Eurographics Symposium on Rendering - Experimental Ideas & Implementations (2016)
E. Eisemann and E. Fiume (Editors)

# Bi-Directional Polarised Light Transport

M. Mojzík T. Skřivan A. Wilkie J. Křivánek

Charles University in Prague, Czech Republic

**Abstract**

*While there has been considerable applied research in computer graphics on polarisation rendering, no principled investigation of how the inclusion of polarisation information affects the mathematical formalisms that are used to describe light transport algorithms has been conducted so far. Simple uni-directional rendering techniques do not necessarily require such considerations: but for modern bi-directional light transport simulation algorithms, an in-depth solution is needed.*
*In this paper, we first define the transport equation for polarised light based on the Stokes Vector formalism. We then define a notion of polarised visual importance, and we show that it can be conveniently represented by a 4 × 4 matrix, similar to the Mueller matrices used to represent polarised surface reflectance. Based on this representation, we then define the adjoint transport equation for polarised importance. Additionally, we write down the path integral formulation for polarised light, and point out its salient differences from the usual formulation for light intensities. Based on the above formulations, we extend some recently proposed advanced light transport simulation algorithms to support polarised light, both in surface and volumetric transport. In doing that, we point out optimisation strategies that can be used to minimise the overhead incurred by including polarisation support into such algorithms.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Display Algorithms

## 1. Introduction

Polarisation is a fundamental property of electromagnetic radiation. Humans are not directly capable of perceiving it (at least not strongly so), but in Computer Graphics, inclusion of the effect can be necessary for highly accurate renderings of scenes that contain specular surfaces and inter-reflections, especially in outdoor settings. As the visual effects caused by the phenomenon are often rather subtle, and as the engineering changes needed to support it in a given rendering codebase are non-trivial, currently only a few rendering research systems offer simulation of the effect as a feature: no commercially available rendering packages are capable of handling the effect. However, continually rising demands on the accuracy and reliability of rendering solutions makes it feasible that polarisation support will become a regular feature of rendering software in the future.

But some issues still remain to be solved on the path to that point. While the basics of polarisation support in rendering systems are reasonably well understood [WW12], rendering technology has advanced considerably in the last years, and with it the complexity of the core light transport algorithms [KKG*14]. As such, our goal in this paper is to provide a solid foundation for the inclusion of polarisation support into a modern, bi-directional renderer. In order to avoid ambiguities related to the way the light transport algorithms are affected by this, we develop a theoretical background that describes exactly what is being computed, to generally understand the role played by the additional information that is present in such a system, and in particular, how visual importance is to be handled. Generally, polarisation renderers require differentiation between data structures that describe light and attenuation: the usually tacitly assumed symmetry of using RGB (or spectral) values for all colour-related quantities in a renderer is broken in such systems. And in a bi-directional polarisation rendering system, visual importance turns out to be yet a third distinct data type: as it turns out, neither light nor attenuation data structures are adequately capable of describing this quantity. This finding is fairly important, as only this knowledge enables one to implement polarisation-capable versions of modern light transport algorithms correctly. Additionally, we provide a reference implementation of polarisation support in bi-directional path tracing, photon mapping and volumetric path tracing, which involves a number of engineering decisions that may not be obvious from the outset.

This paper is organised as follows: in section 2, we first give a brief overview over the area of polarisation rendering. In section 3, we present an overview of the formalisms used to describe light polarisation, and we outline some aspects of light transport theory that we base our later derivations on. In section 4, we conduct a formal derivation of polarised light transport to match the formalisms





DOI: 10.2312/sre.20161215

introduced in the previous section. In section 5, we discuss the implementation we performed to test the validity of our derivations, and we finally offer some conclusions on our findings.

## 2. Related Work

The state of research in polarisation-aware rendering, which is presented in considerable detail in [WWG12], can be briefly summarised as follows: there are known basic techniques, mathematical formalisms and infrastructure components needed to get a prototype path-tracer capable of handling basic scene geometries and surfaces to work [WK90, WTP01, WW10]. Some progress has also been made in covering the handling of some basic effects and settings, such as perfectly specular surfaces, and simple sky-dome models [WUT*04, WW11]. In addition to this, there has been work on materials and geometries for which polarisation is a characteristic feature, such as bi-refringent crystals [TTW94, WW07, LSG12], and rainbows [SML*12]. And recently, there also has been work on more complex sub-surface scattering models that involve polarisation [CPLB14].

Beyond rendering proper, where its role is still limited, polarisation plays a much bigger role in the capture community [MTHI03]. Face scanning devices like light stages routinely use analysis of the polarisation state of reflected light to separate various components of reflected light from each other [GFT*11], and polarisation is generally useful to help with the acquisition of material extended material properties, such as glossiness [GCP*10].

## 3. Background

### 3.1. The Stokes Vector Representation of Polarised Light

**Stokes vectors.** For expressing the polarisation state of light in computer graphics, the Stokes vector representation is commonly used [Gol03]. A Stokes vector has 4 components $S_0$, $S_1$, $S_2$ and $S_3$, where $S_0$ specifies the radiant intensity of light, $S_1$ specifies the preference of horizontal to vertical linear polarisation, $S_2$ specifies the preference of $45^\circ$ to $135^\circ$ linear polarisation and $S_3$ specifier the preference of right to left circular polarisation. Additional constraints on the values are that $S_0 \in \mathbb{R}^+$, $S_1, S_2, S_3 \in [-S_0, S_0]$ and $S_0^2 \geq S_1^2 + S_2^1 + S_3^2$. For example, a non-polarised light of radiant intensity 2 would be a vector $(2,0,0,0)$, while a light with the same intensity that is completely left circularly polarised would be $(2,0,0,-2)$.

**Mueller matrices.** For defining how the intensity or polarisation state of light changes during its propagation, the counterpart to Stokes vectors are Mueller matrices, $4 \times 4$ real-valued matrices. Examples of Mueller matrices are the matrix for ideal linear polarisation filter $\mathbf{P}(\theta)$, where $\theta$ specifies the polarisation angle, and for ideal retarder $\mathbf{Q}(\phi)$, where $\phi$ specifies the induced phase difference:

$$\mathbf{P}(\theta) = \frac{1}{2}\begin{pmatrix} 1 & \cos 2\theta & \sin 2\theta & 0 \\ \cos 2\theta & \cos^2 2\theta & \sin 2\theta \cdot \cos 2\theta & 0 \\ \sin 2\theta & \sin 2\theta \cdot \cos 2\theta & \sin^2 2\theta & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{Q}(\phi) = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\phi & \sin\phi \\ 0 & 0 & -\sin\phi & \cos\phi \end{pmatrix}$$

The permissible operations with these structures are addition of two Stokes vectors, and multiplication of Mueller matrix and a Stokes vector to get another Stokes vector. In addition to these, there are several others that emerge from these, such as the multiplication of two Mueller matrices (thanks to associativity), addition of two Mueller matrices (thanks to distributivity) and multiplication by scalar value $v$ of either of the structures (thanks to a scalar multiple of the identity matrix being a valid Mueller matrix).

**Coordinate systems.** The polarisation of light is described within the 2D plane perpendicular to the direction of light propagation and specifically the $S_1$ and $S_2$ components of the Stokes vectors are dependent on the choice of the coordinate system within that plane. Whenever any calculation is done with regard to Stokes vectors and Mueller matrices, it is important to express all of the elements in the appropriate coordinate systems. Specifically, whenever any two Stokes vectors are added together, they have to be expressed in the same coordinate system; Whenever a Muller matrix multiplies a Stokes vector, the Stokes vector has to be expressed in the same coordinate system that the Mueller matrix expects on its input, while the coordinate system in which the multiplication result is expressed depends on the output coordinate system in which the matrix is defined.

If it would so happen that one quantity is expressed in a different coordinate system than the one expected, it is possible to use the following rotation matrix

$$\mathbf{R}(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\phi) & \sin(2\phi) & 0 \\ 0 & -\sin(2\phi) & \cos(2\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where $\phi$ would be the angle by which it is necessary to rotate around the propagation direction to reach the expected coordinate system. The rotation matrix $\mathbf{R}$ can only fix the orientation of the coordinate system for a specific direction of light propagation. It is not physically meaningful to add Stokes vectors that describe light which propagates in different directions, or to multiply Stokes vector and Mueller matrices that define or expect light in different directions.

