A Hybrid Model for Smoke Simulation

TONG Ruofeng (童若锋) and DONG Jinxiang (董金祥)

Institute of Artificial Intelligence, Department of Computer Science and Engineering, Zhejiang University Hangzhou 310027, P.R. China

E-mail: rft@cs.zju.edu.cn

Received April 18, 2001; revised November 16, 2001.

Abstract A smoke simulation approach based on the integration of traditional particle systems and density functions is presented in this paper. By attaching a density function to each particle as its attribute, the diffusion of smoke can be described by the variation of particles' density functions, along with the effect on airflow by controlling particles' movement and fragmentation. In addition, a continuous density field for realistic rendering can be generated quickly through the look-up tables of particle's density functions. Compared with traditional particle systems, this approach can describe smoke diffusion, and provide a continuous density field for realistic rendering with much less computation. A quick rendering scheme is also presented in this paper as a useful preview tool for tuning appropriate parameters in the smoke model.

Keywords particle system, density function, diffusion, look-up table

1 Introduction

In recent years, smoke simulation has drawn much attention for its prospective application. To simulate gaseous phenomena realistically, many researchers have made great efforts in modeling, movement controlling and rendering, these three main aspects contributing to the reality of simulation.

In realistic rendering, Nishita[1], Klassen[2], Ebert[3] *et al.* have achieved satisfactory results in describing light scattering, reflection and attenuation. In modeling and movement controlling, Peitgen modeled fuzzy objects by fractal approaches[4]. Ebert *et al.* used the turbulent-fields-based density function[3]. These approaches have their own advantages. However, they have difficulty in controlling movement realistically.

A commonly-used method to realistically model the motion of gas is: first generate a wind field (manually or physically), then transfer gas particles using this wind field to get visual effect[5-10]. There are two problems in this method:

(1) How to generate realistic wind fields;

(2) How to generate continuous gas density by discrete particles.

Most researchers addressed the first problem: Sims[5], Wejchert and Haumann[6] generated wind fields by manually placing vortices and flow components. Shinya and Fournier[7] and Sakas[8] used spectral synthesis to give periodic chaotic looking wind fields. And the most convincing wind fields are generated by using the Navier-Stokes equations[11,12].

To produce continuous gas density by discrete particles, Stam and Fiume[9] attached a density function to every particle as one of its attributes. And the total of densities generated by all particles is regarded as the whole density function in the space. But there is no discussion so far concerning how to reflect the diffusion of smoke and the effect of wind field through the variation of particles' density attributes.

Most of the smoke models lack a quick viewing approach, although a quick preview tool is necessary for users to tune the parameters of wind fields to get a satisfying visual effect.

In this paper, a gaseous model in combination with particle systems and density field is presented. In this model, diffusion is described by the variation of density functions of particles, and when the

Supported by the Zhejiang Provincial Natural Science Foundation of China under Grant No.6001107.

influence radius of particle reaches a preset threshold, it will be split into several smaller particles by the wind fields. Using look-up tables, the whole density field can be generated quickly by summing up the density contributions of all the particles. A planting algorithm is used to provide a quick preview tool by projecting all the particles' densities to the screen.

The remainder of this paper is organized as follows. Section 2 introduces the smoke model combined with particle systems and density fields. In Section 3, according to the diffusion equation, an approach to controling the variation of particle's density function is proposed. Section 4 describes how particles move and split under the influence of wind. Then Section 5 presents an efficient rendering approach using projection and look-up tables and Section 6 is the conclusion.

2 A Smoke Model in Combination with Particle System and Density Field

The movement of smoke is mainly composed of the diffusion of smoke and the movement caused by wind fields.

A good smoke model must be able to describe these two aspects, as well as satisfy the demand of realistic rendering. Furthermore, speed is another criterion to evaluate a smoke model.

To satisfy these demands, we attach a density function to a particle to simulate the smoke diffusion and movement, and generate the density field in the space for realistic rendering. Thus, in this approach, the main attributes of a particle are Position: P, Velocity: V, Acceleration: A, Age: T, and Density function at time t: $f(r,t)$ (where r is the distance from a space point X to the particle's position, the density function $f(r,t)$ can be represented by an influence radius R and a density coefficient k).

