Eurographics Symposium on Rendering 2014
Wojciech Jarosz and Pieter Peers
(Guest Editors)

Volume 33 (2014), Number 4

Spectral Ray Differentials

Oskar Elek[1,2] Pablo Bauszat[3,1] Tobias Ritschel[1,2] Marcus Magnor[3] Hans-Peter Seidel[1]

MPI Informatik[1] MMCI Saarland University[2] TU Braunschweig[3]

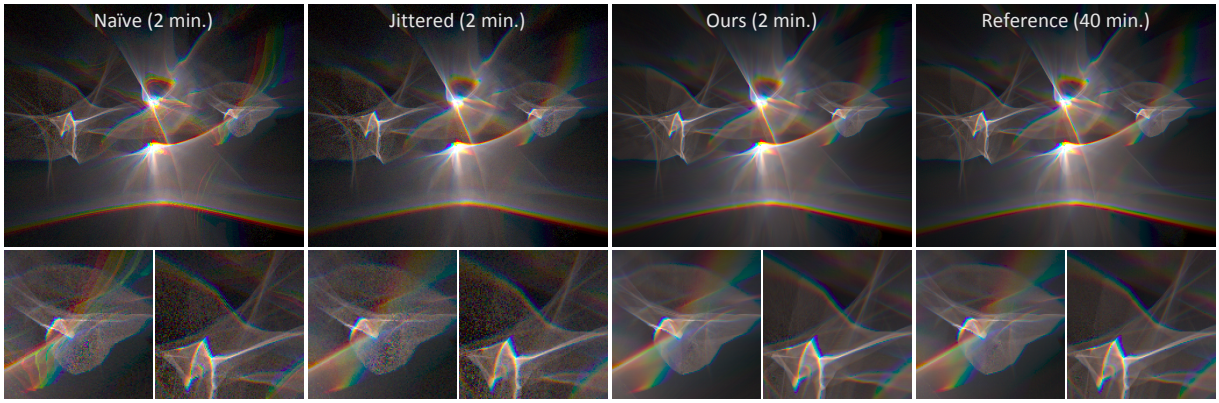

Figure 1: *Complex dispersive caustic rendered with spectral light tracing using 7 spectral bands. The naïve solution exhibits strong aliasing artifacts due to discrete sampling of the spectrum; stochastically jittering the spectral samples solves this problem, but the solution still contains significant chromatic noise. Our proposed spectral differentials allow us to efficiently reconstruct the solution, yielding an image visually identical to the reference (rendered with 35 spectral bands and 4 times more samples).*

Abstract

Light refracted by a dispersive interface leads to beautifully colored patterns that can be rendered faithfully with spectral Monte-Carlo methods. Regrettably, results often suffer from chromatic noise or banding, requiring high sampling rates and large amounts of memory compared to renderers operating in some trichromatic color space. Addressing this issue, we introduce spectral ray differentials, which describe the change of light direction with respect to changes in the spectrum. In analogy with the classic ray and photon differentials, this information can be used for filtering in the spectral domain. Effectiveness of our approach is demonstrated by filtering for offline spectral light and path tracing as well as for an interactive GPU photon mapper based on splatting. Our results show considerably less chromatic noise and spatial aliasing while retaining good visual similarity to reference solutions with negligible overhead in the order of milliseconds.

1. Introduction

When light refracts at a dispersive interface (such as a glass prism) its new direction is wavelength-dependent, resulting in colorful rainbow-like patterns. This is both true for caustics where light disperses before it hits a diffuse surface as well as for objects when directly seen through dispersive interfaces, e. g., a low-quality lens system, exhibiting chromatic aberrations. Creating light or eye paths for rendering in the presence of dispersion then implies that a single unique wavelength has to be used for every path as different wavelengths take different paths. This results either in spectral under-sampling (banding) or chromatic noise, as demonstrated in Fig. 1.

To address this problem, we introduce the concept of *spectral ray differentials*. The common ray, path and photon differentials [Ige99, SW01, SFES07, FD09] describe the wavelength-independent change of a ray, path or photon in the spatial domain with respect to the change of the initial coordinates in the respective domain where they originate from. The domain is defined by the sensor plane for rays and paths, or by the light source coordinates for photons. This information is then used primarily for improved texture filtering and photon density estimation. In contrast, our spectral ray differentials model the spatial change of a ray when interacting with dispersive interfaces per change of wave-

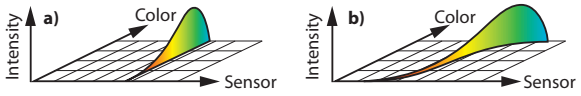

Figure 2: *Interaction of spectral light sample color and intensity as well as a (1D) sensor in a classic* (a) *and our* (b) *image formation model. Commonly, the samples contribute all wavelengths to the same sensor location (alignment with the color axis). In our case, samples at different wavelengths contribute to different sensor locations. This function can have a complicated shape (indicated by a curve), for which we use a piecewise (in the spectrum) linear approximation.*

length. Intuitively, they describe how light transport changes, when wavelength-dependent interactions occur along a path. Spectral ray differentials can be used for reconstruction of wavelength-dependent phenomena in offline as well as interactive rendering which we demonstrate for dispersive ray- and light tracing, as well as interactive GPU photon mapping. Additionally we show an example of how spectral differentials open up new possibilities besides rendering, such as interactive local and non-physical artistic control over dispersion effects.

The main contributions of our paper are:

- Theory of spectral ray differentials.
- A closed-form solution for the change of the spectral ray differential at a general dispersive (dielectric) interface.
- A family of reconstruction techniques to reduce chromatic noise and aliasing from images generated by ray-, light- and photon tracing based on spectral ray differentials.

Overview Our technique extends tracing of light and eye rays by tracking the information of how the samples need to be reconstructed due to dispersive effects when they are connected to the sensor. When tracing a ray, an additional quantity, the spectral ray differential—a simple pair of 3D vectors—is updated at each vertex (interaction). When connecting to the sensor, the spectral ray differential controls how this connection is made for different wavelengths (Fig. 2).

2. Previous work

Realistic image synthesis A reliable but costly approach to render physically accurate images is Monte-Carlo (MC) raytracing [Kaj86]. MC raytracing naturally supports the rendering of dispersive materials [Tho86]. While many renderers assume an RGB model, dispersion is a typical example where spectral rendering excels. Spectral rendering introduces an additional dimension (the spectral domain) into the rendering equation which is often discretized into uniform intervals, so-called bands. Dispersion effects suffer from spectral under-sampling resulting in chromatic noise when samples are jittered inside each band and aliasing artifacts that manifest as color fringes when the bands are sampled in a regular fashion (cf. Fig. 1). The overhead of spectral rendering is due to the higher memory and computational cost of storing and processing color spectra and also due to the fact that more rays need to be traced as each path is monochromatic or carries only a small fraction of the spectrum.

