blur

When we use beams for both the query and the photons, the blur dimensionality can be reduced even further. In this section we show how to obtain a one-dimensional beam × beam blur.

To derive Equation 33 we took the 2D beam radiance estimate for photon points to the limit using photon marching. In the resulting estimate, the 2D kernels around each photon-marching photon overlap when projected onto the camera ray, and the integral is necessary to compute the amount of overlap in the limit. In Section 6.2 we

were faced with a similar situation: 3D kernels would overlap when replicated along a discrete photon beam. In that situation, we were able to construct the kernels to never overlap, and in the limit the integration was eliminated, resulting in a new radiance estimate. We apply this same principle to obtain a new beam  $\times$  beam estimate.

We proceed similarly to Section 7.2.2, but design our 2D kernels to never overlap by construction. We again start with Equation 12 and insert the discrete photon-marching beams as we did in Equation 31; however, this time we will use rectangular kernels perpendicular to the camera ray and oriented with the photon beam:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{\sigma_s}{hw} \sum_{b \in R_b} \sum_{p_b \in R_b} f(\theta_{p_b}) \Phi_{p_b} e^{-\sigma_t t_{p,c}}, \quad (34)$$

where  $w$  is the width of the rectangular kernel perpendicular to the photon beam and  $h$  is the height of the kernel parallel to the photon beam. Expanding the discrete photon beam power  $\Phi_{p_b}$  yields:

$$\begin{aligned} L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) &\approx \frac{\sigma_s}{hw} \sum_{b \in R_b} \sum_{p_b \in R_b} f(\theta_b) \Phi_b e^{-\sigma_t t_{p,b}} e^{-\sigma_t t_{p,c}} \Delta t, \\ &= \frac{\sigma_s}{hw} \sum_{b \in R_b} f(\theta_b) \Phi_b \sum_{p_b \in R_b} e^{-\sigma_t t_{p,b}} e^{-\sigma_t t_{p,c}} \Delta t. \end{aligned} \quad (35)$$

In order to prevent the photon kernels from overlapping during this expansion, the height  $h$  must be a function of the photon-marching step size  $\Delta t$ . Since the kernels are always perpendicular to the camera, they will not overlap when  $h = \Delta t \sin \theta_b$ . This is illustrated in Figure 7. Since only one kernel ever overlaps the ray, this eliminates the need for the inner summation:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{\sigma_s}{\Delta t w} \sum_{b \in R_b} \frac{f(\theta_b) \Phi_b e^{-\sigma_t t_{p,b}} e^{-\sigma_t t_{p,c}}}{\sin \theta_b} \Delta t. \quad (36)$$

We again take the limit of discrete beams to obtain continuous beams

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \lim_{\Delta t \rightarrow 0} \frac{\sigma_s}{\Delta t w} \sum_{b \in R_b} \frac{f(\theta_b) \Phi_b e^{-\sigma_t t_{p,b}} e^{-\sigma_t t_{p,c}}}{\sin \theta_b} \Delta t, \quad (37)$$

and, after expressing the kernel bandwidth  $w$  with the abstract notation used for the previous radiance estimates, we arrive at:

$$L_b(\mathbf{x}_c \leftarrow \vec{\omega}_c, s) \approx \frac{\sigma_s}{\mu_R(r)} \sum_{b \in R_b} \frac{f(\theta_b) \Phi_b e^{-\sigma_t t_b^c} e^{-\sigma_t t_c^b}}{\sin \theta_b}. \quad (38)$$

In this continuous 1D formulation,  $t_b^c$  is the distance along the photon beam to the point closest to the camera ray, and  $t_c^b$  is the distance along the camera ray to the point closest to the photon beam. We illustrate this radiance estimate in Figure 6(c).

**7.3.1 Discussion.** As in Section 6.2, we started with a radiance estimate of dimensionality  $d$  and, due to the limit behavior of the anisotropic blurring kernel, obtained a new estimate of reduced dimension  $d - 1$ . In Section 6.2 we went from an estimate that blurs in 3D to one that blurs in 2D along the beam. In this final beam  $\times$  beam radiance estimate (Equation 38), the power of the photon beams are blurred in only one dimension, along the direction perpendicular to both the camera ray and photon beam.

It is useful to discuss the connection between the 2D Beam  $\times$  Beam radiance estimates in Equations 29 and 33 with the one just derived in Equation 38. Equation 29 can be thought of as constructing a volumetric cylinder around each continuous photon beam, and integrating the power of each cylinder along the eye ray, i.e., individual photon beams would look like smoky volumetric cylinders when rendered on the screen. Equation 38 on the other hand, does not involve any integration and only divides by a 1D length measure  $\mu_R(r)$ . Conceptually, this can be thought of as replacing

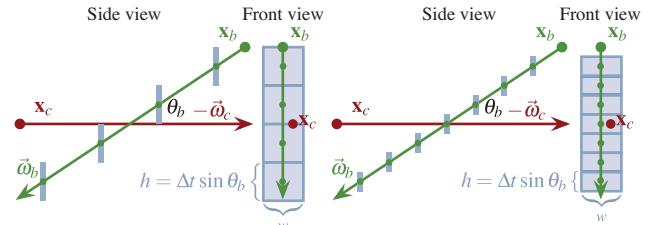


Fig. 7: To obtain the Beam  $\times$  Beam (1D) estimate in Figure 6(c), we consider the limit of non-overlapping, rectangular kernels facing the camera ray.

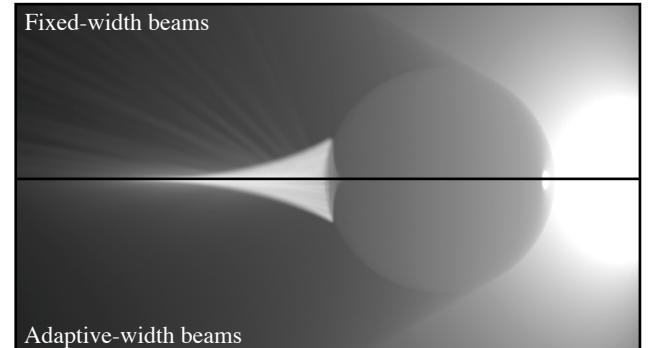


Fig. 8: Fixed-width beams (top) give sub-optimal results since the amount of blur is not adapted to the local beam density. Adaptive-width beams based on photon differentials (bottom) successfully reduce the blurring in focused regions such as the caustic, while simultaneously eliminating artifacts (e.g. banded lines emanating from the caustic) due to insufficient blur.

all the cylinders with flat rectangles which always face the camera (i.e., billboards). This conceptual interpretation is quite useful for implementation, which we discuss in more detail in Section 8.

