CEIG - Spanish Computer Graphics Conference (2016)
Alejandro García-Alonso and Belen Masia (Editors)

Bidirectional Rendering of Polarized Light Transport

Adrian Jarabo and Diego Gutierrez

Universidad de Zaragoza I3A Institute

Abstract

On the foundations of many rendering algorithm is the symmetry between the path traversed by light and its adjoint from the camera. However, several effects, including polarization or fluorescence, break that symmetry and are defined only on the direction of light. This complicates the applicability of bidirectional methods, that exploit the symmetry for effective rendering light transport. In this work we focus on how to include polarization within a bidirectional rendering algorithm. For that, 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.

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

1. Introduction

For many years light transport simulation 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 [GKH*13, HPJ12], for robustly and efficiently handle most common light transport configurations.

Unfortunately, not all light transport operators can benefit from this symmetry. Effects such as polarization and fluorescence are defined with respect to the incoming radiance, and therefore their adjoint (the importance) cannot be modeled symmetrically, or even cannot be modeled at all, given the broken symmetry. While for most common scenes these effects are negligible, there are many examples where they 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 of 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 become crucial when computing light transport in transient-state [JMM*14]. Including these effects is however non-trivial in (steady- and transient-state) bidirectional methods, due to the need of tracking the state of the subpath, and because the operation order matters.

In this work, we focus on developing a non-symmetric, but bidirectional rendering system, in particular focused on polarization. We first formalize non-symmetric polarized light transport by generalizing the well-known path integral formulation [Vea97] to this scenario. This allows us to discuss the needed changes on bidirectional rendering algorithms modeled within this formulation, in particular on bidirectional path tracing, and how these changes are applicable to photon mapping, which leads us to develop polarization-aware variants for both methods.

2. Background & Related Work

In computer graphics we commonly rely on geometric optics for representing light; this allows us to consider light as a bunch of rays transporting energy as a function of its frequency ω and intensity I (amplitude). While this assumption is enough for most applications, this is a severe simplification. It is well known the duality of light as a particle and an electromagnetic wave, where the latter is usually ignored in computer graphics applications. Of course there are some attempts on rendering based on Maxwell equations [Mor81, MMRO13], which include polarization of light, but these works are in general difficult to implement and impractical when trying to render a complex scene.

As all electromagnetic waves, the electric field of light E os-

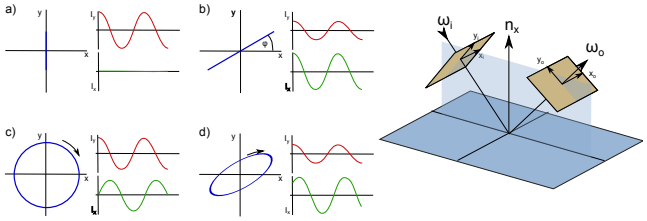

Figure 1: Examples of polarization states (left): (a,b) linear polarized light aligned to 0 and ϕ degrees respectively, (c) circular polarization, and (d) elliptical polarization. The graphs on the sides represent the electric field in the two phasors E_x and E_y, as shown in the geometry of a ray-surface intersection depicted on the right, where the parallel plane $\mathbf{y}$ is the same for both the incoming and outgoing frames. Note that neither elliptical nor circular polarized light rotate physically. (Figure after Shumaker [Shu77] and Wilkie and Weidlich [WW12]).

cillates along a plane parallel to its ray direction $\mathbf{w}$ with a given frequency ω. By choosing a plane perpendicular to $\mathbf{w}$, defined by the orthogonal axis $\mathbf{x}$ and $\mathbf{y}$ (themselves orthogonal to $\mathbf{w}$), we can define the oscillation of E as the combination of two harmonic oscillations on each reference plane $\mathbf{x},\mathbf{y}$:

$$E_{\mathbf{xy}}(r) = \langle I_s \sin(\omega r + \phi_s), I_p \sin(\omega r + \phi_p)\rangle, \quad (1)$$

with r the optical depth traversed by the light, I_s and I_p the wave amplitude, such that $I = I_s^2 + I_p^2$, and ϕ_s and ϕ_p the phase shift of the wave in each plane of reference. Note that we use the subindex s and p for the axis $\mathbf{x}$ and $\mathbf{y}$ respectively, following the typical notation in optics, where s stands for the German terms *senkrecht* (perpendicular) and *parallel*, respectively.

