

# 基于物理模型的实时烟雾模拟

汪继文<sup>1,2</sup>, 杨贤达<sup>2</sup>

(1. 安徽大学 计算机智能与信号处理教育部重点实验室, 安徽 合肥 230039;  
2. 安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

**摘要:**针对使用直接绘制法渲染烟雾时,在高网格分辨率下实时性差的问题,文中提出了一种新的基于纹理的烟雾渲染算法。该算法采用不可压缩的 Navier-Stokes 方程作为烟雾模拟的基本物理计算模型,从而保证烟雾物理运动的真实性,同时利用 OpenGL 中的纹理映射技术渲染烟雾,保证了烟雾渲染效果真实,烟雾模拟细节更能体现出来;另外文中还给出了在烟雾中添加障碍物时边界条件的计算方式。模拟实验结果表明,该算法既能满足烟雾模拟的实时性,又能真实有效地模拟烟雾流动情况。

**关键词:**流体动力学;烟雾模拟;N-S 方程;实时渲染;边界条件

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2014)01-0122-04

doi:10.3969/j.issn.1673-629X.2014.01.031

## Real-time Smoke Simulation Based on Physics Model

WANG Ji-wen<sup>1,2</sup>, YANG Xian-da<sup>2</sup>

(1. Ministry of Education Key Laboratory of Computing & Signal Processing, Anhui University,  
Hefei 230039, China;  
2. College of Computer Science & Technique, Anhui University, Hefei 230039, China)

**Abstract:** In order to avoid the limitations of direct rendering for high grid smoke simulation, a texture rendering method is presented in this paper. This algorithm adopts incompressible Navier-Stokes equations as the basic physics computing model of smoke simulation to guarantee the realistic of smoke movement, simultaneously applies texture mapping in OpenGL to determine the smoke rendering effect. A calculation method of boundary conditions is also presented when obstacle is added in the smoke. The simulation experiment of the smoke phenomenon shows that the algorithm can both run in real-time for high grid sizes and efficiently simulate realistic movement of smoke.

**Key words:** fluid dynamics; smoke simulation; N-S equations; real-time rendering; boundary conditions

## 0 引言

基于物理的流体模拟通过求解物理方程描述流体运动的规律,因此它相对于传统流体模拟<sup>[1-2]</sup>更能具有说服力,也是当前和未来流体动画发展的方向。基于物理的流体模拟可以分为两个大部分,一部分是根据物理模型求解烟雾的速度场和密度场,即 Navier-Stokes (N-S) 方程的数值求解过程。对于物理方程的求解,在 1999 年,Stam<sup>[3]</sup>引入无条件稳定的半拉格朗日方法,通过引入旋度因子解决该方法的数值扩散问题<sup>[4]</sup>,之后在他的基础上产生了许多求解方法,比如 MacCormack 方法<sup>[5]</sup>、BFEC 方法<sup>[6]</sup>。这些方法相比

于传统的半拉格朗日方法精度高,但对于求解可压缩流体时都不能达到动量守恒,而文献[7]则提出了质量和动量守恒的方法,文献[8]提出了求解可压缩或是不可压缩流体都能无条件稳定的完全守恒的半拉格朗日方法,上述算法都是基于网格的算法。还有一种是基于无网格的光滑粒子动力学方法 (SPH)<sup>[9-10]</sup>,这是种纯拉格朗日方法,无须网格计算与划分,此方法将流体看作是相互作用的质点组来描述,然后再求解相应的物理方程;流体模拟的另一部分是流体渲染过程。

文中使用半拉格朗日方法计算烟雾物理模型,在此基础上给出了烟雾中添加障碍物时边界条件的计算

收稿日期:2013-02-24

修回日期:2013-05-28

网络出版时间:2013-09-29

基金项目:安徽省高等学校省级自然科学基金项目(KJ2013A009)

作者简介:汪继文(1958-),男,安徽安庆人,博士生导师,教授,研究方向为流体动画模拟、计算流体力学;杨贤达(1988-),男,安徽安庆人,硕士研究生,研究方向为流体动画模拟。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130929.1521.007.html>

方法,采用纹理映射渲染烟雾,解决了直接绘制方法受网格大小约束的问题。

1 烟雾模拟的物理模型

文中在求解烟雾速度场中使用的是不可压缩的 N-S 方程,方程如下:

