

# TRISTIMULUS VALUE RENDERING USING IMPORTANCE SAMPLING

Yoshihiro. Takemura<sup>†</sup>

Toru. Tamaki<sup>†</sup>

Bisser Raytchev<sup>†</sup>

Kazufumi. Kaneda<sup>†</sup>

<sup>†</sup>Department of Information Engineering, Graduate School of Engineering, Hiroshima University

## ABSTRACT

The paper proposes a novel method for rendering wavelength-dependent phenomena such as dispersion and interference of light in cooperation with Monte Carlo integration and importance sampling. The method, we call tristimulus value rendering, is simple and generates a physically correct image with less computational resource than spectral rendering.

## 1. INTRODUCTION

Physically based rendering is able to generate photorealistic images in various conditions such as displaying a product made of a new material and a scene with atmospheric and optical phenomena, and becomes increasingly important to many fields such as industrial design, movie productions, and so on.

RGB rendering is employed in many applications because R, G, and B intensities of light calculated by a renderer can be instantly used for displaying an image. RGB rendering calculates the intensity of light in RGB color space, i.e., both spectral radiance of light and spectral reflectance of surfaces are converted into RGB tristimulus values in advance of rendering (calculating reflected light from a surface) (see Fig. 1) [1].

RGB rendering, however, cannot generate physically correct images because the conversion from a spectral distribution into tristimulus values using color matching functions is a nonlinear operation. The resultant RGB values are not equal to the real values calculated by Spectral rendering [2].

Spectral rendering calculates spectral intensity of reflected light directly from spectral radiance of incident light and spectral reflectance of surfaces, and then converts the spectral intensity into RGB tristimulus values to display the image (see Fig. 1). Spectral rendering generates physically correct images, but the computational cost is usually expensive because many wavelength samples are required across a wide range of the wavelength of visible light.

We propose a novel method for sampling the wavelength of light to render wavelength dependent phenomena at which spectral rendering mainly targets. We introduce a Monte Carlo integration and an importance sampling to stochastically sample the wavelength across the range of visible light in consideration of human visual perception. The method, we call tristimulus value rendering, is simple but generates a physically correct image with less computational resource than that of spectral rendering.

## 2. PHYSICALLY BASED RENDERING

### 2.1. RENDERING EQUATION

Rendering in computer graphics is solving the rendering equation [3]. Explicitly writing wavelength of light,  $\lambda$ , the rendering equation is expressed by

$$L_o(\mathbf{x}, \vec{\omega}_o, \lambda) = L_e(\mathbf{x}, \vec{\omega}_o, \lambda) + \int_{\Omega} f_s(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o, \lambda) L_i(\mathbf{x}, \vec{\omega}_i, \lambda) |\vec{\omega}_i \cdot \vec{n}| d\vec{\omega}_i \quad (1)$$

Spectral radiance  $L(\mathbf{p}, \lambda)$  observed from a viewpoint is calculated by the rendering equation, where  $\mathbf{p}$  means the position of a pixel, and converted into tristimulus values, X, Y, and Z, considering human visual perception.

$$X(\mathbf{p}) = \frac{1}{k} \int_{\Lambda} L(\mathbf{p}, \lambda) f_{\bar{x}}(\lambda) d\lambda, \quad (2)$$

$$Y(\mathbf{p}) = \frac{1}{k} \int_{\Lambda} L(\mathbf{p}, \lambda) f_{\bar{y}}(\lambda) d\lambda, \quad (3)$$

$$Z(\mathbf{p}) = \frac{1}{k} \int_{\Lambda} L(\mathbf{p}, \lambda) f_{\bar{z}}(\lambda) d\lambda, \quad (4)$$

where  $X(\mathbf{p})$ ,  $Y(\mathbf{p})$ , and  $Z(\mathbf{p})$  are XYZ tristimulus values at pixel  $\mathbf{p}$ ,  $k = \int_{\Lambda} f_{\bar{y}}(\lambda) d\lambda$ ,  $f_{\bar{x}}$ ,  $f_{\bar{y}}$ , and  $f_{\bar{z}}$  are XYZ color matching functions with nonlinear characteristics, and  $\Lambda$  denotes the range of the wavelength of visible light.

