Texture Advection Based Simulation of Dynamic Cloud Scene

Shiguang Liu[1], Ruoguan Huang[2], Zhangye Wang[2], Qunsheng Peng[2], Jiawan Zhang[1], Jizhou Sun[1]
1 School of Computer Science and Technology, Tianjin University, Tianjin 300072, P.R. China
2 State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310027, P.R. China
E-mail: lsg@tju.edu.cn

Abstract

Fast display of realistic dynamic cloud scene is a challenging task for researchers in Computer Graphics. This paper describes a novel method of simulating animations of cloud. We first model the motion of cloud based on the physical theory of cloud formation. Then, the physical model of cloud is solved in discrete sparse grids. Thus, we can get approximate motion of cloud. Next, we advect texture image using the calculated velocity field to add the local detail of dynamic cloud scene. Here the texture image we used is Perlin noise map. The color of arbitrary point in the simulation space is got by calculating the product of the density texture and the regenerated Perlin noise texture. Finally, by changing different noise maps, realistic dynamic cloud scenes in different light environment are generated at high rendering rates.

1. Introduction

Cloud is an essential part for virtual outdoor scenes. Because of this, simulation and animation of cloud has been an active area of research in Computer Graphics for the past twenty years. It can be found wide applications in many domains such as special effects in movie, computer games, flight simulations, etc. Therefore, a lot of methods have been developed to display clouds [2, 3, 4, 5, 11, 12, 15, 16]. Realistic images can be generated by these methods, however, their main purpose is to create static cloud scene. Dynamic clouds with changing shape and color are often included in movies and computer games. Until now, rendering of dynamic cloud scene remains a complicated and tedious task [1, 6, 7, 8, 13, 14, 17, 23].

Realistic visual animation of clouds requires simulation of two aspects of clouds: cloud dynamics and cloud appearance. Cloud dynamics includes the motion of cloud, the phase transition of air and water vapor, etc. Cloud appearance is the color and shading of cloud which is caused by the interaction of light with water droplets in cloud. For simulation of cloud dynamics, some heuristic methods such as cellular automata [1], coupled map lattices [2], etc. are presented. These methods can generate motion of cloud at high speed, but the rendering result is less realistic. To generate realistic motion of cloud, physically based methods are necessary, however, they are time consuming. For simulation of cloud appearance, light propagation in cloud is often considered. The calculation of multiple scattering of cloud is very complicated and these methods are not suitable for simulation of fast display of cloud animations. This paper proposes a new method for fast display of realistic dynamic cloud scene. Our aim is to develop an approach that can create realistic cloud animation as quickly as possible. Our method has the following features.

- We create realistic dynamic cloud scene with a small amount of computation. It is due to that the shape and motion of cloud is simulated based on physics and the cloud model is solved only in sparse discrete grids.
- Compared with previous methods, we do not need to calculate the complex light propagation in cloud, so our rendering speed is much higher.
- Our system is easy to implement in common PCs. By inputting some initial parameters, cloud can be moved automatically. Furthermore, by changing different textures, we can easily simulate cloud scene at different time.

2. Related works

There are two categories of modeling cloud which are heuristic methods and physics based methods. The heuristic methods include particle systems, voxel volumes, procedural noise and textured solids. Reeves introduced particle systems as an approach to modeling clouds and other fuzzy phenomena [3]. Harris modeled

978 -1-4244-1579-3/07/$25.00 © 2007 IEEE

static clouds with particles and rendered each particle as a small, textured sprite [4]. Dobashi et al. used metaballs to model clouds extracted from satellite images [6]. In [7], clouds simulated on a voxel grid were converted to metaballs for rendering with the method of splatting. Ebert has done much work in modeling cloud using procedural noise [8-9]. Schpok et al. extended Ebert's techniques to take advantage of graphics hardware for fast rendering [10]. Gardner used fractal texturing on the surface of ellipsoids to simulate the appearance of clouds [15]. Voss generated cloud by method of fractal geometry [16]. However, These methods focus on simulation of static cloud scene. Neyret used an animated particle system to model behavior of convective clouds [9]. Dobashi et al.used a simple cellular automata model of cloud formation to animate cloud offline [7]. Miyazaki et al. used a coupled map lattice rather than a cellular automaton to simulate the animation of cloud [17].