Another aspect which deserves some discussion is the  $1/\sin \theta_b$  term. By replacing the cylinders with billboards, the integration is eliminated and replaced by a simple evaluation. However, the bounds of this integration (i.e., the distance the camera ray travels through each beam) depend on the relative orientation of the camera ray and the photon beams. If these are perpendicular, the extent of the integration will be roughly the width of the photon beam, but as they stray from perpendicularity, the extent of the camera ray through the photon beam cylinder will increase. The  $\sin \theta_b$  factor takes this change into account at a differential level, similar to the  $\cos \theta$  foreshortening term in the surface reflectance integral.

Unfortunately, the  $\sin \theta_b$  term is in the denominator, which may initially seem problematic due to the potential singularity at  $\theta_b = 0$ , when the photon beam and camera ray are parallel. Furthermore, the transmittance terms are undefined when the beams are parallel, because in this configuration there is no single point closest between two parallel lines. However, we found that in practice this estimate does not suffer from any such numerical problems.

## 8. IMPLEMENTATION

To validate and analyse the nine radiance estimates, we implement each within a simple C++ ray tracer. We followed the details of Jensen [2001] and Jarosz et al. [2008] for efficiently implementing the standard Point  $\times$  Point 3D and Beam  $\times$  Point 2D beam radiance estimates (BRE). Only a minor modification to the BRE is required to obtain our Beam  $\times$  Point 3D estimate from Section 4.2.

Photon beams can also be easily integrated into an existing volumetric photon mapping implementation. The photon shooting pre-

process remains almost completely unchanged: beam segments are stored wherever standard volume photons are stored, with initial beam start points originating at a light source. The only difference occurs at the intersection of photon paths with solid objects: while standard volumetric photon tracing discards a photon if it intersects an object prior to reaching its propagation distance, photon beam tracing stores a beam with an end point at the object intersection. The photon beams map contains a super-set of the information in a standard photon map, as illustrated by the photon paths in Figure 1.

### 8.1 Adaptive Beam Width

In order to estimate the radiance from a photon beams map, we need to be able to compute the local density of beams. For simplicity, the theory in the previous sections has assumed an abstract, fixed blurring kernel. A fixed-width blur can easily be implemented by interpreting each photon beam as a cylinder of a specified radius (Figure 8 top). However, choosing a single blurring width that works well for the whole domain is often impossible, a problem that has been well-studied in density estimation literature [Silverman 1986]. A better technique is to adjust the kernel width based on the local photon beam density. Standard photon mapping uses the k-nearest neighbors to locally adjust the search radius, while Jarosz et al. [2008] used the variable kernel method with a pilot density estimate to assign radii to photon points. Unfortunately, reusing either of these techniques is challenging in our context since photon beams are higher dimensional and we would like the blur radius to potentially vary along the length of the photon beams according to the local density.

We explored several approaches for adapting beam widths, including pdf-based methods [Herzog et al. 2007; Suykens and Willem 2000; Suykens and Willem 2001], and techniques based on pilot density estimates at beam vertices [Jarosz et al. 2008]. Unfortunately, these approaches did not produce satisfactory results. Instead, we found that photon differentials [Schjøth et al. 2007] provide an effective and automatic solution for estimating the divergent spreading and convergent focusing of photon beams as they propagate within a scene.

**Photon Differentials.** Photon differentials are the analogue of ray differentials [Igehy 1999] from the direction of the light. Ray differentials were originally developed to estimate the footprint of camera rays for improved texture filtering, but have also been applied in “reverse” to estimate desired photon footprints or blur-radii for surface-based [Schjøth et al. 2007; Fabianowski and Dingliana 2009] as well as volumetric photon mapping [Schjøth 2009]. We use photon differentials to determine the **size** and **shape** of each photon beam.

A photon differential consists of the photon beam position and direction,  $(\mathbf{x}, \vec{\omega})$ , as well as two auxiliary rays,  $(\mathbf{x}_u, \vec{\omega}_u)$  and  $(\mathbf{x}_v, \vec{\omega}_v)$ , with offset origins and directions. Together, the central beam ray and its two differential rays form a truncated cone, or conical frustum, which defines the size and shape of each photon beam for density estimation. At any point  $t$  along the beam, the elliptical cross-section of the frustum has semi-major and semi-minor axis lengths equal to the Euclidean distance between the point on the central ray,  $r(t) = \mathbf{x} + t\vec{\omega}$ , and the corresponding points on the differential rays,  $r_u(t) = \mathbf{x}_u + t\vec{\omega}_u$  and  $r_v(t) = \mathbf{x}_v + t\vec{\omega}_v$ .

At reflective and refractive surface interactions, we modify differential rays according to Igehy [1999]. For example, in Figure 8, photon beam differentials are modified to focus and spread the beams of light according to the refraction into, and out of the glass sphere. We found that this approach works very well, handling

extreme spreading and focusing caused by caustic light paths.

**Light Source Emission.** Light source emission affects the initial position, direction, and width of each emitted beam’s differential rays. Generally, we aim to set differentials so as to create a tight packing of beams over the spatial and angular extent of the light source. We adopt different emission schemes for different light sources. More specifically, the position and orientation of differential rays is determined by the spatial and directional PDFs of photon emission, and the total number of emitted photons from a light.

For singularity lights, such as point-, spot-, and cosine-lights, the differential ray origins are set to the location of the light source. We set the differential ray directions such that the entire angular extent of the source is covered by the number of emitted beams, while minimizing the amount of overlap between beams. More precisely, if we emit  $N$  photon beams, the solid angle spanned by a beam’s footprint is set to  $1/(pdf(\vec{\omega}) \cdot N)$ , where  $pdf(\vec{\omega})$  is the probability density of emitting a photon with direction  $\vec{\omega}$ . For instance, for an isotropic point-light, each photon beam footprint would be allotted  $4\pi/N$  steradians. Since each beam forms a cone, we can therefore solve for the necessary apex angle between the central ray and the photon differentials. This process is similar to the one used by Fabianowski and Dingliana [2009], but extended to work with general angular distribution. We illustrate this in Figure 9.

Our implementation also supports area light sources. In this case, the surface area PDF must also be considered. As with directions, we allot a portion of the light’s surface area to each of the emitted photon differentials. Conceptually, we discretize the light’s surface into a jittered grid and assign central and differential ray origins based on the jittered grid centers and edges, respectively. Differential and central beam directions are set according to the PDF of the angular emission of the source (see Figure 9). Figure 10 shows a scene that uses photon differentials for area light source emission.

With this photon differential scheme, the radiance estimate would correctly return infinity at the position of a singularity light source (since the width would be zero), while this singularity would be correctly avoided for area light sources due to the finite cross-section.

**Multiple Scattering.** Just as multiple scattering photons can be deposited during standard photon shooting, multiple scattering beams can also be deposited with a random-walk shooting process.

