The Visual Computer manuscript No.
(will be inserted by the editor)

A Fast Rendering Method for Clouds Illuminated by Lightning Taking into Account Multiple Scattering

Yoshinori Dobashi[1], Yoshihiro Enjyo[1], Tsuyoshi Yamamoto[1], Tomoyuki Nishita[2]

[1] Graduate School of Information Science and Technology,
Hokkaido University
Tel.: +81-11-7066530, Fax: +81-11-7168427,
e-mail: {doba, enjyo, yamamoto}@nis-ei.eng.hokudai.ac.jp

[2] Graduate School of Frontier Sciences,
The University of Tokyo
Tel.: +81-4-71363942, Fax: +81-4-71363943,
e-mail: nis@is.s.u-tokyo.ac.jp

Received: date / Revised version: date

Abstract Methods for rendering natural scenes are used in many applications such as virtual reality, computer games, and flight simulators. In this paper, we focus on the rendering of outdoor scenes that include clouds and lightning. In such scenes, the intensity at a point in the clouds has to be calculated by taking into account the illumination due to lightning. The multiple scattering of light inside clouds is an important factor when creating realistic images. However, the computation of multiple scattering is very time-consuming. To address this problem, this paper proposes a fast method for rendering clouds that are illuminated by lightning. The proposed method consists of two processes. First, basis intensities are prepared in a preprocess step. The basis intensities are the intensities at points in the clouds that are illuminated by a set of point light sources. In this precomputation, both the direct light and also indirect light (i.e., multiple scattering) are taken into account. In the rendering process, the intensities of clouds are calculated in real-time by using the weighted sum of the basis intensities. A further increase in speed is achieved by using a wavelet transformation. Our method achieves the real-time rendering of realistic clouds illuminated by lightning.

Key words Real-time Rendering – Clouds – Lightning – Multiple Scattering

1 Introduction

In computer graphics, many research activities have been devoted to the simulation of various natural phenomena such as clouds, water, fire, and terrain. The results of this research have been applied to flight simulators, computer games, visual assessment of outdoor scenes, and so on. Most of the previous methods focused on simulations produced under fine weather conditions. However, simulation under bad weather conditions is also an important factor for some scenarios in the applications mentioned above. Important elements for simulations under bad weather conditions include windstorms, rain, snow, and lightning, and various methods have been developed for simulating some of these conditions [1][2][3][4]. In this paper, we focus on scenes that include lightning and clouds. Several methods have been proposed for the simulation and rendering of lightning [5][6][7][8]. A method that takes into account the effects of light scattering due to clouds and atmospheric particles when they are illuminated by lightning has also been proposed [2]. By using these methods, realistic images can be generated. However, the previous methods do not take into account one important factor that is necessary for creating realistic images, that is, the multiple scattering of light inside the clouds. Furthermore, the computational cost of the previous methods is too high to achieve real-time rendering.

This paper proposes a fast method for rendering lightning and clouds. The intensities of clouds illuminated by lightning are calculated by taking into account the multiple scattering of light. The shape of the lightning is generated by using the method proposed in [5]. We calculate the intensities of the clouds by approximating the lightning as a collection of point light sources. To achieve real-time performance, the proposed method deconstructs the process for the intensity calculation into a preprocess and a rendering process. First, we subdivide the simulation space into a three-dimensional grid, and a virtual point light source is placed at each grid point.

Then, in the preprocess, the intensities of clouds illuminated by each virtual point source are calculated while taking into account the multiple scattering of light. The results are stored as *the basis intensities*. In the rendering process, the intensities of the clouds are calculated very quickly by taking a weighted sum of the basis intensities. The rendering process is further accelerated by using a wavelet transform. By using the proposed method, realistic images including lightning and clouds can be generated in real-time.

The paper is organized as follows. Section 2 discusses previous work that is related to the proposed method. In Section 3, an intensity calculation for clouds that takes into account multiple scattering is explained. Section 4 describes the basic ideas behind the proposed method. Next, Section 5 describes a method for the subdivision of the simulation space that is suitable for making intensity calculations of clouds illuminated by lightning. Section 6 explains the precomputation process, that is, the computation of the basis intensities. Next, Section 7 describes the rendering process for the fast calculation of the intensities of clouds by using the basis intensities. Section 8 demonstrates the usefulness of the proposed method by showing several examples. Finally, Section 9 concludes this paper.

2 Previous Work

The first rendering method that focused on lightning was proposed by Reed and Wyvill in 1994 [5]. After that, Kruszenwski proposed a probabilistic method for modeling the shape of the lightning [6]. This method allows the user to control the shape of lightning. More recently, Kim and Lin proposed a physical based method for the animation of lightning by simulating the propagation process of the lightning [7]. Matsuyama et al. proposed a real-time method for generating and rendering lightning strokes based on the physical phenomena [8]. However, one of the major aims in these papers is to model the shape of the lightning, so these methods do not take into account clouds that are illuminated by the lightning. Clouds are very important in creating realistic images of outdoor scenes that include lightning, since the lightning generally appears from within clouds.

There are many methods for rendering images that takes into account scattering effects due to participating media such as clouds and smoke [9][10][11][12][13][14]. However, these methods do not take into account lightning, since they focus on the rendering of the participating media when illuminated by natural light (sunlight and skylight) and/or by artificial light (such as a spotlight). Dobashi et al. proposed an efficient method for rendering clouds and the atmosphere when illuminated by lightning [2]. However, in this method, the multiple scattering that occurs inside clouds is not taken into account. In addition, the rendering speed of the method is not fast enough to create images in real-time.

