High-detailed fluid simulations on the GPU

Mattias Lagergren* Fido Film
Fredrik Limsäter† Fido Film
Björn Rydahl‡ Fido Film

Introduction

During the last few years, Fido has seen an increasing demand for highly detailed water effects in production. Fluid simulations, aside from being complex and hard to choreograph, put high demands on scalability and simulation speed. As of today, commercial software is slow and does not allow the artist to get quick feedback on simulation results. Following the increasing use of general purpose GPU programming within the special effects industry, Fido wanted to enable artists to create highly detailed and realistic water effects, while maintaining production effectivity and avoiding investments of large server clusters.

This talk will describe the method of simulation and why it is suitable for high-detailed fluid simulations. We will discuss how to make it run simultaneously on modern graphics hardware and the major impact this kind of system has on the way artists work with simulated water effects.

Simulating fine-detail water effects

In order to perform physically based simulations of complex, high-detail water effects, we have developed a toolkit based on the Lagrangian method of smoothed particle hydrodynamics (SPH). SPH is commonly used for fluid simulations and allows for fine detail effects, which make it particularly suited for splashes and other free surface phenomena. When using SPH, the fluid body is represented by a cloud of particles. Through the use of smooth sampling kernels, the particles' own arbitrary properties (such as internal forces) can be calculated. SPH was first introduced as a viable fluid simulation method for the graphics community by [1], and further enhanced by [2] with the weakly compressible SPH.

GPU

The data parallelism inherent to the SPH method makes it especially suited to be solved on multi-core systems such as modern GPUs. To fully exploit the computational power of the GPU we parallelized the entire fluid solver, encompassing collisions, volumetric forces and sinks. Being an interpolation method which relies heavily on identifying the closed neighboring particles, advanced data structures and search algorithms are required. This is achieved by hashing the particles to a grid, sorting them by their hash and then using the hash to find particles in neighboring cells. To perform this sorting we utilize the radix sorting method as developed by [3]. Collisions and volumetric force fields utilize cached texture memory, making these features computationaly cheap.

Directability

Moving computations from the CPU to the GPU gives an enormous increase in speed and a tremendous boost in terms of artist interaction. By gaining the ability to simulate millions of particles in just a few seconds per frame we allow artists to get very quick feedback on what they are doing. The reduced time spent in simulations enables them to have quick turnarounds in order to get the desired result in a faster time and at a higher quality. This way of working with fluid simulations is something we have found virtually impossible with commercial software, which takes many hours or even days to complete large simulations.

Particles	Houdini	Our system	Speedup
50.000	30.6s	0.44s	70x
125.000	113.6s	1.1s	103x
500.000	389.4s	3.0s	130x

Table 1. Our system compared to the built-in Houdini solver, messured in seconds per simulated frame. Spec. Dual Quad Core Xeon 5450, Geforce GTX 470, 8gb of RAM.

We have integrated our simulation software into Houdini, making for a very powerful combination. Artists can easily use all the built in Houdini tools to specify particle emitters, collision objects, sinks and volumetric force fields, which are then solved by the GPU. The resulting particle simulation can then be scrubbed through or converted to a flipbook to get realtime feedback. The individual output particles can also carry a wide range of metadata describing for instance density, curvature and collision data.

Besides being fast and enabling us to increase the detail of our simulations, it remains extremely cost effective. Our simulation software will run on any readily available commercial CUDA-enabled graphics hardware, which allows us to be very flexible when undertaking projects involving water effects. Artists' computers can easily be upgraded, avoiding heavy investments of large server clusters. For Fido, a company working mostly with TVCs, this is an enormous advantage when it comes to the use of simulated fluid effects.

Figure 1. Large water splash, SB12 commercial.

References

[1] Müller, M., Charypar, D., and Gross, M 2003. *"Particle-based fluid simulation for interactive applications",* In SCA 03: Proceedings of the 2003 ACM SIG- GRAPH/Eurographics symposium on Computer animation, pages 154159. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2003.

[2] Becker, M. and Teschner, M. 2007. *"Weakly compressible SPH for free surface flows.",* SCA 07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 209217. 2007.

[3] Satish, N., Harris, M., and Garland, M. 2009. *"Designing Efficient Sorting Algorithms for Manycore GPUs",* In Proceedings of IEEE International Parallel & Distributed Processing Symposium 2009 (IPDPS 2009).

*e-mail: mattias.lagergren@fido.se
†e-mail: fredrik.limsater@fido.se
‡e-mail: bjorn.rydahl@fido.se

SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007