

# Modeling of Clouds from Satellite Images Using Metaballs

Yoshinori Dobashi<sup>†</sup>, Tomoyuki Nishita<sup>††</sup>, Hideo Yamashita<sup>†††</sup>, Tsuyoshi Okita<sup>†</sup>

<sup>†</sup>Hiroshima City University  
3-4-1 Ozukahigashi, Asaminami-ku,  
Hiroshima, 731-3194, Japan  
e-mail: {doba, okita}@im.hiroshima-cu.ac.jp

<sup>††</sup>Fukuyama University  
985 Sanzo, Higashimura-cho,  
Fukuyama, 729-0292, Japan  
e-mail: nis@eml.hiroshima-u.ac.jp

<sup>†††</sup>Hiroshima University  
1-4-1 Kagamiyama,  
Higashi-hiroshima, 739-8527  
e-mail: yama@eml.hiroshima-u.ac.jp

## Abstract

*This paper proposes an image-based modeling of clouds where realistic clouds are created from satellite images using metaballs. The intention of the paper is for applications to space flight simulators, the visualization of the weather information, and the simulation of surveys of the earth. In the proposed method, the density distribution inside the clouds is defined by a set of metaballs. Parameters of metaballs, such as center positions, radii, and density values, are automatically determined so that a synthesized image of clouds modeled by using metaballs is similar to the original satellite image. We also propose an animation method for clouds generated by a sequence of satellite images taken at some interval. The usefulness of the proposed method is demonstrated by several examples of clouds generated from satellite images of typhoons passing through Japan.*

## 1. Introduction

Recently, simulation of natural phenomena, such as water, smokes, fire, clouds, have been attempted in computer graphics[1][2][3][4][5][6][7]. Clouds play an important role when generating images of outdoor scenes, the earth viewed from outer space and the visualization of weather information. This paper proposes a method for modeling large scale clouds viewed from space such as typhoon. A lot of methods have been developed for modeling clouds and can be classified into two groups.

Methods in the first group create clouds by procedural modeling. Voss and Musgrave have tried to create realistic clouds using fractals[8][9]. Gardner has

produced realistic images of clouds by using Fourier synthesis[10]. However, this does not create a true three-dimensional geometric model. Ebert et al. have used solid texture to generate clouds in three dimensional space[11]. They have also developed a method combining metaballs and a noise function to model clouds[12]. Sakas has modeled clouds by using spectral synthesis[13], and Nishita et al. create clouds by generating metaballs using the idea of fractals[6]. Using these methods, however, creating clouds that the user pictures in the mind is very difficult since a lot of parameters have to be specified by trial and error. Stam and Fuime have developed a simple method for modeling clouds[14]. In their method, a user specifies density values at several points in three dimensional space. Then the density distribution of clouds is obtained by interpolating the specified density values. Although this method can create realistic clouds, it is impractical to create large scale clouds viewed from space. Methods for modeling a set of clusters of clouds have also been developed. Musgrave creates typhoon by the procedural approach[16]. Nishita et al. have modeled clouds to generate realistic images of the earth viewed from space[27]. In both these methods, however, clouds are simply modeled by applying 2D fractals. The color and shape of clouds change depending on both the viewpoint and the position of the sun. These methods cannot simulate such kind of effects.

Methods classified in the second group model clouds by simulating the physical process of the clouds forming[15][3][5]. In these methods, however, there is often a high computational cost when simulating clouds over the earth.

To address this problem, we propose a new method for modeling three dimensional clouds based on a satellite image. In recent years, many methods of

image-based rendering and/or modeling have been developed in computer graphics[17][18][19][20][21][22]. The proposed method makes use of the image-based approach to generate clouds. Lee et al. also use satellite images to generate clouds in order to visualize the weather information[23]. However, they use polygons to model them, resulting in bad-looking clouds. In the proposed method, the density distribution of clouds is represented by using metaballs. Parameters of metaballs (center positions, effective radii, and density values) are determined automatically so that a synthesized image of clouds modeled by metaballs coincides with the satellite image. Together with the satellite image, the proposed method makes use of the fact that the color of clouds can be calculated by integrating the scattered light due to particles in them. Using this method, clouds with a shape and color similar to real clouds can be automatically generated. Moreover, this paper also proposes an animation method for clouds generated by a sequence of satellite images. In order to demonstrate the usefulness of the method, it is applied to animations of the visualization of weather information and the earth viewed from space.

