

Physically-Based Simulation of Rainbows

IMAN SADEGHI

UC San Diego

ADOLFO MUNOZ

Universidad de Zaragoza

PHILIP LAVEN

Horley, UK

WOJCIECH JAROSZ

Disney Research Zürich

FRANCISCO SERON and DIEGO GUTIERREZ

Universidad de Zaragoza

HENRIK WANN JENSEN

UC San Diego

In this paper we derive a physically based model for simulating rainbows. Previous techniques for simulating rainbows have used either geometric optics (ray tracing) or Lorenz-Mie theory. Lorenz-Mie theory is by far the most accurate technique as it takes into account optical effects such as dispersion, polarization, interference, and diffraction. These effects are critical for simulating rainbows accurately. However, as Lorenz-Mie theory is restricted to scattering by spherical particles, it cannot be applied to real raindrops which are non-spherical, especially for larger raindrops. We present the first comprehensive technique for simulating the interaction of a wavefront of light with a physically-based water drop shape. Our technique is based on ray tracing extended to account for dispersion, polarization, interference, and diffraction. Our model matches Lorenz-Mie theory for spherical particles, but it also enables the accurate simulation of non-spherical particles. It can simulate many different rainbow phenomena including double rainbows and supernumerary bows. We show how the non-spherical raindrops influence the shape of the rainbows, and we provide a simulation of the rare twinned rainbow, which is believed to be caused by non-spherical water drops.

Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation—; I.3.5 [Computer Graphics]: 3D Graphics and Realism—

General Terms: Rainbows

Additional Key Words and Phrases: Lorenz-Mie theory, optical properties, phase function, appearance modeling, realistic rendering.

1. INTRODUCTION

Rainbows are among the most visually stunning phenomena in nature. They are caused by the interaction of sunlight with small water drops in the atmosphere, and they appear in the form of multicolored arcs. The appearance of rainbows can vary significantly

depending on the lighting conditions and the raindrop distribution. Figure 2 shows examples of a full double rainbow, a close-up showing the darkening of Alexander's band, supernumerary bows and a rare twinned rainbow.

Even though the study of rainbows can be traced back more than two thousand years [Lee and Fraser 2001], they are still not fully understood. For instance, twinned rainbows (which strangely are visible on the primary bow but not the secondary, as seen in Figure 2f and 17), are believed to appear due to water drops not being perfect spheres, but there is no solid theory to confirm this.

The first studies of rainbows assumed simple geometric optics where light is refracted as it enters or leaves the water drop. This model can explain the basic primary and double rainbow configuration, but it fails at explaining supernumerary bows that are caused by interference. To account for interference it is necessary to consider the sunlight as a wavefront interacting with the raindrop. This can be accomplished using Lorenz-Mie theory, which accounts for reflection, refraction, dispersion, polarization, interference, and diffraction, and it turns out that all of these optical effects are necessary to accurately simulate rainbows. Unfortunately, Lorenz-Mie theory is limited to spherical water drops, and this not only yields wrong predictions in some cases, but it ultimately limits the types of rainbows that can be explained as well. It is well-known that water drops become non-spherical as they get larger, and this heavily influences the distribution of the scattered light. Unfortunately, there is no theory available that can explain the consequence of physical water drops, and this is one of the reasons why rainbows continue to be an active research area.

In this paper we develop the first comprehensive model for rainbows in computer graphics. We explain the optical events that cause rainbows, and we develop an accurate ray tracing algorithm that accounts for the full spectrum of optical effects including dispersion, polarization, interference, and an efficient approximation for diffraction. We show how our model matches the results of Lorenz-Mie theory for spherical water drops, and how it extends to also account for non-spherical water drops. The result is the first accurate simulation of sunlight scattered by water drops of realistic non-spherical shape. We show how even a slight variation in the raindrop shape gives rise to changes in the position and shape of the rainbow. Our simulation can be used to explain the strange appearance of twinned rainbows, as well as the more common double

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

© 2011 ACM 0730-0301/2011/11-ART— \$10.00

DOI 10.1145/——.——

<http://doi.acm.org/10.1145/——.——>

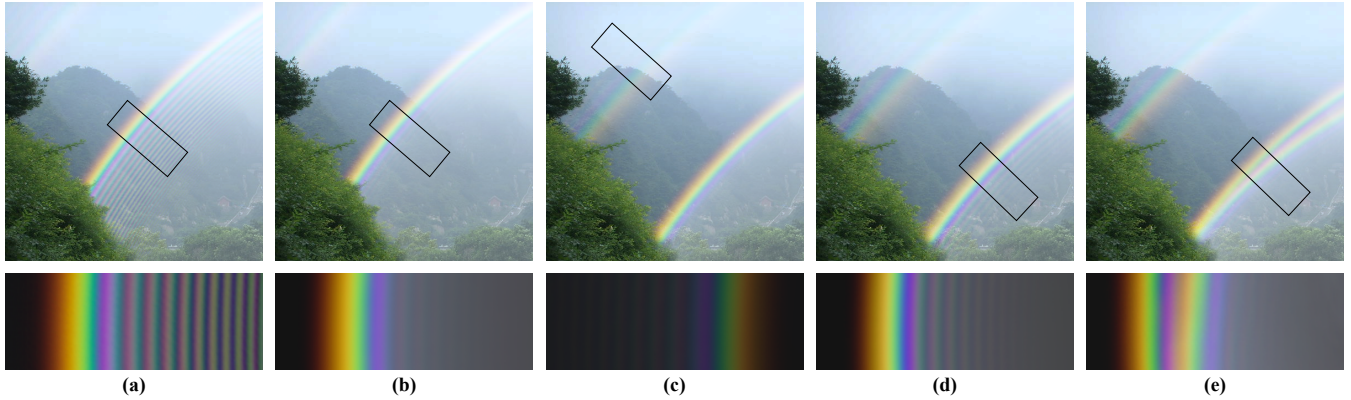


Fig. 1: Our rendering results for different types of rainbows: **(a)** Rainbow derived from Lorenz-Mie theory. **(b)** Single primary rainbow with considering the angular view of the sun. **(c)** Double rainbow with a flipped secondary rainbow. **(d)** Multiple supernumerary rainbows caused by small water drops with uniform sizes. **(e)** Twinned rainbow resulted from mixture of non-spherical water drops and spherical ones.

rainbows and supernumerary arcs (see Figure 1). We also provide a database of tabulated phase functions to the academic community¹.

Our work falls in the domain of precise light simulation beyond the traditional limitations of pure geometric optics. Similar approaches have been undertaken before in graphics, for instance in the field of gem modeling [Guy and Soler 2004] or, the simulation of interference in thin layers [Gondek et al. 1994]. While we focused here on rainbows, we believe the model we develop can be used for other phenomena involving scattering by small particles (for example halos). Furthermore, accurate predictive rendering models of atmospheric phenomena, like the one we present, can have wider-reaching impact in areas such as meteorology e.g., by providing a key component in deducing the size of water drops from photographs [Narasimhan and Nayar 2003].

2. PREVIOUS WORK

Rainbows have traditionally been considered a fascinating topic, from scientists to philosophers, and are arguably one of the most beautiful displays of nature [Greenler 1990; Minnaert 1993; Lynch and Livingston 2001]. Different theories have been developed over the centuries, and some of them have been adopted by the computer graphics community to simulate rainbows with varying degrees of realism. Some techniques are based on a simplification of the process, in order to achieve interactive frame rates, while others present rainbow simulations in the context of atmospheric modeling. However, the complete (and quite complex) physics of rainbow formation has not been fully researched in the field of computer graphics.

Simplified solutions include the work by Musgrave [1989], which follows Descartes’ model from a classical geometric optics perspective. Frisvad and colleagues [2007] presented a real-time simulation using Aristotle’s rainbow formation theory based on reflections in clouds. Although these models may provide intuitive explanations about rainbow formation, geometric optics by itself fails to capture more complex aspects such as supernumerary arcs (see Figure 2e).

Lorenz-Mie theory [Lorenz 1890; Mie 1908] provides an exact solution for scattering by spherical particles in non-absorbing media. Given its computational complexity, it was not deemed useful

until van de Hulst published results as tabulated data [1957]. Unfortunately, this work is limited to very small spheres, and thus not directly suitable for rainbows. This theory was later introduced to the graphics community by Rushmeier [1995], and was used recently to compute scattering properties of different materials [Frisvad et al. 2007].

