Modeling and Simulation Based on Improved Particle System

Wenqiao Zhang & Jing Zhang
Harbin Engineering University, Harbin, China

ABSTRACT: Particle system is an Effective method to simulate the irregular objects which has characteristics of flexibility and adaptability, this paper take in-depth study for particle system, introducing the idea of genetic algorithm, made the improvement of traditional simulation model. The experimental results show that the simulation of improved model is vivid under the complex environment, its real-time characteristics and fidelity is satisfied the requirements.
KEYWORD: Particle system; model of simulation; particle swarm; genetic algorithm

1 INTRODUCTION

In recent years, simulation technology has become one of the hot topic, and the balance of 3D simulation of real-time and realistic is one of the difficulties in the field of visual simulation (D. A. Gladkiy et al, 2009) (HUA Zexi et al, 2012). In 1983, Reeves first proposed particle system, the basic principle is based on the large scale particles to simulate the objects of irregular shape (J Varela et al, 2013) (Mahesh Prakash et al, 2015), and use the changing over time particle properties , to realize the dynamic effects of realistic simulation.

With take in-depth study for the model of simulation, it inspired by the idea of genetic algorithm (Reeves, W.T. 1983) and provides a improved modeling method, using the particle swarm instead of the method of texture mapping of particle to improved the simulation in the complex environment, it uses the unity3d as a development tool to simulate, enhancing the fidelity of appearance and shape.

2 PRINCIPLE OF PARTICLE SYSTEM

The particle system is not a simple static model, but a model is dynamically computed. Typically, Particle system includes several steps (Wang Yinling et al, 2006):

1) Generating particles: the first step is to deal with the number of particles.

2) Initialize properties of particles: including pretreatment of size, shape, color, transparency, life and other attributes of particle.

3) Remove the "demise" of the particle: the particle has its own life cycle, the step is used to detect the life cycle of each particle, if the life cycle is zero, it is necessary to remove the particle from the system.

4) Draw particle: the previous steps is the dynamic process of particles, in order to show the movement of the trajectory and the changes in the properties of particles, it is required to draw each frame of the particle on the screen.

5) Repeat 2), 3), 4), 5), the process of a particle system is formed.

3 TRADITIONAL MODEL OF SIMULATION

3.1 *Definition of attribute of particle*

In the original simulation, it uses a N-dimensional Vector V_N to represent all the attributes of particle, expressed as (W.T. Reeves et al, 1985):

Vn={Speed,Acceleration,Position,Color,Size,Transparency,Flag,Life,Fade}

3.2 *Initialization of particle*

3.2.1 *Location of particle*

In order to a vivid simulation, it simulates the particle system produce a large number of particles in a limited range of sphere in a frame, the related expressions are as follows:

Center of the sphere:

$$P_c=\{X, Y, Z\} \quad (3\text{-}1)$$

Location of i-th particle:

$$Pi= \{PositionX_i, PositionY_i, PositionZ_i\} \quad (3\text{-}2)$$

$$PositionX_i=rand(-1,1)*|X| \quad (3\text{-}3)$$

$$PositionY_i=rand(-1,1)*|Y| \quad (3\text{-}4)$$

$$PositionZ_i=rand(-1,1)*|Z| \quad (3\text{-}5)$$

Limitation of location of i-th particle:

$$\sqrt{(X_i-X)^2+(Y_i-Y)^2+(Z_i-Z)^2} \le R \quad (3\text{-}6)$$

3.2.2 *Number of particle*

The number of particles to decide the degree of fidelity and affect good or bad for the realistic simulation, the total number of particle SPart should satisfy with the expression:

$$SPart=APart+rand(-NPart,Npart) \quad (3\text{-}7)$$

Apart is the number of particles which is settle for the result of realistic simulation, it needs lots of experiments to get, Npart is a random number to meet the diversity of particle.

3.2.3 *Appearance of particle*

In the real world, an object usually use the primary colors (red, green, blue) to represent their color, on the basis of primary colors, the particle system introduce the concept of transparency (Alpha), so the initial particle use components (R, G, B, A) to represent its color IntialColor (R,G,B,A).The expression (YE Lina 2012):

$$IntialColor(R,G,B,A)=MeanColor(R,G,B,A)+Rand(-1,1)*VarColor(R,G,B,A) \quad (3\text{-}8)$$

MeanColor (R, G, B, A) is the mean value of color in the particle system, VarColor (R, G, B, A) is the color variance in the particle system.

