Real-time Rendering of Dynamic Clouds

Xiao-Lei Fan[1a,1b] Li-Min Zhang[2] Bing-Qiang Zhang[3] Yuan Zhang[4]

[1a] Department of Electronics and Information Engineering, Naval Aeronautical and Astronautical University
Yantai 264001, Shandong, China
[1b] Noncommissioned Officers Vocational and Technical Education College, The Second Artillery Engineering University
Qingzhou 262500, Shandong, China
gm_fxl@163.com
[2] Department of Scientific Research, Naval Aeronautical and Astronautical University
Yantai 264001, Shandong, China
iamzlm@163.com
[3] Department of Electronics and Information Engineering, Naval Aeronautical and Astronautical University
Yantai 264001, Shandong, China
zbqwh1981@163.com
[4] No. 5 Department, Naval Aeronautical and Astronautical University
Yantai 264001, Shandong, China
zy_lwb@sina.com

Received 18 January 2014; Revised 10 June 2014; Accepted 13 July 2014

Abstract. This paper established the model of clouds based on cellular automata and developed the method of how to deal with boundary grid points. It also described some of the dynamic aspects of clouds by using promoted transition rules and introduced ascending air current. These techniques could reflect the true characteristics of clouds and enhance the run-time efficiency. Multiple forward scattering model was applied to cellular automata in the illumination calculation process, and the improved phase function was proposed. The simulation results showed that proposed methods could promote realism of clouds and simulation efficiency, and implement real-time rendering of dynamic clouds and large-scale clouds.

Keywords: real-time rendering, dynamic clouds, improved cellular automata, phase function

1 Introduction

Realistic simulation of clouds plays an import role in the simulation of outdoor scenes, which has lots of applications such as flight training, 3D games, and movie fields and so on. However, because of the irregular appearance of clouds, there are no mathematic functions which can describe it. In addition, the varieties of clouds evolution make the simulation more complicated.

Cloud is a complex natural phenomena influenced by a number of variables including macro and micro factors. Dynamic characteristics of cloud formation, growth, movement, extinction are controlled by many physical factors such as fluid dynamics, thermodynamics etc. Then the cloud illumination is also very complex, not only from the scattering of sunlight, but also the reflection among cloud particles as well as the lights from the ground and the sky. All the above factors make the real-time simulation and rendering of clouds hot and difficult in computer graphics field. Within twenty years, domestic and foreign researchers have developed a number of modeling and real-time rendering algorithms on cloud simulation, promoting the development of natural environment simulation, and even the computer graphics.

From the point of view of modeling, the current simulation methods of clouds can be classified into: heuristic methods and physics-based methods.

Heuristic methods include particle systems [1,2], voxel-based simulating methods, fractal methods, procedural noise [3, 4, 5], method of textured solids, and so on.

As for as the voxel-based method is concerned, Neyret used some heuristic transition rules to simulate convective and dynamic effects of cirrocumulus [6]. This method is relatively fast, but its simulation effects are not perfect compared with those physics-based simulation results. Nagel proposed Cellular Automata (CA) model to simulate the formation of clouds in 1992 [7]. Dobashi et al. improved this technique and used some boolean variables and transition rules to simulate cloud formation, extinction, as well as simple wind effect, and they realized more realistic dynamic effects of clouds through some simple calculations [8,9]. Although clouds simulation does not require elaborate interaction just like Surgery Simulation [10], the more realistic clouds can be rendered if the effects of the external forces such as the wind are considered.

Dynamic clouds are also generated using cellular automata method in [11,12,13]. Analogously, an approach called CML(Coupled Map Lattice) which sourced from CA was applied to generate clouds, which paid more attention to the physical properties of clouds in [14,15].

As for the physics-based method, a more detailed physical modeling technique based on PDEs (Partial Differential Equations) was introduced in [16], which used stable simulation algorithm of fluid flow for solving Navier-Stokes equations. Miyazaki et al. generated the animation of clouds according to fluid dynamics [17], but their method was only applied in cumulus simulation. Mizuno et al. used two fluid models to emulate volcanic clouds [18]. Harris also simulated dynamic clouds scene based on physical methods [19]. Large-scale clouds scene was animated in [20], but no dynamic interactivity was achieved.

