A Fast Rendering Method for Shafts of Light in Outdoor Scene

Yoshitaka Moro † Ryo Miyazaki ‡ Yoshinori Dobashi * Tomoyuki Nishita †

†The University of Tokyo, ‡Pixela Corporation, * Hokkaido University
{ymoro,nis}@nis-lab.is.s.u-tokyo.ac.jp, ryo.miyazaki@pixela.co.jp, doba@nis-ei.eng.hokudai.ac.jp

Abstract

Realistic rendering methods of natural phenomena in real-time have a variety of applications, such as flight simulators or computer games. The scattering effect due to atmospheric particles is one of the most important elements in creating realistic outdoor images. Shafts of light are caused by objects which shut out the sunlight, such as clouds and mountains. This paper proposes a method for creating outdoor images including shafts of light caused by mountains and clouds. In our method, shafts of light are displayed by subtracting the intensity of the shielded light. Our method is accelerated by using the fragment shader, which is a function of the latest graphics hardware.

Keywords: Rendering, Natural Phenomena, Atmospheric Scattering, Graphics Hardware, Clouds, Sky, Shafts of Light

1. Introduction

In the field of computer graphics, rendering realistic images based on physical phenomena is one of the most important research subjects. Light passing through the media, such as the air and clouds, is scattered by small particles. This phenomenon is called atmospheric scattering, which is one of the most important elements in creating realistic outdoor images. The air contains two kinds of particles of different sizes, air molecules and aerosols. Air molecules and aerosols obey Rayleigh scattering and Mie scattering theories, respectively. Clouds consist of particles such as water droplets, which obey Mie scattering theory. These two theories correspond to the phase functions with different properties. The phase function of Rayleigh scattering is proportional to power of -4 of the wavelength of light. Mie scattering is strong forward scattering and little dependent on the wavelength. There are many studies of rendering photorealistic images of clouds [19].

When objects, such as clouds and mountains, shut out part of the sun's rays, we can observe dark volumes compared to surroundings in the atmosphere. This effect is called shafts of light. Several methods of calculating shafts of light have been proposed [10]. However, these methods are kinds of ray tracing and take hours to create images.

Methods of using graphics hardware for interactive rendering of the sky and clouds have been proposed [4, 5, 17]. In these methods, shafts of light are rendered by generating sampling planes in the view volume and by rendering the sampling planes and accumulating the intensity of the atmospheric scattering using hardware color blending functions.

This paper proposes a method for creating outdoor images including shafts of light caused by shielding objects (i.e., mountains) and non-uniform translucidus media (i.e., clouds). Our method improves the method proposed by Dobashi et al [4, 5]. In these methods, numbers of the sampling planes are generated in the entire view volume, resulting in increase in the rendering time. In general, the shielded volume of mountains and clouds is far smaller than the entire view volume. Therefore, the rendering time can be reduced by generating the sampling planes in only the shielded volume. Based on this idea, our method displays shafts of light by firstly rendering the atmospheric scattering effects without taking into account shadows of objects and then by subtracting the intensity of the atmospheric scattering corresponding to the shielded volume.

2. Releted Work

There were a number of methods of generating images by taking into account scattering or absorption of light in participating media, such as the sky or clouds [1, 3, 12, 13, 21, 23, 24, 27]. Nishita et al. take into account multiple scattering for rendering the sky [18] or clouds [19]. Studies of rendering shafts of light have also been developed [10, 11, 14, 15, 20, 22, 25]. However, most of these methods use a ray tracing algorithm or scan-line algorithm. The computational costs of these methods are expensive.

To accelerate the calculation, several methods of fast rendering of clouds or shafts of light using graphics hardware have been proposed. Dobashi et al. proposed a fast method for rendering clouds with shafts of light [4]. In this method, the intensity of clouds is calculated by using metaballs.