To display an image on RGB display devices, XYZ tristimulus values are finally converted into RGB tristimulus values using the following linear equation:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad (5)$$

where  $M$  is a  $3 \times 3$  conversion matrix.

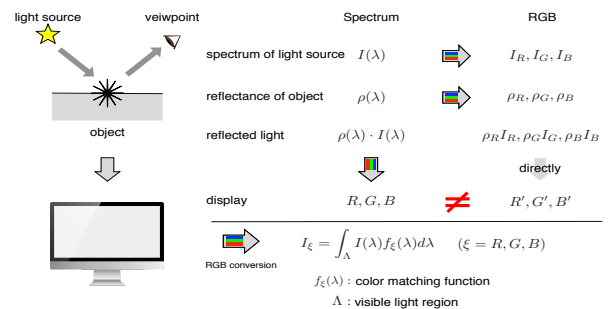


Figure 1: Framework of Spectral rendering and RGB rendering

## 2.2. SPECTRAL RENDERING

Spectral rendering solves the rendering equation in respect to the wavelength of light as described in Section 2.1. In Spectral rendering, one of the most important issue is how to sample the wavelength.

A uniform sampling is a simple and robust approach to sample the wavelength. When rendering optical phenomena such as dispersion, interference, and diffraction of light, we should set a small sampling interval to generate an adequate number of wavelength samples. On the other hand, when a spectral distribution of light is smooth, we may set a larger sampling interval. However, It is difficult to find the optimal sampling interval. Meyer developed a guideline [4] for selection of wavelengths using Gaussian quadrature with the opponent fundamentals. Anyway, to exactly represent a non-smooth spectral distribution, we need a lot of wavelength samples. This results in a large computational cost. To address the problem, several primary researches in spectral rendering have been done on sampling the wavelength [5] and determining the number of wavelength samples [6].

Radziszewski et al. [7] proposed a full spectral rendering in which the wavelength of light is sampled by using multiple importance sampling. The method randomly selects one of the wavelengths to use for path propagation, and weights the path contribution by using a multiple importance estimator. Wilkie et al. [8] proposed a method called *hero wavelength spectral sampling* to simplify and optimize the multiple importance sampling approach. The hero wavelength is randomly sampled first, and then the additional wavelengths are placed at equal distances from the hero wavelength to evenly cover the visible light range. On the other hand, Meng et al. [9] proposed a kind of up-sampling methods for creating spectra that match the input color precisely

## 2.3. MONTE CARLO RAYTRACING

Monte Carlo raytracing can generate photorealistic images. Kajiya introduced a Monte Carlo method (MC) to solve the rendering equation [3]. The method called path tracing (PT) stochastically samples the direction of rays and achieves global illumination in a raytracing approach.

Monte Carlo raytracing often uses importance sampling to accelerate the rendering process. Importance sampling is a technique for quickly reducing the variance of an estimate of a Monte Carlo integration. Introducing important sampling to Monte Carlo raytracing, the direction of reflected rays is determined stochastically based on the BRDF of the surface where a ray hits. That is, more rays are sampled from the direction where the BRDF has a larger value.

In the proposed method, we employ MC and important sampling to stochastically sample the wavelength of light across a wide range of visible light taking into account human visible perception.

## 3. PROPOSED METHOD

A tristimulus value rendering method proposed in this paper calculates radiances under several sets of wavelength samples determined stochastically by using color matching functions. The proposed method is able to generate physically correct results with less computational time and memory than traditional spectral rendering methods.

When spectral radiance of emitted light and spectral reflectance of surfaces are given, the procedure of the proposed method is as follows:

1. Determine several sets of wavelength samples of a photon taking into account human visual perception. We use XYZ color matching functions as probability density functions to stochastically sample the wavelength of visible light.
2. Calculate radiances using a progressive photon mapping (PPM) [10].
3. Calculate XYZ tristimulus values based on the idea of combination of a Monte Carlo integration and an importance sampling.
4. Convert XYZ tristimulus values into RGB values to display the image.

We could use the spectral distribution of light sources and/or the spectral reflectance of objects to sample the wavelength using a multiple importance sampling. In this case, a rendering method has a scene dependence property and we have to adjust rendering parameters each time we render different scenes to get a good performance. On the other hand, our method only depends on human visual perception. This means the proposed method has an advantage of a scene independent property.

### 3.1. DETERMINING WAVELENGTH SAMPLES

We sample sets of wavelengths from the wavelength of visible light using XYZ color matching functions.

First, we construct probability density functions (PDFs) by which the wavelength is stochastically sampled:

$$P_{\xi}(\lambda) = \frac{f_{\xi}(\lambda)}{\int_{\Lambda} f_{\xi}(\lambda) d\lambda}, \quad (6)$$

where  $P_{\xi}(\lambda)$  ( $\xi = \bar{x}, \bar{y}, \bar{z}$ ) are PDFs to sample wavelengths for X, Y, and Z stimulus values, respectively, and  $f_{\xi}(\lambda)$  ( $\xi = \bar{x}, \bar{y}, \bar{z}$ ) are XYZ color matching functions.

We use XYZ color matching functions in stead of RGB color matching functions, because the XYZ color space is wider than the RGB color space and the color matching functions have no negative values.

Next, the PDFs are converted into cumulative distribution functions (CDFs)  $C_{\xi}(\lambda)$  to determine a set of wavelength samples:

$$C_{\xi}(\lambda) = \int_{\lambda_{min}}^{\lambda} P_{\xi}(\lambda) d\lambda, \quad (7)$$

where  $\lambda_{min}$  is the shortest wavelength of visible light.

Finally, we determine a set of wavelengths,  $\lambda_\xi$  ( $\xi = X, Y, Z$ ), of a photon for calculating radiance corresponding to X, Y, and Z stimulus values by using an inverse function method:

$$\lambda_\xi = C_\xi(u)^{-1}, \quad (8)$$

where  $u$  is a uniform random number in  $[0, 1]$ .

### 3.2. CALCULATING RADIANCES

We employ a PPM to calculate radiance. We use  $N$  sets of wavelengths,  $\Lambda\{(\lambda_{1X}, \lambda_{1Y}, \lambda_{1Z}), (\lambda_{2X}, \lambda_{2Y}, \lambda_{2Z}), \dots, (\lambda_{NX}, \lambda_{NY}, \lambda_{NZ})\}$  determined in the previous step to calculate the radiance. That is, we assign  $N$  sets of wavelengths to each photon, and solve the rendering equation (Eq. 1) in respect to the sampled wavelengths using a PPM. The wavelength is sampled using the technique described in Section 3.1 for each photon. After a photon trace path, we get  $N$  sets of radiance values at each photon map and each pixel of an image. These radiance values are instantly converted into tristimulus values by using a straightforward method described in the following section, and just three stimulus values are stored in photon maps and pixels. A PPM repeats a photon trace path updating the photon maps. We change wavelength samples for each photon trace path. That is, the first and the second steps described in Section 3 are repeated in radiance calculation.

### 3.3. CALCULATING XYZ TRISTIMULUS VALUES

As described in section 2.1, when we calculate XYZ tristimulus values from a spectral distribution of radiance, we integrate the spectral radiance multiplied by each XYZ color matching function.

