IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 26, 2021, accepted February 20, 2021, date of publication March 2, 2021, date of current version March 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063279

Hybrid Integration Method for Sunlight Atmospheric Scattering

TOMASZ GAŁAJ[1], FILIP PIETRUSIAK[2], MAREK GALEWSKI[2], RAFAŁ LEDZION[3], AND ADAM WOJCIECHOWSKI[1]

[1]Institute of Information Technology, Lodz University of Technology, 90-924 Łódź, Poland
[2]Institute of Mathematics, Lodz University of Technology, 90-924 Łódź, Poland
[3]Institute of Physics, Lodz University of Technology, 90-924 Łódź, Poland

Corresponding author: Adam Wojciechowski (adam.wojciechowski@p.lodz.pl)

ABSTRACT In computer graphics, the efficient rendering of a clear sky may greatly enhance the realism of synthesised interactive virtual environments. However, light atmospheric scattering, lying behind a reliable sky synthesis, is a computationally demanding process. Obtaining the highest possible accuracy and quality of the synthesised image in an efficient manner is not trivial as even minor integration errors may result in huge image artefacts. In this paper we propose two new approaches, namely the first based on the splines and the second on the Taylor expansion, in order to obtain the efficient computing of the transmittance (optical depth) component of the physically based single scattering model. We used Bruneton's framework to compare the proposed solution with state-of-the-art methods in a single, scattering-based clear sky synthesis, in terms of image quality and performance (time). We managed to improve the quality of the single scattering effect and time performance and we removed the necessity of pre-calculating values to look-up tables for the transmittance part of the scattering integral, reducing exploited memory usage.

INDEX TERMS Computer graphics, light atmospheric scattering, numerical integration, sky model, splines, Taylor expansion, transmittance.

I. INTRODUCTION

The rendering of light atmospheric scattering phenomena is a very time consuming task which requires burdensome computations. This becomes a crucial aspect affecting the performance and perception quality [2] of contemporary games and interactive virtual environments [3], [4]. Many authors [1], [5]–[7], have developed quite accurate methods in order to provide visually acceptable and physically reliable sky renders. These methods rely mainly on big pre-computed look-up tables which allow for the retrieval of rendering coefficients in real-time. Nevertheless, they occupy a considerable amount of memory and any scattering reconfiguration implies the necessity of look-up table recalculation.

Providing an efficient approximation of an atmospheric scattering phenomenon equation or its efficiently re-configurable solution may result in substantial memory and time savings, given that atmospheric scattering boundary conditions require dynamic redefinition and recalculation.

From a mathematical point of view, a light atmosphere-scattering phenomenon is described by an integral formula defining a model of light transmittance inside Earth's atmosphere (the model is presented in the next section of this manuscript). This model does not have a closed-form solution, so it has to be solved numerically. The state-of-the-art methods use the Trapezoidal Rule [8] to calculate such an integral. Our experiments have shown that the selection of other numerical methods for approximating integral formulas contained in the model, may have a considerable impact on the final quality of the image. Additionally, to the best of our knowledge, there were no attempts to approximate the integral formula in the process of light transmittance estimation. Therefore, our main contributions are as follows:

- novel, in the problem domain, approximation methods of the light transmittance integral model, derived in order to efficiently compute optical depth along the given segment on a view direction vector.
- elimination of the necessity of pre-calculating values for look-up tables for the transmittance part of the scattering integral.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaogang Jin.

- competitive numerical methods studies for the approximation of a scattering of the integral model in the context of the quality of clear sky renders.

II. ATMOSPHERIC SCATTERING BACKGROUND

In computer graphics the focus is put on two types of light scattering, namely the Rayleigh and the Mie scattering, which mainly affect the Earth's atmosphere light transmittance and resulting sky renders. The Rayleigh scattering occurs for particles that are smaller than the wavelength of light λ. On the other hand, when particles are bigger than λ then Rayleigh scattering smoothly transitions to Mie scattering. This type of scattering considers aerosols as well as small steam, ice and dust particles.

The major difference between these two types of scattering is that the intensity of Rayleigh scattering depends on the wavelength of scattered light. On the other hand, for larger particles sizes, Mie scattering is not strongly wavelength dependent. In practice, this implies that the blue sky colour is the effect of Rayleigh scattering combined with a lack of violet photon receptors in people's retinas [9]. On the other hand, the grey tones of halos around the sun, clouds and fog are caused by Mie scattering.

A. PHYSICALLY BASED MATHEMATICAL MODEL

In this section we present the equations defining the atmospheric single scattering model, which were proposed by Nishita [10] and were derived from theoretic observations in physics.

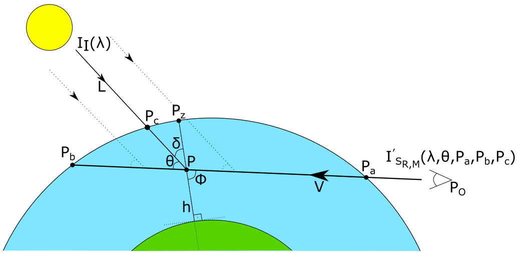


FIGURE 1. **Single scattering schema. The amount of light that enters the viewer's eye along viewing direction $\vec{V}$ (is transmitted from P_a to P_b) is calculated. Then, the amount of light that comes from the sun (is transmitted from P_C to P) that reaches every sample point P (at altitude h) along viewing direction $\vec{V}$ must be calculated. δ and ϕ represent sun-zenith and view-zenith angles respectively.**

1) SCATTERING INTENSITY

Scattering intensity for Rayleigh/Mie scattering is described by Formula 1. It expresses the amount of light that has been scattered exactly at point P (see Figure 1) (- absorbed and then emitted) towards direction $\vec{V}$ and with incident light direction $\vec{L}$ (Fig. 1).

$$I'_{S_{R,M}}(\lambda, \theta, P_a, P_b, P_c) = I_I(\lambda)\beta^s_{R,M}(\lambda)F_{R,M}(\theta) \\ \mathrm{D}_{R,M}(h(P), P) \quad (1)$$

Here λ is the spectral wavelength of the light, $I_I(\lambda)$ stands for the spectral intensity of the incoming light [11]. P_O is the location of the observer, P_a is the point where a ray from observer's position (P_O) enters the atmosphere along the viewing direction $\vec{V}$. P_b is the point where the ray from the observer's position exits the atmosphere or hits the surface. P_c is the intersection point of the incident light vector $\vec{L}$ and the atmosphere. Note that if the observer is inside the atmosphere then $P_O = P_a$ (origin of the ray from the observer's position is already inside the atmosphere). It is also assumed (as in the state-of-the-art methods) that all light rays that arrive at point P are parallel.

The other components: $\beta^s_{R,M}(\lambda)$, $F_{R,M}(\theta)$ and $D_{R,M}(h)$ are coefficients which are defined below for both Rayleigh (R) and Mie (M) scattering.

2) SCATTERING COEFFICIENTS

The Rayleigh scattering equation provides a scattering coefficient for the volume for which its molecular density is known. This coefficient is given by Formula 2.

$$\beta^s_R(\lambda) = \frac{8\pi^3(n^2-1)^2}{3N\lambda^4} \quad (2)$$

Here superscript s indicates that it is a scattering coefficient, subscript R means that it is a coefficient for Rayleigh scattering, λ is a given light's wavelength, n is an index of refraction of the air and N is a molecular density at the sea level.

Mie scattering is similar to Rayleigh scattering, but it applies to such particles that are considerably greater than the scattered light wavelength. These particles (aerosols) may be found at low altitudes of the Earth's atmosphere. Therefore, the equation for the Mie scattering coefficient needs to be slightly different (Formula 3).

$$\beta^s_M = \frac{8\pi^3(n^2-1)^2}{3N} \quad (3)$$

Here subscript M stands for Mie scattering and the rest of the parameters are the same as in the Rayleigh scattering coefficient equation.

3) PHASE FUNCTIONS

The phase function describes the angular dependency of scattered light on the direction of incoming light when its ray collides with a particle. This function takes as a variable the angle θ which is the angle between the scattered light ray and the incoming light ray. The result of this function is the amount of light that has been scattered. Formulas 4 and 5 show phase functions for Rayleigh and Mie scattering respectively.

$$F_R(\theta) = \frac{3}{16\pi}(1+\cos^2(\theta)) \quad (4)$$

$$F_M(\theta) = \frac{3}{8\pi}\frac{(1-g^2)(1+\cos^2(\theta))}{(2+g^2)(1+g^2-2g\cos(\theta))^{\frac{3}{2}}} \quad (5)$$

Here parameter $g \in (-1; 1)$ is an asymmetrical factor which describes the anisotropy of the medium (if it is a forward or a backward scattering). It should be noted that the Mie phase function [12] is very sophisticated and cannot be computed using a single formula. Precisely speaking, in Formula 5, the Henyey-Greenstein approximation of the Mie phase function, further applied in the considered solution, is shown.