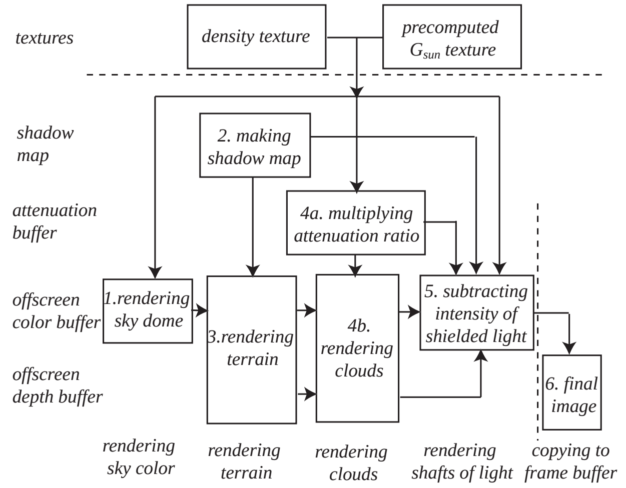


Figure 1. Flowchart of our method. The shadow map and the attenuation buffer are also used for calculating shafts of light.

To render shafts of light, multi spherical shells are generated. Harris and Lastra developed real-time cloud rendering method [9]. This method require two passes in intensity calculation of clouds and ignore light scattering due to atmospheric particles. Dobashi et al. proposed a fast method of rendering shafts of light [7, 6, 8]. In these methods, shafts of light are rendered by generating sampling slices in the view volume. However, aliasing is appeared bacause frame buffer has only 8 bit accuracy. To resolve this problem, Dobashi et al. improve these methods by generating sub-planes to sample the shielded volume precisely [5]. Miyazaki et al. proposed an interactive method for rendering clouds with shafts of light using sampling slices facing mid-way between the sun and the view directions [17]. Sun et al. proposed a real-time rendering method of single scattering of light [26]. However, this method do not take into account shadows of object and cannot render shafts of light.

In previous methods, shafts of light are rendered by accumulating the intensity of scattered light. However, to render outdoor images, a number of sampling slices are required to calculate shafts of light caused by clouds or mountains. In our method, sampling slices are only placed in the shielded volume and the improvement of the computational time can be expected.

3. Overview of Our Method

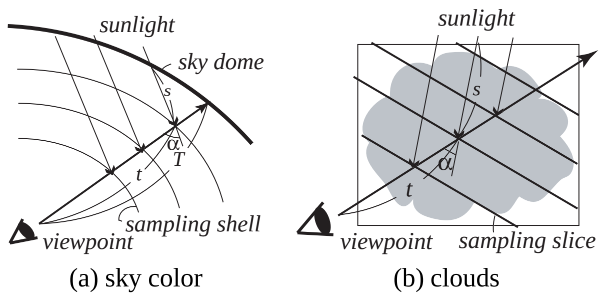
We propose a method for calculating shafts of light by both clouds and mountains. Fig.1 shows the flowchart of our method. Numbers in the figure show the processing order. In our method, the intensity is calculated in the fragment shader and stored in the 16 bit floating point offscreen buffer. The attenuation ratio of the intensity of light reaching the scattering point, g_{sun}, is precomputed and stored as two-dimensional texture. The other required data, such as the density of atmospheric particles, are also stored as textures. First, sky color is rendered as a hemispherical dome with a large radius (Step 1 in Fig.1). The color of the dome (i.e., sky color) is calculated when the sun altitude or the height of the viewpoint is modified. In the previous method, the intensity of sky color is calculated by generating virtual shells and by accumulating the intensity of scattered light stored in the virtual shells [5]. Recently, the fragment shader, which is a function of the latest graphics hardware, is available. In the fragment shader of the latest version, the iterative calculation is possible. In our method, the intensity of scattered light is accumulated in the fragment shader. Next, the terrain is rendered. The *shadow map*, which stores the depth from the light source, is also created in this step (Steps 2 and 3 in Fig.1). Then, clouds are rendered by using a slice-based volume rendering technique [17] (Steps 4a and 4b in Fig.1). In this step, the *attenuation buffer*, which stores the attenuation ratio of the intensity by water droplets in clouds, are also created. When rendering the terrain and the clouds, the atmospheric effects (scattering and absorption) are taken into account but shadows of the terrain and the clouds are not considered. After that, shafts of light are displayed by subtracting the intensity of sunlight shielded by the terrain or the clouds (Step 5 in Fig.1). To calculate the intensity of light reaching the shielded volume, sampling slices that look towards both the viewpoint and the sun are placed. The intensity of light attenuated or shielded by objects in the sampling slice is calculated by making use of the shadow map and the attenuation buffer created in the step of rendering clouds and the terrain. Finally, the image is copied from the offscreen buffer to the frame buffer (Step 6 in Fig.1).

