Atmospheric Scattering - State of the Art

Diogo A. R. Lopes António Ramires Fernandes
Departamento de Informática
Universidade do Minho, Portugal
pg22830@alunos.uminho.pt, arf@di.uminho.pt

Abstract

Atmospheric scattering is the natural phenomenon mainly responsible for the colours we observe in the sky. Over the years, several realistic computer graphics algorithms have been proposed in order to reproduce these colours. This state of the art is motivated by the large amount of scattered information, and by its great potential usage in a wide range of applications like flight simulators, video games and movies. This paper will cover the most important models and will present their evolution over the years. The first part contains a small introduction to the mechanics behind the light scattering phenomena. The second part will cover the earlier models, very much focused in the physical phenomenon. The third section will cover GPU based models, more focused on performance.

Keywords

Modelling of Natural Phenomena, Atmospheric Scattering, GPU Graphics

1. INTRODUCTION

Atmospheric scattering is a natural phenomenon that can be described as the result of the interaction between sunlight and particles in the atmosphere. When we look to the sky during the day we see mostly blue, while at sunset we get a more reddish colour, specially near the horizon. Lord Rayleigh, in 1871, and Gustav Mie, in 1908, have described the behaviour of the light-particles interaction that produces these colours([Strutt 71, Mie 08]).

Rayleigh scattering describes the behaviour of light when interacting with very small particles (most of the particles present in the atmosphere). Mie scattering can be used to describe the behaviour of light interacting with any kind of particles, nevertheless, it is mostly used to describe the interaction with larger particles such as haze.

The core of the atmospheric scattering computational models is the method to solve the scattering equations of Mie and Rayleigh.

Earlier models like [Klassen 87] and [Nishita 93] used ray tracing algorithms in order to solve those equations. Latter models tended to use analytical expressions to reduce the computation complexity in the equations. Examples of this approach are the models presented in [Preetham 99] and [O'Neal 05]. The most recent models, such as [Schafhitzel 07] and [Bruneton 08], resort to pre-computation of the light scattering values in order to achieve better performances.

2. ATMOSPHERIC SCATTERING

In daylight models, the sun is considered the only light source. Due to the distance between the sun and the earth, it is assumed that the light rays travel parallel to each other, and unmolested through the empty space until they enter the earth's atmosphere.

Upon entering the atmosphere, these rays will interact with particles in the atmosphere, the most relevant interaction, regarding colour, being scattering. Scattering occurs when a photon's electromagnetic field hits a particle's electric field in the atmosphere and is deflected into another direction [Strutt 71].

Most models consider two types of particles: air molecules and aerosol particles, the latter being much larger than the former. Air molecules can be found everywhere in the atmosphere where aerosols are more common closer to earth's surface. The result of the interaction is a function of the photon's wavelength and the size of the particle.

To better illustrate the light scattering phenomenon, lets follow a ray's path since it enters the atmosphere until it reaches the camera, `ray` R_1 in `figure` 1.

A ray R_1, a set of photons travelling along a common trajectory, travels from the sun and reaches the top of the atmosphere (P_{sun1}) with a certain intensity I_0. Following the same direction we see that the ray interacts by the first time with a particle at point P_1.

This interaction has the potential to deflect the ray's photons in all directions. Only part of the photons will maintain the initial direction. Each time the ray interacts with a particle it gets further attenuation.

Still considering the initial direction, the ray further interacts with other particles until it reaches P_{scatt}, and the intensity of the ray upon reaching P_{scatt} can be computed

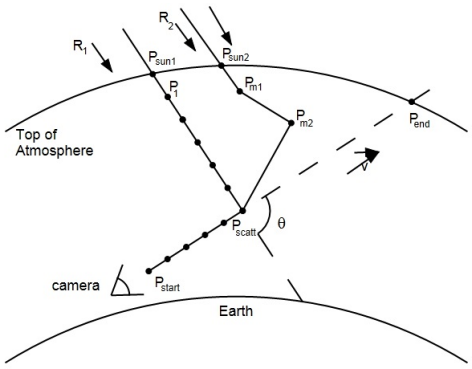

Figure 1. Evaluation of the intensity of a ray along its path

as $I_2 = I_1 att(p_A)$, where p_A is the path from P_sun1 to P_{scatt}. The value of $att()$ represents the light attenuation, or optical depth, which can be translated as the percentage of photons that will follow the same direction.

