A Controllable Method for Animation of Earth-scale Clouds

Yoshinori Dobashi
Hokkaido University
doba@nis-ei.eng.hokudai.ac.jp
http://nis-ei.eng.hokudai.ac.jp/~doba

Tsuyoshi Yamamoto
Hokkaido University
yamamoto@nis-ei.eng.hokudai.ac.jp
http://nis-ei.eng.hokudai.ac.jp/~yamamoto

Tomoyuki Nishita
The University of Tokyo
nis@is.s.u-tokyo.ac.jp
http://nis-lab.is.s.u-tokyo.ac.jp/~nis/

Abstract

Computer generated images of the earth are often used for space flight simulators, computer games, movies, and so on. Clouds are indispensable to the creation of realistic images in these applications. This paper proposes a method for animating clouds surrounding the earth. The method allows the user to control the motion of clouds by specifying the center positions of high/low atmospheric pressure areas on the earth's surface. This data is used as input data and a three-dimensional velocity field is then calculated by solving Navier-Stokes equations. Water vapor is advected along the velocity field. Clouds are then generated due to the phase transition from water vapor to water droplets. We also propose an interactive system that enables the user to interactively control the simulation. The final photorealistic images are rendered by taking into account optical phenomena such as the scattering and absorption of light due to cloud particles.

Keywords: Earth-scale Clouds, Animation, Natural Phenomena, Atmospheric Fluid Dynamics.

1. Introduction

Clouds have a significant visual impact when the earth is viewed from space. In this context, the display of clouds is indispensable when creating realistic images of the earth for space flight simulators, computer games and so on. In particular, in a recent movie entitled "The Day After Tomorrow," the impressive motion of the earth-scale clouds was simulated. These applications imply that a controllable method is required for modeling the earth-scale clouds. One of the most popular approaches is to use two-dimensional texture mapping techniques [1]. Clouds are displayed by mapping textures onto a spherical shell surrounding the earth. The textures are often generated by using procedural techniques or by using satellite images. When using procedural techniques, the user can generate clouds at a low computational cost. However, a tedious trial-and-error process is often necessary to generate the realistic motions of clouds. On the other hand, realistic images can be generated by using satellite images. However, in this case, we cannot control the motions of clouds.

There are several methods for generating a three-dimensional density distribution of clouds [2] [3] [4] [5] [6] [7]. However, these methods are designed to create clouds formed in a local region, that is, over several kilometers. Realistic clouds of the earth can be generated by numerical simulation of atmospheric fluid dynamics that is employed by the numerical weather prediction. However, this method is extremely complicated and the computational cost is very high. Furthermore, it is almost impossible to control the motion of clouds by using these methods.

This paper proposes a controllable method for animating earth-scale clouds. To address the problems mentioned above, we carefully investigated the essential factors of the atmospheric fluid flow that are important for the visual simulation. Based on this investigation, we propose a simple and efficient computational model for the cloud formation. The proposed model is basically derived from the existing model for cloud formation in the local region [3]. We extend this to simulating earth-scale clouds. The extensions include the Coriolis effect, the friction between the atmosphere and the earth's surface, and the formation of raindrops. Furthermore, we designed our method so that the controllability and the reality are balanced. Although our method does not simulate the actual atmospheric fluid dynamics completely, our method reproduces distinctive features of the atmospheric flow such as vortex formation around the low pressure area. The user can control the motion of clouds by specifying the atmospheric pressures on the earth's surface. Then our method computes the detailed shapes and motions of clouds. In reality, the surface pressures are determined as a result of the complex earth's system. By contrast, our method calculates three-dimensional flows based on the specified two-dimensional pressures on the earth's surface. Based on our method, we have constructed an interactive visual simulation system. This system allows the user to interactively generate the desired motions of the earth's clouds. The final photorealistic images are created by taking optical phenomena into account.

The contributions of our method are:

1. Extensions of the local cloud model [3] to simulating the earth-scale clouds.
2. Simulation of circulation of water of the earth to realize formation and extinction of clouds.
3. Controllability of the cloud motion by specifying atmospheric pressures on the earth's surface.

The paper is organized as follows. In Section 2, the previous work related to our method is reviewed. Section 3 provides the basic concept of our method. Section 4 proposes the simulation method for creating earth's clouds and Section 5 proposes our interactive visual simulation system. The usefulness of our method is demonstrated with examples shown in Section 6. Finally, Section 7 concludes the paper.

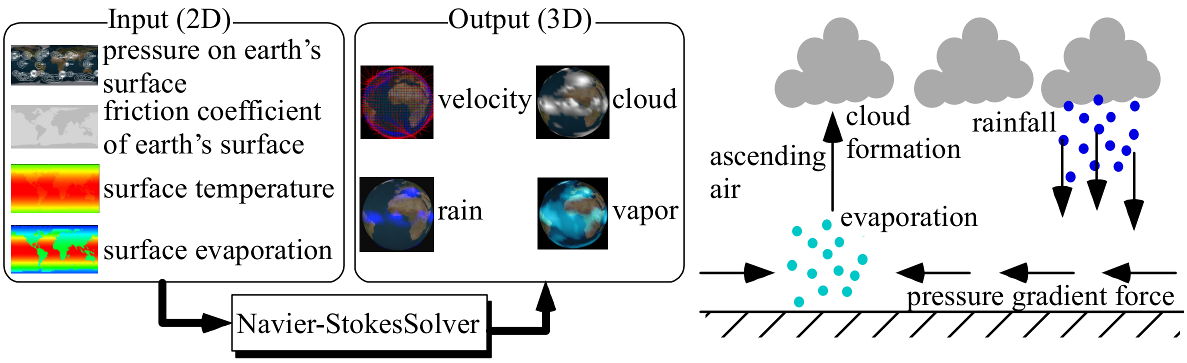
2. Related Work

The previous methods for modeling clouds can be classified into two categories. One uses procedural techniques and the other is based on the physical simulation.

