https://doi.org/10.2352/EI.2024.36.10.IPAS-251

Spectral Rendering: from Input to Rendering Process

Ayoung Kim, Atanas Gotchev; Faculty of Information Technology and Communication Sciences, Tampere Univeirsity; Tampere, Finland

Abstract

Spectral rendering encompasses methods aimed at generating synthetic imagery with realistic color expression by simulating the interaction of light with different materials and wavelengths. Unlike conventional rendering methods that use tristimulus RGB values as input, spectral rendering uses the full spectrum of light as input, which allows for more accurate color reproduction and more realistic visual effects. Spectral rendering poses many new challenges in terms of input data acquisition, memory use, and efficiency. In this paper, we provide a comprehensive survey of the recent advances and open problems in spectral rendering, covering the main stages of the rendering pipeline such as spectrum sampling and reconstruction, RGB to spectrum upsampling, and selection of paths and wavelengths at rendering. We also discuss the applications that benefit from spectral rendering. Our goal is to provide an elaborate overview of the current state-of-the-art and to identify future research directions and opportunities in this exciting field.

Introduction

The recent advances in display technologies and the related image processing methods, such as super-resolution, high dynamic range, and wide color gamut, have steered the demand of consumers toward even better visual quality in terms of higher resolution, faster computation, deeper scene depth, and more realistic color expression. This paper addresses the challenge of realistic color expression, which requires the displayed colors to match the colors perceived in the real world. This requires not only reproducing the surface color of an object, but also considering environmental factors such as light and material properties. To achieve this, spectral rendering uses spectral data as the input for the rendering process instead of conventional RGB tristimulus values.

Recent research on spectral rendering has focused on specific areas, such as spectral sampling, spectral upsampling, spectral rendering techniques, or particular applications, including metamerism, subsurface scattering, color extinction, or wavelength-dependent effects [26, 27]. In this paper, we aim to address spectral rendering from the point of view of the entire pipeline, from input to rendering to display. We review each stage of the pipeline as addressed in the recent research with the goal of formulating unsolved research questions and identifying future research directions and their applicability potential.

Spectral Rendering

Spectral rendering determines the color of the display pixel corresponding to the scene utilizing the spectra. Fig. 1 illustrates the difference between two inputs for color rendering: spectral reflectance and RGB values. The latter depends on color spaces, which are color models covering a diverse range of colors, as marked by triangles on the left side of Fig. 1, while the former utilizes the constant spectrum for a particular color to maintain the brightness of the color under different illumination and provide color fidelity to several light bounces.

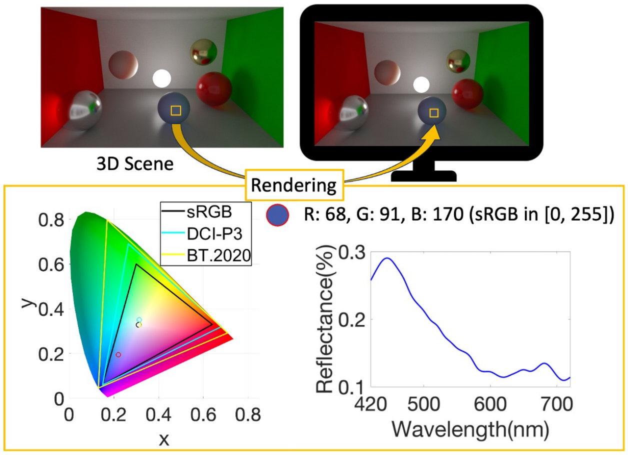


Figure 1: RGB tristimulus values versus spectral reflectance. The triangles on the chromaticity diagram show different color gamuts, and the circles of the same color represent the white points in the corresponding color space. The red circle shows the location of the example color.

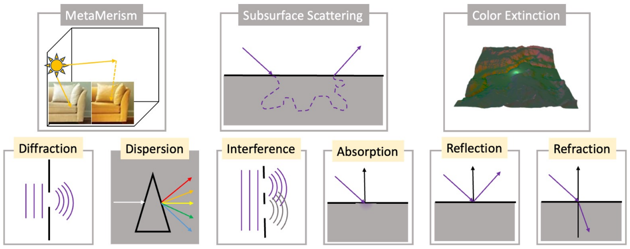


Figure 2: Benefits of spectral rendering for representing light and material properties, and wavelength-dependent effects.