**Measuring polarised light.** A useful property of Stokes vectors is that their components can be directly measured. By doing four different measurements of intensity of light with a retarder and a polarisation filter, we can compute all components of a Stokes vector. When light passes through a retarder and a polarisation filter, its Stokes vector changes from $S$ to $\mathbf{P}(\theta)\mathbf{Q}(\phi)S$. The intensity of this light can be obtained by taking dot product with $e_0^T = (1,0,0,0)$, the measured intensity is $e_0^T\mathbf{P}(\theta)\mathbf{Q}(\phi)S$. We take four different measurements with different polarisation angles $\theta$ and phase shifts

$\phi$. They can be neatly expressed in matrix form

$$\begin{pmatrix} e_0^T\mathbf{P}(0^\circ)\mathbf{Q}(0^\circ)S \\ e_0^T\mathbf{P}(45^\circ)\mathbf{Q}(0^\circ)S \\ e_0^T\mathbf{P}(90^\circ)\mathbf{Q}(0^\circ)S \\ e_0^T\mathbf{P}(45^\circ)\mathbf{Q}(90^\circ)S \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{pmatrix} \quad (1)$$

By inverting the above relation, we can obtain all components of a Stokes vector from four physically doable measurements. Note, however, that we can construct any row vector $w^T$ through linear combination of the four measurement vectors and interpret $w^T S$ as a measurement that we can obtain by a linear combination of the four measurements: but not all these "measurements" are physically meaningful. For example, the canonical basis $e_0^T, e_1^T, e_2^T, e_3^T$ can be thought of as directly measuring components of a Stokes vector, even though such measurements can not be performed in reality. Since we do not share all the restrictions of the real world in computer graphics, we actually use this convenient canonical basis for measurements, instead of the physically meaningful measurements described before. But we will strictly distinguish between a component $S_i$ of a Stokes vector $S$, and a measured value $e_i^T S$. The reason is that adding two Stokes vectors has meaning only when they describe light propagating in the same direction, while summing results from sensors with different orientations is perfectly fine and the sum is just the total energy received by those sensors. This distinction will be important when defining visual importance in the context of polarising light transport. The last thing to mention is that we will always take four simultaneous measurements so instead of working with row vectors for a single measurement we will be working with matrices such as in Eq. 1.

## 4. Theory

**Motivation.** One of the major difficulties when implementing a polarised light transport simulation is to make sure that all the coordinate systems used during the computations meaningfully align. So before generalising the light transport equations, we first discuss how to properly handle this multitude of coordinate systems in a formal way. Much of the literature about polarised light deals with fairly simple geometries where only a few coordinate systems that can be specified manually are needed [Gol03]. However, this is not the case in computer graphics where scene geometry, and by extension, path propagation, can be arbitrarily complex. So fixing a few select coordinate systems beforehand is not a viable option. Instead, we keep track of the corresponding coordinate system for each Stokes vector and Mueller matrix. This is why we need to formulate this "coordinate system bookkeeping" already at the level of our theory.

### 4.1. Stokes Space

**Coordinate system notation.** An orthogonal coordinate system of the 2D plane perpendicular to the light direction $\omega$ is uniquely determined by a unit vector $u$ orthogonal to the direction $\omega$. Such a coordinate system is formed by vectors $u$ and $\omega \times u$; we will denote it by $F_u$ and the set of all coordinate system of this plane by $\mathbb{F}_\omega$. If we want to change the coordinate system of a Stokes vector $S \in \mathbb{R}^4$ from $F_u$ to $F_v$, we have to multiply $S$ by the matrix $\mathbf{R}(\phi)$, where $\phi$ is the oriented angle between $u$ and $v$. For the sake of brevity, we do not want to refer to the specific directions $u, v$ and the angle $\phi$ every time we change the coordinate system. Therefore if we have two coordinate systems $F, G \in \mathbb{F}_\omega$, we denote the corresponding rotation matrix $\mathbf{R}(\phi)$, which transforms a Stokes vector from $F$ to $G$, by $GF^{-1}$. Note that we do not attribute any meaning to the symbol $F^{-1}$ on its own; only the symbol $GF^{-1}$ as whole has one.

**Stokes space definition.** On its own, a Stokes vector $S \in \mathbb{R}^4$ is just a collection of four numbers. To assign a physical meaning to it, we need to know the light direction $\omega$ and the coordinate system $F \in \mathbb{F}_\omega$ in which it is expressed. Therefore, instead of working with just a Stokes vector $S$, we work with pairs $[S,F]$ that consist of a Stokes vector and its coordinate system.

The physical quantity that we are attempting to describe does not depend on any particular coordinate system that we use, only the Stokes vector representation does. A pair $[GF^{-1}S, G]$ for $G \in \mathbb{F}_\omega$ expresses the exact same quantity as $[S,F]$. As we would like to describe the physical quantity itself using the notation of pair $[S,F]$, we say that the following alternate representations are equal:

$$[S,F] = [GF^{-1}S, G], \qquad F, G \in \mathbb{F}_\omega.$$

We denote the space of all pairs $[S,F]$ which describe polarised light traveling in the direction $\omega$ by $\mathbb{S}_\omega$ and call it the *Stokes space*. For a precise definition of the Stokes space see Appendix A.

**Discussion.** At first glance, it may seem overly complicated to define something like the Stokes space. But later on, this formalism will greatly simplify our notation. Also, it bears a close resemblance to the actual implementation of a polarisation-capable renderer, where one has to store a coordinate space together with each Stokes vector. From now on, we refer to pairs $[S,F]$ as Stokes vectors; this should not cause any confusion as most of the time we want to address the actual physical quantity, instead of the particular representation of Stokes vector in a given coordinate system.

**Stokes vector addition.** We can add two Stokes vectors together if, and only if, the light direction is the same for both. In addition, we have to ensure that their coordinate systems are aligned before summing them. This is all easy to formulate in Stokes space: if we have two Stokes vectors $[S,F], [T,G] \in \mathbb{S}_\omega$, their sum is defined as

$$[S,F] + [T,G] = [S + FG^{-1}T, F].$$

It might seem that when written this way, the result of the addition depends on the order of Stokes vectors. But the contrary is true, as we show in Appendix A.

### 4.2. Mueller Space

The situation with Mueller matrices is similar to the Stokes vectors. Without knowing the directions of incoming and outgoing light and the corresponding coordinate systems, a Mueller matrix is just a table of numbers without a physical meaning. Therefore, instead of working with just a Mueller matrix $M \in \mathbb{R}^{4\times4}$, which describes optical element for incoming light in direction[†] $\omega_\mathrm{i}$ and outgoing light in direction $\omega_\mathrm{o}$, we work with triples $[M,F_\mathrm{i},F_\mathrm{o}]$ where $F_\mathrm{i} \in \mathbb{F}_{-\omega_\mathrm{i}}$ and $F_\mathrm{o} \in \mathbb{F}_{\omega_\mathrm{o}}$ are coordinate systems of incoming and outgoing light, respectively. The triple $[M,F_\mathrm{i},F_\mathrm{o}]$ represents a physical quantity and should not depend on the choice of the coordinate systems. Therefore, changing to coordinate systems $G_\mathrm{i} \in \mathbb{F}_{-\omega_\mathrm{i}}, G_\mathrm{o} \in \mathbb{F}_{\omega_\mathrm{o}}$ does not change it:

$$[M,F_\mathrm{i},F_\mathrm{o}] = [G_\mathrm{o}F_\mathrm{o}{}^{-1}MF_\mathrm{i}G_\mathrm{i}{}^{-1},G_\mathrm{i},G_\mathrm{o}].$$

We denote the space of all triples $[M,F_\mathrm{i},F_\mathrm{o}]$ which represents Mueller matrix for incoming light in direction $\omega_\mathrm{i}$ and outgoing light in direction $\omega_\mathrm{o}$ by $\mathbb{M}^{\omega_\mathrm{o}}_{-\omega_\mathrm{i}}$, and call it the *Mueller space*. For a precise definition of Mueller space see Appendix A.

**Convention on incoming light direction.** $\omega_\mathrm{i}$. In computer graphics, the common convention for the incoming light direction $\omega_\mathrm{i}$ is that it is opposite to the actual light propagation direction. This convention was introduced by Veach, because it brings symmetry into equations of light transport [Vea97, pg. 110]. Unfortunately, this symmetry is broken for polarising light anyway. Still, we have decided to conform to this convention because when one implements polarising light transport, one will most probably modify an existing code where this convention is already used. Therefore, in this paper the incoming direction $\omega_\mathrm{i}$ is always opposite to the direction of light propagation, but the subscripts in $\mathbb{F}_\omega, \mathbb{S}_\omega, \mathbb{M}^{\omega_\mathrm{o}}_{-\omega_\mathrm{i}}$ are always the direction of light propagation.