In the following sections, the basic idea of modeling clouds is described in section 2. Next, in section 3, the details of the modeling method are discussed. In section 4, the method for animating the generated clouds is proposed. And in section 5, several examples are demonstrated. Finally in section 6, conclusions of the paper and future works are discussed.

## 2. Basic Idea of Modeling Clouds from a Satellite Image

Various representations can be used for modeling clouds, such as voxels[15], procedural functions[12][16], and metaballs (or blobs)[5][6]. Using metaballs, clouds can be animated easily by moving their centers. Therefore, in this paper, metaballs are used to model them as shown in Fig. 1.

A density value at a point  $P$  in the cloud is given by the following equation.

$$\rho(P) \approx \sum_{j=1}^N q_j f(r_j, R_j), \quad (1)$$

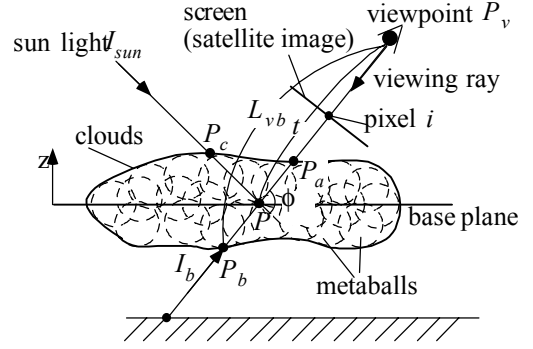


Figure 1: Geometry of clouds.

where  $N$  is the number of metaballs,  $q_j$  the density at the center of a metaball  $j$ ,  $f$  the field function, and  $R_j$  the radius of the metaball. And  $r_j$  is the distance between the point  $P$  and the center position of a metaball,  $C_j$ , that is,  $r_j = |P - C_j|$ . We use the field function proposed by Wyvill et al.[24]. That is,

$$f(r_j, R_j) = -4/9 \cdot (r_j / R_j)^6 + 17/9 \cdot (r_j / R_j)^4 - 22/9 \cdot (r_j / R_j)^2 + 1,$$

when  $r_j \leq R_j$ , otherwise 0.

Determining parameters of the metaballs is equivalent to solving an inverse problem of determining the density distribution inside the clouds so that an image of synthesized clouds is similar to a satellite image. The problem is, however, very complicated and hard to solve. Therefore, we assume that the multiple scattering can be neglected. Furthermore, for modeling clouds, the attenuation of light due to cloud particles is approximated as a constant. Despite these assumptions, there is no unique solution to the problem. Therefore, the parameters of the metaballs are heuristically determined as follows. First, each pixel of the satellite image is classified into either cloud region or background region. To do this, the satellite image is converted to a monotone one. Then pixels with intensities higher than a specified threshold are identified as clouds. Next, one metaball is added at the pixel with the maximum intensity in the cloud region. After that, its radius and density at the center are optimized and the approximated image is calculated using the clouds modeled by metaballs. Then a new metaball is added if the error between the satellite image and the approximated image is greater than a specified threshold. These processes are repeated until the error is less than the threshold.

The overview of the proposed method is as follows.

1. Classifying the pixels in the satellite image into either cloud regions or background regions.

2. Generating a metaball at the pixel with the maximum intensity in cloud regions of the satellite image.
3. Optimizing the radius and the density value at the center of the metaball.
4. Calculating the approximated image using the clouds modeled by metaballs.
5. If the error between the satellite image and the approximated image is less than a specified threshold, stop. Otherwise go to step 6 after calculating the image of difference between the satellite image and the approximated image.
6. Go to step 3 after generating a metaball at the pixel with the maximum intensity in clouds regions of the image of difference.

In the following sections, details of the method are discussed.

### 3. Modeling Clouds Using Metaballs

In this section, the method for determining parameters is proposed after describing the method for calculating the error between the satellite image and the synthesized image by using clouds modeled by metaballs.

#### 3.1 Calculating the Error between Satellite Image and Synthesized Image

The parameters of the metaballs are determined so that the mean square error between the satellite image and the synthesized image is minimized, that is:

$$Err = \frac{1}{M} \sum_{i=1}^M (I_i^{(true)} - I_i)^2 \rightarrow \min, \quad (2)$$

where  $M$  is the number of pixels in the satellite image, and  $I_i^{(true)}$  and  $I_i$  are the intensity of the satellite image and the intensity of the synthesized image at pixel  $i$ , respectively. In the following, the calculation method of  $Err$  is described.

