

一种基于半拉格朗日的液体实时仿真方法^{*}

邹玲, 齐越, 赵沁平

(虚拟现实技术与系统国家重点实验室(北京航空航天大学), 北京 100191)

通讯作者: 邹玲, E-mail: zouling@vrlab.buaa.edu.cn, http://www.vrlab.buaa.edu.cn

摘要: 近些年,在计算机图形学与虚拟现实技术领域中,自然现象的模拟得到了广泛的关注和研究.如何快速且逼真地模拟自然现象,是此类研究的目的.以液体表面作为研究对象,总结了关于液体模拟近年来的部分研究成果;针对三维液体的复杂流体状态,提出了一种基于半拉格朗日的液体实时仿真方法,并对仿真结果进行了表面构建.该方法首先将 Navier-Stokes 方程离散化,并通过求解构造的 Poisson 方程得到每一时间步长的数值解,进而精确驱动粒子运动以构建真实液体表面;之后,利用液体表面追踪及 Marching Cubes 表面重建,生成了真实的液体表面模型.实验结果表明,该仿真方法不但在运算过程中遵循经典的流体力学方程,从而保证了结果的真实性,并且运算速度快且能取得较好的视觉效果.在计算机游戏、电影制作以及医学等领域的仿真,均有广泛的应用前景.

关键词: 液体仿真;纳维-斯托克斯方程;半拉格朗日法;自然现象模拟;实时

中图法分类号: TP391 **文献标识码:** A

中文引用格式: 邹玲,齐越,赵沁平.一种基于半拉格朗日的液体实时仿真方法.软件学报,2013,24(10):2419-2431. <http://www.jos.org.cn/1000-9825/4436.htm>

英文引用格式: Zou L, Qi Y, Zhao QP. Real-Time approach for dynamic liquid simulation using semi-Lagrangian. Ruan Jian Xue Bao/Journal of Software, 2013,24(10):2419-2431 (in Chinese). <http://www.jos.org.cn/1000-9825/4436.htm>

Real-Time Approach for Dynamic Liquid Simulation Using Semi-Lagrangian

ZOU Ling, QI Yue, ZHAO Qin-Ping

(State Key Laboratory of Virtual Reality Technology and Systems (BeiHang University), Beijing 100191, China)

Corresponding author: ZOU Ling, E-mail: zouling@vrlab.buaa.edu.cn, http://www.vrlab.buaa.edu.cn

Abstract: In recent years, natural phenomena simulation attracts a spurt of research attention and interest in computer graphics and virtual reality domain. How to obtain realistic natural phenomena by simulation in an efficient way is the main purpose of this research field. This paper summarizes the main research achievements on the topic of liquid simulation. A real-time liquid simulation method based on semi-Lagrangian is proposed for liquid simulation, followed by a surface reconstruction method using simulated results. In this approach, Navier-Stokes equations are first discretized in both spatial and temporal dimensions. Second, by solving the constructed Poisson equation, numerical solution for each time step is obtained. Third, particle movements are accurately driven by the solved velocity field in order to simulate the dynamics of water. By applying surface tracking and Marching Cubes algorithm, water surface is finally extracted. Experimental results show that the simulation speed of this method is high enough for real time use. Satisfactory visual effect is also obtained, facilitating the usage of this application in computer games, movie making and virtual simulation in medical area.

Key words: liquid simulation; Navier-Stokes equations; semi-Lagrangian; natural phenomena simulation, real time

真实世界中的自然现象丰富多彩,并且存在着多种复杂且丰富的细节信息,例如风、雨、烟雾、火焰、气泡等.对自然现象的模拟仿真,主要可应用于计算机游戏与电影制作、虚拟手术的训练和仿真、机械制造与建筑设计可视化以及科学研究可视化等领域.因此,自然现象的仿真成为虚拟现实技术研究中的一个重要方向.高

* 基金项目: 国家自然科学基金(61073078, 61272348); 国家教育部博士点基金(201111102110018)

收稿时间: 2012-11-07; 修改时间: 2013-03-18; 定稿时间: 2013-05-24

质量的自然现象模拟须满足以下 4 点特性:高效性、高保真度、普适性、易耦合性.如何在数字世界中真实、有效地快速重现此类自然现象,成为计算机图形学领域的热点研究问题.

在多样的自然现象中,流体作为重要且最常见的组成部分,成为模拟仿真的主要对象之一.流体主要分为可压缩流体(如气体)和不可压缩流体(如水、果汁)两部分.在计算机图形学领域,基于物理的流体仿真方法主要分为两大类:基于网格的欧拉方法和基于粒子系统的拉格朗日方法.本文分析并总结了近年来的研究成果,以液体为研究对象,提出并实现了一种基于半拉格朗日的动态液体表面建模方法.在算法中综合了两种主流方法的优点,最后给出了实验结果与分析.

1 流体仿真研究现状

基于物理原理的流体模拟所涉及的现象主要包括:烟雾^[1,2]、液体表面^[3,4]、气泡^[5-7]、火焰^[8,9]、波浪^[10]、云^[11]、粘稠液体^[12,13]等.这些现象都是计算流体力学^[14]领域所关注的方面.基于物理原理的模拟方法,就是在数字世界中依照某些特定的物理法则模拟自然现象.大多数基于物理原理的模拟方法计算量较大,特别是模拟复杂的物体碰撞以及流体运动的过程,如流体的平流、扩散、涡流以及表面张力等细节内容也大大增加了真实模拟流体动态规律的难度.因此,自然现象模拟需设计各种加速方法,以解决计算量大的问题^[15,16].

由于实时性的限制,传统的流体仿真往往采用的是运算和绘制分开的方法,即先进行模型的运算并将计算结果记录下来,当完成全部的仿真计算之后再对结果进行可视化显示.虽然当前一些简化的仿真方法可以达到实时显示的效果,但是这些方法只适用于某些应用,如验证设计概念的可行性、电脑游戏等,对于精度要求较高的应用,如医学仿真、空气动力学仿真等则不适用.当前,计算机图形学对流体仿真的研究需要对物理现象本身有较为精确的求解,并利用最新的计算机技术及高速设备快速得到真实的视觉效果.