The position, velocity and acceleration are used to describe the movement affected by airflow. And the density function is used to describe the diffusion of the smoke and to generate the general density field. The general density field can be obtained by (1):

$$\rho(X,t) = \sum_{i=1}^{n} \rho_i(X,t) = \sum_{i=1}^{n} f_i(r,t) \tag{1}$$

where $\rho(X,t)$ is the general density function at time t, X is the position, n is the number of living particles, while $\rho_i(X,t) = f_i(r,t)$ is the density generated by particle I, r is the distance from X to particle I.

There are two factors contributing to the variation of general density fields: the influence of airflow and the diffusion. Although they can be described in one differential equation, we treat them separately due to the complexity. Airflow makes particles move, and divides those particles with a wide influence range into several particles with smaller influence ranges. The diffusion is described by the variation of the influence radius and density function.

3 Diffusion of Smoke

The diffusion of smoke is isotropic when the influence of airflow is not considered. Therefore, it can be expressed by (2) following the Nernst law:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} - a^2\nabla^2\rho = g(x,y,z,t), & (-\infty < x,y,z < \infty,\ t > t_0) \\ \rho(x,y,z,t_0) = \varphi(x,y,z), & (-\infty < x,y,z < \infty) \end{cases} \tag{2}$$

where $\rho(x,y,z,t)$ is the density function for a particle, a^2 is the diffusion coefficient, $g(x,y,z,t)$ is the density generated at the source of smoke, and $\varphi(x,y,z)$ is the initial condition of density at time t_0.

In this approach, the generation of smoke is described by the birth of particles, and (2) is only used to simulate the variation of density after particles are born. Therefore, $g(x,y,z,t) = 0$, and the solution of (2) is:

$$\rho(x,y,z,t)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\Gamma(x-\xi,y-\eta,z-\zeta,t)\varphi(\xi,\eta,\zeta)d\xi d\eta d\zeta \tag{3}$$

where $\Gamma(x-\xi,y-\eta,z-\zeta,t)=\frac{1}{(2a\sqrt{\pi t})^3}e^{-\frac{(x-\xi)^2+(y-\eta)^2+(z-\zeta)^2}{4a^2t}}$.

Now the problem is how to set the initial condition $\varphi(x,y,z)$ for each particle. The smoke density generated by one particle is conservative since the particle is generated. We might set it to 1. When the particle is just born, the density concentrates on the origin although the influence range of the particle will become wider and wider afterwards because of diffusion. Thus we can regard $\varphi(x,y,z)$ as a Dirac delta function. Applying the shifting property of the Dirac delta function to (3), we can get the density contribution of a particle as (4),

$$\rho(x,y,z,t)=f(r,t)=\frac{1}{(2a\sqrt{\pi(t-t_0)})^3}e^{-\frac{r^2}{4a^2(t-t_0)}},\quad t>t_0 \tag{4}$$

where r is the distance from point (x,y,z) to the position of the particle. Considering the computation amount, we set an influence radius R to each particle. The density function of a particle is then described as:

$$f(r,t)=\begin{cases}\frac{1}{(2a\sqrt{\pi(t-t_0)})^3}e^{-\frac{r^2}{4a^2(t-t_0)}}, & r<R\\ 0, & r\geq R\end{cases} \tag{5}$$

The approach of setting influence radius R is as follows.

Set a density threshold ρ_0 first, then get the influence radius R of each particle by solving equation $f(R,t)=\rho_0$, where ρ_0 is an experimental parameter which is usually within [0.0001, 0.001]. Denoting t_{age} as the age of the particle, we can derive from $f(R,t_{\text{age}})=\rho_0$ the relation between particle age and its influence radius as (6)

$$R=2a\sqrt{-t_{\text{age}}\Big(\ln 8a^3\rho_0+\frac{3}{2}\ln\pi+\frac{3}{2}\ln t_{\text{age}}\Big)} \tag{6}$$

The efficiency of calculating the density contribution of a particle to a certain point (x,y,z) can be improved by using a look-up table.

First, we calculate the influence radius R_j corresponding to a sequence of particle ages t_j ($t_j=t_0+j\Delta t$) by solving $f(R_j,t_j)=\rho_0$. An influence radius look-up table TR is generated to store all these R_j. Then, for each R_j in TR, divide R_j into several intervals by a preset value Δr, calculate the density contributions $f_j^k=f(k\Delta r,t_j)$ according to (5), and store them to a density look-up table TD_j. Thus, each item in TR is linked with a look-up table, as illustrated in Fig.1.

