International Journal of Innovative Computing, Information and Control
Volume **8**, Number **10(B)**, October **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 7169–7184

REAL-TIME OUTDOOR RENDERING USING HYBRID SHADOW MAPS

Hoshang Kolivand and Mohd Shahrizal Sunar

ViCubeLab, Department of Computer Graphics and Multimedia
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
Skudai, Johor 81310, Malaysia
shahinkey@yahoo.com

Received August 2011; revised January 2012

Abstract. *In outdoor rendering, the sun's position, sky color, clouds, shadows, trees and grass are important factors to consider for effective realism. In this paper, three of these are combined. The sun's position is calculated based on specific longitude, latitude, date and time using Julian date; the sky color is created by Perez model and the shadows are generated using a technique called Hybrid Shadow Mapping (HSM). A new method is used to combine these important factors. Another contribution of this paper is a new algorithm to create shadows with higher quality and with higher frames per second when compared with other algorithms such as Layer Variance Shadow Maps and Cascade Shadow Maps. HSM has been tested to justify an increase in the quality and a decrease in the rendering time in a typical application. In this paper, real-time sky color and shadows, with effect of the sun's position are combined to create a realistic outdoor environment without worrying about components such as sky color and shadow position at different times of the day. Commercial games for outdoor rendering are possible beneficiaries of this technique. It could also be advantageous for teachers of physics who teach about earth orbit, and it could be applied in building and architectural designs.*
Keywords: Shadow mapping, Sky color, Sun's position, Variance shadow, Cascade shadow

1. **Introduction.** The fundamentals of rendering atmospheric phenomena have been known in the fields of atmospheric optics and meteorology for many years, but practical use in computer graphics applications has been limited by the complexity of the rendering problems involved. The techniques for rendering outdoor scenes are different from indoor scenes. The sun and the sky color are the main sources of illumination of outdoor scenes.

The principle of calculating the sun's position has been very well known for a long time. The ancient Egyptians were able to calculate the sun's position many years ago by digging a large hole inside one of the pyramids; just once a year, on the king's birthday, the sun would shine on the grave of their king.

To create realistic environment, shadows are very important as they reveal information about the distances between objects in the entire scene. It is a major factor of 3-D graphics for virtual environments, but unfortunately it is expensive to implement in virtual environments. In computer games, shadows give the gamers feelings that trigger the sense of playing in the real world, resulting in maximum pleasure. Games without shadows are not pleasurable to gamers. Nowadays, gamers have taste of virtuality and their imaginations constantly demand more and more realism.

The motivation for this paper is to create realism outdoor rendering without worrying about components such as sky color and shadow position at different times of the day and different days of the year. Combination of all outdoor rendering can be effective for outdoor rendering game engine [1]. Shadow is a critical component which needs more improvement in terms of both speed and realism [1-5].

Objectives:

- To generate sky color modeling with respect to the sun's position during the daytime in specific location, date and time.
- To propose a new algorithm for shadow generation to enhance the quality of shadows generated in high-density scenes with huge numbers of triangles.
- To develop an outdoor rendering application that keeps the position of the sun, sky color and shadow in specific location, date and time.

Organization: This paper is organized as follows. Overviews of related works in two subsections are presented in Section 2. Section 3 presents both methods and materials used in this study. It is divided into six subsections. In Section 4, a new algorithm is proposed and compared with three other related algorithms. A combination technique for the sun's position, sky color and shadow is presented in Section 5. HSM is extensively discussed in Section 6 and compared with current algorithms. The results obtained from the outdoor rendering application developed are also shown in Section 6. The paper ends with a brief conclusion and some suggestions for future development in Section 7.

2. Related Work.

In this section, a brief overview of the calculation of the sun's position, creating sky color and shadow generation in virtual environments is presented. Real-time shadow is an important area of research as evidenced by the volume of researches that has been done in this area.

2.1. The sun's position and sky color.

The sun's position and the amount of sunshine historically has been a very attractive topic for most researchers. For example, in 1958, Nawar et al. [6] worked on the principle amount of sunshine in a day. Kambezidis et al. [7] provided several functions to calculate the sun's position by focusing on factors such as light refraction and right ascension.

Numerous computer graphics researchers have also tried to simulate the atmospheric effect of the sky. Many of them have simulated the light from the sky and the sun by considering the scattering and absorption of light in the earth's atmosphere. Early work in rendering and modeling atmospheric effects was done by Blinn [8]. Blinn proposed a method of modeling Saturn's rings by using thin layers of clouds and dusty surface. Klassen [9] tried to display sky color by taking into consideration spectral distribution due to particles in the atmosphere. However, this method has problems because the atmosphere is approximated as multiple layers of the plane-parallel atmosphere. However, it is assumed to have uniform density. Thus, the method is different from the actual physical phenomenon. Kaneda et al. [10] improved this method by approximating the actual physical phenomenon by considering the spherical atmosphere with air density changing exponentially with altitude. This work had been extended using multiple scattering by Nishita et al. [11]. Most of the proposed methods can display a realistic sky color but have a time constraint in rendering. Tadamura et al. [12] combined both Kaneda's model and the CIE by discussing the relationship between them in 1993.

Dobashi et al. [13], proposed a fast display method of sky color using basic trigonometric functions. In the proposed method, cosine functions are used as the basic function. The sky color in the view direction of an arbitrary sun position can be obtained from stored distributions and displayed more quickly. The method is tested for natural scenes and

architectural design. The method has a major drawback of not achieving photorealism objectives.

Preetham et al. [14] approached an analytic model for rendering the sky. The image generated looks quite impressive in which an inexpensive analytic sky model from Perez et al. [15] (Perez model) that approximates full spectrum daylight for various atmospheric conditions was modeled. At the same time, an enhanced realism of outdoor rendering was achieved by a model developed for aerial perspective.

Sunar and Gordon [16] created a sky dome to simulate the effect of the sun's position on sky color using the Perez model. Li et al. [17] proposed unified volumes representation for light shaft and shadow, which is an efficient method of simulating natural light shafts and shadows with atmospheric scattering effect. Kolivand et al. [18] tried to produce illuminated 3D objects based upon the effects of interaction between the sunlight and sky. Sunkavalli et al. [19] proposed a model for temporal color changes and explored its use for the analysis of outdoor scenes from time-lapse video data.

