

文章编号:1671-6833(2011)01-0103-04

光滑粒子流体动力学流体仿真交互性研究

谭同德,高志国,赵红领,石奇波

(郑州大学 信息工程学院,河南 郑州 450001)

摘要: 为了加强虚拟水体的交互性,在基于物理的水体仿真基础上,应用并扩展了光滑粒子流体动力学技术,提出了一种基于粒子的流体仿真算法.该算法通过对光滑粒子流体动力学基本算法进行扩展,解决了虚拟水体与虚拟场景中的静态实体以及动态实体的交互问题,并给出了仿真算法的程序实现方法.实验表明,该算法既满足了虚拟水体渲染的实时性要求,同时又解决了虚拟现实场景中水体与其它物体的交互性问题.

关键词: 流体仿真;光滑粒子流体动力学;静态实体;动态实体;交互性

中图分类号: TP391 **文献标志码:** A

0 引言

虚拟现实场景中需要对大量的虚拟水体进行仿真,例如虚拟校园中的湖水、喷泉,虚拟实验中的实验用水等.对水体的仿真方法可以分为两种,一种是参数建模的方法^[1],对流体运动现象进行仿真,而不关心流体运动的物理特性.另外一种方法是基于物理的流体仿真方法,这种方法从流体运动的物理特性出发,通过对流体描述方程进行解析来对流体进行仿真.基于物理的流体仿真方法主要分为两种,第一种为欧拉法,是一种基于网格的方法^[2-3];第二种为拉格朗日法^[3],是一种基于粒子的方法,该方法从分析流体各个微团的运动着手,通过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,来研究整个流体的运动.

笔者采用光滑流体动力学方法对流体描述方程进行解析,同时对该方法在交互性方面进行了扩展,使之既能够满足虚拟现实应用程序中的水体渲染的实时性要求,同时又使水体能够和虚拟场景中的其他实体进行交互.

1 光滑粒子流体动力学

1.1 流体动力学基本方程

流体动力学中用来描述不可压缩流体运动的

方程是纳维-斯托克斯方程,这个方程描述了不可压缩流体的动量守恒,它和牛顿第二定律是等价的.其形式如下

$$\frac{Du}{Dt} = -\frac{1}{\rho} \nabla p + \mu \nabla^2 u + f \quad (1)$$

式中: ρ 是流体的密度; u 是流体的速度; p 是流体的压力; μ 是流体的运动黏性系数; f 是作用在流体每个单位质量上的外力(通常是重力或者碰撞产生的力等).为了能够进行流体的仿真,必须对公式1进行解析.

1.2 光滑粒子流体动力学基础

光滑粒子流体动力学(Smoothed Particle Hydrodynamics,简称SPH)是一种用于流体动力学的插值方法,可以通过插值计算出流体的物理特性.

公式2给出了求解任意位置 r 上的标量 $A(r)$ 的SPH方法:

$$A(r) = \sum_j m_j \frac{A_j}{\rho_j} W(r - r_j, h) \quad (2)$$

标量 $A(r)$ 可以是连续流体的任意一个属性,如:密度、压力等; j 表示对流体空间的所有粒子进行遍历; r_j 表示粒子 j 的位置, m_j 表示粒子 j 的质量; ρ_j 表示粒子 j 的密度; A_j 表示粒子 j 的标量值.而 $W(r - r_j, h)$ 叫核函数或平滑核, h 是这个核函数的有效范围,笔者使用的核函数是规范化的,即它具有如下性质:

$$\int W(r - r_j, h) dr = 1 \quad (3)$$

收稿日期:2010-08-11;修订日期:2010-10-20

作者简介:谭同德(1950-),男,河南郑州人,郑州大学教授,博士,主要研究方向为CAD、计算机图形学、虚拟现实等;

高志国(1977-),男,汉,河南安阳人,硕士研究生,主要研究领域为计算机图形学、虚拟现实技术.