We have investigated several differential modulation schemes for multiple scattering beams, including one inspired by the re-

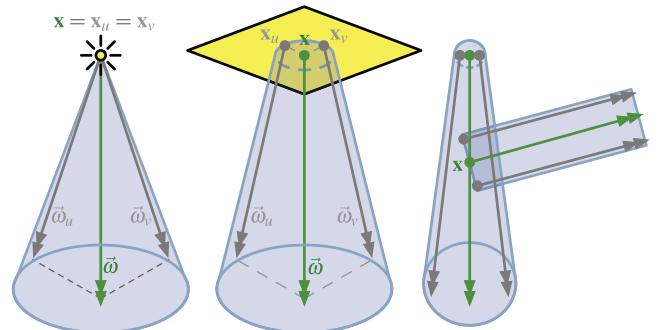


Fig. 9: We use photon differentials to compute the conical blurring frustum around each photon beam. For point light emission (left), the differential ray directions determine the spread of the cone, whereas, for area lights (middle), the positional differentials are also considered. Specular interactions propagate the differentials. Multiple scattering beam differentials (right) are always parallel to the parent beam and spaced at a distance equal to the parent’s radius at the scattering location  $\mathbf{x}$ .

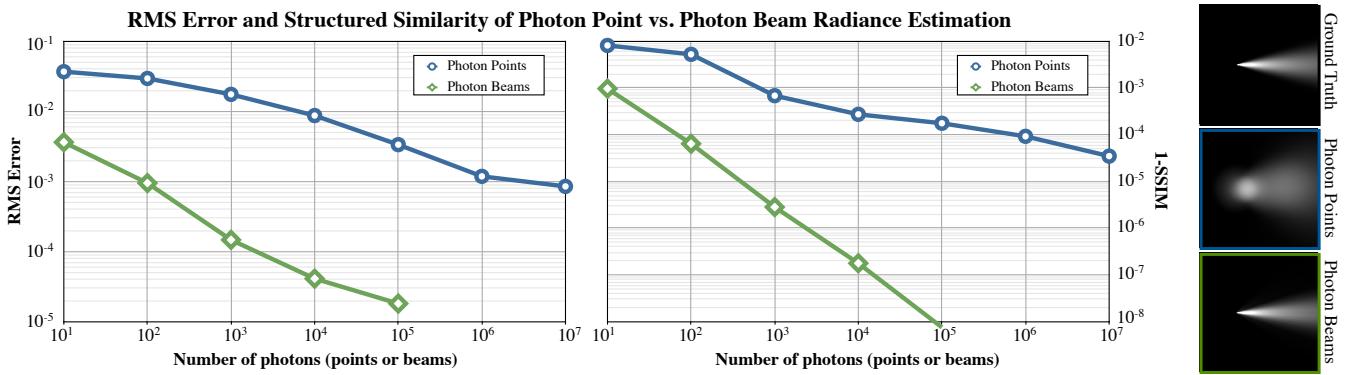


Fig. 11: We compare the error convergence of photon beams vs. photon points on a log-log scale in a test scene containing a focused lighting effect similar to a volumetric caustic. Our method requires 10,000 times fewer photon beams than photon points to obtain the same RMS error. A crop of the reference solution (top) is shown, as well as the photon points (middle) and photon beams (bottom) reconstruction using 100 photon point and beams respectively.

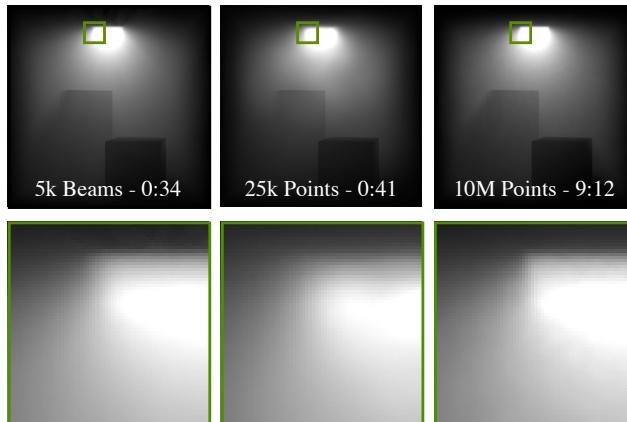


Fig. 10: Adaptive photon beam width for an area light source rendered using our photon differentials (left), compared to a roughly equal-time (middle) and roughly equal quality (right) renderings with photon points.

cent “absorb-and-reemit” analogy suggested by [Fabianowski and Dingliana \[2009\]](#) for diffuse inter-reflections, as well as pdf-based approaches [[Suykens and Willems 2000; Suykens and Willems 2001; Herzog et al. 2007; Schjøth 2009](#)]. Unfortunately, we did not find these techniques to work well with photon beams since the extreme spreading of beams resulted in visibly biased results. Instead, for multiple scattering, beams are always cylinders with widths set according to the footprint of the parent beam at the multiple scattering event location (see Figure 9). We found that this automatic process produces reasonable results. Figure 13 compares photon beams to standard photon mapping in a scene with two bounces of multiple scattering.

## 8.2 Global Beam Width and Smoothing Kernels

In addition to using photon differentials to automatically predict the spreading and focusing of photon beams, we provide the user with a width multiplication factor,  $\alpha$ , which can globally increase or decrease the computed beams widths. This serves the same purpose as the  $k$  parameter used in standard  $k$ -nearest neighbor density estimation, and allows the user to adjust the tradeoff between banding and blur in the rendered image. Furthermore, since we allow the user to choose a smooth weighting kernel, some amount of overlap ( $\alpha > 1$ ) is desirable for optimal results. We experimented with several blur widths and smoothing kernels (including constant,

cone, 3rd- and 5th-order smoothstep, biweight, and Gaussian) and found the biweight kernel [[Silverman 1986](#)] ( $K(x) = \frac{15}{16}(1-x^2)^2$  for  $x \in [0, 1]$ ) with an  $\alpha = 3.0$  strikes a good balance between kernel overlap, blurring and banding. We used this kernel in all our results.

## 8.3 Photon Beam Storage

To compute the beam radiance estimate, we need an efficient way to intersect a camera ray with the collection of photon beams in the scene. Inspired by work in ray-hair intersection [[Nakamaru and Ohno 2002; Apodaca and Gritz 1999](#)], we have explored several acceleration schemes (including KD-Trees, BVHs and grids). After experimenting with these techniques we found that, for our problem domain, a BVH with some modifications performs quite well.

We store photon beams in a BVH constructed using the surface area heuristic [[MacDonald and Booth 1990](#)]. Unfortunately, since many beams may overlap the same region of space (and in fact may share a common vertex such as all beams emanating from a point light source) a naïve application of a BVH is inefficient. To improve partitioning by the BVH, we split beams into sub-beams. The splitting length is set automatically by our construction algorithm to produce sub-beams with approximately unit aspect ratio. At run-time, we are able to quickly obtain *all beams* intersected with an eye ray, allowing rapid radiance estimation.

## 9. RESULTS

We have implemented all nine radiance estimates to validate the theory, and demonstrate the effectiveness of photon beams. All render times for these estimates were measured on a MacBook Pro with a 3.06 GHz Intel Core 2 Duo processor and 4 GB of RAM. Note that our current implementation uses only one core; however, since radiance estimation is a read-only operation, all estimates could easily be parallelized in an optimized implementation. The Cornell box, bumpy sphere, and sphere caustic images are 512 pixels across their larger dimension while all other images are 1024 pixels. All results were rendered with up to 16 samples per pixel.

Figure 3 shows a Cornell box rendered with each of the estimators. Note that even though the query domains and data representations are vastly different among the estimators, the resulting images all faithfully approximate the same solution. Here our intent is to verify the correctness of all the derived estimators, and not on performance, so we omit render times.

In our remaining results we demonstrate the benefits of photon beams over photon points. We compare to the previous state-of-

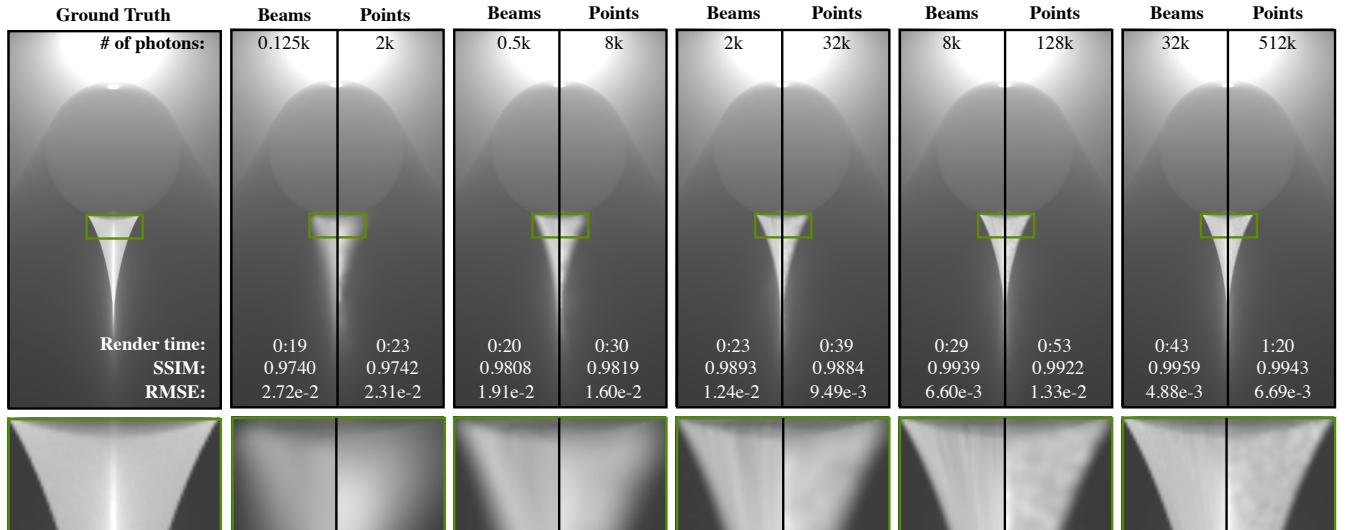


Fig. 12: A progression of renderings of a glass sphere forming a volume caustic. We compare the convergence of radiance estimation using photon beams (left half), and photon points (right half of each image). In each image pair we compare photon beams to a result generated using 16× as many photon points. Photon beams achieve higher quality in less time than photon points, even when using a significantly smaller number of beams than points.



Fig. 13: Cornell box with an area light and multiple scattering. Left to right: photon beams, photon points with roughly equal-time and equal-quality.

the-art Beam × Point 2D estimate in terms of image quality and performance. We found that for this estimator the extra cost of computing photon radii using a pilot density estimate [Jarosz et al. 2008], as opposed to using photon differentials [Schjøth 2009], is negligible and produces superior results. We therefore use this approach for all photon points results in our comparisons. For photon beams, we focus on the Beam × Beam estimators since they avoid ray marching. From a theoretical perspective, all Beam × Beam estimates provide a benefit over photon points in terms of image quality. However, as mentioned in Section 7.1, the Beam × Beam 3D estimate is fairly expensive to implement since it requires a 3D convolution. The 2D and 1D variants are much more practical from an implementation perspective, while retaining all the benefits. We believe the Beam × Beam 1D estimate (Equation 38) and Beam × Beam 2D<sub>1</sub> estimate (Equation 29) are the most promising choices. The remaining results focus on the 1D version because it retains the benefits of a Beam × Beam estimate and is computationally simpler to evaluate.

Figure 10 illustrates the ability of photon beams to efficiently handle area light sources. We render an artifact-free image with just 5k photon beams, whereas using 25k photon points in roughly equal time results in over-blurring (near the light source) and no visible volumetric shadow from the tall box. With 10M photon points, over-blurring artifacts vanish, but the render time is over 16 times longer and variance is still present in the estimate, resulting in a slightly bumpy appearance. Note also that our use of photon differentials works well, even for area light sources, to adaptively blur the photon beams based on the expected local density. The benefits of photon

beams for multiple scattering are clearly visible in Figure 13.

In Figure 11 we perform a quantitative convergence comparison between photon points and photon beams. Unfortunately, there are currently no algorithms which can generate unbiased reference solutions for general caustics from point lights. To circumvent this problem we constructed a focused lighting feature similar to a volume caustic (but which has a readily computable reference solution) using single scattering from a Phong-light with a high exponent. The reference solution was computed using ray marching with a very small step-size. We plot the RMS error, on a **log-log** scale, when reconstructing using photon beams and photon points. We also compute and plot the perceptual error based on the structural similarity index (SSIM) [Wang et al. 2004]. Both graphs show that we would need between 1M and 10M photon points to obtain the same error as just 100 photon beams, an improvement of over 4 orders of magnitude. These photons not only need to be traced, but also stored explicitly in the photon map. Photon beams provide a benefit in both execution time and storage requirements since they are a more compact representation of the photon tracing stage.

For the remaining scenes, we compute ground truth images with volumetric photon mapping using billions of photons. To obtain reference quality results, we set the projected radius of each photon to be smaller than a pixel (to minimizes bias), and, to avoid the memory requirements for storing billions of photons, we compute several hundred independent renderings and average the results. We include the averaged, converged results as ground truth and report RMSE and SSIM values for each image.

