基于 PCISPH 的流体粒子飞溅改进方法

钮倩倩[1],林绿开[1],李 毅[1,2*]
(1. 温州大学 计算机与人工智能学院,浙江 温州 325000;
2. 浙江大学 计算机科学与技术学院,浙江 杭州 310058)

摘 要:流体飞溅是自然界中最常见的流体现象。流体模拟是计算机图形学的一个重要研究分支。流体模拟已经广泛应用于电影、游戏和其他工业数值领域。由于飞溅场景中流体的密度和压力的速率变化非常大,因此模拟对离散化解的精度要求很高。对于流体粒子飞溅模拟数值不稳定,且缺乏真实效果,针对经典的预测校正不可压缩 SPH(PCISPH),该文提出了基于粒子表面流体飞溅改进的泊松压力解的方法,通过替换压力源项提高了模拟精度。实验结果表明,该方法中流体的压力分布较之经典方法更接近现实的流体粒子飞溅模拟效果,并且改进后的方法经过剧烈运动后帧率仍符合实时性仿真效果的要求,保证了良好的压力稳定性和真实感。该文通过 Anaconda3 集成 Python3 和 Taichi 环境实现流体仿真实验。

关键词:光滑粒子流体动力学;流体模拟;粒子飞溅改进;预测校正不可压缩 SPH;泊松方程;太极编程
中图分类号:TP391.41 **文献标识码**:A **文章编号**:1673-629X(2024)02-0060-05
doi:10.3969/j.issn.1673-629X.2024.02.009

Improved Spattering of Fluid Particles Based on Predictive Correction of Incompressible SPH

NIU Qian-qian[1],LIN Lyu-kai[1],LI Yi[1,2*]
(1. School of Computer and Artificial Intelligence,Wenzhou University,Wenzhou 325000,China;
2. School of Computer Science and Technology,Zhejiang University,Hangzhou 310058,China)

Abstract:Fluid splashing is the most common fluid phenomenon in nature. Fluid modeling is an important branch of computer graphics. Fluid simulation has been widely used in film,games,and other industrial numerical fields. Because the density and pressure of the fluid in the splash scene vary largely,the simulation requires high accuracy of discrete solution. For the unstable and lack of real effect of fluid particle splash simulation,we propose an improved Poisson pressure solution method based on particle surface fluid splash for the classical predictive correction of incompressible SPH (PCISPH),which improves the simulation accuracy by replacing the pressure source term. The experimental results show that the pressure distribution of the fluid in this method is closer to the realistic fluid particle splash simulation effect compared with the classical method,and the frame rate of the improved method still meets the requirements of real-time simulation effect after intense movement,which ensures good pressure stability and realism. We implement fluid simulation experiments using Anaconda3 to integrate Python3 and Taichi environments.

Key words:smoothed particle hydrodynamics;fluid simulation;particle splash improvement;prediction correction incompressible SPH;Poisson equation;Taichi programming

0 引 言

对于流体模拟研究经历了较长时间并不断衍生出新的方法。从宏观上来说,流体的模拟可分为基于物理的方法和非物理的方法。非物理的方法往往依赖于纯粹的数学模型,需要通过观测数据对实际数据进行建模,从而达到模拟仿真的效果,该类方法没有引入物理控制方程,所以结果通常存在动量、能量不守恒的问题。

基于物理的方法往往通过离散的数值方法对控制方程来进行求解。根据采样点的不同分布可以将其细分为欧拉方法和拉格朗日方法。欧拉方法是一种网格方法,典型的代表是有限差分方法[1],这种方法放弃了

收稿日期:2023-05-07 **修回日期**:2023-09-07
基金项目:浙江省软科学研究计划重点项目(2022C25033);温州市科技计划项目(R20200025)
作者简介:钮倩倩(1997-),女,硕士研究生,研究方向为流体仿真;通信作者:李 毅(1984-),男,博士后,副教授,硕导,CCF 会员(F9506M),研究方向为计算机视觉、虚拟现实。

