

Simulation of Cumuliform Clouds Based on Computational Fluid Dynamics

R. Miyazaki[†], Y. Dobashi[‡], T. Nishita[†]

The University of Tokyo[†] Hokkaido University[‡]

Abstract

Simulation of natural phenomena is one of the important research fields in computer graphics. In particular, clouds play an important role in creating images of outdoor scenes. Fluid simulation is effective in creating realistic clouds because clouds are the visualization of atmospheric fluid. In this paper, we propose a simulation technique, based on a numerical solution of the partial differential equation of the atmospheric fluid model, for creating animated cumulus and cumulonimbus clouds with features formed by turbulent vortices.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Physically based modeling

1. Introduction

Clouds play an important role when images, such as outdoor scenes, the earth viewed from outer space and the visualization of weather information, are generated. The shapes of clouds depend on the environment under which they are formed, for instance, they depend on the ascending air currents, temperature and humidity. We often observe clouds with their various fascinating appearances, as they change, and as they disappear with time. Therefore, many researchers have tried to create realistic images of clouds.

An important element in the synthetic animation of clouds is the expression of complex cloud dynamics. The best method to achieve this is by simulating the physical processes, especially the atmospheric fluid dynamics, that characterizes the shape of clouds. The modeling of clouds is important in other fields, such as earth science, weather forecasting, and so on. However simulations in these fields do not give priority to the cloud shape, and the time and spatial scale is too large when viewed from the ground. Moreover accurate simulation, as used in forecasting the weather, is not necessarily demanded by computer graphics. On the other hand, many methods using fluid simulation for smoke, gases, and water in computer graphics have been proposed. However, there has been little research based on the physical simulation of clouds, since the exact simulation of atmospheric fluid dynamics is very difficult and computationally expensive. Nevertheless, exact simulation is not important in order to simulate visually convincing clouds. In this pa-

per we propose a simplified atmospheric fluid model. This model allows us to create realistic cloud animation such as cumulus clouds appearing and disappearing, being carried by the wind, and cumulonimbus clouds developing into sky-high like towers.

The rest of the paper is organized as follows. First we discuss the previous work related to clouds and then present an outline of our methods. In the next Section we explain cumuliform clouds simulated in this paper. In Section 3, we discuss the basic equations of atmospheric fluid dynamics and numerical solutions for these. In Section 4, the simulation conditions and some resultant images are shown. Section 5 concludes and discusses future work.

1.1. Previous Work

Various techniques for modeling clouds for use in computer graphics have been proposed in the past 20 years. One approach is the heuristic approach. The methods that take this approach use fractals^{9, 17, 18, 23}, procedural modeling^{3, 4, 5, 6, 13, 14}, qualitative simulation^{12, 16} and stochastic modeling²⁰. Although these techniques can create realistic still images of clouds, they are limited when realistic cloud motion is required.

Dobashi et al. developed a fast method for simulating cloud motion using the idea of a cellular automaton². In their method, however, they use an extremely simplified model for the physical process of cloud formation. There-

fore, complex dynamics are not simulated and only cumulus clouds are demonstrated in their paper. In an image-based modeling approach, Dobashi et al. used satellite images to create clouds such as a typhoon as it appears viewed from space ¹. However, using this method, clouds with the desirable shapes cannot be created, and, in addition, a lot of satellite images are required.

A more natural way to model the motion of clouds is to solve the equations of fluid dynamics directly. In computer graphics, Kajiyama et al. were the first to use numerical methods. In their method, the equations of atmospheric fluid dynamics are solved numerically ¹¹. However, this model does not include adiabatic cooling and the temperature lapse in simulation space, which is important for cumulus cloud dynamics. In addition, the result is not realistic. There are many researches of gas simulations ^{8, 19, 21}. Recently, Stam introduced a stable fluid simulation model ²². This was achieved by a semi-Lagrangian advection. Because a first order integration scheme was used, the simulations suffered from too much numerical dissipation. Although the overall motion looks fluid-like, small scale vortices vanish too rapidly. So Fedkiw et al. introduced a physically consistent vorticity confinement term to model the small scale rolling features that are characteristic of smoke ⁷. Their method is, however, focused on the motion of smoke and the possibility of the application of their method to modeling clouds is not discussed.

