LJMU Research Online

Kolivand, H, Sunar, MS, Saba, T and Ali, H

 ReLiShaft: realistic real-time light shaft generation taking sky illumination into account

http://researchonline.ljmu.ac.uk/id/eprint/9437/

Article

For more information please contact researchonline@ljmu.ac.uk

CrossMark

ReLiShaft: realistic real-time light shaft generation taking sky illumination into account

Hoshang Kolivand[1] · Mohd Shahrizal Sunar[2] ·
Tanzila Saba[3] · Hatem Ali[2]

© The Author(s) 2018

Abstract Rendering atmospheric phenomena is known to have its basis in the fields of atmospheric optics and meteorology and is increasingly used in games and movies. Although many researchers have focused on generating and enhancing realistic light shafts, there is still room for improvement in terms of both qualification and quantification. In this paper, a new technique, called ReLiShaft, is presented to generate realistic light shafts for outdoor rendering. In the first step, a realistic light shaft with respect to the sun position and sky colour in any specific location, date and time is constructed in real-time. Then, Hemicube visibility-test radiosity is employed to reveal the effect of a generated sky colour on environments. Two different methods are considered for indoor and outdoor rendering, ray marching based on epipolar sampling for indoor environments, and filtering on regular epipolar of z-partitioning for outdoor environments. Shadow maps and shadow volumes are integrated to consider the computational costs. Through this technique, the light shaft colour is adjusted according to the sky colour in any specific location, date and time. The results show different light shaft colours in different times of day in real-time.

✉ Hoshang Kolivand
  H.Kolivand@ljmu.ac.uk

[1] Department of Computer Science, Liverpool John Moores University, Liverpool L3 3AF, UK

[2] MaGIC-X (Media and Games Innovation Centre of Excellence), UTM-IRDA Digital
    Media Centre, Universiti Teknologi Malaysia, 81310 Skudai Johor, Malaysia

[3] College of Computer and Information Sciences, Prince Sultan University,
    Riyadh 11586, Saudi Arabia

⁂ Springer

1 Introduction

Rendering atmospheric phenomena is known to have its basis in the fields of atmospheric optics and meteorology. Nevertheless, practical computer graphics applications have been hampered by the complexity of the associated rendering problems. The techniques for rendering outdoor scenes are different from those of their indoor counterparts as the sun and the sky colour are the main resources for consistent illumination of outdoor rendering environments [14, 22, 23, 46, 49, 53].

Sky illumination, light shafts and shadows always play a significant role in outdoor and indoor rendering, which are currently very attractive topics for CG researchers. Furthermore, in sky illumination, modeling of atmosphere particles is a fascinating area to make outdoor renderings as realistic as possible [40].

The most salient aspect of outdoor rendering is natural illumination [38, 45], where the sun provides the lighting and the sky colour visually reveals the time of day. Both the sun and (sometimes) the sky colour contribute to shadow generation. Nevertheless, the main issue with applying illumination in virtual environments is dealing with light sources which refer to the sun and sky. A very important facet of natural illumination concerns lighting and shadows. Although there are several tools to illuminate scenes, they are mostly patch-oriented and not suitable for real-time environments.

The main significant issues influencing the realism of virtual objects in virtual environments [53] and Augmented Reality systems [29] is illumination which generates lighting, light shaft and shadows. Illumination is not only light that comes directly from light sources, but also light that comes indirectly from other surfaces and objects. Taking indirect light into account usually enhances the realism of virtual environments [57]. Light shafts are a type of lighting that contributes in both indoor and outdoor environments. Taking indirect lighting into account for light shafts increases their realism as well.

Usually, physically based lighting is employed to approximate light shafts in virtual environments. In outdoor rendering, calculations are performed to determine the colour of each pixel with respect to the sun's position and the sky colour. As a result of the collision between light and surface, numerous complicated dynamic processes occur, specifying the pixel colour of the light-shaft.

A real-time sky colour with respect to the sun's position in any specific location, date and time is required to determine the sky colour energy on a generated light-shaft to be realistic in outdoor lighting. A light shaft is generated when light passes through an extremely narrow gap. In other words, when light passes through some medias such as clouds, trees or any other object, it is scattered by small particles. Air molecules and aerosols are represented by particles of different sizes. Rayleigh scattering [7] and Mie scattering [37] are appropriate models for air molecules and aerosols, respectively.

In this paper, we present a new technique to generate a realistic light shaft, based on the shadow maps idea [54] and the shadow volumes idea [10] without the need of silhouette detection [26]. Furthermore, indirect lighting from the sky colour is also taken into account.

This paper was inspired by attempts to create realistic outdoor scenes, using the integration of different components such as outdoor illumination and shadow position at different times of the day and different days of the year. The integration of these components can be widely used by game developers and the building design industries.

2 Research background

Many researchers have studied the sun's position and the amount of sunshine at different times of the day and year in various locations. For instance, [16] worked on the level of sunshine at various time of the day. The authors of [21] provide several functions to calculate the sun's position by focusing on factors such as light refraction and right ascension.

Daylight is a combination of all direct and indirect light from the sun, the sky colour and the diffusion of other objects (especially the earth). In other words, daylight includes direct sunlight, diffused sky radiation, and a combination of both of these, reflected from the earth and terrestrial objects. The intensity of skylight or sky luminance is not uniform and depends on the clarity of the sky [39].

The sun and the sky provide the main sources of natural illumination. In natural illumination, the sun generates sunlight, which can be utilized to show how the shadows cast by a structure affect the surrounding area. The angle of sunlight is controlled by the location on the earth, the date and the time of day. Skylight is the most important outdoor illumination component to render a realistic scene [12].

Scattering is also a considerable effect, which is employed in many different methods to create realistic images [4, 11, 13, 22, 23].

Early research on light shaft generation was conducted in [19, 20, 34, 35] whilest, more up to date research has developed qualitative and quantitative methods of light shaft generation [31, 36, 58].

Li et al. [31] proposed unified volume representations for light shafts and shadows, and suggested an efficient method for simulating natural light shafts and shadows under atmospheric scattering effects. Sunkavalli et al. [50] presented a model for temporal colour changes exploring its use for the analysis of outdoor scenes from time-lapse video data.

Yang et al. [56] published a paper on visual effects in computer games, focusing on global illumination. The authors of [5] considered a simple skybox based dynamic lighting method for sky rendering with reduced calculation of GPU and CPU. The researchers performed skybox modelling, ambient lighting and sky layers on the CPU and modelled fog on the GPU.

Shuling et al. [44] and Wang et al. [52] have recently worked on simulation of the sun and the sky on clear days. Shuling et al. [44] focused on Rayleigh scattering and [7] used gradient map sampling to generate the sky illumination. They attempted to simplify the Mie [37] scattering of the atmosphere. Wang et al. [52] computed a sky light using an analytic sky model. They considered night time light sources such as moonlight, starlight, zodiacal light and airglow. In this regard, tone reproduction is the technique applied to render realistic sky.

Outdoor rendering is visualised with sunlight and sky colour among many different indirect lightings [43, 45]. Having a realistic sky as a background is a vital factor to show outdoor environments.

In realistic virtual environments, realism is usually the results of sophisticated factors such as lighting, shading, shadows, reflection and absorption [18]. In this paper, we have considered lighting which includes light shafts and shadows with respect to the sky colour in any specific location of the earth, at any date and time.