描述运动对象本身,转而描述运动对象所处空间场的变化。由于该方法相当于模拟了一种空间内固定的背景网格,通过运动对象对网格的影响来近似描述运动对象,所以不必考虑对象复杂的运动。拉格朗日粒子法是将客观世界中的流体看成由许多粒子组成的整体,该方法的基本思路是首先依照流体运动的物理定律,计算出粒子受到的外力和粒子之间的相互作用力,其次根据牛顿第二定律(Newton´s Second Law of Motion-Force and Acceleration),计算出每个粒子在时间步长内的位置变化量,最后通过位置变化量来模拟出一段流体运动轨迹。主流的拉格朗日[2]粒子法包括 SPH(Smoothed Particle Hydrodynamics,光滑粒子流体动力学)[3-5]和 PBF(Position Based Fluids,基于位置的流体模拟方法)[6]等。该文主要针对 SPH 进行研究。

SPH 是一种基于积分插值理论的插值方法,通过离散采样点(粒子)来近似连续场量的值和导数,这些粒子自身携带某些场量,粒子处的场量通过从相邻粒子的值加权平均得到。有限差分法要求粒子排列在规则网格上,SPH 则可以通过任意排布的粒子来求得近似。每个离散粒子都占据问题域的一小部分。

SPH 源于计算天体物理学[7]领域,用于模拟天体物理现象。Desbrun 等人[8]将 SPH 引入计算机图形学领域,用于模拟可变形体。Müller[9]将 SPH 引入流体模拟,并在此后扩展到各种问题的模拟。

在计算机图形学中,Solenthaler 和 Pajarola 提出了一种有效的 SPH 变体,即 PCISPH(Predictive-Corrective Incompressible SPH,预测校正不可压缩 SPH)[10]。在这种 SPH 变化中,通过迭代预测和校正密度来增强不可压缩性。压力的确定应使压力能够减小密度波动。预测校正不可压缩 SPH(PCISPH)允许更大的时间步长,如 ISPH(隐式 SPH)。此外,PCISPH 解决了粒子级的不可压缩性,因此不需要求解泊松方程。预测校正不可压缩 SPH(PCISPH)在一个模型中同时具有 WCSPH(Weakly Compressible SPH,弱可压缩 SPH)[11]和 ISPH[12]的优点,即每次物理更新的计算成本低,时间步长大。通过流体传播估计的密度值,并以实现不可压缩性的方式更新压力。一旦达到每个单独粒子的先前用户定义的密度变化限制,传播就会停止。其中 WCSPH 是最直接的显式求解算法,基于连续性方程、动量方程和状态方程,直接计算作用在各个颗粒上的压力、体力、粘聚力和表面张力的合力,继而求出加速度进而结合边界条件进行速度位置的更新。

由于飞溅场景中流体的密度和压力的速率变化非常大,因此模拟对离散化解的精度要求很高。传统的不可压缩 SPH[13]在许多场景中都有很好的性能,例如电影效果等。但由于数值耗散,通常不可能长期精确模拟流体的运动变化,在对数值精度要求较高的场景中仍有很大的改进空间。

该文提出一种基于 PCISPH 的流体粒子飞溅的改进方法。使用 Taichi[14]框架实现流体仿真,太极编程语言大大提高了并行编程的生产力。

1　准备工作

1.1　不可压缩泊松方程模型

二维场景中自由表面流体的方程由质量守恒方程和动量守恒方程组成[15]。该文研究的是基于粒子的流体模拟,因此将控制方程转换为拉格朗日形式,如公式 1 和公式 2 所示。

$$\frac{1}{\rho}\frac{D\rho}{Dt}+\nabla\cdot u=0 \tag{1}$$

$$\frac{Du}{Dt}=-\frac{1}{\rho}\nabla P+g+\frac{1}{\rho}\nabla\cdot\mu \tag{2}$$

其中,ρ 表示流体的密度,t 是时间,u 是流体的速度,P 是压力,g 是重力加速度,μ 是剪切应力(包括粘滞力)。由于粘滞力的保留,式(1)和式(2)可以应用于牛顿流体和非牛顿流体。