On the other hand, the proposed method calculates the radiances in respect to stochastically sampled wavelengths based on XYZ color matching functions. Therefore, tristimulus values are calculated by simply averaging the radiances of each photon trace path. That is, an expectation value of radiances is a stimulus value itself:

$$X(\mathbf{p}) = \frac{1}{MN} \sum_{i=1}^M \sum_{\nu=1}^N L_{\nu X}^{(i)}(\mathbf{p}), \quad (9)$$

$$Y(\mathbf{p}) = \frac{1}{MN} \sum_{i=1}^M \sum_{\nu=1}^N L_{\nu Y}^{(i)}(\mathbf{p}), \quad (10)$$

$$Z(\mathbf{p}) = \frac{1}{MN} \sum_{i=1}^M \sum_{\nu=1}^N L_{\nu Z}^{(i)}(\mathbf{p}), \quad (11)$$

where  $M$  is the number of photon trace paths,  $L_{\nu\xi}^{(i)}(\mathbf{p})$  ( $\nu = 1, \dots, N$ ,  $\xi = X, Y, Z$ ) are radiance values calculated in photon trace path  $i$ . Thus, the proposed is able to convert the spectral information instantly to tristimulus values and method needs only three bins to store the radiance values in photon maps as well as pixels. This is an advantage of our tristimulus value rendering method comparing to spectral rendering methods, because the traditional methods

use the number of bins as many as the sampled wavelength samples.

### 3.4. CONVERTING INTO RGB VALUES

XYZ tristimulus values are converted into RGB values by using Equation 5 to display on a RGB display device.

If we want, we can use a tone mapping, such as a Photographic Tone Mapping [11], to compress the dynamic range of intensity. The proposed method directly calculates XYZ tristimulus values. This is also one of the advantage of the proposed method, because the intensity in XYZ color space has more useful information for a tone mapping than those of RGB color space.

## 4. RESULTS

We examined the proposed method comparing to RGB rendering and Spectral rendering:

**RGB rendering (RGB):** Both spectral radiance of emitted light and spectral reflectance of surfaces are converted into RGB values by using RGB color matching functions in advance of rendering (see Fig. 1).

**Spectral rendering (Spectral):** Spectral intensities are calculated directly from spectral radiance of emitted light and spectral reflectance of surfaces. Wavelength of light is uniformly sampled every 1 [nm] across the range of visible light (380 - 780 [nm]). That is, Spectral rendering has uniform 400 wavelength samples.

The proposed method samples  $N$  sets of wavelengths from those discretized every 1 [nm] across the range of visible light.

We measure a root mean square error (RMSE) in XYZ color space:

$$\epsilon = \sqrt{\frac{1}{3N_p} \sum_{i=1}^{N_p} [(X(i) - X_{Ref}(i))^2 + (Y(i) - Y_{Ref}(i))^2 + (Z(i) - Z_{Ref}(i))^2]}, \quad (12)$$

where  $N_p$  is the total number of pixels in an image,  $X(i)$ ,  $Y(i)$ , and  $Z(i)$  are X, Y, and Z intensities of pixel  $i$ , respectively,  $X_{Ref}(i)$ ,  $Y_{Ref}(i)$ , and  $Z_{Ref}(i)$  are X, Y, and Z intensities of a ground truth. The ground truth values were calculated by using the Spectral rendering with uniform 400 wavelength samples described above with a sufficient computation time (24 hours).

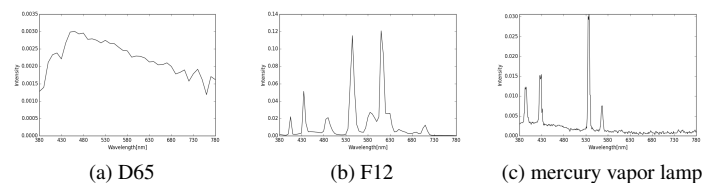
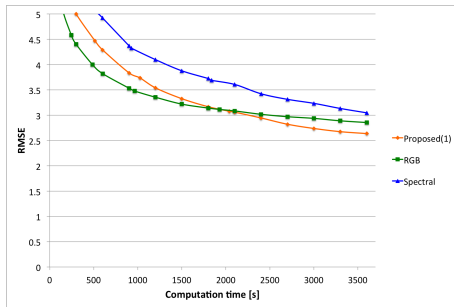