Methods using procedural techniques use the idea of fractals to generate density distribution of clouds. In [1], several algorithms are introduced to create 2D textures, representing the clouds of several types of planets such the Earth and the Venus. Nishita et al. make use of these ideas to generate the earth's clouds [8]. Although the computational cost for procedural techniques is very low, a trial and error process is often required to generate realistic clouds.

Methods based on the physical simulation generate clouds by numerical simulation of the atmospheric fluid dynamics [2] [3] [4] [5] [6]. Although these methods can generate realistic clouds, they focus on the simulation of small-scale clouds formed over a distance of a several kilometers. These methods do not take into account several factors that are important in simulating earth-scale clouds. Yaeger and Upson [9] proposed a method for simulating the motion of the atmosphere of Jupiter. However, their method cannot generate three-dimensional flow since they use a two-dimensional version of the Navier-Stokes equations. Furthermore, their method does not take into account the physical phenomena relating to cloud formation.

Methods developed in the field of remote sensing enable us to create three-dimensional clouds by using satellite images (e.g., [10] [11]). Dobashi et al. employed such techniques to generate clouds surrounding the earth [12]. However, since any estimation of density inside the clouds is a difficult problem to solve, the clouds generated by these methods are represented by a height field indicating the



(a) Simulation process (b) Cloud formation process

Figure 1: Basic concept of our simulation method.

cloud top. These methods cannot generate three-dimensional motions and shapes of the clouds.

3. Basic Idea of Our Method

Fig. 1 shows the overview of our method. As input data, our method requires four distributions of: 1) atmospheric pressure on the earth's surface, 2) frictional coefficients of the earth's surface, 3) surface temperature, 4) evaporation rate from the earth's surface. These are provided as two-dimensional maps as shown in Fig. 1(a). In the following, we denote these maps as a pressure map p_0, a friction map μ_0, a temperature map E_0, and an evaporation map w_0. Using these maps, our method generates four three-dimensional distributions of: 1) cloud density w_{cloud}, 2) density of raindrops w_{rain}, 3) density of water vapor w_{vapor}, 4) velocity $\mathbf{u}$ (see Fig. 1(a)). The basic concept of the proposed method is described in the following (see Fig. 1(b)).

First, pressure gradient forces at the earth's surface are calculated by using the pressure map p_0. By using the forces as external forces, the three-dimensional velocity distribution, $\mathbf{u}$, is calculated by solving the Navier-Stokes equations (NS equations for short). Frictional forces determined from the frictional map μ_0 act on the atmosphere near the earth's surface. In addition, the water vapor is provided to the atmosphere near the surface by reference to the evaporation map w_0 and the temperature map E_0. The water vapor is advected along the velocity distribution $\mathbf{u}$. Clouds are formed by the phase transition from water vapor to water droplets. We further simulate the formation of raindrops to realize the extinction of clouds. We assume that the amount of raindrops generated is proportional to the amount of the clouds. The raindrops fall at a terminal speed and finally absorbed by the ground or sea. The progress of the simulation is visualized in real-time via our interactive system. The user can control the simulation by using the system. The final, photorealistic images are generated off-line by taking into account the scattering and absorption of light due to cloud particles. The effects of atmospheric scattering are also simulated, using the method described in [8].

4. Simulation of Cloud Formation

This section describes our simulation method by using Fig. 2. For Clarity, we describe only the simulation concept. The detailed formulation of our model is presented in Appendix. In our method, the six physical phenomena shown in Fig. 2(a) are simulated. In the following, the calculation method for the velocity distribution of the atmosphere is presented first. Then, the method for simulating cloud formation process is presented.

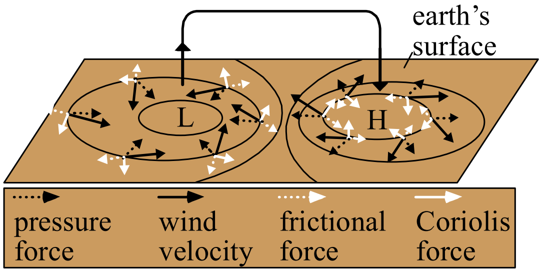
4.1 Computation of Velocity Field

The velocity distribution in the atmosphere is calculated by solving NS equations, taking into account the phenomena 1) through 3) shown in Fig. 2(a).