For rendering glossy objects in real-time, Sloan et al. proposed a method called "precomputed radiance transfer" [15]. This method is applicable to the rendering of clouds while taking multiple scattering into account. However, in this method, information about the lighting is provided in the form of an environmental map. This implies that the method assumes that the light sources are sufficiently far from the objects. In our case, the lightning (i.e. the light source) is very close to the clouds. Therefore, we cannot apply the precomputed transfer method directly to the rendering of clouds illuminated by lightning. Kristensen et al. addressed this problem by precomputing global illumination due to a set of point light sources [16]. The method can display a scene in real-time while taking into account interreflections due to moving point light sources. The method presented in [17] can also handle the presence of light sources near to objects. This method precomputes the radiance transfer from direct illumination to indirect illumination. However, these methods do not aim at the rendering of participating media such as clouds. In addition, the direct illumination must be calculated with only a small amount of computational cost in order to achieve real-time performance. However, in computing the intensities of clouds illuminated by lightning, the computational cost is expensive, even for the case where there is only direct light.

Methods that use graphics hardware to accelerate the computation of multiple scattering have also been proposed [18][19]. However, the light source in these methods is usually only sunlight, and lightning is not taken into account. The computation times that are required for these methods become longer according to the number of light sources that are present.

The method proposed in this paper achieves the real-time rendering of clouds illuminated by lightning. Our method can be considered as an extension of the method described in [16] to the rendering of clouds illuminated by lightning

3 Intensity Calculation for Clouds Illuminated by Lightning

In this section, we describe a method for calculating the intensity of clouds that takes multiple scattering into account. Table 1 includes the notations of the terms used in this section. Note that the notations of other important terms used in the subsequent sections are also included in Table 1.

As shown in Figure 1, the lightning strokes are represented by a set of line segments and are generated by using Reed's method [5]. The intensity of the lightning is specified by the user. The intensities of the lightning strokes are decreased as they branch. The simplest way to compute the intensities of clouds illuminated by lightning is to approximate the lightning as a collection of

Table 1 Notation of important terms in the paper.

symbol	description
b_{iq}	intensity at grid point i for clouds illuminated by point source q
$\hat{b}_{il}$	basis intensity, that is, intensity at grid point i for clouds illuminated by virtual point source placed at grid point l for simulation space
$\hat{B}_{il}$	wavelet transform of $\hat{b}_{il}$
L_q	intensity of point source q
e_l	intensity of virtual point source l
I_i	total intensity of light scattered at grid point i for clouds illuminated by lightning
n_{cld}	number of grid points for clouds
n_{lgt}	number of grid points for simulation space
n_{dir}	number of sampling directions for directional intensity distribution at each grid point for clouds

point light sources [2], so we employed this approach. Point light sources are generated along the strokes of the lightning, as shown in Figure 1. The intensities of the point sources are proportional to the intensities of the lightning strokes. The intensities of clouds are obtained by calculating the illuminations due to these point sources. The density distribution of the clouds is represented using a uniform three-dimensional grid. The interval between the grid points is Δh.

First, let us consider the intensity of light scattered in direction $\mathbf{s}$, $b_{iq}(\mathbf{s})$, at grid point i in clouds illuminated by point source q on the lightning stroke. b_{iq} is calculated approximately by solving the following linear system of equations (see Figure 1).

$$\begin{aligned} b_{iq}(\mathbf{s}) &= \alpha L_q \rho_i F(\mathbf{s}\cdot\mathbf{s}_{iq}) g_{iq}/r_{iq}^2 \\ &+ \alpha\rho_i \sum_{\substack{j=1\\ j\neq i}}^{n_{cld}} b_{jq}(\mathbf{s}_{ij}) F(\mathbf{s}\cdot\mathbf{s}_{ij}) g_{ij} \Delta h^3/r_{ij}^2, \quad (1) \\ &(i = 1, 2, \cdots, n_{cld}) \end{aligned}$$

where n_{cld} is the number of grid points, α is the albedo of the cloud particles, L_q is the intensity of point source q, ρ_i is the density at grid point i, F is the phase function of the cloud particles, $\mathbf{s}_{iq}$ is a unit vector from point source q to grid point i, g_{iq} is the extinction of light due to cloud particles between point source q and grid point i, and r_{iq} is the distance between point source q and grid point i (see Figure 1). To avoid confusion, let us clarify the meanings of the indices here. Index q is a ID number for the point source on the lightning strokes. Indices i and j indicate ID numbers for grid points of the cloud volume. Although the clouds are represented by a three-dimensional grid, we use a single index to refer to each grid point. We use the Henyey-Greenstein function for the phase function F (see [12]). The extinction, g_{iq}, is

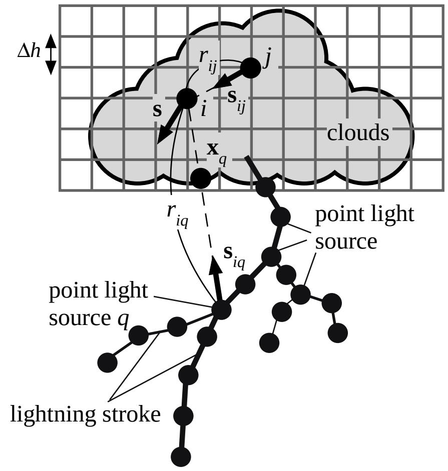


Fig. 1 Intensity calculation for clouds illuminated by lightning.

given by the following equation.

$$g_{iq} = \exp\left(-\kappa \int_0^{|\mathbf{x}_i-\mathbf{x}_q|} \rho(\mathbf{x}_q + t\mathbf{s}_{iq})dt\right), \qquad (2)$$

where κ is an extinction coefficient, $\mathbf{x}_q$ is a point where the light intersects with the clouds, $\mathbf{x}_i$ is the position at grid point i.