2.2. **Shadows.** There are many algorithms to create shadows. Shadow volume and mapping are classical real-time hard techniques to create shadows on arbitrary surfaces. Although shadow volume is accurate enough; it is geometrical and requires more calculations. Shadow mapping is easy to implement being image based. Game makers have used shadow volume with great success. Although it is considered established in gaming industry, it has two expensive phases; one of them is updating volume rendering passes and the other is silhouette detection. This idea was first introduced in 1977 by Crow [20], the method explicitly clips shadow geometry to view the frustum. In 1991, Heidmann [21] published a paper based on shadow volume using stencil buffer, which remains the main shadow volume algorithm to date.

Williams [22] introduced a technique known as shadow maps, which is fast and image based. However, aliasing is a crucial problem of shadow mapping. Reeves et al. [23] proposed a new technique to reduce aliasing by filtering, a technique which is called percentage-closer filtering (PCF). This algorithm, by filtering the depth map with interpolation of binary data, attempts to produce monotonic data in the outline of shadow.

Lokovic and Veach [24] proposed a technique entitled deep shadow maps to reduce aliasing. In contrast to traditional shadow map, that stores one depth for each pixel, deep shadow maps store fractional visibility functions that represent the visibility through a given pixel at all different depth levels.

Fernando et al. [25] resolved shadow mapping aliasing by using a technique to synchronize the size of a pixel in two viewpoints. This is accomplished by storing the shadow map in a hierarchical structure instead of the normal flat two-dimensional view.

Stamminger and Drettakis [26] proposed Perspective Shadow Maps (PSM). PSM in normalized device coordinate space. PSM reduces aliasing and creates sharp outlines as well as standard shadow maps. This idea provides high resolution for near objects and decreasing resolution as the viewer moves away from the viewpoint.

Donnelly and Lauritzen [27] proposed a new algorithm entitled Variance Shadow Maps (VSM) based on shadow mapping to reduce aliasing. In this algorithm, instead of storing a single depth value (z-value) they store mean (σ) and squared of depths distribution (σ^2).

Dimitrov [28] proposed a new technique based on shadow mapping entitled Cascaded Shadow Maps (CSM). Cascaded shadow maps try to reduce aliasing by increasing high resolution for parts of view which are close to the view point and low resolution for other parts of scene that are far away.

Lauritzen and Mccool [29] proposed a beneficial algorithm to solve both VSM problems. They called the algorithm Layered Variance Shadow Maps (LVSM). LVSM divides depth shadow into layers. This technique has some advantages and disadvantages. Although it reduces the texture preciseness, it could solve both problems of VSSM method.

Liu and Pang [30] prepared a very good survey of shadow rendering with attention to projection shadows and shadow volume. In 2010, Yang et al. improved variance shadow maps and published their paper entitled Variance soft shadow mapping (VSSM) [2]. VSSM is based on PCF and exploits recent advances in VSM. VSSM can render high quality soft shadow faster than PCSS. A very good and new survey on shadow mapping can be found in [3] that reveals more important errors of shadow maps.

3. **Materials and Methods.**

In this study, Julian date is used to calculate the sun's position. Perez model is applied to create the sky color because it is suitable to have analytic sky model based on actual atmosphere [15-17]. There are three techniques to generate real-time shadows, Projection Shadow, Shadow Volume and Shadow Mapping that HSM is based on latest one in this study.

3.1. **Sky modeling.**

Latitude is the angular distance between the equator and points north and south of it on the surface of the earth. Longitude is angular distance from the east to the west of the prime meridian, ranging from 0 to180 degrees east to west. Each 15 degrees represents one hour. The angle between earth's orbit and the equator is 23.5 degrees at all times because the angle that the sun can be viewed is not the same at different latitudes. Sunsets and sunrises are produced by earth's rotation. The first place on earth that sees the sunrise each day is Japan. This is the main reason why there are different days of the month, different seasonal months and different seasons of the year across nations. Therefore, the sky is modeled after these parameters.

3.2. **Calculating the sun's position.**

Knowing zenith and azimuth are enough to calculate the position of the sun at any time. To have zenith and azimuth, location, longitude, latitude, date and time are required. Zenith is the angle that indicates the amount of sunrise while the azimuth is the angle that indicates the amount angle that sun turns around the earth.

Iqbal in 1983 proposed a formula to calculate the sun's position [31] and in 1999, Preetham et al. improved on it [14]. It is a common formula to calculate the position of the sun in physics. It can be seen in [18]. With calculation of zenith (θ) and azimuth (ϕ) the sun's position is obvious. To create shadows, the Cartesian coordinates are needed.

3.3. **Effect of sun's position on shadows.**

As the earth moves in its orbit, the sun is seen from earth. Shadow length depends on the position of the sun relative to view position on the earth. When a part of the earth is tilted away from the sun, the sun's position is lower in the sky and shadows are longer. Conversely, when a part of the earth is tilted towards the sun, position of the sun is highest in the sky and the shadows are shorter. Shadows are longest in any day at both sunrise and sunset. The shortest shadows therefore are seen at noon. This explains why some parts of earth see more or less of the sun each day. Measurement of the horizontal shadow angle (HSA) is calculated using the formula:

$$HSA = \rho = \varphi_s - W \tag{1}$$

where W is the orientation of buildings.

The vertical shadow angle (VSA) is calculated by

$$VSA = \sigma = \tan^{-1}\left(\frac{\tan\theta}{\cos\rho}\right) \tag{2}$$

3.4. **Coloring the sky using the CIE XYZ spaces and CIE Yxy spaces.** The XYZ space allows color to be expressed as a mixture of the three tri-stimulus values X, Y and Z. The CIE standard allows a color to be classified as a numeric triple (X, Y, Z). All the CIE-based color spaces are derived from the fundamental XYZ space. CIE XYZ space accepts all colors perceivable by human beings and it is based on experimentally determined color matching functions. Thus, it is a device-independent color space. All visible light can be shown as a positive combination of X, Y and Z. Therefore, Y component is closely associated with the apparent lightness of a color. In general, the mixture of X, Y and Z components that describe a color can be expressed as percentages. Yxy space is another space that determines the XYZ values in terms of x and y chromaticity co-ordinates. To convert XYZ space into Yxy co-ordinates the following formulae are used:

$$Y = Y, \quad x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z} \tag{3}$$

For the other way around, to convert from Yxy co-ordinates to XYZ, the following formulae can be used:

$$X = \frac{x}{y}X, \quad Y = Y, \quad Z = \frac{1-x-y}{y}Y \tag{4}$$

The Z tri-stimulus value is not visible by itself and is combined with the new co-ordinates. In another situation, to convert between the device independence CIE based color space to a device dependent RGB color space in computer graphics, a color transformation matrix is essential. It is useful for mapping the CIE XYZ values to RGB monitor values. Nevertheless, this conversion is not easy. It is because the transformation matrix is dependent upon the behavior of the particular phosphor in a specific monitor. For this reason, creating a general transformation matrix to obtain an accurate color conversion is not possible. Sometimes a CIE XYZ value will give a negative value when converted to a RGB color space, but RGB values cannot be any negative. The negative value is out of the RGB gamut that defines no color. Therefore, a color matching procedure is needed. Hence, some colors that are described in the CIE color space cannot be described in the RGB color space.

