

Real-Time Rendering of Dynamic Clouds Using Multi-Resolution Adaptive Grids

Fan Xiaolei (范晓磊)^{1,2*}, Zhang Limin (张立民)³, Zhong Zhaogen (钟兆根)¹

1. Department of Electronics and Information Engineering, Naval Aeronautical and Astronautical University, Yantai 264001, P. R. China;

2. Noncommissioned Officers Vocational and Technical Education College, the Second Artillery Engineering University, Qingzhou 262500, P. R. China;

3. Department of Scientific Research, Naval Aeronautical and Astronautical University, Yantai 264001, P. R. China

(Received 12 March 2014; revised 17 June 2014; accepted 24 June 2014)

Abstract: The multi-resolution adaptive grids method is proposed to solve the problems of inefficiency in the previous grid-based methods, and it can be used in clouds simulation as well as the interactive simulation between objects and clouds. Oriented bounding box (OBB) hierarchical trees of objects are established, and the resolutions of global and local grids can be selected automatically. The motion equations of fluid dynamics are simplified. Upwind difference is applied to ensure the stability of the simulation process during the discrete process of partial differential equations. To solve the speed problem of existed phase functions, the improved phase function is applied to the illumination calculation of clouds. Experimental results show that the proposed methods can promote the simulation efficiency and meet the need for the simulation of large-scale clouds scene. Real-time rendering of clouds and the interaction between clouds and objects have been realized without preprocessing stage.

Key words: real-time rendering; multi-resolution adaptive grids; clouds simulation; phase function

CLC number: TP391.9 **Document code:** A **Article ID:** 1005-1120(2015)04-0428-10

0 Introduction

Clouds simulation is a research hotspot in the computer graphics field. For a common fluid, simulation methods of clouds can be classified into two categories: heuristic methods and physics-based methods^[1-2]. With the rapid development in hardware and software, physics-based methods have become the mainstream in recent years. The key problem of the physics-based methods is to solve the partial differential equations. Clouds were generated based on fluid dynamics in Refs. [3-5]. The methods are suitable for certain type of clouds simulation. Harris simulated dynamic clouds using physics-based method, and realized the interactive simulation between one aircraft and clouds^[6]. Large-scale clouds scene was simulated based on fluid dynamics without dynamic in-

teraction in Ref. [7].

Grid-based methods are widely used in fluid simulation to solve the partial differential equations. A grid space is generated to calculate the motion of fluids. Navier-Stokes (N-S) equations are discretized by using grids, and finite difference method is used to solve the equations. Many fluid simulations utilize the single grid^[6,8-11]. While representing the shape of interactive objects with fluid, the number of grid cells is increased greatly, and GPU acceleration is hard to implement. The interaction between smoke and rigid object was simulated in Ref. [12], and different resolution grids were established for them. However, the technique failed to achieve the adaptive selection of resolutions and affected its practicality. And simulation is also performed on the occupied grid cells by interactive object, re-

* **Corresponding author:** Fan Xiaolei, Lecturer, E-mail: gm-fxl@163.com.

How to cite this article: Fan Xiaolei, Zhang Limin, Zhong Zhaogen. Real-time rendering of dynamic clouds using multi-resolution adaptive grids[J]. Trans. Nanjing U. Aero. Astro., 2015, 32(4): 428-437.

<http://dx.doi.org/10.16356/j.1005-1120.2015.04.428>

sulting in frequent data transfer. Refs. [13-14] applied adaptive multi-grid method to implement the interaction between smoke and the rigid object. However, they need to calculate all grid cells including interactive object, thus lowering the efficiency.

To solve the above problems, the multi-resolution adaptive grids method is proposed in this paper to simulate clouds scenes and the interaction between clouds and objects in real time. Furthermore, the improved phase function is applied to the model of multiple forward scattering in clouds illumination.

1 Multi-resolution Adaptive Grids Method

1.1 Multi-resolution grids

Multi-resolution grids method regards the whole simulation space as evenly-divided global grid space. Fig. 1 is a two-dimensional (2D) uniform grid space, and the shaded areas represent the interactive objects. The object at the bottom right represents the effect after the movement. Green grids surrounding the interactive objects are called the local grids and they have different resolutions. Blue boxes surrounding the interactive objects are oriented bounding boxes (OBBs) at the top of Fig. 1. Local grids are created at the edge of interactive object. Interactive objects can move freely without the limitation of speed, and their corresponding bounding boxes make the same movement. Fig. 2 is a three-dimensional (3D) case, where the red shadow represents the interactive object, and a separate grid cell is at the right.

Here one also uses the alterable global grid resolution. Since updating global grid resolution is time-consuming, the global grid resolution is not frequently changed according to the judgment rule.

1.2 Splitting and combination of grid cells

1.2.1 Principles of splitting and combination

Both for global grid or local grid, the splitting and combination of grid cells will appear

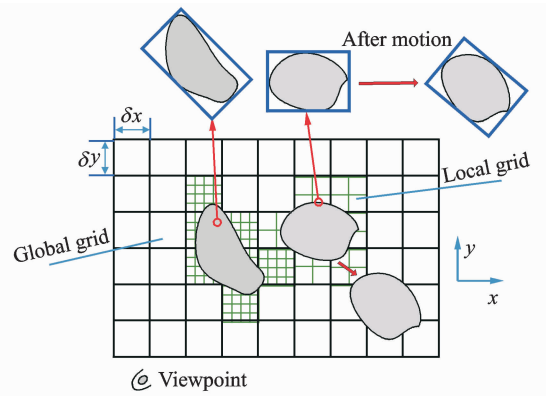


Fig. 1 2D multi-resolution uniform grid

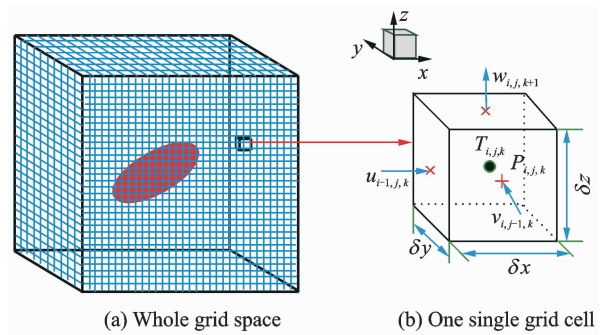


Fig. 2 3D uniform grid

while changing the grid resolution. For the different dimension of the grid, the number of grid cells in the axial direction is 2^n , where n is a positive integer (n can be different).

For global grid, the starting position of the combination and splitting will move on from the origin along each axis. The combination and splitting of local grid cells are implemented along the edge cells occupied by interactive objects. Fig. 3 shows the processes of combination and splitting in 2D, where the reverse process is splitting.

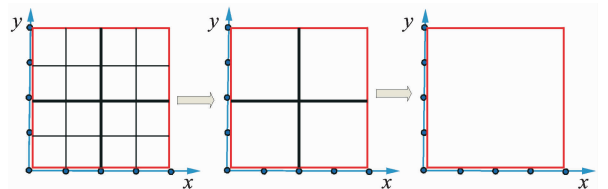


Fig. 3 Combination of 2D grid cells

1.2.2 Data processing after splitting and combination of grid cells

Redundant data will appear after the combination of grid cells. The splitting will generate

new grid cells, and these grids need to be assigned to new data. One uses linear interpolation to handle data transfer and transformation of different resolution grid cells. The data transfer after the combination and splitting is consistent with the case of Fig. 4. When converting high-resolution grids to low-resolution grids, the data of new cells are $A = (a + b + c + d) / 4$, and the data of new cells will be $a = \frac{1}{4}(A_1 + A_3) + \frac{3}{8}A + \frac{1}{8}A_2$ conversely.

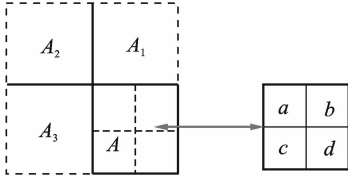


Fig. 4 Data transfer after splitting or combination of grid cells

1.3 Processing occupation of grid cells

1.3.1 Determination of whether the object occupies grid cells

Hierarchical OBB bounding boxes are created for interactive objects in order to accelerate the intersection test between the objects and the fluid cells. Their data are stored in the hierarchical oc-tree data structure. The bounding box of the outermost layer is at the top level of data structure, and the basic information of the object is stored at the bottom level of data structure. One only creates a hierarchical data structure for the surface of the object because the object is rigid and it does not deform during the interaction, which can promote the speed of the creation of the bounding box. A status flag for each grid cell (including global and local grids) is set to mark the occupation state of the grid cell. To determine whether the object occupies the grid cell, one performs intersection test between the hierarchical OBBs of the object and the grid cell in a top-down order. According to the intersection cases between the leaf nodes and grid cells, if they are the cases shown as Figs. 5(a, b), the object does not occupy the grid cell (Disjoint situation certainly does not occupy the grid cell.). Only in the case of

Fig. 5(c), grid cell is occupied by the object, and the status variable of the grid cell is marked as occupied.

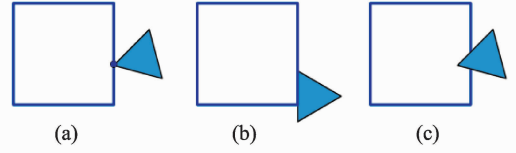


Fig. 5 Relationship between object and grid cell

1.3.2 Data processing of grid cells after object movement

After the motion of the interactive object, new global grid cells will be occupied or some others are unoccupied again. One treats the data on these cells as follows.

(1) From occupied to unoccupied

One handles the data from the edge of occupied area in terms of the order from outside to inside. If status variable of a grid cell is changed from occupied to unoccupied, new data are determined by data interpolation of surrounding unoccupied cells. Then flag variable of the cell is set to unoccupied.

(2) From unoccupied to occupied

It is not necessary to process the data stored in the cell, and the flag variable of the cell is only set to occupied.

After the motion of the object, the resolution of local grid can change and is not consistent with the one of global grid. Just like the red triangle object in Fig. 6, and the higher resolution of the local grid is required. One treats the case as follows:

Step 1 Determine new resolutions of global and local grids;

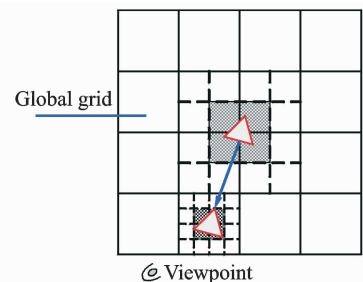


Fig. 6 Change of occupied cells after object movement

Step 2 Split or combine global grid, and deal with its data;

Step 3 Determine the newly occupied and unoccupied cells between the object and the global grid, and the corresponding data processing is needed;

Step 4 For the newly occupied cells of the global grid, split or combine them according to the new resolution of the local grid, i. e., deal with the splitting or combination of the local grid.

Step 5 Determine whether local grid cells are occupied again. When multiple objects are involved in the interaction, the overlap must be avoided.

While transferring data between the grid cells, one just considers the data transfer between global grid cells in different resolutions. For local grid cells, just deals with their status flags to display interactive objects.

1.4 Choosing grid resolutions

The choice of global and local grids resolution is determined by the distance from the viewpoint to the center of simulation space or to the center of the interactive object. It is easy to know the center position of simulation space, and one regards the center of top OBB of the interactive object as the center of the object. Previous data can be reused to decrease the amount of calculation, as shown in Fig. 7.

