

# 基于物理的流体模拟动画综述

柳有权<sup>1,3)</sup>, 刘学慧<sup>1)</sup>, 朱红斌<sup>1,3)</sup>, 吴恩华<sup>1,2)</sup>

<sup>1)</sup> 中国科学院软件研究所计算机科学实验室 北京 100080

<sup>2)</sup> 澳门大学科学技术学院电脑与资讯科学系 澳门

<sup>3)</sup> 中国科学院研究生院 北京 100080

[lyq@ios.ac.cn](mailto:lyq@ios.ac.cn), [lxh@ios.ac.cn](mailto:lxh@ios.ac.cn), [zhuhb@ios.ac.cn](mailto:zhuhb@ios.ac.cn), [ehwu@umac.mo](mailto:ehwu@umac.mo)

**摘 要** 近来基于物理的流体模拟成为计算机动画领域中的一个研究热点, 本文回顾了 this 领域中基于物理的流体模拟的发展情况, 总结了该方向所采用的各类方法, 并结合具体现象的特点分门别类详细展开, 其中总体上可以分为欧拉法和拉格朗日法, 涉及的现象包括烟雾、火焰、爆炸、波浪、气泡、以及自由运动界面等. 最后展望了未来发展的三个重点, 即细节策略, 加速策略和控制策略, 以使整个模拟能够很好地满足人们对真实感、实时性以及灵活性的需求.

**关键词** 计算机动画; 纳维-斯托克斯方程; 流体模拟

**中图法分类号:** TP391.41

## Physically Based Fluid Simulation in Computer Animation

Youquan Liu<sup>1,3)</sup>, Xuehui Liu<sup>1)</sup>, Hongbin Zhu<sup>1,3)</sup>, Enhua Wu<sup>1,2)</sup>

<sup>1)</sup> Laboratory of Computer Science, Institute of Software,  
Chinese Academy of Sciences, Beijing, China, 100080

<sup>2)</sup> Department of Computer and Information Science, Faculty of Science and Technology, University of  
Macau, Macao, China

<sup>3)</sup> Graduate School of the Chinese Academy of Sciences, China, 100080

**Abstract** This paper presents a survey on the development of physically based fluid simulation in computer animation, with a detail introduction to the classification of the topics as well as the methods employed in the field. The methods applied mainly include Eulerian method and Lagrangian method, for simulating the fluid phenomena such as smoke, fire, explosion, wave, bubble, free surface etc. Future work is also discussed, including details techniques, acceleration techniques and control techniques used to meet the demands for reality, real-time, flexibility.

**Keywords** Computer Animation; Navier-Stokes Equation; Fluid Simulation

## 1 引言

在图形学领域, 人们一直试图利用计算机再现周围的真实世界, 然而现实世界虽然看上去简单, 却极其复杂. 对于一些自然现象, 人们可以采用过程描述的方式来获得某种特定的效果, 但是对于有些现象人们无法找到一个简单的模型来描述它, 这个时候只有去追溯该现象本身的物理根源, 只有借助其本身较为精确的物理描述才能真实再现其外在的视觉现象. 随着计算机硬件的不断发展, 计算能力不断增强, PC 机能够完成的算法复杂度不断上升, 使得基于物理的计算机动画的迅速发展成为可能. 在市场方面, 不管是电影特效, 还是视频游戏, 人们对于基于物理的计算机动画的需求也很强烈. 这些因素使得基于物理的计算机动画成为目前一个研究热点, 也成为当前和未来计算机图形学发展的一个重点方向.

本文主要探讨基于物理的流体模拟计算机动画这个领域, 其中涉及的现象包括烟雾<sup>[1]</sup>, 流体运

---

**基金项目:** 本文受到国家重点基础研究发展规划项目 (2002CB312102) 和国家自然科学基金项目 (60223005, 60473105), 以及 Research Grant of University of Macau 的资助.

**作者简介:** 柳有权, 男, 1976 年生, 湖北人, 博士研究生, 研究方向为计算机图形学, 计算机动画; 刘学慧, 女, 1968 年生, 湖南人, 博士, 副研究员, 研究方向为计算机图形学; 朱红斌, 男, 1978 年生, 湖北人, 博士研究生, 研究方向为计算机图形学, 计算机动画; 吴恩华, 男, 1947 年生, 江苏人, 博士, 研究员, 博士生导师, 研究方向为计算机图形学.

动界面<sup>[2, 3]</sup>、波浪<sup>[4]</sup>、气泡<sup>[5]</sup>、火焰<sup>[6, 7]</sup>、以及爆炸<sup>[8]</sup>等。这些现象的模拟在计算流体力学(CFD: Computational Fluid Dynamics)领域里也一直是热点话题,这方面的工程应用多如牛毛,商业化软件如 Fluent, CFX 等,都可以用来专门做流体计算分析。那么,我们是否可以利用这些软件来完成计算,然后利用图形绘制技术将流动的效果真实再现出来呢?既然这样,那作为图形学研究又能做些什么呢?主要原因在于 CFD 更强调对问题本身的精确求解,然后给出一个合理的分析;而计算机图形学则力图真实再现实时流动的视觉效果。而且对于某些现象,甚至流体力学的研究者们都还没有给出一个很好的计算模型来描述其视觉效果,需要图形学的研究者去寻求有效的解决方法。此外,在图形学领域,计算的快速性比计算精度重要,小的计算代价可以让人们方便预览可能的效果,从而进行快速调整生成画面。还有就是对控制的加强,计算机图形学必须为动画师提供一个良好的控制策略,这样通过简单交互就可以设计实现想要的艺术效果。当然,二者本质上来并没有多大的冲突。CFD 构成了基于物理的流体模拟的基础,而图形显示只是最终的目标而已。正如 Carlson 等所认为的那样,编程的容易程度,小的计算代价,可控性,对障碍物处理的难易程度,以及对水或者其他流体的自由表面表达的方便程度,这些因素指导着图形学研究者们从 CFD 中寻求相应的手段<sup>[9]</sup>。

在三维图形软件产品中,也有不少包含了流体模拟的模块。如 Maya 里面的 Maya Fluid Effects™,该模块的方法主要来自 Jos Stam 的论文<sup>[10, 11]</sup>;在 3DS Max 里面则提供有 glu3D 流体插件;独立的软件包如 Next Limit 推出的 RealFlow 和 RealWave 更是出类拔萃,被称为 PC 机上最好的两种流体动力学模拟软件。另外还有一些其他的插件。

此外,这里需要提一下的是流体绘制方面的问题。对于火焰、爆炸这类现象,温度占很重要的位置,这个时候黑体辐射不可忽视,而且还需要考虑亮度过高带来的色适应问题<sup>[6]</sup>。而对于烟雾之类的常温现象<sup>[1]</sup>,其密度场就可以完全表现了。对于烟、火这种参与媒介的绘制通常采用体绘制技术就可以解决。而对于自由运动界面这类问题,则通常需要采用 Marching cubes 方法重新构造出界面多边形,然后采用几何体的绘制方法就可以了,如光线跟踪。因此流体绘制方面的技术取决于计算机图形学领域里的其他方面的进展。而本文关注的是流体模拟中的基于物理的运动建模过程,从当前计算机图形学真实模拟的研究入手,给出一个基于物理的流体模拟综述。