1.2　速度预测

在纳维-斯托克斯(Navier-Stokes Equation)方程中,如果只考虑应力和重力,则从以下公式可以得到粒子的速度和位置。公式如下:

$$\Delta u_{*}=(g+\frac{1}{\rho}\nabla\cdot\mu)\Delta t \tag{3}$$

$$u_{*}=u_{t}+\Delta u_{*} \tag{4}$$

$$r_{*}=r+u_{*}\Delta t \tag{5}$$

其中,u_t 和 r 表示粒子在时间 t 上的速度和位置。u_* 和 r_* 表示预测步骤结束时粒子的中间速度和位置。Δu_* 表示预测期间粒子速度的变化量,Δt 表示时间步长。

1.3　压力密度校正

方程 1 中的计算不能保证流体的不可压缩性,在 X_* 位置处的密度 ρ^* 由 SPH 得出:

$$\rho^{*}=\sum_{j}m_{j}W(|r_{i}-r_{j}|,h) \tag{6}$$

其中,r_i 是起始粒子位置,r_j 是起始粒子的邻居粒子位置,m_j 是邻居粒子质量,h 是光滑核函数半径。在密度 ρ^* 和恒定的不可压缩流体密度 ρ_0 之间存在偏差。

因此,校正步骤旨在将流体密度调整到初始值。在计算过程中,压力项将用于更新中间步骤中计算的粒子速度。公式如下:

$$\Delta u_{**}=-\frac{1}{\rho^{*}}\nabla P_{t+1}\Delta t \tag{7}$$

$$u_{t+1}=u_{*}+\Delta u_{**} \tag{8}$$

其中,Δu_{**} 表示校正过程中粒子速度的变化量,ρ^*

表示预测步骤之后和校正步骤开始之前的中间密度，u_{t+1} 和 P_{t+1} 表示在时间 $t+1$ 时刻的速度和压力值。

强制执行不可压缩性所需的压力项来自方程 1，如下：

$$\frac{1}{\rho_0}\frac{D\rho}{Dt} + \nabla \cdot \Delta u_{**} = 0 \tag{9}$$

获得压力的泊松方程如下：

$$\nabla\left(\frac{1}{\rho^*}\nabla P_{t+1}\right) = \frac{1}{\rho_0 \Delta t}\frac{D\rho}{Dt} \tag{10}$$

2 飞溅改进

压力校正。

由于预测校正不可压缩 SPH(PCISPH)是从预测的密度变化中得出压力变化，因此计算密度变化是整个过程的核心。在 SPH 中密度计算如下：

$$\rho_i = \sum_j m_j W(|r_i - r_j|, h) = \sum_j m_j W_{ij} \tag{11}$$

其中，r_i 表示目标粒子位置，r_j 表示邻居粒子位置。

该文研究流体的飞溅场景。由于该场景中流体运动强烈，压力变化率和变化范围大，容易受到误差累积的影响。因此，提出了改进的泊松压力方程源项，如下所示：

$$\left(\frac{D\rho}{Dt}\right)^* = \left(\sum_{i\neq j} m_j \frac{DW}{Dt}\right)^* \tag{12}$$

其中，上标符号 $*$ 表示在预测中计算所得。

将提出的密度变化解代入传统不可压缩流体的泊松压力方程，得到了改进的泊松压力方程式求解模型：

$$\nabla\left(\frac{1}{\rho^*}\nabla P_{t+1}\right) = \frac{1}{\rho_0 \Delta t}\left(\sum_{i\neq j} m_j \frac{DW}{Dt}\right)^* \tag{13}$$

其中，ρ^* 表示预测中的流体密度。

公式 12 简化如下：

$$\left(\frac{D\rho}{Dt}\right)^* = \left(\sum_{i\neq j} m_j \frac{DW}{Dt}\right)^* =$$

$$\left[\sum_{i\neq j} m_j\left(\frac{\partial W_{ij}}{\partial r_{ij}}\cdot\frac{\partial r_{ij}}{\partial x_{ij}}\cdot\frac{dx_{ij}}{dt} + \frac{\partial W_{ij}}{\partial r_{ij}}\cdot\frac{\partial r_{ij}}{\partial y_{ij}}\cdot\frac{dy_{ij}}{dt}\right)\right]^* =$$

$$\left(\sum_{i\neq j} m_j \nabla_i W_{ij}\cdot u_{ij}\right)^* \tag{14}$$

结合公式 13 和公式 14 得到：

$$\nabla\left(\frac{1}{\rho^*}\nabla P_{t+1}\right) = \frac{1}{\rho_0 \Delta t}\left(\sum_{i\neq j} m_i \nabla_i W_{ij}\cdot u_{ij}\right)^* \tag{15}$$

