



HOKKAIDO UNIVERSITY

Title	Controlling the Simulation of Cumuliform Clouds Based on Fluid Dynamics
Author(s)	KAWAGUCHI, Tatsuki; DOBASHI, Yoshinori; YAMAMOTO, Tsuyoshi
Citation	IEICE Transactions on Information and Systems, E98.D(11), 2034-2037 https://doi.org/10.1587/transinf.2015EDL8085
Issue Date	2015-11
Doc URL	https://hdl.handle.net/2115/62780
Rights	Copyright ©2015 The Institute of Electronics, Information and Communication Engineers
Type	journal article
File Information	e98-d_11_2034.pdf





on Information and Systems

**VOL. E98-D NO. 11
NOVEMBER 2015**

The usage of this PDF file must comply with the IEICE Provisions on Copyright.

The author(s) can distribute this PDF file for research and educational (nonprofit) purposes only.

Distribution by anyone other than the author(s) is prohibited.

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY



The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

LETTER

Controlling the Simulation of Cumuliform Clouds Based on Fluid Dynamics

Tatsuki KAWAGUCHI^{†a)}, Yoshinori DOBASHI[†], and Tsuyoshi YAMAMOTO[†], *Members*

SUMMARY Controlling fluid simulation is one of the important research topics in computer graphics. In this paper, we focus on controlling the simulation of cumuliform cloud formation. Using a previously proposed method for controlling cloud simulation the convergence speed is very slow; therefore, it takes a long time before the clouds form the desired shapes. We improved the method and accelerated the convergence by introducing a new mechanism for controlling the amount of water vapor added. We demonstrate the effectiveness of the proposed method by several examples.

key words: cloud, fluid simulation, control, computer graphics

1. Introduction

In entertainment applications, such as movies and video games, images of fluid phenomena are often used to produce certain visual effects including smoke, fire, clouds, and water. However, capturing real images that satisfy the user's intention is often difficult. Therefore, computer graphics (CG) techniques are often used to synthesize realistic images of fluids.

There are many methods for simulating fluids using CG. One promising approach is physically based simulation of fluid dynamics, which can synthesize realistic animations of fluids. However, even when using fluid simulation, it is still difficult to produce the shapes and motion desired by the user. In order to address this problem, several methods have been proposed for controlling fluid simulation [4]–[6]. Among such control methods, we focus on controlling the simulation of clouds. Dobashi et al [1] proposed a method for controlling the simulation of cumuliform clouds so that the clouds form the desired shapes specified by the user. In this method, the user specifies the desired shape by drawing its outline and the method controls the simulation of the cloud growth until it eventually matches the specified outline. This method is useful for synthesizing images of outdoor scenes because clouds are important background elements in scenes that include the sky.

However, Dobashi's method has two problems. The first one is the slow convergence, which means it takes a long time until the clouds form the desired shapes. The second problem is that the method cannot handle shapes with holes. In this paper, we address the first problem. We in-

vestigated Dobashi's method [1] thoroughly through many experiments and identified the reason for the slow convergence. We present a method to improve the convergence speed and demonstrate the effectiveness of the method.

The rest of this paper is organized as follows. In Sect. 2, we describe the fluid simulation of clouds and the control method proposed by Dobashi et al. In Sect. 3, we discuss the convergence problem in Dobashi's method and propose a method to address it. Next, the experimental results and analysis are shown in Sect. 4. Finally, we conclude the paper in Sect. 5.

2. Previous Method for Cloud Control

In this section, we discuss two related methods. First, we describe the fluid simulation of cumuliform clouds. Next, we describe Dobashi's method for controlling the cloud simulation.

2.1 Fluid Simulation of Cumuliform Clouds

We employ the atmospheric fluid model proposed by Miyazaki et al [2] to simulate the cumuliform clouds. The physical processes for cumuliform cloud formation are illustrated in Fig. 1. First, the ground is heated by the sun and air parcels near the ground start to move upward due to the thermal buoyancy force. The temperature of these air parcels decreases due to adiabatic cooling. The phase transition of water vapor into water drops then occurs due to the decrease in temperature of the air parcels. Moreover, during the phase transition, latent heat is released and this creates additional buoyancy forces that promote further cloud growth into higher regions, resulting in cumuliform cloud formation.