Lee [1998] investigated the differences between results obtained using Lorenz-Mie theory and Airy theory [Airy 1838], including perceptual issues. Jackel and Walter [1997] simulate rainbows by adding a rain layer to the atmosphere and making use again of Lorenz-Mie theory to compute phase functions for single scattering. In their work, raindrop sizes follow a normal distribution. A similar approach with a log-normal distribution was introduced by Riley et al. [2004], who achieve interactive frame rates with simplified lighting models. Phase functions are obtained based on the work by Laven [2003], which implements the algorithm from Bohren and Huffman [1983] to obtain scattered intensities. A simplified, texture-based GPU implementation has also been developed [nVIDIA 2004]. Recently, Gedzelman [2008] explored the influence of the atmospheric environment on the appearance of rainbows; although valid conclusions on overall brightness and visibility were reached, the results did not aim to be photorealistic.

Most of these approaches are based on Lorenz-Mie theory which, unfortunately, can only provide an accurate solution in the case of spherical water drops. However, real water drops diverge from perfect spheres due to the combined effects of gravity and surface tension [Beard and Chuang 1987; Beard et al. 1991; Brangi et al. 1991; Villermaux and Bossa 2009]. This translates into inaccurate simulations in the best case, and the impossibility to simulate certain effects like the twinned rainbow in the worst case. In this paper, we introduce a novel algorithm based on a ray-tracing approach, which for the first time matches the predictions of Lorenz-Mie theory for the ideal case of spherical water drops, but naturally generalizes to handle actual, real-world water drop geometries. This allows us to produce excellent simulations of rainbows, while extending the validity of such simulations to include scattering from non-spherical drops of water.

The most practical solution for computing the scattering properties of particles with arbitrary shapes, are the finite difference time domain (FDTD) methods [Yee 1966; Taflovie 1998]. These methods can simulate Maxwell’s time-dependent equations on a discrete lattice in order to compute the behaviour of the electromag-

¹<http://graphics.ucsd.edu/~iman/Rainbows/>

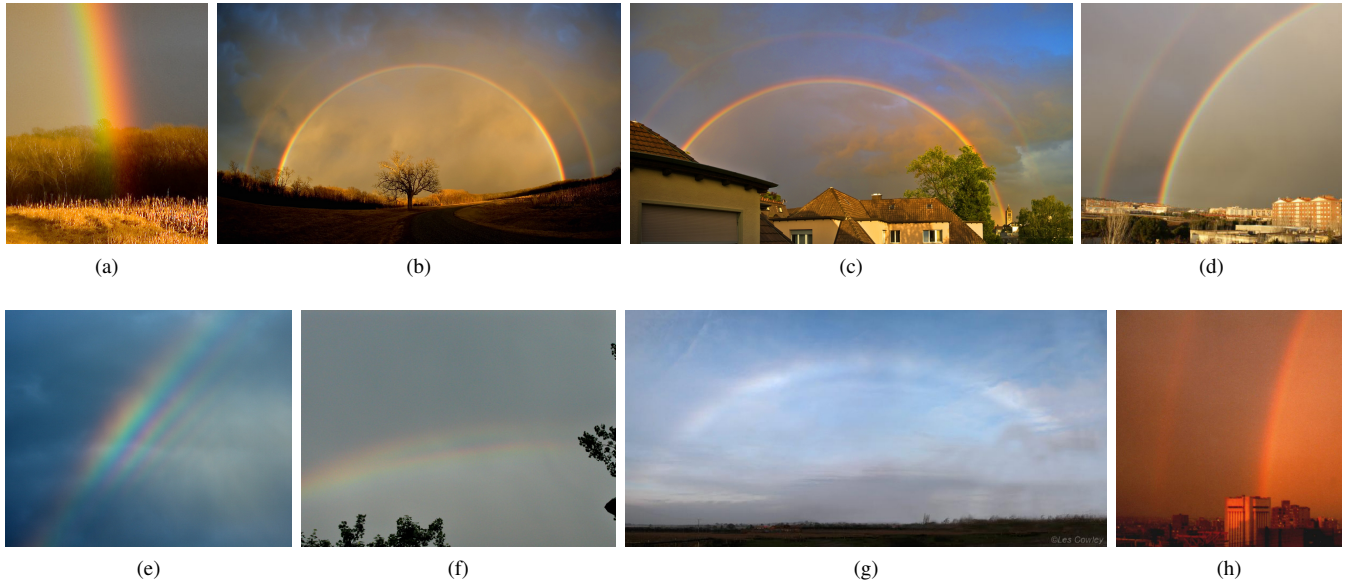


Fig. 2: Different rainbows seen in nature: **(a)** primary rainbow, **(b)** double rainbow (both reproduced with permission ©Dan Bush - http://www.missouriskies.org/rainbow/february_rainbow_2006.html), **(c)** double rainbow, **(d)** Alexander's dark band, **(e)** multiple supernumerary bows (reproduced with permission ©Ian Goddard - <http://www.atoptics.co.uk/rainbows/bowim46.htm>), **(f)** twinned rainbow (reproduced with permission ©Benjamin Kuehne - <http://www.nachtwolke.de/temp/regenbogen2.htm>), **(g)** cloud bow (reproduced with permission ©Les Cowley - <http://www.atoptics.co.uk/rainbows/cldbow.htm>), and **(h)** red bow.

netic wave. These brute-force methods have been used to simulate the light scattering behaviour of complex objects [Umashankar and Taflov 1982] as well as ice crystals [Yang and Liou 1995; Yang and Liou 1996]. However, they are computationally very expensive for three-dimensional grids and can take days on multi-core processors. Furthermore, the generalizations of Lorenz-Mie theory to non-spherical particles by Frisvad et al. [2007] cannot be used as it only applies to the computed scattering cross-section, while the appearance of rainbows are caused by variations in the angular scattering profile (the phase function).

There has been other work focusing on the light scattering by non-spherical particles [Mishchenko et al. 2000; Xu et al. 2010] but none of them focus on the physically-based geometries of water drop particles and their effect on the appearance of rainbows. To the best of our knowledge, we are the first to simulate the light scattering by physically-based water drop shapes.

3. BACKGROUND THEORY

Rainbows are created from the interaction between light and a participating medium composed of water drops suspended in the air. The most important visual effects are due to single scattering (note that we consider multiple light bounces within a single water drop as single-scattering). The complex phase function resulting from this interaction produces the rich and varied angular distribution of radiance we observe as rainbows. Multiple scattering is responsible for the grayish background that appears behind the rainbows themselves. The effect of absorption of light in water reaches a maximum of 3.5×10^{-8} (expressed as the imaginary part of its refractive index [Pope and Fry 1997]). In the following we describe the formation of rainbows, from geometric optics to wave effects, and introduce the

actual shape of water drops, all of which will become the physical basis for our simulation algorithm described in Section 4.

3.1 Geometric Optics

The basic formation of the primary and secondary rainbow can be understood using simple geometric optics, considering ray paths within the circular cross-section of a spherical drop of water. For spherical drops, due to symmetry, the phase function is a 1D function of the scattering angle θ between the incident and outgoing directions. Light rays that undergo one internal reflection in the water drop produce the primary rainbow for red light (700 nm wavelength) for an index of refraction of $\eta = 1.3314$ at a scattering angle of $\theta_{\text{rainbow}} = 137.7^\circ$ and for violet light (400 nm wavelength) for $\eta = 1.3445$ at $\theta_{\text{rainbow}} = 139.6^\circ$ (see Figure 3a). Rainbows can also be generated by light rays that undergo two or more internal reflections: in the case of two internal reflections, the resulting secondary rainbow varies between 129.5° for red light and 126.1° for violet light (see Figure 3b). Note the order of the colors of the secondary rainbow (red on the inside of the arc and violet on the outside) is reversed compared to the primary rainbow, as seen in Figure 2c. The darker area between both rainbows is known as *Alexander's dark band* (better perceived in Figure 2d).

