Real Time Fluid Simulation

using Smoothed-Particle Hydrodynamics and OpenGL

Computer Graphics CS 488

Matthias Untergassmair
munter2@uic.edu

Michael Berg
mberg4@uic.edu

ABSTRACT

This paper explores the field of Smooth Particle Hydrodynamics (SPH), starting at its beginnings as a tool to simulate astrophysical phenomena and following its evolution and implementation as a way to simulate fluids such as water. We will then give a brief mathematical background and ensuing algorithm of our SPH simulation followed by a detailed explanation of it.

Author Keywords

SPH, Smoothed Particle Hydrodynamics, OpenGL, Fluid Simulation, Real Time

Download code at

`https://github.com/munter2/RealTimeFluid`

INTRODUCTION

Smoothed Particle Hydrodynamics (SPH) successfully simulates fluids by breaking up a fluid body into individual parts, or particles. These particles together form a particle system that simulates various gravitational forces. Fluid movement is simulated in the system by moving particles around any particle moved, simulating a rippling, wave effect.

The ripple effect is created by first calculating which particles are surrounding a particle moved, and these surrounding particles are moved according to the movement of the first particle moved. But SPH wasn't originally intended to simulate liquid substances, but astrophysical phenomena.

HISTORY

Given all the different applications for Smoothed Particle Hydrodynamics (SPH), it was first used to simulate interstellar phenomena. Conceived in 1977 by Gingold and Monaghan was an improvement to the Standard Finite Difference Method, which until their breakthrough, was the method to use to simulate astrophysical phenomena. They improved on this method by making "use of Lagrangian description of fluid flow which automatically focuses attention on fluid elements" [2]. In this implementation, particles "move according to the Newtonian equations with forces due to the pressure gradient and other body forces: gravity, rotation and magnetic" [2].

The set of equations utilized in Gingold and Monaghan's SPH implementation is a density distribution calculated at each point

$$\rho_s(r) = \int W(r - r')\rho(r')dr' \tag{1}$$

where W is a function satisfying the condition

$$\int W(r)dr = 1 \tag{2}$$

This density distribution equation is used to calculate the density of the particles surrounding the current particle the equation it's being applied to. A Gaussian function is used for the kernel function, with the form

$$(\frac{1}{\pi h^2})^{\frac{3}{2}} \exp{(\frac{-r^2}{h^2})} \tag{3}$$

and h is

$$h = b(< r^2 > - < r >^2)^{\frac{1}{2}} \tag{4}$$

and *b* is adjustable. The gravitational potential of the particles is denoted by

$$\phi = -G \int \frac{\rho_N(r')}{|r - r'|} dr' \tag{5}$$

with G being the gravitational constant. Substituting in eq. 1 turns the equation of gravitational potential into

$$\phi = -\frac{GM}{N} \sum_{j=1}^{N} \int \frac{W(r - r')}{|r - r'|} dr' \tag{6}$$

To finish the equation, starting with

$$\nabla^2 I_j = -4\pi W(r - r_j) \tag{7}$$

substituting gravitational potential for I

$$\nabla\rho = -\frac{GM}{N} \sum_{j=1}^{N} \left\{ -\frac{4\pi}{u_j^2} \int_0^{u_j} W(u)u^2 du \right\} \nabla u_j \tag{8}$$

where

$$u_j = r - r_j \tag{9}$$

Adding in the Gaussian Function in eq. 3 for W yields

$$\nabla\phi = -\frac{GM}{N} \sum_{j=1}^{N} \frac{2}{u_j} \left(\frac{f}{\pi}\right)^{\frac{1}{2}} \left[\exp{(-fu_j^2)} - \frac{1}{u_j} \int_0^{u_j} \exp{(-fu^2)} du \right] \nabla \tag{10}$$

This equation is the full solution to calculating the gravitational potential of all particles involved in the simulation.

The result of Gingold and Monaghan's theory and implementation was a robust and extendable idea that could easily made more accurate "by increasing the number of particles and by using the devices known to improve Monte Carlo integration methods" [2].

SMOOTHED PARTICLE HYDRODYNAMICS OVERVIEW

MATHEMATICAL BACKGROUND

$$a_i^n = \frac{F_i^n}{m_i} = \dots \tag{11}$$

THE ALGORITHM

In the following, we denote the position for the particle i at time t as x_i^t, its velocity as v_i^t and its acceleration as a_i^t. We omit the vector notation ($\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{a}$) for these quantities, since the following equations are valid for the vectors as well as for each component individually.

As suggested in [3], we use the *Velocity-Verlet* time stepping scheme as follows:

Algorithm 1: Single Timestep with Velocity Verlet Algorithm

Data: x_i^t, $v_i^{t-\frac{\Delta t}{2}}$, a_i^t, Δt
Result: $x_i^{t+\Delta t}$, $v_i^{t+\frac{\Delta t}{2}}$, $a_i^{t+\Delta t}$
$v_i^{t+\frac{\Delta t}{2}} = v_i^{t-\frac{\Delta t}{2}} + \Delta t a_i^t$;
$x_i^{t+\Delta t} = x_i^t + \Delta t v_i^{t+\frac{\Delta t}{2}}$;
$a_i^{t+\Delta t} = a_i^{t+\Delta t}(x_i^{t+\Delta t}, m_i)$ from equation 11 ;

References

1. Akenine-Möller, T., Haines, E., and Hoffman, N. Real-time rendering 3rd edition. Natick, MA, USA: A. K. Peters, Ltd., 2008, 1045. ISBN: 987-1-56881-424-7.

2. R.A. Gingold, J. M. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.”*, (375-389), 1977:

3. S. Adami X. H., N. A. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, (231), 2012: