An Inverse Problem Approach for Automatically Adjusting the Parameters for Rendering Clouds Using Photographs

Yoshinori Dobashi[1,2] Wataru Iwasaki[1] Ayumi Ono[1] Tsuyoshi Yamamoto[1] Yonghao Yue[3] Tomoyuki Nishita[3]
[1]Hokkaido University [2]JST, CREST [3]The University of Tokyo

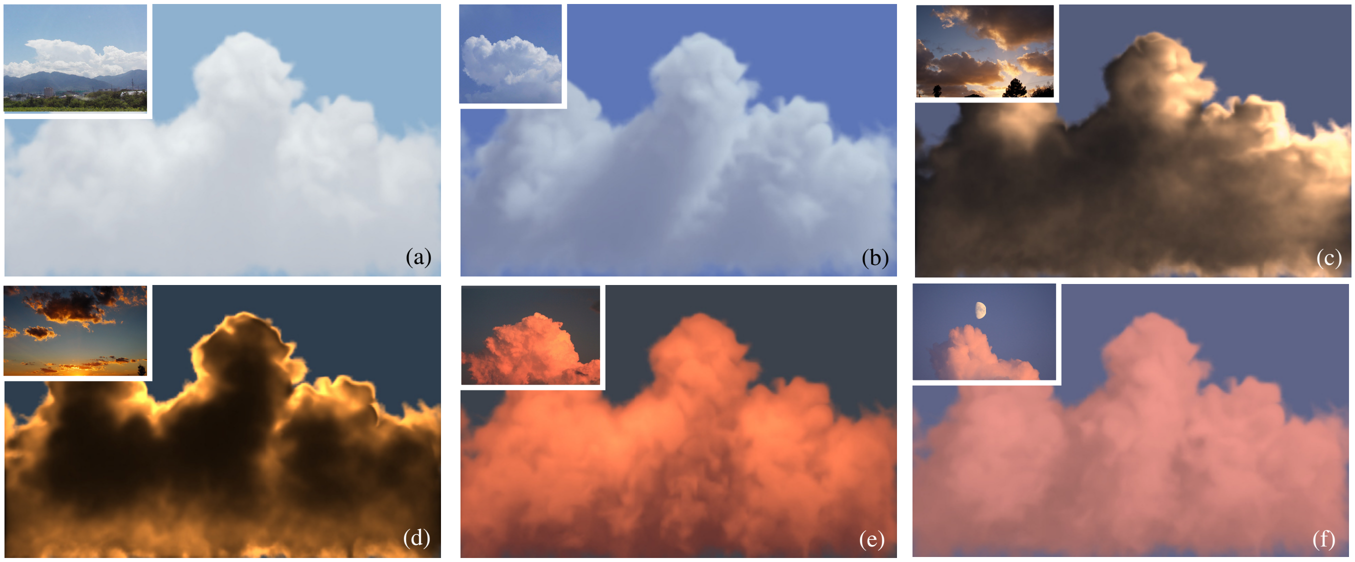


Figure 1: *Examples of our method. The parameters for rendering clouds are estimated from the real photograph shown in the small inset at the top left corner of each image. The synthetic cumulonimbus clouds are rendered using the estimated parameters.*

Abstract

Clouds play an important role in creating realistic images of outdoor scenes. Many methods have therefore been proposed for displaying realistic clouds. However, the realism of the resulting images depends on many parameters used to render them and it is often difficult to adjust those parameters manually. This paper proposes a method for addressing this problem by solving an inverse rendering problem: given a non-uniform synthetic cloud density distribution, the parameters for rendering the synthetic clouds are estimated using photographs of real clouds. The objective function is defined as the difference between the color histograms of the photograph and the synthetic image. Our method searches for the optimal parameters using genetic algorithms. During the search process, we take into account the multiple scattering of light inside the clouds. The search process is accelerated by precomputing a set of intermediate images. After ten to twenty minutes of precomputation, our method estimates the optimal parameters within a minute.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; I.3.3 [Computer Graphics]: Picture/Image Generation;

Keywords: clouds, rendering parameters, inverse problem

Links: DL PDF

1 Introduction

Clouds are important elements when synthesizing images of outdoor scenes to enhance realism. A volumetric representation is often employed and the intensities of the clouds are computed taking into account the scattering and absorption of light in order to display realistic clouds. However, one of the problems is that the quality of the rendered image depends on many parameters, which need to be adjusted manually by rendering the clouds repeatedly with different parameter settings. This is not an easy task since the relationship between the resulting appearance of the clouds and the parameters is highly nonlinear. The expensive computational cost for the rendering process makes this more difficult. This paper focuses on automatic adjustment of the parameters to address this task.

Recently, many real-time methods have been proposed for editing the parameters used in rendering images [Harris and Lastra 2001; Bouthors et al. 2008; Zhou et al. 2008]. These methods are fast so-

ACM Reference Format
Dobashi, Y., Iwasaki, W., Ono, A., Yamamoto, T., Yue, Y., Nishita, T. 2012. An Inverse Problem Approach for Automatically Adjusting the Parameters for Rendering Clouds Using Photographs. *ACM Trans. Graph. 31* 6, Article 145 (November 2012), 10 pages. DOI = 10.1145/2366145.2366164 http://doi.acm.org/10.1145/2366145.2366164.

lutions to the forward rendering problem: the corresponding output image is computed in real-time using the given parameters, allowing one to efficiently find the appropriate parameters that produce a desired image. However, even using these methods, a repetitive trial-and-error process is still required until satisfactory results are obtained. Our aim is to remove the manual trial-and-error process by solving an inverse rendering problem.

One may think that our purpose would be achieved by using image processing techniques such as color transfers between images [Reinhard et al. 2001] that convert a synthetic image into one whose appearance is similar to the photograph. However, the color transfer methods achieve only a superficial conversion. Our experiments showed that these methods cannot produce satisfactory results unless the parameters are carefully chosen to make the appearance of the synthetic clouds similar to the photograph (see Section 5.1). Furthermore, since color transfer methods are image-based methods, we cannot change the viewpoint and lighting conditions such as sunlight directions or colors. The appearance of clouds is more fundamentally determined by the parameters used to render them, and it is these parameters that our method determines.

Our method determines the parameters for rendering clouds such that the appearance of the synthetic clouds is similar to a specified source image. Our purpose is not to estimate physically correct parameters but to find the parameters that can produce an image that is visually similar to the clouds in the input photograph. The parameters estimated by our method are listed in Table 1. We take into account both single and multiple scattering of light inside the clouds in creating the synthetic images. The source image is a photograph of real clouds. We use a color histogram to measure the visual difference between the synthetic image and the photograph. The use of a color histogram is motivated by the fact that it is often used in image retrieval/indexing applications.

Solving the inverse rendering problem, however, is not trivial because the intensity of clouds is a highly nonlinear function of the parameters used to render them. Furthermore, there is seldom a unique solution to this problem: many different sets of parameters can produce similar images. We chose genetic algorithms (GAs) to address this problem because of their two capabilities: 1) they can find the optimal parameters efficiently even for such a highly nonlinear problem and 2) they can find a number of candidates for the optimal parameters during the optimization process. To make use of these capabilities, our system records a specified number of high-ranking parameters and displays the corresponding images to the user. The user then selects one of them as the optimal solution. Since the computation of the multiple scattering is generally time-consuming, we accelerate the computation by precomputing a set of intermediate images and by utilizing a GPU. Using our method, the user can obtain the appropriate parameters to create realistic images of clouds by simply specifying a photograph of real clouds. Once the parameters have been obtained, the user can render the synthetic clouds with various viewpoints, sunlight directions, and sunlight colors. Note that our method does not guarantee the realism of images rendered with viewpoints and sunlight directions that are different from those used to estimate the parameters. However, our experiments showed that realistic images are generated in most cases.