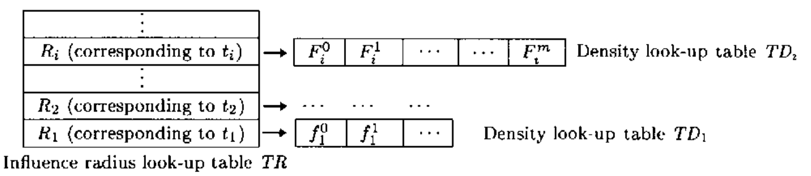

Fig.1. Influence radius look-up table TR and density look-up table TD_i.

To calculate the density contribution of particle I with age t_i to point (x,y,z), first look-up in influence radius look-up table TR to get the influence radius R_i corresponding to age t_i. Then to calculate the distance r from (x,y,z) to the position of particle I, look-up in the density look-up table TD_i, and select f_i^k whose corresponding $k\Delta r$ is the closest one to r. Finally, f_i^k is taken as the density contribution of particle I to point (x,y,z).

4 Influence of Airflow

To realistically simulate smoke, a good wind model is necessary. Ebert, Sakas, Foster, Stam and others[3,8,11,12] have achieved good results in this field. We assume that the wind field is already generated because this paper focuses on how to control the movement of smoke. Compared with traditional particle systems, the particle in this approach has a density function with it. Thus the airflow not only makes particles move but also affects the density function. This is described by the fragmentation of particles.

All living particles are classified into two sets W_1 and W_2 by a threshold R_0. The particles whose influence radii are smaller than R_0 belong to W_1, and the others belong to W_2. For those particles of W_1, we assume that airflow just makes them move; while for those of W_2, airflow makes them both move and split. The detailed approach is as follows.

Divide the space into uniform cubes, and set the acceleration caused by airflow to each cube. For particle J, if $J \in W_1$, the next position of J can be calculated by (1) and (2). Acceleration A can be obtained in the following way.

Assume the influence of particle J has intersections with cubes $C_1, C_2, \ldots, C_n$, the distance from the position of the particle to the center of cube C_i is l_i, and the acceleration caused by airflow in cube C_i is a_i, then the acceleration of particle J can be calculated as $A = \frac{\sum_{i=1}^{n} \frac{a_i}{l_i}}{\sum_{i=1}^{n} \frac{1}{l_i}}$.

If $J \in W_2$, the airflow will split it into several particles of smaller influence radius, and make them move individually. Similar to those particles in W_1, we obtain cubes $C_1, C_2, \ldots, C_n$, which intersect the influence range of particle J, get acceleration a_i and calculate distance l_i. Then different from those in W_1, particle J is deleted and several smaller particles $J_1, J_2, \ldots, J_n$ are generated. J_i has the same velocity as particle J, and its influence radius can be obtained by looking up table.

Calculate the coefficient $k_i = \frac{1/l_i}{\sum_{i=1}^{n} \frac{1}{l_i}}$ first. Then search in the influence radius table TR to find radius R_i which has minimum difference from $k_i R_j$ (R_j is the influence radius of particle J), and set it as the influence radius of particle J_i. Finally, the density coefficient of particle J_i is set as $k_i k$, where k is the density coefficient of particle J when it is not been split. The density coefficient is used to keep the total density generated by a particle unchanged when the fragmentation occurs. The initial density coefficient of a particle is set as 1.0. The position of the new particle at the next time point can be obtained according to the Newton's laws after all the above attributes are set.

5 Efficient Rendering

When the general density field is generated, smoke can be rendered in different approaches according to the demand of optical effect. Ray tracing can describe many optical phenomena such as scattering, reflection and attenuation of light, but it takes long time to generate even a single image. Here we give an efficient rendering approach to provide a preview tool for smoke simulation. It is really useful in the period of tuning appropriate parameters of a smoke model. The main idea is that projecting the density of all the particles to the screen, the accumulation of the density contributions of all the particles at screen pixel (x, y) is used to obtain the grey value at (x, y). Given the attenuation, we have to sort the particles according to their distances to the screen plane. The procedure of the projecting approach is as follows:

Step 1. Make the screen a project plane, open a density map for the screen and set the initial value 0 to each pixel.

