Feedback Control of Cumuliform Cloud Formation based on Computational Fluid Dynamics

Yoshinori Dobashi[*]
Hokkaido University

Katsutoshi Kusumoto[†]
Hokkaido University

Tomoyuki Nishita[‡]
The University of Tokyo

Tsuyoshi Yamamoto[§]
Hokkaido University

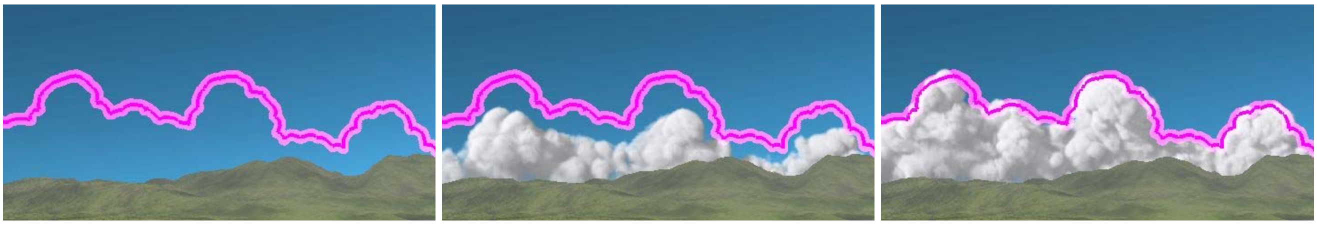


Figure 1: *Clouds generated by our method. The pink curve indicates the desired shape specified by the user. Our method controls the cloud formation process in order to make clouds form the desired shape.*

Abstract

Clouds play an important role for creating realistic images of outdoor scenes. In order to generate realistic clouds, many methods have been developed for modeling and animating clouds. One of the most effective approaches for synthesizing realistic clouds is to simulate cloud formation processes based on the atmospheric fluid dynamics. Although this approach can create realistic clouds, the resulting shapes and motion depend on many simulation parameters and the initial status. Therefore, it is very difficult to adjust those parameters so that the clouds form the desired shapes. This paper addresses this problem and presents a method for controlling the simulation of cloud formation. In this paper, we focus on controlling cumuliform cloud formation. The user specifies the overall shape of the clouds. Then, our method automatically adjusts parameters during the simulation in order to generate clouds forming the specified shape. Our method can generate realistic clouds while their shapes closely match to the desired shape.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.3.6 [Computer Graphics]: Methodology and Techniques;

Keywords: simulation control, clouds, fluid dynamics

1 Introduction

Simulation of natural phenomena has become an important research field in computer graphics. Recently, many methods have been proposed for simulating phenomena related to fluids such as clouds, smoke, fire, and water. Many methods use computational fluid dynamics (CFD) for creating realistic animations [Stam 1999; Foster and Fedkiw 2001; Nguyen et al. 2002; Feldman et al. 2003]. These methods are widely used in applications such as movies and commercial films. In these applications, however, it is often required to create desired shapes and the motion of the phenomena. Although CFD methods can generate realistic shapes and motion, it is difficult for the user to control the simulation result. To address this problem, several methods have been proposed for controlling the numerical simulation process [Treuille et al. 2003; Fattal and Lischinski 2004; McNamara et al. 2004]. In previous methods, the motion of smoke and water to form user-specified shapes such as letters and animals has been calculated. In this paper, we focus on controlling cloud formation.

Clouds are important elements in creating synthetic images of outdoor scenes. CFD is effective for synthesizing realistic clouds [Kajiya and Herzen 1984; Miyazaki et al. 2002; Harris et al. 2003]. To apply CFD techniques to clouds, we have to take into account atmospheric fluid dynamics where many physical parameters influence the cloud formation process. Although methods using CFD can generate realistic clouds, it is almost impossible to adjust the physical parameters manually so that the clouds form the desired shapes.

In this paper, we propose a method for controlling the simulation of the cloud formation process. Although there are many types of cloud, we focus on cumuliform clouds. Cumuliform clouds often form characteristic shapes even in the real world. Our purpose is slightly different from the previous fluid control methods, in which the fluid is controlled so that the smoke or water forms unnatural shapes, such as letters, or human shapes, or animals, etc. In contrast, our method generates realistic clouds that are often observed in actuality, but with their shape closely matched to a user-specified shape. In our method, the user specifies the contours of the clouds viewed from a specific camera position. The simulation process is controlled in order to form the desired shape of the clouds. One straight forward approach to achieve this goal is to apply the previous methods for controlling smoke or water to clouds. However we found that this approach did not produce convincing results. The reason is that there are several physical processes, such as the phase transition from water vapor to water droplets (i.e. clouds), that are not present in other phenomena. We therefore developed a new method that controls the physical parameters affecting the cloud formation process. This results in natural cloud shapes and motion.

[*] e-mail:doba@nis-ei.eng.hokudai.ac.jp
[†] e-mail:kusumoto@nis-ei.eng.hokudai.ac.jp
[‡] e-mail:nis@is.s.u-tokyo.ac.jp
[§] e-mail:yamamoto@ist.hokudai.ac.jp

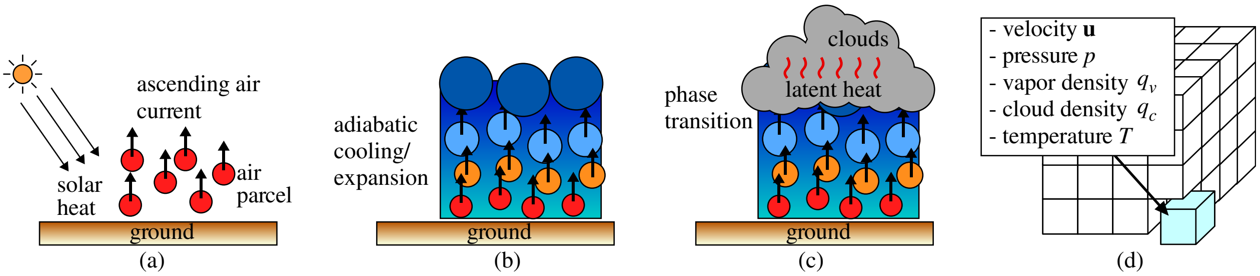


Figure 2: *Overview of the cloud formation process. Rising air parcels are generated due to the heat from the ground (a). The temperature of the parcels decreases due to adiabatic expansion (b). Then, clouds are generated by the phase transition from water vapor to water droplets (c). These processes are simulated by using a three-dimensional grid (d).*

Fig. 1 shows an example of clouds generated by our method. The pink curve indicates the desired shape specified by the user. Clouds grow gradually and finally form the desired shape as shown in Fig. 1.