3.2 Wave Optics

Interference. Although geometric optics can provide a good, basic explanation of the formation of the primary and secondary rainbows, real rainbows exhibit some features that cannot be explained with this model. For example, additional arcs (known as *supernumerary arcs*) occasionally appear on the inside of the primary rainbow (and the outside of the secondary): such arcs are typically violet or blue (Figure 2e). Supernumerary arcs caused great

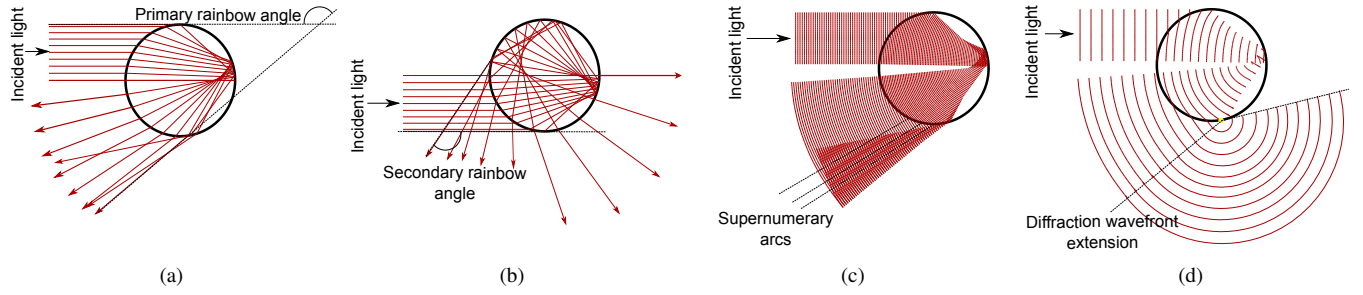


Fig. 3: Generation of rainbows from the point of view of geometric optics and wave optics: (a) primary rainbow angle, after a single internal reflection, (b) secondary rainbow angle, after two internal reflections, (c) supernumerary rainbows are generated from constructive and destructive interference patterns (inspired by Lee and Fraser [2001]), (d) diffraction extends the wavefront and avoids abrupt intensity changes.

consternation [Lee and Fraser 2001], because they were not predicted by geometric optics. However, in the 1830s, scientists such as Young and Airy realized that they were a consequence of the wave theory of light: two rays that have different path lengths, must also have different phases – which result in an interference pattern consisting of a series of maxima and minima as a function of the scattering angle (see Figure 3c). This phase difference between the two rays is also influenced by phase changes due to reflection.

Focal lines. As a wavefront interacts with the water drop, it gets deformed. The surface that represents this wavefront can be differentially defined by the curvature at each point. This curvature defines a radius with respect to a corresponding *focus* or *focal point*. Focal points are actually internal caustics within the water drop. As light passes through a water drop, the collection of all focal points lie along a *focal line* (see Figure 4). Each passage through a focal line along the path results in a phase advance of $\pi/2$ [van de Hulst 1957]. This traversal of focal lines needs to be considered for an accurate estimation of the phase difference between two interfering rays.

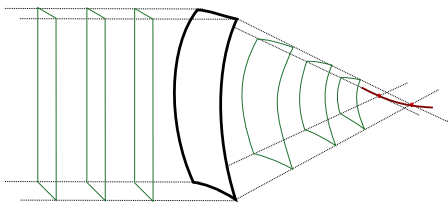


Fig. 4: A focal line defines the curve along which all the differential focal points of the wavefront lie. The thick black represents a portion of the surface of the water drop. The green patches represent the wavefront itself. Notice how they converge to the red line, which is the focal line.

Diffraction. Another failure of geometric optics is that it predicts infinite intensity at θ_{rainbow} and no scattering light when $\theta < \theta_{\text{rainbow}}$, while diffraction predicts that this abrupt radiance gradient cannot happen in reality [van de Hulst 1957] (see Figure 3d). Airy [1838] produced an elegant mathematical solution which avoided both of these problems and proved that the peak intensity of the rainbow does not occur at $\theta_{\text{rainbow}} = 137.86^\circ$, but at a slightly higher value of approximately $\theta = 138.9^\circ$.

Another consequence of diffraction is that scattering from small droplets of water (such as fog in which the droplet radius is typically between $5 \mu\text{m}$ and $20 \mu\text{m}$) can generate rainbows that are essentially white. As the diffraction pattern for small droplets has very broad maxima in terms of θ , the rainbows corresponding to different wavelengths in the visible spectrum tend to overlap each other, thus creating white *fog bows* or *cloud bows*. Therefore, diffraction becomes more relevant as the water drops get smaller.

3.3 Lorenz-Mie Theory

Lorenz-Mie theory [Lorenz 1890; Mie 1908] developed a rigorous solution to the problem of scattering of light from spheres, taking into account not only interference, but polarization and radius distribution as well. Figure 5 shows the result of Lorenz-Mie theory calculations to simulate the scattering of sunlight by a water drop with radius $100 \mu\text{m}$. It shows the primary rainbow near $\theta \approx 139^\circ$, the secondary rainbow near $\theta \approx 127^\circ$ and Alexander's dark band between $\theta \approx 130^\circ$ and $\theta \approx 136^\circ$. Note that the primary and secondary rainbows are strongly polarized: the dominant polarization is given by the perpendicular component of the electric field (with respect to the scattering plane). The colored horizontal bars above the graph in Figure 5 show the colors and relative brightness of the rainbows: the top bar represents perpendicular polarization, the middle bar represents parallel polarization while the lower bar represents unpolarized light (the combination of the two).

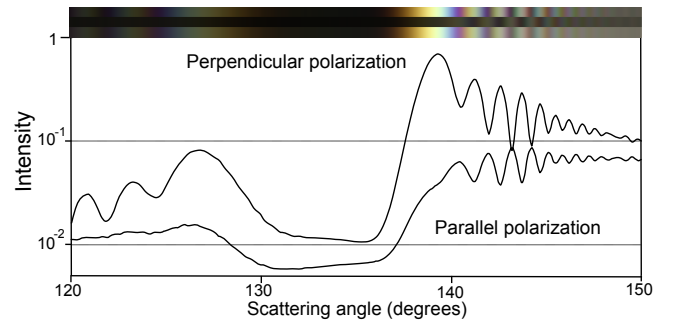


Fig. 5: Graph of intensity as a function of scattering angle for the primary rainbow caused by scattering of sunlight by a spherical drop of water of radius 0.1 mm . The color stripes on top represent, from top to bottom, the phase function for perpendicular polarization, parallel polarization and unpolarized light, respectively.

Figure 6 illustrates how the appearance of the primary and secondary rainbows varies with the radius of the (spherical) water drop, according to Lorenz-Mie theory: this type of diagram was first shown by Lee [1998] and is consequently known as a *Lee diagram*. It shows the complexity of rainbows and their supernumerary arcs.

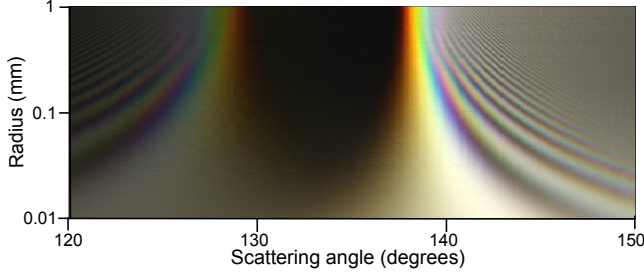


Fig. 6: Lee diagram showing the variation in appearance of primary and secondary rainbows caused by scattering of sunlight by a spherical water drop as a function of radius (Lorenz-Mie theory calculations)

3.4 Non-Spherical Water Drops

We consider physically-based water drops that are the same size and shape throughout the medium. Though water drops suspended in air are never homogeneous (the distribution of shapes and sizes can even be time-varying), given the ability to simulate phase functions for arbitrary geometry, such distributions can be accounted for by calculating and combining a set of phase functions according to the distribution. For each water drop size, we use the model by Beard and Chuang [1987] which accounts for surface tension as well as hydrostatic and aerodynamic pressure. Beard and Chuang proposed a cosine series fit to the model, with the shape of the raindrop profile given by the polar curve

$$r = a[1 + \sum c_n \cos(n\theta)], \quad (1)$$

where a is the radius of the equivalent volume sphere, while the coefficients c_n are listed as tabulated values (see Table I). Figure 7 shows visualizations of several water drop shapes based on this work. Other models and approaches (and even water drop distributions) can be trivially used in our simulations, as the algorithm can handle arbitrary geometries.