使用 SPH 方法对流体进行仿真,需要对纳维-斯托克斯方程(公式1)进行解析。Müller 等人在文献[4]中详细介绍了如何利用 SPH 方法来实现稳定快速的流体仿真。从拉格朗日方法的角度来看,纳维-斯托克斯方程左边的项 Du/Dt 就是一个给定粒子的加速度。如果对于一个给定的粒子,计算出其密度和纳维-斯托克斯方程右边的几项,然后计算出作用在粒子上的合力,进而可以利用运动力学原理来更新粒子的位置,这就是 SPH 的工作原理。

2 用于水体交互的 SPH 方法扩展

交互性是虚拟现实的一个重要的特性,虚拟现实场景中的水体也必然要和场景中的其他实体进行交互。笔者将对 Müller 提出的基本方法进行扩展,使之能够适应虚拟场景中流体交互性的需要。

虚拟现实场景中与流体交互的实体大体分为两类。第一种是静态类型的实体,比如容器(这里假定容器是固定不动的)。第二种是动态类型的实体,比如落入水中的木块,石块等。

2.1 与静态实体的交互

在流体空间保持当前运动状态,SPH 粒子在下一时间步进入到静态实体空间中,此时,必须计算静态实体作用于这个粒子上的反作用力,这个力迫使该粒子重新回到流体空间中。这个反作用力的计算采用文献[5]中提到的公式:

$$F = [k_s d + k_d(v \cdot n)]n \quad (4)$$

式中: d 是该粒子保持当前运动状态在下一时间步进入静态实体中的距离; k_s 和 k_d 分别是弹性常数和阻尼常数,它们均为通过实验得到的经验值; v 是相对速度; n 是碰撞点上的法线向量; F 是作用在流体粒子上的沿法线向量方向的碰撞反作用力,这个力除以该流体粒子质量得到由于碰撞产生的粒子加速度,然后将计算结果加到该粒子的纳维-斯托克斯方程中的 f 项上。

2.2 与动态实体的交互

动态实体限定为刚体(如木块等)。为了方便对刚体进行处理,把刚体也看成是由粒子组成的,这些粒子的更新方式与流体粒子的更新方式相同。在流体与刚体的交界处计算两者之间的压力时,需要做一些特殊的处理。刚体粒子会对流体粒子施加一个压力,使流体粒子离开刚体;流体粒子也会对刚体粒子产生一个压力,使刚体发生平移或旋转。式(5)和式(6)分别是作用在一个刚体粒

子和一个流体粒子上的压力公式。

$$p_{\text{rigid}} = \begin{cases} k^{\text{rigid}}(\rho^{\text{rigid}} - \rho_0^{\text{rigid}}), & \rho \geq \rho_0^{\text{rigid}} \\ 0, & \rho < \rho_0^{\text{rigid}} \end{cases} \quad (5)$$

$$p_{\text{fluid}} = \begin{cases} k^{\text{fluid}}(\rho^{\text{fluid}} - \rho_0^{\text{fluid}}), & \rho \geq \rho_0^{\text{fluid}} \\ 0, & \rho < \rho_0^{\text{fluid}} \end{cases} \quad (6)$$

式中: p^{rigid} 表示作用在刚体粒子上的压力; p^{fluid} 表示作用在流体粒子上的压力; ρ^{rigid} 和 ρ^{fluid} 分别表示当前的刚体或流体粒子的密度; ρ_0^{rigid} 是刚体的初始密度; ρ_0^{fluid} 是流体的初始密度(静止密度); k^{rigid} 和 k^{fluid} 为量纲系数,它们是通过实验得到的经验公式。式(6)会阻止流体粒子渗透到刚体中,因为压力总是正的,这样就形成了一个排斥力,迫使流体粒子向远离刚体粒子的方向运动。

在本方法中,压力分别独立地作用于每一个刚体粒子上,使它们各自独立运动。由于刚体粒子运动的独立性以及不受约束的刚体粒子更新,在一次更新操作之后,刚体并不一定保持原来的形状,因此必须对刚体形状进行修正,以保证刚体不会发生形变。为了做到这一点,在粒子更新之后,利用刚体粒子的位置和运动来计算出这个刚体新的位置和方向,以及其他的一些刚体特征值(质心 r_g , 角速度 ω , 旋转矩阵 R 等),然后再去更新刚体粒子的位置,使它们保持原刚体的形状。

根据上面的叙述,更新的刚体的算法描述如下:

- (1) 更新刚体修正前每个刚体粒子的速度 v_j' ;
- (2) 计算每个刚体粒子的质心速度 v_g ;
- (3) 计算刚体的角速度 ω ;
- (4) 计算刚体的质心 r_g ;
- (5) 计算刚体更新后的旋转矩阵 R ;
- (6) 更新刚体修正后每个刚体粒子的速度 v_j ;
- (7) 更新每个刚体粒子的位置 r_j 。

