International Journal of Hybrid Information Technology
Vol.6, No.5 (2013), pp.171-182
http://dx.doi.org/10.14257/ijhit.2013.6.5.16

Real-time Simulation of Dynamic Clouds Based On Cellular Automata

Wang Hailong[1] and Meng Fanjun[2]

[1]*College of Network Technology, Inner Mongolia Normal University, Hohhot, China*
[2]*Computer and Information Engineering College,*
Inner Mongolia Normal University, Hohhot, China

lzjtuwhl@163.com

Abstract

There are plenty of applications about real-time simulation of realistic clouds. In this paper, the model of the clouds is established based on cellular automata and the method of how to deal with boundary grid points is developed. The efficiency is enhanced and some of the dynamic aspects of clouds are described by using promoted transition rules and introduced ascending air current. Multiple forward scattering model is used in illumination calculation, and Henyey-Greenstein phase function is embedded into forward scattering. The simulation results show that real-time rendering of the clouds is implemented realistically.

Keywords: *Dynamic Clouds; Cellular Automata; Multiple Forward Scattering; Real-time Rendering*

1. Introduction

The sense of reality of cloud is an important component of outdoor scene simulation, which is widely used in flight training, 3D game, and film and television generation. However, since the irregularity of the cloud's appearance, the cloud's appearance can not be described by math function. In addition, great variances of cloud also limit the difficulty of simulation. Cloud is a kind of complex natural phenomenon, influenced by many variables including micro factors and macro factors. Dynamic features of cloud including generation, growth, movement, and disappear are controlled by physical factors including fluid dynamics and thermodynamics. In addition, the illumination of the cloud is also extremely complex, consisted of scattering from sunshine, interreflection among the cloud particle, and light from the ground and the sky. Due to these factors, the real-time simulation and drawing of cloud becomes a hot issue and a difficulty in the study of computer graphics. In recent 20 year, numerous domestic and foreign scholars engage in the simulation study of cloud, in which a lot of modeling and real-time drawing algorithms are proposed to promote the development of natural environment simulation and computer graphics.

Seen from the modeling of cloud, current cloud simulation methods can be divided into 2 types: inspiration methods and physical methods. Inspiration methods mainly include: Particle Systems [1, 2], Voxel Volumes, fractal theory, Procedural Noise [3, 4, 5], and Textured Solids. In the Voxel Volumes, some inspiration rules were applied by Neyret to simulate the dynamic effect of cirrocumulus cloud with convection current characteristic. The method was relatively fast, but the simulation effect was worse than physical methods [6]. A cellular automata was proposed by Nagel as early as 1992 to produce the cloud [7]. The model was very good for the generation of simulation cloud. In 1998, the method was

ISSN: 1738-9968 IJHIT

improved by Dobashi *et al.* In addition, dynamic effect of relatively realistic cloud was concluded through some simple calculations [8]. In the literatures [9, 10, 11], the cloud was also generated by cellular automata. In the literatures [12, 13, 14], a Coupled Map Lattice generated from cellular automata was used to produce dynamic cloud. In physical methods, Partial Differential Equations (PDEs) was used to provide ray tracing algorithm with required cloud data by Kajiya *et al.*, [15]. A more detailed method (PDE physical modeling) was described by Overby, in which fixed fluid simulation algorithm was used to solve the Navier-Stokes equation [16]. The animation of cloud was generated based on fluid dynamics by Miyazaki, but the method was only suitable for the simulation of cumulus cloud [17]. 2 fluid models were used to simulate volcanic cloud by Mizuno [18]. The scene of dynamic cloud was also simulated based on physical method by Harris [19]. In the above methods, the inspiration methods have fast simulation speed, but have sense of reality second to physical methods. However, physical methods have a great amount of calculation, which should be supported by high-performance GPU, resulting in great limitation to real-time application. Since the calculation speed of PDE is slow, the scene drawing is mostly realized offline [19]. In current methods for the simulation of cloud shape, most simulated shapes are only the cumulus cloud. Individual methods draw several clouds, but the implementation platform cannot be provided for common usage. In general, the methods mentioned above cannot draw clouds of any shapes in the same system.