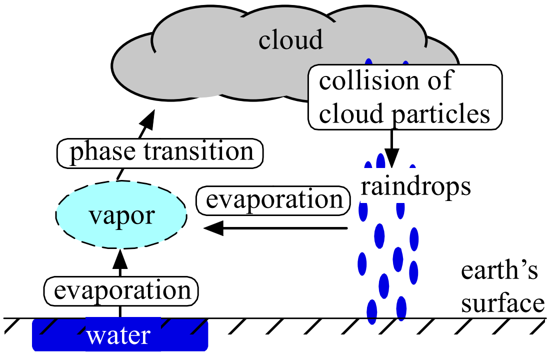
First, to take into account phenomenon 1) in Fig. 2(a), pressure gradient forces, $\mathbf{f}_p$, are calculated by using the pressure map, p_0 and are used as external forces. As shown by the black dotted arrows in Fig. 2(b), the pressure gradient force, $\mathbf{f}_p$, acts towards the gradient of the pressure on the earth's surface and is

1) Motion of the atmosphere according to the pressure on earth's surface
2) Coriolis effect due to the rotation of the earth
3) Deceleration due to friction between the atmosphere and the ground
4) Advection of water vapor and clouds along the atmospheric flow
5) Phase transition from water vapor to water droplet (i.e. cloud)
6) Formation of raindrop

(a) Physical phenomena taken into account in our simulation method.



(b) Computation of 3D atmospheric flow using surface pressure. L and H indicate centers of low and high atmospheric pressure areas, respectively.



(c) Simulation of water circulation.

Figure 2: Simulation concept for atmospheric fluid dynamics including cloud formation.

perpendicular to the isobaric lines of the surface pressure. $\mathbf{f}_p$ acts only in the horizontal direction. By solving NS equations, the atmosphere near the earth's surface converges towards the low pressure area and then an ascending air current is created around the center of the low pressure area (see Fig. 2(b)). When the ascending air reaches the top of the troposphere, it disperses and the atmosphere begins to descend in the high pressure area. In this way, a three-dimensional flow field is formed circulating between the low pressure area and the high pressure area (see the black solid arrows in Fig. 2(b)).

Since the earth rotates, the Coriolis force, $\mathbf{f}_{cori}$, (phenomenon 2 in Fig. 2(a)) begins to operate once the atmosphere starts to move. The magnitude of the Coriolis force is proportional to the magnitude of the velocity. Its direction is perpendicular to the movement of the atmosphere, as shown in Fig. 2(b) (white solid arrows). Fig. 2(b) shows the Coriolis force in the northern hemisphere. In the southern hemisphere, the force is in the opposite direction. Due to the Coriolis force, a counterclockwise vortex is generated around the low pressure area in the northern hemisphere (see Fig. 2(b)).

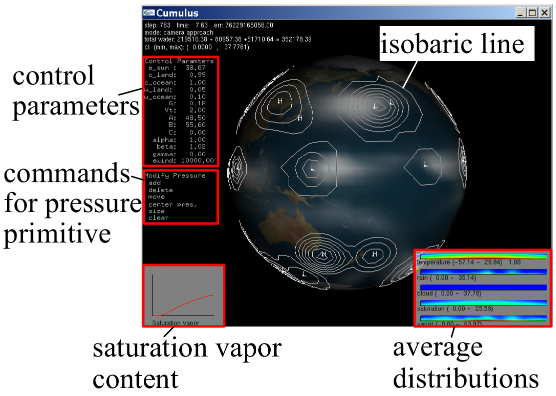
When the atmosphere moves close to the earth's surface, its speed is decreased due to the friction between the atmosphere and the surface of the earth (phenomenon 3 in Fig. 2(a)). This effect is simulated as an external force, $\mathbf{f}_{fric}$, that acts in an inverse direction to the velocity of the atmosphere (see white dotted arrows in Fig. 2(b)). The magnitude of $\mathbf{f}_{fric}$ is proportional to the magnitude of the velocity. The coefficient of proportion is obtained by using the friction map μ_0. In our method, we assume that the frictional force reduces only the horizontal component of the velocity and does not affect the vertical component.

The three-dimensional velocity distribution, $\mathbf{u}$, of the atmosphere is obtained from the numerical simulation of the NS equations, using the above three forces as external forces. For the numerical simulation, we generated three-dimensional grids in the thin volume representing the atmosphere over the earth's surface. These grids were generated by dividing the volume uniformly, using the polar coordinate system. The numerical solver is based on the method described in [13]. We extend the method to handling non-rectangular grids. Please refer to a standard textbook of CFD, such as [14] for the detail of the extension.

4.2 Cloud Formation

Clouds are generated by simulating the phenomena 4) through 6) shown in Fig. 2(a). Together with the formation of clouds, our method simulates the circulation system of water as shown in Fig. 2(c). The details are described in the following.

First, water contained in the earth evaporates into the air. The higher the surface temperature,



```
step: 763   time:   7.63   err: 76229165056.00
mode: camera approach
total water: 219510.38 + 80957.38 +51710.64 = 352178.39
cl  (min, max): (  0.0000  ,   37.7761)
Control Paramters
 e_sun :   38.87
  c_land:    0.99
c_ocean:    1.00
w_land:     0.05
w_ocean:    0.10
      G:    0.18
     Vt:    2.00
      A:   48.50
      B:   55.60
      C:    0.00
  alpha:    1.00
   beta:    1.02
  gamma:    0.00
  ewind: 10000.00
Modify Pressure
 add
 delete
 move
 center pres.
 size
 clear
temperature (-57.14 -  29.84)   1.00
rain (  0.00 -  35.14)
cloud (  0.00 -  37.78)
saturation (  0.00 -  25.59)
vapor (  0.00 -  63.97)
```