One important thing to note is that with our method the optimal parameters can be found even if the density distribution of the clouds is not physically reasonable. In computer graphics, many methods have been proposed for generating the cloud density distribution, such as a procedural approach [Ebert et al. 2009; Schpok et al. 2003], a physically-based simulation [Miyazaki et al. 2002], and an image-based approach [Dobashi et al. 2010]. These methods can produce realistic looking density distributions but they are not guaranteed to be physically-valid. This means that physically correct parameters for rendering clouds do not always provide satisfactory results.

Table 1: *Parameters to be estimated.*

parameter	meaning
$c_{in}(\lambda)$	color component of light incident on clouds (λ: wavelength)
L_{in}	intensity of light incident on clouds
$L_{sky}(q, \lambda)$	intensity of the sky behind the clouds at pixel q
g	asymmetry factor of phase function
σ_t	extinction cross section of cloud particles
β	albedo of cloud particles
L_{amb}	constant ambient term
$\kappa_a(\lambda)$	extinction coefficient of atmospheric particles

Fig. 1 shows an example of our method. The inset in each image is the photograph specified by the user. Our method successfully finds the parameters that can render the synthetic cumulonimbus clouds with a similar appearance to that in the photograph.

2 Related Work

Many methods have been proposed for rendering participating media [Stam 1995; Nishita et al. 1996; Jensen and Christensen 1998; Premoze et al. 2004; Yue et al. 2010]. Although these methods can create realistic images of clouds, the computational cost is expensive. This makes it time-consuming to adjust the parameters so that the desired appearance of clouds is produced. In order to address this problem, real-time rendering methods have been proposed [Harris and Lastra 2001; Bouthors et al. 2008; Zhou et al. 2008]. However, as we have mentioned previously, these methods only solve the forward problem.

There have been many research projects that treat different kinds of inverse rendering problem. In Kawai et al. [1993] and Schoeneman et al. [1993], methods for solving the inverse problem with respect to the lighting parameters were proposed. Following these methods, many methods have been proposed for the inverse lighting problem. A detailed discussion on this subject can be found in Pellacini et al. [2007]. Other than the lighting problem, a general solution to setting parameters was proposed by Marks et al. [1997]. Although this method is useful for browsing the parameter space to obtain intuition of the space, it is not designed to solve inverse problems. More recently, methods for estimating the parameters for rendering hair [Bonneel et al. 2009], translucent objects [Munoz et al. 2011; Wang et al. 2008], and haze [Fattal 2008] have been proposed.

We employ GAs for estimating the parameters for rendering clouds. GAs have also been used in the field of computer graphics such as in texture synthesis [Sims 1991], image based simulation of facial aging [Hubball et al. 2008], image recognition [Katz and Thrift 1994], estimation of reflectance properties [Munoz et al. 2009], and shader simplification [Sitthi-Amorn et al. 2011].

There have been several research studies on the inverse problem related to light scattering. Li and Yang [1997] studied inverse radiation problems. Zhang et al. [2005] proposed a method for determining the optical properties of human skin. These methods use GAs. Kienle et al. [1996] proposed a measurement system for determining the optical properties of biological tissue. In this method, neural networks are trained using Monte Carlo simulations and then used to estimate the optical properties from the measured data. In the field of remote sensing, estimation of the optical properties of

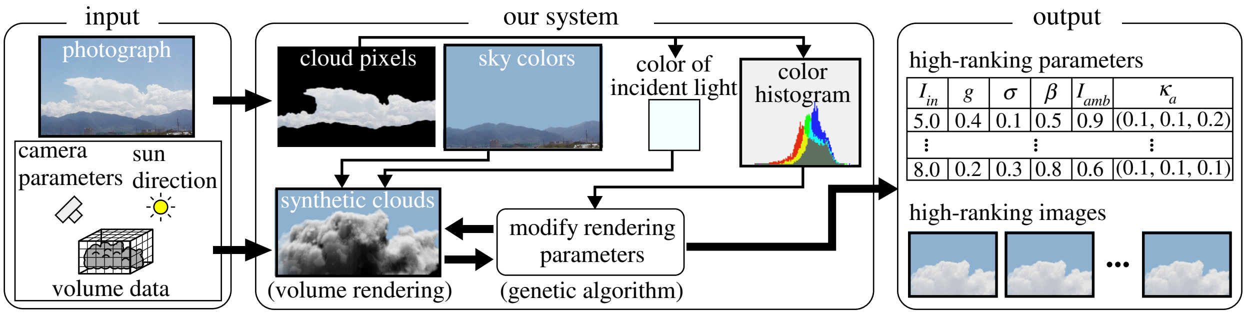


Figure 2: *Overview of our method. The inputs to our system are a photograph of real clouds, volume data of synthetic clouds, the sun direction, and camera parameters. Our system then estimates the parameters for rendering synthetic clouds so that the color histograms of the synthetic image and the photograph become the same. The outputs are a set of high-ranking parameters and corresponding synthetic images.*

clouds is one of the active research areas and many methods have been proposed. Davis and Marshak [2010] have conducted a nice survey on this topic. However, the goal of these research studies is to estimate physically correct parameters while ours is to obtain visually reasonable parameters (potentially non-physical parameters). Our method is designed to be efficient in achieving our goal.

3 Problem Definition

The inputs to our system are volume data representing the density distribution of synthetic clouds, and a photograph of real clouds. The direction of the sunlight and the camera parameters used to render synthetic clouds also need to be specified by the user. The user is responsible for preparing the volume data of the synthetic clouds. It can be generated by, for example, procedural approaches [Ebert et al. 2009] or fluid simulation [Miyazaki et al. 2002]. Our method is not to reconstruct the 3D shape of the clouds in the photograph but to determine the parameters to render synthetic clouds. A photograph of real clouds can be prepared either by taking a picture of the sky or by searching images through the internet. The system called SkyFinder proposed by Tao et al. [2009] might make it easy to find a photograph of clouds with the desired appearance. Our system then searches for the optimal parameters that minimize the following objective function O:

$$\arg\min_{\mathbf{c}} O(\mathbf{I}_{cg}(\mathbf{c}), \mathbf{I}_{usr}), \tag{1}$$

where $\mathbf{c}$ is a vector consisting of the parameters used for rendering the synthetic image $\mathbf{I}_{cg}$. $\mathbf{I}_{usr}$ is the photograph specified by the user. The objective function O measures the visual difference between $\mathbf{I}_{cg}$ and $\mathbf{I}_{usr}$. We use color histograms to compute the visual differences. O is defined by:

$$O = \frac{1}{3} \sum_{\lambda=R,G,B} \sum_{n=0}^{n_L-1} |h_{cg}(n,\lambda) - h_{usr}(n,\lambda)|, \tag{2}$$

where λ is the wavelength sampled at the wavelength corresponding to the RGB color channels, n_L is the number of intensity levels, and h_{cg} and h_{usr} represent histograms of $\mathbf{I}_{cg}$ and $\mathbf{I}_{usr}$, respectively. h_{cg} and h_{usr} are normalized by dividing them by the number of pixels. These histograms are computed using only the pixels corresponding to the clouds. The extraction of the cloud pixels is described in Section 4.1.

The intensity of clouds in the synthetic image $\mathbf{I}_{cg}$ is calculated based on the rendering equations for the clouds [Nishita et al. 1996; Cerezo et al. 2005; Zhou et al. 2008]. The intensity of clouds depends on many parameters, such as the intensities of the sunlight and the skylight, and the optical properties of atmospheric and cloud particles. In our method, the only light source illuminating the clouds is the sun. The skylight is not taken into account as a light source. However, the intensity of light directly reaching the viewpoint from the sky behind the clouds is taken into account. The attenuation and scattering of light due to atmospheric particles between the clouds and the viewpoint are also taken into account. Our purpose is to find the parameters that can reproduce the appearance of the clouds in the input photograph. The parameters estimated by our method are listed in Table 1. We briefly describe these parameters in the following.