3.5. **Perez sky model.** The model is convenient to illuminate arbitrary point (θ_p, γ_p) of the sky dome with respect to the sun's position. It uses CIE standard and could be used for a wide range of atmospheric conditions. Luminance of point (θ_p, γ_p) is calculated using the following formulae:

$$L(\theta_p, \gamma_p) = \left(1 + Ae^{\frac{B}{\cos\theta_p}}\right)\left(1 + Ce^{D\gamma_p} + E\cos^2\gamma_p\right) \tag{5}$$

$$\gamma_p = \cos^{-1}\left(\sin\theta_s \sin\theta_p \cos(\varphi_p - \varphi_s) + \cos\theta_s \cos\theta_p\right) \tag{6}$$

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 received at the earth surface

Essentially, to use Yxy space, the following three components are needed. In each point of view, the Y luminance is calculated by

$$Y = Y_z \frac{L(\theta_p, \gamma_p)}{L(0, \theta_s)} \tag{7}$$

The chromaticity of x and y is calculated by

$$x = x_z \frac{L(\theta_p, \gamma_p)}{L(0, \theta_s)} \tag{8}$$

$$y = y_z \frac{L(\theta_p, \gamma_p)}{L(0, \theta_s)} \tag{9}$$

To color each sky pixel, all of the pixels in the introduced formulas must be calculated iteratively. Involving date and time in specific locations enables the exact color reproduction of each pixel.

3.6. **Skylight distribution coefficients and zenith value.** The zenith values and skylight distribution coefficient are specified in [14]. The precise calculation can be found in [18]. The values of zenith are the functions of turbidity (T) and the sun's position, while a different T value will give a different distribution coefficient. Distribution coefficients for luminance, Y distribution function are computed accurately in [36].

4. **Hybrid Shadow Mapping.** Shadow mapping is a suitable base to implement shadows on arbitrary surfaces. It is easy to load 3ds or ply files in the scene without worrying about silhouette detection such as shadow volume. Three parts of previous algorithms and a new idea to distribute the partitions are combined. Hybrid shadow mapping is constructed based on shadow mapping algorithm. Layer variance shadow map is the second part of the algorithm to create multiple layers for storing each partition in a separate layer. Cascade shadow map is the last part of the new approach, and it distributes resolution in the scene in order to increase the quality of shadow. Logarithm function is selected to distribute layers situation to decrease rendering time. The proposed algorithm is as follows:

Step 1. *Render the whole scene from point of view and store the mean and mean squared of depth distribution.*

Step 2. *Render the whole scene again from the light source point and store the mean and mean squared of depth distribution.*

Step 3. *Split frustum point of view into multiple partitions, depending on the size of the scene using logarithm function starting from the closest object related to the camera point of view.*

View frustum splitting starts from the first object in the scene. This idea allows the GPU to be independent of these parts of the scene which are out of any rendering contribution. This idea, in addition to make the algorithm faster, reduces the number of layers considerably.

The partitions developed by splitting the view frustum using logarithm function are not uniform; some parts of the scene located near the first object are divided into small partitions, whereas other parts located far away do not require much resolution as they are located in large partitions. Moreover, using logarithm function contributes to high speed rendering compared with cascade shadow maps. It should be noted that most of the objects are located around the center of cone in the view frustums. Furthermore,

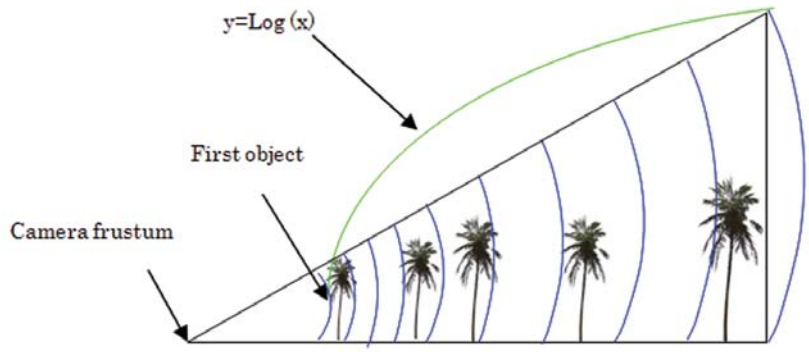

FIGURE 1. Layer separating in hybrid shadow mapping

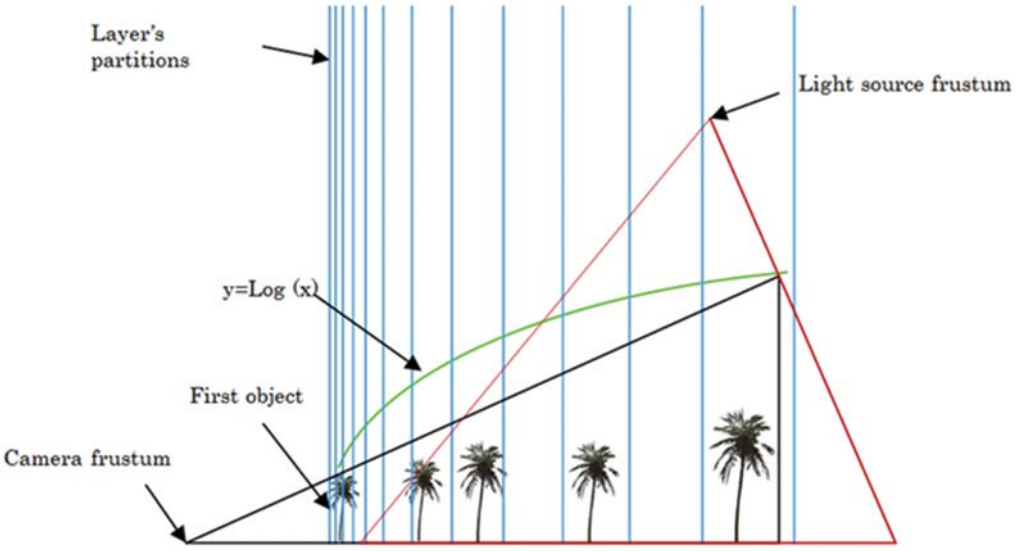

FIGURE 2. Hybrid shadow mapping

curve splitting provides more enhancement in realism of rendering than straight one, as illustrated in Figure 1.

