3D Fluid Dynamics Simulation of Different Smoke Compositions and Containers

ZHENGJIANG HE, University of California, Berkeley, USA
HIROTAKA ISHIHARA, University of California, Berkeley, USA
CHAOMIN LI, University of California, Berkeley, USA
ZHIQI YAN, University of California, Berkeley, USA

There are currently two common approaches to simulate fluids like smoke, a grid-based approach or a particle-based approach. In lecture, we discussed the grid-based approach in detail. However, the particle based smoke simulation is also a popular choice in many situations due to its simplicity compared to a grid based approach. In this paper, we present a particle-based smoke simulation application. We simulated each particle using Naiver-Stoke equation to ensure a more realistic simulation. We then used OpenGL to accelerate the rendering of particles.

ACM Reference Format:
Zhengjiang He, Hirotaka Ishihara, Chaomin Li, and Zhiqi Yan. 2023. 3D Fluid Dynamics Simulation of Different Smoke Compositions and Containers. 1, 1 (April 2023), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The project involves creating a 3D fluid dynamics simulation that models the behavior of an incompressible, homogeneous fluid under the Navier-Stokes equations. The simulation should include the ability to simulate smoke with different compositions and to model the effects of different containers made of various materials and shapes.

2 PROBLEM

We are going to develop a 3D fluid dynamics simulation that estimates the behavior of an incompressible, homogeneous fluid under the Navier-Stokes equations. The simulation should model the movement of smoke and how they interact with different objects, such as barriers / containers made of different materials and shapes.

Our simulation will also model the behavior of smoke with different compositions, which models the movement of particles within the smoke, and how they interact with each other and with the fluid. To simulate the effects of different materials and shapes of the barriers/containers, we will need to implement boundary conditions that represent the surfaces of the containers. These boundary conditions should be able to accurately represent the properties of the materials used to construct the containers and should be able to account for any effects that these materials may have on the fluid and smoke.

Authors' addresses: Zhengjiang He, University of California, Berkeley, USA, ss@berkeley.edu; Hirotaka Ishihara, University of California, Berkeley, USA, ss@berkeley.edu; Chaomin Li, University of California, Berkeley, USA, ss@berkeley.edu; Zhiqi Yan, University of California, Berkeley, USA, ss@berkeley.edu.

Manuscript submitted to ACM

3 IMPLEMENTED ALGORITHMS AND TECHNICAL APPROACHES - NEED TO EXPAND

To simulate the smoke in three-dimensional space, we implemented the particle-based smoke simulation.[1] In our case, we present a simplified implementation of the simulation using Smoothed Particle Hydrodynamics (SPH)[2] for solving the Navier-Stokes equations. Our approach is inspired by the technique proposed by Matthias Muller [4].

3.1 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) is a Lagrangian method that discretizes the fluid into a set of N particles ($i = 0, 1, ..., N-1$). Each particle carries fluid properties (in our case is smoke) such as position, velocity, density, viscosity etc. SPH uses smoothing kernel functions to approximate the continuous fluid properties based on neighboring particles. The technique we follow uses three different kernels, W_{poly6}, W_{spiky}, and $W_{viscocity}$.

The Poly6 kernel is defined as:

$$W_{poly6}(r,h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & 0 \le r \le h \\ 0, & r > h \end{cases} \tag{1}$$

The Spiky kernel is defined as:

$$W_{spiky}(r,h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3, & 0 \le r \le h \\ 0, & r > h \end{cases} \tag{2}$$

The Viscosity kernel is defined as:

$$W_{viscosity}(r,h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1, & 0 \le r \le h \\ 0, & r > h \end{cases} \tag{3}$$

In later experiment, we will try to implement more kernels from the existing solutions.

3.2 Discretization and Solution Strategy

The Navier-Stokes equation is discretized for each particle i as follows:

$$\frac{du_i}{dt} = P_i + V_i + G \tag{4}$$

where u_i is the velocity of the i-th smoothed particle, P_i represents the pressure forces, V_i denotes the viscosity forces, and G is the gravity forces acting on all particles. To solve this system of ordinary differential equations (ODEs), we follow a step-by-step approach to udpate the particles' positions.

3.3 Solving equation

The algorithm we implemented consists of the following steps:

(1) Compute the right-hand side (RHS) for each particle F_i:
   (a) Calculate the distances between all particle positions: $d_{ij} = ||x_i - x_j||_2$
   (b) Compute the density at each particle's position: $\rho_i = \frac{315M}{64\pi L^9} \sum_j (L^2 - d_{ij}^2)^3$
   (c) Calculate the pressure at each particle's position using κ, which indicates the resistance to fluid compression, and a base density ρ_0: $p_i = \kappa(\rho_i - \rho_0)$
   (d) Compute the pressure force of each particle: $P_i = -\frac{45M}{\pi L^6} \sum_j -\frac{x_j - x_i}{d_{ij}} \frac{p_j + p_i}{2\rho_j} (L - d_{ij})^2$

(e) Calculate the viscosity force of each particle: $V_i = \frac{45\mu M}{\pi L^6} \sum_j \frac{u_j - u_i}{\rho_j}(L - d_{ij})$
(f) Sum up the RHS: $F_i = P_i + V_i + G$

To clarify, L and M are the smooth-particle length and the mass of the particle respectively. We pre-set L to be 1.5 of the particle size, and the mass is set to 0.1 unit. After we compute the updated force F_i, the velocity of the particle i can also be updated as $v_i = v_i' + F_i \times \Delta t$, where v_i denotes the new velocity, v_i' denotes the velocity of the last time step, and Δt is the time step we choose to run the simulation.