In Figure 12 we show a sequence of renderings of a volume caustic from a glass sphere (inspired by the scene from Jensen and Christensen [1998]) to examine the qualitative and quantitative convergence behavior of photon beams and photon points. We increase the number of photons from left to right and split each image vertically to compare beams and points. Each split image compares photon beams to 16 times as many photon points. We provide a ground truth image for comparison (using around 1 billion effective photon points) and report RMSE and SSIM values. Even with 16 times as many points as beams, in each case, using photon beams is faster and results in less blur and noise. Photon beams are a perfect

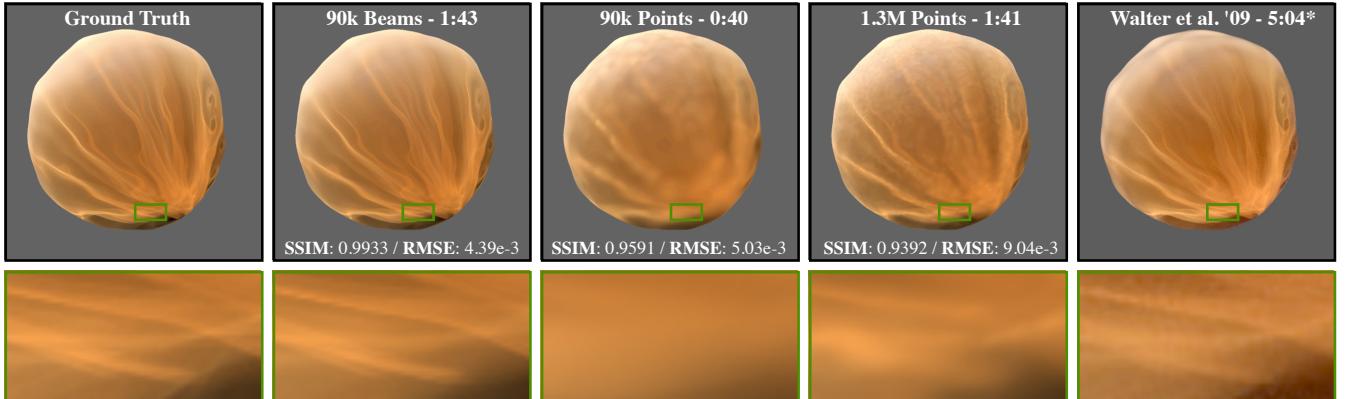


Fig. 14: An equal-time, and equal number-of-photons comparison of photon points and photon beams in the bumpy sphere test scene Walter et al. [2009]. Photon beams accurately capture complex focusing patterns within the sphere with only 90k beams. The same number of photon points renders faster but with excessive blurring. At 1.3M photon points (equal time), blurring and variance still remain. We also provide a ground truth image (left) computed using several billion photons. For reference, we also compare our result with a rendering using Walter’s method (right), which takes 3 times as long to render (\*using a different implementation on a machine with 8 cores), and still contains noise due to the need for ray marching.



Fig. 15: A re-creation of the lighthouse scene from Jarosz et al. [2008]. Using photon beams we can render a higher quality image, using fewer photons and in less time when compared to the beam radiance estimate using photon points.

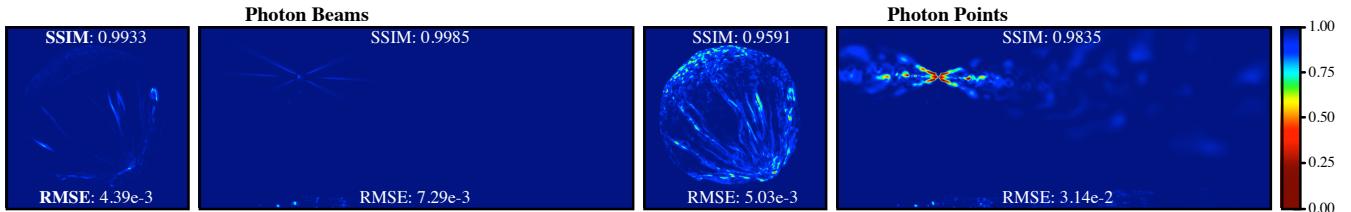


Fig. 16: False-color visualizations of the per-pixel structural similarity index for equal-time renderings of the bumpy sphere and lighthouse scenes.

representation for this type of focused illumination and can therefore converge to a crisp and noise-free solution using only 32k beams in 43 seconds, whereas even with 512k photons and almost twice the time, the caustic is over-blurred and noisy. In Figure 8 we showed the improvement in reconstruction for this scene due to using an adaptive beam width based on photon differentials. Fixed-width beams result in over-blurring and banding, whereas differentials allow for a small beam width in the focused regions of the volume caustic and avoid potential banding issues in sparse regions.

Figure 14 replicates the bumpy sphere scene from Walter et al. [2009] rendered with photon beams, photon points, and Walter’s direct algorithm. We also provide a ground truth image rendered using 2.5 billion photons. In about 100 seconds, we are able to reconstruct extremely sharp lighting features using only 90k photon beams while with the same number of photon points, the BRE cannot resolve these features, suffering from over-blurring. Given equal render time, we can render the scene using 1.3M photon points; however, the resulting image still cannot resolve the small features and suffers from a bumpy or noisy appearance. Moreover, when using photon points, the stochastic photon tracing process results in distracting flickering as this scene is animated, whereas this flickering is not present when using photon beams (see the accompanying

video). The last image was kindly provided by Bruce Walter and rendered using his and his colleagues’ approach [Walter et al. 2009]. One of the strengths of this method is that it directly solves for valid refracted light paths without the need for an intermediate lighting representation, so it does not suffer from blurring and can resolve fine details not possible with standard photon mapping. This image was rendered in a different renderer, and timing was measured on an 8-core, 2.83 Ghz machine with dual Intel Xeon 5440 chips. We see that Walter’s method is about 3 times as slow rendering on 8 cores compared to photon beams on only one core. Though part of this performance penalty could be attributed to differences in implementation framework, this great disparity also suggests algorithmic factors in performance. Part of this could be due to Walter’s method requiring ray marching to solve for the accumulated in-scattered radiance along a ray (so a small amount of noise is still visible) whereas our method solves for the beam radiance explicitly without the need for ray marching, resulting in a completely noise-free solution. Walter’s method also has a number of practical limitations. The method is currently restricted to a single specular bounce, so while it can handle a single refraction in this bumpy sphere scene, it could not render a volume caustic from a sphere as seen in Figure 12. Even in the bumpy sphere scene, there are valid light paths which refract



Fig. 17: An animation frame of underwater sun rays refracting through an ocean surface. Photon beams (left) accurately reconstruct this lighting with only 25k beams. In equal time, 100k photon points (middle) could be stored, but the resulting reconstruction suffers from noisy and over-blurred lighting features.

more than once within the medium due to total internal reflection, which we take into account with photon beams, but which Walter’s method cannot simulate. Note also that although Walter’s method is in theory unbiased, in practice it requires clamping to avoid bright pixels, which can lead to an arbitrary amount of energy loss. Our method does not require such clamping. Lastly, Walter’s method only operates on triangle meshes (admittedly a fairly reasonable restriction). Photon beams can be used with any arbitrary surface representation, including implicit surfaces and procedural geometry.