由于压力加速度 $a^p = -\frac{1}{\rho}\nabla P$，对应于速度变化 $-\Delta t\frac{1}{\rho}\nabla P$，其散度 $-\nabla\cdot(\Delta t\frac{1}{\rho}\nabla P)$ 等于每次的密度变化 $-\rho\nabla\cdot(\Delta t\frac{1}{\rho}\nabla P)$，因此得到预测的密度变化：

$$\rho^* \Delta t\cdot\nabla\left(\frac{1}{\rho^*}\nabla P_{t+1}\right) = \frac{\rho^* \Delta t}{\rho_0 \Delta t}\left(\sum_{i\neq j} m_i \nabla_i W_{ij}\cdot u_{ij}\right)^* \tag{16}$$

又在预测校正不可压缩 SPH(PCISPH)中通过预测密度变化来校正压力，公式如下：

$$\overset{\sim}{p_i} = \frac{-\rho^*_{\mathrm{err}_i}}{\beta\left(-\sum_j \nabla W_{ij}\cdot\sum_j \nabla W_{ij} - (\nabla W_{ij}\cdot\nabla W_{ij})\right)} \tag{17}$$

其中，$\beta = \Delta t^2 m^2 \frac{2}{\rho_0^2}$。

由公式 16 得：

$$\rho^*_{\mathrm{err}_i} = \frac{\rho^* \Delta t}{\rho_0 \Delta t}\left(\sum_{i\neq j} m_j \nabla_i W_{ij}\cdot u_{ij}\right)^*\cdot\Delta t =$$

$$\rho^* \Delta t\cdot\frac{1}{\rho_0}\left(\sum_{i\neq j} m_i \nabla_i W_{ij} u_{ij}\right)^* \tag{18}$$

而原预测校正不可压缩 SPH(PCISPH) $\rho^*_{\mathrm{err}_i} = \rho^* - \rho_0$，该文将 $\rho^*_{\mathrm{err}_i}$ 设为 $\max\left(\frac{\rho^*}{\rho_0}\left(\sum_{i\neq j} m_j \nabla_i W_{ij}\cdot u_{ij}\right)^* \Delta t, \rho^* - \rho_0\right)$，因此校正后的累计压力为：

$$p_i \mathrel{+}= \overset{\sim}{p_i} \tag{19}$$

3 实验结果与分析

3.1 实验环境

该文利用 Python 环境与编辑器，采用 Anaconda3 集成 Python3 和 Taichi 环境，使用 PyCharm 代码编辑平台进行数据预处理以及后续网络框架的搭建与模型训练。

Taichi 起步于 MIT 的计算机科学与人工智能实验室(CSAIL)，设计初衷是便利计算机图形学研究人员的日常工作，帮助他们快速实现适用于 GPU 的视觉计算和物理模拟算法。Taichi 是嵌入于 Python 的，使用即时编译(JIT)架构(如 LLVM，SPIR-V)，将 Python 源代码转化为 GPU 或 CPU 的原生指令，在开发时和运行时均提供优越性能。Taichi 的一大优势在于可移植性，支持多种后端。

3.2 实验对比

由于飞溅场景中流体密度和压力的变化率非常大，因此模拟对离散解的精度要求很高。针对基于粒子的流体仿真在模拟流体飞溅时的数值不稳定性，提出改进的泊松压力方程解，改进流体飞溅现象。将上文推导结果应用到模型中，得到改进后的流体模拟。下图是按时间顺序基于预测校正不可压缩 SPH 的粒子飞溅的传统方法和改进方法的效果对比。图 1 为 8 000粒子数的飞溅效果对比，图 2 为 4 000 粒子数的飞溅效果对比。

