REAL-TIME PARTICLE SIMULATION OF FLUIDS

Zsolt Horváth
Master Degree Programme (2), FIT BUT
E-mail: xhorva03@stud.fit.vutbr.cz

Supervised by: Adam Herout
E-mail: herout@fit.vutbr.cz

Abstract: Physically plausible simulation of fluids in real-time is mostly achieved using approximations of the Navier-Stokes equations. Recent methods simulate fluids by exploiting the capabilities of modern graphics processing units. This article describes a method called Smoothed Particle Hydrodynamics (SPH), which is a numerical approximation of the Navier-Stokes equations. The real-time simulation allows for interactivity which is a great advantage against the offline methods. The main goal of this project was to experiment with the Smoothed Particle Hydrodynamics running real-time on the GPU.

Keywords: particle systems, fluid simulations, Navier-Stokes equations, smoothed particle hydrodynamics, CUDA, GPGPU, marching cubes

1 INTRODUCTION

Real-time simulation of fluids is a hot topic and a major challenge in computer graphics. Fluids are commonly said to be the hardest phenomena to simulate realistically it is harder to simulate detailed fluids interactively. The most common method used to simulate fluids is the Smoothed Particles Hydrodynamics (SPH) [4]. The SPH is based on the Navier–Stokes equations. Because of the complexity of these equations they can be solved analitically only in simple cases. In general, they are solved by a numerial method.

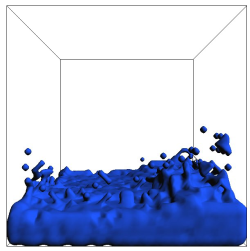

Figure 1: Fluid rendered with the marching cubes method.

The aim of this work was to experiment with the SPH method on the CUDA platform. The experiments are mostly focused on achieving the simulation running in real time with different types of visualization methods and on speeding up the implemented algorithms.

2 SMOOTHED PARTICLE HYDRODYNAMICS

The Smoothed Particle Hydrodynamics method was introduced in 1977 by Monaghan and Gingold [2]. The particle–based methods in the literature are called Lagrangian models [3]. These methods represent the fluids as a discrete set of particles. These particles simulate the flow of the fluid by solving the particle hydrodynamics.

The SPH is an interpolation method for the particle systems. In this method, the field quantities are only defined at discrete particle locations, but can be evaluated anywhere in the space. For this purpose, the SPH distributes property (pressure, density, etc.) in the neighbourhood of any particle using the smoothing kernels [3, 1].

3 IMPLEMENTATION

This section is focused on the description of the simulation and visualization.

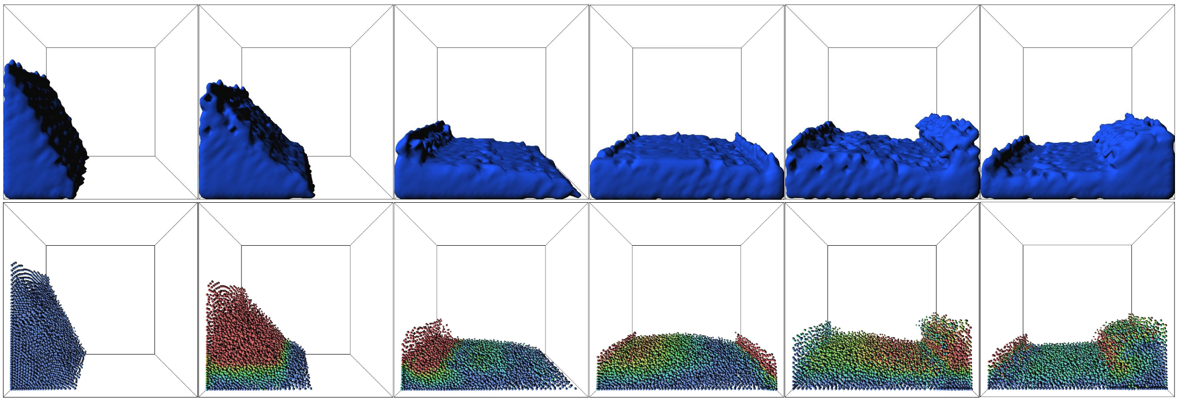

Figure 2: Fluid rendered with the marching cubes (upper) and the point sprites (lower) method.