Annen et al. [2] proposed a technique called Convolution Shadow Maps (CoSMs) which turned out to be an effective arbitrary linear shadow filter. CoSMs replaced storing depth values at each pixel by encoding a binary visibility function. The main difference between

CoSMs and Percentage Closer Filtering (PCF) is that the CoSMs employs Fourier's expansion instead of statistical estimates of the depth distribution. Converging toward the PCF solution is another difference with respect to Variance Shadow Maps when more Fourier coefficients are added. One of the advantages of CoSMs is that they support mip-mapping, which helps to improve the quality of tri-linear and anisotropic filtering and decrease screen space aliasing. Another advantage is the ability to apply blur filtering to directly produce soft shadows. Moreover, CoSMs avoid the non-planarity problem through using double-bounded for its shadow reconstruction function. However, like other techniques, CoSMs have some shortcomings. Most importantly, their quality depends on the truncation order. The higher the truncation order, the more memory is required to save basis textures.

Bruneton et al. [6] presented a realistic atmospheric rendering, taking the sun's position into account. They focused on light shaft rendering with respect to the sun's position with single scattering. Yusov [58] implemented multi-scattering in the same project, considering a light transport equation which is pre-computable for all view points. Nevertheless, a major drawback with this technique is the darkness of the sky at the highest and lowest brightness and realistic sky around the sun at sunrise or sunset.

Ali et al. [1], presented a real-time light shaft generation technique called realTiSoftLS, using downsampling of ray marching to accelerate number of frames per second. They use bilateral filtering to enhance aliasing that is generated due to using downsampling by preserving the edges of generated shadow maps. Combination of light shaft and soft shadow maps provides a cue for revealing distance in large environments. The results are impressive in terms of realism, but still weak when the environment is complex or the resolution is reasonably high.

Kol et al. [25] presented a stylization technique for generating light shafts to be used by artists. In this application, artists are able change the appearance of light scattering- focusing on light shafts- by manipulating the occluder. The authors tried to create a desirable light shaft based on single light scattering but by manipulating the occluders. A similarity of this work with our own work is that of enabling the user to change the colour of the light shaft.

3 ReLiShaft

ReLiShaft is a method for generating real-time realistic light shafts affected by sky illumination with respect to the sun's position in any specific location of Earth, on any date and time. ReLiShaft transfers the energy of the sky's colour with respect to the location, date and time onto a new light shaft.

In this section, the required methods and algorithms are discussed to reveal how ReLiShaft is integrated. This background knowledge will make it easier to understand the whole integration technique. First and foremost, the sun's position is traced. The next step involves the calculation of sky colour in real-time. Generating light-shafts by taking shadows into account is the final requirement before integrating ReLiShaft. The simple framework to employ sky illumination on a light shaft is illustrated in Fig. 1.

3.1 Calculating the sun's position

Julian dating is a sufficiently precise technique to trace the sun's position [48]. Julian dating uses the day of the year as an integer ranging from 1 to 365. By determining the zenith and azimuth for any specific location, date and time using Julian dating, the sun's position can be specified accurately.
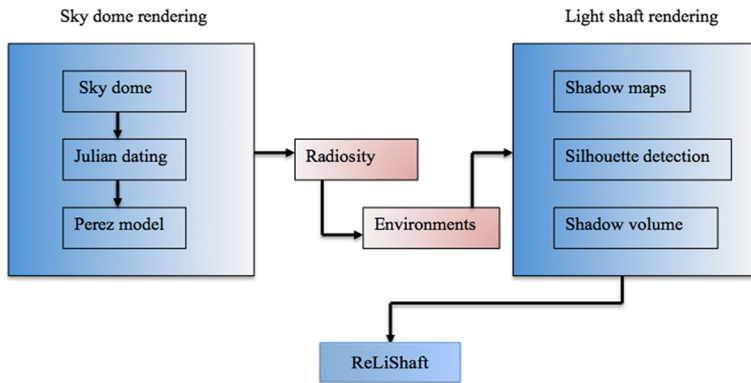
Fig. 1 A framework for ReLiShaft (realistic outdoor light shaft taking sky illumination into account)

3.2 Sky dome and Perez model

A sky dome can be readily generated by 3D software modelling like 3D Studio Max, Maya and even Unity 3D. A mathematical method is employed here as it is not patch-oriented. Creating a hemisphere using 5728 triangles (patches), the realistic sky colour will be available to show the Perez model [41] during daytime.

To generate real-time sky colour, 5728 patches are created mathematically. Each patch will be assigned a different colour, calculated in accordance with the Perez model [28]. These patches generate sky colour. The calculation can be performed at approximately 67 FPS in the experimental results (excluding the energy sharing) which will be discussed later.

The model is convenient to illuminate an arbitrary point (θ_p, γ_p) of the sky dome with respect to the sun's position using the CIE standard ((The Commission Internationale De l' Eclairage)) [9] and could be used for a wide range of atmospheric conditions. Luminance of the point (θ_p, γ_p) is calculated using the following formula:

$$L(\theta_p, \gamma_p) = (1 + Ae^{\frac{B}{\cos\theta_p}})(1 + Ce^{D\gamma_p} + E\cos^2\gamma_p) \tag{1}$$

$$\gamma_p = \cos^{-1}(\sin\theta_s\sin\theta_p\cos(\varphi_p - \varphi_s) + \cos\theta_s\cos\theta_p) \tag{2}$$

A: Darkening or brightening of the horizon
B: Luminance gradient near the horizon
C: Relative intensity of circumsolar region
D: Width of the circumsolar region
E: Relative backscattered light

In addition to the sun's position, the sky colour is a very effective factor in rendering the outdoor environments more realistically. The integration of sky colour and shadows affected by the sun's position can make scenes more realistic, presenting a view similar to that experienced by a human's eye. To calculate the sun's position, the longitude, latitude, date and time are required. To handle these parameters, a suitable GUI is prepared. OpenGL GLUT and GLUI are the libraries used to construct the GUI. Setting the parameters, the software produces the desired sky colour and related shadows with respect to the sun's position. Figure 2 shows the two different GUI samples created in this research.
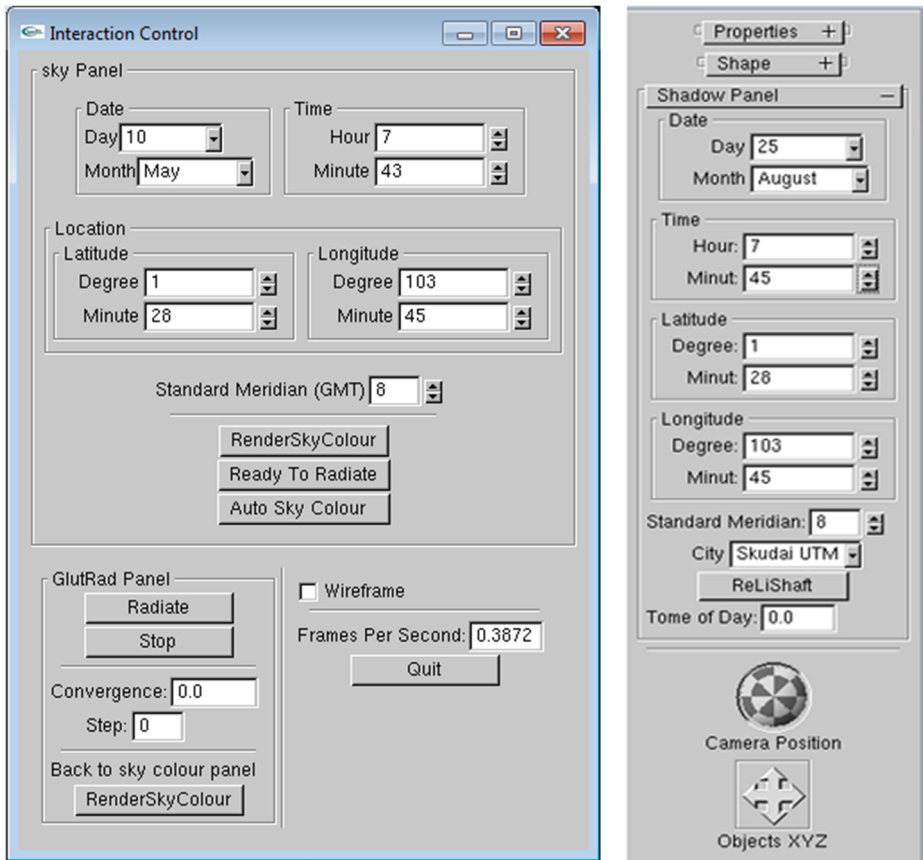
Fig. 2 Two different GUIs, which have been used in our application

3.3 Lighting

The process of lighting in virtual environments and augmented reality systems involves an approximation of real lighting, which is physically based. In outdoor rendering, the colour of each pixel must be calculated with respect to the sun's position and sky illumination. In the collision of light with surfaces, many parameters must be taken into account. Improving realism requires knowledge of the dynamics of photons, which involves studying electromagnetic wave modelling under optics. Two different parameters involved in electromagnetic wave modelling are the number of photons and the energy contained in the photons. In this research, we have considered the energy of sky colour and transferred it to the brightness of a light shaft, resulting in a different appearance of these light shafts at different times of the day. In other words, instead of considering electromagnetic wave modelling with respect to photon modelling, we have generated the sky illumination separately for different time of the day. Then this lighting is taken into account during the light shaft generation and contributes to the colour of the light shaft, rather than computing with the number of photons and their energy. The details of how this energy sharing is employed is presented in the next sections.

3.4 Light scattering in outdoor scenes

In real-time outdoor rendering, sky colour and shadows are the most visually dominating parameters of lighting. Atmospheric scattering in outdoor scenes is one of the most important effects in real-time outdoor rendering to produce realistic outdoor environments. The colour of the sun changes from light red at sunrise to yellow at noon and again to light red at sunset, all of which cause atmospheric scattering. During daytime, the sky colour changes with the time of day and light scattering, all of which have been taken into account in the proposed method.

In outdoor scenes, although the sun is a point light source, shadows are semi-soft due to the long distance from the sun and a lot of indirect lighting which comes from the sky [17]. However, the different skylight due to the different position of the sun during the day or during the year need different lighting in virtual environments to correspond with the real one.

3.5 ReLiShaft

In outdoor environments, light shafts are generated when objects such as clouds, mountains and trees block part of the sun's rays [38]. A dark volume can be observed surroundings the occluders. This effect is called a light shaft in an outdoor environment. Figure 3 illustrates this natural phenomena.

For outdoor light shaft rendering, regular ray marching is replaced by a filtering technique based on rectified shadow maps [24]. Isotropic scattering is also considered to enhance the quality of light shafts for frame rate control. In this regard, a review of light transport is in order.

The regular ray-marching has been replaced with simple filtering to speed up rendering. Annen et al. [2] used ray-marching which consumes more time for both the CPU and GPU.
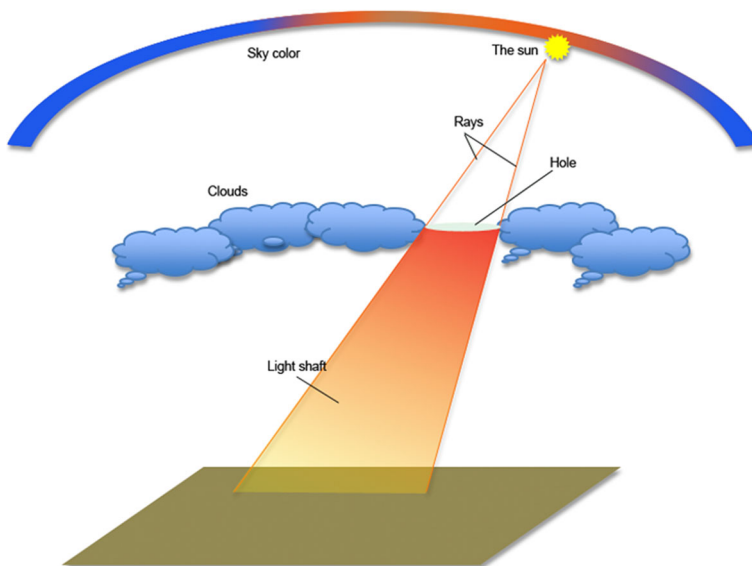

Fig. 3 Theory of light shafts

As shadow maps are considered for scattering contribution, the filtering technique does not rely on the camera's point of view, resulting in independent filtering which leads to a scene independent technique which can be used in any game platform. A rectified shadow map [24] based on matrix multiplication is used to accelerate the technique in the case of an independent camera's view point.

The proposed technique can be used in complex scenarios as shadow map filtering is employed, instead of a GPU-friendly min-max hierarchy previously used by Chen et al. [8].

To enhance the speed of rendering, a light shaft is constructed through integrating shadow volumes and shadow maps [27] and light scattering is taken into account. Shadow maps are employed to avoid silhouette detection, geometrically. The whole scene is rendered from a defined point of view and is stored in a depth buffer. Subsequently, the whole scene is rendered, but from the light source point of view. The area of volume is then recognized without silhouette detection (which is extremely expensive). In the next step, volumes are generated based on the distance between occluders and shadow receivers. These volumes are used to construct the light shaft.

In the first stage, the Z-buffer and the colour buffer are cleared. Ambient and emissive lighting terms are enough for the first rendering. Meanwhile, disabling the stencil buffer is required to prevent any writing in the Z-buffer and the colour buffer. Finally, before employing the Z-buffer test, the scene need to be rendered with only diffuse and specular lighting.

In the cases of light scattering, two regular factors including luminance and w are taken into account which are equivalent with contributing intensity on the same factor (w). The luminance of sky light calculated based on Equation (3).

$$I_{skylight} = \int_{\Omega} \pounds(s)\kappa(s)ds \qquad (3)$$

In this case, the hemispherical integral domain is represented by Ω, while the distribution of sky intensity is presented by $\pounds(s)$. s is a unit vector on the hemisphere. The reflectance function for vector s is represented by $\kappa(s)$.

As was mentioned earlier, a sky dome is a hemisphere with a large radius. Given a local coordinate system with the z-axis as normal and the previously discussed technique for sky dome generation, the luminance of the sky can be calculated.

The radiance of pixel a x in the direction of ω is calculated based on inscattering light, which is calculated by (4) where transmittance τ accounts for outscattering.

$$\pounds(x, \omega) = \int e^{\int_0^{|s\omega|} \sigma_s(x+s'\omega)ds'} \pounds_i(x + s\omega, -\omega)ds, \qquad (4)$$

where σ_s is the scattering coefficient.

Now, with a simulated sky, a light shaft under the sky colour effect can be systemised based on Fig. 3. When part of the sky generating the light shaft is reddish, the effect will be observed on the light shaft in the form of a reddish light shaft. When this is bluish in colour a bluish light shaft is generated. These influences are constructed based on the energy of the sky colour. Sharing sky energy will be explained in the next section.

