

A method for modeling volumetric clouds and subsurface scattering of functionally specified objects

Sergey Vyatkin^{1*} and Boris Dolgovesov¹

¹Institute of Automation and Electrometry SB RAS, 630090 Novosibirsk, Russia

Abstract. Accurate modeling of light transfer in media is important in many fields, such as scientific visualization, medical imaging, synthesis of realistic images, and so on. Complex volumetric structures and translucent materials are used. Light scattering in media is responsible for the appearance of many materials. However, accurate and efficient modeling of lighting effects is a difficult problem. This requires solving not only the visualization equation, but also its volumetric solution, the radiation transfer equation. A method for modeling subsurface scattering of functionally specified objects is presented. The method allows you to simulate the transfer of light in chromatic media. To do this, a modified free-range distribution is used. This allows you to minimize the deviations of the bandwidth of the path. As a result, the variance of the estimate is reduced. The collision coefficients depending on the wavelength are effectively processed. As a result, it is possible to simulate volumetric clouds and subsurface light scattering in translucent materials such as marble, wax, leather, etc. The control coefficients and free run sampling are based on the minimum and maximum values in the media found by random sampling over a short preliminary pass. Reducing the rendering time is achieved by using a different distribution of paths.

1 Introduction

Modeling of light transfer is important in such areas as scientific visualization, medical applications [1], synthesis of realistic images, etc. Visual effects are used, in particular, for translucent materials such as marble, jade, wax, leather, etc. With their help, realistic images, depth and scale are created. However, modeling the transfer of light in such materials presents significant computational complexity. To do this, graphics processors (GPUs) are used in combination with specialized APIs. This makes it possible to interactively trace paths. Spectral rendering is used to accurately simulate light transfer, for example, to calculate light scattering [2, 3]. Spectral rendering is also used to render a scene with fluorescence. Since fluorescence is a physical phenomenon that strongly depends on the wavelength. Methods of fluorescence visualization under global illumination conditions are described in [4-6].

* Corresponding author: sivser@mail.ru

Wavelength samples are used taking into account both the spectra of fluorescent materials and light sources. The paper [7] shows the unification of several methods for sampling trajectories in space and in spectrally varying media. Monte Carlo algorithms of the general Markov chain and algorithms for modeling light transfer are described in [8]. Algorithms require providing only correct inverse sampling functions. This makes them suitable for modeling light transfer. The application of the theory of light transfer in practice is described in the article [9]. Such as obtaining architectural visual effects and light ratings for daytime lighting. As well as the potential of solar radiation on the surfaces of buildings for the implementation of photovoltaic systems. Path tracing is one of the methods of visualizing photorealistic images by modeling the physics of light propagation in a scene. The review [10] describes path-tracing methods. Photorealistic rendering and global illumination are based on models of materials from the visible spectrum. There are also methods for visualizing thermal infrared radiation. The paper [11] describes the physical principles of determining the temperature distribution in a 3D scene. However, the above methods require more resources than traditional RGB rendering pipelines. By increasing the available internal memory on GPUs, spectral mapping becomes possible.

In this paper, the strengths of spectral rendering are used. The proposed spectral path tracer approach is implemented on a CPU. The paper also offers solutions for the use of memory in the processing of asymmetric spectra and large scenes. The method solves the problems of efficient processing of collision coefficients depending on the wavelength. Several vectorized variants of weighted tracking are analyzed. A solution is proposed taking into account these options for setting the collision probabilities. This prevents a geometric increase in the bandwidth of the path. The mathematical basis of the method is derived directly from the radiation transfer equation. The integral expression of the zero collision algorithm is used. This mathematical framework defines spectral path tracking. That is, tracking is implemented directly from the radiation transfer equation. Thanks to this, there is no need for more complex mathematical algorithms. An additional approach is used to efficiently process the collision coefficients depending on the wavelength. Joint tracking of multiple wavelengths increases efficiency and reduces color noise. The proposed multi-wave tracking does not require an explicit choice of wavelength. The construction of several calculation methods is combined, in which path tracking is used. Tracking is based on deviation, using multiple sampling without explicitly estimating or approximating the distance of the probability density function.

