Scientific Visualization, 2025, volume 17, number 4, pages 112 - 123, DOI: 10.26583/sv.17.4.11

This paper is an extended version of a contribution presented
at the Graphicon 2025 conference.

Visualization of Spectral Scenes Using Fourier Series

R. O. Rodionov[1,A,B], E. V. Prikhodko[2,B], V. A. Frolov[3,A,B], A. G. Voloboy[4,A]

[A] Keldysh Institute of Applied Mathematics RAS, Moscow, Russia
[B] IAI Moscow State University, Moscow, Russia

[1] ORCID: 0009-0000-5044-9963, roman.rodionov@graphics.cs.msu.ru
[2] ORCID: 0009-0007-8043-9302, egor.prikhodko@graphics.cs.msu.ru
[3] ORCID: 0000-0001-8829-9884, vladimir.frolov@graphics.cs.msu.ru
[4] ORCID: 0000-0003-1252-8294, voloboy@gin.keldysh.ru

Abstract

This paper presents spectral rendering method that addresses key challenges in storing and processing spectral data. The proposed approach represents light and material properties using truncated Fourier coefficients, allowing spectra to be stored and manipulated compactly. This representation reduces memory usage and computational overhead while preserving the accuracy of spectral information during rendering. The method enables efficient reconstruction of stored spectral functions and simplifies operations such as color conversion. Several strategies for transforming Fourier coefficients within a path tracing framework are investigated, including different spectrum-to-color conversion techniques, such as using zeroth Fourier coefficient to directly convert Fourier-based spectrum to color. Experimental results show that the proposed method provides rendering quality comparable to traditional approaches while producing lower color noise and similar computation times. The method is particularly effective for fast preview and interactive rendering, where low samples per pixel are used and color noise strongly affects visual perception. Also, the paper describes applications of proposed method in neural rendering for storing BRDF using compact neural networks. Furthermore, variance reduction approach based on Fourier coefficients is proposed.

Keywords: spectral rendering, Fourier series, neural material models.

1. Introduction

One of the most promising directions in physically based rendering is spectral rendering: instead of the conventional RGB color model, a real-valued wavelength function describing the spectral power distribution of light (or the reflectance of materials) is used in scenes. The spectrum is usually represented in render systems and optical modeling systems as a set of wavelength-value pairs. This method is used, for example, in the PBRT renderer [1]. Such a method can represent spectral values for materials, light sources, and textures. However, individual rays or photons carrying the spectrum typically store not the entire spectrum, but only a certain sample from it. Here, the typical number of wavelengths in such samples is generally small: 4, 8, 16, 32 [2]. Such a representation is suboptimal both in terms of memory and performance.

Unfortunately, even simply sampling a spectrum necessitates a binary search operation within an array. To avoid this computational cost, some rendering systems [1, 3] precompute material and light source spectra by mapping them into an array with a fixed step, for instance, at 1-nm intervals.

This typically results in significant performance improvement in rendering systems. However, both the binary search approach and especially the constant-wavelength sam-

pling method prove impractical for spectral textures in real-world applications. The conversion from 2D to 3D textures increases memory consumption by one to two orders of magnitude compared to conventional textures. What we need in practice is a storage solution for this 3D function data that maintains both compact size and efficient access capabilities.

Since most spectral functions used in practice are continuous, they can be represented using the Fourier transform. The resulting coefficients can subsequently be reconstructed into the original function using, for example, Fourier series.

Current research on spectral storage methods has already explored using the Fourier transform as a compact and efficient representation of spectral distributions for light sources and material reflectance. In practice, storing sufficiently smooth distributions with high accuracy requires only a small number of coefficients. However, none of the existing studies propose using the Fourier representation directly during the rendering process itself. Implementing this approach could potentially reduce memory usage (including RAM) during rendering and scene storage, decrease color noise by leveraging full spectral information instead of subsampling, and accelerate image generation times.

The aim of this paper is to investigate the application of spectral representation in Fourier coefficient space for path tracing-based rendering systems.

2. Survey of existing methods

The most common approach for representing spectra in spectral rendering involves storing discrete values at fixed wavelength intervals. Typically, the spectrum is represented as a table of the values, that are uniformly distributed across the spectral range. While certain physical material models, such as perfect dielectrics or interference coatings, may reflect or refract light at a single specific wavelength [4], most scenarios require transporting the entire light spectrum. In these cases, discrete representation becomes redundant and inefficient, motivating the search for more compact spectral storage methods.

The study [5] proposes using a small vector of polynomial coefficients to represent reflection spectra. The sum of such vectors would correspond to the sum of their spectra. However, this method is designed for generating synthetic spectra from known colors, and its applicability for representing arbitrary measured spectra remains unexplored.

Furthermore, spectral distribution functions can be represented using Fourier coefficients. Several studies have proposed this approach. The authors of [6] suggest using the MESE (Maximum Entropy Spectral Estimation) method and its modification - Bounded MESE - to reconstruct reflection spectra. Their research demonstrates that MESE can accurately represent continuous spectral functions using only 6 Fourier coefficients. This method effectively avoids artifacts and distortions inherent in direct reconstruction using truncated Fourier series.

