



Rendering of Three-Dimensional Cloud Based on Cloud Computing

Yonghua Xie, Xiaoyong Kou, Ping Li, and Xiaolong Xu^(✉)

School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China

xyh_76@nuist.edu.cn, koux_y@qq.com, 724904461@qq.com, xlxu@ieee.org

Abstract. Cloud modeling and real-time render are of great significance in virtual scene simulation. The lighting model and rendering technology are image synthesis methods, introduced to computer graphics, aiming to simplify virtual scene simulation and enhance the fidelity of complex scenes. Currently, the simulation algorithms applied to cloud scene, inclined to be complicated and computationally intensive. Hence, it is still a key challenge to implement more efficient algorithms to map higher quality three-dimensional clouds. In this paper, a computation-reducing and time-saving method is designed to deal with the above challenge. Technically, the lighting model and rendering technology for the weather research and forecasting (WRF) data are proposed to create the cloud scenes. Then project files are uploaded to the cloud system and directly for real-time rendering at the same time, which can largely save time and reduce the cost of rendering. Finally, adequate experimental analyses are conducted to verify the effectiveness and efficiency of the proposed scheme.

Keywords: Cloud system · Simulation · Three-dimensional · Lighting model · Rendering

1 Introduction

As one of the most common natural phenomena in nature, clouds have been simulated in the field of graphics simulation for a long time [1]. There is still a lot of research on focusing on 3d cloud rendering in that researchers manage to draw three-dimensional clouds of higher quality through more efficient algorithms [2]. The simulation of three-dimensional clouds has developed rapidly in recent years, and researchers proposed a series of cloud simulation and rendering schemes, according to different simulation requirements [3]. Three-dimensional cloud simulation has a wide range of applications, the weather changes in the weather simulation research, especially in the case of extreme weather, the three-dimensional cloud simulation study of meteorology has a very important role; In the field of military simulation imitation air battle scenario, but have to

consider the cloud simulation; In the field of the game, the simulation of the three-dimensional cloud also plays an important role. It is the cloud that plays such an important role in the field of simulation that makes the research of three-dimensional cloud simulation become a hotspot in graphics [4]. So far, three-dimensional cloud simulation can be divided into two categories [5]. The first is to study the physical factors of natural clouds, in the process of simulation, all physical factors of the natural cloud are calculated as far as possible, including gravity, light, wind, temperature, altitude and other factors. The cloud simulated by this kind of method is relatively close to the real cloud, but each frame requires huge computation, which makes real-time rendering difficult [6, 7]. The second approach starts with the appearance of the natural cloud, and only considers whether the cloud looks like the real cloud in the simulation process. This approach is much more efficient than the physical approach, and could even render in real-time on some personal computers [8]. However, there are few researchers using meteorological data to give the simulated clouds real physical information. The effective processing and three-dimensional graphic display of meteorological data could enable weather forecasters to have a more intuitive and in-depth understanding of atmospheric evolution information, so as to make an effective weather forecast. With the development of computer image and graphics technology, scientific computing visualization technology could transform a huge amount of data into static or dynamic images or graphics, providing a powerful means for people to analyze and understand data. With these observations, it continues a challenge to propose a suitable model to endue virtual cloud with real physical significance [9, 10]. In view of this challenge, a computation-reducing and time-saving method is proposed. Our main presentations are enumerated as follows:

- Analysis of the meteorological principles of cloud formation and a spherical particle model is built, which constructs the virtual clouds.
- The reflection and projection characteristics of cloud particles are analyzed. Simplify the lighting model and rendering method to improve real-time interaction.
- Cloud computing mode is adopted to complete the rendering task of three-dimensional cloud.
- The validity and effectiveness of the proposed method are verified by sufficient experimental evaluation and comparative analysis.

The rest of this paper is organized as follows. Section 2 gives basic concepts and definitions. Section 3 explains the method we use. Section 4 shows the performance of our method by experiments. Section 5 lists the relevant work. Section 6 summarizes the conclusions.