Miyazaki et al. improved a qualitative model of cloud simulation ^{24, 25} using CML (Coupled Map Lattice). CML is an extension of cellular automaton and is an approximation technique to reduce the calculation cost. They developed a method that can create various clouds based on simulation of ascending air current and the Bénard convection ¹⁵. Originally the CML model was designed for simulating the Bénard convection. In the simulation of ascending air currents, the temperature distribution is assumed to be invariable in simulation space. Although the shape looks like a cumulus cloud, the advection of the temperature that is fundamental to the dynamics of cumulus clouds cannot be simulated. Moreover, the CML model, whose calculation cost is low, has a deficiency in that the fractal structure of the fluid vortex cannot be generated.

1.2. Our Method

We propose a cloud simulation model. This includes the phase transition and adiabatic cooling that is not included in smoke simulation ⁷. Our method can simulate more realistic clouds than previous cloud models ^{11, 15}. Our model simulates the interaction of the vapor, the cloud, the temperature and the velocity vector. This model is very suitable for the simulation of cumuliform clouds, where a large vertical movement and fluid vortices are essential. In the simulation, an ascending air current is generated due to the buoyancy created by the heat source, which is specified by the user.

The air current carries the temperature and the vapor upwards. Then the vapor coagulates, and clouds are generated. At the same time the latent heat is liberated and this creates additional buoyancy.

After the simulation, we generate metaballs at the center of voxels in order to render the clouds. The density at the center of each metaball is set to the density of the cloud particles in the corresponding voxel. Then, images of clouds are generated using the hardware-accelerated rendering method proposed by Dobashi et al. ².

2. Cumuliform Clouds

The cumuliform clouds (cumulus and cumulonimbus) are generated by strong ascending air currents. The temperature of rising air currents decreases due to adiabatic cooling, so the vapor included in an air parcel causes a phase transition, it coagulates, and the cloud is generated. The latent heat is liberated at that time, resulting in further development of the clouds. Cumuliform clouds are generally dense and have a sharp outline like a cauliflower (see Fig. 1). Cumulonimbus is an advanced stage of cumulus development with considerable vertical extent, in the form of a mountain or a huge tower.



Figure 1: Photographs of cumuliform clouds (left: cumulus, right: cumulonimbus)

3. Cumuliform Cloud Simulation

The simulation space is subdivided into voxels. The number of voxels is $N_x \times N_y \times N_z$. The velocity vector $\mathbf{v} = (v_x, v_y, v_z)$, the vapor density w_{vap} , the cloud (water droplets) density w_{cl} , and the temperature E are assigned to each voxel as state variables at time step t . Each state variable is updated at every time step. The voxel width is h and the time interval is Δt .

3.1. Basic Equations

The atmospheric fluid is modeled by the following partial differential equations. We assume that the air density is constant, so the atmospheric fluid is incompressible. This is called the Boussinesq approximation ¹⁰. For many applications of cloud dynamics in computer graphics and other fields, the Boussinesq equations are sufficient and efficient.

$$\frac{D\mathbf{v}}{Dt} = -\nabla p + \nu \Delta \mathbf{v} + \mathbf{B} + \mathbf{f}, \quad (1)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (2)$$

$$\frac{DE}{Dt} = -\Gamma_d + Q \frac{Dw_{cl}}{Dt} + S_E, \quad (3)$$

$$\frac{Dw_{vap}}{Dt} = -\frac{Dw_{cl}}{Dt}. \quad (4)$$

Eqs. (1) and (2) are the Navier-Stokes equations. Eq. (1) is a vector equation of the velocity field. Eq. (2) is the continuity equation, meaning that it expresses the conservation of mass. The term $D/Dt = \partial/\partial t + \mathbf{v} \cdot \nabla$ is the total derivative operator. The symbol ∇ is the gradient operator and $\Delta (= \nabla \cdot \nabla)$ is the Laplacian operator. Eq. (3) is a scalar equation of the temperature and Eq. (4) is an equation of the vapor and cloud density, where p is the pressure, ν is the viscosity coefficient, Γ_d is the dry adiabatic lapse rate, Q is the coefficient of latent heat, \mathbf{B} is the buoyancy vector, \mathbf{f} is the external force vector, and S_E is the heat source term.

3.2. Numerical Solution

This subsection explains a numerical method for solving Eqs. (1)-(4) as follows.

(1) External Force

The equation for updating velocity is expressed as follows.

$$\mathbf{v}^* = \mathbf{v} + \Delta t \mathbf{f}, \quad (5)$$

where \mathbf{v}^* is the velocity vector after being updated.