Consider that at P_{scatt} some of the photons are scattered towards the camera. So R_1 will leave P_{scatt} with an intensity value of I_{scatt} that is given by $I_{scatt} = I_2 phase(\theta)$, where $phase(\theta)$ determines the percentage of the photons that will travel in a direction with an angle θ (`figure` 1), relative to to original direction. The process of deflecting the light towards a given direction being considered, in this case the direction towards the camera, is also called in-scattering.

Between P_{scatt} and the camera location the ray will suffer further attenuation as it interacts with particles on the way, hence, the ray will reach the camera with an intensity $I_{final} = I_{scatt} att(p_B)$, where p_B is the path between P_{scatt} and the camera. The loss of intensity by the deflection of photons out of the original direction is called out-scattering.

The total amount of intensity reaching the camera in a direction v can be computed as the sum of all the in-scattering contributions caused by rays, parallel to R_1, that interact with particles in the path from the camera to the exit point of the atmosphere along v.

2.1. Computing the Attenuation

Computing the attenuation requires knowing the particle density. In general, according to [Nishita 93], the density for a particular height h can be computed as $\rho(h) = exp(-h/H_{den})$, where H_{den} is a reference height, representing the atmosphere thickness where the particles can be found. For example, in [Nishita 93], for air molecules $H_{den} = 7994m$ and for aerosol particles $H_{den} = 1200m$. We shall refer to the density of air molecules as ρ_M and the density of aerosols as ρ_A. This reflects the fact that both types of particles have an exponential distribution along the atmosphere being more dense near the surface and the concentration of aerosols is only meaningful much closer to the surface.

The size of the particles also affects the interaction with different wavelengths. Attenuation is therefore a function of both the wavelength, λ, and particle density, ρ. Equation 1 shows how to compute the attenuation for particles (of type T, where T can be A or M) of a ray after travelling in path s.

$$att_T(s, \lambda) = exp(-\frac{4\pi K}{\lambda^4} \int^s \rho_T(p)dp) \qquad (1)$$

where K is a constant for the standard atmosphere (molecular density at sea level).

2.2. Computing the Phase Function

The phase function determines the amount of light that is scattered in a certain direction. Gustav Mie presented a phase function that works for all particle sizes, but it is too computationally expensive. Rayleigh presented an approximation that works for particles with small sizes (10 times inferior to the photon wavelength) like air molecules (`equation` 2).

$$phase_M(\theta) = \frac{1}{4\pi}\frac{3}{4}(1 + cos^2(\theta)) \qquad (2)$$

Graphically the function behaves as depicted in `figure` 2.

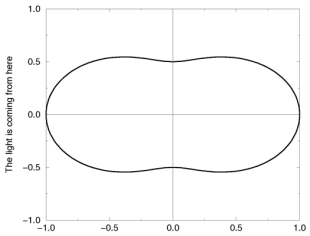

Figure 2. Rayleigh scattering phase function

Some models like [Nishita 93] proposed to use the Henyey-Greenstein phase function as an approximation to Mie's phase function for larger particles. (`equation` 3).

$$phase_A(\theta) = \frac{1}{4\pi}\frac{1 - g^2}{(1 + g^2 - 2gcos(\theta))^{3/2}} \qquad (3)$$

Graphically the function behaves as depicted in `figure` 3.

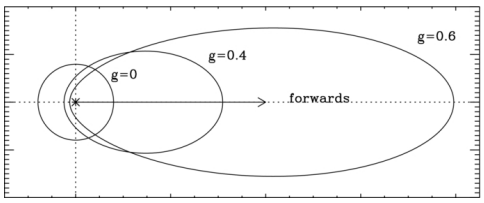

Figure 3. Rayleigh scattering phase function

The difference of these functions results can be better seen in `figure` 4. This figure helps to explain the behaviour of the light-particle interactions (smaller particles deflect particles more evenly than larger particles).

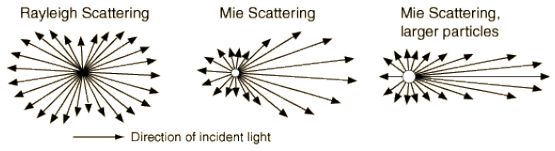

Figure 4. Types of scattering and description of its behaviour

2.3. Putting it all Together

The equation to compute the light intensity in a single wavelength λ that reaches the camera located at P_{start}, along a path s, and considering only a type of particle (A or M) is given by:

$$I_T(\lambda, s, P_{start}) = \int_{P_{start}}^{P_{end}} I_0(\lambda) att_{Tsun}(p) phase_T(\theta) att_{Tview}(p) dp \quad (4)$$

where $att_{Tsun}(p)$ is the attenuation, for particles of type T, between the entrance in the atmosphere until point p in path s, and $att_{Tview}(p)$ is the attenuation between p and the camera position.