A related method for storing spectral textures using this approach was presented in paper [7]. The authors demonstrate that spectral textures, which spectra are converted into Fourier coefficients, can be effectively compressed using the JPEG algorithm. Unlike the MESE method [6], this approach maintains precise spectral values at sample points and employs Fourier series for spectrum reconstruction.

Study [8] proposes a hybrid spectral representation, which stores low-frequency spectral components using a truncated Fourier series, while handling high-frequency components through conventional spectral representation. During rendering, the Fourier-based spectral data is converted into discrete spectral samples. This approach maintains linear computational complexity for spectral operations while enabling full-spectrum transmission.

3. Representing spectrum using cosine Fourier series

Let us consider a signal function $f(\phi)$ with a period of 2π, $f(\phi) \geq 0$. We then represent this function using Fourier coefficients through a set of basis functions:

$$c_j(\phi) = \frac{1}{2\pi}(\exp(-ij\phi)) \in \mathbb{C},\ j = 0 \dots m. \tag{1}$$

The resulting coefficients will be:

$$f_j = \int_{-\pi}^{\pi} f(\phi) c_j(\phi) d\phi \in \mathbb{C},\ j = 0 \dots m. \tag{2}$$

Next, consider the spectral distribution function $g(\lambda)$ over the visible interval $\Lambda = [\lambda_0, \lambda_1]$. We will represent it as an even signal $f(\phi)$: using the substitution

$$\phi(\lambda) = \pi \frac{\lambda - \lambda_0}{\lambda_1 - \lambda_0} - \pi \tag{3}$$

map it to $[-\pi, 0]$ and mirror it relative to zero.

As function $f(\phi)$ is even, we can represent it using coefficients:

$$f_j = \frac{1}{\pi} \int_{-\pi}^{0} f(\phi) \cos(j\phi)\, d\phi \in \mathbb{R},\ j = 0 \dots m. \tag{4}$$

Therefore, we can approximate original signal as trigonometric Fourier series, by truncating high-frequency moments:

$$f(\phi) \cong f_0 + 2 \sum_{j=1}^{m} f_j \cos(j\phi). \tag{5}$$

3.1. Fourier basis in spectral rendering

One of the issues of spectral rendering is color noise, which is a result of using sparse sampling of visible spectrum. This effect is particularly visible in scenes containing light sources or materials, which use spectrum with sharp peaks, or transparent objects with wavelength-dependent refractive indices. This problem can be mitigated by using larger sample size, which on the other side eases computational complexity.

Consider light ray, which intersects with surface of light source or some material, defined by bidirectional reflectance distribution function (BRDF). Energy of incident ray and reflectance for a given direction can be represented by their respective Fourier moments:

$$f(\phi) = \sum_{i=0}^{m} a_i \cos(i\phi),\ g(\phi) = \sum_{j=0}^{m} b_j \cos(j\phi). \tag{6}$$

Then, the energy of reflected ray will be given as:

$$f(\phi) g(\phi) = \sum_{i=0}^{m} \sum_{j=0}^{m} a_i\, b_j \cos(i\phi) \cos(j\phi). \tag{7}$$

We can expand the cosine product:

$$\cos(i\phi) \cos(j\phi) = \frac{1}{2}\left[\cos\big((i+j)\phi\big) + \cos\big((i-j)\phi\big)\right], \tag{8}$$

where $\cos(-k\phi) = \cos(k\phi)$.

As this representation stores a constant number of Fourier coefficients, we need to truncate high-frequency components, which leads to accumulating error for multiple ray scattering events, thus resulting in significant spectral distortions. However, most of the energy of the spectrum is still stored in its first Fourier moments (Fig. 1).

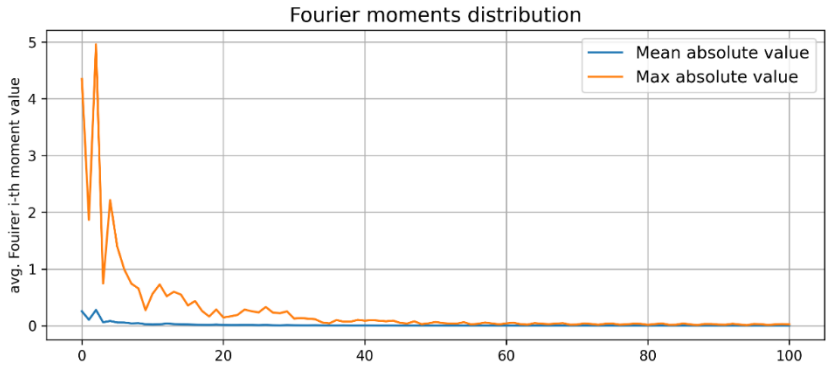

Fig. 1. Absolute values of Fourier coefficients for spectral dataset of lamps [17].

The resulting distribution is then described as a simple convolution of original spectra represented by truncated Fourier series:

$$f(\phi)g(\phi) = \sum_{i=0}^{m} c_i \cos(i\varphi), \tag{9}$$

$$c_i = \frac{1}{2}\sum_{p+q=i} a_p\, b_q \; + \; \frac{1}{2}\sum_{|p-q|=i} a_p\, b_q. \tag{10}$$

The final formula enables computation of Fourier coefficients for the product of two signals.

