Research on GPU-based Real-time Smoke Simulation Volume Rendering Algorithm

Dingsheng Deng¹ *
¹Sichuan University Nationalities, Kangding, Sichuan, 626001, China
*Corresponding author’s e-mail: dds0904@scun.edu.cn

Abstract: With the continuous development of the current film industry and animation industry, in the current smoke simulation, based on the use of GPU, real-time smoke simulation can be realized, so that real-time smoke simulation can be enhanced to achieve more smoke simulation. The ideal effect makes people have a better visual experience. The purpose of this article is to study the GPU-based real-time smoke simulation volume rendering algorithm. This paper proposes a method of setting different transfer functions based on different positions of the imaging plane. By setting different transfer functions for different positions of the imaging plane, different tissue structures can be drawn at one time for comparison and observation. However, setting different transfer functions based on the coordinate information of the imaging plane cannot fully display the volume data information. This paper implements a method of setting different transfer functions based on different spatial positions in the object space, and sets different transfer function schemes for the region of interest in the object space. Highlight, rotate or zoom the drawn object to observe the structure of the region of interest at any angle and size. In addition, this paper realizes the rapid volume cutting of volume rendering results, which is convenient for observing the organization structure information of the occluded position or designated area. Experimental results show that the algorithm in this paper can be used to control the random motion of smoke in three-dimensional space in virtual and real scenes, and can complete various interactive simulations of smoke and obstacles.

1. Introduction
In actual work, the specific process is as follows: first initialize the density field and velocity field on the basis of the CPU, and then obtain the external force of the velocity field and the density field [1]. On this basis, the relevant data is copied from the memory to the video memory, then the density field and velocity field are calculated, then the relevant data is copied to the memory through the video memory, and finally the density field and velocity field are drawn. After completing the above steps, the next step [2] will start. At present, the floating-point performance and data throughput of GPUs have surpassed mainstream CPUs of the same era. In addition, GPU has independent processing core, memory space and programming language, and obtains the programmable ability similar to CPU, all of which make GPU the mainstream computing platform for 3D real-time rendering [3-4].

The emergence and development of GPU has not only greatly improved the speed and quality of computer graphics processing, but also greatly promoted the rapid development of computer graphics-related applications [5]. The graphics pipeline is the most basic and important technical framework in computer graphics. Almost all computer graphics research and development work are based on this framework model. Due to the irregular appearance of smoke and the characteristics of rapid changes, the study of smoke simulation has always been a difficult problem and full of
challenges [6]. In order to show the good interactive characteristics of smoke in real time, this paper mainly did two researches, namely the research of smoke trajectory algorithm in virtual and real scenes and the research of interactive algorithm between smoke and obstacles [7].

First, in order to achieve the experimental effect of controlling the movement of smoke in the virtual scene, a target velocity field whose direction is tangent to the path is generated near the path. By introducing feedback control technology, the simulated velocity field is continuously adjusted to make it consistent with the target velocity field. The effect of real-time movement of smoke along various set trajectories [8]. At the same time, the obstacles are discretized by voxelization, and the smoke and obstacles are uniformly calculated through the two-way coupling equation to obtain the simulated value [9]. Finally, as a modern technology, real-time smoke simulation technology has a very wide range of applications and plays a very important role. This paper analyzes the real-time smoke simulation based on GPU, describes the actual operation process, and finally obtains the real-time smoke simulation effect. As relevant professionals, we should strengthen our understanding and mastery of this technology in order to better apply it [10].

2. Algorithm optimization

2.1 Navier Stokes equation is the most complete equation describing fluid phenomena by Euler method and Lagrange simulation algorithm.

There are two methods to describe the physical quantities of fluid motion in fluid mechanics: Euler method and Lagrangian method. The NS equation is based on different description methods and also has different forms.

Euler’s method:
It is a particle-based method to study the changes over time of the velocity, pressure, density and other parameters of a specified micro cluster in the fluid to study the motion of the entire fluid. If the Lagrangian method is adopted, that is, the particle system-based method is adopted, then the NS equation can be rewritten as:

$$\frac{Du}{dt} = \nu \nabla^2 u - \frac{u_p}{\rho} + f$$ (1)

Degenerate to Newton’s second law:

$$u(x, t + \delta t) = u(x, t) + \delta tF(x, t)$$ (2)

This method is to analyze the force of each relatively independent particle, and calculate the position of these particles at the next moment and other state quantities through integration.

Suppose the affinity of the grid corresponding to the direction d/k(k=1,...,8) is ek (0≤ek≤1), and the probability based on the affinity is assigned to the angle of the main vector Vp is less than the threshold in the direction of angle θ ∈ (0, π/2), calculate:

$$\sum_{k=1}^{8} c_k u(\theta - \cos^{-1}(d_k \cdot V_p)) / A_{sum}$$ (3)

$$A_{sum} = \sum_{i=1}^{8} c_i u(\theta - \cos^{-1}(d_i \cdot V_p))$$ (4)

Where Ak(k=1,...,8) is the probability based on affinity, (d/k·V/p) represents the inner product, and V/p is the unit vector of Vp. When x≥0, u(x) = 1, otherwise u(x) = 0.

$$R_k = (\delta \gamma_1) V_p |D_k + \gamma_2 A_k| / R_{sum}$$ (5)

$$R_{sum} = \sum_{i=1}^{8} (\gamma_1 V_p |D_i + \gamma_2 A_i|)$$ (6)

Among them, Rk(k=1,...,8) is the area of eight adjacent regions, γ1 is the wind factor, which controls the degree of influence of the water droplet movement trajectory by the wind, and at the same time adds the artificial parameter δ for correction. γ2 is the bending factor, which controls the water flow. Then, find the index p(p=1,...,8) that meets the following conditions:
\[ \sum_{k=1}^{p-1} R_k < \alpha \leq \sum_{k=1}^{p} R_k \]  

(7)

Where \( \alpha \in (0,1) \) is a random number generated based on a uniform distribution and \( \sum_{0k=1}^{Rk} = 0 \). Finally, the direction of movement of the water droplet is determined as \( d/p \).

3. Modeling method

3.1 Solving the diffusion term

The new type of smoke simulation algorithm based on physical model is to solve the equation on GPU. This paper intends to use configuration grid to discretize the solution set.

Because the viscous liquid has a certain obstacle to the flow, it will affect the speed of the diffusion. From a certain point of view, it describes the mass exchange between adjacent units, that is, the inflow and outflow. The partial differential equation for viscous expansion is:

\[ \frac{\partial u}{\partial t} = \nu \nabla^2 u \]  

(8)

3.2 Solve the external force term

Because mass force directly affects acceleration, the following formula can be used to calculate the density field change caused by external force:

\[ u(x,t + \delta t) = u(x,t) + \delta t \cdot F(x,t) \]  

(9)

3.3 Solve the advection term

We use the semi-Lagrangian method to calculate the advection term. Instead of calculating where the particle moves in the current time period, we get the advection term. Instead, we track the particle trajectory from each grid cell in reverse time, find the previous position, and copy it. Down:

\[ \sigma(x,t + \delta t) = \sigma(x - u(t) \delta t, t) \]  

(10)

Therefore, this requires that the actual density difference of the real-time smoke must be diffused to a certain extent during the conversion process to make its value equal to zero. This measure can largely guarantee the quality of the two different states of its real-time smoke is conserved. In numerical calculations, on the basis of satisfying the Newman boundary, the existing density difference can be processed by the diffusion equation. The equation can be expressed as:

\[ \frac{\partial \rho}{\partial t} = \nabla^2 \rho \]  

\[ \frac{\partial \rho}{\partial n} = 0 \]  

(11)

On the basis of the constant target state of real-time smoke, substituting the equation of density difference into the relevant diffusion equation, another equation can be obtained, namely:

\[ \frac{\partial \rho}{\partial t} \nabla^2 (\rho - \rho^*) \]  

(12)

At the same time, in order to ensure that the cohesive force in the simulation calculation only appears in places where the real-time smoke density and concentration are large, the \( \rho \) and \( \rho^* \) factors can be substituted into the real-time smoke density and concentration to make certain adjustments, then the cohesive force can be expressed as:
4. Data algorithm evaluation results and research Result

4.1 Comparison of gpu and cpu results

The effect comparison of this smoke simulation is realized by the frame rate of the rendering on the CPU and GPU, and the frame rate comparison of the rendering is shown in Table 1 and Figure 1. Comparing Table 1 and Figure 1, it is not difficult to find that using GPU rendering smoke effect method can achieve real-time requirements, which is better than CPU rendering smoke effect in real-time and stability.

$$G(p, p^*) = \nabla^2 (p - p^*)$$

$$G(p, p^*) = \nabla [p \cdot \nabla (p - p^*)]$$

(13)

### Table 1. Frame rate comparison of rendering smoke effect

| Frame rate number | CPU rendering frame rate (frame/sec) | GPU rendering frame rate (frame/second) |
|-------------------|--------------------------------------|----------------------------------------|
| 123               | 12.145                               | 45.238                                 |
| 132               | 12.452                               | 45.225                                 |
| 245               | 12.562                               | 45.422                                 |
| 261               | 12.223                               | 46.221                                 |
| 289               | 13.110                               | 45.125                                 |

4.2 Treatment method

The method we use is based on Euler grid, which is divided into data processing, simulation process and rendering steps. The data processing stage is based on the smoke bounding box and the smoke simulation effect to set the multi-smoke subregional division strategy and set the smoke source basic data such as the position, initial speed and so on; in the simulation stage, it includes calculation of basic smoke simulation process such as convection term, pressure projection term, external force term, etc., and boundary processing of the smoke, using Dirichlet boundary model; finally using GPU and The CPU parallel acceleration technology completes the implementation of rendering and drawing. Finally, the data obtained in Table 1 shows that under the same operation, the average CPU rendering frame rate is about 12 frames per second, while the average GPU rendering frame rate is about 45 frames per second. The rendering frame rate is much greater than the CPU rendering frame rate.