2 Related Work

First, we discuss methods for controlling fluid simulation, and next the method for simulating cloud formation is also reviewed.

Fluid Control: After the pioneering work by Treuille et al. [2003] for keyframe control of smoke simulation, many control methods have been proposed for the simulation of phenomena related to fluids. McNamara et al. [2004] presented a method for controlling smoke and water by using the adjoint method. Fattal and Lischinski [2004] controlled smoke by adding external forces which they referred to as driving force and gathering terms. Hong and Kim [2004] developed a method for calculating the natural motion of smoke to form a desired shape. This method uses a potential field created from the desired shape for computing the external forces needed to move the smoke. Shi and Yu [2005] proposed a feedback control mechanism for creating the animation of water. Kim et al. [2007] also employed a feedback control mechanism in order to compensate the volume loss or gain for the simulation of bubbles. The basic concept of these methods is similar to ours, however, these are not directly applicable to clouds since they do not take into account the physical phenomena involved in cloud formation. More recently, Thürey et al. [2006] controlled the motion of water by using control particles. Kim et al. [2006] proposed a method to make smoke move along a user-specified path. By using these methods, the motion of smoke and water is controlled so that they form user specified shapes and motion. However, there are no methods for controlling the formation of clouds based on computational fluid dynamics.

Cloud Simulation: There are many methods for creating synthetic clouds. They are roughly classified into two categories. One uses procedural techniques and the other is based on the physical simulation of the cloud formation process.

Methods using procedural techniques generate the density distribution of clouds, using the idea of fractals. Methods in this category are well described in the book by Ebert et al [2002]. Methods for modeling clouds without generating three-dimensional density distributions were also proposed [Trembilski and Broßler 2002; Bouthors and Neyret 2002; Neyret 1997; Gardner 1985]. Although the computational cost for the procedural technique is very low and it is possible to generate the desired cloud shapes, a trial and error process is often required to generate realistic clouds. Creating realistic animations of clouds is also a difficult task.

On the other hand, methods based on physical simulation can generate clouds by numerical simulation [Dobashi et al. 2000; Kajiya and Herzen 1984; Miyazaki et al. 2001; Miyazaki et al. 2002; Harris et al. 2003]. Although these methods have the potential to generate realistic clouds, many physical parameters have to be adjusted to generate convincing results. This further indicates that adjusting the parameters manually to create clouds with desired shapes is almost impossible.

3 Simulation of Cumuliform Cloud Formation

Before describing our method, this section explains the numerical simulation of cumuliform cloud formation. The cumuliform cloud formation processes are summarized in Fig. 2. First, the air is heated by the ground and this creates ascending air currents (Fig. 2(a)). The temperature of the rising air currents decreases due to adiabatic cooling (Fig. 2(b)), so vapor in the air parcel causes a phase transition, condenses , and the cloud is generated (Fig. 2(c)). At that time, the latent heat is liberated, which creates additional buoyancy forces and promotes further growth of the clouds. As we will describe later, our method mainly controls the amount of latent heat. For the numerical simulation, we use the method proposed by Miyazaki et al [2002].

The simulation space is subdivided into $N_x \times N_y \times N_z$ voxels as shown in Fig. 2(d). At each voxel, the velocity $\mathbf{u} = (u, v, w)$, the pressure p, the vapor density q_v, the cloud (water droplets) density q_c, and the temperature T are assigned. These variables are updated at every time step in the simulation by solving the following equations numerically

3.1 Governing Equations

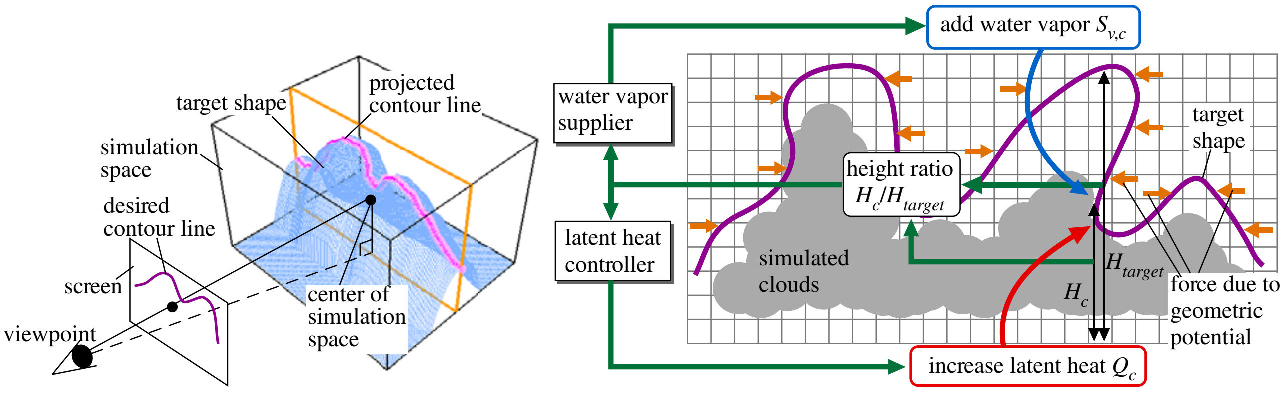
The density of air is assumed to be constant, and the atmosphere is assumed to be an incompressible and inviscid fluid media. Then the motion of the atmosphere is expressed by the following Navier-Stokes equations.