2 Description of the method

2.1 Definition of quadrics and free forms

The main primitives are chosen surfaces of the second order - quadrics. A second-order three-dimensional function is defined by ten coefficients. The same amount of data is needed to describe the spatial location of one triangular face. A more compact representation of the model allows you to reduce memory costs when storing a database, and, consequently, to unload the transport routes between the computing components of the system and memory blocks. A class of free forms is built on this basis (Fig. 1). This means the ability to describe complex geometric objects by specifying a second-order deviation function from the base surface. In addition to the fact that such a representation is much more economical than a polygonal one in terms of memory consumption, during image generation it is necessary to subject only the coefficients of the bearing surface to geometric transformations, because the deviation function remains constant, thus reducing the database.

The surface of a quadric is the surface of a constant level of a second-order function $Q(x, y, z)$. Define a certain perturbation function, $R(x, y, z)$, which has the smoothness property. Let the function be expressed in the form [12]:

$$F'(x, y, z) = Q(x, y, z) + \sum_{i=1}^N f_i R_i(x, y, z), \quad (1)$$

Due to the smoothness of the cue, and preserve this property and, consequently, the surface of level zero will also be smooth. Thus, we have constructed a complex surface using a second-order function as the base one. However, in order to get the desired shape of the surface, you need to learn how to set the appropriate perturbation function in an acceptable way. A smooth perturbation is constructed from a quadratic function as follows. Define the inner region of Q as a part of the space where $0 < Q$. Then in the outer part of Q , we put R identically equal to zero, and in the inner part, we raise Q to the power:

$$R_i(x, y, z) = \begin{cases} Q_i^3(x, y, z), & \text{if } Q_i(x, y, z) \geq 0 \\ 0, & \text{if } Q_i(x, y, z) < 0 \end{cases}, \quad (2)$$

where $Q(x, y, z)$ is the perturbing quadric.

The object, which we call a free form, remains essentially a quadric, but taking into account the addition made by the perturbation to its own basic function. In turn, the free form can be a perturbation function for some other object.

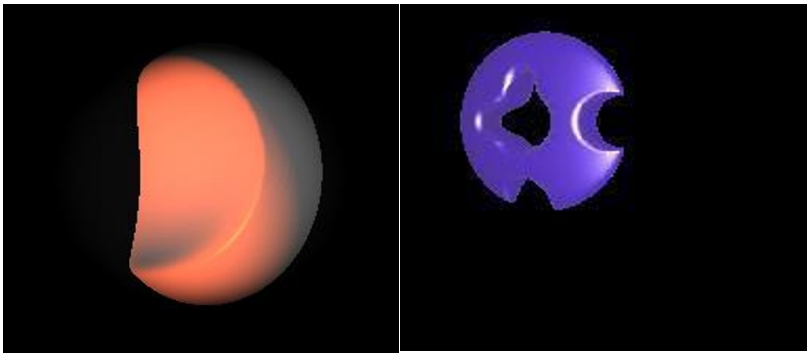


Fig. 1. Surfaces of free forms

2.2 Mathematical model

In order for the result of preliminary visualization of free-form prototypes to be unambiguously represented as OpenGL objects, it is necessary to have a good idea of the data structure of the scene, and its use in the algorithm for constructing the final image.

2.2.1. Mathematical representation of free forms

The primitive quadric is the basis for the construction of all other objects. The surface forming it is a second-order surface.

A quadric is defined using a real continuous describing function of three variables (x, y, z) in the form

$F'(x, y, z) \geq 0$ (1). Quadrics are considered as closed subsets of the Euclidean space E^3 , defined by the describing function $F'(x, y, z) \geq 0$, where F is a continuous real function, and (x, y, z) is a point set by coordinate variables in E^3 . Here $F'(x, y, z) > 0$ sets a point inside the quadric, $F'(x, y, z) = 0$ is a point on the boundary and $F'(x, y, z) < 0$ are points lying outside and not belonging to the quadric [13, 14].

An algebraic inequality of the second degree (with three unknowns x, y, z) is any inequality of the form:

$$F(x, y, z) = A_{11}x^2 + A_{22}y^2 + A_{33}z^2 + A_{12}xy + A_{13}xz + A_{23}yz + A_{14}x + A_{24}y + A_{34}z + A_{44} \geq 0, \quad (3)$$

where x, y and z are spatial variables.