**Operations with Mueller matrices.** A Mueller matrix $[M,F_\mathrm{i},F_\mathrm{o}] \in \mathbb{M}^{\omega_\mathrm{o}}_{-\omega_\mathrm{i}}$ describes how a Stokes vector $[S,F] \in \mathbb{S}_{-\omega_\mathrm{i}}$ is modified when it goes through an optical element:

$$[M,F_\mathrm{i},F_\mathrm{o}][S,F] = [MF_\mathrm{i}F^{-1}S,F_\mathrm{o}].$$

Because we can sum Stokes vectors, we can also sum Mueller matrices. If we have two Mueller matrices $[M,F_\mathrm{i},F_\mathrm{o}],[N,G_\mathrm{i},G_\mathrm{o}] \in \mathbb{M}^{\omega_\mathrm{o}}_{-\omega_\mathrm{i}}$, their sum is

$$[M,F_\mathrm{i},F_\mathrm{o}]+[N,G_\mathrm{i},G_\mathrm{o}] = [M+F_\mathrm{o}G_\mathrm{o}{}^{-1}NG_\mathrm{i}F_\mathrm{i}{}^{-1},F_\mathrm{i},F_\mathrm{o}]$$

The result does not not depend on the order of addition (see Appendix A).

We can multiply Mueller matrices as well. But we have to be extremely careful, as the outgoing light direction of one matrix has to be the incoming light direction of the other matrix. We can multiply a Mueller matrix $[M,F_\mathrm{i},F_t] \in \mathbb{M}^{\omega_t}_{-\omega_\mathrm{i}}$ with a Mueller matrix $[N,G_t,F_\mathrm{o}] \in \mathbb{M}^{\omega_\mathrm{o}}_{\omega_t}$.

$$[N,G_t,F_\mathrm{o}][M,F_\mathrm{i},F_t] = [NG_tF_t{}^{-1}M,F_\mathrm{i},F_\mathrm{o}]$$

Notice that the outgoing coordinate system of $M$ is different from the incoming coordinate system of $N$, but the light direction is the same. Doing the multiplication in reverse order does not make any sense because the outgoing direction of $N$ does not match the incoming direction of $M$.

[†] The common convention in computer graphics is that incoming direction $\omega_\mathrm{i}$ is opposite to the actual direction of light propagation. The following paragraphs discuss this issue.

### 4.3. Importance Space

When performing measurements of polarised light, we have to be careful about the used coordinate system as well. Instead of working with just a row vector $w^T$, which represents a measurement in a coordinate system $F \in \mathbb{F}_\omega$, we work with the pair $[w^T,F]$. The act of measurement of a Stokes vector $[S,G] \in \mathbb{S}_\omega$ is expressed as a dot product, but we have to align the coordinate systems accordingly:

$$[w^T,F][S,G] = w^TFG^{-1}S.$$

The measurement does not depend on the chosen coordinate system, i.e.

$$[w^T,F] = [w^TFG^{-1},G] \qquad G \in \mathbb{F}_\omega$$

We denote the space of all pairs $[w^T,F]$ which represents a measurement of light traveling in direction $\omega$ by $\mathbb{I}_\omega$ and call it the *Importance space*.

**Matrix form of the importance space.** As we discussed in Sec. 3.1 above, we almost always do four simultaneous measurements, $[w_0^T,F],[w_1^T,F],[w_2^T,F],[w_3^T,F]$. We can assemble them into a single matrix $W^T$, and then work with the pair $[W^T,F]$ instead. The act of measuring a Stokes vector $[S,G] \in \mathbb{S}_\omega$ is done with matrix vector multiplication

$$[W^T,F][S,G] = W^TFG^{-1}S.$$

Notice that the result is a vector, but this time with no coordinate system attached to it. Indeed, it is just a collection of four measurements. This may seem counter-intuitive: when we pick $W^T$ equal to the identity matrix and $F = G$, then the result of the measurement is $S$, the four components of the Stokes vector. But keep in mind that it is not a real Stokes vector (member of the Stokes space), instead, it is just a collection of readings from sensors.

We denote the space of all pairs $[W^T,F]$ which represent four simultaneous measurements of light traveling in direction $\omega$ by $\bar{\mathbb{I}}_\omega$. For precise definitions of $\mathbb{I}_\omega$ and $\bar{\mathbb{I}}_\omega$ see Appendix A.

**Multiplication by Mueller matrix.** We can also multiply elements of $\mathbb{I}_\omega$ by a Mueller matrix. If we have a measurement $[w^T,F] \in \mathbb{I}_{\omega_\mathrm{o}}$ and a optical element $[M,F_\mathrm{i},F_\mathrm{o}] \in \mathbb{M}^{\omega_\mathrm{o}}_{-\omega_\mathrm{i}}$ we can include the optical element into the measurement and the new measurement is represented by:

$$[w^T,F][M,F_\mathrm{i},F_\mathrm{o}] = [w^TFF_\mathrm{o}{}^{-1}M,F_\mathrm{i}].$$

### 4.4. Radiometry of polarised light

We have just set the foundations of polarised light transport in such a way that we do not have to worry about coordinate systems much. We now can proceed with definitions of polarised radiance, the bi-directional scattering distribution function (BSDF), and visual importance.

**Radiance.** The central quantity in light transport is radiance. In polarising light transport the radiance becomes a Stokes vector-valued function. Because of the common convention on incoming light direction discussed earlier, we have to make a distinction between incoming and outgoing radiance, they are

$$\begin{aligned}\mathbf{L}_\mathrm{o} &: \mathcal{M} \times \mathcal{S}^2 \to \mathbb{S}_\omega \\ &(\mathbf{x}, \omega) \to \mathbf{L}_\mathrm{o}(\mathbf{x}, \omega) \\ \mathbf{L}_\mathrm{i} &: \mathcal{M} \times \mathcal{S}^2 \to \mathbb{S}_{-\omega} \\ &(\mathbf{x}, \omega) \to \mathbf{L}_\mathrm{i}(\mathbf{x}, \omega)\end{aligned}$$

and the emitted radiance $\mathbf{L}_\mathrm{e}$ is of the same type as outgoing radiance.

**Importance.** In polarising light transport importance becomes either a vector or matrix valued function, depending if one is interested in measuring light polarisation or not. A real-world example for the vector valued importance would be taking just a single picture with a particular polarising filter placed in front of it. An example for the matrix-valued importance would be taking the same picture with four different polarising filters, so one can later reconstruct the picture as taken with any other polarising filter. That allows us to decide on the polarising filter after the rendering has finished.

Because of the convention we have to again make a difference between incoming and outgoing importance, the vector valued importance is

$$\begin{aligned}\mathbf{w}_\mathrm{o}^T &: \mathcal{M} \times \mathcal{S}^2 \to \mathbb{I}_{-\omega} \\ &(\mathbf{x}, \omega) \to \mathbf{w}_\mathrm{o}^T(\mathbf{x}, \omega) \\ \mathbf{w}_\mathrm{i}^T &: \mathcal{M} \times \mathcal{S}^2 \to \mathbb{I}_{\omega} \\ &(\mathbf{x}, \omega) \to \mathbf{w}_\mathrm{i}^T(\mathbf{x}, \omega)\end{aligned}$$

and the matrix valued importance is

$$\begin{aligned}\mathbf{W}_\mathrm{o}^T &: \mathcal{M} \times \mathcal{S}^2 \to \bar{\mathbb{I}}_{-\omega} \\ &(\mathbf{x}, \omega) \to \mathbf{W}_\mathrm{o}^T(\mathbf{x}, \omega) \\ \mathbf{W}_\mathrm{i}^T &: \mathcal{M} \times \mathcal{S}^2 \to \bar{\mathbb{I}}_{\omega} \\ &(\mathbf{x}, \omega) \to \mathbf{W}_\mathrm{i}^T(\mathbf{x}, \omega)\end{aligned}$$

The emitted importance $\mathbf{w}_\mathrm{e}^T$ and $\mathbf{W}_\mathrm{e}^T$ is of the same type as outgoing importance. From now on, we will use only the matrix valued importance even in the theory, and resort to the vector valued one only when necessary.

