An Improved Technique for Full Spectral Rendering

Michal Radziszewski
AGH, Krakow, Poland
mradzisz@student.agh.edu.pl

Krzysztof Boryczko
AGH, Krakow, Poland
boryczko@agh.edu.pl

Witold Alda
AGH, Krakow, Poland
alda@agh.edu.pl

ABSTRACT

In this paper we present an improved approach to full spectral rendering. The technique is optimized for quasi-Monte Carlo ray tracing, however the underlying physical theory can be applied to any global illumination scheme. We start with explanation of the necessity of full spectral rendering in any correct global illumination system. Then we present, step by step, a rendering scheme using full spectrum simulation. First, we give details on a random point sampling as a method of representing spectra, then we introduce improved spectral sampling technique, designed to reduce variance of image of wavelength dependent phenomena, and finally we show how to integrate the novel sampling technique with selected ray tracing algorithms.

Keywords: Full spectrum, quasi-Monte Carlo, ray tracing, rendering.

1 INTRODUCTION

The color phenomenon is caused by a spectral mixture of light, perceived by the human visual system. However, the human visual system cannot distinguish between arbitrary spectral light distributions. Different spectra, which are indistinguishable by human observers, are called metamers. The space of colors recognizable by human observers contains only three independent values, hence the popularity of three component color models.

There are many color models in computer graphics, however most are designed for a specific purpose only. The most common are: RGB designed for displaying images, CMYK for printing and HSV for easy color selection by user. All of these models are to some degree hardware dependent. There is, however, a standard model based on XYZ color space, which is independent of any hardware and can represent all the colors recognizable by a human observer. It was defined by the CIE (Comission Internationale de l'Eclairage) as three weighting functions to obtain x, y and z components from arbitrary spectra. Nevertheless, neither of these models is well suited for rendering, where direct calculations on spectra are the only way to produce correct results [4, 9].

2 NECESSITY OF FULL SPECTRUM

The RGB model is often used for rendering color images. However, this is an abuse of it, since RGB based rendering does not have any physical justification. The model was designed for storage and effective display of images on a monitor screen, but not for physically accurate rendering. The light reflection computation under the assumption of elastic photon scattering is performed by a multiplication of a spectrum that represents an illumination and a spectrum describing a surface reflectance. This multiplication actually must be performed on spectral distribution functions, not on RGB triplets, in order to get proper results.

The RGB based reflection of white light, or light with smoothly varying spectrum, from a surface with smoothly varying reflectance, typically does not produce substantial inaccuracies. However, when at least one of spectra has large variation, the simulation using RGB model becomes visibly incorrect (see Figure 1, for example). Moreover in global illumination, due to multiple light scattering, even white light becomes colorful, causing scattering inaccuracies to accumulate. This makes RGB based global illumination results unable to accurately capture the physical phenomena.

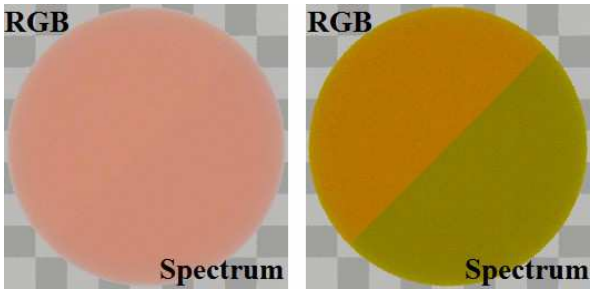

Figure 1: Left image: copper sphere illuminated by a D65 white light. Right image: copper sphere illuminated by a triangular spectral distribution stretched from 535nm to 595nm. Top left half: an RGB model with 645nm, 526nm and 444nm wavelengths. Right bottom half: our full spectral model. For clarity, only diffuse reflection is calculated.

In addition, the most visually distracting error from using an RGB model appears in simulation of phenom-

ena like dispersion. Whenever RGB based light, from a light source with almost parallel output rays, hits a prism, it is scattered into three bands instead of continuous full spectrum, and the rest of the image remains dark (see Figure 2), which looks unrealistic. Using a full spectrum representation gives a continous rainbow of colors. However, good-looking results may be obtained by an RGB representation if the light source angular distribution is conical and divergent enough. A similar trick is a basis of a simple Nvidia shader demo [5]. An address of the texture on a surface, which is seen through glass, is offseted independently for each channel. If texture data is blurred enough, the resulting 'spectrum' is smooth. Nevertheless, both of these methods do not have any physical significance, and obviously are incorrect, but, in some conditions, can look convincing.

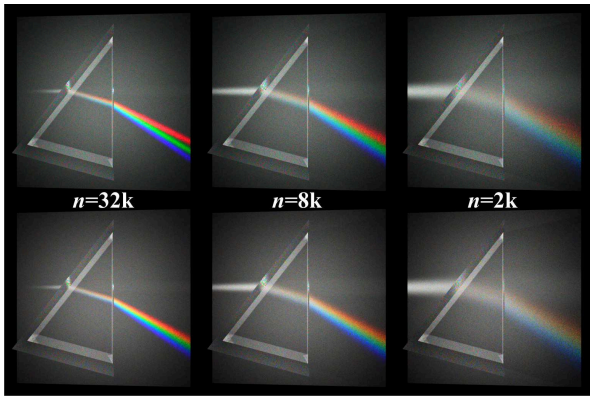

Figure 2: Dispersion on a prism. Top row: RGB model with 645nm, 526nm and 444nm wavelengths. Bottom row: physically correct full spectrum. The light collimation is controlled by a Phong-like function $I\cos^n(\phi)$, with exponent n decreased four times in each subsequent column, and intensity I doubled to compensate light scattering.