2 Modeling and Simplification

2.1 Characteristics of Cloud

Clouds are composed of granular ice in the atmosphere and a large number of water droplets. They play an important role in the operation of the water cycle

and are an important link in the three-state change of water (gas, liquid and solid). Under the action of sunlight, wind and so on, water in nature, as well as water in animals and plants, will continuously evaporate and be converted into gaseous water vapor and dispersed into the atmosphere. In the atmosphere, temperature decreases with increasing altitude, and gaseous vapor may condense or liquefy as a result of the change in temperature, and then turn into ice particles or water droplets. Impurities in the air play a crucial role in the formation of ice particles or water droplets. If the impurities are not present, solid or liquid water cannot be condensed together. Even if some ice particles or water droplets are condensed together, they will quickly evaporate into a gas state due to the absence of impurities. The distribution range of these impurities is very wide. When the impurities encounter ice particles or water droplets, they can quickly condense into larger ice particles or water droplets. Under the influence of air flow, wind speed and other factors, these larger water droplets condense with each other again, and the final condensate is the cloud we see in our daily life. The shapes of clouds are caused by the constant movement of ice particles or water droplets in the clouds under the influence of various meteorological factors. At the same time, clouds show a variety of colors because the sunlight is refracted, reflected, and scattered physically as it passes through clouds.

2.2 Meteorological Representation

In this section, The related elements of clouds are represented by means of meteorological study. The movement and formation of clouds are caused by a variety of meteorological elements. In order to show them objectively, the following is a brief introduction of the physics of the relevant elements. In meteorology, the saturation degree of vapor is generally expressed by relative humidity f , as shown in formula (1) :

$$f = \frac{e}{E} \tag{1}$$

where e is the actual vapor pressure and E is the vapor pressure. With the constant change of temperature, the value of saturated vapor pressure will also change with it. The Clausius-Clapeyron Differential Equation mainly represents the relation function between saturated vapor and humidity, which is expressed as. The relative humidity can be calculated by sending a differential equation through calculation (2):

$$\frac{dE}{E} = \frac{L}{AR} \cdot \frac{dT}{T^2} \tag{2}$$

where A represents the heating equivalent, L represents the latent heat of phase change, T is the transformation temperature, and R represents the gas constant of vapor. After differential solution, it can be obtained, as shown in formula (3):

$$\frac{df}{f} = \frac{de}{e} - \frac{dE}{E} = \frac{de}{e} - \frac{1}{E} \left(\frac{dE}{dT} \right) dT = \frac{de}{e} - \frac{L}{EAR} \cdot \frac{dT}{T^2} \tag{3}$$

According to the above equation, the relative humidity will increase with the increase of water vapor. As the temperature decreases, the relative humidity will also decrease, thus reaching saturation. Generally speaking, the decrease of temperature is the main reason for the formation of clouds. The reason is that vapor will keep rising, while the temperature will gradually decrease with the increase of height when the state of humidity changes. When vapor rises to a certain degree, the physical state of vapor will change, leading to the formation of clouds. The condensation of vapor produces different types of clouds at different heights because the height and size of the air mass will cause different physical changes at different times.

2.3 Simplification

On the basis of the above analysis, clouds are thought to be made up of countless tiny particles with different components of their properties (position, color, speed, temperature, etc.). How to model objects with particle is shown in Fig. 1. The region of cloud generation is defined as the xoz plane parallel to the world coordinate system, a circle with center of (x_0, y_0, z_0) and radius of r . After determining the generation region of particles, it is necessary to determine the number of particles that should be produced in each frame. It directly affects the density of the simulated object, and thus affects the fidelity of simulation and the real-time performance of the system. It is one of the most important parameters of the particle system. If the number of particles is too small, it cannot meet the requirement of reality; if the number of particles is too large, it will increase the processing time of the system and reduce the real-time performance of the particle system. So the definition of the number of particles became a key issue in the whole simulation. In traditional particle systems, the generation of particles is controlled by random functions. There are two methods of measuring the average number of new particles and measuring the area of objects. The two methods ignore the important influence of the number of particles on the simulation, which sometimes leads to the waste of the number of particles.

The number of particles required in the system is related to the scale of smoke production and the distance from the viewpoint. When the cloud scale is certain and the viewpoint is far away, due to the low resolution of human eyes, fewer particles can be used to generate clouds that meet the visual requirements. On the contrary, when the viewpoint is closer, more particles are needed to produce detailed details. Based on this, a function $f(r, d)$, which is determined by the scale of the cloud to be plotted and the distance of the cloud from the viewpoint, is proposed to determine the number of generated particles. Where r is the radius of the region where particles are generated, d is the distance between the cloud and the viewpoint, and $n = f(r, d)$ is used to describe clouds of different sizes that meet the visual effect. In this way, the number of particles can be reduced appropriately, the time of computer processing can be reduced, and the rendering efficiency of the system can be improved. When the direct proportionality function is used to describe the relationship between the number of particles and the distance of viewpoint, if the proportionality coefficient is

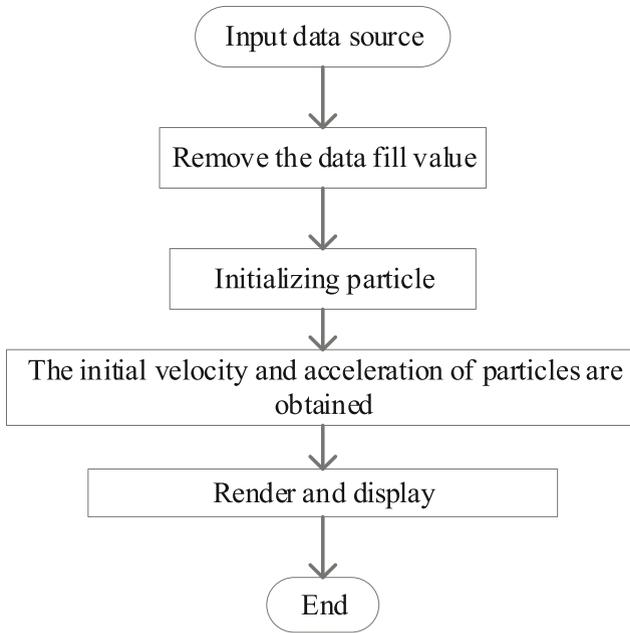


Fig. 1. Particle modeling flow chart

relatively large, the rate of change will change rapidly linearly with the increase of the distance of viewpoint, which makes the number of particles jump too strongly. If the proportionality coefficient is small, the viewpoint distance is far away, but still maintain a more complex state, not to achieve the effect required to simplify the number of particles.

Therefore, as the distance increases, the blurring degree of the object should not change linearly. The number of particles produced by the model increases gradually when the viewpoint is close and the rate of change is large, while the number of particles produced by the model decreases gradually when the viewpoint is far away and the rate of change is small. In order to satisfy the actual situation mentioned above and ensure the sense of reality, take the following non-linear relationship:

$$\alpha = 1 - e^{p/a} \tag{4}$$

where α is the rate of change, $p = -d/d_0$, a is an empirical constant, 280 is appropriate.

Let n_0 be the number of generated particles when the distance of viewpoint is $d_0 = 1$ and $r_0 = 1$, n is the number of particles when the distance of viewpoint is d and radius of cloud is r . At the same time, The rate of change can also be defined as:

$$\alpha = \frac{n_0 - n}{n_0} \times 100\% \quad (5)$$

from Eqs. (1) and (2), it is known that:

$$n = f(r, d) = \left(\frac{r}{r_0}\right)^3 \times n_0 \times e^{p/a} \quad (6)$$

2.4 Illumination and Rendering

The basic task of illumination is to achieve three-dimensional scene rendering and generate realistic images, which is the inevitable development direction of real-time rendering. In rendering, it is necessary to simulate various physical phenomena of light propagation in the scene, such as multiple reflections, refraction, shadow, coloration and causation. It requires an accurate description of geometric features and material properties of various model objects created in the scene, as well as calculation and solution of infinite calculus problems caused by multiple refraction and that reflection. Due to the complexity of calculated lightmass, rendering a realistic image usually takes a long time. Illumination rendering algorithm has always been the focus and hotspot of graphics research. Various researchers focus on how to render high-quality realistic images in less time and achieve real-time rendering requirements.