When considering the result from both types of particles, the final intensity can be computed as:

$$I(\lambda, s, P_{start}) = I_A(\lambda, s, P_{start}) + I_M(\lambda, s, P_{start}) \quad (5)$$

The description above only considers single scattering events, i.e., the ray suffered only one event of in-scattering towards the camera. The full in-scattering value of light intensity must also take into account the scattering of light from rays that have been scattered multiple times. An example appears in `figure` 1 where `ray` R_2 is scattered multiple times until it is finally scattered into the viewing direction at point P_{scatt}. The computation of the final intensity of R_2 that reaches the camera is similar to R_1, however, all rays reaching the atmosphere must be taken into account, as well as all scattering directions in every light-particle interaction, thus making this process highly computationally intensive.

To take into account multiple scattering up to n scattering events the term att_{sun}, in `equation` 4, needs to be replaced by att_{sunMS} (`equation` 6).

$$att_{sunMS}(n) = \sum_{i=0}^{n} \prod_{j=1}^{j=i} att(path(j)) phase(\theta_j) \quad (6)$$

where $path(j)$ is the path travelled by the ray between two consecutive interactions, $j-1$ and j, and $phase(\theta_j$ is the angle between the direction of the ray prior to interaction j and the direction after interaction j.

In the next sections the mechanisms implemented by the various models in order to compute approximations to the scattering equations will be presented together with some resulting images.

All images were obtained by placing the camera in a latitude of 40° and longitude of 0° at $05:00$, $14:00$ and $20:00$ hours.

3. EARLIEST MODELS

The earlyest models attempted to solve the scattering equations by using ray-tracing algorithms and implemented a sampling strategy to solve the integrals presented in section 2.

3.1. Klassen Simple Layer Atmosphere Model

In 1987, Victor Klassen ([Klassen 87]) proposed a model of the atmosphere composed by 2 layers, each with constant density. Furthermore, the top layer, is composed only by air molecules (haze free layer) and the bottom layer, closest to the Earth's surface, is composed by both particle types (haze and air molecules). Klassen's model considers only single scattering and assumes that it is very unlikely that a in-scattering event will occur in the top layer.

Since in-scattering events occur only in the haze layer, in the top layer the attenuation is proportional to the distance covered by the rays in that layer. Hence, when entering the haze layer the intensity is given by `equation` 7.

$$I_1 = att(s_t, D_t) I_0 \quad (7)$$

where I_0 is the intensity before entering the atmosphere, s_t is the path covered by the ray until it reaches the bottom layer, and D_t is the density of the top layer.

Then the intensity at the in-scattering point P_{scatt} can be computed as the product of I_1 and the attenuation between the entrance point on the haze layer, and the intersection with the view direction, see `equation` 8.

$$I_{P_{scatt}} = att(s_h, D_h) I_1 \quad (8)$$

where s_h is the path covered by the ray in the haze layer until it reaches the view direction, and D_h is the density of the haze layer.

From P_{scatt} to the camera the light travels through path s. The full amount of light in the view direction is obtained by sampling the view direction, between the camera and the exit point of the haze layer, and compute the attenuation of all rays that are in-scattered towards the camera. The final equation, for a particular type of particle, is:

$$I_{final} = \sum_{i=0}^{n} att_{view}(s_i, D_h) phase(\theta) att(s_{hi}, D_h) att(s_{ti}, D_t) I_0 \quad (9)$$

where $att_{view}(s_i, D_h)$ is the attenuation between the camera and sample i along path s, and $theta$ is the angle between the view direction and the sun rays.

The attenuation itself is greatly simplified due to the assumptions of constant density within each layer as shown in `equation` 10.

$$att(s, D) = exp(-\frac{4\pi K}{\lambda^4} length(s) D) \tag{10}$$

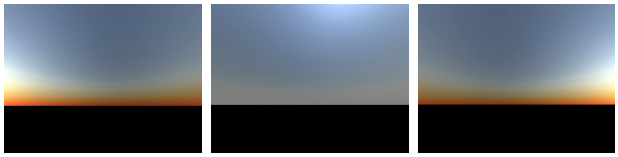

Figure 5. Images using Klassen model of the atmosphere with 2 layers. Sunrise, Midday and Sunset, respectively

The results (see Figure 5) were obtained by using the scattering coefficient values present in [Nishita 93], [Bucholt 95] and [Preetham 99], namely $D_{haze} = D_{top} = 2$, $K = 3e - 32$. Eight samples were taken in each viewing direction, and only three wavelengths were considered: red, green and blue.