3 RELATED WORK

A general description of many popular color models can be found in Stone [16]. Devlin et al. [1] provide references related to data structures for full spectral rendering and algorithms for displaying spectral data. There are several works dedicated to simulation of particular spectral based phenomena. Wilkie et al. [22] simulated dispersion by means of classic (deterministic) ray tracing. Rendering of optical effects based on interference attracted a fair amount of attention. Reflection from optical disks is presented in Stam [15] and Sun et al. [17]. Algorithms for accurate light reflection from thin layers can be found in Gondek et al. [7] and Durikovic and Kimura [3]. The latter paper also shows how this algorithm can be run on contemporary GPUs.

Many papers present methods for representing and operating on spectral data. Peercy [12] designed a spectral color representation as a linear combination of basis functions, chosen in a scene dependent manner. Different algorithm using basis functions is described by Rougeron and Peroche [14]. It uses adaptive projection of spectra to hierarchical basis functions. Sun et al. [18] proposed a decomposition of spectra on smooth functions and set of spikes. Evans and McCool [4] used clusters of many randomly selected spectral point samples. Johnson and Fairchild [9] extended OpenGL hardware rasterization to support full spectra.

Dong [2] points that typically only a part of the scene needs a full spectral simulation and using RGB together with full spectrum can accelerate rendering at cost of only slight quality loss. Ward [21], however, designed a three-component model optimized for rendering, which typically produces images with an acceptable yet imperfect quality, but the model is not general enough and cannot simulate wavelength dependent phenomena like dispersion.

4 REPRESENTING FULL SPECTRA

Full spectral rendering requires an efficient method for representing spectral data. The most common techniques are based on linear combinations of carefully selected basis functions [12, 14, 18] and point sampled continuous functions [4]. Effectiveness of the linear combination approach is strongly dependent on the actual functions and their match to scene spectral distribution. However, the natural solution in Monte Carlo based rendering system is a random point sampling.

4.1 Random Point Sampling

Random point sampling produces noise at low sampling rate, but well-designed variants of this technique converge quickly. Point sampling can effectively handle smooth (like tungsten bulbs) light distributions and very narrow spikes (like neon bulbs) in the same scene. The two greatest strengths of this technique are: randomly selected wavelengths and well defined wavelength value for each spectral sample. The first one ensures correctness, since when more samples are computed, the more different wavelengths are explored, and due to the law of large numbers, the rendering result converges to the true value. The second allows simulating wavelength dependent effects like dispersion at the cost of additional color noise.

It is worth to note that wavelength dependent phenomena cannot be simulated correctly with algorithms based on linear combinations of basis functions with non-zero extent in wavelength space. Even if spectra are represented by unique non-zero coefficients, the corresponding basis functions still have some finite extent, which prevents from doing exact computations with explicit wavelength required.

The simplest approach to point sampled spectra is generation of a single spectral sample per light trans-

port path. However, according to Evans and McCool [4], this technique is inefficient, since it causes a lot of color noise. They proposed using a fixed number of several spectral samples (called a cluster of samples) traced simultaneously along a single light path, which substantially reduces variance with minimal computational overhead.

4.2 Basic Operations

The implementation of multiplication, addition, minimum, etc. operators are obvious, since it is enough to perform appropriate calculation per component, as in RGB model. However, when using full spectrum, computing luminance is a bit more difficult. Particularly, luminace of a spectrum which describes reflectivity of a surface, by definition must be in $[0,1]$ range.

However, computing luminance as a Monte Carlo quadrature of product of reflectance spectrum $r(\lambda)$ and scaled CIE y weighting function, may randomly lead to numerical errors causing luminance to exceed 1.0 threshold. The equation:

$$L \approx \sum_{i=1}^{n} \frac{r(\lambda_i)y(\lambda_i)}{p(\lambda_i)} \Big/ \sum_{i=1}^{n} \frac{y(\lambda_i)}{p(\lambda_i)}, \tag{1}$$

where $r(\lambda)$ is the reflectance, $y(\lambda)$ is CIE y weight and $p(\lambda_i)$ is a probability of selecting given λ_i, solves the issue. It guarantees that the luminance is in $[0,1]$ range, provided that $r(\lambda)$ is also in the specified range.

Wavelength dependent effects can be handled as proposed by Evans and McCool [4] for specular dispersion – by dropping all but one spectral sample from a cluster. This is done by randomly selecting a sample to preserve, with uniform probability. All the samples, except the selected one, are then set to zero, and the power of the chosen one is multiplied by the cluster size. Then the wavelength parameter becomes well defined, and further computations are performed with usage of its actual value. However, when simulated phenomena are not optically perfect, like in Phong-based glossy refraction, it may be more efficient to trace the whole cluster, scaling power of each sample independently. We examine this approach in detail in the next section.