$c_{in}(\lambda)$ and L_{in} are color and intensity components of the light incident on the clouds, respectively. These parameters indicate the sun light reaching the clouds after traveling through the atmosphere. The color c_{in} and intensity L_{in} are separately estimated. $L_{sky}(q,\lambda)$ is the intensity of the sky behind the clouds at pixel q. g, σ_t, and β, are the optical parameters of the cloud particles. g is a parameter of the Henyey-Greenstein function commonly used as the phase function of cloud particles. g controls the anisotropy of the phase function. The extinction cross section σ_t controls the degree of attenuation when light travels through the cloud. The intensity of the light is attenuated exponentially and σ_t determines the exponential decay. β is the albedo of the cloud particles. Next, L_{amb} is a constant ambient term commonly used to compensate the effect of higher order multiple scattering which is usually truncated due to the limitation of computation cost. Although clouds consist of particles with different sizes, we assume that the particles are relatively large compared to the wavelength of the incident light and therefore their optical properties (g, β, and σ_t) are independent of wavelength. We assume the ambient term is also independent of wavelength. This implies that the color of the ambient light is equivalent to $c_{in}(\lambda)$. Finally, κ_a is the extinction coefficient of atmospheric particles. This parameter depends on the wavelength. Among the parameters in Table 1, $c_{in}(\lambda)$ and $L_{sky}(q,\lambda)$ are estimated using image processing techniques (see Section 4.1). L_{in}, g, σ_t, β and L_{amb} are estimated using GAs (see Section 4.2). For GAs, the ranges of g and β are from zero to one but the ranges of L_{in}, σ_t, κ_a, and L_{amb} need to be specified by the user.

4 Estimation Method

The minimization problem defined in the previous section is solved by rendering the clouds repeatedly with various parameter settings using GAs. To render clouds, we take into account both single and multiple scattering. The scattering and absorption due to atmospheric particles between the clouds and the viewpoint are also taken into account. We employ the simplest model where the density of the atmospheric particles is assumed to be uniform and the intensity of scattered light due to atmospheric particles is assume to be constant. Under these assumptions, the intensity of light reaching the viewpoint for a pixel is a blended intensity of the clouds and the sky behind the clouds.

An overview of our system is illustrated by Fig. 2. Before using GAs, our system extracts cloud pixels from the input photograph and estimates the color of the incident light $c_{in}(\lambda)$ and the intensity of the sky behind the clouds $L_{sky}(q, \lambda)$. The color of the incident light is different from that of the sun because the sunlight is attenuated and scattered by atmospheric particles before reaching the clouds. The color histogram of the input photograph is also calculated using the extracted cloud pixels. The rest of the parameters are then estimated using GAs. The images of the synthetic clouds are repeatedly created by using volume rendering techniques with different parameter settings. GAs compute the objective function for each of the candidate parameter sets to measure the quality of the parameters and modify the parameters. Each set of parameters is ranked by the objective function and high-ranking parameter sets are stored. The output of our system is a set of high-ranking parameters and their corresponding images. The details are described in the following subsections.

4.1 Colors of the Sun and the Sky

In order to estimate the intensity of the sky behind the clouds at pixel q, $L_{sky}(q, \lambda)$, we use the method proposed in Dobashi et al. [2010]. First, each pixel in the input photograph is classified into either a cloud pixel or a sky pixel depending on the chroma of each pixel (see [Dobashi et al. 2010] for more details). The color of the sky behind the clouds is then calculated by removing the cloud pixels and then interpolating the colors of the removed pixels from the surrounding sky pixels. For this interpolation, we use the Poisson equation [Perez et al. 2003]. The intensities of the sky at the cloud pixels are calculated by solving $\nabla^2 L_{sky}(q, \lambda) = 0$ with the fixed boundary condition that the intensities at the boundary of the cloud pixels are equal to the average intensity of the neighboring sky pixels. If the above method does not work very well, our system allows the user to verify and modify the result by hand.

The color of the incident light from the sun, $c_{in}(\lambda)$, is calculated in the following way. Since the optical properties of the cloud particles are independent of wavelength, the colors of the bright regions of the clouds are nearly equal to those of the light incident on the clouds. We use this property to determine the color. First, the colors of the cloud pixels are converted into grayscale. Each of the cloud pixels is then classified into either a brighter or a darker pixel. For this classification, we use the method proposed by Otsu [1979]. The average color of the brighter cloud pixels is used for the color of the light incident on the clouds. This approach works well even if the sun is behind the clouds as shown in Fig. 1. However, the method does not work well when the sun is completely hidden by optically thick clouds. In this case, the estimated color would be a gray. This situation happens when the user specifies a photograph of an overcast sky.

4.2 Estimating Parameters using Genetic Algorithms

Besides c_{in} and L_{sky}, there are eight unknown parameters as shown in Table 1 (note that the extinction coefficient κ_a depends on the wavelength λ (= R, G, B)). Our method uses GAs to search for the optimal parameters that minimize the objective function O defined by Eq. 2. In order to increase efficiency, we divide the search process into two parts.

As described above, the intensity of light reaching the viewpoint through pixel q is calculated by blending the intensities of the clouds and the sky. The blending factor is determined by the distance from the viewpoint to the clouds and the extinction coefficient for the atmospheric particles. Thus, the intensity of pixel q of the synthetic image is calculated by:

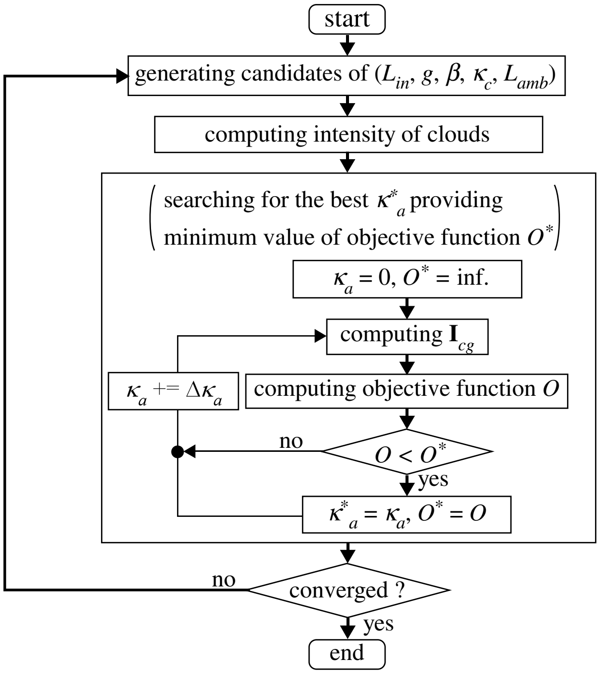

start
generating candidates of (L_{in}, g, β, κ_c, L_{amb})
computing intensity of clouds
searching for the best κ^*_a providing minimum value of objective function O^*
$\kappa_a = 0$, $O^* =$ inf.
computing $\mathbf{I}_{cg}$
$\kappa_a \mathrel{+}= \Delta\kappa_a$
computing objective function O
no
$O < O^*$
yes
$\kappa^*_a = \kappa_a$, $O^* = O$
no
converged ?
yes
end

Figure 3: *Estimation of parameters using a genetic algorithm.*

$$
\begin{aligned}
L_{cg}(q, \lambda) \quad &= \quad L_c(q, \lambda)\exp(-\kappa_a(\lambda)t_c) \\
&\quad\quad +(1.0 - \exp(-\kappa_a(\lambda)t_c))L_{sky}(q, \lambda), \quad (3)
\end{aligned}
$$

where L_c is the intensity of the clouds, L_{sky} is the intensity of the sky behind the clouds, and t_c is the distance from the viewpoint to the clouds. When the sunlight reaches a point between the viewpoint and the clouds, it scatters toward the viewpoint. The second term approximates this component. When the distance t_c becomes infinity, it converges to the intensity of the sky, L_{sky}. The above equation implies that atmospheric effects and the intensity of the clouds can be separately computed, since L_c is independent of κ_a. This leads to the following estimation algorithm: GAs are applied only to the five parameters related to the clouds (L_{in}, g, σ_t, β, L_{amb}) and the extinction coefficient of the atmosphere, $\kappa_a(\lambda)$, is determined using a linear search algorithm. Fig. 3 illustrates the details of our estimation algorithm. First, candidates for the five parameters are randomly sampled using GAs and, using these parameters, the intensity of the clouds is calculated. Next, the value of κ_a that minimizes the objective function O is searched by sampling κ_a at regular intervals. For each sampled value of κ_a, a synthetic image $\mathbf{I}_{cg}$ is created and the function O is calculated. The minimum value of O is then fed back into the GAs and improved candidates for the parameters are generated. By repeating these processes, our method searches for the optimal set of parameters.