The chapters in the paper are as follows: the basic thought of the cellular automata is introduced in the second chapter and the improving method is also proposed based on this; illustration algorithm used in the illustration for sense of reality is introduced in the third chapter, and the multiple forward scattering illustration model and improved phase function are introduced in detail. The fourth chapter is the simulation result and analysis. In the end, the simulation study of cloud is summarized and the outlook is proposed.

2 Improved Cellular Automata

2.1 Basic Thought of Cellular Automata

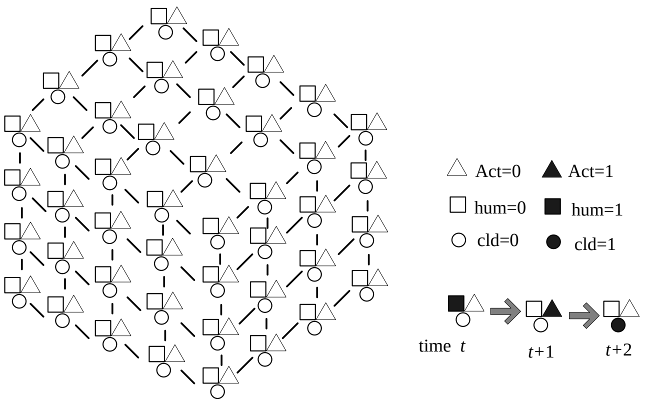
By means of Nagel and Dobashi's modeling method, according to basic thought of CA, the simulation space is subdivided into three-dimensional grids. 3 state variables including hum, cld and act are set on every grid to represent the water vapor, cloud, and the state transformation. The value of every variable is either 0 or 1. The cloud simulation is realized through simple state transformation rule, as shown in Figure 1(a). The method is easily carried out with little calculation amount. The cloud disappear process not realized in the paper of Nagel *et al.*, is solved by Dobashi et al. In addition, a smooth transition is added in Cloud (cld=1) and no cloud (cld=0).

The state of Grid (I, j, k) at time t+1 is generated from the state at time t by the following transformation rules:

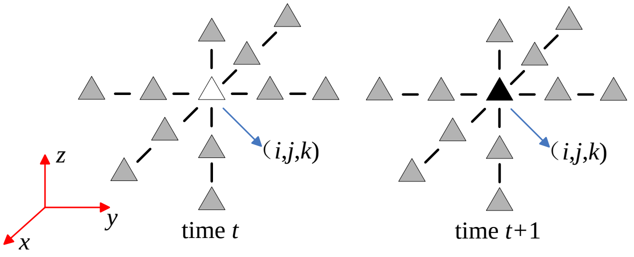
$$act(i,j,k,t+1) = \neg act(i,j,k,t) \wedge hum(i,j,k,t) \wedge f_{act}(\square) \tag{1}$$

$$cld(i,j,k,t+1) = cld(i,j,k,t) \vee act(i,j,k,t) \tag{2}$$

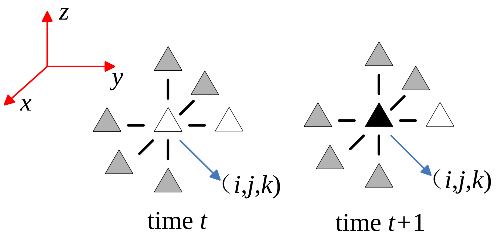
$$hum(i,j,k,t+1) = hum(i,j,k,t) \wedge \neg act(i,j,k,t) \tag{3}$$