The total intensity at grid point i is obtained by summing the intensities due to all of the point sources on the lightning strokes; that is,

$$I_i(\mathbf{s}) = \sum_{q=1}^{n_{ps}} b_{iq}(\mathbf{s}), \qquad (3)$$

where n_{ps} is the number of point sources on the lightning strokes.

To obtain the directional intensity distribution at each grid point, a linear system of equations (Eq. 1) has to be solved numerically. To solve Eq. 1, the extinction term g (Eq. 2) also needs to be calculated. Although g can be precomputed to accelerate the computation, as proposed in [18], solving Eq. 1 still requires high computational cost. So, it is very difficult to compute the intensities of clouds in real-time by directly solving Eq. 1.

4 Basic Concept of Our Method

Figure 2 shows the basic concept behind the proposed method. Note that Figure 2 shows a two-dimensional case for simplicity. As described in the previous section, computing the intensity while taking into account the multiple scattering of light is very time-consuming. Therefore, we deconstruct the computation into two processes: a preprocess and a rendering process. The details of these process are described in the following.

First, in the preprocess, the simulation space is subdivided into a grid and a virtual point light source is

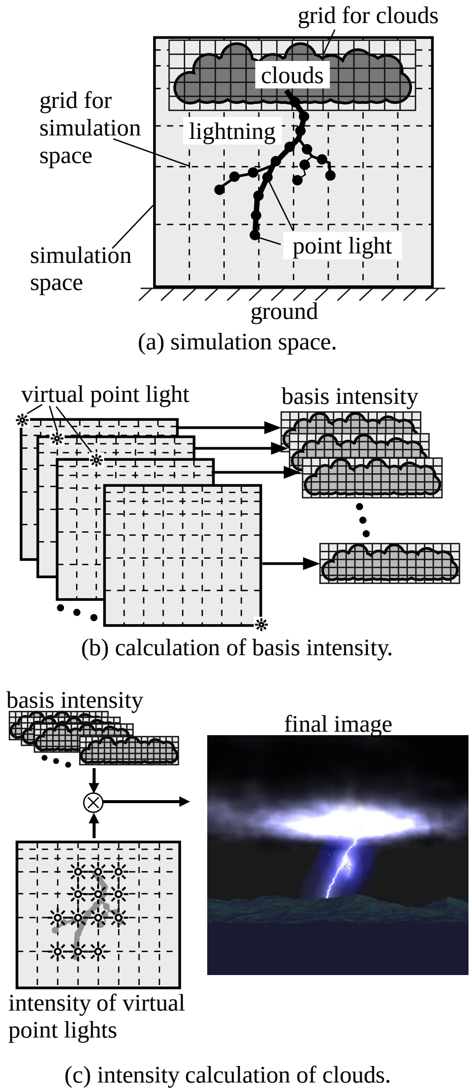


Fig. 2 Basic concept of the proposed method.

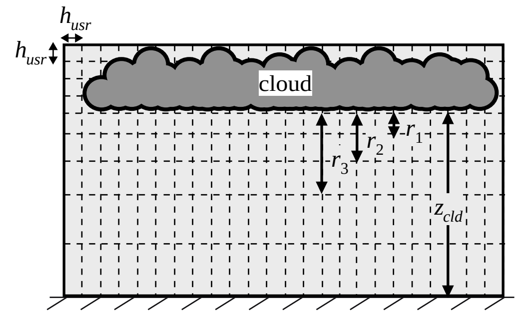


Fig. 3 Grid for simulation space.

placed at each grid point (see Figures 2(a) and (b)). We subdivide the simulation space such that the vertical intervals between the grid points become longer according to their distances from the clouds. This is because the illumination due to the virtual point source decreases rapidly according to the distance from the virtual point source. Then, the intensities of the clouds are calculated taking into account the multiple scattering due to each virtual point source, and the results are stored as basis intensities. In this preprocess, the intensity of a virtual point source is assumed to be 1.0.

Next, in the rendering process, lightning is generated first by using the Reed's method [5]. Next, the intensity of each virtual point source is calculated according to the intensities of the lightning strokes. This is achieved by interpolating the intensities of point light sources placed along the lightning strokes. Then, as shown in Figure 2(c), the intensities of the clouds are obtained simply by computing the weighted sum of the basis intensities. The intensities of the virtual point sources are used as the relative weights. This approach is very efficient because the computationally-expensive calculation (i.e. solving Eq. 1) has already been completed during the preprocess stage. However, the computation of the weighted sum may still take a long time when the number of the grid points for clouds and the grid points for the simulation space are large. Therefore, we use a wavelet transform technique to accelerate the computation. After the intensity at each grid point of the clouds has been obtained, clouds are rendered by using a hardware-accelerated volume rendering technique [20].