As shown in Fig. 1, let the density be  $\rho(t)$  at a point  $P$  in the distance  $t$  from the viewpoint  $P_v$ . Then the intensity  $I_i$  is calculated by the following equation[25].

$$I_i(\lambda) = I_b(\lambda) \exp(-\tau(L_{vb}, \lambda)) + I_{sun}(\lambda) \beta(\phi, \lambda) \times \int_{P_a}^{P_b} \rho(t) \exp(-\tau(t, \lambda) - \tau(s, \lambda)) dt, \quad (3)$$

where  $I_b$  is the background color,  $\lambda$  the wave length,  $L_{vb}$  the distance between the viewpoint  $P_v$  and a point  $P_b$ ,  $\beta(\phi, \lambda)$  the phase function of cloud particles,  $\phi$  the phase angle between the viewing ray and the sun direction,  $I_{sun}$  the intensity of the sun light outside the atmosphere.  $\tau(L_{vb}, \lambda)$  is the optical length from  $P_v$  to  $P_b$ . The optical length is obtained by integrating the density of cloud

particles from  $P_v$  to  $P_b$ . Similarly,  $\tau(t)$  is the optical length from  $P_b$  to  $P$ ,  $\tau(s)$  from  $P_c$  to  $P$  (see Fig. 1). By putting Eq. 1 into Eq. 3, the following equation is obtained.

$$I_i(\lambda) = I_b(\lambda) \exp(-\tau(L_{vb}, \lambda)) + I_{sun}(\lambda) \beta(\phi, \lambda) \int_{P_a}^{P_b} \left\{ \sum_{j=1}^N q_j f_j(r_j, R_j) \exp(-\tau(t, \lambda) - \tau(s, \lambda)) \right\} dt, \quad (4)$$

where,

$$\tau(t, \lambda) = c \int_0^t \sum_{j=1}^N q_j f_j(r_j, R_j) dt, \quad (5)$$

and  $c$  is the extinction coefficient. Since it is difficult to determine the parameters of metaballs so that the intensity of the satellite image coincides with the intensity calculated by Eq. 4, it is approximated as follows. First, the wavelength is neglected since scattering of light due to cloud particles obeys Mie scattering and the spectrum of scattering is not much influenced. Therefore, the satellite image is converted to a monotone one. Second, the attenuation terms  $\exp(-\tau(L_{vb}, \lambda))$  and  $\exp(-\tau(t, \lambda) - \tau(s, \lambda))$  in Eq. 4 are approximated as constants  $\kappa_1$  and  $\kappa_2$ . Note that these approximations are only for modeling clouds. Then, the following equation is obtained.

$$I_i = I_b \kappa_1 + I_{sun} \beta(\phi) \kappa_2 \int_{P_a}^{P_b} \sum_{j=1}^N q_j f(r_j, R_j) dt = I_b \kappa_1 + I_{sun} \beta(\phi) \kappa_2 \sum_{j=1}^N q_j F(r_j, R_j), \quad (6)$$

$(i = 1, \dots, M)$

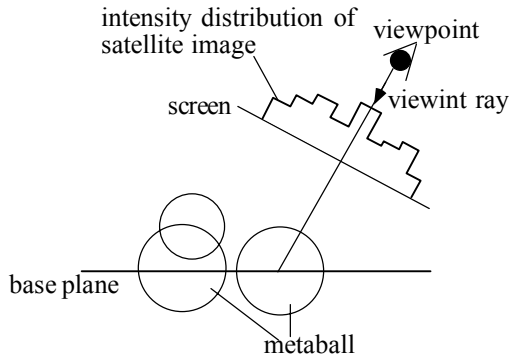
where,

$$F(r_j, R_j) = \int_{P_j^{in}}^{P_j^{out}} f(r_j, R_j) dt, \quad (7)$$

$P_j^{in}$  and  $P_j^{out}$  the intersection point of the viewing ray with the metaball  $j$ .  $F$  is the integrated function of the field function  $f$  and can be calculated analytically[26]. To obtain the background intensity,  $I_b$ , a satellite image without clouds is required. In this paper, it is obtained by assembling the background regions of a set of satellite images by hand. Then, the intensity,  $I_i$ , of the synthesized image can be calculated by Eq. 6. This can be calculated incrementally every time a new metaball is added. Let us assume  $I_i^{(old)}$  is the intensity before adding the metaball  $k$  and  $I_i^{(new)}$  the intensity after adding it. Then,

$$I_i^{(new)} = I_i^{(old)} + I_{sun} \beta(\phi, \lambda) q_k F(r_k, R_k). \quad (8)$$

The second term in Eq. 8 is evaluated at the pixels



**Figure 2: Determination of center of metaballs.**

through which the viewing rays intersect the metaball  $k$ . Therefore,  $I_i$  can be calculated with a small computation cost, resulting in faster the optimization processes for the parameters.