Among the above methods, the realism of physics-based methods is better than heuristic ones, but such methods need large amount of calculations and advanced GPU mostly. Furthermore, most of the scene rendering should be achieved offline that causes the limitations of real-time applications. At the same time the stability problems of the solutions of PDEs exist. Stam proposed an unconditionally stable method [21], but there was numerical dispersing problem. Fedkiw et al. offered an improvement method [22], and the method in [23] can be applied to simulate fluid using large time step without distortion. All in all, the above improved methods still remain large calculations and many restrictions. While the CA approach can greatly improve the efficiency of cloud scene simulation because its transition rules are simple and no complex PDEs need to be solved. Besides, it is easy to implement with high speed and can meet the demand of real-time rendering of large-scale clouds scene with nearly photorealistic rendering results. And the above are the reasons why we use the cellular automata in our study. At the same time, CA approach is improved in our paper. In the process of illumination calculation, we apply the improved phase function to enhance the efficiency of the simulation without reducing the realism of the simulating images.

The paper is organized as follows: Part 1 is the brief introduction. Part 2 describes the basic idea of CA model, and improved method is proposed based on it. Part 3 is about realistic illumination of clouds rendering, and we describe common scattering models and the promoted phase function is proposed. Experimental results and analysis can be found in Part 4. And finally, we conclude by summarizing our research about clouds simulation and describing our planned future work.

2 Improved Method of Cellular Automata

2.1 Basic Idea of Cellular Automata

According to the modeling approaches of Nagel and Dobashi et al., 3D simulating space is subdivided into grid cells, and three state variables *hum*, *cld* and *act* are set on each grid cell which denote vapor, clouds, state transition between vapor and clouds. Each variable is assigned to 0 or 1, and simulation of clouds is achieved through simple state transition rules as shown in Fig. 1(a). This method is relatively simple, and the cost of calculation is very small. Dobashi et al. realized the extinction process of clouds which was not solved in [7], and inserted smooth transition between cloud(*cld*=1) and without cloud(*cld*=0).

The states of grid cell (i, j, k) at time *t*+1 are generated through exerting the next transition rules on the states at time *t*.

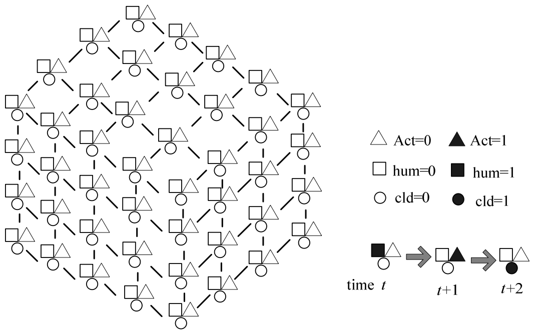
$$act(i,j,k,t+1) = \neg act(i,j,k,t) \wedge hum(i,j,k,t) \wedge f_{act}(\bullet) \quad (1)$$

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

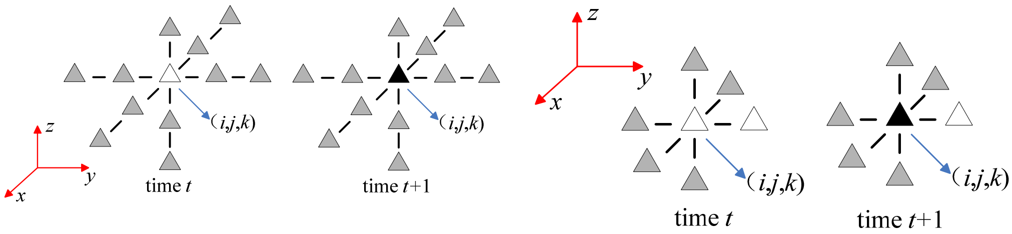
$$hum(i,j,k,t+1) = hum(i,j,k,t) \wedge \neg act(i,j,k,t) \quad (3)$$

where $f_{act}(\bullet)$ is a boolean function, its value is determined by the states of surrounding cells of *act*. Fig. 1(a) shows the above transition rules. If the values of *hum* and $f_{act}(\bullet)$ at time *t* are all 1, then *act* will change to 1 at time *t*+1 and *cld* will be set to 1 at time *t*+2. The value of function $f_{act}(\bullet)$ will be got through the following formula (4).