Light tracing [Arv86] or photon mapping [Jen01] are alternatives that connect light paths and eye paths to simulate caustic phenomena that are difficult to capture when only eye rays are considered. Caustics are a particularly intriguing manifestation of reflection and refraction, where light interacts at a smooth interface and then concentrates at a diffuse (rough) surface. In the case of dispersive interaction the caustic can have a colored, rainbow-like appearance. Photon mapping naturally extends to spectral rendering, but special considerations need to be taken when a photon interacts with a dispersive interface and the aforementioned challenges in storing and processing spectral information persist (cf. [Lai07]). When connecting to the first vertex from the eye, splatting of photons is preferred over density estimation using a k nearest neighbor queries in both offline- [HHK*07, HOJ08] and interactive GPU rendering [ML09]. Our spectral ray differentials apply both to paths starting from the eye and the light (see Sec. 5), and thanks to their improved reconstruction allow tracing and storing significantly less samples to obtain smooth solutions.

Ray differentials The concept of ray differentials was introduced by Igehy [Ige99] and is commonly used to improve texture filtering. It describes the spatial change of a ray (that undergoes several interactions) when changing its sensor coordinate by a small (differential) step. Later, path differentials [SW01] extend on the idea and generalize differentials to full paths, starting at the camera or the light. Photon differentials [SFES07] are a special case of path differentials for light paths, where the traced footprint is used to improve density estimation when connecting to the eye path. Fabianowksi and Dingliana [FD09] have used differentials for diffuse surfaces in the context of photon mapping. To our best knowledge, no previous work has considered using ray differentials within the additional, spectral, domain. We will recall the detailed definition of ray differentials in Sec. 3 and present our extension to spectral ray differentials in Sec. 4.

Reconstruction Recent work has analyzed light transport in the frequency domain [DHS*05]. Under the assumption of a specific geometric and material configuration [MWR12, MWRD13], the resulting filters can become as simple as (or be approximated by) an axis-aligned Gaussian filter that can be applied in the image domain efficiently. Directly applying Fourier light transport theory—which is already mathematically involved—to dispersion is not trivial. Instead of making assumptions about the scene and trying to find closed-form frequency bounds, we rely on tracing actual rays and their differentials. After computing spectral ray differentials, we proceed by applying special, wavelength-

dependent spatially-variant filters to reconstruct the dispersed light (see Sec. 6).

Human eye and lens simulation Dispersive and other chromatic aberration phenomena are prominent in both the human eye and optical lens systems. When simulating these, blurring was used to reduce variance between color bands such as for apertures simulated with Fraunhofer diffraction [KMN*04], the human eye [RIF*09] or lens systems [HESL11]. In these, either the special properties of diffraction are exploited (Fraunhofer diffraction at one wavelength is a scaled copy of another wavelength) or the blur size and direction is defined by the lens system. In comparison, spectral ray differentials allow us to trace this information directly.

Real-time rendering Using different directions to perform a look-up into an environment map is a classic approximation technique for rendering dispersive refractions [Kil01]. Artists can use an image editing application (such as Adobe Photoshop) to turn a regular refraction in an RGB image into a dispersive refraction with a colored edge by blurring with an appropriate empirical filter. One exceptional use of ray differential-like considerations is the filtering of caustics by Wyman and Dachsbacher [WD08]. Without considerations about the spectral behavior of light transport, however, the result always suffers from under-sampling where a spatially blurred version of discrete color bands is visible instead of a continuous spectrum with sharp geometric features.

Chromatic aberration Filtering between different colors is a classic task when dealing with sensors [LP94]. Chromatic aberration is present in many optical systems, and commercial software such as Adobe Photoshop offer tools to reduce it. For a careful analysis of chromatic aberrations and a way to reduce them, see Kang [Kan07]. Our filtering approach has similarity to the warping approach of Boult and Wolberg [BW92] that is used to remove aberration, however with the opposite intent. Remarkably, dispersive effects are present in all real images but often are missing in synthetic images [JF06], a fact that may be used to detect forged images.

3. Background: Ray differentials

First, we recall the concept behind ray and photon differentials as well as the physics of dispersion. Then, we establish a notation for ray and photon differentials leading to the notation of spectral differentials.

Ray differentials The classic ray differentials [Ige99] describe how the position and direction of a ray change when its initial sensor position is differentially offset. The sensor is assumed to be a simple pinhole camera. Let a position $\mathbf{p}$ and direction $\mathbf{d}$ define a ray $\mathbf{R} = (\mathbf{p}, \mathbf{d})$, and x and y be the sensor coordinates. Let also $\mathbf{R}(x,y)$ be the ray through that pixel. The initial ray position $\mathbf{p}(x,y)$ is set to the eye position $\mathbf{e}$ for all rays starting at the sensor. The ray direction

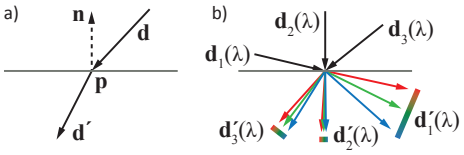

Figure 3: (a) *Refraction for a single monochromatic ray and* (b) *dispersion of rays* $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ *depending on wavelength* λ.

$\mathbf{d}(x,y) = \mathbf{v} + x\,\mathbf{v}_x + y\,\mathbf{v}_y$ depends on the view direction $\mathbf{v}$ and its reference frame spanned by $\mathbf{v}_x$ and $\mathbf{v}_y$ (see Figure 4, left). The ray differentials are defined as the change of position and direction per change of sensor coordinate:

$$\frac{\partial \mathbf{R}}{\partial x} = \left(\frac{\partial \mathbf{p}}{\partial x}, \frac{\partial \mathbf{d}}{\partial x}\right) \quad \text{and} \quad \frac{\partial \mathbf{R}}{\partial y} = \left(\frac{\partial \mathbf{p}}{\partial y}, \frac{\partial \mathbf{d}}{\partial y}\right).$$

Igehy [Ige99] presented closed-form solutions for transport (a ray travelling in free space), reflection and refraction. For more complex camera models (e.g. the thin-lens model), finite differences can be used to compute the initial directional differential.