Kajiya et al. generated cloud data for their ray tracing algorithm by solving a very simple set of PDEs (Partial Differential Equations) [18]. Overby et al. described a similar but slightly more detailed physical model based on PDEs [19]. They used the stable fluid simulation algorithm of [20] to solve the Navier-Stokes equations. Miyazaki et al. simulated animation of cloud based on computational fluid dynamics [21]. Their method is only suitable for cumuliform clouds. Mizuno et al. used a two fluids model to simulate volcanic clouds [22]. Harris et al. have simulated dynamic cloud scene based on physics. All their algorithms are implemented in graphics hardware. But they render the scene offline [23].

Nishita et al. introduced approximations and a rendering technique for global illumination of clouds, accounting for multiple anisotropic scattering and skylight [5]. Dobashi et al. enhanced the realism by computing the shadowing of the terrain by the clouds and shafts of light between the clouds [6]. Harris et al. extended the method of [6] to computing the multiple forward scattering in cloud [4, 23]. Bouthors et al. proposed an approach for capturing the important effects of multiple anisotropic Mie scattering within cloud layer, and the inter-reflections between the ground and the cloud base under sun and sky illumination [24].

3. Model of cloud behavior

In this section, we give the physical model of cloud dynamics, the numerical solution method, as well as the control method of cloud shapes.

3.1. Basic equations

We assume the density of the air is constant. The motion of air is expressed by Navier-Stokes equations:

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

$$\nabla\cdot\vec{u} = 0\,, \qquad (2)$$

Eq. (1) is a vector equation of the velocity field. Eq. (2) is the continuity equation, meaning that it expresses the conservation of mass. $\vec{u}$ is the velocity, ρ is density, υ is viscosity coefficient, and $\vec{f}$ denotes external force. To simulate realistic animation of cloud, the change of temperature field as well as the phase transition between air vapor and water droplets should be considered. So we introduced the following two equations:

$$\frac{\partial T}{\partial t} = \Gamma_d + Q\frac{\partial d_{cl}}{\partial t} + S_T\,, \qquad (3)$$

$$\frac{\partial d_{vap}}{\partial t} = -\frac{\partial d_{cl}}{\partial t}\,, \qquad (4)$$

Eq. (3) is a scalar equation of the temperature which expresses the change of temperature due to the heat source, dry adiabatic process and phase transition. Eq. (4) is an equation of the vapor and cloud density that denotes the phase transition between the vapor and cloud. T is the temperature field, Γ_d is the dryadiabatic lapse rate, Q is the coefficient of latent heat, S_T is the heat source term, d_{vap} and d_{cl} denote the density of the vapor and cloud, respectively.

3.2. Numerical solution

The simulation space is shown in Fig. 1. We divided it into uniform voxels. At time t, velocity vector, the vapor density, the cloud density and the temperature are assigned to each voxel as state variables. Then, each state variable is updated at every time step. The voxel width is h and the time interval is Δt. This subsection explains a numerical method for solving the cloud model (Eqs. (1)-(4)).

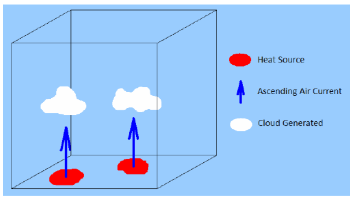


Fig. 1. Our simulation space

(1) Initialization of temperature field

We suppose that the temperature on the surface of the earth is 300K, and temperature will be 0.6 degree lower when the elevation is increased by 100 meters. The initial temperature field of our simulation space is defined as follows:

$$T_h = 300 - 0.6 \times h / 100, \quad (5)$$

(2) External forces

The external forces include the buoyancy, the vorticity confinement, gravity of water droplets and wind force. The last two terms are constants.

(3) Advection and diffusion

We use the semi-Lagrangian scheme [20] for the advection part (the first term at the right hand of Eq.(1)) to compute an intermediate velocity $\vec{u}'$. The velocity field also advects other state variables such as the vapor density, the cloud density and the temperature.

The second term at the right hand of Eq.(1) is the diffusion part which causes the diffusion of velocity field. We calculated it using the implicit method like [20].

(4) Project

In project process, the effects of the pressure as the third part of the right hand of Eq. (1) and the continuity function (Eq.(2)) are calculated. This is equivalent to computing the following Possion equation:

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \vec{u}, \quad (6)$$

We solve Eq. (10) by Gauss Seidel iterative method. Then the gradient of the pressure is subtracted from the velocity vector to satisfy the incompressibility.