The Fourier coefficients of the spectrum obtained through rendering must then be converted into color. This can be achieved through several methods. The simplest conversion approach involves reconstructing the spectrum using the Fourier series. The resulting spectrum is then transformed into coordinates of the XYZ color model:

$$X = \int_\Lambda s(\lambda)\overline{x}(\lambda)d\lambda,\; Y = \int_\Lambda s(\lambda)\overline{y}(\lambda)d\lambda,\; Z = \int_\Lambda s(\lambda)\overline{z}(\lambda)d\lambda. \tag{11}$$

As most spectral rendering systems convert spectrum to color by integrating it over some fixed set of wavelengths [3], we can optimize conversion by precomputing cosine look up table for Fourier series and using it for conversion.

Another conversion method is using the result of convolution with precomputed Fourier moments for $\overline{x}$, $\overline{y}$, $\overline{z}$ functions. Those coefficients will correspond to a product of original spectrum with color-sensitivity functions. Then, the zeroth Fourier moment a_0 of function $f(\phi)$, corresponding to spectrum $g(\lambda)$ is equal to integral of spectrum with some multiplier:

$$a_0 = \int_{-\pi}^{0} f(\phi)d\phi \; = \; \frac{1}{\lambda_1-\lambda_0}\int_{\lambda_0}^{\lambda 1} g(\lambda)d\lambda, \tag{12}$$

which means that X, Y, Z are equal to corresponding Fourier coefficients, multiplied by$(\lambda_1 - \lambda_0)$.

Alternatively, instead of Fourier series, the MESE method can be used to convert the coefficients into a spectrum.

It should be noted that in conventional wavelength-sampled rendering, spectrum-to-color conversion occurs at every iteration (which avoids the need for large framebuffer storage). With Fourier coefficients, this issue is significantly mitigated since the number of coefficients can be substantially smaller than the number of wavelength samples. This enables storing a coefficient framebuffer and deferring color computation until after path tracing is complete.

3.2. Variance reduction using Fourier basis

Beyond direct calculation of spectral illumination using Fourier series, this method can also be applied to reduce variance in classical wavelength-sampled spectral rendering. Since the Fourier-based method provides a correlated biased estimate of the full-spectrum distribution, it naturally lends itself to the control variates technique [9].

Suppose we need to estimate the expected value of the pixel color calculation function for an image $\ell = \mathbb{E}[H(X)]$, where $X-$ represents a wavelength sample from the visible spectrum, and $Y = H(X)$ denotes the pixel color estimate for this sample in the CIE XYZ color representation.

The original Monte Carlo method estimates color as:

$$\widehat{Y_n} = \frac{H(X_1)+\cdots+H(X_n)}{n}. \tag{13}$$

The variance of this is estimated as: $\mathrm{Var}\left[\widehat{Y_n}\right] = \frac{1}{n}\mathrm{Var}[H(X)]$.

Suppose we want to reduce variance for a single ray in path tracing using a small wavelength sample. Let $C = F(X)$ denote the pixel color estimate based on wavelengths sample X from Fourier-based spectrum, $\mu = \mathbb{E}[C]$. Then, the control variate estimation will be:

$$\hat{\ell} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \alpha(C_i - \mu)\right) = \widehat{Y_n} - \alpha\left(\overline{C_n} - \mu\right). \quad (14)$$

If we find optimally tuned parameter $\alpha^* = \frac{\mathrm{Cov}(Y, C)}{\mathrm{Var}[C]}$, variance reduction could be expected:

$$\mathrm{Var}\left[\hat{\ell}\right] = (1 - \rho^2)\mathrm{Var}\left[\widehat{Y_n}\right] < \mathrm{Var}\left[\widehat{Y_n}\right], \quad (15)$$

where ρ is correlation between Y and C [10]. However, optimal value of parameter α^* cannot be precomputed, thus it is often approximated by estimating parameter with small starting sample or on every batch separately, which theoretically may create some bias against optimal solution.

3.3. Spectral material computation and storage

The key feature of spectral rendering is its ability to visualize phenomena that are direct consequences of the spectral nature of light. Only this approach can accurately depict the interaction of light sources and materials with complex spectral distributions, as well as phenomena such as interference, diffraction, dispersion, and polarization. A clear limitation of the considered method is its inability to account for wavelength-dependent light propagation direction, as it transports energy across the entire visible spectrum. Supporting such phenomena requires specialized approaches, which often introduce significant noise in the final image. In contrast, the described method provides a relatively fast approximation for visualizing spectral scenes with minimal memory overhead and virtually no color noise, at the cost of neglecting the aforementioned wavelength-dependent effects.

The ability to propagate the entire spectrum simultaneously introduces another limitation: the difficulty in calculating arbitrary BRDFs. Certain cases remain straightforward. For instance, diffuse materials described by the Lambertian model reduce to a single scalar multiplication scaled by the albedo spectrum, requiring only the storage of a separate spectrum or spectral texture. However, for universal models like Cook-Torrance, as well as for dielectrics, conductors, and many other materials, computing the Fresnel term becomes challenging. One potential solution involves using approximations, such as the Schlick model [11] or a more accurate approach based on spectral reflectance decomposition [12].