$$f_{act}(\bullet) = act(i+1,j,k,t) \vee act(i-1,j,k,t) \vee act(i,j+1,k,t) \vee act(i,j-1,k,t) \\ \vee act(i,j,k+1,t) \vee act(i,j,k-1,t) \vee act(i+2,j,k,t) \vee act(i-2,j,k,t) \\ \vee act(i,j+2,k,t) \vee act(i,j-2,k,t) \vee act(i,j,k-2,t) \quad \textbf{(4)}$$



(a) 3D grid cells and states transition



(b) states transition while simulating creation of clouds using variable act (c) states transition while simulating effect of wind using variable act

Fig. 1. Simulation of clouds using cellular automata

Just as Fig. 1(b), the value of $f_{act}(\bullet)$ is 1 as long as the *act* value of any cell around the grid cell (i,j,k) is 1. The effect of the parallel wind blowing was realized in [8] shown as Fig. 1(c).

$$f_{act}(\bullet) = act(i+1,j,k,t) \vee act(i-1,j,k,t) \\ \vee act(i,j,k+1,t) \vee act(i,j,k-1,t) \\ \vee act(i,j-1,k,t) \quad \textbf{(5)}$$

The further simplification of formula (5) is used in this paper, and we get formula (6).

$$f_{act}(\bullet) = act(i+1,j,k,t) \vee act(i-1,j,k,t) \\ \vee act(i,j,k-1,t) \vee act(i,j-1,k,t) \quad \textbf{(6)}$$

We practise the above simplification based on the following reasons: we take into account the rising process of clouds fully just like the air convection while retaining the parallel wind blowing, which resembles the real process of clouds formation with simple calculations. Simulation results show that the resulting clouds are more realistic.

2.2 Simulating Dynamics of Clouds

In [7] the variable *cld* according to formula (2) will no longer be converted to 0 if it is changed into 1, which results a number of complex changes are difficult to simulate such as extinction of clouds. The improvement technique was proposed in [8] focusing on the situation of no clouds extinction in [7]. The improvement in [8] is the addition of a variable *ext* which represents the extinction of clouds, and its transition rule is shown in formula (7).

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

$f_{ext}(\bullet)$ is a function similar to $f_{act}(\bullet)$, which can be calculated by the states of *ext* of surrounding grid cells. Then the transition rule of *cld* can be adjusted as formula (8).

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

However, if using the above rules,when *ext* is 1, *cld* becomes 0, which causes the frequent switching between generation and extinction of clouds in short time and thereby unnatural image. To avoid this distortion, a disappearing time T_{ext} is added in this paper. While *ext* is 1, *cld* will become 0 through time T_{ext}. This can realize natural transformation of clouds shape.

2.3 Setting the Initial States and Processing the Boundary Grid Cells

While setting the initial states, we don't use the method in [7,8] which assigns 0 or 1 according to a given probability, but generate 0 or 1 through a more concise way randomly, which leaves more time for complex lighting calculation.

At the beginning of simulation, we set the values of *hum* and *act* using 0 or 1 generated randomly. However, if *hum* is assigned as 0, *act* can't be initialized to 1. The values of *cld* are set to 0.

We create clouds through updating the status of each varible according to formulas (1)-(3). During the simulation, the generated random numbers follow their probability distributions, and continuous density distribution function is calculated according to [8].

The transition rules of boundary grid cells are not mentioned in [7,8], and this problem is solved in details in our paper. While processing the state transitions of boundary grid cells, calculations of functions $f_{act}(\bullet)$ and $f_{ext}(\bullet)$ will exceed the simulating space. In this paper, when we deal with such grid cells, we will remove these items if grid cells don't exist, i.e. we ascertain which items of right-hand side should be removed in advance through determining the values of *i*, *j* and *k*.

