

Smoothed Particle Hydrodynamics (SPH): A Versioned Literature Review Aligned to a C++ Implementation

Kallem George Thomas | 2304300@student.uwtsd.ac.uk |

1. Introduction

Smoothed Particle Hydrodynamics (SPH) is a mesh-free, particle-based fluid simulation method commonly used for real-time liquid effects and physically interactive environments. Unlike grid-based approaches, SPH represents fluids using particles that store properties such as density, pressure, and velocity. This project investigates SPH theory alongside the development of a C++ implementation focused on real-time simulation and optimisation for games applications.

Key Features:

- Particle-based fluid representation
- Real-time interactive simulation
- Suitable for dynamic liquid motion
- Adaptable for GPU acceleration

2. Aim and Objectives

The aim of this project was to investigate modern SPH simulation techniques and implement a functional real-time fluid simulation system in C++. The project combines theoretical research with practical implementation to evaluate both realism and performance.

Objectives:

- Research SPH fluid simulation methods
- Implement particle interaction systems
- Investigate optimisation techniques
- Develop interactive simulation controls
- Evaluate performance and realism

3. Research

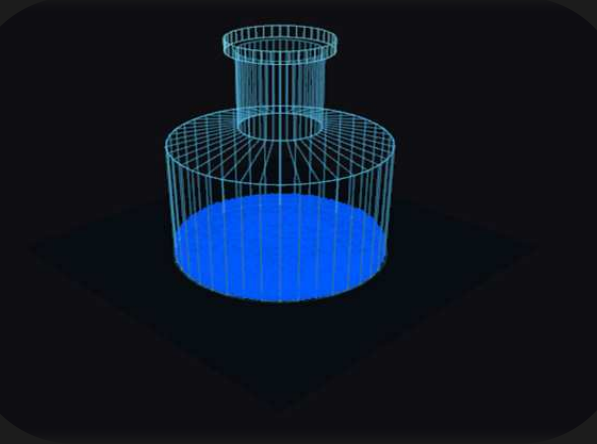
Research focused on the mathematical and computational foundations of fluid simulation. The project explored the Navier–Stokes equations, Lagrangian particle systems, and SPH kernel functions used to calculate pressure, density, and viscosity forces between particles. Modern real-time optimisation techniques such as spatial hashing and neighbour searching were also investigated.

Research Areas

- Eulerian vs Lagrangian methods
- Navier–Stokes fluid dynamics
- SPH kernel functions
- Spatial partitioning optimisation
- Real-time graphics applications

4. Implementation

The SPH system was developed using C++ within Visual Studio using Raylib and OpenCL for rendering and interaction. The implementation includes density and pressure calculations, viscosity forces, collision handling, and neighbour searching through a uniform spatial grid. Additional features such as XSPH smoothing and overlap correction were introduced to improve fluid stability and visual realism.



Implementation Features

- C++ particle simulation framework
- Uniform grid neighbour search
- Pressure and viscosity calculations
- XSPH smoothing implementation
- Real-time rendering and controls
- Performance optimisation systems

5. Conclusion

The project successfully demonstrated the implementation of a real-time SPH fluid simulation system capable of producing interactive liquid behaviour within a C++ environment. Through the research and implementation process, the project explored both the mathematical foundations of fluid simulation and the technical challenges involved in creating stable particle interactions in real time. The final system was capable of simulating pressure, density, viscosity, neighbour interactions, and boundary handling while maintaining interactive frame rates at moderate particle counts.

The research highlighted the advantages of SPH as a flexible, particle-based simulation method that is highly suitable for dynamic fluids and free-surface behaviour. However, the project also identified several limitations, particularly regarding computational expense as particle counts increase. Real-time performance became increasingly difficult to maintain when moving toward larger-scale simulations due to the cost of neighbour searching and force calculations.