4. Intensity Calculation of Sky Color and Clouds

The intensity of light reaching the viewpoint, I_{view}, is given by the following equation:

$$I_{view}(T,\lambda) \quad = \quad I_s(\lambda)\int_0^T g_{sun}(s,\lambda)R(t,\alpha,\lambda)g_{view}(t,\lambda), \qquad (1)$$

where I_s is the intensity of sunlight, g_{sun} is the attenuation ratio of the intensity of light reaching the scattering point from the sun, g_{view} is the attenuation ratio between the scattering point and the viewpoint, λ is the wavelength, t is the distance from the viewpoint to a point on the viewing ray, α is the phase angle (see $alpha$ in Fig.2), s is the distance from the scattering point on the viewing ray to a point



(a) sky color

(b) clouds

Figure 2. Intensity calculation of sky color and clouds.

on the sunlight ray, T is the distance between the viewpoint and the top of the atmosphere.

g_{sun}, g_{view}, and R are calculated by following equations:

$$g_{sun}(s,\lambda) \quad = \quad exp(-\int_0^s (k_r(\lambda)\rho_r(s) + k_m(\lambda)\rho_m(s))ds), \tag{2}$$

$$g_{view}(t,\lambda) \quad = \quad exp(-\int_0^t (k_r(\lambda)\rho_r(t) + k_m(\lambda)\rho_m(t))dt), \tag{3}$$

$$R(t,\alpha,\lambda) \quad = \quad F_r(\alpha)(k_r(\lambda)\rho_r(t) + F_m(\alpha)k_m(\lambda)\rho_m(t), \tag{4}$$

where k is the scattering coefficient, ρ is the density of participating media, F is the phase function. Subscripts "r" and "m" of F, k and τ represent Rayleigh scattering and Mie scattering, respectively. For the phase function of Mie scattering, we use the improved Henyey-Greenstein function described by Cornette [2]. The phase functions of Rayleigh scattering and Mie scattering are given by the following equations:

$$F_r(\theta) \quad = \quad \frac{3}{16\pi}(1 + cos\theta), \tag{5}$$

$$F_{mie}(\theta) \quad = \quad \frac{3(1-g^2)}{8\pi(2+g^2)}\frac{1+\cos^2\theta}{(1+g^2-2g\cos\theta)^{3/2}}. \tag{6}$$

4.1. Rendering the sky

The sky is considered as the hemispherical sky dome whose center is at the viewpoint (see Fig. 2(a)). When rendering sky color, I_{view} is calculated by taking into account both Rayleigh scattering and Mie scattering. The densities of atmospheric particles, air molecules and aerosols, are considered as depending on the altitude. Therefore, the density distribution of atmospheric particles is stored as an one-dimensional texture. The attenuation ratio g_{sun} is precomputed and stored in a two-dimensional texture with the height and the sun altitude at that point. The intensity of the sky I_{view} and the attenuation ratio g_{view} are calculated iteratively in the same pass by using a fragment shader.

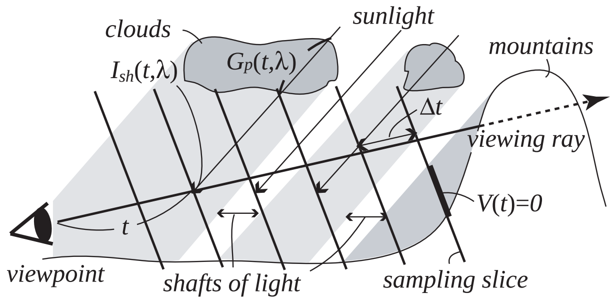


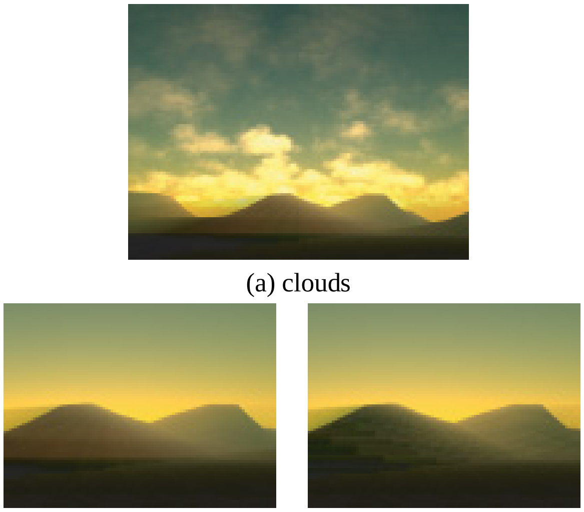
Figure 3. Sampling slices covering shadows of clouds and mountains.