The difference in amplitude and phases between each axis (or phasor, see [BW02]) is responsible of the state of polarization of the light: if $\phi_s - \phi_p = \frac{\pi}{2}n$, with $n \in \mathbb{Z}$, then it is *linearly polarized*, on a plane defined by the amplitude on each phasor, while if $\phi_s - \phi_p = \frac{\pi}{4}n$ and $I_s = I_p$ then light is *circularly polarized*. In all other cases light is *elliptically polarized*. Note that a particular light wavefront can be the superposition of light in several polarization states; if there is no correlation between these states, then the polarization state is considered random, and therefore light gets unpolarized up to certain extend. Figure 1 illustrates different types of polarization.

It is important to realize that in order to sum up polarized light it needs to be defined in the same reference frame. This imposes that we can only sum light with the same propagation direction $\mathbf{w}$, and that the reference frames $\mathbf{x},\mathbf{y}$ of each light bundle need to be aligned (i.e. the perpendicular and parallel planes need to be defined in the same reference frame). This is also important when interacting with matter, given that e.g. the Frensel equations are defined on a fixed reference frame.

Stokes Vectors & Müller Calculus Several models exist for modeling polarization of light: the most straightforward would be to directly track the electromagnetic phasors as light traverses the media. This is very convenient when accounting for a single polarization state in the light beam, and allows modeling complex effects related with the phase occurring in e.g. water drops in rainbows [SML*12] or when modeling thin-layer interference [Hac07]. Additionally, it can be compactly represented as a tuple of complex numbers, known as Jones vector. However, this formulation is pretty inconvenient when dealing with aggregates of randomly polarized light (e.g. partially polarized light or unpolarized light), since we need to track each polarization state.

Coherency Matrices [BW02, Gla95] is another formalism to describe polarized light. In essence encode the same information as Equation (1) in a single matrix, with the advantage of being able to model the polarization state for a mixture of polarization states, as a function of their correlation. Thus, for unpolarized light the correlation will be 0, while for perfectly polarized light this correlation is one. This formulation was used in the first works in computer graphics dealing with polarization [WK90], and its extension for anisotropic crystals [TTW94].

The last form for representing polarization are the so-called *Stokes vectors*, which is the formulation used in this paper. These were introduced in graphics by Sankararayanan [San97], and were adopted by other authors for efficient and practical polarization rendering [FGH99, WTP01, WW12]. They 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]:

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

Intuitively, each component of this model represents a type of polarization: S_0 describe the total intensity of the light wave, which is the typical value used in rendering. The second and third component S_1 and S_2 model the linear polarization at zero and 45 degrees. 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 $j = 1..3$. This formulation has several properties that are interesting for rendering: first, it shares with the coherency matrix that it compactly encodes statistically an aggregate of different polarization states, which allows to represent all possible degrees of polarization. Additionally, it 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.

In order to model the interaction between light modeled using Stokes vectors and matter, a matrix structure called *Müller matrix* is used. It encodes the effect of the BRDF as a 4×4 matrix, representing the linear transformation occurring to the polarized light encoded in S. The form and values of this matrix depends on the type of interactions, and similarly to the Stokes vector it is defined in a particular incoming and outgoing reference frame. Thus, for each interaction we need to fulfill the requirement valid light direction with respect to the Müller 's incoming frame, and that both frames are aligned. Additionally, this matrix form is defined for incoming light, and therefore not for its adjoint (more common in

rendering). This is crucial, since it breaks the symmetry, and therefore imposes a single directionality on the tracer. Previous work has solved this by implementing these effects 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. In this work, we focus on extending the applicability of fully bidirectional methods for polarization-based rendering. In the following section we go deeper on that.

Effects Requiring Polarization As discussed earlier, there are several effects that require polarization for accurate rendering [WW12]. The canonical example is specular reflection: the Fresnel equations, specially for dielectrics, are strongly polarizing. As such, when concatenating the reflection from different specular surfaces polarization might have a subtle but noticeable effect on the look of the reflection, specially when illuminated from polarized area lights such as common flat-screen displays. Unfortunately, these effects are in general restricted to perfectly smooth surfaces, with the notable exception in graphics to the work of He et al. [HTSG91], which is able to account for polarization off glossy surfaces. More recently, a model based on Müller matrices was proposed on the optics community [LMNS12] which, as opposed to He's work, has been validated against measured data.