Dispersion When a ray hits a specular surface it is reflected and/or refracted depending on the nature of the object and the incoming direction. The strength of refraction depends on the ratio of indices of refraction (IOR) between the original and the entered medium, commonly denoted by $\eta(\lambda)$. Often in non-spectral rendering, $\eta(\lambda)$ is defined to be constant. However, dispersion only occurs if $\eta(\lambda)$ is not constant but depends on the wavelength in vacuum λ (Fig. 3).Throughout the rest of the paper, we will shorten the notation of $\eta(\lambda)$ to just η for better readability. Refraction on an interface with normal $\mathbf{n}$ results in a new ray

$$\mathbf{p}' = \mathbf{p} \quad \text{and} \quad \mathbf{d}' = \eta\,\mathbf{d} - \mu\,\mathbf{n}, \quad \text{where}$$
$$\mu = \eta\,(\mathbf{d}\cdot\mathbf{n}) + \sqrt{1 - \eta^2 + \eta^2\,(\mathbf{d}\cdot\mathbf{n})^2}.$$

Consequently, the refracted direction also depends on the wavelength λ.

4. Our approach: Spectral ray differentials

For a ray $\mathbf{R}(\lambda) = (\mathbf{p}, \mathbf{d})$ associated with wavelength λ, we want to compute the partial differentials $\frac{\partial \mathbf{p}}{\partial \lambda}$ and $\frac{\partial \mathbf{d}}{\partial \lambda}$. Initially, if the ray starts at the light or the sensor, its spectral differentials are zero, in contrast to classic ray differentials. When the ray is traced through the scene, the differential is updated for all transport, reflection and refraction events. For transport and reflection (which are wavelength-independent) the equations proposed by Igehy [Ige99] can be used directly by substituting λ as the differential variable instead of the sensor coordinates (Fig. 4, left). However, for a dispersive refraction event (Fig. 4, right) the new direction of the ray now changes with the wavelength-dependent index of refraction η. The

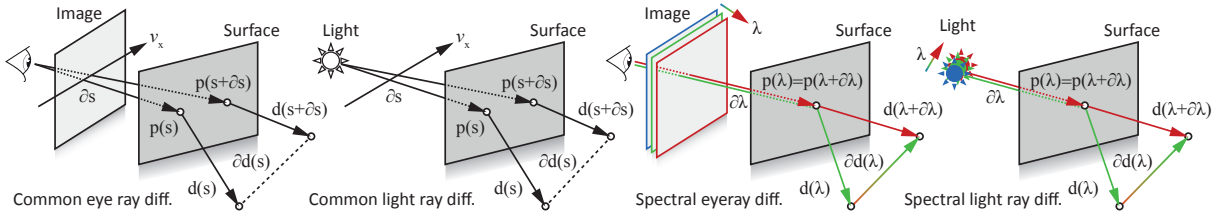

Figure 4: *Notation for, and the concept of the common* (left half) *and our spectral* (right half) *ray differentials.*

spectral ray differentials w. r. t. the wavelength λ

$$\frac{\partial \mathbf{R}}{\partial \lambda} = \left(\frac{\partial \mathbf{p}}{\partial \lambda}, \frac{\partial \mathbf{d}}{\partial \lambda} \right)$$

are updated for an interaction with a dispersive interface by:

$$\frac{\partial \mathbf{p}'}{\partial \lambda} = \frac{\partial \mathbf{p}}{\partial \lambda} \quad \text{and} \quad \frac{\partial \mathbf{d}'}{\partial \lambda} = \frac{\partial (\eta \mathbf{d} - \mu \mathbf{n})}{\partial \lambda}.$$

Note that $\mathbf{p}$, $\mathbf{d}$, $\mathbf{n}$, η and μ are functions of λ. Since refraction does not change the *positional* differential, we are only interested in finding the *directional* spectral differential.

4.1. Directional spectral differential

For better readability, we denote the cosine of the angle between the incoming ray direction and the surface normal as $\theta = \mathbf{d} \cdot \mathbf{n}$. Its partial derivative $\frac{\partial \theta}{\partial \lambda}$ is computed by:

$$\frac{\partial \theta}{\partial \lambda} = \frac{\partial \mathbf{d}}{\partial \lambda} \cdot \mathbf{n} + \mathbf{d} \cdot \frac{\partial \mathbf{n}}{\partial \lambda}$$

The new spectral directional differential is then given by

$$\boxed{\frac{\partial \mathbf{d}'}{\partial \lambda} = \frac{\partial \eta}{\partial \lambda}\mathbf{d} + \eta \frac{\partial \mathbf{d}}{\partial \lambda} - \frac{\partial \mu}{\partial \lambda}\mathbf{n} - \mu \frac{\partial \mathbf{n}}{\partial \lambda}} \qquad (1)$$

with

$$\frac{\partial \mu}{\partial \lambda} = \frac{\partial \eta}{\partial \lambda}\theta + \eta \frac{\partial \theta}{\partial \lambda} + \frac{-\eta \frac{\partial \eta}{\partial \lambda} + \eta \frac{\partial \eta}{\partial \lambda}\theta^2 + \eta^2 \theta \frac{\partial \theta}{\partial \lambda}}{\sqrt{1 - \eta^2 + \eta^2 \theta^2}}.$$

Eq. 1 is our main theoretical contribution; please refer to the supplemental materials for derivation details.

Note that the formulation of the normal differential $\frac{\partial \mathbf{n}}{\partial \lambda}$ by Igehy is still valid for spectral differentials because it is solely based on the surface properties and the positional differential w.r.t. a given variable. In Igehy's case this is the screen coordinate x / y and in our case it is the wavelength λ.

4.2. Differential of η

The function η is defined as the ratio between the index of refraction for the original medium $n_0(\lambda)$ and the entered medium $n_1(\lambda)$. The differential of η is then computed by

$$\frac{\partial \eta}{\partial \lambda} = \frac{\frac{\partial n_0(\lambda)}{\partial \lambda} n_1(\lambda) - n_0(\lambda) \frac{\partial n_1(\lambda)}{\partial \lambda}}{n_1(\lambda)^2}.$$

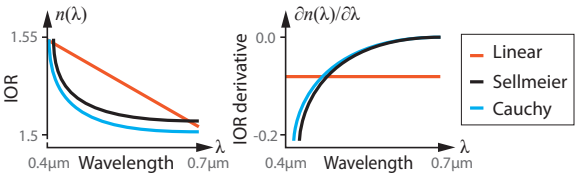

Figure 5: *Dependency of the index of refraction (IOR) on wavelength for a typical material (glass) using different models, and their derivatives.*

The function $n(\lambda)$ models the dependence between the wavelength (in vacuum) and the index of refraction for a certain material. It is often measured or approximated by physically-based models, e. g., the Cauchy and Sellmeier formulas [JW01], or even by a simple linear function (Fig. 5), where the derivative of $n(\lambda)$ is then constant.

Cauchy's model The Cauchy formula is defined as

$$n_{\text{Cauchy}}(\lambda) = B + \frac{C}{\lambda^2} + \frac{D}{\lambda^4} + \dots$$

where B, C, D etc. are empirically measured, material-specific coefficients. Its derivative w. r. t. wavelength is given by

$$\frac{\partial n_{\text{Cauchy}}(\lambda)}{\partial \lambda} = -\frac{2C}{\lambda^3} - \frac{4D}{\lambda^5} - \dots$$

Sellmeier's model Similarly, the generalized Sellmeier formula and its derivative are

$$n_{\text{Sellmeier}}(\lambda) = \sqrt{1 + \sum_k \frac{B_k \lambda^2}{\lambda^2 - C_k}}$$

$$\frac{\partial n_{\text{Sellmeier}}(\lambda)}{\partial \lambda} = -\frac{\sum_k \frac{B_k C_k}{(\lambda^2 - C_k)^2}}{\sqrt{1 + \sum_k \frac{B_k \lambda^2}{\lambda^2 - C_k}}}.$$

5. Tracing spectral differentials

Unbiased spectral rendering requires tracing monochromatic rays which define a unique path through the scene. To use spectral ray differentials when reconstructing a spectral image, the differentials need to be tracked and updated along the ray at interactions that occur, both for camera and light rays.

In our approach, each ray $(\mathbf{p}, \mathbf{d}, \lambda)$ is given by two 3D vectors (position and direction) and a wavelength, and its spectral ray differential $(\frac{\partial \mathbf{p}}{\partial \lambda}, \frac{\partial \mathbf{d}}{\partial \lambda})$ is described by two additional 3D vectors (positional and directional differential).

Note that the spectral differential is only non-zero when a dispersive interface was hit along a path. It is also important that while the dispersive refraction is the only event that causes spectral differentials to become non-zero (and does so only in the directional component), the classic transport and reflection differentiation still needs to be applied. Hence, both the positional and directional differentials have to be maintained during the transport.

In practice, at every non-dispersive event (transport and reflection) the spectral differentials are updated using the classic equations from Igehy [Ige99]. At every dispersive event, we use Eq. 1 to update the directional differential. Listing 1 shows pseudo-code for updating the current differential for a refraction event. It can be seen that the computational overhead is small and only a relatively low number of arithmetic operations needs to be performed.

Listing 1: *Directional spectral ray differential update (Eq. 1).*

```
vec3 updDiff(vec3 p, vec3 ∂p, vec3 d, vec3 ∂d, vec3 n, float η)
{
    vec3 ∂n = normalDifferential(∂p); // Cf. Igehy [1999]
    float θ = dot(d, n);
    float ω = sqrt(1 - sqr(η) + sqr(η)*sqr(θ));
    float μ = η*θ + ω;
    float ∂η = etaDifferential(η);
    float ∂t = dot(∂d, n) + dot(d, ∂n);
    float ∂O = (-η*∂η + η*∂η*sqr(θ) + sqr(η)*θ*∂t)/ω;
    float ∂μ = ∂η*θ + η*∂t + ∂O;
    return ∂η*d + η*∂d - ∂μ*n - μ*∂n;
}
```

6. Reconstruction step

Spectral ray differentials can be used for image filtering and reconstruction in different settings, such as dispersive light tracing (Sec. 6.1), ray tracing (Sec. 6.2) or photon mapping (Sec. 6.3).

The actual application of spectral differentials requires the knowledge of a ray's *footprint* at the position where the illumination is being reconstructed. Effectively this corresponds to finding out how the ray's hit position changes when changing its wavelength by a differentially small amount. Same as Igehy [Ige99] and the approaches that build on ray differentials we use the first-order approximation for spatially displacing the ray according to its differential

$$\mathbf{R}(\lambda + \Delta\lambda) - \mathbf{R}(\lambda) \approx \Delta\lambda \frac{\partial \mathbf{R}(\lambda)}{\partial \lambda} = \Delta\mathbf{R}$$

to estimate the change of a sample from one wavelength λ to a different wavelength $\lambda + \Delta\lambda$. In Sec. 7.5 we analyze the consequences of this.

Notice, however, that as opposed to the classic ray differentials the spectral reconstruction footprint of a ray is only one-dimensional. This is simply a consequence of the spectral domain having just a single dimension.

6.1. Light tracing

In light tracing, rays starting at the light are traced through the scene and eventually connected to the sensor. All paths that contain a dispersive interface can result in a non-zero spectral ray differential. To utilize spectral ray differentials for reconstruction, we can either apply a differential-guided splatting of light samples or a gathering-type image-space filtering approach. We describe both variants in the following.

Splatting This approach directly splats the spectral positional differential (a world space offset) of every light path onto the sensor by drawing a line from $\rho(\mathbf{p} - \Delta\mathbf{p})$ to $\rho(\mathbf{p} + \Delta\mathbf{p})$, where $\rho \in \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is a projective mapping, $\mathbf{p}$ is the ray's hit position and $\Delta\lambda$ is the spectral neighborhood. The size of the spectral neighborhood $\Delta\lambda$ inversely depends on the number of spectral bands. In general, larger values oversmooth the estimate, while smaller values lead to results that are less biased but remain more noisy. The actual colors for each fragment of the line are computed as an extrapolation in the spectrum based on the wavelengths $\lambda \pm \Delta\lambda$ that correspond to the start and end point. The spectral values are first converted to XYZ using the CIE color matching functions [CIE32] and then into the RGB space of the output device. For efficiency a 1D lookup table is used to represent the $\lambda \rightarrow$ RGB mapping. Additionally, to preserve edges induced by the receiver geometry, each fragment along this line is weighted by a term which gives higher weight to fragments with similar geometric properties comparable to joint-bilateral filtering [ED04, PSA*04]. These bilateral weights are obtained by storing the geometric properties (positions and normals) of the receiver surface where the connecting camera rays hit it.

In reality the footprint over the entire spectrum might potentially not be a line or a connected curve, but can consist of multiple, disconnected segments due to the possible divergence of rays with different λ. The correct reproduction of this effect depends only on how finely the spectrum is sampled and obviously is beyond the scope of our approximation. However, the occurrence of this phenomenon is comparatively very rare.

Gathering The other option is, along with producing a spectral radiance image $L(\mathbf{x}, \lambda)$, to create an additional image where the positional differential of each light path is averaged at each pixel. Practically, whenever a light sample is connected to the sensor, we accumulate its positional differential at the sensor location $\mathbf{x} = \rho(\mathbf{p})$. The distribution of offsets is meaningfully represented by a mean if a single caustic is projected onto a pixel and becomes gradually less precise if multiple divergent light paths are averaged, e. g., in the case of overlapping caustics.

