FEDERAL UNIVERSITY OF PERNAMBUCO
CENTER OF INFORMATICS
COMPUTER SCIENCE GRADUATION
2015.2

A Ray Traced Rendering Solution for Particle-Based Fluid Simulation

GRADUATION PROJECT PROPOSAL

Student: Caio José dos Santos Brito (cjsb@cin.ufpe.br)
Supervisor: Veronica Teichrieb (vt@cin.ufpe.br)
Co-Supervisor: Mozart William Santos Almeida (mwsa@cin.ufpe.br)

October 2nd, 2015.

1. Introduction

Fluid Simulation is used in films, games and virtual reality to generate plausible physical results and with high visual quality. In scenarios where high numerical accuracy is needed, such as movies or scientific simulations, the complexity of the simulation and rendering algorithms becomes very complex, therefore demanding offline implementations [1]. When only visual plausibility is needed, as in the cases of video games or some virtual reality applications, the use of a simpler method may be used in order to optimize the processing time, leading to lower numerical precision and visual of the results [2]

In this work, the Lagrangian method Smoothed Particle Hydrodynamics (SPH) will be used to simulate the fluid movement [3]. This method takes a continuous amount of fluid and describes it as a finite number of particles without the use of a mesh [4], which leads to a low memory cost and the system becomes more dynamic. For those reasons, it has been used in games [5] and movies as well [1].

The SPH method is able to simulate fluid with numerical precision, but it consumes a long processing time because a large number of particles that is required to achieve a higher order precision. This performance issue can be minimized by using a parallel solution of the algorithm [6].

The SPH can be resumed in three steps: neighborhood search, the calculation of each particle's acceleration and the time integration. The neighborhood search is a potentially time-consuming step and is usually optimized using an accelerated spatial access structure like a uniform grid [2] or an octree [7], instead of a naïve brute-force search. The considered forces applied in the particle system are gravity, pressure and viscosity [3]. The pressure influence of each particle can be calculated using a state equation [4] or solving the Pressure Poisson equation [8]. The first option is usually chosen for the simplicity and computational efficiency and has a good numerical precision while, in order to calculate the Pressure Poison Equation one needs to solve a sparse linear system, which can make the simulation slower

To calculate the viscosity in the particle position, an artificial viscosity can be used which can introduce shear and bulk viscosity in the simulation [3]. Another option is to use a laminar viscosity [8] in a particle system that do not have the presence of turbulence or use a function that damp the velocity of a particle using the velocity from the neighborhood [9]. This option is easier to tune and can achieve acceptable numerical accuracy for our purposes [10].

The time integration can be done using a local [8] or global [3] scheme and, depending of the method implemented, one scheme can be more appropriated than another. In our work, we chose the global approach.

In order to render the results, it is required to reconstruct the fluid surface using the particles identified as free surface [11]. To visualize the simulation, it is possible to use many methods such as Direct Rendering [12], 3D Scalar Fields [11], Volume Rendering [13] or the Screen Space Approach [14]. Recently, the literature shows that there are two main approaches for rendering the simulation results:

(1) **3D Scalar fields**. This method maps each particle to a scalar value and reconstructs the surface using those values, for instance, using a marching cubes algorithm;

(2) **Screen Space approach**. This approach renders the particles as point sprites and apply a smoothing filter in the surface;

The first option has as major challenge: choosing the right kernel function to determine the scalar density field of the surface particles. To create a smooth surface,

the function is calculated using the neighbors of the particles, which tends to be time consuming, and making the rendering method more appropriated for offline applications [11].

While the second option can be better suited for a real time application, because it renders the surface particles individually without the necessity of applying a function over the neighbors, after the surface is reconstructed, the results may have a jelly look. To overcome this characteristic, a smooth function is applied in the particles normal. To find the best function to create a smooth surface is the major challenge of this kind of rendering technique [15].

2. Objectives

This graduation work aims to study rendering algorithms applied to the Weakly Compressible Smoothed Particle Hydrodynamics. First, an implementation of the method will be improved to simulate a higher number of particles, by adding an accelerated spatial access structure. This version will be tested with different test cases.

The results from the simulation will be rendered using a ray-traced method based on the works of [14] and [15].

The goals to be achieved in this work are:

- Study and implementation of the improvements at the SPH method.
- Study the rendering techniques used in particle based simulation.
- Choose the most appropriated technique for rendering the simulation results using a ray-tracer [16] as core of the rendering algorithm
- Compare the rendering results with the ones found in the literature

3. Methodology

The graduation work will be accomplished in three stages: research, development and validation.