**Asymmetry.** There is an apparent asymmetry of light transport when one uses different types for importance and radiance. This asymmetry is not fundamental: one could also write a system that only uses vector-valued importance, albeit with loss of expressivity (the end result is just a single measurement, not a full Stokes Vector for each pixel). Alternatively, it would also be possible to use matrix-valued radiance. This would allow one to alter the polarisation of light sources after the fact: a feature that is rarely, if ever needed, but adds considerably to the computational cost of the system. So in practice, asymmetric data types are what is normally used in real implementations of polarisation renderers: this allows one to apply a polarising filter to a rendered image, without imposing a too large overhead on the computation.

**Choice of importance.** As we discussed in section 3.1, we directly measure elements of the Stokes vector. Therefore the importance matrix $W^T$ is just an identity matrix, but we have still to specify in which coordinate system we do this measurement. We take the simplest possible choice, the upward direction $u$ with the respect to the camera, and orthogonalise it to the incoming light direction $\omega$:

$$v(\omega) = \frac{u - (u \cdot \omega)\omega}{\|u - (u \cdot \omega)\omega\|}.$$

The coordinate system for incoming light is then $F_{v(\omega)} \in \mathbb{F}_{-\omega}$ and the emitted importance is

$$\mathbf{W}_\mathrm{e}^T(\mathbf{x}, \omega) = [\mathrm{Id}, F_{v(\omega)}].$$

Where Id is the identity matrix. Other choices of importance should be investigated in the future, to better match real, non-ideal polarising filters.

**Measurement equation.** The analogue of the measurement equation [Vea97, pg. 89] in polarising light transport is the *polarised measurement equation*:

$$\mathbf{I} = \int_{\mathcal{M} \times \mathcal{S}^2} \mathbf{W}_\mathrm{e}^T(\mathbf{x}, \omega) \mathbf{L}_\mathrm{i}(\mathbf{x}, \omega) \, \mathrm{d}A(\mathbf{x}) \, \mathrm{d}\sigma_\mathbf{x}^\perp(\omega) \tag{2}$$

Thanks to the definitions of $\mathbb{S}_\omega$ and $\bar{\mathbb{I}}_\omega$, the polarised measurement equation does not differ much from the original measurement equation, but one has to be careful when interpreting it. The same is true about all other equations in polarised light transport, but as this is the first equation of this sort we introduce, we point out the salient aspects in this case. Incoming radiance and importance are both pairs of a vector (or a matrix) and a frame:

$$\begin{aligned}\mathbf{L}_\mathrm{i}(\mathbf{x}, \omega &= [L_\mathrm{i}(\mathbf{x}, \omega), G(\mathbf{x}, \omega)] \\ \mathbf{W}_\mathrm{e}^T(\mathbf{x}, \omega) &= [W_\mathrm{e}^T(\mathbf{x}, \omega), F(\mathbf{x}, \omega)]\end{aligned}$$

So the polarised measurement equation can be also written as

$$\mathbf{I} = \int_{\mathcal{M} \times \mathcal{S}^2} W_\mathrm{e}^T F G^{-1} L_\mathrm{i} \, \mathrm{d}A(\mathbf{x}) \, \mathrm{d}\sigma_\mathbf{x}^\perp(\omega)$$

where we have dropped the function arguments $(\mathbf{x}, \omega)$. This form makes the necessary coordinate system alignment explicit.

**BSDF.** Before stating the light transport equation for polarised light we have to generalise the notion of a BSDF. In polarising light transport a BSDF becomes a Mueller matrix-valued function

$$\begin{aligned}\mathbf{f}_\mathrm{s} &: \mathcal{M} \times \mathcal{S}^2 \times \mathcal{S}^2 \to \mathbb{M}_{-\omega_\mathrm{i}}^{\omega_\mathrm{o}} \\ &(\mathbf{x}, \omega_\mathrm{i}, \omega_\mathrm{o}) \to \mathbf{f}_\mathrm{s}(\mathbf{x}, \omega_\mathrm{i}, \omega_\mathrm{o})\end{aligned}$$

Notice the $-\omega_\mathrm{i}$ that is because of the convention that incoming direction $\omega_\mathrm{i}$ as function argument is opposite to the actual light propagation direction.

**Light transport equation.** The light transport equation stays almost the same as in the non-polarising case, we only replace radiance and BSDF with their polarised counterparts, and we have to be careful about the order of multiplication. The *polarised light* *transport equation* (a.k.a. the polarised rendering equation) is

$$\mathbf{L}_o(\mathbf{x},\omega_o) = \mathbf{L}_e(\mathbf{x},\omega_o) + \int_{\mathcal{S}^2} \mathbf{f}_s(\mathbf{x},\omega_i,\omega_o)\mathbf{L}_i(\mathbf{x},\omega_i)\,\mathrm{d}\sigma^{\perp}_{\mathbf{x}}(\omega_i) \quad (3)$$

Since the incoming direction of the Mueller matrix $\mathbf{f}_s(\mathbf{x},\omega_i,\omega_o)$ coincides with the direction of the incoming radiance $\mathbf{L}_i(\mathbf{x},\omega_i)$, the multiplication is well defined. Similarly the *polarised scattering operator* is defined as

$$(\mathbf{K}\mathbf{L}_o)(\mathbf{x},\omega) = \int_{\mathcal{S}^2} \mathbf{f}_s(\mathbf{x},\omega_i,\omega_o)\mathbf{L}_i(\mathbf{x},\omega_i)\,\mathrm{d}\sigma^{\perp}_{\mathbf{x}}(\omega_i).$$

Transport equations for importance can be derived as in [Vea97, pg. 132] from the measurement equation (2) and the light transport equation (3). The resulting *polarised importance transport equation* is

$$\mathbf{W}^T_o(\mathbf{x},\omega_o) = \mathbf{W}^T_e(\mathbf{x},\omega_o) + \int_{\mathcal{S}^2} \mathbf{W}^T_i(\mathbf{x},\omega_i)\mathbf{f}_s(\mathbf{x},\omega_o,\omega_i)\,\mathrm{d}\sigma^{\perp}_{\mathbf{x}}(\omega_i) \quad (4)$$

and the adjoint operator $\mathbf{K}^*$ is

$$(\mathbf{K}^*\mathbf{W}_o)(\mathbf{x},\omega) = \int_{\mathcal{S}^2} \mathbf{f}_s{}^T(\mathbf{x},\omega_o,\omega_i)\mathbf{W}_i(\mathbf{x},\omega_i)\,\mathrm{d}\sigma^{\perp}_{\mathbf{x}}(\omega_i)$$

**Propagation operator.** Converting outgoing radiance or importance to incoming radiance/importance can be done with the propagation operator $G$ [Vea97, pg. 110]. Its analogue in our theory of polarising light transport is

$$(\mathbf{G}\mathbf{H}_o)(\mathbf{x},\omega) = \begin{cases} \mathbf{H}_o(\mathbf{x}_{\mathcal{M}}(\mathbf{x},\omega),-\omega) & \text{if } d_{\mathcal{M}}(\mathbf{x},\omega) < \infty \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{H}_o$ is either outgoing radiance $\mathbf{L}_o$ or importance $\mathbf{W}_o$, $\mathbf{x}_{\mathcal{M}}(\mathbf{x},\omega)$ is the first point a light ray hits when it starts at the point $\mathbf{x}$ in the direction $\omega$ and $d_{\mathcal{M}}(\mathbf{x},\omega)$ is the distance between $\mathbf{x}$ and $\mathbf{x}_{\mathcal{M}}(\mathbf{x},\omega)$; if $\mathbf{x}_{\mathcal{M}}(\mathbf{x},\omega)$ does not exists then the distance is infinite [Vea97, pg. 110]. With the propagation operator, we can convert outgoing radiance/importance to incoming radiance/importance as follows:

$$\mathbf{L}_i = \mathbf{G}\mathbf{L}_o$$
$$\mathbf{W}_i = \mathbf{G}^*\mathbf{W}_o$$

and the polarised light transport equation 4 as:

$$\mathbf{L}_o = \mathbf{L}_e + \mathbf{K}\mathbf{G}\mathbf{L}_o.$$

Whose solution is

$$\mathbf{L}_o = \sum_{n=0}^{\infty} (\mathbf{K}\mathbf{G})^n \mathbf{L}_e$$

**Path integral formulation.** The last thing to do is to write down the rendering equation in path integral formulation. First, we need to establish some notation. The direction from a point $\mathbf{x}$ to a point $\mathbf{y}$ is denoted by $\mathbf{x} \to \mathbf{y}$ with this we write

$$\mathbf{L}_e(\mathbf{x}\to\mathbf{y}) = \mathbf{L}_e(\mathbf{x},\mathbf{x}\to\mathbf{y})$$
$$\mathbf{W}^T_e(\mathbf{x}\to\mathbf{y}) = \mathbf{W}^T_e(\mathbf{y},\mathbf{y}\to\mathbf{x})$$
$$\mathbf{f}_s(\mathbf{x}\to\mathbf{y}\to\mathbf{z}) = \mathbf{f}_s(\mathbf{y},\mathbf{y}\to\mathbf{x},\mathbf{y}\to\mathbf{z})$$

