

# 图形学作业报告

茶壶倒水的建模与动画

5090379126

虞文豪

## 目录

概述.....	3
算法说明.....	3
N-S 方程.....	3
SPH 方法.....	4
SPH 方法实现.....	5
碰撞检测.....	5
算法实现效果.....	6
程序内容说明.....	7
程序使用说明.....	7

## 概述

我选择完成的是三题中的第二题：茶壶倒水的建模与动画。在实现的算法上使用了 Smoothed Particle Hydrodynamics(以下简称 SPH)方法求解 Navier-Stokes 方程(以下简称 N-S 方程)对水流进行物理模拟。目前的水流有流体的性质，但是要完全与水逼真还需要一些参数或者算法上的改进与调整。

导入了一个 Maya 制作的杯子作为放水的容器。导入了一个用 3ds Max 制作的小房间作为场景，场景比较简陋，之后几次作业中可能会换成更加复杂的场景。

碰撞检测方面目前只是用了简单的圆柱体计算，之后可以进一步改进为刚体粒子模型来计算碰撞。

渲染方面添加了简单的光照与桌面的问题，没有对水进行渲染，因此水流是用粒子来表示的。

程序使用 C++与 OPENGL 开发，使用 glut 库提供的窗口来显示。

目前因为粒子数量没有特别多，而且没有进行水流的渲染，再加上使用了多线程，因此可以一定程度上保持实时性(在我的电脑上跑 2000 个粒子大约 30FPS)，等到之后的作业中加入更复杂的渲染或是对算法进行了进一步改进之后，可能会需要将动画渲染成视频等。

## 算法说明

本题使用了 SPH 算法来对水流进行物理模拟。SPH 是一种无网格算法，它由 Lucy, Gingold 以及 Monaghan 提出，它的核心思想是将流体离散化成粒子，分别对粒子进行属性上的计算以模拟真实流体的物理特性。算法中有许多数学和流体力学的东西，我只是了解并大致理解了一下，具体的推导和深入的理解还需要花更多时间。由于 SPH 很复杂，这里只能比较简单地加以说明，关于 SPH 更具体的说明和算法可以在

[http://kucg.korea.ac.kr/education/CSCE460/ref/p\\_Mue03b.pdf](http://kucg.korea.ac.kr/education/CSCE460/ref/p_Mue03b.pdf) 这篇论文中找到，这次作业中的算法基本就是用了这篇论文中的算法。

## N-S 方程

要模拟流体的物理特性，也就是要根据 N-S 方程来进行流体在各个时间点上状态的求解。

N-S 方程本身是个非常复杂而庞大的方程，通过一些流体性质上的简化可以简化方程，在水流的建模中，可以认为水流是不可压缩的流体，因此在流体模拟的求解过程中主要用到以下两个方程：

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (1)$$

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}, \quad (2)$$

其中  $\rho$  为流体在某点的密度， $\mathbf{v}$  为速率， $p$  为流体受到的压力， $\mathbf{g}$  为外力， $\mu$  为流体的粘滞系数

第一个方程保证了质量和不变。

第二个方程中， $\frac{\partial \mathbf{v}}{\partial t}$  描述了流体的加速度， $\mathbf{v} \cdot \nabla \mathbf{v}$  描述了流体的速度场随着速度而移动，右边的三项分别代表着流体受到的压力，外力以及粘滞力。

## SPH 方法

SPH 方法中最核心的方程为

$$A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h), \quad (3)$$

$A_S(\mathbf{r})$  为空间某点  $\mathbf{r}$  的流体的属性， $m_j$ ， $\rho_j$ ， $A_j$  为在离  $\mathbf{r}$  点距离为  $\mathbf{r}_j$  处的点的质量，密度以及相应的流体属性。 $W(\mathbf{r}, h)$  称为 Smoothing Kernel 函数，它具有全局积分为 1 的性质。

它代表我们可以通过在一个空间中所有离散点被赋予的属性来近似计算出在真实的连续场下这点的属性，因此我们可以通过这个方程以及它的梯度形式

$$\nabla A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \quad (4)$$

和拉普拉斯算子形式

$$\nabla^2 A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h). \quad (5)$$

来计算流体模型中离散的点的流体属性。

## SPH 方法实现

SPH 方法的具体实现基本可以分成四个步骤：计算粒子密度及粒子产生的压力，计算粒子受到的压力，计算粒子受到的粘滞力，利用受力计算加速度并更新位置。

使用

$$\rho_S(\mathbf{r}) = \sum_j m_j \frac{\rho_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) = \sum_j m_j W(\mathbf{r} - \mathbf{r}_j, h). \quad (6)$$