4) OPTICAL DEPTH

Optical depth, or in other words transmittance, expresses how much of the light is being attenuated having travelled the distance $|P_b - P_a|$ in a medium. Formula 6 defines the transmittance.

$$T(P_a, P_b, \lambda) = \beta^s_{R,M}(\lambda) \int_{P_a}^{P_b} e^{\frac{-h(P)}{H_{R,M}}} dP \tag{6}$$

Parameter λ is a wavelength of the attenuated light, $h(P)$ is a function that describes the height above the ground level of a point, P. Optical depth attenuation is the effect of the negative ratio between out-scattering and in-scattering in participating media.

5) SINGLE SCATTERING

Single scattering means that only one light scattering event [13] is being taken into account after which some part of the light travelling towards an eye is deflected away from the viewing direction (out-scattering) or is deflected back on the viewing direction (in-scattering).

In Formula 1, the $D_{R,M}(h(P), P)$ term is expressed as:

$$D_{R,M}(h(P), P) = \int_{P_a}^{P_b} (\rho_{R,M}(h(P)) \cdot e^{-T(P,P_c,\lambda)-T(P,P_a,\lambda)})dP, \tag{7}$$

which depends on the density of air particles $\rho_{R,M}(h) = \exp(-\frac{h}{H_{R,M}})$ that decrease with respect to the altitude h. The division by $H_{R,M}$ is the respective scaling factor for Rayleigh and Mie scattering ($H_{R,M}$ equals either H_R or H_M). We use $H_R = 8000m$ and $H_M = 1200m$ according to [5]. The coefficients β^s_R and β^s_M are calculated by Bruneton's framework [1] during the run time.

It should be noted that the scattering equation model inherently depends on the transmittance integration. The efficient approximation of light transmittance may substantially reduce memory consumption and the time required for integration. Such an observation constituted the main motivation for our study.

The final intensity of single scattered light I'_S is obtained by evaluating and summing the light intensity in 1 for both the Rayleigh (I_{S_R}) and Mie (I_{S_M}) scattering event, i.e. $I'_S = I_{S_R} + I_{S_M}$.

B. TRAPEZOIDAL RULE

The Trapezoidal Rule approximates the area under a function via trapezoids (straight line segments). After this, area of the trapezoid is calculated very easily. This method may be also defined as averaging left and right Riemann Sums. It is expressed by Formula 8.

$$\int_a^b f(x)dx \approx \frac{b-a}{N}\left(\sum_{k=1}^{N-1} f(x_k) + \frac{f(x_N)+f(x_0)}{2}\right) \tag{8}$$

Here $[a, b]$ defines the integration interval and N is the number of samples.

The Trapezoidal Rule is used by most of the methods which compute the atmospheric scattering integral. To compute a single scattering (eq. 7) we need to compute the optical depth (eq. 6) of the medium (in our case it is Earth's atmosphere). It is easy to notice that we need to compute two integrals: the outer one (which is the main integral in formula 7) and the inner one (which describes the transmittance of the medium, eq. 6).

State-of-the-art solutions compute both integrals applying the Trapezoidal Rule, first to the inner integral and then the outer one to obtain the results for the single scattering term (eq. 7). The values of transmittance are stored in a 2D lookup table and the values of the single scattering term are stored in a 3D lookup table. Our solutions substitute computations of the inner integral (transmittance) with alternative approaches (splines or Taylor -based approximation), to reduce the computational overhead which is produced by the Trapezoidal Rule and to increase the quality of clear sky synthesis. Then, we use the computed values of the inner integrals to compute the single scattering term (outer integral).

III. RELATED WORK

One of the first papers concerning the subject of simulating light atmospheric scattering was published by Nishita *et al.* [10]. They presented a method for calculating atmospheric scattering taking into account the Rayleigh and Mie single scattering models. Their method was developed on a set of physically based equations (the Nishita model), for calculating the light's single scattering in the atmosphere. The Nishita model was a basis for the atmospheric scattering research in the field of computer graphics.

In 1996, Nishita *et al.* [14] improved their previous method of calculating atmospheric scattering of the light by taking into account multiple scattering (photons are scattered multiple times by different particles before reaching the observer's eye). None of the methods concentrated on light scattering efficiency but rather simulation numerical veracity.

Another, and widely used model is the Preetham model [15]. It was introduced in 1999 and provided the closed form approximation of the atmospheric scattering model. Therefore, the author was able to calculate colours of the sky in real time. Nevertheless, his approach has many major drawbacks as was pointed out by Zotti *et al.* [16]. In some specific situations, Preetham's model provides negative values of intensity and under some circumstances this model behaves incorrectly (the model breaks down numerically and it has unrealistic luminance distribution [1], [16]).

In 2012, Hosek and Wilkie [17] improved the Preetham model [15] by creating a fully spectral model with more degrees of freedom. As it turned out, the results of their method were not satisfactory from the point of view of rendering quality [1]. Due to inaccurate reconstruction of absolute and relative luminance, both Preetham and Hosek methods resulted in excessive, overestimated rendering relative error (RMSE).

In 2005, O'Neil [7] presented one of the first implementations of accelerating the atmospheric scattering computations on the GPU [18]. To be able to implement this phenomenon on the graphics card he had to simplify Nishita's model [10]. He presented a set of analytic functions that he used for computing the colours of the sky. These functions were approximations of those used in the Nishita's model [10]. Then he evaluated these functions per vertex (in vertex shader [19], [20]), obtaining the real-time performance while greatly sacrificing the quality of the generated image. On contrary to Preetham and Hosek, O'Neil method underestimated absolute luminance what resulted in unacceptable relative error of a synthesized clear sky image.

Haber *et al.* [21] presented a method that was based on [14]. They used a cell grid based on spherical coordinates, centred at the observer's position and computed multiple scattering more precisely than Nishita *et al.* [14] did, namely by integrating over all directions (over all cells in the grid) at each cell in the grid, instead of using eight co-planar specific directions as in [14]. Haber *et al.* also took into account refractive index of the air (curved light paths) and ozone layer absorption. Proposed improvements sacrificed real-time performance of the method.

In 2006, Wenzel [22], for the purposes of CryEngine 2, introduced a method that was based on precalculating the single scattering integral into a 2D lookup texture (LUT). It was the first implementation of a plausible sky rendering in a game engine, but unfortunately still far from expected real sky reference. Wenzel thus became a precursor of approaches of precomputing light scattering model integral values into lookup tables.

The first method that used precomputed single scattering values in real-time was introduced by Schafhitzel *et al.* [23]. The authors succeeded in precomputing single scattering equations in a big 3D lookup table as O'Neil suggested [7]. However, this 3D texture lacked one dimension as the authors did not take into account multiple scattering.

In 2008, Bruneton and Neyret [5] presented a method that was based on Schafhitzel *et al.* [23]. They presented the first real-time method which took multiple scattering into account and they successfully precomputed it in 4D lookup table and also precomputed the transmittance in a 2D lookup table. Their method worked for all viewing directions and virtual camera positions at any daytime, but required two separate precomputed lookup tables.

In 2009, Elek [6] presented a new lookup table based on the atmospheric scattering method. Based on the work of Schafhitzel *et al.* [23] he managed to reduce the dimensionality of the Bruneton 4D multiple scattering LUT [5] to 3D LUT while keeping the quality of the Bruneton method [1]. He also had to precompute the transmittance in an additional 2D lookup table.

In 2014, Yusov [24] proposed an improved parametrization of the Elek method that allowed the attainment of more details at the horizon and reduced the visible artefacts that could appear at the horizon. He also suggested calculating the scattering effect in the epipolar space.

It should be noted that all of the previously mentioned methods share one common feature - they all use the same numerical integration method, that is the Trapezoidal Rule. Moreover, the state-of-the-art methods [5], [6] had to precompute the transmittance integral in a 2D lookup table which is an additional, besides the scattering 3D/4D LUT, memory cost. All LUT-based approaches owe real-time performance to precomputations and resulting increase of memory complexity. LUT-based methods also sacrifice theirs flexibility as any light transmittance reconfiguration or atmospheric conditions changes imply LUT recalculation or further increase of memory complexity.

Real-time performance of clear sky rendering combined with reduction of assigned memory complexity drive motivation of conducted research. Contrary to what is known in the state-of-the-art literature, we have introduced an approximation of the transmittance integral which considerably reduces the time and memory required for light scattering calculation, preserving the quality of the light scattering synthesis. We have also examined several alternative numerical integration approaches for transmittance integral calculation proving that Splines and Taylor-based methods achieve better results than the traditionally used Trapezoidal Rule.

In this work, the investigations are concentrated mainly around single scattering since it is sufficient for proving image quality advancements of elaborated numerical solutions. Single scattering also allows us to confront elaborated solutions with a broader reference background as not all of the reference methods implement multiple scattering. Note that, one can easily enhance our method with multiple scattering and obtain even better quality in the final renders.