(a) Three-dimensional grid and state transformation



(b) State transformation of cloud generation simulated by variable act



(c) State transformation simulated by variable act under wind function

Figure 1. Cloud simulated by cellular automata

$f_{act}(\square)$ is Boolean function, determined by the state surrounding act. Figure 1(a) shows the above transformation rule. If hum and $f_{act}(\square)$ are 1 at time t+1, then act is 1. Meanwhile, cld changes into 1 at time t+2. $f_{act}(\square)$ is concluded from the following Equation (4):

$$\begin{aligned} f_{act}(\square) = {} & act(i+1,j,k,t) \vee act(i-1,j,k,t) \vee act(i,j+1,k,t) \vee act(i,j-1,k,t) \\ & \vee act(i,j,k+1,t) \vee act(i,j,k-1,t) \vee act(i+2,j,k,t) \vee act(i-2,j,k,t) \\ & \vee act(i,j+2,k,t) \vee act(i,j-2,k,t) \vee act(i,j,k-2,t) \end{aligned} \qquad \textbf{(4)}$$

As shown in Figure 1(b), if any act value is 1 in the grey grid surrounding the Grid (i, j, k), $f_{act}(\square)$ is 1. The effect under parallel wind function is realized by changing the transformation rule of $f_{act}(\square)$ in Literature [8].

$$\begin{aligned} f_{act}(\square) = {} & act(i+1, j,k,t) \vee act(i-1, j,k,t) \vee act(i, j,k+1,t) \\ & \vee act(i, j,k-1,t) \vee act(i, j-1,k,t) \end{aligned} \tag{5}$$

The Rule is furtherr simpliefied, and the result is:

$$\begin{aligned} f_{act}(\square) = {} & act(i+1, j,k,t) \vee act(i-1, j,k,t) \\ & \vee act(i, j,k-1,t) \vee act(i, j-1,k,t) \end{aligned} \tag{6}$$

The above simplification is made based on the following reasons: based on the parallel wind function, the cloud uprising process of cloud fully considered, similar to cross-ventilation mode, which is more like the realistic cloud generation process. Meanwhile, the calculation is easier. The simulation result shows the generated cloud is more realistic.

2.2 Simulation of dynamic features of cloud

Based on whether the cloud will disappear after its generation, an improvement is made in Literature [8] based on Literature [7]. This is because the variable cld will not change into 1 after being 1 in Equation (2). Thus, some complex changes of cloud cannot be realized, including cloud disappear. The improvement in Literature [8] is to add the variable ext for cloud disappear. Its transformation rule is:

$$ext(i, j,k,t+1) = \neg ext(i, j,k,t) \wedge cld(i, j,k,t) \wedge f_{ext}(\square) \tag{7}$$

$f_{act}(\square)$ is a function similar to $f_{act}(\square)$, which can be concluded based on the state of ext in surrounding grid. Then, the transformation rule of cld is further adjusted as:

$$cld(i, j,k,t+1) = \neg ext(i, j,k,t) \wedge (cld(i, j,k,t) \vee act(i, j,k,t)) \tag{8}$$

However, when the above rule is used, when ext is 1, cld will be 0. Thus, there will be unnatural phenomenon in frequent switch from cloud generation to cloud disappear. In order to avoid the problem, the time of disappear Text is added. When ext is 1, cld will be 0 after time Text. In this way, the cloud shape can be change naturally.

2.3 Setting of initial state and handling of border grid

In the setting of initial state, 0 or 1 is not determined by the comparison between the random number and given probability which used in Literatures [7, 8]. Instead, easier method is applied to generate 0 or 1 randomly in this paper. The reason is that most time should be spent to calculate the complex illustration.

At the beginning of the simulation, hum value is the random value between 0 and 1. act is also initialized as random value between 0 and 1. However, when hum is 0, act should not be 1. cld is 0.

The cloud generation is realized by updating the states of various variables in Equation (1) to Equation (3). In the simulation process, the random values follow respective probability distribution. The calculation of continuous density distribution function is subject to method in Literature [8].