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} &= -(\mathbf{u}\cdot\nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \mathbf{B} + \mathbf{f}, && (1) \\ \nabla\cdot\mathbf{u} &= 0, && (2) \end{aligned}$$

where ρ is the density, $\mathbf{B}$ is the buoyancy force, and $\mathbf{f}$ is an external forces such as wind. The buoyancy force $\mathbf{B}$ is expressed by the following equation.

$$\mathbf{B} = k_b \frac{T - T_{amb}}{T_{amb}}\mathbf{z} - g q_c \mathbf{z}, \quad (3)$$

where T is the temperature, T_{amb} is the ambient temperature, $\mathbf{z} = (0, 0, 1)$ which points in the upward vertical direction, k_b is the buoyancy coefficient and g is the coefficient for the gravitational force. T_{amb} is a function of height and we assume that it decreases in proportion to the height from the ground. The first term on the



(a) (b)

Figure 3: *Overview of our system. (a) shows the input to our system. The user specifies the desired shape of the clouds by drawing a contour line on the screen. The three-dimensional target shape is generated based on the specified contour line. The simulation of cloud formation is controlled so that the shapes of the clouds become the target shape. (b) shows our control mechanism. Our method measures the height ratio between the simulated clouds and the target shape. The ratio is fed back into the system to determine the additional water vapor and the latent heat coefficient at each time step. The geometric potential field is also used to guide the cloud formation.*

right hand side of Eq. (3) indicates that the buoyancy force is proportional to the difference between the temperatures of the rising air parcel and the surrounding air. The second term indicates the drag force due to the falling water droplets (i.e. clouds).

Next, the phase transition between the water vapor and the clouds is represented by the following equations.

$$\frac{\partial q_c}{\partial t} = -(\mathbf{u}\cdot\nabla)q_c + C_c, \tag{4}$$

$$\frac{\partial q_v}{\partial t} = -(\mathbf{u}\cdot\nabla)q_v + C_c + S_v, \tag{5}$$

$$C_c = \alpha(q_v - q_s) \tag{6}$$

where C_c is the amount of clouds generated by the phase transition, S_v is the amount of water vapor supplied from vapor sources such as a lake, and α is the phase transition ratio. When C_c is less than zero, the amount of clouds q_c is reduced as shown in Eq. (4). This indicates the evaporation of water droplets (i.e. clouds). q_s is the saturation vapor content and the following function is used.

$$q_s = \max(A\exp(-B/(T+C)), q_v + q_c). \tag{7}$$

Finally, the temperature is expressed by the following equation.

$$\frac{\partial T}{\partial t} = -(\mathbf{u}\cdot\nabla)T - \Gamma_d w + QC_c + S_T, \tag{8}$$

where Γ_d is the adiabatic lapse rate, w is the z component of the fluid velocity, and Q is the coefficient of the latent heat. S_T is the heat supplied from the ground. The first term on the right in Eq. (8) indicates the advection by the flow field, the second term indicates the adiabatic cooling of rising air, and the third term indicates the latent heat. The amount of latent heat is assumed to be proportional to the amount of cloud generated by the phase transition as shown by the third term. We used the numbers shown in Table 1 (see Section 6) for the parameters appearing in the equations in this section.

3.2 Initial Status and Boundary Conditions

The initial status of the simulation is specified as follows. The initial temperature T_0 is set to the ambient temperature T_{amb}. The initial water vapor $q_{v,0}$ is set to decrease exponentially from the bottom of the simulation space. The initial temperature and water vapor are constant in the horizontal direction. The initial cloud density $q_{c,0}$ is zero. The initial velocity $\mathbf{u}_0$ is also set to $\mathbf{0}$. Next, for the boundary conditions, a periodic boundary condition is used in the horizontal direction and a fixed boundary condition ($\mathbf{u} = \mathbf{0}$) is used on the bottom and top of the simulation space.

4 Our Control Method for Cloud Formation

Fig. 3 shows the basic idea of our method. As the input to our system, the user specifies a contour line of the desired shape of the clouds as viewed from a desired camera position (see Fig.3(a)). The contour line is drawn on the screen. In Fig. 3(a), the pink curve indicates the desired shape. The contour line is projected onto a plane perpendicular to the xy component of the vector connecting the viewpoint and the center of the simulation space. Then a three-dimensional shape is generated from the contour line. We call this shape the *target shape*. The target shape is generated so that its contour line becomes the same as the one specified by the user.

Our method controls the simulation so that the difference between the target shape and the simulated clouds becomes zero. The effect of wind is not taken into account since it does not contribute very much to the cumuliform cloud formation. The convection due to the buoyancy force is the main driving force for the cumuliform clouds. Our control mechanism is shown in Fig. 3(b). To measure the difference, we use the height ratio R of the top of the simulated clouds to the top of the target shape. The ratio is computed for each column consisting of the grid points above grid point $(i, j, 0)$ at the bottom of the simulation space. The system controls the simulation so that the ratio $R(i, j)$ becomes one for all columns in the simulation space. R is expressed by:

$$R(i,j) = H_c(i,j)/H_{target}(i,j), \tag{9}$$
$$(i = 1,\cdots,N_x; j = 1,\cdots,N_y)$$

where $H_c(i, j)$ is the height of the top of the simulated clouds at each time step and $H_{target}(i, j)$ is the height of the top of the target shape (see Fig. 3(b)). H_c and H_{target} are measured from the bottom of the simulation space. The height of the highest point of the target shape is used for H_{target} in order to make the simulated clouds reach the highest point. This approach is best suited for a desired shape that can be represented by a height field. However, the clouds in the real life are not always represented by a height field. This problem is solved in the following way.