5 SAMPLING OF SPECTRA

Evans and McCool [4] simulate wavelength dependent phenomena by tracing only one spectral sample per path. This particular approach is always correct, and is necessary when a phenomenon is optically perfect, such as refraction on idealized glass. However, when the scattering is not ideal, dropping all but one spectral sample from a cluster, while still being correct, might be extremely wasteful and inefficient. In this section we propose a substantially improved technique.

5.1 Single Scattering Model

For testing purpose, a refraction model with an adjustable, wavelength dependent refraction and imperfection introduced by Phong-based scattering [13], with controllable glossiness is used. An extension to Walter et al. microfacet based refraction [20] supporting dispersion gives better results, but their model is much more complicated and therefore would make evaluation of spectral sampling difficult. Nonetheless, since we have never made assumptions about scattering model, our results are general and, as we have tested, applicable to any wavelength dependent phenomena. For clarity, all tests are based on a single scattering simplification (i.e. light is refracted once, when it enters into glass only). The x component in CIE XYZ space in outgoing direction ω_o is then described by the following formula:

$$I_{CIEx}(\omega_o) = \int_\Lambda \int_\Omega f_s(\omega_i, \omega_o, \lambda) L_\lambda(\omega_i, \lambda) \cdot \cdot w_{CIEx}(\lambda) d\sigma^\perp(\omega_i) d\lambda, \tag{2}$$

where Λ is the space of all visible wavelengths, Ω is the space of all direction vectors, $L_\lambda(\omega_i, \lambda)$ is the radiance incoming from direction ω_i, w_{CIEx} is the CIE weight for x component, and $\sigma^\perp(\omega_i)$ is the projected solid angle measure. The y and z components can be evaluated in a similar way. In the rest of this section, the Formula (2) is written in a simplified, still not confusing, form:

$$I = \int_\Lambda \int_\Omega f(\omega, \lambda) L(\omega, \lambda) w(\lambda) d\sigma^\perp(\omega) d\lambda. \tag{3}$$

5.2 Basic and Cluster Based Monte Carlo Estimators

The Monte Carlo method (Equation 14) can be applied to evaluate the two integrals from Formula (3), which leads to the following estimator:

$$I \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(\omega_i, \lambda_i)}{p_\Omega(\omega_i, \lambda_i)} \frac{w(\lambda_i)}{p_\Lambda(\lambda_i)} L(\omega_i, \lambda_i), \tag{4}$$

where p_Ω is the probability of selection of a given ω_i evaluated with the $\sigma^\perp(\omega)$ measure on Ω and p_Λ is the probability of selection of a given λ_i. Quality of this estimator, and all the further estimators in this section, relies on the assumption that scattering model offers proper importance sampling (Equation 15), i.e. $f(\omega, \lambda) \propto p_\Omega(\omega, \lambda)$ is roughly satisfied. However, this basic estimator is inefficient, because it forces the numbers of spectral and directional samples to be equal. Each directional sample requires additional rays to be traced, which is computationally expensive, while spectral samples are almost for free. This explains the advantage of clusters of spectral samples over a single spectral sample approach.

The main improvement over Evans and McCool method is tracing a full cluster of spectral samples,

even when wavelength dependent phenomenon is encountered. Wavelength dependence can be defined precisely as the dependence of p_Ω on λ. If scattering is not wavelength dependent, directional sampling is not wavelength dependent as well, i.e. $p_\Omega(\omega,\lambda) \equiv p_\Omega(\omega)$. In our method, a particular spectral sample λ_i^s is selected at random from a cluster, and its value is used for sampling ω_i^s. This leads to the color estimator in the form:

$$\begin{aligned} I &\approx \frac{1}{NC}\sum_{i=1}^{N}\sum_{j=1}^{C}\frac{f(\omega_i^s,\lambda_i^j)}{p_\Omega(\omega_i^s,\lambda_i^s)}\frac{w(\lambda_i^j)}{p_\Lambda(\lambda_i^j)}L(\omega_i^s,\lambda_i^j) = \\ &= \frac{1}{NC}\sum_{i=1}^{N}\frac{1}{p_\Omega(\omega_i^s,\lambda_i^s)}\cdot \\ &\quad\cdot\sum_{j=1}^{C} f(\omega_i^s,\lambda_i^j)\frac{w(\lambda_i^j)}{p_\Lambda(\lambda_i^j)}L(\omega_i^s,\lambda_i^j), \end{aligned} \tag{5}$$

where N is the number of traced clusters, C is the number of samples in each cluster, and p_Ω is the probability of selecting scattering direction, calculated for the selected wavelength λ_i^s. The estimator (5) can be more efficient than estimator (4), since it traces C spectral samples at the minimal additional cost. On the other hand, it may deteriorate the importance sampling quality significantly. This happens because all samples with potentially wildly different $f(\omega_i^s,\lambda_i^j)$ values are traced, and just one probability $p_\Omega(\omega_i^s,\lambda_i^s)$ which matches the shape of $f(\omega_i^s,\lambda_i^s)$ only, is used. Whenever a direction ω_i^s with low probability $p_\Omega(\omega_i^s,\lambda_i^s)$ is chosen at random, and at least one of the $f(\omega_i^s,\lambda_i^j)$ has a relatively large value in that direction, the value is no longer cancelled by the probability, leading to the excessively high variance in the rendered image. Moreover, the estimator (5) is incorrect whenever $\exists\lambda_i^s,\omega_i^s\colon\ p_\Omega(\omega_i^s,\lambda_i^s) = 0$ and $\exists\lambda_i^j\colon\ f(\omega_i^s,\lambda_i^j) > 0$, particularly when a wavelength dependent phenomenon is optically perfect, i.e. its f is described by a δ distribution. Thus, the initial version of our new approach is not always better than the traditional technique of tracing only one spectral sample. The question is when the new technique exhibits lower variance and when it does not.