(2) Viscosity Effect

The viscosity effect causes diffusion of the velocity field. This is calculated from the following equation.

$$\mathbf{v}^* = \mathbf{v} + \nu \Delta t \nabla^2 \mathbf{v}, \quad (6)$$

where ν is the viscosity coefficient.

(3) Advection

We use the semi-Lagrangian scheme²² for the advection part that corresponds to the total derivative operator $D/Dt = \partial/\partial t + \mathbf{v} \cdot \nabla$. The velocity field advects the state variables (the velocity field itself \mathbf{v} , the vapor density w_{vap} , the cloud density w_{cl} , and the temperature E). A particle at point \mathbf{x} is traced back over a time Δt and the new state variables for point \mathbf{x} are the state variables that the particle had one time step before. In this simulation, we use a first order integration scheme, so the path traced back corresponds to $\Delta t \mathbf{v}$, and this path is a straight line.

(4) Pressure Effect

The pressure effect requires the concept of the conservation of mass, that is, the pressure term requires $\nabla \cdot \mathbf{v}$ to be 0 in the incompressible fluid. This is equivalent to computing the pressure by the following Poisson equation.

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \mathbf{v}. \quad (7)$$

Eq. (7) is solved by an iterative method. The velocity vector satisfies incompressibility by subtracting the gradient of the pressure from the velocity vector.

$$\mathbf{v}^* = \mathbf{v} - \Delta t \nabla p. \quad (8)$$

(5) Vorticity Confinement

In the advection part, a first order integration scheme is used. However, the simulations suffer from too much numerical dissipation, so small scale vortices vanish too rapidly. Vorticity confinement addresses this problem⁷. First the vorticity vector generating the small scale structure is computed.

$$\mathbf{w} = \nabla \times \mathbf{v}. \quad (9)$$

Next normalized vorticity location vectors that point from lower to higher vorticity concentrations are computed.

$$\mathbf{N} = \frac{\mathbf{k}}{|\mathbf{k}|}, \quad (\mathbf{k} = \nabla |\mathbf{w}|). \quad (10)$$

Then the magnitude and direction of the added force is computed as

$$\mathbf{f}_{con} = \varepsilon h (\mathbf{v} \times \mathbf{w}), \quad (11)$$

where ε is the parameter controlling the amount of small scale detail added back into the velocity vector, and h is the voxel size. \mathbf{f}_{con} is treated as a part of the external force \mathbf{f} of Eq. (1).

(6) Buoyancy

The ascending air current is generated by the buoyancy. The acceleration of this is expressed by the following equation.

$$\mathbf{B} = k_{buo} \frac{E - E_o}{E_o} \mathbf{z} - k_g w_{cl} \mathbf{z}. \quad (12)$$

This equation indicates that the difference between the temperatures E and E_o causes the buoyancy. The weight of the water droplets (i.e. cloud) is also taken into consideration. E_o is the ambient temperature, which is the temperature of the assumed atmosphere that satisfies statics, $\mathbf{z} = (0, 0, 1)$ points in the upward vertical direction, k_{buo} is the buoyancy coefficient and k_g is the gravity coefficient. E_o is a function of height. This is set to decrease in proportion to the height

from the bottom of the simulation space.

(7) Adiabatic Cooling

When air parcels rise, the temperature decreases due to the adiabatic cooling in proportion to the height that the air parcels rise by. This is expressed by the decrease in proportion to the vertical velocity.

$$E^* = E - \Gamma_d \Delta t v_z, \quad (13)$$

where Γ_d is the dry adiabatic lapse rate.

(8) Phase Transition

The cloud (water droplets) density are generated proportional to the difference between the vapor density and the saturation vapor density in each voxel. Then, the vapor density decreases and the temperature increases due to the latent heat. Oppositely, the cloud density decrease if the total weight of the vapor density and the cloud density fall below the saturation vapor density. These are expressed by the following equations.

$$w_{cl}^* = w_{cl} + \Delta t \alpha (w_{vap} - w_{max}), \quad (14)$$

$$w_{vap}^* = w_{vap} - \Delta t \alpha (w_{vap} - w_{max}), \quad (15)$$

$$E^* = E + Q \Delta t \alpha (w_{vap} - w_{max}), \quad (16)$$

where α is the phase transition rate. Q is the coefficient of latent heat. w_{max} is the saturation vapor density that is a function of the temperature and is given by the following equation.

$$w_{max} = \begin{cases} A \exp\left(\frac{-Q}{E+B} + C\right), & \text{if } > w_{vap} + w_{cl}, \\ w_{vap} + w_{cl}, & \text{otherwise,} \end{cases} \quad (17)$$

where A , B and C are parameters for determining the saturation vapor density.