In order to simulate the above processes, Miyazaki's method [2] subdivides the simulation space into a grid, and physical quantities are stored and calculated at each grid point. The governing equations are discretized by using the grid and numerically solved by using the finite difference method. For more details, please refer to Miyazaki's paper [2].

2.2 Controlling Cumuliform Clouds

In Dobashi's method [1], the user firstly specifies the desired shape by drawing its outline. A three-dimensional target shape is created from the outline and the method controls

Manuscript received April 7, 2015.

Manuscript revised June 29, 2015.

Manuscript publicized July 24, 2015.

[†]The authors are with Faculty of Engineering, Hokkaido University, Sapporo-shi, 060–8628 Japan.

a) E-mail: kawaguchi@ime.ist.hokudai.ac.jp

DOI: 10.1587/transinf.2015EDL8085

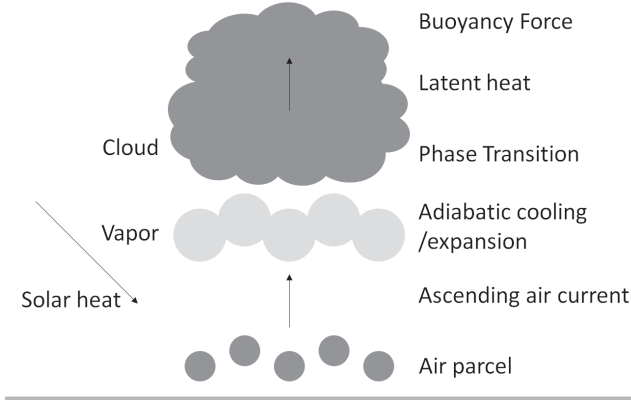


Fig. 1 Atmospheric fluid model for cumuliform clouds

the simulation so that the shape of the cloud matches the target shape.

The cloud simulation is controlled through the following two components: a feedback control and artificial external forces. The feedback control is responsible for the cloud growth in the vertical direction; it automatically adjusts the simulation parameters according to the height difference between the simulated cloud and the target shape. Two parameters are adjusted to control the simulation. One is the latent heat coefficient, which affects the latent heat caused by the phase transition. The other parameter is the amount of water vapor artificially added to promote the phase transition. Together with the feedback control, the artificial external forces are applied in the horizontal direction to prevent the clouds from growing beyond the target shape. For more details, please refer to [1].

3. Proposed Method

In this section we describe our method for improving the convergence speed of the previous method [1]. We performed a number of experiments to investigate the reasons for the slow convergence. From the experiments, we found that the convergence speed depends significantly on the amount of water vapor that is artificially added in order to promote the phase transition. Based on this finding, we propose a new formulation for adding the water vapor. In the following, let us assume that the number of the grid points for simulating the cloud formation is $N_x \times N_y \times N_z$. We also assume that the x and z axes correspond to the horizontal directions and the y axis to the vertical direction.

In the previous method, the amount of additional water vapor, V_c , is computed by the following equation:

$$V_c = C_v q_{amb}(H(x, z)), \quad (1)$$

where C_v is a user-specified parameter that adjusts the amount of water vapor to be added, q_{amb} is the ambient water vapor density, and $H(x, z)$ is the height of the simulated cloud at the horizontal position (x, z) . The ambient water vapor $q_{amb}(H(x, z))$ is the amount of water vapor that can be included in air at its ambient temperature. Since Dobashi et

al assume that the ambient temperature is a function of the height from the ground, both q_{amb} and V_c are also functions of the height. This indicates that the previous method using Eq. (1) does not take into account the degree of cloud growth in adding the water vapor. This causes the following problem. The previous method increases the temperature by adjusting the latent heat coefficient in the regions where the simulated cloud does not reach the target shape. This produces strong buoyancy forces and vertical cloud development is promoted. However, when the user specifies an inappropriately small value for C_v in Eq. (1), the amount of water vapor is also small and the phase transition is less likely to occur. That is, the clouds are not generated and the difference between the simulated clouds and the target shape does not decrease. Therefore, the previous method requires many simulation time steps until a sufficient amount of water vapor is supplied. This is the reason for the slow convergence.

