Volume 0 (1981), Number 0 pp. 1–9 COMPUTER GRAPHICS *forum*

Bidirectional Rendering of Vector Light Transport

Adrian Jarabo Victor Arellano

I3A, Universidad de Zaragoza, Spain

Abstract

On the foundations of many rendering algorithm is the symmetry between the path traversed by light, and its adjoint path starting from the camera. However, several effects, including polarization or fluorescence, break that symmetry, and are defined only on the direction of light. This reduces the applicability of bidirectional methods, that exploit this symmetry for simulating effectively light transport. In this work we focus on how to include these non-symmetric effects within a bidirectional rendering algorithm. We generalize the path integral to support the constraints imposed by non-symmetric light transport. Based on this theoretical framework, we propose modifications on two bidirectional methods, namely bidirectional path tracing and photon mapping, extending them to support polarization and fluorescence, in both steady and transient state.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

For many years simulating light transport has exploited the symmetry of most common scattering operators for building efficient methods for rendering. This symmetry involves that, independently on whether the scattering is computed from the propagation direction of the light, or from its adjoint (i.e. the direction from the camera), the throughput of the light transport would be exactly the same [Vea97].

Starting from ray tracing (as opposed to light tracing), these methods have taken advantage of this symmetry to compute only paths that contribute to the image. Bidirectional methods have gone a step further, seamlessly combining paths from the camera and the light, either by connecting vertices of both subpaths [LW93, VG94], merging them via density estimation [Jen01], or a combination of both approaches [GKDS12, HPJ12], for robustly and efficiently handling most common light transport configurations.

Unfortunately, not all scattering operators can benefit from this symmetry. Effects such as polarization and fluorescence are defined with respect to the incoming radiance. Therefore their adjoint (importance) cannot be modeled symmetrically, or even cannot be modeled at all, given its dependence on the incoming illumination. While in most common scenes these effects are negligible, there are many examples where they might play a crucial role on the final appearance: Rendering birefringent crystals [WW08, LSG12], interreflections between conductors and dielectrics, phosphorescent materials [Gla95], or scattering on turbid organic media [GSMA08] require modeling these non-symmetrical operators on light transport for accurate, predictive results.

Moreover, most of these effects exhibit a strong effect on the temporal domain. Therefore, while for traditional light transport they might be important, they might become crucial when computing light transport in transient state. Including these effects is however non-trivial in bidirectional methods, since these techniques build upon the symmetry between radiance and importance.

In this work, we focus on developing a non-symmetric, but bidirectional rendering system, supporting effects such as polarization or fluorescence, which are dependent on the incoming illumination and therefore break the implicit symmetry of bidirectional methods. We first formalize this type of light transport by generalizing the well-known path integral formulation [Vea97] into what we call the *vector light transport*. This allows us to discuss the required modifications on bidirectional algorithms formulated within this framework, in particular bidirectional path tracing. We then show how these changes are applicable to photon mapping. Finally, we extend this formulation to transient state, and show how under reasonable assumptions transient rendering can be easily modeled within this theoretical framework.

This paper is an extension of our previous work on bidirectional rendering of polarization [JG16], in which we introduced a vector formulation of the path integral and use it to generalize bidirectional path tracing and photon mapping to support polarization. This work was concurrent with the work of Mojzik et al. [MSWK16], which had similar aim and proposed a similar theory. Here we reformulate the *vector path integral* to support a wider number of effects, which we demonstrate by rendering fluorescence, and define it in transient state following the work of Jarabo et al. [JMM*14].

2. Background & Related Work

Light Transport Simulation One of the core goals of computer graphics is to synthesize photorealistic images, by accurately simulating how light interacts with matter. While several methods have been introduced to that aim, bidirectional techniques have stood out as the most robust techniques. These methods compute light transport by tracing random walks from the light and camera, which are later connected by means of deterministic connection [LW93, VG94], density estimation [Jen01, JNSJ11], or combinations of both [HPJ12, GKDS12, KGH*14]. These works are formulated to render scalar radiance, and do not support effects such as polarization or elastic scattering.

Polarization Polarized light can be represented using the so-called *Stokes vectors*. These were introduced in graphics by Sankararayanan [San97], and were adopted by other authors for efficient and practical polarization rendering [FGH99, WTP01, WW12]. Stokes vectors model polarized light (for each wavelength λ) as a 4-vector $S_\lambda = \langle S_0, S_1, S_2, S_3 \rangle$ defined as [WTP01]:

$$\begin{aligned} S_0 &= I_s^2 + I_p^2 \\ S_1 &= I_s^2 - I_p^2 \\ S_2 &= 2 I_s^2 I_p^2 \cos(\phi_s - \phi_p) \\ S_3 &= 2 I_s^2 I_p^2 \sin(\phi_s - \phi_p), \end{aligned} \tag{1}$$

where I_s and I_p are the intensity at the parallel and perpendicular planes of the electromagnetic wave, and ϕ_s and ϕ_p are their respective phases. Each component of S_λ represents a type of polarization: S_0 describe the total intensity of the light wave. The second and third component S_1 and S_2 model the linear polarization at zero and 45 degrees, respectively. Finally, S_3 represents the circular polarization. The components in the Stokes vector must hold that $S_0 \geq \sqrt{S_1^2 + S_2^2 + S_3^2}$, imposing that $S_j \in [-S_0, S_0]$ for $j = 1..3$. Note that these components are defined on a particular reference frame aligned with the propagation direction. This formulation is compact and allows to represent all polarization states, and explicitly encodes the intensity of the light wave typically used in rendering, which allows an easier integration into current rendering systems. It is important to note that since Stokes vectors represent polarized light, they are only additive under the condition of lying on the same reference frame. As we will see later (Section 3), this has an important implication when integrating polarized light in the pixel.

The interaction between polarized light and matter is modeled with a matrix structure called *Müller matrix*, which encodes the effect of the BRDF as a 4×4 matrix, representing the linear transformation occurring to the polarized light. The Müller matrix is defined in a particular incoming and outgoing reference frame, which needs to be aligned with the respective light frames. This matrix form is defined for light paths, and therefore not for its adjoint (importance, more common in rendering).

Previous work has solved this by implementing Müller calculus on single-directional methods, such as recursive path tracing, or light tracing, where it is relatively easy to incorporate each of them [WW12], since there are not multiple cases to consider. Here we focus on extending the applicability of fully bidirectional methods for polarization-based rendering. In the following section we go deeper on that.

Polarization is important in a number of effects, including Fresnel-based specular reflection, glowing specular surfaces [WW11] such as reflecting (i.e. no black bodies) incandescent objects, microfacet-based metal [BWWM12] and layered dielectrics surfaces [WWHN17], light transport in the atmosphere [WZP04] and rainbows [SML*12], or uniaxial [GS04, Hac07, WW08] and biaxial [LSG12, DK13] refringent crystals and gems. Additional details on polarized light transport can be found in optics literature [BW02] or in the excellent tutorial by Wilkie and Weidlich [WW12].