5 Subdivision of Simulation Space

As described in the previous section, the simulation space is subdivided into the grid. The intensities of clouds illuminated by a virtual point source decrease in proportion to the inverse of the square of the distance between the clouds and the virtual point source. This indicates that virtual point sources far from the clouds do not contribute very much to the intensities of clouds. Therefore, a small number of the virtual point sources are sufficient in the region far from the clouds. On the other hand, many virtual point sources are required in the region near to the clouds. Based on this idea, we subdivide the simulation space adaptively according to the contributions of the virtual point sources to the cloud intensities. The details are described in the following.

First, the user specifies a base interval h_{usr}. As shown in Figure 3, the simulation space is subdivided uniformly in the horizontal direction at an interval of h_{usr}. The vertical intervals are determined as follows. Let us denote the height of the bottom of the clouds as z_{cld}. The region above z_{cld} is subdivided uniformly using the base interval h_{usr}. In the region below z_{cld}, grid points are generated at the maximum intervals that satisfy the following equation.

$$\frac{1}{r_l^2} - \frac{1}{r_{l+1}^2} < \epsilon \quad (l = 1, \cdots, n), \tag{4}$$

where r_l is the distance from the bottom of the clouds to each grid point (see Figure 3), and n is the number of the grid points in the vertical direction. In addition to the above condition, we assume $r_{l+1} - r_l > h_{usr}$ in order to prevent the number of the grid points becoming too large. By using Eq. 4, we can generate grid points such that the energy of light reaching the clouds from each virtual light source becomes almost equal.

6 Calculation of Basis Intensity

The basis intensity $\hat{b}_{il}(\mathbf{s})$ is the intensity at grid point i for clouds illuminated by a virtual point source placed at grid point l in the simulation space. $\hat{b}_{il}(\mathbf{s})$ is obtained by solving Eq. 1. The directional distribution $b_{il}(\mathbf{s})$ is represented by sampling the direction $\mathbf{s}$ with a specified number of directions, n_{dir}. In the following, let us denote the intensity in mth sampling direction as $\hat{b}_{il}^{(m)} (m = 1, 2, \cdots, n_{dir})$.

We used the method proposed in [10] to compute $\hat{b}_{il}^{(m)}$. After the computation, a set of the intensities ($\hat{b}_{1l}^{(m)}$, $\hat{b}_{2l}^{(m)}$, $\cdots$,$\hat{b}_{n_{cld},l}^{(m)}$) are obtained. Their wavelet transforms, ($\hat{B}_{1l}^{(m)}$, $\hat{B}_{2l}^{(m)}$, $\cdots$,$\hat{B}_{n_{cld},l}^{(m)}$), are stored as the basis intensities. We use the Haar function for the wavelet basis [21]. $\hat{b}_{il}^{(m)}$ is represented by the following equation.

$$\hat{b}_{il}^{(m)} = \sum_{k=1}^{n_{cld}} \hat{B}_{kl}^{(m)} \phi_k(\mathbf{x}_i), \tag{5}$$

where $\phi_k(\mathbf{x}_i)$ represents the Haar basis function and $\mathbf{x}_i$ is the position of grid point i for the clouds.

The intensity distribution of clouds illuminated by a virtual point source is generally smooth. In this case, there are many coefficients $\hat{B}_{il}^{(m)}$ that are nearly zero. These coefficients make little contribution to the resulting intensities of the clouds. Therefore, we discard those coefficients whose magnitudes are smaller than a user-specified threshold ξ. To do this, the coefficients are sorted in descending order of $|\hat{B}_{il}^{(m)}|$. Here, let us denote the index of each coefficient as $\sigma(i)$ after the sorting process, that is, $|\hat{B}_{\sigma(1)l}^{(m)}| > |\hat{B}_{\sigma(2)l}^{(m)}| > \cdots |\hat{B}_{\sigma(n_{cld})l}^{(m)}|$. Then, $\hat{B}_{\sigma(i)l}^{(m)}(|\hat{B}_{\sigma(i)l}^{(m)}| > \xi)$ is stored together with its index $\sigma(i)$. The number $n_l^{(m)}$ of the coefficients that satisfy $|\hat{B}_{\sigma(i)l}^{(m)}| > \xi$ is also stored. Furthermore, we store the root of the sum of the square of the discarded coefficients, $\gamma_l^{(m)} = \sqrt{\sum_{i=n_l^{(m)}+1}^{n_{cld}} |\hat{B}_{\sigma(i)l}^{(m)}|^2}$. This term, $\gamma_l^{(m)}$, is used to compensate the approximation error induced by discarding the small coefficients. That is, $\gamma_l^{(m)}$ is added as an ambient term in the rendering process.

By using the wavelet transform, the memory requirement for storing the basis intensities is dramatically reduced. Furthermore, the intensity calculation in the rendering process described in the next section is also accelerated.

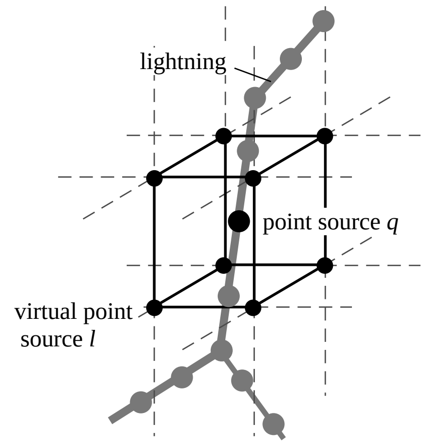


Fig. 4 Intensity calculation of virtual point sources.