∂u/∂t = f - 1/ρ ∇p + k ∇²u - (u · ∇)u (1)

∇ · u = 0 (2)

方程(1)称之为动量方程,以保证动量守恒。其中,u=(u,v)表示速度场;f表示外力场;ρ表示密度;p表示压力;k表示粘性系数,由于文中模拟的烟雾是无粘性流体,所以k=0。因此方程(1)可以简化为无粘性的不可压缩欧拉方程:

∂u/∂t = f - 1/ρ ∇p - (u · ∇)u (3)

方程(3)的右边第一项表示外力项,第二项表示压力项,第三项表示对流项。方程(2)称之为质量方程,以保证烟雾运动中质量守恒。

烟雾的密度场是随着烟雾的速度场变化而变化的,因此密度场的计算物理模型可用如下方程表示:

∂ρ/∂t = S + k ∇²ρ - u · ∇u (4)

其中,k是扩散系数;S表示外界向网格中添加的密度,烟雾的密度取值范围为0~1,密度值为0时表示当前网格点没有烟雾粒子[11]。

2 方程的求解

求解烟雾模拟的物理方程主要是将上述方程(3)和方程(4)拆分,分项求解,它们分别为源力项、对流项、扩散项和粘力项,为了使烟雾模拟具有稳定性,求解速度场时增加了投影项的计算。由于文中模拟的烟雾是不可压缩无粘流体,所以粘力项忽略。最后还得考虑到流体模拟边界条件的计算,而且在文中还给出了在场中添加障碍物时边界条件的计算模型。

首先求源力项,即外力对当前场(速度场与密度场)的影响,文中使用欧拉方法求解方程

∂D/∂t = S源 (5)

其中,D是矢量,表示当前计算的某个场(速度场与密度场)。如果D是速度场,S源表示的是方程(3)的外力f;如果D是密度场,S源表示的就是外界向网格中添加的密度S,使用欧拉方法将方程(5)离散化如下:

Dn+1 = Dn + Δt · S源 (6)

其中,n表示tn时刻;Δt表示时间步长。

其次求解对流项,方程表示为:

∂D/∂t = -u · ∇u (7)

对流项的求解文中采用的是 Stam[3]提出的半拉格朗日方法,它是欧拉法和拉格朗日方法的结合。半拉格朗日方法的核心是将每个网格单元都看作一个粒子,利用速度场回溯追踪粒子上上一时刻的位置,再对这个位置周围网格点的值进行插值求当前时刻粒子的值,计算过程可参考文献[12-13]。

然后求解扩散项,即

∂ρ/∂t = k ∇²ρ (8)

扩散项的求解只针对方程(4),速度场的粘力项为零,不需要考虑。此项的求解文中使用一种隐式求解方法,即

ρi,j^n = ρi,j^(n+1) - α(ρi-1,j^(n+1) + ρi+1,j^(n+1) + ρi,j-1^(n+1) + ρi,j+1^(n+1) - 4ρi,j^(n+1)) (9)

其中,α=k·N·N·Δt,N表示划分网格的大小。上式采用高斯-赛德尔迭代求解当前密度场,迭代的次数越多,精度越高,当然计算速度会变慢,为了保证流体模拟的实时性,一般迭代20次就可以达到要求[11]。

最后求解投影项对速度场的作用,文中使用了 Stam[3]提出的计算方法,通过引入一个投影算子P,最后化解为求解一个关于该压力的泊松方程。具体计算过程可参见文献[3-4]。

综上求解每个时刻场的变化过程总结如下:

速度场计算步骤:

- 初始化速度场 u0
- 对每一个时间步 n = 0,1,2,...
  - uf = addForce(un,Δt) // 源力项
  - ud = diffuse(uf,Δt,k) // 扩散项
  - ua = advect(ud,Δt) // 对流项
  - un+1 = project(ua,Δt) // 投影项

密度场计算步骤:

- 初始化密度场 ρ0
- 对每一个时间步 n = 0,1,2,...
  - ρs = addForce(ρn,Δt) // 源力项
  - ρd = diffuse(ρs,un,Δt,k) // 扩散项
  - ρn+1 = advect(ρd,un,Δt) // 对流项