4. Results

4.1. Conditions of Simulation

For the initial condition, the ambient temperature is specified so that it decreases in proportion to the height from the bottom of the simulation space. The vapor distribution is set to decrease exponentially from the bottom, where the vapor density is less than the saturation vapor density. These are constant in the horizontal direction. The temperature distribution is matched to the ambient temperature. A periodic boundary condition is set in the horizontal direction and

$\mathbf{v} = 0$ is set on the bottom and top of the simulation space. Fig. 2 shows the simulation of cumuliform clouds using this model. The user specifies the heat source, which gives the temperature. The heat source, which can be time-variable, is used as one of the boundary conditions. In the simulation, the ascending air current is due to the buoyancy developed as a result of the temperature specified by the user. The temperature and the vapor are carried upwards. Then the vapor coagulates, and clouds are generated. At the same time latent heat is liberated and this is a factor in creating subsequent buoyancy.

The hotter the heat source is, the stronger the air current generated by the buoyancy is. When the density of the initial vapor is large, a larger amount of cloud is generated. At the same time the latent heat is also liberated. The latent heat promotes subsequent cloud development. In addition, if the ambient temperature lapse rate becomes large, the temperature of the ascending air current becomes hardly lower than ambient temperature, so the cloud keeps developing. To control cloud development, it is important to adjust the heat source, the vapor distribution, and the ambient temperature.

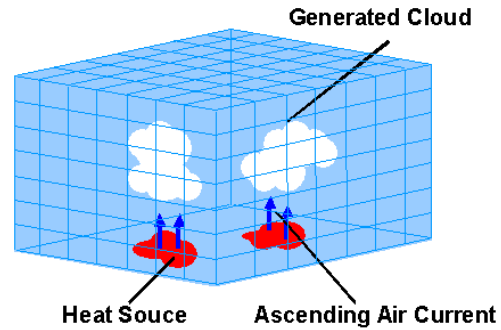


Figure 2: Simulation space

4.2. Example Images

Fig. 3 shows images generated by our method. Fig. 3 (a) shows examples of the cumulus development process in the daytime. Fig. 3 (b) shows cumulus development in the evening. Images at every 200 steps are shown. Fig.3 (c) shows the development process of cumulonimbus cloud. The tower-like cloud is developed by the strong ascending current. The images at every 100 steps are shown. To create these clouds in simulation, the voxel size corresponds to 20[m]. The number of voxels is $150 \times 120 \times 50$ for Figs. 3 (a) and (b). The calculation time for each time step of the simulation is about 5[s]. In the case of cumulonimbus, the number of voxels is $150 \times 120 \times 100$ for Fig. 3 (c). The calculation time for each time step is about 10[s]. The images are rendered by Dobashi's method². We used a HP Visualize (PentiumIII 1GHz) with fx10.