Spectral rendering can efficiently express light, material, and pattern properties (Fig. 2, upper row). *Metamerism* is a phenomenon in which the colors of the same object appear differently under direct and indirect illumination. The color bleeding from the wall surface also affects metamerism. *Subsurface scattering* is a material-related property commonly found in the skin and clothes texture. *Color extinction* refers to the color becoming sparse by participating media, such as fog. *Wavelength-dependent effects* (Fig. 2, lower row) cannot be achieved by RGB-based rendering alone while physical-based rendering can be performed naturally by tracking the wavelengths per path, thus including all wavelength-dependent effects.

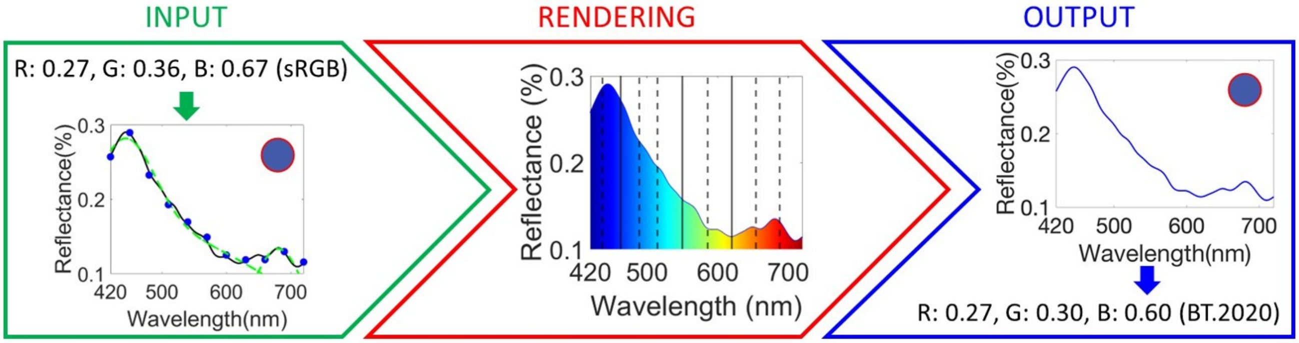


Figure 3: Schematic diagram of the spectral rendering pipeline.

Fig. 3 presents the three stages of the entire spectral rendering process. The input stage is about obtaining spectral data from the real-world scene and efficiently representing it. Optimal wavelengths and corresponding ray paths are determined and rendered during the rendering stage. Finally, the output stage converts the output spectrum into the RGB components in the display color space. The following sections explain the challenges and solutions for each stage in detail.

Input

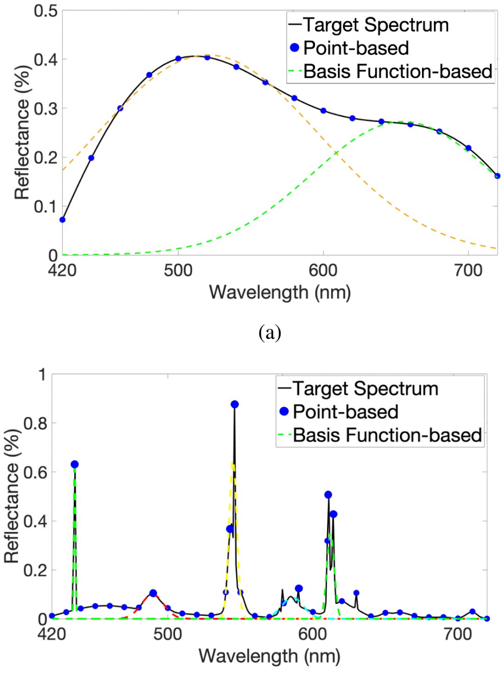
To render with spectral inputs, one needs to consider several aspects: how to acquire the spectral data from the real-world scene, how to represent (sample) the continuous spectrum by discrete models for efficient data manipulation and storage, and how to convert (upsample) RGB data in a specific color space to spectral data if the scene spectrum is not available. The latter is essential also for providing compatibility between spectral rendering pipelines since most rendering tools and pipelines are designed for RGB data.

One can distinguish spectra of natural colors that are smooth in general and spectra of synthetic colors that are spiky due to an abnormal spectrum shifting via fluorescent materials [16]. Spectral data can be acquired directly from the real-world scene via multispectral or hyperspectral cameras. The resulting spectral images are represented in three dimensions, spanning the 1D spectral domain and the 2D spatial domain. Computational snapshot hyperspectral cameras can obtain an entire spectral rectangular cuboid within a single detector integration period via amplitude or phase modulation [1, 2, 7, 9, 10, 24, 30]. However, the critical drawbacks of those spectral cameras are their limited resolution and incompatibility with different camera lenses. Addressing these limitations is important for their practical use in production pipelines.