At the research stage, two aspects will be studied:

(1) The SPH method for fluid simulation;

(2) The rendering methods applied to this kind of simulation.

At the end of this stage, a rendering approach will be chosen to apply in a ray tracer algorithm.

The development stage will be divided in two steps. First, using our SPH implementation already developed, a GPU version will be implemented to optimize the result computation, by not only creating a parallel solution for the algorithm but also adding a spatial hash structure. Then, using the SPH results, a rendering solution will be developed using the ray tracer as rendering platform [16] by adding new rendering modules based on the approach considered more suitable to create an interactive solution with a high visual quality.

The GPU solution will be implemented using NVIDIA's CUDA technology [17], the development environment will be the Microsoft Visual Studio 2013 and using the C/C++ language.

In the validation step, the results will be compared with the ones found in the literature considering both the visual quality and execution time and, by the end of the project, there will be a rendering solution applied to particle-based simulations using a ray tracer platform.

4. Schedule

Activities	August/2015				September/2015				October/2015				November/2015				December/2015				January/2016			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Literature Review and Solution Definition																								
Development of the SPH method in GPU (CUDA)																								
Development of the Rendering Solution																								
Compare the Solution Results with the Literature																								
Write Final Report																								
Make Presentation																								

5. Reference

[1] C. J. Horvarth; B. Solenthaler. Mass preserving multi-scale SPH. **Pixar Technical Memo 13-04, Pixar Animation Studios**, 2013.

[2] M. Ihmsen et al. SPH fluids in computer graphics. 2014.

[3] J. J. Monaghan. Simulating free surface flows with SPH. **Journal of computational physics**, v. 110, n. 2, p. 399-406, 1994.

[4] P. J. Morris; P. J. Fox; Y. Zhu. Modeling low Reynolds number incompressible flows using SPH. **Journal of computational physics**, v. 136, n. 1, p. 214-226, 1997.

[5] S. Green. Screen space fluid rendering for games. In: **Proceedings for the Game Developers Conference**. 2010.

[6] P. Goswami et al. Interactive SPH simulation and rendering on the GPU. In: **Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation**. Eurographics Association, 2010. p. 55-64.

[7] F. Reichl; M. Treib; R. Westermann. Visualization of big SPH simulations via compressed octree grids. In: **Big Data, 2013 IEEE International Conference on**. IEEE, 2013. p. 71-78.

[8] R. Xu; P. Stansby; D. Laurence. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. **Journal of Computational Physics**, v. 228, n. 18, p. 6703-6725, 2009.

[9] H. Schechter; R. Bridson. Ghost SPH for animating water.**ACM Transactions on Graphics (TOG)**, v. 31, n. 4, p. 61, 2012.

[10] A. L. V. E Silva et al. A QUALITATIVE ANALYSIS OF FLUID SIMULATION USING A SPH VARIATION. 2015.

[11] J. Yu; G. Turk. Reconstructing surfaces of particle-based fluids using anisotropic kernels. **ACM Transactions on Graphics (TOG)**, v. 32, n. 1, p. 5, 2013.

[12] J. Yu et al. Explicit mesh surfaces for particle based fluids. In:**Computer Graphics Forum**. Blackwell Publishing Ltd, 2012. p. 815-824.

[13] R. Fraedrich; S. Auer; R. Westermann. Efficient high-quality volume rendering of SPH data. **Visualization and Computer Graphics, IEEE Transactions on**, v. 16, n. 6, p. 1533-1540, 2010.

[14] W. J. Van Der Laan; S. Green; M. Ssinz. Screen space fluid rendering with curvature flow. In: **Proceedings of the 2009 symposium on Interactive 3D graphics and games**. ACM, 2009. p. 91-98.

[15] F. Reichl et al. Interactive Rendering of Giga-Particle Fluid Simulations. In: **Eurographics/ACM SIGGRAPH Symposium on High Performance Graphics**. The Eurographics Association, 2014. p. 105-116.

[16] A. Lira Dos Santos; V. Teichrieb Orientador. Estruturas de aceleração para Ray Tracing em tempo real: um estudo comparativo. 2011.

[17] NVIDIA, C. U. D. A. Compute unified device architecture programming guide. 2007.

6. Signatures

_____

Caio José dos Santos Brito
Student

_____

Veronica Teichrieb
Supervisor

_____

Mozart William Santos Almeida
Co-Supervisor