Using  $I_i$  obtained by the above method, the error,  $Err$ , between the satellite image and the synthesized image is calculated. Metaballs are added repeatedly until the error is less than a specified tolerance.

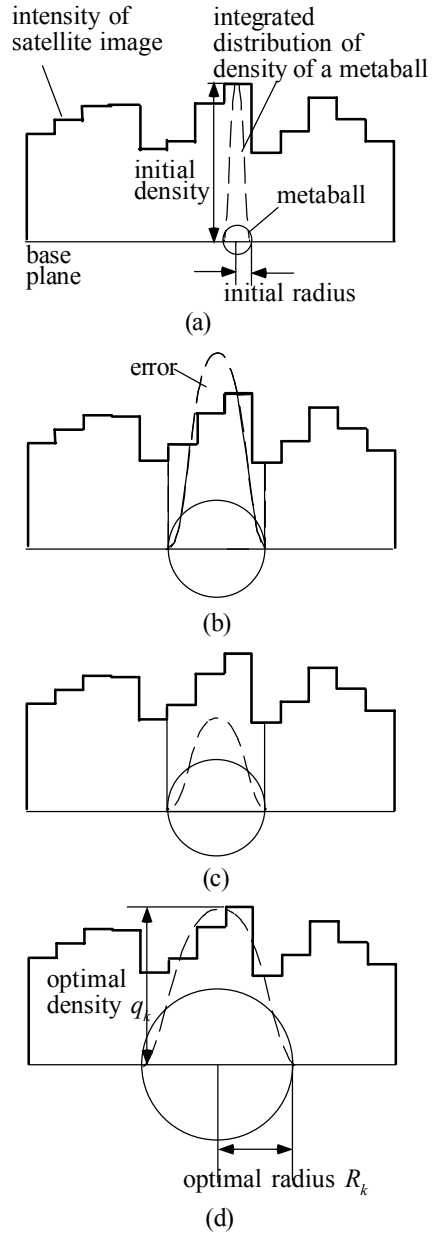
### 3.2 Determining Center of Metaball

When the number of metaballs becomes large, the computation for generating images takes a long time. Therefore, the number of metaballs should be as small as possible. Depending on the position of the center of each metaball, the number of metaballs changes for representing the clouds within the tolerance. The integrated function  $F$  of the field function  $f$  has a peak value at  $r_j=0$  and decreases gradually as  $r_j$  increases, finally becoming zero at  $r_j=R_j$ . The number of metaballs tends to be small when the center position is chosen where the function  $F$  fits the intensity distribution of the satellite image. The reason for this is that the intensity distribution of cloud images is smooth and the distribution around the pixel with the maximum intensity is considered to be similar to the distribution of the function  $F$ . Therefore, the center position is determined on the viewing ray through the pixel with the maximum intensity. In order to determine the center position in three dimensional space, a base plane is set by the user (see Fig. 2). The center point of the metaball is set to the intersection point of the viewing ray with the base plane. When the metaballs are already placed at the intersection point, the intersection point of the viewing ray with the metaballs is calculated and the center is set to the nearest point as shown in Fig. 2. After one metaball is generated, the image of the difference is calculated. Then it is used for determining the center of

the next metaball in order to avoid the same pixel being chosen for the center point.

### 3.3 Optimization of Radius and Density at Center

The basic idea for calculating radius  $R_k$  and density  $q_k$  at the center of the metaball  $k$  is to search for the optimal values by changing them little by little. Fig. 3 shows the process for determining the radius and density. In Fig. 3,



**Figure 3: Optimization steps for the density and radius.**

the base plane is parallel to the image plane and the dotted curve indicates the function,  $F(q_k, R_k)$ . For this

search, the upper limit  $R^*$  for the radius is specified by the user.