## 2 流体模拟研究方法的发展

早期的流体模拟,由于计算能力有限,主要采用参数建模的方法。如文献<sup>[12]</sup>通过将波浪函数表示成一系列线性波型的组合,更进一步将各个波型简化为波形和相位的组合函数,从而合成浅水表面高度场,能处理波折射问题,并采用粒子系统来模拟当波浪破碎或者碰到障碍物的时候形成的浪花。为克服单纯高度场不能模拟水面折叠效果的问题,文献<sup>[13]</sup>基于 Gerstner 模型,采用拉格朗日粒子来模拟波浪参数表面,通过海底深度和坡度来控制正弦函数波形。但在这两篇文献中,表示水的粒子或者网格只是在其初始位置附近运动,所以它们都无法表现真正的流动效果,也无法处理边界给水面带来的影响。又如基于统计的 FFT 经验模型<sup>[14]</sup>可以很好地描述波幅较小的海平面。但是对于以上这些模型,人们觉得控制起来很困难,而且不能模拟一些复杂的、细节更为丰富的效果,于是很多研究者转向基于物理的方法。基于物理的方法主要分为两种:第一种方法是从研究流体所占据的空间中各个固定点处的运动着手,分析被运动流体所充满的空间中每一个固定点上的流体的速度、压强、密度等参数随时间的变化,以及研究由某一空间点转到另一空间点时这些参数的变化,该方法被称为欧拉法,是一种基于网格的方法;第二种方法是从分析流体各个微团的运动着手,即研究流体中某一指定微团的速度、压强、密度等描述流体运动的参数随时间的变化,以及研究由一个流体微团转到其他流体微团时参数的变化,以此来研究整个流体的运动,被称为拉格朗日法,是一种基于粒子的方法。

### 2.1 欧拉法

描述流体现象最为完整的为纳维-斯托克斯方程(NSE: Navier-Stokes Equation), 该方程是根据牛顿第二定律推导出来的, 详细资料可以参阅文献[15]. 下面给出了常用的不可压粘性 NS 方程组的欧拉形式,

$$\text{质量方程} \quad \nabla \cdot \mathbf{u} = 0$$

$$\text{动量方程} \quad \frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} - \nabla p / \rho + \mathbf{f}$$

这里  $\rho$  为密度,  $p$  为压强,  $\mathbf{f}$  为体积力,  $\mathbf{u}$  为速度,  $\nu$  为运动粘性系数, 跟动力粘性系数  $\eta$  的关系为  $\nu = \eta / \rho$ . 基于网格的欧拉法即是将上述方程离散到网格上, 然后计算各个固定网格节点上状态量的变化, 从而来得到整个场. 这里有两种思路进行网格化, 一种是交错网格, 即一般情况下标量, 如压强, 分布在网格单元的中心, 而速度之类的量分布在单元表面, 这种离散的好处是容易保证守恒性条件, 目前多采用此思路<sup>[10]</sup>; 另一种思路则是所有的量都处于同一个位置, 这种方法简单, 不需太多的插值运算, 对各个变量也无需区别对待, 如[16].

早期为了真实地描述流体的运动, [17]引入浅水场方程求解高度场, 通过采用隐式格式构成三对方程组, 从而快速稳定地求解. 同样[18]采用高度场来描述流体表面, 借助假想的各个节点之间的管道, 来计算各个立柱的体积, 以得到表面位置, 物体对流体表面的影响通过相互作用力施加上去, 飞溅效果采用粒子系统来完成. 为了更贴近物理本质规律, 文献[19]采用二维 NS 方程来求解表面速度场, 然后对于流体表面根据压强伯努利方程求解出高度场来表示, 其中模拟了运动物体和其他障碍物对于流体的影响. 采用高度场计算的好处就是整个模拟二维化, 避免了三维复杂耗时的计算, 但是效果不尽理想.

真正采用三维 NS 方程来模拟流体运动始自文献[20], 其中利用 MAC(Marker and Cells)求解流体, 但由于采用显式格式, 时间步长必须满足 CFL 条件(Courant-Friedrichs-Lewy Condition)以使整个计算收敛. 文献[10]采用半拉格朗日法求解对流项, 并结合隐式求解器, 从而保证计算绝对稳定. 自此, 采用 NS 方程来模拟真三维的计算机动画越来越普遍.

## 2.2 拉格朗日法

而如果采用拉格朗日法这种非基于网格的方法, 即基于粒子系统的方法, 则 NS 方程可以写为,

$$\frac{D \mathbf{u}}{D t} = \nu \nabla^2 \mathbf{u} - \nabla p / \rho + \mathbf{f}$$

如果将右边整理成一个力, 则退化为牛顿第二定律:

$$\mathbf{a}_i = d\mathbf{u}_i / d t = \mathbf{f}_i / \rho_i$$

这里  $\mathbf{a}_i$  为粒子  $i$  的加速度,  $\mathbf{u}_i$  为粒子  $i$  的速度,  $\mathbf{f}_i$  为该粒子受到的合力,  $\rho_i$  为该粒子所在位置的密度. 显然, 该类方法就是对于各个相对独立的粒子进行力的分析, 通过积分计算出这些粒子下一个时刻的位置和其他状态量.

Reeves 引入的粒子系统<sup>[21]</sup>, 因为对于非规则物体具有灵活的表现能力, 因此被用来模拟流动<sup>[22]</sup>. 除此之外, 对于流动中出现的飞溅<sup>[18]</sup>、泡沫<sup>[23]</sup>很容易想到采用粒子系统来表现.

文献[24]首先在图形学中引入 SPH(Smoothed Particle Hydrodynamics) 方法来模拟火和其他气态现象. SPH 方法是粒子系统的一种插值方法, 通过引入平滑核来表示周围粒子的影响, 该方法同时被用来求解柔性物体的大变形<sup>[25]</sup>. 文献[26, 27]通过 SPH 来模拟流体自由表面, 同时考虑表面张力的作用, 其中采用 Point splatting 技术, 或者 Marching cubes 算法来绘制表面. [28]则采用 MPS(Moving Particle Semi-Implicit)方法以及 MPS-MAFL 方法模拟多种类型的流动, 该方法将 NS 方程转化为运

动粒子相互作用的形式.

近两年, 文献 [29]将 LBM(Lattice Boltzmann Model)引入图形学领域, LBM 方法也是一种拉格朗日方法, 它不去追踪每一个实际粒子, 而是对粒子分布函数的演变进行描述. 在离散的格子里, 粒子沿着格子轨线向相邻的格子迁移和相互碰撞, 这样分布函数的演变就决定了流体运动的变化过程. 该方法相对于欧拉法的优点在于编程容易, 更容易并行化, 可以很方便地处理复杂边界<sup>[29, 30, 31, 32]</sup>.

拉格朗日法的方法的优点为容易表达, 不需要对整个空间进行处理, 而且容易保证质量守恒, 而且比较容易实施控制. 但拉格朗日法对于平滑运动界面的重建比较难处理, 而且自由界面拓扑的改变必须采用复杂的算法才能构造出该表面的几何, 计算量随着粒子数的增多而加大.

欧拉法和拉格朗日法各有优缺点, 为了更真实地模拟流动, 基于网格的欧拉算法往往结合拉格朗日的粒子算法一起使用, 比如得到广泛应用的半拉格朗日算法<sup>[10]</sup>, 或者用来对流体进行约束, 从而加强交互控制<sup>[33]</sup>.