5.3 Multiple Importance Sampling Estimator

Fortunately, the variance issue can be solved automatically. Simple modification of the estimator (5), which incorporates Multiple Importance Sampling [19] (see Appendix A), gives a better estimator with variance as low as possible in a variety of conditions. The new improved estimator is constructed from the estimator (5) multiplying each cluster by C and a weight W_i^s equal to:

$$W_i^s = \frac{p_\Omega(\omega_i^s,\lambda_i^s)}{\sum_{j=1}^{C} p_\Omega(\omega_i^s,\lambda_i^j)}, \tag{6}$$

where $p_\Omega(\omega_i^s,\lambda_i^s)$ is the probability with which the scattering direction is selected, and the values $p_\Omega(\omega_i^s,\lambda_i^j)$ are hypotethical probabilities of selecting the sampled direction if using λ_i^j value instead. This leads to the final estimator:

$$\begin{aligned} I &\approx \frac{1}{NC}\sum_{i=1}^{N}\frac{CW_i^s}{p_\Omega(\omega_i^s,\lambda_i^s)}\cdot \\ &\quad\cdot\sum_{j=1}^{C} f(\omega_i^s,\lambda_i^j)\frac{w(\lambda_i^j)}{p_\Lambda(\lambda_i^j)}L(\omega_i^s,\lambda_i^j) = \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sum_{j=1}^{C} p_\Omega(\omega_i,\lambda_i^j)}\cdot \\ &\quad\cdot\sum_{j=1}^{C} f(\omega_i^s,\lambda_i^j)\frac{w(\lambda_i^j)}{p_\Lambda(\lambda_i^j)}L(\omega_i^s,\lambda_i^j). \end{aligned} \tag{7}$$

Assuming that a scattering model provides proper importance sampling, the estimator (7) leads to a low variance result. Moreover, the estimator (7) is correct whenever scattering model is correct, i.e. whenever $\forall\omega,\lambda\colon\ f(\omega,\lambda) > 0\ \ p_\Omega(\omega,\lambda) > 0$, so it is applicable even to optically perfect wavelength dependent phenomena. However, in this case it does not provide any benefit over estimator (4). The comparison between the new estimators (5) and (7) and the previous single sample estimator (4) is presented in Figure 3. The glass sphere has linearly varying refraction from 1.35 for 360nm to 1.2 for 830nm and uses Phong based scattering with $n = 1000$. Images are created using only two 16-sample clusters, to show error more clearly.

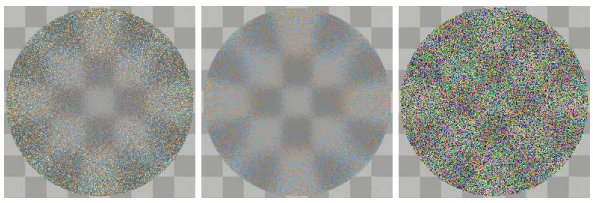

Figure 3: Comparison between new initial estimator (left), new improved estimator (middle) and previous method (right). The new initial estimator exhibits more variance due to lack of proper importance sampling. The color noise from single sample approach makes the rightmost image barely legible.

5.4 Generation of Clusters

In order to generate clusters efficiently, two issues have to be solved, namely: how many samples should a single cluster contain, and how to generate them. The number of spectral samples in a cluster is an important decision for achieving best possible performance. Unfortunately, optimal number of such samples is highly scene dependent. The more variation emission and reflectance spectra have, the more spectral samples a single cluster should contain. Assuming that a scene contains rather smoothly varying spectra (this assumption

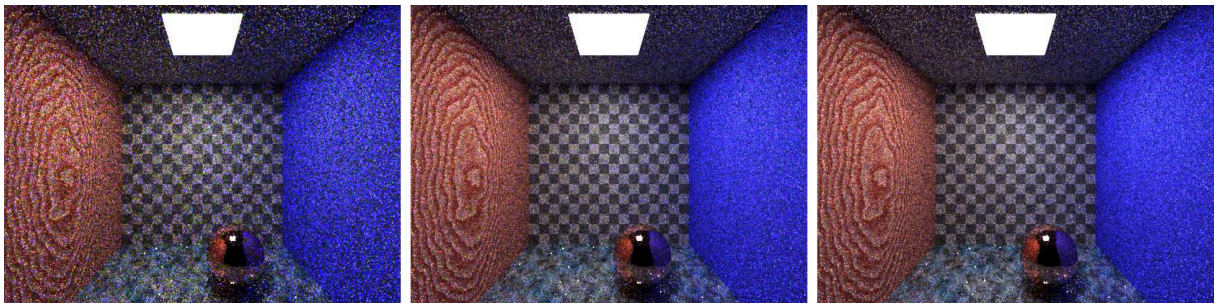

Figure 4: Selection of optimal number of spectral samples for a single cluster: 4 samples (left), 8 samples (middle), 12 samples (right). All images were rendered in 640x480, with 200k image samples (i.e. spectral clusters).

typically is satisfied), it is possible to balance excessive color noise and computational overhead. After a few tests we have found that eight spectral samples are optimal[1]. Four samples cause significant noise and twelve give barely visible improvement (see Figure 4). Rendering time differences between these images have been less than 1%, which confirms the efficiency of a cluster approach.