![Figure 1. Frame rate comparison of rendering smoke effect](image-url)
Figure 1 is a column chart based on Table 1. Figure 1 shows the difference between the CPU rendering frame rate and the GPU rendering frame rate more intuitively, highlighting the advantages of GPU.

4.3 Advantages and disadvantages of GPU and CPU
The CPU is the central processing unit, the core of the computer. Almost all calculations and data processing are given to it. The GPU is the graphics processor, the core of the graphics card, and is responsible for transmitting graphics to the display. The GPU is a new technology that has emerged in recent years and is used as a stream processor. GPU can realize the processing of general-purpose computer through certain technical means. The programmable GPU under the current technical means can realize its vertex-level and pixel-level processing separately in the true sense. Therefore, when the real-time smoke simulation has very high accuracy requirements for image accuracy, the GPU can usually play a more important role. However, the CPU is the core of the computer. Even the best GPU needs the support of the CPU to complete it. In multithreading, the parallel modules are calculated in the best way. The decoding optimization algorithm design based on GPU and CPU coordinated acceleration makes full use of it. High parallelism and a large number of ALU arithmetic logic units, deliberately avoid the weaknesses of multi-branch and strong correlation, and balance the scheduling of computing resources. The memory data is uploaded to the video memory when the CPU operation stage is switched to the GPU. The data is always placed in the video memory during the GPU calculation. The thread mapping method does not need to be extra time-consuming, so the mapping method is theoretically optimal. Therefore, if you need high-precision real-time smoke simulation volume rendering, you need a powerful GPU and CPU.

5. Result
Based on the physical model of smoke simulation, the article successfully realized the real-time rendering of smoke simulation on GPU through real-time volume rendering algorithm, achieving the purpose of automatically generating smoke effects that look natural and real, so as to achieve real-time and real-time smoke rendering effects. Sex. This paper proposes a real-time simulation method for the free fusion of multiple smoke based on Euler's method. A set of multiple smoke area division strategies is proposed on the basis of the traditional smoke hydrodynamic model, and the method of combining local control and overall adjustment is used to smoothly process The problem of multi-smoke fusion boundary, quickly and steadily portray the movement of multi-smoke in the interactive process, and improve the authenticity of simulation. At the same time, a color fusion method is proposed using the light projection algorithm to express the free fusion of multiple smokes. At the same time, advanced GPU technology is applied to accelerate the whole algorithm during the whole conversion process. The simulation algorithm studied in this paper integrates the speed and density of real-time smoke into a color channel with a certain texture, and then transfers it to the relevant pixel program, and finally completes the relevant calculation and processing through the GPU. The entire system can generate 80 frames of images within 3s, which meets its real-time requirements and standards to a large extent, while maintaining the natural flow of smoke to a large extent.

Acknowledgments
This paper was financially supported by the Key Project of Natural Science of Sichuan Provincial Education Department (No. 17ZA0295), 2017 Applied Demonstration Course Project of Sichuan University for Nationalities (No. sfkc201705) and Key Project of Natural Science of Sichuan University for Nationalities (No. XYZB19001ZA).

References
[1] Davidson M, Reed S. Preliminary assessment of real-time air quality instruments for monitoring environmental tobacco smoke[J]. British Actuarial Journal, 2018, 3(2):547-614.
[2] Park K M, Bae C O. Smoke detection in ship engine rooms based on video images[J]. IET Image
[3] Gadeski E, Fard H O, Le Borgne H. GPU deformable part model for object recognition[J]. Journal of Real Time Image Processing, 2018, 14(2):279-291.

[4] Shi X, Luo X, Liang J, et al. Frog: Asynchronous Graph Processing on GPU with Hybrid Coloring Model[J]. IEEE Transactions on Knowledge & Data Engineering, 2018, PP(1):1-1.

[5] Wang L, Zhang Z, Huang C, et al. A GPU-accelerated Parallel Jaya Algorithm for Efficiently Estimating Li-ion Battery Model Parameters[J]. Applied Soft Computing, 2018, 65:12-20.

[6] Jibran H, Hean L C, Gil A J, et al. An upwind cell centred Total Lagrangian finite volume algorithm for nearly incompressible explicit fast solid dynamic applications[J]. Computer Methods in Applied Mechanics & Engineering, 2018, 340:684-727.

[7] Che P, Tang Z, Gong H, et al. An Improved Lagrangian Relaxation Algorithm for the Robust Generation Self-Scheduling Problem[J]. Mathematical Problems in Engineering, 2018, 2018(PT.9):1-12.

[8] Tang X, Zhang L, Wang X. Sparse augmented Lagrangian algorithm for system identification[J]. Neurocomputing, 2019, 330(FEB.22):403-411.

[9] Papageorgiou N S, Vetro C, Vetro F. Multiple solutions with sign information for a (p, 2)-equation with combined nonlinearities[J]. Nonlinear Analysis, 2020, 192:111716.

[10] Papageorgiou N S, Vetro C, Vetro F. On a Robin (p,q)-equation with a logistic reaction[J]. Opuscula Mathematica, 2019, 39(2):227-245.