(5) Thermodynamics and Phase Transition

When vapor rise at a fixed velocity, the temperature decreased in propagation to the vertical velocity $\vec{u}_z$. This process is so-called adiabatic cooling which can be expressed as follows:

$$T' = T - \Gamma_d \cdot \Delta t \cdot \vec{u}_z, \quad (7)$$

Cloud water continuously changes phases from liquid to vapor and vice versa. We express this process as follows:

$$d'_{cl} = d_{cl} + \Delta t \cdot \alpha \cdot (d_{vap} - d_{\max}), \quad (8)$$

$$d'_{vap} = d_{vap} - \Delta t \cdot \alpha \cdot (d_{vap} - d_{\max}), \quad (9)$$

$$T' = T + Q \cdot \Delta t \cdot \alpha \cdot (d_{vap} - d_{\max}), \quad (10)$$

where α is the phase transition rate, Q is the coefficient of latent heat, $d_{\max}$ is the saturation vapor density which can be expressed as follows:

$$d_{\max} = h_l \cdot A \cdot e^{B - \frac{Q}{T - C}}, \quad (11)$$

here h_l is a controlling coefficient for saturated vapor, A, B, Q and C are all empirical coefficients, the values we used are A=217.0, B=19.482, Q=4303.4, C=29.5.

3.3. Control of cloud shape

The shape of cloud can be controlled by the following methods:

(1) Before calculation, initial conditions of velocity, density, etc. are set. Different initial conditions will affect the shape of cloud.

(2) Heat source is the essential motivity of cloud formation. By changing the distribution of heat source, we can create different cloud shapes.

(3) Gravity and buoyancy affect the upward and downward motion of cloud. The coefficient of buoyancy can change the vertical velocity of air, thus affect the shape of cloud.

(4) Adiabatic cooling expressed the relation between velocity field and temperature field. The adiabatic lapse rate may affect the position and range of cloud. Coefficients related to phase transition in Eqs. (8)-(10) also affect the shape of cloud.

4. Texture advection

In this section, we will discuss the method of texture advection.

4.1. Perlin noise texture

Perlin noise function is proposed by Perlin in 1985 [26]. Perlin noise texture is easy to generate, and can express the local detail of cloud well. So we choose Perlin noise texture to express the appearance of cloud.

4.2. Texture coordinate advection

The core technique of texture advection is texture coordinate advection by the velocity field. Each voxel has its texture properties at time t , our aim is to update it after time step Δt . The basic idea can be summarized as follows: a particle at point x is traced back over a time step Δt and the new texture properties for point x are the texture properties that the particle had one tie step before. So the texture properties x at time $t + \Delta t$ are advected from position $p(x,t)$. Hence the equation for updating the texture properties are expressed as follows:

$$Tex(x, t+\Delta t) = Tex(p(x,t)), \qquad (12)$$

here Tex denotes texture properties.

4.3. Advection quality control

To minimize the distortion of texture advection, we set a period l t for the process of advection. After one period, we set the final advected texture as initial condition of next advection cycle. Period l t will affect the quality of advection. If the velocity is relatively small, the value of advection period should be big. On the contrary, if the velocity is relatively big, we should choose a small value for period. The main step of texture advection can be expressed as follows:

(1) Calculate the advection of texture coordinate
(2) Save the final texture of one advection period and replace original texture
(3) After one period, initialize the texture coordinate again and go to step (1).

Fig. 2 shows a result of 2D texture advection. We generate the velocity field by adding a force in the middle of the simulation space.