The transformation rule of border grid is not mentioned in Literature [7, 8]. Since the calculation of $f_{act}(\square)$ and $f_{act}(\square)$的 will exceed the space of simulation in the state transformation of border grid, the method that the grid is deleted if not existed during the calculation of $f_{act}(\square)$ and $f_{act}(\square)$. Namely, the items on the right to be deleted should be determined before judge the value of i, j, k.

3. Illustration Algorithm in Real-time Drawing

3.1 Single scattering

The sense of reality of cloud cannot be separated from the illustration calculation. The scattering of cloud particle meets Mie Scattering rule. Scattering model mainly includes single illustration model and multiple scattering illustration model.

In 1982, the single illustration model for the interaction between the light and cloud particle was firstly proposed by Blinn. In this model, it is assumed that the unit volume cloud has n particles [20]. The cloud particle is the sphere with radius of 4. The density of the cloud is represented by D, a constant value. The shape of the clouds is parallel tetrahedron v. The particles are distributed randomly in the v. If the particle only has first-class scattering, and the feature of multiple scattering in the interaction between the particle and the light is neglected, as shown in Figure 2, N is the unit formal vector of cloud surface, L is the unit vector of incident ray. E is the unit formal vector of particle scattering direction. θ is the included angle of the incidental ray and the incident ray, which is named as phase angle. Under the light from the opposite direction of the observation point, θ is 0.

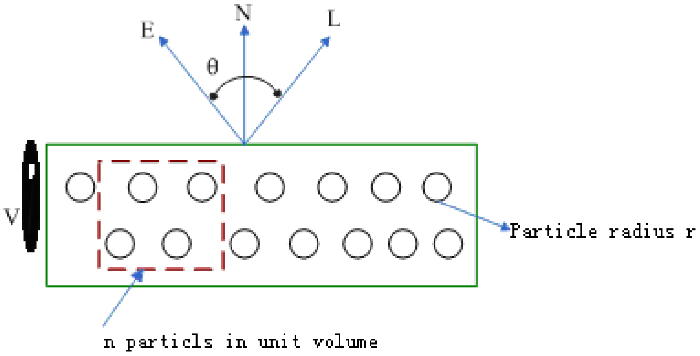


Figure 2. Single scattering

Based on above assumption, the physical process of single scattering between the light and the cloud particle is shown in Figure 3. Firstly, the incident ray reaches certain particle in the clouds with depth T' through the probability $P(0,V_1)=\exp(-nV_1)$: the particle redistributes the energy of incident ray. A part of energy is absorbed, and a part of energy is scattered in line with phase function $p(\phi)$. In the end, the scattered energy reaches the observation point direction through probability $P(0,V_2)=\exp(-nV_2)$ in the direction of the gazing direction. Through the integral of cloud thickness, the scattered effect of the whole clods can be achieved. In the end, the calculation equation of the light intensity is:

$$I=\frac{\omega p(\phi)\gamma\exp(-\gamma/\mu)}{\mu} \tag{9}$$

ω is the particle reflection coefficient; ϕ is the phase angle of the incident ray and reflection ray; γ is the optical density; μ is angle of incidence, and $p(\phi)$ is phase function. The phase function is simplified Rayleigh scattering, of which the forward scattering and the backward scattering have the same energy. Namely:

$$p(\phi) = 3/4(1+\cos^2\phi) \tag{10}$$

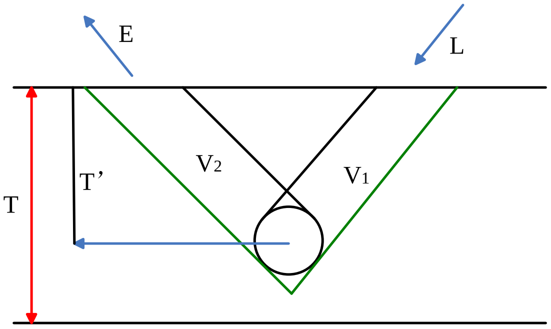


Figure 3. The physical process of single scattering

According to geometrical optics principle and atmosphere scattering feature, a simple method of calculating light intensity by formula is given. However, not all clouds can be simulated by this model. This model is only suitable for the clouds with plane level, and not suitable for the cumulus with irregular shape and high reflection rate.