The light reflection properties of an object surface are usually described by the (bidirectional reflectance distribution function (BRDF). BRDF describes the amount and variation of light reflected, absorbed, and transmitted (refracted) based on the surface properties. Although these reflection distributions are random, they follow certain rules. The reflected energy is concentrated in one direction, and the reflected angle is equal to the incident angle. Diffuse reflection (Lambert body) object surface is rough enough, its radiation brightness is constant in 2π space at the center of the object, that is, the radiation brightness does not change with the point of view, also known as isotropic BRDF. The reflection intensity of the non-Lambert body is uneven in all directions, also known as anisotropy BRDF.

The rendering equation describes the entire physical process in which light energy is emitted from the light source, reflected and refracted by various material surfaces in the scene, and finally enters the observer's eyes. In fact, the process of solving the rendering equation is the rendering process of the scene.

The classical rendering equation is defined as a point on a plane that radiates in all directions in a hemispheric range. Divide all object surfaces in the scene into small regional surfaces, and convert the rendering equation from the integration of hemispheric direction to the integration of all object surfaces in the scene (Fig. 1). In that sense, the rendering equation is obtained as follows:

$$L(x, x') = L_e(x, x') + \int \Omega/2L(x', x'')f(x, x', x'')G(x, x', x'')dx \quad (7)$$

$$G(x, x', x'') = V(x, x')V(x', x'') \cos \theta \tag{8}$$

where x, x', x'' represents the position of the scene; $f(x, x', x'')$ is known as BRDF; $G(x, x', x'')$ is the geometry element, dealing with phenomena of attenuation and occlusion ; $V(x, x')$ is visibility function; If x and x' are visible to each other, then $V(x, x')$ equals 1; otherwise, it is 0 due to occlusion. The integral of all the emitting surfaces that can be directly transmitted to the x' . The scenario discussed in this article does not consider the spontaneous light of an object, so ignore the $L_e(x, x')$ term for the time being.

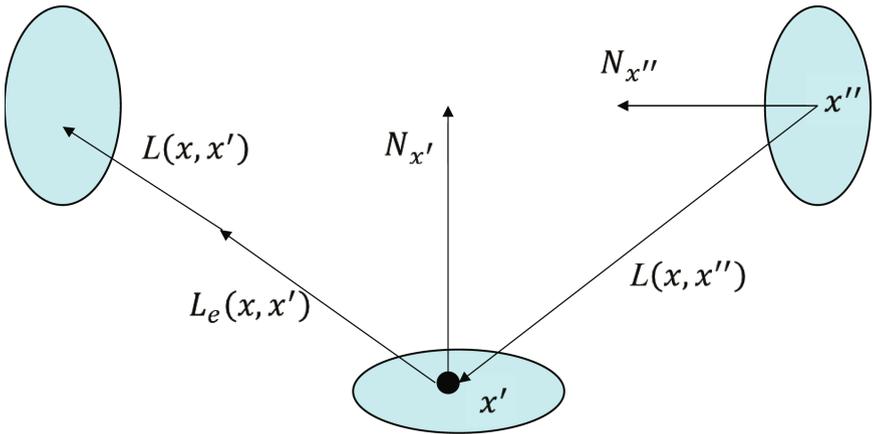


Fig. 2. The integration of all object surfaces in the scene.

Equation (7) shows that light passes through a reflection. If the incident radiation brightness is further distinguished according to different reflection times, then the radiation brightness of the point x' to x , including all radiation brightness that reaches x' through reflection and reaches x through reflection, could be represented as

$$L(x, x') = \sum_{n=1}^{\infty} L_n(x, x') \tag{9}$$

Let us suppose that the scene is divided into multiple small R_i , rest of another patch material within the same everywhere, with the same reflection properties, the incident radiation intensity according to the different reflection paths to further distinguish, for $n + 1$ times of reflection, different reflection paths to point x' radiation brightness after x' reflection to x radiation brightness could be recursively description for:

$$L_{n+1} = \sum_{P_n} \sum_{R=1}^N L'_n(x, x', P_n(R, f_R)) \tag{10}$$

where, $P_n(R, f_R)$ represents the propagation path of light, that is, the light emitted from the source passes through all the surfaces that x passes through after n times of reflection. L'_n describes the luminance of x' towards position x followed n reflections of the path described by P_n .