为了简化运算,笔者采用了如下的一种近似计算方法。

一个刚体的总质量可以用式(7)表示:

$$M = \sum_j m_j \quad (7)$$

式中: m_j 是单个刚体粒子 j 的质量。假设所有粒子的质量都是相等的,那么,对于一个刚体来说,其质心速度 v_g 就可以用式(8)来计算。

$$v_g = \frac{1}{N} \sum_j v_j' \quad (8)$$

式中: N 是刚体粒子的数量。刚体角速度 ω 可以用式(9)计算得到。

$$\omega = \frac{1}{I} \sum_j q_j \times v_j' \quad (9)$$

式中: q_j 是单个刚体粒子 j 在刚体的模型坐标系中的相对于质心 r_g 的位置. r_g 和 q_j 分别由式(10)和式(11)来定义. 刚体的粒子位置向量 q_j 在刚体的模型坐标系中计算, 在仿真过程中不变, 所以式(11)中的 r_{j0} 和 r_{g0} 是指初始化时的粒子的位置和质心位置.

$$r_g = \frac{1}{N} \sum_j r_j \quad (10)$$

$$q_j = r_{j0} - r_{g0} \quad (11)$$

标量变量 I 与刚体的转动惯量有关, 其定义如下.

$$I = \sum_j |q_j|^2 \quad (12)$$

利用式(7)到式(12)可以计算得到刚体的特性值, 来获得整个刚体的更新位置和方向.

要为每个刚体保存相应的旋转矩阵 $R(t)$, 以及质心 $r_g(t)$. 在仿真过程中, 可以利用下面公式来更新它们的值.

$$r_g(t + \Delta t) = r_g(t) + \Delta t v_g(t + \frac{1}{2} \Delta t) \quad (13)$$

$$R(t + \Delta t) = R(t) + \Delta t Q(t + \frac{1}{2} \Delta t)$$

这里的 $Q(t)$ 是在时间 t 上的旋转矩阵的时间变化率, 由式(14)给出. 在时间 $t + 1/2 \Delta t$ 上, 利用时间 $t + 1/2 \Delta t$ 上的粒子速度, 计算得到刚体 ω 的新值, 并利用 $R(t)$, 就可以计算出 Q 的值.

$$Q(t) = \begin{bmatrix} 0 & -\omega_z & -\omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} R(t) \quad (14)$$

一旦得到了刚体的角速度和质心速度, 利用式(15), 既可得到受刚体约束的每个粒子的速度.

$$v_j = v_g + \omega \times q_j \quad (15)$$

本研究中, 刚体粒子的更新方法和流体粒子的更新方法完全相同. 为了强化粒子的刚性, 在刚体粒子更新之后, 首先要利用式(13)计算刚体更新的位置和方向. 然后进一步更新那些组成刚体的单个粒子. 在下一个时间步长 $t + \Delta t$ 上, 每个刚体粒子的位置可以用下面的公式得到.

$$r_j(t + \Delta t) = R(t + \Delta t) (r_j(t) - r_g(t)) + r_g(t + \Delta t) \quad (16)$$

3 仿真算法实现

笔者仿真算法的仿真循环如下:

(1) 首次进入循环, 创建邻居发现网格, 转步

骤 3;

- (2) 更新邻居发现网格;
- (3) 更新粒子的邻居粒子列表;
- (4) 计算粒子的压力以及黏性力;
- (5) 计算作用在粒子上的其他力;
- (6) 更新每个粒子的速度和位置;
- (7) 更新刚体运动;
- (8) 渲染输出, 转步骤 2.

说明: 根据 SPH 核函数特性, 需要对流体空间进行网格分割, 便于能够高效的对当前粒子的邻居粒子进行迭代 (邻居粒子指相对于当前粒子的距离小于平滑半径的粒子), 并把它们保存到当前粒子的邻居粒子列表中. 更新刚体运动使用第 2 节介绍的方法及公式.

4 实验效果及分析

4.1 实验环境介绍

算法实验的硬件环境为: CPU: Pentium4 3.00G; 内存: 1GB; 图形卡: NVIDIA GeForce 7800 GTX.

算法实验的软件环境为算法在 Windows XP 下使用 C++ 语言完成; 图形 API 使用 OpenGL 2.1; 水体表面三角化构造使用 Marching Cubes 方法^[6], 并用 GPU 着色语言 Cg^[7] 做了最终的水体材质、光照、反射折射等处理.