3.2.4 *Velocity of particle*

The initial velocity (size and direction) of the particles is a rigid indicator for simulation:

$$InitialSpeed_x= Rand()*VarSpeed_x \quad (3\text{-}9)$$

$$InitialSpeed_y= Rand()*VarSpeed_y \quad (3\text{-}10)$$

$$InitialSpeed_z= Rand()*VarSpeed_z \quad (3\text{-}11)$$

VarSpeed is a system parameters in the particle system, which represents the random velocity in a controlled range.

3.3 *Update of attribute of particle*

The attribute of particle is under the combined force, including field of gravity, buoyancy, air resistance, is constantly updated all the time in the particle system.

From the microscopic point of view to analyze the motion state of a particle, and assuming the current frame is f, the previous frame is f-1, time-lag is Δt, if it sets velocity attribute is $Velocity_f$, spatial attribute is $Position_f$, appearance attribute is $Color_f$, in frame f, their attributes shall conform to the expression as follow (Ying Tan et al, 2013):

Updated velocity:

$$VelocityX_f=VelocityX_{f-1}+AccelerationX_{f-1}*\Delta t; \quad (3\text{-}12)$$

$$VelocityY_f=VelocityY_{f-1}+AccelerationY_{f-1}*\Delta t; \quad (3\text{-}13)$$

$$VelocityX_f=VelocityX_{f-1}+AccelerationX_{f-1}*\Delta t; \quad (3\text{-}14)$$

Acceleration is the acceleration of particle under the combined force, including field of gravity, buoyancy, air resistance etc (Y. Yin et al, 2002).

Updated location:

$$PositionX_f=PositionX_{f-1}+VeclocityX_{f-1}*\Delta t \quad (3\text{-}15)$$

$$PositionY_f=PositionY_{f-1}+VeclocityY_{f-1}*\Delta t \quad (3\text{-}16)$$

$$PositionZ_f=PositionZ_{f-1}+VeclocityZ_{f-1}*\Delta t \quad (3\text{-}17)$$

Updated appearance:

$$ColorX_f=ColorX_{f-1}+ColorDeltaX_{f-1}*\Delta t \quad (3\text{-}18)$$

$$ColorY_f=ColorY_{f-1}+ColorDeltaY_{f-1}*\Delta t \quad (3\text{-}19)$$

$$ColorZ_f=ColorZ_{f-1}+ColorDeltaZ_{f-1}*\Delta t \quad (3\text{-}20)$$

ColorDelta represents the change of color and transparency in the unit time.

3.4 *Death of particle*

There are three kinds of reasons for the death of particles:

1) The particle is recognized as death when its lifetime is over.

2) When the particle located outside of the screen or predetermined range in the three-dimensional space, the particle is considered as death in order to reduce the unnecessary computation.

3) If the color and transparent of particle reaches to the minimum value or the background color for a long time, the particle is recognized as death although other properties of particle are still in the normal range.

4 IMPROVED MODEL OF SIMULATION

The simulation generated a large number of particles when the bombing started in traditional particle system, and the memory space, CPU occupancy rate is possible too high ,and the initialization of velocity and direction of the large number of particles is random, although it under control but is easy to cause the irregular shape when it exploded, on the other hand, the fidelity is poor if run the traditional particle model which uses the texture mapping method to determine the shape of particle.

This paper is based on the traditional model of particle and introduced into the idea of genetic algorithm, in the initializing stage, it produces only N_0 particles (this value is far less than the final numbers of particles), and with the adaptive analysis of the weight of velocity and location for the N_0 particles in particle system according to the random velocity and location, the fitness value $f(N_i)$ ($Ni \in N_0$)shall conform to the expression as follow:

$$f(N_i) = VWeight \times (Velocity_i - AVelocity) + PWeight(Position_i - APosition), i < N_0 \qquad (4\text{-}1)$$

$$0.6 < \mathrm{VWeight} / \mathrm{PWeight} < 0.8 \qquad (4\text{-}2)$$

VWeight is the weight of velocity, PWeight is the weight of location, $f(N_i)$ is the fitness value of i-th particle and the value less, the better. On the other hand, the weight of the space location is larger and the ratio of 0.75 is appropriate through multiple experiments.