Another approach involves using precomputed lookup tables for reflectance and transmittance. These tables have a resolution of $N \cdot L$, where N is wavelength sample size and L is the number of selected angles. This number can be reduced to $M \cdot L$ (where $M = m + 1$ is number of Fourier coefficients), assuming the spectrum is sufficiently smooth and well-represented by the discrete Fourier transform – an assumption that generally holds true for reflection and transmission spectra. This approach proves particularly valuable for complex materials like single-layer and multi-layer thin films.

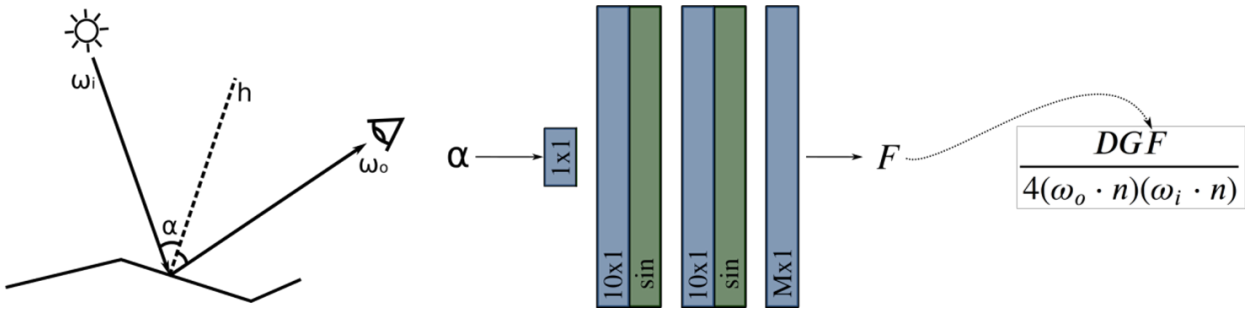

Fig. 2. Calculation of Fresnel component for Cook-Torrance model with neural network.

The pipeline supports defining BRDFs or their components as neural networks. Fig. 2 demonstrates calculating the Fresnel component within the Cook-Torrance model. Following approaches similar to [13, 14], neural BRDF functions are implemented as MLP with a small number of hidden layers and neurons. The number of output neurons matches the

count of Fourier coefficients used. Training employs reference BRDF functions with randomized input parameters. Fig. 3 compares the original, Fourier-reconstructed (using 10 coefficients per observation angle), and neural network-predicted Fourier basis reflectance tables for a dielectric thin film. Example shows that for the smooth distribution, few parameters are required. The neural network in Fig. 2 uses 2 hidden layers with $H = 10$ neurons each, a periodic activation function (sinusoidal), and outputs 10 Fourier coefficients. Training requires approximately 2 minutes on an RTX 4070 GPU.

Using compact neural networks eliminates the need for high-resolution lookup tables and reduces memory consumption for reconstructing smooth signals. Unlike similar work on neural spectral BRDFs [14], a single network pass generates the entire spectrum rather than a single wavelength value, which significantly accelerates rendering. Beyond enabling compact storage, neural materials have another valuable property, such as differentiability, which makes them suitable for inverse rendering. Neural networks are also possible to represent BRDF with spatially varying properties (SVBRDF) that depend on material parameters at specific surface points. These parameters are defined in textures (created by artists or learned through neural network training as in [13]) and fed into the network during material evaluation.

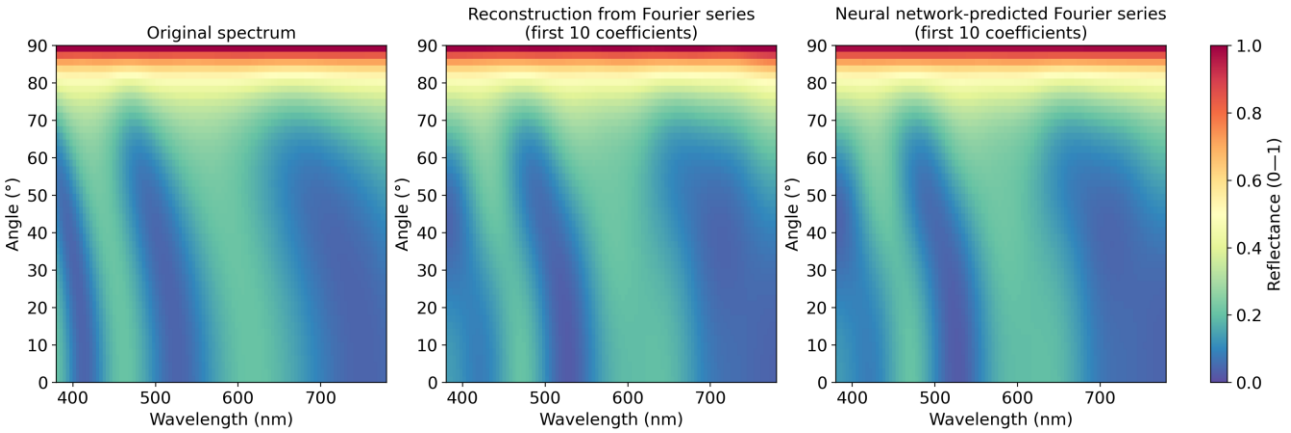

Fig. 3. Reflectance look-up tables:
1) explicit storage (LN parameters),
2) reconstruction from precomputed Fourier series (LM parameters),
3) neural network-predicted Fourier series reconstruction ($H^2 + HM + 3H + M$ parameters)

4. Experiments

4.1. Integrating the proposed method into a spectral visualization pipeline

At the current stage of work, the spectral visualization method using the Maximum Entropy method and the method, which uses truncated Fourier series, have been implemented within the HydraCore 3 rendering system [3].