4. SIMULATING RAINBOW PHASE FUNCTIONS

A key aspect for an accurate simulation of rainbows is the precise computation of the phase function, which defines the angular distribution of radiance for every wavelength. Some of the approaches discussed in the previous work propose efficient methods to render rainbows, but they do not actually *simulate* precise

Table I.: Water drop polar curve coefficients for Equation (1) [Beard and Chuang 1987]. The $a = 0.4$ row has been added to account for spherical water drops. Intermediate values are obtained through linear interpolation.

$a(mm)$	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7
0.4	0	0	0	0	0	0	0	0
1.0	-0.0131	-0.0120	-0.0376	-0.0096	-0.0004	0.0015	0.0005	0
1.5	-0.0282	-0.0230	-0.0779	-0.0175	0.0021	0.0046	0.0011	-0.0006
2.0	-0.0458	-0.0335	-0.1211	-0.0227	0.0083	0.0089	0.0012	-0.0021
2.5	-0.0644	-0.0416	-0.1629	-0.0246	0.0176	0.0131	0.0002	-0.0044
3.0	-0.0840	-0.0480	-0.2034	-0.0237	0.0297	0.0166	-0.0021	-0.0072

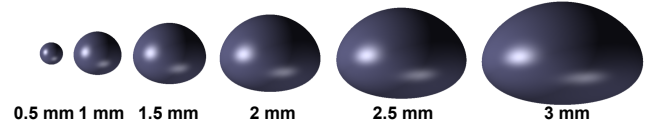


Fig. 7: Physically-based raindrop shape with increasing radii as proposed by Beard and Chuang [1987].

phase functions, which they take from available simulators such as AirySim², BowSim³ and MiePlot⁴. While these simulators do a great job at approximating the phase function of rainbows under some conditions, they all have hard limitations: none of them can handle physically-based water drop shapes, limiting the computations to spheres. BowSim can additionally handle ellipsoids, but it does not consider interference for its simulations. AirySim, however, approximates interference using Airy functions. In this section we focus on this key aspect of rainbow simulation and our proposed solution for arbitrary geometries. This is our primary contribution.

We compute phase functions for non-spherical water drops by taking a *virtual gonioreflectometer* approach. In essence, we simulate the way a collection of light rays scatter off a water drop and gather the resulting information on an infinite collecting sphere. The problem at hand is thus similar to rendering caustics, and therefore a pure Monte Carlo approach would be impractical. Unfortunately, photon mapping would not work either, since interference in this configuration produces extremely high frequency details which the radiance estimation technique would fail to reproduce. To include all the important optical properties of real rainbows we augment our ray tracing computation to account for dispersion, polarization, interference, and diffraction. Unlike Lorenz-Mie theory, which is limited to spherical drops, our approach allows us to use the real shape of the drops and thus produce more accurate simulations.

Our algorithm simulates the phase function by following several steps for each wavelength:

- (1) We cast a grid of rays from an emitting plane that represents the wavefront of a directional light source. Each ray carries wave information represented using phasors.
- (2) These rays interact with the water drop (through reflection and refraction) a number of times and exit the water drop forming patches.
- (3) The outgoing patches are deposited on an infinite collecting sphere and stored in an acceleration structure.
- (4) The phase function is discretized into tabulated form by querying the acceleration structure along a 2D set of sampling directions uniformly distributed in longitude-latitude.

The stored tabulated phase function is later used for rendering. All these steps are illustrated in Figure 8.

In the following, we first explain the basis of our approach from a classic ray-tracing perspective, for the sake of clarity; we then introduce our phasor notation which allows us to efficiently compute interference and polarization.

Casting Rays. Inspired by the beam tracing technique [Heckbert and Hanrahan 1984], and similar to the work by Collins [1994], we follow a wavefront of light by casting a grid of rays (3000×3000

²<http://www.atoptics.co.uk/rainbows/airysim.htm>

³<http://www.atoptics.co.uk/rainbows/bowsim.htm>

⁴<http://www.philiplaven.com/mieplot.htm>

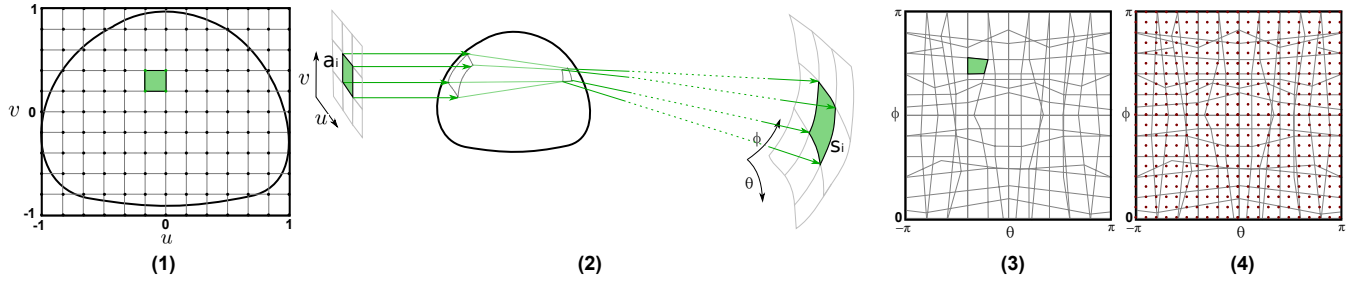


Fig. 8: Steps of the algorithm. (1) Casting the grid of rays towards the particle. (2) Rays are reflected and refracted towards the water drop, forming patches. (3) Outgoing patches are collected in an infinite collecting sphere. (4) The stored patches in the collecting sphere are queried at specific directions, sampling the phase function.

rays for our results). This way, rays that are contiguous and represent the same wavefront can be identified. Rays are perpendicular to a reference emitting plane, representing a collimated light source that emits a parallel wave train (see Figure 8(1)). Each of these rays is propagated through interactions with the water drop, which can be of arbitrary geometry. For our tests we use a physically-based geometric model that accounts for different particle sizes [Beard and Chuang 1987] although any other model or specific geometry could be considered instead.

When a ray interacts with the water drop, its path is reflected and refracted according to the law of reflection and Snell's law, respectively. We account for up to four consecutive interactions: a single reflection, two refractions, two refractions with an internal reflection (primary rainbow) and two refractions plus two internal reflections (secondary rainbow). Though further bounces could easily be handled, they have a negligible effect on the resulting phase function.

Collecting Sphere. When rays exit the water drop, we store the outgoing rays and their corresponding adjacency information as a set of patches on an infinite virtual collecting sphere (see Figure 8(3)). Each vertex of a patch thus represents one outgoing ray and contains wave data.

The energy of the emitting plane is split among all the grid cells according to each grid cell's area a_i . When this energy exits the particle and reaches the collecting sphere, it is transformed into radiance by considering the solid angle s_i of the resulting patch. Therefore, the ratio $\frac{a_i}{s_i}$ determines a patch's contribution to the phase function, which amounts to density estimation. While we could apply this relation directly to photometric units, this would not account for many of the effects that contribute to rainbows such as interference, polarization, focal lines, and diffraction. Instead, we apply this ratio to the corresponding wave data, which is described in the following section.

4.1 Computing Interference

To account for interference and polarization, we characterize light in terms of an electromagnetic field \mathbf{E} perpendicular to the direction of the ray. Defining a coordinate system with the z -axis along the direction of propagation, we can define it in terms of two orthogonal *phasors* of the electromagnetic field [Giancoli 1989]:

$$E_x = A_x e^{i(\frac{2\pi}{\lambda} z - \omega t + \delta_x)} \quad \text{and} \quad E_y = A_y e^{i(\frac{2\pi}{\lambda} z - \omega t + \delta_y)}, \quad (2)$$

where A_x and A_y are the amplitudes, λ is the wavelength, ω is the angular frequency, t is time and δ_x and δ_y represent phase off-

sets. The irradiance carried by a planar electromagnetic wave represented by two phasors is $A_x^2 + A_y^2$.

We assume that all the waves have traveled the same optical path from the sun (and therefore z is a constant reference path) and also consider a stationary simulation of the phase function, where ωt becomes constant. Furthermore, we sample fixed values of λ along the visible spectrum and simulate each independently. As a consequence, the only relevant information for each phasor is the amplitude A and the corresponding phase offset δ , which is the polar representation of the phasor.