The radius  $R_k$  and the density  $q_k$  are determined to minimize the difference of intensity between the synthesized image and the satellite image at pixels inside the radius of the metaball. In Fig. 3, the shaded areas indicate the difference. First, an initial radius and density are set to  $d$  and  $I_i / F(0.0, \delta)$ , respectively (Fig. 3(a)). That is, the integrated value of the field function along the ray through the center of the metaball coincides with the intensity of the pixel corresponding to the center. Next, the density is fixed and the optimal radius that minimize the error is calculated by incrementing the radius by  $d$  (Fig. 3(b)). Then, after making the density smaller by  $w$ , the optimal radius is searched for again (Fig. 3(c)). By making the density smaller, the radius can be larger, hence one metaball can cover a larger area. This results in reducing the number of metaballs. This process is repeated until the optimal radius and density are found (Fig. 3(d)). In this paper,  $d$  is set to the size of one pixel and  $w$  is specified by the user. Note that a metaball is deleted if the density becomes zero while searching.

#### 4. Animation of Clouds

In this section, an animation method for clouds is proposed. First, clouds are modeled from satellite images taken at a certain intervals. Then they are animated by interpolating the parameters of the metaballs.

In traditional morphing techniques, the user specifies

segments. However, this method is not applicable to the animation of clouds. Clouds in one image move according to the flow of the atmosphere, resulting in the clouds in the next image. Therefore, their movement cannot be expressed just by interpolating the feature points. Although the flow analysis of the atmosphere is required for obtaining the exact movement of clouds, this computation is too expensive. So, in the proposed method, the user inputs the flow as Bezier curves by observing a sequence of satellite images. Fig. 4(a) shows an example of flow curves specified by the user. Using these, a flow vector at the center of a metaball is calculated as follows (see Fig. 4(b)). First, the minimum distance  $d_k$  and closest point  $P_k$  from the center point to each Bezier curve are calculated. If the distance  $d_k$  is less than a specified distance  $D$ , the Bezier curve affects the movement of the metaball. Then tangent vector  $\mathbf{v}_k$  at the closest point is calculated. The flow vector  $\mathbf{v}$  at the point is defined as the weighted sum of the tangent vectors, that is,  $\mathbf{v} = \sum_{k=1}^n w_k \mathbf{v}_k$ .  $n$  is the number of Bezier curves inside the domain of influence. The weights,  $w_k$ , is calculated as follows.

First, contribution ratio,  $a_k$ , is calculated by:

$$a_k = \begin{cases} 1.0 - \left(\frac{d_k}{D}\right)^2 & (d_k \leq D) \\ 0.0 & (d_k > D) \end{cases} \quad (9)$$

Then the weight,  $w_k$ , is calculated by normalizing  $a_k$ , that is,

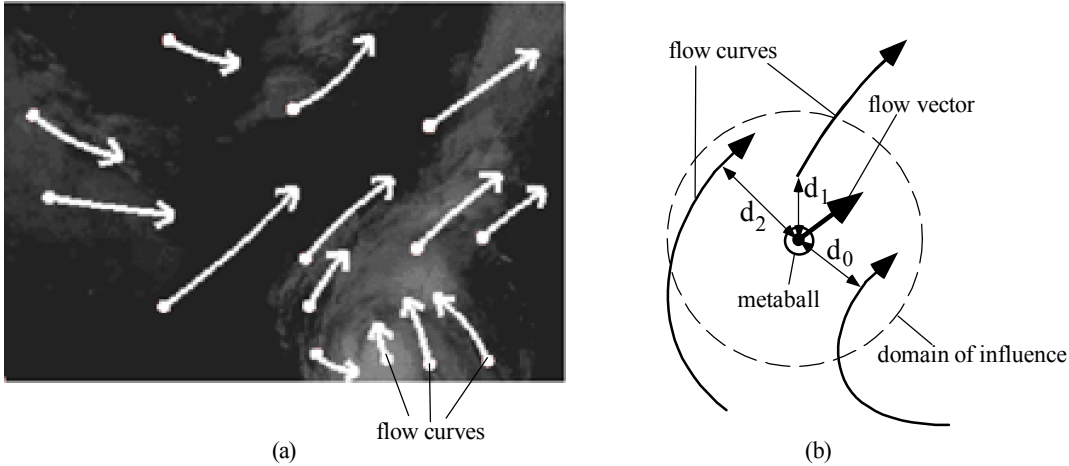


Figure 4: Calculation of flow vectors.

the features of images by using points or line segments and intermediate images are generated by taking correspondence between these feature points or line

$$w_k = a_k / \sum_{j=1}^n a_j . \quad (10)$$

Each metaball moves toward the flow vector  $\mathbf{v}$  at the center of the metaball. In order to realize the smooth transition from one image to the next image, each metaball is faded in and out by changing the density at the center. If all the metaballs are faded in/out at the same time, one can see “cross dissolve” in the transition. To avoid this, different ones are faded in and out randomly at different times.