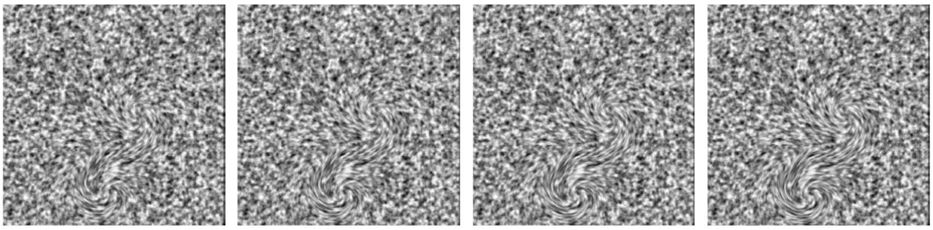

(a) **(b)** **(c)** **(d)**

Fig. 2. Result of 2D texture advection

5. Implementation

We first divide the simulation domain into sparse grids (16 × 16 × 16 or 8 × 8 × 8). After initializing every variables, we solve the physical model of cloud in the sparse grids to generate approximate motion of cloud. Density, pressure, temperature, texture coordinate and other variable are also advected in this process. Then we set the multiplication of the density texture by the noise texture as color of each point in the simulation space (See Fig. 3). Finally, animations of cloud can be displayed using the method of volume rendering. The physical model is solved in a sparse grid and complex light propagation does not needed. So the rendering speed is much faster than previous physically based methods for dynamic cloud simulation such as [17, 21, 22, 23].

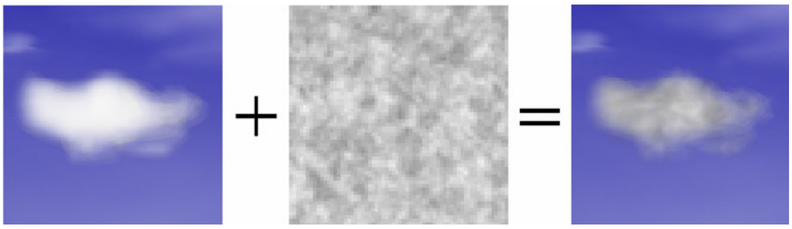


Fig. 3. Result of density texture combing with noise texture

6. Rendering result

With the proposed methods, we successfully generated various types of dynamic cloud scenes realistically on a PC with 2.4GHZ, Pentium IV processor, 1GB memory and an ATI 9800 graphics card at State Key Laboratory of CAD&CG, Zhejiang University. The resolution of simulation space is 16 × 16 × 16 or 8 × 8 × 8 and the resolution of Perlin noise texture is 64 × 64 or 32× 32.

Fig. 4 shows the contrast between results of texture advection and without texture advection. The left one is the rendering result without texture advection and the right one is the rendering result with texture advection. We can see that texture advection adds more local details to the appearance of cloud. Fig. 5 shows rendering results with different noise maps. By changing noise textures, we can simulate cloud scenes at different time. Fig. 6 is a sequence of animation of cloud. Fig. 7 shows dynamic cloud scene at dusk. We test our algorithm by different resolution of simulation and noise textures (See table 1). From the table, we can see that when resolution of simulation space is 16× 16 × 16 and the resolution of noise texture is 64× 64, the rendering rate is about 3fps. When the resolution decreases, the rendering rates will be much higher, however, the rendering quality will be lower, which can be used to simulate cloud scenes far from viewpoint.

Table 1. Rendering rates of different resolution of simulation space and noise texture

Resolution of simulation space	Resolution of noise texture	Rendering rates(fps)
16×16×16	64×64	3.0
	32×32	15.0
	16×16	25.0
8×8×8	64×64	11.0
	32×32	60.0
	16×16	75.0

7. Discussion and future work

We present a novel approach to simulating dynamic cloud scene. We first model the motion of cloud by solving physical equations on sparse discrete grids. Then, Perlin noise texture is advected by the velocity field to add local details of cloud. By changing different noise maps, realistic cloud animations at different time are rendered.

Compared with previous works, we can create dynamic cloud scene at relatively higher rendering rates. However, our method also has some disadvantages. The illumination effects are only visually realistic, and we can not generate the shadow and light shafts in cloud. Furthermore, our method can only simulate cloud scenes of landscape scale. In future work, we will exploit graphics hardware to improve our algorithm to achieve real time rendering. We believe that our method has the potential to rendering large scale natural scenes, which is also one of our research goals.

Acknowledgement

This research was supported in part by 973 Program of China under Grant No. 2002CB312101 and Natural Science Foundation of China under Grant No. 60475013.

References

[1] Y. Dobashi, T. Nishita, T. Okita, "Animation of Clouds Using Cellular Automaton", In *Proceedings of Computer Graphics and Imaging'98*, 1998, pp. 251-256.

[2] Y. Dobashi, R. Miyazaki, S. Yoshida, T. Nishita, "Modeling of Clouds Using a Coupled Map Lattice", In *Proceedings of SIGGRAPH'01, Sketches & Applications*, 2001, pp. 229-237.