Furthermore $GV(\mathbf{x}\leftrightarrow\mathbf{y})$ is the geometry-visibility term [Vea97, pg. 221] between points $\mathbf{x}$ and $\mathbf{y}$. The measurement equation 2 in the path integral formulation has the form:

$$\mathbf{I} = \sum_{k=1}^{\infty} \int_{\mathcal{M}^{k+1}} \mathbf{W}^T_e(\mathbf{x}_{k-1}\to\mathbf{x}_k) \prod_{i=1}^{k-1} \mathbf{f}_s(\mathbf{x}_{i-1}\to\mathbf{x}_i\to\mathbf{x}_{i+1})GV(\mathbf{x}_{i-1}\leftrightarrow\mathbf{x}_i) \mathbf{L}_e(\mathbf{x}_0\to\mathbf{x}_1)GV(\mathbf{x}_0\leftrightarrow\mathbf{x}_1)\,\mathrm{d}A(\mathbf{x}_0)\dots\mathrm{d}A(\mathbf{x}_k)$$

Where the product of Mueller matrices has to be understood as

$$\prod_{i=1}^{k-1} \mathbf{f}_s(\mathbf{x}_{i-1}\to\mathbf{x}_i\to\mathbf{x}_{i+1})GV(\mathbf{x}_{i-1}\leftrightarrow\mathbf{x}_i) = \prod_{i=1}^{k-1} GV(\mathbf{x}_{i-1}\leftrightarrow\mathbf{x}_i)\, \mathbf{f}_s(\mathbf{x}_{k-2}\to\mathbf{x}_{k-1}\to\mathbf{x}_k)\dots\mathbf{f}_s(\mathbf{x}_0\to\mathbf{x}_1\to\mathbf{x}_2)$$

Note that the product is well defined, two consecutive Mueller matrices in the product are

$$\mathbf{f}_s(\mathbf{x}_i\to\mathbf{x}_{i+1}\to\mathbf{x}_{i+2})\mathbf{f}_s(\mathbf{x}_{i-1}\to\mathbf{x}_i\to\mathbf{x}_{i+1})$$

As it should, the outgoing light direction of the second matrix matches the incoming direction of the first matrix.

## 5. Implementation

We used SmallUPBP [KGH*14] as the basis of our reference implementation, and extended this codebase to be polarisation-capable by following the general guidelines given in Wilkie et al. [WWG12]. We built on the path integral formulation defined earlier to include polarisation support into a unidirectional and bidirectional path tracer [LW93], and to include volumetric scattering effects. Additionally, we also implemented a polarisation-capable version of Vertex Connection and Merging [GKDS12, HPJ12].

### 5.1. Standard polarisation support

**Data types.** The backbone of polarisation are structures that hold the Stokes vector and Mueller matrices. In our implementation, those are represented by the `Light` and `Attenuation` structures, respectively. To work with them, it is necessary to implement the eligible operations – linear scaling, Stokes vectors addition, Mueller matrix multiplication etc., as mentioned in the Background section. In our implementation, those are done through C++ operator overloading, which allowed a smooth transition from the original codebase to polarisation capable code. Both of these structures also have to include their associated coordinate reference frames, which makes the implementation of the eligible operations more clear (as opposed to possibly storing them separately). Keeping the frames together with the data structures also makes it easier to insert assertions into the code, e.g. to spot a reversed multiplication order for two Mueller matrices on a path.

**Returning correct types.** The next step is to use the defined structures as the results of BSDF and emission evaluations. Changes to light sources are fairly straightforward in case they are assumed to be producing unpolarised light. Changes to the BSDF are slightly more complicated, but the appropriate Mueller matrices for various surface models for which the matrix are known and available. However, as there is no polarised version of the Phong model, we implemented the Torrance-Sparrow model in our test renderer, in order to have a BRDF model for glossy surfaces.

**Polarisation-capable uni-directional path tracer.** In this simple case, the path throughput takes the form of a matrix with which a Mueller matrix for each surface hit is multiplied. After we reach the light source, we compute the resulting Stokes vector and rotate it according to the sensor orientation. All other variables, fields or arguments have to be changed accordingly for the operations to be valid. When multiplications are done, it may be necessary to reorder the terms to reach eligible operations, as the non-polarised implementation could have assumed commutativity. Because we used operator overloading in our implementation, this step turned out to be very straightforward, as all places that had to be altered were pointed out by the compiler itself.

### 5.2. Polarisation-Capable Bi-directional Path Tracer

**Importance.** We use the matrix importance formulation outlined in the theory section in order to compute the full Stokes vector measurements. Note that the theory uses transposed importance $W^T$ in all occurrences with the exception of the dual operator $K^*$. As we are building upon the path integral formulation, it is much simpler to directly store the transposed importance as that is the term that is used in computation and there is no use for non-transposed importance anywhere in the implementation.

**Separate data type.** Although importance in a form of a matrix might seem as just another instance of a Mueller matrix, the key difference lies in the fact that as opposed to the two reference frames that are stored with Mueller matrices, importance defines only one. This fits with the usage of importance, as is should not be possible to multiply importance and a Mueller matrix from the other side. Also, the product of multiplying a Stokes vector of radiance with importance does not produce an actual Stokes vector, only four separate measurements. Due to this, it is desirable to define a separate structure that represents importance. In our implementation, it is represented by the `Importance` structure. The last operation not yet mentioned is the multiplication of `Importance` and `Light` to produce a measurement. As mentioned, that operation should not just produce another Stokes vector, but rather a measurement value, which in our case would be a generic vector of 4 colour values that is then stored in the resulting image.

**Reference frames.** When defining reference frames for Mueller matrices, it is important to differentiate which frame is the enter and which is the exit frame according to the actual direction of light propagation. That direction may be different from the supplied incident and outgoing directions that would be based on the direction of path generation, unlike regular path tracer, where the mapping is always the same. Note that this would not be helped, if we would have changed the orientation of the frames to both point outwards from the scattering location, as the order in which the resulting Mueller matrix is multiplied with a Stokes vector, or with other Mueller matrix along the path, is still fixed and as such the enter/exit frames have to be set up appropriately.

**Polarisation-capable bi-directional path tracing.** Similar to the modification of a standard path tracer outlined above, the changes that have to be made to the algorithm are straightforward after all of the data types are properly set up. In this case, the only new aspect is the definition of importance. A proper importance matrix has to be defined whenever a path ends (when tracing from a light source) or starts (when tracing from a sensor). That matrix is an identity matrix multiplied with whatever the importance value for a standard computation would be. The associated reference frame is set up to point towards the sensor (as that would be the direction of light propagation) with the other directions based on the "up" direction of the sensor as outlined in section 3.

**Backporting importance to the uni-directional path tracer.** The concept of importance is useful enough to include it even in a uni-directional path tracer, as it properly abstracts the functionality of throughput as a concatenation of Mueller matrices, and the final rotation of the measured radiance to the reference frame of the camera.

### 5.3. Polarisation-Capable Photon Mapping

**Vertex merging.** As VCM is a combination of bi-directional path tracing and photon mapping, what remains to cover is how photon mapping can be made polarisation-capable. As photon tracing corresponds to the light tracing portion of BPT, it is sufficient to just talk about the photon map itself. The only change that actually needs to happen is exchanging the type representing radiance that is stored along the photon to a Stokes vector `Light`.

**Validity of a photon lookup.** The process of averaging radiance over photons amounts to the addition of Stokes vectors and multiplication by scalar value, which are both possible if all photon vectors point in the same direction. Considering that each photon, before being averaged, is multiplied by the BSDF element computed for a specific incident direction $\omega_i$, which is the same for every photon, all of the resulting Stokes vectors end up being defined as radiance in direction $-\omega_i$. They might not share the exact same coordinate system, but the outlined implementation of data types should properly handle such a case.

**Glossy materials.** Note that it is not necessary to store entire Stokes vectors in the photon map, if the only surface BRDF models in the scenes are ideally specular and ideally diffuse, as photons are not stored on specular surfaces and diffuse surfaces are depolarisers. In such case, any photons that would be stored would then lose all polarisation information. In our implementation, we added the Torrance-Sparrow model so that the functionality of photon mapping could be properly examined.