表 1 是在 8 000 个粒子的三维流体模拟实验中，在 GPU 上进行了 9 个独立实验。

基于 NVIDIA 和 Windows 平台，记录实验结果，如表 1 所示(fps 表示程序运行帧率)。

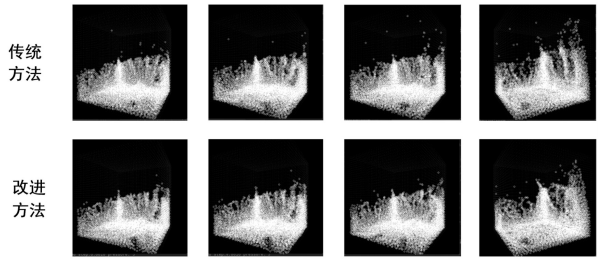

图 1　8 000 粒子飞溅效果对比

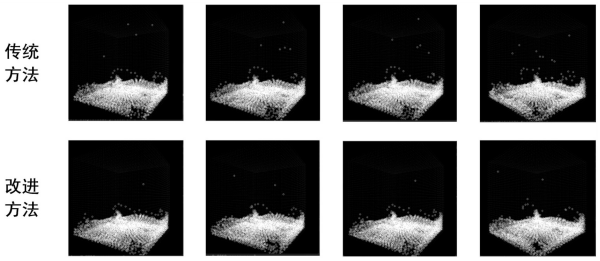

图 2　4 000 粒子飞溅效果对比

表 1　在 GPU 上运行的实验数据　fps

实验	传统方法	改进方法
1	56.76	52.22
2	55.77	52.16
3	59.24	52.09
4	56.01	50.76
5	58.92	51.26
6	52.52	46.36
7	46.63	46.96
8	51.71	45.72
9	45.49	45.23

图 3 是根据表 1 中的数据绘制出的折线图。其中实验编号表示时间先后顺序。从图中可以看出,在文中规定的测试环境下,传统方法和改进方法的帧率基本都处于 45 到 60 之间,但改进方法的帧率比传统方法的帧率略小一点,但是两种方法均达到实时性仿真效果的要求。

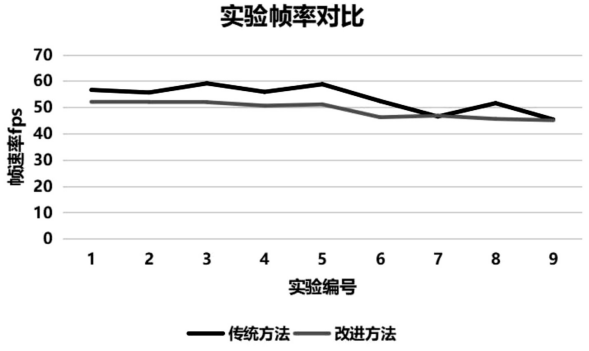

图 3　基于 GPU 的实验程序性能比较

由上面效果比较可以看出,改进后的流体粒子飞溅较之前的传统流体粒子飞溅来说有所变化,改进后的飞溅的粒子更有规律性,符合现实中在重力作用下的效果。流体仿真模拟即要求仿真模拟结果更接近现实,改进后的更符合。从表 1 与图 3 可以看出模拟均达到实时仿真要求,但改进后的性能较之改进前的较差一点。

实验核心部分即为校正密度压力值,表 2 为传统方法中的密度误差和改进方法中的密度误差(数据选择来自传统方法和改进方法飞溅的粒子数量相差较多的情况)。

表 2　校正密度误差值

实验	传统方法	改进方法
1	0.015 770	0.005 173
2	0.013 530	0.004 429
3	0.008 609	0.003 520
4	0.009 659	0.006 828
5	0.013 127	0.005 136
6	0.011 967	0.004 035
7	0.007 878	0.003 410
8	0.003 847	0.003 440
9	0.015 458	0.005 648