By storing the mean and mean squared of a distribution of depth maps, the variance can be efficiently computed over any filter region to reduce aliasing.

As mentioned in the variance shadow maps, the crucial problem is light bleeding. Generating layers for each partition prevents light bleeding. Choosing the first object to start partitioning results in fewer layers compared with layer variance shadow maps.

Logarithm function (Figure 2) changes the uniform distribution of partitioning. In this case, the resolution of near partitions could be as high as 128 or 64 bits, the middle partitions can have medium resolution of 32 or 16 bits and finally, the third part that is far away from the camera can have lesser resolution of around 8 bit or less.

A brief comparison can be made with respect to frames per second, which is the most important factor to consider when creating shadow in real-time rendering. Variance: Variance shadow mapping is an acceptable technique to decrease aliasing. By comparing the left and right side of Figure 2, improvement of soft shadow can be seen in addition

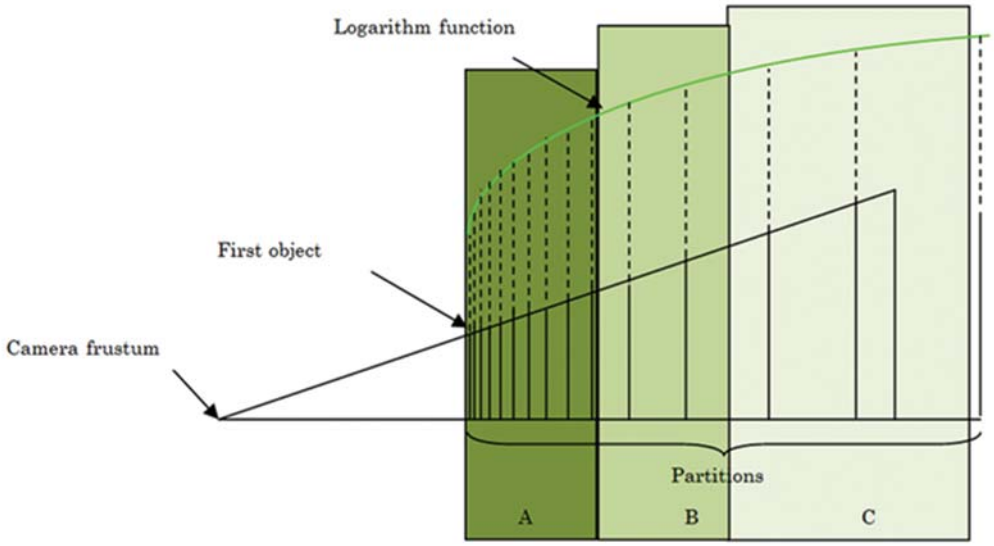

FIGURE 3. Construction of hybrid shadow mapping

to decreasing the aliasing. More discussions about soft shadow and aliasing improvement are provided in next section.

Resolution: Controlling the scene resolution could increase the FPS. If resolution increases, FPS will be decreased. Obviously, when a wide scene is rendered with the same resolution, some parts of the scene located far away from the camera may not be seen very well, wasting the GPUs and CPUs time.

Partitioning: Splitting whole scenes into multiple partitions helps to control the resolution in different parts of a scene. A prominent difference between cascade shadow maps and the new approach is the non-uniform partitions.

Figure 3 reveals some parts of the construction in hybrid shadow mapping. The parts of view frustum which will be used in the new algorithm splits into more partitions, depending on the fluctuation of partitions, divides in three or more classes. The classification is 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 at the middle of the scene, and depends on the size of the scene. Finally, Class C represents those scene parts that are located far away from the camera that are not visible to the viewer. Dispersion index can be used to categorize the scene parts into three classes.

Class A requires high resolution for quality rendering results. 128, 64 bits or at least 32 bits are suitable for this important part. Class B needs less resolution compared with class A. 32 bits or 16 bits is enough for this class depends on the CPU or GPUs ability. Finally, Class C does not need more resolution. Although using this technique can improve rendering quality and FPS, reducing the precise texture is a problem with this idea.

Layering: Layering in the new algorithm prevents light bleeding but when the variance is big, high resolution texturing is needed. The layers distribution based on logarithm functions starts from the first object, resulting in fewer layers in two parts of the scene, which are located near the camera and far away from the camera.

Light bleeding: Although HSM reduces the amount of overlapping, it is equally suffering from this. If N is the number of layers and s_i is i^{th} split location, the upper bound is calculated by Chebychev's inequality. To cover the overlapping of second occluder, instead of $\sigma + \mu$ that was maximization of both depths a and b, $\min(a, b) + \frac{|a-b|}{2}$ can

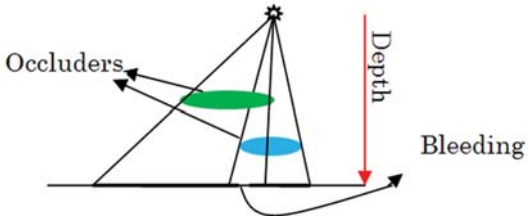

FIGURE 4. Construction of light bleeding

manage the half part of overlapping. Thus, the improvement of overlapping is as follows:

$$[s_i, s_{i+1} + \varepsilon], \quad i = 1, \ldots, N$$

However, although the proposed formula improves the LVSM, that is no guarantee of precise approximation for all situations. Lower bounding of VSM generates bleeding. Upper bounding of LVSM requires more overlapping, while the proposed formula, which can be called medium bounding, decreases the overlapping around 50 percent in general situations. In outdoor rendering that has low depth complexity (compared with indoor rendering) these artifacts are negligible.