3 Realistic Illumination of Dynamic Clouds in Real-time Rendering

3.1 Different Scattering Models

Realistic simulation of clouds is closely related with illumination calculations, and the scattering of clouds particles meet the Mie scattering law. Blinn first proposed the single scattering illumination model of the interaction between light and cloud particles, which based on geometrical optics and atmospheric scattering characteristics [24]. While rendering the clouds, Dobashi et al. and Miyazaki et al. applied single scattering model in clouds rendering [9,17]. The basic idea of this model is that the whole scattered light is equal to the sum of scattered light from the incident direction to the viewpoint direction and that from the backward to the viewpoint direction shown as Fig. 2. The single scattering model is simple and easy to implement. The cost of calculation is relatively small, but they only consider scattering properties from incident direction to viewpoint direction and ignore the scattering characteristics of the particles themselves so that the realism of simulated images is poor. So we can see that the effects of multiple scattering can't be ignored because cloud particles have a high reflectivity with strong forward scattering characteristics.

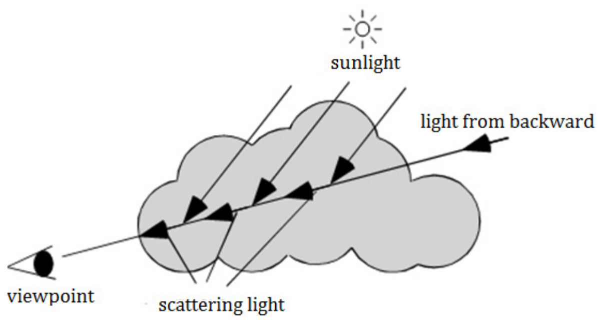


Fig. 2. Single scattering of clouds

Bouthors et al. considered more lighting effects and multiple scattering to render more realistic stratus and cumulus clouds [25,26]. Multiple scattering illumination model is closer to physical characteristics of clouds.

However, because multiple scattering needs to calculate the light intensity in all directions, its time consumption is relatively high.

Harris, on the basis of multiple scattering, improved it in [19]. He indicated that the scattering intensity of clouds particles depends on the forward scattering mainly. In other words, scattered energy is mostly concentrated in a range of small angle toward forward direction. Harris's clouds illumination model is shown in Fig. 3.

So we used multiple forward scattering introduced in [19]. At the same time, the improved phase function is proposed.

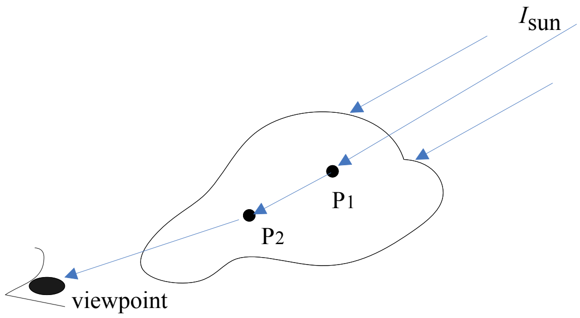


Fig. 3. Multiple forward scattering

3.2 Common Phase Functions

Rayleigh phase function. As shown in formula (9), Rayleigh phase function is applicable to the case of Rayleigh scattering, which requires particle size is much smaller than the wavelength λ of incident light. Rayleigh phase function is simple and fast, but not close to the cloud optical properties. It was used in [19] and yielded good results. But there were some bright spots in some clouds shown as Fig. 5(a).

$$P(\phi) = \frac{3}{4}\frac{(1+\cos^2\theta)}{\lambda^4} \quad \textbf{(9)}$$

Henyey-Greenstein phase function. It described the scattering of radiation in the galaxy firstly [27]. It is useful in scattering calculation of biological organs, water, clouds and many other natural materials, and is an approximation of Mie scattering, expressed as formula (10).

$$P_{HG}(\phi) = \frac{1}{4\pi}\frac{1-g^2}{(1+g^2-2g\cos\phi)^{3/2}} \quad \textbf{(10)}$$

where g is asymmetric factor, controlling the redistribution shapes of the scattered lights.

