Analyzing Performance and Efficiency of Smoothed Particle Hydrodynamics

Rama C. Hoetzlein
Media Arts & Technology Program
University of California, Santa Barbara
Santa Barbara, CA 93106-5110
rch@umail.ucsb.edu

Tobias Höllerer
Department of Computer Science
University of California Santa Barbara
Santa Barbara, CA 93106-5110
holl@cs.ucsb.edu

ABSTRACT

With the rise in performance of modern GPUs, Smoothed Particle Hydrodynamics is an increasingly attractive solution for real-time simulation of fluid flows in visual effects for film and games. Starting with simulations of 2000 particles at 20 frames per second in 2003 [4], smoothed particle hydrodynamics has now been simulated with over 242,000 particles at 4 fps using GPUs [6]. While performance has clearly increased, the terms *performance* and *efficiency* are often used interchangeably. Results of simulations, as published in graphics journals such as SIGGRAPH, are typically reported by giving the number of particles and frame rate for a particular combination of CPU and graphics hardware. This makes comparisons of algorithm implementations difficult since authors must deduce algorithm efficiency for differing hardware. The development of concrete metrics of performance and efficiency will facilitate better comparison between results. Simple metrics are presented here with an analysis of real-time simulations over the past five years.

Categories and Subject Descriptors

I.3.5.i [**Object Modeling**]: Physically based modeling; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms

Smoothed particle hydrodynamics, Simulation, GPU

1. INTRODUCTION

We present a simple metric for evaluating performance and efficiency of real-time particle-based simulations. These metrics suggest new areas for increasing efficiency of smoothed particle hydrodynamics, which are incorporated into our fluid simulator, FLUIDS v.1. In addition, we use these metrics to deduce trends in algorithm implementation over time. Our initial findings show that while performance has greatly increased due to new GPU hardware, there has been a gradual decline in algorithm efficiency over time.

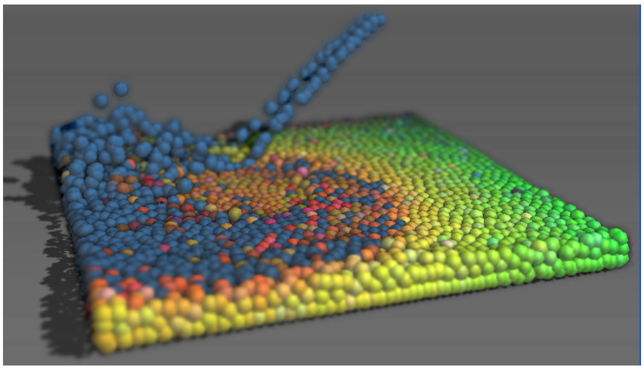

Figure 1: FLUIDS v.1 simulating 3000 particles at 45 fps with shadow maps and depth of field. Paint mixing is simulated with colored particles.

2. METHODS

To standardize simulation results we create simple metrics for performance and efficiency. First, we combine frames per second and number of particles to measure raw performance as the number of particles simulated per millisecond.

$$\mathbf{P}_{raw} = \#particles * fps * (1/1000)$$

Second, we estimate algorithm efficiency as the performance achieved on normalized hardware by dividing raw performance by hardware performance as measured in gigaflops.

$$\mathbf{E} = \mathbf{P}_{raw}/\mathbf{P}_{hardware}$$

The units of E are number of particles simulated in 1 ms on 1 gigaflop of hardware. Table 1 shows the author, year, performance and efficiency of several key real-time smoothed particle hydrodynamics papers from 2003 to 2008. All methods compared use a spatial grid to give $O(kn)$ behavior. Raw simulation rates are given without rendering. While the typical spatial grid technique used in SPH scales linearly, we also observed efficiency differences based on number of particles. In the table, the highest performance measure provided by the author in reported.

Performance estimates for CPU versus GPU hardware are more difficult as the GPU implementations introduce parallelism and memory transfer overhead. Raw results are reported, so these factors will be observed in our efficiency measures. In the future we hope to quantify and eliminate this overhead from our metrics. NVIDIA's own demo achieves the highest GPU efficiency (5.69), similar to Harada

Table 1: **Performance and efficiency of real-time smoothed particle hydrodynamics simulations.**