Spectral Sampling

Spectral sampling refers to the process of representing the continuous spectrum by a discrete model for subsequent storage and processing (c.f. Fig. 1 lower right). A continuous spectrum is directly available when using synthetic scenes. For the case of real-world spectral data, the discrete model must be fit by using the available discretely-sensed data. In both cases, the aim of the discrete model is to ensure the (perfect) reconstruction of any desired spectral component, a process which is referred to as *spectral reconstruction*.

An available continuous spectrum can be simply sampled densely enough. In literature, this has been referred to as *point-based sampling*. Fig. 4 illustrates it by the blue points, which are used to represent the target spectrum on the black solid line. The points can be set by Gaussian quadrature or Riemann summation [18]. Gaussian quadrature effectively integrates polynomials over general weighting functions [5]. Riemann summation is the weighted sum of equally spaced samples, where the weight is the distance between the sample wavelengths [19]. The question is how dense the point grid should be. An insufficient number of sampling points results in color noise, i.e., chromatic aberration. It is a general method applicable to any spectra shape, whether smooth or spiky. However, it is not the best choice for spiky spectra in terms of processing and storage efficiency. Specifically, if the blue points at the peak area of Fig 4b are missing, sampled points cannot represent the spectrum shape accurately,



(a)

(b)

Figure 4: Spectral sampling approaches: Point-based and Basis function-based. (a) a smooth spectrum case for natural colors. (b) a spiky spectrum case mainly for fluorescent colors.

which leads to color errors.

A more elegant signal processing approach is to decompose the spectrum over a set of basis functions, with the aim to match the smoothness of the underlying natural-color continuous function by a *weighted sum of a few basis functions* [4, 21]. Widely used basis functions are the box function, harmonic basis functions, or Gaussians, and the targeted discrete model is provided by the weighting coefficients in the sum. In addition, characteristic vector analysis and principal component analysis (PCA) have been used to select the best set of basis functions in a purely discrete setting [18]. The advantage of basis function-based representation is that the spectrum is represented with fewer parameters compared to the point-based representation. An open research question is how to represent both smooth and spiky spectra simultaneously on a suitable basis. It is worth noting that the amount of computation rises substantially when more basis functions are utilized to represent the shape.

Upsampling

Upsampling can be considered as a particular case of spectral reconstruction from a limited set of spectral components, most often from RGB values. It is an ill-posed problem, particularly due to the metamerism that results in an infinite number of spectra for the same RGB combination. Upsampling assumes a trade-off between performance, speed, and storage while aiming at achieving a unique spectrum per chromaticity and taking into account the range ([0, 1]) and the smoothness of the spectral reflectance shape. There are two general approaches for upsampling: basis function-based and diagram-based.

The concept of discrete (parameterized) models of spectra based on basis decomposition is general enough and can be assumed also for reconstructing spectra from tristimulus values. Such models facilitate simple and practical solutions with fast computation. Frequently, they represent natural colors in the visible range as a smooth spectrum formed by a weighted sum of the primary color spectra (S_R, S_G, S_B)

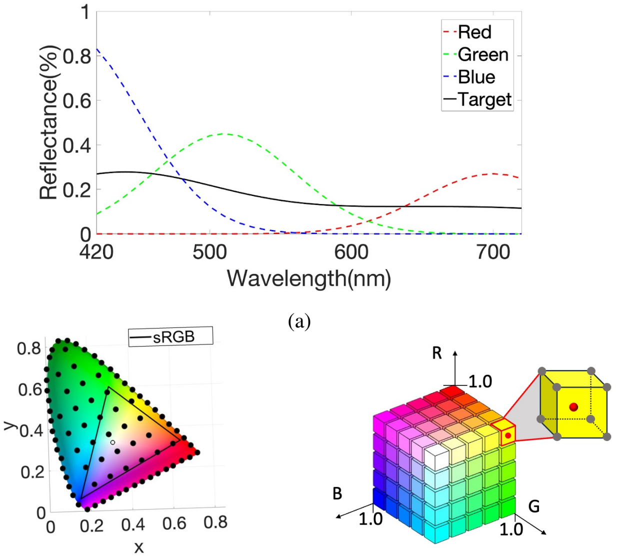
$$S(\lambda) = \omega_{R_\lambda} S_R + \omega_{G_\lambda} S_G + \omega_{B_\lambda} S_B, \tag{1}$$