### 3 流体模拟问题的分类

与流体相关的内容很多, 但由于计算机动画关注的只是人眼可以感知的现象, 因此目前主要的研究集中在低速流, 低速流给人有足够的响应时间, 观赏性强. 下面我们将根据一些目前已有的专题来进行讨论, 如烟雾与云彩, 燃烧与爆炸, 自由运动界面, 混合流和多相流, 非牛顿流体以及流固耦合等, 但是这些问题本身并不孤立存在, 例如非牛顿流体的模拟同样要涉及自由运动界面的问题, 这里为了讨论方便, 我们将分别展开. 另外由于目前整个发展的重点在于基于 NS 方程的方法, 所以这里也主要围绕此来阐述之.

#### 3.1 烟雾与云彩

烟雾问题<sup>[1]</sup>可能是流体现象模拟中最为简单的一类, 不存在自由运动界面的问题, 也没有其他什么特别需要处理的. 云彩的运动跟烟雾类似, 不过为了更真实地模拟云的变化, 文献[34, 35]引入云彩形成的物理机理, 通过热力学作用考虑空气中水分含量的影响.

#### 3.2 燃烧与爆炸

最常见的燃烧现象就是火焰, 它是一种低速流动燃烧的过程. 目前最为精确的模拟方法来自[6], 采用全 NS 方程来求解, 并考虑了燃烧过程. 而[7]则更倾向于为动画师提供更为灵活的控制, 采用了非严格基于物理的方法, 基于粒子系统, 通过基本的火焰轮廓来构造初始状态, 在人工风场和浮力的作用下运动, 加上过程噪声和 Kolmogorov 湍流噪声来形成最终的火焰形状.

文献[36]模拟了火焰在三角形表面的传播过程, 通过采用一个火焰集合表示火, 每一个火焰即为一个可变形的顶点链条, 通过该骨架构造出火焰的外形, 而火焰传播的边界通过一条封闭曲线来表示. [37]模拟了火焰在体数据表示的物体上蔓延的过程, 其传播基于距离场, 且燃烧对物体存在消耗, 其中基于 LBM 方法加入了风对火焰的作用.

快速燃烧就会形成爆炸现象. 爆炸最主要的后果是冲击波, 它以超音速传播, 推动或者撕裂物体, 改变光线折射, 而次后果才是爆炸引起的火球, 尘云等. [8]模拟了冲击波的效果, 冲击波会影响气体的密度, 从而改变折射率, 导致图像扭曲, 整个求解基于三维可压粘性流方程, 并引入能量守恒方程. 但对于视觉来说, 主要效果来自次后果. 为了模拟悬浮粒子造成的爆炸, [38]中结合粒子系统和不可压无粘 NS 方程, 着重模拟了爆炸次后果, 并引入了再次燃烧这个过程, 根据燃烧产物对散度场  $\nabla \cdot \mathbf{u} = \phi$  进行调整, 来模拟气体膨胀过程.

#### 3.3 自由运动界面

对于烟雾、火焰等现象的模拟, 视觉效果只是与密度场和温度场相关, 我们关注的是体的问题;

而对于液体流动，则需要去求解其自由表面，才能将其与周围的环境分开，从而施加对应的光照效果。

对于波浪这种大规模场景，正如[39]所说，波浪共有三个尺度的效果，精细尺度为飞溅和泡沫，中间尺度为水面的波纹，大尺度上为波浪的翻转和破碎。目前大部分计算机动画方面的文献主要处理的是中间尺度的水面<sup>[4, 14]</sup>，这种尺度下的模拟多采用参数建模的方法，或者 FFT 反变换生成海洋表面的小片。而精细尺度下的，如[23]，则需要采用 NS 方程组求解的全数值算法，才能得到流水冲击引起的泡沫和飞溅的真实效果。而大尺度下的波浪模拟，如[39]，同样也需要利用 NS 方程才能很好地描述波浪破碎的现象。作为海平面来说，其形态变化多端，很多研究者只是侧重于模拟某种现象，而文献[40]则采用 LOD 思路将多个模型结合起来，在不同的地方采用不同的策略以满足实际的需要。

除了像波浪这种大规模场景外，人们对于小尺度的液面模拟也很关注，这些运动界面跟海水表面相比，其拓扑随时会发生变化，边界条件也要复杂很多。在基于网格的方法中，目前已经得到广泛应用的方法包括 MAC, level set, front tracking, ghost fluid, volume of fluid, particle level set 等。

文献[20]采用 MAC 方法来表现流体表面，即在每个单元里分布一些无质量标记(Marker)，这些标记在速度场的推动下运动，通过判断这些单元里是否有这些标记来了解是否含有流体。从而依据标记个数的多少利用 marching cubes 算法重构出自由表面。[9, 41]采用了类似的方法。

Level set 方法<sup>[2]</sup>引入一个距离函数  $\phi$ ，定义  $\phi$  为各点到分界面的有符号的垂直距离。这种距离函数，在速度场的推动下移动，如下：

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0$$

$\phi = 0$  即为两种流体的分界面位置，通过  $\phi > 0$  和  $\phi \leq 0$  将界面内外具有不同状态方程的流体分开。

其表面单位法矢为  $\mathbf{n} = \nabla \phi / |\nabla \phi|$ ，曲率为  $k = -\nabla \cdot \mathbf{n}$ 。由于 level set 的体积损失比较严重，[2]在液体内部分布一些无质量粒子来弥补该缺陷，平滑的地方的粒子影响力可以忽略，而在尖锐的地方用粒子来修改当地的  $\phi$  值，该方程的求解采用了高阶迎风格式。

更进一步，从基于体的建模<sup>[2]</sup>转到基于表面的建模<sup>[3]</sup>，引入 particle level set 方法，在表面内外一个很窄的区域都分布若干粒子，利用这些粒子来侦测界面处的不精确度，通过粒子局部的 level set 函数  $\phi_p$  来校正  $\phi$ ，从而保持界面原有的尖锐特征。通常通过将速度外推到空气中以正确处理界面处的速度，获得更为精确的视觉效果。类似地，文献[42]采用 particle level set 来捕获自由运动界面，作者采用 2 倍分辨率以减轻体积损失的问题。文献[43, 44, 45]也是采用 particle level set 来捕获自由运动界面。Particle level set 兼有 front tracking 质量守恒的优点和 volume of fluid 灵活性，以及 level set 的易用性<sup>[46]</sup>。

Volume of fluid(VOF)方法通过定义每一个单元的流体体积比来捕获自由运动界面，即

$$f_t + \mathbf{u} \cdot \nabla f = 0$$

方程形式跟 Level set 类似，但含义不同， $f$  代表的是流体体积比函数。该方法的优点是满足体积守恒，文献[23]结合 CIP(Cubic Interpolated Propagation)方法求解 VOF 方程以进一步提高精度。[39]采用混合 level set 和 VOF 方法(CLSVOF: combined level set and volume of fluid)以综合两者的长处。[5]则是结合 VOF 和 front tracking 方法来获取气泡的表面，也就是空气和液体的分界面，其中 front tracking 方法

用来减小扩散效果.