There are two key features in our control method: feedback control and a geometric potential field. The feedback controller promotes the cloud growth until the clouds reach the top of the target shape, that is, $R(i, j) = 1.0$. When the target shape is not a height field, the clouds may grow outside of the target shape. The geometric potential field [Hong and Kim 2004] is used to address this problem. We generate external forces that prevent the clouds from growing outside the target shape by using the geometric potential field. However, unlike the previous method, we use only the horizontal components of the external forces as shown by orange arrows in Fig. 3(b). This is because the height of the clouds is controlled by the feedback controller. In other words, the feedback controller adjusts the vertical extent and the geometric potential field adjusts the horizontal extent of the clouds.

In the feedback control, cloud growth is promoted by controlling the latent heat and by supplying additional water vapor. To determine the amounts of latent heat and additional water vapor, we developed two functions; a *latent heat controller* and a *water vapor supplier*. The height ratio $R(i, j)$ is fed back into these functions as shown in Fig. 3(b). Then the latent heat controller controls the coefficient for the latent heat Q (see Eq. (8)) at each time step to adjust the amount of latent heat. The water vapor supplier adds water vapor at each time step in the regions where the clouds have not reached the top of the target shape. We chose the latent heat and the water vapor as control variables based on our experimental investigation. We found that these parameters significantly influence cloud growth and are suitable for control variables. Other parameters remain fixed throughout the simulation.

In the original simulation method described in Section 3, Q is constant and the water vapor is supplied only at the bottom of the simulation space. However, these are not sufficient to control cloud formation. For the latent heat, we assign the coefficient $Q_c(i, j)$ for the grid point $(i, j, 0)$ as a control variable. The coefficient for the latent heat for grid point (i, j, k) is the same as $Q_c(i, j)$. The latent heat controller updates $Q_c(i, j)$ at each time step. Next, our method supplies the water vapor at the top of the simulated clouds in addition to the bottom of the simulation space. The water vapor supplier determines the amount of water vapor, $S_{v,c}(i, j)$, to be supplied at each time step. Consequently, there are $N_x \times N_y \times 2$ control variables in our method. To adjust these parameters, we employ a method known as proportional-integral-derivative (PID) controller [Haugen 2004].

Next, we describe about the geometric potential field. The geometric potential field is generated in a preprocessing step so that the potential value becomes large inside the target shape. The external forces are generated in proportion to the gradient of the potential field. As a result, the external forces prevent clouds from growing outside the target shape.

We now explain how our control mechanism works (see Fig. 3(b)). At each time step, we first compute the height ratio $R(i, j)$ expressed by Eq. (9). The ratio is sent to the latent heat controller and the water vapor supplier. Then the water vapor supplier determines the amount of the water vapor, $S_{v,c}$, and adds this at the grid points corresponding to the top of the simulated clouds. This process promotes the phase transition from water vapor to clouds. Next, the latent heat controller increases Q_c where the clouds have not reached the top of the target shape (i.e., $R(i, j) < 1.0$). As shown by the third term on the right hand side of Eq. (8), this increases the temperature when the clouds are generated due to the phase transition. The increase in temperature then results in an increase in the buoyancy force as shown in Eq. (3). This promotes the cloud growth to higher regions. During cloud growth, the external forces due to the geometric potential field push the clouds inwards the inside the target shape. As the clouds approach the top of the target shape, the water vapor supplier decreases the amount of additional water vapor. The latent heat controller also stops increasing the latent heat coefficient. In this way, clouds whose shape closely matches the target shape are generated automatically.

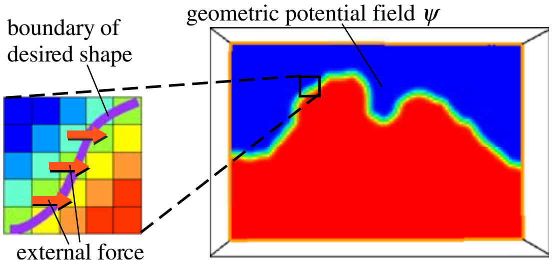


Figure 4: *Example of a geometric potential field. A cross section of the field is shown on the right. The red color corresponds to 1.0 and blue to 0.0. The external force is generated in proportion to the horizontal component of the gradient of the field as shown on the left.*

In our current system, the simulation is terminated manually by the user. Since the system displays the simulated clouds at each time step, the user can see if the clouds have formed the target shape. Although the clouds stay at the target shape even after they have reached the shape, the details might be reduced because the control forces could prevent vortices and detail from developing.

5 Controlling Cloud Simulation

This section describes the generation of the target shape, the geometric potential field, and the feedback control of the simulation.

5.1 Generation of Target Shape

The target shape is generated from the user specified contours of the desired cloud shape. First, the user specifies the region where clouds are to be generated. The center of the simulation space is placed at the center of the specified region. Next, the user draws the contours on the screen using a mouse. Then the contours are projected onto a plane facing the viewpoint and including the center of the simulation space. Then the target shape is generated by using the method proposed by Igarashi et al [1999]. This method thickens the two-dimensional sketch by extracting its medial axis. Since the cumulus clouds usually form tower-like shapes, the method is suitable for our purpose. Next a bounding box of the target shape is generated and is subdivided into a three-dimensional grid. The grid is used to compute the geometric potential field and to simulate the cloud formation process.