where $S(\lambda)$ is the target reflectance and ω is the weight per primary spectrum [3, 13, 14, 17, 20]. Fig. 5a illustrates how the target spectrum (the black solid line) is formed by a weighted sum of the RGB (the dashed lines). The primary spectra refer to the RGB or XYZ spectra, which can be derived through different basis functions, including CIE XYZ spectrum matching functions ($\bar{x}$, $\bar{y}$, $\bar{z}$) in the following integral equation:

$$\begin{aligned} X &= \int_\lambda S(\lambda) I(\lambda) \bar{x}(\lambda) d\lambda, \\ Y &= \int_\lambda S(\lambda) I(\lambda) \bar{y}(\lambda) d\lambda, \\ Z &= \int_\lambda S(\lambda) I(\lambda) \bar{z}(\lambda) d\lambda, \end{aligned} \tag{2}$$

where $S(\lambda)$ is the unknown spectral reflectance and $I(\lambda)$ is an illuminant spectral power distribution (SPD). The CIE XYZ matching functions are well-known spectral sensitivity curves providing numerical descriptions of the CIE standard observer's color response. The XYZ tristimulus values are defined in the uniform color space and can be converted from RGB values in various color spaces via a transformation matrix.

The target spectrum reconstruction accuracy in the basis function-based approaches clearly depends on the spectrum representing functions. To simplify the decomposition, the box function has been considered instead of the basic CIE XYZ spectrum matching functions, to maximize the brightness at a specific color saturation. According to MacAdam [13], a single box function is insufficient to express the smoothness of spectral reflectances for natural colors that results in unwanted color shifts under indirect illumination. Smits [20] has weighted the spectral difference between RGB primary and CMYK secondary colors when combining the spectra of white and RGB colors to ensure smoothness in the target spectrum. The primary and secondary spectra are derived from ten precomputed spectra using box functions. Although it has achieved accurate smooth spectra in sRGB color space, round-trip errors increase at wider color gamuts and with multiple inter-reflections.



(a)

(b)

(c)

Figure 5: Upsampling Methods: (a) a basis function-based approach, (b) a 2D chromaticity diagram-based approach, and (c) a 3D cube-based approach. Each black and gray point value is a measured spectrum or a set of coefficients (e.g., 2nd-order polynomials, Fourier series). .

Using a box function as the basis for the spectrum representation could not achieve maximum brightness when color gamuts got wider than sRGB, so there has been a trial of adopting different basis functions. For example, Otsu et al. [17] have tried to optimize the spectrum representation by searching for a set of basis functions using PCA, a data-driven learning method based on measured spectra [11]. Here, PCA can produce authentic smoothness since it selects a set of three basis functions composing the spectrum through the real spectra. They accelerate the computation speed by pre-defining the three basis functions for each cluster that classify the measured reflectances according to XYZ values. Clustering had benefits for computation, but it caused an issue with the adjacent colors located at the boundaries of different clusters. The round-trip errors occur because their spectra look quite different due to different pre-defined basis functions per cluster. Moreover, the critical drawback is out-of-range values in the output spectrum, such as negative regions.

Besides searching for the basis functions for spectrum representation, Mallett and Yuksel [14] and Burns [3] have concentrated more on finding the weighting factor in the spectrum representation. They get primary spectra using CIE XYZ spectrum matching functions. Mallett and Yuksel [14] have used RGB tristimulus values as weights and conducted spectral primary decomposition, which searches each primary spectrum separately. Assuming $S_{red}(\lambda) + S_{blue}(\lambda) + S_{green}(\lambda) = 1$, the spectrum $S(\lambda)$ can fulfill the energy-conserving condition. They have used sRGB tristimulus values under D65 illumination as an input, thus getting robust results under different illuminants. However, the method has limited performance at wide gamuts because it cannot express highly saturated colors. Burns [3] has applied a hyperbolic tangent-based function $(tanh(x)+1)/2$ to the estimated target spectrum via CIE XYZ spectrum matching functions for the [0, 1] spectral range condition. Optimization has been done by minimizing the square of the reflectance slope to find a smooth spectrum shape. The resulting colors are pleasant within the object color solid, while the boundary values exhibit non-smooth square-shaped spectra.