5. **Combination of Real-Time Sky Color and Shadows.** Since the earth revolves around its orbit, the relative position of the sun to the earth changes. When a part of the earth is tilted away from the sun, the sun's position is lower in the sky and shadows are longer. Conversely, when a part of the earth is tilted towards the sun, the position of the sun is high in the sky and shadows are shorter. Longer shadows therefore in any day appear at sunrise and sunset, and short shadows appear at noon. This tilt of the earth's axis is the main reason why each part of the earth sees more or less sunlight in a day. Different seasons are the consequence of different length of days and nights.

In addition to shadow, sky color is very effective when making realistic outdoor environments. The combination of sky color and shadow affected by the sun's position can make realistic scenes.

6. **Discussion.** There are two aspects to discuss. The first is Hybrid Shadow Mapping compared with previous algorithms. The second is to evaluate the outdoor rendering application developed.

6.1. **Hybrid shadow discussion.** Although curve splitting makes the number of layers consistent, the layer of some objects may change and consequently an enhancement of realism could be achieved. Co-covering of layers is one of the advantages of this method which does not require more precision. It allows running the program on more current GPU due to less precision texture filtering. Another advantage is reducing the amount of overlapping due to non-linear layering. For example, in a scene with 75m width and 450 length, nearest object 10m and farthest object in 380m with three classes, the number of layers is [0.8 n] compared with n for standard LVSM. This means that if LVSM has 10 layers, HSM requires 8 for same quality but frame rendering time is adversely affected (Table 1).

Optionally, by Gaussian blurring all layers and percentage closer filtering [4], soft outline of shadow could be created, resulting in low frames per second. To solve this problem, applying PCF on Class A can be effective.

Cascade shadow mapping technique suffers a problem at contract shadows at the cost of huge amounts of memory and the high cost of filtering [33]. Logarithm function fixes the

FIGURE 5. Left: CSM using store of mean and variance, Center: HSM using medium bounding, Right: HSM using layer overlapping

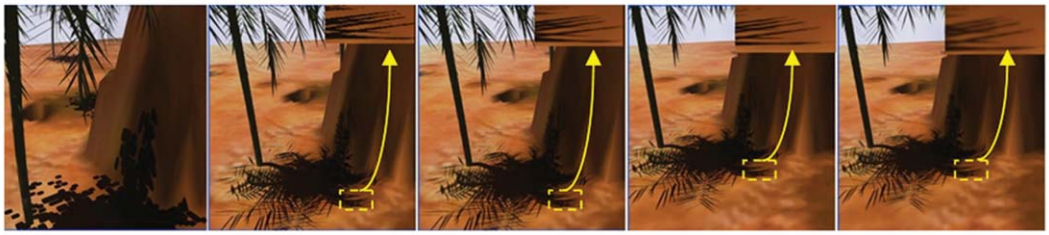

FIGURE 6. Left: SSM, Center-Left: CSM, Center: PCF, Center-Right: LVSM using Gaussian blurring, Right: HSM using Gaussian blurring

TABLE 1. Frame rendering time for a large scene

	$256 * 256$	$512 * 512$	$1024 * 1024$	*Require layers*
VSM	2.9	2.92	3.1	1
$LVSM$	5.5	7	11	15
HSM	5.4	6.3	10.2	12

TABLE 2. Same performance (M coefficient, L layers)

	M	L	FPS
CSM	16	–	52
VSM	–	1	95
$LVSM$	–	8	83
HSM	–	7	87

layer's position without worrying about the number of layers, on the contrary of CSMs, when the number of layers increases.

Figure 6 reveals a comparison between previous algorithms and HSM using Gaussian blurring in $1024 * 1024$ resolutions.

Compared with the new algorithm CSM uses twice the coefficient (M) than does the HSM. This is because CSM stores coefficients in 8 bit while HSM uses 16 bit textures for the same purpose. The result (Table 2) of a scene with about two million triangles is obtained using an AMD Athlon 64 X2 6400+ Windsor 3.2GHz Socket AM2 125W Dual-Core with Graphic Hardware GeForce 7025 NVIDIA nForce 630a. All measurements are taken at a resolution of $1024 * 1024$.

6.2. **Software evaluation.** Sky color, the sun's position and length of shadows in real-time computer games can make a game more realistic [1,10,12,16]. To keep the real

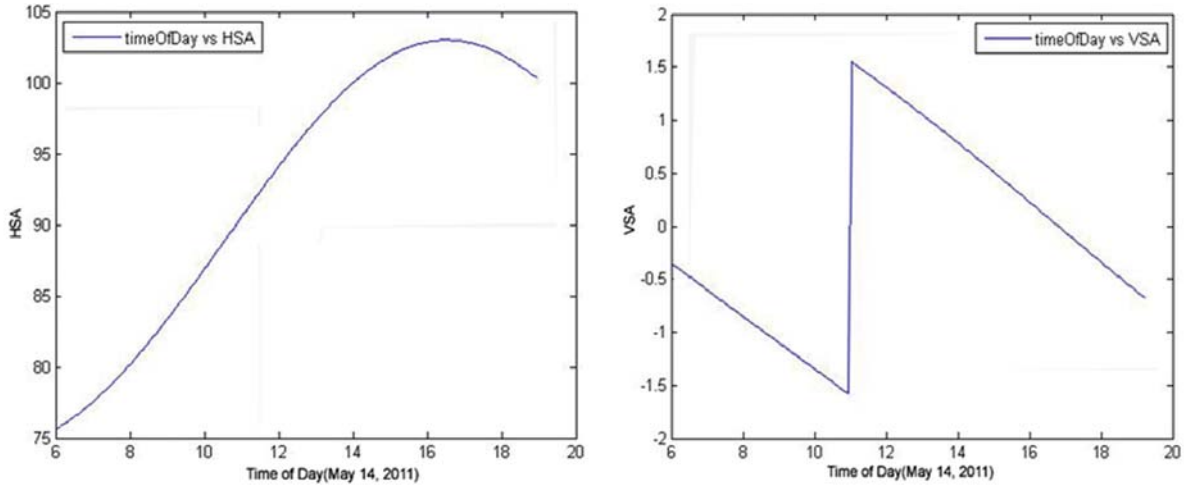

FIGURE 7. HSA and VSA in different time of day (May 14, 2011)

position of the sun and length of shadows in a virtual environment, a substantial number of precise calculations are necessary. For example in building design and architecture, possible recognition of which orientation is best to place a building could be more specific [34].