2.3 Radiation transfer equation

Radiation transfer equation in differential form is [15, 16]:

$$(\omega \cdot \nabla) L(\vec{x}, \omega) = -\mu_t(\vec{x})L(\vec{x}, \omega) + \mu_a(\vec{x})L_e(\vec{x}, \omega) + \mu_s(\vec{x}) \int_{S^2} f_p(\omega, \varpi) L(\vec{x}, \varpi) d\varpi, \quad (4)$$

where $L(\vec{x}, \omega)$ is the radiation parameterized by the point $\vec{x}$ and direction of motion ω , $\mu_t(\vec{x})L(\vec{x}, \omega)$ is the losses caused by absorption and scattering, $\mu_a(\vec{x})L_e(\vec{x}, \omega)$ and $\mu_s(\vec{x}) \int_{S^2} f_p(\omega, \varpi) L(\vec{x}, \varpi) d\varpi$ are the gains caused by radiation $L_e(\vec{x}, \omega)$ and scattering, $f_p(\omega, \varpi)$ is the phase function quantifies the directional density of scattered light.

By solving the Dirac integral, an integral form of the radiation transfer equation is obtained, including zero collisions:

$$L(\vec{x}, \omega) = \int_0^\infty \exp\left(-\int_0^t \mu_t(\vec{x}_s) + \bar{\mu}_n(\vec{x}_s) ds\right) \left[\mu_a(\vec{x}_t)L_e(\vec{x}_t, \omega) + \mu_s(\vec{x}_t)L(\vec{x}_t, \omega) \right] dt, \quad (5)$$

where $\vec{x}_t = \vec{x} - t\omega$ and $\vec{x}_s = \vec{x} - s\omega$.

3 Tracking the path

Efficient processing of collision coefficients with spectral resolution is implemented. The distribution for sample collisions is determined. Then the discrepancy between this distribution and the true distribution of the free path per wavelength is eliminated by reweighing. Tracking occurs only once for all wavelengths. For each collision, the collision weight is calculated with spectral resolution. The weight can be expressed using a spectral basis or as a vector of wavelengths. Three primary colors are used - red, green and blue.

Continuous spectra are processed by transferring a discrete set of wavelengths in each path, using the wavelength selection and combining method. To keep the estimate variance low, a free sampling coefficient is set to limit the maximum value at all wavelengths.

The variance of spectral tracking depends on the variance of the spectrally resolved collision weights. The weight deviates from one, since it is impossible to set the collision

probabilities to cancel fractions for all wavelengths at the same time. By setting the collision probabilities, it is possible to determine the weight of the local collision, as well as the throughput of the path.

The weight of the local collision for all wavelengths is determined using the probability of the maxima for all wavelengths:

$$p_a(\vec{x}) = \max_{\lambda} (\mu_a(\vec{x}, \lambda))^c, \quad (6)$$

$$p_s(\vec{x}) = \max_{\lambda} (\mu_s(\vec{x}, \lambda))^c, \quad (7)$$

$$p_n(\vec{x}) = \max_{\lambda} (\mu_n(\vec{x}, \lambda))^c, \quad (8)$$

$$c = \max_{\lambda} (\mu_a(\vec{x}, \lambda))^c + \max_{\lambda} (\mu_s(\vec{x}, \lambda))^c + \max_{\lambda} (\mu_n(\vec{x}, \lambda))^c, \quad (9)$$

where c is the normalization constant; $\mu_a(\vec{x}) \in [0, \infty)$ is an absorption coefficient; $\mu_s(\vec{x}) \in [0, \infty)$ is a scattering coefficient; $\mu_n(\vec{x})$ zero collision coefficient.