Klassen's model can reproduce various types of day, like foggy days or days with clear sky by adjusting the particle density and the height of the haze layer.

3.2. Multiple Layer Atmosphere Models

Later, [Kaneda 91] implemented single scattering with a continuous atmosphere as described in section 2. This work uses Rayleigh's scattering equation for air molecules and Henyey-Greenstein's scattering equation for aerosols. With a non-uniform atmosphere density, it needs to evaluate the density of air molecules and haze particles along the view direction resulting in a closer approximation to the attenuation equation (`equation` 1).

Optical depth integrals are approximated by sampling the ray. Where in Klassen's model the ray is only sampled from the view direction ($P_{cam} \rightarrow P_{end}$), this model extends the sampling to the full path of the ray since it enters the atmosphere ($P_{sun1} \rightarrow P_{cam}$) (see `figure` 1). This model is more physically accurate than the one proposed by Klassen. On the other hand it requires a much larger computational effort.

In [Nishita 93] several performance improvements are presented. The two major computations, performance wise, are the density and the attenuation. Regarding the density, multiple spherical shells are created such that between each two consecutive shells the density difference is constant. This implies that the radius of the shells varies exponentially, resulting on a larger concentration of shells near the surface. As that the sun rays are parallel, consider a cylinder with its axis in the line that connects the centre of the earth to the sun, and a particular radius. Then all the points in the line resulting from the intersection of a shell with the cylinder (a circle) will have the same attenuation. Consider now a set of cylinders with radius varying inversely proportional to the length of the intersection. A 2D table was built such that for each entry (i,j) the cell contains the attenuation for points in the intersection of shell i with cylinder j. To compute the attenuation for a particular point P, interpolation is used based on the altitude of the point and the distance to the centre of the earth of its projection on a plane perpendicular to the axis of the cylinders. The altitude determines the smallest outer shell that contains P, consider that this is shell i. The projection determines the smallest radius of the cylinder that contains P, assume that this is cylinder j. To determine the value of the attenuation for P linear interpolation of the closest grid locations (i,j), (i,j-1), (i-1,j) and (i-1,j-1) are used. This approach allows for a significant reduction of the required computations when compared to [Kaneda 91]. When comparing to Klassen's, we get a physically more accurate model, although performance wise it is still slower despite of the pre-computations performed.

Results of using the model by Nishita et al. can produce results like the snapshots in figure 6:

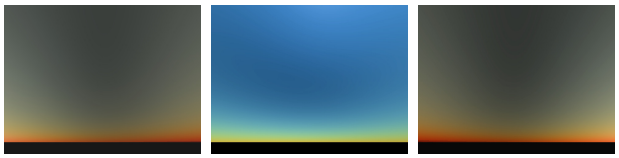

Figure 6. Images using Nishita model of the atmosphere in clear sky, Sunrise, Midday and Sunset respectively

These results were obtained by using coefficients like air and aerosol particles density, present in [Nishita 93], [Bucholt 95] and [Nishita 96], which were mentioned above in `section` 3.1. The g coeficient of Henyey-Greenstein phase function used was $g = 0.99$.

3.3. Nishita Multiple Scattering Model

The early models only considered single scattering events, but that changed in 1996 when Nishita et al. [Nishita 96] proposed to compute the colours of the sky by considering the contribution of multiple scattering, thus providing a better fit to the theoretical model.

For that purpose the atmosphere is divided in voxels and intensity values are stored at the vertices of the grid.

Initially, prior to runtime, the intensity reaching each voxel is computed based only on the optical depth, i.e. the direct path the rays traverse from entering the atmosphere until reaching the voxel. Furthermore, for each voxel, the intensity contributions from other voxels are stored separately. This enables second order scattering. The process can be performed recursively to achieve higher orders of scattering, but as noted by the authors the contributions beyond second order scattering are not very meaningful.

At run time, for each sample on the view direction, first they seek the voxel where the sample is located. Then all

the contributions of the vertices in that voxel are multiplied by the phase function and added together in a linear combination weighted by the distance from the sample to the vertices.

Since this method takes multiple scattering into account the results become more accurate, see `image` 7.