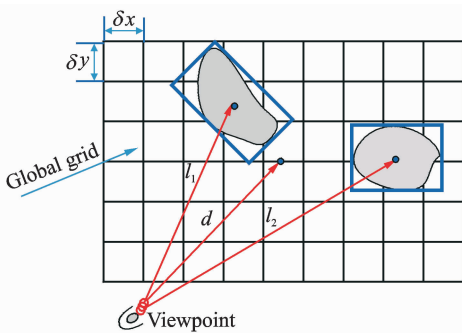


Fig. 7 Distance from the viewpoint to simulation space or to center of object

When the distance d between the viewpoint and the center of simulation space changes, the global grid resolution needs to be adjusted. When

projecting the grid on the computer screen, the grid cell of width h corresponds to $(h \cdot e)/d$ pixels, where e is a constant determined by camera parameters. When $(h \cdot e)/d = 1$, i. e. corresponding to 1 pixel, it can meet display requirement.

One can determine the size of 1 pixel from the known screen resolution and the screen size initially. Assuming that h_{\max} is the size of the grid cell at the highest resolution, according to $(h \cdot e)/d = 1$, h_{\max} is constant if simulation space is determined. Assuming that d_{\max} is the current distance from the viewpoint to the center of simulation space, then $d_{\max} = h_{\max} \cdot e$. Let d_x be the alterable distance between the viewpoint and the center of simulation space, one can get

$$M = \left\lfloor \log_2 \frac{d_x}{d_{\max}} \right\rfloor$$

where $\lfloor \cdot \rfloor$ represents the rounding down and M the change exponent of the global grid resolution. New resolution of global grid changes into $2^{(N-M)}$, where N is the exponent of the highest resolution obtained from the initial size of simulation space and h_{\max} . During the actual simulation, the highest resolution of global grid is set as $2^{(N-1)}$. In order not to change the global grid resolution frequently, one uses $M = \left\lfloor \log_4 \frac{d_x}{d_{\max}} \right\rfloor$ to ensure that the global grid resolution will change only when d_x becomes greater.

Changes of local grid resolution are adopted similarly as the way of global grid. Assuming that l_{\max} is the current distance from the viewpoint to the center of the interactive object, then $l_{\max} = h_{\max} \cdot e$. Assuming that l_x is the alterable distance between the viewpoint and the center of the interactive object, one can get

$$L = \left\lfloor \log_2 \frac{l_x}{l_{\max}} \right\rfloor$$

where L is the change exponent of the local grid resolution. New resolution of local grid changes into $2^{(N-L)}$.

From the adjustment of grid resolution, one can see that the automatic adjustment can be realized when the viewpoint changes or the interactive objects move. Manual adjustment or experimen-

tal determination is not required. While performing the fluid simulation, it can be done to move forward until the last cell along the x , y , z axis directions. If the cells are occupied, they are skipped directly without any processing to promote simulation efficiency, and instability in the simulation does not appear.

2 Solving Controlling Equations by Discretization

2.1 Fluid dynamics equations and their simplification

The clouds movement can be described by the N-S equations for an incompressible flow.

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{F} \quad (1)$$

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

where ρ is the fluid density, Re the Reynolds Number, and $\mathbf{F} = (f_x, f_y, f_z)$ the external force. ∇ the gradient operator, $\nabla \cdot$ the divergence operator, and ∇^2 the Laplacian operator. The first three terms on the right hand of Eq. (1) are the advection term, the pressure term, and the viscous term, respectively.

The advection term is an inertial effect, and the relationship between the effects of advection and viscous terms can be measured by Re . In the clouds simulation, Re is expressed as $Re = (u \cdot L)/\nu$, where u is the characteristic velocity, L the characteristic length, and ν the viscosity coefficient. The values of ν range from $1.3 \times 10^{-5} \text{ m}^2/\text{s}$ to $1.9 \times 10^{-5} \text{ m}^2/\text{s}$ when the temperature is $0-60 \text{ }^\circ\text{C}$ ^[15]. If the characteristic length L is 100 m or more, Re will be greater than 10^7 . This means that the inertial force is much greater than the viscous force, so the viscous term is negligible. N-S equation will be simplified as Euler equation of incompressible fluid.