Following Euler's formula, $e^{ix} = \cos x + i \sin x$, we can represent a phasor $Ae^{i\delta}$ by a complex number (rectangular representation) for which the real part is $A \cos \delta$ and its complex part is $A \sin \delta$. This rectangular representation is efficient for phasor addition (interference) and phasor interpolation. Furthermore, it enables a very straightforward simulation of the interactions between the electromagnetic wave and the water drop, by applying the corresponding Fresnel coefficients (as explained below in the text).

Each ray in our algorithm carries the following information:

- Two phasors E_x and E_y represented by complex numbers.
- The traversed optical path l .

Additionally, during ray tracing, we consider the frame that represents the coordinate system of the two axes of the electromagnetic wave (perpendicular to the propagation direction). These axes are rotated as needed for the different interactions.

Phase shifts need to be taken into account; these occur at the interaction with the water drop, along the traversed optical path, and at focal lines. As rays are traced, we modify the phasors at the interactions with the water drop. Phase shifts due to optical path and focal lines are included after the bilinear interpolation at each patch (see Subsection 4.1.1).

There has been some previous work for ray tracing polarization effects [Wolff and Kurlander 1990; Tannenbaum et al. 1994; Wilkie et al. 2001], all based on coherency matrices. Our approach, on the other hand is similar to using Jones vectors [JONES 1941] and can account for interference.

Ray-Water Drop Interactions. Light interacting with a water drop gets both reflected and refracted, with the total amplitude divided between both rays in terms of the parallel and perpendicular components with respect to the plane of incidence. We rotate the coordinates of the two components of the wave to a parallel-perpendicular coordinate system. As in previous work by Gondek et al. [1994], the respective amplitudes are multiplied by the Fresnel coefficients t_{\parallel} , t_{\perp} , r_{\parallel} and r_{\perp} , which can be found in most optics books [Lipson et al. 1995] and can become complex in the case

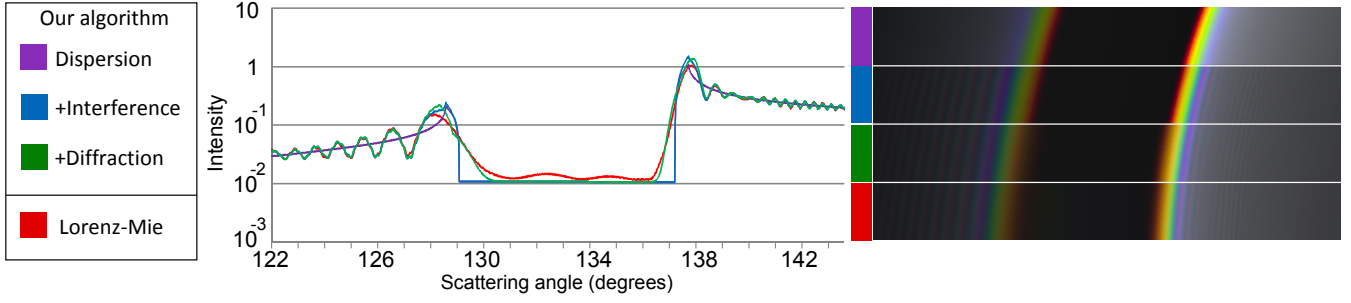


Fig. 9: Comparison between the different phenomena simulated by our method (dispersion, interference and diffraction) and the same simulation from the Lorenz-Mie theory, for a 0.4 mm radius spherical water drop. Left: Graphs for 650 nm wavelength. Right: Renderings for 33 wavelength.

of total internal reflection. We multiply phasors with these potentially complex coefficients using the rectangular representation of complex numbers explained before.

Optical Path. The optical path l is defined as $l = \int_P \eta dp$, where η is the index of refraction and p refers to the differential traversed path. In our case the total optical path traversed by a ray is $l = \eta_i p_i + \eta_t p_t$, where p_i and p_t are the total distances traversed outside and inside the water drop, respectively. Given the impossibility of computing infinite path lengths from the sun, we rely on the fact that interference computations require just *relative* optical paths between different rays; we thus consider the common casting plane to be placed at a distance d from the center of the water drop (which would represent a distance z from the sun in Equation 2). In a similar fashion, we set a second reference plane perpendicular to each outgoing ray, placed at a distance d' from the center of the water drop (*not* from the origin of the ray). The optical path l is accumulated as the ray traverses the water drop by simply adding the Euclidean distances between interactions outside and inside the particle. We account for this effect on phase change during the bilinear interpolation step in Subsection 4.1.1.

Focal Lines. Focal lines must also be considered for an accurate simulation of the phase carried by each ray, given that each passage through a focal line along the path results in a phase advance of $\pi/2$ [van de Hulst 1957]. Unfortunately, both computing focal lines caused by arbitrary geometry and detecting which rays actually traverse a focal line are very complex tasks. However, we can approximate the exact solution by leveraging the fact that we only need to take into account the area close to the rainbow. Furthermore, for interference, it is again only the phase difference that needs to be taken into account. We thus analyze the sign of the derivative of the outgoing angle θ with respect to the impact parameter $b = u^2 + v^2$ (where u and v are the parameters that define the projection plane from which the rays are cast, as illustrated in Figure 8(1)). When this derivative is positive, we consider one extra focal line than when the derivative is negative. The derivative (and therefore the number of focal lines) is easily computed at each of the patches from its corners. We consider that any direction inside a patch represents a ray that has crossed that number of focal lines. Figure 9 shows that this approximation of the real phenomenon leads to accurate interference simulation, needing minimal computational overhead, as opposed to numerically detecting the focal lines and intersecting them with all rays, which would be prohibitively expensive.

4.1.1 Interference. In order to save the phase function to the hard drive (so it can be later used in a renderer), we tabulate it per wavelength by generating a 2D set of directions uniformly distributed in longitude-latitude coordinates. Each of these samples corresponds to a direction \mathbf{r} . We compute the outgoing radiance for a specific direction \mathbf{r} within a patch using bilinear interpolation of the data stored at the four vertices of the patch. This bilinear interpolation is equivalent to assuming that the wavefront at each of the patches is planar, and the error we commit by making this assumption becomes negligible as the resolution of patches increases.

We then combine the interpolated data at each of the patches that contain a direction to account for interference between wavefronts. For efficiency we consider the whole set of patches as a virtual geometry and we create a bounding volume hierarchy (BVH) over them. Given an outgoing direction \mathbf{r} , we find the set of patches $\Upsilon(\mathbf{r})$ that contains \mathbf{r} by tracing a ray from the center of the collecting sphere in that direction. We consider all intersected patches for interpolation and interference.

In Figure 10, we show an example for two patches $k1$ (red) and $k2$ (yellow), which represent two different interfering wavefronts. The ray at each of the four corners v_i^k of each patch contains information about the two corresponding phasors E_{xk} and E_{yk} and the optical path l_k . At each of the patches $k \in \Upsilon(\mathbf{r})$ we bilinearly interpolate this information from the four corners (at the specific direction \mathbf{r}). Furthermore, we calculate the number of traversed focal lines f_k for that patch, as explained above. As the irradiance car-

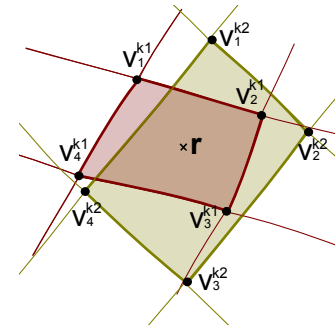


Fig. 10: Direction \mathbf{r} intersects patches $k1$ (red) and $k2$ (yellow) (two parts of two different wavefronts). At each of both patches, bilinear interpolation from all the corners v_i^k result into an interpolated electromagnetic wave per patch. Interference is then computed by combining (adding) all the interpolated electromagnetic waves.