Figure 3: Our interactive visual simulation system.

the more the water evaporates. Therefore, we determine the amount of evaporation by the product of the evaporation map w_0 and the temperature map E_0. The water vapor is advected according to the velocity distribution, $\mathbf{u}$. We assume that the temperature of the atmosphere decreases in proportion to the altitude from the earth's surface. It is known that this property holds, on average, in the troposphere where clouds are formed. As the water vapor moves upward, the temperature drops and the atmosphere containing the water vapor finally reaches the dew point. It is then that the phase transition from water vapor to water droplets occurs, that is, clouds are generated. The amount of clouds generated is determined by a function called the saturation vapor content, representing the maximum amount of water vapor that an air parcel can contain. We use the function proposed by [3] for the saturation vapor content. Clouds are likely to be formed around low pressure areas, since ascending air currents are generated around the low pressure areas.

The extinction of clouds is realized by simulating the formation of raindrops. In our method, the amount of the raindrops generated is assumed to be proportional to the amount of cloud particles. The raindrops fall at a terminal velocity whose magnitude is proportional to the amount of raindrops. Whilst the raindrops are falling, some of them evaporate into the surrounding air. The rest of them, reaching the earth's surface, are absorbed. By taking into account the formation and absorption of the raindrops, our method is able to simulate the circulation of water, maintaining the total amount of water (water vapor + clouds + raindrops) roughly constant.

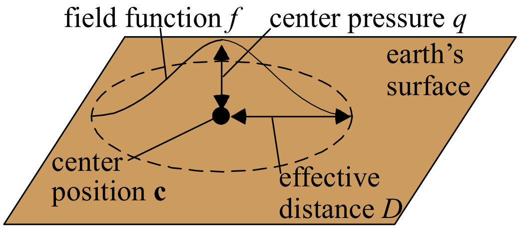


Figure 4: Pressure primitive.

5. Interactive Visual Simulation System

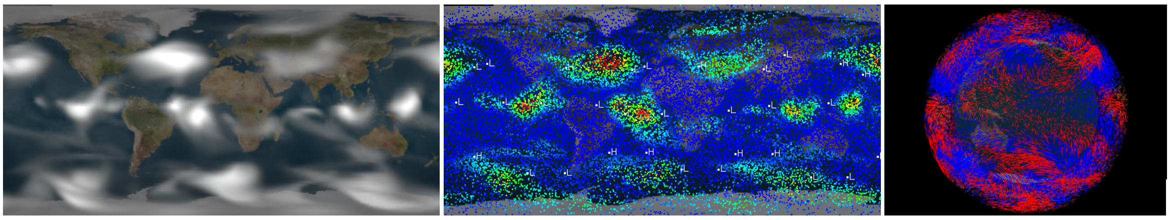
Fig. 3 shows a snapshot of our interactive system. As described previously, our method requires four maps as input data (see Fig. 1). In the following subsections, methods for creating the four maps are described first. Then, the visualization method for simulation is explained.

5.1 Creation of Input Maps

Among the four maps, the pressure map is the most important factor in our method since this directly affects the atmospheric fluid flows. It is impractical to force the user to specify the pressure at every point on the earth. To address this, our system provides the user with *a pressure primitive*. The pressure primitive represents a high/low atmospheric pressure area. A center position, $\mathbf{c}$, a center pressure, q, and an effective distance, D, are assigned to the pressure primitive as shown in Fig. 4. Using these parameters, a field function, f, is defined for each pressure primitive. We use an implicit function, $f(d, D)$, proposed by [15]. This function is unity at $d = 0$ and gradually decreases to zero at $d = D$ (see Fig. 4). The user specifies a set of pressure primitives by using our system. The system computes the pressure at point, $\mathbf{x}$, by a weighted average as:

$$p_0(\mathbf{x}) = \sum_{k=1}^{n} q_k f(|\mathbf{c}_k - \mathbf{x}|, D_k),$$

where n is the number of the pressure primitives, q_k and D_k are the center pressure and the effective distance of the kth pressure primitive, respectively. The time variation of the pressure map is specified using the key-frame method. That is, the user specifies parameters of each of the pressure primitives at several key frames. These parameters are linearly interpolated to create the pressure map at intermediate frames. During the interaction,



(a) Volume rendering (cloud density). (b) Particles (cloud density). (c) Line segments (velocity).

Figure 5: Real-time visualization of simulation.

the pressure map is displayed to the user as an atmospheric pressure pattern such as often seen in weather forecasting programs on TV, as shown in Fig. 3.

Next, the temperature map is created as follows. The temperature of the earth's surface is highest around the equator. This is because the earth's surface around the equator faces perpendicularly to the sun's direction. We assume that the shape of the earth is a sphere. Then the temperature map is created by computing the energy from the sun reaching each point on the sphere. This is similar to shading the sphere illuminated by the sun. The user specifies the energy radiated from the sun, e_{sun}.

The friction and the amount of evaporation depend on the types of the earth's surface such as savanna, desert, ocean, etc. A simple solution is to classify the earth's surface into only two types, the ocean and the land. In this paper, we employ this simple method to create the friction map and the evaporation map. To create the friction map, the user specifies only the two frictional coefficients, c_{land} and c_{ocean}, for the ocean and the land, respectively. Similarly, the evaporation map is created by specifying the evaporation rates, w_{land} and w_{ocean}, for the ocean and the land, respectively. It is also possible to create these maps by taking into account various types of the earth's surface. For this purpose, our system accepts a bit map image that represents the distribution of the frictional coefficients and the evaporation rates.