Cornette-Shanks phase function. Cornette and Shanks amended Henyey-Greenstein phase function and gave a more physically reasonable representation for clouds illumination expressed as formula (11). It was used in [28] while drawing cloud scenes.

$$P_{CS}(\phi) = \frac{3}{2}\frac{(1-g^2)}{(2+g^2)}\frac{(1+\cos^2\theta)}{(1+g^2-2g\cos\theta)^{3/2}} \quad \textbf{(11)}$$

3.3 Improved Phase Function

It can be seen that exponential computations are needed in Henyey-Greenstein and Cornette-Shanks phase functions, which make the computation time greatly improved. Rayleigh phase function is simple, but it is not suitable for the illumination computation of clouds. To solve these problems, improved phase function is proposed based on Cornette-Shanks phase function.

As the scattering of cloud particles focuses on the forward scattering, the exponential item in formula (11) is simplified, and another adjustment item is added. We can get the following simplified phase function defined as formula (12).

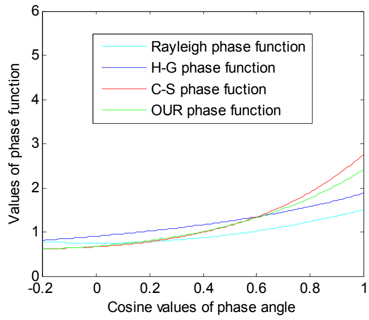
$$P(\phi) = \frac{3}{2}\frac{(1-g^2)}{(2+g^2)}\frac{(1+\cos^2\theta)}{(1+g^2-2g\cos\theta)} + g\cos\theta \quad \textbf{(12)}$$

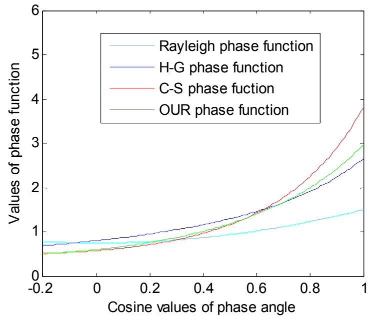
which is equivalent to Rayleigh phase function while g=0. Table 1 lists the calculation times of different phase functions. The calculation time of 100000 times is a statistic period. We get the cosine values through the dot product of the incident light vector and the direction vector in the viewpoint direction. The programming environment is VC6.0, c language and the current hardware configuration.

Table 1. Calculation time of different phase functions(unit: millisecond)

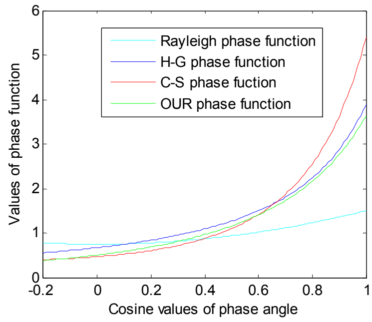
phase function	time
Henyey-Greenstein	0.009905532686417
Cornette-Shanks	0.010388694652533
Our	0.000714062186958

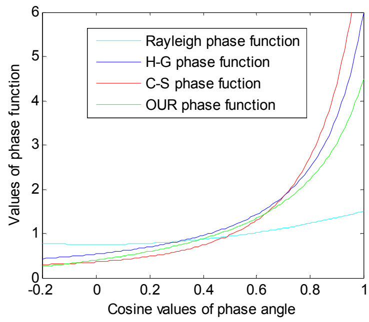
From the calculation times of different phase functions, we can see that there is no great difference between the calculation times of Henyey-Greenstein and Cornette-Shanks phase functions. The calculation time of improved phase function is reduced greatly. Fig. 4 shows that the values of different phase functions when adjustment factor g is different.





(a) g=0.2 (b) g=0.3





(c) g=0.4 (d) g=0.5

Fig. 4. Values of different phase functions

As can be seen from Fig. 4, the proposed phase function approximates the fitting of Henyey-Greenstein and Cornette-Shanks phase functions. Especially when the value of g is small, it is closer to the average of these two functions. Experiments show that the rendering effects of clouds are perfect when the value of g is about 0.3. Thus, we can get pretty rendering scenes just like Henyey-Greenstein and Cornette-Shanks phase functions while using improved phase function, but our phase function is faster in speed and suitable for real-time rendering of dynamic clouds.