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla p + \mathbf{F} \quad (3)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (4)$$

2.2 Discretization of equations and improvement

One uses the finite difference method for sol-

ving differential equations here, and the upwind difference method is introduced to solve the stability problem. Upwind difference is combined with central difference at different stages.

While solving the Euler equations, simulation area is discretized by using multi-resolution grids proposed in this paper, and different unknown variables are set at different cells. Pressure p and temperature T are at the center of cells, and the components of speed u , v , w are at the center of surfaces as shown in Fig. 2. While discretizing the motion equation, upwind difference is adopted for stability.

3 Illumination of Clouds

Realistic simulation of clouds is closely related to the illumination calculations, and the scattering of clouds particles meets the Mie scattering law. While rendering the clouds, multiple scattering illumination model is relatively close to physical characteristics of clouds. However, it is time-consuming, so one uses multiple forward scattering introduced in Ref. [6]. At the same time, the improved phase function is proposed.

3.1 Common phase functions

Rayleigh phase function in Eq. (5) is applicable to the case of Rayleigh scattering. Ref. [6] used the Rayleigh phase function and yielded good results, but there were some bright spots in some clouds, as shown in Fig. 8.

$$P(\theta) = \frac{3}{4} \frac{(1 + \cos^2 \theta)}{\lambda^4} \quad (5)$$

where λ is the wavelength of incident light and θ the phase angle.

Henye-Greenstein phase function proposed by Henye and Greenstein is useful in scattering calculation of biological organs, water, clouds and many other natural materials, and it is an approximation of Mie scattering, expressed as Eq. (6).

$$P_{\text{HG}}(\theta) = \frac{1 - g^2}{(1 + g^2 - 2g \cos \theta)^{3/2}} \quad (6)$$

where g is the asymmetric factor, controlling the



Fig. 8 Phenomenon of bright spots using Rayleigh phase function

redistribution shapes of the scattered lights.

Cornette and Shanks modified Henyey-Greenstein phase function and gave a more physically reasonable representation, Cornette-Shanks phase function, for clouds illumination expressed as Eq. (7). It was used in Ref. [16] while rendering cloud scenes.

$$P_{cs}(\theta) = \frac{3}{2} \frac{(1-g^2)}{(2+g^2)} \frac{(1+\cos^2\theta)}{(1+g^2-2g\cos\theta)^{3/2}} \quad (7)$$

3.2 Improved phase function

It can be seen that exponential computation is required in Henyey-Greenstein and Cornette-Shanks phase functions, thus increasing the computation time greatly. Though rayleigh phase function is simple, it is prone to the phenomenon of bright spots. To solve these problems, the improved phase function is proposed based on the Cornette-Shanks phase function.

As the scattering of clouds particles focuses on the forward scattering, the exponential item in Eq. (7) is simplified, and another adjustment item is added. One can obtain the following simplified phase function.

$$P(\theta) = \frac{3}{2} \frac{(1-g^2)}{(2+g^2)} \frac{(1+\cos^2\theta)}{(1+g^2-2g\cos\theta)} + g\cos\theta \quad (8)$$

It is equivalent to Rayleigh phase function while $g=0$. Table 1 lists the calculation time of different phase functions. The programming environment is VC 6.0, C language and the current hardware configuration. The calculation time of 100 000 times is a statistical period. One obtains the cosine values via the dot product of the incident light vector and the direction vector in the viewpoint direction.

Table 1 Calculation time of different phase functions

Phase function	Time
Henyey-Greenstein	0.009 905 532 686 417
Cornette-Shanks	0.010 388 694 652 533
The proposed method	0.000 714 062 186 958

Compared the calculation time of different phase functions in Table 1, there is no great difference between the calculation time of Henyey-Greenstein and Cornette-Shanks phase functions. The calculation time of the improved phase function decreases greatly. Fig. 9 shows the values of different phase functions when the asymmetric factor g is different.