Fluorescence & phosphorescence Polarization is not the only effect that breaks symmetry on light transport: quantum effects such as fluorescence and phosphorescence are also defined as a function of the incoming light, and therefore they cannot be modeled based on their adjoint. These phenomena are the result of a change of its quantum state due to absorption of light. This change is not permanent, and after some time it returns to its initial state, resulting into reemission of light, in general with lower energy (i.e. light is reemitted red-shifted). In particular Glassner [Gla95], proposed a model for these two effects, based on re-radiation functions. Gutierrez et al. [GSMA08] focused on fluorescence, including its effect as part of the Russian roulette-based termination on photon mapping and a re-radiation term on the density estimation pass. Hullin and colleages [HHA*10] presented a method to capture bispectral re-radiation matrices, which can be used for rendering fluorescent materials. Recently, Nalbach et al. [NSR17] introduced a pipeline for acquiring and reproducing phosphorescent materials, including their temporal response. Finally, closer to us, Wilkie et al. [WTP01] used bi-spectral re-radiation BRDFs within a forward path tracing, but do not describe how to extend this work to bidirectional methods.

Transient Rendering The emergence of transient imaging [JMMG17] have brought an increased interest on simulating transient light transport in graphics and vision [Jar12, PBSC14, ADY*17]. Jarabo et al. [JMM*14] presented a path integral-based framework for rendering transient light transport. Ament et al. [ABW14] demonstrated time-resolved light transport based on the refractive radiative transfer equation. Recently, Marco et al. [MJGJ17] proposed an extension of photon beams for transient rendering in participating media. We show time-resolved light transport in the context of the vector path integral, and show how transient rendering can be modeled as a vector-matrix operation.

3. Vector Path Integral

Here we describe a generalization of the path integral for including polarization. We use the term *vector* as an analogy of the *vector radiative transfer equation* used to model polarized radiative transfer [Cha60]. The path integral [Vea97] is a theoretic framework where the pixel intensity I is computed as an integral over the

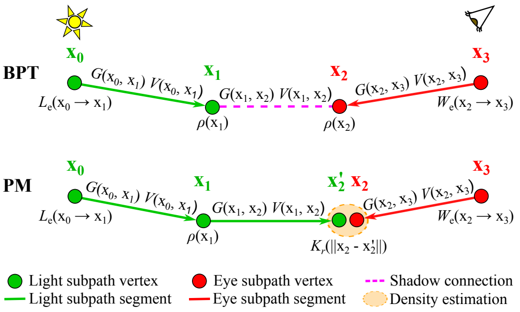

Figure 1: Schematic description of bidirectional path tracing (BPT, top) and photon mapping (PM, bottom). In both algorithms, a sensor and a light subpath are traced from the eye and the light source respectively; these two subpaths are then connected to form a full path, via deterministic shadow connection in the case of bidirectional path tracing, and via an additional random segment and density estimation in photon mapping (Figure after Georgiev et al. [GKDS12]).

space of light transport paths Ω contributing in the pixel:

$$I = \int_{\Omega} f(\overline{\mathbf{x}})\, \mathrm{d}\mu(\overline{\mathbf{x}}), \tag{2}$$

where $\overline{\mathbf{x}} = \mathbf{x}_0 \ldots \mathbf{x}_k$ are the $k+1$ vertices of a length-k path with $k \geq 1$ segments. Vertices $\mathbf{x}_0$ and $\mathbf{x}_k$ lie on a light source and camera sensor respectively, while $\mathbf{x}_1 \ldots \mathbf{x}_{k-1}$ are intermediate scattering vertices. The differential measure $\mathrm{d}\mu(\overline{\mathbf{x}})$ denotes area integration. The path contribution function $f(\overline{\mathbf{x}})$ is the product of the emitted radiance L_e, path throughput $\mathfrak{T}$, and sensor importance W_e:

$$f(\overline{\mathbf{x}}) = L_\mathrm{e}(\mathbf{x}_0 \to \mathbf{x}_1)\, \mathfrak{T}(\overline{\mathbf{x}})\, W_\mathrm{e}(\mathbf{x}_{k-1} \to \mathbf{x}_k). \tag{3}$$

The path throughput is itself the product of the scattering function ρ for the inner path vertices and the geometry G and visibility V terms for path segments:

$$\mathfrak{T}(\overline{\mathbf{x}}) = \left[\prod_{i=1}^{k-1} \rho(\mathbf{x}_i)\right] \left[\prod_{i=0}^{k-1} G(\mathbf{x}_i, \mathbf{x}_{i+1}) V(\mathbf{x}_i, \mathbf{x}_{i+1})\right]. \tag{4}$$

We assume a fractional visibility to account for transmittance within media, as well as opaque objects. The scattering kernel at each vertex is defined as:

$$\rho(\mathbf{x}_i) = \begin{cases} \rho_\mathrm{s}(\mathbf{x}_{i-1} \to \mathbf{x}_i \to \mathbf{x}_{i+1}) & \mathbf{x}_i \text{ on surface,} \\ \rho_\mathrm{p}(\mathbf{x}_{i-1} \to \mathbf{x}_i \to \mathbf{x}_{i+1}) \sigma_s(\mathbf{x}_i) & \mathbf{x}_i \text{ in medium,} \end{cases} \tag{5}$$

where σ_s is the scattering coefficient in the medium, and ρ_s and ρ_p are the surface BSDF and phase function respectively.

Given that in general there is no analytic solution for Equation (2), Monte Carlo solutions are used to approximate the path integral as:

$$\langle I \rangle = \frac{1}{n} \sum_{j=1}^{n} \frac{f(\overline{\mathbf{x}}_j)}{p(\overline{\mathbf{x}}_j)}, \tag{6}$$

that averages the contribution of n random paths $\overline{\mathbf{x}}_j$, sampled with a probability distribution function (pdf) $p(\overline{\mathbf{x}}_j) = p(\mathbf{x}_0 \ldots \mathbf{x}_k)$ the combined probability density of each path's vertex. The probability density of the path is determined by the sampling technique used to obtain the path: for example, *bidirectional path tracing* (BPT) [LW93, VG94] independently generates a subpath $\overline{\mathbf{x}}_w$ from the eye with pdf $p(\overline{\mathbf{x}}_w)$ and a subpath $\overline{\mathbf{x}}_l$ from the light with pdf $p(\overline{\mathbf{x}}_l)$. These are then (optionally) connected using a shadow ray to build the full path $\overline{\mathbf{x}}$ with pdf $p(\overline{\mathbf{x}}) = p(\overline{\mathbf{x}}_l)p(\overline{\mathbf{x}}_w)$ (see Figure 1, top).

Vector Path Integral The *vector path integral* takes a similar form as Equation (2), with a core difference on the definition of the signal being integrated. While Equation (2) integrates a scalar value (i.e. $I \in \mathbb{R}$), the vector form of the path integral integrates a vector $\mathbf{i}$ as:

$$\mathbf{i} = \int_{\Omega} \mathbf{f}(\overline{\mathbf{x}})\, \mathrm{d}\mu(\overline{\mathbf{x}}), \tag{7}$$

with $\mathbf{i}$ and the vector form of the path contribution function $\mathbf{f}(\overline{\mathbf{x}})$ defined by a multidimensional vector in $\mathbb{R}^N$. As an example $\mathbf{f}(\overline{\mathbf{x}})$, lets consider a polarized, monochrome light. In this case, $\mathbf{f}(\overline{\mathbf{x}})$ would be a Stokes vector defined in $\mathbb{R}^4$.

In its vector-form, the scattering kernel at $\mathbf{x}_i$ is no longer a scalar term $\rho(\mathbf{x}_i) \in \mathbb{R}$, but a high-dimensional matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ modeling the relationship between incoming and outgoing light vectors defined in $\mathbb{R}^N$. Using the same example of monochromatic polarized light, in this case $\mathbf{K}$ would be the Müller matrix, defined in $\mathbb{R}^{4\times 4}$. Working with scattering matrices $\mathbf{K}$ breaks transitivity, and therefore we need to define how the operations are concatenated: As opposed to the traditional path integral [Equation (2)], here the order on which the operations are concatenated is important. For that, let us define the concatenation operation as:

$$\prod_{i=1}^{k-1} \mathbf{K}(\mathbf{x}_i) = \mathbf{K}(\mathbf{x}_{k-1})\, \mathbf{K}(\mathbf{x}_{k-2}) \ldots \mathbf{K}(\mathbf{x}_2)\, \mathbf{K}(\mathbf{x}_1). \tag{8}$$

Note that for simplicity we are using the terms "vector" and "matrix" very loosely here: in both cases they might be high-dimensional tensors, and therefore the product operation in Equation (8) would be a tensor product.

Once we have defined the scattering concatenation operator we can define the throughput of path $\overline{\mathbf{x}}$ in matrix form $\mathfrak{T}_\mathbf{v}(\overline{\mathbf{x}}) \in \mathbb{R}^{N\times N}$ as:

$$\mathfrak{T}_\mathbf{v}(\overline{\mathbf{x}}) = \left[\prod_{i=1}^{k-1} \mathbf{K}(\mathbf{x}_i)\right] \left[\prod_{i=0}^{k-1} G(\mathbf{x}_i, \mathbf{x}_{i+1}) V(\mathbf{x}_i, \mathbf{x}_{i+1})\right], \tag{9}$$

where the result of the second product is a scalar, and therefore does not require to be included in the vector form of the scattering kernels.

Finally, by applying Equation (9) to Equation (3) and defining the vector form of the emission $\mathbf{l}_\mathrm{e}(\mathbf{x}_0 \to \mathbf{x}_1) \in \mathbb{R}^N$, we get the vector form of $\mathbf{f}(\overline{\mathbf{x}}) \in \mathbb{R}^N$ as:

$$\mathbf{f}(\overline{\mathbf{x}}) = \mathbf{W}_\mathrm{e}(\mathbf{x}_{k-1} \to \mathbf{x}_k)\, \mathfrak{T}_\mathbf{v}(\overline{\mathbf{x}})\, \mathbf{l}_\mathrm{e}(\mathbf{x}_0 \to \mathbf{x}_1), \tag{10}$$

where $\mathbf{W}_\mathrm{e}(\mathbf{x}_{k-1} \to \mathbf{x}_k) \in \mathbb{R}^{N\times N}$ is the matrix defining the sensor's importance. Similar to Equation (2), the vector version of the path integral defined in Equation (7) can be computed using a Monte Carlo estimator, with the difference of estimating a vector value $\langle \mathbf{i} \rangle \in \mathbb{R}^N$.

Transient Vector Path Integral Similarly to the traditional path integral [Equation (2)], we can extend Equation (7) to support time-resolved light transport. Following the work of Jarabo et al. [JMM*14], we generalize Equations (7), (10) and (9) to transient state as:

$$\mathbf{i} = \int_{\Omega} \int_{\Delta T} \mathbf{f}(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}})\, \mathrm{d}\mu(\overline{\Delta \mathbf{t}})\, \mathrm{d}\mu(\bar{\mathbf{x}}), \tag{11}$$

$$\mathbf{f}(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}}) = \mathbf{W}_{\mathrm{e}}(\mathbf{x}_{k-1} \to \mathbf{x}_k, \Delta t_k)\, \mathfrak{T}_{\mathbf{v}}(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}})\, \mathbf{l}_{\mathrm{e}}(\mathbf{x}_0 \to \mathbf{x}_1, \Delta t_0), \tag{12}$$

$$\mathfrak{T}_{\mathbf{v}}(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}}) = \left[\prod_{i=1}^{k-1} \mathbf{K}(\mathbf{x}_i, \Delta t_i) \right] \left[\prod_{i=0}^{k-1} G(\mathbf{x}_i, \mathbf{x}_{i+1}) V(\mathbf{x}_i, \mathbf{x}_{i+1}) \right], \tag{13}$$

where $\overline{\Delta \mathbf{t}} = \Delta t_0 \ldots \Delta t_k$ is the sequence of time delays of path $\bar{\mathbf{x}}$, and $\mathrm{d}\mu(\overline{\Delta \mathbf{t}})$ is the integration on the temporal domain at each path vertex. Finally, the total temporal delay is

$$t_i^- = \sum_{j=0}^{i-1} \left(t(\mathbf{x}_{\mathbf{x}_j} \leftrightarrow \mathbf{x}_{\mathbf{x}_{j+1}}) + \Delta t_j \right). \tag{14}$$

3.1. Defining vector and scattering matrices

Polarization Our original formulation of the vector path integral [JG16] was defined to support polarization. In this case, assuming a single wavelength, light is represented using a Stokes vector defined in $\mathbb{R}^4$. The scattering interactions $\mathbf{K}$, on the other hand, are modeled by a Müller matrix $\mathbf{M}(\mathbf{x}_i) \in \mathbb{R}^{4\times 4}$. Note that Müller matrices need to be defined in valid reference systems. Therefore, we need to rotate the frames of the incoming and outgoing electromagnetic waves to match the frame on which $\mathbf{M}(\mathbf{x}_i)$ is defined, such that the perpendicular plane of both frames lay in the plane defined by the incoming and outgoing directions. Therefore, the scattering kernel becomes $\mathbf{K} = \mathbf{R}(-\alpha_o)\mathbf{M}(\mathbf{x}_i)\mathbf{R}(\alpha_i)$, where $\mathbf{R}(\alpha)$ is the rotation matrix defined by an angle α defining the rotation along the ray direction, and α_i and α_o are the rotation angles for the incoming and outgoing frames respectively.