4 Experimental Results and Discussion

We have simulated clouds scene by using the improved methods in this paper. The computer that we used was a desktop PC with Intel i5-2380P dual cores 3.10GHz (CPU), 4GB RAM and NVIDIA GeForce GTX 550 Ti (GPU). Most of the computation is carried out by the CPU, while the GPU is only used for displaying the final result.

Fig. 5(a) is the clouds image using Rayleigh phase function in [19], and there are some bright spots. Fig. 5(b) is the clouds image using the improved methods in the paper. Fig. 6 shows the clouds image generated by the rule of formula (5) in [8], and Fig. 7 is a real cumulus photo(the part in small box is the enlarged edge.). Fig. 8 is the clouds image using the improved method in this paper(the part in small box is the enlarged edge.).

It can be seen from Fig. 5(b) and Fig. 8 that clouds images are more realistic after adding convective item, and at the same time the phenomenon of bright spot is eliminated. Compared with the real cloud photo, clouds image generated by our method is relatively close, and especially the effect at the edge is more similar. But there are several dark or bright spots in our clouds images, which is needed to make up in the future work.

Fig. 9(a, b, c) shows the creation process of clouds, and the effect of wind is added during the generation process. Due to the action of wind, the shape of clouds is changed, and even parts of clouds disappear.

Fig. 10(a) is the simulation image generated by the single scattering illumination model in [9], Fig. 10(b) is the effect of clouds using multiple scattering in [26], and Fig. 10(c) shows the results using multiple forward scattering while our improved phase function is embedded in this model. It can be seen that the effect generated by the single scattering is poor, while the effects of multiple scattering and the multiple forward scattering are relatively close, but multiple forward scattering model reduces the cost of calculations and has faster simulation speed. At the same time, the improved phase function can further reduce the computation time. Therefore, the forward scattering illumination model embedded in the improved phase function is more suitable for clouds rendering in real time.

(a) (b)

Fig. 5. Simulated clouds using different phase functions

Fig. 6. Clouds simulated by rule (5) (taken from [8])

Fig. 7. The photo of cumulus

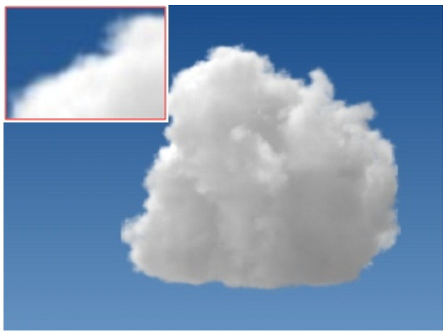
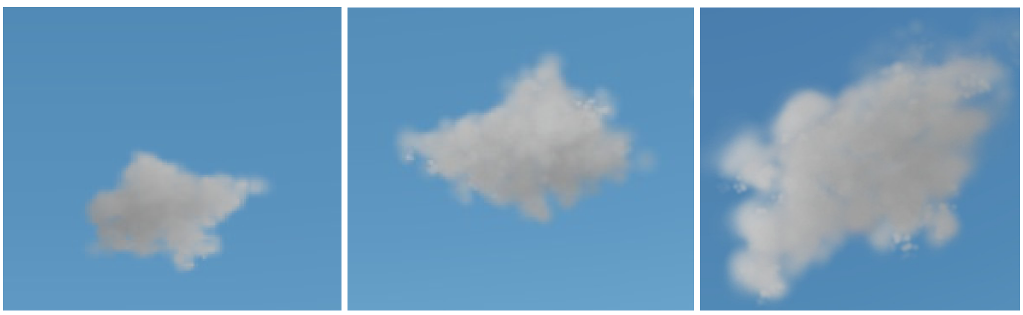


Fig. 8. Simulating clouds using rule (6)



(a) (b) (c)

Fig. 9. Creation of clouds

Fig. 11 is the simulation image generated by our method when lots of clouds are in the simulation space, where sunlight is from the top left.