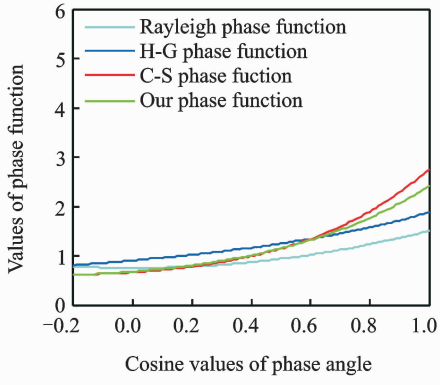
In Fig. 9, the proposed phase function approximates the fitting of Henyey-Greenstein and Cornette-Shanks phase functions. Especially when the value of g is small, it is closer to the average of these two functions. Experiments show that the rendering effects of clouds are perfect when the value of g is 0.3. Thus, one can obtain pretty rendering scenes just like Henyey-Greenstein and Cornette-Shanks phase functions by using the improved phase function. Luckily, it is faster and more suitable for real-time rendering.

4 Experimental Results and Discussion

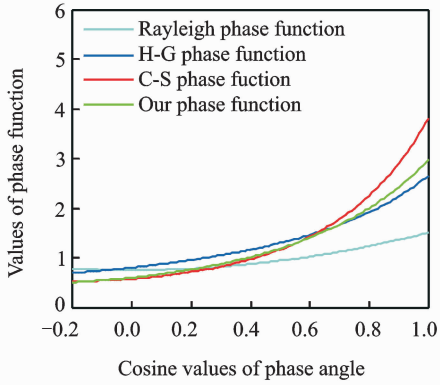
Clouds scenes are simulated by using the proposed methods. The computer is a desktop PC with Intel i5-2380P dual cores 3.10 GHz (CPU), 4 GB RAM and NVIDIA GeForce GTX 550 Ti (GPU).

4.1 Simulating effects

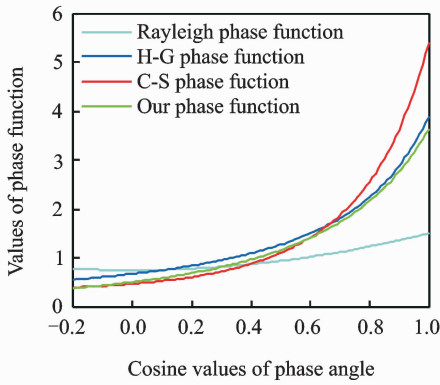
Fig. 10 depicts a single cloud rendered by the improved phase function, Fig. 11 the clouds scene in dawn, Fig. 12 the clouds scene of the daytime at beach, and Fig. 13 the large-scale clouds. There are no interactive objects in the above clouds scenes.



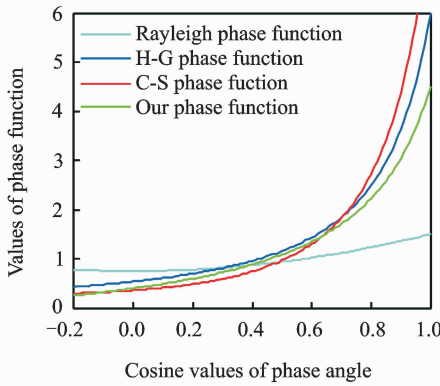
(a) $g=0.2$



(b) $g=0.3$



(c) $g=0.4$



(d) $g=0.5$

Fig. 9 Values of different phase functions



Fig. 10 Single cloud rendered by the improved phase function



Fig. 11 Clouds scene in dawn

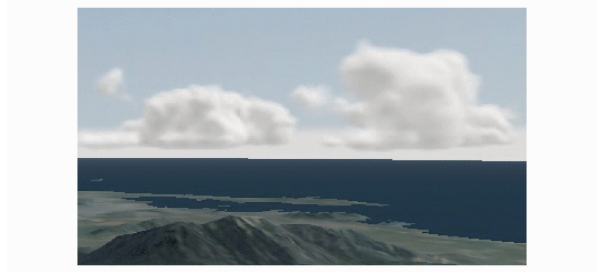


Fig. 12 Clouds scene of daytime at beach



Fig. 13 Large-scale clouds

From the above rendered clouds one can see that our clouds scenes have strong realism and eliminate the phenomenon of bright spots. While rendering large-scale clouds, the proposed methods have a great advantage of ensuring the real-time simulation because they can adjust global

grid resolution adaptively.