### 5.4. A Polarisation-Capable Volumetric Path Tracer

**Polarising media.** Similarly to how regular path tracer was made polarisation-capable, attenuation coming from participating media has to be modified to yield a Mueller matrix instead of a single attenuation value, which is a scalar multiple of an identity matrix. Note that for such matrices the choice of reference frames does not matter, as long as both of them are chosen to be in the direction of light propagation. The scattering events can be altered in the same way the BSDF was altered. Rayleigh scattering is known to be produce polarised light, while Mie scattering can be for rendering purposes considered to be a de-polariser.

**Polarisation-capable volumetric path tracers.** Afterwards, the changes that need to be made to the volumetric algorithms are analogous to those done in the non-volumetric case. Note that some of the evaluation of attenuation by media might require information about the direction of light propagation, even though it was not previously necessary. This problem goes further than just storing the appropriate reference frames in the correct field (enter or exit) as it was with the BSDF. If the attenuation over a given path can be composed from multiple attenuation elements that are concatenated separately from the actual volumetric algorithm, then such concatenation must know the direction of light in order to multiply the Mueller matrices in correct order.

### 5.5. Optimisations

**Storage of path vertices.** Bidirectional path-tracing can be implemented by either storing light vertices and connecting them to camera vertices or vice versa. For camera paths, we need to store the computed importance, which in our case is a $4 \times 4$ matrix, while the radiance that would be stored for light paths, is only a 4 component vector. As such, it may be more beneficial to use the variant of storing light vertices in order to cut down on memory consumption and possible performance decrease caused by manipulating with larger data structures. The BPT implementations in SmallUPBP already stored light vertices, as the same code was reused for Vertex Connection and Merging, so in our case this optimisation was done implicitly just by making the code polarising.

**Differentiating edge cases.** There are various Mueller matrices that we can encounter during rendering. However, some of them pop up more often than others – and those are depolarisers (from diffuse surfaces) and plain attenuations over a ray (for volumetric pathtracing). It is not necessary to store them in their full matrix form, as they are fully described by one colour value. In order to use homogenous data type for all, we might still reserve space for the full matrix, but we do not need to initialize or further touch the remaining space, as long as we store a flag identifying the type of the matrix. Similarly, non-polarised light can also be seen as a special case that can be store and worked with more efficiently. As we are already in the process of distinguishing different types of Stokes vectors and Mueller matrices, it may also be useful to add a zero attenuation and zero vector to the list so that we have a neutral element for addition of Stokes vectors and Mueller matrices. Otherwise, we might have to deal with setting up the reference frames of a zeroed Mueller matrix to be collinear with the expected result.

With these special matrix representations, large numbers of full matrix-matrix multiplications can be replaced with scalar-matrix multiplications or even scalar-scalar multiplications. This leads to a reduction of the polarisation performance penalty, as exactly these matrix operations are the main difference from the non-polarising case. Additionally, some special forms of Mueller matrices and Stokes vectors are invariant to the choice of reference frames, so it is not necessary to perform rotations on them, in order to match the frames during multiplications or additions.

In our implementation, an `Attenuation` structure can be flagged as being either:

- *zero* – stored as only a flag
- *plain* – stores only 1 colour value
- *depolarising* – stores only 1 colour value
- *depolarising enter* – result of multiplying standard Mueller matrix and depolarising `Attenuation`, stores only 4 colour values, only exit reference frame is valid
- *depolarising exit* – result of multiplying a depolarising `Attenuation` and standard Mueller matrix, stores only 4 colour values, only enter reference frame is valid
- *standard* – regular full Mueller matrix with both of its reference frames

The `Light` structure can be flagged as being either zero, non-polarised, or standard.

### 5.6. Results

For generating the result images and measuring the time taken, we ran our tests on a machine with Intel i5-2500 CPU with four cores at 3.3 GHz and 8 GB RAM. As with normal SmallUPBP, Embree was used for ray-scene intersections.

**Measured variants.** For our efficiency comparisons, we evaluated several different variants of SmallUPBP with different configurations:

- `orig` – the original implementation
- `nonpol` – non-polarising variant of the altered code
- `naive` – initial simple implementation of polarised light transport
- `opti` – an optimised implementation of polarisation as described above

The `orig` variant is included in order to anchor our experiments with respect to a known and publicly available codebase. The `nonpol` variant should be the one that is considered to be the reference for measuring the impact of polarised light transport, as it shares the rest of the infrastructure that was altered from `orig` by implementing certain features deemed necessary for testing polarised light transport (beside the polarisation support itself). Note, that as `orig` doesn't support the Torrance-Sparrow BRDF model, it uses the Phong model for rendering glossy surfaces instead.

**Measured algorithms.** We tested the following light transport algorithms in surface-only transport:

- `PT` – Path Tracing with next-event estimation
- `VCM` – Vertex Connection and Merging

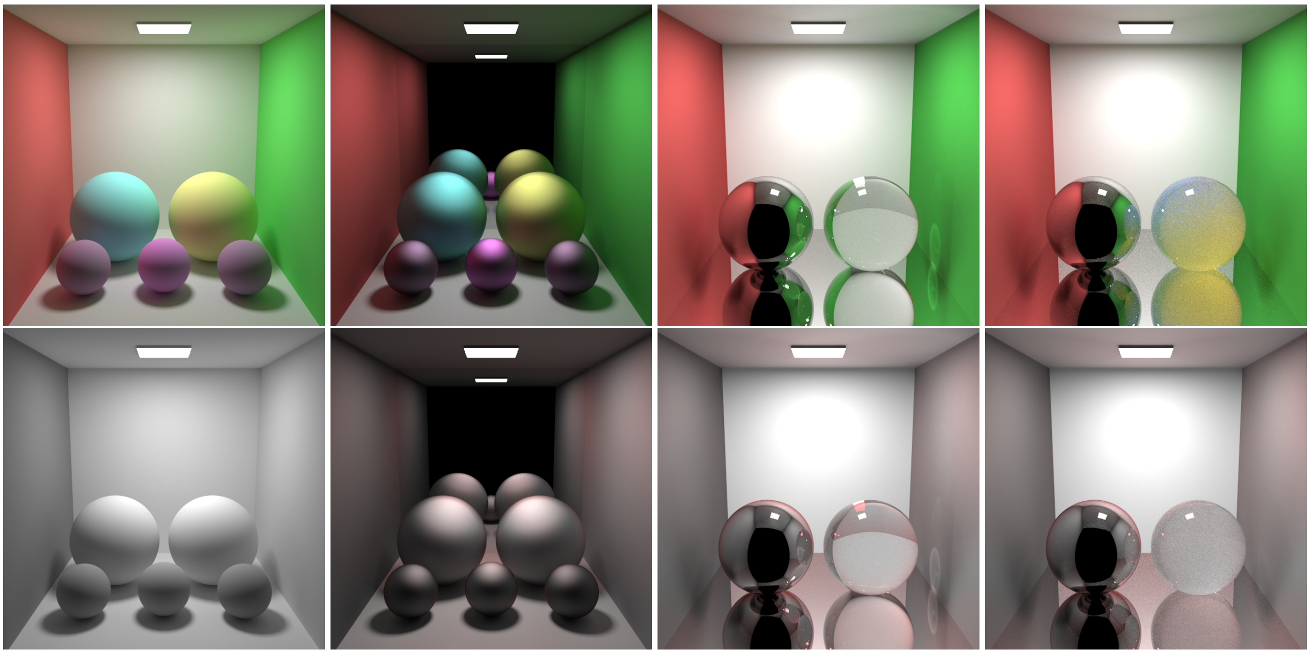

**Figure 1:** *The top row shows the intensity of computed light, bottom row shows the scene in greyscale with degree of polarisation overlaid in red. Scenes from left to right:* `diffuse`*,* `glossy`*,* `glass` *and* `glass` *with participating media enabled.*

For participating media, we ran tests with these volumetric light transport algorithms:

- `VPT` – Volumetric Path Tracing with next-event estimation
- `VBPT` – Volumetric Bidirectional Path Tracing

**Test scenes.** We used the following scenes in our tests:

- `diffuse` – Cornell box with diffuse walls and five diffuse spheres of various colours. There is not polarisation in the scene.
- `glossy` – Cornell box with glossy walls and five glossy spheres of various colours. Every surface is polarising in some way.
- `glass` – Cornell box with a mirror floor, glossy left, right walls and ceiling and a diffuse back wall with a mirror sphere and a glass sphere. For participating media, the glass sphere contains a very dense atmosphere showing off polarised Rayleigh scattering.
- `bathroom` – Complex bathroom scene showcased previously with SmallUPBP, now altered to include a complex IOR for mirror surfaces and to show better results with regular path tracer.