We now use two assumptions to filter the image L. First, we assume the projected offset $\Delta\mathbf{x} = \rho(\Delta\mathbf{p}) \in \mathbb{R}^2$ varies smoothly within the neighborhood of each pixel at position $\mathbf{x}$. Second, we assume that the first-order approximation $L(\mathbf{x}, \lambda + \Delta\lambda) \approx L(\mathbf{x} + \Delta\mathbf{x}, \lambda)$ can predict radiance at one position and one wavelength from the differential and the radiance at different positions and different wavelengths. If this holds, we can simply filter the radiance image with a 1D linear kernel with the size and orientation of $\pm\Delta\mathbf{x}$. At every location $\mathbf{x}$ we average the values between $L(\mathbf{x} - \Delta\mathbf{x})$ and $L(\mathbf{x} + \Delta\mathbf{x})$, effectively corresponding to a convolution with a (spatially-variant) linear kernel. Same as in splatting we employ the bilateral weights based on the geometry of the receiver.

6.2. Eye tracing

Eye tracing traces rays and spectral ray differentials starting from the eye until they arrive at a point $\mathbf{p}_e$ which emits light (e. g., an emitting surface or an environment map). The resulting spectral positional differential describes the offset $\Delta\mathbf{p}$ which is applied to $\mathbf{p}_e$ when the wavelength λ changes. For reconstruction, we want to estimate the change in sensor location $\Delta\mathbf{x}$ when altering λ. Thanks to the reciprocity of light transport and the symmetry of ray differentials, this is again $\Delta\mathbf{x} = \rho(\Delta\mathbf{p})$. Therefore, we can just accumulate the average positional differential and proceed with gathering or scattering as described for light tracing.

6.3. Photon mapping

In the first pass photon mapping starts rays from the light, however, in contrast to light tracing they are not connected to the sensor but stored in a photon map. The second step then traces gathering camera rays and density estimation is used to connect them to the light paths stored in the photon map.

Our approach can be also used to extend photon mapping. We emit monochromatic photons and track their spectral differentials. The photon map then stores the spectral differentials in addition to the photon position, direction, wavelength and flux. In the second pass, monochromatic rays are started from the eye, again tracking spectral differentials. When performing density estimation, instead of using an eye ray's hit position to estimate the density of nearby photons, the density estimation is done at multiple wavelengths and positions, within the range defined by the ray differential, but for both light and eye rays. Essentially, the photon map becomes the differential of the entire light function: it describes where photons land, but also the first order approximation of where they would move if the wavelength was changed by a differential increment.

7. Results

We present implementations for spectral differentials in three common image synthesis approaches (offline light and eye tracing and interactive photon mapping, all implemented on GPU using OpenGL and GLSL), show how spectral differentials could be used for interactive artistic control of physically based rendering and conclude with an additional analysis of our approach. Performance is reported for an NVidia GeForce GTX 770 graphics card. All the presented images were rendered at the resolution of 512^2.

7.1. Application: Light tracing

Fig. 6 shows results of light tracing in combination with spectral ray differentials. We present six scenes, first three being simple and educational and the last three demonstrating more complex and realistic refractive objects (see the caption for the respective render times). The "naïve" results are generated with discrete sampling of the spectrum at regular intervals. We decided to compare against this approach because, despite being overly simplistic, there are still cases when it is used in practice. The "jittered" results are, in contrast, produced by stochastically sampling the spectrum. Both the naïve and the jittered results are generated using 7 spectral bands for the traced samples. The reference is generated using 35 jittered spectral bands and 2–4 times more samples (depending on the scene). All scenes show caustics on a diffuse white surface from objects with an IOR of 1.6–1.4 lit by a white beam light source from the top. The rightmost column shows a line integral-convolution (LIC) visualization [CL93] of the projected positional spectral ray differential.

First, we note that a naïve 7-band spectral rendering suffers from visible spatial aliasing artifacts (Fig. 6, naïve). Jittering the spectrum (Fig. 6, jittered) removes the aliasing but generally contains more noise (i. e., the error is only distributed differently if an equivalent number of samples is used). For both of these spectral-sampling variants we first applied an uniform Gaussian filter with a magnitude just large enough to remove the respective aliasing and noise artifacts. As expected this removes most details in the caustics as well. We then applied our splatting and gathering reconstruction techniques (Sec. 6). Both splatting and gathering work well in the simple scenes with only one simple structured caustic. Gathering fails to remove noise or leads to over-blurring in cases where the accumulated positional differential estimate is noisy as well or when multiple divergent caustics overlap at a same location (e. g., Fig. 6, e and f). However, splatting does not suffer from these problems as each sample is reconstructed independently, and consequently its application removes all visible noise and in the jittered variant leads to results visually indistinguishable from the reference. Notice how details like the diagonal in Fig. 6, c, the diagonal streaks in Fig. 6, d or the overall detail in Fig. 6, e and f, are preserved in comparison to the original unfiltered solutions. Full versions of all the rendered images are available in the supplemental materials.

We also analyze the convergence (in terms of the root-mean-square error, RMSE) of our splatting reconstruction