流体的仿真在流体仿真中占有举足轻重的地位,大量的研究工作关注于各种不同形态的液体流动状态.较早的流体仿真方法主要来源于 Foster 和 Metaxas^[17]、Stam^[18]、Foster 和 Fedkiw^[19]提出的方法.目前,主流的液体模拟方法大体上可以分为两类:基于网格的模拟方法和基于粒子的模拟方法.基于网格的模拟方法将自然空间用整齐的网格划分,特定区域中的液体流动变化状态则记录在每一个网格单元上用于后续绘制,空间离散网格的方法主要包括有限差分法和有限体积法等.此类方法一般可以得到高精度的结果,运用广泛,但对于飞溅等流体效果建模较为困难.在多液体间交互^[20]、液体与不可压缩物体间交互^[21]、液体与可变形物体间交互^[22]等研究工作中,均运用了基于网格的方法.基于粒子的模拟方法以平滑粒子流体力学模拟方法(smoothed particle hydrodynamics,简称 SPH)为代表,该类方法将流体分割为许多离散的元素,即粒子.任何一个粒子的物理属性可以通过综合在该粒子某一空间距离内的其他相关粒子的属性而得到.该方法表达清晰,一般运算速度较快,易于控制且能够保证质量守恒,但对于平滑的运动流体表面建模比较困难,计算量随着粒子数的增多而急剧增大.由 SPH 方法演变出的方法主要有:预测纠正不可压缩 SPH 方法^[23]、基于状态方程的 SPH 方法^[24]、不可压缩 SPH 方法^[25]等.由此可以看出,以上方法各具优、缺点,因此需将基于网格的方法与基于粒子的方法相结合使用,以同时满足精度和速度的要求.本文提出的动态液体表面实时建模将采用一种基于半拉格朗日(semi-Lagrangian)的方法综合使用这两种方法,以充分利用欧拉法具有较高精度的优点以及拉格朗日法在追踪及控制流体运动中的灵活性的特点.后续章节将首先介绍本文方法中基于网格的纳维-斯托克斯方程的数值解法,进而介绍基于粒子的动态液体实时跟踪及表面建模方法.

2 Navier-Stokes 方程及数值解

液体仿真方法大多是以流体力学原理为理论基础,运算方法围绕着如何求解和优化流体控制方程,进而得到方程的数值解作为每一时刻的仿真结果.纳维-斯托克斯(Navier-Stokes)方程由 Navier 和 Stokes 在不同时期提出并加以完善,精确地形式化描述了流体的动态规律.除了用于描述能量守恒的能量方程,纳维-斯托克斯方程通过动量方程和连续性方程分别描述动量守恒和质量守恒的规律.本节将介绍如何基于纳维-斯托克斯方程的动量方程和连续性方程完成速度场的求解.

2.1 流体控制方程

纳维-斯托克斯方程作为最著名的流体控制方程,用于模拟自然界中的各类流体.它也是目前描述流体最精确、最全面的数学模型.本文将利用动量方程和连续性方程研究流体的运动规律,并应用于液体表面的实时建模.常用的不可压缩的纳维-斯托克斯方程形式如下:

$$\frac{\partial \bar{u}}{\partial t} + \bar{u} \cdot \nabla \bar{u} + \frac{1}{\rho} \nabla p = \bar{g} + \nu \nabla \cdot \nabla \bar{u} \quad (\text{动量方程}) \quad (1)$$

$$\nabla \cdot \bar{u} = 0 \quad (\text{连续性方程}) \quad (2)$$

其中, \bar{u} 表示流体速度, ρ 为流体密度(纯水的密度大约是 1000kg/m^3 , 空气的密度大约是 1.3kg/m^3), \bar{g} 是重力加速度, ν 是运动粘度系数(衡量流体的粘度大小), P 为压强, 表示每单元区域的流体对外界所施加的力. 符号 ∇ 及 $\nabla \cdot$ 分别表示梯度 (gradient) 和散度 (divergence) 算子, $\nabla \cdot \nabla$ 称为拉普拉斯 (Laplacian) 算子. 公式 (1) 为动量方程, 公式 (2) 为质量方程或不可压缩条件. 本文将研究对象限制在三维空间, 即约定 $\bar{x} = (x, y, z)^T$, $\bar{u} = (u, v, w)^T$, $\bar{g} = (g_x, g_y, g_z)^T$, 从而将方程 (1) 和方程 (2) 进一步细化为

$$\left. \begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial x} &= \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) + g_x \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial y} &= \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) + g_y \end{aligned} \right\} \quad (3)$$

$$\left. \begin{aligned} \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial z} &= \nu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) + g_z \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} &= 0 \end{aligned} \right\} \quad (4)$$

利用纳维-斯托克斯方程进行流体运动的仿真, 关键是通过离散化得到微分方程的数值解以追踪流体在各个时刻的位置、速度等运动状态. 离散化的方法主要包括以下几种: 有限差分法、有限元法、有限体积法. 其中,

- 有限差分法是数值方法中最经典的方法, 将求解域划分为差分网格, 用有限个网格节点代替连续的求解域, 然后将控制方程的导数用差商代替, 推导出含有离散点上有限个未知数的差分方程组;
- 有限元法是将一个求解域任意分成有限个互不重叠的微小单元, 在每个单元里选择一些合适的节点作为求解函数的插值点, 将控制方程的变量改写成由各变量或其导数的节点值与所选的插值函数组成的线性表达式;
- 有限体积法是将计算区域划分为网格, 并使每个网格点周围有一个互不重复的控制体积, 将控制方程对每一个控制体积分, 从而得到一组离散方程.

以上方法各有优势, 在计算机图形学研究领域, 如何快速似真地模拟液体运动过程是主要目标, 为此, 所选取的差分方法需满足表达简洁、计算速度快、绘制结果逼真的条件. 如 Bridson^[26], Stam^[18] 等人提出的方法均选取有限差分法作为离散方法, 因其将微分问题变为代数问题的近似数值解法形式简单、可快速计算且适用于网格结构, 本文选取有限差分法以满足实时逼真绘制的要求. 虽然离散化后得到的只是研究域中有限多个点的数值解, 但是离散化将差分方程的求解转化为普通的代数方程的求解, 大大降低了计算的复杂性; 并通过提高离散化的精度即可以有效提高数值解的精度, 从而得到逼真的仿真结果.

2.2 Navier-Stokes 方程的离散化

2.2.1 空间离散化

在基于网格的模拟方法中, 空间被划分为标准的有规则形状网格. 许多不同种类的属性需要被存储在网格中的各个单元上, 例如速度、压力、浓度等. Harlow 和 Welch^[27] 提出了一种 MAC 网格技术, MAC 网格是一种错列的网格, 它将不同的属性存储在网格中不同的位置上, 而不是集中在同一节点. 以三维网格为例, 图 1 所示为 MAC 网格的一个单元. 其中, 网格单元 (i, j, k) 的属性 $P_{i,j,k}$ 存储在网格的中心, 而 $u_{i,j,k}, v_{i,j,k}, w_{i,j,k}$ 分别在网格单元间的

交界面上采样,表示网格单元 (i,j,k) 与单元 $(i+1,j,k),(i,j+1,k),(i,j,k+1)$ 之间速度的法向分量.3个速度分量均沿笛卡尔坐标系的轴向,其中, $i=1,\dots,i_{\max},j=1,\dots,j_{\max},k=1,\dots,k_{\max}$.我们可以对速度的空间导数作如下中心差分表示:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j,k} := \frac{u_{i,j,k} - u_{i-1,j,k}}{\delta x}, \left(\frac{\partial v}{\partial y}\right)_{i,j,k} := \frac{v_{i,j,k} - v_{i-1,j,k}}{\delta y}, \left(\frac{\partial w}{\partial z}\right)_{i,j,k} := \frac{w_{i,j,k} - w_{i-1,j,k}}{\delta z} \quad (5)$$

同理,采用上述中心差分方法可以对公式(3)的其他空间差分项进行类似的离散化.中心差分方法的离散化误差具有阶数 $O(\delta x)^2$,采用 MAC 网格后相当于量化步长缩小了一半,从而将量化精度提高了4倍.