Author	Year	# Particles	FPS	Performance	Hardware	Gflops	Efficiency
Muller[4]	2003	2200	20	44.00	P4 1.8 ghz	3.6	12.22
Amada[1]	2003	2000	30	60.00	P4 2.8 ghz	5.6	10.71
Kontar[3]	2004	2000	18	36.00	XP 2200	4.4	8.18
Horvath[2]	2007	30000	0.50	15.00	P4 3.0 ghz	6.0	2.50
Harada (CPU)[6]	2007	262144	0.15	39.32	X6800 2.93	5.86	6.71
Harada (GPU)	2007	262144	4.23	1108.86	8800GTX	345.6	3.21
Harada[7]	2007	49153	17	835.60	8800GTX	345.6	2.42
Zhang[8]	2007	60000	15	900.00	8800GTX	345.6	2.60
NVIDIA[5]	2008	32768	60	1966.08	8800GTX	345.6	5.69
FLUIDS v.1 (CPU)	2008	3000	45	135.00	XP64 3.2 ghz	6.4	21.09

for the CPU (6.71), but still well below Muller's original paper (12.22) which implements a cache-coherent algorithm to optimize for the CPU. This suggests that both industry and academic implementations have not yet been fully optimized for the GPU.

The historical trends are also interesting. While overall performance has jumped by 20x due to new hardware, algorithm efficiency appears to have gradually declined even for the same hardware. While providing generous estimates for GPU parallelism overhead, implementations still appear to be less efficient than earlier authors. This supports our view that standardized measures of performance and efficiency are needed for publications in this area.

3. IMPLEMENTATION

A study of algorithm efficiency has suggested specific areas for improvement. Several of these improvements are incorporated into FLUIDS v.1, a simple, fast, open source implementation of Smoothed Particle Hydrodynamics. While currently running on the CPU only, FLUIDS v.1 is shown to be 3x more efficient than Harada [6], suggesting our GPU version should support 200,000 particles at 30 fps.

Smoothed Particle Hydrodynamics is a Lagrangian solution to the Navier-Stokes fluid equations. Simulation consists of three basic steps: 1) Computing density and pressure of all particles. 2) Computing forces on particles, 3) Advancing the simulation by integration. The essence of the SPH technique is that density, pressure, and force are computed by considering weighted contributions from neighboring particles.

A purely naïve implementation of density in step #1 requires $O(n^2)$ calculations. By inserting particles into a grid a particle need only check neighboring grid cells for contributing particles within a given radius, called the smoothing radius r. We notice that following Muller, all publications we studied use a grid size equal to r, requiring that each particle check 27 grid cells (3x3x3). However, if a grid size of $2r$ is used, then only 8 grid cells (2x2x2) must be checked.

Other efficiency gains in the SPH algorithm were found through basic programming: eliminating variables, manually in-lining vector algebra. Taking a pure programming perspective, the SPH steps are essentially doubly-nested loops. Thus we improve efficiency in FLUIDS v.1 by reducing the inner loop for force computation to just 11 lines of C.

4. CONCLUSIONS

It is well known that choice of algorithms has the most significant effect on scalability. However, for a given algorithm the details of implementation can still influence efficiency by orders of magnitude. This constant factor is often a make-or-break effect in real-time applications. More importantly, in the development of real-time fluid simulation, poor metrics for reporting academic results have possibly masked a steady decline in algorithm efficiency behind rapid advances in hardware performance. This situation may be improved in the future by making clear distinctions between performance and efficiency. FLUIDS v.1 demonstrates a stable, fast, open source implementation of Smoothed Particle Hydrodynamics with a measured efficiency at least twice that of other implementations. Current goals include optimization and measurement of a GPU version of FLUIDS.

5. REFERENCES

[1] T. Amada. Real-time animation of water. Retrieved: http://chihara.aist-nara.ac.jp/people/2003/takasi-a, 2003.

[2] P. Horvath and D. Illes. Sph-based fluid simulation for special effects. In *Central European Seminar on Computer Graphics. Conference, April 2007.*, 2007.

[3] S. Kontar. Real-time fluid simulation. Master's thesis, Brno Univ. of Technology, Brno, Czech Republic, 2004.

[4] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH symposium on Computer animation*, Aire-la-Ville, Switzerland, 2003. Eurographics Association.

[5] NVIDIA. Particle demo from CUDA SDK 10, 2008.

[6] S. K. T. Harada and Y. Kawaguchi. Smoothed particle hydrodynamics on GPUs. In *The Visual Computer*, Petropolis, Brazil, 2007. Springer. From proceedings of Computer Graphics International.

[7] S. K. T. Harada, M. Tanaka and Y. Kawaguchi. Real-time particle-based simulation on GPUs. In *ACM SIGGRAPH 2007 posters*, page 52, New York, NY, USA, 2007. ACM Press.

[8] Y. Zhang, B. Solenthaler, and R. Pajarola. GPU accelerated SPH particle simulation and rendering. In *ACM SIGGRAPH 2007 posters*, page 9, New York, NY, USA, 2007. ACM Press.