Another single scattering illustration model is also proposed by Dobashi in Literature [21], in which the scattering light is consisted of the scattering light from the incident direction to the visual point and the scattering light from backward to the visual point. The single scattering model is due to the calculation of scattering light in every cloud particle, which is used for the simulation of various clouds, as shown in Figure 4.

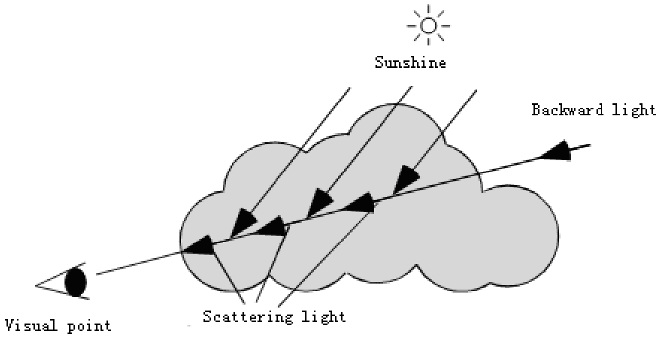


Figure 4. Cloud color calculated based in single scattering

The above single scattering models are simple for calculation, but only scattering feature from the incident ray to the observation point is considered, but the scattering feature of the particle is not considered, resulting in poor sense of reality. Since the cloud particle has very high reflection rat and strong forward scattering feature, the effect of multiple scattering cannot be neglected.

3.2 Multiple scattering

Multiple scattering illustration model is closer to physical feature of cloud than single scattering illustration and better describes the physical process between the cloud and the light. However, not only the incident ray should be calculated, but multiple scattering lights from various particles to various directions should be calculated, resulting in high time complexity. In addition, the scattering directions of various particles are irregular, which can hardly be simulated. Therefore, the multiple scattering functions should be simplified to find an efficient illustration model.

Harris improved the model and pointed out that the scattering intensity of the particle was mainly determined by forward scattering. Namely, the scattering energy can only be integrated into a small angle in the forward direction of the particle. The cloud illustration model of Harris is shown in Figure 5.

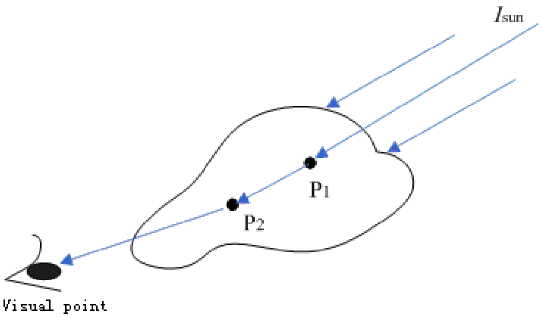


Figure 5. Multiple forward scattering

In this model, the light of arrival particle P is simplified. Only first-class scattering and secondary scattering are considered. The light intensity I (p,l) on the particle p is divided into 2 parts. The light from the incidence direction is consisted of the light not absorbed by the cloud surface and the light reflected from other directions on the particle p. Particle amount with similar radius in unit volume is used to simplify the scattering coefficient as the constant value. The light intensity equation of cloud particle P is:

$$I(P,\omega)=I_0(\omega)\exp(-\int_0^{Ap}\tau(t)dt)+\int_0^{Ap}g(s,\omega)\exp(-\int_s^{Ap}\tau(t)dt)ds \tag{21}$$