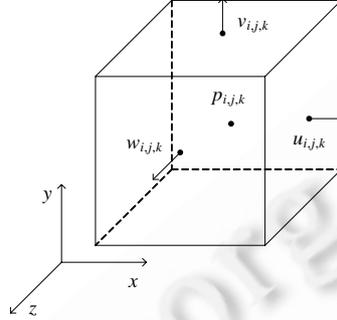


Fig.1 A unit of 3D MAC grid

图1 MAC 网格单元

2.2.2 时间离散化

在流体动力学方程中对时间进行离散化,就是对公式(3)中的 $\partial u/\partial t, \partial v/\partial t$ 及 $\partial w/\partial t$ 这3个时间差分项进行离散化.在仿真过程中,假设每个时间步长为 δt ,则在每个 $n\delta t$ 时刻,我们可以得到:

$$\left(\frac{\partial u}{\partial t}\right)^n := \frac{u^n - u^{n-1}}{\delta t}, \left(\frac{\partial v}{\partial t}\right)^n := \frac{v^n - v^{n-1}}{\delta t}, \left(\frac{\partial w}{\partial t}\right)^n := \frac{w^n - w^{n-1}}{\delta t} \quad (6)$$

其中, $n=1,\dots,N.N\delta t$ 为仿真结束时间.如果我们令:

$$\left. \begin{aligned} F &:= u^{n-1} + \delta t \left(\nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) - \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) + g_x \right) \\ G &:= v^{n-1} + \delta t \left(\nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) - \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) + g_y \right) \\ H &:= w^{n-1} + \delta t \left(\nu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) - \left(u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) + g_z \right) \end{aligned} \right\} \quad (7)$$

联合公式(3)、公式(6)、公式(7)并加以简化,可以得到:

$$u^n = F^{n-1} - \delta t \frac{\partial p^n}{\rho \partial x}, v^n = G^{n-1} - \delta t \frac{\partial p^n}{\rho \partial y}, w^n = H^{n-1} - \delta t \frac{\partial p^n}{\rho \partial z} \quad (8)$$

2.3 求解Poisson方程

完成 Navier-Stokes 公式(3)的空间和时间离散化后,为了求解每一步的压强 p ,我们将公式(8)代入公式(4):

$$\frac{\partial u^n}{\partial x} + \frac{\partial v^n}{\partial y} + \frac{\partial w^n}{\partial z} = \frac{\partial F^{n-1}}{\partial x} - \delta t \frac{\partial^2 p^n}{\rho \partial x^2} + \frac{\partial G^{n-1}}{\partial y} - \delta t \frac{\partial^2 p^n}{\rho \partial y^2} + \frac{\partial H^{n-1}}{\partial z} - \delta t \frac{\partial^2 p^n}{\rho \partial z^2} = 0 \quad (9)$$

即

$$\frac{\partial^2 p^n}{\partial x^2} + \frac{\partial^2 p^n}{\partial y^2} + \frac{\partial^2 p^n}{\partial z^2} = \frac{1}{\delta t} \left(\frac{\rho \partial F^{n-1}}{\partial x} + \frac{\rho \partial G^{n-1}}{\partial y} + \frac{\rho \partial H^{n-1}}{\partial z} \right) \quad (10)$$

公式(10)为压强 p 的三维 Poisson 方程,方程左侧为关于 p 的拉普拉斯算子,即 $\nabla \cdot \nabla p$,对于该算子中二阶偏导

的离散化,可以在公式(5)所示的中心差分上再作一阶差分.事实上,Poisson 方程也可以表示为矩阵方程的形式,即 $Ax=b$,只不过系数矩阵 A 隐含在拉普拉斯算子 $\nabla \cdot \nabla$ 中.目前,已有很多成熟方法求解此类线性方程组,然而按照上述空间离散化的方法,在每个时间步长都需要求解 $i_{\max} \times j_{\max} \times k_{\max}$ 个未知变量 $p_{i,j,k}$.当离散化精度较高,即 $i_{\max}, j_{\max}, k_{\max}$ 均较大时,经典的高斯消元法(Gaussian elimination)在存储消耗及时间消耗上都会比较高.在实际应用中,一些迭代的方法往往更加可行,例如雅克比迭代法(Jacobi iteration)^[28]、Gauss-Seidel 方法、SOR(successive overrelaxation)方法等等.其他一些方法,如共轭梯度法(conjugate gradient)^[29]、Multigrid^[30]方法、MICCG(modified incomplete cholesky conjugate gradient)^[26]等虽更高效且具有更好的收敛特性,但相对于传统迭代法较为复杂,因此在很多实时方法中,传统迭代法仍被广泛采用.

获得公式(10)关于压强的求解结果后,应用压强关于 3 个方向的空间导数公式(11),并代入公式(8)即可得到整个液体内部在 $n\delta t$ 时刻的速度场.

$$\left(\frac{\partial p^n}{\partial x} \right)_{i,j,k} := \frac{p_{i,j,k}^n - p_{i-1,j,k}^n}{\delta x}, \left(\frac{\partial p^n}{\partial y} \right)_{i,j,k} := \frac{p_{i,j,k}^n - p_{i,j-1,k}^n}{\delta y}, \left(\frac{\partial p^n}{\partial z} \right)_{i,j,k} := \frac{p_{i,j,k}^n - p_{i,j,k-1}^n}{\delta z} \quad (11)$$

3 基于粒子的实时液体仿真

上述章节首先通过将 Navier-Stokes 方程离散化转换为代数方程,进而通过求解构造的 Poisson 方程得到每一时间步长的数值解.本文提出的一种半拉格朗日仿真方法将利用求解的结果精确地驱动液体内部的粒子运动,以达到真实反映液体表面形状的目的.

3.1 基于粒子的动态传输过程

动态液体的实时仿真即由速度传送液体本身和在液体中的其他变量的过程.在采用了上述基于网格的空间和时间离散化的流体控制方程之后,对该仿真过程的应用是为了获取网格单元中的属性值在 δt 后的更新值,如一段时间步长后液体表面的位置、某些粒子的速度等等.基于粒子的流体仿真一般是将流体的速度属性应用于各个粒子,也可计算温度等其他属性.在计算机图形学研究中,基于粒子的仿真由于其在构建形变物体上的优势,在最近的研究中受到广泛青睐.例如,很多研究人员利用粒子方法来构建软性物体^[31,32]、进行交互式流体仿真^[24]以及应用 SPH 方法进行流体表面构建^[23-25]等等.由于基于粒子的仿真方法被称为拉格朗日方法,结合前面介绍的基于网格的流体动力学方程求解,本文提出了一种基于半拉格朗日的方法用于液体表面实时绘制.这种方法既充分利用欧拉式的精确求解结果,又应用了粒子的理念以完成动态液体任意形状的灵活构建,充分发挥了这两种不同方法的优势.

在基于粒子的流体仿真中,粒子作为构成整个流体的基本单元,每一个粒子都是一个单独的实体.为此,我们需要定义如下的属性结构来描述每一个粒子.为简化起见,可以假设粒子的质量、密度等属性恒定,只讨论在速度场的驱动下粒子位置的变化,该变化也直接决定了所要仿真的液体表面的形状细节.