图 4 是由表 2 数据绘制出的折线图,其中实验编号表示运行时间先后顺序。

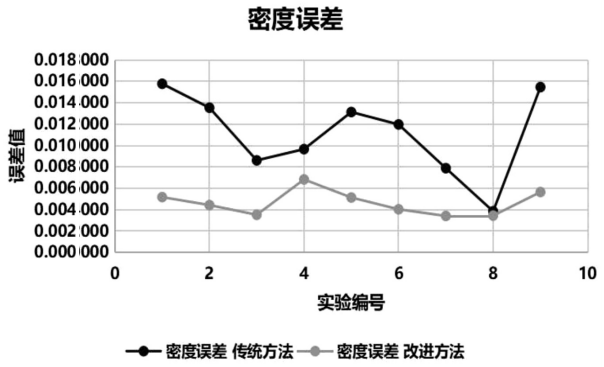

图 4　校正的密度误差值对比

虽然改进后的帧率校之传统方法有所下降,但是基于 PCISPH 的流体仿真是通过预测校正密度压力值从而得到仿真效果。实验中通过改进密度误差从而改进压力值实现流体仿真,从表 2 和图 4 中可以看出,改进后的密度误差明显小于传统方法的密度误差,并且基本上在千分之五左右。改进后密度误差变小,增强了流体的不可压缩性,减小了压力噪声,得到了更真实、更稳定的飞溅仿真效果。从而验证了改进方法的有效性。

4　结束语

该文主要针对预测校正不可压缩 SPH(PCISPH)流体仿真进行粒子飞溅改进,针对改进减小密度误差,增强流体的不可压缩性,减小了压力噪声,从而得到了

优于原模型的仿真效果。但是该模型也存在相应的问题,模型不够稳定,该文添加较多粒子来减小不稳定带给实验结果的影响,在未来将对这些方面进行深入研究,以达到稳定且较大时间步长的流体模拟。同时,该文没有对该模型进行渲染,未来将对模型进行渲染。

参考文献:

[1] MARGHANY M. Finite difference model for modeling sea surface current from RADARSAT-1 SAR data[C]//2009 international geoscience and remote sensing symposium. Cape Town:IEEE,2009:487-490.

[2] UMMENHOFER B, PRANTL L, THUEREY N, et al. Lagrangian fluid simulation with continuous convolutions [C]//International conference on learning representations. Addis Ababa:ICLR,2020.

[3] 华 磊. 基于 SPH 的统一流体模拟及细节优化研究[D]. 镇江:江苏大学,2021.

[4] DAN K, JAN B, BARBARA S, et al. A survey on SPH methods in computer graphics[J]. Computer Graphics Forum,2022,41(2):737-760.

[5] MARKUS I, JENS O, BARBARA S, et al. SPH fluids in computer graphics[C]//The 35th annual conference of the european association for computer graphics. Strasbourg:EA, 2014:21-42.

[6] MACKLIN M,MÜLLER M. Position based fluids[J]. ACM Transactions on Graphics,2013,32(4):1-12.

[7] 吴德阳,唐 勇,刘浩阳,等. 基于物理的不可压缩流体模拟技术综述[J]. 高技术通讯,2020,30(11):1189-1204.

[8] PEREGRINE D H. Water-wave impact on walls[J]. Annual Review of Fluid Mechanics,2003,35(1):23-43.

[9] BULLOCK G N,OBHRAI C,PEREGRINE D H,et al. Violent breaking wave impacts. part 1: results from large-scale regular wave tests on vertical and sloping walls[J]. Coastal Engineering,2007,54(8):602-617.

[10] SOLENTHALER B, PAJAROLA R. Predictive-corrective incompressible SPH[M]//ACM SIGGRAPH 2009 papers. New York:ACM,2009:1-6.

[11] BECKER M,TESCHNER M. Weakly compressible SPH for free surface flows[C]//Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on computer animation. Goslar:EA,2007:209-217.

[12] AI Mingjing, LI Feng, ZHENG Aiyu. Improved method of splash fluid simulation based on ISPH[C]//Proceedings of the 2nd international conference on computer science and software engineering. New York:CSSE,2019:79-83.

[13] HU X Y,ADAMS N A. An incompressible multi-phase SPH method[J]. Journal of Computational Physics, 2007, 227 (1):264-278.

[14] HU Yuanming, ANDERSON L, LI T, et al. DiffTaichi: differentiable programming for physical simulation[J]. arXiv: 1910.00935v3,2019.

[15] 张义群. 基于 SPH 的流体模拟中流固边界及液体表面处理算法的研究[D]. 合肥:合肥工业大学,2020.