We verified the improved method using different scales of voxel data. Table 2 shows the real-time frame rates while rendering the above clouds scenes in different data scales. While counting these data, the grid resolution does not exceed 512*512*512 scale.

As can be seen from the statistical data, the proposed method can achieve the real-time rendering of clouds under the current software and hardware conditions when the scale of grid resolution is no more than 256*256*256. When higher grid resolution of simulation space is applied, real-time rendering is difficult to achieve, and artifact occurs. With the increase of data scales, rendering effect of images is getting better. When the data scale reaches 128*128*128, visual requirement can be met.

(a) clouds in single scattering (taken from [9]) (b) clouds in multiple scattering (taken from [25])

(c) clouds in multiple forward scattering

Fig. 10. Clouds in different scattering models

Fig. 11. Scene with lots of clouds

Table 2 Frame rates of rendering in different data scales (unit: frame per second)

Different clouds scene		Single cloud	Common clouds scene	Large-scale cloud	Realism of image
Rendering frame rates	64*64*64	211	197	182	bad
	128*128*64	117	109	102	good
	128*128*128	98	83	53	better
	256*128*128	74	50	29	better
	256*256*256	38	29	15	best

5 Conclusion

In our study the model of clouds is built based on CA, and we have improved state transition rules to accelerate the running and reflect the real characteristics of clouds. At the same time, a clear approach of dealing with boundary cells is introduced. During the rendering, the illumination model of multiple forward scattering is applied to enhance realism, and the improved phase function is embedded in multiple forward scattering model. Through the above work, realistic simulation of dynamic clouds can be achieved in real time.

The techniques of clouds simulation have been greatly developed after years of research. However, due to the complex characteristics of clouds and the enhancement of applications demand, realistic rendering of clouds scene in real time is still one of the most challenging subjects in computer graphics. There are still many issues to be further solved, which are also our future research work. For example, the real-time and realistic rendering of large-scale natural scene will continue to be the focus and challenge of research in many applications such as flight simulation, games etc. In the existing techniques, some technology can provide realistic rendering quality, but the requirement of real-time interaction can't be achieved. Some strategies, like simplified computing, GPU acceleration and so on, may ensure the requirement of speed, but the rendering quality is affected greatly. How to compromise between the realism and real-time interactivity is another emphasis in our future work. As we all know there are complex dynamics in clouds, and most existing methods can only simulate clouds formation, dissipation and other simple movement. How to quickly and vividly show clouds in a complicated simulating environment and emulate the process from clouds to rain is a future research direction. In addition, the establishment of generic simulation platform suitable for all types of clouds will have great significance in flight training, weather simulation, science visualization and other fields.

Acknowledgement

Our research is partly supported by the project of the National Natural Science Foundation of China (NO. 61102167).

References

[1] W. Reeves, "Particle System-A Technique Modeling a Class of Fuzzy Objects," *ACM Transaction on Graphics*, Vol. 2, No. 2, pp. 359-376, 1983.

[2] J. van der Burg, "Building an Advanced Particle System," *Gamasutra*, June 2000, http://www.gamasutra.com/features/20000623/vanderburg_01.htm.

[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, pp. 357-366, 1990.

[4] D. S. Ebert, "Interactive Cloud Modeling and Photorealistic Atmospheric Rendering," in *Proceedings of SIGGRAPH'04*, pp. 238-256, 2004.

[5] Y. Qi, X. K. Shen, M. Y. Duan, H. L. Cheng, “A Method of Rendering Clouds with Perlin Noise,” *Journal of System Simulation*, Vol.14, No. 9, pp. 1204-1207, 2002.

[6] F. Neyret, “Qualitative Simulation of Convective Clouds Formation and Evolution,” in *Proceedings of the 8th Eurographics Workshop on Computer Animation and Simulation*, pp. 113-124, 1997.

[7] K. Nagel, E. Raschke, “Self-organizing Criticality in Cloud Formation?,” *Phisica A*, Vol. 182, No. 4, pp. 519-531, 1992.

[8] Y. Dobashi, T. Nishita, T. Okita, “Animation of Clouds Using Cellular Automaton,” in *Proceedings of Computer Graphics and Imaging’98*, pp. 251-256, 1998.