4.2 SPH 水体与静态实体的交互

SPH 水体与静态实体交互 (无刚体) 实验参数的选取如表 1 前 4 行参数所示, 实验渲染效果如图 1 所示. 图 1 中的 4 幅图像分别演示了仿真程序中使用鼠标移动水槽时水体的运动动画的 4 帧, 其中 SPH 粒子数为 3 000. 从渲染效果图和实验结果可知, 无论实时性和水体运动的合理性都能得到满足.

表 1 本算法实验参数

Tab.1 Experiment parameters for the algorithm

参数	数值	参数	数值
平滑范围 h/m	0.0125	弹性常数 k_e	5 000.0
时间步长 $\Delta t/m$	0.005	流体粒子质量 m_j/g	0.000 205 43
黏性系数 μ	0.2	流体粒子数/个	3 000
阻尼常数 k_d	128.0	刚体粒子质量	$n * mfluidj (n > 0)$
刚体粒子数/个	208		

4.3 SPH 水体与动态实体的交互

SPH 水体与动态实体 (刚体) 交互实验参数诸如平滑半径, 时间步长, 弹性系数等的选择同表

1 所示,与刚体有关的参数选取如表 1 最后一行所示,实验效果如图 2 所示. 其中刚体粒子质量为流体粒子质量 0.6 倍,即表 1 中刚体粒子质量表达式中的 $n=0.6$,图 2 中的 4 幅图像分别演示了仿真程序中使用鼠标移动水槽时水体的运动以及与水中木块的交互动画的 4 帧. SPH 粒子数和刚体粒子数之和为 3 208,算法效率较没有刚体交互的情况有所下降. 从图及实验可知水体与刚体之间的交互同样满足运动的合理性要求.

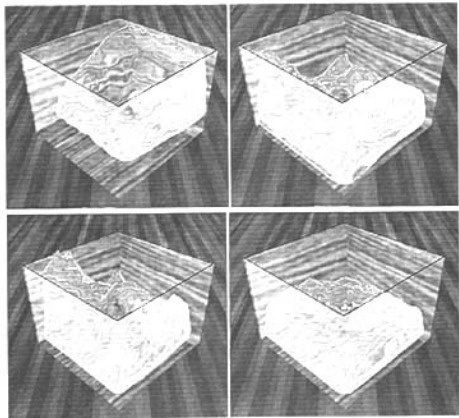


图 1 SPH 水体与静态边界交互渲染效果图
Fig.1 The renderings for interaction between sell water and statil bwders

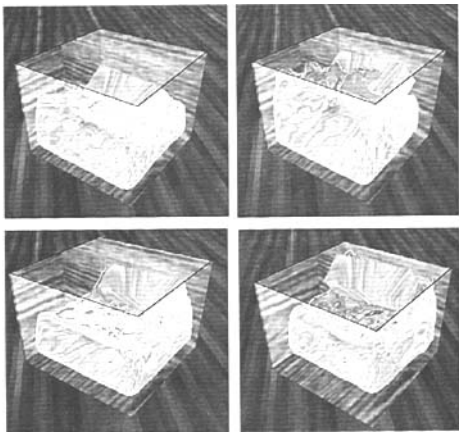


图 2 刚体与 SPH 水体交互的渲染效果图
Fig.2 The renderings for intoraction between SPH water and rigid body

表 2 给出了算法的执行效率,从表 2 可以看出,当 SPH 粒子数控制在 1 000 ~ 3 000 之间时,SPH 水体的渲染可以满足实时性的要求,同时也可看到在刚体与流体之间的交互的处理对 SPH 水体仿真算法效率具有一定的影响.

表 2 算法的执行效率

Tab.2 The efficiency of the algorithm

SPH 粒子数	无刚体时的帧速率	有刚体时的帧速率
1 000	45.74	39.81
2 000	22.06	19.78
3 000	15.77	14.31
4 000	8.35	7.61

5 结论

在 SPH 基本算法^[4]的基础上对其进行了扩展,使其能够与虚拟现实场景中的静态实体边界以及动态实体进行实时交互. 将本算法应用于虚拟现实应用程序(如数字化校园,虚拟实验等)中,可以使虚拟场景中的水体具有高度的可交互性,有助于提高用户对虚拟现实场景的沉浸感,加深用户的印象.

参考文献:

[1] 柳有权,刘学慧,朱红斌,等. 基于物理的流体模拟动画综述[J]. 计算机辅助设计与图形学学报, 2005,17(12): 2581 - 2589.