Fluorescence and phosphorescence Both fluorescence and phosphorescence involve elastic scattering, which means that light incoming with a given wavelength λ can be reflected with different color. More precisely, for an incoming light defined on the continuous spectrum $L_{\mathrm{i}}(\lambda)$, the outgoing light $L_{\mathrm{o}}(\lambda)$ is

$$L_{\mathrm{o}}(\lambda) = \int_0^{\infty} L_{\mathrm{o}}(\lambda')\rho(\lambda' \to \lambda)\, \mathrm{d}\lambda', \tag{15}$$

where ρ is the bispectral and bidirectional scattering function (note that we omit the directional dependence for simplicity). To model this effect, we therefore need to leave the monochromatic assumption. By discretizing Equation (15), we can pose $L_{\mathrm{o}}(\lambda)$ and $L_{\mathrm{i}}(\lambda)$ as vectors $\mathbf{l}_{\mathrm{o}} \in \mathbb{R}^L$ and $\mathbf{l}_{\mathrm{i}} \in \mathbb{R}^L$ respectively, with L the number of discretizations of the continuum $\lambda \in \mathbb{R}$. Following the same discretization, the bispectral scattering operator ρ is also modeled as the matrix $\mathbf{K} \in \mathbb{R}^{L\times L}$, which relates incoming and outgoing wavelengths λ. The diagonal of $\mathbf{K}$ models the common inelastic scattering (i.e. no energy transfer between wavelengths), while off-diagonal terms model elastic scattering. Note that Equation (15) models local transport in a single vertex of the path. However, it is easy to see that we can concatenate scattering operators $\mathbf{K}$ following Equation (8), resulting on a path throughput $\mathfrak{T}_{\mathbf{v}}(\bar{\mathbf{x}}) \in \mathbb{R}^{L\times L}$, and a vector integral $\mathbf{i}(\lambda) \in \mathbb{R}^L$ defined as a function of λ.

Time-resolved light transport Equation (11) models theoretically the integration in transient state for vector light transport. However, the vector form defined by Equation (7) can also be used for transient rendering, eliminating the need of sampling $\mathrm{d}\mu(\overline{\Delta \mathbf{t}})$. The most important operations in the temporal domain in this context are the delays due to light propagation and scattering. Assuming that we can define the incoming and outgoing light as continuous function in the temporal domain $L_{\mathrm{i}}(t)$ and $L_{\mathrm{o}}(t)$ respectively, both operations can be modeled as convolutions on the temporal domain, as:

$$L_{\mathrm{o}}(t) = \int_0^t T(t') L_{\mathrm{i}}(t - t')\, \mathrm{d}t', \tag{16}$$

where T is the convolution kernel, which for propagation is a delta function centered at t_i^- [Equation (14)], and for scattering is the temporal smoothing function due to e.g. microscopic multiple scattering [HHdD16]. Similar to fluorescence or phosphorescence, we can discretize $L_{\mathrm{o}}(t)$ and $L_{\mathrm{i}}(t)$ in the temporal domain, as $\mathbf{l}_{\mathrm{o}} \in \mathbb{R}^T$ and $\mathbf{l}_{\mathrm{i}} \in \mathbb{R}^T$, with T the number of frames. With this formulation, we can then just easily compute both propagation and scattering delays as scattering matrices $\mathbf{K} \in \mathbb{R}^{T\times T}$, characterizing the outgoing temporal profile due to scattering and propagation of an incoming temporal profile. For example, a propagation delay of t frames in an input temporal profile would be modeled by a transport matrix $\mathbf{K}$ defined as the identity matrix shifted to the right t frames. Similar to Equation (15), Equation (16) models the effect locally, but it can be similarly integrated in Equation (8), resulting on a vector integral $\mathbf{i}(\lambda) \in \mathbb{R}^T$. This means that we keep track of the full temporal profile for each path contribution. Interestingly, this novel formulation allows for an easy integration with other light transport effects, without the need of temporal sampling the scattering delays. Note that, as opposed to the previous phenomena, the propagation delays are bidirectional. This symmetry can be broken in the scattering delays.

Combining different domains In order to combine different domains, such as wavelength and Stokes parameters, we only need to elevate the dimensionality of both the vector light and the matricial operators, while taking into account the operation order of each dimensions on the high-dimensional tensors. For example, for colored polarized light our vector integral $\mathbf{i}$ would have dimensionality $\mathbf{i} \in \mathbb{R}^{4\times L}$, resulting in a Stokes vector for each color sample. In order to handle both polarization and reradiation we would need four-dimensional scattering *tensors* $\mathbf{K} \in \mathbb{R}^{4^2\times L^2}$ (i.e. each component of the Müller matrix would be a reradiation matrix itself). This results into high-dimensional throughput tensors $\mathfrak{T}_{\mathbf{v}}(\bar{\mathbf{x}}) \in \mathbb{R}^{4^2\times L^2}$, and might result into costly computations. However, as we discussed later, the cost can be severely alleviated by doing the expensive tensor computations only when strictly required.

4. Bidirectional Rendering of Vector Light Transport

In Section 3 we have described the mathematical framework within which we will work, making explicit the differences between traditional scalar rendering and the novel vector formulation. Here we describe the algorithmic and implementation details for developing a bidirectional rendering within this framework.

Bidirectional methods [VG94, Jen01, GKDS12, HPJ12] compute the path integral by sampling several light paths joining the light and the sensor. This is done by generating two different random walks (*subpaths*), each starting from the initial and final vertices of the path. These are then joined by means of a deterministic shadow connection, creating a full contributing path. These two random walks have different probabilities, depending on the sampling strategy used to create the subpath.

When extending these methods to our framework, we detect that the random walk from the light source (the light subpath $\bar{\mathbf{x}}_l$) fits naturally in the vector path integral , since it follows the sequence of events occurring to light since it is emitted. For each new scattering event in point $\mathbf{x}_i$, we would compute its scattering kernel $\mathbf{K}(\mathbf{x}_i)$, and apply it to the accumulated throughput of the path following Equation (8). Generating the random walk from the sensor, as well as performing the shadow connection, is a bit trickier. The key difference between them is that, as discussed by Mojzik et al. [MSWK16], while the light subpath is defined by a light vector, the sensor subpath is defined by an importance matrix. In the following, we explain them on more details.