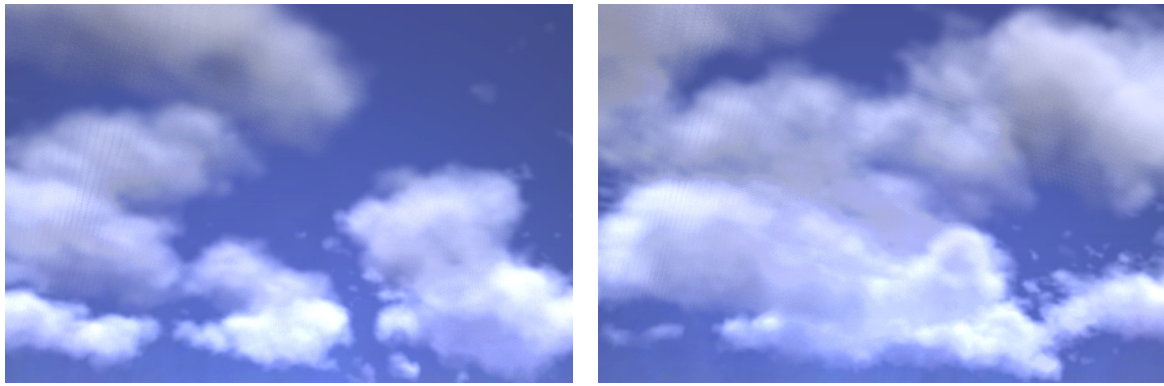
5. Conclusion and Future Work

In this paper we have proposed an atmospheric fluid model in which the interaction of the vapor, the cloud, the temperature and the velocity field are taken into consideration. This model includes the phase transition and adiabatic cooling not included in smoke simulations. Our model allows us to simulate more realistic cumuliform clouds than previous cloud models. Since the resulting clouds are obtained as a three-dimensional density distribution, realistic clouds can be rendered that take the light scattering due to cloud particles into account.

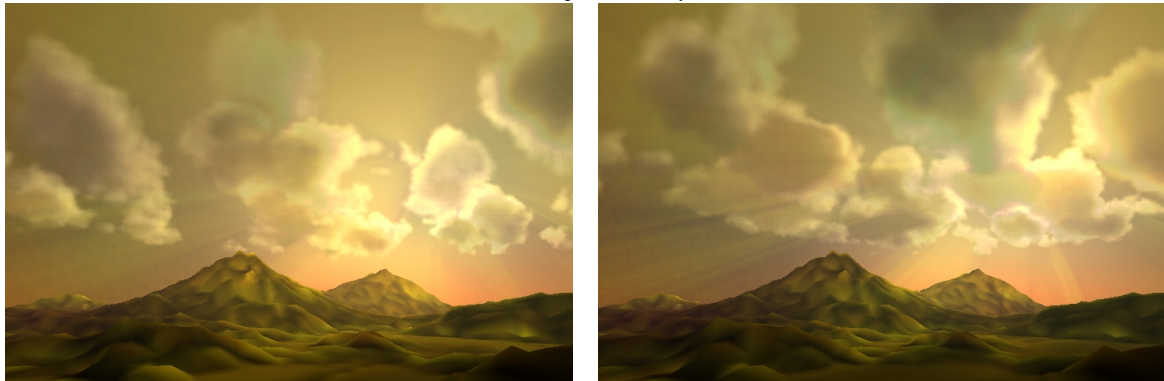
We are investigating variants of this model to achieve more realistic cloud dynamics and simulate other kinds of clouds. Moreover, we want to simulate clouds where interactions with geographical features take place, for instance, mountain.