4.1 *Combining the idea of genetic algorithm*

According to the obtained fitness of particle, retention of 40%~50% of particles which has better fitness to produce the next generation of particles, each preserved particle will generate a small particle swarm is defined as:

$$S_i = m \frac{Y_{\max} - f(N_i) + \varepsilon}{\sum_{i=1}^{N_0}(Y_{\max} - f(N_i)) + \varepsilon} \times \frac{(T_f - T)^2}{(\mathrm{T_f} - T)^4 + rand(1,10)} \qquad (4\text{-}3)$$

S_i is the number of particles in frame T_f produced by the i-th particle, T is the frame when the initialization of particle swarm, m is a constant to limit the total number of particles, its value according to the number of N_0 to set, Y_{max} is the worst fitness value, ε is a minimal constant to prevent the denominator is 0, rand (1, 10) is a random number between 1 and 10, in order to show the diversity.

In the particle system, in order to reduce the excessive computation caused by particle swarm, it uses the form of point-based rendering to screen, each particle can only be described as a vertex vector and reduce the computation greatly.

4.2 *Appearance of particle*

In order to ensure the simulation of improved model is better, it sets the particles retained by genetic algorithm as a emitting surface which is a circle with radius changed by time and symmetrically launch particle, each particle keeps the distance of a particle radius, In order to reduce computational complexity, the collision detection is no need, and the radius defined as:

$$\begin{cases} R_i = k\sin(rand(0,1) \times (T_f - T)), T_f < T_c & (4-4) \\ R_i = p\dfrac{(T_f - T)^2}{(\mathrm{T_f} - T)^4 + rand(1,5)}, T_f \geq T_c & (4-5) \end{cases}$$

R_i is the radius of the emitting surface generated by i-th particle in frame T_f, T is the frame when the initialization of particle swarm, k, p is a constant to limit the length of R_i, T_c is the frame when the two formulae is equal, rand (1, 5) is equal to expression (4-3), the sketch map of the relationship between T_f and R_i is Figure.1, the sketch map of emitting surface is Figure.2.

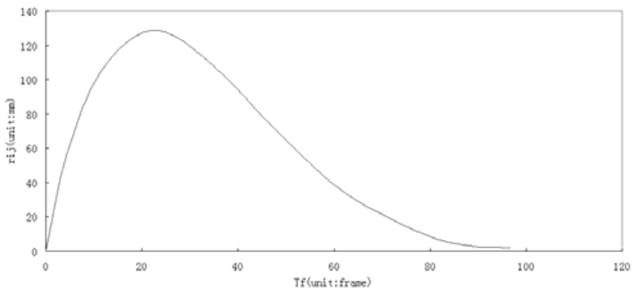

Figure.1 Relationship between T_f and R_i

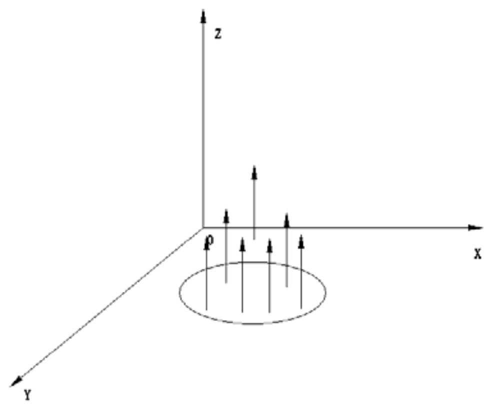

Figure.2 Sketch map of emitting surface

Moreover, the improved model set the brightness of each particle which is inversely proportional to the distance of the location of particle to the center of emitting surface, the expression as follow:

$$B_{i0} = l\sin\left(\frac{(R_i - r_{ij}) \times \pi}{2Ri}\right) \times B_{i0} + B_{\min}, j < S_i \qquad (4-6)$$

B_{ij} is the brightness of the j-th particle in the particle swarm, r_{ij} is the distance of the location of j-th particle to the center of emitting surface in the particle swarm, R_i is radius of the emitting surface, B_{i0} is a brightness of center of emitting surface and a constant brightness, l is a constant and B_{min} is a constant brightness to limit the brightness too bright or too dark, the sketch map of the relationship between B_{ij} and r_{ij} is Figure.3.