**Performance.** Table 1 shows the measured results for various combinations of scenes and algorithms with each measured variant. The values given are only comparable in each row separately, as different setups may produce different results due to the complexity of the scene and efficiency of algoritms themselves. Not all combinations are present, as the impact of each variant on different algorithms can be gathered from the presented subset.

**Impact of modifications.** Even though a number of changes were made, specifically the introduction of a complex index of refraction, more complex Fresnel term calculations, a Torrance-Sparrow BRDF model for glossy surfaces instead of a Phong model and a framebuffer capable of storing both non-polarised and polarised results, the altered code seems to be outperforming the original with the only exception being the volumetric `bathroom`. As the original SmallUPBP was never properly optimised, it can be explained

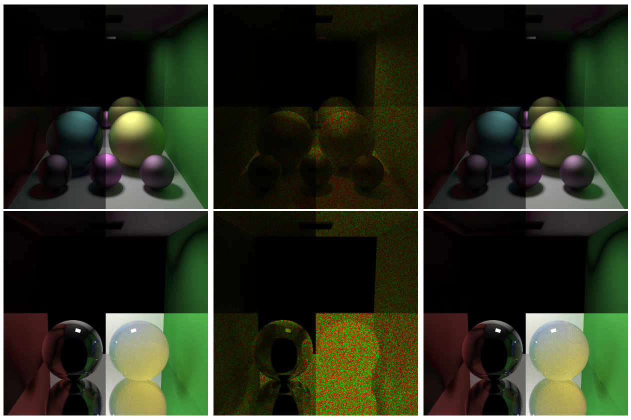

**Figure 2:** *Our bi-directional polarised light transport algorithms converge to the same results as polarised unidirectional path tracing. Each quadrant of the images shows one of the Stokes component, from top to bottom, left to right: $S_3$, $S_1$, $S_2$, $S_0$. Left image rendered with* `PT` *(top) or* `VPT` *(bottom), right image rendered with* `VCM` *(top) or* `VBPT` *(bottom) and the middle image shows the difference between the two. Top row shows scene* `glossy` *and bottom row shows scene* `glass`*.*

| scene+algorithm | orig | nonpol | naive | opti |
|---|---|---|---|---|
| diffuse +PT | 14520 | 14001 | 9129 | 12425 |
| glossy +PT | 12076 | 18263 | 13728 | 15074 |
| glossy +VCM | 3354 | 5839 | 4533 | 4859 |
| glass +PT | 11883 | 15665 | 10642 | 13638 |
| glass +VCM | 4105 | 5540 | 3388 | 4732 |
| glass +VPT | 7372 | 8401 | 2445 | 6477 |
| glass +VBPT | 2855 | 4020 | 1097 | 2589 |
| bathroom+VCM | 628 | 626 | 420 | 524 |
| bathroom+VBPT | 561 | 376 | 165 | 267 |

**Table 1:** *The number of iterations that were run for the specified combination of an algorithm and a scene for the different algorithm variants when rendering images at* $256 \times 256$ *pixels for 30 minutes.*

through inadvertent optimisation of some portions of the code when it was being modified to be capable of running in polarisation mode.

**Slowdown of naive approach.** In the naive implementation, we only replaced the types used within the algorithms with Stokes vectors and Mueller matrices and occasionally repaired the ordering of terms in computations. The memory consumption increase as a result of that is unavoidable. As that approach leads to a high number of matrix multiplications, which includes frame rotations in the form of additional matrix multiplications, it is expected that we would observe some sort of a slowdown. In the case of `PT` and `VCM`, that results in a slowdown to approx. 65-75 % of `nonpol`. However, in the case of volumetric algorithms `VPT` and `VBPT`, the slowdown reaches approx. 30 %, of `nonpol`, as there is a greater amount of matrix multiplications conducted when attenuation of a medium is computed, even when the medium is clear and results in an identity matrix.

**Efficiency of optimizations.** The optimisations do result in a performance increase, decreasing the slowdown of `PT` and `VCM` to approx. 85 % of `nonpol`. The volumetric algorithms `VPT` and `VBPT` achieve even higher reduction to approx. 60 % of `nonpol`, which is still higher slowdown than for non-volumetric algorithms, but it can be expected as the number of attenuation concatenations is greater. The efficiency of distinguishing individual attenuation types varies depending on the amount of depolarisers in the scene – `diffuse` has no polarising surfaces, while `glossy` has no depolarising surfaces. But we do observe speedup even for scene `glossy`, as one of the distinguished matrices was the zero matrix, thanks to which initialising and passing it around is done much more efficiently. Note that importance, even thought it start off as only a scalar multiple of an identity matrix, still has to keep its reference frame and as such is treated as a full matrix. As such, operations with full matrices cannot be completely eliminated.

## 6. Conclusion

We have presented a theoretical treatment of how modern rendering techniques can be made to properly work with polarised light. We also presented results from our reference implementation of these findings, discussed optimisation strategies that can be applied to polarisation rendering systems, and analysed the performance of our test implementation.

## References


[CPLB14] COLLIN C., PATTANAIK S. N., LIKAMWA P., BOUATOUCH K.: Computation of polarized subsurface BRDF for rendering. In *Graphics Interface* (2014), Kry P. G., Bunt A., (Eds.), Canadian Information Processing Society / ACM, pp. 201–208. URL: http://dl.acm.org/citation.cfm?id=2619648. 2

[GCP*10] GHOSH A., CHEN T., PEERS P., WILSON C. A., DEBEVEC P. E.: Circularly polarized spherical illumination reflectometry. *ACM Trans. Graph 29*, 6 (2010), 162. URL: http://dx.doi.org/10.1145/1882261.1866163. 2

[GFT*11] GHOSH A., FYFFE G., TUNWATTANAPONG B., BUSCH J., YU X., DEBEVEC P. E.: Multiview face capture using polarized spherical gradient illumination. *ACM Trans. Graph 30*, 6 (2011), 129. URL: http://doi.acm.org/10.1145/2070781.2024163. 2

[GKDS12] GEORGIEV I., KRIVÁNEK J., DAVIDOVIC T., SLUSALLEK P.: Light transport simulation with vertex connection and merging. *ACM Trans. Graph 31*, 6 (2012), 192. URL: http://doi.acm.org/10.1145/2366145.2366211. 6

[Gol03] GOLDSTEIN D.: *Polarized Light*. Marcel Dekker Inc., New York, USA, 2003. 2, 3

[HPJ12] HACHISUKA T., PANTALEONI J., JENSEN H. W.: A path space extension for robust light transport simulation. *ACM Trans. Graph. 31*, 6 (2012), 191. URL: http://doi.acm.org/10.1145/2366145.2366210, doi:10.1145/2366145.2366210. 6

[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 (Aug. 2014), 1–13. 6

[KKG*14] KŘIVÁNEK J., KELLER A., GEORGIEV I., KAPLANYAN A., FAJARDO M., MEYER M., NAHMIAS J.-D., KARLÍK O., CANADA J.: Recent advances in light transport simulation: Some theory and a lot of practice. In *ACM SIGGRAPH 2014 Courses* (New York, NY, USA, 2014), SIGGRAPH '14, ACM, pp. 17:1–17:6. URL: http://doi.acm.org/10.1145/2614028.2615438, doi:10.1145/2614028.2615438. 1

[LSG12] LATORRE P., SERÓN F. J., GUTIERREZ D.: Birefringence: calculation of refracted ray paths in biaxial crystals. *The Visual Computer 28*, 4 (2012), 341–356. URL: http://dx.doi.org/10.1007/s00371-011-0619-2. 2

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional Path Tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)* (Alvor, Portugal, Dec. 1993), Santo H. P., (Ed.), pp. 145–153. 6

[MTHI03] MIYAZAKI D., TAN R. T., HARA K., IKEUCHI K.: Polarization-based inverse rendering from a single view. In *ICCV* (2003), pp. 982–987. URL: http://dx.doi.org/10.1109/ICCV.2003.1238455. 2

[SML*12] SADEGHI I., MUNOZ A., LAVEN P., JAROSZ W., SERON F., GUTIERREZ D., JENSEN H. W.: Physically-based simulation of rainbows. *ACM Transactions on Graphics 31*, 1 (Jan. 2012), 3:1–3:12. doi:http://dx.doi.org/10.1145/2077341.2077344. 2