Sensor Subpath When computing the sensor subpath $\bar{\mathbf{x}}_w$, we need to take into account that we are starting the sequence of events on the reverse order. Thus we are not tracking vector magnitudes, but matricial operators. This is key, since it affects on how the scattering kernels at each vertex of the subpath are defined. The main difficulty is to keep track on whether the incoming or outgoing directions follow the light direction (ω_i and ω_o respectively), or its adjoint (which we denote $\hat{\omega}_\mathrm{i}$ and $\hat{\omega}_\mathrm{o}$). For each new scattering event in the random walk, we sample the subpath new direction $\hat{\omega}_\mathrm{o}$ based on the previous direction $\hat{\omega}_\mathrm{i}$. We use the same sampling routine for light and importance tracing, based on intensity in the case of polarization, and in the diagonal of the scattering matrix in the case of fluorescence. The reason is that it allows to sample intensity in the former case, and that reemision is practically Lambertian [HHA*10], so most of the directionality is kept in the diagonal of the reemision matrix. Then, we create the scattering kernel $\mathbf{K}(\mathbf{x}_i)$ in the frame defined by the light incoming and outgoing directions, $\omega_\mathrm{i} = -\hat{\omega}_\mathrm{o}$ and $\omega_\mathrm{o} = -\hat{\omega}_\mathrm{i}$ respectively.

With that in place, and taking into account that for the sensor's subpath we decrement the indices of the subpath vertices (i.e. the subpath vertices are generated in the reverse order, starting by vertex k to vertex 0), we then compute the subpath throughput using Equation (8). Therefore, as opposed to multiplying $\mathbf{K}(\mathbf{x}_i)$ to the left to the accumulated throughput as in the light random walk, we need to apply each new scattering kernel on the right.

Shadow Connection In order to join the light and sensor subpaths, we again need to be careful on the reference frame of the scattering operator, and on the order at which the events are computed. In this case, for a light subpath $\bar{\mathbf{x}}_l$ with length m and partial throughput $\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_l)$, and sensor subpath $\bar{\mathbf{x}}_w$ with length $k-m$ and partial throughput $\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_w)$, we obtain the final throughput as:

$$\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}) = \mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_w)\mathbf{K}(\mathbf{x}_{k-m})G(\mathbf{x}_{k-m},\mathbf{x}_m)V(\mathbf{x}_{k-m},\mathbf{x}_m)\mathbf{K}(\mathbf{x}_m)\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_l). \tag{17}$$

As discussed earlier, note that $\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_w) \in \mathbb{R}^{N\times N}$ defines matricial importance, while $\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_l) \in \mathbb{R}^{N}$ represents vectorial light.

4.1. Photon Mapping

As shown by Georgiev et al. [GKDS12] and Hachisuka et al. [HPJ12], photon mapping (PM) [Jen01] can be understood as a variant of bidirectional path tracing, which differs from the standard formulation on how the sensor and light subpaths are connected. While in bidirectional path tracing we connect the last two vertices by means of a deterministic shadow connection, in photon mapping we merge the last two vertices by using a density estimation kernel. While this introduces bias, it has been shown that in the limit the algorithm is consistent (i.e. converges to the correct solution, we refer to a recent course [HJG*13] for details).

This means that we can define PM under the path integral framework, with some small modifications. Therefore, we can introduce photon mapping in our vector formulation of the path integral. In fact, the main difference with respect to BDPT is how the eye and light subpaths are joint. While for bidirectional path tracing we make use of Equation (17), here we define the throughput of the path resulting from merging the sensor $\bar{\mathbf{x}}_w$ and light $\bar{\mathbf{x}}_l$ subpaths as:

$$\mathfrak{T}'_\mathbf{v}(\bar{\mathbf{x}}) = \mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_w)\mathcal{K}_R(\|\mathbf{x}_{k-m} - \mathbf{x}_m\|)\mathbf{K}(\mathbf{x}_{k-m})\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_l), \tag{18}$$

where $\mathcal{K}_R$ is the spatial smoothing kernel with bandwidth R. Note that given that we are *merging* vertices $\mathbf{x}_{k-m}$ and $\mathbf{x}_m$ (see Figure 1) we only have to apply one scattering kernel $\mathbf{K}(\mathbf{x}_{k-m})$. Additionally, note that the scattering kernel $\mathbf{K}(\mathbf{x}_{k-m})$ is defined with incoming direction the one from the light subpath (the incoming direction of $\mathbf{x}_m$), while the outgoing direction is the inverse of the (virtual) incoming direction for the sensor subpath's last vertex $\mathbf{x}_{k-m}$.

4.2. Implementation

We implemented our vector-based rendering on top of an in-house physically-based renderer written in C++. In many cases, working with the full scattering matrices $\mathbf{K}$ was not needed, given that e.g. light was unpolarized, the scattering kernel itself was a depolarizer, or the scattering event was inelastic. We add a flag to both the Stokes vectors, Müller matrices, and reradiation matrices to discard computations depending on the type of light and interaction being computed. This significantly increases performance while not affecting the accuracy of results.

While our renderer supports spectral rendering, our tests have been performed using RGB (i.e. $L = 3$). In case of using a large number of wavelengths the costs of the multiplication of scattering kernels might be prohibitive. However, even considering their small size ($3 \times 3 \times 4 \times 4$) scattering matrices supporting polarization and fluorescence are relatively sparse for most common operations, which can be exploited to increase performance. Note that preintegrating the reradiation matrix into RGB is accurate for single reflection, but might lead to problems in full global illumination solutions, as discussed by Meng et al. [MSHD15].

Finally, we have included five types of scattering kernels: a Lambertian BRDF acting as a depolarizer [WW12], a smooth conductor BRDF with complex index of refraction modeled with the Fresnel equations for conductors [WW12], a Fresnel smooth dielectric BSDF with support to both transmission and reflection [Azz04], a fluorescent material based on measured data for chlorophyll [GSMA08], and a weakly polarizing Mie phase function obtained using MiePlot [Lav15].

5. Results

We demonstrate our implementation by rendering a set of scenes with complex light interactions, including dielectric, conductors and participating media. The selected results feature polarization, fluorescence, and time-resolved light transport.

Figures 2, 3, and 4 show different scenes showing polarization. These scenes include several diffuse surfaces, as well as both conductor and dielectric mirrors, dielectric transparent objects, and participating media with different degrees of scattering. For visualizing the polarization, we use the techniques described by Wilkie and Weidlich [WW10], depicting the degree of polarization, the degree of circular polarization, and orientation of linear polarization. In both cases, this visualization is super-imposed to the luminance image: brighter red means higher degree of polarization, while color codes the orientation of linear polarization. We can observe how conductors are bad polarizers, and polarize light only slightly. However, dielectrics are very good polarizers. Moreover, in Figure 2 we can see how linear polarization switches to circular polarization when the frames of reference of reflection are disaligned (e.g. in poles of the dielectric sphere or in the top of the planar mirror). In addition, we can see how scattering media tends to slightly polarize light transport (Figure 3), although the main polarizer in that scene are still dielectrics.

Figure 5 shows the effect of introducing fluorescence, where a chlorophyll bunny is illuminated by white light. The direct reflection of the bunny is mostly green, while red and blue are mainly absorbed. However, most of the absorbed blue is later reemited in the red band, which in the end changes the appearance of the bunny. This effect can be seen more clearly in transient state: Figure 6 shows several frames of a time-resolved render of the same scene. Here we can see the temporal behavior of fluorescence, where light that is absorbed by the bunny is later reemitted red-shifted. The full animation can be seen in the accompanying supplemental video.