Level set, Front tracking, Volume of fluid 的缺点<sup>[46]</sup>是无法处理界面处的密度、粘性和压强的不连续性,以至不能表现表面张力的作用,为了克服这个毛病,必须引入其他源项.而 ghost fluid 法则可以避免界面处的平滑处理,该方法利用函数  $\phi$  将含有两种流体的流场分成各含有一种流体的两个流

场.设界面内的流体为流体 2( $\phi \leq 0$ ),界面外的流体为流体 1( $\phi > 0$ ).对于一个只含有流体 2 的流场(称之为流场 2),界面内的点为真实点,具有真实的物理量;而界面外的点为流体 2 的虚拟点,其物理量为界面内真实点物理量的延拓.如[6]采用 level set 来追踪气体反应区的移动隐式曲面,为了解决两种不同的气体计算过程出现的界面法向速度不连续的问题,引入幽灵(Ghost)速度值,利用切向速度的连续性来得出正常值.对于界面压强的不连续,采用 ghost fluid 法来处理细薄火焰头部. Ghost fluid 方法的出现,成功地解决了接触间断的模拟问题,该方法没有数值耗散,能较好地消除非物理的振荡,且能轻易地处理接触间断两边具有不同状态方程的流体.

基于拉格朗日方法的流体表面捕获算法,如[26, 47],需要先确定那些在流体表面的粒子,然后根据这些粒子构造等值曲面,然后多边形化.文献[28]则是构造粒子系统下的 level set 方程,基于网格来重建自由表面.

关于自由运动界面的计算方法可以参考文献[48].

### 3.4 混合流和多相流

若流动系统中存在多种流体称为混合流,而若相态多于一个,则称为多相流.对于火山云, [49]认为其由两种流体构成,一个为岩浆(包括固体物质和火山气体),一个是拖曳的气体,其对应质量分数在速度场的作用下运动.其中忽略固体物质所占体积,这两种流体的耦合根据混合气体状态方程得出密度公式,简化为质量比的函数,从而通过浮力作用施加到 NS 方程上.

文献[5]通过气泡在水中的运动模拟了空气、水两相流问题,其中采用 VOF 结合 Front Tracking 来捕获气泡表面,通过物质场采用最小应力表面张力方法来计算表面张力,然后通过 Marching cubes 算法构造界面多边形,并通过修正因子来确保总体质量守恒. [28]基于 MPS 方法模拟了两种截然不同的液体混合的过程,不过这两种液体并不相互渗透和发生反应.

由于爆炸物中粒子和气体存在相互作用, [38]通过对每一个粒子施加一个由于速度差导致的拖曳力作用,反作用力施加到对应的流体单元上.

$$\mathbf{f} = \alpha_d r^2 (\mathbf{u} - \dot{\mathbf{x}}) \|\mathbf{u} - \dot{\mathbf{x}}\|$$

$\alpha_d$  为拖曳力系数,  $\mathbf{u}$  为粒子位置的流体速度,  $\dot{\mathbf{x}}$  为粒子速度,  $r$  为粒子半径.除了力的相互作用外,还有热量的相互传递过程,形式跟力类似.

两种不同的物质除了简单混合外,可能还会发生化学反应,其中燃烧现象都可以归到这一类.如在[6]中,认为燃气首先混合好,然后在蓝色中心界面处发生化学反应,生成高温气体产物.这里对于燃气和燃烧气体产物各自采用了一套不可压 NS 方程组来进行模拟,关键问题是确保各个界面处的质量守恒和动量守恒.同样也模拟了固体可燃物转变为燃气的过程,但是没有去模拟具体的燃烧反应.

文献[50]则模拟了更一般化的气态物反应过程.但这里的化学反应过程考虑的较为简单,其中反应热简化为与反应速度成线性关系,通过设计不同的速度散度场可以得到不同的反应效果.

### 3.5 相变和非牛顿流体

如果流动行为满足牛顿粘性定律,即剪应力与剪切应变速率之间满足线性关系,粘性不随速度

而变,称为牛顿流体,如水,空气等.而把不满足线性关系的流体称为非牛顿流体,如胶状物,牙膏等.

文献[51]建立物体的弹簧质量模型,然后将弹簧刚度和温度关联起来,模拟了热弹性变形的过程.[52]认为柔性物体是一个封闭不可穿透的弹性细胞膜,该细胞膜包含了粘性可压液体.物体表面采用弹簧质量系统来表示,内部利用 NS 流体运动方程求解变形,其中力项来自固体各个节点的弹力和环境压力,从而可以模拟柔性物体运动变形情况.

文献[9]结合 MAC 方法模拟了任意粘性的流体,甚至固体也可以当作流体处理,其中粘性随温度、水分含量而变化,从而实现了融化、凝固和流动效果.通过将粒子分割到高分辨率的体里来抽取精确表面.具体求解中,为了保证稳定对于扩散项采用隐式欧拉代替显式格式,利用共轭梯度法来迭代计算.在求温度扩散项的时候采用了比共轭梯度法快的三对角求解方法.

而对于火山熔岩的流动过程,文献[53]中模拟了传热过程,并将粘性和温度关联起来,没有涉及状态的变化.采用 SPH 方法来构造熔岩,在流动过程中,粘性随温度下降呈指数上升.

文献[44]通过在 NS 方程中加入弹性应力张量  $\mu_e \nabla \varepsilon / \rho$  成功地模拟了粘弹性流动现象,并采用 von Mises 判据来判断是否发生塑性流动,如果应变偏差的 Frobenius 范数值大于屈服点,则从弹性体变为流体,同时以正比于超出量的速度流动.

文献[47]基于拉格朗日粒子算法,将流体与物体之间的力组织成粘附矩阵的结构,通过将物体表面也粒子化表示,从而可以很方便地表达粘性流体与物体的相互作用.

然而精确模拟融化现象需涉及到传热和相变物理过程,计算过于复杂,故有人采用一些简单模型来模拟这类现象,如[54]采用三维自动格子机(Cellular Automata)的方法,通过重力和传播两个变换规则来传递热量和流体,这种方法的计算量相对于完全基于物理的方法要小很多.

### 3.6 流固耦合

流固耦合问题简单来讲就是如何考虑流体和固体之间的相互作用,比如水中漂浮的物体,风中飞扬的旗帜.很多文献只处理了单向作用,如固体对流体的作用,即固体的运动是已知的,利用它来改变流体的运动状态,而固体本身的运动不受流体影响;或者流体对固体的作用,即固体在流体的作用下发生运动,但固体的运动却不影响流体.

文献[55]采用基本流动元素组合的方法模拟了树叶在风中飞舞的情景,[30]采用 LBM 模拟了气泡和羽毛在风中的运动,因为针对的都是轻盈的物体,所以都只考虑了物体受到的流体作用,而没有考虑流体受到的影响.[20]通过压强梯度计算流体中的刚性物体受到的力的作用,然后通过拉格朗日方程来计算这些刚性物体的运动,同样没有考虑物体对流体施加的力的作用.

相比之下,只考虑固体对流体的影响似乎要更简单一些,因为通常固体的运动规律跟流体不同.如在文献[1, 2, 3]中,固体的运动是已知的,这样通过修改固体所占网格上的状态量以反映流体变化.

显然,只考虑单向的作用并不足以表现真实的运动,所以很多研究者开始进行流固耦合的思考.在[18]中物体与流体的相互影响通过相互作用力施加的,不过这个作用力是通过试探性得到的.[8]在模拟爆炸的过程中,不仅考虑了流固的相互作用,还考虑了因为冲击波导致的物体破碎.该文献中固体采用了两种表示方式,一种为几何体表示,用来求物体受的力,另一种为体素表示,用来求流体的位置.固体受到的力来自静压和流固两者速度差的平方,而固体对流体的影响则通过每一个体素内气体容积的改变来施加作用.