- 粒子的位置,即在空间 3 个维度上的坐标 $\bar{x} = (x, y, z)^T$;
- 粒子的速度,即三维速度向量 $\bar{u} = (u, v, w)^T$;
- 单个粒子的质量,记为 m ;
- 粒子的密度,记为 ρ ;
- 压强,记为 p ;
- 颜色,即 (r, g, b) 值;
- 外力,即 $\bar{g} = (g_x, g_y, g_z)^T$.

在获得整个空间域的速度场之后,需要计算分布于单个网格中粒子的速度,以计算下一时刻每个粒子的速度.通常,单个粒子的速度是通过粒子当前所处的位置对其临近 MAC 网格的速度进行线性插值得到,这样得到的结果往往比直接采用粒子所处网格的速度更为精确.图 2 表示了单个粒子在二维平面中,即在 z 轴方向的索引 k 取固定值的情况下,在速度场驱动下的运动过程(如图中的空心粒子).图 2 以 MAC 网格(均匀分布 3×3 个粒子)

的纵向速度 v 为例,描述了位置为 (x,y) 的空心粒子在当前速度场的运动属性.

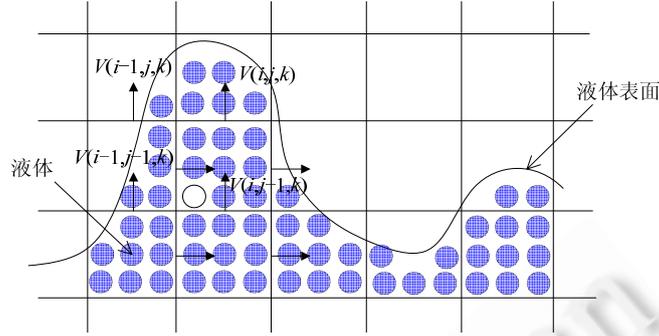


Fig.2 Particle movement in velocity field

图 2 速度场中粒子的运动过程

从图 2 可以看出,目标粒子的纵向速度可以从其周围 4 个网格,即 $(i-1,j-1,k)$, $(i,j-1,k)$, $(i-1,j,k)$ 及 (i,j,k) 的纵向速度插值得到.在二维情况下,如图 2 所示目标粒子的纵向速度 $v(x,y)$ 可以由这 4 个速度通过双线性插值得到.同理,在三维条件下,应用类似的推导可以通过对 8 个单元格的速度进行三线性插值得到任何粒子的速度值.以 x 方向速度分量为例,给出位于 (x,y,z) 粒子的三线性速度插值的结果 $u(x,y,z)$:

令

$$\begin{aligned} x_1 &= (i-1)\delta x, y_1 = (j-1-1/2)\delta y, z_1 = (k-1-1/2)\delta z, \\ x_2 &= i\delta x, y_2 = (j-1/2)\delta y, z_2 = (k-1/2)\delta y, \end{aligned}$$

则有,

$$\begin{aligned} u_1 &= [(x_2-x)(y_2-y)u_{i-1,j-1,k-1} + (x-x_1)(y_2-y)u_{i,j-1,k-1} + (x_2-x)(y-y_1)u_{i-1,j,k-1} + (x-x_1)(y-y_1)u_{i,j,k-1}]/(\delta x\delta y), \\ u_2 &= [(x_2-x)(y_2-y)u_{i-1,j-1,k} + (x-x_1)(y_2-y)u_{i,j-1,k} + (x_2-x)(y-y_1)u_{i-1,j,k} + (x-x_1)(y-y_1)u_{i,j,k}]/(\delta x\delta y). \end{aligned}$$

从而得到插值后的速度:

$$u(x,y,z) = [(z_2-z)u_1 + (z-z_1)u_2]/\delta z \quad (12)$$

其中, $i = \text{int}(x/\delta x) + 1$, $j = \text{int}(y/\delta y + 0.5) + 1$, $k = \text{int}(z/\delta z + 0.5) + 1$, $\text{int}(\cdot)$ 为取整函数.同理,可计算得到粒子在 y 方向和 z 方向的速度分量.完成单个粒子速度的精确插值之后,可以直接应用前向欧拉(forward Euler)方法更新每一个粒子下一时刻的位置,如公式(13)所示:

$$x^{n+1} = x^n + \delta t u^n, y^{n+1} = y^n + \delta t v^n, z^{n+1} = z^n + \delta t w^n \quad (13)$$

3.2 边界条件

在粒子的平流过程中,不可避免地会与外围或液体内部的固体发生碰撞,例如海浪拍打海岸、水缸里的水波碰到缸壁返回等等.由于流体经常与固体表面发生交互运动,这些固体表面也是最常见的边界条件,对这种边界条件的解析往往通过约定位于边界的流体的速度来实现,常用的方法有不可滑动条件(no-slip condition)及自由滑动条件(free-slip condition)^[26].不可滑动条件针对的是具有粘性的液体在固体表面速度的限定条件:一方面,由于代表液体成分的粒子不能流入或流出固体表面,从而液体粒子在固体表面的法向速度分量为 0;另一方面,液体的粘性直接对固体表面的切向速度也产生了影响.为此,在不可滑动边界条件下,常常约定在固体表面的速度等于固体的速度,即 $\vec{u} = \vec{u}_{solid}$.当固体静止时,有 $\vec{u} = 0$.自由滑动条件只对于液体在固体表面的法向速度分量进行约束,即满足 $\vec{u} \cdot \hat{n} = \vec{u}_{solid} \cdot \hat{n}$.同样,当固体静止时,有 $\vec{u} \cdot \hat{n} = 0$.本文在实验中采用了自由滑动条件作为固体表面的边界条件.

在建模液体表面的过程中,自由表面(free surface)是另外一种需要考虑的边界,即除了与固体边界接触之外的液体表面,如海面的波浪、雨滴溅起的水花等.不同于固体边界,自由表面是由于液体的不断流动所形成的形状不断变化的边界.本文将所研究的自由表面限定为水与空气之间的接触面,因为这种表面更为常见,对此类现

象仿真更具有现实意义.自由表面的形状,事实上就是待建模的主要液体表面.根据文献[24]所述,空气比水要轻 700 倍,因此,空气对水的作用并不显著.在仿真过程中为反映出压强差,可以方便地将自由表面的压强限定为 $p=0$.为避免压强值从 0(空气中)直接突变到非零值(水中),直接与水交界的空气网格中,压强值可以通过线性外推得到.假设第 i 个网格为液体网格,压强为 p_i ;第 $i+1$ 个网格为紧邻的空气网格,压强为 p_{i+1} ;二者间自由表面的位置为 $x_F=(1-\theta)x_i+\theta x_{i+1}$, θ 为自由表面介于两个网格之间的线性比例,则为了保证 $p(x_F)=0$,可以线性外推得到第 $i+1$ 个网格的压强 $p_{i+1}=-\frac{p_i}{\theta}$.