The traditional Blinn-Phong illumination model describes the illumination effect of the scene from the different reflection characteristics of the object surface to diffuse, specular and ambient light, as well as the attenuation of the light along the propagation path. The algorithm in this paper assumes that the path of light is transformed only when it touches the surface the of object, regardless of refraction and ambient light, and uses Blinn-Phong's BRDF to represent the material of the object. Different materials will be described by changing the parameters of the function. The function could be described as

$$f_{BP} = \frac{K_d}{\pi} + K_s \cdot \frac{n_s + 2}{2\pi} \cos^{n_s} \gamma \quad (11)$$

where K_d stands for the diffuse reflection coefficient; K_s stands for high light coefficient; γ represents the angle between the incident, exit and the normal direction of the reflection position; N_s represents the roughness of the object surface. N_s uniformly sampled n_s times within the selected range to obtain N_s linearly independent BRDFs. Diffuse reflection usually presents a primary and sensitive visual effect, so the diffuse reflection of the BRDF sample is directly used as a basis. The remaining $N_s - 1$ BRDFs are extracted and dimensionalized by principal component analysis to obtain $N_b - 1$ eigenvectors, a total of N_b bases are obtained.

The BRDF has a set of basis, and the BRDF function on face R could be expressed with them

$$f_R(x, x', x'') = \sum_{i=1}^{N_b} c_i f_i(x, x', x'') \quad (12)$$

where f_i represents the i th basis of BRDF on surface R ; c_i is the coefficient corresponding to the i basis; N_b is the number of bases. Substitute Eq. (11) into Eq. (9), and recurse to n to get

$$L_n(x, x') = \sum_{\{R_i\}=1}^N \sum_{\{f_i\}=1}^{N_b} (L'(x, x', p(R, f_i))) \prod_{i=1}^n c_i \quad (13)$$

The above is the result of n reflections, just two reflections, where n equals 2, and take the point of view x_e at x .

After obtaining BRDF bases, all faces in the scene are select BRDF from these bases, and calculate the corresponding illumination effect of various distribution combinations of these bases by using ray projection, so that $L'(x_e, x', P_1(R, f'))$ and $L'(x_e, x', P_2(R, f'))$ in Eq. (12) could be obtained. Assuming that there are n facets in the scene, all n facets are introduced into Eq. (12) to get

$$L_n(x, x') = \sum_{f' \in B(R')} c'_i \sum_{\{R_i\}=1}^N \sum_{\{f_i\} \in B(R')} (L'(x, x', P_n, f') \prod_{i=1}^n c_i) \quad (14)$$

where R' represents the face where x' is; f' is the corresponding BRDF. Take two reflections into account and take x as the viewpoint x_e , then the first reflection is

$$L_1(x, x') = \sum_{f' \in B(R')} c'_i L'(x_e, x', P_1, f') \quad (15)$$

The second reflection is

$$L_2(x, x') = \sum_{f' \in B(R')} c'_i \sum_{\{R_i\}=1}^N \sum_{\{f_i\} \in B(R')} (L'(x_e, x', P_2, f') c_i) \quad (16)$$

2.5 Algorithm Structure

(1) The light source is represented by illumination function $I(x, y, z, \theta, \varphi, \lambda)$ with 6 properties, where (x, y, z) represents the position coordinate of the light source; (θ, φ) said the direction of the light in the three-dimensional space; λ is the energy intensity of the light source. (2) The direct illumination and radiation luminance I_d of each plane is an attribute of the plane, which can be directly accessed when calculating the first reflection or the second reflection. (3) The algorithm in this paper only calculates two reflections to satisfy the user's requirements. The first reflection mainly provides direct illumination, which is provided by all the direct visible light sources in the scene. The second reflection provides indirect light, which is contributed by the light provided by other surfaces through the second reflection, and the light energy is preserved in the surface property I_i .

3 Experiment and Evaluation

This paper puts a large number of scenario component location calculations into the cloud computing platform and transfers them to the cloud server using dynamic component processing. After the implementation of cloud computing services, different task forms are assigned according to the quantity and demand detection in the overall environment. Service cost must be calculated for some large service forms. Segmentation of service modes can effectively promote the efficiency of pattern execution and strategy formulation. The frame data to represent the contrast result, contrast experiment result is shown in Fig. 3.