Glowing specular surfaces [WW11], such as reflecting (i.e. no black bodies) incandescent objects, also emit polarized light, and therefore the appearance of their reflection on specular surfaces (or even with themselves) can suffer from changes when including polarization into the computations. The atmosphere is also affected by polarization, and is in fact the main polarized light source in outdoor scenes: for example, polarization needs to be taken into account when accurately modeling the complex light-matter interactions occurring in waterdrops, which are responsible for e.g. rainbows [SML*12]. In the context of atmospheric light, Wilkie et al. [WZP04] proposed a parametric sky model accounting for polarization, mainly due to Rayleigh scattering.

Finally, some crystals present birefringence: this effect is the result of the internal structure of the crystal, which lead to different index of refraction and optical axis for the two phasors of light (parallel and perpendicular). This in the end produces two different refraction images due to ordinary and extraordinary refractions (note that for bi-axial crystals both refractions are extraordinary). Several works have dealt with this effect: including solutions for uniaxial [GS04, Hac07, WW08] and biaxial [LSG12, DK13] crystals and gems.

Other Non-Symmetric Effects Polarization is not the only effect that breaks simmetry on light transport: quantum effects such as fluorescence and phosphorescence are also defined as a function of the incoming light, and therefore we cannot model them based on their adjoint. 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. Finally, 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.

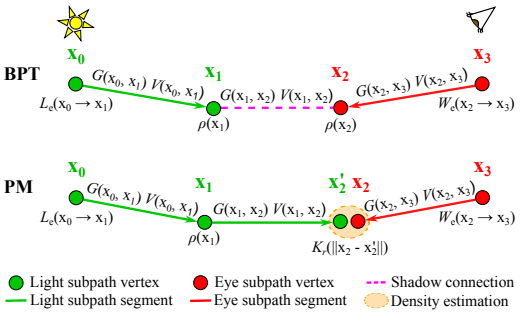

Figure 2: Schematic description of bidirectional path tracing (BPT, top) and photon mapping (PM, bottom). In both algorithms, a sensor and a light subpath are traced from the eye and the light source respectively; these two subpaths are then connected to form a full path, via deterministic shadow connection 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]).

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 space of light transport paths Ω contributing in the pixel:

$$I = \int_{\Omega} f(\bar{\mathbf{x}})\, \mathrm{d}\mu(\bar{\mathbf{x}}), \tag{3}$$

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

$$f(\bar{\mathbf{x}}) = L_e(\mathbf{x}_0 \to \mathbf{x}_1)\, \mathfrak{T}(\bar{\mathbf{x}})\, W_e(\mathbf{x}_{k-1} \to \mathbf{x}_k). \tag{4}$$

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:

$$\mathfrak{T}(\bar{\mathbf{x}}) = \left[\prod_{i=1}^{k-1} \rho(\mathbf{x}_i)\right] \left[\prod_{i=0}^{k-1} G(\mathbf{x}_i, \mathbf{x}_{i+1}) V(\mathbf{x}_i, \mathbf{x}_{i+1})\right]. \tag{5}$$

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:

$$\rho(\mathbf{x}_i) = \begin{cases} \rho_s(\mathbf{x}_{i-1} \to \mathbf{x}_i \to \mathbf{x}_{i+1}) & \mathbf{x}_i \text{ on surface,} \\ \rho_p(\mathbf{x}_{i-1} \to \mathbf{x}_i \to \mathbf{x}_{i+1}) \sigma_s(\mathbf{x}_i) & \mathbf{x}_i \text{ in medium,} \end{cases} \tag{6}$$