The search process terminates if one of the following three conditions is satisfied: 1) the objective function becomes smaller than a specified threshold ϵ_O, 2) the best parameters are unchanged during a specified number of successive iterations n_{suc}, and 3) the number of iterations exceeds a specified number n_{max}.

For the GAs, we basically follow the standard approach

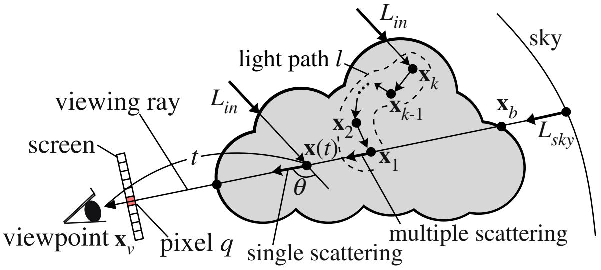


Figure 4: *Intensity calculation of clouds.*

[Goldberg 1989]. Each cloud parameter is quantized and converted into a corresponding binary bit string. Then, the bit strings for all the cloud parameters are connected into a single longer bit string that is used as the gene of an individual. Initially, GAs generate n individuals using random numbers, where n is specified by the user. GAs iteratively generate a new set of n individuals based on a fitness function that evaluates the quality of each individual. We use the inverse of our objective function, $1/(O+1)$, as the fitness function so that the maximum is one. New individuals are generated through two genetic operators, one called a crossover operator and the other a mutation operator. We use a two-point crossover. We also employ a so-called elitist selection strategy, by which the best individual at a certain iteration is carried over to the next iteration. We chose these genetic operators experimentally. Although there are different operators such as a one-point crossover, we have not observed any significant differences in the convergence of the objective function. The elitist selection, however, is important to increase the convergence speed. For more details about GAs, there are many good textbooks such as the one by Goldberg [1989].

4.3 Acceleration of Intensity Calculation of Clouds

Although the estimation method described in the previous subsection works well, the computational cost is very expensive. At each iteration of GA, there are n individuals and so we need to calculate the synthetic image, $\mathbf{I}_{cg}$, n times. In our experiments, we use $n = 20$ and our method usually takes 50 iterations before convergence. In this case, the clouds need to be rendered $1,000$ times. This is very time-consuming especially when taking into account multiple scattering. During optimization, we assume a fixed camera position and a fixed sun direction. Under these assumptions, we can precompute a set of images with different parameter settings. Then, the synthetic images $\mathbf{I}_{cg}$ can be generated efficiently by interpolating the precomputed images. In the following, details of the precomputation are described. Note that the notation used in the following equations is different from the traditional kind in order to clarify the relationship between the functions in the intensity calculation and the unknown parameters.

The most time-consuming part is calculating the intensity of the clouds, L_c. L_c is expressed by the sum of the following three components: the intensity due to single scattering L_s, the intensity due to multiple scattering L_m, and the intensity of light reaching the viewpoint from the background sky behind the clouds after attenuated by the cloud particles, L_b. That is,

$$L_c(q,\lambda) = L_s(q,\lambda) + L_m(q,\lambda) + L_b(q,\lambda). \tag{4}$$

We precompute a set of images for each of the three components, as described in the following sections. We assume that the sun is the only light source. Skylight is not taken into account.

Single scattering: Let us consider a point $\mathbf{x}(t)$ on the viewing ray as shown in Fig. 4, where t indicate a distance from the viewpoint $\mathbf{x}_v$. Light incident on the clouds reaches point $\mathbf{x}(t)$ after being attenuated by the cloud particles. It then scatters at point $\mathbf{x}(t)$ toward the viewpoint and is attenuated again by the cloud particles between the viewpoint and point $\mathbf{x}(t)$. The intensity of the scattered light is determined by the phase function. The intensity due to single scattering, L_s, is obtained by accumulating the scattered intensities along the viewing ray. The constant ambient term L_{amb} is included in this single scattering component. We assume that each cloud particle emits a certain amount of ambient light and so the intensity due to the ambient light at a point is proportional to the cloud density. The total intensity due to the ambient light reaching the viewpoint is then obtained by accumulating the ambient light along the viewing ray taking into account the attenuation due to cloud particles. Let us denote the attenuation between an arbitrary pair of points $\mathbf{x}$ and $\mathbf{y}$ as $\tau(\mathbf{x},\mathbf{y},\sigma_t)$, which depends on the extinction cross section σ_t. The intensity due to single scattering is then represented by:

$$\begin{aligned} L_s(q,\lambda) &= c_{in}(\lambda) L_{in} \beta p(\theta, g) \\ &\times \int_0^T \sigma_t \rho(\mathbf{x}(t)) \tau(\mathbf{x}_l, \mathbf{x}(t), \sigma_t) \tau(\mathbf{x}_v, \mathbf{x}(t), \sigma_t) dt \\ &+ c_{in}(\lambda) L_{amb} \int_0^T \rho(\mathbf{x}(t)) \tau(\mathbf{x}_v, \mathbf{x}(t), \sigma_t) dt, \end{aligned} \tag{5}$$

where p is the phase function, θ is called the phase angle defined by the angle between the incident light direction and the scattered light direction, g is the asymmetry factor, and $\rho(\mathbf{x}(t))$ is number density of cloud particles at point $\mathbf{x}(t)$. $\mathbf{x}_l$ represents the position of the sun and T is the length of the intersection segment between the viewing ray and the clouds. In the first term on the right, the phase function p is outside of the integral since the sun is assumed to be a parallel light source and therefore the phase angle θ is the same at every point on the viewing ray. The last term on the right indicates the effect of the constant ambient term. In the above equation, the two integral terms can be precomputed for a different value of σ_t sampled at regular intervals. These integral terms are evaluated using the ray-marching method.

Multiple scattering: We use a Monte-Carlo method, a forward path tracing, to compute the multiple scattering. To compute the intensity of a pixel, many light paths are randomly generated and their contributions are calculated. The intensity of a pixel is estimated as the average of the contributions [Yue et al. 2010]. A light path is constructed from the viewpoint by randomly generating successive scattering events. Let us consider a light path l through pixel q consisting of k points where scattering events occur (see Fig. 4). When the probability of generating the light path is P, the contribution of this path to the pixel intensity can be written as:

$$\begin{aligned} \Delta_m^{(k)}(\lambda) &= L_{in} c_{in}(\lambda) \beta^k \tau(\mathbf{x}_v, \mathbf{x}_1, \sigma_t)/P \\ &\times \prod_{i=1}^{k} p(\theta_i, g) \sigma_t \rho(\mathbf{x}_i) \tau(\mathbf{x}_i, \mathbf{x}_{i+1}, \sigma_t), \end{aligned} \tag{6}$$

where θ_i represents the phase angle at point $\mathbf{x}_i$. $\mathbf{x}_{k+1}$ corresponds to the position of the sun $\mathbf{x}_l$. For more details on the computation for multiple scattering, please see the supplemental document. We could precompute the multiple scattering component for various values of β, g, and σ_t but this would be very time-consuming. Instead, our method precomputes the following function for each pixel q by sampling g and σ_t regularly:

$$\begin{aligned} f_m^{(k)}(q, g, \sigma_t) &= E[\tau(\mathbf{x}_v, \mathbf{x}_1, \sigma_t)/P \\ &\times \prod_{i=1}^{k} p(\theta_i, g) \sigma_t \rho(\mathbf{x}_i) \tau(\mathbf{x}_i, \mathbf{x}_{i+1}, \sigma_t)], \end{aligned} \tag{7}$$

where E represents the average of the contributions from many light paths. We precompute the above function for $k \geq 2$ since

Table 2: *Volume resolutions and precomputation times. Note that, for Fig. 14, multiple scattering is not taken into account.*

figure	volume resolution	precomputation
Fig. 1	194 × 122 × 194	21 min.
Fig. 5	150 × 256 × 150	12 min.
Fig. 12	402 × 402 × 66	28 min.
Fig. 14	202 × 82 × 102	4 sec.

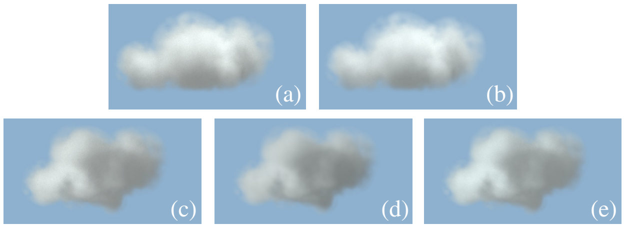


Figure 5: *Verification of our method. (a) synthetic source image, (b) image with parameters estimated by using (a), (c) source image with a different viewpoint and a different sunlight direction, (d) image with parameters used for (b), and (e) image rendered with parameters estimated by using (c).*

the single scattering component is calculated by Eq. 5. Then, the intensity of light due to multiple scattering is calculated efficiently by:

$$L_m(q,\lambda) = L_{in}c_{in}(\lambda)\sum_{k=2}^{n_m}\beta^k f_m^{(k)}(q,g,\sigma_t), \tag{8}$$

where n_m is the maximum number of scattering events and is specified by the user.

Attenuated intensity of the sky: L_b is expressed by the following equation.

$$L_b(q,\lambda) = \tau(\mathbf{x}_v,\mathbf{x}_b,\sigma_t)L_{sky}(q,\lambda), \tag{9}$$

where L_{sky} is the intensity of the sky behind the clouds calculated in Section 4.1. Our method precomputes $\tau(\mathbf{x}_v,\mathbf{x}_b,\sigma_t)$ that represents the attenuation due to cloud particles on the viewing ray.

The precomputed data for the above three components are stored in the form of images with a floating point precision. Then, computation of the intensity of the clouds results from a set of image composition operations. We use a GPU to accelerate both the precomputation and the intensity calculation of clouds.

5 Results

This section shows some experimental results. We used a desktop PC with an Intel Corei7-2600K 3.40 GHz (CPU) and an NVIDIA GeForce GTX590 (GPU). The search ranges for the parameters used for rendering were: $0 \leq L_{in} \leq 10$ and $0 \leq L_{amb},\beta,g,\sigma_t,\kappa_a(\lambda) \leq 1$. The cloud parameters $(L_{in},L_{amb},\beta,g,\sigma_t)$ were quantized with 32 bit precision. The interval for κ_a for the linear search was 0.1. We took into account up to fourth-order multiple scattering. The sampling interval of σ_t and g for precomputation was 0.1. The precomputed data is linearly interpolated in computing the intensity of clouds for arbitrary values of g and σ_t. The image size was 320 × 240 during the estimation process. The computational and memory costs are proportional to the image resolution, so we used images with as low a resolution as possible for the estimation. Since the resolution of the cloud volume used in this paper was around 200^3, we found, experimentally, that 320 × 240 images were sufficient. In addition, the low resolution image is sufficient for estimating only eight parameters. The size of the precomputed data was 141 MB. Note that, after determining the parameters, the final images were rendered with higher resolution. The size of the images shown in this paper is 640 × 480. The criteria to terminate the search process are: $\epsilon_O = 0.5$, $n_{suc} = 20$, and $n_{max} = 100$ (see the third paragraph of Section 4.2). For the experiments shown in Section 5.1, the search process was forcibly iterated until 100 times for validation purpose. For other examples, the number of iterations ranged approximately from 20 to 80 (see supplemental video showing the optimization process). The parameters used to render synthetic clouds are shown in the supplemental material.

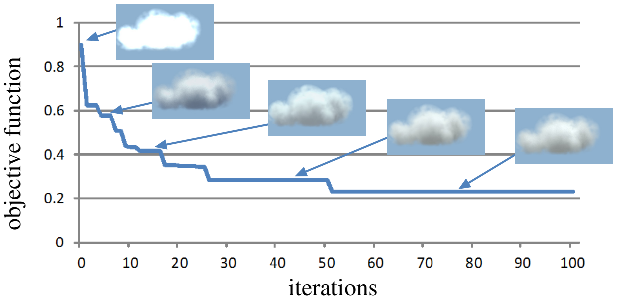


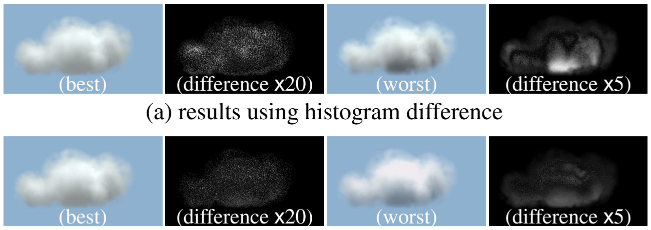
Figure 6: *Transition of the objective function.*

As mentioned in Section 4.3, the number of individuals for GAs, n, is 20. Using our method, the computation time for a single iteration is 0.1 sec. The volume resolutions and the precomputation times are shown in Table 2. Without our acceleration method, it took about 30 seconds to render a single image taking into account multiple scattering. Therefore, considering the clouds have to be rendered $n = 20$ times for each iteration, our method achieves approximately 6,000 times faster computation for the optimization. When including the precomputation, the speedup ratio reduces to 50 to 100 times. However, once the precomputation has been done the user can try different photographs, unless the user changes the sunlight direction and the viewpoint used to render the synthetic clouds.

5.1 Experimental Results

In order to investigate the capability of our method, we conducted several experiments. Note that, in the experiments shown in this section, we use the parameters of the highest rank in order to avoid subjective judgment.

First, in order to verify the ability of our method to find the optimal parameters, we tested the method using a synthetic image as a source image (see Fig. 5). We first rendered the image of synthetic clouds shown in Fig. 5(a) and used this image as the source image. The density distribution of the clouds was generated using the procedural technique [Ebert et al. 2009]. Fig. 5(b) shows the image using the optimal parameters estimated by our method. Next, we changed the viewpoint and the sunlight direction, and rendered the clouds again. Fig. 5(c) shows the image with the true parameters. Figs. 5(d) and (e) correspond to the images rendered with the parameters used for Fig. 5(b) and with the parameters estimated by using Fig. 5(c), respectively. Figs. 5(c) and (d) show one of the limitations of our method in that the appearance of the clouds rendered using the estimated parameters is different from that of the source clouds when the viewpoint and the sunlight direction are different from the ones used for the estimation process. However, by optimizing the parameters using Fig. 5(c), the appearance becomes



(a) results using histogram difference

(b) results using pixel-by-pixel intensity difference

Figure 7: *Comparison of results between histogram difference and intensity difference.From left to right, the best image, difference between the best image and Fig. 5(a), the worst image, and difference between the worst image and Fig. 5(a).Intensities of the difference images are multiplied by 20 or 5 for visualization purpose.*

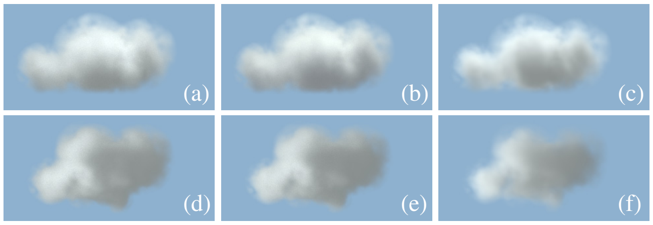


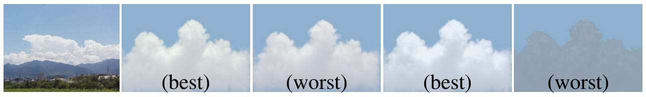
Figure 8: *Estimation results with different input directions of the sun. (a) through (c) are estimated by Fig. 5(a) and (d) through (f) are by Fig. 5(c).*

similar (Fig. 5(e)). The parameters for Figs. 5(c) and (e) are different (see supplemental material). This indicates that there are multiple local minima and our method finds one of them. This result is satisfactory for the purpose of finding the parameters that make the synthetic clouds similar to the source image. Fig. 6 shows the convergence of the objective function. This figure demonstrates that the optimal parameters can be found before 100 iterations. Some of the images during the optimization are shown in this figure.

In order to investigate the validity of using the histogram, we compare our objective function with an objective function using square sum of the pixel-by-pixel intensity differences. For this comparison, we use the synthetic clouds shown in Fig. 5(a). We executed our method a hundred times for both of the objective functions. We assume that the extinction coefficient for atmospheric particles, κ_a, is independent of the wavelength λ for this experiment. Fig. 7 shows the best and the worst images obtained by using these objective functions. Although the pixel-by-pixel intensity differences produce slightly better results than the histogram difference, the quality of the results is almost the same.

Next, we investigated the sensitivity of our method to the direction of the sun, specified by the user. We use Figs. 5(a) and (c) as source images. For estimating the parameters, we chose three different sun directions, which were 10, 20, and 40 degrees from the true directions. The results are shown in Fig. 8. For Figs. 8(a) through (c), Fig. 5(a) was used to estimate the parameters, and for Figs. 8(d) through (f), Fig. 5(c) was used. As shown in these images, our method can find appropriate parameters that can produce clouds with similar appearance, even if the sun direction is different from the true direction.

Next, in order to investigate the validity of using GAs, we compared our results with results obtained by a gradient-based optimization method. We chose the simplest method called the descent gradient method [Snyman 2005]. Using the source image and the synthetic clouds shown in Fig. 1(a), we executed both methods a hundred times with different initial parameters determined randomly. Then, we computed the averages and standard deviations of the objective function. The averages/standard deviations of our method and the descent gradient method are 0.40/0.01 and 1.32/0.55, respectively. The descent gradient method often converged to a local minimum and it depended significantly on the initial parameters. Fig. 9 shows the images rendered with the best and worst parameter sets. Fig. 9(a) shows the source image. The images in Figs. 9(b) and (c) correspond to the best and the worst parameters obtained by our method and the descent gradient method, respectively.



(a) photo (b) our method (c) descent gradient

Figure 9: *Comparison of images obtained by our method (a) and the descent gradient method (b).*

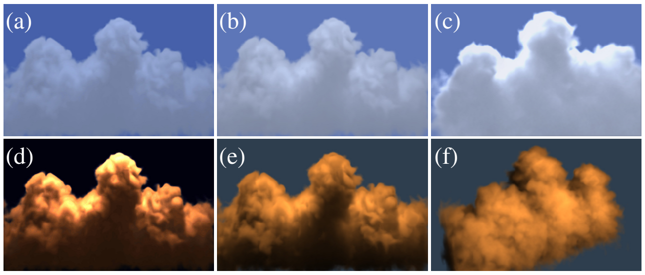


Figure 10: *Comparison of our method with the color transfer method. (a)(d) color transfer results, (b)(e) our results, and (c)(f) images with different viewpoints and light directions.*

Next, in order to demonstrate the importance of estimating the parameters, we compared our results with results using the color transfer method [Reinhard et al. 2001] (Fig. 10). We used two photographs as the source images for the color transfer, as shown in the inset images of Figs. 1(b) and (d). For the synthetic image before color transfer, we used the image shown in Fig. 1(a). This image was rendered with both single and multiple scattering. The results obtained by the global color transfer method [Reinhard et al. 2001] are shown in Figs. 10(a) and (d). The color transfer method was applied to the cloud pixels only. Figs. 10(b) and (e) show images rendered with parameters estimated by our method. In estimating the parameters, the same direction of the sun as in Fig. 1(a) was used. In Figs. 10(a) and (d), the colors are successfully transferred but the translucency of the clouds is not reproduced, making the resulting images unrealistic. Our method successfully reproduced the transparency and subtle color variations and realistic images were generated. Furthermore, after the optimal parameters of the clouds are obtained, we can easily create synthetic images with different viewpoints and sunlight directions, as shown in Figs. 10 (c) and (f).

Finally, Fig. 11 demonstrates the attribute that allows the user to modify the parameters estimated by our method. We modified the parameters used to create Fig. 10(c). The color of the clouds is changed by modifying the color of the incident light $c_{in}(\lambda)$ (Fig. 11(a)). In Fig. 11(b), the extinction cross section of the cloud particles σ_t is modified to increase the transparency of the clouds.

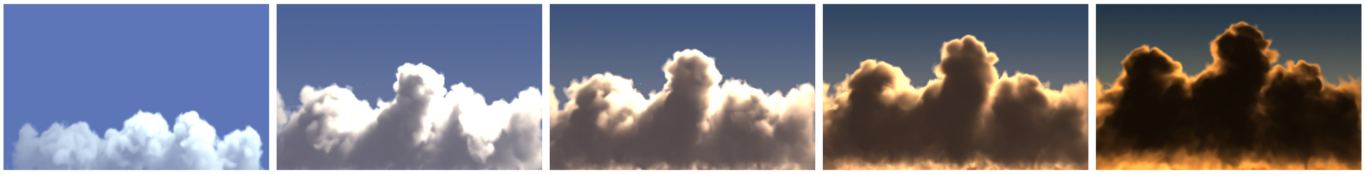


Figure 13: *Snapshots from an animation of cumulonimbus development. The leftmost and the rightmost images are rendered with the parameters used for Fig. 1(a) and Fig. 1(d), respectively. The parameters for the in-between images are linearly interpolated.*

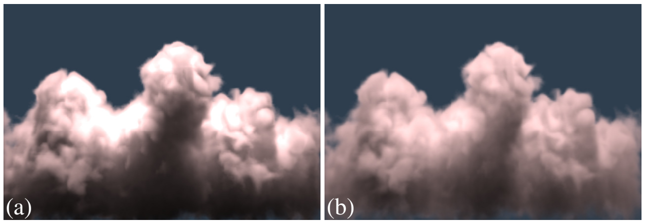


Figure 11: *Examples of editing parameters after optimization. After the color of incident light ($c_{in}(\lambda)$ is modified as shown in (a), the extinction cross section σ_t is decreased (b).*

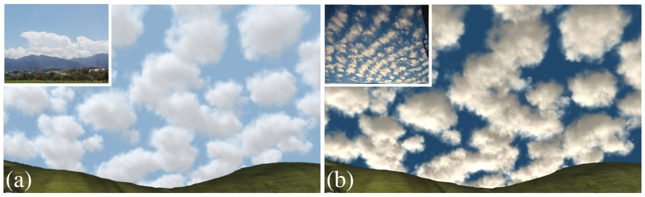


Figure 12: *Example of cumulus clouds. White clouds at daytime (a) and yellowish clouds at sunset (b). The inset in each image is used to optimize the parameters to render these clouds.*