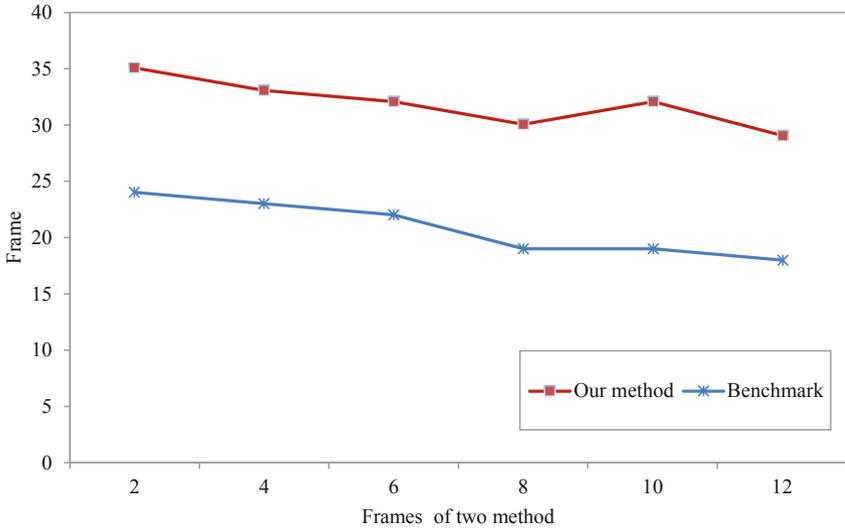


Fig. 3. Frames of two methods.

From Fig. 3, when the game thread is serialized with the rendering thread, the Frame floats around 30, while when the game thread is parallel with the rendering thread, the Frame floats around 20, which reduces the time spent on a Frame by about 40. Figure 4 shows the rendering results of our method and Fig. 5 shows Benchmark's, obviously, the former has more details.



Fig. 4. An example of cumulus cloud.

When the cloud cluster is 20 and the number of particles is within the range of [5,000, 8000], the frame number of Benchmark rendering is between 27 and 50. The frame number of dynamic cloud rendering with time variable is between 43 and 65. The rendering effect is shown in Fig. 2. The test results show that this algorithm greatly improves the rendering efficiency and Fig. 7 is the result (Fig 6).



Fig. 5. An example of cloud.

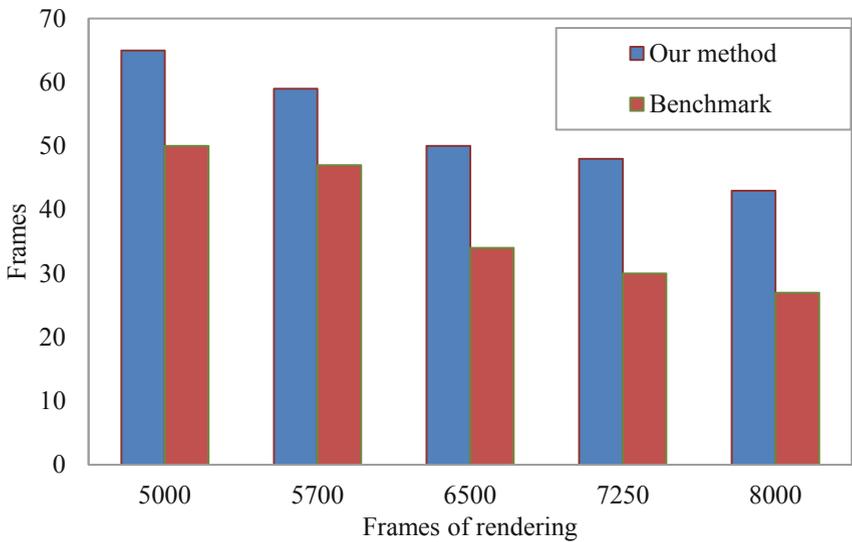


Fig. 6. Frames of rendering.



Fig. 7. An example of stratus cloud.