The above investigation provides us with the following new method to improve the convergence speed. The problem with the previous method is the fact that the amount of additional water vapor is independent of the vertical cloud development caused by the increased temperature. Therefore, our method addresses this problem by adding water vapor in proportion to the temperature. That is, we compute the amount of the additional water vapor by the following equation.

$$V_c = C_v q_s(T(x, H(x, z), z)), \quad (2)$$

where q_s is the saturated water vapor density. The saturated water vapor density $q_s(T(x, H(x, z), z))$ is the maximum amount of water vapor that can be contained in air at temperature $T(x, H(x, z), z)$. Therefore, the amount of additional water vapor now becomes a function of temperature $T(x, y, z)$.

4. Experimental Results

This section shows some experimental results and demonstrates the efficiency of our method. We first describe our experimental setup and then show the results.

4.1 Experimental Setup

In order to verify the efficiency of the proposed method, we compared it with the previous method using the following experimental setup. We specified the simple target shape shown in Fig. 2 for our experiment. The left image in Fig. 2 is the user-specified outline and the corresponding three-dimensional target shape shown in the right image of Fig. 2 was created using the method described in [1]. For both the proposed and the previous methods, we executed a set of simulations with different values for C_v (see Eqs. (1) and (2)), ranging from 0.3 to 2.0 sampled at regular intervals of 0.1. Then, we measured the convergence speed of each simulation. The number of grid points is shown in Table 1.

Next, before showing the results, let us briefly discuss

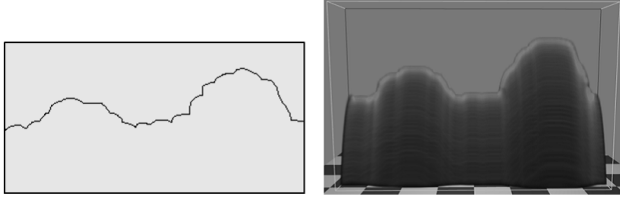


Fig. 2 The shape used for the experiments: a hand-drawn outline (left) and the 3D volume generated from the outline (right).

Table 1 Simulation parameters used in the experiments

grid size	n	ε_c	R_{thresh}
$300 \times 150 \times 100$	5	0.001	0.15

the convergence condition used to terminate the simulation. A straightforward condition is to use a user-specified threshold for the difference between the target shape and the shape of the simulated cloud. However, we found that the simulated cloud lost its realistic appearance when the shape completely matched the target shape. Therefore, we introduced new convergence conditions; we consider the simulation has converged when the cloud growth stops or the cloud grows beyond the target shape. The details are described in the following.

First, we define the coincidence ratio $\bar{R}$ by:

$$\bar{R} = \frac{1}{N_x N_z} \sum_{l=0}^{N_x-1} \sum_{m=0}^{N_z-1} |1 - R(l, m)|, \quad (3)$$

$$R(l, m) = H(l, m) / H_{target}(l, m), \quad (4)$$

where R is the ratio between the height of the simulated clouds H and the height of the target shape H_{target} at the horizontal position (l, m) . When the coincidence ratio becomes zero, the shape of the simulated cloud is identical to the target shape. Using the coincidence ratio, we assume the simulation has converged if the following two conditions are satisfied:

$$\bar{R}(t) - \bar{R}(t - n) > \varepsilon_c, \quad (5)$$

$$\bar{R}(t - n) \leq R_{thresh}, \quad (6)$$

where t is time and n , ε_c , and R_{thresh} are specified by the user to adjust the convergence conditions. Eq. (5) means that the coincidence ratio increases by ε_c during n time steps. Eq. (6) indicates that the coincidence ratio is smaller than R_{thresh} at time step $t - n$. By using these conditions, we can consider the controlled simulation has converged even when the shape of the simulated clouds is different from the target shape. The user can adjust the degree of deviation by using the three parameters n , ε_c , and R_{thresh} . Table 1 shows the values of these parameters used for the experiments described in the next subsection.