We also examine how alternative numerical integration methods affect the quality of the light scattering synthesis using Root-Mean-Square Error (RMSE) and ΔE (CIE 1976) [25] metrics.

IV. PROPOSED METHODS

A. GENERAL OVERVIEW

In what follows, we describe the path that led us to two methods. To the best of our knowledge, they were not described in scientific papers and both focus on simplifying Formula 6 - the transmittance calculations. More precisely, the integral that appears there reads:

$$\int_{P_a}^{P_b} e^{\frac{-h(P)}{H_{R,M}}}\, dP. \tag{9}$$

The first goal was to analytically simplify the problem of calculating the integral in Formula 9 for any segment $[P_a, P_b]$ inside the planet's atmosphere.

Precisely speaking, we are dealing here with the integration of the real function over the interval connecting two points in the $\mathbb{R}^3$ space. Unfortunately, there is no analytical solution to such an integral. The question is whether this problem can be reduced to a simpler one, or whether the same calculations will be correct for arbitrary pairs of points and what are the common line segments features, when spanned between different corresponding point pairs.

B. PROBLEM SIMPLIFICATION

Taking any two points inside the atmosphere of our planet P_a, P_b we can determine the distance between them. This segment is treated as a subset of a line in $\mathbb{R}^3$. We can see it in Figure 2.

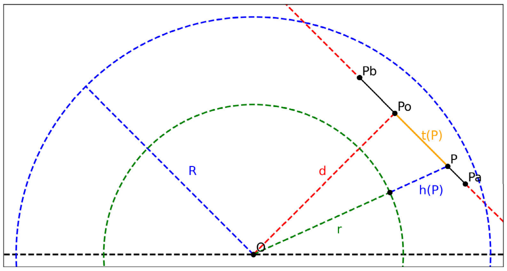


FIGURE 2. **Standard situation, ray is located at the distance d from the centre of the planet O with radius r, inside the atmosphere of radius R. Along the segment $[P_a, P_b]$, we see the sample point P which is the integration variable. Point P_o is the point with the minimum distance from the centre of the planet on the considered ray. The value $h(P)$ is the height of the point P above the planet's surface. The value $t(P) := \|P - P_o\|$.**

We can see that this line is located at distance d from the centre of the planet O with radius r, inside the atmosphere of radius R. On this line you can easily find point P_o with the minimum distance from the centre of the planet. On the interval between points P_a and P_b, we see point P which is the integration variable. The value $h(P)$ is the height of the point P above the planet's surface. The value $t(P) := \|P - P_o\|$, is the distance from point P to point P_o. In Figure 3 we can observe that it is possible to rotate the coordinate system in such a way, that the line becomes parallel to the horizontal axis of the system.

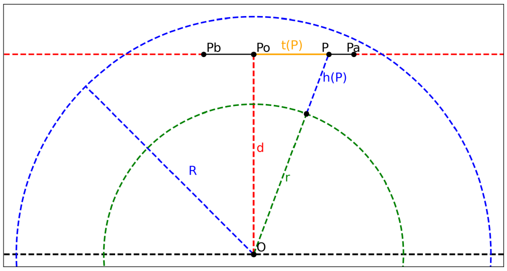


FIGURE 3. **Standard situation coordinate frame rotated in such a way, that point P_O is directly above the planet center O.**

If the line determined by the points P_a and P_b has a distance d smaller than r, the situation can be described in a similar way (see Figure 4).

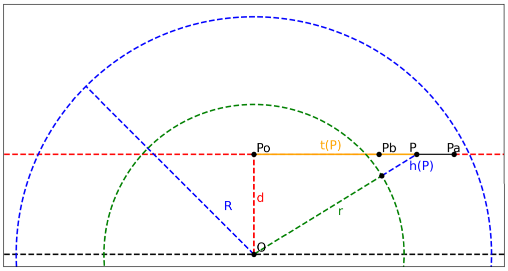


FIGURE 4. **Standard situation coordinate frame rotated in such way, that point P_O is directly above the planet centre O, when $d < r$.**

Taking into account the explanation of the notation and the drawings, we can see that

$$h(P) = \sqrt{d^2 + t^2(P)} - r,$$

moreover, from the line parameterization we have that there exists exactly one t_p such, that

$$P = P_o + t_P \cdot v \tag{10}$$

where $v := \frac{P_a - P_b}{\|P_a - P_b\|}$, then $\|v\| = 1$. Moreover, finding t_P, with the known P_a, P_b and P, is an analytically solvable and easy task. Hence we have

$$\begin{aligned} t(P) &= \|P - P_o\| \\ &= \|P_o + t_P \cdot v - P_o\| = \|t_P \cdot v\| = |t_P| \end{aligned}$$

And finally we see that

$$h(P) = \sqrt{d^2 + t_P^2} - r.$$

In conclusion, the following equality can be written

$$\int_{P_a}^{P_b} e^{\frac{-h(P)}{H_{R,M}}} dP = \int_{t_{P_a}}^{t_{P_b}} e^{\frac{-\sqrt{d^2+t_P^2}+r}{H_{R,M}}} dt_P, \tag{11}$$

where t_{P_a}, t_{P_b} are parameters of corresponding points with respect to Formula 10.

Therefore by the above mentioned calculations our problem has been written in a simpler manner. The answer to the question for which pairs of points we must perform the same calculations is: for such pairs of points which lie on lines of the same distance d from the center of the planet.

C. GENERAL IDEA

The distance from the centre of the planet for the lines passing through the atmosphere is in the interval $[0, R]$. Suppose we can calculate the integrals of the following form

$$\int_{t_{min}(d)}^{t_{P_b}} e^{\frac{-\sqrt{d^2+t_P^2}+r}{H_{R,M}}} dt_P, \tag{12}$$

for $t_{P_b} < t_{max}(d)$, where functions $t_{min}, t_{max} : [0, R] \to \mathbb{R}$ are defined by the following formulas

$$t_{min}(d) = \begin{cases} \sqrt{r^2 - d^2}, & \text{for } d < r, \\ 0, & \text{for } r \leq d \leq R. \end{cases}$$

$$t_{max}(d) = \sqrt{R^2 - d^2}.$$

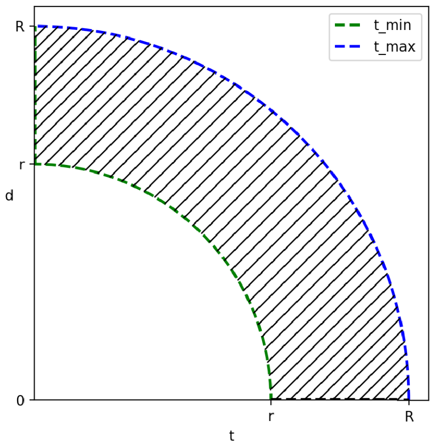


FIGURE 5. Region on which, we assume that, we know value of integrals from Formula 12.

In other words, we can calculate the integrals along the horizontal lines in the marked area in Figure 5, the lower limit of the integrals is on the green curve, and the upper limit of the integrals is on the same horizontal line between the green and the blue curve.

With the above assumption, we can compute the integral from Formula 9 using Formula 11 as follows

$$\int_{P_a}^{P_b} e^{\frac{-h(P)}{H_{R,M}}}\, dP = \int_{t_{min}(d)}^{t_{P_b}} e^{\frac{-\sqrt{d^2+t_P^2}+r}{H_{R,M}}}\, dt_P + \int_{t_{min}(d)}^{t_{P_a}} e^{\frac{-\sqrt{d^2+t_P^2}+r}{H_{R,M}}}\, dt_P \tag{13}$$

while the situation is as shown in Figure 3 (points are on both sides of P_O), in case both points are on the same side of the point P_O the same can be calculated as follows

$$\int_{P_a}^{P_b} e^{\frac{-h(P)}{H_{R,M}}}\, dP = \int_{t_{min}(d)}^{t_{P_b}} e^{\frac{-\sqrt{d^2+t_P^2}+r}{H_{R,M}}}\, dt_P - \int_{t_{min}(d)}^{t_{P_a}} e^{\frac{-\sqrt{d^2+t_P^2}+r}{H_{R,M}}}\, dt_P \tag{14}$$

Now you can see that with the assumption that we are able to compute integrals from Formula 12, we can calculate the transmittance value from Formula 6 for any points P_a and P_b located in the planet's atmosphere.

D. FIRST APPROACH

The first idea was to calculate the extreme values of Formula 9 for some number of d parameter values. By extreme we mean the minimum and maximum value of such an integral.

The minimum is of course zero when we integrate from $t_{min}(d)$ to $t_{min}(d)$, since the integration interval is then a singleton.

From the non-negativity of the integrated function, it is also easy to notice that the maximum value is obtained when we integrate from $t_{min}(d)$ to $t_{max}(d)$ because the integration interval is then the longest possible.