For the part of indoor rendering, ray marching based on epipolar sampling [15] is employed, while for outdoor rendering, it is replaced by the filtering technique where radiance of a ray toward the point of view at point x from direction ω_i is calculated by (5) due

to the single direction light source from the sun. We used the probability distribution (p) [42] for light scattering from ω' to ω.

$$Ł_i(x, z) = \sigma(z) \int_{4\pi} p(z, \omega, \omega') L(z, \omega') d\omega' \tag{5}$$

Z-partitioning is a well-known and widely used technique in current game engines. In Z-partitioning, the view frustum is split along the z-axis. The shadow map will be stored separately for each partition. Figure 4a reveals the idea of Z-partitioning for enhancement of rendering speed. The view frustum is split into more partitions, depending on the fluctuation of partitions being divided in three or more classes. The classification occurs due to the assignment of appropriate resolution for each class. Class A, which is located from the first object to almost $x = 1$ in $d = \log(x)$, is a high relative frequency of partitions. Class B is located in the middle of the scene, and depends on the size of the scene. Finally, Class C represents those scene parts located far away from the camera which are not visible to the viewer. The dispersion index can be used to categorize the scene parts into three classes.

Class A requires high resolution for quality rendering results. To achieve this, 128 bits, 64 bits or a minimum of 32 bits are suitable. Class B needs less resolution compared to class A and thus 32 bits or 16 bits suffice for this class. Finally, class C can be rendered with very low resolution with no effect on the quality of the shadow. Although using this technique can improve rendering quality and FPS, reduction of the precise texture is a problem with this idea.
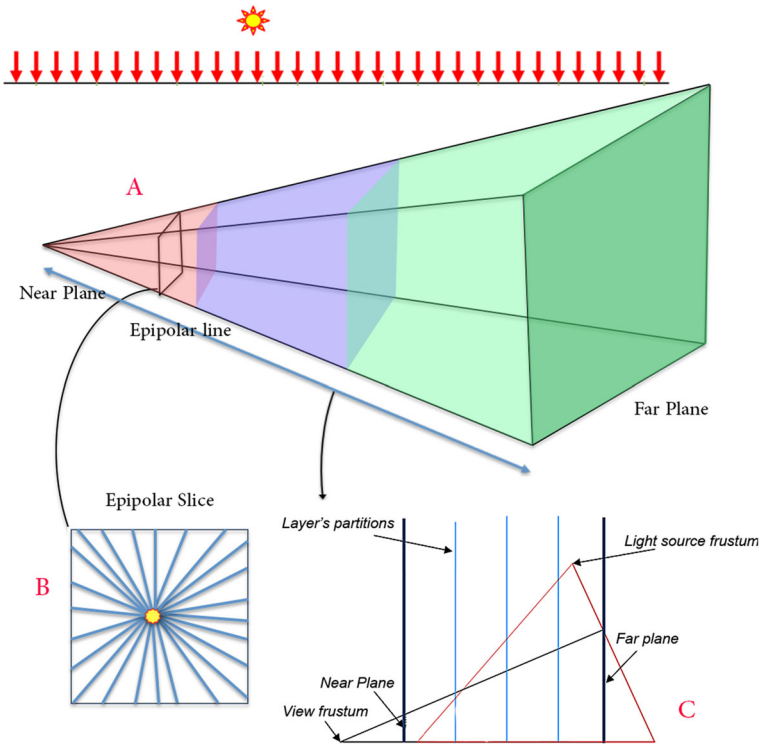

Fig. 4 **a**: Camera view including light shaft A filtering on regular epipolar on z-partitioning, **b**: vertical slice of epipolar, **c**: z-partitioning by capturing four different shadow maps for each partition

4 Implementation

The sky colour is generated based on the Perez model [28]. The Julian dating is employed to control the sun's position in any specific location, date and time. The shadow generation is based on an improvement of shadow maps [54]. Light-shaft generation is also based on shadow maps.

Generating sky colour based on the Perez model [41] has been documented earlier [27, 30]. Some results are illustrated in Fig. 5.

Sharing the sky colour energy onto a light shaft is employed based on a hemicube technique for regular radiosity. The radiosity technique considers shooting rays from one patch of the sky dome to particles of a light shaft.

A hemicube technique is applied on the top of the volume (ToV) which includes many patches itself. By shooting rays from the S_{ith} patch of the sky (including 5728 patches) to the V_{jth} patches of volume, on the top. The colour energy of the sky is transferred to each patch of ToV. These shootings are continuously updated until a hemicube is placed on the S_{ith} patch of the sky. The hemicube is then employed until all sky colour energy is received by each patch on the top of the volume. A continuous reiteration is applied to update the values of radiosity until the desirable ones are achieved.

Each patch is surrounded by a hemicube consisting of 5 planar surfaces; four for the sides and one for the top. The form factors are pre-calculated to reduce the time of rendering. A visibility-test is taken into account to enhance the rendering speed which is presented in Algorithm 1:

Algorithm 1 Hemicube visibility-test radiosity

For each patch i
  For each patch j apart from i
   If $A_j.visible = true$
     $F_{ij} = E_{ij}A_i/A_j$
     $R_j = R_j + R_i(P_i F_{ij})$
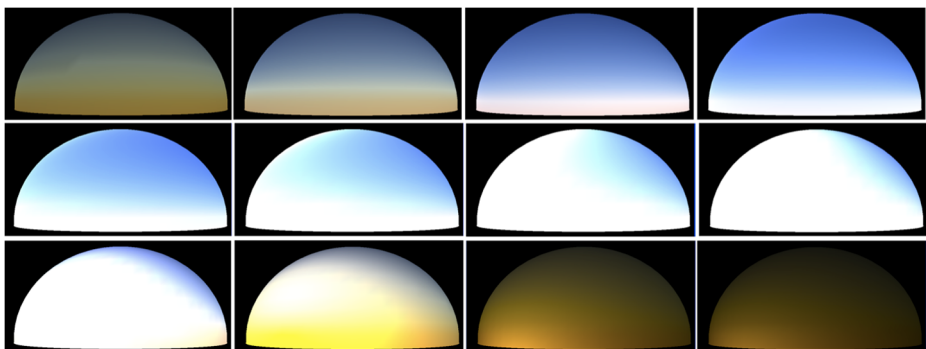   *End if*
  Next j
Next i

Fig. 5 Virtual sky colour at different times of the day captured from a horizontal view

Where,

R_i: Radiosity
P_i: Reflection
A_i: Area of ith patch
E_{ij}:   Amount of energy from i^{th} patch to j^{th} patch
F_{ij}:   Form factor from i^{th} patch to j^{th} patch [47].

Algorithm 2 is a representation of Algorithm 1.

Algorithm 2 Reiterations radiosity

$For\ each\ patch\ i$
 $Set\ the\ hemicube,\ calculate\ form\ factor\ F_{ij};$
 $For\ each\ patch\ j\ apart\ from\ i$
  $If\ A_j.visible = true$
   $\Delta Rad = P_j \Delta R_i F_{ij} A_i / A_j$
   $\Delta R_j = \Delta R_j + \Delta Rad$
   $R_j = R_j + \Delta Rad$
  $End\ if$
 $Next\ j$
$Next\ i$
$\Delta R_i = 0$

---

where, ΔR_j is the un-shot radiosity.

The value of form factor F_{ij} is calculated using the hemicube which is placed on the ith patch of sky. Nevertheless, the cameras are to be placed on the all planes of the hemicube for each patch.