4 SELECTED TECHNIQUES

To display the rendered frames of our smoke simulation system, our team used NanoGUI as the user interface. This is similar to what we did in HW4, where we utilized NanoGUI to display the results of a particle simulation. NanoGUI is a lightweight user interface library and provides a variety of widgets and tools that can be used to create user interfaces for displaying our simulation in OpenGL.

Eigen is versatile, fast and reliable C++ template library for linear algebra[3]. Since NanoGUI is already using Eigen as its linear algebra library, by sharing the same library, we simplified dependency of our project.

Dividing the simulation space into grids and parallelizing the inner-grid Navier-Stokes computations using OpenMP are two key techniques that our team is utilizing to improve the performance and accuracy of our system for rendering realistic smoke in 3D space. By breaking down the simulation into smaller, more manageable parts, we can optimize the computation process and speed up the simulation. Furthermore, these techniques are essential for ensuring that our system accurately reflects the real-world behavior of smoke particles and produces high-quality visual results.

5 CURRENT PROGRESS AND RESULTS

5.1 Current Progress

Our team is currently focused on developing a system for rendering realistic smoke in a 3D space. To accomplish this, we have implemented a particle system from scratch, which allows us to simulate the movement and behavior of individual smoke particles. Additionally, we have incorporated the Navier-Stokes equation into our system, which helps us to model the fluid dynamics of the smoke particles in a more accurate and realistic manner.

To create a user-friendly interface for our system, we have utilized OpenGL and NanoGUI to design and implement an intuitive UI that allows users to interact with and control the smoke simulation. One of the key challenges we faced was parallelizing the particle calculation process across all grids in the 3D space, which we have achieved by leveraging the power of OpenMP, a popular shared-memory parallelization API.

5.2 Next Steps

For our next steps, we will continue improving the realism of our smoke rendering system, delivering more convincing 3D smoke simulation in real time. In addition, we plan to introduce containers or barriers to the simulation, which will allow us to more accurately model the collision and interaction of smoke particles with physical objects. This will add an extra layer of complexity to the simulation, requiring us to develop advanced collision detection and response algorithms.

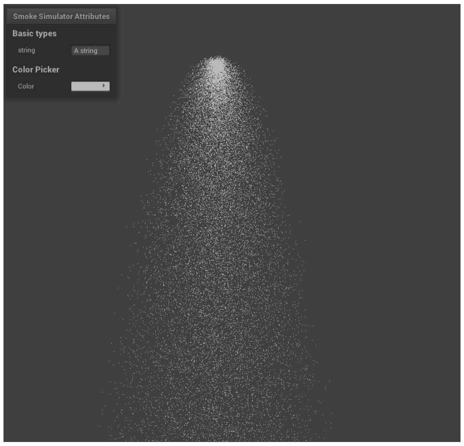

Fig. 1. Particle Sim w/o Buoyancy

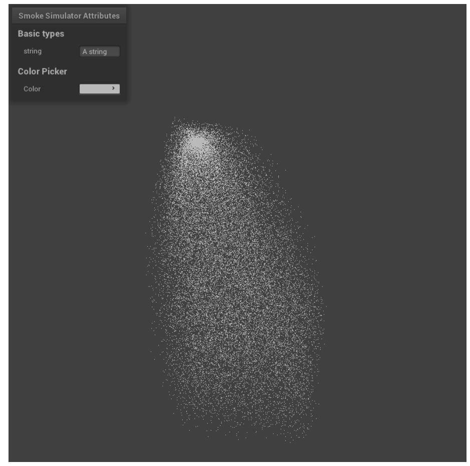

Fig. 2. View Angle Rotated Particle Sim

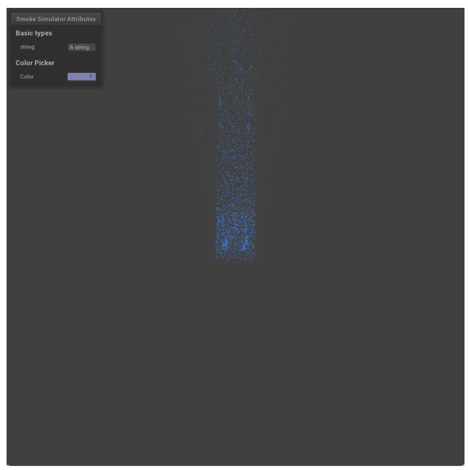

Fig. 3. Buoyancy force

REFERENCES

[1] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. 2001. Visual Simulation of Smoke. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 15–22. https://doi.org/10.1145/383259.383260

[2] R. A. Gingold and J. J. Monaghan. 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181, 3 (12 1977), 375–389. https://doi.org/10.1093/mnras/181.3.375 arXiv:https://academic.oup.com/mnras/article-pdf/181/3/375/3104055/mnras181-0375.pdf

[3] Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

[4] Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-Based Fluid Simulation for Interactive Applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, Goslar, DEU, 154–159.