3.3 稳定性分析

选取时间步长的主要目的,是为了快速完成仿真过程而尽量不丢失数值精度.因此,在不损失效果的前提下,需要选取尽可能长的时间步长.而在仿真过程中,为保证数值求解的稳定性,尽可能避免计算结果的震荡,需要在算法设计过程中考虑离散化步长的稳定性条件.这个稳定性条件可以通过分析粒子在单个时间步长 δt 运动的距离与空间量化步长 $\delta x, \delta y$ 及 δz 之间的关系得到,其中较为著名的是 CFL(Courant-Friedrichs-Lewy)条件.具体选取过程中 CFL 条件的方法是:选取足够小的时间量化步长 δt 值,以保证粒子在速度场中移动的距离小于单个空间量化步长.从上述第 3.1 节粒子速度的线性插值方法不难看出,单个粒子的速度必不超过其周边网格速度的最大值.因此,CFL 条件在上述三维量化过程中可以表示为

$$|u_{\max}| \delta t < \delta x, |v_{\max}| \delta t < \delta y, |w_{\max}| \delta t < \delta z \quad (14)$$

从这个条件可以直接得到时间步长的取值为

$$\delta t := \min \left(\frac{\delta x}{|u_{\max}|}, \frac{\delta y}{|v_{\max}|}, \frac{\delta z}{|w_{\max}|} \right) \quad (15)$$

时间步长的选取越大,则更易损失仿真的精度.但在计算机图形学中,流体仿真的目的是为了得到看似真实的效果,下降一定量的精度,并不一定会导致模拟可视效果的明显不真实性.因此,在实际情况中,研究人员往往为了提高运算速度而损失一定量的精度,从而采用改进 CFL 条件的方法以获得更大的时间步长.例如,Foster 等人^[19]选取的时间步长是 CFL 条件选取结果的 5 倍,并获得了很好的模拟效果.因此,时间步长的选取可针对实际的研究需求来确定.在本文实验中,则直接采用了如公式(16)所示的条件,即增加一个缩放因子 k_{CFL} 以增加 δt 选取的灵活性.

$$\delta t := k_{\text{CFL}} \min \left(\frac{\delta x}{|u_{\max}|}, \frac{\delta y}{|v_{\max}|}, \frac{\delta z}{|w_{\max}|} \right) \quad (16)$$

需要说明的是,在液体仿真的实际应用过程中,往往会考虑外力(如重力)的作用,在这种情况下,上述的 u_{\max} 并非一成不变.一种更为鲁棒的稳定性条件是:将外力产生的加速度考虑在内,而采用 $\bar{u}_{\max} = \max(|\bar{u}|) + \delta t |\bar{g}|$. 结合公式(16),即可得到 $\bar{u}_{\max} = \max(|\bar{u}|) + \sqrt{k_{\text{CFL}} \delta \bar{x} |\bar{g}|}$ 作为修正后的最大速度值.

4 液体表面跟踪及建模

上述章节详细介绍了利用欧拉法求解流体控制方程,并通过得到的速度场驱动粒子运动的仿真方法.为验证该仿真的整体效果以及对后续应用如真实感绘制的可行性,本节设计实现了基于上述半拉格朗日仿真方法的液体表面跟踪及建模算法,并用于液体表面的生成.该算法主要包括两个部分:标量场计算及表面重建.

在基于粒子的液体仿真中,液体的表面通常被定义为一个由标量场决定的等值面(isosurface),因此,标量场的计算将直接影响液体表面的构建效果.本文为了验证上述基于半拉格朗日的仿真方法,采用了如公式(17)所示的标量场^[24,33]:

$$\phi(\bar{x}) = \sum_j \frac{m_j}{\rho_j} W(\bar{x} - \bar{x}_j, h) \quad (17)$$

其中, W 为各向同性的平滑核.常用的核函数有高斯核、poly6 核、三次样条核等^[24,33,34].本文在实验中选取 poly6 核,并且具有如下形式:

$$W_{poly6}(\vec{r}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & 0 \leq r \leq h \\ 0, & r < 0, r > h \end{cases} \quad (18)$$

另外,公式(17)中液体粒子的密度 ρ_j 可以通过其临近粒子质量进行加权求和的方法近似得到:

$$\rho_j = \sum_i m_i W(\vec{x}_i - \vec{x}_j, h) \quad (19)$$

至此,标量场函数 $\phi(\vec{x})$ 给定了一个粒子仿真后形成的等值面作为最终的液体表面.其中,在计算相应位置的标量,即给定三维空间位置 \vec{x} 的 $\phi(\vec{x})$ 值时,需要首先确定对计算标量值有贡献的粒子.由于这个过程的计算量较大,在表面构建过程中,我们只对液体表面及临近的网格计算其相应的标量场以减少计算量.为了方便从邻近的众多粒子中确定哪些粒子对标量值的计算产生贡献,我们在算法中采用了文献[35]中提出的数据结构,即为仿真过程中的每一个粒子构建一个句柄(handler)以区分不同的粒子和网格标识id,结构startIndices包含 $i_{\max} \times j_{\max} \times k_{\max}$ 个元素以记录每个网格中包含粒子的句柄入口值.这样,仿真空间中任意网格所包含的粒子信息都可以通过访问上述结构获取,进而方便地由公式(17)完成标量值的求解.

在表面重建中采用的是经典的MC(marching cubes)方法^[36].MC方法最早在利用医学图像获得人体组织三维模型的研究中提出,并在计算机图形学其他领域及流体仿真中得到广泛的应用^[33,35,37].本文采用经典的MC算法对基于半拉格朗日的液体实时仿真方法作进一步的验证.

完成液体表面网格的确定和标量场的计算后,即可应用MC方法以获得标量场所确定的等值面^[37].MC方法通过遍历整个空间域的网格来完成所有表面三角面片的构建.在每个网格的处理过程中,如果立方网格8个顶点的标量值都为正或都为负,则整个网格都在表面一侧,因而没有任何三角形面片生成;否则,根据网格顶点标量值的分布关系,MC算法通过线性插值获得三角形面元的顶点,以进一步生成整个液体表面.由于每个立方网格都有8个顶点并且每个顶点的标量值可能为正也可能为负,因此单个网格将会产生 $2^8=256$ 种可能的配置,从而产生256种液体表面与网格的相交关系,而应用立方网格自身的对称性,可以将这个数量减少到15种独一无二的情况.由于本文在表面建模中仅针对液体表面及邻近的网格执行MC算法,避免了在整个三维空间的所有网格中运行MC算法,从而可以大大减少在非表面区域进行标量场计算和MC算法运行的冗余计算量.

5 实验结果与分析

本文提出了一种基于半拉格朗日的液体实时仿真方法,以上述介绍的流体控制方程的求解及基于粒子的液体仿真方法为基础,应用了液体表面跟踪及建模的方法生成最终表面.仿真实验的目的是检验本文提出的基于半拉格朗日液体仿真方法的效果和实时性,并通过液体表面的建模验证该仿真方法对液体表面生成及后续应用的可行性.因此,实验部分主要包括两方面的内容:实时液体仿真及仿真液体表面生成.我们在实验中采用OpenGL可视化工具,并加入简单材质属性、视角变化控制以及环境光照等效果.为了验证算法在普通便携式PC上运行的可行性,整个仿真实验过程我们选用的硬件运行环境为Pentium(R) Dual-Core CPU(2.20GHz)、2.00GB内存、NVIDIA GeForce GT 240M显卡.