The efficient generation of spectral samples proves to be more difficult. Spectra should be importance sampled, but there are at least three factors, which should affect choice of p_Λ, namely: sensor (camera, human eye, etc.) sensitivity, light source spectral distribution and reflectance properties of materials. However, often only sensor is taken into account, and it is assumed that its sensitivity is well described by CIE y weighting function. Unfortunately, despite producing good quality grayscale images, importance sampling wavelength space with respect to the y function causes excessive color noise, and, contrary to common knowledge, is suboptimal. Ideally, a sampling probability should take into account all three x, y, and z components. After some experiments, we found that following probability gives good results:

$$p_\Lambda(\lambda) = N^{-1} f_\Lambda(\lambda),\ f_\Lambda(\lambda) = \frac{1}{\cosh^2(A(\lambda - B))}, \quad (8)$$

where $A = 0.0072nm^{-1}$ and $B = 538.0nm$ are empirically evaluated constants and $N = \int_{\lambda_{min}}^{\lambda_{max}} f_\Lambda(\lambda)d\lambda$ is a normalization factor. Results of this improved technique are presented in Figure 5.

Moreover, since spectra are typically smooth, sampling them with quasi-Monte Carlo (QMC) low discrepancy sequences instead of random numbers improves results. However, care must be taken when QMC sampling is applied to cluster based spectra. When a wavelength dependent effect is to be simulated, a single sample from the cluster has to be chosen. This choice is tricky due to peculiarities of QMC sampling. In case of true random numbers, selection of first sample from a cluster always works correctly. On the other hand, it is a serious error to select an every nth sample from a low discrepancy sequence. In the latter case, we assign a separate (pseudo)random sequence for a such selection of a spectral sample, in addition to sequence used for randomizing cluster samples. Results of QMC sampling are presented in Figure 5.

Figure 5: Various methods of sampling spectra. Top row: 2000K blackbody radiation. Bottom row: D65 spectrum. Left column: spectra sampled using random numbers and our importance sampling, with various numbers of samples. Middle column: comparison of luminance based importance sampling (top halfs) with our p_Λ (bottom halfs) using 128 spectral samples. Right column: spectra sampled using Sobol low discrepancy sequence and our p_Λ, using 4 and 8 spectral samples.

5.5 Results and Discussion

Some more comparison between single spectral sample approach and improved technique is presented in Figure 6. Images in top row use previous settings (refraction coefficient from 1.35 for 360nm to 1.2 for 830nm and glossiness coefficient $n = 1000$). Next, images in bottom row use much sharper settings (refraction coefficient from 1.5 for 360nm to 1.2 for 830nm and glossiness coefficient $n = 4000$). Images from first and second column are rendered to have approximately the same quality, and images from second and third column are rendered with the same number of samples (i.e.

[1] Due to Intel SSE instruction set optimization, our implementation requires the number of samples to be divisible by four.

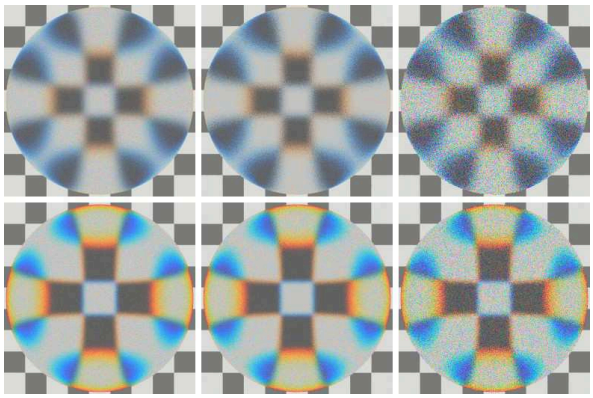

Figure 6: Imperfect refraction with dispersion. Top left image uses previous approach with a massive number of 900 samples per pixel. Top middle image uses new technique with just 50 samples per pixel, yet it has similar quality. Top right image again uses previous approach, but with 50 samples per pixel. However, gains from using the new technique are less spectacular when glossiness or dispersion is increased. Bottom row images use 900, 100, and 100 samples, respectively.

Settings	C	MIS	SSS
$n = 1000$ $\eta = [1.35, 1.20]$	1	$1.26 \cdot 10^{-1}$	$2.47 \cdot 10^{-1}$
	4	$6.67 \cdot 10^{-2}$	$2.02 \cdot 10^{-1}$
	16	$2.63 \cdot 10^{-2}$	$1.34 \cdot 10^{-1}$
	64	$1.22 \cdot 10^{-2}$	$7.56 \cdot 10^{-2}$
	256	$5.32 \cdot 10^{-3}$	$3.83 \cdot 10^{-2}$
$n = 4000$ $\eta = [1.50, 1.20]$	1	$2.07 \cdot 10^{-1}$	$2.46 \cdot 10^{-1}$
	4	$1.33 \cdot 10^{-1}$	$1.96 \cdot 10^{-1}$
	16	$7.39 \cdot 10^{-2}$	$1.29 \cdot 10^{-1}$
	64	$3.84 \cdot 10^{-2}$	$7.37 \cdot 10^{-2}$
	256	$1.74 \cdot 10^{-2}$	$3.72 \cdot 10^{-2}$

Table 1: Comparison of error of our method (MIS) and a single spectral sample approach (SSS), for C 8-sample spectral clusters per pixel. The error is evaluated as a difference between the tested image and the reference image, averaged over all pixels and color components. The pixel values are normalized to $[0,1]$ range.

traced rays). The average numerical error for various numbers of rays for scene from Figure 6 is summarized in Table 1.