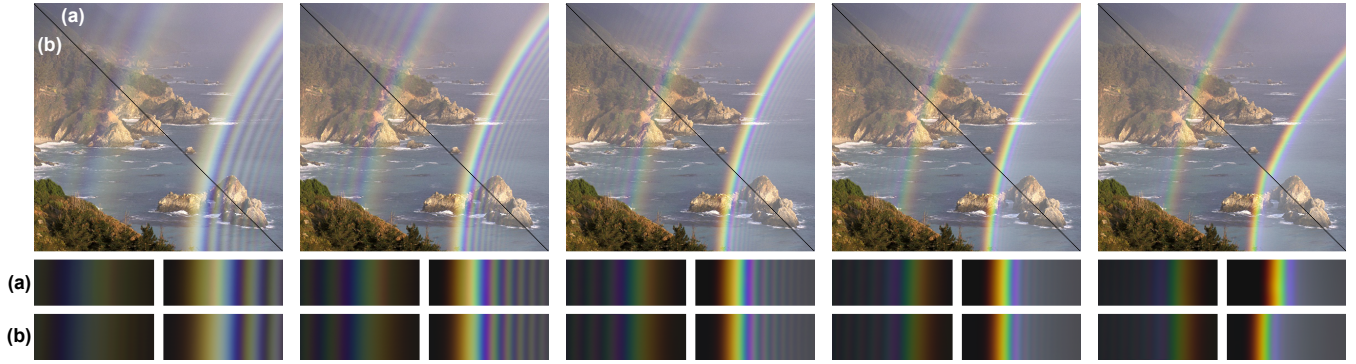


Fig. 11: Comparison between our method (a) and Lorenz-Mie theory (b) for different water drop sizes. Top: Renders. Bottom: Plots of the phase function for the regions of the primary and secondary rainbows. From left to right: Water drops of radius 0.1 mm, 0.2 mm 0.3 mm, 0.4 mm and 0.5 mm, respectively. Our method matches Lorenz-Mie theory for small water drops, which are spherical, but predicts different behavior as the radius increases by accounting for non-spherical drop shapes.

ried by a planar electromagnetic wave represented by two phasors is $A_x^2 + A_y^2$ we account for the corresponding radiance (applying the $\frac{a_i}{s_i}$ factor, as stated in the previous section) by multiplying each of the amplitudes by $\sqrt{\frac{a_i}{s_i}}$. We then obtain the new phasor information E'_{xk} and E'_{yk} including the phase shift due to the traversed optical path and to the number of traversed focal lines for each interpolated wave as:

$$E'_{xk} = E_{xk} e^{i(\frac{2\pi}{\lambda} l_k + \frac{\pi}{2} f_k)}, \quad E'_{yk} = E_{yk} e^{i(\frac{2\pi}{\lambda} l_k + \frac{\pi}{2} f_k)}. \quad (3)$$

We then compute the final outgoing radiance due to interference by adding all the traversed phasors:

$$E_x(\mathbf{r}) = \sum_{k \in \Upsilon(\mathbf{r})} E'_{xk}, \quad E_y(\mathbf{r}) = \sum_{k \in \Upsilon(\mathbf{r})} E'_{yk}, \quad (4)$$

where $E_x(\mathbf{r})$ and $E_y(\mathbf{r})$ are the two components of the wave that exits the water drop towards \mathbf{r} .

4.1.2 Diffraction. A fundamental problem with using ray-tracing techniques to simulate rainbows lies in the fact that geometric optics predicts *infinite* intensity at the rainbow angle with a very abrupt transition to zero intensity, as shown by the blue curve in Figure 9. In contrast, Lorenz-Mie theory predicts that maximum intensity occurs slightly above the geometric rainbow angle (see green curve at about 138.5° in Figure 9). Note also that this transition is softened so the intensity at the geometric rainbow angle is less than the maximum intensity, with some light being scattered into the zone below the geometric rainbow angle where no geometric rays can penetrate.

This process is very similar to diffraction by a knife edge in which some light appears in the shadow zone. An accurate calculation of the effect of diffraction on the rainbow light field would require the application of the Huygens-Fresnel principle for each differential point on each wavefront, which is time-consuming and impractical. Such techniques would also be able to predict the supernumerary arcs, but this is not necessary as Figure 11 shows that the supernumerary arcs predicted by our ray-tracing technique are already in very close agreement with Lorenz-Mie theory. Hence, we need only to address the diffraction effect, which can be efficiently approximated by performing a post process on the computed phase function, by first identifying very sharp transitions in intensity at a given wavelength and then smoothing out the sharp

peaks by applying a domain-specific kernel. The size of the chosen kernel depends on the size of the water drop. For efficiency reasons, we choose a simple Gaussian kernel, summarized in Table II for different radii. The values have been obtained from Lorenz-Mie theory for spherical drops, which our results show offer a good approximation. For the secondary rainbow we double the standard deviation of the kernel, to account for the fact that light has been reflected twice inside the water drop.

Table II.: Standard deviation of the Gaussian filter diffraction approximation for various water drop sizes.

Radius (mm)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
σ (degrees)	0.70	0.45	0.30	0.25	0.22	0.20	0.18	0.17	0.16	0.15

The addition of the diffraction filter produces some fairly subtle changes in our simulations, as can be seen in the simulations shown in Figure 9. In essence, the diffraction filter softens the transitions near the rainbow angle thus giving a better match to the Lorenz-Mie simulations. It is important to acknowledge that the parameters for our diffraction filter have not been thoroughly validated. Further work based on the application of the Huygens-Fresnel principle could overcome such concerns, but the increased accuracy of such techniques would be outweighed by an immense increase in computational complexity. In these circumstances, the diffraction filter seems to be a sensible approximation that adequately addresses a fundamental limitation of geometric optics.

5. RESULTS

We have used our technique to simulate several phase functions, and then used those phase functions to render images depicting various types of rainbows. Unless stated otherwise, each of the results shown on this section has been simulated by casting rays from a 3000×3000 grid for each wavelength, uniformly sampling 33 different wavelengths between 380 and 720 nm. The resulting phase functions were sampled at an angular resolution of 1800×14400 (which is dense enough to account for the cusp of the rainbow and the high frequency details of interference) and stored on disk. Rainbows are obtained by ray marching and computing single scattering along the volume, importance-sampling the sun (which is modeled as a disc subtending a solid angle of 0.5°). On an Intel(R) Xeon(R)



Fig. 12: The inserts in these images show how our model can reproduce the rainbows in the underlying photographs. Only the background color of the insert has been matched to the specific photograph. Top row, from left to right: double rainbow (background reproduced with permission ©Les Cowley - <http://www.atoptics.co.uk/rainbows/adband.htm>), full double rainbow (background reproduced with permission ©Karl Kaiser - <http://home.eduhi.at/member/nature>) and supernumerary bows. Bottom row, from left to right: Multiple supernumerary bows (background reproduced with permission ©Ian Goddard - <http://www.atoptics.co.uk/rainbows/bowim46.htm>), cloud bow (background reproduced with permission ©Les Cowley - <http://www.atoptics.co.uk/rainbows/cldbow.htm>) and red bow.

Table III. : The list of parameters used to produce the rendering results in Figure 12.

Figure	Top Left	Top Middle	Top Right	Bottom Left	Bottom Middle	Bottom Right
Water Drop Size	0.4 mm	0.4 mm	0.3 mm	0.3 mm	0.1 mm	0.4 mm
FOV	20°	100°	30°	10°	100°	30°
Lens Type	Rectilinear	Fisheye	Rectilinear	Fisheye	Rectilinear	Rectilinear
Background Color	(107,114,118)	(183,202,212)	(172,172,172)	(69,99,112)	(141,180,223)	(154,83,58)
Intensity	55%	100%	90%	80%	60%	80%
Illumination	D65	D65	D65	D65	D65	D65 + Rayleigh

CPU X5570 at 2.93 GHz, using 8 GB of RAM, our simulations took an average of 350 minutes to compute for all 33 wavelengths.

To validate our algorithm, we simulated the phase function of spherical water drops of different sizes, comparing our results with the predictions of Lorenz-Mie theory. Figure 9 shows this comparison for a 0.4 mm spherical water drop on a log-scale: purple represents dispersion, the pure geometric interpretation of the phase function of the rainbow; blue adds interference, and therefore the oscillations of the supernumerary arcs appear; green adds diffraction, eliminating the high intensity peaks at the geometric rainbow angle; and red represents the simulation from Lorenz-Mie theory. Notice the similarity between the green line (our complete simulation) and the red line (Lorenz-Mie simulation). The main differences are observed in the Alexander band, due to our diffraction approximation. Figure 11 shows rainbow renderings from the simulated phase functions, again exploring variations in size; for small sizes, where drops can be considered spherical, our results match Lorenz-Mie's predictions. However, larger drop sizes (0.5 mm in the figure) stop being spherical and consequently our algorithm predicts a different behavior.