The basis function-based approach does not work well for colors that are not formed by combining primary colors. For example, subtractive colors, such as yellow, magenta, and cyan, are challenging because any overlap between the primary spectra causes undesired nonlinear absorption during the linear color combination. Specifically, the purple spectrum demonstrates concentrated energy at both long and short wavelengths, with little power at medium wavelengths. On the other hand, the yellow spectrum contains short waves throughout the visible wavelength range.

Diagram-based methods structure the RGB-to-spectrum mapping process using a 2D chromaticity diagram and a 3D cube-shaped lookup table. A diagram-based approach can address more complex representative coefficients for each spectrum of given RGB values in intuitive and memory-efficient lookup tables. Each point in a diagram has a measured spectrum or a set of coefficients (e.g., 2nd-order polynomials, Fourier series) for the corresponding coordinates. Meng et al. [15] have interpolated the color gamut area as a grid system in Fig. 5b with an equal brightness assumption (X+Y+Z = 1). The target spectrum is estimated by weighting the adjacent points. The grid interval is reduced in half at the color gamut boundary to alleviate artifacts due to interpolation. Since it is critical to accurately and quickly choose which neighboring points to use, they have divided the cases into two categories: inner and boundary points. Subsequently, bilinear interpolation has been employed for inner points and triangular interpolation for boundary points. Each grid point has a pre-measured spectrum, which guarantees the smoothness of the estimated spectrum. Since the suggested grid covers the whole chromaticity diagram, it performs well even in wide color spaces. However, the performance is strongly influenced by the point density of the grid, which is directly related to the memory requirement. Also, in this system, they did not consider the spectrum range constraints, and re-scaling in the end to fit in the [0, 1] range caused the color error.

3D cube-based lookup tables illustrated in Fig. 5c expand one more dimension from the 2D table for more degrees of freedom, e.g., 3D coordinate system instead of 2D chromaticity coordinates, more adjacent points, or brightness factors [8, 22, 23, 25]. Wang et al. [25] increase the table dimension for the XYZ coordinate system and more adjacent points when constructing the output spectrum. The voxel includes the spectral reflectances from input XYZ based on measured 1400 reflectances and generated reflectances for saturated colors. The saturated color spectra generated via Bouguer's law in Eq. 3 primarily aim at achieving the wide gamut of color spaces

$$S_{sat}(\lambda) = 10^{-S_{meas}(\lambda)l}, \tag{3}$$

where S_{sat} is the spectral reflectance of a saturated color, and S_{meas} is the measured reflectance. When the light path length l is increased for the same wavelength λ, the color becomes more saturated. A target spectrum at the red point in Fig. 5c is derived by linear interpolation of the eight closest colors among measured and generated reflectances marked with gray voxels at the yellow cube in Fig. 5c. The eight XYZ values of the closest colors form the cube, which surrounds the target XYZ value to satisfy the reflectance value condition. The target spectrum is calculated on the same-shaped cube as the XYZ cube. It can achieve smoothness of spectrum like other measured data-based methods. Explicitly considering the saturated color spectra differentiates from the methods so far, but the reliability of the generated spectra needs to be examined carefully.

For a more memory-efficient 3D lookup table, the spectrum is parameterized by spectral coefficients that support fast transformation into the spectrum [8, 22]. For example, the second-order polynomial coefficients (c_i) of Jakob and Hanika [8] in Eq. 4 significantly save the memory, where runtime memory usage is comparable to saving three RGB floats. The coefficients are calculated by the CERES solver, and easily transformed into the spectrum through the sigmoid function in Eq. 5.

$$S(\lambda) = Sigmoid(c_o\lambda^2 + c_1\lambda + c_2), \tag{4}$$

where

$$Sigmoid(x) = \frac{1}{2} + \frac{x}{2\sqrt{1+x^2}}. \tag{5}$$

The sigmoid function simplifies the evaluation and reflects the energy conservation requirement of the nonlinear mapping. On the other hand, Tódová et al. [22, 23] have defined the spectrum using Fourier coefficients, i.e., a combination of trigonometric moments and the 0th moment, for the moment representations.