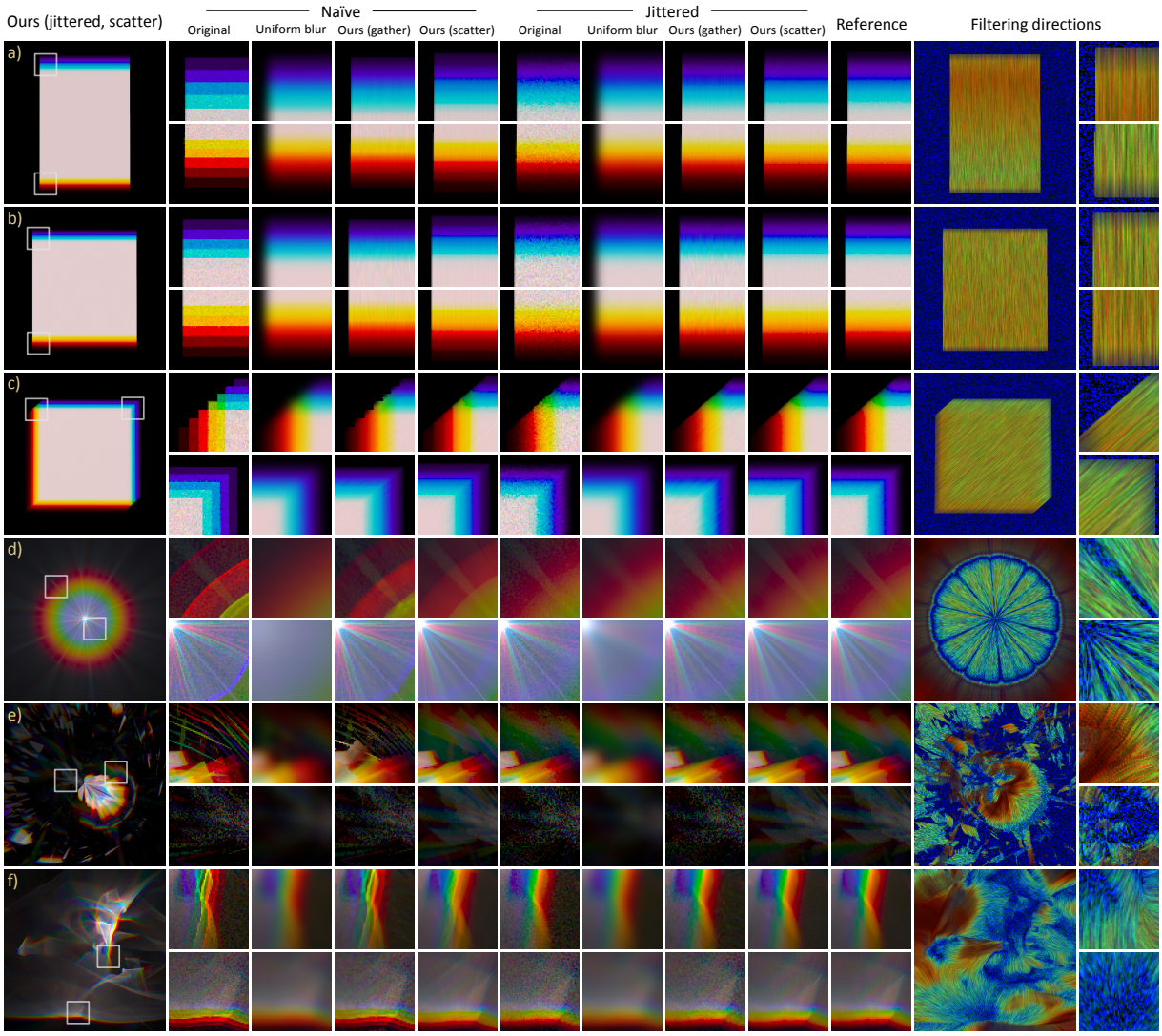

Figure 6: *Analysis of caustics scenes discussed in detail in Sec. 7.1. The dispersive objects are:* (a) *single planar interface,* (b) *triangular prism,* (c) *two slabs rotated against each other by 90 degrees,* (d) *sphere,* (e) *diamond, and* (f) *Blender monkey. The rendering times for these scenes were (naïve/jittered vs. reference):* (a,b) *5 seconds vs. 1 minute,* (c) *10 seconds vs. 2 minutes,* (d) *15 seconds vs. 4 minutes,* (e) *2 minutes vs. 60 minutes,* (f) *1.5 minutes vs. 30 minutes.*

compared to the original solutions for both the naïve and jittered variants in Fig. 7. It can easily be seen that our solutions are not only numerically closer to the reference in all our test cases, but also converge to it much faster. This is thanks to the fact that the splatted samples are 1D lines as opposed to just point samples in the original solutions.

7.2. Application: Eye tracing

Fig. 8 shows results of our eye tracing implementation for three sample scenes (glass icosahedron, diamond gemstone and monkey composed of water). The reference is again generated using 35 jittered spectral bands and 40 samples per pixel, the naïve and jittered variants used 5 spectral bands and 4 samples per pixel.

The comparison is similar to the light tracing case (Sec. 7.1), we however only compare to our gathering reconstruction, since the rendering and the resulting reconstruction natively take place in the image space. The artifacts inherent to the naïve and jittered variants are present here in a similar form, as expected. Our gathering reconstruction also suffers from similar problems (noisy guide images of the bilateral filter and overlapping positional differentials), but in

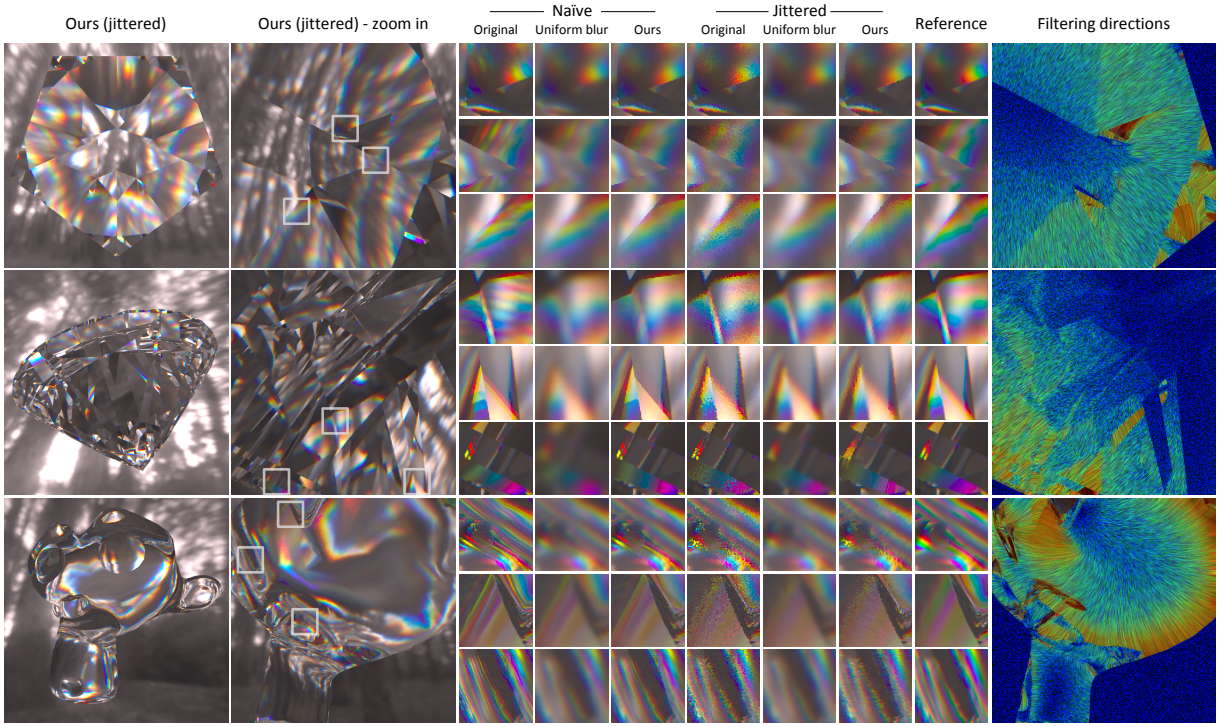

Figure 8: *Comparison of three dispersive sample scenes – icosahedron, gem and Blender monkey* (vertical) *– computed using dispersive ray tracing and reconstructed using different methods (*horizontal*) discussed in detail in Sec. 7.2.*