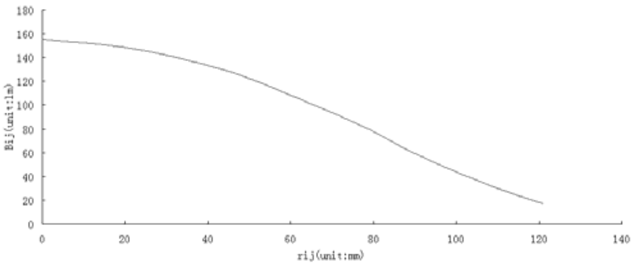

Figure.3 Relationship between B_{ij} and r_{ij}

4.3 *Displacement range of particle*

To further the shape of the simulation of improved model is higher, it sets the amplitude of explosion according to the fitness value of particles, the expression as follow:

$$A_i = q\frac{f(N_i) - Y_{\min} + \varepsilon}{\sum_{i=1}^{N_0}(Y_{\min} - f(N_i)) + \varepsilon} \qquad (4-7)$$

A_i is the displacement range of particle swarm generated by i-th particle, q is a constant to limit the displacement range of particles, its value according to the number of N0 to set, Y_{max} is the best fitness value, ε is equal to expression (4-3).

In the initializing stage, the model needs no set the life cycle of particle, just need to notice the particle begin to disappearing when the displacement of particle beyond displacement range A_i. This method can guarantee the authenticity of simulation is higher, on the other hand, it can improve the efficiency of CPU without calculate the time when the particle was die at all times.

5 IMPLEMENTATION

The model use the unity3d engine to implement and set the gravity field, spatial buoyancy field, air resistance field in 3D virtual space to simulate a complex environment, and it identifies the acceleration property of particle according to force calculated by Newton's second law.

In this model, it use the pointolite for elementary particle which is different from the traditional model that use the technology of texture mapping, and it ignores calculations of collision of each particle in the virtual environment, to further reduce the computation of CPU, it uses the form of point-based rendering to screen, each particle can only be described as a vertex vector.

In the improved model of simulation the particle released in a random initial position and controlled by all kinds of field parameters to simulate the motion locus, it starts to explode in a limited range of sphere when the velocity of initial particle is zero, the flow chart after the explosion is defined as Figure.4:

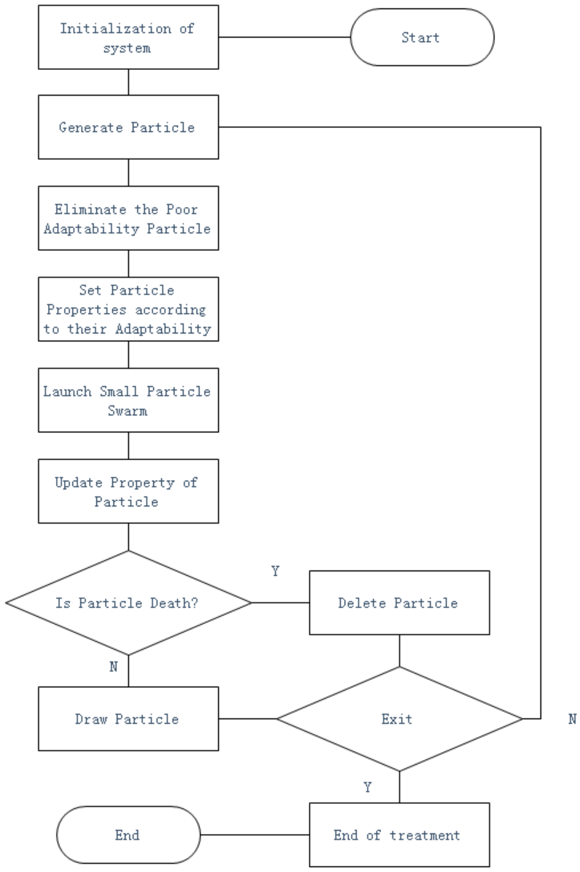

Figure.4 Flow chart of improved model

6 RESULTS

Based on the improved model, using the Unity3d engine on ordinary PC (I3 3220,3.3GHz, 4G of memory, gtx650 graphics) to implement the fireworks explosion and wave, Figure.5 and Figure.6 is the traditional model of simulation of fireworks and wave, Figure.7 and Figure.8 is improved model of simulation of fireworks and wave.