5.2 Geometric Potential Field

First, an initial potential field is generated by assigning 0 or 1 depending on whether each grid point is inside the target shape or not. We assign 1 to the inside grid points and 0 to the outside grid points. Next, we apply a three-dimensional Gaussian filter to the initial field in order to create a smooth and continuous potential field. The resulting field is the geometric potential field. Fig. 4 shows an example of the geometric potential field. During simulation, the external force due to the geometric potential field, F, is calculated at each grid point. Since our method uses only the horizontal component of F as shown in Fig. 4, F is expressed by the following equation.

$$F(q_c, \psi) = q_c \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \nabla\psi, \qquad (10)$$

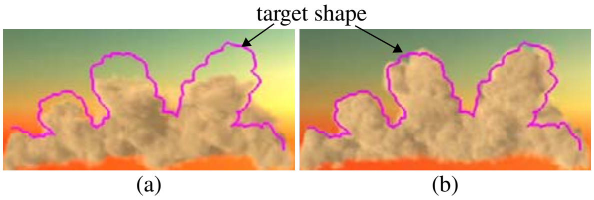


Figure 5: *Comparison of different control. (a) uses proportional (P) control only while (b) uses proportional-integral (PI) control. PI controller can fill the gaps between the clouds and the target shape.*

where q_c is the density of clouds and ψ indicates the potential field. The external force is updated at each time step of the simulation by using Eq. (10). Since the external force is proportional to the gradient of the potential field, it only works near the boundary of the target shape. In addition, no external forces are generated at the grid points where no clouds exist since F is also proportional to the cloud density q_c.

5.3 Latent Heat Controller

For the latent heat controller, we use the PID control mechanism. However, for our purpose, we found that proportional-integral (PI) control is sufficient. In this case, the latent heat controller updates the coefficient for the latent heat, $Q_c(i,j)$ at each time step, according to the following equation.

$$\begin{aligned} Q_c(i,j) &= K_P(i,j)\Delta_H(i,j) \\ &+ K_I(i,j)\int_0^D \Delta_H(i,j,\tau)d\tau, \qquad (11) \\ \Delta_H(i,j) &= 1.0 - R(i,j) \\ &= \frac{H_{target}(i,j) - H_c(i,j)}{H_{target}(i,j)}, \qquad (12) \end{aligned}$$

where $\Delta_H(i,j)$ is the normalized height difference between the top of the clouds and the target shape at each time step. Let us now explain the meaning of Eq. (11). The first term on the right updates Q_c in proportion to the difference Δ_H. This term is called the proportional controller and K_P is called the proportional gain. The second term updates Q_c in proportion to the integral of the difference Δ_H and is called an integral controller. The second term contributes when the accumulated difference becomes large. D is the duration for the accumulation and K_I is called the integral gain. When clouds have grown into the region near the top of the target shape, the difference Δ_H becomes very small. In this case, the first term cannot contribute to the updating of Q_c. As a result, when using the proportional controller only, small gaps between the simulated clouds and the target shape are left. The second term removes the gaps and updates Q_c until Δ_H becomes zero. Fig. 5 shows the effect of these controllers. Fig. 5(a) shows the result using only the proportional controller. The clouds cannot reach the target shape. By using the integral controller, clouds that match the target shape are obtained as shown in Fig. 5(b).

For the latent heat controller expressed by Eq. 11, the control parameters, K_P and K_I need to be specified. We determine these parameters experimentally. However, it is difficult to determine all these parameters since K_P and K_I are defined for $i = 1, \cdots, N_x$ and $j = 1, \cdots, N_y$. From our experiments, we found that larger values are required for K_P and K_I at regions where the top of the target shape is high. This is because the cloud growth in such regions has to be promoted further than the regions where the top of the target shape is low. Therefore, we assume that K_P and K_I are proportional to the height of the target shape. That is,

$$\begin{cases} K_P(i,j) = \kappa_P \bar{H}_{target}(i,j) \\ K_I(i,j) = \kappa_I \bar{H}_{target}(i,j) \end{cases}, \qquad (13)$$

where κ_P and κ_I are the proportionality coefficients. $\bar{H}_{target}$ is the height of the target shape divided by the height of the simulation space. As a result, the two parameters, κ_P and κ_I are specified by the user.

5.4 Water Vapor Supplier

The water vapor supplier adds water vapor at each time step if the following condition is satisfied.

$$R(i,j) < \frac{1}{N_x N_y}\sum_{l=1}^{N_x}\sum_{m=1}^{N_y} R(l,m). \qquad (14)$$

Eq. (14) indicates that the water vapor is supplied at the grid points where the ratio $R(i,j)$ is less than the average of the ratios. That is, the water vapor supplier tries to make the ratio of the cloud growth the same for all grid points. Therefore, the top of the cloud at all grid point reaches the top of the target shape almost simultaneously.

The amount of additional water vapor, $S_{v,c}$, is given by the following equation.

$$S_{v,c}(i,j) = c_v q_{v,0}(i,j,k_{top}), \qquad (15)$$

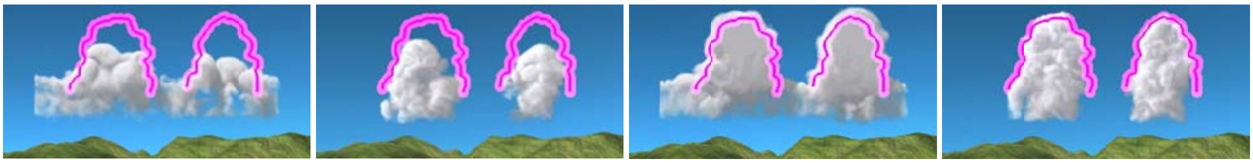
where c_v is a control parameter for the water vapor supplier and is specified by the user. $q_{v,0}$ is the initial water vapor at the beginning of the simulation and (i,j,k_{top}) indicates the grid point corresponding to the top of the simulated clouds.

6 Results

Examples shown in this section were calculated by using a desktop PC with Intel Core2 Extreme X9650 with nVidia GeForce 8800 GTX. The simulation was carried out on the CPU and the GPU is used for rendering clouds. We used the method proposed by Dobashi et al. [2000] for the rendering. Please refer to the accompanying movie files for animations of the examples shown in this section. The simulation space is subdivided into a 320 × 80 × 100 grid. The average computation time for each time step of the simulation was 7 seconds. The additional computational cost due to our control mechanism is very low and is less than one percent of the total computation cost.