文献[41]采用 MAC 方法来模拟流体流动,NS 方程的求解采用半拉格朗日法求解对流项.而固体被看作一个质点相连的集合,采用弹簧质量模型(spring-mass model)来表示,两个节点之间的作用力采用粘弹性模式来描述,而方向即为两个节点的位置差,根据牛顿第二定律来计算粒子新的位置.对于流固耦合通过界面来进行,即通过这些与固体接触的无质量标示(Markers)来传递力的作用,这个力也采用粘弹性描述,直接施加在固体节点上,而对于流体,则在上面得出的界面标示作用力的

基础上通过位置划分施加到对应流体单元上. 该方法的缺点是力场离散的太粗, 时间步长不能太大.

文献[43]提出刚性流体(Rigid Fluid)方法, 即将刚体作为流体的一部分, 并融入到流体的整个计算过程中, 采用拉格朗日因子来耦合两者的相互作用. 其中刚体求解器通过约束其运动为围绕质心的平动和转动, 隐性地实施了固体的刚性要求, 增加的计算量与刚体的个数线性相关.

而为了处理变形固体与流体的耦合作用, 文献[27]根据高斯积分在固体表面分布若干边界粒子, 其中流体采用 SPH 方法来表述, 而固体则采用有限元(FEM)方法求解, 通过这些虚拟粒子, 流体粒子与固体三角形面片的相互作用转化为粒子与粒子的相互作用, 粒子间的排斥力和吸附力采用类 Lennar-Jones 模型.

### 3.7 非真实感流体

这里所讨论的非真实感并非光照上的非真实感处理, 而是指采用流体模拟的思路来实现一些自然界中并非存在的现象, 或者我们可以称为艺术流动, 这些现象很有趣. [56]实现了任意拓扑结构上的流动现象模拟, 结合 Catmull-Clark 细分曲面将 NS 方程变换到一般坐标系下的参数空间. 而[57]则更进一步推广到三角形拓扑上, 而且不需要参数化过程以避免参数扭曲的可能.

为了满足艺术家的一些假想, 文献[58]引入图像做为温度源项, 采用二维可压 NS 方程组模拟流动效果. 为了让流动能按照艺术家的想法来运动, [59]提出关键帧的方法, 通过指定烟雾密度和速度的关键帧, 建立作用力的目标函数, 利用准牛顿优化算法使得满足约束, 这里不仅计算变量, 而且还要计算变量的导数, 因此算法很慢. 为了改善效率, [60, 61]采用伴随方法(Adjoint Method)从而提高了导数求解的效率, 但这个方法又增大了存储量的要求. 跟前两者不同的是, [62]并不追求对关键帧的精确匹配, 该方法在 NS 方程中引入  $v_f F(\rho, \rho^*)$  和  $-v_d u$  两项, 第一项中  $F$  为驱动力, 其跟

关键帧密度场的梯度成正比, 第二项为衰减项. 同时在密度对流方程中引入聚项  $v_g G(\rho, \rho^*)$  来消除数值耗散. 由于该方法不是精确匹配, 所以虽然计算速度要快, 但在效果上较前面两种方法差一些. 文献[63]则通过在 NS 动量方程中添加  $-\nabla U_{control}$  项利用几何势能来控制其流动从初始状态到某目标状态, 这里几何势能场可以在预处理中完成, 因此计算量要小很多.

而最能体现艺术家设计意图的, 无疑是非真实感的交互绘画系统了, 如[64]利用浅水场方程结合三层模型模拟水彩颜料在纸面上的运动, 从而得到水彩画风格, 但是由于条件收敛导致速度很慢. 文献[65]则求解粘性流动模型以模拟厚涂颜料的绘画, 其中还结合 VOF 方法求解这种高粘性流体的自由表面, 由于结合了硬件加速, 对于低分辨率可以达到交互程度. 文献[66]则直接在[1]的基础上, 结合深度差异的非真实感绘制算法再现了烟雾动画的卡通效果.

## 4 研究动向

尽管基于物理的流体模拟的整个基础是 CFD, 但由于关注的最终目标存在差异, 所以研究的重点有所侧重. 对于计算机动画来说, 人们期望更多的细节特征, 人们期望能找到一种快速稳定的求解算法, 这样保证人们在交互设计的过程中就能观察到效果, 从而对效果不断进行调整直至满意; 另外, 人们期望能够对流体的运动进行充分的控制, 使其流动能够满足人们的意愿或者想象, 而不必关心背后的物理.

### 4.1 细节策略

对于细节的加强是出于人们对真实感的要求, 细节的损失会导致严重地失真. 例如在烟雾模拟中, 由于存在数值耗散或者期望流动富于变化, 需要加强湍流效果. [67]采用 Kolmogorov 能量谱模拟微观场, 然后叠加到宏观场上, 文献[1, 2, 6]采用漩涡约束  $\omega = \nabla \times \vec{u}$  来增强细节, 而在[68]中则



采用了更复杂的  $k-\varepsilon$  湍流模拟来加强效果. 另外[1]在模拟烟雾的时候采用高阶插值以提高精度.

而对于自由表面, 采用 **particle level set** 代替单纯的 **level set** 即是一例, 因为前者可以保持自由表面的尖锐特征.

今年 SIGGRAPH 的几篇流体方面的文章无不体现着这一点, 如文献[69]通过将拉格朗日涡粒子和欧拉网格结合的方法来增强湍流效果. 而文献[70]则考察了小尺度流动中不连续性, 通过压强阶跃条件和速度梯度的不连续性来对表面张力建模. 文献[71]通过对接触角的分析来解析小尺度下的表面张力, 引入虚拟面来改善三相交接处的力, 从而得到更为真实的效果.

显然, 随着研究的深入, 人们会去关注更多的细节, 因为只有细节的突出, 才会体现的更真实. 而细节本身的模拟又来自对于物理模型的更为精细准确的描述和求解.

## 4.2 加速策略

显然, 首要的加速策略是寻求大时间步长模拟的方法, 这样可以大大减少中间步的数量. 文献[10]引入的半拉格朗日方法无疑是一个突破, 结合隐式迭代的方法保证了整个算法的绝对稳定性, 从而可以以任意时间步长模拟流动. 另一个策略就是减少计算每一步的计算量, 对于基于网格的方法, 采用有限差分方法求解 NS 方程时, 通常采用笛卡尔正交网格, 整个计算区域均匀离散. 但很多时候计算域边界很复杂, 如果采用正交网格, 则为了保证计算精度, 需要在突变的区域划分很精细才行. 在 CFD 领域通常采用贴体网格或者非结构化网格的思路, 设计一种与求解域边界的曲面形状一致的网格以减少总的网格数. 在计算机动画领域, [44]采用八叉树结构来离散整个计算域, 在细节区域分布较密的网格以捕获小尺度的细节特征. 为了能够模拟超大规模的问题, 减少计算量, [72]则是直接采用二维 NS 模拟的方法, 然后结合一个中等尺度的 Kolmogorov 速度场来近似逼近三维情形. 而在[45]中则采用移动网格窗口(Moving Grid Windowing)技术, 将计算区域限定在只感兴趣的地方, 更进一步, [73]基于伽利略不变性(Galilean Invariance), 采用自适应构架来追踪动态网格, 并可以调整整个计算域形状, 去掉那些对计算影响很小的体元, 从而避免整个场景的网格化. 高的计算精度<sup>[23]</sup>也可以在保持效果的前提下减少计算的网格数, 中间迭代算法的高效性同样也决定着每一个时间步的计算效率, 如效率较高的共轭梯度法就得到广泛应用<sup>[1]</sup>.