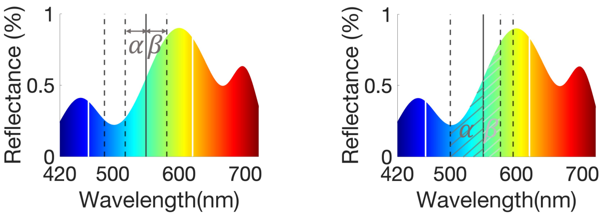
Furthermore, the 3D cube coordinates of Jakob and Hanika [8] are more logically improved than basic RGB coordinates. They have designed the 3D cube by extending the three quadrilateral areas of the optimized polynomials via a brightness factor for more precise smoothness in the brightest colors. Specifically, connecting red, green, and blue points to white points and each complementary color in the sRGB color gamut constructs the polynomial rectangular area. The proposed cube structure demonstrates remarkable performance over smoothness and a wide gamut. Thus, the follow-up work by Tódová et al. [22, 23] expands the 3D cube of Jakob and Hanika with a customization option: adding extra voxel values based on user-selected moment representations with unique IDs at specific RGB colors. The wide gamut beyond sRGB is addressed by node clipping, which matches wide gamut colors to the closest boundary sRGB value along the axis connected to the center of the destination gamut while preserving high saturation. The customization option at the upsampling lookup table is interesting and has potential, especially in the artistic view, but there is lots of room for improvement in data structure and computation speed.

Renderng Process

Spectral rendering determines the spectral contributions to a pixel by computing multiple reflection paths from each light source to the viewer, with spectral data as sensed, modeled, and represented in the input stage. The main challenge in the spectral rendering process is how to select the ray paths and wavelengths to trace for optimal performance. We compare single-wavelength and multiple-wavelength tracing methods and analyze various techniques for multiple-wavelength tracing.

The first attempt at spectral rendering was using a single wavelength per path by adding a wavelength dimension as one dimension of the Monte Carlo path integration. It achieves physically-based rendering by tracing the wavelength but causes significant color noise relative to RGB rendering due to the loss of color information. That is, RGB has three color components, while a single wavelength represents one color component, so only one-third of the color information can reach the final pixel.

To transmit more color information, Evans and McCool [6] have produced a cumulative normalized SPD of stratified random wavelengths to track multiple wavelengths per path. That is, the average SPD of multiple random wavelengths inside a wavelength cluster assigned to a specific optical path is transferred to the pixel sensor. The wavelengths are importance-sampled using normalized illuminant SPDs. Multiple wavelengths within the same cluster may follow a single path through importance sampling of reflection directions. While Monte Carlo rendering decides the path from a light source to the pixel, multiple spectral rendering decides the path depending on the wavelength.



(a) Uniform Wavelength Sampling

(b) Non-uniform Wavelength Sampling

Figure 6: Multiple wavelength spectral rendering techniques, where the distance or the area $\alpha = \beta$. The solid black line is the current hero wavelength that selects the supplementary wavelengths indicated by the dashed lines. The solid white lines are the other hero wavelengths.

Other multiple wavelength tracing methods have developed actively following Evans and McCool [6]. Among them, the most remarkable work is a *uniform wavelength sampling* method known as hero wavelength sampling [29]. Hero wavelength sampling effectively solves the color noise problem with trivial computation overhead. Hero wavelengths are determined randomly to steer the direction of a light path without spectral aliasing. With respect to a hero, i.e. central wavelength, constantly spaced supplementary wavelengths are selected to cover all wavelength ranges as shown in Fig. 6a. The jth supplementary wavelength λ_j at a given hero wavelength λ_h is mathematically described as

$$\lambda_j(\lambda_h) = (\lambda_h - \lambda_{min} + \frac{j}{C}(\lambda_{max} - \lambda_{min}))\mathbf{mod}(\lambda_{max} - \lambda_{min}) + \lambda_{min}, \tag{6}$$

where λ_{min} and λ_{max} are the minimum and maximum wavelength, respectively, and the total number of supplementary wavelengths is $C = 3$. Assuming every wavelength generates a particular path, supplementary wavelengths follow the given path generated by a hero wavelength via the combined sample density, which is the probability of sampling a set of wavelengths on a fixed path. The set of three uniformly-distributed supplementary wavelengths and a hero wavelength per path is integrated with around equal weighting driven from the balance heuristic. Equal weighting raises the advantages of using several wavelengths in the case of smooth spectra. On the other hand, it is not efficient if the spectrum has a strong wavelength dependency, where the spectral energy is concentrated on a particular region.