In Figure 15 we show a frame from an animated re-creation of the lighthouse scene [Jarosz et al. 2008]. In 25 seconds we render a frame with no artifacts using 700 photon beams, while even with 10k photon points and slightly greater render time, low-frequency noise is clearly visible, which also flickers when animated.

Figure 16 visualizes false-color per-pixel structural similarity for the bumpy sphere and lighthouse scenes and reports RMSE and SSIM values. Photon points have difficulty reconstructing sharp lighting features like the start of the lighthouse beam, or the intricate light patterns inside the bumpy sphere, resulting in noticeable error.

Finally, Figure 17 visualizes underwater beams of light refracting through an animated ocean water surface. The ground truth image required around 8 billion photon points for a converged result. Here, the sun *beams* are much more efficiently represented with photon *beams*, resulting in a more faithful, noise-free reconstruction without temporal flickering. In contrast, with an equal-time photon point rendering, the individual beams of light are barely discernible. The accompanying video shows an animated comparison.

## 10. DISCUSSION

In addition to providing a practical rendering algorithm for participating media, the radiance estimates we present also offer interesting insights into connections between otherwise fairly distinct rendering algorithms. In this section we discuss how photon beams can be seen as a generalization of a number of other algorithms.

### 10.1 Generalization of Photon Mapping

Photon beams can clearly be seen as a generalization of standard photon mapping where a more compact data representation is used to connect the photon tracing and radiance estimation stages. Our theory of photon beams can also be interpreted as a generalization of the ray mapping algorithm [Lastra et al. 2002; Havran et al. 2005; Herzog et al. 2007] to participating media. This suggests the possibility of applying photon beams to other related problems where photon maps and ray maps have been applied, including the rendering of hair [Moon and Marschner 2006] and subsurface scattering [Jensen 2001]. Moreover, a single collection of photon beams can be used to estimate lighting within media using our estimators, and lighting on surfaces using ray mapping density estimation. This

eliminates the need to store distinct representations for these two types of lighting simulations.

### 10.2 Generalization of Beam Tracing Methods

Light beam tracing [Watt 1990; Iwasaki et al. 2001; Ernst et al. 2005] simulates reflective or refractive caustics by extruding beams or prisms by the bending light at vertices of a triangle mesh. This has been applied to participating media as well [Nishita and Nakamae 1994], and in fact the computation of in-scattered radiance with these methods is nearly mathematically identical to the radiance estimates in Equations 29 and 33. In this sense, photon beams can be viewed as a generalization of beam tracing methods, where the beam representation is no longer dictated by the geometric representation. Instead, emitted photons implicitly define beams, whereas mesh triangles defines beams/prisms in beam tracing methods. This is a significant benefit since photon beam complexity depends on induced radiometric complexity, not geometric complexity.

### 10.3 Generalization of Shadow Maps

Photon beams can also be viewed as a direct generalization of shadow maps in participating media. This connection can be seen by observing individual beams in Figure 3. When simulating single scattering from a spot-light, photon beams contain the same information as a shadow map – the visibility from surface points back to the light source. If the photons were emitted on a regular grid from the spot-light, then these two data representations would have a one-to-one mapping: each photon beam corresponds to a single shadow map pixel. Hence, for situations where a shadow map could be used, we could treat the shadow map as an implicit representation of the photon beams and, without any acceleration structure, could perform camera ray intersection with all shadow map beams by stepping through the shadow map pixel grid. Note, however, that photon beams are more flexible than shadow maps since they allow for area light sources and multiple specular and diffuse bounces.

### 10.4 Connection to VPL-Methods and Final Gather

Virtual point-light methods such as Instant Radiosity [Keller 1997] and Lightcuts [Walter et al. 2006] interpret a photon map’s photons as point-light sources that illuminate other, potentially distant, regions of the scene. In this interpretation, the contribution of each photon or VPL includes an inverse-squared distance term which needs to be clamped to avoid singularities near the VPL. In contrast, in the density estimation interpretation, this clamping is not necessary and each photon’s contribution is spread within a small spatial neighborhood around the photon, inducing a blurring of the illumination. The generalization of photon points to photon beams provides another way to arrive at this dual interpretation.

With photon beams, much like VPL methods, a photon's power contributes to a large region in space, potentially far away from the start of the beam. In effect, the point at the start of the photon beam is a photon point (see Figure 1) that illuminates the scene with a beam of light in a particular direction. Furthermore, if we use a conical blur around the beam with the estimate in Equation 25,

$$L_i(\mathbf{x} \rightarrow \vec{\omega}) \approx \sum_{b \in R_b} \frac{f(\theta_b)\Phi_b e^{-\sigma_t t_{\mathbf{x},b}}}{\mu_R(t_{\mathbf{x},b}^2)}, \quad (39)$$

we re-introduce the familiar inverse-squared falloff term present in VPL methods since the cross-section of the cone goes to zero as we approach the start of the photon beam. Note that this radiance estimate is identical to looping over a collection of VPLs with power  $\Phi_b$ . Hence, by simulating each photon beam as a photon “cone”, photon beams effectively compute a VPL solution where the VPLs are beam- or spot-lights, and would obtain mathematically similar results to unclamped VPL methods for participating media.

The connection to VPL methods goes even further. The main motivation for Lightcuts is to reduce the number of shadow rays necessary to evaluate lighting from VPLs. Given that Equation 39 is effectively a VPL estimate, we can further note that it actually requires no shadow rays. The visibility information established during the photon tracing stage is re-used within each beams' search region during radiance estimation, thereby eliminating the need for any shadow rays. Photon beams can therefore be thought of as a form of Instant Radiosity without the need for any shadow rays, plus with the added ability to estimate the integrated in-scattered radiance along the length of an entire camera ray.