5.1 实时液体仿真

液体仿真实验中,我们以三维液体仿真为实验对象,并将液体视为不可压缩流体,即其密度在整个仿真过程中保持恒定.速度场的求解采用不可压缩纳维-斯托克斯方程对所要研究的空间域和时间域进行离散化求解,并由求解出的速度场驱动粒子进行液体的仿真.初始化过程中设置了初始条件,并在整个仿真过程中应用了上述介绍的边界条件和稳定性条件.

通过本文算法获得的三维液体粒子仿真结果如图3所示,图中展示的是单个水滴落入水面的波动过程.图中分别列举了以 δt 为0.01s时间间隔,在0.84s,2.67s及3.57s的仿真时间时粒子的分布情况.仿真过程采用 $30 \times 30 \times 30$ 网格分布,每个网格分布有27个粒子(27 particles per cell,简称ppc),且在每个维度上都随机排列.另外,为了模拟液体在方形水缸中的运动状况,我们对液体周围4个方向及底部全部采用自由滑动条件.

由于每一时刻各个单元格的速度值及粒子的位置都通过仿真获得,图3在显示液体形状的同时,还给出了

对应时刻的速度场情况(如图3中第2行所示).仿真初始条件中,我们给定水滴纵向向下的初速度,并在向下重力的作用下完成整个仿真过程.在0.84s显示的是水滴刚落入水面时形成的圆形水涡及水涡周围形成的水花.从速度场分布可以看出,最大速度分布在水滴落点的周围,这与当前水花向外扩散的实际情况相吻合.当水花向外扩散结束后,更多的液体向中心回流,这从2.67s时的速度场可以看出.此时,液体中部具有最大速度,粒子沿着这个方向向上移动,并形成水柱的雏形.水滴形成的水柱在速度场的作用下不断上升,在3.57s时达到最高值.在这个过程中,形成水柱的粒子也不断减速,完成速度上行到下行的过渡.整个液面在这样的周期性速度场的影响下继续完成若干次起伏,最终在10s左右的时间基本趋于平稳.

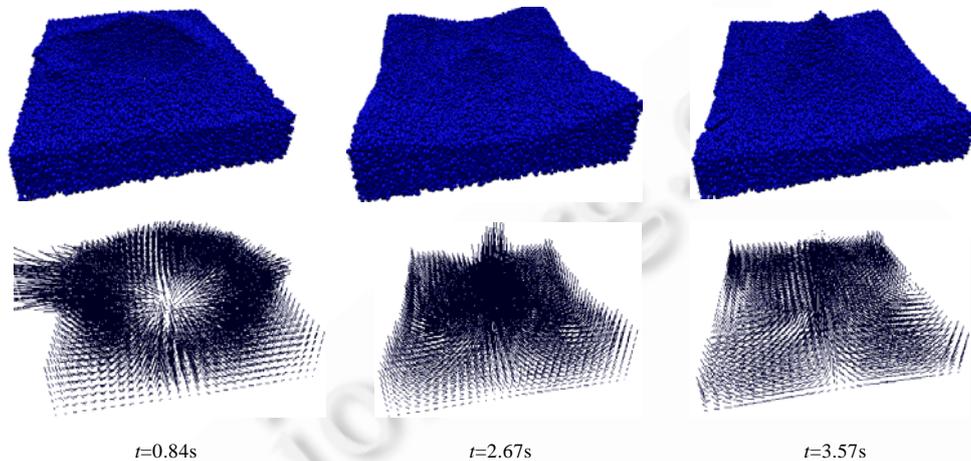


Fig.3 Visualization of 3D fluid particle simulation

图3 三维液体粒子仿真结果

为了反映本文提出的仿真方法的时间复杂度,我们分别对算法在不同离散粒度情况下的时间损耗情况进行了比较.图4所示的是仿真刷新帧率与空间离散化粒度之间的曲线关系.在图4中,单个网格粒子个数由不同图例所表示的曲线进行反映.图4中的曲线分别对空间网格精细度为 $10 \times 10 \times 10$, $20 \times 20 \times 20$, $30 \times 30 \times 30$, $40 \times 40 \times 40$ 以及单个网格粒子个数为8, 27, 64和125几种情况进行分析.仿真帧率通过对仿真过程运行10次并对帧率(frame per second,简称fps)求平均值的方法获得,单次仿真的运行时间为20s, δt 为0.01s.

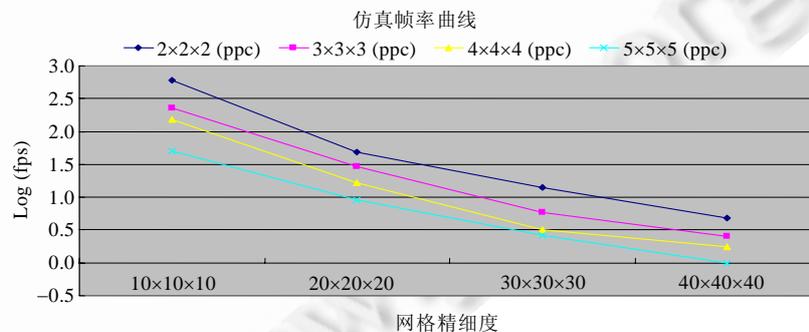


Fig.4 FPS curve for particle simulation

图4 粒子仿真刷新帧率曲线

图4中,纵轴采用了帧率的对数值作为最终的显示结果.从图中不难看出,当采用固定ppc值时,仿真帧率的对数值 $\log(\text{fps})$ 基本上随空间网格的精细度呈线性趋势下降.这反映出仿真计算量随空间网格分割精细度的增加呈幂级增加的特点,从而与算法计算复杂度 $O(N^3)$ 的实际情况相符合,其中, N 是单个维度上分割的网格总数.

本文提出的基于半拉格朗日方法的粒子仿真由于在求解流体控制方程数值解时主要的目的是获得仿真空间的速度场,继而驱动空间中的粒子运动,因此仿真的主要计算量更多与速度场的网格精细度和求解相关而非粒子个数.这一点也可以从仿真帧率曲线上看出来,例如在 $5 \times 5 \times 5$ ppc 曲线中,当网格精细度为 $20 \times 20 \times 20$ 时,整个仿真空间中基本保持有 1 000 000 个粒子,此时的帧率为 9.06fps;而在 $3 \times 3 \times 3$ ppc 曲线中,当网格精细度为 $30 \times 30 \times 30$ 时,整个仿真空间中约有 729 000 个粒子,此时对应的仿真帧率为 5.92fps.尽管后者的粒子个数较前者少很多,但是由于后者对空间网格划分得更细,在求解速度场时将耗费更多的计算量,从而有更低的仿真帧率.