Figure 6 shows the idea after the shooting. When a hemicube is employed each patch gets the specific energy of the sky colour. Therefore, the light that reaches the patch acts on the single volume related to that patch. This effect is based on the generated sky colour with respect to the sun's position at any specific location, date and time.

A linear equation system integrating the radiosity, reflection and the form factor of each patch ($R_j = R_j + R_i(P_i F_{ij})$) is employed to determine the patch with the greatest amount of
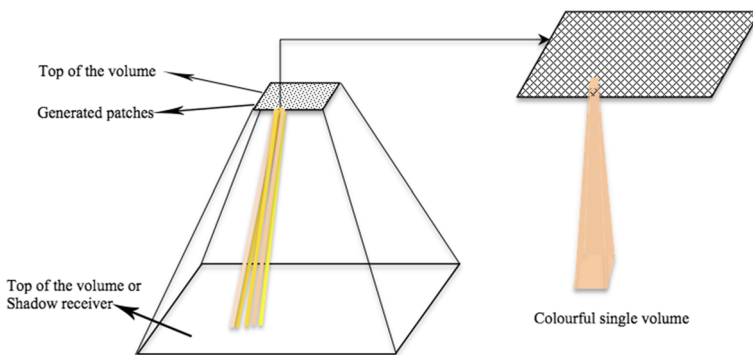

Fig. 6 Single patch volume on ToV

energy. The patch with highest energy is then the first to shoot. This process is continuously reiterated for all patches near to the light source in the direction of light and ToV. ToV is the radiosity caster scene which we have taken into consideration in this research. The highest value of $\Delta R_i A_i$ is enough to determine the highest amount of energy.

5 Results and discussion

Figure 7a shows the real sky colour and Fig. 7b the virtual sky colour, June 15th at Universiti Teknologi Malaysia at different times of day at 8:48 am. In the two left pictures during the sunrise the top of the sky is a little dark blue while it is getting to be lighter and lighter on the top of the sun. When the sun is low in the sky, sunlight tends to become yellow or orange in colour. This is because more atmosphere is traversed by the sunlight on the way to the viewer. The blue light has been scattered away from the line of sight. Figure 7c was captured at 5:19 pm on a different day and different place with (A) which is the Eiffel Tower in Paris. Figure 7d depicts a simulation of Fig.7c at the same location, date and time. In the two right hand pictures during midday, the sun's position is near the horizon, the sky is light and between these two positions the sky is blended light and blue. Readers may wonder how this simulated sky is realistic. The main concern is to simulate the light source position and colour of the sky apart from the sun, which needs to be simulated in real-time. It should be mentioned that the objects we are looking for are generated in real-time, signifying that by setting the location, date and time, the sky colour will be generated automatically with respect to these parameters.

The result of realistic light shafts under a sky colour effect in our application is illustrated in Fig. 8 at different times of the day at 7:45 am and 11:45 am on 25 August, respectively.

Figure 9 is an outdoor environment at some different times in the morning of Aug 25th at Universiti Teknologi Malaysia. Figure 9a, b are generated in the morning when the sky looks yellowish. The interaction between sky colours and light shaft is distinct in these pictures. The pictures are captured at different times when the sun is moving slowly. The effect of the light shaft is also different. Figure 9c and d show the same outdoor environment shown in A and B at a different time, at noon on Aug 25th at Universiti Teknologi Malaysia. Figure 9c and d are generated between 10 am and 11 am when the sky looks bluish. The effect of light shaft is also different.

Figure 10 illustrates the use of light shaft compared to direct lighting in different environments. Figure 10a and b represent the same environments. In Fig. 10a direct lighting is employed while in (B) light shaft under sky illumination effect is taken into account. This is the same for Fig. 10c and d and also Fig. 10e and f but in different environments and different times of the day.

In the case of sky colour simulation in real-time rendering, our method can be compared with the latest method on outdoor rendering for light shaft generation [6, 58] and [25]. Figure 11a is the result of Yusov [58] and (B) shows sunrise creation using several transfer functions by Kol et al. [25] using linear interpolation with five user defined points in a time domain which extends colour modifications of light shafts. (C) is the result of our method to simulate the sky colour. In comparison, Fig. 11a looks realistic during the sunrise and sunset while at other times of the day, there are no more changes. The results show that from sunrise to sunset the sky looks the same. (B) mostly appears to be generated by an artist, rather than by realistic sky colour creation, as it is the same as the real sky colours. (C) depicts our results for the sky colour. From sunrise to sunset the sky changes smoothly according to the real sky with respect to the location, date and time as shown in Fig. 7.
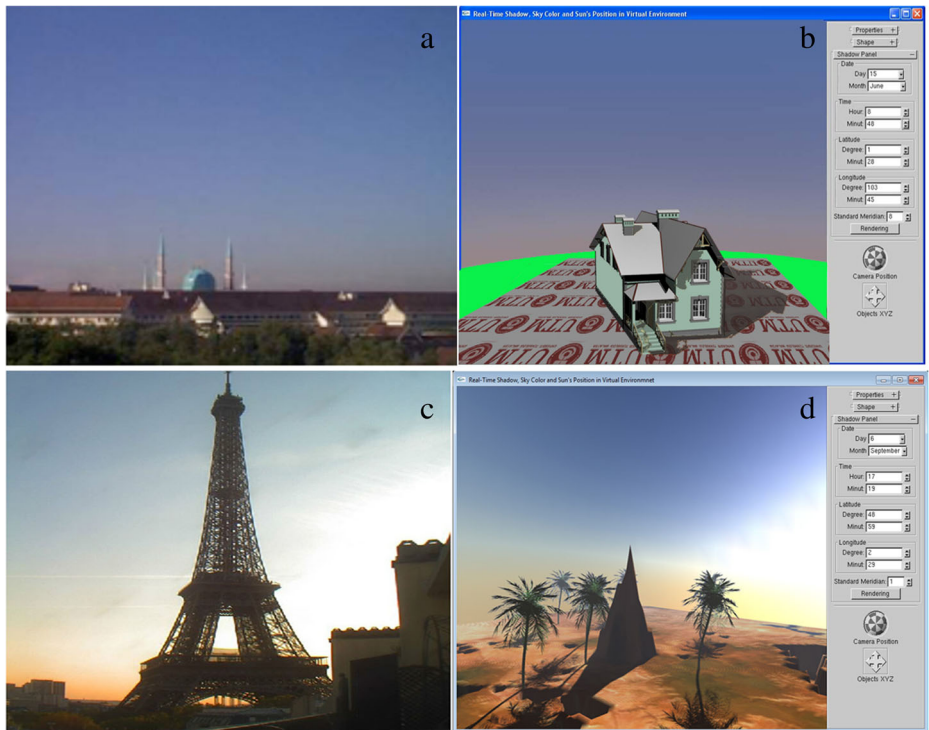
Fig. 7 Real sky colour and virtual sky colour, **a**: Real sky colour in Universiti Teknologi Malaysia 15 June. **b**: Result of the our application (UTM, 15 June) at a latitude of 1.28 and longitude of 103.45 at 8:48 am, **c**: Real sky colour in Paris 6 September 2013 at 5:19 pm, **d**: Result of the our application (Eiffel Tower , 6 September) at a latitude of 48.59 and longitude of 2.29 at 5:19 pm