Our method accurately reproduces several rainbow-related phenomena seen in nature. For instance, our algorithm can trivially reproduce the primary and secondary rainbows as seen in Figure 12 top-left and top-middle). Also, by simulating interference we are able to simulate supernumerary bows (Figure 12 top-right and bottom-left). By including also the effect of diffraction on the rainbow, we can simulate phenomena such as the cloud bow (Figure 12 bottom middle) in which the colors of the rainbow disappear into a whitish bow. Simulating the effect of Rayleigh scattering allows us to mimic the effect of a sunset on a rainbow (Figure 12 bottom-right).

When superimposing simulations on images of natural rainbows, as in Figure 12, it is necessary to estimate the focal length of the camera lens. Fortunately, this information is often available in the EXIF data embedded in digital images. The angular performance of most camera lenses can be approximated by a rectilinear mapping function, except for fisheye lenses which are better modeled by equidistant, equal-area, stereographic or orthographic mapping functions. However, even with full information about the camera and its lens, it is also necessary to know the aiming point of the

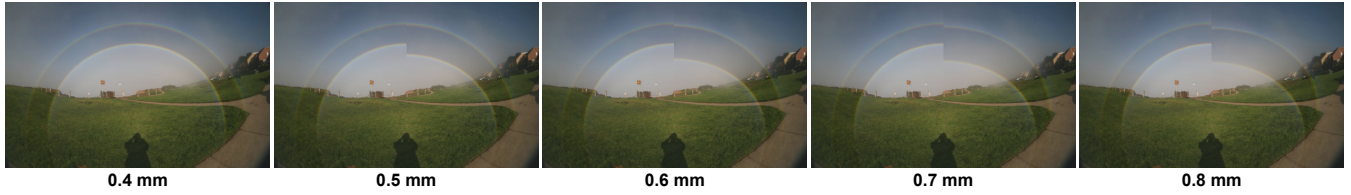


Fig. 13: Comparison of renderings of rainbows owed to different water drop radii between Lorenz-Mie (left region on each image) and our solution (right region of each image). As the 0.4 mm radius water drop is spherical, both algorithms lead to equal phase functions. As the particle gets bigger, the geometry becomes non-spherical and therefore Lorenz-Mie is unable to simulate it, while our solution takes it into account. Notice, also, that the variation on the secondary rainbow is quite unnoticeable compared to the variation on the primary rainbow, in agreement with the formation of twinned rainbows.

camera relative to the direction of the Sun or the anti-solar point. In practice, some of this information is typically missing (along with the obvious fact that the photographer is unlikely to provide any information about the size of the water drops causing the rainbows). Consequently, the simulation parameters generally need to be adjusted by trial and error to get a good match with the original image. The parameters used to produce the rendering results in Figure 12 are listed in Table III.

Figure 1 shows an overview of the different rainbow-related phenomena our algorithm is able to simulate.

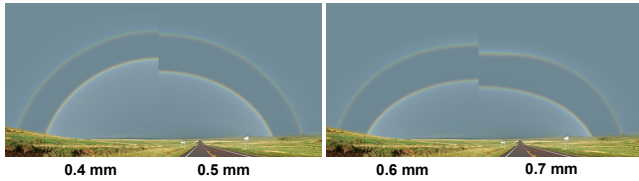


Fig. 14: The effect of different water drop radii on the apparent geometry of the rainbow.

As discussed in Section 3.4, as water drops get larger they get deformed due to the impact of air resistance. This drastically affects the appearance of the final rainbow as shown in Figure 13. Producing these physically accurate phase functions for large water drops is, to our knowledge, not possible using any other method. Figure 14 shows the effect of the size of the water drop on the apparent geometry of a full rainbow.

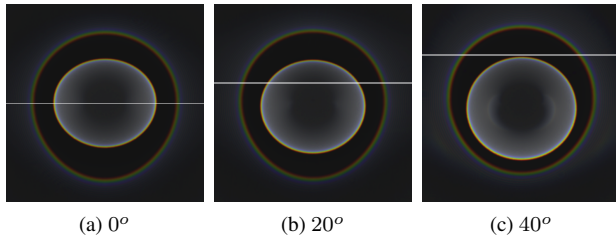


Fig. 15: The effect of the inclination of the sun on a non-spherical water drop (radius 0.5 mm) alters the apparent geometry of the rainbow. This would not be the case for spherical water drops. The gray line indicates the horizon line.

While the phase function for spherical water drops is invariant to the inclination of the sun, non-spherical water drops produce very different phase functions for each incident direction of light. In other words, rainbows are actually the result of an anisotropic phase

function within an anisotropic medium. Figure 15 shows the effect of the inclination of the sun on a 0.5 mm non-spherical particle. We have set up the viewing direction to be parallel to the direction from the center of the sun, and we show the full (theoretical) 360° rainbow. For reference, the gray line indicates the horizon. Note that a rainbow due to spherical water drops would look identical in all these images.

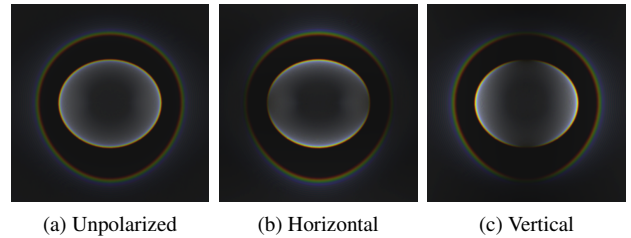


Fig. 16: The effect of viewing a rainbow through different polarizing filters (assuming 0.5 mm radius non-spherical water drop)

Furthermore, our algorithm naturally takes into account polarization. This enables us to explore the effects of different light polarization filters on the perception of the rainbow given by a water drop of arbitrary geometry. Figure 16 shows an example of this for unpolarized viewing and for viewing through polarizing filters with their transmission axes aligned horizontally and vertically.



Fig. 17: Left: Photograph of a rare twinned rainbow (reproduced with permission ©Benjamin Kuehne - <http://www.nachtwolke.de/temp/regenbogen2.htm>). Right: Twinned rainbow simulated using our algorithm, generated from two showers of 0.4 mm and 0.45 mm radius water drops.

Lastly, twinned rainbows can only be explained by a combination of two types of water drops with different sizes where at least one of them is non-spherical. Figure 17 shows a simulation of a twinned rainbow caused by two showers of 0.4 mm and 0.45 mm radius water drops. Note that slight changes in water drop sizes

alter their geometry and have a drastic impact on the resulting rainbow. Similar results have been reported by simulating the light scattering from ellipsoids using BowSim⁵. However, to our knowledge, this is the first time that such a complex rainbow has been simulated, based on light scattering from water drops with realistic shapes.

6. CONCLUSION AND FUTURE WORK

We have presented the first comprehensive model of rainbows suitable for computer graphics applications. We have validated it against Lorenz-Mie theory for the case of spherical water drops, and shown how it naturally overcomes the limitations of such theory.

However, our simulations have some limitations and can be improved in the future. One of the main approximations of our model is the diffraction filter. This filter introduces small errors around and inside Alexander's dark band, specially for small water drops. Further work based on the application of the Huygens-Fresnel principle can improve this component. For efficiency reasons, we also make approximations for computing the traversed focal lines, although less efficient analytical solution might give more accurate results as the water drop diverges from a spherical shape. Furthermore, by applying interpolation our simulations assume that the wavefront is planar within each patch, which is an approximation of the real phenomena. This can be easily improved by increasing the resolution of the casting grid, although it would be interesting to explore other types of interpolation (such as bicubic) and their interpretations from the optical point of view. In addition, our renderings use a homogeneous size and density distribution of water drops which in turn will make all the features sharper and more noticeable. Real-world imperfections would contribute to additional blurring of the rainbow. This issue can be resolved by calculating and combining a series of phase functions according to the water drop size distribution. Finally, matching a reference photograph with rendering results is a manual process. A potential extension to our research would aim to use computer vision techniques to automate this process.