7 Real-time Intensity Calculation for Clouds

After the lightning is generated, the intensities of the virtual point sources are calculated according to the intensities of the lightning strokes. Then, the intensities of the clouds are calculated. Since the basis intensities are stored after the wavelet transform, the final result is obtained by using the inverse wavelet transform. The details of this process are described in the following.

The intensities of the virtual point sources at each grid point for the simulation space are calculated as follows. For each point source q on the lightning strokes, we first find the eight neighboring grid points, as shown in Figure 4. Then the intensity of point source q is distributed to the virtual point sources at the eight grid points with appropriate weights. We use the weighting function that is used for trilinear interpolation to calculate the weights in order to distribute the intensity of point source q. These processes are repeated for all of the point sources on the lightning strokes. Consequently, the intensity e_l of virtual point source l is expressed by the following equation.

$$e_l = \sum_{q \in \Theta_l}^{n_q} w_{lq} L_q, \tag{6}$$

where Θ_l indicates a set of point sources affecting virtual point source l, w_{lq} is the weight representing the contribution of point source q to virtual point source l, n_q is the number of point sources whose associated weights w_{lq} are non-zero, and L_q is the intensity of point source q. By using this method, no energy is lost because the sum of the intensities of all of the virtual point sources is equal to the sum of the intensities of all of the point sources on the lightning strokes.

Next, the intensity in mth sampling direction at grid point i for clouds, $I_i^{(m)}$, is calculated by using the weighted sum of the basis intensities, that is,

$$I_i^{(m)} = \sum_{l=1}^{n_{lgt}} e_l \hat{b}_{il}^{(m)}, \quad (i = 1, 2, \cdots, n_{cld}) \tag{7}$$

```
CalculateIntensity( ) {
    Calculate intensities of virtual sources, e_l;
    for(m = 1; m <= n_dir; m++) {
        for(k = 1; k <= n_cld; k++)  {H_k^(m) = 0.0;}
        for(l = 1; l <= n_lgt; l++) {
            for(k = 1; k <= n_l^(m) ; k++) {
                H^(m)_σ(k) = H^(m)_σ(k) + e_l B^(m)_σ(k)l ;
            }
        }
        (I_1^(m), I_2^(m), ..., I_ncld) = IWT(H_1^(m), H_2^(m), ..., H^(m)_ncld);
    }
}
```

Fig. 5 A pseudo code for calculating $I_i^{(m)}$. IWT is a function for the inverse wavelet transform.

where n_{lgt} is the number of virtual point sources whose intensities are non-zero. The basis intensity $\hat{b}_{il}^{(m)}$ is expressed by using the Haar basis function as shown in Eq. 5. However, we discarded the coefficients that satisfies $|\hat{B}_{il}^{(m)}| \leq \xi$. Therefore, $I_i^{(m)}$ is expressed as:

$$I_i^{(m)} = \sum_{l=1}^{n_{lgt}} e_l \Big(\sum_{k=1}^{n_l^{(m)}} \hat{B}_{\sigma(k)l}^{(m)} \phi_{\sigma(k)}(\mathbf{x}_i) \Big) = \sum_{k=1}^{n_l^{(m)}} H_k^{(m)} \phi_{\sigma(k)}(\mathbf{x}_i), \quad (8)$$

where $H_k^{(m)}$ is given by:

$$H_k^{(m)} = \sum_{l=1}^{n_{lgt}} e_l \hat{B}_{\sigma(k)l}^{(m)} \ . \quad (9)$$

Eqs. 8 and 9 indicate that we can obtain the intensities of clouds $I_i^{(m)}$ by first calculating $H_k^{(m)}$ and then inversely transforming them. Figure 5 shows a pseudo code for computing $I_i^{(m)}$ using Eqs. 8 and 9. After computing $I_i^{(m)}$, we add the following ambient term, $I_{amb}^{(m)}$, calculated by using $\gamma_l^{(m)}$ described in the previous section.

$$I_{amb}^{(m)} = \sum_{l=1}^{n_{lgt}} e_l \gamma_l^{(m)}. \quad (10)$$

When we compute the intensities of clouds by using Eq. 7, Eq. 7 has to be calculated n_{cld} (number of grid points for the clouds) times. That is, the computational cost is proportional to $n_{cld} \times n_{lgt}$. However, the computational cost incurred by using Eqs. 8 and 9 is considerably less, since $n_l^{(m)}$ in Eq. 8 (the number of $\hat{B}_l^{(m)}$ stored as the basis intensity) is far smaller than n_{cld}. The computational cost for the inverse wavelet transform is small. In our typical experiments shown in the next section, the computation time was reduced to half of the time without the above method. As a result, the proposed method allows us to render the clouds in real-time.