随着硬件的发展, 尤其是图形硬件可编程性的出现, 许多研究者采用 GPU 来加速整个计算. 如[34, 35, 74]利用 GPU 来加速求解云彩运动, 文献[75]给出了一个很好的 GPU 实例. 类似的工作如[76, 77]. 为了处理复杂的边界条件, [16, 78]采用形成修正纹理和偏移纹理的方法来处理可能出现的复杂边界, 保证了整个基于 GPU 计算的高效性.

对于非基于网格的算法, 目前由于 GPU 开始支持绘制到顶点的功能, 这样人们可以利用 GPU 来构造粒子系统, 通过像素程序对粒子的位置和其他状态值进行更新, 计算出新的顶点位置, 直接送给顶点处理器, 而不需要将这些数据返回给主内存操作, 从而大大提高效率, 如[79, 80], 两者略有不同, 前者基于 nVidia 显卡, 后者则是在 ATI 显卡上的实现, 该方法借助了 ATI 显卡的 Superbuffer 的功能.

显然, 除了伴随硬件性能的不不断提升带来的自然加速, 更多的研究者将针对新硬件的体系结构来设计整个算法, 以充分利用硬件新特性带来的好处, 另外结合算法本身的不断创新使整个流体模拟向实时性不断迈进.

## 4.3 控制策略

只有为流体模拟提供灵活的控制, 动画师才能够设计出想要的结果. 显然, 基于 NS 方程组, 我们可以对方程中各个控制参数或者初始值进行调整, 如流体本身的属性, 粘性系数; 或者对边界条件进行调整, 通过设置各种障碍物和外部作用力来让流动满足动画师的意愿, 如通过空气这边的外推速度和风速, 可以得到风力拖曳流体表面的效果[3]. 但是动画师更关注高级别的控制, [81]提出控制器的概念, 这些控制器并不一定物理严格, 但给设计者带来很大的灵活性, 通过对外部压

强,以压强梯度体现的表面张力,或者内部压强,甚至是速度的处理来加强控制.而[59, 61]通过基于梯度的优化算法来设计力的作用,以控制整个流动来满足关键帧要求,前面讨论的非真实感流体问题都属于控制策略的问题.

文献[45]通过控制粒子的方法来对速度,粘性, level set 和速度散度进行控制,并提供了两种基本控制形状(球形各向同性控制和圆柱各向异性控制),而每个粒子的影响量则通过三种下降曲线来体现,在具体实施控制的时候又提供了软控制或者硬控制两种模式,从而为整个流动提供了一种灵活的控制机制.类似地,[33]则采用迁移径向基函数(ARBF: Advected Radial Basis Function)的方式来增强对流体模拟的控制,而 ARBF 即为采用粒子来对流体进行重新参数化的过程,通过改变粒子的位置达到控制流体的目的.

文献[39]则提出通过构建二维垂直切片库的方式,为动画师提供了一个灵活的三维波浪破碎模拟的交互方式.[7]为了模拟形态各异的火焰,同样也采用了为用户提供多种标准的火焰外轮廓的方式.这种库的模式可以为动画师提供一些预定义的模版或初始状态,从而减小了设计的工作量.

灵活的控制策略使得算法可以很好地体现艺术家的设计意图,而不需要这些艺术家去懂得算法背后的物理问题,也使得整个软件系统变得更为易用.

## 5 结语

基于物理的流体动画研究一直是计算机图形学中的一个热点,尤其是这几年的发展,使得不管是真实感效果上,还是从模拟的复杂程度上都比以前有了很大的提高,但是算法速度能够满足交互程度的并不多.

Industrial Light and Magic 的动画总监 Hal Hickel 说,过去的十年计算机图形业一直专注于制造烟雾、水火的真实效果,而现在这个行业开始转向设法创作面部表情、毛发和衣服运动的真实效果[82].但是尽管如此,由于基于物理的计算复杂度的限制和光照计算的耗时性,设计一个交互设计更为方便、能达到实时性、能几乎完全真实地体现自然界中各种流动现象的系统,仍然是一个很有挑战性的问题,仍然存在很多课题值得深入研究.

## 参考文献

- [1] Ronald Fedkiw, Jos Stam, Henrik Wann Jensen. Visual Simulation of Smoke[A]. In: Proceedings of SIGGRAPH[C], New York: ACM Press, 2001.15~22.
- [2] Nick Foster, Ronald Fedkiw. Practical Animation of Liquids[A]. In: Proceedings of SIGGRAPH[C], New York: ACM Press, 2001.23-30.
- [3] Douglas Enright, Stephen Marschner, Ronald Fedkiw. Animation and Rendering of Complex Water Surfaces[J]. ACM Transactions on Graphics, 2002, 21(3): 736~744.
- [4] Damien Hinsinger, Fabrice Neyret, Marie-Paule Cani. Interactive Animation of Ocean Waves[A]. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation[C], San Antonio, Texas, New York: ACM Press, 2002.161~166.
- [5] Jeong-Mo Hong, Chang-Hun Kim. Animation of bubbles in liquid[J]. Computer Graphics Forum, 2003, 22(3): 253~262, September 2003.
- [6] Duc Quang Nguyen, Ronald Fedkiw, Henrik Wann Jensen. Physically Based Modeling and Animation of Fire[J]. ACM Transactions on Graphics, 2002, 21(3): 721~728.
- [7] Amauld Lamorlette, Nick Foster. Structural Modeling of Flames for a Production Environment[J]. ACM Transactions on Graphics, 2002, 21(3): 729~735.
- [8] Gary D. Yngve, James F. O'Brien, Jessica K. Hodgins. Animating explosions[A]. In: Proceedings of SIGGRAPH[C], New York: ACM Press/Addison-Wesley Publishing Co., 2000.29~36.