4 Related Work

In recent years, researchers at home and abroad have proposed a series of cloud simulation and rendering methods to achieve better simulated performance, according to different simulation requirements. In terms of ways of modeling, it can be divided into individual growth-based model and physics-based model [11].

The modeling method based on individual growth can obtain the visual shape of cloud by using the visual morphological features, without simulating the real physical process of cloud formation. Meanwhile, the density distribution of cloud is generated by fractal theory and the effect of cloud is simulated by texture generation [12]. This method has a small amount of cloud computing and is easy to implement. When the viewpoint is far away from the cloud, it has a good effect. However, when the viewpoint is very close to the cloud, and even requires the effect of penetrating the cloud, this method cannot meet such requirements [13]. Currently, the commonly used methods include body process method, fractal method and particle system method.

The volume process method adopts noise (such as Perlin noise) function to simulate the complexity of the cloud, and applies the same rendering technology to every element in the scene [14], which requires a large amount of computation and is difficult to generate real-time animation. Fractal method adopts iterative or recursive method combined with deformation ball to calculate individual growth, which is suitable for generating static fine graphics [15]. Relative to the front two methods, the method of particle system direct simulation of particles in the clouds, and give the particle size, position and color attributes, such as through texture mapping, mapping eight side ring, the formation of different types of cloud, and realized the real-time simulation of large-scale three-dimensional cloud, but the method from the angle of the nonphysical to describe,

not well reflect the cloud physics authenticity. Direct concept, easy to control the shape of clouds, suitable for real-time flight visual simulation. However, the physics-based modeling method is to build the cloud model by simulating the meteorological generation process of cloud, which requires too much computation and cannot meet the current needs of image simulation. On the basis of Harris and according to the laws of atmospheric physics, Li et al. derived a new net buoyancy formula and simulated the formation of cumulus clouds in a more real way [16]. However, this method was realized by increasing cloud sources when simulating a large number of clouds, which was not suitable for large-scale cloud simulation.

According to the existing work, we have learned that particle system and view-based multi-level detail model are beneficial to the simulation of large-scale cloud shape; Body element algorithm, program noise algorithm and texture mapping algorithm were used to simulate the shape of cloud with higher detail level [17]. On the basis of voxel grid [18], the algorithm for solving partial differential equations can be used to simulate the dynamic effect of cloud to get better results. Using spherical harmonic function method, finite element integration method, Monte Carlo method and other methods to accurately solve the multiple scattering is beneficial to the illumination calculation of clouds with higher trueness [19]. Mining Linear integration method and linear interpolation method are beneficial to the real-time rendering of cloud in large-scale scene [20]. Different algorithms should be developed to simulate different characteristics of the cloud. From the initial drawing a cloud in the smaller scope, then draw a certain scale of cloud, the cloud rendering technology has made great progress, but the technology is still far can not meet the demand of the simulation of flight simulator in flight simulator, the scene big wide scope of speed, real-time generate all kinds of different requirements of clouds is very challenging.

5 Conclusion

In this paper, simulation methods and graphic algorithm have been studied, which applies to irregular fuzzy objects , such as clouds, smoke and fog. Specifically, simplification of particle system and rendering method of three-dimensional cloud has been established based on cloud computing. We simplify the lighting model and rendering method to improve real-time interaction. Cloud computing mode is adopted to complete the rendering task of the three-dimensional cloud. At last, the experimental results show that the proposed method has achieved that the cloud generated has meteorological significance, using real meteorological data. While effectively reducing the number of particles, the realistic visual effect is guaranteed and the rendering efficiency is improved. Lots of meteorological factors have influence on the cloud formation , and it is difficult to build a model taking all factors into consideration. In the future, more effective methods with better performance will be proposed.

Acknowledgment. This research is supported by the National Natural Science Foundation of China under grant no. 41675155.

References

1. Grabner, A., Roth, P.M., Lepetit, V.: 3d pose estimation and 3d model retrieval for objects in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3022–3031 (2018)
2. Xu, X., Liu, Q., Zhang, X., Zhang, J., Qi, L., Dou, W.: A blockchain-powered crowdsourcing method with privacy preservation in mobile environment. *IEEE Trans. Comput. Soc. Syst.* **6**(6), 1407–1419 (2019)
3. Berger, M., Li, J., Levine, J.A.: A generative model for volume rendering. *IEEE Trans. Vis. Comput. Graph.* **25**(4), 1636–1650 (2018)
4. Kniss, J., Premoze, S., Hansen, C., Shirley, P., McPherson, A.: A model for volume lighting and modeling. *IEEE Trans. Vis. Comput. Graph.* **9**(2), 150–162 (2003)
5. Qi, L., Chen, Y., Yuan, Y., Fu, S., Zhang, X., Xu, X. A qos-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. In: World Wide Web, pp. 1–23 (2019)
6. Chawla, I., Osuri, K.K., Mujumdar, P.P., Niyogi, D.: Assessment of the weather research and forecasting (wrf) model for simulation of extreme rainfall events in the upper ganga basin. *Hydrol. Earth Syst. Sci.* **22**(2) (2018)
7. Choi, T., Ghan, S., Chin, S.: Biological property-based artificial scar synthesis using inverse lighting. *Multimedia Syst.* **24**(4), 407–418 (2017). <https://doi.org/10.1007/s00530-017-0564-7>
8. Bae, S.Y., Hong, S.Y., Lim, K.S.S.: Coupling wrf double-moment 6-class micro-physics schemes to rrtmg radiation scheme in weather research forecasting model. In: *Advances in Meteorology*, 2016 (2016)
9. Xu, X., Dou, W., Zhang, X., Chen, J.: Enreal: an energy-aware resource allocation method for scientific workflow executions in cloud environment. *IEEE Trans. Cloud Comput.* **4**(2), 166–179 (2015)
10. Qi, L., He, Q., Chen, F., Dou, W., Wan, S., Zhang, X., Xu, X.: Finding all you need: web APIs recommendation in web of things through keywords search. *IEEE Trans. Comput. Soc. Syst.* **6**(5), 1063–1072 (2019)
11. Ahasan, M.N., Debsarma, S.K.: Impact of data assimilation in simulation of thunderstorm (squall line) event over bangladesh using wrf model, during saarc-storm pilot field experiment 2011. *Nat. Hazards* **75**(2), 1009–1022 (2015)
12. Li, Z., Shafiei, M., Ramamoorthi, R., Sunkavalli, K., Chandraker, M.: Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image (2019). arXiv preprint [arXiv:1905.02722](https://arxiv.org/abs/1905.02722)
13. Smirnova, T.G., Brown, J.M., Benjamin, S.G., Kenyon, J.S.: Modifications to the rapid update cycle land surface model (ruc lsm) available in the weather research and forecasting (wrf) model. *Mon. Weather Rev.* **144**(5), 1851–1865 (2016)
14. She, J., Tan, X., Guo, X., Tan, J., Liu, J.: Rendering 2d lines on 3d terrain model with optimization in visual quality and running performance. *Trans. GIS* **21**(1), 169–185 (2017)
15. Xu, X., Yuan, C., Liang, X., Shen, X.: Rendering and modeling of stratus cloud using weather forecast data. In: 2015 International Conference on Virtual Reality and Visualization (ICVRV), pp. 246–252. IEEE (2015)
16. Sur, F., Blaysat, B., Grediac, M.: Rendering deformed speckle images with a boolean model. *J. Math. Imaging Vis.* **60**(5), 634–650 (2018)
17. Qi, L., et al.: Structural balance theory-based e-commerce recommendation over big rating data. *IEEE Trans. Big Data* **4**(3), 301–312 (2016)

18. Drăgan, I., Selea, T., Fortiș, T.-F.: Towards the integration of a HPC build system in the cloud ecosystem. In: Barolli, L., Terzo, O. (eds.) CISIS 2017. AISC, vol. 611, pp. 916–925. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-61566-0_87
19. Yáñez-Morroni, G., Gironás, J., Caneo, M., Delgado, R., Garreaud, R.: Using the weather research and forecasting (wrf) model for precipitation forecasting in an andean region with complex topography. *Atmosphere* **9**(8), 304 (2018)
20. Rautenhaus, M., et al.: Visualization in meteorology a survey of techniques and tools for data analysis tasks. *IEEE Trans. Vis. Comput. Graph.* **24**(12), 3268–3296 (2017)