Then, extreme integral values, numerically using the Simpson method, have been calculated. We use the Simpson method due to the fact that it works more accurately in our case. The idea to use extreme values is to interpolate between 0 and $\int_{t_{min}(d)}^{t_{max}(d)} e^{\frac{-\sqrt{d^2+t_P^2}+r}{H_{R,M}}}\, dt_P$ with a certain sigmoidal function of the variable t_P, with its shape parameters depending, in a non-linear manner, on d. This dependency was descried by some hardcoded values and was consciously omitted in the first approach description due to poor generalisation of the solution.

E. SPLINE METHOD

The second approach is to use splines to approximate the integrated function over the entire area of the region (Figure 5). Splines are integrated easily, so by approximating the integrated function, we will be able to immediately calculate the approximate value of transmittance.

The most convenient way is to span a spline on a rectangular mesh with constant spacing for each of the coordinates. This is why we divide our area of interest into two sub-areas.

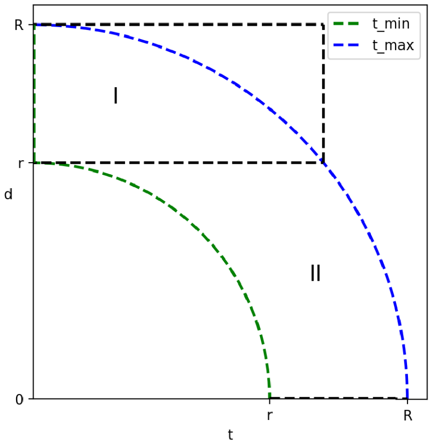


FIGURE 6. Regions for spline interpolation (our method).

First we will discuss the area I. It is delimited by a black dotted line from the top, bottom and right and a green dotted line to the left, see Figure 6. Thus it becomes the set $\left\{(t,d) \in \mathbb{R}^2 : t \in \left[0, \sqrt{R^2-r^2}\right], d \in [r,R]\right\}$.

Taking some integers N, K, we define set of points (t,d), where $t \in \left\{0 + \frac{\sqrt{R^2-r^2}}{N-1} \cdot i\right\}$ for $i \in \{0, 1, \ldots, N-1\}$, and $d \in \left\{r + \frac{R-r}{K-1} \cdot j\right\}$ for $j \in \{0, 1, \ldots, K-1\}$, which is simply a $N \times K$ size rectangular mesh in area I. We calculate the values of the integral function on this mesh $e^{-\frac{\sqrt{d^2+t^2}-r}{H_{R,M}}}$. With these values, we can compute spline 2d. You can see that the mesh also covers the area behind the blue curve (outside of the atmosphere), which is done on purpose.

However, it is difficult to use splines in the second area II, because of its irregularity. It is delimited by a black dotted line from the top and bottom, blue dotted line from the right and a green dotted line from the left, see Figure 6. To do this we will use the change of variable formula as follows

$$\int_{t_{min}(d)}^{t_{max}(d)} e^{-\frac{\sqrt{d^2+t^2}-r}{H_{R,M}}} dt = \left| \begin{array}{l} h = \sqrt{d^2+t^2}-r \\ dh = \frac{\sqrt{(h+r)^2-d^2}}{h+r} dt \\ dt = \frac{h+r}{\sqrt{(h+r)^2-d^2}} dh \end{array} \right| = \int_0^{R-r} e^{\frac{-h}{H_{R,M}}} \frac{h+r}{\sqrt{(h+r)^2-d^2}} dh.$$

Using the above calculations, we will compute the splines on the set $\left\{(h,d) \in \mathbb{R}^2 : h \in [0, R-r], d \in [0, r]\right\}$. Similarly to the area I we take some integers N, K, define the set of points (h,d), where $h \in \left\{0 + \frac{R-r}{N-1} \cdot i\right\}$ for $i \in \{0, 1, \ldots, N-1\}$ and $d \in \left\{0 + \frac{r}{K} \cdot j\right\}$ for $j \in \{0, 1, \ldots, K-1\}$, which is also a rectangular mesh of size $N \times K$ on the substitution-transformed area II. For the points in this mesh, we calculate the values of the integrand after substitution, i.e. $e^{\frac{-h}{H_{R,M}}} \frac{h+r}{\sqrt{(h+r)^2-d^2}}$. As before, with these values we can compute spline 2d on the substitution-transformed II area.

With both splines, to compute the integral from the Formula 9 we need to proceed with the following steps:

1) Find parameter d connected to our points (Figures 2, 3 and 4).
2) Calculate t_{P_a} and t_{P_b} if $d \geq r$, otherwise calculate $h(P_a)$ and $h(P_b)$.
3) Substitute computed d parameter in spline 2d, to get spline 1d.
4) Calculate spline integrals from 0 to t_{P_a} and from 0 to t_{P_b} if $d \geq r$, otherwise calculate spline integrals from 0 to $h(P_a)$ and from 0 to $h(P_b)$.
5) Use the above values as in Formulas 13 or 14.

F. TAYLOR FORMULA-BASED APPROXIMATION

The method we are going to describe is inspired by both the integration by substitution and by the approximation of a function using the Taylor formula.

As before, it is enough to observe that for any $d \in [0, R]$, $t_{P_b} \in [t_{min}(d), t_{max}(d)]$ we will be able to estimate the value of the integral from Formula 12.

Now we can try procedure of change of variable

$$\int_{t_{min}(d)}^{t_{P_b}} e^{-\frac{\sqrt{d^2+t_P^2}-r}{H_{R,M}}} dt_P \approx \left| \begin{array}{lcl} h(t_P) &=& \sqrt{d^2+t_P^2}-r \\ dh &=& \frac{t_P}{\sqrt{d^2+t_P^2}} dt_P \\ dt_P &=& \frac{\sqrt{d^2+t_P^2}}{t_P} dh \end{array} \right| \approx \int_{h(t_{min}(d))}^{h(t_{P_b})} \frac{\sqrt{d^2+t_P^2}}{t_P} \cdot e^{\frac{-h}{H_{R,M}}} dh, \tag{15}$$

and further approximation

$$\begin{aligned} \int_{h(t_{min}(d))}^{h(t_{P_b})} & \frac{\sqrt{d^2+t_P^2}}{t_P} \cdot e^{\frac{-h}{H_{R,M}}} dh \\ &\approx \frac{\sqrt{d^2+t_P^2}}{t_P} \int_{h(t_{min}(d))}^{h(t_{P_b})} e^{\frac{-h}{H_{R,M}}} dh \\ &\approx -\frac{H_{R,M}\sqrt{d^2+t_P^2}}{t_P} \left(e^{\frac{-h}{H_{R,M}}} \right|_{h(t_{min}(d))}^{h(t_{P_b})} \\ &\approx -\frac{H_{R,M}}{h'(t_P)} \left(e^{\frac{-h}{H_{R,M}}} \right|_{h(t_{min}(d))}^{h(t_{P_b})}. \end{aligned} \tag{16}$$

It is an approximation, not equality, because function h should be differentiable. Observe that when $d = t_{min}(d) = 0$, h it is not differentiable. We will show however that such an approach will lead to some clue as to how to approximate our integral. The value of the expression $\sqrt{d^2+t_P^2}/t_P$ for the considered physical conditions of the atmosphere is approximately constant and can be extracted before the integral. The problem with the non-differentiability of the function h can be solved by using some suitable approximation which in turn would be differentiable.

$$\begin{aligned} \widetilde{h}\left(t_{P_b}\right) := & \frac{a}{2} h''(\tau_b(d)) \left(t_{P_b} - \tau_b(d)\right)^2 \\ & + h'(\tau_b(d)) \left(t_{P_b} - \tau_b(d)\right) \\ & + h(\tau_b(d)), \end{aligned}$$

where $a, b \in [0, 1]$, and $\tau_b(d) := t_{min}(d) + b(t_{max}(d) - t_{min}(d))$.

We see that $\widetilde{h}$ is nothing else but the Taylor expansion of the function h at point $\tau_b(d)$. It may be more difficult to determine the order of the above expansion. When $a = 0$ this is the first order, while for $a = 1$ this is the second order. So we can say that in general it is an expansion of the order $1 + a$, i.e. a kind of interpolation between the expansion of the first and second order.

This observation is further justified by the fact that function h around the point $t_{min}(d)$ behaves like a quadratic function, and to infinity as a linear function. Therefore the function on the interior of the interval has growth between linear and quadratic.