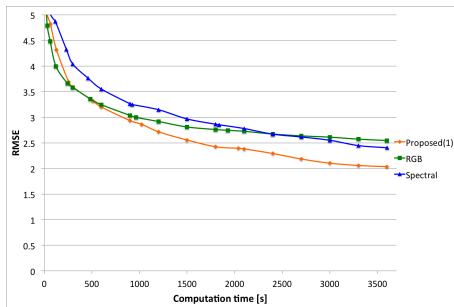
Figure 2: Spectral distribution of light sources



Figure 3: Scene with dispersion effects



(a) RMSEs in the whole image



(b) RMSEs in the region of a glass

Figure 4: RMSEs of each method in Fig. 3

In RGB rendering, the resultant output are R, G, and B values. To calculate a RMSE, we inversely converted RGB values into XYZ values (refer to Eq. 5). We also converted spectral intensities calculated by Spectral rendering into XYZ values using Eqs. 2 - 4.

All the experiments were done on a computer with Mac OS X 10.11.1, 3.5 GHz Intel Xeon E5 (6-core, 12-threads), 64 GB memory without GPU acceleration.

#### 4.1. SCENE WITH DISPERSION EFFECTS

Figure 3 shows the first test scene with a polyhedral glass under a spot light (a CIE standard illuminant D65 light source (see Fig. 2a)). Emitted light is refracted by the glass and iridescent patterns appear on the floor.

Figure 4a shows the transitions of RMSEs of the whole region of an image as computation time goes on. The proposed method used a single set of wavelengths ( $N = 1$ ).

The proposed method achieved the best performance among the other methods, that is, RMSEs of the proposed

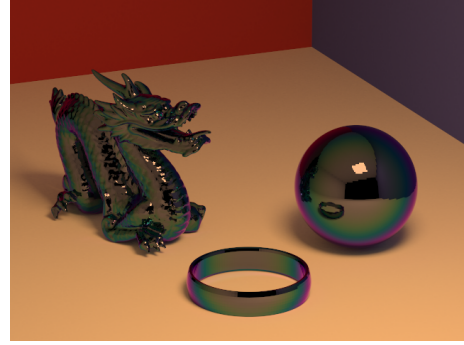
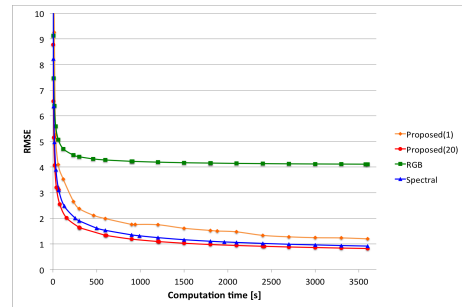
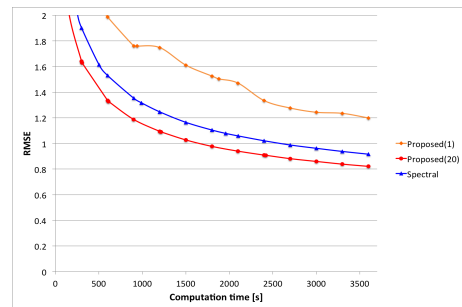