In Fig. 12, A and B are real pictures at different times of the day with different directions of lighting. C is one of the latest works in light shaft generation for indoor rendering [1]. D is the result of [31], while E is an outdoor light shaft [58]. F represents epipolar sampling with single scattering [15]. G is another latest work in light shaft generation using single
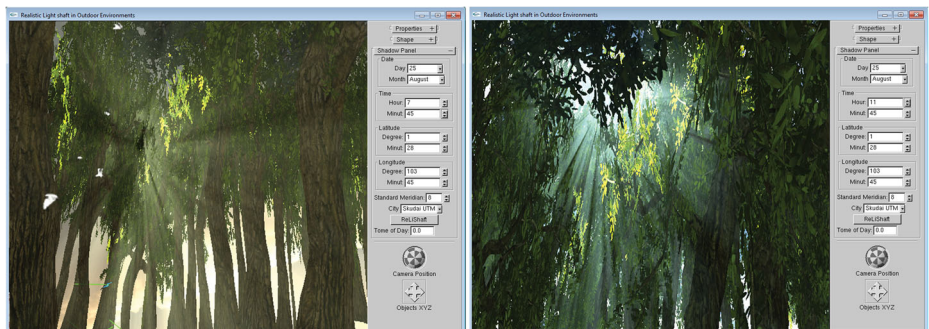

Fig. 8 Realistic light shaft with the effect of sky illumination in the presented application at different times of a day in the same location
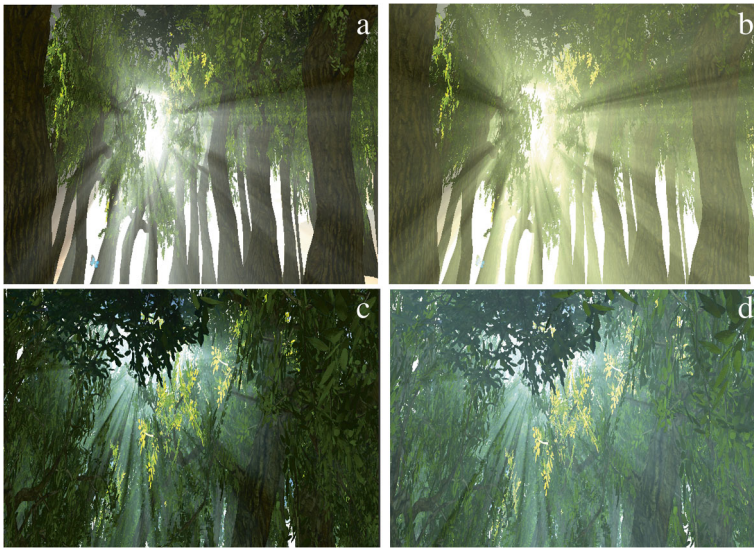
Fig. 9 Realistic light shaft with the effect of sky illumination. A and B: in the morning at 7:45 am, C and D at 11:45 in Universiti Teknologi Malaysia in different points of view

scattering. A similarity of this work with our work is enabling the user to change the colour of the light shaft. In Kol et al.'s work [25], the user changes the colours using a transfer function while in our work, the colour of the light shaft is adjusted using the effect of real-time generated sky colour. H and I are results of ReLiShaft in different situations. Real light shafts and the latest techniques are illustrated to show the difference between the previous methods and the proposed method based on a visualization technique. The desired outdoor light shaft is achieved by comparing not only the previous work, but also the real light shaft
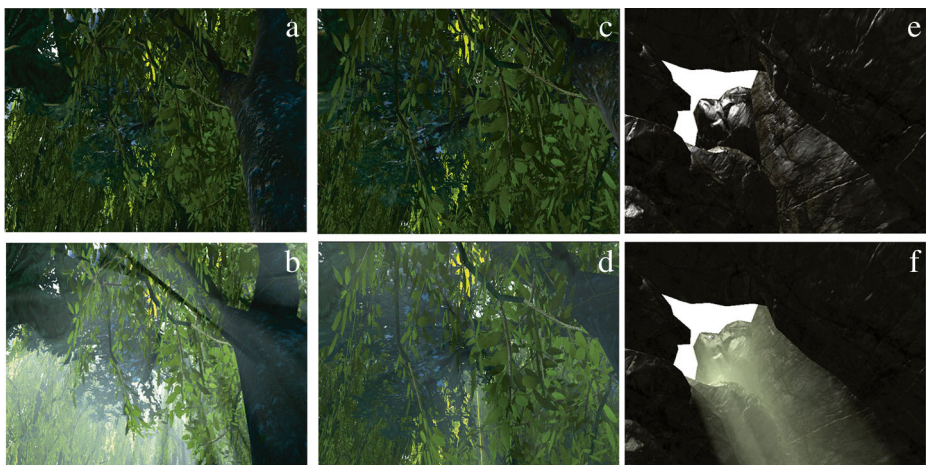

Fig. 10 A comparison between direct lighting and realistic light shafts with the effect of sky illumination at different locations, dates and times. Up: real-time rendering using directional lighting. Down: real-time rendering using realistic light shafts with effects of sky illumination
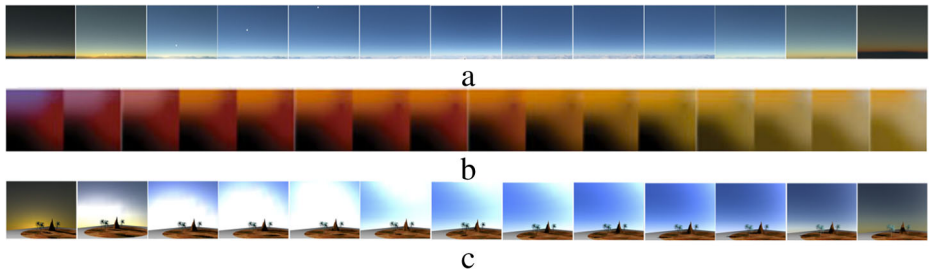
Fig. 11 A comparison between the latest sky colour generation techniques [25, 58] and our results in sky colour generation

which are the best benchmark for the current work as advocated by most researchers [14, 32, 33, 48, 55].

The results of this section show how the objectives and aims of the research, to generate realistic light shafts are achieved.

The final integrated software is implemented with three different resolutions and with different numbers of triangles. Two cases of 66000 and 188000 triangles respectively are tested. Table 1 elucidates a quantitative comparison between the latest state-of-the-art in
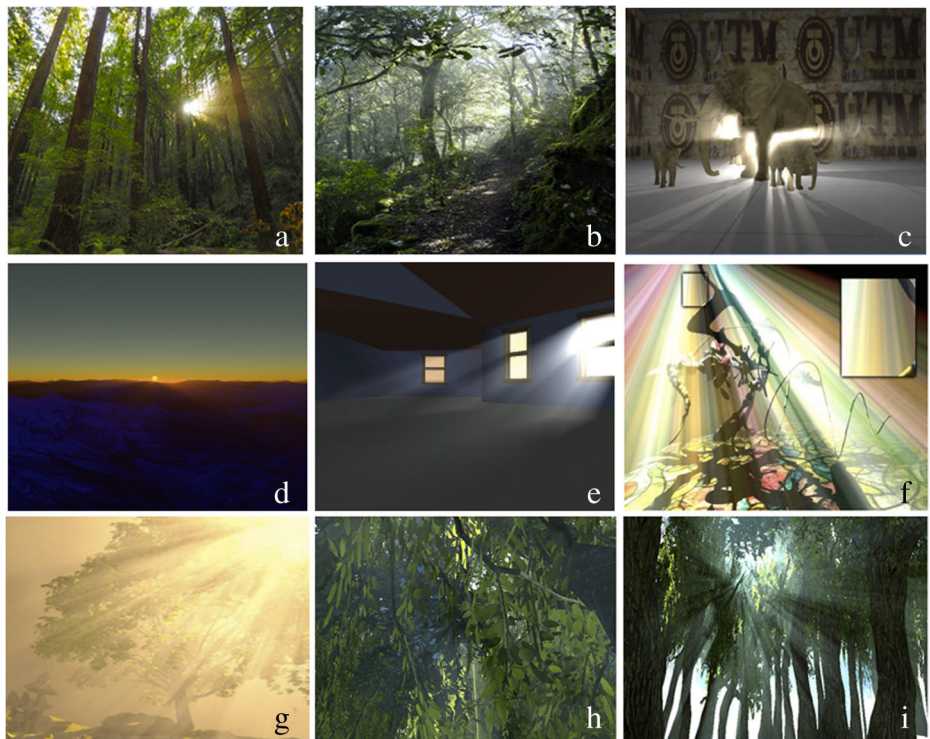