Recently, Hašan et al. [2009] introduced the concept of the virtual sphere-light (VSL) as a way to avoid the problems with the inverse squared falloff term in surface-based VPL methods. Essentially, each VSL's energy is spread across a finite area on nearby surfaces by expanding a sphere, and the corresponding energy “splat” is treated as an area light source when evaluating the indirect illumination at other scene locations. The singularity is avoided since the lighting contribution is integrated over a finite solid angle. We implicitly benefit from this same enhancement with the photon beams framework. With a conical blur region, the beam width falls off to zero at the start of the beam and induces an infinite radiance; however, the cross-section at the start of a beam need not be zero. In our implementation, beams emitted from area lights have a finite cross-section and, for secondary scattering, we use the photon differentials' footprint to compute the cross-section at the start of the child beam. This induces a conical frustum shape and avoids the division by zero. Hence, photon beams can be seen as a generalization of VPLs where the clamping is avoided by always forcing non-zero starting cross-sections (replacing potentially unbounded energy loss with blurring), and allows for computing the beam radiance directly using the Beam  $\times$  Beam estimates (without the need for ray marching).

Finally, using similar reasoning, the photon beam radiance estimates can also be interpreted as a selective form of final gathering. For instance, Equations 25 and 39 compute the value of in-scattered radiance at a point  $\mathbf{x}$  by examining all photon points (beam start points) which “send energy” towards  $\mathbf{x}$ . This could instead have been computed by tracing final gather rays out from  $\mathbf{x}$  to find these photons. However, as mentioned above, in contrast to final gathering or VPL methods, the beam radiance estimate does not require any ray intersections with the scene to determine visibility. The visibility between the query location and all photon points is encoded in the beams themselves. Furthermore, the Beam  $\times$  Beam radiance estimates effectively compute final gathering from all points along the length of a camera ray without the need for ray marching.

## 11. LIMITATIONS & FUTURE WORK

The connections discussed above suggest a number of interesting avenues of future work. In addition, our prototype implementation, though effective, does still have a number of limitations which provides opportunities for further improvement and investigation.

### 11.1 Heterogeneous Media

One of the limitations of our current implementation is the restriction to homogeneous media. This, however, is not a theoretical restriction of photon beams, but simply of our prototype implementation. The theory applies equally well to heterogeneous media.

To handle heterogeneous media, we would need to numerically compute the transmittance along camera rays and photon beams. Heterogeneous media along the camera ray could easily be integrated into the current framework without much modification; however, handling heterogeneous transmittance along photon beams is potentially more costly. To do so efficiently, the photon beams data representation could be extended to store a 1D transmittance function along each beam. This is analogous to the generalization of shadow maps to deep shadow maps [Lokovic and Veach 2000], which store a 1D transmittance function over depth instead of just binary visibility. This transmittance would be computed and stored during the photon tracing stage for efficient evaluation during rendering. As with deep shadow maps, these 1D transmittance functions could be significantly compressed for efficient storage.

### 11.2 Efficient Beam $\times$ Beam Query

We currently use a BVH over all beams to accelerate the Beam  $\times$  Beam queries (Section 8.3). Although the BVH performs reasonably well, there is still room for improvement. We are currently exploring ways to develop a tailored data structure for beam storage and intersection, taking into account the unique geometric properties and access patterns of this particular problem domain.

Furthermore, we handle adaptive blur kernels by defining beam widths using photon differentials. This worked quite well for single scattering and caustics, but was not as successful for multiple scattering. Another interesting avenue of future work could explore the use of a true k-nearest-neighbor approach to adapting the blurring kernel with photon beams. This is a challenging geometric problem since an appropriate definition of “nearest” would need to be derived.

## 12. CONCLUSION

We have developed a novel, comprehensive theory of volumetric radiance estimation. This theory allows for estimating in-scattered radiance at a point, or the accumulated in-scattered radiance along a camera beam. Both of these operations can be performed using the standard photon map representation as done in previous work. Moreover, our theory generalizes both of these operations by introducing the concept of photon beams, a more compact, and expressive intermediate representation of lighting in participating media. The combination of these two data representations and two query operations results in a collection of nine distinct radiance estimates for computing complex lighting in participating media.

Due to the increased expressiveness of this new lighting representation, photon beams have a significant performance and quality benefit over standard volumetric photon mapping, while requiring less photons. This representation can also be viewed as a way to implicitly increase the effective “resolution” of a photon map, thereby reducing bias (blur) and also significantly reducing variance. Using this representation, we are able to render extremely sharp details

(such as volume caustics) using just tens of thousands of photon beams, whereas this would have required millions or billions of photons points with previous methods.

#### ACKNOWLEDGMENTS

The authors would like to thank Bruce Walter for providing the bumpy sphere scene data as well as the image rendered using his algorithm in Figure 14. We would also like to thank Craig Donner for providing base code for generating the water surface.