Since the weights of local collisions can exceed one, multiplying the sequence of them can lead to a geometric increase in the throughput of the path. Therefore, we include the bandwidth of the path in the average probability of collisions. At each i -th vertex of the path, the bandwidth of the built-under path is included. Let the probabilities be express in the forms:

$$p_a(\vec{x}_i) = \text{avg}_{\lambda} (\mu_a(\vec{x}_i, \lambda) w(\vec{y}_{i-1}, \lambda))^c, \quad (10)$$

$$p_s(\vec{x}_i) = \text{avg}_{\lambda} (\mu_s(\vec{x}_i, \lambda) w(\vec{y}_{i-1}, \lambda))^c, \quad (11)$$

$$p_n(\vec{x}_i) = \text{avg}_{\lambda} (\mu_n(\vec{x}_i, \lambda) w(\vec{y}_{i-1}, \lambda))^c, \quad (12)$$

$$c = \text{avg}_{\lambda} (\mu_a(\vec{x}, \lambda) w(\vec{y}_{i-1}, \lambda)) + \text{avg}_{\lambda} (\mu_s(\vec{x}, \lambda) w(\vec{y}_{i-1}, \lambda)) + \text{avg}_{\lambda} (\mu_n(\vec{x}, \lambda) w(\vec{y}_{i-1}, \lambda)), \quad (13)$$

where $w(\vec{y}_{i-1}, \lambda) = \prod_{j=1}^{i-1} w(x_j, \lambda)$.

The average bandwidth over all wavelengths for all vertices of the path is equal to one. The maximum bandwidth for any wavelength is. This avoids a geometric increase in the capacity of the path during successive collisions. These probabilities reduce the amount of unwanted color noise. In addition, they prevent bright chromatic emissions.

For example, the material does not emit, it is possible to reduce the throughput. For this purpose, absorption collisions and redistribution of absorption probability between two types of collisions are not simulated.

Table 1 shows the coefficients used in the method.

Table 1. Collision coefficients

Symbol	Description
$\mu_a(\vec{x}) \in [0, \infty)$	Absorption coefficient
$\mu_s(\vec{x}) \in [0, \infty)$	Scattering coefficient
$\mu_t(\vec{x}) = \mu_a(\vec{x}) + \mu_s(\vec{x})$	Extinction coefficient
$\bar{\mu} \in (\mu_t^u, \infty)$	Path sampling rate

$\mu_n(\vec{x}) = \mu - \mu_t^u - \mu_t^h(\vec{x})$	Direct scattering coefficient
---	-------------------------------

Spectral tracking includes the following steps:

- 1) Is set $\bar{\mu}$ to limit the maximum value $\mu_t(\vec{x}, \lambda)$ for all wavelengths.
- 2) Each component μ_a^c and μ_s^c is set to the minimum values $\mu_a(\vec{x}, \lambda)$ and $\mu_s(\vec{x}, \lambda)$, accordingly, for all wavelengths.
- 3) Probabilities are established.
- 4) Equations (10), (11) and (12) are used.
- 5) Collision coefficients are vectorized for efficient processing of chromatic media.
- 6) The material is decomposed into an achromatic component and a chromatic component. Collision probabilities are derived from maxima or averages over all wavelengths.
- 7) The history of the path is taken into account.
- 8) If the material does not radiate, the completion rate decreases due to a decrease in the growth of the path capacity weights.

4 Results

Testing was performed on a 12 core 2.5 GHz Intel Xeon E5-2678 v3.

Figure 2 shows materials with spectrally varying absorption, scattering and extinction coefficients. Scene is generated using the path tracking method.

Spectral tracking increases the total number of path segments, the number of searches and visualization time by 20-25%. This is because smaller steps are used on average. Since the sampling coefficient of the free path should limit extinction for all wavelengths in a material with high chromaticity.

Spectral tracking data depends on the value of the sampling coefficient of the free path. Noise increases when the limiting piecewise constant $\mu(x)$ decreases. Using the proposed method, subsurface scattering is simulated (Fig. 2). Figure 2 shows translucent objects with a smooth dielectric boundary. Which covers a color volume. The free run sampling coefficients were based on the minimum and maximum values for the entire volume. They were calculated using a random sample in a short preliminary pass.

The reduction in rendering time during spectral tracking is due to the different distribution of paths. As well as lower throughput, since the algorithm completes paths earlier.

The effectiveness of path tracking also depends on how tightly the extinction coefficient can be linked from below. This is a difficult task for volumes and materials with a high degree of heterogeneity. Performance strongly depends on how well the spatial structure isolates and subdivides heterogeneous regions.