- [9] Mark Carlson, Peter J. Mucha, R. Brooks Van Horn, III, Greg Turk. Melting and flowing[A]. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation[C], New York: ACM Press, 2002.167~174.
- [10] Jos Stam. Stable Fluids[A]. In: Proceedings of SIGGRAPH[C], New York: ACM Press, 1999.121~128.
- [11] Jos Stam. Real-Time Fluid Dynamics for Games[A]. In: Proceedings of the Game Developer Conference[C], 2003.
- [12] Darwyn R. Peachey. Modeling Waves and Surf[J]. Computer Graphics, 1986, 20(4): 65~74.
- [13] Alain Fournier, William T. Reeves. A Simple Model of Ocean Waves[J]. Computer Graphics, 1986, 20(4): 75~84.
- [14] Jerry Tessendorf. Simulating Ocean Waves. SIGGRAPH '99 Course Notes & SIGGRAPH'2000: Course Notes 25: Simulating Nature: From Theory to Practice, Pages: 3.1-3.18.
- [15] Daozhi Liu, Computational Fluid Dynamics[M], Beijing: Beihang University Press, 1989(in Chinese)  
(刘导治, 计算流体力学基础, 北京: 北京航空航天大学出版社, 1989.)
- [16] Youquan Liu, Xuehui Liu, Enhua Wu. Real-Time 3D Fluid Simulation on GPU with Complex Obstacles[A]. In: Proceedings of Pacific Graphics 2004, Seoul, 2004.247~256.
- [17] Michael Kass, Gavin Miller. Rapid, Stable Fluid Dynamics for Computer Graphics[J]. Computer Graphics, 1990, 24(4): 49~57.
- [18] James F. O'Brien, Jessica K. Hodgins. Dynamic Simulation of Splashing Fluids[A]. In: Proceedings of the Computer Animation, 1995.198~208.
- [19] Jim X. Chen, Niels da Vitoria Lobo, Charles E. Hughes, J. Michael Moshell. Real-Time Fluid Simulation in a Dynamic Virtual Environment[J]. IEEE Computer Graphics and Applications, 1997, 17(3): 52~61.
- [20] Nick Foster, Dimitri Metaxas. Realistic Animation of Liquids[J]. Graphical Models and Image Processing, 1996, 58(5): 471~483.
- [21] William T. Reeves. Particle Systems - A Technique for Modeling a Class of Fuzzy Objects[J]. Computer Graphics, 1983, 17(3): 359~376.
- [22] Gavin Miller, Andrew Pearce. Globular Dynamics: A Connected Particle System for Animating Viscous Fluids[J]. Computers and Graphics, 1989, 13(3): 305~309.
- [23] Tsunemi Takahashi, Hiroko Fujii, Atsushi Kunitatsu, Kazuhiro Hiwada, Takahiro Saito, Ken Tanaka, Heihachi Ueki. Realistic Animation of Fluid with Splash and Foam[J], Computer Graphics Forum, 2003, 22(3): 391~400.
- [24] Jos Stam, Eugene Fiume. Depicting fire and other gaseous phenomena using diffusion processes[A]. In: Proceedings of SIGGRAPH, Los Angeles, 1995.129~136.
- [25] Mathieu Desbrun, Marie-Paule Gascuel. Smoothed particles: A new approach for animating highly deformable bodies[A]. In: Proceedings of the 7th Eurographics Workshop on Animation and Simulation, Poitiers, 1996.61~76.
- [26] Matthias Müller, David Charypar, Markus Gross. Particle-based fluid simulation for interactive applications[A]. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, 2003.154~159.
- [27] Matthias Müller, Simon Schirm, Matthias Teschner, Bruno Heidelberger, Markus Gross. Interaction of Fluids with Deformable Solids[J]. Journal of Computer Animation and Virtual World (CASA2004), 15(3-4): 159~171.
- [28] Simon Premože, Tolga Tasdizen, James Bigler, Aaron Lefohn, Ross T. Whitaker. Particle-Based Simulation of Fluids[J]. Computer Graphics Forum, 2003, 22(3): 401~410.
- [29] Wei Li, Xiaoming Wei, Arie Kaufman. Implementing Lattice Boltzmann Computation on Graphics Hardware[J]. The Visual Computer, 2003, 19: 444~456.
- [30] Xiaoming Wei, Ye Zhao, Zhe Fan, Wei Li, Suzanne Yoakum-Stover, Arie Kaufman. Blowing in the Wind. In: Proceedings of the 2003 ACM Siggraph/Eurographics Symposium on Computer Animation, San Diego, 2003.75~85.
- [31] Xiaoming Wei, Wei Li, Klaus Mueller, Arie Kaufman. The Lattice-Boltzmann Method for Gaseous Phenomena[J]. IEEE Transactions on Visualization and Computer Graphics, 2004, 10(2): 164~176.
- [32] Wei Li, Zhe Fan, Xiaoming Wei, and Arie Kaufman, GPU-Based Flow Simulation with Complex Boundaries[R].

Technical Report 031105, Computer Science Department, SUNY at Stony Brook, 2003.

- [33] Frédéric Pighin, Jonathan M. Cohen, Maurya Shah. Modeling and Editing Flows Using Advected Radial Basis Functions[A]. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, Grenoble, 2004.223~232.
- [34] Mark J. Harris, William V. Baxter III, Thorsten Scheuermann, Anselmo Lastra. Simulation of Cloud Dynamics on Graphics Hardware[A]. In: Proceedings of Graphics Hardware, San Diego,. 2003.92~101.
- [35] Mark J. Harris. Real-Time Cloud Simulation and Rendering[D]. PhD thesis of The University of North Carolina at Chapel Hill, 2003.
- [36] Philippe Beaudoin, Sébastien Paquet, Pierre Poulin. Realistic and Controllable Fire Simulation[A]. In: Proceedings of Graphics Interface, 2001.159~166.
- [37] Ye Zhao, Xiaoming Wei, Arie Kaufman. Voxels On Fire[A]. In: Proceedings of IEEE Visualization, Seattle, 2003.271~279.
- [38] Bryan E. Feldman, James F. O'Brien, Okan Arikan. Animating Suspended Particle Explosions[J]. ACM Transactions on Graphics, 2003, 22(3): 708~15.
- [39] Viorel Mihalef, Dimitris Metaxas, Mark Sussman. Animation and Control of Breaking Waves[A]. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2004.315~324.
- [40] Lasse Staff Jensen, Robert Goliáš. Deep-Water Animation and Rendering[A], GDCE ,2001.
- [41] Olivier Genevaux, Arash Habibi, Jean-Michel Dischler. Simulating Fluid-Solid Interaction[A]. In: Proceedings of Graphics Interface, 2003.31~38.
- [42] Tolga G. Goktekin, Adam W. Bargteil, James F. O'Brien. A Method for Animating Viscoelastic Fluids[J]. ACM Transactions on Graphics, 2004, 23(3): 463~468.
- [43] Mark Carlson, Peter J. Mucha, Greg Turk. Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid[J]. ACM Transactions on Graphics, 2004, 23(3): 377~384.
- [44] Frank Losasso, Frédéric Gibou, Ron Fedkiw. Simulating Water and Smoke with an Octree Data Structure[J]. ACM Transactions on Graphics, 2004, 23(3): 457~462.
- [45] Nick Rasmussen, Douglas Enright, Duc Quang Nguyen, S. Marino, N. Sumner, Willi Geiger, S. Hoon, Ronald Fedkiw. Directable Photorealistic Liquids[A]. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, Grenoble, 2004.193~202.
- [46] Duc Nguyen, Doug Enright, Ron Fedkiw. Simulation and Animation of Fire and Other Natural Phenomena in the Visual Effects Industry[A]. Western States Section, Combustion Institute, Fall Meeting, UCLA, 2003.
- [47] Kevin Steele, David Cline, Parris K.Egbert, Jonathan Dinerstein. Modeling and rendering viscous liquids[J]. Journal of Computer Animation and Virtual World (CASA2004), 2004, 15(3-4): 183~192.
- [48] Ruxun Liu, Zhifeng Wang, Numerical Simulation methods and Free-Surface Tracking, Hefei: University of Science and Technology of China Press, 2001.(in Chinese)  
(刘儒勋、王志峰, 数值模拟方法和运动界面追踪, 合肥: 中国科学技术大学出版社, 2001.)
- [49] Ryoichi Mizuno, Yoshinori Dobashi, Bing-Yu Chen, Tomoyuki Nishita. Physics Motivated Modeling of Volcanic Clouds as a Two Fluids Model[A]. In: Proceedings of Pacific Graphics, Canmore, 2003.440~444.
- [50] Insung Ihm, Byungkwan Kang, Deukhyun Cha. Animation of Reactive Gaseous Fluids through Chemical Kinetics[A]. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, Grenoble, 2004.203~212
- [51] Demetri Terzopoulos, John Platt, Kurt Fleischer. Heating and melting deformable models (From goop to glob)[A]. In: Proceedings of Graphics Interface, 1989.219~226.
- [52] Daniel Nixon, Richard Lobb. A Fluid-Based Soft-Object Model[J]. IEEE Computer Graphics and Applications archive, 2002, 22(4) 68~75.
- [53] Dan Stora, Pierre-Olivier Agliati, Marie-Paule Cani, Fabrice Neyret, Jean-Dominique Gascuel. Animating Lava Flows[A]. In: Proceedings of Graphics Interface, Kingston, 1999.203~210.

