Raining Simulation based on Improved Particle Systems algorithms

YE Li-na
Computer center of public course，Wuhan Institute of Shipbuilding Technology
WSPC
Wuhan，China
e-mail：504292159@qq.com

Abstract—To solve the real-time property trouble of the generation of the virtual Reality scene with the traditional particle systems algorithms. Firstly, the traditional particle systems technique is introduced in the paper, then an improved algorithms of particle systems which is used in the raining simulation is brought forward, and in the end an vivid raining visual effect is obtained. The improved particle systems algorithms is proved to be very simple and useful，and it is can be conveniently introduced into the various kinds of cartoon system.

Keyword Words- Computer simulation；Particle Systems；Raining Simulation；Real-time Generation；Algorithms (key words)

I. INTRODUCTION

With the rapid development of computer graphics theory and graphics hardware to generate and simulate the natural landscape has become one of the hot topics in computer graphics research, especially for rain, snow and other natural features dynamic simulation is more challenging，this is due to the surface shape in the real world of natural scenery has extremely rich surface texture details and irregular, appearance their logical structure is difficult to express, if the conditions of the scene in real-time computer generated using the traditional approach to computer graphics modeling, rendering and then implemented on the scene, a huge amount of calculation will be generated graphic display insurmountable bottleneck. So how to enhance the realness and real-time performance of the natural spectacle has become a very important content of the computer graphics.

In consideration of the visual simulation of real-time requirements, firstly the traditional algorithm based on particle system is introduced, and then to simulate the natural rainfall, for example, the properties characteristic analyzing of rain is proposed based on particle system in real time simulation of natural rainfall improved algorithm, which can establish and implement a process to simulate rainfall particle system, in which the physical models were simplified during the rainfall simulation to realize both the realness and real-time performance. In fact the algorithms can be easily embedded into a variety of animation. Which will have a extensive prospects.

II. RAIN SCENE SIMULATION ALGORITHM BASED ON TRADITIONAL PARTICLE SYSTEM

Assume that the system model space coordinate system using the right-hand coordinate system, the initial position of the camera is (0,0,0), and the direction is the z-axis negative direction. The variables used including the intermediate body parameters, window size, the camera parameters. Intermediate body parameters include: horizontal viewing angle and vertical viewing angle are α and β, nearly horizontal distance is dnear, far plane distance is dfar; Windows Size: windows height is hscreen, windows width (h,p,r) is wscreen; The current camera position is (xv ,yv ,zv); camera angles around x, y and z coordinate axis is (h,p,r), since during the simulation h and r approximately equal to zero, therefore can be simplified to (0,p,0).

A. Calculation of the number of particles

In the rain scene simulation, new particles are produced by a controlled random process produces, one moment, the number of particles entering the system is the difference between the number of particles default in the system and the number that currently exists in the system, plus a number of particles defined by a stochastic process[1-2]. Formula below:

$$\begin{aligned}&NewParticleNum = ExpectedNum - ActiveParticleNum +\\&R_0 \times C_0 \times ExpectedNum\end{aligned} \tag{1}$$

Among them：ExpectedNum is the number of particles expected value, ActiveParticleNum is the current number of particles, R0 is a random number [-1, 1] among uniformly distributed, C0 is the number of particles randomly varying parameter range set by the user, for controlling the actual number of new particles generated. When NewParticleNum is less than zero, the system does not add new particles.

B. Setting the initial properties of new particles

Particle system needs to determine the initial position of the new particle according to the parameters of the current viewpoint position and intermediate body parameters[3]. Because the system is real-time system, to minimize cost, taking into account the line of sight direction of a game character is generally parallel with the horizontal features, the new particles generation region (Fig .1) is defined as a three-dimensional model space area S.

Suppose dmax be a maximum visible range under the current screen resolution and body parameters of the particle horizon distance, dnear is one of the intermediate body parameters specified by the user; dmax is defined as a particle projection height in pixels on the screen as a particle distance from the viewpoint[4], which is calculated as follows:

$$d_{\max} = h_{screen} \times \frac{h_p}{2 \times tg(\beta / 2)} \tag{2}$$

Wherein: hp is the particle height, uint (m), for ease of calculation, in order to establish the origin of the viewpoint spherical coordinate system (r,φ,θ), model space coordinate system and the coordinate system conversion relationship is as follows:

$$x = x_v + r \sin\varphi \sin\theta \tag{3}$$

$$y = y_v + r \cos\varphi \tag{4}$$

$$z = z_v + r \sin\varphi \cos\theta \tag{5}$$

In the spherical coordinate system, the coordinates of the initial position of the particle i is[5]:

$$r_i = d_{near} + (d_{\max} - d_{near}) \times R1 \tag{6}$$

$$\varphi_i = (\pi - \beta)/2 + R2 \times \beta / 2 \tag{7}$$

$$\theta_i = p - \pi - \alpha / 2 + \alpha \times R3 \tag{8}$$

Wherein R1, R2, R3 is a random number uniformly distributed between [0, 1], when they are substituted into equation (4). The model space coordinates of the initial particles can be got. For distances greater than from the point of view of the rain of particles, due to its screen height less than one pixel, has little effect on the visual effects, therefore not to be drawn at the time of rendering.

