Workshop Series: Smoothed Particle Hydrodynamics (SPH)

Benjamin Kenwright[1]*

Abstract
This practical explains and demonstrates the valuable and important technique of Smoothed Particle Hydrodynamics (SPH). We aim to provide the basic knowledge on SPH for numerical applications, such as, virtual environments and video game terrains. The student after being presented with the basic equations and concepts should attempt to implement an uncomplicated real-time interactive fluid effect using SPH (i.e., in either 2D or 3D - particles moving under constraint forces based upon the Smoothed Particle Hydrodynamics principle).

Keywords
Fluid, Interactive, Real-Time, SPH, Smoothed Particle Hydrodynamics, Particles, Forces, Splashing, Waves

[1] *Workshop Series (www.xbdev.net) - Benjamin Kenwright*

Contents

Introduction

Particle Fluid Effects The topic of this practical is the implementation of a real-time particle fluid simulation effect (i.e., an interactive and controllable fluid that splashes and sloshes). The user should be able to interact with the simulation while it is running (e.g., through the mouse or keyboard) to control the fluid - such as, adding additional fluid and/or injecting forces and turbulence into the system. The fluid simulator should be implemented using component-based programming principles (i.e., flexible set of library functions, such as, vector and matrix classes).

Tasks

- Visually display a simple particle fluid simulation
- User input (e.g., mouse or keyboard) to control and move the fluid around (i.e., ability to add forces to interact with the fluid)
- Add additional fluid into the effect in real-time (e.g., tap water)
- Mix in multiple fluids with different properties (e.g., water and oil) and display them using different colours
- Experiment with different fluid Kernels to gain a solid understanding of limitations and visual outcomes

1. Overview

Principles and Concepts A robust and computationally efficient fluid simulation is indispensable in interactive real-time virtual environments, such as games. A realistic effect that is computationally efficient and straightforward brings an otherwise static and in inflexible scene to life.

Dynamic Fluid Effects We can expand the underpinnings of fluid mechanics to encapsulate the fundamental properties of fluid and gas by:

- simplifying the concept down to particles (i.e., compared to grid based methods)
- expand the working principle to represent various fluid types and viscosities

2. Smoothed Particle Hydrodynamics (SPH)

Fluids & Gasses Smoothed-particle hydrodynamics (SPH) is a computational method used for simulating fluid flows. It was initially developed for astrophysical problems but has been used in many fields of research, including astrophysics, ballistics, graphics, and video games. SPH is a mesh-free Lagrangian method (where the coordinates move with the fluid), and the resolution of the method can easily be adjusted with respect to variables such as the density.

SPH calculates an estimate for density, pressure, and viscous forces based upon a set of randomly distributed particles.

Figure 1. Basic Particle Fluid Effects - Pouring a fluid in real-time into a container (a)-(d) - as the container fills up we mix in a second fluid material.

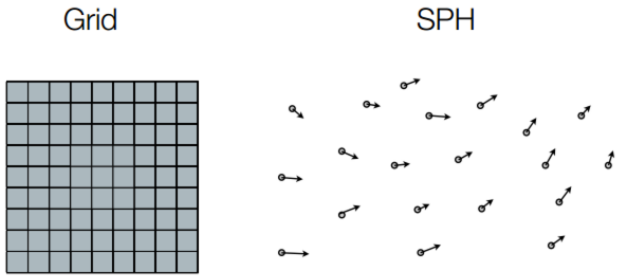

Figure 2. Grid vs SPH - Visually comparing a grid based method with smoothed particle hydrodynamic method (SPH).

3. Technical Details

Particle Density The density of a particle i (ρ_i) can be expressed as:

$$\rho_i = \sum_j m_j W_{ij} \tag{1}$$

where W_{ij} is the kernel function that weights the contribution of the sample positions r_j according to their distance r_i. For example:

$$\begin{aligned} q &= \frac{||r_i - r_j||}{h} \\ W_{ij} &= W(q) = W\left(\frac{||r_i - r_j||}{h}\right) \end{aligned} \tag{2}$$

where h is called the smoothing length (i.e., it is not necessarily the particle distance or the size of the compact support W_{ij}).