#### REFERENCES

- APODACA, A. A. AND GRITZ, L. 1999. *Advanced RenderMan: Creating CGI for Motion Pictures*. Morgan Kaufmann, San Francisco, CA, USA.
- ARVO, J. AND KIRK, D. 1990. Particle transport and image synthesis. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*. ACM, New York, NY, USA.
- BOUDET, A., PITOT, P., PRATMARTY, D., AND PAULIN, M. 2005. Photon splatting for participating media. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*. GRAPHITE '05. ACM, New York, NY, USA, 197–204.
- CEREZO, E., PÉREZ, F., PUEYO, X., SERON, F. J., AND SILLION, F. X. 2005. A survey on participating media rendering techniques. *The Visual Computer* 21, 5.
- CHANDRASEKAR, S. 1960. *Radiative Transfer*. Dover Publications, NY.
- COLEMAN, W. 1968. Mathematical verification of a certain Monte Carlo sampling technique and applications of the technique to radiation transport problems. *Nuclear Science and Engineering* 32, 76–81.
- ERNST, M., AKENINE-MÖLLER, T., AND JENSEN, H. W. 2005. Interactive rendering of caustics using interpolated warped volumes. In *Proceedings of Graphics Interface*. Waterloo, Ontario, Canada, 87–96.
- FABIANOWSKI, B. AND DINGLIANA, J. 2009. Interactive global photon mapping. *Computer Graphics Forum (Proceedings of EGSR 2009)* 28, 4, 1151–1159.
- HAVRAN, V., BITTNER, J., HERZOG, R., AND SEIDEL, H.-P. 2005. Ray Maps for Global Illumination. In *16th EG Workshop on Rendering*. 43–54.
- HAŠAN, M., KRIVÁNEK, J., WALTER, B., AND BALA, K. 2009. Virtual spherical lights for many-light rendering of glossy scenes. *Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009)* 28, 5.
- HECKBERT, PAUL, S. AND HANRAHAN, P. 1984. Beam tracing polygonal objects. In *SIGGRAPH*. ACM, New York, USA, 119–127.
- HERZOG, R., HAVRAN, V., KINUWAKI, S., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2007. Global illumination using photon ray splatting. In *Computer Graphics Forum (Proceedings of Eurographics 2007)*. Blackwell, Czech Republic, 503–513.
- IGEHY, H. 1999. Tracing ray differentials. In *SIGGRAPH*. ACM Press/Addison-Wesley Publishing Co., New York, USA, 179–186.
- IWASAKI, K., NISHITA, T., AND DOBASHI, Y. 2001. Efficient rendering of optical effects within water using graphics hardware. In *Pacific Graphics*. IEEE Computer Society, Washington, DC, USA, 374.
- JAROSZ, W., ZWICKER, M., AND JENSEN, H. W. 2008. The Beam Radiance Estimate for Volumetric Photon Mapping. *Computer Graphics Forum (Proceedings of Eurographics 2008)* 27, 2 (Apr.), 557–566.
- JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., Natick, MA, USA.
- JENSEN, H. W. AND CHRISTENSEN, P. H. 1998. Efficient Simulation of Light Transport in Scenes with Participating Media Using Photon Maps. In *SIGGRAPH*. ACM Press, New York, USA, 311–320.
- KAJIYA, J. T. 1986. The rendering equation. In *SIGGRAPH*. ACM Press, New York, USA, 143–150.
- KELLER, A. 1997. Instant radiosity. In *SIGGRAPH*. Computer Graphics Proceedings, Annual Conference Series. 49–56.
- KRÜGER, J., BÜRGER, K., AND WESTERMANN, R. 2006. Interactive screen-space accurate photon tracing on GPUs. In *Rendering Techniques*.
- LAFORTUNE, E. P. AND WILLEMS, Y. D. 1993. Bi-directional path tracing. In *Compugraphics*.
- LAFORTUNE, E. P. AND WILLEMS, Y. D. 1996. Rendering participating media with bidirectional path tracing. In *EG Rendering Workshop*. 91–100.
- LASTRA, M., UREÑA, C., REVELLES, J., AND MONTES, R. 2002. A Particle-Path based Method for Monte Carlo Density Estimation. In *13th EG Workshop on Rendering*. EG Association, 7–14.
- LOKOVIC, T. AND VEAH, E. 2000. Deep shadow maps. In *SIGGRAPH*. ACM Press, New York, NY, USA, 385–392.
- MACDONALD, D. J. AND BOOTH, K. S. 1990. Heuristics for ray tracing using space subdivision. *The Visual Computer* 6, 3, 153–166.
- MITCHELL, D. AND HANRAHAN, P. 1992. Illumination from curved reflectors. In *SIGGRAPH*. ACM, New York, USA, 283–291.
- MOON, J. T. AND MARSCHNER, S. R. 2006. Simulating multiple scattering in hair using a photon mapping approach. In *Transactions on Graphics (Proceedings of SIGGRAPH 2006)*. ACM, NY.
- NAKAMARU, K. AND OHNO, Y. 2002. Ray tracing for curves primitive. In *WSCG*. 311–316.
- NISHITA, T., MIYAWAKI, Y., AND NAKAMAE, E. 1987. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. In *SIGGRAPH*. ACM, New York, USA, 303–310.
- NISHITA, T. AND NAKAMAE, E. 1994. Method of displaying optical effects within water using accumulation buffer. In *SIGGRAPH*. ACM, New York.
- PAULY, M., KOLLIG, T., AND KELLER, A. 2000. Metropolis light transport for participating media. In *11th EG Workshop on Rendering*. 11–22.
- PEGORARO, V. AND PARKER, S. G. 2009. An Analytical Solution to Single Scattering in Homogeneous Participating Media. *Computer Graphics Forum (Proceedings of Eurographics 2009)* 28, 2, 329–335.
- PHARR, M. AND HUMPHREYS, G. 2004. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, San Francisco, USA, Chapter 14, 641–644.
- RAAB, M., SEIBERT, D., AND KELLER, A. 2008. Unbiased global illumination with participating media. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*. Springer, 591–606.
- SCHJØTH, L. 2009. Anisotropic density estimation in global illumination: a journey through time and space. Ph.D. thesis, University of Copenhagen.
- SCHJØTH, L., FRISVAD, J. R., ERLEBEN, K., AND SPORRING, J. 2007. Photon differentials. In *GRAPHITE*. ACM, New York, USA, 179–186.
- SCHJØTH, L., OLSEN, O. F., AND SPORRING, J. 2006. Diffusion based photon mapping. In *GRAPP 2006: Proceedings of the First International Conference on Computer Graphics Theory and Applications, Setúbal, Portugal, February 25–28, 2006*. 168–175.
- SCHJØTH, L., SPORRING, J., AND OLSEN, O. F. 2008. Diffusion based photon mapping. *Computer Graphics Forum (Proceedings of Eurographics 2008)* 27, 8 (Dec.), 2114–2127.
- SILVERMAN, B. 1986. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability. Chapman and Hall, New York.
- SUN, B., RAMAMOORTHI, R., NARASIMHAN, S. G., AND NAYAR, S. K. 2005. A practical analytic single scattering model for real time rendering. *Transactions on Graphics (Proceedings of SIGGRAPH 2005)* 24, 3, 1040–1049.
- SUYKENS, F. AND WILLEMS, Y. D. 2000. Adaptive filtering for progressive monte carlo image rendering. In *Vis. and Interactive Digital Media*.
- SUYKENS, F. AND WILLEMS, Y. D. 2001. Path differentials and applications. In *Rendering Techniques*. Springer-Verlag, London, UK, 257–268.

- VEACH, E. 1997. Robust Monte Carlo methods for light transport simulation. Ph.D. thesis, Stanford, CA, USA.
- VEACH, E. AND GUIBAS, L. 1994. Bidirectional estimators for light transport. In *Fifth Eurographics Workshop on Rendering*. 147–162.
- WALTER, B., ARBREE, A., BALA, K., AND GREENBERG, D. P. 2006. Multidimensional lightcuts. *Transactions on Graphics (Proceedings of SIGGRAPH 2006)* 25, 3.
- WALTER, B., ZHAO, S., HOLZSCHUCH, N., AND BALA, K. 2009. Single scattering in refractive media with triangle mesh boundaries. *Transactions on Graphics (Proceedings of SIGGRAPH 2009)* 28, 3 (Aug.).
- WANG, Z., BOVIK, A. C., SHEIKH, H. R., AND SIMONCELLI, E. P. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE transactions on image processing* 13, 4, 600–612.
- WATT, M. 1990. Light-water interaction using backward beam tracing. In *SIGGRAPH*. ACM, New York, USA, 377–385.
- ZINKE, A. AND WEBER, A. 2006. Efficient ray based global illumination using photon maps. In *Vision, Modeling, and Visualization 2006 (VMV 2006)*. Akademische Verlagsgesellschaft Aka GmbH, Berlin, 113–120.

Received November 2009; revised July 2010; accepted July 2010