In the following examples, we set the control parameters, κ_I, κ_P, and c_v, to 4.95, 0.6, and 0.5, respectively. For tuning κ_P and κ_I, we used a well-known technique developed by Ziegler and Nichols [1942] (see Appendix A). We determined the value of c_v by a process of trial and error. However, once the appropriate values for these parameters have been found, we can generate various shapes of clouds as shown in the following examples. The parameters involved in the cloud simulation are summarized in Table 1. These are determined according the previous methods on the cloud simulation [Miyazaki et al. 2002; Harris et al. 2003]. We adjusted the parameters so that the time step and the grid interval become 1.0 as shown in the table.

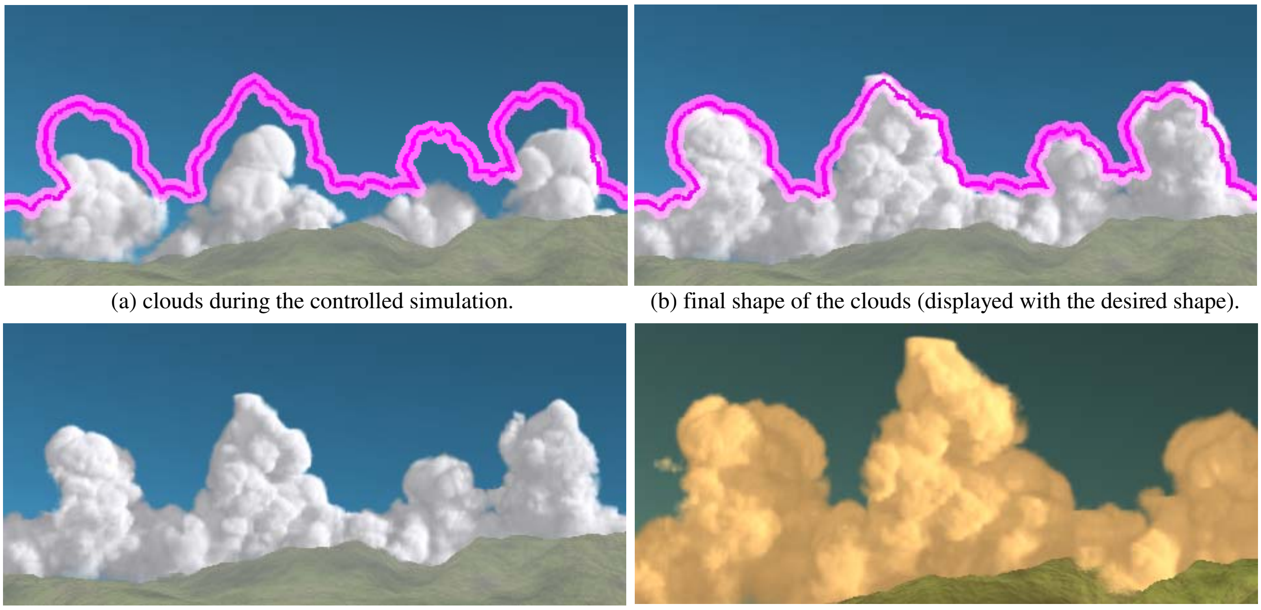
First, in order to demonstrate the effectiveness of our method, we compared our method with one of the existing methods, as shown in Fig. 6. We implemented the method proposed by Fattal et al. [2004] and combined with the cloud simulation. Figs. 6(a) and (b) show the clouds before and after reaching the target shape, respectively. The images on the left are generated by using Fattal's method and the images on the right by our method. The clouds generated by



(Fattal's method) (our method) (Fattal's method) (our method)

(a) clouds before reaching the target shape. (b) clouds after reaching the target shape.

Figure 6: *Comparison between results using our method and Fattal's method.*



(a) clouds during the controlled simulation.

(b) final shape of the clouds (displayed with the desired shape).

(c) final shape of the clouds (without the desired shape).

(d) clouds viewed from different viewpoint at different time.

Figure 7: *Clouds with complex shapes.*

Table 1: Parameter setting of cloud simulation.

parameters	meaning	value
dt	time step	1.0
dh	grid interval	1.0
k_b	coefficient for buoyancy force (Eq. 3)	1.0
g	coefficient for gravitational force (Eq. 3)	0.2
T_{amb}	ambient temperature (Eq. 3)	linear change from 2.5 (bottom) to 3.0 (top)
S_v	vapor supply from vapor source (Eq. 5)	0.0
α	phase transition rate (Eq. 6)	0.1
(A, B, C)	coefficients for saturation vapor content (Eq. 7)	(100, 3, -2.3)
Γ_d	adiabatic lapse rate (Eq. 8)	0.02
S_T	heat supplied from the ground (Eq. 8)	0.2

the previous method are not very realistic. In addition, clouds are generated where they should not be. Our method can generate realistic clouds while their shapes closely match the target shapes. Fig. 6 indicates the importance of our control mechanism specifically designed for clouds.

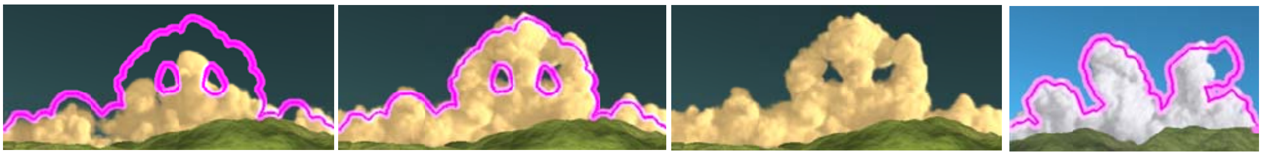
Next, Figs.1 and 7 show practical examples. In Fig.1, the desired shape is relatively simple. In this case, the target shape becomes the height field. The shape of the clouds generated by our method is almost the same as the desired shape. In Fig. 7, we specify a more characteristic shape. In this case, the target shape is not the height field. However, our method can generate clouds that closely match the desired shape. Fig. 7 (d) shows the same clouds viewed from a different viewpoint at a different time. Since the clouds are generated in three-dimensions, we can render the clouds with arbitrary viewpoints and illumination.