[2] STAM J. Stable fluids [C]//In: Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, California, 1999:121 - 128.

[3] LIU G R, LIU M B 著. 光滑粒子流体动力学——一种基于网格的粒子法[M]. 韩旭,杨刚,强洪夫译. 长沙:湖南大学出版社, 2005:5 - 12.

[4] MATTHIAS M, DAVID C, MARKUS G. Particle - based fluid simulation for interactive applications[C]. // Proceedings of the 2003 ACM SIGGRAPH Eurographics Symposium on Computer Animation, Eurographics Association, 2003:154 - 159.

[5] MOORE M, WILHELMS J. Collision detection and response for computer animation[C]. // Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, 1998:289 - 298.

[6] JULES B. An implicit surface polygonizer[M]. Graphics Gems IV. Academic Press. 1994. 324 - 349.

[7] RANDIMA F, MARK J. K 著. Cg 教程——可编程实时图形权威指南[M]. 洪伟,刘亚妮,李骑,等译. 北京:人民邮电出版社, 2003:153 - 178.

[8] MICHAEL D. 编著. 游戏编程精粹 6[M]. 孟宪斌,罗岱,徐丹,等译. 北京:人民邮电出版社, 2007:161 - 174.

(下转第 120 页)

- 的传递(I) — 数组[J]. 武汉大学学报:工学版, 2008, 41(3): 71 - 76.
- [2] 任慧, 周振红, 张成才. 混合计算工程中复合数据的传递(II) — 派生类型[J]. 武汉大学学报:工学版, 2008, 41(4): 63 - 68.
- [3] 周振红, 毕苏萍, 张成才. 混合计算工程中复合数据的传递(III) — 派生类型内嵌数组[J]. 武汉大学学报:工学版, 2009, 42(4): 482 - 486.
- [4] 周振红, 郭恒亮, 张君静, 等. Fortran 90/95 高级程序设计[M]. 郑州:黄河水利出版社, 2005.
- [5] 毕苏萍, 周振红. Visual Fortran 创建 Win32 API 式的 DLL [J]. 计算机工程与设计, 2008, 29(18): 4868 - 4871.
- [6] Visual Basic[EB/OL](2009-1-15). <http://www.microsoft.com/msdn/>.
- [7] Compaq Visual Fortran[EB/OL](2007-12-17). <http://www.compaq.com/fortran/>.
- [8] 毕苏萍, 张军, 周振红. CVF 对创建 Fortran COM 组件的支持[J]. 郑州大学学报:工学版, 2009, 30(2): 88 - 90.

Several Ways to Pass Single String between Fortran and VB

ZHOU Zhen-hong¹, WANG Guo-bin¹, BI Su-ping²

(1. School of Water Conservancy and Environment, Zhengzhou University, Zhengzhou 450001, China; 2. School of Civil Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: To make an interface contain richer data types in Fortran/VB mixed-language programming computation engineering, it is researched to pass single string by built-in and OLE variant types. On the basis of a discussion of string representation, storage and argument passing for the two languages, strings with a length or not are successfully passed. According to string function calling mechanism of CVF, an alternative approach is proposed to call Fortran string functions with VB. Having taken full advantage of CVF language extension, it is solved to pass a string pointer. Based on recognition of the structure of OLE variant types, the way to pass string variant types is presented with the support of CVF for Unicode BSTR. The feasibility of above-mentioned passing methods has been demonstrated by demo programming.

Key words: mixed-language programming computation engineering; interface; string data; built-in type; OLE variant type

(上接第 106 页)

Research for Interactivity of Smoothed Particle Hydrodynamics Fluid Simulation

TAN Tong-de¹, GAO Zhi-guo¹, ZHAO Hong-ling¹, SHI Qi-bo¹

(1. School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: A fluid simulation algorithm of particles based on physical theory of hydrodynamic simulation and using and extending the smooth particle hydrodynamics technique was put forward to enhance the interactivity of virtual water. The interaction problem of virtual water with static entities and dynamic entities in virtual scenes was solved by extending the basic algorithm of SPH (smoothed particle hydrodynamics) and the development method of programing was given. Experiments show that the virtual water requirement of real-time rendering was satisfied and the interactive problem of the water with static objects and dynamic objects was also solved by the algorithm.

Key words: fluid simulation; smoothed particle hydrodynamics; static entity; dynamic entity; interactivity