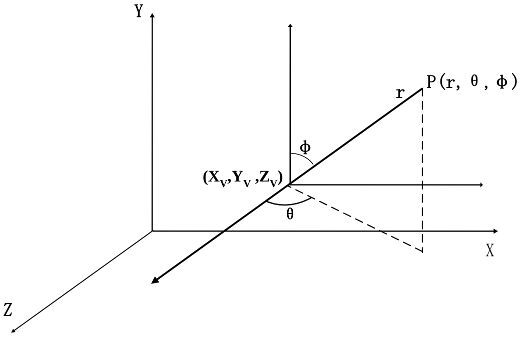

Figure 1. Spherical coordinate system definition.

When the initial position of a new particle is determined, its initial velocity and acceleration also should be assigned. The initial velocity of particles is determined by the current wind speed, wind direction, and stochastic processes. The wind speed set as v_w, direction of η. Rain initial velocity of particles (m • s-1) is calculated as follows:

$$v_{rx0} = v_w \cos\eta \tag{9}$$

$$v_{ry0} = -10.0 \tag{10}$$

$$v_{rz0} = v_w \sin\eta \tag{11}$$

Rain particle acceleration a is:

$$a_{rx0} = a_{rz0} = 0.0 m \cdot s^{-2} \tag{12}$$

$$a_{ry0} = -9.8 m \cdot s^{-2} \tag{13}$$

C. *State of the particle update*

Particle attributes required frame update. The attributes to be updating include: location, speed[6]. Particle X direction, the Z-direction velocity are affected by current wind speed and direction. When the wind speed changes, the particle X direction, the corresponding change in velocity in the Z direction, the same calculation method with the formula (11). In particle system, rain and snow particles should move in three-dimensional space. In accordance with the movement low of the object, the object acceleration of particles in a, velocity v and position s have the following constraints:

$$v = v_0 + \int a \, \mathrm{d}_t \tag{14}$$

$$\mathrm{s} = \mathrm{s}_0 + \int a \, \mathrm{d}_t \tag{15}$$

D. Lifecycle and condition

Lifecycle of particles often in units of frames. Particle position is outside the body or particle horizon fell to the ground as a condition of particles to end its life. In the 3D system, due to the complexity of the scene, the time cost for this operation is large, it is difficult to implement in real time. Therefore, when the conditions are simplified vertical y coordinate values of particles smaller than the ground vertical plane coordinate values, particle life cycle ends[7].

E. Particles drawing

When the state of the particle is set up, the state of the particles is needed to draw them.

In summary, the life cycle of the whole particle system process is shown in Fig .2 as follows:

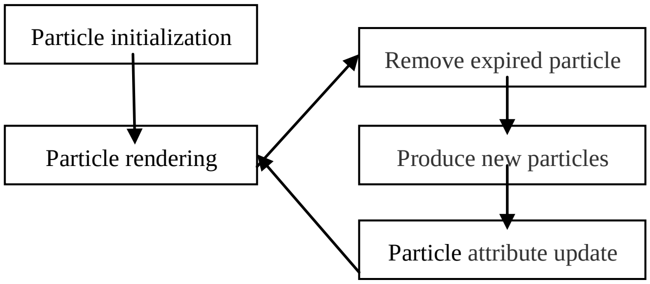

Figure 2. Particle system structure flowchart

III. IMPROVED ALGORITHM RAIN SCENE SIMULATION SYSTEM BASED ON PARTICLE

Since a large number of particle unit are used to express irregular objects, thus even for a small space in the body region, the number of particles constituting it is amazing, traditional particle system algorithm previously described ask for a excessively high storage and calculation capacity of the computer, so that the graphical display to implement real-time interactive computer environment in general is difficult, slow, can’t meet the requirements of real-time rendering 3D scenes. Therefore, to overcome the slowness of the traditional particle system algorithm, traditional rain scene simulation algorithm was optimized and improved in the following several aspects in order to achieve real-time response of the system in the paper:

A. Data structures simplified

The more real the natural scenery is simulate to be, the more simulate to be number of particles are required, the greater the computational cost is needed. Therefore, the design of a good particle system data structure is very important, otherwise it will greatly reduce the speed of the particle system. In order to reduce as much as possible the rendering time of these particles, while based on the characteristics of rainfall scenario, when the rain scene simulation particle system modeling, the short-term (is a line) is used to represent the rain of particles, according to the passage of time, according to two direction of the vector vertex of the decision, according to a certain rate of decline. And its specific position is calculated subsequently, then in this position the particles are drawn. Rain particle structure is constructed as follows:

```
// Rain particle structure
struct RainParticle
{
 float x, y, z;        // Location
 float fDspeed;        // The rate of decline
};
RainParticle   rain[PARTICLENUM];   // Rain particle arrays
```

Of course, the rain of particles can also be used to represent with a small quadrangle, etc., but when the quadrilateral is used, the texture should be added, which will greatly increase the cost of the entire particle system operation. Besides using segments represent raindrops also eliminate anti-aliasing process cumbersomely.

B. Simplify the relative position of the recognition model

By contrast to the scene scenery, the size of the particles are so small, that the particle horizon body position in three-dimensional space can be simplified to "point" of crop problems. First calculate body composed of six surface horizon body parameters according to the horizon, the horizon of its normal vector pointing to the inside of the body. By calculating the line of sight of the six sides of the body and the center of the particle disi(i=1,2,…6), can determine the position relationship of the six surfaces and the particles. If the particles are judged inside of all the six sides, then it is within the line of sight of the body, then it is subsequently needed to judge whether the particles are landed in order to determine whether the end of its life cycle; yet if the particles stand outside any one side, then the life cycle of the particle ends.

To determine the particle falls on the floor or other landscape scene, you need to traverse all the fragment space model of the current scene, and the whereabouts of the particle trajectories should intersect. In the 3D system, due to the complexity of the scene, the time cost for this operation is large, it is difficult to realize in real time. Therefore, the algorithm condition is simplified when the vertical y coordinate values of particles smaller than when they are in the vertical position to that of the ground plane coordinate values, particle life cycle ends.

C. Particle system simplified

During the design of the particle system, from "heavier force field simulations, wind simulation" to other environmental flame, waves, waterfalls, fountains, branches swing, and both are derived from the physical model. Using (2) and (3) particle motion state quantity calculating integral calculation is necessary, for which we use methods such accelerated motion of the object to simplify the motion state calculated discrete case, the algorithm is:

Assuming the particles in the life cycle, acceleration remains unchanged. Y direction velocity of particles rain affected only by its initial acceleration. Due to the time interval between frames is small, in order to simplify the calculation, the particle motion between frames is regarded as uniform motion. The current position of the particle i is:

$$x_i = x_{i0} + \sum_{m=1}^{m=N} v_{xm} T_m \qquad (16)$$

$$y_i = y_{i0} + \sum_{m=1}^{m=N} v_{ym} T_m \qquad (17)$$

$$z_i = z_{i0} + \sum_{m=1}^{m=N} v_{zm} T_m \qquad (18)$$

Wherein Tm is the time interval from m-1 frame to the m frame, (vxm, vym, vzm) is the particle velocity m for the age of the m, N is the current age of the particles (in units of frames). The accelerated motion method can meet both the accuracy and the simplicity performance. On the other hand, higher acceleration efficiency of recursive method can be used to position and velocity computation to calculate the adjacent particles.

D. Location Promotion Strategy

In this paper, all the particles of the particle system implemented in the program are randomly generated initialization time, when the particles fall to the ground like a traditional particle system is not as immediately freed memory occupied by the particles, and then allocating new memory and a certain amount of re-emitted particles complement, rather simply to enhance its position to the highest point, the benefits of doing that:

1) Avoiding death and generation of the dynamic particles, eliminating the need for the operation to free the memory of the death of particles, also saving a lot of time to reallocate memory for new particles;

2) Avoid fragmentation that is caused by frequent allocation of free space memory;

3) Making the data structure eliminates some of the value of life parameters (in this case to determine whether the vertical coordinates of the particles exceed the scene), the particles data structure become more compact to be dealed with.

Promotion Strategy introduction position greatly improved the performance of the program running.