References

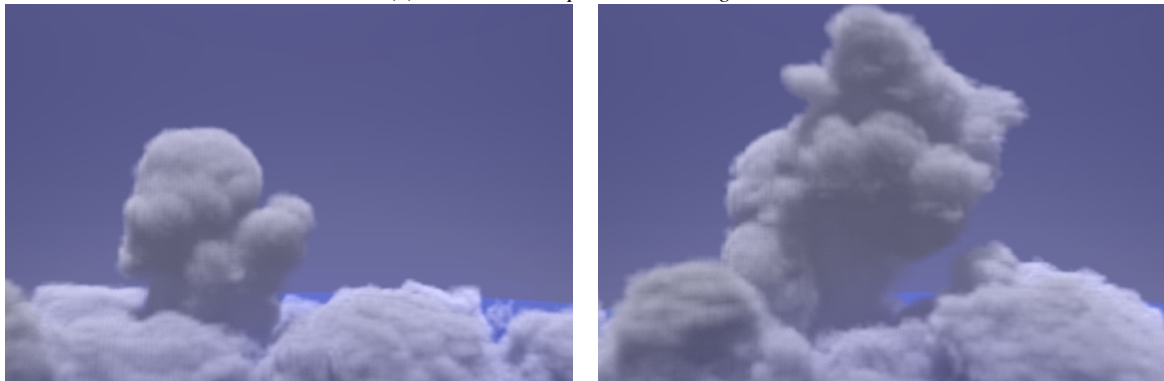
1. Y. Dobashi, T. Nishita, H. Yamashita, T. Okita, "Using Metaballs to Modeling and Animate Clouds from Satellite Images," *The Visual Computer*, Vol. 15, No. 9, 1998, pp. 471-482.
2. Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, T. Nishita, "A Simple, Efficient Method for Realistic Animation of Clouds," *Proc. SIGGRAPH'00*, 2000, pp. 19-28.
3. D. S. Ebert, R. E. Parent, "Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-Buffer Techniques," *Computer Graphics*, Vol. 24, No. 4, 1990, pp. 357-366.
4. D. S. Ebert, W. E. Carlson, R. E. Parent, "Solid Spaces and Inverse Particle Systems for Controlling the Animation of Gases and Fluids," *The Visual Computer*, 10, 1990, pp. 471-483.
5. D. S. Ebert, "Volumetric Modeling with Implicit Functions: A Cloud is Born," *Visual Proc. SIGGRAPH'97*, 1997, pp. 147.
6. D. S. Ebert, "Simulating Nature: From Theory to Application," *Course Note 26 of SIGGRAPH'99*, 1999, pp. 5.1-5.52.
7. R. Fedkiw, J. Stam, H. W. Jensen. "Visual Simulation of Smoke," *Proc. SIGGRAPH'01*, 2001, pp. 15-22.
8. N. Foster, D. Metaxas, "Modeling the Motion of a Hot, Turbulent Gas," *Proc. SIGGRAPH'97*, 1997, pp. 181-188.
9. G.Y. Gardner, "Visual Simulation of Clouds," *Computer Graphics*, Vol.19, No. 3, 1985, pp. 279-303.
10. R. A. Houze, "Cloud Dynamics" International Geophysics Series Vol.53, Academic Press, New York, 1993.
11. J. T. Kajiya, B. P. V. Herzen, "Ray Tracing Volume Densities," *Computer Graphics*, Vol. 18, No. 3, 1984, pp. 165-174.
12. T. Kikuchi, K. Muraoka, and N. Chiba, "Visual Simulation of Cumulonimbus Clouds," *The Journal of The Institute of Image Electronics and Electronics Engineers of Japan*, Vol. 27, No. 4, 1998, pp. 317-326 (in Japanese).
13. N. Max, "Light Diffusion through Clouds and Haze," *Graphics and Image Processing*, Vol. 13, No. 3, 1986, pp. 280-292.
14. N. Max, R. Crawfis, D. Williams, "Visualizing Wind Velocities by Advecting Cloud Textures," *Proc. Visualization'92*, 1992, pp. 179-183.
15. R. Miyazaki, S. Yoshida, Y. Dobashi, T. Nishita. "A Method for Modeling Clouds based on Atmospheric Fluid Dynamics," *PACIFIC GRAPHICS 2001 Conference Proceeding*, 2001, pp 363-372.
16. F. Neyret, "Qualitative Simulation of Convective Clouds Formation and Evolution," *EGCAS'97*, 1997, pp. 113-124.
17. T. Nishita, T. Sirai, K. Tadamura, E. Nakamae, "Display of The Earth Taking into Account Atmospheric Scattering," *Proc. SIGGRAPH'93*, 1993, pp. 175-182.
18. T. Nishita, Y. Dobashi, E. Nakamae, "Display of Clouds Taking into Account Multiple Anisotropic Scattering and Sky Light," *Proc. SIGGRAPH'96*, 1996, pp.379-386.
19. J. Stam, E. Fiume, "Turbulent Wind Fields for Gaseous Phenomena," *Proc. SIGGRAPH'93*, 1993, pp. 369-376.
20. J. Stam, "Stochastic Rendering of Density Fields," *Proc. Graphics Interface'94*, 1994, pp. 51-58.
21. J. Stam, E. Fiume, "Dipicting Fire and Other Gaseous Phenomena Using Diffusion Processes," *Proc. SIGGRAPH'95*, 1995, pp. 129-136.
22. J. Stam, "Stable Fluids," *Proc. SIGGRAPH'99*, 1999, pp. 121-128.
23. R. Voss, "Fourier Synthesis of Gaussian Fractals: 1/f noises, landscapes, and flakes," *SIGGRAPH'83: Tutorial on State of the Art Image Synthesis*, 10, 1983.
24. T. Yanagita and K. Kaneko, "Rayleigh-Benard Convection; Patters, Chaos, Spatiotemporal Chaos and Turbulent," *Physica D*, Vol. 82, 1995, pp. 288-313.
25. T. Yanagita and K. Kaneko, "Modeling and Characterization of Clouds Dynamics," *Phys. Rev. Lett.* Vol. 78, No. 22, 1997, pp. 4297-4300.



(a) Cumulus development in daytime



(b) Cumulus development in evening



(c) Cumulonimbus development

Figure 3: Examples