4.2 Experimental Results

We measured the number of time steps required for the simulation to converge and the results are shown in Fig. 3. The

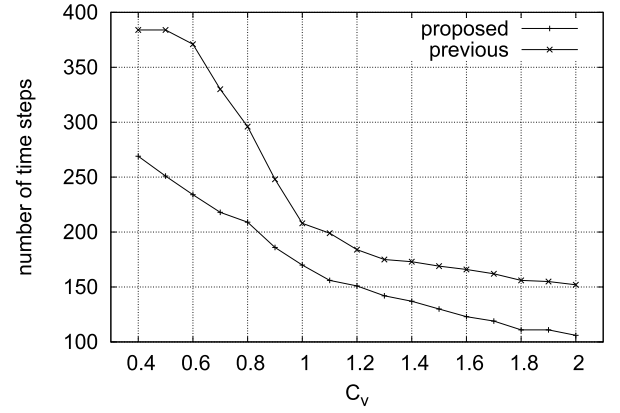


Fig. 3 Comparison between the convergence speeds of the proposed method and the previous method.

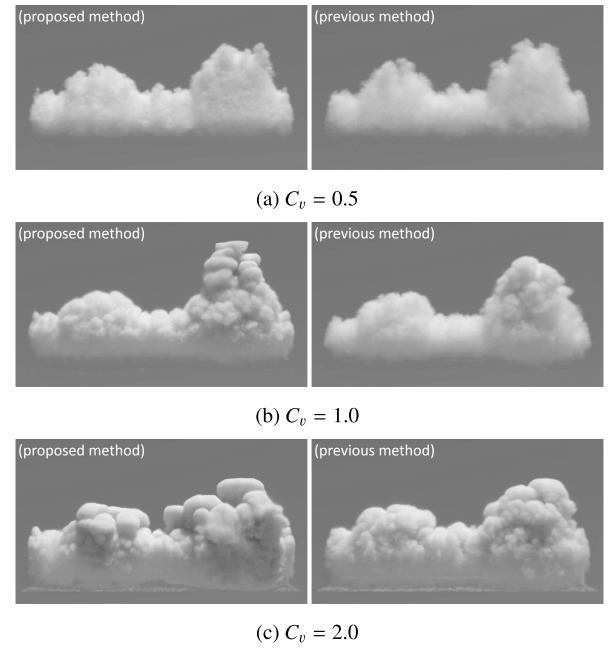


Fig. 4 Shapes of simulated clouds with different values of C_v . The left column shows clouds simulated by the proposed method and the right column by the previous method.

horizontal axis represents the value of C_v . The vertical axis is the number of time steps.

Figure 4 shows images of the simulated clouds for $C_v = 0.5, 1.0$, and 2.0 , respectively, and Fig. 5 shows snapshots during the simulation at $t = 100, 150$, and 200 , respectively. For Fig. 5, C_v is set to 0.5 . To visualize the clouds, we used the Mitsuba renderer [3].

4.3 Discussion

From Figs. 3 and 5, we see that the amount of additional vapor clearly affects the convergence speed. The convergence speed of the proposed method is faster than that of the previous method. However, from Fig. 4, the shapes of the clouds are not very realistic when $C_v = 1.0, 2.0$ (proposed method)

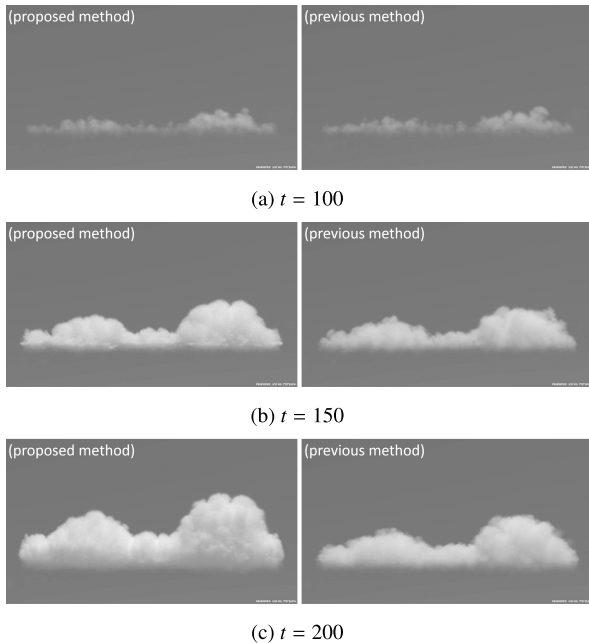


Fig. 5 Snapshots from the simulation. The left column shows clouds simulated by the proposed method and the right column by the previous method. $C_v = 0.5$ is used for creating these images.

