Literature Review

Andrew Pregent

Smoothed Particle Hydrodynamics (SPH) is a Lagrangian simulation method first proposed independently by L.B. Lucy and R.A. Gingold and J.J. Monaghan for simulations in astrophysics[4][12]. J.J. Monaghan later extended the method to free surface flows[13]. Refer to Ting Ye, et al. for a modern review of various advances in the method.

Mathieu Desbrun and Marie-Paule Gascuel applied the Courant-Friedrichs-Lewy criterion to SPH, providing an upper bound on the time step based on the kernel support size and the maximum particle velocity[3]. This means that in practice the time step must often be very small in order for the simulation to remain stable. Predictive Corrective Incompressive SPH (PCISPH) attempts to address this problem for incompressible fluids such as water, where the problem is exacerbated by the high stiffness required in the equation of state (EOS)[16].

Another approach is to avoid a global time-step altogether. Prashant Goswami and Christopher Batty propose segmenting the time-step by spatial chunks.[5]. Asychronous SPH allows every particle to have its own time frame[14][2][3]. This is more efficient when there are only a few fast particles, as is often the case. Reinhardt also suggest using multiple queues in parallel, a method due to Kale and Lew, though they are unsure if this will scale well in practice on the GPU.[14][11].

Much research has been done to bring SPH to the GPU. Much of the difficulty lies in efficiently searching a fixed distance neighborhood of each particle, since a brute force search of every particle pair is infeasible. The work of Ihmsen et al. provided much of the groundwork with an early parallel implementation.[10] For this search they used a Z-index sort, a space filling curve which provides a cache-friendly ordering for the particles. Amada et al. present a partial GPU implementation which relies on the CPU for the neighborhood search, providing the information to the GPU as a texture.[1] Harada et al. present an early fully GPU implementation[7]. Later Hérault make use of the programmable pipeline to create a CUDA implementation[9], which they later released as open source[8]. Finally, Rustico et al. extend this to multiple GPUs.[15]

The neighborhood search has also garnered interest on its own. Ohno, Nitta and Nakai look at optimizing the neighborhood search for the GPU. Recently, Groß and Köster look at utilizing modern features of GPUs[6]. In particular, they leverage atomic memory synchronization between work groups in order to optimize the search further.

References

[1] Takashi Amada et al. "Particle-based fluid simulation on GPU". In: *ACM workshop on general-purpose computing on graphics processors*. Vol. 41. Citeseer. 2004, p. 42.

[2] Xiaojuan Ban et al. "Adaptively Stepped SPH for Fluid Animation Based on Asynchronous Time Integration". In: *Neural Comput. Appl.* 29.1 (Jan. 2018), pp. 33–42. ISSN: 0941-0643. DOI: `10.1007/s00521-016-2286-8`. URL: `https://doi.org/10.1007/s00521-016-2286-8`.

[3] Mathieu Desbrun and Marie-Paule Gascuel. "Smoothed particles: A new paradigm for animating highly deformable bodies". In: *Computer Animation and Simulation'96*. Springer, 1996, pp. 61–76.

[4] R. A. Gingold and J. J. Monaghan. "Smoothed particle hydrodynamics: theory and application to non-spherical stars". In: *Monthly Notices of the Royal Astronomical Society* 181.3 (Dec. 1977), pp. 375–389. ISSN: 0035-8711. DOI: `10.1093/mnras/181.3.375`. eprint: `https://academic.oup.com/mnras/article-pdf/181/3/375/3104055/mnras181-0375.pdf`. URL: `https://doi.org/10.1093/mnras/181.3.375`.

[5] Goswami, Prashant and Batty, Christopher. "Regional Time Stepping for SPH". In: (2014). URL: `http://hdl.handle.net/10012/11857`.

[6] Julian Groß, Marcel Köster, and Antonio Krüger. "Fast and Efficient Nearest Neighbor Search for Particle Simulations." In: *CGVC*. 2019, pp. 55–63.

[7] Takahiro Harada, Seiichi Koshizuka, and Yoichiro Kawaguchi. "Smoothed particle hydrodynamics on GPUs". In: *Computer Graphics International*. Vol. 40. SBC Petropolis. 2007, pp. 63–70.

[8] A. Hérault et al. *GPU-SPH*. http://www.ce.jhu.edu/dalrymple/GPU/GPUSPH/Home.html.

[9] Alexis Hérault, Giuseppe Bilotta, and Robert A Dalrymple. "Sph on gpu with cuda". In: *Journal of Hydraulic Research* 48.sup1 (2010), pp. 74–79.

[10] Markus Ihmsen et al. "A Parallel SPH Implementation on Multi-Core CPUs". In: *Computer Graphics Forum* 30.1 (2011), pp. 99–112. DOI: `https://doi.org/10.1111/j.1467-8659.2010.01832.x`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2010.01832.x`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2010.01832.x`.

[11] Kedar G Kale and Adrian J Lew. "Parallel asynchronous variational integrators". In: *International Journal for Numerical Methods in Engineering* 70.3 (2007), pp. 291–321.

[12] L. B. Lucy. "A numerical approach to the testing of the fission hypothesis." In: 82 (Dec. 1977), pp. 1013–1024. DOI: `10.1086/112164`.

[13] J.J. Monaghan. “Simulating Free Surface Flows with SPH”. In: *Journal of Computational Physics* 110.2 (1994), pp. 399–406. ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1994.1034. URL: https://www.sciencedirect.com/science/article/pii/S0021999184710345.

[14] Stefan Reinhardt et al. “Fully Asynchronous SPH Simulation”. In: *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. SCA ’17. Los Angeles, California: Association for Computing Machinery, 2017. ISBN: 9781450350914. DOI: 10.1145/3099564.3099571. URL: https://doi.org/10.1145/3099564.3099571.

[15] Eugenio Rustico et al. “A journey from single-GPU to optimized multi-GPU SPH with CUDA”. In: *7th SPHERIC Workshop*. 2012, p. 56.

[16] B. Solenthaler and R. Pajarola. “Predictive-Corrective Incompressible SPH”. In: *ACM SIGGRAPH 2009 Papers*. SIGGRAPH ’09. New Orleans, Louisiana: Association for Computing Machinery, 2009. ISBN: 9781605587264. DOI: 10.1145/1576246.1531346. URL: https://doi.org/10.1145/1576246.1531346.