来计算粒子所在位置的流体密度

然后使用

$$p = k(\rho - \rho_0), \quad (7)$$

计算出粒子产生的压力，这里  $k$  为气体常数，与温度有关， $\rho_0$  为流体静止密度，由于是模拟的水流，因此这里是  $1000\text{kg/m}^3$ 。

再使用

$$\mathbf{f}_i^{\text{pressure}} = - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (6)$$

计算出在该点处流体受到的压力

使用

$$\mathbf{f}_i^{\text{viscosity}} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h). \quad (7)$$

计算出该点处的粘滞力

将这些计算出的力加上重力计算出加速度，然后更新粒子的速度以及位置得到新的流体状态。

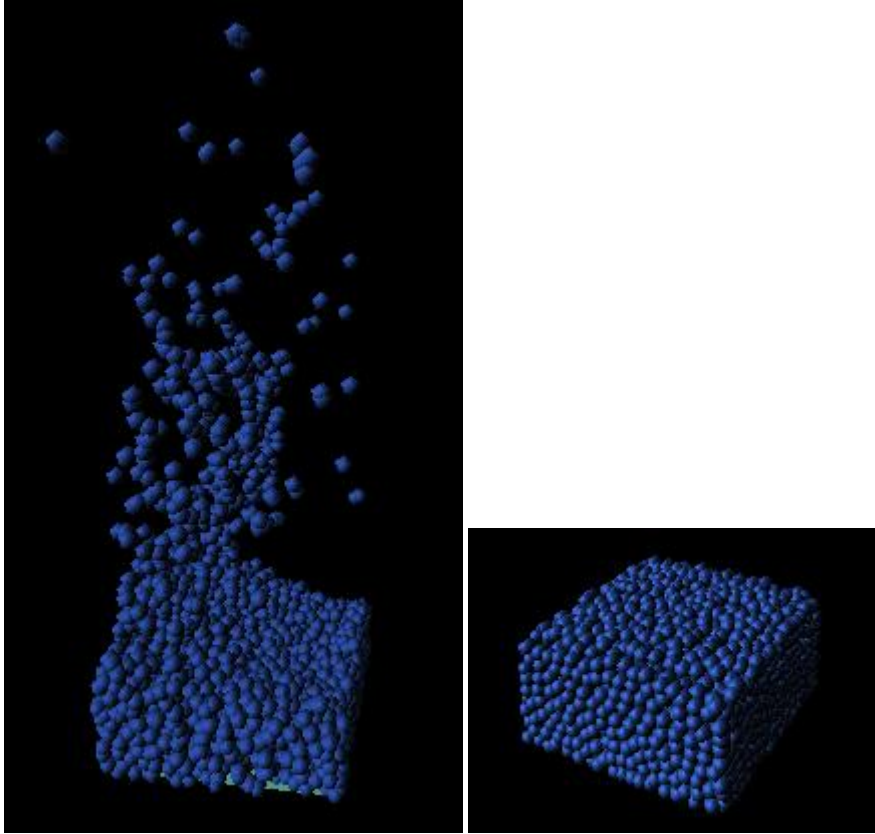
## 碰撞检测

这次作业中我只是用了非常简单的碰撞检测，在和茶杯差不多的位置设置了一个圆柱体，通过计算粒子与圆柱体的相对位置来确定是否碰撞以及碰撞后的位置与速度。

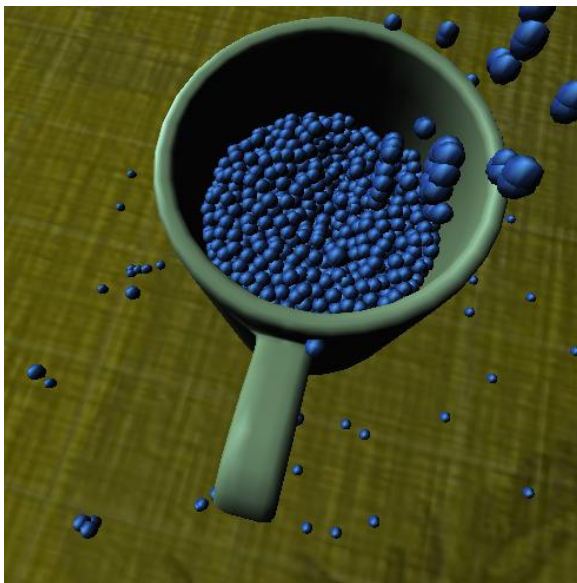