8 Examples

In this section, the usefulness of the proposed method is demonstrated by examining several examples. In the following examples, the density distribution of the clouds is generated by using the method presented in [14]. The number of grid points for the clouds is 128×128×16. However, for computing intensities of clouds (both the direct light and multiple scattering), we use the density distribution downsampled to 32×32×4 due to the limitation of the memory required for storing the intensity distribution at each grid point. The number of the sampling directions, n_{dir}, is 26. The simulation space is subdivided into 20×20×7. The computer that we used was a desktop PC with a Pentium IV 3.5 GHz (CPU) and an nVidia GeForce 8800GTX (GPU). Most of the computation is carried out by the CPU, while the GPU is only used for displaying the final result. The precomputation time for the basis intensities took 60 minutes. The memory requirement for the uncompressed basis intensities is proportional to (number of grid points for clouds)×(the number of grid points for simulation space)×(number of sampling directions). We store the basis intensities with a floating point precision, that is, four bytes are required to store a single scalar value. Therefore, in this particular example, the memory requirement for the uncompressed basis intensities is (32×32×4) × (20×20×7) × 26 × 4 bytes = 1137.5 MB. This data is compressed by using the Haar wavelets. The threshold ξ for the compression was 5.0×10^{-10}. Then the size is reduced to 86 MB. This indicates that compression to 1/13 of the original size has been achieved without degrading the quality of the final images, as shown in the following examples.

First, we verified the effectiveness of the proposed method by using a simple example, as shown in Figure 6. In Figure 6(a), the clouds are rendered by taking into account the single scattering only. Figure 6(b) shows a cross section of the clouds in Figure 6(a). The intensities are converted into pseudo colors. Figure 6(c) shows the same scene but with multiple scattering taken into account. Figure 6(d) shows the corresponding cross section image. To create Figures 6(b) and (d), the intensities are normalized so that the maximum intensity corresponds to 1.0. As shown in these images, by taking into account the multiple scattering, the clouds become brighter and the reality of the image is improved. Especially, Figures 6(b) and (d) indicate the importance of the multiple scattering. We created the same image as Figure 6(c) by using one of the previous methods [10]. This method can also calculate multiple scattering very efficiently. Figure 6(e) is the image using the previous method. Figure 6(f) is a difference image between Figures 6(c) and 5(e). In Figure 6(f), the difference is normalized so that 10 % corresponds to white, where 100 % means that the difference is 255. As shown in these images, the proposed method can create an image with almost the same accuracy as the previous method [10].

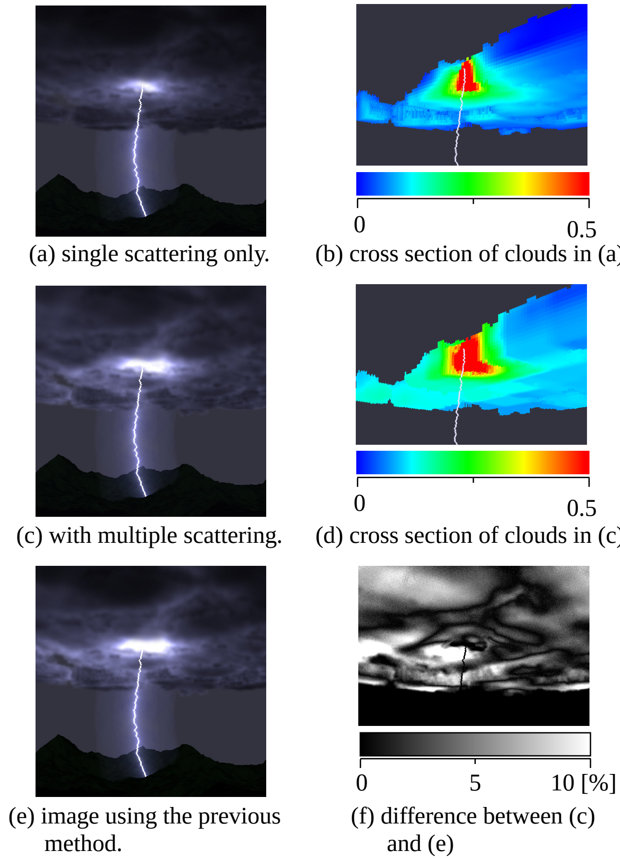
Table 2 Comparison of computation times [seconds].

figure	proposed	previous
6(b)	0.06	2.12
7(a)	0.06	2.10
7(b)	0.06	2.76
7(c)	0.06	2.88
7(d)	0.06	3.44

In these images, the glow around the lightning is rendered by drawing texture-mapped quadrilaterals along the lightning strokes. The texture represents the atmospheric scattering due to lightning and is created by using the method described in [2].

Figure 7 shows some more complex examples, including various types of lightning. The lightning strokes are generated randomly from several seed points positioned inside the clouds. There are two (Figures 7(a) and (b)), three (Figure 7(c)), and five (Figure 7(d)) seed points. There are different numbers, shapes, and colors of lightning strokes. Realistic images including lightning and clouds can be generated, as shown in these images.

The computation times for the examples shown in this section are summarized in Table 2. Table 2 shows the times that were required for computing the intensities at all of the grid points for the clouds. We did not include the time for volumetric rendering of the clouds in order to verify the effectiveness of the proposed method. As shown in this table, the proposed method can render an image between 30 and 60 times faster than the previous method [10]. As we mentioned before, the proposed method requires a fairly long precomputation time, one hour in this example. However, once the precomputation has finished, clouds illuminated by arbitrary shapes of lightning strokes are rendered very quickly. Images with arbitrary viewpoints can also be rendered in real-time. Due to these benefits, we believe that the proposed method is worth spending the long precomputation. Furthermore, even if the number of lightning strokes is increased, the computation time required by the proposed method does not increase. This is because only the calculation of intensities for the virtual point sources (see Section 7) is proportional to the number of lightning strokes and its cost is very small, while the computation time required for the previous method is higher because the computational cost for the direct light from the lighting strokes is increased according to the number of lightning strokes that are included.