3 边界条件

上述方程求解过程是基于网格的,文中将场景划分为(N+2)×(N+2)的二维网格空间,速度场与密度场的计算都存储于每个网格单元的中心,如图1所示。

之所以划分为  $N + 2$  是因为网格计算中需要考虑到边界条件的计算<sup>[13-14]</sup>, 图 1 中的第  $0, N + 1$  行(列)为边界网格单元。要使得计算结果满足不可压缩条件, 在边界处的速度必须为 0, 而网格点的值被定义为网格中心, 因此边界点的值可由与之相邻的两个网格单元中的值的平均值求出<sup>[15]</sup>。比如速度场中的网格左边界网格单元的值可使用如下公式:

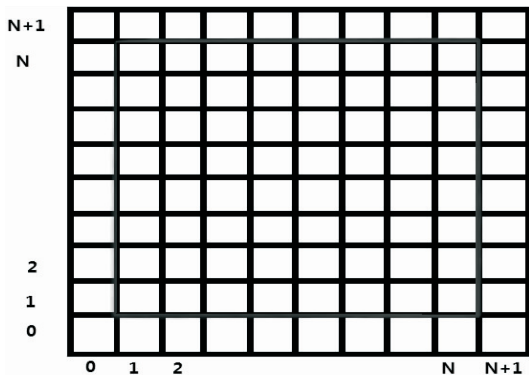


图 1 计算网格

$$\frac{u_{0,j} + u_{1,j}}{2} = 0, j \in [1, N] \quad (10)$$

四个顶点网格单元可用相邻网格单元平均值求出, 如图 1 中  $(0,0)$  单元格的速度场可由  $(0,1), (1,0)$  单元格的速度场求出, 公式如下:

$$u_{0,1} = \frac{u_{0,1} + u_{1,0}}{2} \quad (11)$$

上面给出的是场景周围四个边界的计算方法, 更一般的, 文中给出了在场景中增加一个正方形时, 对该正方形固体边界条件的计算模型, 计算方法如下:

$$\begin{aligned} \text{正方形左边界: } u_{x\text{begin},j} &= -u_{x\text{begin}-1,j} & j \in (y\text{begin}, \\ v_{x\text{begin},j} &= v_{x\text{begin}-1,j} & y\text{end}) \end{aligned}$$

$$\begin{aligned} \text{正方形右边界: } u_{x\text{end},j} &= -u_{x\text{end}+1,j} & j \in (y\text{begin}, \\ v_{x\text{end},j} &= v_{x\text{end}+1,j} & y\text{end}) \end{aligned}$$

$$\begin{aligned} \text{正方形上边界: } u_{i,y\text{end}} &= -u_{i,y\text{end}+1} & i \in (x\text{begin}, \\ v_{i,y\text{end}} &= v_{i,y\text{end}+1} & x\text{end}) \end{aligned}$$

$$\begin{aligned} \text{正方形下边界: } u_{i,y\text{begin}} &= -u_{i,y\text{begin}-1} & i \in (x\text{begin}, \\ v_{i,y\text{begin}} &= v_{i,y\text{begin}-1} & x\text{end}) \end{aligned}$$

其中  $(x\text{begin}, y\text{begin}), (x\text{end}, y\text{end})$  表示的是正方形左下角和右上角网格点的坐标。

## 4 烟雾渲染

文中烟雾渲染采用的是 OpenGL 纹理映射技术与色彩混合技术。

OpenGL 是一个跨平台的图形程序接口, 功能强大, 也可以说它是一个方便程序员调用的底层图形库。OpenGL 中的纹理映射技术一方面可以产生相对逼真的烟雾效果, 另一方面相比于直接绘制法烟雾渲染速度快。