For efficient selection of wavelengths with high energy in hero wavelength sampling, *non-uniform wavelength sampling* provides more freedom with flexible hero wavelength sampling or supplementary wavelength sampling that is not bound by the hero wavelength. Since non-uniform wavelength sampling can select informative wavelengths where the energy is concentrated, as demonstrated in Fig. 6b, it can achieve equivalent performance as uniform wavelength sampling with tracing fewer wavelengths.

A task-specific non-uniform wavelength sampling method has been proposed by Kutz et al. [12] who have used weighted delta tracking to select hero wavelengths that prevent wavelength-dependent extinction in the participating media. The weights are derived from the collision probability at the maximum or average across all wavelengths. Eliminating null collisions allows efficient spectrum rendering in participating media with hundreds of scattering events in each direction. As a result, color noise and bright color outliers are decreased. A more general method has been proposed by West et al. [28] who have sampled the supplementary wavelengths using stochastic multiple importance sampling (SMIS), while hero wavelengths have been selected randomly as original hero wavelength sampling. In particular, the sampling probability density function of the supplementary wavelengths is proportional to the product of the observation response and the light source SPDs and creates a stratified probability pattern. SMIS demonstrates flexible importance sampling of wavelengths on the continuous spectrum domain and outperforms the original hero wavelength sampling in locating the peak energy wavelength with less color noise.

Output

The output color of the spectral rendering is in the form of a spectrum and needs to be converted to the RGB tristimulus values of the display color space. First, the standard XYZ matching functions in Eq. 2 transform the spectrum into XYZ values of the uniform color space. Second, the transformation matrix converts the XYZ values into the output RGB values in the display color space. The two-step process adequately handles the spectrum-to-RGB conversion task in the output stage.

Conclusions

To achieve realistic color expression considering environmental factors such as light and material properties, spectral rendering requires spectral data as the input for the rendering process instead of conventional RGB tristimulus values. This paper has reviewed the entire spectral rendering pipeline, from input to output. Finding spectral representations at the input stage is essential for the subsequent stages. Since the high peak reflectance area is informative for identifying the major wavelengths, the extended spectra representation can be applied at the rendering stage. The output stage corresponds to the inverse of upsampling to obtain the RGB values of the display color space from the output spectrum. When looking at the entire spectral rendering stages at once, discovering high-energy regions of the spectrum is a common issue, though there are various tasks in each stage. Consequently, joint optimization of various stages with common spectral representation information might be a future research direction. Furthermore, backpropagation can be applied to update the main wavelength selection when addressing spectral rendering from a holistic perspective.

References

[1] Henry Arguello, Samuel Pinilla, Yifan Peng, Hayato Ikoma, Jorge Bacca, and Gordon Wetzstein. Shift-variant color-coded diffractive spectral imaging system. *Optica*, 8(11):1424–1434, Nov 2021.

[2] Seung-Hwan Baek, Incheol Kim, Diego Gutierrez, and Min Kim. Compact single-shot hyperspectral imaging using a prism. *ACM Transactions on Graphics*, 36(6):1–12, Nov 20, 2017.

[3] Scott A. Burns. Numerical methods for smoothest reflectance reconstruction. *Color research and application*, 45:8–21, 2020.

[4] Jozef Cohen. Dependency of the spectral reflectance curves of the munsell color chips. *Psychonomic science*, 1:369–370, 1964.

[5] Philip J. Davis. *Methods of numerical integration*. Academic Press, 2. ed. edition, 1984.

[6] Glenn F. Evans and Micheal D. McCool. Stratified wavelength clusters for efficient spectral monte carlo rendering. In *Proceedings of the 1999 Conference on Graphics Interface '99*, page 42–49, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[7] M. E. Gehm, R. John, D. J. Brady, R. M. Willett, and T. J. Schulz. Single-shot compressive spectral imaging with a dual-disperser architecture. *Optics Express*, 15(21):14013–14027, Oct 17, 2007.

[8] Wenzel Jakob and Johannes Hanika. A low-dimensional function space for efficient spectral upsampling. *Computer Graphics Forum*, 38:147–155, 11 2019.

[9] Daniel Jeon, Seung-Hwan Baek, Shinyoung Yi, Qiang Fu, Xiong Dun, Wolfgang Heidrich, and Min Kim. Compact snapshot hyperspectral imaging with diffracted rotation. *ACM Transactions on Graphics*, 38(4):1–13, Jul 12, 2019.