Fig. 12 A comparison between realistic light shaft, **a** and **b**: real light shaft, **c**: [1], **d** [31], **e**: [58], **f**: Epipolar sampling with single scattering [15], **g**: [25], **h** and **i**: Our method (ReLiShaft)

Table 1 Comparison between ReLiShaft and state-of-the-art in light shaft generation techniques

Method	66000Tri			188000Tri		
	1024^2	2048^2	4096^2	1024^2	2048^2	4096^2
Volumetric lighting [51]	45	26	11	26	16	8
Unified volumes [31]	68	38	19	30	19	11
Epipolar sampling [15]	38	20	12	21	10	7
Volumetric shadows [3]	67	38	20	32	23	14
realTiSoftLS [1]	74	41	19	41	25	13
ReLiShaft	62	26	20	57	28	18

light shaft generation and ReLiShaft. As can be observed, realTiSoftLS [1], Unified Volumes [31] and Volumetric Shadows [3] have the highest FPS when the number of triangles is not so high. However when the number of triangles increased to 188000, ReLiShaft shows the best results with the lowest level of tolerance compared to the 66000 triangles. The result from this quantitative comparison highlights the importance of the newly proposed method as there is slight variation in FPS when the number of triangles increases. The minimal level of variation makes it an appropriate method to be taken into account for different high quality real-time rendering like conventional and interactive games.

In conclusion, as can be seen from the FPS indicators, the presented method is stable for different type of scenes in the case of complexity. Therefore, it could be used for real-time rendering.

The results of Table 1 are obtained using an Alienware PC, Intel Core i7-2670QM CPU @2.20 GHz and 8.0 GB RAM with Graphic Hardware GeForce 7025 NVIDIA nForce 630a. The experiential results emanating from the present software in the case of virtual environments, which include the effect of sky colour on the light shaft, are mentioned earlier.

6 Conclusion

In this paper, a new technique is presented to generate realistic light shafts for real-time outdoor rendering. The sun's position is traced based on Julian dating. Sky colour is generated based on the Perez model. The light shaft is created on the basis of an integration of image-based and geometrically-based techniques. The new technique is free from heavy part of geometric techniques and free from the aliasing of image-based techniques. The effect of sky illumination is transferred to the light shaft, based on a hemicube technique for regular radiosity.

Augmented Reality (AR) is the next step we are going to focus on. Applying realistic light shafts in augmented reality under sky colour effects in AR is in need of more improvement in computational speed. Simplifying the calculation of the sky colour through reducing the number of sky dome patches may resolve this issue.

References

1. Ali HH, Sunar M, Kolivand H (2017) Realistic real-time rendering of light shafts using blur filter: considering the effect of shadow maps. Multimed Tools Appl, 1–16
2. Annen T, Mertens T, Bekaert P, Seidel H-P, Kautz J (2007) Convolution shadow maps. In: Proceedings of the 18th eurographics conference on rendering techniques. Eurographics Association, pp 51–60
3. Billeter M, Sintorn E, Assarsson U (2010) Real time volumetric shadows using polygonal light volumes. In: Proceedings of the conference on high performance graphics, pp 39–45
4. Blinn JF (1982) Light reflection functions for simulation of clouds and dusty surfaces. ACM SIGGRAPH Comput Graph 16(3):21–29
5. Braun H, Cohen M (2010) A simple model for real time sky rendering. NVIDIA, Technical Report
6. Bruneton E, Neyret F (2008) Precomputed atmospheric scattering. In: Computer graphics forum, vol 27, no 4. Wiley Online Library, pp 1079–1086
7. Bucholtz A (1995) Rayleigh-scattering calculations for the terrestrial atmosphere. Appl Opt 34(15):2765–2773. Optical Society of America
8. Chen J, Baran I, Durand F, Jarosz W (2011) Real-time volumetric shadows using 1d min-max mipmaps. In: Symposium on interactive 3D graphics and games. ACM, pp PAGE–7
9. CIE (2012) Online: http://www.cie.co.at/cie/home.html, cited 1st June, 2012
10. Crow F (1977) Shadow algorithms for computer graphics. Comput Graph 11(2):242–247
11. Dobashi Y, Kaneda K, Nakashima T, Yamashita H, Nishita T, Tadamura K (1994) Skylight for interior lighting design. In: Computer graphics forum, vol 13, no 3. Wiley Online Library, pp 85–96
12. Dobashi Y, Kaneda K, Yamashita H, Nishita T (1996) Method for calculation of sky light luminance aiming at an interactive architectural design. Comput Graph Forum (Proc EUROGRAPHICS'96) 15(3):112–118
13. Dobashi Y, Kaneda K, Yamashita H, Okita T, Nishita T (2000) A simple, efficient method for realistic animation of clouds. In: Proceedings of the 27th annual conference on computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co, pp 19–28
14. Dobashi Y, Yamamoto T, Nishita T (2002) Interactive rendering of atmospheric scattering effects using graphics hardware. In: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on graphics hardware, pp 99–107
15. Engelhardt T, Dachsbacher C (2010) Epipolar sampling for shadows and crepuscular rays in participating media with single scattering. In: Proceedings of the 2010 ACM SIGGRAPH symposium on interactive 3D graphics and games. ACM, pp 119–125
16. Glover J, McCulloch J (1958) The empirical relation between solar radiation and hours of sunshine. Q J Roy Meteorol Soc 84(360):172–175
17. Goldwasser SM (2015) American evaluation association. [Online]. Available: http://www.repairfaq.org/sam/lasersaf.htm
18. Jansen FW, Chalmers A (1993) Casting shadows in real time. In: Michael, F, Cohen, CP, Sillion, F (ed) Fourth eurographics workshop on rendering, pp 27–46
19. Jensen HW, Christensen PH (1998) Efficient simulation of light transport in scences with participating media using photon maps. In: Proceedings of the 25th annual conference on computer graphics and interactive techniques. ACM, pp 311–320
20. Kajiya JT, Von Herzen BP (1984) Ray tracing volume densities. In: ACM SIGGRAPH computer graphics, vol 18, no 3. ACM, pp 165–174
21. Kambezidis H, Asimakopoulos D, Helmis C (1990) Wake measurements behind a horizontal-axis 50 kw wind turbine. Solar Wind Tech 7:177–184
22. Kaneda K, Okamoto T, Nakamae E, Nishita T (1991) Photorealistic image synthesis for outdoor scenery under various atmospheric conditions. Vis Comput 7:247–258
23. Klassen R (1987) Modeling the effect of the atmosphere on light. ACM Trans Graph 6:215–237