## 5. Examples

Fig. 5 shows examples of clouds using satellite images viewed from space. Fig. 5(a) is an infrared image of a typhoon taken from the meteorological satellite, “HIMAWARI”. Fig. 5(b) shows clouds modeled by metaballs. By comparing Fig. 5(a) with Fig. 5(b), it can be seen that the proposed method can generate clouds similar to those in satellite images. Fig. 5(c) shows the same clouds viewed from a different viewpoint. Since clouds are modeled in three dimensional space, they are seen from different viewpoints and shadows are cast on Japan. Note that, to generate Figs. 5(b) and (c), the color of the clouds are calculated by taking into account the single scattering of light, i.e. by using Eq. 4. As shown in Figs. 5(c). The infrared image used here is available on the internet<sup>1</sup>, and was recorded at 18:00 on September 15, 1995.

In this example, the number of metaballs is 8506 and the calculation time for generating the clouds is 13 seconds using Silicon Graphics Indigo2 MAXIMUM IMPACT (195 MHz MIPS R10000). The calculation time for generating Fig 5(b) is 3.7 minutes on the same machine.

Next, the proposed method is applied to the making of animations. We made two. For these animations, thirteen satellite images were used to generate clouds and nine hundred images were produced by using the method described in section 4. Fig. 6 shows images from the one, entitled “Weather Report”, that visualizes weather information. In this animation, the movement of a typhoon passing across Japan is visualized. These examples show that the proposed method can generate realistic clouds even if the viewpoint is varied.

Fig. 7 shows examples of the animation of the earth viewed from space. In this animation, the position of the sun changes. The color of the earth is calculated by taking into account the effect of the scattering of both sunlight and sky light due to the atmosphere by using the method developed by Nishita et. al.[27]. In Fig. 7(a), the

sun is behind the viewer, turning the color of the clouds red. In Figs. 7(b) and (c), the sun is above the clouds. And finally in Fig. 7(d), the sun is hidden behind the earth. These examples show that the optical effects can be simulated since 3D clouds are generated. These effects cannot be simulated just by mapping the satellite image as textures.

## 6. Conclusions

By using metaballs, a method for modeling clouds in three dimensional space from a single satellite image has been proposed. By making use of a satellite image of real clouds, the proposed method can generate clouds with authentic shape and color. We have also proposed an animation method for clouds modeled by metaballs. Using the proposed method, 3D clouds are generated and they can be viewed from arbitrary directions. The color of the clouds can be calculated by simulating the optical effects. Furthermore, to animate clouds, they are generated from a series of satellite images and shapes of clouds at each frame are obtained by interpolating them.

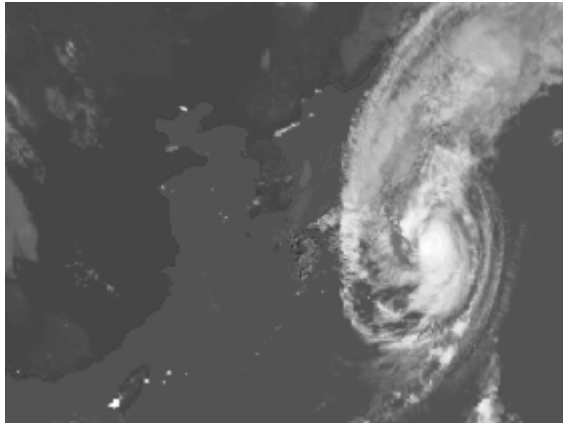
There are several future projects that remain: (1) In order to increase realism by adding various plausible fine features, some kind of turbulence model such as the one used in [14] or [12] should be included. (2) To model clouds more accurately, methods for the estimation of the height of clouds from satellite images[23] should be incorporated. (3) Extending the proposed method to generate clouds from photographs taken from the ground will be useful for modeling clouds such as cumulonimbi.