4.2. Rendering clouds

Clouds are calculated by taking into account only Mie scattering due to water droplets. The density distribution of water droplets (i.e., clouds) is non-uniform. The distribution is stored as volume data using three-dimensional grids. The volume data is used as a 3D texture. The clouds are rendered by using a slice-based volume rendering technique [17]. An attenuation buffer is prepared to store the attenuation ratio between the sun and each points on the sampling slice, $G_p(t,\lambda)$. The intensity on each sampling slice is multiplied by the attenuation ratio stored in the attenuation buffer and the resulting values are accumulated in the frame buffer (see Fig. 2(b)).

5. Rendering Shafts of Light

Shafts of light can be observed when the sunlight is shielded by mountains or clouds. Dobashi et al. generated sampling planes in the view volume for rendering shafts of light [5]. However, this approach requires a long computation time since many sampling planes are required to cover the entire view volume. The rendering time is proportional to the number of the sampling planes. We address this problem as follows. In general, the shielded volume is far smaller than the entire view volume. The rendering time can be reduced by generating the sampling planes in only the shielded volume. Based on this idea, shafts of light are rendered in two steps. First, the atmospheric scattering effects are calculated without taking into account shadows of objects. Second, the intensity of the atmospheric scattering corresponding to the shielded volume are subtracted. Sampling slices facing mid-way between the sun and the view directions are generated to cover the shielded volume. When we do not take into account the shadows of ob-


(a) clouds

(b) $n = 83$

(c) $n = 32$

Figure 4. Result images. n is the number of sampling slices.

jects, the atmospheric scattering effects are computed analytically. Therefore, in this case, the rendering time for the atmospheric scattering effects is very short and negligible. Therefore, even in the worst case where the view volume is entirely in the shielded volume, the rendering time of our approach is the same as that of Dobashi's approach. The volume shielded by mountains is obtained by creating a shadow map of the terrain. Shafts of light caused by clouds are computed by making use of the attenuation ratio of clouds, $G_p(t, \lambda)$, calculated in the cloud rendering step. The intensity of shielded sunlight at the intersection point of each sampling slice and the viewing ray, I_{sh}, is given by the following equation:

$$I_{sh}(t_k, \lambda) \quad = \quad (1 - G_p(t_k, \lambda)V(t_k))I_v(t_k, \lambda)\Delta t_k, \quad (7)$$

where k is the sampling slice number, V is the visibility function which returns 0 if the sampling point is shielded by the mountains, otherwise returns 1. The intensity on each sampling slice without considering shadows of objects, $I_v(t_k, \lambda)$, is computed by assuming the constant density of atmospheric particles. Δt_k is the interval of the sampling slices along the viewing ray (see Fig. 3).

6. Results

We used a machine with an Intel Pentium 4 3.6GHz and an nVidia Geforce 6800 GT to measure rendering time. The number of the sampling slices of the clouds is 83. To calculate the intensity of clouds, we use the density data of water droplets generated by using Miyazaki's method [16]. Fig. 4(a) shows outdoor images including clouds, the sky and shafts of light. Figs. 4(b)(c) show images of shafts of light by using the different number of sampling slices without rendering clouds. Calculating cost of Fig. 4 are 0.3 fps, 11.4 fps, and 21.2 fps, respectively. In Fig. 4(c), artifacts due to the insufficient number of slices are shown. Therefore, we use 83 sampling slices to create other images including clouds.

7. Conclusion

In this paper, we have proposed an interactive rendering method for shafts of light. Our method enables us to render shafts of light caused by clouds and mountains. Sampling slices are generated only in the shielded volume, which is far smaller than the entire view volume in general. Therefore, our rendering method achieves interactive speed by reducing the number of sampling slices.

In future work, we discuss the problem prosed by the multiple scattering of light. In our method, the multiple scattering effect is approximated as a constant term. However, for participating media with a high albedo, such as clouds, multiple scattering effect is important. An efficient method for the multiple scattering has to be developed.

References

[1] J. F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. volume 16, pages 21–29, July 1982.

[2] W. M. Cornette and J. G. Shanks. Physical reasonable analytic expression for the single scattering phase function, 1992.