之后打算通过算法在茶杯的位置建立一个刚体粒子模型，即用与 SPH 算法一样的思想将茶杯离散化，然后将这些粒子与流体的粒子一起进行 SPH 计算，但是不进行速度、粘滞力以及位置更新的计算，相当于用原子做了一面墙。这样的方法应该会比简单的反弹粒子要更加

真实一些。

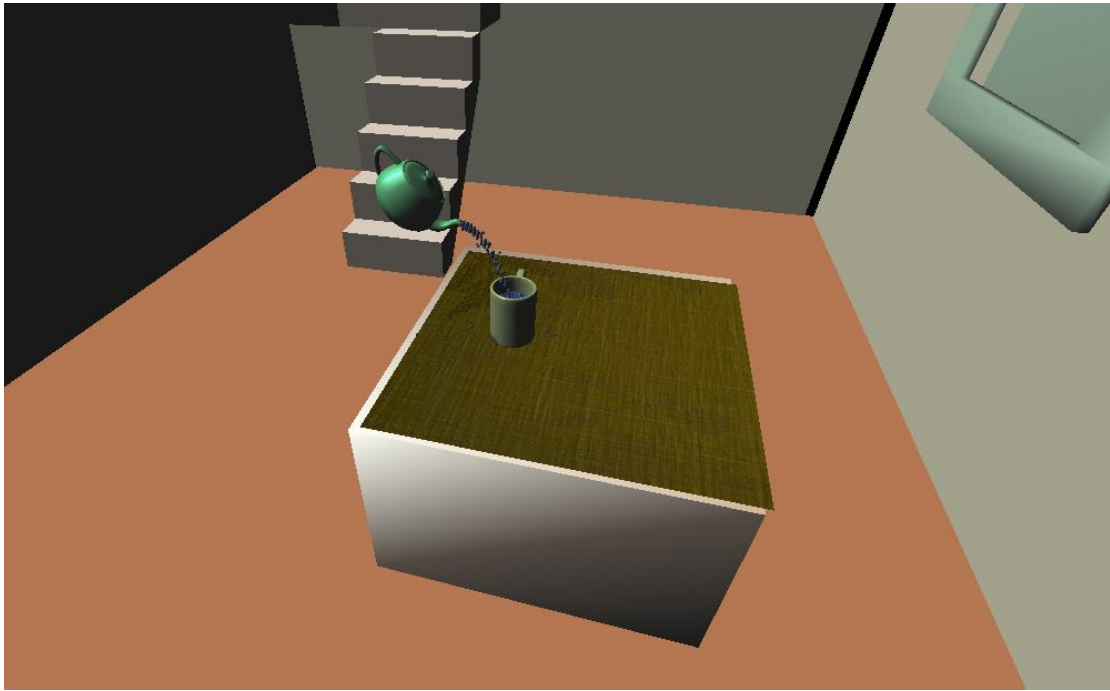
## 算法实现效果



流体物理性质的模拟



碰撞检测与水的积累



远处效果

## 程序内容说明

提交了两个.exe 文件，一个是 SPH\_FLUID.exe，它是在初期模拟流体性质用的。因为加了杯子以及场景变大，为了速度的关系粒子的量设了上限 8000 个，而在粒子比较散的情况下这个算法不能很好地模拟流体的性质，所以上传了一份之前的结果，这个程序在一个较小的空间中模拟了流体。

另一个文件为 TEA\_POT.exe，它是这次作业最后的成果，它模拟了从茶壶讲水倒入一个杯子的过程。

这两个程序都可以通过类似的操作方式来控制流体。

## 程序使用说明

两个程序操作的方式都相同

使用左键进行视角旋转

右键进行平移

‘w’ 和 ‘s’ 键进行视角远近的调整

空格键控制模拟的开始与暂停

‘o’ 键控制是否增加水流

‘[’ 与 ‘]’ 控制重力的方向，可以利用这两个键来观察流体的性质。