Analysis of two limit cases could give more insight into how this new technique works, and when it is most effective. The analysis is based on the assumption that $f(\omega,\lambda) \propto p_\Omega(\omega,\lambda)$ is roughly satisfied. Otherwise, the multiple importance cannot help much in reducing variance. First, when wavelength dependence is negligible, all the scattering probabilities become more and more independent on λ: $p_\Omega(\omega_i^s,\lambda_i^j) \approx p_\Omega(\omega_i^s)$. The weight W_i^s then becomes:

$$W_i^s = \frac{p_\Omega(\omega_i^s,\lambda_i^s)}{\sum_{j=1}^{C} p_\Omega(\omega_i^s,\lambda_i^j)} \approx \frac{p_\Omega(\omega_i^s)}{\sum_{j=1}^{C} p_\Omega(\omega_i^s)} \to \frac{1}{C}, \quad (9)$$

and the estimator:

$$\begin{aligned} I &\approx \frac{1}{N}\sum_{i=1}^{N} \frac{W_i}{p_\Omega(\omega_i^s,\lambda_i^s)} \cdot \\ &\quad \cdot \sum_{j=1}^{C} f(\omega_i^s,\lambda_i^j) \frac{w(\lambda_i^j)}{p_\Lambda(\lambda_i^j)} L(\omega_i^s,\lambda_i^j) \to \\ &\to \frac{1}{NC}\sum_{i=1}^{N}\sum_{j=1}^{C} \frac{f(\omega_i^s,\lambda_i^j)}{p_\Omega(\omega_i^s)} \frac{w(\lambda_i^j)}{p_\Lambda(\lambda_i^j)} L(\omega_i^s,\lambda_i^j) \quad (10) \end{aligned}$$

which is an estimator of a simple, wavelength independent, scattering. On the other hand, when scattering becomes more and more glossy and wavelength dependence is significant, with probability close to one the f becomes close to zero for all directions except ω_i^s. The rare cases, when $f(\omega_i^j,\lambda_i^j)$ is large and $j \neq s$, have low weight W_i^s, and therefore cannot affect the estimator much. Moreover, all the probabilities but the selected one go to zero, and therefore W goes to one, which leads to estimator equal to:

$$\begin{aligned} I &\approx \frac{1}{N}\sum_{i=1}^{N} \frac{W_i^s}{p_\Omega(\omega_i^s,\lambda_i^s)} \cdot \\ &\quad \cdot \sum_{j=1}^{C} f(\omega_i^s,\lambda_i^j) \frac{w(\lambda_i^j)}{p_\Lambda(\lambda_i^j)} L(\omega_i^s,\lambda_i^j) \to \\ &\to \frac{1}{N}\sum_{i=1}^{N} \frac{f(\omega_i^s,\lambda_i^s)}{p_\Omega(\omega_i^s,\lambda_i^s)} \frac{w(\lambda_i^j)}{p_\Lambda(\lambda_i^j)} L(\omega_i^s,\lambda_i^j), \quad (11) \end{aligned}$$

which is equivalent to the one sample estimator. This behaviour of estimator (7) is presented in Figure 7.

The former approach to spectral rendering separates scattering into two cases: wavelength independent scattering, and costly simulation of wavelength dependent phenomena using single spectral sample estimator. On the other hand, our method does not depend on such classification. Due to automatically computed weights, it adjusts itself to these two limit cases, and to the broad spectrum of intermediate cases, when scattering is wavelength dependent, but imperfect. The computational cost of our method depends on strength of wavelength dependence and optical perfection of material. These factors cause the cost to increase, but it never exceeds the cost of single spectral sample estimator.

6 SAMPLING OF LIGHT TRANSPORT PATHS

In this section we describe an integration of our full spectral sampling with selected light transport algorithms – a case when there is more than one wavelength dependent scattering encountered on the same light path. The extension of single scattering approach for Path Tracing [10] and Bidirectional Path Tracing [19] is, however, obvious. The wavelength λ_i^s is selected once for a whole path, and reused at each scattering. The weight W_i^s is therefore computed for the