Given that the illumination comes from a point light source, and the caustic paths due to smooth dielectric and conductors are dominant, we use a stochastic progressive [HJ09, KZ11] version of our vector photon mapping (Section 4.1) in scenes Figures 2, 3, and 4, although we also account for multiple bounces on the sensor subpath, and perform deterministic shadow connections with the light. We computed 1000 iterations, with 16 samples per pixel and 5M photon random walks on each iteration. We compute the initial kernel radius using the 20 nearest photons. Note that without bidirectional methods such as photon mapping these scenes, where caustic light paths dominate, could be very hard, or even impossible, to render. For the *bunny* scene (Figures 5 and 6), we use a vector-based implementation of bidirectional path tracing, given that there is less caustic transport, with 25K samples per pixel for rendering the full temporal profile.

In terms of performance, adding vector-based effects such as polarization or fluorescence increases the cost with respect to scalar rendering (Table 1), due to the vector-to-matrix and the matrix-to-matrix products. However, we avoid these costly computations unless they are strictly needed, although with multiple scattering these operations are common. The need of tracking the reference frame in the case of polarization also introduces additional costs. Note, however, that the vector and matrix operations are not optimized; using vectorized code could reduce the costs significantly. Additionally, while vector representation of light and scattering is relatively compact (or sparse), the memory cost is much higher than in traditional rendering: For example, for polarized photons we need to store four floats for a single wavelength, in contrast to the single float per wavelength in scalar rendering, plus the need of storing the full frame, not only the photon's direction. Finally, in Figure 7 we analyze the convergence with respect to the number of samples for Figures 3 and 4 (polarized light transport), compared against scalar rendering: Given the additional dimensions introduced by rendering Stokes vectors, the error is slightly higher than scalar rendering.

Figure	Support	Technique	Scalar	Vector
2*	Polarization	Photon Mapping	8	14
3	Polarization	Photon Mapping	0.72	0.83
4	Polarization	Photon Mapping	0.79	0.79
5	Transient Fluorescence	BDPT	-	1.27

Table 1: Comparison of the average cost (in seconds) per iteration between scalar and vector rendering, for the render examples shown in the paper. Note that Figure 2 was rendered in a lower-end workstation.

6. Conclusions & Future Work

In this work we have generalized the path integral formulation, and its transient counterpart, to support vector-based light transport. This is needed to support effects such as polarization or fluorescence, and imposes a set of constraints on the scattering operators and their concatenation. Interestingly, these modifications do not reduce generality, but extends its range of applicability to transport operators breaking symmetry. Based on this theoretical framework, we have described the required changes to well-known (scalar) bidirectional rendering methods defined within Veach's [Vea97], and the unifed path integral formulation [GKDS12, HPJ12]. In particular, we have shown how to include vector light transport in both bidirectional path tracing and photon mapping.

We have shown that this form of representing stochastic light transport might be powerful for representing effects beyond traditional vector-based light (i.e. polarization or spectral rendering): In particular, we have shown that transient light transport can also be modeled in a vector-matrix representation. We believe this is an interesting approach for rendering the full plenoptic function.

There is of course several future work ahead. First of all, we have proposed just a few areas of application to the new form of vector light transport. Modeling other non-symmetric effects in a vector-based representation might increase the range of effects that are representable in current render engines. From a practical perspective, developing sampling techniques aware of the particularities of the effect is a promising avenue of future work. Here we have just used standard scalar-based techniques for both the random walk sampling and deterministic connection of light and sensor paths. Finally, proposing new stochastic shadow connections or

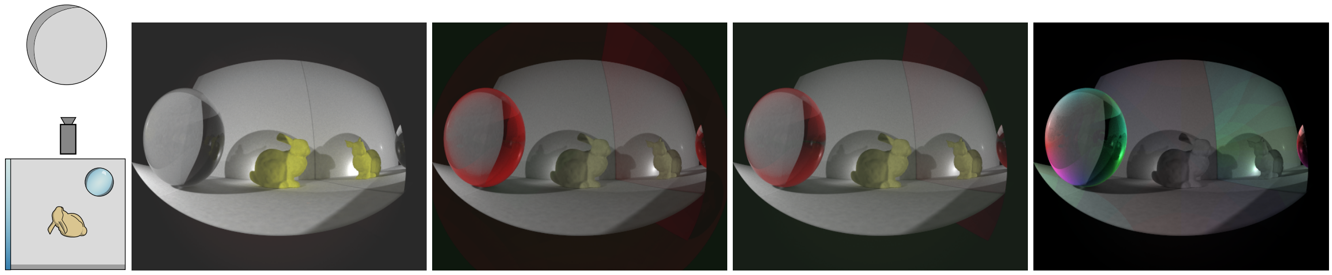

Figure 2: *Mirror* scene, depicting a scene with a dielectric sphere and a conductor mirror reflected on a spherical conductor mirror. From left to right: scene setup, radiance, degree of polarization, degree of circular polarization, and orientation of linear polarization (color codes orientation). The degree of polarization and circular polarization are encoded in red (birghter is higher). While the conductors only polarize light weakly, the dielectric specular reflection is highly polarized. In addition, the concatenation of multiple reflections on curved surfaces shifts the linear polarization towards circular polarization.

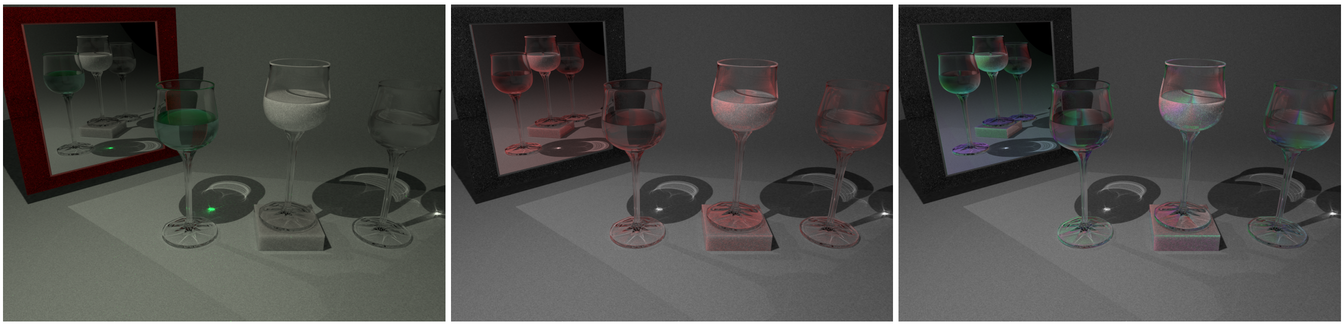

Figure 3: *Cups* scene, depicting a scene with a dielectric mirror, and a set of glass cups filled with absorbing, scattering and a non-participating dielectric media. From left to right: radiance, degree of polarization, and orientation of linear polarization. The scene shows the slightly polarizing effect of very scattering and optically thick media, specially visible in the middle cup.

optimal combinations of sampling techniques, based on e.g. polarization state or the bispectral scattering kernel, would significantly increase the efficiency when rendering such effects.