- [54] Xiaoming Wei, Wei Li, Arie Kaufman. Melting and Flowing of Viscous Volumes[A]. In: Proceedings of the 16th International Conference on Computer Animation and Social Agents, New Brunswick , 2003.54~59.
- [55] Jakub Wejchert, David Haumann. Animation aerodynamics[J]. Computer Graphics, 1991, 25(4): 19~22.
- [56] Jos Stam. Flows on Surfaces of Arbitrary Topology[J]. ACM Transactions on Graphics, 2003, 22(3): 724~731.
- [57] Lin Shi, Yizhou Yu. Inviscid and incompressible fluid simulation on triangle meshes[J]. Journal of Computer Animation and Virtual World (CASA2004), 2004, 15(3-4): 173~181.
- [58] Patrick Witting. Computational fluid dynamics in a traditional animation environment[A]. In: Proceedings of SIGGRAPH, 1999.129~136.
- [59] Adrien Treuille, Antoine McNamara, Zoran Popovic, Jos Stam. Keyframe Control of Smoke Simulations[J]. ACM Transactions on Graphics, 2003, 22(3): 716~723.
- [60] Antoine McNamara, Adrien Treuille, Zoran Popovic, Jos Stam. Fluid Control Using the Adjoint Method[J]. ACM Transactions on Graphics, 2004, 23(3): 449~456.
- [61] Jos Stam. Simulation and Control of Physical Phenomena in Computer Graphics (Invited Talk)[A]. In: Proceedings of Pacific Graphics, Seoul, 2004.171~173.
- [62] Raanan Fattal, Dani Lischinski. Target-Driven Smoke Animation[J]. ACM Transactions on Graphics, 2004, 23(3): 441~448.
- [63] Jeong-mo Hong, Chang-hun Kim. Controlling fluid animation with geometric potential[J]. Journal of Computer Animation and Virtual World (CASA2004), 2004, 15(3-4): 147~157.
- [64] Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischery, David H. Salesin. Computer-Generated Watercolor[A]. In: Proceedings of SIGGRAPH, 1997.421~430.
- [65] William Baxter, Yuanxin Liu, Ming C. Lin. A Viscous Paint Model for Interactive Applications[J]. Journal of Computer Animation and Virtual Worlds (CASA2004), 2004, 15(3-4): 433~441.
- [66] Andrew Selle, Alex Mohr, Stephen Chenney. Cartoon rendering of smoke animations[A]. In: Proceedings of NPAR, 2004.57~60.
- [67] Jos Stam, Eugene Fiume. Turbulent Wind Fields for Gaseous Phenomena[A]. In: Proceedings of SIGGRAPH, Anaheim, 1993.369~376.
- [68] Satoru Yoshida, Tomoyuki Nishita. Modeling of Smoke Flow Taking Obstacles into Account[A]. In: Proceedings of Pacific Graphics, 2000.135~145.
- [69] Andrew Selle, Nick Rasmussen, Ronald Fedkiw. A Vortex Particle Method for Smoke, Water and Explosions[A]. To appear in SIGGRAPH2005.
- [70] Jeong-Mo Hong, Chang-Hun Kim. Discontinuous Fluids[A]. To appear in SIGGRAPH2005.
- [71] Huamin Wang, Peter J. Mucha, Greg Turk. Water Drops on Surfaces[A]. To appear in SIGGRAPH2005.
- [72] Nick Rasmussen, Duc Quang Nguyen, Willi Geiger, Ronald Fedkiw. Smoke Simulation For Large Scale Phenomena[J]. ACM Transactions on Graphics, 2003, 22(3): 703~707.
- [73] Maurya Shah, Jonathan M. Cohen, Sanjit Patel, Penne Lee, Frédéric Pighin. Extended Galilean Invariance for Adaptive Fluid Simulation[A]. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Grenoble, 2004.213~221.
- [74] Mark J. Harris, Greg Coombe, Thorsten Scheuermann, Anselmo Lastra. Physically-Based Visual Simulation on Graphics Hardware[A]. In: Proceedings of Graphics Hardware, Saarbrücken, 2002.109~118.
- [75] Mark J. Harris. Flo: A real time fluid float simulator written in Cg[OL], 2003. <http://www.markmark.net/gdc2003/>
- [76] Jens Krüger, Rüdiger Westermann. Linear Algebra Operators for GPU Implementation of Numerical Algorithms[J]. ACM Transactions on Graphics, 2003, 22(3): 908~916.
- [77] Chris Batty, Mark Wiebe, Ben Houston. High Performance Production-Quality Fluid Simulation via NVIDIA's QuadroFX[OL]. [http://film.nvidia.com/docs/CP/4449/frantic\\_GPUAccelerationofFluids.pdf](http://film.nvidia.com/docs/CP/4449/frantic_GPUAccelerationofFluids.pdf). 2003.

- [78] Enhua Wu, Youquan Liu, Xuehui Liu. An Improved Study of Real-Time Fluid Simulation on GPU[J]. Journal of Computer Animation and Virtual World(CASA2004), 2004, 15(3-4): 139~146.
- [79] Lutz Latta, Building a Million Particle System[A]. In: Proceedings of the Game Developer Conference, March 2004.
- [80] Peter Kipfer, Mark Segal, Rudiger Westermann. UberFlow: A GPU-Based Particle Engine[A]. Proceedings of Graphics Hardware 2004, <http://www.cg.in.tum.de/Research/Publications/UberFLOW>, August 2004.
- [81] Nick Foster, Dimitris Metaxas. Controlling fluid animation[A]. In: Proceedings of Computer Graphics International, 1997.178~188.
- [82] Stefanie Olsen. Hollywood whistles a high-tech 'toon[OL]. [http://news.zdnet.com/2100-1040\\_22-5431677-3.html](http://news.zdnet.com/2100-1040_22-5431677-3.html), October 29, 2004.

#### 作者联系方式:

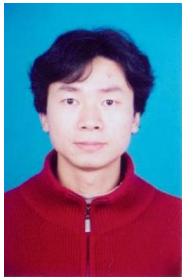
柳有权, 刘学慧, 朱红斌, 吴恩华  
中科院软件研究所计算机科学实验室  
北京 8718 信箱  
100080  
电话: 010-62522028/62562796  
Email: [lyq@ios.ac.cn](mailto:lyq@ios.ac.cn)  
Website: <http://lcs.ios.ac.cn/~lyq/>

#### 作者简介:



**柳有权**, 男, 1976 年生, 湖北人, 博士研究生, 研究方向为计算机动画, 计算机图形学;

**刘学慧**, 女, 1968 年生, 湖南人, 博士, 副研究员, 研究方向为计算机图形学;



**朱红斌**, 男, 1978 年生, 湖北人, 博士研究生, 研究方向为计算机动画, 计算机图形学;



**吴恩华**, 男, 1947 年生, 江苏人, 博士, 研究员, 博士生导师, 研究方向为虚拟现实, 计算机图形学.