In cold places, a building needs to remain in a position in which shadows lie at the back of the building; in warm places, building should be located in such a position that shadows lie in front of the building [35].

Over a period of one year, the amount of sunshine in the southern hemisphere is more than the northern hemisphere except for latitudes below than 1.5 degrees. Thus, the length of shadow is less and as a result, the direction of building should not be against the sun.

Vertical shadow angle and horizontal shadow angle on May 14, 2011 in the same place were measured and the result is as follows. Figure 7(left) reveals that HSA changes smoothly like a part of normal distribution. Figure 7(right) reveals that VSA decreases between 6 and 11 sharply and around 11 increases sharply and again decreases to –0.5 at 19. It means that to focus on amount of shadow in reorganization of building design orientation, the time between 11 and 16 is most important. Figure 7(right) shows more vertical angle changing occurs between 10 and 12 around 11. This means that although we should not be worried about amount of shadow, for best direction of building it is important to have less sunshine in warm places and more sunshine in cold places.

With different Julian days from 1 to 365, at different times, the amount of HSA and VSA were measured and the following results obtained. Figure 8 illustrates that the center of the change is between 6:00 to 10:45 and 13:45 to 17:45. Figure 8 is amount of horizontal and vertical shadow angle at different times of day. Figure 8(left) illustrates that the highest VSA is in the first three quarters. Figure 8(right) shows that the highest HSA is in the third and final one-third. They can be used to position the orientation of a building.

Real and software generated data for sunrise and sunset time at different times of year are compared in Figure 9. The real sunrise and sunset are obtained from [32]. Figure 10(left) shows the difference between real sunrises according to the results obtained from the developed software. The graph illustrates that on the first of January, the difference between real sunrise and the software generated result is low and on the first of March and first of September the maximum difference occurs. This difference is approximately

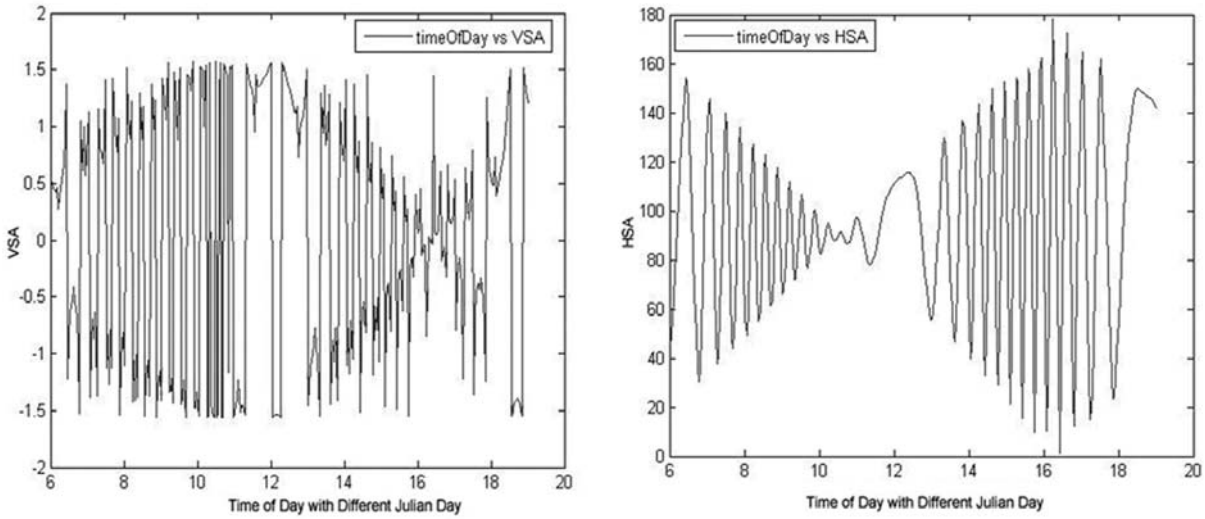

FIGURE 8. HSA and VSA in different Julian day

	January		February		March	
Sunrise real	07:07	07:17	07:17	07:15	07:15	07:06
Sunset real	19:10	19:20	19:20	19:20	19:20	19:13
Sunrise software	07:9	07:02	07:13	07:07	07:04	06:53
Sunset software	19:20	19:23	19:27	19:25	19:20	19:21
	April		May		June	
Sunrise real	07:05	06:57	06:57	06:57	06:57	07:02
Sunset real	19:13	19:07	19:07	19:09	19:09	19:15
Sunrise software	07:13	06:53	06:54	07:01	06:54	06:59
Sunset software	19:22	19:24	19:23	19:26	19:25	19:30
	July		August		September	
Sunrise real	07:02	07:06	07:06	07:01	07:01	06:52
Sunset real	19:15	19:17	19:17	19:10	19:09	18:58
Sunrise software	07:02	07:00	07:01	06:53	06:50	06:42
Sunset software	19:27	19:21	19:18	19:10	19:06	19:02
	October		November		December	
Sunrise real	06:52	06:47	06:47	06:52	06:53	07:07
Sunset real	18:58	18:51	18:50	18:55	18:55	19:09
Sunrise software	06:42	06:44	06:45	06:55	06:57	07:09
Sunset software	19:01	18:59	19:00	19:03	19:10	19:19

FIGURE 9. Real and software generated data sets for both sunrise and sunset at UTM on 14 May, 2011

10 minutes with the minimum being on the first of July. These differences, for the last day of each month, are almost opposite. The maximum difference occurred on the last day of January and the minimum difference on the last day of December.

Figure 10(right) shows these differences based on sunset. The initial effect of the graph on the right is harmonic minimum and maximum for the first and last days. Evidently,