Table 2 Stably controllable range of C_v

	our method	previous method
range of C_v	0.3 ~ 0.9	0.4 ~ 1.0

or $C_v = 2.0$ (previous method). This is due to the excessive amount of additional water vapor. In order to find the range of C_v that produces realistic clouds, we examined all the images of the clouds generated in the above experiment. From our investigation, we found that the clouds looked realistic when we chose C_v from the ranges shown in Table 2. Although these are determined by our subjective evaluations and valid only for the target shape shown in Fig. 2, we found experimentally that realistic clouds are likely to be generated for various target shapes.

Our experiments have demonstrated that the convergence speed is faster using our method. However, adding large amounts of water vapor often results in clouds with unrealistic appearances. Thus, there is a trade off between the convergence speed and the realism of the resulting clouds. Although it is difficult to find the optimal value of C_v , we recommend using $C_v = 0.5$. From our experiments, this value often produced satisfactory results with a reasonable convergence speed. Figure 6 shows an image created with $C_v = 0.5$ for a target shape that is different from the one shown in Fig. 2. Realistic clouds are generated.

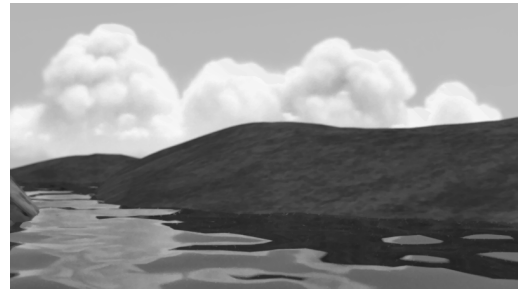


Fig. 6 Application of our method to the synthesis of clouds in a typical outdoor scene.

5. Conclusion

In this study, we proposed a method for improving the convergence speed of Dobashi's method [1]. We identified the important factor that significantly affects the convergence speed, that is, the amount of additional water vapor. In order to improve the convergence speed, we proposed a new way to computing the amount of additional water vapor. Our experiment clearly demonstrates the efficiency of the proposed method.

However, there are a few issues to be addressed in the future. The cloud growth does not depend on the water vapor only. There are many other factors, such as the latent heat, the heat source, and so on. The parameters related to these factors are not taken into account in this paper and will be studied in the future. Also, we intend to address another limitation of the previous method [1]; it cannot handle target shapes with holes. We are planning to extend the method to those shapes by introducing new external forces.

References

- [1] Y. Dobashi, K. Kusumoto, T. Nishita, and T. Yamamoto, "Feedback Control of Cumuliform Cloud Formation based on Computational Fluid Dynamics," In Proc. SIGGRAPH '08, Article 94, 2008.
- [2] R. Miyazaki, S. Yoshida, T. Nishita, and Y. Dobashi, "A method for modeling clouds based on atmospheric fluid dynamics," In PG '01: Proceedings of the 9th, Pacific Conference on Computer Graphics and Applications, pp.363–372, 2001.
- [3] W. Jakob, "Mitsuba renderer," <http://www.mitsuba-renderer.org>, 2010.
- [4] R. Fattal and D. Lischinski, "Target-driven smoke animation," In SIGGRAPH '04: ACM SIGGRAPH 2004 Papers, pp.441–448, 2004.
- [5] J.-M. Hong and C.-H. Kim, "Controlling fluid animation with geometric potential: Research article," Comput. Animat. Virtual Worlds, vol.15, no.3-4, pp.147–157, 2004.
- [6] N. Thürey, R. Keiser, M. Pauly, and U. Rüdè, "Detailed-preserving fluid control," In SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp.7–12, Eurographics Association, 2006.