3.1 Simulation

The computation of the fluid flow is divided into three steps: the density and pressure evaluations, the pressure and viscosity force computations, the velocity and position integrations. The velocity is computed by integrating the sum of the internal and external forces. The internal forces consist of the pressure and viscosity forces; the external forces are computed form the particle–boundary interactions.

The simulation uses a uniform grid which was inspired by the work of Green [6]. The space is divided into uniformly sized cells. Every particle is assigned to a cell according to its center position. This method helps to find the adjacent particles while evaluating the field quantities with the smoothing kernels. This approach is faster, than testing the distances between each particle and ignoring the distant ones.

3.2 Visualization

For rendering the simulation results, two methods were used: the marching cubes and point sprites rendering. The point sprites method is computed in the GPU's fragment shader. The marching cubes [5] method is an algorithm for rendering isosurfaces of volumetric data. The algorithm is sped-up by using the lookup tables. This method uses the GPU to track and extract the surface of the fluid. The additional normal interpolation and tessellation of the vertices is done for rendering visually appealing results.

4 RESULTS

By using the GPU, the SPH method, and the marching cubes method the goal of this work was achieved. The simulation is running in real-time with 60000 particles when no rendering method is used. When using lower grid resolutions there is a performace drop, compared to higher resolutions. This is caused by the high density of the particles in the cells, searching for the adjacent particles takes longer time. The benchmark results of the implemented SPH method and the rendering methods are summarized in Table 1.

At higher grid resolutions, there is much higher performace drop when the marching cubes and/or the tessellation is enabled. This is because the simulation grid. It is used for rendering and sampling the volume by the implemetation of the marching cubes. This solution reduces the required density computations and saves time.

The testing was done on NVidia GeForce 540M graphics card on Linux platform.

	Number of particles: 15000		
Grid res.	30x30x30	50x50x50	80x80x80
SIM	6	5	4
MC	2	5	12
NI	8	11	15
Sum	16	21	31

	Number of particles: 30000		
Grid res.	30x30x30	50x50x50	80x80x80
SIM	19	13	9
MC	2	5	12
NI	9	12	18
Sum	30	30	39

	Number of particles: 45000		
Grid res.	30x30x30	50x50x50	80x80x80
SIM	31	22	18
MC	3	5	13
NI	9	12	20
Sum	43	39	51

	Number of particles: 60000		
Grid res.	30x30x30	50x50x50	80x80x80
SIM	45	32	24
MC	3	5	13
NI	10	13	30
Sum	58	60	67

Table 1: The benchmark shows the results measured in milliseconds at different grid resolutions. *SIM* is the simulation, *MC* is the Marching cubes rendering, and *NI* stands for the normal interpolation. In each table, the upper row contains the number of particles used in the benchmark.

5 CONCLUSION

Many experiments were made in this work to speed up the SPH and rendering methods. The uniform grid allows for fast access to adjacent particles. In the future, further experiments can be made to speed up the searching and the sorting of the particles. To improve visual details of the implemented SPH method, the surface tension computation can be added, too. This additional computation has influence on the surface details of the fluid and makes it more realistic. Other experiments can be made to improve the existing algorithms of rendering, e.g. extract more accurate approximation of the surface with marching cubes or implement the raycasting method with depth peeling, which provides much more plausible results.

REFERENCES

[1] J. Hongbin, D. Xin. On criterions for smoothed particle hydrodynamics kernels in stable field. *Journal of Computational Physics*, 202:699–709, 2005.

[2] J. J. Monaghan, R. A. Gingold. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.

[3] M. Kelager. *Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics [online]*. PhD thesis, University of Copenhagen, Department of Computer Science, 2006.

[4] M. Müeller, D. Charypar, M. H. Gross. *Particle-Based Fluid Simulation for Interactive Applications*. SCA '03 Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, 2003. ISBN 1-58113-659-5.

[5] P. Bourke. Polygonising a scalar field [online]. `http://paulbourke.net/geometry/polygonise/`, 1994.

[6] S. Green. Particle Simulation using CUDA [online]. NVidia Corporation, `http://developer.download.nvidia.com/compute/DevZone/C/html/C/src/particles/doc/particles.pdf`, 2010.