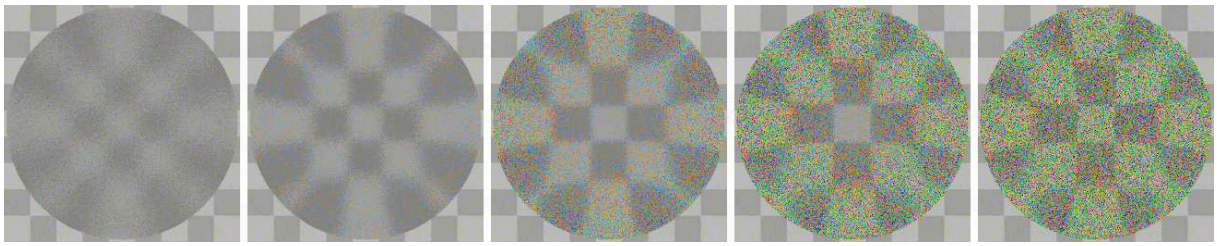

Figure 7: Analysis of behaviour of estimator (7) with increasing glossiness and wavelength dependence of scattering. Wavelength independent scattering (leftmost image). Optically perfect wavelength dependent scattering (rightmost image). Intermediate cases (middle). All the images are rendered with just four clusters.

whole path, using products of probabilities instead of probabilities of single scatterings. Assuming that the sampled path is build by recursively sampling f_s and tracing rays in sampled directions, the W_i^s is given by the following expression:

$$W_i^s = \frac{\prod_{k=1}^{m} p_\Omega(\omega_{ki}^s, \lambda_i^s)}{\sum_{j=1}^{C} \prod_{k=1}^{m} p_\Omega(\omega_{ki}^s, \lambda_i^j)}, \qquad (12)$$

where k is the number of a scattering event and m is the length of the sampled path. Intuitively, a weight W_i^s is a ratio of probability of generating the whole path using selected wavelength λ_i^s to the sum of probabilities of generating such a path using each wavelength from a cluster. If a light transport algorithm generates a path in a different way, or does not use a concept of light transport paths, the weight W_i^s has to be computed in a different manner.

The notable case, where spectral sampling causes difficulties, is Jensen's Photon Mapping, designed to work with RGB triplets [8]. There are two issues: first, there are no light transport paths, which connect light source and camera, and second, millions of individual photons have to be stored, causing excessive memory consumption if full spectrum is used to describe them. A recent work [11] addresses memory issues. Unfortunately, this algorithm converts photons' spectra to RGB prior to storing them in a map, and converts RGB to spectra again when searching through photons.

Our approach, on the other hand, is designed to converge *always* to the true result with increased number of photons, and therefore significant compression of spectral data is unsuitable. We trace and store clusters of photons with different wavelengths, instead of describing them by RGB triplets. First, in order to explore wavelength space properly, each emitted photon cluster must have individually chosen wavelengths. The obvious place for optimization is that one emitted photon cluster typically corresponds to several stored photon clusters, and therefore cluster wavelengths are stored once for each emission. Moreover, for storing energy, one can experiment with a non-standard floating point format instead of IEEE single precision. Using 8-sample clusters requires 32B of data for individual stored photon, not to mention an additional 32B for each emission, which is far more than 12B required by an RGB based implementation. If a compact float format with shared exponent is used, the latter can be compressed even to 4B, however, with potential loss of image quality. We have left this for further research.

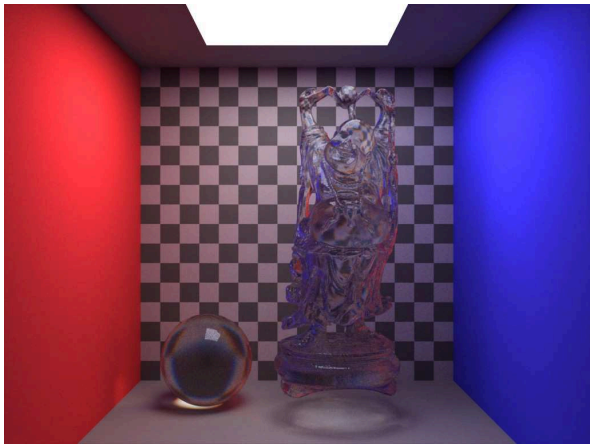

Figure 8: Full spectral rendering of a non-trivial scene. Dispersion is slightly exaggerated to render spectral sampling quality more prominent.

When a photon is about to be stored, its energy is multiplied by weight given by Equation (12), which accounts for all encountered wavelength dependent scattering events. In the second pass, rendering of photon map is performed. Camera rays should be weighted similarly prior to photon map lookups. In the classic Photon Mapping, photons are searched in a sphere centered around the intersection point. The sphere radius should be chosen carefully: too small causes noise and too large – blurriness. We extend this approach to wavelength search as well. If a photon cluster is decided to be used in a flux estimate by a sphere test, additional tests are performed on individual photons (with associated wavelengths) using a spectral search distance in a wavelength space. Similarly as with the spatial radius, the spectral distance must be chosen carefully.

7 CONCLUSIONS

We have presented an improved approach to full spectral rendering. Full spectral algorithms realize a model

which is necessary to achieve physically plausible illumination in 3D scenes. The result image rendered with proposed spectral sampling, extended Walter et al. microfacet refraction [20], and Bidirectional Path Tracing is presented in Figure 8. The computational cost of a full spectral simulation in comparison with an RGB model is significant only for simple scenes containing few primitives. The computational complexity of ray tracing typically is logarithmic with respect to the number of primitives, and independent of a color representation. Therefore, when a scene becomes sufficiently complex, the overhead of a physically correct algorithm becomes negligibly small. On the other hand, the memory overhead depends on a particular algorithm. It is negligible for Path Tracing and Bidirectional Path Tracing, but is substantial for Photon Mapping.