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 (3), Monte Carlo solutions are used to approximate the path

integral as a Monte Carlo estimator:

$$\langle I \rangle = \frac{1}{n} \sum_{j=1}^{n} \frac{f(\bar{\mathbf{x}}_j)}{p(\bar{\mathbf{x}}_j)}, \tag{7}$$

that averages the contribution of n random paths $\bar{\mathbf{x}}_j$, sampled with a probability distribution function (pdf) $p(\bar{\mathbf{x}}_j) = p(\mathbf{x}_0 ... \mathbf{x}_k)$ 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 $\bar{\mathbf{x}}_w$ from the eye with pdf $p(\bar{\mathbf{x}}_w)$ and a subpath $\bar{\mathbf{x}}_l$ from the light with pdf $p(\bar{\mathbf{x}}_l)$. These are then (optionally) connected using a shadow ray to build the full path $\bar{\mathbf{x}}$ with pdf $p(\bar{\mathbf{x}}) = p(\bar{\mathbf{x}}_l)p(\bar{\mathbf{x}}_w)$ (see Figure 2, top).

Vector Path Integral The *vector path integral* takes a similar form as Equation (3), with important differences on the path contribution function $f(\bar{\mathbf{x}})$ (Equation (11)), in particular in the path throughput $\mathfrak{T}(\bar{\mathbf{x}})$ and in the sensor importance $W_e(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k)$. In the following we will assume monochromatic light (i.e. singe wavelength).

In its vector-form, the scattering kernel at $\mathbf{x}_i$ is no longer a scalar term $\rho(\mathbf{x}_i)$, but a Müller matrix $\mathbf{M}(\mathbf{x}_i)$ modeling the scattering interaction with the Stokes vector representing light. This imposes two constraints: the first one is that, as discussed earlier, the product between the incoming radiance and $\mathbf{M}(\mathbf{x}_i)$, as well as the outgoing radiance, has to be in valid reference systems. Therefore, we need to rotate the frames of the incoming and outgoing lights to match the frame on which $\mathbf{M}(\mathbf{x}_i)$ is defined, such that the $\mathbf{y}$ of both frames lay in the plane defined by the incoming and outgoing directions (see Figure 1). Therefore, the scattering kernel becomes $\mathbf{S} = \mathbf{R}(-\alpha_o)\mathbf{M}(\mathbf{x}_i)\mathbf{R}(\alpha_i)$, 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.

In addition to the rotation, working with Müller matrices imposes the need of defining how the operations are concatenated: as opposed to the traditional path integral, here the order on which the operations are concatenated is important. For that, let us define the concatenation operation as:

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

where $\mathbf{S}(\mathbf{x}_i)$ is the scattering operator in matrix form. Interestingly, this formulation would be also valid for other matrix-based formulation of the path integral e.g. re-radiation matrices for fluorescence. The order imposed in Equation (8) needs to be applied also to the contribution function.

This leaves the vector path throughput $\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}})$ as:

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

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

The second important change of the vector contribution function $f_\mathbf{v}(\bar{\mathbf{x}})$ with respect to the standard one occurs in the sensor importance $W_e(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k)$: as described in Section 2, we cannot sum two polarized beams of light incoming a single point if they have different ray direction $\mathbf{w}$. Therefore, the sensor importance needs to: *a)* rotate the incoming radiance to the same reference frame as the sensor, and *b)* transform the light to the same output reference frame, so it can be added. Thus, the vector sensor importance becomes $\mathbf{W}_e(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k) = \mathbf{R}(\alpha_e) W_e(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k) \mathbf{R}(\alpha_i)$, where $\mathbf{R}(\alpha_e)$ is the transformation applied to the captured light so that is in the same reference frame as the sensor.

By applying Equation (9) to Equations (3) and (11) we get the path integral formulation in vector form:

$$I = \int_\Omega f_\mathbf{v}(\bar{\mathbf{x}})\, \mathrm{d}\mu(\bar{\mathbf{x}}), \tag{10}$$

$$f_\mathbf{v}(\bar{\mathbf{x}}) = \mathbf{W}_e(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k)\, \mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}})\, L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1). \tag{11}$$

This equation can be computed in the same manner as the traditional path integral, following Equation (7), and can be straightforwardly extended to support the *transient path integral* proposed by Jarabo et al. [JMM*14]. In the following, we explain how this generalized formulation changes the computations in bidirectional path tracing.