Now we would like to use $\widetilde{h}$ instead of h. Using the Formulas 15, 16, one can suggest the following approximation

$$\begin{aligned} \int_{t_{min}(d)}^{t_{P_b}} & e^{-\frac{\sqrt{d^2+t_P^2}-r}{H_{R,M}}} dt_P \\ &\approx -\frac{H_{R,M}}{\widetilde{h}'(\tau_b(d))} \left(e^{\frac{-\widetilde{h}}{H_{R,M}}} \right|_{\widetilde{h}(t_{min}(d))}^{\widetilde{h}(t_{P_b})} \\ &= -\frac{H_{R,M}}{h'(\tau_b(d))} \left(e^{\frac{-\widetilde{h}}{H_{R,M}}} \right|_{\widetilde{h}(t_{min}(d))}^{\widetilde{h}(t_{P_b})} \\ &= -\frac{H_{R,M}}{h'(\tau_b(d))} \left(e^{\frac{-\widetilde{h}\left(t_{P_b}\right)}{H_{R,M}}} - e^{\frac{-\widetilde{h}(t_{min}(d))}{H_{R,M}}} \right). \end{aligned} \tag{17}$$

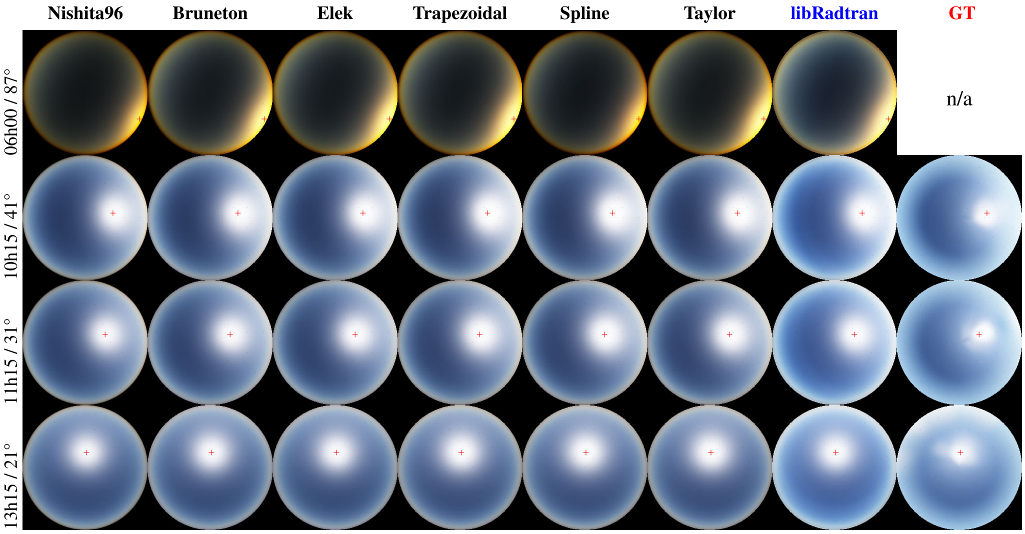


FIGURE 7. Rendering. Fisheye skydome rendering of the spectral radiance obtained with each model, convolved with the CIE color matching functions, converted from XYZ to linear sRGB, and tone mapped with $1 - e^{-kL}$, for several daytimes / sun zenith angle values (the red cross indicates the sun direction). The measurements are interpolated using bicubic spherical interpolation before rendering. Compare with Fig. 13 in [26].

The last thing to do is to choose values for a, b parameters, for which overall results are the most accurate - this part is described in the *Results and discussion* section. If these results are satisfactory, this method provides that we calculate the integral from the Formula 6 without any loop, directly from Formulas 13, 14 and 17.

V. RESEARCH METHODOLOGY

In the presented research, we have considered the calculations of the transmittance (Formula 6) and how it affects the single scattering term (Formula 7) for all methods (proposed and referenced). Such an approach should reveal the advantages of the proposed solutions and legibility of the core approximation ideas without the need of adding the multiple scattering complexity overhead.

To compare the proposed model with the references in a systematic way, we used Bruneton's framework [1]. We have analysed the same aspects (listed below) as Bruneton *et al.* did and compared it to the ground truth (GT) measurements of a real clear sky made by Kider *et al.* [26], computing RMSE and ΔE (CIE 1976) [25], a metric which is described by Formula 18.

$$\Delta E = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2} \qquad (18)$$

In this formula, L stands for the lightness from black to white, a from green to red and b from blue to yellow.

We also included libRadtran [27] results to get to know how far other methods are from the most accurate approximation the model developed by physicists (which is not usable in real-time applications or even in the film industry).

We performed a grid search of hyper parameters for the Splines method (Table 1) and the Taylor method. For the Splines method, we performed a grid search with respect to the lowest value of height and distance points (i.e. lowest grid density) that does not affect the quality of the final image (Table 1). Furthermore, we compared different types of splines to check whether they have any impact on reducing the computation time and improving image quality (Table 2). For the Taylor method, we performed the grid search of the parameters a and b for which the cumulative error (between the Ground Truth (GT) from Bruneton's framework [1] and the approximate formula proposed above) was the lowest.

Apart from these, we also compared the Trapezoidal Rule with the proposed methods in terms of the computation time and image quality using RMSE and ΔE metrics (see Table 3) compared with the measurements (GT). The lower the RMSE and ΔE values are, the better the method is.

Moreover, we also provide a discussion of the memory consumption of the proposed solutions and the state-of-the-art - the Bruneton method [5].

Furthermore, we provide the comparison of the proposed methods and other widely known methods (Nishita93 [10], Bruneton [5], Elek [6], the Trapezoidal Rule, libRadtran [27]) taking into account different aspects:

- Fisheye skydome renders (Figure 7) – are renders of the spectral radiance that were obtained for each method, convolved with the CIE color matching functions, converted from XYZ to linear sRGB and finally tone mapped using a $1 - e^{-kL}$ function. These images were rendered for several times of day/sun zenith angle values. The red cross indicates the sun direction. The measurements were interpolated using bicubic spherical interpolation prior to rendering.
- Absolute luminance figures (Figure 8) – present the absolute luminance in $cd.m^{-2}$ that was obtained for each method, using the same color scale as in Figure 7 from [16]. The measurements were interpolated using

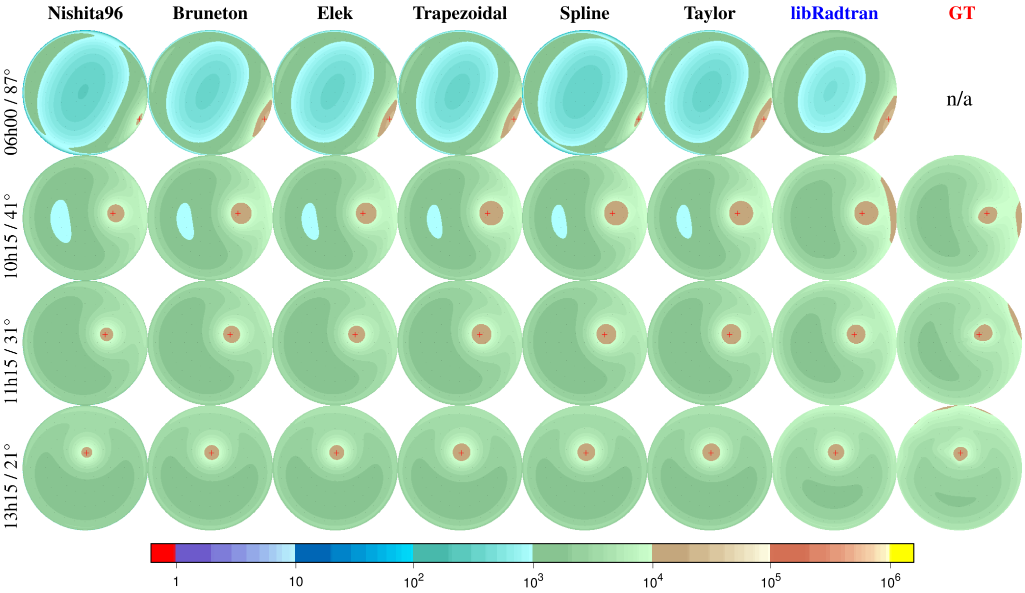


FIGURE 8. Absolute luminance. The absolute luminance in $cd.m^{-2}$ obtained with each model, using the same color scale as in Fig. 7 in [16]. The measurements are interpolated using bicubic spherical interpolation before being converted to luminance values.