文中使用了两种纹理方法渲染烟雾: 第一种方式是通过密度场制作出一张纹理, 即将密度场的值保存在一张纹理中, 伪代码如下:

```
//将密度场 dens 数组中的值赋值于纹理数组 texture 中
· 对每一个网格单元  $(i, j), i, j \in (0, N + 2)$ 
·  $\text{int index} = ((N + 2) * I + j);$ 
·  $\text{int ind} = \text{index} \ll 2;$ 
·  $\text{if}(! \text{is\_bnd}(N, i, j)) // \text{边界判断}, (i, j) \text{ 单元格是否是边界}$ 
·  $\text{texture}[\text{ind}] = \text{red} * \text{dens}[\text{index}]; // \text{纹理中} R \text{ 通道}$ 
·  $\text{texture}[\text{ind} + 1] = \text{green} * \text{dens}[\text{index}]; // \text{纹理中} G \text{ 通道}$ 
·  $\text{texture}[\text{ind} + 2] = \text{blue} * \text{dens}[\text{index}]; // \text{纹理中} B \text{ 通道}$ 
·  $\text{texture}[\text{ind} + 3] = \text{alpha} * \text{dens}[\text{index}]; // \text{纹理中的} A \text{ 通道}$ 
```

上面代码中, texture 是纹理数组, dens 存储的是密度场的值。red, green, blue, alpha 变量取值范围为  $0 \sim 1$ , 可通过设定它们的值来确定最终渲染烟雾的颜色, 比如当  $\text{red} = 1, \text{green} = \text{blue} = \text{alpha} = 0$  时, 则最终烟雾的颜色为红色, 实验效果见文中第 5 部分。

第二种纹理烟雾渲染方式是将外部图片制作成一张纹理, 这种纹理映射技术可以使得烟雾不再是单一的颜色, 而是彩色, 并且纹理 RGB 三个颜色通道可通过 OpenGL 中的混合技术分离出来。实验效果见文中的第 5 部分。

## 5 实验结果与分析

文中实验硬件环境是 Intel(R) Core™ i3 CPU 2.7 GHz, 内存为 2.0 GB 的 PC 机, 软件平台是 VS2010 和开放的图形库 OpenGL3.2, 系统网格分辨率为  $200 \times 200$ , 即  $N = 198$ 。

实验效果如图 2 所示。

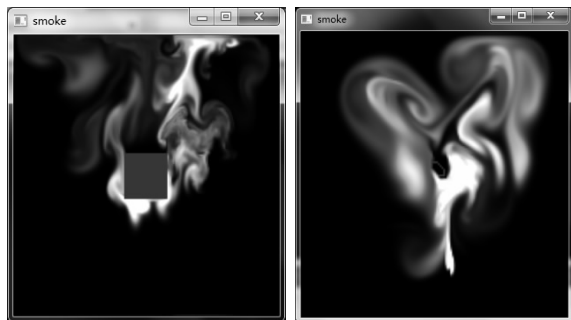


图 2 基于第一种纹理方法的烟雾渲染效果

图2使用的是第一种纹理技术的实验结果,上两图中间是一个静止的障碍物,左图使用的是规则障碍物,而右图使用的是不规则障碍物,其中扩散因子  $k = 0.001$ ,  $t = 0.1$ ,  $\text{red} = \text{blue} = 1$ ,  $\text{green} = 0.5$ ,  $\alpha = 0$ 。

图3采用的是第二种纹理技术渲染得到的效果,可使用OpenGL中的glTexEnvf函数将纹理的RGB颜色通道提取出来,glTexEnvf函数参数设置如下:

glTexEnvf (GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_BLEND)

最后一个参数设置为GL\_BLEND。比如如果想提取红色,则背景颜色或者绘制时的颜色设定为红色,可使用函数glColor3f(1.0,0.0,0.0)来实现。

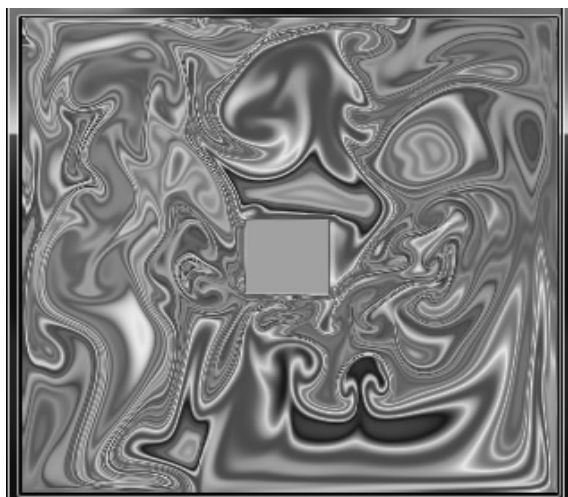


图3 基于第二种纹理方法的烟雾渲染效果

使用纹理渲染烟雾比直接根据烟雾的密度场绘制烟雾速度快,直接绘制的方法在高网格分辨率的情况下实时性较差,同一环境下,直接绘制的方法网格分辨率  $N$  的取值一般在128左右,若  $N$  值增大,渲染速度慢,不满足实时性;反之若采用文中纹理的绘制方法,网格分辨率可达到256左右,烟雾的细节更能体现出来,更逼真。