4. Bidirectional Rendering of Polarization

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

Bidirectional methods [VG94, Jen01, GKH*13, HPJ12] compute the path integral by sampling several light paths joining the light and the sensor. This is done by sampling two different random walks (*subpaths*), each starting from the initial and final vertices of the path, which 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 formulation, 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 $\mathbf{S}(\mathbf{x}_i)$, 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. 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. This is key, since if affects on how the scattering kernels at each vertex of the subpath are defined. Thus, the main difficulty is to keep track on whether the incoming or outgoing directions are real (i.e. in the direction of the light, ω_i and ω_o respectively) or are a result on how the subpath is being constructed

(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, since we sample proportionally with the intensity (i.e. S_0), which thanks to Helmholtz reciprocity is symmetric in both the light subpath and its adjoint (sensor subpath). Then, we create the scattering kernel $\mathbf{S}(\mathbf{x}_i)$ 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 apply to the subpath accumulated throughput using Equation (8). Therefore, as opposed to multiplying $\mathbf{S}(\mathbf{x}_i)$ 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 we are using, 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:

$$\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}) = \mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_w)\mathbf{S}(\mathbf{x}_{k-m})G(\mathbf{x}_{k-m},\mathbf{x}_m)V(\mathbf{x}_{k-m},\mathbf{x}_m)\mathbf{S}(\mathbf{x}_m)\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_l). \tag{12}$$

Sampling Vector Scattering Kernels Given that the maximum value on the Stokes vector S is always its first component S_0 (light intensity), we opt for sampling the scattering outgoing direction with pdf $p(\omega_\mathrm{o}|\omega_\mathrm{i})$ proportional to the variation on S_0. In essence, we use traditional non-polarized scattering sampling procedures to sample ω_o. The immediate consequence of this choice is that we can use the sampling techniques when generating both the light and sensor subpaths, given that for this component there is symmetry according to Helmholtz reciprocity. Of course, this choice is only valid for polarization rendering, and might be suboptimal for strongly polarized light, as well as for other non-symmetric light transport. These cases would require specialized sampling routines aware of e.g. the polarization state of incoming light.

4.1. Photon Mapping

As shown by Georgiev et al. [GKH*13] 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 together 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).

Both Georgiev et al. [GKH*13] and Hachisuka et al. [HPJ12] shown that this means that we can define PM under the path integral framework, with some small variants, which means that we can introduce photon mapping in our vector formulation of the path integral. In fact, the main difference arises on how to merge the eye and light subpaths. While for bidirectional path tracing we make use of Equation (12), 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:

$$\mathfrak{T}'_\mathbf{v}(\bar{\mathbf{x}}) = \mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_w)K_R(\|\mathbf{x}_{k-m}-\mathbf{x}_m\|)\mathbf{S}(\mathbf{x}_{k-m})\mathfrak{T}_\mathbf{v}(\bar{\mathbf{x}}_l), \tag{13}$$

where 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 2) we do only have to apply one scattering kernel $\mathbf{S}(\mathbf{x}_{k-m})$, since otherwise we would be applying it twice. Additionally, note that the scattering kernel $\mathbf{S}(\mathbf{x}_{k-m})$ 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 polarized rendering on top of an in-house physically-based rendering witten in C++, which was limited to spectral radiance values. To avoid inherent costs of supporting polarization when not needed, we fully templatize the light in our light transport, allowing to select the spectral or polarized version of the code in compiler time.

In many cases, working with the full Müller matrices was not needed, given that e.g. light was unpolarized or the scattering kernel itself was a depolarizer. We add a flag to both the Stokes vectors and Müller 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; this results into scattering kernels of size $3\times4\times4$ 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 (4×4) Müller matrices are relatively sparse for most common operations, which can be exploited to increase performance.

We have included four types of scattering kernels, which are the standard when rendering with polarization: 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], and a weakly polarizing Mie phase function obtained using MiePlot [Lav15].

Finally, there is no standard file format for Stokes images. In order to store the resulting images, we opt for compressing them into Radiance HDR (.hdr), although any other HDR format (e.g. EXR) would work. However, we need to take into account that $S_{1..3}$ might be negative. We instead store a normalized value $S'_{1..3}$, imposing that $S'_{1..3} \geq 0$, as:

$$S'_{1..3} = \frac{S_{1..3}}{S_0} + 1. \tag{14}$$

Given this normalization, other linear image formats could be used instead.