5.2 Practical Examples

Fig. 1 shows an example of cumulonimbus clouds generated by fluid simulation [Miyazaki et al. 2002]. The inset in each image is the input photograph of the clouds. By estimating the parameters for rendering the clouds, the subtle color variations observed in the photograph are reproduced in the synthetic clouds. Fig. 12 shows examples of cumulus clouds rendered using the parameters obtained by our method. The volume data of the clouds are generated by fluid simulation [Miyazaki et al. 2002]. Figs. 12(a) and (b) show the cloud at daytime and sunset, respectively. This example demonstrates that our method can handle multiple clusters of clouds. Fig. 13 shows an application of our method to create an animation of dynamic clouds. We used two parameters to render the daytime and the sunset clouds in Fig. 1 to create the animation with the position of the sun changing. The parameters were linearly interpolated. Fig. 13 shows snapshots from the animation. Realistic color transitions are realized. Fig. 14 shows an example of unnatural clouds. The clouds were generated using controlled simulation [Dobashi et al. 2008]. In this example, we replaced the real clouds in the source photograph (shown in the inset image) with the synthetic clouds, rendered using the optimized parameters. This example was created by taking into account single scattering only. The synthetic clouds are naturally composited onto the real photograph. Finally, Fig. 15 shows an application of our method for creating an animation where the sun and the viewpoint move. The parameters for rendering clouds are estimated for Figs. 15(a) and (f) by using the photographs shown in Figs. 1(a) and (c), respectively. From Fig. 15(a) to (c), the viewpoint moves but the parameters for rendering the clouds are the same. From Figs. 15(d) to (f), both the viewpoint and the parameters are interpolated.

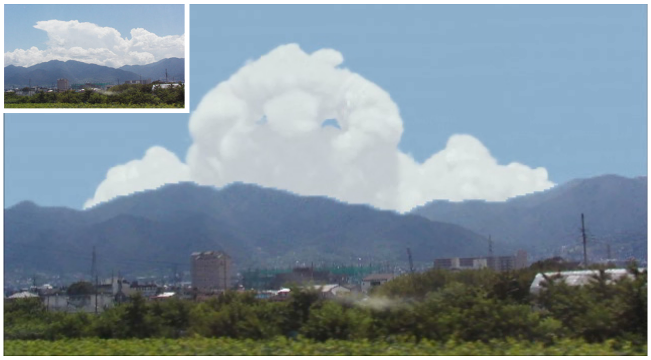


Figure 14: *Replacement of the clouds in the photograph.*

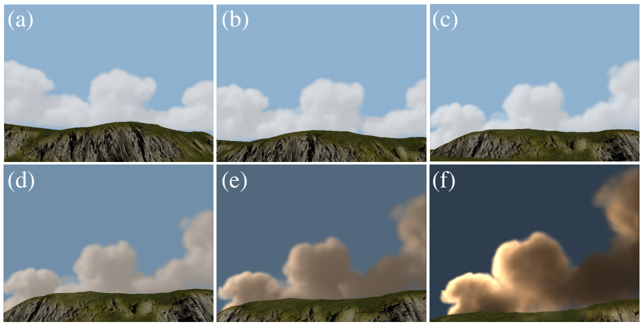


Figure 15: *Application of our method to an animation where the sun and the viewpoint move.*

6 Discussion

First, we would like to discuss our objective function, i.e., the histogram difference. The choice of the objective function is important to find the appropriate parameters. One of the straightforward approaches is to use pixel-based intensity differences between synthetic images and photographs. However, we cannot employ this approach since the shapes of the synthetic clouds and the real clouds in the photograph are different. The problem treated in this paper is equivalent to searching for the optimal image from an infinite number of images with different parameter settings. Image retrieval applications treat a similar problem and color histograms are often employed to measure the visual difference between images. This is the main reason for our use of a color histogram. Bonneel et al. [2009] also used color histograms to estimate the parameters for hair rendering from a single photograph. We examined several metrics for histogram matching such as difference, correlation, Chi-square, intersection, Bhattacharyya distance, and Earth Mover's Distance (see OpenCV manual for details). Since our experiments showed that the qualities of the results were similar for all of these, we chose the simplest one, the histogram difference. In some cases, however, our method might converge to the wrong local minimum and produce an unnatural appearance. This is one of the reasons why our system records a set of high-ranking parameters. An example of the wrong parameters is shown at the end of the supplemental video showing the optimization process. The rank 7 image corresponds to the wrong parameters. However, there are also good results among the high-ranking parameters.

One limitation of our method is that the direction of the sun is fixed during the optimization process. When the direction is changed, the precomputed data has to be recalculated. Therefore, the user needs to be careful when choosing the direction. A simple solution to this problem is to perform the precomputation for multiple directions of the sun, though this results in a long precomputation time and increased size of the precomputed data. Another solution would be to use a method to estimate the sun direction from photographs [Lalonde et al. 2012].

Currently, we do not take into account the skylight as a light source. We could approximate the effects of the skylight by making the ambient term wavelength-dependent. However, this approach results in a slower convergence speed since the search space is significantly increased. It would be much better to estimate the intensity distribution of the sky from a photograph [Lalonde et al. 2012] before using GAs in order to efficiently estimate the effects of the skylight.

In the examples shown in this paper, we took into account up to fourth-order multiple scattering. Although our method can handle higher order scattering, increasing the scattering order results in greater amount of precomputed data. One solution would be to cluster high orders of scattering into several sets (e.g., 2, 3-4, 5-6, and so on) as proposed by Bouthors et al. [2008]. Furthermore, we can compress the precomputed data using, e.g., wavelets. These would reduce the storage cost for the precomputed data.

The parameters estimated by our method are valid only for the viewpoint and the sunlight direction used in the optimization process. The appearance of synthetic clouds rendered with a different viewpoint and a different sunlight direction would be different from the appearance of the clouds in the photograph. However, in our experiments, we could successfully create realistic images even if the viewpoint and the sunlight direction were changed. Although there are a few cases where the results are not very realistic, we were able to obtain good results in most cases. The supplemental video showing some applications of our method includes good results as well as not very realistic results (around 0'44" in the video). This problem could be addressed by using multiple photographs taken under different conditions. However, we suggest the simpler solution of estimating multiple sets of parameters at multiple key frames using multiple photographs. The parameters are then interpolated to render the clouds between the key frames. The examples shown in Figs. 13 and 15 use this method.

Some care needs to be taken in choosing the input photograph. In our method, we assume that the density of atmospheric particles is uniform. However, in the real world, the density decreases exponentially with the height from the ground. This sometimes results in a color variation of light incident on the clouds. Such color variation cannot be handled by our method. This is a typical failure case of our method and an example is shown in Fig. 16. As shown in the inset of Fig. 16, the color of the clouds in the photograph is reddish near the horizon but is white around the top of the clouds. This effect is not reproduced in the synthetic image rendered with the parameters estimated by our method (Fig. 16). This problem can be addressed by including the rate of the exponential decay of the density as an additional unknown variable. Except for the above problem, our method can find the parameters for any combination of input photograph and volume data. However, even if the histograms are similar, the synthetic clouds may not seem to be similar to the clouds in the photograph, when the cloud types are different, e.g., cumulus and cirrus.

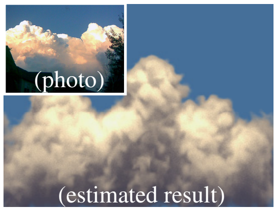


Figure 16: *A failure case.*

Our method does not take into account the physical validity in estimating the parameters. The parameters obtained by the method are simply the ones that can render synthetic clouds similar to those in the photograph. We believe that this is sufficient for many applications in computer graphics. Although we use photographs to determine the parameters in this paper, it is also possible to use images designed by the user. After rendering an initial image, the user can use image editing tools to modify the color, transparency, and contrast of the image. Then the edited image can be used to estimate the parameters using our method.

7 Conclusion and Future Work

We have proposed a method for solving the inverse rendering problem for clouds: estimating the parameters that can render a specified appearance. The difference in appearance is measured using color histograms. We used genetic algorithms to solve the inverse problem. We accelerated the computation by precomputing a set of intermediate images and by utilizing a GPU. Using our method, the optimal parameters can be found within a minute. We demonstrated the capability of our method using a set of examples.