## 6 结束语

文中虽然通过纹理渲染方法提高了烟雾渲染的速度和真实性,但是文中的方程计算都是基于CPU的,随着网格分辨率的增加,烟雾模拟的速度主要受到方程计算的限制,因此在以后的工作中考虑将方程的计算移植到GPU中,利用显卡计算的并行性,以此来达到流体模拟的实时性和真实性,这是一个具有挑战性的问题,值得研究。

## 参考文献:

- [1] Ebert D S, Musgrave K, Peachy D. Texturing and modeling: A procedural approach [M]. San Francisco: Morgan Kaufmann Publishers Inc., 1994:267-294.
- [2] 王盛邦,纪庆革. 基于粒子系统的实时烟雾仿真[J]. 中山大学学报(自然科学版), 2008, 47(2): 9-13.
- [3] Stam J. Stable fluids [C]//Proc of SIGGRAPH 99. New York: ACM, 1999:121-128.
- [4] Fedkiw R, Stam J, Jensen H W. Visual simulation of smoke [C]//Proceedings of the 28th annual conference on computer graphics and interactive techniques. New York: ACM, 2001:15-22.
- [5] Selle A, Fedkiw R, Kim B M, et al. An unconditionally stable MacCormack method [J]. Journal of scientific computing, 2008, 35(2-3): 350-370.
- [6] Kim B, Liu Y, Llamas I, et al. Flow-fixer: Using BFECC for fluid simulation [C]//Proceedings of eurographics workshop on natural phenomena. Switzerland: Eurographics Association Aire-la-Ville, 2005:51-56.
- [7] Lentine M, Aanjaneya M, Fedkiw R. Mass and momentum conservation for fluid simulation [C]//Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation. New York, NY, USA: ACM, 2011:91-110.
- [8] Lentine M, Gr  tarsson J, Fedkiw R. An unconditionally stable fully conservative semi-Lagrangian method [J]. Journal of computational physics, 2011, 230(8): 2857-2879.
- [9] Matthias D C, Gross M. Particle-based fluid simulation for interactive applications [C]//Proceeding of the 2003 ACM SIGGRAPH/Eurographics symposium on computer animation. Switzerland: Eurographics Association Aire-la-Ville, 2003:154-159.
- [10] 陈志宏,张正兰. 动画烟雾的动态模拟算法研究[J]. 计算机仿真, 2009, 26(2): 217-220.
- [11] Stam J. Real-time fluid dynamics for games [C]//Proceedings of the game developer conference. [s. l.]: [s. n.], 2003.
- [12] Birdson R. Fluid simulation for computer graphics [M]. [s. l.]: Press by A K Peters/CRC, 2008:35-37.
- [13] 褚曦丹,汪继文,刘君. 实时稳定的二维烟雾模拟[J]. 计算机技术与发展, 2011, 21(5): 91-94.
- [14] 谢晓强. 不可压缩流体的边界层问题[D]. 上海:复旦大学, 2008.
- [15] Harris M J. Fast fluid dynamics simulation on the GPU [C]//Proceeding SIGGRAPH'05. New York, NY, USA: ACM, 2005.

基于物理模型的实时烟雾模拟

作者:

[汪继文](#), [杨贤达](#), [WANG Ji-wen](#), [YANG Xian-da](#)

作者单位:

[汪继文, WANG Ji-wen\(安徽大学 计算机智能与信号处理教育部重点实验室, 安徽 合肥 230039; 安徽大学 计算机科学与技术学院, 安徽 合肥 230039\)](#), [杨贤达, YANG Xian-da\(安徽大学 计算机科学与技术学院, 安徽 合肥, 230039\)](#)

刊名:

[计算机技术与发展](#)

ISTIC

英文刊名:

[Computer Technology and Development](#)

年, 卷(期):

[2014\(1\)](#)

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_wjtz201401031.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjtz201401031.aspx)