Figure	Scene	# Photons	Unpolarized	Polarized
3	*mirror*	527K	8	14
4	*bunny*	300K	17	33

Table 1: Comparison of the average cost (in seconds) per iteration between rendering with and without polarization, for the render examples shown in the paper. The number of photons in the scene show the average number of photons per iteration (1M photon shots).

5. Results

We demonstrate our implementation by rendering two scenes with complex light interactions, including dielectric, conductors and participating media. One of the scenes (Figure 3) consists of a scene reflected on a metallic (aluminum) spherical mirror, which is what the camera looks at. The scene includes several diffuse surfaces, as well as another planar conductor mirror, and a dielectric non-absorbing sphere made of glass. The second scene (Figure 4) is a dielectric tank (glass) filled with milk diluted in water, containing two diffuse planes, and a bunny, also diffuse. Most light transport suffers polarization given that light has to pass through a dielectric, which is highly polarizing. The medium itself slightly polarizes light, although after multiple scattering this polarization is lost.

In both cases our scenes are illuminated by a point, unpolarized light. Given that the the caustic paths due to smooth dielectric and conductors are dominant (in the case of the tank all transport is caustic), we use a stochastic progressive [HJ09, KZ11] version of our polarized photon mapping (Section 4.1) in both scenes, although we also account for multiple bounces on the sensor subpath, and perform deterministic shadow connections with the light. We computed 500 iterations, with 16 samples per pixel and shooting 1000000 photon random walks on each iteration. We compute the initial kernel radius using the 25 nearest photons. Note that without bidirectional methods such as photon mapping these scenes, where specular light paths dominate, could be very hard, or even impossible, to render.

For visualizing the polarization in the scene, we use the techniques described by Wilkie and Weidlich [WW10], in particular the degree of polarization (from blue –unpolarized– to yellow –strongly polarized) and the relative degree of linear polarization (from blue –circularly polarized – to yellow –linearly polarized). In Figure 3 we can observe how the two mirrors (conductors) are bad polarizers, and polarize light only slightly. However, the dielectric is a very good polarizer. Moreover, we can see how when the frames of the spherical mirror and both the dielectric sphere and the planar mirror coincide then the linear polarization is preserved. However, when the frame of reference has to change (e.g. in poles of the dielectric sphere or in the top of the planar mirror) the linear polarization switches to circular polarization.

These observations also hold for Figure 4, where we can see how as the optical depth increases (more scattering), then the polarization is lost. This is because the multiple scattering tends to weaken the effect of polarization. We can also see how, since the dielectric surfaces are planes, then most of the polarization is linear, and only when multiple scattering dominates the circular polarization starts to be visible.

In terms of performance, adding polarization 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 also introduces additional costs. Note, however, that the vector and matrix operations are not optimized; using vectorial code could reduce the costs significantly. Additionally, while the representation of polarization is compact, the memory cost is much higher than traditional rendering: for example, for 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.

6. Conclusions & Future Work

In this work we aimed for two things: first of all, formalizing the notion of polarization by generalizing the path integral including the constraints of this type of illumination. Interestingly, these modifications do not reduce generality, but extends its range of applicability to transport effects where the Helmholtz reciprocity is broken. Based on this theoretical framework, we have described the required changes to well-known (scalar) bidirectional rendering methods that can be defined within the traditional and its extended unified path integral framework. In particular, we have shown how to include polarization in both bidirectional path tracing and photon mapping.

An additional goal was to provide links to the references needed to implement a physically-accurate rendering with polarization, and in particular to the Müller matrices used to define light-matter interactions, as well as details on how to implement bidirectional methods, so that it complements the excellent tutorial on rendering polarization by Wilkie and Weidlich [WW12].

There is of course several future work ahead. First of all, on a theoretical level our proposed vector path integral can be straightforwardly extended to other non-symmetric effects. Adding these new effects is an obvious and interesting first step for demonstrating the generality of the approach. Showing how our formulation fits into the transient path integral [JMM*14], and analyzing the time-resolved effects of polarization is a very interesting avenue for future work. While some of them have been already demonstrated in a single-directional basis (e.g. birefringence, flourescence), several questions at very small temporal resolution (i.e. femtosecond resolution) still remain, including how to model the retardance of phase shifts when using a super-short impulse illumination within a Stokes vector formulation. Finally, there is still a lack of BRDF models supporting polarization in graphics, beyond the boundary cases of perfect Lambertian or smooth surfaces. Adding polarization to microfacet models [HHdD16], probably with support for multi-layered materials [JdJM14] is a very interesting avenue of future work.