Step 2. Sort all of the particles by their positions in a descending order of their Z components.

Step 3. Project the densities of particles to the screen one by one according to the order sorted in Step 2.

In Step 3, the projection of particles' influence range on the screen is regarded as a circle approximately. The center of projection circle (x_p, y_p) can be calculated as $x_p = k(x_1 - x_0) + x_0$, $y_p = k(y_1 - y_0) + y_0$, $k = \frac{z - z_0}{z_1 - z_0}$, where (x_0, y_0, z_0) is the coordinate of viewpoint and (x_1, y_1, z_1) is the particle's position, z is the Z coordinate of screen. The radius of the projection circle is $R_p = kR$, where R is the influence radius of the particle. In order to use a look-up table to speed up rendering, we approximately treat the projection density of a particle to be isotropic although actually it is not. Thus the projection density on a screen point (x, y) is given by (7).

$$\rho_p(r) = \begin{cases} \int_{-\sqrt{R^2-(r/k)^2}}^{\sqrt{R^2-(r/k)^2}} \frac{1}{(2a\sqrt{\pi t_j})^3} e^{-\frac{(r/k)^2+z^2}{4a^2 t_j}} dz, & r < R_p \\ 0, & r \geq R_p \end{cases} \quad (7)$$

where r is the distance from (x, y) to (x_p, y_p).

The integral is calculated with r changing from 0 to R_p at a fixed interval and stored in the projection density look-up table beforehand.

After we get the projection density of a particle at pixel (x, y), the new density at (x, y) can be calculated as

$$\rho_{\text{new}} = \rho_{\text{old}} + \rho_p e^{-k_e \rho_{\text{old}}}$$

where ρ_{old} is the density at (x, y) before the current projection, k_e is the attenuation coefficient. After all the living particles have been projected, we can get the grey value of (x, y) in the screen according to its density.

6 Conclusion

We have presented a smoke model based on the combination of particle system and density function. This model makes it possible to naturally describe the diffusion of the smoke and the influence of the airflow by means of particles' movement and fragmentation, and density function's variation. Continuous density field can also be generated by a relatively small amount of particles. The density contribution of particles with different ages can be calculated beforehand and repeatedly used for all the particles by storing in look-up tables. This dramatically improves the speed compared with the conventional particle systems.

References

[1] Nishita T, Miyawaki Y, Nakamae E. A shading model for atmospheric scattering considering luminous intensity distribution of light source. *ACM SIGGRAPH Computer Graphics*, 1987, 21(4): 303–310.

[2] Klassen R V. Modeling the effect of the atmosphere on light. *ACM Transactions on Graphics*, 1987, 6(3): 215–237.

[3] Ebert D S, Parent R E. Rendering and animation of gaseous phenomena by combining fast volume and scanline A-buffer techniques. *ACM SIGGRAPH Computer Graphics*, 1990, 24(4): 357–366.

[4] Peitgen H O, Saupe D. The Science of Fractal Images. Springer Verlag, Berlin, Heidelberg, New York, 1988.

[5] Sims K. Particle animation and rendering using data parallel computation. *ACM SIGGRAPH Computer Graphics*, 1990, 24(4): 405–413.

[6] Wejchert J, Haumann D. Animation aerodynamics. *ACM SIGGRAPH Computer Graphics*, 1991, 25(4): 19–22.

[7] Shinya M, Fournier A. Stochastic motion-motion under the influence of wind. In *Proceedings of Eurographics'92*, 1992, pp.119–128.

[8] Sakas G. Modeling and animating turbulent gaseous phenomena using spectral synthesis. *The Visual Computer*, 1993, 9(4): 200–212.

[9] Stam J, Fiume E. Turbulent wind fields for gaseous phenomena. In *Proc. SIGGRAPH'93*, 1993, pp.369–376.

[10] Rudolf M J, Raczkowski J. Modeling the motion of dense smoke in the wind field. *Computer Graphics Forum*. 2000, 19(3): 21–29.

[11] Foster N, Metaxas D. Modeling the motion of a hot, turbulent gas. In *Proc. SIGGRAPH'97*, 1997, pp.181–188.

[12] Stam J. Stable fluids. In *Proc. SIGGRAPH'99*, 1999, pp.121–128.