5.2 Visualization

Our system visualizes the simulation in real-time with simple and fast rendering methods as shown in Fig. 5. Our system provides two types of display mode. In the first mode, the simulation is visualized by mapping the result onto a sphere representing the earth (see Fig. 3). In the second mode, the simulation is displayed using the Mercator projection (Figs. 5(a) and 5(b)). In both modes, real-time visualization is achieved by using either a simple volume rendering technique or particles. Fig. 5(a) shows the visualization of cloud density using the volume rendering technique. In this case, the colors from black to white are assigned according to the density of clouds. Fig. 5(b) shows the visualization of cloud density by using 30,000 particles. Colors from blue to red are assigned to each particle according the density of clouds. The velocity distribution is visualized by drawing line segments as shown in Fig. 5(c). The red lines indicate that the direction of the velocity is upward (i.e., ascending air current) and the blue lines indicate that the direction is downward (descending air current).

After the simulation has completed, photorealistic images are generated offline. Our simulator generates a coarse distribution of clouds. The detailed shape of small-scale clouds is generated by using an advection texture technique [16]. The colors of clouds are calculated by taking into account the scatterings and the absorptions of light due to both the clouds and the atmosphere. We employ the method proposed in [8] to create final images.

6. Results

Figs. 6 and 7 show examples of the proposed method. In Fig.6, the basic atmospheric flow is generated. Fig. 6(a) shows the user-specified pressure map. In reality, many low atmospheric pressure areas appear randomly around the equator, especially, over the Pacific Ocean. To simulate this, the pressure primitives with low center pressures are generated randomly. In addition, around mid latitude, high pressure and low pressure areas appear alternatively. These atmospheric pressure areas move westwards. To simulate this, pressure primitives with high and low center pressures are placed alternatively (see Fig. 6(a)). They move to the west at a user-specified speed. Using this pressure map, the cloud formation process is simulated. Fig. 6(b) shows a density distribution of clouds obtained by the simulation. Around the equator, clouds are generated due to strong ascending air currents. Since the Coriolis force is nearly zero around the equator, the distribution of clouds is comparatively simple. On the other hand, at the mid latitudes, complex distributions are generated due to the Coriolis force. Fig. 6(c) shows the final image. Fig. 6(d) shows an image from a different viewpoint at different time.

Next, we simulated collision of two huge typhoons as shown in Fig. 7. Figs. 7(a) and (b) show the pressure maps before and after the collision. Figs. 7(c) through (f) show the resulting images. The viewpoint approaches to the point of the collision. As shown in this example, we can simulate the phenomena that could hardly ever occur in the real world.

The computation time for the simulation took about 0.1 seconds for each time step on a desktop PC with a Pentium IV 3.6GHz. The number of grids was 160 x 80 x 4. The rendering time for the final images (Figs. 6(c), 6(d), 7(c), 7(d), 7(e), and 7(f)) ranged from 1 second to 10 seconds.

7. Conclusion

In this paper, we have proposed the method for animating earth-scale clouds. We have developed a computational model for simulating atmospheric fluid flow. In our method, the user specifies the atmospheric pressures on the earth's surface. This enables the user to control the motion of clouds. Using this data, the method generates the detailed shapes and the motions of clouds surrounding the earth. We have also proposed the interactive visual simulation system based on the proposed method. The user can adjust the control parameters interactively and the modification to the parameters is immediately reflected in the simulation. The progress of the cloud formation process is visualized in real-time. The user can create desired shapes and motion of clouds by using this system. By using the proposed method, the realistic animation of the earth-scale clouds has been achieved.

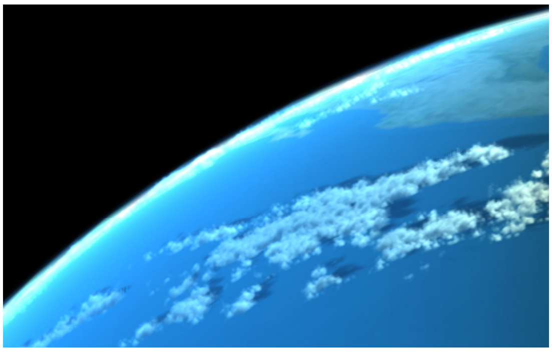


Figure 8: Combination of the proposed method and a method for small-scale clouds.

In our current model, land altitude is not taken into account. Therefore, the proposed method cannot generate clouds due to mountains. So one of the future works is to reflect the land altitude on the simulation. A further speedup of the simulation and the rendering is indispensable in order to use our method in real-time applications such as games. Currently, our method is not fast very much since most of the computations are performed by software. This can be improved by making the most of the graphics hardware. Another future research will address the incorporation of our method with earlier methods for local cloud formation. That is, by combining our method with the previous methods, a seamless simulation of natural phenomena from the earth-scale clouds to the small-scale clouds might become possible. For example, in Fig. 8, cellular automata [7] and the proposed method are combined. The cellular automata are used to generate small-scale clouds based on the global distribution of clouds calculated by the proposed method.