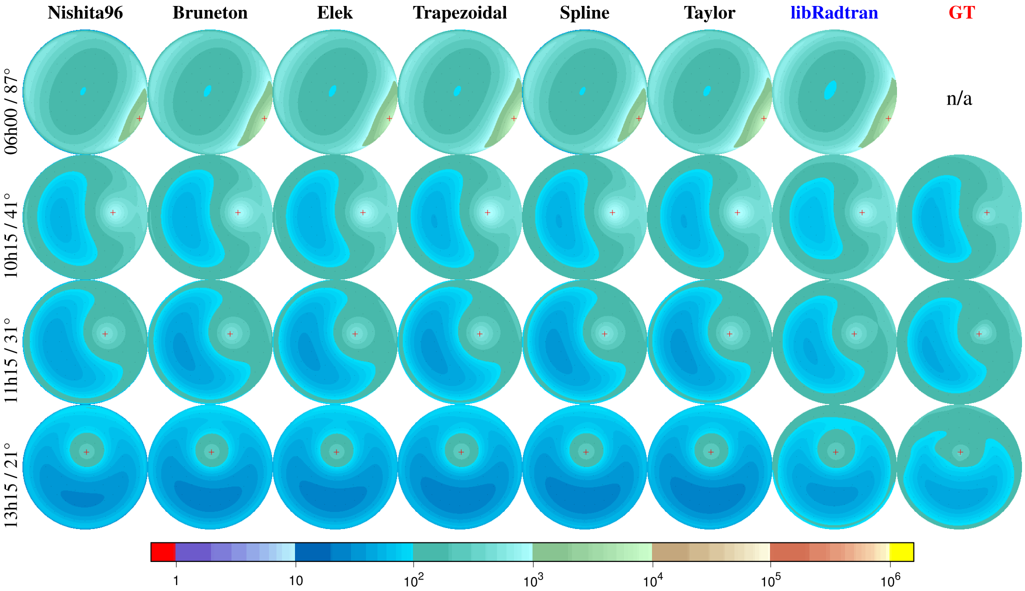


FIGURE 9. Relative luminance. The luminance obtained with each model, in percentage of the zenith luminance, and using the same color scale as in Fig. 6 in [16]. The measurements are interpolated using bicubic spherical interpolation before being converted to luminance values.

bicubic spherical interpolation prior to being converted to luminance values.

- Relative luminance figures (Figure 9) – were obtained for each numerical integration method, in percentage of the zenith luminance. The same color scale was used as in Figure 6 in [16]. The measurements were interpolated using bicubic spherical interpolation prior to being converted to luminance values.
- Chromaticity figures (Figure 10) – present the rg - chromaticity, $(r, g, b)/\max(r, g, b)$, that was computed for each method. The measurements were interpolated using bicubic spherical interpolation before being converted to chromaticity values.
- Relative error figures (Figure 12) – present the relative error that was computed at the 81 sampling points (and summed equally over the common range supported by

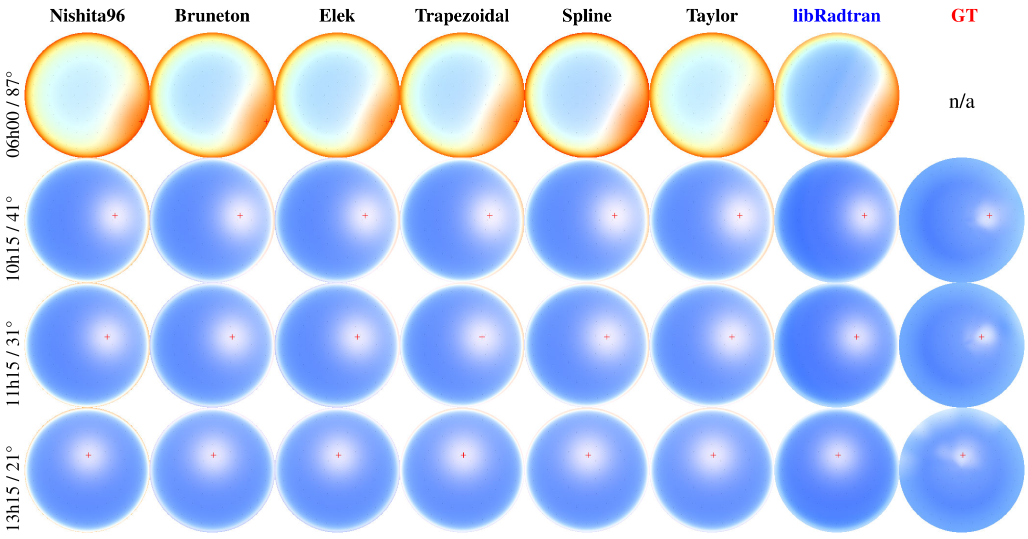


FIGURE 10. Chromaticity. The rg-chromaticity, $(r, g, b)/\max(r, g, b)$, obtained with each model. The measurements are interpolated using bicubic spherical interpolation before being converted to chromaticity values.

all methods, i.e. between 360 and 720 nm), and then interpolated with spherical bicubic interpolation.

TABLE 1. Grid search with respect to the lowest value of height and distance points that don't affect the final quality of the image. Number of samples along the viewing ray was constant and was equal to 16 during the grid search.

Time [s]	RMSE	ΔE	Height points	Distance points
115.83	30.1	15.61	50	20
73.84	30.1	15.64	20	20
75.3	30.1	15.65	20	10
73.37	30.3	15.73	20	5
61.17	31.2	16.22	10	10
68.86	30.6	15.85	15	5
66.41	55.9	34.72	15	2

We also present a qualitative comparison of the all considered methods (Table 4) and exemplary renders produced by each method (Figure 13).

VI. RESULTS AND DISCUSSION

A. THE CHOICE OF THE HYPER PARAMETERS

To be able to perform the best quality renders we had to do preliminary research to obtain the hyper parameters of the Splines and Taylor methods. To do so, we performed a grid search for both methods.

1) THE SPLINES METHOD GRID SEARCH

For the Splines method we had 4 hyper parameters that needed to be selected, namely they were: the grid size (height points and distance points), the number of samples along the viewing ray ($\vec{V}$) and the splines' type.

First, we checked for what are the lowest values of height and distance points that don't affect the image quality. To do so, for the purposes of the grid search, we assumed the constant value for the number of samples along the viewing ray to be 16 (see Table 1).

TABLE 2. Comparison between different types of splines in terms of time and quality of the image. The grid size is equal to 20 × 20, the number of samples along the viewing ray is equal to 16.

Spline type	Time [s]	RMSE	ΔE
cubic	70.1	30.1	15.64
akima	69.8	30.1	15.63
catmull-rom	70.8	30.1	15.63
linear	69.07	30.5	15.85
monotone hermite	71.65	30.3	15.74
hermite	71.84	30.1	15.64

Analysing the snippet of the grid search results in Table 1 we can see that the best performing grid sizes are:

- 50 × 20 - highest quality, the lowest performance,
- 20 × 20 or 20 × 10 - highest performance with the acceptable drop in the image quality.

In the following experiments we decided to use a grid size of 50 height and 20 distance points, as we were focused on achieving the highest render quality. We did the same in choosing the value of the number of samples along the viewing ray. In this case 1024 numbers of samples was used, since the higher the number of samples along the viewing ray, the better is the obtained quality.

What is more, we also analysed the influence of different types of splines on the time performance and the image quality of the Splines method (see Table 2). Looking at the results in Table 2 we can observe that we have several winning candidates, namely: cubic, akima, catmull-rom and hermite. We chose hermite splines in our implementation as they are widely known types of splines and provide a compromise between time performance and quality metrics, especially

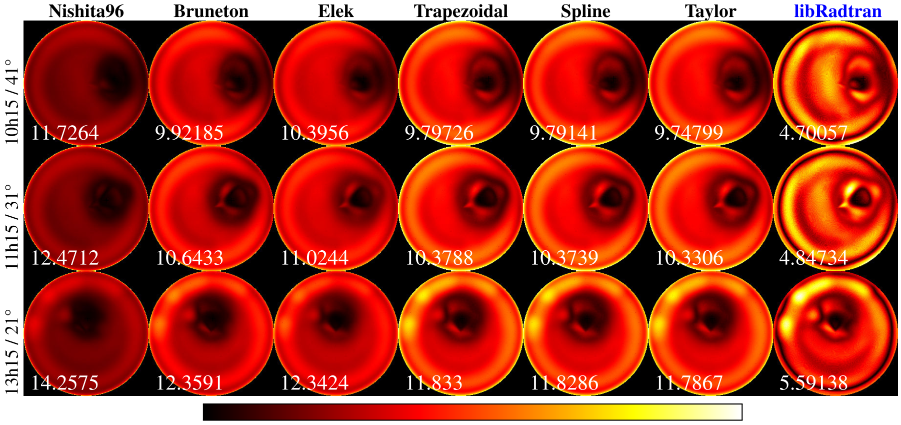


FIGURE 11. Delta E error. The Delta E difference between measurements (GT) and the selected methods (per pixel), computed in CIE L*a*b* space.