实时性是该实验另一个需要分析的内容,这一点可从图 4 中的曲线分析得到.当网格精细度为 $10 \times 10 \times 10$, $20 \times 20 \times 20$ 时,无论单个网格粒子个数取上述何值,都可以将平均帧率保持在每秒 10 帧以上或接近 10 帧,从而具有较好的实时性.而此时,仿真所驱动的粒子个数最高已经达到了 100 万个.另外,在网格精细度为 $30 \times 30 \times 30$, ppc 为 $2 \times 2 \times 2$ 时,仿真帧率也高达 14.26fps,也能很好地满足实时性的要求,并得到较好、较精细的仿真效果(如图 3 所示的仿真结果).当 ppc 进一步提高到 $3 \times 3 \times 3$ 时,对应的仿真帧率为 5.92fps,也基本能够满足仿真中进行交互的要求.这里需要说明的是,本实验为验证算法在普通个人 PC 上应用和进一步扩展的能力,在实验中并未采用高性能计算机或工作站.不难想象,通过提高计算机的运行速度,或采用最新的计算方法,如并行算法等,对本算法进行优化,可以进一步提高本文提出算法的实时性.

5.2 仿真液体表面生成

上述实验主要围绕基于粒子的液体仿真及实时性分析展开,为了验证这种仿真方法对液体表面重建的能力,以及后续应用更高级可视化方法的可行性,本文还对液体粒子仿真的结果进行了表面生成的实验.应用本文第 4 节介绍的液体表面跟踪及建模的方法,对仿真结果通过完成标量场的计算以及执行 MC 算法,最终绘制得到了如图 5 所示的液体表面.

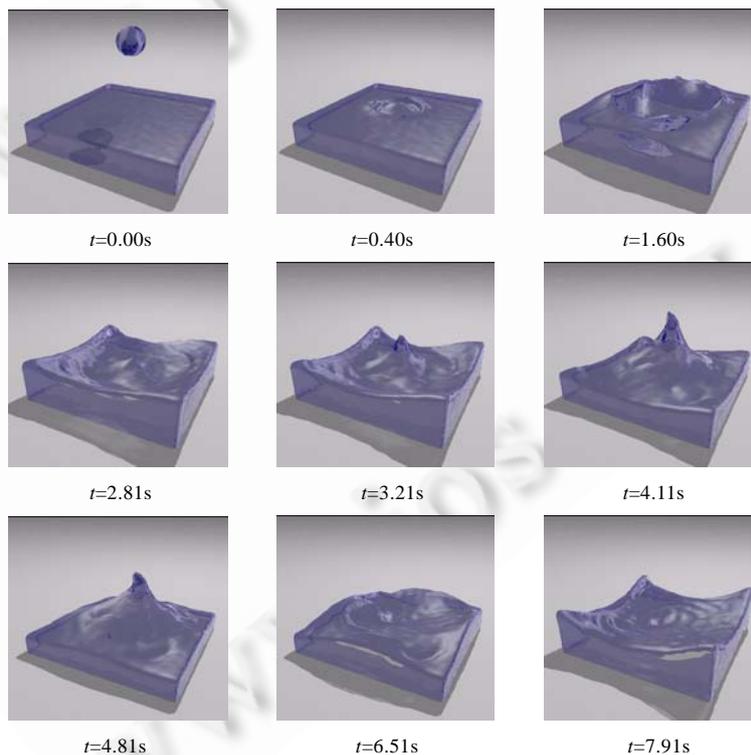


Fig.5 Surface constructed for 3D fluid

图 5 三维流体表面生成结果

实验过程中在保证运算速度的情况下,为了提高绘制精度,采用了 $40 \times 40 \times 40$ 的空间网格分割方法,绘制过程采用光线追踪的方法。

图 5 中包括了初始配置、水涡扩散及水柱上升与下落几个关键现象的程序运行效果。图 5 在 $t=0.00s$ 时显示了单个下落水滴及静止水面的基本配置情况;在仿真时间 $0.40s$ 至 $1.60s$ 的过程中,可以反映出水滴刚入水及造成水涡向四周扩散的变化效果;在 $2.81s$ 至 $4.11s$ 时间过程中,列举了从水涡到达四周边界后向中部回流最终形成完整水柱的过程。另外,图 5 中分别用 $4.81s$ 、 $6.51s$ 及 $7.91s$ 的仿真结果逼真地反映出水柱从最高点开始下降,最终到达水平面以下最低点的整个运动过程。由此,我们可以得到这样的结论:利用本文提出的一种基于半拉格朗日方法进行的液体仿真,对液体表面的生成方法简单、效果真实,符合自然规律。

6 结束语

液体作为自然现象的重要组成部分,对液体的仿真成为计算机图形学的研究热点。本文总结了近些年液体仿真的主要研究方法与成果,结合流体仿真原理,提出并实现了一种基于半拉格朗日的实时液体仿真方法,并针对该方法的粒子仿真结果进行了液体表面的跟踪和建模。实验结果表明,该方法针对三维的液体仿真具有运算速度快、视觉效果逼真的优势。将来的研究方向将集中在如何利用更逼真的渲染方法对复杂环境中的液体表面进行绘制,以及对本文算法进行扩展,以运用于大规模高精度且形式多变的流体仿真中。

References:

- [1] Mullen P, Crane K, Pavlov D, Tong YY, Desbrun M. Energy-Preserving integrators for fluid animation. *ACM Trans. on Graphics*, 2009,28(3):381–388. [doi: 10.1145/1531326.1531344]
- [2] Fedkiw R, Stam J, Jensen HW. Visual simulation of smoke. In: Pockock L, ed. *Proc. of the 28th Annual Conf. on Computer Graphics and Interactive Techniques*. New York: ACM Press, 2001. 15–22. [doi: 10.1145/383259.383260]
- [3] Enright D, Marschner S, Fedkiw R. Animation and rendering of complex water surfaces. *ACM Trans. on Graphics*, 2002,21(3): 736–744. [doi: 10.1145/566570.566645]
- [4] Tomita Y, Tsuruno R. Motion texture animation of water surface. In: Grimm C, ed. *Proc. of the SIGGRAPH 2010*. New York: ACM Press, 2010. 110–111. [doi: 10.1145/1836845.1836963]
- [5] Hong JM, Lee HY, Yoon JC, Kim CH. Bubbles alive. *ACM Trans. on Graphics*, 2008,27(3):481–484. [doi: 10.1145/1399504.1360647]
- [6] Kim D, Song OY, Ko HS. A practical simulation of dispersed bubble flow. *ACM Trans. on Graphics*, 2010,29(4):701–705. [doi: 10.1145/1833349.1778807]
- [7] Thurey N, Sadlo F, Schirm S, Muller-Fischer M, Gross M. Real-Time simulations of bubbles and foam within a shallow water framework. In: Gleicher M, Thalmann D, eds. *Proc. of the 2007 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. Aire-la-Ville: Eurographics Association, 2007. 191–198.
- [8] Horvath C, Geiger W. Directable, high-resolution simulation of fire on the GPU. *ACM Trans. on Graphics*, 2009,28(3):411–418. [doi: 10.1145/1531526.1531347]
- [9] Hasinoff SW. Three-Dimensional reconstruction of fire from images [MS. Thesis]. Toronto: University of Toronto, 2002.
- [10] Thurey N, Muller-Fischer M, Schirm S, Gross M. Real-Time breaking waves for shallow water simulations. In: *Proc. of the 15th Pacific Conf. on Computer Graphics and Applications*. Washington: IEEE Computer Society, 2007. 39–46. [doi: 10.1109/PG.2007.54]
- [11] Harris MJ. Real-Time cloud simulation and rendering [Ph.D. Thesis]. Chapel Hill: The University of North Carolina at Chapel Hill, 2003.
- [12] Wicke M, Ritchie D, Klingner BM, Burke S, Shewchuk JR, O'Brien JF. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. on Graphics*, 2010,29(4):491–4911. [doi: 10.1145/1778765.1778786]
- [13] Wojtan C, Turk G. Fast viscoelastic behavior with thin features. *ACM Trans. on Graphics*, 2008,27(3):471–478. [doi: 10.1145/1399504.1360646]
- [14] Hughes WF, Brighton JA. *Schaum's Outline of Fluid Dynamics*. 3rd ed., New York: McGraw-Hill Book, 1999. 1–362.