Fig. 8 shows examples where the target shapes are completely different from height fields. In Fig. 8(a), the user specifies an unnatural shape of clouds, a skull. The leftmost image shows the clouds during the simulation. The center and the rightmost images show the final shape of clouds with and without the desired shape, respectively. Although there are differences between the simulated clouds and the desired shape, our method succeeded in creating the convincing result. Next, Fig. 8(b) shows a failure case. In this case, the rightmost part of the clouds does not match the target shape although the target shape is rather natural compared to Fig. 8(a).

Finally, in Fig. 9, we generate clouds that resemble the real clouds in a photograph. The contour line of the real clouds is extracted. Then this contour line is used to control the simulation by our method (Fig. 9(a)). Fig. 9(b) is the simulated clouds. The similar shape has been obtained. Fig. 9(c) shows the same clouds viewed from a different viewpoint. Fig. 9(d) is the same clouds at evening.

7 Discussion and Future Work

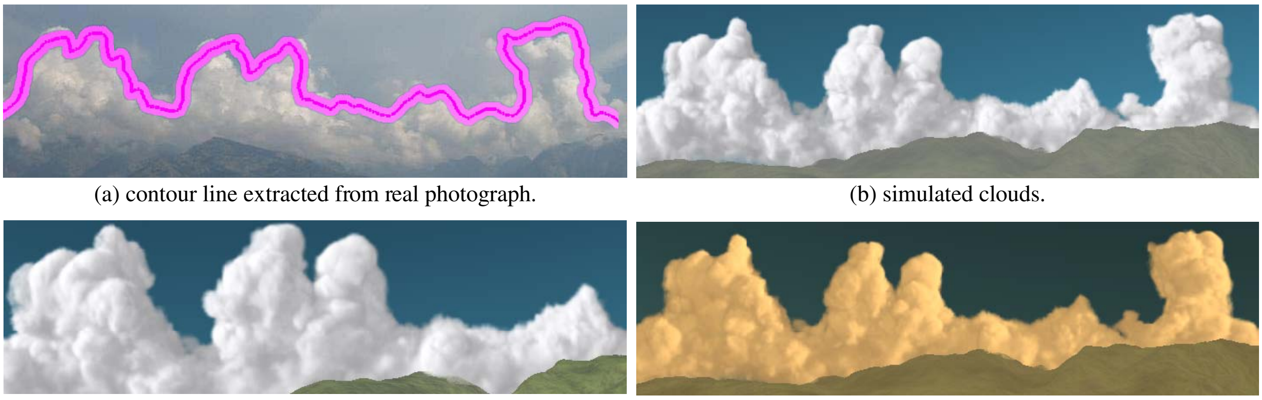
Our control mechanism is fairly indirect. Therefore, our feedback controllers cannot guarantee that the clouds will completely form the desired shape. However, the indirect control makes it possible to promote the cloud growth as naturally as possible. Furthermore, in terms of our purpose for generating realistic-looking clouds, it is



(a) unnatural shape of clouds (shape of a skull).

(b) a failure case.

Figure 8: *Examples of clouds generated by using the target shapes that are completely different from height fields.*



(a) contour line extracted from real photograph.

(b) simulated clouds.

(c) different viewpoint.

(d) different illumination.

Figure 9: *Clouds generated by using a real photograph.*

not so important that the shape of the simulated clouds matches the target shape completely. If the clouds completely match the target shape, the user has to specify realistic target shape. The indirect control allows the user to specify the rough shape of the clouds and finally generate the realistic clouds.

Our method controls the cloud motion in the vertical direction. The horizontal movement of the clouds is not controlled. As described in Section 3, the cloud formation process mainly affects the vertical movement of the clouds since the forces due to cloud formation work only in the vertical direction (see Eq. (3)). Therefore, it is difficult to control the horizontal movement by controlling the cloud formation process. The previous fluid control methods might be suitable for horizontal control. Therefore, it is expected that the combination of our method with one of the previous methods provides us with a way to control the cloud motion in both vertical and horizontal directions.

We have designed our method so that it can handle a desired shape that is not a height field. However, it is still difficult to handle a shape that is very different from a height field. The reason for this is the same as the one described above. That is, the cumuliform cloud formation process affects the vertical movement of the clouds. To generate clouds whose shape is very different from the height field, horizontal external forces are required. This problem can also be solved by combining our method with the previous methods.

In this paper, our goal for generating realistic clouds forming the desired shape has been achieved. Extending the method to handling multiple target shapes is our next important research direction. By specifying a set of target shapes at several key-frames, it is expected to realize the shape transition of clouds. We have carried out some experiments about this issue. We have found that a mechanism to make clouds disappear need to be introduced to realize this extension. Our next future work is to find the mechanism.

In our current method, the user specifies the shape of clouds viewed from only a single direction. This is suitable for clouds that exist far from the viewpoint. However, in some applications such as flight simulators, this is not the case. Extending our method to handing this kind of situation is also an important future work. One solution could be to specify the multiple target shapes viewed from multiple viewpoints.

Although we used the Ziegler-Nichols method for tuning the control parameters, this method does not always provide 'good' parameters. According to our experiments, a trial and error process is still required. In the field of control and system engineering, more sophisticated methods have been proposed [Astrom and Haqqlund 1995; O'Dwyer 2006]. We are currently investigating these methods for automatic determination of the parameters.

Finally, extension of our method to other types of clouds such as stratus is also an interesting area for future work. There seems to be other important factors that are not considered in our method, such as Bénard convection [Miyazaki et al. 2001]. We need to investigate such factors in order to realize this extension.

8 Conclusion

We have proposed a simple method for controlling cumuliform cloud simulation. Our method can generate clouds with desired shapes specified by the user. In the method, the cloud formation process is controlled by our feedback controller and external forces calculated by the geometric potential field. For feedback control, we developed a latent heat controller and a water vapor supplier. By controlling the amounts of latent heat and additional water vapor through these controllers, clouds grow naturally and converge into the desired shape. Our method provides a simple way to generate realistic clouds with desired shapes.