To work with Fourier series, several methods for converting the resulting coefficients into color were implemented: using Fourier series, using a lookup table, and using convolution with color matching functions. Two cases were considered: conversion to color at the end of rendering (where the Fourier coefficients were stored in a separate buffer) and on each iteration.

The implemented methods were compared by rendering several scenes (Fig. 4-6). The PSNR and CIE76 ΔE metrics were used as a quality measure. CIE76 ΔE can be calculated with a following formula:

$$\Delta E = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}, \quad (16)$$

where (L_1, a_1, b_1), (L_2, a_2, b_2) are colors in CIELAB color space.

Rendering was performed at a resolution of 512x512 with 64 samples per pixel. The comparison results can be seen in Table 1.

Table 1. Comparison of different rendering methods (CPU: AMD Ryzen 7 9700x @ 3.80 GHz, 32 GB RAM).

	Fourier series	**Lookup table**	**Zeroth coefficient**	**MESE**	**Spec32**[1]
CIE ΔE	**11.5498**	**11.5498**	11.5577	11.5578	11.6477
PSNR	17.30	17.30	17.29	17.29	**17.39**
Rendering time (spectral buffer), sec	7.8	7.5	**7.4**	9.8	–
Rendering time (RGB buffer), sec	31.1	11.1	**7.5**	162.3	8.3

The results indicate that the conversion method based on the zeroth Fourier coefficient was the fastest in both cases, while maintaining a low error compared to using the full set of Fourier coefficients. Furthermore, the table demonstrates that the Fourier-based methods achieve a lower error compared to conventional spectral rendering with a sample size of 32.

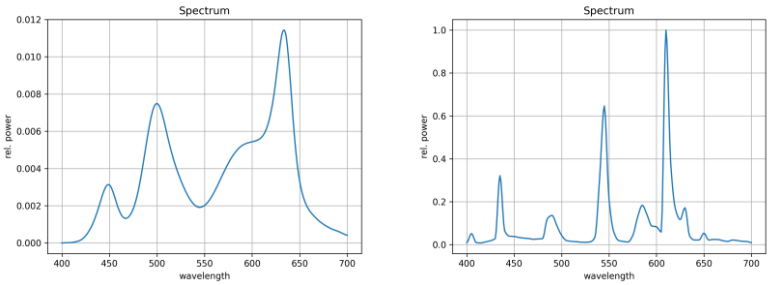

Fig. 4. Spectral power distribution plots of the lamps in Scene 1.

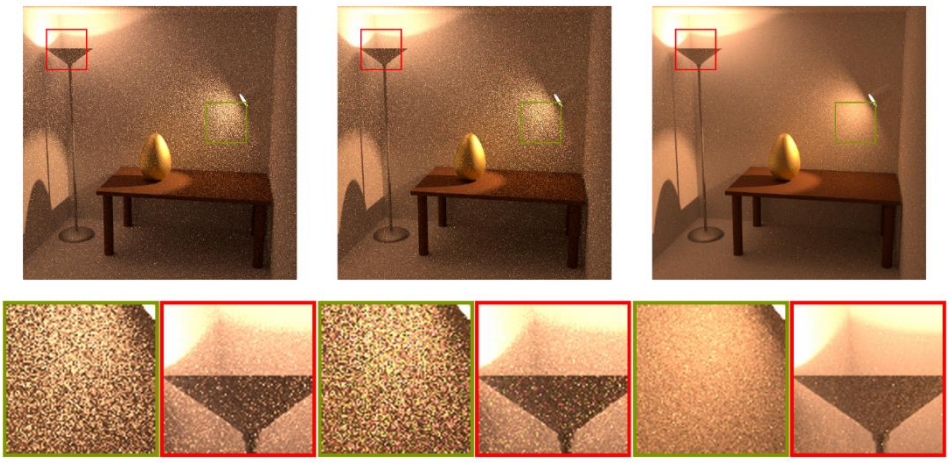

Fig. 5. Scene 1 (256 spp): our method, baseline spectral rendering (32 spectral samples), and high-sample reference.

[1] Rendering time with a separate spectral buffer was not measured.

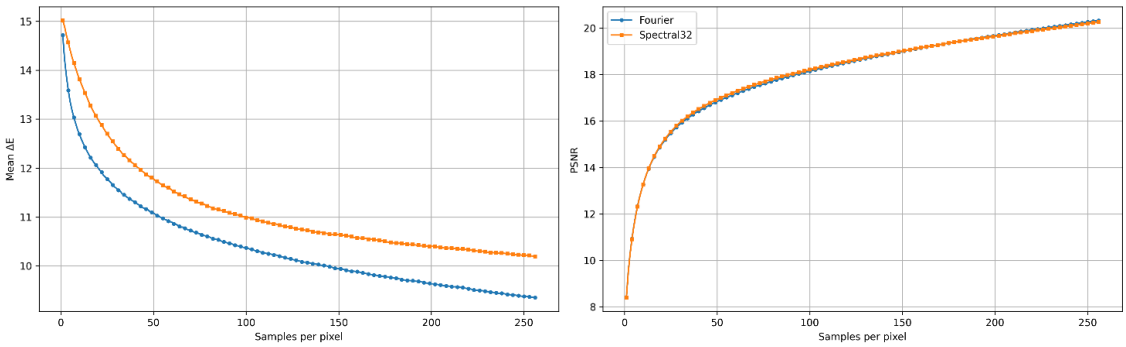

Fig. 6. Average CIE ΔE and PSNR vs. samples per pixel (spp) for rendering (scene 1)

As seen in Figure 6, the Fourier-based method yields a lower average color error than the 32-sample spectrum. Furthermore, the PSNR is approximately equal for both methods, which can be attributed to spatial noise having a greater impact on this metric than color inaccuracy.