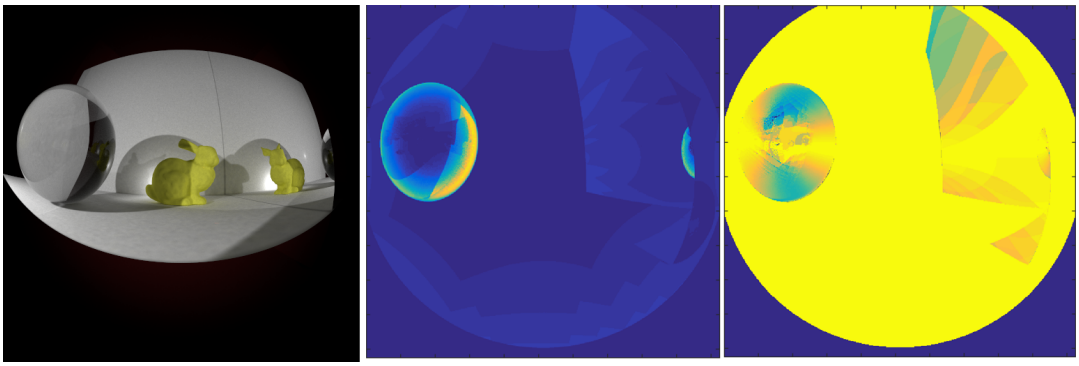

Figure 3: *Mirror* scene, depicting a scene with a dielectric sphere and a conductor mirror reflected on a spherical conductor mirror (left). The image in the center shows the degree of polarization (yellow is higher), while on the right we represent the ratio of linear polarization vs circular polarization (yellow is more linearly polarized). While the conductors only polarize light weakly, the dielectric specular reflection is highly polarized. In addition, the concatenation of multiple reflections on curved surfaces cause that the linear polarized light from specular reflection shifts towards circular polarization.

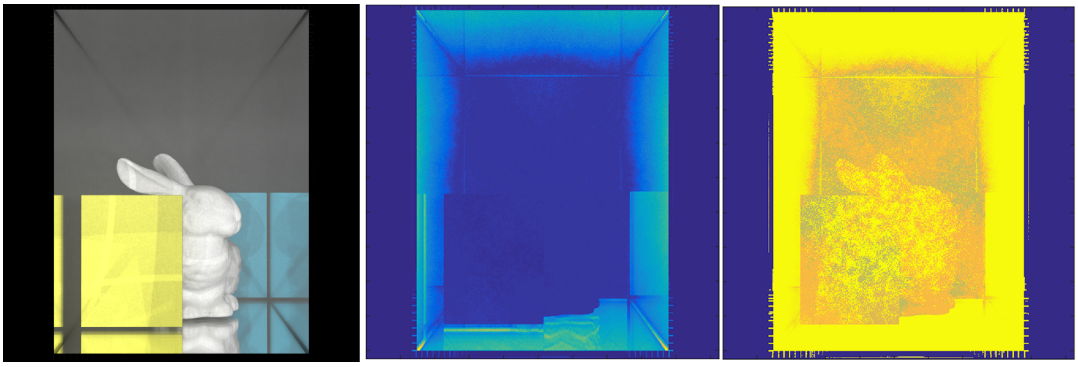

Figure 4: *Tank* scene, depicting a scene with a set of Lambertian (depolarizers) objects within a tank filled with milk diluted in water. The tank surfaces are dielectric surfaces (glass) (left). The image in the center shows the degree of polarization (yellow is higher), while on the right we represent the ratio of linear polarization vs circular polarization (yellow is more linearly polarized). The boundaries of the tank strongly linearly polarize light, which as the optical depth increases, and therefore the amount of scattering suffered by light, it becomes more unpolarized.