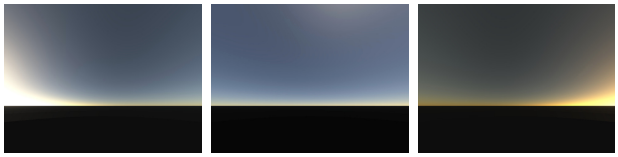

Figure 7. Images using Nishita et al. model of the atmosphere in clear sky with multiple scattering

This results were obtained by gathering 2 levels of scattering, and coefficients such as air and aerosol particles density are those presented in [Nishita 93], [Bucholt 95] and [Nishita 96], and are the same as the ones presented in the previous subsection.

3.4. Analytical Models

The main problems of the previous models is the computational time required to produce results with good quality. So, in 1995 Dobashi et al. [Dobashi 95] and Preetham et al. [Preetham 99] in 1999 proposed to reduce the number of computations necessary to obtain the colours of the sky.

To achieve this, both proposed a set of analytical functions that are capable of, given a viewing direction and the position of the sun, describe the light intensity for each visible light wavelength, and consequently, the colour of the sky in that viewing direction.

To create a set of functions capable of describing the distribution of light in the atmosphere, Dobashi et al. proposed to use a number of basis functions to express the light distribution on the atmosphere. The basis functions in [Dobashi 95] were cosine functions. The expression can be seen as the product of two terms where the first expresses the light distribution all over the atmosphere ($\Psi_i(v)$), and the second expresses the weight ($W_i^{(\lambda)}(\phi_{sun})$) that each cosine function has for a particular position in the atmosphere and for a particular viewing direction (`equation` 11).

$$L_\lambda(v, \phi_{sun}) = \sum_{i=0}^{N} W_i^{(\lambda)}(\phi_{sun})\Psi_i(v) \qquad (11)$$

These functions parameters and specifications are modelled by approximating their results with the ones produced by high quality ray tracing models, i.e., Dobashi et al. used the AS model of [Kaneda 91] to compute the values of light intensity for a series of configurations, then the functions were modelled according to the results obtained. The usage of this method reduces the quality of the result but it speeds up the rendering process, because, instead of solving the scattering integrals by sampling the ray's path, analytical expressions are used.

In 1999, Preetham et al. followed the same idea as Dobashi et al. but instead of using basis functions to compute the sky colours, a different set of functions proposed by Perez et al. in 1993 [Perez 93] were used. Preetham et al. developed a method that based on measurements of light intensity distributions in different weather conditions, modelled analytic expressions that could reproduce the colour of the sky with great accuracy and speed.

For his model, Preetham et al. considers a term called turbidity that is a measure of the fraction of scattering due to haze as opposed to molecules [Preetham 99], hence, turbidity is a measure of the thickness of the atmosphere. So, a higher value of turbidity means a denser haze, generating fog, and a lower value means clear sky.

To compute the distribution of sunlight for a given direction Preetham et al. uses `equation` 12 presented by Perez et al.

$$\begin{aligned} Y &= YzF(\theta,\phi)/F(0,\theta_s); \\ F(\theta,\phi) &= (1 + Ae^{B/cos\theta})(1 + Ce^{D\phi} + Ecos^2\phi); \end{aligned} \qquad (12)$$

where $F(\theta, \phi)$ is Perez et al. model of luminance, and Yz is the value of intensity at zenith. This formula of luminance describes the sky light distribution according to the direction of incident sunlight and some factors, such as, darkening or brightening of the horizon (A), luminance gradient near the horizon (B), relative intensity of the circumsolar region (C), width of the circumsolar region (D) and relative backscattered light (E). To model these parameters, Preetham et al. used the results produced by the model of Nishita et al. with multiple scattering [Nishita 96].

With these functions computing the sunlight intensity, in order to obtain the colour of the sky, it is only necessary to compute the light attenuation in every viewing direction. For the attenuation Preetham et al. uses pre-computed values for the fraction of light that is scattered ($S(\theta, t)$), and for the "extinction" of light ($\beta(\theta, \phi, s)$). The intensity value is computed by:

$$L(\lambda, s, \theta, \phi, t) = YS(\theta, t)\beta(\theta, \phi, s, t) \qquad (13)$$

were t is the turbidity, and $\beta(\theta, \phi, s, t)$ and $S(\theta, t)$ are:

$$\beta(\theta, \phi, s, t) = e^{(-\alpha \frac{\beta_{air}(\theta)+\beta_{aero}(\theta)}{\beta_{air}+\beta_{aero}})s} \qquad (14)$$

$$S(\theta, t) = I_{air}\frac{\beta_{air}(\theta)}{\beta_{air}} + I_{aero}\frac{\beta_{aero}(\theta)}{\beta_{aero}} \qquad (15)$$

α is a decay constant, s is the distance between the camera and the object (in the case of the sky, the top of the atmosphere) and the other coefficients represent the angular coefficients or phase functions ($\beta_{air/aero}(\theta)$) and the extinction coefficients or light attenuation ($\beta_{air/aero}$).