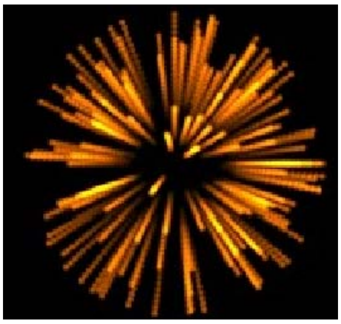

Figure.5 traditional model

Figure.6 improved model

Figure.7 traditional model

Figure.8 improved model

Seeing from the above Figure, the traditional model of simulation particle using matrix texture mapping methods and effect of simulation is poorer, moreover, it have not shown the physical properties of particle under the complex environment (gravity field, spatial buoyancy field, air resistance field), In contrast ,the result of improved model of simulation is realistic and better than the traditional model.

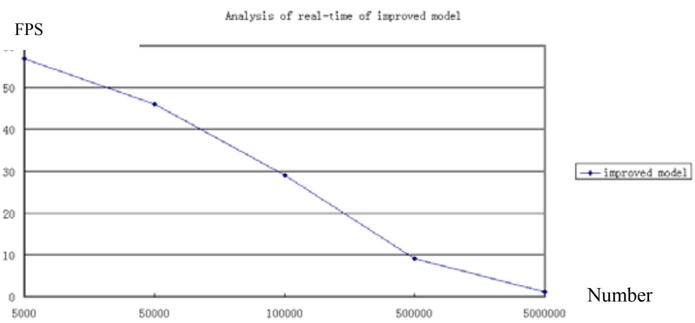

Figure.9 Analysis of real-time improved model

Seeing from above Figure.7, according to the analysis of improved model, it easy to see the result that adapt to the real-time, and through multiple experiments, the simulation results have been better when the number of particles in the 5000level, the FPS between 50 and 60 is far beyond the basic needs of FPS30.

7 CONCLUSION

This paper goes deep into analyzing the traditional model of fireworks and wave, on the basis proposed an improved model of fireworks and wave which introduces the idea of genetic algorithm, absorb the advantage of traditional model, and it obtains a series of simulation results on the Unity3d engine, moreover, we analyze the inadequacy of traditional model and improved it on the initializing stage ,the updated stage of attribute of particle to further effectively enhance the simulation of fidelity and visual sense.

REFERENCES

D. A. Gladkiy, I. V. Belago, N. A. Elykov, S. A. Kuzikowski. GPU processing of particle system animation[J]. Optoelectronics, Instrumentation and Data Processing, 2009, 456:.

HUA Zexi, WANG Yingchun. “Study of Explosion Simulation Based on Particle System”. Computer Science..no.4, vol.39, pp.278-281, 2012.ZHOU Shumin, SUN Yamin, LU Ling, CHEN Zhifeng.”Fire Simulation

J Varela, CG Soares. Multi-grid discretization of the wave spectrum for the real-time numerical simulation of the sea surface in computer graphics[J]. IEEE Computer Graphics & Applications, 2013(1):1.

Mahesh Prakash, Paul W. Cleary, Soon Hyoung Pyo, Fletcher Woolard. A new approach to boiling simulation using a discrete particle based method[J]. Computers & Graphics, 2015, 53:118-126.

Reeves, W.T. “Particle Systems–A Technique for Modelling a Class of Fuzzy Objects”. Computer Graphics, 1983, 17(3): 359-376

Wang Yinling, Xing Lining, Yan Shiliang. Probability simulation optimization approach using orthogonal genetic algorithm[J]. Wuhan University Journal of Natural Sciences, 2006, 116:.

W.T. Reeves and R. Blau, “Approximate and probabilistic algorithms for shading and rendering structured particle system”, Computer Graphics, 1985, vol. 19, no. 3, pp.313-322

YE Lina. ”Raining Simulation Based on Improved Particle Systems Algorithms”. Software Guide.no.12, vol.11, pp.46-48, 2012

Ying Tan, Chao Yu, Shaoqiu Zheng, Ke Ding. Introduction to Fireworks Algorithm[J]. International Journal of Swarm Intelligence Research (IJSIR), 2013, 44:.

Y. Yin, Y.C. Jin, and H.X. Ren, “Real-Time simulation of natural phenomena”, Acta Simulata Systematica Sinica, 2002, vol. 14.no. 9, pp.1217-1219.