Particle Pressure The pressure for each particle is calculated from its density using Equation 3 below:

$$p_i = k\left(\left(\frac{\rho_i}{\rho_0}\right)^7 - 1\right) \tag{3}$$

where ρ_0 is the desired rest density of the fluid, k is a user-defined stiffness constant that scales pressure, pressure gradient, and the resulting pressure force.

for *all particle i* **do**
  find neighbour j
end
for *all particle i* **do**
  $\rho_i = \sum_j m_j W_{ij}$
  compute p_i from ρ_i
end
for *all particle i* **do**
  $a_i^{pressure} = -\frac{1}{\rho_i}\nabla p_i$
  $a_i^{viscosity} = \nu\nabla^2 v_i$
  $a_i(t) = a_i^{pressure} + a_i^{viscosity} + a_i^{other}$
end
for *all particle i* **do**
  $v_i(t+\Delta t) = v_i(t) + \Delta t\ a_i(t)$
  $r_i(t+\Delta t) = r_i(t) + \Delta t\ v_i(t+\Delta t)$
end

Algorithm 1: Simple SPH Fluid Algorithm

Example Kernel Function (Gaussian) An example of the smoothing interpolation kernel W is thus given by Equation 4 below (i.e., Gaussian Kernel):

$$\begin{aligned} W(r,h) &= \frac{1}{(\pi h^2)^{\pi/2}}\ exp\ (\frac{-r^2}{h^2}) \\ r &= r_i - r_j \end{aligned} \tag{4}$$

The Gaussian kernel function is adequately smooth even for higher order derivatives and provides stable and accurate results even for disordered particles distributions. Even if it is not compactly supported it approaches rapidly to zero and can be regarded as compact in numerical computations; however for higher order derivatives the radius of the support may increase significantly in order to assure the above requirement.

The Gaussian Kernel may be more natural but has the disadvantage that strictly speaking the region of influence of each SPH particle is infinite, even though the kernel may have very low values at large distances.

4. Summary

We have introduced an uncomplicated real-time particle fluid simulator for interactive dynamic scenes. Combining basic

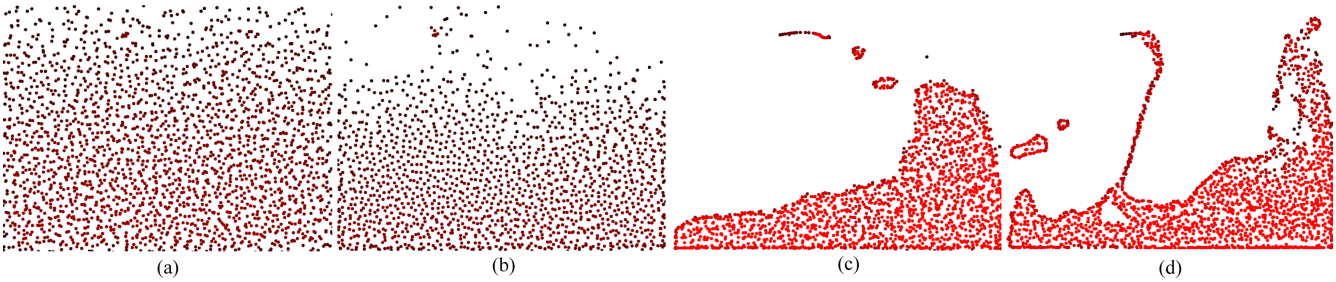

Figure 3. Component Features - (a) Pressure forces, (b) pressure forces with viscosity, (c) pressure forces with surface tension, and (d) pressure forces, viscosity, and surface tension. Simply enforcing particle pressure constraints under gravity does not synthesize the illusion of a fluid-like system. Adding viscosity is analogous to damping causing the fluid to be less chaotic. Combining the different components (i.e., surface tension, viscosity, and pressure forces) produces an aesthetically pleasing result.