Figure 5: Scene with interference effects



(a) RMSEs in the whole image



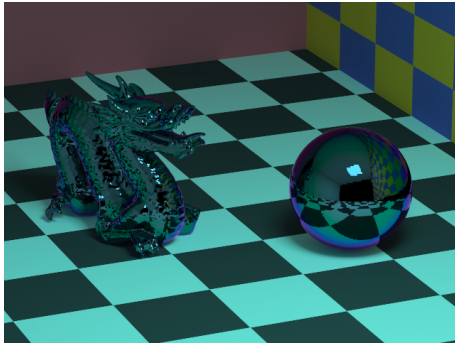
(b) Closeup view of RMSEs

Figure 6: RMSEs of each method in Fig. 5

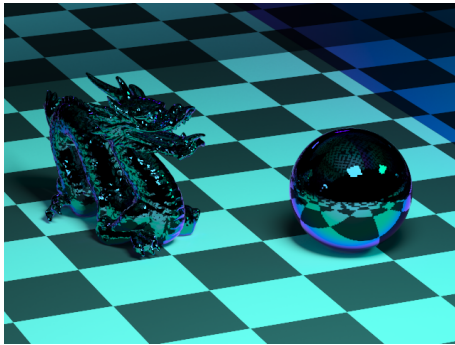
method is smallest after passing the computation time over 2,000 seconds. In another expression, the proposed method generates a high quality image within a certain RMSE in the shortest computation time than the other methods.

RGB rendering quickly decreases RMSEs in an early stage of computation, but the method cannot reduce RMSEs even though it spend a lot of time. Spectral rendering could give a high quality image, but the RMSE of Spectral rendering decreases very slowly.

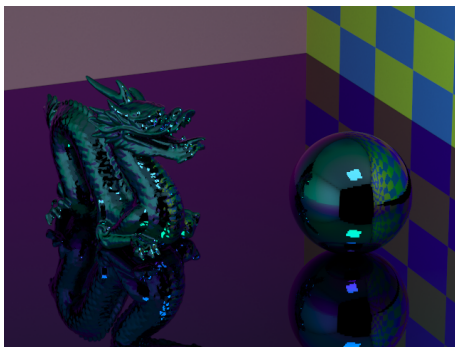
We focused on a glass regions shown in red rectangles in Fig. 3. The transitions of RMSEs of region is shown in Figs. 4b. The transitions of RMSEs are almost the same tendency as those of the whole image. A careful observation gives us the proposed method has an advantage in the glass region (see Fig. 4b), where light dispersion occurs in many spots on the glass (see Fig. 3).



(a) Two objects with interference of light



(b) Two objects and walls with interference of light



(c) Two objects and floor with interference of light

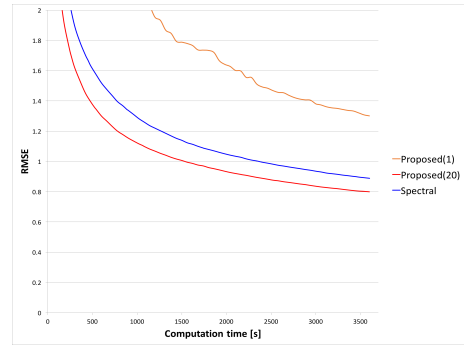
Figure 7: Scene with different interference areas

## 4.2. SCENE WITH INTERFERENCE EFFECTS

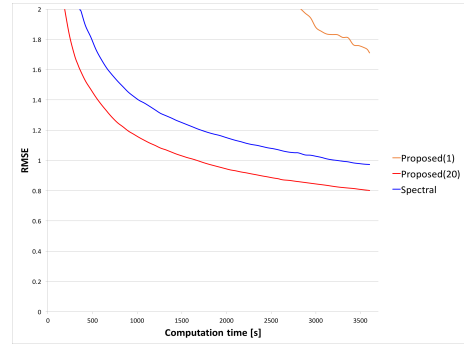
The second test scene has optical effects of interference of light (see Fig. 5). Iridescent colors appear on surfaces coated with a thin film ( $\text{SiO}_2$ ). The spectral distribution of a light source was set to a CIE standard illuminant F12 (see Fig. 2b).

The transitions of RMSEs are shown in Fig. 6a. The proposed method used 20 sets of wavelengths ( $N = 20$ ) as well as a single set of wavelengths ( $N = 1$ ). To closely observe the transitions, the RMSE axis is scaled in Fig. 6b.