$I_0(\omega)$ is the light intensity out of the cloud in the direction ω of incident ra. Ap is the depth of particle P to the border of the cloud in direction ω. τ(t) is the disappear coefficient of particle t. $g(s,\omega)$ represents the light of location s scattered from all directions to direction ω.

$$g(s,\omega)=\int f(s,\omega,\omega')I(s,\omega')d\omega' \tag{32}$$

$f(s,\omega,\omega')$ is the distribution function of secondary direct scattering, which determines the ratio of light in location s from 了 ω’ to ω. $I(s,\omega')$ is the light intensity of location s in ω’ direction.

$$f(s,\omega,\omega')=\alpha(s)\cdot\tau(s)\cdot p(\omega,\omega') \qquad \textbf{(43)}$$

$\alpha(s)$ is the diffuse reflection coefficient of the medium in location s. $p(\omega,\omega')$ is the scattering phase function.

Based on the above hypothesis, only forward scattering of all directions is calculated. Thus ω=l, and ω'=-l. Only forward scattering from all directors with the angle γ is calculated. Then, the equation (12) is changed into:

$$g(s,l)=f(s,l,-l)I(s,-l)\cdot\lambda/4\pi \qquad \textbf{(54)}$$

During the calculation process, like Henyey-Greenstein phase function is introduced in the multiple forward scattering illustration model. The equation is as follows:

$$p_{HG}(\phi)=\frac{1}{4\pi}\frac{1-g^2}{(1-2g\cos\phi+g^2)^{3/2}} \qquad \textbf{(65)}$$

The symmetry factor g controls the anisotropy of scattering. When g is positive, most incident light is scattered forward. When g is negative, most incident light is scattered backward. When g is 0, the light is scattered with anisotropy. A very useful result of Mie Scattering is that when the particle is greater than the wavelength of the light, there will be a great anisotropy of scattering. Through the verification, a good effect is achieved when g is 0.3.

4 Simulation Result and Analysis

The improved method is used by the author to simulate the cloud. Operation Environment: Intel ® Core TM i5-2380P CPU 3.10GHz (Dual Core). RAM 4.00G. Graphics Card: NVIDIA GeForce GTX 550 Ti.

Figure 6 (a, b, c) gives the cloud generation process, which is influenced by wind.

(a) (b) (c)

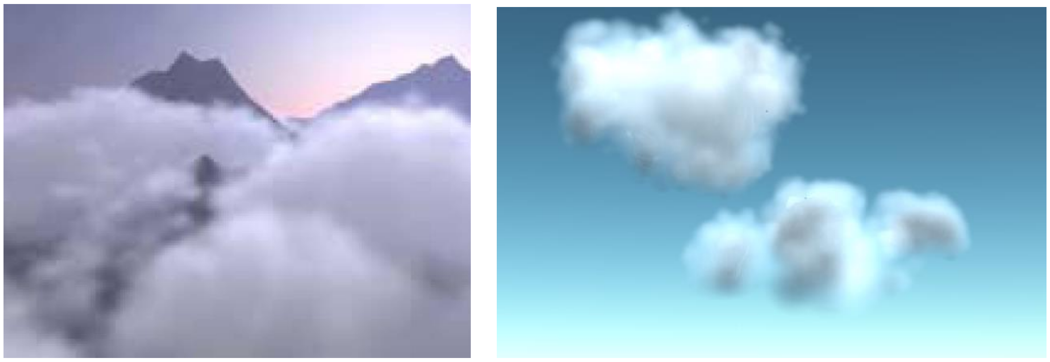
Figure 6. Cloud generation process

Figure 7 is the cloud generated from Rule (5) of Literature [8]. Figure 8 is the cloud generated from improved method. It can be found that the generated cloud is more realistic with the convective term.