- [15] Liu YQ, Liu XH, Zhu HB, Wu EH. Physically based fluid simulation in computer animation. *Journal of Computer-Aided Design & Computer Graphics*, 2005,17(12):9–21 (in Chinese with English abstract).
- [16] Zhao QP. Data acquisition and simulation of natural phenomena. *Science China (Series F)*, 2011,41(4):385–416 (in Chinese with English abstract).
- [17] Foster N, Metaxas D. Realistic animation of liquids. *Graphical Models Image Processing*, 1996,58(5):471–483. [doi: 10.1006/gmip.1996.0039]
- [18] Stam J. Stable fluids. In: Waggenspack W, ed. *Proc. of the 26th Annual Conf. on Computer Graphics and Interactive Techniques*. New York: ACM Press, 1999. 121–128. [doi: 10.1145/311535.311548]
- [19] Foster N, Fedkiw R. Practical animation of liquids. In: Pockock L, ed. *Proc. of the 28th Annual Conf. on Computer Graphics and Interactive Techniques*. New York: ACM Press, 2001. 23–30. [doi: 10.1145/383259.383261]
- [20] Losasso F, Shinar T, Selle A, Fedkiw R. Multiple interacting liquids. *ACM Trans. on Graphics*, 2006,25(3):812–819. [doi: 10.1145/1179352.1141960]
- [21] Carlson M, Mucha PJ, Turk G. Rigid fluid: Animating the interplay between rigid bodies and fluid. *ACM Trans. on Graphics*, 2004, 23(3):377–384. [doi: 10.1145/1186562.1015733]
- [22] Chentanez N, Goktekin TG, Feldman BE, O'Brien JF. Simultaneous coupling of fluids and deformable bodies. In: O'Sullivan C, ed. *Proc. of the 2006 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. Aire-la-Ville: Eurographics Association, 2006. 83–89.
- [23] Solenthaler B, Pajarola R. Predictive-Corrective incompressible SPH. *ACM Trans. on Graphics*, 2009,28(3):401–406. [doi: 10.1145/1531326.1531346]
- [24] Muller M, Charypar D, Gross M. Particle-Based fluid simulation for interactive applications. In: Parent R, Singh K, eds. *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. Aire-la-Ville: Eurographics Association, 2003. 154–159.
- [25] Hu XY, Adams NA. An incompressible multi-phase SPH method. *Journal of Computational Physics*, 2007,227(1):264–278. [doi: 10.1016/j.jcp.2007.07.013]
- [26] Bridson R. *Fluid Simulation for Computer Graphics*. CRC Press, 2008. 1–246.
- [27] Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *Physics of Fluids*, 1965,8(12):2182–2189. [doi: 10.1063/1.1761178]
- [28] Golub GH, Van Loan CF. *Matrix Computations*. 3rd ed., The Johns Hopkins University Press, 1996. 1–694.
- [29] Bolz J, Farmer I, Grinspun E, Schroeder P. Sparse matrix solvers on the GPU: Conjugate gradients and multigrid. *ACM Trans. on Graphics*, 2003,22(3):917–924. [doi: 10.1145/882262.882364]
- [30] Goodnight N, Woolley C, Lewin G, Luebke D, Humphreys G. A multigrid solver for boundary value problems using programmable graphics hardware. In: Molnar S, Pfister H, eds. *Proc. of the 2003 ACM SIGGRAPH/Eurographics Conf. on Graphics Hardware*. Aire-la-Ville: Eurographics Association, 2003. 102–111.
- [31] Desbrun M, Gascuel M. Smoothed particles: A new paradigm for animating highly deformable bodies. In: Boulic R, Hegron G, eds. *Proc. of the Eurographics Workshop on Computer Animation and Simulation*. New York: Springer-Verlag, 1996. 61–76. [doi: 10.1007/978-3-7091-7486-9_5]
- [32] Tonnesen D. *Dynamically coupled particle systems for geometric modeling, reconstruction, and animation [Ph.D. Thesis]*. Toronto: University of Toronto, 1998.
- [33] Yu J, Turk G. Reconstructing surfaces of particle-based fluids using anisotropic kernels. In: Faure F, Larboulette C, eds. *Proc. of the 2010 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. Aire-la-Ville: Eurographics Association, 2010. 217–225.
- [34] Fraedrich R, Auer S, Westermann R. Efficient high-quality volume rendering of SPH data. *IEEE Trans. on Visualization and Computer Graphics*, 2010,16(6):1533–1540. [doi: 10.1109/TVCG.2010.148]
- [35] Akinci G, Ihmsen M, Akinci N, Teschner M. Parallel surface reconstruction for particle-based fluids. *Computer Graphics Forum*, 2012,31(6):1797–1809. [doi: 10.1111/j.1467-8659.2012.02096.x]
- [36] Lorensen WE, Cline HE. Marching cubes: A high resolution 3D surface construction algorithm. In: Stone MC, ed. *Proc. of the 14th Annual Conf. on Computer Graphics and Interactive Techniques*. New York: ACM Press, 1987. 163–169. [doi: 10.1145/37402.37422]

- [37] Hong W. An adaptive sampling approach to incompressible particle-based fluid [Ph.D. Thesis]. Texas: Texas A&M University, 2009.

附中文参考文献:

- [15] 柳有权,刘学慧,朱红斌,吴恩华.基于物理的流体模拟动画综述.计算机辅助设计与图形学学报,2005,17(12):9-21.
[16] 赵沁平.自然现象的数据获取与模拟.中国科学(F 辑),2011,41(4):385-416.



邹玲(1986—),女,山东淄博人,博士生,主要研究领域为流体仿真,计算机图形学.
E-mail: zouling@vrlab.buaa.edu.cn



赵沁平(1948—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为虚拟现实,分布式应用系统.
E-mail: zhaoqp@vrlab.buaa.edu.cn



齐越(1969—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为多媒体技术,虚拟现实,计算机图形学.
E-mail: qy@vrlab.buaa.edu.cn

www.jos.org.cn

www.jos.org.cn