24. Klehm O, Seidel H-P, Eisemann E (2014) Filter-based real-time single scattering using rectified shadow maps. J Comput Graph Tech 3(3):7–34
25. Kol TR, Klehm O, Seidel H-P, Eisemann E (2017) Expressive single scattering for light shaft stylization. IEEE Trans Visual Comput Graph 23(7):1753–1766
26. Kolivand H, Sunar M (2011) New silhouette detection algorithm to create real-time volume shadow. In: 2011 workshop on digital media and digital content management. IEEE, pp 270–274
27. Kolivand H, Sunar M (2012) An overview on based real-time shadow techniques in virtual environment. Telkomnika 10(1):171–178
28. Kolivand H, Sunar M (2012) Real-time outdoor rendering using hybrid shadow maps. Int J Innov Comput Inf Control (IJICIC) 18(10B):7169–7184
29. Kolivand H, Sunar M (2014) Covering photo-realistic properties of outdoor components with the effects of sky color in mixed reality. Multimed Tools Appl 72(3):2143–2162
30. Kolivand H, Sunar M (2014) Realistic real-time outdoor rendering in augmented reality. PloS one 9(9):e108334
31. Li S, Wang G, Wu E (2007) Unified volumes for light shaft and shadow with scattering. In: 2007 10th IEEE international conference on IEEE computer-aided design and computer graphics, pp 161–166
32. Liu N, Pang M (2009) A survey of shadow rendering algorithms: projection shadows and shadow volumes. In: Second international workshop on computer science and engineering, pp 488–492
33. Liu Y, Qin X, Xing G, Peng Q (2010) A new approach to outdoor illumination estimation based on statistical analysis for augmented reality. Comput Anim Virt Worlds 21(3–4):321–330. Wiley Online Library
34. Max NL (1986) Atmospheric illumination and shadows. In: ACM SIGGRAPH computer graphics, vol 20, no 4. ACM, pp 117–124
35. Max NL (1986) Light diffusion through clouds and haze. Comput Vis Graph Image Process 33(3):280–292
36. McGuire M, Enderton E (2011) Colored stochastic shadow maps. In: Symposium on interactive 3D graphics and games. ACM, pp 89–96
37. Mie G (2011) Online: http://en.wikipedia.org/wiki/gustavmie
38. Moro Y, Miyazaki R, Dobashi Y, Nishita T (2017) A fast rendering method for shafts of light in outdoor scene
39. Nishita T, Nakamaei E (1986) Continuous tone representation of three-dimensional objects illuminated by sky light. Comput Graph 20(4):112–118
40. Nishita T, Nakamae E, Dobashi Y (1996) Display of clouds and snow taking into account multiple anisotropic scattering and sky light. In: Rushmeier H (ed) SIGGRAPH 96 conference proceedings, annual conference series, pp 379–386
41. Perez R, Seals R, Michalsky J (1993) All-weather model for sky luminance distribution - preliminary configuration and validation. Sol Energy 50:235–245
42. Pharr M, Humphreys G (2004) Physically based rendering: from theory to implementation. Morgan Kaufmann
43. Preetham A, Shirley P, Smith B (1999) A practical analytic model for daylight. In: Computer graphics, (SIGGRAPH '99 Proceedings), pp 91–10
44. Shuling D, Tizhou RCQ (2009) Real-time simulation of sky and sun in clear day [J]. J Comput-Aided Design Comput Graph 3:005
45. Rönnberg S (2004) Real-time rendering of natural illumination. Citeseer
46. Salesses P, Schechtner K, Hidalgo CA (2013) The collaborative image of the city: mapping the inequality of urban perception. PloS one 8(7):e68400
47. Shao M-Z, Badler I (1993) A gathering and shooting progressive refinement radiosity method
48. Sunar M (2001) Sky colour modelling. Master Thesis, University of Hull
49. Sunar M, Kari S, Bade A (2003) Real-time of daylight sky colour rendering and simulation for virtual environment. In: IASTED international conference on applied simulation and modeling (ASM 2003), pp 3–5
50. Sunkavalli K, Matusik W, Pfister H, Rusinkiewicz S (2007) Factored time-lapse video. ACM Trans Graph 26:1–10
51. Tóth B, Umenhoffer T (2009) Real-time volumetric lighting in participating media. Eurographics Short Papers, vol 14
52. Wang G, Ji Z, Zhang Z (2012) Realistic sky rendering in real-time. Gaojishu Tongxin/Chinese High Technol Lett 22(8):791–796
53. Wang L, Yang Z, Ma Z, Zhao Q (2012) Approximating global illumination on mesostructure surfaces with height gradient maps. Vis Comput 28(4):329–339
54. Williams L (1978) Casting curved shadows on curved surfaces. SIGGRAPH '78 12(3):270–274
55. Xing G, Liu Y, Qin X, Peng Q (2012) A practical approach for real-time illumination estimation of outdoor videos. Comput Graph 36:857–865
56. Yang X, Yip M, Xu X (2009) Visual effects in computer games. Computer 42(7):48–56. iEEE

57. Yi J, Mao X, Chen L, Xue Y, Rovetta A, Caleanu C-D (2015) Illumination normalization of face image based on illuminant direction estimation and improved retinex. PloS one. https://doi.org/10.1371/journal.pone.0122200
58. Yusov E (2015) Outdoor light scattering sample update. [Online]. Available: http://software.intel.com/en-us/blogs/2013/06/26/outdoor-light-scattering-sample

Hoshang Kolivand received his MS degree in applied mathematics and computer from Amirkabir University of Technology, Iran, in 1999, and his PhD from Media and Games Innovation Centre of Excellence (MaGIC-X) in Universiti Teknologi Malaysia in 2013. He has completed his Post Doctoral in Augmented Reality in UTM. Previously he worked as a lecturer in Shahid Beheshti University, Iran and then as a Senior Lecturer in Universiti Teknologi Malaysia. Currently he is a Senior Lecturer in Liverpool John Moores University. He has published numerous articles in international journals, conference proceedings and technical papers, including chapters in books. He is an active reviewer of many conference and international journals. He has also published many books in object-oriented programming and mathematics. His research interests include Computer Graphics, Virtual Reality and Augmented Reality.

Mohd Shahrizal Sunar obtained his PhD from National University of Malaysia in 2008. His major field of study is real-time and interactive computer graphics and virtual environment. He received his MSc in Computer Graphics and Virtual Environment (2001) from The University of Hull, UK and BSc degree in Computer Science majoring in Computer Graphics (1999) from Universiti Teknologi Malaysia. He served as academic member at Computer Graphics and Multimedia Department, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia since 1999. Since 2009, he had been given responsiblity to lead the department. The current research program that he lead are Driving Simulator, Augmented Reality, Natural Interaction and Creative Content Technology. He had published numerous articles in international as well as national journals, conference proceedings and technical papers including article in magazines. Dr. Shahrizal is an active professional member of ACM SIGGRAPH. He is also a member Malaysian Society of Mathematics and Science.

Tanzila Saba earned her PhD at Universiti Teknologi Malaysia. Currently she is a senior lecturer at King Abdul Aziz University in Saudi Arabia.

Hatem Ali just finished his PhD at Media and Games Innovation Centre of Excellence (MaGIC-X) in Universiti Teknologi Malaysia.