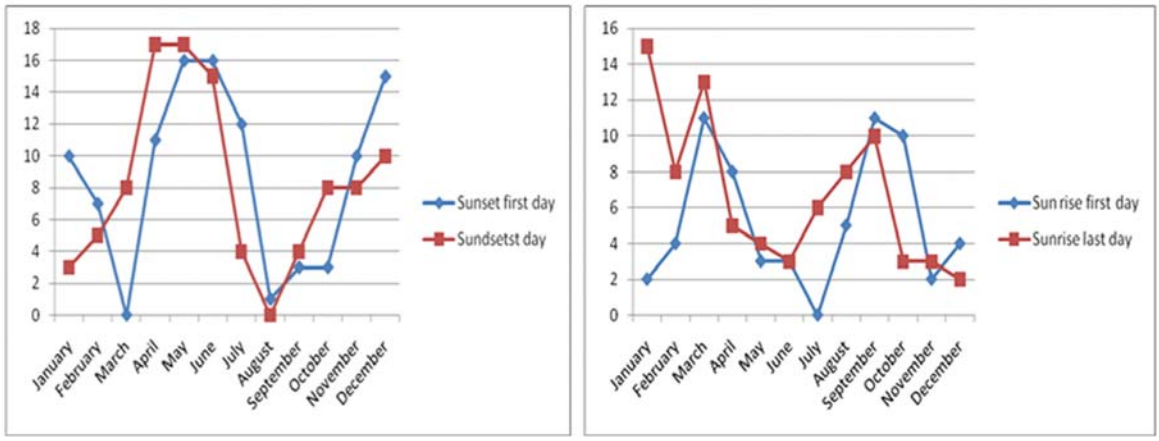

FIGURE 10. Different real sunrise and sunset and results of our application

the minimum difference is 0 and the maximum difference is 17 minutes on the last day of April and May respectively.

Figure 11 shows the effect of the sun's position on sky color and shadow on the first of January in UTM at latitude of 1.28 and longitude of 103.45 at different times of the day.

The sky color changes with the position of the sun during the day and even at night. In nature, image quality varies every minute. The changes are not only in terms of sun angle and intensity of radiation, which are different every time, but it is affected by changes in the color of light and contrast. At dawn, when the sun is not yet over the horizon, the sky along the horizon will be golden shiny and bright purple. When the sun can be seen over the horizon, the sky becomes yellow and then an attractive blue color.

These differences could help to rely on the formulae. To improve the formulae, approximation with real value could be applied where there is greatest difference. Combination of sky color and shadow in real-time environment is done successfully using the above technique. This software, in addition to helping game makers generate outdoor games without worrying about shadow position and sky color at different times of day and different day of year, also makes it possible for teachers of physics to teach about earth's orbit and effect of the sun on shadows.

7. **Conclusion and Future Work.** Shadows and the sky color as determined by the sun's position are the most important factors to consider when creating realism in outdoor environments. The main contribution of this paper is to develop an application for keeping daytime sky color and shadow with effect of the sun's position. Thus, three objectives were stated in order to achieve the goal. First, the dome sky, which was modeled using mathematical functions, is a better way to represent the sky. The algorithm not only supports a triangular mesh, but also a quadratic mesh. The sun's position needs to be calculated by Julian. Then, the color of the sky had to be carried out using Perez model which yields the result in CIE Yxy space. Secondly, the new algorithm to improve the shadow generation based on shadow mapping increases quality and FPS compared with cascade shadow mapping and layer variance shadow mapping. Finally, by a combination of sky color and shadows with respect to the sun's position, an outdoor rendering software to keep the three components simultaneously is generated.

A model that gives the sky color and hybrid shadow mapping according to the sun's position and the specific location, date and time has been developed. The application was developed using C++ as programming language and OpenGL for the interface.

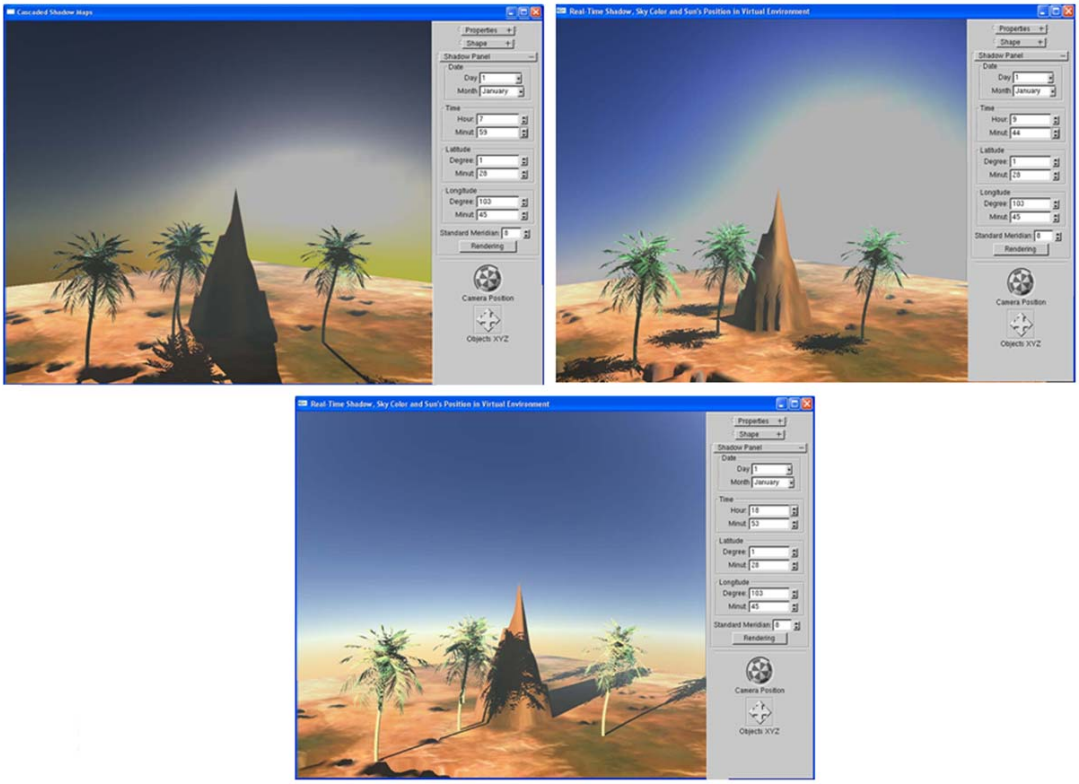

FIGURE 11. Hybrid shadow mapping (1024 ∗ 1024), January 1st at Universiti Teknologi Malaysia at a latitude of 1.28 and longitude of 103.45 at different times of day

There are some improvements that could be made to this application in the near future. It is possible that the application be modeled not only with the sky but also with other natural phenomena such as cloud, landscape and the ocean. Soft shadow is another natural phenomenon that can make this application more complete and realistic. We hope that this application could be a good platform for extension to outdoor rendering and photorealism.

Acknowledgment. This research was supported by Vot. J13000.7282.4F085 FRGS grant at the UTMVicubeLab, Department of Computer Graphics and Multimedia, Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia. Thanks to Dr. Folorunso Olufemi for good suggestions during the revision of this paper.