[TTW94] TANNENBAUM D. C., TANNENBAUM P., WOZNY M. J.: Polarization and birefringency considerations in rendering. *Computer Graphics 28*, Annual Conference Series (July 1994), 221–222. URL: http://www.acm.org:80/pubs/citations/proceedings/graph/192161/p221-tannenbaum/. 2

[Vea97] VEACH E.: *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, 1997. 4, 5, 6

[WK90] WOLFF L. B., KURLANDER D.: Ray tracing with polarization parameters. *IEEE Computer Graphics & Applications 10*, 6 (Nov. 1990), 44–55. 2

[WTP01] WILKIE A., TOBLER R. F., PURGATHOFER W.: Combined rendering of polarization and fluorescence effects. In *Proceedings of*

**Figure 3:** *Bathroom scene. Left: intensity of the computed light, right: degree of polarisation overlaid over a grayscale image*

*the 12th Eurographics Workshop on Rendering Techniques, London, UK, June 25-27, 2001* (2001), Gortler S. J., Myszkowski K., (Eds.), Springer, pp. 197–204. 2

[WUT*04] WILKIE A., ULBRICHT C., TOBLER R. F., ZOTTI G., PURGATHOFER W.: An analytical model for skylight polarisation. In *Eurographics Symposium on Rendering* (Norrköping, Sweden, 2004), Keller A., Jensen H. W., (Eds.), Eurographics Association, pp. 387–397. URL: http://www.eg.org/EG/DL/WS/EGWR/EGSR04/387-397.pdf. 2

[WW07] WEIDLICH A., WILKIE A.: Realistic rendering of birefringency in uniaxial crystals. *ACM Transactions on Graphics accepted for publication* (2007). 2

[WW10] WILKIE A., WEIDLICH A.: A Standardised Polarisation Visualisation for Images. In *Proceedings of the Spring Conference on Computer Graphics 2010 (SCCG 2010)* (May 2010). 2

[WW11] WILKIE A., WEIDLICH A.: A Physically Plausible Model for Light Emission from Glowing Solid Objects. In *Eurographics Symposium on Rendering* (Prague, Czech Republic, 2011), Reinhard E., Ramamoorthi R., (Eds.), Eurographics Association. 2

[WW12] WILKIE A., WEIDLICH A.: Polarised light in computer graphics. In *SIGGRAPH Asia 2012 Courses* (New York, NY, USA, 2012), SA '12, ACM, pp. 8:1–8:87. URL: http://doi.acm.org/10.1145/2407783.2407791, doi:10.1145/2407783.2407791. 1

[WWG12] WILKIE A., WEIDLICH A., GHOSH A.: Polarised light in computer graphics. In *SIGGRAPH Asia 2012 Course Notes* (2012). 2, 6

**Appendix A:** Stokes, Mueller and Importance spaces.

Here we formally define the spaces $\mathbb{S}_\omega$,$\mathbb{M}_{\omega_i}^{\omega_o}$ and $\mathbb{I}_\omega$.

First we define the space $S_\omega$ of all pairs a Stokes vector and its coordinate system

$$S_\omega =_{\text{def}} \mathbb{R}^4 \times \mathbb{F}_\omega$$

On this space we define non-commutative addition and scalar multiplication

$$(S,F)+(T,G) =_{\text{def}} (S+FG^{-1}T,F)$$
$$\lambda(S,F) =_{\text{def}} (\lambda S,F) \qquad \lambda \in \mathbb{R}$$

In addition we define an equivalence relation $\sim$, two pairs $(S,F),(T,G) \in S_\omega$ are equivalent if

$$(S,F) \sim (T,G) \overset{\text{def}}{\iff} (S,F) = (FG^{-1}T,F)$$

The Stokes space $\mathbb{S}_\omega$ is just a quotient space of $S_\omega$

$$\mathbb{S}_\omega =_{\text{def}} S_\omega /_\sim$$

Operations on $S_\omega$ induce addition and scalar multiplication on $\mathbb{S}_\omega$ and the addition becomes commutative, for $[S,F],[T,G] \in \mathbb{S}_\omega$

$$[S,F]+[T,G] = [S+FG^{-1}T,F] = [T+GF^{-1}S,G] = [T,G]+[S,F]$$

With these operations the space $\mathbb{S}_\omega$ forms a vector space and if we choose arbitrary coordinate system $F \in \mathbb{F}_\omega$ then the elements $[e_0,F],[e_1,F],[e_2,F],[e_3,F]$ form a basis therefore $\mathbb{S}_\omega$ is isomorphic to $\mathbb{R}^4$. We also define scalar product on the Stokes space

$$\langle [S,F],[T,G]\rangle_{\mathbb{S}_\omega} =_{\text{def}} S^T FG^{-1}T$$

Mueller space $\mathbb{M}_{-\omega_i}^{\omega_o}$ is nothing else then the space of all linear transformations from the space $\mathbb{S}_{-\omega_i}$ to the space $\mathbb{S}_{\omega_o}$. Alternatively, the Mueller space can be also constructed as a quotient space.

$$\mathbb{M}_{-\omega_i}^{\omega_o} =_{\text{def}} \left(\mathbb{R}^{4\times 4} \times \mathbb{F}_{-\omega_i} \times \mathbb{F}_{\omega_o}\right) /_\sim$$

where the equivalence relation is defined as

$$(M, F_\mathrm{i}, F_\mathrm{o}) \sim (N, G_\mathrm{i}, G_\mathrm{o}) \overset{\text{def}}{\iff} (M, F_\mathrm{i}, F_\mathrm{o}) = (F_\mathrm{o} {G_\mathrm{o}}^{-1} N G_\mathrm{i} {F_\mathrm{i}}^{-1}, F_\mathrm{i}, F_\mathrm{o})$$

It can be shown that these two definitions coincides. The first one is useful to realize what the Mueller space does and the second definition is useful for representing its elements. A Mueller matrix $[M, F_\mathrm{i}, F_\mathrm{o}] \in \mathbb{M}^{\omega_\mathrm{o}}_{-\omega_\mathrm{i}}$ acts on a Stokes vector $[S, F] \in \mathbb{S}_{-\omega_\mathrm{i}}$ as

$$[M, F_\mathrm{i}, F_\mathrm{o}][S, F] =_{\text{def}} [M F_\mathrm{i} F^{-1} S, F_\mathrm{o}].$$

The sum of Mueller matrices $[M, F_\mathrm{i}, F_\mathrm{o}], [N, G_\mathrm{i}, G_\mathrm{o}] \in \mathbb{M}^{\omega_\mathrm{o}}_{-\omega_\mathrm{i}}$ does not depend on the order

$$[M, F_\mathrm{i}, F_\mathrm{o}] + [N, G_\mathrm{i}, G_\mathrm{o}] = [M + F_\mathrm{o} {G_\mathrm{o}}^{-1} N G_\mathrm{i} {F_\mathrm{i}}^{-1}, F_\mathrm{i}, F_\mathrm{o}] =$$
$$[N + G_\mathrm{o} {F_\mathrm{o}}^{-1} M F_\mathrm{i} {G_\mathrm{i}}^{-1}, G_\mathrm{i}, G_\mathrm{o}] = [N, G_\mathrm{i}, G_\mathrm{o}] + [M, F_\mathrm{i}, F_\mathrm{o}]$$

The Importance space $\mathbb{I}_\omega$ is just a dual space of the Stokes space $\mathbb{S}_\omega$.

$$\mathbb{I}_\omega =_{\text{def}} \mathbb{S}^*_\omega$$

It can be also defined as a quotient space

$$\mathbb{I}_\omega =_{\text{def}} \left(\mathbb{R}^{1\times 4} \times \mathbb{F}_\omega\right) \big/ _\sim$$

where the equivalence is defined as

$$(w^T, F) \sim (z^T, G) \overset{\text{def}}{\iff} (w^T, F) = (z^T G F^{-1}, F)$$

The action of an element $[w^T, F] \in \mathbb{I}_\omega$ on a Stokes vector $[S, G] \in \mathbb{S}_\omega$ is

$$[w^T, F][S, G] =_{\text{def}} w^T F G^{-1} S$$

Thanks to the inner product on Stokes space we can identify $\mathbb{I}_\omega$ with $\mathbb{S}_\omega$ and the action of $[w^T, F]$ on $[S, G]$ can be done with the inner product

$$\langle [w, F], [S, G] \rangle = \langle [w, F], [S, G] \rangle_{\mathbb{S}_\omega}$$