We also used Intel® Open Image Denoise [15] library to evaluate the impact of our spectral representation technique on the denoised visualization of the spectral scene. The baseline spectral rendering brings noticeable color distortions, whereas our approach produces more stable results (Fig. 7).

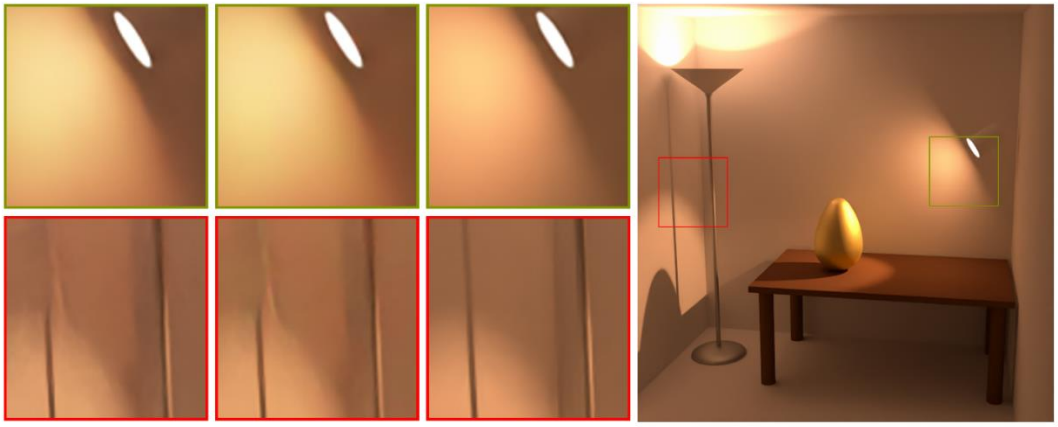

Fig. 7. Scene 1 (256 spp, denoising): our method, baseline spectral rendering (32 spectral samples), and high-sample reference.

Additional experiments were performed with a spectrum with high-frequency peaks. The spectral distribution of the source can be seen in Fig. 8. The results are presented in Figs. 9, 10.

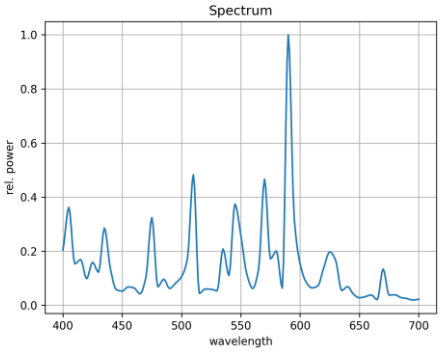

Fig. 8. Spectral distribution plot of the light source for scene 2.

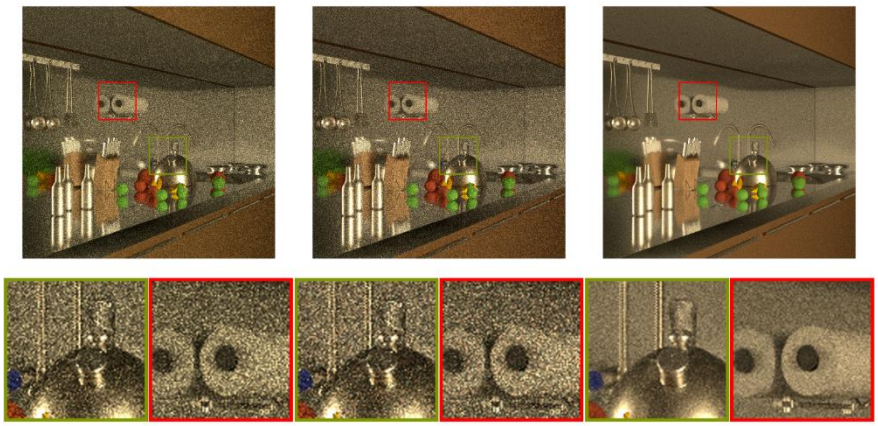

Fig. 9. Scene 2 (256 spp): our method, baseline spectral rendering (32 spectral samples), and high-sample reference.