A feature of the proposed spectral path tracking method is the ability to process unlimited sampling coefficients of free path and control attenuation. The method allows you to quickly perform a preliminary calculation of the optimal sampling coefficients.

The method works well when searching in memory is expensive and data sets are large. In addition, if they are subject to cache overload and expensive high-order interpolation is used. If the volume is modeled with complex procedural functions, the savings can be significant.

The method is also effective for arbitrary values of collision coefficients if the sampling coefficient of the free run limits the extinction coefficient. Using the method, it is possible to calculate the optimal heuristic balance weight if the same tracking is used for sampling free range and transmission coefficient.



Fig. 2. Modeling of subsurface light scattering.



Fig. 3. A volumetric cloud with a functionally defined object inside.

Figure 3 shows a cloud visualized using the proposed method. Chromatic media are processed using the method and collision estimates are reduced.

Figure 4 shows graphs of changing RGB extinction for maximum values in the media. The abscissa axis shows the values of distance, the ordinate shows extinction.

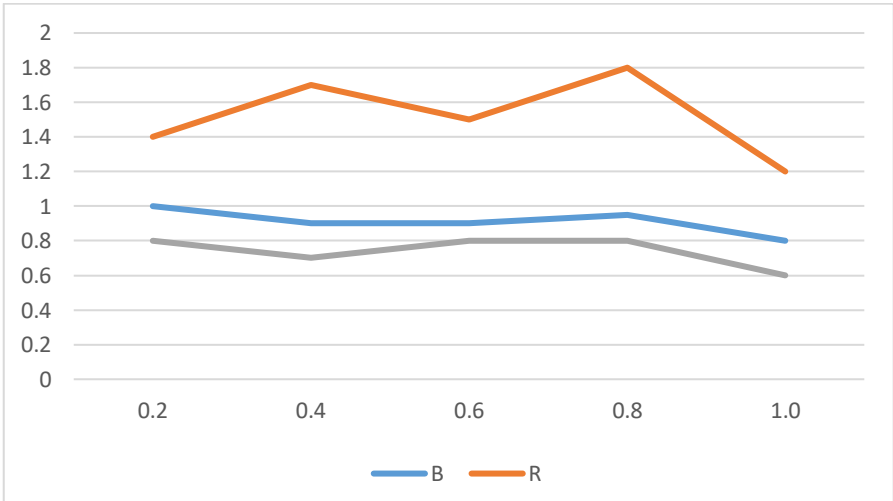


Fig. 4. Changing the RGB extinction for the maximum values in the media.

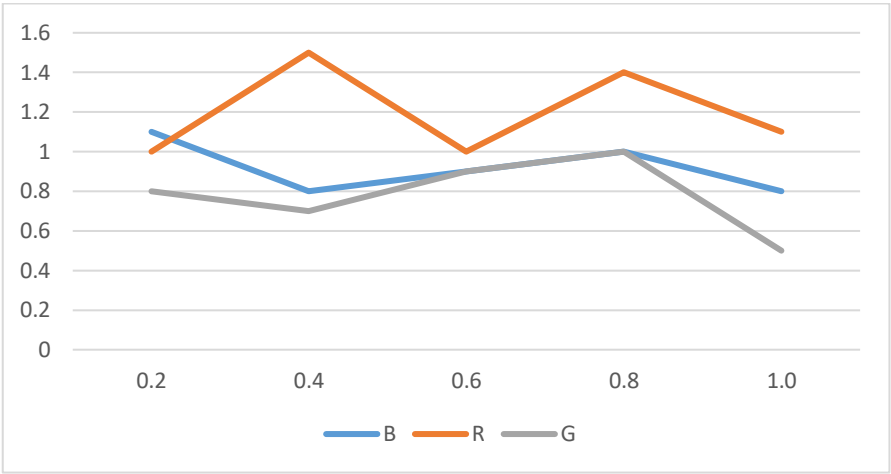


Fig. 5. Changing the RGB extinction for the average values in the media.

Figure 5 shows graphs of changing RGB extinction for average values in the medium. Figures 4 and 5 show the throughput of the path with multiple collisions and the average values of deviations depending on the distance. The distance is calculated using spectral tracking with different collision probabilities.