[3] Y. Dobashi, K. Kaneda, H. Yamashita, and T. Nishita. Skylight for interior lighting design, 1994.

[4] Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, and T. Nishita. A simple efficient method for realistic animation of clouds. In *SIGGRAPH'00*, pages 19–28, 2000.

[5] Y. Dobashi, T. Nishita, and T. Yamamoto. Interactive rendering of atmospheric scattering effects using graphics hardware. In *Graphics Hardware'02*, pages 99–108, 2002.

[6] Y. Dobashi, T. Okita, and T. Nishita. Interactive rendering of shafts of light using a hardware-accelerated volume rendering technique. In *SIGGRAPH'00 Technical Sketches*, page 219, 2000.

[7] Y. Dobashi, T. Yamamoto, and T. Nishita. Interactive rendering method for displaying shafts of light. In *Pacific Graphics'00*, pages 31–37, 2000.

[8] Y. Dobashi, T. Yamamoto, and T. Nishita. An accurate, fast method using graphics hardware for rendering shafts of light, 2001.

[9] M. J. Harris and A. Lastra. Real-time cloud rendering. In A. Chalmers and T.-M. Rhyne, editors, *EG 2001 Proceedings*, volume 20(3) of *Computer Graphics Forum*, pages 76–84. Blackwell Publishing, 2001.

[10] H. W. Jansen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *SIGGRAPH'98*, pages 311–320, 1998.

[11] J. T. Kajiya and B. P. V. Herzen. Ray tracing volume densities. In *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, volume 18, pages 165–174, July 1984.

[12] K. Kaneda, T. Okamoto, E. Nakamae, and T. Nishita. Highly realistic visual simulation of outdoor scene under various atmospheric conditions. In *CG International'90*, pages 117–131, 1990.

[13] R. V. Klassen. Modeling the effect of the atmosphere on light, 1987.

[14] N. L. Max. Atmospheric illumination and shadows. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 117–24, August 1986.

[15] N. L. Max. Light diffusion through clouds and haze. *Computer Vision, Graphics and Image Processing*, 33(3):280–292, March 1986.

[16] R. Miyazaki, Y. Dobashi, and T. Nishita. Simulation of cumuliform clouds based on computational fluid dynamics. In *EUROGRAPHICS 2002, Short Presentation*, pages 405–410.

[17] R. Miyazaki, Y. Dobashi, and T. Nishita. A fast rendering method of clouds using shadow-view slices. In *CGIM 2004*, pages 93–98, 2004.

[18] T. Nishita, Y. Dobashi, K. Kaneda, and H. Yamashita. Display method of the sky color taking into account multiple scattering. In *Pacific Graphics'96*, pages 117–132, 1996.

[19] T. Nishita, Y. Dobashi, and E. Nakamae. Display of clouds taking into account multiple anisotropic scattering and sky light. In *SIGGRAPH'96*, pages 379–386, 1996.

[20] T. Nishita, Y. Miyawaki, and E. Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources, 1987.

[21] T. Nishita and E. Nakamae. Continuous tone representation of three-dimensional objects illuminated by sky light, 1986-8.

[22] T. Nishita and E. Nakamae. Method of displaying optical effects within water using accumulation buffer. In *Proceedings of SIGGRAPH '94*, pages 373–381, July 1994.