A MONTE CARLO ESTIMATORS

This section briefly describes Importance and Multiple Importance Sampling methods. Consult [6] and [19], for more details. Let I be the integral to evaluate:

$$I = \int_{\Psi} f(x) d\mu(x). \tag{13}$$

The basic Monte Carlo estimator of this integral is:

$$\tilde{I} \approx F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}, \tag{14}$$

where $\forall x\colon\ f(x) \neq 0\ \ p(x) > 0$.

A variance of estimator (14) usually can be decreased if the p is made near proportional to f, or at least to a part of it. This technique is called Importance Sampling. Particularly, when $p \propto f$, the variance is zero. However, to obtain normalization constant, f must be integrated analytically. This is impossible, otherwise Monte Carlo integration would not be necessary.

Suppose that there are i potentially good probability densities p_i for sampling f. If Importance Sampling is used, the p_i used for sampling $f(x)$ has to be chosen at algorithm design time. This can have disastrous consequences, if the p_i poorly matches the actual $f(x)$ shape. In this case, Importance Sampling can actually *increase* variance over sampling with uniform probability. However, Multiple Importance Sampling [19] has been designed to improve the Importance Sampling when the appropriate p_i cannot be chosen at the design time. The algorithm samples from each of these p_i and calculates the final estimator as a weighted sum of these samples:

$$\tilde{I} = \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} w_i(X_{ij}) \frac{f(X_{ij})}{p_i(X_{ij})},\ \forall x \sum_{i=1}^{n} w_i(x) = 1. \tag{15}$$

The appropriate choice of weights w_i:

$$w_i(x) = \frac{p_i(x)}{\sum_{j=1}^{n} p_j(x)} \tag{16}$$

is crucial for obtaining low variance results.

REFERENCES

[1] Kate Devlin, Alan Chalmers, Alexander Wilkie, and Werner Purgathofer. Tone reproduction and physically based spectral rendering. In *State of the Art Reports, Eurographics 2002*, pages 101–123, September 2002.

[2] Weiming Dong. Rendering Optical Effects Based on Spectra Representation in Complex Scenes. In *Computer Graphics International*, pages 719–726, 2006.

[3] Roman Durikovic and R. Kimura. GPU Rendering of the Thin Film on Paints with Full Spectrum. In *Proceedings of the IEEE Conference on Information Visualization*, pages 751–756, 2006.

[4] Glenn F. Evans and Michael D. McCool. Stratified wavelength clusters for efficient spectral monte carlo rendering. In *Graphics Interface*, pages 42–49, 1999.

[5] Randima Fernando and Mark J. Kilgard. *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*. Addison-Wesley, Boston, MA, USA, 2003.

[6] George S. Fishman. *Monte Carlo: Concepts, Algorithms and Applications*. Springer-Verlag, New York, USA, 1999.

[7] Jay S. Gondek, Gary W. Meyer, and Jonathan G. Newman. Wavelength dependent reflectance functions. In *SIGGRAPH 1994 Proceedings*, volume 28, pages 213–220, 1994.

[8] Henrik Wann Jensen. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[9] Garrett M. Johnson and Mark D. Fairchild. Full-spectral color calculations in realistic image synthesis. *IEEE Computer Graphics and Applications*, 19(4):47–53, 1999.

[10] James T. Kajiya. The rendering equation. In *SIGGRAPH 1986 Proceedings*, pages 143–150, New York, NY, USA, 1986.

[11] Gorm Lai and Niels Jorgen Christensen. A compression method for spectral photon map rendering. In *WSCG 2007 Proceedings*, pages 95–102, 2007.

[12] Mark S. Peercy. Linear color representations for full spectral rendering. In *SIGGRAPH 1993 Proceedings*, volume 27, pages 191–198, 1993.

[13] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[14] Gilles Rougeron and Bernard Peroche. An adaptive representation of spectral data for reflectance computations. In *EGRW 1997 Proceedings*, pages 127–138, 1997.

[15] Jos Stam. Diffraction shaders. In *SIGGRAPH 1999 Proceedings*, pages 101–110, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[16] Maureen Stone. *A Field Guide to Digital Color*. AK Peters, Natick, MA, USA, 2003.

[17] Yinlong Sun, David F. Fracchia, Mark S. Drew, and Thomas W. Calvert. Rendering iridescent colors of optical disks. In *EGRW 2000 Proceedings*, pages 341–352, 2000.

[18] Yinlong Sun, David F. Fracchia, Mark S. Drew, and Thomas W. Calvert. A spectrally based framework for realistic image synthesis. *The Visual Computer*, 17(7):429–444, 2001.

[19] Eric Veach and L. J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH 1995 Proceedings*, pages 419–428, 1995.

[20] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet Models for Refraction through Rough Surfaces. In *Eurographics Symposium on Rendering*, pages 195–206, Grenoble, France, 2007.

[21] Gregory J. Ward and Elena Eydelberg-Vileshin. Picture Perfect RGB Rendering Using Spectral Prefiltering and Sharp Color Primaries. In *EGRW 2002 Proceedings*, pages 117–124, 2002.

[22] A. Wilkie, R. Tobler, and W. Purgathofer. Raytracing of Dispersion Effects in Transparent Materials. In *WSCG 2000 Conference Proceedings*, 2000.