Fig. 14 shows the clouds scene with one interactive airplane, and Fig. 15 the clouds scene in which several aircrafts are involved in the interaction.



Fig. 14 Clouds scene with one airplane



Fig. 15 Clouds scene with several airplanes

4.2 Comparison and analysis of simulation time

During the clouds simulation, one verifies the methods of updating the resolutions of global and local grids proposed in the previous literatures, and compares them with the proposed methods.

The time of displaying the simulation results once is treated as one statistical period, which includes the time of simulation calculations and the interaction between fluid and objects. The running time under different grid resolutions is given in Table 2. In Table 2, grid resolution represents the highest grid resolution, where [GLOBAL] is the method of updating global grid resolution, applied in many references such as Refs. [8-10], [LOCAL] the method of updating local grid resolution [12-14], and [OUR] the method proposed in this paper. No statistics obtained suggests that the computation time is too long to achieve real-time rendering.

One can draw the following conclusions according to the data in Table 2. If the highest grid resolution is fixed, the time of [GLOBAL] oscillates greatly at different time steps because updating the grid resolution or not is not fixed, and the time of other methods changes slightly. As far as the consuming time is concerned, the method for updating local grid resolution is faster than that for updating global grid resolution, and the proposed method is faster than that for updating local grid resolution.

Table 2 Time comparison of different methods in different grid resolutions

Grid resolution	Method		
	[GLOBAL]	[LOCAL]	[OUR]
$128 \times 128 \times 128$	0.006 2	0.005 1	0.002 8
$256 \times 256 \times 128$	0.010 5	0.020 1	0.005 9
$256 \times 256 \times 256$	0.031 7	0.036 2	0.010 2
$512 \times 512 \times 256$	0.172 9	0.159 3	0.037 5
$512 \times 512 \times 512$		0.325	0.077 4
$1\ 024 \times 512 \times 512$			0.160 7

The reasons why the speed of the proposed method is faster mainly in the next aspects. One updates the resolutions of global and local grids according to the requirement, but the efficiency of [LOCAL] method is influenced by the initial resolution of the global grid. Although [OUR] method needs to update the global grid resolution, its updating frequency is much slower than [GLOBAL]. It is a compromise between these two methods. The combination and splitting of grid cells and data transfer are simple. Grid cells occupied by interactive objects are not involved in the simulation calculation. Since OBB hierarchical bounding boxes of interactive objects are established, high speed can be obtained while determining the occupation of grid cells. The improved phase function also promotes the efficiency of illumination calculations.

Frame rate statistics of rendering the above scenes are presented in Table 3. While counting

these data, the grid resolution does not exceed $512 \times 512 \times 512$. One presets the maximum frame rate to 120 f/s, and 120 f/s will be displayed if true frame rate is more than 120. As illustrated by the statistical data, the proposed method can

achieve the real-time rendering when the scale of the grid resolution is no more than $512 \times 512 \times 512$. When the higher grid resolution of simulation space is adopted, real-time rendering is hard to get, and artifact thus occurs.

Table 3 Rendering frame rates of clouds scenes in different grids resolutions

Different clouds scene	Single cloud	In dawn	Large-scale clouds	Clouds with one aircraft	Clouds with two aircrafts
256 × 256 × 128	120	120	106	109	103
256 × 256 × 256	120	112	79	84	81
512 × 512 × 256	120	55	30	33	32
512 × 512 × 512	93	33	16	18	15

5 Conclusions

The proposed multi-resolution adaptive grid method is simple and easy to implement. OBBs of interactive objects can promote the speed of intersection test between the interactive objects and the grid cells. According to the occupied flags of grid cells, the occupied grid cells are not involved in simulation calculations. The proposed method is faster than the method updating global grid resolution or local grid resolution only, and can implement real-time rendering of dynamic clouds and its interaction with other rigid objects. While changing the viewpoint or moving the interactive objects, the requirement of real-time rendering can be met. The simplification of motion equations and their discretization in upwind difference can promote efficiency and ensure stability. The improved phase function can accelerate the illumination calculations of clouds, but it does not reduce the realism of clouds scenes. However, there is the limitation that the number of grid cells in the axial direction is 2^n in the proposed multi-resolution grid method, and the problem will be addressed in the future research.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 61102167).