[9] Y. Dobashi, K. Kaneda, H. Yamashita, T. Okita, T. Nishita, “A Simple, Efficient Method for Realistic Animation of Clouds,” in *Proceedings of SIGGRAPH 2000*, pp. 19-28, July 2000.

[10] Shu-Ya Chen, Shih-Ching Yeh, M. McLaughlin, A. Rizzo, C. Winstein, “Haptic Virtual Reality Pinch Task for Chronic Hemiparesis,” *Journal of Computers*, Vol. 25, No. 1, pp. 48-55, 2014.

[11] B. Qin, T. Lv, “Real-time dynamic cloud modeling and rendering,” *Computer Graphics, Imaging and Vision: New Trends (CGIV’05)*, pp. 285-290, 2005.

[12] J. B. Xv, C. Yang, J. Zhao, L. D. Wu, “Fast Modeling of Realistic Clouds,” *Computer Network and Multimedia Technology*, pp. 1-4, 2009.

[13] E. M. Upchurch, S. K. Semwal, “Dynamic Cloud Simulation Using Cellular Automata and Texture Splatting,” in *Proceedings of the 2010 Summer Computer Simulation Conference*, pp. 270-277, 2010.

[14] T. Yanagita, K. Kaneko, “Coupled Map Lattice Model for Convection,” *Physics Letters A*, vol. 175, pp. 415-420, 1993.

[15] R. Miyazaki, S. Yoshida, Y. Dobashi, T. Nishita, “A Method for Modeling Clouds Based On Atmospheric Fluid Dynamics,” in *Proceedings of Pacific Graphics 2001*, pp. 363–372, 2001.

[16] D. Overby, Z. Melek, J. Keyser, “Interactive Physically-Based Cloud Simulation,” in *Proceedings of Pacific Graphics 2002*, pp. 469-470, 2002.

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

[18] R. Mizuono, Y. Dobashi, Y. B. Chen, T. Nishita, “Physics Motivated Modeling of Volcanic Clouds as a Two Fluid Model,” in *Proceedings of Pacific Graphics’03*, pp. 434-439, 2003.

[19] M. J. Harris, “Real-time Cloud Simulation and Rendering,” Chapel Hill: University of North Carolina, 2003.

[20] W. Ren, X. H. Liang, S. Ma, X. K. Shen, “A Real-Time Simulation Method for Large-Scale 3D Clouds,” *Journal of Computer-Aided Design & Computer Graphics*, Vol. 22, No. 4, pp. 662-669, 2010.

[21] J. Stam, “Stable Fluids,” *Proceedings of ACM Siggraph’99*, pp. 121-128, 1999.

[22] R. Fedkiw, J. Stam, W. H. Jesen, “Visual Simulation of Smoke,” *Proceedings of SIGGRPH’01*, pp. 15-22, 2001.

[23] M. Lentine, M. Cong, S. Patkar, R. Fedkiw, “Simulating Free Surface Flow with Very Large Time Steps,” *SCA'12 Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 107-116, 2012.

[24] F. J. Blinn, “Light Reflection Functions for Simulation of Clouds and Dusty Surfaces,” *Proceedings of SIGGRAPH’82*, pp. 21-29, August 1982.

[25] A. Bouthors, F. Neyret, S. Lefebvre, “Real-time Realistic Illumination and Shading of Stratiform Clouds,” in *Proceedings of Eurographics Workshop on Natural Phenomena*, pp. 41-50, 2006.

[26] A. Bouthors, F. Neyret, N. Max, C. Crassin, “Interactive Multiple Anisotropic Scattering in Clouds”, *ACM Symposium on Interactive 3D Graphics and Games*, pp. 173-182, 2008.

[27] C. L. Henyey, L. J. Greenstein, "Diffuse Radiation in the Galaxy," *Astrophys. J.*, Vol. 93, pp. 70-83, 1941.

[28] R. Miyazaki, Y. Dobashi, T. Nishita, "A Fast Rendering Method of Clouds Using Shadow-view Slices," *Proceedings of International Conference on Computer Graphics and Imaging (CGIM'04)*, pp. 93-98, 2004.