[3] W. Reeves, "Particle System-A Technique Modeling a Class of Fuzzy Objects", *ACM Transaction on Graphics*, Vol.2(2), 1983, pp. 359-376.

[4] M.J. Harris, A. Lastra, "Real-Time Cloud Rendering", *Computer Graphics Forum*, Vol. 20(3), 2001, pp. 76-84.

[5] T. Nishita, Y. Dobashi, E. Nakamae, "Display of Clouds Taking into Account Multiple Anisotropic Scattering and Sky Light", In *Proceedings of SIGGRAPH'96*, 1996, pp. 379-386.

[6] Y. Dobashi, T. Nishita, Y. Yamashita, T. Okita, "Using Metaballs to Modeling and Animating Clouds from Satellite Images", *The Visual Computer,* Vol. 15(9), 1998, pp. 471-482.

[7] Y. Dobashi, T. Nishita, H. Yamashita, T. Okita, "A Simple, Efficient Method for Realistic Animation of Clouds", In *Proceedings of SIGGRAPH'00*, 2000, pp. 18-28.

[8] 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(4), 1990, pp. 357-366.

[9] D.S. Ebert, "Volumetric Modeling with Implicit Functions: A Cloud is Born", In *Proceedings of SIGGRAPH'97*, 1997, pp. 147-155.

[10] J. Schpok, J. Simons, D.S. Ebert, C. Hansen, "A Real Time Cloud Modeling, Rendering and Animation System", In *Proceedings of Eurographics/SIGGRAPH Symposium on Computer Animation 2003*, 2003, pp. 160-166.

[11] N. Max, "Light Diffusion Through Clouds and Haze", *Graphics and Image Processing*, Vol. 13(3), 1986, pp. 280-292.

[12] N. Max, "Efficient Light Propagation for Multiple Anisotropic Volume Scattering", In *Proceedings of the Fifth Eurographics Workshop on Rendering*, 1994, pp. 87-104.

[13] H.S. Liao, T.C. Ho, J.H. Chung, C.C. Lin, "Fast Rendering of Dynamic Clouds", *Computer Graphics*, Vol. 29(1), 2005, pp. 29-40.

[14] N. Wang, "Realistic and Fast Cloud Rendering Cloud Rendering", *Journal of Graphics Tools*, Vol. 9(3), 2004, pp. 21-40.

[15] G.Y. Gardner, "Visual Simulation of Clouds", *omputer Graphics*, Vol. 19(3), 1985, pp. 279-303.

[16] R. Voss, "Fourier Synthesis of Gaussian Fractals: 1/f

Noises, Landscapes, and Flakes", In *Proceedings of SIGGRAPH'83*, 1983.

[17] R. Miyazaki, S. Yoshida, Y. Dobashi, T. Nishita, "A Method for Modeling Clouds Based Atmospheric Fluid Dynamics", In *Proceedings of Pacific Graphics 2001*, 2001, pp. 363-372.

[18] J.T. Kajiya, B.P. Herzen, "Ray Tracing Volume Densities", *Computer Graphics*, Vol. 18(3), 1984, pp. 165-174.

[19] D. Overby, Z. Melek, J. Keyser, "Interactive Physically-Based Cloud Simulation", In *Proceedings of Pacific Graphics 2002*, 2002, pp. 469-470.

[20] J. Stam, "Stable Fluids", In *Proceedings of SIGGRAPH'99*, 1999, pp. 121-128.

[21] R. Miyazaki, Y. Dobashi, T. Nishita, "Simulation of Cumuliform Clouds Based on Computational Fluid Dynamics", In *Proceedings of Eurographics 2002 short presentation*, 2002, pp. 405-410.

[22] R. Mizuono, Y. Dobashi, B.Y. Chen, T. Nishita, "Physics Motivated Modeling of Volcanic Clouds As a Two Fluid Model", In *Proceedings of Pacific Graphics'03*, 2003, pp. 434-439.

[23] M.J. Harris, W.V. Baxter, T. Scheuerman, A. Lastra, "Simulation of Cloud Dynamics on Graphics Hardware", In *Proceedings of Graphics Hardware'03*, 2003, pp. 92-101.

[24] A. Bouthors, F. Neyret, S. Lefebvre, "Real-Time Realistic Illumination and Shading of Stratiform Clouds", In *Proceedings of Eurographics Workshop on Natural Phenomena 2006*, 2006.