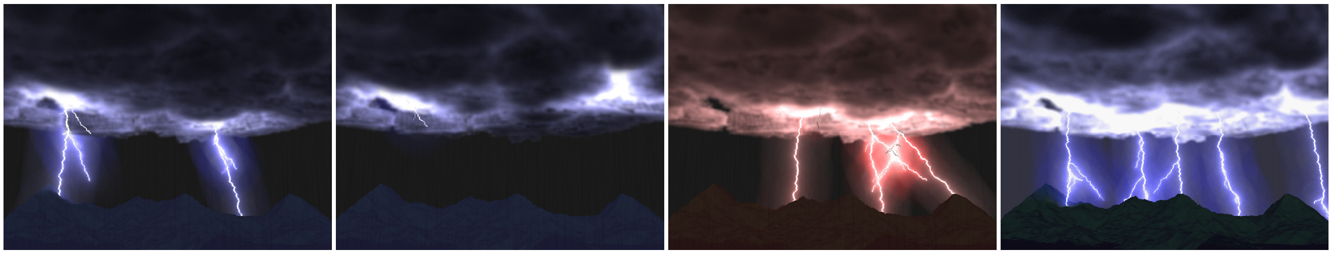
(a) single scattering only. (b) cross section of clouds in (a).
(c) with multiple scattering. (d) cross section of clouds in (c).
(e) image using the previous method. (f) difference between (c) and (e)

Fig. 6 Clouds illuminated by a single lightning stroke.

9 Conclusions

In this paper, we have proposed a fast method for calculating the intensities of clouds illuminated by lightning. The proposed method takes into account the multiple scattering of light inside the clouds. In the method, the simulation space is subdivided into a grid. In a preprocess step, the intensities of clouds illuminated by virtual point sources placed at each grid point are calculated and stored as the basis intensities. The clouds are then rendered in real-time by using the basis intensities. The memory requirement for the basis intensities is reduced by using the wavelet transform, and this results in a significant reduction in the computational cost for the intensity calculation. We have demonstrated that the proposed method can create realistic images of clouds in real-time.

There are a few things still to be done in the future. First, a method for rendering rain is needed. When there are bad weather conditions where the lightning is observed, in most cases it is also raining. Therefore, the rendering of rain becomes an important factor. Next, a further acceleration is required when using the proposed method in real-time applications such as computer games that compute many graphic components other than the rendering of clouds. One of the solutions to this problem is the use of a GPU. The computation of cloud intensity by using the proposed method is considered as a kind of vector product. The GPU could calculate this kind of vector product very efficiently by using programmable. The compression of the basis intensities is also an important issue. Although the proposed method has achieved a high compression ratio,



(a) lightning strokes from two seed points. (b) lightning strokes in clouds (two seed points). (c) pink lightning strokes (three seed points). (d) many lightning strokes (five seed points).

Fig. 7 Various examples.

the memory requirement for the basis intensities is still around one hundred megabytes. To reduce the memory requirements, we can use spherical harmonic functions for compactly representing the directional intensity distribution at each grid point. Finally, an extension of the proposed method to dynamic clouds has to be studied in the future since shapes of clouds change very rapidly under the bad weather conditions. A simple solution is to generate time-varying volume data of clouds in advance and to apply the proposed method repeatedly to each volume data. However, this requires huge amount of costs for both the computation and the memory capacity for the basis intensities. The costs could be reduced by making use of the coherency of the time-varying volume data.

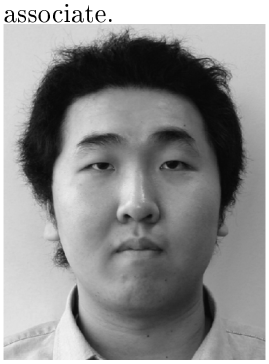
Acknowledgements This research is partly supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) from the Ministry of Internal Affairs and Communications of Japan.