E. Strengthening depth perception algorithm

In the virtual scene, for participants to achieve "virtual reality" feeling, complex three-dimensional depth perception of the natural scenery is extremely important. Although under normal circumstances, for the descent of raindrops, converted to a perspective transformation in accordance with a display plane and the three dimensional environment simulation rendering can realize certain degree of depth perception effects; however in this paper, the method of the multilayer film to enhance the depth of simulated rain point information, that the size of raindrops to be smaller, and the distance to the moving speed of raindrops more slowly, to give a stronger sense of depth.

IV. CONCLUSIONS

From the angle of real-time and realistic performance of the simulation system, firstly the traditional particle system simulation algorithms rain scene were studied, followed by particle properties simplifying, rainfall simplified physical model, location, promotion strategies and depth perception strengthened algorithm, making raindrops falling particles generated much faster, which can ensure both the real-time performance and the reality a lot, the effect screenshots is shown in Fig .2, it is proved that the technology has a strong application background, and it can be easily integrated into a variety of three-dimensional animation systems.

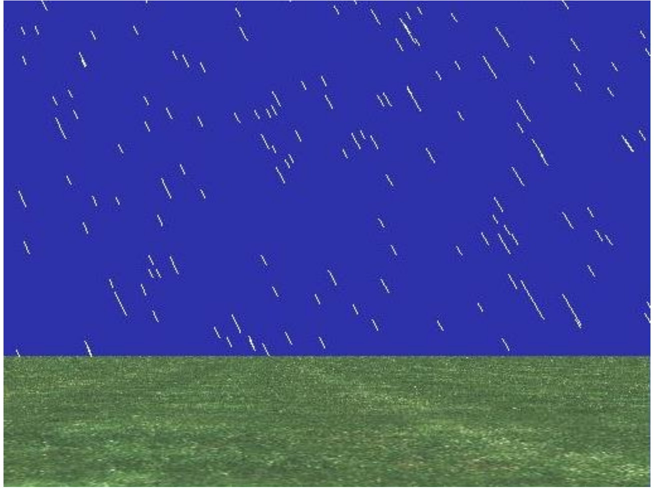

Figure 3. Rain scene simulation based on improved particle system algorithms

REFERENCES

[1] Coutinho B B, Oliveira A A F, Atencio Y P, et al. Ｒain scene ani-mation through particle systems and surface flow simulation by SPH［C］// 23rd SIBGＲAPI, 2010: 255- 262.

[2] Tatarchuk N. Artist- directable real- time rain rendering in city envi-ronments［C］//ACM Siggraph Courses, 2006: 23- 64.

[3] Ding W, Zhu Z, Chen X, et al. Real- time rain and snow rendering［C］// Second International Conference on Agro-Geoinformatics, 2013: 32- 35.

[4] Sun B, Ｒamamoorthi Ｒ, Narasimhan S G, et al. A practical ana-lytic single scattering model for real time rendering [J]. ACM Transactions on Graphics, 2005, 24 (3): 1040- 1049. .

[5] Strube de Lima D, Braun H, Raupp Musse S. A Model for Real Time Ocean Breaking Waves Animation [C]// Proceeding of 2010 Brazilian Symposium on Games and Digital Entertainment. USA: IEEE, 2010: 19-24.

[6] Kass M, Miller G. Rapid, stable fluid dynamics for computer graphics [C]// SIGGRAPH '90 Proceeding of the 17th annual conference on Computer Graphics and Interactive Techniques. USA: ACM, 1990: 49-57.

[7] Wang Shunli, Kang Fengju, Wang Dinghua. Ocean wave real-time simulation based on adaptive fusion [C]// Proceeding of the 32nd Chinese Control Conference (CCC). USA: IEEE, 2013: 8557-8560.

[8] Dobashi Y, Nishita T, Yamashita H. Using metaballs to modeling and animate clouds from satellite images [J]. The Visual Computer (S1432-2315), 1999, 15(9).:

[9] Fan N, Zhang N. Real- time simulation of rain and snow in virtualenvironment［C］// International Conference on Industrial Control and Electronics Engineering, 2012.

[10] Wang B, Peng J L, Kuo C J. Cumulus cloud synthesis with similarity solution and particle/voxel modeling [C]// Proceedings of the International Symposium on Visual Computing’08. Berlin, Germany: Speringer-Verlag, 2008.

[11] Enright D, Marschner S, Fedkiw R. Animation and rendering of complex water surfaces [J]. ACM Transactions on Graphics (S0730-0301), 2002, 21(3).

[12] Obashi Y, Yamamoto T, Nishita T. A Controllable Method for Animation of Earth-scale Clouds [C]// Proceedings of CASA’06. Chichester, USA: John Wiley and Sons Ltd. 2006.