Figure 7. Cloud generated from Rule (5)

Figure 8. Cloud generated from Rule (6)

Figure 9(a) is the cloud effect in the single scattering illustration model in Literature [19]. Figure (b) is the effect under multiple scattering. Figure 9(c) is the simulation effect of multiple forward scattering. It can be found that the effect of single scattering is relatively poor, and the effects of multiple scattering and multiple forward scattering can be hardly identified. Nevertheless, multiple forward scattering reduces the calculation complexity and increases the generation speed. Therefore, multiple forward scattering illustration model is more suitable for cloud drawing.



(a) cloud effect under single scattering (b) effect of multiple scattering

(c) effect of multiple forward scattering illustration

Figure 9. Cloud effect under different scattering models

Different volume data levels are used to verify the improved method. When voxel grid data size is 64*64*64, the drawing frame rate is about 75fps. When voxel grid data size is 128*128*128, the drawing frame rate is about 30fps. However, when the data size is

256*256*256, the drawing rate decreases to fewer than 10fps. Thus, in current software and hardware conditions, when the grid data is in the level of 1003, the real-time interactive drawing can be realized with relatively good sense of reality in appearance. Please refer to Table 1 for details.

Table 1. Drawing frame rate under different data sizes

Data size	Drawing frame rate (fps)	Sense of reality of the picture
64*64*64	70~80	Poor
128*128*128	About 30	Relatively Good
256*256*256	<10	Good

5 Next Study Work

The cloud model is established based on the cellular automata technology. In order to accelerate the operation speed and reflect the real feature of the cloud, the transformation rule among states is improved. During the drawing process, multiple forward scattering illustration model is used to enhance the sense of reality. In addition, Henyey-Greenstein phase function similar to Mie scattering is used in the forward scattering illustration model. By above methods, the real-time effect can be realized in cloud simulation.

Through the study of many years, the simulation technology for sense of reality of cloud is developed greatly. However, due to the complexity of the cloud and increasing application demand, the real-time drawing of cloud scene with sense of reality is still one of the most challenging issues in computer graphics, and has many problems to be further studied. These problems are also our research focus in the future. For example, many applications including flight simulation and game have requirement on sense of reality and real time for large-scale natural scene, the real-time drawing of cloud with sense of reality is still the key point and difficult point in future study. In current study results, the vivid rendering quality can be provided by some technologies, but the speed cannot meet the requirement of real-time interaction. Some strategies including simplifying calculation and GPU acceleration are applied to guarantee the speed requirements, but the rendering quality is influenced. How to find a reasonable compromise between the real time and sense of reality should be further studied. For another example, the cloud has complex dynamic features, so only simple movement processes including cloud generation and dissipation can be simulated by current methods. How to fast and vividly reflect the change of cloud under complex environment and the change from cloud to rain is also the direction of future study. Besides, the construction of common simulation platform suitable for various clouds is also very important to fields including flight simulation, weather simulation, and scientific visualization.

Acknowledgements

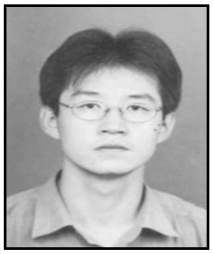
This research was supported by Scientific Research Project of Higher Education of Inner Mongolia Autonomous Region,China (NJZY13052).