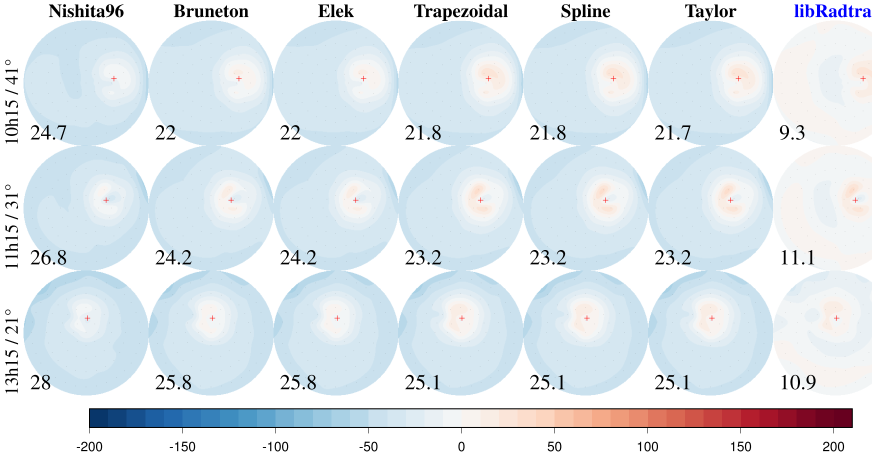


FIGURE 12. Relative error. The relative error compared with the measurements (GT), in % and using the same color scale as in [26], computed at the 81 sampling points (and summed equally over the common range supported by all models, i.e. between 360 and 720 *nm*), and then interpolated with spherical bicubic interpolation. The bottom left number is the RMSE in $mW/(m^2.sr.nm)$ (computed over the 81 sampling points and all the wavelengths between 360 and 720 *nm*).

because ΔE differences between hermite and akima splines are negligible.

2) THE TAYLOR METHOD GRID SEARCH

To choose the best values for a, b parameters for the Taylor method we performed a grid search. We tested for both parameters set $\{0.00, 0.01, 0.02, \ldots, 1.00\}$. The computing error between the Ground Truth (GT) from Bruneton's framework [1] and the approximate Formula 17 proposed above, was made for a certain number of parameter $d \in [0, R]$ and a certain number of parameter $t \in [t_{min}(d), t_{max}(d)]$. We chose a pair of parameters for which the cumulative error was the lowest, in our case it was $a = 0.45$, $b = 0.45$.

B. COMPARISON WITH THE TRAPEZOIDAL RULE

As our methods are based on pure numerical methods, we decided to also compare the Splines and Taylor methods with the pure Trapezoidal Rule (see Table 3).

We can see that the Taylor method provides both the shortest computational time and the highest image quality in terms of RMSE and ΔE metrics. When it comes to the Splines method, it is on average two times slower than the Trapezoidal Rule and much slower than the Taylor method (12.5 times slower in the worst case, 7 times slower in the best case). However, its potential is noticeable when the number of samples along the viewing ray is small (64 is the threshold value in Table 3). In this case, the Splines method outperforms the Trapezoidal Rule in terms of image quality.

C. MEMORY CONSUMPTION DISCUSSION

When it comes to comparing the memory requirements of the proposed methods in comparison to the state-of-the-art method [1] it is worth noting that the Bruneton method uses several lookup tables that take some additional space (according to [5] 50MB of additional memory space is required to store the set of lookup tables for one planet). On the

TABLE 3. Time, RMSE and ΔE comparison between the chosen methods. Time is a cumulative time for calculating all the data using Bruneton's framework [1]. Splines method used Hermite splines with 50 height and 20 distance points.

Method name	Time [s]	RMSE	ΔE	No. view samples	No. light samples
Trapezoidal	3,586.4	23.6	10.68	1,024	128
	2,655.78	23.6	10.7	768	128
	1,769.27	23.7	10.76	512	128
	892.22	23.9	10.92	256	128
	448.3	24.3	11.22	128	128
	230.03	25.1	11.76	64	128
	118.43	26.9	12.83	32	128
	62.85	33.4	15.74	16	128
	34.74	54.4	23.31	8	128
Taylor	539.48	23.5	10.67	1,024	1
	405.71	23.6	10.7	768	1
	274.08	23.7	10.76	512	1
	141.36	23.9	10.94	256	1
	73.17	24.3	11.29	128	1
	38.23	25.2	12.01	64	1
	21.1	26.7	13.35	32	1
	12.63	30.1	15.61	16	1
	8.56	38.6	20.28	8	1
Splines	6,805.25	23.6	10.68	1,024	1
	5,125.21	23.6	10.71	768	1
	3,403.17	23.7	10.76	512	1
	1,711.71	23.9	10.94	256	1
	866.91	24.3	11.3	128	1
	438.11	25.2	12.02	64	1
	222.45	26.7	13.35	32	1
	115.83	30.1	15.61	16	1
	59.96	38.6	20.28	8	1

other hand, both of the proposed methods, Splines-based and Taylor-based, do not need to store intermediate results in lookup tables. Therefore, we do not need to precompute additional lookup tables and thus save the memory space.

D. QUANTITATIVE COMPARISON WITH THE STATE-OF-THE-ART METHODS

We also compared the proposed methods with other widely known methods (Nishita93 [10], Bruneton [5], Elek [6], the Trapezoidal Rule, libRadtran [27]) taking into account renders of spectral radiance (Figure 7), absolute (Figure 8) and relative luminance (Figure 9), rg-chromaticity (Figure 10), ΔE (Figure 11) and RMSE (Figure 12) metrics.

Looking at Figure 11 we can see that apart from the libRadtran method, we can observe that the most perceptually correct results are obtained using the Taylor and Spline methods respectively. They outperform others by approximately 3%, in terms of the ΔE metric (Figure 11), the state-of-the art method [5] - so far the best solution in the literature.

On the other hand, when we look at the RMSE values (Figure 12) we observe that the Taylor and Spline methods are still better than [5], but by approximately 2.1% on average.

E. QUALITATIVE COMPARISON WITH THE STATE-OF-THE-ART METHODS

Considering the qualitative comparison of the state-of-the-art methods and the proposed ones (see Table 4) we can notice two things. First of all, the average (i.e. the mean value computed across the renders during different daytimes and sun positions) RMSE and ΔE values are the lowest for the Taylor and Spline methods. They are lower than Bruneton's method by approximately 2.9% (the Taylor method) and 2.5% (the Splines method) respectively.

Both presented methods offer the ability to render the atmosphere from all the viewpoints (from the outside and inside the atmosphere), compute the aerial perspective and support sunset/sunrise effect. What is more, the Taylor and Spline methods do not need the additional memory space, hence the memory complexity of the precomputations is *n/a*.

On the other hand, Bruneton's method has the advantage over the proposed methods in terms of time performance (render time in Table 4). Since his method is a lookup table-based method, during the rendering time he only needs to fetch several values from the lookup tables to compute the value of the pixel both for the transmittance and single scattering terms. The Taylor method has the time complexity equal to $O(n)$ because during the rendering time we still need to evaluate single scattering term using the Trapezoidal Rule, but we are able to compute the transmittance term using the analytical formula. The Splines-based method has the worst time complexity due to the computational overhead during the rendering phase which is the evaluation of the spline grid.

Besides, the Splines method is the slowest one, it has the advantage of flexibility when compared to the Bruneton and Taylor methods. If one would like to render several different planets (with different planet and atmosphere radii), they

TABLE 4. Qualitative comparison. Summary of the qualitative evaluation of the 8 clear sky models. The precomputation time and memory complexity for the Nishita96 model is for a single sun zenith angle θ_s only–the Bruneton and Elek models precomputations are for *n* such angles.

Model	Supported viewpoints	Aerial perspective	Sunset sunrise	Precompute time	Precompute memory	Render time	RMSE $mW/(m^2.sr.nm)$	Delta E error
Nishita96	in atmosphere	yes	yes	$O(n^3)$	$O(n^3)$	$O(n)$	26.6	12.736
Bruneton	all	yes	yes	$O(n^6)$	$O(n^4)$	$O(1)$	24.2	10.943
Elek	all	yes	yes	$O(n^6)$	$O(n^4)$	$O(1)$	24.2	11.256
Trapezoidal	all	yes	yes	n/a	n/a	$O(n^2)$	23.6	10.676
Spline	all	yes	yes	n/a	n/a	$O(n^3)$	23.6	10.671
Taylor	all	yes	yes	n/a	n/a	$O(n)$	23.5	10.629

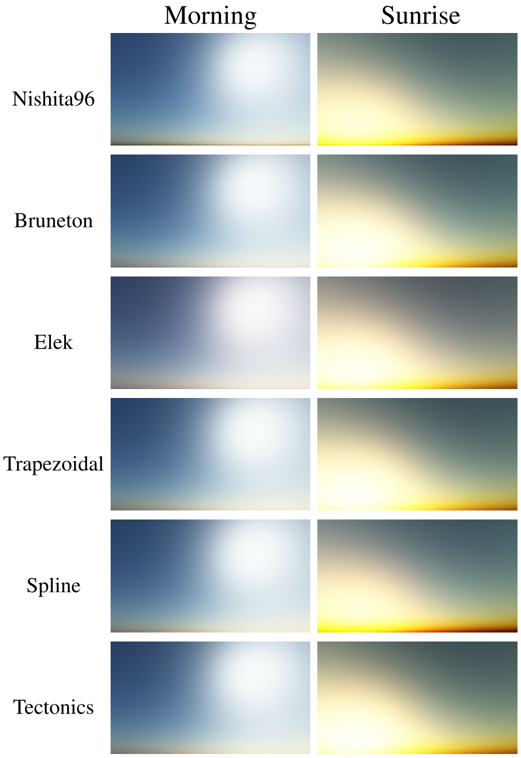


FIGURE 13. Exemplary rendered viewpoints from the ground level. The images were generated for each model for different daytimes (i.e. late afternoon - top, morning - bottom).

will need to compute several different lookup tables when using the Bruneton method or perform several different grid searches for the Taylor method. In case of the Splines method, the user does not need to do anything other than run the Splines method with different planets' parameters.