This research also opens other potential lines of investigation. Though we did not focus on performance in our work, we believe that our algorithm could be adapted to the GPU, greatly accelerating the phase function simulation. Further development on our phase function simulator could lead to new and generalized global illumination algorithms, taking into account phenomena such as interference or diffraction. We foresee that a wide set of disciplines, such as meteorology or remote sensing, could benefit from our technique.

Acknowledgments

We would like to thank Toshiya Hachisuka and Krystle de Mesa for proofreading the paper and their constructive discussions. Thanks also to Mapi for the rainbow image in Figure 2d and Jurgen Schulze for referring us to the facility of Project GreenLight. And finally, we like to thank the anonymous reviewers for their helpful suggestions and comments.

This research has been partially funded by NSF Project GreenLight (award no.: 0821155), a Marie Curie grant from the Seventh Framework Programme (grant agreement no.: 251415), the Spanish Ministry of Science and Technology (TIN2010-21543) and the Gobierno de Aragn (projects OTRI 2009/0411 and CTPP05/09).

REFERENCES

- AIRY, G. 1838. On the intensity of light in the neighbourhood of a caustic. *Trans. Cambridge Philos. Soc.* 6, 379–403.
- BEARD, K. V. AND CHUANG, C. 1987. A new model for the equilibrium shape of raindrops. *Journal of the Atmospheric Sciences* 44, 11, 1509–1524.
- BEARD, K. V., KUBESH, R. J., AND III, H. T. O. 1991. Laboratory measurements of small raindrop distortion. part i: Axis ratios and fall behavior. *Journal of the Atmospheric Sciences* 48, 5, 698–710.
- BOHREN, C. AND HUFFMANN, D. 1983. *Absorption and Scattering of Light by Small Particles*. Wiley.
- BRINGI, V., CHANDRASEKAR, V., AND XIAO, R. 1991. Raindrop axis ratios and size distributions in florida rainshafts: an assessment of multiparameter radar algorithms. *IEEE Transactions on Geoscience and Remote Sensing* 36, 3, 703–715.
- COLLINS, S. 1994. Adaptive splatting for specular to diffuse light transport. In *Fifth Eurographics Workshop on Rendering*. 119–135.
- FRISVAD, J. R., CHRISTENSEN, N. J., AND FALSTER, P. 2007. The aristotelian rainbow: from philosophy to computer graphics. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*. GRAPHITE '07. ACM, New York, NY, USA, 119–128.
- FRISVAD, J. R., CHRISTENSEN, N. J., AND JENSEN, H. W. 2007. Computing the scattering properties of participating media using lorenz-mie theory. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*. ACM, New York, NY, USA, 60.
- GEDZELMAN, S. 2008. Simulating rainbows in their atmospheric environment. *Applied Optics* 47, 34, 176–181.
- GIANCOLI, D. C. 1989. *Physics for Scientists and Engineers*. Prentice Hall.
- GONDEK, J. S., MEYER, G. W., AND NEWMAN, J. G. 1994. Wavelength dependent reflectance functions. In *SIGGRAPH '94*. ACM, New York, NY, USA, 213–220.
- GREENLER, R. 1990. *Rainbows, Halos, and Glories*. Cambridge University Press.
- GUY, S. AND SOLER, C. 2004. Graphics gems revisited. *ACM Transactions on Graphics (Proceedings of the SIGGRAPH conference)*.
- HECKBERT, P. S. AND HANRAHAN, P. 1984. Beam tracing polygonal objects. *SIGGRAPH Comput. Graph.* 18, 119–127.
- JACKËL, D. AND WALTER, B. 1997. Modeling and rendering of the atmosphere using Mie-scattering. *Computer Graphics Forum* 16, 4, 201–210. ISSN 1067-7055.
- JONES, R. C. 1941. A new calculus for the treatment of optical systems. *J. Opt. Soc. Am.* 31, 7 (Jul), 488–493.
- LAVEN, P. 2003. Simulation of rainbows, coronas, and glories by use of Mie theory. *Applied Optics* 42, 3, 436–444.
- LEE, R. L. 1998. Mie theory, Airy theory, and the natural rainbow. *Applied Optics* 37, 9, 1506–1519.
- LEE, R. L. AND FRASER, A. B. 2001. *The rainbow bridge: rainbows in art, myth, and science*. Pennsylvania State University Press.
- LIPSON, S., LIPSON, H., AND TANHAUSER, D. 1995. *Optical Physics*, Third edition ed. Cambridge.
- LORENZ, L. 1890. Lysbevgelser i og uden for en af plane lysblger belyst kugle. det kongelig danske videnskabernes selskabs skrifter. *Rkke, Naturvidenskabelig og Mathematisk Afdeling* 6, 1, 262.
- LYNCH, D. AND LIVINGSTON, W. 2001. *Color and Light in Nature (second edition)*. Cambridge University Press.
- MIE, G. 1908. Beitr age zur optik tr uber medien, speziell kolloidaler metall osungen. *Annalen der Physik* 25, 3, 377445.
- MINNAERT, M. 1993. *Light and Colour in the Outdoors*. Springer-Verlag.

⁵<http://www.atoptics.co.uk/rainbows/twin1.htm>

- MISHCHENKO, M. I., HOVENIER, J. W., AND TRAVIS, L. D. 2000. *Light scattering by nonspherical particles : theory, measurements, and applications*.
- MUSGRAVE, F. K. 1989. Prisms and rainbows: A dispersion model for computer graphics. In *Graphics Interface '89*. 227–234.
- NARASIMHAN, S. G. AND NAYAR, S. K. 2003. Shedding light on the weather. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. 665 – 672.
- NVIDIA. 2004. Rainbows and fogbows: adding natural phenomena. Sdk white paper.
- POPE, R. AND FRY, E. 1997. Absorption spectrum (380-700 nm) of pure water. ii. integrating cavity measurements. *Applied Optics* 36, 33, 8710–8723.
- RILEY, K., EBERT, D. S., KRAUS, M., TESSENDORF, J., AND HANSEN, C. 2004. Efficient rendering of atmospheric phenomena. In *Proceedings of the 2004 Eurographics Symposium on Rendering*, D. Fellner and S. Spencer, Eds. 375–386.
- RUSHMEIER, H. 1995. Input for participating media. Realistic Input for Realistic Images, ACM SIGGRAPH '95 Course Notes. Also appeared in the ACM SIGGRAPH '98 Course Notes - A Basic Guide to Global Illumination.
- TAFLOVE, A. 1998. *Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House Inc.
- TANNENBAUM, D. C., TANNENBAUM, P., AND WOZNY, M. J. 1994. Polarization and birefringency considerations in rendering. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. SIGGRAPH '94. ACM, New York, NY, USA, 221–222.
- UMASHANKAR, K. AND TAFLOVE, A. 1982. A novel method to analyze electromagnetic scattering of complex objects. *Electromagnetic Compatibility, IEEE Transactions on EMC-24*, 4 (nov.), 397–405.
- VAN DE HULST, H. C. 1957. *Light Scattering by Small Particles*. Dover Publications Inc.
- VILLERMAUX, E. AND BOSSA, B. 2009. *Nature Physics* 5, 697–702.
- WILKIE, A., TOBLER, R. F., AND PURGATHOFER, W. 2001. Combined rendering of polarization and fluorescence effects. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*. Springer-Verlag, London, UK, 197–204.
- WOLFF, L. B. AND KURLANDER, D. J. 1990. Ray tracing with polarization parameters. *IEEE Comput. Graph. Appl.* 10, 44–55.
- XU, F., LOCK, J. A., AND TROPEA, C. 2010. Debye series for light scattering by a spheroid. *J. Opt. Soc. Am. A* 27, 4 (Apr), 671–686.
- YANG, P. AND LIOU, K. N. 1995. Light scattering by hexagonal ice crystals: comparison of finite-difference time domain and geometric optics models. *J. Opt. Soc. Am.* 12, 162–176.
- YANG, P. AND LIOU, K. N. 1996. Finite-difference time domain method for light scattering by small ice crystals in three-dimensional space. *J. Opt. Soc. Am.* 13, 2072–2085.
- YEE, K. 1966. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *Antennas and Propagation, IEEE Transactions on* 14, 3 (may), 302–307.

Received April 2011; accepted July 2011