References

[1] W. Reeves, “Particle System-A Technique Modeling a Class of Fuzzy Objects”, ACM Transaction on Graphics, vol. 2, no. 2, **(1983)**, pp. 359-376.
[2] J. van der Burg, “Building an Advanced Particle System”, [EB/OL] Gamasutra, (2000) June, http://www.gamasutra.com/features/20000623/vanderburg_01.htm.
[3] D. S. Ebert and 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, “Interactive cloud modeling and photorealistic atmospheric rendering”, Proc of SIGGRAPH, **(2004)**, pp. 238-256.
[5] Y. Qi and X. Shen, “Method of Cloud Drawing Based on Perlin Noise”, System Simulation Journal, vol. 14, no. 9, **(2002)**, pp. 1204-1207.
[6] F. Neyret, “Qualitative simulation of convective clouds formation and evolution”, Proceedings of the 8th Eurographics Workshop on Computer Animation and Simulation, **(1997)**, pp. 113-124.
[7] K. Nagel and E. Rasehke, “Self-organizing criticality in cloud formation?”, Physica A, **(1992)**, pp. 519-531.
[8] Y. Dobashi, T. Nishita and T. Okita, “Animation of Clouds Using Cellular Automata”, Proceedings of Computer Graphics and Imaging’98, **(1998)**, pp. 251-256.
[9] Q. Bo and L. V. Tao, “Real-time dynamic cloud modeling and rendering”, Computer Graphics, Imaging and Vision: New Trends, **(2005)**, pp. 285-290.
[10] J. Xu, C. Yang, J. Zhao and L. Wu, “Fast Modeling of Realistic Clouds”, Computer Network and Multimedia Technology, **(2009)**, pp. 1-4.
[11] E. M. Upchurch and S. K. Semwal, “Dynamic Cloud Simulation Using Cellular Automata and Texture Splatting”, Proceedings of the 2010 Summer Computer Simulation Conference, **(2010)**, pp. 270-277.
[12] T. Yanagita and K. Kaneko, “Coupled map lattice model for convection”, Physics Letters A, vol. 175, **(1993)**, pp. 415–420.
[13] R. Miyazaki, S. Yoshida, Y. Dobashi and T. Nishita, “A method for modeling clouds based on atmospheric fluid dynamics”, In Proceedings of Pacific Graphics 2001, **(2001)**, pp. 363–372.
[14] Y. Dobashi, T. Nishita, R. Miyazaki and S. Yoshida, “Modeling and Dynamics of Clouds Using a Coupled Map Lattice”, Proc. Siggraph 2001 Technical Sketches, Los Angeles (USA), **(2001)** August, pp. 229.
[15] J. T. Kajiya and B. P. Herzen, “Ray Tracing Volume Densities”, Computer Graphics, vol. 18, no. 3, **(1984)**, pp. 165-174.
[16] D. Overby, Z. Melek and J. Keyser, “Interactive Physically-Based Cloud Simulation”, Proceedings of Pacific Graphics 2002, **(2002)**, pp. 469-470.
[17] R. Miyazaki, Y. Dobashi and T. Nishita, “Simulation of Cumuliform Clouds Based on Computational Fluid Dynamics”, In Proceedings of Eurographics 2002 short presentation, **(2002)**, pp. 405-410.
[18] R. Mizuono, Y. Dobashi, B. Y. Chen and T. Nishita, “Physics Motivated Modeling of Volcanic Clouds As a Two Fluid Model”, Proceedings of Pacific Graphics’03, **(2003)**, pp. 434-439.
[19] M. J. Harris, “Real-time cloud simulation and rendering”, Chapel Hill: University of North Carolina, **(2003)**.
[20] J. F. Blinn, “Light Reflection Functions for Simulation of Clouds and Dusty Surfaces”, Computer Graphics (SIGGRAPH '82 Proceedings), **(1982)** August, pp. 21-29.
[21] Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita and T. Nishita, “A Simple, Efficient Method for Realistic Animation of Clouds”, Computer Graphics (SIGGRAPH 2000 Proceedings), **(2000)** July, pp. 19-28.

Authors



Hailong Wang

He received the BS in computer science from North Jiaotong University,China,in 1998,and received the MS in computer science from Lanzhou Jiaotong University,China,in 2007.Currently,he is an assistant professor in Computer&Information Engineering College at Inner Mongolia Normal University,China. His research interests include embeded system and muti-core porcessors,and also fault tolerance and real-time database.

Meng Fanjun received the BS and MS degrees in computer science from Inner Mongolia Normal University, China, in 1999 and 2007. Currently, his research interests include scheduling techniques and parallel algorithms for clusters, and also multi-core processors and software techniques for I/O-intensive applications.