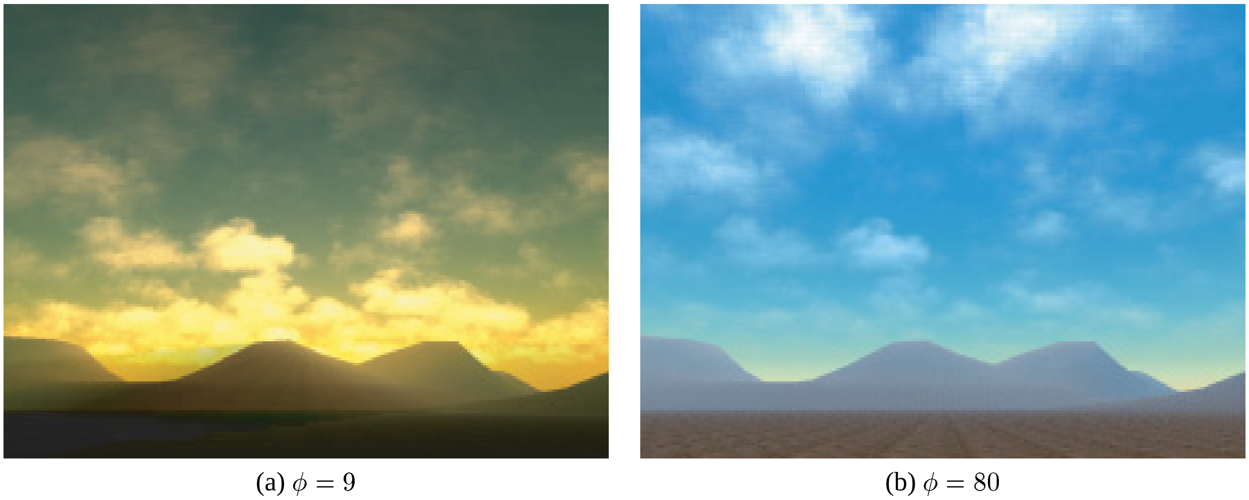
[23] T. Nishita, T. Shirai, K. Tadamura, and E. Nakamae. Display of the earth taking into account atmospheric scattering. In *SIGGRAPH'93*, pages 175–182, 1993.

[24] A. Preetham, P. Shirley, and B.Smits. A practical analytic model for daylight. In *SIGGRAPH'99*, pages 91–199, 1999.

[25] H. E. Rushmeier and K. E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium, 1987.

[26] B. Sun, R. Ramamoorthi, S. G. Narasimhan, and S. K. Nayar. A practical analytic single scattering model for real time rendering. volume 24, pages 1040–1049, 2005.

[27] K. Tadamura, E. Nakamae, K. Kaneda, M. Baba, H. Yamashita, and T. Nishita. Modeling of skylight and rendering of outdoor scenes, 1993.



(a) $\phi = 9$ (b) $\phi = 80$

Figure 5. Outdoor images in different sun altitude.

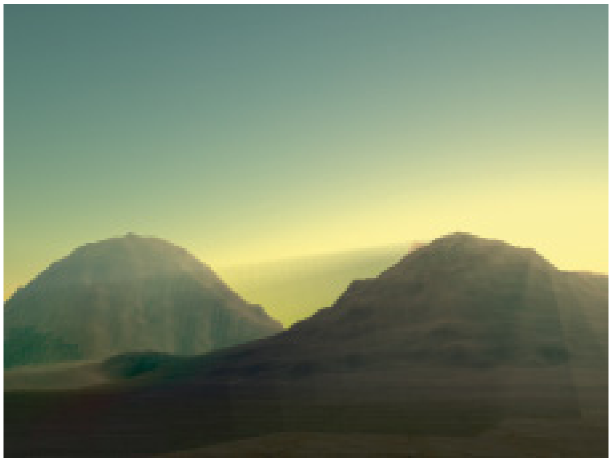
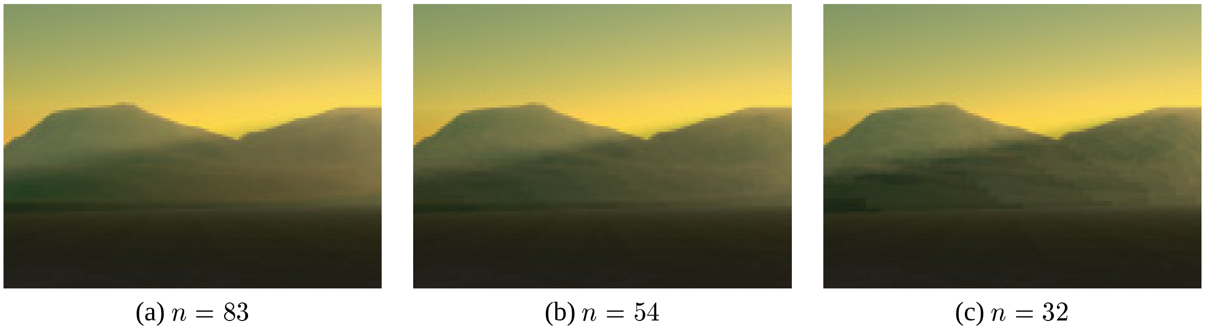


Figure 6. Shafts of light.



(a) $n = 83$ (b) $n = 54$ (c) $n = 32$

Figure 7. Shafts of light created by the different number of shadow slices n.