One of our future works is to find an optimal sampling pattern for each of the parameters. Currently, the parameters estimated by genetic algorithms are sampled regularly. However, adaptive sampling would be more suitable for some of the parameters, such as the extinction cross section σ_t, since the intensity of the clouds is considered to be exponentially proportional to the change in those parameters. Finding the optimal sampling patterns would improve the performance of our method. Another future work is to make the parameters spatially variable. This makes it possible to adjust the local appearance of the clouds.

References

Bonneel, N., Paris, S., Panne, M. V. D., Durand, F., and Drettakis, G. 2009. Single photo estimation of hair appearance. *Computer Graphics Forum 28*, 4, 1171–1180.

Bouthors, A., Neyret, F., Max, N., Bruneton, E., and Crassin, C. 2008. Interactive multiple anisotropic scattering in clouds. In *Proceedings of ACM Symposium on Interactive 3D Graphics and Games*, 173–182.

Cerezo, E., Perez, F., Pueyo, X., Seron, F. J., and Sillion, F. X. 2005. A survey on participating media rendering techniques. *The Visual Computer 21*, 5, 303–328.

Davis, A. B., and Marshak, A. 2010. Solar radiation transport in cloudy atmosphere: a 3D perspective on observations and climate impacts. *Reports on Progress in Physics 73*, 2.

DOBASHI, Y., KUSUMOTO, K., NISHITA, T., AND YAMAMOTO, T. 2008. Feedback control of cumuliform cloud formation based on computational fluid dynamics. *ACM Transactions on Graphics 27*, 3, Article 94.

DOBASHI, Y., SHINZO, Y., AND YAMAMOTO, T. 2010. Modeling of clouds from a single photograph. *Computer Graphics Forum 29*, 7, 2083–2090.

EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., AND PERLIN, K. 2009. *Texturing and Modeling: A Procedural Approach Third Edition*. Morgan Kaufmann.

FATTAL, R. 2008. Single image dehazing. *ACM Transactions on Graphics 27*, 3, Article 72.

GOLDBERG, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Lerning*. Addison-Wesley Professional.

HARRIS, M. J., AND LASTRA, A. 2001. Real-time cloud rendering. *Computer Graphics Forum 20*, 3, 76–84.

HUBBALL, D., CHEN, M., AND GRANT, P. 2008. Image-based aging using evolutionary computing. *Computer Graphics Forum 27*, 2, 607–616.

JENSEN, H. W., AND CHRISTENSEN, P. H. 1998. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of ACM SIGGRAPH 1998*, 311–320.

KATZ, A., AND THRIFT, P. 1994. Generating image filters for target recognition by genetic learning. *PAMI 16*, 9, 906–910.

KAWAI, J. K., PAINTER, J. S., AND COHEN, M. F. 1993. Radioptimization – goal based rendering. In *Proc. ACM SIGGRAPH 1993*, 147–154.

KIENLE, A., LILGE, L., PATTERSON, M. S., HIBST, R., STEINER, R., AND WILSON, B. C. 1996. Spatially resolved absolute diffuse reflectance measurements for noninvasive determination of the optical scattering and absorption coefficients of biological tissue. *Applied Optics 35*, 13.

LALONDE, J.-F., EFROS, A. A., AND NARASIMHAN, S. G. 2012. Estimating natural illumination from a single outdoor image. *International Journal on Computer Vision 98*, 2, 123–145.

LI, H., AND YANG, C. Y. 1997. A genetic algorithm for inverse radiation problems. *International Journal of Heat and Mass Transfer 40*, 7, 1545–1549.

MARKS, J., ANDALMAN, B., BEARDSLEY, P. A., FREEMAN, W., GIBSON, S., HODGINS, J., KANG, T., MIRTICH, B., PFISTER, H., RUML, W., RYALL, K., SEIMS, J., AND SHIEBER, S. 1997. Design galleries: A general approach to setting parametes for computer graphics and animation. In *Proc. ACM SIGGRAPH 1997*, 380 – 400.

MIYAZAKI, R., DOBASHI, Y., AND NISHITA, T. 2002. Simulation of cumuliform clouds based on computational fluid dynamics. In *Proceedings of EUROGRAPHICS 2002 Short Presentations*, 405–410.

MUNOZ, A., MASIA, B., TOLOSA, A., AND GUTIERREZ, D. 2009. Single-image appearance acquisition using genetic algorithms. In *Proceedings of Computer Graphics, Visualization, Computer Vision and Image Processing 2009*, 24–32.

MUNOZ, A., ECHEVARRIA, J. I., LOPEZ-MORENO, J., SERON, F., GLENCROSS, M., AND GUTIERREZ, D. 2011. BSSRDF estimation from single images. *Computer Graphics Forum 30*, 2, 455–464.

NISHITA, T., DOBASHI, Y., AND NAKAMAE, E. 1996. Display of clouds taking into account multiple anisotropic scattering and sky light. In *Proceedings of ACM SIGGRAPH 1996*, 379–386.

OTSU, N. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybemetics 9*, 1, 62–66.

PELLACINI, F., BATTAGLIA, F., MORLEY, R. K., AND FINKELSTEIN, A. 2007. Lighting with paint. *ACM Transactions on Graphics 26*, 2, Article 9.

PEREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Transactions on Graphics 22*, 3, 313–318.

PREMOZE, S., ASHIKHMIN, M., TESSENDORF, J., RAMAMOORTHI, R., AND NAYAR, S. 2004. Practical rendering of multiple scattering effects in participating media. In *Proc. Eurographics Symposium on Rendering*, 52–63.

REINHARD, E., ASHIKHIMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Computer Graphics and Applications 21*, 5, 34–41.

SCHOENEMAN, C., DORSEY, J., SMITS, B., ARVO, J., AND GREENBERG, D. 1993. Painting with light. In *Proc. ACM SIGGRAPH 1993*, 143–146.

SCHPOK, J., SIMONS, J., EBERT, D. S., AND HANSEN, C. 2003. A real-time cloud modeling, rendering, and animation system. In *Proceedings of Symposium on Computer Animation 2005*, 160–166.

SIMS, K. 1991. Artificial evolution for computer graphics. In *Proc. ACM SIGGRAPH 1991*, 319–328.

SITTHI-AMORN, P., MODLY, N., WEIMER, W., AND LAWRENCE, J. 2011. Genetic programming for shader simplification. *ACM Transactions on Graphics 30*, 6, Article 152.

SNYMAN, J. 2005. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms (Applied Optimization)*. Springer.

STAM, J. 1995. Multiple scattering as a diffusion process. In *Proc. Eurographics Workshop on Rendering*, 41–50.

TAO, L., YUAN, L., AND SUN, J. 2009. Skyfinder: Attribute-based sky image search. *ACM Transactions on Graphics 28*, 3, Article 68.

WANG, J., ZHAO, S., TONG, X., LIN, S., LIN, Z., DONG, Y., GUO, B., AND SHUM, H.-Y. 2008. Modeling and rendering of heterogeneous translucent materials using the diffusion equation. *ACM Transactions on Graphics 27*, 1, Article 9.

YUE, Y., IWASAKI, K., CHEN, B.-Y., DOBASHI, Y., AND NISHITA, T. 2010. Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Transactions on Graphics 29*, 6, Article 177.

ZHANG, R., VERKRUYSSE, W., CHOI, B., VIATOR, J., JUNG, B., SVAASAND, L., AGUILAR, G., AND NELSON, J. 2005. Determination of human skin optical properties from spectrophotometric measurements based on optimization by genetic algorithms. *Journal of Biomedical Optics 10*, 2.

ZHOU, K., REN, Z., LIN, S., BAO, H., GUO, B., AND SHUM, H.-Y. 2008. Real-time smoke rendering using compensated ray marching. *ACM Transactions on Graphics 27*, 3, Article 36.