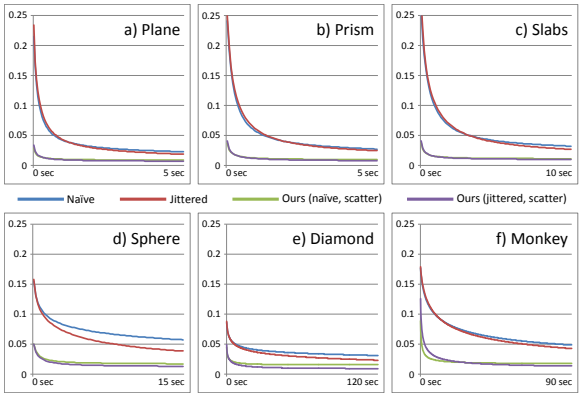

Figure 7: *Convergence for the caustic result images Fig. 6 (RMSE between the result and the reference images).*

general produces results with less noise and closer to the reference. We opted for rather conservative settings regarding blur magnitudes and bilateral weighting, so that some noise can remain but the edges in the image are respected. The remaining problems, however, are only visible at strong close-up views and the rendered objects under normal viewing conditions generally look very plausible.

7.3. Application: Photon mapping

Fig. 9 shows image-space dispersion in an interactive rendering setting using spectral differentials for eye and light paths at the same time. To generate the images with approximate reflection, refractions and caustics, we use a GPU-based photon mapper based on splatting [ML09] in combination with caustic mapping [WD06]. Along with the eye-refraction and the caustic, the spectral differential is computed and averaged in screen space per-pixel. In a final pass, a 1D filter with 5 jittered taps in the direction defined by the positional spectral differential is used to convolve the RGB image with RGB spectral response curves. The filter is very fast, requiring roughly 1 ms per megapixel.

7.4. Artistic control

Artistic control over physically-based image synthesis is difficult to achieve. Our positional differential offset field can be used to guide artistic effects and can be controlled locally in image space by simple scaling. Starting from an image without dispersion (Fig. 10, a) the user paints a multiplicative 2D or 3D control field (Fig. 10, c, shown inverted) and the system interactively produces a plausible dispersed image by applying the filter step using an offset field that is modified by the user guide, leading to the result Fig. 10, b. Note how such local and interactive control would be very difficult – if

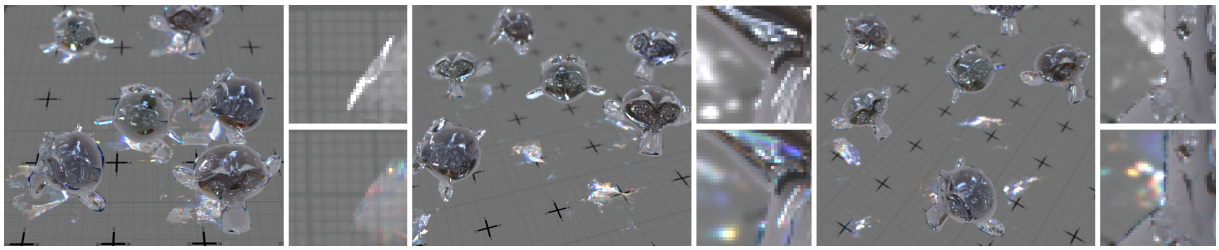

Figure 9: *Frames from the supplemental video, showing interactive image-space dispersion using spectral differentials (2048^2 caustic map, 40 ms rendering, ca. 1 ms filtering). Insets compare without* (top) *and with* (bottom) *image space dispersion.*

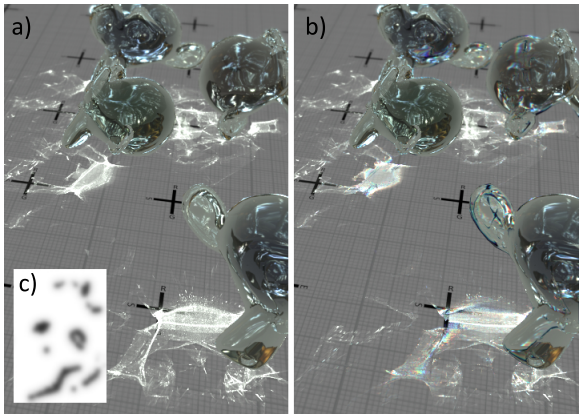

Figure 10: *Interactive dispersion editing (See Sec. 7.4).*

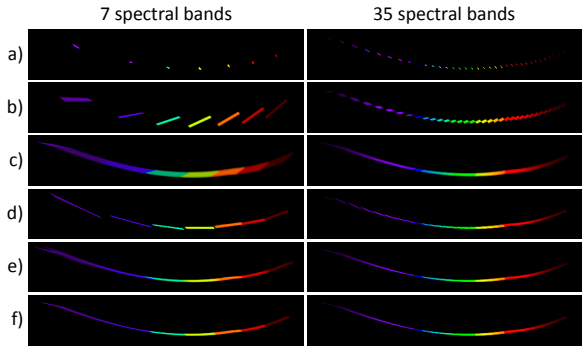

Figure 11: *Narrow pencil of light dispersed by a curved interface. We compare the 0^{th}* (a) *and (our) 1^{st} order* (b,c,d,e) *differential approximation to the reference* (f). *Nonlinearities of the interface and the transport itself cause deviation of our approach in this difficult case, but with increasing number of spectral bands it converges to the reference, as opposed to the 0^{th} order approach. This is a limitation intrinsic to all 1^{st} order approximations, including the classic ray differentials.*

not impossible – to achieve by modifying physical surface parameters such as IOR on the scene surface.

7.5. General analysis

The limitations of the first-order approach (as discussed in section Sec. 6) are demonstrated in Fig. 11, and are intrinsic to all such approximations, including the standard ray differentials. We compare renderings of a caustic resulting from a narrow beam of light entering a curved refractive object. Fig. 11, a, shows the naïve variant of sampling the spectrum, resulting in discrete discontinuous sampling of the caustic. Fig. 11, b, c, d, e, show the caustic rendered with spectral differentials – rows b, c show (naïve and jittered) finite differences (which essentially correspond to tracing a second 'virtual' ray alongside the actual one) and rows d, e our (naïve and jittered) analytic differentials, leading to more continuous behavior closer to the reference (Fig. 11, f) thanks to the fact that we take into account also differentials of the normals which are influenced by the object curvature.

While our filtering results in images with less noise, it might not be immediately obvious if they actually should be closer to a reference, as any blur is resulting in a biased solution. The plot of the RMSE in Fig. 7 however shows, that even with the blur, our results are closer to a reference and converge to a very similar solution.