[10] Ayoung Kim, Ugur Akpinar, Erdem Sahin, and Atanas Gotchev. Computational hyperspectral imaging with diffractive optics and deep residual network. In *2022 10th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6, 2022.

[11] Oili Kohonen, Jussi Parkkinen, and Timo Jääskeläinen. Databases for spectral color science. *Color research and application*, 31:381–390, 2006.

[12] Peter Kutz, Ralf Habel, Yining Karl Li, and Jan Novák. Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Trans. Graph.*, 36(4), jul 2017.

[13] David L. MacAdam. The theory of the maximum visual efficiency of colored materials. *Journal of the Optical Society of America (1930)*, 25:249, 1935.

[14] Ian Mallett and Cem Yuksel. Spectral Primary Decomposition for Rendering with sRGB Reflectance. In Tamy Boubekeur and Pradeep Sen, editors, *Eurographics Symposium on Rendering - DL-only and Industry Track*. The Eurographics Association, 2019.

[15] Johannes Meng, Florian Simon, Johannes Hanika, and Carsten Dachsbacher. Physically Meaningful Rendering using Tristimulus Colours. *Computer Graphics Forum*, 2015.

[16] M. Mojzík, A. Fichet, and A. Wilkie. Handling fluorescence in a unidirectional spectral path tracer. *Computer Graphics Forum*, 37:77–94, 07 2018.

[17] H. Otsu, M. Yamamoto, and T. Hachisuka. Reproducing spectral reflectances from tristimulus colours. *Computer graphics forum*, 37:370–381, 2018.

[18] Mark S Peercy. Linear color representations for full speed spectral rendering. pages 191–198. Association for Computing Machinery, 1993.

[19] Brent Smith, Charles Spiekermann, and Robert Sember. Numerical methods for colorimetric calculations: A comparison of integration methods. *Color research and application*, 17:384–393, 1992.

[20] Brian Smits. An rgb-to-spectrum conversion for reflectances. *Journal of graphics tools*, 4:11–22, 1999.

[21] W. S. Stiles, G. Wyszecki, and N. Ohta. Counting metameric object-color stimuli using frequency-limited spectral reflectance functions. *Journal of the Optical Society of America (1930)*, 67:779, 1977.

[22] Lucia Tódová, Alexander Wilkie, and Luca Fascione. Moment-based Constrained Spectral Uplifting. In Adrien Bousseau and Morgan McGuire, editors, *Eurographics Symposium on Rendering - DL-only Track*. The Eurographics Association, 2021.

[23] Lucia Tódová, Alexander Wilkie, and Luca Fascione. Wide gamut moment-based constrained spectral uplifting. *Computer Graphics Forum*, 41(6):258–272, 2022.

[24] Ashwin Wagadarikar, Renu John, Rebecca Willett, and David Brady. Single disperser design for coded aperture snapshot spectral imaging. *Applied Optics (2004)*, 47(10):B44–B51, Apr 01, 2008.

[25] Qiqi Wang, Haiying Xu, and Yinlong Sun. Practical construction of reflectances for spectral rendering. In *International Conference in Central Europe on Computer Graphics and Visualization*, 2004.

[26] Andrea Weidlich, Alex Forsythe, Scott Dyer, Thomas Mansencal, Johannes Hanika, Alexander Wilkie, Luke Emrose, and Anders Langlands. Spectral imaging in production: Course notes siggraph 2021. Association for Computing Machinery, Inc, 8 2021.

[27] Andrea Weidlich, Chloe Legendre, Carlos Aliaga, Christophe Hery, Jean Marie Aubry, Jirí Vorba, Daniele Siragusano, and Richard Kirk. Practical aspects of spectral data in digital content production. Association for Computing Machinery, Inc, 8 2022.

[28] Rex West, Iliyan Georgiev, Adrien Gruson, and Toshiya Hachisuka. Continuous multiple importance sampling. *ACM Transactions on Graphics*, 39, 7 2020.

[29] A. Wilkie, S. Nawaz, M. Droske, A. Weidlich, and J. Hanika. Hero wavelength spectral sampling. *Computer Graphics Forum*, 33:123–131, 2014.

[30] Nan Xu, Hao Xu, Shiqi Chen, Haiquan Hu, Zhihai Xu, Huajun Feng, Qi Li, Tingting Jiang, and Yueting Chen. Snapshot hyperspectral imaging based on equalization designed doe. *Opt. Express*, 31(12):20489–20504, Jun 2023.