REFERENCES

[1] K. Boulanger, *Real-Time Realistic Rendering of Nature Scenes with Dynamic Lighting*, University of Rennes I, France, 2008.
[2] B. Yang, Z. Dong, J. Feng, H. P. Seidel and J. Kautz, Variance soft shadow mapping, *Computer Graphics Forum*, vol.29, no.7, pp.2127-2134, 2010.
[3] D. Scherzer, M. Wimmer and W. Purgathofer, A survey of real-time hard shadow mapping method, *Computer Graphics Forum*, vol.30, no.1, pp.169-186, 2011.
[4] Z. Noh, M. S. Sunar and Z. Pan, A review on augmented reality for virtual heritage system, *Lecture Notes in Computer Science*, vol.5670, pp.50-61, 2009.
[5] H. Kolivand and M. S. Sunar, Silhouette detection for real-time shadow in virtual environments, *The International Journal of Virtual Reality*, vol.10, no.4, pp.45-51, 2011.

[6] S. Nawar, A. B. Morcos and J. S. Mikhail, Photoelectric study of the sky brightness along sun's meridian during the March 29, *New Astronomy*, vol.12, pp.562-568, 2007.
[7] H. D. Kambezidis, D. N Asimakopoulos and C. G. Helmis, Wake measurements behind a horizontal-axis 50 Kw wind turbine, *Solar Wind Tech.*, vol.7, pp.177-184, 1990.
[8] J. F. Blinn, Light refection functions for simulation of clouds and dusty surfaces, *ACM Transactions on Graphics*, vol.16, pp.21-29, 1982.
[9] R.V. Klassen, Modeling the effect of the atmosphere on light, *ACM Transactions on Graphics*, vol.6, pp.215-237, 1987.
[10] K. Kaneda, T. Okamoto, E. Nakamae and T. Nishita, Photorealistic image synthesis for outdoor scenery under various atmospheric conditions, *The Visual Computer*, vol.7, pp.247-258, 1991.
[11] T. Nishita, E. Nakamae and Y. Dobashi, Display of clouds and snow taking into account multiple anisotropic scattering and sky light, *Proc. of SIGGRAPH'96, Annual Conference Series*, LA, USA, pp.379-386, 1996.
[12] K. Tadamura, E. Nakamae, K. Kaneda, M. Baba, H. Yamashita and T. Nishita, Modeling of skylight and rendering of outdoor scenes, *Computer Graphics Forum*, vol.12, pp.189-200, 1993.
[13] Y. Dobashi, T. Nishita, K. Kaneda and H. Yamashita, A fast display method of sky colour using basis functions, *The Journal of Visualization and Computer Animation*, vol.8, pp.115-127, 1997.
[14] A. J. Preetham, P. Shirley and B. Smith, A practical analytic model for daylight, *Proc. of SIGGRAPH'99, Computer Graphics*, pp.91-100, 1999.
[15] R. Perez, R. Seals and J. Michalsky, All-weather model for sky luminance distribution – Preliminary configuration and validation, *Solar Energy*, vol.50, pp.235-245, 1993.
[16] M. S. Sunar and N. A. Gordon, Real-time daylight sky color modeling, *Advances in Computer Graphics and Virtual Environment*, pp.37-55, 2001.
[17] S. Li, G. Wang and E. Wu, Unified volumes for light shaft and shadow with scattering, *IEEE*, pp.161-166, 2007.
[18] H. Kolivand, A. Amirshakarami and M. S. Sunar, Real-time projection shadow with respect to sun's position in virtual environments, *International Journal of Computer Science Issues*, vol.8, no.6(3), pp.80-84, 2011.
[19] K. Sunkavalli, W. Matusik, H. Pfister and S. Rusinkiewicz, Factored time-lapse video, *ACM Transactions on Graphics*, vol.26, pp.110, 2007.
[20] F. Crow, Shadow algorithms for computer graphics, *Computer Graphics*, vol.11, pp.242-247, 1977.
[21] T. Heidmann, Real shadows real time, *IRIS Universe*, vol.18, pp.28-31, 1991.
[22] L. Williams, Casting curved shadows on curved surfaces, *SIGGRAPH'78*, vol.12, no.3, pp.270-274, 1978.
[23] W. T Reeves, D. H. Salesin and P. L. Cook, Rendering antialiased shadows with depth maps, *Proc. of SIGGRAPH'87, Computer Graphics*, vol.21, no.4, pp.557-562, 1987.
[24] T. Lokovic and E. Veach, Deep shadow maps, *Proc. of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pp.385-392, 2000.
[25] R. Fernando, S. Fernadez, K. Bala and D. P. Greenberh, Adaptive shadow maps, *Proc. of ACM SIGGRAPH*, pp.387-390, 2001.
[26] M. Stamminger and G. Drettakis, Perspective shadow maps, *Proc. of SIGGRAPH*, 2002.
[27] W. Donnelly and A. Lauritzen, Variance shadow maps, *Proc. of the 2006 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp.161-165, 2006.
[28] R. Dimitrov, *Cascaded Shadow Maps, VNIDIA*, 2007.
[29] A. Lauritzen and M. Mccool, Layered variance shadow maps, *Proc. of Graphics Interface 2008 Toronto, Ontario, Canada, Canadian Information Processing Society*, pp.139-146, 2008.
[30] N. Liu and M. Y. Pang, A survey of shadow rendering algorithms: Projection shadows and shadow volumes, *The 2nd International Workshop on Computer Science and Engineering*, pp.488-492, 2009.
[31] M. Iqbal, *An Introduction to Solar Radiation*, Academic Press, 1983.
[32] *http://www.gaisma.com/en/location/alor-setar.html.*
[33] T. Annen, T. Mertens, H. P. Seidel, E. Flerackers and J. Kautz, Exponential shadow maps, *Proc. of Graphics Interface 2008*, pp.155-161, 2008.
[34] B. Maringka and D. W. Utomo, The effect of shadow to building envelops towards thermal performance in apartments at tropical areas, *International Conference on Engineering, Environment, Economics, Safety and Health*, 2009.
[35] M. R. Emmanuel, An urban approach to climate-sensitive design strategies for the tropics, *Taylor*, Francis, 2005.

[36] H. Kolivand and M. S. Sunar, Real-time sky color with effect of sun's position, *International Journal of Scientific and Engineering Research*, vol.2, no.7, pp.2229-5518, 2011.