[25] R. Fedkiw, J. Stam, H.W. Jesen, "Visual Simulation of Smoke", In *Proceedings of SIGGRPH'01*, 2001, pp. 15-22.

[26] K. Perlin, "An Image Synthesizer", In *Proceedings of SIGGRAPH'85*, 1985, pp. 287-296.

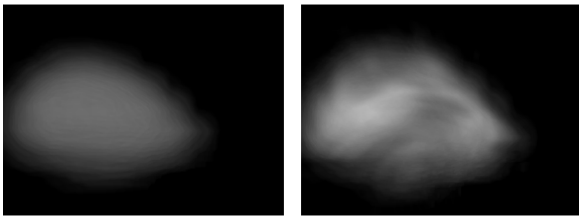


Fig. 4. Contrast between results of texture advection and without texture advection

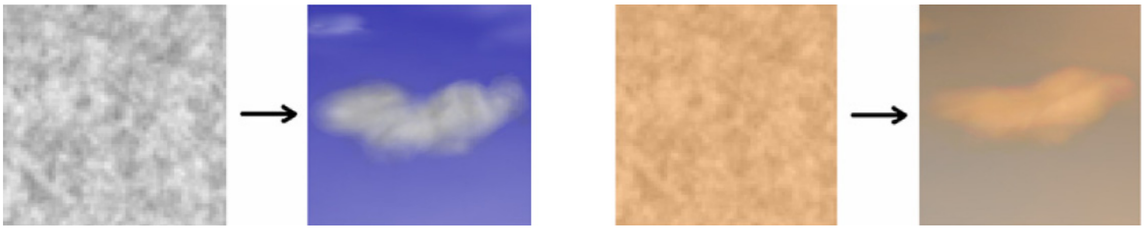
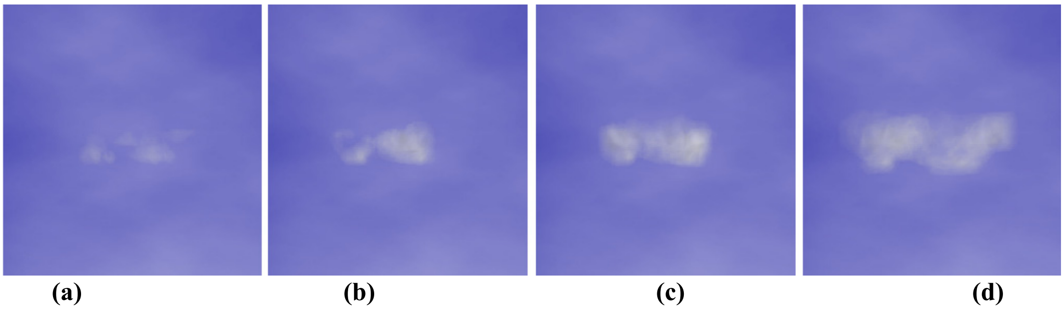
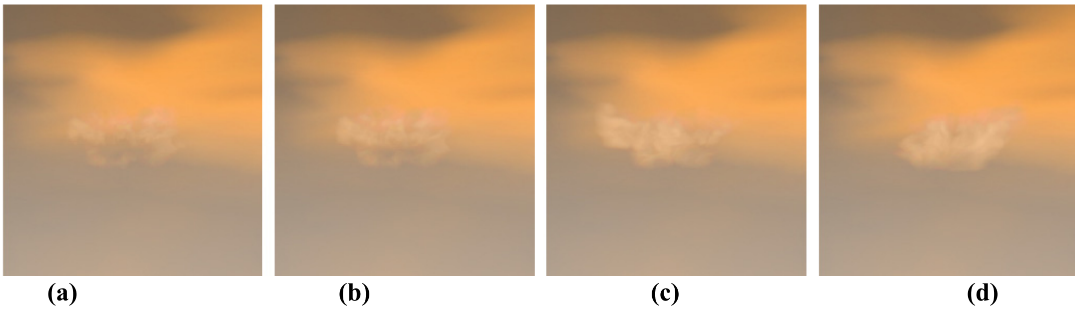


Fig. 5. Results of different texture advection



(a) **(b)** **(c)** **(d)**

Fig. 6. Animation sequence of cloud scene



(a) **(b)** **(c)** **(d)**

Fig. 7. Dynamic cloud scene at sunset