Acknowledgements

We want to thank Diego Gutierrez and Adolfo Muñoz for discussions about polarized light and insightful comments on the manuscript, Rihui Wu for his feedback on the results, Ibón Guillén for helping with the figures, and the annonymous reviewers and members of the Graphics & Imaging Lab for their constructive suggestions. This research has been partially funded by DARPA (project REVEAL), the European Research Council (Consolidator Grant, project CHAMELEON), and the Spanish Ministerio de Economía y Competitividad (projects TIN2016-78753-P and TIN2014-61696-EXP).

References

[ABW14] AMENT M., BERGMANN C., WEISKOPF D.: Refractive radiative transfer equation. *ACM Trans. Graph. 33*, 2 (2014). 2

[ADY*17] ADAM A., DANN C., YAIR O., MAZOR S., NOWOZIN S.: Bayesian time-of-flight for realtime shape, illumination and albedo. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 5 (2017), 851–864. 2

[Azz04] AZZAM R.: Phase shifts that accompany total internal reflection at a dielectric–dielectric interface. *JOSA A 21*, 8 (2004). 5

[BW02] BORN M., WOLF E.: *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, 2002. 2

[BWWM12] BERGER K., WEIDLICH A., WILKIE A., MAGNOR M.: Modeling and verifying the polarizing reflectance of real-world metallic surfaces. *IEEE Computer Graphics and Applications 32*, 2 (2012), 24–33. 2

[Cha60] CHANDRASEKHAR S.: *Radiative Transfer*. Dover, 1960. 2

[DK13] DEBELOV V. A., KOZLOV D. S.: A local model of light interaction with transparent crystalline media. *IEEE Transactions on Visualization and Computer Graphics 19*, 8 (2013). 2

[FGH99] FRENIERE E. R., GREGORY G. G., HASSLER R. A.: Polarization models for monte carlo ray tracing. In *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation* (1999). 2

[GKDS12] GEORGIEV I., KŘIVÁNEK J., DAVIDOVIČ T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Trans. Graph. 31*, 6 (2012). 1, 2, 3, 4, 5, 6

[Gla95] GLASSNER A. S.: A model for fluorescence and phosphorescence. In *Photorealistic Rendering Techniques*. 1995. 1, 2

[GS04] GUY S., SOLER C.: Graphics gems revisited: fast and physically-based rendering of gemstones. *ACM Trans. Graph. 23*, 3 (2004). 2

[GSMA08] GUTIERREZ D., SERON F., MUÑOZ A., ANSON O.: Visualizing underwater ocean optics. *Computer Graphics Forum 27*, 2 (2008). 1, 2, 6

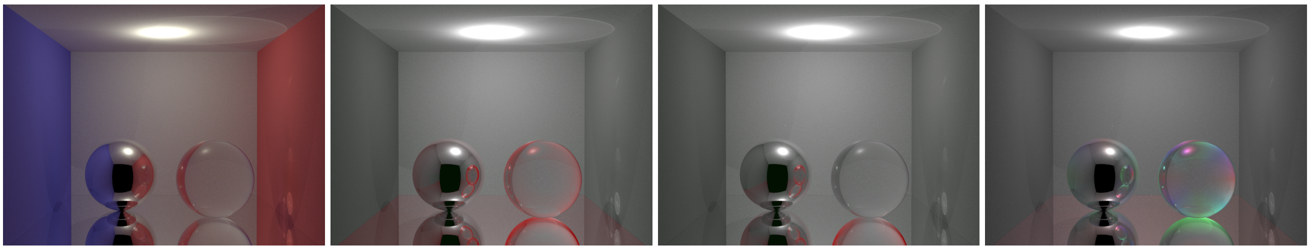

Figure 4: *Cornell* box with a conductor mirror in the floor, and two spheres (one conductor and one dielectric). From left to right: scene setup, radiance, degree of polarization, degree of circular polarization, and orientation of linear polarization. The accumulative effect of polarization, in particular how it switches from linear to circular polarization, is clearly visible in the multiple reflections on both spheres.

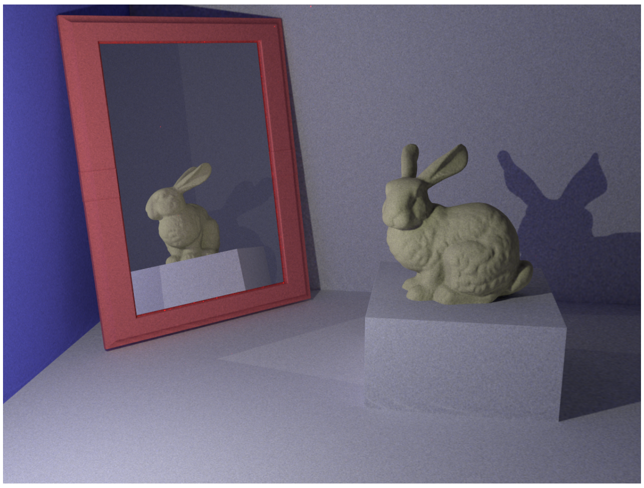

Figure 5: Fluorescent *bunny* made of chlorophyll. Although the direct reflection is bright green, some blue light is absorbed and re-emitted in the red band, resulting a paler yellowish appearance.

[Hac07] HACHISUKA T.: Generalized polarization ray tracing using a Monte Carlo method. CSE 272, University of California at San Diego, 2007. 2

[HHA*10] HULLIN M. B., HANIKA J., AJDIN B., SEIDEL H.-P., KAUTZ J., LENSCH H. P. A.: Acquisition and analysis of bispectral bidirectional reflectance and reradiation distribution functions. *ACM Trans. Graph. 29*, 4 (2010). 2, 5

[HHdD16] HEITZ E., HANIKA J., D'EON E., DACHSBACHER C.: Multiple scattering microfacet BSDFs with the Smith model. *ACM Trans. Graph. 35*, 4 (2016), 58. 4

[HJ09] HACHISUKA T., JENSEN H. W.: Stochastic progressive photon mapping. *ACM Trans. Graph. 28*, 5 (2009). 6

[HJG*13] HACHISUKA T., JAROSZ W., GEORGIEV I., KAPLANYAN A., NOWROUZEZAHRAI D.: State of the art in photon density estimation. In *ACM SIGGRAPH ASIA 2013 Courses* (2013). 5

[HPJ12] HACHISUKA T., PANTALEONI J., JENSEN H. W.: A path space extension for robust light transport simulation. *ACM Trans. Graph. 31*, 6 (2012). 1, 2, 4, 5, 6

[Jar12] JARABO A.: *Femto-photography: Visualizing light in motion*. Master's thesis, Universidad de Zaragoza, 2012. 2