In Figure 13 we can see the end results of each method during the rendering of the clear sky in the late afternoon and in the morning.

VII. CONCLUSION

We have presented two methods to approximate the transmittance formula which results in an efficient single-scattered sky rendering. The Splines-based method gives very good results in terms of quality of the image and outperforms the state-of-the-art Bruneton method. Moreover, it provides flexibility when one would like to render several different planets without the need of precomputing several different lookup tables [1] or performing several different grid searches of the hyperparameters (the Taylor method). The Splines method seems to be a decent candidate for the movie industry.

On the other hand, the Taylor based method is a very good candidate to be implemented in a real time software like games. It can be computed quickly and gives slightly better results than the Splines method. We believe that Taylor-based method could be merged with the Bruneton method to reduce the memory consumption of the Bruneton method even further and to enhance its quality (i.e. the quality of the final images).

We have high hopes that this research may lead to the invention of new methods to approximate atmospheric scattering integrals with real-time performance.

REFERENCES

[1] E. Bruneton, "A qualitative and quantitative evaluation of 8 clear sky models," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 12, pp. 2641–2655, Dec. 2017.

[2] K. Fornalczyk, P. Napieralski, D. Szajerman, A. Wojciechowski, P. Sztoch, and J. Wawrzyniak, "Stereoscopic image perception quality factors," in *Proc. 22nd Int. Conf. Mixed Design Integr. Circuits Syst. (MIXDES)*, Jun. 2015, pp. 129–133.

[3] A. Wojciechowski, "Camera navigation support in a virtual environment," *Bull. Polish Acad. Sci., Tech. Sci.*, vol. 61, no. 4, pp. 871–884, Dec. 2013.

[4] L. Di Trapani and T. Inanc, "NTGsim: A graphical user interface and a 3D simulator for nonlinear trajectory generation methodology," *Int. J. Appl. Math. Comput. Sci.*, vol. 20, no. 2, pp. 305–316, Jun. 2010.

[5] E. Bruneton and F. Neyret, "Precomputed atmospheric scattering," in *Computer Graphics Forum*, vol. 27. Hoboken, NJ, USA: Wiley, 2008, pp. 1079–1086.

[6] O. Elek, "Rendering parametrizable planetary atmospheres with multiple scattering in real-time," in *Proc. Central Eur. Seminar Comput. Graph.*, 2009, pp. 1–8.

[7] S. O'Neil, "Accurate atmospheric scattering," *GPU Gems*, vol. 2, pp. 253–268, Dec. 2005.

[8] C. Canuto and A. Tabacco, *Mathematical Analysis I*, vol. 1. Milano, Germany: Springer, 2008.

[9] P. Gibbs, "Why is the sky blue?" *Original Usenet Phys. FAQ.*, vol. 3, p. 2015, Oct. 1997.

[10] T. Nishita, T. Sirai, K. Tadamura, and E. Nakamae, "Display of the Earth taking into account atmospheric scattering," in *Proc. 20th Annu. Conf. Comput. Graph. Interact. Techn.*, 1993, pp. 175–182.

[11] R. Stair, R. G. Johnston, and T. C. Bagg, "Spectral distribution of energy from the sun," *J. Res. Nat. Bur. Standards*, vol. 53, no. 2, pp. 113–119, 1954.

[12] J. R. Frisvad, N. J. Christensen, and H. W. Jensen, "Computing the scattering properties of participating media using lorenz-mie theory," *ACM Trans. Graph.*, vol. 26, no. 3, p. 60, Jul. 2007.

[13] W. Jarosz, *Efficient Monte Carlo methods for Light Transportation Scattering Media*. Princeton, NJ, USA: Citeseer, 2008.

[14] T. Nishita, Y. Dobashi, K. Kaneda, and H. Yamashita, "Display method of the sky color taking into account multiple scattering," in *Proc. Pacific Graph.*, vol. 96, 1996, pp. 117–132.

[15] A. J. Preetham, P. Shirley, and B. Smits, "A practical analytic model for daylight," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn.*, 1999, pp. 91–100.

[16] G. Zotti, A. Wilkie, and W. Purgathofer, "A critical review of the preetham skylight model," in *Proc. Int. Conf. Central Eur. Comput. Graph., Short Commun. Papers*, 2007, pp. 1–5.

[17] L. Hosek and A. Wilkie, "An analytic model for full spectral sky-dome radiance," *ACM Trans. Graph.*, vol. 31, no. 4, p. 95, 2012.

[18] W. Andrzejewski, A. Gramacki, and J. Gramacki, "Graphics processing units in acceleration of bandwidth selection for kernel density estimation," *Int. J. Appl. Math. Comput. Sci.*, vol. 23, no. 4, pp. 869–885, 2013.

[19] P. Lipiński, "On domain selection for additive, blind image watermarking," *Bull. Polish Acad. Sci., Tech. Sci.*, vol. 60, no. 2, pp. 317–321, 2012.

[20] D. Puchala, "Approximating the KLT by maximizing the sum of fourth-order moments," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 193–196, Mar. 2013.

[21] J. Haber, M. Magnor, and H.-P. Seidel, "Physically-based simulation of twilight phenomena," *ACM Trans. Graph.*, vol. 24, no. 4, pp. 1353–1373, 2005.

[22] C. Wenzel, "Real-time atmospheric effects in games," in *Proc. ACM SIGGRAPH*, 2006, pp. 113–128.

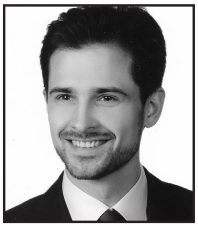
[23] T. Schafhitzel, M. Falk, and T. Ertl, "Real-time rendering of planets with atmospheres," in *Proc. Int. Conf. Central Eur. Comput. Graph., Visualizat. Comput. Vis.*, 2007.

[24] E. Yusov, "High performance outdoor light scattering using epipolar sampling," *GPU Pro*, vol. 5, pp. 101–126, May 2014.

[25] A. R. Robertson, "The CIE 1976 color-difference formulae," *Color Res. Appl.*, vol. 2, no. 1, pp. 7–11, 1977.

[26] J. T. Kider, Jr., D. Knowlton, J. Newlin, Y. K. Li, and D. P. Greenberg, "A framework for the experimental comparison of solar and skydome illumination," *ACM Trans. Graph.*, vol. 33, no. 6, p. 180, 2014.

[27] B. Mayer and A. Kylling, "Technical note: The libRadtran software package for radiative transfer calculations–description and examples of use," *Atmos. Chem. Phys.*, vol. 5, no. 7, pp. 1855–1877, Jul. 2005.



TOMASZ GAŁAJ is currently pursuing the Ph.D. degree with the Faculty of Technical Physics, Information Technology and Applied Mathematics, Institute of Information Technology, Lodz University of Technology. He is also an author of the real-time rendering class blog and vertex engine for proof-of-concept applications. His research interests include high-performance computer graphics and animation, computer games, and computer simulations.

FILIP PIETRUSIAK received the master's degree from the Faculty of Technical Physics, Information Technology and Applied Mathematics, Institute of Information Technology, Lodz University of Technology, where he is currently pursuing the Ph.D. degree.

MAREK GALEWSKI received the M.Sc., Ph.D., and D.Sc. degrees in mathematics from the Faculty of Mathematics and Computer Science, University of Lodz Technology, in 1997, 2002, and 2009, respectively. His research interests include variational methods for both continuous and discrete boundary value problems, nonlinear optimization, and the theory of nonlinear abstract equations.

RAFAŁ LEDZION received the M.Sc. degree in physics from the Faculty of Physics and Chemistry, University of Lodz, in 1997, and the Ph.D. degree in physical sciences (specialization in solid state physics) from the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology, in 2002. His research interests include solid-state physics, linear and nonlinear optics, light modulation issues, and electro-optical effect in liquids and solids.

ADAM WOJCIECHOWSKI received the M.Sc. degree in computer science from the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology, in 1999, the Ph.D. degree in computer science, (specialization in computer graphics and vision) from the Faculty of Automatics, Electronics and Computer Science, Silesian University of Technology, in 2005, and the D.Sc. degree in computer science from the Faculty of Mechanical Engineering and Computer Science, Czestochowa University of Technology, in 2014. His research interests include human–computer interaction, computer graphics, machine learning, computer vision, computer games, virtual environments, and computer simulations.