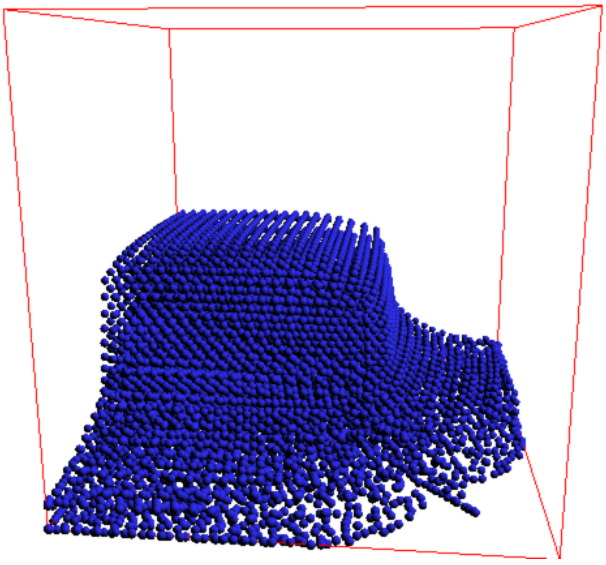

Figure 4. Uncomplicated Simulation Smoothed Particle Hydrodynamics (SPH) - Collapsing block of spheres.

particle and pressure principles to create a stable interactive effect suitable for real-time environments, such as games.

5. Exercises

This practical only gives a brief taste of fluid effects. As an exercise for the student to help enhance their understanding:

Intermediate

- Implement a scene with multiple fluid types (e.g., oil and water mixing)
- Have the fluid turn into a gas
- Create a more complex scene (e.g., beyond a simple cube)
- Drop rigid bodies into the scene and have the liquid react by splashing
- Implement an implicit surface, such as, marching cubes, to create a realistic visual fluid effect

Acknowledgements

We would like to thank all the students for taking time out of their busy schedules to provide valuable and constructive feedback to make this practical more concise, informative, and correct. However, we would be pleased to hear your views on the following:

- Is the practical clear to follow?
- Are the examples and tasks achievable?
- Do you understand the objects?
- Did we missed anything?
- Any surprises?

The practicals provide a basic introduction for getting started with physics-based animation effects. So if you can provide any advice, tips, or hints during from your own exploration of simulation development, that you think would be indispensable for a student's learning and understanding, please don't hesitate to contact us so that we can make amendments and incorporate them into future practicals.

Recommended Reading

Particle-based fluid simulation for interactive applications, Müller, Matthias and Charypar, David and Gross, Markus, Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, 154–159,2003, Eurographics Association

Weakly compressible SPH for free surface flows, Becker, Markus and Teschner, Matthias, Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, 209–217,2007, Eurographics Association

Game Inverse Kinematics: A Practical Introduction (2nd Edition) Kenwright. ISBN: 979-8670628204

Kinematics and Dynamics Paperback. Kenwright. ISBN: 978-1539595496

Game Collision Detection: A Practical Introduction (Paperback). Kenwright. ISBN: 978-1511964104

Game C++ Programming: A Practical Introduction (Paperback). Kenwright. ISBN: 978-1516838165

Computational Game Dynamics: Principles and Practice (Paperback). Kenwright. ISBN: 978-1501018398

Game Physics: A Practical Introduction (Paperback). Kenwright. ISBN: 978-1471033971

Game Animation Techniques: A Practical Introduction (Paperback). Kenwright. ISBN: 978-1523210688

Appendix

Radial Basis Kernels

A variety of functions can be used as the radial basis kernel $R(r)$. Some of the most common choices in the literature are:

Gaussian	$R(r) = exp(-(r/c)^2)$
Hardy Multiquadric	$R(r) = \sqrt{r^2 + c^2}$
Inverse Multiquadric	$R(r) = 1/\sqrt{r^2 + c^2}$
Thin Plate Spline	$R(r) = r^2 \, log \, r$ (in two dimensions)
Laplacian Splines (or Polyharmonic)	$R(r) \propto \begin{cases} r^{2s-n} \, log \, r & \text{if } 2s-n \text{ is an even integer} \\ r^{2s-n} & \text{otherwise} \end{cases}$

Radial Based Functions (RBS)

Radial basis functions are a popular choice for scattered interpolation. The interpolant is a linear combination of non-linear functions of distance from the data points:

$$\sum_i \sum_k w_i \, w_k \, R(||x_i - x_j||)$$