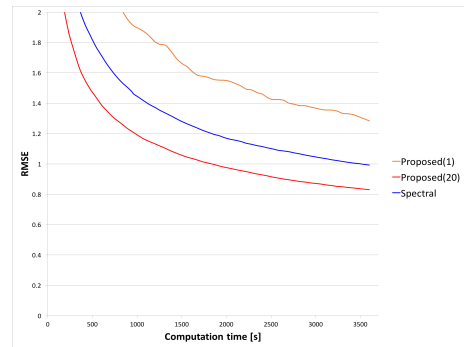
When we used 20 sets of wavelengths (Proposed(20)), the proposed method achieved the best performance.



(a)



(b)



(c)

Figure 8: RMSEs of each method in Fig. 7

## 4.3. SCENE WITH DIFFERENT INTERFERENCE AREAS

We tested the proposed method to render three types of scenes with different areas of interference of light under the illumination of a mercury vapor lamp, whose spectrum consists of bright lines (see Fig. 2c). Figure 7 shows the rendered scenes by the proposed method. The computation time in each image was 3,600 seconds. The transition of RMSEs of the proposed method and the spectral rendering are shown in Fig. 8. The proposed method with 20 sets of wavelengths achieved the best performance under any conditions.

## 5. CONCLUSIONS

We proposed a novel method for stochastically sampling the wavelength of light based on XYZ color matching functions. The proposed method uses a Monte Carlo integration and an importance sampling to sample the wavelength of light and to calculate tristimulus values instantly. The proposed method generates a high quality image with less computational time and memory than traditional spectral rendering methods when rendering optical phenomena such as dispersion, interference and diffraction of light.

We need further research on how many sets of wavelengths are required to draw the best performance from the proposed method. To investigate the usefulness of the proposed method thoroughly, we should examine the proposed method in various kinds of scenes. In our experiment, we used a PPM renderer to test the proposed method. We are interested in the performance when we use different renderers. The proposed method is simple and could be easily implemented to other renderers.

## REFERENCES

- [1] S. Watanabe, S. Kanamori, S. Ikeda, B. Raytchev, T. Tamaki, and K. Kaneda. Performance improvement of physically based spectral rendering using stochastic sampling. In *CCIW 2013*, volume 7786 of *Lecture Notes in Computer Science*, pages 184–198, 2013.
- [2] N. Isaivathi, S. Ikeda, B. Raychev, T. Tamaki, and K. Kaneda. Research on light light spectrum and color representation in computer graphics rendering. In *The Institute of Image Electronics Engineers of Japan The 269th Workshop*, 2014.
- [3] James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, 1986.
- [4] Gary W. Meyer. Wavelength selection for synthetic image generation. *Computer Vision, Graphics, and Image Processing*, 41(1):57 – 79, 1988.
- [5] S. Ikeda, T. Tamaki, B. Raytchev, and K. Kaneda. A study on sampling the wavelength of light in spectral rendering. *Information Processing Society of Japan*, 2015-CG-158(3):1–6, 2015.
- [6] Y. Takemura, S. Ikeda, K. Kaneda, T. Tamaki, and B. Raytchev. Determination of the number of wavelength samples for spectral rendering. *Visual Computing/Graphics and CAD Joint Symposium*, pages 24:1–24:4, 2015.
- [7] Michal Radziszewski, Krzysztof Boryczko, and Witold Alda. An improved technique for full spectral rendering. *Journal of WSCG*, 17(1-3):9–16, 2009.
- [8] Alexander Wilkie, S. Nawaz, Marc Droske, Andrea Weidlich, and Johannes Hanika. Hero wavelength spectral sampling. *Comput. Graph. Forum*, 33(4):123–131, 2014.
- [9] Johannes Meng, Florian Simon, Johannes Hanika, and Carsten Dachsbacher. Physically meaningful rendering using tristimulus colours. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)*, 34(4):31–40, June 2015.
- [10] T. Hachisuka, S. Ogaki, and H. W. Jensen. Progressive photon mapping. *ACM Transaction on Graphics.*, 27(5):130:1–130:8, 2008.
- [11] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., 2005.