$I_{air/aero}$ is a polynomial function and represents the amount of light that is being scattered in the full path of the view direction. This function can be solved using a system of linear equations. Due to space restrictions the

system is not presented here, however, is fully detailed in [Preetham 99].

Note that the values of $S(\theta, t)$, with the exclusion of $I_{air/aero}$, and the values of $\beta(\theta, \phi, s, t)$ are pre-computed for all range of θ and ϕ. So the heaviest part of this method is computing $I_{air/aero}$. The other values are quickly obtained by a single equation (value of Y) or by just accessing a table ($\beta(\theta, \phi, s, t)$).

The turbidity value used to obtain `figure` 8 is 2. The remaining values can be found in [Preetham 99].

Figure 8. Images using Preetham model of the atmosphere in clear sky

The work of Preetham et al. allowed to obtain the colours of the sky faster without a significant loss in quality due to the parameter fitting according to a high quality model [Nishita 96].

This work was deeply analized in [Zotti 07]. The authors compared the results in [Preetham 99] with actual measurements and the ISO/CIE 2003 [oIC04] for standard general sky luminance distributions and conclude that the results for values of turbidity below 2.0 are not correct. Furthermore, the Preetham model does not properly reproduce the noticeable darkening of the sky in the antisolar hemisphere when the sun is low, with luminance values about 2–5 times too high. Also, the brightness peak towards the sun is not as steep as it can be measured or is modelled by the CIE Clear-Sky models. As a consequence of this analysis a modified version was proposed in [Hosek 12] that claims to significantly improve the rendition of sunsets and high atmospheric turbidity setups. Additionally, this model takes into account the ground albedo. From a mathematical point of view the models are very similar, with [Hosek 12] adding more coefficients to gain more control and be able to produce more accurate results.

4. Modern Models

When the graphic units became programmable, the AS models presented started to take advantage of its properties in order to enhance the performance of some already existent AS models.

4.1. GPU technique based Models

One of the first GPU models was presented by Dobashi et al. in 2002 [Dobashi 02]. The process of computing the colours of the sky is very similar to the one proposed in [Nishita 93]. The main difference consisted in using sampling planes instead of sampling directions, and with that small difference the GPU's could optimise the computation times.

In 2004 Sean O'Neal analysed [Nishita 93] and tried to optimise its performance [O'Neal 04]. O'Neal realised that the number of required computation for a single pixel was too costly, so he proposed 2 major changes. One to improve the look-up table, another to fit some of the optical depth calculations into analytical functions.

The look-up table proposed by Nishita was able to reduce the number of calculations by half by pre-calculating the optical depth of the sunlight, and so reducing the cost of calculating the in-scattering component. O'Neal proposed an improved look-up table that also pre-computed the optical depth of the viewing ray. For that purpose two dimensions were considered, (x, y), where x represents a specific altitude from the ground and y represents a vertical angle. For each pair (x, y) a ray is fired and it collects the atmosphere density in that viewing direction. With this optimisation both $att_{sun}(s)$ and att_{view} can be obtained interpolating the values in the look-up tables. In 2005, O'Neal,

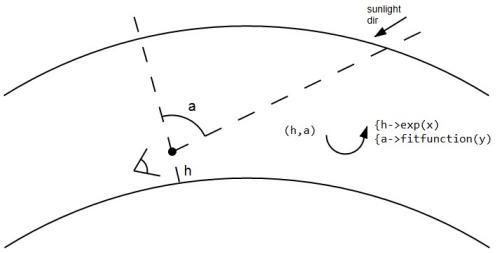

Figure 9. Look-up table elaboration by O'Neal

[O'Neal 05], went even further and tried to replace these look-up tables by analytical expressions in order to use the potential of the GPU. In order to do that, O'Neal plotted the results obtained for the various heights of x with the several angles of y and realised that the results followed an exponential distribution for the values of x with different scales in y. Knowing this, O'Neal proposed the usage of 2 analytical expressions. One that explained the exponential distribution in x, and a scale/fit function that explained the difference in scale in angle y.

To obtain the colours of the sky this method uses `equation` 4 and replaces the attenuation components of $\rho(s) = exp(x)$ and the $\int^s = scale(y)D$, where D is the length of path s. For att_{view}, y is the angle of vision from the camera, and for att_{sun}, y is the angle made between the camera and the sun.