References

[1] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley, Texturing & Modeling: A Procedural Approach 3rd Edition, The Morgan Kaufmann Series (2003).

[2] J. T. Kajiya and B. P. V. Herzen, Ray Tracing Volume Densities, *Computer Graphics (Proc. SIGGRAPH'84)*, *18*, 3, 165-174 (1984).

[3] R. Miyazaki, Y. Dobashi, and T. Nishita, Simulation of Cumuliform Clouds based

on Computational Fluid Dynamics, *Proc. EUROGRAPHICS 2002 Short Presentations*, 405-410 (2002).

[4] M. J. Harris, W. V. Baxter, T. Scheuemann, and A. Lastra, Simulation of Cloud Dynamics on Graphics Hardware, *Proc. Graphics Hardware 2003*, Annual Conference Series, 92-101 (2003).

[5] Overby, D., Melek, Z., and Keyser, J. (2002). Interactivephysically-based Cloud Simulation, *Proc. Pacific Graphics 2002*, 469 (2002).

[6] R. Miyazaki, S. Yoshida, Y. Dobashi, and T. Nishita. A Method for Modeling Clouds Based on Atmospheric Fluid Dynamics, *Proc. Pacific Graphics 2001*, 363-372 (2001).

[7] Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, T. Nishita. A Simple, Efficient Method for Realistic Animation of Clouds, *Proc. SIGGRAPH'2000*, 19-28 (2000).

[8] T. Nishita, T. Sirai, K. Tadamura, and E. Nakamae, Display of The Earth Taking into Account Atmospheric Scattering, *Proc. SIGGRAPH'93,* Annual Conference Series, 175-182 (1993).

[9] L. Yaeger and C. Upson, Combining Physical and Visual Simulation, Creation of the Planet Jupiter for the Film 2010, *Computer Graphics (Proc. SIGGRAPH'86), 20*, 4, 85-93 (1986).

[10] C. D. Rodgers, Retrieval of Atmospheric Temperature and Composition from Remote Measurements of Thermal Radiation, *Review of Geophysics and Space Physics*, *14*, 609-624 (1976).

[11] W. Hibbard and D. Santek, Visualization Large Data Sets in the Earth Science, *IEEE Computer*, *22*, 8, 53-57 (1989).

[12] Y. Dobashi, T. Nishita, H. Yamashita, and T. Okita, Using Metaballs to Modeling and Animate Clouds from Satellite Images, *The Visual Computer*, *15*, 9, 471-482 (1998).

[13] J. Stam, "Stable Fluids," *Proc. SIGGRAPH'99*, 121-128 (1999).

[14] T. J. Chung, Computational Fluid Dynamics, 95-97, Cambridge University Press (2002).

[15] G. Wyvill and A. Trotman, Ray-Tracing Soft Objects," *Proc. Computer Graphics International 1990*, 439-475 (1990).

[16] N. Max, R. Crawfis, and D. Williams, Visualizing Wind Velocities by Advecting Cloud Textures, *Proc. IEEE Visualization 1992*, 179-183 (1992).

[17] E. X. Berry, Modification of the Warm Rain Process, Preprints 1st Not. Conf. Weather Modification, Albany, Amer. Meteor. Soc., 81-88 (1968).

Appendix: Our Simulation Model of Earth's Atmosphere

Our simulation method is based on [3]. We extend the method to including the important factors for the earth-scale flow. First, the velocity distribution, **u**, of the atmosphere is obtained from the numerical analysis of the following incompressive NS equations.

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \upsilon \nabla^2 \mathbf{u} + \mathbf{f} \,,$$

$$\nabla \cdot \mathbf{u} = \mathbf{0} \,,$$

where ρ is the density of the atmosphere, p is the pressure, υ is the viscosity, **f** is an external force. Three types of external forces are considered, namely, the pressure gradient force on the earth's surface, $\mathbf{f}_p$, the frictional force, $\mathbf{f}_{fric}$, between the atmosphere and the earth's surface, and the Coriolis force, $\mathbf{f}_{cori}$. These are expressed by the following equations.

$$\mathbf{f}_p = -\frac{1}{\rho}\nabla p_0 \,,$$

$$\mathbf{f}_{fric} = -\mu_0 \overleftrightarrow{\mathbf{u}}_0 \,,$$

$$\mathbf{f}_{cori} = -2\Omega \mathbf{r} \times \mathbf{u} \,,$$

where p_0 is the atmospheric pressure on the earth's surface provided as the pressure map, μ_0 is the frictional coefficient provided as the friction map, $\overleftrightarrow{\mathbf{u}}_0$ is the horizontal component of the velocity of the atmosphere at the earth's surface, Ω is the rotation speed of the earth, **r** is the rotation axis.

Next, the temperature of the atmosphere is calculated from the following equation.

$$E = e_{sun} E_0 - \Gamma h \,,$$

where e_{sun} is the energy of the sun reaching the earth, E_0 is the surface temperature provided as the temperature map, Γ is the rate of temperature drop, and h is a height from the earth's surface.