8. Discussion and Conclusion

Limitations Our gathering-based implementations currently assume that at a single image location the dispersive caustic is dominated by photons that are dispersed at one interface only (or from multiple interfaces that are tilted around only one axis). This is not the case, e. g., when two caustics cross each other and for transparent surfaces. In such areas, our approach gracefully degrades to the unfiltered original. The splatting-based solution is more application-specific but does not suffer from this limitation. In conclusion, the scattering solution is to be preferred but software limitations or performance requirements can make gathering a necessary alternative that we find important to have analyzed.

A common problem of image reconstruction techniques, such as our proposed approach, is that they introduce bias

into the final result due to the filtering of sample colors in the image domain. However, on the other hand noise is reduced at the cost of visually more appealing blur. A progressive version that shrinks the spectral filter support to zero in the limit would not suffer from this issue and remains an interesting topic for future work.

As in the previous work on differentials it is to be assumed that the differentiated function is sufficiently smooth. Spectrally-varying absorption or emission does change the function smoothness and understanding its effect on our approach requires further investigation.

Finally, we did not consider distribution effects or participating media. It should be possible to apply spectral ray differentials to glossy or diffuse light transport, but future work is required.

Conclusion We proposed an approach to reduce chromatic noise in dispersive effects such as caustics, and applied it in offline ray-tracing as well as interactive GPU-based photon splatting algorithms. In future work, we would like to combine spectral and classic differentials and apply differentials to filtering of other advanced phenomena such as diffraction or polarization.

Acknowledgements We thank Antti Oulasvirta for inspiring discussions, Oliver Klehm, Karol Myszkowski and Alexander Wilkie for proofreading, and the anonymous reviewers for their feedback.

References

[Arv86] ARVO J.: Backward ray tracing. In *Developments in Ray Tracing. ACM SIGGRAPH Course Notes* (1986), pp. 259–63. 2

[BW92] BOULT T. E., WOLBERG G.: Correcting chromatic aberrations using image warping. In *Proc. CVPR* (1992), pp. 684–87. 3

[CIE32] CIE: Commission internationale de l'Eclairage proceedings. Cambridge University Press, 1932. 5

[CL93] CABRAL B., LEEDOM L. C.: Imaging vector fields using line integral convolution. In *Proc. SIGGRAPH* (1993), pp. 263–70. 6

[DHS*05] DURAND F., HOLZSCHUCH N., SOLER C., CHAN E., SILLION F. X.: A frequency analysis of light transport. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3 (2005), 1115–26. 2

[ED04] EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph. (Proc. SIGGRAPH) 23*, 3 (2004), 673–678. 5

[FD09] FABIANOWSKI B., DINGLIANA J.: Interactive global photon mapping. In *Comp. Graph. Forum (Proc. EGSR)* (2009). 1, 2

[HESL11] HULLIN M., EISEMANN E., SEIDEL H.-P., LEE S.: Physically-based real-time lens flare rendering. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 4 (2011), 108. 3

[HHK*07] HERZOG R., HAVRAN V., KINUWAKI S., MYSZKOWSKI K., SEIDEL H.-P.: Global illumination using photon ray splatting. In *Comp. Graph. Forum (Proc. Eurographics)* (2007), vol. 26(3), pp. 503–13. 2

[HOJ08] HACHISUKA T., OGAKI S., JENSEN H. W.: Progressive photon mapping. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 27*, 5 (2008), 130. 2

[Ige99] IGEHY H.: Tracing ray differentials. In *Proc. SIGGRAPH* (1999), pp. 179–86. 1, 2, 3, 5

[Jen01] JENSEN H. W.: *Realistic image synthesis using photon mapping*. AK Peters, Ltd., 2001. 2

[JF06] JOHNSON M. K., FARID H.: Exposing digital forgeries through chromatic aberration. In *Proc. ACM Multimedia and Security* (2006), pp. 48–55. 3

[JW01] JENKINS F., WHITE H.: *LSC Fundamentals of Optics*. McGraw-Hill Education, 2001. 4

[Kaj86] KAJIYA J. T.: The rendering equation. In *Proc. SIGGRAPH* (1986), pp. 143–50. 2

[Kan07] KANG S. B.: Automatic removal of chromatic aberration from a single image. In *Proc. CVPR* (2007), pp. 1–8. 3

[Kil01] KILGARD M. J.: Chromatic aberration. CEDEC 2001 Talk. NVIDIA Corporation., 2001. 3

[KMN*04] KAKIMOTO M., MATSUOKA K., NISHITA T., NAEMURA T., HARASHIMA H.: Glare generation based on wave optics. In *Proc. Pacific Graphics.* (2004), pp. 133–40. 3

[Lai07] LAI A.: A compression method for spectral photon map rendering. In *Proc. WSCG* (2007). 2

[LP94] LAROCHE C. A., PRESCOTT M. A.: Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients, 1994. US Patent 5,373,322. 3

[ML09] MCGUIRE M., LUEBKE D.: Hardware-accelerated global illumination by image space photon mapping. In *Proc. HPG* (2009), pp. 77–89. 2, 8

[MWR12] MEHTA S. U., WANG B., RAMAMOORTHI R.: Axis-aligned filtering for interactive sampled soft shadows. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31* (2012), 163. 2

[MWRD13] MEHTA S. U., WANG B., RAMAMOORTHI R., DURAND F.: Axis-aligned filtering for interactive physically-based diffuse indirect lighting. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4 (2013), 96. 2

[PSA*04] PETSCHNIGG G., SZELISKI R., AGRAWALA M., COHEN M., HOPPE H., TOYAMA K.: Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. (Proc. SIGGRAPH) 23*, 3 (2004), 664–672. 5

[RIF*09] RITSCHEL T., IHRKE M., FRISVAD J. R., COPPENS J., MYSZKOWSKI K., SEIDEL H.-P.: Temporal glare: Real-time dynamic simulation of the scattering in the human eye. *Comp. Graph. Forum (Proc. Eurographics) 28*, 2 (2009), 183–92. 3

[SFES07] SCHJØTH L., FRISVAD J. R., ERLEBEN K., SPORRING J.: Photon differentials. In *Proc. GRAPHITE* (2007), pp. 179–86. 1, 2

[SW01] SUYKENS F., WILLEMS Y. D.: Path differentials and applications. In *Proc. EGSR* (2001), pp. 257–68. 1, 2

[Tho86] THOMAS S. W.: Dispersive refraction in ray tracing. *The Visual Computer 2*, 1 (1986), 3–8. 2

[WD06] WYMAN C., DAVIS S.: Interactive image-space techniques for approximating caustics. In *Proc. I3D* (2006), pp. 153–60. 8

[WD08] WYMAN C., DACHSBACHER C.: Reducing noise in image-space caustics with variable-sized splatting. *J Graphics Tools 13*, 1 (2008), 1–17. 3