An important aspect of this method is that as opposed to the method in [Nishita 93], this model can use only one sample to compute the colour of the sky with only a slight reduction in quality. However the method allows the inclusion of more samples in the view ray in order to increase the quality of the result. Some results produced by O'Neal's AS model are shown in `figure` 10.

The exp and $scale$ functions used to obtain the results are the ones in [O'Neal 05], as are the scattering coefficients. Five samples were considered for each view ray. With this optimisations it was possible to compute the colours of the

Figure 10. Images using O'Neal model of the atmosphere in clear sky

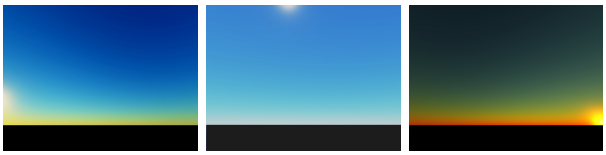

Figure 11. Images using Bruneton model of the atmosphere in clear sky

sky in real-time and still obtain a good quality result.

4.2. Pre-computed Models

After the work of O'Neal, Schafhitzel et al. in 2007 [Schafhitzel 07] and Bruneton et al. in 2008 [Bruneton 08] also took advantage of the programmability of the GPUs.

These models wanted to combine high performance with rendering quality, hence, they tried to perform the highest number of pre-computations possible in order to perform less operations in rendering time.

Nishita et al.[Nishita 93] proposed to pre-compute optical depth for sunlight and so removing the in-scatter integral. O'Neal[O'Neal 04] went further and pre-computed the out-scatter integral for the viewing direction and used analytic expressions to approximate the other integrals. Schafhitzel et al. pre-computed every single integral by assuming not only every possible sunlight and viewing direction but also every possible camera height, and store them in a 3D texture. The stored values were obtained with `equation` 4 with the exception of the sun light intensity (I_0).

With this approach Schafhitzel et al. only needs to fetch the values of the texture and use them to obtain the amount of light that travels towards the camera.

So they replaced the calculations of complex integrals for the calculations of indexes in a 3D table, and so the amount of in-scattered light can be given by `equation` 16. However, Schafhitzel et al. only computes single scattering events.

$$I(\lambda, h, a, b) = I_0(\lambda) texture(h, a, b) \qquad (16)$$

In 2008 Bruneton et al. went further and pre-computed calculations for multiple scattering as well. For this purpose Bruneton et al. pre-computes and stores the values of the multiple scattering values in textures. In a process similar to the one used by Nishita et al. with multiple scattering [Nishita 96], Bruneton et al. uses multiple passes to compute this texture. At each pass the result texture sums the values of `n` scattering order for each sunlight direction, viewing direction and camera position/height.

The difference between Schafhitzel et al. and Bruneton et al. resides basically in the usage of `equation` 4 att_{sun}, where the first is using `equation` 1 and in the second case the `equation` is 6.

Bruneton et al. model produces results like the ones in `figure` 11.

These images were obtained using the coefficients present in [Nishita 93] and [Nishita 96] with 4 orders of scattering computed.

This scheme of pre-computing the heavy integrals of the scattering equations allowed these modern models to produce high quality results in real-time. Bruneton et al. proposed some improvements for the usage of the pre-computed textures specially when rendering detailed terrains, but due to text space limitation its not approached here, however its is fully explained in [Bruneton 08].

5. PERFORMANCE OVERVIEW

The results presented for [Klassen 87], [Nishita 93], [Nishita 96] and [Preetham 99] were obtained with CPU running algorithms. The results obtained with [O'Neal 05] and [Bruneton 08] were based in GPU algorithms. The test machine had an Intel $i5-3210M$ processor with 2 2.5GHz cores and a Nvidia GeForce $630m$. The table presents average frame rates for our implementation of these algorithms.

Algorithm	Time
Klassen	1.42fps
Nishita 1993	0.85fps
Nishita 1996	0.056fps
Preetham	1.1fps
O'Neal 2005	230fps
Bruneton	250fps

As can be seen the simple model of Klassen produces faster results than [Nishita 93] with single scattering because it does not need to compute the density of the atmosphere particles along the view direction. We can also see that the computational cost for the multiple scattering algorithm of [Nishita 96] is about 15 times heavier than the model with single scattering [Nishita 93]. That is to be expect because far more calculations were added per frame.

The pre-computations allied to the analytical expression used in [Preetham 99] improve the time of rendering, but the most notorious aspect of these results is the power of the GPU an its usage. The results of [O'Neal 05] and [Bruneton 08] achieved real-time rendering, which is essential in several applications. Note that Bruneton et al. produces results with multiple scattering, whereas O'Neal does not.

6. CONCLUSION

The physical formulation of the problem is highly complex from a computational point of view. Sampling the integrals

and making assumptions that simplify the required computations, such as considering only single scattering and a constant density atmosphere, was the approach adopted in [Klassen 87]. In [Nishita 93] exponential decreasing density atmosphere is considered but with a performance penalty. Later, in [Nishita 96] multiple scattering is introduced, again with a penalty performance. Pre-processing and curve-fitting was the approach used the later models. Programmable GPUs also play a major role in this history as major contributors performance wise.

From Klassen's single scattering CPU based ray-tracing algorithm to Brunetton et al. GPU multiple scattering model there has been a great evolution. Quality wise earlier models are equivalent to their more recent counterparts. Performance is the key factor that has really changed. Thanks to modern graphic architectures and some pre-processing it is now possible to have realistic skies in real-time computer graphics applications.

7. ACKNOWLEDGEMENTS

Work partially funded by National Funds through FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2014

References

[Bruneton 08] Eric Bruneton and Fabrice Neyret. Precomputed atmospheric scattering. *Comput. Graph. Forum*, 27(4):1079–1086, 2008.

[Bucholt 95] Anthony Bucholt. Rayleigh-scattering calculations for the terrestrial atmosphere. 1995.

[Dobashi 95] Yoshinori Dobashi, Tomoyuki Nishita, Kazufumi Kaneda, and Hideo Yamashita. A fast display method of sky colour using basis functions. *Journal of Visualization and Computer Animation*, 8(2):115–127, 1995.

[Dobashi 02] Yoshinori Dobashi, Tsuyoshi Yamamoto, and Tomoyuki Nishita. Interactive rendering of atmospheric scattering effects using graphics hardware. pages 99–107, 2002.

[Hosek 12] Lukas Hosek and Alexander Wilkie. An analytic model for full spectral sky-dome radiance. *ACM Trans. Graph.*, 31(4):95:1–95:9, July 2012.

[Kaneda 91] Kazufumi Kaneda, Takashi Okamoto, Eihachiro Nakamae, and Tomoyuki Nishita. Photorealistic image synthesis for outdoor scenery under various atmospheric conditions. *The Visual Computer*, 7(5-6):247–258, 1991.

[Klassen 87] R. Victor Klassen. Modeling the effect of the atmosphere on light. *ACM Trans. Graph.*, 6(3):215–237, July 1987.

[Mie 08] G. Mie. Beiträge zur optik trüber medien, speziell kolloidaler metallösungen. *Ann. Phys.*, 330(3):377–445, January 1908.

[Nishita 93] Tomoyuki Nishita, Takao Sirai, Katsumi Tadamura, and Eihachiro Nakamae. Display of the earth taking into account atmospheric scattering. pages 175–182, 1993.

[Nishita 96] Tomoyuki Nishita, Yoshinori Dobashi, Kazufumi Kaneda, and Hideo Yamashita. Display method of the sky color taking into account multiple scattering. 1996.

[oIC04] International Commission on Illumination CIE. Spatial distribution of daylight - cie standard general sky. 2004.

[O'Neal 04] Sean O'Neal. Real-time atmospheric scattering. 2004.

[O'Neal 05] Sean O'Neal. Accurate atmospheric scattering. 2005.

[Perez 93] Richard Perez and Others. All-Weather Model for Sky Luminance Distribution—Preliminary Configuration and Validation. *Solar Energy*, 50(3):235–245, 1993.

[Preetham 99] A. J. Preetham, Peter Shirley, and Brian Smits. A practical analytic model for daylight. pages 91–100, 1999.

[Schafhitzel 07] Tobias Schafhitzel, Martin Falk, and Thomas Ertl. Real-time rendering of planets with atmospheres. *Journal of WSCG*, 15(1-3):91–98, 2007.

[Strutt 71] J. W. Strutt. XV. On the light from the sky, its polarization and colour. *Philosophical Magazine Series 4*, 41(271):107–120, February 1871.

[Zotti 07] Georg Zotti, Alexander Wilkie, and Werner Purgathofer. A critical review of the preetham skylight model. In Vaclav Skala, editor, *WSCG ' 2007 Short Communications Proceedings I*, pages 23–30. University of West Bohemia, January 2007.