References

1. Garg, K. and Nayar, S. K.: Photorealistic Rendering of Rain Streaks. ACM Trans. on Graphics, **25**(3), 998 – pp 1002 (2006)
2. Dobashi, Y., Yamamoto, T., and Nishita, T.: Efficient Rendering of Lightning Taking into Account Scattering Effects due to Clouds and Atmospheric Particles. In Proceedings of Pacific Graphics 2001, pp 290 – pp 299 (2001)
3. Nishita, T., Iwasaki, H., Dobashi, Y., and Nakamae, E.: A Modelling and Rendering Method for Snow by Using Metaballs. Computer Graphics Forum, **16**(3), 357 – pp 364 (1997)
4. Fearing, P.: Computer Modelling of Fallen Snow. In Proceedings of SIGGRAPH 2000, pp 37 – pp 46 (2000)
5. Reed, T. and Wyvill, T.: Visual Simulation of Lightning. In Proceedings of SIGGRAPH 1994, pp 359 – pp 364 (1994)
6. Kruszewski, P.: A Probabilistic Technique for the Synthetic Imagery of Lightning. Computers & Graphics, **23**(2), 287 – pp 293 (1999)
7. Kim, T. and Lin, M. C.: Physically Based Animation and Rendering of Lightning. In Proceedings of Pacific Graphics 2004, pp 267 – pp 275 (2004)
8. Matsuyama, K., Fujimoto, T., and Chiba, N.: Real-time Animation of Spark Discharge. The Visual Computer, **22**(9-11), 761 – pp 771 (2006)
9. Kajiya, J. T. and Herzen, B. P. V.: Ray Tracing Volume Densities. Computer Graphics, **18**(3), 165 – pp 174 (1984)
10. Max, N.: Efficient Light Propagation for Multiple Anisotropic Volume Scattering. In Proceedings of the Fifth Eurographics Workshop on Rendering, pp 87 – pp 104 (1994)
11. Stam, J. and Fiume, E.: Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes. In Proceedings of SIGGRAPH 1995, pp 129 – pp 136 (1995)
12. Nishita, T., Dobashi, Y., and Nakamae, E.: Display of Clouds Taking into Account Multiple Anisotropic Scattering and Sky Light. In Proceedings of SIGGRAPH 1996, pp 379 – pp 386 (1996)
13. Jansen, H. W. and Christensen, P. H.: Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps. In Proceedings of SIGGRAPH 1998, pp 311 – pp 320 (1997)
14. Dobashi, Y., Kaneda, K., Yamashita, H., Okita, T., and Nishita, T.: A Simple, Efficient Method for Realistic Animation of Clouds. In Proceedings of SIGGRAPH 2000, pp 19 – pp 28 (2000)
15. Sloan, P.-P., Kautz, J., and Snyder, J.: Precomputed Radiance Transfer for Real-time Rendering in Dynamic, Low-frequency Environments. ACM Trans. on Graphics, **21**(3), 527 – pp 536 (2002)
16. Kristensen, A. W., Akenine-Moller, T., and Jensen, H. W.: Precomputed Local Radiance Transfer for Real-time Lighting Design. ACM Trans. on Graphics, **24**(3), 1208 – pp 1215 (2005)
17. Hasan, M., Pellacini, F., and Bala, K: Direct-to-indirect Transfer for Cinematic Relighting. ACM Trans. on Graphics, **25**(3), 1089 – pp 1097 (2006)
18. Daubert, K., Heidrich, W., Kautz, J., Dischler, J.-M, and Seidel, H.-P.: Efficient Light Transport Using Precomputed Visibility. IEEE Computer Graphics & Applications, **23**(3), 28 – pp 37 (2003)
19. Szimary-Kalos, L., Sbert, M., and Umenhoffer, T.: Real-time Multiple Scattering in Participating Media with Illumination Networks. In Proceedings of Eurographics Symposium on Rendering 2005, pp 277 – pp 282 (2005)
20. Cabral, B., Cam, N., and Foran, J.: Accelerated Volume Rendering and Tomographics Reconstruction Using Texture Mapping Hardware. In Proceedings of 1994 Symposium on Volume Visualization, pp 91 – pp 98 (1994)
21. Stollnitz, E. J., DeRose, T. D., and Salesin D. H.: Wavelets for Computer Graphics A Primer, part 1. IEEE

Computer Graphics & Applications, **15**(3), 76 – pp 84 (1995)

YOSHINORI DOBASHI is an associate professor at Hokkaido University in the graduate school of information science and technology, Japan since 2006. His research interests center in computer graphics including realistic Image synthesis, efficient rendering, sound modeling for virtual reality applications. Dobashi received his BE, ME and Ph.D in Engineering in 1992, 1994, and 1997, respectively, from Hiroshima University. He worked at Hiroshima City University from 1997 to 2000 as a research associate.



YOSHIHIRO ENJYO works for Fuji Electric FA Components & Systems Co, Ltd. He received his BE and ME from Hokkaido University. His research interests center in computer graphics.

TSUYOSHI YAMAMOTO is a Professor of Information Science and Technology at Hokkaido University. He received a Bachelor's and Master's degrees in Electronics Engineering from Hokkaido University in 1976 and 1978, and a degree of Doctor of Engineering in Electrical Engineering from Hokkaido University in 1986. From 1978 to 1980, he worked at Fujitsu Ltd. on LSI design of microcomputer peripherals. In 1982, he appointed as a lecturer at Department of Electrical Engineering, Hokkaido University. Since then, he worked in the fields of Computer Graphics, Symbolic Computation, Medical Image Processings and Information Media Technologies.

TOMOYUKI NISHITA received the B.E., M.E., and Ph.D. degrees in Electrical Engineering from the Hiroshima University, Japan, in 1971, 1973, and 1985, respectively. He worked for Mazda Motor Corp. from 1973 to 1979. He has been a lecturer at the Fukuyama University since 1979, then became an associate professor in 1984, and later became a professor in 1990. He moved to the Department of Information Science of the University of Tokyo as a professor in 1998 and now is a professor at the Department of Complexity Science and Engineering of the University of Tokyo since 1999. He received Research Award on Computer Graphics from IPSJ in 1987, Steven A. Coons Awards from ACM SIGGRAPH in 2005, and CG Japan Award in 2006. His research interest is mainly for computer graphics. He is a member of IEICE, IPSJ, ACM, and IEEE.