Appendix A. Ziegler-Nichols Method

We determine κ_P and κ_I, based on the Ziegler-Nichols method [Ziegler and Nichols 1942]. This method is a well-known technique for tuning the parameters of PID controllers. We apply the method in the following way. First, experimental simulations are carried out several times with the proportional controller only. Let us denote the proportional gain for the experimental simulations as κ_C. The simulations are carried out with increasing κ_C. and the controller tries to make the clouds reach a target height specified by the user. In our experiment, the target height is set to 80 percent of the simulation space. When κ_C is small, the clouds cannot reach the target height. However, as κ_C increases, the clouds can reach the height and, at a certain value of κ_C, the cloud growth exhibits periodical oscillations. That is, clouds repeatedly exceed and fall below the target height. Let us denote the period of the oscillation as T_C. Then, according to Ziegler and Nichols [1942], the proportional and the integral gain, κ_P and κ_I, of our latent heat controller are determined by the following equations.

$$\kappa_P = 0.45\kappa_C, \quad \kappa_I = \frac{\kappa_P}{0.85T_C}. \tag{16}$$

Acknowledgments

We thank the SIGGRAPH reviewers for their constructive comments; Dr. Ken Anjyo for valuable discussions. This research is partly supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) from the Ministry of Internal Affairs and Communications of Japan.

References

ASTROM, K. J., AND HAQQLUND, T. 1995. *PID Controllers: Theory, Design, and Tuning*. International Society for Measurement and Con.

BOUTHORS, A., AND NEYRET, F. 2002. Modeling clouds shape. In *Proceedings of Eurographics 2004 (short papers)*.

DOBASHI, Y., KANEDA, K., YAMASHITA, H., OKITA, T., AND NISHITA, T. 2000. A simple, efficient method for realistic animation of clouds. In *Proceedings of ACM SIGGRAPH 2000, Annual Conference Series*, 19–29.

EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., PERLIN, K., AND WORLEY, S. 2002. *Texturing & modeling: a procedural approach*. Morgan Kaufman.

FATTAL, R., AND LISCHINSKI, D. 2004. Target-driven smoke animation. *ACM Transactions on Graphics 23*, 3 (Aug.), 439–446.

FELDMAN, B. E., O'BRIEN, J. F., AND ARIKAN, O. 2003. Animating suspended particle explosions. In *Proceedings of ACM SIGGRAPH 2003*, 708–715.

FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Proceedings of ACM SIGGRAPH 2001*, 23–30.

GARDNER, G. Y. 1985. Visual simulation of clouds. *Computer Graphics (Proceedings of SIGGRAPH 1985) 19*, 3 (July), 297–304.

HARRIS, M. J., BAXTER, W. V., SCHEUEMANN, T., AND LASTRA, A. 2003. Simulation of cloud dynamics on graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware 2003*, 92–101.

HAUGEN, F. 2004. *PID Control Of Dynamic Systems*. Tapir Forlag.

HONG, J.-M., AND KIM, C.-H. 2004. Controlling fluid animation with geometric potential. *Computer Animation and Virtual Worlds 15*, 3-4 (July), 147–157.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a sketching interface for 3d freeform design. In *Proceedings of ACM SIGGRAPH 1999*, 409–416.

KAJIYA, J. T., AND HERZEN, B. P. V. 1984. Ray tracing volume densities. *Computer Graphics (Proceedings of SIGGRAPH 1984) 18*, 3 (Aug.), 165–174.

KIM, Y., MACHIRAJU, R., AND THOMPSON, D. 2006. Path-based control of smoke simulations. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 33–42.

KIM, B., LIU, Y., LLAMAS, I., JIAO, X., AND ROSSIGNAC, J. 2007. Simulation of bubbles in foam with the volume control method. *ACM Transactions on Graphics 26*, 3 (July), Article 98.

MCNAMARA, A., TREUILLE, A., POPOVIC, Z., AND STAM, J. 2004. Fluid control using the adjoint method. *ACM Transactions on Graphics 23*, 3 (Aug.), 449–456.

MIYAZAKI, R., YOSHIDA, S., NISHITA, T., AND DOBASHI, Y. 2001. A method for modeling clouds based on atmospheric fluid dynamics. In *Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, 363–372.

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.

NEYRET, F. 1997. Qualitative simulation of convective cloud formation and evolution. In *Proceedings of Eurographics Computer Animation and Simulation Workshop 1997*, 113–124.

NGUYEN, D. Q., FEDKIW, R., AND JENSEN, H. W. 2002. Physically based modeling and animation of fire. *ACM Transactions on Graphics 21*, 3 (Aug.), 721–728.

O'DWYER, A. 2006. *Handbook of Pi And Pid Controller Tuning Rules*. Imperial College Press.

SHI, L., AND YU, Y. 2005. Taming liquids for rapidly changing targets. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 229–236.

STAM, J. 1999. Stable fluids. In *Proceedings of ACM SIGGRAPH 1999, Annual Conference Series*, 121–128.

THÜREY, N., KEISER, R., PAULY, M., AND RÜDE, U. 2006. Detail-preserving fluid control. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 7–12.

TREMBILSKI, A., AND BROSSLER, A. 2002. Surface-based efficient cloud visualisation for animation applications. In *Proceedings of The 10-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2002 (WSCG 2002)*, 453–460.

TREUILLE, A., MCNAMARA, A., POPOVIC, Z., AND STAM, J. 2003. Keyframe control of smoke simulations. *ACM Transactions on Graphics 22*, 3 (July), 716–723.

ZIEGLER, J. G., AND NICHOLS, N. B. 1942. Optimum settings for automatic controllers. *Transactions of the A. S. M. E. 64* (Nov.), 759–768.