## References

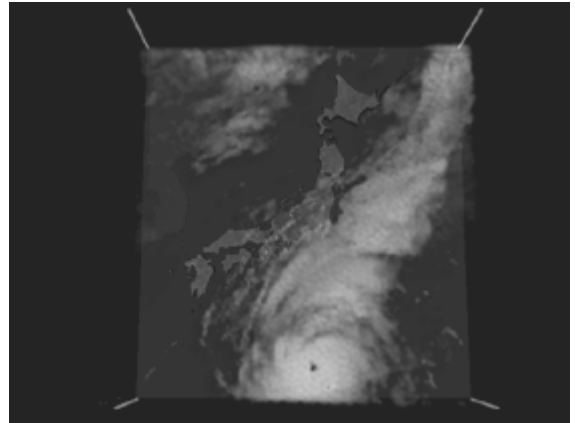
- [1] R. V. Klassen, “Modeling the Effect of the Atmosphere on Light,” *ACM Trans. on Graphics*, vol.6, No.4, pp. 215-237 (1987).
- [2] K. Kaneda, G. Yuan, E. Nakamae, T. Nishita, “Photorealistic Visual Simulation of Water Surfaces Taking into account Radiative Transfer,” *Proc. of CG&CAD'91 (China)*, pp. 25-30 (1991).
- [3] J. Stam, E. Fiume, “Turbulent Wind Fields for Gaseous Phenomena,” *Proc. of SIGGRAPH'93*, pp. 369-376 (1993).
- [4] N. Max, “Efficient Light Propagation for Multiple Anisotropic Volume Scattering,” *Proc. of the Fifth Eurographics Workshop on Rendering*, pp. 87-104 (1994).
- [5] J. Stam, E. Fiume, “Dipicting Fire and Other Gaseous Phenomena Using Diffusion Processes,” *Proc. of SIGGRAPH'95*, pp. 129-136 (1995).
- [6] T. Nishita, Y. Dobashi, E. Nakamae, “Display of Clouds Taking into Account Multiple Anisotropic Scattering and Sky Light,” *Proc. of SIGGRAPH'96*,

<sup>1</sup> Several sites provide satellite images. For example, <http://www.jwa.go.jp/gms.html>.

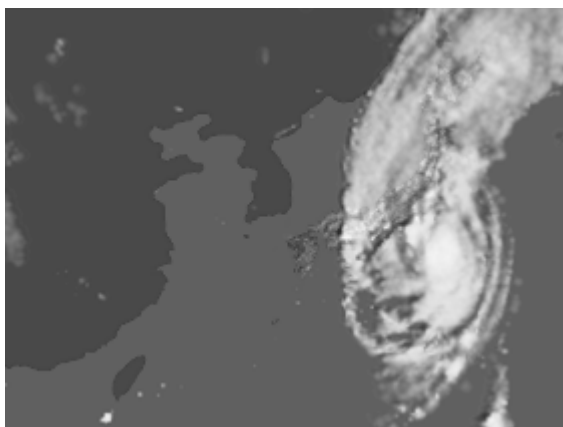
- pp. 379-386 (1996).
- [7] N. Foster, D. Metaxas, "Modeling the Motion of a Hot, Turbulent Gas," Proc. of SIGGRAPH'97, pp.181-188 (1997).
- [8] R. Voss, "Fourier Synthesis of Gaussian Fractals: 1/f noises, landscapes, and flakes," SIGGRAPH'83: Tutorial on State of the Art Image Synthesis, Vol. 10 (1983).
- [9] F. K. Musgrave, "A Note on Ray Tracing Mirages," IEEE Computer Graphics and Applications, Vol. 10, No. 6, pp. 10-12 (1990).
- [10] G.Y. Gardener, "Visual Simulation of Clouds," Computer Graphics, Vol.19, No.3, pp. 279-303 (1985).
- [11] D. S. Ebert and R. E. Parent, "Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-Buffer Techniques," Computer Graphics, Vol. 24, pp. 357-366 (1990).
- [12] D. S. Ebert, "Volumetric Modeling with Implicit Functions: A Cloud is Born," Visual Proceedings, SIGGRAPH'97, pp. 147 (1997).
- [13] G. Sakas, "Modeling and Animating Turbulent Gaseous Phenomena Using Spectral Synthesis," The Visual Computer, Vol. 9, No. 4, pp. 200-212 (1993).
- [14] J. Stam, E. Fiume, "A Multiple-Scale Stochastic Modelling Primitive," Proc. of Graphics Interface'91, pp. 24-31 (1991).
- [15] J. T. Kajiya and B. P. V. Herzen, "Ray Tracing Volume Densities," Computer Graphics, Vol. 18, pp. 165-174 (1984).
- [16] D. S. Ebert, F. K. Musgrave, P. Peachey, K. Perlin, and S. Worley, "Texturing and Modeling: A Procedural Approach," AP Professional, pp. 267-294 (1994).
- [17] S. M. Seitz and C. R. Dyer, "View Morphing," Proc. of SIGGRAPH'96, pp. 21-30 (1996).
- [18] M. Levoy and P. Hanrahan, "Light Field Rendering," Proc. of SIGGRAPH'96, pp. 31-42 (1996).
- [19] S. J. Gortler, R. Grzeszczuk, R. Szeliski, M. F. Cohen, "The Lumigraph," Proc. of SIGGRAPH'96, pp. 43-54 (1996).
- [20] Y. Horry, K. Anjyo, K. Arai, "Tour Into Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image," Proc. of SIGGRAPH'97, pp.225-232 (1997).
- [21] D. Terzopoulos and A. Witkin, "Physically-based Models with Rigid and Deformable Components," IEEE Computer Graphics and Applications, Vol. 8, No. 6, pp. 41-51 (1988).
- [22] P. E. Debevec, C. J. Taylor, J. Malik, "Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach," Proc. of SIGGRAPH'96, pp. 11-20 (1996).
- [23] C. A. Lee, C. Kesselman, S. Schwab, "Near-Real-Time Satellite Image Processing: Metacomputing in CC++," IEEE Computer Graphics and Applications, Vol. 16, No. 4, pp. 79-84 (1996).
- [24] G. Wyvill and A. Trotman, "Ray-Tracing Soft Objects," Proc. of CG International, pp. 439-475 (1990).
- [25] K. Kaneda, T. Okamoto, E. Nakamae, T. Nishita, "Photorealistic Image Synthesis for Outdoor Scenery under Various Atmospheric Conditions," The Visual Computer, Vol.7, No.5&6, pp. 247-258 (1991).
- [26] T. Nishita and E. Nakamae, "A Method for Displaying Metaballs by using Bezier Clipping," Computer Graphics Forum, Vol. 13, No. 3, pp. 271-280 (1994).
- [27] T. Nishita, S. Takao, T. Katsumi, E. Nakamae, "Display of The Earth Taking into Account Atmospheric Scattering," Proc. of SIGGRAPH'93, pp. 175-182 (1993).



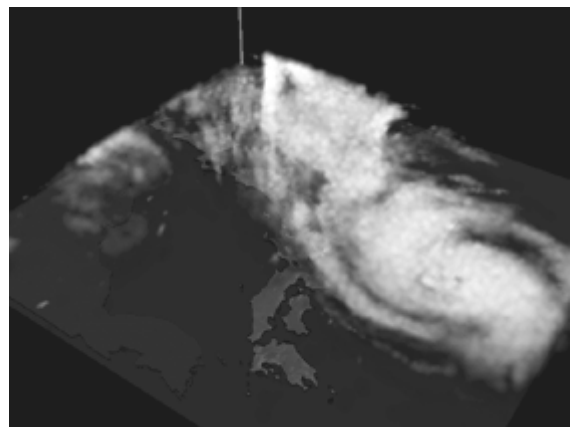
(a) a satellite image.



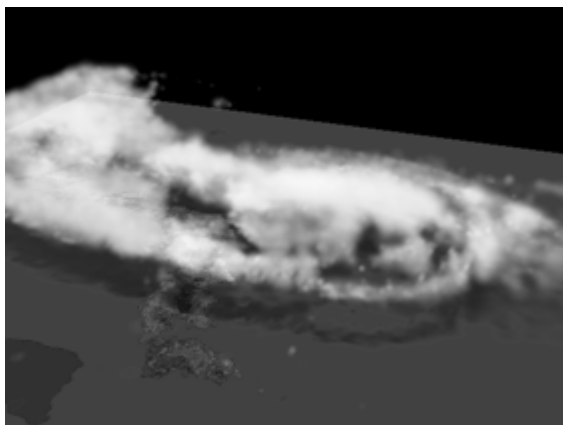
(a)



(b) clouds modeled by metaballs.



(b)



(c) clouds viewed from different viewpoint.

Figure 5: Clouds generated from a satellite image.

Figure 6: Examples of the visualization of weather information.