The densities of water vapor w_{vapor}, cloud w_{cloud}, and raindrops w_{rain} are calculated by the following equations, respectively.

$$\frac{\partial w_{vapor}}{\partial t} = -(\mathbf{u} \cdot \nabla) w_{vapor} + w_0 E_0 - P_{vc} + P_{rv},$$

$$\frac{\partial w_{cloud}}{\partial t} = -(\mathbf{u} \cdot \nabla) w_{cloud} + P_{vc} - P_{cr},$$

$$\frac{\partial w_{rain}}{\partial t} = -(\mathbf{u} \cdot \nabla) w_{rain} - V_T \mathbf{g} w_{rain} + P_{cr} - P_{rv},$$

where w_0 is the amount of evaporation from the earth's surface provided as the evaporation map, P_{vc} is the amount of clouds generated by the phase transition from the vapor to water droplets, P_{rv} is the amount of evaporation from the raindrops, P_{cr} is the amount of the raindrops generated by the collision of the cloud particles, V_T is the terminal velocity of the raindrops, **g** is the gravitational vector. The above three equations simulate the circulation of water. The operator $-(\mathbf{u} \cdot \nabla)$ in these equations represents the advection due to the atmospheric flow. The transitions of water between vapor, clouds, and raindrops are represented by P_{vc}, P_{cr}, P_{rv}, respectively. The evaporation from clouds to water vapor are not taken into account in the above equations since including this factor had little effects on the resulting cloud patterns in our experiment. The phase transition from water vapor to clouds, P_{vc}, is formulated as:

$$P_{vc} = \alpha (w_{vapor} - w_{\max}),$$

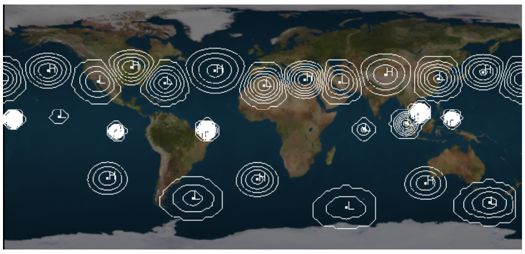
$$w_{\max} = \max\left(A \exp(-B / E), \quad w_{vapor} + w_{cloud}\right),$$

where α is a phase transition rate, w_{max} is the saturation vapor content, A and B are the parameters which determine the shape of the saturation vapor function. Note that we use an absolute temperature to ensure that E is always positive. Since our interest is in the formation of clouds, we use extremely simplified equations for P_{cr} and P_{rv}:

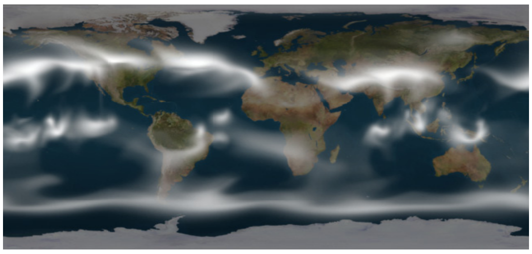
$$P_{cr} = \beta w_{cloud},$$

$$P_{rv} = \gamma w_{rain},$$

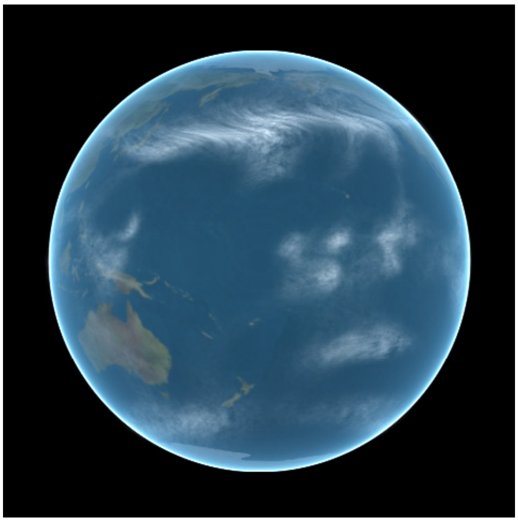
where β and γ are a raindrop formation rate and an evaporation rate from the raindrop, respectively. More precise models can be found in [17].



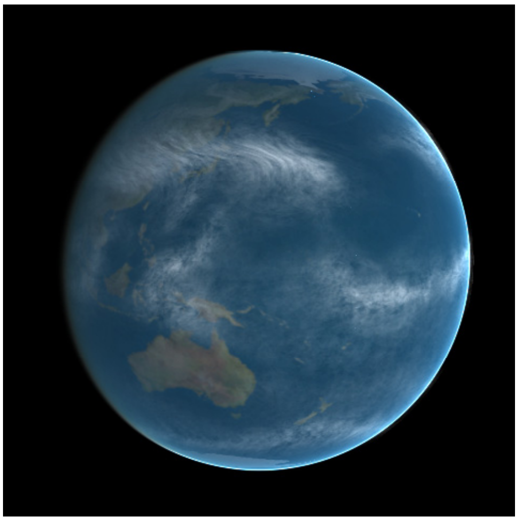
(a) A pressure map.



(b) Density distribution of clouds.