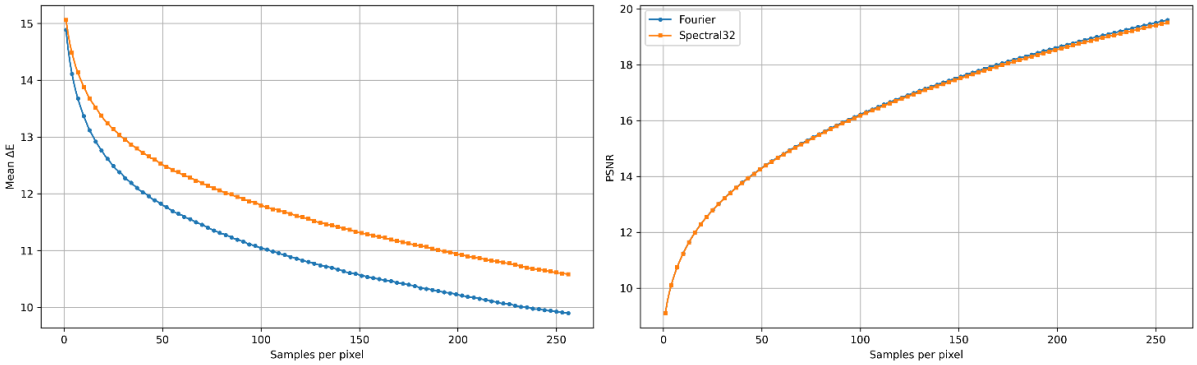

Fig. 10. Average CIE ΔE and PSNR vs. samples per pixel for rendering (scene 2)

Based on the rendering results for Scene 2, it can be observed that the proposed method significantly reduced color noise on test scenes.

4.2. Variance reduction via control variates

To test the Fourier method when used as a control variable, an experiment was conducted: the direct illumination of a color calibration target was calculated for various light sources. For each method, the results were measured and compared against a reference.

The calibration target (ColorChecker® Classic 2002 GretagMacBeth, Fig. 11) contains 24 colors. The light-surface interaction was calculated using the following methods:

1) uniform wavelength sampling over the spectrum;
2) stratified sampling (uniform sampling within fixed wavelength ranges);
3) hero wavelength spectral sampling [16];
4) method based on a Fourier basis;
5) the Fourier basis used as a control variable for methods 1)-3).

The color deviation from the ground truth was evaluated using the CIE76 ΔE metric.

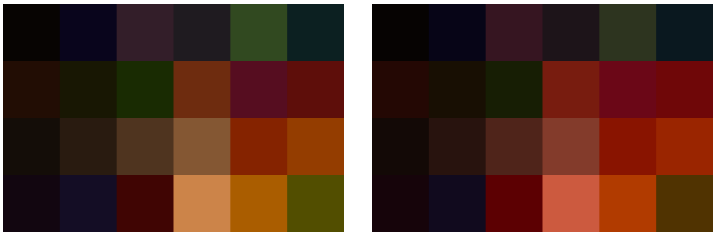

Fig. 11. A color calibration target illuminated by two different sources: ZJU-B-8 and CIE 507 [17].

Measurements for the two sources are presented in Fig. 12, 13. Methods based on wavelength sampling employ 12 coefficients for spectrum representation. The combined control variate approach uses 8 wavelengths and 4 Fourier coefficients.

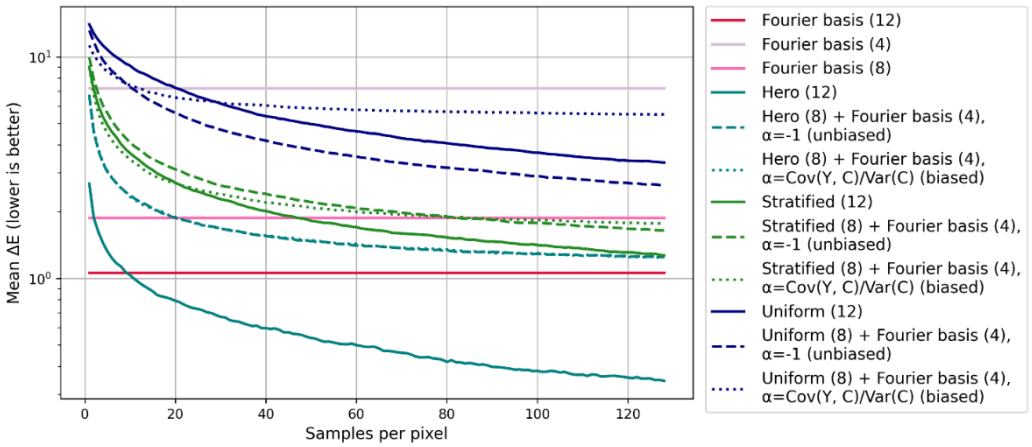

Fig. 12. Mean ΔE for direct illumination with a ZJU-B-8 standard illuminant.

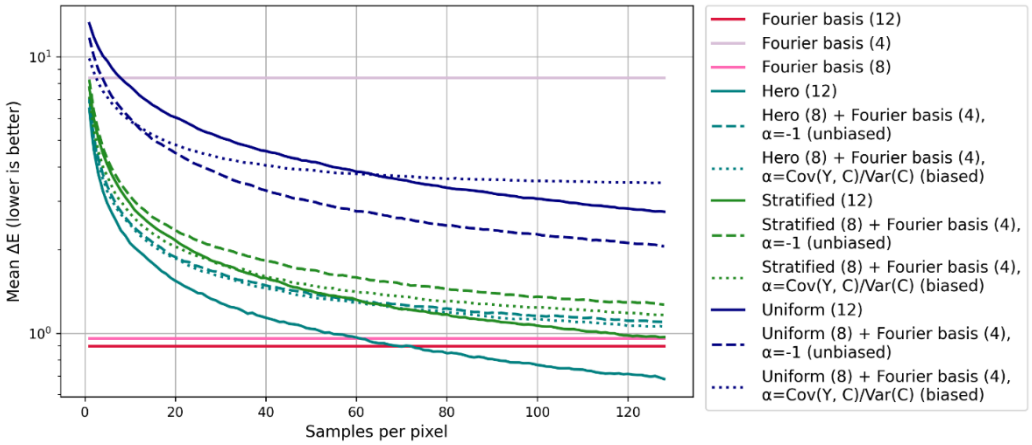

Fig. 13. Mean ΔE for direct illumination with a CIE 507 light source.