In this case, we consider the medium as non-absorbing, i.e. the albedo is equal to one, with uncorrelated scattering coefficients. The spectral tracker creates paths with an average number of scattering events. Track history accounting provides better path throughput and less estimation noise.

Figure 6 shows the scattering distribution for Figures 4 and 5. The abscissa axis shows the values of distance, the ordinate shows density.

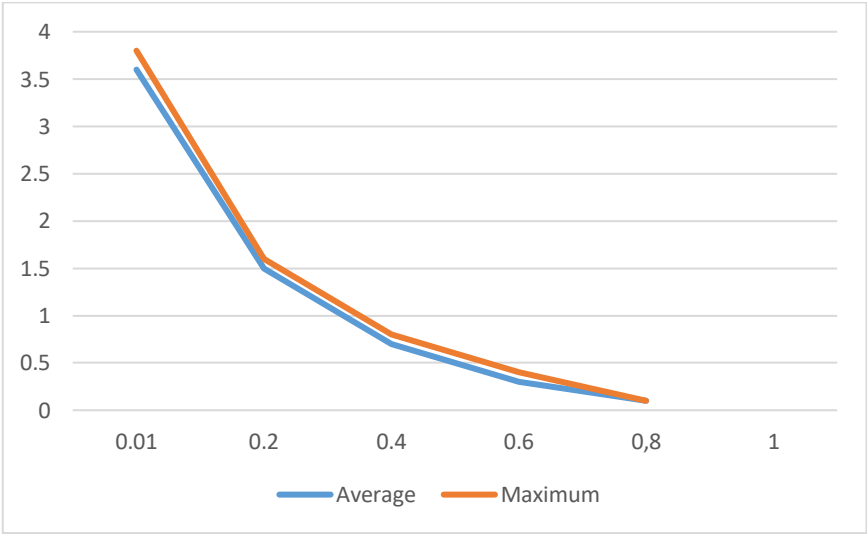


Fig. 6. Scattering distribution.

Table 2 compares the proposed path tracking algorithm with the octal tree algorithm, shows the effect of the depth of the octal tree on performance: as the depth increases, the calculation time increases. In the table. 2 also shows a decrease in the calculation time of the path tracking as the depth increases.

Table 2. Comparison of algorithms

Depth of the octal tree	Change in the calculation time of the octal tree, the number of times relative to the depth of one	
	Octal Tree algorithm	Path tracking algorithm
4	4.82	4.43
8	11.41	20.46
16	14.21	19.12

Table 2 shows that the decrease in the calculation time of the path tracking algorithm begins to decrease after depth 8.

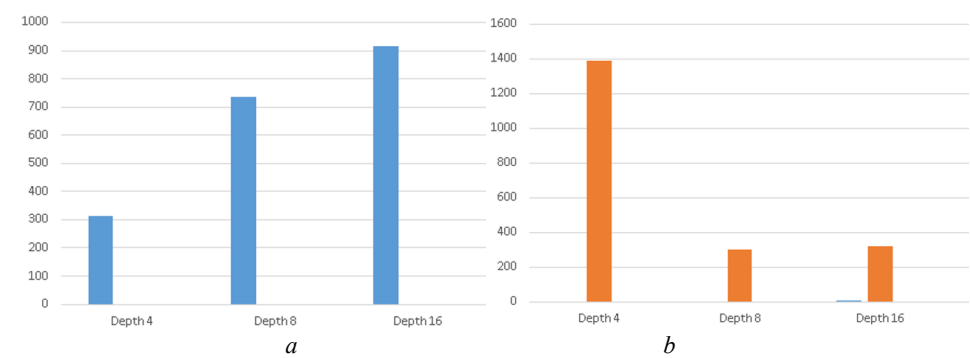


Fig. 7. Diagrams of changes in the calculation time of the octal tree (a) and path tracking (b).

Figures 7, a, b show diagrams of the increase in the calculation time of the octal tree depending on the depth and the decrease in the calculation time of the path tracking, respectively. The time on the ordinate axis is indicated in milliseconds.