[Jen01] JENSEN H. W.: *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001. 1, 2, 4, 5

[JG16] JARABO A., GUTIERREZ D.: Bidirectional rendering of polarized light transport. In *Proc. CEIG '16* (2016). 1, 4

[JMM*14] JARABO A., MARCO J., MUÑOZ A., BUISAN R., JAROSZ W., GUTIERREZ D.: A framework for transient rendering. *ACM Trans. Graph. 33*, 6 (2014). 1, 2, 4

[JMMG17] JARABO A., MASIA B., MARCO J., GUTIERREZ D.: Recent advances in transient imaging: A computer graphics and vision perspective. *Visual Informatics 1*, 1 (2017). 2

[JNSJ11] JAROSZ W., NOWROUZEZAHRAI D., SADEGHI I., JENSEN H. W.: A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Trans. Graph. 30*, 1 (2011). 2

[KGH*14] KŘIVÁNEK J., GEORGIEV I., HACHISUKA T., VÉVODA P., ŠIK M., NOWROUZEZAHRAI D., JAROSZ W.: Unifying points, beams, and paths in volumetric light transport simulation. *ACM Trans. Graph. 33*, 4 (2014). 2

[KZ11] KNAUS C., ZWICKER M.: Progressive photon mapping: A probabilistic approach. *ACM Trans. Graph. 30*, 3 (2011). 6

[Lav15] LAVEN P.: Mieplot. http://www.philiplaven.com/mieplot.htm, 2015. 6

[LSG12] LATORRE P., SERON F., GUTIERREZ D.: Birefringency: Calculation of refracted ray paths in biaxial crystals. *The Visual Computer 28*, 4 (2012). 1, 2

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Compugraphics '93* (1993). 1, 2, 3

[MJGJ17] MARCO J., JAROSZ W., GUTIERREZ D., JARABO A.: Transient photon beams. In *Proc. CEIG '17* (2017). 2

[MSHD15] MENG J., SIMON F., HANIKA J., DACHSBACHER C.: Physically meaningful rendering using tristimulus colours. *Computer Graphics Forum 34*, 4 (2015), 31–40. 5

[MSWK16] MOJZIK M., SKRIVAN T., WILKIE A., KRIVANEK J.: Bidirectional polarised light transport. In *Proc. EGSR '16* (2016). 1, 5

[NSR17] NALBACH O., SEIDEL H.-P., RITSCHEL T.: Practical capture and reproduction of phosphorescent appearance. *Computer Graphics Forum 36*, 2 (2017). 2

[PBSC14] PITTS P., BENEDETTI A., SLANEY M., CHOU P.: *Time of Flight Tracer*. Tech. rep., Microsoft, 2014. 2

[San97] SANKARANARAYANAN S.: *Modelling polarized light for computer graphics*. PhD thesis, 1997. 2

[SML*12] SADEGHI I., MUÑOZ A., LAVEN P., JAROSZ W., SERON F., GUTIERREZ D., JENSEN H. W.: Physically-based simulation of rainbows. *ACM Trans. Graph. 31*, 1 (2012). 2

[Vea97] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford, 1997. 1, 2, 6

[VG94] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Proc. of EGRW '94* (1994). 1, 2, 3, 4

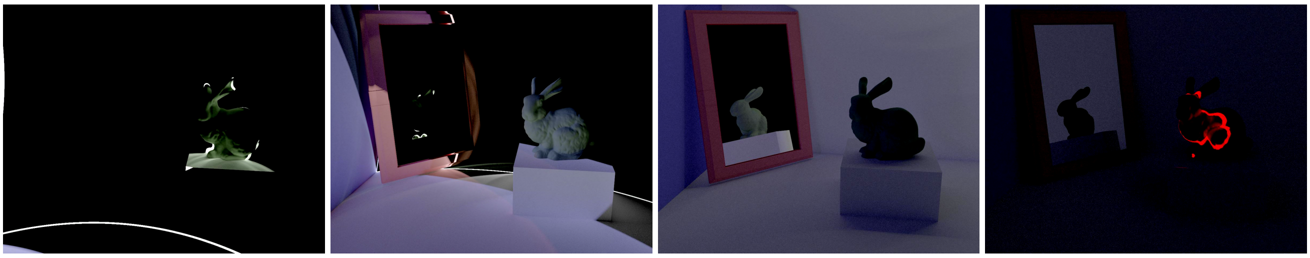

Figure 6: Time-resolved render for the *bunny* scene: From left to right, light travels from the light source to the bunny, which reflects most green light and absorbs red and blue light. Note that this reflection before reradiation would be the result if a scalar version of the algorithm were used. Light continues propagating, illuminating the scene via multiple reflections, while the mirror starts reflecting the bunny. After several diffuse reflections, light loses its directionality, becoming mainly diffuse lighting. Finally, after a few nano-seconds, absorbed light is reemited by the fluorescent bunny red-shifted. We refer to the supplemental video for the full animation.

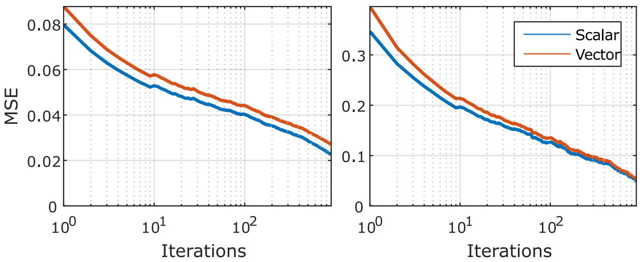

Figure 7: Convergence plots for our vector light transport (polarization) compared against traditional scalar rendering, for Figure 3 (left) and Figure 4 (right).

[WTP01] WILKIE A., TOBLER R. F., PURGATHOFER W.: Combined rendering of polarization and fluorescence effects. In *Proc. EGSR '04* (2001). 2

[WW08] WEIDLICH A., WILKIE A.: Realistic rendering of birefringency in uniaxial crystals. *ACM Trans. Graph. 27*, 1 (2008). 1, 2

[WW10] WILKIE A., WEIDLICH A.: A standardised polarisation visualisation for images. In *Proc. SCCG '10* (2010). 6

[WW11] WILKIE A., WEIDLICH A.: A physically plausible model for light emission from glowing solid objects. *Computer Graphics Forum 30*, 4 (2011). 2

[WW12] WILKIE A., WEIDLICH A.: Polarised light in computer graphics. In *SIGGRAPH Asia 2012 Courses* (2012). 2, 5

[WWHN17] WANG C., WILKIE A., HARCUBA P., NOVOSAD L.: Virtual ellipsometry on layered micro-facet surfaces. *Optics Express 25*, 19 (2017), 22971–22990. 2

[WZP04] WILKIE A., ZOTTI G., PURGATHOFER W.: An analytical model for skylight polarisation. In *Proc. EGSR '04* (2004). 2