5. Discussion and conclusion

This work describes a spectral rendering method based on truncated Fourier coefficients. The comparison results of the considered rendering variants using Fourier coefficients, presented in Table 1, demonstrate that the proposed method achieves rendering quality comparable to the traditional approach. On scenes with light sources whose spectral distribution has prominent peaks, the method provides a gain in CIE ΔE metric.

We investigated various strategies for transforming these coefficients during rendering. The results further show that the optimal strategy for final spectrum-to-color conversion is to use the zeroth coefficient from the convolution of the spectrum with the color matching functions. It is also noteworthy that the MESE method, despite having proven itself as a viable spectral storage technique [6], provided no substantial advantage in combination with our approach and significantly increased rendering time.

The experiments demonstrated that the considered control variate method can provide a noticeable quality improvement in the ΔE metric, but only when using uniform or stratified wavelength sampling. Employing more advanced sampling techniques negates the benefits of the proposed idea.

The results demonstrate that our method achieves a lower level of color noise in the final image compared to traditional approaches, while maintaining comparable rendering quality and time. It's ability to produce low color noise even with a low sample-per-pixel

count highlights its potential for fast preview or interactive scene visualization, especially at low resolutions where color noise is most noticeable and significantly impacts perception. This approach also has potential applications in other areas, which will be explored in detail in future research.

Acknowledgments

The study was supported by the Russian Science Foundation, grant 25-11-00054, https://rscf.ru/project/25-11-00054/.

References

1. Physically based rendering system PBRT 4. https://github.com/mmp/pbrt–v4
2. Pharr M., Jakob W., Humphreys G. Physically Based Rendering: from theory to implementation. Fourth ed. Section 4.6.5. https://www.pbr–book.org/4ed/Radiometry,_Spectra,_and_Color/Color#ChoosingtheNumberofWavelengthSamples
3. Physically based rendering system HydraCore3. https://github.com/Ray–Tracing–Systems/HydraCore3
4. Evans G. F., Mccool M. D.: Stratified wavelength clusters for efficient spectral Monte Carlo rendering, Graphics Interface ’99, Morgan Kaufmann, 1999, pp. 42–49.
5. Jakob W., Hanika J.: A low–dimensional function space for efficient spectral upsampling, Computer Graphics Forum, 2019, Vol. 38 № 2, pp. 147–155.
6. Peters C., Merzbach S., Hanika J., Dachsbacher C.: Using moments to represent bounded signals for spectral rendering, ACM Transactions on Graphics, 2019, Vol. 38 № 4, pp. 1–14.
7. Fichet A., Peters C.: Compression of Spectral Images using Spectral JPEG XL, Journal of Computer Graphics Techniques (JCGT), 2025, Vol. 14 № 1, pp. 49–69
8. Sun Y.: A spectrally based framework for realistic image synthesis, Visual Comp. 2001, Vol. 17, pp. 429–444.
9. Lemieux C.: Control Variates, Wiley StatsRef: Statistics Reference Online, 2017, . 1–8.
10. Botev Z., Ridder A.: Variance reduction, Wiley StatsRef: Statistics Reference Online, 2017, pp. 1–6.
11. Schlick C.: An inexpensive BRDF model for physically–based rendering, Computer graphics forum, 1994, Vol. 13, pp. 233–246.
12. Belcour L, Bati M, Barla P.: Bringing an accurate fresnel to real–time rendering: a preintegrable decomposition, SIGGRAPH 2020: special interest group on computer graphics and interactive techniques conference talks. ACM, 2020, pp. 1–2.
13. Zeltner T., Rousselle F., Weidlich A., Clarberg P., Novák J., Bitterli B., Evans A., Davidovič T., Kallweit S., Lefohn A.: Real–time Neural Appearance Models, ACM Transactions on Graphics, 2024, Vol. 43 №3, pp. 1–17.
14. Chitnis S., Sole A., Chandran S.: Spectral bidirectional reflectance distribution function simplification, Journal of Imaging, 2025, Vol. 11 № 1, 18 p.
15. Áfra, A. T. (2025). Intel® Open Image Denoise. https://www.openimagedenoise.org
16. Wilkie A., Nawaz S., Droske M., Weidlich A., Hanika J.: Hero wavelength spectral sampling, Computer Graphics Forum, 2014, Vol. 33 № 4, pp. 123–131.
17. Royer, M. (2020). Real Light Source SPDs and Color Data for Use in Research. https://figshare.com/articles/dataset/Real_Light_Source_SPDs_and_Color_Data_for_Use_in_Research/12947240