5 Conclusion

The paper presents a method for modeling volumetric clouds and subsurface scattering of functionally specified objects. Spectral tracking reduces the variance when simulating subsurface light scattering. The method provides higher performance than known modern methods [14]. The method efficiently processes collision coefficients with spectral resolution. The proposed approach is based on the radiation transfer equation and the integral formula of the zero collision algorithm. The method can be used in computer graphics and scientific visualization, as well as for other areas of computer science.

References

1. C. Caredda, L. Mahieu-Williame, R. Sablong, M. Sdika, *Applied Sciences* **10(15)**,5158 (2020) <https://doi.org/10.3390/app10155158>

2. D. Murray, A. Fichet, R. Pacanowski, *Efficient Spectral Rendering on the GPU for Predictive Rendering*, Book, Chapter 42, A. Marrs, P. Shirley, I. Wald (eds.), *Ray Tracing Gems II* https://doi.org/10.1007/978-1-4842-7185-8_422021

3. A. Weidlich, A. Forsythe, S. Dyer, T. Mansencal, *Spectral imaging in production: course notes Siggraph 2021*. Conference: SIGGRAPH '21: Special Interest Group on Computer Graphics and Interactive Techniques Conference, August 2021. <https://doi.org/10.1145/3450508.3464582>

4. W. Yamamoto, B. Raytchev, T. Tamaki, K. Kaneda, *Spectral rendering of fluorescence using importance sampling*. Conference: SIGGRAPH Asia 2018, December 2018 <https://doi.org/10.1145/3283289.3283365>
5. M. Mojzik, A. Fichet, A. Wilkie, Computer Graphics Forum **37(4)**, 77-94 (2018) <https://doi.org/10.1111/cgf.13477>
6. A. Jung, J. Hanika, C. Dachsbacher, Computer Graphics Forum **39(2)**, 373-384 (2020) <https://doi.org/10.1111/cgf.13937>
7. B. Miller, I. Georgiev, W. Jarosz, ACM Transactions on Graphics **38(4)**, 1-13 (2019) <https://doi.org/10.1145/3306346.3323025>
8. J. Pantaleoni, ACM Transactions on Graphics **36(4)**, (2016) <https://doi.org/10.1145/3072959.3073677>
9. N. Jakica, *State-of-the-art review of solar design tools and methods for assessing daylighting and solar potential for building-integrated photovoltaics*", Project: IEA PVPS Task 15 BIPV - Action E.5 BIPV Design and Performance, January 2017, Renewable and Sustainable Energy Reviews 81. DOI:10.1016/j.rser.2017.05.080
11. P. H. Christensen, W. Jarosz, Foundations and Trends in Computer Graphics and Vision **10(2)**, 103-175 (2016) <https://doi.org/10.1561/06000000073>
12. L. Harake, E. Burkard, *Towards Physically Based Material Appearance in the Thermal Infrared Spectrum: A Short Survey*. Towards Physically Based Material Appearance in the Thermal Infrared Spectrum: A Short SurveyAt: Karlsruhe, Germany, July 2018. pp. 37-40 <https://doi.org/10.2312/mam.20181198>
13. S. I. Vyatkin and B. S. Dolgovesov, Optoelectronics, Instrumentation and Data Processing **58(3)**, 291–297 (2022) <https://doi.org/10.3103/S8756699022030116> ISSN 8756-6990
14. S.I. Vyatkin, B.S. Dolgovesov, *Modeling of Light Transfer in Spatially Correlated Media for Functionally Specified Scenes*. 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Yekaterinburg, Russian Federation, pp. 1560-1564 (2022,). <https://doi.org/10.1109/SIBIRCON56155.2022.10017079>.
15. S.I. Vyatkin, B.S. Dolgovesov, Programming and Computer Software **48(5)**, 322–330 (2022) <https://doi.org/10.1134/S0361768822050061>
16. A. Fasso, A. Ferrari, P. Sala, Radiation Protection Dosimetry **137(1-2)**, 118-33 (2009) <https://doi.org/10.1093/rpd/ncp190>
17. J. Leppänen, Annals of Nuclear Energy **37**, 715–722 (2010) <https://doi.org/10.1016/j.anucene.2010.01.011>