Acknowledgements

We want to thank Adolfo Muñoz and Rihui Wu for discussions about polarized light and comments on the results, and the annonymous reviewers for their constructive suggestions. This work has been funded by the Spanish Ministry of Economy and Competitiveness through project LIGHTSLICE, and DARPA through project REVEAL.

References

[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

[Cha60] CHANDRASEKHAR S.: *Radiative Transfer*. Dover, 1960. 3

[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). 3

[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). 3

[GKH*13] GEORGIEV I., KŘIVÁNEK J., HACHISUKA T., NOWROUZEZAHRAI D., JAROSZ W.: Joint importance sampling of low-order volumetric scattering. *ACM Trans. Graph. 32*, 6 (2013). 1, 4, 5

[Gla95] GLASSNER A. S.: A model for fluorescence and phosphorescence. In *Photorealistic Rendering Techniques*. 1995. 1, 2, 3

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

[GSMA08] GUTIERREZ D., SERON F., MUÑOZ A., ANSON O.: Visualizing underwater ocean optics. *Computer Graphics Forum 27*, 2 (2008). 1, 3

[Hac07] HACHISUKA T.: Generalized polarization ray tracing using a Monte Carlo method. CSE 272, University of California at San Diego, 2007. 2, 3

[HHdD16] HEITZ E., HANIKA J., D'EON E., DACHSBACHER C.: Multiple scattering microfacet BSDFs with the Smith model. *ACM Trans. Graph.* (2016), to appear. 6

[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, 4, 5

[HTSG91] HE X. D., TORRANCE K. E., SILLION F. X., GREENBERG D. P.: A comprehensive physical model for light reflection. *SIGGRAPH Comput. Graph. 25*, 4 (1991). 3

[JdJM14] JAKOB W., D'EON E., JAKOB O., MARSCHNER S.: A comprehensive framework for rendering layered materials. *ACM Trans. Graph. 33*, 4 (2014). 6

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

[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, 4, 6

[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. 5

[LMNS12] LETNES P. A., MARADUDIN A. A., NORDAM T., SIMONSEN I.: Calculation of the mueller matrix for scattering of light from two-dimensional rough surfaces. *Physical Review A 86*, 3 (2012). 3

[LSG12] LATORRE P., SERON F., GUTIERREZ D.: Birefringency: Calculation of refracted ray paths in biaxial crystals. *The Visual Computer 28*, 4 (2012). 1, 3

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Compugraphics '93* (1993). 1, 4

[MMRO13] MUSBACH A., MEYER G. W., REITICH F., OH S. H.: Full wave modelling of light propagation and reflection. *Computer Graphics Forum 32*, 6 (2013). 1

[Mor81] MORAVEC H. P.: 3d graphics and the wave theory. *SIGGRAPH Comput. Graph. 15*, 3 (1981). 1

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

[Shu77] SHUMAKER J. B.: Distribution of optical radiation with respect to polarization. *Self–Study Manual on Optical Radiation Measurements, Part 1* (1977). 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, 3

[TTW94] TANNENBAUM D. C., TANNENBAUM P., WOZNY M. J.: Polarization and birefringency considerations in rendering. In *SIGGRAPH '94* (1994). 2

[Vea97] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford, 1997. 1, 3

[VG94] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Proc. of Eurographics Rendering Workshop* (1994). 1, 4

[WK90] WOLFF L. B., KURLANDER D. J.: Ray tracing with polarization parameters. *IEEE Computer Graphics and Applications 10*, 6 (1990). 2

[WTP01] WILKIE A., TOBLER R. F., PURGATHOFER W.: Combined rendering of polarization and fluorescence effects. In *Proc. EGSR '04* (2001). 2, 3

[WW08] WEIDLICH A., WILKIE A.: Realistic rendering of birefringency in uniaxial crystals. *ACM Trans. Graph. 27*, 1 (2008). 1, 3

[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). 3

[WW12] WILKIE A., WEIDLICH A.: Polarised light in computer graphics. In *SIGGRAPH Asia 2012 Courses* (2012). 2, 3, 5, 6

[WZP04] WILKIE A., ZOTTI G., PURGATHOFER W.: An analytical model for skylight polarisation. In *Proc. EGSR '04* (2004). 3