References:

- [1] Qiu Hang, Yang Ke, Chen Yu, et al. Survey on realistic simulation of cloud[J]. Computer Science, 2011,38(6):14-19. (in Chinese)
- [2] Pan Qiuyu, Bi Shuoben, Lu Lianghu, et al. Fast algorithm based on particle system for simulating 3D dynamic clouds[J]. Journal of System Simulation, 2014,26(1):85-89,106. (in Chinese)
- [3] Miyazaki R, Yoshida S, Dobashi Y, et al. A method for modeling clouds based on atmospheric fluid dynamics[C]// Proceedings of Pacific Graphics 2001. [S. l.]:[s. n.], 2001:363-372.
- [4] Miyazaki R, Dobashi Y, Nishita T. Simulation of cumiform clouds based on computational fluid dynamics[C]// Proceedings of Eurographics 2002 short presentation. [S. l.]:[s. n.], 2002:405-410.
- [5] Tang Zhao, Wu Pingbo. Real-time modeling and rendering of 3D cloud and its application in industrial simulations[J]. Journal of Computer-Aided Design & Computer Graphics, 2007,19(8):1051-1055. (in Chinese)
- [6] Harris M J. Real-time cloud simulation and rendering [D]. Chapel Hill: University of North Carolina, 2003.
- [7] Ren Wei, Liang Xiaohui, Ma Shang, et al. A real-time simulation method for large-scale 3D clouds[J]. Journal of Computer-Aided Design & Computer Graphics, 2010,22(4):662-669. (in Chinese)
- [8] Fedkiw R, Stam J, Jesen H W. Visual simulation of smoke[C]// Proceedings of SIGGRAPH'01. [S. l.]: [s. n.], 2001:15-22.

- [9] Carlson M, Mucha P J, Turk G. Rigid fluid: Animating the interplay between rigid bodies and fluid [J]. *ACM Trans on Graphics*, 2004,23(3):377-384.
- [10] Guendelman E, Selle A, Losasso F, et al. Coupling water and smoke to thin deformable and rigid shells [J]. *ACM Trans on Graphics*, 2005,24(3):973-981.
- [11] Yang Yang, Yang Xubo. Staggered grid structure for smoke simulation[J]. *Journal of Image and Graphics*,2014,19(5):781-788. (in Chinese)
- [12] Dobashi Y, Matsuda Y, Yamamoto T, et al. A fast simulation method using overlapping grids for interactions between smoke and rigid objects[J]. *Computer Graphics Forum*, 2008,27:477-486.
- [13] Lentine M, Zheng W, Fedkiw R. A novel algorithm for incompressible flow using only a coarse grid projection [J]. *ACM Trans Graph*, 2010,29:114.1-114.9.
- [14] English R E, Qiu L, Yu Y, et al. An adaptive discretization of incompressible flow using a multitude of moving Cartesian grids[J]. *Journal of Computational Physics*, 2013,254(12):107-154.
- [15] Griebel M, Dornseifer T, Neunhoeffler T. Numerical simulation in fluid dynamics; A practical introduction [M]. Philadelphia: SIAM Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics, 1998.
- [16] Miyazaki R, Dobashi Y, Nishita T. A fast rendering method of clouds using shadow-view slices[C]//*Proceedings of International Conference on Computer Graphics and Imaging (CGIM'04)*. Anaheim: ACTA Press, 2004:93-98.

(Executive editor: Zhang Tong)