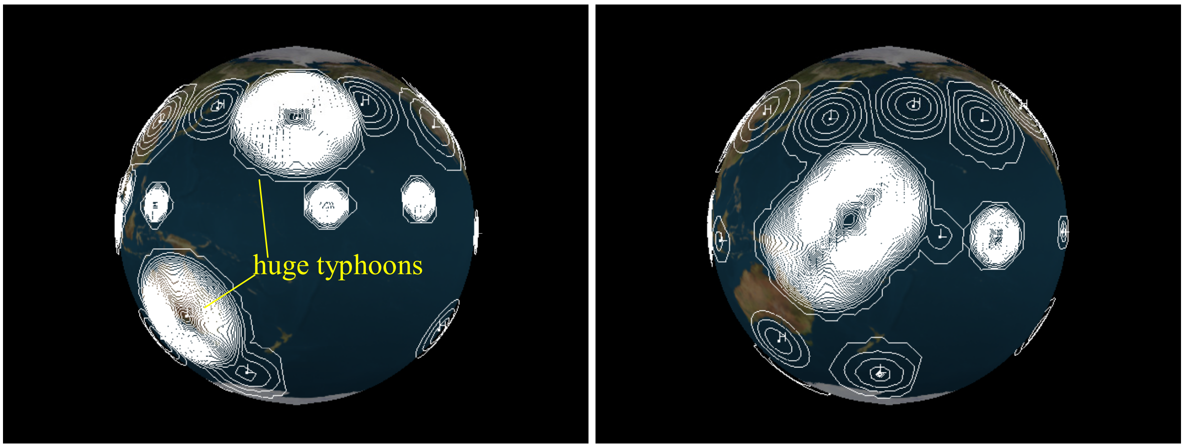


(c) Final image.

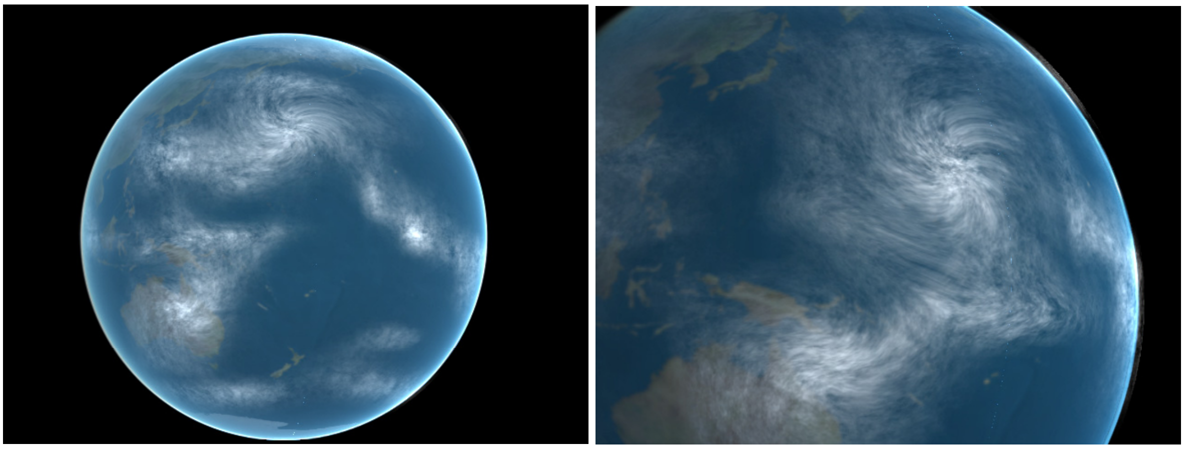


(d) Image from a different viewpoint.

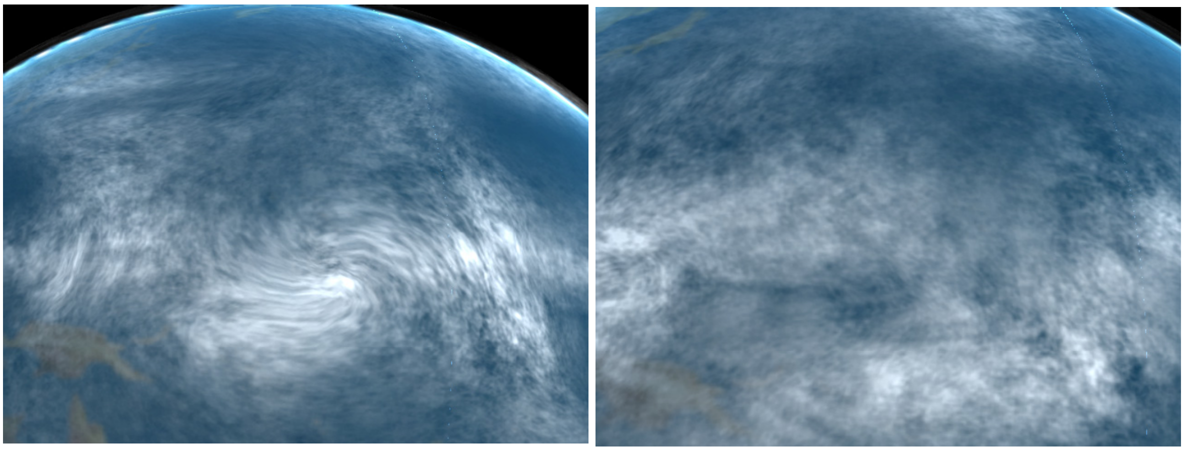
Figure 6: Simulation of atmospheric clouds of the earth.



(a) Pressure map before two typhoons collide. (b) Pressure map after two typhoons collide.



(c) Two typhoons are approaching. (d) Image just before typhoons collide.



(e) Image just after the collision. (f) Typhoons are disappearing.

Figure 7: Simulation of collision of two huge typhoons.